Semidefinite Programming. Methods and algorithms for energy management

Agnès Gorge Maher

UNIVERSITE PARIS-SUD

ECOLE DOCTORALE INFORMATIQUE DE PARIS-SUD (ED 427)

Laboratoire de Recherche en Informatique (LRI)

DISCIPLINE INFORMATIQUE

THESE DE DOCTORAT

soutenue le 26/09/2013 par

Agnès Gorge

Semidefinite Programming : methods and algorithms for energy management

Directeur de thèse : Abdel LISSER Professeur (LRI - Université Paris-Sud)

Composition du jury :

Président du jury :		
Rapporteurs :	Franz RENDL	Professeur (Université Alpen-Adria, Klagenfurt)
	Didier HENRION	Directeur de recherche (LAAS CNRS)
Examinateurs :	Alain DENISE	Professeur (LRI - Université Paris-Sud)
	Abdel LISSER	Professeur (LRI - Université Paris-Sud)
	Abdelatif MANSOURI	Professeur (Université Cadi Ayyad, Marrakech)
	Michel MINOUX	Professeur (LIP6 - UPMC)
	Riadh ZORGATI	Docteur (EDF R & D)

Abstract

This thesis aims at exploring the potentialities of a powerful optimization technique, namely Semidefinite Programming, for addressing some difficult problems of energy management. This relatively young area of convex and conic optimization has undergone a rapid development in the last decades, partly thanks to the design of efficient algorithms for their resolution, and because numerous NP-hard problems can be approached using semidefinite programming.

In the present thesis, we pursue two main objectives. The first one consists of exploring the potentiality of semidefinite programming to provide tight relaxations of combinatorial and quadratic problems. This line of research was motivated both by the promising results obtained in this direction [108, 186, 245] and by the combinatorial and quadratic features presented by energy management problems. The second one deals with the treatment of uncertainty, an issue that is also of paramount importance in energy management problems. Indeed, due to its versatility, SDP is well-known for providing numerous possibilities of dealing with uncertainty. In particular, it offers a way of modelling the deterministic counterpart of robust optimization problems, or more originally, of distributionnally robust optimization problems.

The first part of this thesis contains the theoretical results related to SDP, starting by the underlying theory of convex and conic optimization, followed by a focus on semidefinite programming and its most famous applications. In particular, we provide a comprehensive and unified framework of the different methods proposed in the literature to design SDP relaxations of QCQP

The second part is composed of the last three chapters and presents the application of SDP to energy management. Chapter 4 provides an introduction to energy management problems, with a special emphasis on one of the most challenging energy management problem, namely the Nuclear Outages Scheduling Problems. This problem was selected both for being a hard combinatorial problem and for requiring the consideration of uncertainty. We present at the end of this chapter the different models that we elaborated for this problem.

The next chapter reports the work related to the first objective of the thesis, i.e., the design of semidefinite programming relaxations of combinatorial and quadratic programs. On certain problems, these relaxations are provably tight, but generally it is desirable to reinforce them, by means of tailor-made tools or in a systematic fashion. We apply this paradigm to different models of the Nuclear Outages Scheduling Problem. Firstly, we consider a complete model that takes the form of a MIQP. We apply the semidefinite relaxation and reinforce it by addition of appropriate constraints. Then, we take a step further by developing a method that automatically generates such constraints, called cutting planes. For the design of this method, we start by providing a framework for unification of many seemingly disparate cutting planes that are proposed in the literature by noting that all these constraints are linear combinations of the initial quadratic constraints of the problem and of the pair-wise product of the linear constraints of the problem (including bounds constraints).

Subsequently, we focus on specific part of the problem, namely the maximal lapping constraint, which takes the form of $a^T x \notin [b, c]$, where x are binary variables. This constraint presents modelling difficulty due to its disjunctive nature. We aim at comparing three possible modelisations and for each of them, computing different relaxations based on semidefinite programming and linear programming. Finally, we conclude this chapter by an experiment of the Lasserre's hierarchy, a very powerful tool dedicated to polynomial optimization that builds a sequence of semidefinite relaxations whose optimal values tends to the optimal value of the considered problem.

Thus, we fulfilled the first objective of this thesis, namely exploring the potentiality of semidefinite programming to provide tight relaxations of combinatorial and quadratic problems. The second objective is to examine how semidefinite programming can be used to tackle uncertainty. To this end, three different works are carried out. First, we investigate a version of the nuclear outages scheduling

problem where uncertainty is described in the form of equiprobable scenarios and the constraints involving uncertain parameters have to be satisfied up to a given level of probability. It is well-known that this model admits a deterministic formulation by adding binary variables. Then the obtained problem is a combinatorial problem and we apply semidefinite programming to compute tight bounds of the optimal value.

We have also implemented a more original way of dealing with uncertainty, which admits a deterministic counterpart, or a conservative approximation, under the form of a semidefinite program. This method, that has received much attention recently, is called *distributionnally robust optimization* and can be seen as a compromise between stochastic optimization, where the probability distribution is required, and robust optimization, where only the support is required. Indeed, in distributionnally robust optimization, the support and some moments of the probability distribution are required. In our case, we assume that the support, the expected value and the covariance are known and we compare the benefits of this method w.r.t other existing approaches, based on Second-Order Cone Program, that rely on the application of the Boole's inequality, to convert the joint constraint into individual ones, combined to the Hoeffding's inequality, in order to get a tractable conservative approximation of the chance constraints.

Finally, we carried out a last experiment that combines both uncertainty and combinatorial aspects. Indeed, many deterministic counterpart or conservative approximation of Linear Program (LP) subject to uncertainty give rise to a Second-Order Cone Program (SOCP). In the case of a Mixed-Integer Linear Program, we obtain a MISOCP, for which there is no reference resolution method. Then we investigate the strength of the SDP relaxation for such problems. Central to our approach is the reformulation as a non convex Quadratically Constrained Quadratic Program (QCQP), which brings us in the framework of binary quadratically constrained quadratic program. This allows to apply the well-known semidefinite relaxation for such problems. When necessary, this relaxation is tightened by adding constraints of the initial problem. We report promising computational results indicating that the semidefinite relaxation improves significantly the continuous relaxation (112% on average) and often provides a lower bound very close to the optimal value. In addition, computational time for obtaining these results remains reasonable.

In conclusion, despite practical difficulties mainly due to the fact that SDP is not a mature technology yet, it is nonetheless a very promising optimization method, that combines all the strengths of conic programming and offers great opportunities for innovation at EDF R&D, both in energy management and engineering or financial issues.

Résumé

Cette thèse se propose d'explorer les potentialités qu'offre une méthode prometteuse de l'optimisation convexe et conique, la programmation semi-définie positive (SDP), pour les problèmes de management d'énergie.

La programmation semi-définie positive est en effet l'une des méthodes ayant le plus attiré l'attention de la communauté scientifique ces dernières années, du fait d'une part de la possibilité de pouvoir résoudre ses instances en temps polynomial grâce à des solveurs performants. D'autre part, il s'est avéré que de nombreux problèmes d'optimisation NP-difficiles peuvent être approximés au moyen d'un SDP.

Ce rapport débute par un résumé des principaux résultats relatifs à ce domaine de l'optimisation. Le chapitre 1 contient un rappel des fondamentaux de l'optimisation convexe et conique, puis nous présentons les bases théoriques et les algorithmes de la SDP dans le chapitre 2. Enfin, dans le chapitre 3 nous décrivons les applications de la SDP qui présentent le plus d'intérêt dans ce contexte. En particulier, nous proposons une vision claire et unifiée des différentes méthodes recensées dans la littérature pour construire les relaxations SDP de problèmes quadratiques et combinatoires.

Les applications de la SDP au management d'énergie constituent la seconde partie de ce rapport. Le management d'énergie est pr'esent'e au chapitre 4, avec une attention particulière portée au problème de planification des arrêts nucléaires. Le chapitre 5 est consacré au premier axe de cette thèse, visant à utiliser la SDP pour produire des relaxations de problèmes combinatoires et quadratiques, comme suggéré par de nombreux résultats prometteurs dans ce domaine. Si une première relaxation SDP, dénommée relaxation SDP standard, peut-être obtenue très simplement, il est généralement souhaitable de renforcer cette dernière par un ajout de contraintes valides, pouvant être déterminées par l'étude de la structure du problème ou à l'aide de méthodes plus systématiques.

En particulier, nous expérimentons la relaxation SDP sur différentes modélisations du problème de planification des arrêts nucléaires, réputé pour sa difficulté combinatoire. Nous commençons par étudier une modélisation proche de celle utilisée à l'opérationnel, donnant lieu à un MIQP, sur lequel la relaxation SDP standard est appliquée , puis renforcée au moyen d'un procédé classique. Puis nous proposons une méthode plus systématique permettant de déterminer automatiquement une contrainte appropriée à ajouter au problème afin de renforcer la relaxation SDP. Cette méthode repose sur le constat que toutes les contraintes proposées dans la littérature pour renforcer la relaxation SDP peuvent être vues comme des combinaisons linéaires de contraintes quadratiques et de produits deux à deux des contraintes linéaires du problème, y compris les contraintes de borne. Alors, parmi toutes ces contraintes, il suffit de sélectionner la plus violée par la relaxation SDP courante.

Dans la suite, nous nous intéressons à une contrainte particulière du problème de planification des arrêts nucléaires, à savoir la contrainte de recouvrement maximal entre arrêts, pouvant être formulée de la façon suivante : $a^T x \notin [b, c]$, avec x un vecteur de variables binaires. Cette contrainte disjonctive admet plusieurs modélisations. Nous avons donc comparé pour chacune d'entre elles, un ensemble de relaxations, à la fois linéaires et SDP. Enfin, nous concluons ce chapitre par une expérimentation de la hiérarchie de Lasserre. Cette théorie très puissante considère un problème d'optimisation polynomial quelconque et construit une suite de SDP dont la valeur optimale tend vers la solution du problème initial.

Le second axe de la thèse, portant sur l'application de la SDP à la prise en compte de l'incertitude, donne lieu à 3 études. Dans la première, nous travaillons sur une version du problème des arrêts nucléaires dans laquelle l'incertitude se présente sous la forme de scénarios équiprobables et les contraintes concernées par les incertitudes sont à satisfaire en probabilité. Il est alors classique de formuler ce problème de façon déterministe en ajoutant une variable binaire par contrainte et par scénario, ce qui donne lieu à un grand problème combinatoire. Nous pouvons alors appliquer la relaxation SDP à ce problème, selon l'approche présentée dans le premier axe de cette thèse.

Nous avons également mis en oeuvre une méthode plus originale pour la prise en compte de l'incertitude, permettant de reformuler le problème, ou d'en donner une approximation conservative, sous la forme d'un SDP. Cette méthode, qui a fait l'objet de nombreux travaux récemment, est connue sous le nom d'optimisation distributionnellement robuste. Il s'agit en fait d'un compromis entre l'optimisation stochastique, qui nécessite une connaissance parfaite des lois de probabilités utilisées, et l'optimisation robuste, qui ne requiert que la connaissance du support des variables aléatoires. En effet, l'optimisation distributionnellement robuste ne nécessite pas de connaître la distribution de probabilité, mais uniquement son support et certains de ses moments. Nous appliquons donc cette méthode à un problème d'équilibre offre-demande dans lequel la demande et la disponibilité des moyens de production sont soumis à des aléas, dont on connaît le support, l'espérance et la covariance. Nous nous appliquons donc à estimer l'apport de cette méthode par rapport à une méthode de type robuste basée sur la connaissance du support et de l'espérance, permettant de formuler une approximation conservative du problème comme un SOCP par l'application des inégalités de Boole et de Hoeffding.

En dernier lieu, nous procédons à une expérimentation combinant les deux approches explorés dans ce rapport. En effet, de nombreux problèmes à données aléatoires admettent un équivalent, ou une approximation, pouvant s'écrire sous la forme d'un SOCP. Dans le cas où le problème initial contient des variables entières, le problème obtenu est alors un MISOCP, pour lesquels il n'existe pas de méthodes de résolution de référence. Nous nous intéressons ici à l'utilisation de relaxations SDP pour ce type de problème. Le principe est de convertir le MISOCP en MIQCQP, puis d'appliquer la relaxation SDP standard, qui est ensuite renforcée par l'ajout de contraintes du problème initial mises au format SDP. Cette approche donne des résultats encourageants, avec une relaxation SDP nettement meilleure que la relaxation continue. Les solutions obtenues sont même très proches de l'optimal sur de nombreuses instances tout en conservant un temps de calcul raisonnable.

En conclusion, en dépit de nombreuses difficultés pratiques, imputables au fait que cette technologie n'est pas encore tout à fait mature, la SDP n'en reste pas moins une méthode extrêmement prometteuse, combinant toute la puissance de l'optimisation conique. Elle offre de nombreuses opportunités d'innovation, aussi bien en management d'énergie que dans d'autres domaines tels que l'ingénierie ou la gestion de portefeuille d'actifs financiers.

Acknowledgement

First I would like to express my deepest gratitude to my thesis supervisor Professor Abdel Lisser for its excellent supervision and infinite help throughout my thesis. I am also deeply thankful to Doctor Riadh Zorgati that supervised this thesis from EDF side. Both have believed in this project from the beginning, and without them, nothing would have been achieved. I also thank them for their constant logistic support, their ideas and encouragement, as well as for all of the laughs and good times that we shared. I am also greatly indebted to Professor Abdelatif Mansouri for welcoming me in its department at the Cadi Ayyad University in Marrakech.

I am deeply thankful for Professor Didier Henrion and Professor Franz Rendl for their effort of reviewing this thesis and for their constructive criticism concerning the material of this dissertation. I extend my thanks to the others members of the jury, Professor Michel Minoux and Sandrine Charousset for accepting to read and evaluate my work.

I am obliged to the management team of the department OSIRIS EDF R& D, Yannick Jacquemart, François-Régis Monclar, Sandrine Charousset, Ala Ben Abbes, Marc Ringeisen and Fabrice Chauvet, for offering me this great opportunity and for arranging and supporting this work.

My appreciation also goes to all the LRI members and staff, as well as the CEPHYTEN team, for their constant kindness, availability and support.

I would like to thank all my colleagues from R36 and LRI who make my moments in Clamart and Orsay full of interest and pleasure. Thank you also to all the PhD students that motivate me to pursue a PhD. I also owe a debt of gratitude to the Sebti family for their invaluable help, their constant kindness and good mood, all the small and long talks, and for the friday's couscous !

Last but not least, I would like to thank all my family members : my grand-parents, parents and sisters and my husband and son, for their love, support and much more besides during all these years.

Contents

List of Figures

List of Tables

Synthèse

Cette thèse a pour objet d'évaluer l'apport de la programmation semi-définie positive (SDP), méthode prometteuse de l'optimisation conique, pour la résolution pratique des problèmes d'optimisation rencontrés en management d'énergie. Elle est motivée par les résultats récents reportés dans la littérature exhibant un fort potentiel de la SDP pour le traitement des problèmes à caractère combinatoire et/ou aléatoire marqué tels que ceux fréquemment rencontrés management d'énergie, et pour lesquels les méthodes classiques de résolution sont limitées. La thèse a pour ambition d'évaluer ce qu'une méthode alternative comme la SDP pourrait apporter quant à la résolution de ces problèmes en général, avec une attention particulière portée au problème de la planification des arrêts des centrales nucléaires pour rechargement du combustible et maintenance.

Le travail a consisté à identifier les problèmes concernés par notre démarche, à les modéliser de façon appropriée et à expérimenter la mise en oeuvre numérique de leur résolution à l'aide de la SDP. Il est décliné selon deux axes. Concernant le premier, nous investiguons, d'un point de vue théorique et numérique, les potentialités de la SDP pour l'élaboration de relaxations performantes de problèmes NP-difficiles présentant un caractère combinatoire et/ou quadratique. Concernant le second, nous exploitons la puissance de modélisation de la SDP pour la prise en compte de la nature aléatoire des problèmes d'optimisation. Afin de préciser le contexte scientifique de la thèse, nous rappelons préalablement quelques éléments d'introduction à la programmation conique et plus particulièrement à la SDP. Puis nous présentons les caractéristiques des problèmes d'optimisation rencontrés en management d'énergie et développons différentes approches pour leur traitement par la SDP.

Introduction à l'optimisation conique et à la SDP

L'optimisation conique peut être vue comme une extension naturelle de la programmation linéaire dans laquelle l'orthant positif de $\mathbb{R}^n$ est remplacé par un cône convexe $\mathcal{K}$. Ce formalisme présente de nombreux avantages, en particulier le problème dual admet également une formulation conique faisant intervenir $\mathcal{K}^*$, le cône dual de $\mathcal{K}$, ce qui confère au problème d'intéressantes propriétés de symétrie et une extension des théorèmes de dualité faible et forte de la programmation linéaire.

Si $\mathcal{K} \subset \mathbb{R}^n$, alors un problème conique (P_P) et son dual (P_D) sont définis par la donnée d'un vecteur $c \in \mathbb{R}^n$ et d'une matrice $(A, b) \in \mathbb{R}^{m,n+1}$, de la façon suivante :

$$(P_P) \begin{cases} \inf & c^T x \\ \text{s.t.} & Ax = b \\ & x \in \mathcal{K} \end{cases} \qquad (P_D) \begin{cases} \sup & b^T z \\ \text{s.t.} & y = c - A^T z \\ & y \in \mathcal{K}^* \end{cases}$$

L'optimisation conique bénéficie des bonnes propriétés de l'optimisation convexe, notamment concernant sa complexité. Ainsi, par une application directe du résultat de Grötschel et al. [118], la solution optimale peut être approchée aussi finement que voulu en temps polynomial à condition qu'il existe un oracle de séparation polynomial pour $\mathcal{K}$. La méthode utilisée, dite *des ellipsoïdes*, est donc fondamentale sur le plan théorique. En pratique, elle s'est révélée peu performante et a rapidement été supplantée par d'autres méthodes polynomiales plus efficaces. En particulier, selon le résultat fondamental de Nesterov et Nemirovski [206], il est possible d'étendre les méthodes de points-intérieurs, initialement conçues pour la programmation linéaire, à n'importe quel problème convexe du moment qu'il existe une fonction barrière pour l'ensemble réalisable exhibant une propriété de régularité spécifique dite d'*auto-concordance*.

La programmation semi-définie positive (SDP) est le cas particulier de la programmation conique dans lequel le cône $\mathcal{K}$ est $\mathbb{S}^n_+$, c'est-à-dire l'ensemble des matrices semi-définies positives. Rappelons qu'une matrice X est semi-définie positive, ce qui est noté $X \succcurlyeq 0$, si elle est symétrique et si toutes ses valeurs propres sont positives ou nulles. Une autre définition fréquemment utilisée est la suivante : $X \in \mathbb{S}^n_+ \Leftrightarrow u^T X u \geq 0,\ \forall u \in \mathbb{R}^n$.

Un SDP primal (SDP_P) et son dual (SDP_D) se présentent donc de la façon suivante :

$$(SDP_P)\begin{cases}\inf & A_0 \bullet X \\ \text{s.t.} & A_j \bullet X = b_j,\ j = 1,...,m \\ & X \succcurlyeq 0\end{cases} \qquad (SDP_D)\begin{cases}\sup & b^T z \\ \text{s.t.} & A_0 - \sum_{j=1}^{m} A_j z_j \succcurlyeq 0\end{cases}$$

où $\bullet$ désigne le produit scalaire sur les matrices symétriques, à savoir $M \bullet N = \sum_{i=1}^{n}\sum_{j=1}^{n} M_{ij}N_{ij}$, pour $M, N \in \mathbb{S}^n$.

Parmi les méthodes de l'optimisation conique, la SDP se positionne comme la méthode polynomiale possédant la plus grande puissance de modélisation. En particulier, elle subsume d'autres méthodes connues de l'optimisation conique telles que la programmation linéaire (LP) et l'optimisation conique du second-ordre (SOCP). Ces caractéristiques attrayantes ont suscité un grand intérêt parmi la communauté scientifique ces dernières années. Il en résulte de nombreux travaux portant aussi bien sur la théorie sous-jacente à ces problèmes que sur leurs méthodes de résolution et leur applicabilité.

Concernant l'aspect théorique, de nombreux résultats ont été établis sur la géométrie des SDP, c'est-à-dire sur la caractérisation de leurs ensembles réalisables primal et dual. Il en ressort des résultats sur la caractérisation des points extrêmes de ces ensembles (en particulier, leur rang) [211], sur l'unicité des solutions optimales [10] ou encore sur la dimension de leurs faces et facettes [25]. Des travaux poussés ont également été menés sur la dualité des problèmes SDP [29], débouchant sur l'identification de 11 configurations possibles, différant par la présence ou non de la dualité forte et par le fait que les valeurs optimales primale et duale existent ou non, et sont atteintes ou non.

Les méthodes de résolution ont également fait l'objet de nombreux travaux. Les algorithmes de points-intérieurs restent à ce jour les plus étudiés et les plus usités pour leur efficacité et leur applicabilité à n'importe quel SDP. Initiés par Alizadeh en 1991 [7], ces méthodes ont donné lieu aux deux solveurs SDP les plus réputés, à savoir CSDP [53] et DSDP [34]. D'autres méthodes issues de la programmation non linéaire ont également été testées. Parmi elles, citons les méthodes de relaxation lagrangienne [158] et les méthodes de faisceaux [129].

L'intérêt pour la SDP s'est encore accru ces dernières années lorsque de nombreuses applications ont été identifiées dans des domaines variés tels que le contrôle, les statistiques, la finance, la localisation, l'optimisation robuste et l'ingénierie. Parmi toutes ces applications, l'utilisation de la SDP pour approximer le célèbre problème des moments généralisés (GPM), a particulièrement attiré l'attention de la communauté scientifique, de par la généricité de ce problème et son applicabilité à l'optimisation quadratique et combinatoire. Considérons par exemple le problème quadratique suivant :

$$(QCQP)\begin{cases}\min & x^T P_0 x + 2p_0^T x + \pi_0 \\ \text{s.t.} & x^T P_j x + 2p_j^T x + \pi_j \leq 0,\ j = 1,...,m\end{cases} \quad \text{avec } P_j \in \mathbb{S}^n, p_j \in \mathbb{R}^n, \pi_j \in \mathbb{R},\ j = 0,...,m \tag{1}$$

Ce problème à objectif et contraintes quadratiques (QCQP) est convexe si et seulement $P_j \succcurlyeq 0,\ j = 0,...,m$. Autrement, il appartient à la classe des problèmes NP-difficiles. Il suffit pour le comprendre de remarquer que de nombreux problèmes difficiles peuvent se mettre sous cette forme, en particulier les problèmes combinatoires à variables binaires puisque $x_i \in \{0,1\}$ est équivalent à la contrainte quadratique $x_i^2 - x_i = 0$. On peut établir très simplement une relaxation SDP de ce problème, dite *standard*, de la façon suivante :

$$\begin{cases}\inf & Q_0 \bullet Y \\ \text{s.t.} & Q_j \bullet Y \leq 0,\ i = 1,...,m \\ & Y_{1,1} = 1 \\ & Y \succcurlyeq 0\end{cases} \quad \text{avec } Q_j = \begin{pmatrix}\pi_j & p_j^T \\ p_j & P_j\end{pmatrix},\ j = 0,...,m \tag{2}$$

Cette relaxation peut être obtenue et interprétée de nombreuses façons, la plus simple d'entre elles consistant à introduire une nouvelle variable $Y = \tilde{x}\tilde{x}^T$, à remplacer les formes quadratiques $x^T P_j x + 2p_j^T x + \pi_j$ par leur équivalent $Q_j \bullet Y$, puis à relaxer la contrainte $Y = \tilde{x}\tilde{x}^T$ en $Y_{1,1} = 1$ et $Y \succcurlyeq 0$.

L'idée d'une relaxation SDP est attribuée à Lovász [186] et à Shor [245], mais ce sont les travaux de Goemans et Williamsons [108] offrant une garantie sur l'optimalité de la borne ainsi obtenue dans le cas d'un $\{-1,1\}$-QP, qui ont déclenché le véritable engouement que l'on connaît pour la SDP.

Cependant, dans le cas d'un QCQP quelconque, il est généralement nécessaire de renforcer cette relaxation standard pour la rendre véritablement efficace. Pour cela, il suffit d'ajouter des contraintes valides au problème quadratique initial, puis d'appliquer la relaxation SDP standard à ce nouveau QCQP. Toute la difficulté réside donc dans l'identification des contraintes valides (ou coupes) les plus efficaces. Cette façon de voir unifie de nombreux travaux recensés dans la littérature, voir par exemple [7, 127, 176, 177, 230].

Une autre source importante d'application pour la SDP provient de sa capacité à approximer le problème des moments généralisés (GPM). Ce problème (GPM_P) et son dual (GPM_D) se définissent de la façon suivante :

$$(GPM_P)\begin{cases} \min & \int_{\mathcal{S}} h(\omega)\mathrm{P}(\omega)d\omega \\ \text{s.t.} & \int_{\mathcal{S}} f_i(\omega)\mathrm{P}(\omega)d\omega = b_i,\ i=1,...,m \\ & \mathrm{P}\in\mathcal{M}(\mathcal{S}) \end{cases} \qquad (GPM_D)\begin{cases} \max & \sum_{i=1}^{m} b_i z_i \\ \text{s.t.} & \sum_{i=1}^{m} f_i(\omega)z_i \le h(\omega),\quad \forall\omega\in\mathcal{S} \end{cases}$$

Dans le primal, la variable d'optimisation n'est pas un vecteur euclidien comme c'est le cas généralement, mais P, une mesure positive ou nulle sur $\mathcal{B}(\mathcal{S})$, la σ-algèbre de Borel de $\mathcal{S}\subset\mathbb{R}^n$. Il est cependant possible de la considérer comme un vecteur de dimension infinie, dans lequel chaque composante correspond à une valeur de $\mathrm{P}(\omega)$ pour tout $\omega\in\mathcal{S}$. Le problème devient alors linéaire et le dual se déduit simplement comme le dual d'un programme linéaire, le nombre infini de variables induisant un nombre infini de contraintes. Prises ensemble, ces contraintes prennent la forme de la positivité sur $\mathcal{S}$ de la fonction $f_z(\omega)=h(\omega)-\sum_{i=1}^{m} f_i(\omega)z_i$ dont les coefficient dépendent de la variable duale $z\in\mathbb{R}^m$.

De nombreux problèmes peuvent se modéliser comme des instances particulières de ce problème, mais l'intérêt n'en est que purement théorique car il n'existe pas de méthode de résolution générale connue pour ce problème. Toutefois, deux restrictions se révèlent très intéressantes puisqu'il existe alors une suite d'approximations SDP dont la valeur optimale tend vers la valeur initiale du (GPM). Ces restrictions sont les suivantes : on suppose premièrement que $\mathrm{P}(\mathcal{S})$ est borné et on va prendre $P(\mathcal{S})=1$, ce qui revient à supposer que P est une mesure de probabilité. Deuxièmement, on se place dans un cadre polynômial, où les fonctions $h, f_i,\ i=1,...,m$ sont des polynômes et où $\mathcal{S}$ est un ensemble semi-algébrique.

C'est de ces restrictions que le problème tire son nom, puisque le primal se formule alors via des combinaisons linéaires de moments de P et toute la difficulté du problème se retrouve dans la dernière contrainte de la formulation ci-dessous, imposant à y d'être le vecteur des moments associés à P :

$$\begin{cases} \min & \sum_{\kappa\in\mathbb{N}^n_d} \mathsf{h}_\kappa y_\kappa \\ \text{s.t.} & \sum_{\kappa\in\mathbb{N}^n_d} \mathsf{f}_{\mathsf{i}\kappa} y_\kappa = b_i,\ i=1,...,m \\ & y_\kappa = \mathrm{E}_{\mathrm{P}}\left(\omega^\kappa\right),\ \kappa\in\mathbb{N}^n_d \qquad (\text{ signifie que } y \text{ est le vecteur des moments de P}) \\ & \mathrm{P}\in\mathcal{M}(\mathcal{S}) \end{cases}$$

où $\mathbb{N}^n_d=\{\kappa\in\mathbb{N}^n : \sum_{i=1}^n \kappa_i\le d\}$ et pour tout polynôme f de degré d sur $\mathbb{R}^n$, f est le vecteur de ses coefficients, c'est-à-dire $f(x)=\sum_{\kappa\in\mathbb{N}^n_d}\mathsf{f}_\kappa x^\kappa$ où $x^\kappa=\prod_{i=1}^n x_i^{\kappa_i}$ est un monôme.

Or il existe une relaxation SDP de la contrainte imposant à y d'être un vecteur de moment sur $\mathcal{S}$, via la semi-définie positivité d'une matrice dépendant de y et de $\mathcal{S}$. Il s'avère que le dual de ce SDP peut également être interprété comme une approximation (conservative cette fois) de (GPM_D),

en appliquant le théorème de Putinar. En effet, ce dernier établit des conditions suffisantes pour la positivité d'un polynôme sur un ensemble semi-algébrique, qui font appel à des matrices semi-définie positives. De plus, ces relaxations dépendent d'un paramètre entier r, dit le rang, qui lorsqu'il tend vers l'infini, fait tendre la valeur optimal des SDP ainsi obtenus vers la valeur optimale du problème initial. Cette séquence de SDP est connue sous le nom de *hiérarchie de Lasserre* [171].

Parmi les instances que nous pourrons ainsi approximer, se trouve le problème classique des moments (CPM), dans lequel les fonctions f_i sont les monômes de degré inférieur à d et h est la fonction indicatrice d'un certain ensemble $\mathcal{K}$. Ce problème revient à minimiser la probabilité $\mathrm{P}[\omega \in \mathcal{K}]$ connaissant les moments d'ordre inférieur à d et le support de P. De nombreuses variantes de ce problème ont été étudiés par le passé, donnant lieu à des inégalités célèbres telles que les inégalités de Chebyshev ou de Markov.

En conclusion, même si les applications de la SDP sont légions, elles sont difficiles à identifier car la contrainte d'imposer à une matrice d'être semi-définie positive n'apparaît pas naturellement. Afin de prendre du recul sur la façon dont ces applications émergent, nous proposons une classification de ces processus en 3 catégories :

- identification d'une matrice SDP : lorsque l'une des définitions possibles d'une matrice SDP se retrouve clairement dans le problème. C'est le cas par exemple si l'on requiert d'une fonction quadratique en x d'être positive ou nulle pour tout x : $f(x,y) = \tilde{x}^T P(y)\tilde{x} \geq 0,\ \forall x \in \mathbb{R}^n \Leftrightarrow P(y) \succcurlyeq 0$;
- relaxation : lorsqu'on relaxe la contrainte $Y \in \mathcal{S}$, avec $\mathcal{S} \subset \mathbb{S}^n_+$, par $Y \succcurlyeq 0$. Exemple : $\mathcal{S} = \{xx^T : x \in \mathbb{R}^n\}$;
- exploitation d'un résultat reposant sur l'existence d'une matrice SDP. Exemple : f convexe $\Leftrightarrow \nabla^2 f \succcurlyeq 0$.

Parmi les résultats reposant sur l'existence d'une matrice SDP se trouve le S-Lemma [212], d'importance cruciale pour notre étude. Ce lemme donne en effet une condition suffisante, parfois nécessaire, pour qu'une contrainte quadratique soit valide sur un ensemble défini par des contraintes quadratiques. Plus concrètement, si l'on cherche une matrice Q telle que la contrainte $\tilde{x}Q\tilde{x} \geq 0$ soit valide sur $\mathcal{K} = \{x \in \mathbb{R}^n : \tilde{x}Q_j\tilde{x} \geq 0,\ j = 1,...,m\}$, on peut approximer cette condition de façon conservative par l'application du S-Lemma, c'est-à-dire par la condition qu'il existe $\lambda_j \geq 0,\ j = 1,...,m$ tels que $Q - \sum_{j=1}^m \lambda_j Q_j \succcurlyeq 0$. Cette approximation conservative est au coeur de la relaxation SDP de problèmes quadratiques et d'approximation des problèmes de moments d'ordre 2.

Présentation des problèmes de management d'énergie

Ce paragraphe vise à introduire le management d'énergie, avec une attention particulière portée au problème de planification des arrêts des centrales nucléaires.

Le management d'énergie regroupe l'ensemble des problèmes relatifs à la production, à l'approvisionnement, au transport et à la consommation d'énergie, plus particulièrement l'électricité et le gaz naturel. De façon très simplifiée, on s'attache à satisfaire l'équilibre offre-demande à tout moment et sur tout point du réseau, à moindre coût. Du fait de l'importance stratégique de l'énergie dans notre société, les enjeux associés à ces problèmes sont colossaux, aussi bien sur le plan économique qu'industriel, social et écologique. De par sa taille, le problème d'optimisation sous-jacent est un problème difficile. A ceci s'ajoute un contexte géo-économico-politique très changeant, pour ne pas dire aléatoire, la nécessité de prendre en compte de nombreuses subtilités technologiques sur le fonctionnement des moyens de production et de transport, la modélisation de mécanismes de marché complexes, soumis à une demande très inélastique, et l'importance de l'impact climatique à la fois sur la consommation et sur les moyens de productions. Concernant la gestion de l'électricité, à laquelle nous nous limiterons désormais, le levier de gestion que constitue habituellement la constitution de réserve est très contraint. En effet, il n'est pas rentable de stocker l'énergie directement sous forme électrique (batteries, ...). En revanche, les stocks d'eau des barrages constituent une réserve d'électricité puisqu'on peut facilement (et gratuitement) les

convertir sous cette forme. La difficulté est que la majeure partie de cette réserve provient des apports climatiques et est donc soumise à des aléas.

Le respect de l'équilibre offre-demande donne lieu à un gigantesque problème d'optimisation. Afin de le pouvoir traiter, une première décomposition consiste à regrouper les variables de décisions par horizon de temps :

- Sur le long-terme (de 10 à 20 ans) sont prises les décisions d'investissement qui déterminent la structure du portefeuille;
- Au moyen-terme (de 1 à 5 ans), il est nécessaire de planifier l'utilisation de certains actifs, comme les centrales nucléaires, et en particulier leurs arrêts pour rechargement et maintenance (ce problème sera détaillé par la suite), les stratégies d'utilisation des barrages, ou encore de souscrire certains contrats d'approvisionnement;
- A court-terme se réalise l'équilibre offre-demande proprement dit, via l'élaboration des programmes de fonctionnement de chaque centrale, complétés par des achats/ventes sur les marchés de l'électricité.

Cette décomposition donne lieu à un grand nombre de sous-problèmes d'optimisation, chacun se distinguant par une finesse de modélisation différente et des difficultés variées.

Parmi elles, et ce d'autant plus que le problème se situe à long-terme, se trouve la prise en compte des incertitudes pouvant affecter les données d'entrées, comme les aléas climatiques, les indisponibilités des moyens de production ou les prix sur les différents marchés de l'énergie. Ces données sont autant de variables aléatoires à prendre en compte dans les modèles, dont il est difficile de déterminer précisément la distribution de probabilité du fait de la complexité des processus impliqués. Cependant, les observations historiques de ces processus nous fournissent une connaissance partielle de ces lois de distributions, menant aux représentations suivantes :

- Une approximation déterministe utilisant la valeur moyenne ou la valeur dans le pire cas;
- Une représentation robuste établissant que la variable évolue dans un ensemble donnée;
- Une représentation "distributionnellement robuste" dans laquelle on suppose connus le support et les k premiers moments de la distribution de probabilité;
- Une représentation stochastique dans laquelle on suppose connue la distribution de probabilité. En particulier, c'est le cas lorsqu'on utilise des scénarios de réalisations de la variable, issus par exemple des observations historiques.

Parmi les différents problèmes de management d'énergie, nous nous intéressons particulièrement au problème de la planification des arrêts des centrales nucléaires.

L'objectif de ce problème est de déterminer, sur un horizon de temps à moyen-terme (1 à 5 ans), le meilleur moment pour arrêter les réacteurs afin d'y effectuer les opérations nécessaires de rechargement en combustible et de maintenance, de façon à perturber le moins possible la satisfaction de l'équilibre offre-demande. De par l'importance de la production nucléaire en France, ce problème présente des enjeux financiers importants. Il est difficile à résoudre du fait de sa nature combinatoire, liée à la modélisation de l'état "en marche" ou "en arrêt" des centrales.

L'horizon de temps considéré comporte N_t pas de temps et un parc nucléaire de N_ν réacteurs. La vie d'un réacteur $i \in \mathcal{N}_\nu$, avec $\mathcal{N}_\nu = \{1, ..., N_\nu\}$ se décompose en cycles indicés par $j = 1, ..., J_i$ sur l'horizon de temps, chaque cycle étant constitué d'une campagne de production suivi d'un arrêt de durée $\delta_{i,j}$. L'arrêt du cycle j peut débuter à toute date de l'ensemble $\mathcal{E}_{(i,j)} \subset \{1, ..., N_t\}$ et pour chaque date possible, on définit une variable binaire $x_{i,j,t}$, valant 1 si et seulement si l'arrêt débute effectivement à cette date. On doit alors satisfaire une contrainte dite *d'affectation* imposant à chaque cycle (i, j) une et une seule date d'arrêt : $\sum_{t \in \mathcal{E}_{(i,j)}} x_{i,j,t} = 1$. On déduit de ces variables deux grandeurs essentielles pour notre modèle. Tout d'abord la date effective du début de l'arrêt du cycle (i, j) qui vaut $t_{i,j} = \sum_{t \in \mathcal{E}_{(i,j)}} t x_{i,j,t}$. Puis la *disponibilité nucléaire*, c'est-à-dire la puissance totale des tranches n'étant pas arrêtée au pas de temps t, qui s'exprime également comme une fonction affine de x.

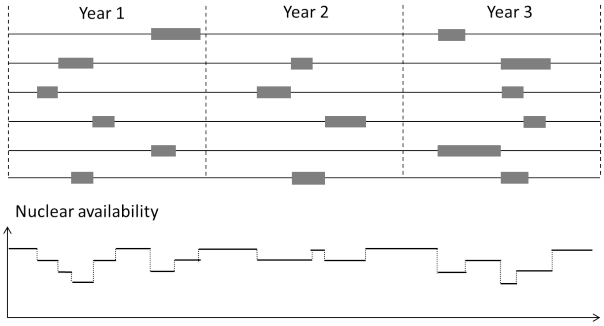

Figure 1: Un exemple de planning et de disponibilité nucléaire

La contrainte d'équilibre offre-demande n'est pas prise en compte explicitement dans le modèle. En fait, on suppose qu'il existe toujours suffisamment de moyens de productions alternatifs et d'offre de vente sur le marché pour satisfaire la demande, mais que leur utilisation se traduit par un coût croissant de satisfaction de l'équilibre offre-demande. Ce coût est à l'objectif que l'on cherche à minimiser. Il prend la forme d'une fonction convexe, linéaire par morceaux, de la disponibilité nucléaire. Dans un objectif de concision du modèle, cette fonction pourra être approximée par une fonction convexe quadratique.

Les nombreuses contraintes du problème sont liées principalement aux exigences de sûreté et à la disponibilité des ressources utilisées pendant les arrêts. Ces contraintes peuvent nécessiter l'introduction de variables continues, modélisant par exemple la production de la tranche pendant le cycle. Elles sont généralement linéaires, sauf la contrainte dite de *recouvrement maximal*, qui impose à certaines paires d'arrêts (i, j) et (i', j') de ne pas dépasser une certaine valeur N de recouvrement. Cette contrainte peut être vue comme une disjonction : $t_{i,j} - t_{i',j'} < N - \delta_{i,j}$ ou $t_{i,j} - t_{i',j'} > -N + \delta_{i',j'}$. Elle admet 3 modélisations, dont une l'une est quadratique :

- l'exclusion 2 à 2, qui interdit toutes les paires d'instanciation menant à une violation de cette contrainte : $x_{i,j,t} + x_{i',j',t'} \leq 1$ pour tout t, t' tels que $t - t' \in]N - \delta_{i,j},\ -N + \delta_{i',j'}[$;
- la formulation "bigM", qui repose sur l'introduction d'une variable binaire $z \in \{0, 1\}$, valant 0 ou 1 selon la partie de la disjonction qui est satisfaite : $t_{i,j} - t_{i',j'} \leq N - \delta_{i',j'} + M_1 z$ et $t_{i,j} - t_{i',j'} \geq -N + \delta_{i,j} - M_2(1 - z)$
- la formulation quadratique $(t_{i,j} - t_{i',j'} - N + \delta_{i,j})(t_{i,j} - t_{i',j'} + N - \delta_{i',j'}) \geq 0$.

Enfin, nous verrons que les différentes variantes de ce problème étudiées dans cette thèse varient également dans la façon dont sont pris en compte les aléas qui affectent la demande et la disponibilité des centrales nucléaires.

Ce problème est donc une application de choix pour notre étude, puisqu'il présente un caractère combinatoire et est soumis à des incertitudes. Nous commencerons par prendre en compte l'aspect combinatoire dans l'axe 1, avant de traiter l'incertitude dans l'axe 2.

Axe 1 : la SDP pour la relaxation de problèmes combinatoires et quadratiques
Le premier axe de cette thèse vise à utiliser la SDP pour produire des relaxations de problèmes combinatoires et quadratiques, comme suggéré par les nombreux résultats prometteurs dans ce domaine. L'obtention de ces relaxations commence généralement par l'implémentation de la relaxation SDP standard, qui doit ensuite être renforcée au moyen de coupes. Celles-ci peuvent être déterminées par l'étude de la structure du problème ou à l'aide de méthodes plus systématiques. Cette approche sera expérimentée sur différentes versions du problème de planification des arrêts nucléaires décrit ci-dessus, choisi pour sa nature combinatoire et ses possibles composantes quadratiques.

Nous commençons par utiliser une modélisation relativement complète, proche de celle utilisée en exploitation, donnant lieu à un problème quadratique à variables mixtes (MIQP pour Mixed Integer Quadratic Program) de grande taille auquel nous allons appliquer une relaxation SDP. Plus précisément, le modèle étudié correspond à une version déterministe du problème, pour laquelle l'objectif est une fonction quadratique de la disponibilité nucléaire et la formulation de la contrainte de recouvrement maximale est linéaire, correspondant à la formulation "bigM". Parmi les contraintes, la contrainte d'affectation est utilisée pour générer des contraintes quadratiques valides permettant de renforcer la relaxation SDP. Suivant le principe de Sherali-Adams [240], on la multiplie par l'une des variables binaires impliquées, ce qui donne lieu à la contrainte quadratique suivante : $\sum_{t' \in \mathcal{E}_{i,j},\ t' \neq t} x_{i,j,t} x_{i,j,t'} = 0$, que l'on ajoute à la relaxation SDP.

On fait suivre la relaxation SDP d'une procédure d'arrondi randomisé permettant d'obtenir une solution entière. Le principe de cet arrondi est d'interpréter la valeur d'une variable dans la solution relaxée comme la probabilité que la variable vaille 1. On tire alors des solutions entières suivant la loi de probabilité ainsi obtenue, jusqu'à obtenir une solution qui satisfasse toutes les contraintes.

Supposons que la relaxation produise une valeur optimale p_r et que la valeur optimale du problème initial soit p^*. Alors la qualité de la relaxation se mesure à son gap, égal à $(p^* - p_r)/p_r$, qui doit être le plus faible possible.

Sur la figure 2, on reporte le gap et la valeur de la solution arrondie obtenue, pour la relaxation SDP standard (*SDP*), la relaxation SDP renforcée (*SDP-R*) et la relaxation obtenue en relaxant la contrainte d'intégrité des variables binaires (*QP*), pour des jeux de données avec $N_t = 156$ et $N_\nu \leq 20$, menant à des problèmes d'environ 200 à 1300 variables binaires. On observe donc que les relaxations SDP donnent des gaps plus faibles, donc meilleurs, que la relaxation QP.

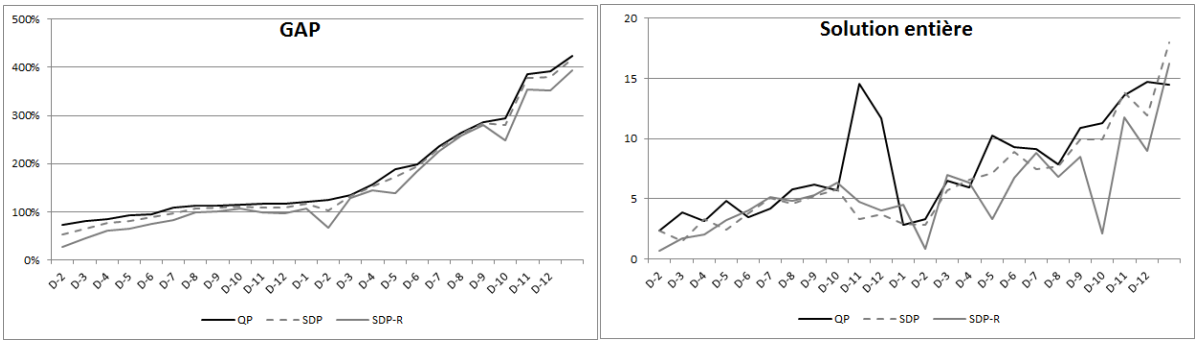

Figure 2: Résultats de la relaxation SDP et de l'arrondi randomisé sur le problème des arrets nucléaire

En conclusion, le gap moyen est réduit de 1.80% à 1.71% pour la relaxation SDP standard, et jusqu'à 1.56% pour la relaxation SDP renforcée, tout en conservant des temps de calculs raisonnables ($\leq$ 1030s pour SDP et $\leq$ 2231s pour SDP-R). Ces améliorations peuvent sembler faibles, mais elles sont à comparer aux gaps, eux-mêmes faibles, du fait d'une importante part constante dans la fonction objectif. Le gain sur l'arrondi randomisé est également significatif puisqu'il permet de ramener la perte d'optimalité de 7.75% à 6.41% et 5.59% pour les relaxations QP, SDP et SDP-R respectivement.

Suite à cette première expérimentation, nous proposons une méthode systématique permettant de déterminer une relaxation SDP renforcée d'un QCQP à variables bornées. Puis nous appliquons cette méthode à une version allégée du problème de planification des arrêts nucléaires, donnant lieu à un 0/1-QCQP, incluant un objectif quadratique, des contraintes quadratiques correspondant à la formulation quadratique de la contrainte de recouvrement maximal et des contraintes linéaires d'égalité et d'inégalité.

La méthode utilisée pour obtenir la relaxation SDP repose sur le constat que toutes les contraintes quadratiques valides proposées dans la littérature pour renforcer la relaxation SDP peuvent être vues

comme des combinaisons linéaires de contraintes quadratiques et de produits deux à deux des contraintes linéaires du problème, y compris les contraintes de borne. Alors, parmi toutes ces contraintes, il reste à sélectionner la contrainte la plus violée par la relaxation SDP courante, dans l'esprit d'un algorithme de séparation classique. Les différentes étapes de la méthode sont représentées sur la Figure 3.

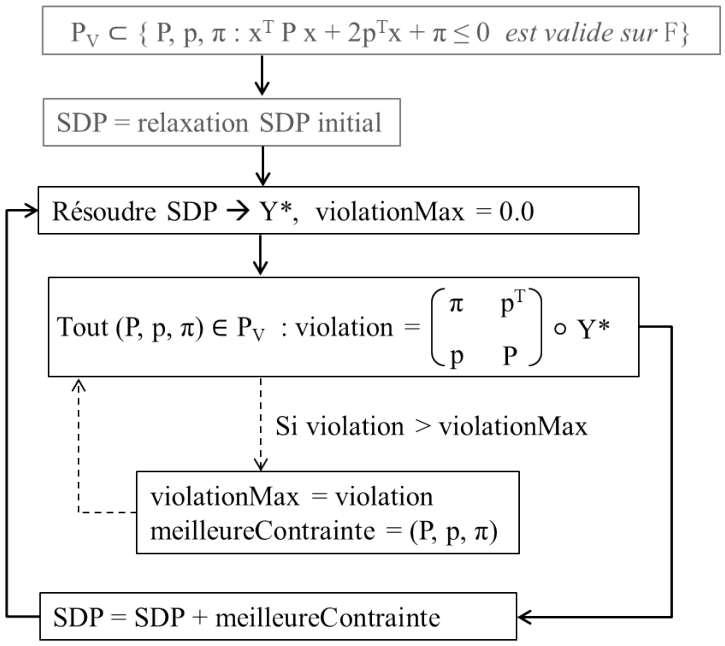

Figure 3: Algorithme de renforcement automatique d'une relaxation SDP

Nous testons également une version alternative de cette phase de séparation, visant à imposer à la contrainte obtenue d'être convexe. Dans ce cas, au lieu de sélectionner une contrainte valide parmi tous les produits deux à deux de contraintes linéaires, on considère toutes les combinaisons positives de ces contraintes et des contraintes initiales du problème. Le problème de séparation étant alors un SDP, cette méthode se révèle trop coûteuse en temps de calcul par rapport au gain obtenu sur les relaxations.

Au-delà du principe même de la méthode, notre contribution sur ce point a consisté à établir un certain nombre de preuves visant à écarter d'office des contraintes valides qui n'apportent rien pour le renforcement de la relaxation SDP. L'exemple le plus typique est la contrainte $(a^T x + b)^2 \geq 0$, valide pour n'importe quelles valeurs de a et b, mais inutile pour la relaxation SDP. Nous avons également montré que notre approche unifie un grand nombre de contraintes valides proposées dans la littérature pour le renforcement de la relaxation SDP [14, 127, 177, 213, 230].

Cette approche est également mise en oeuvre numériquement sur des jeux de données du problème des arrêts nucléaires possédant entre 200 et 1000 variables binaires et de 100 à 500 contraintes, en limitant à 100 le nombre de contraintes quadratiques valides ajoutées. Il en ressort que le gap moyen obtenu est de 6.88% pour la relaxation SDP renforcée, contre 6.97% pour la relaxation SDP standard et 25.76% pour la relaxation linéaire. L'avantage de la méthode est qu'elle permet également d'identifier les contraintes quadratiques les plus intéressantes pour le renforcement de la relaxation SDP. Ainsi on observe que les contraintes choisies en priorité sont celles issues du produit de deux contraintes linéaires qui partagent un grand nombre de variables. Plus précisément, pour une contrainte construite comme le produit des deux contraintes linéaires C_1 et C_2, on définit son ratio de recouvrement comme le rapport entre le nombre de variables impliqué dans C_1 et C_2 et le nombre de variables impliqué dans C_1 ou C_2. La décroissance des ratios des contraintes sélectionnées au cours des itérations est représentée sur la Figure 4.

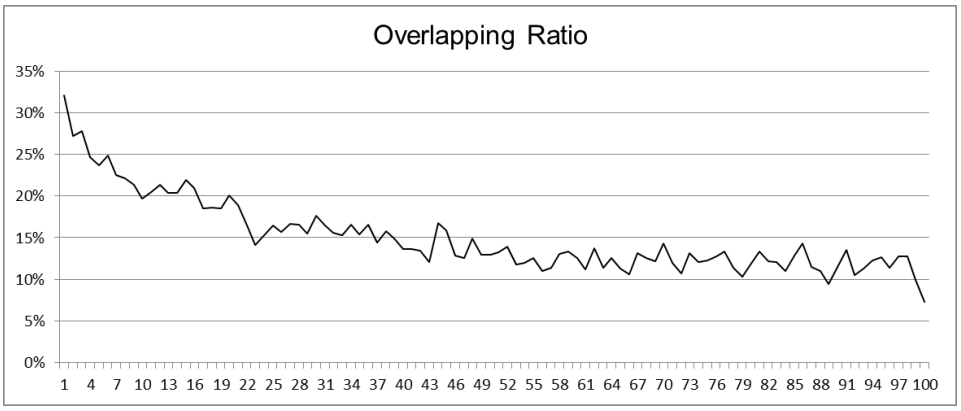

Figure 4: Ratio de recouvrement des contraintes sélectionnées

Pour poursuivre sur cet axe d'étude, nous nous attachons à comparer les différentes modélisations de la contrainte de recouvrement maximal du point de vue de la performance des relaxations obtenues. Il nous a également semblé important d'expérimenter la hiérarchie de Lasserre, une théorie très puissante considérant un problème d'optimisation polynomial quelconque et construisant une suite de SDP dont la valeur optimale tend vers la solution du problème initial. Pour ce faire, nous travaillons sur une version simplifiée du problème de planification des arrêts, dans laquelle ne sont considérées que les contraintes de recouvrement maximal et d'affection, et dans laquelle l'objectif est une fonction linéaire de la disponibilité nucléaire. 12 relaxations SDP sont testées, différant par :

- la mise au carré des contraintes linéaires;
- l'ajout des contraintes dite de type *RLT* (Reformulation Linearization Technique) ($x_i x_j \geq 0$, $x_i x_j \leq x_i$,$x_i x_j \leq x_j$ et $x_i x_j \geq 1 - x_i - x_j$);
- l'ajout de contraintes de type Sherali-Adams;
- l'ajout de contraintes triangulaires.

Pour chacune de ces relaxations SDP, nous construisons la relaxation linéaire équivalente, en linéarisant le QCQP intermédiaire ayant mené à l'obtention de la relaxation SDP. Puis une expérimentation est réalisée sur un ensemble de jeux de données regroupés en classe "i-j", dans lequel i correspond à la taille de l'instance et j à la modélisation utilisées pour la contrainte de recouvrement maximal ($j = 1$ pour la formulation "bigM", $j = 2$ pour l'exclusion 2 à 2 et $j = 3$ pour la formulation quadratique). Les résultats sont calculés en moyenne sur les 100 instances de chaque taille. Parmi toutes les relaxations SDP testées, les trois ci-dessous se démarquent, dont les gaps sont reportés sur la figure 5.5 :

- SDP-4, obtenues en élevant au carré toutes les contraintes linéaires ;
- SDP-7, obtenues en ajoutant à SDP-4 le produit de toutes les contraintes linéaires par toutes les contraintes de borne, dans l'esprit de Sherali-Adams;
- SDP-10, obtenues en ajoutant à SDP-7 les contraintes RLT.

Il ressort de cette expérimentation les constats suivants : pour les 3 modélisations, la meilleure relaxation SDP est obtenue tout d'abord en élevant les contraintes linéaires au carré, puis en ajoutant les produits de toutes les contraintes linéaires par toutes les contraintes de borne, puis les contraintes RLT. Parmi les 3 modélisations, l'exclusion 2 à 2 offre le meilleur potentiel pour la relaxation SDP par rapport à la relaxation LP, ce qui s'explique par le fait que le nombre de contraintes linéaires est très grand et que le renforcement est d'autant plus efficace que le nombre de contraintes linéaires est élevé.

Une dernière expérimentation consiste à mettre en oeuvre la hiérarchie de Lasserre sur ce problème. Le lecteur est renvoyé à l'article [171] pour davantage de détails sur cette séquence de problème

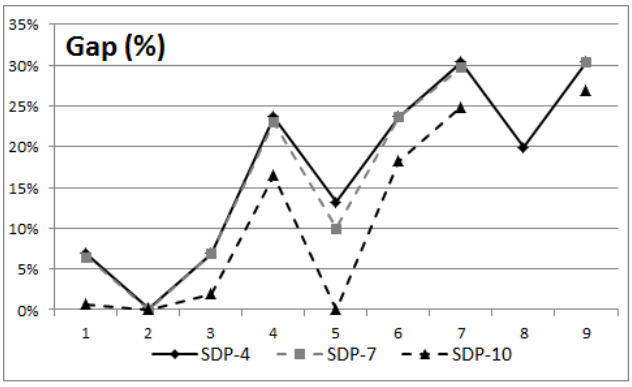

Figure 5: Comparaison des gaps des relaxations SDP-4, SDP-7 et SDP-10

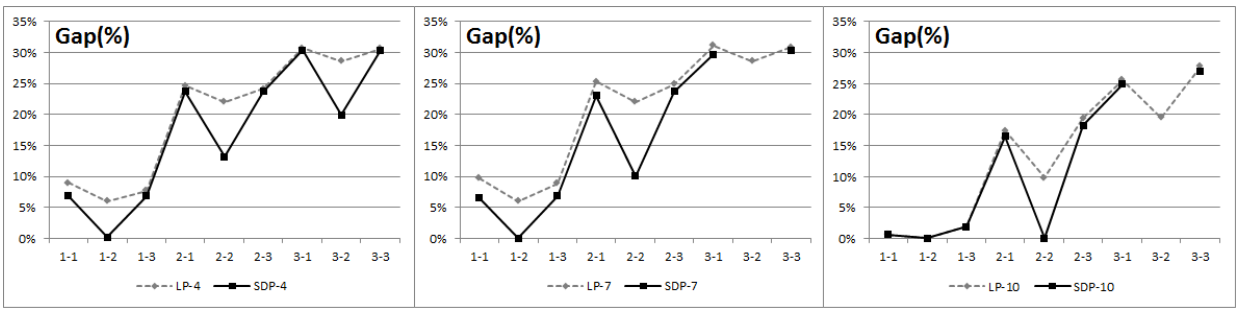

Figure 6: Comparaison des gaps SDP et LP pour les relaxations 4, 7 et 10

SDP menant à la solution optimale. Le rang 1 de cette hiérarchie correspond à la relaxation SDP standard et on s'attelle donc à résoudre le rang 2 pour les instances de taille 1. On observe alors que sur toutes les instances, le gap vaut 0.0%, ou autrement dit, le SDP fournit la valeur optimale du problème entier, et ceci avec des temps de calculs inférieurs à 10 s. Cette hiérarchie tient donc ses promesses en ce qui concerne la qualité de la borne mais il demeure une difficulté majeure quant à son applicabilité à des problèmes de plus grande taille, puisqu'en l'état des solveurs, il n'est pas possible de dépasser un nombre de variables de l'ordre de 30.

Axe 2 : la SDP pour la prise en compte de l'incertitude

Le second axe de la thèse, portant sur l'application de la SDP à la prise en compte de l'incertitude, se décline en trois études. Dans la première étude, nous travaillons sur une version du problème des arrêts nucléaires dans laquelle l'incertitude se présente sous la forme de scénarios équiprobables et les contraintes concernées par les incertitudes sont à satisfaire en probabilité. Il est alors classique de formuler un équivalent déterministe de ce problème comme un problème combinatoire de grande taille en ajoutant une variable binaire par contrainte et par scénario. Nous appliquons alors la relaxation SDP standard à ce problème et la faisons suivre d'un arrondi randomisé.

Le problème considéré correspond au modèle décrit ci-dessus, dans lequel l'objectif et les contraintes de recouvrement maximal sont quadratiques. Les expérimentations sont menées avec 10 scénarios, sur des jeux de données comportant de 700 à 2400 variables binaires. Les résultats sont illustrés sur la courbe de la figure 7, dans laquelle la solution entière représente l'écart entre la solution de l'arrondi randomisé et la meilleure solution entière trouvée par CPLEX en 1800 s. Ces résultats sont très prometteurs, avec un gap moyen de 2.76% avec la relaxation SDP contre 53.35% pour la relaxation linéaire, et ceci sans le moindre renforcement.

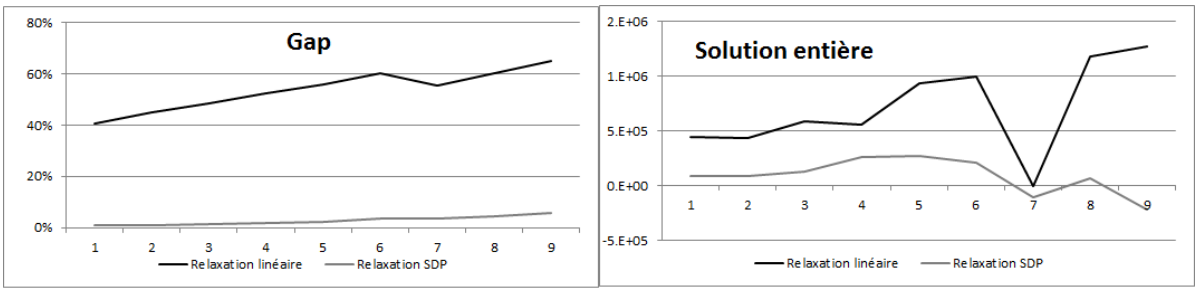

Figure 7: Résultats de la relaxation SDP et de l'arrondi randomisé sur le problème des arrêts des centrales nucléaire stochastique

La deuxième étude met en oeuvre une méthode plus ambitieuse pour la prise en compte de l'incertitude, permettant de reformuler le problème, ou d'en donner une approximation conservative, sous la forme d'un SDP. Cette méthode, qui a fait l'objet de nombreux travaux récemment, est connue sous le nom d'optimisation distributionnellement robuste. Il s'agit en fait d'un compromis entre l'optimisation stochastique, qui nécessite une connaissance parfaite des lois de probabilités utilisées, et l'optimisation robuste, qui ne requiert que la connaissance du support des variables aléatoires. En effet, l'optimisation distributionnellement robuste ne nécessite pas de connaître la distribution de probabilité, mais uniquement son support et certains de ses moments. Nous appliquons cette méthode à un problème d'équilibre offre-demande dans lequel la demande et la disponibilité des moyens de production sont soumis à des aléas, dont on connaît le support, l'espérance et la covariance. Notre objectif est d'estimer l'apport de cette méthode par rapport à une méthode de type robuste basée sur la connaissance du support et de l'espérance, permettant d'obtenir une approximation conservative du problème sous forme d'un SOCP.

Le problème d'équilibre offre-demande considéré peut se mettre sous la forme d'un problème à contrainte en probabilité jointe linéaire :

$$\min_{x\in F} \{c^T x : \mathrm{P}[g(x,\xi) \leq 0] \geq 1-\varepsilon\}$$

avec :

- c un vecteur de $\mathbb{R}$. et $F \subset \mathbb{R}^n$ un polyèdre;
- $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^T$ une fonction affine de x et de ξ telle que $g(x,\xi)_t = \tilde{x}^T A_t \tilde{x}$;
- ξ un vecteur aléatoire de $\mathbb{R}^m$ dont on connaît le support $\mathcal{S} = \{\xi \in \mathbb{R}^m : a_i \leq \xi_i \leq b_i\}$, l'espérance μ et la covariance Σ ;
- ε le niveau de probabilité requis quant à la satisfaction de la contrainte.

On définit $\mathcal{P}(\mathcal{S})$ comme l'ensemble des distributions de probabilités de support $\mathcal{S}$, d'espérance μ et de covariance Σ. Plus précisément $\mathcal{P}(\mathcal{S}) = \{\mathrm{P} \in \mathcal{M}(\mathcal{S}) : \Omega_{\mathrm{P}}(\xi) = \Omega\}$ avec $\Omega = \begin{pmatrix} 1 & \mu^T \\ \mu & \mu\mu^T + \Sigma \end{pmatrix}$. On cherche alors à satisfaire la contrainte pour toute les distributions de $\mathcal{P}(\mathcal{S})$, c'est-à-dire :

$$(C_1) \min_{x\in F} \{c^T x : \mathrm{P}[g(x,\xi) \leq 0] \geq 1-\varepsilon, \forall \mathrm{P} \in \mathcal{P}\} \quad \text{ou} \quad (C_2) \min_{x\in F} \{c^T x : \inf_{\mathrm{P}\in\mathcal{P}(\mathcal{S})} \mathrm{P}[g(x,\xi) \leq 0] \geq 1-\varepsilon\}$$

La formulation (C_2) fait intervenir le problème de moments $\inf_{\mathrm{P}\in\mathcal{P}(\mathcal{S})} \mathrm{P}[g(x,\xi) \leq 0]$, auquel s'applique la hiérarchie de Lasserre. Notre contribution se décline alors en trois points. Tout d'abord, nous montrons que le SDP obtenu en appliquant la hiérarchie de Lasserre peut également être obtenu très simplement en utilisant le S-Lemma et qu'il unifie un grand nombre d'approximations SDP proposées dans la littérature, en particulier celle de Zymler et al. [270] mais aussi [40, 41, 68, 257], ainsi que

d'inégalités célèbres sur les probabilités, à savoir les inégalités de Markov et de Cantelli. Puis nous donnons des conditions suffisantes pour que cette approximation soit exacte. Enfin nous comparons les bornes obtenues à des méthodes plus classiques, sur des problèmes académiques et d'équilibre offre-demande.

Nous commençons par montrer grâce au S-Lemma qu'une condition suffisante pour que la contrainte en probabilité $\mathrm{P}[g(x,\xi) \leq 0] \geq 1-\varepsilon$ soit respectée pour tout $\mathrm{P} \in \mathcal{P}(\mathcal{S})$ est qu'il existe une matrice M qui satisfasse le système SDP suivant :

$$\begin{cases} \Omega \bullet M \leq \varepsilon \\ M - \sum_{i=1}^{m} \lambda_{0,s} W^i \succcurlyeq 0 \\ M - M_0 - \sum_{i=1}^{m} \lambda_{t,s} W^i + \tau_t Y^t(x) \succcurlyeq 0, \ t = 1, ..., T \\ \lambda \geq 0, \ \tau \geq 0 \end{cases} \quad \text{où} \quad \begin{cases} W^i = \frac{1}{2} \begin{pmatrix} -2a_i b_i & a_i + b_i \\ a_i + b_i & -2 \end{pmatrix} \\ Y^t(x) = \frac{1}{2} \begin{pmatrix} 2\tilde{x}^T A_{*,1} & \tilde{x}^T A_{*,2...m+1} \\ A^T_{*,2...m+1} \tilde{x} & 0 \end{pmatrix} \\ M_0 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \end{cases}$$

Nous montrons que cette approximation est exacte dès lors que $T = 1$, c'est-à-dire lorsque la contrainte en probabilité est individuelle. Puis nous procédons à des comparaisons numériques pour le cas particulier où aucune variable de commande x n'est présente ($n = 0$). Il s'agit alors de déterminer une borne inférieure d'une probabilité, avec ou non prise en compte de la covariance, sachant qu'une borne inférieure est meilleure lorsqu'elle est élevée puisque l'approximation associée sera d'autant moins conservative.

Nous nous intéressons en premier lieu au cas $m = 1$ et $g(x,\xi) = \xi$. Alors, sans la prise en compte de la covariance, l'optimisation distributionnellement robuste mène à la borne de Markov, à savoir μ/a, si $[a,b]$ est le support de ξ. Cette borne est comparée aux valeurs obtenues par optimisation distributionnellement robuste avec covariance et à la borne dite "robuste", basée sur la connaissance du support et de l'espérance, permettant de formuler une approximation conservative du problème sous la forme d'un SOCP via l'application de l'inégalité de Hoeffding [68, 269]. Les résultats sont transcrit sur la courbe de la figure 6.1. Pour chaque jeu de données sont indiquées la borne de Markov, la borne robuste valant $1 - \exp(-2\mu^2)(\|b-a\|^2)$, ainsi que la plus petite (*Min var*) et la plus grande (*Max var*) borne obtenue par optimisation distributionnellement robuste avec covariance.

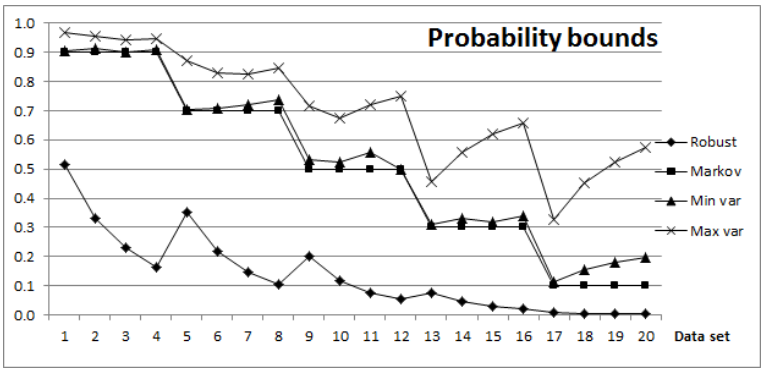

Figure 8: Comparaison de différentes bornes inférieurs de $\min_{\mathrm{P} \in \mathcal{P}} \mathrm{P}[\xi \leq 0]$

Nous considérons ensuite le cas où $m \geq 1$ et $g(x,\xi) = e^T \xi$. Les jeux de données sont construits aléatoirement, en tirant les valeurs de a_i, b_i, μ_i et de la matrice de covariance. La figure 6.3 illustre la comparaison entre la borne robuste $(1 - \exp(-2(e^T\mu)^2 / \|b-a\|^2))$, la borne de Markov $(e^T\mu / e^T a)$ obtenue en considérant $e^T\xi$ comme une variable aléatoire de moyenne $e^T\mu$ et de support $[e^T a, e^T b]$, et les résultats de l'optimisation distributionnellement robuste avec et sans covariance.

Enfin, il reste à étudier le cas où $T \geq 1$, avec $g_t(x,\xi) = d_t^T \tilde{\xi}$. Pour ce problème, on compare la borne robuste, qui s'avère être négative donc inutile, et les bornes distributionnellement robustes. Les résultats sont représentés sur la courbe 6.4.

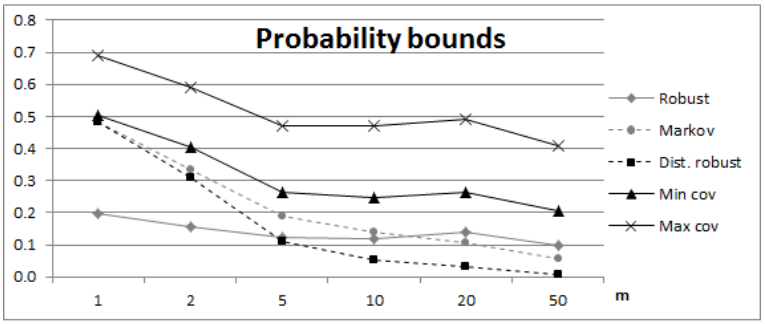

Figure 9: Comparaison de différentes bornes inférieures de $\min_{\mathrm{P}\in\mathcal{P}} \mathrm{P}[e^T\xi \leq 0]$ pour $m \geq 1$

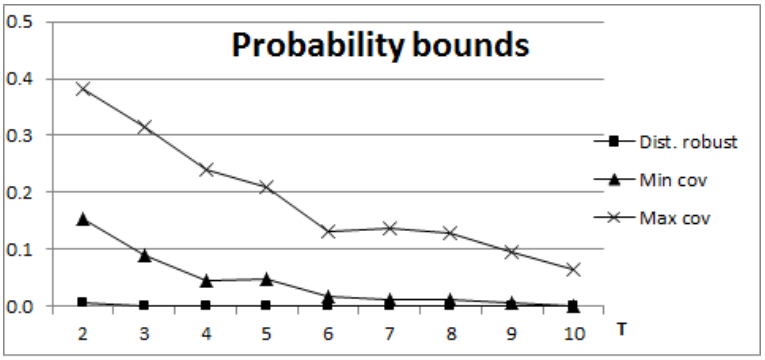

Figure 10: Comparaison de différentes bornes inférieures de $\min_{\mathrm{P}\in\mathcal{P}} \mathrm{P}[d_t^T\tilde{\xi} \leq 0,\ t = 1, ..., T]$

Il ressort de ces comparaisons que les bornes distributionnellement robuste avec covariance sont clairement les plus performantes. Il reste à les appliquer à un véritable problème d'optimisation avec des variables de commandes. Pour cela, on considère le problème d'équilibre offre-demande suivant, dans lequel on minimise un coût de production linéaire, tout en satisfaisant en probabilité l'équilibre offre-demande à chaque pas de temps :

$$\begin{cases} \min & c^T x \\ \text{s.t.} & \mathrm{P}\left[\sum_{i=1}^{N} D_{i,t} x_{t,i} \geq D_{0,t},\ t = 1, ..., T\right] \geq 1 - \varepsilon \\ & \sum_{t=1}^{T} x_{t,i} \leq r_i,\ i = 1, ..., N \\ & x_{t,i} \in [0, 1],\ i = 1, ..., N,\ t = 1, ..., T \end{cases}$$

où :

- T est le nombre de pas de temps considérés;
- N est le nombre d'unité de production;
- $c_{t,i}$ est le coût unitaire de production de l'unité de production i au pas de temps t;
- $x_{t,i}$ représente la production de l'unité de production i au pas de temps t;
- $D_{0,t}$ est une variable aléatoire représentant la demande au pas de temps t ;
- $D_{i,t}$ est une variable aléatoire représentant le coefficient de disponibilité de l'unité de production i au pas de temps t;
- r_i représente la production maximale de l'unité de production i sur l'horizon de temps.

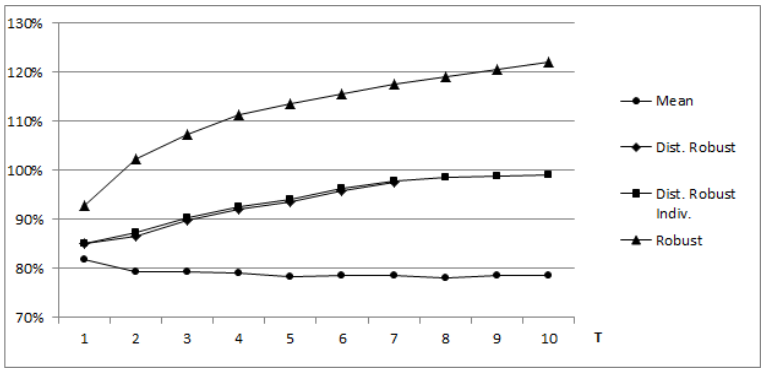

Figure 11: Comparaison du ratio p_w^*/p^* pour $\varepsilon = 0.8$

T	$\varepsilon = 0.5$			$\varepsilon = 0.4$			$\varepsilon = 0.3$			$\varepsilon = 0.2$		
	p_s^*	p_{dri}^*	loss	p_s^*	p_{dri}^*	loss	p_s^*	p_{dri}^*	loss	p_s^*	p_{dri}^*	loss
1	262.9	261.3	-0.59%	267.0	265.6	-0.53%	271.8	271.2	-0.23%	277.5	279.4	0.69%
2	512.9	538.6	5.02%	521.2	548.6	5.26%	530.8	562.5	5.98%	542.5	582.1	7.31%
3	782.3	848.6	8.47%	794.5	866.5	9.06%	808.9	886.8	9.63%	826.5	895.2	8.32%
4	1040.1	1162.7	11.79%	1056.6	1182.8	11.95%	1076.0	1191.1	10.70%	1099.9	1192.7	8.44%
5	1290.6	1473.8	14.19%	1311.4	1483.9	13.15%	1336.0	1484.3	11.10%	1366.3	1489.1	8.99%
6	1538.3	1762.4	14.57%	1563.3	-	-	1592.8	-	-	1629.4	-	-
7	1780.2	2040.6	14.63%	1810.5	-	-	1846.0	-	-	1889.4	-	-
8	2021.3	2325.5	15.05%	2057.1	-	-	2098.9	-	-	2149.9	-	-
9	2263.9	-	-	2305.7	-	-	2354.1	-	-	2412.9	-	-
10	2513.0	-	-	2560.1	-	-	2614.8	-	-	2681.1	-	-

Table 1: Comparaison avec l'approche stochastique

Les données sont calculées sur la base des observations historiques. Pour chaque jeu de données sont comparées :

- p_m^*, la valeur optimale du LP obtenu en remplaçant l'aléa par sa valeur moyenne;
- p_{dr}^*, la valeur optimale obtenue par optimisation distributionnellement robuste;
- p_{dr}^*, la valeur optimale obtenue par optimisation distributionnellement robuste, avec approximation de la probabilité jointe par une somme de probabilités individuelles, via l'inégalité de Boole;
- p_r^*, la valeur optimale obtenue par approche robuste;
- p_w^*, la valeur optimale du LP obtenu en remplaçant l'aléa par sa valeur dans le pire cas.

Pour $\varepsilon = 0.8$, les résultats obtenus sont reportés sur la figure 6.5.

Nous constatons que la perte liée à l'approximation de la probabilité jointe par une somme de probabilité individuelle est faible. De plus, la borne distributionnellement robuste est nettement meilleure que la borne robuste, ce qui peut s'expliquer par le fait qu'elle exploite une information supplémentaire, à savoir la covariance.

Nous procédons finalement à une comparaison avec une approche stochastique, basée sur l'hypothèse que $g(x, \xi)$ suit une loi normale. La probabilité jointe est approximée par une probabilité individuelle, comme déjà évoqué. Nous observons que la perte d'optimalité liée à l'approche distributionnellement robuste n'est pas aussi grande qu'on aurait pu le craindre. En particulier, pour $T = 1$, cette approche est même meilleure que l'approche stochastique, du fait sans doute qu'on y exploite une information supplémentaire : le support.

Cet axe d'étude se termine par une troisième étude combinant les axes combinatoires et stochastiques explorés préalablement. Il s'avère que de nombreux problèmes aléatoires admettent un équivalent

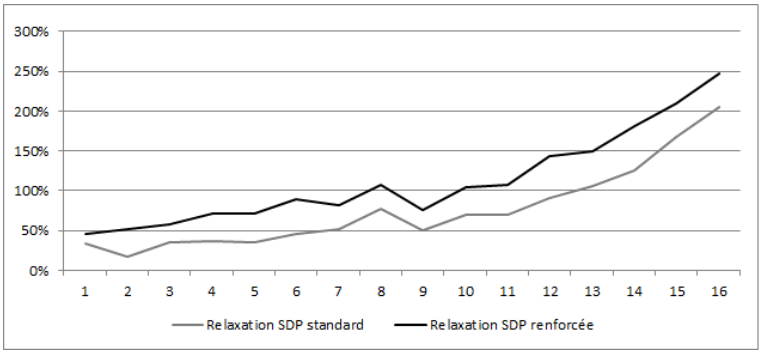

Figure 12: Amélioration de la relaxation SDP par rapport à la relaxation continue

déterministe, ou une approximation, sous forme d'un SOCP et dans le cas où le problème initial comporte des variables entières, le problème obtenu est alors un MISOCP, pour lesquels il n'existe pas de méthodes de résolution de référence. Nous nous intéressons ici à l'utilisation de relaxations SDP pour ce type de problème. Le principe est de convertir le MISOCP en MIQCQP, puis d'appliquer la relaxation SDP standard, qui est ensuite renforcée par l'ajout de contraintes du problème initial mises au format SDP.

Cet ajout de contraintes est rendu nécessaire par le fait que la formulation d'un SOCP comme un QCQP est généralement synonyme d'une perte de convexité. En effet, cette reformulation suit le principe suivant :

$$\|Ax+b\| \leq c^T x + d \quad \Leftrightarrow \begin{cases} x^T(A^TA - cc^T)x + 2(b^TAx - dc^Tx) + b^Tb - d^2 \leq 0 \\ c^Tx + d \geq 0 \end{cases}$$

Or la matrice $A^TA - cc^T$ n'est généralement pas semi-définie positive. On montre le résultat suivant :

Proposition *Soit $A \in \mathbb{R}^{m,n}$ une matrice de rang plein et $c \in \mathbb{R}^{n,1}$. Alors la matrice symétrique $A^TA - cc^T$ est semi-définie positive si et seulement si il existe $u \in \mathbb{R}^{m,1}$, avec $\|u\| \leq 1$, tel que $c = A^Tu$.*

Cette conversion d'un SOCP en QCQP illustre parfaitement la différence entre convexité et convexité abstraite. En effet, l'ensemble réalisable du QCQP est nécessairement convexe, puisqu'il est équivalent à l'ensemble réalisable du SOCP donc le QCQP est convexe au sens abstrait. Cependant, cet ensemble n'est pas décrit à l'aide de contraintes convexes, donc le problème n'est pas convexe.

Afin d'exploiter la structure particulière du QCQP ainsi obtenue, et de restaurer sa convexité, nous renforçons la relaxation SDP standard au moyen des contraintes SDP obtenues en écrivant les contraintes SOCP directement sous forme SDP, suivant l'équivalence bien connue :

$$\|Ax+b\| \leq c^Tx + d \Leftrightarrow \begin{pmatrix} (c^Tx+d)I & Ax+b \\ (Ax+b)^T & c^Tx+d \end{pmatrix} \succcurlyeq 0$$

On obtient ainsi la borne une relaxation SDP standard et une relaxation SDP renforcée. Pour chacune d'entre elle, on calcule l'amélioration par rapport à la relaxation continue, comme suit : $r = \frac{p_s - p_c}{p_c}$, où p_s est la borne SDP et p_c la borne continue. On trace ces deux indicateurs sur la figure 12.

Cette approche donne des résultats encourageants, avec une relaxation SDP nettement plus performante que la relaxation continue. Les solutions obtenues sont même très proches de l'optimal sur de nombreuses instances tout en conservant un temps de calcul raisonnable, de l'ordre de quelques minutes pour des instances à quelques centaines de variables binaires.

Conclusions et perspectives

Cette thèse avait pour objet d'évaluer les potentialités de la programmation semi-définie positive (SDP) pour les problèmes d'optimisation issus du management d'énergie. Nous avons montré qu'il existe de nombreuses opportunités d'innovation pour le traitement de ces problèmes par la SDP, en particulier pour les problèmes présentant un caractère combinatoire ou quadratique, ou pour la prise en compte de l'aléa.

Concernant l'axe quadratique et combinatoire, la SDP fournit des relaxations convexes de ces problèmes NP-difficiles. La difficulté est alors de déterminer le bon niveau de compromis entre taille du SDP obtenu et qualité de la relaxation. A un extrême se situe la relaxation SDP standard, simple à obtenir et présentant quasiment le même nombre de contraintes que le problème initial. Cette relaxation n'est généralement pas très performante. A l'autre extrême se trouve les relaxations SDP obtenues en appliquant la hiérarchie de Lasserre. Celles-ci sont extrêmement performantes mais sont malheureusement de trop grande taille pour être résolues pour des problèmes de plus de 30 variables binaires. Enfin, des relaxations intermédiaires peuvent être construites en ajoutant des contraintes quadratiques valides au problème initial, obtenues par exemple en multipliant des contraintes linéaires valides entre elles.

Du côté de la prise en compte de l'aléa, la SDP se prête plutôt aux approches robustes et distributionnellement robuste. En effet, certains résultats bien connus de la littérature montrent que des problèmes robustes peuvent s'écrire sous forme de problème conique du second-ordre, qui est un cas particulier de SDP. Nous avons étendu cette équivalence à des problèmes possédant des variables binaires, pour lesquels l'équivalent robuste est donc un problème conique du second-ordre à variables binaires et nous avons montré que la SDP produit de très bonnes relaxations de ces problèmes difficiles. Dans le cadre de l'optimisation distributionnellement robuste, de nombreux travaux proposent des approximations conservatrices de contraintes en probabilité sous forme d'inégalités linéaires matricielles. Afin d'offrir une vision claire sur ces approches, nous les avons unifié et présentons un certain nombre d'expérimentations numériques illustrant la pertinence de cette façon de prendre en compte l'aléa.

Ainsi, en dépit de nombreuses difficultés pratiques imputables au fait que cette technologie n'est pas encore tout à fait mature, la SDP n'en reste pas moins une méthode extrêmement prometteuse, combinant toute la puissance de l'optimisation conique. De nombreuses applications méritent encore d'être explorées, aussi aussi bien en management d'énergie que dans d'autres domaines tels que l'ingénierie ou la gestion de portefeuille d'actifs financiers.

Introduction

As is suggested by its title "Semidefinite Programming : Methods and Algorithms for Energy Management", the present thesis aims at identifying potential applications of Semidefinite Programming (SDP), a promising optimization technique, to the problems related to the management of electricity production. This encompasses various optimization problems that share the requirement of satisfying the equilibrium between the electricity demand and the production of several types of units, while respecting various technical constraints.

These global optimization problems, also known as Unit Commitment Problems, differ in the considered electricity board, horizon time and time steps and by the way of addressing uncertainty. A good conduct of all these issues is crucial for the economic performance of the company, its environmental impact and the social welfare of its customers, but presents several difficulties.

Firstly, from a modelling point of view, deciding the adequate level of simplification of the complex phenomenon that are involved is not an easy task, which is complicated by the necessity of being consistent with the precision of the input data. Furthermore, the French power mix includes hydraulic, nuclear, and classical thermal power plants, with each generation unit having its own technical constraints. Therefore, the adopted model shall vary in accordance with the considered production units and with the time horizon, from a few hours to a few decades.

Secondly, from an optimization point of view, these problems lead to severe challenges, mainly due to their large size, the presence of non-linearities and the uncertainty that affects the data. Even on small time horizon, they are intractable with a direct frontal approach and it is therefore necessary to proceed to many approximations and decompositions to solve them.

Among the variety of optimization techniques that are employed, two of them can be distinguished. Firstly, Mixed-Integer Linear Programming is generally used for tackling combinatorial problems. Secondly, some Conic Programming approaches are investigated, in particular for addressing uncertainty. This thesis follows this research line since Semidefinite Programming is currently the most sophisticated area of Conic Programming that is polynomially solvable.

More precisely, SDP is the optimization over the cone of positive semidefinite matrices of a linear objective function subject to linear equality constraints. It can also be viewed as a generalization of Linear Programming where the nonnegativity constraints on vector variables are replaced by positive semidefinite constraints on symmetric matrix variables.

The past few decades have witnessed an enormous interest for SDP due to the identification of many theoretical and practical applications, e.g., combinatorial optimization (graph theory), spectral optimization, polynomial optimization, engineering (systems and control), probability and statistics, financial mathematics, etc... In parallel, the development of efficient SDP solvers, based on interior-point algorithms, also contributed to the success of this method.

In the present thesis, we pursue two main objectives. The first one consists of exploring the potentiality of semidefinite programming to provide tight relaxations of combinatorial and quadratic problems. This line of research was motivated both by the promising results obtained in this direction [108, 186, 245] and by the combinatorial and quadratic features presented by energy management problems. The second one deals with the treatment of uncertainty, an issue that is also of paramount importance in energy management problems. Indeed, due to its versatility, SDP is well-known for providing numerous possibilities of dealing with uncertainty. In particular, it offers a way of modelling the deterministic counterpart of robust optimization problems, or more originally, of distributionnally robust optimization problems.

This thesis is organized as follows. The first part is composed of the first three chapters and provides an overview of the main results concerning SDP, starting by the first chapter that contains the theory of convex and conic optimization. This is followed by a focus on SDP : its underlying theory is presented in the second chapter and its most famous applications are discussed in the third chapter.

The second part is composed of the last three chapters and presents the application of SDP to energy management. Chapter 4 provides an introduction to energy management problems, with a special emphasis on one of the most challenging energy management problem, namely the Nuclear Outages Scheduling Problems. This problem was selected both for being a hard combinatorial problem and for requiring the consideration of uncertainty. We present at the end of this chapter the different models that we elaborated for this problem.

The next chapter reports the work related to the first objective of the thesis, i.e., the design of semidefinite programming relaxations of combinatorial and quadratic programs. On certain problems, these relaxations are provably tight, but generally it is desirable to reinforce them, by means of tailor-made tools or in a systematic fashion. We apply this paradigm to different models of the Nuclear Outages Scheduling Problem. Firstly, we consider a complete model that takes the form of a MIQP. We apply the semidefinite relaxation and reinforce it by addition of appropriate constraints. Then, we take a step further by developing a method that automatically generates such constraints, called cutting planes. For the design of this method, we start by providing a framework for unification of many seemingly disparate cutting planes that are proposed in the literature by noting that all these constraints are linear combinations of the initial quadratic constraints of the problem and of the pair-wise product of the linear constraints of the problem (including bounds constraints).

Subsequently, we focus on specific part of the problem, namely the maximal lapping constraint, which takes the form of $a^T x \notin [b, c]$, where x are binary variables. This constraint presents modelling difficulty due to its disjunctive nature. We aim at comparing three possible modelisations and for each of them, computing different relaxations based on semidefinite programming and linear programming. Finally, we conclude this chapter by an experiment of the Lasserre's hierarchy, a very powerful tool dedicated to polynomial optimization that builds a sequence of semidefinite relaxations whose optimal values tends to the optimal value of the considered problem.

In the last chapter, we cope with the second objective of this thesis, i.e., to examine how semidefinite programming can be used to tackle uncertainty. To this end, three different works are carried out. First, we investigate a version of the nuclear outages scheduling problem where the random variables have a discrete probability distribution that takes the form of equiprobable scenarios and the constraints involving uncertain parameters have to be satisfied up to a given level of probability. It is well-known that this model admits a deterministic formulation by the addition of binary variables. Then the obtained problem is a combinatorial problem and we apply semidefinite programming to compute tight bounds of the optimal value.

We have also implemented a more original way of dealing with uncertainty, which admits a deterministic counterpart, or a conservative approximation, under the form of a semidefinite program. This method, that has received much attention recently, is called *distributionnally robust optimization* and can be seen as a compromise between stochastic optimization, where the probability distribution is required, and robust optimization, where only the support is required. Indeed, in distributionnally robust optimization, the support and some moments of the probability distribution are required. In our case, we assume that the support, the expected value and the covariance are known and we compare the benefits of this method w.r.t other existing approaches. In particular, we compare to an approach that relies on the combined application of the Boole's and Hoeffding's inequalities to provide a conservative approximation of the problem in the form of a Second-Order Cone Program.

Finally, we carried out a last experiment that combines both uncertainty and combinatorial aspects. Indeed, many deterministic counterpart or conservative approximation of Linear Program (LP) subject to uncertainty give rise to a Second-Order Cone Program (SOCP). In the case of a Mixed-Integer Linear Program, we obtain a MISOCP, for which there is few reference methods in the literature. Then we investigate the strength of the relaxation SDP for such problems. Central to our approach is the reformulation as a non convex Quadratically Constrained Quadratic Program (QCQP), which brings us in the framework of binary quadratically constrained quadratic programs. This allows to derive a semidefinite relaxation of the problem. When necessary, this relaxation is tightened by valid quadratic constraints derived from the initial problem. We report encouraging computational results

indicating that the semidefinite relaxation improves significantly the continuous relaxation (112% on average) and often provides a lower bound very close to the optimal value. In addition, computational time for obtaining these results remains reasonable.

In brief, our contribution can be outlined as follows. First, we provide a comprehensive and unified framework of the different methods proposed in the literature to design SDP relaxations of QCQP. This framework relies on the definition of a standard SDP relaxation that can be obtained in a systematic fashion. It is generally necessary to reinforce this relaxation by adding valid constraints to the initial QCQP. In particular, we proved that for 0/1-LP, the standard SDP relaxations yields the same optimal value than the linear relaxation. In this case, it is essential to reinforce the standard SDP relaxation to justify the use of SDP. To the best of our knowledge, this equivalence had not been clearly highlighted in the literature. In order to apply the semidefinite relaxation to the NOSP, we designed several models that emphasize one or another aspects of this problem. Then we study and analyse different possibilities to reinforce the semidefinite relaxation : on the one hand, we applied some recipes proposed in the literature and on the other hand, we designed and applied an automatic method based on a separation algorithm. Regarding the treatment of uncertainty, we applied the distributionnally robust approach to the NOSP and derived SDP conservative approximation of the obtained problem. We emphasize the connection existing between this approach, the Generalized Problem of Moments and other famous works exploiting the knowledge of moments for optimizing under uncertainty. Finally, our last contribution concerns the area of MISOCP, for which we propose a simple method to derive a semidefinite relaxation, which turns out to be tight.

This thesis ends with three appendices. The first ones summarizes the notations that are used along the thesis. The latter two are provided in order to keep this document self-contained. They contain respectively the mathematical and optimization backgrounds that are required to address the concepts covered in this thesis.

Part I

Fundamentals of Semidefinite Programming

Table of Contents

Introduction

Semidefinite programming is a relatively young area of optimization, dating back to the late seventies. However, it has since received a great deal of attention in the optimization literature. This interest arose to a peak in the nineties, with the development of efficient algorithms to solve them [206] in 1994 and the milestone application of SDP to combinatorial optimization [108] in 1995.

The purpose of this chapter is to describe the theory of Semidefinite Programming by providing definitions and theoretical facts in order to yield insight into how defining, solving and applying such optimization problem in multiple contexts. A particular emphasis is placed on how using SDP to get relaxation of NP-hard problem, such as quadratic or combinatorial, and how this relaxation can lead to the design of efficient approximation algorithms. We also review some applications of SDP to stochastic and robust optimization. Finally, we describe the Generalized Problem of Moment, another application of SDP that has attracted major interest recently.

As a subfield of convex and conic optimization, Semidefinite Programming benefits from all the theoretical and practical results of these areas, that are summarized in Chapter 1. In Chapter 2, we formally define Semidefinite Programming and review the main related works. In Chapter 3, we discuss several applications and special cases of Semidefinite Programming with an emphasis on applications potentially valuable for energy management.

Throughout this document, we focus on pointing out the difficulties pertaining to a practical implementation of Semidefinite Programming and we discuss the current issues associated with this area. This state of the art review, associated with the appendices containing mathematical and optimization backgrounds, serves as a reference to keep this thesis self-contained. The material in this chapter is, for the most part, based on the handbook of Semidefinite Programming [259] and on the following surveys [57, 58, 208, 252].

Chapter 1

Convex and conic optimization

« But we all know that the world is not linear ! »

(H. Hotelling)

Convex optimization is an important class of optimization problems that subsumes linear and semidefinite programming. Such problems are of central importance in optimization since convexity is generally considered as the true discriminant between "easy" and "hard" problems in optimization. In the famous Rockafellar's terms [229] "*the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity.*"

Indeed, a fundamental result of convex analysis states that any locally optimal solution of a convex optimization problem is then guaranteed globally optimal (see Theorem 2.4.45). In practice, this means that a local optimality guarantee is sufficient for global optimality, and can serve for instance for an algorithmic termination test.

Furthermore, there is an elegant duality theory for these problems, that satisfies a weak duality property, even strong duality under a Slater-type condition. These properties are detailed in Paragraph 1.1.3.

Finally, convex optimization problem can be solved efficiently. From a theoretical point of view, it follows then from the Ellipsoid method and from the results of [118] than any linear function can be optimized over a convex set in polynomial time as long as we can design a *separation oracle* that runs in polynomial time. The existence of this procedure is guaranteed by the Separating Hyperplane Theorem (see Theorem 2.2.23) but it might be computationally expensive, especially when the feasible set is not specified explicitly.

However, it is well known that the Ellipsoid method is of limited practical value. Fortunately, the development of efficient, reliable numerical algorithms was made possible by the results of Nesterov and Nemirovski [206] about the applicability of interior-point methods to convex optimization problem. This extension relies on the definition of a *barrier function* for the feasible set, i.e., a function that tends to infinity when approaching its boundary. It was shown in [206] that these methods reach the optimal solution within a given accuracy in a polynomial time as long as the barrier function exhibits a certain regularity property : the *self-concordance*. These methods have been sucessfully implemented and are employed by various solvers for linear, quadratic and semidefinite optimization.

These attractive characteristics justify why a key method in global optimization consists of determining a sequence of convex problems that solve or approximate the original problem, as for instance :

- an equivalent reformulation, for instance, by means of the convex hull of the feasible set;
- a conservative approximation;

– a relaxation, for instance, through the use of a convex underestimator of the objective function;

– a decomposition into subproblems, for instance by partitioning the feasible set into convex pieces.

Thus, convexity permeates all field of optimization. Besides, problems that are natively convex arise in a variety of applications such as control systems, data analysis, statistics , finance, chemistry or localization problems. However, recognizing a convex problem can be a very challenging task. Indeed, more often than not, the most natural formulation is not convex and it may be a hard work to determine its expression in a convex way. For an exhaustive discussion on convex optimization, see the following references [59, 104].

1.1 Definitions and duality

In this section, we define two subfields of optimization, namely conic and convex optimization. For the latter, two definitions can be found in the literature. In this thesis, we make the choice to consider the more restrictive one, that do not include conic optimization. On the other hand, we show that any convex optimization problem can be converted into a conic optimization and therefore, convex optimization is a subfield of conic optimization.

The main sources for this section are the excellent discussion of Glineur about Conic Optimization [105] and the reference book of Boyd and Vandenberghe about Convex Optimization [59].

1.1.1 Definitions

Definition 1.1.1 *Conic optimization problem*
Let $\mathcal{K}$ be a proper cone (see Def. 2.2.34) of $\mathbb{R}^n$. Then the following optimization problem is a conic optimization problem :

$$\begin{cases} \inf & c^T x \\ s.t. & Ax = b \\ & x \in \mathcal{K} \end{cases}$$

for any $c \in \mathbb{R}^n$ and $(A, b) \in \mathbb{R}^{m,n+1}$.

Thus, the feasible set of a conic optimization problem is the intersection of the proper cone $\mathcal{K}$ with the hyperplane $\{x \in \mathbb{R}^n : Ax = b\}$.

In particular, this framework includes the following famous optimization area :

– $\mathcal{K} = \mathbb{R}^n_+$ (nonnegative orthant) $\rightarrow$ Linear Programming (LP) ;

– $\mathcal{K} = \{\begin{pmatrix} x_0 & x^T \end{pmatrix} : \|x\| \leq x_0\}$ (second-order cone) $\rightarrow$ Second-Order Conic Programming (SOCP);

– $\mathcal{K} = \mathbb{S}^n_+$ (cone of positive semidefinite matrices) $\rightarrow$ Semidefinite Programming (SDP).

These optimization areas are listed in such a way that each area includes the previous one. For example, any SOCP problem can be put under the form of a SDP.

Definition 1.1.2 *Convex optimization problem*
The following optimization problem is convex if the functions $f_i,\ i = 0, ..., m$ are convex (see Def. 2.4.33).

$$\begin{cases} \inf & f_0(x) \\ s.t. & f_i(x) \leq 0,\ i = 1, ..., m \\ & x \in \mathbb{R}^n \end{cases}$$

Remark that equality constraint are allowed since $f_i(x) = 0$ is equivalent to $f_i(x) \leq 0$ and $-f_i(x) \leq 0$, provided that the function f_i be both convex and concave, or equivalently, linear. Without loss of generality, we might consider that the objective function is also linear. If it is not the case, it suffices to convert the problem into the following convex optimization problem :

$$\begin{cases} \inf & x_0 \\ \text{s.t.} & f_0(x) \leq x_0 \\ & f_i(x) \leq 0, \ i = 1, ..., m \\ & x \in \mathbb{R}^n \end{cases}$$

The above definition implies that the feasible set $\mathcal{F} = \{x \in \mathbb{R}^n : f_i(x) \leq 0, \ i = 1, ..., m\}$ of an optimization problem is convex (see Def. 2.2.8). The converse is not true, which means that with this definition, there exists optimization problem with a convex objective function and a convex feasible set, that are not convex optimization problem.

Example 1.1.3 *Consider for instance the set :* $\mathcal{F} = \{x \in \mathbb{R}^n : \|Ax + b\| \leq d^T x + e\}$. *This set is convex, since it is the intersection of the second-order cone and of an hyperplane. However, the following formulation as a QCQP, obtained by squaring the inequality, involves potentially non-convex function,*

$$x \in \mathcal{F} \Leftrightarrow \begin{cases} d^T x + e \geq 0 \\ x^T (A^T A - dd^T)x + 2(bA - ed^T)x + b^2 - e^2 \leq 0 \end{cases}$$

The matrix $A^T A - dd^T$ *might be not positive semidefinite, for instance if* $A = 0$, *in which case the optimization problem is not convex.*

More generally, for an arbitrary QCQP, determining if there exists an equivalent convex formulation is a NP-hard problem.

An optimization problem consisting of optimizing a convex objective function over a convex set is called *abstract convex optimization problem* in the literature. It was shown in [205] that any abstract convex optimization problem (and therefore, any convex optimization problem), can be written as a conic optimization problem. A very comprehensive proof can be found in [105]. Clearly a conic optimization problem is an abstract convex problem and these two classes of problem are therefore equivalent.

These problems are more prevalent in practice that it is a priori thought, either directly or by means of an approximation. Recognizing such a problem has significant advantages :

- If a local minimum exists, then it is a global minimum ;
- There is an underlying fairly complete theory, which induces strong duality under certain conditions ;
- Under mild computability and boundedness assumptions, these problem are polynomially solvable.

The polynomial solvability is proved by applying the Ellipsoid method, that can be viewed as an algorithmic realization of the separation theorem for convex sets. This method requires the computation in polynomial time of a separating hyperplane, also called *separation oracle*, for some non feasible solutions x. For a convex optimization problem, the subgradients of the functions f_i such that $f_i(x) > 0$ are used. For the most widespread conic programs, it is also possible to compute such an hyperplane. But in general, for abstract convex optimization problems, determining a separation oracle might be impossible in polynomial time.

Note that for those both problems, we use the terminology inf instead of min since it may happen that the infimum of $c^T x$ on $\mathcal{F}$ is not attained. Indeed, the feasible set is necessarily closed but it might be unbounded, and then not compact, which prevents from applying the Weierstrass' theorem 2.1.27.

Definition 1.1.4 *The problem is said :*

- *infeasible if $\mathcal{F} = \emptyset$. We write this $p^* = +\infty$. Otherwise, the problem is said feasible ;*
- *asymptotically solvable if there exists a sequence of points of $\mathcal{F}$ whose objective values tend to p^* but whose limit is not feasible;*
- *solvable if there exists $x^* \in \mathcal{F}$ such that $c^T x^* \leq c^T x,\ \forall x \in \mathcal{F}$;*
- *unbounded if for all real C, there exists $x \in \mathcal{F}$ such that $c^T x \leq C$. Then the infimum is $p^* = -\infty$.*

1.1.2 Convex duality

We consider the following convex problem in the standard form :

$$\left\{ \begin{array}{lll} p^* = & \inf & c^T x \\ & \text{s.t.} & f_i(x) \leq 0,\ i = 1, ..., m \\ & & x \in \mathbb{R}^n \end{array} \right. \tag{1.1}$$

The Lagrangian of this problem, obtained by augmenting the objective function with a weighted sum of the constraint functions, is as follows :

$$\begin{array}{rccl} L & : & \mathbb{R}^n \times \mathbb{R}^m & \to \mathbb{R} \\ & & (x, y) & \mapsto L(x, y) = c^T x + \sum_{i=1}^m y_i f_i(x) \end{array} \tag{1.2}$$

Then $p^* = \inf_{x \in \mathbb{R}^n} \sup_{y \in \mathbb{R}^m_+} L(x, y)$. The dual problem is obtained by switching inf and sup, i.e., $d^* = \sup_{y \in \mathbb{R}^m_+} \inf_{x \in \mathbb{R}^n} L(x, y)$. By defining the *Lagrange dual function* $l : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ as follows : $l(y) = \inf_{x \in \mathbb{R}^n} L(x, y)$, it comes that the dual problem is :

$$\left\{ \begin{array}{lll} d^* = & \sup & l(y) \\ & \text{s.t.} & y \geq 0 \\ & & y \in \mathbb{R}^m \end{array} \right. \tag{1.3}$$

For any $y \geq 0$, $l(y)$ is a lower bound of p^* and therefore the *weak duality*, i.e., $d^* \leq p^*$, holds, and this is valid for any optimization problem, as discussed in Paragraph 3.1.5.1.

Sufficient conditions for strong duality are given by the Slater's theorem. They involve the property of *strict feasibility* of an optimization problem, which means that $\text{rint}(\mathcal{F}) \neq \emptyset$ if $\mathcal{F}$ is the feasible set. In the case of the problem (1.1), by expliciting the equality constraints $Ax = b$, it comes to require that there exists x such that $Ax = b$ and $f_i(x) < 0,\ i = 1, ..., m$.

Theorem 1.1.5 *Slater's theorem*
Let us consider (P) a convex optimization problem and its dual (D). $\mathcal{F}$ and $\mathcal{F}^*$ denote the feasible set of (P) and (D) respectively. Then,

- *If $\mathcal{F}$ and $\text{rint}(\mathcal{F}^*)$ are not empty, then (P) has a non-empty compact set of solutions and $p^* = d^*$*
- *If $\text{rint}(\mathcal{F})$ and $\mathcal{F}^*$ are not empty, then (D) has a non-empty compact set of solutions and $p^* = d^*$*
- *If $\text{rint}(\mathcal{F})$ and $\text{rint}(\mathcal{F}^*)$ are not empty, then both (P) and (D) have a non-empty compact set of solutions and $p^* = d^*$*

In short, the existence of a strictly feasible solution for one problem guarantees that the other problem attains its optimum. Furthermore, it suffices that at least one problem be strictly feasible for strong duality to hold. Having one problem strictly feasible is known as *Slater's condition*, which is one particular case of constraints qualification (see Paragraph 3.1.5.3).

In the case where the functions $f_i,\ i = 1, ..., m$ are differentiable, we can apply the KKT necessary conditions for optimality (see Paragraph 3.1.5.2), provided that the constraints satisfy the constraints qualification, which is the case if the primal is strictly feasible (Slater's condition). Furthermore, if the problem is convex, even if the constraint are not qualified, then the KKT conditions are sufficient for local optimality and therefore, for global optimality. We deduce from this the following theorem :

Theorem 1.1.6 *KKT for convex optimization problems*
Consider the problem (1.1) and assume that there exists $\bar{x} \in \mathbb{R}^n$ such that $A\bar{x} = b$ and $f_i(\bar{x}) < 0,\ i = 1, ..., m$. Then x^* is an optimal solution of the problem if and only if the exists $y^*, z^* \in \mathbb{R}^m \times \mathbb{R}^p$ such that :

$$\begin{cases} Ax^* = b,\ f_i(x^*) \leq 0,\ i = 1, ..., m & \textit{(primal feasibility)} \\ y_i^* \geq 0,\ i = 1, ..., m & \textit{(dual feasibility)} \\ y_i^* f_i(x^*) = 0,\ i = 1, ..., m & \textit{(complementary slackness)} \\ c + \sum_{i=1}^m y_i^* \nabla f_i(x^*) - {z^*}^T A = 0 & \textit{(Lagrangian stationarity)} \end{cases} \tag{1.4}$$

1.1.3 Conic duality

Consider the following conic optimization problem, with $\mathcal{K}$ a proper cone of $\mathbb{R}^n$, $c \in \mathbb{R}^n$ and $(A, b) \in \mathbb{R}^{m,n+1}$:

$$\begin{cases} p^* = & \inf & c^T x \\ & \text{s.t.} & Ax = b \\ & & x \in \mathcal{K} \end{cases} \tag{1.5}$$

The conic formulation provides a very elegant formulation of the dual problem by means of the dual cone $\mathcal{K}^*$ of $\mathcal{K}$ (see Def. 2.2.40). Let us begin by defining the Lagrangian involving the equality constraints :

$$\begin{array}{rccl} L: & \mathbb{R}^n \times \mathbb{R}^m & \to \mathbb{R} \\ & (x, y) & \mapsto L(x, y, z) = c^T x - y^T x + z^T (b - Ax) \end{array}$$

Then $p^* = \inf_x \sup_{y \in \mathcal{K}^*} L(x, y, z)$. Indeed, $\sup_{y \in \mathcal{K}^*} -y^T x = 0$ if $x \in \mathcal{K}$, $+\infty$ otherwise. Then the dual problem is obtained by switching sup and inf : $d^* = \sup_{y \in \mathcal{K}^*} \inf_x L(x, y, z)$:

$$\inf_x L(x, y, z) = \inf_x \quad b^T z + (c - y - A^T z)^T x = \begin{cases} b^T z \text{ if } c - y - A^T z = 0 \\ -\infty \text{ otherwise} \end{cases}$$

Therefore we conclude :

$$\begin{cases} d^* = & \sup & b^T z \\ & \text{s.t.} & y = c - A^T z \\ & & y \in \mathcal{K}^* \end{cases}$$

We note that it has the same structure as the primal, that is the intersection of a cone with an affine subspace. Furthermore, from Proposition 2.2.43, we know that the dual cone of a proper cone is also a proper cone. With this formulation, in virtue of Prop. 2.2.42, it is easy to show that the dual of the dual is the primal.

As for any optimization problem, the weak duality holds. The convexity of the feasible set brings additional properties, involving the *strict feasibility*, i.e. the existence of strictly feasible solution : $x \in \text{rint}(\mathcal{K})$ such that $Ax = b$.

Theorem 1.1.7 *Slater's theorem for conic optimization*
Let us consider (P) a conic optimization problem and its dual (D). Then,

- *If (P) is feasible and (D) is strictly feasible, then (P) has a non-empty compact set of solutions and $p^* = d^*$*

– *If (P) is strictly feasible and (D) is feasible, then (D) has a non-empty compact set of solutions and $p^* = d^*$*
– *If (P) and (D) are strictly feasible, then both (P) and (D) have a non-empty compact set of solutions and $p^* = d^*$*

For more complete results on duality of conic programs, we refer the reader to the reference [29] and to [120] for a very comprehensive summary.

To conclude this paragraph, we derive the KKT conditions for a conic optimization problem. Under the assumptions of constraints qualifications, for instance if there exists a primal strictly feasible point, then the following conditions are both sufficient and necessary for the optimality of x^* :

Theorem 1.1.8 *KKT for conic optimization problems*
Consider the conic optimization problem (1.5) and assume that there exists $\bar{x} \in \text{int}(\mathcal{K})$ such that $A\bar{x} = b$. Then x^* is an optimal solution of the problem if and only if the exists $y^*, z^* \in \mathbb{R}^n \times \mathbb{R}^p$ such that :

$$\begin{cases} Ax^* = b,\ x^* \in \mathcal{K} & \textit{(primal feasibility)} \\ y^* \in \mathcal{K}^* & \textit{(dual feasibility)} \\ {y^*}^T x^* = 0 & \textit{(complementary slackness)} \\ c - y^* + A^T z^* = 0 & \textit{(Lagrangian stationarity)} \end{cases}$$

1.2 Complexity and algorithms

Since the claim by Klee and Minty [157] that the simplex method is not polynomial, the question of the complexity of a linear program, and more generally, of any mathematical program, was raised. It was partly answered by Khachiyan [153] in 1979, when he adapted the Ellipsoid method to Linear Programming and proved its polynomial complexity. Then this work was extended to the optimization of a linear objective over a convex set, under the existence of a polynomial time separation oracle for this convex set [118]. This allows to classify as polynomial a large part of the convex optimization problem.

However, it is worth noticing that all abstract convex optimization problems are not polynomial (unless P=NP), the most famous example being the problem of minimizing a linear function over the cone of co-positive (or completely positive) matrices.

In practice, the Ellipsoid method does not work very well, especially when compared to the simplex, which gives an excellent example that theory can not always be relied upon for predicting applicability. But some other methods, called *Interior-point method* and sparked by the seminal work of Karmarkar [150], proved to be numerically very efficient, provided that there exists barrier functions for the feasible set satisfying the property of *self-concordance*.

1.2.1 Ellipsoid method

Complexity of Linear Programs was an open problem until 1979, with the discovery of the first worst-case polynomial-time algorithm for Linear Programming. Sprouted from the iterative methods of Shor (1977) and the approximation algorithms of A. Nemirovski and D. Yudin (1976), this so-called *Ellipsoid method* is due to L. Khachiyan [153] who designed the method and proved its polynomiality. Thus, to solve a problem with n variables that can be encoded in L input bits, the algorithm uses $O(n^4 L)$ pseudo-arithmetic operations on numbers with $O(L)$ digits.

However, it turns out the method performs poorly compared to the simplex method (even though not polynomial). But the Ellipsoid method is nevertheless of great theoretical value, since it proved that LP is in class P and this result was extended in [118, 119] to any problem with a linear objective

for which there exists a *separation oracle*, i.e., a polynomial way of checking whether a given vector belongs to the feasible set, and if not, exhibiting a violated linear inequality.

Note that the fundamental version of Ellipsoid method applies to *feasibility problem* , i.e., given a set $\mathcal{K}$, find $x \in \text{int}(\mathcal{K})$. However, it is possible to transform an optimization problem into a feasibility problem for instance through a binary search on the value of the objective function. For example, for a minimization problem with a linear objective $c^T x$, the feasibility of $\{x \in \mathcal{K} : c^T x \leq \gamma_k\}$ is tested at each step k, with γ_k the sequence determined by the binary search.

Finally, this pioneering method opened the door to numerous interior point methods because, unlike the simplex, the solution is reached by iterating on interior-point of the feasible set. However, in the literature, it is not considered as being part of the interior-point methods. For details and further references on this topic, the reader is directed to [15, 220].

1.2.1.1 Basic idea

The original version of the Ellipsoid method applies to the case where $\mathcal{K}$ is a polyhedron. More precisely, $\mathcal{K} = \{x \in \mathbb{R}^n : Ax \leq b\}$, with $(A, b) \in \mathbb{R}^{m,n+1}$. The method aims at finding x in the interior of $\mathcal{K}$, i.e., such that $Ax < b$ (strict feasibility). The original version of Khachiyan also requires that $\mathcal{K}$ be bounded and assume that the input data are integer or rational numbers and L is the length of their binary encoding.

The basic idea is to generate a sequence of ellipsoids containing $\mathcal{K}$, that can be viewed as bounding volumes used to locate $\mathcal{K}$. If a center of any ellipsoid in this sequence belongs to $\mathcal{K}$, then it is discovered. Otherwise the process stops when the volume of the current ellipsoid is too small to contain $\mathcal{K}$, which implies that $\mathcal{K}$ is empty.

The algorithm proceeds as follows :

1: Find an ellipsoid $E_0 \supset \mathcal{K}$ and its center x_0 ;
2: **while** $x_0 \notin \mathcal{K}$ **do**
3: Find an inequality (=separation oracle) (π_0, π) such that $\pi^T x \leq \pi_0, \forall x \in \mathcal{K}$ and $\pi^T x_0 > \pi_0$;
4: Push the (π_0, π) until it hits x_0, giving you a half-ellipsoid HE that contains $\mathcal{K}$;
5: Find a new ellipsoid $E_1 \supset HE$, such that : $\frac{volume(E_1)}{volume(E_0)} \leq e^{-1/2n} < 1$;
6: $E_0 \leftarrow E_1$;
7: **end while**

We do not get into the details of the construction of the ellipsoid E_1 and the volume formula, see [119, 220] for a whole explanation of these elements.

At each iteration, the ellipsoid containing $\mathcal{K}$ is shrunk by factor at least $f = \exp(-\frac{1}{2(n+1)}) < 1$. Consequently, within a number $O((n+1)^2 L)$ of iterations, the incumbent ellipsoid reaches a volume less than twice the volume of $\mathcal{K}$, which guarantees that the center of the ellipsoid belongs to $\mathcal{K}$.

This algorithm can be applied to determine a solution of the system $Ax \leq b$, as stated by the following equivalence :

$$Ax \leq b \text{ is feasible } \Leftrightarrow Ax < b + \epsilon \text{ is feasible for all } \epsilon > 0 \tag{1.6}$$

The "if" part is clear. Conversely, the theorem of the alternatives (Theorem 2.3.50) states that $Ax \leq b$ has no solution if and only if there exists $y \in \mathbb{R}^m$ such that $y \geq 0$, $A^T y = 0$ and $b^T y < 0$. Then, for any $\epsilon > 0$, $(b+\epsilon)^T y = b^T y + \epsilon^T y < 0$ for sufficiently small ϵ. This is in contradiction with $Ax < b + \epsilon$ since $y \geq 0$ implies that $0 = (Ax)^T y < (b + \epsilon)^T y$.

Moreover, the following equivalence states that it is sufficient to take one value $\epsilon < 1/n2^{2L}$ for deciding if $Ax \leq b$ is feasible or not :

$$Ax \leq b \text{ is feasible } \Leftrightarrow \text{ for any } 0 < \epsilon < 1/n2^{2L}, Ax < b + \epsilon \text{ is feasible} \tag{1.7}$$

Consequently, it suffices to pick any $\epsilon \in]0, 1/n2^{2L}[$ and to apply the ellipsoid method to $Ax < b + \epsilon$.

1.2.1.2 Extension to convex optimization

The beauty of the Ellipsoid method is that it does not require a complete and explicit description of $\mathcal{K}$. It suffices to be able to test whether a given $x_0 \in \mathcal{K}$ (so-called *membership testing*), and if not, to provide a separating hyperplane. This constitutes a so-called *separation oracle*. Thus, as presented by Grotschel,Lovász and Schrijver in [118], the Ellipsoid method can be extended to the problem of finding a point in the interior of an arbitrary convex set $\mathcal{K}$. Indeed, it can be viewed as an algorithmic realization of the separation theorem for convex sets (see Theorem 2.2.23).

The fundamental result of convex optimization is that, if the separation oracle runs in polynomial time and returns a separating hyperplane of polynomial size, so does the algorithm and the associated problem is then proved to belong to the class P.

As the set $\mathcal{K}$ is convex, then the separation oracle exists necessarily but might be impossible to compute in polynomial-time. For example, the problem of optimizing a linear objective over the cone of co-positive (or completely positive) matrices is convex but is NP-hard because of the lack of polynomial-time separation oracle.

Recall that to apply the Ellipsoid method to an optimization problem, we have to resort to a binary search on the optimal value. The question that arises is when to stop the binary search. With a linear program, the optimal is known to be a rational of bounded repesentation size, and therefore it is necessary attained within a finite number of steps. But for a general set $\mathcal{K}$, it is possible that the optimal can not be attained in a finite number of steps, for example if it is irrational.

This issue is not specific to binary search. In fact, there are some convex sets for which feasibility can not be determined via the Ellipsoid method. An extreme example is the case where the convex set is a single point.

For this reason, besides the initial notion of separation oracle, which is subsequently denoted *strong separation oracle*, we define a relaxed notion, the *weak separation oracle*, that allows for approximations.

Definition 1.2.1 *Strong separation oracle*
A strong separation oracle for $\mathcal{K}$, when given as input $x_0 \in \mathbb{R}^n$, either returns the assertion that $x_0 \in \mathcal{K}$, or $c \in \mathbb{R}^n$ such that $c^T x < c^T x_0$ for all $x \in \mathcal{K}$.

Definition 1.2.2 *Weak separation oracle*
A weak separation oracle for $\mathcal{K}$, when given as input $x_0 \in \mathbb{R}^n$ and a rational $\epsilon > 0$, either returns the assertion that $x_0 \in \mathcal{K}^{+\epsilon}$, or $c \in \mathbb{R}^n$ such that $\|c\|_\infty \geq 1, c^T x < c^T x_0 + \epsilon$ for all $x \in \mathcal{K}^{-\epsilon}$, where

- *$\mathcal{K}^{+\epsilon} = \{y : \|y - x\| \leq \epsilon,$ for some $x \in \mathcal{K}\}$ the set of points "almost" in $\mathcal{K}$;*
- *$\mathcal{K}^{-\epsilon} = \{x \in \mathcal{K} : B(x, \epsilon) \subset \mathcal{K}\}$ the set of points "deep" in $\mathcal{K}$.*

The constraint $\|c\|_\infty \geq 1$ is required to prevent $c = 0$ to be solution.

This leads to the following theorem [118], a fundamental result of convex optimization :

Theorem 1.2.3
Consider the optimization problem - $\min c^T x : x \in \mathcal{K}$ - with $\mathcal{K}$ a convex set. Assume that there exists a weak separation oracle for $\mathcal{K}$ and a rational R such that $\mathcal{K} \subset B(0, R)$. Then, the so-called ϵ-weak optimization over $\mathcal{K}$, i.e.,

- *either asserts that $\mathcal{K}^{-\epsilon}$ is empty;*
- *or returns $y \in \mathcal{K}^{+\epsilon}$ such that $c^T x \leq c^T y + \epsilon, \forall x \in \mathcal{K}^{-\epsilon}$.*

can be solved in polynomial time of (n, R, ϵ).

1.2.1.3 Optimization versus Separation

Definition 1.2.4 *Separation algorithm*
For a given class of inequalities, a separation algorithm is a procedure which, given a vector x as input, either finds an inequality in the class which is violated by x, or proves that none exists (see [118]).

Given a convex optimization problem of the form $\min_{x\in\mathcal{C}} c^T x$, its associated *Separation Problem* consists of deciding whether a given x_0 belongs to $\mathcal{C}$ and if not, return a *certificate*, i.e. a linear constraint valid for $\mathcal{C}$ that is violated by x_0.

Then, it is equivalent to solve the separation problem in polynomial time or to solve the optimization problem in polynomial time. Indeed, by using Ellipsoid method (see 1.2.1) it is clear that if you can separate in polynomial time, then you can solve the optimization problem in polynomial time. Conversely, if we can optimize any linear objective over a polyhedron P, then for any $\begin{pmatrix}\pi_0 & \pi^T\end{pmatrix}^T$, we can compute $\pi^* = \max_{x\in P} \pi^T x$ and compare it to π_0. If $\pi^* \leq \pi_0$ then $\begin{pmatrix}\pi_0 & \pi^T\end{pmatrix}^T$ belongs to $P^\bullet$, otherwise, there exists $x^* \in P$ such that $\pi^T x^* = p^* > \pi_0$, which serves as a certificate that $\pi \notin P^\bullet$.

Consequently, if we are able to optimize over P, then we are able to solve the separation problem over $P^\bullet$. Then, we deduce that we are able to solve the separation problem over P as illustrated on the following diagram :

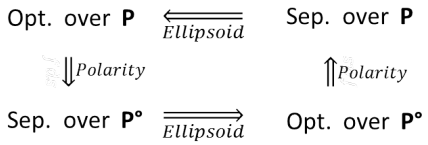

Figure 1.1: Equivalence between optimization and separation

1.2.2 Subgradient and bundle methods

Subgradient method are an extension of the Gradient Descent method (see Paragraph 3.2.2), for handling non-differentiable functions. Their main advantages are their simple implementation and the fact that they can easily be applied to large-scale problems, combined with decomposition techniques. In addition, there require little storage. In the case of differentiable problems, these first-order methods converge to a KKT solution and therefore they are particularly appropriate for convex optimization problem.

During the 1970s, a new motivation for this class of methods was triggered by the work of Lemaréchal, that proposed a new method called *Bundle Method*, for the unconstrained minimization of a non-differentiable convex function. The basic idea is to store the successive generated subgradient and to use this *bundle* of supporting hyperplane to define a piecewise linear underestimator of the function to minimize. Then, the problem can be solved by Linear Programming. This method can be applied to constrained optimization by minimizing the Lagrangian of the problem, for fixed Lagrangian multipliers.

Bundle method is today the reference method for non-differentiable convex optimization problems. In particular, it can be well applied to the case where the variables set can be decomposed into subset, almost independent one from the others. We refer the reader to [137] for a good overview on these methods.

1.2.3 Interior-point methods

Interior-points methods are so called because their main idea is to iterate into the interior of the feasible set of the considered problem. They were introduced by Karmarkar in 1984 [150] and played a key role in the development of mathematical programming. In 1996, Freund and Mizuno [97] wrote "Interior-point methods have permanently changed the landscape of mathematical programming theory, practice and computation". In particular, they was one triggering factor of the major development of semidefinite programming even if they were initially designed for linear programming. Indeed, in 1988, a major breakthrough was achieved by Nesterov and Nemirovski [206], who extended this interior-point approach to general convex problems, while conserving the polynomiality under relevant conditions. For this reason, these methods are crucial for convex optimization.

In more practical terms, interior-point methods are an extension of the Newton's method to constrained optimization problem. The Newton's method and its possible application to optimization problem involving equality constraint are briefly described at Appendix 3.2.3. Furthermore, a first glimpse on interior-point methods for linear programming is provided at Appendix 3.3.2.

In this paragraph, we complete these preliminaries by considering a problem involving convex inequalities or conic constraints. To do this, we get rid off these constraints by adding to the objective a barrier function of that goes to infinity at the boundaries of the feasible region. Then, reducing the strength of this barrier at each iteration allows to arbitrarily approach the solution of the problem.

In the case where the whole objective function (original objective augmented by the barrier function), exhibits the property of *self-concordance*, then the method reaches any desired precision in a polynomial number of iterations, which makes the method polynomial.

Due to their efficiency and popularity, studies on that topic have flourished, which yield a wide range of algorithms that shares the same basic principles but whose individual features may vary a lot. At first, the **iterate space** can vary : a method is said to be *primal*, *dual* or *primal-dual* when its iterate belong respectively to the primal space, the dual space or the Cartesian product of these spaces.

Further, the methods are called *feasible* when the iterates are necessarily feasible, that is they satisfy both the equality and nonnegativity constraints. In the case of *infeasible* method, the iterates may not satisfy the equality constraints, but are still required to satisfy the nonnegativity conditions. That's the **type of iterate** criteria.

Besides that, the **type of step** can vary : some algorithms, called *short-stem methods* uses a very short step at each iteration, leading to a high number of iteration. The *long-step methods*, which are prevalent in practice, are allowed to take much longer steps.

Finally, interior-point methods can be classified into three major categories depending on the **type of algorithm** :

- Affine-scaling algorithms ;
- Projective methods with a potential function ;
- Path-following algorithms.

This paragraph is organized as follows : first, we introduce the requirements necessary to the acquaintance of the methods. Then, we will give the key idea of each type of algorithms, in order to highlight their underlying principles. Finally, we will describe in detail the path-following primal-dual method for linear programming. Indeed, it is very popular and implemented in many currently available codes.

1.2.3.1 Preliminaries

Barrier function
Let us consider the following problem : $\min c^T x : Ax = b, x \in \mathcal{K}$, where $\mathcal{K}$ is either a proper cone, or a set defined by means of convex inequalities : $\mathcal{K} = \{x \in \mathbb{R}^n : f_i(x) \leq 0, \quad i = 1, ..., m\}$.

This problem is equivalent to $\min f_0(x) + I_{\mathcal{K}}(x) : Ax = b$ if $I_{\mathcal{K}}(x)$ is a function that returns 0 if $x \in \mathcal{K}$, $+\infty$ otherwise. But this function is not differentiable, which makes impossible the application of the Newton's method. The idea is to approximate it by a differentiable function, defined on $\text{int}(\mathcal{K})$, that tends to $+\infty$ as x approaches the boundary of $\mathcal{K}$. Such a function is called a *barrier function* for $\mathcal{K}$. In the convex optimization case, we can take for instance $\phi(x) = -\sum_{i=1}^{m} \log(f_i(x))$.

Then, the problem becomes :

$$(P_\mu) \begin{cases} \min & c^T x + \mu\phi(x) \\ \text{s.t.} & Ax = b \end{cases} \tag{1.8}$$

where μ is a positive parameter. Intuitively, we understand that adding $\mu\phi(x)$ to the objective exerts a repelling force from the boundary of $\mathcal{K}$ and therefore prevents the constraint $x \in \mathcal{K}$ to be violated.

Central path
In the case of a convex optimization problem, applying KKT conditions to the problem (1.8) yields the following system :

$$(KKT_\mu) \begin{cases} Ax = b,\ f_i(x) \leq 0,\ i = 1, ..., m \\ y \geq 0 \\ -y_i f_i(x) = \mu,\ i = 1, ..., m \\ c + \sum_{i=1}^{m} y_i \nabla f_i(x) + A^T z = 0 \end{cases}$$

Thus, this problem is similar to the KKT system of the original problem (see Theorem 1.1.6) except that the right-hand term complementarity condition equals μ instead of 0. Thus, if μ tends to 0, then the solution of the system (KKT_μ) tends to the solution of the original KKT system, which is an optimal solution for the considered problem.

Then the central path is defined as the set of the solution of (KKT_μ) as μ varies. By analogy, this definition is extended to conic optimization problems :

$$(KKT_\mu) \begin{cases} Ax = b,\ x \in \mathcal{K} \\ y = c - A^T z,\ y \in \mathcal{K}^* \\ y_i x_i = \mu,\ i = 1, ..., n \\ c - y + z^T A = 0 \end{cases}$$

If (x, y, z) belongs to the central path, then the duality gap equals $c^T x - b^T y = x^T y = n\mu$. For this reason, μ is called the *duality measure*.

Self-concordance
Analysing the Newton's method for the unconstrained minimization of a convex, twice differentiable, function f suffers from the drawback of depending on three unknown constant, that are dependent on affine change of coordinates.

One significant result of Nesterov and Nemirovski in [206] is to show that this is not the case any more whenever the function f has the property of *self-concordance*, which is affine-invariant.

Definition 1.2.5 *Self-concordance*
A convex function $f : \mathbb{R} \to \mathbb{R}$ is self-concordant if $|f'''(x)| \leq 2f''(x)^{3/2}$. A convex function $f : \mathbb{R}^n \to \mathbb{R}$ is self-concordant if the function $\hat{f}(t) = f(x + tv)$ is self-concordant for all direction v.

In particular, the linear and quadratic convex functions are self-concordant.

1.2.3.2 Path-following algorithms

As suggested by their denomination, the main idea behind these methods is to follow the central path up to a limit point. Very roughly, the principle is the following : given an initial iterate v_0 and a sequence of positive number real numbers decreasing to zero μ_k, use Newton's method to compute v_{k+1} from v_k such that v_{k+1} belongs to the central path with the duality measure μ_{k+1}.

The difficulty is that to find a point which is exactly on the central path may require a high number of Newton's iterations. By limiting this number, we compute points that are approximatively on the central path, and thus, only loosely follow the central path.

Let us give some elements on each stage of the method :

1. $\mu_{k+1} = \sigma\mu_k$ where σ is a constant strictly between 0 and 1
2. the next iterate v_{k+1} is computed by applying one single Newton step Δv_k
3. in order to ensure that v_{k+1} is feasible, the Newton's step is damped : $v_{k+1} = v_k + \alpha_k \Delta v_k$, where α_k is maximal.

The Newton's step is computed as the solution of a system, which depends on the space of iterate :

- Primal-dual system : $\Delta v_k = \begin{pmatrix} \Delta x_k & \Delta y_k & \Delta s_k \end{pmatrix}$ such that :

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S_k & 0 & X_k \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta y_k \\ \Delta s_k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -X_k S_k e + \sigma\mu_k e \end{pmatrix}$$

  This system is actually obtained as the application of the Newton's method to the resolution of the KKT conditions.

- Primal system : $\Delta v_k = \begin{pmatrix} \Delta x_k \end{pmatrix}$. We cannot deduce the Newton's step from the KKT conditions anymore since they involve both primal and dual variables. We apply instead a single minimizing Newton's step to the (P_μ) barrier problem, as described in paragraph3.2.3.

$$\begin{pmatrix} \mu_k X_k^{-2} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x_k \\ -y_k \end{pmatrix} = \begin{pmatrix} -c + \sigma\mu_k X_k^{-1} e \\ 0 \end{pmatrix}$$

- Dual system : $\Delta v_k = \begin{pmatrix} \Delta y_k & \Delta s_k \end{pmatrix}$. As for the primal, we apply a single minimizing Newton's step to the (D_μ) :

$$\begin{pmatrix} A^T & I \\ AS_k^{-2}A^T & 0 \end{pmatrix} \begin{pmatrix} \Delta y_k \\ \Delta s_k \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{b}{\sigma\mu_k} - AS_k^{-1}e \end{pmatrix}$$

where $X_k = \mathrm{Diag}(x_k)$ and $S_k = \mathrm{Diag}(s_k)$.

1.2.3.3 Affine scaling algorithms

Affine scaling methods are a variant of Karmarkar's original method. This method used projective transformations and was not described in terms of central path or Newton's method. Later, researchers simplified this algorithm, removing the need for projective transformations and obtained a class of method called *affine scaling* algorithms. It was later discovered that these methods have been previously proposed by Dikin [85], 17 years before Karmarkar.

Let us illustrate the basic idea of these methods on the primal linear problem :

$$(P) \begin{cases} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{cases}$$

This problem is hard to solve because of the nonnegativity constraints, which give the feasible region a polyhedral shape. Let us consider the current iterate x_k and replace the polyhedral feasible region by an inscribed ellipsoid centered at x_k. The idea is that minimizing the objective function on this ellipsoid is easier that on a polyhedron. The obtained solution will be taken as next iterate.

The first step is to scale the data of the problem in order to map the current iterate x_k to e, by using the matrix $D = \text{Diag}(x_k)$. That's how the method gets its denomination.

$$(P) \begin{cases} \min & (Dc)^T w \\ \text{s.t.} & ADw = b \\ & w \geq 0 \end{cases}$$

Thus, the current iterate for w is e. We replace the constraint $w \geq 0$ by the $\|w - e\| \leq 1$, which is a restriction of the feasible set. Then, the solution can be analytically computed via a linear system, which leads to the next iterate x_{k+1}.

1.2.3.4 Potential-reduction algorithms

Instead of targeting a decreasing sequence of duality measures, the method of Karmarkar made use of a potential function to monitor the progress of its iterate. A potential function is a way to measure the worth of an iterate. Its main two properties are the following :

- it tends to $-\infty$ if and only if the iterates tends to optimality
- it tends to $+\infty$ when the iterates tends to the boundary of the feasible region without tending to an optimal solution

The main goal of a potential reduction algorithm is simply to reduce the potential function by a fixed amount δ at each step, hence its name.

In practical terms, once the Newton's step Δv_k has been computed (as in the path-following method), the potential function is used to determine α_k so that $v_k + \alpha_k \Delta v_k$ minimizes this function over α_k.

An example of such a potential function for primal-dual method is given by Tanabe-Todd-Ye :

$$\phi_\rho(x, s) = \rho \log(x^t s) - \sum_i \log(x_i s_i)$$

1.2.3.5 Enhancements

Infeasible algorithms
The algorithms we have described above rely on the assumption that there exists a strictly feasible solution that can be used as starting point. However, such a point is not always available. In some cases, such a solution doesn't exist, otherwise it exists but finding it is quite as difficult as solving the whole linear program.

Two strategies can be adopted to handle such cases : embed the problem into a larger one that admits a strictly feasible solution (that is the object of the next paragraph), or modify the algorithm to make it work with infeasible iterates. Therefore, we will have iterates that respect the positivity constraint ($(x, s) > 0$) but not necessarily the equality constraints $Ax_k = b$ and $A^T y_k + s_k - c = 0$.

The idea is simply to target a next iterate that respect the equality constraints. For this, it suffices not to suppose that $A\Delta x_k = 0$, but $A\Delta x_k = Ax_k - b$, and the same for the other equality constraint. This leads to the following system :

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S_k & 0 & X_k \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta y_k \\ \Delta s_k \end{pmatrix} = \begin{pmatrix} c - A^T y_k - s_k \\ b - Ax_k \\ -X_k S_k e + \sigma \mu_k e - \Delta X_k^a \Delta S_k^a e \end{pmatrix}$$

Thereby, Newton's step will tend to reduce both the duality gap and the infeasibility at the same time.

Homogeneous self-dual embeddding
As mentioned below, another way to handle infeasibility is to embed our problem in a larger linear program that admits a known strictly feasible solution.

Let (x_0, y_0, s_0) be a point that respect the positivity constraint, that is $(x_0, s_0) > 0$. We define the following variables :

$$\begin{aligned} \hat{b} &= b - Ax_0 \\ \hat{c} &= c - A^T y_0 - s_0 \\ \hat{g} &= b^T y_0 - c^T x_0 - 1 \\ \hat{h} &= x_0^T s_0 + 1 \end{aligned}$$

Then we consider the following problem :

$$\begin{cases} \min & \hat{h}\theta \\ \text{s.t.} & Ax - b\tau + \hat{b}\theta = 0 \\ & -A^T y + c\tau - \hat{c}\theta - s = 0 \\ & b^T y - c^T x - \hat{g}\theta - \kappa = 0 \\ & -b^T y + c^T x + \hat{g}\tau = -\hat{h} \\ & x \geq 0, s \geq 0, \theta \geq 0, \kappa \geq 0 \end{cases}$$

The point $(x, y, s, \tau, \kappa, \theta) = (x_0, y_0, s_0, 1, 1, 1)$ is a strictly feasible solution for this problem. Here is a brief description of the new variables :

- τ is the homogenizing variable ;
- κ is measuring infeasibility ;
- θ refers to the duality gap in the original problem.

This program have the following properties :

- It is homogeneous, that is its right-hand side is the zero vector (except for the last equality that is a homogenizing constraint)
- It is self-dual
- The optimal value is 0 and $\theta^* = 0$
- Given a solution $(x^*, y^*, s^*, \tau^*, \kappa^*, 0)$, either $\tau^* > 0$, or $\kappa^* > 0$
  - if $\tau^* > 0$ then $(x^*/\tau^*, y^*/\tau^*, s^*/\tau^*)$ is an optimal solution to the original problem
  - if $\kappa^* > 0$ then the original problem has no finite optimal solution.

Since this problem have strictly feasible starting point, we can apply the path-following method, then, using the above-mentioned properties, we can readily obtain the optimal solution of the original problem or to detect its infeasibility.

The difficulty is that it is twice as large as the original problem. However, it is possible to take advantage of its self-duality to solve it with nearly the same computational cost as the original problem.

The Mehrotra predictor-corrector algorithm

This algorithm is an enhancement of the primal-dual path-following method. We have seen previously that one crucial point for this kind of method is the choice of the constant σ. The idea here is to adapt this constant to the current iterate. Moreover, once σ_k has been determined, the method re-uses some computational work to improve the current iterate.

About the constant σ, usually two choices are possible :

- Choosing σ nearly equal to 1, which means not reducing much the duality measure between two iterations. The advantage is that the current iterate will be close to the central path, so it allows to take almost full Newton's step without violating constraints. The disadvantage is that this step is usually short, and an iteration does not do much progress toward the solution.
- Choosing a small value for σ produces a large Newton's step which provides a good progress toward optimality, but the associated iterate is usually infeasible, so the step has to be damped. Moreover, this kind of step tends to move the iterate away from the central path.

This first idea of the Mehrotra's algorithm is to adapt σ to the current iterate. If the latter is not far from the central path and there is a far target easy to attain, a small value of σ is appropriate, in order to capitalize on this positive situation. On the other hand, if the current iterate is far from the central path, a small σ will have the effect of moving it closer, so that progress may be done at the next iteration.

For this, we carry out a first stage, called *predictor stage*, where a Newton's step $(\Delta x_k^a, \Delta y_k^a, \Delta s_k^a)$ is computed with $\sigma = 0$. That comes to move straigth toward the optimal solution. Then the maximum length of step are computed separately for the primal and the dual variables :

$$\begin{aligned}\alpha_p^a &= \arg\max\{\alpha \in [0,1] : x_k^a + \alpha \Delta x_k^a \geq 0\} \\ \alpha_d^a &= \arg\max\{\alpha \in [0,1] : s_k^a + \alpha \Delta s_k^a \geq 0\}\end{aligned}$$

The associated duality measure can be computed as following :

$$\mu_{k+1}^a = \frac{(x_k^a + \alpha_p^a \Delta x_k^a)^T (s_k^a + \alpha_d^a \Delta s_k^a)}{n}$$

If μ_{k+1}^a is much smaller than μ_k it means that much progress can be done toward the optimality, so σ has to be small. Otherwise a centrality correction is needed. This is put into practice by the following heuristic, which have proved to be very efficient in practice :

$$\sigma = \left(\frac{\mu_{k+1}^a}{\mu_k}\right)^3$$

Now, we can carry out the *corrector stage* by computing the Newton's step $(\Delta x_k, \Delta y_k, \Delta s_k)$ using this value of σ. Then we take the maximal feasible step lengths separately for the primal and the dual spaces.

The second idea of this algorithm is to improve the current iterate by using the computationnal work of the predictor stage. In this stage, since $\sigma = 0$, we target a zero value for each $x_i s_j$ product. After applying the full predictor step :

$$\begin{aligned}x_i s_j &= (x_{k,i}^a + \alpha_p^a \Delta x_{k,i}^a)(s_{k,j}^a + \alpha_d^a \Delta s_{k,j}^a) \\ &= \Delta x_{k,i}^a \Delta s_{k,j}^a\end{aligned}$$

since the equation of the Newton's system, as a first-order approximation, leads to :

$$x_{k,i}^a \Delta s_{k,j}^a + s_{k,j}^a \Delta x_{k,i}^a = -x_{k,i}^a s_{k,j}^a$$

Consequently, $x_i s_j$ measures the error due to the first-order approximation. The idea is to consider it as an approximation of the same error in the corrector stage, by using it in the right-hand term of the system :

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S_k & 0 & X_k \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta y_k \\ \Delta s_k \end{pmatrix} = \begin{pmatrix} c - A^T y_k - s_k \\ b - A x_k \\ -X_k S_k e + \sigma \mu_k e - \Delta X_k^a \Delta S_k^a e \end{pmatrix}$$

Let us point out that this correction is equivalent to compute a Newton's step from $v_k + \Delta v_k$ toward the solution of $F_\mu(v) = 0$. If we define the function $G_\mu(\Delta v) = F_\mu(v_k + \Delta v)$, it comes to apply the Newton's method to the equation $G_\mu(\Delta v) = 0$. Let $\Delta^2 v$ be the obtained Newton's step for this equation, finally the whole step is the sum $\Delta v + \Delta^2 v$. In our case, it can be proved that it would be the same as the Newton's step obtained by adding the second-order term in the right-hand size of the equation, if both corrector and predictor step were computed with the same value of σ.

Although there is no theoretical complexity bound on it yet, Mehrotra's predictor-corrector method is widely used in practice. If each iteration is marginally more expensive than a standard interior point algorithm, the additional overhead is usually paid off by a reduction in the number of iterations needed to reach an optimal solution. It also appears to converge very fast when close to the optimum.

1.3 Special cases of convex optimization

The most widespread subfield of convex optimization is Linear Programming, which is covered in Appendix 3.3. The main topic of this thesis, i.e., Semidefinite Programming, is also a subfield of conic programming, and will be discussed in the details in Chapter 2. Finally, we are interested here in a well-known subfield of conic programming, namely Second-Order Conic Programming (SOCP), and in a special convex optimization problem, when all the involved function are quadratic.

1.3.1 Second-Order Conic Programming

Second-order conic programming, as the name suggests, is a special case of conic programming where the cone is the second-order cone $\mathcal{K}_L$, also called *Lorentz cone* or *ice-cream cone*. This cone is the set of vectors of $\mathbb{R}^n$ such that the euclidian norm of the $n-1$ first components is less than or equal to the n-th component :

$$\mathcal{K}_L = \left\{ \begin{pmatrix} x_0 \\ x \end{pmatrix} \in \mathbb{R}^n : \|x\| \leq x_0 \right\}$$

A second-order conic program is therefore a problem of the form :

$$(P_{SOCP}) \begin{cases} \min & c_1^T x_1 + c_2^T x_2 + \cdots + c_r^T x_r \\ \text{s.t.} & A_1 x_1 + A_2 x_2 + \cdots + A_r x_r = b \\ & x_i \in \mathcal{K}_L, i = 1, \cdots, r \end{cases}$$

Proposition 1.3.1 *The second-order cone $\mathcal{K}_L$ is self-dual.*

Therefore, the dual problem of (P_{SOCP}) is the following :

$$(D_{SOCP}) \begin{cases} \min & b^T y \\ \text{s.t.} & c_i - A_i^T y \in \mathcal{K}_L, i = 1, \cdots, r \end{cases}$$

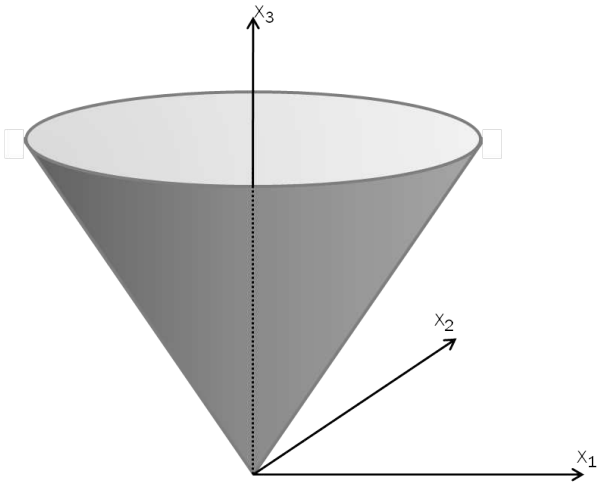

Figure 1.2: The Lorentz cone of $\mathbb{R}^3$

The second-order constraint of this problem is usually encountered in the following form :

$$\left\| -\bar{A}_i^T y + \bar{c}_i \right\| \leq \hat{A}_i y + \hat{c}_i$$

where $\begin{pmatrix} \bar{A}_i & \hat{A}_i \end{pmatrix} = A_i$ and $\begin{pmatrix} \bar{c}_i & \hat{c}_i \end{pmatrix} = c_i$.

These problems have many applications in various areas of engineering, in robust optimization, or for problems involving sums or maxima of norms. The general results on self-concordance barriers can be applied to SOCP, yielding several efficient primal-dual methods for solving such problems, which make them very useful in practice.

Least-square
A particular SOCP that is very famous in optimization is the problem of Least-Square, an uncontrained problem where the objective is to minimize the norme of a linear form :

$$\min \|Ax - b\|$$

where $A \in \mathbb{R}^{p \times n}$ and $b \in \mathbb{R}^n$. A_i are the rows of A. These problems are often used to determine the parameters of a system so as to minimize the error with respect to a set of measure. They are convex because the objective function, as the composition of a norm and a linear function, is convex.

The specificity here is that the solution of this problem can be expressed analytically, as the solution of the system $(A^T A)x = A^T b$. If the matrix A is full rank, then so is $A^T A$ and $x = (A^T A)^{-1} A^T b$. Relying on these results, some very efficient algorithms have been designed. They solve the problem in a time approximatively proportional to $n^2 p$, which can be reduced by exploiting some special structure in the matrix A. Considering these features, the resolution of least-square problems is said to be a *mature technology*, that can be used by people who do not need to know the details, with a sufficiently high level of reliability for permitting, for example, their use in embedded systems.

1.3.2 Convex Quadratically Constrained Quadratic Program (CQCQP)

A Quadratically Constrained Quadratic Program (QCQP) is an optimization defined as the minimization of a quadratic function over a feasible set defined through quadratic function :

$$\begin{cases} \min & x^T P_0 x + 2p_0^T x + \pi_0 \\ \text{s.t.} & x^T P_i x + 2p_i^T x + \pi_i \leq 0, \ i = 1, ..., m \end{cases}$$

This problem is convex and is called a Convex QCQP (CQCQP) if and only if $P_i \succcurlyeq 0, \ i = 0, ..., m$, in which case there exists efficient solvers, such as CPLEX [143], Gurobi [121] or MOSEK [11].

It turns out that a CQCQP can be expressed as a SOCP. This is straightforward thanks to the following equivalence :

$$x^T PPx + p^T x + \pi \leq 0 \quad \Leftrightarrow \left\| \begin{pmatrix} 1/2(1 + p^T x + \pi) \\ Px \end{pmatrix} \leq 1/2(1 - p^T x - \pi) \right\|$$

The converse is generally not true. Indeed, if it is easy to convert a SOCP into a QCQP, there is no reason that the latter be convex, which illustrate the difference between a convex optimization problem and an abstract convex optimization problem (see Example 1.1.3). Indeed,

$$\|Ax + b\| \leq d^T x + e \Leftrightarrow \begin{cases} x^T (A^T A - dd^T)x + 2(bA - ed^T)x + b^2 - e^2 \leq 0 \\ d^T x + e \geq 0 \end{cases}$$

Generally, the matrix $A^T A - dd^T$ is not psd, unless $d = Au$ with $\|u\| \leq 1$.

1.4 Conclusion

In this chapter, we introduce the notion of convex and conic optimization, starting by the basic definitions. Regarding convex optimization, two definitions can be found in the literature. As in the reference in convex optimization [59], we consider the more restrictive one, that states that a convex optimization problem is the minimization of a convex function on $\mathbb{R}^n$ subject to constraints of the form $f_i(x) \leq 0$ with $f_i : \mathbb{R}^n \to \mathbb{R}$ a convex function.

A less restrictive definition can sometimes be found, that consider the minimization of a convex function on a convex set. Such a problem is said to be an *abstract convex optimization problem*. Clearly a convex problem is an abstract convex problem.

Conic optimization deals with the optimization of a linear function over the intersection of an hyperplane with a proper cone. It is worth noticing than any abstract convex problem can be written as a conic program and conversely. These notions are therefore equivalent and we can use these terms interchangeably. However, the formulation as a conic program is preferable since the notion of dual cone enables very elegant formulations of the dual problem and of the Karush-Kuhn-Tucker (KKT) conditions of optimality.

Conic programs exhibit very interesting properties. First, it can be seen from the "abstract convex" definition that any local optimum is also a global optimum. As the KKT conditions for optimality are generally sufficient for local optimality, they are in this case sufficient for global optimality. Regarding duality, there exists sufficient conditions for strong duality. These conditions also guarantee the necessity of the Karush-Kuhn-Tucker conditions for global optimality.

Finally, from a complexity and resolution point of view, it was shown that they are polynomially solvable as soon as there exists a polynomially computable separation oracle for their feasible set. More practically, there exists efficient solvers based on interior-point methods for several kinds of conic programs.

In conclusion, even if convex and conic programming may appear to be quite restrictive, they are of central importance in optimization. Furthermore they subsume numerous interesting special mathematical programs. In particular, in the next chapter we focus on semidefinite programming, a special case of conic programming where the cone $\mathcal{K}$ is the cone of the positive semidefinite matrices, which can be solved efficiently with interior-point methods.

Chapter 2

Semidefinite Programming : Theory and Algorithms

This chapter provides the reader with a first look at Semidefinite Programming (SDP) and introduces the theoretical basis of this relatively young area of optimization.

SDP can be phrased as follows : it consists of the minimization or maximization of a linear function of a matrix variable X subject to linear constraints on X and to the constraint that X be positive semidefinite. As the set of n-dimensional positive semidefinite is a convex cone, semidefinite programming is clearly a special case of conic programming and therefore we can apply all the notions and results presented in Chapter 1. It can also be understood as an extension of Linear Programming (LP), where the nonnegative orthant constraint in the latter is replaced instead by the cone of positive semidefinite matrices. Similarly to LP, SDP has an elegant duality theory and presents interesting results on the geometry of the associated feasible set, the so-called *spectrahedron* and of the optimal set ,i.e., the set containing the optimal solutions. In particular, it is interesting to characterize whether the optimal set is unique.

These works led to the extension of several algorithms of LP to SDP. In LP, the fact that all the optimal solution are vertices of the feasible set gave rise to the simplex method. This result also holds for SDP, which leads to the extension of the simplex to SDP. However the most effective methods for solving a SDP, also an extension of an algorithm designed for LP, are the interior-point methods.

SDP has been one of the most developed topics in optimization during the last decades. Among this huge amount of literature, we propose the following outline. The first section supplies the fundamental definitions for addressing SDP, in particular how defining and identifying a positive semidefinite matrix, as well as the fundamental properties of the cones containing these matrices. To make this thesis self-contained, some complementary definitions are provided in Appendix 2.3, in particular background regarding symmetric matrices in Appendix 2.3.4. The second section further examines the duality theory of SDP and its consequences on the geometry of the related sets. Finally, the last section is a little guide to the different ways of solving a SDP. First, we provide theoretical results on the complexity of a SDP. Then we present the different various versions of interior-point methods for SDP, before briefly discussing some other approaches, such as bundle methods, augmented Lagrangian methods, cutting planes algorithms and simplex. Finally, we review the different available solvers for SDP.

2.1 Definitions

2.1.1 Positive definite and semidefinite matrices

Recall that the set of symmetric matrix of order n is denoted by $\mathbb{S}_n$ and when needed, this set can be regarded an Euclidean space, since it is isomorphic to $\mathbb{R}^{t(n)}$. The associated inner product is the Frobenius inner product defined at Definition 2.3.8.

Definition 2.1.1 *A symmetric matrix $A \in \mathbb{S}^n$ is positive semidefinite (psd), denoted $A \succcurlyeq 0$ if A satisfies any one of the following equivalent conditions :*

- $x^T Ax \geq 0$ *for all* $x \in \mathbb{R}^n$;
- *All its eigenvalues are nonnegative ;*
- *All the principal minors of A are non-negative ;*
- *There exists a symmetric matrix B such that $A = BB^T$.*

By requiring that B be psd, B is unique and is called *square root* of A, which is denoted by $A^{1/2}$. Furthermore, $\text{rank}(A^{1/2}) = \text{rank}(A)$.

Definition 2.1.2 *A symmetric matrix $A \in \mathbb{S}^n$ is positive definite (pd), denoted $A \succ 0$ if A satisfies any one of the following equivalent conditions :*

- $A \succcurlyeq 0$ *and A is nonsingular;*
- $x^T Ax > 0$ *for all* $x \in \mathbb{R}^n_*$;
- *All its eigenvalues are positive ;*
- *All the leading principal minors of A are positive;*
- *There exists a nonsingular symmetric matrix B such that $A = BB^T$.*

By requiring that B be positive definite, B is unique and is called *square root* of A, which is denoted by $A^{1/2}$.

As an illustration,

$$\begin{pmatrix} x & z \\ z & y \end{pmatrix} \succcurlyeq 0 \Leftrightarrow x \geq 0, y \geq 0 \text{ and } xy \geq z^2$$
$$\begin{pmatrix} x & z \\ z & y \end{pmatrix} \succ 0 \Leftrightarrow x > 0, y > 0 \text{ and } xy > z^2$$

Definition 2.1.3 *The set of positive (resp. semi)definite matrices of $\mathbb{S}^n$ is denoted by $\mathbb{S}^n_+$ (resp. $\mathbb{S}^n_{++}$).*

Definition 2.1.4 *A matrix $A \in \mathbb{S}^n$ is negative semidefinite (resp. definite), which is denoted by $A \preccurlyeq 0$ (resp. $A \prec 0$) if $-A$ is positive semidefinite (resp. definite).*

Proposition 2.1.5 *Properties of positive (semi)definite matrices*
Let $A \in \mathbb{S}^n$ be positive (resp. semi)definite, then the following properties are satisfied :

- *det*$(A) >$ *(resp.* $\geq$*)* 0 ;
- *The diagonal entries of A are positive (resp. nonnegative) ;*
- *Any principal submatrix of A is positive (resp. semi)definite ;*
- $A_{ii} = 0 \Rightarrow A_{ij} = 0$, *for all* $i, j = 1, ..., n.$;

Proposition 2.1.6 *Operations over positive (resp. semi)definite matrices*

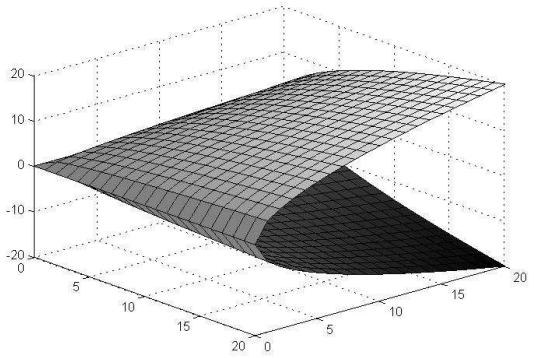

Figure 2.1: Boundary of the set of psd matrices in $\mathbb{S}^2$

- *Any nonnegative (resp. positive) combination of positive (resp. semi)definite matrices is positive (resp. semi)definite ;*
- *For* $A, B \succcurlyeq 0$, $A \otimes B \succcurlyeq 0$ *;*
- $A \oplus B \succcurlyeq 0$ *(resp.* $\succ 0$*) if and only* $A, B \succcurlyeq 0$ *(resp.* $\succ 0$*);*
- *For* $A, B \succcurlyeq 0$ *(resp.* $\succ 0$ *) such that* $AB = BA$*, then* AB *is positive (resp. semi) definite.*

Proposition 2.1.7 *Let us consider a psd matrix* $A \in \mathbb{S}^n_+$*. Then for any* $x \in \mathbb{R}^n$

$$x^T Ax = 0 \Leftrightarrow Ax = 0$$

Clearly, this result is also valid for negative semidefinite matrices.

Proposition 2.1.8 *Let us consider a set of* m *psd matrices* $A_i,\ i = 1, ..., m$*. Then* $\operatorname{rank}(\sum_{i=1}^m A_i) \geq \max_i \operatorname{rank}(A_i)$.

Proposition 2.1.9 *Let us consider a symmetric matrix with block form* $X = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$ *such that* $\operatorname{rank}(X) = \operatorname{rank}(A)$*. Then* $X \succcurlyeq 0 \Leftrightarrow A \succcurlyeq 0$.

In this case, one says that X is a *flat extension* of A.

Theorem 2.1.10 *Fejer Theorem*
A matrix $A \in \mathbb{S}^n$ *is psd if and only if* $A \bullet X \geq 0$ *for any* $X \in \mathbb{S}^n_+$.
A matrix $A \in \mathbb{S}^n$ *is positive definite if and only if* $A \bullet X > 0$ *for any nonzero* $X \in \mathbb{S}^n_+$.

Corollary 2.1.11 *A matrix* $A \in \mathbb{S}^n$ *is psd if and only if* $\begin{pmatrix} 1 \\ x \end{pmatrix}^T A \begin{pmatrix} 1 \\ x \end{pmatrix} \geq 0$ *for any* $x \in \mathbb{R}^{n-1}$.

Corollary 2.1.12 *Let* A, B *be psd matrices. Then* $A \bullet B = 0 \Leftrightarrow AB = 0$.

Corollary 2.1.13

$$A \succcurlyeq 0, B \succ 0 \quad \Rightarrow \quad A \bullet B > 0 \tag{2.1}$$

Proposition 2.1.14 *Gram matrix*

$A \in \mathbb{S}^n$ with $rank(A) = r$ is psd if and only if A arises as the Gram matrix of some collection of n-vectors $W = \{w_1, \cdots, w_n\}$ containing exactly r independent vectors, i.e.,

$$A_{ij} = w_i^T w_j, \text{ for all } i, j = 1, ..., n$$

In particular, if $A \in \mathbb{S}^n$ is positive definite, the vectors w_i have to be linearly independent ($r = n$).

To see this, it suffices to use the rows of $A^{1/2}$ as elements of W.

Proposition 2.1.15 *Cholesky factorization*

$A \in \mathbb{S}^n$ is pd if and only if there exists an unique nonsingular lower triangular matrix $L \in \mathbb{R}^{n,n}$ with positive diagonal entries such that $A = LL^T$.

$A \in \mathbb{S}^n$ is psd if and only if there exists a lower triangular matrix $L \in \mathbb{R}^{n,n}$ with such that $A = LL^T$. Such a matrix L is not unique in general.

When available, the Cholesky decomposition is a powerful tool for solving linear system, roughly twice as efficient as the LU decomposition.

Theorem 2.1.16 *Schur complement*

Let us consider a symmetric matrix M with the following block definition $M = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$. If $A \succ 0$, the following equivalences hold :

$$\begin{array}{rcl} M \succcurlyeq 0 & \Leftrightarrow & C - B^T A^{-1} B \succcurlyeq 0 \\ M \succ 0 & \Leftrightarrow & C - B^T A^{-1} B \succ 0 \end{array}$$

The matrix $C - B^T A^{-1} B$ is called Schur complement of M.

In particular, this result is used to check whether a matrix $M \in \mathbb{S}^n$ is psd in $O(n^3)$ arithmetic operations. With $n > 1$, M can be written in the form $M = \begin{pmatrix} \beta & b^T \\ b & B \end{pmatrix}$, with $\beta \in \mathbb{R}, b \in \mathbb{R}^{n-1}$ and $B \in \mathbb{S}^{n-1}$. If $\beta < 0$ or ($\beta = 0$ and $b \neq 0$), it comes that A is not psd. Otherwise,

- if $\beta = 0$ and $b = 0$, $\{M \succcurlyeq 0 \Leftrightarrow B \succcurlyeq 0\}$.
- if $\beta > 0$, $\{M \succcurlyeq 0 \Leftrightarrow \beta B - bb^T \succcurlyeq 0\}$

Thus, at each iteration, the dimension of the considered matrix decreases of one. In at most n iterations, we get a 1-dimensional matrix, which is psd if and only if its component is nonnegative.

In the view of exploiting this procedure within an optimization context, it is desirable to determine x such that $x^T M x < 0$ if M is not psd. If $\beta < 0$ or ($\beta = 0$ and $b \neq 0$), computing such an x is straightforward. Otherwise, if such an x is known for the Schur's complement : $x^T(\beta B - bb^T)x < 0$, then the augmented vector $\begin{pmatrix} \frac{b^T x}{\beta} & x \end{pmatrix}$ works for M. Finally, we conclude this topic by mentioning that this algorithm can be extended to compute the Cholesky factorization.

2.1.2 The positive semidefinite cone

Proposition 2.1.17 *$\mathbb{S}_+^n$ is a full-dimensional proper cone, called the psd cone.*

This non-polyhedral cone can be seen has the intersection of the halfspaces $H_z = \{X \in \mathbb{S}^n : z^T X z \geq 0\}$, for any $z \in \mathbb{R}^n$, and is therefore closed and convex. It is solid since the positive definite matrices comprise the cone interior, while all singular psd matrices reside on the cone boundary.

The following proposition is a direct application of the Fejer's theorem (Theorem 2.1.10).

Proposition 2.1.18 *The cone $\mathbb{S}_+^n$ is self-dual.*

Proposition 2.1.19 *The extreme rays of $\mathbb{S}_+^n$ are given by $\{\alpha uu^T : \alpha \geq 0\}$, where u is a nonzero vector in $\mathbb{R}^n$. In other words, all extreme rays of $\mathbb{S}_+^n$ are generated by rank-1 matrices.*

We provide classical results about the *facial structure* of the cone $\mathbb{S}_+^n$.

Proposition 2.1.20
Let $A \in \mathbb{S}_+^n$ a rank-r matrix, with $n > 0$. The smallest face of $\mathbb{S}_+^n$ containing A, denoted $F(A)$, has the following expression :

$$F(A) = \{X \in \mathbb{S}_+^n : \mathcal{N}(A) \subset \mathcal{N}(X)\}$$

If $A = U\Lambda U^T$ is the eigenvalue factorization of A, in virtue of Corollary 2.3.45, it comes that $\mathcal{N}(A) = \mathcal{N}(U)$ As a consequence, all the matrices of the form $F(A) = \{UVU^T : V \in \mathbb{S}_+^r\}$ and therefore, $\dim(F(A)) = \frac{r(r+1)}{2}$.

For instance, $\dim(F(A)) = 0$ if and only if $A = 0$. Furthermore, not all dimensions are represented, in particular the psd cone has no facet.

Example 2.1.21 *Consider $\mathbb{S}_+^2$. If A is a full rank matrix, then $\mathcal{N}(A) = \emptyset$ and $F(A) = \mathbb{S}_+^2$. If A is a rank-1 matrix uv^T, with $u, v \in \mathbb{R}^2$, then $\mathcal{N}(A) = \{x \in \mathbb{R}^2 : v^T x = 0\}$ and $F(A) = \{xv^T, \forall x \in \mathbb{R}^2\}$.*

Finally, by continuity of the eigenvalues, we get the following statement, which will be useful to define the notion of strict feasibility of a semidefinite program :

Proposition 2.1.22 *Interior and boundary of $\mathbb{S}_+^n$*

$$\begin{aligned}\text{bnd}(\mathbb{S}_+^n) &= \{X \in \mathbb{S}_+^n : \text{rank}(X) < n\} \\ \text{int}(\mathbb{S}_+^n) &= \{X \in \mathbb{S}_+^n : \text{rank}(X) = n\} = \mathbb{S}_{++}^n\end{aligned}$$

Theorem 2.1.23 *The set $\mathcal{S} = \{X \in \mathbb{S}^n : I \succcurlyeq X \succcurlyeq 0, \text{Tr}(X) = k\}$ for an integer $1 \leq k \leq n$, is the convex hull of the set $\mathcal{T} = \{YY^T : Y \in \mathbb{R}^{n,k}, Y^T Y = I_k\}$. Furthermore, $\mathcal{T}$ is the set of extreme points of $\mathcal{S}$.*

2.1.3 Semidefinite Programming

Semidefinite programming is the exact implementation of conic programming with the psd cone :

$$\left\{\begin{array}{lll} p^* = & \inf & A_0 \bullet X \\ & \text{s.t} & A_i \bullet X = b_i,\ i = 1, ..., m \\ & & X \succcurlyeq 0 \end{array}\right. \tag{2.2}$$

By applying duality for conic programming (see 1.1.3), with the self-duality of $\mathbb{S}_+^n$ in mind, the dual problem in the so-called *standard dual form* reads :

$$\left\{\begin{array}{lll} d^* = & \sup & b^T y \\ & \text{s.t} & A_0 - \sum_{i=1}^m A_i y_i \succcurlyeq 0 \end{array}\right. \tag{2.3}$$

The resultant constraint is called a *Linear Matrix Inequality (LMI).*

We use inf and sup instead of min and max since the infimum might not be attained. A very famous example is :

$$\begin{cases} p^* = & \inf & \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \bullet X \\ & \text{s.t} & \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \bullet X = 2 \\ & & X \succcurlyeq 0 \end{cases}$$

$X = \begin{pmatrix} x_1 & 1 \\ 1 & x_2 \end{pmatrix} \succcurlyeq 0$ if and only if $x_1 x_2 \geq 1$. Thus, $p^* = 0$ but there is no feasible solution that attains this value, since $\begin{pmatrix} 0 & 1 \\ 1 & x_2 \end{pmatrix}$ cannot be psd, in virtue of Proposition 2.1.5.

Generally, we assume that the matrices $A_i,\ i = 1, ..., m$ are linearly independent. Otherwise, either the system $A_i \bullet X = b_i,\ i = 1, ..., m$ has no solution, either it has an infinity of solution, or we can replace it by an equivalent but smaller system which involves linearly independent matrices.

It is possible to involve several variable matrices $X_k,\ k = 1, ..., l$ since it suffices to consider the whole variable $X = X_1 \oplus ... \oplus X_l$, which is psd if and only if $X_k \succcurlyeq 0,\ k = 1, ..., l$, and $\sum_{k=1}^{l} A_{i,k} \bullet X_k = \oplus_{k=1}^{l} A_{i,k} \bullet X$. Then the problem can be written as follows :

$$\begin{cases} p^* = & \inf & \sum\limits_{k=1}^{l} A_{0,k} \bullet X_k \\ & \text{s.t} & \sum\limits_{k=1}^{l} A_{i,k} \bullet X_k = b_i,\ i = 1, ..., m \\ & & X_k \succcurlyeq 0,\ k = 1, ..., l \end{cases} \tag{2.4}$$

Several primal variables X_k leads to several LMI in the dual :

$$\begin{cases} d^* = & \sup & b^T y \\ & \text{s.t} & A_{0,k} - \sum\limits_{i=1}^{m} A_{i,k} y_i \succcurlyeq 0,\ k = 1, ..., l \end{cases} \tag{2.5}$$

Thus, it is possible to use 1-dimensional variables to play the role of slack variables, which allow to consider inequality constraints instead of equality constraint in the primal. The consequences for the dual are non-positivity constraints on the corresponding variables :

$$\begin{cases} \inf & A_0 \bullet X \\ \text{s.t} & A_i \bullet X \leq b_i,\ i = 1, ..., m_i \\ & A_i \bullet X = b_i,\ i = m_i + 1, ..., m_i + m_e \\ & X \succcurlyeq 0 \end{cases} \quad \text{dual with} \quad \begin{cases} \sup & b^T y \\ \text{s.t} & A_{0,k} - \sum\limits_{i=1}^{m} A_{i,k} y_i \succcurlyeq 0,\ k = 1, ..., l \\ & y_i \leq 0,\ i = 1, ..., m_i \end{cases}$$

The notion of strict feasibility of a SDP results directly from the definition for conic programming (Definition 1.1.4) and from the fact that the interior of $\mathbb{S}_+^n$ is $\mathbb{S}_{++}^n$:

- X is a primal strictly feasible solution if X is primal feasible and $X \succ 0$;
- y is a dual strictly feasible solution if y is dual feasible and $A_0 - \sum_{i=1}^{m} A_i y_i \succ 0$;

Remark that another way of obtaining this dual is to consider the semidefinite constraint $X \succcurlyeq 0$ as an infinite number of linear constraint : $X \bullet uu^T \geq 0$, for any $u \in \mathbb{R}^n$. Then the dual of this infinite-dimensional LP involve an infinite number of variables, one for each vector u, and by denoting v_u this variable, we get the following constraint :

$$A_0 - \sum_{i=1}^{m} A_i y_i = \sum_{u \in \mathbb{R}^n} v_u u u^T \tag{2.6}$$

As $v_u \geq 0$, this means that $A_0 - \sum_{i=1}^m A_i y_i$ is a nonnegative combination of rank 1 matrices, which is equivalent to have $A_0 - \sum_{i=1}^m A_i y_i \succcurlyeq 0$ by Theorem 2.3.40.

As for any optimization problem, the weak duality $d^* \leq p^*$ holds. Conditions for strong duality are detailed in the next paragraph.

In the sequel, we generally consider the following SDP :

$$\left\{\begin{array}{lll} p^* = & \inf & A_0 \bullet X \\ & \text{s.t} & A_i \bullet X = b_i,\ i = 1, ..., m \\ & & X \succcurlyeq 0 \end{array}\right. \quad \text{dual with} \quad \left\{\begin{array}{lll} d^* = & \sup & b^T y \\ & \text{s.t} & Z = A_0 - \sum_{i=1}^m A_i y_i \\ & & Z \succcurlyeq 0 \end{array}\right. \tag{2.7}$$

where the matrix $A_i,\ i = 1, ..., m$ are linearly independent matrices, i.e., $A_i,\ i = 1, ..., m$ span an m dimensional linear space in $\mathbb{S}^n$. $\mathcal{F}$ and $\mathcal{F}^*$ denote the primal and dual feasible set respectively. Such sets, defined as the intersection of the semidefinite cone with an affine space, are called *spectrahedron*.

2.2 Duality and geometry

The dual problem can be easily formulated by applying the dual theory for conic programming. In this section, we further examine this duality theory and its consequences in terms of characterisation of the set of optimal solutions.

2.2.1 Strong duality

We consider the primal and dual formulation provided at (2.7). As for conic duality, *strong duality* does not hold in general. This is a fundamental difference with Linear Programming, illustrated on the following examples (from Lovász) :

$$\left\{\begin{array}{ll} \sup & -X_{3,3} \\ \text{s.t} & X_{1,2} + X_{2,1} + X_{3,3} = 1 \\ & X_{2,2} = 0 \\ & X \succcurlyeq 0 \end{array}\right. \quad \text{dual with} \quad \left\{\begin{array}{lll} \inf & y_1 & \\ & & \\ \text{s.t} & \begin{pmatrix} 0 & y_1 & 0 \\ y_1 & y_2 & 0 \\ 0 & 0 & y_1 + 1 \end{pmatrix} & \succcurlyeq 0 \end{array}\right.$$

The optimal value of the primal is -1 whereas the dual optimum is 0. This example is also enlightening since it illustrates the non-continuity of the optimal value w.r.t the coefficient of the problem. Indeed, if one sets the top left entry of the dual matrix to $\epsilon > 0$, then the dual optimum drops from 0 to -1

Another difference comes from the fact that there are some instances where a finite optimal value is not attained. Here is an example of this phenomenon :

$$\left\{\begin{array}{lll} \inf & y_1 & \\ & & \\ \text{s.t} & \begin{pmatrix} y_1 & 1 \\ 1 & y_2 \end{pmatrix} & \succcurlyeq 0 \end{array}\right.$$

The positive semidefiniteness is equivalent to require $y_1 \geq 0, y_2 \geq 0, y_1 y_2 \geq 1$. Having $y_1 \geq 0$ and for any $\epsilon > 0$, $y_1 = \epsilon, y_2 = 1/\epsilon$ feasible, imply that the optimal value is 0. However, $y_1 = 0$ is not feasible, therefore this optimal value is not achieved.

This was never an issue with LP : whenever the LP was feasible and its optimal value was bounded, then there was a feasible point that achieved this value.

Fortunately, Theorem 1.1.7 states that under Slater's conditions, i.e., the existence of a strictly feasible solution for the primal or/and for the dual, the strong duality holds. Another version of this theorem is given below :

Theorem 2.2.1 *Consider the primal-dual SDP pair (2.7). If both problems are feasible and if either one problem is strictly feasible, then $p^* = d^*$, the other problem attains its optimal value and for every $\epsilon > 0$, there exist feasible solutions X, y such that $C \bullet X - b^T y < \epsilon$.*

Thus, if the primal problem is strictly feasible, then the dual attains its optimum and conversely. Furthermore, if both problems are strictly feasible, then the optimal solutions are achieved in both problems.

As a direct application of KKT conditions for conic programming (see Theorem 1.1.8), when a strictly feasible solution exists for the primal, KKT conditions becomes necessary and sufficient conditions for optimality. The only difference with conic programming comes from the complementarity condition which is slightly modified. Indeed, for a primal and dual solutions X and (y, Z), its initial form is $X \bullet Z = 0$, which is equivalent to have $XZ = 0$ in virtue of Corollary 2.1.12.

Theorem 2.2.2 *KKT for Semidefinite Programming*
Consider the semidefinite problem (2.7) and assume that there exists $X \succ 0$ such that $A_i \bullet X = b_i,\ i = 1, ..., m$. Then X is an optimal solution of the problem if and only if the exists $(y, Z) \in \mathbb{R}^m \times \mathbb{S}^n$ such that :

$$\begin{cases} A_i X = b_i,\ i = 1, ..., m,\ X \succcurlyeq 0 & \textit{(primal feasibility)} \\ Z \succcurlyeq 0 & \textit{(dual feasibility)} \\ XZ = 0 & \textit{(complementary slackness)} \\ A_0 - Z + \sum_{i=1}^{m} A_i y_i = 0 & \textit{(Lagrangian stationarity)} \end{cases}$$

It is worth noticing than the Slater's condition for the dual is easily satisfied. It suffices that the primal contains a constraint of the form $I \bullet X \leq R$, which is equivalent to bounding the trace of X.

Indeed, in this case, the dual of the problem (2.7) becomes :

$$\begin{cases} \sup & b^T y - R y_0 \\ \text{s.t.} & A_0 - \sum_{i=1}^{m} y_i A_i + y_0 I \succcurlyeq 0 \\ & y_0 \geq 0 \end{cases}$$

and this problem admits a strictly feasible solution for sufficiently large value of y_0, for instance any $y_0 > -\lambda_{\min}(A^0)$. This trick is commonly used by the SDP solvers for being in the scope of strong duality.

Note that the primal strict feasibility is not as easy to recover. In particular, the presence of constraint $A_i \bullet X = 0$, with $A_i \succcurlyeq 0$, prevents the feasibility of $X \succ 0$ in virtue of Corollary 2.1.13.

We refer the reader to [120] for an excellent and detailed overview of the duality theory for SDP, following the presentation in [29]. It turns out that 11 case of duality are identified and described.

To conclude this paragraph, we mention a alternative dual problem, obtained through another kind of duality, i.e., not by Lagrangian duality, whose associated dual problem always satisfies strong duality. The main reference on the subject is the seminal paper of Ramana, Tüncel, and Wolkowicz [222].

2.2.2 Conversion of a primal standard form into a dual standard form and conversely

For a practical use of SDP, it is sometimes necessary to convert a primal form into a dual one and conversely. In theoretical terms, this conversion is simple. Indeed, the space of n-dimensional symmetric matrices is isomorphic to an Euclidean space of dimension $t(n)$ and $\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m$ such that $\mathcal{A}(X) =$

$(A_i \bullet X)_{i=1,...,m}$ is a linear operator. Since $\mathcal{F} \neq \emptyset$, we can assume that there exists $X_0 \in \mathbb{S}^n$ such that $\mathcal{A}(X_0) = b$. Then the primal and dual feasibility can be formulated as :

$$\begin{cases} X \in \mathcal{F} \Leftrightarrow X - X_0 \in \mathcal{N}(\mathcal{A}) \\ Z - A_0 \in \mathcal{R}(\mathcal{A}) \end{cases}$$

Then, it suffices to consider the linear operator $\mathcal{A}^\perp$ to invert the representation, $\mathcal{A}^\perp : \mathbb{S}^n \to \mathbb{R}^{m'}$ with $m + m' = t(n)$ with $\mathcal{A}^\perp(X) = (B_i \bullet X)_{i=1,...,m'}$ such that the matrices B_i form a basis of $\mathcal{N}(\mathcal{A})$. Then

$$\begin{cases} X \in \mathcal{F} \Leftrightarrow X - X_0 \in \mathcal{R}(\mathcal{A}^\perp) \\ Z - A_0 \in \mathcal{N}(\mathcal{A}^\perp) \end{cases}$$

In practice, $\mathcal{A}^\perp$ can be determined by solving $t(n) - m$ linear systems.

2.2.3 Geometry

The objective of a study of SDP from a geometric point of view is to characterize the set of primal and dual optimal solutions. In particular, we are interested in characterizing the uniqueness of the optimal solutions and by deriving bound on their ranks. This entails studying the facial structure of the primal and dual feasible sets and involves three fundamentals notions :

- Faces, dual faces and extreme points of $\mathcal{F}$ and $\mathcal{F}^*$;
- Nondegeneracy of a primal or a dual solution ;
- Strict complementarity.

The considered SDP is (2.7). For convenience we define the linear operator $\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m$ such that $\mathcal{A}(X) = \begin{pmatrix} A_1 \bullet X \\ \vdots \\ A_m \bullet X \end{pmatrix}$ and its adjoint : $\mathcal{A}^* : \mathbb{R}^m \to \mathbb{S}^n$, i.e., $\mathcal{A}(y) = \sum_{i=1}^m y_i A_i$. We assume that $m \leq t(n)$ and since the matrices $A_i,\ i = 1, ..., m$ are assumed to be linearly independent, we have $\text{rank}(A_i,\ i = 1, ..., m) = m$.

The main references for geometry of semidefinite programs are [10, 25, 211] but comprehensive summary can be found in the standard references [176, 259].

2.2.3.1 Nondegeneracy and strict complementarity

In the sequel, we denote by $F(x, S)$ the smallest face of the set S that contains $x \in S$. Then, in virtue of Theorem 2.2.28 and Prop. 2.1.20, we have the following characterization of $F(X, \mathcal{F})$:

Proposition 2.2.3 *Let $\mathcal{F}$ be the primal feasible set of the SDP (2.7) and $X \in \mathcal{F}$. Then $F(X, \mathcal{F}) = \{M \in \mathbb{S}^n_+ : A_i \bullet M = b_i\ i = 1, ..., m,\ \mathcal{N}(X) \subset \mathcal{N}(M)\}$.*

Recall that X is an extreme point of $\mathcal{F}$ if $F(X, \mathcal{F}) = \{X\}$. In the sequel, we call such an X a *basic solution* of the SDP.

Proposition 2.2.4 *X is a basic solution if and only if $\mathcal{N}(\mathcal{A}) \cap \text{lin}(F(X, \mathbb{S}^n_+)) = \{0\}$.*

A strongly related property that might characterize the elements of $\mathcal{F}$ is the nondegeneracy. It requires the definition of *complementary face*.

Definition 2.2.5 *Complementary face*
Let F be a face of $\mathbb{S}_+^n$. Then the complementary face $F^\triangle$ is defined as :

$$F^\triangle = \{Z \in \mathbb{S}_+^n : X \bullet Z = 0,\ \forall X \in F\} \tag{2.8}$$

Definition 2.2.6 *$X \in \mathcal{F}$ is nondegenerate if $\mathcal{R}(\mathcal{A}^*) \cap \text{lin}(F(X, \mathbb{S}_+^n)^\triangle) = \{0\}$.*

More generally, for an optimization program defined through constraints : $\min f(x) : f_i(x) \leq 0,\ i = 1, ..., m$, the degeneracy of a feasible point is defined as follows :

Definition 2.2.7 *Degeneracy*
A feasible solution $\bar{x}$ is called degenerate if all the gradients of the active constraints at $\bar{x}$ are linearly dependent.

The difficulty of applying this to SDP is that the concept of active constraint is not well-defined. However, by considering the gradient as the orthogonal complement of the level set of the constraint at $\bar{x}$, it can be extended to SDP by replacing the level set by the smallest face of $\mathbb{S}_+^n$ that contains $\bar{x}$. Then, we replace the gradient by the orthogonal complement of this face. This set is called *tangent space* and is defined as follows :

Definition 2.2.8 *Tangent space*
Let $X \in \mathbb{S}^n$, with r its rank and $X = U\Lambda U^T$ its eigenvalue factorization. Then the tangent space at X is :

$$\mathcal{T}_X = \left\{ U \begin{pmatrix} V & W \\ W^T & 0 \end{pmatrix} U^T, V \in \mathbb{S}^r, W \in \mathbb{R}^{r,n-r} \right\} \tag{2.9}$$

Definition 2.2.9 *Primal degeneracy*
Let $X \in \mathcal{F}$, with r its rank and $X = U\Lambda U^T$ its eigenvalue factorization. Then X is primal nondegenerate if $\mathcal{T}_X + \mathcal{N} = \mathbb{S}^n$, with $\mathcal{N} = \{Y \in \mathbb{S}^n : A_i \bullet Y = 0,\ i = 1, ..., m\}$.

The major advantages of nondegeneracy property is that it ensures that the optimal solution are unique.

Theorem 2.2.10 *If $X \in \mathcal{F}$ is optimal and nondegenerate, then the associated dual optimal solution (y, Z) is basic and is therefore unique.*

The converse is generally not true, except when *strict complementarity* holds. Recall that if $X \in \mathcal{F}$ and $(y, Z) \in \mathcal{F}^*$ are complementary primal and dual solutions, then $XZ = 0$. This implies that X and Z commutes (see Proposition 2.3.39) and therefore that they share a common system of eigenvectors : $X = U\Lambda U^T$ and $Z = UMU^T$, with diagonal matrices Λ, M such that $\Lambda_i M_i = 0,\ i = 1, ..., n$. As a consequence, $\text{rank}(X) + \text{rank}(Z) \leq n$.

We say that the *strict complementarity* holds when $\text{rank}(X) + \text{rank}(Z) = n$, which means that for $i = 1, ..., n$, $\Lambda_i = 0$ or (but not and) $M_i = 0$. This is also equivalent to require that $X + Z \succ 0$.

Theorem 2.2.11 *Let us consider an optimal solution $X \in \mathcal{F}$. If X admits an unique complementary solution $(y, Z) \in \mathcal{F}^*$ such that strict complementary holds, then X is nondegenerate.*

In other words, if strict complementary holds, then the primal (resp. dual) nondegeneracy is a necessary and sufficient condition for a dual (resp. primal) optimal solution to be unique.

Example 2.2.12 *Let us consider a SDP obtained as the reformulation of a SOCP (see Paragraph 3.2.4). Then the strict complementarity holds if and only if the gradient of the objective function is a strictly positive combination of the gradient of the tight constraints.*

2.2.3.2 Solutions rank

In this paragraph, we aim at characterizing the rank of the optimal solution of the SDP (2.7). First of all, a bound on the maximal rank can be simply derived from the complementarity slackness, for primal and dual optimal solution X^* and (y^*, Z^*) :

$$\text{rank}(X^*) + \text{rank}(Z^*) \leq n \tag{2.10}$$

However, we are mostly interested by deriving bound on the minimal rank of optimal solutions. Indeed, in a large number of applications (see for instance Paragraph 3.3), it is desirable to get a solution of smallest rank possible. To this end, we use the result from [25] stating that, if $A_i,\ i = 0, ..., m$ are sufficiently generic, then the optimal is attained on a basic solution, or equivalently, on a face of $\mathcal{F}$ with dimension 0. For this reason, we study the relationship between the dimension of the faces of $\mathcal{F}$ and the rank of the matrices within these faces.

Theorem 2.2.13 *Let $X \in \mathcal{F}$, with $\text{rank}(X) = r$ and such that $X = QQ^T$ with $Q \in \mathbb{R}^{n,r}$. Then,*

$$\dim(F(X, \mathcal{F})) = t(r) - \text{rank}(Q^T A_i Q,\ i = 1, ..., m)$$

where $\text{rank}(Q^T A_i Q,\ i = 1, ..., m) = \dim\{\sum_{i=1}^m y_i Q^T A_i Q\}$.

Since $\text{rank}(Q^T A_i Q,\ i = 1, ..., m) \leq m$, it comes that any face F of $\mathcal{F}$ that contains X has dimension $\dim(F) \geq t(r) - m$.

Corollary 2.2.14 *The following statements hold :*

$$X \text{ is a basic solution } \Leftrightarrow t(r) = \text{rank}(Q^T A_i Q,\ i = 1, ..., m)$$
$$X \text{ is a basic solution } \Rightarrow t(r) \leq m$$

A fundamental result from Barvinok [25] states that there exists a basic solution $X \in \mathcal{F}$ with $\text{rank}(X) \leq r$ such that $t(r+1) > m$. Let d the smallest positive integer that satisfies this inequality. It is remarkable that d is independent on n. The variation of d as a function of m is plotted on the Figure 2.2.3.2 :

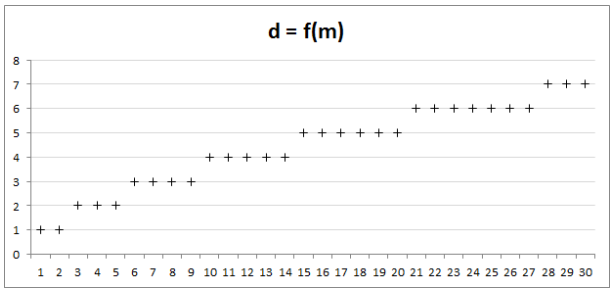

Figure 2.2: d as a function of m

In particular, we see that for $m = 2$, a rank-1 solution exists, which proves the tightness of the SDP relaxation of a QCQP with one constraint (see Paragraph 3.3).

A refinement of this result was given in [26] for the case where $\mathcal{F}$ is a nonempty and bounded set. Then if $m = t(r+1)$ for some $1 \leq r \leq n-2$, then there exists a basic solution $X \in \mathcal{F}$ with $\text{rank}(X) \leq r$.

The proof of Barvinok is not constructive but a simplex type algorithm for determining such a matrix was proposed in [210].

In conclusion, we mention that in general, finding the lowest-rank SDP solution is a NP-hard problem, whereas finding the highest-rank solution is a polynomial problem. Moreover, proving the uniqueness of a SDP solution can be done in polynomial time.

2.3 Complexity and algorithms

The wide applicability of SDP has entailed increasing demand for efficient and reliable solvers. In response to this need, a range of sophisticated algorithms have been proposed in the literature and several solvers are now available. For the main part, they are based on a primal-dual interior-point method, as an application of the breakthrough achieved by Nesterov & Nemirovski in 1988 [206]. This result, presented at Paragraph 1.2.3, states that a conic program can be solved in polynomial time by interior-point methods provided that the cone admits a barrier function with the property of self-concordance. In the case of SDP, such a function exists and is given by $-\log\det(X)$. Indeed, for $X \in \text{int}(X)$, i.e., $X \succ 0$, we have $\det(X) > 0$. When X approaches the boundary of the cone, formed by the singular positive semidefinite matrices, then $\det(X)$ tends to 0.

In this section, we begin by presenting the theoretical results on the complexity of a SDP. Then we present the different various versions of interior-point methods for SDP, before briefly discussing some other approaches, such as bundle methods, augmented Lagrangian methods, cutting planes algorithms and simplex. Finally, we review the different available solvers for SDP.

2.3.1 Complexity

Let us consider the SDP (2.7), whose matrices A_i are assumed to be of full rank. The most general result on its complexity is given by the following claim : the problem (2.7) can be solved up to any desired accuracy with interior-point algorithms that are polynomial in the RAM model.

This statement reveals two drawbacks in the complexity of a semidefinite program : the algorithms are not polynomial in the bit number model and they don't tackle the exact resolution of the problem but only an approximation.

Through this section, we address this difficult question as follows. First, we relate the above statement to the Ellipsoid method and explain why the bit model polynomiality is not guaranteed, except when some suitable conditions are provided. Finally, we discuss the problem of the exact resolution of a semidefinite program.

Application of the Ellipsoid method to a SDP
The first question that naturally arises in this context is whether their exists a weak separation oracle for SDP and whether it can be computed in polynomial time w.r.t. the input binary size of the problem L and the desired accuracy ε. For this, we assume of course that the coefficients of (2.7) are rational.

Thus, given a matrix X that satisfies the linear equalities, it suffices to check if X is "almost" psd, or find an hyperplane that "almost" separates X from $\mathbb{S}^n_+$:

$$\exists Y \in \mathbb{S}^n \ : \|Y - X\|_F \leq \varepsilon \text{ or } \exists v : v^T X v < \varepsilon \tag{2.11}$$

This can be done by means of the *outer product Cholesky factorization* methods (see Appendix 2.1.15 for the definition of the Cholesky factorization), combined with the error analysis of Higham [134]. Given a matrix $X \in \mathbb{S}^n$, this method runs in $O(n^3)$ iterations and proceeds as follows. If X is "almost" psd, then a matrix U such that $\left\|UU^T - X\right\|_F < \|X\|_F 2^{-l}$ is computed by encoding each real on l bits. If it fails, the appropriate vector v can be constructed.

The difficulty lies in the fact that the error is relative : $\|X\|_F 2^{-l}$ where we would like an absolute bound ε. Let R be an integer such that $\|X\|_F \leq R$ holds for any feasible solution of the problem. Then it suffices to take $l = \log(R/\varepsilon)$.

Then the direct application of the Ellipsoid method results in the following theorem :

Theorem 2.3.1 *Let us consider the problem (2.7) with rational coefficients of maximum bitlength L. Assume that the bowl $\{X \in \mathbb{S}^n : \|X\|_F \leq R\}$ contains the feasible set $\mathcal{F}$ of (P) and let $\varepsilon > 0$ be a*

rational number. Then there is an algorithm that runs in polynomial time w.r.t. L and $\log(R/\varepsilon)$ that produces one of the two following outputs :

- *A matrix $X^* \in \mathbb{S}^n$ satisfying the equality constraints such that*

$$\begin{array}{l} \|Y - X^*\|_F \leq \varepsilon \text{ for some } Y \in \mathcal{F} \\ A_0 \bullet X^* \geq \sup\{A_0 \bullet X : X \in \mathcal{F}_\varepsilon\} - \varepsilon \end{array} \tag{2.12}$$

- *a certificate that there is no solution $X \in \mathcal{F}_\varepsilon$.*

where $\mathcal{F}_\varepsilon = \{X \in \mathcal{F} : \begin{array}{l} A_i \bullet Y = b_i,\ i = 1, ..., m \\ \|X - Y\|_F \leq \varepsilon \end{array} \Big\} Y \succcurlyeq 0$ is the set of the ε-deep feasible solution of the problem (2.7).

From this theorem, it is clear that if R has polynomially many digits w.r.t. L, then the bit model complexity is polynomial. However, there are some pathological instances where this is not the case. See for instance (taken from [176]) the following matrices $Q_1(x) = x_1 - 2$ and $Q_i(x) = \begin{pmatrix} 1 & x_{i-1} \\ x_{i-1} & x_i \end{pmatrix}$ for $i = 2, ..., n$. Then $x_i \geq 2^{2^i - 1}$ is required for $Q_1 \oplus ... \oplus Q_n(x) \succcurlyeq 0$ and the solution has therefore an exponential bitlength.

Note that the validity of $\|X\|_F \leq R$ implies that $I \bullet X \leq R$ is also valid, which ensures that the strong validity holds as stressed at the end of the section 2.2.1. This is a key assumption for the interior-point methods described at Section 2.3.2 to work.

Exact resolution of a SDP

As explained at Appendix 3.1.2, an optimization problem can be solved into a decision problem thanks to the binary search on the optimal value. The difficulty here is that this value may have a bit size not polynomially bounded w.r.t; L (see example above) or irrational, as illustrated below :

$$\max x : \begin{pmatrix} 1 & x \\ x & 2 \end{pmatrix} \succcurlyeq 0 \rightarrow p^* = \sqrt{2} \tag{2.13}$$

Consequently, the binary search might take exponential time to reach the optimal value.

Furthermore, let us consider the feasibility problem $\exists y : A(y) = A_0 + \sum_i y_i A_i \succcurlyeq 0$. It turns out that its complexity is an open problem. Indeed, testing whether a matrix is psd is polynomial ($O(n^3)$ in the RAM model using Cholesky factorization, but it is not known whether this holds for the bit model of computation.

In [221], Ramana showed that it belongs to coNP in the RAM model and it lies either in the intersection of NP and coNP, or outside the union of NP and coNP. Finally, Porkolab and Khachiyan [152] show that this problem can be solved in $O(nm^4) + n^{O(\min\{m,n^2\})}$ arithmetic operations involving $Ln^{O(\min\{m,n^2\})}$-bit numbers.

2.3.2 Interiors-points methods

Interior-points methods for SDP have sprouted from the seminal work of Nesteror & Nemirovksi [206] who stated the theoretical basis for an extension of interior-methods to conic programming and proposed three extensions of IPM to SDP : the Karmarkar's algorithm, a projective method and Ye's potential reduction method. In parallel, in 1991, Alizadeh [7] also proposed a potential reduction projective method for SDP. Then in 1994, Boyd and Vandenberghe presented an extension of Gonzaga & Todd algorithm for LP that uses approximated search direction and able to exploit the structure of the matrix.

Subsequently, many attempts have been made to apply interior-point methods to SDP. It appears that the most widely used methods belong to the class of primal-dual path following with predictor-corrector, that leads to implementation of practically efficient solvers [53, 248, 254, 263]. Among them,

we discuss in more details the software CSDP [53], based on a method proposed in [130], since we used this solver for our numerical experiments. For the same reason, we also describe the solver DSDP [34], which implements a potential reduction algorithm for SDP.

2.3.2.1 Prerequisites

Very roughly, interior-point methods for a cone $\mathcal{K}$ are Newton-type methods applied to the minimization of $f(x)+\mu F(x)$, where f is the objective function and F a barrier function of $\mathcal{K}$. Then, at each iteration, the current solution moves along a determined search direction and μ is decreased in order to come as close as desired to the optimal solution. Two notions are fundamental to evaluate the quality of the current solution : the *optimality*, i.e., the distance to optimal solution which is measured through the *duality measure* μ and the *centrality*, i.e., the distance from the boundary of $\mathcal{K}$. Indeed, the more central is the current solution, the larger may be the size of the next step.

We apply to the particular case of SDP, where $\mathcal{K} = \mathbb{S}_+^n$. We consider the primal and dual SDP defined at 2.7, for which we assume that strong duality holds, for instance by means of Slater's constraints qualification. Then, KKT provides necessary and sufficient conditions for optimality :

$$\begin{cases} X \succcurlyeq 0,\ A_i X = b_i,\ i = 1, ..., m & \text{(primal feasibility)} \\ Z \succcurlyeq 0,\ Z = A_0 + \sum_{i=1}^{m} A_i y_i & \text{(dual feasibility)} \\ XZ = 0 & \text{(complementary slackness)} \end{cases}$$

If the complementarity condition is perturbed by the introduction of a parameter $\mu > 0$: $XZ = \mu I$, then we can show that the obtained system has an unique solution. The set of such solution when μ varies : $(X, y, Z)_{\mu \in \mathbb{R}_+}$ constitutes the central path.

As a matter of fact, with $F(X) = -\log\det(X)$, as $\frac{\partial F}{\partial X}(X) = -X^{-1}$ this perturbed system corresponds to the KKT conditions of the following problem :

$$\begin{cases} \inf & A_0 \bullet X + \mu F(X) \\ \text{s.t.} & A_i \bullet X = b_i, i = 1, ..., m \\ & X \succ 0 \end{cases}$$

Combined to the fact that this function is convex and self-concordante (see for instance [59] for a proof), we recover that F is a barrier function for $\mathbb{S}_+^n$.

$XZ = \mu I$ implies that X and Z commute and therefore they share a common basis of eigenvectors. As a consequence, $XZ = \mu I$ if and only if $\lambda_i(X)\lambda_i(Z) = \mu,\ i = 1, ..., n$ where $\lambda_i(X)$ and $\lambda_i(S)$ are the eigenvalues corresponding to the same eigenvectors.

One of the main difficulty in the implementation of interior-point methods to SDP lies in the necessity to make symmetric the feasible direction obtained by solving the Newton's system. Consider for instance the following system :

$$\begin{cases} A_i \bullet \Delta X = 0 \\ \sum_{i=1}^{m} \Delta y_i A_i + \Delta Z = 0 \\ Z\Delta X + X\Delta Z = \mu I - XZ \\ X \succcurlyeq 0, Z \succcurlyeq 0 \end{cases}$$

Clearly, the second equation imposes that ΔZ is symmetric. On the other hand, unless (X, y, Z) belongs to the central path, XZ is generally not symmetric and neither is ΔX. Imposing that ΔX be symmetric leads to a system with more equations than unknowns and therefore, there may be no feasible solution :

$A_i \bullet \Delta X = 0$	$\rightarrow$	m equations	$\Delta y \in \mathbb{R}^n$	$\rightarrow$	m variables
$\sum_i \Delta y_i A_i + \Delta Z = 0$	$\rightarrow$	$\frac{n(n+1)}{2}$ equations	$\Delta Z \in \mathbb{S}^n$	$\rightarrow$	$\frac{n(n+1)}{2}$ variables
$Z\Delta X + X\Delta Z = \tau I - XZ$	$\rightarrow$	n^2 equations	$\Delta X \in \mathbb{S}^n$	$\rightarrow$	$\frac{n(n+1)}{2}$ variables

A first possibility, called *a-posteriori symmetrization*, is to solve the system with $\Delta X \in \mathbb{R}^{n\times n}$ then to keep only the symmetric part of ΔX : $\Delta X \leftarrow \frac{\Delta X + \Delta X^T}{2}$.

Another possibility is to make symmetric the last equation, which reduces its dimension. To this end, we use a symmetrizing operator H_P paramterized by a singular matrix P : $H_P(M) = \frac{1}{2}(PMP^{-1} + (PMP^{-1})^T)$. This can be seen as the composition of the classical symmetrizing operator $M \rightarrow (1/2(M + M^T)$ and of the scaling $M \rightarrow PMP^{-1}$. Then, the last equation becomes $H_P(Z\Delta X) + H_P(X\Delta Z) = \tau I - H_P(XZ)$ and yields ΔX symmetric.

This idea was introduced by [265] and tested with various matrices P, whose most famous are :

- $P = I$: direction AHO (Alizadeh, Haeberly, Overton [9])
- $P = X^{-1/2}$ or $P = Z^{1/2}$ ([265])
- $P^T P = X^{-1}$ or Z ([198])
- $P = W^{-1/2}$ with $W = X^{1/2}(X^{1/2}ZX^{1/2})^{-1/2}X^{1/2}$, direction NT (Nesterov et Todd [207]). W is called *scale matrix*.

Currently, there is no clear consensus about the best choice for P, which remains an open question. The benefits and shortcomings of about 20 possible search directions are discussed by Todd in [254].

2.3.2.2 Primal-dual path following with predictor-corrector technique

In this paragraph, we report the algorithm proposed in [130] and implemented in the solver CSDP [53]. This primal-dual path following method uses the predictor-corrector technique of Mehrotra and has the advantage of not requiring any specific structure of the problem matrices. For the problem 2.7, the method involves the following steps :

1. Let $S = (X, y, Z)$ the incumbent solution;
2. Compute the barrier parameter τ as a function of S and deduce the corresponding barrier problem (P_τ) ;
3. Compute $\Delta S = (\Delta X, \Delta y, \Delta S)$ as the sum of the predictor $\Delta\hat{S}$ (Newton's method) and of the corrector $\Delta\bar{Z}$ (second-order method), so as to make $S + \Delta S$ the closer possible of the solution of (P_τ) ;
4. Compute α_p and α_d such that $Z = (X + \alpha_p\Delta X, y + \alpha_d\Delta y, Z + \alpha_d\Delta Z)$ be the best feasible solution w.r.t. α_p and α_d ;
5. Go back to stage 1 until the solution reaches the desired precision.

We provide some more details on the steps $2, 3, 4$.

Step 2 τ is computed as half the duality measure $\frac{Z\bullet X}{n}$. This choice is justified by good practical results obtained with this simple heuristic for LP.

Step 3 There are several possibilities for the linearisation of the optimality condition $XZ = \tau I$. For instance

$$\begin{aligned} X - \tau Z^{-1} = 0 \quad &\Leftrightarrow \tau I - Z^{1/2} X Z^{1/2} = 0 \\ &\Leftrightarrow \tau I - X^{1/2} Z X^{1/2} = 0 \\ &\Leftrightarrow ZX - \tau I = 0 \\ &\Leftrightarrow XZ - \tau I = 0 \\ &\Leftrightarrow XZ + ZX - 2\tau I = 0 \end{aligned}$$

In the present method, the chosen condition is $ZX - \tau I = 0$. It does not preserve symmetry and therefore, only the symmetric part of the obtained search direction is kept.

Step 4 α_p and α_d are computed as the solution of the following problems :

$$\left\{ \begin{array}{ll} \max & \alpha_p \\ \text{s.t.} & \mathcal{A}(X + \alpha_p \Delta X) = b \\ & X + \alpha_p \Delta X \succcurlyeq 0 \end{array} \right. \qquad \left\{ \begin{array}{ll} \max & \alpha_d \\ \text{s.t.} & \mathcal{A}^T(y + \alpha_d \Delta y) - A_0 = Z + \alpha_d \Delta Z \\ & Z + \alpha_d \Delta Z \succcurlyeq 0 \end{array} \right.$$

Results on various instances of SDP have proved the practical efficiency of this method.

2.3.2.3 Potential reduction

This method was proposed by [34] and implemented in the solver DSDP. The basic principle of the potential reduction methods is to define a *potential function* that measures the quality of the current solution and the maximize the decrease of this function at each iteration. In the case of SDP, this function is as follows : $\Phi_\rho(X, Z) = \rho \log(X \bullet Z) - \log\det(XZ)$ with $\rho > n$. The first term is the duality measure that expresses the optimality of the current solution, while $\log\det(XZ)$ is a measure of centrality. This function is used in the algorithm of DSDP. By defining the scalar $z = A_0 \bullet X$, the dual form becomes :

$$\Phi_\rho(z, y) = \rho \log(z - b^T y) - \log\det(A_0 - \sum_{i=1}^{m} y_i A_i)$$

We define the linear operator $\mathcal{A}^* : \mathbb{R}^m \to \mathbb{S}^n$ such that $\mathcal{A}^*(y) = \sum_{i=1}^m y_i A_i$ and the matrix norm : $\|M\|_\infty = \max_{i=1,\dots,n} \{|\lambda_i(A)|\} \leq \|M\|_F$. If (X_k, Z_k, y_k) is the current solution with $z_k = A_0 \bullet X_k$, then :

$$\Phi_\rho(z_k, y) - \Phi_\rho(z_k, y_k) \leq \nabla\Phi_\rho(z_k, y_k)^T (y - y_k) + \frac{\left\|(Z_k)^{-1/2} \mathcal{A}^T(y - y_k)(Z_k)^{-1/2}\right\|_F}{2(1 - \left\|(Z_k)^{-1/2} \mathcal{A}^T(y - y_k)(Z_k)^{-1/2}\right\|_\infty)}$$

Thus, if (y_k, Z_k, v_k) is the current solution, we aim at solving the following problem,where $\alpha < 1$ is a constant.

$$\left\{ \begin{array}{ll} \min & \nabla\Phi_\rho(z_k, y_k)^T (y - y_k) \\ \text{s.t.} & \left\|(Z_k)^{-1/2} \mathcal{A}^T(y - y_k)(Z_k)^{-1/2}\right\|_F \leq \alpha \end{array} \right.$$

This problem is the minimization of a linear function in an ellipsoid. Hence, the optimal solution y_{k+1} has an analytic form :

$$y_{k+1} = y_k + \beta d(z_k) \text{ with } \left\{ \begin{array}{l} d(z_k) = -(M_k)^{-1} \nabla\Phi_\rho(z_k, y_k) \\ \beta = \alpha(-\nabla\Phi_\rho(z_k, y_k) d(z_k))^{-1/2} \end{array} \right.$$

where $M_k = \{M_k\}_{i,j \in [m]}$ with $M_{k\,i,j} = A_i(Z_k)^{-1} \bullet (Z_k)^{-1} A_j$ and $\nabla\Phi_\rho(z, y) = -\frac{\rho}{z - b \bullet y} b + \mathcal{A}(Z^{-1})$.

From a computational point of view, the difficulty lies in the computation of $d(z_k)$, since it requires to compute $(M_k)^{-1}$ and $\mathcal{A}(Z_k^{-1})$. However, the authors showed that this stage can be simplified in the case when A_i are rank-1 matrices : $A_i = a_i a_i^T$ with $a_i \in \mathbb{R}^n$. Then, $M_{k\,i,j} = (a_i^T (Z_k)^{-1} a_j)^2$ and it suffices to factorize $Z_k = LL^T$, then to solve $Lw_i = a_i, \forall i = 1, \cdots, m$ to get $a_i^T (Z_k)^{-1} a_j = w_i w_j$. Other variants of the method exist, in particular to exploit the sparsity of Z.

2.3.3 Other algorithms for SDP

Interior-points methods are the most famous methods for SDP but a variety of alternative approaches have been proposed and implemented. Some of them comes from nonlinear optimization (Augmented Lagrangian), from eigenvalue optimization (Spectral Bundle), or from linear programming (Cutting Planes). Finally, the simplex method was extended from LP to conic programming and was applied specifically to SDP.

2.3.3.1 Spectral Bundle

The spectral bundle method for SDP was proposed in 2000 by Rendl and Helmberg in [129] and was implemented in the software *SBmethod*, that later became *ConicBundle*. The specificity is that it adresses a special case of SDP, where the trace of the primal matrix is equal to a known constant value : $\text{Tr}(X) = a$. This might seem a little restrictive at first sight however it includes all the problem generated as relaxation of combinatorial problem. It offers the advantage of addressing large instances of SDP.

The main idea is to cast the problem into a eigenvalue optimization problem. We consider the SDP 2.7 and we assume that there exists $\alpha \in \mathbb{R}^m$ such that $\sum_{i=1}^m A_i\alpha_i = I$, in which case the trace of any primal solution is equal to $a = \alpha^T b$. Clearly, the optimal primal solution is not $X = 0$ and, from complementarity slackness, it comes that necessarily the optimal dual solution is singular. Consequently, $Z \succcurlyeq 0$ can be replaced by $\lambda_{\max}(A_0 - \sum_{i=1}^m A_i y_i) = 0$.

Then, we built the Lagrangian using b_0 as multiplier of this constraint, which is optimal : $f(y) = b^T y + a\lambda_{\max}(A_0 - \sum_{i=1}^m A_i y_i)$. Thus, we are interested in minimizing this convex, non-smooth function, and therefore we can apply the bunde method (see Paragraph 1.2.2).

2.3.3.2 Augmented Lagrangian

An augmented Lagrangian method was implemented in the software *PENSDP* [158]. The main idea is that $Z \succcurlyeq 0 \Leftrightarrow \Phi_\rho(Z) \succcurlyeq 0$, with

$$\Phi_\rho \ : Z = U\ \text{Diag}(\lambda_1, ..., \lambda_n)\ U^T \quad \mapsto \quad U\ \text{Diag}(\rho\phi(\lambda_1/\rho), ..., \rho\phi(\lambda_n/\rho))\ U^T$$

where $\phi : \mathbb{R} \to \mathbb{R}$ is a strictly convex and strictly increasing function. Then the dual of the SDP (2.7) is equivalent to $\min b^T y : \Phi_\rho(A_0 - \sum_{i=1}^m y_i A_i) \succcurlyeq 0$, whose Lagrangian is : $L(y, U) = b^T y + B \bullet \Phi_\rho(A_0 - \sum_{i=1}^m y_i A_i)$.

Then, the algorithm consists of 3 steps :

1. $y^{k+1} = \text{argmin} L(y^k, U^k)$;
2. $U^{k+1} = D\phi(A_0 - \sum_{i=1}^m y_i^k A_i, U^k)$;
3. $\rho^{k+1} < \rho^k$.

where $D\phi(Z, U)$ is the directional derivative of ϕ at Z in the direction U. Then it remains to define the penalty function ϕ. Let just say that this function has to satisfy a number of properties and is chosen as a compromise between computational efficiency and impact on the convergence of the method.

2.3.3.3 Cutting planes algorithm

A SDP can be viewed a LP with an infinite number of constraints and turns out that a finite number of these constraints suffices to ensure the feasibility of the solution. Thus, a SDP can be solved by a sequence of LP, in the vein of a cutting planes algorithm.

More precisely,

$$\begin{aligned} X \succcurlyeq 0 \quad & \Leftrightarrow X \bullet uu^T \geq 0,\ u \in \mathbb{R}^n \\ & \Leftrightarrow X \bullet uu^T \geq 0,\ u \in \mathbb{R}^n\ : \|u\| = 1 \\ & \Leftrightarrow X \bullet u_i u_i^T \geq 0,\ i = 1, ..., k \end{aligned}$$

Then, the number of constraints is finite, but might be arbitrarily large. In the case of the solution of a dual SDP (2.7), it comes that $k \leq m$. Then the difficulty lies in identifying the collection $\{u_i\}_{i=1,...,m}$. The algorithms based on this approach are actually a direct implementation of the ellipsoid method : a linear relaxation is solved at each iteration and a separation oracle return a linear cutting planes of the incumbent solution. Then, the key of this method lies in the design of an efficient separation oracle.

This approach was investigated by Mitchell and Krishnan in [163] and an unifying framework for all approaches involving cutting planes were provided in [164].

2.3.3.4 Simplex

The extension of the simplex to conic programming relies on a thorough geometric analysis of conic programs and by the following outline of the simplex algorithm [210] : given a basic feasible solution, i.e., a feasible solution which is an extreme point of the feasible set,

- Constructs a complementary dual solution;
- If this solution is dual feasible, declares optimality;
- If not, constructs an improving extreme ray of the cone of feasible direction;
- After a linesearch in this direction, reaches a new basic solution.

Each iteration is carried out in $O(n^3)$ arithmetic operations. This method is primal and an open question is whether it could be extended to dual space, in order to perform warm-start after addition of cutting planes.

2.3.4 Solvers

Although many solvers have been developed in the last twenty years to handle semidefinite programming, this area, unlike LP, is still in its infancy, and most codes are offered by researcher to the community for free use and can handle moderate sized problems. The table 2.1 identifies the different software and their associated programming language :

Let us mention the fact that SDPA, SDSP, CSDP and SBMethod have parallel version, called SDPARA, SDSPP, Parallel CSDP and Parallel SBMethod respectively. Besides, two frontend tools are available for interfacing the problems with different solvers :

- CVX Matlab based modeling system for convex optimization, using standard Matlab expression syntax. It supports two solvers (SeDuMi and SDPT3).
- YALMIP, a free MATLAB Toolbox for rapid optimization modeling with support for typical problems. It interfaces about 20 solvers, including most famous SDP solvers.

Another simple possibility for comparing several solvers is to use the standard file format SDPA, which corresponds to the SDP (2.5), where several LMI constraints are possible. This format is accepted by most of the SDP solvers.

Software	Availability	Algorithm	Interface	Reference
CSDP	Public	IPM (Primal-Dual path)	C	[53]
DSDP	Public	IPM (Potential reduction)	C, Matlab	[34]
MOSEK	Commercial	IPM (Primal-Dual path)	Matlab	[11]
PENSDP	Commercial	Augmented Lagrangian	C, Fortran, Matlab	[158]
SeDuMi	Public	IPM (Self-dual method)	Matlab	[248]
SB	Public	Bundle method	C/C++	[129]
SDPA	Public	IPM (Primal-dual path)	C	[263]
SDPLR	Public	Augmented Lagrangian	C, Matlab	[63]
SDPNAL	Public	Augmented Lagrangian	Matlab	[266]
SDPT3	Public	IPM (Primal-dual path)	Matlab	[254]

Table 2.1: The different SDP solvers

This non-exhaustive list shows the increasing interest for semidefinite programming. However, and this is one of the most serious difficulties for using SDP, the best solver choice is very dependent on the structure of the problem (sparsity and rank of the matrices A_i, presence of a constraint $\mathrm{Tr}(X) = 1$, strict feasibility, degeneracy,...). A detailed comparison of 8 SDP solvers can be found at [197], which reveals this fickleness.

2.4 Conclusion

SDP is characterized by a powerful underlying theory based on the properties of the semidefinite matrices. These theoretical results led to the development of efficient resolution methods, in particular the interior-point methods. This is generally considered as the first reason of the interest for this optimization area. The second one is its versatility, i.e., its ability to model, embed or approximate a wide range of optimization problems. This topic is the subject of the next chapter of this thesis.

In conclusion, an interesting side effect of this line of research is that it has brought various areas of research into contact, such as numerical issues for solving large linear systems, convex analysis and all the domains concerned by its broad applicability. The next chapter presents the most famous of these applications, with a special focus on how SDP can be used to derive relaxations of NP-hard combinatorial problems or of particular instances of the Generalized Problem of Moments.

Chapter 3

Special cases and selected applications of Semidefinite Programming

A large part of the interest for SDP stems from the applicability of such problems to various areas of optimization. Some problems can be solved exactly by SDP, for instance in control and system theory where the existence of a certain semidefinite matrix is a necessary condition for the system stability, according to the theory of Lyapunov. This specific constraint is actually a Linear Matrix Inequality (LMI), which makes SDP appear as a tool tool tailored specifically for control optimization.

However, SDP has many other applications. It arises in a number of approximations algorithms for NP-hard problems, in particular for quadratic and combinatorial optimization problems. These algorithms are based on the design of tight relaxations of the problem that take the form of a SDP and are therefore solvable in polynomial time. A famous problem that admits such a so-called *semidefinite relaxation* is the Generalized Problem of Moments, a very versatile optimization problem that subsumes various problems in global optimization, related for instance to algebra, probability and statistics or financial mathematics.

In practice, it seems peculiar to need that a matrix variable X be positive semidefinite. Indeed, this notion is not very intuitive and may seem quite far from real-life constraints. In order to understand and take a global view on the processes that lead to the emergence of a semidefinite constraint, we classified them into three main mechanisms :

- by requiring that the variable have one of the properties that define semidefiniteness;
- by applying results relying on the existence of a psd matrix ;
- by requiring that the variable have a very specific structure which induces semidefiniteness.

This brings up the question of how SDP encompasses various optimization problem. Generally, this is done by converting one constraint of the problem into the requirement that a matrix defined as a linear function of the problem variables be positive semidefinite. This is the subject of the second section of this chapter.

The third section provides an overview of the use of SDP for relaxing combinatorial and quadratic problems, that are gathered in the framework of Quadratically Constrained Quadratic Programs (QCQP).

The next section is devoted to a central problem of optimization, namely the Generalized Problem of Moments (GPM) and presents how SDP can be applied to this problem. It is interesting to note the the dual of the GPM subsumes polynomial optimization (which itself subsumes combinatorial optimization), while the primal deals with the optimization of a function of the moments of a random variable, subject to various requirements of the moments of this random variable. Thus, this problem establishes a bridge between the two objectives of the thesis, namely combinatorial aspect and uncertainty in optimization. The fifth section describes other applications of SDP to optimization under uncertainty, in particular the seminal results of SDP for robust optimization.

Finally, in the sixth section, we discuss other well-known applications of SDP. We start by the most famous of them, namely the use of SDP for control theory. Then we discuss the application of SDP to the problem consisting of recovering a low rank matrix given a sampling of its entries. Then, we briefly review how SDP can be used to tackle the trust region subproblem, i.e., the minimization of a quadratic function subject to one quadratic constraint, a problem that is widely used in global optimization. We are also interested in how SDP is used for the sensor-network localization problem and we conclude this section by the recent application of SDP to data analysis problems.

3.1 Three mechanisms for identifying a semidefinite constraint

By contrast to Linear Programming, recognizing the underlying structure of a SDP is not intuitive and often requires an advanced analysis of the problem. In this section, we identify three main mechanisms that get a semidefinite constraint :

- by requiring that the variable have one of the properties that define semidefiniteness;
- by applying results relying on the existence of a psd matrix ;
- by requiring that the variable have a very specific structure which induces semidefiniteness.

3.1.1 Properties defining semidefiniteness

3.1.1.1 Nonnegative eigenvalues

One possible definition for a matrix X being psd is that all its eigenvalue are nonnegative, or equivalently $\lambda_{\min}(X) \geq 0$. As a consequence, there are some close connections between SDP and spectral optimization : a simple example is given here, where the following SDP delivers the largest eigenvalue of A : $\min t \ : tI - A \succcurlyeq 0$.

Another famous example is the maximization over x of the sum of the r largest eigenvalue of the linear combination $\sum_i A_i x_i$, which comes to solve the following SDP :

$$\begin{cases} \min & rt + I \bullet X \\ \text{s.t.} & tI + X - \sum_i A_i x_i \succcurlyeq 0 \\ & X \succcurlyeq 0 \end{cases}$$

3.1.1.2 Infinite number of constraints

The most commonly used definition of the positive semidefiniteness of a matrix $M \in \mathbb{S}^n$ is that $x^T M x \geq 0, \ \forall x \in \mathbb{R}^n$, or equivalently $xx^T \bullet M \geq 0, \ \forall x \in \mathbb{R}^n$. For fixed x, $xx^T \bullet M \geq 0$ is a linear constraint, and therefore $M \succcurlyeq 0$ can be interpreted as infinite number of linear constraints. Moreover, from Corollary 2.1.11, $M \succcurlyeq 0$ can also be used to replace the infinite number of constraints : $\tilde{x}\tilde{x}^T \bullet M \geq 0, \ \forall x \in \mathbb{R}^{n-1}$.

This process is typically used for obtaining a necessary and sufficient conditions for the nonnegativeness of a quadratic function on $\mathbb{R}^n$. Assume that z is a command variable and that the matrix $M(z) \in \mathbb{S}^n$ defines the following quadratic function : $f(z,x) = \tilde{x} M(z) \tilde{x}$. Then, $f(z,x) \geq 0, \ \forall x \in \mathbb{R}^n \Leftrightarrow M(z) \succcurlyeq 0$. In the same vein,

$$x^T M(z) x \geq \max_{k=1,...l} \{c_k^T x\}, \ \forall x \in \mathbb{R}^n \quad \Leftrightarrow \quad M(z) - \begin{pmatrix} 0 & 1/2 c_k^T \\ 1/2 c_k & 0 \end{pmatrix} \succcurlyeq 0, \ k = 1, ..., l$$

3.1.1.3 Existence of a square-root

The existence of a square root, or equivalently, of a Gram decomposition, allows to model linear combination of terms $x_i^T x_j$ where $x_0, ..., x_n$ are m-dimensional vectors and $x_0 = \begin{pmatrix} 1 & ... & 1 \end{pmatrix}$. Then, if $f(x_0, .., x_n) = \sum_{i,j=0}^{n} A_{i,j} x_i^T x_j$, then $f(x_0, .., x_n)) = b$ can be replaced by $A \bullet X = b$ with $X \succcurlyeq 0$. This comes to replace $x_i^T x_j$ by X_{ij}. This substitution is an equivalence if $\text{rank}(X) \leq m$, in particular if $m = n + 1$, otherwise it is only a relaxation.

3.1.2 Results relying on the existence of semidefinite matrix

3.1.2.1 Hessian of convex function

It is well-known that a differentiable function is convex if and only if its Hessian is everywhere psd. This applies particularly to quadratic function since their Hessian is constant.

For example, assume that we aim at approximating a function f by a convex quadratic function $\hat{f} = x^T P x + 2p^T x + \pi$ in order to minimize the distance to a certain number of noisy estimates of f : $\phi_i = f(x_i) + \epsilon_i, i = 1, ..., N$. Then we aim at minimizing $\left\| \begin{pmatrix} x_1^T P x_1 + 2p^T x_1 + \pi - \phi_1 \\ \vdots \\ x_N^T P x_N + 2p^T x_N + \pi - \phi_N \end{pmatrix} \right\|_k$ while satisfying $P \succcurlyeq 0$. For $k = 1, 2$ or $+\infty$, the resulting problem can be formulated as a SDP. This problem is known as the *convex quadratic regression problem* and is often encountered in optimization, when we aim at approaching a "black box" function by a convex quadratic function.

3.1.2.2 Schur's complement

Recognizing Schur complements in a nonlinear expression may lead to the reformulation of the expression as a LMI : let $f, g : \mathbb{R}^n \to \mathbb{R}$ and $v : \mathbb{R}^n \to \mathbb{R}^m$ some functions. Then,

$$\begin{cases} \|v(x)\|^2 \leq f(x)g(x) \\ f(x) > 0 \end{cases} \quad \Leftrightarrow \quad \begin{pmatrix} g(x) & v(x)^T \\ v(x) & f(x)I \end{pmatrix} \succcurlyeq 0$$

The right-hand term of the equivalence is a LMI whenever the functions v, f, g are linear. Remark that the requirement $f(x) > 0$ can be reduced to $f(x) \geq 0$ if $g(x) \geq 0$ holds. Indeed, $f(x) > 0$ is required since the equivalence does not hold for $f(x) = 0$ and $g(x) < 0$.

This process is widely used for reformulating problem as SDP, as discussed at Paragraph 3.2. The general form corresponds to the rational optimization problem, whereas the case $f(x) = 1$ leads to the reformulation of a QCQCP, and $f(x) = g(x)$ corresponds to a SOCP. In this latter case $f(x) \geq 0 \Rightarrow g(x) \geq 0$ and the requirement $f(x) > 0$ is therefore not necessary.

3.1.2.3 S-Lemma

S-Lemma is a special cases of the so-called *S-procedure* that aims at finding necessary and sufficient conditions for the following implication to hold : $q_j(x) \geq 0, j = 1, ..., m \Rightarrow q_0(x) \geq 0$, for some functions $q_j : \mathbb{R}^n \to \mathbb{R},\ j = 0, ..., m$.

An obvious sufficient condition for this implication to hold is the existence of $\lambda \geq 0$ such that $L(x, \lambda) = q_0(x) - \sum_{j=1}^{m} \lambda_j q_j(x) \geq 0, \forall x \in \mathbb{R}^n$. When this condition is also necessary, the S-procedure is said to be *lossless* and this happens in two important special cases. The first one is treated within the Farkas' theorem 2.3.49 and concerns the case when q_0 is convex and $q_j,\ j = 1, ..., m$ are concave.

In the special case where the functions q_j are quadratic : $q_j(x) = \tilde{x} Q_j \tilde{x}$, the condition $L(x, \lambda) \geq 0\ \forall x \in \mathbb{R}^n$ is equivalent to the LMI $Q_0 - \sum_{j=1}^{m} \lambda_j Q_j \succcurlyeq 0$. Then the S-Lemma states that this condition is necessary for $m = 1$.

Lemma 3.1.1 *S-Lemma*
Let $q_j(x) = \tilde{x}^T Q_j \tilde{x},\ j = 0, 1$ be quadratic functions such that $q_1(\bar{x}) > 0$ for some $\bar{x} \in \mathbb{R}^n$. Then

$$[q_1(x) \geq 0 \Rightarrow q_0(x) \geq 0] \Leftrightarrow [Q_0 - \lambda Q_1 \succcurlyeq 0 \text{ for some real } \lambda \geq 0]$$

The implication $q_1(x) \geq 0 \Rightarrow q_0(x) \geq 0$ is equivalent to $y^T Q_1 y \geq 0 \Rightarrow y^T Q_0 y \geq 0$. Indeed, if $y = (x_0, x)$, with $x_0 \neq 0$, then $y^T Q_j y = x_0^2 q_i(x/x_0)$. If $y = (0, x)$, the same holds by continuity. This leads to the matrix form of the S-Lemma :

Theorem 3.1.2 *S-Lemma*
Let Q_1 and Q_0 be two symmetric n-matrices and assume that $y^T Q_1 y > 0$ for some vector $y \in \mathbb{R}^n$. Then the implication $y^T Q_1 y \geq 0 \Rightarrow y^T Q_0 y \geq 0$ is valid if and only if $Q_0 - \lambda Q_1 \succcurlyeq 0$ for some real $\lambda \geq 0$.

In conclusion, let us remark that assessing the lossless of the S-procedure is worthwhile, since it means that any constraint valid over the set $\mathcal{S} = \{x \in \mathbb{R}^n : q_j(x) \geq 0,\ j = 1, ..., m\}$ dominates a positive combination of $q_j(x) \geq 0,\ j = 1, ..., m$. Consequently, if we are looking for "tight" constraint over $\mathcal{S}$, i.e., valid constraints that are not dominated by another valid constraint, it suffices to restrict the search to the positive combination of $q_j(x) \geq 0,\ j = 1, ..., m$.

There is also a close connexion with the Lagrangian duality. Indeed, consider the problem $(P)\ p^* = \min q_0(x) : x \in \mathcal{S}$. Clearly, $p^* = \max p : [x \in \mathcal{S} \Rightarrow q_0(x) - p \geq 0]$. Then the lossless of the S-procedure guarantees the strong duality, since the problem becomes equivalent to $\max p : \min_x L(x, \lambda) \geq p, \lambda \geq 0$, or equivalently $\max_{\lambda \geq 0} \min_x L(x, \lambda)$, which is exactly the Lagrangian dual of (P). Thus, the lossless of the S-procedure is equivalent to strong duality.

3.1.2.4 S.o.s polynomials

Another way of introducing a semidefinite constraint is related to the possibility of formulating a polynomial as a sum of squares of polynomials. A polynomial p of $\mathcal{P}^{n,2d}$ is said to be *sum of squares representable (s.o.s.)* if there exists r polynomials p_i of degree at most d such that $p = \sum_{i=1}^r p_i^2$.

This property is equivalent to the existence of an appropriate semidefinite matrix, as stated by the following theorem :

Theorem 3.1.3

$$p \text{ s.o.s.} \quad \Leftrightarrow \quad \exists M \succcurlyeq 0 : p(x) = p_{n,d}(x)^T M p_{n,d}(x)$$

where $M \in \mathbb{S}^{b_n(d)}$ and $p_{n,d} : \mathbb{R}^n \to \mathbb{R}^{b_n(d)}$ is a basis of $\mathcal{P}^{n,d}$.

We refer the reader to Appendix 2.5 for the notations and definitions related to polynomials. In particular, an example of basis of $\mathcal{P}^{n,d}$ is given in Appendix 2.5.3.

Relying on the fact that two polynomials are equal if and only if their coefficients are equal, the latter condition can be formulated as follows :

$$\exists M \in \mathbb{S}_+^{b_n(d)} : B^{d,\kappa} \bullet M = p_\kappa,\ \forall \kappa \in \mathbb{N}_{2d}^n$$

where the matrices $B^{d,\kappa}$ are defined at Definition 3.1.4 and p_κ is the coefficient of the polynomial p corresponding to the monomial x^κ. This typically corresponds to the feasible set of a semidefinite program.

Definition 3.1.4 $B^{d,\kappa}$
For κ in $\mathbb{N}_{2d}^n$, we define the matrices $B^{d,\kappa} \in \mathbb{S}^{b_n(d)}$ such that

$$B^{d,\kappa}_{\kappa_1,\kappa_2} = \begin{cases} 1 \text{ if } \kappa_1 + \kappa_2 = \kappa \\ 0 \text{ otherwise} \end{cases}$$

3.1.3 Particular semidefinite matrices

3.1.3.1 The matrices xx^T and $\tilde{x}\tilde{x}^T$

For a given vector $x \in \mathbb{R}^n$, the matrix xx^T, as well as its augmented form $\tilde{x}\tilde{x}^T$, are semidefinite matrices. But the converse does not hold, as claimed by the following equivalences :

$$\begin{array}{lcc} X = xx^T & \Leftrightarrow & \{X \succcurlyeq 0,\ \mathrm{rank}(X) = 1\} \\ X = \tilde{x}\tilde{x}^T & \Leftrightarrow & \{X \in \mathbb{S}^n,\ \mathrm{rank}(X) = 1,\ X_{1,1} = 1\} \\ X = \tilde{x}\tilde{x}^T & \Leftrightarrow & \{X \succcurlyeq 0,\ \mathrm{rank}(X) = 1,\ X_{1,1} = 1\} \end{array}$$

Such rank-1 matrices are frequently encountered, in particular in the representation of quadratic forms.

3.1.3.2 Laplacian matrices

We consider a weighted graph $G(V, E)$ as defined in Appendix 2.7, with $V = [n]$, $E = [m]$ and $W_{ij\,(ij)\in E}$ the weights of the edges. Then its Laplacien matrix L, defined as the n-symmetric matrix in which $L_{ij} = \sum_j W_{ij}$ if $i = j$, $-W_{ij}$ otherwise, is positive semidefinite. This follows immediately from the factorization $L = BB^T$ with $B \in \mathbb{R}^{n,m}$ indexed by V and E, such that $B_{ve} = W_e$ if v is an end of e, 0 otherwise.

3.1.3.3 Covariance matrices

The covariance matrix of a random vector (see Example 2.6.31) is necessarily psd. The converse also holds, i.e., any psd matrix is the covariance matrix of a random vector. Let us consider a random vector $X : \Omega \to \mathbb{R}^n$ of probability distribution P and mean $\mu \in \mathbb{R}^n$. Then, Σ is psd since :

$$\Sigma = \int_\Omega (X(\omega) - \mu)(X(\omega) - \mu)^T \mathrm{P}(\omega) d\omega \quad \Rightarrow \quad u^T \Sigma u = \int_\Omega (u^T (X(\omega) - \mu))^2 \mathrm{P}(\omega) d\omega \geq 0$$

3.1.3.4 Moment matrices

For a random vector $X : \Omega \to \mathbb{R}^n$, the *truncated moment vector* of order $2r$ is defined as $(y_\kappa)_{\kappa \in \mathbb{N}^n_{2r}}$ where $y_\kappa = \mathrm{E}[X^\kappa]$ is the moment of X associated to κ (see Appendix Definition 2.6.30). The components of this vector can be dispatched within a specific matrix, called *moment matrix*.

Definition 3.1.5 *Moment matrix*
The moment matrix of y is a symmetric matrix indexed by $\mathbb{N}^n_r$ and defined in the following way :

$$M_r(y) = \{M_r(y)_{\kappa_1,\kappa_2}\}_{\kappa_1,\kappa_2 \in \mathbb{N}^n_r} \text{ with } M_r(y)_{\kappa_1,\kappa_2} = y_{\kappa_1+\kappa_2}$$

Proposition 3.1.6 *For any truncated moment vector $y \in \mathbb{R}^{b_n(2r)}$, $M_r(y) \succcurlyeq 0$.*

Proof 3.1.7

$$\begin{aligned} u^T M_r(y) u &= \sum_{\kappa_1,\kappa_2} u_{\kappa_1} u_{\kappa_2} y_{\kappa_1+\kappa_2} \\ &= \sum_{\kappa_1,\kappa_2} u_{\kappa_1} u_{\kappa_2} \mathrm{E}[X^{\kappa_1+\kappa_2}] \\ &= \mathrm{E}[\sum_{\kappa_1,\kappa_2} u_{\kappa_1} u_{\kappa_2} X^{\kappa_1} X^{\kappa_2}] \\ &= \mathrm{E}[(u^T X)^2] \geq 0 \end{aligned}$$ □

The question that arises is whether the converse also holds. Indeed, we generally use this matrix in order to certify that a given y is a truncated moment vector, i.e., that there exists a random vector X such that y be the truncated moment vector of X.

Then, $M_r(y) \succcurlyeq 0$ is only a relaxation of this requirement, since $M_r(y) \succcurlyeq 0$ is necessary but generally not sufficient for y being a truncated moment vector. Combined to the condition $M_r(y)_{0,0} = 1$, the sufficiency holds for $n = 1$, which corresponds to the *Hamburger moment problem*.

Finally, we remark that the constraint $M_r(y) \succcurlyeq 0$ is a LMI since $M_r(y) = \sum_{\kappa \in \mathbb{N}^n_{2r}} B^{d,\kappa} y_\kappa$, where the matrices $B^{d,\kappa}$ are defined at Definition 3.1.4.

3.1.3.5 Localizing matrices

We extend the results of the previous paragraph by considering a random vector with a support $\mathcal{S}$: $X : \Omega \to \mathcal{S} \subset \mathbb{R}^n$.

Assume that y is the truncated moment vector of order $2r$ of X. In the same spirit as for the moment matrix, we define a matrix that involves y and and a polynomial p, which is called *localizing matrix* associated with y and p.

Definition 3.1.8 *Localizing matrix*
The localizing matrix associated with $y \in \mathbb{R}^{b_n(2r)}$ and $p \in \mathcal{P}^{n,2d}$, is the matrix indexed by $\mathbb{N}^n_{r-d}$ and defined as follows :

$$M_{r-d}(p,y)_{\kappa_1,\kappa_2} = \sum_{\kappa \in \mathbb{N}^n_{2d}} \mathsf{p}_\kappa y_{\kappa+\kappa_1+\kappa_2}$$

$M_{r-d}(p,y)$ can also be seen as the moment matrix of the vector $p*y \in \mathbb{R}^{b_n(2(r-d)}$ where $(p*y)_\kappa = \sum_{\kappa' \in \mathbb{N}^n_{2d}} \mathsf{p}_{\kappa'} y_{\kappa'+\kappa}$.

Proposition 3.1.9 *For any truncated moment vector $y \in \mathbb{R}^{b_n(2r)}$ supported on $\mathcal{S}$, for any polynomial $p \in \mathcal{P}^{n,2d}$ non-negative on $\mathcal{S}$, $M_{r-d}(p,y) \succcurlyeq 0$.*

Proof 3.1.10 *For any vector $u \in \mathbb{R}^{b_n(r-d)}$,*

$$\begin{aligned} u^T M_{r-d}(p,y) u &= \sum_{\kappa_1,\kappa_2 \in \mathbb{N}^n_{r-d}} \sum_{\kappa' \in \mathbb{N}^n_{2d}} \mathsf{p}_{\kappa'} \mathrm{E}(X^{\kappa'+\kappa_1+\kappa_2}) u_{\kappa_1} u_{\kappa_2} \\ &= \mathrm{E}\left(\sum_{\kappa_1,\kappa_2 \in \mathbb{N}^n_{r-d}} \left(\sum_{\kappa' \in \mathbb{N}^n_{2d}} \mathsf{p}_{\kappa'} X^{\kappa'} \right) X^{\kappa_1} X^{\kappa_2} u_{\kappa_1} u_{\kappa_2} \right) \\ &= \mathrm{E}\left(p(X) (u^T X)^2 \right) \geq 0 \end{aligned}$$

□

In particular, if $\mathcal{S}$ is a semi-algebraic set : $\mathcal{S} = \{x \in \mathbb{R}^n : p_i(x) \geq 0,\ i = 1,...,m\}$ where $p_i \in \mathcal{P}^{n,2d}$, $i = 1,...,m$, then $M_{r-d}(p_i,y) \succcurlyeq 0$ holds for $i = 1,...,m$.

Remark 3.1.11 *If $p(x) = 1$, $M_d(p,y) = M_d(y)$ and $M_{r-d}(p,y)_{(0,...,0),(0,...,0)} = \mathsf{p}^T y$.*

Observe that $M_{r-d}(p,y)$ can be expressed as a linear combination of y : $\sum_{\kappa \in \mathbb{N}^n_{2r}} B^{r-v,\kappa}(p)\ y_\kappa$ with the matrices $B^\kappa_{r-v}(p)$ defined as follows.

Definition 3.1.12 *Consider a polynomial $p \in \mathcal{P}^{n,2v}$. We define the matrices $B^{r-v,\kappa}(p) \in \mathbb{S}^{b_n(d)}$ for $r \geq v$ and $\kappa \in \mathbb{N}^n_d$:*

$$B^{r-v,\kappa}(p)_{\kappa_1,\kappa_2} = \begin{cases} \mathsf{p}_{\kappa-\kappa_1-\kappa_2} & \text{if } \kappa \geq \kappa_1+\kappa_2 \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } \kappa_1,\kappa_2 \in \mathbb{N}^n_{r-v}$$

In particular, if $p(x) = 1$, $B^{r,\kappa}(p) = B^{r,\kappa}$, the matrix defined at Definition 3.1.4.

3.2 Special cases of Semidefinite Programming

In this section, we show how some particular convex optimization problems can be embedded in a SDP framework. The objective here is not to solve these problems as SDP, since a tailor-made algorithm is generally more efficient, but to propose a unification framework for these problems in order to highlight an underlying hierarchy in convex optimization problems : LP $\subset$ CQCQP $\subset$ SOCP $\subset$ SDP.

3.2.1 Linear Programming

The formulation of a Linear Program in the form a SDP comes from the equivalence between the componentwise nonnegativity of a vector v and Diag(v) being psd :

$$(LP) = \left\{ \begin{array}{ll} \min & c^T x \\ \text{s.t.} & a_i^T x \leq b_i,\ i = 1, ..., m \end{array} \right. \equiv \left\{ \begin{array}{ll} \min & c^T x \\ \text{s.t.} & \text{Diag}(b) - \sum_{i=1}^{m} \text{Diag}(a_i) x_i \succcurlyeq 0 \end{array} \right.$$

3.2.2 Rational optimization

SDP can also be used to formulate some rational optimization problem, as for instance :

$$(P) = \left\{ \begin{array}{ll} \min & \frac{(c^T x)^2}{d^T x} \\ \text{s.t.} & Ax \leq b \end{array} \right.$$

and it is assumed that $Ax \leq b \Rightarrow d^T x > 0$. Then

$$(P) \equiv \left\{ \begin{array}{lll} \min & t & \\ \text{s.t.} & t \geq \frac{(c^T x)^2}{d^T x} & \\ & Ax \leq b & \end{array} \right. \equiv \left\{ \begin{array}{ll} \min & t \\ \text{s.t.} & \begin{pmatrix} t & c^T x \\ c^T x & d^T x \end{pmatrix} \succcurlyeq 0 \\ & \text{Diag}(b - Ax) \succcurlyeq 0 \end{array} \right.$$

The equivalence between these two formulations comes from the application of the Schur's theorem (Theorem 2.1.16), which is possible since $d^T x > 0$ on the feasible set.

3.2.3 Convex Quadratically Constrained Quadratic Programming

We consider the following CQCQP, whose convexity is ensured by $P_i \succcurlyeq 0,\ i = 0, ..., m$:

$$\left\{ \begin{array}{ll} \min & x^T P_0 x + 2p_0^T x + \pi_0 \\ \text{s.t.} & x^T P_i x + 2p_i^T x + \pi_i \leq 0,\ i = 1, ..., m \end{array} \right. \equiv \left\{ \begin{array}{ll} \min & t \\ \text{s.t.} & x^T P_0 x + 2p_0^T x + \pi_0 \leq t \\ & x^T P_i x + 2p_i^T x + \pi_i \leq 0,\ i = 1, ..., m \end{array} \right.$$

There are two possibilities for formulating this problem as a SDP. The first one relies on the equality : $x^T P_i x + 2p_i^T x + \pi_i = (A_i x + b_i)^T (A_i x + b_i) - c_i^T x - d_i$, where A_i is the square root of P_i, and b_i, c_i, d_i follow. Then, by applying Schur's theorem (Theorem 2.1.16), the corresponding constraint can be formulated as a LMI :

$$(A_i x + b_i)^T (A_i x + b_i) - c_i^T x - d_i \leq 0 \Leftrightarrow \begin{pmatrix} I & (A_i x + b_i) \\ (A_i x + b_i)^T & c_i^T x + d_i \end{pmatrix} \succcurlyeq 0$$

Another possibility is to apply the SDP relaxation of a QCQP (see Paragraph 3.3.2), which is exact in the convex case. Then, the problem becomes :

$$\begin{cases} \min & \begin{pmatrix} \pi_0 & p_0^T \\ p_0 & P_0 \end{pmatrix} \bullet X \\ \text{s.t.} & \begin{pmatrix} \pi_i & p_i^T \\ p_i & P_i \end{pmatrix} \bullet X \leq 0, \ i = 1, ..., m \\ & \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \bullet X = 1 \\ & X \succcurlyeq 0 \end{cases}$$

The optimal value of this problem equals the optimal value of the corresponding CQCQP. However, the difficulty lies in the fact that the optimal solution of the SDP is not necessarily a rank-1 matrix and therefore it might be difficult to recover an optimal solution of the CQCQP.

3.2.4 Second-Order Conic Programming

Let us consider the following SOCP :

$$(SOCP) \begin{cases} \min & c^T x \\ \text{s.t.} & \|Ax + b\| \leq c^T x + d \end{cases}$$

Again, the Schur's complement is used to convert the SOCP constraint into the following LMI :

$$\|Ax + b\| \leq c^T x + d \Leftrightarrow \begin{pmatrix} c^T x + d & (Ax + b)^T \\ Ax + b & (c^T x + d)I \end{pmatrix} \succcurlyeq 0$$

For the case $c^T x + d = 0$, we can not apply the Schur's complement but then the Prop. 2.1.5 ensures that $Ax + b = 0$.

3.3 SDP for combinatorial and quadratic optimization

This section provides an overview of the use of SDP for relaxing combinatorial and quadratic problems, that are gathered in the framework of Quadratically Constrained Quadratic Programs (QCQP). A QCQP is an optimization problem with a quadratic objective function and quadratic constraints :

$$(QCQP) \begin{cases} \min_{x \in \mathbb{R}^n} & x^T P_0 x + 2p_0^T x + \pi_0 \\ \text{s.t.} & x^T P_j x + 2p_j^T x + \pi_j \leq 0, j = 1, ..., m \end{cases} \tag{3.1}$$

for $P_j \in \mathbb{S}^n, p_j \in \mathbb{R}^n, \pi_j \in \mathbb{R}$ for $j = 0, ..., m$.

This problem is convex if and only if all the matrices $P_j, ij = 0, ..., m$ are psd. Otherwise it is NP-hard [141]. Indeed, it generalizes many difficult problems such as 0/1 linear programming, fractional programming, bilinear programming or polynomial programming.

This field includes all the combinatorial problems that can be written as quadratic problem with bivalent variables. Indeed, a bivalent constraint can be considered as a special case of quadratic constraints, by formulating it as following :

$$\begin{aligned} &x_i \in \{0, 1\} \Leftrightarrow x_i^2 - x_i = 0 \\ &x_i \in \{-1, 1\} \Leftrightarrow x_i^2 - 1 = 0 \\ &x_i \in \{a, b\} \Leftrightarrow (x_i - a)(x_i - b) = 0 \end{aligned}$$

This kind of combinatorial problem is very widespread since it any bounded integer variable $x \leq N$ can be written as a weighted sum of $\lfloor \log(N) \rfloor + 1$ binary variables. It also includes polynomial problems since any polynomial problems can be reduced to a quadratic problem at the expense of additional variables. Thus, the application of these problems is therefore larger that it appears at first glance, which explains why it is of primary importance in optimization.

A detailed review of this key problem is given in Appendix 3.5. We just recall that, because of the complexity, this problem is generally solved via an enumerative approach such as a Branch & Bound scheme and that obtaining a lower bound of the optimal solution is crucial for these procedures. For this, two main approaches are available : RLT (see 3.5.3), which yields a linear program, or the semidefinite relaxation described below. Those relaxations are not incompatible and can be combined together, as shown by Anstreicher in its comparison [14].

The growing interest of researchers for semidefinite relaxation can be traced back to the milestone result of Lovász [186] regarding the *Theta function* of a graph. Then a first semidefinite relaxation of QCQP was proposed by Shor in 1987 [245]. However, the real breakthrough was arguably achieved by Goemans & Williamsons [108] who opened the door on the application of the semidefinite relaxation to approximation algorithm by giving an assessment of the potential of this relaxation.

This section is structured as follows. In a first part, we give a brief overview of the fundamental results regarding semidefinite relaxation of combinatorial problems. Then, we present the standard semidefinite relaxation of QCQP and some related theoretical considerations. In the third part, we explain how the standard semidefinite relaxation can be reinforced and we expose some hierarchies of semidefinite relaxation for 0/1-LP that reach optimality.

There are a number of references on this subject within the related literature. We refer the reader to the most famous [107, 127, 176, 177, 259], with a special emphasis on [176] which is relatively recent, very complete and comprehensive.

3.3.1 Seminal works

3.3.1.1 The *theta function* of Lovász

The first use of SDP to relax difficult combinatorial problem can be attributed to Lovász in its seminal paper [186] published in 1979. This work addresses the problem of computing the stability number $\alpha(G)$ of a graph G, i.e., the size of the maximal stable set of G. We recall that a set of vertices of G is stable if none of its elements are joined by an edge of G. This problem is known to be NP-hard and is of interest in graph theory.

A natural upper bound $\alpha(G)$ is given by $\chi(G)$, the minimum cardinality of a collection of cliques C_i that together include all the nodes of G : $\min n : G \subset \cup_{i=1}^{n} C_i$. Clearly, since each node in a stable set must be in a different clique in a clique cover : $\alpha(G) \leq \chi(G)$. Moreover, for a perfect graph, $\alpha(G) = \chi(G)$.

In his paper, Lovász shows that a quantity called *theta number* $\theta(G)$, computed as the result of a SDP, is such that $\alpha(G) \leq \theta(G) \leq \chi(G)$.

$$\left\{ \begin{array}{lll} \theta(G) = & \max & ee^T \bullet X \\ & \text{s.t.} & I \bullet X = 1 \\ & & X_{ij} = 0 \text{ for } (ij) \in E \\ & & X \succcurlyeq 0 \end{array} \right. \tag{3.2}$$

To see that this problem produces an upper bound of $\alpha(G)$, it suffices to consider a maximum stable set of G of cardinality $\alpha(G)$ and its indicator vector $x \in \{0,1\}^{|V|}$. As $\alpha(G) \geq 1$ (since any single vertex is a stable set), we have $e^T x = \alpha(G) > 0$. Then we can define the matrix $X = \frac{1}{e^T x} x x^T$ which is a feasible solution of the problem (3.2) with an objective $ee^T \bullet X = e^T x = \alpha(G)$. Consequently, the optimal solution of the problem (3.2) is necessarily greater than $\alpha(G)$.

Therefore, $\theta(G)$ is an upper bound of $\alpha(G)$ that coincides with $\alpha(G)$ in the case of a perfect graph. Thus, SDP enables Lovász to develop the only known polynomial algorithm for the stability problem in a perfect graph.

3.3.1.2 A polynomial approximation for MAX-CUT by Goemans & Williamson

In 1995, another connection between SDP and graph theory was established by Goemans & Williamsons [108]. In this work, they introduced the use of SDP for approximation algorithms with a scheme that has been taken up by several authors since then.

This work applies to the problem MAX-CUT : given a graph $G = (V, E)$ with a weight w_e for each edge $e \in E$, the objective is to find a 2-partition of V such that the edges across the partition, or in the cut δ, have maximum total weight.

This problem can be formulated as a $-1/1$-QP : $\max \frac{1}{4} x^T L x : x \in \{-1, 1\}$, where L is the weighted Laplacian of the graph (see Def. 3.1.3.2). This problem is a famous NP-hard problem [151]. Moreover, any unconstrained $-1/1$-QP can be reduced to a MAX-CUT problem up to a constant K since any matrix $P \in \mathbb{S}^n$ can be written as the sum of a Laplacian matrix and of a diagonal matrix :

$$P = L + \text{Diag}((M_{ii} - \sum_{j \in [n], j \neq i} P_{ij})_{i \in [n]})$$

Consequently, for any $x \in \{-1, 1\}^n$, $x^T P x = x^T P x + K$, with $K = \sum_{i=1}^n P_{ii} - \sum_{i \neq j, i, j \in [n]} P_{ij}$. More generally, in [128], the authors show how any 0/1-QP, with possibly linear term in the objective, can be transformed into a MAX-CUT instance.

Goemans and Williamson showed that SDP yields a strong relaxation of this problem. The distinguishing feature of their work was to offer a guarantee on the quality of the obtained bound, stemming from the construction of a feasible solution via a randomized rounding procedure. Thus, by denoting p_{SDP} the optimal value of the semidefinite relaxation, p_{SOL} the expected cost of the obtained feasible solution and p_{OPT} the MAX-CUT optimal value, we have :

$$0.87856.\ p_{SDP} \leq p_{SOL} \leq p_{OPT}$$

This guarantee was outstanding since an example of graph 5-cycles was given in [83], with a ratio $p_{OPT}/p_{SDP} = 0.88445..$, which indicates that the best possible ratio for is less than this value. These values are represented on the diagram of the Figure 3.3.1.2.

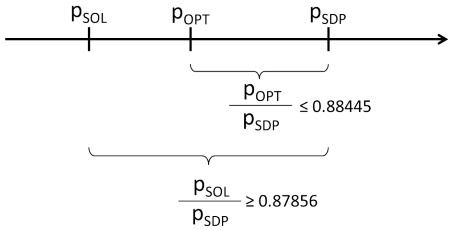

Figure 3.1: Ratios of the MAX-CUT SDP relaxation

Since then, a theoretical bound established by Håstad in 2001, stating that approximating MAX-CUT to within $16/17 = 0.941..$ is NP-hard [126]. Moreover, it has been shown by Khot et al. [155] that, if the unique games conjecture is true, then 0.87856 is the best possible approximation ratio for MAX-CUT.

In the sequel, we briefly recall how obtaining the semidefinite relaxation. We do not give much details since this is similar to the standard semidefinite relaxation scheme, described at Paragraph 3.3.2. The idea is to lift the problem to the space of symmetric matrices of size n by introducing the new variable $X = xx^T$. As $x^T Lx = L \bullet xx^T$, MAX-CUT is equivalent to :

$$\begin{cases} \max & L \bullet X \\ \text{s.t.} & X_{ii} = 1, \ i = 1, ..., n \\ & X = xx^T \end{cases}$$

Having $X = xx^T$ is strictly equivalent to requiring that $X \succcurlyeq 0$ and $\text{rank}(X) = 1$. The rank constraint being not convex, it is dropped and the semidefinite relaxation follows :

$$\begin{cases} \max & L \bullet X \\ \text{s.t.} & X_{ii} = 1, \ i = 1, ..., n \\ & X \succcurlyeq 0 \end{cases}$$

The advantage of this relaxation is that it yields not only a bound but also a value for X, which is exploited in order to derive a feasible solution via a randomized rounding procedure. To do so, a Gram representation of X is determined, i.e., a collection of vectors V such that :

$$V = \{v^{(1)}, \cdots, v^{(n)}\} \text{ such that } X_{ij} = v_i^T v_j, \forall i, j = 1, ..., n$$

Such a representation exists if and only if X is psd (see 2.1.14). Furthermore, $\forall i, X_{ii} = 1$ implies that $\|v_i\| = 1$, so v_i belongs to the unit sphere.

The next step, based on *random hyperplane technique* consists of rounding the value of v_i into $\{-1, 1\}$. For this, we draw h as a uniformly generated vector of the unit sphere, and we cut the unit sphere by the hyperplane $\{x : h^T x = 0\}$ normal to h. Then, the feasible solution is built by assigning the value 1 or -1 to x_i according to whether v_i lies on one side or the other of the hyperplane, as illustrated on figure 3.3.1.2.

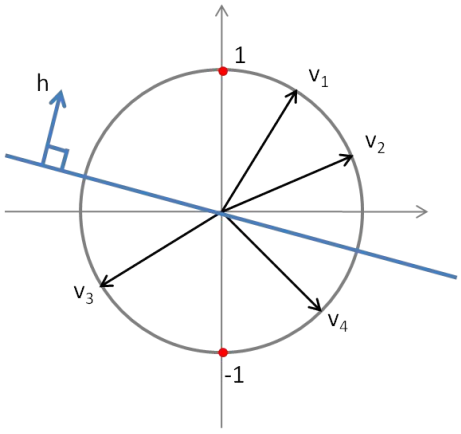

Figure 3.2: Randomized rounding procedure

Let us denote by $v_i \ \triangle_h \ v_j$ the fact that v_i and v_j be separated by h. Then, the cost of the obtained cut equals $C(h) = \sum_{(ij) \in E, v_i \triangle_h v_j} w_{ij}$. As h is uniformly distributed over the sphere, the probability of having $v_i \ \triangle_h \ v_j$ is equal to θ_{ij}/π, where θ_{ij} is the angle between the vectors v_i and v_j : $\cos(\theta_{ij}) = v_i^T v_j = X_{ij}$. Then the expected value of the cost of the obtained feasible solution is :

$$\text{E}(C(h)) = \sum_{(ij) \in E} w_{ij} \text{P}[v_i \ \triangle_h \ v_j] = \sum_{(ij) \in E} w_{ij} \frac{\theta_{ij}}{\pi}$$

Dividing it by the optimal value of the semidefinite relaxation, we get :

$$\frac{p_{SOL}}{p_{SDP}} = \frac{\sum_{(ij)\in E} w_{ij}\frac{\theta_{ij}}{\pi}}{1/4L \bullet \cos(\theta)} = \frac{\sum_{(ij)\in E} w_{ij} 2\theta_{ij}}{\sum_{(ij)\in E} w_{ij}\pi(1-\cos(\theta_{ij}))}$$

The function $\theta \mapsto \frac{2\theta}{\pi(1-\cos(\theta)}$ admits a minimal value equal to 0.87856... With nonnegative weights, $w_{ij} \geq 0$ enables to deduce the desired bound for $\frac{p_{SOL}}{p_{SDP}}$. This result was extended to graph with negative weights in [108].

In conclusion, this work has played a crucial role in the development of SDP-based applications. It has been shown subsequently [127, 175] that the SDP relaxation can be embedded in the general scheme of SDP standard relaxation of QCQP detailed at paragraph 3.3.2.

3.3.2 The standard SDP relaxation of QCQP

The standard SDP relaxation of a QCQP was introduced by Shor in [245]. Since then, it was proposed by several authors for particular cases of QCQP. In particular, this relaxation was used in the seminal work of Goemans & Williamson (see Paragraph 3.3.1.2) and forms the first rank of the Lovász-Schrijver and Lasserre hierarchies of semidefinite relaxation for 0/1-LP.

This relaxation is very simple, but has a strong theoretical basis and gives rise to a variety of possible interpretations. Its weak point is the treatment of purely linear terms and in particular, the standard SDP relaxation of a 0/1-LP turns out to be strictly equivalent to its linear relaxation. For this reason, we will see that a better way to handle linear constraints is to reformulate them as quadratic constraints, as discussed in Section 3.3.3.

3.3.2.1 Definition

Let us consider the QCQP (3.1) and define $Q_j = \begin{pmatrix} \pi_j & p_j^T \\ p_j & P_j \end{pmatrix}$. Then the standard SDP relaxation of QCQP is as follows :

$$\left\{\begin{array}{ll} \inf & Q_0 \bullet Y \\ \text{s.t.} & Q_j \bullet Y \leq 0,\ i = 1, ..., m \\ & Q_{m+1} \bullet Y = 1 \\ & Y \succcurlyeq 0 \end{array}\right. \quad \text{dual with} \quad \left\{\begin{array}{ll} \sup & y_{m+1} \\ \text{s.t.} & Q_0 - \sum_{j=1}^{m+1} y_j Q_j \succcurlyeq 0 \\ & y_j \leq 0,\ j = 1, ..., m \end{array}\right. \tag{3.3}$$

with $Q_{m+1} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$, so that $Q_{m+1} \bullet Y = Y_{1,1}$.

3.3.2.2 Interpretation

Both the primal and the dual forms can be interpreted. Regarding the primal, the key point is the reformulation of a quadratic form $\tilde{x}^T Q \tilde{x}$ into $Q \bullet \tilde{x}\tilde{x}^T$. Then the problem reads :

$$\left\{\begin{array}{ll} \inf & Q_0 \bullet Y \\ \text{s.t.} & Q_j \bullet Y \leq 0,\ j = 1, ..., m \\ & Y = \tilde{x}\tilde{x}^T \end{array}\right.$$

The constraint $Y = \tilde{x}\tilde{x}^T$ is non convex and captures all the difficulty of the problem. As stated in Paragraph 3.1.3.1, this constraint is equivalent to $Y \succcurlyeq 0, Y_{1,1} = 1$ and $\text{rank}(Y) = 1$. Thus, the standard SDP relaxation is obtained by dropping the rank-1 constraint.

Another way of viewing this relaxation is to observe that if $Y = \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix}$, then $Y \succcurlyeq 0$ is equivalent to $X \succcurlyeq xx^T$ by applying Schur's complement, where the equivalence would require $X = xx^T$.

An interpretation of the dual can be obtained by reformulating the QCQP into :

$$\left\{ \begin{array}{ll} \max & p \\ \text{s.t.} & \tilde{x}^T Q_j \tilde{x} \leq 0, \ j = 1, ..., m \end{array} \right. \Rightarrow \quad \tilde{x}^T Q_0 \tilde{x} - p \geq 0$$

Under this form, it is a direct application of the S-Lemma (see Paragraph 3.1.2.3). A sufficient condition for the constraint to hold is that there exists nonpositive scalars $y_j, \ j = 1, ..., m$ such that $Q_0 - pQ_{m+1} - \sum_{j=1}^m y_j Q_j \succcurlyeq 0$. The condition is sufficient but not necessary, which leads to a conservative approximation of the dual, and therefore to a relaxation of the primal.

This condition is necessary for $m = 1$ (according to S-Lemma 3.1.1) and for $Q_j \succcurlyeq 0$ (according to Farkas' Lemma 2.3.49), provided that a strictly primal feasible solution exists, which comes to require that strong duality holds.

In the convex case, i.e. $P_j \succcurlyeq 0, \ j = 0, ..., m$, even when strong duality does not hold, we can easily prove that the primal standard SDP relaxation is tight.

Proof 3.3.1 *Let p^* and p_S^* denote the optimal values of the QCQP and of the standard SDP relaxation respectively, and assume that $p_S^* < p^*$.*

Let $Y = \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix}$ be the optimal SDP solution, then x is feasible for the QCQP. Indeed, from Fejer's theorem (2.1.10), $X - xx^T \succcurlyeq 0$ and $P_j \succcurlyeq 0$ imply that $x^T P_j x \leq P_j \bullet X$ and therefore $x^T P_j x + 2p_j^T x + \pi_j \leq P_j \bullet X + 2p_j^T x + \pi_j \leq 0$. Using the same rational, it comes that the objective associated to x is smaller to $p_S^* < p^*$ which is a contradiction. □

Generally, if at least one matrix $P_j, \ j = 0, ..., m$ is not psd, then the standard semidefinite relaxation only provides a lower bound of the optimal solution of (3.1). It may even happen that this relaxation be unbounded, even when all the original variables have finite bounds. An example of this phenomenon is the minimization of a concave function over a bounded polyhedra. Here is this problem and its standard SDP relaxation :

$$\left\{ \begin{array}{ll} \min & x^T P_0 x + 2p_0^T x \\ \text{s.t.} & Ax \leq b \\ & 0 \leq x_i \leq 1, \ i = 1, ..., n \end{array} \right. \rightarrow \left\{ \begin{array}{ll} \inf & P_0 \bullet X + 2p_0^T x \\ \text{s.t.} & Ax \leq b \\ & 0 \leq x_i \leq 1, \ i = 1, ..., n \\ & \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succcurlyeq 0 \end{array} \right.$$

with $P_0 \preccurlyeq 0$.

If the feasible set of the original problem is not empty and contains a solution x, then (x, xx^T) is a feasible solution of the semidefinite relaxation. Moreover, for any feasible solution (x, X) of the semidefinite relaxation, (x, X') with $X' - X \succ 0$ is also feasible and, according to Fejer's theorem, $P_0 \bullet X' < P_0 \bullet X$, so the minimum value goes to negative infinity.

This example points out an important shortcoming of the standard SDP relaxation which is the treatment of the linear constraints. We will see that transforming the linear constraint into equivalent quadratic constraints is a key tool to strengthen this relaxation.

3.3.2.3 Tightness of the standard SDP relaxation

In this section, we focus on the feasibility problem associated to a QCQP : $\exists x \in \mathbb{R}^n : q(x) = 0$, where $q : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a multi-dimensional quadratic mapping such that $q_j(x) = Q_j \bullet \tilde{x}\tilde{x}^T$. The restriction

to equality constraint is not a loss of generality, since any quadratic inequality can be converted into a quadratic equality by adding the square of a slack variable : $q_j(x) \leq 0 \Leftrightarrow q_j(x) + z_j^2 = 0$.

The problem is therefore equivalent to the question whether $0 \in q(\mathbb{R}^n)$. The following theorem [220] gives a new highlight on this statement :

Theorem 3.3.2

$$0 \in \text{conv}(q(\mathbb{R}^n)) \quad \Leftrightarrow \quad 0 \in \{y \in \mathbb{R}^m : y_j = Q_j \bullet Y,\ j = 1, ..., m \text{ for some } Y \in \mathbb{S}_+^{n+1}, Y_{1,1} = 1\}$$

In particular, if $q(\mathbb{R}^n)$ is convex, the semidefinite feasibility problem is equivalent to the quadratic feasibility problem. But the question whether the image of a quadratic mapping is convex or not is NP-hard [219]. The theorem 3.3.2 was established by using the following Lemma [220]:

Lemma 3.3.3 *Let* $\phi : \mathbb{R}^n \to \mathbb{S}^n \times \mathbb{R}^n$ *such that* $\phi(x) = (xx^T, x)$. *Then* $\text{conv}(\phi(\mathbb{R}^n)) = \{(X, x) \in \mathbb{S}^n \times \mathbb{R}^n : X - xx^T \succcurlyeq 0\}$.

Going back to an arbitrary QCQP, this provides a new perspective on the standard SDP relaxation :

$$\left\{ \begin{array}{ll} \min & P_0 \bullet X + 2p_j^T x + \pi_0 \\ \text{s.t.} & P_0 \bullet X + 2p_j^T x + \pi_0 \\ & (X, x) \in \phi(\mathbb{R}^n) \end{array} \right. \rightarrow \left\{ \begin{array}{ll} \min & P_0 \bullet X + 2p_j^T x + \pi_0 \\ \text{s.t.} & P_0 \bullet X + 2p_j^T x + \pi_0 \\ & (X, x) \in \text{conv}(\phi(\mathbb{R}^n)) \end{array} \right.$$

Observe that $\text{conv}(\{x \in \mathcal{F} : Ax = b\}) \subset \{x \in \text{conv}(\mathcal{F}) : Ax = b\}$ and the inclusion is generally strict. This explains the difference between the SDP relaxation and the relaxation that would be obtained by replacing the whole feasible set by its convex hull.

3.3.2.4 Connection with Lagrangian relaxation

In this section, we recall the connection between the semidefinite and the Lagrangian relaxation of QCQP [55, 91, 98, 177]. We form the Lagrangian of the problem 3.1 by associating a non-negative variable $y_j, j = 1, ..., m$ to each constraints :

$$L(x, y) = x^T P(y)x + 2p(y)^T x + \pi(y)$$

with $P(y) = P_0 + \sum_{j=1}^m y_j P_j$, $p(y) = p_0 + \sum_{j=1}^m y_j p_j$ and $\pi(y) = \sum_{j=1}^m y_j \pi_j$. Then, the Lagrangian dual of the problem is $\sup_{y \in \mathbb{R}_+^m} \inf_{x \in \mathbb{R}^n} L(x, y)$. For fixed y, this problem deals with the minimization of the quadratic function $q_y(x) = x^T P(y)x + 2p(y)^T x + \pi(y)$.

In appendix 2.5.3, we discuss the properties of quadratic functions and in particular we have the following result : q_y admits a minimum value on $\mathbb{R}^n$ if and only if there exists a real y_{m+1} such that $\begin{pmatrix} \pi(y) - y_{m+1} & p(y)^T \\ p(y) & P(y) \end{pmatrix} \succcurlyeq 0$ and the minimal value is then larger than y_{m+1}. Hence, the dual problem is equivalent to $\sup y_{m+1}$: $\begin{pmatrix} \pi(y) - y_{m+1} & p(y)^T \\ p(y) & P(y) \end{pmatrix} \succcurlyeq 0$. This is exactly the dual form of the standard SDP relaxation (3.3).

3.3.2.5 Another interpretation of the standard SDP relaxation

This section covers the work of Fujie and Kojima in [98], who proposed an original perspective on the standard SDP relaxation. We consider the problem (3.1) and denote $\mathcal{F}$ its feasible set. The parametric notations $q(.; P, p, \pi)$ is used to denote $q : x \mapsto x^T Px + 2p^T x + \pi$ and therefore $\mathcal{F} = \{x \in \mathbb{R}^n : q(x; P_j, p_j, \pi_j) \leq 0,\ j = 1, ..., m\}$. We continue to denote by Q_j the symmetric matrices such that $q(x; P_j, p_j, \pi_j) = \tilde{x}^T Q_j \tilde{x}$.

We say that a constraint is valid for $\mathcal{F}$ if it holds for any points of $\mathcal{F}$. We denote by $\mathcal{Q}$ the set of all the convex quadratic inequalities that are valid for $\mathcal{F}$: $\mathcal{Q} = \{q(.; P, p, \pi) : P \succcurlyeq 0,\ x^T Px + 2p^T x + \pi \leq 0,\ \forall x \in \mathcal{F}\}$. Then, the convex hull of $\mathcal{F}$ is completely determined by all the convex valid inequalities (or all the linear valid inequalities) for $\mathcal{F}$, i.e.,

$$\text{conv}(\mathcal{F}) = \{x \in \mathbb{R}^n : q(x) \leq 0, \forall q \in \mathcal{Q}\}$$

The difficulty is that it is generally not possible to determine completely $\mathcal{Q}$. But there is a subset of $\mathcal{Q}$ which is very simple to determine, that comprises all the non-negative combinations of the original constraints : $\mathcal{R} = \{q(.; P, p, \pi) : q(.; P, p, \pi) = \sum_{j=1}^{m} q(.; P_j, p_j, \pi_j) \text{ for some } \lambda \in \mathbb{R}^m_+,\ P \succcurlyeq 0\} \subset \mathcal{Q}$. Then Fujie and Kojima proved in [98] that for a QCQP with a linear objective $c^T x$, the standard SDP relaxation is equivalent to $\min c^T x : q(x) \leq 0, \forall q \in \mathcal{R}$.

More precisely, with $\mathcal{F}_Q = \{x \in \mathbb{R}^n : q(x) \leq 0, \forall q \in \mathcal{R}\}$ and $\mathcal{F}_S = \{x \in \mathbb{R}^n : \exists X \succcurlyeq xx^T : Q_j \bullet \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \leq 0,\ j = 1, ..., m\}$, i.e, the projection on $\mathbb{R}^n$ of the standard SDP relaxation feasible set, then the fundamental result of [98] is that $\mathcal{F}_Q = \text{cl}(\mathcal{F}_S)$.

3.3.2.6 Application to 0/1 LP

In the particular case of a mixed 0/1 Linear Program, i.e. a problem where the only non-linear constraints are the binary constraints : $x_i^2 = x_i$, the standard semidefinite relaxation is equivalent to the continuous relaxation, i.e., the relaxation obtained by replacing $x_i \in \{0, 1\}$ by $x_i \in [0, 1]$.

$$\left\{ \begin{array}{ll} \min & a_0^T x - b_0 \\ \text{s.t.} & a_j^T x \leq b_j,\ j = 1, ..., m \\ & x_i \in \{0, 1\},\ i = 1, ..., n \end{array} \right. \quad (3.4) \qquad \left\{ \begin{array}{ll} \min & a_0^T x - b_0 \\ \text{s.t.} & a_j^T x \leq b_j,\ j = 1, ..., m \\ & x_i \in [0, 1],\ i = 1, ..., n \end{array} \right. \quad (3.5)$$

The standard SDP relaxations of the problems (3.4) and (3.5) are as follows :

$$\left\{ \begin{array}{ll} \min & Q_0 \bullet Y \\ \text{s.t.} & Q_j \bullet Y \leq 0,\ j = 1, ..., m \\ & Q_{m+1} \bullet Y = 1 \\ & D_i \bullet Y = 0,\ i = 1, ..., n \\ & Y \succcurlyeq 0 \end{array} \right. \quad (3.6) \qquad \left\{ \begin{array}{ll} \min & Q_0 \bullet Y \\ \text{s.t.} & Q_j \bullet Y \leq 0,\ j = 1, ..., m \\ & Q_{m+1} \bullet Y = 1 \\ & D_i \bullet Y \leq 0,\ i = 1, ..., n \\ & Y \succcurlyeq 0 \end{array} \right. \quad (3.7)$$

where $Q_j = \begin{pmatrix} b_j & 1/2a_j^T \\ 1/2a_j & 0 \end{pmatrix},\ j = 0, ..., m$, $Q_{m+1} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ and $D_i = \begin{pmatrix} 0 & 1/2e_i^T \\ 1/2e_i & e_i e_i^T \end{pmatrix}$.

With $Y = \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix}$, $D_i \bullet Y \leq 0$ is equivalent to $X_{ii} = x_i$. As a principal submatrix of Y, $\begin{pmatrix} 1 & x_i \\ x_i & X_{ii} \end{pmatrix} \succcurlyeq 0$ which is equivalent to $x_i = X_{ii} \leq x_i^2$ and therefore $x_i \in [0, 1]$. Consequently, the SDP relaxation is at least as tight as the continuous relaxation. We go further by proving their equivalence.

The proof follows the principle illustrated on Figure 3.3.2.6. Both the problems (3.4) and (3.5) are QCQP and therefore we can apply the standard SDP relaxation. In the case of the continuous problem (3.5), the latter is tight since the problem is convex. Then it suffices to show that both standard SDP relaxation are equivalent.

Proof 3.3.4 *Clearly, (3.7) is a relaxation of (3.6). Conversely, if Y^* is an optimal solution of the problem (3.7), we show that there exists a feasible solution of the problem (3.6) that has the same objective value.*

Let $v \in \mathbb{R}^{n+1}$ such that $v_1 = 0$, $v_{i+1} = Y^*_{1,i+1} - Y^*_{i+1,i+1}$ for $i = 1, ..., n$. Observe that $D_i \bullet Y^* \leq 0$ implies that $v \geq 0$ and therefore $Y^* + \text{Diag}(v) \succcurlyeq 0$ is a feasible solution of (3.6). The objective function

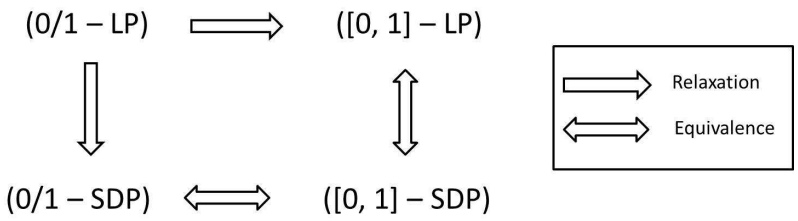

Figure 3.3: Relationship between a 0/1 LP, its linear relaxation and their semidefinite relaxations

is not impacted by adding those diagonal terms, since the objective diagonal coefficients are zero, so the optimal value of this solution equals $Q_0 \bullet Y^*$. □

3.3.3 Divers way of reinforcing the standard semidefinite relaxation

A very general manner of tightening the standard SDP relaxation is to reformulate the considered QCQP into an equivalent one, by adding valid quadratic constraints, then to apply the standard SDP relaxation on the QCQP so obtained.

Several recipes have been proposed to generate such valid constraints. Generally, they exploit particularly properties of the problem, such as linear constraints (and in particular, bounding constraints), or binary constraints. A very general recipe to generate valid quadratic constraints involves two stages. First, generate quadratic constraints by multiplying all the linear constraints together. Second, generate new valid constraints as non-negative combinations of the constraints thus obtained and of the initial quadratic constraints.

As stated by the result of Fujie and Kojima (see Paragraph 3.3.2.5), it is useless to consider the convex quadratic constraints that can be generate as a non-negative combination of the original constraints, since all these constraints are implicitly considered by the standard SDP relaxation.

Let us consider the quadratic problem obtained by adding all the pairwise product of linear constraints. From S-Lemma, we know that there may exist valid quadratic constraints that do not formulate as a non-negative combination of quadratic constraints. This is for instance the case of the well-known hypermetric constraints for 0/1 problems [128]. This motivates the investigation of other methods for generating valid constraints. In the particular case of 0/1-LP, such a method, based on the Lift & Project principle, was proposed by Lovász & Schrijver [187]. This method yields a hierarchy of semidefinite relaxations that converges to the convex hull of the feasible set in a finite number of iterations, and is presented in the second paragraph of this section. An experimentation and comparison of the different methods for reinforcing the standard SDP relaxation can be found in Section 5.

3.3.3.1 Exploiting the linear constraints

As already mentioned, there is a systematic way of exploiting the linear constraint to generate valid quadratic constraints :

- multiply all the linear constraints together ;
- make non-negative combinations of the constraints so obtained and of the initial quadratic constraints of the problem.

However, this leads to an infinite number of constraints. Among them, we identify those which was pointed out in the literature. We start by considering a linear equality constraint : $a^T x - b = 0$. Two possibilities were suggested in [91]. The first one is the square one : $(a^T x - b)^2 = 0$. Aside from

its conciseness, it offers the advantage of making useless the linear constraints. In other words, it is not only valid, but its semidefinite relaxation is sufficient for the satisfaction of $a^Tx - b = 0$. Indeed,

$$\left.\begin{array}{l} aa^T \bullet X - 2ba^Tx + b^2 = 0 \\ aa^T \bullet (X - xx^T) \geq 0 \end{array}\right\} \Rightarrow aa^T \bullet xx^T - 2ba^Tx + b^2 = (a^Tx - b)^2 \leq 0$$

On the other hand, this approach leads to a constraint $Q_i \bullet X = 0$ with $Q_i \succcurlyeq 0$, which prevents the existence of primal strictly feasible solution, as explained in 2.2.1. For this reason, the second possibility, inspired from RLT [241] (see Appendix 3.4.3.2), is preferable :

$$a^Tx = b \Leftrightarrow \left\{\begin{array}{l} a^Tx = b \\ (a^Tx - b)x_i = 0,\ i = 1, ..., n \end{array}\right.$$

Another valid but not sufficient formulation was proposed in [213] : $(a^Tx)^2 - b^2 = 0$. This constraint is included in our method since it can be viewed as $(a^Tx - b)^2 + 2b(a^Tx - b) = 0$. Having $2b \geq 0$ is not necessary since for equality constraints, any linear (and no more non-negative) combinations of constraints is valid.

Regarding inequality constraints, we start by making a key assumption, stating that the feasible set is bounded and therefore there exists $u, l \in \mathbb{R}^n$ such that the constraints $l_i \leq x_i \leq u_i,\ i = 1, ..., n$ hold. If this is not the case in the original problem, this might be derived from an educated guess about the solution. This allows to formulate any linear inequality $a^Tx \geq b$ as a range inequality $b \leq a^Tx \leq c$ and therefore we consider that all the linear inequality constraints are under this form. Then, the following quadratic formulation proposed in [127, 177, 213, 230] immediately follows :

$$b \leq a^Tx \leq c \Leftrightarrow (a^Tx - b)(a^Tx - c) \leq 0 \Leftrightarrow x^Taa^Tx - (b + c)a^Tx + bc = 0$$

Those constraints are convex since $aa^T \succcurlyeq 0$. Consequently, in the same way as for the equality square reformulation, this reformulation makes the original constraint useless in the SDP relaxation.

In particular, the bounding constraint $l_i \leq x_i \leq u_i$ becomes $(x_i - u_i)(x_i - l_i) \leq 0$. Combining these constraints with the two linear constraints leads to the following valid constraint :

$$(x_i - u_i)(x_i - l_i) + \max\{0, u_i + l_i\}(x_i - u_i) - \min\{0, u_i + l_i\}(l_i - x_i) = x_i^2 - \max\{u_i^2, l_i^2\}$$

This is exactly what is suggested in [14] to bound the diagonal of X, in order to avoid that the SDP relaxation be unbounded.

Furthermore, multiplying together the bounding constraints leads to the well-known RLT constraints [241], that form the first lifting of the three hierarchies of 0/1-LP [21, 187, 240]:

$$\begin{array}{lcl} (u_i - x_i)(u_j - x_j) \geq 0 & \Rightarrow & -X_{ij} + u_ix_j + u_jx_i - u_iu_j \leq 0 \\ (u_i - x_i)(x_j - l_j) \geq 0 & \Rightarrow & X_{ij} - u_ix_j - l_jx_i + u_il_j \leq 0 \\ (x_i - l_i)(u_j - x_j) \geq 0 & \Rightarrow & X_{ij} - l_ix_j - u_jx_i + u_jl_i \leq 0 \\ (x_i - l_i)(x_j - l_j) \geq 0 & \Rightarrow & -X_{ij} + l_ix_j + l_jx_i - l_il_j \leq 0 \end{array}$$

In the same vein, a constraint $a^Tx \geq b$ with $b \geq 0$ gives rises to the following valid constraints (see [131, 176, 178]) :

- (i) $(a^Tx)^2 - ba^Tx \geq 0$ (extended squared representation);
- (ii) $ba^Tx - b^2 \geq 0$;
- (iii) $(a^Tx)^2 - b^2 \geq 0$ (squared representation)

These constraints are embedded in our approach :

$$\begin{array}{ll} (i) & (a^Tx)^2 - ba^Tx = (a^Tx - b)^2 + b(a^Tx - b) \\ (ii) & ba^Tx - b^2 = b(a^Tx - b) \\ (iii) & (a^Tx)^2 - b^2 = (a^Tx - b)^2 + 2b(a^Tx - b) \end{array}$$

Finally, multiplying $a^T x - b \geq 0$ by the bound constraints $u_i - x_i \geq 0$ or $x_i - l_j \geq 0$ also leads to valid quadratic constraints. In particular, in the case of binary variables, we recover the lift step of the rank 1 of the Sherali-Adams hierarchy.

In conclusion, we note that the treatment of equality constraints differs significantly from the treatment of inequality constraint. Then the question that naturally arises is whether it is pertinent to convert equalities into inequalities (by duplication) and vice-versa (by means of slack variables). To the best of our knowledge, this question remains open and has not yet been thoroughly studied, neither from the practical or theoretical point of view.

3.3.3.2 The Lovász-Schrijver hierarchy of semidefinite relaxation for 0/1 Linear Programs

In [187], Lovász and Schrijver opened the door to the use of SDP to relax arbitrary 0/1-LP. More precisely, by applying a finite sequence of Lift & Project operations, their approach leads to a hierarchy of semidefinite relaxation that attains the convex hull of the feasible set within at most n steps, with n the number of binary variables. This approach admits a linear and a semidefinite variants.

We consider a polyhedron $K = \{x \in \mathbb{R}^n : Ax \leq b\}$ and the polytope $P = \text{conv}(K \cap \{0,1\}^n)$. Necessarily $P \subset K$ and the Lovász-Schrijver hierarchy (or LS-hierarchy) consists of building some sets $N_r(K)$ such that :

$$\tilde{P} \subset N_n(K) \subset ... \subset N_1(K) \subset \tilde{K}$$

where $\tilde{K}$ and $\tilde{P}$ are respectively the homogenization of K and P.

$N(K)_r$ is built as follows. First we lift the set P from $\mathbb{R}^n$ to $\mathbb{S}^{n+1}$ by introducing some product variables $\{Y_{ij}\}_{i,j=0,...,n}$ such that $Y_{0,i} = Y_{i,0} = x_i$ and $Y_{i,j} = x_i x_j$ for $i, j > 0$. $x \in P$ implies that $x_i^2 = x_i$, then $Y_{ii} = x_i = Y_{i,0}$.

These variables are gathered within a matrix $Y \in \mathbb{S}^{n+1}$ indexed by $\{0, ..., n\}$. Then $Y_{i,*} = x_i \begin{pmatrix} 1 & x \end{pmatrix}$. As $x_i \geq 0$, necessarily $Y_{i,*} \in K$. In the same way, $Y_{0,*} - Y_{i,*} = (1 - x_i) \begin{pmatrix} 1 & x \end{pmatrix} \in K$. Finally, such a matrix is psd and has rank 1. By relaxing these two latter conditions, we get the following set $M_1(K)$:

$$M_1(K) = \{Y \in \mathbb{S}^{n+1} : \text{Diag}(Y) = Y_{0,*},\ Y_{i,*} \in K,\ Y_{0,*} - Y_{i,*} \in K,\ \text{for } i = 1, ..., n\}$$

Then, $N_1(K)$ is the projection of $M_1(K)$ onto $\mathbb{R}^{n+1}$:

$$N_1(K) = \{y \in \mathbb{R}^{n+1} : y = Y_{0,*} \text{ for some } Y \in M_1(K)\}$$

$M_1^+(K)$ is a reinforcement of $M_1(K)$ obtained by requiring that the matrices be psd and $N_1^+(K)$ is its projection onto $\mathbb{R}^{n+1}$:

$$M_1^+(K) = \{Y \in \mathbb{S}_+^{n+1} : Y \in M_1(K)\}$$
$$N_1^+(K) = \{y \in \mathbb{R}^{n+1} : y = Y_{0,*} \text{ for some } Y \in M_1^+(K)\}$$

This process is applied recursively, with $M_{r+1}(K) = M_1(N_r(K))$ and similarly for the semidefinite variant. From a theoretical point of view, two results are fundamental. First, it was shown in [187] that this hierarchy attains P in less than n iterations. Furthermore, if a separation oracle is polynomially available for K, then it can be used to determine such a separation oracle for $M_1(K)$ or $M_1^+(K)$. Then, by applying a result of [119], the projections $N_1(K)$ and $N_1^+(K)$ also admit a separation oracle polynomially computable.

Thus, in theory it is possible to optimize over any $N_r^+(K)$ or $N_r(K)$ by using the ellipsoid method, but this is not feasible in practice since this method is computationally prohibitive. Consequently, we would like to compute a polyhedral description of $N_r(K)$ or $N_r^+(K)$ in order to apply any method of linear programming to optimize over it. The difficulty is that such a description is not easy to compute

and may require a huge number of constraints. Furthermore, to determine the set $N_r^+(K)$, it is necessary to determine all the previous set $N_{r'}^+(K)$, for $t' = 1, ..., t-1$.

For this reason, these relaxation are said to be *implicit*. Instead of providing an explicit description of the convex hull, there are rather used to provide a valid constraint , violated by an incumbent solution, in order to reinforce the linear relaxation K.

We remark that the set $M_1^+(K)$ could be obtained by applying the standard SDP relaxation to the QCQP obtained by multiplying all the linear constraints by x_i and $1 - x_i$ and we conclude this paragraph by noticing that there exists three other hierarchies of linear relaxation for 0/1-LP, based on integer rounding ([111], see Appendix 3.4.2.1) or on Lift & Project ([21, 240], see Appendix 3.4.3). Regarding SDP hierarchies, apart the Lovász-Schrijver one, one finds only the Lasserre hierarchy, presented in another paragraph 3.4.3 due to its closeness to the Generalized Moment Problem. It was proved in [173] that the Lasserre's hierarchy can be viewed as a generalization of the Lovász-Schrijver semidefinite hierarchy.

3.3.3.3 Cutting planes

In this section, we discuss some cutting planes that have been proposed to strengthen the standard semidefinite relaxation. To a large extent, such works concern a more restrictive part of QCQP, mainly MIQCQP, where the cuts exploits the fact the variables are integer. In the interest of concision, we restrict ourselves to the cuts for *pure* MIQCQP, i.e., not MILP. We just mention that a large number of cutting planes for MILP can be generated by applying the Lift & Project method, in particular by 3 hierarchies [21, 187, 240] that yields the convex hull of the feasible set in a finite number of iterations .

Another classical way to generate cutting planes for semidefinite relaxation comes from linear disjunctions based on the integrity of the variables. Having $a^T x \leq b$ **or** $c^T x \leq d$ such that those both constraints are incompatible can be expressed exactly by the quadratic constraint $(a^T x - b)(c^T x - d) \leq 0$.

In [84], Deza and Laurent introduced an automatic method to generate such valid disjunctions by exploiting integrity. For any integer vector $b \in \mathbb{Z}^n$ such that $b^T e$ is odd, $2b^T x \leq b^T e - 1$ or $2b^T x \geq b^T e + 1$. These cuts, called *hypermetric inequality* are applied to semidefinite relaxation in [128]. The most famous of them are the so-called *triangle inequalities*, obtained for any indices $i \neq j \neq k$ by picking successively $b = -e_i - e_j - e_k$, $b = -e_i + e_j + e_k$, $b = e_i - e_j + e_k$ and $b = e_i + e_j - e_k$:

$$
\begin{array}{l}
(i)\ x_i + x_j + x_k \leq x_{ij} + x_{ik} + x_{jk} + 1 \\
(ii)\ x_{ik} + x_{jk} \leq x_k + x_{ij} \\
(iii)\ x_{ij} + x_{ik} \leq x_i + x_{jk} \\
(iv)\ x_{ij} + x_{jk} \leq x_j + x_{ik}
\end{array}
$$

Another contribution in this vein was made in [131]. The constraint $a^T x - b \geq 0$, with a and b integer leads to the valid disjunction $a^T x - b \leq 0$ or $a^T x - b \geq 1$, i.e., $(a^T x - b)(a^T x - b - 1) \leq 0$.

Some other disjunctions can be used to generate valid constraints. In [146], the authors discussed the generation of valid quadratic cuts for 0/1 convex QCQP, i.e., a special case of QCQP where the non-convexity is due exclusively to the binary constraints. Then, the generation of the cut follows the well-known principle of a cutting plane algorithm [21], where a separation problem is solved at each iteration in order to determine a cut that is both valid and violated by the current relaxed solution. The relaxation solved at each iteration is a convex QCQP, and the cut generation is based on disjunctive programming.

In [235], the authors proposed valid disjunctions based on the constraint $A \bullet (X - xx^T) \leq 0$ that holds for any matrix A whenever $X - xx^T = 0$ is valid. By picking $A \succcurlyeq 0$, such a constraint may improve the semidefinite relaxation, since the latter implies that $A \bullet (X - xx^T) \geq 0$. The difficulty is that the quadratic term $x^T Ax$ do not appear in the semidefinite relaxation. To overcome this difficulty, this term is replaced by a valid linear disjunction, for a rank 1 matrix : $A = cc^T$. The vector c is

built as the best positive combination of eigenvectors of the incumbent solution $X - xx^T$. Remark that the disjunction generated here is not exclusive, which means that both parts of the disjunction may be satisfied. In this case, multiplying the linear constraints to get a valid quadratic constraints in not possible. Instead, a valid linear constraint is derived by applying Balas' technique [18]. Finally, another work from the same authors [236] uses a semidefinite program to compute valid quadratic convex cuts, by getting rid of the lifted variables (projection).

3.3.4 Using SDP to convexify a Mixed-Integer QCQP

A recent approach to deal with MIQCQP is to use SDP to convexify the quadratic functions. Thus the SDP is not used to relax the problem but to reformulate it.

The first work in this sense was carried out by Hammer & Rubin in 1970 [124], for a 0/1-QP with linear constraints. Let $f(x) = x^T Px + 2p^T x + \pi$ be the objective to minimize and define v the vector with every components equal to $\lambda_{\min}(P)$. Since the variables are binary : $x_i^2 = x_i$ and it follows that $f'(x) = x^T(P - \text{Diag}(v))x + 2(p+v)^T x + \pi$ is convex and reaches the same value that f over $\{0,1\}^n$.

This basic principle inspired Billionnet [49] which developed it in a more sophisticated way. The idea is still to perturb the objective function but the semidefiniteness of the matrix P' so obtained is ensured by a SDP, devised in order to keep the same objective values on $\{0,1\}^n$ and to maximize the tightness of the continuous relaxation of the problem. The motivation behind this method is to cast the problem into a 0/1-CQCQP for which efficient solvers are available.

This method, called *Convex Quadratic Reformulation*, has since then been extended to larger classes of problem. The problem addressed in [49] was a 0/1-QP with equality linear constraint and it was extended by the same authors [46] to general MIQP with equality and inequality constraints. Finally Letchford & Galli [179] developed the method for the very general case of MIQCQP.

3.4 Semidefinite relaxations of the Generalized Problem of Moments

3.4.1 Introduction

3.4.1.1 Definition of the Generalized Problem of Moments

The Generalized Problem of Moments (GPM) is an optimization problem where the optimization variable is not a vector of an Euclidean space as usual, but a non-negative measure P on $\mathcal{B}(\mathcal{S})$, the Borel σ-algebra on $\mathcal{S} \subset \mathbb{R}^n$:

$$\begin{cases} \min & \int_{\mathcal{S}} f_0(\omega) P(\omega) d\omega \\ \text{s.t.} & \int_{\mathcal{S}} f_i(\omega) P(\omega) d\omega = b_i, \ i = 1, ..., m \\ & \mathrm{P} \in \mathcal{M}(\mathcal{S}) \end{cases}$$

where $\mathcal{M}(\mathcal{S})$ denotes the set of non negative measures over $\mathcal{B}(\mathcal{S})$ (see Def. 2.6.3) and $f_i : \mathcal{S} \to \mathbb{R}, \ i = 0, ..., m$ are measurable functions. By considering each value $\mathrm{P}(\omega), \ \omega \in \mathcal{S}$ as a variable, this problem can be seen as a linear program with an infinite number of variable, or equivalently a *semi-infinite* linear program.

As such, this problem is intractable and is essentially used as a theoretical modelling tool. However, there is a slight restriction of this problem that admits a hierarchy of SDP relaxation, i.e., a sequence of SDP whose optimal values approaches the optimal value of the GPM as closely as desired, although the problem remains NP-hard. This restriction consists of assuming that P is a probability measure (or equivalently, that $\int_{\mathcal{S}} P(\omega) d\omega$ is bounded), that the functions $f_i, \ i = 0, ..., m$ are polynomials and $\mathcal{S}$ is a semi-algebraic set (see Def. 2.5.5).

This is the origin of the name of the problem, since in this case, the problem involves linear combination of the moments y_κ (see Def. 2.6.30) of the random variable associated to the probability measure P :

$$\left\{\begin{array}{ll} \min & \mathrm{E_P}\left(\sum\limits_{\kappa\in\mathbb{N}^n_d} \mathsf{f}_{0\kappa}\omega^\kappa\right) \\ \text{s.t.} & \mathrm{E_P}\left(\sum\limits_{\kappa\in\mathbb{N}^n_d} \mathsf{f}_{i\kappa}\omega^\kappa\right) = b_i,\ i=1,...,m \\ & \mathrm{P}(\mathcal{S}) = 1 \\ & \mathrm{P}\in\mathcal{M}(\mathcal{S}) \end{array}\right. \equiv \left\{\begin{array}{ll} \min & \sum\limits_{\kappa\in\mathbb{N}^n_d} \mathsf{f}_{0\kappa} y_\kappa \\ \text{s.t.} & \sum\limits_{\kappa\in\mathbb{N}^n_d} \mathsf{f}_{i\kappa} y_\kappa = b_i,\ i=1,...,m \\ & y_\kappa = \mathrm{E_P}\left(\omega^\kappa\right),\ \kappa\in\mathbb{N}^n_d \\ & y_{(0,...,0)} = 1 \\ & \mathrm{P}\in\mathcal{M}(\mathcal{S}) \end{array}\right. \tag{3.8}$$

where f_i are of d-degree polynomials such that $f_i(\omega) = \sum_{\kappa\in\mathbb{N}^n_d} \mathsf{f}_{i\kappa}\omega^\kappa$.

Remark that y is a vector indexed by the element of $\mathbb{N}^n_d = \{\kappa\in\mathbb{N}^n \ : \sum_{i=1}^n \kappa_i \leq d\}$ and has dimension $b_n(d) = \binom{n+d}{d}$, that might get very large.

The key concept to be defined is that of *$\mathcal{S}$-truncated moment vector*, i.e, for a finite sequence y, is their a probability measure P supported on $\mathcal{S}$ such that y be the sequence of moments associated to P. Indeed, this characterization captures all the difficulty of the problem, which does not involves the probability measure P any more :

$$\left\{\begin{array}{ll} \min & \sum\limits_{\kappa\in\mathbb{N}^n_d} \mathsf{f}_{0\kappa} y_\kappa \\ \text{s.t.} & \sum\limits_{\kappa\in\mathbb{N}^n_d} \mathsf{f}_{i\kappa} y_\kappa = b_i,\ i=1,...,m \\ & y_{(0,...,0)} = 1 \\ & y \text{ is a } \mathcal{S}\text{-truncated moment vector} \end{array}\right.$$

Subsequently, the constraint imposing the support $\mathcal{S}$ are referred to as *support constraints*, whereas the other constraints are denoted *moments constraints*.

The connection with SDP comes from the *moment matrix* and *localizing matrices* associated to y (see Def. 3.1.5 and Def. 3.1.8) since their semidefiniteness are necessary for y to be a $\mathcal{S}$-truncated moment vector. It turns out that a connection can also be established between SDP and the dual of the GPM, which involves polynomial non negativity condition. This connection relies on the fact that a sufficient condition for the non negativity of a polynomial is the existence of a s.o.s. representation, which can be formulated as a LMI (see 3.1.2.4).

The GPM proved to be a powerful tool for modelling some complex problems and enabled for instance the emergence of the *distributionnally robust* optimization framework [270]. Polynomial optimization, and more specifically 0/1 polynomial optimization, can be modeled as particular instance of the GPM, and we will see some specificity of the Lasserre's hierarchy for these problems. For a more detailed discussion on this problem, we refer the reader to the handbook [12] and the references therein [171, 174, 193, 209].

3.4.1.2 Historical overview

This problem has sprouted from an attempt of unification of the works of famous mathematicians like Chebyshev, Markov or Hoeffding, on the existence and uniqueness of a probability measure having specified moments and support, and on how deriving a bound, tighter as possible, on the expected value of such probability measures. This problem can be cast in a particular instance of the GPM

called *Classical Problem of Moment (CPM)* by taking $f_0(\omega) = \omega$ and f_i the monomials of degree up to d :

$$\begin{cases} \min & \mathrm{E_P}[\omega] \\ \text{s.t.} & \mathrm{E_P}[\omega^\kappa] = b_\kappa, \ \forall \kappa \in \mathbb{N}^n_d \\ & \mathrm{P}[\omega \in \mathcal{S}] = 1 \end{cases}$$

The origin of this problem can be traced back to the early work of Stieltjes that defined the *Moment Problem*, as the above problem with $n = 1$ and $\mathcal{S} = \mathbb{R}_+$. Then, the problem was extended to $\mathcal{S} = \mathbb{R}$ by Hamburger in 1921 and to a bounded interval ($\mathcal{S} = [0, 1]$) by Hausdorff in 1923.

The fundamental connection with psd matrices was established by Hamburger with the key result stating that a sequence is feasible for its problem if and only if its moment matrix is positive semidefinite :

Theorem 3.4.1 *Hamburger's theorem*
Let us consider a vector $y \in \mathbb{R}^{2d+1}$ indexed by $0, ..., 2d$ with $y_0 = 1$. Then y is a $\mathbb{R}$-truncated moment vector if and only if the moment matrix associated to y, $M_d(y)$, is positive semidefinite.

In this particular case, $M_d(y) \in \mathbb{S}^{d+1}$ with $M_d(y)_{i,j} = y_{i+j}$ for $i, j = 0, ..., d$. In the same vein, for $\mathcal{S} = \mathbb{R}_+$, a similar result involves both the moment and localizing matrices :

Theorem 3.4.2
Let us consider a vector $y \in \mathbb{R}^{2d+1}$ indexed by $0, ..., 2d$ with $y_0 = 1$ and define the polynomial $p \in \mathcal{P}^{1,1}$ such that $p(x) = x$. Then y is a $\mathbb{R}$-truncated moment vector if and only if $M_d(y) \succcurlyeq 0$ and $M_d(p, y) \succcurlyeq 0$, where $M_d(p, y) = \begin{pmatrix} y_1 & y_2 & \cdots & y_d \\ y_2 & y_3 & \cdots & y_{d+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_d & y_{d+1} & \cdots & y_{2d-1} \end{pmatrix}$ is the localizing matrix associated to the polynomial p.

Finally, a similar result holds for the case where $\mathcal{S} = \mathbb{R}^n$ and $d = 1$:

Theorem 3.4.3
Let y be a sequence indexed by the elements of $\mathbb{N}^n_2$ with $y_{(0,...,0)} = 1$. Then $\mathbb{R}^n$-truncated moment vector if and only if $M_1(y) \succcurlyeq 0$, with

$$M_1(y) = \begin{pmatrix} y_{(0,...,0)} & y_{(1,...,0)} & \cdots & y_{(0,...,1)} \\ y_{(1,...,0)} & y_{(2,0,..,0)} & \cdots & y_{1,...,1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{(0,...,1)} & y_{(0,...,1,1)} & \cdots & y_{(0,...,2)} \end{pmatrix}$$

Motivated by the earlier work of Curto and Fialkow [77] about moment matrices, Lasserre extended the connection between truncated moment vectors and semidefinite matrices to the polynomial restriction of the GPM.

3.4.1.3 Duality

In this section, we establish the duality between the GPM and a problem whose constraints consists of the non negativity of some polynomials over $\mathcal{S}$. We consider a "linear" formulation of the polynomial restriction of the GPM (3.8) obtained by considering each value $\mathrm{P}(\omega),\ \omega \in \mathcal{S}$ as a variable. Then the problem is a semi-infinite LP with non-negative variables and its dual is therefore a Linear Program with a finite number of variables but infinitely many constraints :

$$\begin{cases} \max & \sum_{i=1}^{m} b_i z_i \\ \text{s.t.} & \sum_{i=1}^{m} f_i(\omega) z_i \leq f_0(\omega), \quad \forall \omega \in \mathcal{S} \end{cases}$$

The whole constraints are equivalent to the non-negativity of the polynomial f_z on $\mathcal{S}$, where $f_z(\omega) = f_0(\omega) - \sum_{i=1}^m z_i f_i(\omega)$ is a polynomial whose coefficients are linear functions of z. We refer the reader to [174] for a more theoretical vision of the duality between moment vectors and non-negative polynomials.

The weak duality can be easily established. Let p^*, d^* and and P^*, z^* be the optimal values and the optimal solutions of the primal and dual GPM respectively :

$$d^* = \sum_{i=1}^m z_i^* b_i = \sum_{i=1}^m z_i^* \int_{\mathcal{S}} f_i(\omega)\mathrm{P}^*(\omega)d\omega = \int_{\mathcal{S}} \sum_{i=1}^m z_i^* f_i(x)\mathrm{P}^*(\omega)d\omega \leq \int_{\mathcal{S}} f_0(\omega)\mathrm{P}^*(\omega)d\omega = p^*$$

In [145], Isii proved the following theorem that provides Slater's type sufficient conditions for strong duality.

Theorem 3.4.4 *The combination of the three following conditions is sufficient for the strong duality to hold.*

(i) the functions $f_j,\ j = 0, ..., l$ are linearly independent;

(ii) b is an interior-point of the closure of the moment space, defined as $\{b \in \mathbb{R}^{l+1} : \exists \mathrm{P} \in \mathcal{M}(\mathcal{S}) : b_j = \mathrm{E}_{\mathrm{P}}(f_j(\xi)),\ j = 0, ..., l\}$;

(iii) both the primal and dual problem have feasible solutions.

In particular, in the case of the CPM, strong duality holds if b is an interior-point of the set of the moment vectors.

3.4.2 Non-negative polynomials and sum of squares

As mentioned before, there is a duality relation between the constraint that a vector be a $\mathcal{S}$-feasible moment sequence and the constraint that a polynomial be non-negative over $\mathcal{S}$. This leads to the study of such polynomials.

3.4.2.1 Non-negativity of a polynomial on $\mathbb{R}^n$

Definition 3.4.5 *Positive and nonnegative polynomial*
A polynomial p is positive (resp. nonnegative) on $\mathcal{S}$ if $p(x) > 0$ (resp. $p(x) \geq 0$) for all $x \in \mathcal{S}$.

We denote by $\mathcal{P}_+^{n,d}$ and $\mathcal{P}_{++}^{n,d}$ the set of d-degree polynomials that are nonnegative and positive over $\mathbb{R}^n$ respectively. Note that these sets are empty when d is odd. We are interested in characterizing the fact that $p \in \mathcal{P}_+^{n,d}$, that is finding necessary and/or sufficient checkable conditions on the coefficients of p so that this property be satisfied. Such conditions are called *nichtnegativstellensatz* and the equivalent for characterizing the positivity of a polynomial are called *positivstellensatz*.

To this end, the property that a polynomial be s.o.s. (see Paragraph 3.1.2.4), which implies the non-negativity on $\mathbb{R}^n$, is crucial. Let denote $\Sigma^{n,d}$ the polynomials in $\mathcal{P}^{n,d}$ that are s.o.s. : $\Sigma^{n,d} \subset \mathcal{P}_+^{n,d}$. This relationship was made more precise by Hilbert in 1888.

Theorem 3.4.6 *Hilbert's theorem*

$$\Sigma^{n,d} = \mathcal{P}_+^{n,d} \Leftrightarrow n = 1, d = 2 \text{ or } (n,d) = (2,4)$$

Thus, apart three special cases ($(n,2d) = (1,2d),(n,2),(2,4)$), there exists some non-negative polynomials that do not admit a s.o.s representation, the most famous example being the Motzkin polynomial $x^2y^4 + x^4y^2 - 3x^2y^2 + 1$. The argument used to show that this polynomial is not a sum of square can be extended when one adds a constant to the polynomial. This yields an example of positive polynomial that is not a sum of square and show that the inclusion $\Sigma^{n,d} \subset \mathcal{P}^{n,d}_{++}$ is also strict.

Thus the s.o.s. property is generally not stronger than the non-negativity. However it is very interesting from a computational viewpoint. Indeed, while the problem of testing the nonnegativity of a polynomial of degree greater than four is NP-hard, one can test efficiently whether a polynomial is s.o.s. by solving a SDP, as discussed in Paragraph 3.1.2.4 :

Theorem 3.4.7 *Sum of square representation*
Let us consider a polynomial p of even degree $2d$. Then $p \in \Sigma^{n,2d}$ if and only is s.o.s. if and only there exists a matrix $Q \in \mathbb{S}_+^{b_n(d)}$ such that $\sum_{\kappa_1+\kappa_2=\kappa} Q_{\kappa_1,\kappa_2} = \mathsf{p}_\kappa,\ \forall \kappa \in \mathbb{N}^n_d$.

Proof 3.4.8 *First observe that the equalities $\sum_{\kappa_1+\kappa_2=\kappa} Q_{\kappa_1,\kappa_2} = \mathsf{p}_\kappa$ means that $p(x) = z(x)^T Q z(x)$ with $z(x)$ the vector of d-degree monomials $z(x) = (1, x_1, ..., x_n, x_1^2, x_1x_2, ...x_n^d)$.*

Then, if $Q = U\Lambda U^T$ is the eigenvalue factorization of Q : $p(x) = \sum_i \lambda_i (Uz(x))_i^2$ and $Q \succcurlyeq 0$ implies that $\lambda_i \geq 0$ and therefore p is s.o.s.

Conversely, if $p(x) = \sum_i u_i(x)^2$, then the polynomials u are at most of degree d and $u_i(x) = \mathsf{u}_i^T z$. Then $Q = UU^T$ where U is the matrix with column vectors u_i. □

Then the s.o.s condition can be formulated as : $\begin{cases} B^{d,\kappa} \bullet Q = \mathsf{p}_\kappa,\ \forall \kappa \in \mathbb{N}^n_d \\ Q \succcurlyeq 0 \end{cases}$ with the matrices $B^{d,\kappa}$ from Def. 3.1.4. Thus, deciding whether a polynomial admits a s.o.s. representation can be settled by solving a SDP whose variable Q has size $b_n(d)$ and with $b_n(2d)$ equations.

In the three particular cases of Hilbert's theorem 3.4.6, this s.o.s. condition is equivalent to the non-negativity of the polynomial. In particular, $d = 2$ corresponds to the very interesting case of quadratic functions that are studied at Appendix 2.5.3. Very briefly, we recover the fact that $p(x) = x^T P x + 2p^T x + \pi$ is non negative over $\mathbb{R}^n$ if and only if $Q = \begin{pmatrix} \pi & p^T \\ p & P \end{pmatrix} \succcurlyeq 0$.

3.4.2.2 Non-negativity of a polynomial over a semi-algebraic set $\mathcal{S}$

The univariate case : $n = 1$
With $n = 1$, the Theorem 3.4.6 states that a polynomial p is nonnegative over $\mathbb{R}$ if and only if it is a sum of square. We have just seen that this is equivalent to require that a certain matrix made of the coefficient of p be psd. This result can be extended to the nonnegativity of p over some smaller sets $\mathcal{S}$, by applying the Theorem 3.4.6 to $p(f(t))\ \forall t \in \mathbb{R}$ with f such that $f(\mathbb{R}) = \mathcal{S}$. This method was applied in [41] which leads to semidefinite conditions for a polynomial being non-negative on the following set :

- $\mathcal{S} = \mathbb{R}$;
- $\mathcal{S} = \mathbb{R}_+$;
- $\mathcal{S} = [0, a], a > 0$;
- $\mathcal{S} = [a, +\infty[$;
- $\mathcal{S} =]-\infty, a]$;
- $\mathcal{S} = [a, b], a < b$;

In the case where $\mathcal{S}$ is an union of such sets, it suffices to impose these conditions over all the subsets that constitute $\mathcal{S}$.

General case
Clearly, the non-negativity of a d-degree polynomial p over a semi-algebraic set $\mathcal{S} = \{x \in \mathbb{R}^n : g_k(x) \geq 0,\ k = 1, ..., l\}$ can be conservatively approximated by the following condition :

$$p = u_0 + \sum_{k=1}^{l} u_k g_k \quad \text{for some} \quad u_k \in \mathcal{P}_+^{n,d},\ k = 0, ..., l$$

A sufficient condition for $u_k \in \mathcal{P}_+^{n,d}$ is $u_k \in \Sigma^{n,d}$, and therefore another conservative approximation is :

$$p = u_0 + \sum_{k=1}^{l} u_k g_k \text{ for some } u_k \in \Sigma^{n,d},\ k = 0, ..., l$$

Under suitable conditions on $\mathcal{S}$, the Putinar's theorem 2.5.13 states that the latter conservative approximation is in fact equivalent to the positivity of p on $\mathcal{S}$. This allows to reduce the conservativeness of the approximation as much as desired. Indeed, the following condition : $p(x) \geq 0,\ \forall x \in \mathcal{S}$ implies that $\forall \varepsilon > 0$, $p(x) + \varepsilon > 0,\ \forall x \in \mathcal{S}$. This polynomial approximates p within ε and its positivity can be determined by a SDP. Thus, we are able to asymptotically solve the problem of the nonnegativity of p over $\mathcal{S}$ by a sequence of SDP.

3.4.3 Semidefinite relaxation of the GPM : the Lasserre's hierarchy

3.4.3.1 Definition

The previous sections provide all the necessary elements to build the hierarchy of semidefinite relaxations of the GPM and of its dual. This hierarchy was proposed by Lasserre that proved its convergence to the optimal value of the GPM without assuming strong duality but provided that $\mathcal{S}$ satisfies certain conditions (in particular, compactness). We consider the following polynomial instance of the GPM and its dual :

$$\left\{\begin{array}{ll} \inf & \int_{\mathcal{S}} f_0(\omega)\mu(\omega)d\omega \\ \text{s.t.} & \int_{\mathcal{S}} f_i(\omega)\mu(\omega)d\omega = b_i,\ i = 1, ..., m \\ & \mu \in \mathcal{M}(\mathcal{S}) \end{array}\right. \qquad \left\{\begin{array}{ll} \sup & b^T z \\ \text{s.t.} & f_0(\omega) - \sum_{i=1}^m z_i f_i(\omega) \geq 0,\ \forall \omega \in \mathcal{S} \\ & z \in \mathbb{R}^m \end{array}\right.$$

with $\mathcal{S} = \{x \in \mathbb{R}^n : g_k(x) \geq 0,\ k = 1, ..., l\}$. $f_j,\ j = 0, ..., m$ and $g_k,\ k = 1, ..., l$ are polynomials of degree at most d and $f_0(0) = 0$. Furthermore, we assume here that the constraint indexed by $i = 1$ corresponds to P being a probability distribution, i.e., $f_1(\omega) = 1$ and $b_1 = 1$.

Then, the primal problem is equivalent to

$$\left\{\begin{array}{ll} \min & \sum_{\kappa \in \mathbb{N}_d^n} \mathsf{f}_{0\kappa} y_\kappa \\ \text{s.t.} & \sum_{\kappa \in \mathbb{N}_d^n} \mathsf{f}_{i\kappa} y_\kappa = b_i,\ i = 1, ..., m \\ & y \text{ is a } \mathcal{S}\text{-truncated moment vector} \end{array}\right.$$

As discussed in paragraph 3.1.3.4, a necessary condition for y to be a truncated moment vector is that its moment matrix be positive semidefinite. This matrix can be expressed as a LMI, by means of the matrices $B^{\kappa,d}$ defined at Def. 3.1.4 :

$$y \in \mathbb{R}^{b_n(2d)} \text{ is a truncated moment vector} \Rightarrow \sum_{\kappa \in \mathbb{N}_{2d}^n} y_\kappa B^{\kappa,d} \succcurlyeq 0$$

It was also noticed in the paragraph 3.1.3.5 that for any $2v$-degree polynomial p that is non negative over $\mathcal{S}$, then the localizing matrix $M_{r-d}(p,y)$ associated to y and p is positive semidefinite. In particular, as $\mathcal{S}$ is defined through the nonnegativity of the polynomials $g_k,\ k=1,...,l$, then $M_{r-v_k}(g_k,y)\succcurlyeq 0$ holds for $k=1,...,l$, with v_k such that $\deg(g_k)=2v_k$ or $\deg(g_k)+1=2v_k$ depending on the parity of $\deg(g_k)$.

Let g_0 be the 0-degree polynomial such that $g_0(x)=1$. We observed that $M_r(g_0,y)=M_r(y)$. Thus, we have necessary conditions expressed as LMI for y to be a $\mathcal{S}$-truncated moment vector, which leads to the following semidefinite relaxation of the GPM :

$$(Q_r)\begin{cases} q_r^* = & \inf & \mathsf{f_0}^T y \\ & \text{s.t.} & \mathsf{f_i}^T y = b_i,\ i=1,...,m \\ & & \sum\limits_{\kappa\in\mathbb{N}^n_{2r}} B^{r-v_k,\kappa}(g_k)y_\kappa \succcurlyeq 0,\ k=0,...,l \\ & & y\in\mathbb{R}^{\mathbb{N}^n_{2r}} \end{cases} \tag{3.9}$$

Similarly to v_k, let w_j be such that $\deg(f_j)=2w_j$ or $\deg(f_j)+1=2w_j$. If p^* denotes the optimal value of the GPM, then $q_r^*\leq p^*$ for $r\geq r_0=\max\{\{v_k\}_{k=0,...l},\{w_j\}_{j=0,...m}\}$.

Note that $M_{r-v}(p,y)$ is a principal submatrix of $M_{r+1-v}(\mathsf{p}*y)$ and therefore $M_{r+1-v}(p,y)\succcurlyeq 0\Rightarrow M_{r-v}(p,y)\succcurlyeq 0$. As a consequence, any feasible solution of (Q_{r+1}) leads (by truncation) to a feasible solution of (Q_r) with the same objective value and therefore $q_r^*\leq q_{r+1}^*$.

It turns out that the dual of the obtained SDP can be interpreted as a conservative approximation of the dual of the GPM. Recall that the unique constraint of the dual GPM is the constraint of nonnegativity of the polynomial $f_z=f_0-\sum_{i=1}^m z_i f_i$ on $\mathcal{S}$. According to Putinar's theorem, (Theorem 2.5.13), under certain conditions on $\mathcal{S}$, $f_z(\omega)>0$ on $\mathcal{S}$ is equivalent to the existence of $l+1$ s.o.s. polynomials $u_k,\ k=0,...,l$ such that $f_z=\sum_{k=0,...,l}u_k g_k$.

Replacing $f_z(x)\geq 0$ on $\mathcal{S}$ by $f_z(x)>0$ on $\mathcal{S}$ and assuming that the polynomials $u_k,\ k=0,...,l$ are at most of degree $2(r-v_k)$ leads to the following conservative approximation of the dual GPM :

$$(Q_r^*)\begin{cases} \sup & b^T z \\ \text{s.t.} & \sum\limits_{k=0}^{l} u_k g_k = f_0-\sum_{i=1}^m z_i f_i \\ & B^{r-v_k,\kappa}\bullet X^k = \mathsf{u}_{k\kappa}, \forall\kappa\in\mathbb{N}^n_{2(r-v_k)},\ k=0,...,l \\ & X^k\succcurlyeq 0,\ k=0,...,l \\ & z\in\mathbb{R}^m,\ u_k\in\mathbb{R}^{b_n(2(r-v_k))} \\ & X^k\in\mathbb{S}^{r-v_k},\ k=0,...,l \end{cases}$$

The equivalence between the requirement that the polynomials $u_k,\ k=0,..,l$ be s.o.s. and the semidefinite constraint results from Theorem 3.4.7.

Thus, a relation of duality between these two problems begins to take shape. It remains to replace $\sum\limits_{k=0}^{l} u_k g_k$ by its expression in function of X_k. For $\kappa\in\mathbb{N}^n_r$:

$$\begin{aligned} \left\{\sum_{k=0}^{l} u_k g_k\right\}_\kappa &= \sum_{k=0}^{l}\left[\sum_{\kappa_1+\kappa_2=\kappa}\mathsf{g}_{k\kappa_1}\mathsf{u}_{k\kappa_2}\right] \\ &= \sum_{k=0}^{l}\left[\sum_{\kappa_1+\kappa_2=\kappa}\mathsf{g}_{k\kappa_1}(B^{r-v_k,\kappa_2}\bullet X^k)\right] \\ &= \sum_{k=0}^{l}\left[\left(\sum_{\kappa_1+\kappa_2=\kappa}\mathsf{g}_{k\kappa_1}B^{r-v_k,\kappa_2}\right)\bullet X^k\right] \\ &= \sum_{k=0}^{l} B^{r-v_k,\kappa}(g_k)\bullet X^k \end{aligned}$$

This makes the relation of duality to appear :

$$(Q_r^*)\begin{cases} \sup & b^T z \\ \text{s.t.} & \mathsf{f}_{0\kappa} - \sum_{i=1}^{m} z_i \mathsf{f}_{i\kappa} = \sum_{k=0}^{l} B^{r-v_k,\kappa}(g_k) \bullet X^k, \ \forall \kappa \in \mathbb{N}^n_{2(r-r_0)} \\ & X^k \succcurlyeq 0, \ k = 0, ..., l \\ & z \in \mathbb{R}^m, X^k \in \mathbb{S}^{r-v_k}, \ k = 0, ..., l \end{cases}$$

If we assume that strong duality holds between (Q_r) and (Q_r^*), we can easily deduce the convergence of this hierarchy of relaxation. Indeed, under appropriate conditions on $\mathcal{S}$, namely the Putinar's conditions (see Def. 2.5.12), the Putinar's theorem states that there exists $r \geq r_0$ such that the semidefinite conditions of (Q_r^*) are sufficient for the strict positivity of f_z on $\mathcal{S}$. From this, Lasserre [171] deduced that $\forall \epsilon, \exists r(\epsilon) : d^* - \epsilon \leq d_r^*$ for any $r \geq r(\epsilon)$. Combining this with weak duality, it comes that $d^* - \epsilon \leq d_r^* \leq p_r^* \leq p^*$. From the construction of (Q_r) it comes that $p_r^* \leq p_{r+1}^* \leq p^*$. Then it follows immediately from strong duality that $\lim_{r \to +\infty} p_r^* = p^*$. This result was also proved without the strong duality assumption, see [171].

3.4.3.2 Application to polynomial optimization

Let us consider a polynomial optimization problem :

$$\begin{cases} p^* = & \min & f_0(x) \\ & \text{s.t.} & g_k(x) \geq 0, \ k = 1, ..., l \\ & & x \in \mathbb{R}^n \end{cases}$$

or equivalently, $p^* = \min_{x \in \mathcal{S}} f_0(x)$ where $\mathcal{S} = \{x \in \mathbb{R}^n : g_k(x) \geq 0, \ k = 1, ..., l\}$.

Then the relation between this problem and the GPM is double-sided. Indeed, this problem can be formulated either as a moment problem or as a polynomial non-negativity problem. This latter formulation is obvious : $p^* = \max z : f_0(x) - z \geq 0, \forall x \in \mathcal{S}$, whereas the moment formulation relies on the following proposition (see [174] for the proof) :

Proposition 3.4.9 $\min_{x \in \mathcal{S}} f_0(x) = \begin{cases} \min & \int_{\mathcal{S}} f_0(\omega) \mathrm{P}(\omega) d\omega \\ s.t. & \mathrm{P} \in \mathcal{M}(\mathcal{S}) \end{cases}$

Consequently, any polynomial optimization problem can benefit from the GPM results. For instance in the case of a QCQP, i.e., a polynomial problem where all the involved polynomials are of degree 2, the rank 1 of the Lasserre's hierarchy corresponds to the standard SDP relaxation (see paragraph 3.3.2).

Another famous kind of polynomial problems are the 0/1-polynomial problems, whose feasible set are included in $\{0,1\}^n$. This implies automatically that $\mathcal{S}$ satisfies the Putinar's conditions and the convergence of the hierarchy is ensured. Furthermore, the corresponding SDP are simplified to a specific form with at most 2^n primal variables, regardless of the rank in the hierarchy. To see this, it suffices to recall that the primal variable y_κ represents the moment associated to κ, i.e., $\int_{\mathcal{S}} \omega^\kappa \mathrm{P}(\omega) d\omega$. If $\omega \in \{0,1\}^n$, then $\omega^\kappa = \omega^{\kappa'}$ for any κ' with $\kappa'_i \geq 1$ if $\kappa_i = 1$, 0 otherwise. Consequently, it suffices to define the variable y_κ for $\kappa \in \{\kappa \in \mathbb{N}^n : \kappa_1 \leq 1, \ i = 1, ..., n\}$, a set that contains only 2^n elements.

Finally, Lasserre proved in [170] that in the case of 0/1 programs, the feasible set of the SDP relaxations reaches the convex hull of the feasible set of the original problem in at most n iterations. As a consequence, the optimal value is attained in at most n iterations. We refer the reader to [132, 168, 169, 172] for complementary reading on this subject.

3.4.3.3 Comparison with other hierarchies of relaxation for 0/1-LP

Embedding of the Sherali-Adams hierarchy of linear relaxations
The Sherali-Adams hierarchy of relaxations for 0/1-LP is a sequence of Linear Programs that leads to the full representation of the convex hull of the original problem. This process is described at Appendix 3.4.3.2, following the original paper [240].

An alternate view of this hierarchy is proposed in [173], that makes it to appear as a subcase of the Lasserre's hierarchy. A crucial element for this reformulation lies in an equivalence between a set of linear constraints (since the Sherali-Adams relaxations are linear programs) and a semidefinite constraint, that can roughly be understood as the equivalence between a matrix being psd and all its eigenvalues being nonnegative. Thus, it appears that all the linear constraints of the Sherali-Adams relaxation can be gathered within some semidefinite constraints, that concern some principal submatrices of the matrices $M_n(y)$ and $M_{n-v_k}(g_k, y)$ of the Lasserre's hierarchy.

Comparison with the Lovász-Schrijver hierarchy of semidefinite relaxations
The comparison with the Lovász-Schrijver hierarchy of semidefinite relaxations, described at paragraph 3.3.3.2, was also presented in [173]. It reveals that at a same rank r, the Lasserre relaxation is tighter than the Lovász-Schrijver relaxation. But the latter involves $O(n^{r-1})$ matrices of order $n+1$, i.e., $O(n^{r+1})$ variables, instead of one matrix of order $O(n^r)$, i.e., $O(n^{2r})$ variables.

3.4.4 Applications

The moment paradigm was exploited in a variety of applications by Bertsimas and al. which are summarized in the Chapter 16 of [259]. They include porfolio management, queuing network and probability bounding, as detailed below.

3.4.4.1 Probability

The problem of finding the best possible bound on the probability that the random vector X belongs to a set $S \in \mathcal{R}$ can be modeled as a (CMP), by taking for f_0 the indicator of S denoted $\mathbb{1}_S$:

$$\mathrm{P}[X \in S] = \int_S \mathrm{P}(\omega) d\omega = \int_{\mathcal{R}} \mathbb{1}_S(\omega)\mathrm{P}(\omega) d\omega$$

For example, in the electricity generation context, we are able to determine the minimal and maximal probability that some may satisfy a given demand, whereas these means of production are subject to random failures, whose mean is known.

The probability bounding problem deals with the minimization of a probability $\mathrm{P}[\xi \in \mathcal{K}]$ on a set of probability distribution with known moments, where $\mathcal{K}$ is a semi-algebraic set. It is an instance of the GPM where the function f_j are some monomials and the objective function h is a piecewise polynomial, namely the indicator function of $\mathcal{K}$. If $\mathcal{K} = \{\xi \in \mathbb{R}^n : h_k(\xi) \geq 0,\ k = 1, ..., l\}$, then the considered probability is equivalent to the following joint probability :

$$\mathrm{P}[\xi \in \mathcal{K}] = \mathrm{P}[h_k(\xi) \geq 0,\ k = 1, ..., l]$$

3.4.4.2 Portfolio management

We consider here a problem that concerns European Call option, that is the right, but not the obligation, to buy an agreed quantity of a particular asset at a certain time (the *maturity* T) for a certain price (the *strike* s). The buyer pays a fee for this right and he hopes that the price of the asset will rise in

the future so that he makes a gain up to the strike price. Thus, if ω is the price of the asset at time T, the payoff of such an option is $\max\{0, \omega - s\}$.

It has been shown that under the non-arbitrage assumption, the price of such an option is given by $q(s) = \mathrm{E}_P[\max\{0, \omega - s\}]$, where the expectation is taken over the martingale measure P. Suppose that we are interested in obtaining an upper bound on $q(s)$, given that we have estimated the mean μ and the variance σ of the price ω. Then we solve the following problem :

$$\begin{cases} \sup & \mathrm{E}[\max\{0, \omega - s\}] \\ \text{s.t.} & \mathrm{E}_P[\omega] = \mu \\ & \mathrm{E}_P[\omega^2] = \mu^2 + \sigma^2 \end{cases}$$

More generally, this can be extended to multivariate case, by considering n options, with the knowledge of the first and second moments of the random vector ω of the options prices. We still want to determine a bound over the price of the first option :

$$\begin{cases} \sup & \mathrm{E}[\max\{0, \omega_1 - s\}] \\ \text{s.t.} & \mathrm{E}_P[\omega^\kappa] = q_\kappa, \ \forall \kappa \in \mathbb{N}_2^n \end{cases}$$

This can easily be reduced to the previously studied layout. Indeed, the dual constraint $\sum_{\kappa \in \mathbb{N}_2^n} \omega^\kappa y_\kappa \geq \Phi(\omega), \ \forall \omega \in \mathcal{R}$ is indeed equivalent here to the non-negativity of the two polynomials :

$$\begin{cases} \sum\limits_{\kappa \in \mathbb{N}_2^n} \omega^\kappa y_\kappa - \omega_1 + s \geq 0, \ \forall \omega \in \mathcal{R} \\ \sum\limits_{\kappa \in \mathbb{N}_2^n} \omega^\kappa y_\kappa \geq 0, \ \forall \omega \in \mathcal{R} \end{cases}$$

3.4.4.3 Queuing system

A very prominent and substantial example of application of the moment paradigm was provided by Bertsimas and Mora in [38] and concerns the study of queuing network.

Very roughly, the underlying idea is to characterise the state of a dynamic system by a set of time-dependent random vectors L^t, which are functions of uncertain parameters $(\xi_0, ..., \xi_t)$ and of a *scheduling policy*, i.e., a set of command variables that depends on the state of the system.

The objective is not to optimize explicitly this policy, which is a very hard problem, but to compute bounds on a certain criteria, for any policy that satisfies the following conditions :

- stationarity, i.e., the probability distribution (and hence the moments) of L^t, is independent of the time ;
- stability, i.e., the mean of the L^t is finite ;

These properties, associated to other characteristics and relationship between component of L^t, induce linear constraints that restrict the moments of L^t. Given that the optimization criteria is a linear combination of the moments of L^t, we are typically in the framework of the GPM and the moment matrices have to be psd.

Two points deserve a particular attention. First, some of the random variables describing the state of the system are required to be binary. Then the second-order moment of such a variable equals its means and the corresponding equality constraint can be imposed to the moment matrix. This makes an interesting connexion with semidefinite relaxation of combinatorial problems.

Second, the scheduling policy is expressed via the the value given to the probability of the conditional variables, for instance $L_i = 1 | L_j = 1$. Indeed, these probability can be interpreted as a measure of the consequences of the decision made for the case when L_j equals 1.

3.5 SDP for facing uncertainty

3.5.1 Semidefinite programming for robust optimization

As explained in the paragraph 3.7.3, Robust Optimization is a distribution-free methodology consisting in optimizing the *worst-case* on a given uncertainty set $\mathcal{U}$, so that the solution be feasible for any realization of uncertainty. See [31] for a complete account on the subject. In this section, we show that semidefinite programming is a powerful tool for dealing with such problems, as established by several authors [32, 89].

3.5.1.1 Robust least-squares

In this section, we review the work presented in [89]. A least-square problem consists of minimizing the distance (Euclidean norm) between a vector of observations b and the result of linear transformation $A \in \mathbb{R}^{n\times m}$, for an input x to determine.

We consider such a problem where the input data A and b are unknown but are bounded and presumed to belong to the following ellipsoids (see Def. 2.2.65) :

$$A \in \{A(\omega) = A^0 + \sum_{i=1}^{p} \omega_i A^i : \|\omega\| \leq 1,\ \omega \in \mathbb{R}^p\}$$
$$b \in \{b(\omega) = b^0 + \sum_{i=1}^{p} \omega_i b^i : \|\omega\| \leq 1,\ \omega \in \mathbb{R}^p\}$$

with $(A^i, b^i) \in \mathbb{R}^{n\times m} \times \mathbb{R}^n,\ i = 0, ..., p$. Thus the robust version of the problem is :

$$\min_x \max_{\|\omega\|\leq 1} \|A(\omega)x + b(\omega)\|$$

Equivalently, we can minimize the square of the distance $f(x) = \max_{\|\omega\|\leq 1} \|A(\omega)x + b(\omega)\|^2$. Then, by defining some appropriate matrices : $M^0(x) = A^0x + b^0 \quad M(x) = \sum_{i=1}^{p} A^i x + b^i$:

$$f(x) = \max_{\|\omega\|\leq 1} \left\|M^0(x) + M(x)\omega\right\|^2 = \max_{\|\omega\|\leq 1} \tilde{\omega}^T \begin{pmatrix} M^0(x)^T M^0(x) & M^0(x)^T M(x) \\ M(x)^T M^0(x) & M(x)^T M(x) \end{pmatrix} \tilde{\omega}$$

Thus, we aim at maximizing a quadratic function while satisfying the constraint $\|\omega\| \leq 1$, which has a quadratic form : $\omega^T\omega \leq 1$. The problem is therefore a QCQP and by applying the S-Lemma 3.1.2.3, it is equivalent to the following SDP :

$$\begin{cases} \min_{x,y,\lambda} & y \\ \text{s.t.} & \begin{pmatrix} y - M^0(x)^T M^0(x) - \lambda & -M^0(x)^T M(x) \\ -M(x)^T M^0(x) & \lambda I - M(x)^T M(x) \end{pmatrix} \succcurlyeq 0 \\ & \lambda \geq 0 \end{cases}$$

Finally, the linearity w.r.t x is recovered by applying the Schur's complement :

$$\begin{cases} \min_{x,y,\lambda} & y \\ \text{s.t.} & \begin{pmatrix} y - \lambda & 0 & M^0(x)^T \\ 0 & \lambda I & M(x)^T \\ M^0(x) & M(x) & I \end{pmatrix} \succcurlyeq 0 \end{cases}$$

3.5.1.2 Robust problems with ellipsoidal uncertainty set

It was shown in [32, 90] that certain classes of robust optimization problems with ellipsoidal uncertainty set can be reformulated as SDP or SOCP :

- If the initial problem is a robust LP, there exists an exact formulation of the robust counterpart as a SOCP ;
- If the initial problem is a robust CQCQP (Convex Quadratically Constrainted Quadratic Program), there exists an exact formulation of the robust counterpart as a SDP ;
- If the initial problem is a robust SOCP, under certain conditions on the uncertainty, there exists an exact formulation of the robust counterpart as a SDP.

We illustrate this principle on the case of a simple CQCQP :

$$(P)\begin{cases}\min & c^T x \\ \text{s.t.} & x^T A^T A x \leq 1,\ \forall A \in \{A^0 + \sum_{j=1}^{k} u_j A^j \mid \|u\|_2 \leq 1\}\end{cases}$$

The idea is to replace the variable A, for which the constraint of belonging to $\mathcal{U}$ is quite complicated, by u, for which the constraint is simple : $\|u\|_2 \leq 1$. By applying the following equivalence :

$$\|u\|_2 \leq 1 \Leftrightarrow 1 - u^T u = \begin{pmatrix}1\\u\end{pmatrix}^T \begin{pmatrix}1 & 0\\0 & -I\end{pmatrix}\begin{pmatrix}1\\u\end{pmatrix} \geq 0$$

The problem can be written as following :

$$(P)\begin{cases}\min & c^T x \\ \text{s.t.} & \begin{pmatrix}1\\u\end{pmatrix}^T \left(\begin{pmatrix}1 & 0\\0 & 0\end{pmatrix} - F(x)^T F(x)\right)\begin{pmatrix}1\\u\end{pmatrix} \geq 0 \qquad \text{with } F(x) = (A^0 x, ..., A^k x) \\ & \forall \begin{pmatrix}1\\u\end{pmatrix}^T \begin{pmatrix}1 & 0\\0 & -I\end{pmatrix}\begin{pmatrix}1\\u\end{pmatrix} \geq 0\end{cases}$$

The S-lemma (see paragraph 3.1.2.3) enables us to transform this into a SDP. The constraint becomes : there exists $\lambda \geq 0$ such that :

$$\begin{pmatrix}1 & 0\\0 & 0\end{pmatrix} - F(x)^T F(x) - \lambda \begin{pmatrix}1 & 0\\0 & -I\end{pmatrix} \succcurlyeq 0 \Leftrightarrow \begin{pmatrix}1-\lambda & 0\\0 & \lambda I\end{pmatrix} - F(x)^T F(x) \succcurlyeq 0$$

By recognizing the Schur complement, this is equivalent to :

$$\begin{pmatrix}\begin{pmatrix}1-\lambda & 0\\0 & \lambda I\end{pmatrix} & F(x)^T\\ F(x) & I\end{pmatrix} \succcurlyeq 0$$

Finally, the robust counterpart of the problem is :

$$(P)\begin{cases}\min & q^T x \\ \text{s.t.} & \begin{pmatrix}\begin{pmatrix}1-\lambda & 0\\0 & \lambda I\end{pmatrix} & F(x)^T\\ F(x) & I\end{pmatrix} \succcurlyeq 0\end{cases}$$

Another approach was proposed by Bertsimas and Sim [43], that is not a reformulation but an approximation and that preserves complexity of the original problem, i.e., a LP remains a LP, and so on. On the other hand, the obtained solution is not robust any more, but under appropriate assumptions, a guarantee on the probability of feasibility can be provided.

3.5.2 Semidefinite programming for distributionnally robust optimization

The term of *distributionally robust optimization* was coined by Calafiore and El-Ghaoui in [68]. It deals with optimization facing incompletely specified uncertainty. This means that only partial information is available on the probability distribution of the uncertain parameters. Such a framework is widespread in the real-world, since evaluating precisely a probability distribution is generally a challenging task. Then the aim is to overcome the ambiguity on probability distribution that prevents from applying the classical stochastic programming methods.

This relatively recent way to deal with uncertainty appears as a compromise between stochastic programming, where the probability distribution is supposed to be perfectly known, and the robust optimization, where only information on the support is required. Thus, in the distributionally approach, the probability distribution is partially specified though certain characteristics, such as support and order up to k moments, or properties such as symmetry, independence or radiality. In the case where information concerns only moments and support, the problem is strongly related to the GPM (see 3.4) which establishes a bridge with semidefinite programming. This approach was exploited in [40, 41, 259].

The available information on the probability distribution is used to define the class $\mathcal{P}$ of matching distributions and the optimization is made on the worst probability distribution of this class, in a robust perspective. This leads for instance to a distributionally robust treatment of chance-constraints, as studied in [270] : $\max_{\mathrm{P}\in\mathcal{P}} \mathrm{P}[f(x,\xi) \geq 0] \geq 1-\varepsilon$. It turns out that for a class $\mathcal{P}$ characterized with mean and second moment matrix, the obtained problem corresponds to the well-known CVaR approximation of chance-constraints [201].

The distributionnally robust approach coincides with the minimax problem, a very classical approach used in decision theory, where the problem to solve is : $\min_{x\in X} \max \mathrm{E}_{\mathrm{P}}(U(f(x,\xi))$, with U is an utility function that reflects the risk aversion of the decision makers. In this context, the incomplete description of the probability distribution is called *ambiguity*. This approach has received a great deal of attention since the pionneering work of [237] and its application to the newsvendor problem. Recently, Delage and Ye [82] proved that this problem can be solved in polynomial time by the ellipsoid method in the case where X is convex with a separation oracle, U is concave piecewise linear, $f(\xi,x)$ is concave in ξ and convex in x and one can find subgradients of f in polynomial time, even when the moments are themselves subject to uncertainty.

3.5.3 Semidefinite programming for two-stages optimization

In this section based on the work of Terlaky et al. [81], we present an application of the convex quadratic regression problem, detailled in paragraph 3.1.2.1, to two-stages stochastic programming. This kind of problem, also called *problem with recourse*, has been introduced in Appendix 3.7.1. We consider here a particular case of such problems, linear with a second stage problem where only the right-hand term is random :

$$\left\{\begin{array}{lll} \min & c^T x + \mathrm{E}[\mathcal{Q}(x,\omega)] & \\ \text{s.t.} & Ax \leq b & \\ & x \geq 0 & \\ & \mathcal{Q}(x,\omega) = \left\{\begin{array}{ll} \min & q^T y \\ \text{s.t.} & Tx + Wy = \omega \\ & y \geq 0 \end{array}\right. & \end{array}\right. \tag{3.10}$$

We assume that any linear program involved (first-stage and second-stage, for any value of x and ω) has finite optimal solution. We denote by $\mathcal{Q}(x) = \mathrm{E}[\mathcal{Q}(x,\omega)]$ and the idea here is to replace $\mathcal{Q}$ by a quadratic approximation, optimised on a set of value (x_i, ϕ_i), where ϕ_i is the result of the optimization of the second-stage problem for $x = x_i$ and for ω approximated by a discrete sampling (Monte-Carlo method). The convex quadratic approximation is made regarding the process described

at §3.1.2.1. According to the choice of the norm L_1, L_2 or L_∞ for the distance to minimize between the set of points (x_i, ϕ_i) and the approximation, this process comes down to solving a SDP or a mixed SOCP-SDP.

Let us denote x^TQx+p^Tx+r the quadratic approximation thus obtained, then the approximated problem is the following Quadratic Program :

$$\begin{cases} \min & x^TQx + (c+p)^Tx + r \\ \text{s.t.} & Ax \leq b \\ & x \geq 0 \end{cases}$$

As the objective function is convex, this problem can be solved by any quadratic solver.

The authors proposed to improve iteratively the quadratic approximation by adding successively some new points to the set (x_i, ϕ_i), as summarized in the following algorithm :

1. Determine a set S of N pairs (x_i, ϕ_i) ;
2. Compute (by semidefinite optimization) a quadratic convex approximation of $\mathcal{Q}$ base on S ;
3. Solve the Quadratic Program thus obtained. Let us denote x^* the corresponding solution ;
4. Determine the optimal value ϕ^* of the second-stage problem for x^*, with a discrete approximation of ω;
5. If x^* is not "good enough", add (x^*, ϕ^*) to S and return to 2.

In [81], the authors conducted some numerical experiments, up to $N = 3000$ point in an $n = 50$ dimension space. The results indicate that this algorithm is more efficient that the classical least-square method, especially by using the L_2-approximation.

Another use of SDP for two-stages optimization was proposed by Bertsimas et al. in [37]. The considered problem is quite similar to the problem (3.10) except that the cost q of the second-stage problem is also uncertain. Furthermore, the probability distribution of the random parameters is not known exactly, but in distributionnally robust spirit, it is chosen from a set of distributions with known mean and second moment matrix. If $\mathcal{P}$ denotes this class, then we aim at minimizing the worst-case over these distribution : $\min c^Tx + \max_{\mathrm{P}\in\mathcal{P}} \mathrm{E}_{\mathrm{P}}[\mathcal{Q}(x,\omega)]$. For this reason, it is named *minimax stochastic optimization problem*. In [37], the authors also considered the possibility of incorporating a risk measure into the objective, by means of an utility function. Then, for some particular utility function, when only the objective of the second-stage problem is uncertain, then a SDP formulation is provided. For uncertainty in the right-hand side of the second-stage problem, it is shown that the problem is NP-hard. However, a SDP formulation is given for the particular case when the extreme points of the second-stage dual feasible region are explicitly known.

On the same topic, another approach was proposed by Lisser et al. [100]. The key idea is to convert a stochastic program with a discrete distribution of probability into a deterministic one by addition of binary variables. Then the problem becomes a large combinatorial problem, whom the SDP relaxation can be applied to.

3.6 Other applications of interest of SDP

3.6.1 Control theory

There are many applications in control theory that arises naturally as SDP since their constraints can be expressed as LMI. We describe here a very simple case. Suppose that the variable $x = x(t)$ satisfies the following system :

$$\frac{dx}{dt}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t), \quad \left|u_i(t)\right| \leq \left|y_i(t)\right|, i = 1, \cdots, p$$

We want to determine whether $x(t)$ necessarily remains bounded. This holds if and only if there is some P such that $v(x) := x^T P x$ remains uniformly bounded. A sufficient condition for that is that the function v is nonincreasing for any x and u that satisfy the initial system. Such a function is called a *Lyapunov function*.

We want a matrix P such that :

$$\left|u_i(t)\right| \leq \left|C_i x(t)\right|, \forall i \Rightarrow \frac{dV}{dt}(t) = \begin{pmatrix} x(t) \\ u(t) \end{pmatrix}^T \begin{pmatrix} A^T P + PA & PB \\ B^T P & 0 \end{pmatrix} \begin{pmatrix} x(t) \\ u(t) \end{pmatrix} \leq 0$$

By denoting $z = \begin{pmatrix} x(t) \\ u(t) \end{pmatrix}$ and defining the appropriate T_i, this inclusion can be written :

$$\forall z, \ z^t T_i z \leq 0, \forall i \Rightarrow z^t T_0 z \leq 0$$

This is exactly the scope of the S-lemma (Theorem 3.1.2) and therefore a sufficient condition for the validity of this implication is the following :

$$\exists \tau_1, \cdots, \tau_p \text{ such that } \sum_{i=1} \tau_i T_i - T_0 \succcurlyeq 0$$

3.6.2 Minimum rank matrix completion

We are interested in the following problem :

$$\begin{cases} \min & \text{rank}(X) \\ \text{s.t.} & X_{i,j} = a_{i,j}, \ \forall (i,j) \in \Omega \\ & X \in \mathbb{R}^{m,n} \end{cases}$$

The motivation for this problem is to recover a low rank matrix X given a sampling of its entries. It is of great interest in various fields, such as control, statistics and signal processing. Indeed, it frequently happens that the data entries are incomplete, because of errors or noise, or because they are too large to be stored or transmitted entirely.

This problem is nonconvex, NP-hard and might therefore be extremly hard to solve exactly. However, some approximations are tractable, and in particular the most famous of them, based on the *nuclear norm approximation* reduces to a SDP.

Let denote $\sigma(X)$ the vector of singular values of X (see Def. 2.3.14) and consider the approximation consisting of replacing $\text{rank}(X) = \|\sigma(X)\|_0$ by $\|\sigma(X)\|_1$, the sum of the singular values of X, also denoted *nuclear norm* of X.

$$\begin{cases} \min & \|\sigma(X)\|_1 \\ \text{s.t.} & X_{i,j} = a_{i,j}, \forall (i,j) \in \Omega \\ & X \in \mathbb{R}^{m,n} \end{cases} \tag{3.11}$$

Generally it is also intractable, excepts in the case where X is psd, since in this case, the singular values are the eigenvalues of X and $\|\sigma(X)\|_1 = \sum_{i=1}^n \sigma(X)_i = \text{Tr}(X) = I \bullet X$. The following lemma enables to consider a psd matrix instead of an arbitrary matrix X :

Lemma 3.6.1 *Let us consider a matrix* $X \in \mathbb{R}^{m,n}$ *:* $\text{rank}(X) \leq r \Leftrightarrow \exists Y, Z : \begin{cases} \text{rank}(Y) + \text{rank}(Z) \leq 2r \\ \begin{pmatrix} Y & X \\ X^T & Z \end{pmatrix} \succcurlyeq 0 \end{cases}$

By combining these two tricks, we get the following SDP :

$$\begin{cases} \min & 1/2(I \bullet Y + I \bullet Z) \\ \text{s.t.} & X_{i,j} = a_{i,j},\ \forall (i,j) \in \Omega \\ & \begin{pmatrix} Y & X \\ X^T & Z \end{pmatrix} \succcurlyeq 0 \end{cases}$$

Then the optimal value of this SDP is equal to the optimal value of the problem (3.11). For a more detailed discussion on this topic, see for instance [70].

3.6.3 Trust region subproblem

The trust region subproblem concerns the minimization of a possibility non convex quadratic function subject to a norm constraint. This problem is important in non linear programming, for which some algorithms are based on sequential quadratic approximations that are minimized within a *trust region*, i.e., an hypersphere around the current solution, defined by the norm constraint.

A generalization of this problem gives rise to the following problem : $\min q_0(x) : q_1(x) \leq 0$, with q_0 and q_1 two quadratic functions. Then the SDP relaxation of this problem yields the optimal solution, as stated by the S-lemma, provided that $q_1(\bar{x}) < 0$ for some $\bar{x}$ [225]. Another way of understanding why the SDP relaxation is exact comes from the application of the Pataki's rank theorem (see 2.2.14), stating that the extreme points of the feasible set have rank 1. Therefore, $X = xx^T$, which implies that optimal value of the SDP relaxation is feasible for the quadratic program.

In conclusion, the SDP formulation provides a tool for easily solving this problem and the many extensions that arise from the consideration of additional linear or quadratic constraints.

3.6.4 The sensor-network localization problem

The sensor-network localization problem consists of determining the position of a set of sensors that have are deployed in a given area and whose distance toward a certain number of their neighbours is known. The position of a subset of sensors, the so-called anchors, is assumed to be known. Let $x_i \in \mathbb{R}^d,\ i = 1, ..., n$ and $a_i \in \mathbb{R}^d,\ i = 1, ..., m$ the position of the sensors and anchors respectively. We know the distance d_{ij} between sensors i and j for all the pairs of sensors in $\mathcal{N}$, and for all the pair of sensors-anchors in $\mathcal{M}$. Then the objective is to minimize $\sum_{(i,j)\in\mathcal{N}} | \|x_i - x_j\|^2 - d_{ij}^2 | + \sum_{(i,j)\in\mathcal{M}} | \|x_i - a_j\|^2 - d_{ij}^2 |$.

By defining the matrix $X = (x_1, ..., x_n) \in \mathbb{R}^{d,n}$, the problem can be rewritten as following by introducing the convenient matrix $H_{ij} \in \mathbb{S}^{d+n}$:

$$\min \left\{ \sum_{(i,j)\in\mathcal{M}\cup\mathcal{N}} \left| H_{ij} \bullet \begin{pmatrix} I_d & X \\ X^T & Y \end{pmatrix} - d_{ij}^2 \right| : Y = X^T X \right\}$$

This problem is not convex but the relaxation of $Y = X^T X$ into $\begin{pmatrix} I_d & X \\ X^T & Y \end{pmatrix} \succcurlyeq 0$ leads to a SDP [56].

A more complicated variant of this problem exists when the distance are not assumed to be known exactly, but perturbed with random noises, see for instance [50].

3.6.5 Data analysis

A recent and attractive application of SDP is the field of data analysis and more particularly principal component analysis (PCA), which aims at reducing the dimension of a data set by finding the principal

components. This allows to make the data less redundant in order to reveal underlying structure and to facilitate analysis.

More specifically, consider a matrix $X \in \mathbb{R}^{m,n}$ containing the value of n attributes on a sample of m individuals. Then a component is a linear combination of the column of X : $C = \sum j = 1^n u_j X_{*,j} = Xu$ and the principal component is the component that maximize the variance of the sample : $\max u^T \Sigma u$: $\|u\| = 1$ with $\Sigma \in \mathbb{S}^n$ such that $\Sigma_{ij} = \frac{1}{m} X^T X - \bar{X}\bar{X}^T$, with $\bar{X} = \frac{1}{m} X^T e$ the sample mean of X.

This problem is easy to solve by SDP, as a direct application of the trust region subproblem. It becomes more difficult when ones aims at finding *sparse* vector u, i.e., a vector u with many components equal to 0 : $\|u\|_0 \leq k$ with $\|u\|_0 = |\{i \in [n] : u_i \neq 0\}|$. Then it suffices to apply the same approximation as in paragraph 3.6.2, i.e., relax $\|u\|_0 \leq k$ into $\|u\|_1 \leq k$ which is convex. This leads to the following SDP relaxation :

$$\begin{cases} \min & \Sigma \bullet U \\ \text{s.t.} & I \bullet U = 1 \\ & ee^T \bullet |U| \leq k \\ & U \succcurlyeq 0 \end{cases}$$

We refer the reader to [79] for more details on this problem and to [69, 166] for more applications of SDP in the same vein.

3.7 Conclusion

In this chapter, we presented numerous applications of SDP that illustrate its versatility. In order to emphasize the underlying processes that make a SDP to emerge, we started by identifying three main mechanisms for obtaining a SDP constraint :

- by requiring that the variable have one of the properties that define semidefiniteness;
- by applying results relying on the existence of a psd matrix ;
- by requiring that the variable have a very specific structure which induces semidefiniteness.

With this in mind, it comes easily that several classical optimization problems can be written in the form of a SDP. It is also very simple to present how obtaining the standard SDP relaxation of a QCQP and the possible reinforcement of this relaxation comes simply from the addition of valid constraint to the initial QCQP.

We also made a particular focus on the relationship between SDP and the Generalized Problem of Moment (GPM). In particular, the polynomial instances of the GPM can be approximated as closely as desired by the Lasserre hierarchy of SDP relaxations. This is very interesting since this problem subsumes polynomial optimization, combinatorial optimization and some optimization problems under uncertainty.

In the next chapters, we apply these recipes to energy management problems, with a particular emphasis on problems facing uncertainty or combinatorial issues.

Part II

Application of Semidefinite Programming to energy management

« *In theory, theory and practice are the same. In practice, they are different.* »

(A. Einstein)

Table of Contents

Chapter 4

Introduction to energy management

Energy management is an umbrella term for management problems related to the production and consumption of energy. Due to the central importance of energy in our modern industrialised economy, the implications of this subject are considerable, simultaneously of ecological, economic, industrial and social nature. Indeed, the main objective is to save costs, which allows the suppliers to propose energy at best price, in order to enable access to all users and to improve firms' competitiveness. Furthermore, energy management deals with the way energy is produced, transmitted, stored, distributed, transported and consumed, and therefore has impact on the environment and climate via the consumption of non-renewable resource, emission of greenhouse gas, production of nuclar waste, etc... Last but not least, it shall ensure the permanence of the supply, since serious breakdowns have enormous consequences for all the users and must be avoided.

In the expression "energy management", the term "energy" generally refers to electricity and gas. The management of these both commodities shares several characteristics, in particular the objective of satisfying the match between supply and demand by making the best use of an asset portfolio. For EDF R&D, the major difference between these two subjects comes from the fact that the gas portfolio only contains financial assets, whereas electricity portfolio also contains physical assets. Furthermore, gas supply is subject to transportation and storage constraints that are not considered in electricity models. In the sequel, we restrict our attention to problems related to electricity management, even if we may continue to use the term of energy management.

Energy management problems for electricity generally take the form of Unit Commitment Problems, where one aims at deciding which generation units should be running at each period so as to satisfy the demand at least cost, in coordination with optimal management of the financial assets. The specificity of electricity over other commodities is that it does not lend itself well to storage, which induces the constraint of matching the demand at each time step. As the more efficient generation units are generally the less flexible, a very simple strategy for satisfying this constraint consists of turning on in priority these generators, then, when the demand increases, turning on the other generators which can start easily. By efficiency, we mean that the *marginal cost*, i.e., the cost of producing one additional unit, which generally includes fuel and maintenance costs, is low and therefore, this strategy allows to minimize the overall cost of production. However, it is confronted with the fact that certain means of production can only produce a finite amount of energy, called *reserve*. In this case, it is necessary to consider this constraint in the strategy, and one possibility is to replace its marginal cost by a value-in-use, that captures the future profit earned by the saving of one reserve unit.

Energy management problems for electricity differ mainly in the size and sample of the time horizon, the uncertainty representation and how are modelled the generations units. These choices are made according to the targeted decision variables and to operational constraints (availability of the relevant data, upper limit on the resolution time,..) and leads to various optimization problems, depending on the nature of the variables (real, integer, binary/logical) and of the constraints (linear,

piecewise linear, non-linear, non-convex, quadratic,...).

In order to reduce the complexity, a decision process was established at EDF R&D that consists of optimizing from long-term (several years) to short-term (a few hours), in order to exploit the decisions made at larger time horizons, as well as economical and physical indicators computed by the optimization. Three main time horizons are considered, whose associated decisions and modelling are as follows :

- At long-term (the next ten to twenty years), some investment decisions are made, based on long-term impact surveys in which different investment scenarios are simulated and analysed. We deduce from this the main characteristics of the production portfolio: type of power plants, capacity, emission of green house gases...
- At mid-term (the next one to five years), the objective is to schedule the outages of nuclear power plants for refuelling and maintenance, to manage hydro stock and supply contracts, to evaluate and to master physical risk of supply shortage and financial risk on markets.
- At short-term (two weeks to half an hour), it remains to schedule the outages of thermal power plants for maintenance, to evaluate risks, to decide which interruption contracts options are exercised, to schedule the daily generation scheduling satisfying the day ahead forecasted load and respecting all constraints of production units (which thermal and hydraulic power units should be activated and at which level of production) and finally to adapt online the generation schedules at the real load.

Thus, this strategy of time decomposition gives rise to a number of optimization problems, that comply with a wide variety of difficulties and challenges.

In this chapter, we provide a brief overview of the modelling components that are used in the problem that will be considered hereafter. We start by describing the main characteristic of the generation units in the first section, before discussing about the demand in the second section and the demand/supply equilibrium in the third section. Then, we discuss the different ways of representing and handling uncertainty. Finally, in the last section, we describe one of the most challenging energy management problem, namely the nuclear outages scheduling problem (NOSP).

4.1 The demand/supply equilibrium

4.1.1 The supply

In 2012, the physical assets of EDF represented a combined production capacity of 128.7 GW and generated 541 TWh. This global amount breaks down as follows (figures from 2012):

- nuclear power stations (63.1 GW, 74.8% generation) ;
- fossil-fired power stations : coal,fuel-oil and gas (27.8 GW, 8.8% generation) ;
- hydroelectric power stations (25.4 GW, 11.8% generation) ;
- wind, photovoltaic and other renewable power systems (12.4 GW, 4.6% generation).

This physical offer is completed by a financial offer. Indeed, since the opening of the electricity market in 2007, EDF has the possibility of buying and selling electricity on the market. More precisely, this offer takes the form of 3 possible contracts : Futures, Exchange and Interruption Option Contracts (IOC). Futures are standard electricity market contracts, such as day-ahead, week-ahead and week-end-ahead contracts. Exchange contracts define conditions of exchange of electricity: quantity, prices and period. Interruption Option Contracts allow to strongly incite certain customers to interrupt their load charge for the next day in exchange of preferential tariffs for the rest of the year. In addition, EDF has also the possibility of buying and selling electricity on the spot market. The difficulty associated to the optimization of this additional lever stems mainly from the volatility of the spot price, due to the fact that the market depth is limited and the demand inelastic.

Among the physical assets, the thermal generation units are the nuclear and fossil-fired power stations. Their production must satisfy various requirements, related to their maximal capacity of production, their fuel stock or their authorized levels of production and is characterized by costs of production (starting cost, fix and proportional costs, fuel costs ...).

Compared to other thermal generation units, these plants are subject to very specific technical constraints. In particular, maintenance and refueling operations shall be carried out regularly, which leads to frequent shutdowns of the plants. Furthermore, their marginal cost is low but the starting cost is high and it takes a long time to bring it to full power. At the other extreme, fossil-fired power stations can be started up rapidly but their marginal cost is higher. Therefore, nuclear plants operate as baseload, whereas fossil-fuel power plants are used rather for satisfying the peaks of demand. However, one specificity of the French electricity board is that the high proportion of the nuclear generation leads to the necessity of using it also for peak production, which is unique in the world.

The hydraulic park is made up of valleys, i.e., a coherent set of connected hydro reservoirs and production units, characterized by its topology describing connected reservoirs and hydro production units and by a wide variety of constraints such as capacity of turbines, levels of production, bounds constraints issued from policies of exploitation of reservoirs, ... Hydro power offers many advantages over the other energy sources. First of all it is fueled by water. It is therefore a clean renewable power source and its marginal cost is equal to 0. Furthermore, it is very flexible and can be viewed as a storage of power, that is supplied by hydraulic inflows and by a few pump stations. On the other hand, it is renewable but not infinite and shall adapt to water inflows. The key point is therefore to keep this unknown quantity of power for the moment when it is needed the most. This is done by means of an adequate computation of the value-in-use of these reserves, also called water value, which is subsequently used instead of the marginal cost. Finally, the water reserve is also a living, recreational, natural and economic space, which induces several constraints in their management.

The technique for managing reserves is extended to other generation units which are subject to stock constraints, such as nuclear power plants and IOC. Regarding nuclear power, the stock constraint stems from the fact that at short-term, the fuel remaining in the reactor must last until the next outage, which is fixed. Regarding IOC, the number of days concerned by these contracts is finite and stipulated. As with the water reserve, the management of such stocks is made by computing value-in-use that allow to decide the use of these stocks in the present or in the future. This computation is actually the expression of a strategy, that depends on the considered time horizon and shall take uncertainties into account, while optimizing an economic criterion. For one stock, the problem is complex but tractable by dynamic programming. When, we have to define a strategy for the whole stocks, the problem becomes very challenging.

Finally, the production of other sources of renewable energy, such as wind farm or photovoltaic stations, is imposed by the climatic conditions and does not give rise to a short-term management. Related issues concern rather its predictability and the deployment of new power stations.

4.1.2 The demand

The demand or load charge is a time series, expressed in MW, that contains the consumption of all the EDF consumers along the time. Satisfying this demand faces a number of difficulties. First, it fluctuates wildly within the day, the week and the year in response to variation of economic and domestic activities, and of climatic factors such as temperature and nebulosity. This variability gives rise to peak of demand, for which almost all the generation units are requested. On the other hand, during off-peak periods, some generation units, generally the ones with high marginal costs and high flexibility, are cycled down.

The other major difficulty regarding the demand lies in its non predictability in the long or mid-term and in a precise manner in the short-term. Indeed, it is strongly related to economic activities and climatic factors that are difficult to predict on the long term. Furthermore, a part of the production is dedicated for sale on the market and is related to the supply/offer equilibrium of the other European country. Then, predicting the EDF demand requires the forecast of the 3 following elements :

– the French electricity consumption;
– EDF market share;
– the sales volume at interconnections.

Regarding French consumption, it is modelled as a function of various explanatory variables, such as the date or climate parameters. A long term, we consider additional variables of political, economic or technological nature.

In practical terms, over a short-term horizon, the demand is assumed to be known, by combining an inertia effect and weather forecasts. At medium-term, we generally use a finite set of scenarios, elaborated on historical realizations. The time-variability is managed by discretizing the time-horizon in a finite number of time-step on which the demand is assumed to be constant.

4.1.3 Satisfying the demand/supply equilibrium

The specificity of electricity is that it does not lend itself well to storage, which prevents from efficient way of banking energy against a time of sudden demand. As a consequence, it is necessary to produce continuously the amount of electricity that is delivered from the grid.

However, one cannot avoid major incidents due to climatic conditions or failures. These breakdowns can be considered in several ways. In stochastic models, it can be modelled as a probability that the constraint of demand/supply equilibrium is not satisfied. In a Lagrangian penalization spirit, it can also be considered as a generation unit, with a large production capacity and a very high production cost, in order to penalize any resort to this virtual mean of production.

Regarding the management of the demand/supply equilibrium, a very challenging issue is currently emerging. In order to satisfy this requirement, the network manager now has the possibility of inflecting the demand. This is made possible by the installation of a new technology of meter, which allows a real-time adjustment of the peak/off-peak periods. These periods are notified to consumers via different frequency signals, which control the use of certain servo systems, such as hot water tanks and convector heaters. This process allows the postponement of a part of the demand over periods of lowest load, and thus making less use of expensive peak means of production. At the present time, it suffers from the rigidity of the definition of the peak/off-peak periods but the arrival of the new meters, which make this definition flexible, is a lever for improvement and gives rise to new optimization challenges.

4.2 Representing and handling uncertainty

One key difficulty of energy management stems from the fact that a large portion of the data involved in these problems are subject to some degree of uncertainty. This includes the following data :

– the demand and the hydraulic inflows which are very climate sensitive, particularly for temperature and cloud cover ;
– the availability of the production units, subject to random failure and to environmental limitations ;
– the duration of the nuclear outages, that may vary according to various technical incidents ;
– fuel and electricity markets prices;
– wind generation.

Due to the complexity of the underlying processes, the probability distribution of these random variables is generally not available. However, from historical observations, we deduce an estimate of their moments and support. This leads to the following possible representations of uncertainty :

– A deterministic approximation that uses the expected value or the worst-case value;

– A robust representation where the uncertain parameters belongs to a given uncertainty set;
– A distributionnally robust representation where the support and the first k moments of the probability distribution are known;
– A stochastic representation, by considering historical observations or Monte-Carlo simulation as equiprobable scenarios.

Ideally, when the system is dynamic, which means that the realization of uncertain parameters are known over time, the results of the optimization should be a strategy, i.e., a set of decisions which are functions of the past randomness outcomes. This mode is also called *closed-loop strategy*, by contrast to open-loop strategy where the optimization outputs is independent of the uncertain outcomes and therefore correspond to concrete decisions. Closed-loop strategies are clearly preferable but much more difficult to model and solve.

In conclusion, solving energy management problems in an uncertain setting is a very challenging task. The objective are then threefold : to define the most appropriate formulations, to design computationally tractable algorithms, and finally, to qualify the obtained solutions.

4.3 The Nuclear Outages Scheduling Problem (NOSP)

« *Essentially, all models are wrong, but some are useful.* »

(G. Box)

In our thesis we focus on one of the most challenging energy management problem, namely the nuclear outages scheduling problem (NOSP). This problem consists, for a horizon of time of two to five years, of determining the best scheduling for the nuclear outages, i.e., for shutting down the nuclear power plants to proceed to refuelling and maintenance operations, while satisfying the offer-demand equilibrium and the technical constraints at minimal cost. This problem is of the highest importance for EDF because of the major economical stakes that are associated to the nuclear production in France. Furthermore, the cost of an outage may vary considerably according to its scheduling : during winter, an outage may cost twice as much as the same outage during summer. This scheduling is also very important since it has a huge impact on the risk of failure which is strictly controlled.

NOSP can be seen as a variant of the Unit Commitment Problem where a part of the technical constraints follow from the outages. Modelling these constraints requires the use of binary variables, to represent whether the plant is *online* or *offline*. This problem is therefore a huge combinatorial problem. The difficulty is compounded by constraints arising from the limitation of resources used for refuelling operations, which strongly constraint the outages scheduling.

This problem can also be seen as the allocation of the nuclear availability, i.e., the maximal capacity of production of the nuclear park, at the time when it is most needed for the respect the offer-demand equilibrium. On Figure 4.1 is given a small example of nuclear outages scheduling with the corresponding nuclear availability.

This optimization problem is under uncertainty since the demand and certain features of the production facilities are random. The consequences on the design of the model are twofold : first, the question of how is represented the uncertain parameters arises. Second, how precisely shall we model how a nuclear power plant operates. Indeed, it is useless to be very precise in the modelling if the relevant accuracy data is not available. At the present time, this problem is solved by the expectancy method, where the random parameters are replaced by their expected values. The resulting optimization problem is then easy to solve but yields solutions with poor robustness properties.

This section is organized as follows. First we provide a detailed description of the key features of the problem. The Section 2 is devoted to a summary of the different notations, for a better readability.

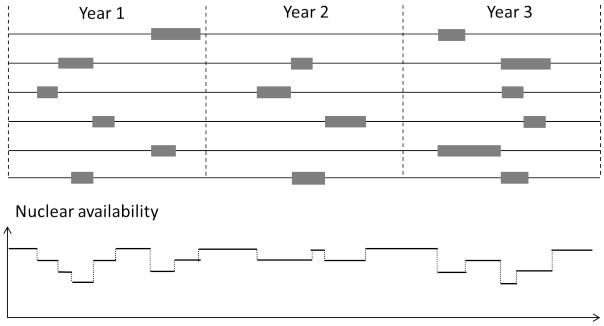

Figure 4.1: A small example of nuclear outages scheduling

Then, the Section 3 proposes different models of this problem, that rely on different simplifications and assumptions, in order to focus on some particular aspects of the problem.

This problem has been studied extensively and we refer the reader to [95, 96, 154, 214] for other references dealing with this problem.

4.3.1 Description of the problem

In this section, we start by describing the considered production facilities, namely the nuclear and the fossil-fuel power plants since the other facilities are not considered in our models. Then we introduce the demand constraint and explicit the notions of peak/off-peak periods. Finally we figure out how to cope with the dynamic nature of the problem.

For easy readability, we start by introducing few notations : the index i stands for the production facilities and we denote by $\mathcal{N}_\nu$, $\mathcal{N}_\theta$ the sets of indices of the nuclear and fossil-fuel power plants respectively. The number of nuclear and fossil-fuel power plants is N_ν and N_θ respectively. t indexes the time and N_t is the number of time steps in the time horizon. In our models, one time step corresponds to one week. Finally, the unit JEPP, used for measuring nuclear generated energy, corresponds to the production of a plant at full power during a whole day. In French it means "Day Equivalent Full Power" and it can be converted in MWh if the maximal power W_i of the plant is known : 1 JEPP equals $24W_i$ MWh. Thus, this amount depends on the concerned plant and therefore we can not combine some quantities of JEPP coming from different plants. The JEPP is also used for the fuel stock, which is measured as the amount of energy that can be produce by the remaining fuel in storage. The other notations are introduced progressively and summarized in Paragraph 4.3.2.

4.3.1.1 Nuclear power plants

The life of a nuclear power plant can be decomposed into cycles. Each cycle consists of a so-called production campaign, followed by an outage, during which the plant is not able to produce. Each outage lasts a given number of weeks denoted $\delta_{i,j-1}$.

The j-th cycle of the plant i is denoted by the pair (i, j), with j lying from 1 to J_i. We denote $\mathcal{C}_\nu = \{(i, j) : i \in \mathcal{N}_\nu, j = 1, ..., J_i\}$. Indeed, we assume that the number of cycles of the time horizon J_i is constant, which means that an outage can not be thrown out of the time horizon.

Note that the last cycle starts during the horizon time but its end (which corresponds to the outage) occurs beyond the horizon time. By convenience we denote $\mathcal{C}^*_\nu = \{(i, j) : i \in \mathcal{N}_\nu, j = 1, ..., J_{i-1}\}$ the set of cycles that have an outage within the time horizon.

Outages The beginning date of the outage (i,j) shall belong to the set $\mathcal{E}_{(i,j)} \subset \{1,...,N_t\}$. We assume that these sets are mutually disjoint : $\mathcal{E}_{(i,j)} \cap \mathcal{E}_{(i,j')} = \emptyset$, for any $(i,j),(i,j') \in \mathcal{C}_\nu, j \neq j'$. In the sequel, we call these sets the *search spaces*.

To each possible beginning date $t \in \mathcal{E}_{(i,j)}$ of each outages $(i,j) \in \mathcal{C}_\nu^*$, we associate a binary variable $x_{i,j,t}$ that is equal to 1 if and only if the outage actually starts at t. Then, as only one date is assigned to each outage, a so-called *assignment constraint* has to be satisfied :

$$\sum_{t \in \mathcal{E}_{(i,j)}} x_{i,j,t} = 1, \quad \forall (i,j) \in \mathcal{C}_\nu^* \tag{4.1}$$

Thus, we are able to give an expression of the outages beginning dates $t_{i,j} = \sum_{t \in \mathcal{E}_{(i,j)}} t x_{i,j,t}, \ \forall (i,j) \in \mathcal{C}_\nu^*$. For convenience, we also define $t_{i,0} = 1$ and $t_{i,J_i} = N_t$.

In the sequel, we may use the notation $x_{i,j,t}$ without specifying $t \in \mathcal{E}_{(i,j)}$, in which case, we simply consider that $x_{i,j,t} = 0$ for any $t \notin \mathcal{E}_{(i,j)}$.

Modulation The ideal operating level of a nuclear power plant is at its maximal capacity. Reducing its production may alter the state of the plant, which requires more maintenance afterwards. For this reason, the time when the plant does not produce at full power shall be limited. In practice, a maximal quantity of *non-production*, called *modulation* is imposed at each cycle. This value is denoted by $M_{i,j}$ and is homogeneous to an amount of energy. If $m_{i,j}$ is the modulation of the cycle (i,j), we have the constraints :

$$m_{i,j} \leq M_{i,j}, \ (i,j) \in \mathcal{C}_\nu \tag{4.2}$$

The notions of cycles, production campaign and modulation are illustrated on Figure 4.2.

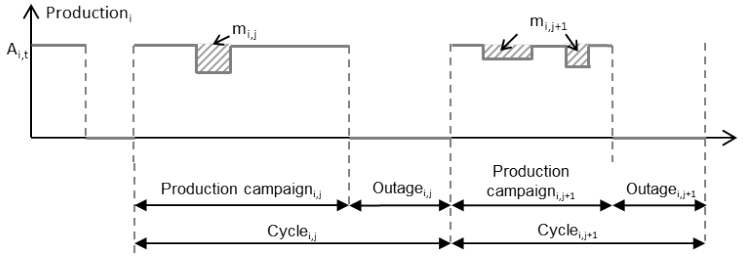

Figure 4.2: Notions of cycles, production campaign and modulation

Availability During their campaigns, nuclear plants can produce up to their maximal power W_i multiplied by a dimensionless stochastic coefficient $A_{i,t}(\xi^\nu) \in [0,1]$, where ξ_i^ν is the stochastic process related to random failures affecting the production of the nuclear power plants. Thus, at each time step t, the production can not exceed $A_{i,t}(\xi^\nu)W_i$.

Besides, the production of the plants vanishes during the outages. The outages variables are used to determine if a plant i is offline or online at time step t, i.e., $\sum_{j:t \in \mathcal{E}_{(i,j)}} x_{i,j,t} = 1$ if and only if the plant is offline at t. We are therefore able to determine the *nuclear availability* at time step t, i.e., the maximal capacity of production of the nuclear park. It depends both on the outages scheduling and on $A_{i,t}(\xi^\nu)$ and is therefore a random state variable :

$$c_t(\xi^\nu) = \sum_{(i,j) \in \mathcal{C}_\nu} A_{i,t}(\xi^\nu) W_i \left(\sum_{t'=t-\delta_{i,j}+1}^{t} x_{i,j,t'} \right) \tag{4.3}$$

We also define $C_t(\xi^\nu) = \sum_{i \in \mathcal{N}_\nu} A_{i,t}(\xi^\nu) W_i$, the maximal nuclear availability of the nuclear park at time step t.

Reload At each cycle (i, j) is associated an amount of reload $r_{i,j}$, expressed in JEPP, that lies within the interval $[\underline{R}_{i,j}, \overline{R}_{i,j}]$. By convention the reload of the cycle (i, j) is carried out during the outage at the end of the cycle and is therefore used for the production of the cycle $(i, j+1)$

Stock The fuel stock of a nuclear plant decreases when the plant produces and rises at each outage as illustrated on Figure 4.3.

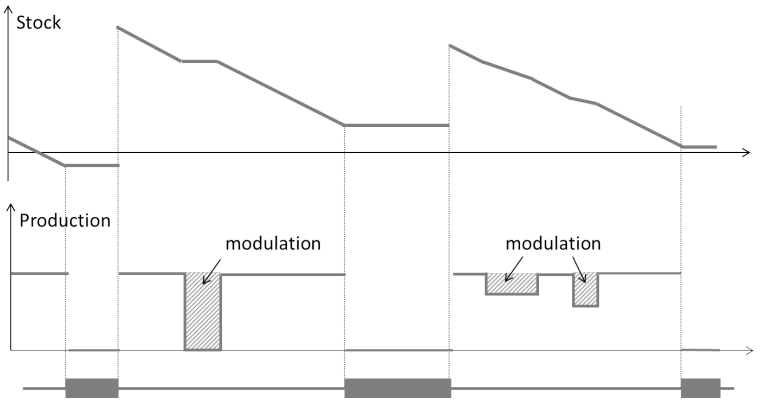

Figure 4.3: Variation of the stock of a plant along the cycles

A specificity of the nuclear power plants is that only a part (usually a third or a quarter) of the nuclear fuel is unloaded at each outage. We denote $\beta_{i,j}$ the part that is not unloaded at outage (i, j), i.e.,the part that stays in the reactor during the reloading operations. As a consequence, the beginning stock of a cycle is equal to the amount of the reload plus the part $\beta_{i,j-1}$ of the ending stock of the previous cycle.

Another specificity of the nuclear fuel stock is that the energy that it contains tends to zero but never vanishes. Therefore, we decide by convention that a given level of stock is zero, which implies that the stock becomes negative when the plant continues to produce beyond this level. This also means that as soon as the stock passes through this level, we consider that we consume a part of the stock of the following cycle.

To compute the ending stock of the cycle (i, j), we use a flow equation, stating that it is equal to the beginning stock minus the stock used for production. By definition of a JEPP, if a plant i produces at full power (W_i) during all its cycle, then the stock used for production, is equal to the duration of the cycle in days. During the campaign of the cycle $(i, j+1)$, the plant produces at $A_{i,t}(\xi^\nu)W_i$ at time t and by subtracting the modulation achieved throughout the cycle, it comes that the stock used for production equals $7 \sum_{t=t_{i,j}+\delta_{i,j}}^{t_{i,j+1}} A_{i,t}(\xi^\nu) - m_{i,j+1}$

Regarding the beginning stock, it can be computed as the sum of amount of reload of the previous cycle and of the part of the final stock of the previous cycle that is not unloaded. Finally, the ending stock, a random variable that it depends of the availability of the plant anddenoted by $f_{i,j}(\xi_\nu)$, can be computed as follows :

$$f_{(i,j+1)}(\xi_\nu) = 7 \sum_{t=t_{i,j}+\delta_{i,j}}^{t_{i,j+1}} A_{i,t}(\xi^\nu) - m_{i,j+1} + r_{i,j} + \beta_{i,j} f_{i,j}, \ \forall (i, j+1) \in \mathcal{C}_\nu^* \tag{4.4}$$

For $j = 0$, we replace $r_{i,j} + \beta_{i,j} f_{i,j}$ by the initial stock of the plant, which is a data of the problem.

For safety reasons, reloading operations can not occur when the reactivity of the core is too high. This yields an upper bound $\overline{F}_{i,j}$ on the ending stock of a cycle. A lower bound $\underline{F}_{i,j}$ is also given :

$$f_{(i,j)}(\xi_\nu) \in [\underline{F}_{i,j}, \overline{F}_{i,j}], \ \forall (i,j) \in \mathcal{C}^*_\nu \tag{4.5}$$

Resources constraint The nuclear power plants are scattered over N_g geographical sites and we denote $\mathcal{J}_k$ the set of plants of the site k.

On each of these sites, some resources required for maintenance and reloading operations are shared among the plants, which imposes constraints between the different outages. First, it is impossible to have more than N^p_k ongoing outages over $\mathcal{J}_k$ at each time step t :

$$\sum_{i \in \mathcal{J}_k} \sum_{j=1}^{J_i} \sum_{t'=t-\delta_{i,j}+1}^{t} x_{i,j,t'} \leq N^p_k, \ t = 1, ..., N_t, \ k = 1, ..., N_g \tag{4.6}$$

Furthermore, the outages of $\mathcal{J}_k$ have to preserve a minimal space N^l_k between them, or a maximal lapping if $N^l_k < 0$, which can be formulated as follows :

$$t_{i,j} - t_{i',j'} \notin \ ]-N^l_k - \delta_{i,j}, \ N^l_k + \delta_{i',j'}[, \ (i,j),(i',j') \text{ with } i, i' \in \mathcal{J}_k, \ k = 1, ..., N_g \tag{4.7}$$

Indeed, let us consider two outages $1 = (i,j)$ and $2 = (i',j')$ of $\mathcal{J}_k$ such that $\delta_1 \leq \delta_2$. Then the lapping $\Delta = \max\{t_1, \ t_2\} - \min\{t_1 + \delta_1, \ t_2 + \delta_2\}$ and three configurations are possible, as illustrated on the Figure 4.4 :

- $t_1 \leq t_2$ and $t_1 + \delta_1 \leq t_2 + \delta_2$, then $\Delta = t_2 - t_1 - \delta_1$;
- $t_1 \geq t_2$ and $t_1 + \delta_1 \geq t_2 + \delta_2$, then $\Delta = t_1 - t_2 - \delta_2$;
- $t_1 \geq t_2$ and $t_1 + \delta_1 \leq t_2 + \delta_2$, then $\Delta = -\delta_1$;

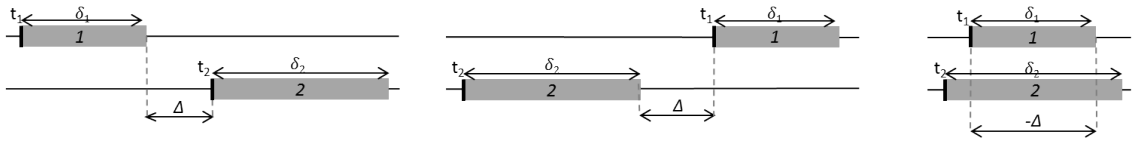

Figure 4.4: 3 possibles configurations for computing the lapping between 2 outages

Since $\delta_1 \leq \delta_2$, $t_1 \leq t_2 \Rightarrow t_1 + \delta_1 \leq t_2 + \delta_2$ and $t_1 \leq t_2$ is sufficient for the first case. In the same way, $t_1 + \delta_1 \geq t_2 + \delta_2$ is sufficient for the third case. As a consequence, the three configurations become :

- $t_1 \leq t_2$, then $\Delta = t_2 - t_1 - \delta_1$;
- $t_1 \geq t_2$ and $t_1 + \delta_1 \leq t_2 + \delta_2$, then $\Delta = -\delta_1$;
- $t_1 + \delta_1 \geq t_2 + \delta_2$, then $\Delta = t_1 - t_2 - \delta_2$;

This leads to variation of Δ represented on Figure 4.5.

From examining this curve, we observe that if $N^l_k \leq -\delta_1$, then the constraint is necessarily satisfied. Otherwise, we recover the formulation proposed in (4.7).

This constraint presents modelling difficulty due to its nonconvex and disjunctive nature. A first possibility for modelling it relies on the fact that the number of combinations of dates that violates this constraint is finite, and therefore, it suffices to forbid all these combinations :

$$x_{1,t} + x_{2,t'} \leq 1 \text{ for all } t, t' \text{ such that } t - t' \in \ ]-N^l_k - \delta_2, \ N^l_k + \delta_1[\tag{4.8}$$

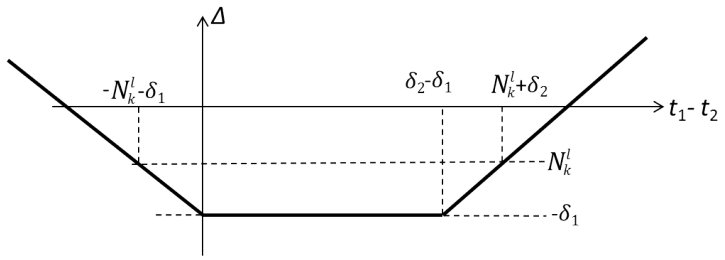

Figure 4.5: The Δ variation as a function of $t_1 - t_2$

This formulation, referred to as *pairwise exclusion*, is efficient but leads to a possibly very large number of constraints.

Another possibility, called "big M" formulation, requires the introduction of a new binary variable, that equals 0 if $t_1 - t_2 \leq -N_k^l - \delta_2$ and 1 if $t_1 - t_2 \geq N_k^l + \delta_1$. Let z be this variable :

$$\begin{cases} t_1 - t_2 \leq -N_k^l - \delta_2 + M_1 z \\ t_1 - t_2 \geq N_k^l + \delta_1 - M_2(1 - z) \\ z \in \{0, 1\} \end{cases} \tag{4.9}$$

The best values for M_1 and M_2 can easily be computed by using the maximal and minimal values of $t_1 - t_2$.

The third formulation is quadratic. Indeed, since it is impossible to have both $t_1 - t_2 < -N_k^l - \delta_1$ and $t_1 - t_2 > N_k^l - \delta_2$, the constraints is equivalent to :

$$(t_1 - t_2 + N_k^l + \delta_1)(t_1 - t_2 - N_k^l - \delta_2) \geq 0 \tag{4.10}$$

In conclusion, let us remark that for some pairs of cycles (i, j) and (i', j'), the sets $\mathcal{E}_{(i,j)}$ and $\mathcal{E}_{(i',j')}$ are such that only one part of the disjunction is possible. In this case, the constraint is a simple linear constraint, for instance if $t_{i',j'} - t_{i,j} \leq -N_k^l - \delta_{i',j'}$ can not happen, then the constraint becomes $t_{i',j'} - t_{i,j} \geq N_k^l + \delta_{i,j}$.

Production cost The production cost of the nuclear power plants is proportional to the amount of fuel reloaded, with a proportionality coefficient $\gamma_{i,j}$. We deduce from this the cost associated to the stock remaining at the end of the time horizon, that has not been consumed and therefore shall not be paid. Overall, the nuclear production cost equals $\sum_{(i,j)\in\mathcal{C}_\nu^*} \gamma_{i,j} r_{i,j} + \sum_{i=1}^{N_\nu} \gamma_{i,J_i-1} f_{i,J_i}(\xi_\nu)$.

4.3.1.2 Fossil-fuel power plants

The only constraint imposed on the production of the fossil-fuel power plants is the constraint of maximal production. In the same way as for the nuclear power plants, the production of the plant i is characterized by a maximal production W_i and a stochastic availability coefficient $A_{i,t}(\xi^\theta)$. Therefore, it shall not exceed the quantity $A_{i,t}(\xi^\theta)W_i$.

The fossil-fuel production carries a cost that is proportional to the amount of production, with a proportionality coefficient $\gamma_{i,t}^\theta$. To minimize the global cost, as there is not any constraint imposed on these facilities, we give priority to the less expensive one. Consequently, the cost of a fossil-fuel production p_t follows a convex piecewise linear curve as shown at Figure 4.6.

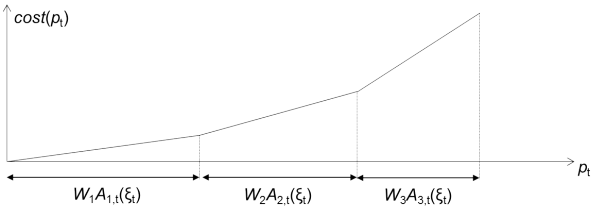

Figure 4.6: Fossil-fuel production cost

4.3.1.3 The demand

In reality, the demand to satisfy is a continuous function of the time, but for simulation purpose, we discretize each time step t into several periods and we assume that the demand is constant over each period. These periods can be grouped in two categories :

- Off-peak periods, when the demand is low (for example, during the night)
- Peak periods, when the demand is high (for example, during the evening)

For each time step, the peak and off-peak time step are gathered, so that each time steps contains only two periods, a peak and an off-peak one. Consequently, the demand constraint is twofold : the peak and the off-peak one, as illustrated on Figure 4.7.

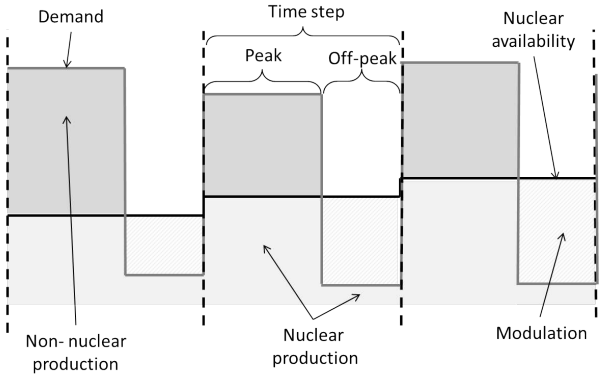

Figure 4.7: The demand for the Nuclear Outages Scheduling Problem

On the one hand, on peak periods, the production have to satisfy the peak load denoted by $D_t(\xi^\delta)$. Two simplifications are made concerning these periods : we assume that this demand is larger than the nuclear availability and that the nuclear production is maximal. Consequently, we do not need a variable for the nuclear production since we can use the nuclear availability $c_t(\xi^\nu)$ instead. The peak demand constraint can therefore be formulated as follows :

$$c_t(\xi^\nu) + \sum_{i \in \mathcal{N}_\theta} W_i A_{i,t}(\xi^\theta) y_{i,t} \geq D_t(\xi^\delta),\ t = 1, ..., N_t \tag{4.11}$$

where $y_{i,t} \in [0, 1]$ is the production of the fossil-fuel plant i at time t.

On the other hand, during off-peak periods, the off-peak demand has to be satisfied. In the same way as for peak time step, we make the following assumption : the off-peak demand is lower than the nuclear availability, and the fossil-fuel production vanishes. It is tantamount to assuming that the only production during off-peak period is nuclear.

Thus, at each off-peak period, the nuclear power plants have to satisfy a given level of demand. Another way of expressing this is as follows : given the nuclear availability, a certain level of *non-production* is allowed. As mentioned below, this amount is called *modulation*. The fact that the system aims at minimizing its production cost guarantees that the whole modulation will be used.

We make an additional simplification : instead of having an authorized amount of modulation per time step, we gather them in a global amount of modulation to use throughout the time horizon. This value, $\overline{M}(\xi^\mu)$, is a random variable expressed in MWh (not in JEPP since it concerns all the plants), which we assume is independent on the problem variables. Then, the off-peak constraint writes :

$$24 \sum_{(i,j)\in\mathcal{C}_\nu} W_i m_{i,j} \leq \overline{M}(\xi^\mu) \tag{4.12}$$

where the multiplication by $24W_i$ is required to convert the modulation $m_{i,j}$ in JEPP into MWh.

4.3.1.4 A dynamic problem

A key difficulty of the nuclear outages scheduling problem comes from its dynamic nature, which means that certain decisions can be made once uncertainty is removed. It is typically the case of the fossil-fuel production $y_{i,t}$ since in practice this quantity is determined once the demand and the availability of the production facilities are known, so that the supply meet the demand. On the other hand, the other decisions variables are *static*, i.e., they are made once and for all at the beginning of the time horizon. This context would require a closed-loop optimization but unfortunately, this may not be possible to model, depending on which representation of uncertainty is chosen.

With a discrete representation of uncertainties (multi-scenario), it suffices to define one dynamic variable per scenario, which can be very costly in terms of number of variables. With a stochastic representation of uncertainty, the problem falls in the framework of multi-stages programming, a well-defined paradigm but very hard to solve. Extensions to robust representation also exist. However, in our problem, we have the possibility to proceed differently, by removing the fossil-fuel production variables y from the model. Indeed, these variables y are involved only in the peak-demand constraint and in the objective function, and we will see that for these two terms, there are equivalent or approximated formulations that do not involve y.

Regarding the peak-demand constraint, it is strictly equivalent to replace the fossil-fuel production by the fossil-fuel maximal production, because the maximal production of the plants is the only constraint that the variables y must satisfy. Indeed, the peak demand constraint is satisfied if and only if the availability of the various production facilities is greater than the demand. This allows us to remove the fossil-fuel production variable from this constraint, as expressed below :

$$c_t(\xi^\nu) + \sum_{i=1}^{N_\theta} y_{i,t} A_{i,t}(\xi^\theta) W_i \geq D_t(\xi^\delta) \text{ for some } y_{i,t} \in [0,1] \Leftrightarrow c_t \geq D_t(\xi^\delta) - \sum_{i=1}^{N_\theta} A_{i,t}(\xi^\theta) W_i$$

It remains to express the cost of this production. Recall that if p_t is the fossil-fuel production at time t, then its cost is $f(p_t)$, with f a convex piecewise linear function. The difficulty is that the function f depends on the stochastic parameters $A_i(\xi^\theta$. Furthermore, implementing such a function in our model requires the definition of one continuous variable by pieces, which brings back to the starting point.

A way to overcome this difficulty is to approximate f by a deterministic quadratic function q_t be this function, then the objective becomes :

$$\min q_t(D_t(\xi^\delta) - c_t(\xi^\nu)) \tag{4.13}$$

Thus, the objective function does not involve the variables y any more.

4.3.2 Notations summary

In the above table (Table 4.1), we summarize all the notations used to present the NOSP. The first part of the table contains the variables while the second lists the notations related to the data :

$x_{i,j,t}$	binary variable, equal to 1 if and only if the outage (i,j) starts at t
$c_t(\xi^\nu)$	nuclear availability at time t
$m_{i,j}$	modulation of the cycle (i,j)
$r_{i,j}$	reload of the cycle (i,j)
p_t	fossil-fuel production at time t
$t_{i,j}$	beginning date of the outage of the cycle (i,j)
$f_{i,j}(\xi_\nu)$	ending stock of the cycle (i,j)
N_t	number of time steps during the time horizon
N_ν	number of nuclear units
N_θ	number of fossil-fuel units
N_g	number of nuclear geographical sites
N_s	number of scenarios
$\mathcal{C}_\nu$	set of nuclear cycles
$\mathcal{C}_\nu^*$	set of nuclear cycles with outage inside the time horizon
J_i	index of the last cycle of the nuclear unit i
$\mathcal{E}_{(i,j)}$	set of possible beginning dates of the outage (i,j)
$\delta_{i,j}$	duration of the outage (i,j)
W_i	maximal power of the (nuclear or fossil-fuel) power plant i
$A_{i,t}(\xi^\nu)$	stochastic coefficient of failure of the nuclear power plant i at time t
$C_t(\xi^\nu)$	maximal availability of the nuclear park at time step t
$A_{i,t}(\xi^\theta)$	stochastic coefficient of failure of the fossil-fuel power plant i at time t
$M_{i,j}$	maximal modulation of cycle (i,j)
$[\underline{R}_{i,j}, \overline{R}_{i,j}]$	interval of possible reload for the cycle (i,j)
$\beta_{i,j}$	the part of the fuel that is unload at each outage
$[\underline{F}_{i,j}, \overline{F}_{i,j}]$	interval of the possible ending stock of the cycle (i,j)
$\mathcal{J}_k$	set of nuclear power plants located on the site k
N_k^l	maximal number of parallel outages on the site k
N_k^p	minimal space between two outages on the site k
$\gamma_{i,j}$	proportional cost of the reload of the cycle (i,j)
$\gamma_{i,t}^\theta$	proportional cost of the fossil-fuel power plant i at time t
$\underline{D}_t(\xi^\delta)$	peak load at time t
$\overline{M}(\xi^\mu)$	global amount of modulation to achieve
q_t	quadratic function that associates a cost to the amount of fossil-fuel production at time t

Table 4.1: Notations used for the Nuclear Outages Scheduling Problem

4.3.3 The models

In order to emphasize the combinatorial and stochastic nature of the problem, some additional simplifications are introduced, which leads to different models. We present them in order of how close to the real problem they are.

4.3.3.1 Model 1

This model is a direct application of the description given at Section 4.3.1 with the quadratic objective function and the quadratic formulation of the maximal lapping constraint. Regarding uncertainty, we use a stochastic approach and we require that the constraints involving uncertain parameters be satisfied up to a given level of probability (chance-constraints). This leads to the following stochastic problem :

$$
\left\{
\begin{array}{lll}
\min & \mathrm{E}[\sum_{t=1}^{N_t} q_t(D_t(\xi^\delta) - c_t(\xi^\nu)) - \sum_{i=1}^{N_\nu} \gamma_{i,J_i-1} f_{i,J_i}(\xi_\nu)] + \sum_{(i,j)\in\mathcal{C}_\nu^*} \gamma_{i,j} r_{i,j} & (4.13) \\
\text{s.t.} & \mathrm{P}[\underline{F}_{i,j} \leq f_{i,j}(\xi_\nu) \leq \overline{F}_{i,j}] \geq 1-\varepsilon,\ \forall (i,j) \in \mathcal{C}_\nu^* & (4.5) \\
& \mathrm{P}[24 \sum_{(i,j)\in\mathcal{C}_\nu} W_i m_{i,j} \leq \overline{M}(\xi^\mu)] \geq 1-\varepsilon & (4.12) \\
& \sum_{i\in\mathcal{J}_k} \sum_{j=1}^{J_i} \sum_{t'=t-\delta_{i,j}+1}^{t} x_{i,j,t'} \leq N_k^p,\ t \in [N_t],\ k \in [N_g] & (4.6) \\
& \sum_{t\in\mathcal{E}_{(i,j)}} x_{i,j,t} = 1,\ \forall (i,j) \in \mathcal{C}_\nu^* & (4.1) \\
& (t_{i,j} - t_{i',j'} + N_k^l + \delta_{i,j})(t_{i,j} - t_{i',j'} - N_k^l - \delta_{i',j'}) \geq 0, & \\
& \qquad \forall i,j,i',j' : i \neq i' \in \mathcal{J}_k,\ k \in [N_g] & (4.10) \\
& f_{(i,j+1)}(\xi_\nu) = 7 \sum_{t=t_{i,j}+\delta_{i,j}}^{t_{i,j+1}} A_{i,t}(\xi^\nu) - m_{i,j+1} + r_{i,j} + \beta_{i,j} f_{i,j},\ \forall (i,j+1) \in \mathcal{C}_\nu^* & (4.4) \\
& c_t(\xi^\nu) = \sum_{(i,j)\in\mathcal{C}_\nu} A_{i,t}(\xi^\nu) W_i \left(\sum_{t'=t-\delta_{i,j}+1}^{t} x_{i,j,t'} \right) & (4.3) \\
& t_{i,j} = \sum_{t\in\mathcal{E}_{(i,j)}} t x_{i,j,t},\ \forall (i,j) \in \mathcal{C}_\nu^* & \\
& m_{i,j} \in [0, M_{i,j}],\ \forall (i,j) \in \mathcal{C}_\nu & (4.2) \\
& r_{i,j} \in [\underline{R}_{i,j}, \overline{R}_{i,j}],\ \forall (i,j) \in \mathcal{C}_\nu^* & \\
& x_{i,j,t} \in \{0,1\},\ \forall t \in \mathcal{E}_{i,j},\ \forall (i,j) \in \mathcal{C}_\nu^* &
\end{array}
\right. \tag{4.14}
$$

We assume that the probability distributions of ξ^δ, ξ^θ and ξ^ν are concentrated on a finite number of scenarios $s = 1, ..., N_s$, obtained from historical observation. We assume that all these scenarios have the same probability $1/N_s$ of occurrence. Then we can derive a deterministic formulation of the constraints related to the ending stock (4.5) and to the off-peak constraint (4.12). Recall that the peak demand constraint is implicit since the fossil-fuel production variables have been removed.

This formulation is very classical in stochastic programming and is obtained as follows : for each considered constraint and each scenario, we introduce a binary variable that must be equal to 0 if the constraint is satisfied. Then, it suffices to impose that a minimal number of these variables are equal to 0. More precisely, let us consider the following joint chance-constraint :

$$\mathrm{P}[a_i(\xi)^T x \leq b_i(\xi),\ i = 1, ..., m] \geq 1 - \varepsilon$$

where $a_i(\xi)$ and $b_i(\xi)$ are random vectors and variable for $i = 1, ..., m$, represented by their respective N_s scenarios $\{a_{i,s}\}_{s=1,\cdots,N_s}$ and $\{b_{i,s}\}_{s=1,\cdots,N_s}$. π_s is the probability of occurrence of the scenario s and x is the decision vector. Then the constraint can be formulated as :

$$
\left\{
\begin{array}{l}
a_{i,s}^T x - M z_s \leq b_{i,s},\ s = 1, ..., N_s,\ i = 1, ..., m \\
\sum_{s=1}^{N_s} \pi_s (1 - z_s) \geq 1 - \varepsilon \\
z_s \in \{0,1\},\quad s = 1, ..., N_s
\end{array}
\right.
$$

where M is a positive scalar, large enough to guarantee that the inequalities $a_{i,s}^T x \leq b_{i,s} + M,\ s = 1, ..., N_s,\ i = 1, ..., m$ hold for any feasible x.

In our model, the off-peak demand constraint leads to an individual chance-constraint and the ending stock constraints give rise to one range chance-constraints (i.e., $m = 2$ and $a_1(\xi) = -a_2(\xi)$) for each cycle $(i, j) \in \mathcal{C}^*_\nu$.

Regarding the objective, we aim at minimizing its expected value, computed as follows the sum of the objective value of each scenario, multiplied by $1/N_s$. With this formulation, the fact that the coefficient of the quadratic function vary according to the scenario is not problematic since it suffices to use the relevant coefficients for each scenario.

Finally, we obtain a MIQCQP, with a large number of binary variables and linear constraint, where the quadratic term involves only binary variables.

4.3.3.2 Model 2

This model is the deterministic version of the problem described at Section 4.3.1, with the quadratic objective function and the "big M" formulation of the maximal lapping constraint.

Thus, it is almost similar to the model (4.14), by replacing the constraint (4.10) by the constraint (4.9) and by considering the deterministic version of the stochastic parameters :

$$
\left\{
\begin{array}{lll}
\min & \sum\limits_{t=1}^{N_t} q_t(D_t - c_t) - \sum\limits_{i=1}^{N_\nu} \gamma_{i,J_i-1} f_{i,J_i} + \sum\limits_{(i,j)\in\mathcal{C}^*_\nu} \gamma_{i,j} r_{i,j} & (4.13)\\
\text{s.t.} & \underline{F}_{i,j} \leq f_{i,j} \leq \overline{F}_{i,j},\ \forall (i,j) \in \mathcal{C}^*_\nu & (4.5)\\
& 24 \sum\limits_{(i,j)\in\mathcal{C}_\nu} W_i m_{i,j} \leq \overline{M} & (4.12)\\
& \sum\limits_{i\in\mathcal{J}_k} \sum\limits_{j=1}^{J_i} \sum\limits_{t'=t-\delta_{i,j}+1}^{t} x_{i,j,t'} \leq N^p_k,\ t \in [N_t],\ k \in [N_g] & (4.6)\\
& \sum\limits_{t\in\mathcal{E}_{(i,j)}} x_{i,j,t} = 1,\ \forall (i,j) \in \mathcal{C}^*_\nu & (4.1)\\
& t_{i,j} - t_{i',j'} \leq -N^l_k - \delta_{i',j'} + M_{i,j} z_{i,j,i',j'},\ \forall i,j,i',j' : i \neq i' \in \mathcal{J}_k,\ k \in [N_g] & (4.9)\\
& t_{i,j} - t_{i',j'} \geq N^l_k + \delta_{i,j} - M_{i',j'}(1 - z_{i,j,i',j'}),\ \forall i,j,i',j' : i \neq i' \in \mathcal{J}_k,\ k \in [N_g] & (4.9)\\
& f_{(i,j+1)} = 7 \sum\limits_{t=t_{i,j}+\delta_{i,j}}^{t_{i,j+1}} A_{i,t} - m_{i,j+1} + r_{i,j} + \beta_{i,j} f_{i,j},\ \forall (i,j+1) \in \mathcal{C}^*_\nu & (4.4)\\
& c_t = \sum\limits_{(i,j)\in\mathcal{C}_\nu} A_{i,t} W_i \left(\sum\limits_{t'=t-\delta_{i,j}+1}^{t} x_{i,j,t'} \right) & (4.3)\\
& t_{i,j} = \sum\limits_{t\in\mathcal{E}_{(i,j)}} t x_{i,j,t},\ \forall (i,j) \in \mathcal{C}^*_\nu & \\
& m_{i,j} \in [0, M_{i,j}],\ \forall (i,j) \in \mathcal{C}_\nu & (4.2)\\
& r_{i,j} \in [\underline{R}_{i,j}, \overline{R}_{i,j}],\ \forall (i,j) \in \mathcal{C}^*_\nu & \\
& x_{i,j,t} \in \{0,1\},\ \forall t \in \mathcal{E}_{i,j},\ \forall (i,j) \in \mathcal{C}^*_\nu & \\
& z_{i,j,i',j'} \in \{0,1\},\ \forall i,j,i',j' : i \neq i' \in \mathcal{J}_k,\ k \in [N_g] &
\end{array}
\right. \tag{4.15}
$$

This model is dedicated to study the combinatorial aspect of the problem. The obtained problem has mixed variables (binary and continuous), linear constraints and a convex quadratic objective. A key point is that the quadratic terms involves only binary variables.

4.3.3.3 Model 3

We elaborated this model in order to test sophisticated SDP relaxations. To this end, we need to reduce the size of the problem and in particular, we aim at removing the continuous variables in order to focus on the combinatorial aspect of the problem.

The major simplification that we make then is to replace the constraint on the ending stock and the constraint on maximal modulation by a constraint that imposes a time interval between successive outages of the same plant :

$$\underline{T}_{(i,j)} \leq t_{i,j} - t_{i,j-1} \leq \bar{T}_{(i,j)}, \quad \forall (i,j) \in \mathcal{C}_\nu \tag{4.16}$$

As a consequence, the definition of the final stock $f_{i,j}$ is not required any more. Furthermore, we neglect the reload $r_{i,j}$ and the associated cost, as well as the modulation $m_{i,j}$ and the off-peak demand constraint (4.12).

The rest is similar to the description given at Section 4.3.1, with the following precisions : we are in a deterministic framework and the objective is quadratic, as well as the formulation of the maximal lapping constraint.

Finally, we obtain a pure binary problem, with quadratic objective and constraints, whose linear constraints are both equality and inequality constraints. We summarize it as follows :

$$\left\{\begin{array}{lll}
\min & \sum_{t=1}^{N_t} q_t(D_t - c_t) & (4.13) \\
\text{s.t.} & \underline{T}_{(i,j)} \leq t_{i,j} - t_{i,j-1} \leq \bar{T}_{(i,j)}, \quad \forall (i,j) \in \mathcal{C}_\nu & (4.16) \\
& \sum_{i \in \mathcal{J}_k} \sum_{j=1}^{J_i} \sum_{t'=t-\delta_{i,j}+1}^{t} x_{i,j,t'} \leq N_k^p, \ t \in [N_t], \ k \in [N_g] & (4.6) \\
& \sum_{t \in \mathcal{E}_{(i,j)}} x_{i,j,t} = 1, \ \forall (i,j) \in \mathcal{C}_\nu^* & (4.1) \\
& (t_{i,j} - t_{i',j'} + N_k^l + \delta_{i,j})(t_{i,j} - t_{i',j'} - N_k^l - \delta_{i',j'}) \geq 0, & \\
& \qquad \forall i,j,i',j' : i \neq i' \in \mathcal{J}_k, \ k \in [N_g] & (4.10) \\
& c_t = \sum_{(i,j) \in \mathcal{C}_\nu} A_{i,t} W_i \left(\sum_{t'=t-\delta_{i,j}+1}^{t} x_{i,j,t'} \right) & (4.3) \\
& t_{i,j} = \sum_{t \in \mathcal{E}_{(i,j)}} t x_{i,j,t}, \ \forall (i,j) \in \mathcal{C}_\nu^* & \\
& x_{i,j,t} \in \{0,1\}, \ \forall t \in \mathcal{E}_{i,j}, \ \forall (i,j) \in \mathcal{C}_\nu^* &
\end{array}\right. \tag{4.17}$$

4.3.3.4 Model 4

The model we describe in this section is not really a simplification, rather a focus on a very precise part of the nuclear outages scheduling problem, namely the maximal lapping constraint. To this end, we consider only one outage per nuclear power plants and the only constraints to satisfy are :

- the assignment constraint (4.1);
- the maximal lapping constraint (4.7).

Regarding the maximal lapping constraint, the three possible formulations will be considered, i.e., "big M", pairwise exclusion and quadratic formulations.

Finally, the objective is defined via a bunch of functions q_t, as follows : $\sum_{t=1}^{N_t} q_t(D_t - c_t)$. In concert with the other models, the functions q_t should be quadratic, however for this study, we consider that it is linear. Indeed, the quadratic objective reduces the tightness of the maximal lapping constraints since it tends to spread the outages all over the time horizon. Furthermore, a large part of the gap comes from the linearization of the objective function. Consequently, we obtain the same gap for the 3 models of the maximal lapping constraints, which is not interesting for our study. By contrast, using a linear objective yields gaps that differ significantly among the different models.

We start with the model 4−1 that consider the "big M" version of the maximal lapping constraint :

$$\begin{cases} \min & \sum_{t=1}^{N_t} q_t(D_t - c_t) & (4.13) \\ \text{s.t.} & \sum_{t \in \mathcal{E}_{(i,j)}} x_{i,j,t} = 1, \ \forall (i,j) \in \mathcal{C}_\nu^* & (4.1) \\ & t_{i,j} - t_{i',j'} \leq -N_k^l - \delta_{i',j'} + M_{i,j} z_{i,j,i',j'}, \ \forall i,j,i',j' : i \neq i' \in \mathcal{J}_k, \ k \in [N_g] & (4.9) \\ & t_{i,j} - t_{i',j'} \geq N_k^l + \delta_{i,j} - M_{i',j'}(1 - z_{i,j,i',j'}), \ \forall i,j,i',j' : i \neq i' \in \mathcal{J}_k, \ k \in [N_g] & (4.9) \\ & c_t = \sum_{(i,j) \in \mathcal{C}_\nu} A_{i,t} W_i \left(\sum_{t'=t-\delta_{i,j}+1}^{t} x_{i,j,t'} \right) & (4.3) \\ & t_{i,j} = \sum_{t \in \mathcal{E}_{(i,j)}} t x_{i,j,t}, \ \forall (i,j) \in \mathcal{C}_\nu^* & \\ & x_{i,j,t} \in \{0,1\}, \ \forall t \in \mathcal{E}_{i,j}, \ \forall (i,j) \in \mathcal{C}_\nu^* & \\ & z_{i,j,i',j'} \in \{0,1\}, \ \forall i,j,i',j' : i \neq i' \in \mathcal{J}_k, \ k \in [N_g] & \end{cases} \tag{4.18}$$

Then, the model 4−2 uses the pairwise exclusion formulation of the maximal lapping constraint :

$$\begin{cases} \min & \sum_{t=1}^{N_t} q_t(D_t - c_t) & (4.13) \\ \text{s.t.} & \sum_{t \in \mathcal{E}_{(i,j)}} x_{i,j,t} = 1, \ \forall (i,j) \in \mathcal{C}_\nu^* & (4.1) \\ & x_{i,j,t} + x_{i',j',t'} \leq 1, \ \forall t,t' : t - t' \in \ ]-N_k^l - \delta_{i',j'}, \ N_k^l + \delta_{i,j}[, & \\ & \qquad \forall i,j,i',j' : i \neq i' \in \mathcal{J}_k, \ k \in [N_g] & (4.8) \\ & c_t = \sum_{(i,j) \in \mathcal{C}_\nu} A_{i,t} W_i \left(\sum_{t'=t-\delta_{i,j}+1}^{t} x_{i,j,t'} \right) & (4.3) \\ & t_{i,j} = \sum_{t \in \mathcal{E}_{(i,j)}} t x_{i,j,t}, \ \forall (i,j) \in \mathcal{C}_\nu^* & \\ & x_{i,j,t} \in \{0,1\}, \ \forall t \in \mathcal{E}_{i,j}, \ \forall (i,j) \in \mathcal{C}_\nu^* & \end{cases} \tag{4.19}$$

Finally, with the quadratic formulation of the maximal lapping constraint, we obtain the model $4-3$:

$$\begin{cases} \min & \sum_{t=1}^{N_t} q_t(D_t - c_t) & (4.13) \\ \text{s.t.} & \sum_{t \in \mathcal{E}_{(i,j)}} x_{i,j,t} = 1, \ \forall (i,j) \in \mathcal{C}_\nu^* & (4.1) \\ & (t_{i,j} - t_{i',j'} + N_k^l + \delta_{i,j})(t_{i,j} - t_{i',j'} - N_k^l - \delta_{i',j'}) \geq 0, & \\ & \qquad \forall i,j,i',j' : i \neq i' \in \mathcal{J}_k, \ k \in [N_g] & (4.10) \\ & c_t = \sum_{(i,j) \in \mathcal{C}_\nu} A_{i,t} W_i \left(\sum_{t'=t-\delta_{i,j}+1}^{t} x_{i,j,t'} \right) & (4.3) \\ & t_{i,j} = \sum_{t \in \mathcal{E}_{(i,j)}} t x_{i,j,t}, \ \forall (i,j) \in \mathcal{C}_\nu^* & \\ & x_{i,j,t} \in \{0,1\}, \ \forall t \in \mathcal{E}_{i,j}, \ \forall (i,j) \in \mathcal{C}_\nu^* & \end{cases} \tag{4.20}$$

4.3.3.5 Model 5

With this model, we aim at investigating a new way of handling uncertainty, namely the distributionnally robust approach. To this end, in order not to compound the difficulties, we put aside the combinatorial nature of the problem and we tackle a continuous version of the problem, that can now be considered as a classical problem of supply/demand equilibrium.

More specifically, we consider a system of N_θ production units, characterized by a deterministic time-dependent production cost $\gamma_{t,i}$ for the plant i at time step t. The essence of the problem is to determine the production of the plants i at each time step t : $x_{t,i} \in [0,1]$, in order to meet the uncertain demand $D_t(\xi^\delta)$ at each time step. The power plants are subject to random failure, represented by the coefficient $A_{i,t}(\xi^\theta)$

Furthermore, some technical constraints state that the prescribed production of a plant i over the time-horizon shall not exceed a given amount r_i. More precisely, these constraints stand for the necessity of shutting down the plants to proceed to maintenance operations, and is therefore independent of the uncertain availability of the plants.

These requirements are summarized in the following concise formulation :

$$\begin{cases} \min & \gamma^T x \\ \text{s.t.} & \mathrm{P}\left[\sum_{i=1}^{N} A_{i,t}(\xi^\theta) x_{t,i} \geq D_t(\xi^\delta),\ t = 1, ..., N_t\right] \geq 1 - \varepsilon \\ & \sum_{t=1}^{T} x_{t,i} \leq r_i,\ i = 1, ..., N_\theta \\ & x_{t,i} \in [0,1],\ i = 1, ..., N_\theta,\ t = 1, ..., N_t \end{cases} \tag{4.21}$$

This problem is therefore a linear problem with a joint chance-constraint. At paragraph 6.2, we compare the results obtained by exploiting two different levels of knowledge about the uncertain parameters ξ. First, we assume that only the support and the expected value are known, in which case we approximate the problem in a robust way by combining Boole's and Hoeffding's inequalities, which leads to a SOCP. Second, we consider the additional information provided by the second-order moment and exploit it in the spirit of distributionnally robust optimization.

4.3.4 Comparison of the different models

The table 4.2 gives a comparaison of the main features of the different models.

Models	Uncertainty	Objective	Max. Lapping	Nature of the problem
1	Scenarios	Quadratic	Quadratic	MIQCQP
2	Deterministic	Quadratic	Big M	MIQP
3	Deterministic	Quadratic	Quadratic	IQCQP
4-1	Deterministic	Linear	Big M	ILP
4-2	Deterministic	Linear	Pairwise exclusion	ILP
4-3	Deterministic	Linear	Quadratic	IQCQP
5	Dist. Robust	Linear	-	LP

Table 4.2: Comparison of the different models

This Table clearly shows that the models 2,3 and 4 are dedicated to the study of the combinatorial aspect in chapter 5, whereas the models 1 and 5 are intended to treatment of uncertainty in chapter 6.

4.4 Conclusion

Energy management is the combination of the managements of huge and heterogeneous portfolio of production units, in coordination with management of financial assets, that range from a short term in a few hours to a long term in many years. Resulting optimization problems have to comply with various difficulties and can be formulated as linear or fully quadratic stochastic optimization programs of huge size with mixed variables. Their resolution is therefore extremely hard, especially since the time required for their resolution is constrained by operational processes. Finally, the permanent changes

and evolutions of the energy sector pose new problems with specific difficulties and require a constant scalability of the implemented tools.

In this chapter, we provide the main elements to get acquainted with this domain, starting by the components of the demand/supply equilibrium, i.e., the different production units and financial assets, the demand and specific details regarding this constraint. We also discuss different manners of accounting with uncertainty into energy management problems.

Then, we focused on the Nuclear Outages Scheduling Problem (NOSP), an energy management problem famous for its combinatorial feature, that involves major economic stakes. We describe the problem in detail and propose 5 models of this problem, with varying degrees of precision, that emphasize different features of the problem. We compared these models in Table 4.2 in order to point out the differences and similarities between them, in particular regarding the way of considering uncertainty.

Thanks to all these elements, we can move on to the next chapters, where SDP is applied to the different models described here. The chapter 5 focus on combinatorial features and involves the models 2, 3 and 4, whereas the models 1 and 5 are intended to treatment of uncertainty in chapter 6.

Chapter 5

SDP for combinatorial problems of energy management

In spite of all the promising results obtained with SDP for approximating hard combinatorial problems, only few people have embarked on practically attacking such problems.

In this section, we are interested in confronting this theory with practice. Indeed, in energy management, we are faced with problems presenting combinatorial features, due to the fact that certain decisions corresponds to indivisible quantity, or for modelling certain "all or nothing" behaviours. It is also useful for taking piecewise phenomenon into account.

More precisely, we are interested in assessing the quality of the semidefinite relaxation for three of the combinatorial problems described in Chapter 4. All the theoretical elements necessary to acquaint oneself with this relaxation are provided in the Section 3.3. In particular, in the paragraph 3.3.2, we present in detail a systematic way of obtaining a semidefinite relaxation of a QCQP, the so-called *standard semidefinite relaxation*. We explain how this relaxation applies to 0/1-LP but need to be reinforced in order to outperform linear relaxation.

The first section of this chapter is taken from the paper [115] and presents a first approach to build a semidefinite relaxation of the nuclear outages scheduling problem modelled as described in Paragraph 4.3.3.2. This relaxation scheme is completed by a randomized rounding procedure allowing to recover a feasible solution from the optimal solution of the semidefinite program.

The second section is the restitution of the work reported in the submitted paper [113]. It presents a generic scheme for deriving and tightening the semidefinite relaxation of a QCQP and report an application of this method to the nuclear outages scheduling problem, more specifically, to the model described in paragraph 4.3.3.3.

For these two chapters, we chose not to provide the papers [115, 113] in their entirety in order to avoid the duplications. Regarding the methodology, we put aside the elements regarding the SDP theory since they are given in Chapter 2. Furthermore, we do not explicit the way we build the standard SDP relaxation of a QCQP since the latter is described in detail in Paragraph 3.3.2. Finally, as regards energy management and application problems, we refer the reader to the Chapter 4.

The third section contains complementary works dealing with the application of SDP to the model 4 of the NOSP, presented in Paragraph 4.3.3.4. This model is quite simple, in order to consider small instances, and focus on a very difficult constraint arising in NOSP : the maximal lapping constraint. A part of the work consists of comparing three possible models of this constraint, that can be seen as a linear disjunction : $a^T x \leq b$ or $a^T x \geq c$, where x are binary variables. In the fourth section, we compare several classical reinforcement of the standard semidefinite relaxations for this problem. We go a step further at the end of the section, by experimenting the hierarchy of semidefinite relaxation proposed by Lasserre for polynomial problems.

5.1 A first attempt of SDP relaxation for the nuclear outages scheduling problem

This section reports the work presented in the paper [115] which investigates semidefinite relaxation for the NOSP modelled as described in Paragraph 4.3.3.2. This relaxation scheme is completed by a randomized rounding procedure allowing to recover a feasible solution from the optimal solution of the semidefinite program.

The considered problem is a deterministic version of the NOSP that is described in detail in Paragraph 4.3.3.2. With respect to the other models proposed for this problem, it is rather complete. In particular, we optimize both the scheduling of outages, which induces binary decision variables, and the amount of supplied fuel and the nuclear power plants production planning, which corresponds to bounded continuous decision variables.

This problem is therefore a huge M-0/1-QP (Mixed 0/1 Quadratic Program). The quadratic feature comes from the objective function. Its compact formulation is as follows :

$$(P)\begin{cases} \min_{x,y} & x^tQx + p^tx + q^ty \\ \text{subject to} & Ax + By \leq c \\ & y \leq \bar{y} \\ & x \in \{0,1\}^{N_x},\ y \in \mathbb{R}_+^{N_y} \end{cases} \tag{5.1}$$

It is worth noticing that the quadratic terms involve only binary variables, which enables to solve the exact problem with CPLEX. However, this is not the primary objective of this work, which aims rather at comparing the strength of two possible relaxations for this problem.

First, we apply the standard semidefinite relaxation described at Paragraph 3.3.2 to the QCQP obtained by formulating the binary constraints as quadratic equalities $x_i^2 = x_i$. In a second step, we reinforce the SDP relaxation by adding some cuts based on the Sherali-Adams approach. We describe this process at Paragraph 5.1.1.

Then, we compare it to the relaxation in the form of a Quadratic Program that is obtained by relaxing $x_i \in \{0,1\}$ into $x_i \in [0,1]$. This so-called *continuous relaxation* can be solved with CPLEX since the objective function is convex.

Finally, the solutions of these relaxations are then used to compute a feasible solution, by using a randomized rounding scheme, described in the paragraph 5.1.2. Numerical results are reported in paragraph 5.1.3.

5.1.1 Reinforcing the standard semidefinite Relaxation

At this point, we consider that the standard SDP relaxation described in Paragraph 3.3.2 is implemented for the problem 5.1. We explained in Paragraph 3.3.3.1 that the standard SDP relaxation is generally not the most appropriate. In particular, adding some valid quadratic constraints may improve its bound. In this section, we apply the Sherali-Adams [240] principle described in detail in Appendix 3.4.3.2. Briefly, let $Ax = b$ be a set of linear constraints and x_i a binary variable, the constraints $Axx_i = bx_i$ is valid. We apply this idea to the uniqueness constraint (4.1), with all the variables x_i that appear in the constraint. By using $x_i^2 = x_i$ it comes :

$$\sum_{t' \in \mathcal{E}_{i,j},\ t' \neq t} x^{\nu}_{i,j,t} x^{\nu}_{i,j,t'} = 0,\ \forall t \in \mathcal{E}_{i,j},\ \forall (i,j) \in \mathcal{C}^*_{\nu} \tag{5.2}$$

5.1.2 Randomized rounding procedure

Randomization has proved to be a powerful resource to yield a feasible binary solution from a fractional one. The basic idea is to interpret the fractional value as the probability of the variable to take the value 1. Then the values of the binary variables are drawn according to this law and this process is iterated until the solution satisfies the constraints.

Here, we slightly change this principle, in order to find more easily a feasible solution : instead of deciding successively if a binary variable is 0 or 1, for each cycle, we choose one date among the possible beginning date for the associate outage, by using the fractional value as probability, since their sum is equal to one from the assignment constraint. Thus, the assignment constraint is necessarily respected by the integer solution.

Then, the values of the lapping variables x^{λ} follow. About the continuous variables, for the modulation x^{μ}, we keep the value of the relaxation and for the reload x^{ρ}, we take the minimal values that respects the ending stock constraint.

5.1.3 Numerical experiments

Data	Nb of	Opt		RelaxQP			RelaxSDP			RelaxSDP-Q		
set	bin. var.	Obj	Time	Gap	Time	RR	Gap	Time	RR	Gap	Time	RR
D-1	215	3 343	1	0.73	0.02	2.35	0.54	12	2.35	0.26	12	0.70
D-2	278	3 254	21	0.80	0.00	3.88	0.64	19	1.49	0.46	21	1.70
D-3	341	3 174	183	0.94	0.02	4.86	0.82	31	2.43	0.65	36	3.25
D-4	406	3 110	1 286	1.10	0.02	4.23	0.97	44	5.04	0.83	54	5.14
D-5	469	3 051	7 200	1.18	0.02	11.70	1.08	63	3.72	0.96	79	4.04
D-6	530	2 994	5 780	1.17	0.03	14.56	1.09	81	3.35	1.00	108	4.73
D-7	215	3 297	2	1.24	0.02	3.31	1.03	5	2.82	0.68	6	0.82
D-8	278	3 223	8	1.89	0.03	10.28	1.72	8	7.15	1.38	11	3.35
D-9	341	3 176	39	2.94	0.08	11.31	2.81	15	9.95	2.49	64	2.11
D-10	406	3 133	169	3.91	0.13	14.69	3.80	26	11.94	3.52	98	8.98
D-11	469	3 070	76	3.87	0.18	13.56	3.78	38	13.81	3.53	147	11.79
D-12	530	3 024	232	4.25	0.20	14.47	4.17	53	17.98	3.95	236	16.20
D-13	539	12 580	7 200	0.85	0.05	3.16	0.77	154	3.28	0.61	171	2.08
D-14	698	12 431	7 200	0.95	0.10	3.47	0.89	252	3.76	0.76	286	4.06
D-15	852	12 290	7 200	1.13	0.14	5.78	1.08	373	4.58	0.99	436	4.83
D-16	1 011	12 156	7 200	1.14	0.14	6.16	1.09	578	5.19	1.02	750	5.29
D-17	1 170	12 034	7 200	1.15	0.22	5.72	1.12	791	5.77	1.08	1008	6.36
D-18	1 322	11 939	7 200	1.35	0.27	6.47	1.32	1030	5.67	1.30	1308	7.00
D-19	537	12 679	7 200	1.21	0.16	2.80	1.16	68	2.95	1.07	310	4.48
D-20	695	12 464	7 200	1.57	0.54	5.96	1.52	137	6.56	1.44	447	6.31
D-21	853	12 289	7 200	1.98	0.94	9.28	1.94	242	8.91	1.85	805	6.74
D-22	1 008	12 159	7 200	2.37	1.90	9.15	2.33	382	7.47	2.27	1113	8.80
D-23	1 165	12 034	7 200	2.65	2.95	7.87	2.62	628	7.70	2.58	2106	6.86
D-24	1 316	11 915	7 200	2.87	3.65	10.93	2.84	823	9.89	2.80	2231	8.52
Av.	651.83	7700.85	4224.91	1.80	0.49	7.75	1.71	243.88	6.41	1.56	493.46	5.59

Table 5.1: Results of exact search, relaxations and randomized rounding

Numerical experiments have been performed on a three years time horizon (156 weeks), with one outage per year for each plant and two nuclear parks (respectively 10 and 20 nuclear power plants for the data set 1 to 12, and 13 to 24). Each park is declined into two versions which differ from the maximum amount of reload ($\bar{R}_{i,j}$) and modulation ($M_{i,j}$). Finally, six instances have been tested for each data set, varying by the size of the search spaces associated to the outages dates variables (7 to 17 possibles dates).

All the computations was made on an Intel(R) Core(TM) i7 processor with a clock speed of 2.13 GHz. In order to compare the solutions in the same conditions, the CPLEX results are obtained without activating the preprocessing. For each data set we computed :

– *Opt* : the best solution found within the time limit (2 hours) by using CPLEX-Quadratic 12.1. The time value 7200 means that the time limit has been reached, so the obtained integer solution is not optimal ;
– *RelaxQP* : the continuous relaxation solved with CPLEX-Quadratic 12.1;
– *RelaxSDP* : the standard SDP relaxation solved with the SDP solver CSDP 6.1.1 (cf [53]);
– *RelaxSDP-Q* : the reinforced SDP relaxation solved computed with CSDP 6.1.1 ;

For each data set, the table 5.1 reports the number of binary variables, the value of the objective function (in currency unit), the computational time in second and, for each kind of relaxation, the associated gap (Gap) and the relative gap of the randomized rouding (RR), whose formula are given below. The last line (Av.) gives the average of the previous lines.

$$Gap = \frac{p_{opt} - p_{relax}}{p_{relax}} \qquad RR = \frac{p_{RR} - p_{opt}}{p_{opt}}$$

Analysis of the results
First we observe that CPLEX reaches the limited time for relatively small instances (e.g. 469 binary variables). This is in line with our expectations that this kind of problem is very hard for CPLEX, despite a quite small gap attained with continuous relaxation.

This may be related to the fact that, due to the demand constraint, the variable part of the objective function is very small w.r.t the absolute value of the cost. In other words, the optimal value is high, even with a "perfect" outages scheduling. Let us denote P the best possible objective value for a given data set, computed by considering the largest possible search space, and let's consider the variable part of the objective function, that is $p - P$, if p is the objective value. Then, the gap would increase, as shown in the following formula :

$$\frac{p_{opt} - p_{relax}}{p_{relax} - P} > \frac{p_{opt} - p_{relax}}{p_{relax}}$$

This illustrates the importance of considering the relative improvement of the gap achieved by semidefinite relaxation, rather that its absolute value.

For example, on the data set D-1, the gap is almost divided by three. Unfortunately, this ratio decreases as the number of binary variables raises, whereas the gap increases. This can be explained by the fact that the integer solution provided here is not optimal, considered that the computational time of CPLEX is limited. Let us denote by $p'_{opt} > p_{opt}$ this value : then the ratio computed with this value is greater than the ratio computed with p :

$$\frac{p_{opt} - p_{relaxCPLEX}}{p_{opt} - p_{relaxSDP}} > \frac{p'_{opt} - p_{relaxCPLEX}}{p'_{opt} - p_{relaxSDP}}$$

On average, the gap improves from 1.80% to 1.71% with the standard SDP relaxation and to 1.56% with the addition of valid equalities. This latter improvement is promising, even though it comes at high additional computational cost, particularly on the larger instances. This can be ascribed to the fact that SDP solvers are only in their infancy, especially compared to a commercial solver like CPLEX.

Finally, the randomized rounding yields satisfying results : due to the random aspect of the procedure, there are still some data set where the continuous relaxation gives better results than the semidefinite relaxation, but on average the loss of optimality reduces from 7.75% to 6.41% and 5.59%, which is significant when considering the huge amount at stake.

5.1.4 Conclusion

We investigated semidefinite relaxations for a MIQP (Mixed-Integer Quadratic Program) version of the scheduling of nuclear power plants outages. Comparison of the results obtained on significant data sets shows the following main results. First, our MIQP is extremely hard to solve with CPLEX. Second, semidefinite relaxations provide a tighter convex relaxation than the continuous relaxation. In our experiments the gap between the optimal solution and the continuous relaxation is on average equal to 1.80% whereas the semidefinite relaxation yields an average gap of 1.56%. Third, the computational time for computing these semidefinite relaxations is reasonable. Exploiting those results in a randomized rounding procedure instead of the result of the continuous relaxation leads to a significant improvement of the feasible solution.

In the view of these preliminary results, additional investigations will concern i) introduction of more valid inequalities, ii) evaluation of others SDP resolution techniques, for instance *Conic Bundle* for facing problems of huge size.

5.2 Generating cutting planes for the semidefinite relaxation of quadratic programs

The purpose of this section is to present a generic scheme for tightening the semidefinite relaxation of a QCQP and to report an application of this method to the NOSP, more precisely, to the model 3 described in paragraph 4.3.3.3. This work corresponds to the submitted paper [113].

This scheme can be described as follows. For a given QCQP, we start by deriving the standard SDP relaxation as described in Paragraph 3.3.2. In parallel we built a set of valid quadratic constraints for this QCQP, by multiplying all the linear constraints of the QCQP, including the bound constraints, between them. We denote $\mathcal{P}_S$ this set of constraints completed by the initial constraints of the considered QCQP.

Then, similarly to a separation algorithm, we select iteratively the most violated constraint among $\mathcal{P}_S$ and we add it to the semidefinite relaxation, where they act as cutting planes. In order to generate more efficient cutting planes, we investigates another version of the separation problem, where the constraint is selected among all the suitable combinations of elements of $\mathcal{P}_S$ and is required to be convex. In this case, the separation problem is a SDP.

We apply this method to the model 3 of NOSP (see Paragraph 4.3.3.3), which is a 0/1-QCQP. In order not to immediately restrict ourself to a particular 0/1-QCQP, we start by testing it on randomly generated instances of 0/1-QCQP, called *working instances*.

In short, our contribution is threefold. First, we design an automatic method to tighten the standard SDP relaxation of a QCQP, based on the pairwise products of the linear constraints of the problem. Besides, we provide a set of proofs that some products of linear constraints do not need to be considered. Finally, we show that our framework unify many seemingly disparate constraints for tightening semidefinite relaxation that are proposed in the literature.

This section is organized as follows. First, we introduce QCQP and present how SDP applies to this area. Our main contribution is given in the next paragraph by describing the elaboration of the set $\mathcal{P}_2$ and the design of the separation algorithm. We also discuss how our approach relates to prior works on cutting planes generation for semidefinite relaxation. Finally, we report experimental results and give a conclusion.

In all the section, we consider the following QCQP :

$$\left\{ \begin{array}{ll} \min_{x\in\mathbb{R}^n} & x^T P_0 x + 2p_0^T x + \pi_0 \\ \text{subject to} & x^T P_j x + 2p_j^T x + \pi_j \leq 0,\ j = 1, ..., m_q \\ & b_j \leq a_j^T x \leq c_j,\ j = 1, ..., m_l \end{array} \right. \tag{5.3}$$

where $P_j \in \mathbb{S}^{n\times n}, p_j \in \mathbb{R}^n, \pi_j \in \mathbb{R},\ j = 0, ..., m_q$ and $a_j \in \mathbb{R}^n, b_j \in \mathbb{R}, c_j \in \mathbb{R},\ j = 1, ..., m_l$ are the problem parameters. The feasible set of this problem is denoted $\mathcal{F}$ and p^* is its optimal value. With this formulation, we emphasize the linear constraints because of their key role in the process of building the semidefinite relaxation. Writing them as range constraints is a mild loss of generality. It suffices for instance to assume that all the variables are bounded for getting easily such a formulation. We make this assumption and suppose that these bounds are included within the linear constraints. Without loss of generality, we assume that these bounds are $[0, 1]$, by means of an affine transformation.

5.2.1 State of the art of the semidefinite relaxation of QCQP

The problem (5.3) is convex if and only if all the matrices P_j are positive semidefinite. Otherwise it may harbor many local minimal and is NP-hard [141]. To see this, one only need to notice that it generalizes many difficult problems as Polynomial Programming or Mixed 0-1 Linear Programming, since the binary constraints can be treated as two quadratic inequalities : $x_i \in \{0, 1\} \Leftrightarrow \{x_i^2 \leq x_i, x_i^2 \geq x_i\}$.

Finally, QCQP arises directly in a wide range of practical applications [125, 59], partly due to their ability to model Euclidean distances. Moreover, this optimization problem is central to well-known iterative methods such as trust-region sequential quadratic programming. For all these reasons, it is now considered as one of the most challenging optimization problems and an important work has been carried out to solve this general problem and its special cases.

Generally, methods for solving a QCQP are derived from nonlinear programming. In particular, the Branch & Bound procedure is appropriate since two convex relaxations are available, based on a linear relaxation called Reformulation-Linearization Technique (RLT) [17, 241, 183] or on semidefinite relaxation [224, 65]. A comparison of these relaxations can be found in [14] that shows that combining those approaches leads to an enhancement of their respective bounds.

Another possibility for relaxing a QCQP into a convex problem was proposed by Kim and Kojima in [156]. This relaxation produces a Second-Order Cone Program and can be considered as a compromise between the semidefinite and the linear relaxation.

In the particular case of a convex QCQP, previously studied by Hao [125] under theoretical and computational aspects, an interior-point method was proposed in [4] to solve this polynomial-time solvable problem and the connection with Second-Order Conic Programming was established in [185].

Finally, a QCQP can also be viewed as a particular polynomial program, with all the polynomials of degree 2. As such, we can apply the Lasserre's hierarchy of SDP relaxations, whose optimal value approximate the optimal value of the original problem as closely as desired. However, the size of the SDP increases rapidly, which makes it difficult to use in practice. We refer the reader to the new handbook [12] and to the seminal papers of Lasserre on this hierarchy [169, 172].

5.2.1.1 Relaxing a QCQP into a SDP

We recall hereafter the classical way for deriving the standard semidefinite relaxation of the problem (5.3), i.e., a relaxation that can be written as a Semidefinite Program. This approach was initially proposed for linear integer programs by Lovász and Schrijver in [187] and extended to QCQP by Fujie and Kojima in [98]. It turns out that this relaxation is the dual of the so-called *Shor relaxation*, i.e., another semidefinite relaxation for QCQP in the form of Linear Matrix Inequalities introduced in [245]. For an extensive discussion on the use of the semidefinite relaxation for QCQP, we refer the reader to the recent survey [23].

For a *purely* quadratic program, i.e., the problem (5.3) with $m_l = 0$, obtaining the semidefinite relaxation is straightforward. First, we reformulate equivalently the problem by introducing an augmented matrix of variables Y and the convenient matrices Q_j :

$$\begin{cases} \min & Q_0 \bullet Y \\ \text{subject to} & Q_j \bullet Y \leq 0,\ j = 1, ..., m_q \\ & Y = \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix} \end{cases} \tag{5.4}$$

where $Q_j = \begin{pmatrix} \pi_j & p_j^T \\ p_j & P_j^T \end{pmatrix},\ j = 0, ...m_q$.

With this formulation, all the non-linearity is pushed into the last constraint, which comes to impose to the matrix Y to be of rank 1, positive semidefinite and with $Y_{00} = 1$, if Y indices start at zero. Then, the semidefinite relaxation is obtained by relaxing the rank-1 constraint and requiring only that Y be positive semidefinite and $Y_{00} = 1$.

$$(P_S) \begin{cases} \min & Q_0 \bullet Y \\ \text{subject to} & Q_j \bullet Y \leq 0, j = 1, ..., m \\ & Q_{m_q+1} \bullet Y = 1 \\ & Y \succcurlyeq 0 \end{cases} \tag{5.5}$$

where $Q_{m_q+1} = e_0 e_0^T$. We denote x and X the elements of Y defined as follows :

$$Y = \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \tag{5.6}$$

Then, by applying Schur's complement, $Y \succcurlyeq 0$ is equivalent to $X - xx^T \succcurlyeq 0$ and the semidefinite relaxation comes to relax $X = xx^T$ into $X - xx^T \succcurlyeq 0$.

A connection with Lift & Project [21, 240, 187] can be established here. In these methods, the first step denoted *lifting* consists of extending the variable space to a higher dimensional space, by introducing new variables. This is exactly what is done by introducing the variable X, that lifts the problem from the space of the n-vectors to the space of the n symmetric matrices. The difference between our approach and the classical Lift & Project is that, instead of projecting it to get valid inequalities in the original space, we solve the problem in the lifted space and project the obtained solution by picking the vector x.

Note that any convex constraint, i.e., with $P_j \succcurlyeq 0$, of the original problem is necessarily respected by the projected solution, since $X - xx^T \succcurlyeq 0$ implies in this case that $P_j \bullet (X - xx^T) \geq 0$. Consequently, when the problem (5.3) is convex, the semidefinite relaxation yields the optimal solution. See for instance Proposition 1.4.1 in [86] or [161].

In [98], Fujie & Kojima took step further by showing the equivalence of this relaxation with a relaxation obtained by considering all the convex inequalities generated as nonnegative combination of the constraints. They also established the equivalence with the Lagrangian relaxation of the quadratic constraints, as noticed subsequently by several authors [177, 55, 91].

By considering the linear constraints as particular quadratic constraints, where the quadratic term is null : $a_j^T x - b_j = x^T P_j x + 2p_j^T x + \pi_j$, with $P_j = 0, p_j = 1/2 b_j, \pi_j = -b_j$, we apply the latter scheme to the problem (5.3) with $m_l > 0$. We refer to this relaxation as the standard semidefinite relaxation because of its simplicity. On the downside, it has two major drawbacks. First, in the case of a Mixed 0/1 LP, it provides the same bound as the continuous relaxation, which is much easier to solve. It may also happen that the semidefinite relaxation is unbounded, even when all the original variables have finite bounds, due to the fact that the connection between x and X - i.e., $X \succcurlyeq xx^T$ - is too weak. Such a situation occurs for instance whenever Q_0 is not psd and $\text{diag}(X)$ is not bounded.

As detailed in the next section, this connection can be reinforced by adding redundant quadratic constraints, called *seminal constraints*, to the original problem and applying the standard semidefinite relaxation to this reinforced problem.

In this paper, we push further in this direction by developping a method for automatically designing these seminal constraints. The tools that are used, i.e., the identification of the best convex combination of a set of constraints, is well-known and was already used in particular in the convexification approach of Billionnet et al. [45, 46, 47, 48]. However, the philosophy is very different. The approach of Billionnet proposes a method to get a convex QCQP reformulation of the problem, whereas we build a SDP relaxation of the problem that is tightened iteratively. Furthermore, by constrast with convexification, our approach does not require that the variables be integer.

Finally, the convexification approach was extended to MIQCQP by [179], to convexify the continuous relaxation and to use it within a Branch and Bound procedure. A comprehensive overview for this field can be found in [64].

5.2.1.2 Handling the linear constraints

The treatment of the linear constraints and their transformation into quadratic constraints plays a key role in the design of the semidefinite relaxation. The most natural quadratic formulation, with a null quadratic term, for linear constraints leads to the standard semidefinite relaxation. A tighter relaxation, referred to as *initial semidefinite relaxation*, is produced by following the recipes of [213, 177, 127, 230]. Let us consider a range inequality :

$$b \leq a^T x \leq c \Leftrightarrow (a^T x - b)(a^T x - c) \leq 0 \Leftrightarrow x^T a a^T x - (b+c)a^T x + bc \leq 0 \tag{5.7}$$

Those constraints are convex since $aa^T \succcurlyeq 0$. Consequently, the projected solution of the semidefinite relaxation necessarily respects the quadratic constraints and therefore the original linear constraints are useless.

Regarding the equality constraints, as suggested in [91], we keep the standard formulation as well as the products of the constraint by each variable of the problem :

$$a^T x = b \Leftrightarrow \left\{ \begin{array}{l} a^T x = b \\ (a^T x - b)x_i = 0, i = 1, ..., n \end{array} \right. \tag{5.8}$$

Note that is necessary to keep the linear constraint, otherwise there is no guarantee that the constraint be satisfied by the projected solution of the semidefinite relaxation. On the other hand, this formulation ensures that there is no duality gap, as opposed to the more concise formulation $(a^T x - b)^2 = 0$.

Finally, the so-called initial semidefinite relaxation is built as the standard semidefinite relaxation of the QCQP obtained by setting all the quadratic constraints and the above quadratic formulation of the linear constraints.

5.2.1.3 Addition of cutting planes to strengthen the relaxation

In this section, we discuss some cutting planes that have been proposed to strengthen the standard semidefinite relaxation. To a large extent, such works concern a more restrictive part of QCQP, mainly MIQCQP, where the cuts exploits the fact the variables are integer. In the interests of concision, we restrict ourselves to the cuts for *pure* MIQCQP, i.e., not MILP. We just mention that a large number of cutting planes for MILP can be generated by applying the Lift & Project method, in particular by 3 hierarchies [21, 240, 187] that yields the convex hull of the feasible set in a finite number of iterations .

In [14], it is suggested to bound the diagonal of X : $X_{ii} \leq \max\{u_i^2, l_i^2\}$, where l_i and u_i are the bounds of x_i, in order to avoid that the SDP relaxation be unbounded.

Another classical way to generate valid quadratic constraints stems from a particular case of linear disjunction. Indeed, the disjunction $(a^T x \leq b) \vee (a^T x \geq c)$ with $b < c$, can be formulated as the quadratic constraint $(a^T x - b)(a^T x - c) \leq 0$.

In [84], Deza and Laurent introduced an automatic method to generate such valid disjunctions by exploiting the integrity of the variables. For any integer vector $b \in \mathbb{Z}^n$ such that $b^T e$ is odd, $2b^T x \leq b^T e - 1$ or $2b^T x \geq b^T e + 1$. These cuts, called *hypermetric inequality* are applied to semidefinite relaxation in [128]. The most famous of them are the so-called *triangle inequalities*, obtained for any indices $i \neq j \neq k$ by picking successively $b = -e_i - e_j - e_k$, $b = -e_i + e_j + e_k$, $b = e_i - e_j + e_k$ and $b = e_i + e_j - e_k$:

$$\begin{array}{l} (i)\ x_i + x_j + x_k \leq X_{ij} + X_{ik} + X_{jk} + 1 \\ (ii)\ X_{ik} + X_{jk} \leq x_k + X_{ij} \\ (iii)\ X_{ij} + X_{ik} \leq x_i + X_{jk} \\ (iv)\ X_{ij} + X_{jk} \leq x_j + X_{ik} \end{array} \tag{5.9}$$

Another contribution in this vein was made in [131]. The constraint $a^T x - b \geq 0$, with a and b integer leads to the valid disjunction $a^T x - b \leq 0$ or $a^T x - b \geq 1$, i.e., $(a^T x - b)(a^T x - b - 1) \leq 0$.

Some other disjunctions can be used to generate valid constraints. In [146], the authors discussed the generation of valid quadratic cuts for 0/1 convex QCQP, i.e., a special case of QCQP where the non-convexity is due exclusively to the binary constraints. Then, the generation of the cut follows the well-known principle of a cutting plane algorithm [21], where a separation problem is solved at each iteration in order to determine a cut that is both valid and violated by the current relaxed solution. The relaxation solved at each iteration is a convex QCQP, and the cut generation is based on disjunctive programming.

In [235], the authors proposed valid disjunctions based on the constraint $A \bullet (X - xx^T) \leq 0$ that holds for any matrix A whenever $X - xx^T = 0$ is valid. By picking $A \succcurlyeq 0$, such a constraint may improve the semidefinite relaxation, since the latter implies that $A \bullet (X - xx^T) \geq 0$. The difficulty is that the quadratic term $x^T Ax$ do not appear in the semidefinite relaxation. To overcome this difficulty, this term is replaced by a valid linear disjunction, for a rank 1 matrix : $A = cc^T$. The vector c is built as the best positive combination of eigenvectors of the incumbent solution $X - xx^T$. Remark that the disjunction generated here is not exclusive, which means that both parts of the disjunction may be satisfied. In this case, multiplying the linear constraints to get a valid quadratic constraints in not possible. Instead, a valid linear constraint is derived by applying Balas' technique [18].

Finally, an other paper from the same authors [236] share many techniques with our paper, but these techniques are used differently. They also use a semidefinite program to compute valid quadratic convex cuts but the objective of the separation are different. They aim at getting rid of the lifted variables (projection), whereas we aim at selecting the best constraint among a set of generated constraints. Note that both approaches could easily be combined.

5.2.2 A separation problem for generating cutting planes for the semidefinite relaxation

For the QCQP (5.3), we aim at strengthening the standard semidefinite relaxation described at Paragraph 3.3.2 by adding valid quadratic constraints to the original problem. These seminal constraints are built following two steps, according to the principle presented at Paragraph 3.3.3.1 :

(i) valid quadratic constraints are generated as pairwise product of the linear constraints, including the bound constraints ;

(ii) suitable combination of these constraints and of the initial constraints of the problem are taken.

Let us denote by $\mathcal{P}_S$ the set of the initial constraints of the problem augmented by all the constraints built at the step (i) and assume that $\mathcal{P}_S = \{\mathrm{q}_j(x) \leq 0,\ j = 1, ..., m_i;\ \ \mathrm{q}_j(x) = 0,\ j = m_i + 1, ..., m_i + m_e\}$, then a suitable combination of elements of $\mathcal{P}_S$ is defined as follows :

$$\sum_{j=1}^{m_i+m_e} \lambda_j \mathrm{q}_j(x) \leq 0 \text{ for } \lambda_j \geq 0,\ j = 1, ..., m_i \text{ and } \lambda_j \in \mathbb{R},\ j = m_i + 1, ..., m_i + m_e$$

Such a constraint is necessarily valid for (5.3). Our approach consists of selecting the most appropriate constraint among the suitable combination of $\mathcal{P}_S$ and adding it to the problem in order to reinforce the semidefinite relaxation. The selection constitutes the *separation problem*. Notice that, even if the selected constraint is quadratic, the associated constraint that is added to the semidefinite relaxation is linear and is therefore denoted a *cutting plane*.

This approach is motivated by some earlier works. The notion of surrogate constraints, generated as suitable combination of constraints, was introduced by Glover [106]. Balas exploited this notion to compute projections of polyhedra [22] and to characterize the convex hull of a disjunction of polyhedra [19]. In [243], Sherali introduced the idea of multiplying linear constraints to generate valid quadratic constraints, in order to reinforce the RLT relaxation of QCQP. Our approach is also based on the work of Kojima and Tunçel [161] that proposed a SDP-based iterative procedure to reach the convex hull of a compact set represented by quadratic inequalities. Finally, a recent development of Saxena et al. [235] on MIQCQP serve our work as an inspiration. In this work, a set of valid disjunctions is generated, then the most appropriate combination of them is selected through a linear or a semidefinite program.

5.2.2.1 Separation algorithm

The scheme of the method is to add successively some valid quadratic constraints to the original problem in order to strengthen the semidefinite relaxation. The algorithm is given below :

1: $\mathcal{P}_C = \mathcal{P}$
2: $(P_C) : \min\{\mathrm{q}_0(x) : \mathrm{q}_j(x) \leq 0, \forall \mathrm{q}_j \in \mathcal{P}_C\}$
3: $\tilde{Y} = \begin{pmatrix} 1 & \tilde{x}^T \\ \tilde{x} & \tilde{X} \end{pmatrix} \leftarrow$ Solution of the standard semidefinite relaxation of (P_C).
4: **if** $\tilde{x}$ is feasible for the original problem **then**
5: STOP
6: **else**
7: solve the problem $(S_{\tilde{X},\tilde{x}})$. Let q^* be the optimal value and q the optimal solution.
8: **end if**
9: **if** $q^* > t$ **then**
10: $\mathcal{P}_C \leftarrow \mathcal{P}_C \cup \{\mathrm{q}\}$. Go to 2.
11: **else**
12: STOP
13: **end if**

The set $\mathcal{P}$ contains the quadratic constraints corresponding to the initial semidefinite relaxation, as detailed in paragraph 3.3.3.1. The set $\mathcal{P}_S$ is the union of $\mathcal{P}$ and of the pairwise products of the linear constraints. Its design is central to our approach and is discussed in paragraph 5.2.2.2. For sake of simplicity, we assume that all the constraints of $\mathcal{P}_S$ are inequalities, by splitting the equalities. t is the violation threshold, i.e., a non-negative value close to 0, that represents the minimum violation required to add a cut.

The separation problem $(S_{\tilde{X},\tilde{x}})$ is the key element of the method. It aims at determining the best suitable combination of elements of $\mathcal{P}_S$, so as to maximise a given criteria, namely the violation of the obtained constraint by the incumbent solution $(\tilde{X}, \tilde{x})$. This optimization problem is as follows :

$$(S_{\tilde{X},\tilde{x}}) \begin{cases} \max & \sum_{i=1}^{r} \lambda_i [P_i \bullet \tilde{X} + 2p_i^T \tilde{x} + \pi_i] \\ \text{subject to} & \sum_{i=1}^{r} \lambda_i P_i \succcurlyeq 0 \\ & \sum_{i=1}^{m} \lambda_i \leq 1 \\ & \lambda \in \mathbb{R}_+^m \end{cases}$$

where $\mathcal{P}_S := \{\mathrm{q}(.; P_i, p_i, \pi_i), i = 1, ..., r\}$.

This problem is therefore a semidefinite program. The constraint $\sum_{i=1}^{m} \lambda_i \leq 1$, denoted *normalization conditions*, is necessary to truncate the feasible set, otherwise the problem is unbounded.

Besides, by imposing that $\sum_{i=1}^{r} \lambda_i P_i \succcurlyeq 0$, we restrict our research to the convex constraints. Indeed, this property ensures that the corresponding constraints are satisfied by the projected solution. This might seem useless by the result of [98] that states precisely that all the convex combinations of constraints of $\mathcal{P}$ are satisfied by the semidefinite relaxation. The difference here is that we consider the convex combinations of $\mathcal{P}_S$ which is larger than $\mathcal{P}$. Thus, at each iteration, we get closer to the semidefinite relaxation of the huge QCQP obtained by adding all the constraints of $\mathcal{P}_S$.

To conclude this section, we make the connection with the S-procedure [262]. We are seeking for a quadratic inequality $\mathrm{q}(x) \leq 0$ that is valid over a set $\mathcal{F}$, defined through a set of quadratic constraints : $\mathcal{F} = \{x \in \mathbb{R}^n : \mathrm{q}_j(x) \leq 0,\ j = 1, ..., m\}$. Formally, q is such that :

$$\mathrm{q}_j(x) \leq 0,\ j = 1, ..., m \quad \Rightarrow \quad \mathrm{q}(x) \leq 0$$

The S-procedure states that a sufficient condition for a function q to be valid is that $\mathrm{q}(x) - \sum_{j=1}^{m} \lambda_j \mathrm{q}_j(x) \leq 0, \forall x \in \mathbb{R}^n$ for some $\lambda \geq 0$. In the case where $m = 1$, it is said to be *lossless*, which means that any valid constraints over $\mathcal{F}$ admits such a representation, or in other words, the sufficient condition is also necessary. But generally, this is not the case and this procedure is only a conservative approximation. This is precisely what we are doing in our approach. In order to limit the conservativeness of the approximation we extend the set $\mathcal{P}$ to a larger set $\mathcal{P}_S$.

5.2.2.2 Designing $\mathcal{P}_S$

A key issue of our method lies in designing the set $\mathcal{P}_S$, that shall contain appropriate valid quadratic constraints. We start by adding to $\mathcal{P}_S$ the quadratic constraint of the initial problem. Then, we build all the pairwise products of its linear constraints, including the bound constraints. Only one-sided linear constraints are considered, by splitting the range inequalities $b_i \leq a_i^T x \leq c_i$ into two one-sided constraints.

Among all the valid quadratic constraints that are generated as a pairwise product of linear constraints, the following ones stand out :

- $(a^T x - b)^2 \geq 0$;
- $(a^T x - b)(c^T x - d) = 0$, for any valid equality $c^T x = d$;
- $(a^T x - b)(c^T x - d) \geq 0$, for any valid inequality $c^T x - d \geq 0$ and $b \leq \sum_{i=1}^{n} \min\{0, a_i\}$.

The latter inequalities are valid since $(a^T x - b) \geq 0$ holds for any $b \leq \sum_{i=1}^{n} \min\{0, a_i\}$, as $x \in [0, 1]^n$. We show that these infinite number of constraints, and some others, are useless for our approach.

We denote $\mathcal{P}_S = \{\mathrm{q}_i, i = 1, ..., m\}$. If $(\tilde{X}, \tilde{x})$ is the solution of the incumbent semidefinite relaxation, we compute $\gamma_i = P_i \bullet \tilde{X} + 2p_i^T \tilde{x} + \pi_i$ for each element $\mathrm{q}_i(P_i, p_i, \pi_i)$ of $\mathcal{P}_S$. Then a constraint is violated by $(\tilde{X}, \tilde{x})$ if $\gamma_i > 0$ and necessarily $\gamma_i \leq 0$ for each constraint $\mathrm{q}_i \in \mathcal{P}_C$.

The constraint associated with the quadratic function $\mathrm{q}(.; P, p, \pi)$ is convex (resp. concave) if $P \succcurlyeq 0$ (resp. $P \preccurlyeq 0$). It is linear if it is both convex and concave ($P = 0$). The following result allows to remove the concave constraints (including the linear ones) from $\mathcal{P}_S$.

Proposition 5.2.1 *Removing from $\mathcal{P}_S$ the concave constraints does not change the optimal solution of the separation problem.*

Proof 5.2.2 *We start by proving that for any concave $\mathrm{q}_i(.; P_i, p_i, \pi_i) \in \mathcal{P}_S$,$\gamma_i \leq 0$. Indeed, by construction of $\mathcal{P}_S$, either q_i is a quadratic constraint of the initial problem, and then it also belongs to $\mathcal{P}_C$, which implies that its semidefinite relaxation $\gamma_i = P_i \bullet \tilde{X} + 2p_i^T \tilde{x} + \pi_i \leq 0$ is satisfied.*

Otherwise, q_i is a product of linear constraint. As $\mathcal{P}$ is built so as to guarantee that the projected solution $\tilde{x}$ satisfies all the linear constraints. Consequently, $\mathrm{q}_i(\tilde{x}) \leq 0$. As $P_i \preccurlyeq 0 \Rightarrow P_i \bullet (\tilde{X} - \tilde{x}\tilde{x}^T) \leq 0$, then $\gamma_i = P_i \bullet \tilde{X} + 2p_i^T \tilde{x} + \pi_i \leq \mathrm{q}_i(x) \leq 0$.

Let us consider the optimal solution $\mathrm{q}^*(.;P,p,\pi)$ *and the optimal value* q^* *of the separation problem. We denote by* $I_c \subset [r]$ *the set of indices of elements of* $\mathcal{P}_S$ *such that* q_i *is concave. Then* $P = \sum_{[r]\setminus I_c} \lambda_i P_i + \sum_{I_c} \lambda_i P_i$. $P \succcurlyeq 0$ *and* $\sum_{I_c} \lambda_i P_i \preccurlyeq 0$ *imply that* $\sum_{[r]\setminus I_c} \lambda_i P_i \succcurlyeq 0$. *Consequently the solution obtained by setting* $\lambda_i = 0, i \in I_c$ *is feasible. Its cost is* $q^* - \sum_{i\in I_c} \lambda_i\gamma_i \geq q^*$, *and necessarily the variables* λ_i *associated to such constraints are equal to zero.* □

In particular, the constraints $(a^Tx - b)^2 \geq 0$ are concave and therefore useless. More generally, a constraint made as a product of linear constraints is concave or convex if and only if its quadratic term has rank 1. To see this, we consider two linear inequalities : $a^Tx - b \geq 0$ and $c^Tx - d \geq 0$. After symmetrization, the quadratic term of their product is $x^T(1/2ac^T + ca^T)x$.

Proposition 5.2.3 *A matrix* M *of the form* $M = ac^T + ca^T$ *is positive (resp. negative) semidefinite if and only if* a *and* c *are colinear, i.e.,* $a = 0$ *or* $c = 0$ *or there exists a real* $\rho > 0$ *(resp.* $\rho < 0$*) such that* $c = \rho a$.

Proof 5.2.4 *Suppose that* $c = \rho a$ *with* $\rho > 0$*: it is clear that* $M = 2\rho aa^T \succcurlyeq 0$. *As well, if* $a = 0$ *or* $c = 0$, $M = 0 \succcurlyeq 0$. *Conversely, if* a *and* c *are not colinear, then* $a \neq 0$ *and* $c \neq 0$ *and we can define the vector* $u = \frac{a}{\|a\|} - \frac{c}{\|c\|} \neq 0$. *Then* $a^Tu > 0$, $c^Tu < 0$ *and* $u^TMu = u^Tac^Tu + u^Tca^Tu = 2(a^Tu)(c^Tu) < 0$ *and the matrix* M *is not positive semidefinite. For the case where* $\rho < 0$, *the proof is similar.*□

Consequently, the only possibility for a constraint $(a^Tx-b)(c^Tx-d) \leq 0$, with $a \neq 0, c \neq 0$ to be convex is that $c = \rho a$, with $\rho > 0$. By defining $b' = d/\rho$, we have $(a^Tx - b)(a^Tx - b') \leq 0$ which corresponds to $\min\{b, b'\} \leq a^Tx \leq \max\{b, b'\}$.

Proposition 5.2.5 *Removing from* $\mathcal{P}_S$ *the suitable normalized combinations of other elements of* $\mathcal{P}_S$ *does not change the optimal solution of the separation problem.*

Proof 5.2.6 *Let us consider the optimal solution* $\mathrm{q}^* = \sum_{i=1}^r \lambda_i\mathrm{q}_i$ *and assume that* $\mathrm{q}_r = \sum_{i=1}^{r-1} \mu_i\mathrm{q}_i$ *with* $\sum_{i=1}^{r-1} \mu_i \leq 1$. *Then,* $\mathrm{q}^* = \sum_{i=1}^{r-1}(\lambda_i + \lambda_r\mu_i)\mathrm{q}_i$ *and* $\sum_{i=1}^{r-1} \lambda_i + \lambda_r\mu_i \leq 1$ *so the optimal solution remains feasible by removing* q_r. □

In our algorithm, we make the choice of removing all the constraints that are suitable combination of other elements of $\mathcal{P}_S$, even if not normalized, since these constraints can be viewed as the multiplication of a suitable normalized combination of elements of $\mathcal{P}_S$ by a nonnegative constant. As a consequence, the following constraints are not placed in $\mathcal{P}_S$:

(i) For an equality constraint $c^Tx - d = 0$, all the constraints $(a^Tx - b)(c^Tx - d) = 0$;

(ii) For an inequality $c^Tx - d \geq 0$, all the constraints $(a^Tx - b)(c^Tx - d) \geq 0$ with a and b such that $b \leq \sum_{i=1}^n \min\{0, a_i\}$.

Indeed, these constraints are already suitable combinations of elements of $\mathcal{P}_S$:

(i) $(a^Tx-b)(c^Tx-d) = \sum a_i(c^Tx-d)x_i - b(c^Tx-d)$ is a suitable combination of $(c^Tx-d)x_i = 0$ and $c^Tx - d = 0$ which belongs to $\mathcal{P}_S$;

(ii) $(a^Tx-b)(c^Tx-d) \geq 0$ is a suitable combination of (c^Tx-d), $(c^Tx-d)x_i$ and $(c^Tx-d)(1-x_i), i = 1, ..., n$:

$$\begin{aligned}(a^Tx - b)(c^Tx - d) = \quad & \sum_{i=1}^n \max\{0, a_i\}(c^Tx - d)x_i \\ & - \sum_{i=1}^n \min\{0, a_i\}(c^Tx - d)(1 - x_i) \\ & + (\sum_{i=1}^n \min\{0, a_i\} - b)(c^Tx - d)\end{aligned}$$

In summary, we discussed how to build $\mathcal{P}_S$ and how to eliminate some useless constraints. More formally, the notion of domination of a quadratic constraint by another is assessed by the S-lemma [212]. Thus, a constraint $\mathrm{q}_j(.;Q_j)$ with Q_j not positive semidefinite, dominates another one $\mathrm{q}_k(.;Q_k)$ if and only if there exists $\lambda \geq 0$ such that $Q_k - \lambda Q_j \preccurlyeq 0$. One could think of detecting in this way the pair-wise dominance but in practice this is computationally prohibitive.

Finally, some valid constraints proposed in the literature, such as the hypermetric inequalities, are a priori not included in our approach. In order to measure the impact of this lack, we will consider the possibility of adding them directly into our set $\mathcal{P}_S$.

5.2.3 Application to the Nuclear Outages Problem

In this section, we report on computational experiments conducted to analyse the performance of our approach on two classes of instances of the NOSP : some small randomly generated instances, called *working instances* and some *real-life instances*.

This section is organized as follows. We first explain in detail the benchmark of instances employed. The second paragraph gives much practical informations about the computational experiments. Then, we report the numerical results and we discuss further considerations about them. Finally, we analyse the generated cuts and we experiment to add some of them directly to some new instances.

5.2.3.1 Model summary

We propose here a formulation of the model that emphasizes the structure of the problem. The indices of the constraints have been omitted for sake of clarity. Thus, we have a 0/1 QCQP with linear constraints :

$$\begin{cases} \min & x^T P_0 x + 2p_0^T x & (4.13) \\ \text{subject to} & A_1 x = b_1 & (4.1) \\ & b_2 \leq A_2 x \leq b_2' & (4.16) \\ & A_3 x \leq b_3 & (4.6) \\ & x^T P_4 x + p_4^T x + \pi_4 \leq 0 & (4.10) \\ & x \in \{0,1\}^n \end{cases} \tag{5.10}$$

5.2.3.2 The benchmark of instances

The first class of instances are randomly generated instances of 0/1-QCQP. In order to get close of NOSP, these instances contain linear assignment-type equality constraints and linear two-way inequality constraints. Furthermore, all the variables are required to be binary.

Instead of restricting ourself to the case of a convex objective function and non-convex quadratic constraints, as in the NOSP, we allow ourselves a slight generalization, leading to the nine following classes of instances described in the Table 5.2. For each class, we specify whether the objective and constraints are linear, convex quadratic or nonconvex quadratic, in which case we simply write "quadratic". 50 instances of each class are generated, differing in their number of binary variables, that varies from 11 up to 60.

The *real-life* instances are extracted from actual real-life data sets. Their size varies with the number of nuclear power plants (10 or 20), with the time-horizon (3 or 4 outages per plant) and with the size of the search space of each outages (7, 9, 11, 13 or 15 possible beginning dates). This is summarized in Table 5.3, that contains the following columns :

- Column 1 : the class ;
- Column 2 : the number of nuclear power plants ;

Class	Objective	Constraint
1	linear	linear
2	convex	linear
3	quadratic	linear
4	linear	convex
5	convex	convex
6	quadratic	convex
7	linear	quadratic
8	convex	quadratic
9	quadratic	quadratic

Table 5.2: Classification of the working instances

- Column 3 : the number of considered outages in the time horizon ;
- Column 4 : the size of the search space of each outages ;
- Column 5 : the number of binary variables of the resulting problem ;
- Column 6 : the number of linear constraints of the resulting problem ;
- Column 7 : the number of quadratic constraints of the resulting problem.

For each class of instance, we built 25 instances, that differ in the demand (5 scenarios) and in the search spaces (5 possibilities).

Class	# nuclear plants	# outages per plant	Size of search spaces	# binary variables	# linear constraints	# quadratic constraints
1	10	3	7	259	104	7
2	10	3	9	332	122	14
3	10	3	11	405	143	24
4	10	3	13	478	157	34
5	10	3	15	551	166	42
6	10	4	7	322	128	9
7	10	4	9	413	150	18
8	10	4	11	504	176	31
9	10	4	13	595	193	43
10	10	4	15	686	205	53
11	20	3	7	497	183	7
12	20	3	9	638	205	16
13	20	3	11	779	235	27
14	20	3	13	920	255	42
15	20	3	15	1061	268	55
16	20	4	7	623	228	9
17	20	4	9	800	258	20
18	20	4	11	977	295	34
19	20	4	13	1154	319	53
20	20	4	15	1331	338	68

Table 5.3: Size of the real-life instances

5.2.3.3 Description of the computational experiments

We compare the gap obtained with the following relaxations :

- LR : the linear relaxation;
- SDP0 : the initial semidefinite relaxation;
- SDP1 : the reinforced semidefinite relaxation, with a classical separation;
- SDP2 : the reinforced semidefinite relaxation, with the semidefinite separation problem;

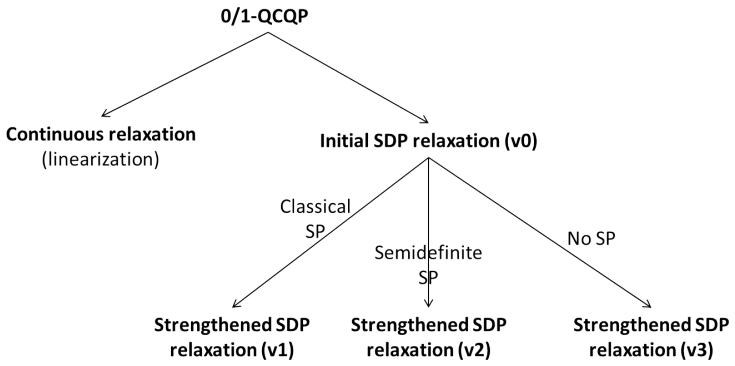

Figure 5.1: Outline of the different relaxations

– SDP3 : the reinforced semidefinite relaxation, by adding all the violated constraints of the set $\mathcal{P}_S$.

The classical separation that is used for the relaxation SDP1 consists of selecting the most violated constraint among $\mathcal{P}_S$ at each iteration. For sake of clarity, these relaxation are illustrated in the diagram of Figure 5.1. The last relaxation (v3) is obtained by adding all the valid quadratic constraints to our problem. The aim is to measure the potentiality of our method, by providing a bound on its gain. The linear relaxation is obtained by linearizing the quadratic terms with the RLT approach and replacing $x_i \in \{0, 1\}$ by $x_i \in [0, 1]$.

Regarding the semidefinite relaxation, the initial one (v0) is defined in paragraph 3.3.3.1, whereas (v2) correspond to the reinforced versions defined in paragraph 5.2.2.

To compute the gap, we start by solving exactly the problem, which is possible on small instances by using the commercial solver CPLEX 12.1. For larger instances, we compute a feasible solution by setting a maximal running time of two hours per computation. This solver is also used to solve the linear relaxation. As a SDP solver, we use DSDP 5.8. All our experiments were performed on a 2.5 GHz Intel x86 with 16 GB memory. Finally, we compute the gap using the following formula :

$$G_R = \frac{p^* - p_R}{p^*} \tag{5.11}$$

where p^* is the value of the best feasible solution computed within 2 hours by CPLEX and p_R is the result of the relaxation R at hand.

5.2.3.4 Numerical results

In Table 5.4, we provide the gap G_R obtained on the working instances, on average on the 50 instances of each class, where R is one on the five studied relaxations. In order to evaluate more precisely the tightness of the relaxations, we also report in column entitled T_R, the number of instances where the gap is less than 1%.

On instances of classes 1 to 3, where the constraints are linear, the gap of the linear relaxation is quite small. Consequently, there is no room for progress for other relaxations. On the other hand, when this gap becomes larger, the semidefinite relaxation is much more efficient than the linear one. With convex constraints (instances of classes 4 to 6), the initial semidefinite relaxation closes the gap from roughly 80% to 15%, so there is no need for reinforcement. Finally, when non convex quadratic constraints are added to the problem, adding some cuts is necessary to improve the performance of the semidefinite relaxation.

Class	G_{LR}	T_{LR}	G_{SDP0}	T_{SDP0}	G_{SDP1}	T_{SDP1}	G_{SDP2}	T_{SDP2}	G_{SDP3}	T_{SDP3}
1	6.24%	0	6.18%	0	5.64%	0	5.84%	0	4.74%	1
2	6.36%	0	6.21%	0	5.62%	1	5.94%	0	4.83%	3
3	5.47%	1	6.16%	0	3.87%	5	4.41%	2	3.30%	7
4	37.27%	0	10.97%	1	9.64%	2	10.43%	1	8.90%	3
5	40.51%	0	10.69%	4	8.08%	6	9.44%	4	6.79%	6
6	40.62%	0	13.50%	0	10.24%	1	11.43%	1	9.26%	1
7	17.49%	0	18.21%	0	13.53%	0	14.24%	0	11.16%	1
8	19.31%	0	19.42%	0	14.99%	0	16.11%	0	13.05%	1
9	20.90%	0	20.93%	0	14.04%	1	14.99%	1	11.14%	2

Table 5.4: Gap of the different relaxations on working instances

Not surprisingly, obtaining such a gap has a cost in terms of running time. In Table 5.5, we report this value in seconds (RT_R), except for the linear relaxation and the initial semidefinite relaxation, since the latter are computed almost instantaneously. For the reinforced version v1 and v2, we also provide the number of iterations (N_R).

Class	RT_{SDP1}	N_{SDP1}	RT_{SDP2}	N_{SDP2}	RT_{SDP3}
1	2.0	11.41	3 278.3	5.59	649.7
2	4.8	17.26	4 432.5	4.40	862.6
3	16.2	48.60	33 147.3	26.94	542.8
4	6.3	29.00	7 547.3	8.00	749.9
5	7.1	30.00	8 803.6	9.92	830.1
6	21.1	57.72	39 077.8	25.48	813.9
7	20.7	52.62	60 626.0	38.22	617.1
8	27.1	62.02	59 947.5	40.00	936.5
9	24.8	65.76	71 973.3	50.12	792.1

Table 5.5: Running time and number of iterations of the different procedures

Thus, we observe that the relaxation SDP2 is not efficient compared to SDP1, even if it reaches almost the same gap in a much smaller number of iterations. But the additional running time of each iteration does not justify that we continue in that direction and in the sequel, we consider only the semidefinite relaxation SDP1.

Before that, we carry out an analysis of the selected constraint at the first iteration of SDP1 and SDP2, in 3 cases :

- when these separations problems leads exactly to the same result;
- when SDP1 yields a better constraint than SDP2;
- when SDP2 yields a better constraint than SDP1.

In the first case, the most violated constraint of $\mathcal{P}_S$ is the product of two upper bounds constraints : $(1 - x_i)(1 - x_j) \geq 0$. The semidefinite separation chooses this constraint with a coefficient $\lambda_i = 0.5$ and makes it semidefinite by adding the constraint $x_i^2 - x_i \leq 0$ and $x_j^2 - x_j \leq 0$ with coefficients 0.25. It is therefore normal that the semidefinite relaxations yields the same bounds, since the $x_i^2 - x_i \leq 0$ and $x_j^2 - x_j \leq 0$ belongs to $\mathcal{P}_C$ and the semidefinite relaxation satisfies all the convex combination of constraints of $\mathcal{P}_C$.

In the second case, the most violated constraint of $\mathcal{P}_S$ is the product of two lower bounds constraints, which is also found in the semidefinite separation solution, combined with two other products of bound constraints and three binary constraints. Finally, in the third case, the most violated constraint of $\mathcal{P}_S$ is the product of a bound constraint and of a weight constraint. Once again, this constraint is used in the semidefinite separation and combined with a large number of products of bound constraint and binary constraint to get a convex constraint.

Thus, we observe that the binary constraints are always used to make the solution convex. This motivates us to experiment a set $\mathcal{P}_S$ without the binary constraints, in order to force the semidefinite

separation to be more "creative", without penalizing the obtained SDP relaxation since the binary constraints belong to $\mathcal{P}_C$, but the results were inconclusive.

Finally, we proceeded to a complementary experiment with the working instances. As mentioned earlier, the triangular inequalities are not natively included in the set $\mathcal{P}_S$, so we added them in order to assess the impact of this lack. Then we observe an increase in the number of iterations (N) and of the running time (RT), without significantly improving the gap. Consequently, we keep our initial set $\mathcal{P}_S$ without these inequalities.

At this point, we have all the elements to proceed to numerical experiments on real-life instances of the Nuclear Outages Problems, as reported in Table 5.6. Whenever possible, i.e., when a feasible integer solution has been found by CPLEX in less than 2 hours, we compute the gap (G_R) of each relaxation R. The first column (#0/1 OK) gives the number of instances over 25 that are in this case. Besides, for each semidefinite relaxation, we compute a relative enhancement E of the relaxation R w.r.t. the linear relaxation, whose corresponding formula is as follows :

$$E_R = \frac{p_R - p_{LR}}{p_{LR}} \tag{5.12}$$

Class	# 0/1 OK	G_{LR}	G_{SDP0}	G_{SDP1}	E_{SDP0}	E_{SDP1}
1	25	14.68%	2.57%	2.36%	14.22%	14.46%
2	23	16.87%	4.24%	4.08%	15.28%	15.47%
3	23	19.94%	6.89%	6.72%	16.31%	16.52%
4	19	28.79%	8.66%	8.58%	17.17%	17.27%
5	24	25.20%	11.87%	11.81%	17.92%	18.00%
6	22	15.24%	2.76%	2.62%	14.70%	14.91%
7	22	18.11%	4.73%	4.57%	16.28%	16.49%
8	22	22.62%	7.66%	7.52%	17.75%	17.94%
9	18	31.28%	11.56%	11.50%	18.65%	18.73%
10	23	32.74%	15.41%	15.36%	20.08%	20.15%
11	25	22.59%	1.77%	1.69%	26.94%	27.04%
12	21	24.69%	2.98%	2.92%	28.87%	28.95%
13	22	27.47%	5.16%	5.09%	31.03%	31.13%
14	24	30.22%	7.05%	6.99%	33.33%	33.42%
15	21	34.97%	12.24%	12.21%	35.14%	35.19%
16	21	23.47%	1.76%	1.70%	28.14%	28.22%
17	20	26.09%	3.28%	3.23%	30.76%	30.83%
18	19	28.70%	5.34%	5.29%	32.98%	33.06%
19	22	33.26%	8.73%	8.67%	36.63%	36.72%
20	16	38.31%	14.66%	14.63%	38.20%	38.24%

Table 5.6: Comparison of the semidefinite relaxations to the linear relaxation on real-life instances

Note that the reinforced relaxation SDP1 is computed with a maximum of 100 additional cuts. This reinforcement produces on average an enhancement of 0.27% of the semidefinite relaxation SDP1 w.r.t. the initial semidefinite relaxation SDP0. On the whole, this yields an average improvement of 24.64% w.r.t the linear relaxation. This might seem not very significant but let us mention that the variable part of the cost is very small over the feasible set of solutions. By denoting $\mathcal{F}$ the feasible set and $\epsilon = \max_{x\in\mathcal{F}} f_0(x) - \min_{x\in\mathcal{F}} f_0(x)$, we have ϵ that is very small w.r.t $\min_{x\in\mathcal{F}} f_0(x)$. Consequently, the variation on the gap are also very small.

5.2.3.5 Analysis of the selected cutting planes

One may also think of our approach as a tool for identifying the most useful cutting planes. Thus, we may add directly these constraints in our semidefinite relaxation. In order to proceed to such an analysis, we group the constraints into classes, depending on the linear constraints that are involved :

– *Bound* means that the constraints is initially a bound constraint;

– *Lapping* concerns the maximal lapping constraints (4.7), which is linear when $\mathcal{E}_{i,j}$ is such that only one part of the disjunction is feasible ;

– *MaxTime* and *MinTime* are the constraints linking two successive outages (4.16) ;

– *Parall* are the constraints on the maximum number of parallel outages (4.6).

These 5 categories of linear constraints yields 15 categories for the products. In Table 5.7, we report the class of linear constraints (*Cst1* and *Cst2*) that yields the category at hand. The fourth column is the percentage of these categories, on average on all the instances. Finally, the last column gives the number of selected constraints of each categories on all the instances.

Class	Cst1	Cst2	Repartition	# selected csts
1	Bound	Bound	80.27%	0
2	Lapping	Lapping	0.15%	16142
3	MaxTime	MaxTime	0.06%	3215
4	MinTime	MinTime	0.00%	312
5	Parall	Parall	0.13%	0
6	Bound	Lapping	6.46%	653
7	Bound	MaxTime	4.22%	52
8	Bound	MinTime	0.82%	31
9	Lapping	MaxTime	0.19%	14203
10	Lapping	MinTime	0.03%	6033
11	MinTime	MaxTime	0.02%	1797
12	Bound	Parall	5.71%	0
13	Lapping	Parall	0.20%	173
14	MaxTime	Parall	0.13%	10
15	MinTime	Parall	0.03%	7

Table 5.7: 15 categories of additional constraints

We observe that the classes 2 and 9 are selected a large number of times w.r.t the other classes, whereas they represent a low proportion of the whole constraints. There are also a significant number of constraints belonging to classes 10, 3 and 11. Surprinsingly, RLT-type constraints, i.e., class 1 are never selected, whereas they represent 80% of the whole contraints. The constraints involving a *Parall* constraint are also rarely used.

This suggests that the most relevant constraints are those made of linear constraints that involve a high number of variables and with high coefficients. Indeed, the *time* and *lapping* constraints involves the variables of two outages, with the value of the time step as coefficient ($t \in \{1, ..., N_t\}$, whereas the constraint on the maximum number of parallel outages and the bound constraint use 1 as coefficient.

Indeed, it is clear that the more $\|Q\|_F$, the Frobenius norm of Q, is large, the more the constraint $\mathrm{q}(x; Q) \leq 0$ is violated. To avoid this, we experimented to normalize the violation γ_i associated to the constraints $\mathrm{q}_i(x; Q_i) \leq 0$ by dividing it by $\|Q_i\|_F$, but the results were inconclusive.

In addition to this categorization, we are interested in an other indicator, that reflects if some variables are shared by the two linear constraints. Let q be a quadratic constraint obtained as the product of the two linear constraints $a^T x \leq b$ and $c^T x \leq d$:

$$t_{\mathrm{q}} = \frac{\#\{i : a_i \neq 0, c_i \neq 0\}}{\#\{i : a_i \neq 0\} + \#\{i : c_i \neq 0\}} \tag{5.13}$$

The curve of the Figure 5.2 gives this ratio at each iteration on average on all the instances. We observe that this ratio decreases through iterations. Furthermore, over all the selected constraints, 28.4% of them have a non-nul ratio, whereas over all the built constraints, only 0.98% of the built constraints have this property. We deduce from this that the more the initial linear constraints share some variables, the more the quadratic constraints are efficient.

This comes from the fact that, the more the variables are shared between the two linear constraints, the more there are some squares in the obtained constraints, which are constrained to be equal

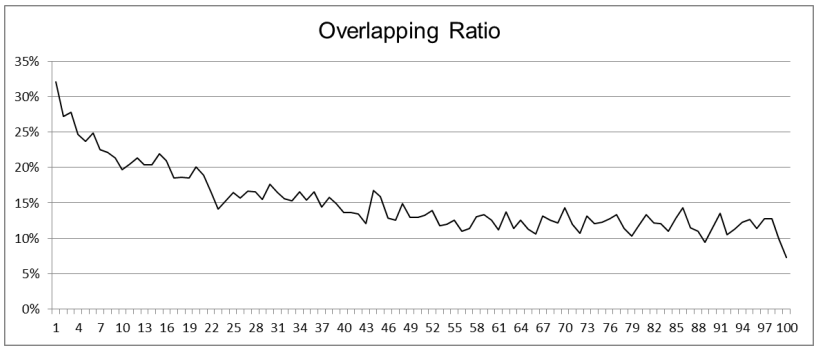

Figure 5.2: The overlapping ratio of the selected constraints

to the projected variable. On the other hand, most of the non-square variables are not constrained at all and can therefore take any value.

Following these elements, we build a new semidefinite relaxation similarly to the initial semidefinite relaxation but with the additional constraints of class 2 and 9 with an overlapping ratio greater than 0.1. We denote SDP4 this new relaxation and report corresponding numerical results in Table 5.8, where the second column gives the average number of additional constraints. For each class given in the first column and for each relaxation $R = SDP4$, $R = SDP0$ or $R = SDP1$, we provide E_R the enhancement w.r.t. to the linear relaxation, computed using the formula (5.12). For $R = SDP4$, $R = SDP0$, we also provide the running time RT_R. We do not provide the running time of RT_{SDP1} since this value includes several iterations and is therefore not comparable to the running time of a single relaxation.

Class	# additional csts	E_{SDP4}	RT_{SDP4}	E_{SDP0}	RT_{SDP0}	E_{SDP1}
1	161.60	14.39%	25.88	14.22%	9.64	14.46%
2	167.20	15.41%	63.92	15.28%	22.40	15.47%
3	146.60	17.26%	108.28	17.10%	49.96	17.32%
4	114.80	21.45%	165.04	21.40%	76.52	21.52%
5	82.40	17.94%	213.8	17.92%	107.96	18.00%
6	192.40	14.85%	53.4	14.70%	18.84	14.91%
7	205.40	16.43%	124.08	16.28%	45.80	16.49%
8	173.60	18.71%	216.52	18.57%	89.36	18.76%
9	133.00	22.40%	286.12	22.35%	145.96	22.45%
10	95.60	21.72%	418.12	21.70%	222.68	21.78%
11	255.00	27.00%	143.28	26.94%	58.80	27.04%
12	278.40	28.92%	313.2	28.87%	157.60	28.95%
13	264.00	31.10%	544	31.03%	290.00	31.13%
14	220.60	33.36%	834.72	33.33%	491.08	33.42%
15	161.40	35.15%	1366.84	35.14%	787.64	35.19%
16	304.80	28.20%	271.8	28.14%	103.20	28.22%
17	340.80	30.81%	607.76	30.76%	233.12	30.83%
18	312.40	33.05%	1097.6	32.98%	483.72	33.06%
19	252.40	36.65%	1930.44	36.63%	953.64	36.72%
20	191.00	38.21%	3288.08	38.20%	1650.36	38.24%

Table 5.8: Reinforcement of the initial semidefinite relaxation

Remark that when the search spaces spread (instances of classes 5, 10, 15 and 20), the number of linear lapping constraints decreases and so does the corresponding quadratic constraints. For this reason, the gain on the gap is quite small on these instances, w.r.t the gain on the other instances. Finally, on average, we get a enhancement of 25.15% w.r.t the linear relaxation.

To overcome the problem of the small number of linear lapping constraints, additional constraints may be selected rather in the classes involving *MinTime* and *MaxTime* constraints. Finally, in order

to reduce the computational time on large instances, it might be worthwhile to be more restrictive on the overlapping ratio, in order to reduce the number of additional constraints.

5.2.4 Conclusion

In this section, we propose a tool for generating and analysing valid cuts to reinforce the semidefinite relaxation of a QCQP. These cuts are generated as pairwise products of the linear constraint of the problem and added to the QCQP, before applying the semidefinite relaxation. At each iteration, the constraint that is the most violated by the incumbent solution, is thus selected.

We experimented several variants of this basic idea. First, we try to select the most violated convex quadratic constraints, in order to be more efficient in the semidefinite relaxation. Indeed, a convex quadratic constraint is necessarily satisfied by the projected solution of the semidefinite relaxation. We obtain a reinforcement of the semidefinite relaxation but the computational cost is too high compared to the improvement of the bound. We also try to consider the triangular inequalities into our initial set of constraints but the impact on the bound was very low.

We applied this scheme to a real-life QCQP, the Nuclear Outages Problem, a problem characterized by assignement constraints, non convex quadratic constraints that model some disjunctions and a convex objective function. On this problem, the semidefinite relaxation improve by 25.15% the gap of the linear relaxation. This enhancement, combined to a Branch & Bound or to another enumerative approach, could contribute to tackle the problem even if a hard work on computational time would be necessary for this. Indeed, the current resolution with CPLEX is not satisfying, since there are a number of instances where, after two hours, CPLEX is very far from the optimal solution. On 12.4% of instances, CPLEX can not even produce a feasible solution within this time limit.

In conclusion, the main advantage of our approach is that it is not sensitive to structural properties of the problem. In an real-world framework, this method can be used to determine the most relevant cuts on working instances. Then, these cuts can be added directly to operational instances to provide an efficient semidefinite relaxation.

5.3 SDP relaxations for three possible formulations of the maximal lapping constraint

In this section, we aim at comparing some classical reinforcements of the standard SDP relaxation in order to determine the most appropriate for each of the three versions of the model 4 of the NOSP. These models varies in their formulations of the minimal lapping constraint. We recall that these constraint is a disjunctive constraint of the form $a^T x \notin]b, c[$, where x is a binary vector. Then the three models are as follows :

- The model 4-1 uses the "big M" formulation : $a^T x - My \leq b, \;\; a^T x + M(1-y) \geq c, \;\; y \in \{0,1\}$;
- The model 4-2 uses the pairwise exclusion formulation : $x_i + x_j \leq 1$ for all (i,j) such that $a_i + a_j \in [b, c]$;
- The model 4-3 uses the quadratic formulation : $(a^T x - b)(a^T x - c) \geq 0$.

5.3.1 Various SDP relaxations

The main objective of this section is to compare various possible SDP relaxations that were proposed in the literature. They are constructed over two steps. First we built a QCQP equivalent to the initial QCQP by adding valid quadratic constraints and removing redundant constraints. Then we apply the standard SDP relaxation (see Paragraph 3.3.2). Thus, we compare the following relaxations :

- SDP-1 : the standard SDP relaxation ;

- SDP-2 : the standard SDP relaxation with squared equalities (replace $a_i^T x - b_i = 0$ by $(a_i^T x - b_i)^2 = 0$) ;
- SDP-3 : the standard SDP relaxation with squared inequalities (replace $b_i \leq a_i^T x \leq c_i$ by their square $(a_i^T x - b_i)(a_i^T x - c_i) \leq 0$) ;
- SDP-4 : the standard SDP relaxation with squared equalities and inequalities ;
- SDP-5 : the standard SDP relaxation with positivity constraints (corresponding to $x_i x_j \geq 0$);
- SDP-6 : the standard SDP relaxation with the 4 classes of RLT constraints (corresponding to $x_i x_j \geq 0$, $x_i x_j \leq x_i$,$x_i x_j \leq x_j$ and $x_i x_j \geq 1 - x_i - x_j$);
- SDP-7 : the standard SDP relaxation with Sherali-Adams constraints (multiply all the linear equalities and inequalities by x_i and $1 - x_i$);
- SDP-8 : the standard SDP relaxations with the triangle inequalities (see Paragraph 3.3.3.3);
- SDP-9 : the combination of the relaxations 4 and 7 ;
- SDP-10 : the combination of the relaxations 6 and 7;
- SDP-11 : the combination of the relaxations 4 and 6;
- SDP-12 : the combination of the relaxations 4, 6 and 7;

For each relaxation SDP-i, we also compute the equivalent linear relaxation LP-i, obtained by applying the Reformulation-Linearization Technique (see Appendix 3.5) to the QCQP to which we apply the standard SDP relaxation.

We remark that having bounded variables allows us to assume that all the linear inequalities are range constraints. Indeed, if it is not the case, the complementary bound can easily be computed from the variables bounds.

Regarding inequalities, one also might think of converting them into equalities : $b \leq a^T x \leq c$ is equivalent to $a^T x - y = 0$, with y a bounded variable : $y \in [b, c]$. If the equality remains under its linear form, then it is strictly equivalent to write the bound constraint under its linear or quadratic form. Indeed, the only quadratic term is y^2 and therefore we can constraint the associated component of the SDP variable without impacting the final solution. In either case, this is of no interest since it is strictly equivalent to the standard relaxation, except that an additional variable is added. Then, it remains two possibilities :

- square equality $(a^T x - y)^2 = 0$ and linear inequality $b \leq y \leq c$;
- square equality $(a^T x - y)^2 = 0$ and square inequality $(b - y)(c - y) \leq 0$.

The second one is exactly equivalent to the square formulation of the original constraint, but leads to several solver failures, and the first one is less tight. Consequently, it seems more appropriate not to convert inequalities into equalities.

5.3.2 Numerical results and analysis

This section presents an analysis of the obtained results from three different angles. First we provide a description of the data set. Then, we compare the SDP relaxations to each other in order to find the best compromise between quality of the bounds and computation time. Second, a comparison with the equivalent LP relaxation is provided so as to assess the adequacy of SDP. Finally, we compare the relaxations obtained for the three formulations of the maximal lapping constraint.

5.3.2.1 Elaboration of the data sets

In the experiment below, we use 600 data sets of 6 different sizes. The data sets of a same size differ only by the production cost and the maximal power of the plants. The table 5.9 summarizes the characteristics of the different sizes of instances. For each size given in the first column, we provide :

– Column 2 : the number of time steps of the time horizon (N_t) ;
– Column 3 : the number of power plants (N_ν) ;
– Column 4 : the number of sites (N_g) ;
– Column 5 : the minimal space between outages, the same for each site ($-N_k^l$) ;
– Column 6 : the duration of the outages, the same for each outages ($\delta_{i,j}$).

Size	# time steps	# power plants	# sites	Minimal space	Outage duration
1	6	2	1	1	2
2	10	3	1	2	2
3	14	4	1	2	2
4	18	5	2	3	2
5	22	6	2	3	2
6	26	8	3	4	2

Table 5.9: Gap of the different relaxations on working instances

The search spaces, i.e., the set $\mathcal{E}_{i,j}$ where the beginning outages dates may lie, are defined as $\{1, ..., N_t - \delta_{i,j}\}$, which corresponds to all the possible dates in the period with the exception of the dates for which a part of the outage is beyond the horizon time.

The 100 instances of the same size differ by the production cost. For each time step, this cost is defined through a coefficient γ_t such that $q_t(x) = \gamma_t x$. In order to simulate the seasonality of the marginal costs, for each instance, the coefficients γ_t are drawn from a Gaussian distribution with mean $\mu_t = 0.144(t - N_t/2)^2 + 10$ and covariance $\mu_t/5.0$. This has been chosen in order to attain 100 for $t = 50$ and $N_t = 50$. Regarding the maximal powers, there are also drawn from a Gaussian distribution, with mean 1.0 and covariance 0.1.

5.3.2.2 Comparison of the SDP relaxations to each other

We start by reporting in Table 5.10 the robustness of the different SDP relaxations SDP-i, i.e., the number on instances that succeeded out of a total of 100. We also report the size of the instances in terms of number of variables (# var) and constraints (# cst), of the original PLNE to give an idea of the size of the instances. The first column indicates the class of instances where "i-j" is the class of instances of size "i" with the model $4 - j$ of the maximal lapping constraint. Note that this experiment was performed by using CPLEX 12.1 [143] and CSDP 6.1 [53] as solvers for LP and SDP respectively. SDP was solved on a 2.5 GHz Intel x86 and LP on an Intel Core i7 at 2.13 GHz.

The reason for failure lies in memory storage problems since the SDP is too large to be stored in memory. In the light of these elements, we abandon the relaxation SDP-8 (with the triangle inequalities), that involves too many constraints. This is confirmed by the results that we obtained on classes 1-1 and 1-2 since these relaxations barely improve the obtained bound.

In Table 5.11 and 5.12, for each relaxation SDP-i, we provide the size of the obtained SDP, under the form "$v|c$". If the SDP is under the form 2.4, v equals the sum of the sizes (in number of rows) of the primal variables X_k and c is the number of primal constraints.

We report in Tables 5.13 and 5.14 the gaps of the SDP and LP relaxations, calculated as the average of the gaps of the 100 instances of each class. The gaps are calculated using the formula $(p_r^* - p^*)/p^*$ where p_r^* is bound obtained with the relaxation and p^* the optimal value of the integer problem.

Remark that the average that are computed on less than 100 instances can not be compared to each other. For instance, the average gap obtained with SDP-11 might be less than the average gap of SDP-12 whereas, for each instance where those both relaxation succeeded, SDP-12 is tighter than SDP-11.

We make the following observations :

Class	# var	# cst	SDP-1	SDP-2	SDP-3	SDP-4	SDP-5	SDP-6	SDP-7	SDP-8	SDP-9	SDP-10	SDP-11	SDP-12
1-1	11	4	100	100	100	100	100	100	100	100	100	100	100	100
1-2	10	21	100	100	100	100	100	100	100	100	100	100	100	100
1-3	55	183	100	100	100	100	100	100	100	100	100	100	100	100
2-1	30	9	100	100	100	100	100	100	100	100	100	100	33	100
2-2	27	156	100	100	100	100	100	100	100	100	100	100	100	100
2-3	378	1410	100	100	100	100	100	100	100	100	100	100	23	100
3-1	58	16	100	100	100	100	100	100	100	0	100	100	0	100
3-2	52	478	100	100	100	100	100	100	0	0	0	0	100	0
3-3	1378	5314	100	100	100	100	100	100	100	0	100	100	0	100
4-1	89	13	100	100	100	100	100	100	0	0	0	0	0	0
4-2	85	537	100	100	100	100	100	100	0	0	0	0	0	0
4-3	1921	7353	100	100	100	100	100	100	0	0	0	0	0	0
5-1	132	18	100	100	100	100	100	0	0	0	0	0	0	0
5-2	126	1020	100	100	100	100	100	0	0	0	0	0	0	0
5-3	4032	15636	100	100	100	100	100	0	0	0	0	0	0	0
6-1	208	24	100	100	100	100	0	0	0	0	0	0	0	0
6-2	200	1968	71	72	72	100	0	0	0	0	0	0	0	0
6-3	7600	29616	72	76	72	100	0	0	0	0	0	0	0	0

Table 5.10: Robustness of the different SDP relaxations

Class	SDP-1	SDP-2	SDP-3	SDP-4	SDP-5	SDP-6
1-1	14|16	14|16	13|15	13|15	69|71	234|236
1-2	30|32	30|32	30|32	30|32	75|77	210|212
1-3	12|14	12|14	12|14	12|14	57|59	192|194
2-1	37|40	37|40	34|37	34|37	472|475	1777|1780
2-2	181|184	181|184	181|184	181|184	532|535	1585|1588
2-3	31|34	31|34	31|34	31|34	382|385	1435|1438
3-1	71|75	71|75	65|69	65|69	1724|1728	6683|6687
3-2	527|531	527|531	527|531	527|531	1853|1857	5831|5835
3-3	59|63	59|63	59|63	59|63	1385|1389	5363|5367
4-1	98|103	98|103	94|99	94|99	4014|4019	15762|15767
4-2	618|623	618|623	618|623	618|623	4188|4193	14898|14903
4-3	90|95	90|95	90|95	90|95	3660|3665	14370|14375
5-1	145|151	145|151	139|145	139|145	8791|8797	34729|34735
5-2	1141|1147	1141|1147	1141|1147	1141|1147	9016|9022	32641|32647
5-3	133|139	133|139	133|139	133|139	8008|8014	31633|31639
6-1	225|233	225|233	217|225	217|225	21753|21761	86337|86345
6-2	2161|2169	2161|2169	2161|2169	2161|2169	22061|22069	81761|81769
6-3	209|217	209|217	209|217	209|217	20109|20117	79809|79817

Table 5.11: Sizes of the different SDP relaxations (1)

- On the linear models ($4-1$ and $4-2$) SDP-1 is equivalent to the linear relaxation LP-1 whereas on the quadratic model ($4-3$), SDP-1 is worth than LP-1 ;
- SDP-2 and SDP-3 yield very little improvement w.r.t. SDP-1;
- SDP-4 has almost the same size than SDP-1 and yields a significantly better gap, especially with the model $4-2$;
- SDP-5 and SDP-6 do not improve SDP-1 and are less robust;
- SDP-9 does not improve SDP-4;
- SDP-12 does not improve SDP-10;
- SDP-11 suffers from a lack of robustness and is not more effective than SDP-10.

From these observations, we draw the following conclusions. First the standard SDP relaxation is not satisfactory and the SDP-4 relaxation is to be preferred in any case. Indeed, for a same size of SDP and a computational time of the same order of magnitude, the gap yielded by SDP-4 is on average

Class	SDP-7	SDP-8	SDP-10	SDP-11	SDP-12
1-1	58|104	57|103	278|324	233|235	277|323
1-2	410|452	410|452	590|632	210|212	590|632
1-3	12|54	12|54	192|234	192|194	192|234
2-1	397|580	394|577	2137|2320	1774|1777	2134|2317
2-2	8443|8608	8443|8608	9847|10012	1585|1588	9847|10012
2-3	31|196	31|196	1435|1600	1435|1438	1435|1600
3-1	1463|1931	1457|1925	8075|8543	6677|6681	8069|8537
3-2	49823|50243	49823|50243	55127|55547	5831|5835	55127|55547
3-3	59|479	59|479	5363|5783	5363|5367	5363|5783
4-1	1522|2417	1518|2413	17186|18081	15758|15763	17182|18077
4-2	91058|91913	91058|91913	$> 10^5$|$> 10^5$	14898|14903	$> 10^5$|$> 10^5$
4-3	90|945	90|945	14370|15225	14370|14375	14370|15225
5-1	3313|4903	3307|4897	37897|39487	34723|34729	37891|39481
5-2	$> 10^5$|$> 10^5$	$> 10^5$|$> 10^5$	$> 10^5$|$> 10^5$	32641|32647	$> 10^5$|$> 10^5$
5-3	133|1651	133|1651	31633|33151	31633|31639	31633|33151
6-1	6881|10217	6873|10209	92993|96329	86329|86337	92985|96321
6-2	$> 10^5$|$> 10^5$	$> 10^5$|$> 10^5$	$> 10^5$|$> 10^5$	81761|81769	$> 10^5$|$> 10^5$
6-3	209|3417	209|3417	79809|83017	79809|79817	79809|83017

Table 5.12: Sizes of the different SDP relaxations (2)

19.1%, compared with 22.5% with SDP-1. The figure 5.3 compares the gaps and running times of the first four SDP relaxations.

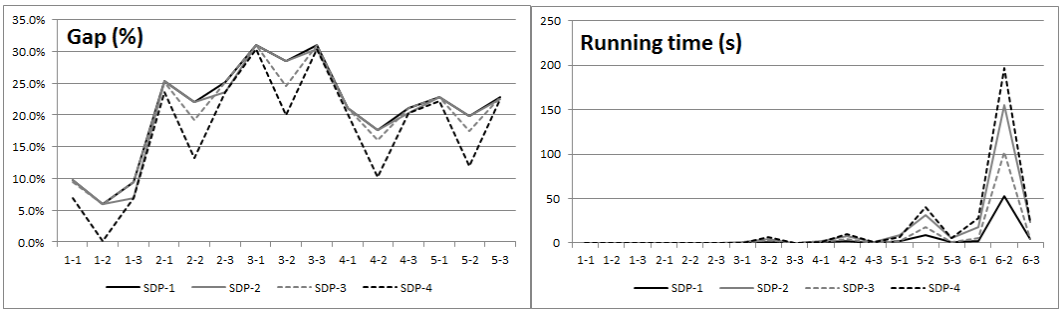

Figure 5.3: Comparison of the gaps and running time of the relaxations SDP-1, SDP-2, SDP-3 and SDP-4

It is interesting to note that SDP-2 and SDP-3 are rather ineffective w.r.t SDP-4 whereas SDP-4 is a combination of those both relaxations. Regarding the LP relaxations, these transformations do not bring anything, as illustrated on Figure 5.4. The SDP relaxations that retains some interest are therefore :

- SDP-4, obtained by "squaring" all the linear constraints ;
- SDP-7, obtained by adding the product of the linear constraints (except bounds constraints) by the bounds constraints, called Sherali-Adams constraints;
- SDP-10, obtained by adding the Sherali-Adams and the RLT contraints.

The fact that SDP-12 is not more effective than SDP-10 show that once all the product of linear constraint have been added, it is useless to consider the square form of the linear constraints. For equality constraints, this is explained by the fact that the square of the equality is a suitable combination of the Sherali-Adams and RLT constraints (see Paragraph 5.2.2.2) and is therefore necessarily satisfied. On the other hand, the square of inequality constraints (except bound constraints), may not be included in SDP-10, which explains the tiny difference between SDP-10 and SDP-12 gaps. The gaps of the three selected relaxations are illustrated on Figure 5.5.

Class	SDP-1	SDP-2	SDP-3	SDP-4	SDP-5	SDP-6	SDP-7	SDP-8	SDP-10	SDP-11	SDP-12
1-1	9.7%	9.7%	9.4%	6.9%	9.7%	9.7%	6.5%	6.1%	0.7%	0.6%	0.6%
1-2	6.0%	6.0%	6.0%	0.2%	6.0%	6.0%	0.0%	0.0%	0.0%	0.0%	0.0%
1-3	9.4%	6.9%	9.4%	6.9%	9.1%	8.9%	6.9%	6.9%	1.9%	1.9%	1.9%
2-1	25.3%	25.3%	25.1%	23.7%	25.3%	25.3%	23.0%	22.4%	16.5%	15.7%	16.4%
2-2	22.0%	22.0%	19.3%	13.2%	22.0%	22.0%	10.0%	10.0%	0.0%	0.0%	0.0%
2-3	25.1%	23.7%	25.1%	23.7%	25.0%	24.9%	23.7%	23.7%	18.3%	15.8%	18.3%
3-1	31.1%	31.1%	31.0%	30.4%	31.1%	31.1%	29.7%	29.2%	24.9%	0.0%	24.8%
3-2	28.6%	28.6%	24.5%	19.9%	28.6%	28.6%				0.0%	
3-3	31.0%	30.3%	31.0%	30.3%	31.0%	30.9%	30.3%	30.3%	26.9%	0.0%	26.9%
4-1	21.2%	21.2%	21.1%	20.3%	21.2%	21.2%					
4-2	17.7%	17.7%	16.0%	10.2%	17.7%	17.7%					
4-3	21.1%	20.3%	21.1%	20.3%	21.0%	20.9%					
5-1	22.9%	22.9%	22.8%	22.3%	22.9%						
5-2	19.9%	19.9%	17.5%	12.2%	19.9%						
5-3	22.8%	22.3%	22.8%	22.3%	22.8%						
6-1	31.0%	31.0%	31.0%	30.4%							
6-2	29.1%	29.4%	27.1%	20.2%							
6-3	31.6%	31.0%	31.6%	30.4%							

Table 5.13: Gap of the SDP relaxations

Class	LP-1	LP-2	LP-3	LP-4	LP-5	LP-6	LP-7	LP-8	LP-10	LP-11	LP-12
1-1	9.7%	9.7%	9.0%	9.0%	9.7%	9.7%	0.7%	0.6%	0.7%	9.0%	0.6%
1-2	6.0%	6.0%	6.0%	6.0%	6.0%	6.0%	0.0%	0.0%	0.0%	6.0%	0.0%
1-3	8.9%	7.7%	8.9%	7.7%	8.9%	8.9%	1.9%	1.9%	1.9%	7.7%	1.9%
2-1	25.3%	25.3%	24.7%	24.7%	25.3%	25.3%	17.4%	17.3%	17.4%	24.7%	17.3%
2-2	22.0%	22.0%	22.0%	22.0%	22.0%	22.0%	9.9%	9.9%	9.9%	22.0%	9.9%
2-3	24.9%	24.1%	24.9%	24.1%	24.9%	24.9%	19.4%	19.4%	19.4%	24.1%	19.4%
3-1	31.1%	31.1%	30.7%	30.7%	31.1%	31.1%	25.6%	25.5%	25.6%	30.7%	25.5%
3-2	28.6%	28.6%	28.6%	28.6%	28.6%	28.6%				28.6%	
3-3	30.9%	30.6%	30.9%	30.6%	30.9%	30.9%	27.8%	27.8%	27.8%	30.6%	27.8%
4-1	21.2%	21.2%	20.7%	20.7%	21.2%	21.2%					
4-2	17.7%	17.7%	17.7%	17.7%	17.7%	17.7%					
4-3	20.9%	20.6%	20.9%	20.6%	20.9%	20.9%					
5-1	22.9%	22.9%	22.5%	22.5%	22.9%						
5-2	19.9%	19.9%	19.9%	19.9%	19.9%						
5-3	22.8%	22.6%	22.8%	22.6%	22.8%						
6-1	31.0%	31.0%	30.7%	30.7%							
6-2	28.8%	28.8%	28.8%	28.8%							
6-3	30.9%	30.8%	30.9%	30.8%							

Table 5.14: Gap of the LP relaxations

Regarding running time, SDP-4 is significantly faster than the two others : on average on the 800 common instances, it runs in 0.1s, compared with 354.1 s for SDP-7 and 1018.4 s for SDP-10. Furthermore, SDP-7 and SDP-10 can be used only on very small instances, as show the systematic failure on the instances of class 3-2, for which the original problem involves 52 variables and 478 constraints.

5.3.2.3 Comparison of the three formulations of the maximal lapping constraint

As illustrated on the Figures 5.6 and 5.7, the SDP and LP gaps are * smaller with the pairwise exclusion formulation. This can be explained by the fact that the associated problem involves a larger number of linear constraints and therefore offers a better potential for reinforcement. On the other hand, the quadratic formulation has the worst gaps, for reasons that are similar.

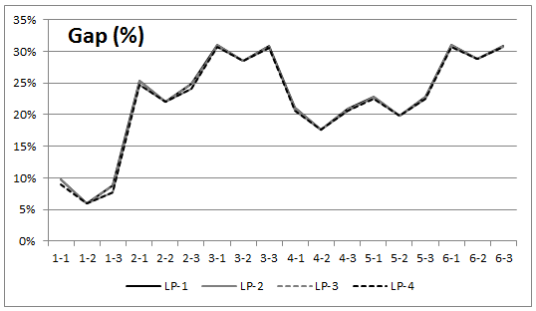

Figure 5.4: Comparison of the gaps of the relaxations LP-1, LP-2, LP-3 and LP-4

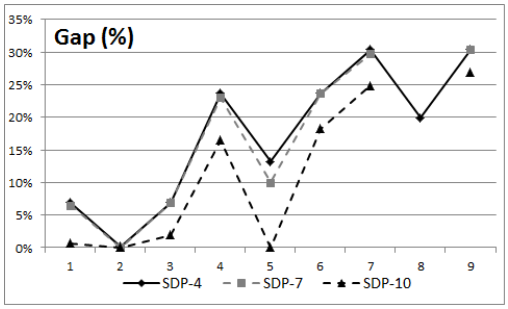

Figure 5.5: Comparison of the gaps of the relaxations SDP-4, SDP-7 and SDP-10

5.3.2.4 Comparison of the SDP relaxations with the LP relaxations

Moving on to SDP versus LP comparison, we show the gaps of the LP and SDP retained relaxations on Figure 5.8. Clearly, SDP is worthwhile only with the model $4-2$. The question that remains is whether this holds because the model $4-2$ involves a larger number of linear constraints, or because of the nature of these constraints. It is remarkable that on instances of class 2-2, the average SDP-10 gap is zero, which implies that SDP-10 closes the gap on all the instances of this class. On the same class, the average LP-10 relaxation has a gap of 9.85%. This leads to the comparison of the three models. As depicted on Figure 5.6, from a SDP relaxation point of view, it is equivalent to consider the "big M" formulation or the quadratic one.

We note that we do not raise the issue of computational time. Indeed, this comparison, which is clearly in favour of LP, is biased since the LP solver is CPLEX, a very powerful commercial solver, which benefited from years of research, whereas the SDP solver is CSDP, a free tool proposed in 1999 by a researcher.

In view of all these observations, we draw the following conclusions : for the three models, the best SDP relaxation is obtained by first of all, squaring the linear constraints (including the bound constraints), then adding the products of the linear constraints by the bound constraints.

Among the three models, the pairwise exclusion model offers the best improvement of SDP w.r.t. LP and the quadratic model the worst. Indeed, this model is less suitable for reinforcement since it involves a smallest number of linear constraints. Furthermore, as shown on Figure 5.9, the linear relaxation is generally as tight as the SDP one for this model. Actually, compared to the linear relaxation, the SDP relaxation is worthwhile only with the model $4-2$ that corresponds to the pairwise exclusion formulation of the maximal lapping constraint.

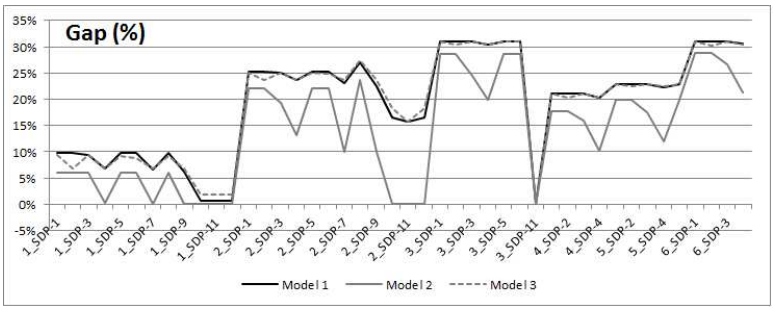

Figure 5.6: Comparison of the gaps of the SDP relaxations for the models $4-1$, $4-2$ and $4-3$

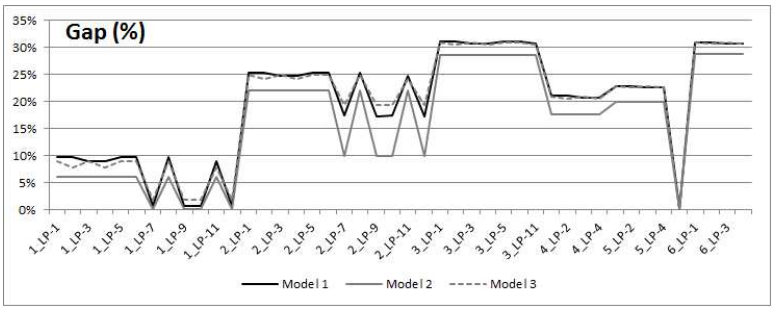

Figure 5.7: Comparison of the gaps of the LP relaxations for the models $4-1$, $4-2$ and $4-3$

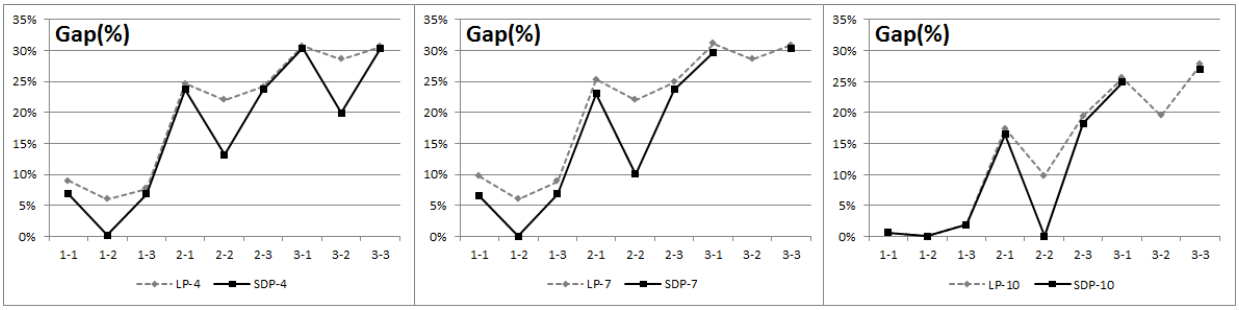

Figure 5.8: Comparison of the gaps of the SDP and LP relaxations for the reinforcement 4, 7 and 10

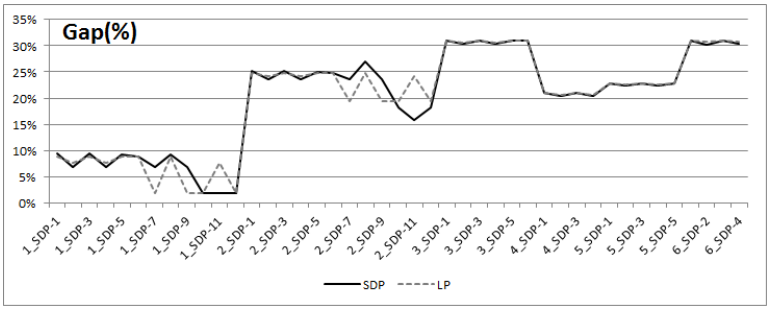

Figure 5.9: Comparison of the gaps of the SDP and LP relaxations for the model 3

5.3.3 Reaching optimality via Lasserre's hierarchy

A systematic and very powerful way of deriving SDP relaxations of combinatorial problems, and more generally, of polynomial problems, is proposed by the Lasserre's hierarchy of SDP relaxations, described

in Paragraph 3.4.3.

In this paragraph, we apply this technique to the three models of the nuclear outages scheduling problem described in Paragraph 4.3.3.4. Recall that the rank 1 of this hierarchy is equivalent to the standard SDP relaxation, which has already been covered in the section above. For memory storage reasons, the rank 2 of the hierarchy could not be tested on instances that are not of class 1, and the same applies for the rank > 2 on all the instances.

Then, on average on the 100 instances described in Paragraph 5.3.2.1, the rank 2 of the Lasserre's hierarchy gives the following gaps :

Class	Gap		Running time	
	Lasserre-rank 1	Lasserre-rank 2	Lasserre-rank 1	Lasserre-rank 2
1-1	9.74%	0.00%	0.01	2.42
1-2	6.03%	0.00%	0.01	1.06
1-3	9.39%	0.00%	0.02	4.72

Table 5.15: Gap and running time of the Lasserre rank-1 and rank-2 relaxations

The gaps are calculated with the formula $(p_r^* - p^*)/p^*$ where p_r^* is bound obtained with the relaxation and p^* the optimal value of the integer problem. Thus, having an average gap that vanishes indicates that the relaxation gives the integer optimal value on all the instances. Furthermore, the resolution of the rank-2 relaxation takes from 1s to 9s which remains very reasonable.

In conclusion, the Lasserre's hierarchy therefore keeps its promises regarding the tightness of the obtained bound. Unfortunately, in practice, it obviously suffers from the size of the generated SDP since it can not be applied to problems of size more than 2, i.e, with a number of variables of the order of thirty.

5.4 Conclusion

In this chapter, we investigated the potential of SDP for real-life combinatorial problems. First, we derived a SDP relaxation of the MIQP corresponding to the model 2 of the NOSP, described in paragraph 4.3.3.2. This problem is extremely hard to solve with CPLEX and SDP yiels a bound that outperforms the linear bounds, with an average gap that decreases from 1.80% to 1.56%. To reach this bound, it was necessary to reinforce the standard SDP relaxation by means of valid quadratic constraints obtained by applying the Sherali-Adams principle to the assignment constraint. The latter was chosen as the most efficient among all the quadratic constraint obtained by applying the Sherali-Adams principle to all the constraints of the problem.

In the second section, we proposed a more systematic method to generate and select the most appropriate valid quadratic constraints to reinforce the SDP relaxation. To this end, we consider all the pairwise products of the linear constraints of the problem, then we add the most violated of these constraints to the SDP relaxation. We experimented several variants of this basic idea. First, we try to select the most violated convex quadratic constraints, in order to be more efficient in the semidefinite relaxation. Indeed, a convex quadratic constraint is necessarily satisfied by the projected solution of the semidefinite relaxation. We obtain a reinforcement of the semidefinite relaxation but the computational cost is too high compared to the improvement of the bound. We also try to consider the triangular inequalities into our initial set of constraints but the impact on the bound was very low.

By applying this method to the NOSP, it comes that our SDP relaxation improves by 25.15% the gap of the linear relaxation. Thus the advantages of this method are twofold. First, it can be used to enhance the Branch & Bound resolution of the problem, especially on difficult instances where CPLEX fails to produce a feasible solution within two hours. Second, this method can be used to determine the most relevant cuts on working instances. Then, these cuts can be added directly to operational instances to provide directly an efficient semidefinite relaxation.

Finally, in the last section, we implemented numerous possible reinforcements of the SDP relaxations to the three equivalents models $4-1$, $4-2$ and $4-3$ (see Paragraph 4.3.3.4). The objective is to make a comparison between the possible reinforcements and between the models. We also aim at comparing the SDP relaxation with the linear relaxation reinforced in the same way than the SDP relaxation.

In conclusion, we experimentally observed that, for the three models, the best SDP relaxation is obtained by first of all, squaring the linear constraints (including the bound constraints), then adding the products of the linear constraints by the bound constraints. Among the three models, the pairwise exclusion model offers the best improvement of SDP w.r.t. LP and the quadratic model the worst. Indeed, this model is less suitable for reinforcement since it involves a smallest number of linear constraints. Furthermore, as shown on Figure 5.9, the linear relaxation is generally as tight as the SDP one for this model. Actually, compared to the linear relaxation, the SDP relaxation is worthwhile only with the model $4-2$ that corresponds to the pairwise exclusion formulation of the maximal lapping constraint.

Chapter 6

Applying SDP to optimization under uncertainty

Optimization under uncertainty faces several challenges. First it is necessary to formalize the available knowledge about random data. At best, their probability distribution is perfectly known, but generally, only a partial knowledge of this distribution is available. Besides, it is necessary to select the optimization criteria and to determine how to consider the constraints that involve random data.

Assuming that the probability distribution is perfectly known enables to optimize the expected value of the deterministic objective and to satisfy the constraints up to a given level of probability, called chance-constraints. This brings us in the framework of Stochastic Programming, that was introduced in the 1950s and is so far the most commonly used way of accounting for uncertainty. In particular, when it is difficult to describe precisely the probability distribution, the latter is approximated via a discrete probability distribution made of a finite set of scenarios, obtained for instance by a Monte-Carlo sampling or by historical observations.

However, this approach makes sense only when the optimization is repeated many times, since probability can be therefore interpreted as a frequency. This becomes much more problematical when the optimization occurs only a few times. In this case, it can be interesting to consider an alternative way of accounting for uncertainty, namely the robust optimization that adopts a worst-case perspective. To this end, only the knowledge of the support of the random data is required.

Recently, a compromise between these two approaches, called *distributionnally robust optimization* was proposed, that requires to know the support of the random data as well as some of their moments. We refer the reader to the Appendix 3.7 for the most famous results in optimization under uncertainty.

Historically, the first connection between SDP and uncertainty can be attributed to Ben-Tal and Nemirovksi [32] and El-Ghaoui [89] that show that the robust counterpart of certain optimization problems can be formulated as SDP. A new step was taken with the establishment of the connection between SDP and the Generalized Moment Problem [171], that allows to use SDP for distributionnally robust optimization. Finally, SDP was also exploited for Stochastic Programming, as see for instance [100, 81].

In this chapter, we investigate some of these approaches. The first section consists of the paper [116] where we apply the standard SDP relaxation to a stochastic version of the NOSP with a scenario-based representation of random data (see Paragraph 4.3.3.1). This computation is followed by a randomized rounding procedure in order to derive a feasible solution of the problem.

The second section, that constitutes the submitted paper [112], investigates the use of SDP for dealing with the distributionnally robust optimization of the problem of supply/demand presented at Paragraph 4.3.3.5.

Finally, the last section relates the work presented in [114]. We investigate SDP relaxations for mixed 0-1 Second-Order Cone Program, i.e. Second-Order Cone Program (SOCP) (see Paragraph 1.3.1) in which a specified subset of the variables are required to take on binary values. The reasons for our interest in these problems lie in the fact that SOCP are famous for providing formulations or conservative approximations of robust Linear Programs, see for instance [32, 185, 269]. By a natural extension, MISOCP can be used to reformulate or approximate robust MILP.

Thus, each section of this chapter corresponds to a published or submitted paper. Similarly to the choice made in the previous chapter, we chose not to provide the papers in their entirety in order to avoid the duplications. Instead, we refer the reader to the papers [116, 114, 112], to the Chapter 4 for energy management problems and models, to the Chapter 2 for the an overview of SDP and to Paragraph 3.3.2 for a detailed explanation of the standard SDP relaxation.

6.1 SDP for optimizing with a discrete representation of uncertainty

In this section, as presented in the paper [116], we apply the standard SDP relaxation to the model 1 of the nuclear outages scheduling problem (see Paragraph 4.3.3.1). This computation is followed by a randomized rounding procedure in order to derive a feasible solution of the problem.

We recall that the model 1 is a direct application of the description of the NOSP given at Section 4.3.1 with the quadratic objective function and the quadratic formulation of the maximal lapping constraint. Regarding uncertainty, we use a stochastic approach and we require that the constraints involving uncertain parameters be satisfied up to a given level of probability (chance-constraints).

We assume that the probability distributions of the uncertain parameters are discrete and concentrated on a finite number of scenarios $s = 1, ..., N_s$ obtained from historical observation. Then, as explained in Paragraph 4.3.3.1, the joint chance-constraints can be expressed in a deterministic fashion by introducing binary variables.

The objective is to minimize the expected value of the production cost, that can be easily computed as the sum of the production cost of each scenario, weighted by their probability. Finally, we obtain a MIQCQP, with a large number of binary variables and linear constraint, where the quadratic term involves only binary variables.

Then, we apply the standard SDP relaxation (see Paragraph 3.3.2) to this problem. Experiments have been carried out on 10 data sets, built from a real-life problem describing the 58 french nuclear power plants on a five years time horizon. For each of the data set, 10 scenarios are considered. This choice is made in order to keep a reasonable number of binary variables and to focus on the combinatorial aspect coming from the outages.

For each of the 10 considered data set, the horizon time contains 156 weeks, also called time steps and corresponds to 4 cycles and 3 outages for each plant. The other features are presented in Table 6.1. For each data set, denoted by their number in the first column, this table contains the number of nuclear power plants in the column 2 and the number of outages in the column 3. The column 4 indicates the number of possible beginning date for each outage i.e., the size of the search spaces $\mathcal{E}_{i,j}$. These sets are built as the symmetric space around an initial date of outage, which corresponds to a feasible solution. The last column of Table 6.1 indicates the number of binary variables of the instances, without considering the binary variables resulting from the linearization since they are not used in the semidefinite relaxation.

6.1.1 Semidefinite Relaxation Lower Bounds

Results are presented in Table 6.2. We compare our semidefinite relaxation to the Integer Linear program described in Paragraph and to its continuous relaxation, denoted herein *Exact resolution* and *Linear relaxation* respectively.

Data set	Number of nuclear power plants	Number of outages	Size of the search spaces	Number of binary variables
1	10	30	7	701
2	10	30	9	776
3	10	30	11	870
4	10	30	13	965
5	10	30	15	1 047
6	10	30	17	1 123
7	20	60	7	1 846
8	20	60	9	2 034
9	20	60	11	2 226
10	20	60	13	2 409

Table 6.1: Description of the data sets

All numerical test were performed with an Intel(R) Core(TM) i7 processor with a clock speed of 2.13 GHz. The semidefinite relaxation is solved using DSDP 5.8 solver [34]. The exact formulation and its associated linear relaxation have been solved with the CPLEX 12.1 solver.

Data	Exact resolution		Linear relaxation			SDP relaxation		
set	Value	Time	Value	Time	Gap	Value	Time	Gap
1	5 031	1 800	3 578	0,06	40,59%	4 988	298	0,85%
2	4 931	1 800	3 397	0,08	45,15%	4 875	448	1,15%
3	4 836	1 800	3 262	0,16	48,27%	4 775	633	1,27%
4	4 782	1 800	3 134	0,28	52,59%	4 691	957	1,94%
5	4 723	1 800	3 026	0,58	56,08%	4 613	1329	2,36%
6	4 670	1 800	2 913	0,80	60,34%	4 512	1682	3,52%
7	20 547	1 800	13 677	0,33	50,23%	20 051	4048	2,47%
8	20 443	1 800	13 159	0,71	55,36%	19 736	6216	3,58%
9	20 392	1 800	12 740	1,07	60,07%	19 506	9251	4,55%
10	20 417	1 800	12 384	1,71	64,87%	19 279	11520	5,9%

Table 6.2: Results of the relaxations

In Table 6.2, the data set is given in the first column, which refers to Table 6.1. The “Value" columns indicate the result of the resolutions and the columns “Time" indicate the computational time in second. In the case of the exact resolution, this time is limited to 1800s and is attained for all the data set. Consequently, the associated result is not the exact optimal value, but an upper bound.

Let $p*$ be the optimal value or an upper bound and p_r be the result of a relaxation, the gap g in columns 6 and 9 is computed as follows :

$$g = \frac{p* - p_r}{p_r} \tag{6.1}$$

These results show that the semidefinite relaxation is definitely more powerful that the continuous one. Indeed, the average gap is 53.35% for the continuous relaxation and 2.76% for the semidefinite relaxation. We can see that both gaps increase as the size of the data set goes up, but this rise is more important for the continuous relaxation gap. This suggests that our semidefinite relaxation could be even more useful on larger instance.

6.1.2 Deriving Feasible Solutions from the Relaxed Solutions

Results of the randomized rounding procedure, obtained from the continuous relaxation and from the semidefinite relaxation are presented by columns 3, 4, 5, 6 of Table 6.3:

Data set	Exact resolution	LP relaxation		SDP relaxation	
		Value	Gap	Value	Gap
1	5 031	5 475	8,11%	5 122	1,77%
2	4 931	5 366	8,1%	5 021	1,78%
3	4 836	5 428	10,91%	4 962	2,54%
4	4 782	5 336	10,37%	5 041	5,13%
5	4 723	5 661	16,58%	4 997	5,5%
6	4 670	5 669	17,61%	4 885	4,38%
7	20 547	†	†	†	†
8	20 443	†	†	20 335	-0,53%
9	20 392	21 574	5,48%	20 459	0,33%
10	20 417	21 686	5,85%	20 205	-1,05%

Table 6.3: Using a randomized rounding procedure to derive a feasible solution

† indicates that no feasible solution has been found after 1000 iterations. This occurs when none of the outages variables with a positive value yield a feasible solution. We observe that the SDP relaxations induces a better robustness of the randomized rounding procedure since the infeasible case occurs less frequently.

For each of the data set denoted by their index in the first column, the column 2 gives the value of the reference solution, as in the second column of Table 6.2. Then, for each of the considered relaxation (linear and semidefinite), we give the values of the obtained feasible solutions in columns 3 and 5. The associated gap g given in columns 4 and 6 is computed as follows:

$$g = \frac{p_a - p*}{p_a} \tag{6.2}$$

where $p*$ is the reference solution and p_a is the result of the randomized rounding.

We can see that our results become closer from the upper bound of the solution found by CPLEX within 1800s when the size of the instances goes up, since this upper bound differs more and more from the optimal value. We even improve this solution on data sets 8 and 10. More generally, the solutions obtained from the semidefinite relaxation are clearly better than those obtained from the LP relaxation, with a minimal difference of 5.15% of gap between them. On average, the gap is 10.38% with the continuous relaxation, whilst the SDP relaxation leads to 2.20% gap only.

6.1.3 Conclusion

In electrical industry, optimizing the nuclear outages scheduling is a key factor for safety and economic efficiency. In order to take into account uncertainties on the production and the demand and focusing on the main constraints, we propose a probabilistic formulation of this challenging large-size stochastic mixed nonlinear problem. By using individual chance constraints, we obtain a large mixed-integer Quadratically Constrained Quadratic Program.

Transforming it into an Integer Linear Program by using Fortet linearization leads to a large-scale problem very difficult to solve with commercial solvers. We propose an alternative approach involving

semidefinite relaxation. This relaxation gives lower bounds that are very close to the optimal value, with a gap equal to 2.76% on average, whereas the continuous relaxation has on average a gap of 53.35%. Then these bounds are used to build feasible solutions by the mean of a randomized rounding procedure. Our approach performs well since the optimality gap is 2.20% for the SDP relaxation versus 10.38% for the continuous relaxation.

These promising results suggest future work to refine the semidefinite bounds. In particular, we aim at investigating the use of valid inequalities. Another perspective for this work is to embed the bound computation within a Branch & Bound procedure.

6.2 Handling a chance-constraint with a distributionnally robust approach

In this section, we investigate the use of SDP for dealing with the distributionnally robust optimization of the problem of supply/demand presented at Paragraph 4.3.3.5. This work is reported in the submitted paper [112]. The reasons for our interest in this approach is twofold. Firstly, the distributionnally robust optimization does not require the knowledge of the probability distribution of the uncertain parameters, as this is the case for stochastic optimization. However, it exploits some characteristics of the probability distribution, namely its moments of order up to k and its support, which makes this approach less conservative than the robust approach, which exploits only the support.

Secondly, SDP provides a very elegant way of dealing with these problems, as presented at Paragraph 3.5.2. Generally, the obtained SDP is a conservative approximation of the problem and in some particular cases, it may even be an exact reformulation of the problem.

The supply/demand equilibrium problem under uncertainty that we consider can be formulated as the following jointly chance-constrained linear program :

$$\min_{x \in F} \{c^T x : \mathrm{P}[g(x,\xi) \leq 0] \geq 1 - \varepsilon\} \tag{6.3}$$

where $x \in \mathbb{R}^n$ is the command vector, $c^T x$ its deterministic linear cost and $F \subset \mathbb{R}^n$ a deterministic polyhedron. Uncertain parameters are represented by the m-dimensional random vector ξ and g is a measurable function from $\mathbb{R}^n \times \mathbb{R}^m$ to $\mathbb{R}^T$, whose components g_t are affine w.r.t x and ξ. For $T = 1$, the constraints reduces to a so-called *individual* chance-constraint. Otherwise, this is a joint chance-constraint which requires that the T inequalities be jointly satisfied with a probability at least $1 - \varepsilon$, with ε a given probability threshold. In this case, we call *sub-constraints* the inequalities $g_t(x,\xi) \leq 0,\ t = 1, ..., T$.

Even when the probability distribution of the random vector is known, solving a chance-constrained problem is a highly challenging task. First, checking the feasibility of a given candidate solution is complicated since it requires the computation of a T-dimensional integral. For instance, it was shown in [201] that it is NP-hard to compute the probability of a weighted sum of uniformly distributed variables being nonpositive. Second, the feasible region defined by a chance-constraint is generally non convex, even disconnected.

In our case, only a partial knowledge about the probability distribution is available. More precisely, we assume that the support of ξ as well as some of its moments are known. Then, we optimize over all the probability distributions that match these characteristics. By denoting $\mathcal{P}(\mathcal{S})$ this set, the problem can be formulated as :

$$\begin{cases} \min & c^T x \\ \text{s.t.} & \mathrm{P}[g(x,\xi) \leq 0] \geq 1 - \varepsilon, \quad \forall \mathrm{P} \in \mathcal{P}(\mathcal{S}) \\ & x \in F \end{cases} \tag{6.4}$$

In other words, we obtain a guarantee on the feasibility of the solution by replacing the unknown quantity $\mathrm{P}[g(x,\xi) \leq 0]$ by its lower bound $\min_{\mathrm{P}\in\mathcal{P}(\mathcal{S})} \mathrm{P}[g(x,\xi) \leq 0]$, computed by making the best possible use of the available information.

A recent approach to this problem, proposed in [270], approximates the obtained constraint under the form of a Linear Matrix Inequality (LMI). Thus, the obtained problem takes the form of a Semidefinite Program (SDP), a convex optimization problem for which efficient solvers are available.

In this paper, we aim at investigating more in depth in this direction for the particular *quadratic case*, defined as follows :

- g is affine in x and ξ, but the same rationale would apply to g quadratic in ξ ;
- $\mathcal{S}$ is defined through a set of quadratic constraints;
- the first and second-order moments of ξ are given.

Our main contribution is to unify this approach with the works of Calafiore and El-Ghaoui [68], Lasserre [171], Bertsimas et al. [38, 40, 41] and Comanor et al. [257], for this particular case. We provide a simple way to recover all these results by applying the well-known S-Lemma, which is possible due to our restriction to the quadratic case.

Furthermore, we apply this so-called *distributionnally robust approach* to the supply-demand equilibrium problem and we aim at comparing this method to existing approaches in order to measure its efficiency. To this end, we set aside the covariance information and consider only the support and expected values in a robust approach, as proposed in [68, 269]. This approximation relies on the application of the Boole's inequality, to convert the joint constraint into individual ones, combined to the Hoeffding's inequality, in order to get a tractable conservative approximation of the constraints. The obtained problem takes the form of a Second-Order Cone Program (SOCP), a special case of conic programming which has received significant attention in the literature [8], and for which numerous efficient solvers are available [248].

We also aim at measuring the potential loss w.r.t the case where the uncertainty would be perfectly known. As an illustration, we consider the case where $g(x,\xi)$ follows a Gaussian distribution. To solve the problem, we resort to the approximation proposed in [73], which also gives rise to a SOCP.

In short, we compare three ways of dealing with uncertainty that goes hand in hand with a level of knowledge :

- a *robust approach*, where nothing is known about the probability distribution apart its support and expected value;
- a *distributionnally robust approach*, where the support and the two first moments of the probability distribution are known;
- a *stochastic approach* where the probability distribution is perfectly known.

Naturally, the more precise is our knowledge about the uncertainty, the smaller the set $\mathcal{P}(\mathcal{S})$ is and the more accurate is the solution. However the distributionnally robust approach appears as a good compromise between the robust approach, which might be over conservative and the stochastic approach which require a high level of knowledge, generally not available.

This paper is organized as follows. In section 6.2.1, we review briefly the approaches proposed in the literature to tackle such joint chance-constraints. We also provide background on the distributionnally robust framework, its connexion with Semidefinite Programming and with the Generalized Problem of Moment. We present our main contribution in section 6.2.2, by detailing several particular cases of the considered problem and for each of them, making the link with other works of the literature. Then, in section 6.2.3, we report and analyse numerical results arising from the comparison with the robust and the stochastic approaches. Finally, we conclude and discuss future work in section 6.2.4.

Notations The problem (6.3) uses the following notations :

- $x \in \mathbb{R}^n$ is the command variable ;

- $c \in \mathbb{R}^n$ is the cost vector ;
- $F \subset \mathbb{R}^n$ is a polyhedron defined by a set of linear inequalities ;
- $\xi \in \mathbb{R}^m$ is a random vector, of mean $\mu \in \mathbb{R}^m$, covariance $\Sigma \in \mathbb{S}^m$ and support $\mathcal{S} = \{\xi \in \mathbb{R}^m : a_i \leq \xi_i \leq b_i,\ i = 1, ..., m\}$;
- $\mathcal{P}(\mathcal{S}) = \{\mathrm{P} \in \mathcal{M}(\mathcal{S}) : \Omega_{\mathrm{P}}(\xi) = \Omega\}$ with $\Omega = \begin{pmatrix} 1 & \mu^T \\ \mu & \mu\mu^T + \Sigma \end{pmatrix}$;
- $\mathcal{M}(\mathcal{S})$ denotes the set of the probability distributions supported on $\mathcal{S}$;
- $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^T$ a measurable function such that $g_t(x, \xi) = \tilde{x} A^t \tilde{\xi},\ t = 1, ..., T$;
- $A^t \in \mathbb{R}^{n+1, m+1}$ whose indices starts at zero ;
- ${A^t}_{i,*}$ and ${A^t}_{*,j}$ denotes the i-th row and j-th column of A^t respectively ;
- $\exists x \in F : g(x, \mu) \leq 0$;
- $1 - \varepsilon$ is the prescribed probability level, with $0 < \varepsilon < 1$;
- the variable ξ involved in a same subconstraint $g_t(x, \xi) \leq 0$ are independent one from another.

Clearly, $\Omega \succcurlyeq 0$. Moreover, unless one components of ξ is an exact affine combination of the others, $\Sigma \succ 0$ holds and therefore we assume in the sequel that $\Sigma \succ 0$, which is equivalent to $\Omega \succ 0$ by applying Schur's complement.

Finally, $\mathcal{S}$ is defined above as a box but, when possible, we extend this notion to define $\mathcal{S}$ through a set of quadratic constraints : $\mathcal{S} = \{\xi \in \mathbb{R}^m : \tilde{\xi}^T W^s \tilde{\xi} \geq 0,\ s = 1, ..., S\}$. Two possible representation of a box in this form are discussed at the end of the Paragraph 6.2.2.3.

For any $\mathrm{P} \in \mathcal{M}(\mathcal{S})$, $\mathrm{E}_{\mathrm{P}}(\tilde{\xi}^T W^s \tilde{\xi}) \geq 0$, which is equivalent to $\mathrm{E}_{\mathrm{P}}(W^s \bullet (\tilde{\xi}\tilde{\xi}^T)) \geq 0$ and therefore $W^s \bullet \Omega \geq 0$. We assume in the sequel that this condition holds for $s = 1, ..., S$.

Throughout this section, we use the following notations. If v is a n-dimensional vector, $M(v)$ denotes the matrix in $\mathbb{S}^n$ such that $M_{1,1} = v_1$, $M_{1,i} = M_{i,1} = 1/2 v_i,\ i = 2, ..., n$ and $M_{i,j} = 0$ otherwise. In particular, $e_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \in \mathbb{R}^{n+1}$ and therefore $M(e_0) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{S}^{n+1}$.

Furthermore, if ξ is a random vector of probability distribution P, $\Omega_{\mathrm{P}}(\xi) = \mathrm{E}(\tilde{\xi}\tilde{\xi}^T)$ denotes its second-order moment matrix. This matrix is indexed by $i, j = 0, ..., m$ such that $\Omega_{\mathrm{P}}(\xi)_{0,0} = 1$ and for $i, j = 1, ..., m$, $\Omega_{\mathrm{P}}(\xi)_{0,j} = \mathrm{E}_{\mathrm{P}}(\xi_j)$ and $\Omega_{\mathrm{P}}(\xi)_{i,j} = \mathrm{E}_{\mathrm{P}}(\xi_i \xi_j)$.

6.2.1 Literature review

6.2.1.1 Handling a joint chance-constraint

In this section, we give a brief overview for dealing with a general joint chance-constraint $\mathrm{P}[g(x, \xi) \leq 0] \geq 1 - \varepsilon$, according to the nature of the function $g : \mathbb{R}^n \times \mathcal{S} \to \mathbb{R}^T$ and to the knowledge available about the random vector ξ.

Incorporating such constraints into an optimization problem leads to a so-called *chance-constrained programs*. They were introduced by Charnes and Cooper [71] in the late fifties and has since been studied extensively, see for instance [201, 216, 227]. These constraints are a very valuable modelling tool, especially when the optimization repeats many time, since they can be viewed as a way to ensure a certain stability of performance.

A classical way to handle chance-constraints is based on the Monte-Carlo method and consists of approximating the probability distribution by a discrete distribution generated by random sampling. A major advantage of this method is that it can be done for an arbitrary probability distribution of ξ and function g, as long as g is affine in x. Unfortunately, it was shown in [66] that a minimum of $O(n/\varepsilon)$ samples were required to guarantee the feasibility of the approximated solution, which is computationally prohibitive for small value of ε.

In the particular case of an individual chance-constraint ($T = 1$) where $\varepsilon \leq 0.5$, $g(x,\xi) = \xi^T x - v$ and ξ has a multivariate normal distribution, the chance-constraint admits a tractable convex formulation in the form of a deterministic second-order cone program [256]. This reformulation forms the basis of the stochastic approach described at Paragraph 6.2.3.6.

However, in general, individual chance-constraints are very hard to enforce numerically. The question that naturally arises is how the properties of $f : x \mapsto \mathrm{P}[g(x,\xi) \leq 0]$ can be derived from those of g. A common difficulty, for instance, is that convexity of g with respect to x may not lead to convexity of f with respect to x.

Hence, a natural way to overcome this difficulty is to look for a convex conservative approximation of the constraint. It was shown by Nemirovski and Shapiro [201] that, for $T = 1$, the least conservative convex approximation is based on the CVaR, a risk measure defined by Rockafellar and Uryasev in [227], that, for any $0 < \eta < 1$, associates to a random variable X the mean of its γ-quantile on $\gamma \in [1-\eta, 1]$. Then, $\mathrm{P}[X \leq \mathrm{CVaR}_\eta(X)] \geq 1-\eta$. By applying this to the random variable $g(x,\xi)$, we get the following implication : $\mathrm{CVaR}_\varepsilon(g(x,\xi)) \leq 0 \Rightarrow \mathrm{P}[g(x,\xi) \leq 0] \geq 1-\varepsilon$ and therefore, $\mathrm{CVaR}_\varepsilon(g(x,\xi)) \leq 0$, which is convex, is a conservative approximation of our chance-constraint.

Regarding joint chance-constraints, the Boole's inequality proved to be very useful in order to approximate a T-joint chance-constraint by T individual chance-constraint. Indeed, it states that :

$$\mathrm{P}[g_t(x,\xi) \leq 0,\ t = 1,...,T] \geq \sum_{t=1}^{T} \mathrm{P}[g_t(x,\xi) \leq 0] + 1 - T$$

Consequently, it suffices that the sum of the individual probability be greater than $T-\varepsilon$ to ensure the satisfaction of the constraint. By distributing equally this probability among the T constraints, we get the following conservative approximation :

$$\mathrm{P}[g_t(x,\xi) \leq 0] \geq 1 - \varepsilon/T,\ t = 1,...,T \quad \Rightarrow \quad \mathrm{P}[g_t(x,\xi) \leq 0,\ t = 1,...,T] \geq 1 - \varepsilon$$

However, this conservative approximation suffers from its potential lack of tightness, especially when the constraints involves a large number of correlated constraints as pointed out in [72], that also proposed a new scheme to address this problem. It is based on the following statement, that holds for any strictly positive $\alpha \in \mathbb{R}^T$ and that allows to convert a joint chance-constraint into an individual one :

$$g_t(x,\xi) \leq 0,\ t = 1,...,T \quad \Leftrightarrow \quad g_\alpha(x,\xi) = \max_{t=1,...,T} \alpha_t g_t(x,\xi) \leq 0 \tag{6.5}$$

Hence, in [72], the authors propose to replace the joint chance-constraint by the following individual one : $\mathrm{P}[g_\alpha(x,\xi) \leq 0] \geq 1-\varepsilon$ and to apply the CVaR approximation to this constraint. Unfortunately, there is little guidance on how to choose the scaling parameters α, although the impact on the tightness of the approximation is significant. Optimizing the choice of α leads to a non convex problem and is therefore intractable.

6.2.1.2 SDP and the Generalized Problem of Moments

We give some key definitions and results to recall the connexion existing between Semidefinite Programming (SDP) and the Generalized Problem of Moments (GPM). We refer the reader to the section 3.4 for more details.

We recall that the Generalized Problem of Moments (GPM) is an optimization problem defined as follows :

$$(GPM_P) \begin{cases} p^* = \min & \mathrm{E_P}[h(\xi)] \\ \text{s.t.} & \mathrm{E_P}[f_j(\xi)] = b_j, j = 0,...,l \\ & \mathrm{P} \in \mathcal{M}(\mathcal{S}) \end{cases} \quad (GPM_D) \begin{cases} d^* = \max & b^T z \\ \text{s.t.} & \sum_{j=0}^{l} z_j f_j(\xi) \leq h(\xi), \forall \xi \in \mathcal{S} \end{cases} \tag{6.6}$$

where $\mathcal{M}(\mathcal{S})$ is the set of probability measure supported on $\mathcal{S} \subset \mathbb{R}^m$ and $f_j : \mathcal{S} \to \mathbb{R},\ i = 0, ..., l$ are measurable functions. We assume that the constraint $\int_{\mathcal{S}} \mathrm{P}(\xi) d\xi = 1$, due to P being a probability measure, is explicitly present with $f_0(\xi) = 1$ and $b_0 = 1$.

The dual relationship between (GPM_P) and (GPM_D) can be understood by considering the primal as a linear program with an infinite number of non-negative variable, i.e., $\mathrm{P}(\xi)$ for each value $\xi \in \mathcal{S}$. The dual follows and has an infinite number of linear constraint, that can be interpreted together as the non-negativity on $\mathcal{S}$ of the function $f_z(\xi) = h(\xi) - \sum_{j=0}^{l} z_j f_j(\xi)$. There exists Slater's type sufficient conditions for the strong duality of this problem (see Theorem 3.4.4).

Lasserre's hierarchy of semidefinite relaxation

The GPM subsumes a variety of optimization problem, including polynomial optimization, and is therefore NP-hard. However, in the polynomial case, i.e., when $f_j,\ i = 1, ..., l$ are polynomials and $\mathcal{S}$ is a semi-algebraic set : $\mathcal{S} = \{x \in \mathbb{R}^n : g_s(x) \geq 0,\ s = 1, ..., S\}$, with $g_s,\ s = 1, ..., S$ also polynomials of degree d_s, Lasserre designed a hierarchy of SDP whose optimal value converges to the optimal value of the GPM.

The key ingredient for interpreting these SDP is the notion of *moment vector* associated to a random vector ξ of probability distribution P, which denotes the vector y indexed by $\kappa \in \mathbb{N}^n_d$ such that $y_\kappa = \mathrm{E}_{\mathrm{P}}[\xi^\kappa]$. Conversely, $y \in \mathbb{R}^{\mathbb{N}^n_d}$ is said to be a moment vector if there exists a corresponding probability distribution P.

Defining the notion of moment vector enables to reformulate the GPM. Indeed, for any polynomial f_j of degree d with coefficients $\mathrm{f}_{\mathrm{j}_\kappa}$, $\mathrm{E}_{\mathrm{P}}[f_j(\xi)] = \sum_{\kappa \in \mathbb{N}^n_d} \mathrm{f}_{\mathrm{j}_\kappa} y_\kappa$ and the constraint is therefore linear in y. All the difficulty is now pushed back in the constraint that y be effectively a moment vector. This is where the SDP comes in, as pointed out in [77, 145]. Indeed, some necessary conditions for y to be a moment vector on $\mathcal{S}$ takes the form of a LMI.

More precisely, if g is a polynomial non-negative over $\mathcal{S}$ then the *localizing matrix* associated with g and y has to be positive semidefinite, and this matrix, indexed by the elements of $\mathbb{N}^n_d$ is defined as follows, for any rank $r \geq v = \left\lfloor \frac{d+1}{2} \right\rfloor$:

$$M_{r-v}(g, y)_{\kappa_1, \kappa_2} = \sum_{\kappa \in \mathbb{N}^n_{2v}} \mathsf{g}_\kappa y_{\kappa_1 + \kappa_2 + \kappa},\ \forall \kappa_1, \kappa_2 \in \mathbb{N}^n_{r-v}$$

This matrix can be formulated as a linear combination of the suitably defined matrices $B^\kappa(\mathsf{g})$ and of y :

$$M_{r-v}(g, y) = \sum_{\kappa \in \mathbb{N}^n_{2r}} B^\kappa(\mathsf{g})\ y_\kappa \quad \text{with} \quad B^\kappa(\mathsf{g})_{\kappa_1, \kappa_2} = \begin{cases} \mathsf{g}_{\kappa - \kappa_1 - \kappa_2} \text{ if } \kappa \geq \kappa_1 + \kappa_2 \\ 0 \text{ otherwise} \end{cases},\ \forall \kappa_1, \kappa_2 \in \mathbb{N}^n_{r-v}$$

In particular, for $g_0(x) = 1$, $M_r(g_0, y)$ corresponds to the so-called *moment matrix* of y and its positive semidefiniteness holds for any moment vector y.

The r-th rank of the Lasserre hierarchy is built by deriving these constraints for all the polynomials g_s defining $\mathcal{S}$ and for the polynomial $g_0(x) = 1$. The relaxation is defined for all integer $r \geq v_s = \left\lfloor \frac{d_s+1}{2} \right\rfloor,\ s = 0, ..., S$ and it was proved in [171] that $\lim_{r \to +\infty} p^*_r = p^*$:

$$(LH^r_P) \begin{cases} p^*_r = & \min & \mathsf{h}^T y \\ & \text{s.t.} & \mathsf{f_j}^T y = b_j,\ j = 0, ..., l \\ & & M_{r-v_s}(g_s, y) \succcurlyeq 0,\ s = 0, ..., S \\ & & y \in \mathbb{R}^{\mathbb{N}^n_{2r}} \end{cases} \tag{6.7}$$

The dual of this SDP writes as follows :

$$(LH_D^r)\begin{cases} d_r^* = & \max & b^T z \\ & \text{s.t.} & \sum_{s=0}^{S} B^\kappa(g_s) \bullet N_s = \mathsf{h}_\kappa - \sum_{j=0}^{l} z_j \mathsf{f}_{\mathsf{j}_\kappa},\ \forall \kappa \in \mathbb{N}^n_{2(r-v_s)} \\ & & N_s \succcurlyeq 0,\ s = 0, ..., S \\ & & z \in \mathbb{R}^{l+1},\ N_s \in \mathbb{S}^{\mathbb{N}^n_{r-v_s}},\ s = 0, ..., S \end{cases} \tag{6.8}$$

It is remarkable that (LH_D^r) can directly be derived as a conservative approximation of (GPM_D). Indeed, the constraints of this problem are a direct application of the Putinar's theorem (see for instance [171]) that provides a sufficient condition for the non-negativity of the polynomial $f_z(\xi) = h(\xi) - \sum_{j=0}^{l} z_j f_j(\xi)$ on the semi-algebraic set $\mathcal{S}$. This insight allows to deploy an extension regarding the nature of the objective function h. More specifically, when h is piecewise polynomial, i.e., $h(\xi) = h_k(\xi)$ if $\xi \in \mathcal{S}_k$ where $\{\mathcal{S}_k\}_{k=1,...,K}$ is a partition of semi-algebraic sets of $\mathbb{R}^n$, then the dual moment problem becomes :

$$(GPM_D)\begin{cases} \max & b^T z \\ \text{s.t.} & \sum_{j=0}^{l} z_j f_j(\xi) \leq h_k(\xi),\ \forall \xi \in \mathcal{S} \cap \mathcal{S}_k,\ k = 1, ..., K \end{cases}$$

Thus, the constraint still concerns the non-negativity of one or more polynomials over one or more semi-algebraic sets and therefore the corresponding problem can still be approximated as a SDP. As an illustration, let us consider the case where $h(\xi) = h_1(\xi)$ if $g_{S+1}(\xi) \geq 0$ and $h(\xi) = h_2(\xi)$ otherwise, with $h_1(\xi) \leq h_2(\xi)$ everywhere. Then, the obtained SDP is as follows :

$$(LH_D^r)\begin{cases} d_r^* = & \max & b^T z \\ & \text{s.t.} & \sum_{s=0}^{S+1} B^\kappa(g_s) \bullet N_s^1 = \mathsf{h}_{\mathsf{1}_\kappa} - \sum_{j=0}^{l} z_j \mathsf{f}_{\mathsf{j}_\kappa},\ \forall \kappa \in \mathbb{N}^n_{2(r-v_s)} \\ & & \sum_{s=0}^{S} B^\kappa(g_s) \bullet N_s^2 = \mathsf{h}_{\mathsf{2}_\kappa} - \sum_{j=0}^{l} z_j \mathsf{f}_{\mathsf{j}_\kappa},\ \forall \kappa \in \mathbb{N}^n_{2(r-v_s)} \\ & & N_s^1 \succcurlyeq 0,\ s = 0, ..., S+1 \\ & & N_s^2 \succcurlyeq 0,\ s = 0, ..., S \\ & & z \in \mathbb{R}^{l+1},\ N_s \in \mathbb{S}^{\mathbb{N}^n_{r-v_s}},\ s = 0, ..., S \end{cases} \tag{6.9}$$

Finally, in the case where $h, f_j,\ j = 1, ..., l$ and $g_s, s = 1, ..., S$ are degree-2 polynomials, i.e., $h(\xi) = \tilde{\xi}^T P_0 \tilde{\xi}$, $f_j(\xi) = \tilde{\xi}^T P_j \tilde{\xi}$ and $g_s(\xi) = \tilde{\xi}^T Q_s \tilde{\xi}$, the semidefinite rank 1 relaxation of Lasserre is exact for $S \leq 1$. This comes from the fact that in this case, the non-negativity constraint of f_z on $\mathcal{S}$ is a direct application of the well-known S-lemma [212] that we recall hereafter :

Lemma 6.2.1 *S-Lemma*
Let $P, Q_s,\ s = 1, ..., S$ be $m+1$-dimensional symmetric matrices. Then

$$\left.\begin{array}{l} P - \sum_{s=1}^{S} \lambda_s Q_s \succcurlyeq 0 \\ \lambda \geq 0 \end{array}\right\} \Rightarrow \tilde{\xi}^T P \tilde{\xi} \geq 0,\ \forall \xi \in \{\xi \in \mathbb{R}^m : \tilde{\xi}\ Q_s\ \tilde{\xi} \geq 0,\ s = 1, ..., S\}$$

The converse holds for $S = 0$ and for $S = 1$ whenever Q_1 is not negative semidefinite. In these cases, the S-Lemma is said to be lossless.

Then it suffices to apply this to the non-negativity of $f_z(\xi) = \tilde{\xi} P \tilde{\xi}$ with $P = P_0 - \sum_{j=0}^{l} z_j P_j$ to recover (LH_D^1), by taking $N_0 = P - \sum_{s=1}^{S} \lambda_s Q_s$ and $N_s = \lambda_s,\ s = 1, ..., S$.

6.2.1.3 Distributionally robust optimization

Distributionally robust optimization deals with optimization facing incompletely specified uncertainty, meaning that only a partial knowledge of the probability distribution of the uncertain parameters is available.

Such a framework is widespread in the real-world, since evaluating precisely a probability distribution is generally a challenging task. Then the aim of distributionnally robust optimization is to overcome this ambiguity, that prevents from applying the classical stochastic programming methods, by considering the worst-case on all the matching probability distribution.

This relatively recent way to deal with uncertainty appears as a compromise between stochastic programming, where the probability distribution is supposed to be perfectly known, and the robust optimization, where only information on the support is required. Thus, in the distributionally robust approach, the probability distribution is partially specified though certain characteristics, such as support and order up to k moments. We could think to exploit other properties of the probability distribution, such as symmetry, independence or being radial, but it is beyong the scope of this paper.

The available information on the probability distribution is used to define the class $\mathcal{P}$ of matching distributions and we perform a worst-case optimization by requiring that the chance-constraint be satisfied for all the distributions of $\mathcal{P}$. Hence, the distributionally robust variant of the constraint (6.3) is :

$$\mathrm{P}[g(x,\xi) \leq 0] \geq 1-\varepsilon,\ \forall \mathrm{P} \in \mathcal{P} \qquad \text{or equivalently} \qquad \min_{\mathrm{P}\in\mathcal{P}} \{\mathrm{P}[g(x,\xi) \leq 0]\} \geq 1-\varepsilon$$

The underlying paradigm of distributionnally robust optimization has first been introduced in the economics literature under the name of *ambiguity* and *min-max stochastic programming*, applied for instance by Scarf to the newsvendor problem in [237].

Regarding optimization, the term of *distributionally robust optimization* was coined by Calafiore and El-Ghaoui in [68] where it was shows that the satisfaction of an individual affine chance-constraint over a distribution class defined by the first two moments yields a second-order conic constraint. Iyengar and Erdogan [148] proposed to approximate the problem by satisfying the chance-constraint on a sample of the distribution class. Several works very close to our framework concern the problem of the minimization of an expected value under ambiguity, without considering chance-constraints. See for instance [82, 30].

In this paper, we implement the approach proposed by Zymler, Kuhn and Rustem in [270]. The objective is to approximate a joint chance-constraint by Worst-Case CVaR in a distributionally robust framework where the support and the two first moments are known. The key point is that the obtained problem admits a formulation in the form of a semidefinite program. More precisely, similarly to the process proposed in [72] and described at Paragraph 6.5, the joint constraints are converted into an individual one and the CVaR approximation is applied. An exact formulation of the worst-case over the considered class of distribution is found as a semidefinite program.

What is outstanding in this method is that, in the individual case, it appears that the obtained semidefinite constraint is exactly equivalent to the original chance-constraint, i.e., the loss attributable to the CVaR approximation vanishes. In the case of a joint chance-constraint, the original constraint is equivalent to considering the approximated constraint for all the value of the scaling parameters α. In the next section, we show that we obtain the same result by considering the problem from a moment problem perspective.

6.2.2 Unification of the distributionally robust approach for chance-constraint with the moment approach

The considered problem (6.4) is equivalent to $\min_{x\in F} c^T x : p^*(x) \geq 1-\varepsilon$, where $p^*(x)$ is the optimal value of the following moment problem :

$$\begin{cases} p^*(x) = & \min & \mathrm{P}[g_t(x,\xi) \leq 0,\ t = 1,...,T] & \\ & \text{s.t.} & \mathrm{P} \in \mathcal{P}(\mathcal{S}) & \text{with} \quad \mathcal{P}(\mathcal{S}) = \{\mathrm{P} \in \mathcal{M}(\mathcal{S}) : \Omega_{\mathrm{P}}(\xi) = \Omega\} \end{cases} \tag{6.10}$$

In this section, we propose a very simple process, based on the dual of the problem (6.10) and on the S-Lemma, to derive a conservative approximation of the constraint $p^*(x) \geq 1-\varepsilon$ in the form of a semidefinite constraint. Very roughly, we replace $p^*(x) \geq 1-\varepsilon$ by $d^*(x) \geq 1-\varepsilon$. As a consequence of weak duality, this is sufficient to guarantee the feasibility of the solution, and when strong duality holds (which we assume is the case here), this is also necessary. As the dual moment problem is a maximization problem, the constraint holds if and only if there exists a dual solution with an objective value greater or equal to $1-\varepsilon$. In some particular cases, it follows directly from our rationale that the obtained approximation is exact.

Then we show that the obtained SDP is similar both to the one established in [270] and to the first rank of the Lasserre's hierarchy of the moment problem (6.10). This problem can also be seen as an instance of the probability bounding problem defined by Bertsimas and described in the Paragraph 3.4.4.1. It suffices to consider the T-dimensional random vector $\omega = -g(x,\xi)$ and the probability $\mathrm{P}[\omega \in \mathbb{R}^T_+]$. Finally, in the individual case without support requirement ($\mathcal{S} = \mathbb{R}^m$), the semidefinite formulation reduces to a Second-Order Conic constraint, as pointed out by Calafiore and El-Ghaoui in [68].

Table 6.4 contains a summary of the different cases studied in this section. For each paragraph, we provide m, n, T, S that correspond to the notations defined at the end of the introduction. D denotes the order of moments considered to define the class $\mathcal{P}$. More practically, $D = 1$ if we consider only the expected value μ of ξ, whereas $D = 2$ if we consider both its expected value μ and the covariance Σ. The other columns are as follows :

- *Bound* : indicates the bound that is obtained or a reference to the equation at hand;
- *Num. Study* : the paragraph where a numerical study of the particular case at hand can be found;
- *Reference* : papers containing a SDP approximation that is subsumed in the paragraph ;
- *Exact ?* : *Yes* if the approximation is exact, *No* otherwise.

Paragraph	m	n	T	S	D	Bound	Num. Study	References	Exact ?
6.2.2.1	≥ 1	0	1	≥ 0	1	(6.13)	§ 6.2.3.3		No
6.2.2.1	≥ 1	0	1	≥ 0	2	(6.12)	§ 6.2.3.3		No
6.2.2.2	1	0	1	0	1	$= 1$			Yes
6.2.2.2	1	0	1	0	2	Cantelli's bound	§ 6.2.3.3	[41, 68]	Yes
6.2.2.2	1	0	1	≥ 1	2	(6.15)	§ 6.2.3.3	[41]	Yes
6.2.2.2	1	0	1	≥ 1	1	Markov's bound			Yes
6.2.2.3	≥ 1	≥ 0	1	≥ 0	2	(6.17)			No
6.2.2.4	≥ 1	≥ 0	≥ 1	≥ 0	2	(6.18)	§ 6.2.3.3	[257]	No

Table 6.4: Summary of the considered cases

6.2.2.1 Case of an individual chance-constraint ($T = 1$) without command variables ($n = 0$)

In this section, we consider the individual case ($T = 1$) of the chance-constraint involved in the problem (6.3), without dependency to the command variable x ($n = 0$). By calling d the first row of A^1, the

constraint becomes :

$$\mathrm{P}[d^T\tilde{\xi} \leq 0] \geq 1-\varepsilon,\ \forall \mathrm{P} \in \mathcal{P} = \{\mathrm{P} \in \mathcal{M}(\mathcal{S}) : \Omega_{\mathrm{P}}(\xi) = \Omega\}$$

with $\mathcal{S} = \{\xi \in \mathbb{R}^m : \tilde{\xi}^T W^s \tilde{\xi} \leq 0\}$. The use of the matrix $\Omega = \begin{pmatrix} 1 & \mu^T \\ \mu & \mu\mu^T + \Sigma \end{pmatrix}$ implicitly supposes that the expected value μ and covariance Σ of ξ are available but for comparison, we also consider the case where only μ is available.

By considering the dual of this moment problem and applying the S-Lemma (Lemma 6.2.1), we derive a conservative approximation of this moment problem in the form of a semidefinite system. In the case where $S \leq 1$, it follows from the S-Lemma that this conservative approximation is exact. We show that the obtained SDP is equivalent to the SDP proposed by Zymler in [270] and to the rank 1 of the Lasserre hierarchy [171].

Let us first consider the case that involves covariance. With $n = 0$, $T = 1$ and $g(x, \xi) = d^T\tilde{\xi}$, the subproblem (6.10) reduces to the following instance of the GPM, for which we provide the dual problem obtained by applying the duality relationship of (6.6) :

$$\left\{\begin{array}{lll} p^* = & \min & \mathrm{P}[d^T\tilde{\xi} \leq 0] \\ & \text{s.t.} & \Omega_{\mathrm{P}} = \Omega \\ & & \mathrm{P} \in \mathcal{M}(\mathcal{S}) \end{array}\right. \quad \text{dual with} \quad \left\{\begin{array}{lll} d^* = & \max & 1 - \Omega \bullet M \\ & \text{s.t.} & \tilde{\xi} M \tilde{\xi} \geq 1,\ \forall \xi \in \mathcal{S} :\ d^T\tilde{\xi} \geq 0 \\ & & \tilde{\xi} M \tilde{\xi} \geq 0,\ \forall \xi \in \mathcal{S} \end{array}\right. \tag{6.11}$$

Proof 6.2.2 *We define $\mathcal{K} = \{\xi \in \mathbb{R}^m : d^T\tilde{\xi} \leq 0\}$. Then, we recover the problem (6.6) with $h = \mathbb{1}_{\mathcal{K}}$ and f_j the monomials of degree 0,1 and 2. Consequently, the dual problem writes :*

$$\left\{\begin{array}{ll} \max & z_{0,0} + \sum_{i=1}^m \Omega_{0,i} z_{0,i} + \sum_{i \leq j} \Omega_{i,j} z_{i,j} \\ s.t. & z_{0,0} + \sum_{i=1}^m z_{0,i}\xi_i + \sum_{i \leq j} z_{i,j}\xi^i\xi^j \leq \mathbb{1}_{\mathcal{K}}(\xi),\ \forall \xi \in \mathcal{S} \end{array}\right.$$

We define $M' \in \mathbb{S}^{m+1}$ the matrix indexed by $i, j = 0, ..., m$ such that $M'_{i,i} = z_{i,i}$ for $i = 0, ..., m$, $M'_{i,j} = z_{i,j}/2$ for $i, j = 1, ..., m,\ i \neq j$ and we develop the constraint depending on whether $\xi \in \mathcal{K}$ or not :

$$\left\{\begin{array}{ll} \max & \Omega \bullet M' \\ s.t. & \tilde{\xi} M' \tilde{\xi} \leq 0,\ \forall \xi \in \mathcal{S} \cap \mathcal{K}^C \\ & \tilde{\xi} M' \tilde{\xi} \leq 1,\ \forall \xi \in \mathcal{S} \cap \mathcal{K} \end{array}\right.$$

By a change of variable : $M = M(e_0) - M'$, the problem becomes :

$$\left\{\begin{array}{ll} \max & 1 - \Omega \bullet M \\ s.t. & \tilde{\xi} M \tilde{\xi} \geq 1,\ \forall \xi \in \mathcal{S} \cap \mathcal{K}^C \\ & \tilde{\xi} M \tilde{\xi} \geq 0,\ \forall \xi \in \mathcal{S} \cap \mathcal{K} \end{array}\right.$$

As $\tilde{\xi} M \tilde{\xi} \geq 1 \Rightarrow \tilde{\xi} M \tilde{\xi} \geq 0$, we can replace $\forall \xi \in \mathcal{S} \cap \mathcal{K}$ by $\forall \xi \in \mathcal{S}$. Furthermore, $\mathcal{K}^C = \{\xi \in \mathbb{R}^m : d^T\tilde{\xi} > 0\}$ but by continuity of the polynomial $\tilde{\xi} M \tilde{\xi}$, the associated constraint is equivalent by considering the closure of $\mathcal{K}^C$. Thus we recover the dual form of (6.11). □

Remark 6.2.3 *We can express the dual of the moment problem $\max_{\mathrm{P} \in \mathcal{P}(\mathbb{R}^m)} \mathrm{P}[d^T\tilde{\xi} \geq 0]$ in the same way. Then, we observe that $d^* \geq 1 - \varepsilon$ is equivalent to $\max_{\mathrm{P} \in \mathcal{P}(\mathbb{R}^m)} \mathrm{P}[d^T\tilde{\xi} \geq 0] \leq \varepsilon$. This proves that it is equivalent to consider $\mathrm{P}[d^T\tilde{\xi} \leq 0] \geq 1 - \varepsilon$ or $\mathrm{P}[d^T\tilde{\xi} \geq 0] \leq \varepsilon$ whenever strong duality holds.*

Then, it suffices to apply the S-lemma (Lemma 6.2.1) to the two constraints of this problem to get a conservative approximation of the dual problem under the form of a SDP :

$$\left\{\begin{array}{lll} d^* = & \max & 1-\Omega\bullet M \\ & \text{s.t.} & M - M(e_0) - \sum_{s=1}^{S}\lambda_{1,s}W^s - \tau M(d) \succcurlyeq 0 \\ & & M - \sum_{s=1}^{S}\lambda_{0,s}W^s \succcurlyeq 0 \\ & & \lambda_0 \geq 0,\ \lambda_1 \geq 0,\ \tau \geq 0 \end{array}\right.$$

This problem is equivalent to the rank 1 of semidefinite relaxations of Lasserre (6.9). To see this, it suffices to recall that the matrix M contains the variables z and therefore $b^T z = 1 - \Omega \bullet M$. Furthermore the two LMI correspond to the matrices N_0^1 and N_0^2 respectively. $\lambda_{0,s}$ and $\lambda_{1,s}$ corresponds to N_s^1 and N_s^2 respectively for $s = 1, ..., S$ and τ corresponds to N_{S+1}^1.

Finally, a sufficient condition for $d^* \geq 1 - \varepsilon$ to hold is that the following system be feasible :

$$\left\{\begin{array}{l} \Omega \bullet M \leq \varepsilon \\ M - \sum_{s=1}^{S}\lambda_{0,s}W^s \succcurlyeq 0 \\ M - M(e_0) - \sum_{s=1}^{S}\lambda_{1,s}W^s - \tau M(d) \succcurlyeq 0 \\ \lambda_0 \geq 0,\ \lambda_1 \geq 0,\ \tau \geq 0 \end{array}\right. \tag{6.12}$$

The approximation of the problem (6.11) proposed by Zymler et al. in [270], which is as follows, is equivalent to (6.12) :

$$\left\{\begin{array}{l} \Omega \bullet M \leq \varepsilon\beta \\ M - \sum_{s=1}^{S}\lambda_{0,s}W^s \succcurlyeq 0 \\ M - \beta M(e_0) - \sum_{s=1}^{S}\lambda_{1,s}W^s - M(d) \succcurlyeq 0 \\ \lambda_0 \geq 0,\ \lambda_1 \geq 0 \end{array}\right.$$

Proof 6.2.4 *According to Fejer's theorem, $\Omega \succ 0$ and $M \succcurlyeq 0$ implies that $\Omega \bullet M \geq 0$ and therefore $\beta \geq 0$. Furthermore, $\beta = 0$ implies $M = 0$ which is impossible since the first semidefinite constraint can not be satisfied. Consequently, $\beta > 0$ and we obtain the problem (6.12) by substituting $1/\beta$ with τ, M/β with M, λ_0/β with λ_0 and λ_1/β with λ_1.* □

In the case where the covariance is not known, i.e., $\mathcal{P} = \{\mathrm{P} \in \mathcal{M}(\mathcal{S}) : \mathrm{E}_{\mathrm{P}}(\xi) = \mu\}$, the problem can be approximated with the same rationale. The constraint corresponding to the second-order moments are removed in the primal, which corresponds to variables imposed to 0 in the dual, i.e., $M_{i,j} = 0,\ i, j = 1, ..., m$. Then, by noting $z \in \mathbb{R}^{m+1}$ the remaining variables, the problem becomes :

$$\left\{\begin{array}{l} \tilde{\mu}^T z \leq \varepsilon \\ M(z) - \sum_{s=1}^{S}\lambda_{0,s}W^s \succcurlyeq 0 \\ M(z) - M(e_0) - \sum_{s=1}^{S}\lambda_{1,s}W^s - \tau M(d) \succcurlyeq 0 \\ \lambda_0 \geq 0,\ \lambda_1 \geq 0,\ \tau \geq 0 \end{array}\right. \tag{6.13}$$

In summary, we derived a SDP approximation of the moment problem that lies implicitly in the problem (6.4). The obtained SDP is similar to those obtained by Lasserre in [171] and Zymler et al. in [270], however the underlying rationale are quite different. For our part, we place ourself in a setting where all the functions of ξ are polynomials of degree less than 2. Thanks to this restriction, we can apply the S-Lemma, which provides a very simple way of obtaining the SDP.

The Lasserre's approach is much more general since the considered polynomials can be of any degrees. Furthermore, he obtains a hierarchy of SDP approximation whose optimal values tends to the optimal value of the initial problem, whereas we only design the approximation corresponding to the rank 1 of this hierarchy. One benefit of this formulation is that it allows to maximize the probability that $d^T\tilde{\xi} \leq 0$ by minimizing the value of ε. On the downside, the extension for involving a command variable x is not compatible with the Lasserre's formulation. Thus, we have to choose between the maximization of the probability and the consideration of a command variable.

The rationale given in the paper of Zymler et al. [270] is different too. Indeed, the SDP is obtained by first considering the classical CVaR approximation of the probability and using the formulation of the CVaR as the optimal value of a minimization problem. Then, exploiting the existence of a convenient saddle point and applying the S-Lemma, leads to the SDP at hand.

In the next paragraph, we apply this result to the particular case when $m = 1$ and $g(x, \xi) = \xi$.

6.2.2.2 Case of the individual chance-constraint $\mathrm{P}[\xi \leq 0] \geq 1 - \varepsilon$

In this paragraph, we apply the results of the previous paragraph to the case when $m = 1, n = 0, T = 1$ and $g(x, \xi) = \xi$. This leads to the study of the problem $\min_{\mathrm{P}\in\mathcal{P}} \mathrm{P}[\xi \leq 0]$. We study three cases depending on whether we consider the variance Σ or not and if we consider that the support is known or not. For these cases, we show that that the obtained approximation is exact. This is obvious when the support is not considered since $S = 0$ and $\mathcal{S} \cap \mathcal{K} = \mathbb{R}^m_+$ has a concise quadratic representation. We will see that this is still the case by injecting support information, even for $S > 1$.

Without variance and support
We start to study the case where no assumption is made about the support of ξ and the information about the covariance is neglected. It suffices to apply the problem (6.13) which becomes :

$$\begin{cases} z_0 + \mu z \leq \varepsilon \\ z_0 + \xi z \geq 0, \ \forall \xi \\ z_0 + \xi z \geq 1, \ \forall \xi \geq 0 \end{cases}$$

The optimal value of this problems equals 1.

Proof 6.2.5 *$z_0 + \xi z \geq 0, \ \forall \xi \Rightarrow z = 0$, $z_0 + \xi z \geq 1, \ \forall \xi \geq 0 \Rightarrow z_0 \geq 1$ so the optimal value is greater than 1. As $z_0 = 1$, z_0 is feasible, the optimal value equals 1.* $\square$

Indeed, for any value of μ, we can build a probability distribution such that $\mathrm{P}[\xi > 0] = p$, for any value of $0 < p < 1$. It suffices for instance to consider the distribution such that $\mathrm{P}[\xi = v_1] = p$ and $\mathrm{P}[\xi = (\mu - pv_1)/(1-p)] = 1 - p$.

With variance but without support
By contrast, considering the variance Σ leads to an interesting bound on $\mathrm{P}[\xi \geq 0]$. Indeed, deriving the problem (6.12) leads to the following system :

$$\begin{cases} \Omega \bullet M \leq \varepsilon \\ M - \begin{pmatrix} 1 & \tau \\ \tau & 0 \end{pmatrix} \succcurlyeq 0 \\ M \succcurlyeq 0, \ \tau \geq 0 \end{cases}$$

Minimizing ε so that the system be feasible leads to the following primal and dual SDP that yields an upper bound of $\mathrm{P}[\xi \geq 0]$:

$$\begin{cases} \min & \Omega \bullet M \\ \text{s.t.} & M - \begin{pmatrix} 1 & \tau \\ \tau & 0 \end{pmatrix} \succcurlyeq 0 \\ & M \succcurlyeq 0, \ \tau \geq 0 \end{cases} \qquad \text{dual with} \qquad \begin{cases} \max & x \\ \text{s.t.} & \Omega \succcurlyeq \begin{pmatrix} x & y \\ y & z \end{pmatrix} \succcurlyeq 0 \\ & y \geq 0 \end{cases} \tag{6.14}$$

Then, the dual optimal value is $\frac{\Sigma}{\mu^2+\Sigma}$ if $\mu < 0$ and 1 otherwise.

Proof 6.2.6 *The constraint* $\Omega \succcurlyeq \begin{pmatrix} x & y \\ y & z \end{pmatrix}$ *implies that* $1 - x \geq 0$ *so the dual optimal value is less than* 1. *If* $\mu \geq 0$, $\begin{pmatrix} x & y \\ y & z \end{pmatrix} = \begin{pmatrix} 1 & \mu \\ \mu & \mu^2 + \Sigma \end{pmatrix}$ *is feasible and therefore the optimal value equals* 1. *If* $\mu < 0$, $\Omega \succcurlyeq \begin{pmatrix} x & y \\ y & z \end{pmatrix}$ *implies that* $x \leq 1 - \frac{(\mu-y)^2}{\mu^2+\Sigma-z}$. *The right-hand side is a decreasing function of* y *and* z, *so we take for* y *and* z *the smallest feasible value, i.e.,* $y = 0$ *and* $z = 0$, *which leads to the optimal value* $1 - \frac{\mu^2}{\mu^2+\Sigma} = \frac{\Sigma}{\mu^2+\Sigma}$. □

Remark 6.2.7 *This bound is stated by the Cantelli's inequality, a generalization of the Chebyshev's inequality that states that*

$$\mathrm{P}[X - \mu \geq \lambda] \begin{cases} \leq \frac{\Sigma}{\Sigma+\lambda^2} & \text{if } \lambda > 0 \\ \geq \frac{\lambda^2}{\Sigma+\lambda^2} & \text{if } \lambda < 0 \end{cases}$$

This bound leads to the following equivalence, if $\mu < 0$:

$$\begin{aligned} \mathrm{P}[\xi \geq 0] \leq \varepsilon \quad & \Leftrightarrow \tfrac{\Sigma}{\mu^2+\Sigma} \leq \varepsilon \\ & \Leftrightarrow \Sigma \leq \tfrac{\varepsilon}{1-\varepsilon}\, \mu^2 \\ & \Leftrightarrow \sqrt{\Sigma} \leq -\sqrt{\tfrac{\varepsilon}{1-\varepsilon}}\, \mu \end{aligned}$$

Then it suffices to replace the single random variable ξ by $\tilde{x}^T\xi$, with $\xi \in \mathbb{R}^m, x \in \mathbb{R}^{m-1}$, to recover the SOCP formulation proposed in [68] for the chance-constraint $\mathrm{P}[\tilde{x}^T\xi \geq 0] \leq \varepsilon$, given the mean μ' and covariance Σ' of ξ, namely $\sqrt{\tilde{x}^T\Sigma'\tilde{x}} + \sqrt{\frac{\varepsilon}{1-\varepsilon}}\, \tilde{x}^T\mu' \leq 0$.

Proof 6.2.8 *The result follows immediately be replacing* Σ *by* $\tilde{x}^T\Sigma'\tilde{x}$, *and* μ *by* $\tilde{x}^T\mu'$. □

In conclusion, we make the connection with the work of Bertsimas and Popescu presented in [41]. They provide SDP formulation for bounding the probability $\mathrm{P}[\xi \in \mathcal{K}]$ ($m = 1$) given the first k moments of ξ, for different forms of set $\mathcal{K}$. We recover the problem at hand in this paragraph by taking $\mathcal{K} = \mathbb{R}_+$ and $k = 2$.

Then, the SDP proposed in [41] is :

$$\begin{cases} \min & \Omega \bullet \begin{pmatrix} z_0 & z_1 \\ z_1 & z_2 \end{pmatrix} \\ \text{s.t.} & \begin{pmatrix} z_0 & 0 & \nu_1 \\ 0 & z_1 - \nu_2 & 0 \\ \nu_1 & 0 & z_2 \end{pmatrix} \succcurlyeq 0 \\ & \begin{pmatrix} z_0 & z_1 \\ z_1 & z_2 \end{pmatrix} \succcurlyeq 0 \\ & \nu_1, \nu_2 \in \mathbb{R} \end{cases}$$

By permuting the columns and rows 2 and 3 of the 3×3 matrix and taking $\nu_1 = z_1$ and $\nu_2 = z_1 - \tau$, it becomes $\begin{pmatrix} z_0 & z_1 & 0 \\ z_1 & z_2 & 0 \\ 0 & 0 & \tau \end{pmatrix}$. Then it suffices to define $M = \begin{pmatrix} z_0 & z_1 \\ z_1 & z_2 \end{pmatrix}$ to recover the problem (6.14).

With variance and support

This paragraph focuses on the case where the variance and the support are considered. We start by proving the following lemma :

Lemma 6.2.9 *Let $\mathcal{S} = \{\xi \in \mathbb{R} : \tilde{\xi} W^s \tilde{\xi} \geq 0,\ s = 1, ..., S\}$. Then, $\mathcal{S}$ admits a concise quadratic representation, i.e., there exists a matrix W such that $\mathcal{S} = \{\xi \in \mathbb{R} : \tilde{\xi} W \tilde{\xi} \geq 0\}$.*

Proof 6.2.10 *We define $\mathcal{S}_s = \{\xi \in \mathbb{R} : \tilde{\xi} W^s \tilde{\xi} \geq 0\}$ for $s = 1, ..., S$. Then $\mathcal{S}_s$ can be either :*

- *the empty set (if $W^s \prec 0$);*
- *a bounded interval (if $W^s_{2,2} \leq 0$) ;*
- *a half-bounded interval (in the linear case, i.e., $W^s_{2,2} = 0$) ;*
- *the union of two half-bounded intervals (if $W^s_{2,2} \geq 0$) ;*
- *$\mathbb{R}$ (if $W^s \succcurlyeq 0$) ;*

For each of this kind of set, it is easy to determine a matrix W such that the set be represented in the form $\{\xi \in \mathbb{R} : \tilde{\xi} W \tilde{\xi} \geq 0\}$. Furthermore, the intersection of such sets takes necessarily one of these 5 forms. To see this, it suffices to consider all the possible kinds of intersection of two of these sets, then to proceed recursively. It follows that $\mathcal{S}$ admits a concise quadratic representation. □

Consequently, with $m = 1$, the problem boils down to a problem for which the SDP approximation is exact. If $\mathcal{S} = \{\xi \in \mathbb{R}^m : \tilde{\xi}^T W \tilde{\xi} \geq 0\}$ and $\mathcal{S} \cap \mathcal{K} = \{\xi \in \mathbb{R}^m : \tilde{\xi}^T Y \tilde{\xi} \geq 0\}$, then the constraint $\mathrm{P}[\xi \leq 0] \leq 1 - \varepsilon,\ \forall P \in \mathcal{P}$ is equivalent to :

$$\begin{cases} \Omega \bullet M \leq \varepsilon \\ M - \lambda W \succcurlyeq 0 \\ M - M(e_0) - \tau Y \succcurlyeq 0 \\ \lambda \geq 0,\ \tau \geq 0 \end{cases} \tag{6.15}$$

In particular, if $\mathcal{S} = [a, b]$ with $a < 0 < b$, then $\mathcal{S} \cap \mathcal{K} = [0, b]$ and it suffices to take $W = \begin{pmatrix} -ab & (a+b)/2 \\ (a+b)/2 & -1 \end{pmatrix}$ and $Y = \begin{pmatrix} 0 & b/2 \\ (b/2 & -1 \end{pmatrix}$.

In conclusion, by using the same rationale as in the previous paragraph, we show that the obtained SDP is equivalent to the SDP proposed by Bertsimas et al. in [41] for the different form of support.

With support but without variance

In this paragraph, we consider the case where $\mathcal{P} = \{\mathrm{P} \in \mathcal{M}(\mathcal{S}) : \mathrm{E}_{\mathrm{P}}(\xi) = \mu\}$.We show that in this case, we recover the Markov's inequality : for any variable ξ of expected value μ such that $\mathrm{P}[\xi \geq a] = 1$, $\mathrm{P}[\xi \geq 0] \leq 1 - \mu/a$. By taking the complement, it follows that $\mathrm{P}[\xi \leq 0] \geq \mu/a$.

To this end, we assume that $\mathcal{S}$ is bounded below, i.e., that there exists $a \in \mathbb{R}$ such that $a = \min \xi : \xi \in \mathcal{S}$. We also assume that $a < 0$, $\mu \leq 0$ and $\mathcal{S} \cap \mathbb{R}_+ \neq \emptyset$ otherwise the bound $\min_{\mathrm{P} \in \mathcal{P}} \mathrm{P}[\xi \leq 0]$ is trivial.

As explained in the previous paragraph, the S-Lemma is lossless in this case and therefore, it is equivalent to consider directly the dual of the moment problem, as presented in (6.11). Then the problem becomes :

$$\begin{cases} \max & 1 - z_0 - \mu z \\ \text{s.t.} & z_0 + \xi z \geq 1,\ \forall \xi \in \mathcal{S} \cap \mathbb{R}_+ \\ & z_0 + \xi z \geq 0,\ \forall \xi \in \mathcal{S} \end{cases}$$

Then, the optimal value of this problem is the Markov's bound μ/a.

Proof 6.2.11 *We consider the two cases $z \leq 0$ and $z \geq 0$ successively.*

(1) if $z \leq 0$, as $\mathcal{S} \cap \mathbb{R}_+ \neq \emptyset$, the second constraint implies that $z_0 \geq 1$. Then $(z_0, z) = (1, 0)$ is optimal, since taking $z_0 > 1$ or $z < 0$ would decrease the objective. Consequently the associated optimal value is 0.

(2) if $z \geq 0$, then the affine function $z_0 + \xi z$ is an increasing function in ξ and takes its minimal value on the left side of the considered set. Then the constraints becomes : $z_0 \geq 1$ and $z \leq -z_0/a$. The optimal value of μ/a is then attained for $z_0 = 1$ and $z = -1/a$. $\square$

6.2.2.3 Injecting dependence to a command variable x ($n \geq 1$)

We consider an additional level of complexity by assuming that the probability depends of a command variable x through the function $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, affine in (x, ξ) : $g(x, \xi) = \tilde{x}\, A\, \tilde{\xi}$. Thus, the individual chance-constraint is as following :

$$\min_{\mathrm{P} \in \mathcal{P}(\mathcal{S})} \mathrm{P}[g(x, \xi) \leq 0] \geq 1 - \varepsilon \tag{6.16}$$

A first way of handling this constraint is to consider $g(x, \xi)$ as a single random variable with moment matrix $\Omega(x) = \begin{pmatrix} 1 & x^T A \mu \\ x^T A \mu & (Axx^T A) \bullet \Sigma \end{pmatrix}$. However the problem involves a quadratic function of x and is therefore difficult to handle. The only case where this is done is discussed at the end of the paragraph 6.2.2.2 and leads to the SOCP formulation proposed in [68].

Another possibility is to consider the probability from a different perspective, as $\mathrm{P}[\xi \in \mathcal{K}(x)]$, with $\mathcal{K}(x) = \{\xi \in \mathbb{R}^m : g(x, \xi) \leq 0\}$. This set takes the form of a semi-algebraic set : $\mathcal{K}(x) = \{\xi \in \mathbb{R}^m : \tilde{\xi} Y(x) \tilde{\xi} \geq 0\}$ with $Y(x) = M(A^T x)$. Note that a more general form of $Y(x)$ allows to consider any function g that is quadratic, nor linear, w.r.t. ξ, provided that $Y(x)$ is not negative definite.

Then, we obtain the following SDP approximation. The right-hand side problem is its reformulation according to the formalism of Zymler et al. [270] :

$$\left\{ \begin{array}{l} \Omega \bullet M \leq \varepsilon \\ M - \sum_{s=1}^{S} \lambda_{0,s} W^s \succcurlyeq 0 \\ M - \sum_{s=1}^{S} \lambda_{1,s} W^s - M(e_0) - \tau Y(x) \succcurlyeq 0 \\ \lambda_0 \geq 0,\ \lambda_1 \geq 0,\ \tau \geq 0 \end{array} \right. \quad \text{or} \quad \left\{ \begin{array}{l} \Omega \bullet M \leq \varepsilon\beta \\ M - \sum_{s=1}^{S} \lambda_{0,s} W^s \succcurlyeq 0 \\ M - \sum_{s=1}^{S} \lambda_{1,s} W^s - \beta M(e_0) - Y(x) \succcurlyeq 0 \\ \lambda_0 \geq 0,\ \lambda_1 \geq 0 \end{array} \right. \tag{6.17}$$

The Zymler's formulation offers the major advantage of not multiplying the matrix $Y(x)$ by a variable τ, which makes it tractable, contrarily to the Lasserre's formulation. On the other hand, the Lasserre's formulation can be applied to the minimization of a probability and is therefore more desirable when there is no command variable x.

We observe that this system is not feasible for $\varepsilon < 1$ unless $Y(x) \bullet \Omega \leq 0$, which is equivalent to require that $\mathrm{E}_{\mathrm{P}}(g(x, \xi) \leq 0)$. Indeed, by Fejer's theorem :

$$\left. \begin{array}{l} \Omega \succ 0 \\ M - \sum_{s=1}^{S} \lambda_{1,s} W^s - M(e_0) - \tau Y(x) \succcurlyeq 0 \end{array} \right\} \Rightarrow \Omega \bullet M - 1 \geq \sum_{s=1}^{S} \lambda_{1,s} \Omega \bullet W^s + \tau \Omega \bullet Y(x) \geq 0$$

From a probabilistic point of view, this means that if the constraint is not satisfied on average, then there exists a probability distribution such that $\mathrm{P}[g(x, \xi) \leq 0] = 0$.

In what follows, we discuss the exactness of the obtained SDP approximation (6.17). We recall that this approximation is based on two transformations : the switch to the dual moment problem and

the S-Lemma. The resulting condition is always sufficient and let us discuss successively the case when it is also necessary, or in other words, when it is *lossless*.

Regarding the duality switch, its lossless is equivalent to strong duality, for which Theorem 3.4.4 provides sufficient conditions. However, checking whether the conditions (ii) and (iii) holds is a challenging task, even in the quadratic case, and we only assume here that strong duality holds. Let us just mention that in the case where $\mathcal{S} = \mathbb{R}^m$, the primal feasibility is equivalent to $\Omega \succcurlyeq 0$ (see for instance [41]) and Ω lies in the interior of the moment space if and only if $\Omega \succ 0$.

Regarding the S-Lemma, it is lossless if both $\mathcal{S}$ and $\mathcal{S} \cap \text{cl}(\mathcal{K}^C)$ admits a so-called *concise quadratic representation*, i.e., $\mathcal{S} = \{\xi \in \mathbb{R}^m : \tilde{\xi} W \tilde{\xi} \geq 0\}$ and $\mathcal{S} \cap \text{cl}(\mathcal{K}^C) = \{\xi \in \mathbb{R}^m : \tilde{\xi} Y \tilde{\xi} \geq 0\}$. Regarding the support, this is equivalent to require that $S \leq 1$.

Remark 6.2.12 *In this case, by applying the S-Lemma, $\xi \in \mathcal{S} \cap \text{cl}(\mathcal{K}^C) \Rightarrow \xi \in \mathcal{S}$ implies that $W - \nu Y \succcurlyeq 0$ for some real $\nu \geq 0$.*

Another case for which the S-Lemma is lossless is the linear case, for which the S-Lemma boils down to the Farkas Lemma :

Lemma 6.2.13 *Let $p, q_s,\ s = 1, ..., S$ be $m+1$-dimensional symmetric vectors. Then*

$$\left.\begin{array}{l} p - \sum_{s=1}^{S} \lambda_s q_s \geq 0 \\ \lambda \geq 0 \end{array}\right\} \Leftrightarrow p^T \tilde{\xi} \geq 0,\ \forall \xi \in \{\xi \in \mathbb{R}^m : q_s^T \tilde{\xi} \geq 0,\ s = 1, ..., S\}$$

Consequently, if $\mathcal{S}$ is a polyhedron, as $\mathcal{K}$ is also defined linearly since $g(x, \xi)$ is affine in ξ, the S-Lemma is lossless as soon as there exists a vector v such that $M = M(v)$ is solution of the system (6.17).

We deduce from this that if $\mathcal{S}$ is a polytope and for $\varepsilon = 1$, the SDP approximation is equivalent to the deterministic constraint obtained by replacing ξ by its mean $\mu : \tilde{x}^T A \tilde{\mu} \leq 0$.

Proof 6.2.14 *We define $\mathcal{F}_m = \{x \in \mathbb{R}^n : \tilde{x}^T A \tilde{\mu} \leq 0\}$ and $\mathcal{F}_\varepsilon$ the subset of $\mathbb{R}^n$ for which the Zymler's formulation of the system 6.17 is feasible. Then we show that $\mathcal{F}_m = \mathcal{F}_1$.*

(1) We prove that for any $\varepsilon \in [0, 1]$, $\mathcal{F}_\varepsilon \subset \mathcal{F}_m$. This is straightforward by applying Fejer's theorem to $\Omega \succcurlyeq 0$ and $M - \beta M(e_0) - \sum_{s=1}^{S} \lambda_{1,s} W^s - Y(x) \succcurlyeq 0$. Hence, $\Omega \bullet M - \sum_{s=1}^{S} \lambda_{1,s} \Omega \bullet W^s - \beta - \Omega \bullet Y(x) \geq 0$. As $\Omega \bullet M \leq \varepsilon\beta$ and $\Omega \bullet Y(x) = \tilde{x}^T A \tilde{\mu}$, it comes that $\tilde{x}^T A \tilde{\mu} \leq -[(1-\varepsilon)\beta + \sum_{s=1}^{S} \lambda_{1,s} \Omega \bullet W^s]$. Then the result can be deduce from $\beta \geq 0$ and $\Omega \bullet W^s \geq 0,\ s = 1, ..., S$.

(2) We prove that $\mathcal{F}_m \subset \mathcal{F}_1$. Let us consider $x_0 \in \mathcal{F}_m$. As $\mathcal{S}$ is compact, the function $\tilde{\xi}^T Y(x_0) \tilde{\xi}$ necessarily admits a minimum value γ over $\mathcal{S}$. Then, we define $M = Y(x_0) - \gamma M(e_0)$, which implies that $\tilde{\xi}^T M \tilde{\xi} \geq 0,\ \forall \xi \in \mathcal{S}$. Thus, we are in position to apply the "linear" form of the S-Lemma and it follows that there exists $\lambda_{0,s} \geq 0$ such that $M - \sum_{s=1}^{S} \lambda_{0,s} W^s \succcurlyeq 0$.

Then, if $\varepsilon = 1$, M is feasible for $\beta = -\gamma$. Indeed,

$$\left\{\begin{array}{ll} \Omega \bullet M - \beta = Y(x_0) \bullet M = \tilde{x}^T A \tilde{\mu} \leq 0 & \text{since } x \in \mathcal{F}_m \\ M - \sum_{s=1}^{S} \lambda_{0,s} W^s \succcurlyeq 0 & \text{by definition of } M \\ M - \sum_{s=1}^{S} \lambda_{1,s} W^s - \beta M(e_0) - Y(x) = 0 & \text{with } \lambda_{1,s} = 0,\ s = 1, ..., S \end{array}\right.$$

This proves that $x_0 \in \mathcal{F}_1$. $\square$

Finally, we show that this result can be applied to the case where $\mathcal{S}$ is a box, even when it is represented through quadratic constraints. Let us recall that there are two possible representations for a box $\mathcal{S} = \{\xi \in \mathbb{R}^m : a_i \leq \xi_i \leq b_i\}$: the linear one $\{\xi \in \mathbb{R}^m : \tilde{\xi}^T Z^i \tilde{\xi} \geq 0,\ i = 1, ..., 2m\}$ and the quadratic one $\{\xi \in \mathbb{R}^m : \tilde{\xi}^T W^i \tilde{\xi} \geq 0,\ i = 1, ..., m\}$ with

$$Z^{2i-1} = \begin{pmatrix} -a & 1/2 \\ 1/2 & 0 \end{pmatrix} \quad Z^{2i} = \begin{pmatrix} b & -1/2 \\ -1/2 & 0 \end{pmatrix} \quad W^i = \begin{pmatrix} -a_i b_i & (a_i + b_i)/2 \\ (a_i + b_i)/2 & -1 \end{pmatrix},\ i = 1, .., m$$

The above result can be applied by using the linear representation since a box is then a particular polytope. The result still holds by using the quadratic representation of $\mathcal{S}$. Indeed, the S-Lemma is lossless for one of these representations if and only if it is lossless for the other one. This can be easily seen from the fact that there exists $\pi_i \geq 0,\ i = 1, ..., 2m$ such that $Z^{2i-1} - \pi_{2i-1} W^i \succcurlyeq 0$ and $Z^{2i} - \pi_{2i} W^i \succcurlyeq 0$ for $i = 1, ..., m$ as a direct application of the S-lemma. Conversely, there exists $\rho_i \geq 0,\ i = 1, ..., m$ such that $W^i - \rho_i (Z^{2i-1} + Z^{2i}) \succcurlyeq 0$ by taking $\rho_i = (b_i - a_i)/2$. Then $Q - \sum_{i=1}^{2m} \lambda_i Z^i \succcurlyeq 0 \Rightarrow Q - \sum_{i=1}^{m} (\lambda_{2i-1}\pi_{2i-1} + \lambda_{2i}\pi_{2i}) W^i \succcurlyeq 0$ and $Q - \sum_{i=1}^{m} \lambda_i W^i \succcurlyeq 0 \Rightarrow Q - \sum_{i=1}^{m} \lambda_i \rho_i (Z^{2i-1} + Z^{2i}) \succcurlyeq 0$.

This indicates that the quadratic representation is preferable since it gives rise to an equivalent SDP approximation of smaller size.

In summary, we proved that in the case where $\mathcal{S}$ is a box, even quadratically represented, and $g(x, \xi)$ is affine in ξ then the SDP approximation proposed both by Lasserre [171] and Zymler [270] to handle the distributionnally robust version of the individual chance-constraint (6.16) is exact.

6.2.2.4 Handling a joint chance-constraint

The previous problem can be extended naturally to the case of joint chance-constraints, by defining $\mathcal{K}(x)$ through several constraints : $\mathcal{K}(x) = \{\xi \in \mathrm{R}^m : \tilde{\xi}\ Y^t(x)\ \tilde{\xi} \leq 0,\ t = 1, ..., T\}$. We define the sets $\mathcal{K}_t(x) = \{\xi \in \mathrm{R}^m : \tilde{\xi}\ Y^t(x)\ \tilde{\xi} \geq 0\}$ for $t = 1, ..., T$, hence the closure of $\overline{\mathcal{K}(x)}$ is $\bigcup_{t=1}^{T} \mathcal{K}_t(x)$ and the dual problem becomes :

$$\begin{cases} \Omega \bullet M \leq \varepsilon \\ \tilde{\xi} M \tilde{\xi} \geq 0,\ \forall \xi \in \mathcal{S} \\ \tilde{\xi} M \tilde{\xi} \geq 1,\ \forall \xi \in \mathcal{S} \cap \mathcal{K}_t(x),\ t = 1, ..., T \end{cases}$$

The following semidefinite program is therefore a conservative approximation of this constraint :

$$\begin{cases} \Omega \bullet M \leq \varepsilon \\ M - \sum\limits_{s=1}^{S} \lambda_{0,s} W^s \succcurlyeq 0 \\ M - M(e_0) - \sum\limits_{s=1}^{S} \lambda_{t,s} W^s + \tau_t Y^t(x) \succcurlyeq 0,\ t = 1, ..., T \\ \lambda \geq 0,\ \tau \geq 0 \end{cases} \tag{6.18}$$

The difficulty here is that we can not get rid of the non-linearity with the same trick as used above, since this trick work for only one non-linearity, or more precisely if there is only one variable τ to "hide". Generally, $\tau_t \neq \tau_{t'}$ so this is impossible. But there exists a scaling $\alpha \in \mathbb{R}^T_{++}$ of the constraints that induces the situation where $\tau_1 = ... = \tau_T$ is feasible. This means that we are now interested in the following chance-constraint, which is equivalent to the original one : $\min_{\mathrm{P} \in \mathcal{P}(\mathcal{S})} \mathrm{P}[\alpha_j g_j(x, \xi) \leq 0,\ j = 1, ..., T] \geq 1 - \varepsilon\}$.

For instance, if τ is feasible, the scaling $\alpha_j = 1/\tau_j$ works. In this case, the problem is equivalent to :

$$\begin{cases} \Omega \bullet M \leq \varepsilon\beta \\ M - \sum\limits_{s=1}^{S} \lambda_{0,s} W^s \succcurlyeq 0 \\ M - \sum\limits_{s=1}^{S} \lambda_{t,s} W^s - \beta M(e_0) - \alpha_t Y^t(x) \succcurlyeq 0, \ t = 1, ..., T \\ \lambda \geq 0 \end{cases} \tag{6.19}$$

This is the result of Zymler, saying that, for $s = 0$, there exists a scaling α for which this semidefinite system is equivalent to the original chance-constraint. Otherwise, for an arbitrary of α, this reformulation is equivalent to impose $\tau_1 = ... = \tau_T = 1/\beta$ and is the obtained problem is therefore a conservative approximation of the original chance-constraint.

Note that we recover the problem studied in [257] : $\min \mathrm{P}[\xi \in \mathcal{K}]$ with $\mathcal{K} = \{\xi \in \mathbb{R}^m : \tilde{\xi}^T Y^t \tilde{\xi} < 0, \ t = 1, ..., S\}$. This is actually a particular case of joint chance-constraint without support information and without command variables. The obtained SDP approximation, which was proved exact in [257], is as follows :

$$\begin{cases} p^* = & \max & \Omega \bullet M \\ & \text{s.t.} & M - M(e_0) + \tau_t Y^t \succcurlyeq 0, \ t = 1, ..., T \\ & & M \succcurlyeq 0, \ \tau \geq 0 \end{cases}$$

Finally, we investigate the SDP approximation obtained by converting the joint chance-constraint into several individual chance-constraint by means of the Boole's inequality. Instead of requiring that the joint probability be greater than $1 - \varepsilon$, we impose that the sum of the individual probability be greater than $1 - \varepsilon$ and the corresponding semidefinite program is as follows :

$$\begin{cases} \sum\limits_{t=1}^{T} \Omega \bullet M_t \leq \varepsilon \\ M_t - \sum\limits_{s=1}^{S} \lambda_{0,s} W^s \succcurlyeq 0, \ t = 1, ..., T \\ M_t - M(e_0) - \sum\limits_{s=1}^{S} \lambda_{t,s} W^s + \tau_t Y^t(x) \succcurlyeq 0, \ t = 1, ..., T \\ \lambda \geq 0, \ \tau \geq 0 \end{cases}$$

In order to "hide" the variables τ_t, we impose the probability level for each constraint, generally ε/T. Then, the problem becomes :

$$\begin{cases} \Omega \bullet M_t \leq \frac{\varepsilon}{T}\beta_t, \ t = 1, ..., T \\ M_t - \sum\limits_{s=1}^{S} \lambda_{0,s} W^s \succcurlyeq 0, \ t = 1, ..., T \\ M_t - \beta_t M(e_0) - \sum\limits_{s=1}^{S} \lambda_{t,s} W^s - Y^t(x) \succcurlyeq 0, \ t = 1, ..., T \\ \lambda \geq 0 \end{cases} \tag{6.20}$$

The advantages of this approximation are twofold : first, there is no loss due to the necessity of having $\tau_1 = ... = \tau_T$. Second, if the constraint t involves only $m_t \geq m$ components of ξ, then the matrix M_t can be of size $(m_t+1)(m_t+2)/2$ instead of $(m+1)(m+2)/2$, which can lead to a significant reduction of the problem size. This implies to define the appropriate matrices $W^{s,t}$ for defining the projection of the support on the concerned set of variables, as well as the corresponding matrix Ω_t. The major drawback of this process is that it comes to neglect the correlation existing between the variables that are not involved in a same constraint.

6.2.3 Numerical studies

In this section, we consider the uncertain optimization problem $\min_{x \in F} c^T x : \tilde{\xi} A^t \tilde{x} \leq 0,\ t = 1, ..., T$ where x is the command vector and ξ a random vector of probability distribution P, expected value μ and covariance Σ. We experiment the following approaches to deal with the problem the that goes hand in hand with a level of knowledge :

1. the *mean optimization*, which exploits only the mean of ξ;
2. the *worst-case optimization*, which exploits only the support of ξ;
3. the *robust approach*, which exploits the mean and the support of ξ;
4. the *distributionnally robust approach* or *DR approach*, which exploits the mean, the support and the covariance of ξ;
5. the *stochastic approach* where the probability distribution of ξ is available.

The first two approaches are deterministic approximations of the problem that gives rise to linear programs. They are detailed in the paragraph 6.2.3.1. The last three approaches consider that the constraints must be satisfied up to a given level of probability. Approaches 3 and 5 are described in Paragraph 6.2.3.2 and 6.2.3.6 respectively, whereas the fourth approach constitutes the main subject of this paper and is discussed in Section 6.2.2.

All our numerical experiments were performed on a 2.5 GHz Intel x86 with 16 GB memory. The solvers used are CSDP 6.1.0 for SDP, SeDuMi 1.3 for SOCP and CPLEX 12.1 for LP.

6.2.3.1 Mean and worst-case approaches

The mean optimization consists of solving the uncertain optimization problem by replacing the uncertain parameter ξ by its expected value μ, which gives rise to the following Linear Program $\min c^T x_{x \in F} : \tilde{\mu}^T A^t \tilde{\xi} \geq 0,\ t = 1, ..., T$.

The worst-case optimization follows the same principle except that the worst-case value of ξ is used instead of its expected value. This value is generally difficult to determine but in the particular problem that we consider (the supply/demand equilibrium problem), it is a trivial task.

6.2.3.2 Robust approach

In this paper, we compare the distributionnally robust approach with the method proposed in [68, 269] to handle a chance-constraint for which only the support and the expected value of the uncertain data are known. In a slight abuse of language, we call *robust* such an approach, even if this term refers generally to the case where only the support is available.

In this section, we provide a short description of this approach. We saw in paragraph 6.2.1.1 that the Boole's inequality can be used to safely approximate a joint chance-constraint into a set of individual chance-constraints. This approximation can be combined to the Hoeffding's inequality to provide a conservative approximation of the indivdual chance-constraints, as soon as the following hypothesis are satisfied :

- the support of ξ is a closed box : $\mathcal{S} = \{\xi \in \mathbb{R}^m : a_i \leq \xi_i \leq b_i\}$;
- the component of the function g are affine w.r.t x and ξ : $g_t(x, \xi) = \tilde{x} A^t \tilde{\xi}$;
- the components of ξ involved in a same sub-constraint are independent one to another;
- the expected value of ξ, denoted μ, is known.

In contrast to the distributionnally robust approach, the knowledge of the covariance of ξ is not required.

By applying the Boole's inequality, we obtain the following individual chance-constraints :

$$\mathrm{P}[\tilde{x}^T A^t \tilde{\xi} \geq 0] \leq \varepsilon/T,\ t = 1, ..., T$$

In the sequel, we consider any one of these constraints and drop the superscript t for sake of clarity. Based on the principle described in [68] and applied in [269], the Hoeffding's inequality is used to derive a conservative approximation of this constraint in the form of a second-order conic constraint. The Hoeffding's theorem is as follows :

Theorem 6.2.15 *Let consider a sequence of m independent real random variables X_i supported on the intervals $[\underline{X}_i, \overline{X}_i]$ and let $S = \sum_{i=1}^m X_i$. Then for any real $\tau \geq 0$,*

$$\mathrm{P}[S \geq \mathrm{E}(S) + \tau] \leq \exp(\frac{-2\tau^2}{\left\|\overline{X} - \underline{X}\right\|^2})$$

As a consequence, $\mathrm{P}[S \leq \mathrm{E}(S)+\tau] \geq 1-\exp(-2\tau^2/\left\|\overline{X} - \underline{X}\right\|^2)$ and therefore $\exp(-2\tau^2/\left\|\overline{X} - \underline{X}\right\|^2) \leq \varepsilon/T$ is a conservative approximation of $\mathrm{P}[S \leq \mathrm{E}(S) + \tau] \geq 1 - \varepsilon/T$.

$$\begin{aligned}
\exp(-2\tau^2/\left\|\overline{X} - \underline{X}\right\|^2) \leq \varepsilon/T \quad &\Leftrightarrow -2\tau^2 \leq \ln(\varepsilon/T)\ \left\|\overline{X} - \underline{X}\right\|^2 \\
&\Leftrightarrow \left\|\overline{X} - \underline{X}\right\|^2 \leq \delta^2\tau^2 && \text{with } \delta = \sqrt{-1/2\ln(\varepsilon/T)} \\
&\Leftrightarrow \left\|\overline{X} - \underline{X}\right\| \leq \delta\tau && \text{since } \tau \geq 0
\end{aligned}$$

It is possible to apply this approximation to $\tilde{x}^T A\tilde{\xi} \leq 0$ since the variables $\tilde{\xi}_i$ used within a same subconstraint are supposed to be independent of one another. To this end, we isolate the term that does not depend on ξ : $\tilde{x}^T A\tilde{\xi} = S + h(x)$ with $h(x) = A_0^T\tilde{x}$ and $S = \tilde{x}^T A_1 \xi$, with $A = \begin{pmatrix} A_0 & A_1 \end{pmatrix}$, $A_0 \in \mathbb{R}^{n+1,1}$ and $A_1 \in \mathbb{R}^{n+1,m}$. Then,

$$\begin{aligned}
\tilde{x}^T A\tilde{\xi} \leq 0 \quad &\Leftrightarrow S \leq -h(x) \\
&\Leftrightarrow S \leq \tau + \mathrm{E}(S) \text{ with } \tau = -\mathrm{E}(S) - h(x) = -(\tilde{x}^T A_1\mu + A_0^T\tilde{x}) = -\tilde{x}^T A\tilde{\mu}
\end{aligned}$$

This choice implicitly implies that the constraint $\tilde{x}^T A\tilde{\mu} \leq 0$ holds. Otherwise, the only possible lower bound of $\mathrm{P}[\tilde{x}^T A\tilde{\mu} \leq 0]$ would be zero since $\mathrm{P}[\tilde{x}^T A\tilde{\xi} \leq 0] = 0$ by taking for P the Dirac distribution of value μ.

Then, it suffices to express $\left\|\overline{X} - \underline{X}\right\|$ as a function of x. With $[a_i,\ b_i]$ the support of ξ_i, $\left\|\overline{X} - \underline{X}\right\| = \left\|\tilde{x}^T A_1(b-a)\right\|$ and the obtained second-order constraint is therefore :

$$\left\|\tilde{x}^T A_1(b-a)\right\| \leq -\delta\ \ \tilde{x}^T A\tilde{\mu}$$

Thus, we obtain a conservative approximation of the problem in the form of a SOCP. In the rest of the paper, this approach is referred to as the robust approach.

6.2.3.3 Comparison of the bounds obtained without command variables

In this paragraph, we put aside the dependence to the command variable x by considering the case where $n = 0$. Then, by denoting by d_t the first row of the matrix A^t, we are interested in the following moment problem :

$$\left\{\begin{array}{ll} \min & \mathrm{P}[d_t^T\tilde{\xi} \leq 0,\ t = 1, ..., T] \\ \text{s.t.} & \Omega_{\mathrm{P}}(\xi) = \Omega \\ & \mathrm{P} \in \mathcal{M}(\mathcal{S}) \end{array}\right.$$

where $\mathcal{S} = \{\xi \in \mathbb{R}^m : a_i \leq \xi_i \leq b_i,\ i = 1, ..., m\}$.

With $m = 1$ and $T = 1$

We start by studying the most simple class of instances of this problem, i.e., $m = 1$ and $T = 1$. For sake of simplicity and without loss of generality, we pick $d = \begin{pmatrix} 0 & 1 \end{pmatrix}^T$. Thus, the probability at hand is $\mathrm{P}[\xi \leq 0]$ and we aim at determining a lower bound of $\min_{\mathrm{P} \in \mathcal{P}} \mathrm{P}[\xi \leq 0]$, i.e., B such that $B \leq \min_{\mathrm{P} \in \mathcal{P}} \mathrm{P}[\xi \leq 0]$, or equivalently, $B \leq \mathrm{P}[\xi \leq 0]$, $\forall \mathrm{P} \in \mathcal{P}$. Thus, we aim at obtaining the largest bound possible.

We pick $a = -1$ everywhere. The other coefficients b, μ and Σ vary as indicated in Table 6.5. We restrict ourselves to the following values so the instance be feasible :

- $\mu \in [a, b]$ since the expected value necessarily lies in the support ;
- $\mu \leq 0$ since $\mathrm{E}_{\mathrm{P}}(g(x, \xi)) \leq 0$ is required ;
- $\Sigma \in [0, \Sigma^{\max}]$ with $\Sigma^{\max} = -(\mu - a)(\mu - b)$ so that $W^s \bullet \Omega \geq 0$.

Regarding the covariance Σ, five possible values are computed, corresponding to $20\%, 40\%, 60\%$ and 80% of $\Sigma^{\max}$. For each data set, we report 7 values :

- in the column labelled *Robust*, the bound $1 - \exp(-2\mu^2)(\|b - a\|^2)$, following the the robust approach described in paragraph 6.2.3.2;
- in the column labelled *Markov*, the bound computed via the Markov's inequality (μ/a), that corresponds to the DR approach without considering the variance (see Paragraph 6.2.2.2), in order to make the comparison with the robust approach that neither consider the variance ;
- in the columns labelled *Dist. Robust*, the bound computed via the DR approach with consideration of the variance. Each of these five columns corresponds to a different value of Σ : the label $p\%$ means that $\Sigma = p\%\Sigma^{\max}$.

		Without Σ		Dist. Robust with Σ			
μ	b	Robust	Markov	20%	40%	60%	80%
-0.9	0.50	0.513	0.9	0.967	0.935	0.906	0.915
-0.9	1.00	0.333	0.9	0.955	0.914	0.912	0.931
-0.9	1.50	0.228	0.9	0.944	0.902	0.922	0.941
-0.9	2.00	0.165	0.9	0.933	0.909	0.928	0.947
-0.7	0.50	0.353	0.7	0.872	0.773	0.704	0.752
-0.7	1.00	0.217	0.7	0.828	0.706	0.748	0.799
-0.7	1.50	0.145	0.7	0.788	0.722	0.774	0.827
-0.7	2.00	0.103	0.7	0.752	0.738	0.792	0.846
-0.5	0.50	0.199	0.5	0.714	0.556	0.533	0.600
-0.5	1.00	0.118	0.5	0.625	0.525	0.600	0.675
-0.5	1.50	0.077	0.5	0.556	0.560	0.640	0.720
-0.5	2.00	0.054	0.5	0.500	0.583	0.667	0.750
-0.3	0.50	0.077	0.3	0.446	0.309	0.384	0.459
-0.3	1.00	0.044	0.3	0.331	0.377	0.468	0.559
-0.3	1.50	0.028	0.3	0.317	0.418	0.518	0.619
-0.3	2.00	0.020	0.3	0.337	0.445	0.552	0.659
-0.1	0.50	0.009	0.1	0.112	0.184	0.256	0.328
-0.1	1.00	0.005	0.1	0.154	0.253	0.352	0.451
-0.1	1.50	0.003	0.1	0.179	0.294	0.410	0.525
-0.1	2.00	0.002	0.1	0.196	0.322	0.448	0.574

Table 6.5: Different lower bounds of $\min_{\mathrm{P} \in \mathcal{P}} \mathrm{P}[\xi \leq 0]$

The Figure 6.1 presents the robust and Markov bounds as well as the best (*Min var*) and worst (*Max var*) bounds obtained by considering different values of covariance.

These curves suggest several remarks. First, the robust bound is very conservative and depends to a large extent of the value of b. Not surprinsingly, the smallest b is, the better it is. Even for the smallest possible value of b, i.e., $b = 0$, the Markov's bound is better. Indeed, by noting $x = \mu/a$, the difference between these bounds is $x - (1 - exp(-2x^2))$, which is nonnegative for $x \in [0, 1]$ as illustrated on Figure 6.2.

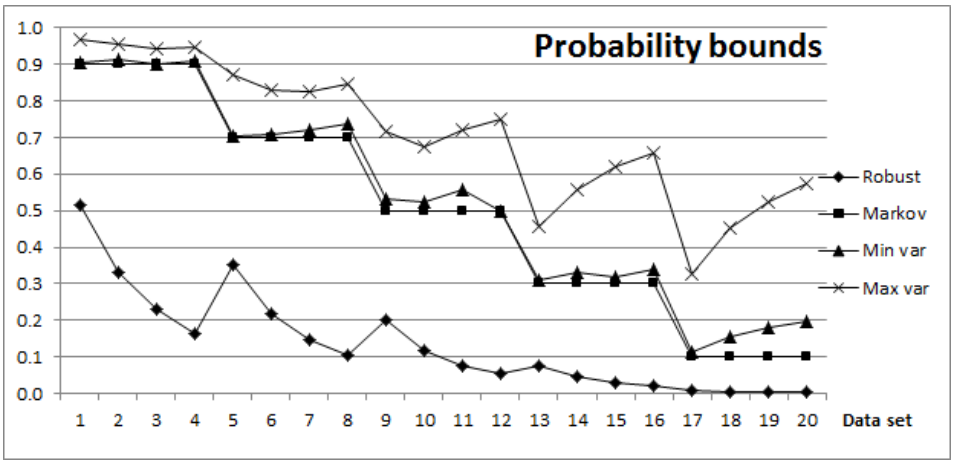

Figure 6.1: Comparison of different lower bounds of $\min_{\mathrm{P}\in\mathcal{P}} \mathrm{P}[\xi \leq 0]$

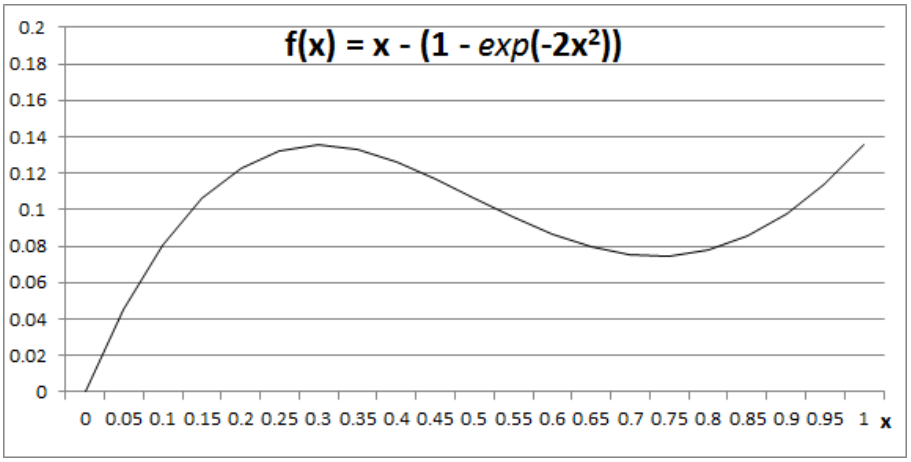

Figure 6.2: Variation of the robust and Markov bounds w.r.t. $x = \mu/a$ for $b = 0$

Finally and not surprisingly, the knowledge of the covariance considerably improves the DR bound, and not knowing the covariance is equivalent to the worst-case of all the possible covariance.

With $m \geq 1$ and $T = 1$
We are interested in the study of the probability $\mathrm{P}[e^T\xi \leq 0]$ with $m \geq 1$. The value of a_i, b_i and μ_i are drawn uniformly at random, in such a way that $\mu_i \in [a_i, b_i]$, $0 \in [e^T a, e^T b]$ and $e^T\mu \leq 0$. For each value of m, 100 instances are generated and we provide in the Table 6.6 the mean of the different bounds, i.e. :

- The robust bound, equal to $1 - \exp(-2(e^T\mu)^2/\left\|b-a\right\|^2)$ in column 2;
- The Markov bound, equal to $e^T\mu/e^T a$ in column 3, obtained by considering $e^T\xi$ as one random variable of expected value $e^T\mu$ and minimal value $e^T a$;
- The DR bound obtained without considering covariance in column 4;
- The DR bound obtained by considering different values of covariance, in the columns labelled by "With Σ".

By coherence with the robust approach, we assume that the variable ξ are independent, and therefore the covariance matrix Σ is diagonal. As previously, the diagonal values are chosen to be

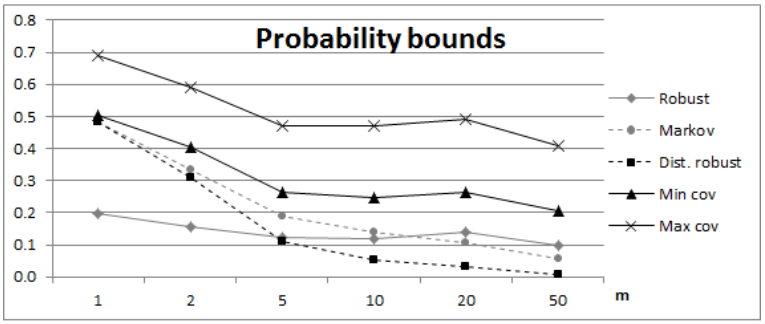

Figure 6.3: Comparison of the obtained lower bounds of $\min_{\mathrm{P}\in\mathcal{P}} \mathrm{P}[e^T\xi \leq 0]$ for $m \geq 1$

$10\%, 30\%, 50\%, 70\%$ and 90% of $\Sigma_i^{\max} = -(\mu_i - a_i)(\mu_i - b_i)$. Thus, the label "$p\%$" means that the covariance is taken as $\Sigma = p\%\Sigma^{\max}$.

	Without Σ			Dist. Robust with Σ			
m	Markov	Hoeffding	Dist. Robust	20%	40%	60%	80%
1	0.481	0.197	0.481	0.618	0.555	0.569	0.616
2	0.336	0.157	0.311	0.571	0.464	0.434	0.451
5	0.189	0.121	0.109	0.468	0.358	0.299	0.268
10	0.138	0.117	0.053	0.470	0.352	0.287	0.246
20	0.105	0.138	0.031	0.492	0.373	0.307	0.264
50	0.055	0.097	0.008	0.409	0.301	0.244	0.206

Table 6.6: Different lower bounds of $\min_{\mathrm{P}\in\mathcal{P}} \mathrm{P}[e^T\xi \leq 0]$

The Figure 6.3 represents the three obtained bounds (Robust, Markov and DR) obtained without covariance, as well as the best (*Min cov*) and worst (*Max cov*) DR bounds obtained with the covariance.

Clearly, exploiting the covariance within the DR approach yields the best bounds. Without considering the covariance, the Hoeffding's bound becomes the best one as the number of involved random variables increases. This comes from the fact that Hoeffding's inequality relies on the assumption that the variables are independent, which is not the case for the two other bounds. Markov's bound and DR bounds are often similar, but the DR is always better, since it exploits all the available information, whereas Markov exploits only $e^T\mu$ and e^Ta.

Remark 6.2.16 *The Markov's bound may appear as very efficient and easy to compute but the difficulty arises when trying to optimise this bound, since the function is not convex whenever a depends linearly of a command variable x.*

With $m > 1$ and $T > 1$

Finally, we compare the bounds for a joint chance-constraint with $T = 2$ to 10. Here again, the values of a_i, b_i and μ_i are drawn randomly for $m = 20$. We also draw the vectors d_t and bound the probability $\mathrm{P}[d_t^T\tilde{\xi} \leq 0,\ t = 1, ..., T]$ on 100 instances for each size of problems.

To compute the Hoeffding's bound, we sum the bounds obtained for each individual chance-constraint, according to Boole's inequality :

$$\mathrm{P}[d_t^T\tilde{\xi} \leq 0,\ t = 1, ..., T] \geq \sum_{t=1}^{T} \mathrm{P}[d_t^T\tilde{\xi} \leq 0] + 1 - T \geq 1 - \sum_{t=1}^{T} \exp\left(\frac{-2(d_t^T\mu)^2}{\left\|d_t^T(b-a)\right\|^2}\right)$$

We report the results in the Table 6.7 and Figure 6.4 in the same way than in the Table 6.5 and Figure 6.1. On Figure 6.4, the Hoeffding's bound does not appear since it is negative. This illustrates

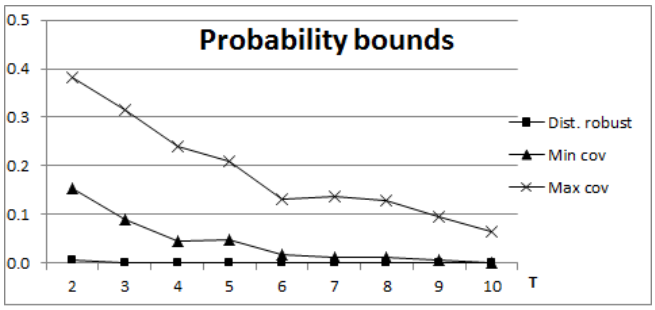

Figure 6.4: Comparison of the obtained lower bounds of $\min_{\mathrm{P}\in\mathcal{P}} \mathrm{P}[d_t^T\tilde{\xi} \leq 0,\ t = 1, ..., T]$ for different values of T

the fact that using the Boole's inequality highly reduces the tightness of the Hoeffding's bound. Finally, in this context, the DR bound with covariance is the only method that yields workable outcomes.

	Without Σ		Dist. Robust with Σ			
T	Robust	Dist. Robust	20%	40%	60%	80%
2	-0.616	0.006	0.381	0.259	0.194	0.153
3	-1.392	0.001	0.315	0.188	0.125	0.089
4	-2.135	0.000	0.241	0.126	0.074	0.047
5	-2.886	0.001	0.211	0.112	0.070	0.048
6	-3.737	0.001	0.131	0.053	0.028	0.018
7	-4.446	0.000	0.136	0.055	0.026	0.013
8	-5.058	0.000	0.130	0.051	0.024	0.013
9	-5.927	0.000	0.097	0.029	0.012	0.006
10	-6.987	0.000	0.064	0.015	0.005	0.002

Table 6.7: Different lower bounds of $\min_{\mathrm{P}\in\mathcal{P}} \mathrm{P}[d_t^T\tilde{\xi} \leq 0,\ t = 1, ..., T]$

6.2.3.4 The problem of supply/demand equilibrium under uncertainty

The problem addressed hereafter is taken from electrical industry and is a sub-problem of the Unit Commitment Problem (UCP) which aim at minimizing a global production cost while satisfying offer-demand equilibrium and operational constraints of a mix of power generation units (hydraulic valleys, nuclear plants and classical thermal units - coal, fuel and gas-) on a discrete time horizon.

More specifically, we consider a time horizon of T time steps and a system of N production units, characterized by a deterministic time-dependent production cost $c_{t,i}$ for the plant i at time step t. The essence of the problem is to determine the production of the plants i at each time step t : $x_{t,i} \in [0, 1]$, in order to meet the uncertain demand $D_{0,t}$ at each time step. The power plants are subject to random failure and their availability $D_{i,t}$ is therefore uncertain. This random vector results from the combination of many complex phenomena, such as climate conditions, consumer behaviour or unit failures and its distribution is therefore very difficult to determine precisely. However it is possible to estimate its expected values, covariance matrix and support $\mathcal{S}$, which is assumed to be a closed box, from historical data.

Some technical constraints state that the prescribed production of a plant i over the time-horizon shall not exceed a given amount r_i. More precisely, these constraints stand for the necessity of shutting down the plants to proceed to maintenance operations, and is therefore independent of the uncertain

availability of the plants. These requirements are summarized in the following formulation :

$$\begin{cases} \min & c^T x \\ \text{s.t.} & \mathrm{P}\left[D_{0,t} - \sum_{i=1}^{N} D_{i,t} x_{t,i} \leq 0,\ t = 1, ..., T\right] \geq 1 - \varepsilon \\ & \sum_{t=1}^{T} x_{t,i} \leq r_i,\ i = 1, ..., N \\ & x_{t,i} \in [0,1],\ i = 1, ..., N,\ t = 1, ..., T \end{cases}$$

This problem falls within the considered scope (6.3) with $m = T(N+1)$, $n = TN$ and $F = \{x \in [0,1]^n : \sum_{t=1}^{T} x_{t,i} \leq r_i,\ i = 1, ..., N\}$, since the functions involved in the probability are affine. The random vector ξ contains $D_{i,t}$, for $i = 0, ..., N$ and $t = 1, ..., T$. Each component of g represent the supply/demand equilibrium at one time step and therefore the assumption that the components of ξ involved in a same sub-constraint are independent one to another corresponds to the independence of $D_{i,t}$ and $D_{j,t}$ for any $i \neq j$. However, there may be some correlation between the components involved in different sub-constraints. From a modelling point of view, this is justified by the fact that the availability of the power plants is independent of the demand and from the availability of the other means of production, but there is a strong correlation between these values over time.

In the above numerical experiments, the considered park is composed of $N = 18$ power plants. The support, mean and covariance of the random variables δ_t and $D_{i,t}$ are deduced from a set of 100 historical observations.

6.2.3.5 Numerical results for the problem of supply/demand equilibrium

Applying the distributionnally robust approach to the supply/demand equilibrium problem leads to the results reported in Table 6.2.3.5. More precisely, for a varying number of time steps T, we compare :

- p^*_m, the optimal value of the LP obtained by the mean approach (see Paragraph 6.2.3.1) ;
- p^*_{dr}, the optimal value of the SDP obtained using the distributionnally robust paradigm (see Paragraph 6.2.2.4);
- p^*_{dri}, the optimal value of the SDP obtained using the distributionnally robust paradigm with converting the joint-chance into T individual chance-constraint (see Paragraph 6.2.2.4) ;
- p^*_r, the optimal value of the SOCP obtained using applying the Hoeffding's inequality (see Paragraph 6.2.3.2);
- p^*_w, the optimal value of the LP obtained by the worst-case approach (see Paragraph 6.2.3.1).

Finally, p^*_{dr}, p^*_{dri} and p^*_r are computed for three different values of ε : 0.8, 0.5 and 0.1.

We observe that the computation failures are always due to a non convergence of the solver. Clearly, the occurrence of this phenomenon is related to the size of the problems. The latter are reported in Table 6.2.3.5 in terms of number of variables (*# var*) and constraints (*# cst*). This table also shows the computation time in seconds in the columns labeled *time*. When the problem was solved for several values of ε, the reported value corresponds to the largest computation time.

The columns *SOCP*, *SDP* and *SDP-indiv* corresponds the computation of p^*_r, p^*_{dr} and p^*_{dri} respectively. The columns *LP* corresponds to both the computation of p^*_m and p^*_w. Indeed the corresponding LP have the same size and the running time reported here corresponds to the largest of the two running time, even if in practice these two values are very close.

Figure 6.5 shows the variation of the ratios p^*_w/p^* for $p^* = p^*_m$ (*mean*), $p^* = p^*_{dr}$ (*Dist. Robust*), $p^* = p^*_{dri}$ (*Dist. Robust Indiv.*) and $p^* = p^*_r$ (*Robust*) w.r.t T for $\varepsilon = 0.8$.

The immediate conclusion that can be drawn from these results is that for the particular problem of offer/demand equilibrium, there is few loss in splitting the joint chance-constraint into several

T	p_m^*	$\varepsilon = 0.8$			$\varepsilon = 0.5$			$\varepsilon = 0.1$			p_w^*
		p_{dr}^*	p_{dri}^*	p_r^*	p_{dr}^*	p_{dri}^*	p_r^*	p_{dr}^*	p_{dri}^*	p_r^*	
1	241.9	251.8	251.8	274.4	261.3	261.3	297.2	293.4	293.4	338.9	296.1
2	471.6	514.6	519.3	609.2	536.7	538.6	639.8	590.6	590.6	715.1	594.7
3	716.2	809.3	815.2	967.5	846.5	848.6	1007.7	†	897.3	1113.9	902.2
4	948.3	1104.7	1111.4	1335.4	1160.3	1162.7	1387.2	†	1195.6	1529.1	1200.9
5	1174.1	1403.3	1715.3	1702.4	†	1762.4	1765.2	†	†	1940.8	1498.3
6	1397.6	1707.0	2016.7	2057.0	†	2040.6	2128.7	†	†	2332.4	1780.1
7	1617.2	2011.2	2314.6	2420.5	†	2325.5	2501.7	†	†	2734.6	2059.7
8	1835.4	†	2589.8	2796.9	†	2158.8	2888.5	†	†	3153.7	2347.2
9	2055.5	†	2873.2	3154.8	†	†	3254.6	†	†	3545.2	2618.1
10	2279.9	†	3138.7	3539.0	†	†	3649.0	†	†	3971.0	2900.8

† : the computation failed

Table 6.8: Solving the distributionnally robust supply/demand equilibrium

T	LP			SOCP			SDP			SDP-indiv		
	# var	# cst	time	# var	# cst	time	# var	# cst	time	# var	# cst	time
1	18	37	0.001	74	38	0.30	267	115	0.58	267	115	0.58
2	36	92	0.001	204	114	0.44	931	322	14.11	534	248	2.78
3	54	129	0.001	354	228	0.36	1994	587	98.63	801	363	8.73
4	72	166	0.001	542	380	0.27	3456	928	550.63	1068	478	17.42
5	90	203	0.001	768	570	0.48	5317	1345	1713.18	1335	593	297.763
6	108	240	0.001	1032	798	0.31	7577	1838	4179	1602	708	49.89
7	126	277	0.002	1334	1064	0.42	10236	2407	16312	1869	823	79.25
8	144	314	0.002	1674	1368	0.51	13294	3052	†	2136	938	107.6
9	162	351	0.001	2052	1710	0.62	16751	3773	†	2403	1053	174.13
10	180	388	0.002	2468	2090	1.72	20607	4570	†	2670	1168	221.14

† : the computation failed

Table 6.9: Size of the obtained problem and computational time

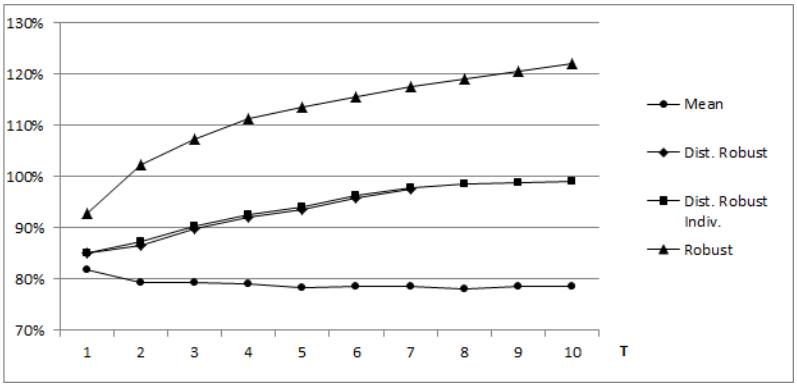

Figure 6.5: Comparison of the ratio p_w^*/p^* for $\varepsilon = 0.8$

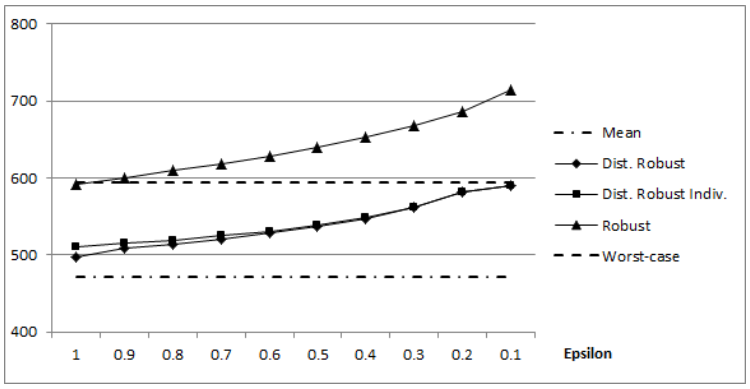

Figure 6.6: Variation of p^* w.r.t. ε for $T = 2$

individual ones. Consequently, the relatively poor performance of the robust approach is not due to this approximation, but to the Hoeffding's inequality, which is overly conservative and that do not involve the covariance of the random variables. This illustrates the importance of considering at least $\mathrm{E}(\xi_i^2)$ for obtaining a reasonably conservative approximation.

Figure 6.6 illustrates the variation of p^*_{dr} (*Dist. Robust*) , p^*_{dri} (*Dist. Robust Indiv.*) and p^*_r (*Robust*) w.r.t ε for $T = 2$.

Clearly, the robust bound is very bad, and becomes as worst as ε decreases. One possible explanation is that the robust bound, based on Hoeffding's inequality, does not exploit the knowledge of the interval $[a_i, b_i]$, but only the value of $b_i - a_i$, which means that it considers all the random variable with support $[\mu_i - \lambda(b_i - a_i), \mu_i + (1-\lambda)(b_i - a_i)]$ for all $\lambda \in [0, 1]$. In our data set, the mean is rather at the center of the interval ($\lambda = 0.42$ on average on the 10 time step) and this may explain why the Hoeffding's approach yields such poor results, even worse than the "worst-case" optimization.

6.2.3.6 Comparison with a stochastic approach

This section provides a comparison of the distributionnally robust approach with a stochastic approach. In both case, we neglect the dependency between the different terms of the joint chance-constraint, which comes to consider that all the random variables are independent from each other. By coherence, we use the same assumption in the distributionnally robust approach, which allows to split the joint chance constraint into several individual ones, as described in Paragraph 6.2.2.4.

Regarding the stochastic approach, with idea of the central limit theorem, we approximate the random variable $g_t(x, \xi) = \tilde{x} A^t \tilde{\xi}$ by a normally distributed variable. The mean and variance of this variable can be computed as a function of x :

$$\mathfrak{m}_t(x) = \tilde{x} A^t \tilde{\mu} \qquad \mathfrak{v}_t(x) = \sum_{i=1}^{m} (\tilde{x}^T A^t_{*,i})^2 \Sigma_{i,i}$$

By assuming that the rows are independent, we have the equivalent deterministic formulation:

$$\mathrm{P}\left[g_t(x,\xi) \leq 0,\ t = 1, ..., T\right] \geq 1 - \varepsilon \quad \Leftrightarrow \quad \mathrm{P}\left[g_t(x,\xi) \leq 0\right] \geq (1-\varepsilon)^{y_t}, \sum_{t=1}^{T} y_t = 1; y_t \geq 0$$

Then, for $\varepsilon \leq 0.5$, by using F the cumulative function of the Gaussian distribution, and F^{-1} its inverse, we have : $\mathrm{P}\left[g_t(x,\xi) \leq 0\right] = F(\mathfrak{m}_t(x)/\sqrt{\mathfrak{v}_t(x)})$ and therefore :

$$\mathrm{P}\left[g_t(x,\xi) \leq 0\right] \geq (1-\varepsilon)^{y_t} \Leftrightarrow \mathfrak{m}_t(x) \geq F^{-1}((1-\varepsilon)^{y_t}) \sqrt{\mathfrak{v}_t(x)}$$

T	$\varepsilon = 0.5$			$\varepsilon = 0.4$			$\varepsilon = 0.3$			$\varepsilon = 0.2$			$\varepsilon = 0.1$		
	p_s^*	p_{dri}^*	loss	p_s^*	p_{dri}^*	loss	p_s^*	p_{dri}^*	loss	p_s^*	p_{dri}^*	loss	p_s^*	p_{dri}^*	loss
1	263	261	-0.6%	267	266	-0.5%	272	271	-0.2%	278	279	0.7%	285	293	2.8%
2	513	539	5.0%	521	549	5.3%	531	563	6.0%	542	582	7.3%	559	591	5.7%
3	782	849	8.5%	795	867	9.1%	809	887	9.6%	826	895	8.3%	852	897	5.4%
4	1040	1163	11.8%	1057	1183	11.9%	1076	1191	10.7%	1100	1193	8.4%	1134	1196	5.4%
5	1291	1474	14.2%	1311	1484	13.2%	1336	1484	11.1%	1366	1489	9.0%	1410	1487	5.5%
6	1538	1762	14.6%	1563	-	-	1593	-	-	1629	-	-	1682	-	-
7	1780	2041	14.6%	1810	-	-	1846	-	-	1889	-	-	1951	-	-
8	2021	2326	15.1%	2057	-	-	2099	-	-	2150	-	-	2221	-	-
9	2264	-	-	2306	-	-	2354	-	-	2413	-	-	2495	-	-
10	2513	-	-	2560	-	-	2615	-	-	2681	-	-	2773	-	-

Table 6.10: Comparison with a stochastic approach

Finally, we apply the piecewise tangent approximation of $F^{-1}(\cdot)$ proposed in [73] and report the obtained results in Table 6.10. For each value of ε, we report p_s^* the optimal value obtained by the stochastic approach, as well as p_{dri}^* and the loss computed as $(p_{dri}^* - p_s^*)/p_s^*$.

We should make two important remarks about the stochastic approach. First, it does not make use of the support of the probability distribution. Furthermore, it does not require any assumption regarding the probability distribution of ξ. Finally, it is only an approximation, which can not said to be conservative, but which is more precise when the number of random variables involved in each sub-constraint is large.

It is interesting that for $T = 1$ the distributionnally robust approach yields a cheaper solution than the stochastic one. This can be explained by the fact that the stochastic approach do not take the support into account. This illustrates how tight is the conservative approximation made by the distributionnally robust approach in the particular case of a individual chance-constraint.

Not surprisingly, for $T > 1$, we observe that the stochastic approach is more effective than the distributionnally robust one. This is due to the additional assumption that we make in the stochastic approach, stating than $g_t(x, \xi)$ is a Gaussian variable. However, the result is not as affected as one might have thought, since the loss does not exceed 16%. Clearly, the loss increases depending on the number of time steps T. In particular, there is a huge difference between $T = 1$ and $T = 2$. This can be explained by the fact that for $T = 1$, i.e., for individual chance-constraints, the distributionnally robust approach is less conservative than for joint chance-constraints, as explained in Paragraph 6.2.2.4.

6.2.4 Conclusion

In this section, we investigate the distributionnally robust paradigm for addressing a joint chance-constraint that leads to a very elegant use of Semidefinite Programming. We apply this approach to a problem where the uncertain parameters are characterized by their mean, covariance and support, which is a box. This framework allows a comparison with the robust approach that uses the Hoeffding's inequality to establish a conservative approximation in the form of a SOCP.

Our main contributions consists of exhibiting the relationship between the distributionnally robust paradigm described in [270] and the SDP relaxation of the Generalized Problem of Moments designed by Lasserre in [171]. This allows a new interpretation of the SDP proposed in [270] and provides a new insight on the different levels of loss w.r.t. the original problem. We also proposed a simple way to exploit the sparsity of the constraint matrices in order to reduce the size of the obtained SDP.

Finally, numerical comparisons are presented on a supply/demand equilibrium problem. These results confirms that the distributionnally robust approach appears as a compromise between the mean and worst-case optimization. Furthermore, exploiting the covariance of the random variables gives to this method a significant advantages w.r.t the robust method that uses only the mean and the support. Finally we observe that the approximation exploiting the sparsity of the constraint matrices is accurate

for this problem, whereas it induces a sharp decrease of the size of the obtained SDP and of the computational time.

The distributionnally robust paradigm offers a new insight on optimization under uncertainty. Much remains to be done in this line of research, in particular we can think of using this approach to maximize a probability, instead of setting it into a constraint. This could be done for instance by means of a binary search on the desired level of probability. We also could think of a method for optimizing the coefficient α, in order to improve further the obtained bound.

6.3 Combining uncertainty and combinatorial aspects

This section contains the work presented in the paper [114]. We investigate SDP relaxations for mixed 0-1 Second-Order Cone Program, i.e. Second-Order Cone Program (SOCP) (see Paragraph 1.3.1) in which a specified subset of the variables are required to take on binary values.

The reasons for our interest in these problems lie in the fact that SOCP are famous for providing formulations or conservative approximations of robust Linear Programs, see for instance [32, 185, 269]. By a natural extension, MISOCP can be used to reformulate or approximate robust MILP.

MISOCP can be viewed as a combination between MILP and second-order cone programming. They just started to benefit from the great advances made in both areas. Thus, until recently, the only method for solving them was a basic Branch and Bound, i.e. a succession of continuous relaxation followed by rounding of the fractional solution. A first attempt to improve this algorithm was proposed by Çezik and Iyengar ([147]). In 2005, they extended Gomory cuts [111] and some other hierarchies of relaxations from Mixed-Integer Linear Programming to Mixed-Integer Cone Programming (SOCP and SDP). They also proposed linear valid inequality based on elements of the dual cone. This approach were promising but suffered from a lack of implementation, in particular, no instructions were given explaining how to pick up the most relevant inequalities among all the discussed ones.

In 2009, Drewes and Ulbrich [87] extended this work, with Lift & Project based linear and convex quadratic cuts and integrated it into a Branch and Cut. They also proposed a Branch and Bound based on a linear outer approximation of MISOCP. Another contribution on this topic was made by Atamturk and Narayanan ([16]) in 2010. By lifting the problem into a higher dimensional space, there generated some strong cutting planes and incorporated them within a Branch and Bound.

In this section, we propose an original approach for these problems, by exploiting the effectiveness of semidefinite relaxation for combinatorial optimization problems. Central to our approach is the reformulation of a MISOCP as a non convex quadratic program, where the non convexity stems both from the binary constraints and from the quadratic formulation of the second-order cone constraints. This brings us in the framework of binary quadratically constrained quadratic program (MIQCQP), which admits a relaxation as a semidefinite program. Actually, this relaxation, which is a generalization of the semidefinite relaxation for 0-1 linear program, has been extensively studied, see for instance [230] or [128] for a more detailed presentation.

The present work consists of defining such a semidefinite relaxation for MISOCP and in determining whether and how much it may improve the continuous relaxation. To the best of our knowledge, such a study has not been done so far. Our approach is depicted on the following diagram :

The initial problem to solve is the MISOCP at the root. On the right side is our contribution, with at first the reformulation of the MISOCP as a MIQCQP, followed by a relaxation as a SDP. Down right, the SDP relaxation is tightened by adding some constraint of the initial problem, which leads to problem (P_R). On the left side, we compute the continuous relaxation, which is a standard SOCP.

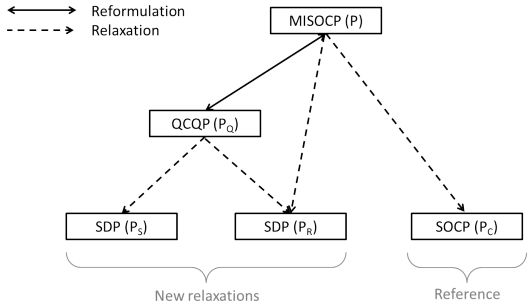

Figure 6.7: The different relaxation and reformulation of a MISOCP

6.3.1 Reformulation of a MISCOP as a MIQCQP

6.3.1.1 Definition of the Initial MISOCP

The problem we consider is a particular second-order cone program, where the objective function is an euclidean norm and a subset of the variables are binary :

$$(P)\begin{cases} \min_{x\in\mathbb{R}^n} & \|A_0x+b_0\| \\ \text{subject to} & \|A_ix+b_i\| \leq c_i^Tx+d_i,\ i=1,\cdots,m \\ & x_j\in\{0,1\},\ j=1,\cdots,r \end{cases} \tag{6.21}$$

where $r \leq n$ is the number of binary variables, $A_i \in \mathbb{R}^{m_i,n}$, $b_i \in \mathbb{R}^{m_i}$, $c_i^T \in \mathbb{R}^n$, $d_i \in \mathbb{R}$, for $i = 0,\cdots,m$. Except for the binary variables, this problem may be reduced to a standard second-order cone program, by linearizing the objective function as follows :

$$(P^1)\begin{cases} \min_{t\in\mathbb{R},\ x\in\mathbb{R}^n} & t \\ \text{subject to} & \|A_0x+b_0\| \leq t \\ & \|A_ix+b_i\| \leq c_i^Tx+d_i,\ i=1,\cdots,m \\ & x_j\in\{0,1\},\ j=1,\cdots,r \end{cases} \tag{6.22}$$

Adding a superscript to the name of a problem denotes a reformulation of this problem, whereas a subscript means that the problem is transformed. For instance, (P_C) denotes the continuous relaxation of (P). (P_C) is a standard SOCP that can be easily solved with a SOCP solver (see for instance see [248]) :

$$(P_C)\begin{cases} \min_{t\in\mathbb{R},\ x\in\mathbb{R}^n} & t \\ \text{subject to} & \|A_0x+b_0\| \leq t \\ & \|A_ix+b_i\| \leq c_i^Tx+d_i,\ i=1,\cdots,m \\ & 0\leq x_j\leq 1,\ j=1,\cdots,r \end{cases} \tag{6.23}$$

The choice of a norm as objective function has been made to fully exploit the potentiality of the semidefinite relaxation. Any linear objective function can be written under this form provided that a lower bound of its optimal value is known.

6.3.1.2 Formulation as a (Non Convex) Quadratically Constrained Quadratic Program

The MISOCP (P) presented above can be formulated as a non-convex QCQP. Specifically, given a second-order cone constraint : $\|Ax+b\| \leq c^Tx+d$ and squaring it, we obtain the following equivalence :

Proposition 6.3.1

$$\|Ax+b\| \leq c^T x + d \quad \Leftrightarrow \begin{cases} x^T(A^TA - cc^T)x + 2(b^TAx - dc^Tx) + b^Tb - d^2 \leq 0 \\ c^Tx + d \geq 0 \end{cases} \tag{6.24}$$

Proof 6.3.2 *It appears immediately that the first inequality implies $c^Tx+d \geq 0$. The inequality involves two non-negative values and can be lifted to the square.* □

Consequently, by noting that minimizing a positive quantity is equivalent to minimize its square, we have the following MIQCQP formulation for (P) :

$$(P_Q) \begin{cases} \min_{x\in\mathbb{R}^n} & x^TQ_0x + p_0^Tx + r_0 \\ \text{subject to} & x^TQ_ix + p_i^Tx + r_i \leq 0,\ i = 1,\cdots,m \\ & c_i^Tx + d_i \geq 0,\ i = 1,\cdots,m \\ & x_j \in \{0,1\},\ j = 1,\cdots,r \end{cases} \tag{6.25}$$

with

$$\begin{array}{l} Q_0 = A_0^TA_0,\ p_0 = 2A_0^Tb_0,\ r_0 = b_0^Tb_0 \\ Q_i = A_i^TA_i - c_ic_i^T,\ p_i = 2(A_i^Tb_i - d_ic_i),\ r_i = b_i^Tb_i - d_i^2, i = 1,\cdots,m \end{array} \tag{6.26}$$

This problem is generally not tractable by standard commercial solvers since it is not convex. However, in some particular cases described below, a resolution can be performed, which allows us to evaluate more accurately the quality of our relaxation :

- When all the variables are binary, it is therefore possible to reduce the problem to a linear one, by using for instance the well-known Fortet linearization;
- Or when all the matrices Q_i are positive semidefinite.

For a standard SOCP (without binary variable), the formulation (P_Q) is not worthwhile since it may induce a loss of convexity. Indeed, the matrices $Q_i = A_i^TA_i - c_ic_i^T$ may not be positive semidefinite. More precisely,

Proposition 6.3.3 *Let $A \in \mathbb{R}^{m,n}$ be a full rank matrix and $c \in \mathbb{R}^{n,1}$. The symmetric matrix $A^TA - cc^T$ is positive semidefinite if and only if there exists $u \in \mathbb{R}^{m,1}$, with $\|u\| \leq 1$, such that $c = A^Tu$.*

In order to prove this proposition, we need the following lemma :

Lemma 6.3.4 *Let $A \in \mathbb{R}^{m,n}$ and $B \in \mathbb{R}^{l,n}$ be two full rank matrices. Then we have the following equivalence :*

$$\mathcal{N}(A) \subset \mathcal{N}(B) \Leftrightarrow \exists M \in \mathbb{R}^{l,m} \text{ such that } B = MA \tag{6.27}$$

Proof 6.3.5 *If $B = MA$, then $Ax = 0 \Rightarrow Bx = 0$ so $\mathcal{N}(A) \subset \mathcal{N}(B)$. Conversely, according to the rank theorem, there exists a base of $\mathbb{R}^n$, $(e_1,\cdots,e_n)$, such that $(Ae_1,\cdots,Ae_k)$ is a base of the range-space of A, denoted $\mathcal{R}A$ and $(e_{k+1},\cdots,e_n)$ is a base of $\mathcal{N}(A)$. Then, according to the theorem of existence of linear application, there exists a unique linear application f such that :*

$$f(A(e_i)) = B(e_i),\ i = 1,\cdots k$$

For $i = k+1,\cdots,n$, $Ae_i = 0$ and $Be_i = 0$ since $\mathcal{N}(A) \subset \mathcal{N}(B)$. So, if M is the matrix associated with the application f, then $MA = B$. □

We are now in position to prove proposition 6.3.3.

Proof 6.3.6 *If $c = A^T u$ with $\|u\| \leq 1$, it is clear that $A^T A - cc^T$ is positive semidefinite. Indeed, for any $x \neq 0$:*

$$\begin{array}{ll} x^T(A^TA - cc^T)x & = x^T A^T Ax - x^T cc^T x \\ & = \|Ax\|^2 - \left\|c^T x\right\|^2 \\ & = \|Ax\|^2 - \left\|u^T Ax\right\|^2 \\ & \geq \|Ax\|^2 - \left\|u^T\right\| . \|Ax\|^2 \\ & \geq \|Ax\|^2 - \|Ax\|^2 = 0 \end{array} \tag{6.28}$$

Conversely, a necessary condition for $A^T A - cc^T \succcurlyeq 0$ is that Ker$A \subset$ Ker c^T. *Otherwise, if $x \in$* Ker$A, x \notin$ Ker c^T *: $x(A^TA - cc^T)x = -\left\|c^T x\right\|^2 \leq 0$.*

By applying the previous proposition, with $l = 1$, it comes that there is a vector $u \in \mathbb{R}^m$ such that $c^T = u^T A$. If $m \leq n$ there is $x_0 \in \mathbb{R}^n$ such that $u = Ax_0$. Then,

$$\begin{array}{ll} x_0(A^TA - cc^T)x_0 & = \|Ax_0\|^2 - \left\|u^T Ax_0\right\|^2 \\ & = \|Ax_0\|^2 - \|u\|^2 . \|Ax_0\|^2 \end{array} \tag{6.29}$$

So $\|u\| \leq 1$ is required for the positive semidefiniteness of the matrix. If $m > n$, we pick u_0 in the image of A such that $c = A^T u_0$. This means that the same conclusion holds. □

6.3.2 The Semidefinite Relaxation

We apply to the MIQCQP (P_Q) the standard semidefinite relaxation described at Paragraph 3.3.2 :

$$(P_Q^1) \left\{ \begin{array}{ll} \min_{x \in \mathbb{R}^n, X \in \mathbb{S}^n} & Q_0 \bullet X + p_0^T x + r_0 \\ \text{subject to} & Q_i \bullet X + p_i^T x + r_i \leq, \ i = 1, \cdots, m \\ & c_i^T x + d_i \geq 0, \ i = 1, \cdots, m \\ & \text{Diag}(X)_j = x_j, \ j = 1, \cdots, r \\ & X = xx^T \end{array} \right. \tag{6.30}$$

where $\mathbb{S}^n$ denotes the set of symmetric matrices of dimension n and Diag(X) stands for the vector made up with diagonal components of X.

The last constraint is non convex and captures all the difficulty of the problem. By relaxing it into the convex constraint $X - xx^T \succcurlyeq 0$, we obtain the semidefinite relaxation of the problem :

$$(P_S) \left\{ \begin{array}{ll} \min_{x \in \mathbb{R}^n, X \in \mathbb{S}^n} & Q_0 \bullet X + p_0^T x + r_0 \\ \text{subject to} & Q_i \bullet X + p_i^T x + r_i \leq 0, \ i = 1, \cdots, m \\ & c_i^T x + d_i \geq 0, \ i = 1, \cdots, m \\ & \text{Diag}(X)_j = x_j, \ j = 1, \cdots, r \\ & X \succcurlyeq xx^T \end{array} \right. \tag{6.31}$$

Under this notation, the continuous relaxation of (P) can be formulated as following :

$$(P_C^1) \left\{ \begin{array}{ll} \min_{x \in \mathbb{R}^n, X \in \mathbb{S}^n} & Q_0 \bullet X + p_0^T x + r_0 \\ \text{subject to} & Q_i \bullet X + p_i^T x + r_i \leq 0, \ i = 1, \cdots, m \\ & c_i^T x + d_i \geq 0, \ i = 1, \cdots, m \\ & 0 \leq x_j \leq 1, \ j = 1, \cdots, r \\ & X = xx^T \end{array} \right. \tag{6.32}$$

Subsequently, we discuss properties of the semidefinite relaxation and compare it to the continuous relaxation. First we prove that if all the matrices Q_i are positive semidefinite, then the semidefinite relaxation necessarily outperforms the continuous one. Then, we explain how one can extend this result to the general case, by reinforcing the semidefinite relaxation with constraints of the initial problem.

6.3.2.1 Semidefinite Case

If all the matrices Q_i are positive semidefinite, the semidefinite relaxation is necessarily better than the continuous one.

Proposition 6.3.7 *Let p_c^* and p_s^* denote the optimal values of (P_C) and (P_S) respectively. If all the matrices $Q_i,\ i = 1, \cdots, r$ are positive semidefinite, then $p_c^* \leq p_s^*$.*

Proof 6.3.8 *Let us consider a feasible solution (X_s, x_s) of (P_S). The constraint $X_s \succcurlyeq x_s x_s^T$ is equivalent, by using Schur complement, to $X' = \begin{pmatrix} X_s & x_s \\ x_s^T & 1 \end{pmatrix} \succcurlyeq 0$, which implies that any submatrices of X', and in particular $\begin{pmatrix} (x_s)_j & (x_s)_j \\ (x_s)_j & 1 \end{pmatrix}$ must be positive semidefinite. This is true if and only if $0 \leq (x_s)_j \leq 1$. Moreover, $Q_i \succcurlyeq 0$ and $X_s - x_s x_s^T \succcurlyeq 0$ implies that $Q_i \bullet (X_s - x_s x_s^T) \geq 0$. So $(x_s x_s^T, x_s)$ is a feasible solution of (P_C^1) and the associated objective value $Q_0 \bullet x_s x_s^T + p_0^T x_s + r_0$ is therefore bigger than p_c^*. Likewise we have $Q_0 \bullet (X_s - x_s x_s^T) \geq 0$, so*

$$p_c^* \leq Q_0 \bullet x_s x_s^T + p_0^T x_s + r_0 \leq Q_0 \bullet X_s + p_0^T x_s + r_0 = p_s^* \tag{6.33}$$

□

6.3.2.2 General Case

In the general case, the continuous relaxation (P_C) may be better than the semidefinite relaxation (P_S). In order to overcome this problem, we take benefit of the structure of the initial problem. We rely on the fact that a second-order cone constraint can be formulated as a semidefinite constraint, in the following way :

$$\left\|A_i x + b_i\right\| \leq c_i^T x + d_i \Leftrightarrow Y = \begin{bmatrix} (c_i^T x + d_i) I & A_i x + b_i \\ (A_i x + b_i)^T & c_i^T x + d_i \end{bmatrix} \succcurlyeq 0 \tag{6.34}$$

where I is the identity matrix of appropriate dimension. Adding these constraints to (P_S) guarantees the feasibility of $(x_s x_s^T, x_s)$ for (P_C^1), if (X_s, x_s) is a feasible solution of (P_S). Consequently, we can extend the proof presented in section 6.3.2.1 to the general case. The addition is necessary only if Q_i is not positive semidefinite, since the feasibility is already guaranteed otherwise. From a practical point of view, it is easier to formulate the constraints in their original form if the solver used for the resolution allows it, which is the case for us.

The problem thus obtained is denoted by (P_R). Subsequently, (P_S) and (P_R) are referred to as *basic* and *reinforced* semidefinite relaxation respectively.

6.3.3 Numerical Experiments

In this section, we report numerical results showing the validity of the relaxation we propose. For this, instances of MISOCP are randomly generated, according to the number of variables n. The number of constraint m is half the number of variables and the number of binary variables r is 0, $n/2$ or n. The coefficients of the elements A_i, b_i, c_i are drawn from uniform distribution within the interval $[-10.0, 10.0]$ and d_i is computed in order to ensure the existence of a feasible solution. More precisely, a binary solution x_0 is drawn and $d_i = \|A_i x0 + b_i\| - c_i^T x_0$. 20 instances are considered for each size of the problem.

Integer optimal solution are provided whenever CPLEX can solve MIQCQP formulation (P_Q) to optimality. In this case we use CPLEX 11.2 on an Intel x86 processor (1.99 GHz). Otherwise, only lower bounds obtained by both continuous relaxation and semidefinite relaxation (basic and reinforced) are given. These computations are performed with the software SeDuMi 1.3 (see [248]), on a Intel Core(TM) i7 processor (2.13 GHz).

6.3.3.1 General Case

The integer optimal solution of the problem (P) is generally not available and it is therefore impossible to compute the gap between optimal value and lower bounds obtained through relaxation. For this reason, we define the following indicators r_s and r_r as the relative difference between semidefinite bounds p_s or p_r (for the basic and reinforced relaxation respectively) and the continuous bound p_c :

$$r_s = \frac{p_s - p_c}{p_c} \qquad r_r = \frac{p_r - p_c}{p_c} \tag{6.35}$$

Then, semidefinite bounds become better as r_s or r_r increase. Furthermore, a positive value for these indicators means that the semidefinite bound is tighter than the continuous one. In table 6.11, we report the indicators r_s and r_r and the CPU time in seconds, for each resolution. Each result is the mean value computed on the 20 instances. The last line of the table contains the average of the previous lines. We also report the size of the considered instances in terms of number of variables and number of binary variables, in the columns 2 and 3.

Data set	Nb of var	Nb of bin. var.	CPU time (P_C)	CPU time (P_S)	CPU time (P_R)	r_s	r_r
P1	20	10	0.6	0.9	1.1	33.2%	45.6%
P2	40	20	1.0	1.9	3.0	17.5%	52.2%
P3	60	30	1.8	4.3	7.2	34.6%	58.5%
P4	80	40	3.1	8.4	12.2	36.5%	71.3%
P5	100	50	4.8	13.1	23.6	34.9%	71.9%
P6	150	75	9.8	41.1	63.3	45.3%	89.2%
P7	200	100	18.4	108.1	157.9	51.3%	81.3%
P8	250	125	32.2	242.3	351.4	77.3%	108%
P9	20	20	0.6	0.7	1.0	49.6%	76.1%
P10	40	40	1.1	1.7	2.4	70.2%	104.5%
P11	60	60	2.0	4.6	6.4	70.2%	106.9%
P12	80	80	3.0	7.5	11.6	90.9%	143.9%
P13	100	100	4.7	12.7	20.7	105.7%	148.8%
P14	150	150	9.7	39.2	54.6	126%	181.2%
P15	200	200	18.4	101.0	140.8	166.9%	210.3%
P16	250	250	32.1	216.1	318.0	205%	247.7%
Av.	113	84	9.0	50.2	73.4	75.9%	112.3%

Table 6.11: Comparison of the relaxations (P_C), (P_S) and (P_R)

We observe on this table that both semidefinite relaxations improve significantly the continuous one. The basic semidefinite relaxation improves the tightness of the bound of 75.9% on average w.r.t. the continuous relaxation. However, this result is negative for 4.7% of the instance, which means that the semidefinite relaxation is weaker than the continuous one.

This drawback can be overcome through the reinforcement of the semidefinite relaxation. In this way, the obtained bound is always tighter than the continuous one. Furthermore, on average, it achieves an improvement of about 112% of the continuous relaxation, for a running time that remains reasonable (73.4 s versus 9.0 s for the continuous relaxation and 50.2 s for the basic semidefinite relaxation). We observe that the difference between semidefinite and continuous relaxation increases as the size of the instances increases. Indeed, on average, the continuous relaxation is less tight on larger instances, so there is a larger possibility of improvement for the semidefinite relaxation.

6.3.3.2 Special Cases

In this section, we examine experimental results for some particular cases of the initial MISOCP where the integer optimal solution can be found. This allows us to measure precisely the loss of optimality due to the relaxations.

Semidefinite Case
We are interested in the special case where all the matrices Q_i are positive semidefinite. Then, (P_S) and (P_R) are equivalent.

Data set	Nb of var	Nb of bin. var.	CPU time (P)	CPU time (P_C)	CPU time (P_S)	r_s	g_c	g_s
P17	20	10	2.2	0.6	0.9	25.4%	15.3%	2.3%
P18	40	20	67.7	1.0	1.9	42.3%	24.6%	2.4%
P19	60	30	506.2	1.7	4.2	30.6%	15.4%	0.1%
P20	80	40	1863.5	3.0	8.9	35.2%	17.7%	0%
P21	100	50	2007.1	4.6	13.9	39.1%	18.2%	0%
P22	20	20	2.0	0.5	0.8	63.2%	58.2%	17.9%
P23	40	40	225.8	1.0	1.7	76.5%	77.9%	28%
P24	60	60	969.7	1.7	3.7	66.5%	48.9%	10.3%
P25	80	80	2037.0	3.0	8.0	63.8%	30.2%	0%
P26	100	100	†	4.6	12.5	71.9%	†	†
Av.	60	45	853.4	2.2	5.6	51.4%	34%	6.8%

Table 6.12: Comparison of the relaxations (P_C) and (P_S) in the semidefinite case

† : no integer solution found within the time limit

In Table 6.12, we report results obtained on such instances, generated by drawing, for each constraint i, a vector $u \in \mathbb{R}^{m_i}$ such that $||u|| \leq 1$. Then the associated vector c_i is computed as $A_i^T u$. By using the optimal value p^*, we compute the following gap :

$$g_c = \frac{p^* - p_c}{p_c} \qquad g_s = \frac{p^* - p_s}{p_s} \tag{6.36}$$

We observe that the semidefinite relaxation improves the continuous relation up to 51.4% on average. This improvement reaches 182% for some instances. Furthermore, the value of the semidefinite relaxation almost achieves the integral solution (gap less than 0.5%) for 72.8% of the instances for which the integer solution is available. On another hand, the instances with a still large semidefinite gap (more than 5%), are those with a very large continuous gap : 113% on average, versus 49% for the semidefinite gap.

Briefly, in instances with a reasonable continuous gap, the semidefinite relaxation provides a bound close to the integral solution. Otherwise, this gap is divided by more than two.

Fully Binary Case
In the case where all the variables are binary, the integer solution can be computed by CPLEX, which allows us to compute the gap. However, the computational time for solving such 0-1 linear program is extremely high, with little improvement of the solution over time. Therefore the running time of CPLEX was limited to 3600 s, and the gap computed with formula (6.36), is therefore an upper bound of the true gap. The instances for which an integer solution has been reached within the time limit are reported in Table 6.13.

Data set	Nb of bin. var.	CPU time (P)	g_c	g_s	g_r
P10	20	224.9	88.3%	55.3%	34.4%
P11	40	1667.6	105.9%	51.1%	39.6%
P12	60	2567.7	37.6%	0.6%	0.6%
P13	80	3335.3	30.6%	0.1%	0.1%

Table 6.13: Comparison of the gap of the relaxations (P_C), (P_S) and (P_R) in the full binary case

Thus, we see that the semidefinite relaxation gives almost the integer solution on largest instances (P12 and P13). More precisely, about a third of the concerned instances have a semidefinite gap less

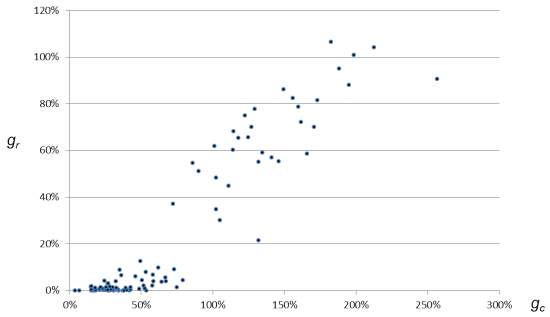

Figure 6.8: g_r as a function of g_c

than 0.5%. This is illustrated on figure 6.8 that represents the gap of the relaxation (P_R) as a function of the gap of the relaxation (P_C). One point represents one instance, either a fully binary one or a semidefinite one, as described in the previous subsection 6.3.3.2.

Thus, we observe that the semidefinite gap g_r is by and large very close to 0% whenever g_c is less than 50%. Furthermore, for a gap g_c less than 80%, the semidefinite gap remains small : less than 7%. Otherwise, we have improvements in the order of factor 2.

6.3.3.3 Unconstrained Case

When no constraints are imposed, the problem we addressed is the well-known least-squares estimation problem with binary variables. We aim at minimizing $||A_0x + b_0||$ where $(A_0)_i$ and $(b_0)_i$ can be interpreted as biased inputs and outputs of simulation i, $i = 1, \cdots, m_0$. To built the data set, a n-dimension binary vector x and a $5n$-dimension vector $\tilde{A}_0$ are drawn, and $\tilde{b}_0 = \tilde{A}_0x$ is computed. Then A_0 and b_0 are built as $\tilde{A}$ and $\tilde{b}$ perturbed by a Gaussian noise. The associated numerical results are reported in Table 6.14.

Data set	Nb of var	Nb of bin. var.	CPU time (P)	CPU time (P_C)	CPU time (P_S)	r_s	g_c	g_s
P27	20	10	0.0	0.4	0.3	62.9%	33.3%	1.4%
P28	40	20	0.0	0.5	0.6	65.8%	35.3%	1.8%
P29	60	30	0.4	0.6	0.8	64.9%	35.2%	2.1%
P30	80	40	2.6	1.0	1.3	63.8%	34.8%	2.2%
P31	100	50	36.2	1.5	2.0	64.4%	35.2%	2.2%
P32	20	20	0.0	0.4	0.3	99.4%	55.6%	3.9%
P33	40	40	0.2	0.5	0.6	94.8%	57%	6.4%
P34	60	60	1.7	0.6	0.9	98%	59.4%	7%
P35	80	80	20.0	1.0	1.4	88%	53.8%	6.8%
P36	100	100	176.7	1.5	2.1	93.7%	56%	7.1%
Av.	60	45	23.8	0.8	1.0	79.6%	45.6%	4.1%

Table 6.14: Comparison of the relaxations (P_C) and (P_S) in the unconstrained case

In this framework, the semidefinite relaxation is very efficient : whereas the continuous gap provides an average gap of 45.6%, the semidefinite relaxation is very close to the optimal solution, as the average gap reaches less than 5%. The high quality of these results can be explained by the fact that, when all the variables are binary, the quadratic formulation (P_Q) is an unconstrained binary quadratic problem, which is well-known (see for example [128, 175]) to be equivalent to the MAX-CUT problem up to an additive constant. For the MAX-CUT problem, it has been shown by Goemans and Williamson (cf [109]) that the gap provided by the SDP relaxation is guaranteed to reach at most 12.2%. In our context, this result can not be applied directly, because of the additive constant, but we still

observe that the maximal gap obtained is 10.24%. Finally, the running time for solving the semidefinite relaxation is very small, an average of 1.0 s. This is hardly larger than the running time of (P_C), i.e., 0.8 s on average. This is very small compared to the time required for finding the integral solution, that is 23.8 s on average.

6.3.4 Conclusion

This section introduces a semidefinite relaxation for MISOCP through a reformulation as a MIQCQP. When all the constraints of the MIQCQP are convex, this semidefinite relaxation is necessarily better than the continuous relaxation. This result can be extended to the general case by adding some constraints of the initial problem to the semidefinite relaxation. This approach provides lower bounds that are very satisfying. Firstly, on general instances, it improves significantly the continuous relaxation. For instances with a small continuous gap, it almost gives the integral solution. Otherwise, the improvement is of an order of magnitude 2. The results are even better for unconstrained instances that correspond to a least-square estimation. For these instances, the gap is on average divided by 10 w.r.t. the continuous relaxation. Finally, all these results are obtained within a reasonable amount of time.

In conclusion, the study of MISOCP is of great interest in the context of optimization under uncertainty since several uncertain optimization programs admits reformulation in the form of a SOCP, and by extension, MISOCP can be used to reformulate such programs with integer variables. It turns out that semidefinite programming offers an efficient framework for dealing with MISOCP. These promising results suggest interesting prospects for using semidefinite relaxation in a Branch & Bound procedure, in complementarity with existing works about generation of cutting planes for MIQCQP. One may also apply a rounding procedure to recover a feasible solution.

6.4 Conclusion

This chapter is devoted to the question of how SDP can be used to handle uncertainty in optimization problems. To this end, we conducted three studies, each one requiring a different knowledge of the random parameters.

The first section examines a stochastic paradigm where the probability distribution is discrete and takes the form of equiprobable scenarios taken from the historically observed realizations. In this way, the constraints that must be satisfied up to a given level of probability (chance-constraints) admit a deterministic formulation that involves an additional binary variable. Then, the whole problem becomes a large Mixed-Integer Quadratically Constrained Quadratic Program (MIQCQP). It is therefore possible to apply the SDP relaxation to this problem, which yields an average gap of 2.76%, to compare to 53.35%, the average gap of the linear relaxation. Combined to a randomized rounding approach, the SDP approach provide feasible solution with an average optimality gap of 2.20%, versus 10.38% for the linear relaxation.

In the second section, we investigated a more original paradigm for addressing uncertainty, namely the distributionnally robust paradigm. This approach offers two main advantages. First, a perfect knowledge of the probability distribution of the random data is not required. It suffices to know its support and its first moments and the optimization is made on the worst case w.r.t. all the probability distribution that share these characteristics. Second, the obtained problem can be reformulated, or at least conservatively approximated, by a SDP. This process is closely related to the use of SDP for the Generalized Problem of Moment (GPM). We compared this paradigm with a robust approach based on Hoeffding's inequality, that establishes a conservative approximation of the problem in the form of a SOCP, on the supply/demand equilibrium problem presented in paragraph 4.3.3.5. These results confirms that the distributionnally robust approach appears as a compromise between the mean and worst-case optimization. Furthermore, exploiting the covariance of the random variables gives to this method a significant advantages w.r.t the robust method that uses only the mean and the support.

Finally, in the third section, we turn our attention to MISOCP problems. These problems are very little studied whereas they are prevalent for addressing uncertainty since they emerge for instance as the robust counterpart of MILPs. A reformulation of these problems as QCQP enables to apply the standard SDP relaxation. In order to recover the convexity lost by this reformulation, it is interesting to convert certain SOCP constraint into SDP constraints in a manner that preserves convexity and add them to the SDP relaxation. Thus, we obtain very satisfying lower bounds compared to the continuous relaxation.

In conclusion, SDP is an elegant and powerful tool for addressing uncertainty, that is appropriate for various representations of the random data.

Conclusions and perspectives

As already mentioned, our objective in this thesis was to assess the interest of Semidefinite Programming for the problems of energy management. Two lines of research were identified : the first one aimed at exploiting a well-known strength of SDP : its ability to provide tight relaxations of combinatorial or quadratic programs. The second one was dedicated to study the potentiality of SDP for addressing uncertainty in optimization problems.

The first one is studied in chapter 5. This part of the work was well-defined since the use of SDP for deriving relaxations of QCQP has been extensively studied. Indeed, obtaining tight convex relaxations is a key issue in optimization since these relaxations are at the core of iterative methods for non-convex programming, such as Branch & Bound or Branch & Cut. Then the challenge was to :

- select an interesting problem to handle and model it in an appropriate fashion;
- among the extensive literature on SDP relaxations, select the best recipe to apply to get a tight SDP relaxations;
- apply this to instances that are both difficult to solve with the commercial solver CPLEX and tractable but the SDP solver ;
- get a step further w.r.t. existing approaches by proposing new theoretic results or ideas on this topic that has been covered in-depth before.

We briefly recall that for any QCQP there exists a systematic procedure for building a SDP relaxation, called *standard SDP relaxations* (see Paragraph 3.3.2). Generally, it is desirable to convert the original QCQP into a more complex one, by adding valid quadratic constraints, then to apply the standard SDP relaxation to this latter QCQP, in order to make the relaxation tighter. For instance, on a 0/1-LP, which can be viewed as a particular QCQP, the standard SDP relaxation yields the same bound than the linear relaxation. This illustrates that the combinatorial aspect is not sufficient for exhibiting interest in SDP relaxations. Quadratic features are also necessary and therefore, the key problem is QCQP. Then, we have two options :

- working on a 0/1-LP and converting it into an equivalent QCQP ;
- working directly on QCQP;

The second option is more interesting for us since very efficient solvers already exist for 0/1-LP, that are difficult to compete. For this reason, we opted for problems with native quadratic features. These problems also contains linear constraint, in particular the bounds constraints, and their combinations is a key element for generating valid quadratic constraints.

The determination of the most appropriate valid quadratic constraints is a well-studied problem, very close to the combinatorics, where one aims at determining the linear constraints that describe the convex hull of a polytope.

Inspired by all the works in this vein and by the fact that almost all the valid quadratic constraints proposed in the literature can be formulated as suitable combination of the initial quadratic constraints of the problem and of the pair-wise product of the linear constraints of the problem (including bounds constraints), we proposed a separation method, aiming at selecting the best one among all the constraints

that can be generated in this way. Thus, on the model 3 of the Nuclear Outages Scheduling Problem, we reduced by 25.15% the gap of the linear relaxation.

We also experimented several standard recipe for reinforcing the standard SDP relaxations. What emerges from this study is that the first thing to be done is to square the linear constraints. Then, adding some products of linear constraints enables to effectively tighten the relaxation, but the addition of all these constraints quickly renders the problem intractable. This study also provided the opportunity to compare three possibles models for a class of disjunctive constraint of the form $a^T x \notin]b, c[$, where x is a binary vector. Among these models, 2 are linear and one is quadratic. Clearly, one of the two linear models leads to relaxations that are largely tighter than with the other models, for all the considered SDP relaxations. We also consider LP relaxations, built from Reformulation-Linearization Technique and reinforced in the same fashion than the SDP relaxations.

An alternative way of exploiting the potential of SDP for combinatorial problems is to address MISOCP problems, that appear for instance when taking the robust counterpart of a MILP. The MISOCP can be converted into a QCQP at the cost of a loss of convexity. Then we apply the SDP relaxation to this QCQP, and we restore the lost convexity by converting the SOCP constraint into SDP ones. This approach delivers lower bounds that are very encouraging, since they consistently outperform the continuous relaxation in a reasonable amount of time.

A last work on this subject was proposed in Section 5.3.3. We implemented the Lasserre hierarchy on small instances of the Nuclear Outages Scheduling Problems. The obtained results are quite impressive : the integer solution is recovered at the rank 2 of the Lasserre's hierarchy on all the considered instances, that include both 0/1-LP and 0/1-QCQP.

Even if overall SDP provides tighter relaxations than LP, this technique faces numerous practical difficulties, that are not encountered with LP, due to the fact that SDP solvers are still in their infancy. In particular, we encountered the following problem :

- storage memory problems that make the SDP intractable, by and large for problems with primal matrices variables dimension and number of primal constraints of the order of 10^4;
- the computational time grows quickly and can amount to a number of hours for problems with primal matrices variables dimension and number of primal constraints of the order of 10^3;
- even all smaller instances, all the SDP solvers that we experimented (CSDP, DSDP, SeDuMi) encountered difficulties during the resolution, such as not attaining the full accuracy, a lack of progress along the iteration which prevents from providing the solution, or the returned solution which is not optimal;
- when an error occurs in the design of the SDP, it is very difficult to trace back its origin;
- there is no direct way of handling inequality primal constraints. To do so, we have to add slack variables, which increases the size of the problem;
- the choice of the best SDP solvers depends on the considered instance.

Thus, in spite of strong theoretical results and a significant work on resolution methods, the use of SDP for real-life combinatorial problems is not yet fully operational. However, in a bigger picture prospective, SDP may reveal another part of its potential for relaxation of QCQP with non convex terms that involves continuous variables. Indeed, on these problems, the resolution methods are not as advanced as for convex QCQP, or for QCQP with binary variables. Thus, the SDP relaxation, with all its possible reinforcement, is a real asset compared to the other available convex relaxations.

The second objective of the thesis deals with uncertainty, and more specifically, with the consideration of of chance-constraints. To this end, we first used a discrete representation of uncertainties with scenarios approximation which leads to obtaining a large mixed-integer Quadratically Constrained Quadratic Program. Then, we applied SDP relaxations on this problem and exploited the obtained bounds to build a good feasible solution by means of a randomized rounding procedure.

The second work touching on this topic concerns distributionnally robust optimization. This recent approach consists of the worst-case optimization on a set of probability distribution, defined

via their support and moment sequence. This approach, that can be seen as a compromise between robust and stochastic optimization, is very beneficial in our case. First, it is highly relevant from a modelling point of view, since it does not require to make assumption on a probability distribution that we generally ignore, but it exploits some available knowledge about the moment and support of the probability distribution. Second, SDP provides elegant approximation framework for these problems, that gives the optimal solution in certain cases.

In our study, we established a relationship between this approach, based on the paper of Zymler et al. [270] and the use of SDP for handling the Generalized Problem of Moments, mainly carried out by Calafiore and El Ghaoui [68], Bertsimas et al. [38, 40, 41] and Lasserre [171]. Furthermore, we applied this approach to an energy management problem, namely the supply-demand equilibrium problem. Finally, we compared the results to those obtained via robust and stochastic approaches.

This work opens perspectives for other applications in energy management and more generally, for other optimization issues at EDF R&D. First, we may think of applying the strength of SDP for combinatorial and quadratic optimization to certain subproblems of the hydro power management problems. Indeed, these problems have a strong combinatorial flavor, because of the discretization of operating power. Furthermore, these subproblems are part of an augmented Lagrangian decomposition, which make quadratic terms to appear in the objective function. These problems are therefore very good candidate for the SDP relaxation, even if their huge size will certainly pose a serious problem.

Besides energy management, there are other promising applications of SDP at EDF R&D. In particular, all the problems related to asset portfolio management from a financial point of view or to risk management on the energy market. For instance, it would be interesting to assess if the distributionnally robust paradigm is relevant for these problems.

One also may think of applying SDP to a current engineering problem encountered on maintenance of nuclear power systems that leads to a Binary Least-Square problem. We refer the reader to [268] for a detailed description of this problem. Very briefly, it concerns the non destructive evaluation of steam generator tubes of nuclear power plants, which appears within the framework of inverse problems. These problems aims at estimating a large number of input parameters of a model, linear or not, with a small amount of potentially noisy output data, which can not be observed directly but only indirectly by the computation of a deterministic model. Methods used for tackling these problems covers a wide field, from the classical method of least squares to Bayesian methods.

In the present case, the problem is cast into a Least-Square problem, where a part of the variable is required to be binary. Then the SDP relaxation could be applied to this both combinatorial and quadratic problem.

Finally, the application of SDP to MISOCP is very promising and there is work to be done in that direction. First, from an algorithmic point of view, it could be interesting to implement a MISOCP solvers based on the integration of the SDP relaxation into an enumerative method such as Branch & Bound. We could also think of applying this method for tackling real-life MISOCP, such as the robust counterpart of a 0/1-LP under uncertainty.

To conclude, this thesis was above all an applicative work, whose actual purpose is a solid groundwork and a practical experimentation on real-life problems of SDP, rather than new theoretical results. To this end, we endeavoured to build a very practical "user manual" for SDP, both for obtaining relaxations of NP-hard problem and for facing uncertainty. We also identified the key issues associated to these two axis in an operational perspective, based on a detailed study of the existing literature on these subjects.

With this work, we identified a bunch of applications that reveals the potential of SDP as a modelling tool. Despite practical difficulties mainly due to the fact that SDP is not a mature technology yet, it is nonetheless a very promising optimization method, that combines all the strengths of conic programming and offers great opportunities for innovation.

Bibliography

[1] *Xpress Optimization Suite*, 2012.

[2] W. Adams and H. Sherali. A tight linearization and an algorithm for 0-1 quadratic programming problems. *Management Science*, 32(10):1274–1290, 1986.

[3] P. Adasme, A. Lisser, and I. Soto. A quadratic semidefinite relaxation approach for OFDMA resource allocation. *Networks*, 59(1):3–12, 2012.

[4] I. Adler and F. Alizadeh. Primal-dual interior point algorithms for convex quadratically constrained and semidefinite optimization problem. Technical report, Rutger Center for Operations Research, 1994.

[5] A. A. Ahmadi, A. Olshevsky, P. A. Parrilo, and J. N. Tsitsiklis. Np-hardness of deciding convexity of quartic polynomials and related problems. *Mathematical Programming*, 2011.

[6] F. A. Al-Khayyal and J. E. Falk. Jointly constrained biconvex programming. *Mathematics of Operation Research*, 8(2):273–286, 1983.

[7] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5:13–51, 1993.

[8] F. Alizadeh and D. Goldfarb. Second-order cone programming. *Mathematical Programming*, 95(1):3–51, 2003.

[9] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton. Primal-dual interior-point methods for semidefinite programming: Convergence rates, stability and numerical results. *SIAM Journal on Optimization*, 5:13–51, 1994.

[10] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton. Complementarity and nondegeneracy in semidefinite programming. *Mathematical Programming*, 77(2):111–128, 1997.

[11] E. D. Andersen, C. Roos, and T. Terlaky. On implementing a primal-dual interior-point method for conic quadratic optimization. *Mathematical Programming*, 95(2):249–277, 2003.

[12] M. Anjos and J. Lasserre, editors. *Handbook of Semidefinite, Conic and Polynomial Optimization: Theory, Algorithms, Software and Applications*. International Series in Operational Research and Management Science, 2012.

[13] M. F. Anjos. An improved semidefinite programming relaxation for the satisfiability problem. Technical report, Department of Electrical & Computer Engineering, University of Waterloo, Canada, 2002.

[14] K. M. Anstreicher. Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. *Journal of Global Optimization*, 43(2–3):471–484, 2009.

[15] A. Arbel. *Exploring Interior-Point Linear Programming: Algorithms and Software*. Foundations of Computing Series. MIT Press, 1993.

[16] A. Atamtürk and V. Narayanan. Conic mixed-integer rounding cuts. 2010.

[17] C. Audet, P. Hansen, B. Jaumard, and G. Savard. A branch and cut algorithm for nonconvex quadratically constrained quadratic programming. *Mathematical Programming*, 87(1):131–152, 2000.

[18] E. Balas. *Annals of Discrete Mathematics 5 : Discrete Optimization*, chapter Disjunctive Programming, pages 3–51. Springer-Verlag, 1979.

[19] E. Balas. Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Applied Mathematics*, 89(1–3):3–44, 1998.

[20] E. Balas. Projection, lifting and extended formulation in integer and combinatorial optimization. *Annals of Operations Research*, 140:125–161, 2005.

[21] E. Balas, S. Ceria, and G. Cornuejols. A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Mathematical Programming*, 58:295–323, 1993.

[22] E. Balas and W. Pulleyblank. The perfectly matchable subgraph polytope of a bipartite graph. *Networks*, 13(4):495–516, 1983.

[23] X. Bao, N. V. Sahinidis, and M. Tawarmalani. Semidefinite relaxations for quadratically constrained quadratic programming: A review and comparisons. *Mathematical Programming*, 129(1):129–157, 2011.

[24] G. P. Barker and D. Carlson. Cones of diagonally dominant matrices. *Pacific J. Math.*, 57(1):15–32, 1975.

[25] A. Barvinok. Problems of distance geometry and convex properties of quadratic maps. *Discrete and Computational Geometry*, 13:189–202, 1995.

[26] A. Barvinok. A remark on the rank of positive semidefinite matrices subject to affine constraints. *Discrete and Computational Geometry*, 25:23–31, 2001.

[27] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1957.

[28] R. Bellman and K. Fan. On systems of linear inequalities in hermitian matrix variables. *Proceedings of the Symposium on Pure Mathematics*, 7, 1963.

[29] A. Ben-Israel, A. Charnes, and K. Kortanek. Duality and asymptotic solvability over cones. *Bulletin of American Mathematical Society*, 75(2):318–324, 1969.

[30] A. Ben-Tal, D. Bertsimas, and D. B. Brown. A soft robust model for optimization under ambiguity. *Operations research*, 58(2–2):1220–1234, 2010.

[31] A. Ben-Tal, L. El-Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics, 2009.

[32] A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1):1–13, 1999.

[33] A. Ben-Tal and A. Nemirovski. *Analysis, algorithms, and engineering applications*, chapter Lectures on modern convex optimization. Society for Industrial and Applied Mathematics (SIAM), 2001.

[34] S. J. Benson, Y. Ye, and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization*, 10(2):443–461, 2000.

[35] D. P. Bertsekas. *Constrained Optimization and Lagrange Multipliers*. Athena scientific, 1996. first published 1982.

[36] D. Bertsimas, D. B. Brown, , and C. Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.

[37] D. Bertsimas, X. V. Doan, K. Natarajan, and C.-P. Teo. Models for minimax stochastic linear optimization problems with risk aversion. *Mathematics of Operations Research*, 35(3):580–602, 2010.

[38] D. Bertsimas and J. Niño Mora. Optimization of multiclass queuing networks with changeover times via the achievable region approach: Part ii, the multi-station case. *Mathematics of Operations Research*, 24(2):331–361, 1999.

[39] D. Bertsimas and I. Popescu. Moment problems via semidefinite programming: Applications in probability and finance. 2000.

[40] D. Bertsimas and I. Popescu. On the relation between option and stock prices: A convex optimization approach. *Operations Research*, 50(2):358–374, 2002.

[41] D. Bertsimas and I. Popescu. Optimal inequalities in probability theory: a convex optimization approach. *SIAM Journal on Optimization*, 15(3):780–804, 2005.

[42] D. Bertsimas and M. Sim. Price of robustness. *Operations Research*, 52:35–53, 2004.

[43] D. Bertsimas and M. Sim. Tractable approximations to robust conic optimization problems. *Mathematical Programming Serie B*, 107:5–36, 2006.

[44] D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization*. Series in Optimization and Neural Computation. Athena Scientific, 1997.

[45] A. Billionnet and S. Elloumi. Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem. *Mathematical Programming*, 109:55–68, 2007.

[46] A. Billionnet, S. Elloumi, and A. Lambert. Extending the QCR method to general mixed integer programs. *Mathematical Programming*, 131:381–401, 2012.

[47] A. Billionnet, S. Elloumi, and M.-C. Plateau. Convex quadratic programming for exact solution of 0-1 quadratic programs. Technical report, CEDRIC laboratory, CNAM-Paris, France, 2005.

[48] A. Billionnet, S. Elloumi, and M.-C. Plateau. Quadratic 0-1 programming : tightening linear or quadratic convex reformulation by use of relaxations. *RAIRO*, 42(2):103–121, 2008.

[49] A. Billionnet, S. Elloumi, and M.-C. Plateau. Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: the QCR method. *Discrete Applied Mathematics*, 157:1185–1197, 2009.

[50] P. Biswas, T.-C. Liang, K.-C. Toh, T.-C. Wang, and Y. Ye. Semidefinite programming approaches for sensor network localization with noisy distance measurements. *IEEE Transactions on Automation Science and Engineering*, 3, 2006.

[51] P. Bonami. Lift-and-project cuts for mixed integer convex programs. *Lecture Notes in Computer Science*, 6655:52–64, 2011.

[52] P. Bonami, J. Lee, S. Leyffer, and A. Wachter. More branch-and-bound experiments in convex nonlinear integer programming. Preprint ANL/MCS-P1949-0911, Argonne National Laboratory, Mathematics and Computer Science Division.

[53] B. Borchers. CSDP, a C library for semidefinite programming. *Optimization Methods and Software*, 11, 1999.

[54] B. Borchers and J. Mitchell. An improved branch and bound algorithm for mixed integer nonlinear programming. *Computers and Operations Research*, 21:359–367, 1994.

[55] S. Boyd and A. d'Aspremont. Relaxations and randomized methods for nonconvex qcqps. Technical report, Stanford University, 2003.

[56] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan. Linear matrix inequalities in system and control theory. *SIAM*, 1994.

[57] S. Boyd and L. Vandenberghe. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.

[58] S. Boyd and L. Vandenberghe. Applications of semidefinite programming. *Applied Numerical Mathematics*, 29:283–299, 1999.

[59] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[60] F. Brouaye. *La modélisation des incertitudes*. Eyrolles, 1990.

[61] C. Buchheim, A. Caprara, and A. Lodi. An effective branch-and-bound algorithm for convex quadratic integer programming. *Mathematical Programming*, pages 1–27, 2011.

[62] S. Burer and A. N. Letchford. Non-convex mixed-integer nonlinear programming: A survey. *Surveys in Operations Research and Management Science*, 17(2):97–106, 2012.

[63] S. Burer and R. D. C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Math. Program.*, 95(2):329–357, 2003.

[64] S. Burer and A. Saxena. Old wine in a new bottle: The MILP road to MIQCP. Technical report, Dept of Management Sciences, University of Iowa, 2009.

[65] S. Burer and D. Vandenbussche. A finite branch-and-bound algorithm for nonconvex quadratic programming via semidefinite relaxations. *Mathematical Programming*, 113(2):259–282, 2008.

[66] G. Calafiore and M. Campi. *Multiple Participant Decision Making*, chapter Decision making in an uncertain environment: the scenario-based optimization approach, pages 99–111. Advanced Knowledge International, 2004.

[67] G. Calafiore and M. Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Contro*, 51(5):742–753, 2006.

[68] G. Calafiore and L. El Ghaoui. On distributionally robust chance-constrained linear programs. *Journal of Optimization Theory and Applications*, 130(1):1–22, 2006.

[69] E. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? Technical report, Department of Statistics, Stanford University, 2009.

[70] E. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9:717–772, 2009.

[71] A. Charnes and W. Cooper. Chance-constrained programming. *Management Science*, 6:73–89, 1959.

[72] W. Chen, M. Sim, J. Sun, and C. Teo. From CVaR to uncertainty set : Implications in a joint chance-constrained optimization. *Operations research*, 58(2):470–485, 2010.

[73] J. Cheng and A. Lisser. A second-order cone programming approach for linear programs with joint probabilistic constraints. *Operations Research Letters*, 40(5):325–328, 2012.

[74] V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4:305–337, 1973.

[75] V. Chvátal. *Linear Programming*. Series of Books in the Mathematical Sciences. W.H. Freeman, 1983.

[76] S. Cook. The complexity of theorem proving procedures. *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[77] R. Curto and L. A. Fialkow. The truncated complex k-moment problem. *Trans. Amer. Math. Soc.*, 352:2825–2855, 2000.

[78] R. Dakin. A tree search algorithm for mixed integer programming problems. *Computer Journal*, 8:250–255, 1965.

[79] A. D'Aspremont, L. El-Ghaoui, M. I. Jordan, and G. R. G. Lanckriet. A direct formulation for sparse pca using semidefinite programming. *SIAM Review*, 49(3):434–448, 2007.

[80] E. De Klerk, H. Van Maaren, and J. Warners. Relaxations of the satisfiability problem using semidefinite programming. *Journal of Automated Reasoning*, 24(1–2):37–65, 2000.

[81] I. Deák, I. Pólik, A. Prékopa, and T. Terlaky. Convex approximations in stochastic programming by semidefinite programming. *Annals of Operations Research*, pages 1–12, Oct. 2011.

[82] E. Delage and Y. Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations research*, 58(3):595–612, 2010.

[83] C. Delorme and S. Poljak. Laplacian eigenvalues and the maximum cut problem. *Mathematical Programming*, 62(3 Serie 1):557–574, 1993.

[84] M. Deza and M. Laurent. Applications of cut polyhedra. 1992.

[85] I. Dikin. Iterative solution of problems of linear and quadratic programming. *Soviet Math. Dokl*, 8, 1967.

[86] Y. Ding. *On Efficient Semidefinite Relaxations for Quadratically Constrained Quadratic Programming*. PhD thesis, University of Waterloo, Canada, 2007.

[87] S. Drewes and S. Ulbrich. *Mixed integer second order cone programming*. PhD thesis, Technische Universität Darmstadt, 2009.

[88] M. A. Duran and I. E. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36:307–339, 1986.

[89] L. El Ghaoui. Robust solutions to least-square problems to uncertain data matrices. *SIAM Journal of Matrix Analysis and Applications*, 18:1035–1064, 1997.

[90] L. El Ghaoui, F. Oustry, and H. Lebret. Robust solutions to uncertain semidefinite programs. *SIAM Journal of optimization*, 9(1):33–52, 1998.

[91] A. Faye and F. Roupin. Partial lagrangian relaxation for general quadratic programming. *4'OR, A Quarterly J. of Operations Research*, 5(1):75–88, 2007.

[92] R. Fletcher and S. Leyffer. Solving mixed integer programs by outer approximation. *Mathematical Programming*, 66:327–349, 1994.

[93] C. A. Floudas. *Nonconvex Optimization And Its Applications series*, volume 37, chapter Deterministic Global Optimization: Theory, Methods and Applications. Kluwer Academic Publishers, Boston, 2000.

[94] R. Fortet. Applications de l'algebre de boole en recherche opérationnelle. *Revue Francaise de Recherche Operationnelle*, 4:17–26, 1960.

[95] F. Fourcade, T. Eve, and T. Socroun. Improving lagrangian relaxation: an application to the scheduling of pressurized water reactor outages. *IEEE transactions on power systems*, 12(2):919–925, 1997.

[96] F. Fourcade, E. Johnson, M. Bara, and P. Cortey-Dumont. Optimizing nuclear power plant refueling with mixed-integer programming. *European Journal of Operational Research*, pages 269–280, 1997.

[97] R. Freund. Introduction to semidefinite programming (SDP). MIT OpenCourseWare (Massachusetts Institute of Technology -OpenCourseWare), 2004.

[98] T. Fujie and M. Kojima. Semidefinite programming relaxation for nonconvex quadratic programs. *Journal of Global Optimization*, 10:367–380, 1997.

[99] K. Fukuda, T. M. Liebling, and F. Margot. Analysis of backtrack algorithms for listing all vertices and all faces of a convex polyhedron. *Computational Geometry*, 8:1–12, 1997.

[100] A. Gaivoronski, A. Lisser, and R. Lopez. Knapsack problem with probability constraints. *Journal of Global Optimization*, 49(3):397–413, 2011.

[101] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[102] A. M. Geoffrion. A generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 1972.

[103] P. E. Gill, W. Murray, M. A. Saunders, J. A. Tomlin, and M. H. Wright. On projected newton barrier methods for linear programming and an equivalence to karmarkar's projective method. *Mathematical Programming*, 36:183–209, 1986.

[104] F. Glineur. *Topics in convex optimization: Interior-point methods, conic duality and approximations*. PhD thesis, Polytechnic Faculty of Mons, 2000.

[105] F. Glineur. Conic optimization: an elegant framework for convex optimization. *Belgian Journal of Operations Research, Statistics and Computer Science*, 41:5–28, 2001.

[106] F. Glover. Surrogate contraints. *Operations Research*, 16:741–749, 1968.

[107] M. X. Goemans. Semidefinite programming and combinatorial optimization. *Documenta Mathematica*, 3:657–666, 1998.

[108] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995.

[109] M. X. Goemans and D. P. Williamson. Approximation algorithms for max-3-cut and other problems via complex semidefinite programming. *Journal of Computer and System Sciences*, 68(2):442–470, 2004.

[110] J. Goh and M. Sim. Distributionally robust optimization and its tractable approximations. *Operations research*, 58(4):902–917, 2010.

[111] R. E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematics Society*, 64(5):275–278, 1958.

[112] A. Gorge, J. Cheng, A. Lisser, and R. Zorgati. Investigating a distributionnally robust approach for handling a joint chance-constraint. Submitted to SIAM.

[113] A. Gorge, A. Lisser, and R. Zorgati. Generating cutting planes for the semidefinite relaxation of quadratic programs. Submitted to COR.

[114] A. Gorge, A. Lisser, and R. Zorgati. Semidefinite relaxations for mixed 0-1 second-order cone program. *Lecture Notes in Computer Science*, 7422:81–92, 2012.

[115] A. Gorge, A. Lisser, and R. Zorgati. Semidefinite relaxations for the scheduling nuclear outages problem. In *Proceedings of the 1st International Conference on Operations Research and Enterprise Systems*, pages 386–391, 2012.

[116] A. Gorge, A. Lisser, and R. Zorgati. Stochastic nuclear outages semidefinite relaxations. *Computational Management Science*, 9(3):363–379, 2012.

[117] I. E. Grossmann, J. Viswanathan, and A. Vecchietti. Dicopt : A discrete continuous optimization package. Technical report, Carnegie Mellon University, Pittsburgh, USA, 2001.

[118] L. S. A. Grötschel, Martin; Lovász. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

[119] L. S. A. Grotschel, Martin; Lovász, editor. *Geometric Algorithms and Combinatorial Optimization*. Springer, 1988.

[120] G. Gruber. *On semidefinite programming and applications in combinatorial optimization*. Shaker Verlag, department of mathematics, university of klagenfurt edition, 2000.

[121] Z. Gu, E. Rothberg, and R. Bixby. Gurobi. www.gurobi.com.

[122] O. K. Gupta and A. Ravindran. Branch and bound experiments in convex nonlinear integer programming. *Management Science*, 31(12):1533–1546, 1985.

[123] P. Hammad. *Cours de probabilités*. Cujas, 1984.

[124] P. Hammer and A. Rubin. Some remarks on quadratic programming with 0-1 variables. *RAIRO*, 3(67–79), 1970.

[125] E. P.-H. Hao. Quadratically constrained quadratic programming: some applications and a method for solution. *Mathematical Methods of Operations Research*, 26:105–119, 1982.

[126] J. Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.

[127] C. Helmberg. Semidefinite programming for combinatorial optimization. Habilitationsschrift, TU Berlin, Konrad-Zuse-Zentrum, 2000.

[128] C. Helmberg and F. Rendl. Solving quadratic (0; 1)-problems by semidefinite programs and cutting planes. *Mathematical Programming*, 82(3):291–315, 1998.

[129] C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. *SIAM Journal on Optimization*, 10:673–696, 2000.

[130] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz. An interior-point method for semidefinite programming. *SIAM Journal on Optimization*, 6:342–361, 1996.

[131] C. Helmberg, F. Rendl, and R. Weismantel. A semidefinite programming approach to the quadratic knapsack problem. *Journal of Combinatorial Optimization*, 4:197–215, 2000.

[132] D. Henrion, J.-B. Lasserre, and J. Löfberg. Gloptipoly 3: moments, optimization and semidefinite programming. *Optimization Methods and Software*, 24(4–5):761–779, 2009.

[133] R. Henrion. Introduction to chance constraint programming. Technical report, Tutorial paper for the Stochastic Programming Community HomePage, 2004. http://stoprog.org.

[134] N. J. Higham. Analysis of the cholesky decomposition of a semi-definite matrix. In *Reliable Numerical Computation*, pages 161–185. University Press, 1990.

[135] H. Hindi. A tutorial on convex optimization. In *Proceedings of American Control Conference*, Boston, USA, 2004.

[136] H. Hindi. A tutorial on convex optimization ii : Duality and interior-point method. In *Proceedings of American Control Conference*, Minneapolis, Minnesota, USA, 2004.

[137] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis And Minimization Algorithms: Part 1: Fundamentals*. Springer-Verlag, 1996.

[138] W. Hoeffding. Probability inequalities for sums of bounded random variables. *American Statistical Association Journal*, 58:13–30, 1963.

[139] R. A. Horn and C. R. Johnson. *Matrix Analysis*. 1990.

[140] R. Horst. On the global minimization of concave functions : Introduction and survey.

[141] R. Horst, P. M. Pardalos, and N. V. Thoai. *Introduction to Global Optimization*. Kluwer Academic Publishers, 2000.

[142] R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches*. Springer, 1990.

[143] IBM. Ibm ilog cplex optimizer. http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/, 2012.

[144] A. Ilan and R. D. Monteiro. Interior path following primal-dual algorithms. part i : Linear programming. *Mathematical Programming*, 44:27–42, 1988.

[145] K. Isii. On sharpness of chebyshev-type inequalities. *Ann. Inst. Stat. Math.*, 14:185–197, 1963.

[146] G. Iyengar and M. Çezik. Cut generation for mixed 0-1 quadratically constrained programs. Technical report, CORC TR-2001-02, 2001.

[147] G. Iyengar and M. Çezik. Cuts for mixed 0-1 conic programs. *Math. Prog. Serie A*, 104:179–200, 2005.

[148] G. Iyengar and E. Erdogan. Ambiguous chance constrained problems and robust optimization. *Math. Prog. Series B*, 107(1–2):37–61, 2006.

[149] D. Karger, R. Motwani, and M. Sudan. Approximate graph coloring by semidefinite programming. *J. ACM*, 45(2):246–265, Mar. 1998.

[150] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[151] R. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, pages 85–103, 1972.

[152] L. Khachiyan and L. Porkolab. On the complexity of semidefinite programs. *Journal of Global Optimization*, 10(4):351–365, 1997.

[153] L. G. Khachiyan. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20:191–194, 1979.

[154] M. O. I. Khemmoudj, H. Bennaceur, and M. Porcheron. When constraint programming and local search solve the scheduling problem of electricité de france nuclear power plant outages. In *12th International Conference on Principles and Practice of Constraint Programming (CP'06)*, pages 271–283. LNCS, september 2006.

[155] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for max-cut and other 2-variable CSPs? *SIAM Journal on Computing*, 37(1):319–357, 2007.

[156] S. Kim and M. Kojima. Second order cone programming relaxation of nonconvex quadratic optimization problems. *Optimization Methods and Software*, 15:201–224, 2001.

[157] V. Klee and G. J. Minty. How good is the simplex algorithm. In *Proceedings of the Third Symposium on Inequalities*, pages 159–175. University of California, Los Angeles, 1972.

[158] M. Kocvara and M. Stingl. On the solution of large-scale SDP problems by the modified barrier method using iterative solvers. *Mathematical Programming, Series B*, 109(2–3):413–444, 2007.

[159] M. Kojima, S. Mizuno, and A. Yoshise. An $o(\sqrt{n}l)$ iteration potential reduction algorithm for linear complementarity problems. *Mathematical Programming*, 44:1–26, 1989.

[160] M. Kojima, S. Mizuno, and A. Yoshise. A polynomial-time algorithm for a class of linearity complementarity problems. *Mathematical Programming*, 44:1–26, 1989.

[161] M. Kojima and L. Tunçel. Cones of matrices and successive convex relaxations of nonconvex sets. *SIAM Journal on Optimization*, 10(3):750–778, 1998.

[162] S. Kosuch and A. Lisser. On two-stage stochastic knapsack problems with or without probability constraint. *Discrete Applied Mathematics*, 159(16):1827–1841, 2011.

[163] K. Krishnan and J. E. Mitchell. A cutting plane lp approach to solving semidefinite programming problems. Technical report, Dept. of Mathematical Sciences, RPI, Troy, NY, 2001.

[164] K. Krishnan and J. E. Mitchell. A unifying framework for several cutting planes methods for semidefinite programming. *Optimization methods and software*, 21(1):57–74, 2006.

[165] B. Kulis, A. C. Surendran, and J. C. Platt. Fast low-rank semidefinite programming for embedding and clustering. In *in Eleventh International Conference on Artifical Intelligence and Statistics, AISTATS*, 2007.

[166] G. R. G. Lanckriet, N. Cristianini, P. Barlett, L. El-Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *The Journal of Machine Learning Research*, 5:27–72, 2004.

[167] A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(2):497–520, 1960.

[168] J.-B. Lasserre. An explicit exact SDP relaxation for nonlinear 0-1 programs. In *Proceedings of the 8th International IPCO Conference on Integer Programming and Combinatorial Optimization*, pages 293–303, London, UK, UK, 2001. Springer-Verlag.

[169] J.-B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM J. Optimization*, pages 796–817, 2001.

[170] J.-B. Lasserre. An explicit equivalent positive semidefinite program for nonliner 0-1 programs. *SIAM Journal of Optimization*, 12:756–769, 2002.

[171] J.-B. Lasserre. A semidefinite programming approach to the generalized problem of moments. *Mathematical Programming*, 112:65–92, 2008.

[172] J.-B. Lasserre. Moments and sums of squares for polynomial optimization and related problems. *Journal of Global Optimization*, 45:39–61, 2009.

[173] M. Laurent. A comparison of the Sherali-Adams, Lovász-Schrijver and Lasserre relaxations for 0-1 programming. *Mathematics of Operations Research*, 28(3):470–496, 2003.

[174] M. Laurent. *Emerging Applications of Algebraic Geometry*, volume 149, chapter Sums of squares, Moment matrices and optimization over polynomials, pages 157–270. Springer, ima volumes in mathematics and its applications edition, 2009.

[175] M. Laurent, S. Poljak, and F. Rendl. Connections between semidefinite relaxations of the max-cut and stable set problems. *Mathematical Programming*, 77:225–246, 1997.

[176] M. Laurent and F. Rendl. *Handbook on Discrete Optimization*, chapter Semidefinite Programming and Integer Programming, pages 393–514. Elsevier B.V., 2005.

[177] C. Lemaréchal and F. Oustry. Semidefinite relaxations and lagrangian duality with application to combinatorial optimization. Technical report, INRIA, 1999.

[178] C. Lemaréchal and F. Oustry. *Advances in Convex Analysis and Global Optimization*, chapter SDP relaxations in combinatorial optimization from a Lagrangian viewpoint. Kluwer, 2001.

[179] A. Letchford and L. Galli. Reformulating mixed-integer quadratically constrained quadratic programs. Technical report, Department of Management Science, Lancaster University, United Kingdom, 2011.

[180] S. Leyffer. Integrating SQP and branch-and-bound for mixed integer nonlinear programming. *Computational Optimization and Applications*, 18:295–309, 1998.

[181] L. Liberti. *Reformulation and Convex Relaxation Techniques for Global Optimization*. PhD thesis, Imperial College London, 2004.

[182] L. Liberti. Introduction to global optimization. Technical report, LIX, Ecole polytechnique, 2008.

[183] J. Linderoth. A simplicial branch-and-bound algorithm for solving quadratically constrained quadratic programs. *Mathematical Programming*, 103(2):251–282, 2005.

[184] A. Lisser and R. Lopez. Application of semidefinite relaxation and VNS for multiuser detection in synchronous CDMA. *Networks*, 5:187–193, 2010.

[185] M. S. Lobo, L. Vandenberghe, H. Lebret, and S. Boyd. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284:193–228, 1998.

[186] L. Lovász. On the shannon capacity of a graph. *IEEE Transactions on Information Theory*, 25(1):1–7, 1979.

[187] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal of Optimization*, 1:166–190, 1991.

[188] A. Makhorin. The GNU linear programming kit (GLPK). `http://www.gnu.org/software/glpk/glpk.html`, 2000.

[189] J. Malick and F. Roupin. Numerical study of semidefinite bounds for the k-cluster problem. *Electronic Notes in Discrete Mathematics*, 36:399–406, 2010.

[190] G. P. McCormick. Computability of global solutions to factorable nonconvex programs : Part I - convex underestimating problems. *Mathematical Programming*, 10:146–175, 1976.

[191] N. Megiddo. *Progress in Mathematical Programming : Interior-Point and related methods*, chapter Pathways to the optimal set in linear programming, pages 131–158. Springer-Verlag, New York, 1988.

[192] N. Megiddo. Linear programming. Technical report, For the Encyclopedia of Microcomputers, 1991.

[193] M. Mevissen. Introduction to concepts and advances in polynomial optimization. Technical report, Institute for Operations Research, ETH Zurich, 2007.

[194] M. Minoux. *Programmation mathématique : Théorie et algorithmes*. Tec et Doc - Lavoisier, 2007.

[195] R. Misener and C. A. Floudas. Global optimization of mixed-integer quadratically-constrained quadratic programs (MIQCQP) through piecewise-linear and edge-concave relaxations. *Mathematical Programming*, 2012.

[196] H. Mittelmann. Benchmarks for optimization software. `http://plato.asu.edu/bench.html`, 2011.

[197] H. Mittelmann. Several SDP-codes on sparse and other SDP problems. `http://plato.asu.edu/ftp/sparse_sdp.html`, 2011.

[198] R. Monteiro. Implementation of primal-dual methods for semidefinite programming based on monteiro and tsuchiya newton directions and their variants. Technical report, School Industrial and Systems Engineering, Georgia Tech., Atlanta, 1997.

[199] T. Motzkin, R. H., T. G.L., and T. R.M. *Contributions to theory of games*, volume 2, chapter The double description method. Princeton University Press, 1953.

[200] G. L. Nemhauser and L. A. Wolsey. *Integer and combinatorial optimization*. Wiley-Interscience, 1998.

[201] A. Nemirovski and A. Shapiro. Convex approximations of chance constrained programs. *SIAM Journal of Optimization*, 17(4):969–996, 2006.

[202] A. Nemirovski and A. Shapiro. *Probabilistic and Randomized Methods for Design under Uncertainty*, chapter Scenario Approximations of Chance Constraints. Springer, 2006.

[203] Y. Nesterov. Semidefinite relaxation and nonconvex quadratic optimization. *Optimization methods and software*, 9:141–160, 1998.

[204] Y. Nesterov. *High Performance Optimization*, chapter Squared functional systems and optimization problems, pages 405–440. Kluwer Academic Publishers, 2000.

[205] Y. Nesterov and A. Nemirovski. 'Conic' formulation of a convex programming problem and duality. *Optimization & Software*, 1:10–31, 1992.

[206] Y. Nesterov and A. Nemirovski. *Interior-point polynomial methods in convex programming*. SIAM, 1994.

[207] Y. Nesterov and M. Todd. Self-scaled barriers and interior-point methods for convex programming. *Mathematics of operations research*, 22(1):1–42, 1997.

[208] M. Overton and H. Wolkowicz. Semidefinite programming (foreword to a special issue on the subject). *Mathematical Programming*, 77:105–110, 1997.

[209] P. A. Parrilo. Semidefinite programing relaxations for semialgebraic problems. *Mathematical Programming*, 96(2):293–320, 2003.

[210] G. Pataki. Cone-LP's and semidefinite programs: Geometry and a simplex-type method. In W. Cunningham, S. McCormick, and M. Queyranne, editors, *Integer Programming and Combinatorial Optimization*, volume 1084 of *Lecture Notes in Computer Science*, pages 162–174. Springer Berlin Heidelberg, 1996.

[211] G. Pataki. On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Mathematics of operations research*, 23(2), 1998.

[212] I. Polik and T. Terlaky. A survey of the s-lemma. *SIAM Review*, 49(3):371–418, 2007.

[213] S. Poljak, F. Rendl, and H. Wolkowicz. A recipe for semidefinite relaxation for (0,1)-quadratic programming. *Journal of Global Optimization*, 7:51–73, 1995.

[214] M. Porcheron, A. Gorge, O. Juan, T. Simovic, and G. Dereu. Challenge ROADEF/EURO 2010 : A large-scale energy management problem with varied constraints. Technical report, EDF R&D, 2009. `http://challenge.roadef.org`.

[215] F. A. Potra and W. S. J. Interior-point methods. In *Society for Industrial and Applied Mathematics (SIAM)*. SIAM, 2000.

[216] A. Prékopa. On probabilistic constrained programming. In *Proceedings of the Princeton Symposium on Mathematical Programming*, pages 113–138. Princeton University Press, 1970.

[217] A. Prékopa. *Stochastic Programming*. Springer, Kluwer, Dordrecht, Boston, 1995.

[218] I. Quesada and I. E. Grossmann. An lp/nlp based branch and bound algorithm for convex minlp optimization problems, 1992.

[219] M. Ramana and A. J. Goldman. Quadratic maps with convex images. Technical report, Rutgers center for operations research, 1995.

[220] M. V. Ramana. PhD thesis, Johns Hopkins University, 1993.

[221] M. V. Ramana. An exact duality theory for semidefinite programming and its complexity implications. *Mathematical Programming*, 77:129–162, 1997.

[222] M. V. Ramana, L. Tüncel, and H. Wolkowicz. Strong duality for semidefinite programming. *SIAM Journal on Optimization*, 7(3):641–662, 1997.

[223] F. Rendl. *50 Years of Integer Programming*, chapter Semidefinite Relaxations for Integer Programming, pages 1–41. Springer, 2010.

[224] F. Rendl, G. Rinaldi, and A. Wiegele. A branch and bound algorithm for max-cut based on combining semidefinite and polyhedral relaxations. In *IPCO*, pages 295–309, 2007.

[225] F. Rendl and H. Wolkowicz. A semidefinite framework for trust region subproblems with applications to large scale minimization. *Mathematical Programming*, 77(1):273–299, 1997.

[226] J. Renegar. Linear programming, complexity theory and elementary functional analysis. *Mathematical Programming*, 70:279–351, 1994.

[227] R. Rockafellar and S. Uryasev. *Stochastic Optimization : algorithms and applications*, chapter Conditional Value-at-Risk : optimization approach, pages 411–435. Kluwer, 2001.

[228] R. T. Rockafellar. Optimization under uncertainty. Lecture Notes, University of Washington.

[229] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[230] F. Roupin. From linear to semidefinite programming: an algorithm to obtain semidefinite relaxations for bivalent quadratic problems. *Journal of Combinatorial Optimization*, 8(4):469–493, 2004.

[231] A. Ruszczynski and A. Shapiro, editors. *Stochastic Programming*, volume 10 of *Handbooks in Operations Research and Management Science*. Elsevier, Amsterdam, 2003.

[232] N. V. Sahinidis. BARON: A general purpose global optimization software package. *Journal of Global Optimization*, 8:201–205, 1996.

[233] N. V. Sahinidis. Optimization under uncertainty: state-of-the-art and opportunities. *Computers and Chemical Engineering*, 28:971–983, 2004.

[234] M. J. Saltzman. COIN-OR: An open-source library for optimization. *Programming Languages and Systems in Computational Economics and Finance*, 2002. Kluwer.

[235] A. Saxena, P. Bonami, and J. Lee. Convex relaxations of non-convex mixed integer quadratically constrained programs: Extended formulations. *Mathematical Programming*, 124(1–2):383–411, 2010. Series B - Special Issue: Combinatorial Optimization and Integer Programming.

[236] A. Saxena, P. Bonami, and J. Lee. Convex relaxations of non-convex mixed integer quadratically constrained programs: Projected formulations. *Mathematical Programming*, 130(2):359–413, 2011.

[237] H. Scarf. *Studies in the Mathematical Theory of Inventory and Production*, chapter A min-max solution of an inventory problem, pages 201–209. Stanford University Press, 1958.

[238] A. Schrijver. *Theory of linear and integer programming.* Series in discrete mathematics and optimization. Wiley-interscience, 1986.

[239] H. D. Sherali. RLT: a unified approach for discrete and continuous nonconvex optimization. *Annals of Operations Research*, 149(1):185–193, 2007.

[240] H. D. Sherali and W. P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3):411–430, 1990.

[241] H. D. Sherali and W. P. Adams. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems.* Kluwer Academic Publishers, Dordrecht, 1999.

[242] H. D. Sherali and B. M. P. Fraticelli. Enhancing rlt relaxations via a new class of semidefinite cuts. Technical report, Virginia Polytechnic Institute and State University, 2000.

[243] H. D. Sherali and C. H. Tuncbilek. A reformulation-convexification approach for solving nonconvex quadratic programming problems. *Journal of Global Optimization*, 7(1):1–31, 1995.

[244] N. Shor. Utilization of the operation of space dilation in the minimization of convex function. *Cybernetics*, 6:7–15, 1970.

[245] N. Shor. Quadratic optimization problems. *Soviet Journal of computer Systems Sciences*, 25:1–11, 1987.

[246] A. M.-C. So and Y. Ye. Theory of semidefinite programming for sensor network localization. *Mathematical Programming*, 109(2):367 – 384, 2007.

[247] R. A. Stubbs and S. Mehrotra. A branch and cut method for 0-1 mixed convex programming. *Mathematical Programming*, 86(3):515–532, 1999.

[248] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. 1998.

[249] C. Swamy. Correlation clustering: Maximizing agreements via semidefinite programming. *Proceedings of SODA*, pages 519–520, 2004.

[250] M. Tawarmalani and N. Sahinidis. *Nonconvex Optimization And Its Applications series*, volume 65, chapter Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications,. Kluwer Academic Publishers, Boston, 2002.

[251] M. Todd. A study of search directions in primal-dual interior-point methods for semidefinite programming. *Optimization methods and software*, 11(1–4):1–46, 1999.

[252] M. Todd. Semidefinite optimization. *Acta Numerica*, 10:515–560, 2001.

[253] M. J. Todd. Potential-reduction methods in mathematical programming. *Mathematical Programming*, 76(1):3–45, 1997.

[254] K. Toh, M. J. Todd, and R. Tutuncu. SDPT3 — a Matlab software package for semidefinite programming. *Optimization Methods and Software*, 11:545–581, 1999.

[255] L. Trevisan. Approximation algorithms for unique games. *Theory of Computing*, 4(1):111–128, 2008.

[256] C. Van de Panne and W. Popp. Minimum cost cattle feed under probabilistic problem constraint. *Management Science*, 9:405–430, 1963.

[257] L. Vandenberghe, S. Boyd, and K. Comanor. Generalized chebyshev bounds via semidefinite programming. *SIAM Review*, 49(1):52–64, 2007.

[258] T. Westerlund and F. Pettersson. *A Cutting Plane Method for Solving Convex MINLP Problems*. Åbo Akademi, 1994.

[259] H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*. Kluwer Academic Publishers, 2000.

[260] M. Wright. The interior-point revolution in optimization : history, recent developments and lasting consequences. *Bulletin of the American Mathematical Society*, 42:39–56, 2005.

[261] V. A. Yakubovich. S-procedure in nonlinear control theory. *Vestnik Leningrad. Univ.*, pages 62–77, 1971. in Russian.

[262] V. A. Yakubovich. The S-procedure and duality theorems for nonconvex problems of quadratic programming. *Vestnik Leningrad. Univ.*, 1:81–87, 1973.

[263] M. Yamashita, K. Fujisawa, K. Nakata, M. Nakata, M. Fukuda, K. Kobayashi, and K. Goto. A high-performance software package for semidefinite programs: SDPA 7. Technical report, Dept. of Mathematical and Computing Science, Tokyo Institute of Technology, 2010.

[264] Y. Ye. Approximating quadratic programming with bound and quadratic constraints. *Mathematical programming*, 84:219–226, 1999.

[265] Y. Zhang. On extending primal-dual interior-point algorithms from linear programming to semidefinite programming. Technical report, Department of Mathematics and Statistics, University of Maryland, Baltimore County, 1995.

[266] X. Zhao, D. Sun, , and K.-C. Toh. A newton-CG augmented lagrangian method for semidefinite programming. *SIAM Journal of Optimization*.

[267] X. J. Zheng, X. L. Sun, D. Li, and Y. F. Xu. On zero duality gap in nonconvex quadratic programming problems. *Journal of Global Optimization*, 52(2):229–242, 2012.

[268] R. Zorgati and B. Duchêne. Chance-constrained programming : a tool for solving linear eddy current inverse problem. *Studies in Applied Electromagnetics and Mechanics, Electromagnetic Nondestructive Evaluation (XII)*, 32:305–312, 2009.

[269] R. Zorgati, W. Van Ackooij, and A. Gorge. Uncertainties on power systems. probabilistic approach and conic approximation. *IEEE 11th International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, 2010.

[270] S. Zymler, D. Kuhn, and B. Rustem. Distributionnally robust joint chance constraint with second-order moment information. *Mathematical Programming*, 2011.

Appendix

This appendix, and particularly the mathematical and optimization backgrounds part, serves as a reference to keep this document self-contained. To this end, the mathematical concepts and results that are used throughout this thesis are stated briefly, without proof.

Chapter 1

Notations and abbreviations

1.1 General remarks and abbreviations

First we make a very general comment on the term *SDP*, which denotes the field of Semidefinite Programming, or a Semidefinite Program, depending on the context. We use the acronym *psd* to indicate that a matrix is positive semidefinite matrix.

Furthermore, an optimization problem is formulated as follows : (P) $\min_{x\in\mathcal{R}}\{f_0(x) : f_i(x) \leq 0,\ i = 1, ..., m\}$ or :

$$(P)\begin{cases} p^* = & \min_{x\in\mathcal{R}} & f_0(x) \\ & \text{s.t.} & f_i(x) \leq 0,\ i = 1, ..., m \end{cases}$$

This means that we are interested in finding the minimal value p^* of f_0 over $\mathcal{F} = \{x \in \mathcal{R} : f_i(x) \leq 0,\ i = 1, ..., m\}$ as well as a minimizer, referred to as the *optimal solution*. This notation implicitly assumes that a minimum of f_0 exists and is attained. When this is possibly not the case, we use inf and sup instead.

The term *s.t.* is an abbreviation for *subject to* and the inequalities $f_i(x) \leq 0$ are called the *constraints* of the problem, whereas f_0 is the *objective function*. This formulation serves as a reference but we will also consider maximization problems, equality constraints and constraints of the form $x \in S \subset \mathbb{R}^n$. The latter will be noted $\max_{x\in S} f(x)$ or $\max\{f(x) : x \in S\}$.

Below is the list of the abbreviations denoting the various optimization area :

LP	Linear Programming
ILP	Integer Linear Programming
MILP	Mixed Integer Linear Programming
0/1-LP	Binary Linear Programming
M0/1-LP	Mixed Binary Linear Programming
SDP	Semidefinite Programming
QP	Quadratic Programming
QCQP	Quadratically Constrained Quadratic Programming
CQCQP	Convex Quadratically Constrained Quadratic Programming
0/1-QCQP	Binary Quadratically Constrained Quadratic Programming
MIQCQP	Mixed Integer Quadratically Constrained Quadratic Programming
SOCP	Second-Order Conic Programming

In our terminology, a bivalent variable denote a variable that belong to a set of two elements. Among them are the binary variables ($\{0, 1\}$) and the boolean variables $(1, 1\}$).

1.2 Notations

1.2.1 Spaces

$\mathbb{R}$	the real numbers	
$\mathbb{R}^n$	vector space of real n-vectors	
$\mathbb{R}^n_*$	$\mathbb{R}^n \setminus \{0\}$	
$\mathbb{R}^n_+$	$\{x \in \mathbb{R} : x \geq 0\}$	
$\mathbb{R}^n_{++}$	$\{x \in \mathbb{R} : x > 0\}$	
$\mathbb{R}^{n,m}$	vector space of real n-by-m matrices	
$\mathbb{S}^n$	vector space of real n-by-n symmetric matrices	
$\mathbb{S}^n_+$	vector space of real n-by-n positive semidefinite matrices	Def. 2.1.3
$\mathbb{S}^n_{++}$	vector space of real n-by-n positive definite matrices	Def. 2.1.3
$\bar{\mathbb{R}}$	the extended real numbers : $\mathbb{R} \cup \pm\infty$	
$\mathbb{N}^n$	set of integer n-vectors	
$\mathbb{N}^n_d$	subset of $\mathbb{N}^n$ such that $\sum_{i=1}^n \kappa_i \leq d$	Def 2.5.1
$\mathbb{N}^n_{*,d}$	$\mathbb{N}^n_d \setminus (0, ..., 0)$	
$\mathcal{P}^{n,d}$	set of polynomials from $\mathbb{R}^n$ to $\mathbb{R}$ of degree at most d	

1.2.2 Algebra

Here, we assume that the dimension of the following vectors and matrices will be made clear by the context. Note that u_i denotes the i-th component of a vector whereas $u^{(i)}$ will be used to denote the i-th element of a collection $\{u^{(i)}\}_{i \in [n]}$.

1.2.2.1 Vectors

$u^T v$	standard (Euclidian) inner product of the vectors u and v	
$\|u\|$	Euclidean norm of $u \in \mathbb{R}^n$	
$\|u\|_k$	l_k-norm of u	Def. 2.1.12
e	all one vector in $\mathbb{R}^n$ (usually)	
e_i	ith standard basis vector in $\mathbb{R}^n$ (usually)	
0	all zero vector in $\mathbb{R}^n$ (usually)	
$u \geq v$	component-wise inequality of the vectors u and v	
$u = v$	component-wise equality of the vectors u and v	
$u \neq v$	holds if $u = v$ does not hold, or $u_i \neq v_i$ for some i	
$\tilde{u}$	the augmented vector of $u \in \mathbb{R}^n$: $\tilde{u} = \begin{pmatrix} 1 & u^T \end{pmatrix}^T$	

1.2.2.2 Matrices

A_{ij}	(i,j)-th component of the matrix A (i-th row, j-th column)	
A_{i*}	i-th row of the matrix A	
A_{*j}	j-th column of the matrix A	
I	identity matrix of $\mathbb{R}^{n,n}$ (usually)	
D_i	$D_i = e_i e_i^T \in \mathbb{S}^n$ (usually)	
$\|A\|$	Frobenius norm of $A \in \mathbb{R}^{n,m}$	Def. 2.3.1
A^T	transpose of $A \in \mathbb{R}^{m,n}$	
A^{-1}	inverse of a nonsingular matrix $A \in \mathbb{R}^{n,n}$	
$\mathrm{Tr}(A)$	trace of $A \in \mathbb{R}^{n,n}$, i.e. $\mathrm{Tr}(A) = \sum_{i=1}^n A_{ii}$	
$A \bullet B$	Frobenius inner product of $A, B \in \mathbb{R}^{m,n}$	Def. 2.3.1
$\mathrm{Diag}(u)$	diagonal matrix of $\mathbb{S}^n$ made of the components of $u \in \mathbb{R}^n$	
$\mathrm{diag}(A)$	vector of $\mathbb{R}^n$ made of the diagonal elements of $A \in \mathbb{R}^{n,n}$	
$\mathrm{rank}(A)$	rank of a matrix $A \in \mathbb{R}^{n,n}$	
$\det(A)$	determinant of the matrix A	
$\lambda_k(A)$	k-th eigenvalue of the matrix X in the increasing order, $\lambda_1(A) = \lambda_m in(A)$ and $\lambda_n(A) = \lambda_{max}(A)$	
$A \succ 0$	A is positive definite (pd)	Def. 2.1.2
$A \succcurlyeq 0$	A is positive semidefinite (psd)	Def. 2.1.1
$A \succcurlyeq B$	if $A - B$ is positive semidefinite (Löwner partial order)	
$A^{1/2}$	positive semidefinite square root of A	
$A \otimes B$	Kronecker product	Def. 2.3.7
$A \oplus B$	The block-diagonal matrix made of A and B : $\begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}$	
$\mathcal{N}(A)$	null-space of the matrix A	Def. 2.3.18
$\mathcal{R}(A)$	range-space of the matrix A	Def. 2.3.18

1.2.2.3 Polynomials

$\mathbb{R}[\mathsf{x}]$	set of polynomials in $x_1, ..., x_n$ variables with real coefficients	Def. 2.5.4
$\mathbb{R}_d[\mathsf{x}]$	set of polynomials of $\mathbb{R}[x_1, ..., x_n]$ of degree at most d	Def. 2.5.4
$\deg(p)$	degree of the polynomial p	Def. 2.5.3
$\mathsf{p} \in \mathbb{R}^{b_n(d)}$	vector of $\mathbb{R}^{b_n(d)}$ containing the coefficients of the d-degree polynomial p	Def. 2.5.3
$p \geq 0$ on $\mathcal{S}$	$p(x) \geq 0,\ \forall x \in \mathcal{S}$: p is non-negative on $\mathcal{S}$	Def. 2.5.7
$p > 0$ on $\mathcal{S}$	$p(x) > 0,\ \forall x \in \mathcal{S}$: p is positive on $\mathcal{S}$	Def. 2.5.7
p s.o.s.	p is a sum of square	Def. 2.5.8
$p(.; P, p, \pi)$	2-degree polynomial such that $p(x) = x^T P x + 2p^T x + \pi$	§2.5.3
$p(.; Q)$	2-degree polynomial such that $p(x) = \tilde{x}^T Q \tilde{x}$	§2.5.3

1.2.3 Functions and sequences

$\mathrm{dom} f$	domain of the function f	
$\mathrm{epi} f$	epigraph of the function f	Def 2.4.32
$\nabla f(x)$	Gradient of the function f at x	Def 2.4.12
$\nabla^2 f(x)$	Hessian of the function f at x	Def 2.4.19
$\partial \nabla f(x_0)$	Subgradient of the function f at x	Def 2.4.21
$J_f(x)$	The Jacobian matrix of the function f	Def 2.4.20

1.2.4 Sets

$\text{conv}(S)$	convex hull of the set $S \subset \mathcal{R}$	Def 2.1.2
$\text{cone}(S)$	conic hull of the set $S \subset \mathcal{R}$	Def 2.1.2
$\text{aff}(S)$	affine hull of the set $S \subset \mathcal{R}$	Def 2.1.2
$\text{lin}(S)$	affine hull of the set $S \subset \mathcal{R}$	Def 2.1.2
$\text{int}(S)$	interior of the set $S \subset \mathcal{R}$	Def 2.1.28
$\text{rint}(S)$	relative interior of the set $S \subset \mathcal{R}$	Def 2.1.31
$\text{cl}(S)$	closure of the set $S \subset \mathcal{R}$	Def 2.1.29
$\text{bnd}(S)$	boundary of the set $S \subset \mathcal{R}$	Def 2.1.30
$\dim(S)$	dimension of the set $S \subset \mathcal{R}$	Def 2.1.5
$S^\perp$	orthogonal set of $S \subset \mathcal{R}$	Def 2.1.10
S^*	dual cone of the set S	Def. 2.2.40
S°	polar of the set S	Def. 2.2.62
S^C	complement of the set S	
$\lvert S \rvert$	cardinal of the set S	
$\tilde{S}$	homogenization of S, i.e., $\{\lambda \begin{pmatrix} 1 & x^T \end{pmatrix}^T : \lambda \in \mathbb{R},\ x \in S\}$	
$S_1 + S_2$	set of all the sum of vectors from S_1 and S_2	
$S_1 \times S_2$	Cartesian product of S_1 and S_2	
$-S$	set of all the vectors whose negative lie in S	
$S^\perp$	set of all the vectors that are orthogonal to all the vectors of S	Def. 2.1.10
$\mathbb{1}_S$	indicator function of S	Def. 2.1.18
$B(x, r)$	$B(x, r) = \{y : \lVert y - x \rVert_2 \leq r\}$: the ball of radius r with center x	

1.2.5 Uncertainties

$\mathcal{M}(\Omega)$	Set of non negative measure over the Borel σ-algebra of Ω	Def. 2.6.3
$\mathcal{N}(\mu, \sigma)$	Gaussian distribution of mean μ and variance σ^2	Def. 2.6.14
$\text{P}[A]$	Probability of an event A	Def. 2.6.9
$\text{E}(X)$	Expected value of a random variable X	Def. 2.6.27
$\text{var}(X)$	Variance of X	Def. 2.6.31
$\sigma(X)$	Standard deviation of X	Def. 2.6.31
$\text{VaR}_\varepsilon(X)$	ε-Value at Risk of the real random variable X	Def. 2.6.42
$\text{CVaR}_\varepsilon(X)$	ε-Conditional Value at Risk of the real random variable X	Def. 2.6.44

1.2.6 Miscellaneous

$[n]$	the set of integer from 1 to n, $[n] = \{1, ..., n\}$
$n!$	factorial of n
$\binom{n}{k}$	binomial coefficient of n and k, equal to $n!/k!(n-k)!$
$b_n(d)$	binomial coefficient of $n + d$ and d, equal to $\binom{n+d}{d}$
$t(n)$	$t(n) = n(n+1)/2$, i.e., the n-th triangular number
$s_1 \bigvee s_2$	a statement that is true if and only if at least one of the two statements s_1 or s_2 is true

Chapter 2

Mathematical Background

The objective of this section is to state briefly, without proof, the mathematical concepts and results that are used throughout this thesis, in order to keep it self-contained. It is mainly based on [59, 229] for convex analysis and on [139, 259] for linear algebra.

Unless otherwise stated, we work in an Euclidean space $\mathcal{R}$. Generally, $\mathcal{R} = \mathbb{R}^n$ or $\mathcal{R} = \mathbb{S}^n$. By default, we use notations valid for $\mathbb{R}^n$, i.e., x_i to designate the i-th component of x and $x^T y$ for the inner product. Remark that there is a canonical mapping from $\mathbb{S}^n$ to $\mathbb{R}^{t(n)}$ and therefore, if needed, $\mathbb{S}^n$ can be considered in the same way as the Euclidean space $\mathbb{R}^{t(n)}$.

2.1 Basic concepts

2.1.1 Vector spaces

Definition 2.1.1 *Combinations*
Given a collection of vectors $\{x^{(i)}\}_{i=1,..,m}$ of $\mathcal{R}$ and a scalar coefficient vector $\lambda \in \mathbb{R}^m$, the combination $\sum_{i=1}^m \lambda_i x^{(i)}$ is said to be :

- *positive if $\lambda > 0$;*
- *affine combination if $\sum_i \lambda_i = 1$;*
- *convex if $\lambda \geq 0$ and $\sum_i \lambda_i = 1$.*
- *linear otherwise ;*

For example, the set of convex combinations of two points is the *line segment* connecting these two points.

For each of this four adjectives, we define a corresponding hull :

Definition 2.1.2 *Hull*
Given a subset $S \subset \mathcal{R}$:

- $\text{cone}(S)$ *is the set of the positive combination of elements of S;*
- $\text{aff}(S)$ *is the set of the affine combination of elements of S;*
- $\text{conv}(S)$ *is the set of the convex combination of elements of S;*
- $\text{lin}(S)$ *is the set of the linear combination of elements of S;*

These set can be viewed as the smallest cone, affine set, convex set, linear set that contains S respectively.

Definition 2.1.3 *Linear independence*
A collection of vectors $\{x^{(i)}\}_{i=1,...m}$ *of* $\mathcal{R}$ *are linearly independent if there exists no vector of scalars* $\lambda \in \mathbb{R}_*^m$ *such that* $\sum_{i=1}^m \lambda_i x^{(i)} = 0$.

Definition 2.1.4 *Affine independence*
A collection of vectors $\{x^{(i)}\}_{i=1,...,m}$ *of* $\mathcal{R}$ *are affinely independent if* $\sum_{i=1}^m \lambda_i x^{(i)} = 0$ *and* $\sum_{i=1}^m \lambda_i = 0$ *together imply that* $\lambda_i = 0$ *for all* $i = 1, ..., m$.

Definition 2.1.5 *Dimension*
The dimension of a set S, *denoted* $\dim(S)$, *is the maximal number of affinely independent vectors in* S *minus* 1.

Definition 2.1.6 *Span*
The span of a collection of vectors $V = \{e^{(i)}\}_{i=1,...,m}$ *is the set of linear combination of elements of* V :

$$\text{span}(V) = \left\{ \sum_{i=1}^m \lambda_i e^{(i)} : \lambda \in \mathbb{R}^m \right\}$$

Definition 2.1.7 *Basis*
A basis of the vector space $\mathcal{R}$ *is a collection of linearly independent vectors* V *such that* $\text{span}(V) = \mathcal{R}$.

Definition 2.1.8 *Dual vector space*
Let $\mathcal{R}$ *be a real vector space. Its dual vector space, denoted* $\mathcal{R}^*$, *consists of all linear maps* $L : A \to \mathbb{R}$.

2.1.2 Hilbert space, inner product and norms

Definition 2.1.9 *Hilbert space*
A Hilbert space is a vector space over the field of the real or complex numbers endowed with an inner product.

Definition 2.1.10 *Orthogonal sets*
Two vectors $x, y \in \mathcal{R}$ *are orthogonal if* $x^T y = 0$.

Let $S \subset \mathcal{R}$. *A vector* $y \in \mathcal{R}$ *is orthogonal to* S *if it is orthogonal to any vectors of* S. *Finally,* $S^\perp$ *denotes the set of all the orthogonal vectors to* S, *called orthogonal set of* S.

Definition 2.1.11 *Norm*
A norm $\|.\|$ *on* $\mathcal{R}$ *is a function* $\mathcal{R} \to \mathbb{R}_+$ *that satisfies the following properties for any* $x, y \in \mathcal{R}$:

$\|x\| = 0$ *if and only if* $x = 0$	*(positivity)*
$\|\lambda x\| = \|\lambda\|\|x\|$ *for any scalar* λ	*(homogeneous)*
$\|x + y\| \leq x + y$	*(triangle inequalities)*

Definition 2.1.12 l_k*-norm*
For $k \in [1, +\infty[$, *the* l_k*-norm of a vector* $x \in \mathbb{R}^n$ *is defined by* $\|x\|_k = (\sum_{i=1}^n |x_i|^k)^{1/k}$.

The max-norm, or l_∞-norm is defined as $\|x\|_\infty = \max_{i \in [n]} |x_i|$. By abuse of terminology, the cardinal of x, i.e., $\sum_{i:x_i \neq 0} 1$ is sometimes called the *norm 0* of x and written $\|x\|_0$. However this is not a norm since it is not homogeneous.

Definition 2.1.13 *Dual norm*
Let $\|.\|$ *be a norm. Its dual norm, denoted* $\|.\|_*$ *is defined as* $\|y\|_* = \sup\{x^T y : \forall x \text{ such that } \|x\| \leq 1\}$.

If $p, q \in [1, +\infty[$ satisfy $\frac{1}{p} + \frac{1}{q} = 1$, then l_p and l_q are dual to each other. In particular, the Euclidean norm l_2 is self-dual. In the sequel, $\|.\|$ denotes the Euclidean norm.

Theorem 2.1.14 *Cauchy-Schwarz inequality*
The Cauchy-Schwarz inequality states that $x^T y \leq \|x\|\|y\|$ and the equality holds if and only if x and y are dependent.

Definition 2.1.15 *Orthonormal basis*
A basis $\{e^{(i)}_{i=1,...,n}$ of a Hilbert space $\mathcal{H}$ is orthonormal if $\|e^{(i)}\| = 1, i = 1, ..., n$ and ${e^{(i)}}^T e^{(j)} = 0$ for all $i \neq j$.

Definition 2.1.16 *Euclidean space*
An Euclidean space is a finite-dimensional real Hilbert space.

Definition 2.1.17 *Incidence vector*
Let $\mathcal{U}$ be a finite set with n elements. The incidence vector of $S \subset \mathcal{U}$ is a vector $v \in \{0,1\}^n$ whose entries are labeled with the elements of $\mathcal{U}$: $v_u = 1$ if $u \in S$, otherwise $v_u = 0$.

Definition 2.1.18 *Indicator function*
The indicator function of $S \subset \mathcal{R}$, denoted $\mathbb{1}_S : \mathcal{R} \to \{0,1\}$ is such that $\mathbb{1}_S(x) = 1$ if $x \in S$, $\mathbb{1}_S(x) = 0$ otherwise.

2.1.3 Topology

Definition 2.1.19 *Open and closed ball*
An open (resp.closed) ball around a point $a \in \mathcal{R}$ is a subset of $\mathcal{R}$ of the form $\{x \in \mathcal{R} : \|x - a\| < r\}$ (resp. $\{x \in \mathcal{R} : \|x - a\| \leq r\}$).

Definition 2.1.20 *Open set*
A set $S \subset \mathcal{R}$ is open if it contains an open ball around each of its point.

Definition 2.1.21 *Closed set*
A set $S \subset \mathcal{R}$ is closed if its complement is open.

Example 2.1.22 *$]a, b[$ is open. If f is continuous function over $\mathcal{R}$: $\{x \in \mathcal{R} : f(x) \leq a\}$ is closed.*

Proposition 2.1.23 *In an Euclidean space, a set S is closed if and only if S contains the limit point of each convergent sequence of points in S.*

Definition 2.1.24 *Bounded set*
A set $S \subset \mathcal{R}$ is bounded if there exists $M \in \mathbb{R}$ such that $\|x\| \leq M$ for all $x \in S$.

Definition 2.1.25 *Compact set*
A set S is said compact if for every arbitrary collection of subset $\{S^{(i)}\}_{i \in \mathcal{I}}$ such that $\cup_{iin\mathcal{I}} S^{(i)} = S$, there exists $\mathcal{J}$, a finite subset of $\mathcal{I}$ such that $\cup_{iin\mathcal{J}} S^{(i)} = S$.

Theorem 2.1.26 *Bolzano-Weierstrass theorem*
In an Euclidean space, a set is compact if and only if it is closed and bounded.

Theorem 2.1.27 *Weierstrass' theorem*
The image of a compact set by a continuous real-valued function is compact.

In other words, a continuous real-valued function f of a compact set S into $\mathbb{R}$ attains its maximum and minimum in S, i.e., there are points $x_1, x_2 \in S$ such that $f(x_1) \leq f(x) \leq f(x_2), \forall x \in S$.

As a consequence, if S is a closed set of $\mathcal{R}$ then it is possible to define the function $d_S : \mathcal{R} \to \mathbb{R}$, that measure the distance from x to S :

$$d_S(x) = \min\{\|x - s\| : s \in S\}$$

Indeed, for any point of $s_0 \in S$, $d_S(x)$ is the infimum of $\|x - s\|$ over $S \cap \{s : \|s - x\| \leq \|s_0 - x\|\}$ which is compact. Consequently, the infimum is necessarily attained. If $\bar{s} \in S$ is such that $\|x - \bar{s}\| = d_S(x)$, then $\bar{s}$ is said to be a *nearest point* of S to x. Such a point may not be unique.

Definition 2.1.28 *Interior*
The interior of a set S, denoted $\text{int}(S)$*, is the union of all open set contained in S.*

Definition 2.1.29 *Closure*
The closure of a set S, denoted $\text{cl}(S)$*, is the intersection of all closed set contained in S.*

$\text{int}(S)$ is open since the union of any family of open sets is open and $\text{cl}(S)$ is closed since the intersection of any family of closed set is closed.

Definition 2.1.30 *Boundary*
The boundary of a set S, denoted $\text{bnd}(S)$*, is* $\text{cl}(S) \setminus \text{int}(S)$.

Let present the concept of *relative topology*, which involves the notion of affine hull of a set S (see Def. 2.1.2). Indeed, $\dim(S) = \dim(\text{aff}(S))$ and it is interesting to study S as a subset of $\text{aff}(S)$. In particular, the notion of *relative interior* will be very useful in optimization :

Definition 2.1.31 *Relative interior*
Let S be a subset of $\mathcal{R}$ and $x_0 \in S$. We say that x_0 is a relative interior point of S, denoted $x_0 \in \text{rint}(S)$ *if there is $r > 0$ such that $\{x \in \text{aff}(S) : ||x - x_0|| < r\} \subseteq S$.*

Loosely speaking, $\text{rint}(S)$ would be the interior of S if $\mathcal{R} = \text{aff}(S)$. For example, let us consider a segment in $\mathbb{R}^2$: $S = \{\begin{pmatrix} x_1 x_2 \end{pmatrix}^T : 0 \leq x_1 \leq 1, x_2 = 0\}$. Then $\dim(S) = 1$ and $\text{int}(S) = \emptyset$. However, the relative interior is not empty : $\text{rint}(S) = \{\begin{pmatrix} x_1 x_2 \end{pmatrix}^T : 0 < x_1 < 1, x_2 = 0\}$.

2.2 Geometry

2.2.1 Halfspaces and hyperplanes

Definition 2.2.1 *Halfspace*
A set $H \subset \mathcal{R}$ is an halfspace if it is of the form $H = \{x \in \mathcal{R} : a^T x \leq \alpha\}$ for some nonzero $a \in \mathcal{R}$ and scalar α.

Definition 2.2.2 *Hyperplane*
An set $H \subset \mathcal{R}$ is an hyperplane if it is of the form $H = \{x \in \mathcal{R} : a^T x = \alpha\}$ for some nonzero $a \in \mathcal{R}$ and scalar α. a is called the normal vector of the hyperplane H.

An hyperplane is therefore an affine set of dimension $n - 1$ that divides $\mathcal{R}$ in two halfspace : $H^- = \{x \in \mathcal{R} : a^T x \leq \alpha\}$ and $H^+ = \{x \in \mathcal{R} : a^T x \geq \alpha\}$. It has dimension $n - 1$.

Definition 2.2.3 *Supporting hyperplane*
An hyperplane H is a supporting hyperplane of an arbitrary set S if $S \cap H \neq \emptyset$ and S is contained in one of the two halfspaces generated by H : $S \subset H^-$ or $S \subset H^+$.

$S \cap H \neq \emptyset$ implies that there exists $x_0 \in S$ such that $a^T x_0 = \alpha$. Then a is said to support $\mathcal{C}$ at x_0. The support is proper if $a^T x > \alpha$ for some $x \in S$.

Definition 2.2.4 *Separating hyperplane*
Let us consider two sets $S, T \subset \mathcal{R}$. The hyperplane H is a separating hyperplane for S and T if $S \subset H^-$ and $T \subset H^+$.

Definition 2.2.5 *Strongly separating hyperplane*
Let us consider two sets $S, T \subset \mathcal{R}$. The hyperplane $H = \{x \in \mathcal{R} : a^T x = \alpha\}$ is said to strongly separates S and T if there is an $\varepsilon > 0$ such that $a^T x \leq \alpha - \epsilon, \forall x \in S$ and $a^T x \geq \alpha + \varepsilon, \forall x \in T$.

Definition 2.2.6 *Valid inequality*
An inequality $f(x) \leq 0$ is said to be valid for a set S if it holds over S.

Definition 2.2.7 *Dominated inequality*
A linear inequality $a^T x \leq b$ dominates another linear inequality $c^T x \leq d$ if there exists $\lambda > 0$ such that $a = \lambda c$ and $\lambda b \leq d$. If $\lambda b = d$ the constraints are said to be equivalent.

2.2.2 Convex sets

Definition 2.2.8 *Convex set*
A set $\mathcal{C}$ of $\mathcal{R}$ is said convex if it is closed under convex combination, i.e., if $\lambda x + (1-\lambda)y \in \mathcal{C}$ whenever $x, y \in \mathcal{C}$ and $0 \leq \lambda \leq 1$.

Geometrically, the set of the convex combination of x and y makes up the line segment connecting x to y. This implies that $\mathcal{C}$ contains all the line segments connecting two points of $\mathcal{C}$.

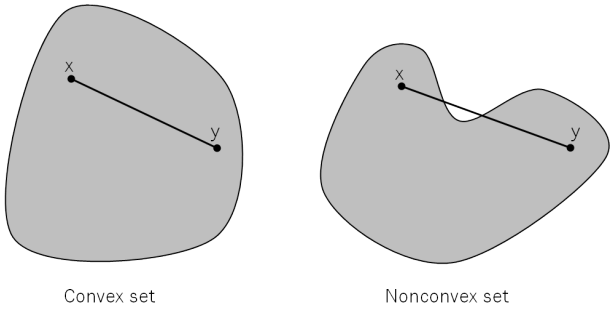

Figure 2.1: Convex and non-convex set

Example 2.2.9 *The set $\{\begin{pmatrix} x_0 & x \end{pmatrix} \in \mathbb{R}^{n+1} : \|x\|_k \leq x_0\}$ is convex. The proof is straightforward by using the triangle inequality for the norm $\|.\|_k$.*

An ellipsoid (see Def. 2.2.65) is a convex set.

Let S_1 and S_2 be two convex sets such that $S_1 \cap S_2 = \emptyset$. Then $S_1 \cup S_2$ is not convex.

Proposition 2.2.10 *The intersection of a collection of convex sets is itself convex.*

Remark : as illustrated in Example 2.2.9, the union of a collection of convex sets might be not convex.

Proposition 2.2.11 *The projection of a convex set onto some of its coordinates is convex. More precisely, if $S \subset \mathbb{R}^n \times \mathbb{R}^m$ is convex, then $\{x \in \mathbb{R}^n : (x, y) \in S\}$ is convex.*

Example 2.2.12 *If S is convex, then its convex hull is himself.*

Definition 2.2.13 *Extreme point*
Let $\mathcal{C}$ be a convex subset of $\mathcal{R}$. An extreme point of $\mathcal{C}$ is a point $x \in \mathcal{C}$ such that

$$x = \lambda y + (1 - \lambda)z \text{ for } \lambda \in [0, 1], y, z \in \mathcal{C} \Rightarrow x = y \text{ or/and } x = z$$

In other words, an extreme point is a point that does not belong to the interior of any segment lying entirely in $\mathcal{C}$.

Theorem 2.2.14 *Carathéodory's theorem*
Let $\mathcal{C}$ be a compact convex subset of $\mathbb{R}^n$. Then there exists an integer $m \leq n + 1$ such that every point x of $\mathcal{C}$ may be written as a convex combination of m extreme points of $\mathcal{C}$.

Corollary 2.2.15 *Let $S \subset \mathbb{R}^n$. There exists an integer $m \leq n + 1$ such that each $x \in \text{conv}(S)$ may be written as a convex combination of m elements of affinely independent points in S.*

This theorem facilitates the characterization of $\text{conv}(S)$ since each point can be written as a convex combination of a given number of the points of S. This enables the following "explicit" characterization of $\text{conv}(S)$:

$$x \in \text{conv}(S) \quad \Leftrightarrow \quad \begin{cases} \sum_{i=1}^{n+1} \lambda_i x_i = x \\ x_i \in S, \ i = 1, ..., n + 1 \\ \lambda_i \in [0, 1], \ i = 1, ..., n + 1 \end{cases} \quad \text{has a solution}$$

This Carathéodory's theorem is also used to prove the following proposition :

Proposition 2.2.16 *In $\mathbb{R}^n$, the convex hull of a compact set is compact.*

Proposition 2.2.17 *Let $\mathcal{C}$ be a nonempty closed convex set, then for all $x \in \mathcal{R}$, x admits an unique nearest point x_0 in $\mathcal{C}$. Moreover, x_0 is the solution of the inequality*

$$(x - x_0)(y - x_0) \leq 0, \forall y \in \mathcal{C}($$

Consequently, for any nonempty closed convex set $\mathcal{C}$, we are able to define the function $p_{\mathcal{C}} : \mathcal{R} \to \mathcal{C}$ such that $p_{\mathcal{C}}(x) = x_0$. This function is called *projection* over $\mathcal{C}$.

Definition 2.2.18 *Projection over a convex set*
Let us consider a convex set $\mathcal{C} \subset \mathcal{R}^n$ and an element x of $\mathcal{R}$. The projection of x over $\mathcal{C}$ is the unique minimizer of $\min_{x_0 \in \mathcal{C}} \|x - x_0\|$.

We can extend this definition to the notion of projection onto a subspace of $\mathcal{R}$.

Definition 2.2.19 *Projection onto a subspace*
Let us consider two subspaces of $\mathcal{R}$ such that $\mathcal{R} = \mathcal{R}_1 \times \mathcal{R}_2$. Then the projection of an element $x = (x_1, x_2)$ of $\mathcal{R}$ onto $\mathcal{R}_1$ yields x_2.

Indeed, there is an isomorphism between the image of this projection and the image of the projection of $\mathcal{R}$ onto its subset $\{(x_1, x_2) \in \mathcal{R} : x_2 = \bar{x}_2\}$, for any $\bar{x}_2 \in \mathcal{R}_2$.

Definition 2.2.20 *Projection of a set onto a subspace*
Let us consider two subspaces of $\mathcal{R}$ such that $\mathcal{R} = \mathcal{R}_1 \times \mathcal{R}_2$. Then the projection of a subset $S \subset \mathcal{R}$ onto $\mathcal{R}_1$ is

$$\{x_1 \in \mathcal{R}_1 : (x_1, x_2) \in S \text{ for some } x_2 \in \mathcal{R}_2\}$$

Moreover, the following theorem shows that the unicity of the nearest point is a sufficient condition for convexity :

Theorem 2.2.21 *Motzkin's characterization of convex sets*
Let $\mathcal{C}$ be a nonempty closed set. Assume that for all $x \in \mathcal{R}$, x has an unique nearest point in $\mathcal{C}$, then $\mathcal{C}$ is convex.

Proposition 2.2.22 *If a properly supports the convex set $\mathcal{C}$ at x_0, then the relative interior of $\mathcal{C}$ does not meet the supporting hyperplane. That is, $a^T x > a^T x_0$ for all $x \in \text{rint}(\mathcal{C})$.*

Theorem 2.2.23 *Strong Separating Hyperplane Theorem Let $\mathcal{C}$ a nonempty closed convex subset of $\mathcal{R}$ and $x \in \mathcal{R} \setminus \mathcal{C}$. Then there exists an hyperplane H that strongly separates $\mathcal{C}$ and x.*

This theorem is crucial for convex optimization since it is used to guarantee the existence of a *separation oracle* within the Ellipsoid method. As a consequence, any convex optimization problem can be solved in polynomial time as soon as such a separation oracle can be computed in polynomial time.

Corollary 2.2.24
Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be disjoint nonempty closed convex subsets of $\mathcal{R}$. Then $\mathcal{C}_1$ and $\mathcal{C}_2$ can be separated. If, moreover, $\mathcal{C}_1$ is compact, then $\mathcal{C}_1$ and $\mathcal{C}_2$ can be strongly separated.

Definition 2.2.25 *Faces of a convex set*
Let $\mathcal{C}$ be a convex subset of $\mathcal{R}$. A convex subset $\mathcal{F}$ of $\mathcal{C}$ is a face of $\mathcal{C}$ whenever the following condition holds :

$$\lambda x + (1 - \lambda) y \in \mathcal{F} \text{ for some } \lambda \in]0, 1[, x, y \in \mathcal{C} \Rightarrow x, y \in \mathcal{F}$$

In other words, if a relative interior of the line segment between x and y lies in $\mathcal{F}$, then the whole line segment lies in $\mathcal{F}$. $\mathcal{C}$ and $\emptyset$ are called the *trivial faces* of $\mathcal{C}$ and faces of dimension 1 are the extreme points of $\mathcal{F}$.

Definition 2.2.26 *Exposed face*
A face is exposed if it is a set of the form $H \cap \mathcal{C}$ where H is a non trivial supporting plane of $\mathcal{C}$.

What makes these set interesting for optimization is the following theorem :

Theorem 2.2.27 *Let $\mathcal{C}$ be a convex set and f a linear function. Then the minimum of f over $\mathcal{C}$ are attained on exposed faces of $\mathcal{C}$, i.e.*

$$\{x \in \mathcal{C} : f(x) = \min_{x \in \mathcal{C}} f(x)\} \text{ is an exposed face of } \mathcal{C} \tag{2.1}$$

The same holds for the maximum.

Theorem 2.2.28 *Faces of intersection Theorem Let $\mathcal{C}_1, \mathcal{C}_2$ be convex subsets of $\mathcal{R}$. Then*

$$\mathcal{F} \text{ is a face of } \mathcal{C}_1 \cap \mathcal{C}_2 \quad \Leftrightarrow \quad \mathcal{F} = \mathcal{F}_1 \cap \mathcal{F}_2 \text{ for some } \mathcal{F}_i \text{ faces of} \mathcal{C}_i$$

2.2.3 Cones

Definition 2.2.29 *Cone*
$\mathcal{K} \subset \mathcal{R}$ is a cone if it is closed under positive scalar multiplication :

$$\forall \lambda > 0, x \in \mathcal{K} \Rightarrow \lambda x \in \mathcal{K}$$

Note that many authors define a cone as a nonempty set, closed under non-negative scalar multiplication, instead of positive scalar multiplication. This is the same as assuming that any cone contains the origin, which is not what we do here.

Theorem 2.2.30
$\mathcal{K}$ is a convex cone, i.e., a cone that is convex, if and only if it is closed under addition and positive scalar multiplication.

Example 2.2.31 *The following sets are famous convex cones :*

- *The non-negative orthant :* $\{x \in \mathbb{R}^n : x_i \geq 0, i = 1, ..., n\}$ *;*
- *The positive orthant :* $\{x \in \mathbb{R}^n : x_i \geq 0, i = 1, ..., n\}$ *;*
- *The second-order cone or Lorentz cone or ice-cream cone :* $\{(x_0, x) \in \mathbb{R}^{n+1} : \|x\| \leq x_0\}$.

Definition 2.2.32 *Pointed cone*
A cone $\mathcal{K}$ is pointed if $\mathcal{K} \cap \{x : -x \in \mathcal{K}\} = \{0\}$.

This definition implies that a pointed cone contains the origin but does not contain any straight line passing through the origin.

Definition 2.2.33 *Solid cone*
A cone $\mathcal{K}$ is said to be solid if $\text{int}(\mathcal{K}) \neq \emptyset$.

In other words, a cone $\mathcal{K}$ is solid if it is full dimensional.

Definition 2.2.34 *Proper cone*
A cone is said proper if it is convex, closed, pointed and solid.

Example 2.2.35 *$\mathbb{R}^n_+$ is a proper cone.*

Definition 2.2.36 *Ray*
The ray generated by a non-zero vector $x \in \mathcal{R}$ is the set $\{\lambda x : \lambda \geq 0\}$.

Clearly, every cone contains the whole ray R_x together with any of its non-zero elements x.

Definition 2.2.37 *Extreme ray in $\mathcal{K}$*

Given a closed convex cone $\mathcal{K}$, a ray R is called extreme (in $\mathcal{K}$) if $\forall x, y \in \mathcal{K}, x + y \in R \Rightarrow x, y \in R$

Proposition 2.2.38 *Every closed pointed cone can be generated by a positive combination of its extreme rays.*

By a slight abuse of language, we say that a cone K is *generated* by some vectors if it is the set of positive combination of these vectors.

Theorem 2.2.39 *Carathéodory's theorem for cones*
Let $S \subset \mathcal{R}$. There exists an integer $m \leq n$ such that each $x \in \text{cone}(S)$ may be written as a nonnegative combination of m elements of affinely independent elements of S. In particular, $m \leq n+1$.

Definition 2.2.40 *Dual cone*

Let S be a subset of $\mathcal{R}$. Then $S^* = \{y \in \mathcal{R} : y^T x \geq 0, \forall x \in S\}$ denotes its dual cone.

Example 2.2.41 *The dual of the cone $\mathcal{C}_p = \{x \in \mathbb{R}^n : \|x_1, ..., x_{n-1}\|_p \leq x_n\}$ is the cone $\mathcal{C}_q$ with $1/p + 1/q = 1$.*

Proposition 2.2.42 *Let S be a subset of $\mathcal{R}$. Then S^{**} is the closure of the smallest convex cone containing S.*

Theorem 2.2.43 *Properties of the dual cone Let S be a nonempty subset of $\mathcal{R}$. Then*

(i) S^* is a closed convex cone ;

(ii) If S is a closed convex cone, then $S^{**} = S$

(iii) If S is solid ($\text{int}(S) \neq \emptyset$), then S^* is pointed;

(iv) S is a proper cone if and only if S^* is a proper cone.

Proposition 2.2.44 *With $\mathcal{C}_1, \mathcal{C}_2 \subset \mathcal{R}$ two convex cones, the following properties holds :*

- $\mathcal{C}_1^{**} = \text{cl}(\mathcal{C}_1)$;
- $\mathcal{C}_1 \subset \mathcal{C}_2 \Rightarrow \mathcal{C}_2^* \subset \mathcal{C}_1^*$;
- $(\mathcal{C}_1 + \mathcal{C}_2)^* = \mathcal{C}_1^* \cap \mathcal{C}_2^*$;

Furthermore, if $\mathcal{C}_1, \mathcal{C}_2$ are closed, such that $\text{rint}(\mathcal{C}_1) \cap \text{rint}(\mathcal{C}_2) \neq \emptyset$, then $(\mathcal{C}_1 \cap \mathcal{C}_2)^* = \mathcal{C}_1^* + \mathcal{C}_2^*$

Definition 2.2.45 *Recession cone*
Let S be a subset of $\mathcal{R}$ and x a point of S. Then $S^R(x) = \{d \in \mathcal{R} : x + \lambda d \in \mathcal{P}, \forall \lambda \geq 0\}$ is the recession cone of S at x. The nonzero elements of $S^R(x)$ are called the rays of S.

The recession cone is the set of all directions along which we can move indefinitely from x and still be in S.

Proposition 2.2.46 *A set is bounded if and only if its recession cone at any point is trivial, i.e., it contains only* 0.

Definition 2.2.47 *Simplicial cone*
A cone $K \subset \mathbb{R}^n$ is called a simplicial cone if it is generated by a finite number of linearly independent vectors.

Clearly, $\dim(K)$ equals the number of the generating vectors.

2.2.4 Polyhedra and polytopes

Polyhedra and polytopes play a key role in optimization as they are the geometrical representation of a feasible region expressed by linear constraints.

Definition 2.2.48 *Polyhedron*
A set $\mathcal{P} \subset \mathcal{R}$ is a polyhedron if it can be expressed as $P = \{x \in \mathcal{R} : a_i^T x \leq b_i,\ i = 1, ..., m\}$.

A polyhedron can be viewed as the intersection of of a finite number of halfspace of $\mathcal{R}$ and is therefore convex. As such, it admits some extreme points, that are also called *vertices*.

Definition 2.2.49 *Polytope*
A set P is called a polytope if it is the convex hull of a finite number of elements of $\mathcal{R}$.

We will see later that a set is a polytope if and only if it is a bounded polyhedron.

Example 2.2.50 *A simplex S in $\mathcal{R}$ is the convex hull of a set of d affinely independent vectors of $\mathcal{R}$. It is a polytope of dimension n. In particular, the standard simplex $S = \{x \in \mathbb{R}_+^n : e^T x = 1\}$ is the convex hull of the vectors $\{e_i\}_{i=1,..,n}$.*

Definition 2.2.51 *Polyhedral cone*
A set that is both a cone and a polyhedron is called a polyhedral cone. It can be represented in the form $P = \{x \in \mathcal{R} : a_i^T x \geq 0, i = 1, ..., m\}$.

Proposition 2.2.52 *Let $\mathcal{P} = \{x \in \mathcal{R} : a_i^T x \leq b_i,\ i = 1, ..., m\}$ a polyhedron of $\mathcal{R}$ and x a point of $\mathcal{P}$. Then the recession cone of $\mathcal{P}$ at x can be formulated as following :*

$$\mathcal{P}^R(x) = \{d \in \mathcal{R} : a_i^T d \leq 0,\ i = 1, ..., m\}$$

The recession cone of a polyhedron is therefore a polyhedral cone and is independent of the considered point x. Therefore, it is denoted by $\mathcal{P}^R$ and its elements are called the rays of $\mathcal{P}$. For a polytope, the recession cone is trivial, by applying Prop. 2.2.46.

Proposition 2.2.53 *$r \in \mathcal{P}^R$ is an extreme ray of the polyhedron $\mathcal{P} \subset \mathcal{P}$ if it is nonzero and if there are $n - 1$ linearly independent constraints binding at r.*

The following theorem is the basis for polyhedral combinatorics :

Theorem 2.2.54 *Minkowski-Weyl main theorem for polyhedra*
A polyhedron $P \subset \mathbb{R}^n$ can be represented as

$$P = \text{conv}(V) + \text{cone}(R)$$

for finite set $V, R \subset \mathbb{R}^n$. In particular, if P is pointed, V is the set of extreme points (vertices) of P and R is the set of extreme rays of P.

Conversely, if V and R are finite subsets of $\mathbb{R}^n$, then there exists a matrix $A \in \mathbb{R}^{m,n}$ and a vector $b \in \mathbb{R}^m$ for some m such that :

$$\text{conv}(V) + \text{cone}(R)$$

Such a representation of a polyhedron is called *canonical representation*.

This theorem gives rise to two corollaries, with $V = \{0\}$ or $R = \{0\}$. With $R = \{0\}$, which is equivalent to impose that the polyhedra be bounded, then it is a polytope :

Corollary 2.2.55 *A set is a polytope if and only if it is a bounded polyhedra.*

In other words, a polytope can be represented in two ways : either as the convex hull of a set of vertices ($\mathcal{V}$-representation), or as the intersection of half-spaces ($\mathcal{H}$-representation). In theory, we can always convert from one representation to another. In particular, being able to determine the polyhedral representation of a polytope is very desirable for discrete optimization.

In practice, some algorithms were designed to perform this conversion. The most famous is maybe the double description method, initially proposed in 1953 [199], that can also be considered as a constructive proof of the Minkowski-Weyl theorem. Let us cite also the Fourier-Motzkin elimination that can be used in this framework but not very efficiently, and a more recent algorithm, called *backtrack* [99], easier to implement. These algorithms have resulted in numerous software such as *PORTA* or *Polymake*.

This may be very useful for discrete optimization but we must bear in mind that a "small" $\mathcal{V}$-representation can lead to a $\mathcal{H}$-representation involving a huge number of inequalities and vice-versa. For example, a d-cube have $2d$ facets and 2^d vertices.

Corollary 2.2.56 *Every pointed polyhedral cone is the conic hull of its (finitely many) extreme rays.*

In other words, a cone $\mathcal{C}$ is polyhedral, i.e., $\exists A \in \mathbb{R}^{n,m} : \mathcal{C} = \{x : Ax \geq 0\}$) if and only if it is finitely generated, i.e., $\exists B \in \mathbb{R}^{k,n} :: \mathcal{C} = \{\lambda B : \lambda \in \mathbb{R}^k_+\}$.

Proposition 2.2.57 *A face of a polyhedron $\mathcal{P}$ is of the form $\{x \in \mathcal{P} : a^T x = b\}$ where $a^T x \leq b$ is some valid inequality of $\mathcal{P}$.*

For a polyhedron $\mathcal{P}$, faces of dimension 1 are called *edges* and face of dimension $\dim(\mathcal{P}) - 1$ are called *facets*.

The following theorem plays a central role in linear programming.

Theorem 2.2.58 *Let us consider a polyhedron $\mathcal{P} \subset \mathcal{R}$.*

If $\max\{c^T x : x \in \mathcal{P}\}$ is finite then there is an optimal solution that is an extreme point of $\mathcal{P}$.

If $\max\{c^T x : x \in \mathcal{P}\}$ is unbounded then $\mathcal{P}$ has an extreme ray r^* such that $c^T r^* > 0$.

Proposition 2.2.59 *An inequality is valid for a polyhedron P if and only if it is either equivalent or dominated by a conic combination of inequalities defining P.*

2.2.4.1 Projection

We consider a polyhedron $P = \{(x, y) \in \mathbb{R}^{p+q} : Ax + By \leq c\}$. Let recall that its projection onto the x-space is :

$$\text{Proj}_x(P) = \{x \in \mathbb{R}^p : (x, y) \in P \text{ for some } y \in \mathbb{R}^q\}$$

The objective is to find a polyhedral representation of $\text{Proj}_x(P)$. Two methods can be used for this :

- The Fourier-Motzkin elimination, a mathematical algorithm for eliminating variables from a system of linear inequalities. Here, we aim at eliminating the variables y.
- The Balas-Pulleyblank elimination [22], where several variables are eliminated at a time.

To proceed to the Balas-Pulleyblank elimination, we define the so-called *projection cone*.

Definition 2.2.60 *Projection cone*
Let us consider a polyhedron $P = \{(x, y) \in \mathbb{R}^{p+q} : Ax + By \leq c\}$. *Its projection cone associated with* x *is the following polyhedral cone :*

$$W = \{u : uB = 0, u \geq 0\}$$

In other words, W contains all the positive combinations of the rows of B that vanish. This allows to state the following theorem, which is fundamental for Lift & Project.

Theorem 2.2.61

$$\text{Proj}_x(P) = \{x \in \mathbb{R}^p : uAx \leq uc, y \in \text{ext}(W)\}$$

where $\text{ext}(W)$ *denotes the set of extreme rays of* W.

2.2.4.2 Polarity

The following notion of *polar* set is closely related to the dual cone.

Definition 2.2.62 *Polar*
Let S *be a subset of* $\mathbb{R}^n$. *Then its polar is the set* $S^\circ = \{\left(\pi_0, \pi^T\right)^T \in \mathbb{R}^{n+1} : \pi^T x \leq \pi_0, \forall x \in S\}$.

In other words, the elements of the polar corresponds to all the valid linear inequality over S. This set is closed by nonnegative combination, consequently S° is a convex cone. Furthermore, it is clear that $S^\circ = (\text{conv}(S))^\circ$.

Proposition 2.2.63 *Given a nonempty polyhedron* $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ *with* $\text{rank}(A) = n$, P° *is a polyhedral cone described by :*

$$\pi x^k - \pi_0 \leq 0, \text{ for } k \in K$$
$$\pi r^j \leq 0, \text{ for } j \in J$$

where $\{x^k\}_{k \in K}$ *and* $r^j_{j \in J}$ *are the extreme points and extreme rays of* P.

The following theorem is the major result on polarity.

Theorem 2.2.64
Given a nonempty polyhedron $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ *with* $\text{rank}(A) = n$ *and* $\dim(P) = n$, *then* $\begin{pmatrix}\pi_0 & \pi^T\end{pmatrix}^T$ *with* $\pi \neq 0$ *is an extreme ray of* P° *if and only if it defines a facet of* P.

2.2.5 Ellipsoid

Let us consider the unit sphere $\{x \in \mathbb{R}^n : x^T x \leq 1\}$ and its image by the affine transformation $x \mapsto y = Ax + a$ with A a positive definite matrix of $\mathbb{S}^n$. This leads to the definition of an ellipsoid.

Definition 2.2.65 *Ellipsoid*
Given a positive real ρ, *an affine mapping* $\Pi : \mathbb{R}^n \to \mathbb{R}^m$ *and a matrix* $Q \in \mathbb{R}^{l,n}$, *the ellipsoid* $\mathcal{E}_{\Pi,Q,\rho} \subset \mathbb{R}^m$ *is defined as* $\mathcal{E}_{\Pi,Q,\rho} = \{\Pi(x) : \|Qx\| \leq \rho\}$.

In particular, if A is a positive definite matrix and $a \in \mathbb{R}^m$, $\{y \in \mathbb{R}^m : (y-a)^T A^{-1}(y-a) \leq 1\} = \{Ax + a : x^T x \leq 1\}$ is an ellipsoid and a is its center.

Proposition 2.2.66 *The volume of the ellipsoid* $E(a, A)$ *is proportional to* $\det(A)$.

2.3 Linear Algebra

2.3.1 Matrices

Definition 2.3.1 *Matrix*
A matrix $A \in \mathbb{R}^{m,n}$ is an m-byn array of real numbers. If $n = m$ the matrix is said to be square.

Definition 2.3.2 *Submatrix*
A submatrix of a given matrix A is a matrix obtained by deleting rows and columns of A.

Definition 2.3.3 *Transpose*
The transpose of the matrix $A \in \mathbb{R}^{m,n}$, denoted A^T, is a matrix of $\mathbb{R}^{n,m}$ such that $A^T_{i,j} = A_{j,i}$.

Adding two matrices is straightforward with a component-wise addition. Likewise is defined the scalar multiplication. Regarding the product of two matrices, things are a little more complicated and there are three possibilities. The first one is the most usual one, denote matrix multiplication.

Definition 2.3.4 *Matrix multiplication*
Let $A \in \mathbb{R}^{k,m}$ and $B \in \mathbb{R}^{m,n}$. Then $AB \in \mathbb{R}^{k,n}$ is defined as $(AB)_{i,j} = \sum_{l=1}^{m} A_{i,l}B_{l,j}$.

Note that this product is not commutative, even if $k = n$. However it has some other good properties.

Proposition 2.3.5 *The matrix multiplication is :*

- *Associative : $(AB)C = A(BC)$;*
- *Distributive over matrix addition : $A(B + C) = AB + AC$;*
- *Scalar multiplication : $\lambda(AB) = (\lambda A)B = A(\lambda B)$;*
- *Commutative by transpose : $(AB)^T = B^T A^T$.*

The Hadamard product of two matrices of the same dimension is the component wise product.

Definition 2.3.6 *Hadamard product*
Let $A \in \mathbb{R}^{m,n}$ and $B \in \mathbb{R}^{m,n}$. Then the Hadamard product of A and B, denoted $A \circ B \in \mathbb{R}^{m,n}$ is defined as $(A \circ B)_{i,j} = A_{i,j}B_{i,j}$.

Finally, the Kronecker product applies to matrices of any dimension :

Definition 2.3.7 *Kronecker product*
Let $A \in \mathbb{R}^{k,l}$ and $B \in \mathbb{R}^{m,n}$. Then the Kronecker product of A and B, denoted $A \otimes B \in \mathbb{R}^{km,ln}$ is defined as $(A \otimes B)_{m(i-1)+i',n(j-1)+j'} = A_{i,j}B_{i',j'}$.

We are interested in the vector space $\mathbb{R}^{n,m}$ of real $n \times m$ matrices. Any element of $\mathbb{R}^{n,m}$ can be viewed as an element of $\mathbb{R}^{nm}$ and consequently, $\mathbb{R}^{n,m}$ is canonically embedded with an Euclidean structure, by importing the Euclidean structure of $\mathbb{R}^{nm}$. In particular it is endowed with the so-called *Frobenius* inner product.

Definition 2.3.8 *Frobenius inner product and Frobenius norm*
The Frobenius inner product is defined as following :$A \bullet B = \mathrm{Tr}(A^T B) = \mathrm{Tr}(B^T A) = \sum_{i=1}^{n}\sum_{j=1}^{m} A_{ij}B_{ij}$ for any $A, B \in \mathbb{R}^{n,m}$. For a matrix $A \in \mathbb{R}^{n,m}$, its associated Frobenius norm is $\|A\|_F = \sqrt{A \bullet A}$.

Proposition 2.3.9 *For $A \in \mathbb{R}^{n,m}, u \in \mathbb{R}^n, v \in \mathbb{R}^m$ we have $u^T A v = A \bullet uv^T$.*

Definition 2.3.10 *Rank*
Let us consider a matrix $A \in \mathbb{R}^{n,m}$. Its rank, denoted $\text{rank}(A)$ *is a nonnegative integer defined as the largest number of columns of A that constitutes a linearly independent set.*

Proposition 2.3.11 $\text{rank}(A^T) = \text{rank}(A)$

Consequently, the rank may equivalently be defined in terms of linearly independent rows.

Proposition 2.3.12 *For two matrices $A, B \in \mathbb{R}^{n,m}$,* $\text{rank}(AB) \leq \min\{\text{rank}(A), \text{rank}(B)\}$ *and* $\text{rank}(A+B) \leq \text{rank}(A) + \text{rank}(B)$.

Theorem 2.3.13
Let us consider a matrix $A \in \mathbb{R}^{n,m}$. There exists a factorization of the form $A = U\Sigma V^T$ where U and V are $n \times n$ and $m \times m$ unitary matrix (see Def. 2.3.37), and Σ is an $n \times m$ diagonal matrix with nonnegative real numbers on the diagonal. Such a factorization is called the singular value decomposition of A.

Definition 2.3.14 *Singular values of a matrix*
Let us consider a matrix $A \in \mathbb{R}^{n,m}$ and $A = U\Sigma V^T$ its singular value decomposition. The diagonal entries σ_i of Σ are called the singular values of A.

Proposition 2.3.15 *The rank of a matrix $A \in \mathbb{R}^{n,m}$ equals the number of its non-zero singular values.*

2.3.2 Linear mapping

Definition 2.3.16 *Linear mapping*
A mapping $f : \mathbb{R}^n \to \mathbb{R}^m$ is linear if it satisfies the condition of additivity ; $f(x)+f(y) = f(x+y), \forall x, y \in \mathbb{R}^n$ and homogeneity : $f(\lambda x) = \lambda f(x), \forall x \in \mathbb{R}^n, \lambda \in \mathbb{R}$.

Proposition 2.3.17 *Let f be a mapping from $\mathbb{R}^n$ to $\mathbb{R}^m$. Then there exists a matrix $A \in \mathbb{R}^{m,n}$ such that $f(x) = Ax, \forall x \in \mathbb{R}^n$.*

We will see that the properties of this mapping are closely linked with the properties of the matrix A.

Definition 2.3.18 *Null-space and range-space*
Let f be a linear mapping from $\mathbb{R}^n$ to $\mathbb{R}^m$. The null-space or kernel of f is the set $\mathcal{N}(f) = \{x \in \mathbb{R}^n : f(x) = 0\}$ and the range-space or image of f is the set $\mathcal{R}(f) = \{y \in \mathbb{R}^m : y = f(x)$ for some $x \in \mathbb{R}^n\}$.

By analogy, we denote by $\mathcal{N}(A) = \{x \in \mathbb{R}^n : Ax = 0\}$ the null-space of A and $\mathcal{R}(A) = \{y \in \mathbb{R}^m : y = Ax$ for some $x \in \mathbb{R}^n\}$ the range-space of A, for any matrix $A \in \mathbb{R}^{m,n}$.

Proposition 2.3.19 *Let us consider a matrix $A \in \mathbb{R}^{m,n}$. Then* $\dim(\mathcal{R}(A)) = \text{rank}(A)$

Theorem 2.3.20 *Rank theorem*
Let us consider a matrix $A \in \mathbb{R}^{m,n}$. The range-space of A^T and the null-space of A form a direct sum decomposition of $\mathbb{R}^n$: $\forall x \in \mathbb{R}^n, x = x_1 + x_2$ for some $x_1 \in \mathcal{N}(A), x_2 \in \mathcal{R}(A^T)$.

Furthermore $\dim(\mathcal{N}(A)) + \text{rank}(A) = n$.

This theorem leads to the following fundamental results.

Corollary 2.3.21 *Let us consider a matrix* $A \in \mathbb{R}^{n,m}$*:* $(\mathcal{R}(A^T))^\perp = \mathcal{N}(A)$

Corollary 2.3.22 *Let us consider two matrices* $A \in \mathbb{R}^{l,n}, B \in \mathbb{R}^{m,n}$*. We have the following equivalence :*

$$\mathcal{N}(A) \subset \mathcal{N}(B) \Leftrightarrow \exists M \in \mathbb{R}^{m,l} \text{ such that } B = MA$$

Let us consider a $m \times n$ matrix M of rank n. The QR-factorization of M finds an orthonormal m-by-m matrix Q and upper triangular m-by-n matrix R such that $M = QR$. If we define $Q = [Q1\ Q2]$, where $Q1$ is m-by-n and $Q2$ is m-by-$(m-n)$, then the columns of $Q2$ form an orthonormal basis of the null space of A^T.

2.3.3 Square matrices

From now on, we restrict our attention to square matrices.

Definition 2.3.23 *Nonsingular matrix*
A matrix $A \in \mathbb{R}^{n,n}$ *is said nonsingular or invertible if there exists a matrix in* $\mathbb{R}^{n,n}$*, denoted* A^{-1} *such that* $AA^{-1} = A^{-1}A = I$*, where* I *denotes the identity matrix of* $\mathbb{R}^{n,n}$*.*

Proposition 2.3.24 *Let us consider a matrix* $A \in \mathbb{R}^{n,n}$*. A is nonsingular if and only if its rank is equal to* n*. Otherwise it is said to be singular.*

Proposition 2.3.25 *A matrix* $A \in \mathbb{R}^{n,n}$ *is singular if and only if there exists* $x \in \mathbb{R}^n_*$ *such that* $Ax = 0$*.*

Definition 2.3.26 *Determinant*
Let A *be a square matrix of* $\mathbb{R}^{n,n}$*. Then the determinant of* A*, denoted* $\det(A)$ *is defined as :*

$$\det(A) = \sum_{\sigma \in \Sigma} (-1)^\sigma \prod_i A_i \sigma(i)$$

where Σ *is the set of the permutation of the indices* $\{1, ..., n\}$*.* $(-1)^\sigma = 1$ *or* -1 *depending on the permutation* σ *being odd or even.*

Generally, when n exceeds 3, the calculation of A from the definition is impractical, so we use a more practical method based on matrix decomposition.

Proposition 2.3.27 *Let us consider matrices* $A, B \in \mathbb{R}^{n,n}$*. Their determinants present the following characteristics*

- $\det(\lambda A) = \lambda \det(A^T)$
- $\det(AB) = \det(A)\det(B)$ *;*
- *Rows and columns can be interchanged without affecting the absolute value (but affecting the sign) of the determinant. In particular* $\det(A) = \det(A^T)$*.*
- *The determinant of a triangular matrix is the product of its diagonal entries ;*
- *The matrix* A *is nonsingular if and only if* $\det(A) \neq 0$ *;*

The following matrices are of great interest for Integer Linear Programming.

Definition 2.3.28 *Totally unimodular matrix*
A (possibly not square) matrix $A \in \mathbb{R}n, m$ is totally unimodular if every square submatrix has determinant 0, 1 or -1.

From the definition, it follows that any totally unimodular matrix has only 0, 1 or -1 entries. The interest for Integer Linear Programming comes from the following theorem.

Theorem 2.3.29 *Let $A \in \mathbb{R}^{n,m}$ be a totally unimodular matrix and $b \in \mathbb{Z}^m$ an integer vector. Then all the vertices of the polyhedron $\{x : Ax \leq b\}$ are integer.*

Definition 2.3.30 *Principal submatrix*
Let $A \in \mathbb{R}^{n,n}$ be a square matrix. Its principal submatrices are the submatrices obtained by removing k rows and the same k columns. Its leading submatrices are the principal submatrices obtained by removing the k last rows and columns.

Definition 2.3.31 *Minor*
A minor of a matrix $A \in \mathbb{R}^{n,m}$ is the determinant of a square submatrix of A. If A is square, its principal minors are the determinants of its principal submatrices and its leading principal minors are the determinants of its leading submatrices.

Proposition 2.3.32 *Factorization LU*
Let us consider a square nonsingular matrix $A \in \mathbb{R}^{n,n}$. Then there exists a unit lower triangular matrix $L \in \mathbb{R}^{n,n}$ and an upper triangular matrix $U \in \mathbb{R}^{n,n}$ such that $A = LU$.

Computing L and U can been carried out in $O(n^3)$ floating point operations.

Definition 2.3.33 *Eigenvalues and eigenvectors*
Let $A \in \mathbb{R}^{n,n}$. If $x \in \mathbb{R}^n$ and $\lambda \in \mathbb{C}$ satisfy $Ax = \lambda x$ then λ is called eigenvalues of A and x is called an eigenvector of A associated with λ.

Remark that A is nonsingular if and only if it does not admit 0 as eigenvalue.

Proposition 2.3.34 *Let us consider a square matrix $A \in \mathbb{R}^{n,n}$. The set of eigenvalues of A coincides with the root of the characteristic polynomial of A, defined by $p(x) = \det(xI - A)$.*

Definition 2.3.35 *Multiplicity*
The multiplicity of an eigenvalue of a matrix $A \in \mathbb{R}^{n,n}$ is the multiplicity of the eigenvalue as a zero of the characteristic polynomial of A.

This definition of multiplicity is also known as the *algebraic multiplicity*. It is equal or larger than the *geometric multiplicity* of an eigenvalue λ defined as the maximal number of linearly independent eigenvectors associated with λ.

Each matrix $A \in \mathbb{R}^{n,n}$ has, among the complex numbers, exactly n eigenvalues, counting multiplicities. Consequently, we denote by $\{\lambda_k(A)\}_{k=1,\ldots,n}$ the set of eigenvalues ranked in increasing order.

Proposition 2.3.36 *Let us consider a square matrix of $A \in \mathbb{R}^{n,n}$. Then*

$$\sum_{i=1}^{n} \lambda_i(A) = \mathrm{Tr}(A) \qquad \prod_{i=1}^{n} \lambda_i(A) = \det(A)$$

Definition 2.3.37 *Unitary matrix*
Let us consider a square matrix of $A \in \mathbb{R}^{n,n}$. Then A is unitary if $A^T A = AA^T = I$.

2.3.4 Symmetric matrices

We provide here some famous results of linear algebra concerning symmetric matrices or, more generally, *Hermitian* matrices. For sake of clarity, we restrict ourselves to real matrices here but we refer the reader to [139] for a more general exposure.

Definition 2.3.38 *Symmetric matrix*
A square matrix $A \in \mathbb{R}^{n\times n}$ is symmetric if $A^T = A$.

The set of symmetric matrices of $\mathbb{R}^{n\times n}$ is a vector space denoted $\mathbb{S}^n$.

Proposition 2.3.39 *For $A, B \in \mathbb{S}^n$, the following statements holds :*
- *$\lambda A + \mu B \in \mathbb{S}^n$ for any real scalars λ, μ, i.e., $\mathbb{S}^n$ is closed under linear combination ;*
- *AB is symmetric if and only if A and B commute and $\mathbb{S}^n$ is not closed under matrix product ;*
- *$AA^T = A^TA = (AA^T)^T \in mathbbS^n$;*
- *If A is nonsingular, $A^{-1} \in \mathbb{S}^n$;*
- *For any matrix $M \in \mathbb{R}^{n,n}$, $\frac{1}{2}(M + M^T) \in \mathbb{S}^n$ is known as the symmetric part of M.*

The following theorem and its corollaries are fundamental for the acquaintance of the area.

Theorem 2.3.40 *Spectral theorem for symmetric matrices*
A square matrix $A \in \mathbb{R}^{n,n}$ is symmetric if and only if there exists orthonormal matrix $U \in \mathbb{R}^{n\times n}$ ($U^{-1} = U^T$) such that :

$$A = U\Lambda U^T$$

where Λ is a real diagonal matrix containing the eigenvalues of A and the column-vectors $\{U_{*,j}\}_{j=1,\dots,n}$ of U are the eigenvectors of A : $AU_{*,j} = \Lambda_{jj}U_{*,j}$.

This decomposition is known as the *eigenvalue factorization* of A.

Corollary 2.3.41 *All the eigenvalues of a symmetric matrix are real.*

Corollary 2.3.42 *Let A a square matrix of $\mathbb{R}^{n,n}$. If A is symmetric, then A has a set of n eigenvectors that forms an orthonormal basis of $\mathbb{R}^n$.*

Corollary 2.3.43 *A can be written as a linear combination of the rank 1 matrices $v_iv_i^T$:*

$$A = \sum i = 1^n \Lambda_{ii} U_{i,*} {U_{i,*}}^T$$

Corollary 2.3.44 *A symmetric matrix $A \in \mathbb{S}^n$ has rank 1 if and only if there exists a vector $v \in \mathbb{R}^n$ and a real λ such that $A = \lambda vv^T$.*

Corollary 2.3.45 *Let A be a symmetric matrix and $A = U\Lambda U^T$ its eigenvalue factorization. Then $\mathcal{N}(A) = \mathcal{N}(U)$.*

The following results are known as the variational characterization of the eigenvalues of a symmetric matrix. There are crucial for the relationship between Semidefinite Programming and eigenvalue optimization.

Theorem 2.3.46 *Rayleigh-Ritz Theorem*
Let $A \in \mathbb{S}^n$ be a symmetric matrix. Then

$$\begin{array}{l}\lambda_1(A)x^Tx \leq x^TAx \leq \lambda_n(A)\\ \lambda_1 = \min_{x\neq 0} \frac{x^TAx}{x^Tx} = \min_{x^Tx=1} x^TAx\\ \lambda_n = \max_{x\neq 0} \frac{x^TAx}{x^Tx} = \max_{x^Tx=1} x^TAx\end{array}$$

Theorem 2.3.47 *Courant-Fisher Theorem*
Let $A \in \mathbb{S}^n$ be a symmetric matrix and k an integer such that $1 \leq k \leq n$. Then

$$\begin{array}{l}\min\limits_{u_1,\cdots,u_{n-k}\in\mathbb{R}^n} \max\limits_{x\in\mathbb{R}^n, x\neq 0, x\perp u_1,\cdots,u_{n-k}} \frac{x^TAx}{x^Tx} = \lambda_k(A)\\ \max\limits_{u_1,\cdots,u_{k-1}\in\mathbb{R}^n} \min\limits_{x\in\mathbb{R}^n, x\neq 0, x\perp u_1,\cdots,u_{k-1}} \frac{x^TAx}{x^Tx} = \lambda_k(A)\end{array}$$

Theorem 2.3.48 *Weyl Theorem*
Let A and B be symmetric matrices of $\mathbb{S}^n$ and k an integer such that $1 \leq k \leq n$. Then

$$\lambda_k(A) + \lambda_1(B) \leq \lambda_k(A+B) \leq \lambda_k(A) + \lambda_n(B)$$

2.3.5 Farkas' lemma

This section was inspired by [212] that contains a in-depth review of the Farkas' Lemma. Its most general form is the following theorem for convex functions :

Theorem 2.3.49 *Farkas' theorem*
Let $g_0, g_1, ..., g_m : \mathbb{R}^n \to \mathbb{R}$ be convex functions, $\mathcal{C} \subset \mathbb{R}^n$ a convex set and let us assume that there exists $\bar{x} \in \text{rint}(\mathcal{C})$ such that $g_1(\bar{x}) < 0, ..., g_m(\bar{x}) < 0$. Then one and only one of the following system has a solution :

$$(S_1)\begin{cases} g_0(x) < 0\\ g_i(x) \leq 0,\ i = 1, ..., m\\ x \in \mathcal{C}\end{cases} \qquad (S_2)\begin{cases} g_0(x) + \sum_{i=1}^m y_i g_i(x) \geq 0, \forall x \in \mathcal{C}\\ y_1, ..., y_m \geq 0\end{cases}$$

Applying this theorem to linear function yields the following theorem.

Theorem 2.3.50 *Theorem of the alternatives*
Let A be a real $n \times m$-matrix and b a real n-vector. Then one and only one of the following system has a solution :

$$(S_1)\ Ax \leq b \qquad (S_2)\begin{cases} A^Ty = 0\\ b^Ty < 0\\ y \geq 0\end{cases}$$

We provide some other "alternatives" that can be derived from this theorem :

Corollary 2.3.51 *Let A be a real $n \times m$-matrix and b a real n-vector. Then, for the following pairs of system, one and only one system has a solution :*

- $(S_1)\begin{cases} x \geq 0\\ Ax \leq b\end{cases} \qquad (S_2)\begin{cases} A^Ty \leq 0\\ b^Ty > 0\\ y \leq 0\end{cases}$
- $(S_1)\begin{cases} x \geq 0\\ Ax = b\end{cases} \qquad (S_2)\begin{cases} A^Ty \leq 0\\ b^Ty > 0\end{cases}$

– $(S_1)\left\{\begin{array}{l} x>0 \\ Ax\leq 0 \end{array}\right. \qquad (S_2)\left\{\begin{array}{l} A^Ty<0 \\ y\leq 0 \end{array}\right.$

– $(S_1)\left\{\begin{array}{l} c^Tx>d \\ Ax\leq b \end{array}\right. \qquad (S_2)\left\{\begin{array}{l} A^Ty=c \\ d-b^Ty\geq 0 \\ \\ y\geq 0 \end{array}\right.$

This lemma is very useful for getting necessary and sufficient conditions for a linear constraint $c^Tx \leq d$ being valid on the polyhedron $\mathcal{P} = \{x \in \mathbb{R}^m : Ax \leq b\}$. Indeed, this constraint is valid if and only if the system $\{Ax \leq b;\ -c^Tx < -d\}$ admits no solution, and therefore it is valid if and only if c is a positive combination of rows of A, and d is greater than the corresponding combination of the components of b. Another formulation for these alternatives is given by means of implication and render this interpretation more comprehensive :

$$\begin{array}{l}\big(A_i^Tx-b_i\leq 0,\ i=1,...,n \quad \Rightarrow \quad c^Tx-d\leq 0\big) \\ \qquad\qquad \Leftrightarrow \big(c^Tx-d=y_0+\sum_{i=1}^n y_i(A_i^Tx-b_i) \text{ for some } y_i\geq 0,\ i=0,...,n\big)\end{array}$$

Applying this result with $(S_2) = Ad \leq 0, e_i^Td > 0$ for all indices $i = 1, ..., n$ leads to the following corollary :

Corollary 2.3.52 *The polyhedron $\mathcal{P} = \{x \geq 0 : Ax \leq b\}$, with $A \in \mathbb{R}^{n\times m}$ and $b \in \mathbb{R}^n$ is unbounded if and only if there exists $d > 0$ such that $Ad \leq 0$.*

The Farkas theorem can also be applied to convex quadratic function. With $\mathcal{C} = \mathbb{R}^n$ and $g_i(x) = \tilde{x}^TQ_i\tilde{x}$, such that $\tilde{x_0}^TQ_i\tilde{x_0} < 0,\ i = 1, ...m$ for some $x_0 \in \mathbb{R}^n$, then

$$\left[\forall x\in\mathbb{R}^n,\ \ \tilde{x}^TQ_i\tilde{x}\leq 0,\ i=1,...,m \Rightarrow \tilde{x}^TQ_0\tilde{x}\geq 0\right] \Leftrightarrow \left[Q_0+\sum_{i=1}^m y_iQ_i \succcurlyeq 0 \text{ for some } y\in\mathbb{R}^m_+\right]$$

However, whenever one of the function is not convex, only the left part ($\Leftarrow$) remains true. The S-lemma (see 3.1.2.3) re-establish the equivalence for $m = 1$.

Finally, the Farkas' lemma was extended to a linear system involving matrix.

Lemma 2.3.53 *Semidefinite version of Farkas' lemma Let us consider a collection of $n+1$ symmetric matrices $A_1, ..., A_m \in \mathbb{S}^n$. Then one and only one of the following systems has a solution :*

$$(S_1)\left\{\begin{array}{l} A_i\bullet X=0,\ i=1,...,m \\ X\succcurlyeq 0,\ X\neq 0 \end{array}\right. \qquad (S_2)\ \sum_{i=1}^n y_iA_i\succ 0$$

2.4 Multivariate functions

2.4.1 Continuity

We are interested here in multivariate functions, that is functions from $\mathbb{R}^n \mapsto \mathbb{R}^m$, determined by its m real-valued components $f_i : \mathbb{R}^n \mapsto \mathbb{R}$. For this reason, we will often restrict our attention to real-valued functions.

Definition 2.4.1 *Image*
Let us consider a function $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}^m$ and a set $S \subset \Omega$. The image of S under f, denoted $f(S)$, is the set of output of f over S, i.e.,

$$f(S) = \{y \in \mathbb{R}^m : y = f(x) \text{ for some } x \in S\}$$

Definition 2.4.2 *Limit of a function*
Let us consider a function $f : \Omega \subset \mathbb{R}^n \mapsto \mathbb{R}^m$ and $x_0 \in \Omega$, $y_0 \in \mathbb{R}^m$. The limit of f at x_0 equals y_0, denoted $\lim_{x \to x_0} f(x) = y_0$ if for every $\varepsilon > 0$, there exists $\delta > 0$ such that :

$$\|x - x_0\| < \delta \Rightarrow \|f(x) - y_0\| < \varepsilon$$

Definition 2.4.3 *Continuous function*
A function $f : \Omega \subset \mathbb{R}^n \mapsto \mathbb{R}^m$ is said continuous at a point $x_0 \in \Omega$ if $\lim_{x \to x_0} f(x) = f(x_0)$.

It can be easily proved that sums, differences, products, quotients (under non-zero conditions), and compositions of continuous multivariate functions are continuous.

Let recall that continuity is required in the Theorem 2.1.27, which is central in optimization since it gives condition for attaining minimum and maximum of a real-valued function over a compact set.

A weaker property, called semi-continuity (or semicontinuity) may sometimes be sufficient.

Definition 2.4.4 *Semi-continuity*
Let us consider a function $f : \Omega \subset \mathbb{R}^n \mapsto \bar{\mathbb{R}}$ and $x_0 \in \Omega$. f is upper (resp. lower) semi-continuous at x_0 if for every $\varepsilon > 0$, there exists a neighborood S of x_0 such that $f(x) \leq f(x_0) + \varepsilon$ (resp. $f(x) \leq f(x_0) - \varepsilon$) $\forall x \in S$.

f is upper (resp. lower) semi-continuous if it is (resp. lower) semi-continuous at every point of its domain.

Proposition 2.4.5 *Let us consider a function $f : \Omega \subset \mathbb{R}^n \mapsto \mathbb{R}$. f is continuous at x_0 if and only if it is lower and upper semi-continuous at x_0.*

The semi-continuity is sufficient for a weaker variant of the Weierstrass theorem.

Proposition 2.4.6 *A lower semi-continuous function on a compact set attains its minimum and an upper semi-continuous function on a compact set attains its maximum.*

Definition 2.4.7 *Coercive function*
Let us consider a function $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$. f is coercive if $\lim_{\|x\| \to +\infty} f(x) = +\infty$.

2.4.2 Differentiability

Definition 2.4.8 *Directional derivative*
Let us consider a function $f : \Omega \subset \mathbb{R}^n \mapsto \mathbb{R}$, $x_0 \in \Omega$ and u a vector of $\mathbb{R}^n$. The directional derivative of f along the direction u at x_0 is :

$$\nabla_u f(x_0) = \lim_{h \to 0} \frac{f(x_0 + hu) - f(x_0)}{h}$$

Definition 2.4.9 *Partial derivative*
Let us consider a function $f : \Omega \subset \mathbb{R}^n \mapsto \mathbb{R}$, $x_0 \in \Omega$ and i an integer in $[n]$. The partial derivative of f with respect to x_i at the point x_0 is :

$$\frac{\partial f}{\partial x_i}(x_0) = \lim_{h \to 0} \frac{f(x_0 + he_i) - f(x_0)}{h}$$

Thus, the partial derivative w.r.t. index i is the directional derivative along e_i.

Proposition 2.4.10 *Let us consider a function $f : \Omega \subset \mathbb{R}^n \mapsto \mathbb{R}$, $x_0 \in \Omega$ and i, j two integers in $[n]$. Let denote g the function defined as the partial derivative of f w.r.t x_i : $g(x) = \frac{\partial f}{\partial x_i}(x)$. Then*

$$\frac{\partial f^2}{\partial x_i \partial x_j}(x_0) = \frac{\partial g}{\partial x_j}(x_0) = \frac{\partial f^2}{\partial x_j \partial x_i}(x_0)$$

By taking $i = j$ we get the twice partial derivative, also denoted $\frac{\partial f^2}{\partial x_i^2}(x_0)$.

Definition 2.4.11 *Jacobian*

Let us consider a function $f : \Omega \subset \mathbb{R}^n \mapsto \mathbb{R}^m$ and $x_0 \in \Omega$. The Jacobian of f at x_0 is the matrix $J \in \mathbb{R}^{n,m}$ such that $J_{i,j} = \frac{\partial f_j}{\partial x_i}(x_0)$.

With directional derivative, one require that the function admits a derivative along the directions u but there is not any requirement regarding the relationship between these derivatives. By requiring such a relationship, we get the notion of *Gâteaux-differentiability.*

Definition 2.4.12 *Gâteaux-differentiability*

Let us consider a function $f : \Omega \subset \mathbb{R}^n \mapsto \mathbb{R}^m$ and $x_0 \in \Omega$. f is Gâteaux-differentiable, or G-differentiable at x_0 if it admits a directional derivative along any direction $u \in \mathbb{R}^n$ and if the following application :

$$\begin{aligned} &\mathbb{R}^n \to \mathbb{R}^m \\ &u \mapsto \nabla_u f(x_0) \end{aligned}$$

is continuous and linear.

By denoting ∇f this mapping, we get $\nabla_u f(x_0) = \nabla f(x_0)^T u, \forall u \in \mathbb{R}^n$. Then $\nabla f(x_0)$ is the gradient of f at x_0.

The function f is called G-differentiable on Ω if it is G-differentiable at every $x \in \Omega$.

Definition 2.4.13 *Fréchet-differentiability*

Let us consider a function $f : \Omega \subset \mathbb{R}^n \mapsto \mathbb{R}^m$ and $x_0 \in \Omega$. f is said to be Fréchet-differentiable or F-differentiable at x_0 if there exists a linear function $L : \mathbb{R}^n \mapsto \mathbb{R}^m$, a function $\Phi : \mathbb{R}^n \mapsto \mathbb{R}^n$ and $\varepsilon > 0$ such that

$$\lim_{h \to 0} \Phi(h) = 0 \qquad f(x_0 + h) = f(x_0) + L(h) + \Phi(h)^T h, \text{ for } \|h\| \leq \varepsilon$$

When all of these things are so, the linear function L is called the derivative of f at x_0, written $D_f x_0$.

The function f is called F-differentiable on Ω if it is F-differentiable at every $x \in \Omega$.

The Fréchet-differentiability is stronger than the Gâteaux-differentiability, as shown in the following proposition.

Proposition 2.4.14 *Let us consider a function $f : \Omega \subset \mathbb{R}^n \mapsto \mathbb{R}^m$. If f is F-differentiable at $x_0 \in \Omega$ with a derivative $D_f x_0$, then f is G-differentiable at x_0 and $\nabla f(x_0)^T = D_f x_0$.*

Definition 2.4.15 *Continuously differentiable function*

Let us consider a function $f : \Omega \subset \mathbb{R}^n \mapsto \mathbb{R}^m$. If f is G-differentiable and its gradient is a continuous function, then f is said to be continuously differentiable.

Proposition 2.4.16 *If f is continuously differentiable, then f is F-differentiable.*

Consequently, for a continuously differentiable function, Fréchet and Gâteaux differentiabilities are equivalent and therefore, we don't need to distinguish these concepts.

Definition 2.4.17 *Partial derivative of order k*
Let us consider a function $f : \Omega \subset \mathbb{R}^n \mapsto \mathbb{R}^m$ and $k \in \mathbb{N}^n$. The partial derivative of f at $x_0 \in \Omega$ has the i-th component :

$$\frac{\partial^k f_i}{\partial x_1^{k_1} ... \partial x_n^{k_n}}(x_0)$$

Definition 2.4.18 *Smooth function*
Let us consider a function $f : \Omega \subset \mathbb{R}^n \mapsto \mathbb{R}^m$. For $k \in \mathbb{N}^n$, f is of class C^k if its partial derivative of order $k' \in \mathbb{N}^n$ exists and are continuous for each k' such that $\sum_{i=1}^n k'_i \leq k$.

f is smooth, or of class C^∞ if it is of class C^k for any $k \in \mathbb{N}^n$.

Definition 2.4.19 *Hessian*
Let us consider a function $f : \Omega \subset \mathbb{R}^n \mapsto \mathbb{R}$ of class C^2. Then the Hessian of f at $x_0 \in \Omega$, denoted $\nabla^2 f(x_0)$, is a matrix of $\mathbb{S}^n$ such that

$$\nabla^2 f(x_0)_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}(x_0)$$

Definition 2.4.20 *Hessian*
Let us consider a function $f : \Omega \subset \mathbb{R}^n \mapsto \mathbb{R}^m$. Its Jacobian matrix is the matrix $J_f(x) \in \mathbb{R}^{m,n}$ such that $J_f(x)_{i,j} = \frac{\partial f_j}{\partial x_i}(x)$.

Definition 2.4.21 *Sub-gradient*
Let us consider a function $f : \Omega \subset \mathbb{R}^n \mapsto \mathbb{R}$ and $x_0 \in \Omega$. A vector $v \in \mathbb{R}^n$ is a subgradient of f at x_0 if there exists a open convex set $S \subset \Omega$ containing x_0 such that :

$$f(x) - f(x_0) \geq v^T (x - x_0)$$

The set of all subgradients at x_0 is called subdifferential at x_0 and denoted $\partial \nabla f(x_0)$.

$\partial f(x_0)$ contains only one point if and only if f is G-differentiable at x_0.

2.4.3 Optimality

In this section, we restrict our attention to real-value functions : $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$.

Definition 2.4.22 *Infimum and supremum*
Let us consider a function $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$. The infimum (resp. supremum) of f over Ω is the least upper (resp. most lower) bound of the image of Ω under f. More formally, $\sup_{x\in\Omega} f$ (resp. $\inf_{x\in\Omega} f$) is the smallest (resp. largest) $y_0 \in \bar{\mathbb{R}}$ such that $y_0 \geq y$ (resp. $y_0 \geq y$) , $\forall y \in f(\Omega)$.

Definition 2.4.23 *Extremum*
Let us consider a function $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ and a point $x^* \in \Omega$. f admits a minimum (resp. maximium) over Ω at x^* if $f(x^*) \leq f(x)$ (resp.$f(x^*) \leq f(x)$) for all $x \in \Omega$.

An extremum is either a minimum or a maximum. By analogy with the infimum and supremum, we denote by $f(x^*) = \max_{x\in\Omega} f$ or $f(x^*) = \min_{x\in\Omega} f$.

Definition 2.4.24 *Neighbourhood*
Let us consider a set $S \subset \mathbb{R}^n$ and $x \in S$. S is a neighbourhood of x if it includes an open set U that contains x : $x \in U \subset S$.

Definition 2.4.25 *Local minimum and maximum*
Let us consider a function $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ and a point $x^* \in \Omega$. f admits a local minimum (resp. maximum) at x^* if there exists a neighbourhood U of x^* such that the restriction of f to U admits a minimum (resp. maximum) at x^*.

Definition 2.4.26 *Bouligand Tangent cone*
Let us consider a closed set $X \subset \mathcal{R}$. Its Bouligand tangent cone at x , denoted $\mathcal{T}_X(x)$ is defined as follows :

$$\mathcal{T}_X(x) = \{\lim_{i\to\infty} \frac{x_i - x}{t_i} : x_i \to x, t_i \to 0\}$$

This set contains all the direction d such that there exists a sequence $d_k \lim d$ and $y_k = \{y \in X : y = x + \lambda d_k\}$ tends to x when $k \to \infty$.

In the case where X is defined through a set of functional equality and inequality constraints, and if these constraints satisfy the so-called *constraints qualifications*, then this cone can be easily computed by using the gradient of the constraints active at x. In particular, if $X = \{x \in \mathbb{R}^n : Ax = b\}$ then $\mathcal{T}_X(x) = \mathcal{N}(A)$.

Theorem 2.4.27 *First-order optimality conditions Let us consider the following optimization problem : $(P) \min f(x) : x \in X$ where $X \subset \mathbb{R}^n$ is closed and $f : \mathbb{R}^n \to \mathbb{R}$.*

If x^* is a local minimum of (P) and if f is differentiable in x^*, then

$$\forall d \in \mathcal{T}_X(x^*), d^T \nabla f(x^*) \geq 0$$

with $\mathcal{T}_X(x^*)$ is the Bouligand tangent cone of X at x^*.

The following theorem is an application to the case where $X = \mathbb{R}^n$.

Theorem 2.4.28 *Let us consider a continuously differentiable function $f : \Omega \to \mathbb{R}$ with Ω an open subset of $\mathbb{R}^n$. If f admits a local extremum at $x^* \in \Omega$ then $\nabla f(x^*) = 0$.*

Note that this condition is necessary but not sufficient for local optimality. In the case when S is not open, the condition applies to $\text{int}(\Omega)$ but there might be some local optima on the boundary of Ω that may not satisfy this condition.

Definition 2.4.29 *Stationary point*
Let us consider a continuously differentiable function $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$. $x^* \in \Omega$ is a stationary point of f if $\nabla f(x^*) = 0$.

Definition 2.4.30 *Saddle-point*
Let us consider a continuously differentiable function $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$. A stationary point $x^* \in \Omega$ that is not a local extremum is called a saddle-point of f.

In particular, if $f : \Omega_1 \times \Omega_2 \to \mathbb{R}$, $(x^*, y^*) \in \Omega_1 \times \Omega_2$ is such that $f(x, y^*) \leq f(x^*, y^*) \leq f(x^*, y)$ for any $x \in \Omega_1, y \in \Omega_2$, then (x^*, y^*) is a saddle-point.

The name derives from the fact that in two dimensions the surface resembles a saddle that curves up in one direction and curves down in a different direction :

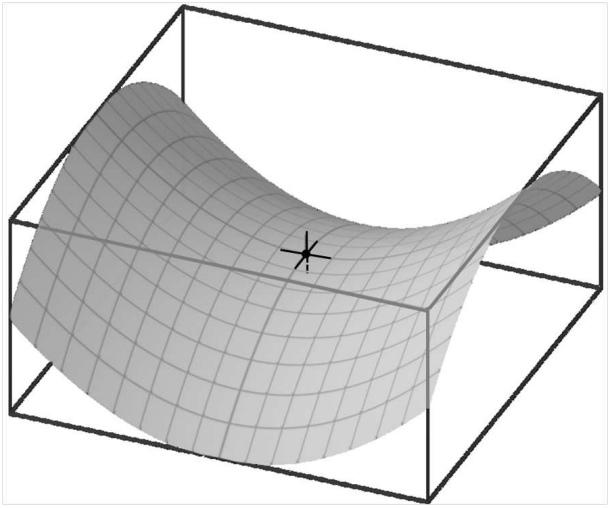

Figure 2.2: A saddle-point on the graph of $f(x, y) = x^2 - y^2$

Theorem 2.4.31 *Second-order conditions Let us consider a twice continuously differentiable function $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ and a stationary point $x^* \in \Omega$. If x^* is a local minimum of f, then $\nabla^2 f(x^*) \succcurlyeq 0$.*

Combined with the first-order condition $\nabla f(x^*) = 0$, $\nabla^2 f(x^*) \succ 0$ is sufficient for x^* to be a local minimum.

2.4.4 Convexity

Definition 2.4.32 *Epigraph*
The epigraph of a function $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$, denoted epi$f$*, is the subset of $\mathbb{R}^{n+1}$ of points lying on or above its graph, i.e. :*

$$\text{epi} f = \{(x, t) \ : \ x \in \Omega, f(x) \leq t\}$$

Definition 2.4.33 *Convex function*
A function $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ is convex if its epigraph is convex.

This implies that $\mathsf{dom} f$ is convex and that for all $x, y \in \mathsf{dom} f$, for all $\lambda \in [0, 1]$,

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$$

Proposition 2.4.34 *A function $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ is convex if and only if it is convex when restricted to any line that intersects its domain. In other words, if for any $x, y \in \mathrm{dom} f$, the function $g : \mathbb{R} \to \mathbb{R}$ with $g(t) = f(x + ty)$ is convex.*

Example 2.4.35 *A norm is a convex function. Indeed,*

$$\begin{aligned} ||\lambda x + (1 - \lambda)y|| &\leq ||\lambda x|| + ||(1 - \lambda)y|| (\textit{ by the triangular inequality}) \\ &= |\lambda|||x|| + |(1 - \lambda)|||y|| \\ &= \lambda||x|| + (1 - \lambda)||y|| \end{aligned}$$

Pictorially, the graph of a convex function "bends upward". More formallly, it lies below or on the straight line segment connecting two points, for any two points in the interval, as illustrated in the figure below :

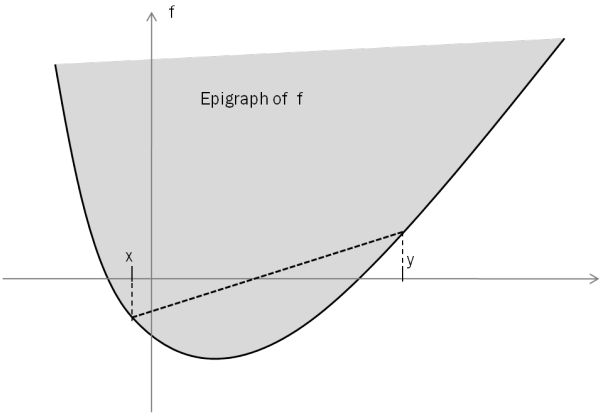

Figure 2.3: Convex function

Definition 2.4.36 *Concave function*
A function f is concave if the function $-f$ is convex.

Example 2.4.37 *A function f is affine : $f(x) = a_0 + \sum_{i=1}^{n} a_i x_i$ if and only if it is both convex and concave.*

2.4.4.1 Conditions for convexity

Proposition 2.4.38 *Pointwise maximum If $f_{i\,i=1,\ldots,k}$ are convex functions of Ω to $\mathbb{R}$ then their pointwise maximum f, defined above, is also convex :*

$$\begin{aligned} f : & \qquad \Omega \to \mathbb{R} \\ x \mapsto & \max_{i=1,..,k} f_i(x) \end{aligned}$$

Consequently, by noting that $-f = \min_{i=1,..,k} -f_i(x)$ is concave, the pointwise minimum of a set of concave functions is also concave.

Proposition 2.4.39 *Minimization Let us consider a fonction $f : \Omega_1 \times \Omega_2 \to \mathbb{R}$ which is convex w.r.t $(x, y) \in \Omega_1 \times \Omega_2$. Then, for any convex set $\mathcal{C} \subset \Omega_1$, $g(y) = \inf_{x \in \mathcal{C}} f(x, y)$ is convex w.r.t. y.*

Proposition 2.4.40 *Composition by affine mapping Let us consider a function $f : \Omega \to \mathbb{R}$. If f is convex (resp. concave), then the composition of f with the affine mapping $x \mapsto Ax+b$: $g : x \mapsto f(Ax+b)$ is convex (resp. concave).*

Proposition 2.4.41 *A set $S = \{x \in \mathbb{R}^n : f_i(x) \leq 0, i = 1, \cdots, m\}$ is convex if all the functions f_i are convex.*

The converse is not true. For example, the set $\{x \in \mathbb{R}^n : -x^2 + 1 \leq 0, x \leq 0\}$ is convex although the function that associates $-x^2 + 1$ to x is not convex. Let provide another example crucial for optimization.

Example 2.4.42 *Let $f(x) = \|Ax + b\| - c^T x - d$, f is convex, by sum and composition of the norm with an affine mapping. Consequently, the set $S = \{x \in \mathbb{R}^n : f(x) \leq 0\}$ is convex. It is clear that an equivalent definition of S is :*

$$\begin{aligned} S &= \{x \in \mathbb{R}^n : \|Ax + b\| \leq c^T x + d\} \\ &= \{x \in \mathbb{R}^n : \|Ax + b\|^2 \leq (c^T x + d)^2, \quad c^T x + d \geq 0\} \\ &= \{x \in \mathbb{R}^n : x^T (A^T A - cc^T)x + 2(b^T A - dc^T)x + b^2 - d^2 \leq 0, \quad c^T x + d \geq 0\} \end{aligned}$$

Generally, the function $x \mapsto x^T (A^T A - cc^T)x + 2(b^T A - dc^T)x + b^2 - d^2$ is not convex and yet S is always convex.

Proposition 2.4.43 *A convex function $f : \Omega \to \mathbb{R}$ is differentiable at $x_0 \in \Omega$ if and only if its subdifferential is made up of only one vector, which is the derivative of f at x_0.*

Proposition 2.4.44 *Let us consider a function $f : \Omega \to \mathbb{R}$ twice differentiable, i.e. such that its Hessian exists at each point of Ω. Then, f is convex if and only if $\mathrm{dom} f$ is convex and its Hessian is positive semidefinite at each $x \in \mathrm{dom} f$.*

The following theorem accounts for convexity being so important within optimization. It states that a local minimum of a convex function over a convex set is necessary a global minimum :

Theorem 2.4.45
Let us consider a convex function $f : \Omega \to \mathbb{R}$ and a convex set $\mathcal{S} \subset \Omega$. Given a point $x^* \in \mathcal{S}$, suppose there is a ball $\mathcal{B} \subset \mathcal{S}$ such that $f(x^*) \leq f(x), \ \forall x \in \mathcal{B}$. Then $f(x^*) \leq f(x), \ \forall x \in \mathcal{S}$.

In the case of a continuously differentiable function, the first-order optimality conditions are therefore sufficient.

Theorem 2.4.46 *Optimality condition of convex functions Let us consider a convex function $f : \Omega \to \mathbb{R}$ continuously differentiable. Then x^* is a global optimum of f if and only if $\nabla f(x^*) = 0$.*

In particular, quadratic functions $x \mapsto x^T Ax + b^T x + c$ have a constant Hessian A, so their convexity follows immediately from the positive semidefinitess of A. Otherwise, convexity may be very hard to recognize. Even for multivariable polynomials, deciding if the function is convex is NP-hard [5]. However, the weaker property of *quasi-convexity* is easier to recognize.

Definition 2.4.47 *Quasi-convexity and quasi-concavity*
Let us consider a function $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$. f is quasi-convex if its domain and all its sublevel set $\{x \in \text{dom}f : f(x) \leq \alpha\}$ for $\alpha \in \mathbb{R}$ are convex.

A function f is quasi-concave if $-f$ is quasi-convexe.

Example 2.4.48 *Let us consider an interval $[a, b]$ of $\mathbb{R}$. The indicator function $\mathbb{1}_{[a,b]}$ is quasi-concave on $\mathbb{R}$ but it is not concave.*

It is essential to note that the sum of two quasi-convex (or quasi-concave) functions is not necessarily quasi-convex (or quasi-concave). For instance $\mathbb{1}_{[a,b]} + \mathbb{1}_{[c,d]} = \mathbb{1}_{[a,b]\cup[c,d]}$ is not quasi-convex as soon as $[a, b]$ are $[c, d]$ disjoint.

Proposition 2.4.49 *$f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ is quasi convex if and only if* dom$f$ *is convex and for any $x, y \in \Omega$ and any scalar $0 < \lambda 1$*

$$f(\lambda x + (1\lambda)y) \leq \max\{f(x), f(y)\}$$

Proposition 2.4.50 *Convex functions are quasi-convex.*

Proposition 2.4.51 *Let us consider a subset S of $\mathcal{R}$ defined through m functions g_i : $S = \{x \in \mathcal{R} : g_i(x) \leq 0, i = 1, ..., m\}$. If all the functions g_i are quasi-convex, S is convex.*

Definition 2.4.52 *Convex lower envelope*
Let us consider a function $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$. Its convex lower envelope is the function $\hat{f}(x) = \inf\{t : (x, t) \in \text{conv}(\text{epi}f)\}$.

$\hat{f}$ is an underestimator of f over Ω, that is $\hat{f}(x) \leq f(x), \forall x \in \Omega$. The convex lower envelope is therefore the pointwise supremum of all the convex underestimator of f over Ω.

In the same way, we define an overestimator and the concave upper envelope which is the pointwise infimum of concave overestimator of f.

Example 2.4.53 *If f is the bilinear function $f(x, y) = xy$ over the rectangle $R = \{(x, y) : l_x \leq x \leq u_x, l_y \leq y \leq u_y\}$, then its convex lower envelope is $\max\{l_y x + l_x y - l_x l_y, u_y x + u_x y - u_x u_y\}$ and its concave upper envelope is $\max\{u_y x + l_x y - l_x u_y, l_y x + u_x y - l_y u_x\}$.*

Together, these two functions constitutes the so-called McCormick relaxation.

Proposition 2.4.54 *Let us consider a function $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ and its convex lower envelope $\hat{f}$. Then*

$$\hat{f}(x) = \max_v \{v^T x : v^T \hat{x} \leq f(\hat{x}), \forall \hat{x} \in \Omega\}$$

Example 2.4.55 *α-BB convex underestimator*
Let us consider a function $f : \Omega \subset \mathbb{R}^n$ and a rectangle R of Ω : $R = \{x \in \Omega : l_i \leq x_i \leq u_i\}$. An underestimator of f can be constructed by substracting a positive quadratic term from f : $f^{\leq}(x) = f(x) - \sum_{i=1}^n \alpha_i (u_i - x_i)(x_i - l_i)$. A necessary and sufficient condition for $f^{\leq}$ to be convex is that its Hessian, i.e., $\nabla^2 f(x) + \text{Diag}(\alpha)$ be positive semidefinite, for any $x \in \Omega$. This convex underestimator is very classical in the literature and is called α-BB convex underestimator.

Definition 2.4.56 *Log-concave function*
A function $f : \mathcal{R} \to \mathbb{R}$ is log-concave if its domain is a convex set and if it satisfies the inequality $f(\lambda x + (1 - \lambda)y) \geq f(x)^\lambda f(y)^{1-\lambda}$, for any $x, y \in$ dom$f$ and $0 < \lambda < 1$.

If f is strictly positive over dom$f$, then this definition is equivalent to requiring that the natural logarithm of f, $\ln(f)$, be concave. Note that any concave function that is nonnegative on its domain is log-concave, since ln is concave.

Proposition 2.4.57 *Let us consider a subset $S \subset \mathcal{R}$. If S is convex, then its indicator function $\mathbb{1}_S$ is log-concave.*

Proposition 2.4.58 *If $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is log-concave, then the following function g is log-concave.*

$$g(x) = \int f(x, y)dy$$

The following theorem results from a strong connexion between two-person zero-sum games and linear programming.

Theorem 2.4.59 *Von Neumann's theorem*
Let A be a $n \times m$ matrix, $X = \{x \in \mathbb{R}^n : e^T x = 1,\ x \geq 0\}$ and $Y = \{y \in \mathbb{R}^n : e^T y = 1,\ y \geq 0\}$. Then the quadratic function $x^T Ay$ admits a saddle-point over $X \times Y$, i.e.,

$$\min_{x \in X} \max_{y \in Y} x^T Ay = \max_{y \in Y} \min_{x \in X} x^T Ay$$

Its generalization leads to the following theorem.

Theorem 2.4.60 *Sion's minimax theorem Let us consider a function $f : X \times Y \to \mathbb{R}$ with*

- *X be a compact convex subset of a linear topological space ;*
- *Y a convex subset of a linear topological space ;*
- *$f(x, \cdot)$ upper semicontinuous and quasiconcave w.r.t. y on $Y, \forall x \in X$;*
- *$f(\cdot, y)$ is lower semicontinuous and quasi-convex w.r.t. x on $X, \forall y \in Y$.*

then,

$$\min_{x \in X} \sup_{y \in Y} f(x, y) = \sup_{y \in Y} \min_{x \in X} f(x, y)$$

The extension of the inequality in the basic definition of a convex function to integral leads to the so-called *Jensen's inequality* :

Theorem 2.4.61 *Let μ be a probability measure over the probability space $(\Omega, \mathcal{A}, \mu)$, f a convex function that is μ-integrable and ϕ a convex function on the real line :*

$$\phi(\int_\Omega g\mu(x)dx) \leq \int_\Omega \phi \bullet f\mu(x)dx \tag{2.2}$$

In particular, for a random variable X, this implies that $\phi(\mathrm{E}[X]) \leq \mathrm{E}[\phi(X)]$ for any convex function ϕ.

2.5 Polynomials

There is a strong relationship between SDP and polynomials. Within this section, we provide the necessary theoretical background to acquaint oneself with this area. In this context, we restrict ourselves to real multivariable polynomials, i.e. polynomials from $\mathbb{R}^n$ to $\mathbb{R}$.

2.5.1 Definition and notations

Definition 2.5.1
$\mathbb{N}^n_d = \{\kappa \in \mathbb{N}^n : \sum_{i=1}^n \kappa_i \leq d\}$ *contains* $b_n(d) = \binom{n+d}{d}$ *different elements.*

Definition 2.5.2 *Monomial*
The monomial associated to $\kappa \in \mathbb{N}^n_d$ *is the function* $\mathbb{R}^n \to \mathbb{R}$ *defined as* $x^\kappa = \prod_{i=1}^n x_i^{\kappa_i}$. *Its degree is* $\sum_{i=1}^n \kappa_i$.

Definition 2.5.3 *Polynomial*
A polynomial is a function $\mathbb{R}^n \to \mathbb{R}$ *defined as a weighted sum of monomials :* $p(x) = \sum_\kappa \mathsf{p}_\kappa x^\kappa$. *Its degree, denoted* $\deg(p)$ *is the largest degree of all its monomial and* p *is its coefficients vector.*

Note that polynomials are continuous and smooth functions.

Definition 2.5.4
The set of all polynomials in $x_1, ..., x_n$ *with real coefficients is written as* $\mathbb{R}[x_1, ..., x_n]$. $\mathbb{R}_d[x_1, ..., x_n]$ *contains the polynomials of* $\mathbb{R}[x_1, ..., x_n]$ *with degree at most* d.

$\mathbb{R}_d[x_1, ..., x_n]$ may be abbreviated as $\mathbb{R}_d[\mathsf{x}]$ where x stands for the n-tuples $(x_1, ..., x_n)$ when the dimension n is clear by the context. $\mathbb{R}_d[x_1, ..., x_n]$ is isomorphic to $\mathbb{R}^{b_n(d)}$ and is therefore a vector space of dimension $b_n(d)$. For example, a typical basis for $\mathbb{R}_d[x_1, ..., x_n]$, called *standard monomial basis* is :

$$u_d(x) = (1, x_1, \cdots, x_n, x_1^2, x_1x_2, \cdots, x_{n-1}x_n, x_n^2, x_1^3, \cdots, x_n^d)^T$$

and any polynomial $p \in \mathbb{R}_d[\mathsf{x}]$ is represented by its coefficient vector in this basis.

Definition 2.5.5 *Semi-algebraic set*
A semi-algebraic set $\mathcal{S}$ *is a subset of* $\mathbb{R}^n$ *defined by a finite sequence of polynomial inequalities :*

$$\mathcal{S} = \{x \in \mathbb{R}^n : p_i(x) \leq 0,\ i = 1, ..., m\}$$

where $p_i,\ \ i = 1, \cdots, m$ *are polynomials.*

Such a set is closed. For example, intervals, half-space or half-plane are some particular semi-algebraic sets.

Example 2.5.6 *An ellipsoid is a semi-algebraic set defined via one convex quadratic function.*

2.5.2 Positivity of polynomials over semi-algebraic sets

In this section, we provide theorems that give some conditions over the structure of a polynomial so that it satisfies a positive, nullity or non-negativity conditions over a semi-algebraic set. This theorems are known under the name of *positivstellensatz*, *nullstellensatz*,*nichtnegativstellensatz* respectively, from the German words *Stellen* (places) and *Satz* (theorem). The first such result was the Hilbert's nullstellensatz for complex numbers. In this section, we restrict out attention to results for real numbers, concerning rather positivity and nonnegativity.

Definition 2.5.7 *Positive and nonnegative polynomial*
A polynomial p *is positive (resp. nonnegative) on a semi-algebraic set* $\mathcal{S}$ *if* $p(x) > 0$ *(resp.* $p(x) \geq 0$*) for all* $x \in \mathcal{S}$.

A sufficient condition for $p(x) \geq 0$ on $\mathbb{R}^n$ is that p be a *sum of square*.

Definition 2.5.8 *Sum of square*
A polynomial p is a sum of square, denoted s.o.s. if there exists some polynomials $\{u_j\}_{j=1,...,m}$ such that $p = \sum_{j=1}^m u_j^2$.

Lemma 2.5.9 *If $p \in \mathbb{R}[x]$ is s.o.s. then* $\deg(p)$ *is even and any decomposition $p = \sum_{j=1}^m u_j^2$ satisfies* $\deg(u_j) \leq \deg(p)/2,\ j = 1, ..., m$.

The most general condition for non-negativity of a polynomial is the Sengle's Positivstellensatz. It can be seen as a generalization of Farkas lemma. It states that, for a system of polynomial equations and inequalities, either there exists a solution,or we can exhibit a polynomial identity ensuring that no solution exists. For sake of brevity, we only give the corollary of this theorem that is relative to non-negativity of polynomials.

Corollary 2.5.10 *Stengle's Positivstellensatz*
Let $F = \{f_j\}_{j=1,...,m}$ be a finite family of polynomials in $\mathbb{R}[\mathrm{x}]$ and $\mathcal{S} = \{x \in \mathbb{R}^n : f_j(x) \geq 0,\ j = 1, \cdots, m\}$ its associated semi-algebraic set. P denotes a set of polynomials built as combination of product of elements of F and s.o.s polynomials : $P = \{ \sum_{J \subset [m]} p_J f_J : p_J \in \mathbb{R}[\mathrm{x}],\ p_J$ s.o.s, $f_J = \prod_{j \in J} f_j\}$.

Then, for any polynomial f_0,

$$f_0(x) \leq 0 \text{ on } \mathcal{S} \Leftrightarrow \exists p, q \in P \text{ and } k \in \mathbb{N} \text{ such that } f_0 p + f_0^{2k} + q = 0$$

Thus, this theorem provides necessary conditions for a polynomial being positive or nonnegative over an arbitrary semi-algebraic set. However, these conditions involves a product ($f_0 p$) which render it impossible to exploit in practice.

For this reason, we present two other positivstellensatz, namely the Schmüdgen's and the Putinar's positivstellensatz, that give necessary conditions for a polynomial to be positive over a semi-algebraic set $\mathcal{S}$ provided that the latter satisfies certain conditions, in particular compacity.

Theorem 2.5.11 *Schmüdgen's positivstellensatz*

Let us consider a polynomial $p \in \mathbb{R}[\mathrm{x}]$, a compact semi-algebraic set $\mathcal{S} = \{x \in \mathbb{R}^n : f_j(x) \geq 0,\ j = 1, \cdots, m\}$ and the set P defined in Theorem 2.5.10. Then, we have the following implication :

$$p(x) > 0 \text{ on } \mathcal{S} \Rightarrow p \in P$$

The necessary conditions provided by the Putinar's positivstellensatz are easier to verify, but on the other hand, it is more restrictive about $\mathcal{S}$, that has to satisfy the following *Putinar's conditions* :

Definition 2.5.12 *Putinar's conditions*
A semi-algebraic set $\mathcal{S} = \{x \in \mathbb{R}^n : f_j(x) \geq 0,\ j = 1, \cdots, m\}$ satisfies the Putinar's conditions if :

- *$\mathcal{S}$ is compact ;*
- *There exists $d \in \mathbb{N}$ and $u \in \mathbb{R}_d[\mathrm{x}]$ such that*
  - *$\{x \in \mathbb{R}^n : u(x) \geq 0\}$ is compact ;*
  - *$u = u_0 + \sum_{j=1}^m u_j f_j$ for some s.o.s. polynomials $u_j,\ j = 0, ..., m$.*

These conditions are easily verified. It suffices for example that one set $\{x \in \mathbb{R}^n : f_j(x) \geq 0\}$ be compact. It is also verified if $\mathcal{S}$ is a bounded polyhedron. Anyhow, it is always possible to ensure its validity by adding the constraint $\|x\|^2 \leq a^2$ for a sufficiently large value of a.

Theorem 2.5.13 *Putinar's positivstellensatz*
Let us consider a polynomial $p \in \mathbb{R}_d[\mathsf{x}]$ and a semi-algebraic set $\mathcal{S} = \{x \in \mathbb{R}^n : f_j(x) \geq 0,\ j = 1, \cdots, m\}$ that satisfies the above Putinar's conditions (Def. 2.5.12). Then we have the following equivalence :

$$p(x) > 0 \text{ on } \mathcal{S} \Leftrightarrow p = p_0 + \sum_{j=1}^{m} f_j p_j \text{ for some s.o.s. polynomials } p_i,\ i = 0, ..., m$$

This result is fundamental for polynomial optimization since it provides a "linear" sufficient condition for $p \geq 0$ on $\mathcal{S}$.

+

2.5.3 Quadratic functions

A *quadratic* function is a 2-degree polynomial. Specific notations are used for these polynomials. The most usual is $p(x) = x^T P x + 2p^T x + \pi$ with $P \in \mathbb{S}^n$, $p \in \mathbb{R}^n$ and $\pi \in \mathbb{R}$. Note that requiring that P be symmetric is not a loss of generality.

A polynomial can also be represented in a parametric fashion by $p(.; P, p, \pi)$ or $p(.; Q)$ where Q is the augmented matrix $Q = \begin{pmatrix} \pi & p^T \\ p & P \end{pmatrix}$:

$$p(x) = x^T P x + 2p^T x + \pi = \tilde{x}^T Q \tilde{x} \tag{2.3}$$

where $\tilde{x} = \begin{pmatrix} 1 & x^T \end{pmatrix}^T$ is the homogenization of x.

Thus, a quadratic function over $\mathbb{R}^n$ can be entirely represented by a $n+1$-dimensional symmetric matrix. Not surprinsingly, its property are therefore strongly related to the properties of this matrix.

As any polynomial, a quadratic function $p(.; P, p, \pi)$ is twice differentiable, its gradient at x is equal to $2(Px + p)$ and its Hessian is equal to $2P$. From this follows the following proposition :

Proposition 2.5.14 *Let us consider a quadratic function $p(.; P, p, \pi) = p(.; Q)$. Then*

- *(i) p is convex if and only if $P \succcurlyeq 0$;*
- *(ii) p admits a minimum if and only if $P \succcurlyeq 0$ et $p \in \mathcal{R}(P)$;*
- *(iii) p is nonnegative over $\mathbb{R}^n$ if and only if $Q \succcurlyeq 0$.*

Moreover, when the statement (ii) holds, $\arg\min p = \{x \in \mathbb{R}^n : Px = -p\}$.

If p admits a minimum p^*, then $p(x) \geq p^*, \forall x \in \mathbb{R}^n$, which is equivalent to the positivity of $p(x) - p^*$ over $\mathbb{R}^n$. We deduce from this the following proposition.

Proposition 2.5.15 *Let us consider a matrix $P \in \mathbb{S}^n_+$, a vector $p \in \mathbb{R}^n$ and a real π. Then we have the following equivalence :*

$$p \in \mathcal{R}(P) \Leftrightarrow \exists p^* \text{ such that } \begin{pmatrix} \pi - p^* & p^T \\ p & P \end{pmatrix} \succcurlyeq 0$$

2.6 Uncertainty

This section is based for main part on the lectures [60, 123].

2.6.1 Probability measure

Intuitively speaking, a measure on a set S is a way to assign a non-negative real number to each subset of S so that this number can be interpreted as the size of the subset. To qualify such a function as a measure, some conditions must be satisfied, in particular countable additivity, that states that the size of the union of a sequence of disjoint subsets is equal to the sum of the sizes of the subsets. However it is generally impossible to satisfy these conditions on all the subset of S. To settle this problem, a measure is defined only on certain subsets, called *measurable*, that forms a σ-algebra.

Definition 2.6.1 *σ-algebra*
Let Ω be a set and 2^Ω the set of all the subset of Ω. Then $\Sigma \subset 2^\Omega$ is a σ-algebra on Ω if it satisfies the following properties :

- *Σ is non empty ;*
- *Σ is closed under complementation : if $A \in \Sigma$ then $A^C := \Omega \setminus A \in \Sigma$*
- *Σ is closed under countable unions : If $\{A_i\}_{i\in I}$ is a countable collections of elements of Σ, then $\cup_{i\in I} A_i \in \Sigma$.*

As a consequence, a σ-algebra is closed under countable intersection. If Σ is a σ-algebra over Ω, then (Ω, Σ) is a *measurable space.*

Example 2.6.2 *If Ω is a topological space, then the Borel σ-algebra on Ω is the smallest σ-algebra containing all open sets (or, equivalently, all closed sets). For example, the Borel algebra on the real, denoted $\mathcal{B}_\mathbb{R}$ is the smallest σ-algebra on $\mathbb{R}$ that contains all the intervals.*

A measure on Ω is a way to assign to each element of a σ-algebra on Ω a real nonnegative number, intuitively interpreted as the size of the subset.

Definition 2.6.3 *Measure*
Let us consider a set Ω and Σ a σ-algebra over Ω. A function $\mu : \Sigma \to \bar{\mathbb{R}}$ is called a measure if it satisfies the following properties :

- *Non-negativity : $\mu(E) \geq 0, \forall E \in \Sigma$;*
- *Countable additivity : $\mu(\cup_{i\in I} E_i) = \sum_{i\in I} \mu(E_i)$ for all countable collections $\{E_i\}_{i\in I}$ of pairwise disjoint elements of Σ ;*
- *Null emptyset : $\mu(\emptyset) = 0$.*

Example 2.6.4 *Lebesgue measure*
If $\Omega = \mathbb{R}$ and Σ is its Borel σ-algrebra, then a possible measure over Ω is the Lebesgue measure that associates to any interval $[a, b]$ the value $b - a$.

Definition 2.6.5 *Measure space*
Let us consider a set Ω, Σ a σ-algebra over Ω and a measure μ over Ω. Then the triplet (Ω, Σ, μ) is called a measure space.

Definition 2.6.6 *Finite signed measure*
Let Σ be a σ-algebra over a set Ω. A function $\mu : \Sigma \to \mathbb{R}$ is a finite signed measure if it satisfies the properties of countable additivity and null emptyset.

Thus, the difference with a measure is that negative values are allowed, but not infinite values.

Definition 2.6.7 *Finite measure*
Let Σ be a σ-algebra over a set Ω. A function $\mu : \Sigma \to \mathbb{R}$ is a finite measure if it satisfies the properties of non-negativity, countable additivity and null emptyset.

Thus, the only difference with a measure is that infinite values are not allowed.

Proposition 2.6.8 *The set of finite measure is a convex cone.*

We are particularly interested in a special class of measure, where measuring the whole space yields 1.

Definition 2.6.9 *Probability measure*
Let (Ω, Σ, μ) be a measure space. Then μ is a probability measure if $\mu(\Omega) = 1$. Then (Ω, Σ, μ) is called a probability space.

In this case, the elements of Σ are called *events* and $\mu(S)$ for $S \in \Sigma$ is the probability of the event S.

Definition 2.6.10 *Measurable application*
Let us consider two measurable spaces (Ω, Σ) and (Ω', Σ'). An application $X : \Sigma \to \Sigma'$ is said measurable if it is inversible and if $\forall S' \in \Sigma', X^{-1}(S') \in \Sigma$, where X^{-1} is the inverse application of X.

2.6.2 Random variables

Definition 2.6.11 *Random variable*
Let us consider a probability space $\mathcal{P} = (\Omega, \Sigma, \mu)$. A real random variable on $\mathcal{P}$ is a measurable application $X : \Sigma \to \mathcal{B}_{\mathbb{R}}$.

If the set $\{X(\omega) : \omega \in \Sigma\}$ is finite or countable, the random variable is said *discrete*.

Example 2.6.12 *For example, Ω is a sequence of n die rolls and X is the occurrence of one face among n, or X is the sum of the rolls.*

Definition 2.6.13 *Probability distribution*
Let us consider a random variable on a probability space (Ω, Σ, μ). Its probability distribution is the image of μ by X, that is the function P such that :

$$\begin{array}{rccc} \mathrm{P}: & \mathcal{B}_{\mathbb{R}} & \to & [0,1] \\ & B & \mapsto & \mu(X^{-1}(B)) \end{array}$$

Traditionally, $\mathrm{P}(B)$ is rather written $\mathrm{P}[X \in B]$.

Having a random variable X of probability distribution P is denoted by $X \equiv \mathrm{P}$.

Example 2.6.14 *A famous example of law of probability for a real random variable is the Gaussian or Normal distribution, denoted $\mathcal{N}(\mu, \sigma^2)$, where μ, σ are parameters that are sufficient to characterize the distribution.*

For discrete random variable on $\{0, ..., n\}$, a possible law is the Binomial distribution, where $\mathrm{P}[X = k] = \binom{n}{k} q^{n-k}(1-q)k$. Another one is the Poisson distribution on $\mathbb{N}$ with $\mathrm{P}[X = k] = \exp -\lambda \frac{\lambda^k}{k!}$.

Definition 2.6.15 *Random vectors*
Let us consider a probability space $\mathcal{P} = (\Omega, \Sigma, \mu)$ be a probability space. A real random vector on $\mathcal{P}$ is a measurable application $X : \Sigma \to \mathcal{B}_{\mathbb{R}^n}$.

Generally, $X_1, ..., X_n$ are function of a same probability space. By considering Ω of the above example 2.6.12, we could have X_1 the occurrence of a given face, and X_2 the sum of the rolls. Thus, X_1 and X_2 are correlated. However, this also includes the case when $X_1, ..., X_n$ is a sequence of n independent variables. This is the case for instance if X_i is the result of the i-th roll for $i = 1, ..., n$.

In the sequel, we use $\mathbb{R}^n$ and the term n-random variable to embrace real random variable ($n = 1$) and real random vector ($n > 1$). The case of the discrete random variable is not treated here.

Definition 2.6.16 *Independent random variables*
Let us consider a n_1-random variables X_1 and a n_2-random variables X_2. Then X_1 and X_2 are independent if for all $S_1 \in \mathcal{B}_{\mathbb{R}^{n_1}}$ and $S_2 \in \mathcal{B}_{\mathbb{R}^{n_2}}$, we have $\mathrm{P}[(X_1 \in S_1) \cap (X_2 \in S_2)] = \mathrm{P}[X_1 \in S_1]\mathrm{P}[X_2 \in S_2]$.

A sequence of random variables is independent and identically distributed (*i.i.d.*) if each random variable has the same probability distribution as the others and all are mutually independent.

Definition 2.6.17 *Support*
The support $\mathcal{S}$ of a n-random variable X is the smallest subset of $\mathbb{R}^n$ such that $\mathbb{P}[X \in \mathcal{S}] = 1$.

Definition 2.6.18 *Cumulative distribution function*
Let us consider a n-random variable X. The cumulative distribution function F of X is defined as following :

$$\begin{array}{rccl} F: & \mathbb{R}^n & \to & [0,1] \\ & x & \mapsto & F(x) = \mathrm{P}[\cap_{i=1}^n X_i \leq x_i] \end{array}$$

Example 2.6.19 *Let us consider a Gaussian real random variable $X \equiv \mathcal{N}(\mu, \sigma)$. Its cumulative distribution function is $F(x) = \Phi(\frac{x-\mu}{\sigma})$ where Φ is a symmetric function whose numerical values are known, as well as those of its inverse Φ^{-1}.*

Definition 2.6.20 *Probability density*
Let us consider a n-random variable X and its repartition function F. The function f is called probability density of f if it is an integrable function $f : \mathbb{R}^n \to [0,1]$ such that

$$F(x) = \int_{-\infty}^{x_1} ... \int_{-\infty}^{x_n} f(t1, ..., t_n)\ dt_1 ... dt_n \quad \forall x \in \mathbb{R}^n$$

In this case, X is said to be a continuous random variable.

Example 2.6.21 *The probability density of $\mathcal{N}(\mu, \sigma)$ is equal to $f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp - \frac{(x-\mu)^2}{2\sigma^2}$.*

Proposition 2.6.22 *Let us consider a continuous n-random variable X with repartition function F and density f and $a, b \in \mathbb{R}^n$. Then,*

$$\mathrm{P}[a < X \leq b] = F(b) - F(a) = \int_{a_1}^{b_1} ... \int_{a_n}^{b_n} f(t1, ..., t_n)\ dt_1 ... dt_n$$

Proposition 2.6.23 *If X is a continuous n-random variable and $S \subset \mathbb{R}^n$ is a countable union of single-valued sets, then $\mathrm{P}[X \in S] = 0$.*

In particular, for any vector $x_0 \in \mathbb{R}^n$, $\mathrm{P}[X = x_0] = 0$.

The following property plays a key role in the treatment of chance-constraints.

Definition 2.6.24 *Log-concave random variable*
A random variable is said to be log-concave if it admits a log-concave density function.

Example 2.6.25 *Many famous distribution have the property of log-concavity, in particular the uniform, normal, beta, exponential, and extreme value distributions*

2.6.3 Moments

Definition 2.6.26 *Integral of a finite random variable*
Let us consider a discrete random variable on the probability space (Ω, Σ, μ) that takes its value into the finite subset $\{x_i\}_{i=1,\dots,m}$ of $\mathbb{R}$ with probability $\mathrm{P}[X = x_i]$. Then, the integral of X over Ω, denoted $\int_\Omega X(\omega)\mu(\omega)d\omega$ or more simply $\int_\Omega X d\mathrm{P}(X)$ is defined as following :

$$\int_\Omega X(\omega)\mu(\omega)d\omega = \sum_{i=1}^{m} x_i \mu(E_i) = \sum_{i=1}^{m} x_i \mathrm{P}[X = x_i]$$

where $E_i = \{\omega \in \Omega : X(\omega) = x_i\}$.

This definition is extended to any n-random variable X on the probability space (Ω, Σ, μ) via a mechanism that we will not detail here.

Definition 2.6.27 *Expected value*
Let us consider a real random variable defined on the probability space (Ω, Σ, μ). Then its expected value, denoted $\mathrm{E}(X)$ is the value of $\int_\Omega X d\mathrm{P}(X)$.

If X is a n-random variable, its expected vector is the vector of the expected values of its components.

Example 2.6.28 *Let us consider a n-random variable X, a set $S \subset \mathbb{R}^n$ and $\mathbb{1}_S$ its indicator function. Then $\mathbb{1}_S(X)$ is a real random variable and its expected value is $\mathrm{E}(\mathbb{1}_S(X)) = \mathrm{P}[X \in S]$. Indeed,*

$$\int_\Omega \mathbb{1}_S(X)(\omega)\mu(\omega)d\omega = \int_{X^{-1}(S)} \mu(\omega)d\omega = \mu(X^{-1}(S)) = \mathrm{P}(S)$$

Definition 2.6.29 *Integrability*
A random variable X is integrable if $\mathrm{E}(|X|) < +\infty$.

Definition 2.6.30 *Moments*
Let us consider a n-random variable and an integer vector $\kappa \in \mathbb{N}_k^n$ such that $\sum_{i=1}^n \kappa_i = k'$. Then, the moment of X associated to κ is said to be of order k' and is defined as $\mathrm{E}(X^\kappa)$.

Consequently, $\mathrm{E}(X^\kappa) = \int_{\omega\in\Omega} X^\kappa(\omega)\mu(\omega)d\omega$.

Example 2.6.31 *In particular, moments of order 1 form the expected vector of X ($\mathrm{E}(X)$) and moments of order 2 form a matrix M such that $C = M - \mathrm{E}(X)\mathrm{E}(X)^T$ is the covariance matrix of X :*

$$\begin{aligned} C_{ij} &= \mathrm{E}(X_i X_j) - \mathrm{E}(X_i)\mathrm{E}(X_j) \\ &= \mathrm{E}((X_i - \mathrm{E}(X_i))(X_j - \mathrm{E}(X_j))) \end{aligned}$$

In the case when $n = 1$, $\mathrm{var}(X) = \mathrm{E}((X - \mathrm{E}(X))^2)$ is the variance of X and $\sigma(X) = \sqrt{\mathrm{var}(X)}$ is the standard deviation.

2.6.4 Laws and inequalities

The following theorem is fundamental for the connection between probability and statistic. It expresses the fact that the statistical average of a sample converges to the expected value of the underlying distribution when the size of the sample increases :

Theorem 2.6.32 *Law of large numbers*
Let us consider an infinite sequence of i.i.d. integrable random variables with expected value $\mathrm{E}(X1) = \mathrm{E}(X2) = ... = \mu$. *Then the average of these variables* $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$ *converges almost surely to* μ *when* n *tends to* $+\infty$*,i.e.,*

$$\mathrm{P}[\lim_{n\to+\infty} \bar{X}_n = \mu] = 1$$

An application of this theorem is that by repeating the same random experiment an infinite number of times, the relative frequencies of occurrence of each of the events will coincide with their probabilities.

The following theorem states that, given certain conditions, the average of a sufficiently large number of independent random variables will be approximately normally distributed.

Theorem 2.6.33 *Central limit theorem*
Let us consider an infinite sequence of i.i.d. integrable random variables with expected value μ *and variance* $\sigma^2 < +\infty$. *Let* $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$. *Then the random variable* $Z_n = \sqrt{n}(\bar{X}_n - \mu)$ *converges in distribution to* $\mathcal{N}(0,\sigma)$*, i.e., for any real number* β *:*

$$\lim_{n\to+\infty} \mathrm{P}[Z_n \leq z] = \Phi(z/\sigma)$$

where Φ *is the cumulative distribution function of the standard Gaussian distribution.*

The following inequality, discovered by Boole in 1854, approximates the probability of a *composite event* $\bigcup_{i=1}^{n} A_i$ by the sum of the probability of its *simple events* A_i. It was shown by Frechet in 1940 than no better approximation can be obtained without additional information.

Theorem 2.6.34 *Boole inequality*
Let us consider a n*-random variable with a probability law* P *and a sequence of arbitrary event* $A_i \in \mathcal{B}_{\mathbb{R}^n}$ *for* $i = 1, ..., n$. *Then*

$$\mathrm{P}[\bigcup_{i=1}^{n} A_i] \leq \sum_{i=1}^{n} \mathrm{P}[A_i]$$

It is important to note that this result does not require that the events $A_i, ..., A_n$ be independent.

Corollary 2.6.35

$$\mathrm{P}[\bigcap_{i=1}^{n} A_i] \geq \sum_{i=1}^{n} \mathrm{P}[A_i] + 1 - n$$

With additionnal information, such as the knowledge of $\mathrm{P}[A_i \cup A_j]$, the approximation can be refined, as stated by the following so-called textitBonferroni's inequality.

Theorem 2.6.36 *Bonferroni's inequality*
Let $S_k = \sum_{1\leq i_1<...<i_k\geq n} \mathrm{P}[A_{i_1} \cup ... \cup A_{i_k}]$. *Then for* $k \in \{1, ..., n\}$ *:*

$$(-1)^{k-1}\mathrm{P}[\bigcap_{i=1}^{n} A_i] \leq (-1)^{k-1} \sum_{j=1}^{k} (-1)^{j-1} S_j$$

When $k = n$ *the equality holds and the resulting identity is the inclusion-exclusion principle.*

At this point, we provide some inequalities connecting moments and probability.

Let X be a nonnegative real random variable on (Ω, Σ, μ), $t > 0$ a real. Then $\mathbb{1}_{[t,+\infty[}(X)$ is also a random variable and $X(\omega) \geq t\mathbb{1}_{[t,+\infty[}(X(\omega)), \forall \omega \in \Omega$.

Consequently, $\mathrm{E}(X) \geq \mathrm{E}(t\mathbb{1}_{[t,+\infty[}(X)) = t\mathrm{P}[X \geq t]$ and we deduce from this the Markov's inequality.

Theorem 2.6.37 *Markov's inequality*
Let us consider a real random variable X and a real $t > 0$:

$$\mathrm{P}[|X| \geq t] \leq \frac{\mathrm{E}(|X|)}{t}$$

By applying this to the real random variable $(X - \mathrm{E}(X))^2$, we get the Bienaymé-Chebyshev's inequality.

Theorem 2.6.38 *Bienaymé-Chebyshev's inequality*
Let us consider a real random variable X with finite expected value $\mathrm{E}(X)$ and finite non-zero variance $\mathrm{var}(X)$. Then for any real $t > 0$:

$$\mathrm{P}[|X - \mathrm{E}(X)| \geq t] \leq \frac{\mathrm{var}(X)}{t^2}$$

We are now interested in inequalities that give bounds on the probability that the sum of random variables deviates from its mean. These inequalities are included among the famous Bernstein's inequalities.

Theorem 2.6.39 *Azuma-Hoeffding's inequality*
Let us consider n independent real random variables $X_1, ..., X_n$ with support $[a_i, b_i]$, meaning that $\mathrm{P}[X_i \in [a_i, b_i]] = 1$ for $i = 1, ..., n$. Let $\bar{X}$ be the mean of these variables, i.e. the random variable defined as $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$. Then for any real number $t \geq 0$:

$$\mathrm{P}[\bar{X} - \mathrm{E}(\bar{X}) \geq t] \leq \exp\left(-\frac{2t^2n^2}{\|b - a\|^2}\right)$$

2.6.5 Risk measure

In this section, we introduce the notion of *risk measure*, stemming from the financial literature. This term is a slight abuse of terminology since it is not a measure in the sense of Definition 2.6.3. Note that the literature does not reach a consensus on the precise definition of a risk measure.

Definition 2.6.40 *Risk measure*
Let us consider a probability space $\mathcal{P} = (\Omega, \Sigma, \mu)$ and a set $\mathcal{R}$ of random variable over $\mathcal{P}$. A mapping $\rho : \mathcal{R} \to \bar{\mathbb{R}}$ such that

- *For $X \in \mathcal{R}$ and a real number $\lambda \geq 0$, $\rho(\lambda X) = \lambda\rho(X)$ (positive homogeneous) ;*
- *For $a \in \mathbb{R}$, $\rho(X + a) = \rho(X) - a$ (translative) ;*
- *For $X, X' \in \mathcal{R}$, $X \leq X' \Rightarrow \rho(X) \geq \rho(X')$ (monotone).*

Definition 2.6.41 *Coherent risk measure*
A risk measure ρ is said to be coherent it it has so so-called subadditivity property, i.e., for X, X' two real random variable on a probability space $\mathcal{P}$, $\rho(X + X') \leq \rho(X) + \rho(X')$.

Two risk measures are particularly useful : the *Value at Risk* (VaR) and the *Conditional Value at Risk* (CVaR), also called *Expected shortfall*.

Definition 2.6.42 VaR
Let us consider a real random variable X with probability law P *and a real number $\varepsilon > 0$. The, the* ε-VaR *of X is defined as following :*

$$\mathrm{VaR}_\varepsilon(X) = \inf\{\beta \in \mathbb{R} : \mathrm{P}[X \leq \beta] \geq \varepsilon\}$$

Proposition 2.6.43 *Let us consider a real random variable X with probability distribution* P *and a real number $\varepsilon > 0$. Then, we have the following equivalence :*

$$\mathrm{P}[X \leq \beta] \geq \varepsilon \Leftrightarrow \beta \geq \mathrm{VaR}_\varepsilon(X)$$

Definition 2.6.44 CVaR
Let us consider a real random variable X with probability law P *and a real number $\varepsilon > 0$. The, the* ε-CVaR *of X is defined as following :*

$$\mathrm{CVaR}_\varepsilon(X) = \inf_{\beta \in \mathbb{R}} \left\{ \beta + \frac{1}{\varepsilon} \mathrm{E}[(X - \beta)^+] \right\}$$

Thus $\mathrm{CVaR}_\varepsilon$ evaluates the conditional expectation of X above the ε)-VaR. This risk measure is widely used since it is both coherent and convex.

Proposition 2.6.45 *Let us consider a real random variable X with probability law* P *and a real number $\varepsilon > 0$. Then,*

$$\mathrm{P}[X \leq \mathrm{CVaR}_\varepsilon(X)] \geq 1 - \varepsilon$$

2.7 Graph

In this section, we give a brief introduction to graph theory, largely inspired by [200] and mainly defining the notions that are used within some famous combinatorial optimization problem.

Definition 2.7.1 *Graph*
A graph $G = (V, E)$ consists of a finite, nonempty set $V = \{v_1, ..., v_n\}$ and a set $E = \{e_1, ..., e_m\}$ whose elements are subset of V of size 2, that is $e_k = \{v_i, v_j\}$ where $v_i, v_j \in V$. The elements of E are called nodes and the element of E are called edges. If $e_k = \{v_i, v_j\}$, then we say that e_k is incident to v_i and v_j.

Unless other specified, we will assume that a graph $G = (V, E)$ is *simple*, i.e. that its edges are distinct and if $e = (v_i, v_j)$ then $v_i \neq v_j$.

Definition 2.7.2 *Incidence matrix*
Let us consider a graph $G = (V, E)$ with $|V| = n$ and $|E| = m$. Then the incidence matrix is the matrix A of $\{0, 1\}^{n,m}$ indexed by V and E such that $A_{v_i, e_k} = 1$ if and only if e_k is incident to v_i.

Definition 2.7.3 *Incidence set*
Let us consider a graph $G = (V, E)$ and a node $v_i \in V$. The incidence set of v_i, denoted $\delta(v_i) \subset E$ is the set of edges incident to v_i.

Definition 2.7.4 *Adjacency matrix*
Let us consider a graph $G = (V, E)$ with $|V| = n$. Then the incidence matrix is the matrix A' of $\{0, 1\}^{n,n}$ indexed by V such that $A_{v_i,v_j} = 1$ if and only if $(v_i, v_j) \in E$.

Definition 2.7.5 *Complete graph*
A graph $G = (V, E)$ is called complete if it contains all the possible edges, i.e. $|\delta(v_i)| = |V| - 1$ for all $v_i \in V$.

Definition 2.7.6 *Complement*
Let us consider a graph $G = (V, E)$. Then the complement of G is a simple graph $\bar{G} = (V, \bar{E})$ where $\bar{E} = \{(v_i, v_j) : (v_i, v_j) \notin E,\ v_i, v_j \in V\}$.

Definition 2.7.7 *Subgraph*
Let us consider a graph $G = (V, E)$, $V' \subset V$ and $E(V') = \{\{v_i, v_j\} \in E : v_i \in U, v_j \in U\}$. If $E' \subset E(V')$ then $G' = (V', E')$ is a subgraph of G. If $V = V'$ then G' is a spanning subgraph. If $E' = E(V')$ then G' is the subgraph induced by V'.

Definition 2.7.8 *Cut*
Let us consider a graph $G = (V, E)$. A cut is a partition of V into two disjoints subsets V_1 and V_2 and the cut-set is the set of edges whose end points are in different subsets of the partition. Edges are said to be crossing the cut if they are in its cut-set.

In other words, a cut-set is a subset of edges $F \subset E$ such that the subgraph (V, F) is bipartite.

Definition 2.7.9 *Stable set*
Let us consider a graph $G = (V, E)$ and a subset of nodes $V' \subset V$. V' is a stable set if there exists no $(v_i, v_j) \in E$ with $v_i, v_j \in V'$.

Definition 2.7.10 *Clique*
Let us consider a graph $G = (V, E)$ and a subset of nodes $V' \subset V$. V' is a clique if for all distinct pair of nodes $v_i, v_j \in V'$, $(v_i, v_j) \in E$.

The chromatic number of a graph G is defined as the minimum number of colors required to color the nodes of G so that no adjacent nodes have the same color. More formally,

Definition 2.7.11 *Chromatic number*
The chromatic number of a graph G is the minimum number k so that there exists a partition $V_1, ..., V_k$ of V where each V_i does not contain any element of E.

Definition 2.7.12 *Perfect graph*
A graph G is called perfect if the chromatic number of every induced subgraph of G equals the size of the largest clique of that subgraph.

Chapter 3

Optimization Background

The mathematical discipline of Optimization, also known as *Mathematical Programming*, can be defined as the selection of a best element w.r.t a given criteria $f(x)$, among a set of available alternative $\mathcal{F}$. This paradigm is represented under the following form :

$$\begin{cases} \min & f(x) \\ \text{s.t.} & x \in \mathcal{F} \end{cases}$$

Solving such a problem mainly consists of determining one minimizer, or optimal solution, x^*, and the optimal value $p^* = f(x^*)$. For arbitrary f and $\mathcal{F}$, this value has no analytical expression and must be calculated through a "black box" process, i.e., through an algorithm. In full generality, this process can take an infinite time, but there are some special cases where efficient algorithms exist.

In particular, in the case where f is linear and $\mathcal{F}$ is a polyhedron, the *simplex algorithm*, designed by Dantzig in 1947, solves the problem very efficiently. This milestone is generally considered as the official inception of optimization, even if some all-important works were carried out since the 18th century by forerunners such as Lagrange, Newton or Cauchy. Indeed, the methods designed to solve such problems require large amount of computational effort and their practical implementation were therefore carried out in parallel with the development of computer technology. It is currently one of the most active areas of applied mathematics.

This appendix aims at collecting together the definition and results related to optimization that are mentioned in the thesis. The particular case of convex optimization is treated in the main part of the thesis 1.

3.1 Generalities on optimization

3.1.1 Definition

Definition 3.1.1 *Optimization problem*
An optimization problem (P) consists of a set $\mathcal{F} \subset \mathbb{R}^n$, (the so-called feasible set) and a real-valued function $f_0 : \mathcal{F} \to \mathbb{R}$, (the so-called objective function), to minimize or to maximize over $\mathcal{F}$.

When $\mathcal{F}$ is defined through inequalities involving functions, i.e., $\mathcal{F} = \{x \in \mathbb{R}^n : f_i(x) \leq 0, i = 1, ..., m\}$, then the optimization problem can be written in the following form :

$$(P) \begin{cases} \min_{x \in \mathbb{R}^n} & f_0(x) \\ \text{s.t.} & f_i(x) \leq 0, \ i = 1, ..., m \end{cases}$$

In this case, we say that $\mathcal{F}$ is defined *explicitly*.

In the sequel, unless stated otherwise, we restrict ourself to the minimization case where the functions $f_i, i = 0, ..., m$ are continuous and $\mathcal{F}$ is a closed non-empty set embedded in a n-dimensional Euclidian space. This excludes, for example, optimization over the set of the probability distribution.

Remark that this framework includes maximization problems by replacing f_0 by its opposite. Equality constraints are supported by splitting them into two inequalities. Finally, if f_0 is not continuous, it can be replaced by an additional variable t, related to f_0 by the constraint $f_0(x) - t \leq 0$. For the same reason, when needed, we may consider without loss of generality that the objective function is linear.

In the case where the number of variables and (resp. or) the number of constraints is infinite, the problem is said to be (resp. *semi*) *infinite dimensional*. These problems, that are more challenging than finite-dimensional ones, won't be considered here, unless explicitly stated.

We are interested both in getting the optimal value, denoted p^*, and a minimizer x^*. In general, f_0 may fail to have an minimum over $\mathcal{F}$. For this reason, using min is somewhat inappropriate and should be replace by inf. Nevertheless, we allow ourselves this abuse of language since we consider that f_0 has an optimum over $\mathcal{F}$ in $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$, even if it may be not attained. Moreover, if $\mathcal{F}$ is bounded or if f_0 is coercive, then necessarily a minimum exists and is attained.

Definition 3.1.2 *Equivalent problems*
We say that two problems are equivalent, which is denoted by $(P_1) \equiv (P_2)$, if they have the same optimal value and if the optimal solution of one leads to the optimal solution of the other in polynomial time.

For example, the following problems are equivalent :

$$(P) \begin{array}{ll} \min & f_0(x) \\ \text{subject to} & x \in \mathcal{D} \end{array} \qquad (P^1) \begin{array}{ll} \min & t \\ \text{subject to} & f_0(x) \leq t \\ & x \in \mathcal{D} \\ & t \in \mathbb{R} \end{array}$$

This allows to consider problems with linear objective without loss of generality.

Definition 3.1.3 *Relaxation and conservative approximation*

The relaxation (P_r) of a problem (P) is a problem with the same function to minimize and whose feasible set of solution is included in the feasible set of (P). Conversely, (P_c) is a conservative approximation of the problem (P) if (P) is a relaxation of (P_c).

A simple way to obtain a relaxation of a problem is to remove one of its constraint. Another possibility is to embed $\mathcal{F}$ in a broader set. For example, an *outer approximation* is a technique that relaxes a problem by embedding $\mathcal{F}$ into a linear set.

Definition 3.1.4 *Outer Approximation*
An outer approximation is a special case of relaxation applied to problem with convex feasible set, obtained by replacing the feasible set by a larger polyhedron computed through the tangents at suitable boundary points.

Definition 3.1.5 *Projection*
A projection(P_P) of a problem (P) is an equivalent problem to (P) such that the solution of (P_P) is a projection of the solution of (P) on a less-dimensional space. Conversely, (P_L) is a lift of the problem (P) if (P) is a projection of (P_L).

Finally, we define the homogenization of an optimization problem as the embedding of its feasible set into a +1 dimensional space :

Definition 3.1.6 *Homogenization*
Let us consider an optimization problem (P) $\min c^T x : x \in \mathcal{F}$. *Its homogenization is defined as*

$$(P') \ \min f\begin{pmatrix} y \\ y_0 \end{pmatrix} : \begin{pmatrix} y_0 \\ y \end{pmatrix} \in \tilde{\mathcal{F}}$$

where $\tilde{\mathcal{F}}$ *is the homogenization of* $\tilde{\mathcal{F}}$.

(P) and (P') are closely related but are not equivalent since $0 \in \tilde{\mathcal{F}}$ even if $0 \notin \mathcal{F}$.

3.1.2 Complexity

The content of this paragraph is mainly taken from [60, 101].

Definition 3.1.7 *Decision problem*
A decision problem associates to an input a bivalent ("yes-or-no") answer.

For example, the input is an integer number and the decision problem is to decide whether this number is a prime number or not. Another famous example is the problem SAT that is presented in Paragraph 3.1.4.2.

In 1936, Turing developped a conceptual machine, the so-called *Turing machine* and thereby laid the fundations of the *computational complexity theory*. This machine is an abstract model of how works any calculation machine and the Church-Turing hypothesis claims that any computable function can be carried out on such a machine.

The Turing machine provides a way to measure the amount of spatial and temporal resources used by an algorithm in the worst-case, as a function of the size of the problem. The latter is traditionally expressed in the number of bits (bit length) used to represent the data, but some other, simpler, measures are also possible, like for instance the number of items involved in the problem. This is associated with the assumption that each elementary arithmetic operation takes one unit time, instead of the number of bit operations, and both constitutes the RAM model (in contrast to the Turing, or *bit* number model).

The comparison of these two concepts is relevant if and only if all the input of the algorithm consist of integers. Indeed, computers cannot represent real numbers precisely since the number of bits for storing is finite.

The amount of resources used by an algorithm obviously depend on the considered instance. In order to eliminate this dependency, three indicators are considered :

- The worst-case performance, used for instance for real time programming ;
- The average-case performance, that gives a general idea of running time but it often difficult to establish ;
- The best-case performance. It can be very useful to identify which are the instances that are concerned with these cases.

Definition 3.1.8 *Reduction*
A decision problem (P_R) *is the reduction of a problem* (P) *if there exists a polynomial way of converting the input of* (P) *into input of* (P_R). *Equivalently, we say that* (P) *reduces to* (P_R).

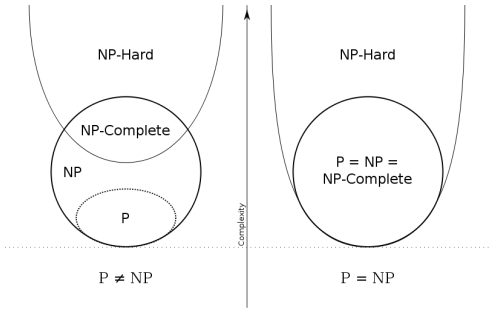

Figure 3.1: Relationship between a 0/1 LP, its linear relaxation and their semidefinite relaxations

In other words, (P) is, up to a polynomial conversion, a particular case of (P_R). Then the following theorem state the intuitive fact that if (P_R) is polynomially solvable, the same holds for (P).

This allows for the definition of *complexity classes* to classify the *decision problems* according to the amount of spatial and temporal resources that their resolution requires on a Turing machine. More generally, a complexity class is is a set of problems of related resource-based complexity and has a typical definition of the form: the set of problems that can be solved by an abstract machine M using $O(f(n))$ of resource R, where n is the size of the input. In particular, a problem of input size n is said to be of complexity class $DTIME(f(n))$ (or $TIME(f(n))$), if it can be solved on a deterministic Turing machine in $O(f(n))$ computation steps. This means that there exists an algorithm that solve this problem on a "normal" machine in $O(f(n))$ computation time, without any restriction on the amount of memory space used. This is the most common complexity measure used, because computational time is often prohibitive for large problems. In this thesis, we restrict ourselves to this temporal complexity, with the following complexity classes, to cite only the most important :

- P is the set of all decision problems which can be solved in polynomial time by a Turing machine;
- *NP* is the set of all decision problems for which the input where the answer is "yes" admit proofs of the fact that the answer is indeed "yes" and these proofs are verifiable in polynomial time by a Turing machine;
- *co-NP* is the set of all decision problems whose complement, i.e. the problem that answers "yes" if the answer is "no" and vice versa) belongs to NP ;
- *NP-hard* are the problems for which there exists a NP-complete problem that reduces to them;
- *NP-complete* are the problems both NP and NP-hard.

Remark that the definition of NP-hard and NP-complete may seem going round in circles. This difficulty is overcome by the following theorem [76] :

Theorem 3.1.9 *Cook's theorem SAT is NP-complete.*

Subsequently, Karp [151] showed that 21 famous combinatorial problems were NP-complete. The complexity class P is contained in NP, as illustrated on the Euler diagram above :

Proving whether P = NP or not is currently one of the most famous open problem in computer science, even it is widely believed that this is not the case. This is a key question since it comes to ask whether polynomial time algorithms actually exist for NP-complete problems.

This theory can be easily extended to optimization problems. Such problems don't belong to NP since NP contains only decision problems but they are naturally associated with a decision problem (D) in the following manner :

$$(P)\ \min_{x\in\mathcal{F}} f(x) \qquad (D)\ \text{Is there } x \in \mathcal{F} \text{ such that} x \leq K$$

If (P) is solved, then (D) is solved. Namely, (P) reduces to (D). Consequently, having (D) NP-hard implies that (P) is NP-hard.

Furthermore, by doing a binary search on K (see Paragraph 3.2.1) which is polynomial, one can embedd the resolution of (D) into a polynomial time algorithm for (P), which implies that (D) also reduces to (P).

Another more efficient possibility for converting an optimization problem into a decision problem comes about when sufficient and necessary optimality conditions are available for the problem, but this is not always the case.

In the case of 0-1 Linear Program, the associated decision problem is NP-complete. This can be proven with a reduction from SAT to 0-1LP, where SAT is the satisfiability problem, the first problem to be proven NP-complete. In this reduction, a binary variables is assigned to each positive literal and each clause is rewritten as a linear constraint, e.g. $(x_1 \vee \bar{x_2})$ becomes $y_1 + (1 - y_2) \geq 1$.

ILP reduces to 0-1LP and MILP reduces to ILP, so both are NP-hard. More generally, any Mixed-Integer Program can be reduced to MILP and is therefore NP-hard. In contrast, Khachiyan [153] proved the polynomiality of Linear Programming (LP) by using the ellipsoid method (see Paragraph 1.2.1). This result was extended to the optimization of a linear objective over a convex set, under the existence of a polynomial time separation oracle for this convex set [118].

3.1.3 Classification of optimization problems

Depending on the characteristics of the feasible set and of the objective function, some properties can be attributed to (P), impacting significantly the scope of (P) and the method for its resolution. These properties are generally non-exclusive and are given above by order of importance. Indeed the key property of an optimization problem lies in its convexity, since its guarantees that the problem be polynomially solvable.

We consider the problem :

$$(P) \begin{cases} \min_{x \in \mathbb{R}^n} & f_0(x) \\ \text{s.t.} & x \in \mathcal{F} \end{cases}$$

When needed, $\mathcal{F} = \{x \in \mathbb{R}^n : f_i(x) \leq 0,\ i = 1, ..., m\}$.

Table 3.1: Properties of an optimization problem

Property	**Definition**
Convex	if all the functions $f_i, i = 0, ..., m$ are convex
Linear	if all the functions $f_i, i = 0, ..., m$ are linear
Quadratic	if all the functions $f_i, i = 0, ..., m$ are quadratic
Polynomial	if all the functions $f_i, i = 0, ..., m$ are polynomial
Combinatorial	if the set $\mathcal{F}$ is finite
Differentiable	if the functions $f_i, i = 0, ..., m$ are differentiable
Uncertain	if the functions $f_i, i = 0, ..., m$ depends on random variables
Multi-stage	if a part of the decisions can be made once the uncertainty is released
Semi-infinite dimensional	if the number of variables or constraints is infinite

3.1.4 Focus on combinatorial optimization

Combinatorial (or discrete) optimization deals with all the optimization problems that have a finite (or possibly countable infinite) number of feasible solutions. The difficulty for handling them comes from the possibly very large number of elements of the feasible set. This area of optimization includes famous problems as Traveling Salesman Problem or Multidimensional Knapsack. More generally, it has a widespread range of application due to the following possibilities that it offers :

- managing indivisible resources ;
- Modeling "yes-or-no" decisions ;
- Enforcing logical conditions ;
- Modeling fixed costs ;
- Modeling piecewise linear functions.

See [200] for an excellent survey on the topic and examples of applications.

The majority of combinatorial optimization can be formulated as Mixed-Integer Programs, i.e. optimization problems with integrality restrictions on some of the variables :

$$\begin{cases} \min & f_0(x,y) \\ \text{s.t.} & f_i(x,y) \leq 0,\ i = 1,...,m, \\ & x \in \mathbb{Z}^n \end{cases}$$

When all the variables are required to be integer, the problem is said to be a *pure integer program*. Otherwise, the term *mixed* refers to the presence of continuous variables. A natural relaxation for these problem is the *continuous relaxation*, which is obtained by relaxing the integrality constraints :

$$\begin{cases} \min & f_0(x,y) \\ \text{s.t.} & f_i(x,y) \leq 0,\ i = 1,...,m \end{cases}$$

Some problems in the area are polynomial. A famous example is the problem of determining the shortest path in a weighted graph, whose a particular polynomial algorithm, the so-called Bellman-Ford algorithm, laid the fundation for dynamic programming. In the case of Integer Linear Program, if the feasible set is defined by linear systems with totally unimodular matrices, then the optimal solution of its linear relaxation is integer and the problem is therefore polynomial.

But majority of the combinatorial problem share the property of being NP-hard. Indeed, the discrete property make them loose some useful properties for optimization such as continuity, convexity or duality. In this case, three kind of resolution methods are possible :

- Enumerative method, i.e. exploring the whole set of feasible solutions. Lead to the optimal solution in exponential time.
- Approximation algorithms, i.e. polynomial-time algorithms that computes for every instance of the problem a solution with some guaranteed quality.
- Heuristic, i.e., a method that provide a feasible solution without any guarantee in terms of solution quality or running time.

For mixed-integer programs, enumerative methods include Branch & Bound and Branch & Cut methods, that are detailled for MILP in Paragraph 3.4.1 and BandCforMILP. These methods rely on the computation of lower bounds, which accounts for the motivation of solving efficiently tight relaxations of the problem.

To conclude this section and as an illustration, we briefly describe the MAX-CUT problem, that played a key role in arousing interest for semidefinite relaxation of combinatorial problems.

3.1.4.1 MAX-CUT

Given a graph $G = (V, E)$ with a weight w_e for each edge $e \in E$, the objective is to find a subset $W \subset V$ such that the edge cut, i.e., the set of edges with exactly one extremity in W, have maximum total weight. Let M is the symmetric matrix indexed by the elements of E that contains the weight of the edges : $M_{ij} = w_{(i,j)}$ and for each vertices of $i \in G$, we define a bivalent variable x_i that equals 1 if $i \in W$, -1 otherwise. Then the problem is : $\max \sum_{(ij)\in V} \frac{1-x_i x_j}{2} : x \in \{-1, 1\}^{|V|} = \max \frac{1}{4} x^T L x : x \in \{-1, 1\}^{|V|}$ where L is the weighted Laplacian of the graph, that is $L_{ii} = \sum_{j:(i,j)\in E} w_{ij}$, $L_{ij} = -w_{ij}$ for $(i, j) \in E$, $L_{ij} = 0$ otherwise.

This problem is therefore a quadratic program with the sole constraint that the constraint be bivalent. It was shown (see for instance [128, 175]) that any unconstrained bivalent quadratic problem, can be converted, up to an additive constant, into an instance of MAX-CUT.

A very interesting result related to MAX-CUT involves the Unique Game Conjecture, that states that the following decision problem is NP-hard. Given a weight graph $G = (V, E)$, with the weights $w_e,\ e \in E$ and a real $0 < \varepsilon < 1$, does there exist an affectation of real x_i to the elements of V such that $x_i + x_j = w_{(i,j)} \mathrm{mod} k$ for at least $(1 - \varepsilon)|E|$ elements of E ?

Several inapproxability results rely on this conjecture and in particular, the following one concerns MAX-CUT :

Theorem 3.1.10 *[155]*
Suppose the Unique Game Conjecture. Then it is NP-hard to find an approximation algorithm for MAX-CUT with a guarantee better than 0.878*.*

3.1.4.2 SAT and MAX-SAT

The satisfiability problem, commonly abbreviated SAT, is a decision problem consisting of deciding whether there exists an assignment for a set of bivalent variables $(x_1, ..., x_n)$ taking the value true or false, such that a set of clause be satisfied. A clause is a disjunction of literals and a literal is either a variable x_i or its negation $\bar{x_i}$. For instance the clause $x_1 \bigvee \bar{x_2}$ is satisfied if $x_1 = true$ or $x_2 = false$, where the "or" is not exclusive.

The MAX-SAT problem is an optimization problem related to SAT. Instead of requiring that all the clauses be satisfied, we aim at maximizing the number of satisfied clauses, or the sum of the weight of the satisfied clauses if such a weight is attributed to the clauses. Another variant of this problem is MAX-kSAT, where each clause is of length at most k. In particular, it was shown in [101] that MAX-2SAT is NP-hard.

3.1.4.3 Disjunctive Programming

Disjunctive Programming is a special case of Mathematical Programming where the feasible set of the problem is defined as the union of several sets. It is therefore an example of nonconvex optimization.

The following formalism is used : $\min f(x) : x \in \mathcal{F}_1 \bigvee ... \bigvee x \in \mathcal{F}_m$ or equivalently $\min f(x) : \bigvee_{i=1}^{m} x \in \mathcal{F}_i$.

In particular, it includes 0/1 programming since the constraints $x_i \in \{0, 1\}$ can be formulated as $x_i = 0 \bigvee x_i = 1$. We can even consider that it includes any combinatorial problem with a finite number of feasible solutions.

In the case of Disjunctive Linear Programming, the objective is to minimize a linear function while satisfying a system of conjunctive and disjunctive linear constraints :

$$\min c^T x : \bigvee_{i=1}^{m} \bigwedge_{j=1}^{k_i} a_{ij} x \leq b_{ij}$$

The feasible set is therefore the union of individual polyhedra.

From a modelling point of view, the disjunctive formulation is very powerful. It is the most natural way of stating many problems involving logical conditions. It can also be useful for modeling piecewise defined functions.

A classical way of modeling exclusive disjunction, i.e. disjunction that can not be satisfied simultaneously, is to convert it into a 0/1 program, through the use of a "big M" binary variable. Thus, we recover a special case of combinatorial optimization.

3.1.5 Lagrangian of an optimization problem

Let us consider the following optimization problem :

$$\begin{cases} \min_x & f_0(x) \\ & f_j(x) \leq 0,\ j = 1, ..., m \\ & g_j(x) = 0,\ j = 1, ..., p \\ & x \in \mathcal{K} \end{cases} \tag{3.1}$$

where $f_j,\ j = 0, ..., m$ and $g_j,\ j = 1, ...p$ are functions from $\mathbb{R}^n$ to $\mathbb{R}$ and $\mathcal{K}$ is a subset of $\mathbb{R}^n$.

To the optimization problem (3.1), whose optimal value is denoted by p^*, we associate a function $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^{m+p} \to \mathbb{R}$, called *Lagrangian*, that combines the objective and the violation of the constraints :

$$\mathcal{L}(x, \lambda, \mu) = f_0(x) + \sum_{j=1}^{m} \lambda_j f_j(x) + \sum_{j=1}^{p} \mu_j g_j(x)$$

The λ_j and μ_j are called *Lagrangian multipliers* and the *Lagrange dual function* $l : \mathbb{R}^{m+p} \to \mathbb{R}$ is defined as following :

$$l(\lambda, \mu) = \inf_{x in \mathcal{K}} \mathcal{L}(x, \lambda, \mu)$$

Two important observations have to be mentioned about the function l. First it is concave as the infimum of a family of affine function. Second, if $\lambda \geq 0$ then $l(\lambda, \mu) \leq p^*$.

The fundamental statement regarding the Lagrangian is that it has the same optimal value as the original problem. Indeed, it takes the same value as the objective for all values of x that satisfies the primal constraints, and is positive infinity if the constraints are violated :

$$\inf_{x \in \mathcal{K}} \sup_{\lambda \geq 0} \mathcal{L}(x, \lambda, \mu) = \inf_{x \in \mathcal{K}} \begin{cases} f_0(x) & \text{if } f_j(x) \leq 0,\ j = 1, ..., m; g_j(x) = 0,\ j = 1, ..., p \\ +\infty & \text{otherwise} \end{cases} = p^*$$

This leads to the first utilization of the Lagrangian, by defining another optimization problem, the *dual problem*, closely related to the original problem. Under some regularity conditions, the Lagrangian can be used to derive some necessary (sometimes sufficient) first-order optimality conditions, namely the Karush-Kuhn-Tucker (KKT) conditions.

3.1.5.1 Lagrangian duality

The *Lagrangian dual* of the problem is obtained by permuting the minimization and the maximization in the previous formulation 3.2 :

$$\begin{aligned} d^* &= \sup_{\lambda \geq 0} \inf_{x \in \mathcal{K}} \mathcal{L}(x, \lambda, \mu) \\ &= \sup l(\lambda, \mu) : \lambda \geq 0 \end{aligned}$$

This problem, called *dual problem* by opposition to the *primal problem* (3.1), is necessarily convex, as the maximization on a convex set $(\mathcal{R}_+^m \times \mathcal{R}^p)$ of the concave function l.

It can be interpreted as the selection of the best lower bound among the set of lower bounds $\{l(\lambda, \mu) : \lambda \geq 0\}$. Thus, $d^* \leq p^*$, which is constitutes the *weak duality*. The *strong duality* holds whenever $d^* = p^*$ but this is generally not the case. The strong duality can also be interpreted in terms of the existence of a saddle-point of the Lagrangian.

The primal and dual problems are intimately connected "as the two faces of the same coin" following the standard expression.

In the particular case of convex (or conic) programming, we will see that an elegant theory yields an explicit formulation for this problem. However, this is not the case in general. Furthermore, it is generally not differentiable, even when all the function f_j and g_j are differentiable.

3.1.5.2 Karush-Kuhn-Tucker (KKT) conditions

Under some regularity conditions, the Lagrangian can be employed to derive necessary first-order optimality conditions for the following optimization problem, a special case of the problem (3.1) with $\mathcal{K} = \mathbb{R}^n$.

First, we discuss the case where $m = 0$, i.e., the only constraints are equality constraints. We define $X = \{x \in \mathbb{R}^n : \mathsf{g}(x) = 0\}$ with $\mathsf{g} : \mathbb{R}^n \to \mathbb{R}^p$ and we assume that these constraints satisfy the *constraints qualification*, which comes to state the following equality :

$$\mathcal{T}_X(x) = \mathcal{N} J_{\mathsf{g}}(x)$$

with $\mathcal{T}_X(x)$ the Bouligand Tangent cone of X at x and $J_{\mathsf{g}}(x)$ the Jacobian matrix of g at x.

Thanks to this assumption and by applying the first-order condition of optimality 2.4.27, we derive the following theorem :

Theorem 3.1.11 *Lagrange theorem Let x^* be a local solution of the optimization problem : $\min_x f_0(x) : g_j(x) = 0,\ j = 1, ..., p$. Assume that f_0 and $g_j,\ j = 1, ..., p$ be differentiable at x^* and that the constraints satisfy the constraints qualification. Then, there exists a vector μ^* such that :*

$$\nabla f_0(x^*) + \sum_{j=1}^{p} \mu_j^* \nabla g_j(x^*) = 0$$

By assessing the feasibility of x^*, we obtain the following system of necessary conditions :

$$\begin{aligned} &\nabla f_0(x^*) + J_{\mathsf{g}}(x^*)^T \mu^* = 0 \\ &\mathsf{g}(x^*) = 0 \end{aligned}$$

This is equivalent to require that the gradient of the Lagrangian vanishes : $\nabla_{(x,\mu)} \mathcal{L}(x^*, \mu^*) = 0$.

By extending this process to an optimization with inequality constraints, we obtain the following Karush-Kuhn-Tucker (KKT) conditions.

Theorem 3.1.12 *Let x^* be a local solution of the following optimization problem :*

$$\begin{cases} \min_x & f_0(x) \\ & f_j(x) \leq 0,\ j = 1, ..., m \\ & g_j(x) = 0,\ j = 1, ..., p \end{cases}$$

Assume that f_0 and $f_j,\ j = 1, ..., m$ and $g_j,\ j = 1, ..., p$ be differentiable at x^* and that the constraints satisfy the constraints qualification. Then, there exists vectors λ^* and μ^* such that :

$$\begin{array}{l}\nabla f_0(x^*) + \sum_{j=1}^m \lambda_j^* \nabla f_j(x^*) + \sum_{j=1}^p \mu_j^* \nabla g_j(x^*) = 0 \\ \mathsf{f}(x) \leq 0 \\ \mathsf{g}(x) = 0 \\ \lambda^* \geq 0 \\ \lambda_i^* f_i(x^*) = 0,\ i = 1, ..., m\end{array}$$

Thus, the first constraint is obtained by setting the Lagrangian partial derivative to zero. The following ones are necessary to ensure the primal feasibility of x^* and the dual feasibility of (λ^*, μ^*). The last constraint, referred to as *complementary slackness*, can be seen as a consequence of a zero duality gap. Indeed,

$$\begin{array}{ll} f_0(x^*) & = l(\lambda^*, \mu^*) \\ & = \inf_x f_0(x) + \sum_{j=1}^m \lambda_j^* f_j(x) + \sum_{j=1}^p \mu_j^* g_j(x) \\ & \leq f_0(x^*) + \sum_{j=1}^m \lambda_j^* f_j(x^*) + \sum_{j=1}^p \mu_j^* g_j(x^*) \end{array}$$

Hence, as $g_j(x^*) = 0,\ j = 1, ..., p$, we have $\sum_{j=1}^m \lambda_j f_j(x^*) \geq 0$. As $g_j(x^*) \leq 0,\ j = 1, ..., m$ and $\lambda_j^* \geq 0$, necessarily $\lambda_j f_j(x^*) = 0,\ j = 1, ..., m$.

These constraints can be interpreted as requiring that the Lagrange multipliers associated to an inactive constraints ($f_i(x^*) < 0$) is zero. The complementarity is *strict* if

$$\left(f_i(x^*) < 0 \Leftrightarrow \lambda_i = 0\right),\ i = 1, ..., m$$

In the case when the strong duality holds, the Lagrangian admits a saddle-point, which is then a solution of the KKT system. In particular, a saddle-point is a stationary point, which is consistent with the first constraint which makes the gradient of the Lagrangian to vanish.

3.1.5.3 Constraints qualification

The KKT conditions hold only under the constraint qualification assumption, which rules out certain irregularities on the boundary of the feasible set.

Some well-known sufficient conditions to constraints qualification are appended below :

- if $f_j,\ j = 1, ..., m$ and $g_j,\ j = 1, ..., p$ are affine (Linearity) ;
- if the problem is convex and if there exists a strictly feasible point, i.e. x_0 such that $f_j(x_0) < 0,\ j = 1, ..., m$ and $g_j(x_0) = 0,\ j = 1, ..., p$ (Slater's condition) ;
- if the gradients of the active inequality constraints and the gradients of the equality constraints are linearly independent at x^* (Linear independence constraint qualification);
- if the gradients of the active inequality constraints and the gradients of the equality constraints are positive-linearly independent at x^* (Mangasarian - Fromovitz constraint qualification).

3.1.6 Optimization over the convex hull

This section yields some fundamental results regarding the relaxation of a problem over its convex hull. More precisely, let us consider the following optimization problem : $(P) \min_{x \in \mathcal{F}} f(x)$ and its relaxation $(P_C) \min_{x \in \text{conv}(\mathcal{F})} f(x)$.

$\text{conv}(\mathcal{F})$ is the convex hull of $\mathcal{F}$, the smallest convex set such that $\mathcal{F} \subseteq \text{conv}(\mathcal{F})$ (see Def. 2.1.2). Consequently, (P_C) is a convex relaxation of (P) and is theoretically easier to solve than (P), provided that the representation of $\text{conv}(\mathcal{F})$ be tractable.

With p_C^* and p^* the optimal values of (P) and (P_C) respectively, it is clear that $p_C^* \leq p^*$. In the case where f is concave and $\text{conv}(\mathcal{F})$ is compact, it is straightforward that (P) and (P_C) have the same optimal value.

Proof 3.1.13 *Assume that $p_C^* < p^*$. There exists $x_0 \in \text{conv}(\mathcal{F})$, $x_0 \notin \mathcal{F}$ such that $f(x_0) < f(x)$, $\forall x \in \mathcal{F}$. As $x_0 \in \text{conv}(\mathcal{F})$, there exists $x_1, x_2 \in \mathcal{F}$ and $\lambda \in [0,1]$ such that $x_0 = \lambda x_1 + (1-\lambda)x_2$, and from the concavity of f : $\lambda f(x_1) + (1-\lambda)f(x_2) < f(x)$, $\forall x \in \mathcal{F}$ which is a contradiction.* □

Nevertheless, (P) and (P_C) do not have necessarily the same minimizers, since there may be some minimizer of (P_C) that do not belong to $\mathcal{F}$ [140].

This equivalence is widely used in particular for Integer Linear Programming since $\mathcal{F}$ is finite and f is linear. The difficulty in this case is to compute an description of $\text{conv}(\mathcal{F})$ in the form of a polyhedron, in order to write this relaxation in the form of a Linear Program.

3.2 Algorithms of particular interest for optimization

In this section,largely based on the book of Minoux [194], we define some general properties of algorithm, before describing in detail some particular algorithms that play a special role in optimization.

An algorithm is a step-by-step process for computing a function of arbitrary inputs. It is said to be *exact* if it computes the exact solution. Otherwise, it is said to be an *heuristic*. The term *meta-heuristic* refers to heuristics that are not dedicated to a special problem. In the case where the algorithm is not exact, but is a polynomial-time algorithms with a guarantee on the obtained solution, then it is an *approximation algorithm*. If p^* is the exact solution of the problem, and $\tilde{p}$ the solution of the approximation algorithm with

$$\rho p^* \leq \tilde{p} \leq p^*$$

then the algorithm is a *ρ-approximation algorithm* and the factor ρ is the *relative performance guarantee* of the algorithm.

When an algorithm involves random data, it is said to be a *randomized algorithm*. In the case of a randomized approximation algorithm, the quality measure is assessed through the expected value of the relative performance guarantee.

Finally, in the case of an optimization problem, where the function to compute is an optimum over a given set, we distinguish the exact algorithms from algorithms that compute a local optimum, that we call *local optimization algorithm*.

3.2.1 Binary search

This algorithm aims at determining the smallest element of a discrete and sorted set of values $\mathcal{S} = \{s_1, ..., s_n\}$ that satisfies a certain statement $f(s) = 1$, with the following property :

$$f(\bar{s}) = 1 \Rightarrow f(s) = 1, \forall s \geq \bar{s}$$
$$f(\bar{s}) = 0 \Rightarrow f(s) = 0, \forall s \leq \bar{s}$$

Then the binary search, described below, converges to the solution, i.e., s_i such that $f(s_i) = 0$ and $f(s_{i+1}) = 1$, in at most $\log(n)$ iterations.

```
1: Let i = ⌊n/2⌋
2: while f(s_i) = 1 OR f(s_{i+1}) = 0 do
3:    if i = n then
4:       return "Not found"
5:    else
```

```
      if  f(s_i) = 1 then
        i ← i − ⌊i/2⌋
      else
        i ← i + ⌊i/2⌋
      end if
    end if
  end while
  return  s_i
```

This algorithm is used to convert an optimization problem $(P) \min f(x) : x \in \mathcal{F}$ into a decision problem $\exists t, x \in \mathcal{F} : t \geq f(x)$. Indeed, the smallest value of t that results in a "yes" answer of this decision problem is the optimal value of (P) and his

3.2.2 Gradient descent

This method aims at determining a local minima of a differentiable multivariate function $f : \mathbb{R}^n \to \mathbb{R}$.

This method is also known as the *method of the steepest descent*. It is based on the fact that the function f decreases fastest by moving in the direction of the negative gradient and is therefore a first-order method. Thus, at each iteration, move from x_k to $x_{k+1} = x_k - \gamma_k \nabla f(x_k)$. The method stops when $\|\nabla f(x_k)\| < \varepsilon$ for a given tolerance threshold ε.

$\gamma_k > 0$ is a small real value called *step size* and can be chosen in different ways :

- define $g(\gamma) = f(x_k - \gamma \nabla f(x_k))$ and solve $g'(\gamma) = 0$;
- determine a value of γ that satisfies Wolfe conditions, for example with a backtracking line search

The first method for determining γ leads to the so-called *conjugate gradient method*. One of the characteristic of this method is that two successive directions are orthogonal to each other : $d_k^T d_{k+1} = 0$, which may cause bad convergence properties for ill-conditioned functions.

Under relevant conditions, convergence to a local minimum can be guaranteed and the number of iterations required to obtain $\|f(x_k)\| \leq \varepsilon$ is at most $O(\varepsilon^{-2})$.

This method can be extended to constrained optimization. Let us consider for instance the constraints $f_i(x) \leq 0,\ i = 1, ..., m$, with f_i differentiable functions. We denote $J(x_k)$ the set of the indices of the active constraints at x_k. Then d is a feasible descent direction if $d^T \nabla f_i(x_k) \leq 0,\ i \in J(x_k)$ and $d^T \nabla f(x_k) \leq 0$. Such a direction can be computed by projecting the negative gradient of f onto the tangent plane to the active constraint surfaces. With an appropriate choice of the step length, this method converges to a KKT solution.

The main advantages of this method lies in its simplicity, but its convergence may be very slow, even for problem that are quite well conditioned.

3.2.3 Newton's method

The Newton's method, also called *Newton-Raphson method*, was originally designed to find the root of a system of equations $S(x) = 0$, with $S : \mathbb{R}^n \to \mathbb{R}^m$. The underlying idea is to solve at each iteration the linear equation obtained by equating to zero the first-order approximation of the function. With $J_S(x)$ the Jacobian matrix of S :

$$x_{k+1} = x_k + \delta_k \text{ such that } J(x_k)\delta_k + f(x_k) = 0$$

It remains to solve the linear system $J_S(x_k)\delta_k = -f(x_k)$. Thus, we obtain a sequence of x_k that converges to a root of S.

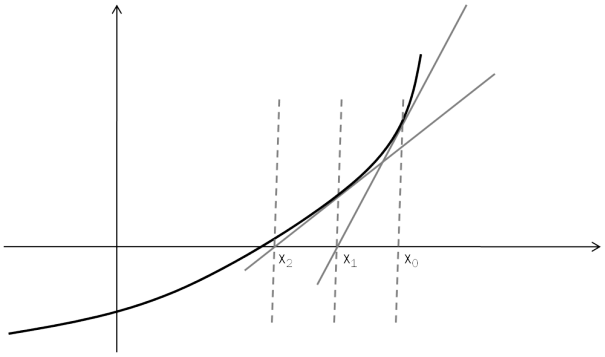

Figure 3.2: Newton's method

This method can easily be extended to unconstrained optimization of a function $f : \mathbb{R}^n \to \mathbb{R}$ by solving the first-order optimality condition : $\nabla f(x^*) = 0$ if x^* is a local minimizer of f for any function f twice differentiable. In this case, $S = \nabla f$ and therefore $J_S(x) = \nabla^2 f(x)$. The system to solve at each iteration is therefore :

$$\nabla^2 f(x_k)^T \delta_k = -\nabla f(x_k)$$

If f is strictly convex, then $\nabla^2 f(x_k) \succ 0$, which ensures that the system has a unique solution.

Remark that this method is equivalent to minimize the second-order approximation of the function at each iteration. This is therefore a *second-order* methods.

The Newton's method can also be applied to an optimization problem with equality constraint : $\min f(x) : Ax = b$, with A a full rank matrix of size p. Indeed, it suffices to solve the KKT system :

$$(KKT) \begin{cases} \nabla f(x) + A^T \lambda = 0 \\ Ax = b \end{cases}$$

By assuming that $Ax_k = b$ and by denoting w the dual variables for the equality constraints, the system to solve at each iteration is as following :

$$(KKT_k) \begin{pmatrix} \nabla^2 f(x_k) & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \delta_k \\ w \end{pmatrix} = \begin{pmatrix} -\nabla f(x_k) \\ 0 \end{pmatrix}$$

The determination of an initial feasible point may sometimes be a challenging task. In this case, there exists a version of the algorithm that includes the necessity of computing such a feasible point.

From a computational point of view, the main difficulty comes from the resolution at each iteration, of the system (KKT_k). An efficient way to do this is based on the variable elimination technique, which requires $O(p^2 n + p^3)$ elementary operations.

In conclusion, let us mention the Quasi-Newton method. As its name suggest, it is very close to the Newton's method but, in order to avoid the repeated computation of the Hessian, the latter is replaced by an estimate based on successive gradients.

3.2.4 Lagrangian methods

Consider the following constrained optimization problem :

$$\begin{cases} \min & f(x) \\ \text{s.t.} & g_i(x) \leq 0, \ i = 1, ..., m \end{cases}$$

Recall that the Lagrangian of this problem $\mathcal{L}(x,\lambda) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x)$ is defined for $\lambda \geq 0$ and the dual function of the problem is $l(\lambda) = \inf_x \mathcal{L}(x,\lambda)$. This function is always concave.

The underlying idea of the classical Lagrangian method is to exploit the concavity of the dual function l and the fact that it is relatively easy to compute a sub-gradient (or a gradient when l is differentiable) of l.

When a saddle-point exists (no duality gap), the algorithm yields an optimal solution of the original problem. Otherwise, it provides only approximate solution of the optimal solution, as well as lower bounds of the optimal value. One of the most famous of these algorithm is the so-called *Uzawa algorithm*.

Combining this approach with the penalty approaches leads to the *Augmented Lagrangian method*. The idea of the penalty approaches consists of converting a constrained optimization problem into an unconstrained one by means of a *penalty function* that to penalize the violation of the constraints. A classical penalty of an equality constraint $g(x) = 0$ being $g(x)^2$, the augmented Lagrangian approach consists of solving a sequence of problem of the form :

$$\min_{x,s\geq 0} f(x) + \sum_{i=1}^{m} \lambda_i (g_i(x) + s_i) + r \sum_{i=1}^{m} (g_i(x) + s_i)^2$$

3.3 Linear Programming

The definition of Linear Programming and the design of the simplex method in 1948 by Dantzig are generally considered as the milestone that sparked off Mathematical Programming. Since that time, Linear Programming has been the most widely used technique of Mathematical Programming, with a variety of scientific and technical applications, such as logistics, scheduling, network or finance. The application of Linear Programming to economic planification even brought the Nobel Prize to Kantorovich in 1975.

In this thesis, this technique is often used as a reference since it offers a relaxation framework for a variety of problems, in particular for Quadratically Constrained Quadratic Programs (QCQP).

3.3.1 Basic results in Linear Programming

Linear Programming can be viewed as the the special case of Conic Programming using the nonnegative orthant $\mathbb{R}^n_+$ as cone. This cone being self-dual, we can easily formulate its dual by applying the duality for Conic Programs 1.1.3:

$$\left\{ \begin{array}{lll} \min_{x\in\mathbb{R}^n} & c^T x & \\ \text{s.t} & Ax = b & \\ & x \geq 0 & \end{array} \right. \overset{dual}{\longleftrightarrow} \left\{ \begin{array}{lll} \max_{y\in\mathbb{R}^m} & b^T y & \\ \text{s.t} & c - A^T y \geq 0 & \end{array} \right. \tag{3.2}$$

The strong duality holds whenever both problems are feasible. Otherwise, if the primal is not feasible, then the dual is unbounded and conversely.

Regarding the complexity, Renegar [226] proved in 1995 that Linear Programming can be solved in at most $O(n^k, L)$ computational steps, where n is the number of variables, k a small constant (3.5 is known with the Karmarkar's interior-point method and 4 for the Ellipsoid method) and L measure the bit-length of the input. In practice, these algorithms perform generally much better that predicted by this bound.

Two major classes of algorithms are available for Linear Programming. The first ones follow the line of the Simplex method devised by Dantzig. This method relies on the fact that the optimal solution

(or at least one of them) is an extreme point of the feasible polyhedra. This algorithm explores these extreme points successively in order to improve the objective at each iteration.

Following the original tabular variant, numerous variants of this algorithm have been proposed, starting with the revised primal and dual simplex, then for instance, versions that handle bound constraints natively within the algorithm. The advantages of this algorithm are its high practical efficiency, although it has been proved to be not polynomial on pathological instances [157], and its warm-start capability.

The second class of methods for Linear Programming gathers the broad range of methods, which, by constrast with the Simplex, reach the optimum by progressing through the interior of the feasible domain. These so-called *Interior-points methods*, are described in detail hereafter.

A synthetic comparison of these methods is difficult. It is generally admitted that the dual simplex method is best on arbitrary instances, but there are some instances where the primal simplex may work best. On degenerate or very large-scale instances, the interior-point methods perform generally better than simplex method.

On the whole the dual simplex method is best for most LP problems, there are some instances where the primal simplex may work best. The barrier method typically should be used for very large sparse models or models that are experiencing numerical difficulties.

Most of the solver make automatically the choice of the algorithm, although it may also be parameterized by the user. Among the numerous commercial solvers available on the market for Linear Programming, we distinguish IBM ILOG CPLEX [143] and Xpress [1], as the most popular. There are also freely-available solvers that perform well, although their efficiency is not comparable yet with above mentioned commercial solvers. Among them are CLP [234] or GLPK [188], to cite only those that performs the best on the recent benchmark [196].

Thus, very large-scale linear programs, with up to millions of variables, constraints, and nonzeros, can be solved. For such problems, it can be convenient to apply decomposition methods. For instance, is the Benders decomposition, that requires the following specific structure :

$$\begin{cases} \min & C_0x_0 + C_1x_1 + C_2x_2 \\ \text{s.t.} A_0^1x_0 + A_1^1x_1 = b^1 & \\ & A_0^2x_0 + A_2^2x_2 = b^2 \end{cases}$$

In other words, there exists a partition of the constraints such that only a subset of variables (x_0) are involved into several subsets of constraints. The Benders methods consists of solving the dual by adding successive violated valid cutting planes. So the approach is called "row generation". In contrast, DantzigâĂŞWolfe decomposition uses "column generation". This approach is based on the following structure :

$$\begin{cases} \min & C_1x_1 + C_2x_2 \\ \text{s.t.} A_1x_1 = b_1 & \\ & A_2x_2 = b_2 \\ & B_1x_1 + B_2x_2 = b_3 \end{cases}$$

Thus, this structure is based on a decomposition of the variables, with only a part of the constraints that is shared. The idea is to use the extreme-point representation of the polytopes $A_ix_i = b_i$ and to replace the variables x_i by a convex combination of these extreme points, that are successively introduced.

It is beyond the scope of this thesis to provide an exhaustive review of this broad topic. We restrict ourselves to a brief review of the interior-point methods, since they were subsequently extended to Conic Programming, and in particular Semidefinite Programming. For more detailed information on this well-studied area, we refer the reader to the references [44, 75, 200].

3.3.2 Interior-point methods for Linear Programming

In contrast to the simplex algorithm, interior-point methods reach the optimal solution by traversing the interior of the feasible set. Interior-point methods for Linear Programming is per se a standalone topic and we restrict ourselves to the most celebrated of these methods, namely the seminal projective method of Karmarkar, the primal-dual path-following and the potential reduction methods. We also give insight to some common tools, namely the Mehrotra's predictor-corrector technique, the consideration of infeasibility and the crossover procedure. The content of this section is mainly based on the following references [15, 192, 238, 260] and for a complete and comprehensive review on this, we point the reader to [215].

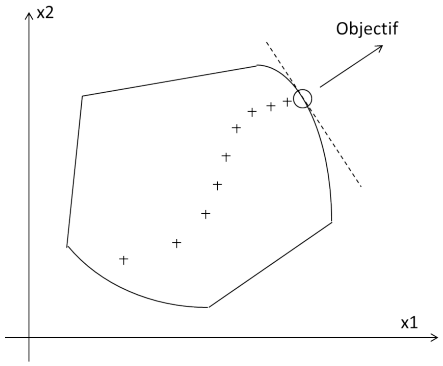

Figure 3.3: Reaching the optimal solution via Interior-Points methods

Let us consider the following Linear Program, written under its standard form and its dual :

$$\left\{\begin{array}{ll}\min & c^T x \\ \text{s.t.} & Ax = b \\ & x \in \mathcal{K}\end{array}\right. \overset{dual}{\longleftrightarrow} \left\{\begin{array}{ll}\max & b^T y \\ \text{s.t.} & c - A^T y = s \\ & s \in \mathcal{K}^*\end{array}\right.$$

where $\mathcal{K}$ is the nonnegative orthant of $\mathbb{R}^n$ and $(A, b) \in \mathbb{R}^{m,n+1}$, with $\text{rank}(A) = m$. For sake of clarity, we denote by K^* the dual cone of $\mathcal{K}$, even if this cone is self-dual, i.e., $\mathcal{K}^* = \mathcal{K}$).

The interior-point methods are also referred to as *barrier methods* since they are based on the use of a *barrier function* of the cones $\mathcal{K}$ and $\mathcal{K}^*$. Such a function has the property of tending to infinity when the its argument tends to the boundary of$\mathcal{K}$ from inside. More formally,

Definition 3.3.1 *Barrier function*
F is a barrier function of $\mathcal{K}$ if it is defined over $\text{int}(\mathcal{K})$ *and*

$$\lim_{x \to \text{bnd}(\mathcal{K})} F(x) \;=\; +\infty$$

The umbrella term "interior-point method" covers a variety of methods. First was the *ellipsoid method*, initially proposed for convex minimization problem by Shor [244]. Then it was adapted to Linear Programming by Khachiyan [153], which proved its worst-case polynomiality ($O(n^4 L)$ and thereby the polynomiality of Linear Programming, which were still an open problem.

In practice, this algorithm performs very poorly, in particular compared to the simplex. However it is nowadays a very important theoretical tool for developing polynomial-time algorithms for a large class of convex optimization problems, as detailed at paragraph 1.2.1.

With this work, Khachiyan spurred a new wave of research in linear programming, the most famous of them being the *projective scaling algorithm* designed by Karmarkar in 1984 [150]. This

breakthrough has not only improved on Khachiyan's method with a worst-case polynomial bound of $O(n^{3.5}L)$), but has outperformed the simplex method on fairly large-scale problems.

The *affine scaling* method, a variant of Karmarkar's algorithm was proposed independently by several researchers. Later it was discovered that this algorithm had already been proposed by Dikin [85] in the 1960s. This method is not very interesting, both from theoretical point of view, since it has not been proved that it converges in polynomial time, and from practical point of view, since it is not competitive with other methods. Furthermore, it has been shown that it coincides with a particular case of the path following method, by targeting directly the limit point of the central path (i.e., $\tau = 0$). For these reasons, we do not get into more details about this method and move on directly to the path following methods.

The underlying idea of the path following methods is to aim at each iteration a point of the central path, i.e., a point of the following set :

$$\{(x \in \mathbb{R}^n, s \in \mathbb{R}^n, y \in \mathbb{R}^m) : \begin{cases} Ax = b \\ c - A^T y = s \\ x_i s_i = \tau, \ i = 1, ..., n \\ x \in \mathcal{K}, s \in \mathcal{K}^* \end{cases} \text{ for some } \tau \geq 0\}$$

Note that for a given value of τ, the associated solution (x, y) are the solution of the perturbed KKT system : instead of requiring $x^T c - x^T A^T y = 0$ as in the original KKT system, we allow a relaxation of this equality parameterized by τ.

At this point, a connection have to be made with the barrier functions, which allows to extend the definition of central path to any conic optimization problem, provided that their cone admits a barrier function F. Then an intuitive way of relaxing the conic constraint, is to add this function, up to a parameter τ, in the objective. In the case of Linear Programming, we show that this yiels the perturbed KKT system. Indeed, a simple barrier function for the nonnegative orthant is the negative logarithm : $F(x) = -\ln(x)$. Thus, we consider the following optimization problems :

$$\begin{cases} \min & c^T x - \tau \sum_{i=1}^{n} \log x_i \\ \text{s.t.} & Ax = b \end{cases} \quad \begin{cases} \max & b \bullet y + \tau \sum_{i=1}^{n} \log s_i \\ \text{s.t.} & c - A^T y = s \end{cases}$$

These two problems are not dual to each other but by expressing the KKT optimality conditions for both problems yields the same system, corresponding to the central path system for the accurate value of τ. τ is called the *complementary gap*.

When τ tends to 0, the central path converges to an optimal solution of the problem, to the *analytic center* to be exact, i.e., the unique optimal solution that satisfies the strict complementarity condition : $x^* + s^* > 0$. The underlying of the path-following method is to use the central path as a guideline to reach the optimal solution while remaining in the feasible region. At each iteration k, the method choose a target on the central path, corresponding to a updated complementary gap τ_k, and moves towards it by stepping along Newton's direction while staying in the feasible region. Global convergence of the algorithm to the strict complementary solution is ensured by a proper choice of the shrinking sequence τ_k and of the step size of the Newton line-search.

The potential reduction methods offer an alternative way to compute this step size. They rely on the definition of a function measuring the quality (or potential) of a solution by combining proximity to the set of optimal solutions and centrality within the feasible region. Then, it suffices to establish a guarantee on the reduction of this potential function at each iteration to obtain a bound on the number of iterations required to reach the optimal solution at the desired precision.

To conclude, we discuss some generalities about interior-point methods. In the classical variant, the successive solutions are all strictly feasible. Consequently, since the optimal solution lies on the frontier of the feasible set, it will never be attained. However, beyond a given proximity, we may

consider that the obtained precision is sufficient. We may also apply a rounding procedure yielding the exact optimal solution.

However there is also a class of interior-point method than handle infeasible incombent solutions. In particular, it avoids the necessity of computing a feasible solution for initialization, which might be problematic. Indeed, an infeasible interior-point method can start from any interior point of the positive orthant. Another way to circumvent the initialization problem is to resort to the self-dual embedding technique.

Moreover, each variant may work in three spaces : the primal space, the dual or their cartesian product : the primal-dual space.

Among all the possible approaches, the primal-dual path-following method including a number of enhancements such as Mehrotra's predictor-corrector technique, has been the most popular since the 1990s. For practical purpose, since it has proven to be the most effective and for theoretical purpose, since it can be readily extended to general conic programs whenever the cone in question admits a self-concordant barrier function.

As a last remark, we point out that the time complexity of the interior-point methods is measured with respect to the length of the binary encoding of the input, which requires that the data are integer or rational numbers. It is not known at present whether or not there are algorithms for LPs whose running time is polynomial and depends solely on the number of variables and constraints of the problem.

3.3.2.1 Projective algorithm of Karmarkar

In 1984, Karmarkar [150] introduced a new interior-point algorithm for Linear Programming, which, in constrast to the Ellipsoid method, seemed to be competitive with the simplex method. This triggered a revolution in the field of Linear Programming and led researchers to reconsider this family of methods.

One of the novelty brought by this method was the notion of *potential function*, that does not interfere directly within the algorithm but is a tool for analysing the algorithm and proving its convergence and its polynomiality.

This algorithm works on problem of the form :

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = \mathbf{0} \\ & e^T x = 1 \\ & x \geq 0 \end{array}$$

where $A \in \mathbb{R}^{n \times m}$ is a rank m matrix and $c \in \mathbf{R}^n$. Without loss of generality, it is assumed that $Ae = 0$ and the optimal value of the problem equals 0.

The basic idea is transform the problem via a projective affine scaling map, so that the current solution is transformed into the "central point" e, then to take a step along the projected steepest-descent direction in the transformed space, and finally to map the resulting point back to its corresponding position in the original space.

Given the current solution x^k, the transformation maps a vector $x \in \mathbb{R}^n$ to a vector $p(x) = \frac{\text{Diag}(x^k)^{-1}x}{e^T \text{Diag}(x^k)^{-1}x}$.

It can be viewed as the combination of two operations : the first scale the variables so that the current point goes to e, the second scales each resulting point by the sum of its variables so that $\sum_{i=1}^{n} p(x)_i = 1$, which comes to project the feasible set onto the simplex $S = \{x \in \mathbb{R}^n : e^T x = 1\}$.

This algorithm runs in $O(nL)$ iterations, which has been superseded since then by other interior-point methods, as detailed below. It is very close to the primal affine scaling algorithm, where the transformation made at each iteration is defined by $p(x)_j = \frac{x_j}{x_j^k}$, but this algorithm is not believed to be polynomial in the worst-case. It has also been shown [103] that Karmarkar's algorithm is equivalent to the path following method with a particular choice of the barrier parameter.

3.3.2.2 Primal-dual path-following algorithm

The underlying idea of path following algorithm is attributed to Meggido [191], Kojima, Mizuno and Yoshise [160] and Adler and Monteiro [144]. It consists of solving the perturbed KKT system by Newton's method and shrinking the perturbation in order to tend to the original KKT system.

$$\{(x \in \mathbb{R}^n, s \in \mathbb{R}^n, y \in \mathbb{R}^m) : \begin{cases} Ax = b \\ c - A^T y = s \\ x_i s_i = \tau, \ i = 1, ..., n \\ x \in \mathcal{K}, s \in \mathcal{K}^* \end{cases} \quad \text{for some } \tau \geq 0\}$$

We recall that the set of solution of these perturbed KKT systems is called *central path* and that the level of perturbation is the *complementary gap*.

From a current iterate x^k and a targeted complementary gap τ_k, the algorithm works as following :

- The search direction is computed by performing a Newton's step ;
- The new iterate is computed by moving along this direction, with a step size α_k ;
- The new targeted complementary gap τ_{k+1} is computed from the *duality measure* μ_k of the new iterated : $\tau_{k+1} = \sigma_k \mu_k$

Note that only one Newton step is made at each iteration. Thus, the central path is not attained, the aim being only not to deviate too far from it.

The *duality measure* of a primal-dual solution (x, y) equals $(x^T s)/n$. It vanishes at optimal and is equal to τ on the corresponding point of the central path. Then two parameters, α_k and σ_k have to be settled at each iteration. They are correlated, since large values of σ_k implies potentially a loss of centrality and the necessity of taking small values for α_k in order to keep the current solution in the neighborhood of the central path. These choices are usually made on a heuristic basis.

Regarding the complexity, these methods have been proved to require $O(\sqrt{n}L)$ iterations, with at most $O(n^3)$ arithmetic operations at each iteration, which brings the whole complexity to $O(n^{3.5}L)$. There are some special case where this can be reduce, as for instance [144] with a whole complexity of $O(n^3 L)$.

3.3.2.3 Potential-reduction methods

This section is mainly based on the excellent survey of Todd on the potential-reduction methods [253]. These methods are based on the idea of reducing a so-called *potential function* at each iteration. This function combines the objective function with a measure of distance from the constraints boundaries, or, equivalently, a measure of the *centrality* of the solution. The rational for this being that the more a solution is central, the more it can be improved at the next iteration.

This potential function is used both to measure the quality of the current solution and to determine how to improve it to generate the next iterate. This choice is guided solely by the objective of minimizing this function. Furthermore, if you can compute a guarantee on the decrease of this function at each iteration, then we obtain a guarantee on the number of iterations required to attain the optimal solution within a given accuracy.

We consider the linear program in standard form defined at 3.2.

The centrality is measured by using *barrier function*, as defined above. For example the function $F(x) = -\sum_{i=1}^{n} \ln(x_i)$ is a barrier function for the nonnegative orthant, since it tends to $+\infty$ when x approaches its boundary. This is not the only barrier function for this set, think for instance to the inverse function $\sum_i 1/x_i$. But the advantage of the logarithmic function lies in its self-concordance property, a property that allows to bound the errors on the Taylor approximation of F and ∇F.

Incorporating the minimization of the objective leads to the potential function ϕ, parameterized by $\rho \geq n$ and ξ a lower bound of the optimal value :

$$\phi(x) = \rho \ln(c^T x - \xi) + F(x)$$

The key point is that ϕ automatically increases the importance of the objective part as optimality is approached. This is to compare with the equivalent of the path-following method :

$$\phi(x) = (c^T x - \xi) + \mu F(x)$$

where the parameter μ has to be reduced "by hand", otherwise the importance of objective part would vanishes when $c^T x - \xi$ shrinks.

A primal-dual potential function is :

$$\phi(x, s) = \rho \ln(x^T s) + F(x) + F(s)$$

The first symmetric pure potential-reduction algorithm was proposed by Kojima, Mizuno and Yoshise in [159]. The idea is to apply the steepest descent to the minimization of the function ϕ.

In conclusion, for the primal-dual potential reduction with $\rho = N + O(\sqrt{n})$ is reduced at each iteration of a positive absolute constante δ, then the ϵ accuracy is attained within $O(\sqrt{n} \ln(1/\epsilon))$ iterations.

3.4 Mixed-Integer Linear Programming

As earlier mentioned, most of the combinatorial problems can formulated as optimization problem with integer variables. Among them are the Mixed-Integer Linear Programs (MILP) :

$$\begin{cases} p^* = & \min_{x \in \mathbb{R}^n} & c^T x \\ & \text{s.t.} & Ax \leq b \\ & & x_i \in \mathbb{Z},\ i \in I \subset [n] \end{cases} \tag{3.3}$$

One main reason for the success of MILP is its huge modelling flexibility, that allow to model a wide range of applications [200].Another one is the existence of effective solvers [143, 1] that handle problems with hundreds of thousands of variables and constraints in a matter of minutes.

These problems are obviously non convex, since their feasible set is not even connected. As for general nonlinear programs, most of the methods to solve these problems rely on convex relaxations, the most natural being the following linear relaxation :

$$\begin{cases} p^* = & \min_{x \in \mathbb{R}^n} & c^T x \\ & \text{s.t.} & Ax \leq b \end{cases}$$

If the matrix A is totally unimodular, then the solution of this relaxation is integer and therefore, is the optimal solution of the problem (3.3). Apart from this case, the problem is generally NP-hard.

Most MILP can be formulated in several ways. Moreover, the choice of this formulation is of crucial importance to solving the model. Indeed, efficiency of enumerative methods are highly dependent on the sharpness of the linear relaxation, which varies w.r.t the formulation.

We illustrate the versatility of the formulation on the following example. We consider a binary vector $x \in \{0, 1\}$ connected by a disjunctive constraint, i.e., a constraint $x \in \mathcal{P}_1 \cup \mathcal{P}_2$), with $\mathcal{P}_1 = \{x \in \mathbb{R}^n : A_1 x \leq b_1\}$, $\mathcal{P}_2 = \{x \in \mathbb{R}^n : A_2 x \leq b_2\}$ two polyhedra.

A first formulation require the introduction of two auxiliary binary variables $y_i,\ i = 1, 2$, that codes 0 if $x \in \mathcal{P}_i$. Then a possible formulation is :

$$\begin{cases} A_1x - M_1y_1 \leq b_1 \\ A_2x - M_2y_2 \leq b_2 \\ y_1 + y_2 \geq 1 \end{cases}$$

The second formulation exploits the fact that x can take only a finite number of values. Indeed, we may enumerate all the values of x, $\tilde{x}$, that do not satisfy the constraint and impose $x \neq \tilde{x}$:

$$\left\{ \sum_{i=1}^{n} (\tilde{x}_i + (-1)^{\tilde{x}_i} x_i \geq 1, \forall \tilde{x} \in \{0,1\}^n, \tilde{x} \notin \mathcal{P}_1 \cup \mathcal{P}_2 \right)$$

Clearly, this formulation contains a larger number of constraints, but its linear relaxation is sharper.

More generally, there are usually several polyhedra P such that $\mathcal{F} = P \cup \mathbb{Z}^I$. The choice of P is decisive for getting a tight bound on the optimal value since the smaller is P, the tighter is the linear relaxation. Ideally, $P = \text{conv}(\mathcal{F})$ because in this case, the linear relaxation yields the optimal value of the MILP. A fundamental line of research for solving MILP aims at determining some valid constraints, called *cutting planes*, that reduce P. However, describing totally the convex hull of $\mathcal{F}$ is generally too complicated to determine and may involve an exponential number of constraints. In this case, we have to work with a "relaxed" description of the convex hull and it is necessary to combine this approach with an enumerative method to get the optimal solution.

In the sequel, we start by presenting the basic enumerative method, namely the Branch & Bound method. Then, we provide some theoretical insights about polyhedra and generic methods to determine cutting planes. We provide a comparison of these methods. Finally, we show how combining these two approaches within a Branch & Cut method.

3.4.1 Branch & Bound for MILP

First proposed by Land and Doig [167] in 1960, this algorithm is the most widely used tool for solving discrete optimization problems : $\min_{x \in \mathcal{F}} f(x)$ where $\mathcal{F}$ is a finite discrete set. This algorithm applies thus in a more general framework than MILP.

The above problem necessarily admits a solution, since $\mathcal{F}$ is finite, but finding it by enumerating all the elements of $\mathcal{F}$ is generally irrelevant since $\mathcal{F}$ may contain a high number of elements that it might be time-consuming to identify and evaluate. The Branch & Bound algorithm is a general paradigm that consists of a generic strategy for exploring the feasible set and of a way to discard massively some fruitless solutions, in order avoid the systematic enumeration of all the solutions. It relies on two key procedures :

- **Branching** : splits the feasible set $\mathcal{F}$ into two or more smaller set whose union covers $\mathcal{F}$. Its recursive application generates a tree structure of subsets of $\mathcal{F}$.
- **Bounding** : computes upper and lower bounds for the optimal solution on a given subset of $\mathcal{F}$. The lower bound can differ per subset whereas the upper bound holds for all the subsets since it is the cost of the best known solution.

Then the key idea is to eliminate a feasible subset whenever its lower bound is equal or larger than the current upper bound. This step is called pruning and allow to fathom the subproblem without solving it. The recursion stops when the current subset is reduced to a single element, or when the upper bound match the lower bound. Either way, a minimum over $\mathcal{F}$ is attained.

A simple branching operation for MILP is to pick a variable and to split its definition set into two subset :

$$x_i \in [a,b] \cup \mathbb{Z} \rightarrow \begin{cases} x_i \in [a,s] \cup \mathbb{Z} \\ x_i \in [s+1,b] \cup \mathbb{Z} \end{cases} \quad \text{for any } s \in [a,b] \cup \mathbb{Z}$$

Generally x_i is chosen as the most fractional components of the current solution. By denoting $\tilde{x}_i$ its value, then $s = \lfloor \tilde{x}_i \rfloor$.

Regarding the lower bound, it is generally computed by linear relaxation, but some other relaxation techniques (Lagrangian, ...) are possible. In the case where the objective function is not convex, a relaxation can be obtained by replacing f by a lower function g such that $g(x) \leq f(x), \forall x \in \mathcal{F}$.

Once the branching is done, that subdivided the current subset into two or more subset (or node) to be investigate, two strategies, *eager* or *lazy*, are possible for selecting the node to explore. In the eager strategy, bounds are calculated as soon as nodes are available, then each non discarded subset is stored in a pool of live nodes together with its bound. The lazy strategy consists of chosing one node, computing its lower bound and exploring it if it is not discarded.

3.4.2 Polyhedral combinatorics and cutting planes

We consider the minimization of a linear objective $c^T x$ over a finite discrete set $\mathcal{F}$. As already noticed, this problem is equivalent to the minimization of $c^T x$ over $\text{conv}(\mathcal{F})$. The objective of the polyhedral combinatorics is to describe the polytope $P = \text{conv}(\mathcal{F})$ in terms of linear inequalities : $\text{conv}(\mathcal{F}) = \{x \in \mathbb{R}^n : Fx \leq h\}$.

Such a representation necessarily exists. Indeed, any polytope P can be specified in two ways : as the convex hull of its vertex set $\mathcal{V}$ ($\mathcal{V}$-representation, see Corollary 2.2.55) , or as the intersection of the set $\mathcal{H}$ of its facet-inducing halfspaces ($\mathcal{H}$-representation). In theory, we can always convert from one representation to another. This would be very useful to find the $\mathcal{H}$-representation of the convex hull of a set of points. In practice, determining this representation might not be efficient from a computational point of view. For that reason, we generally only determine some valid inequalities that strengthen the linear relaxation. Such inequalities are called *cutting planes*.

3.4.2.1 Chvatal-Gomory hierarchy

The umbrella term of *cutting planes* was introduced by Gomory in [111]. For the first time, he proposed a general method to determine cutting planes, i.e., that does not depend on the particular problem structure.

This method applies to Integer Linear Programs, i.e., $\mathcal{F} = \{x \in \mathbb{Z}^n : Ax \leq b\}$, and relies on the fact that an integer linear combination of integer is integer. Consequently, if the inequality $d^T x \leq e$ is valid, with d integer, then $d^T x \leq \lfloor e \rfloor$ is also valid for $\mathcal{F}$.

By adding all the inequalities obtainable this way, we obtain a new polytope P^1, the so-called *elementary closure* of P, such that :

$$\text{conv}(\mathcal{F}) \subset P^1 \subset P = \{x \in \mathbb{R}^n : Ax \leq b\}$$

A recursive iteration of this process, i.e., $P^2 = (P^1)^1$, provides a hierarchy of relaxation of $\mathcal{F}$. In [74], Chvátal proved that this hierarchy converges to $\text{conv}(\mathcal{F})$ in a finite number of steps.

This method is remarkable since it has opened the door to cutting planes based approach but it suffers from two major shortcomings. First the number of iterations required for attaining $\text{conv}(\mathcal{F})$ might be very large and depends not only on the size of the problem but also on the coefficients of the system $Ax \leq b$.

The other difficulty comes from the fact that the separation problem is NP-hard. In other words, for an incumbent solution $\tilde{x}$, finding a violated inequality or showing that there are not, can not be done in polynomial time, because there is an exponential number of such inequalities. Consequently, there is no known way to implement this method in polynomial time.

However, these cuts are used in one way or another in most of the commercial MILP solvers. In particular, the simplex tableau is easily suitable for generating such cuts. Despite this fact, these cuts are dominated by other general cuts for MILP, among them the cuts generated from the well-known Lift & Project scheme.

3.4.3 Hierarchies of relaxation for MILP : the Lift & Project approach

We consider the minimization of a linear objective $c^T x$ over a discrete set defined as the intersection of a polyhedron P with the integer lattice : $\mathcal{F} = P \cup \mathbb{Z}^n = \{x \in \mathbb{Z}^n : Ax \leq b\}$.

A hierarchy of relaxation is a succession of set P^r such that $P^{r+1} \subseteq P^r$ and $\mathcal{F} = P^r \cup \mathbb{Z}^I$ for any rank r :

$$P^r \subseteq P^{r-1} \subseteq ... \subseteq P^1 \subseteq P$$

Five hierarchies of relaxation, all converging to $\text{conv}(\mathcal{F})$, are described in this thesis. We already presented the seminal Gomory-Chvatal hierarchy (GC-hierarchy) in the previous paragraph, and in this paragraph we describe the purely linear hierarchies, i.e., the Balas-Ceria-Cornuejols hierarchy (BCC-hierarchy) and the Sherali-Adams hierarchy (SA-hierarchy), as well as the Lift & Project scheme that underlies these hierarchies.

The linear and semidefinite version of the Lovász-Schrijver hierarchy (LS and LS_+ hierarchy), as well as the Lasserre hierarchy, also a semidefinite one, are described in paragraph 3.3.3.2 and 3.4.3 respectively. A comparison of SA-hierarchy, LS and LS_+-hierarchy and Lasserre hierarchy is provided in paragraph 3.4.3.3.

3.4.3.1 Lift & Project scheme

The idea of Lift & Project for 0/1-LP was introduced by Sherali and Adams in [240]. The overarching idea is that the projection of a polytope may have more facets than the polytope itself. Even if the polytope P has exponentially many facets, we may be able to represent it as the projection of a polytope Q in higher (but still polynomial) dimension, having only a polynomial number of facets.

This approach tends to describe the convex hull of the feasible set in two steps. The first step consists of converting the 0/1-LP into an equivalent problem that involve additional variables and constraints. We say that the problem is *lifted* into a higher dimensional space. The constraints that are added have to exploit the integer nature of the original variables.

In a second step, the problem is *projected* back on to the original space in order to get rid of the new variables. Generally, the whole polyhedral representation of the projection is not computed, it suffices to solve the separation problem, i.e., find a valid constraint violated by the current relaxed solution. In theory, if we are able to compute such a constraint in polynomial time, then in virtue of the equivalence between separation and optimization, we are also able to optimize in polynomial time. In practice, this constraint are used as cutting planes to reinforce the linear relaxation.

Regarding the projection step, we recall (see also paragraph 2.2.4.1) that the projection of the polyhedron $P = \{(x, y) \in \mathbb{R}^{n+n'} : Ax + By \leq c\}$ onto the x-space is the set

$$\text{Proj}_x(P) = \{x \in \mathbb{R}^n : (x, y) \in P \text{ for some } y \in \mathbb{R}^{n'}\}$$

and the projection cone of P associated with x is the following polyhedral cone :

$$W = \{u : uB = 0, u \geq 0\}$$

Then, any element of W defines a valid constraint for $\text{Proj}_x(P)$: $uAx \leq uc$. This comes to make positive combination of the inequalities of P so that the coefficients of y vanish. Not only these constraints are valid, but the Balas theorem states that considering only the constraints generated by extreme rays of W is sufficient to describe $\text{Proj}_x(P)$.

In practice, we rather solve the separation problem. If $\tilde{x}$ is the current solution :

$$\begin{cases} \max & u(A\tilde{x} - c) \\ \text{s.t.} & uB = 0 \\ & u \geq 0 \end{cases}$$

The extension of this projection scheme to any set defined as the intersection of a cone with hyperplanes is described at paragraph 3.3.3.2.

3.4.3.2 Sherali-Adams hierarchy

In the Sherali-Adams Lift & Project, the lift step relies on the idea of multiplying some linear inequalities between them, in order to make some quadratic terms appear. At rank 1 of the Sherali-Adams Lift & Project, the linear constraints are multiplied by all the bound constraints : $x_i \geq 0$ and $1 - x_i \geq 0$ for all variables x_i.

Then, replacing the square of the binary variables by themselves reinforces the continuous relaxation. This is illustrated on a very simple example. We aim at minimizing x over $\mathcal{F}_0 = \{x \in \{0,1\} : x \geq \frac{1}{2}\}$. The continuous relaxation gives $x^* = 1/2$. By multiplying $1 - x \geq 0$ and $x \geq 1/2$, we get :

$$x(1-x) \geq 1/2(1-x) \Leftrightarrow 0 \geq 1 - x \Leftrightarrow x \geq 1 \Rightarrow x = 1$$

To handle the obtained nonlinearities, some new variables y_{ij} are introduced to replace the product $x_i x_j$, except for x_i^2 which is replaced by x_i. The constraint $y_{ij} = x_i x_j$ is relaxed but, by assuming that the system $Ax \leq b$ include explicitly the bound constraints $0 \leq x_i \leq 1$, we get the following relaxation :

- $y_{ij} \geq 0$;
- $y_{ij} \leq x_i$;
- $y_{ij} \leq x_j$;
- $y_{ij} \geq x_i + x_j - 1$.

These are well-known inequalities, introduced by Fortet [94] for binary variables and generalized by McCormick [190] for bounded variables.

With binary variables, this relaxation is sufficient to impose $y_{ij} = x_i x_j$. Consequently, these problems are strictly equivalent to the original problem. The difference comes from their linear relaxation. One could think of solving directly this "lifted" relaxation before projecting back the obtained solution. This is possible but not necessarily efficient since the problem size increases tremendously. The projection step, based on the idea of combining the constraints so as to generate a valid constraint that does not involve the additional variables, allows to overcome this difficulty.

Consider the following 0/1-LP : $\min c^T x : a_j^T x - b_j \leq 0,\ j = 1, ..., m, x \in \{0,1\}^n$. Briefly, the Sherali-Adams relaxation of rank $r \leq n$ is obtained through the following steps :

1. define the set $\mathcal{K}_r = \{(I,J) \subset [n] \times [n] : I \cap J = \emptyset,\ |I| + |J| = r\}$ and the factors $f_{I,J}(x) = \prod_{i \in I} x_i \prod_{i \in J}(1 - x_i)$ for all $(I,J) \in \mathcal{K}_r$;
2. relax the feasible set of the problem into $\{x \in [0,1] : f_{I,J}(x)(a_j^T x - b_j) \leq 0,\ j = 1, ..., m, \forall (I,J) \in \mathcal{K}_r\}$;
3. exploit the binarity of the variables x_i by replacing all x_i^k by x_i for $k \geq 1$;
4. linearize the obtained feasible set by introducing some new variables : $y_H = \prod_{i \in H} x_i, \forall H \subset [n]$, with $y_{\{i\}} = x_i$;

At this point, the problem can either be solved in the *lifted* space, or being projected to derive some valid inequalities in order to tighten the original problem. In practice, the projection consists of getting rid off the variable y and therefore the constraints are determined as valid combination of the *lifted* constraints that do not involves the variables y. This method was initially called *Reformulation-Linearization Technique(RLT)* [2, 241] since the Lift step can be seen as a reformulation followed by a linearization.

3.4.3.3 The BCC hierarchy and its connection with the disjunctive cuts of Balas

In its seminal paper [18], Balas designed a systematic method based on disjunctive programming to find valid inequalities for any generic MILP, in order to separate a given point from the feasible set. In [21], this work was embedded in the more general framework of Lift & Project.

We consider the following ILP (for the sake of simplicity we don't consider the "mixed-integer" case here, but the results can be easily extended to MILP) :

$$(P)\begin{cases} \min & c^T x \\ \text{subject to} & Ax \leq b \\ & x \in \mathbb{Z}^n \end{cases} \tag{3.4}$$

We note $\mathcal{F} = \{x \in \mathbb{Z}^n : Ax \leq b\}$, $K = \{x \in \mathbb{R}^n : Ax \leq b\}$ and (P_r) the relaxation $\min_{x \in K} c^T x$. Solving the relaxation (P_r) generally leads to a fractionary solution x^r, i.e. such that $x_i^r \in]k, k+1[$ for at least one index i. Then a possibility to eliminate this solution is to consider the following disjunction :

$$(P_d)\begin{cases} \min & c^T x \\ \text{subject to} & x \in K \\ & x_i \leq k \bigvee x_i \geq k+1 \end{cases}$$

Some other linear disjunctions are possible, for instance in the case of a 0/1-LP with a constraint of the form $\sum_{i \in I} x_i = 1$, with $I \subset [n]$, the following disjunction is valid : $\bigvee_{i \in I} x_i = 1$. Another possibility for a 0/1-LP is $(\alpha x \leq \beta - 1) \bigvee \alpha x \geq \beta$ for any $(\alpha, \beta) \in \mathbb{Z}^{n+1}$.

Such a disjunctive problem is not tractable by a linear solver. Furthermore its feasible set is disjoint and therefore nonconvex. The major contribution of Balas is to provide a method to optimize over the convex hull of this set, or more precisely, to determine an inequality valid over the convex hull of this set, that maximize the violation of x^r. This constitutes the so-called *separation step.*

This principle is illustrated on the following figure :

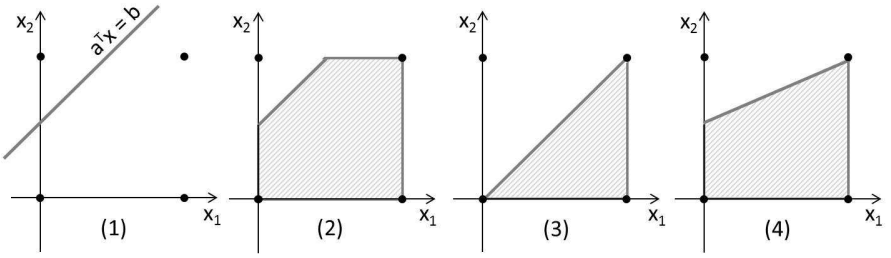

Figure 3.4: Illustration of the disjunctive cuts principle : (1) 0/1 LP, (2) Feasible set of the continuous relaxation K, (3) Convex hull of the feasible set, (4) Convex hull of the feasible set of the disjunctive cut on x_1

It remains to explain how we obtain a polyhedral description, at least partial, of this convex hull. In other words, we seek for linear inequalities valid over the following set :

$$\text{conv}(\{\bigvee_{i=1}^{m} D^i x \leq d^i\})$$

By definition of the convex hull, x^r belongs to this set if and only if there exists a vector λ and some vectors x^i such that :

$$\begin{array}{l} \sum \lambda_i x^i = x^r \\ D^i x^i \leq d^i,\ i = 1, ..., m \\ \sum_{i=1}^{m} \lambda_i = 1 \\ \lambda_i \geq 0,\ i = 1, ..., m \end{array}$$

As such, these conditions are not tractable by a linear solver since they involve some products of variables in the first equality : $x^i \lambda_i$. The idea of Balas was to replace this product by a new variable y^i :

$$\begin{array}{l} \sum y^i = x^r \\ D^i y^i - d^i \lambda_i \leq 0,\ i = 1, ..., m \\ \sum_{i=1}^{m} \lambda_i = 1 \\ \lambda_i \geq 0,\ i = 1, ..., m \end{array}$$

Consequently, any feasible solution of the problem (3.4) must belong to the projection of following polyhedra onto the x-space :

$$\left\{ \begin{array}{l} Ax \leq b \\ Ay^i \leq b\lambda_i,\ i = 1, ..., m \\ \sum y^i = x \\ D^i y^i - d^i \lambda_i \leq 0,\ i = 1, ..., m \\ \sum_{i=1}^{m} \lambda_i = 1 \\ \lambda_i \geq 0,\ i = 1, ..., m \end{array} \right. \tag{3.5}$$

Consequently, we aim at finding a valid linear inequality for the projection of this polyhedra, that is violated by x^r. We apply the Theorem 2.2.61 that comes to search for a valid combination of the constraints where the coefficient of the variables y and λ be equal to zero, so as to maximize its violation of x^r. We recall that a valid combination of constraints is a weighted sum, where the coefficient are non-negative or non-positive according of the sens of the inequality. Coefficients associated to equality constraints are unconstrained, since multiplying a equality by any scalar does not alter its validity. The coefficients that allow such a projection belongs to the so-called *projection cone*, see [18, 20].

Note that the polyhedral description obtained by this projection method is not facet-inducing and may even have redundant inequalities.

In 1993, Balas, Ceria and Cornuejols published a paper [21] where they embedded this approach in the more general framework of Lift & Project. Regarding the lift step, instead of multiplying the linear constraints by all the binary variables and their complement, as in the SA-hierarchy, the linear constraints are multiplied by one single binary variable x_{i_0} and its complement $1 - x_{i_0}$.

Then the process is reiterated by recursion, i.e. the MILP resulting of an iteration is used as original problem of the next iteration and the procedure is re-applied with another choice for i_0. This leads to the obtention of the so-called *BCC hierarchy* of relaxation that attains the convex hull in at most n iterations, where n is the number of binary variables.

The connection with disjunctive programming is made through the following theorem. By denoting $\mathcal{F}_{i_0}$ the feasible set of the reformulated problem of Balas for the index i_0, we have :

Theorem 3.4.1 *The projection of $\mathcal{F}_{i_0}$ onto the x-space is equal to*

$$\text{conv}\{\mathcal{F} \cup \{x : x_j \in \{0, 1\}\}$$

3.4.4 Combining enumerative and cutting planes approach : the Branch & Cut algorithm

Combining enumerative algorithms and cutting planes approaches leads to the so-called Branch & Cut hybrid approaches. In this algorithm, at each node, the current linear relaxation is strengthen iteratively

by adding cutting planes until the relaxed solution be integer. Otherwise it continues with the Branch & Bound exploration once no more cutting planes are found. The efficiency of this method largely depends upon the quality of the cutting planes used, that can roughly be divided into two strands : the general purpose techniques, that we just described, and techniques dedicated to the problem structure.

Note that the generated cutting planes may be either global cuts, i.e., valid for all feasible integer solutions, or local cuts, meaning that they are valid only for the currently considered subset of the original feasible set.

3.5 Quadratically Constrained Quadratic Programming

A Quadratically Constrained Quadratic Programming (QCQP) is an optimization problem where the objective and constraints functions are quadratic :

$$(P)\begin{cases} \min_{x\in\mathbb{R}^n} & x^T P_0 x + 2p_0^T x + \pi_0 \\ \text{subject to} & x^T P_j x + 2p_j^T x + \pi_j \leq 0,\ j = 1, ..., m \end{cases} \tag{3.6}$$

where $P_j \in \mathbb{S}^{n\times n}, p_j \in \mathbb{R}^n, \pi_j \in \mathbb{R},\ j = 0, ..., m_q$ are the problem parameters. The feasible set of this problem is denoted by $\mathcal{F}$ and p^* is its optimal value.

When all the matrices P_i are psd, (P) is convex. Otherwise it may harbor many local minimal and is NP-hard [141]. To see this, one only need to notice that it generalizes many difficult problems as Polynomial Programming or Mixed 0-1 Linear Programming, since the binary constraints can be treated as two quadratic inequalities : $x_i \in \{0, 1\} \Leftrightarrow \{x_i^2 \leq x_i, x_i^2 \geq x_i\}$.

QCQP arises directly in a wide range of practical applications [59, 125], partly due to their ability to model Euclidean distances. Moreover, this optimization problem is central to well-known iterative methods such as trust-region sequential quadratic programming. Finally, it is worth noticing that any polynomial can be reformulate into a quadratic function by adding new variables and constraints, and therefore QCQP can be extended to all the polynomial optimization. For all these reasons, it is now considered as one of the most challenging optimization problems and an important work has been carried out to solve this general problem and its special cases.

Methods for solving such problems, for instance Branch & Bound [183] or Branch & Cut [17], generally need to solve convex relaxations of restricted variants, or approximations, of the problem. The main strength of QCQP w.r.t to general NLP, is that there exists two easily computable way of computing convex relaxations. The first one relies on SDP and is discussed in Paragraph 3.3.2. The second one is the reformulation-linearization technique (RLT) detailed in Paragraph 3.5.3. These relaxations are compared by Anstreicher in [14].

Although not detailed here for sake of brevity, the Generalized Bender's decomposition also provide an interesting framework for solving such problems [102].

3.5.1 Duality

The Lagrangian of the problem (3.6) is $\mathcal{L}(x, \lambda) = x^T(P(\lambda)x + 2p(\lambda)^T x + \pi(\lambda)$ with $P(\lambda) = P_0 + \sum_{i=1}^m \lambda_i P_i,\ p(\lambda) = p_0 + \sum_{i=1}^m \lambda_i p_i$ and $\pi(\lambda) = \pi_0 + \sum_{i=1}^m \lambda_i \pi_i$. It is a quadratic function w.r.t. x. Consequently, the dual problem can be written :

$$\max_{\lambda\geq 0} \min_x x^T(P(\lambda)x + 2p(\lambda)^T x + \pi(\lambda)$$

This problem can be written as the following SDP :

$$\begin{array}{ll} \max_{\lambda\geq 0} & d \\ \text{s.t.} & \begin{pmatrix} \pi(\lambda) - d & p(\lambda)^T \\ p(\lambda) & P(\lambda) \end{pmatrix} \succcurlyeq 0 \end{array}$$

Strong duality holds if the problem is convex or if there is only one quadratic constraints. To see this, consider the following reformulation of the problem (3.6) :

$$\begin{array}{ll} \max & p \\ \text{s.t.} & x^T P_j x + 2p_j^T x + \pi_j \leq 0,\ j = 1, ..., m \Rightarrow x^T P_0 x + 2p_0^T x + \pi_0 \geq p \end{array}$$

Thus, we are exactly in position to apply the S-Lemma (Lemma 3.1.1). Indeed, it provides a sufficient condition for the primal "implication" constraint to hold and this condition is exactly the dual LMI constraint. This condition is also necessary if $m = 1$ or if the functions $f, g_1, ..., g_m$ are convex, since we are therefore in the situation of applying the Farkas' Theorem (Theorem 2.3.49).

3.5.2 Convex case

The case of a convex QCQP, i.e. where all the matrices P_i are psd, was studied by Hao [125] under theoretical and computational aspects, an interior-point method was proposed in [4] to solve this polynomial-time solvable problem and the conversion into Second-Order Conic Programming (see Paragraph 1.3.1) was established in [185]. Thus, methods for solving such problems are well managed and available in several solvers.

3.5.3 Reformulation-Linearization Technique

This technique, widely used to get linear relaxations of quadratic problems, consists of replacing the bilinear terms xy by its convex or concave envelope over a rectangular region :

Theorem 3.5.1 *McCormick [6, 190] The convex lower envelope and concave upper envelope of the function $f(x, y) = xy$ over the rectangular region $\{(x, y) : l_x \leq x \leq u_x,\ l_y \leq y \leq u_y\}$ are given by the expressions :*

$$\begin{array}{l} \max_{\{} l_y x + l_x y - l_x l_y,\ u_y x + u_x y - u_x u_y\} \\ \min_{\{} u_y x + l_x y - l_x u_y,\ l_y x + u_x y - u_x l_y\} \end{array}$$

Then it suffices to replace each product $x_i x_j$ by a new variable y_{ij} and approximate the equality $y_{ij} = x_i x_j$ by imposing that y_{ij} lies between the convex lower envelope and concave upper envelope of xy. This leads to the so-called *McCormick inequalities* :

$$\begin{array}{l} y_{ij} \geq u_j x_i + u_i x_j - u_i u_j \\ y_{ij} \geq l_j x_i + l_i x_j - l_i l_j \\ y_{ij} \leq u_j x_i + l_i x_j - l_i u_j \\ y_{ij} \leq l_j x_i + u_i x_j - u_i l_j \end{array}$$

This step is called *linearization* whereas the introduction of y_{ij} is the *reformulation* step.

Applying this scheme to a 0/1-LP leads to the *Fortet inequalities*.

$$\begin{array}{l} y_{ij} \geq x_i + x_j - 1 \\ y_{ij} \geq 0 \\ y_{ij} \leq x_i \\ y_{ij} \leq x_j \end{array}$$

Indeed, it suffices to consider that $l_x = l_y = 0$, $u_x = u_y = 1$. Furthermore, in this case, we have $y_{ii} = x_i$.

This relaxation is the basis for the Lift & Project Sherali-Adams hierarchies of relaxation of 0/1-LP (see Paragraph 3.4.3) [241].

3.5.4 Quadratic programming

The particular case of Quadratic Programming deals with the following problems :

$$(P)\begin{cases} \min_{x\in\mathbb{R}^n} & x^T P_0 x + 2p_0^T x + \pi_0 \\ \text{subject to} & a_j^T x \leq b_j,\ j = 1, ..., m \end{cases} \tag{3.7}$$

Having P_0 psd places us in the afore mentionned convex case. Otherwise, the problem may harbor several local minima and is NP-hard. Existing algorithms for such a problem are based on Branch & Bound by dividing the feasible set into several subregions and to compute lower bounds by means of linear or semidefinite relaxations. A possibility for branching strategy is to exploit the first-order KKT conditions. See [65, 243] for seminal works on this topic.

3.5.5 Algorithms

Generally, methods for solving a QCQP are derived from nonlinear programming (see Paragraph 3.6.2). The specificity is that two convex relaxations are available, based on the Reformulation-Linearization Technique (see Paragraph 3.5.3) and on SDP (see Paragraph 3.3.2). A comparison of these relaxations can be found in [14] that shows that combining those approaches leads to an enhancement of their respective bounds.

Branch & Bound based approaches based on these relaxations lead to several implementation depending on the subdivision of the space (rectangular, triangular, simplicial), as well summarized in [183].

Another possibility for relaxing a QCQP into a convex problem was proposed by Kim and Kojima in [156]. This relaxation produces a Second-Order Cone Program and can be considered as a compromise between the semidefinite and the linear relaxation. The BARON solver [232] ha been designed and implemented on the basis of these algorithms.

It is worth noticing that the hypothesis of a compact feasible set is very common and useful for this kind of problem. In particular, if allows to reduce Mixed-Integer QCQP into QCQP, by means of a base two reformulation : $y \in \mathbb{Z}, l_y \leq y \leq u_y \Leftrightarrow y = \sum_{i=0}^{\lfloor \log_2(u_y - l_y + 1)\rfloor} 2^i x_i$ with $x_i \in \{0, 1\}$.

Nevertheless, there exists some dedicated algorithms for MIQCQP. In particular, a convexification of the continuous relaxation of the problem [179] allows to use this relaxation within a Branch and Bound procedure. A comprehensive overview for this field can be found in [64].

3.6 (Mixed-Integer) Nonlinear Programming

This section, without any claim of being exhaustive, is supplied as a resource for the reader to develop a better understanding of Nonlinear Programming (NLP) and Mixed-Integer Nonlinear Programming (MINLP). The class of problem to be considered here is :

$$\begin{cases} \min & f_0(x, y) \\ \text{s.t.} & x \in \mathcal{F} \\ & x \in \mathbb{R}^l,\ y \in \mathbb{Z}^n \end{cases} \quad \text{with } \mathcal{F} = \{(x, y) : f_i(x, y) \leq 0,\ i = 1, ..., m\} \tag{3.8}$$

A natural approach consists of solving the *continuous relaxation* of the problem, obtained by dropping the integer constraint, and round off the minimizer to the nearest integer. In the case where this is not appropriate, alternate methods must be investigated.

To address this paradigm, we will first focus on two subcases. Firstly, we consider the case where no variable is required to be integer ($n = 0$) and present the methods for addressing local optimization

of such a nonconvex problem. Then we give a cursory review of the prominent methods for solving a *global optimization problem*, when the sole local optimization is insufficient. Secondly, we explore the case where the continuous relaxation of the problem is convex. Finally, these three parts are used as tools in the fourth paragraph to treat the problem (3.8) in its full generality.

Note that the particular case of the continuous ($n = 0$) convex optimization is treated in detail in the main part of this thesis (see 1.2).

We refer the reader to the classical book on global optimization [141] and to the excellent survey [182], from which this section is largely derived.

3.6.1 Local optimization

Determining a local optimum of a nonconvex optimization problem is NP-hard. The methods for local optimization of nonconvex problems are mainly based on the general optimization algorithms described at section 3.2

3.6.1.1 Sequential Quadratic Programming

When the functions involves in the problem are twice continuously differentiable, a possibility comes from the extension of Newton's method to constrained problem. This iterative method, called *Sequential Quadratic Programming*, solve a sequence of optimization subproblems, each of which optimizes a quadratic approximation of the objective subject to linearization of the constraints. In order to maintain the validity of the approximations, the optimization is limited to a so-called *trust region*, typically a convex set defined as a box around the current point $\{x : -1 \leq e_i^T x \leq 1\}$. This leads to the addition of two linear constraints.

As a consequence, a quadratic optimization problem withonly constraints is known as the *trust region problem*.

Thus, this approach comes to solve a possibly nonconvex quadratic problem at each step. It yields very good results for medium size problems and have been implemented in many NLP packages, including NPSOL, NLPQL, OPSYC, OPTIMA, MATLAB, and SQP.

3.6.2 Global optimization

In full generality, solving such a Non Linear Programm is NP-hard and this problem constitutes one of the most challenging area of optimization. In a very general setting, the methods are typically based on two separate phases, exploiting the "divide-and-conquer principle. This paragraph aims at describing in more details some implementation of this basic principle.

First, the *global phase* consists of an exploration of the exhaustive search space, while the *local phase* determines a locally optimal point, relying on a convex relaxation of a subproblem of the original problem, as described in the previous paragraph 3.6.1. Regarding the global phase, the challenge consists of avoiding the multiple computation of the same local optimum. Some algorithms resort to uncertain parameters for this phase and two main approaches emerge in this framework : the sampling and the escaping approach. In the sampling, the starting points are determined a priori, whereas in the escaping approach, the starting points are determined recursively by exploiting the previous local search. The most expedient of the meta-heuristics relying on these principles are *simulated annealing*, *tabu search* and *variable neighbourhood*.

In the case of deterministic algorithm, the first phase is referred to as the *Branch & Select* method. This method, covered in the first part of this section, is the most widely used approach for global optimization since it does not rely on particular structure of the problem. In fact, they can be used even without analytic description of the objective and constraints function. In this case, the values are provided by black-box procedures.

Finally, an alternate approach called Branch & Infer, consists of using constraint propagation techniques in order to tighten bounds on variables.

3.6.2.1 Branch & Select

Branch & Select include well-known method inspired from MILP such as Branch & Bound, Branch & Cut and Branch & Reduce. Very roughly, it operates as the following steps :

1. Produce a partition P of $\mathcal{F}$;
2. Solve local optimization problem for each $M \in P$. Denote x_M the optimizer ;
3. $x^* = \min\{x^*\} \cup \{x_M : M \in P\}$;
4. Remove from $\mathcal{F}$ the elements of P that that can be shown not to contain the global solution;
5. Refine the partition of $\mathcal{F}$.

The 4th step is crucial and rely generally on the knowledge of an upper bound γ of the optimal solution and of a lower bound of the optimal solution over a restricted region M of $\mathcal{F}$. Thus, if $\min f(x) : x \in M \geq \gamma$ then the global optimizer is known not to belong to M. Consequently, the efficiency of this process, known as *fathoming*, is intimately tied to the quality, or *tightness* of the upper and lower bounds.

In the more specific case of the *spatial Branch & Bound* algorithm, the lower bound is obtained by solving a convex relaxation of the problem. This relaxation is obtained in two stages. First, the nonlinear term are replaced by an additional variable and the corresponding equality constraint is added. In the second stage, the nonlinear terms are replaced by the corresponding convex under and overestimators.

In another variant of Branch & Select, the so-called α Branch & Bound, the functions are assumed to be twice differentiable and the convex underestimators can therefore be constructed automatically.

To complete this section, we mention two other variant of the Branch & Select algorithm. First is the Branch & Reduce, where special attention is paid on reducing the range of the variables. Finally, by similarity to MILP are the Branch & Cut methods, that aims at tightening the convex relaxation by adding valid cuts, in order to get a better lower bounds of the local optima.

3.6.3 Convexification

When the problem is non-convex, a possible approach consists of approximation the problem by a convex one. To this end, each non convex function f_i is replaced by a *convex under-estimate* g_i such that $g_i(x) \leq f_i(x)$, $\forall x$. Therefore replacing f_i by g_i for all non convex f_i leads to a convex approximation of the problem. This is equivalent to add a new variable z_i and approximate $z_i = f_i(x)$ by the inequality $g_i(x) \leq z_i$. The advantage with this approach is that it can benefit from other approximation, for example the one using h_i, a concave over-estimate of f_i : $z_i \leq h_i(x)$.

In some particular case, such as the quadratic ones (see Paragraph 3.5.3), one can characterise the so-called convex lower envelope and concave upper envelopes, which are the tightest possible convex under-estimator and concave over-estimator.

In full generality, we resort to less tight estimators. A famous convex under-estimator is the α-estimator. It applies to twice-differentiable function over a rectangular region and is parameterized by a non-negative vector α. For example, on the region $\mathcal{R} = \{x \in \mathbb{R}^n : 0 \leq x \leq e\}$, it takes the form $f_\alpha(x) = f(x) + x^T \mathrm{Diag}(\alpha) x - \alpha^T x$. This function is necessarily convex for sufficiently large values of α.

3.6.4 Mixed-Integer Convex Programming

The definition of MINLP generally includes the assumption that the continuous relaxation of the problem is convex, which places us in the field of Mixed-Integer Convex Programming (MICP). Approaches for solving these problems are mainly based on Branch & Bound with potentially addition of cutting planes, and Outer Approximation for which there exists a guarantee of convergence to global optimal solution.

3.6.4.1 Branch & Bound

The idea of extending Branch & Bound to MICP can be attributed to Dakin in 1965. This can be done in a very natural manner by solving the continuous relaxation of the problem at each node. The problem were successively studied in [122] regarding to the branching choices. Other works on this topic are [54, 180, 218]. Recently, all the results related to this approach were summarized in [52].

In conclusion, Branch & Bound for MICP is outperformed by approaches based on outer approximation. But there can be useful on instances where OA based methods fails. Furthermore, they can be improved by combination with a cutting planes approaches. See for instance [51, 247, 258] that rely on disjunctive programming and on Lift & Project. The challenge is that the problem defined to compute these cuts is very complex and often more difficult than the continuous relaxation of the problem. The special case of Mixed Integer Conic Programming was the subject of dedicated development, such as [16, 147].

3.6.4.2 Outer Approximation Algorithm

In 1986 Duran and Grossman [88] proposed an algorithm for a particular class of MICP where the involved functions are linear w.r.t. the integer variables. This algorithm, based on the concept of *Outer Approximation*, can be described in words as follows:

1. Solve the (convex) continuous relaxation of the problem and denote x_0 the optimizer;
2. Determine a tangent of $\mathcal{F}$ at x_0;
3. Add this tangent to a set of linear constraints. The obtained MILP is referred to as the *master problem*;
4. Solve the master problem;
5. Fix the integer variable to the master problem integer solution and solve the continuous relaxation;
6. Go to step 2. To prevent cycling, add constraints to cut off the previously found integer solution;
7. Stop when the master problem becomes infeasible or when one termination criteria is satisfied.

Thus, an *outer approximation* of $\mathcal{F}$, i.e. an inclusion $\mathcal{F}$ into a linear set, is built and improved iteratively and the corresponding MILP is solved. More precisely, Outer Approximation refers to the linear approximation of a convex set defined through the tangent hyperplane at boundary points. The convexity of the set ensures that the original set lies inside the outer approximation, as suggested by its name. The more tangent hyperplane are accumulated, the more precise is the approximation. See for instance [92] for a extension of this algorithm to consider nonlinearities w.r.t. the integer variables. This algorithm has proved to be very successful in practice and is implemented for instance in the codes AlphaECP, DICOPT or FilMINT.

3.6.5 Mixed-Integer NonLinear Programming

Algorithms for solving general MINLP are mostly based on extension of approaches developed for MICP, i.e. Branch & Bound and Outer Approximation. These extensions relies on the construction of a convex relaxation of the original problem, for instance by using the convex lower envelope of the function involved in the problem [93, 250].

For the special case of MIQCQP, the convexification approach is based on SDP and is addressed further in Paragraph 3.3.4.

This allowed to develop general-purpose MINLP solvers. In particular, the most commonly used off-the-shelf general solvers, i.e., BARON and Couenne, implement a spatial Branch & Bound algorithm based on a separable reformulation of the problem which enables a convexification of univariate functions.

Application specific approaches, based on piecewise linear approximations of nonlinearities, are also frequently employed. For an exhaustive overview on this topic, we refer the reader to the excellent and very recent survey [62].

3.7 Optimization under uncertainty

In this section, we provide an introduction to optimization under uncertainty. This is by no means exhaustive but aim at helping the reader to acquaint himself with the tools used in the main part of this thesis. The main sources of this section are [217, 228, 233].

The whole optimization process (modelling and resolution) is based on the assumption that suitable data are well-defined and available at decision time. This is generally not the case and then decisions must be taken in the face of uncertainty.

Broadly speaking, an optimization problem with uncertain data can be written as :

$$\begin{array}{ll} \min_{x\in\mathbb{R}^n} & f_0(x) \\ \text{s.t.} & f_i(x,\xi) \leq 0,\ i=1,...,l \end{array} \tag{3.9}$$

where ξ is a m-dimensional random vector on the probability space $\{\Omega, \Sigma, \mathrm{P})$ and $f_i, i = 1, ..., l$ some functions from $\mathbb{R}^n \times \mathbb{R}^m$ to $\mathbb{R}$. Without loss of generality, we assume that the objective function is deterministic : $f_0 : \mathbb{R}^n \to \mathbb{R}$.

As such, this problem does not make any sense. Indeed f_0 can be viewed as a set of functions, parameterized by the value of ξ and therefore $\min_x f_0(x, \xi)$ is not well-defined.

As a consequence, this problem has to be reduced to a deterministic one to be solvable. For this, one defines some mapping $\mathcal{L}_i$ that associates to the random variable $f_i(x, \xi)$ a deterministic value $\mathcal{L}_i(f_i(x, \xi)) = g_i(x)$. In the sequel, we will refer to such a mapping $\mathcal{L}_i$ as an *indicator*.

Then the problem becomes deterministic :

$$\begin{array}{ll} \min_{x\in\mathbb{R}^n} & g_0(x) \\ \text{s.t.} & g_i(x) \leq 0 \end{array}$$

Choosing adequate indicators $\mathcal{L}_i$ is crucial for both the tractability and the meaning of the optimization problem.

However, not all the indicators are possible, depending on the information available on the probability distribution of the random vector ξ. Indeed, a crucial distinction must be made between the case when the probability distribution of ξ is perfectly known, which brings us in the framework of *Stochastic Programming (SP)* and the case when only partial information is available.

Within Stochastic Programming, by considering a real random vector ξ, an indicator $\mathcal{L}$ can be any mapping that associates a deterministic value to ξ, for instance :

– the expected value operator : $\mathcal{L}(\xi) = \mathrm{E}(\xi)$;

– a probability guarantee : $\mathcal{L}(\xi) = \min\{u \in \mathbb{R} : \mathrm{P}[\xi \leq u] \geq 1 - \varepsilon\}$;

– a worst-case value : $\mathcal{L}(\xi) = \min\{u : \mathrm{P}[\xi \leq u] = 1\}$;

– any risk measure as defined at Subsection 2.6.5.

Computing these indicators requires the knowledge of the probability distribution μ. When only a partial information over μ is available, i.e., $\mu \in \mathcal{P}$ where $\mathcal{P}$ is a family of possible distributions, then a possibility is to optimize the worst case of the indicator $\mathcal{L}$ over $\mathcal{P}$:

– the worst-case expected value operator : $\mathcal{L}(\xi) = \max_{\mu \in \mathcal{P}}\{\mathrm{E}_\mu(\xi)\}$;

– a worst-case probability guarantee : $\mathcal{L}(\xi) = \max_{\mu \in \mathcal{P}} \min\{u \in \mathbb{R} : \mathrm{P}_\mu[\xi \leq u] \geq 1 - \varepsilon\}$;

– a worst-case value : $\mathcal{L}(\xi) = \max_{\mu \in \mathcal{P}} \min\{u : \mathrm{P}_\mu[\xi \leq u] = 1\}$;

– the worst-case of any risk measures as defined at Subsection 2.6.5.

The most widespread application of this principle is the *robust optimization* where $\mathcal{P}$ is the set of all the random vectors with a given support $\mathcal{S}$: $\mathcal{P} = \{\mu : \mathrm{P}_\mu[\xi \in \mathcal{S}] = 1\}$. Furthemore, the indicators are the worst-case value : $\mathcal{L}(\xi) = \max\{u : u \in \mathcal{S}\}$.

A more general framework that has attracted the focus of recent research is the *distributionally robust optimization*, where $\mathcal{P}$ is defined via the support and the moments of order less than a given integer k.

3.7.1 Stochastic programming

3.7.1.1 Optimization with recourse

In many problems with uncertainty, the uncertainty will be resolved at some known time in the future. In this case, a key modeling concept lies in the ability to take into account the fact that some decisions do not have to be taken "here and now", but can be made on a 'wait and see' basis, after the uncertainty is resolved. This leads to the classical approach of stochastic programming : the *two-stages optimization*. The decision variables are partitioned in two subsets, the first containing the decision that have to be made before the actual realization of the uncertainty (*static variables*), the other one (*dynamic variables*) are the decisions that can be adjusted after the veil of uncertainty.

Another terminology in the literature for dynamic variables is *recourse variables* and *optimization with recourse* for the associated optimization subfield. This terminology suggests that the dynamic variables are used to fine-tune the decisions made in the first stage, based on the specific outcome of the uncertain parameters.

By similarity with control theory, this kind of optimization is called *closed loop*. This means that the dynamic variables are function of the realization of the random variable ξ. IN this case, $y(\xi)$ is referred to as a *decision rule, strategy, or policy*, i.e, a rule for determining the value of y under all possible circumstances. For instance, in the discrete case, y can be a table of values.

Such a problem can be written under the following form so as to emphasize that y is a function of ξ and of the static variables x :

$$\left\{\begin{array}{lll} \min & \mathrm{E}[f(x, y(x,\xi), \xi)] & \\ \text{s.t.} & \left\{\begin{array}{lll} y(x,\xi) = & \mathrm{argmin}_y & g(x,y,\xi) \\ & \text{s.t.} & (x,y) \in Y(\xi) \end{array}\right. & \\ & x \in X & \end{array}\right.$$

Generally, f has an additive structure : $f(x,y,\xi) = h(x)+g(x,y,\xi)$ and the problem can therefore be formulated as :

$$\begin{cases} \min & h(x) + \mathrm{E}[l(x,\xi)] \\ \text{s.t.} & \begin{cases} l(x,\xi) = & \min_y \quad g(x,y,\xi) \\ & \text{s.t.} \quad (x,y) \in Y(\xi) \end{cases} \\ & x \in X \end{cases}$$

This is in contrast with the *open loop* optimization where all the variables are assumed to be static and the problem is optimized by considering indicator over the distribution of ξ, such as its expected value.

In the case where the uncertain parameters has a finite number N of realizations, we can always form the full deterministic equivalent linear program by introducing one variable y by realization of ξ.

$$\begin{cases} \min & h(x) + \sum_{k=1}^{N} g(x,y_k,\xi_k) \\ \text{s.t.} & (x,y_k) \in Y(\xi_k),\ k = 1,...,N \\ & x \in X \end{cases}$$

With a large number of realizations, this problems becomes quite large. Nevertheless, in the case where it is linear, an implementation of the Benders decomposition known as *L-shaped method* was designed to solve this problem.

The two-stage optimization can be readily extended to multi-stage optimization by modeling the uncertainty as a random process. In this case, the static variables corresponds to the decision that have to be taken "a priori" and some dynamic variables are used at each stage. This draws the connection with the field of *Dynamic stochastic programming*. A system is said *dynamic* when it changes over time. This evolution may be affected by decisions but also by uncertain parameters. Dynamic optimization is concerned with optimization of such systems over time.

It is generally assumed that the uncertainty is stochastic and in particular, that the system forms a Markov chain. Then, the problem can be modeled as finding the best path in the corresponding graph, which enable to apply the well-known Principle of Optimality of Bellman [27].

3.7.1.2 Discrete probability distribution (scenarios) : the Monte-Carlo approximation

The stochastic programming framework relies on the assumption that the uncertain parameters are random variable whose probability distribution are known. In practice, such distribution can be very difficult to estimate and a very common technique to overcome this difficulty is to approximate them by means of a sampling of independent realizations.

Such a sampling can be obtained either as a sample drawn from the distribution or from historical data. In the latter case, this means assuming that the sequence of past demands represents a sample drawn from the same distribution that governs the future demands and is more often than not "an act of faith rather than a solid inference from the experimental data" ([228]).

In the case where the obtained sample is *representative*, i.e., drawn from the relevant distribution, this approximation is justified on the following theoretical ground of the law of large numbers, that states that the average of the samples is an approximation of the expected value of the random variable.

At the end, the probability distribution is assumed to be discrete. This makes easier to incorporate them into the problem, as explained for the case of the multi-stages optimization or chance-constraints. This can also be used within simulation based approaches, allowing for instance to estimate the gradient of the function $\mathrm{E}(f_i(x,\xi))$ and to use it within a descent method.

It is worth noticing that in the case where the optimization is performed separately for each scenario, there is no theoretical guidance about the compromise between the obtained solutions that should actually be adopted. Indeed, these solutions may be inconsistent with each other and very risk-sensitive.

3.7.2 Chance-constraints

We are interested in the case when the indicator $\mathcal{L}$ is a given level of probability $1-\varepsilon$, in a SP perspective, i.e. when the probability distribution P is known. Then a constraint $f(x,\xi) \leq 0$ becomes a *chance-constraint* or *probabilistic constraints*, a notion introduced for the first time in [71] :

$$\mathrm{P}[f(x,\xi) \leq 0] \geq 1-\varepsilon \tag{3.10}$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^l$ is the function whose component the functions f_i of the problem (3.9) and P is the probability distribution associated to the random variable $f(x,\xi)$.

This leads to a very intuitive and meaningful deterministic version of the problem (3.9) since it means that we authorise a certain probability of violation of the feasibility. Generally, this implies that $l > 1$ and the chance-constraint is then a *joint* constraint. We could also think of defining one chance-constraint for each constraint of the problem. Then, the constraints are said to be *individual* and are much easier to tackle. It makes less sense from a modeling point of view, but we will see that it is possible to approximate a joint constraint into a bunch of individual constraints.

Such constraints offer the benefit of ignoring the severe consequences of rare events, which makes them less conservative than the worst-case approach. Their probabilistic guarantee may be satisfactory for the decision-makers whenever the situation repeats itself many times. Such a guarantee becomes much more problematic when applied to a unique action.

However, some difficulties are associated with this modelization. Firstly, as already mentioned, such a constraint makes sense only if the uncertain data are of stochastic nature and if their underlying distribution is known. Secondly, determining which value of ε to use is far from obvious. Last but not least, in most cases, chance-constraint are computationally intractable. Even in a simple individual case, it may be difficult to evaluate the probability. A famous example can be found in [201] where Shapiro and Nemirovski point out that computing the left-hand side of 3.10 with f linear w.r.t. x and some uniformly distributed random variables ξ is already NP-hard. Another major difficulty stems from the fact that generally the feasible set of a chance-constraint is not convex.

A possibility to overcome these difficulty is to resort to the Monte-Carlo simulation and replace the chance-constraint by the requirement that the inner constraint must hold on at least $N(1-\varepsilon)$ sample points, if N is the sample size. The major advantage of this approach is that no structural assumption about the distribution is required. On the other hand, this approximation must be made over $O(n/\varepsilon)$ samples to guarantee the feasibility of the solution, which becomes too computationally costly when ε is small [67].

Generally, the way to overcome these difficulties is to resort to approximations. In order to stay "on the safe side", we must restrict ourselves to conservative or *safe* approximations, i.e. approximations that guarantees the satisfaction of the original constraint. Obviously, this generally results in a loss of optimality.

However, there are some particular cases that are exactly tractable. We start by presenting them before tackling harder constraints and their related approximations.

For an extensive discussion on chance-constraints, we refer the reader to the standard references [133, 201, 216, 217, 231].

3.7.2.1 Individual linear constraint with Gaussian distribution

The simplest possible case of chance-constraint is obtained by taking $m = 1$, g an affine function w.r.t x : $g(\omega, x) = \xi_0(\omega) + \sum_{i=1}^{n} \xi_i(\omega) x_i$ and ξ is a Gaussian $(n+1)$-dimensional random vector : $\xi \equiv \mathcal{N}(\mu, M)$. Then, for $\varepsilon < 0.5$, the corresponding probabilistic constraint can be exactly formulated as a second-order conic constraint (SOC), as firstly established in [256].

Indeed, for a fixed x and its homogenisation $\tilde{x} = \begin{pmatrix} 1 & x^T \end{pmatrix}^T$, $g(\omega, x) = \xi^T \tilde{x}$, is also a Gaussian random variable :

$$\xi^T \tilde{x} \equiv \mathcal{N}(\mu^T \tilde{x}, \tilde{x}^T M \tilde{x})$$

Consequently,

$$\mathrm{P}[\xi^T \tilde{x} \leq 0] = \Phi(\frac{\mu^T \tilde{x}}{\sqrt{\tilde{x}^T M \tilde{x}}})$$

and the constraint becomes

$$\begin{aligned} \mathrm{P}[\xi^T \tilde{x} \leq 0] \geq 1 - \varepsilon \quad &\Leftrightarrow \Phi(\tfrac{\mu^T \tilde{x}}{\sqrt{\tilde{x}^T M \tilde{x}}}) \geq 1 - \varepsilon \\ &\Leftrightarrow \tfrac{\mu^T \tilde{x}}{\sqrt{\tilde{x}^T M \tilde{x}}} \geq \Phi^{-1}(1 - \varepsilon) \\ &\Leftrightarrow \mu^T \tilde{x} \geq \Phi^{-1}(1 - \varepsilon) \|M^{1/2} \tilde{x}\| \end{aligned}$$

As a consequence, with $\varepsilon < 0.5$, then $\Phi^{-1}(1 - \varepsilon) \geq 0$ and the constraint is a typical second-order constraint.

3.7.2.2 Convexity of the feasible set

Let denote $\mathcal{F}$ the feasible set associate with the chance-constraint (3.10) :

$$\mathcal{F} = \{x \in \mathbb{R}^n : \mathrm{P}[f(x, \xi) \leq 0] \geq 1 - \varepsilon\}$$

The convexity of $\mathcal{F}$ is established in the following case :

- if $f(x, \xi) = Ax - \xi$ with A a deterministic matrix and ξ a log-concave random variable [216];
- more generally, the set defined by the constraint $\mathrm{P}[(x, \xi) \in X] \geq 1 - \varepsilon$ is convex whenever X is a deterministic convex set and ξ is log-concave ;
- if $m = 1$ and $f(x, \xi) = \xi^T \tilde{x}$, with ξ symmetric log-concave and $\varepsilon < 1/2$.
- if ξ is log-concave and if the components of f are quasi-concave as functions of x and ξ simultaneously.

Unfortunately, having $\mathcal{F}$ is not sufficient for solving efficiently the problem. The existence of a poly-time separation oracle is also necessary. This implies being able to compute the probability in polynomial time. The only case where both requirements are satisfied is a subcase of the third item, where the random vector is governed by a radial distribution. Indeed, in this case, the chance constraint can be converted into second-order constraints.

3.7.2.3 Computationally tractable safe approximations of individual chance constraints

This paragraph is mainly derived from [201] that provides an original paradigm for dealing with linear individual chance constraints of the form :

$$\mathrm{P}[f(x, \xi) \leq 0] \geq 1 - \varepsilon \text{ with } f(x, \xi) = \tilde{x}^T A \tilde{\xi},\ A \in \mathbb{R}^{n,m} \tag{3.11}$$

As such, this constraint is in full generality neither convex nor tractable and the aim is therefore to find a *computationally tractable safe approximations* , i.e., an approximation which is both

- *safe*, i.e., is sufficient to guarantee the feasibility of the original constraint

– *computationally tractable*, i.e., both convex and efficiently computable.

With a variable substitution $w = A\tilde{x}$, the constraint becomes $p(w) = \mathrm{P}[w^T\tilde{\xi} > 0] \leq \varepsilon$. An explicit formulation for p is :

$$p(w) = \int_{\mathbb{R}^m} \mathbb{1}_{\mathbb{R}_{++}}(w^T\tilde{\xi})\mathrm{P}(\xi)d\xi$$

The key trick consists of replacing $\mathbb{1}_{\mathbb{R}_{++}}$ by a convex overestimator γ such that $\gamma(z) \geq \mathbb{1}_{\mathbb{R}_{++}}(z)$, $\forall z \in \mathbb{R}$. Indeed $q(w) = \int \gamma(w^T\tilde{\xi})\mathrm{P}(\xi)d\xi$ is convex whenever γ is convex, since $w^T x$ is affine. Furthermore, the safeness of the approximation is guaranteed by $p(w) \leq q(w)$. Consequently, $q(w) \leq \varepsilon$ is a convex safe approximation of the constraint (3.11).

Exploiting the invariance of $\mathbb{1}_{\mathbb{R}_{++}}$ by positive scaling : $\mathbb{1}_{\mathbb{R}_{++}}(z) = \mathbb{1}_{\mathbb{R}_{++}}(\frac{z}{\alpha},\ \forall \alpha > 0$, leads to the following convex overestimator : $\gamma(\frac{z}{\alpha}) \geq \mathbb{1}_{\mathbb{R}_{++}}(z)$, $\forall z \in \mathbb{R}$ and a safe convex approximation is therefore :

$$G(w) = \inf_{\alpha>0}\{\alpha q(\frac{w}{\alpha}) - \alpha\varepsilon\} \leq 0$$

The safeness comes from the lower semicontinuity of q and the convexity is ensured by Proposition 2.4.39.

Among the possible function γ, we restrict our attention to the ones that satisfy the following properties, by similarity with $\mathbb{1}_{\mathbb{R}_{++}}$:

– γ is a nonnegative monotone function;

– $\gamma(0) \geq 1$;

– $\gamma(z) \to 0$ as $z \to -\infty$.

Such functions are referred to as *generators*. A typical example is the function $\gamma(z) = \exp(z)$.

For a given generator γ, some other safe approximations can be established, by replacing q by any convex overestimator q^+ of q, i.e., $q^+(w) \geq q(w)$, $\forall w$. This leads to the following safe convex approximation :

$$G^+(w) = \inf_{\alpha>0}\{\alpha q^+(\frac{w}{\alpha}) - \alpha\varepsilon\} \leq 0 \tag{3.12}$$

Choosing q^+ efficiently computable makes this approximation tractable. Furthermore, this additional level of approximation enables to consider the case where the chance constraint has to be satisfied for a set of probability distribution $\mathrm{P} \in \mathcal{P}$. To see this, we add the notation P to $q(w) : q_{\mathrm{P}}(w)$ to underlie that $q(w)$ depends of P.

Then, having q^+such that : $q^+(w) \geq q_{\mathrm{P}}(w)$, $\forall \mathrm{P} \in \mathcal{P}$ is sufficient for the final approximation to hold. Thus, we reduce a set of constraint : one for each element of $\mathcal{P}$ into a single constraint.

As an illustration, we show that this approach enables to recover the Azuma-Hoeffding inequality (2.6.39). We are interested in $p(w) = \mathrm{P}[w^T\xi > 0] \leq \varepsilon$, $\forall \mathrm{P} \in \mathcal{P}$, with $\mathcal{P}$ the set of zero mean probability distribution supported on $[-1, 1]^m$ such that the random variables are independent from each other.

Then, by taking $\gamma(z) = \exp(z)$ and $q^+(w) = \exp\{\frac{1}{2}\sum_{i=1}^m w_i^2\}$, we get the approximation at hand. Indeed,

$$\begin{array}{lll} \mathrm{P}[\mathrm{P}[w^T\xi > 0] & \leq \mathrm{E}[\exp w^T\xi] & \text{(introduction of the generator)} \\ & = \prod_{i=1}^m \mathrm{E}[\exp w_i\xi_i] & \text{(independence)} \\ & \leq \prod_{i=1}^m \exp(\frac{w_i}{2}) & \\ & = q^+(w) & \end{array}$$

The last inequality is detailed in [201] Lemma 2.1 and relies on the assumption regarding the support and the mean of the elements of $\mathcal{P}$.

Among this scheme of approximation, we aim at finding the best one, i.e., the one that minimize the deviation from the original constraint. Clearly, for any generator γ, the best q^+ is $\sup_{\mathrm{P}\in\mathcal{P}} \mathrm{E}_{\mathrm{P}}\left(\gamma(w^T\tilde{\xi})\right)$.

Moreover, it is proved in [201] that the best generator is $\gamma^*(z) = \max\{0, 1+z\}$. Then, a fundamental result is that in this case, the corresponding approximation (3.12) is the CVaR approximation :

$$G^+(w) = \inf_{\beta}\left\{\beta + \frac{1}{\varepsilon}\mathrm{E}((w^T\tilde{\xi})^+)\right\} = \mathrm{CVaR}_{\varepsilon}(w^T\tilde{\xi})$$

As a consequence, the CVaR approximation is the least conservative convex approximation of a chance constraint.

3.7.2.4 Approximation of joint chance constraint by individual chance constraints

There are basically two approaches to approximate a joint chance constraint into individual chance constraints. The first one is a conservative approximation based on the Boole inequality 2.6.34. The second one convert the joint chance constraint into an equivalent individual chance-constraint. But, in line with the "no free lunch" principle, the obtained chance-constraint is much harder to handle.

We consider the following joint chance-constraint : $\mathrm{P}[f_i(x,\xi) \leq 0,\ i = 1,...,m] \geq 1-\varepsilon$ where $f_i : \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}$.

By applying Boole inequality, it comes that for any sequence of $\{\varepsilon_i\}_{i=1,...,m}$ such that $\sum_{i=1}^m \varepsilon_i \leq \varepsilon$, requiring that $\mathrm{P}[f_i(x,\xi) \leq 0] \geq 1-\varepsilon_i,\ i = 1,...,m$ is sufficient for the satisfaction of the joint chance constraint. In particular, it suffices to divide ε equally among the constraint : $\varepsilon_i = \varepsilon/m$. This approximation is simple but generally not tight, in particular when the individual events $f_i(x,\xi)$ are not independent.

Another possibility, introduced recently in [72], comes from the following trick : $f_i(x,\xi) \leq 0,\ i = 1,...,m \ \Leftrightarrow g(x,\xi) = \max_i f_i(x,\xi) \leq 0$. The same occurs by scaling the functions f_i via a vector $\alpha > 0$: $f_i(x,\xi) \leq 0,\ i = 1,...,m \ \Leftrightarrow g_\alpha(x,\xi) = \max_i \alpha_i f_i(x,\xi) \leq 0$. Consequently, the joint chance-constraint can be converted into the following individual chance constraint :

$$\mathrm{P}[g_\alpha(x,\xi) \leq 0,\ i = 1,...,m] \geq 1-\varepsilon$$

3.7.3 Robust optimization

Robust optimization consists of optimizing what may happen in the *worst case* w.r.t a given set of uncertain data. It traces back to the early 70s with the work of Soyster on robust linear optimization but the interest for robust optimization really started with the work of Ben-Tal and Nemirovski [32] and El Ghaoui et al. [89] in the late 90s. We refer the reader to the survey [36] for a complete review on the subject.

A robust problem can be formulated as following :

$$(P)\begin{cases} \min_{x\in\mathcal{K}} & f_0(x) \\ \text{s.t.} & f_i(x,\xi) \leq 0, \forall \xi \in \mathcal{U},\ i = 1,...,m \end{cases}$$

where $\mathcal{U}$ is the (closed) uncertainty set. If $\mathcal{U}$ has an infinite number of elements, then the problem has an infinite number of constraints and is therefore a so-called *semi-infinite optimization program*.

This approach has two major advantages w.r.t stochastic optimization. Firstly, it is not necessary that the probability distribution be available, only the support, then referred to as the *uncertainty set* has to be specified. Secondly -and very importantly - the solutions generated by this approach are

immune to any realizations of the uncertain parameters in the uncertainty set. Thus, in constrast to stochastic programming, these solutions make sense even in a single-outcome situation.

On the downside, it may be difficult to specify the uncertainty set since it represents a tradeoff between robustness and performance. A large uncertainty set leads to a conservative optimization and may affect severely the optimal value reached by the objective. A possible remedy is to restrict ourselves to a smaller uncertainty set. But there are no free lunch since it weakens the guarantee on the feasibility of the solution.

Generally speaking, due to its infinite number of constraints, a robust problem is computationnally intractable. However, in some particular cases, it can be formulated as a "standard" optimization problem, i.e., with a finite number of constraints. In fact, a sufficient condition for the problem to be tractable is that the feasible set be convex with an efficiently computable separation oracle. When such a formulation exists, it is referred to as the *robust counterpart* of the original problem.

Equivalently, the robust problem (P) reads :

$$(P)\begin{cases} \min_{x\in\mathcal{K}} & f_0(x) \\ \text{s.t.} & \max_{\xi\in\mathcal{U}} f_i(x,\xi) \leq 0,\ i=1,...,m \end{cases}$$

Then the structure of the subproblems $\max_{\xi\in\mathcal{U}} f_i(x,\xi)$, and in particular the shape of the uncertainty set, is determinant for the complexity of solving (P). Specifying $\mathcal{U}$ as an ellipsoid (see Def. 2.2.65) is both interesting from the tractability point of view, as illustrated above on the robust linear programming example, and from a modeling point of view since numerous sets are encompassed within this framework, for example polyhedra.

Let us consider the example of robust linear programming with ellipsoidal uncertainty set, proposed by Ben-Tal and Nemirovksi in [32]. Without loss of generality, it can be written as :

$$(P)\begin{cases} \min_{x\in\mathbb{R}^n} & c^T x \\ \text{s.t.} & A_i^T x \leq 0,\ i=1,...,m, \forall A_i \in \mathcal{U}_i \\ & b_i^T x \leq 0,\quad i=1,...,p \end{cases}$$

where $A_i,\ i=1,...,m$ are uncertain parameters. The uncertainty set is the union of ellipsoidal regions, one for each constraints : $\mathcal{U}_i = \{A_i^0 + B_i u : \|u\| \leq \rho\}$.

Then the robust counterpart is :

$$(P)\begin{cases} \min_{x\in\mathbb{R}^n} & c^T x \\ \text{s.t.} & A_i^0 x \leq -\rho\|B_i x\|,\ i=1,...,m \\ & b_i^T x \leq 0,\quad i=1,...,p \end{cases}$$

Indeed, $\max_{A_i\in\mathcal{U}_i} A_i^T x = \max_{\|u\|\leq\rho}(A_i^0 + B_i u)^T x = A_i^{0^T} x + \rho\|B_i^T x\|$.

It was also shown in [32] that if the uncertainty sets $\mathcal{U}_i$ are polyhedral, then the robust counterpart is a linear program and that the robust counterpart of a SOCP with ellipsoidal uncertainty sets is a semidefinite program, as detailed at paragraph 3.5.1. Another major contribution on this topic was provided by El-Ghaoui and Lebret [89] that showed that the robust least square problem admits a SOCP robust counterpart when the uncertainty set is ellipsoidal.

As a conclusion are given some elements about how to build the uncertainty set and how it allows to choose the corresponding level of probabilistic protection. As a first key, if it can be asserted that the probability for the uncertain parameters not to belong to the uncertainty set is less than ε, then the robust solution is guaranteed to satisfy the $1-\varepsilon$ associated chance constraint. Second, when considering $\mathcal{U}$ is too expensive or lead to no feasible solution, a possible remedy is to consider a smaller uncertainty set $\mathcal{N} \subset \mathcal{U}$ and to authorize violations of the constraint for $u \in \mathcal{U} \setminus \mathcal{S}$, in a controlled manner so that larger violations are allowed as the distance of u from $\mathcal{N}$ increases. A distance function was proposed in [42] for linear program, based on the number of parameter by constraints that do not belong to the corresponding subset of $\mathcal{N}$.