



HAL
open science

Méthodes d'optimisation non différentiable pour la résolution de grands problèmes. Applications à la gestion à moyen-terme de la production.

Gregory Emiel

► **To cite this version:**

Gregory Emiel. Méthodes d'optimisation non différentiable pour la résolution de grands problèmes. Applications à la gestion à moyen-terme de la production.. Mathématiques [math]. Université Panthéon-Sorbonne - Paris I, 2008. Français. NNT: . tel-00363868

HAL Id: tel-00363868

<https://theses.hal.science/tel-00363868>

Submitted on 24 Feb 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE

présentée pour l'obtention du titre de

Docteur de l'Université Paris 1, Panthéon-Sorbonne

Spécialité : Mathématiques Appliquées

par **Grégory Emiel**

Sujet : *Méthodes d'optimisation non différentiable
pour la résolution de grands problèmes.
Applications à la gestion à moyen-terme de la production.*

Soutenance le 7 novembre 2008 devant le jury composé de :

Rapporteurs :	Claude Lemaréchal Dominikus Noll	Inria Rhône-Alpes Université Paul Sabatier, Toulouse
Examineurs :	Kengy Barty Jean-Marc Bonnisseau	Électricité De France R&D Université Paris 1
Direction :	Claudia Sagastizábal	CEPEL & IMPA, Rio de Janeiro
Co-direction :	Frédéric Bonnans	INRIA Saclay & CMAP, Ecole Polytechnique

Cette thèse a été effectuée conjointement à l'IMPA de Rio de Janeiro et dans le département OSIRIS d'EDF R&D, en partenariat avec le CERMSEM de l'université de Paris I et l'INRIA.



Résumé

Cette thèse s'intéresse à la résolution de problèmes d'optimisation non-différentiable de grandes tailles résultant le plus souvent d'une relaxation Lagrangienne opérée sur un problème difficile. Cette technique est couramment utilisée pour appréhender des problèmes linéaires avec nombres entiers ou des problèmes convexes complexes. Une application particulière est la résolution d'un problème de gestion de production d'électricité où la contrainte de demande est relâchée. Dans cette approche, un programme maître est responsable de la coordination de sous-problèmes locaux propres à chaque unité de production. Cette coordination consiste en l'optimisation d'une fonction duale séparable et non-différentiable, ce qui peut être réalisé, par exemple, par une méthode de faisceau. Au chapitre 2, nous présentons les outils de base de l'optimisation non-différentiable et quelques résultats récents sur des algorithmes incrémentaux ou utilisant des informations inexactes. Toutefois, dans certaines situations, le problème dual peut être lui-même très difficile à résoudre et nécessiter des stratégies adaptées.

Par exemple, lorsque le nombre de contraintes dualisées est très élevé (exponentiel en la dimension du problème primal), une dualisation explicite peut s'avérer impossible ou la mise à jour des variables duales peut échouer. Afin de réduire la dimension de l'espace dual, différentes heuristiques ont été proposées. Elles introduisent une procédure de séparation qui sélectionne de façon dynamique un sous-ensemble de contraintes devant être dualisées au fil des itérations. Cette approche de type relax-and-cut a démontré son efficacité sur de nombreux problèmes combinatoires. Au chapitre 3, nous démontrons la convergence primale-duale d'une telle stratégie lorsqu'un algorithme de sous-gradient est utilisé pour l'étape duale, avec des hypothèses minimales sur la procédure de séparation.

Une autre limite de la relaxation Lagrangienne peut apparaître lorsque la fonction duale est séparable en un grand nombre de sous-fonctions, ou que celles-ci restent difficiles à évaluer. Dans ce cas, les temps de calcul nécessaires à la résolution de tous les sous-problèmes locaux peuvent être prépondérants sur l'ensemble du processus. Une stratégie naturelle consiste alors à tirer partie de la structure séparable en effectuant des itérations duales en n'ayant évalué qu'un sous-ensemble des sous-fonctions. Ce type d'approche incrémentale a déjà été appliqué avec des algorithmes de sous-gradient et les résultats numériques reportés dans la littérature sont prometteurs. Au chapitre 4, nous employons une version particulière des méthodes de faisceau, réputées pour leurs stabilité et précision. Nous illustrons les performances de l'algorithme sur un problème réel de gestion de production pour le mix énergétique français.

Enfin, au chapitre 5 nous nous sommes intéressés plus spécifiquement à la gestion de production d'électricité à moyen terme. Le parc de production d'EDF est composé de centrales hydrauliques, nucléaires et thermiques à flammes, chacune ayant des contraintes de fonctionnement spécifiques. Sur un horizon de temps de 1 à 3 ans, le problème de gestion de production est un problème d'optimisation stochastique de grande taille qui ne peut être résolu par une approche frontale. Nous présenterons un logiciel développé par la R&D de EDF et utilisant la relaxation Lagrangienne de la contrainte de satisfaction de la demande pour résoudre ce problème. La fonction duale est maximisée à l'aide d'une

méthode de faisceau proximale. Nos travaux ont permis de proposer des améliorations pour le code de faisceau actuellement utilisé, afin de stabiliser et d'accélérer le processus et d'extraire des informations utiles concernant la solution du problème primal initial. Nous avons également suggéré une heuristique permettant la valorisation des réserves d'énergie hydrauliques et nucléaires à partir des commandes optimales obtenues.

Abstract

This manuscript deals with large-scale non-smooth optimization that may typically arise when performing Lagrangian relaxation of difficult problems. This technique is commonly used to tackle mixed-integer linear programming – or large-scale convex – problems. For example, a classical approach when dealing with power generation planning problems in a stochastic environment is to perform a Lagrangian relaxation of the coupling constraints of demand. In this approach, a master problem coordinates local subproblems, specific to each generation unit. The master problem deals with a separable non-smooth dual function which can be maximized with, for example, bundle algorithms. In chapter 2, we introduce basic tools of non-smooth analysis and some recent results regarding incremental or inexact instances of non-smooth algorithms. However, in some situations, the dual problem may still be very hard to solve.

For instance, when the number of dualized constraints is very large (exponential in the dimension of the primal problem), explicit dualization may no longer be possible or the update of dual variables may fail. In order to reduce the dual dimension, different heuristics were proposed. They involve a separation procedure to dynamically select a restricted set of constraints to be dualized along the iterations. This relax-and-cut type approach has shown its numerical efficiency in many combinatorial problems. In chapter 3, we show Primal-dual convergence of such strategy when using an adapted subgradient method for the dual step and under minimal assumptions on the separation procedure.

Another limit of Lagrangian relaxation may appear when the dual function is separable in highly numerous or complex subfunctions. In such situation, the computational burden of solving all local subproblems may be preponderant in the whole iterative process. A natural strategy would be here to take full advantage of the dual separable structure, performing a dual iteration after having evaluated only a subset of the dual local subfunctions, instead of all of them. This type of incremental approach has already been applied for subgradient algorithms and the reported computational results are promising. In chapter 4, we use instead a specialized variant of bundle methods known by their stability and precision. We give numerical illustrations for the EDF's mid-term generation planning problem.

In chapter 5, we focus on this specific problem. The French power mix includes hydraulic, nuclear, and classical thermal power plants, with each generation unit having its own technical constraints. For an horizon of one to three years, the generation planning problem is a large scale stochastic program, untractable with a direct frontal approach. We will present a software using Lagrangian decomposition on a scenario tree to tackle this problem. The dual function is maximized using a proximal bundle method. As a byproduct of our work we proposed some improvements of the algorithms to fasten and stabilize the resolution process and extract useful informations regarding the primal solutions. We also developed a heuristical valuation of water and nuclear ressources.

Remerciements

Mes premiers remerciements s'adressent à Claudia Sagastizábal qui a encadré cette thèse. Son soutien, sa bonne humeur, sa disponibilité et son exigence ont été des éléments indispensables à la réussite de ce travail. Plus encore que pour ses qualités de directrice de thèse, c'est pour son amitié que je la remercie, en espérant que l'on se retrouve bientôt autour d'un churrasco argentin et d'une caïpirinha dont Misha a le secret.

Je tiens ensuite à remercier Frédéric Bonnans et Kengy Barty qui ont co-encadré cette thèse. Les discussions que nous avons eu lors de mes séjours en France ont contribué à améliorer la qualité du travail présenté et à ouvrir des pistes de recherches futures. Je suis également particulièrement reconnaissant à Claude Lemaréchal et Dominikus Noll pour l'intérêt qu'ils ont porté à cette thèse me faisant l'honneur d'en être les rapporteurs, ainsi qu'à Jean Marc Bonnisseau qui a présidé le jury et qui m'a accordé sa confiance en m'acceptant au sein du programme doctoral de l'Université Paris 1.

Les travaux de cette thèse se sont déroulés dans le cadre d'un partenariat international entre EDF R&D en France et l'IMPA au Brésil. Ce projet complexe n'aurait pas pu voir le jour sans la collaboration et la détermination de nombreuses personnes auxquelles je tiens à montrer ici toute ma gratitude : Claudia et Frédéric à nouveau, mais également René Aïd, Karima Boukir, Pascal Martinetto, Luciano Leal de Sousa et Kengy Barty pour EDF R&D, ainsi que Cesar Camacho pour l'IMPA. Cette liste ne saurait être complète sans citer Léonard Bacaud, mon prédécesseur dans la filière franco-brésilienne des mathématiques appliquées et Sophie, indirectement à l'origine de ce projet.

J'ai également une pensée pour tous les membres du département OSIRIS de EDF R&D que j'ai eu le plaisir de côtoyer durant ces trois années de thèse ainsi qu'au cours de mon stage de DEA. Je remercie particulièrement tous ceux avec lesquels j'ai le plus eu l'occasion de travailler et d'échanger : Karima, Pascal, Luciano, Kengy, Marc, Riadh, JS, Cyrille, Anès, Romain, Arnaud L., Pierre, Babakar, Arnaud P., Nazim, Alice, Sinda.

De l'autre côté de l'océan, mes remerciements vont à tous ceux qui ont facilité mon adaptation à cette ville fascinante qu'est Rio de Janeiro : le personnel administratif et les étudiants de l'IMPA, en particulier mes collègues de bureaux, Didier, Heudson et Vanessa. Je remercie également les professeurs et étudiants du groupe d'optimisation et tout spécialement Alfredo Iusem et Mikhail Solodov qui ont contribué à la maturation de mes travaux de recherche lors des échanges que nous avons eu en séminaire.

Mes derniers remerciements vont à mes parents, mes frères et ma soeur pour leur soutien permanent durant ces (très) longues années d'études. Je réserve une pensée particulière à mon père, qui aurait aimé vivre ce moment.

Enfin, je dédis ce travail à ma femme, Isabelle, avec laquelle j'ai partagé mes plus beaux moments de ces dernières années et qui m'a supporté et soutenu dans les périodes de doute.

Table des matières

1	Introduction et résumé général	15
1.1	La gestion de production	15
1.2	Optimisation non-différentiable	16
1.3	Relaxation Lagrangienne Dynamique	16
1.4	Méthodes inexactes et incrémentales	17
1.5	Organisation de la thèse et principales contributions	17
	Bibliographie	19
2	Optimisation non-différentiable : méthodes de sous-gradient, méthodes de faisceau	21
2.1	Introduction	21
2.2	Optimisation non-différentiable : quelques rappels	23
2.2.1	Éléments d'analyse convexe	23
2.2.2	Direction de descente	24
2.2.3	Conditions d'optimalité	25
2.3	Méthodes de sous-gradient	26
2.3.1	Propriétés de base	26
2.3.2	Reconstruction primale	27
2.3.3	Méthodes inexactes	29
2.3.4	Méthodes incrémentales	29
2.4	Méthodes de faisceau	30
2.4.1	Une méthode de plans coupants stabilisée	30
2.4.2	Propriétés de base	31
2.4.3	Solution primale	34
2.4.4	Méthodes inexactes	34
2.4.5	Méthodes incrémentales	37
2.5	Conclusion	37
	Bibliographie	42
3	Dynamic Lagrangian Relaxation	43
3.1	Introduction	43
3.2	Dynamic Lagrangian Relaxation	44
3.3	Dynamic subgradient methods	46
3.3.1	Subgradient iterates in a dynamic setting	46

3.3.2	Choosing index sets	50
3.3.3	Convergence results	51
3.4	Dynamic Bundle Methods	55
3.4.1	Bundle iterates in a dynamic setting	56
3.4.2	Choosing index sets	57
3.4.3	Convergence results	57
3.5	Conclusion	57
	Bibliography	60
4	Inexact and Incremental Approaches	61
4.1	Inexact bundle methods with varying accuracy	62
4.1.1	Inexact linearizations with varying accuracy	62
4.1.2	Convergence analysis	67
4.2	An alternative approach when $E_{x^\ell}^g = 0$ and $E_{x^\ell}^f \rightarrow 0$	70
4.2.1	Inexact classical bundle method	70
4.2.2	Convergence analysis	72
4.3	Incremental bundle methods	74
4.3.1	Motivations	74
4.3.2	Related works	74
4.3.3	Incremental bundle methods	74
4.4	Mid term generation planning	76
4.4.1	Price decomposition on a scenario tree	76
4.4.2	Numerical illustrations	78
4.5	Conclusion	79
	Bibliography	82
5	Gestion à moyen terme de l'électricité	83
5.1	Introduction et résumé	83
5.2	La gestion de production	84
5.2.1	Description du portefeuille	85
5.2.2	Coût marginal et valeur d'usage	87
5.2.3	Différents horizons de temps	87
5.2.4	GMT : un problème stochastique multi-périodes de grande taille	90
5.3	Présentation d'un outil de GMT : SOPRANO	94
5.3.1	Décomposition par les prix sur un arbre d'aléas	95
5.3.2	Description des sous-problèmes	97
5.3.3	Jeux de données utilisés	99
5.4	Quelques faits stylisés	100
5.4.1	Obtention de solutions primales approchées	100
5.4.2	Sélection et compression du faisceau	103
5.5	Calcul de stratégies pour la simulation	104
5.5.1	Valeur d'usage d'un stock d'énergie	104
5.5.2	Valorisation face à un signal de prix	104
5.5.3	Valorisation à production complémentaire fixée	105

5.5.4	Résultats numériques	106
5.5.5	Test des stratégies en simulation	106
5.6	Apports d'une méthode dynamique de faisceaux	108
5.7	Conclusion	110
	Bibliographie	115
6	Conclusions et Perspectives	117
6.1	Contributions de la thèse	117
6.2	Perspectives	118

Table des figures

2.1	Non-différentiabilité	24
2.2	Direction du sous-gradient	24
2.3	Boîte Noire / Oracle	25
2.4	Linéarisation inexacte	35
3.1	Dynamic Lagrangian Relaxation Scheme	45
4.1	Price decomposition	77
4.2	Evaluation of local subproblems is preponderant.	78
4.3	Incremental bundle method	79
5.1	Gestion de production	88
5.2	Arbre de scenarios	93
5.3	Décomposition/Coordination par les prix	96
5.4	Puissance admissible au cours d'une campagne (source:[Mal05])	98
5.5	Recupération primale	102
5.6	Influence de la procédure de nettoyage du faisceau	103
5.7	Valeurs d'usage des réserves hydrauliques	106
5.8	Valeurs d'usage pour deux tranches nucléaires	107
5.9	Première itération de coordination	109
5.10	Relaxation dynamique Vs classique	109

Chapitre 1

Introduction et résumé général

Sommaire

1.1 La gestion de production	15
1.2 Optimisation non-différentiable	16
1.3 Relaxation Lagrangienne Dynamique	16
1.4 Méthodes inexactes et incrémentales	17
1.5 Organisation de la thèse et principales contributions	17
Bibliographie	19

La libéralisation récente du secteur de l'énergie en Europe a modifié sensiblement les problématiques auxquelles doivent faire face les différents producteurs. En particulier, l'ouverture de la concurrence et l'apparition des marchés de l'électricité a considérablement augmenté l'incertitude sur la marge financière. En contrepartie, ces marchés offrent une opportunité pour la couverture des risques physiques (défaillance) et financiers.

Le portefeuille d'actifs d'EDF, déjà conséquent avec 58 tranches nucléaires, une vingtaine de groupes thermiques classiques et de très nombreuses usines hydrauliques, se trouve complexifié par la prise en compte des actifs financiers. La gestion optimale d'un tel portefeuille en environnement fortement incertain est un véritable défi théorique et numérique pour la recherche en mathématiques appliquées.

Dans ce contexte, Electricité de France, comme d'autres acteurs énergétiques européens, a lancé un ensemble de projets de recherche visant à développer les techniques adaptées pour appréhender ce nouvel environnement. Cette thèse s'inscrit dans cette politique : elle traite de l'application des méthodes d'optimisation non-différentiable [HUL96] pour la résolution de problèmes de très grande taille.

1.1 La gestion de production

La gestion de production d'électricité vise à optimiser l'utilisation d'un portefeuille composé d'actifs de production (centrales) et d'actifs financiers (marchés). Les décisions devant être prises sont très variées et vont de l'ajustement infra-journalier aux investissements dans de nouveaux actifs à l'horizon pluri-annuel. Compte tenu de la taille du problème et des différents horizons d'étude, une décomposition en sous-problèmes avec des horizons d'étude spécifiques devient nécessaire.

Au cours de cette thèse, nous nous sommes particulièrement intéressé à la gestion à moyen-terme de la production dont l'horizon d'étude varie de 1 à 3 ans. L'un des objectifs de la gestion moyen-terme est le calcul d'indicateurs permettant de valoriser les différentes

réserves d'énergie (hydrauliques, nucléaires) en fonction des économies futures qu'ils permettront de réaliser. Ces réserves représentent un enjeu stratégique majeur puisqu'elles apportent une flexibilité importante dans la gestion de la production d'électricité, celle-ci étant par nature non-stockable sous sa forme primaire.

L'une des difficultés de la gestion à moyen-terme est le caractère stochastique des paramètres du problème comme la demande d'électricité ou les apports hydrauliques. Ainsi, bien que la modélisation du parc soit beaucoup plus simplifiée qu'aux horizons courts, la résolution reste très complexe.

1.2 Optimisation non-différentiable

Dans ce contexte, il est généralement inévitable de recourir à des méthodes de décomposition. Une technique courante est la relaxation Lagrangienne [Lem01] de la contrainte de demande, qui couple les moyens de production, aussi appelée décomposition par les prix. Le problème primal est remplacé par un problème dual décomposable, plus simple, mais généralement non-différentiable. Des algorithmes spécifiques doivent donc être employés pour tenir compte du caractère non lisse de la fonction objectif. Parmi les possibilités, les méthodes de faisceau (e.g. [BGLS06, Ch. 9]) sont particulièrement efficaces : à la fois stables et précises.

Toutefois, le problème dual résultant de la relaxation Lagrangienne, s'il est plus simple que le problème initial, est lui aussi souvent de grande taille. **Le fil rouge de cette thèse a été de considérer des évolutions possibles pour que les algorithmes d'optimisation non-différentiable classiques puissent être appliqués à des problèmes de très grande taille.**

Nous avons poursuivi deux directions différentes, liées à cette même idée.

- Lorsque la dimension du problème dual est très élevée, les algorithmes d'optimisation non-différentiable peuvent échouer dans la mise à jour des variables duales. Comme possible réponse à cette limitation, nous avons étudié le comportement de ces algorithmes dans le cadre d'une *relaxation Lagrangienne dynamique*.
- La relaxation Lagrangienne permet souvent de bénéficier de la structure séparable de la fonction duale. L'évaluation de la fonction est donc facilitée car elle peut se faire individuellement pour chaque sous-fonction. Cependant, si le nombre de sous-fonctions est élevé et que leur évaluation n'est pas triviale, le calcul de la fonction duale peut à nouveau poser des limites pour les algorithmes employés. Dans ce cas, il y a un enjeu important à pouvoir effectuer une mise à jour des variables duales en n'évaluant qu'une partie des sous-fonctions, dans une approche de type *incrémentale*.

1.3 Relaxation Lagrangienne Dynamique

La relaxation Lagrangienne est couramment utilisée pour calculer des bornes sur des problèmes d'optimisation avec nombres entiers ou pour résoudre des problèmes convexes de grande taille. Toutefois, lorsque le nombre de contraintes à dualiser est très grand (exponentiel en la dimension du problème primal), une dualisation explicite peut s'avérer impossible ou inutile. Or, l'intérêt pour de tels problèmes n'a cessé de croître lors des deux dernières décennies, en particulier en optimisation combinatoire, mais aussi pour résoudre des problèmes continus mais couplés de grande taille.

Afin de réduire la dimension de l'espace dual, de nombreux auteurs ont proposé récemment des heuristiques basées sur une relaxation Lagrangienne partielle. Dans cette approche, seul un sous-ensemble de contraintes difficiles est dualisé, tandis que les autres contraintes sont ignorées. Une procédure de séparation doit permettre de mettre à jour

dynamiquement ce sous-ensemble, en sélectionnant quelques contraintes violées par l’itéré primal courant.

Cette approche de type “relax-and-cut”, que nous appellerons *Relaxation Lagrangienne Dynamique* (RLD) dans la suite, s’est montrée très performante numériquement. La plupart des auteurs exploitant cette idée utilisent les algorithmes de sous-gradients pour la mise à jour des variables duales. Toutefois, plus récemment, des méthodes plus robustes ont été retenues, comme les méthodes de faisceau ou de centre analytique.

Dans le cadre de cette relaxation Lagrangienne dynamique, la fonction duale que l’on cherche à optimiser varie au fil des itérations. Par conséquent, les analyses classiques de convergence pour les algorithmes d’optimisation non-différentiable ne peuvent s’appliquer. Ici, il est nécessaire de baser l’analyse sur des raisonnements de type primal-dual pour étudier la convergence.

Une bonne gestion des contraintes entrantes ou sortantes est cruciale pour le bon déroulement de l’algorithme. Une approche naïve consisterait à exiger de la procédure de séparation d’être en mesure de trouver toutes les contraintes violées pour un point primal quelconque. Toutefois, cette exigence est idéaliste pour la plupart des problèmes pratiques. Il est donc important d’isoler les propriétés minimales à respecter afin de garantir la convergence de la méthode.

Ce travail a été fait par Belloni et Sagastizábal ([BS08]) lorsqu’une méthode de faisceau est utilisée pour la mise à jour des variables duales. Lors de cette thèse, nous avons démontré des résultats similaires pour les algorithmes de sous-gradients ([ES08a]).

1.4 Méthodes inexactes et incrémentales

La relaxation Lagrangienne d’une contrainte couplante permet de remplacer un problème primal difficile par un problème dual séparable. Par exemple, dans le cadre de la gestion de production électrique, la relaxation de la contrainte de satisfaction de la demande permet d’exhiber un problème dual décomposable par unité de production. Ainsi, l’évaluation de la fonction duale peut être faite de façon décentralisée, par une résolution séquentielle de sous-problèmes plus simples.

Lorsque le nombre de sous-problèmes est élevé, ou que ces sous-problèmes restent non-triviaux à résoudre, l’étape d’évaluation de la fonction duale peut être beaucoup plus coûteuse en termes de temps de calcul que l’itération de l’algorithme dual proprement dit. Dans ce cas, il peut être intéressant de mettre à jour les variables duales en n’ayant évalué qu’une partie des sous-fonctions, les plus faciles par exemple.

Ces approches de type incrémentales ont été étudiées dans le cadre des algorithmes de sous-gradient. Nous avons étudié une possible application dans le cadre des méthodes de faisceau. L’une des caractéristiques de ces méthodes est la présence d’un test de descente, qui permet de stabiliser le processus itératif en garantissant la qualité des directions choisies pour une sous-suite des itérés. Ce test repose intrinsèquement sur la valeur exacte de la fonction au nouvel itéré, valeur non disponible si l’on se contente de n’évaluer qu’un ensemble de sous-fonctions. Toutefois, grâce à des résultats récents de Kiwiel [Kiw06] sur les méthodes de faisceaux inexactes, nous avons pu obtenir des garanties de convergence. Les résultats numériques, sur des problèmes réels de gestion de production, semblent démontrer le potentiel d’une telle approche.

1.5 Organisation de la thèse et principales contributions

Organisation de la thèse. La première partie (Ch. 2) de cette thèse propose un rappel des principales notions d’optimisation non-différentiable. Nous nous intéressons

plus particulièrement aux méthodes de sous-gradient et de faisceau qui ont été au centre de nos recherches. En particulier, nous présentons des résultats récents concernant les approches inexactes ou incrémentales qui ont inspiré une partie de notre travail.

Dans la seconde partie (Ch. 3), nous nous intéressons à la relaxation Lagrangienne dynamique. Il s'agit d'étudier le comportement des algorithmes d'optimisation non-différentiable lorsque l'on procède à une relaxation partielle du problème initial et que l'on met à jour le sous-ensemble des contraintes dualisées au fil des itérations.

La troisième partie (Ch. 4) porte sur l'étude des méthodes incrémentales en optimisation non-différentiable, en particulier les méthodes de faisceau. Notre approche est à la fois théorique et appliquée à un problème réel de gestion de production.

Enfin, dans une dernière partie (Ch. 5) nous abordons plus en détail la gestion de production à moyen terme, en particulier les problématiques liées à la gestion des groupes de production avec contraintes de réserves. Nous serons amené à présenter un code industriel développé par la R&D d'EDF et les améliorations que nous avons proposées. L'objectif de ce code est de fournir des indicateurs pertinents aux gestionnaires des 58 tranches nucléaires dans le moyen-terme. La méthodologie de résolution repose sur la décomposition par les prix et un algorithme de faisceaux est chargé d'effectuer l'étape de coordination. Les améliorations proposées portent sur la récupération de solutions primales et le calcul de stratégies (feedbacks) locales pour les réserves d'énergies afin de permettre de simuler la gestion coordonnée du parc de production sur des scénarios quelconques.

Nous avons essayé de faire en sorte que chaque chapitre de la thèse puisse être lu indépendamment des autres. En particulier, une bibliographie clôture chacun d'entre eux. Les deux chapitres centraux présentant les principales contributions théoriques sont en anglais.

Contributions, publications et participation à des conférences.

Les travaux réalisés durant cette thèse ont donné lieu à plusieurs présentations au cours de conférences internationales ou de séminaires.

En particulier, les résultats sur l'application de méthodes de sous-gradient dans le cadre d'une relaxation Lagrangienne dynamique ont été présentés à la *Conférence EURO 2007* [EUR07] à Prague. Un article [ES08a] est en cours de préparation et sera soumis prochainement.

Les résultats sur l'application d'une stratégie incrémentale avec une méthode de faisceau ont été présentés à la *Conférence EngOpt 2008* à Rio de Janeiro et ont été publiés dans le Proceeding [ES08b].

Enfin, nous avons effectué plusieurs présentations lors de séminaires ou groupes de travail, notamment :

- à trois reprises, pour le *Seminário de Otimização* ([IMP]) de l'Institut de Mathématiques Pures et Appliquées, à Rio de Janeiro ;
- en interne à EDF R&D, lors de présentations techniques au département et de la *Journée des Thèses* ([EDF07]) ;
- au séminaire du Laboratoire de Finance des Marchés d'énergies (FiME) à l'Institut Henri Poincaré ([FIM]).

Bibliographie

- [BGLS06] J.F. Bonnans, J.C. Gilbert, C. Lemaréchal, and C. Sagastizábal, *Numerical optimization : Theoretical and practical aspects*, Springer, 2006.
- [BS08] A. Belloni and C. Sagastizábal, *Dynamic Bundle Methods*, Mathematical Programming (2008).
- [EDF07] EDF, *Journée des thésards du département OSIRIS – EDF R&D*, 2007.
- [ES08a] G. Emiel and C. Sagastizábal, *Dynamic Subgradient Methods*, Optimization Online, http://www.optimization-online.org/DB_HTML/2008/08/2077.html (2008).
- [ES08b] ———, *Incremental bundle methods for mid-term power planning*, Proceedings to the International Conference EngOpt – Rio de Janeiro, http://www.engopt.org/en/approved_papers.html (2008).
- [EUR07] EURO, *European Conference on Operational Research – Prague*, 2007.
- [FIM] FIME, *Séminaire du Laboratoire de Finance des Marchés d'énergies – <http://www.ihp.jussieu.fr/pratique/seminaires.html>*.
- [HUL96] J.B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms 1&2*, Springer, 1996.
- [IMP] IMPA, *Seminário de Otimização – <http://seminariosimpa.br>*.
- [Kiw06] K.C. Kiwiel, *A proximal bundle method with approximate subgradient linearizations*, SIAM J. on Optimization **16** (2006), no. 4, 1007–1023.
- [Lem01] C. Lemaréchal, *Lagrangian relaxation*, Computational Combinatorial Optimization (M. Jünger and D. Naddef, eds.), Springer Verlag, Heidelberg, 2001, pp. 115–160.

Chapitre 2

Optimisation non-différentiable : méthodes de sous-gradient, méthodes de faisceau

Sommaire

2.1	Introduction	21
2.2	Optimisation non-différentiable : quelques rappels	23
2.2.1	Eléments d'analyse convexe	23
2.2.2	Direction de descente	24
2.2.3	Conditions d'optimalité	25
2.3	Méthodes de sous-gradient	26
2.3.1	Propriétés de base	26
2.3.2	Reconstruction primale	27
2.3.3	Méthodes inexactes	29
2.3.4	Méthodes incrémentales	29
2.4	Méthodes de faisceau	30
2.4.1	Une méthode de plans coupants stabilisée	30
2.4.2	Propriétés de base	31
2.4.3	Solution primale	34
2.4.4	Méthodes inexactes	34
2.4.5	Méthodes incrémentales	37
2.5	Conclusion	37
	Bibliographie	42

2.1 Introduction

Les principaux problèmes d'optimisation auxquels nous nous sommes intéressé au cours de cette thèse ont une structure particulière. Ils sont issus de la dualisation par relaxation Lagrangienne d'un problème difficile, mais séparable par blocs. Le problème initial, dit primal, s'écrit sous la forme suivante :

$$\begin{cases} \min_{\{p^i\}_{i=1,\dots,Nu}} & \sum_{i=1}^{Nu} C^i(p^i) \\ \text{t.q.} & \sum_{i=1}^{Nu} p^i = d \quad (\text{contrainte couplante}) \\ & p^i \in \mathcal{P}^i, \end{cases} \quad (2.1)$$

où pour nos applications les fonctions C^i sont convexes et les ensembles $\text{conv}(\mathcal{P}^i)$ sont compacts, et $d \in \mathbb{R}^N$.

La résolution du problème (2.1) serait à priori grandement simplifiée si la contrainte couplante n'était pas prise en compte : il suffirait alors de résoudre séquentiellement Nu sous-problèmes indépendants. Le Lagrangien du problème, obtenu par relaxation de la contrainte couplante, bénéficie de cette structure séparable. Il est défini par :

$$L(p, x) := \sum_{i=1}^{Nu} \left(C^i(p^i) - \langle x, p^i - \frac{d}{Nu} \rangle \right) \quad (2.2)$$

$$=: \sum_{i=1}^{Nu} L^i(p, x), \quad (2.3)$$

où $x \in \mathbb{R}^N$ est le vecteur des multiplicateurs de Lagrange. La théorie de la dualité (e.g. [BGLS06, Ch. 8]) nous dit que l'inégalité suivante est toujours vérifiée :

$$(Primal) \Leftrightarrow \min_p \max_x L(p, x) \geq \max_x \min_p L(p, x) \Leftrightarrow (Dual). \quad (2.4)$$

Pour garantir l'existence d'une solution duale nous supposons qu'il existe un point de Slater : $\bar{p} = (\bar{p}^1, \dots, \bar{p}^n)$ tel que $\bar{p}^i \in \text{ri conv } \mathcal{P}^i$ et $\sum_{i=1}^{Nu} \bar{p}^i = d$. Sous cette condition, l'ensemble des solutions du problème dual est non vide, compact et convexe ([HUL96, Chap7,Thm2.3.2]). On note f^{opt} la solution du problème dual.

L'écart entre les valeurs des problèmes primal et dual est appelé saut de dualité. Le problème étant bien qualifié grâce à l'existence d'un point de Slater, le saut de dualité est nécessairement nul si le problème primal est convexe. Pour se placer dans le cadre d'un problème dual de minimisation, nous définissons la fonction duale par

$$\begin{aligned} f(x) &:= - \sum_{i=1}^{Nu} \min_{p^i} L^i(p^i, x) \\ &=: \sum_{i=1}^{Nu} f^i(x). \end{aligned} \quad (2.5)$$

Le problème dual consiste donc en

$$\begin{cases} \min & f(x) = \sum_{i=1}^{Nu} f^i(x), \\ \text{t.q.} & x \in \mathbb{R}^N \end{cases} \quad (2.6)$$

où les fonctions f_i sont nécessairement convexes mais peuvent ne pas être partout différentiables. La condition de Slater assure également que f est coercive. Il s'agit donc d'un problème d'optimisation convexe, séparable, non-différentiable : sa résolution doit faire appel à des techniques adaptées.

Enfin, rappelons que lorsque l'on évalue la fonction f en un point x^ℓ donné, les vecteurs $p_{x^\ell}^i$, solutions des sous-problèmes

$$\min_{p^i} C^i(p^i) - \langle x^\ell, p^i - \frac{d}{Nu} \rangle, \quad (2.7)$$

fournissent un sous gradient pour les fonctions f^i en x^ℓ . En effet, on peut montrer que $(p_{x^\ell}^i - \frac{d}{Nu}) \in -\partial f^i(x^\ell)$ pour $i = 1 \dots Nu$. On note $s^{i,\ell} := -(p_{x^\ell}^i - \frac{d}{Nu})$ et $s^\ell := \sum_{i=1}^{Nu} s^{i,\ell}$ avec également $s^\ell \in \partial f(x^\ell)$.

Dans ce chapitre, nous nous intéresserons aux méthodes permettant la résolution de problèmes du type (2.6). Nous nous sommes placé dans le cas non-contraint. Toutefois, les propriétés énoncées se généralisent au cas contraint. Dans un premier temps, nous présentons quelques rappels d'analyse convexe pour l'étude des méthodes d'optimisation non-différentiable. Nous nous intéressons ensuite aux algorithmes de sous-gradient et de faisceau, ainsi qu'aux développements récents concernant notamment les variantes inexactes ou incrémentales de ces méthodes.

2.2 Optimisation non-différentiable : quelques rappels

Pour étudier les problèmes d'optimisation non-différentiable, il est nécessaire de recourir à des outils spécifiques dont l'étude est basée sur l'analyse convexe (e.g. [HUL01]). En effet, l'hypothèse de différentiabilité utilisée généralement en optimisation doit ici être remplacée par des arguments utilisant les propriétés de la convexité. Dans la première section de ce chapitre, nous introduirons rapidement quelques éléments de l'analyse convexe requis pour l'étude de l'optimisation non-différentiable. Nous aborderons ensuite deux méthodes souvent retenues pour la résolution de ce type de problèmes : les algorithmes de sous-gradient et de faisceau. Dans les deux cas, nous présenterons les méthodes de récupération primale et les avantages pouvant être tirés de la structure sommable du problème. Nous nous intéresserons également à l'application de ces méthodes dans un contexte inexact.

Nous utilisons dans ce qui suit une notation standard. Le produit scalaire euclidien entre deux vecteurs x et y dans \mathbb{R}^N est noté $\langle x, y \rangle$ et la norme correspondante $|x|^2 = \langle x, x \rangle$.

2.2.1 Éléments d'analyse convexe

Définition 2.2.1 Une fonction f de \mathbb{R}^N dans \mathbb{R} est convexe si pour tout $x, y \in \mathbb{R}^N$ et $\lambda \in [0, 1]$ l'inégalité suivante est vérifiée :

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$$

Une propriété géométrique importante d'une fonction convexe est qu'en tout point, il existe une fonction affine minorant la fonction et qui coïncide avec elle en ce point. Les fonctions convexes sont donc des fonctions dont l'épigraphe est supporté par des hyperplans tangents.

Lorsque la fonction est différentiable en un point x , l'ensemble de plans tangents est limité à un seul élément : le gradient $\nabla f(x)$. Toutefois, aux points de non-différentiabilité, il existe une infinité de plans tangents. L'ensemble des pentes de ces plans supportant l'épigraphe de la fonction en x est appelé sous-différentiel et est défini par :

$$\partial f(x) := \{s \in \mathbb{R}^N : f(x) + \langle s, y - x \rangle \leq f(y) \text{ pour tout } y \in \mathbb{R}^N\}.$$

Pour tout élément du sous-différentiel $s \in \partial f(x)$, appelé sous-gradient, l'inégalité suivante est vérifiée :

$$s \in \partial f(x) \Leftrightarrow f(\cdot) \geq f(x) + \langle s, \cdot - x \rangle. \quad (2.8)$$

Comme nous sommes en dimension finie et que la fonction est définie sur \mathbb{R}^N tout entier, nous pouvons affirmer que f est nécessairement continue et localement lipschitzienne ([HUL96, Ch.1, Thm 3.1.1]). De plus, elle est presque partout différentiable ([HUL96,

Ch.1,Thm 4.2.1]). Toutefois, bien que l'ensemble des points de non-différentiabilité soit de mesure nulle, en pratique la solution du problème (2.6) se trouve très souvent dans cet ensemble, comme c'est le cas sur la figure 2.1 à droite. Graphiquement, ces points sont caractérisés par des "coins".

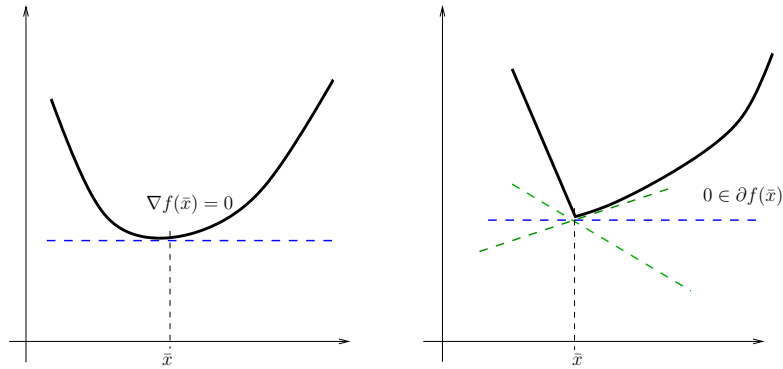


FIG. 2.1 – Non-différentiabilité

2.2.2 Direction de descente

Les algorithmes utilisés pour résoudre des problèmes d'optimisation non-différentiable ne diffèrent pas beaucoup, au premier abord, des algorithmes standard. A l'itératin ℓ -ième, ils consistent schématiquement en la recherche d'une direction de descente d^ℓ et d'une longueur de pas $t_\ell > 0$. Le nouvel itéré s'obtient ensuite par $x^{\ell+1} = x^\ell + t_\ell d^\ell$. Toutefois, certaines caractéristiques de l'analyse non-différentiable rendent nécessaire l'étude de méthodes spécifiques.

Tout d'abord, en optimisation différentiable, la direction opposée au gradient est toujours une direction de descente. Cette propriété ne s'étend pas à l'optimisation non-différentiable, comme le montre le graphique 2.2 où les lignes de niveaux des normes L^2 (différentiable, à gauche) et L^1 (non-différentiable, à droite) sont représentées. Au point de non-différentiabilité $x^\ell = (1, 0)$, la direction $s \in -\partial f(x^\ell)$ représentée sur la figure de droite n'est pas une direction de descente pour la norme L^1 .

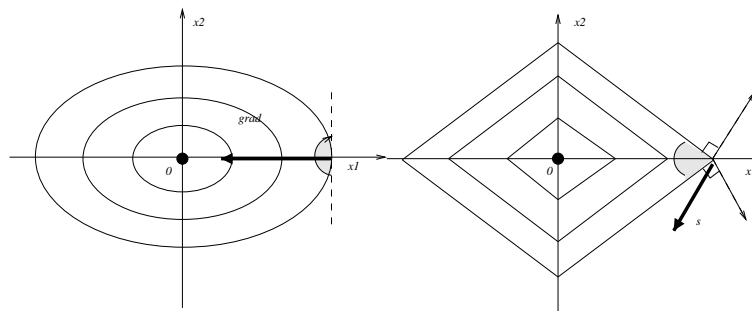


FIG. 2.2 – Direction du sous-gradient

En réalité, seule une direction faisant un angle obtus avec tous les éléments du sous-différentiel est une direction de descente ([HUL96, Ch.7,Thm 1.1.2]). On voit donc que

la connaissance de la structure du sous-différentiel en l'itéré courant est importante pour générer un nouveau point de qualité.

Cependant, dans la plupart des applications pratiques, le sous-différentiel n'est pas connu dans son intégralité. Le plus souvent, seul un sous-gradient est disponible. Les algorithmes doivent donc calculer un nouvel itéré en ayant une connaissance très limitée du problème traité. On parle dans ce contexte d'optimisation basée sur une *boîte-noire*, ou sur un *oracle*. Cette dénomination illustre le fait que pour un itéré x^ℓ donné, l'algorithme ne dispose par un appel à cet oracle que de la valeur de la fonction $f(x^\ell)$ et d'un sous gradient $s^\ell \in \partial f(x^\ell)$, afin de calculer le nouveau point $x^{\ell+1}$. La manière dont ces valeurs sont calculées par l'oracle (ou boîte noire) n'est pas connue par l'algorithme utilisé pour minimiser le problème (2.6), comme l'illustre le graphique 2.3.

Afin de générer de bonnes directions dans ce contexte de boîte noire, il y a donc un enjeu à tenter d'approcher la structure du sous-différentiel. C'est l'un des objectifs des méthodes de faisceau, comme nous le verrons à la section (2.4).

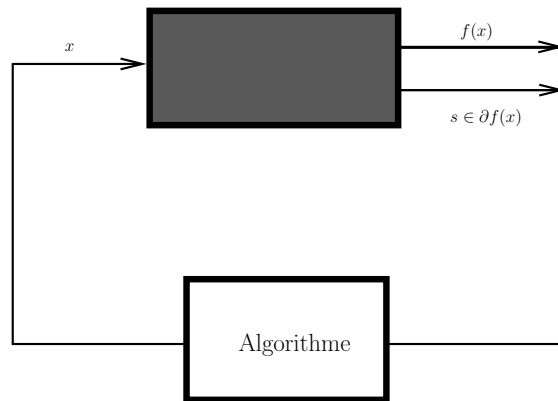


FIG. 2.3 – Boîte Noire / Oracle

2.2.3 Conditions d'optimalité

Enfin, signalons comme autre caractéristique importante des problèmes non-différentiables la difficulté d'établir des tests d'arrêts qui soient applicables. En effet, la condition d'optimalité pour un point \bar{x} s'écrit à l'aide de l'une des trois propriétés équivalentes :

$$\left\{ \begin{array}{l} (i) \quad f(x) \geq f(\bar{x}) \text{ pour tout } x \in \mathbb{R}^N, \\ (ii) \quad 0 \in \partial f(\bar{x}), \\ (iii) \quad f'(\bar{x}, d) \geq 0 \text{ pour tout } d \in \mathbb{R}^N. \end{array} \right.$$

Toutefois, la condition d'optimalité ci-dessus ne fournit pas un test d'arrêt applicable dans le contexte d'une optimisation de type boîte noire. En effet, pour l'itéré courant, un appel à la boîte noire nous fournit une information minimale, constituée de la valeur de la fonction et d'un sous-gradient. Par conséquent, même si l'itéré courant est optimal, i.e. $0 \in \partial f(x^\ell)$, lorsque la fonction est non-différentiable en ce point il est probable que le sous-gradient s^ℓ renvoyé par l'oracle ne permette pas de déclencher le test d'arrêt : la norme $|s^\ell|$ peut rester plus grande qu'une constante $\eta > 0$. Pour résoudre ce problème, on introduit une notion d'optimalité approchée :

$$\exists \bar{s} \in \partial_\epsilon f(\bar{x}) : |\bar{s}| < \eta, \quad \epsilon < \delta,$$

où $\partial_\epsilon f(\bar{x})$ dénote le sous-différentiel approché de f en \bar{x} :

$$\partial_\epsilon f(x) = \{s \in \mathbb{R}^N : f(y) \geq f(x) + \langle s, y - x \rangle - \epsilon, \forall y \in \mathbb{R}^N\}.$$

En effet, si cette condition d'optimalité approchée est vérifiée, alors l'inégalité suivante s'en déduit par Cauchy-Schwartz : pour tout $y \in \mathbb{R}^N$,

$$f(y) \geq f(\bar{x}) - |\bar{s}| |y - \bar{x}| - \epsilon.$$

Cette condition relâchée fait donc intervenir deux paramètres. Nous verrons à la section 2.4.2 que les méthodes de faisceaux utilisent cette condition comme test d'arrêt.

Dans ce qui suit, nous présentons deux approches parmi les plus utilisées pour résoudre les problèmes d'optimisation non-différentiable : les algorithmes de sous-gradient et de faisceau. Ces méthodes sont compatibles avec le contexte de boîte noire expliqué ci-dessus. Les travaux présentés dans les chapitres suivants portent sur ces deux méthodes et les propriétés présentées ci-dessous seront reprises par la suite.

2.3 Méthodes de sous-gradient

Les méthodes de sous-gradient sont les plus simples pour résoudre les problèmes d'optimisation non différentiable. Elles ont été introduites par Shor [Sho62]. Les premiers résultats de convergence sont dus à Ermol'ev [Erm66], Polyak [Pol67, Pol69] ou Nemirovskii [NY78]. Shor [Sho85] présente un aperçu général des premiers développements sur les méthodes de sous-gradient.

2.3.1 Propriétés de base

Ces méthodes s'apparentent aux algorithmes de gradient dans le cas différentiable, en effectuant un pas dans la direction opposé au sous-gradient. Le schéma standard est le suivant :

Algorithme 2.3.1 (*de sous-gradient*)

1. **Initialisation** : Choisir $x^0, t_0 \geq 0, \ell = 0$.
2. **Oracle** : Appeler la boîte noire en $x^\ell : (f(x^\ell), s^\ell \in \partial f(x^\ell))$.
3. **Mise à jour du pas** : Choisir une longueur de pas $t_\ell > 0$.
4. **Itération** : Définir $x^{\ell+1} := x^\ell - t_\ell s^\ell$; changer ℓ en $\ell + 1$; revenir en 2.

Comme nous l'avons dit, la direction $d^\ell := -s^\ell$ n'est pas nécessairement une direction de descente. Toutefois, si la longueur de pas t^ℓ est suffisamment petite, le nouvel itéré $x^{\ell+1}$ se rapprochera de la solution du problème. En effet, en appliquant l'inégalité de sous-gradient (2.8) on obtient que pour tout $y \in \mathbb{R}^N$,

$$\begin{aligned} |x^{\ell+1} - y|^2 &= |x^\ell - y|^2 + t_\ell^2 |s^\ell|^2 + 2t_\ell \langle y - x^\ell, s^\ell \rangle \\ &\leq |x^\ell - y|^2 + t_\ell^2 |s^\ell|^2 + 2t_\ell (f(y) - f(x^\ell)). \end{aligned}$$

En remplaçant y par \bar{x} solution du problème (2.6), on a que $f(\bar{x}) - f(x^\ell) < 0$, donc une longueur de pas faible doit permettre de se rapprocher de \bar{x} . En réalité, il suffit de choisir une longueur de pas telle que ([Pol69])

$$t_\ell = \alpha \left(\frac{(f(x_\ell) - f(\bar{x}))}{|s^\ell|^2} \right), \text{ avec } \alpha \in]0, 2[.$$

Toutefois, cela suppose que la valeur optimale du problème (2.6) soit connue. Sur la figure 2.2 à droite, on observe effectivement qu'un léger déplacement permet de se rapprocher de l'origine.

Une règle classique garantissant la convergence théorique de l'algorithme consiste à choisir une série divergente de carré sommable pour les longueurs de pas :

$$\sum_{\ell} t_{\ell} = \infty \text{ et } \sum_{\ell} t_{\ell}^2 < \infty. \quad (2.9)$$

Théorème 2.3.2 (Ermol'ev, [Erm66]) *Considérons l'algorithme (2.3.1) où à l'étape 3 la longueur de pas t_{ℓ} satisfait l'hypothèse (2.9). Alors $\{x^{\ell}\}$ converge vers une solution du problème (2.6).*

En pratique, l'application d'une telle stratégie résulte en des performances numériques insuffisantes. Les stratégies de longueur de pas les plus efficaces cherchent à approcher $f(\bar{x})$ en calculant des estimations de la valeur optimale du problème (e.g. [HK71, GK99, KLL99]). D'autres approches consistent à utiliser l'information obtenue lors des précédents appels à l'oracle, en plus du sous-gradient actuel, afin d'établir la nouvelle direction. Cela permet de limiter les phénomènes de zig-zag en maintenant une trajectoire globale plus lisse et d'accélérer ainsi la vitesse de convergence lorsque l'on s'approche de la solution ([Sho70]).

Plusieurs remarques sur les limitations de ces méthodes peuvent être faites :

- on ne dispose pas de test d'arrêt naturel permettant de détecter que l'on est suffisamment proche de la solution ;
- ces méthodes ne garantissent pas la descente de l'objectif entre deux itérés consécutifs, pouvant mener à des phénomènes oscillatoires et à de mauvaises performances numériques ;
- les méthodes de mise à jour des longueurs de pas les plus efficaces sont parfois dépourvues de résultats de convergence ;
- la vitesse de convergence est souvent faible, en particulier lorsque l'on s'approche de la solution.

Malgré ces limites, ces méthodes restent très largement utilisées par les praticiens, notamment dans la communauté combinatoire. Ce succès est du en partie à leur simplicité d'implémentation, mais également à la qualité de leur comportement numérique pour des problèmes de grande taille, lorsque la précision de la solution approchée importe peu.

Dans le chapitre 3, nous étudions le comportement de ces méthodes dans le cadre de la relaxation Lagrangienne dynamique, particulièrement adaptée à la résolution de problèmes où un très grand nombre de contraintes doit être dualisé. Cette approche consiste à considérer un problème dual où seul un sous-ensemble des contraintes difficiles est dualisé, les autres contraintes n'étant pas considérées. Au fil des itérations, ce sous-ensemble évolue. De nombreux auteurs ont appliqué des heuristiques basées sur ce principe en coordination avec un algorithme de sous-gradient pour mettre à jour les variables duales. Nous nous sommes intéressés aux propriétés théoriques de convergence de ces méthodes.

2.3.2 Reconstruction primale

En pratique, le problème d'optimisation non-différentiable que nous considérons est en réalité le dual du problème 2.1 :

$$\begin{cases} \min_{\{p^i\}_{i=1, \dots, Nu}} & \sum_{i=1}^{Nu} C^i(p^i) \\ \text{t.q.} & \sum_{i=1}^{Nu} p^i = d \\ & p^i \in \mathcal{P}^i. \end{cases}$$

Pour un vecteur dual $x \in \mathbb{R}^N$ donné, l'évaluation de la fonction duale définie en (2.5) est faite séquentiellement, en résolvant Nu sous-problèmes locaux qui s'écrivent pour un indice i et un point x^ℓ donnés :

$$\min_{p^i} C^i(p^i) - \langle x^\ell, p^i - \frac{d}{Nu} \rangle.$$

Les solutions locales renvoyées $p^\ell := (p^{1,\ell}, \dots, p^{Nu,\ell})$ ne convergent à priori pas vers une solution du problème primal initial (2.1), sauf cas particulier de *stricte* convexité.

Toutefois, pour un problème primal convexe, les méthodes de sous-gradient permettent généralement de reconstruire une suite primale convergeant vers une solution. La technique la plus couramment utilisée pour obtenir asymptotiquement une solution consiste à construire une suite, dite *ergodique*, moyennant les solutions primales successives. Ces procédures ont été introduites par Nemirovskii et Yudin [NY78] et Shor [Sho85] pour calculer des solutions primales optimales pour des problèmes linéaires. Elles ont été validées dans un cadre théoriques plus général par Larsson et al. [LPS99], Serali et Choi [SC96], Anstreicher et Wolsey [AW93]. Plus récemment, Nedić et Ozdaglar [NO07] donnent une estimation du taux de convergence de la solution primale lorsqu'une stratégie de longueur de pas constante est retenue pour la mise à jour de l'itéré dual.

En posant $\Sigma_\ell := \sum_{j=0}^{\ell} t_j$, la suite ergodique proposée par Shor ([Sho85]) est définie par

$$\hat{\pi}^\ell = \sum_{j=0}^{\ell} \alpha_\ell^j p^j, \text{ où } \alpha_\ell^j := \frac{t_j}{\Sigma_\ell}. \quad (2.10)$$

Théorème 2.3.3 (Shor([Sho85]), Larsson et al.[LPS99]) *Pour un problème primal 2.1 convexe, considérons l'algorithme (2.3.1) où à l'étape 3 la longueur de pas t_ℓ satisfait l'hypothèse (2.9). Alors $\{x^\ell\}$ converge vers une solution du problème (2.6), la suite ergodique $\{\hat{\pi}^\ell\}$ définie par (2.10) est bornée et tout point d'accumulation π^{acc} est solution de (2.1).*

Une limitation de cette construction ergodique vient du fait que les nouveaux sous-gradients entrent dans la définition de la suite avec un poids de plus en plus faible. Autrement dit, on donne plus d'importance à des itérations anciennes, à priori moins bonnes que les itérations plus récentes. Cela a comme conséquence une convergence lente de la suite ergodique. Serali et al. [SC96] proposent des règles générales liant les coefficients α_ℓ^j aux longueurs de pas t_ℓ afin de conserver les propriétés de convergence. Cela permet de définir la suite ergodique en donnant plus de poids aux itérations récentes.

Une autre possibilité pour produire des solutions primales sur la base d'une méthode de sous-gradient pour un problème linéaire est l'algorithme *de Volume* (Barahona et Anbil [BA00]). Cet algorithme partage certaines idées des méthodes de faisceaux en effectuant un test de la qualité de l'itéré généré ([BMS02]). Cet algorithme tire son nom du fait que la solution primale d'un problème linéaire de grande taille est approchée par le volume entre les faces actives à la solution duale.

Nesterov [Nes05] propose une approche nouvelle pour construire un algorithme de sous-gradient primal-dual, cette approche étant motivée par le fait que dans les approches classiques l'information apportée par les dernières itérations entrent avec un poids de plus en plus faible dans la construction d'une solution primale ergodique. Dans [Nes05], les longueurs de pas dual sont mises en relation avec une suite primale en charge d'accumuler l'information passée. Cela permet une plus grande flexibilité dans le choix des poids donnés aux itérés successifs pour la reconstruction d'une solution primale. Une approche similaire est également proposée par Ruszczyński ([Rus07]).

2.3.3 Méthodes inexactes

Supposons que dans l'algorithme 2.3.1, le sous-gradient utilisé s^ℓ soit inexact, i.e. $s^\ell \in \partial_{\varepsilon_\ell} f(x^\ell)$ avec $\varepsilon_\ell \geq 0$. Un résultat général de convergence pour ce cadre est donné par [CL93].

Proposition 2.3.4 [CL93, Proposition 1.2] *Considérons l'algorithme 2.3.1 avec un oracle inexact tel que $s^\ell \in \partial_{\varepsilon_\ell} f(x^\ell)$. Si*

$$\begin{aligned} \varepsilon_\ell &\rightarrow 0, \\ t_\ell |s^\ell|^2 &\rightarrow 0, \\ \sum_{\ell} t_\ell &= \infty, \end{aligned}$$

alors $\liminf_{\ell \rightarrow \infty} f(x^\ell) = f^{opt}$.

Dans [AF07], les auteurs étudient la convergence d'une méthode inexacte de sous-gradient pour un problème contraint tout en y associant les techniques de *deflection* (stabilisation du processus par l'utilisation des sous-gradients passés).

Les auteurs de [AHS94] proposent d'appliquer une méthode inexacte de sous-gradient pour résoudre un problème d'optimisation stochastique où la fonction objectif définie comme une intégrale multi-dimensionnelle n'est évaluée que de façon approchée.

Les propriétés de convergence des méthodes inexactes de sous-gradient peuvent également se révéler utiles pour l'analyse des méthodes incrémentales ([SZ98],[NB01],[Kiw03]) que nous décrivons ci-dessous.

2.3.4 Méthodes incrémentales

Les approches incrémentales visent à tirer partie de la structure séparable de la fonction à optimiser dans le but d'accélérer le processus de résolution. En effet, lorsque le problème résolu par l'oracle pour évaluer la fonction f consiste en l'évaluation de nombreuses sous-fonctions indépendantes, il peut être intéressant de ne pas attendre l'évaluation de chaque sous-fonction pour mettre à jour la variable duale. C'est le cas par exemple du problème (2.6) auquel nous nous intéressons.

Les méthodes incrémentales sont utilisées depuis longtemps en optimisation différentiable, notamment pour la résolution de problèmes liés aux réseaux de neurones (e.g. [LT94, MS94]) ou à des problèmes de moindres carrés (e.g. [Ber97]). Ces méthodes ont un très bon comportement numérique général mais perdent souvent en efficacité lorsque l'on se rapproche de la solution. En effet, pour garantir la convergence la longueur des pas doit tendre vers 0 en $\frac{1}{\ell}$, ce qui ralentit le processus. Au contraire, si la longueur des pas est maintenue plus grande qu'une certaine constante, les itérés convergent vers un "cycle limite" autour de la solution ([SZ98]).

Il existe dans la littérature deux approches visant à tirer parti de la structure sommable du problème. Dans les travaux [SZ98] ou [NB01], des itérations de sous-gradient sont effectuées séquentiellement le long des sous-gradients locaux $s^{i,\ell} \in \partial f^i(x^\ell)$ renvoyés par chaque sous-problème. Une itération s'interprète donc comme un cycle de Nu sous-itérations de sous-gradient.

L'algorithme est le suivant :

Algorithme 2.3.5 (*de sous-gradient incrémental*)

1. **Initialisation** : Choisir x^0 , $t_0 \geq 0$, poser $x_0^0 := x^0$, $\ell = 0$.
2. **Mise à jour du pas** : Choisir une longueur de pas $t_\ell > 0$.
3. **Mise à jour incrementale** : Pour $i = 0, \dots, Nu - 1$,
 - [3.a.] **Oracle** : Appeler la boîte noire en $x_i^\ell : (f^i(x_i^\ell), s^{i,\ell} \in \partial f^i(x_i^\ell))$.
 - [3.b.] **Sous-itération** : Définir $x_{i+1}^\ell := x_i^\ell - t_\ell s^{i,\ell}$; changer i en $i + 1$.
4. **Itération** : Poser $x^{\ell+1} = x_{Nu}^\ell$, $x_0^{\ell+1} = x^{\ell+1}$, $\ell = \ell + 1$, revenir en 2.

Proposition 2.3.6 ([SZ98],[NB01]) *Considérons l'algorithme (2.3.5) où à l'étape 2. la longueur de pas t_ℓ satisfait l'hypothèse (2.9). Alors $\{x^\ell\}$ converge vers une solution du problème (2.6).*

La preuve de convergence de cet algorithme peut se faire en montrant que les conditions d'application de la Proposition 2.3.4 sont respectées, autrement dit qu'un cycle de sous-itérations s'apparente à une itération d'une méthode inexacte de sous-gradient dont l'erreur de linéarisation tend vers 0 (c.f. [NB01]).

Ces méthodes incrémentales ont été initialement analysées par Solodov et Zavriev [SZ98] pour des problèmes du type (2.6). Bertsekas et Nedić [NB01] donnent des résultats de convergence théoriques pour différentes stratégies de longueurs de pas et signalent des résultats numériques prometteurs. Des résultats théoriques complémentaires sont également donnés par Kiwiel [Kiw03]. Les résultats empiriques de [NB01] tendent à montrer que les meilleures performances sont obtenues lorsque l'on effectue une permutation aléatoire du cycle de sous-fonctions avant chaque itération.

2.4 Méthodes de faisceau

Les méthodes de faisceau (voir [HUL96, Kiw85b, BGLS06]) sont considérées parmi les plus efficaces pour la résolution de problème non-différentiables. Dans cette section, nous présentons les principes de base de ces méthodes ainsi que certains développements récemment proposés dans la littérature.

2.4.1 Une méthode de plans coupants stabilisée

Comme nous l'avons vu, les méthodes de sous-gradient n'utilisent à chaque itération que la dernière information reçue de la boîte noire. Les méthodes de plans coupants sont une proposition d'amélioration naturelle en conservant l'ensemble de l'information générée au sein d'un faisceau (*bundle*) d'hyperplans supports défini par :

$$\{(x^j, f(x^j), s^j \in \partial f(x^j)), \text{ pour } j \in \mathcal{B}_\ell\} .$$

Dans le cadre des méthodes de plans coupants, l'ensemble de l'information passée est conservée : $\mathcal{B}_\ell = \{1, \dots, \ell\}$. Ce faisceau de plans coupants permet de définir une approximation linéaire par morceaux minorante pour la fonction f , exacte aux points $\{x^j\}_{j=0 \dots \ell}$:

$$\check{f}_\ell(y) = \max_{j \in \mathcal{B}_\ell} \{f(x^j) + \langle s^j, y - x^j \rangle\} , \text{ pour tout } y \in \mathbb{R}^N . \quad (2.11)$$

Le nouvel itéré $x^{\ell+1}$ est obtenu comme solution du programme linéaire suivant

$$x^{\ell+1} \in \arg \min_{y \in C} \check{f}_\ell(y) .$$

où C est un ensemble contenant la solution du problème.

Toutefois, ces méthodes souffrent de nombreuses limitations. En particulier, ce ne sont pas des méthodes de descente et l'algorithme peut osciller fortement lorsque des coupes de pentes faibles sont ajoutées au modèle. Un exemple célèbre de problème où ces méthodes obtiennent de très mauvais résultats est donné par Nemirovskii ([NY79, §4.3.6]).

Un autre limitation vient de la complexité croissante du programme linéaire à résoudre à chaque itération. En effet, aucune sélection ou compression du faisceau de plans coupants ne peut être effectuée sans perdre la garantie de convergence.

Les méthodes de faisceau construisent également une approximation linéaire par morceaux de la fonction objectif, mais introduisent un terme de stabilisation et un contrôle des nouveaux itérés afin de garantir la descente. De plus, il est possible de borner la taille du faisceau en procédant à la suppression de certaines coupes ou à la compression de l'information.

2.4.2 Propriétés de base

Les méthodes de faisceau génèrent des *points candidats*, notés x^ℓ , qui servent à améliorer le modèle linéaire de f . Certains points candidats ont un rôle particulier : il s'agit des *centres de stabilité*, ils forment une sous-suite des points candidats et sont notés $\hat{x}^{k(\ell)}$ où $k(\ell)$ est incrémenté de 1 à chaque fois qu'un nouveau candidat est déclaré centre de stabilité. Les centres de stabilité doivent permettre de décroître de façon significative la valeur de la fonction objectif. Lorsqu'un point candidat est jugé apte à devenir centre de stabilité, on parle de *pas sérieux* pour l'algorithme. À l'inverse, lorsqu'un point candidat ne sert qu'à l'amélioration du modèle, il s'agit d'un *pas nul*. À l'issue de chaque itération, le modèle de f est amélioré par l'ajout d'un nouveau plan support.

Les points candidats sont obtenus à partir d'une version stabilisée du problème linéaire (2.4.1), afin d'éviter les phénomènes oscillatoires caractéristiques des méthodes de plans coupants. Cette stabilisation peut se faire par exemple par région de confiance ou par pénalisation du déplacement. Nous ne nous intéresserons ici qu'aux méthodes de faisceau dites proximales, où un terme de type proximal pénalise l'écart quadratique du nouvel itéré au centre de stabilité courant. Signalons toutefois que l'ensemble des méthodes de faisceau présentes dans la littérature peuvent être analysées sous dans le cadre de la théorie des méthodes de faisceau généralisées ([Fra02]).

Avec une méthode proximale, le nouveau candidat est obtenu comme solution du programme quadratique suivant :

$$x^{\ell+1} = \arg \min_{y \in \mathbb{R}^N} \tilde{f}_\ell(y) + \frac{1}{2} \mu_\ell |y - \hat{x}^{k(\ell)}|^2, \quad (2.12)$$

où le paramètre de pénalisation $\mu_\ell > 0$ est appelé *paramètre proximal*.

Cette stabilisation proximale a également comme avantage de permettre de limiter le nombre de plans coupants dans le faisceau, sans perdre les propriétés de convergence. Pour prendre en compte cette possibilité dans nos notations, l'ensemble \mathcal{B}_ℓ contient désormais l'indice des coupes présentes dans le faisceau à l'itération ℓ et non plus toutes les coupes générées depuis l'initialisation de l'algorithme. Le modèle de plans coupants peut toujours s'écrire comme (2.11).

Toutefois, nous nous plaçons désormais dans un cadre plus général en considérant une version désagrégée de ce modèle. En effet, lors de l'évaluation de la fonction duale (2.5) en x^ℓ , nous disposons de la valeur de chaque sous-fonction en ce point et d'autant de sous-gradients :

$$f^i(x^\ell) \text{ et } s^{i,\ell} \in \partial f^i(x^\ell), \text{ pour tout } i = 1 \dots Nu.$$

Cette information nous permet donc de construire un modèle de plans coupants pour

chaque sous-fonction :

$$\check{f}_\ell^i(y) = \max_{j \in \mathcal{B}_\ell^i} \left\{ f^i(x^j) + \langle s_\ell^j, y - x^j \rangle \right\}, \text{ pour tout } y \in \mathbb{R}^N.$$

où \mathcal{B}_ℓ^i représente le faisceau de coupes conservées pour la fonction f^i . Puisque $\sum_{i=0}^{Nu} \check{f}_\ell^i(\cdot) \geq \check{f}_\ell(\cdot)$, le modèle ainsi construit est plus fin que le modèle agrégé. Dans ce qui suit, nous conservons une désagrégation totale en Nu sous-modèles, toutefois il est possible d'effectuer une désagrégation partielle en regroupant certaines sous-fonctions par paquets. Pour alléger la notation, nous notons désormais $\check{f}_\ell := \sum_{i=0}^{Nu} \check{f}_\ell^i$.

Sans perte de généralité, une écriture plus condensée peut être proposée afin de ne pas avoir à conserver tous les vecteurs x^j pour $j \in \mathcal{B}_\ell^i, i = 1 \dots Nu$. Il suffit pour cela de prendre comme point de référence le centre de stabilité courant $\hat{x}^{k(\ell)}$. En effet, les sous-gradients s_ℓ^j sont également des sous-gradients approchés pour f^i en $\hat{x}^{k(\ell)}$: $s_\ell^j \in \partial_{e_{j,\ell}^i} f^i(\hat{x}^{k(\ell)})$ où les erreurs de linéarisation $e_{j,\ell}^i$ sont définies par

$$e_{j,\ell}^i := f^i(\hat{x}^{k(\ell)}) - f^i(x_i) - \langle s_\ell^j, \hat{x}^{k(\ell)} - x^i \rangle. \quad (2.13)$$

Les modèles peuvent donc se réécrire :

$$\check{f}_\ell(y) = f^i(\hat{x}^{k(\ell)}) + \max_{j \in \mathcal{B}_\ell} \{-e_{j,\ell}^i + \langle s_\ell^j, y - \hat{x}^{k(\ell)} \rangle\}. \quad (2.14)$$

Le modèle de plans coupants approchant la fonction f permet également de calculer une estimation de la descente de l'algorithme. On définit ainsi la *descente nominale* $\delta_{\ell+1}$ par :

$$\delta_{\ell+1} := f(\hat{x}^{k(\ell)}) - \check{f}_\ell(x^{\ell+1}) \geq 0. \quad (2.15)$$

Ce terme fournit un test d'arrêt naturel pour la méthode.

Le sous-problème (2.12) est un problème quadratique contraint par les coupes contenues dans le faisceau. En passant au dual et en considérant les variables duales $\{\alpha_j^i\}_{j \in \mathcal{B}_\ell^i}$ associées aux coupes, il est possible d'exhiber une structure particulière pour la solution $x^{\ell+1}$. En effet, le problème dual de (2.12) s'écrit

$$\begin{cases} \min_{\alpha_j^i} & \frac{1}{2} \left| \sum_{i=0}^{Nu} \sum_{j \in \mathcal{B}_\ell^i} \alpha_j^i s_\ell^j \right|^2 + \mu_\ell \sum_{i=0}^{Nu} \sum_{j \in \mathcal{B}_\ell^i} \alpha_j^i e_j^i \\ \text{t.q.} & \sum_{j \in \mathcal{B}_\ell^i} \alpha_j^i = 1, \alpha_j^i \geq 0, \forall i \end{cases} \quad (2.16)$$

La structure particulière de ce problème peut être exploitée afin d'employer des méthodes de résolution très efficaces. L'algorithme proposé par [Kiw94] permet notamment de tirer partie des résolutions successives de sous-problèmes très proches. Ainsi, bien qu'un problème quadratique soit résolu à chaque itération d'une méthode de faisceau proximale, le coût en temps de calcul n'est pas beaucoup plus élevé que si un programme linéaire devait être résolu, comme pour les méthodes de plans coupants.

En écrivant les conditions d'optimalité pour (2.16) on obtient les résultats suivants sur la solution $x^{\ell+1}$ du sous-problème (2.12).

Proposition 2.4.1 (e.g. Lemme 9.8 [BGLS06])

$$x^{\ell+1} = \hat{x}^{k(\ell)} - \frac{1}{\mu_\ell} \hat{s}^{\ell+1}, \quad (2.17)$$

où le *sous-gradient agrégé* $\hat{s}^{\ell+1}$ vérifie :

$$\hat{s}^{\ell+1} = \sum_{i=0}^{Nu} \hat{s}_i^{\ell+1}, \text{ avec} \quad (2.18)$$

$$\hat{s}_i^{\ell+1} = \sum_{j \in \mathcal{B}_\ell^i} \alpha_j^i s_\ell^j \text{ et } \sum_{j \in \mathcal{B}_\ell^i} \alpha_j^i = 1. \quad (2.19)$$

On a de plus :

$$\begin{aligned} \hat{s}_i^{\ell+1} &\in \partial \hat{\varepsilon}_{\ell+1}^i f^i(\hat{x}^{k(\ell)}), \text{ où } \hat{\varepsilon}_{\ell+1}^i = \sum_{j \in \mathcal{B}_i^{\ell}} \alpha_j^i e_{j,\ell}^i, \\ \hat{s}^{\ell+1} &\in \partial \hat{\varepsilon}_{\ell+1} f(\hat{x}^{k(\ell)}), \text{ où } \hat{\varepsilon}_{\ell+1} = \sum_{i=0}^{Nu} \hat{\varepsilon}_{\ell+1}^i, \\ \hat{s}^{\ell+1} &\in \partial \check{f}_\ell(x^{\ell+1}) \text{ et } \hat{s}_i^{\ell+1} \in \partial \check{f}_\ell^i(x^{\ell+1}). \end{aligned} \quad (2.20)$$

Le terme $\hat{\varepsilon}_{\ell+1}$ est nommé **erreur de linéarisation agrégée**.

Enfin, la descente nominale s'exprime à l'aide des éléments agrégés :

$$\delta_{\ell+1} = \hat{\varepsilon}_{\ell+1} + \frac{1}{2\mu_\ell} |\hat{s}^{\ell+1}|^2. \quad (2.21)$$

Pour maintenir les propriétés de convergence des méthodes de faisceau, seule l'information contenue dans le sous-gradient agrégé \hat{s}^ℓ doit être conservée, en plus de la dernière réponse de l'oracle. Cela signifie que les coupes ne participant pas à la construction de ce vecteur (i.e. telles que $\alpha_j^i = 0$) peuvent être supprimées. De plus, le faisceau peut être entièrement compressé à la fin d'une itération si les coupes $\hat{s}^{\ell+1}$ et $s^{\ell+1}$ sont conservées ([BGLS06, Lemme 9.10]). Toutefois, lorsqu'une coupe est supprimée, le modèle n'est plus exact au point correspondant.

Enfin, signalons que la mise à jour du paramètre μ_ℓ est primordiale pour garantir un bon comportement numérique de la méthode. Lemaréchal et Sagastizábal proposent un approche basée sur l'analyse convexe et la théorie de quasi-Newton ([LS97b]). L'idée est d'essayer d'approcher au second ordre la fonction objectif et d'accélérer ainsi la convergence de la méthode.

Nous pouvons maintenant écrire l'algorithme de faisceau proximal sous sa forme la plus simple.

Algorithme 2.4.2 (de faisceau proximal)

1. **Initialisation** : choisir $x^1 \in \mathbb{R}^N$, $\kappa < 1$, $m < 1$, évaluer $(f^i(x^1), g^i(x^1))$ pour $i = 1 \dots Nu$, définir \check{f}_i^1 , poser $\ell = k(\ell) = 1$ et $\hat{x}^1 = x^1$
2. **Candidat** : Résoudre (2.12) pour obtenir $x^{\ell+1}$, $\delta_{\ell+1}$, $\hat{s}^{\ell+1}$, et $\hat{\varepsilon}_{\ell+1}$.
3. **Test d'arrêt** : Si $\delta_{\ell+1} = 0$, stop.
4. **Oracle** : Evaluer $(f^i(x^{\ell+1}), g^i(x^{\ell+1}))$ pour $i = 1 \dots Nu$.
5. **Test de descente** : Si $\sum_{i=1}^{Nu} f^i(x^{\ell+1}) \leq \sum_{i=1}^{Nu} f^i(\hat{x}^{k(\ell)}) - m\delta_{\ell+1}$, déclarer un pas sérieux, sinon déclarer un pas nul.
 Si $x^{\ell+1}$ donne un pas sérieux, poser $\hat{x}^{k+1} = x^{\ell+1}$, $k = k + 1$, choisir $\mu_{\ell+1} \leq \mu_{\max}$.
 Si $x^{\ell+1}$ donne un pas nul, choisir $\mu_{\ell+1} \in [\mu_\ell, \mu_{\max}]$.
6. **Mise à jour** : Définir $\check{f}_{\ell+1}^i$ pour $i = 1 \dots Nu$, $\ell = \ell + 1$, revenir en 2.

De nombreux raffinements peuvent être apportés à cette méthode, notamment pour la stratégie de mise à jour du paramètre proximal μ_ℓ comme nous l'avons signalé précédemment. Il est également possible d'introduire un paramètre de recherche linéaire dans le programme quadratique (2.12) lorsque l'algorithme effectue des pas nuls ([LS97b]).

Les garanties de convergence sont liées à l'évolution du paramètre proximal μ_ℓ et sont rappelées dans le théorème 2.4.3.

Théorème 2.4.3 *Supposons que l'algorithme 2.4.2 génère un dernier pas sérieux $\hat{x}^{k(\text{last})}$ suivi d'une infinité de pas nuls $\{x^\ell\}_{\ell > \text{last}}$. Si de plus la suite $\{\mu_\ell\}_{\ell > \text{last}}$ est croissante, alors*

- le dernier pas sérieux $\hat{x}^{k(\text{last})}$ minimise f ;
- $\delta_{\ell+1} \rightarrow 0$.

Si au contraire l'algorithme génère une infinité de pas sérieux, en notant \mathcal{L}_s l'ensemble des indices des candidats choisis comme centres de stabilité, supposons que la sous-suite $\{\mu_\ell\}_{\ell \in \mathcal{L}_s}$ soit telle que

$$\sum_{\mathcal{L}_s} \frac{1}{\mu_\ell} = \infty \text{ et } \sum_{\mathcal{L}_s} \mu_\ell = \infty.$$

Alors la suite des pas sérieux est bornée et tout point d'accumulation est optimal pour le problème (2.6).

2.4.3 Solution primale

A l'instar de ce que nous avons présenté pour les méthodes de sous-gradient au §2.3.2, les solutions locales $p^\ell := (p^{1,\ell}, \dots, p^{Nu,\ell})$ renvoyées lors de l'évaluation de chaque sous-fonction par l'oracle ne convergent pas vers une solution du problème primal initial (2.1), sauf cas particulier de stricte convexité.

Toutefois, comme cela est montré par [FK00], si le problème primal est convexe les méthodes de faisceau fournissent naturellement une suite d'éléments convergeant vers une solution de (2.1). En effet, de façon intuitive, lorsque l'on résout le sous-problème (2.12) via son dual (2.16), on résout en réalité une approximation du bi-dual du problème (2.1). Les sous-gradients agrégés $\hat{s}_i^{\ell+1}$ fournissent donc des solutions approchés au problème primal (2.1). A la convergence, i.e. lorsque $\hat{s}^{\ell+1} = 0$ et $\hat{\varepsilon}_{\ell+1} = 0$, on a que $0 = \hat{s}^{\ell+1} \in \partial f(\bar{x})$ où $\bar{x} \in \mathbb{R}^N$ est solution du problème dual (2.6) et une solution du primal est alors donnée par

$$\hat{p}^{i,\ell} = \sum_{j \in B_i^c} \alpha_j^i p_j^i, \text{ pour } i = 1 \dots Nu.$$

Si l'algorithme ne s'arrête pas à l'étape 3, tout point d'accumulation de la suite $\{\hat{p}^{i,\ell}, = 1 \dots Nu\}$ (bornée) est solution du problème primal (2.1). Lorsque le problème primal n'est pas convexe, la suite $\{\hat{p}^\ell\}$ converge vers la solution du problème primal convexifié ([FK00]). Cette propriété est utilisée par [DL05, DGL05] comme point de départ d'une heuristique afin d'obtenir une solution approchée réalisable pour un problème primal avec nombres entiers.

2.4.4 Méthodes inexactes

Comme nous l'avons vu pour les méthodes de sous-gradients, il peut être difficile pour certains problèmes particulièrement complexes d'obtenir des solutions exactes aux sous-problèmes

$$\max_{p^i} C^i(p^i) - \langle x, p^i - \frac{d}{Nu} \rangle. \quad (2.22)$$

Ce peut-être le cas par exemple lorsque les sous-problèmes comportent des variables en nombres entiers et sont eux-mêmes résolus par relaxation Lagrangienne.

L'erreur introduite par l'oracle inexact peut porter à la fois sur la valeur de la fonction évaluée en l'itéré x^ℓ et sur le sous-gradient renvoyé. On note $(f_{x^\ell}^i, s_{x^\ell}^i)$ les informations renvoyées par la boîte noire dans ce contexte, leur caractère inexact s'exprimant à l'aide des deux inégalités suivantes :

$$\begin{aligned} f_{x^\ell}^i &\geq f(x^\ell)^i - E_i^f, \text{ et} \\ f(\cdot)^i &\geq f_{x^\ell}^i + \langle s_{x^\ell}^i, \cdot - x^\ell \rangle - E_i^s, \end{aligned}$$

où E^f et E^s représentent la précision de l'oracle. On déduit aisément les deux relations suivantes pour la coupe inexacte :

$$\begin{aligned} s_{x^\ell}^i &\in \partial_{E_i^f + E_i^s} f(x^\ell) \text{ et} \\ f_{x^\ell}^i &\in [f(x^\ell) - E_i^f; f(x^\ell) + E_i^s]. \end{aligned}$$

On note $E^f = \sum_{i=1}^{Nu} E_i^f$ et $E^s = \sum_{i=1}^{Nu} E_i^s$. Une coupe agrégée (f_{x^ℓ}, s_{x^ℓ}) inexacte est représentée sur la figure 2.4.

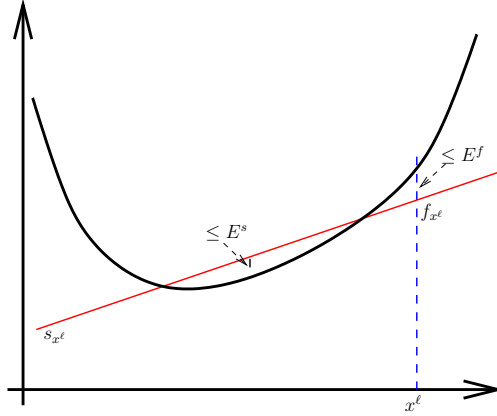


FIG. 2.4 – Linéarisation inexacte

Les méthodes de faisceau peuvent être adaptées à ce contexte. On citera en particulier les travaux de Kiwiel [Kiw85a, Kiw95, Kiw06] et [Hin01, Sol03].

[Kiw85a, Kiw95] considèrent des coupes inexactes dont l'erreur est contrôlable et peut être progressivement diminuée. [Hin01] traite le cas d'erreurs dont l'ampleur n'est pas connue mais qui ne portent que sur le sous-gradient ($E^f = 0$). [Kiw06] propose d'unifier ces approches au sein d'un même cadre théorique où les inexactitudes portent sur f et s et la marge d'erreur $E^f + E^s$ est fixée mais n'est à priori pas connue. Dans [KL07], ce cadre est étendu au cas contraint et cette approche est utilisée pour construire une méthode de faisceau inexacte adaptée à certaines applications en génération de colonnes. Une approche légèrement différente est retenue par [Sol03] pour le cas où l'erreur ne porte que sur l'évaluation de la fonction : l'auteur s'intéresse à la précision requise sur l'oracle pour obtenir la solution recherchée avec une certaine tolérance. L'algorithme employé est une méthode de faisceau classique avec un test d'arrêt adapté.

La principale différence entre la méthode proposée par [Kiw06] et une méthode classique de faisceau réside dans l'introduction d'un traitement spécifique du paramètre proximal lorsqu'un test permet de détecter que des erreurs importantes ont été introduites autour du centre de stabilité courant. En effet, les erreurs de linéarisation peuvent perturber le fonctionnement d'une méthode de faisceau classique et entraîner une descente nominale négative. Ce sera le cas si la valeur renvoyée par l'oracle au centre de stabilité, $f_{\hat{x}^{k(\ell)}}$, est inférieure au minimum du modèle de plans coupants dans la région de confiance définie par le paramètre proximal μ_ℓ . Si tel est le cas, il convient de choisir $\mu_{\ell+1} < \mu_\ell$ pour espérer obtenir un nouveau candidat avec une descente nominale positive.

Malgré l'imprécision des linéarisations renvoyées, il est toujours possible de définir un modèle de plans coupants approchant les sous-fonctions f^i par :

$$\varphi_\ell^i(y) = f_{\hat{x}^{k(\ell)}}^i + \max_{j \in B_\ell} \{-e_{j,\ell}^i + \langle s_{x^j}^i, y - \hat{x}^{k(\ell)} \rangle\},$$

où les erreurs de linéarisation sont définies par

$$e_{j,\ell}^i := f_{\hat{x}^{k(\ell)}}^i - f_{x_j}^i - \langle s_{x_j}^i, \hat{x}^{k(\ell)} - x_j^i \rangle.$$

Le nouveau point candidat donc être obtenu en résolvant (2.12) avec φ_ℓ^i remplaçant \check{f}_ℓ^i . Comme dans le cas exact, on obtient alors automatiquement le sous-gradient agrégé $\hat{s}^{\ell+1}$ et l'erreur de linéarisation agrégée $\hat{\varepsilon}_{\ell+1}$. Cependant, les modèles linéaires par morceaux φ_ℓ^i peuvent se trouver au-dessus des fonctions f^i en certains points. En notant $\Phi_\ell := \sum_{i=1}^{Nu} \varphi_\ell^i$, on peut toutefois garantir l'inégalité suivante :

$$\Phi_\ell(y) \leq f(\cdot) + E^s.$$

La descente nominale, définie par $\delta_{\ell+1} := \sum_{i=1}^{Nu} f_{\hat{x}^{k(\ell)}}^i - \Phi_\ell(x^{\ell+1})$ vérifie toujours $\delta_{\ell+1} = \hat{\varepsilon}_{\ell+1} + \frac{1}{\mu_\ell} |\hat{s}^{\ell+1}|^2$. Toutefois, $\hat{\varepsilon}_{\ell+1}$ peut être négatif et provoquer ainsi une descente nominale négative. En l'absence d'erreurs de linéarisations, la descente nominale doit satisfaire

$$\delta_{\ell+1} \geq \frac{1}{2} \mu_\ell |x^{\ell+1} - \hat{x}^{k(\ell)}|^2. \quad (2.23)$$

L'inégalité (2.23) peut être employée pour détecter ces erreurs.

Une mesure de l'optimalité de la solution courante x^ℓ est donnée par

$$V_\ell := \max\{\hat{\varepsilon}_{\ell+1}, |\hat{s}^{\ell+1}|\} \quad (\text{cf [Kiw06, Eq.2.16]}).$$

L'algorithme de faisceau dans le contexte inexact devient donc :

Algorithme 2.4.4 (de faisceau inexacte – [Kiw06])

1. **Initialisation** : choisir $x^1 \in \mathbb{R}^N, \kappa < 1, m < 1$, évaluer $(f_{x^1}^i, s_{x^1}^i)$ pour $i = 1 \dots Nu$, définir φ_1^i , poser $\ell = k(\ell) = 1$ et $\hat{x}^1 = x^1$
2. **Candidat** : Résoudre (2.12) avec φ_ℓ^i remplaçant \check{f}_ℓ^i pour obtenir $x^{\ell+1}$, $\delta_{\ell+1}$, $\hat{s}^{\ell+1}$, et $\hat{\varepsilon}_{\ell+1}$.
3. **Test d'arrêt** : Si $V_\ell = 0$, stop.
4. **Atténuation du bruit** : Si (2.23) n'est pas vérifié, poser $NAP = 1$, $\mu_{\ell+1} = \kappa \mu_\ell$, $\ell = \ell + 1$, aller en 2.
5. **Oracle** : Évaluer $(f_{x^{\ell+1}}^i, s_{x^{\ell+1}}^i)$ pour $i = 1 \dots Nu$.
6. **Test de descente** : Si $\sum_{i=1}^{Nu} f_{x^{\ell+1}}^i \leq \sum_{i=1}^{Nu} f_{\hat{x}^{k(\ell)}}^i - m \delta_{\ell+1}$, déclarer un pas sérieux, sinon déclarer un pas nul.
 Si $x^{\ell+1}$ donne un pas sérieux, poser $\hat{x}^{k+1} = x^{\ell+1}$, $f_{\hat{x}^{k+1}}^i = f_{x^{\ell+1}}^i$, $k = k + 1$, $NAP = 0$, choisir $\mu_{\ell+1} \leq \mu_{\max}$.
 Si $x^{\ell+1}$ donne un pas nul, si $NAP = 0$ choisir $\mu_{\ell+1} \in [\mu_\ell, \mu_{\max}]$, si $NAP = 1$ fixer $\mu_{\ell+1} = \mu_\ell$.
7. **Mise à jour** : Définir $\varphi_{\ell+1}^i$ pour $i = 1 \dots Nu$, $\ell = \ell + 1$, revenir en 2.

Les résultats de convergence donnés par [Kiw06] sont les suivants :

- Si la suite des pas sérieux est infinie, alors cette suite est bornée et tout point d'accumulation \hat{x}^{acc} est $E^f + E^s$ -optimal pour f :

$$f(\hat{x}^{acc}) \leq f^{opt} + E^f + E^s,$$

et la suite des valeurs $f_{\hat{x}^{k(\ell)}}$, décroissante, converge vers f^∞ telle que

$$f^\infty \leq f^{opt} + E^s.$$

- Si la suite des pas sérieux est finie, le dernier pas sérieux \hat{x}^{last} est $E^f + E^s$ -optimal pour f :

$$f(\hat{x}^{last}) \leq f^{opt} + E^f + E^s, \text{ et}$$

$$f_{\hat{x}^{last}} \leq f^{opt} + E^s.$$

Si de plus l’algorithme n’effectue qu’un nombre fini de pas d’atténuation du bruit, la suite $\{\mu_\ell\}$ se stabilise et la suite des pas nuls converge vers le dernier point de stabilité généré \hat{x}^{last} .

Au chapitre 4, nous adaptons ces résultats dans le cadre d’erreurs variant au fil des itérations afin d’utiliser ce cadre théorique pour étudier le comportement d’une méthode de faisceau incrémentale où les variables duales sont mises à jour après l’évaluation d’une partie seulement des sous-fonctions f^i . Nous nous intéressons également au comportement d’une méthode de faisceau plus classique, sans étape d’atténuation du bruit, lorsque l’erreur de l’oracle ne porte que sur la valeur des fonctions, i.e. $E^s = 0$.

2.4.5 Méthodes incrémentales

À notre connaissance, il n’y a pas à ce jour d’étude visant à appliquer une forme incrémentale des méthodes de faisceau pour des problèmes du type (2.6) où la fonction objectif s’écrit comme une somme de sous-fonctions éventuellement non-différentiables. Il existe toutefois des approches basées sur l’idée d’une évaluation partielle des sous-fonctions en conjonction avec une méthode de faisceau. Par exemple, Gaudioso et al. ([GGM06]) considèrent un problème où la fonction objectif est définie comme le maximum d’un nombre fini de sous-fonctions. L’algorithme de faisceau qu’ils proposent ne nécessite d’évaluer qu’une partie des sous-fonction à chaque itération. Cependant, la méthode non-incrémentale à la base de [GGM06] n’est pas complètement satisfaisante car elle nécessite une compression complète du faisceau à chaque pas sérieux.

On citera également les travaux de Kiwiel [KRR95, Kiw08] où l’auteur considère une fonction définie comme la somme de deux sous-fonctions qui sont représentées alternativement par leur linéarisation à chaque itération.

Au chapitre 4 on considère une méthode de faisceau pouvant accepter un oracle renvoyant une information incomplète. Nous montrons que cette approche peut s’interpréter comme une application des méthodes de faisceau avec linéarisation inexactes présentées à la section précédente. Les résultats numériques ([ES08]) sur un problème de gestion de production à moyen terme montrent qu’une telle stratégie peut s’avérer intéressante dans ce contexte où l’évaluation des sous-fonctions relatives à chaque usine de production est très coûteuse.

2.5 Conclusion

Pour conclure ce chapitre, nous dressons un aperçu de quelques travaux récents sur l’optimisation non-différentiable concernant en particulier les méthodes de faisceau.

Un pan important de ces recherches porte sur l’utilisation d’informations d’ordre 2 sur la fonction objectif via l’étude de sa régularisée de Moreau-Yosida. Comme en optimisation différentiable avec les méthodes de quasi-Newton, cet aspect est crucial afin d’accélérer la convergence. On citera en particulier les travaux de Mifflin ([Mif96],[MSQ98]) ou de Lemaréchal et Sagastizábal ([LS96],[LS97a],[LS97b]).

Une approche différente visant à accélérer la résolution des problèmes non-différentiables a récemment été développée. Elle repose sur la théorie dite “ $\mathcal{V}\mathcal{U}$ ” introduite par [LOS00] (voire également [Ous99, MS04]). L’idée générale repose sur une décomposition de l’espace en un point x donné en deux sous-espaces orthogonaux \mathcal{V} et \mathcal{U} de telle façon que autour de x la non-différentiabilité de f soit essentiellement due à sa restriction dans \mathcal{V} . Sous

certaines conditions, il est théoriquement possible de définir une trajectoire différentiable (rapide) tangente à l'espace \mathcal{V} et d'exhiber ainsi une expansion d'ordre 2 pour f . Dans [MS05] les auteurs montrent qu'une méthode de faisceau peut être utilisée pour construire une telle trajectoire tout en approchant les sous-espaces \mathcal{V} et \mathcal{U} . D'autres développements liés à l'utilisation d'une méthode de faisceau pour identifier la structure de ces sous-espaces sont proposés par [DSS08].

Un autre domaine d'importance est l'application des méthodes de faisceau à des problèmes non-différentiables non-convexes. Il s'agit en particulier des problèmes d'optimisation de valeurs propres, présents dans de nombreuses applications pratiques ([ANP08], [Ous00]) et qui peuvent être résolus à l'aide des méthodes de faisceau spectrales ([HR99]) lorsqu'ils sont convexes. Apkarian, Noll et al. [AN05a, AN05b, NPR08, ANP08] proposent d'adapter ces méthodes pour prendre en compte la non-convexité.

Sans être exhaustif, nous citons également les approches visant à résoudre des problèmes non-différentiables contraints ([SS05]) où des variantes à "mémoire limitée" adaptées à des problèmes de très grande taille [HMM07].

Bibliographie

- [AF07] G. Antonio and A. Frangioni, *Deflected conditional approximate subgradient methods*, Tech. Report 07-20, Dipartimento di Informatica, Università di Pisa, 2007.
- [AHS94] K.T. Au, J.L. Higle, and S. Sen, *Inexact subgradient methods with applications in stochastic programming.*, Math. Program. **63** (1994), 65–82.
- [AN05a] P. Apkarian and D. Noll, *Spectral bundle methods for non-convex maximum eigenvalue functions : first-order methods*, Math. Program. **104** (2005), no. 2-3, 701–727.
- [AN05b] ———, *Spectral bundle methods for non-convex maximum eigenvalue functions : second-order methods*, Math. Program. **104** (2005), no. 2-3, 729–747.
- [ANP08] P. Apkarian, D. Noll, and O. Prot, *A trust region spectral bundle method for nonconvex eigenvalue optimization*, SIAM Journal on Optimization **10** (2008), no. 1, 281 – 306.
- [AW93] K. Anstreicher and L. Wolsey, *On dual solutions in subgradient optimization*, Unpublished, 1993.
- [BA00] F. Barahona and R. Anbil, *The volume algorithm : producing primal solutions with a subgradient method*, Mathematical Programming **87** (2000), no. 3, 385–399.
- [Ber97] D.P. Bertsekas, *A new class of incremental gradient methods for least squares problems*, SIAM J. Optim. **7** (1997), 913–926.
- [BGLS06] J.F. Bonnans, J.C. Gilbert, C. Lemaréchal, and C. Sagastizábal, *Numerical optimization : Theoretical and practical aspects*, Springer, 2006.
- [BMS02] L. Bahiense, N. Maculan, and C. Sagastizábal, *The volume algorithm revisited : relation with bundle methods*, Mathematical Programming **94** (2002), no. 1, 41–69.
- [CL93] R. Correa and C. Lemaréchal, *Convergence of some algorithms for convex minimization*, Math. Program. **62** (1993), no. 2, 261–275.
- [DGL05] L. Dubost, R. Gonzalez, and C. Lemaréchal, *A primal-proximal heuristic applied to the french unit-commitment problem*, Math. Program. **104** (2005), no. 1, 129–151.
- [DL05] A. Daniilidis and C. Lemaréchal, *On a primal-proximal heuristic in discrete optimization*, Math. Program. **104** (2005), no. 1, 105–128.
- [DSS08] A. Daniilidis, C. Sagastizábal, and M. Solodov, *Identifying structure of nonsmooth convex functions by the bundle technique*, IMPA, <http://w3.impa.br/~sagastiz/pub-list.html>, July 2008.
- [Erm66] Y. Ermol’ev, *Methods for solving nonlinear extremal problems*, Kibernetika **2/4** (1966).

- [ES08] G. Emiel and C. Sagastizábal, *Incremental bundle methods for mid-term power planning*, Proceedings to the International Conference EngOpt – Rio de Janeiro, http://www.engopt.org/en/approved_papers.html (2008).
- [FK00] S. Feltenmark and K.C. Kiwiel, *Dual applications of proximal bundle methods, including Lagrangian relaxation of nonconvex problems*, SIAM J. on Optimization **10** (2000), no. 3, 697–721.
- [Fra02] A. Frangioni, *Generalized bundle methods*, SIAM J. on Optimization **13** (2002), no. 1, 117–156.
- [GGM06] M. Gaudioso, G. Giallombardo, and G. Miglionico, *An incremental method for solving convex finite min-max problems*, Math. Oper. Res. **31** (2006), no. 1, 173–187.
- [GK99] J.-L. Goffin and K. Kiwiel, *Convergence of a simple subgradient level method*, Mathematical Programming **85**(1) (1999), 207–211.
- [Hin01] M. Hintermüller, *A proximal bundle method based on approximate subgradients*, Comput. Optim. Appl. **20** (2001), no. 3, 245–266.
- [HK71] M. Held and R.M. Karp, *The traveling-salesman problem and minimum spanning trees. II*, Math. Programming **1** (1971), no. 1, 6–25.
- [HMM07] N. Haarala, K. Miettinen, and M. Makela, *Globally convergent limited memory bundle method for large-scale nonsmooth optimization*, Math. Program. **109** (2007), no. 1, 181–205.
- [HR99] C. Helmberg and F. Rendl, *A spectral bundle method for semidefinite programming*, SIAM J. on Optimization **10** (1999), no. 3, 673–696.
- [HUL96] J.B. Hiriart-Urruty and C. Lemaréchal, *Convex analysis and minimization algorithms 1&2*, Springer, 1996.
- [HUL01] ———, *Fundamentals of convex analysis*, Grundlehren Text Editions, Springer, 2001.
- [Kiw85a] K.C. Kiwiel, *An algorithm for nonsmooth convex minimization with errors*, Mathematics of computation **45** (1985), no. 171.
- [Kiw85b] ———, *Methods of descent for non-differentiable optimization*, Lecture Notes in Mathematics, Springer, 1985.
- [Kiw94] ———, *A Cholesky dual method for proximal piecewise linear programming*, Numerische Mathematik **68** (1994), no. 3, 325–340.
- [Kiw95] ———, *Approximations in proximal bundle methods and decomposition of convex programs*, J. Optim. Theory Appl. **84** (1995), no. 3, 529–548.
- [Kiw03] ———, *Convergence of approximate and incremental subgradient methods for convex optimization*, SIAM J. on Optimization **14** (2003), no. 3, 807 – 840.
- [Kiw06] ———, *A proximal bundle method with approximate subgradient linearizations*, SIAM J. on Optimization **16** (2006), no. 4, 1007–1023.
- [Kiw08] ———, *An alternating linearization bundle method for convex optimization and nonlinear multicommodity flow problems*, Rapport de recherche INRIA (2008), no. 6420.
- [KL07] K.C. Kiwiel and C. Lemaréchal, *An inexact bundle variant suited to column generation*, Math. Program. Ser. A. (2007).
- [KLL99] K.C. Kiwiel, T. Larsson, and P. O. Lindberg, *The efficiency of ballstep subgradient level methods for convex optimization*, Math. Oper. Res. **24** (1999), no. 1, 237–254.
- [KRR95] K.C. Kiwiel, C. H. Rosa, and A. Ruszczyński, *Decomposition via alternating linearization*, Tech. Report WP-95-051, Laxenburg, 1995.

- [LOS00] C. Lemaréchal, F. Oustry, and C. Sagastizábal, *The \mathcal{U} -Lagrangian of a convex function*, Transactions of the AMS **352** (2000), no. 2, 711–729.
- [LPS99] T. Larsson, M. Patriksson, and A. Strömberg, *Ergodic, primal convergence in dual subgradient schemes for convex programming*, Math. Program. **86** (1999), no. 2, Ser. A, 283–312.
- [LS96] C. Lemaréchal and C. Sagastizábal, *More than first-order developments of convex functions : primal-dual relations*, Journal of Convex Analysis **3** (1996), no. 2, 1–14.
- [LS97a] ———, *Practical aspects of the Moreau-Yosida regularization theoretical preliminaries*, Siam Journal on Optimization **7** (1997), no. 2, 367–385.
- [LS97b] ———, *Variable metric bundle methods : from conceptual to implementable forms*, Math. Program. **76** (1997), no. 3, 393–410.
- [LT94] Z.Q. Luo and P. Tseng, *Analysis of an approximate gradient projection method with applications to the backpropagation algorithm*, Optim. Methods Softw. **4** (1994), 85–101.
- [Mif96] R. Mifflin, *A quasi-second-order proximal bundle algorithm.*, Math. Program. **73** (1996), no. 1, 51–72.
- [MS94] O. Mangasarian and M. Solodov, *Serial and parallel backpropagation convergence via nonmonotone perturbed minimization*, Optim. Methods Softw. **4** (1994), 103–116.
- [MS04] R. Mifflin and C. Sagastizábal, *On the relation between \mathcal{U} -hessians and second-order epi-derivatives*, European Journal of Operational Research **157** (2004), no. 1, 28–38.
- [MS05] ———, *A \mathcal{VU} algorithm for convex minimization*, Math. Program. **104** (2005), no. 2-3, 583–608.
- [MSQ98] R. Mifflin, D. Sun, and L. Qi, *Quasi-newton bundle-type methods for non-differentiable convex optimization*, SIAM J. on Optimization **8** (1998), no. 2, 583–603.
- [NB01] A. Nedić and D. Bertsekas, *Incremental subgradient methods for nondifferentiable optimization*, SIAM J. on Optimization **12** (2001), no. 1, 109–138.
- [Nes05] Y. Nesterov, *Primal-dual subgradient methods for convex problems*, CORE Discussion Paper (2005).
- [NO07] A. Nedić and A. Ozdaglar, *Approximate primal solutions and rate analysis for dual subgradient methods*, Tech. Report 2753, LIDS, 2007.
- [NPR08] D. Noll, O. Prot, and A. Rondepierre, *A proximity control algorithm to minimize nonsmooth and nonconvex functions*, Pacific Journal of Optimization (2008).
- [NY78] A.S. Nemirovskii and D.B. Yudin, *Cesaro convergence of gradient method approximation of saddle points for convex-concave functions*, Doklady Akademii Nauk SSSR **239** (1978).
- [NY79] A.S. Nemirovsky and D.B. Yudin, *Problem complexity and method efficiency in optimization*, Interscience Series in Discrete Mathematics, John Wiley and Sons Ltd, 1979.
- [Ous99] F. Oustry, *The \mathcal{U} -lagrangian of the maximum eigenvalue function*, SIAM J. on Optimization **9** (1999), no. 2, 526–549.
- [Ous00] ———, *A second-order bundle method to minimize the maximum eigenvalue function*, Mathematical Programming **89** (2000), 1–33.

- [Pol67] B.T. Polyak, *A general method of solving extremum problems*, Doklady Akademii Nauk SSSR **174** (1967), no. 1, 33–36.
- [Pol69] ———, *Minimization of unsmooth functionals*, USSR Computational Mathematics and Mathematical Physics **9** (1969), 14–29.
- [Rus07] A. Ruszczynski, *A merit function approach to the subgradient method with averaging*, Optimization Methods and Software **23** (2007), no. 1.
- [SC96] H.D. Sherali and G. Choi, *Recovery of primal solutions when using subgradient optimization methods to solve lagrangian duals of linear programs*, Oper. Res. Lett. **19** (1996), no. 3, 105–113.
- [Sho62] N.Z. Shor, *Application of the gradient-descent method to solution of the network transport problem*, Cybern. Oper. Res. **1** (1962), 9–17.
- [Sho70] ———, *Utilization of the operation of space dilatation in the minimization of convex functions*, Kibernetika (1970).
- [Sho85] ———, *Minimization methods for nondifferentiable functions*, Springer Series in Computational Mathematics, vol. 3, Springer-Verlag, Berlin, 1985.
- [Sol03] M. Solodov, *On approximations with finite precision in bundle methods for nonsmooth optimization*, J. Optim. Theory Appl. **119** (2003), no. 1, 151 – 165.
- [SS05] C. Sagastizábal and M. Solodov, *An infeasible bundle method for nonsmooth convex constrained optimization without a penalty function or a filter*, SIAM Journal on Optimization **16** (2005), no. 1, 146–169.
- [SZ98] M. Solodov and S. Zavriev, *Error stability properties of generalized gradient-type algorithms*, J. Optim. Theory Appl. **98** (1998).

Chapter 3

Dynamic Lagrangian Relaxation

Sommaire

3.1 Introduction	43
3.2 Dynamic Lagrangian Relaxation	44
3.3 Dynamic subgradient methods	46
3.3.1 Subgradient iterates in a dynamic setting	46
3.3.2 Choosing index sets	50
3.3.3 Convergence results	51
3.4 Dynamic Bundle Methods	55
3.4.1 Bundle iterates in a dynamic setting	56
3.4.2 Choosing index sets	57
3.4.3 Convergence results	57
3.5 Conclusion	57
Bibliography	60

Lagrangian relaxation is commonly used to generate bounds for mixed-integer linear programming problems. However, when the number of dualized constraints is very large (exponential in the dimension of the primal problem), explicit dualization is no longer possible. In order to reduce the dual dimension, different heuristics were proposed. They involve a separation procedure to dynamically select a restricted set of constraints to be dualized along the iterations. This relax-and-cut type approach has shown its numerical efficiency in many combinatorial problems. We show primal-dual convergence of such a strategy when using an adapted subgradient method for the dual step.

3.1 Introduction

We consider the following optimization problem:

$$(\mathcal{P}) : \begin{cases} \max_p C(p) \\ p \in \mathcal{Q} \subseteq \mathbb{R}^{m_p} \\ g_j(p) \leq 0, \quad j = 1, \dots, n. \end{cases} \quad (3.1)$$

Our analysis covers two situations. First, it includes the case where (\mathcal{P}) is a mixed-integer linear program (MILP), which is the framework where the Dynamic Lagrangian

Relaxation appears in the literature. However, we may also consider a general convex program. The following assumptions resume this alternative :

$$\left. \begin{array}{lll} C : \mathbb{R}^{m_p} \rightarrow \mathbb{R} & \text{MILP} & \text{Convex} \\ g_j : \mathbb{R}^{m_p} \rightarrow \mathbb{R} & \text{affine} & \text{concave} \\ \mathcal{Q} & \text{conv } \mathcal{Q} \text{ compact} & \text{convex} \\ & & \text{compact} \end{array} \right\} \quad (3.2)$$

Our interest lies in applications where the number n is a huge number, possibly exponential in the primal dimension m_p . Introducing the notation $L = \{1, \dots, n\}$, a problem dual to (3.1) is

$$(\mathcal{D}) : \min_{x \geq 0} f(x), \quad \text{where} \quad f(x) := \max_{p \in \mathcal{Q}} \left\{ C(p) - \sum_{j \in L} g_j(p)x_j \right\} \quad (3.3)$$

is the dual function, possibly nondifferentiable. To ensure that the dual problem has a solution, we also require satisfaction of a Slater-type condition on the primal problem. Namely, there exists $p^0 \in \text{conv } \mathcal{Q}$ such that $g_j(p^0) < 0$ for all $j = 1, \dots, n$.

We recall that regarding the assumption (3.2), any algorithm involving the dual (D) can at best generate primal points solving a convexified form of the primal problem (P) (see [LR01] and references therein):

$$(\mathcal{P})_{\text{conv}} : \begin{cases} \max_p C(p) \\ p \in \text{conv } \mathcal{Q} \subseteq \mathbb{R}^{m_p} \\ g_j(p) \leq 0, \quad j = 1, \dots, n, \end{cases} \quad (3.4)$$

where \mathcal{Q} from (3.1) is replaced by its convex hull $\text{conv } \mathcal{Q}$. If the primal problem is convex, $(\mathcal{P})_{\text{conv}}$ is nothing else but the primal itself.

3.2 Dynamic Lagrangian Relaxation

Lagrangian relaxation is a common framework to compute bounds for complex mixed integer linear programming problems. The interest in solving problems with an exponential number of *hard-constraints* has recently grown. In this context, a complete dualization may be untractable or not computationally affordable. To overcome this difficulty, various heuristic approaches have been proposed, using a *partial* Lagrangian relaxation. In such methods only a subset J of the constraints is taken into account to build up the dual function

$$(\mathcal{D})_J : \begin{cases} \min_x f(x) \\ x \geq 0, x_{L \setminus J} = 0. \end{cases}$$

The subset of dualized constraints is updated along the process, yielding a *relax and cut* process or *dynamic* Lagrangian relaxation. In such a setting, the dual function is not fixed anymore and classical dual convergence analysis cannot be applied: a new primal-dual analysis must be developed instead.

The dynamic Lagrangian relaxation ideas can be traced back to the 80's with the Restricted Lagrangian Approach of Balas and Christofides [BC81] for the Traveling Salesman Problem. Other contributions have been given by Gavish [Gav85] for the Centralized Network Design, by Escudero, Guignard and Malik [EGM94] for the Sequential Ordering Problem, and by Hunting, Faigle and Kern for the Edge-weighted Clique Problem. [HFK01]. Good numerical results are reported for different large-scale combinatorial problems by Lucena et al. (e.g. [Luc92] and [Luc05]).

Essentially, all of these methods involve two main phases, one dual and the other one primal. While the dual block searches for optimality, the primal blocks aims at feasibility in (3.4). A third block, sometimes referred to as *separation step* but called “constraints management” step here, is in charge of coordinating the primal and dual phases.

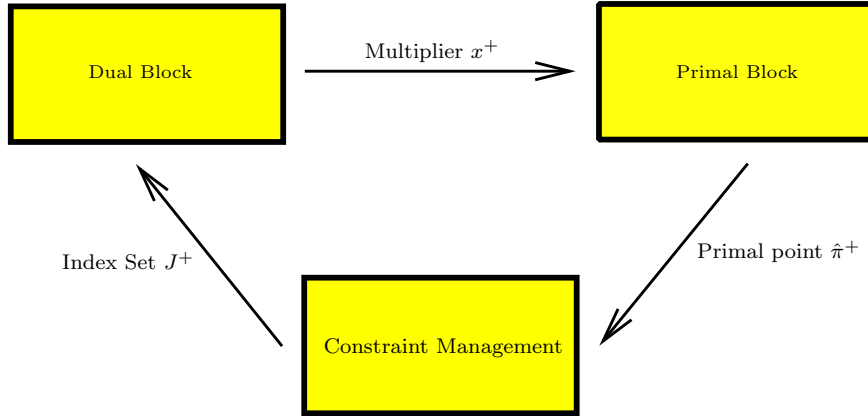


Figure 3.1: Dynamic Lagrangian Relaxation Scheme

The main steps performed by each block shown in Fig. 3.1 are the following:

Dual block: Having an index set J , defines a dual direction and stepsize, to produce a new multiplier x^+ , with $x_{L \setminus J}^+ = 0$.

Primal block: Having a multiplier x^+ , evaluates the dual function, to find a new primal point p^+ and defines a new convex primal point $\hat{\pi}^+$.

Constraints Management block: Having the new dual-primal information, uses the dual point to drop some constraints of the index set J and selects new constraints violated by the primal point to define a new index set J^+ .

In this type of methods, the separation oracle used to find violated inequalities in the constraints management step is of prior importance. Consider, for example, a dynamic method that makes a subgradient dual iteration. The previous primal iteration gave a primal point p satisfying $f(x) = C(p) - \sum_j g_j(p)x_j$, and the new dual iterate x^+ is defined by projecting $x + \lambda g(p)$ on the dual feasible set $\{x \geq 0 : x_{L \setminus J} = 0\}$. If the separation procedure is applied to p , and this procedure finds that no constraint out of J is violated by p , this means that x^+ is also the projection of $x + \lambda g(p)$ on the positive orthant. In other words, x^+ would correspond to a dual iterate in the static setting, that has a known convergence theory.

A first naïve approach, reflecting this reasoning, would be to require from the separation oracle to be able to find all constraints violated by any given primal point. This apparently convenient requirement, however, is exceedingly strong for combinatorial problems. Typically, separation procedures can find *some* of the constraints violated by a point, but not all of them. At best, for some classes of (easier) problems, the most violated constraint can be found. This is the so-called “maximum violation” oracle. For other problems, only heuristic procedures can be designed, and there is no guarantee of exact separation. Along the lines of [BS08], we focus here on *smart* separation oracles, that not only include the maximum violation oracle, but also some heuristic techniques, cf. (3.13) below.

The scheme in Fig.3.1 is very broad and comprehensive. In [BC81], [Gav85], and [EGM94], the dual phase proceeds to a full dual resolution by subgradient methods between two updates of the working index set J . By contrast, the *nondelayed relax and*

cut method of [Luc92] and [Luc05] updates constraints indices “on the fly” along iterations, without waiting for the full dual resolution of the relaxed restricted dual problem. More recently, more robust algorithms have been employed for the dual phase, like bundle methods in [FLR06] and [Hel04], or analytical center cutting planes method in [BdMV06]. Except for [Hel04], no convergence theory was developed for such methods. In [Hel04], the theory covers a spectral bundle method for the dual phase, provided a maximum violation oracle is used at the separation step and no constraints are dropped at null steps. In [BS08], general primal–dual convergence results are given for a dynamic method using a bundle algorithm for the dual iterations, with minimal assumptions on the separation procedure.

In view of the wide application of subgradient methods, we are interested here in obtaining similar results when a subgradient algorithm is used for the dual phase.

It is important to stress the fact that the dynamic Lagrangian relaxation can be interpreted as a column generation algorithm applied to the dual of (3.1). Column generation ([DW60],[GG61] and the more recent survey [Lue01]) is a powerful technique to solve large scale linear programs. However, these methods are often problem–dependent, in the sense that they require customization to each particular application when choosing the entering columns. To ensure convergence of the algorithm to the optimal dual solution, no general rule is given on which column should be added.

3.3 Dynamic subgradient methods

The paper is organized as follows. In section 3.3.1 we give some preliminary results on dynamic subgradients without any requirement on the constraint management block. Section 3.3.2 addresses the problem of selecting the index set of dualized constraints. In section 3.3.3 we present our convergence results. We conclude with some remarks.

Our notation is fairly standard. The Euclidean inner product between two vectors x and y is denoted by $\langle x, y \rangle$ and the norm $|x|^2 = \langle x, x \rangle$. We sometimes write x_J to refer to a subset J of components of a vector $x \in \mathbb{R}^n$.

Subgradient methods provide a simple yet efficient approach to solve dual problems arising from Lagrangian relaxation. They were first studied by Ermoliev [Erm76] and Polyak [Pol87]. Primal recovery procedures based on averaging of past subgradients were introduced by Nemirovskii and Yudin [NY78] and Shor [Sho85]. This technique has been later on computationally tested and completed by Sherali and Choi [SC96], Larsson et al. [LPS99], Kiwiel [Kiw03] among others. Recently, Nedić and Ozdaglar [NO07] gave estimates on the convergence rate of the primal averaged sequence when using a dual subgradient with constant stepsize.

We start by giving intermediate primal–dual results extending to the dynamic setting previous results in [SC96], [LPS99], and [Kiw03].

3.3.1 Subgradient iterates in a dynamic setting

Keeping Fig. 3.1 in mind, we consider an algorithmic pattern which has its ℓ^{th} iteration defined by the dual and primal blocks given below.

Dual block: Given the current dual iterate x^ℓ , a primal point $p^\ell \in \mathcal{Q}$ such that $f(x^\ell) = C(p^\ell) - \langle x^\ell, g(p^\ell) \rangle$ (and, hence, the subgradient $-g(p^\ell) \in \partial f(x^\ell)$) is available. Having a working index set J_ℓ , and a dual stepsize $\lambda_\ell > 0$, the new dual iterate is

$$\begin{aligned} x^{\ell+1} &:= \underset{\{x \geq 0: x_{L \setminus J_\ell} = 0\}}{\text{proj}} \left(x^\ell + \lambda_\ell g(p^\ell) \right) \\ &= x^\ell + \lambda_\ell (g(p^\ell) - \nu^\ell). \end{aligned}$$

Accordingly, the normal vector ν^ℓ has the expression

$$\nu_j^\ell = \begin{cases} \min(0, \frac{1}{\lambda_\ell} x_j^\ell + g_j(p^\ell)) & \text{if } j \in J_\ell, \\ g_j(p^\ell), & \text{otherwise.} \end{cases} \quad (3.5)$$

Primal block: Evaluates the dual function at $x^{\ell+1}$, to find a new primal point $p^{\ell+1}$. Defines a new convex primal point

$$\hat{\pi}^{\ell+1} = \sum_{i=0}^{\ell+1} \omega_{p_i}^{\ell+1} p^i,$$

using primal weights $\omega_{p_i}^\ell$, which satisfy for each ℓ the relations

$$\omega_{p_i}^\ell \geq 0 \text{ for } i = 0, \dots, \ell \quad \text{and} \quad \sum_{i=0}^{\ell} \omega_{p_i}^\ell = 1. \quad (3.6)$$

The Constraints Management block, defining the next working index set $J_{\ell+1}$, is given below, in § 3.3.2. Without further specifying this third block, it is possible to state some preliminary results that will be useful in the sequel.

Primal intermediate results

In a static setting, subgradient methods provide a simple way to construct a convergent primal sequence by averaging all past primal information in an ergodic sum. The results below, relating primal weights $\omega_{p_i}^\ell$ from (3.6) with dual stepsizes λ_i , basically follow [SC96], [LPS99], [Kiw03], and are also valid in our dynamic setting.

Proposition 3.3.1 *Given primal weights $(\omega_{p_i}^\ell)_{i=0}^\ell$ as in (3.6) and dual stepsizes $(\lambda_i)_{i=0}^\ell$, let the corresponding primal-dual weights be defined as*

$$\omega_{pd_i}^\ell := \frac{\omega_{p_i}^\ell}{\lambda_i} \text{ for } \ell \geq 0, i = 0, 1, \dots, \ell.$$

The following holds for any sequence $\{y^\ell\}$ converging to $y^* \in \mathbb{R}^n$:

(i) If for any $i_0 \geq 0$, $\lim_{\ell \rightarrow \infty, \ell > i_0} \omega_{p_i}^\ell = 0$, then

$$\lim_{\ell \rightarrow \infty, \ell > i_0} \left(\sum_{i=i_0}^{\ell} \omega_{p_i}^\ell y^\ell \right) = y^*.$$

(ii) If

$$\left. \begin{array}{l} \omega_{pd_i}^\ell \geq \omega_{pd_{i-1}}^{\ell-1} \text{ for all } \ell \text{ and } i \text{ such that } \ell \geq i > 0 \\ \omega_{pd_0}^\ell \rightarrow 0 \text{ and } \max_{i=1, \dots, \ell} (\omega_{pd_i}^\ell - \omega_{pd_{i-1}}^{\ell-1}) \rightarrow 0 \text{ as } \ell \rightarrow \infty \end{array} \right\} \quad (3.7)$$

then

$$\omega_{pd_i}^\ell \rightarrow 0 \text{ and } \omega_{p_i}^\ell \rightarrow 0 \text{ as } \ell \rightarrow \infty \text{ for all } i = 0, \dots, \ell.$$

(iii) If, in addition to (3.7),

$$\exists \bar{\omega}_{pd} > 0 \text{ such that } \omega_{pd_i}^\ell < \bar{\omega}_{pd} \text{ for all } \ell \geq 0, \quad (3.8)$$

then, for any $i_0 \geq 0$,

$$\sum_{i=i_0}^{\ell} \omega_{pd_i}^\ell (y^{i+1} - y^i) \rightarrow 0 \text{ as } \ell \rightarrow \infty.$$

Proof. Item (i) is known as Silverman-Toeplitz's theorem in e.g. [DS88].

To show item (ii), let first show recursively that $\omega_{\text{pd}_i^\ell} \rightarrow 0$ as $\ell \rightarrow \infty$. By (3.7), the relation holds for $i = 0$: $\omega_{\text{pd}_0^\ell} \rightarrow 0$. Suppose now that the relation holds for some $i \geq 0$. Then, by (3.7), $(\omega_{\text{pd}_{i+1}^\ell} - \omega_{\text{pd}_i^\ell}) \rightarrow 0$ as $\ell \rightarrow \infty$. Hence, we get $\omega_{\text{pd}_{i+1}^\ell} \rightarrow 0$ for any $i \geq 0$. Now, since $\omega_{\text{pd}_{i,\ell}} = \frac{\omega_{\text{p}_{i,\ell}}}{\lambda_i}$, we deduce that $\omega_{\text{p}_i^\ell} \rightarrow 0$ as $\ell \rightarrow \infty$, for any $i \geq 0$, as desired.

For the proof of the final item, we refer to [SC96, Thm 1]. \square

The primal convex weights sequence from (3.6) can be considered as an extension of the traditional ergodic sequence, introduced by [Sho85], where

$$\omega_{\text{p}_i^{\ell, [\text{Sho85}]}} := \frac{\lambda_i}{\sum_{j=0}^{\ell} \lambda_j} \quad \text{and} \quad \omega_{\text{pd}_i^{\ell, [\text{Sho85}]}} = \frac{1}{\sum_{j=0}^{\ell} \lambda_j}.$$

In this case, assumptions (3.7) and (3.8) hold trivially. The interest of considering a broader setting is that it is possible to use for example $\omega_{\text{p}_i^\ell} := \frac{1}{\ell}$, a choice that gives more weight to more recent subgradients.

Dual intermediate results

Associated with the primal weights of (3.6), we define the following ergodic objects, relevant for the dual block:

$$\begin{aligned} \hat{s}^\ell &:= - \sum_{i=0}^{\ell} \omega_{\text{p}_i^\ell} g(p^i) && \text{(ergodic subgradient)} \\ \hat{\nu}^\ell &:= \sum_{i=0}^{\ell} \omega_{\text{p}_i^\ell} \nu^i && \text{(ergodic normal element)} \\ \bar{f}_\ell(x) &:= \sum_{i=0}^{\ell} \omega_{\text{p}_i^\ell} (f(x^i) - \langle g(p^i), x - x^i \rangle) && \text{(ergodic model for } f) \end{aligned}$$

We now give some basic relations for various dual objects.

Proposition 3.3.2 *With the definitions above, the following holds.*

- (i) *The sequences $\{\hat{s}^\ell\}$ and $\{\hat{\nu}^\ell\}$ are bounded.*
- (ii) *The affine model \bar{f}_ℓ is a lower approximation for f , and*

$$\bar{f}_\ell(x) = \sum_{i=0}^{\ell} \omega_{\text{p}_i^\ell} C(p^i) + \langle \hat{s}^\ell, x \rangle. \quad (3.9)$$

- (iii) *For any $x \geq 0$ such that $x_{L \setminus J_{\ell+1}} = 0$*

$$|x^{\ell+1} - x|^2 \leq |x^\ell - x|^2 + |\lambda_\ell g^\ell|^2 + 2\lambda_\ell \langle g^\ell, x^\ell - x \rangle. \quad (3.10)$$

If, in addition $f(x) < f(x^\ell)$,

$$|x^{\ell+1} - x|^2 < |x^\ell - x|^2 + \lambda_\ell^2 |G|^2 \quad (3.11)$$

where the vector G has components $G_j = \max_{p \in \mathcal{Q}} |g_j(p)|$, for $j = 1, \dots, n$.

Proof. The sequence $\{g(p^i)\}$ is bounded, because $\text{conv } Q$ is assumed to be bounded by (3.2). From (3.5), for $j \in J_i$, $|\nu_j^i| = |\min(0, \frac{1}{\lambda_i} x_j^i + g_j(p^i))| \leq |g_j(p^i)|$, since $x_j^i \geq 0$. For $j \notin J_i$, $\nu_j^i = g_j(p^i)$. Hence $|\nu_j^i| \leq G_j$ for any $j \in L$. Therefore, the sequence $\{\nu^i\}$ is bounded. The sequences $\{\hat{s}^\ell\}$ and $\{\hat{\nu}^\ell\}$ are obtained as convex combinations of sequences $\{g(p^i)\}$ and $\{\nu^i\}$, respectively. Therefore, $|\hat{s}_j^\ell| \leq G_j$ and $|\hat{\nu}_j^\ell| \leq G_j$ for any $j \in L$.

Item (ii) follows from the definition of the ergodic model, recalling that $-g(p^i) \in \partial f(x^i)$. Finally, the first inequality in item (iii) is well-known in the static setting and the proof is essentially the same here. However, the result holds only for $x \geq 0$ satisfying $x_{L \setminus J_{\ell+1}} = 0$.

More specifically, in the expansion below, the nonexpansivity of the projection yields the relations

$$\begin{aligned} |x^{\ell+1} - x|^2 &= \left| \text{proj}_{J_{\ell+1}} \left(x^\ell + \lambda_\ell g^\ell - x \right) \right|^2 \\ &\leq |x^\ell + \lambda_\ell g^\ell - x|^2 \\ &= |x^\ell - x|^2 + |\lambda_\ell g^\ell|^2 + 2\lambda_\ell \langle g^\ell, x^\ell - x \rangle. \end{aligned}$$

Finally, (3.11) is straightforward, using the subgradient inequality to show that the last right hand side term in (3.10) is nonpositive. \square

We now state a dynamic version of results in [SC96] and [LPS99].

Lemma 3.3.3 *Consider a dynamic subgradient setting with dual and primal blocks as in § 3.3.1. Suppose the primal weights sequence $\{\omega_{p_i^\ell}\}$ satisfies (3.6), and the dual stepsizes sequence is vanishing: $\lambda_\ell \rightarrow 0$ as $\ell \rightarrow \infty$. Assume that (3.8) holds for the associated primal–dual weights $\{\omega_{pd_i^\ell}\}$. If there is some x^∞ , not necessarily optimal, such that $x^\ell \rightarrow x^\infty$, then*

$$\hat{s}^\ell + \hat{\nu}^\ell \rightarrow 0, \quad (3.12a)$$

$$\langle \hat{\nu}^\ell, x^\ell \rangle \rightarrow 0, \quad (3.12b)$$

$$\bar{f}_\ell(x^\ell) \rightarrow f(x^\infty). \quad (3.12c)$$

as $\ell \rightarrow \infty$.

Proof. The relation in (3.12a) follows from observing that $\hat{s}^\ell + \hat{\nu}^\ell = \sum_{i=0}^{\ell} \omega_{pd_i^\ell} (x^i - x^{i+1})$ and applying item (iii) in Proposition 3.3.1. We denote the set of asymptotically active indices $J_\infty := \{j \in L : x_j^\infty > 0\}$.

To show (3.12b) note that, since $x^\ell \rightarrow x^\infty$, there exists some iteration ℓ' such that $x_{J_\infty} > 0$ for any $\ell > \ell'$. As a result, for any $\ell \geq \ell'$, $\nu_{J_\infty}^\ell = 0$. Hence, by item (i) in Proposition 3.3.1, $\hat{\nu}_j^\ell \rightarrow 0$. Furthermore, for $j \in L \setminus J_\infty$ $x_j^\ell \rightarrow 0$, and $\{\hat{\nu}_j^\ell\}$ is bounded. Hence, (3.12b) follows.

To show (3.12c), we start by splitting the sum defining the ergodic model in three terms, with $0 < k < \ell$:

$$\begin{aligned} \bar{f}_\ell(x^\infty) &= \sum_{i=0}^{\ell} \omega_{p_i^\ell} \left(f(x^i) - \langle g(p^i), x^\infty - x^i \rangle \right) \\ &= \sum_{i=0}^{\ell} \omega_{p_i^\ell} f(x^i) - \sum_{i=0}^k \omega_{p_i^\ell} \langle g(p^i), x^\infty - x^i \rangle - \sum_{i=k}^{\ell} \omega_{p_i^\ell} \langle g(p^i), x^\infty - x^i \rangle. \end{aligned}$$

As $\ell \rightarrow \infty$, from item (i) in Proposition 3.3.1, we see that the first right hand side term goes to $f(x^\infty)$ while the second term goes to 0. As for the third term, it can be bounded below using (3.10):

$$-\sum_{i=k}^{\ell} \omega_{p_i^\ell} \langle g(p^i), x^\infty - x^i \rangle \geq \frac{1}{2} \sum_{i=k}^{\ell} \omega_{pd_i^\ell} (|x^{i+1} - x^\infty|^2 - |x^i - x^\infty|^2) - \frac{1}{2} \sum_{i=k}^{\ell} \omega_{p_i^\ell} \lambda_i G^2.$$

In this expression, both right hand side terms go to 0, by items (iii) and (i) in Proposition 3.3.1. It follows that $\lim \bar{f}_\ell(x^\infty) \geq f(x^\infty)$, which together with item (i) yields that $\bar{f}_\ell(x^\infty) \rightarrow f(x^\infty)$. Finally, since $\bar{f}_\ell(x^\ell) = \bar{f}_\ell(x^\infty) + \langle \hat{s}^\ell, x^\ell - x^\infty \rangle$ and $\langle \hat{s}^\ell, x^\ell - x^\infty \rangle \rightarrow 0$, we obtain that $\bar{f}_\ell(x^\ell) \rightarrow f(x^\infty)$, as desired. \square

All the results stated so far depend only on the dual and primal blocks in the dynamic setting. In particular, (3.12c), together with (3.9), and the concavity of C in (3.2), implies that $C(\hat{\pi}^{acc}) \geq f(x^\infty)$ for any accumulation point $\hat{\pi}^{acc}$ of $\{\hat{\pi}^\ell\}$. If furthermore this accumulation point was feasible for the convexified primal problem $(\mathcal{P})_{conv}$, by weak duality we would obtain an optimal primal–dual pair solving $((\mathcal{P})_{conv}, (\mathcal{D}))$. For this purpose, assumptions on the Constraints Management block and its separation procedure will be of central importance.

3.3.2 Choosing index sets

In the dynamic scheme of Fig. 3.1, primal and dual phases are coordinated by the constraints management block, that we detail below.

In order to define the next working set, the constraints management block finds new constraints to dualize and drops some indices from the current working set. For the first aim, we assume that a *Separation Oracle*, detecting violated inequalities not already included in the working index set, is available. Specifically, a Separation Oracle is a procedure $SepOr$ that, given a primal point $p \in \mathbb{R}^{m_p}$, identifies inequalities violated by p , i.e., some index j such that $g_j(p) > 0$ for j in a subset of L . The output of the procedure is a set of indices which can be empty only if no inequality is violated. In Combinatorial Optimization, this corresponds to having an *exact separation* algorithm for the family of inequalities defined by the hard constraints.

With these elements, the third block of the dynamic setting has the following structure.

Constraints Management block: Having the new convex primal point $\hat{\pi}^{\ell+1}$ and the working index set, calls the separation procedure to find indices to be included in the next working set:

$$I_\ell \subset SepOr(\hat{\pi}^{\ell+1}).$$

Having the new dual information, possibly drops inactive constraints, defining the set of removed indices O_ℓ .

The new working set is defined as $J_{\ell+1} := (J_\ell \setminus O_\ell) \cup (I_\ell \setminus J_\ell)$.

For the dynamic scheme to be convergent, each part of the constraints management block must satisfy some assumptions that will be referred to as “smart oracles” and “good steps”.

Smart oracles

The “maximum violation oracle” is a standard example of separation oracle. It defines $SepOr(p)$ as a subset of

$$\left\{ j \in L : g_j(p) = \max_{l \in L} \{g_l(p) > 0\} \right\}.$$

Depending on the application, other separating procedures may be preferable. In many cases, the particular structure of \mathcal{Q} and g can be exploited so that the Separation Oracle reduces to solving a simple optimization problem. Nevertheless, there are families of inequalities for which no efficient exact separation procedure is known so far. For this reason, along the lines of [BS08], we make the following minimal assumptions on the separation oracle:

given a fixed number $\beta > 0$ and the working set J , $SepOr(p) \subseteq J \implies$ for all $l \in L \setminus J$, $\left\{ \begin{array}{l} \text{either } g_l(p) \leq 0 \\ \text{or } g_l(p) > 0 \text{ but there is } j \in J : g_l(p) \leq \beta g_j(p). \end{array} \right.$	(3.13)
---	--------

The maximum violation oracle satisfies this assumption with $\beta = 1$, but other choices are also possible. In fact, *any violated inequality* can be separated. Moreover, if many inequalities are returned by the Separation Oracle, any nonempty *subset* of such inequalities can be used. This is a feature of practical interest when using heuristic approaches, either because no exact separation procedure is available, or because there are inequalities that appear as “more promising” for some reason, intrinsic to the nature of the problem. For instance, any efficient approximation algorithm for the maximum violation oracle problem could be used to define a separation oracle that satisfies (3.13). Finally, even if an exact separation oracle is available, it may still be interesting to start the process with some fast heuristic, to increase the overall efficiency of the method.

Removal at Good steps

In nonsmooth optimization, subgradients methods are of the nonmonotone type, the sequence of objective values $\{f(x^\ell)\}$ is not decreasing. We are interested in a subsequence with decreasing objective values, defined by the so-called *record points*, for which we use the notation $f_{rec}^k := f(x_{rec}^k)$, with $k = k(\ell)$. To each iteration ℓ corresponds a record point $x_{rec}^{k(\ell)}$ with the current best value function, generated at some past iteration.

Asymptotically, constraints will be dropped from the index set only for a subsequence of record points, called of *good steps*. Good steps are declared according to the following rule, depending on a positive real parameter η_k

$$\text{If } \begin{cases} f(x^{\ell+1}) < f_{rec}^{k(\ell)} \\ \text{and } I_\ell \subseteq J_\ell \\ \text{and } \max_{J_\ell} \{\hat{\nu}_j^\ell\} \leq \eta_k \end{cases} ; \text{ then } \ell \text{ is a good step, denoted } \ell \in \mathcal{L}_g . \quad (3.14)$$

Remark 3.3.4 *In a static setting, the last two conditions in (3.14) are always satisfied. Indeed, if $J_\ell \equiv L$, from (3.5) $\nu_j^i \leq 0$ for all $j \in L$, so $\hat{\nu}_j^\ell \leq 0$ for all $j \in L$. Hence, in the static case, good steps are nothing but record points.*

Giving a nonnegative integer ℓ_{rem} , the set of removed indices is defined as follows:

$$O_\ell = \begin{cases} \{j \in J_\ell : x_j^{\ell+1} = 0\} & \text{if } \ell \text{ is a good step or } \ell \leq \ell_{rem} \\ \emptyset & \text{otherwise.} \end{cases} \quad (3.15)$$

The parameter ℓ_{rem} makes the algorithm highly flexible. At initial iterations, i.e., far from a solution, constraints are always dropped from the working index set. Subsequently, at later iterations, the algorithm becomes more cautious about constraints removals, allowing them only at special iterations, deemed good enough. Similar conditions are required in the dynamic bundle method in [BS08], where constraints are dropped only at the so-called serious steps.

With this setting, there are two mutually exclusive situations: either the sequence of good steps is infinite, or there is a last good step followed by infinitely many “bad” steps. In the latter case, (3.15) eventually blocks any removal, and the index set stabilizes from some iteration on. The former case, by contrast, yields an infinite subsequence of iterates $\ell_g \in \mathcal{L}_g$ for which $SepOr(\hat{\pi}^{\ell_g+1}) \subseteq J_{\ell_g}$ and, thus, (3.13) holds.

We now show that in both situations the dynamic algorithm converges to a primal-dual pair solving the convexified primal problem $(\mathcal{P})_{conv}$ from (3.4) and its dual (\mathcal{D}) , given in (3.3).

3.3.3 Convergence results

We start by analyzing convergence for an asymptotically static algorithm, with constraints management set that eventually blocks all removals from the index sets.

Theorem 3.3.5 (Stabilized index sets) *Consider a dynamic subgradient setting with dual and primal blocks as in § 3.3.1, satisfying the assumptions in Lemma 3.3.3, and with dual stepsizes sequence such that*

$$\sum_{\ell=1}^{\infty} \lambda_{\ell} = +\infty \quad \text{and} \quad \sum_{\ell=1}^{\infty} \lambda_{\ell}^2 < +\infty. \quad (3.16)$$

Let the constraints management block be such that eventually all removals are blocked: $O_{\ell} = \emptyset$ for ℓ sufficiently large. If the separation oracle satisfies (3.13), then any cluster point of the primal sequence $\{\hat{\pi}^{\ell}\}$ solves problem (3.4), while the dual sequence converges to a solution to (3.3).

Proof. By the assumption that removals are blocked, the working indices stabilize at a fixed set after certain iteration $last$, and the *static* subgradient methods' convergence theory applies. In particular, by (3.11) and (3.16), the sequence $\{x^{\ell}\}$ is bounded. Moreover, the sequence converges to a solution x^{∞} to the restricted dual problem:

$$(\mathcal{D})_{J_{stab}} : \begin{cases} \min f(x) \\ x \geq 0 \\ x_{L \setminus J_{stab}} = 0. \end{cases}$$

By Lemma (3.3.3), as $\ell \rightarrow \infty$ the relations $f_{\ell}(x^{\ell}) \rightarrow f(x^{\infty})$, $\hat{s}^{\ell} + \hat{\nu}^{\ell} \rightarrow 0$ and $\langle \hat{\nu}^{\ell}, x^{\ell} \rangle \rightarrow 0$ hold. Together with (3.9) and the concavity of C in (3.2), we have that

$$C(\hat{\pi}^{\ell}) \geq \bar{f}_{\ell}(x^{\ell}) - \langle \hat{s}^{\ell} + \hat{\nu}^{\ell}, x^{\ell} \rangle + \langle \hat{\nu}^{\ell}, x^{\ell} \rangle. \quad (3.17)$$

Let $\hat{\pi}^{acc}$ be an accumulation point of the (bounded) sequence $\{\hat{\pi}^{\ell}\}$, and consider the subsequence $\{\hat{\pi}^{\ell}\}_{\ell \in \mathcal{L}}$ converging to $\hat{\pi}^{acc}$. Passing to the limit in (3.17) as $\mathcal{L} \ni \ell \rightarrow \infty$ gives the relation

$$C(\hat{\pi}^{acc}) \geq f(x^{\infty}). \quad (3.18)$$

We now show that $\hat{\pi}^{acc}$ satisfies $g_j(\hat{\pi}^{acc}) \leq 0$ for any $j \in L$. Consider first $j \in J_{stab}$: for any $\ell \geq last$ we have $\nu_j^{\ell} \leq 0$, so by Proposition (3.3.1), $g_j(\hat{\pi}^{acc}) \leq \limsup(-\hat{s}^{\ell}) = \limsup \hat{\nu}_j^{\ell} \leq 0$. Now, for $j \in L \setminus J_{stab}$, since $J_{\ell+1} = J_{stab}$ for $\ell \geq last$, and $J_{\ell+1} = J_{\ell} \setminus O_{\ell} \cup (I_{\ell} \setminus J_{\ell})$ with $O_{\ell} = \emptyset$, eventually the inclusion $I_{\ell} \subset J_{\ell}$ holds. With assumption (3.13), this gives that $g_j(\hat{\pi}^{acc}) \leq 0$ for $j \notin J_{stab}$. Hence, any ergodic primal cluster point $\hat{\pi}^{\infty}$ is also feasible for the convexified primal problem.

As in [[BS08], Thm 6], with (3.18) primal feasibility ensures by weak duality that x^{∞} is a solution to (3.3) while $\hat{\pi}^{acc}$ is a solution to the convexified primal problem (3.4). \square

Theorem 3.3.5 gives in a straightforward manner the following two results.

Corollary 3.3.6 (Finite number of good steps) *Consider a dynamic subgradient setting with dual and primal blocks as in § 3.3.1, satisfying the assumptions in Lemma 3.3.3, and with dual stepsizes sequence satisfying (3.16). Let the constraints management step be given by (3.15). Suppose that the separation oracle satisfies (3.13). Assume there is a last iteration index after which there are no more good steps:*

$$\hat{x}^{k(\ell)} = \hat{x}^{k(last)} \quad \text{for all } \ell \geq last.$$

Then any cluster point of the primal sequence $\{\hat{\pi}^{\ell}\}$ solves problem (3.4), while the dual sequence converges to a solution to problem (3.3). Furthermore, $\hat{x}^{k(last)}$ is also a solution of (3.3).

Proof. Since no good step is declared after iteration ℓ_{ast} , by (3.15), the constraints management rule eventually stops removing indices from the index set which stabilizes to J_{stab} , and Lemma 3.3.5 applies. Now, for contradiction purposes, suppose that $\hat{x}^{k(\ell_{ast})}$ is not solution to (3.3). Then, since x^ℓ converges to a solution to (3.3), after some iteration $\ell_1 > \ell_{ast}$, it holds that $f(x^\ell) < f(\hat{x}^{k(\ell_{ast})})$ for all $\ell \geq \ell_1$. Furthermore, once the index set stabilizes we have $I_\ell \subset J_\ell$. Finally, reasoning like in the theorem above, for any $j \in J_{stab}$, $\limsup \nu_j^\ell \leq 0$. As a consequence, a good step should occur for some $\ell_2 > \ell_{ast}$, contradicting our assumption. Hence, $x^{k(\ell_{ast})}$ must solve problem (3.3). \square

When considering a convex program (\mathcal{P}), the next corollary can be seen as a unification of the previous results in [SC96], [LPS99] and [AW07] for concave objective function, convex constraints and with general rules for the definition of the ergodic sequence.

Corollary 3.3.7 (Primal recovery in the static setting) *Consider a subgradient setting with dual and primal blocks as in § 3.3.1, satisfying the assumptions in Lemma 3.3.3, and with dual stepsizes sequence satisfying (3.16). Let the constraints managements step set $J_\ell = L$ at every iteration. If the separation oracle satisfies (3.13), then any cluster point of the primal sequence $\{\hat{\pi}^\ell\}$ solves problem (3.4), while the dual sequence converges to a solution to (3.3).* \square

Corollary 3.3.6 puts in evidence an interesting issue: as long as removals are eventually blocked, it does not really matter what defines a “good” step, except for the fact that it is a record point. At first sight, then, the last two relations in (3.14) could be deemed superfluous. We now show that these relations are in fact fundamental for showing convergence when infinitely many good steps are generated by the dynamic algorithm.

We start with an intermediate result showing convergence of the sequence of dual steps (not necessarily to a solution) in a general context.

Lemma 3.3.8 *Consider a dynamic subgradient scheme where constraints may be dropped only at record points, and with stepsizes satisfying (3.16). Then the corresponding dual sequence $\{x^\ell\}$ converges to some point x^∞ .*

Proof. Note first that infinite good steps imply that the the records sequence $\{x_{rec}^k\}$ is infinite too. We claim that this sequence is bounded. By Prop. IV.3.2.5 in [HUL96], sub-level sets of a convex function f are compact if and only if f is 0-coercive, which is implied in our case by the Slater condition on the primal problem. Let f^∞ be the limiting value of the strictly decreasing real sequence $\{f_{rec}^\ell\}$, and let \mathcal{I}^∞ the the corresponding iso-level set of f : $\mathcal{I}^\infty := \{x \in \mathbb{R}^n : f(x) = f^\infty\}$, which is nonempty and compact. Since the records subsequence $\{x_{rec}^k\}$ satisfies that $d(x_{rec}^k, \mathcal{I}^\infty) \rightarrow 0$, our claim follows. We now show that the whole dual sequence is bounded, by examining the length of each k -cycle, i.e., of iterations between consecutive records x_{rec}^k and x_{rec}^{k+1} . More precisely, we show that the maximal length Σ_k of a cycle k tends to 0 as $k \rightarrow \infty$. Clearly, $\Sigma_k^2 \leq \max_{k(\ell+1)=k} |x^{\ell+1} - x_{rec}^{k(\ell)}|^2$. For each cycle k , its first point is $x_{rec}^{k(\ell)}$, which is also the best function value generated so far. After this point is found, with our assumption no constraint is dropped during the cycle. As a result, for all iterations in the cycle, i.e. for ℓ such that $k(\ell) = k$, we have that $x_{rec}^k \geq 0$ with $(x_{rec}^k)_{L \setminus J_{\ell+1}} = 0$. By (3.11), for any ℓ such that $k(\ell) = k$ the relations below hold:

$$\begin{aligned} |x^{\ell+1} - x_{rec}^{k(\ell)}|^2 &\leq |x^\ell - x_{rec}^k|^2 + (\lambda_\ell G)^2, \\ \text{so, } \Sigma_k^2 &\leq \sum_{\ell: k(\ell)=k} (\lambda_\ell G)^2. \end{aligned}$$

Therefore, by (3.16), the cycle length's $\Sigma_k \rightarrow 0$ as $k \rightarrow \infty$, as desired. Consider now the sequence $\{x^\ell\}$, and notice that

$$\begin{aligned} d(x^\ell, \mathcal{I}^\infty) &\leq d(x^\ell, x_{rec}^{k(\ell)}) + d(x_{rec}^{k(\ell)}, \mathcal{I}^\infty) \\ &\leq \Sigma_{k(\ell)} + d(x_{rec}^{k(\ell)}, \mathcal{I}^\infty). \end{aligned}$$

Since as $\ell \rightarrow \infty$, also $k(\ell) \rightarrow \infty$, the results above ensure that both right hand side terms converge to 0. We conclude that the whole sequence is bounded and verifies $d(x^\ell, \mathcal{I}^\infty) \rightarrow 0$, and $f(x^\ell) \rightarrow f^\infty$ as $\ell \rightarrow \infty$.

Let x^{acc} be a cluster point of $\{x^\ell\}$:

$$\forall \epsilon > 0, \forall \ell_0 \geq 0, \exists \ell \geq \ell_0 : |x^\ell - x^{acc}| < \epsilon.$$

By construction, $f(x^{acc}) = f^\infty \leq f(x_\ell)$ for any ℓ . We denote by J_{acc} the set of active indices of x^{acc} and by $\rho := \min\{x_i^{acc} : i \in J_{acc}\}$. We observe the relation $J_{acc} \cap O_\ell \neq \emptyset$ is only possible if $|x^\ell - x^{acc}|^2 > \rho$. Now, given any $\epsilon > 0$ such that $\epsilon < \rho$, by (3.16), there exists ℓ_0 such that $\sum_{i=\ell_0}^\infty \lambda_i^2 < \frac{\epsilon}{2}$. Moreover, as x^{acc} is a cluster point, we can pick an iteration $\ell_1 \geq \ell_0$ such that $|x_{\ell_1} - x^{acc}| < \frac{\epsilon}{2}$. At that iteration, it holds that $J_{\ell_1} \supset J_{acc}$, and the inclusion $J_\ell \supset J_{acc}$ for $\ell \geq \ell_1$ as long as no index is dropped from J_{acc} . In other words, as long as $J_{acc} \cap O_\ell \neq \emptyset$. For contradiction purposes, suppose there is some iteration $\ell_2 > \ell_1$ such that $J_{acc} \cap O_{\ell_2} \neq \emptyset$. Then, for any $\ell \in \{\ell_1, \dots, \ell_2\}$ we have that $J_{acc} \subset J_\ell$, and inequality (3.11) holds with $x = x^{acc}$:

$$\begin{aligned} |x^{\ell_2} - x^{acc}|^2 &\leq |x^{\ell_2-1} - x^{acc}|^2 + (\lambda_\ell G)^2 \\ &\leq |x^{\ell_1} - x^{acc}|^2 + \sum_{i=\ell_1}^\infty \lambda_i^2 \\ &< \epsilon. \end{aligned}$$

It follows that, in order to drop any index in J_{acc} at an iteration ℓ_2 , the inequality $|x^{\ell_2} - x^{acc}|^2 > \rho$ must hold. The contradiction follows and we conclude that at record points, i.e., at ℓ such that $x^{\ell+1} = x_{rec}^{k(\ell+1)}$, it must hold that $O_\ell \cap J_{acc} = \emptyset$. It follows that the above inequality is true for any $\ell \geq \ell_1$, i.e., $|x^\ell - x^{acc}|^2 < \epsilon$. Summing up, we have shown that

$$\forall \epsilon > 0, \exists \ell_0 \text{ such that } |x^\ell - x^{acc}|^2 < \epsilon \text{ for all } \ell \geq \ell_0,$$

and the dual sequence converges, as stated. \square

Theorem 3.3.9 *Consider a dynamic subgradient setting with dual and primal blocks as in § 3.3.1, satisfying the assumptions in Lemma 3.3.3, and with dual stepsizes sequence satisfying (3.16). Let the constraints management step be given by (3.15). Suppose that the separation oracle satisfies (3.13). If, in addition, the parameter sequence $\{\eta_k\}$ in (3.14) goes to 0 as $k \rightarrow \infty$ and there is an infinite number of good steps, then any cluster point of the primal subsequence of good steps $\{\hat{\pi}^{\ell_g+1}\}_{\ell_g \in \mathcal{L}_g}$ solves problem (3.4), while the dual sequence converges to a solution to (3.3).*

Proof. The constraints management step in (3.15) is more restrictive than the one considered in Lemma 3.3.8. Therefore, the results therein hold and, hence, there is an x^∞ such that $x^\ell \rightarrow x^\infty$ as $\ell \rightarrow \infty$. It follows that Lemma 3.3.3 also applies. Moreover, since $\text{conv}(Q)$ is compact, $\{\hat{\pi}^\ell\}_{\ell \in \mathcal{L}_g}$ has cluster points in $\text{conv}(Q)$. Let $\hat{\pi}^{acc}$ be an accumulation point, and consider the subsequence $\{\hat{\pi}^\ell\}_{\ell \in \mathcal{L}_{acc} \subset \mathcal{L}_g}$ converging to $\hat{\pi}^{acc}$. By (3.12c), $f(x^\ell) \rightarrow f(x^\infty)$ as $\ell \rightarrow \infty$ in \mathcal{L}_{acc} . Together with (3.9) and the concavity of C , we have that

$$C(\hat{\pi}^\ell) \geq \bar{f}_\ell(x^\ell) + \langle g(\hat{\pi}^\ell) - \hat{\nu}^\ell, x^\ell \rangle + \langle \hat{\nu}^\ell, x^\ell \rangle.$$

Passing to the limit as $\mathcal{L}_{acc} \ni \ell \rightarrow \infty$ gives the relation $C(\hat{\pi}^{acc}) \geq f(x^\infty)$. By weak duality, to show that $\hat{\pi}^{acc}$ is a solution to (3.4), we only need to check that $g(\hat{\pi}^{acc}) \leq 0$. The third inequality in (3.14) and our assumption on the parameter sequence $\{\eta_k\}$ ensure that $\lim_{\mathcal{L}_{acc} \ni \ell \rightarrow \infty} |(\hat{\nu}^\ell)_{J_\ell}| = 0$. But, since g is either an affine or a convex function by (3.2), $g(\hat{\pi}^\ell) \leq -\hat{s}^\ell$, with $\hat{s}^\ell + \hat{\nu}^\ell \rightarrow 0$ by (3.12a). It follows that

$$\lim_{\mathcal{L}_{acc} \ni \ell \rightarrow \infty} (g(\hat{\pi}^\ell))_{J_\ell} \leq 0.$$

Finally, by (3.14), for all $\ell \in \mathcal{L}_g$ the inclusion $I_\ell \subset J_\ell$ holds, and assumption (3.13) on the separation procedure gives that for all $\ell \in \mathcal{L}_{acc} \subset \mathcal{L}_g$,

$$\max_{i \in L \setminus J_\ell} \{g_i(\hat{\pi}^\ell)\} \leq \beta \max_{i \in J_\ell} \{g_i(\hat{\pi}^\ell)\}.$$

Hence, any cluster point $\hat{\pi}^{acc}$ of the ergodic primal sequence $\{\hat{\pi}^\ell\}_{\ell \in \mathcal{L}}$ is feasible for the convexified primal problem (3.4).

As in [[BS08], Thm 6], primal feasibility ensures by weak duality that that x^∞ is a solution to dual problem (3.3) while $\hat{\pi}^{acc}$ is a solution to (3.4). \square

Concluding Remarks

We have shown convergence of dynamic subgradient algorithms for a rather general setting. More specifically, we have shown primal-dual convergence for dual and primal blocks as in § 3.3.1, and constraints management step given by (3.14)-(3.15), under standard assumptions. These assumptions are:

- for the primal weights sequence $\{\omega_{p_i}^\ell\}$, satisfaction of (3.6);
- for the dual stepsizes sequence $\{\lambda_\ell\}$, positivity and (3.16);
- for the associated primal–dual weights $\{\omega_{pd_i}^\ell\}$, satisfaction of (3.7)-(3.8);
- and for the separation procedure, satisfaction of (3.13).

In particular, the traditional ergodic sequence, that takes $\omega_{p_i}^{\ell, [\text{Sho85}]} = \frac{\lambda_i}{\sum_{j=0}^{\ell} \lambda_j}$, gives primal-dual weights for which (3.7) and (3.8) trivially hold.

Our asymptotic results could be used to design sound stopping tests. For example, the method could stop at a good step, provided the duality gap is sufficiently small.

3.4 Dynamic Bundle Methods

As me mentioned in the introduction, more robust non-smooth optimization methods have been recently used for the dual step, in particular bundle methods [Hel04, FG99, FGRS06, FLR06]. A convergence proof is given in [Hel04] for an algorithm tailored to solve semidefinite relaxations of combinatorial problems, with a dual step that uses the so-called spectral bundle method. To choose subsets J , a maximum violation oracle detect the most violated inequality. Belloni and Sagastizábal [BS08] studied convergence properties of dynamic Lagrangian relaxation techniques with general bundle methods for the dual step and minimal assumptions on the separation procedure. For the sake of completeness, we recall here the main results.

We consider an identical primal problem defined as

$$(\mathcal{P}) : \begin{cases} \max_p C(p) \\ p \in \mathcal{Q} \subseteq \mathbb{R}^{m_p} \\ g_i(p) \leq 0, \quad i = 1, \dots, n, \end{cases} \quad (3.19)$$

where $C : \mathbb{R}^{m_p} \rightarrow \mathbb{R}$ is a concave function and for the hard constraints, the functions $g_i : \mathbb{R}^{m_p} \rightarrow \mathbb{R}$, $i = 1, \dots, n$, are affine. Easy constraints are included in the (possibly discrete) set, \mathcal{Q} , for which we assume that its convex hull $\text{conv } \mathcal{Q}$ is compact. Furthermore, for any $\bar{x} \geq 0$, we denote by $N_J(\bar{x})$ the normal cone to the set $\{x \geq 0 : x_{L \setminus J} = 0\}$ at point \bar{x} , defined as

$$N_J(\bar{x}) = \{\nu \in \mathbb{R}^n : \langle \nu, x - \bar{x} \rangle \leq 0 \text{ for } x \text{ such that } x_J \geq 0\}$$

3.4.1 Bundle iterates in a dynamic setting

Compared to the previous analysis regarding dynamic subgradients methods, only the dual step is modified, where an adapted proximal bundle method is involved. As presented in section 2.4.2, bundle methods can be seen as stabilized cutting plane algorithms. Past information computed by the primal oracle is kept in a bundle \mathcal{B}_ℓ .

$$\mathcal{B}_\ell := \{C(p^i), g(p^i)\}_{i \in \mathcal{B}_\ell}.$$

This information is used to build a model for the dual function f , with the cutting planes function

$$\check{f}_\ell(x) := \max_{i \in \mathcal{B}_\ell} \{C(p^i) - \langle g(p^i), x \rangle\}.$$

Bundle methods generate a sequence of trial points x^ℓ at which functions values and subgradients are evaluated. The subsequence of stability centers $\{\hat{x}^{k(\ell)}\}$ is chosen so that the corresponding objective functions are significantly decreasing. New dual variables are provided by the solution of a *restricted* quadratic program :

$$\begin{aligned} x^{\ell+1} &:= \arg \min_{x \geq 0} \check{f}_\ell(x) + \frac{1}{2} \mu_\ell |x - \hat{x}^{k(\ell)}|^2. \\ &\text{s.t. } x_{L \setminus J_\ell} = 0 \end{aligned} \quad (3.20)$$

Compared to the classical static instance of a bundle iteration, an additional constraint zeroing components out of J_ℓ is introduced. As a consequence, we have to solve a quadratic program of *reduced dimension*. After evaluating the dual function at $x^{\ell+1}$, this trial point will be declared new stability center if the descent test below is satisfied :

$$f(\hat{x}^{k(\ell)}) - \check{f}_\ell(x) \geq m \Delta_\ell,$$

where $\Delta_\ell := (f(\hat{x}^{k(\ell)}) - f(x^{\ell+1}))$ is the predicted descent. Whether or not $x^{\ell+1}$ is declared as new stability center, the cutting planes model is updated with the new element $(C(p^{\ell+1}), p^{\ell+1})$ after a possible compression of the current information.

Writing the optimality conditions for (3.20), we know that there are simplicial multipliers $\{\alpha_j\}_{j \in \mathcal{B}_\ell}$ and a normal element $\hat{\nu}^\ell \in N_{J_\ell}(x^\ell)$ such that

$$\begin{aligned} x^{\ell+1} &= \hat{x}^{k(\ell)} - \frac{1}{\mu_\ell} (\hat{s}^\ell + \hat{\nu}^\ell), \text{ where} \\ \hat{s}^\ell &= \sum_{j \in \mathcal{B}_\ell} \alpha_j g(p^j) \text{ is the aggregate subgradient} \\ &\text{and } \hat{\nu}^\ell \text{ is called aggregate normal element.} \end{aligned}$$

We define the aggregate primal element as $\hat{\pi}_\ell := \sum_{j \in \mathcal{B}_\ell} \alpha_j p^j$. Since functions g_i are affine, we have that $\hat{s}^{\ell+1} = g(\hat{\pi}_\ell)$. Moreover, we know that $\hat{s}^\ell \in \partial \check{f}_\ell(x^{\ell+1})$.

Usually, the algorithm is stopped when the predicted descent is smaller than a given tolerance.

3.4.2 Choosing index sets

The conditions on the constraint management block required to ensure primal-dual convergence of dynamic bundle methods are similar to those introduced for subgradients. As in the subgradient case, the separation procedure must satisfy the following

$$\boxed{\begin{array}{l} \text{given a fixed number } \beta > 0 \text{ and the working set } J, \\ \text{SepOr}(p) \subseteq J \implies \text{for all } l \in L \setminus J \\ \left\{ \begin{array}{l} \text{either } g_l(p) \leq 0 \\ \text{or } g_l(p) > 0 \text{ but there is } j \in J : g_l(p) \leq \beta g_j(p). \end{array} \right. \end{array}} \quad (3.21)$$

For removing inequalities, there is a slight difference between the two approaches. We recall that in the subgradient case, inactive constraints can be dropped from the index set only for a subsequence of record points, called of *good steps*, declared according to the following rule:

$$\text{If } \left\{ \begin{array}{l} f(x^{\ell+1}) < f_{rec}^{k(\ell)} \\ I_\ell \subseteq J_\ell \\ \max_{J_\ell} \{\hat{\nu}_j^\ell\} \leq \eta_k \end{array} \right. , \text{ then } \ell \text{ is a good step i.e. } \ell \in \mathcal{L}_g , \quad (3.22)$$

In the bundle setting, instead of the notion *good steps*, we will use the classical definition of *serious steps*. Here, primal weights are directly related to the dual directions computed by the quadratic subproblem. Hence, there is no need of the third condition defining good steps, since it is always satisfied. The removal condition is simpler and is defined by the following rule:

$$\text{If } \left\{ \begin{array}{l} f(x^{\ell+1}) < f(\hat{x}^{k(\ell)}) - m\Delta_\ell \\ I_\ell \subseteq J_\ell \end{array} \right. \text{ then } O_\ell = \{j \in J_\ell : x_j^\ell = 0\}, \text{ else } O_\ell = \emptyset. \quad (3.23)$$

3.4.3 Convergence results

Theorem 3.4.1 (Belloni & Sagastizabal, [BS08]) *Consider a dynamic bundle setting with a constraint management step satisfying (3.21) and (3.23), and suppose it loops forever. Suppose, in addition, that the sequence of proximal parameters $\{\mu_\ell\}$ satisfies at serious steps*

$$\mu_\ell \leq \mu_{max} \text{ and at null steps } \sum_{\ell: \ell, \ell+1 \text{ are null steps}} \frac{\mu_\ell}{\mu_{\ell+1}^2} = \infty$$

if infinitely many successive null steps occur. Then the primal dual sequence $(\hat{\pi}_\ell, x^\ell)$ has a cluster point (π^{acc}, x^{acc}) such that π^{acc} solves problem (3.4) while x^{acc} is a solution to (3.3).

3.5 Conclusion

Not surprisingly, the assumptions and convergence results are close to those for dynamic bundle methods of [BS08]. However, when using bundle methods, primal weights are a by-product of the quadratic programming problem defining the dual directions. There is no need of the third condition defining good steps, simply because the condition is always satisfied, by construction. The need of this third condition appears for the dynamic subgradients due to the discrepancy between the factors defining the dual direction (λ_ℓ) and those defining the primal point $(\omega_{p_i}^\ell)$.

An interesting extension could be to explore the relation of our dynamic scheme with Nesterov's primal-dual subgradient method presented in [Nes05]. In this approach, dual

stepsizes are related to a certain primal sequence, responsible for averaging past information. This setting provides additional flexibility for the primal reconstruction while avoiding the fact of having the newest primal information entering the ergodic sequence with vanishing weights ($\omega_{p_i}^\ell \rightarrow 0$ as $\ell \rightarrow \infty$). A similar framework is studied by Ruszczyński in [Rn07].

Bibliography

- [AW07] K. Anstreicher and L. Wolsey, *Two “well-known” properties of subgradient optimization*, Mathematical Programming (2007).
- [BC81] E. Balas and N. Christofides, *A restricted Lagrangian approach to the traveling salesman problem*, Math. Programming **21** (1981), no. 1, 19–46.
- [BdMV06] F. Babonneau, O du Merle, and J.-P. Vial, *Solving large-scale linear multicommodity flow problems with an active set strategy and proximal-accpm*, Operations Research **54** (2006), no. 1, 184–197.
- [BS08] A. Belloni and C. Sagastizábal, *Dynamic bundle methods*, Mathematical Programming (2008).
- [DS88] N. Dunford and J.T. Schwartz, *Linear operators, general theory*, vol. 1, Wiley, 1988.
- [DW60] G.B. Dantzig and P. Wolfe, *Decomposition principle for linear programs*, Operational Research **8** (1960).
- [EGM94] L. F. Escudero, M. Guignard, and K. Malik, *A Lagrangian relax-and-cut approach for the sequential ordering problem with precedence relationships*, Ann. Oper. Res. **50** (1994), 219–237, Applications of combinatorial optimization.
- [Erm76] Y. Ermoliev, *Methods of stochastic programming*, Optimization and Operations Research (1976).
- [FG99] A. Frangioni and G. Gallo, *A bundle type dual-ascent approach to linear multicommodity min cost flow problems*, INFORMS Journal on Computing **11** (1999), no. 4, 370–393.
- [FGRS06] I. Fischer, G. Gruber, F. Rendl, and R. Sotirov, *Computational experience with a bundle approach for semidefinite cutting plane relaxations of max-ut and equipartition*, Mathematical Programming **105** (2006), no. 2-3,B, 451–469.
- [FLR06] A. Frangioni, A. Lodi, and G. Rinaldi, *New approaches for optimizing over the semimetric polytope*, Mathematical Programming **104** (2006), no. 2-3,B, 375–388.
- [Gav85] B. Gavish, *Augmented lagrangean based algorithms for centralized network design*, IEEE Transactions on Communications **33** (1985), no. 12, 1247 – 1257.
- [GG61] P.C. Gilmore and R.E. Gomory, *A linear programming approach to the cutting-stock problem*, Operational Research **9** (1961).

- [Hel04] C. Helmsberg, *A cutting plane algorithm for large scale semidefinite relaxations, The sharpest cut*, MPS/SIAM Ser. Optim., SIAM, Philadelphia, PA, 2004, pp. 233–256.
- [HFK01] M. Hunting, U. Faigle, and W. Kern, *A Lagrangian relaxation approach to the edge-weighted clique problem*, European J. Oper. Res. **131** (2001), no. 1, 119–131.
- [HUL96] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex analysis and minimization algorithms*, Springer Verlag, Heidelberg, 1996, Two volumes - 2nd printing.
- [Kiw03] K. C. Kiwiel, *Convergence of approximate and incremental subgradient methods for convex optimization*, SIAM J. on Optimization **14** (2003), no. 3, 807–840.
- [LPS99] T. Larsson, M. Patriksson, and A. Strömberg, *Ergodic, primal convergence in dual subgradient schemes for convex programming*, Math. Program. **86** (1999), no. 2, Ser. A, 283–312.
- [LR01] C. Lemaréchal and A. Renaud, *A geometric study of duality gaps, with applications*, Mathematical Programming **90** (2001), no. 3, 399–427.
- [Luc92] A. Lucena, *Steiner problem in graphs : Lagrangean relaxation and cutting-planes*, COAL Bulletin **21** (1992).
- [Luc05] ———, *Non delayed relax-and-cut algorithms*, Ann. Oper. Res. **140** (2005), 375–410.
- [Lue01] M. Luebbecke, *Engine scheduling by column generation*, Ph.D. thesis, Universitaet Braunschweig, 2001.
- [Nes05] Y. Nesterov, *Primal-dual subgradient methods for convex problems*, CORE Discussion Paper (2005).
- [NO07] A. Nedić and A. Ozdaglar, *Approximate primal solutions and rate analysis for dual subgradient methods*, Tech. Report 2753, LIDS, 2007.
- [NY78] A.S. Nemirovskii and D.B. Yudin, *Cesaro convergence of gradient method approximation of saddle points for convex-concave functions*, Doklady Akademii Nauk SSSR **239** (1978).
- [Pol87] B.T. Polyak, *Introduction to optimization*, Optimization Software (1987).
- [Rn07] A. Ruszczyński, *A merit function approach to the subgradient method with averaging*, Optimization Methods and Software **23** (2007), no. 1.
- [SC96] H. D. Sherali and G. Choi, *Recovery of primal solutions when using subgradient optimization methods to solve lagrangian duals of linear programs*, Oper. Res. Lett. **19** (1996), no. 3, 105–113.
- [Sho85] N. Z. Shor, *Minimization methods for nondifferentiable functions*, Springer Series in Computational Mathematics, vol. 3, Springer-Verlag, Berlin, 1985.

Chapter 4

Inexact and Incremental Approaches

Sommaire

4.1 Inexact bundle methods with varying accuracy	62
4.1.1 Inexact linearizations with varying accuracy	62
4.1.2 Convergence analysis	67
4.2 An alternative approach when $E_{x^\ell}^g = 0$ and $E_{x^\ell}^f \rightarrow 0$	70
4.2.1 Inexact classical bundle method	70
4.2.2 Convergence analysis	72
4.3 Incremental bundle methods	74
4.3.1 Motivations	74
4.3.2 Related works	74
4.3.3 Incremental bundle methods	74
4.4 Mid term generation planning	76
4.4.1 Price decomposition on a scenario tree	76
4.4.2 Numerical illustrations	78
4.5 Conclusion	79
Bibliography	82

Generation planning of a huge power mix leads to large scale optimization problems. An efficient way to tackle such problem is to perform Lagrangian relaxation on the coupling constraint of demand. The resulting dual problem is separable, which eases the evaluation of the dual function. The dual problem is solved through sequential solution of local subproblems, restricted to each production unit. Due to the high number of units and the complexity of some local subproblems, the computational burden of evaluating sequentially all local subfunctions is preponderant in the whole iterative process.

In this context, we consider how to take full advantage of the dual separable structure. The idea is to make a dual iteration after having evaluated only a subset of the dual local subfunctions, instead of all of them. This type of incremental approach has already been applied for subgradient algorithms. In this work we use instead a *disaggregate bundle method*.

Such incremental bundle approach is related to bundle methods with inexact linearizations. In our framework however, inexactness varies along iterations. This case being not strictly covered by the existing literature, we analyse the related theory in the first section of this chapter. In the second section, we present our incremental strategy for bundle

methods and show how to interpret it as iterations of a bundle method with inexact linearizations. The chapter ends with an application of the approach to the French mid-term generation planning problem, with numerical results that confirm the potential of such a strategy.

A short version of this work was published in the Proceedings of EngOpt 2008 – International Conference on Engineering Optimization ([ES08]).

4.1 Inexact bundle methods with varying accuracy

We present here a convergence analysis of bundle methods with inexact linearizations where inaccuracies can vary along the iterative process. This case is slightly different from the one studied by Kiwiel ([Kiw06]) that was briefly presented in section 2.4.4. The motivation for this extended analysis is given in Section 4.2, where we are interested in convergence properties when inaccuracies vanish along the iterative process.

4.1.1 Inexact linearizations with varying accuracy

Recent developments of bundle methods have shown that those algorithms may handle inexact informations to achieve approximate optimality for non-smooth optimisation programs. Inaccuracies may affect the evaluation of the objective function, or the subgradient returned by the available oracle.

In [Kiw85, Kiw95], the author proposes an adapted bundle method to deal with an oracle providing inexact linearizations with a controllable accuracy that can be driven to 0. Unlike [Kiw85, Kiw95], [Hin01] proposes a methodology to deal with inexact subgradient evaluations without knowledge of the approximation quality. It requires however the exact evaluation of the objective function. This case is generalized in [Kiw06] where unknown inaccuracies may affect both the function value and the subgradient. This framework is further extended for constrained optimization in [KL07]. A different point of view is retained in [Sol03] : the author studies the precision required by the oracle to achieve optimality with a given tolerance using an almost standard bundle method.

We presented briefly in section 2.4.4 the methodology proposed in [Kiw06] to apply bundle methods when an inexact oracle is available. We show here that similar convergence results can be derived when inaccuracies are no longer constant, but depend on the evaluation point. Since our interest lies in separable problems where the objective is defined as a sum of subfunctions, we consider the following generic optimization problem:

$$(\mathcal{P}) : \left\{ \min_{x \in \mathbb{R}^n} (f(x) = (f^1 + f^2)(x)) \right\}, \quad (4.1)$$

where f^1 and f^2 are convex functions, possibly non-smooth. The function f is assumed to be coercive and we denote by f^{opt} the optimal value of problem (4.1) with $f^{opt} > -\infty$. This holds for example when (4.1) comes from lagrangian relaxation of a primal problem satisfying a Slater-type condition. The analysis below could be easily generalized to $Nu \geq 2$ subfunctions. For simplicity, we assume here that f^1 is easy to evaluate, while f^2 is hard to compute and time consuming. Hence, an exact oracle is available for f^1 , providing for any $y \in \mathbb{R}^n$ the values $f^1(y)$ and $g^1(y) \in \partial f^1(y)$. However, for f^2 , an inexact oracle only provides (f_y^2, g_y^2) with *unknown* but *bounded* accuracy:

$$f_y^2 \geq f^2(y) - E_y^f \text{ and} \quad (4.2)$$

$$f^2(\cdot) \geq f_y^2 + \langle g_y^2, \cdot - y \rangle - E_y^g, \quad (4.3)$$

such that:

$$E_x^f \leq E_{max}^f \text{ and } E_x^g \leq E_{max}^g. \quad (4.4)$$

In section 4.3.3 we interpret the incremental setting as an instance of this framework. Error is unknown, varying, but bounded by (4.4).

We notice that

$$\begin{aligned} g_y^2 &\in \partial_{E_y^f + E_y^g} f^2(y) \quad \text{and} \\ f_y^2 &\in [f(y) - E_y^f; f(y) + E_y^g]. \end{aligned}$$

Our framework is slightly different from [Kiw06], where it is assumed that, for any $y \in \mathbb{R}^n$:

$$\begin{aligned} f_y^2 &\geq f^2(y) - E^f \quad \text{and} \\ f^2(\cdot) &\geq f_y^2 + \langle g_y^2, \cdot - y \rangle - E^g. \end{aligned}$$

As in the exact setting of section 2.4.2, inexact bundle methods generate a sequence of trial points x^ℓ at which functions values and subgradients are evaluated. The subsequence of stability centers $\{\hat{x}^{k(\ell)}\}$ is chosen so that the corresponding objective functions are significantly decreasing. Trial points are provided by the solution of the quadratic program (4.5):

$$x^{\ell+1} := \arg \min_{y \in \mathbb{R}^n} \left\{ \Phi_\ell(y) + \frac{1}{2} \mu_\ell |y - \hat{x}^{k(\ell)}|^2 \right\}, \quad (4.5)$$

where Φ_ℓ is the disaggregate cutting planes model approximating f , $\mu_\ell > 0$ is the prox-parameter and $\hat{x}^{k(\ell)}$ is the current proximal center. As for the cutting-planes model \tilde{f}_ℓ of section 2.4.2, Φ_ℓ is defined by:

$$\begin{aligned} \Phi_\ell(\cdot) &:= (\tilde{f}_\ell^1 + \varphi_\ell^2)(\cdot), \quad \text{with} \\ \tilde{f}_\ell^1(\cdot) &:= f^1(\hat{x}^{k(\ell)}) + \max_{j \in \mathcal{B}_\ell^1} \left\{ -e_{j,\ell}^1 + \langle g^1(x^j), \cdot - \hat{x}^{k(\ell)} \rangle \right\} \quad \text{and} \end{aligned} \quad (4.6)$$

$$\varphi_\ell^2(\cdot) := f_{\hat{x}^{k(\ell)}}^2 + \max_{j \in \mathcal{B}_\ell^2} \left\{ -e_{j,\ell}^2 + \langle g_{x_j}^2, \cdot - \hat{x}^{k(\ell)} \rangle \right\}, \quad (4.7)$$

where \mathcal{B}_ℓ^1 and \mathcal{B}_ℓ^2 denote the current bundles of cutting planes. The linearization errors $e_{j,\ell}^{1,2}$ measure the difference between cutting planes contained in $\mathcal{B}_\ell^{1,2}$ and the function value returned by the oracle for the current stability center:

$$\begin{aligned} e_{j,\ell}^1 &= f^1(\hat{x}^{k(\ell)}) - f^1(x^j) - \langle g^1(x^j), \hat{x}^{k(\ell)} - x^j \rangle, \\ e_{j,\ell}^2 &= f_{\hat{x}^{k(\ell)}}^2 - f_{x_j}^2 - \langle g_{x_j}^2, \hat{x}^{k(\ell)} - x^j \rangle. \end{aligned} \quad (4.8)$$

When using an exact oracle, i.e. for f^1 , the linear model underestimates the real function and is exact for previous candidates if the corresponding elements have not been deleted from the bundle. As a consequence, linearization errors $e_{j,\ell}^1$ are always nonnegative. However, in the inexact context $e_{j,\ell}^2$ may be negative. Indeed, by (4.2), (4.3) and (4.8), $e_{j,\ell}^2$ is only known to satisfy

$$e_{j,\ell}^2 \geq -(E_{\hat{x}}^f + E_{x_j}^g). \quad (4.9)$$

Furthermore, model φ_ℓ^2 may overestimate the real function f^2 at some points, since by (4.3) only inequality (4.10) holds:

$$\varphi_\ell^2(\cdot) \leq f^2(\cdot) + \max_{j \in \mathcal{B}_\ell^2} E_{x_j}^g. \quad (4.10)$$

Writing the optimality conditions for (4.5), we know that there are simplicial multipliers $\{\alpha_j^1\}_{j \in \mathcal{B}_\ell^1}$ and $\{\alpha_j^2\}_{j \in \mathcal{B}_\ell^2}$ such that

$$\begin{aligned} x^{\ell+1} &= \hat{x}^{k(\ell)} + \frac{1}{\mu_\ell} \hat{s}^{\ell+1}, \text{ where} \\ \hat{s}^{\ell+1} &= \sum_{j \in \mathcal{B}_\ell^1} \alpha_j^1 g^1(x^j) + \sum_{j \in \mathcal{B}_\ell^2} \alpha_j^2 g_{x^j}^2 \text{ is the aggregate subgradient.} \end{aligned}$$

The vector $\hat{s}^{\ell+1}$ is a convex combination of inexact subgradients of f . From the optimality conditions of (4.5), we know that $\hat{s}^{\ell+1} \in \partial \Phi_\ell(x^{\ell+1})$. We define the corresponding aggregate linearization for the model as

$$\Phi_\ell^{\text{lin}}(\cdot) = \Phi_\ell(x^{\ell+1}) + \langle \hat{s}^{\ell+1}, \cdot - x^{\ell+1} \rangle.$$

The aggregate linearization error $\hat{\varepsilon}_{\ell+1} := \sum_{j \in \mathcal{B}_\ell^1} \alpha_j^1 e_{j,\ell}^1 + \sum_{j \in \mathcal{B}_\ell^2} \alpha_j^2 e_{j,\ell}^2$ corresponds to the difference between the value of the oracle at the proximal center and the value of the aggregate linearization at that point:

$$\hat{\varepsilon}_{\ell+1} = (f^1(\hat{x}^{k(\ell)}) + f_{\hat{x}^{k(\ell)}}^2) - \Phi_\ell^{\text{lin}}(\hat{x}^{k(\ell)}).$$

As a convex combination of linearization errors ($e_{j,\ell}^1 + e_{j,\ell}^2$), the aggregate linearization error satisfies:

$$\hat{\varepsilon}_{\ell+1} \geq -(E_{\hat{x}}^f + \sum_{j \in \mathcal{B}_{\ell 2}} \alpha_j^2 E_{x_j}^g) \geq E_{\max}^f + E_{\max}^g. \quad (4.11)$$

The predicted descent takes as reference the value returned by the oracle for the current stability center and is defined as $\delta_{\ell+1} = \delta_{\ell+1}^1 + \delta_{\ell+1}^2$, where

$$\begin{aligned} \delta_{\ell+1}^1 &= f^1(\hat{x}^{k(\ell)}) - \tilde{f}^1(x^{\ell+1}), \text{ and} \\ \delta_{\ell+1}^2 &= f_{\hat{x}^{k(\ell)}}^2 - \varphi^2(x^{\ell+1}). \end{aligned}$$

A serious step is declared when the following inequality is satisfied:

$$\left(f^1(\hat{x}^{k(\ell)}) + f_{\hat{x}^{k(\ell)}}^2 \right) - \left(f^1(x^{\ell+1}) + f_{x^{\ell+1}}^2 \right) \geq m \delta_{\ell+1} \quad (4.12)$$

We recall that the predicted descent can be expressed in terms of the norm of the aggregate subgradient and the aggregate linearization error (e.g. [BGLS06, Lemme 9.8]):

$$\delta_{\ell+1} = \frac{|\hat{s}^{\ell+1}|^2}{\mu_\ell} + \hat{\varepsilon}_{\ell+1}. \quad (4.13)$$

Using the fact that $x^{\ell+1}$ solves the quadratic program (4.5), we also have that

$$\begin{aligned} \delta_{\ell+1} &\geq \frac{1}{2} \mu_\ell |x^{\ell+1} - \hat{x}^{k(\ell)}|^2 - \left(\tilde{f}_\ell^1(\hat{x}^{k(\ell)}) + \varphi_\ell^2(\hat{x}^{k(\ell)}) - f^1(\hat{x}^{k(\ell)}) - f_{\hat{x}^{k(\ell)}}^2 \right) \\ &= \frac{1}{2} \mu_\ell |x^{\ell+1} - \hat{x}^{k(\ell)}|^2 - \left(\varphi_\ell^2(\hat{x}^{k(\ell)}) - f_{\hat{x}^{k(\ell)}}^2 \right). \end{aligned}$$

Hence, as soon as a negative linearization error occurs, the predicted descent may be negative too. The inequality (4.14) below can be used as an accuracy test and should be satisfied if only ‘‘small errors’’ have been introduced in the model φ^2

$$\delta_{\ell+1} > \frac{1}{2} \mu_\ell |x^{\ell+1} - \hat{x}^{k(\ell)}|^2. \quad (4.14)$$

By (4.13), we have the following equivalent inequalities for (4.14):

$$(4.14) \Leftrightarrow \frac{|\hat{s}^{\ell+1}|^2}{2\mu_\ell} > -\hat{\varepsilon}_{\ell+1} \quad (4.15a)$$

$$\Leftrightarrow \delta_{\ell+1} > \frac{|\hat{s}^{\ell+1}|^2}{2\mu_\ell}. \quad (4.15b)$$

Finally, as in [Kiw06, Eq.2.16], we define the optimality measure as

$$V_\ell := \max\{|\hat{s}^{\ell+1}|, \hat{\varepsilon}_{\ell+1}\}.$$

Indeed, by linearity of Φ_ℓ^{lin} we know that:

$$\begin{aligned} \Phi_\ell^{lin}(\cdot) &= \Phi^{lin}(\hat{x}^{k(\ell)}) + \langle \hat{s}^{\ell+1}, \cdot - \hat{x}^{k(\ell)} \rangle \\ &= (f^1(\hat{x}^{k(\ell)}) + f_{\hat{x}^{k(\ell)}}^2) + \langle \hat{s}^{\ell+1}, \cdot - \hat{x}^{k(\ell)} \rangle - \hat{\varepsilon}_{\ell+1}. \end{aligned}$$

By the subgradient inequality, $\Phi_\ell^{lin}(\cdot) \leq \Phi_\ell(\cdot)$, so we have that for any $y \in \mathbb{R}^n$,

$$(f^1(\hat{x}^{k(\ell)}) + f_{\hat{x}^{k(\ell)}}^2) \leq \Phi_\ell(y) - \langle \hat{s}^{\ell+1}, y - \hat{x}^{k(\ell)} \rangle + \hat{\varepsilon}_{\ell+1}.$$

By (4.10), this implies that for any $y \in \mathbb{R}^n$,

$$(f^1(\hat{x}^{k(\ell)}) + f_{\hat{x}^{k(\ell)}}^2) \leq f(y) + \max_{j \in \mathcal{B}_L^2} E_{x^j}^g + \langle \hat{s}^{\ell+1}, y - \hat{x}^{k(\ell)} \rangle + \hat{\varepsilon}_{\ell+1}. \quad (4.16)$$

Using again (4.13), we deduce the following inequalities regarding the optimality measure $V_\ell := \max\{|\hat{s}^{\ell+1}|, \hat{\varepsilon}_{\ell+1}\}$:

$$V_\ell \leq \max\{\sqrt{2\mu_\ell \delta_{\ell+1}}, \delta_{\ell+1}\}, \text{ if (4.14) holds, and} \quad (4.17)$$

$$V_\ell < \sqrt{-2\mu_\ell \hat{\varepsilon}_{\ell+1}} \leq \sqrt{2\mu_\ell (E_{max}^f + E_{max}^g)}, \text{ otherwise.} \quad (4.18)$$

The following relations justify the choice of V_ℓ as optimality measure.

Suppose that for a subsequence $\{x^\ell\}_{\ell \in \mathcal{L}}$, $V_\ell := \max\{|\hat{s}^{\ell+1}|, \hat{\varepsilon}_{\ell+1}\} \rightarrow 0$ as $\mathcal{L} \ni \ell \rightarrow \infty$, and suppose furthermore that the corresponding subsequence of stability centers $\{\hat{x}^{k(\ell)}\}_{\ell \in \mathcal{L}}$ forms a bounded sequence.

Then, passing to the limit in inequality (4.16) above, we have that

$$\lim_{\mathcal{L} \ni \ell \rightarrow \infty} (f^1(\hat{x}^{k(\ell)}) + f_{\hat{x}^{k(\ell)}}^2) \leq f^{opt} + \limsup_{\ell \in \mathcal{L}} \left(\max_{j \in \mathcal{B}_\ell^2} E_{x^j}^g \right).$$

Moreover, for any cluster point \hat{x}^{acc} of $\{x^\ell\}_{\ell \in \mathcal{L}}$, passing to the limit in (4.2),

$$f(\hat{x}^{acc}) - \limsup_{\ell \in \mathcal{L}} E_{\hat{x}^{k(\ell)}}^f \leq \lim_{\mathcal{L} \ni \ell \rightarrow \infty} (f^1(\hat{x}^{k(\ell)}) + f^2_{\hat{x}^{k(\ell)}}).$$

Summing up, x^{acc} is an approximate solution to (4.1) with

$$f(\hat{x}^{acc}) \leq f^{opt} + \limsup_{\ell \in \mathcal{L}} E_{\hat{x}^{k(\ell)}}^f + \limsup_{\ell \in \mathcal{L}} \left(\max_{j \in \mathcal{B}_\ell^2} E_{x^j}^g \right). \quad (4.19)$$

In particular, if for some iteration ℓ we have that $V_\ell = 0$, the stability center $\hat{x}^{k(\ell)}$ is an approximate solution of (4.1) with

$$f(\hat{x}^{k(\ell)}) \leq f^{opt} + E_{\hat{x}^{k(\ell)}}^f + \max_{j \in \mathcal{B}_\ell^2} E_{x^j}^g.$$

Similarly, if the stability center sequence stabilizes, i.e. if there is some L such that $\hat{x}^{k(\ell)} = \hat{x}^{k(L)}$ for all $\ell \geq L$, $\hat{x}^{k(L)}$ is an approximate solution of (4.1) with

$$f(\hat{x}^{k(L)}) \leq f^{opt} + E_{\hat{x}^{k(L)}}^f + \limsup_{\ell \in \mathcal{L}} \left(\max_{j \in \mathcal{B}_\ell^2} E_{x^j}^g \right). \quad (4.20)$$

The adapted bundle method for inexact linearizations proposed in [Kiw06] introduces a prox-parameter management step based on inequality (4.14) and on a noise attenuation parameter $NAP \geq 0$. Schematically, when (4.14) does not hold, μ_ℓ is reduced in order to get $\delta_{\ell+2} > \delta_{\ell+1}$ and help ensuring (4.14). This additional step is called here of Noise Management.

Algorithm 4.1.1 (*inexact bundle method with varying inaccuracy*)

1. **Initialization:** choose $x^1 \in \mathbb{R}^n$, $\kappa < 1$, $m < 1$, set $NAP = 0$, compute $(f^1(x^1), g^1(x^1))$ and $(f_{x^1}^2, g_{x^1}^2)$, define \check{f}_1^1 and φ_1^2 , set $\ell = k = 1$ and $\hat{x}^1 = x^1$.
2. **New candidate:** solve (4.5) to get $x^{\ell+1}$, $\delta_{\ell+1}$, $\hat{s}^{\ell+1}$, and $\hat{\varepsilon}_{\ell+1}$.
3. **Stopping test:** If $V_\ell = 0$, stop.
4. **Noise management:** if (4.14) does not hold, set $NAP = 1$, $\mu_{\ell+1} = \kappa\mu_\ell$, $\ell = \ell + 1$, go to 2.
5. **Oracle:** compute $f^1(x^{\ell+1}), g^1(x^{\ell+1})$ and $f_{x^{\ell+1}}^2, g_{x^{\ell+1}}^2$
6. **Descent test:** If $f^1(x^{\ell+1}) + f_{x^{\ell+1}}^2 \leq f^1(\hat{x}^{k(\ell)}) + f_{\hat{x}^{k(\ell)}}^2 - m\delta_{\ell+1}$, declare a Serious Step, otherwise declare a Null Step.
If $x^{\ell+1}$ gave a serious step, set $\hat{x}^{k+1} = x^{\ell+1}$, $NAP = 0$, $f_{\hat{x}^{k+1}}^2 = f_{x^{\ell+1}}^2$, $k = k + 1$, choose $\mu_{\ell+1} \leq \mu_{\max}$.

If $x^{\ell+1}$ gave a null step: if $NAP = 0$ choose $\mu_{\ell+1} \in [\mu_\ell, \mu_{\max}]$, else if $NAP = 1$ take $\mu_{\ell+1} = \mu_\ell$.

7. **Update:** Define $\check{f}_{\ell+1}^1$ and $\varphi_{\ell+1}^2$, $\ell = \ell + 1$, loop to 2.

If the algorithmic scheme above does not stop, three situations may occur:

- an infinite loop between steps 2. and 4., driving μ_ℓ to 0;
- a finite number of serious steps, followed by an infinite number of null steps;
- an infinite number of serious steps.

4.1.2 Convergence analysis

The inexact setting presented above is quite similar to the one proposed by Kiwiel [Kiw06] but we allow the inaccuracy (E_x^f, E_x^g) to vary along iterations. We will see here that the convergence results for the inexact bundle method of [Kiw06] remain almost unchanged.

In the following, we study separately these three cases to show that they all yield an approximate minimizer in (4.1).

Infinite loop of noise management:

Suppose that at some iteration L , an infinite loop between steps 2. and 4. begins, so that for $\ell \geq L$ neither the stability center $\hat{x}^{k(\ell)} = \hat{x}^{k(L)}$ nor the model $\Phi_\ell = \Phi_L$ change. Hence, when solving sequentially the quadratic program (4.5), only μ_ℓ is updated. Since $\mu_\ell \rightarrow 0$, using (4.18), we have that

$$0 \leq V_\ell < \sqrt{2\mu_\ell E^{\max}} \rightarrow 0 \text{ as } \ell \rightarrow \infty .$$

Applying (4.20) we have that if an infinite loop of Noise Management occurs in algorithm scheme 4.1.1, $\hat{x}^{k(L)}$ is an approximate solution to (4.1) with

$$f(\hat{x}^{k(L)}) \leq f^{\text{opt}} + \limsup_{\ell \in \mathcal{L}} \left(\max_{j \in \mathcal{B}_\ell^2} E_{x^j}^g \right) + E_{\hat{x}^{k(L)}}^f .$$

When inaccuracies are fixed and an infinite loop of Noise Management occurs, we recall the result of [Kiw06] :

$$f(\hat{x}) \leq f^{\text{opt}} + E^f + E^g .$$

We suppose now that the algorithm never enters an infinite loop of noise management.

Finite number of serious steps:

Here, after some iteration last , no serious step is declared. Hence, either noise management steps or null steps are done for $\ell \geq \text{last}$. The stability center does not move: for $\ell \geq \text{last}$, $\hat{x}^{k(\ell)} = x^{k(\text{last})} =: \hat{x}$.

If the number of noise management steps is infinite, i.e., if (4.14) does not hold for an infinite subsequence of iterates $\{x^{\ell+1}\}_{\ell \in \mathcal{L}}$, we have again that $\mu_\ell \rightarrow 0$ as $\mathcal{L} \ni \ell \rightarrow \infty$. The previous developments hold for that subsequence and $V_\ell \rightarrow 0$ as $\mathcal{L} \ni \ell \rightarrow \infty$.

Hence, by (4.20), if a finite number of serious steps occurs followed by an infinite number of Noise Management steps, \hat{x} is an approximate solution to (4.1) with

$$f(\hat{x}) \leq f^{opt} + \limsup_{\ell \in \mathcal{L}} \left(\max_{j \in B_\ell^2} E_{x^j}^g \right) + \limsup_{\ell \in \mathcal{L}} E_{\hat{x}^{k(\ell)}}^f.$$

Suppose now that only a finite number of noise management steps are done. There is some iteration $L \geq \ell_{ast}$ such that (4.14) holds. Consequently, μ_ℓ is increasing for $\ell > L$, and $\mu_\ell \rightarrow \bar{\mu} \leq \mu_{max}$ as $\ell \rightarrow \infty$. In the following, we derive standard arguments for bundle methods to show that $\delta_\ell \rightarrow 0$. We define the partial linearization of the objective of (4.5) by:

$$L_\ell(x) := \Phi_\ell^{lin}(x) + \frac{1}{2}\mu_\ell|x - \hat{x}|^2.$$

First, using Lemma 9.10 in [BGLS06], we know that the rules to apply compression/selection on the bundle guarantee that

$$\Phi_\ell^{lin}(\cdot) \leq \Phi_{\ell+1}(\cdot).$$

By (4.10),

$$\begin{aligned} L_\ell(\hat{x}) &= \Phi_\ell^{lin}(\hat{x}) \\ &\leq \Phi_{\ell+1}(\hat{x}) \\ &\leq f(\hat{x}^{k(\ell)}) + E_{max}^g \end{aligned} \quad (4.21)$$

Similarly, evaluating L_ℓ at $x^{\ell+1}$ and using the fact that $\mu_{\ell+1} \geq \mu_\ell$ gives

$$\begin{aligned} L_\ell(x^{\ell+2}) &\leq \Phi_{\ell+1}(x^{\ell+2}) + \frac{1}{2}\mu_{\ell+1}|x^{\ell+2} - \hat{x}|^2 \\ &= \Phi_{\ell+1}^{lin}(x^{\ell+2}) + \frac{1}{2}\mu_{\ell+1}|x^{\ell+2} - \hat{x}|^2 \\ &= L_{\ell+1}(x^{\ell+2}). \end{aligned}$$

Furthermore, $x^{\ell+1}$ being the solution to (4.5), $\nabla L_\ell(x^{\ell+1}) = 0$ and by Taylor's expansion,

$$L_\ell(\cdot) = L_\ell(x^{\ell+1}) + \frac{1}{2}\mu_\ell|\cdot - x^{\ell+1}|^2.$$

Hence,

$$\begin{aligned} L_\ell(x^{\ell+2}) &= L_\ell(x^{\ell+1}) + \frac{1}{2}\mu_\ell|x^{\ell+2} - x^{\ell+1}|^2, \text{ and} \\ L_\ell(\hat{x}) &= L_\ell(x^{\ell+1}) + \frac{1}{2}\mu_\ell|\hat{x} - x^{\ell+1}|^2. \end{aligned}$$

Using the developments above, the fact that $\mu_\ell \geq \mu_L$, and (4.21), we obtain the following inequalities:

$$L_\ell(x^{\ell+1}) + \frac{1}{2}\mu_L|x^{\ell+2} - x^{\ell+1}|^2 \leq L_{\ell+1}(x^{\ell+2}), \text{ and} \quad (4.22)$$

$$L_\ell(x^{\ell+1}) + \frac{1}{2}\mu_L|x^{\ell+1} - \hat{x}|^2 = L_\ell(\hat{x}) \leq f(\hat{x}) + E_{max}^g. \quad (4.23)$$

From (4.23) and (4.22), we deduce that the sequence $\{L_\ell(x^{\ell+1})\}_{\ell \geq L}$ is increasing and bounded. Hence,

$$\exists L^\infty < \infty : L_\ell(x^{\ell+1}) \rightarrow L^\infty \text{ and } |x^{\ell+1} - x^\ell| \rightarrow 0 \text{ as } \ell \rightarrow \infty. \quad (4.24)$$

Furthermore, the sequence of null steps $\{x^\ell\}$ is bounded, by (4.23). As a consequence, $\partial_{E_f^{max} + E_g^{max}} f(\cdot)$ being locally bounded (e.g. [HUL96] Prop 6.2.2), $\{g^\ell\}$ is also bounded.

Since only null steps occur for $\ell > L$, the descent test is not satisfied and we have that $(f^1(x^{\ell+1}) + f_{x^{\ell+1}}^2) - \Phi_\ell(x^{\ell+1}) > (1 - m)\delta_\ell$. However, since the model $\Phi_{\ell+1}$ includes the information returned by the oracle for the last iterate $x^{\ell+1}$,

$$(f^1(x^{\ell+1}) + f_{x^{\ell+1}}^2) \leq \Phi_{\ell+1}(x^{\ell+2}) + \langle g^{\ell+1}, x^{\ell+1} - x^{\ell+2} \rangle,$$

so, together with the definition of partial linearization, we obtain that

$$\begin{aligned} (f^1(x^{\ell+1}) + f_{x^{\ell+1}}^2) - \Phi_\ell(x^{\ell+1}) &\leq \Phi_{\ell+1}(x^{\ell+2}) - \Phi_\ell(x^{\ell+1}) + |g^\ell| |x^{\ell+1} - x^{\ell+2}| \\ &= \Phi_{\ell+1}^{lin}(x^{\ell+2}) - \Phi_\ell^{lin}(x^{\ell+1}) + |g^\ell| |x^{\ell+1} - x^{\ell+2}| \\ &= L_{\ell+1}(x^{\ell+2}) - L_\ell(x^{\ell+1}) + |g^\ell| |x^{\ell+1} - x^{\ell+2}| \\ &\quad - \frac{1}{2}\mu_{\ell+1}|x^{\ell+2} - \hat{x}|^2 + \frac{1}{2}\mu_\ell|x^{\ell+1} - \hat{x}|^2 \end{aligned}$$

Using (4.24) and $\mu_\ell \rightarrow \bar{\mu} \leq \mu_{max}$ we obtain that all terms on the right hand side vanish as $\ell \rightarrow \infty$. As a consequence, $(f^1(x^{\ell+1}) + f_{x^{\ell+1}}^2) - \Phi_\ell(x^{\ell+1}) \rightarrow 0$ as $\ell \rightarrow \infty$. Hence,

$$0 \leq (1 - m)\delta_\ell < (f^1(x^{\ell+1}) + f_{x^{\ell+1}}^2) - \Phi_\ell(x^{\ell+1}) \rightarrow 0.$$

By (4.17), for $\ell > L$ we have that $V_\ell \leq \max\{\sqrt{2\mu_\ell\delta_{\ell+1}}, \delta_{\ell+1}\}$ since (4.14) holds. Hence, $V_\ell \rightarrow 0$ as $\ell \rightarrow \infty$.

Hence, by (4.20), if a finite number of serious steps occurs, \hat{x} is an approximate solution to (4.1) with

$$f(\hat{x}) \leq f^{opt} + \limsup_{\ell \rightarrow \infty} \left(\max_{j \in \mathcal{B}_\ell^2} E_{x^j}^g \right) + \limsup_{\ell \rightarrow \infty} E_{\hat{x}^{k(\ell)}}^f.$$

When inaccuracies are fixed and a finite number of serious steps occurs, we recall the result of [Kiw06] :

$$f(\hat{x}) \leq f^{opt} + E^f + E^g.$$

Infinite number of serious steps:

For notational convenience, when the sequence of serious steps is infinite, we denote by \mathcal{K} the subset of iterates that are chosen as serious steps. Hence, $\{x^k\}_{k \in \mathcal{K}}$ and $\{\delta^k\}_{k \in \mathcal{K}}$ correspond to the successive stability centers and predicted descents (in the more formal notation of the algorithmic pattern, x^k corresponds to \hat{x}^k and δ_k to $\delta_{\ell+1}$ with ℓ the iteration yielding $\hat{x}^k = x^{\ell+1}$).

By construction, the sequence $\{f^1(x^k) + f_{x^k}^2\}_{k \in \mathcal{K}}$ is strictly decreasing. Hence,

$$(f^1(x^k) + f_{x^k}^2) \rightarrow f^\infty \geq f^{opt} - E_{max}^f \text{ as } \mathcal{K} \ni k \rightarrow \infty.$$

Applying the descent test inequality we have that for two successive stability centers x^{k+} and x^k :

$$(f^1(x^{k+}) + f_{x^{k+}}^2) - (f^1(x^k) + f_{x^k}^2) > m\delta_{k+}.$$

By summing this inequality for all serious steps, we deduce that

$$\sum_{k \in \mathcal{K}} \delta_k < \infty, \text{ with } \delta_k > 0.$$

Hence, $\delta_k \rightarrow 0$ as $\mathcal{K} \ni k \rightarrow \infty$. Then, by (4.17), for $k \in \mathcal{K}$ we have that $V_k \leq \max\{\sqrt{2\mu_{max}}\delta_{k+1}, \delta_{k+1}\}$ since (4.14) holds, and again $V_k \rightarrow 0$ as $\mathcal{K} \ni k \rightarrow \infty$. Recall that $\{x^k\}_{k \in \mathcal{K}}$ is bounded by coercivity of f .

Hence, by (4.19), if an infinite number of serious step are declared in algorithmic scheme 4.1.1, any accumulation point x^{acc} of the bounded sequence $\{x^k\}_{k \in \mathcal{K}}$ is such that :

$$f(\hat{x}^{acc}) \leq f^{opt} + \limsup_{\ell \in \mathcal{L}} \left(\max_{j \in \mathcal{B}_\ell^2} E_{x^j}^g \right) + \limsup_{\ell \in \mathcal{L}} E_{\hat{x}^{k(\ell)}}^f.$$

When inaccuracies are fixed and an infinite number of serious steps occurs, we recall the result of [Kiw06] :

$$f(\hat{x}^{acc}) \leq f^{opt} + E^f + E^g.$$

4.2 An alternative approach when $E_{x^\ell}^g = 0$ and $E_{x^\ell}^f \rightarrow 0$

In many practical situations, the error term $E_{x^\ell}^g$ is null and consequently the cutting planes always stay below the real objective function (cf. (4.3)). Examples of inexact oracles proposed in [Kiw06] correspond to this framework, and so does the case of incremental bundle methods presented in the next section.

When the error vanishes along the iterative process, i.e., when $E_{x^\ell}^f \rightarrow 0$ as $\ell \rightarrow \infty$, we could expect to be able to find (asymptotically) an optimal solution, without any error.

When applying the results presented in Section 4.1 to the particular setting $E_x^f \rightarrow 0$ but $E_x^g = 0$ for any x , the following can be said for algorithm 4.1.1:

1. if a finite number of serious steps occurs, the last serious step $\hat{x}^{k(last)}$ provides an approximate solution with error on the objective lower than $E_{\hat{x}^{k(last)}}^f$;
2. if an infinite number of serious steps occurs, any cluster point of the sequence of serious steps is solution to (4.1).

However, in this framework, as soon as the inexact oracle returns a value $f^1(x^{\ell+1}) + f_{x^{\ell+1}}^2$ below the infimum value of the cutting planes model Φ_ℓ , an infinite noise management loop will start, because only negative predicted descents can be produced. In these circumstances, the last serious step $x^{k(last)}$ will be an approximate solution to (4.1), with error on the objective bounded by $E_{\hat{x}^{k(last)}}^f$, by (4.20).

As a by-product of the analysis in Section 4.1, we studied the behavior of a method working with function evaluations of *unknown* and *vanishing* accuracy, but without noise attenuation step. We sometimes refer to this method as *inexact classical bundle method*.

4.2.1 Inexact classical bundle method

Close to this setting, [Kiw95] considers the case of varying accuracies that can be arbitrarily modified along the process to be driven to 0. Under this assumption, the method finds asymptotically an optimal solution to (4.1). In [Hin01], the case of unknown and vanishing accuracy is handled, but it requires exact function evaluation while subgradients may be computed with errors.

In [Sol03], the author gives a bound on the accuracy necessary to achieve a given tolerance. The algorithm involved is almost identical to a proximal bundle method and

as a consequence share similarities with our approach. However, vanishing errors may not asymptotically produce optimal solutions.

Our analysis starts from the following observation: when adding a new cutting plane to the bundle after a null step, the value $f_{\hat{x}^{k(\ell)}}$ may not be the best estimate available for $f(\hat{x}^{k(\ell)})$. Indeed, we have that

$$f(\hat{x}^{k(\ell)}) \geq \Phi_\ell(\hat{x}^{k(\ell)}) \geq f_{\hat{x}^{k(\ell)}}.$$

Hence, one could consider replacing $(f^1(\hat{x}^{k(\ell)}) + f_{\hat{x}^{k(\ell)}}^2)$ by the possibly better value $\Phi_\ell(\hat{x}^{k(\ell)})$. This results in an alternative definition for the predicted descent. More precisely,

$$\begin{aligned} \text{either } \delta_{\ell+1} &:= (f^1(\hat{x}^{k(\ell)}) + f_{\hat{x}^{k(\ell)}}^2) - \Phi_\ell(x^{\ell+1}) \text{ as in last section,} \\ \text{or } \hat{\delta}_{\ell+1} &:= \Phi_\ell(\hat{x}^{k(\ell)}) - \Phi_\ell(x^{\ell+1}) \text{ in this section.} \end{aligned} \quad (4.25)$$

As a consequence, in Step 6 of the algorithmic pattern 4.1.1 page 66, two different descent tests may be performed: $x^{\ell+1}$ qualifies as new stability center either if (4.12) holds, i.e., if

$$(f^1(x^{\ell+1}) + f_{x^{\ell+1}}^2) - (f^1(\hat{x}^k) + f_{\hat{x}^k}^2) > m \left(f^1(\hat{x}^{k(\ell)}) + f_{\hat{x}^{k(\ell)}}^2 - \Phi_\ell(x^{\ell+1}) \right) \quad (4.26)$$

$$\text{or if: } (f^1(x^{\ell+1}) + f_{x^{\ell+1}}^2) - \Phi_\ell(\hat{x}^{k(\ell)}) > m \left(\Phi_\ell(\hat{x}^{k(\ell)}) - \Phi_\ell(x^{\ell+1}) \right). \quad (4.27)$$

Actually, the two alternative definitions for predicted descent are related through the linearization errors $e_{j,\ell}^{1,2}$ used in the definition of the cutting planes models (4.6) and (4.7). Indeed, we have

$$\hat{\delta}_{\ell+1} = \delta_{\ell+1} + \max_{j \in \mathcal{B}_\ell^2} \{-e_{j,\ell}^2\}.$$

Similarly to (4.13), the predicted descent $\hat{\delta}_{\ell+1}$ can be expressed in terms of the aggregate subgradient, the proximal parameter and the linearization errors:

$$\hat{\delta}_{\ell+1} = \frac{1}{\mu_\ell} |\hat{s}^{\ell+1}|^2 + \hat{\varepsilon}_{\ell+1} + \max_{j \in \mathcal{B}_\ell^2} \{-e_{j,\ell}^2\}.$$

Furthermore, $\hat{\varepsilon}_{\ell+1}$ may be negative but, since $\hat{\varepsilon}_{\ell+1}$ is a convex combination of linearization errors by definition,

$$\hat{\varepsilon}_{\ell+1} + \max_{j \in \mathcal{B}_\ell^2} \{-e_{j,\ell}^2\} \geq 0.$$

As a consequence, if $\hat{\delta}_{\ell+1} \rightarrow 0$ and using the fact that $\mu_\ell \leq \mu_{max}$, we have that:

$$\begin{aligned} \hat{\varepsilon}_{\ell+1} + \max_{j \in \mathcal{B}_\ell^2} \{-e_{j,\ell}^2\} &\rightarrow 0 \text{ and} \\ |\hat{s}^{\ell+1}| &\rightarrow 0. \end{aligned}$$

It is important to notice that $\hat{\delta}_{\ell+1}$ is always nonnegative. Hence, the additional information used in the new definition avoids having to perform noise attenuation steps. Indeed, recall that in the algorithmic pattern presented in the previous section, when $\delta_{\ell+1}$ is not considered large enough to cope with the inaccuracy, the proximal parameter μ_ℓ is increased and a new trial point is generated, before creating any new cutting plane. Here, we consider instead a more (inexact) classical bundle method, following the algorithmic scheme of section 2.4.2 page 33, where $\delta_{\ell+1}$ is replaced by $\hat{\delta}_{\ell+1}$.

Algorithm 4.2.1 (“inexact classical” bundle method)

1. **Initialization:** choose $x^1 \in \mathbb{R}^n$, compute $(f^1(x^1), g^1(x^1))$ and $(f_{x^1}^2, g_{x^1}^2)$, define \check{f}_1^1 and φ_1^2 , set $\ell = k = 1$ and $\hat{x}^1 = x^1$.
2. **New candidate:** solve (4.5) to get $x^{\ell+1}$, $\hat{\delta}_{\ell+1}$, $\hat{s}^{\ell+1}$, and $\hat{\varepsilon}_{\ell+1}$.
3. **Stopping test:** If $V_\ell = 0$, stop.
4. **Oracle:** compute $f^1(x^{\ell+1}), g^1(x^{\ell+1})$ and $f_{x^{\ell+1}}^2, g_{x^{\ell+1}}^2$
5. **Descent test:** If $f^1(x^{\ell+1}) + f_{x^{\ell+1}}^2 \leq f^1(\hat{x}^{k(\ell)}) + f_{\hat{x}^{k(\ell)}}^2 - m\hat{\delta}_{\ell+1}$, declare a Serious Step, otherwise declare a Null Step.
 If $x^{\ell+1}$ gave a serious step, set $\hat{x}^{k+1} = x^{\ell+1}$, $f_{\hat{x}^{k+1}}^2 = f_{x^{\ell+1}}^2$, $k = k + 1$, choose $\mu_{\ell+1} \leq \mu_{\max}$.
 If $x^{\ell+1}$ gave a null step, choose $\mu_{\ell+1} \in [\mu_\ell, \mu_{\max}]$.
6. **Update:** Define $\check{f}_{\ell+1}^1$ and $\varphi_{\ell+1}^2$, $\ell = \ell + 1$, loop to 2.

4.2.2 Convergence analysis

In the following, we study separately the two possible asymptotic behaviors for Algorithm 4.2.1 : infinite or finite number of serious steps.

Infinite number of serious steps

First, recall that in the case of an infinite subsequence of iterations for which $\hat{\delta}_{\ell+1} \rightarrow 0$, since

$$0 \leq \max_{j \in \mathcal{B}_\ell^2} \{-e_{j,\ell}^2\} \leq E_{\hat{x}^{k(\ell)}}^f \rightarrow 0,$$

we have that $\hat{\varepsilon}_{\ell+1} \rightarrow 0$ and $|\hat{s}^{\ell+1}| \rightarrow 0$. We now show that for the infinite subsequence of iterations yielding a serious steps, the corresponding predicted decrease $\hat{\delta}_{\ell+1}$ converges to 0.

Once more, for notational convenience, we denote by \mathcal{K} the subset of iterates that are chosen as serious steps, with $k < k^+$ two iterations yielding successive serious steps x^k and x^{k^+} with corresponding predicted descents $\hat{\delta}_k$ and $\hat{\delta}_{k^+}$.

In the new setting, when performing a serious step, we cannot guarantee that we obtain a better point: the relation $f(x^k) > f(x^{k^+})$ does not necessarily hold. Moreover, and contrary to the case developed in §4.1.1, neither can we say that the inexact values returned by the oracle are improving: we may have $(f^1(x^k) + f_{x^k}^2) \leq (f^1(x^{k^+}) + f_{x^{k^+}}^2)$. Namely, by (4.25) and (4.27), only the following inequality holds:

$$\Phi_{k^+}(x^k) - (f^1(\hat{x}^{k^+}) + f_{x^{k^+}}^2) \geq m\hat{\delta}_{k^+}.$$

Using (4.10) and (4.9), it follows that:

$$\begin{aligned} (f^1(x^k) + f_{x^k}^2) - (f^1(x^{k^+}) + f_{x^{k^+}}^2) &> m\hat{\delta}_{k^+} + ((f^1(x^k) + f_{x^k}^2) - \Phi_{k^+}(x^k)) \\ &= m\hat{\delta}_{k^+} - \max_{\mathcal{B}^{k^+}} \{-e_{j,k^+}^2\} \\ &\geq m\hat{\delta}_{k^+} - E_{\hat{x}^k}^f. \end{aligned} \quad (4.28)$$

We define $f^\infty := \liminf_{k \in \mathcal{K}} (f^1(x^k) + f_{x^k}^2)$ and notice that $f^\infty > -\infty$, because f is bounded from below by assumption.

We first show that $\liminf_{k \in \mathcal{K}} \hat{\delta}_k = 0$. Indeed, for contradiction purposes, suppose that $\liminf_{k \in \mathcal{K}} \hat{\delta}_k > 0$. Then, since $E_{x^k}^f \rightarrow 0$, there exists $\eta_0 > 0$ and an iteration $k_0 \in \mathcal{K}$ such that $(m\hat{\delta}_{k^+} - E_{x^k}^f) \geq \eta_0$ for all $k \geq k_0$.

However, summing up inequality (4.28) for $k \geq k_0$ would result in having

$$\liminf_{k \in \mathcal{K}} (f^1(x^k) + f_{x^k}^2) = -\infty,$$

a contradiction. Hence, $\liminf \hat{\delta}_k = 0$. Let us denote by \mathcal{J} a subset of iterates such that $\hat{\delta}_k \rightarrow 0$ as $\mathcal{J} \ni k \rightarrow \infty$.

Now, we denote by k^- the predecessor in \mathcal{K} of a stability center x^k with $k \in \mathcal{J}$, and $f_{\mathcal{J}}^\infty := \liminf_{k \in \mathcal{J}} (f^1(x^{k^-}) + f_{x^{k^-}}^2)$. Let $\mathcal{I} \subset \mathcal{J}$ be such that $(f^1(x^{k^-}) + f_{x^{k^-}}^2) \rightarrow f_{\mathcal{J}}^\infty$ as $\mathcal{I} \ni k \rightarrow \infty$. The level sets of f being bounded by coercivity, the corresponding sequence $\{x^{k^-}\}_{k \in \mathcal{I}}$ is bounded too. Let x^{acc} be a cluster point of $\{x^{k^-}\}_{k \in \mathcal{I}}$ and $\mathcal{H} \subset \mathcal{I}$ the subset of iterates such that $x^{k^-} \rightarrow x^{acc}$. Then, we have $x^{k^-} \rightarrow x^{acc}$ with $\delta_k \rightarrow 0$ as $\mathcal{H} \ni k \rightarrow \infty$.

Summing up, if an infinite number of serious steps occurs in the algorithmic scheme 4.2.1, the sequence of stability centers has at least one cluster point x^{acc} solution of (4.1).

Finite number of serious steps

Suppose there exists an iteration L such that for all $\ell > L$, only null steps occur with $\mu_{max} \geq \mu_{\ell+1} \geq \mu_\ell$ and the stability center does not change, i.e. $\hat{x}^{k(\ell)} = \hat{x}$. Let us show that $\hat{\delta}_\ell \rightarrow 0$ as $\ell \rightarrow \infty$ and that the last serious step is a solution to (4.1).

As in the previous section, we use the partial linearization of the objective of the quadratic program (4.5): $L_\ell(x) := \Phi_\ell^{lin}(x) + \frac{1}{2}\mu_\ell|x - \hat{x}^{k(\ell)}|^2$. Again, by (4.22) and (4.23), we have that

$$L_\ell(x^{\ell+1}) \uparrow L_\infty < \infty \quad \text{and} \quad |x^{\ell+1} - x^\ell| \rightarrow 0,$$

with $\{x^\ell\}$ and $\{g^\ell\}$ bounded.

With the previous developments, we also obtain that $(f^1(x^{\ell+1}) + f_{x^{\ell+1}}^2) - \Phi_\ell(x^{\ell+1}) \rightarrow 0$ as $\ell \rightarrow \infty$. But $(f^1(x^{\ell+1}) + f_{x^{\ell+1}}^2) - \Phi_\ell(x^{\ell+1}) = (f^1(x^{\ell+1}) + f_{x^{\ell+1}}^2) - \Phi_\ell(\hat{x}) + \Phi_\ell(\hat{x}) - \Phi_\ell(x^{\ell+1})$. Hence, since the descent test does not hold for $\ell \geq L$,

$$(1 - m)\hat{\delta}_{\ell+1} < f_x^{\ell+1} - \Phi_\ell(x^{\ell+1}) \rightarrow 0 \quad \text{as} \quad L \leq \ell \rightarrow \infty.$$

As a consequence,

$$\hat{\delta}_{\ell+1} = \hat{\varepsilon}_{\ell+1} + \frac{1}{\mu_\ell} |\hat{s}^{\ell+1}|^2 + \max_{j \in \mathcal{B}_\ell^2} \{-e_{j,\ell}^2\} \rightarrow 0.$$

Since $\hat{\varepsilon}_{\ell+1} + \max_{j \in \mathcal{B}_\ell^2} \{-e_{j,\ell}^2\} \geq 0$ and $\mu_\ell \leq \mu_{max}$, we have that $|\hat{s}^{\ell+1}|^2 \rightarrow 0$. Furthermore, $x^{\ell+1} = \hat{x} - \frac{1}{\mu_\ell} \hat{s}^{\ell+1}$, so $x^{\ell+1} \rightarrow \hat{x}$. As a consequence, since $E_{x^\ell}^f \rightarrow 0$, $\Phi_\ell(\hat{x}) \rightarrow f(\hat{x})$ as $\ell \rightarrow \infty$, so $\max_{j \in \mathcal{B}_\ell^2} \{-e_{j,\ell}^2\} \rightarrow 0$, and it follows that $\hat{\varepsilon}_{\ell+1} \rightarrow 0$. We can now conclude that \hat{x} is solution to (4.1).

Summing up, if a finite number of serious steps occurs in the algorithmic scheme 4.2.1, the last stability center \hat{x} is a solution of (4.1).

We have shown that an almost classical bundle method may achieve optimality when linearization are computed with inexact but vanishing inaccuracies. Our setting does not involve any noise management step as in [Kiw06] but rather includes incoming information into the definition of the predicted descent used in the descent test.

4.3 Incremental bundle methods

4.3.1 Motivations

An efficient way to tackle large-scale problems of generation planning is to perform Lagrangian relaxation on the coupling constraint of demand. The dual function is evaluated through sequential evaluation of local subfunctions. In practice however, solving all corresponding subproblems can still be very costly.

For the particular problem of mid-term generation planning, the cost of one iteration of the master problem is practically negligible, when compared to the CPU time needed to solve all the local subproblems. The 58 nuclear subproblems, in particular, are very bulky and time consuming. This difficulty led us to consider the possibility of performing a dual iteration after solving *only a subset* of local subproblems.

In this section, we consider an incremental strategy to take full advantage of the dual separable structure. The idea is to make a dual iteration after having evaluated only a subset of the dual local subfunctions, instead of all of them. We show how this approach is related to bundle methods with inexact linearizations. We conclude with an application to the French mid-term generation planning problem, with numerical results that confirm the potential of our strategy.

4.3.2 Related works

Incremental subgradient methods for non-smooth optimization have been proposed by Solodov and Zavriev [SZ98], see also [NB01]. These methods perform subgradient iterations sequentially, along the subgradients g^i of each subfunction f^i : a short description is given in §2.3.4. Promising numerical results are reported in [NB01] and these methods are further studied from a theoretical point of view by Kiwiel [Kiw03].

For bundle methods, Gaudioso et al. [GGM06] study the case of a non-smooth objective defined as the pointwise maximum of a large number of non-smooth subfunctions, but to our knowledge there is so far no study of incremental bundle methods for the *sum* of subfunctions. Bundle methods do not seem to be straightforwardly customizable to the “sum” setting. However, recent results on inexact bundle methods by Kiwiel [Kiw06] (extending previous works [Kiw85], [Kiw95], [Hin01], [Sol03] and completed recently for constrained optimization by [KL07]) shed a new light on how to proceed to devise incremental-like bundle methods in our setting of interest.

4.3.3 Incremental bundle methods

When skipping the evaluation of some subfunctions in a disaggregate bundle method, the lacking information needs to be replaced by some approximation. In this case, a natural substitute for f^i would be to use the individual cutting planes model, \tilde{f}_ℓ^i . Such replacement is closely related to the inexact bundle methods studied by various authors an especially to the framework developed in [Kiw06].

To simplify the presentation, we stay in the simpler case of minimizing the sum of only two functions:

$$\left\{ \min_{x \in \mathbb{R}^n} (f^1 + f^2)(x) . \right.$$

Functions f^i result from Lagrangian relaxation of a primal problem

$$\begin{cases} \max_{\{p^i\}_{i=1,2}} & C^1(p^1) + C^2(p^2) \\ \text{s.t.} & g^1(p^1) + g^2(p^2) = 0 \\ & p^i \in \mathcal{P}^i \end{cases} .$$

where functions g^i are affine, C^i concave and sets \mathcal{P}^i compact. Hence, dual functions are defined by $f^i(\cdot) = \max_{p \in \mathcal{P}^i} \{C^i(p) + \langle g^i(p), \cdot \rangle\}$. Typically, f^1 is easy to evaluate while f^2 is very hard, and time consuming. Inexactness appears when evaluating the ‘‘difficult’’ subfunction, f^2 . Replacing this subfunction by the corresponding current cutting planes model can be interpreted as an inexact evaluation.

Indeed, recall that in the inexact setting of [Kiw06], for any $x \in \mathcal{R}^N$ the oracle returns (f_x^2, g_x^2) satisfying:

$$\begin{aligned} f_x^2 &\geq f^2(x) - E^f \text{ and} \\ f^2(\cdot) &\geq f_x^2 + \langle g_x^2, \cdot - x \rangle - E^g, \end{aligned}$$

where E^f and E^g are fixed but unknown accuracies.

Now, suppose that instead of computing f^2 we use any $p^2 \in \mathcal{P}^2$ to extrapolate inexact values: $f_x^2 = C^2(p^2) + \langle g^2(p^2), x \rangle$, and $g_x^2 = g^2(p^2)$, for some $p^2 \in \mathcal{P}^2$. But, we know that $f^2(\cdot) = \max_{p \in \mathcal{P}^2} \{C^2(p) + \langle g^2(p), \cdot \rangle\}$. Hence, taking any $p^2 \in \mathcal{P}^2$, and choosing f_x^2 and g_x^2 as above, we have that:

$$\begin{aligned} f^2(y) &= \max_{p \in \mathcal{P}^2} \{C^2(p) + \langle g^2(p), y \rangle\} \\ &\geq C^2(p^2) + \langle g^2(p^2), y \rangle \\ &= f_x^2 + \langle g^2(p^2), y - x \rangle. \end{aligned}$$

Hence, such an oracle is compatible with the inexact setting of [Kiw06], with varying accuracy $E_x^f := C(\bar{p}_x^2) - C(p^2) + \langle g^2(\bar{p}_x^2) - g^2(p^2), x \rangle$ and $E^g = 0$. We notice that since \mathcal{P}^2 is compact, inaccuracy is bounded. A natural choice for p^2 would be, for example, to use some past response from the oracle. Past information is kept in the bundle to build the cutting planes model \tilde{f}^2 . Hence, the best choice for $f_{x^{\ell+1}}^2$ when using only the information of the bundle would be to evaluate instead the current cutting planes model. Indeed, since $\hat{g}^{2,\ell} \in \partial \tilde{f}_\ell^2(x^{\ell+1})$ and $\tilde{f}_\ell^2 \leq f$, when we use for the new iterate the inexact values $(f_{x^{\ell+1}}^2, g_{x^{\ell+1}}^2) := (\tilde{f}_\ell^2(x^{\ell+1}), \hat{g}^{2,\ell})$, we have

$$\begin{aligned} f^2(y) &\geq \tilde{f}_\ell^2(y) \\ &\geq \tilde{f}_\ell^2(x^{\ell+1}) + \langle \hat{g}^{2,\ell}, y - x^{\ell+1} \rangle \\ &= f_{x^{\ell+1}}^2 + \langle g_{x^{\ell+1}}^2, y - x^{\ell+1} \rangle. \end{aligned}$$

We still are in the setting of [Kiw06] with $E^g = 0$ and E_x^f varying along iterations and such that

$$E_{x^{\ell+1}}^f \leq \min_{j \in \mathcal{B}_\ell^2} \{C(\bar{p}_{x^{\ell+1}}^2) - C(\bar{p}_{x^j}^2) + \langle g^2(\bar{p}_{x^{\ell+1}}^2) - g^2(\bar{p}_{x^j}^2), x^{\ell+1} \rangle\}. \quad (4.29)$$

Since \mathcal{P}^i are compact sets, the error $E_{x^\ell}^f$ is bounded and satisfy assumption (4.4). Hence, the convergence analysis of section 4.1.2 applies and such incremental strategy will produce asymptotically an approximate solution. Developments of section 4.2 were motivated by the study of this incremental strategy as we conjectured that in such a setting, inaccuracies would be asymptotically driven to 0.

Finally, remark that the aggregate subgradient $\hat{g}^{2,\ell}$ is always a convex combination of exact subgradients of f^2 .

In the next section, we apply such a strategy to solve EDF’s mid-term generation planning problem using a price decomposition approach on a scenario tree.

4.4 Mid term generation planning

Electricité de France's (EDF) generation system is mainly composed by nuclear power plants, which account for about 75% of the total generation, with hydraulic and classical thermal plants accounting for the remaining power. The optimal management of such a generation mix roughly consists in optimizing generation costs to meet future loads of electricity. Short, middle and long term planning problems have their own objectives and specific modeling of operational constraints and random data. We focus here on mid-term generation planning, that considers a time horizon of 1 to 3 years.

In this framework, operational constraints are usually modeled with far less details than for short term scheduling. For example, hydroplants are often aggregated into energy equivalent reservoirs, also, thermal starting costs and ramp constraints are disregarded. For nuclear plants, it may be important to include stock constraints. The reason is that nuclear refueling happens only once every 18 months. In addition, there are specific constraints, linking the available power of each plant to the current level of combustible. Since the French generation mix has 58 nuclear reactors, a high number of reservoirs dynamics has to be optimized; and this, without even considering hydraulic stocks, which increase even more the state dimension of the problem.

For mid-term planning, many elements of the problem are no longer deterministic. In our case, the level of demand is the main source of uncertainty, particularly during winter periods. Other parameters that introduce randomness are the water inflows, the availability of thermal plants, or future prices on the spot market. Generation units are modeled with convex cost functions and constraints. Finally, all units are coupled by the satisfaction of demand constraint.

Hence, even with a description of the power mix that is simpler than the one used for short term scheduling, we have to deal with a large-scale optimization problem. Decomposition techniques may be used that replace the initial problem by a sequence of simpler problems, coordinated by a master program.

4.4.1 Price decomposition on a scenario tree

An approach currently used at EDF involves applying price decomposition on a scenario tree (e.g. [BLRS01]). Nonanticipativity constraints are implicitly represented in the tree structure : two scenarios which are undistinguishable before time t correspond to a single branch in the tree. The control at time t is then the same for both scenarios. Here, we simplify the notation by assuming that the only stochastic parameter is the demand. For practical applications, the scenario tree easily contains more than 50.000 nodes, which results in an optimization problem with more than 10^7 variables and 10^8 constraints.

We denote with a subscript $n \leq N$ each node of the scenario tree, d_n the demand at that node and π_n the probability of reaching this node from the initial step. Generation units are referred to with a subscript $i \leq Nu$, hence $p^i := (p_1^i, \dots, p_N^i)$ stands for the generation vector of unit i over all the nodes in the tree.

With this notation, our optimization problem can be schematically written as:

$$\left\{ \begin{array}{ll} \min_{\{p^i\}_{i=1, \dots, Nu}} & \sum_{i=1}^{Nu} \sum_{n=1}^N \pi_n C_n^i(p_n^i) \\ \text{s.t.} & \sum_{i=1}^{Nu} p_n^i - d_n = 0, n = 1, \dots, N \quad (\text{coupling constraints of demand}) \\ & p^i \in \mathcal{P}^i \quad (\text{technical constraints for plant } i). \end{array} \right. \quad (4.30)$$

For our application, functions C^i are convex and sets \mathcal{P}^i are convex and compact. We assume furthermore that problem (4.30) satisfies a Slater type condition. Applying price-decomposition on this problem amounts to penalizing the violation of the demand con-

straint. More specifically, let the Lagrangian of problem (4.30) be defined by

$$L(p, x) := \sum_{i=1}^{Nu} \left(\sum_{n=1}^N \pi_n C_n^i(p_n^i) - \langle x, p^i - \frac{d}{Nu} \rangle \right) =: \sum_{i=1}^{Nu} L^i(p, x), \quad (4.31)$$

where $x \in \mathbb{R}^N$ is the vector of Lagrange multipliers. Then, by duality theory (see e.g. [BGLS06, Ch. 8]), we have that

$$(4.30) \Leftrightarrow \min_p \max_x L(p, x) \geq \max_x \min_p L(p, x) \quad (4.32)$$

The right hand side in Eq.(4.32) is the so-called dual problem, concave and possibly non-smooth. The primal problem being convex and well qualified, primal and dual optimal values are the same. Associated with Eq.(4.31) there is the dual function, decomposable by generation units, and defined by

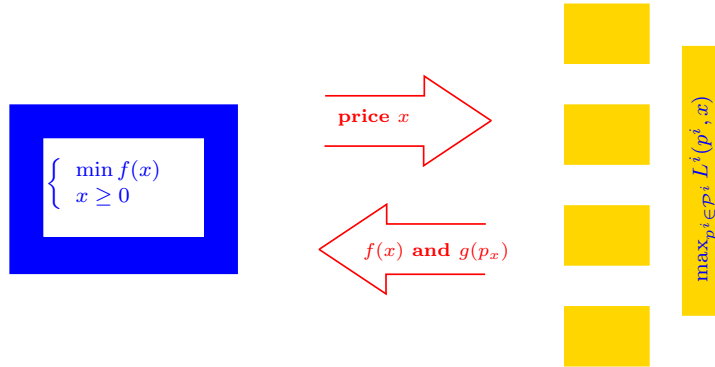
$$\begin{aligned} \theta(x) &:= \min_p L(p, x) = \sum_{i=1}^{Nu} \min_{p^i} L^i(p^i, x) \\ &=: \sum_{i=1}^{Nu} \theta^i(x), \end{aligned} \quad (4.33)$$

where the terms θ^i are the *dual subfunctions*. To place our problem in the more traditional minimization framework, we consider the equivalent problem of minimizing $f(\cdot) := -\theta(\cdot)$. Hence, the dual problem can be rewritten as

$$\begin{cases} \min & f(x) = \sum_{i=1}^{Nu} f^i(x). \\ \text{s.t.} & x \in \mathbb{R}^N \end{cases} \quad (4.34)$$

Price decomposition yields a separable dual function that is easy to compute, because the dual evaluation can be performed separately for each generation unit, as schematically shown in Figure 4.1.

Figure 4.1: Price decomposition



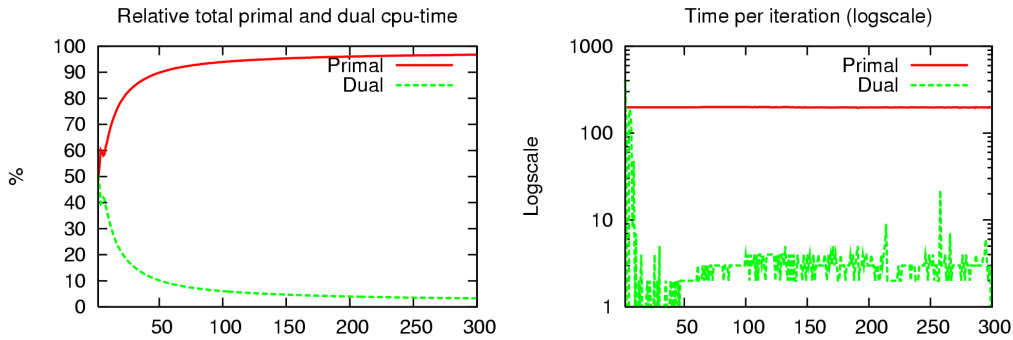
When evaluating each subfunction f^i at some point x , the corresponding primal point \bar{p}_x^i , minimizing $L^i(\cdot, x)$ from Eq.(4.33), may not be unique. Namely, the set of possible primal points is in correspondence with the subdifferential of f :

$$\partial f(x) = \text{conv} \left\{ \sum_{i=1}^{Nu} \bar{p}_x^i \text{ for all } \bar{p}_x^i \text{ such that } f^i(x) = -L^i(\bar{p}_x^i, x) \text{ for } i = 1, \dots, Nu \right\} - \{d\}.$$

In this expression, the symbol conv stands for the convex hull operation. For notational convenience, we use the short notation $g(\bar{p}_x)$ for a subgradient of the form $\sum_{i=1}^{Nu} \bar{p}_x^i - d$, and sometimes we write $g(\bar{p}_x) = \sum_{i=1}^{Nu} g^i(\bar{p}_x^i)$, with $g^i(\bar{p}_x^i) = \bar{p}_x^i - \frac{d}{Nu}$ for $i = 1, \dots, Nu$. In our setting, evaluation of all couples $(f^i(x), g^i(\bar{p}_x)) \in \partial f(x)$ is very costly. As mentioned, an incremental strategy may exploit the separable structure by performing dual steps without fully solving local subproblems, i.e., without having exact values for all the local primal points \bar{p}_x^i , but only some $\bar{p}_x^i \in \mathcal{P}^i$ for $i \in \{1, \dots, Nu\}$.

Mid-term generation planning for EDF's power mix appears as a good candidate to apply the incremental strategy presented in section 4.3.3. Indeed, figure 4.2 illustrates a typical relation between primal and dual CPU times (in red and green lines, respectively), for a generation planning problem with a 1 year horizon over a scenario tree with 20000 nodes, when the bundle method in [LS97] is used for the master program.

Figure 4.2: Evaluation of local subproblems is preponderant.



Primal cpu-times are far more expensive than dual ones during the whole iterative process. In the next section, we present a numerical illustration showing that an incremental strategy may achieve a significant improvement in such setting.

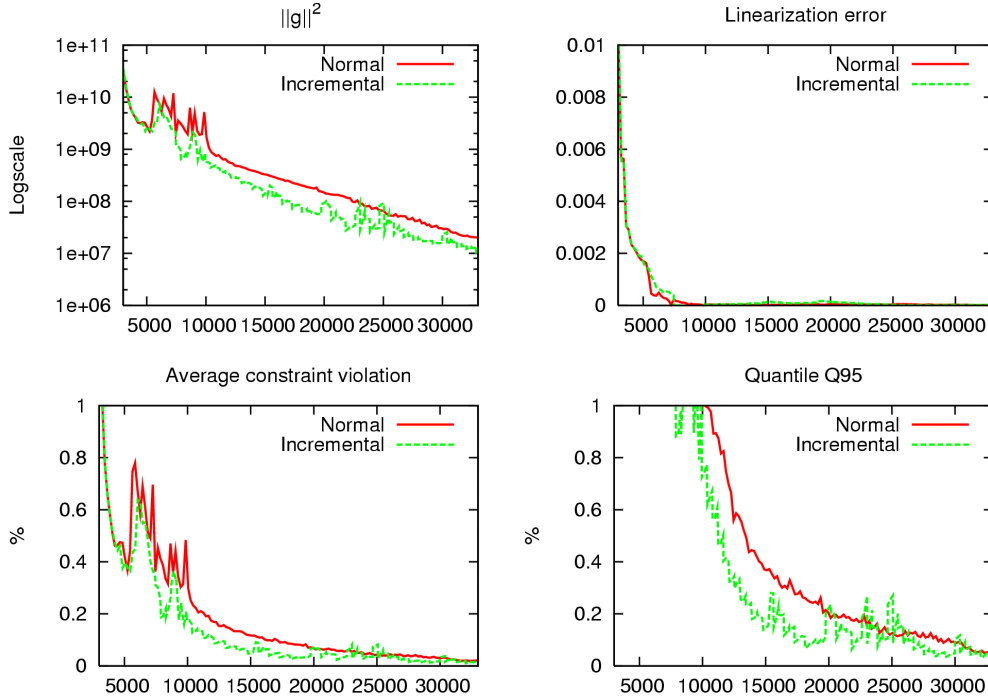
4.4.2 Numerical illustrations

To validate the incremental approach, we applied it on a real dataset with 58 nuclear plants and 85 classical thermal plants and aggregate representation of the spot market. The optimization of generation planning has been made over a one year horizon, with a scenario tree of about 20000 nodes computed out of 484 scenarios. Computations were made on a SUN Sparc, Sun-Fire-V490.

We chose to part the 58 nuclear plants into 3 random subsets. We noticed that performing dual steps with partial information at the very beginning of the procedure yielded rather poor results. Hence, we decided to start the incremental like approach after 20 iterations of the dual master program. Figure 4.3 reports the results on various indicators. For all indicators, CPU time on the horizontal axis is given in clock ticks of the processor.

In the upper part, we plotted the two traditional parameters used in the stopping test for bundle methods. Approximate convergence is achieved when both values reach the corresponding tolerance chosen by the user. We see on these plots that, as usual for bundle methods, the parameter that decreases less easily is the L^2 -norm of the aggregate subgradient. For our application, a tolerance of 10^8 for $|\hat{g}|^2$ is usually chosen. For a same precision, we see the benefits of using an incremental approach, that uses 25% less CPU time.

Figure 4.3: Incremental bundle method



In the lower part of figure 4.3, we plotted two indicators of the overall quality of the approximate primal solution \hat{p} : the average violation of demande constraint and the 95%-th quantile of the distribution of violations over the scenario tree. These values are linked to \hat{g} so, not surprisingly, we see that the incremental approach performs better than the classical bundle method.

4.5 Conclusion

In this work, we introduced a general incremental bundle method for structured non-smooth optimization problems where the objective function is written as the sum of subfunctions. This approach allows to perform dual steps without evaluating all subfunctions. The approach is particularly interesting when the cost of one dual step is negligible when compared to the CPU time required for solving subproblems associated to subfunctions evaluations. Our analysis lies on recent advances regarding bundle methods using inexact linearizations. We followed the developments in [Kiw06] to adapt them in a framework where inaccuracies are unknown and may vary along iterations.

Our first numerical experiments on a real-life problem of mid-term power planning show that we may achieve faster convergence than with a traditional approach. However, more tests still need to be done on other data sets, with e.g. bigger scenario trees. Furthermore, we believe that significant improvements may be obtained with a tighter disaggregation of the bundle and a more clever strategy to chose which subset of subproblems to evaluate.

Bibliography

- [BGLS06] J.F. Bonnans, J.C. Gilbert, C. Lemaréchal, and C. Sagastizábal, *Numerical optimization: Theoretical and practical aspects*, Springer, 2006.
- [BLRS01] L. Bacaud, C. Lemaréchal, A. Renaud, and C. Sagastizábal, *Bundle methods in stochastic optimal power management: a disaggregate approach using preconditioners*, *Comp. Opt. and App.* **20** (2001), no. 3, 227 – 244.
- [ES08] G. Emiel and C. Sagastizábal, *Incremental bundle methods for mid-term power planning*, Proceedings to the International Conference EngOpt – Rio de Janeiro, http://www.engopt.org/en/approved_papers.html (2008).
- [GGM06] M. Gaudioso, G. Giallombardo, and G. Miglionico, *An incremental method for solving convex finite min-max problems*, *Math. Oper. Res.* **31** (2006), no. 1, 173–187.
- [Hin01] M. Hintermüller, *A proximal bundle method based on approximate subgradients*, *Comput. Optim. Appl.* **20** (2001), no. 3, 245–266.
- [HUL96] J.B. Hiriart-Urruty and C. Lemaréchal, *Convex analysis and minimization algorithms 2*, Springer, 1996.
- [Kiw85] K.C. Kiwiel, *An algorithm for nonsmooth convex minimization with errors*, *Mathematics of computation* **45** (1985), no. 171.
- [Kiw95] ———, *Approximations in proximal bundle methods and decomposition of convex programs*, *J. Optim. Theory Appl.* **84** (1995), no. 3, 529–548.
- [Kiw03] ———, *Convergence of approximate and incremental subgradient methods for convex optimization*, *SIAM J. on Optimization* **14** (2003), no. 3, 807 – 840.
- [Kiw06] ———, *A proximal bundle method with approximate subgradient linearizations*, *SIAM J. on Optimization* **16** (2006), no. 4, 1007–1023.
- [KL07] K.C. Kiwiel and C. Lemaréchal, *An inexact bundle variant suited to column generation*, *Math. Program. Ser. A.* (2007).
- [LS97] C. Lemaréchal and C.A. Sagastizábal, *Practical aspects of the Moreau-Yosida regularization: theoretical preliminaries*, *SIAM J. on Optimization* **7** (1997), no. 2, 367 – 385.
- [NB01] A. Nedič and D. Bertsekas, *Incremental subgradient methods for nondifferentiable optimization*, *SIAM J. on Optimization* **12** (2001), no. 1, 109–138.
- [Sol03] M.V. Solodov, *On approximations with finite precision in bundle methods for nonsmooth optimization*, *J. Optim. Theory Appl.* **119** (2003), no. 1, 151 – 165.

- [SZ98] M.V. Solodov and S.K. Zavriev, *Error stability properties of generalized gradient-type algorithms*, J. Optim. Theory Appl. **98** (1998), no. 663 - 680.

Chapitre 5

Gestion à moyen terme de l'électricité

Sommaire

5.1	Introduction et résumé	83
5.2	La gestion de production	84
5.2.1	Description du portefeuille	85
5.2.2	Coût marginal et valeur d'usage	87
5.2.3	Différents horizons de temps	87
5.2.4	GMT : un problème stochastique multi-périodes de grande taille	90
5.3	Présentation d'un outil de GMT : SOPRANO	94
5.3.1	Décomposition par les prix sur un arbre d'aléas	95
5.3.2	Description des sous-problèmes	97
5.3.3	Jeux de données utilisés	99
5.4	Quelques faits stylisés	100
5.4.1	Obtention de solutions primales approchées	100
5.4.2	Sélection et compression du faisceau	103
5.5	Calcul de stratégies pour la simulation	104
5.5.1	Valeur d'usage d'un stock d'énergie	104
5.5.2	Valorisation face à un signal de prix	104
5.5.3	Valorisation à production complémentaire fixée	105
5.5.4	Résultats numériques	106
5.5.5	Test des stratégies en simulation	106
5.6	Apports d'une méthode dynamique de faisceaux	108
5.7	Conclusion	110
	Bibliographie	115

5.1 Introduction et résumé

L'électricité ne se stocke pas, ce qui en fait un produit à part pour lequel l'offre et la demande doivent être, à chaque instant, parfaitement équilibrées. Ceci nécessite une gestion coordonnée, depuis les décisions d'investissements à l'horizon pluriannuel jusqu'aux ajustements journaliers permettant de réaliser effectivement cet équilibre. Pour réaliser au meilleur coût cet équilibre, il existe différents leviers et les décisions et acteurs concernés dépendent beaucoup de l'horizon de gestion considéré. Par conséquent, une

approche traditionnelle consiste à considérer des problèmes plus simples, restreints à un horizon de temps particulier. La gestion de production est ainsi souvent divisée en gestion de long-terme, de moyen-terme et de court-terme.

Cependant, compte tenu de la taille et de la complexité du parc de production d'EDF, quel que soit l'horizon de temps considéré, le problème à résoudre est un problème de très grande taille. Les méthodes de décomposition permettent de résoudre ce type de problèmes en appliquant la devise "divide ut regnes"^{*}. Le problème initial est remplacé par une séquence de sous-problèmes beaucoup plus simples, *coordonnés* par un *programme maître* chargé d'ajuster les paramètres de la décomposition afin de s'approcher de la solution du problème initial. Dans de nombreux schémas de décomposition, notamment la décomposition dite *par les prix* autour de laquelle nous avons articulé notre travail, le programme maître est un problème d'optimisation non-différentiable. La phase de coordination doit donc faire appel à des algorithmes spécialisés.

Dans ce chapitre, nous décrivons dans un premier temps le problème de la gestion de production d'électricité, en insistant sur la gestion à moyen-terme (GMT), sur laquelle se sont concentrées nos expérimentations numériques.

Nous présentons ensuite un outil de gestion moyen-terme de EDF R&D, SOPRANO, qui propose de résoudre le problème en appliquant une technique de décomposition par les prix sur un arbre de scénarios : la contrainte de demande est dualisée et la solution optimale du problème dual fournit les prix marginaux du système production/consommation sur l'arbre.

Parallèlement à nos travaux théoriques, nous avons mené un audit du code de coordination duale utilisé par ce logiciel. Il s'agit d'une méthode de faisceau proximale avec une mise à jour du paramètre proximal basée sur la théorie de quasi-Newton comme proposé par [LS97]. Nous avons implémenté plusieurs enrichissements ainsi que quelques corrections portant notamment sur le test d'arrêt, l'obtention de solutions primales ou la compression du faisceau.

Une fois le problème de gestion moyen terme (GMT) résolu sur un arbre, une heuristique permet de générer des stratégies de gestion locales, pour chaque réserve (hydraulique, nucléaire, contrat d'effacement). Cette heuristique repose sur une programmation dynamique locale (i.e. pour une réserve isolée), face au signal de prix constitué des variables duales optimales. Pour réaliser cette valorisation, nous avons proposé une alternative qui utilise des solutions primales optimales approchées, également disponibles une fois le problème GMT résolu, en lieu et place de ce signal de prix.

Enfin, nous avons testé l'apport éventuel d'une relaxation Lagrangienne dynamique (cf. chapitre 3) afin de réduire la dimension de l'espace dual et du problème quadratique résolu lors des premières étapes de la coordination. D'autres expérimentations portant sur une coordination incrémentale avec un algorithme de faisceau adapté ont été présentées au chapitre 4.

5.2 La gestion de production

Schématiquement, la gestion de production a pour objectif de satisfaire la demande d'électricité à moindre coût. Pour cela, le portefeuille d'actifs à disposition doit être appréhendé dans sa globalité. Il comporte des actifs physiques, constitués des groupes de production gérés par le producteur et des actifs financiers regroupant les contrats existants pour l'achat ou la vente d'énergie électrique. Pour un groupe comme EDF, la part des actifs physiques dans le portefeuille est fortement prédominante.

D'un point de vue général, tout problème de gestion de production peut s'écrire comme un problème d'optimisation : pour un horizon de temps et un portefeuille d'actifs

^{*} "Diviser pour régner."

donnés, on cherche à satisfaire à moindre coût la demande d'électricité, tout en respectant différentes contraintes de fonctionnement. Il s'agit à priori d'un problème d'optimisation stochastique, puisque de nombreux paramètres ne sont pas connus de façon certaine.

En environnement déterministe, en considérant un portefeuille composé de Nu actifs sur un horizon de temps décomposé en T périodes, un problème de gestion de production simple peut généralement s'écrire

$$\left\{ \begin{array}{l} \min_{\{p^i\}_{i=1, \dots, Nu}} \sum_{t=1}^T \sum_{i=1}^{Nu} C_t^i(p_t^i, x_t^i) \\ p_t^i \in \mathcal{P}_t^i \\ x_t^i \in \mathcal{X}_t^i \\ (p^i, x^i) := ((p_1^i, x_1^i) : t = 1 \dots T) \in \mathcal{D}^i \\ (p_t, x_t) := ((p_1^i, x_1^i) : i = 1 \dots Nu) \in \mathcal{S}_t \end{array} \right. \quad (5.1)$$

où C_t^i est la fonction de coût de l'usine i à la date t , p_t^i la variable de commande (e.g. production, marche/arrêt) et x_t^i représente son état (volume du stock hydraulique, durée de l'arrêt en cours, etc.). L'ensemble \mathcal{P}_t^i représente les contraintes de fonctionnement statiques (e.g. contrainte de puissance maximale) et \mathcal{X}_t^i les contraintes sur l'état ($x_t^i \in \{0, 1\}$, par exemple). Dans ce cadre très simple, sans prendre en compte les contraintes \mathcal{D}^i et \mathcal{S}_t , le problème est séparable à la fois par usine et par pas de temps et pourrait donc être résolu séquentiellement usine par usine et période par période.

Une modélisation plus fine doit cependant être utilisée en pratique, afin de prendre en compte les contraintes dynamiques propres à chaque usine et les contraintes couplant les différents éléments du système de production. Les premières sont représentées par l'ensemble \mathcal{D}^i . Il s'agit par exemple de l'équation de conservation de la matière liant l'énergie turbinée au volume du réservoir hydraulique, ou de la durée minimale d'un arrêt avant de pouvoir disposer à nouveau d'une usine thermique. Ces contraintes introduisent un couplage temporel sur le problème à résoudre. Il n'est donc plus question de résoudre séquentiellement des sous-problèmes sur une période, sauf à faire appel à des techniques de décomposition appropriées (programmation dynamique).

Enfin, tout problème de gestion de production présente également un couplage spatial des différents moyens de production, introduit par les contraintes \mathcal{S}_t . La contrainte de satisfaction de la demande couple entre eux les différents moyens de production, empêchant une résolution séquentielle usine par usine. D'autres contraintes couplantes peuvent apparaître si l'on considère une modélisation plus détaillée du système :

- les contraintes dites *de réserves*, qui imposent de disposer d'une réserve de puissance disponible à la hausse en un temps maximal donné pour faire face à une défaillance imprévue d'un élément du système ;
- les contraintes de réseau afin de prendre notamment en compte les pertes ou les limites de transmission ;
- les contraintes environnementales limitant la quantité de CO2 émise par le système.

Enfin, dans de nombreux cas cette modélisation déterministe n'est pas suffisante et il convient de prendre en compte les nombreuses incertitudes qui pèsent sur les paramètres du système, comme par exemple :

- la demande, qui est fortement corrélée aux aléas climatiques en France et n'est donc pas parfaitement prévisible pour un horizon temporel de plus d'une journée ;
- la pluviométrie, qui permet le remplissage des réserves hydrauliques.

5.2.1 Description du portefeuille

Le portefeuille de production d'EDF regroupe des actifs physiques et financiers, les premiers étant largement prépondérants. Comme nous le verrons, la modélisation de ces actifs varie en fonction de l'horizon de temps retenu.

Les actifs physiques regroupent les centrales hydrauliques, thermiques, ainsi que marginalement les moyens de production basés sur des énergies renouvelables. Le tableau 5.2.1, montre la répartition des moyens de production pour le groupe EDF, avec la prédominance du nucléaire.

	France	Monde
Nucléaire	85.7%	70.5%
Thermique à flamme	8.1%	21.2%
Hydraulique	5.0%	8%
Autres	0.9%	

TAB. 5.1 – Répartition des moyens de production (2005)

Chaque actif dispose de contraintes de fonctionnement qui lui sont propres.

Pour les centrales hydrauliques, les contraintes font intervenir les interconnexions entre réservoirs, les rendements des turbines en fonction de la hauteur de chute, etc. Les coûts de production sont très faibles et pourraient être négligés en considérant que le combustible utilisé est gratuit car fournit par les apports hydrauliques naturels. Toutefois, ce point de vue ne rend pas compte des économies potentiellement réalisées dans le futur en stockant ces apports saisonniers pour les utiliser aux moments opportuns. La gestion optimale du portefeuille de production passe par la juste valorisation de cette source d'énergie. On nomme couramment *valeur d'usage* le prix associé à cette valorisation optimale : elle peut être assimilée au coût marginal de production d'un Kwh par ce stock. Nous verrons à la section 5.2.2 que les valeurs d'usage des stocks d'énergie sont à la fois un objectif et un outil de la gestion de production en fonction de l'horizon d'étude considéré.

Pour les centrales thermiques classiques, les principales contraintes de fonctionnement sont, par exemple, les durées minimales de fonctionnement ou d'arrêt, les paliers de production, etc. La fonction de coût est spécifique à chaque usine. Elle comprend les coûts de démarrage, qui peuvent dépendre de la durée du dernier arrêt, et les coûts de production liés aux coûts de combustible, éventuellement non linéaires.

Les centrales thermiques nucléaires diffèrent assez largement des centrales conventionnelles en raison des contraintes sur les stocks de combustible qui ne peuvent être rechargés que lors des périodes d'arrêt du réacteur. Ces arrêts ont une fréquence très faible (1 tous les 18 mois) et l'évolution du stock de combustible est une donnée pouvant s'avérer importante pour la gestion pluriannuelle. De même que pour l'hydraulique, les coûts de production doivent prendre en compte une juste valorisation du stock de combustible.

Le portefeuille de production comprend également des éléments sur lesquels le producteur n'a pas la main, comme les usines hydrauliques au fil de l'eau ou les éoliennes. La production basée sur les énergies renouvelables est encore très marginale en France, mais elle est amenée à jouer un rôle important dans le futur en raison du renchérissement des combustibles fossiles et de l'introduction de contraintes environnementales.

Les actifs financiers regroupent à la fois les produits pouvant être achetés ou vendus sur les marchés spot ou à terme de l'électricité, et les options tarifaires d'effacement de la courbe de charge. Ces options sont des clauses contractuelles passées avec des clients afin de permettre une double tarification : plus faible que la moyenne en temps normal, mais très élevée certains jours. Les jours d'application de cette option de tarification sont à la main du producteur et permettent de faire baisser la demande sur certaines journées particulièrement tendues, les clients concernés préférant réduire leur consommation plutôt que de payer un prix très élevé. Toutefois, le nombre de jours d'effacement est limité et ne correspond qu'à la période hivernale de forte demande.

5.2.2 Coût marginal et valeur d'usage

Le *coût marginal* du système correspond au coût de production d'un kWh supplémentaire. Plus précisément, c'est la variation du coût global de production optimisé sur toute la période de gestion, due à une production supplémentaire de 1 kWh à un instant donné. Nous verrons à la section 5.3.1 que l'on peut retrouver cette valeur via la résolution du problème dual obtenu par relaxation de la contrainte de demande.

En simplifiant considérablement le problème, on peut supposer que chaque groupe de production est caractérisé par un coût proportionnel constant et qu'aucune dynamique de stocks n'est à prendre en compte. La gestion optimale du parc consiste alors à *empiler* les moyens de production par coût proportionnel croissant jusqu'à satisfaction de la demande. Le coût marginal du système correspond dans ce cas au coût proportionnel du groupe le moins cher disponible à la hausse.

Pour GMT, les centrales thermiques classiques sont effectivement modélisées selon ces hypothèses très simples. Toutefois, les actifs devant satisfaire des contraintes de stocks doivent faire l'objet d'une attention particulière. Il s'agit naturellement des groupes hydrauliques, mais aussi des groupes nucléaires et des contrats tarifaires (nombre limité d'effacements sur une année). Le coût d'un tel actif doit intégrer le gain futur espéré lié à l'économie du stock aujourd'hui. En effet, le fait de garder en stock une certaine quantité d'énergie diminue le risque d'être contraint d'utiliser des groupes de production très chers pour satisfaire la demande lors d'un pic de consommation. Inversement, si les stocks d'énergie sont maintenus de manière systématique à un niveau trop élevé, on se force à utiliser des moyens de production chers alors qu'on aurait pu s'en passer. Des gaspillages contraints peuvent aussi intervenir pour les stocks hydrauliques si les apports sont trop importants pour le volume du réservoir.

Il faut donc solliciter ces stocks aux meilleurs moments et pour cela, comparer entre une utilisation immédiate qui permet d'économiser le coût marginal, ou une utilisation différée. On économise alors la *valeur d'usage* du stock qui est le gain futur moyen rapporté par une unité de stock.

Le calcul des valeurs d'usage des stocks est l'un des principaux enjeux de la gestion moyen terme de la production. Elle permet de simuler sur un ensemble de scénarios la coordination entre les divers moyens composant le parc de production, afin de fournir des indicateurs de gestion et des indicateurs financiers. C'est également une information importante pour la gestion à court terme.

5.2.3 Différents horizons de temps

Les décisions associées à la gestion de production sont très variées et concernent des périodes de temps très différentes. Schématiquement, les décisions d'investissement se prennent aujourd'hui pour dans dix ans, les arrêts de centrales nucléaires se planifient aujourd'hui pour les 5 prochaines années, et les arrêts de centrales thermiques classiques se décident du jour au lendemain.

Il n'est pas envisageable de résoudre le problème dans sa globalité pour l'ensemble des horizons d'étude. Une approche couramment retenue consiste à décomposer le problème en sous-problèmes spécifiques aux différents horizons de temps. On retrouve généralement un découpage en 3 parties : la gestion de production de long terme, de moyen terme et de court terme.

Pour le long terme, les décisions portent avant tout sur la structure du portefeuille et sont prises sur des critères financiers, alors qu'à très court terme on s'intéresse principalement à l'équilibre offre-demande en concertation avec le gestionnaire de réseau.

Les indicateurs calculés pour un horizon de temps donné sont ensuite utilisés comme donnée d'entrée pour la gestion sur un horizon différent. Cette transmission d'information se fait avant tout du long terme vers le court terme. Ainsi, les arrêts des centrales nucléaires

servent comme paramètre à la gestion des réserves (hydrauliques, nucléaires, effacements). De même, la valorisation des énergies en réserve (valeurs d'usage) est transmise aux gestionnaires du court terme.

En fonction de l'horizon de temps considéré, la modélisation retenue varie énormément. Pour le court terme, il est nécessaire de prendre en compte de manière précise l'ensemble des contraintes de fonctionnement du parc. En effet, les plannings de production calculés ont alors vocation à être effectivement appliqués, la modélisation doit donc être fidèle. Le problème d'optimisation considéré comporte de nombreuses variables entières et un très grand nombre de contraintes. Cependant, à l'horizon journalier, la plupart des paramètres du système peuvent être considérés comme déterministes. A l'opposé, lorsque l'horizon d'étude s'étend, la modélisation déterministe ne peut plus être adoptée, mais les éléments du parc peuvent être modélisés de façon plus grossière. Le graphique 5.1 reprend cette opposition entre la structure d'information et la représentation physique du parc, en fonction de l'horizon d'étude.

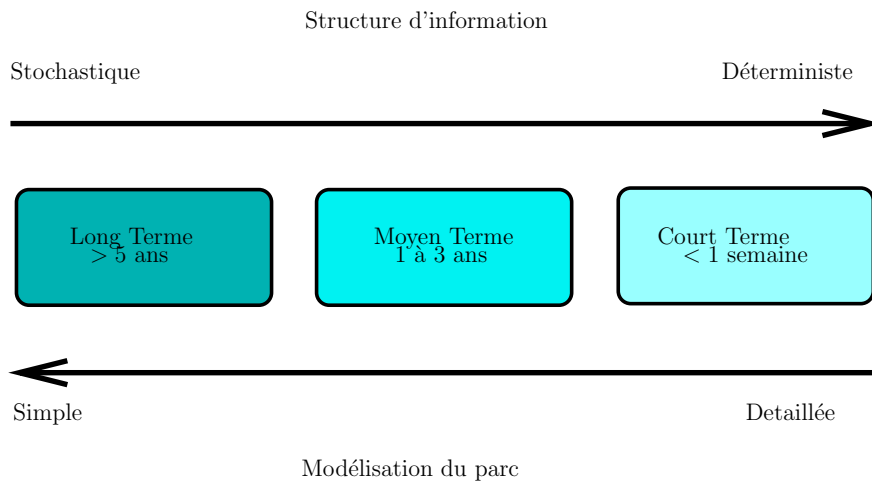


FIG. 5.1 – Gestion de production

Nous décrivons dans ce qui suit les principales caractéristiques de ces 3 grands domaines, tels qu'ils sont traités à EDF et chez la plupart des électriciens, en insistant sur la gestion moyen terme sur laquelle se sont concentrées nos applications numériques.

Gestion court-terme

La gestion court terme regroupe la gestion hebdomadaire, à caractère prévisionnel, et la gestion journalière et infra-journalière qui ont vocation à fournir des commandes réalisables pour les opérationnels.

- La gestion hebdomadaire (7 à 12 jours) s'intéresse aux placements des maintenances des centrales, aux couvertures à court-terme et aux propositions de déclenchements d'options tarifaires. Des études récentes ont montré l'intérêt de considérer une approche stochastique pour cet horizon de temps sur le parc EDF. [App08] propose une approche basée sur la programmation robuste, avec des règles de décision linéaires pour optimiser la gestion d'une vallée hydraulique à l'horizon hebdomadaire en environnement stochastique. La décomposition par les prix sur un arbre de scénarios a également été appliquée dans [BLRS01, ?].
- Au journalier (1 à 2 jours), on détermine le programme de marche à transmettre au régulateur, les arrêts et démarrages de centrales thermiques, et un éventuel

déclenchement d'options tarifaires. Une modélisation déterministe est généralement utilisée, avec toutefois une représentation fidèle de l'ensemble du parc de production. Une revue de littérature récente et très complète sur les modélisations retenues et les méthodologies de résolution est proposée par [Din07].

- Gestion infra-journalière : les plannings proposés par la gestion journalière peuvent être amenés à évoluer au cours d'une journée en fonction des événements fortuits constatés sur le parc ou sur demande du régulateur.

Gestion moyen-terme (GMT)

Contrairement à la gestion court terme, la gestion moyen terme n'a pas vocation à fournir des plannings de productions réalisables. Son objectif est d'établir des politiques de gestion robustes face aux aléas pesant sur les paramètres du système. En France, il s'agit en particulier du niveau de la demande, fortement corrélée à d'autres variables comme la température ou la nébulosité. Parmi les autres paramètres stochastiques, on trouve les apports hydrauliques, les indisponibilités fortuites des centrales ou les prix de marché de l'électricité et des combustibles fossiles.

La modélisation des différents actifs de production est relativement simplifiée. Par exemple, dans le cas de la France, les réserves hydrauliques sont agrégées en réservoirs équivalents, en fonction de leur constante de temps (durée de remplissage ou de vidange). Les interconnexions entre lacs ne sont donc pas prises en compte et il en est de même pour les variations du rendement en fonction de la hauteur de chute. Pour les centrales thermiques classiques, on se ramène généralement à des usines de coût proportionnel constant, dont les seules contraintes sont les bornes sur la puissance. Souvent, une approche similaire est employée pour les centrales thermiques nucléaires mais la tendance actuelle est d'utiliser une modélisation plus fine. En effet, si les centrales classiques peuvent être rechargées à tout moment, le combustible nucléaire ne peut être renouvelé que lors des périodes d'arrêt du réacteur, ce qui n'intervient en moyenne que tous les 18 mois. Lors d'un cycle de fonctionnement, la puissance maximale fournie dépend du niveau du stock, en particulier en fin de cycle. Par conséquent, il est important de prendre en compte l'évolution du stock de combustible comme cela est naturellement fait pour une usine hydraulique. Ce raffinement dans la modélisation est d'autant plus important que la production électrique française est très majoritairement d'origine nucléaire.

La nécessité de prendre en compte les aléas et les dimensions tant spatiale (nombre de groupes) que temporelle (sur 5 ans) imposent de décomposer le problème en trois sous-problèmes qui sont traités successivement.

1. *Placement des arrêts* des centrales nucléaires : sur un horizon long de 5 ans le planning des arrêts doit satisfaire un certain nombre de contraintes de ressources qui limitent le nombre de maintenances simultanées. Le problème à traiter est combinatoire et de grande taille. Nous nous référons à [KPB06] pour une présentation précise de ce problème à EDF et de méthodes qui peuvent être utilisées pour le résoudre (programmation par contraintes).
2. Calcul des valeurs d'usage des réserves agrégées (*phase d'optimisation*) : l'objectif est ici de calculer sur un horizon de 2 à 3 ans une stratégie de gestion optimale des différentes réserves (hydrauliques, nucléaires, effacements) face aux aléas qui affectent le système offre/demande. Une valeur d'usage est calculée pour chaque réserve en fonction de la date, du niveau de la réserve considérée et éventuellement d'autres variables d'état. Ces valeurs d'usage permettront, dans la phase de simulation, d'arbitrer entre l'utilisation immédiate ou ultérieure des réserves.
3. Simulation de la gestion coordonnée des réserves face aux aléas (*phase de simulation*) : sur un ensemble de scénarios d'aléas, on procède à la coordination entre les divers actifs composant le parc de production, afin de fournir des indicateurs de

gestion comme les coûts marginaux, des indicateurs financiers concernant la marge du groupe, ou des bilans énergétiques. Cela est réalisé en connaissant la valorisation des réserves définie à l'étape précédente et traduite à l'aide des valeurs d'usage.

La méthodologie de résolution s'apparente donc à la recherche d'une stratégie de gestion en *boucle ouverte*, établie une fois pour toutes avant d'entreprendre la simulation. En pratique toutefois, le processus est réitéré chaque semaine pour fournir des valeurs d'usage et chroniques de coûts marginaux aux gestionnaires, implémentant ainsi dans les faits un calcul de stratégie avec rebouclage.

Gestion long-terme

La gestion long terme a pour objectif d'optimiser la structure des actifs physiques du portefeuille. Il s'agit donc de décider des investissements dans de nouveaux moyens de production en prévision de la hausse de la consommation ou du vieillissement des actifs existants. Une description de méthodes de valorisation d'actifs utilisés pour la prise de décision avec des applications au contexte d'EDF est présentée dans [Por08].

5.2.4 GMT : un problème stochastique multi-périodes de grande taille

Le problème traité au moyen terme est un problème d'optimisation stochastique. En effet, de nombreux paramètres sont très incertains sur un horizon de temps de 2 ans : la demande, les apports hydrauliques, l'indisponibilité non programmée de certains groupes thermiques, les prix de marché. La résolution d'un tel problème, compte tenu des dimensions spatiales (une centaine d'usines) et temporelles (3 pas de temps par jour sur 1 à 3 ans) est un véritable défi méthodologique et numérique.

Le formalisme retenu est du type *wait and see* ([BL97]) : pour chaque période t , une décision p_t est prise pour un état du système v_t et pour une réalisation de l'aléa ξ_t connue. Pour la première période, l'information est donc entièrement disponible. Pour les périodes suivantes, l'histoire du processus aléatoire (ξ_1, \dots, ξ_t) est entièrement connue lorsque la commande p_t doit être décidée.

Pour GMT, les commandes correspondent aux productions et l'état du système aux niveaux des réserves. Les contraintes regroupent des contraintes *statiques* sur le fonctionnement des centrales, *dynamiques* avec les contraintes de flot sur les stocks, et la contrainte *couplante* de satisfaction de la demande à chaque date.

On note $C_t(p_t, v_t, \xi_t)$ la fonction de coût du système à la date t pour une commande p_t , un état v_t et l'aléa ξ_t . Le processus aléatoire $\{\xi_t\}$ est défini sur un espace Ω_t muni d'une probabilité \mathbb{P} et on lui associe sa filtration naturelle \mathcal{F}_t . On note $\mathbb{E}_{\mathcal{F}_t}$ l'espérance mathématique conditionnelle à la réalisation de $\{\xi_1 \dots \xi_t\}$.

Le problème de GMT consiste à chercher à chaque date t la commande minimisant l'espérance des coûts de production futurs sachant le passé, en satisfaisant les contraintes du système production/consommation. Le problème (5.1) devient donc :

$$\left\{ \begin{array}{l} \min_{\{p^i\}_{i=1, \dots, Nu}} \mathbb{E}_{\mathcal{F}_1} \left\{ \sum_{t=1}^T \sum_{i=1}^{Nu} C_t^i(p_t^i, v_t^i, \xi_t) \right\} \\ p_t^i \in \mathcal{P}_t^i(\xi_t) \\ v_t^i \in \mathcal{X}_t^i(\xi_t) \\ (p^i, v^i) \in \mathcal{D}^i(\xi) \\ (p_t, x_t) \in \mathcal{S}_t(\xi_t) \end{array} \right. \quad (5.2)$$

Les commandes p_t recherchées sont désormais des fonctions de l'aléa. Compte tenu du formalisme *wait and see*, elles doivent être \mathcal{F}_t -mesurables. Cette contrainte de mesurabilité sur les commandes est aussi appelée contrainte de non-anticipativité lorsque l'espace

d'aléas est de cardinalité finie comme nous le verrons plus tard. En appliquant la règle des espérances itérées (voire e.g. [JP04]) et en omettant les contraintes de fonctionnement pour alléger la notation, le problème à résoudre peut également s'écrire

$$\min_{(p_1, v_1)} \left\{ C_1(p_1, v_1) + \mathbb{E}_{\mathcal{F}_1} \left\{ \min_{(p_2, v_2)} C_2(p_2, v_2, \xi_2) + \mathbb{E}_{\mathcal{F}_2} \{ \dots \} \right\} \right\}. \quad (5.3)$$

Programmation dynamique stochastique Une manière d'aborder un tel problème est d'utiliser la programmation dynamique stochastique : sous l'hypothèse d'indépendance temporelle des aléas, le calcul de la commande optimale peut se faire à l'aide de l'équation rétrograde de Bellman [BD59]. En notant $\alpha_t(v_t)$ la fonction donnant le coût de gestion optimal sur l'horizon $[t, T]$ pour le vecteur d'état v_t , cette équation s'écrit :

$$\alpha_t(v_t) = \min_{(p_t, v_{t+1})} C_t(p_t, v_t, \xi_t) + \alpha_{t+1}(v_{t+1}).$$

Le schéma standard de résolution consiste alors à discrétiser l'espace d'état et à résoudre le problème en partant de la date finale sur une grille représentant l'ensemble des états possibles du système. Cette procédure s'apparente donc à une décomposition temporelle pour se limiter à la résolution de sous-problèmes sur une période et repose sur l'approximation d'un continuum d'états en un ensemble discret. Enfin, sauf cas particuliers, le calcul des espérances mathématiques passe également par une approximation et une discrétisation du processus stochastique sous-jacent (e.g. par simulation de Monte Carlo).

Toutefois, lorsque la dimension du vecteur d'état est grande (5.3) un tel schéma n'est pas envisageable. Explorer tous les points d'une grille en dimension élevée, même avec une discrétisation grossière, est inenvisageable. De plus, en pratique les aléas sont autocorrélés et cette approche ne peut pas être appliquée sans une manipulation consistant à inclure la mémoire du processus dans la description de l'état, ce qui augmente à nouveau la dimension du problème.

Plusieurs approches tentent d'adapter le schéma de la programmation dynamique stochastique pour remédier à la "malédiction de la dimension". Par exemple, l'approche actuellement utilisée pour résoudre GMT à EDF consiste à résoudre le problème en ne considérant qu'une *réserve principale* face à une agrégation de toutes les autres réserves regroupées sous le nom de *réserve complémentaire* ([LT85, Tur80]). La dimension du vecteur d'état peut ainsi être maintenue raisonnable tout en prenant en compte une modélisation réaliste des aléas. Toutefois, l'outil en question n'est utilisé que pour la valorisation des réserves hydrauliques agrégées en trois stocks, sans contraintes de stocks sur les groupes nucléaires.

Programmation dynamique sans discrétisation de l'état. Les limites d'application de la programmation dynamique sont dues à la discrétisation préalable de l'espace d'état. Afin de s'affranchir de ce problème, une possibilité consiste à choisir heuristiquement quelques points particuliers où les fonctions de Bellman seront approchées.

La programmation dynamique stochastique duale [PP91, CP99, DB06, PG08] repose sur ce principe et permet de considérer un espace d'état de grande dimension. Une itération est constituée d'une phase progressive de simulation qui permet de sélectionner un échantillon de points de stocks "intéressants", et d'une phase rétrograde qui utilise le principe de Bellman et la dualité pour modéliser les fonctions $\alpha_t(\cdot)$ par des modèles de plans coupants composés de coupes de Benders ([Ben62]). Ce schéma *forward/backward* est répété jusqu'à ce qu'un test d'arrêt valide la qualité de la solution approchée ainsi obtenue. Cette méthode connaît un important succès en gestion de production avec des implémentations opérationnelles chez plusieurs électriciens (e.g. [PCK99, RG92]).

Toutefois, cette approche connaît certaines limites. En effet, bien qu'une représentation riche des aléas soit possible il est nécessaire de procéder à une discrétisation a priori du

processus aléatoire. De plus, l'utilisation de la dualité suppose que les fonctions α_t soient convexes en le vecteur d'état. Pour des raisons de perte de convexité, il n'est notamment pas possible de prendre en compte l'autocorrélation d'un aléa présent dans la fonction objectif. Enfin, des tests numériques ([Emi05]) ont montré que le nombre de pas de temps considéré pouvait être un frein à la convergence de la méthode. En effet, lors de la phase rétrograde, les erreurs minorantes apportées par l'approximations des vraies fonctions α_t par des modèles de plans coupants s'additionnent. A la date initiale, la fonction α_1 a ainsi tendance à être fortement minorante. Une solution à ce problème peut être d'agréger les pas de temps par paquets en résolvant ainsi des sous-problèmes un peu plus complexes car multi-périodiques ([DS08]).

La méthode d'agrégation des stocks et la programmation dynamique duale peuvent s'interpréter comme des implémentations particulières de la programmation dynamique approximée [BD59]. Celle-ci a pour objectif d'approcher les fonctions de Bellman à partir d'une sous-famille de fonctions (fonctions linéaires, polynomiales, etc.) en utilisant l'équation de la programmation dynamique. Cette méthode n'implique pas de discrétisation a priori de l'espace d'état, mais repose sur une discrétisation de l'espace des fonctions utilisées pour approcher les fonctions de Bellman. Des expérimentations ont été menées pour appliquer ces méthodes sur le problème GMT à EDF [BGRS06a].

Enfin, les méthodes d'apprentissage sont une autre approche visant à bénéficier du principe de la programmation dynamique sans discrétisation a priori de l'espace d'état. L'idée est d'estimer fonctionnellement et de façon non-paramétrique les fonctions de Bellman. Les auteurs de [BGRS08, BGRS06b] proposent une approche basée sur des fonctions-noyau pour approcher les fonctions de Bellman en effectuant des tirages de Monte-Carlo et en testant progressivement les différentes stratégies de gestion possibles.

Pour toutes ces approches, une fois le problème résolu on obtient des *stratégies de gestion*, matérialisées par des fonctions de Bellman. Elles permettent de simuler la gestion coordonnée du parc de production face à un scénario d'aléa quelconque.

Programmation stochastique sur arbre et décomposition Comme alternative à la décomposition temporelle de la programmation dynamique, une approche répandue consiste à approcher le processus stochastique $\{\xi_t\}$ par un échantillon fini $\{\xi_t^i\}_{i=1\dots m}$, obtenu par exemple par tirage de Monte Carlo. Pour conserver une bonne représentativité et garantir que la solution du problème discrétisé soit proche de la vraie solution, les échantillons doivent être suffisamment fournis. Des résultats de Shapiro (e.g. [Sha03, Ch.6]) basés sur la théorie des grandes déviations ([DZ98]) donnent une borne supérieure pour la taille m de l'échantillon afin d'obtenir une solution avec une tolérance donnée. Cependant, en pratique cette borne est souvent inexploitable en multi-périodes, car très élevée. De plus, cela suppose de connaître la loi régissant le processus stochastique ou de disposer d'une bonne approximation.

La méthodologie généralement retenue consiste à exploiter un ensemble de scénarios d'évolutions probables des aléas en construisant à partir de ces données une structure arborescente : les scénarios proches selon une norme à définir sont agrégés partiellement afin de se ramener à une structure arborescente.

Les techniques de réduction de scénario ([HR03]) visent à garantir une taille raisonnable pour l'arbre généré tout en gardant une bonne représentativité de l'aléa et la stabilité de la solution ([Str04, HRS06]).

Un arbre compte autant de scénarios que de nœuds finaux (5 sur l'arbre de la figure 5.2). Ces scénarios partent d'un même point initial et divergent progressivement au fil du temps. Pour une date donnée, des scénarios non dissociables doivent être associés à des commandes identiques. C'est ce que l'on appelle contrainte de non-anticipativité. Il s'agit d'une forme particulière de discrétisation de la contrainte de mesurabilité ([Bar04]) : les commandes sont prises connaissant uniquement le passé. Les méthodes particulières

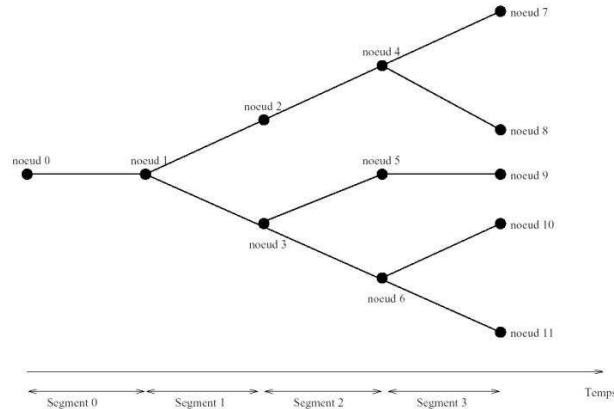


FIG. 5.2 – Arbre de scenarios

([Dal06]) sont une forme alternative de discrétisation de la structure d'information à l'aide de techniques d'approximation de l'espérance conditionnelle.

Compte tenu de la complexité du problème GMT, le problème à résoudre sur l'arbre de scénarios reste difficile et des méthodes de décomposition doivent malgré tout être employées. En plus de la *décomposition temporelle* des variantes de la programmation dynamique présentée précédemment, on peut également décomposer le problème spatialement, ou scénario par scénario. Pour une présentation générale des méthodes de décomposition pour l'optimisation stochastique, nous nous référons à [Rus03, Ch.3].

La *décomposition spatiale* est possible par relaxation Lagrangienne des contraintes couplant entre eux les moyens de production à chaque date. Il s'agit typiquement de la contrainte de demande, ou des contraintes de réserve de puissance. Le problème obtenu, dit dual, est séparable usine par usine et sa résolution est ainsi simplifiée. Nous reviendrons sur cette technique à la section 5.3.1.

Lorsque le problème initial n'est pas convexe (en particulier pour la gestion à court terme), cette approche ne fournit pas une solution primale réalisable et la valeur optimale du problème dual peut être strictement inférieure à celle du problème initial. Des techniques de Lagrangien augmenté ([Roc74]) permettent en théorie de réduire ce *saut de dualité* et de récupérer une solution primale approchée réalisable. Dans cette approche, un terme proximal est ajouté au Lagrangien, ce qui rompt la séparabilité du problème. Toutefois, il est possible de retrouver cette propriété en linéarisant le terme proximal au voisinage du point courant à chaque itération, mais alors la minimisation est instable et parfois non convergente. Le principe du problème auxiliaire ([Coh84]) consiste à ajouter au terme linéarisé une fonction différentiable fortement convexe dont le rôle est de re-stabiliser le problème ainsi obtenu. Cependant, lorsque le Lagrangien augmenté est appliqué de cette façon, les garanties sur la réduction du saut de dualité ne sont pas maintenues.

Une version linéarisée du Lagrangien augmenté est appliquée pour résoudre le problème de gestion à court terme de EDF ([Pel94]). Ce problème étant non-convexe, cette méthode permet en théorie de réduire le saut de dualité et d'obtenir une solution primale réalisable. Sur le même problème, les auteurs de [DGL05] proposent à partir d'un Lagrangien simple une approche heuristique pour obtenir une solution primale réalisable et de bonne qualité à partir de la solution du problème dual.

La *décomposition par scénarios* est effectuée en relâchant la contrainte de non-anticipativité qui impose des commandes identiques pour deux scénarios en un noeud commun. Cette approche est aussi connue sous le nom de *progressive hedging* [RW91], qui est une application de l'algorithme de décomposition proximale ([LM79, MOT95, Mah95]) pour résoudre un problème d'optimisation stochastique sur arbre décomposé par scénarios. A chaque itération, il s'agit de résoudre pour chaque scénario le problème déterministe auquel est ajouté :

- un terme linéaire à l'aide d'un multiplicateur associé à la contrainte couplante (celle de non-anticipativité),
- un terme proximal qui pénalise la distance à un certain centre.

Une fois que toutes les solutions des sous-problèmes sont obtenues, le vecteur agrégé est projeté sur l'espace des solutions non-anticipatives en moyennant les commandes sur les noeuds de l'arbre. Le résultat constitue le centre de la prochaine itération. Cette approche s'apparente à un Lagrangien augmenté local appliqué sur les sous-problèmes. C'est une méthode simple à mettre en oeuvre mais des travaux restent à accomplir pour améliorer la vitesse de convergence sur de grands problèmes à l'aide de règles efficaces pour la mise à jour des paramètres de pénalisation proximale ([Len08]).

Contrairement aux approches inspirées de la programmation dynamique, les méthodes de décomposition sur un arbre (relaxation Lagrangienne, progressive hedging) ne fournissent pas de stratégies permettant de simuler la coordination du parc face à un échantillon de scénarios. Des heuristiques peuvent être appliquées pour construire de telles stratégies, comme nous le verrons au §5.4.1. Alternativement, ces approches peuvent être reproduites "*en ligne*" : pour un scénario donné, un arbre d'aléas est généré et le problème est résolu afin d'obtenir les commandes pour la première semaine, le processus est ensuite répété pour les semaines suivantes. [BGRS06a] propose une comparaison de la programmation dynamique approximée et de la programmation stochastique sur arbre avec rebouclage pour le problème GMT de EDF. Les résultats obtenus sont à l'avantage de la programmation stochastique sur arbre avec rebouclage.

5.3 Présentation d'un outil de GMT : SOPRANO

SOPRANO-SIMEON est une chaîne logicielle développée par la R&D de EDF pour la gestion moyen terme, afin d'effectuer les étapes 2 (Optimisation) et 3 (Simulation) présentées à la page 89. Un premier logiciel, SOPRANO, est responsable de l'optimisation du parc. La méthodologie mathématique repose sur la décomposition par les prix sur un arbre d'aléas proposée par [BLRS01]. L'une des sorties du logiciel est constituée de valeurs d'usage pour chacune des réserves considérées. Ces valeurs d'usage s'apparentent à des stratégies de gestion et permettent de procéder à l'empilement des actifs de production face à un scénario d'aléas quelconque. C'est le rôle du second logiciel, SIMEON, qui simule les stratégies obtenues sur un grand nombre de scénarios afin de fournir des indicateurs de gestion : coûts marginaux, indicateurs financiers, bilans énergétiques. Dans [AFBM05], cette application est utilisée pour évaluer l'impact des règles de limitation des émissions de CO2 sur la gestion du parc de production d'EDF.

Une partie du travail de thèse a consisté en un audit du bloc de coordination de SOPRANO, basé sur les méthodes de faisceau avec changement de métrique présentées dans [LS97]. La plupart des applications numériques effectuées au cours de la thèse ont été faites avec ce logiciel.

Le problème GMT est représenté sur un arbre de scénarios composé de N noeuds de probabilités $\pi_n, n = 1 \dots N$. Par rapport au problème (5.2), l'indice temporel t est remplacé par les indices n des noeuds de l'arbre. Le portefeuille de production est constitué

de Nu actifs. Le problème traité se réécrit donc :

$$\left\{ \begin{array}{l} \min_{\{p^i\}_{i=1,\dots,Nu}} \quad \sum_{n=1}^N \sum_{i=1}^{Nu} \pi_n C^i(p_n^i, v_n^i) \\ \text{t.q.} \quad \forall n, \sum_{i=1}^{Nu} p_n^i - d_n = 0 \quad (= g(p)) \\ p^i \in \mathcal{P}^i \\ v^i \in \mathcal{X}^i \\ (p^i, v^i) \in \mathcal{D}^i \end{array} \right. \quad (5.4)$$

La complexité de ce problème dépend notamment de la taille du portefeuille (Nu), des caractéristiques des actifs à disposition et du nombre de nœuds N de l'arbre. La contrainte de satisfaction de la demande constitue l'unique contrainte couplant les unités entre elles. Les contraintes dynamiques \mathcal{D}^i ne concernent que les groupes de production faisant l'objet de contraintes de stocks : usines hydrauliques, nucléaires et contrats tarifaires. Il s'agit des contraintes de flots liant l'énergie en stock à l'énergie utilisée et aux éventuels apports. Les variables d'état représentent les niveaux de ces réserves. Pour un horizon de deux ans et à raison d'une discrétisation temporelle avec trois périodes par jour (heures de pointe, de base, creuses), l'arbre de scénarios atteint aisément la taille de 50000 nœuds.

La réalisabilité du problème est garantie par l'ajout d'usines thermiques fictives de coûts proportionnels élevés pouvant être appelées en ultime recours afin de satisfaire la demande.

5.3.1 Décomposition par les prix sur un arbre d'aléas

La décomposition par les prix est le nom couramment donné à la méthode consistant à résoudre le problème dual obtenu par relaxation Lagrangienne de la contrainte de satisfaction de la demande.

Dans le problème (5.4), la contrainte de demande est relâchée et une pénalisation de l'écart de bouclage est introduite dans l'objectif. La fonction ainsi obtenue est le Lagrangien du problème. Comme dans [NR00, BLRS01], on introduit les probabilités des nœuds de l'arbre dans la pénalisation de la contrainte. Bicaud et al. [BLRS01] montrent que cette approche s'apparente à un préconditionnement du problème dual et qu'il est optimal sous certaines hypothèses simples mais réalistes concernant la modélisation des centrales. En omettant les contraintes de fonctionnement, ce Lagrangien "*stochastique*" s'écrit :

$$L(p, x) := \sum_{i=1}^{Nu} \left(\sum_{n=1}^N \pi_n C_n^i(p_n^i, v_n^i) - \langle x, p^i - \frac{d}{Nu} \rangle_{\pi_n} \right) =: \sum_{i=1}^{Nu} L^i(p, x). \quad (5.5)$$

où le produit scalaire $\langle \cdot, \cdot \rangle_{\pi_n}$ est défini par $\langle x, y \rangle_{\pi_n} := \sum_{n=1}^N \pi_n x_n y_n$.

Cette fonction est séparable spatialement et peut donc être évaluée séquentiellement usine par usine. Le vecteur $x \in \mathbb{R}^N$ est le vecteur des multiplicateurs de Lagrange. La théorie de la dualité ([BGLS06, Ch. 8]) nous assure que :

$$(5.4) \Leftrightarrow \min_p \max_x L(p, x) \geq \max_x \min_p L(p, x) \quad (5.6)$$

Le problème de droite dans l'équation (5.6) est le problème dual, concave et éventuellement non-différentiable. Lorsque le problème primal est convexe et bien qualifié (e.g. existence d'un point de Slater), les problèmes primal et dual ont la même valeur optimale. Lorsque ce n'est pas le cas, l'écart entre ces deux valeurs est appelé "*saut de dualité*". On définit

la fonction duale associée au Lagrangien (5.5) par

$$\begin{aligned} \theta(x) &:= \min_p L(p, x) = \sum_{i=1}^{Nu} \min_{p^i} L^i(p^i, x) \\ &=: \sum_{i=1}^{Nu} \theta^i(x), \end{aligned} \quad (5.7)$$

où les fonctions θ^i sont les sous-fonctions duales, également concaves et éventuellement non-différentiables. Afin d'être dans le contexte plus traditionnel d'un problème de minimisation convexe, on considère de façon équivalente la fonction f définie par $f(\cdot) := -\theta(\cdot)$. Le problème dual se réécrit donc

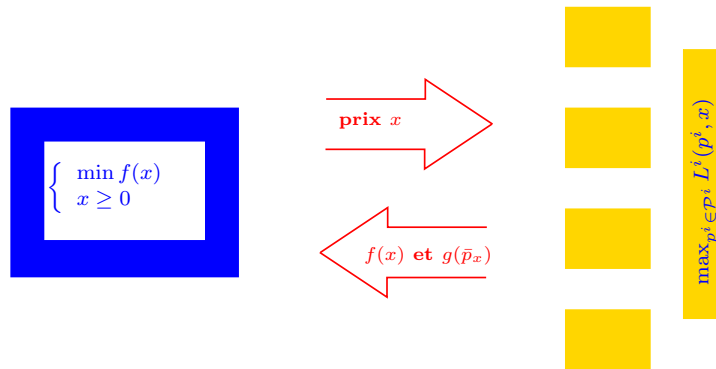
$$\begin{cases} \min & f(x) = \sum_{i=1}^{Nu} f^i(x) \\ \text{t.q.} & x \in \mathbb{R}^N. \end{cases} \quad (5.8)$$

L'évaluation des sous-fonctions duales est effectuée en résolvant des sous-problèmes d'optimisation locaux de la forme :

$$\begin{cases} f^i(x) := \min_{\{p^i\}} & \sum_{n=1}^N \pi_n \left(C_n^i(p_n^i, v_n^i) - x_n \left(p_n^i - \frac{d_n}{Nu} \right) \right) \\ \text{t.q.} & p^i \in \mathcal{P}^i \\ & v^i \in \mathcal{X}^i \\ & (p^i, v^i) \in \mathcal{D}^i. \end{cases} \quad (5.9)$$

Ces sous-problèmes s'interprètent comme l'optimisation du bilan économique de chaque usine, pour un prix de vente de l'énergie défini par le vecteur x . L'évaluation de $f = \sum_i f^i$ donne les solutions locales $\bar{p}_x := (\bar{p}_x^i)_{i=1 \dots Nu}$ qui sont des plannings de production, et elle fournit également un sous-gradient $g(\bar{p}_x) \in \partial f(x)$. Cette information peut être utilisée par un algorithme d'optimisation non-différentiable pour converger vers la solution du problème (5.8), selon le schéma classique de décomposition/coordination de la figure 5.3. Dans le logiciel SOPRANO, le bloc de coordination en bleu est effectué par une méthode de faisceau proximale inspirée de [LS97]. Au paragraphe §5.3.2 nous décrivons brièvement la structure des sous-problèmes locaux représentés en jaune sur la figure 5.3.

FIG. 5.3 – Décomposition/Coordination par les prix



Pour un problème convexe, les multiplicateurs optimaux (prix x) correspondent à la sensibilité de la valeur optimale du problème pour une variation infinitésimale de la demande. Par conséquent, ils correspondent aux coûts marginaux du système à l'équilibre production-consommation. Si le problème n'est pas convexe, ces mêmes multiplicateurs fournissent une bonne approximation des coûts marginaux (cf. [BGLS06, Section 10.2]).

5.3.2 Description des sous-problèmes

Le sous-problème thermique classique

A l'horizon moyen-terme, les groupes thermiques classiques sont modélisés très simplement. Les coûts de production sont linéaires et constants, notés c_{Th}^i , et les contraintes portent uniquement sur les puissances minimales et maximales. Celles-ci varient éventuellement sur l'arbre en fonction de baisses de puissance ou des indisponibilités fortuites. On doit donc résoudre le problème (5.10) :

$$\begin{cases} f^i(x) := \min_{\{p^i\}} & \sum_{n=1}^N \pi_n \left(c_{Th}^i p_n^i - x_n \left(p_n^i - \frac{d_n}{Nu} \right) \right) \\ \text{t.q.} & p_n^i \in [p_n^{i,min}; p_n^{i,max}]. \end{cases} \quad (5.10)$$

La solution \bar{p}^i peut s'exprimer en fonction des paramètres du problème :

$$\begin{aligned} \bar{p}_n^i &= p_n^{i,max}, \text{ si } x_n > c_{Th}^i \\ \bar{p}_n^i &= 0, \text{ si } x_n < c_{Th}^i \\ \bar{p}_n^i &\in [p_n^{i,min}, p_n^{i,max}], \text{ si } x_n = c_{Th}^i. \end{aligned}$$

Le sous-problème hydraulique

Le sous-problème hydraulique est plus complexe en raison des contraintes dynamiques sur le niveau du stock. Les coûts de production sont considérés comme nuls mais le stock final est valorisé selon une courbe de valeur de fin de jeu concave $V(\cdot)$, supposée ici affine. On note \mathcal{O}_T l'ensemble des nœuds finaux de l'arbre. Les apports hydrauliques apparaissant au début d'un nœud n sont notés a_n et les déversements forcés lorsque les apports sont trop importants sont notés s_n^i . Le sous-problème hydraulique peut désormais s'écrire :

$$\begin{cases} f^i(x) := \min_{\{p^i\}} & \sum_{n=1}^N \pi_n \left(-x_n \left(p_n^i - \frac{d_n}{Nu} \right) \right) - \sum_{n \in \mathcal{O}_T} \pi_n V^i(v_n^i) \\ \text{t.q.} & v_{n+1}^i = v_n^i + a_n^i - p_n^i - s_n^i \\ & p_n^i \in [p_n^{i,min}, p_n^{i,max}] \\ & v_n^i \in [v_n^{i,min}, v_n^{i,max}] \\ & s_n^i \geq 0, v_1^i \end{cases} \quad (5.11)$$

Sous cette modélisation, ce problème est linéaire et peut-être résolu par des solveurs comme CPLEX[[Ilo](#)] ou COIN-CLP[[COI](#)]. Pour un arbre de 50.000 nœuds, ce problème comprend environ 70.000 variables et 200.000 contraintes. Entre deux résolutions successives du même sous-problème, seules les variables x_n ont changé. Une résolution par programmation dynamique peut également être envisagée.

Le sous-problème nucléaire

Les réacteurs nucléaires utilisent un combustible faiblement enrichi. Il suffit de 3 ou 4 années pour épuiser complètement le combustible. Il est donc nécessaire de recharger régulièrement le réacteur, tous les 18 mois environ, au cours d'un arrêt d'environ un mois. Lors d'un rechargement, un tiers ou un quart du combustible est renouvelé. En régime de croisière, le réacteur fonctionne donc avec une part de combustible peu irradié, une part intermédiaire et une part de combustible proche de l'épuisement. Les plannings d'arrêts avec ou sans rechargement sont des constantes d'entrée du modèle.

Chaque période de fonctionnement entre deux arrêts est appelée *campagne nucléaire*. Un *cycle nucléaire* désigne la succession d'une campagne et d'un arrêt. La durée d'une campagne dépend de sa *longueur naturelle*, durée pendant laquelle la tranche peut produire à pleine puissance. La fin de la longueur naturelle de la campagne correspond

à une concentration dite de *bore nul*. Cette longueur naturelle dépend également du stock d'énergie disponible en début de campagne. On peut utiliser une quantité variable d'énergie durant une campagne : lorsqu'on n'atteint pas le bore nul lors de l'arrêt, de l'énergie a été gaspillée car du combustible non complètement irradié sera remplacé, tandis qu'un dépassement du bore nul entraînera une longueur naturelle plus courte pour la campagne suivante.

Le *niveau du stock* de combustible est contraint au cours de chaque campagne. En plus de la contrainte sur la dynamique du stock, 3 contraintes limitent le niveau en certaines périodes spécifiques de la campagne nucléaire :

- une contrainte de borne inférieure sur le stock, dite de *prolongation*, qui doit être vérifiée à la date de l'arrêt pour rechargement. Sur le graphique 5.4 par exemple, on peut utiliser jusqu'à 120% de l'énergie correspondant à la longueur naturelle de la campagne ;
- une contrainte dite d'*anticipation* limite le niveau du stock en fin de campagne ;
- enfin, la loi de rechargement lie le niveau de stock à la date d'arrêt et le niveau à la date de reprise.

Les contraintes de fonctionnement limitent également le domaine de *puissance admissible* au cours d'une campagne en fonction du niveau d'usure du combustible. En théorie, le domaine admissible est discontinu et est similaire à celui représenté sur la figure 5.4 pour une tranche de 900 MW. Avant le bore nul, la puissance produite par la tranche est soit comprise entre les valeurs P_{min} et P_{max} , soit elle est nulle, ce qui signifie que la tranche est à l'arrêt. Après 90% d'usure du combustible, la puissance minimale de la tranche est imposée égale à la puissance maximale de la tranche. Après le bore nul (100%), la tranche ne peut plus s'arrêter, sauf cas exceptionnels liés à des raisons de sûreté. Sa puissance est imposée à sa puissance maximale, qui suit alors un *profil de baisse de puissance en prolongation*. Enfin, le nombre des baisses de puissance (*modulation*) au cours d'une campagne est en théorie limité. Ces contraintes sur la puissance admissible rendent le sous-problème nucléaire non convexe : il s'agit d'un problème linéaire en nombres entiers. En pratique, ces contraintes sont simplifiées lors de la phase d'optimisation et le problème est maintenu linéaire et convexe.

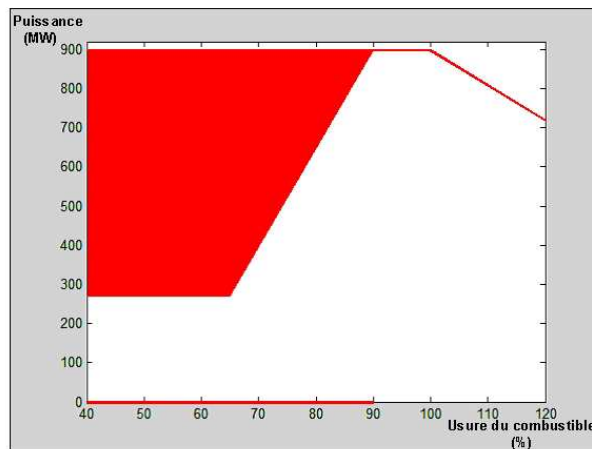


FIG. 5.4 – Puissance admissible au cours d'une campagne (source : [Mal05])

Schématiquement, le sous-problème nucléaire s'écrit :

$$\left\{ \begin{array}{l} f^i(x) := \min_{\{p^i\}} \sum_{n=1}^N \pi_n \left(-x_n(p_n^i - \frac{d_n}{Nu}) \right) - \sum_{n \in \mathcal{O}_T} V(v_n^i) \\ \text{t.q.} \quad v_{n+1}^i = v_n^i - p_n^i \\ p_n^i \in [p_n^{i,min}, p_n^{i,max}] \\ v_n^i \in [v_n^{i,min}, v_n^{i,max}] \quad (\text{Anticipation et Prolongation}) \\ v_n^i \text{ satisfait la loi de rechargement.} \end{array} \right. \quad (5.12)$$

Il s'agit d'un programme linéaire qui peut être résolu par des solveurs standard. Pour un arbre de 50.000 nœuds, il comprend environ 60.000 variables et 360.000 contraintes. Entre deux résolutions successives du même sous-problème, seules les variables x_n ont changé. Une résolution par programmation dynamique peut également être envisagée.

Le sous-problème "Marchés"

Le marché spot est modélisé comme un ensemble de groupes thermiques classiques, symbolisant les possibilités d'achat et de vente d'énergie. La taille d'un acteur comme EDF est telle que les décisions d'achat et de vente sur le marché influencent le prix des contrats. Ce phénomène est pris en compte par l'ajout de plusieurs groupes thermiques de puissance limitée et de coût croissant. Le marché à terme n'est pour le moment pas pris en compte.

Les contrats d'effacement

Dans la version courante de SOPRANO, les contrats d'effacement ne sont pas pris en compte. Toutefois, il pourraient être inclus à l'avenir. Ils s'agit d'un stock de 22 jours d'effacement à placer sur chaque scénario sur une période précise de l'hiver où la demande d'électricité est particulièrement élevée. L'introduction de ces contrats rend le problème non convexe. Actuellement, ils ne sont pas traités par SOPRANO.

5.3.3 Jeux de données utilisés

Jeu de données simplifié : Pour des raisons de confidentialité des données et du fait que la thèse se soit déroulée principalement hors des locaux d'EDF, la plupart de nos travaux ont été réalisés sur un jeu de données fictif simplifié mais calibré de façon à offrir une représentation réaliste d'une vision agrégée du parc. Le portefeuille d'actifs de production comporte :

- 4 tranches nucléaires ;
- 2 centrales hydrauliques dont une usine principale et une usine secondaire réversible (capable d'alterner entre turbinage et pompage pour remplir le réservoir) ;
- 4 centrales thermiques classiques.

L'arbre de scénarios est généré à l'aide d'un échantillon de scénarios représentant les indisponibilités fortuites, les apports hydrauliques pour l'usine principale et la demande.

Jeu de données réel : Lors de séjours au sein d'EDF, nous avons pu tester les évolutions du code sur les données internes. Un jeu de données standard comporte :

- 58 tranches nucléaires ;
- 85 centrales thermiques classiques réelles ou modélisant le marché.

A l'heure actuelle, SOPRANO est utilisé pour la valorisation des seules tranches nucléaires, sans prendre en compte les réserves hydrauliques, dont la production est fixée de façon heuristique sur l'arbre. Cela pourrait être amené à évoluer dans un proche avenir. Les contrats d'effacements ne sont également pas pris en compte. Un tel jeu de données se différencie du précédent principalement par le grand nombre de réserves à optimiser. Dans

une approche par décomposition par les prix sur un arbre de scénarios, une itération est constituée d'une étape duale de coordination et d'une étape primale de résolution successive de sous-problèmes d'optimisation propres à chaque actif. Les sous-problèmes nucléaires étant complexes à résoudre, les temps de calcul de l'étape primale deviennent prépondérants, contrairement au cas du jeu simplifié. Cette constatation nous a amené aux développements présentés au chapitre 4 où une version incrémentale de l'algorithme de faisceau est employée afin de se limiter à la résolution d'un sous-ensemble de sous-problèmes locaux à chaque itération. Les applications numériques de la section 4.4 ont été réalisées avec un jeu de données réel.

5.4 Quelques faits stylisés

En premier lieu, une approche par décomposition se justifie si le problème ne peut être résolu frontalement par les meilleurs solveurs du commerce. Des tests avec CPLEX [FM08] ont permis d'établir qu'effectivement une résolution directe du problème SOPRANO n'est pas possible en environnement stochastique. La limitation provient de la taille du problème : il faudrait environ 20Go de mémoire vive pour stocker le problème posé sur un arbre de 20000 nœuds. Une approche par décomposition est donc justifiée. Dans [FM08], le problème SOPRANO est résolu avec CPLEX en déterministe et les résultats ont permis de valider la qualité des solutions obtenues avec la méthodologie de décomposition par les prix.

Comme tout code opérationnel, SOPRANO doit *in fine* satisfaire des contraintes strictes de temps de calcul. Le code actuel nécessite près d'une nuit de calculs sur des machines performantes pour obtenir une précision satisfaisante. Différentes stratégies doivent donc être envisagées pour accélérer le processus.

Ces temps de calcul se répartissent en deux catégories : la résolution des sous-problèmes locaux (primal) et la résolution du programme quadratique de la méthode de faisceau (dual). Comme nous l'avons dit au §4.3.1 avec l'illustration de la figure 4.2, au delà de la première itération où l'étape duale est particulièrement coûteuse, les temps de calcul primaux sont prépondérants. Nos préoccupations ont donc été principalement de réduire les temps de calcul pour la première itération de coordination, et de limiter les appels aux sous-problèmes locaux. Comme réponse au premier point, nous proposons une approche dynamique de la relaxation de la contrainte couplante, inspirée des travaux présentés au chapitre 3, et nous donnons des résultats numériques au §5.6. Pour le second point, nous avons développé une approche incrémentale de la méthode de faisceau et nous avons présenté les résultats au chapitre 4, à la section 4.4.

Au cours de nos développements sur le code de coordination, nous avons également amélioré certaines fonctionnalités. Il s'agit en particulier de la possibilité de "nettoyer" le faisceau de plans coupants, ceci pouvant avoir un impact sensible sur les temps de calcul duaux et sur la convergence du processus. Les résultats sont présentés au §5.4.2. Nous avons également implémenté la récupération des solutions primales ergodiques $\hat{p}^{i,\ell}$, définies au §2.4.3. Au paragraphe §5.4.1, nous illustrons la convergence de cette suite ergodique vers la solution du problème (5.4). Cette information est ensuite utilisée à la section 5.5 afin d'établir des stratégies de gestion locales pour chaque réserve, utilisables en simulation.

5.4.1 Obtention de solutions primales approchées

Comme nous l'avons rappelé à la section 2.4.3, les solutions locales $p^\ell := (p^{1,\ell}, \dots, p^{Nu,\ell})$ renvoyées lors de l'évaluation de chaque sous-fonction ne convergent pas vers une solution du problème primal initial (5.4), sauf cas particulier de stricte convexité. En pratique, les sous-problèmes ne sont pas strictement convexes et possèdent un continuum de solutions.

Par exemple, comme nous l'avons expliqué au §5.3.2, la solution $\bar{p}^{i,\ell}$ d'un sous-problème thermique classique peut s'exprimer en fonction des paramètres du problème :

$$\begin{aligned}\bar{p}_n^{i,\ell} &= p_n^{i,max}, \text{ si } x_n^\ell > c_{Th}^i \\ \bar{p}_n^{i,\ell} &= 0, \text{ si } x_n^\ell < c_{Th}^i \\ \bar{p}_n^{i,\ell} &\in [p_n^{i,min}, p_n^{i,max}], \text{ si } x_n^\ell = c_{Th}^i.\end{aligned}$$

En pratique, si $x_n^\ell = c_{Th}^i$, on aura $\bar{p}_n^{i,\ell} = p_n^{i,min}$ ou $\bar{p}_n^{i,\ell} = p_n^{i,max}$. Si cette usine est celle réalisant l'équilibre offre demande, la solution du problème sera à priori une valeur particulière de l'intervalle $[p_n^{i,min}, p_n^{i,max}]$. On ne peut donc pas espérer la convergence de ces plannings de production vers la solution du problème.

Toutefois, comme cela est montré par [FK00], les méthodes de faisceau fournissent naturellement une suite d'éléments convergeant vers une solution du problème (5.4) : il s'agit des vecteurs ergodiques $\hat{p}^{i,\ell}$ présentés au §2.4.3. Ces vecteurs sont des combinaisons convexes de solutions locales $\bar{p}^{i,k}$ obtenues lors des itérations précédentes et s'interprètent donc également comme des plannings de production. Dans [DGL05], ils sont dénommés "pseudo-plannings" pour tenir compte du fait que si les contraintes de fonctionnement des usines ne sont pas convexes, ce vecteur ergodique ne sera pas nécessairement réalisable.

Afin de reconstruire ces vecteurs pour chaque usine, il est nécessaire de stocker le faisceau de façon entièrement désagrégée. Les jeux de données réels étant constitués d'une centaine d'usines, chaque nouvel appel aux sous-problèmes locaux fournit autant de vecteurs de taille égale au nombre de nœuds de l'arbre. La taille mémoire nécessaire au stockage de l'information duale est donc considérablement augmentée par rapport à un stockage agrégé.

L'utilisation d'un modèle de plans coupants entièrement désagrégé n'est pas envisageable, le programme quadratique de la coordination devenant trop complexe. Nous avons donc choisit une modélisation partiellement désagrégée pour le sous-problème quadratique, mais le stockage de l'information est effectué de façon entièrement désagrégée.

Sur le graphique 5.5, on trace différents indicateurs relatifs à l'écart de bouclage production/demande sur l'arbre de scénarios :

- en haut à gauche, l'erreur de bouclage relative moyenne, exprimée en pourcentage de la demande ;
- en haut à droite, l'écart type de cette erreur sur les nœuds de l'arbre ;
- en bas à gauche, le quantile à 95% ;
- et enfin en bas à droite le quantile à 99%.

Nous donnons les résultats pour trois configurations de la méthode de faisceau sur 150 itérations de coordination, pour un arbre de scénarios comportant 25000 nœuds, sur le jeu de données *simplifié* présenté au §5.3.3. Les temps de calcul sont représentés en abscisse :

- En rouge, le nombre de coupes dans le faisceau est limité à 30 éléments par *paquet*. Lorsque le faisceau est plein, l'élément agrégé remplace un élément inactif ou actif mais avec une erreur de linéarisation importante.
- En vert, le nombre de coupes est toujours limité à 30 éléments mais les coupes inactives sont régulièrement supprimées, toutes les 20 itérations. Cet exemple sera repris à la section suivante (§5.4.2).
- En bleu, le nombre de coupes est limité à 150, aucune compression d'information n'est donc nécessaire.

Pour les trois exemples, l'écart moyen converge rapidement vers une valeur très faible et il en est de même pour l'écart type. En revanche, il reste des erreurs importantes sur certains nœuds au bout d'un grand nombre d'itérations, comme le montre les courbes des quantiles à 95% et 99%. Nous avons retrouvé ce problème lors de tous nos tests numériques et nous avons constaté que les nœuds concernés ont une durée très courte

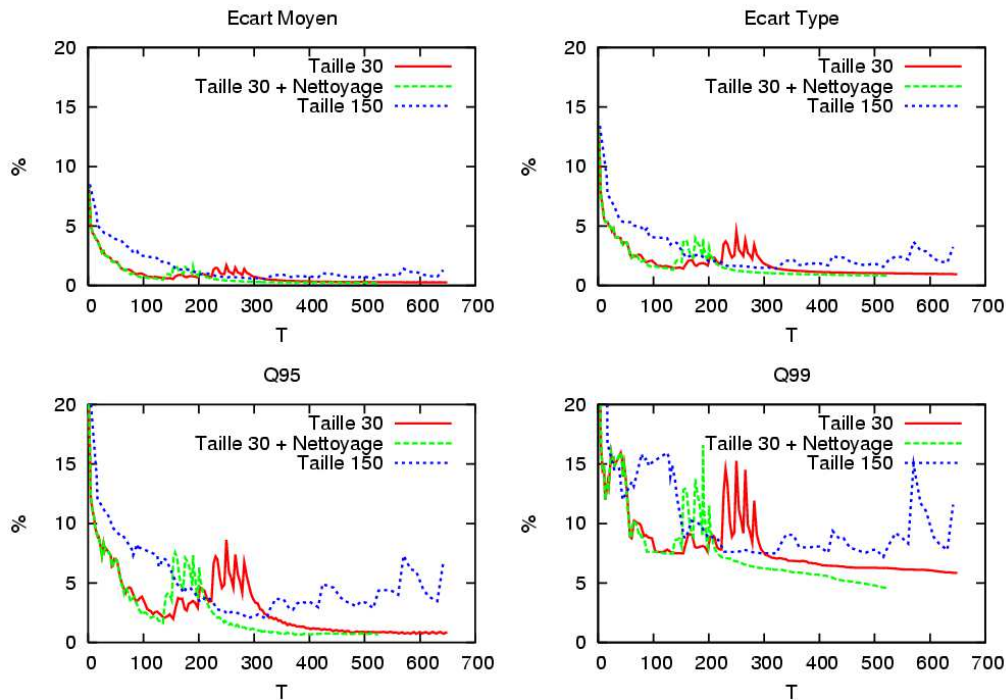


FIG. 5.5 – Recupération primale

et correspondent à des scénarios extrêmes (de probabilité faible). Le poids de ces nœuds dans l'objectif est donc très faible et l'erreur de bouclage y a un impact négligeable sur la valeur de la fonction duale. Une autre explication pour ces écarts réside dans la forme de la courbe d'offre qui peut présenter des sauts importants pour de faibles variations de prix lorsque des groupes de pointe très chers sont appelés. Sur un jeu de données réel, avec un nombre de centrales beaucoup plus grand, ces effets de seuils sont moindres et les écarts de bouclages sont rapidement satisfaisant sur l'ensemble de l'arbre.

On observe un autre phénomène intéressant sur la figure 5.5 : les courbes bleues sont celles qui obtiennent les moins bons résultats alors que l'on s'autorise à stocker un plus grand nombre de coupes, et ce désavantage persiste si l'on ne considère que le nombre d'itérations. Cela tend à montrer qu'il est préférable de limiter la taille du faisceau.

Le problème traité par SOPRANO étant convexe, les solutions $\hat{p}^{i,\ell}$ obtenues satisfont les contraintes de fonctionnement des centrales. Si la modélisation venait à évoluer pour inclure des variables en nombres entiers, il serait intéressant d'envisager une technique de récupération primale comme celle présentée dans [DGL05].

La récupération de ces solutions primales présente plusieurs intérêts. Tout d'abord, cette information se révèle intéressante en soi car elle permet de fournir à l'exploitant du logiciel les commandes optimales sur l'arbre de scénarios, ainsi que les trajectoires de stocks correspondantes. En particulier, l'algorithme de coordination étant stoppé avant la convergence complète, il est rassurant de pouvoir constater la qualité de la solution primale approchée en terme de satisfaction de la demande. De plus, nous avons proposé d'utiliser cette information afin d'effectuer une valorisation locale des réserves d'énergie à l'aide d'une heuristique. Nous reviendrons sur ce point à la section 5.5.

5.4.2 Sélection et compression du faisceau

Comme nous l’avons rappelé au chapitre 2, section 2.4.2, les méthodes de faisceau permettent de borner le nombre de plans coupants utilisés pour approcher la fonction duale.

En réalité, les propriétés de convergence de ces méthodes sont maintenues à partir du moment où l’information contenue dans le sous-gradient agrégé (ou *ergodique*) \hat{s}^ℓ est conservée, en plus de la coupe renvoyée par les sous-problèmes locaux. Cela signifie que les coupes inactives peuvent être supprimées, ou que le faisceau peut être entièrement compressé à la fin d’une itération ([BGLS06, Lemme 9.10]). Toutefois, lorsqu’une coupe est supprimée, le modèle n’est plus exact au point correspondant.

Actuellement, ces propriétés ne sont pas réellement exploitées dans le code de coordination de SOPRANO. Lorsque le faisceau est plein (pour une limite sur le nombre de coupes fixée par l’utilisateur), à chaque nouvelle itération, deux coupes anciennes sont remplacées par le sous-gradient agrégé et la nouvelle coupe. En pratique, on constate que les itérations duales sont plus lentes une fois cette limite atteinte. Cela s’explique par la procédure de démarrage à chaud du programme quadratique qui devient légèrement moins efficace lorsque des coupes sont remplacées (démarrage “*tiède*”).

Nous avons donc envisagé la possibilité de supprimer périodiquement les coupes non actives du faisceau. Cela a pour effet de repousser le remplissage du faisceau et donc le ralentissement des itérations duales. De plus, nous avons constaté que la convergence du processus était ainsi améliorée, en particulier pour les postes avec un mauvais bouclage. Sur le graphique 5.5, nous avons représenté en vert les résultats obtenus avec un faisceau limité à 30 coupes où l’on supprime les coupes inactives toutes les 20 itérations. La solution primale obtenue est plus rapidement satisfaisante. En particulier, les écarts de bouclages extrêmes sont sensiblement diminués. Sur la figure 5.6, nous nous intéressons aux valeurs de la fonction duale au fil des itérations, pour les mêmes instances de la méthode de faisceau. A nouveau, la méthode avec nettoyage périodique du faisceau converge plus rapidement. Les résultats sont donnés relativement au nombre d’itérations de coordination et aux temps de calcul, l’avantage de la méthode avec nettoyage étant plus net dans ce second cas (à gauche). Rappelons que ces résultats sont obtenus sur le jeu de données simplifié avec lequel les temps de calculs de la phase dual restent significatif sur l’ensemble du processus. Sur un jeu de données réel, le temps pris par la résolution des sous-problèmes est tel que les gains de temps sur le dual ne sont plus visibles (cf. figure 4.2).

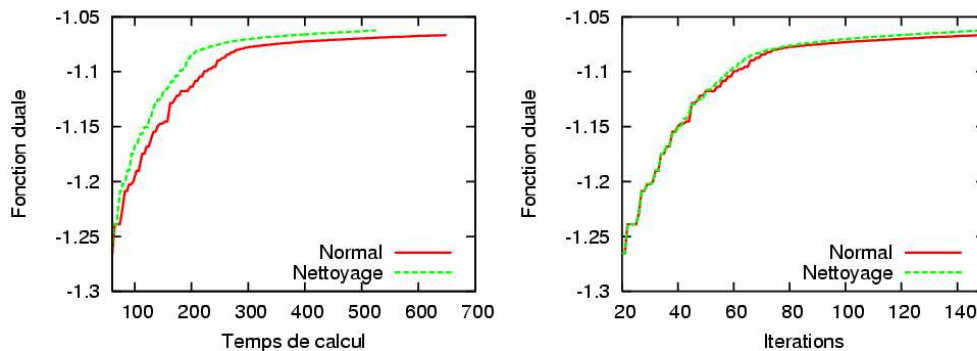


FIG. 5.6 – Influence de la procédure de nettoyage du faisceau

Les applications numériques ont donc permis de montrer qu’une procédure de nettoyage des coupes inactives appliquée régulièrement sur le faisceaux de coupes supports permettait de stabiliser la convergence du processus tout en accélérant les itérations duales.

5.5 Calcul de stratégies pour la simulation

Comme nous l'avons expliqué au §5.2.3, l'un des objectifs de la gestion de production moyen terme est de fournir une valorisation des énergies de réserve afin de servir d'indicateur pour les prises de décision sur l'horizon court terme. Traditionnellement, cette valorisation se donne sous la forme de la valeur d'usage d'un stock. Ces valeurs d'usage doivent également permettre de simuler la gestion coordonnée du parc de production face à un échantillon d'aléas quelconque afin de fournir un ensemble d'indicateurs de gestion à moyen terme comme des chroniques de coûts marginaux, des indicateurs financiers, de risque, ou de bilans énergétiques.

5.5.1 Valeur d'usage d'un stock d'énergie

Idéalement, en résolvant le problème global par programmation dynamique, on aurait à disposition les fonctions de Bellman pour chaque noeud de l'arbre d'aléa. La valeur d'usage d'un stock en un noeud de l'arbre et pour un vecteur d'état donné est alors obtenue en calculant la sensibilité de la valeur de Bellman pour une utilisation d'un kwh du stock. Toutefois, la dimension du vecteur d'état du problème rend inenvisageable une telle approche. Des formulations approchées de la programmation dynamique, comme la programmation dynamique duale ou la programmation dynamique approximée présentées à la section 5.2.4, fournissent également une information de ce type.

Au contraire, la méthodologie retenue par SOPRANO ne donne pas accès à cette information. A la convergence de la méthode, on ne dispose que des commandes optimales sur l'arbre et des coût marginaux du système à l'équilibre production/consommation. Ces données ne permettent pas à priori de simuler la gestion du parc sur un échantillon de scénarios quelconque.

Afin de produire des stratégies utilisables en simulation, une heuristique a été mise en place. La valorisation d'un stock par SOPRANO est effectuée via le calcul d'une "valeur de Bellman locale". Les dérivées de ces valeurs en fonction du stock forment les valeurs d'usage, assimilables à des coûts d'utilisation d'une unité de stock au noeud considéré. Ces valeurs de Bellman locales peuvent donc être assimilées à des stratégies de gestion (ou *feedbacks*) face aux aléas.

L'heuristique actuelle utilise les solutions duales optimales sur l'arbre de scénarios pour obtenir des valorisations/fonctions de Bellman locales pour chaque réserve prise séparément. Il s'agit d'une valorisation locale dite *sur signal de prix*. Nous proposons une approche différente utilisant les commandes optimales en lieu et place de ce signal de prix.

Dans les deux cas, une étape supplémentaire consiste à moyenner sur chaque pas de temps les fonctions de Bellman des noeuds correspondants afin d'obtenir une unique fonction par pas de temps.

5.5.2 Valorisation face à un signal de prix

Le calcul actuel des stratégies locales se fait à l'aide d'une dernière étape de décomposition par les prix où les sous-problèmes locaux sont résolus par programmation dynamique.

Comme à chaque itération de la phase de décomposition/coordination, les sous-problèmes locaux "placent" leur production en chaque noeud de l'arbre d'aléas étant donnée la rémunération offerte par les variables duales afin d'optimiser leur bilan économique. Or, à la convergence, les variables duales optimales correspondent aux coûts marginaux du système réalisant l'équilibre offre-demande.

On note λ le vecteur des variables duales optimales obtenues à la convergence. Pour une réserve i , le sous-problème à résoudre s'écrit donc :

$$\left\{ \begin{array}{l} \min_{\{p^i\}} \sum_{n=1}^N \pi_n \left(-\lambda_n(p_n^i - \frac{d_n}{Nu}) \right) - \sum_{n \in \mathcal{O}_T} \pi_n V^i(v_n^i) \\ \text{t.q. } p^i \in \mathcal{P}^i \\ v^i \in \mathcal{X}^i \\ (p^i, v^i) \in \mathcal{D}^i \end{array} \right. .$$

Chaque sous-problème représentant un stock d'énergie calcule ainsi sa gestion optimale pour une rémunération de l'énergie donnée par les coûts marginaux sur l'arbre. Une résolution par programmation linéaire ne fournirait que les commandes optimales pour ce stock. En employant la programmation dynamique stochastique présentée au §5.2.4, on peut récupérer les stratégies locales de gestion optimale face aux coûts marginaux du système sous la forme de fonctions de Bellman en chaque noeud de l'arbre. Ces stratégies correspondent aux plannings de production renvoyés lors de la dernière itération de décomposition par les prix. Le problème n'étant pas strictement convexe, la somme de ces plannings ne satisfait à priori pas la contrainte de demande.

Cette heuristique permet donc de générer des stratégies de gestion locales pour chaque réserve. Celles-ci seront ensuite fournies à un simulateur moyen terme ou aux gestionnaires du court terme.

Dans cette approche, on n'utilise qu'une seule information : les variables duales obtenues à la convergence de la décomposition. On suppose donc que quelle que soit la stratégie choisie par la réserve considérée, le coût marginal du système ne change pas. Cela a pour effet d'entraîner une variation faible de la valeur d'usage de la réserve en fonction du niveau de stock, autrement dit une faible courbure des fonctions de Bellman. Lorsque l'on simule la coordination des différents actifs, ce manque de courbure et le fait d'utiliser une seule et même information pour construire toutes les stratégies entraîne un couplage insuffisant des différents moyens de production.

5.5.3 Valorisation à production complémentaire fixée

Lors de l'audit que nous avons mené sur le code de faisceau responsable de la coordination dans l'outil SOPRANO, nous avons constaté que la possibilité d'extraire une solution primale approchée n'était pas exploitée. Nous l'avons donc implémentée (cf. §5.4.1) et nous avons envisagé la possibilité d'utiliser cette information lors de la construction de stratégies locales.

Étant données les commandes optimales pour chaque réserve sur l'arbre de scénarios, on peut choisir de résoudre le problème de gestion de production global en ne considérant qu'une réserve, les autres ayant une production fixée. Cela revient à résoudre le problème de gestion de production incluant une unique réserve pour une demande défalquée des productions des réserves complémentaires. Le parc de production est donc constitué des usines thermiques et de la réserve considérée.

On divise le portefeuille d'actifs de production en deux catégories :

- les indices $i \in \mathcal{N}^{Th}$ représentent les centrales thermiques incluant éventuellement la modélisation du marché spot ;
- les indices $j \in \mathcal{N}^R$ représentent les réserves et les commandes optimales associées $\{\hat{p}^j\}_{j \in \mathcal{N}^R}$.

En considérant une réserve particulière $k \in \mathcal{N}^R$, le problème à résoudre pour la réserve k s'écrit :

$$\left\{ \begin{array}{l} \min_{\{p^i\}_{i \in \mathcal{N}^{Th} \cup \{k\}}} \sum_{n=1}^N \sum_{i \in \mathcal{N}^{Th}} \pi_n C^i(p_n^i, v_n^i) - \sum_{n \in \mathcal{O}_T} \pi_n V^k(v_n^k) \\ \text{t.q. } \forall n, \sum_{i \in \mathcal{N}^{Th}} p_n^i + p_n^k = d_n - \sum_{j \in \mathcal{N}^R \setminus \{k\}} \hat{p}_n^j \\ p^i \in \mathcal{P}^i, p^k \in \mathcal{P}^k \\ v^i \in \mathcal{X}^i, v^k \in \mathcal{X}^k \\ (p^k, v^k) \in \mathcal{D}^k \end{array} \right.$$

A nouveau, une résolution par programmation dynamique permet d'extraire des plannings de production et une stratégie de gestion correspondante sur l'arbre (fonctions de Bellman). Les plannings ainsi obtenus correspondent aux commandes optimales et satisfont la contrainte de demande.

5.5.4 Résultats numériques

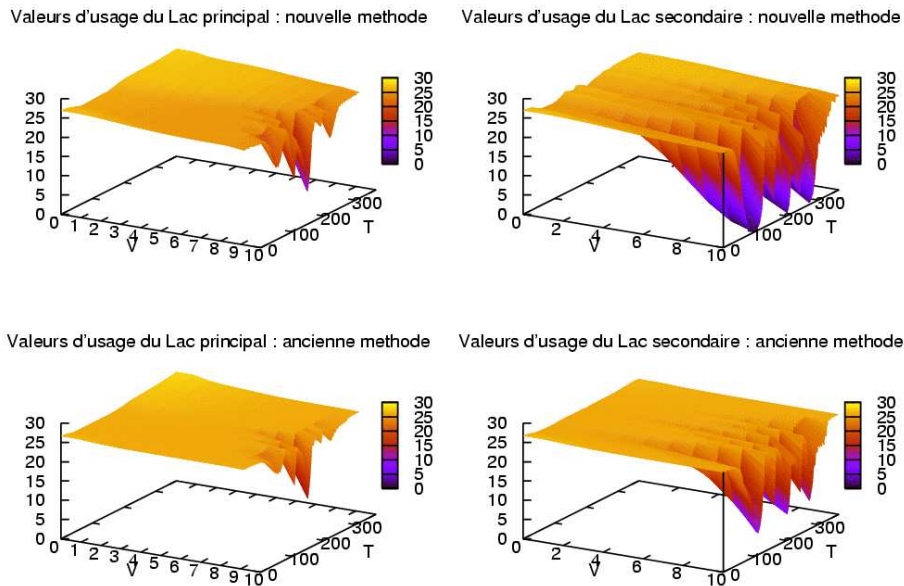


FIG. 5.7 – Valeurs d'usage des réserves hydrauliques

Les illustrations proposées sur les figures 5.7 et 5.8 ont été obtenues avec le jeu de données simplifié présenté à la section 5.3.3. Il n'inclut que deux réserves hydrauliques (un lac principal avec apports, une usine réversible pouvant turbiner ou pomper de l'énergie), quatre réserves nucléaires et quatre usines thermiques. L'horizon de temps est de deux ans et comporte 3 pas de temps par jour. Pour représenter les valorisations obtenues à l'aide des deux méthodes, nous avons choisi un problème déterministe sur 2 ans.

Sur les graphiques de la figure 5.7, nous représentons les valeurs d'usage obtenues en fonction du volume et du pas de temps sur la première année pour les réserves hydrauliques. A droite sont représentées les valeurs d'usage du lac principal, à gauche celles de l'usine réversible. On remarque que les valeurs d'usage calées sur les plannings optimaux des réserves sont effectivement plus courbées que celles calées sur signal de prix.

La figure 5.8 s'intéresse aux deux premières tranches nucléaires présentes dans le portefeuille. L'approche proposée introduit également une courbure plus importante. Intuitivement, cela tend à maintenir le stock éloigné de la zone où apparaît la contrainte de baisse de puissance en prolongation, avant l'arrêt pour rechargement.

5.5.5 Test des stratégies en simulation

L'un des objectifs de la valorisation des réserves est de permettre de simuler la gestion coordonnée du parc de production face à des scénarios d'aléas quelconques. Cette étape est réalisée par un autre logiciel, SIMEON. Lors de cette étape, la modélisation retenue pour

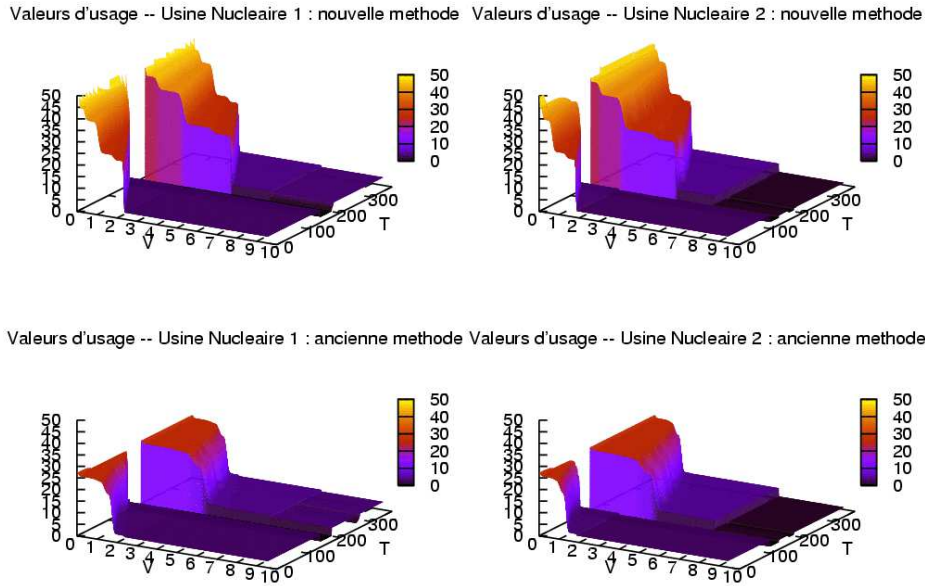


FIG. 5.8 – Valeurs d’usage pour deux tranches nucléaires

les différents éléments du parc est évolutive et peut être beaucoup plus fine que celle de SOPRANO. En particulier, les contraintes de fonctionnement du nucléaire peuvent faire apparaître un nombre maximal de modulations. Des contraintes sur des réserves d’énergie disponibles à la hausse à tout instant pour faire face à des événements fortuits peuvent également être prises en compte. Ce simulateur n’est pas propre à l’outil SOPRANO et ne nécessite en entrée que la donnée de fonctions de Bellman permettant un arbitrage pour l’utilisation des différentes énergies avec contraintes de réserves. Il peut donc être vu comme un outil de comparaison entre différentes méthodologies pour le calcul de stratégies de gestion.

Afin de procéder à une première validation de l’approche proposée au §5.5.3 pour valoriser les réserves, nous l’avons comparé à la valorisation sur signal de prix sur un jeu de données réel. Les réserves dites *de pointe* (hydraulique et contrats d’effacement) ne sont pas incluses au modèle car c’est le cadre actuellement retenu pour le logiciel SOPRANO. La simulation est effectuée sur un ensemble de 484 scénarios d’aléas (demande, indisponibilités fortuites, prix et profondeur du marché spot).

Le tableau 5.5.5 renseigne les coûts moyens de gestion obtenus sur l’ensemble des scénarios avec la valorisation des réserves face au signal de coûts marginaux (M1.) et la valorisation à production complémentaire fixée (M2.).

	Total(M€)	Nucl.	Therm.	Achat	Vente	Def.
M1.	1596.07	-2176.26	1305.05	382.93	2062.30	22.04
M2.	1594.07	-2176.56	1303.80	383.51	2061.57	21.75

TAB. 5.2 – Coût moyen de gestion en simulation

Le coût du nucléaire correspond à la variation de valeur du stock de combustible entre le début et la fin de la période d’étude compte tenu de la valorisation donnée par les valeurs de fin de jeu (utilisée en optimisation également).

La stratégie M2 permet donc un gain théorique moyen de 2M€par an. Pour donner un

ordre de comparaison, la gestion du nucléaire sur coût proportionnel, donc sans gestion des stocks, entraîne un surcoût annuel d'environ 10M€. La méthode de valorisation à production complémentaire fixée entraîne donc potentiellement une amélioration de 20% des gains espérés sur un horizon annuel.

Une vision agrégée des production des différentes usines permet de comprendre l'origine de ces différences de coûts de gestion. En effet, la modulation nucléaire (production à puissance réduite) moyenne dans le cas de la stratégie M2 est légèrement plus importante que pour M1 (+10%). Cette modulation permet de repousser l'arrivée au niveau de stock de bore nul qui entraîne une baisse de puissance imposée. Le tableau 5.3 représente la participation de chaque type d'actif sur la production la production de l'ensemble du parc, en moyenne sur l'ensemble des scénarios. Les différences sont très faibles.

	Total(TWh)	Nucl.	Therm.	Achat	Vente	Def.
M1.	550.59	80.01(%)	12.64	1.22	6.12	0.00
M2.	550.59	79.98(%)	12.64	1.24	6.14	0.00

TAB. 5.3 – Participation de chaque type d'actif

Les résultats obtenus avec les deux approches diffèrent très peu mais sont plutôt en faveur de la seconde méthode. Il convient toutefois les relativiser puisqu'ils ont été obtenus pour un jeu de donnée particulier : la date de début d'horizon est un élément déterminant du coût global de gestion. Une analyse plus fine reste donc nécessaire pour valider cette approche.

Enfin, nous n'avons pas abordé la question de la synthèse des stratégies issues de l'arbre de scénarios. En effet, à la fin de la phase de valorisation, nous disposons de fonctions de Bellman en chaque noeud de l'arbre. Actuellement, une simple moyenne sur l'ensemble des noeuds d'un même pas de temps permet d'obtenir des fonctions par pas de temps. Toutefois, une grande quantité d'information est perdue lors de cette étape. Il serait donc certainement judicieux d'envisager de conserver une plus grande quantité d'information en indexant par exemple les stratégies en fonction du niveau de la demande.

5.6 Apports d'une méthode dynamique de faisceaux

Nous avons présenté au chapitre 3 la Relaxation Lagrangienne Dynamique comme une approche possible pour la résolution de problèmes de grande taille où le nombre de contraintes à dualiser est très important. L'idée est la suivante : afin de réduire la dimension de l'espace dual, seul un sous-ensemble de contraintes est dualisé, tandis que les autres contraintes sont ignorées. Une procédure de séparation doit permettre de mettre à jour dynamiquement ce sous-ensemble, en sélectionnant quelques contraintes violées par l'itéré primal courant.

Cette approche est particulièrement pertinente lorsque les contraintes à dualiser sont des contraintes d'inégalité dont seul un sous-ensemble est actif à l'optimum. Ici, le problème résolu par SOPRANO est effectivement de grande taille, mais les contraintes à dualiser sont des contraintes d'égalité, donc toutes actives à l'optimum. Par conséquent, une approche dynamique telle que celle présentée au chapitre 3 ne semble pas être d'une grande utilité. De plus, les temps de calcul de la phase primale sont largement prépondérants sur l'ensemble du processus, ce qui rend moins cruciaux les gains de temps sur les itérations duales. Toutefois, les nombreux tests menés sur le code SOPRANO pour différents jeux de données nous ont amené à constater deux difficultés qu'une approche par relaxation dynamique pourrait aider à améliorer.

En premier lieu, on constate que la première itération de l'étape duale peut s'avérer très coûteuse lorsque la dimension de l'arbre grandit. Considérons par exemple un problème

de gestion de production *simplifié* où les centrales nucléaires sont modélisées comme des centrales thermiques classiques avec un coût proportionnel fixé. Dans ce cas de figure, les temps de calcul des phases primales et duales sont comparables sur l'ensemble du processus, mais la première itération duale est de loin la plus coûteuse.

Cela s'explique du fait que le sous-problème quadratique résolu lors d'une itération de l'algorithme de faisceau est en réalité plus complexe que le sous-problème générique (2.12) présenté au chapitre 2. Une contrainte de positivité est introduite afin de ne pas autoriser de prix marginaux négatifs : la production doit être au moins égale à la demande. Une borne supérieure est également ajoutée, égale au coût fictif associé à la défaillance du système (impossibilité de satisfaire la demande).

Comme le montre la figure 5.9, les temps de calcul de la première itération du programme quadratique (en rouge) augmentent très rapidement avec la taille de l'arbre.

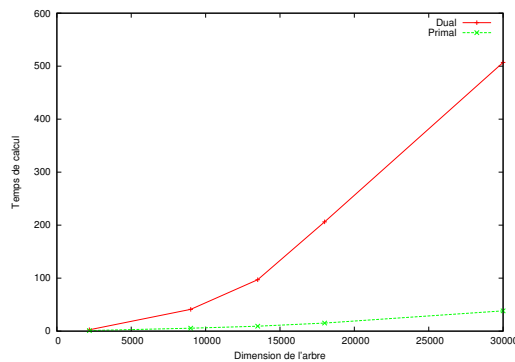


FIG. 5.9 – Première itération de coordination

On voit qu'il faut 500 fois plus de temps pour effectuer la première itération duale sur un arbre de 30.000 nœuds que sur un arbre de 2500 nœuds. Partant de ce constat, nous avons testé une relaxation Lagrangienne dynamique où l'on résout un sous-problème quadratique de dimension réduite lors des premières itérations en ne visitant l'espace dual que selon certaines directions. Lorsque de nouvelles contraintes sont ensuite "libérées", le programme quadratique peut utiliser l'information de l'itération précédente pour résoudre plus rapidement le problème.

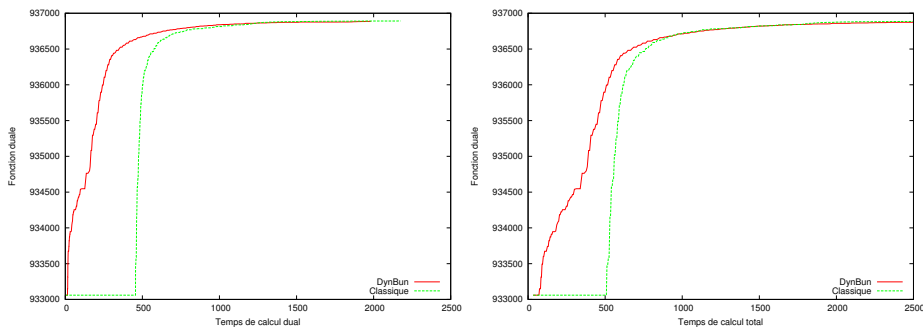


FIG. 5.10 – Relaxation dynamique Vs classique

Sur la figure 5.10, un arbre de 30.000 nœuds est considéré. Nous illustrons le comportement de l'algorithme classique (en vert) et de la méthode dynamique où à chaque pas de descente 5.000 nouvelles coordonnées duales sont libérées. Nous n'avons pas retenu de stratégie particulière pour sélectionner les variables libérées : ceci est effectué progres-

sivement en partant de la fin de l'arbre de scénarios. À gauche sur la figure 5.10, seuls les temps de calcul de la phase duale sont pris en compte, tandis que sur la figure de droite la phase primale est également incluse. Comme cela était espéré, la méthode dynamique est bien plus rapide que la méthode statique. Dans une optique où les temps de calculs de la phase primale seraient considérablement réduits, par exemple via une parallélisation de la résolution des sous-problèmes locaux, une approche dynamique de la relaxation Lagrangienne au-cours des premières itérations de coordination pourrait donc s'avérer très utile.

Nous avons également considéré une autre piste d'application pour une approche dynamique de la relaxation Lagrangienne sur le problème de gestion de production. Sur des problèmes de grande taille ($N > 20.000$), certains postes, très minoritaires, conservent un mauvais bouclage même après un très grand nombre d'itérations. En bloquant les prix des postes ayant un très bon bouclage (très majoritaires), on peut espérer améliorer la solution approchée. Malheureusement, les résultats n'ont pas été concluants. Le problème étant fortement couplé dans le temps, le fait de ne bouger les prix qu'en certains nœuds a un effet sur l'ensemble de l'arbre et a tendance à dégrader rapidement la solution sur les nœuds non considérés. Pour traiter ce problème, il pourrait s'avérer intéressant d'envisager l'utilisation d'un terme proximal désagrégé. Actuellement, le terme proximal μ_ℓ est un scalaire positif et introduit par conséquent une pénalisation uniforme du déplacement selon toutes les directions de l'espace dual. Il est théoriquement possible de définir une norme plus générale en introduisant pour l'itéré courant ℓ la norme associée à une matrice définie positive d'ordre N , M_ℓ :

$$|\cdot|_\ell^2 := \langle M_\ell \cdot, \cdot \rangle.$$

[LS97] propose des règles de mises à jour inspirées de la théorie de quasi-Newton pour la métrique associée à M_ℓ . Toutefois, la résolution du programme quadratique au centre de la méthode de faisceau devient alors beaucoup plus complexe et le solveur dont nous disposons, basé sur [Kiw86], n'est pas compatible avec ce cadre plus général.

Pour finir, le code de faisceau de SOPRANO ayant montré de très bonnes performances globales, il serait intéressant de l'utiliser pour résoudre certains problèmes en nombres entiers ayant fait l'objet d'une application de la relaxation Lagrangienne dynamique avec méthode de sous-gradient dans la littérature ([Luc92, BS08]). Dans ce but, des contacts ont été pris avec Alexandre Belloni, qui a réalisé de nombreuses expérimentations numériques dans le cadre de [BS08], mais nous n'avons pas eu suffisamment de temps pour mener à bien ces applications.

5.7 Conclusion

Au cours de ce chapitre, nous avons détaillé la problématique de la gestion de production moyen terme. Nous avons présenté un logiciel de gestion moyen terme utilisé pour permettre une coordination efficace des différentes tranches nucléaires : SOPRANO.

Ce logiciel repose sur une méthodologie de décomposition par les prix sur un arbre de scénarios, en appliquant un algorithme de faisceau pour l'étape de coordination. Durant nos travaux, nous avons audité, corrigé et enrichi ce code de faisceau. En particulier, il est désormais possible d'extraire simplement les commandes optimales approchées sur l'arbre et une procédure de nettoyage du faisceau permet d'améliorer légèrement la stabilité de l'algorithme et les temps de calcul de la phase duale.

Nous avons également proposé une heuristique nouvelle afin de calculer des stratégies de gestion locales pour chaque réserve d'énergie. Elle consiste à calculer par programmation dynamique la stratégie optimale sur l'arbre pour une réserve en fixant les commandes des réserves complémentaires à leur productions optimales. SOPRANO ayant actuellement pour objectif principal la coordination du seul parc nucléaire, il serait peut-

être pertinent d'enrichir cette heuristique en incluant le niveau des réserves hydrauliques dans la description de l'état pour cette programmation dynamique finale. Il s'agirait donc pour chaque tranche nucléaire de calculer la stratégie de gestion coordonnée avec les deux réserves hydrauliques principales, en fixant les productions des tranches nucléaires complémentaires. La dimension du vecteur d'état resterait raisonnable. On obtiendrait ainsi des valeurs d'usage fonctions du pas de temps, du niveau de stock de la réserve et du niveau des deux réserves hydrauliques principales, ce qui devrait améliorer la qualité de la coordination des différents éléments du parc.

Bibliographie

- [AFBM05] M. Avellà-Fluvià, K. Boukir, and P. Martinetto, *Handling a CO2 reservoir in mid term generation scheduling*, Proceedings 15th Power System Computation Conference (2005).
- [App08] R. Apparigliato, *Règles de décision pour la gestion du risque : Application à la gestion hebdomadaire de la production électrique.*, Ph.D. thesis, Ecole Polytechnique – EDF R&D, 2008.
- [Bar04] K. Barty, *Contributions à la discrétisation des contraintes de mesurabilité pour les problèmes d’optimisation stochastique*, Ph.D. thesis, ENPC, 6 2004.
- [BD59] R. Bellman and S.E. Dreyfus, *Functional approximations and dynamic programming*, Math. tables and other aides to computation **13** (1959), 247–251.
- [Ben62] J.F. Benders, *Partitioning procedures for solving mixed variable programming problems*, Numerische Mathematik **4** (1962), 173–184.
- [BGLS06] J.F. Bonnans, J.C. Gilbert, C. Lemaréchal, and C. Sagastizábal, *Numerical optimization : Theoretical and practical aspects*, Springer, 2006.
- [BGRS06a] K. Barty, P. Girardeau, J-S. Roy, and C. Strugarek, *Comparaison de méthodes pour la gestion de production nucléaire à moyen terme*, Tech. report, EDF R&D, 2006.
- [BGRS06b] ———, *A q-learning algorithm with continuous state space*, Optimization Online (2006).
- [BGRS08] ———, *Application of kernel-based stochastic gradient algorithms to option pricing*, Monte Carlo Methods Appl. **14** (2008), no. 2, 99–127.
- [BL97] J. Birge and F. Louveaux, *Introduction to stochastic programming*, Springer, 1997.
- [BLRS01] L. Bacaud, C. Lemaréchal, A. Renaud, and C. Sagastizábal, *Bundle methods in stochastic optimal power management : a disaggregate approach using preconditioners*, Comp. Opt. and App. **20** (2001), no. 3, 227 – 244.
- [BS08] A. Belloni and C. Sagastizábal, *Dynamic bundle methods*, Mathematical Programming (2008).
- [Coh84] G. Cohen, *Décomposition et coordination en optimisation déterministe différentiable et non différentiable*, Ph.D. thesis, Université de Paris Dauphine, 1984.
- [COI] COIN-OR, <https://projects.coin-or.org/Clp/>, *Coin clp solver : User guide*.
- [CP99] Z.L. Chen and W.B. Powell, *Convergent cutting plane and partial sampling algorithm for multistage stochastic linear programs with recourse*, Journal of Optimization Theory and Applications (1999).
- [Dal06] A. Dallagi, *Méthodes particulières en commande optimale stochastique*, Ph.D. thesis, Université Paris I – ENPC – EDF R&D, 2006.

- [DB06] C.J. Donohue and J.R. Birge, *The abridged nested decomposition method*, Algorithmic Operations Research (2006).
- [DGL05] L. Dubost, R. Gonzalez, and C. Lemaréchal, *A primal-proximal heuristic applied to the french unit-commitment problem*, Math. Program. **104** (2005), no. 1, 129–151.
- [Din07] A. Diniz, *A Lagrangian relaxation decomposition approach to the short term security constrained optimal hydrothermal scheduling – Application to the Brazilian system*, Ph.D. thesis, COPPE/UFRJ Rio de Janeiro, 2007.
- [DS08] A.L. Diniz and T.N. Santos, *Multi-period stage definition for the multi stage benders decomposition approach applied to hydrothermal scheduling*, Proceedings – Engopt – http://www.engopt.org/en/approved_papers.html, 2008.
- [DZ98] A. Dembo and O. Zeitouni, *Large deviations techniques and applications*, Springer, 1998.
- [Emi05] G. Emiel, *Programmation dynamique duale pour la gestion de production à moyen-terme.*, Master thesis, Université Paris 6 – EDF R&D, 2005.
- [FK00] S. Feltenmark and K.C. Kiwiel, *Dual applications of proximal bundle methods, including Lagrangian relaxation of nonconvex problems*, SIAM J. on Optimization **10** (2000), no. 3, 697–721.
- [FM08] X. Faure-Miller, *Résolution frontale du problème SOPRANO*, Rapport de stage, Ecole Polytechnique – EDF R&D, 2008.
- [HR03] H. Heitsch and W. Römisich, *Scenario reduction algorithms in stochastic programming*, Computational Optimization and Applications **24** (2003), 187–206.
- [HRS06] H. Heitsch, W. Römisich, and C. Strugarek, *Stability of multistage stochastic programs*, SIAM Journal on Optimization **17** (2006), 511–525.
- [Ilo] Ilog, Inc., <http://www.ilog.com/products/cplex/>, *Cplex solver : User’s manual*.
- [JP04] J. Jacod and P. Protter, *Probability essentials*, Springer, 2004.
- [Kiw86] K.C. Kiwiel, *A method for solving certain quadratic programming problems arising in nonsmooth optimization*, IMA J. Numer. Anal. **6** (1986), 137–152.
- [KPB06] M.O.I. Khemmoudj, M. Porcheron, and H. Bennaceur, *When constraint programming and local search solve the scheduling problem of electricité de france nuclear power plant outages*, Proceeding to the 12th International Conference on Principles and Practice of Constraint Programming (2006), 271–283.
- [Len08] A. Lenoir, *Modèles et algorithmes pour la planification de production à moyen terme avec demande incertaine : Application de techniques de décomposition décentralisée*, Ph.D. thesis, ISIMA Clermont Ferrand – EDF R&D, 2008.
- [LM79] P.L. Lions and B. Mercier, *Splitting algorithms for the sum of two nonlinear operators*, SIAM J. Numer. Anal. **16** (1979), 964–979.
- [LS97] C. Lemaréchal and C. Sagastizábal, *Variable metric bundle methods : from conceptual to implementable forms*, Math. Program. **76** (1997), no. 3, 393–410.
- [LT85] J. Leveugle and P. Torrion, *Comparaison de différentes méthodes d’optimisation, appliquées à la gestion annuelle du système offre-demande d’électricité français.*, Tech. Report M85-108, EDF R&D – Etudes Economiques Générales, 1985.
- [Luc92] A. Lucena, *Steiner problem in graphs : Lagrangean relaxation and cutting-planes*, COAL Bulletin **21** (1992).

- [Mah95] P. Mahey, *Separable Augmented Lagrangians for the decomposition of large convex programs*, Investigacion Operativa **5** (1995), 1–26.
- [Mal05] B. Malgat, *Modélisation de contraintes de fonctionnement des centrales nucléaires*, Mémoire de fin d'études, ISIMA – EDF R&D, 2005.
- [MOT95] P. Mahey, S. Oualibouch, and Pham Dinh Tao., *Proximal decomposition on the graph of a maximal monotone operator*, SIAM J. on Optimization **5** (1995), no. 2, 454–466.
- [NR00] M. P. Nowak and W. Romisch, *Stochastic Lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty*, Annals of Operations Research **100** (2000), 251–272.
- [PCK99] M.V.F. Pereira, N. Campodónico, and R. Kelman, *Application of stochastic dual dynamic programming and extensions to hydrothermal scheduling*, Tech. report, PSRI, 1999.
- [Pel94] F. Pellegrino, *Optimiser les plannings de production sur la journée : le modèle apogee*, Tech. Report HR-35/94/019, EDF R&D, 1994.
- [PG08] A.B. Philpott and Z. Guan, *On the convergence of stochastic dual dynamic programming and related methods*, Stochastic Programming E-Prints (2008).
- [Por08] A. Porchet, *Problèmes de valorisation et d'organisation dans les marchés d'énergie*, Ph.D. thesis, CREST Paris – EDF R&D, 2008.
- [PP91] M.V.F. Pereira and L.M.V.G. Pinto, *Multi-stage stochastic optimization applied to energy planning*, Mathematical Programming **52** (1991), 359–375.
- [RG92] T.A. Røtting and A. Gjelsvik, *Stochastic dual dynamic programming for seasonal scheduling in the norwegian power system*, IEEE Transactions on Power Systems **7** (1992), 273–279.
- [Roc74] R.T. Rockafellar, *Augmented lagrange multiplier functions and duality in nonconvex programming*, SIAM Journal on Control and Opt. **12** (1974), 268–285.
- [Rus03] A. Ruszczyński, *Handbook of stochastic optimization*, ch. Decomposition Methods, Elsevier Science Publishers B.V., 2003.
- [RW91] R.T. Rockafellar and R. Wets, *Scenarios and policy aggregation in optimization under uncertainty*, Mathematics of operations research **16** (1991), no. 1, 119–147.
- [Sha03] A. Shapiro, *Handbook of stochastic optimization*, ch. Monte Carlo sampling methods, Elsevier Science Publishers B.V., 2003.
- [Str04] C. Strugarek, *On the Fortet-Mourier metric for the stability of Stochastic Optimization Problems, an example*, Stochastic Programming E-Print Series (SPEPS) **2004-25** (2004).
- [Tur80] A. Turgeon, *Optimal operation of multi-reservoir power systems with stochastic inflows*, Water Resources Research (1980).

Chapitre 6

Conclusions et Perspectives

Sommaire

6.1 Contributions de la thèse	117
6.2 Perspectives	118

Au cours de cette thèse, nous nous sommes intéressés à la résolution de problèmes non-différentiables de grande taille de la forme

$$\begin{cases} \min & f(x), \\ \text{t.q.} & x \in \mathbb{R}^N \end{cases}$$

où f est une fonction non-partout différentiable.

L'une des principales motivations des algorithmes d'optimisation non-différentiable réside dans la résolution de problèmes obtenus par relaxation lagrangienne de certaines contraintes. Cette manipulation vise à remplacer un problème initial difficile par un problème dit dual ayant de bonnes propriétés, comme la séparabilité. Par exemple, pour un problème de gestion de production d'électricité posé sur un arbre de scénarios, le fait de relâcher la contrainte de satisfaction de la demande permet de découpler les actifs du portefeuille. Dans ce cas particulier, le problème dual est de la forme

$$\begin{cases} \min & f(x) = \sum_{i=1}^{Nu} f^i(x), \\ \text{t.q.} & x \in \mathbb{R}^N \end{cases}$$

où les fonctions f^i sont convexes et éventuellement non-différentiables.

Le problème dual résultant de la relaxation Lagrangienne est structurellement plus simple que le problème initial. Toutefois, il peut malgré tout être également de grande taille et s'avérer très difficile à résoudre. Le fil rouge de cette thèse a été de considérer des évolutions possibles pour que les algorithmes d'optimisation non-différentiables classiques puissent être appliqués à des problèmes de grande taille.

6.1 Contributions de la thèse

Après une courte introduction, nous avons présenté au chapitre 2 les notions de base de l'optimisation non-différentiable. Nous avons également proposé un aperçu de travaux récents dans ce domaine, portant notamment sur des approches avec évaluations inexactes ou des stratégies incrémentales.

Une première limite de l'optimisation non-différentiable porte sur la dimension du problème dual, ici notée N . Lorsque cette valeur est très élevée, les temps de calcul pour

évaluer la fonction duale peuvent être prohibitifs et les algorithmes classiques peuvent échouer à exhiber une solution du problème en un temps raisonnable. Dans le chapitre 3, nous avons présenté une approche dynamique de la relaxation Lagrangienne afin de traiter ce cas de figure : afin de réduire la dimension de l'espace dual, seul un sous-ensemble de contraintes est dualisé, tandis que les autres contraintes sont ignorées. Une procédure de séparation doit permettre de mettre à jour dynamiquement ce sous-ensemble, en sélectionnant quelques contraintes violées par l'itéré primal courant. Une bonne gestion des contraintes entrantes ou sortantes est cruciale pour garantir la convergence du processus. Nos travaux ont permis d'isoler les propriétés minimales à respecter afin de garantir la convergence de la méthode lorsqu'un algorithme classique de sous-gradient est employé pour la mise à jour des variables duales. Nous avons ainsi complété les résultats antérieurs de Belloni et Sagastizábal concernant les méthodes de faisceau.

En plus de la dimension N du problème, une seconde difficulté peut venir de l'évaluation des Nu sous-fonctions f^i pour un problème séparable. Ces sous-fonctions peuvent être elles-même complexes, où très nombreuses. Dans ce cas de figure, il peut être intéressant d'appliquer une stratégie incrémentale visant à mettre à jour les variables duales sans attendre l'évaluation de toutes les sous-fonctions. Dans le chapitre 4, nous avons envisagé l'application d'une méthode de faisceau dans ce contexte. Les résultats numériques, sur un problème réel de gestion de production, semblent démontrer le potentiel d'une telle approche.

Enfin, dans le chapitre 5 nous nous sommes intéressés plus spécifiquement à la gestion de production d'électricité. Après une présentation générale de la problématique, nous avons illustré l'application des méthodes de faisceau pour résoudre ce problème de grande taille posé sur un arbre de scénarios. Nous avons utilisé pour cela un logiciel développé par la R&D de EDF utilisant une méthode de faisceau proximale pour coordonner un problème rendu séparable par relaxation de la contrainte de demande. Nos travaux ont permis de proposer des améliorations pour le code de coordination actuellement utilisé. Nous avons également suggéré une heuristique permettant la valorisation des réserves d'énergie à partir des commandes optimales obtenues sur l'arbre de scénarios.

6.2 Perspectives

Nos travaux offrent plusieurs possibilités d'extensions. Celles-ci portent sur des aspects académiques ou sur des applications pour la gestion de production d'électricité.

Nos résultats théoriques sur la relaxation Lagrangienne dynamique sont relatifs aux méthodes de sous-gradient classiques. Récemment, Nesterov et Ruszczyński ont proposé des méthodes de sous-gradient primales/duales différentes offrant plus de souplesse que les méthodes classiques. Il serait intéressant d'étudier le comportement de ces approches dans un contexte dynamique.

Parallèlement à nos travaux théoriques sur la relaxation dynamique, nous avons adapté notre code de faisceau pour prendre en compte une version dynamique de la relaxation Lagrangienne. Nos applications numériques ont porté sur des problèmes de gestion de production, peu adaptés pour cette approche. Il serait intéressant d'utiliser ce code pour résoudre certains problèmes en nombres entiers ayant fait l'objet d'une application de la relaxation Lagrangienne dynamique avec méthode de sous-gradient dans la littérature.

Concernant les méthodes de faisceau incrémentales, nous n'avons pas abordé la question du choix des sous-fonctions à évaluer à chaque itération. Nous avons suivi des stratégies systématiques et il serait sûrement judicieux d'essayer d'utiliser les informations disponibles à chaque itération pour choisir les fonctions pouvant apporter le plus d'information au faisceau. Nous pensons également que les résultats théoriques de convergence pourraient être améliorés en imposant certaines hypothèses sur la gestion du paramètre proximal et sur la fréquence avec laquelle une sous-fonction doit être évaluée.

Enfin, nos travaux numériques sur le code de gestion de production SOPRANO nous ont amenés à faire différentes propositions d'améliorations. Parmi celles-ci, l'heuristique permettant une valorisation des réserves d'énergie, assimilable à une stratégie de gestion locale, semble s'avérer intéressante pour EDF. Les tests numériques sur cette partie ont montré que cette approche permettait d'améliorer le coût moyen de gestion en simulation. Notre heuristique pourrait également être enrichie afin de rendre plus robustes les stratégies ainsi obtenues. Le logiciel SOPRANO ayant comme objectif principal la gestion coordonnée des réserves nucléaires, il pourrait par exemple s'avérer pertinent d'inclure la description des réserves hydrauliques agrégées dans l'état du système lors de la programmation dynamique finale. Cela supposerait d'effectuer une programmation dynamique en dimension 3, ce qui reste raisonnable avec les moyens de calculs actuels.

Cette thèse s'intéresse à la résolution de problèmes d'optimisation non-différentiable de grandes tailles résultant le plus souvent d'une relaxation Lagrangienne d'un problème difficile. Cette technique est couramment utilisée pour appréhender des problèmes linéaires avec nombres entiers ou des problèmes convexes complexes. Le problème dual obtenu est non-différentiable - éventuellement séparable - et peut être résolu par exemple par un algorithme de faisceau. Le Chapitre 2 propose une revue de littérature des méthodes d'optimisation non-différentiable. Dans certaines situations, le problème dual peut être lui-même très difficile à résoudre et nécessiter des stratégies adaptées.

Par exemple, lorsque le nombre de contraintes dualisées est très élevé, une dualisation explicite peut s'avérer impossible ou la mise à jour des variables duales peut échouer. Au Chapitre 3, nous étudions les propriétés de convergence lorsqu'une relaxation Lagrangienne dynamique est effectuée : seul un sous-ensemble de contraintes est dualisé à chaque itération, ce qui permet de réduire la dimension du problème dual.

Une autre limite de la relaxation Lagrangienne peut apparaître lorsque la fonction duale est séparable en un grand nombre de sous-fonctions, ou que celles-ci restent difficiles à évaluer. Une stratégie naturelle consiste alors à tirer partie de la structure séparable en effectuant des itérations duales en n'ayant évalué qu'un sous-ensemble des sous-fonctions. Au chapitre 4, nous proposons d'utiliser une méthode de faisceau dans ce contexte incrémental.

Enfin, le Chapitre 5 présente des applications numériques sur des problèmes de gestion de production d'électricité.

This manuscript deals with large-scale non-smooth optimization that may typically arise when performing Lagrangian relaxation of difficult problems. This technique is commonly used to tackle mixed-integer linear programming - or large-scale convex - problems. The resulting dual problem deals with a non-smooth - often separable - function which can be maximized with, for example, bundle algorithms. Chapter 2. gives a general insight on nonsmooth optimization. However, in some situations, the dual problem may still be very hard to solve.

For instance, when the number of dualized constraints is very large, explicit dualization may no longer be possible or the update of dual variables may fail. In Chapter 3, we study the convergence properties of a dynamic Lagrangian relaxation scheme where a restricted set of constraints is dualized along the iterations, hence reducing the dimension of the dual problem.

Another limit of Lagrangian relaxation may appear when the dual function is separable in highly numerous or complex subfunctions. In such situation, the computational burden of solving all local subproblems may be preponderant in the whole iterative process. A natural strategy would be here to perform a dual iteration after having evaluated only a subset of the dual local subfunctions, instead of all of them. In Chapter 4, we apply this incremental approach with a proximal bundle method.

The last Chapter presents numerical applications on EDF's mid-term power generation planning problem.

Mots Clés : Optimisation non-différentiable, Méthodes de faisceau, Méthodes incrémentales, Méthodes inexactes, Gestion de production, Optimisation stochastique.