Relaxations in mixed-integer quadratically constrained programming and robust programming

Guanglei Wang

To cite this version:

Guanglei Wang. Relaxations in mixed-integer quadratically constrained programming and robust programming. Discrete Mathematics [cs.DM]. Institut National des Télécommunications, 2016. English. NNT : 2016TELE0026 . tel-01454964

Spécialité : Informatique

Ecole doctorale : Informatique, Télécommunications et Electronique de Paris

Présentée par

Guanglei Wang

Pour obtenir le grade de
DOCTEUR DE TELECOM SUDPARIS

Relaxations en programmation mixte en nombres entiers avec contraintes quadratiques et en programmation robuste

Soutenue le 28 Novembre, 2016
devant le jury composé de :

Rapporteur :	Dritan Nace	Professeur, Heudiasyc-CNRS, Université de Technologies de Compiègne
Rapporteur :	Leo Liberti	Directeur de recherche, LIX-CNRS, École polytechnique
Examinateur :	Sourour Elloumi	Professeur, ENSTA ParisTech
Examinateur :	Evripidis Bampis	Professeur, LIP6-CNRS, Université Pierre et Marie Curie
Examinateur :	José Neto	Maître de Conférences, SAMOVAR-CNRS, Télécom SudParis
Encadrant de thèse :	Adam Ouorou	Chercheur, HDR, Orange Labs Recherche
Directeur de thèse :	Walid Ben-Ameur	Professeur, SAMOVAR-CNRS, Télécom SudParis

Numéro National de Thèse (NNT) : 2016TELE0026

Speciality: Computer Science

Ecole doctorale : Informatique, Télécommunications et Electronique de Paris

Presented par

Guanglei Wang

For obtaining the
DOCTOR OF PHILOSOPHY DEGREE OF TELECOM SUDPARIS

Relaxations in Mixed-Integer Quadratically Constrained Programming and Robust Programming

Defense on 28 November, 2016
Defense Committee:

Reviewer:	Dritan Nace	Professor, Heudiasyc-CNRS, Université de Technologies de Compiègne
Reviewer:	Leo Liberti	Research Director, LIX-CNRS, École polytechnique
Examiner:	Sourour Elloumi	Professor, ENSTA ParisTech
Examiner:	Evripidis Bampis	Professor, LIP6-CNRS, Université Pierre et Marie Curie
Examiner:	José Neto	Assistant Professor, SAMOVAR-CNRS, Télécom SudParis
Advisor:	Adam Ouorou	Researcher, HDR, Orange Labs Research
Advisor:	Walid Ben-Ameur	Professor, SAMOVAR-CNRS, Télécom SudParis

National Number of Thesis (NNT) : 2016TELE0026

Declaration

I declare that this thesis was composed by myself and that the work contained therein is my own, except where explicitly stated otherwise in the text.

Guanglei Wang

Acknowledgement

First of all, I would like to express my sincere gratitude and appreciation to my thesis advisors, Walid Ben-Ameur and Adam Ouorou, for their invaluable guidance, constant encouragement, and endless patience during my PhD study. Thanks to them, I have the opportunity to conduct and complete my thesis in Orange Labs and Telecom SudParis. I will be forever honored and grateful for working with them.

My gratitude extends to Prof. Dritan Nace and Prof. Leo Liberti for their time in reviewing my manuscript and insightful comments. I also would like to thank Prof. Evripidis Bampis, Prof. Sourour Elloumi, Dr. José Neto for accepting the invitation to participate in the defense. At the initial stage of my PhD study, I got quite valuable suggestions and encouragement from José, and for that I am really grateful.

It is a good opportunity to thank Eric Gourdin and Nabil Benameur for their trust and unlimited support, and other colleagues, Pierre Bauguion, Amal Benhamiche, Matthieu Chardy, Yannick Carlinet, Bruno Kauffmann, Ruby Krishnaswamy, Luca Muscariello, Philippe Olivier, Nancy Perrot, Alain Simonian, Christian Tanguy, for their constant help during my stay in Orange Labs.

To all my friends, Vincent Angilella, Paul Beaujean, Mathieu Besson, Antoine Glorieux, Claudio Imbrenda, Thibaut Lefebvre, Wuyang Li, Jakub Mareček, Léonce Mekinda, Yassine Naghmouchi, Felipe Olmos, Bobby Ung, Yuhui Wang, Xuan Zeng, etc., thank you for the wonderful discussions about new things, coffee breaks, dinners and jokes. Your optimism, kindness and perseverance made my PhD experience unique and unforgettable.

Finally, I would like to express my deepest gratitude and appreciation to my parents for their relentless support and unconditional love. Their kindness, affection, generosity, diligence encourage me to follow my convictions and passions. My dearest wife, Jing Yang, your love, support, and your faith in me fuel me to persevere in the face of difficulties. I am indebted to you so much that words can hardly describe.

Résumé

Contexte technologique

Le Cloud Computing, ou informatique dans le nuage, est une nouvelle tendance porteuse dans les technologies de l'information et de la communication. La virtualisation et l'utilisation des réseaux sont les technologies clés pour mettre en place une infrastructure de cloud computing. Le marché du cloud computing, qui permet de payer à l'usage de ressources de calcul et/ou de stockage connait une croissance forte. Dans ce paradigme, les serveurs de calcul ou de stockage sont partagés par plusieurs utilisateurs/clients de manière sécurisée au travers de l'utilisation de machines virtuelles (VM). L'intérêt pour les clients est la réduction de leurs investissements dans des ressources informatiques physiques en payant les ressources suivant leur utilisation. Pour l'Opérateur, le partage d'une infrastructure commune permet de mutualiser les coûts, ce qui se reflète dans des prix compétitifs. L'économie d'échelle permet aussi à l'Opérateur de réduire ses coûts et sa capacité à utiliser un nombre minimum de ressources physiques pour assurer un service à un grand nombre de clients, est essentiel pour lui assurer du profit.

Un problème central pour mettre en place une infrastructure de cloud computing est la gestion des data centers qui abritent les machines physiques. On considère ici que les data centers sont constitués d'un certain nombre de machines physiques constituées en groupes. L'organisation des data centers suit une architecture en trois tiers suivant la hiérarchie des switchs du réseau (accès, agrégation et core), architecture la plus couramment utilisée de nos jours, voir [83]. Typiquement, un utilisateur demande à exécuter une application qui consiste en une demande d'un nombre donné de machines virtuelles avec des demandes de ressources. La première phase du déploiement d'une application est de décider de l'affectation des machines virtuelles constituant l'application à un certain nombre de machines physiques. Généralement, ce placement se fait en optimisant un objectif (maximiser le nombre de machines physiques libres) sous des contraintes exprimant les besoins des utilisateurs et les contraintes des administrateurs du data center. Cependant, la définition de ce problème évolue avec les évolutions technologiques (possibilité de migrer les machines virtuelles dans le data center sans interrompre le service, méthodes avancées de contrôle de la consommation énergétique, etc), et l'amélioration des pratiques. De nouvelles préoccupations continuent d'apparaître, notamment sur l'évolution des fonctionnalités comme les demandes de VLANs (Virtual Local Area Networks) pour supporter un groupe de machines virtuelles, Le cloud computing a quatre composantes principales: utilisateurs, allocations des ressources, machines virtuelles et machine physiques.

Généralement, ce placement se fait en optimisant un objectif (maximiser le nombre de machines physiques libres) sous des contraintes exprimant les besoins des utilisateurs et les contraintes des administrateurs du data center. Cependant, la définition de ce problème évolue avec les évolutions technologiques (possibilité de migrer les machines virtuelles dans le data center sans interrompre le service, méthodes avancées de contrôle de la consommation énergétique, etc), et l'amélioration des pratiques.De nouvelles préoccupations continuent d'apparaître, notamment sur l'évolution des fonctionnalités comme les demandes de VLANs (Virtual Local Area Networks) pour supporter un groupe de machines virtuelles, Le cloud computing a quatre composantes principales: utilisateurs, allocations des ressources, machines virtuelles et machine physiques.

L'utilisateur soumet une demande de service qui contient les demandes détaillées de ressources (puissance de calul, mémoire, etc.). Ces ressources sont ensuite provisionnées en forme de machines virtuelles en temps réel suivant les disponibilités et un accord sur les niveaux de service selon le SLA (Service Level Agreement). Enfin, les machines virtuelles sont allouées à un ensemble de machines physiques qui exécutent la demande. Ce processus prend aussi en compte les facteurs économiques. Ce dernier concerne surtout l'opérateur dont l'objectif est de maximiser son revenu ou bien minimiser son coût opérationnel. Par exemple, réduire le nombre de serveurs physiques utilisés permet d'éteindre ceux qui ne sont pas utilisés, ou bien rentabiliser au maximum l'usage de ses serveurs physiques. Ici nous nous intéressons en particulier à l'allocation des machines virtuelles aux machines physiques. En partant d'un placement statique, un modèle devra être construit pour allouer un ensemble donné de machines virtuelles à des machines physiques. Ce placement devra prendre en compte les ressources de calcul et la capacité du réseau (mais pas les temps de calcul). L'objectif du problème est de minimiser un coût d'affectation et de routage en prenant en compte les contraintes matérielles.

Modèle mathématique

Le problème précédemment décrit se modélise sous la forme du programme mixte en nombre entier avec des contraintes quadratiques. Nous introduisons la notation suivante pour le développement de formulations mathématiques.

- R ensemble des requêtes virtuelles.
- $H = (S, E)$ H graphe physique.
- $G^r = (V^r, L^r)$ le graphe de la requête virtuelle r.
- S ensemble des serveurs.
- E ensemble des arcs du graphe H.
- V^r ensemble des machines virtuelles de la requête r.
- L^r ensemble des arcs de G^r.
- C^{ri} demande en ressources de calcul CPU de la VM i de V^r.

- M^{ri} mémoire requise pour VM i de V^r.
- C_k capacité de calcul du serveur k.
- M_k mémoire disponible au serveur k.
- F_k coût fixe du serveur k.
- A_k coût additionnel linéaire induit par la charge CPU du serveur k.
- F^{rij} demande de débit du lien virtuel (i, j).
- B_e capacité de transmission (bande passante) du lien e.
- W_e coût fixe du lien e.
- P_{kp} le plus court chemin pour $k - p$, $(k, p) \in S \times S : k \neq p$.
- x_k^{ri} 1 si le VM i de la requête r est affectée au serveur k.
- θ_k 1 si le serveur k est utilisé.
- ϕ_e 1 si le lien e est utilisé.

Le modèle résultant est donné ci-dessous:

$$\min \quad \sum_{k\in\mathcal{S}} F_k\theta_k + \sum_{k\in S} A_k \sum_{r\in R}\sum_{i\in V^r} c^{ri}x_k^{ri} + \sum_{e\in E} W_e\phi_e. \tag{$\mathbb{P}$}$$

$$\text{s.t.} \quad \sum_{k\in S} x_k^{ri} = 1, \qquad \forall r \in R, i \in V^r, \tag{AC}$$

$$\sum_{i\in V^r} x_k^{ri} \leq \theta_k, \qquad \forall r \in R, \forall k \in S, \tag{LC}$$

$$\sum_{r\in R}\sum_{i\in V^r} c^{ri}x_k^{ri} \leq C_k\theta_k, \qquad \forall k \in S, \tag{KP}$$

$$\sum_{r\in R}\sum_{i\in V^r} m^{ri}x_k^{ri} \leq M_k\theta_k, \qquad \forall k \in S, \tag{KP'}$$

$$\sum_{r\in R}\ \sum_{\substack{k,p\in S:\\ k\neq p, e\in P_{kp}}}\ \sum_{(i,j)\in L^r} f^{rij}x_k^{ri}x_p^{rj} \leq B_e\phi_e, \ \ \forall e \in E, \tag{QC}$$

$$\theta_k, \phi_e, x_k^{ri} \in \{0, 1\}, \forall r \in R, i \in V^r, k \in S, e \in E.$$

L'objectif est de minimiser le coût total additif composé de trois termes: les coûts fixes encourus par commutation sur les serveurs, le coût supplémentaire provenant de la charge CPU, et les coûts fixes de l'utilisation des liens. Nous modélisons le coût supplémentaire induit par la charge CPU comme une fonction linéaire de la charge tandis que la mémoire est indépendante de la charge. Les Contraintes (AC) signifient que chaque machine virtuelle doit être affectée à un seul serveur.

(LC) modélise le fait que les machines virtuelles d'une même requête sont généralement affectées à des serveurs différents pour des raisons pratiques (sécurité, fiabilité). Les contraintes (KP, KP') expriment le fait que la mémoire et le CPU disponibles au niveau de chaque serveur sont limités. Les contraintes (QC) sont liées à la limitation de la bande passante. Ce problème est donc clairement un programme en 0,1 à contraintes quadratiques que nous nous proposons de résoudre.

Notons également que les paramètres du problème peuvent être incertains. Il est en effet difficile de déterminer à l'avance les débits requis pour chaque demande. Ceci nous incite donc à envisager les techniques d'optimisation robuste.

Outils mathématiques

Nous nous intéressons donc aux programmes mathématiques du type (1)

$$\begin{array}{ll} \min & \mathbf{c}_0^T\mathbf{x} \\ \text{s.t.} & q_i(\mathbf{x}) \leq \mathbf{0},\ i = 1, \dots, m, \\ & \mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n. \end{array} \tag{1}$$

Où l'ensemble $q_i(\mathbf{x}), \mathcal{X}$ sont donné par $q_i(\mathbf{x}) = \mathbf{x}^T\mathbf{Q}_i\mathbf{x} + \mathbf{c}_i^T\mathbf{x} + d_i$ $(i = 0, \dots, m)$, $\mathcal{X} = \big\{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} \leq \mathbf{b},\ x_j \in \mathbb{Z}, j \in J\big\}$ et $J \subseteq \{0, \dots, n\}$. Les c_i sont des vecteurs de dimension n, et les d_i sont des scalaires.

On suppose que $\mathbf{K} := \mathcal{X} \cap \{\mathbf{x} : q_i(\mathbf{x}) \leq \mathbf{0}\ (i = 1, \dots, m)\}$ est compact et non-vide. Ces programmes sont notés MIQCP. Il n'est pas trivial de relaxer ces problèmes MIQCP sous la forme de programmes convexes dans l'espace d'origine. En passant par l'idée du lifting (élevant le problème dans un espace de plus grande dimension), on peut reformuler (1) en ajoutant des variables supplémentaires donnant ainsi (2).

$$\begin{array}{ll} \min & \mathbf{c}_0^T\mathbf{x} \\ \text{s.t.} & \langle \mathbf{Q}_i, \mathbf{X} \rangle \leq \mathbf{0},\ i = 1, \dots, m, \\ & \mathbf{x} \in \mathcal{X}, \\ & \mathbf{X} = \mathbf{x}\mathbf{x}^T, \end{array} \tag{2}$$

Où $\mathbf{X}$ est une matrice réelle de taille $n \times n$. Notons que la contrainte $\mathbf{X} = \mathbf{x}\mathbf{x}^T$ n'est pas convexe. Une approche classique consiste à relâcher cette contrainte et rajouter des inégalités valides pour renforcer la relaxation convexe. Ceci inclut l'approche de McCormick [119] où la contrainte $X_{ij} = x_i x_j$ sur le domaine rectangulaire $x_i \in [l_i, u_i], x_j \in [l_j, u_j]$ est remplacée par les inégalités

$$\max\{u_i x_j + u_j x_i - u_i u_j, l_i x_j + l_j x_i - l_i l_j\} \leq X_{ij} \leq \min\{l_i x_j + u_j x_i - l_i u_j, l_j x_i + u_i x_j - u_i l_j\}$$

Une autre linéarisation a été proposée par Glover [74] utilisant de l'ordre de $\mathcal{O}(n)$ variables. Citons également la technique de reformulation-linéarisation de Sherali et Adams [145]. D'autres inégalités peuvent être obtenues dynamiquement par les techniques de programmation disjonctive [17, 16].

Des relaxations non-linéaires peuvent être également considérées. Il est en effet possible de relaxer la contrainte $\mathbf{X} = \mathbf{x}\mathbf{x}^T$ en la remplaçant par: $\mathbf{X} - \mathbf{x}\mathbf{x}^T \succeq 0$. Cette relaxation peut être également obtenue par relaxation lagrangienne. Une autre technique consiste à convexifier les fonctions quadratiques. En effet, en considérant une matrice $\mathbf{C}_i$ semi-définie-positive, la contrainte

$$q_i(\mathbf{x}) = q_i(\mathbf{x}) = \mathbf{x}^T\mathbf{Q}_i\mathbf{x} + \mathbf{c}_i^T\mathbf{x} + d_i \leq 0$$

est relâchée grâce à

$$\begin{aligned} \langle \mathbf{x}\mathbf{x}^T, \mathbf{C}_i\rangle + \mathbf{c}_i^T\mathbf{x} + \langle \mathbf{Q}_i - \mathbf{C}_i, \mathbf{X}\rangle + d_i \leq 0, \\ \mathbf{X} \succeq \mathbf{x}\mathbf{x}^T. \end{aligned}$$

Il est facile de prouver que déterminer la meilleure convexification se ramène à la résolution d'un programme semi-défini-positif ($\mathbf{X} - \mathbf{x}\mathbf{x}^T \succeq \mathbf{0}$) [39].

Une autre méthode pour obtenir des bornes pour le problème d'origine consiste à relâcher lagrangiènement certaines contraintes en les intégrant dans la fonction objectif et leur associant un ensemble de multiplicateurs exprimant le poids donné à ces contraintes de relâchées. Il s'agit ensuite de résoudre le problème dual qui consiste à trouver les meilleurs multiplicateurs maximisant la fonction objectif pénalisée. Un compromis est à trouver entre la difficulté de résoudre le problème d'évaluation (pour des multiplicateurs données) et la qualité de la borne.

D'autres paradigmes ont été récemment proposés, tel que la programmation copositive. Il s'agit de remplacer la contrainte $\mathbf{X} - \mathbf{x}\mathbf{x}^T = \mathbf{0}$ par une contrainte du type: $\mathcal{S}(\mathbf{x}, \mathbf{X}) \in \mathrm{CP}$, Où $\mathcal{S}(\mathbf{x}, \mathbf{X}) = \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \mathbf{X} \end{pmatrix}$ et CP est le cône des matrices complètement positives qui sont les matrices qui s'écrivent sous la forme $\mathbf{B}\mathbf{B}^T$ où $\mathbf{B}$ est une matrice réelle non négative. Le cône dual du cône des matrices complètement positives est le cône des matrices copositives qui sont les matrices $\mathbf{X}$ vérifiant $\mathbf{v}^T\mathbf{X}\mathbf{v} \geq \mathbf{0}$ pour tout vecteur $\mathbf{v}$ non négatif.

Un autre paradigme utilisé dans le contexte de programmation polynomiale (incluant donc la programmation quadratique) est l'approche SoS (somme de carrés) de Lasserre [101, 102] et Parrilo [135, 136]. Il s'agit de remplacer la contrainte de positivité d'un polynôme par une contrainte imposant que ce polynôme est la somme de carrés de polynômes. Or si on se restreint à des sommes de carrés de polynômes de degré inférieur à une constante, on peut déterminer si un polynôme s'écrit sous la forme d'une telle somme en résolvant un programme semi-défini-positif.

L'autre outil mathématique que nous utilisons dans cette thèse pour traiter l'incertitude est l'optimisation robuste. Rappelons que les paramètres de la demande peuvent être incertains. Il est en effet difficile de déterminer à l'avance les débits requis pour chaque demande. Ceci nous incite donc à envisager les techniques d'optimisation robuste.

Nous nous intéressons à des problèmes linéaires incertains

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b}, \\ & \mathbf{x} \in \mathbb{R}^n. \end{aligned}$$

Nous supposons que $\mathbf{x}$ est divisée en $\mathbf{x} = (\mathbf{u}, \mathbf{v})$, où $\mathbf{u}$ représente les variables non ajustables et $\mathbf{v}$ les variables ajustables. La contrepartie robuste de ce problème incertain à l'étude s'écrit

$$\begin{aligned} \min_{u,v} \quad & \mathbf{c}^T\mathbf{u} \\ \text{s.t.} \quad & \mathbf{U}\mathbf{u} + \mathbf{V}\mathbf{v}(\xi) \leq \mathbf{b}, \ [\mathbf{U}, \mathbf{b}] \in \Xi, \end{aligned} \tag{FARC}$$

où les paramètres incertains sont $\mathbf{U} \in \mathbb{R}^{m \times n}$ et $\mathbf{b} \in \mathbb{R}^m$ alors que $\mathbf{V}$ et $\mathbf{c}$ sont supposées être connues. On note par $\xi \equiv [\mathbf{U}, \mathbf{b}] \in \Xi$ les paramètres incertains appartenant à l'ensemble d'incertitude fixé Ξ supposé être compact, convexe et avec un intérieur non vide. ξ sera considéré comme un vecteur.

Les variables non ajustables sont parfois interprétées comme variables du type"ici et maintenant", tandis que les variables ajustables peuvent être considérés comme des variables "attend et regarde". Cette contrepartie robuste ci-dessus est généralement appelé contrepartie robuste entièrement ajustable (FARC). FARC est parfois appelé la contrepartie robuste dynamique puisque $\mathbf{v}$ dépend de ξ. FARC peut être considérée comme un problème d'optimisation à deux étages où $\mathbf{u}$ sont les variables de premier étage et $\mathbf{v}$ sont les variables de deuxième étape.

Pour obtenir un problème d'optimisation traitable, plusieurs approches ont été proposées. Si nous limitons v à une fonction constante par rapport aux paramètres incertains, FARC devient simplement la contrepartie robuste standard statique notée SRC. Si les variables ajustables v dépendent affinement de ξ:

$$\mathbf{v}(\xi) = \mathbf{w} + \mathbf{W}\xi, \quad \xi \equiv [\mathbf{U}, \mathbf{b}] \in \Xi,$$

où $\mathbf{w}$ et $\mathbf{W}$ sont à optimiser, on obtient l'approche (AARC). Si $\mathbf{v}$ est exprimé en un polynôme en ξ ayant une degré ne dépassant pas une constante fixe, le problème de la contrepartie solide est alors lié au test de positivité d'un polynôme. Ensuite, on peut utiliser des techniques connexes, par exemple, la hiérarchie sos mentionné précédemment pour résoudre le problème. Il y a aussi d'autres approches, telles que, les politiques affines par morceaux, les approches multistatiques [27], l'adaptabilité finie [33].

Même si un grand nombre de propositions en optimisation robuste sont apparues, il y a encore des défis à relever. Tout d'abord, à notre connaissances, aucune approche n'est suffisamment générale pour englober la robustesse statique, la robustesse affine et robustesse entièrement ajustable ou dynamique décrites précédemment. Deuxièmement, comme observé dans [37], il n'y a pas de façon systématique pour jouer sur le compromis entre la performance des politiques qui en découlent et de la complexité de la politique. Troisièmement, les paramètres incertains d'un problème d'optimisation peuvent être parfois difficiles à observer. Dans plusieurs applications, seul un sous-ensemble de ces paramètres ou certains agrégats d'entre eux peut être observé.

Motivations et Contributions

Motivations

Comme mentionné précédemment, le problème d'affectation des machines virtuelles fait intervenir des contraintes quadratiques en 0-1. Sa structure combinatoire est assez complexe car il contient l'affectation quadratique et les problèmes de sac à dos quadratique.

La possibilité de traiter ce problème repose en grande partie sur le développement de techniques mixtes de la programmation entière avec contraintes quadratiques. Cependant, comme mentionné dans [52, 50, 51], même si des progrès importants ont été réalisés, les résultats "révolutionnaires" sont encore à venir et de nombreux problèmes fondamentaux ne sont pas traitées. En effet, la relaxation linéaire standard conduit à un grand écart, tandis que les techniques comme la relaxation semi-définie est informatiquement coûteuse dans la procédure de Branch-and-bound. Ainsi, le développement des techniques et des programmes de relaxation efficaces pour le modèle déterministe est la tâche centrale pour le problème d'affectation des machines virtuelles de nuages.

MIQCP couvre non seulement le problème d'affectation des machines virtuelles, mais aussi un large éventail d'autres applications et des problèmes de recherche, par exemple, des problèmes d'affectation quadratique [81, 153], les problèmes max-cut [92], et d'autres applications [70, 13, 125]. Nous nous sommes aussi intéressés à revoir les techniques de relaxation générale et établir des résultats théoriques pour les problèmes MIQCP.

Motivé par le problème d'affectation avec des exigences incertaines, nous déployons des efforts visant à développer un nouveau paradigme général pour les programmes linéaires avec paramètres incertains dans le cadre de l'optimisation robuste. Les Programmes linéaires avec des paramètres incertains peuvent être très difficiles si l'on considère des approches complètement dynamiques (du type FARC). Ainsi, des politiques plus restrictives doivent être proposées. En outre, dans la pratique, les paramètres incertains d'un problème d'optimisation peuvent être parfois difficiles à observer. Alors, comment intégrer l'incertitude d'une manière suffisante tout en permettant que des décisions soient prises sur la base d'informations incomplètes ? Y'a t-il un moyen systématique de faire un compromis entre la complexité de calcul et la qualité de la solution ?

Contributions

Les principaux résultats de cette thèse sont structurés en trois thèmes, à savoir la procédure de solution du problème d'affectation des machines virtuelles au chapitre 3, le calcul d'enveloppe convexe et d'enveloppe concave des fonctions générales bilinéaires au chapitre 4, une nouvelle hiérarchie générale de relaxations dans le contexte d'optimisation robuste au chapitre 5.

Plus précisément, le chapitre 3 est consacré à plusieurs techniques de relaxation qui accélèrent la procédure de solution du problème d'affectation de machines virtuelles. Dans le chapitre suivant, nous examinons quelques méthodes d'approximation pour la construction de sous-estimateurs convexes d'une fonction bilinéaire, qui est utile pour la construction d'une relaxation de la version étendue du modèle d'affectation des machines virtuelles ($\mathbb{P}$). Pour résoudre le problème de l'incertitude, nous proposons un nouveau paradigme flexible et efficace appelé optimisation robuste multipolaire dans le chapitre 5. Dans ce qui suit, nous donnons plus de détails sur les contributions de la thèse.

Dans le Chapitre 3, nous nous concentrons la résolution du problème d'affectation des machines virtuelles. Tout d'abord, nous proposons de reformuler le problème par le biais des techniques de linéarisation classiques permettant de le transformer en MILP (programme linéaire à variables mixtes). Nous avons implémenté le relaxation de McCormick ainsi que la relaxation de Glover. Les résultats numériques montrent que, même si le modèle basé sur la relaxation de Glover a moins variables que celui basé sur la relaxation de McCormick, ses performances de calcul ne sont pas beaucoup mieux que ce dernier. La qualité de la relaxation semble être l'élément le plus déterminant. On remarque que nous pouvons appliquer les inégalités RLT pour renforcer le modèle basé sur la relaxation de McCormick et de supprimer un certain nombre de contraintes redondantes. D'un autre côté, il peut être compliqué d'appliquer RLTs au modèle basé sur la linéarisation de Glover. Par conséquent, nous allons nous concentrer sur la linéarisation McCormick. Ensuite, nous utilisons la RLTs et nous ajoutons des inégalités valides pour renforcer le modèle McCormick. Notre premier résultat montre que les inégalités RLT peuvent éliminer de nombreuses inégalités McCormick redondantes. En outre, nous avons proposé deux types d'inégalités valides en exploitant la structure du problème. Dans l'ensemble, plusieurs formulations sont proposées et ils sont codés en C ++ et résolus par CPLEX 12.6.3. Leurs performances de calcul sont évaluées sur certains cas de problème. Les résultats numériques montrent que la combinaison des types d'inégalités valides peut accélérer sensiblement la résolution du problème. Nous renvoyons le lecteur au Tableau 3.2 pour plus de détails.

Alors que les inégalités valides sont souvent très efficaces pour le renforcement de la relaxation linéaire, il y a des cas où l'écart de relaxation peut être de 20%(ce qui peut rendre le branch-and-bound coûteux en temps de calcul). Ainsi, nous avons proposer des relaxations lagrangiennes pour décomposer le problème résolvant ainsi le sous-problème relâché de manière efficace avec des bornes inférieures de bonne qualité. En particulier, nous proposons une décomposition qui conduit à une séquence de sous problèmes associés à chaque demande, chaque serveur, et chaque lien. Nous donnons en outre un résultat géométrique sur la force de la borne lagrangienne montrant qu'elle est généralement beaucoup plus forte que la borne de la relaxation continue. Une hiérarchie de relaxation est également proposée en considérant une séquence de couverture de l'ensemble de la demande. Nous pouvons, par exemple, diviser l'ensemble des demandes (requêtes) en certains sous-ensembles disjoints et décomposer le problème en fonction de ce partitionnement. Les bornes de la relaxation lagrangienne résultante est meilleure que celle basée sur une simple décomposition des demandes. Pour résoudre le problème dual, nous utilisons l'algorithme in-out proposé par Ben-Ameur et Neto [25] avec un bon choix de des paramètres, notamment le premier point intérieur.

Nos expériences numériques montrent que: (1) Pour les petites instances de problèmes que CPLEX peut résoudre à l'optimum en un temps raisonnable, nous montrons que la décomposition lagrangienne avec in-out peut fournir rapidement des bornes quasi-optimales. (2) Pour les instances où CPLEX nécessite des heures de temps CPU pour les traiter, le schéma de décomposition lagrangienne avec in-out peut produire des bornes de bonne qualité en un temps moindre. De plus, on remarque que, en moyenne, la décomposition lagrangienne est environ dix fois plus rapide que CPLEX (avec ses paramètres standards) pour atteindre la même borne. La hiérarchie proposée est très attrayante pour les instances de problèmes avec un grand nombre de demandes, que CPLEX 12.6.3 ne peut résoudre dans la pratique.

Ensuite, nous introduisons une nouvelle formulation induite par des questions de symétries du problème. Cette formulation permet de réduire considérablement le nombre de termes bilinéaires dans le modèle, et comme prévu, semble plus efficace que les modèles précédents. On peut traiter le modèle résultant comme une reformulation de l'agrégation du modèle original. Pour les instances de problèmes qui ne peuvent pas bénéficier directement de cette nouvelle formulation, nous proposons une heuristique pour générer des solutions possibles et les bornes supérieures correspondantes.

Notez que, dans la pratique, l'arrivée des machines virtuelles est dynamique et imprévisible. Le modèle ($\mathbb{P}$) ne peut guère fonctionner dans ce cas. A la fin du Chapitre 3, nous proposons un modèle qui optimise certaines politiques d'affectation.

Dans le chapitre 4, nous discutons de quelques approches pour l'approximation des enveloppes convexes et enveloppes concaves des fonctions bilinéaires

$$f(\mathbf{x}) = \sum_{i<j} Q_{ij} x_i x_j, (1 \leq i < j \leq n),$$

en particulier sur les hypercubes en raison de leur généralité. Une propriété de la fonction bilinéaire f dûe à Sherali [146] et Rikun [140] est que son enveloppe convexe et concave sur un hypercube sont sommet-polyédriques, à savoir les enveloppes de f sur un hypercube coïncident avec les enveloppes de sa restriction aux sommets de l'hypercube. Cela nous permet de définir l'enveloppe convexe de f comme suit.

$$\min_{\lambda_i} \left\{ \sum_{i=1}^{2^n} \lambda_i f(\mathbf{v}_i) : \sum_{i=1}^{2^n} \lambda_i \mathbf{v}_i = \mathbf{x}, \sum_{i=1}^{2^n} \lambda_i = 1, \lambda_i \geq 0, i = 1, \ldots, 2^n \right\},$$

où $\mathbf{v}_i$ $(i = 1, \ldots, 2^n)$ sont les sommets d'un hypercube à n dimensions. Cependant, comme le nombre de points extrêmes d'un hypercube est exponentiel, la caractérisation complète de ces enveloppes polyédriques est en général difficile.

Nous examinons tout d'abord un estimateur basé sur un programme SDP, où on considère l'intersection d'une contrainte de SDP avec un ensemble d'inégalités quadratiques valides pour l'hypercube. Notre contribution est liée aux travaux d' Anstreicher [11]. Nous montrons que les deux variantes suivantes de formulations pour approcher l'enveloppe convexe des fonctions bilinéaires sont équivalentes.

$$\begin{aligned} \min\ & \frac{1}{2}\langle \mathbf{Q}, \mathbf{X} \rangle \\ \text{s.t.}\ & X_{ii} \leq x_i,\ (i = 1, \ldots, n), \\ & \max\{x_i + x_j - 1, 0\} \leq X_{ij} \leq \min\{x_i, x_j\},\ i < j, \\ & \mathcal{S}(\mathbf{x}, \mathbf{X}) \succeq 0, \end{aligned}$$

$$\begin{aligned} \min\ & \frac{1}{2}\langle \mathbf{Q}, \mathbf{X} \rangle \\ \text{s.t.}\ & X_{ii} = x_i,\ (i = 1, \ldots, n), \\ & \max\{x_i + x_j - 1, 0\} \leq X_{ij} \leq \min\{x_i, x_j\},\ i < j, \\ & \mathcal{S}(\mathbf{x}, \mathbf{X}) \succeq 0 \end{aligned}$$

Ensuite, nous proposons une nouvelle approche en considérant un outil prédéfini pour approcher les enveloppes convexe et concave de f sur un hypercube. Plus précisément, pour un hypercube à n dimensions, nous définissons une couverture comme étant une ensemble C tel que l'ensemble des indices $\{1,\ldots,n\}$ est couvert par set C. Ensuite, le polytope associé à la couverture C est explicitement caractérisé par

$$\mathcal{C}_C = \left\{ (\mathbf{x}, (X_{ij})_{i<j}) \in [0,1]^n \times [0,1]^{\frac{n(n-1)}{2}} : \mathbf{X}_I = \sum_{l=1}^{2^{|I|}} \lambda_I^l \mathbf{V}_I^l, \mathbf{x}_I = \sum_{l=1}^{2^{|I|}} \lambda_I^l \mathbf{v}_I^l, \sum_{k=1}^{2^{|I|}} \lambda_I^k = 1, \boldsymbol{\lambda}_I \geq \mathbf{0},\ I \in C. \right\},$$

où $\mathbf{V}_I^l = \mathbf{v}_I^l (\mathbf{v}_I^l)^T$, et $\mathbf{v}_I^l$ est un sommet de l'hypercube de dimension $|I|$.

Un estimateur convexe de fonction bilinéaire f pourrait être donné par :

$$\frac{1}{2} \min_{(\mathbf{x},(X_{ij})_{i<j}) \in \mathcal{C}_C} \langle \mathbf{Q}, \mathbf{X} \rangle.$$

Cette approche conduit à une reformulation linéaire et nous sommes en mesure d'établir des liens avec diverses inégalités et la méthode reformulation-linéarisation-technique (RLT) proposée dans [2].

Enfin, on montre que l'enveloppe convexe (resp. Concave) d'une fonction bilinéaire f est affinement équivalente à l'enveloppe concave (resp. Convexe) d'une fonction enveloppe polyédrique. Plus précisément, nous considérons la fonction polyédrique

$$\phi : [0,1]^n \ni \mathbf{x} \mapsto \phi(\mathbf{x}) = \sum_{i<j} Q_{ij} |x_i - x_j|.$$

Nous avons montré que

$$\widehat{f}(\mathbf{x}) = \frac{1}{2} \sum_{i<j} Q_{ij}(x_i + x_j) - \frac{1}{2} \check{\phi}(\mathbf{x}), \quad \forall \mathbf{x} \in [0,1]^n,$$
$$\check{f}(\mathbf{x}) = \frac{1}{2} \sum_{i<j} Q_{ij}(x_i + x_j) - \frac{1}{2} \widehat{\phi}(\mathbf{x}), \quad \forall \mathbf{x} \in [0,1]^n,$$

où $\check{f}$ $(\widehat{f})$ est l'enveloppe convexe (concave) d'une fonction f. Avec ces résultats, nous montrons que si $Q_{ij} \geq 0$, l'enveloppe concave d'une fonction bilinéaire f est entièrement caractérisé par la relaxation de McCormick.

Nos résultats numériques montrent que dans l'ensemble, les estimateurs basés sur la couverture sont généralement performants.

Après les résultats numériques, nous fournissons quelques idées pour une nouvelle approche pour approximer l'enveloppe convexe à un point $\mathbf{x}$ prédéfini lorsque tous les coefficients de f ne sont pas négatifs. L'approche est motivée par la propriété que l'enveloppe concave de la fonction ϕ est sommet-polyédrique sur toutes les polytopes arbitraires si les coefficients Q_{ij} $(i < j)$ sont non négatifs. Nous considérons un polytope qui couvre un hypercube et a moins de 2^n sommets. Ensuite, nous montrons que l'enveloppe convexe f sur ce polytope (qui peut être obtenu facilement) est une borne inférieure de l'enveloppe convexe de f. D'autres études sont nécessaires pour trouver un moyen raisonnable de définir des polytopes appropriés.

Dans le chapitre 5, nous considérons les programmes linéaires impliquant des paramètres incertains et proposons une nouvelle contrepartie robuste traitable pour approcher la stratégie dynamique (FARC). Pour introduire le nouveau paradigme, nous introduisons deux outils, à savoir, la matrice de l'ombre et l'ensemble de pôles. Une matrice de l'ombre pourrait être utilisée pour représenter le fait que les actions ou décisions sont généralement prises sur des informations incomplètes de l'incertitude. Si la matrice d'ombre est une matrice d'identité dont la taille est le nombre de paramètres incertains, nous disons que nous avons une mesure complète de l'incertitude. Désignons par $\mathbf{P} \in \mathbb{R}^{n_0 \times \dim(\Xi)}$ une matrice d'ombre, où n_0 est la dimension de l'ombre (à savoir, l'information partielle résultant) et $\dim(\Xi)$ est la dimension de l'ensemble d'incertitude Ξ. L'information résultante partielle est définie par

$$\Xi_P := \mathbf{P}\Xi \equiv \{\mathbf{P}\xi, \xi \in \Xi\}.$$

Un élément clé de l'approche multipolaire est un ensemble fini de pôles, qui sont des vecteurs dans l'espace image de la matrice d'ombre. On note Ω un tel pôle-set. Nous disons que Ω est un ensemble de pôles Ξ_P si pour tout $\xi \in \Xi$, $\mathbf{P}\xi$ appartient à l'enveloppe convexe de Ω (une combinaison convexe de pôles) notée $\operatorname{conv}\Omega$. Étant donné un ensemble Ξ_P, une collection d'ensembles de pôles Ξ_P est définie comme

$$\mathcal{F}_{\Xi_P} := \{\Omega : \Xi_P \subseteq \operatorname{conv}\Omega\}.$$

Évidemment, les points extrêmes de Ξ_P forment un possible ensemble de pôles, à savoir, $\operatorname{ext}(\Xi_P) \in \mathcal{F}_{\Xi_P}$.

Pour un $\Omega \in \mathcal{F}_{\Xi_P}$ fixé, pour chaque $\xi \in \Xi$ un coefficient positif λ_ω^ξ est associé à chaque pôle ω. Par définition d'un ensemble de pôles, le système suivant a une solution

$$\begin{aligned}
&\sum_{\omega \in \Omega} \lambda_\omega^\xi \omega = \mathbf{P}\xi,\\
&\sum_{\omega \in \Omega} \lambda_\omega^\xi = 1,\\
&\lambda_\omega^\xi \geq 0, \quad \omega \in \Omega.
\end{aligned}$$

Soit Λ_ξ l'ensemble des vecteurs de poids Λ_ξ satisfaisant le système ci-dessus pour une $\xi \in \Xi$ donné. Dans le paradigme considéré, chaque pôle est associé à une action de recours, et l'action de recours lorsque $\xi \in \Xi$ est considéré, est approchée par une combinaison convexe des opérations de recours associées aux pôles. Plus précisément, soit $\mathbf{v}_\omega$ l'action de recours associée au pôle ω dans le système ci-dessus. Nous avons besoin des variables ajustables $\mathbf{v}(\xi)$ afin de se limiter à

$$\mathbf{v}(\xi) = \sum_{\omega \in \Omega} \lambda_\omega^\xi \mathbf{v}_\omega,$$

Où $\boldsymbol{\lambda}^\xi \in \Lambda_\xi$. Nous pouvons facilement présenter la contrepartie robuste multipolaire définie par

$$\begin{aligned}
\Pi_\Xi(\mathbf{P}, \Omega) = \min_{\mathbf{u},\mathbf{v}} \quad & \mathbf{c}^T\mathbf{u}\\
\text{s.t.} \quad & \mathbf{U}\mathbf{u} + \mathbf{V}\sum_{\omega \in \Omega} \lambda_\omega^\xi \mathbf{v}_\omega \leq \mathbf{b}, \ \xi \in \Xi, \ \boldsymbol{\lambda}^\xi \in \Lambda_\xi.
\end{aligned}$$

La contrepartie robuste multipolaire cherche une paire de solution non-ajustable u et un ensemble d'actions de recours liées aux pôles $\mathbf{v}_\omega, \omega \in \Omega$ tel que la fonction objectif est minimisée lorsque les paramètres incertains varient dans l'ensemble d'incertitude Ξ. En bref, étant donné Ξ, l'approche robuste multipolaire peut être vue comme une fonction d'ensemble d'un Ω pôle-set et une matrice d'ombre P. On note la fonction par

$$\Pi_\Xi : \mathbb{R}^{n_0 \times \dim(\Xi)} \times \mathcal{F}_{\Xi_P} \ni (\mathbf{P}, \Omega) \mapsto \Pi_\Xi (\mathbf{P}, \Omega) \in \mathbb{R}$$

et on désigne par $\Pi_\Xi (\mathbf{P}, \Omega)$ la valeur robuste par rapport à $(\mathbf{P}, \Omega)$. En outre, nous appelons $(\mathbf{u}, \mathbf{v})$ solution multipolaire.

Nous avons ensuite prouvé que la contrepartie robuste multipolaire généralise la contrepartie robuste statique, contrepartie robuste affine ajustable, ainsi que contrepartie robuste entièrement ajustable. Par ailleurs, dans le cadre de l'optimisation robuste multipolaire, on montre que si l'ensemble d'incertitude Ξ est un simplex, la contrepartie robuste affine ajustable est équivalente à la contrepartie robuste entièrement ajustable dans le sens où les valeurs des fonction objectif sont égales à l'optimum.

Nous montrons également que la contrepartie robuste multipolaire est traitable soit par une procédure de génération de coupe ou d'une formulation compacte. Un résultat important est que, étant donné une matrice d'ombre P, la fonction Π_Ξ est monotone croissante par rapport à l'inclusion de l'enveloppe convexe des ensembles de pôles. En outre, nous montrons que l'approche multipolaire peut générer une séquence de bornes supérieure et une séquence de bornes inférieures en même temps et les deux séquences convergent vers la valeur robuste des FARC sous certaines hypothèses modérées.

Comme indiqué précédemment, l'approche multipolaire est basée sur la définition d'un ensemble de pôles. Pour le construire, nous commençons par générer un simplex puis nous calculons la meilleure transformation homothétique de ce simplex lui permettant de contenir un ensemble convexe donné. Un algorithme efficace est proposé pour calculer cet ensemble homothétique. En tant que conséquence directe de notre algorithme, nous offrons une preuve simple des résultats géométriques de [128] liés aux hypercubes. Les ensembles de pôles obtenus après cette transformation homothétique sont alors améliorés en utilisant une procédure de raffinement basée sur la génération de coupes qui est utilisée pour mettre à jour l'ensembles de pôles.

Pour illustrer numériquement l'approche multipolaire, un problème de lobbying est considéré où un lobby vise à minimiser le budget nécessaire pour convaincre un ensemble d'électeurs en tenant compte d'un modèle de dynamique d'opinion raisonnable sous une certaine incertitude. Dans nos expériences numériques, nous utilisons notre contrepartie robuste multipolaire pour modéliser le problème. Nous considérons des hypercubes ou des boules comme ensemble d'incertitude. Généralement, la contrepartie robuste multipolaire ferme de manière significative l'écart entre contrepartie robuste affine et la contrepartie robuste entièrement ajustable, en particulier pour les incertitudes du type hypercube. En outre, comme on pouvait s'y attendre, la valeur robuste diminue à mesure que plus d'informations est utilisée dans la contrepartie robuste multipolaire. Enfin, nous avons illustré le bénéfice de l'adaptabilité.

Dans l'ensemble, Résumons les principaux résultats de cette thèse comme suit.

- Une formulation exacte et plusieurs reformulations pour le problème d'affectation de machines virtuelles dans le cloud. Nous utilisons plusieurs inégalités valides pour renforcer la formulation exacte, accélérant ainsi l'algorithme de résolution de manière significative. En outre, une décomposition lagrangiennes efficace et une formulation induite par symétrie sont proposées.

- Deux approches sont proposées pour la construction d'enveloppes convexes et concaves pour l'optimisation bilinéaire sur un hypercube. Nous établissons plusieurs connexions théoriques entre différentes techniques et nous discutons d'autres extensions possibles.

- Un nouveau paradigme sur les problèmes linéaires généraux avec des paramètres incertains. Nous proposons une hiérarchie convergente de problèmes d'optimisation robuste - approche robuste multipolaire, qui généralise les notions de robustesse statique, robustesse affinement ajustable, et la robustesse entièrement ajustable.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Summary

Many real life problems are characterized by making decisions with current information to achieve certain objectives. Mathematical programming has been developed as a successful tool to model and solve a wide range of such problems. However, many seemingly easy problems remain challenging. And some easy problems such as linear programs can be difficult in the face of uncertain inputs. Motivated by a telecommunication problem where assignment decisions have to be made such that the cloud resources are utilized in a minimum-cost way, we employ several mathematical programming tools to solve the problem efficiently and develop new tools for general theoretical problems. In brief, our work can be summarized as follows.

- An exact formulation and several reformulations on the cloud virtual machine mapping problem. We use several valid inequalities to strengthen the exact formulation, thereby accelerating the solution procedure significantly. In addition, an effective Lagrangian decomposition and a symmetry-induced formulation are proposed.
- A couple of new perspectives on the construction of convex and concave envelopes for bilinear optimization over a hypercube. We establish several theoretical connections between different techniques and provide a novel approach.
- A novel paradigm on general linear problems with uncertain parameters. We propose a hierarchical and convergent framework of adjustable robust optimization – *multipolar robust approach*, which generalizes notions of static robustness, affinely adjustable robustness, fully adjustable robustness and fills the gaps in-between.

Our goal is to provide a mapping solution of cloud virtual machines to cloud operators such that the resource of data center networks are utilized economically and efficiently in a cloud computing environment. This easy-to-state problem however is hard to solve even for small size problem instances as it is a nonconvex quadratically constrained problem. Then, several questions arise. How to improve the scalability of computation? How to model the problem in the face of uncertainty? Can we generalize some techniques or provide new perspectives to some theoretical problems? This thesis is planned to address these questions.

1.2 Thesis motivation

This thesis is motivated by an optimization problem in the context of cloud computing where a number of different sets of virtual machines need to be assigned to servers in a date-center network. In this section, we present the problem and some background information. Then, we review some existing work and point out several challenges.

1.2.1 Mapping of cloud virtual machines

Cloud computing paradigm emerges as a combination of many advanced technologies, e.g., hardware virtualization, Web services, distributed computing and date center automation. It significantly improves the scalability and flexibility of the utilization of network resources while cutting the IT cost as a pay-as-you-go service.

In a cloud computing environment, clients of telecommunication operators demand cloud resources for different applications. Cloud resources primarily include memory and CPU, which are usually packed as *virtual machines* (VMs). To meet practical requirements (e.g., data transmission, live migration), communications between virtual machines are required. Therefore, a request from a single client can be described as a set of virtual machines with associated network throughput.

For each demand from a client, cloud operators organize their activities in a top-down process. It involves pricing, accounting, resource allocation and mapping procedures. In the mapping procedure, cloud operators map the virtual requests to physical network machines by the technology of virtualization, which vertically scales the usage of hardware resources (CPU and memory).

Today, with an increasing number of demands on cloud services, the utilization of server virtualization in *Infrastructure as a Service* (IaaS) and *Platform as a Service* (PaaS) grows rapidly. Therefore, improving the scalability and reducing the power consumption of data center networks becomes a practical issue to cloud operators. Several solutions are available to alleviate this situation. For example, one can change the existing network architecture and routing protocol. Alternatively, one may consider optimizing the placement of VMs on the hosts of virtualised servers under multiple constraints while minimizing the power consumption.

This thesis focuses on the assignment of virtual resource to physical servers. In general, a *Virtual Network Function* (VNF) is responsible for the implementation of functionalities of multiple VMs (a request) on top of the hardware networking infrastructures. Possible mapping solutions are limited to certain hard constraints, for example, resource capacity constraints, traffic routing constraints. The solution to this problem is how to map the cloud resources and which servers and links should be used. Our objective is to minimize the power consumption cost incurred by the mapping decisions, which will also be called mapping cost. And henceforth, we will call this problem mapping problem.

To conclude this section, we illustrate the mapping problem by an example in Figure 1.1. In the example, we have two virtual requests which are also two VNFs. The left-hand-side one has three VMs and the right-hand-side one has two VMs. Also, VMs within the same request communicate with each other. One feasible solution is that for

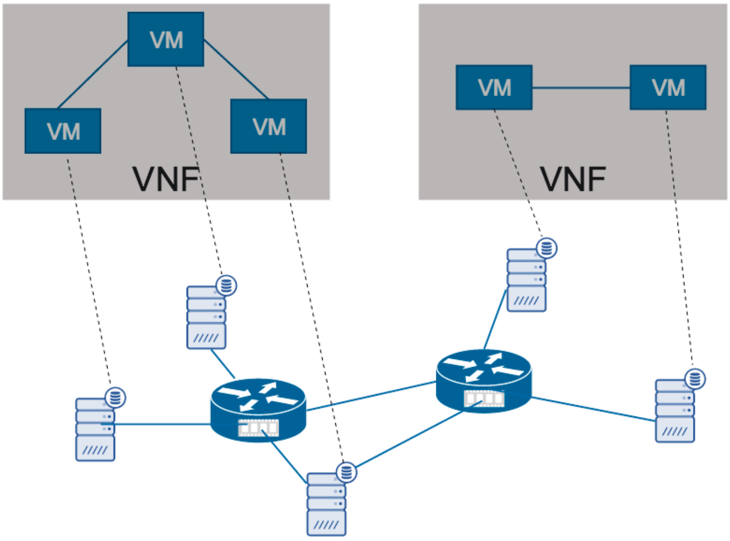

Figure 1.1: An illustration of the mapping procedure

each request, VMs are mapped to different severs and the communication throughput between each pair of VMs is routed between the corresponding servers that VMs are mapped to.

1.2.2 Problem background

In this section, we provide some basic information on virtual requests.

As Cloud Computing is built on the virtualization technologies, VMs are important components. After a customer's request is received, a set of VMs will be allocated on servers to execute a specific program, even on different operating systems. Without special restrictions, a server can usually run multiple VMs simultaneously. During operating, one VM is isolated from the other, but some communication throughput will be generated between VMs within the same request. Note also that in practice, each request involves a small number of VMs.

Apart from the basic requirements, some additional features are required for advanced computing services. In order to improve the utility of data center resource, VMs should be dynamically started or stopped, and sometimes live migration should be conducted, i.e. move the VM from one physical to another. To ensure the service availability, VMs of the same request should be separately allocated.

In general, VMs are usually categorized by the type, e.g., small, medium, large, xlarge [154] (see Table 1.1). In addition, one may estimate the distribution of the throughput based on real statistics. It has been observed that in practice, while 80% around 800 Kbytes/min, 4% of them have a rate 10 times higher (see [122] for example). Furthermore, one may need to be aware that all the VMs communications between two corresponding servers (which host VMs) should route on a single path, as multipath routing may cause discrepancies among arrivals of data at destination.

Table 1.1: Virtual machine types

	CPU (core)	Memory (GB)
Small	1	1
Medium	2	2
Large	4	4
XLarge	8	16

1.2.3 Related work and challenges

Virtualization technology enables the emergence of cloud computing as a flexible and on-demand service. It has been observed that virtualization can also impact the computation and communication performance of cloud services. These impacts are greatly due to the placement of virtual machines in a virtualization based data-center network, which has attracted a considerable amount of studies. Those studies differ from each other in terms of the objective function, resource associated constraints, and solution methods. We review in this section several investigations related to the mapping problem.

Google in 2012 proposed a challenge organized by the French Operational Research and Decision Aid Society (ROADEF) and the European Operational Research society (EURO), where a set of VMs assigned to a set of machines needs to be reassigned to minimize the assignment cost while balancing the usage of physical machines under several resource constraints. As reported in [126], the proposal takes into account capacity constraints of servers regarding CPU, memory, storage as well as location constraints. However, it does not include bandwidth constraints respecting the throughput requirements among VMs. Exact formulations of this problem are mixed integer linear programs. To deal with large scale problems, different heuristics were proposed. Similar work has been conducted in [68], where authors constructed a structural representation of virtual services. The root of the structure holds all the information associated with the structure and each child is associated with VM type information and constraints. However, the problem formulation does not concern the constraint of bandwidth incurred by communication flows. One of the specific challenges on the collaborations among data centers is that a local cloud site has no information on the specifics of remote sites (i.e. the amount of available capacity). To address the scalability of data centers, authors in [122] introduced the traffic-aware virtual machine placement model where only bandwidth constraints are considered and it results in a Quadratic Assignment Problem (QAP). Rather than solving the problem exactly, a two-tier approximate algorithm was proposed. Furthermore, traffic patterns and network architectures are taken into account as some impacts of the scalability.

In the context of *virtual network embedding* (VNE), similar mapping problems are also considered. Houidi et al. [86] proposed a general approach to virtual network provision among multiple substrate networks. The virtual provision process is characterized by four stages: resource matching, splitting, embedding and binding. In the phase of

VNE, a mixed integer linear programming model was developed and an exact algorithm was proposed for the solution. Their global objective is to maximize the acceptance ratio. They also considered the dynamic case, where the arrival of VMs is dynamic and unpredictable over time. Heuristics on sequential request processing and VM embedding with parallel request processing where discussed. Note that due to the scalability of cloud computing, a graph splitting procedure is not necessarily considered in the cloud environment. Authors in [133] presented a unified resource allocation framework for IaaS clouds based on VNE techniques. They applied the methodology proposed in [59] to correlate the node and link mapping phase, leading to a mixed integer program. The formulation allows splittable traffic flows (i.e., multiple routing paths) for any O-D pair. Authors considered two phases solution procedure: node mapping and link mapping. In node mapping phase, they applied random rounding techniques [139] to correlate flow variables and binary variables. A potential substrate node which maximizes the products of flow variables and binary variables is then selected. Decision on the mapping of virtual links are made by solving a Multi-Commodity Network Flow (MCNF) problem.

A dynamic placement model for clustered web applications was proposed in [93]. In their proposal, a placement controller decides to start or stop servers as needed to control the number and placement of application instances, which reduces overheads (e.g., configuration file loading), but it does not allow combinations of applications to be deployed and executed on application servers. They considered multiple objectives, i.e., maximize the acceptance ratio, minimize the placement changes, and balance the loads of servers, while respecting some linear knapsack constraints. To solve the problem, a heuristic method was proposed.

In the face of uncertain demands of VMs, authors in [54] presented a model to place VMs across multiple cloud providers, where they assumed four components in the cloud environment: user, VM repository, cloud provider, and cloud broker. The algorithm was designed for cloud brokers to allocate VMs onto cloud providers and it assumes that there are three phases provisioning resources: reservation, utilization and on-demand plans. Whenever the demand exceeds the amount of reservation resources, additional prices (usually greater than the sum of the corresponding reservation and utilization cost) are incurred. They used stochastic integer programming techniques (a two-stage recursive formulation) to minimize the total cost. The first stage variable is defined as the number of VM reservation, and the second stage variables are defined as the numbers of VMs in utilization phrase and on-demand phase. The probability distributions of uncertain parameters of demands and prices are given. Instead of considering bandwidth constraints explicitly, authors associated with each VM a bandwidth cost.

A recently related thesis [120] studied the virtual network infrastructure provisioning in distributed cloud environment, where an exact model taking into account bandwidth constraints was proposed. To improve the scalability, a heuristic method was proposed based on graph partition and bipartite graph matching techniques.

While a large number of papers in the mapping problem have appeared, there are still some challenges. First, to the best of our knowledge, few mathematical programming techniques have developed to the solution procedure of the bandwidth constrained mapping problem. Second, parameters of demands are usually uncertain, for example, the communication throughput between VMs fluctuates largely.

1.3 Content of the PhD

1.3.1 The scope

In this thesis, we address challenges presented at the end of last section. Point by point, let us identify the focus of our thesis.

- As mentioned earlier, the mapping problem is a 0-1 quadratically constrained problem. Its combinatorial structure is rather complex as it contains the quadratic assignment and the quadratic knapsack problems. The tractability of this problem relies heavily on the development of techniques on Mixed-Integer Quadratically Constrained Programming (MIQCP). However, as mentioned in [52, 50, 51], even though significant progress has been made, the "breakthrough" results are yet to come and many fundamental problems have not been handled. Indeed, standard linear relaxations bring large gaps, while techniques such as semi-definite relaxation is computationally costly in the branch-and-bound procedure. Thus developing effective relaxation techniques and schemes for the deterministic model is the central task of the mapping problem.

- As MIQCP covers not only the mapping problem, but also a wide range of other applications and research problems, for example, quadratic assignment problems [81, 153], pooling and blend [70, 13, 125], max-cut problems [92], we are also interested in revisiting general relaxation techniques and establish some theoretical results for MIQCP problems.

- Motivated by the mapping problem with uncertain demands, we devote efforts to developing a general novel paradigm for linear programs with uncertain parameter in the framework of robust optimization. Linear programs with uncertain parameters can be very difficult when we consider adaptability of recourse actions. Thus, relaxation techniques need to be proposed. Moreover, in practice, the uncertain parameters of an optimization problem can be sometimes difficult to observe. So how to incorporate sufficient uncertainty to a model while allowing decisions to be made upon incomplete information? And is there a systematical way to make a trade-off between the computational complexity and the quality of the solution?

1.3.2 Outline and contributions

The remainder of this thesis is organized in five chapters. Chapter 2 provides an overview on relaxation techniques of MIQCPs. Chapter 3 and Chapter 4 focus on relaxations techniques on MIQCPs from an industrial perspective and a research perspective. Chapter 5 is devoted for developing novel approaches dealing with uncertainty in the framework of robust optimization.

In Chapter 2, we review some relaxation techniques on MIQCP. These techniques include general algorithmic frameworks, polyhedral relaxations, nonlinear relaxations, Lagrangian relaxation, as well as copositive programming and moment-SoS hierarchies. Derivations on some important results are given.

In Chapter 3, we firstly present several ideas to reformulate the mapping problem to get the problem solved efficiently. Further, a Lagrangian-based decomposition is proposed, reducing the problem into a number of subproblems of small sizes, which are then solved by a cutting plane algorithm. A set of experiments is carried out by implementing the branch-and-cut procedure and the accelerated cutting plane algorithm (a.k.a. in-out algorithm) with CPLEX 12.6.3 routines. The numerical results show that several global optima and lower bounds of good quality can be found in a reasonable time by the proposed approaches. Finally, we proposed a reformulation exploiting symmetries of virtual requests, which can handle some larger instances.

Chapter 4 discusses a couple of approaches to approximating the convex and concave envelopes of bilinear functions, particularly over hypercubes due to its generality. The first approach is based on a semi-definite program. The second approach considers some predefined set covers of a hypercube and leads to a linear program. Then we establish a connection between the convex envelope of a bilinear function and the concave envelope of a polyhedral function. Numerical experiments are conducted to compare the two approaches. As a perspective, a novel approach to approximate the envelopes is discussed and illustrated.

In Chapter 5, we study linear programs involving uncertain parameters and propose a new tractable robust counterpart which contains and generalizes several other models including the existing *Affinely Adjustable Robust Counterpart* and the *Fully Adjustable Robust Counterpart*. It consists in selecting a set of *poles* whose convex hull contains some projection of the uncertainty set, and computing a recourse strategy for each data scenario as a convex combination of some optimized recourses (one for each pole). We show that the proposed *multipolar robust counterpart* is tractable and its complexity is controllable. Further, we show that under some mild assumptions, two sequences of upper and lower bounds converge to the optimal value of the fully adjustable robust counterpart. To illustrate the approach, a robust problem related to lobbying under some uncertain opinions of authorities is studied. Several numerical experiments are carried out showing the advantages of the proposed robustness framework and evaluating the benefit of adaptability.

Finally, in Chapter 6, we conclude the thesis by evaluating the results and giving some directions of future research.

Chapter 2

Literature Review

2.1 Summary

In this chapter, we review some relaxation techniques on mixed-integer quadratically constrained programs. In particular, we focus on cases where quadratic constraints are nonconvex and survey valid inequalities of different forms to relax the problem. In addition, we present a geometric view of Lagrangian relaxation, which sheds light on the design of Lagrangian relaxation schemes. Finally, we review two recent paradigms, namely, copositive programming and moment-SoS hierarchies in a concise way.

2.2 Introduction

A Mixed-Integer Quadratically Constrained Program (MIQCP) is an optimization problem of the following form

$$\begin{aligned} \min \quad & \mathbf{c}_0^T \mathbf{x} \\ \text{s.t.} \quad & q_i(\mathbf{x}) \leq \mathbf{0}, \; i = 1, \ldots, m, \\ & \mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n, \end{aligned} \tag{2.1}$$

where $q_i(\mathbf{x}) = \mathbf{x}^T \mathbf{Q}_i \mathbf{x} + \mathbf{c}_i^T \mathbf{x} + d_i \; (i = 1, \ldots, m), \mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} \leq \mathbf{b}, \; x_j \in \mathbb{Z}, j \in J\}$, and $\mathbf{c}_i \; (i = 1, \ldots, m)$ are n-dimensional vectors, $d_i \; (i = 1, \ldots, m)$ are scalars. We do not assume any convexity property of functions q_i but explicitly assume that $\mathcal{X}$ is a ***compact*** set in an n-dimensional real space. We further assume that the constraint set denoted by $\mathbf{K} := \mathcal{X} \cap \{\mathbf{x} : q_i(\mathbf{x}) \leq \mathbf{0} \; (i = 1, \ldots, m)\}$ is ***nonempty*** and ***compact***, so that $\mathbf{c}_0^T \mathbf{x}$ attains its minimum at some $\hat{\mathbf{x}} \in \mathbf{K}$.

Our assumption on set $\mathcal{X}$ can be justified by the following reasons. First, it is known that the decision version of a general MIQCP without restriction on the topological property of $\mathcal{X}$ is undecidable. Jeroslow [88] showed that minimizing a linear form over quadratic constraints in integer variables is not computable by a recursive function (i.e., no computing device can be programmed to compute the optimums of all problems in the problem class.). One way to address this issue is to ensure the compactness of set $\mathcal{X}$. Second, for most of practical optimization applications, the compactness of $\mathcal{X}$ usually holds[1].

MIQCP constitutes a very general family of problems and there are several ways to categorize them. Given a problem instance $((q)_{i=1}^m, \mathcal{X})$, we can categorize it by the convexity, linearity, and integrality of the problem.

- By convexity. If all $\mathbf{Q}_i$ $(i = 1, \ldots, m)$ are positive semidefinite (psd) matrices, we say that (2.1) is a convex MIQCP; otherwise, if there exists one $\mathbf{Q}_i$ not being psd, we say that (2.1) is a nonconvex MIQCP.

- By linearity. If $\mathbf{Q}_i$ $(i = 1, \ldots, m)$ are null matrices (all components are zero), program (2.1) becomes a Mixed-Integer Linear Program (MILP); otherwise it is a nonlinear (quadratic) program.

- By integrality. If $\mathcal{X}$ does not contain integrality constraints, (2.1) becomes a continuous quadratically constrained problem.

Note that formulation (2.1) also includes optimization problems with quadratic objective function as one can move the objective function to the constraint set by introducing an additional variable. In addition, despite the fact that polynomial optimization contains the family of MIQCP problems, we notice that any polynomial optimization problem might be reduced in the form of (2.1) by introducing auxiliary variables.

Since the objective function of (2.1) is linear, the global optima of (2.1) lie among extreme points of the constraint set. Henceforth, we will focus on the constraint set of (2.1). It has been commonly acknowledged that convexity plays a fundamental role in mathematical programming in the tractability of an optimization problem. Rockafellar in [141] pointed out that "In fact the watershed in optimization isn't between linearity and nonlinearity, but convexity and non-convexity.". However, the definition of *convex optimization* varies in the literature. By saying a convex problem in the form of (2.1), Recockafallar [141] refers to cases where $\mathcal{X}$ is convex and q_i $(i = 1, \ldots, m)$ are convex *relative to* set $\mathcal{X}$ (or equivalently, $\mathbf{K}$ is convex) as a convex optimization problem, while Boyd and Vanderghe [45] refers to the situation that $q_i (i = 1, \ldots, m)$ are convex functions and the algebraic representation of $\mathcal{X}$ is convex. Following Lasserre [104], we say that (2.1) is a convex optimization problem if $\mathbf{K}$ is convex and we say that (2.1) is a *convex program* (or has a *convex representation*) if q_i $(i = 1, \ldots, m)$ and the algebraic representation of $\mathcal{X}$ are convex.

We remark that the development of convex programming, including paradigms (e.g., linear programming, conic programming) and algorithms, has brought success in solving convex MIQCPs. But it appears that breakthroughs for nonconvex MIQCPs are yet to come.

Outline. In this chapter, we will review convex relaxation techniques handling a nonconvex MIQCP. In Section 2.3, we present two general algorithmic frameworks dealing with nonconvex MIQCP along with several definitions and basic concepts. Then we discuss both classical and recent techniques on polyhedral relaxation in Section 2.4.

[1] Special cases where unbounded variables are necessary, e.g., [111], will not be considered in model (2.1).

In Section 2.5, we discuss several semidefinite relaxations and their equivalent reformulations, followed by dynamic strategies of adding valid inequalities. We also mention several quadratic convexification techniques and their connection with the standard semidefinite relaxation. Section 2.6 revisits the classic Lagrangian relaxation from a geometric view, which sheds light on the strength of the associated Lagrangian bound. Finally, we presents two novel paradigms briefly.

Notation. Throughout this chapter, for any matrix $\mathbf{C}$, $\mathbf{C}^T$ denotes its transpose and C_{ij} represents its i^{th} row j^{th} column component. For a set S, $\operatorname{conv} S$ stands for its convex hull; $\operatorname{Int} S$ refers to its interior and cl denotes its closure. If S is finite, $|S|$ stands for its cardinality. For a general function $f : S \mapsto \mathbb{R}$, we denote by $\check{f}$ (resp.$\widehat{f}$) the convex (resp. concave) envelope of f over a n dimensional hypercube. The inner product between two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times n}$ is denoted by $\langle \mathbf{A}, \mathbf{B} \rangle$. For a symmetric matrix $\mathbf{X} \in \mathbb{S}^n$ and vector $\mathbf{x} \in \mathbb{R}^n$, $\mathcal{S}(\mathbf{x}, \mathbf{X})$ refers to $\begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \mathbf{X} \end{pmatrix}$ in $\mathbb{S}^{n+1}$. $\mathbb{S}^n_+$ represents the cone of $n \times n$ real positive semidefinite matrices.

2.3 General algorithmic frameworks

It is known that a nonconvex MIQCP is much harder to solve than a convex one. For a convex MIQCP, dropping the integrality constraints usually leads to a tractable convex program. Methods that address convex MIQCPs include, for example, generalized Benders' decomposition [72], outer approximation [66], Non-Linear Programming (NLP) or Linear Programming (LP) based Branch-and-Bound (BB) algorithms [138]. We refer readers to excellent surveys [43, 77, 21] for details. In contrast, for a nonconvex MIQCP, dropping the integrality constraints leads to a nonconvex program, which is essentially a global optimization problem. To address this issue, one usually considers replacing the nonconvex functions q_i $(i = 1, \dots, m)$ with their *convex underestimators.*

Definition 2.1 *For a nonconvex function $f : \mathcal{X} \mapsto \mathbb{R}$, its convex underestimator denoted by $\underline{f}$ over $\mathcal{X}$ is a convex function such that $\underline{f}(\mathbf{x}) \le f(\mathbf{x})$, $\forall \mathbf{x} \in \mathcal{X}$. Its convex envelope is defined as the supremum functional among the set of all convex underestimators of f. Likewise, a concave overestimator of f denoted by $\overline{f}$ over $\mathcal{X}$ is a concave function such that $f(\mathbf{x}) \le \overline{f}(\mathbf{x})$, $\forall \mathbf{x} \in \mathcal{X}$ and its concave envelope is the infimum functional among the set of all concave overestimators.*

Another methodology is to find a convex relaxation of set $\mathbf{K}$. Relations between these two methodologies can usually be established by Lagrangian duality (see, for example, [107, 117]). Details about these techniques will be discussed in subsequent sections and we now briefly review two general algorithmic frameworks for exact solutions.

In general, the replacements above will produce a mixed-integer convex program. Following [110], we call it an *auxiliary problem* of (2.1). For the following discussion, it is useful to present definitions below.

Definition 2.2 *[110] Let P be auxiliary problem of* (2.1) *and $\mathbf{F}(P)$ refers to the feasible region of P. If $\mathbf{K} \subsetneq \mathbf{F}(P)$, we say that P is a relaxation of* (2.1)*. If problem P preserves* *local (resp. global) optimality information of* (2.1)*,i.e., there is a mapping from set* $\mathbf{F}(P)$ *to set* $\mathbf{K}$*, such that its restriction the set of all local (resp. global) optima of* P *is surjective with respect to local (resp.global) optima of* (2.1)*, then we say that* P *is a local (resp. global) reformulation of* (2.1)*. If* P *is a local and global reformulation of* (2.1)*, we say that* P *is an exact reformulation of* (2.1)*.*

Note that a relaxation of P might also be an exact reformulation of the original problem (2.1) even though the feasible region $\mathbf{F}(P)$ is strictly larger than that of the original problem. A family of such relaxations has been explicitly studied and characterized by Ben-Ameur and Neto [8] and they call them *optimality-equivalent* relaxations.

Two general algorithmic frameworks are often employed depending on the type of the auxiliary problem P. If P is a global reformulation of (2.1), then we can employ any method dealing with convex Mixed-Integer Non-linear Programs (MINLPs) to solve P and then recover the optima of the original problem (2.1) by certain mapping function. Methods reformulating a nonconvex MIQCP to a convex MIQCP include, but not limited to, conversions to MILP [19], quadratic convex reformulations [40, 38], copositive reformulation [49]. The efficiency of the solution procedure is heavily influenced by the strength of reformulations.

If P is a convex relaxation of (2.1), one can rely on global optimization methods, e.g., spatial branch-and-bound, branch and reduce. Spatial Branch-and-Bound (sBB) was first proposed by McCormick [119]. Like standard branch-and-bound methods, sBB method recursively partitions the searching space in a convergent way, relying on evaluating a sequence of subproblems. Specifically, sBB first solves the convex relaxation P, which provides a lower bound. Then, it seeks to solve the (nonconvex) nonlinear problem with a local solver. If the resulting solution is feasible to (2.1), then it provides an upper bound. If the gap between upper bound and lower bound is within some predefined tolerance, the algorithm terminates and outputs the optimal solution. If not, it partitions the domain of a variable, leading to two new subproblems. Each subproblem is solved to get upper and lower bound repeatedly. In general, techniques tightening the bound of variables helps to improve the strength of convex relaxations.

Apart from the strength of convex estimators, the convergence rate also depends on the partition strategy and bounding techniques of a sBB algorithm. For example, Linderoth [113] proposed a simplicial sBB algorithm for solving QCQPs over a box, where searching space is recursively partitioned into triangles and rectangles. The author showed theoretically and computationally, that the convergence of the proposed sBB algorithm is generally faster than algorithms that partition a searching space to rectangles.

Another well-known sBB algorithm in global optimization community is the so-called "αBB" method, which was proposed by Floudas and co-workers [9]. For nonconvex MIQCPs, the principle idea of "αBB" method is to construct convex underestimator of nonconvex functions (q_i) by adding certain quadratic terms with multipliers α, such that the resulting Hessian matrix $Q(\alpha)$ is positive semidefinite. Investigations on the convergence of αBB algorithms and McCormick's sBB algorithm with respect to the diameter of searching region have been discussed in [9, 42].

We remark that the both standard BB and sBB frameworks heavily rely on the strength of relaxations. In what follows, we will discuss different convex relaxations to problem (2.1).

It is usually not trivial to relax a nonconvex MIQCP to a convex program in the original space. By the idea of *lifting*, one can reformulate problem (2.1) in a higher dimensional space at the expense of additional variables as follows:

$$\begin{aligned} \min \quad & \mathbf{c}_0^T \mathbf{x} \\ \text{s.t.} \quad & \langle \mathbf{Q}_i, \mathbf{X} \rangle \leq \mathbf{0}, \; i = 1, \ldots, m, \\ & \mathbf{x} \in \mathcal{X}, \\ & \mathbf{X} = \mathbf{x}\mathbf{x}^T, \end{aligned} \tag{2.2}$$

where $\mathbf{X}$ is an $n \times n$ real symmetric matrix. Note that $\mathbf{X} = \mathbf{x}\mathbf{x}^T$ is nonconvex in variables $(\mathbf{x}, \mathbf{X})$. Obtaining a relaxation of (2.2) can usually be done by relaxing the constraint $\mathbf{X} = \mathbf{x}\mathbf{x}^T$ with various valid inequalities. Of-course, one can also project valid inequalities to original space and solve a relaxed problem in the original space (see, e.g., [143, 144]). In what follows, we review several well-known methods to generate valid inequalities of different forms.

2.4 Polyhedral relaxations

The most common methods of converting a nonconvex MIQCP to a convex one might be polyhedral relaxation. Among those, we mention the following.

- McCormick in [119] proposed linear relaxations of quadratic term $X_{ij} = x_i x_j$ over the rectangular domain $x_i \in [l_i, u_i], x_j \in [l_j, u_j]$ by the following inequalities

  $$\max\{u_i x_j + u_j x_i - u_i u_j, l_i x_j + l_j x_i - l_i l_j\} \leq X_{ij} \leq \min\{l_i x_j + u_j x_i - l_i u_j, l_j x_i + u_i x_j - u_i l_j\}$$

  These inequalities are referred to as McCormick inequalities in the literature. Note that it takes $\mathcal{O}(n^2)$ variables and constraints. McCormick also pointed out that the constraints above characterize the exact convex and concave envelope of $X_{ij} = x_i x_j$ over rectangles. One can also find detailed proof in [6].

- Glover in [74] proposed to linearize the product of a binary variable and a bounded variable with $\mathcal{O}(n)$ variables. For example, let us linearize $\sum\limits_{i=1}^{n} \sum\limits_{j=1}^{n} x_i D_{ij} x_j$, where $(x_i, x_j) \in \{0,1\}^2$ and $\mathbf{D}$ is a square matrix. We introduce n new variables, $w_i = x_i \sum\limits_{j} D_{ij} x_j$ and add $4n$ constraints, i.e., for each i, we have:

  $$\begin{gathered} D_i^- x_i \leq w_i \leq D_i^+ x_i, \\ \sum_j D_{ij} x_j - D_i^+ (1 - x_i) \leq w_i \leq \sum_j D_{ij} x_j - D_i^- (1 - x_i), \end{gathered}$$

  where $D_i^- = \max\{\sum\limits_j D_{ij}^- x_j\}, D_i^+ = \max\{\sum\limits_j D_{ij} x_j\}$.

- Sherali and Adams in [145] proposed the so-called reformulation-linearization techniques (RLTs) to reformulate general non-convex quadratic problems. One may treat RLTs as an unified way to generate valid inequalities in variables $(\mathbf{x}, \mathbf{X})$ by some basic algebraic manipulations (i.e., addition and multiplication) on inequalities in original space. For example, let us generate RLT inequalities for problem (2.2) by exploiting set $\mathcal{X}$.

  First, for interval constraints $\mathbf{x} \in [\mathbf{l}, \mathbf{u}]$, we multiple pairs of nonnegative terms $(u_i - x_i), (x_i - l_i), (u_j - x_j)$ and $(x_j - l_j)$ to get inequalities:

  $$\begin{aligned} (u_i - x_i)(u_j - x_j) \geq 0, \quad (u_i - x_i)(x_j - l_j) \geq 0, \\ (x_i - l_i)(u_j - x_j) \geq 0, \quad (x_i - l_i)(x_j - l_j) \geq 0. \end{aligned}$$

  Replacing $x_i x_j$ with variables X_{ij} leads to McCormick constraints mentioned above. Second, for linear equality $\mathbf{a}_i\mathbf{x} = b_i$, we multiply any x_j at both sides to obtain quadratic equalities involving $x_i x_j$ terms; For linear inequality of the form $\mathbf{a}_i\mathbf{x} \leq b_i$, we can multiply both sides $(x_j - l_j)$ and $(u_j - x_j)$ respectively. In addition, one can multiple pairs of constrains within set $\mathcal{X}$ to get constraints involving quadratic terms.

  Later Adams and Sherali [3] showed that a hierarchy of increasingly stronger relaxations can be obtained by generating higher order polynomials constraints. In the light of representations of polynomials in real algebraic geometry, Lasserre [102] showed that RLT hierarchy falls exactly within the type of linear relaxation hierarchies based on Krivine-Stengle's positivity certificate. Liberti and Pantelides [112] showed that one can sometimes eliminate unnecessary bilinear terms in MIQCP by augmenting a subset of RLT constraints.

- For unconstrained 0-1 quadratic programs, there has been a considerable amount of investigations devoted to generating valid inequalities. Most of them are attempts to approximate 0-1 polytopes for classical combinatorial problems. Let us mention cut polytope [61], boolean quadratic polytope [132] (a.k.a. correlation polytope [61]) and quadratic assignment polytope [89, 142]. We refer readers to [61] for details. Recently, some interesting results have been established between Boolean Quadratic Polytope (BQP) and the polytope of Quadratic Problem over a Box (QPB), where

  $$\begin{aligned} \text{QPB} &= \text{conv}\left\{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^{\frac{n(n+1)}{2}} : y_{ij} = x_i x_j, 1 \leq i \leq j \leq n,\ \mathbf{x} \in [0,1]^n\right\}, \\ \text{BQP} &= \text{conv}\left\{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^{\frac{n(n-1)}{2}} : y_{ij} = x_i x_j, 1 \leq i < j \leq n,\ \mathbf{x} \in \{0,1\}^n\right\}. \end{aligned}$$

  Burer and Letchford [50] proved that the projection of QPB (by ignoring elements of y_{ii}, $(i = 1, \ldots, n)$) is exactly BQP, which implies that all valid inequalities in variables y_{ij} $(1 \leq i < j \leq n)$ of BQP is valid for QPB.

There are also many other techniques to generate strong linear valid inequalities in a specific context. For example, authors in [121] proposed a novel linearization technique for hub location problems, where they assumed that the unit transport cost between hubs is proportional to the distance between them. Specifically, they consider linearizing $y_{ij} = \sum_{k,l} d_{kl}x_{ik}x_{jl}$, where d_{kl} represents the distance between locations k and l, x_{ik}, x_{jl} are 0-1 decision variables assigning hub i, j to location k, l respectively. They decomposed the distance d_{kl} as the Euclidean norm of two vectors associated with location k and l, which can then be bounded by linear valid inequalities in a static or dynamic way.

2.4.1 Disjunctive cuts

A family of linear cuts can also be generated dynamically in the framework of *disjunctive programming*. Classical disjunctive programming [15] has been studied intensively in the context of MILPs [16, 17]. It was then generalized by Grossmann and co-workers for general MINLPs [79, 78]. Following investigations in [16, 20], we review ideas from disjunctive programming and present a procedure to generate (linear) disjunctive cuts.

Disjunctions are logical operations (composed by basic logical operators "or") that return true if at least one of their operands is true. Classical disjunctive programming [16] requires a linear relaxation and a disjunction of (2.1). Let us denote by $\mathcal{LP}$ a polyhedral relaxation of $\mathbf{K}$. Regarding problem (2.1), a disjunction can be written as

$$F = \bigcup_{k=1}^{q} F_k, \quad \text{and} \quad F_k = \left\{ \mathbf{x} \in \mathcal{LP} : \mathbf{H}^k\mathbf{x} \le \mathbf{h}^k \right\} \ (k = 1, \dots, q)$$

where each $\mathbf{H}^k$ is a $m_k \times n$ real matrix, $\mathbf{h}^k \in \mathbb{R}^{m_k}$, m_k is the number of linear inequalities. We say that this disjunction is satisfied if there exists k such that $F_k \neq \emptyset$. For any $\mathbf{x} \in \mathbf{K}$, disjunction F is satisfied. A single problem may has many disjunctions. For example, a standard BB procedure partitions the searching space by imposing *integer disjunctions* on integer variable x, $x_i \le \alpha \vee x_i \ge \alpha + 1, \ \alpha \in \mathbb{Z}$ in a finite number of steps [20]. For a nonconvex MIQCP, disjunctions are usually taken among non-convex constraints. We postpone related techniques in Section 2.5.1.

Given a current point $\hat{\mathbf{x}} \in \mathcal{LP}$, the central task in disjunctive programming is to generate a cut $\boldsymbol{\pi}^T\mathbf{x} \le \pi_0$ for $\operatorname{cl}\operatorname{conv} F$ that is violated by $\hat{\mathbf{x}}$. This cut is referred to as *disjunctive cut*. Assume that $\mathcal{LP} := \left\{ \mathbf{x} \in \mathbb{R}^n_+ : \mathbf{B}\mathbf{x} \le \boldsymbol{\beta} \right\}$ with $\mathbf{B} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{\beta} \in \mathbb{R}^n$. Balas [16] showed that the cut identified by $(\boldsymbol{\pi}, \pi_0)$ is valid for $\operatorname{cl}\operatorname{conv}\ F$ if and only if $\boldsymbol{\pi}$ belongs to the following set

$$\begin{aligned}\Pi := \{\mathbf{y} \in \mathbb{R}^n : \exists(\mathbf{u}^k, \mathbf{v}^k) \ge \mathbf{0}, \text{s.t. } \mathbf{y} \le (\mathbf{B})^T\mathbf{u}^k + (\mathbf{H}^k)^T\mathbf{v}^k,& \\ (\mathbf{u}^k)^T\boldsymbol{\beta} + (\mathbf{v}^k)^T\mathbf{h}^k \le \pi_0, \ (k = 1, \dots, q)\}&\end{aligned}$$

Observe that Π is related to the *reverse polar* of F. Several properties of *reverse polar* and its application in deriving deep disjunctive cuts were given in [53]. Given Π, an automatic procedure for generating a cut that is maximally violated relies on solving a so-called *Cut Generating Linear Programming* (CGLP) problem [17]:

$$\max_{(\boldsymbol{\pi}, \pi_0) : \boldsymbol{\pi} \in \Pi} \ \boldsymbol{\pi}^T\hat{\mathbf{x}} - \pi_0 \tag{2.3}$$

If the optimum of (2.3) is nonpositive, $\hat{\mathbf{x}} \in F$, otherwise, we generate a disjunctive cut.

As remarked by Balas [16] and Saxena et al. [143], a *normalization constraint* of the form $\sum_{k=1}^{q}((\mathbf{u}^k)^T\xi + (\mathbf{v}^k)^T\xi^k) = 1$ is usually added to the constraint set Π to ensure the strength and numerical stability of the resulting cuts. For details of implementations, we refer readers to [16, 20, 143, 144].

2.5 Nonlinear relaxations

In addition to polyhedral relaxations, two methodologies are frequently used to relax (2.1), namely, semidefinite relaxations and quadratic convexifications.

2.5.1 Semidefinite relaxations

It is well-known that semidefinite programming (SDP) can be used to derive strong relaxations for a wide range of optimization problems. Most SDP relaxations consider relaxing $\mathbf{X} = \mathbf{x}\mathbf{x}^T$ to $\mathbf{X} - \mathbf{x}\mathbf{x}^T \succeq 0$, which is convex with respect to $(\mathbf{x}, \mathbf{X})$. Equivalently, one can replace the constraint $\mathbf{X} = \mathbf{x}\mathbf{x}^T$ with $\mathcal{S}(\mathbf{x}, \mathbf{X}) \succeq \mathbf{0}$ and $\operatorname{rank}(\mathcal{S}(\mathbf{x}, \mathbf{X})) = 1$ (see, e.g., [151]), then drop the rank-one constraint. This standard relaxation is often referred to as Shor's relaxation. It is easy to verify that this standard SDP relaxation can be derived by Lagrangian duality on problem (2.1)(see, e.g., [107, 109]).

One can strengthen the SDP relaxations with valid inequalities, e.g., triangle inequalities, McCormick inequalities, and RLT inequalities. Anstreicher [10] showed the benefits of such combinations in terms of bound strength. Later, Anstreicher and Burer [12] showed that the combination of SDP relaxation and RLT inequalities describes the convex hull of $\left\{(\mathbf{x}, \mathbf{X}) : \mathbf{X} = \mathbf{x}\mathbf{x}^T, \mathbf{x} \in [\mathbf{l}, \mathbf{u}] \subset \mathbb{R}^n\right\}$ if $n \leq 2$.

Buchheim and Wiegele [48] investigated SDP relaxations for unconstrained mixed-integer quadratic programs whose constraint set is of the form $\mathcal{X}_1 \times \cdots \times \mathcal{X}_n$. In addition to the standard SDP relaxation, they propose to separate points (x_i, X_{ii}) from each $\operatorname{conv}\{(x_i, x_{ii}) : x_i \in \mathcal{X}_i\}$ in a branch-and-bound scheme.

Solving large scale SDP problem is much more expensive than solving LPs or SOCPs with the same number of variables as SDP. On the one hand, special purpose solvers for conic programming cannot handle arbitrary convex constraints; on the other hand, general purpose nonlinear solvers (such as Ipopt [152]) are not designed to handle conic constraints [143]. Alternative to adding SDP constraint explicitly, one can also add convex quadratic constraints or linear constraints in an iterative way. Kim and Kojima [95] proposed to rewrite constraint $\mathbf{X} - \mathbf{x}\mathbf{x}^T \succeq 0$ as convex quadratic constraints:

$$\mathbf{x}^T\mathbf{C}\mathbf{x} \leq \langle \mathbf{C}, \mathbf{X} \rangle, \ \forall \mathbf{C} \in \mathbb{S}^n_+, \tag{2.4}$$

or as linear constraints

$$\langle \mathcal{S}(\mathbf{x}, \mathbf{X}), \mathbf{Z} \rangle \geq 0, \ \forall \mathbf{Z} \in \mathbb{S}^{n+1}_+. \tag{2.5}$$

In addition, they showed that for a given $\mathbf{C} \in \mathbb{S}^n_+$, convex quadratic inequality (2.4) is stronger than linear inequality (2.5) with matrix $\mathbf{Z} = \begin{pmatrix} \beta & \mathbf{b}/2 \\ \mathbf{b}/2 & \mathbf{C} \end{pmatrix}$, where $\beta \in \mathbb{R}, \mathbf{b} \in \mathbb{R}^n$.

Indeed, we have

$$
\begin{aligned}
0 \leq (1, \mathbf{x}^T)\mathbf{Z}\begin{pmatrix}1\\ \mathbf{x}\end{pmatrix} &= \mathbf{x}\mathbf{C}\mathbf{x}^T + \mathbf{b}^T\mathbf{x} + \beta \\
&\leq \langle \mathbf{C}, \mathbf{X}\rangle + \mathbf{b}^T\mathbf{x} + \beta \\
&= \langle \mathcal{S}(\mathbf{x}, \mathbf{X}), \mathbf{Z}\rangle
\end{aligned}
$$

where the first inequality follows from the positive semi-definiteness of $\mathbf{Z} \in \mathbb{S}_+^{n+1}$ and the second inequality comes from (2.4). Authors in [95] then decompose the matrix $\mathbf{Q}_i$ into $\mathbf{Q}_i^+, \mathbf{Q}_i^-$ by the concept of difference-of-convex programming [85], where $\mathbf{Q}_i^+ = \sum\limits_{i=1:\lambda_i>0}^{n} \lambda_i \mathbf{v}_i \mathbf{v}_i^T$, $\mathbf{Q}_i^- = \sum\limits_{i=1:\lambda_i<0}^{n} \lambda_i \mathbf{v}_i \mathbf{v}_i^T$, and $\mathbf{v}_i$ is the eigenvector corresponding to eigenvalue λ_i. Therefore $\mathbf{Q}_i^+, -\mathbf{Q}_i^- \in \mathbb{S}_+^n$ and can be used to replace $\mathbf{C}$ in (2.4).

Of-course, there are other ways to choose a proper psd matrix $\mathbf{C}$. For example, Saxena et al. [143, 144] add valid convex quadratic inequalities of the form (2.4) with $\mathbf{C} = \mathbf{v}\mathbf{v}^T$ in an iterative way, where $\mathbf{v}$ is the eigenvector corresponding to negative eigenvalue of matrix $(\hat{\mathbf{X}} - \hat{\mathbf{x}}\hat{\mathbf{x}}^T)$ at a given point $(\hat{\mathbf{x}}, \hat{\mathbf{X}})$. Moreover, they consider adding constraints enforcing $\mathbf{X} - \mathbf{x}\mathbf{x}^T \preceq 0$ (which is nonconvex) in the spirit of disjunctive programming. Specifically, with eigenvector $\mathbf{v}$, it holds that $(\mathbf{x}^T\mathbf{v})^2 \geq \mathbf{v}^T\mathbf{X}\mathbf{v}$, which reduce to $w^2 \geq z$ by a change of variables. Given point $(\hat{w}, \hat{z})$ and assume that $w \in [l, u]$, a disjunction is taken as $F_1 \vee F_2$, where $F_1 = \{l \leq w \leq \hat{w}, z \leq w(l + \hat{w}) - l\hat{w}\}$, and $F_2 = \{\hat{w} \leq w \leq u, z \leq w(u + \hat{w}) - u\hat{w}\}$. Then, one can generate a disjunctive cut by solving problem (2.3). In addition, authors [143] showed that adding disjunctive cuts sequentially reveals efforts of convexifications of $\mathbf{K}$.

2.5.2 Quadratic convexifications

A number of quadratic relaxation techniques, e.g., α-BB method [4], the smallest eigenvalue method [82], difference-of-convex programming [85], and Quadratic Convex Reformulation (QCR) [40, 38, 39], have been proposed in the literature. Among those methods, let us present QCR method [39] in detail. Assuming that $\mathbf{x} \in \mathcal{X} \subset \mathbb{Z}^n$, the so-called QCR method has two phases, the convexification phase by solving an SDP and the standard BB solution phase relying on the convex quadratic program as the output of convexification phase. The key distinction between QCR and other approaches lies in the unconventional way (to reformulate the original problem) of solving an SDP. Specifically, they rewrite each q_i with a psd matrix $\mathbf{C}_i$

$$
\langle \mathbf{x}\mathbf{x}^T, \mathbf{C}_i\rangle + \mathbf{c}_i^T\mathbf{x} + \langle \mathbf{Q}_i - \mathbf{C}_i, \mathbf{X}\rangle + d_i \leq 0 \tag{2.6}
$$

and $\mathbf{X} = \mathbf{x}\mathbf{x}^T$. Therefore, the resulting formulation is an exact reformulation of (2.2). Authors in [39] then relax $\mathbf{X} = \mathbf{x}\mathbf{x}^T$ by some linear inequalities leading to a convex program. To get the best possible choice of $(\mathbf{C}_i)_{i=1}^m$, they solve an SDP problem. Interestingly, the SDP problem is simply Shor's relaxation strengthened by linear inequalities.

To conclude this section, let us mention that SDP relaxations and quadratic convexifications are in fact two methodologies. On the one hand, SDP relaxations can be looked at as attempts to the convexification of $\mathbf{X} = \mathbf{x}\mathbf{x}^T$ over set $\mathcal{X}$; on the other hand, quadratic convexifications can be regarded as efforts in constructing the convex underestimator for each $q_i(x)$ over $\mathcal{X}$. Anstreicher [11] showed that given the same information (valid inequalities), the latter is usually dominated by the former. We remark that the continuous relaxation of the quadratic convex program obtained by the QCR method gets exactly the same optimum of an continuous SDP program strengthened by linear inequalities.

2.6 Lagrangian relaxation

Lagrangian relaxation is a powerful tool in optimization, even nonconvex. As advocated by Lemaréchal, it is an essential method to generate bounds and bring new perspective on standard relaxations, e.g., SDP relaxation, to combinatorial problems (see, e.g., [109, 108, 107]). Designing a proper Lagrangian relaxation (or decomposition) scheme is not a trivial task as one would like to have an easier problem (compared with the original (2.1)) while ensuring strong bounds. We present a couple of geometric results on the strength of Lagrangian relaxation in the literature. Let us assume that q_i $(i = 1, \ldots, m)$ are complicated constraints in problem (2.1). Relaxing these constraints with a dual vector $\boldsymbol{\lambda} \in \mathbb{R}^n$ leads to the following lagrange function

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{c}_0^T \mathbf{x} + \sum_{i=1}^m \lambda_i q_i(\mathbf{x})$$

and the corresponding dual function is

$$\Psi(\boldsymbol{\lambda}) := \begin{cases} \inf_{x \in \mathcal{X}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) & \boldsymbol{\lambda} \geq \mathbf{0}, \\ -\infty, & \text{otherwise.} \end{cases}$$

Recall the assumption that $\mathcal{X}$ is compact, therefore the minimum is attainable.

Let us introduce variables $\boldsymbol{\gamma} \in \mathbb{R}^m$ and move the nonlinear components q_i $(i = 1, \ldots, m)$ to the constraint set, i.e.,

$$\Psi(\boldsymbol{\lambda}) := \min \left\{ \mathbf{c}_0^T \mathbf{x} + \sum_{i=1}^m \lambda_i \gamma_i : \ (\mathbf{x}, \boldsymbol{\gamma}) \in S \right\} \tag{2.7}$$

where $S = \big\{(\mathbf{x}, \boldsymbol{\gamma}) \in \mathbb{R}^{n+m+1} : \mathbf{x} \in \mathcal{X}, q_i(\mathbf{x}) \leq \gamma_i \ (i = 1, \ldots, m)\big\}$. As the objective function is linear, we can safely replace S with its convex hull $\operatorname{conv} S$. Therefore, we have

$$\Psi(\boldsymbol{\lambda}) = \min \left\{ \mathbf{c}_0^T \mathbf{x} + \sum_{i=1}^m \lambda_i \gamma_i : \ (\mathbf{x}, \boldsymbol{\gamma}) \in \operatorname{conv} S \right\}. \tag{2.8}$$

Now observe that (2.8) is exactly the dual objective function of

$$\begin{aligned} \min \ & \mathbf{c}_0^T \mathbf{x} \\ \text{s.t. } & (\mathbf{x}, \boldsymbol{\gamma}) \in \operatorname{conv} S, \\ & \boldsymbol{\gamma} \leq \mathbf{0}. \end{aligned} \tag{2.9}$$

Clearly, Ψ is bounded. If Slater's condition holds that $\exists(\mathbf{x},\gamma) \in \operatorname{conv} S$ such that $\boldsymbol{\gamma} < \mathbf{0}$, we can safely define

$$\max_{\boldsymbol{\lambda}\geq\mathbf{0}} \Psi(\boldsymbol{\lambda}) \tag{2.10}$$

By strong duality, the optimal value of (2.10) is equal to the optimal value of (2.9). Lemaréchal [108] has shown that the constraint set of (2.9) is equivalent to $S' \cap \operatorname{conv} S$, where $S' = \{(\mathbf{x},\boldsymbol{\gamma}) \in \mathbb{R}^{n+m} : \boldsymbol{\gamma} = \mathbf{0}\}$. This reads exactly $\exists(\mathbf{x},\mathbf{0}) \in \operatorname{conv} S$.

Now notice that the original problem (2.1) is equivalent to

$$\min \left\{\mathbf{c}_0^T\mathbf{x} : (\mathbf{x},\mathbf{0}) \in \operatorname{conv}\{(\mathbf{x},\mathbf{0}) \in S\}\right\} \tag{2.11}$$

Clearly, we can see that the duality gap between (2.10) and (2.1) is introduced due to the general relation that

$$\operatorname{conv}\{(\mathbf{x},\mathbf{0}) \in S\} \subseteq \{(\mathbf{x},\mathbf{0}) \in \operatorname{conv} S\}.$$

In the light of this geometric result, one can have an intuition on the quality of the bounds obtained by a Lagrangian relaxation scheme.

Moreover, we can show that the dual bound provided by any Lagrangian relaxation scheme is no worse than the optimum of the continuous relaxation of the primal problem, provided that q_i $(i = 1,\dots,m)$ are convex. To formalize this result, let us denoted by $\overline{\mathcal{X}}$ the polyhedral set obtained by dropping the integrality constraints of $\mathcal{X}$.

Theorem 1 *If q_i $(i = 1,\dots,m)$ are convex functions, the optimal value of (2.10) is greater than or equal to the optimum of the continuous relaxation of problem (2.1) (by replacing $\mathcal{X}$ with $\overline{\mathcal{X}}$).*

Proof: Let C be the constraint set of continuous relaxation of (2.1), i.e.,

$$C = \left\{(\mathbf{x},\mathbf{0}) : \ \mathbf{x} \in \overline{\mathcal{X}}, q_i(\mathbf{x}) \leq 0 \ (i = 1,\dots,m)\right\}$$

It is sufficient to show that $\{(\mathbf{x},\mathbf{0}) \in \operatorname{conv} S\} \subseteq C$. Let $(\overline{\mathbf{x}},\mathbf{0}) \in \operatorname{conv} S$, then there exists $(n+2)$ points $(\mathbf{x}_k,\boldsymbol{\gamma}_k) \in S$, such that $(\overline{\mathbf{x}},\mathbf{0}) = \sum_{k=1}^{n+2} \lambda_k(\mathbf{x}_k,\boldsymbol{\gamma}_k)$, where $\boldsymbol{\lambda} \in \mathbb{R}_+^{n+2}$ and $\sum_{k=1}^{n+2} \lambda_k = 1$. Obviously, it holds that $\overline{\mathbf{x}} \in \overline{\mathcal{X}}$. For each $i = 1,\dots,n$, we have

$$q_i(\overline{\mathbf{x}}) \leq \sum_{k=1}^{n+2} \lambda_k q(\mathbf{x}_k) \leq \sum_{k=1}^{n+2} \lambda_k \gamma_k = 0,$$

where the first inequality is due to the convexity of function q_i, and the second inequality comes from the fact that $(\mathbf{x}_k,\gamma_k) \in S$. Therefore $(\overline{\mathbf{x}},\mathbf{0}) \in C$, which completes the proof. ■

We now consider a special case where $q_i(\mathbf{x})$ $(i = 1,\dots,m)$ are linear functions. With little trouble, one can derive the following result.

Theorem 2 *[108] If all functions q_i, $1,\dots,m$ are linear functions, the dual optimal value of $\Psi(\boldsymbol{\lambda})$ is equal to the optimal value of the following primal problem*

$$\min\left\{\mathbf{c}_0^T\mathbf{x} : \mathbf{x} \in \operatorname{conv}\mathcal{X}, q_i(\mathbf{x}) \leq 0, \ (i = 1,\dots,m)\right\}.$$

This geometric interpretation of Lagrangian relaxation will be used to design a Lagrangian decomposition scheme in Chapter 3.

2.7 Recent paradigms

Recall that relaxing (2.1) amounts to approximating conv $\left\{(\mathbf{x}, \mathbf{X}) : \mathbf{X} = \mathbf{x}\mathbf{x}^T,\ \mathbf{x} \in \mathcal{X}\right\}$. In convex optimization, there exists an important subclass called conic optimization. It optimizes a linear form over the intersection of an affine subspace and a convex cone. Looking at the aforementioned techniques from the perspective of conic optimization, we may realize that we were actually using polyhedral cone, second-order cone, and psd cone to strengthen our relaxations. A natural question arises: can we find some other convex cones that are "smaller"?

To answer this question, we briefly review two recently developed frameworks, namely, copositive programming and moment-SoS hierarchy that have been used to solve nonconvex MIQCPs.

2.7.1 Copositive programming

Copositive programming deals with two convex cones: the cone of copositive matrices and the cone of completely positive matrices, which are dual to each other. An $n \times n$ real symmetric matrix $\mathbf{X}$ is copositive if its quadratic form $\mathbf{v}^T\mathbf{X}\mathbf{v}$ is nonnegative for all $\mathbf{v} \in \mathbb{R}^n_+$. $\mathbf{X}$ is a completely positive matrix if it can be factorized as $\mathbf{B}\mathbf{B}^T$, where $\mathbf{B} \in \mathbb{R}^{n\times k}_+$. It is easy to verify that both cones are closed and convex.

The cone of completely positive matrices is smaller than the cone of psd matrices, and thus one may expect stronger relaxations introduced by enforcing $\mathbf{X}$ being completely positive. Unfortunately, it has been recently proved that checking if a matrix belongs to the completely positive cone is NP-hard [62]. And it is well-known that the membership problem of copositive matrix cone is co-NP-complete [127]. We refer readers to excellent surveys [84, 65] for details about copositive matrices and copositive programming.

Burer [49] showed a quite general result that every quadratic problem with linear and binary constraints has a copositive representation. More precisely, he showed that if a quadratic binary problem

$$\begin{aligned}
\min\ & \mathbf{x}^T\mathbf{Q}\mathbf{x} + 2\mathbf{c}^T\mathbf{x} \\
\text{s.t.}\ & \mathbf{c}_i^T\mathbf{x} = d_i\ (i = 1, \ldots, m), \\
& \mathbf{x} \geq \mathbf{0}, \\
& x_j \in \{0, 1\}\ (j \in J),
\end{aligned} \tag{2.12}$$

satisfies some key condition, i.e., $\mathbf{c}_i^T\mathbf{x}_i = d_i\ (i = 1, \ldots, m)$ and $\mathbf{x} \geq \mathbf{0}$ implies $x_j \leq 1\ (j \in J)$, then it can be rewritten as the following completely positive problem:

$$\begin{aligned}
\min\ & \langle \mathbf{Q}, \mathbf{X}\rangle + 2\mathbf{c}^T\mathbf{x} \\
\text{s.t.}\ & \mathbf{c}_i^T\mathbf{x} = d_i\ (i = 1, \ldots, m), \\
& \langle \mathbf{c}_i\mathbf{c}_i^T, \mathbf{X}\rangle = d_i^2\ (i = 1, \ldots, m), \\
& x_j = X_{jj}\ (j \in J) \\
& \mathcal{S}(\mathbf{x}, \mathbf{X}) \in \text{CP}\,.
\end{aligned}$$

where CP represents the cone of completely positive matrices and $J \subseteq \{1, \dots, n\}$. As pointed in [49], this key condition can be easily satisfied by augmenting the constraint $x_j + s_j = 1$ with slack variable $s_j \geq 0$ for all $j \in J$. Note that (2.12) encompasses a number of NP-hard problems, e.g., all 0-1 linear integer programs, all unconstrained 0-1 quadratic programs. It is still unknown if a general MIQCP problem of the form (2.1) has a copositive representation or not.

Based on the definition of completely positive matrix, one can approximate the completely positive matrix cone from outside to any precision using a sequence of polyhedral-semidefinite relaxations [135]. For example, when $n \leq 4$, it is known that the so-called *doubly nonnegative* matrix cone (the intersection of SDP cone and componentwise nonnegative symmetric matrix cone) characterizes the cone of completely positive matrices exactly [118]. For more general cases, several approaches have been proposed to approximate the copositive cone and its dual by testing positivity of polynomials. Some of those are based on moment-SoS hierarchies, which will be reviewed subsequently.

2.7.2 Moment-SoS hierarchies

In this section, we briefly review the well-known relaxation techniques, moment-SoS hierarchies due to Lasserre [101, 102], Parrilo [135, 136]. For more details, we refer readers to the survey of Laurent [106].

Let us focus on problems of the form (2.1) and introduce $c(\mathbf{x}) = \mathbf{c}^T\mathbf{x}$. $c \in \mathbb{R}[\mathbf{x}]$ is a polynomial (with degree 1) in variables $\mathbf{x}$ over the real field. We represent a polynomial c by $c(\mathbf{x}) = \sum_{\boldsymbol{\alpha}} c_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}}$, where $\boldsymbol{\alpha} \in \mathbb{N}^n_d := \left\{ \boldsymbol{\alpha} \in \mathbb{N}^n : \sum_{i=1}^{n} \alpha_i \leq d \right\}$ and $\mathbf{x}^{\boldsymbol{\alpha}}$ stands for the monomial $x_1^{\alpha_i} \dots x_n^{\alpha_n}$. We identify c by its coefficients $(c_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha} \in \mathbb{N}^n_d}$. As observed by Lasserre [101], it holds that

$$p^* := \min_{\mathbf{x} \in \mathbf{K}} c(\mathbf{x}) = \min_{\mu \in \mathcal{M}(\mathbf{K})} \int_{\mathbf{K}} c \, d\mu. \tag{2.13}$$

where the second minimum is taken over all the probability measures μ supported by $\mathbf{K}$ (i.e., $\int_{\mathbf{K}} d\mu = 1$). Indeed, it holds that $\int_{\mathbf{K}} c \, d\mu \geq \int_{\mathbf{k}} p^* d\mu = p^*$, $\forall \mu \in \mathcal{M}(\mathbf{K})$. On the other hand, for any $\mathbf{x} \in \mathbf{K}$, let μ be the Dirac measure at any point $\mathbf{x} \in \mathbf{K}$, i.e., $\mu := \delta_{\mathbf{x}}$, so we will have $\int_{\mathbf{K}} c \, d\mu = c(\mathbf{x})$ showing that $p^* \geq \min_{\mu \in \mathcal{M}(\mathbf{K})} c \, d\mu$. Further, observe that (2.13) is an *infinite-dimensional* LP problem. Nevertheless, we have that $\int_{\mathbf{K}} c \, d\mu = \sum_{\boldsymbol{\alpha}} c_{\boldsymbol{\alpha}} \int_{\mathbf{K}} \mathbf{x}^{\boldsymbol{\alpha}} d\mu = \sum_{\boldsymbol{\alpha}} c_{\boldsymbol{\alpha}} y_{\boldsymbol{\alpha}}$, where $y_{\boldsymbol{\alpha}} = \int_{\mathbf{K}} \mathbf{x}^{\boldsymbol{\alpha}} d\mu (i = 1, \dots, m)$ is called *moment of order* $\boldsymbol{\alpha}$ for measure μ supported by $\mathbf{K}$. Let $y_0 = \int_{\mathbf{K}} d\mu = 1$. Problem (2.13) then becomes a finite dimensional LP problem:

$$p^* = \min \left\{ \sum_{\boldsymbol{\alpha}} c_{\boldsymbol{\alpha}} y_{\boldsymbol{\alpha}} : y_0 = 1, \mathbf{y} \in \mathbf{M}_d \right\} \tag{2.14}$$

where $\mathbf{M}_d = \left\{ y_{\boldsymbol{\alpha}} : \exists \mu \in \mathcal{M}(\mathbf{K}) \text{ s.t. } y_{\boldsymbol{\alpha}} = \int_{\mathbf{K}} \mathbf{x}^{\boldsymbol{\alpha}} d\mu, \forall \boldsymbol{\alpha} \in \mathbb{N}^n_d \right\}$. It is easy to verify that $\mathbf{M}_d$ is a convex cone. So problem (2.14) is a convex problem.

Next, consider the dual of (2.13)

$$d^* := \sup_{\gamma \in \mathbb{R}} \left\{ \gamma : c(\mathbf{x}) - \gamma \geq 0, \ \forall \mathbf{x} \in \mathbf{K} \right\}. \tag{2.15}$$

Now, let us introduce the cone of nonegative polynomial with degree d (in our case, $d = 1$) over $\mathbf{K} : \ominus_d = \{f \in \mathbb{R}[\mathbf{x}]_d : f(\mathbf{x}) \geq 0 \ \forall \mathbf{x} \in \mathbf{K}\}$. Therefore (2.15) is equivalent to

$$d^* = \sup_{\gamma \in \mathbb{R}} \{\gamma : c - \gamma \in \ominus_d\}. \tag{2.16}$$

So (2.16) is also a convex problem. In fact, $\mathbf{M}_d$ and $\ominus_d$ form a dual pair of convex cones. Assume that the compact set $\mathcal{X}$ has an algebraic representation $\{\mathbf{x} : g_j(x) \leq 0 \ (j = 1, \ldots, k)\}$, then we can use some powerful results from real algebraic geometry to characterize cone $\ominus_d$ and $\mathbf{M}_d$. Let us introduce the *quadratic module* $\mathcal{Q}(q_1, \ldots, q_m, g_1, \ldots, g_k)$ generated by $q_1, \ldots, q_m, g_1, \ldots, g_k$:

$$\mathcal{Q}(q_1, \ldots, q_m, g_1, \ldots, g_k) := \left\{ f \in \mathbb{R}[\mathbf{x}] : f = -\sum_{i=1}^{m} \sigma_i q_i - \sum_{j=1}^{k} \sigma_{j+m} g_j + \sigma_0 \right\}$$

where $(\sigma)_{i=0}^{m+k}$ are *sums-of-squares* (SoS) polynomials in variables $\mathbf{x}$. A polynomial f is a sums of squares of polynomials if it can be written as $f = \sum_{j=1}^{l} u_j^2$ for some $u_1, \ldots, u_l \in \mathbb{R}[\mathbf{x}]$. A nice result is that any nonnegative quadratic polynomial is a sum of squares, which follows from celebrated results of Hilbert (see, e.g., [106]).

Putinar's Positivstellensatz (See [106, 101]) provides a positivity certificate for a polynomial over set $\mathbf{K}$.

Theorem 3 *[137] Under the assumption that $\exists M > 0$, s.t. $M - \sum_{i=1}^{n} \mathbf{x}_i^2 \in \mathcal{Q}(q_1, \ldots, q_m, g_1, \ldots, g_k)$, for $f \in \mathbb{R}[\mathbf{x}]$, if $f > 0$ on $\mathbf{K}$, then $f \in \mathcal{Q}(q_1, \ldots, q_m, g_1, \ldots, g_k)$.*

If we replace the constraint $(c - \gamma) \in \ominus_d$ with Putinar's positivity certificate $(c - \gamma) \in \mathcal{Q}(q_1, \ldots, q_m, g_1, \ldots, g_k)$, we get a lower bound of the dual problem (2.16). In addition, the membership problem of the quadratic module reduces to solving an SDP, provided that the degree of SoS is bounded.

Indeed, let us denote by $\mathbf{z}_d := (\mathbf{x}^{\boldsymbol{\alpha}} = \mathbf{x}_1^{\alpha_1} \ldots \mathbf{x}_n^{\alpha_n} : \sum_{i=1}^{n} \alpha_i \leq d)$ the vector that contains all monomials of degree at most d. Then for any $u_j \in \mathbb{R}[\mathbf{x}]_d$ (a polynomial with degree at most d), we have that $u_j(\mathbf{x}) = \mathbf{u}_j^T \mathbf{z}_d$ and thus $\sum_{j=1}^{l} u_j^2 = \mathbf{z}_d^2 (\sum_j \mathbf{u}_j^T \mathbf{u}_j) \mathbf{z}_d$. Therefore σ is an SoS polynomial if and only if $\exists \mathbf{L} \in \mathbb{S}_+^n$, $\sigma = \mathbf{z}_d \mathbf{L} \mathbf{z}_d$. The size of the matrix $\mathbf{L}$ is $\binom{n+d}{d}$. The strength of the relaxation is increased as the bounded degree number increases. Similar arguments can also be derived for the primal side (see [101, 135]). Consequently, the sequence of the primal side relaxations is called moment hierarchy and the sequence of the dual side relaxations is called SoS hierarchy. Notice that one can use different positivity certificates to replace Putinar's Positivstellensatz, which may reduce to different forms of convex problems, e.g., LP, SOCPs [103].

Chapter 3

Optimal Mapping of Cloud Virtual Machines

3.1 Summary

One of the challenges of cloud computing is to assign virtual machines to physical machines optimally and efficiently. The aim of cloud operators is to minimize the mapping cost while respecting constraints regarding location, assignment and capacity. After establishing the exact MIQCP model, we propose several ideas to reformulate the problem in order to accelerate solution procedure. Further, a Lagrangian-based decomposition is proposed, reducing the problem to a number of subproblems that are significantly easier to handle and maintaining a theoretical guarantee on the quality of bounds. Also, a reformulation exploiting symmetries of virtual requests is presented to reduce the number of bilinear terms involved in the formulation. Numerical experiments are conducted showing the effectiveness of the proposed approaches.

3.2 Introduction

Since the background and a brief history of the mapping problem have been introduced in Chapter 1, we focus on the solution procedure in this chapter. The organization of this chapter and its contributions are summarized below.

In Section 3.3, we provide an exact formulation which falls within the realm of MIQCPs. To solve the problem, we propose to reformulate the problem via classical linearization techniques, thereby handling the problem in the framework of MIP. Then we employ the Reformulation-Linearization-Technique (RLT) and various valid inequalities to strengthen the model. Some numerical experiments are conducted to show the effectiveness and limitations of these methods.

Section 3.4 considers a Lagrangian decomposition scheme based on each virtual request to achieve lower bounds of good quality for relatively larger problem instances. We show, both theoretically and numerically, these bounds are generally much stronger than those obtained via continuous relaxations.

In Section 3.5, we introduce a novel formulation that exploits symmetries of each virtual request and allows a reduction of bilinear terms. Numerical experiments suggest that the resultant model is more scalable than the aforementioned ones. To leverage the usage of the new formulation, a heuristic is proposed.

Followed by concluding remarks, we discuss a possible formulation that optimizes certain *mapping polices* as a future research direction to capture the dynamic arrival of virtual requests.

3.3 A general model

Remember that a virtual request consists of a set of virtual machines and virtual communications between them. Therefore, we may represent each virtual request as a directed graph. Our goal is to map such graphs to a physical network. Henceforth, we will use the following notation to construct the mathematical model.

- Sets

– R	set of virtual request.
– $H = (S, E)$	graph of a physical network.
– $G^r = (V^r, L^r)$	a graph of virtual network for request $r \in R$.
– S	set of servers in the physical network.
– E	set of directed edges in the physical network.
– V^r	set of VMs of request r.
– L^r	set of directed virtual links of request r.

- Parameters

– c^{ri}	required CPU of VM $i \in V^r$.
– m^{ri}	required memory of VM $i \in V^r$.
– C_k	CPU capacity of server k.
– M_k	memory capacity of server k.
– F_k	fixed cost of server $k \in S$.
– A_k	additional cost of server k imposed from CPU loads.
– f^{rij}	required throughput associated with logical link $(i, j) \in L^r$.
– B_e	bandwidth of edge $e \in E$.
– W_e	fixed cost of edge $e \in E$.
– P_{kp}	shortest $k - p$ path, $(k, p) \in S \times S : k \neq p$.

- Variables

– $x_k^{ri} \in \{0, 1\}$	1 if VM i of request r is mapped to server k.
– $\theta_k \in \{0, 1\}$	1 if server k is used (switched on).
– $\phi_e \in \{0, 1\}$	1 if edge e is used (switched on).

To help readers bear in mind the notation, we remark that for a parameter or a variable, its subscripts (if it has) are associated with physical resources and its superscripts (if it has) are associated with virtual resources.

In computer networks, the traffic between an O-D pair is usually routed on a single path. As will be shown soon, the resulting formulation contains rather complicated (nonconvex quadratic) constraints. Hence, we make the following assumption in order to reduce the complexity of the problem.

Assumption 1 *The traffic between any O-D pair is routed on the shortest path in terms of hops.*

With the above notations, we construct the exact mathematical model of the mapping problem as follows.

$$\min \sum_{k\in\mathcal{S}} F_k\theta_k + \sum_{k\in S} A_k \sum_{r\in R}\sum_{i\in V^r} c^{ri}x_k^{ri} + \sum_{e\in E} W_e\phi_e. \tag{$\mathbb{P}$}$$

$$\text{s.t.} \sum_{k\in S} x_k^{ri} = 1, \qquad r\in R, i\in V^r, \tag{AC}$$

$$\sum_{i\in V^r} x_k^{ri} \le \theta_k, \qquad r\in R, k\in S, \tag{LC}$$

$$\sum_{r\in R}\sum_{i\in V^r} c^{ri}x_k^{ri} \le C_k\theta_k, \qquad k\in S, \tag{KP}$$

$$\sum_{r\in R}\sum_{i\in V^r} m^{ri}x_k^{ri} \le M_k\theta_k, \qquad k\in S, \tag{KP'}$$

$$\sum_{r\in R}\ \sum_{\substack{k,p\in S:\\ k\neq p, e\in P_{kp}}}\ \sum_{(i,j)\in L^r} f^{rij}x_k^{ri}x_p^{rj} \le B_e\phi_e, \ \ e\in E, \tag{QC}$$

$$\theta_k, \phi_e, x_k^{ri} \in \{0,1\}, r\in R, i\in V^r, k\in S, e\in E. \tag{BC}$$

In what follows, we refer to ($\mathbb{P}$) as the above model with the interpretation below.

- The objective is to minimize the total cost, which is additively composed of three terms: the fixed cost incurred by switching on servers, the additional cost coming from the CPU load, and the fixed cost from the usage of links. We model the additional cost induced by CPU load as a linear function to represent the fact that CPU is usually categorized as load dependent resource, while memory is load independent [93].

- Constraints (AC) mean that each virtual machine must be mapped to a single server. Constraints (LC) model the fact that virtual machines are usually mapped separately in a cloud environment due to some practical issues, e.g., security, reliability.

- Constraints (KP, KP') ensure that for each server, the aggregated required CPU, memory resource cannot exceed its limits. Constraints (QC) emphasis the fact that for each edge, the aggregated throughputs on the edge cannot exceed the bandwidth.

Before the solution procedure, we analyze the difficulties and structure of the problem briefly. From the perspective of mathematical programming, the combination of integrality constraints and bilinear constraints (nonconvex) makes the problem rather difficult. In terms of the problem structure, it shares features of several well-known combinatorial problems, e.g., the multiple knapsack problem and quadratic assignment problem. Authors in [7] have proved that the mapping problem of the form ($\mathbb{P}$) is strongly NP-hard even if $|R| = 1$ by showing a polynomial time reduction from the maximum stable set problem.

3.3.1 McCormick inequalities and Glover's linearization

Model ($\mathbb{P}$) is a 0-1 non-convex quadratically constrained problem. A fundamental idea to deal with such problems is *lifting* it to a higher dimensional space [51, 52]. By introducing new variables y_{kp}^{rij} and enforcing $y_{kp}^{rij} = x_k^{ri} x_p^{rj}$ for each $(r, i, j, k, p) : i \neq j, k \neq p$, we lift the problem to a higher dimensional space and thus results in a MIP problem. This comes at a price of introducing non-convex equations. Simple convex relaxations can be achieved by linearization techniques. In our thesis, we use two linearization techniques: McCormick inequalities [119] and Glover's linearization [74].

A direct use of McCormick's inequalities produces an approximation for the non-convexity, i.e., for each $(r, i, j, k, p) : i \neq j, k \neq p$, we add four inequalities,

$$x_k^{ri} + x_p^{rj} - 1 \leq y_{kp}^{rij}, \tag{3.1a}$$

$$y_{kp}^{rij} \leq x_k^{ri}, \tag{3.1b}$$

$$y_{kp}^{rij} \leq x_p^{rj}, \tag{3.1c}$$

$$y_{kp}^{rij} \geq 0. \tag{3.1d}$$

With this relaxation, we can solve problem ($\mathbb{P}$) by a MIP model:

$$\mathbb{P}_{\text{MC}} : \{\mathbb{P}\} \cap \{(3.1a - 3.1d)\} \tag{$\mathbb{P}_{\text{MC}}$}$$

Notice that McCormick relaxation introduces $\sum_{r \in R} |V_r|(|V_r| - 1)|S|(|S| - 1)$ variables. In contrast, Glover's linearization requires $\sum_{r \in R} |V_r||S|(|S| - 1)$ additional variables. For each (r, k, p, i), it enforces $w_{kp}^{ri} = x_k^{ri} \sum_{j:j \neq i} x_p^{rj} f^{rij}$ by constraints

$$0 \leq w_{kp}^{ri} \leq \max_{j:j \neq i}\{f^{rij}\} x_k^{ri}, \tag{3.2a}$$

$$\sum_{j:j \neq i} x_p^{rj} f^{rij} + (x_k^{ri} - 1) \max_{j:j \neq i}\{f^{rij}\} \leq w_{kp}^{ri} \leq \sum_{j:j \neq i} f^{rij} x_p^{rj}. \tag{3.2b}$$

Consequently, constraints (QC) become,

$$\sum_{(k,p):k \neq p, e \in P_{kp}} \sum_{r \in R, i \in V_r} w_{kp}^{ri} \leq D_e \phi_e, \quad e \in E. \tag{3.3}$$

The resultant model is

$$\min \quad \{f(\theta, \phi, x) : (\text{AC}), (\text{LC}), (\text{KP}), (\text{KP'}), (\text{BC}), (3.2\text{a}), (3.2\text{b}), (3.3)\}. \tag{$\mathbb{P}_{\text{GL}}$}$$

Our numerical experiments indicate that model ($\mathbb{P}_{\text{GL}}$) outperforms McCormick's relaxation for some small instances. However for most instances, even though model ($\mathbb{P}_{\text{GL}}$) introduces much less continuous variables than McCormick's relaxation, its overall computational performance are generally no better than those of model ($\mathbb{P}_{\text{MC}}$). Thus we believe that the main difficulty is not the number of variables and constraints. Besides, we can apply RLT inequalities to strengthen ($\mathbb{P}_{\text{MC}}$) and remove a number of redundant constraints (see Lemma (3.1)), whereas it appears complicated to apply RLTs to model ($\mathbb{P}_{\text{GL}}$). Hence, we will focus on techniques strengthening the McCormick linearization. One may refer to Appendix of this chapter for some numerical results on the comparison between two models.

3.3.2 Polyhedral relaxations

In this section, we propose several valid linear inequalities to strengthen ($\mathbb{P}_{\text{MC}}$).

One of the most straightforward and powerful methods is to add valid linear inequalities. An important subset of those can be derived by the RLTs which is known to strengthening non-convex discrete and continuous formulations [145]. We apply the RLT to constraints (AC), which produces $\sum\limits_{r\in R}|V^r|(|V^r|-1)|S|$ additional constraints,

$$\sum_{k\in S:k\neq p} y_{pk}^{rji} = x_p^{rj}, \quad r\in R, i,j\in V^r : i\neq j, p\in S. \tag{AC$_{\text{RLT}}$}$$

Let us denote the formulation in the sequel as

$$\mathbb{P}_{\text{RLT1}} : \{(\mathbb{P})\} \cap \{(\text{AC}_{\text{RLT}}), (3.1\text{d})\}\,. \tag{$\mathbb{P}_{\text{RLT1}}$}$$

Note that constraints (3.1a-3.1c) are implied by (AC_{RLT}) and model ($\mathbb{P}_{\text{RLT1}}$) is indeed stronger than formulation ($\mathbb{P}_{\text{MC}}$). Let $\mathbf{F}(\mathbb{P}_{\text{RLT1}})$ be the feasible region of ($\mathbb{P}_{\text{RLT1}}$) and $\mathbf{F}(\mathbb{P}_{\text{MC}})$ be the feasible region of ($\mathbb{P}_{\text{MC}}$).

Lemma 3.1 $\operatorname{conv}\mathbf{F}(\mathbb{P}_{\text{RLT1}}) \subseteq \operatorname{conv}\mathbf{F}(\mathbb{P}_{\text{MC}})$.

Proof: It is sufficient to show that McCormick inequalities (3.1a), (3.1b), (3.1c) are implied by constraints of ($\mathbb{P}_{\text{RLT1}}$). It is obvious that (3.1b), (3.1c) are implied by (AC_{RLT}). For each $(r,i,j,k,p) : k<p, i\neq j$, we have

$$\begin{aligned}
x_k^{ri} + x_p^{rj} - 1 &= y_{kp}^{rij} + \sum_{p'>k:p'\neq p} y_{kp'}^{rij} + \sum_{p'<k} y_{p'k}^{rji} + x_p^{rj} - 1\\
&= y_{kp}^{rij} + \sum_{p'>k:p'\neq p} y_{kp'}^{rij} + \sum_{p'<k} y_{p'k}^{rji} - \sum_{s:s\neq p} x_s^{rj}\\
&= y_{kp}^{rij} - x_k^{rj} + \sum_{p'>k:p'\neq p} (y_{kp'}^{rij} - x_{p'}^{rj}) + \sum_{p'<k} (y_{p'k}^{rji} - x_{p'}^{rj})\\
&\leq y_{kp}^{rij} - x_k^{rj}\\
&\leq y_{kp}^{rij}\,.
\end{aligned}$$

The first equality and the first inequality follows from (AC_{RLT}). The second equality comes from (AC). ■

Of-course we can also apply the RLTs to (LC), which adds $2\sum\limits_{r\in R}|V^r||S|(|S|-1)$ constraints.

$$\sum_{i\in V^r:i\neq j} y_{kp}^{rij} - x_p^{rj}\theta_k \leq 0, \quad r\in R, j\in V^r, (k,p)\in S^2 : k\neq p, \tag{LC$_{\text{RLT}}$}$$

$$\sum_{i\in V^r} x_k^{ri} - \sum_{i\in V^r:i\neq j} y_{kp}^{rij} - \theta_k + x_p^{rj}\theta_k \leq 0, \quad r\in R, j\in V^r, (k,p)\in S^2 : k\neq p. \tag{LC$_{\text{RLT}'}$}$$

Notice that we can simply replace the left-hand-side constraints with their convex envelopes which are readily identified by McCormick inequalities (see Section 2.4 for details). Specifically, we replace $x_p^{rj}\theta_k$ in (LC$_{\text{RLT}}$) with $\min\left\{\theta_k, x_p^{rj}\right\}$. Likewise, we replace $x_p^{rij}\theta_k$ in (LC$_{\text{RLT}'}$) with $\max\left\{0, \theta_k + x_p^{rj} - 1\right\}$.

$$\mathbb{P}_{\text{RLT2}} : \{(\mathbb{P})\} \cap \{(\text{AC}_{\text{RLT}}), (\text{LC}_{\text{RLT}}), (\text{LC}_{\text{RLT}'}), (\text{LC}), (3.1\text{d})\}. \tag{$\mathbb{P}_{\text{RLT2}}$}$$

Clearly, the continuous relaxation of ($\mathbb{P}_{\text{RLT1}}$) is stronger than that of ($\mathbb{P}_{\text{MC}}$), but is weaker than that of ($\mathbb{P}_{\text{RLT2}}$). In practice, it entails a trade-off between the strength of relaxations and their sizes to determine a modest model that is robust to the inputs.

Apart from linear valid inequalities coming from the RLT, some strong valid inequalities can be derived by exploiting the problem structure. Specifically, for each $r\in R, (i,j)\in L^r$, if there is a required throughput between i and j, i.e., $f^{rij} > 0$, then for each link e, we have

$$\sum_{(k,p)\in S^2:k\neq p, e\in P_{kp}} y_{kp}^{rij} \leq \phi_e. \tag{Cut}$$

The number of such additional constraints is just $\mathcal{O}(|R||E|)$, but surprisingly, these simple constraints can often accelerate the solution process of a standard MIP solver. Similarly, for each $r\in R, (k,p)\in S^2 : e\in P_{kp}$, if there exists some throughput mapped to P_{kp}, then for each link e, we have

$$\sum_{(i,j)\in L^r} y_{kp}^{rij} \leq \phi_e. \tag{Cut'}$$

Notice that the number of such constraints is $\mathcal{O}(|R||S|^2|E|)$, a quartic polynomial in the number of physical servers, so it might be computationally beneficial to add them dynamically.

To conclude this section, we present two MIP models for the solution procedure of problem ($\mathbb{P}$).

$$\mathbb{P}_{\text{V1}} : \{(\mathbb{P}_{\text{RLT1}})\} \cap \{(\text{Cut}), (\text{Cut'})\}, \tag{$\mathbb{P}_{\text{V1}}$}$$

$$\mathbb{P}_{\text{V2}} : \{(\mathbb{P}_{\text{RLT2}})\} \cap \{(\text{Cut}), (\text{Cut'})\}. \tag{$\mathbb{P}_{\text{V2}}$}$$

3.3.3 Numerical experiments

In this section, we evaluate the effectiveness and limitations of the formulations by conducting some numerical experiments on formulations ($\mathbb{P}_{\text{MC}}$), ($\mathbb{P}_{\text{RLT1}}$), ($\mathbb{P}_{\text{RLT2}}$), ($\mathbb{P}_{\text{V1}}$) and ($\mathbb{P}_{\text{V2}}$). Regarding the problem instances, we assume that all the VMs in a request communicate with each other and fix the number of VMs for each request to be 5. The small physical network instance with 8 nodes is generated randomly. Other instances are taken from SND library [129]. All the models are coded in C++ and solved via CPLEX 12.6.3 on a Mac with Inter Core i5 clocked at 2.7 GHz and with 8 GB of RAM.

As CPLEX 12.6.3 can handle MIQCPs where quadratic terms are multiplications of binary variables, we also use it to solve ($\mathbb{P}$) directly as a benchmark to ($\mathbb{P}_{\mathrm{MC}}$). Results are summarized in Table 3.1. For each problem instance, the size of the physical network (number of servers, number of edges), the number of requests, the number of binary variables, as well as CPU time and the number of branching nodes are reported. As might be expected, the computational time increases exponentially for both formulations as the size of the problem instances increase. Clearly, the simple McCormick relaxation based model outperforms the routines of CPLEX solver for this mapping problem.

Table 3.1: Numerical results for $\mathbb{P}$ and $\mathbb{P}_{\mathrm{MC}}$

Branch-and-Cut statistics						
$(\lvert S\rvert, \lvert E\rvert)$	#Req.	#Bin.	$\mathbb{P}$		$\mathbb{P}_{\mathrm{MC}}$	
			#CPU(s)	#Nodes	#CPU(s)	#Nodes
(8, 20)	1	68	3.33	88	0.03	0
(8, 20)	2	108	10.00	1704	0.45	229
(12, 30)	1	102	371	53305	1.18	16
(12, 30)	2	162	-	-	134	8897
(12, 30)	4	282	-	-	4180	10766

–Time limit: 36000 Seconds (10hs)

Let us now focus on the performance of models ($\mathbb{P}_{\mathrm{RLT1}}$), ($\mathbb{P}_{\mathrm{RLT2}}$), ($\mathbb{P}_{\mathrm{V1}}$) and ($\mathbb{P}_{\mathrm{V2}}$). To show the effectiveness of RLT-based and additional valid inequalities, we first solve the continuous relaxations of different formulations. Then we solve each problem using standard CPLEX 12.6.3 Branch-and-Cut with default settings. Results are summarized in Table 3.2.

Continuous relaxation figures in the table show that for each problem instance, the strength of formulations increases as more valid inequalities are added, which is within our expectation. However notice that the differences w.r.t. the optimality gap between ($\mathbb{P}_{\mathrm{RLT1}}$) and that of ($\mathbb{P}_{\mathrm{RLT2}}$) is negligible while the computational cost of solving ($\mathbb{P}_{\mathrm{RLT2}}$) can be ten times higher than that of solving ($\mathbb{P}_{\mathrm{RLT1}}$). This is probably because the former has $2\sum_{r\in R}|V^r||S|(|S|-1)$ more constraints than the latter. Another observation is that adding (Cut),(Cut') to models ($\mathbb{P}_{\mathrm{RLT1}}$) and ($\mathbb{P}_{\mathrm{RLT2}}$) reduces around 40% gap, which confirms the necessity of exploring additional valid inequalities in addition to the RLT based ones.

To balance the strength of ($\mathbb{P}_{\mathrm{V1}}$) and ($\mathbb{P}_{\mathrm{V2}}$) and their problem sizes, we put constraints

(Cut'), ($\mathrm{LC}_{\mathrm{RLT}}$), ($\mathrm{LC}_{\mathrm{RLT}'}$) to the `user-cut pool` of CPLEX. The B&B results imply that compromises have to be made between the strength of some valid inequalities and their computational costs. For example, both ($\mathbb{P}_{\mathrm{RLT2}}$) and ($\mathbb{P}_{\mathrm{V2}}$) are stronger (cf. Columns '#Nodes') than ($\mathbb{P}_{\mathrm{MC}}$), but they are also computationally more costly. On the other hand, model ($\mathbb{P}_{\mathrm{RLT1}}$) and ($\mathbb{P}_{\mathrm{V1}}$), that are slightly weaker than model ($\mathbb{P}_{\mathrm{RLT2}}$) and ($\mathbb{P}_{\mathrm{V2}}$), are computationally five times as efficient as model ($\mathbb{P}_{\mathrm{MC}}$). For the largest instance, ($\mathbb{P}_{\mathrm{RLT1}}$) reduces the computation time from $13hs$ to $3.74hs$, and ($\mathbb{P}_{\mathrm{V1}}$) performs best if we add constraints (Cut') in an iterative way. In addition, the results also suggest two factors that influence the computational performance: the number of virtual requests and the size of the physical network. For a fixed number of requests, the computation time increases gradually as the size of the physical network increases; on the other hand, given a fixed physical network, the computation time rises significantly as the number of requests grows. Generally, the size of virtual requests might affect computational performance heavier than that of a physical network. Overall, the computational experiments indicate that model ($\mathbb{P}_{\mathrm{V1}}$) performs best among all models.

Table 3.2: Numerical results for different formulations

Continuous relaxation statistics

$(\|S\|, \|E\|)$	#Req.	$(\mathbb{P}_{\text{MC}})$		$(\mathbb{P}_{\text{RLT1}})$		$(\mathbb{P}_{\text{RLT2}})$		$(\mathbb{P}_{\text{V1}})$		$(\mathbb{P}_{\text{V2}})$	
		#Cpu(Sec.)	Gap(%)	#Cpu(Sec.)	Gap(%)	#Cpu(Sec.)	Gap(%)	#Cpu(Sec.)	Gap(%)	#Cpu(Sec.)	Gap(%)
(8,20)	1	0.00	63.9	0.05	54.17	0.01	53.77	0.12	15.01	0.37	12.38
(8,20)	2	0.02	60.8	0.01	51.69	3.06	42.38	0.18	18.89	10.54	18.89
(12,30)	1	0.02	79.6	0.01	78.38	1.03	77.55	0.43	8.00	0.73	6.52
(12,30)	2	0.07	74.7	0.02	72.66	12.21	70.78	1.49	9.70	51.05	8.05
(12,30)	4	0.27	65.88	0.22	64.58	27.31	63.04	10.82	8.52	100.59	7.99
(12,30)	5	0.67	73.3	0.72	69.57	53.92	68.08	11.57	20.32	154.42	20.15

Branch-and-Cut statistics

$(\|S\|, \|E\|)$	#Req.	$(\mathbb{P}_{\text{MC}})$		$(\mathbb{P}_{\text{RLT1}})$		$(\mathbb{P}_{\text{RLT2}})$		$(\mathbb{P}_{\text{V1}})$		$(\mathbb{P}_{\text{V2}})$	
		#Cpu(Sec.)	#Nodes	#Cpu(Sec.)	#Nodes	#Cpu(Sec.)	#Nodes	#Cpu(Sec.)	#Nodes	#Cpu(Sec.)	#Nodes
(8,20)	1	0.29	0	0.77	0	5.6	5	0.53	0	2.81	10
(8,20)	2	0.45	229	4.37	17	42.1	72	0.54	0	1.69	0
(12,30)	1	1.18	16	4.17	22	83.1	68	0.59	0	4.53	3
(12,30)	2	134	8897	348	3013	128	965	15.40	12	61.11	28
(12,30)	4	4180	10766	2062.8	4039	2108	3165	125.78	34	919.19	56
(12,30)	5	47837.75(13h)	513682	13490.20(3.74h)	12799	-	-	3050(0.84h)	139	5770.97(1.65h)	365

– Time limit: 54000 Seconds (15hs).

3.4 A Lagrangian decomposition

The previous numerical results show that the gap incurred by some convex relaxation of the mapping programs is typically large. While several proposed linear valid inequalities are quite effective in strengthening the linear relaxation, there are cases where the gap can be 20%. The goal of this section is to use Lagrangian relaxation to decompose the problem thereby solving the relaxed problem efficiently along with lower bounds of good quality. Of-course, there are plenty of ways to decompose the problem and we have tried many. In this section, we present a request based decomposition scheme, which has a nice interpretation in the context of cloud computing. It also guarantees that the resultant lower bound dominates the optimum of the continuous relaxation of formulation ($\mathbb{P}_{\text{V1}}$) which is numerically the strongest one among the aforementioned models. Numerical experiments suggest that the proposed scheme can computationally be attractive for large scale problems. Let us recall ($\mathbb{P}_{\text{V1}}$) below.

$$\min \sum_{k\in\mathcal{S}} F_k\theta_k + \sum_{k\in S} A_k \sum_{r\in R}\sum_{i\in V^r} c^{ri}x_k^{ri} + \sum_{e\in E} W_e\phi_e \tag{$\mathbb{P}_{\text{V1}}$}$$

$$\text{s.t.} \quad \sum_{k\in S} x_k^{ri} = 1, \qquad r\in R, i\in V^r, \tag{AC}$$

$$\sum_{p\in S:p\neq k} y_{kp}^{rij} = x_k^{ri}, \qquad r\in R, i,j\in L^r, k\in S, \tag{AC_{RLT}}$$

$$\sum_{i\in V^r} x_k^{ri} \leq \theta_k, \qquad r\in R, k\in S, \tag{LC}$$

$$\sum_{r\in R}\sum_{i\in V^r} c^{ri}x_k^{ri} \leq M_k\theta_k, \qquad k\in S, \tag{KP}$$

$$\sum_{r\in R}\sum_{i\in V^r} m^{ri}x_k^{ri} \leq C_k\theta_k, \qquad k\in S, \tag{KP'}$$

$$\sum_{r\in R}\sum_{\substack{i,j\in V^r:\\ i\neq j}}\sum_{\substack{k,p\in S:\\ k\neq p, e\in P_{kp}}} f^{rij}y_{kp}^{rij} \leq B_e\phi_e, \qquad e\in E, \tag{QC}$$

$$\sum_{(k,p)\in S^2:k\neq p,e\in P_{kp}} y_{kp}^{rij} \leq \phi_e, \qquad e\in E, r\in R, i,j\in V^r : f^{rij}>0, \tag{Cut}$$

$$\sum_{(i,j)\in L^r} y_{kp}^{rij} \leq \phi_e, \qquad e\in P_{kp}, k\neq p\in S, r\in R, \tag{Cut'}$$

$$y_{kp}^{rij} \geq 0, \qquad r\in R, i,j\in V^r : i\neq j, k\neq p\in S, \tag{3.1d}$$

$$\theta_k, \phi_e, x_k^{ri} \in \{0,1\}, \qquad r\in R, i\in V^r, k\in S, e\in E.$$

3.4.1 A request based decomposition

We first present a decomposition leading to a sequence of subproblems associated with each request, each sever, and each link. To this end, we disaggregate constraints (KP), (KP'), (QC) by reformulating them with a handful of auxiliary variables with corresponding interpretations.

- w_k^r reserved CPU for request r on server k.
- z_k^r reserved Memory for request r on server k.
- κ_e^r reserved bandwidth for request r on link e.

The equivalent counterpart of (KP), (KP'), (QC) is then

$$\sum_{i \in V_r} c^{ri} x_k^{ri} \leq w_k^r, \quad r \in R, k \in S, \tag{3.5}$$

$$\sum_{i \in V_r} m^{ri} x_k^{ri} \leq z_k^r, \quad r \in R, k \in S, \tag{3.6}$$

$$\sum_{\substack{i,j \in V_r: \\ i \neq j}} \sum_{\substack{k,p \in S: \\ k \neq p, e \in P_{kp}}} f^{rij} y_{kp}^{rij} \leq \kappa_e^r, \quad r \in R, e \in E, \tag{3.7}$$

$$\sum_{r \in R} w_k^r \leq C_k \theta_k, \quad k \in S, \quad \boldsymbol{\lambda} \in \mathbb{R}_+^{|S|} \tag{3.8}$$

$$\sum_{r \in R} z_k^r \leq M_k \theta_k, \quad k \in S, \quad \boldsymbol{\mu} \in \mathbb{R}_+^{|S|} \tag{3.9}$$

$$\sum_{r \in R} \kappa_e^r \leq B_e \phi_e, \quad e \in E. \quad \boldsymbol{\sigma} \in \mathbb{R}_+^{|E|} \tag{3.10}$$

To make the problem separable by request while ensuring the strong lower bounds, we proceed the following. First, we copy variables $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ by introducing following constraints:

$$\theta_k^r = \theta_k, \quad r \in R, k \in S, \quad \boldsymbol{\eta} \in \mathbb{R}^{|R| \times |S|} \tag{3.11}$$

$$\phi_e^r = \phi_e, \quad r \in R, e \in E. \quad \boldsymbol{\zeta} \in \mathbb{R}^{|R| \times |E|} \tag{3.12}$$

Then, we replace θ_k with θ_k^r in constraints (LC). Likewise, we replace ϕ_e with ϕ_e^r in constraints (QC). Finally, we denote the resulting reformulation of $(\mathbb{P}_{\text{V1}})$ as

$$\mathbb{P}'_{\text{v1}} : \{(\mathbb{P}_{\text{V1}})\} \cap \{(3.8), (3.9), (3.10), (3.11), (3.12)\}. \tag{$\mathbb{P}'_{\text{v1}}$}$$

Lemma 3.2 *The projection of the continuous relaxation of* $\mathbf{F}(\mathbb{P}'_{\text{v1}})$ *in variables* $(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\phi})$ *is exactly the continuous relaxation of set* $\mathbf{F}(\mathbb{P}_{\text{V1}})$.

Proof: It holds that the $\text{Proj}_{(\mathbf{x},\mathbf{y},\boldsymbol{\theta},\boldsymbol{\phi})} \mathbf{F}(\mathbb{P}'_{\text{v1}}) \subseteq \mathbf{F}(\mathbb{P}_{\text{V1}})$. The reverse also holds by letting $w_k^r = \sum_{i \in V_r} c^{ri} x_k^{ri} \leq w_k^r, z_k^r = \sum_{i \in V_r} m^{ri} x_k^{ri}, \kappa_e^r = \sum_{\substack{i,j \in V_r: \\ i \neq j}} \sum_{\substack{k,p \in S: \\ k \neq p, e \in P_{kp}}} f^{rij} y_{kp}^{rij}$ for each $r \in R, k \in S, e \in E$. ■

Relaxing the five constraints with associated Lagrangian multipliers $\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\eta}, \boldsymbol{\zeta}$ leads to the Lagrangian over variables $(\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\eta}, \boldsymbol{\zeta}; \mathbf{w}, \mathbf{z}, \boldsymbol{\kappa}, \boldsymbol{\theta}, \mathbf{x}, \boldsymbol{\phi})$. For ease of notation, let

$\mathbf{v}^* = (\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\eta}, \boldsymbol{\zeta})$, $\mathbf{v} = (\mathbf{w}, \mathbf{z}, \boldsymbol{\kappa}, \boldsymbol{\theta}, \mathbf{x}, \boldsymbol{\phi})$. The Lagrangian is then

$$\begin{aligned}
\mathcal{L}\left(\mathbf{v}^*, \mathbf{v}\right) = & \sum_{r\in R}\left(\sum_{k\in S}\left(A_k \sum_{i\in V_r} c^{ri} x_k^{ri} + w_k^r \lambda_k + \mu_k z_k^r + \theta_k^r \eta_k^r\right) + \sum_{e\in E}\left(\sigma_e \kappa_e^r + \zeta_e^r \phi_e^r\right)\right) \\
& + \sum_{k}\left(F_k - C_k\lambda_k - M_k\mu_k - \sum_{r\in R}\eta_k^r\right)\theta_k \\
& + \sum_{e}\left(W_e - B_e\sigma_e - \sum_{r\in R}\zeta_e^r\right)\phi_e.
\end{aligned}$$

The Lagrangian problem reduces to $|R| + |S| + |E|$ subproblems, i.e., for each $r \in R$, we obtain $L^r(\mathbf{v}^*)$ by solving

$$\min \quad \sum_{k\in S}\left(A_k \sum_{i\in V_r} x_k^{ri} a_{ri} + \lambda_k w_k^r + \mu_k z_k^r + \theta_k^r \eta_k^r\right) + \sum_{e\in E}\left(\sigma_e \kappa_e^r + \zeta_e^r \phi_e^r\right) \qquad (\mathrm{Sub}_r)$$

$$\begin{array}{lll}
\text{s.t.} & \displaystyle\sum_{k\in S} x_k^{ri} = 1, & i \in V^r, \\
& \displaystyle\sum_{p\in S: p\neq k} y_{kp}^{rij} = x_k^{ri}, & i, j \in L^r, k \in S, \\
& \displaystyle\sum_{i\in V^r} x_k^{ri} \leq \theta_k^r, & k \in S, \\
& \displaystyle\sum_{i\in V_r} c^{ri} x_k^{ri} \leq w_k^r, & k \in S, \\
& \displaystyle\sum_{i\in V_r} m^{ri} x_k^{ri} \leq z_k^r, & k \in S, \\
& \displaystyle\sum_{\substack{i,j\in V_r:\\ i\neq j}} \sum_{\substack{k,p\in S:\\ k\neq p, e\in P_{kp}}} f^{rij} y_{kp}^{rij} \leq \kappa_e^r, & e \in E, \\
& \displaystyle\sum_{(k,p)\in S^2: k\neq p, e\in P_{kp}} y_{kp}^{rij} \leq \phi_e^r, & e \in E, i, j \in V^r : f^{rij} > 0, \\
& \displaystyle\sum_{(i,j)\in L^r} y_{kp}^{rij} \leq \phi_e^r, & e \in P_{kp}, k \neq p \in S, \\
& y_{kp}^{rij} \geq 0, & i, j \in V^r : i \neq j, k \neq p \in S, \\
& \theta_k^r, \phi_e^r, x_k^{ri} \in \{0, 1\}, & i \in V^r, k \in S, e \in E.
\end{array}$$

Note that one can even relax the integrality of $\boldsymbol{\theta}^r$ and $\boldsymbol{\phi}^r$ to reduce the complexity of solving a subproblem associated with each request. For each $k \in S$, and $e \in E$, we solve the following problems analytically by checking the coefficients.

$$\begin{aligned}
L^k(\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\eta}) &= \min_{\theta_k \in\{0,1\}}\left(F_k - C_k\lambda_k - M_k\mu_k - \sum_{r\in R}\eta_k^r\right)\theta_k, \\
L^e(\boldsymbol{\sigma}, \boldsymbol{\zeta}) &= \min_{\phi_e \in\{0,1\}}\left(W_e - B_e\sigma_e - \sum_{r\in R}\zeta_e^r\right)\phi_e.
\end{aligned}$$

Consequently, the dual objective denoted by $\Psi(\mathbf{v}^*)$ is

$$\Psi(\mathbf{v}^*) = \sum_{r\in R} L^r(\mathbf{v}^*) + \sum_{k\in S} L^k(\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\eta}) + \sum_{e\in E} L^e(\boldsymbol{\sigma}, \boldsymbol{\zeta}).$$

And the dual problem is then

$$\max_{\mathbf{v}^*} \Psi(\mathbf{v}^*). \tag{3.13}$$

Before the solution procedure of (3.13), we verify that evaluating $\Psi(\mathbf{v}^*)$ is relatively easier than solving ($\mathbb{P}_{\mathrm{V1}}$) itself. In particular, when the number of requests is large such that CPLEX cannot handle (due to the increasing number of binary variables), this decomposition scheme might be attractive. In addition, the lower bound achieved by this decomposition is stronger than the bound obtained by solving the continuous relaxation of ($\mathbb{P}_{\mathrm{V1}}$). We formalize this result below.

Theorem 4 *The optimum of* (3.13) *dominates that of continuous relaxation of* ($\mathbb{P}_{\mathrm{V1}}$).

Proof: According to Theorem 2, the optimum of (3.13) is equivalent to the optimum of minimization of the objective function of ($\mathbb{P}_{\mathrm{V1}}$) over $\mathcal{S}$, where

$$\mathcal{S} = \mathrm{conv}\{\bigtimes_{r\in R} S^r\} \bigcap \{(\mathbf{v}, \mathbf{y}) : (3.8), (3.9), (3.10), (3.11), (3.12)\}\,.$$

$\bigtimes_{r\in R} S^r$ represents the Cartesian product of $(S^r)_{r\in R}$, where S^r represent the constraint set of each subproblem (Sub_r). It also holds that $\mathrm{conv}\{\bigtimes_{r\in R} S^r\} = \bigtimes \mathrm{conv}\{S^r\}$. Moreover, it holds that $\mathcal{S}$ is “smaller” than the continuous relaxation of $\mathbf{F}(\mathbb{P}'_{\mathrm{v1}})$, which completes the proof in conjunction with Lemma 3.2. ■

Theorem 4 also indicates a hierarchy of request based Lagrangian relaxations. We may consider a sequence of covers of set $\{S^r\}_{r\in R}$. For example, we can partition set $\{S^r\}_{r\in R}$ into disjoint subsets and each subset is associated with two requests. The resulting Lagrangian relaxation bound will be larger than the one based on single request decomposition. When the size of each subset is $|R|$, we get problem ($\mathbb{P}'_{\mathrm{v1}}$).

3.4.2 The solution procedure

In this section, we develop a solution procedure of (3.13). Ben-Ameur and Neto [25] proposed the so-called `in-out` algorithm for convex programs, which is treated as an acceleration of cutting-plane and column generation algorithms. We employ this algorithm to solve a Lagrangian dual problem. Let us recall several features of problem (3.13).

- The objective function $\Psi(\mathbf{v}^*)$ is additively composed of functions L^r, L^k, L^e and all functions are concave functions in variables $\mathbf{v}^*$.
- For each input $\mathbf{v}^*$, we have evaluations of functions L^r, L^k, L^e and access to their subgradients at $\mathbf{v}^*$.

Therefore, we are able to construct the following master problem by approximating each concave function with a sequence of subgradients and evaluations.

$$\max \quad \sum_{r\in R} \gamma^r + \sum_{k\in S} \gamma^k + \sum_{e\in E} \gamma^e \tag{$\mathbb{M}$}$$

$$
\begin{aligned}
\text{s.t.}\quad & \gamma^r \le L^r(\mathbf{v}^*) + \sum_{k\in S} w_k^{rn}(\lambda_k - \lambda_k^n) \\
& \quad + \sum_{k\in S} z_k^{rn}(\mu_k - \mu_k^n) + \sum_{k\in S} \theta_k^{rn}(\eta_k^r - \eta_k^{rn}) \\
& \quad + \sum_{e\in E} \kappa_e^{rn}(\sigma_e - \sigma_e^n) + \sum_{e\in E} \phi_e^{rn}(\zeta_e^r - \zeta_e^{rn}), && r\in R, n = 1,\ldots,N, \\
& \gamma^k \le L^k(\boldsymbol{\lambda},\boldsymbol{\mu},\boldsymbol{\eta}) - \alpha_k\theta_k^n(\lambda_k - \lambda_k^n) \\
& \quad - \beta_k\theta_k^n(\mu_k - \mu_k^n) - \theta_k^n\sum_{r\in R}(\eta_k^r - \eta_k^{rn}), && k\in S, n = 1,\ldots,N, \\
& \gamma^e \le L^e(\boldsymbol{\sigma},\boldsymbol{\zeta}) - D_e\phi_e^n(\sigma_e - \sigma_e^n) - \phi_e^n\sum_{r\in R}(\zeta_e^r - \zeta_e^{rn}), && e\in E, n = 1,\ldots,N, \\
& \sum_{r\in R}\gamma^r + \sum_{k\in S}\gamma^k + \sum_{e\in E}\gamma^e \le U \\
& \lambda_k, \mu_k, \sigma_e \ge 0, && r\in R, k\in S, e\in E.
\end{aligned}
$$

where U is an upper bound of $\Psi(\mathbf{v}^*)$. A trivial choice of U can be an upper bound of the primal cost function, e.g., $\sum\limits_{k\in S} F_k + \sum\limits_{e\in E} W_e\phi_e + \sum\limits_{k\in S} A_k \sum\limits_{r\in R} \max\limits_{i\in V^r} c^{ri}$.

To use the `in-out` algorithm, we need to choose a feasible point of the dual problem (3.13). A key observation is that any $\mathbf{v}^*$ defined before is feasible to (3.13). For each $\mathbf{v}^*$, let us denote by $\mathbf{L}(\mathbf{v}^*) \in \mathbb{R}^{|R|+|S|+|E|}$ the vector $((L^r(\mathbf{v}^*))_{r\in R}, (L^k(\boldsymbol{\lambda},\boldsymbol{\mu},\boldsymbol{\eta}))_{k\in S}, (L^e(\boldsymbol{\sigma},\boldsymbol{\zeta}))_{e\in E})$. Correspondingly, any $(\mathbf{v}^*,\ \mathbf{L}(\mathbf{v}^*))$ is a feasible point to the master problem ($\mathbb{M}$). The `in-out` algorithm usually outperforms the standard cutting-plane (or column generation) algorithms since it allows relatively flexible choose of the subsequent evaluation points. Following Ben-Ameur and Neto [25], let us call the evaluation point separation point, the optimal solution of ($\mathbb{M}$) out-point, and call the feasible point in-point. The separation point is chosen as a convex combination of the in-point and out-point. Next, we state the `in-out` Algorithm when applied to problem (3.13).

The choice of the initial in-point: Even though each subproblem (Sub_r) (a MILP) is easier than problem ($\mathbb{P}_{\mathrm{V1}}$), it is still computationally costly. This motivates us to reduce the number of iterations of Algorithm 1. To this end, we may need to select a good in-point with some cost. Specifically, we first solve the continuous relaxation of ($\mathbb{P}'_{\mathrm{v1}}$) to optimality and take the dual multipliers corresponding to constraints (3.8), (3.9), (3.10), (3.11) and (3.12) respectively as the initial values of $\mathbf{v}^*_{in}$. Then we solve the subproblems with $\mathbf{v}^*_{in} \in \mathbb{R}^{|R|+|S|+|E|}$, which outputs $\boldsymbol{\gamma}_{in}$. In the sequel, we shall denote by $\mathbf{v}^{*0}_{in}$ and $\boldsymbol{\gamma}^0_{in}$ the respective initial values of $\mathbf{v}^*_{in}$ and $\boldsymbol{\gamma}_{in}$. We show that at the initialization procedure, we obtain a lower bound that dominates the continuous relaxation of ($\mathbb{P}_{\mathrm{V1}}$).

Lemma 3.3 *The value of* $\sum\limits_{r\in R}\gamma^{r0}_{in} + \sum\limits_{k\in S}\gamma^{k0}_{in} + \sum\limits_{e\in E}\gamma^{e0}_{in}$ *dominates the optimal objective value of the continuous relaxation of* ($\mathbb{P}_{\mathrm{V1}}$)

Proof: First, by Lemma 3.2 and strong duality, the optimum of the continuous relaxation of ($\mathbb{P}_{\mathrm{V1}}$) is equal to the minimization of $\mathcal{L}(\mathbf{v}^{r0}_{in},\ \cdot)$ over continuous relaxation of ($\mathbb{P}'_{\mathrm{v1}}$). Second, observe that all the integrality constraints are not relaxed in subproblems associated with L^r, L^k, L^e. Combining the above observations completes the proof. ■

```
Algorithm 1: In-out Algorithm
Let $(\mathbf{v}^*_{in}, \boldsymbol{\gamma}_{in}), (\mathbf{v}^*_{out}, \boldsymbol{\gamma}_{out})$ be the initial in-point and out-point
choose an $\alpha \in (0, 1]$
while $\sum_{r\in R} \gamma^r_{out} + \sum_{k\in S} \gamma^k_{out} + \sum_{e\in E} \gamma^e_{out} > \sum_{r\in R} \gamma^r_{in} + \sum_{k\in S} \gamma^k_{in} + \sum_{e\in E} \gamma^e_{in} + \epsilon$ do
    define the separation point $(\mathbf{v}^*_{sep}, \boldsymbol{\gamma}_{sep}) = \alpha(\mathbf{v}^*_{out}, \boldsymbol{\gamma}_{out}) + (1-\alpha)(\mathbf{v}^*_{in}, \boldsymbol{\gamma}_{in})$
    evaluate function $L^r, L^k, L^e$ with input $\mathbf{v}^*_{sep}$
    if $\sum_{r\in R} L^r(\mathbf{v}^*_{sep}) + \sum_{k\in S} L^k(\mathbf{v}^*_{sep}) + \sum_{e\in E} L^e(\mathbf{v}^*_{sep}) < \sum_{r\in R} \gamma^r_{sep} + \sum_{k\in S} \gamma^k_{sep} + \sum_{e\in E} \gamma^e_{sep}$
    then
        add cuts at point $\mathbf{v}^*_{sep}$ by concavity
        if $\sum_{r\in R} L^r(\mathbf{v}^*_{sep}) + \sum_{k\in S} L^k(\mathbf{v}^*_{sep}) + \sum_{e\in E} L^e(\mathbf{v}^*_{sep}) > \sum_{r\in R} \gamma^r_{in} + \sum_{k\in S} \gamma^k_{in} + \sum_{e\in E} \gamma^e_{in}$
        then
            $(\mathbf{v}^*_{in}, \boldsymbol{\gamma}_{in}) \longleftarrow (\mathbf{v}^*_{sep}, \mathbf{L}(\mathbf{v}^*_{sep}))$
        end
        solve ($\mathbb{M}$)
    else

    end
    $(\mathbf{v}^*_{in}, \boldsymbol{\gamma}_{in}) \longleftarrow (\mathbf{v}^*_{sep}, \mathbf{L}(\mathbf{v}^*_{sep}))$
end
```

Remark 3.1 *Our numerical results show that the lower bound at the initialization procedure, i.e.,* $\sum_{r\in R} \gamma^{r0}_{in} + \sum_{k\in S} \gamma^{k0}_{in} + \sum_{e\in E} \gamma^{e0}_{in}$*, is generally close to the dual optimum, i.e, the optimal objective value of* (3.13)*. Therefore, for computational benefits, one can terminate Algorithm 1 at the early stage of the algorithms or even after the initialization procedure.*

3.4.3 Numerical experiments

In this section, we carry out some numerical results to evaluate the effectiveness of the proposed decomposition scheme. All test instances have a moderate number of small request (i.e., each request has 5 VMs). Model ($\mathbb{P}_{\mathrm{V1}}$) is used as the benchmark to the request based decomposition. Algorithm 1 is implemented in C++. Numerical results are summarized in Table 3.3.

For each problem instance, we solve model ($\mathbb{P}_{\mathrm{V1}}$) exactly and solve Lagrangian dual problem (3.13) by Algorithm 1. MILPs are solved by CPLEX12.63. To keep the size of model ($\mathbb{P}_{\mathrm{V1}}$) moderate, we put valid inequalities of (Cut') to the `user-cut pool` of CPLEX. Optimality gap is quantified by gap $= \frac{\text{Exact}-v}{\text{Exact}}$, where v is replaced by the optimum of the continuous relaxation of ($\mathbb{P}_{\mathrm{V1}}$) and the lower bound obtained from Algorithm (1) respectively.

Results in Table 3.3 show that the request based decomposition provides strong lower bounds efficiently in the following senses.

1. For small problem instances that CPLEX can solve to optimality in a reasonable time, Algorithm 1 can provide near optimality bound quickly.

2. For instances that CPLEX takes hours of CPU time (e.g. instances with 5 and 6 requests), Algorithm 1 can produce strong bounds in hundreds of seconds. Additionally, Algorithm 1 is generally about ten times faster than the standard CPLEX branch-and-cut algorithm in reaching the same optimality gap (in the last column of Table 3.3.

Table 3.3: Evaluation of Lagrangian decomposition

$(\lvert S\rvert, \lvert E\rvert)$	#Req.	#Bin.	$\mathbb{P}_{V1}$		Decomposition 3.13	
			#CPU(s)	Root gap(%)	#CPU(s)	Gap(%)
(12, 30)	2	162	15.40	9.70	21.38	0
(12, 30)	3	222	65.08	9.26	60.92	0.55
(12, 30)	4	282	125.78	8.52	93.23	1.85
(12, 30)	5	342	3050.21	20.15	139.78	12.72
(12, 30)	6	402	32492.00 (9hs)	24.09	278.56	16.21

Another observation is that as the number of requests increases, the strength of formulation ($\mathbb{P}_{V1}$) and that of Lagrangian decomposition scheme appear weaker. On the one hand, it implies that the convex relaxation of ($\mathbb{P}_{V1}$) is weaker for instances with relatively larger number of requests. On the other hand, it reveals the fact that the relaxed coupling constraints (3.8) and (3.9), (3.10) become more important in ($\mathbb{P}_{V1}$) as the number of requests goes up.

For problem instances having more than 6 requests, CPLEX cannot solve the to optimality within the 10 hours time limit. We use the proposed hierarchy in Section 3.4.1 to generate a sequence of strong bounds in a reasonable time. More precisely, we consider two levels of Lagrangian decomposition. Level 1 refers to the one where the problem ($\mathbb{P}_{V1}$) is decomposed by each single request, each server, and each link. Level 2 refers to the one where we decompose the problem by each pair of requests, each server, and each link.

Due to the lack of exact optimal values, we measure the strength of the above two decomposition schemes by the closed gap defined below

$$\text{closed gap} = \frac{\text{Lagrangian bound} - \text{continuous relaxation value}}{\text{upper bound} - \text{continuous relaxation value}}$$

where the upper bound is the one provided by CPLEX after 10 hours branch-and-cut. The numerical results are summarized in Table 3.3. We highlight the following observations. First, as might be expected, the bound provided by Level 2 decomposition is stronger than the one obtained by Level 1. Second, it appears that as the number of requests increases, Level 2 decomposition does not perform significantly better than the Level 1 decomposition.

Table 3.4: Evaluation of the Lagrangian decomposition hierarchy

$(\lvert S\rvert, \lvert E\rvert)$	#Req.	#Bin.	Level 1		Level 2	
			#CPU(s)	Closed gap(%)	#CPU(s)	Closed gap(%)
(12, 30)	8	522	535.12	38.91	907.42	45.01
(12, 30)	10	642	612.71	32.99	1534.88	35.63
(12, 30)	20	1284	1924.31	18.29	3247.95	20.12

–Time limit: 36000 Seconds (10hs)

Finally, it might be interesting to integrate the request based Lagrangian decomposition scheme into the branch and cut procedure so that the exact solution procedure could be further accelerated.

3.5 A symmetry-induced model

In this section, we reformulate the initial model ($\mathbb{P}$) by exploiting the symmetric structure of the virtual requests. To this end, we introduce the definition below.

Definition 3.1 *For each request r, a virtual machine i is equivalent to a virtual machine j if and only if the following conditions hold.*

- *The frequencies of CPU resource of VM i and j are equal.*
- *The capacity of memory resource of VM i and j are equal.*
- *The throughput requirements of logical links between VM i and VM j are equal, i.e., $f^{rij} = f^{rji}$.*
- *For each VM l where $l \neq i, j$, the throughput requirements of logical links (i, l) and (j, l) are same, i.e., $f^{ril} = f^{rjl}$ and $f^{rli} = f^{rlj}$.*

It is clear that the equivalence is well defined as it is a reflective, symmetric, transitive relation. This definition allows a partition of a request (a set of virtual machines) V^r. Each member of a partition consists of equivalent VMs. We may interpret a member of a partition as a VM *type*. A type t is identified by the frequency of CPU c^t, the memory capacity m^t, and the throughput f^t. We emphasize here that the type we defined here is also associated with the throughput, whereas in practice, the term "type" is generally characterized by CPU, memory (see section 2.4). For the sake of further development, we introduce additional parameters and variables below:

- $\mathcal{T}^r$: set of VM types required in virtual request $r \in R$.
- V^{rt}: set of VMs of type $t \in \mathcal{T}^r$ in request r, i.e., $V^r = \bigcup_{t \in T^r} V^{rt}$.

- x_k^{rt} : 1 if we mapping a type t VM of request r to server k, 0 otherwise.

It is clear that for each request, feasible solutions are equivalent among decision variables associated with same VM type under permutation in the sense that their corresponding objective values are equal. As will be shown below, this simple feature allows us to deduce an equivalent model with fewer binary variables and constraints.

We may also exploit symmetries among edges to reduce the number of (QC). It is obvious that in the context of formulation ($\mathbb{P}$), two edges e and e' are equivalent if and only if $W_e = W_{e'}$, $B_e = B_{e'}$, and they are related to the same set of servers in terms of shortest path routing. As a result, redundant edges do not need to be considered.

Consequently, the objective function becomes

$$\sum_{k\in\mathcal{S}} F_k\theta_k + \sum_{k\in S} A_k \sum_{r\in R}\sum_{t\in\mathcal{T}^r} c^t x_k^{rt} + \sum_{e\in E} W_e\phi_e. \tag{3.14}$$

The assignment constraint (AC) is rewritten as

$$\sum_{k\in S} x_k^{rt} = \left|V^{rt}\right|,\ r\in R, t\in\mathcal{T}^r. \tag{3.15}$$

The location constraint (LC) is replaced with

$$\sum_{t\in\mathcal{T}^r} x_k^{rt} \le \theta_k,\ r\in R, k\in S. \tag{3.16}$$

The knapsack constraint associated with CPU and the one associated with memory constraints can be rewritten as

$$\sum_{r\in R}\sum_{t\in\mathcal{T}^r} c^t x_k^{rt} \le C_k\theta_k, \quad k\in S, \tag{3.17}$$

$$\sum_{r\in R}\sum_{t\in\mathcal{T}^r} m^t x_k^{rt} \le M_k\theta_k, \quad k\in S. \tag{3.18}$$

The bilinear type knapsack constraint can then be replaced by

$$\sum_{r\in R}\sum_{t\in\mathcal{T}^r}\sum_{\substack{k,p\in S:\\ k\neq p, e\in P_{kp}}} f^t x_k^{rt}x_p^{rt} + \sum_{r\in R}\sum_{t,t'\in\mathcal{T}^r}\sum_{\substack{k,p\in S:\\ k\neq p, e\in P_{kp}}} f^{tt'} x_k^{rt}x_p^{rt'} \le D_e\phi_e, \quad e\in E. \tag{3.19}$$

In a special case where VMs within a single request are of same type, (3.19) is equivalent to

$$\sum_{r\in R}\sum_{\substack{k,p\in S:\\ k\neq p, e\in P_{kp}}} f^r x_k^r x_p^r \le D_e\phi_e, \quad e\in E. \tag{3.20}$$

This symmetry-induced formulation is then

$$\min_{x,\theta,\phi}\left\{(3.14) : (3.15), (3.16), (3.17), (3.18), (3.19), x, \theta, \phi \in \{0,1\}\right\}. \tag{$\mathbb{SP}$}$$

Proposition 1 *Formulation ($\mathbb{SP}$) is equivalent to ($\mathbb{P}$) in a sense that for any feasible solution of ($\mathbb{P}$), there exits a feasible solution of ($\mathbb{SP}$) such that their objective values are equal.*

As stated before, bilinear terms and binary variables make ($\mathbb{P}$) and ($\mathbb{SP}$) hard to solve. We compare the sizes of these two problems in terms of the numbers of binary variables and constraints. Notice that the number of variables associated with linear constraints and the number of bilinear constraints in both formulations are same. So we just report in Table 3.5 the number of bilinear variables and bilinear terms and linear constrains. It is obvious that if $|\mathcal{T}^r| < |V^r|$, formulation ($\mathbb{SP}$) is probably attractive. In general, this condition holds in the mapping problem. First, in a virtualized environment, the combination of CPU and memory for virtual machines is designed in an optimal way such that the computing resource of servers can be allocated efficiently. Second, it is really tough for clients to quantify exact the requirements of CPU, memory, and throughputs. Thus cloud operators usually design several VM types. For example, Amazon Elastic Cloud Computing (EC2) mainly provides “small” VMs configured with 1.7 GB memory and 1 compute unit [154]. Third, even if throughput between VMs are not strictly categorized, there exists strong correlations between VM types and throughputs among them, i.e., the “larger” (regarding CPU and memory) two VMs are, the more throughputs will be needed in-between.

Table 3.5: Problem sizes of ($\mathbb{P}$) and ($\mathbb{SP}$)

Model	Bilinear variables	Linear constraints	Bilinear terms
($\mathbb{P}$)	$\sum_{r\in R} \|V^r\|\|S\|$	$\sum_{r\in R} \|V^r\| + (2+\|R\|)\|S\|$	$\|S\|\|V^r\|(\|S\|-1)(\|V^r\|-1)/2$
($\mathbb{SP}$)	$\sum_{r\in R} \|\mathcal{T}^r\|\|S\|$	$\sum_{r\in R} \|\mathcal{T}^r\| + (2+\|R\|)\|S\|$	$\|S\|\|\mathcal{T}^r\|^2(\|S\|-1)/2$

The partition of a request can be done before the solution procedure. First, we partition the request graph by characteristics of CPU and memory. Then for each subset of the partition, we further partition it based on the throughput values.

To conclude this section, we remark that one may employ all the relaxation techniques proposed previously for ($\mathbb{P}$) apply to model ($\mathbb{SP}$) directly for resolution. Yet, implications among constraints need to be examined carefully. For example, if one introduce RLT constraints based on assignment constraints (3.15), the McCormick constraints cannot be eliminated as ($\mathbb{P}_{\text{RLT1}}$). Strong inequalities of the form (Cut) cannot be employed either. In brief, we will use the following formulation to solve ($\mathbb{SP}$).

$$\mathbb{SP}_{\text{V1}} : \{(\mathbb{SP})\} \cap \{(\text{AC}_{\text{RLT}}), (3.1\text{a}) - (3.1\text{d}), (\text{Cut'})\}\,. \qquad (\mathbb{SP}_{\text{V1}})$$

Note also that the strength of formulation ($\mathbb{SP}_{\text{V1}}$) is usually weaker than the general formulation ($\mathbb{P}_{\text{V1}}$).

3.5.1 A heuristic

For model ($\mathbb{SP}$), it is clear that when $|\mathcal{T}^r| = 1$, the benefits can be significant and when $|\mathcal{T}^r| = |V^r|$, the two models are equivalent. However, for problem instances not satisfying the equivalence conditions defined in 3.1, the advantages of model ($\mathbb{SP}$) can hardly manifest. To address this issue, we propose a heuristic to obtain a feasible solution with an upper bound.

Assume that $|\mathcal{T}^r|$ is large and we want to find a feasible solution of ($\mathbb{P}$) in a reasonable time. Clearly the VMs can be easily categorized regarding the requirements of CPU and memory, thus we elaborate on partitioning VMs by throughput values. Intuitively, VMs with similar f^{rij} will be grouped in the same set. Meanwhile, we want to minimize the amount of throughput among two subsets. To this aim, we consider minimizing the normalized cut (that minimizes the multi-cuts while balancing sizes of each subgraph) for a virtual request graph G^r. The problem itself is NP-hard, but we might as well to solve it approximately usng the well-known `shi-malik` algorithm [147]. To be specific, we proceed the following. For VMs in the same subset, every throughput value is augmented to the maximum; for VMs among two different subsets, each mutual throughput value is shifted to the largest one. This setting allows us to make use of model $\mathbb{SP}$, which outputs a feasible solution and an upper bound. Intuitively, the upper becomes stronger as the number of subsets decreases. When the number of subsets is equal to the number of VMs in G^r, we get the global optimum of the mapping problem.

3.5.2 Numerical experiments

In this section, we test the effectiveness of model ($\mathbb{SP}$) numerically. We choose to test on problem instances of a single request with 10, 20, 30, 40 VMs respectively. Model ($\mathbb{SP}$) and the aforementioned heuristic coded in C++.

We carry out experiments on two sets of problem instances. For all problem instances, VMs are categorized by CPU and Memory. In addition, for the first set of problem instances, throughput among a same category of VMs are assumed to be the same, whereas for the second set of problem instances, throughput values are usually different.

Numerical results of the first problem instance set are summarized in Table 3.6. Note that the optimum of ($\mathbb{SP}_{\text{V1}}$) is exactly the optimum of ($\mathbb{P}_{\text{V1}}$). As expected, the symmetry-induced model ($\mathbb{SP}_{\text{V1}}$) performs significantly better than ($\mathbb{P}_{\text{V1}}$) for all the tested problem instances. This is mainly due to the reduction of binary variables and bilinear terms. However, as the size of the physical graph increases, the solution time of ($\mathbb{SP}_{\text{V1}}$) also increases exponentially. For example, for 10 VMs, it takes around 400 seconds for instance $(25, 90)$ while it just takes 1.22 seconds for instance $(12, 30)$.

Now let us focus on the second problem instance set. Following the observation from [122] that 80% of throughput values are quite low and 4% of them have a 10 times higher rate, we generate throughput values with a probability 0.8 being lower than 2 Mbps/min and with a probability 0.2 lying between 10 Mbps/min and 30 Mbps/min. We employ the heuristic proposed in Section 3.5.1 to partition virtual request G^r to N subsets. Results are summarized in Table 3.7. The optimality gap induced by an upper bound is quantified by Gap $= \frac{\text{Upper bound - Exact value}}{\text{Exact value}}$. Exact values are computed using ($\mathbb{P}_{\text{V1}}$). We stress two observations:

- For each problem instance, as might be expected, the quality of upper bounds become better as the size of the partition increases.

Table 3.6: Performances of ($\mathbb{SP}_{\mathrm{V1}}$) and ($\mathbb{P}_{\mathrm{V1}}$)

$(\|S\|, \|E\|)$	#VMs.	($\mathbb{SP}_{\mathrm{V1}}$)		($\mathbb{P}_{\mathrm{V1}}$)	
		Cpu(Sec.)	#Nodes	#Cpu(sec.)	#Nodes
(12, 30)	10	1.22	0	13.68	0
(25, 90)	10	424	121	1216	30068
(25, 90)	20	1080	150	3680	73808
(40, 178)	10	522	0	1024	25197
(40, 178)	20	824	121	13428(3.7hs)	310124
(40, 178)	30	4780	875	-	-
(40, 178)	40	4410	1950	-	-

Time limit: 36000 Seconds (10hs).

- As the size of the problem instance increases, the gap becomes larger and the influence of the partition size N becomes more significant.

Table 3.7: Upper bounds of ($\mathbb{P}$)

$(\|S\|, \|E\|)$	#VMs.	N	#Cpu(sec.)	#Nodes	Gap(%)
(12, 30)	10	2	2.72	0	9.22
(12, 30)	10	4	3.67	0	4.21
(25, 90)	10	2	739	121	11.69
(25, 90)	10	4	1280	187	8.90
(25, 90)	20	4	2182	478	23.70
(25, 90)	20	7	4821	157	15.84
(25, 90)	25	4	1328	952	24.33
(25, 90)	25	10	3120	1253	13.89

To conclude this section, we remark that there is still a large number of symmetries in model ($\mathbb{SP}_{\mathrm{V1}}$). For example, in a physical network, two servers k, p might be equivalent in the sense that $(\theta_k, \theta_p) = (0, 1)$ or $(\theta_k, \theta_p) = (1, 0)$ leads to the same mapping cost. One may introduce symmetry breaking constraints statically to the model or dynamically during a branching-and-cut procedure.

3.6 A perspective

In practice, the arrival of VMs is dynamic and unpredictable. The static mapping model ($\mathbb{P}$) can hardly work in that case. In this section, we propose a model that optimizes certain mapping policies as a perspective.

In a cloud computing environment, requests might be categorized. For example, let us consider the cloud service of Platform as a Service (PaaS), where platforms might be Linux, Windows, Unix, etc.. Let set R be set of request types, and let $\mathbf{x}^r$ denote the mapping polices for requests of type r. Specifically, we may interpret x_k^{ri} as the probability that VM i will be assigned to sever k. The model can be developed simply by relaxing the integrality constraints of $\mathbf{x}$ in model ($\mathbb{P}$). The objective function turns out to be the expectation of future cost. When requests of type r arrive, cloud provider will map x_k^{ri} percent of VM i to server k.

Note that unlike a MILP or a convex MIQCP, relaxing integrality constraints of $\mathbf{x}$ does not lead to an easier problem. In fact, it is computationally more challenging than problem ($\mathbb{P}$). For problem ($\mathbb{P}$), one can get the optimal solution and objective value using standard finite branch and bound algorithms. For this model where continuous variables are involved in bilinear constraints (QC), one may need to consider spatial branch and bound framework for two reasons. First, the optimal solution to a given relaxation is not necessarily feasible. Second, the convex underestimators of bilinear constraints may need to be updated in each subdivision of the search region. Many related fundamental problems, such as, convex and concave envelopes of a general bilinear function over a hypercube need to be investigated.

3.7 Conclusion

We proposed a model for the optimal mapping of virtual machines in a cloud environment, by taking into account link capacity constraints, location constraints, and knapsack constraints regarding CPU and memory. We then strengthened the model by employing different levels of RLT and adding strong valid inequalities. Computational results confirm the power of valid inequalities, especially those that exploit the problem structure. To handle relatively larger problem instances, we proposed a request based Lagrangian decomposition scheme. We show, both theoretically and numerically, that this decomposition can produce bounds greatly stronger than continuous relaxation bounds. In addition, a novel model exploiting symmetries of a request was discussed and analyzed. Finally, we pointed out that it would be interesting to associate each request with certain mapping policy and investigate related techniques to solve the resulting model.

Appendix: numerical comparison of model $(\mathbb{P}_{\text{MC}})$ and model $(\mathbb{P}_{\text{GL}})$

We provide numerical results on some instances to compare the differences of two different models $(\mathbb{P}_{\text{MC}})$ and $(\mathbb{P}_{\text{GL}})$.

Table 3.8: Continuous relaxation statistics of $(\mathbb{P}_{\text{MC}})$ and $(\mathbb{P}_{\text{GL}})$

$(\lvert S\rvert, \lvert E\rvert)$	#Req.	$\mathbb{P}_{\text{MC}}$		$\mathbb{P}_{\text{GL}}$	
		Cpu(Sec.)	Gap(100%)	#Cpu(sec.)	Gap(100%)
(8,20)	1	0.00	63.9	0.00	63.9
(8,20)	2	0.02	60.8	0.01	60.8
(12,30)	1	0.02	79.6	0.01	79.6
(12,30)	2	0.07	74.7	0.06	74.7
(12,30)	4	0.38	65.88	0.64	65.88

Table 3.9: Branch and Cut statistics of $(\mathbb{P}_{\text{MC}})$ and $(\mathbb{P}_{\text{GL}})$

$(\lvert S\rvert, \lvert E\rvert)$	#Req.	$\mathbb{P}_{\text{MC}}$		$\mathbb{P}_{\text{GL}}$	
		Cpu(Sec.)	#Nodes	#Cpu(sec.)	#Nodes
(8,20)	1	0.29	0	0.53	0
(8,20)	2	0.45	229	0.38	251
(12,30)	1	1.18	16	0.52	284
(12,30)	2	134	8897	129	9016
(12,30)	4	4180	10766	3966	14750

Chapter 4

Convex and concave envelopes of bilinear functions

4.1 Summary

Convex and concave envelopes of nonconvex functions play an important role in global optimization. We discuss in this chapter a couple of approaches to approximating the convex and concave envelopes of bilinear functions, especially over hypercubes due to its generality. The first approach is based on a semidefinite reformulation. The second approach considers some predefined set covers of a hypercube and leads to a linear program. Then we establish a connection between the convex envelope of a bilinear function and the concave envelope of a polyhedral function. Numerical experiments are conducted to compare the two approaches. As a possible extension, a novel approach to approximate the envelopes is discussed and illustrated.

4.2 Introduction

There has been a great amount of studies devoted to developing convex underestimators and concave overestimators of nonlinear functions $f(\mathbf{x})$ over various polyhedral sets. One of the most important motivations in optimization community is that, one can reformulate a complicated usually non-convex problem as an easier problem with convex representations of the objective function and constraints. Such convex reformulations can then be solved repeatedly in branch-and-bound algorithms, where searching spaces are refined in a convergent way. In general, the computational efficiency of branch-and-bound is greatly influenced by the strength of convex reformulations.

Among such investigations, the construction and approximations of convex and concave envelopes of bilinear functions over boxes draws much attention. This is mainly due to the following reasons. First, many important problems, for example, quadratic assignment problems [81, 153], pooling and blend [70, 13, 125] involve bilinear terms. Second, boxes are naturally outer approximations of polytopes and therefore the estimators of convex and concave envelopes of f over a box are also valid estimators for function f over polytopes belonging to the box. In addition, most branch-and-bound algorithms partition the searching space by divisions of the feasible region into boxes, which also motivates investigations in deriving strong convex and concave estimators over boxes. It is well known that a n-dimensional box is simply linear transformations of a hypercube. Therefore, we consider in this chapter the convex (resp. concave) envelope of a bilinear function (4.1) over a hypercube $H_n = [0,1]^n$:

$$f(\mathbf{x}) = \sum_{i<j} Q_{ij} x_i x_j \tag{4.1}$$

where $Q_{ij} \in \mathbb{R}$, $1 \leq i < j \leq n$. There are two main research directions on the construction of envelopes of f over H_n.

The first direction is motivated by the definition that the convex envelope of any bounded f over any non-empty compact set P is a function

$$\breve{f} : P \ni \mathbf{x} \mapsto \breve{f}(\mathbf{x}) = \inf \{u : (\mathbf{x}, u) \in \operatorname{conv} \operatorname{epi} f\},$$

where $\operatorname{epi} f = \{(\mathbf{x}, u) : f(\mathbf{x}) \leq u\}$ and conv of a set represents its convex hull. Geometrically, one can interpret the convex envelope of f as the bottom of the convex hull of the epigraph of f over P. Since P is compact, the definition is equivalent to

$$\min_{\lambda_i} \left\{ \sum_{\mathbf{v}_i \in P} \lambda_i f(\mathbf{v}_i) : \lambda_i \mathbf{v}_i = \mathbf{x}, \sum_{v_i \in P} \lambda_i = 1, \lambda_i \geq 0 \right\}. \tag{4.2}$$

The concave envelope $\widehat{f}(\mathbf{x})$ of f over P is defined symmetrically by replacing minimization to maximization. When $P = H_n$, a nice property of bilinear function f due to Sherali [146] and Rikun [140] is that its convex and concave envelopes over a hypercube H_n are vertex polyhedral, i.e., the envelopes of f on H_n coincide with the envelopes of its restriction to the vertices of H_n. This property allows us to simplify the definition (4.2) to the following,

$$\breve{f}(\mathbf{x}) = \min_{\lambda_i} \left\{ \sum_{i=1}^{2^n} \lambda_i f(\mathbf{v}_i) : \sum_{i=1}^{2^n} \lambda_i \mathbf{v}_i = \mathbf{x}, \sum_{i=1}^{2^n} \lambda_i = 1, \lambda_i \geq 0, i = 1, \dots, 2^n \right\}, \tag{4.3}$$

where $\mathbf{v}_i, i = 1, \dots, 2^n$ are vertices of H_n. However, as the number of extreme points of a hypercube is exponential in the dimension, it is known that a full characterization of such polyhedral envelopes is in general difficult. The characterizations of convex and concave envelopes in lower dimensional hypercubes have been studied in [123].

The second research line can be treated as the dual of first research direction since the convex (resp. concave) envelope is interpreted as the pointwise supreme (resp. infimum) of an affine under-(resp. over-) estimator of f over P. For example, the convex envelope is

$$\breve{f}(\mathbf{x}) = \max_{\boldsymbol{\alpha}, \gamma} \ \{\gamma : \boldsymbol{\alpha}^T (\mathbf{x} - \mathbf{v}) + \gamma \leq f(\mathbf{v}), \ \forall \ v \in P\}, \tag{4.4}$$

where $(\boldsymbol{\alpha}, \gamma) \in \mathbb{R}^{n+1}$ defines the supporting hyperplane of the $\breve{f}$ at point $\mathbf{x}$. When $P = H_n$, one can replace set P with its extreme points due to the vertex polyhedral property of the convex envelope of f. However, the number of extreme points is exponential. A cutting-plane algorithm was proposed in [18] to find a facet of the convex envelope by separating the supporting function $(\boldsymbol{\alpha}, \gamma) \in \mathbb{R}^{n+1}$ at a pre-defined point. One of the most frequently used relaxations to approximate the convex and concave envelopes of $f(\mathbf{x})$ over a hypercube due to McCormick [119] is the following inequalities

$$\max\{x_i + x_j - 1, 0\} \leq x_i x_j \leq \min\{x_i, x_j\}, \; i < j. \tag{4.5}$$

We will refer to these inequalities as McCormick inequalities or McCormick relaxation. McCormick also pointed out that (4.5) characterize the exact convex and concave envelope over rectangles. One can also find detailed proof in [6]. However, for higher dimensional ($n \geq 3$) cases, term-wise McCormick relaxation, in general, do not provide a characterization of the convex and concave envelope over a hypercube. The triangle inequalities proposed by Padberg [132] in concert with McCormick inequalities provide an explicit characterization of convex and concave envelopes of $f(\mathbf{x})$ over a cube ($n = 3$). As noticed by [123], the number of facets defining the convex and concave envelopes is indicated by the number of simplices induced by the *triangulations* of a hypercube and unfortunately, the number grows super-exponentially in the number of dimension. Locatelli and Schoen [114] utilized the definition (4.4) and derived explicit convex envelopes for some general bivariate functions over various polytopes in $\mathbb{R}^2$.

A number of papers are also dedicated to giving explicit characterizations of convex and concave envelopes over special polytopes. For example, Locatelli and Schoen [114] used the definition (4.4) and derived explicit convex envelopes for some general bivariate functions over various polytopes in $\mathbb{R}^2$. Sherali and Alameddine derived the convex envelope of a bivariate function over the so-called D-polytope. Linderoth [113] analytically derived the explicit functions for the disjoint triangular regions of a rectangle via disjunction $x \leq y$ or $x \geq y$. Tawarmalani et al. [150] derived convex and concave envelopes for several nonlinear functions via detecting certain polyhedral divisions of a hyper-rectangle.

To deal with the difficulty in (4.3) and (4.4), one natural approach is to consider extended formulations with moderate sizes thereby obtaining strong convex and concave estimators. However, as observed by Luedtke et al. [116], extended formulations do not always bring improvements compared with McCormick relaxations (4.5). A comprehensive study on the strength of McCormick relaxations of multilinear functions over a hypercube is provided in [116]. In particular, it is shown that if the coefficients of a bilinear function $f(\mathbf{x})$ defined as (4.1) are nonnegative, i.e., $Q_{i,j} \geq 0, \; \forall \; i < j$, the ratio of the gap between McCormick's overestimator and underestimator constructed via (4.5) over the true gap of the convex and concave envelopes is always less than 2. As proposed by Lasserre and Thanh [105], one can also construct a convex polynomial $p_d \in \mathbb{R}[\mathbf{x}]$ with degree no greater than d that underestimates f. Based on Putinar's Positivstellensatz [137], p_d can be approximated as close as desired by solving a hierarchy of semidefinite programs.

We investigate in this chapter the strength of different approaches on approximating the convex and concave envelopes of the bilinear f over H_n. The outline and contributions are summarized below.

Outline and contributions. In Section 4.3, we firstly review a semidefinite program (SDP) based estimator, which is based on the intersection of a SDP constraint and Mc-

Cormick inequalities. This work has been proposed by Anstreicher [11]. Our contribution lies in some theoretical results on two variants of formulations on bilinear functions as a special case of general quadratic functions over a box. As a consequence, an alternative proof to a result from Burer and Letchford [50] is provided. Then we propose a novel approach by considering a predefined tool to approximate the convex and concave envelope of f over H_n. This approach leads to a linear reformulation and we are able to establish connections with various inequalities and the reformulation-linearization-technique (RLT) method proposed in [2]. Finally, we show that the convex (resp. concave) envelope of a bilinear function f is affinely equivalent to the concave (resp. convex) envelope of a polyhedral function. In Section 4.4, we report a set of numerical results on a number of problem instances to compare these two approaches. Following the numerical results, we propose an interesting research direction on approximating the convex envelope valid over a hypercube when all coefficients of f are nonnegative. Concluding remarks follow in Section 4.6.

Notation. Throughout this chapter, for any matrix $\mathbf{C}$, $\mathbf{C}^T$ denotes its transpose and C_{ij} represents its i^{th} row j^{th} column component. $\mathbf{C}_{I,J}$ represents the submatrix composed by set I of columns of $\mathbf{C}$ and set J of rows of $\mathbf{C}$. When $I = J$, we simply use $\mathbf{C}_I$ in place of $\mathbf{C}_{I,I}$. For a set $\mathbf{S}$, we use $\operatorname{conv} S$ to represent its convex hull. If S is finite, we use $|S|$ to represent its cardinality. We use H_n to represent an n-dimensional hypercube, i.e., $H_n = [0,1]^n$. For a general function $f : H_n \mapsto \mathbb{R}$, we denote by $\check{f}$ (resp.$\widehat{f}$) the convex (resp. concave) envelope of f over an n-dimensional hypercube. The inner product between two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times n}$ are denoted by $\langle \mathbf{A}, \mathbf{B} \rangle$. For a square matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{diag}(\mathbf{A}) \in \mathbb{R}^n$ represents the vector of diagonal elements. Conversely, for a vector $\mathbf{a} \in \mathbb{R}^n$, $\mathrm{Diag}(a) \in \mathbb{S}^n$ stands for the $n \times n$ diagonal matrix filled by vector a. For a symmetric matrix $\mathbf{X} \in \mathbb{S}^n$ and vector $x \in \mathbb{R}^n$, we use $\mathcal{S}(\mathbf{x}, \mathbf{X})$ to represent the matrix $\begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \mathbf{X} \end{pmatrix}$ in $\mathbb{S}^{n+1}$. We use vector $\tau(\mathbf{X}) = (X_{ij})_{1\le i\le j\le n}$ to represent a vector that is the upper triangular part of the matrix $\mathbf{X}$ and $\tau_+(\mathbf{X}) = (X_{ij})_{1\le i<j\le n}$ to represent the strictly upper triangular part.

4.3 Main results

In this section, we first review a SDP based convex (resp. concave) underestimator (resp. overestimator) and establish some connections between two formulations. Then, we propose a novel way to construct estimators based on a pre-defined cover set. Finally, we conclude this section by making a connection with the convex and concave envelopes of a polyhedral function, which implies some results in the literature.

4.3.1 SDP based estimators

Given a vector x, we are interested in constructing a convex under-estimator and a concave over-estimator that are valid over a hypercube. Towards this aim, we consider the following assumption.

Assumption 2 *A set of valid inequalities involving quadratic terms is available. For each $k \in \mathcal{K}$,*

$$\mathbf{x}^T \mathbf{A}_k \mathbf{x} + \mathbf{b}_k^T \mathbf{x} + d_k \leq 0,\ \forall \mathbf{x} \in H_n, \tag{4.6}$$

where $\mathcal{K}$ is an index set.

Note that Assumption 2 in general holds for bilinear optimization problems over any compact set. For example, such valid inequalities can be McCormick's inequalities (4.5), Padberg's triangle inequalities [132], RLT based inequalities [2]. The principal idea is to use these valid inequalities to construct strong estimators. Let us start by constructing the convex underestimator. We associate nonnegative multipliers $\boldsymbol{\alpha} \in \mathbb{R}_+^{|\mathcal{K}|}$ with the set of inequalities. Then we add the negative terms $\sum\limits_{k\in\mathcal{K}} \boldsymbol{\alpha}_k(\mathbf{x}^T\mathbf{A}_k\mathbf{x} + \mathbf{b}_k^T\mathbf{x} + d_k)$ to $f(x)$. For ease of presentation, we define $f_{\boldsymbol{\alpha}}(\mathbf{x}) = \mathbf{x}^T\mathbf{Q}(\boldsymbol{\alpha})\mathbf{x} + \mathbf{c}(\boldsymbol{\alpha})^T\mathbf{x} + p(\boldsymbol{\alpha})$, where $Q(\boldsymbol{\alpha}) = \frac{1}{2}Q + \sum\limits_{k\in\mathcal{K}} \boldsymbol{\alpha}_k\mathbf{A}_k, \mathbf{c}(\boldsymbol{\alpha}) = \sum\limits_{k\in\mathcal{K}} \alpha_k\mathbf{b}_k$ and $p(\boldsymbol{\alpha}) = \sum\limits_{k\in\mathcal{K}} \alpha_k d_k$.

Recall that $f_{\boldsymbol{\alpha}}$ is a convex underestimator of f over H_n if and only if $f_{\boldsymbol{\alpha}} \leq f$ and $f_{\boldsymbol{\alpha}}$ is convex over the hypercube H_n. Obviously, it holds that $f_{\boldsymbol{\alpha}}(\mathbf{x}) \leq f(\mathbf{x}),\ \forall \mathbf{x} \in H_n$. The convexity of $f_{\boldsymbol{\alpha}}$ can be ensured by restricting $Q(\boldsymbol{\alpha})$ to be positive semidefinite. Let us denote by $\mathcal{A}$ the set of feasible choices of α:

$$\mathcal{A} := \left\{\boldsymbol{\alpha} \in \mathbb{R}_+^n : \mathbf{Q}(\boldsymbol{\alpha}) \in \mathbb{S}_+^n\right\}.$$

We ensure the non-emptiness of set $\mathcal{A}$ by explicitly adding valid inequalities $x_i^2 \leq x_i$ ($i = 1, \ldots, n$) to inequality system (4.6). As a consequence, it also follows that $\mathcal{A}$ has a nonempty interior. The strongest convex underestimator of the form $f_{\boldsymbol{\alpha}}$ can be obtained by solving the following problem

$$\sup_{\boldsymbol{\alpha}\in\mathcal{A}}\ f_{\alpha}(\mathbf{x}). \tag{4.7}$$

Problem (4.7) is then strictly feasible. The dual problem of (4.7) reads

$$\begin{aligned} \inf_{\mathbf{X}} \quad & \frac{1}{2}\langle \mathbf{Q}, \mathbf{X} + \mathbf{x}\mathbf{x}^T\rangle \\ \text{s.t.} \quad & \langle \mathbf{A}_k, \mathbf{X} + \mathbf{x}\mathbf{x}^T\rangle + \mathbf{b}_k^T\mathbf{x} + d_k \leq 0,\ \forall k \in \mathcal{K}, \\ & \mathbf{X} \in \mathbb{S}_+^n. \end{aligned} \tag{4.8}$$

The feasible region of (4.8) is nonempty (e.g., take $\mathbf{X}$ as the null matrix of size n). In addition, as (4.7) is strictly feasible, the dual optimum is attained and is equal to the primal optimum of (4.7).

Remark 4.1 *Note that in model (4.7), we just exploit the information of a set of valid inequalities in $\mathcal{K}$ and we restrict $\mathbf{Q}(\boldsymbol{\alpha}) \in \mathbb{S}_+^n$. Other than using more valid inequalities, we may replace the SDP matrix cone with "larger" convex cones in geometric sense e.g, copositive cone, in the hope that the underestimator values will be increased. From its dual model (4.8), we see that if we replace the SDP matrix cone with a "smaller" convex cone, e.g., completely positive matrix cone, the underestimator values will be higher. However, checking the membership of copositive cone is co-NP-complete [127] and checking if a matrix is completely positive cone is NP-hard [62]. One can approximate the copositive* *matrix cone by* doubly nonnegative *matrix cone, i.e., the intersection of SDP cone and componentwise nonnegative symmetric matrix cone. For more details, we refer readers to [11, 84].*

By change of variables $\mathbf{X}' = \mathbf{X} + \mathbf{x}\mathbf{x}^T$, one can rewrite (4.8) as

$$\min_{\mathbf{X}'} \left\{ \frac{1}{2}\langle \mathbf{Q}, \mathbf{X}' \rangle : (\mathbf{x}, \mathbf{X}') \in \mathbf{K} \right\}, \tag{4.9}$$

where $\mathbf{K} := \big\{(\mathbf{x}, \mathbf{X}') \in H_n \times \mathbb{S}^n : \langle \mathbf{A}_k, \mathbf{X}' \rangle + \mathbf{b}_k^T \mathbf{x} + d_k \leq 0, \ \forall k \in \mathcal{K}, \mathcal{S}(\mathbf{x}, \mathbf{X}') \succeq 0\big\}$. Notice that when $\mathbf{x}$ is not fixed, problem (4.9) is exactly the Shor's relaxation in conjunction with valid inequalities. Formally, we summarize the foregoing as follows.

Theorem 5 *The strength of the SDP relaxation* $\min\limits_{(\mathbf{x},\mathbf{X}')\in\mathbf{K}} \frac{1}{2}\langle \mathbf{Q}, \mathbf{X}' \rangle$ *is equivalent to pointwise supremum of quadratic convexifications of quadratic functions defined by* (4.7).

Theorem 5 clearly implies that any quadratic convexification based underestimator is dominated by SDP-based estimator. This result in fact coincides with Theorem 2 in [11], where they claim that the well-known $\boldsymbol{\alpha}$BB method (c.f., [4, 148]) are dominated by SDP based relaxation in conjunction with valid inequalities. Recall that convex underestimators of a bilinear function f are usually used to compute lower bounds of good quality over a hypercube for a nonconvex problem involving f. Therefore, for practical applications, it might be worth computing a sequence of $(\boldsymbol{\alpha})_{i=1}^k$ by solving (4.7) approximately. Then a sequence of underestimators $(f_{\boldsymbol{\alpha}_i})_{i=1}^k$ can be obtained. A formula of the convex underestimator $\underline{f}$ of a bilinear function f over a hypercube H_n is then

$$\underline{f}(\mathbf{x}) = \max_{\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_k} f_{\boldsymbol{\alpha}_i}(\mathbf{x}).$$

By definition, $\mathbf{K}$ is closed, convex, and bounded. Let us consider a SDP based convex underestimator defined by

$$\check{f}_{sdp}(\mathbf{x}) := \min_{\mathbf{X}\in F(\mathbf{x})} \frac{1}{2}\langle \mathbf{Q}, \mathbf{X} \rangle \tag{4.10}$$

where $F(\mathbf{x}) := \{\mathbf{X} \in \mathbb{S}^n : (\mathbf{x}, \mathbf{X}) \in \mathbf{K}, X_{ii} \leq x_i \ (i = 1, \ldots, n)\}$. For the sake of clarity, $X_{ii} \leq x_i \ (i = 1, \ldots, n)$ are explicitly written out despite the redundancy w.r.t. set $\mathbf{K}$. The SDP based concave overestimator $\widehat{f}_{sdp}$ can be defined symmetrically. An important observation is that values of $\widehat{f}_{sdp}(\mathbf{x}), \check{f}_{sdp}(\mathbf{x})$ do not change if we replace $X_{ii} \leq x_i$ with $X_{ii} = x_i$. To see this, let us introduce set $F'(\mathbf{x}) := \{\mathbf{X} \in \mathbb{S}^n : (x, X) \in \mathbf{K}, X_{ii} = x_i \ (i = 1, \ldots, n)\}$ and define $\mathcal{F}(\mathbf{x}) := \left\{\tau(\mathbf{X}) \in \mathbb{R}^{\frac{n(n+1)}{2}} : \mathbf{X} \in F(\mathbf{x})\right\}$ and $\mathcal{F}'(x) := \left\{\tau_+(\mathbf{X}) \in \mathbb{R}^{\frac{n(n-1)}{2}} : \mathbf{X} \in F'(\mathbf{x})\right\}$.

We show below that by considering the $n(n-1)/2$ components of $\tau(\mathbf{X})$ corresponding with the strictly upper triangular part of X we get a member of $\mathcal{F}'(\mathbf{x})$.

Theorem 6 *For any* $\mathbf{x} \in H_n$, $\operatorname{Proj} \mathcal{F}(\mathbf{x}) = \mathcal{F}'(\mathbf{x})$.

Proof: It is clear that $\mathcal{F}'(\mathbf{x}) \subseteq \operatorname{Proj} \mathcal{F}(\mathbf{x})$. We now show that the reverse also holds. Let $(X_{ij})_{i \leq j}$ be any point of $\mathcal{F}(\mathbf{x})$. It holds that $\mathbf{X} - \mathbf{x}\mathbf{x}^T \succeq \mathbf{0}$. Since $X_{ii} \leq x_i$, it holds that $\operatorname{Diag}(x - \mathbf{diag}(X)) + \mathbf{X} - \mathbf{x}\mathbf{x}^T \succeq \mathbf{0}$. By Schur's lemma, it means that $\mathbf{X} + \operatorname{Diag}(\mathbf{x} - \mathbf{diag}(\mathbf{X})) \in F'(\mathbf{x})$ and therefore $(X_{ij})_{i<j} \in \mathcal{F}'(\mathbf{x})$. Hence $\operatorname{Proj} \mathcal{F}(\mathbf{x}) \subseteq \mathcal{F}'(\mathbf{x})$. ■

Theorem 6 immediately yields the desired result.

Corollary 4.1 *If the objective function is bilinear, i.e.,* $\mathbf{diag}(\mathbf{Q}) = 0$*, then for each* $\mathbf{x} \in H_n$*, the following equalities hold*

$$\min_{\mathbf{X} \in F(\mathbf{x})} \langle \mathbf{Q}, \mathbf{X} \rangle = \min_{\mathbf{X} \in F'(\mathbf{x})} \langle \mathbf{Q}, \mathbf{X} \rangle \text{ and } \max_{\mathbf{X} \in F(\mathbf{x})} \langle \mathbf{Q}, \mathbf{X} \rangle = \max_{\mathbf{X} \in F'(\mathbf{x})} \langle \mathbf{Q}, \mathbf{X} \rangle.$$

Proof: Since the diagonal associated terms vanish, the support of the objective function is essentially $\operatorname{Proj} \mathcal{F}(\mathbf{x})$, which is exactly $\mathcal{F}'(\mathbf{x})$ by Theorem 6. ■

Now let us consider set QPB_n and the polytope BQP_n that are explicitly studied in [50], where

$$\text{QPB}_n = \text{conv}\left\{ (\mathbf{x}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^{\frac{n(n+1)}{2}} : y_{ij} = x_i x_j, 1 \le i \le j \le n,\ x \in H_n \right\},$$
$$\text{BQP}_n = \text{conv}\left\{ (\mathbf{x}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^{\frac{n(n-1)}{2}} : y_{ij} = x_i x_j, 1 \le i < j \le n,\ x \in \{0,1\}^n \right\}.$$

The following result due to Burer and Letchford [50] is a consequence of Theorem 6.

Corollary 4.2 $\operatorname{Proj}_{\mathbf{x},(y_{ij})_{i<j}} \text{QPB}_n = \text{BQP}_n$

Proof: It is trivial that $\text{BQP}_n \subseteq \operatorname{Proj}_{\mathbf{x},(y_{ij})_{i<j}} \text{QPB}_n$. We show that the reverse also holds. Let all valid inequalities of QPB_n be included in set $\mathbf{K}$, then we have $\text{QPB}_n = \{(\mathbf{x}, \tau(\mathbf{X})) : \mathbf{X} \in F(\mathbf{x}), \mathbf{x} \in H_n\}$. By Theorem 6, we have $\operatorname{Proj}_{\mathbf{x},(y_{ij})_{i<j}} \text{QPB}_n = \{(\mathbf{x}, \tau_+(\mathbf{X})) : \mathbf{X} \in F'(\mathbf{x}), \mathbf{x} \in H_n\}$. Let $(\overline{\mathbf{x}}, \tau_+(\overline{\mathbf{X}})) \in \operatorname{Proj}_{\mathbf{x},(y_{ij})_{i<j}} \text{QPB}_n$, then there exist some points $(\mathbf{x}^k)$ with associated convex combination coefficients $\lambda_k \ge 0, \sum_k \lambda_k = 1$ such that $\overline{\mathbf{x}} = \sum_k \lambda_k \mathbf{x}^k$ and $\overline{\mathbf{X}} = \sum_k \lambda_k \mathbf{x}^k (\mathbf{x}^k)^T$. In addition, $X_{ii} = x_i$ $(i = 1, \dots, n)$ holds. These facts lead to $x_i^k \in \{0,1\}$, $\forall \lambda_k > 0$ $(i = 1, \dots, n)$. Therefore the extreme points of $\operatorname{Proj}_{\mathbf{x},(y_{ij})_{i<j}} \text{QPB}_n$ are binary, showing that $(\overline{\mathbf{x}}, \tau_+(\overline{\mathbf{X}})) \in \text{BQP}_n$. ■

4.3.2 Cover based estimators

In this section, we present a novel way to construct convex and concave estimators via linear programming. To formalize the idea, we introduce definitions below.

Definition 4.1 *A cover of a set* S *is a collection* C *of subsets of* S *whose union is the whole set* S*, i.e.,* $S = \bigcup_{c \in C} c$.

It is clear that $\{S\}$ is a cover of itself. Any partition of S is a cover. To highlight the fact that BQP_n defined above is associated with a set $J \subseteq \{1, \dots n\}$, we will use BQP_J. We can lift BQP_J to a higher dimensional space as

$$\mathcal{C}_J = \left\{ (\mathbf{x}, \mathbf{y}) \in H_{n(n+1)/2} : (\mathbf{x}_J, \mathbf{y}_J) \in \text{BQP}_J \right\}.$$

Definition 4.2 *The polytope induced by a cover* C *of* $\{1, \dots, n\}$ *is defined as* $\mathcal{C}_C = \bigcap_{c \in C} \mathcal{C}_c$.

Lemma 4.1 *Any cover* C*-induced polytope is an outer approximation of the* BQP_n*, i.e.,* $\text{BQP}_n \subseteq \mathcal{C}_C$*. Moreover, the equality holds if* $\{1, \dots, n\} \in C$*.*

Proof: Let $(\overline{\mathbf{x}}, \overline{\mathbf{y}})$ be an arbitrary point in BQP_n. It is easy to verify that $(\overline{\mathbf{x}}, \overline{\mathbf{y}})$ satisfies constraints of $\mathcal{C}_c$, for all $c \in C$. Thus it holds that $(\overline{\mathbf{x}}, \overline{\mathbf{y}}) \in \mathcal{C}_C$ and therefore $\mathrm{BQP}_n \subseteq \mathcal{C}_C$. If $\{1, \ldots, n\} \in C$, it holds that $\mathcal{C}_C \subseteq \mathrm{BQP}_n$, which leads to $\mathrm{BQP}_n = \mathcal{C}_C$. ■

By Lemma 4.1, the convex underestimator of a bilinear function f can be constructed as

$$\breve{f}_C(\mathbf{x}) = \frac{1}{2} \min_{(\mathbf{x}, \tau_+(\mathbf{X})) \in \mathcal{C}_C} \langle \mathbf{Q}, \mathbf{X} \rangle. \tag{4.11}$$

Note that $\mathcal{C}_C$ can be explicitly described by

$$\mathcal{C}_C = \{(\mathbf{x}, \tau_+(\mathbf{X})) \in H_{n(n+1)/2} : \mathbf{X}_c = \sum_{l=1}^{2^{|c|}} \lambda_c^l \mathbf{V}_c^l, \mathbf{x}_c = \sum_{l=1}^{2^{|c|}} \lambda_c^l \mathbf{v}_c^l, \sum_{k=1}^{2^{|I|}} \lambda_c^k = 1, \lambda_c \geq 0,\ c \in C.\}$$

where $\mathbf{V}_c^l = \mathbf{v}_c^l (\mathbf{v}_I^l)^T$ and $\mathbf{v}_c^l$ is a vertex of $H_{|c|}$.

To improve the bound obtained from (4.11), one might consider a partition $\mathcal{P}$ as a cover of $\{1, \ldots, n\}$ and add RLT inequalities to bound $\mathbf{X}_{I,J}$ where $I, J \in \mathcal{P}$, i.e.,

$$\begin{aligned}
&\mathbf{X}_{I,J} = \sum_{k=1}^{2^{|I|}} \sum_{p=1}^{2^{|J|}} \gamma^{kp} \mathbf{V}_{I,J}^{kp}, \\
&\sum_{k=1}^{2^{|I|}} \gamma^{kp} = \lambda_J^p,\ p = 1, \ldots, 2^{|J|}, \\
&\sum_{p=1}^{2^{|J|}} \gamma^{kp} = \lambda_I^k,\ k = 1, \ldots, 2^{|I|}, \\
&\gamma^{kp} \geq 0,\ k = 1, \ldots, 2^{|I|}, p = 1, \ldots, 2^{|J|},
\end{aligned} \tag{4.12}$$

where $\mathbf{V}_{I,J}^{k,p} = \mathbf{v}_I^k (\mathbf{v}_J^p)^T, \forall k, p$.

Proposition 2 *For any $I, J \in \mathcal{P}$, $\mathrm{BQP}_{I \cup J}$ is characterized by*

$$S = \left\{(\mathbf{x}, \tau_+(\mathbf{X})) \in \mathbb{R}^{|I \cup J|} \times \mathbb{R}^{\binom{|I \cup J|}{2}} : (4.12), \mathcal{C}_I \cap \mathcal{C}_J\right\}$$

Proof: It holds that $\mathrm{BQP}_{I \cup J} \subseteq S$ since all the inequalities of S are valid inequalities to $\mathrm{BQP}_{I \cup J}$. So it suffices to show that the reverse also holds. Suppose $(\overline{\mathbf{x}}, \tau(\overline{\mathbf{X}})) \in S$. Then $(\overline{\mathbf{x}}_I, \overline{\mathbf{x}}_J)$ can be written as a finite convex combination of binary points of $H_{|I|+|J|}$ by imposing $\gamma^{kp} = \lambda_I^k \lambda_J^p$. Also notice that $\sum_{k=1}^{2^{|I|}} \sum_{j=1}^{2^{|J|}} \gamma^{kp} = 1$. Therefore $(\overline{\mathbf{x}}_{I \cup J}, \tau_+(\overline{\mathbf{X}}))$ can be written as a finite convex combination of vertices of $\mathrm{BQP}_{I \cup J}$ ending the proof. ■

Remark 4.2 *Alternative to adding constraint set (4.12) to formulation (4.11) associated with partition $\mathcal{P}$, one can also generate a new cover set whose members are all possible unions of two subsets of $\mathcal{P}$, thereby obtaining an equivalent formulation. One can check that both formulations have the same number of constraints and variables. For example if $\mathcal{P} = \{p_1, p_2, p_3\}$, the newly generated cover set can be $\{p_1 \cup p_2, p_1 \cup p_3, p_2 \cup p_3\}$.*

The concave overestimator $\widehat{f}_C(\mathbf{x})$ can also be obtained in the same way by maximizing the objective function. Also, note that one can make a trade-off between the quality of estimations and computational performance via the choice of the cover C.

In practice, one can surely combine this cover-based estimators with valid inequalities to strengthen the formulation. To avoid adding redundant cuts, we now study the strength of cover-based estimators by establishing connections between various valid inequalities and the choice of cover sets.

Proposition 3 *For* BQP_n*, various sets of inequalities in the same dimensional space of* BQP_n *can be identified by proper choices of cover set* C *of the index set* $\{1, \ldots, n\}$ *in the sense that the cover-induced polytope* $\mathcal{C}_C$ *defined before can be characterized by some inequalities. For example, McCormick inequalities [119, 6] are identified by a collection of all two subsets. Triangle inequalities [132] joint with McCormick inequalities are identified by three subsets. When the graph induced by* $Q_{ij}, i < j$ *has no* K_4 *minors, triangle inequalities and McCormick inequalities are equivalent to the characterization of* BQP_n *[132, 61].*

As mentioned before, we may choose a partition of the index set as a cover. Intuitively, a partition should be chosen by considering the coefficients of the bilinear function and the size of each subset. To balance these two criteria, we may consider minimizing normalized cut defined in [147]. Since it is NP-hard to solve the problem, one may consider the algorithm of [147] to solve it approximately. Alternatively, one can consider multiple random partitions of the index set at the same time. However, according to our numerical experiments, the benefits are not significant.

4.3.3 Extensions

In this section, we show that the convex (rep. concave) envelope of a bilinear function $f(\mathbf{x}) = \sum_{i<j} Q_{ij} x_i x_j$ is affinely equivalent to the concave (rep. convex) envelope of a polyhedral function. This connection implies some well-known results in the literature.

Lemma 4.2 *For each pair of* $(x_i, x_j) \in H_2$*, it holds that*

$$x_i x_j \leq \frac{1}{2}(x_i + x_j - |x_i - x_j|)$$

with equality at binary points.

Let us define the function below:

$$\phi : H_n \ni \mathbf{x} \mapsto \phi(\mathbf{x}) = \sum_{i<j} Q_{ij} |x_i - x_j|. \tag{4.13}$$

It is clear that function $\phi(\mathbf{x})$ is polyhedral. In addition, it enjoys the following property.

Lemma 4.3 *The convex hull of* $\{(\mathbf{x}, \phi(\mathbf{x})) : \mathbf{x} \in H_n\}$ *is solely determined by the set of points* $\{(\mathbf{x}, \phi(\mathbf{x})) : \mathbf{x} \in \{0,1\}^n\}$*, i.e.,*

$$\mathrm{conv}\{(\mathbf{x}, \phi(\mathbf{x})) : \mathbf{x} \in H_n\} = \mathrm{conv}\{(\mathbf{x}, \phi(\mathbf{x})) : \mathbf{x} \in \{0,1\}^n\}$$

Proof: By definition, $\mathrm{conv}\{(\mathbf{x}, \phi(\mathbf{x})) : x \in \{0,1\}^n\} \subseteq \mathrm{conv}\{(\mathbf{x}, \phi(\mathbf{x})) : x \in H_n\}$. Let $(\overline{\mathbf{x}}, \phi(\overline{\mathbf{x}}))$ be an extreme point of $\mathrm{conv}\{(\mathbf{x}, \phi(\mathbf{x})) : \mathbf{x} \in H_n\}$. Suppose that there exists

$\bar{x}_k \in (0,1)$ for some $1 \leq k \leq n$, and define $S_k = \{j : \bar{x}_j = \bar{x}_k, 1 \leq j \leq n\}$. Let ϵ be a sufficiently small quantity. Then, define x^+ by

$$x_j^+ = \begin{cases} \bar{x}_j & j \notin S_k \\ \bar{x}_j + \epsilon & j \in S_k \end{cases}, \ j = 1, \ldots, n.$$

Similarly, define x^- by

$$x_j^- = \begin{cases} \bar{x}_j & j \notin S_k \\ \bar{x}_j - \epsilon & j \in S_k \end{cases}, \ j = 1, \ldots, n,$$

and $\phi(\mathbf{x}^+)$, $\phi(\mathbf{x}^-)$ are

$$\phi(\mathbf{x}^+) = \phi(\overline{\mathbf{x}}) - \epsilon \sum_{\substack{i \in S_k, j \notin S_k: \\ x_i < x_j}} Q_{ij} + \epsilon \sum_{\substack{i \in S_k, j \notin S_k: \\ x_i > x_j}} Q_{ij}$$
$$\phi(\mathbf{x}^-) = \phi(\overline{\mathbf{x}}) + \epsilon \sum_{\substack{i \in S_k, j \notin S_k: \\ x_i < x_j}} Q_{ij} - \epsilon \sum_{\substack{i \in S_k, j \notin S_k: \\ x_i > x_j}} Q_{ij}.$$

This implies that $(\overline{\mathbf{x}}, \phi(\overline{\mathbf{x}})) = \frac{1}{2}(\mathbf{x}^+, \phi(\mathbf{x}^+)) + \frac{1}{2}(\mathbf{x}^-, \phi(\mathbf{x}^-))$, which contradicts the assumption that $(\overline{\mathbf{x}}, \phi(\overline{\mathbf{x}}))$ is an extreme point. Thus $\text{ext}\{\text{conv}\{(\mathbf{x}, \phi(\mathbf{x})) : \mathbf{x} \in H_n\}\} = \{(\mathbf{x}, \phi(\mathbf{x})) : \mathbf{x} \in \{0,1\}^n\}$. ■

Note that Lemma 4.3 holds regardless of the signs of $Q_{ij}, \forall i < j$. With above lemmas, we arrive at the following result.

Theorem 7 *Given a bilinear function $f(\mathbf{x}) = \sum_{i<j} Q_{ij} x_i x_j$, its convex (resp. concave) envelope is affinely equivalent to the concave (resp.convex) envelope of $\phi(\mathbf{x}) = \sum_{i<j} Q_{ij}|x_i - x_j|$, i.e.,*

$$\widehat{f}(\mathbf{x}) = \frac{1}{2}\sum_{i<j} Q_{ij}(x_i + x_j) - \frac{1}{2}\check{\phi}(\mathbf{x}), \quad \forall x \in H_n,$$
$$\check{f}(\mathbf{x}) = \frac{1}{2}\sum_{i<j} Q_{ij}(x_i + x_j) - \frac{1}{2}\widehat{\phi}(\mathbf{x}), \quad \forall x \in H_n.$$

Proof: It follows from Lemma 4.2 that

$$f(\mathbf{x}) = \frac{1}{2}\sum_{i<j} Q_{ij}(x_i + x_j) - \frac{1}{2}\phi(\mathbf{x}), \forall \mathbf{x} \in \{0,1\}^n.$$

As mentioned in Section 4.2, convex and concave envelopes of bilinear functions are vertex polyhedral. Lemma 4.3 show that envelopes ϕ are also vertex polyhedral. By the definition of (4.3), one can get the desired result. ■

Note that Theorem 7 indicates that we can approximate concave and convex envelops of f by approximating the convex and concave envelope of ϕ over H_n. Conversely, all techniques on approximating convex and concave envelopes of f also apply to function ϕ over a hypercube. In addition, this relation implies the following.

Corollary 4.3 *If $Q_{ij} \geq 0$, the concave envelope of a bilinear function f is characterized by McCormick over-estimator.*

Proof: If $Q_{ij} \geq 0$, $\phi(\mathbf{x})$ is convex. Then it holds that for any $x \in H_n$,

$$\begin{aligned}\widehat{f}(\mathbf{x}) &= \frac{1}{2}\sum_{i<j} Q_{ij}(x_i + x_j) - \frac{1}{2}\sum_{i<j} Q_{ij}|x_i - x_j| \\ &= \sum_{i<j} Q_{ij} \min\left\{x_i, x_j\right\},\end{aligned}$$

which completes the proof. ■

Note that this result has been proved independently in [116] and [123]. We may regard the above result as an alternative proof.

Corollary 4.4 *[119] When $n = 2$, McCormick inequalities characterize the convex and concave envelopes of a bilinear function $f(x)$ exactly.*

Proof: Let $n = 2$, for each $\mathbf{x} \in H_2$, it is easy to see that

$$\widehat{\phi}(\mathbf{x}) = \begin{cases} Q_{12}\min\{x_i + x_j, 2 - x_i - x_j\} & Q_{12} \geq 0, \\ Q_{12}\max\{x_i - x_j, x_j - x_i\}, & Q_{12} \leq 0. \end{cases}$$

and

$$\check{\phi}(\mathbf{x}) = \begin{cases} Q_{12}\max\{x_i - x_j, x_j - x_i\}, & Q_{12} \geq 0 \\ Q_{12}\min\{x_i + x_j, 2 - x_i - x_j\}, & Q_{12} \leq 0. \end{cases}$$

By Theorem 7, we have

$$\check{f}(\mathbf{x}) = \begin{cases} q_{12}\max\left\{0, x_1 + x_2 - 1\right\}, & q_{12} \geq 0, \\ q_{12}\min\left\{x_1, x_2\right\} & q_{12} \leq 0, \end{cases}$$

and

$$\widehat{f}(\mathbf{x}) = \begin{cases} q_{12}\min\left\{x_1, x_2\right\}, & q_{12} \geq 0, \\ q_{12}\max\left\{0, x_1 + x_2 - 1\right\} & q_{12} \leq 0. \end{cases}$$

which is exactly the McCormick's under (over-) estimator. ■

4.4 Numerical experiments

In this section, we compare the strength of the proposed SDP based estimators $\check{f}_{sdp}, \widehat{f}_{sdp}$ and cover based estimators $\check{f}_C, \widehat{f}_C$ by evaluating their values, as well as their gaps on a set of problem instances. $\check{f}_{sdp}$ is defined in consistent with (4.10), where constraints in $F(\mathbf{x})$ are

$$\begin{aligned} X_{ii} &\leq x_i,\ \ i = 1, \ldots, n, \\ X_{ij} &\geq \max\left\{x_i + x_j - 1, 0\right\},\ \ 1 \leq i < j \leq n \\ X_{ij} &\leq \min\left\{x_i, x_j\right\},\ \ 1 \leq i < j \leq n, \\ \mathcal{S}(\mathbf{x}, \mathbf{X}) &\succeq 0. \end{aligned}$$

Likewise, $\widehat{f}_{sdp}$ is defined symmetrically. The convex hull gap at a pre-defined point x is quantified as

$$\text{gap}(x) = \widehat{f}(\mathbf{x}) - \check{f}(\mathbf{x}),$$

where we respectively use $\widehat{f}_{sdp}(\mathbf{x}), \check{f}_{sdp}(\mathbf{x})$ and $\widehat{f}_C(\mathbf{x}), \check{f}_C(\mathbf{x})$ in place of $\widehat{f}(\mathbf{x}), \check{f}(\mathbf{x})$ for SDP based estimators and cover-based estimators. The instances are randomly generated following the rules below.

1. The dimension of the cube n belongs to $\{10, 20, 30\}$ and the density of the matrix $\mathbf{Q}$ denoted by ρ belong to $\{0.5, 0.75, 1\}$.

2. For nonnegative $\mathbf{Q}$, any nonzero component equals 1. For mixed signed entries of $\mathbf{Q}$, $\frac{1}{3}$ of the nonnegative components are -1 and the others are 1.

3. We randomly generate 500 test points within a hypercube with respect to different dimensions.

4. For the cover based estimators, we use a partition $\mathcal{P}$ as a cover. The partition of each considered index set $\{1, \dots, n\}$ has been done beforehand using `Shi-Malik` algorithm [147]. The size of partition is 4 for dimension 10, 7 for dimension 20, and 10 for dimension 30.

All formulations are formulated by YALMIP [115] and problem instances are solved by CPLEX 12.62 with default settings on a Mac with Inter Core i5 clocked at 2.7 GHz and with 8 GB of RAM.

Table 4.1 exhibits the associated approximated values for mixed coefficients cases. For each type of estimators, the evaluations of approximated convex and concave estimators are displayed as the averages of values of 500 test points. We report several observations below.

- Both SDP-based estimators and cover-based estimators perform significantly better than McCormick's inequalities.

- Values of both SDP-based estimators and cover-based estimators are close to values of the convex and concave envelopes. Even when the density is 1, the cover-based estimators can be close to values returned by the SDP-based estimators.

- Even though the strength of SDP-based estimators is generally stronger than that of cover-based estimators. There are cases that cover-based estimators dominate the SDP-based estimators, e.g., row $(10, 0.75)$. Of-course, the strength of cover-based estimators generally depends on the choice of the cover set.

Recall from Corollary 4.3 that the concave envelopes of bilinear functions over hypercubes are characterized by McCormick overestimators if all bilinear coefficients are nonnegative. Hence we omit values associated with concave overestimators in Table 4.2 and summarize observations below.

- The proposed cover-based convex estimators are evidently stronger than McCormick's estimators.

- If the matrix $\mathbf{Q}$ is not dense, e.g., $0.75, 0.5$, the cover-based convex estimators can be competitive with the SDP-based convex estimators.

- If the matrix $\mathbf{Q}$ is very dense, the SDP-based convex estimators exhibit distinct advantages over other estimators.

Table 4.1: An evaluation of convex and concave estimators (mixed entries)

	MC-estimators			SDP-estimators			cover-estimators			Envelopes		
(n, Density)	$\check{f}_{MC}$	$\widehat{f}_{MC}$	gap	$\check{f}_{SDP}$	$\widehat{f}_{SDP}$	gap	$\check{f}_{C}$	$\widehat{f}_{C}$	gap	$\check{f}$	$\widehat{f}$	gap
(10,0.5)	-0.06	3.76	3.83	0.20	3.40	3.20	0.20	3.43	3.23	0.39	3.21	2.82
(10,0.75)	0.08	5.77	5.69	0.86	5.05	4.19	0.97	5.00	4.03	1.08	4.82	3.74
(10, 1)	0.04	7.56	7.52	1.33	6.23	4.90	1.47	6.16	4.69	1.65	5.93	4.28
(20,0.5)	-0.18	15.96	16.15	2.26	13.75	11.49	2.13	14.04	11.91	3.50	13.40	9.90
(20,0.75)	-0.01	24.09	24.10	5.12	19.57	14.46	4.47	20.79	16.32	5.68	18.36	12.68
(20,1)	0.68	32.80	32.12	8.65	25.18	16.52	8.00	26.52	18.52	9.05	22.51	14.46
(30, 0.5)	-0.71	35.98	36.69	6.50	29.17	22.67	4.10	31.59	27.49	-	-	-
(30, 0.75)	-0.07	54.59	54.66	13.53	41.68	28.16	9.10	46.29	37.19	-	-	-
(30, 1)	-0.91	72.15	73.06	19.74	53.07	33.34	14.71	58.19	43.48	-	-	-

Table 4.2: An evaluation of convex estimators (nonnegative coefficients)

(n, Density)	$\check{f}_{MC}$	$\check{f}_{SDP}$	$\check{f}_{C}$	$\check{f}$
(10, 0.5)	3.79	4.21	4.30	4.33
(10, 0.75)	5.77	6.97	6.88	7.17
(10, 1)	7.65	10.55	9.18	10.63
(20, 0.5)	15.05	18.85	16.77	19.47
(20, 0.75)	23.27	30.87	26.63	31.71
(20, 1)	31.07	45.10	36.01	45.27
(30, 0.5)	35.22	44.88	42.34	-
(30, 0.75)	52.96	72.07	63.90	-
(30, 1)	70.75	90.22	79.75	-

4.5 A perspective

As a future research direction, we briefly discuss a novel approach to construct convex underestimator when all the coefficients of a bilinear function f are nonnegative. Recall the concept of *pole-set* introduced in [26].

Definition 4.3 *[26] Given a convex set S, a finite set Ω is called a pole-set of S if and only if any element $v \in S$ belongs to the convex hull of Ω (a convex combination of poles) denoted by* $\operatorname{conv}\Omega$.

Obviously, extreme points of S form a pole-set of itself. The extreme points of any simplex enclosing S also form a pole-set of S. The polyhedral function ϕ defined in (4.13) has a desired property below.

Lemma 4.4 *The concave envelope of $\phi(\mathbf{x})$ is vertex polyhedral over arbitrary polytopes if $Q_{ij} \geq 0,\ \forall i < j$.*

Now assume that a pole-set Ω of a hypercube H_n is available, we are then able to construct a convex underestimator of f via constructing a concave overestimator of ϕ over a hypercube.

Theorem 8 *Let Ω be a pole-set of a hypercube H_n, then if $Q_{ij} \geq 0, \forall i < j$, for each $\mathbf{x} \in H_n$,*

$$\breve{f}(\mathbf{x}) \geq \frac{1}{2}\sum_{i<j} Q_{ij}(x_i + x_j) - \frac{1}{2}\widehat{\phi}_\Omega(\mathbf{x}), \tag{4.14}$$

where $\widehat{\phi}_\Omega(\mathbf{x}) = \max\left\{\sum_{\omega\in\Omega} \lambda_\omega \phi(\omega) : \lambda_i \geq 0, \sum_{\omega\in\Omega} \lambda_\omega = 1, \sum_{\omega\in\Omega} \lambda_\omega \omega = \mathbf{x}\right\}$.

Proof: By definition, it follows that

$$\widehat{\phi}_{H_n}(\mathbf{x}) \leq \widehat{\phi}_\Omega(\mathbf{x}).$$

Since $Q_{ij} \geq 0,\ \forall i < j$, Lemma 4.4 implies

$$\widehat{\phi}_\Omega(\mathbf{x}) = \max\left\{\sum_{\omega\in\Omega} \lambda_\omega \phi(\omega) : \lambda_i \geq 0, \sum_{\omega\in\Omega} \lambda_\omega = 1, \sum_{\omega\in\Omega} \lambda_\omega \omega = \mathbf{x}\right\}.$$

It completes the proof in conjunction with Theorem 7. ■

One can see that when the pole-set Ω is the set of extreme points of H_n, the concave envelope of $\phi(\mathbf{x})$ is obtained and therefore we get the exact convex envelope of $f(\mathbf{x})$. Of-course, the quality of the approximation relies on the choice of pole-set Ω. An investigation on the construction of pole-sets has been carried out in [26]. We illustrate here the method with an example.

Example 1 *Let us approximate the convex envelope values of a bilinear function $f(\mathbf{x})$ with $Q_{ij} = 1, 1 \leq i < j \leq 5$ over a 5-dimensional cube. In general, we need to use 32 extreme points to get the optimality certificate. Let us take a pole-set*

$$\Omega = \left\{(1, \mathbf{x}) : x \in \{0,1\}^4\right\} \bigcup \left\{(0, x) : x \in \operatorname{ext}\{\mathbf{x} \in \mathbb{R}^4 : \|x - \mathbf{0.5}\|_1 \leq 2\}\right\},$$

where ext *represent the set of extreme points of a set. It is clear that the pole-set has 24 points, where 16 of them come from extreme points of a 4-dimensional cube and 8 of them are members of a norm-1 ball centered at **0.5** with radius 2.*

For point $\mathbf{x} = (0.5, 0.5, 0.5, 0.5, 0.5)$, the exact value of convex envelope is $\breve{f}(\mathbf{x}) = 2$, the value of McCormick relaxation gives $\breve{f}_{Mc}(\mathbf{x}) = 0$ and our pole-set based convex estimator returns $\breve{f}_\Omega(\mathbf{x}) = 1.0625$.

4.6 Conclusion

In this chapter, we have discussed two different ways to approximate the convex and concave envelopes of a general bilinear function over hypercubes. The first approach aims at finding the best estimators of the bilinear function in a convex quadratic form at a pre-specified point, which turns out to be a semidefinite program parameterized at the point. We then established a connection between quadratic convexification based techniques and the standard semidefinite relaxation for bilinear optimization problems. We also demonstrated that two seemingly different formulations are in fact equivalent by establishing a geometric connection between their feasible regions. The second approach focused on approximating the boolean quadratic polytope by adding fully-described BQPs of lower dimensions.

In addition, some pole-set based estimators have been proposed as a perspective. Further studies are needed to find a reasonable way to define the pole-sets.

Chapter 5

Multipolar Robust Optimization

5.1 Summary

We consider linear programs involving uncertain parameters and propose a new tractable robust counterpart which contains and generalizes several other models including the existing *Affinely Adjustable Robust Counterpart* and the *Fully Adjustable Robust Counterpart*. It consists in selecting a set of *poles* whose convex hull contains some projection of the uncertainty set, and computing a recourse strategy for each data scenario as a convex combination of some optimized recourses (one for each pole). We show that the proposed *multipolar robust counterpart* is tractable and its complexity is controllable. Further, we show that under some mild assumptions, two sequences of upper and lower bounds converge to the optimal value of the fully adjustable robust counterpart. To illustrate the approach, a robust problem related to lobbying under some uncertain opinions of authorities is studied. Several numerical experiments are carried out showing the advantages of the proposed robustness framework and evaluating *the benefit of adaptability*.

5.2 Introduction

Uncertainty in optimization parameters arises in many applications due to the difficulty to measure data or because of their variability. To deal with uncertainty, there are mainly two approaches: stochastic optimization and robust optimization. In the first case, some probabilistic assumptions are made about the uncertain data [41, 60, 90]. One is then interested in computing a solution optimizing some moments of random variables depending on the data. Another variant, known as chance constrained programming [55], consists in imposing that some constraints are satisfied only with some probability.

Robust optimization is a more recent approach dealing with uncertainty. It does not require specifications of the exact distribution of the problem's parameters. Roughly speaking, uncertain data are assumed to belong to a known compact set, called uncertainty set, and we aim at finding a solution that is *immunized* against all possible realizations in the uncertainty set. An early contribution related to robust optimization is the work of Soyster [149] followed by intensive investigations in the last 20 years starting with [31, 67] in the context of convex optimization and the book [98] dealing with discrete optimization. Almost at the same time, and in an independent way, a lot of work was initiated in [69] and [64] on robust optimization in communication networks dealing with uncertain traffic matrix, see [27] for a survey.

Robust optimization and stochastic programming are related in numerous ways. For example, using some knowledge about the distribution of uncertain data, it is sometimes possible to define an uncertainty set in such a way that the robust solution is an approximated solution of a chance constrained problem (see, e.g., [29, 32, 97] for details and references). An approach combining robust optimization and stochastic programming consists in computing solutions that are distributionally robust where the distribution of parameters is assumed to vary within some set (for example, when the mean and the covariance matrix are known) (see, e.g., [73, 75]).

The definition of the uncertainty set is a critical issue since a bad choice might lead to very expensive solutions. One way to alleviate overconservatism of the robust approach is to assume that a subset of the decision variables are adjustable on the realization of the uncertain data. Let us for example consider the following linear problem

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \le \mathbf{b}, \\ & \mathbf{x} \in \mathbb{R}^n, \end{aligned}$$

involving uncertain parameters. We assume that $\mathbf{x}$ is partitioned as $\mathbf{x} = (\mathbf{u}, \mathbf{v})$, where $\mathbf{u}$ represents the non-adjustable and $\mathbf{v}$ the adjustable variables. The robust counterpart of this uncertain problem under consideration reads

$$\begin{aligned} \min_{u,v} \quad & \mathbf{c}^T\mathbf{u} \\ \text{s.t.} \quad & \mathbf{U}\mathbf{u} + \mathbf{V}\mathbf{v}(\xi) \le \mathbf{b}, \;\; [\mathbf{U}, \mathbf{b}] \in \Xi, \end{aligned} \tag{FARC}$$

where the uncertain parameters are $\mathbf{U} \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$, while $\mathbf{V}$ and $\mathbf{c}$ are assumed to be known. We denote by $\xi \equiv [\mathbf{U}, \mathbf{b}] \in \Xi$ the uncertain parameters belonging to the uncertainty set Ξ assumed to be ***compact, convex and with a non-empty interior***. ξ will be considered as a vector in the rest of this chapter.

The non-adjustable variables are sometimes interpreted "here and now" variables, while the adjustable ones can be seen as "wait and see" variables. This robust counterpart above is generally called *fully-adjustable robust counterpart* (FARC). FARC is sometimes called the dynamic robust counterpart since $\mathbf{v}$ depends on ξ. FARC can be seen as a two-stage optimization problem where $\mathbf{u}$ are the first-stage variables and $\mathbf{v}$ are the second-stage variables.

If variables $\mathbf{v}$ are also static, then FARC simply becomes the standard *static robust counterpart* denoted by SRC. Cases where FARC and SRC are equivalent have been pointed out in [30] where it is shown that adaptability does not lead to any improvement in the constraint-wise uncertainty case. Still, FARC is generally much less conservative than SRC. In other words, there is generally some *benefit of adaptability*. Solving FARC is, unfortunately difficult in general cases as shown by many authors [30, 56, 124]. Another concern related to FARC is the inherent difficulty of implementing the solution $\mathbf{v}(\xi)$ in a practical way.

To get a tractable optimization problem and also to alleviate some overconservatism of SRC, an affinely adjustable approach was proposed in [30], where the adjustable variables $\mathbf{v}$ are not fully adaptable (dynamic), but are assumed to depend on the uncertain data ξ in an affine way:

$$\mathbf{v}(\xi) = \mathbf{w} + \mathbf{W}\xi, \quad \xi \equiv [\mathbf{U}, \mathbf{b}] \in \Xi,$$

where $\mathbf{w}$ and the elements of matrix $\mathbf{W}$ are new decision variables (a.k.a. affine decision rules). The induced formulation is called *affine-adjustable robust counterpart* (AARC). An affine approach was also independently proposed for network optimization problems where the traffic matrix is supposed to be uncertain and the way how traffic is splitted through network's paths is optimized [94, 24, 131]. Further developments appeared in [14, 130].

Applying affine decision rules naturally leads to less expensive solutions than those obtained by the static approach. The performance gap quantified by the difference between optimum of AARC and the optimum of FARC was discussed for robust linear problems with right-hand-side uncertainty in [34, 36]. One of the results of [34] states that AARC is equivalent to FARC when the uncertainty set is a simplex. Some tight approximation bounds relating the optimum of AARC to that of FARC in the right-hand-side uncertainty case are also given there.

Related investigation on problems with some special uncertainty sets (integer sublattices of the unit hypercube) are discussed in [87], where they provide sufficient conditions such that the associated affinely adjustable decision rules lead to exact optimum of FARC.

The affine approach is related to the well-known linear or first-order decision rules used in the context of multi-stage stochastic optimization [71]. Linear decisions rules were also used in [99] in the context of stochastic programming not only to get upper bounds (as done above) but also to get lower bounds by properly approximating the dual problem using linear decision rule.

As observed in [58], even though AARC has been successfully applied to several problems, its performance might be unsatisfactory under situations where the adjustable variables exhibit high nonlinearity in terms of the uncertain parameters. This led to some extensions of the affine approach in [57, 58] (see also references therein) by reparametrizing the uncertainties and then applying the affinely adjustable approach. Roughly speaking, a new set of variables is introduced (for example the positive and the negative parts of the original uncertainties), and the adjustable variables are assumed to affinely depend on the new set of parameters. A similar idea is also proposed in [36] in the context of one-dimensional constrained multistage robust optimization.

Other extensions of affine decision rules have been proposed in literature. In [37] polynomial recourse actions are considered where $\mathbf{v}$ is expressed as a polynomial in uncertainty parameters with degree no larger than a fixed constant. The complexity of the robust counterpart problem is then related to testing the positivity of a polynomial. Using some recent results in algebraic geometry stating that under mild conditions, a positive polynomial can be expressed as a sum of squares (not a priori bounded), the robust counterpart is approximated by considering sums of squares of degree no larger than a fixed constant. As a sum of squares can be represented by a semidefinite programming [101], the proposed robust counterpart can be efficiently handled [37].

Another robust approach dealing with uncertainty, termed as *multi-static* approach in [27], was proposed and studied in [23, 28, 156]. It consists in partitioning the uncertainty set Ξ into a finite number of subsets $\Xi_1, \ldots, \Xi_p$ and using a recourse action $\mathbf{v}_i$ for each subset Ξ_i. In other words, if $\xi \in \Xi_i$, then we take $\mathbf{v} = \mathbf{v}_i$. The recourse actions $\mathbf{v}_i$ are of course subject to optimization. A quite close idea is proposed in [33], where it was called *finite adaptability*. The performance of finite adaptability in a fairly general class of multi-stage stochastic and adaptive optimization problems was investigated in [35].

One can also combine finite adaptability and the affinely-adjustable approach by partitioning the uncertainty set into some subsets and considering some optimized specific affine decision rules for each subset. This was also considered in [23, 27] in the context of network design problems. This type of adaptability might also be called *piecewise-affine adaptability*. Piecewise-affine rules were also considered in several other papers such as [22, 75].

We mention that dynamic programming approaches have been successfully designed to solve certain robust optimization problems. For example, authors in [97] proposed a dynamic programming approach to solve the chance-constrained robust kapsack problem iteratively. Very recently, Agra et al. [5] decomposed a general robust optimization model whose dynamic programming value function is convex and separable, to a master problem and a separation problem. To solve the separation problem, they proposed a couple of dynamic programming approaches, which allows a FPTAS.

While a great number of proposals in robust optimization have appeared, there are still challenges. First, to the best of our knowledges, none are general enough to encompass static robustness, affinely adjustable robustness and fully adjustable robustness. Second, as observed in [37], there is no systematic way to influence the trade-off between the performance of the resulting policies and the computational complexity required to obtain them. Third, the uncertain parameters of an optimization problem can be sometimes difficult to observe. In several applications, only a subset of such parameters or some aggregates of them can be observed.

The objective of our work is to provide a framework addressing those challenges at the same time. Our contributions are four-fold:

1. A novel approach. We propose a hierarchical and convergent framework of adjustable robust optimization – *multipolar robust approach*, which generalizes notions of static robustness, affinely adjustable robustness, fully adjustable robustness and fill the gaps in-between. As a byproduct, a new way to look at the affine adaptability is proposed. The result of [34] stating that affine rules are optimal when the uncertainty set is a simplex is also obtained as a consequence of the multipolar approach.

2. A comprehensive analysis. We show that the *multipolar robust counterpart* is tractable by either a cut generation procedure or a compact formulation. Further, we prove that the multipolar approach can generate a sequence of upper bounds

and a sequence of lower bounds at the same time and both sequences converge to the robust value of FARC under some mild assumptions.

3. A general constructive algorithm of *pole-sets*. The multipolar approach is based on some tools related to the uncertainty set, that we term as pole-sets. For their construction, we start with a simplex and then compute the best homothetic transformation of this simplex to allow it to enclose a given convex set. An efficient algorithm is proposed to compute such homothetic set. As a byproduct, we provide a very simple proof of the geometric results of [128] related to hypercubes. The pole-sets obtained after this homothetic transformation are then improved using a tightening procedure.

4. An application. To numerically illustrate the multipolar approach, a lobbying problem is considered where a lobby aims to minimize the budget needed to convince a set of voters taking into account a reasonable opinion dynamics model under some uncertainty. The benefit of adaptability is clearly shown for this problem.

Outline. In Section 5.3, we present the concept and ingredients of multipolar robust optimization and show that static robustness, affinely adjustable robustness, fully adjustable robustness are special cases of multipolar robust framework. In Section 5.4, we discuss the tractability, the monotonicity and the convergence of the proposed approach. A simple illustrative example is described in Section 5.5. In Section 5.6, we propose algorithms for pole-set generation. Section 5.7 is dedicated to a numerical example on a lobbying problem under several uncertainty scenarios. Finally, concluding remarks follow in Section 5.8.

Notation. Throughout this chapter, we use Ξ to represent a compact convex uncertainty set and ξ to denote a member of Ξ. We use $\mathbf{I}$ to denote the identity matrix. Vectors and matrices are marked in bold, and their scalar components are presented in italic. Given any matrix $\mathbf{C}$, $\mathbf{C}^T$ denotes its transpose. We also use $[\mathbf{C}, \mathbf{D}]$ to denote the matrix where $\mathbf{C}$ and $\mathbf{D}$ are concatenated by columns assuming they have the same number of rows. Similarly, $(\mathbf{C}, \mathbf{D})$ denotes the matrix obtained by row concatenation of two matrices $\mathbf{C}$ and $\mathbf{D}$ when they have the same number of columns. Observe that $\mathbf{v} = (v_1, \ldots, v_n)$ is then a vector and $[v_1, \ldots, v_n] = \mathbf{v}^T$. We use δ_{ij} to represent the Kronecker's delta function, where $\delta_{ij} = 1$ if $i = j$, 0 otherwise. For a set $S \in \mathbb{R}^n$, we use $\text{ext}(S)$ to represent the set of its extreme points, $\text{conv}\, S$ to represent its convex hull and $\dim(S)$ to denote its dimension. If S is finite, we use $|S|$ to represent its cardinality. We also use the standard notation for usual norms: $\|\cdot\|_\infty$ for the infinity norm, $\|\cdot\|_1$ for the Manhattan norm and $\|\cdot\|_2$ for the Euclidean norm.

5.3 The multipolar robust optimization concept

In this section, we introduce the main ingredients of multipolar robustness and then setup the multipolar robust counterpart as a novel approximation of FARC.

Shadow matrix. Like other robust approaches, multipolar approach is also based on an uncertainty set Ξ. In addition, we consider a matrix associated with certain operations on the uncertain information, which can be data aggregation, filtering, and selection. Note that these operations can either be natural or artificial. Natural operations are induced by the difficulty of measurements or shortage of data. For example, in communication networks, traffic flows are usually observed in an aggregated manner (the consequence of aggregating uncertain demands from multiple origin-destination pairs). Nevertheless, adjustable recourse actions should be implemented based the observed partial information. On the other hand, artificial operations can be certain techniques to control the complexity of the multipolar robust counterpart, as explained in the concluding remarks of this section. We call the associated matrix of an operation *shadow matrix* since the operation either reduces the size of the multipolar robust counterpart or is a direct consequence of observations. We use $\mathbf{P} \in \mathbb{R}^{n_0 \times \dim(\Xi)}$ to denote a shadow matrix, where n_0 is the dimension of the shadow (i.e., the resulting partial information) and $\dim(\Xi)$ is the dimension of the uncertainty set Ξ. The resulting partial information is defined by

$$\Xi_P := \mathbf{P}\Xi \equiv \{\mathbf{P}\xi, \xi \in \Xi\}. \tag{5.1}$$

When $\mathbf{P}$ is identity matrix, we have a complete measure of uncertainty.

We will assume that $\mathbf{P}$ is full row rank matrix. Consequently, Ξ_P is also compact, convex and has a non-empty interior.

Pole-set. A key component of the multipolar approach is a finite set of *poles*, which are given vectors in the range space of the shadow matrix. We denote by Ω such a *pole-set.* We say that Ω is a pole-set of Ξ_P iff for any $\xi \in \Xi$, $\mathbf{P}\xi$ belongs to the convex hull of Ω (a convex combination of poles) denoted by $\operatorname{conv}\Omega$. Given a set Ξ_P, a collection of pole-sets of Ξ_P is defined as

$$\mathcal{F}_{\Xi_P} := \{\Omega : \Xi_P \subseteq \operatorname{conv}\Omega\}. \tag{5.2}$$

Obviously, extreme points of Ξ_P form a pole-set, i.e., $\operatorname{ext}(\Xi_P) \in \mathcal{F}_{\Xi_P}$.

Multipolar robust counterpart. We now setup the multipolar robust counterpart w.r.t. an uncertainty set Ξ, a shadow matrix $\mathbf{P}$, a pole-set $\Omega \in \mathcal{F}_{\Xi_P}$. For each $\xi \in \Xi$, we consider a weight λ_ω^ξ for each pole ω in Ω. Then, for each scenario $\xi \in \Xi$, the following system has a solution

$$\begin{aligned}
&\sum_{\omega\in\Omega} \lambda_\omega^\xi \omega = \mathbf{P}\xi,\\
&\sum_{\omega\in\Omega} \lambda_\omega^\xi = 1, \qquad (5.3)\\
&\lambda_\omega^\xi \geq 0, \;\; \omega \in \Omega.
\end{aligned}$$

Let Λ_ξ be the set of weight vectors $\boldsymbol{\lambda}^\xi$ satisfying the above system for a given $\xi \in \Xi$. In the considered paradigm, each pole is associated with a recourse action, and the recourse action in the presence of $\xi \in \Xi$ is approximated by a convex combination of the recourse actions associated with the poles. Specifically, let vector $\mathbf{v}_\omega$ be the recourse action associated with pole ω in the above system. We require the adjustable variables $\mathbf{v}(\xi)$ to be restricted to

$$\mathbf{v}(\xi) = \sum_{\omega\in\Omega} \lambda_\omega^\xi \mathbf{v}_\omega, \tag{5.4}$$

where $\boldsymbol{\lambda}^\xi \in \Lambda_\xi$. We can readily present the *multipolar robust counterpart* defined by

$$\Pi_\Xi(\mathbf{P},\Omega) = \min_{\mathbf{u},\mathbf{v}} \quad \mathbf{c}^T\mathbf{u} \tag{MRC}$$

$$\text{s.t.} \quad \mathbf{U}\mathbf{u} + \mathbf{V}\sum_{\omega\in\Omega} \lambda_\omega^\xi \mathbf{v}_\omega \leq \mathbf{b},\ \xi \in \Xi,\ \boldsymbol{\lambda}^\xi \in \Lambda_\xi. \tag{5.5}$$

Following the spirits of robust optimization, the multipolar robust counterpart (MRC) seeks a pair of non-adjustable solution $\mathbf{u}$ and a set of recourse actions related to poles $\mathbf{v}_\omega, \omega \in \Omega$ such that the objective function is minimized while hedging against the uncertainty set Ξ. In brief, given Ξ, the multipolar robust approach can bee seen as a set function of a pole-set Ω and a shadow matrix $\mathbf{P}$. We denote the function by

$$\Pi_\Xi : \mathbb{R}^{n_0\times\dim(\Xi)} \times \mathcal{F}_{\Xi_P} \ni (\mathbf{P},\Omega) \mapsto \Pi_\Xi(\mathbf{P},\Omega) \in \mathbb{R}$$

and call $\Pi_\Xi(\mathbf{P},\Omega)$ *multipolar robust value* w.r.t. $(\mathbf{P},\Omega)$. Also, we call $(\mathbf{u},\mathbf{v})$ *multipolar solution*.

To conclude this section, we add few remarks on the concept of shadow matrix and pole-set to clarify the motivation behind these ingredients.

- Note that by (5.5), the solution is protected against the considered uncertainty Ξ. Neither a shadow matrix $\mathbf{P}$ nor a pole-set Ω changes the uncertainty set, so $\mathbf{P}$ and Ω are not used to approximate the uncertainty set.

- Observe that $1 + \dim(\Xi_P) \leq |\Omega|$, so we can use the shadow matrix $\mathbf{P}$ to reduce the number of recourse actions and therefore the number of variables of (5.5). Reducing the number of poles leads to an MRC which is easier to solve as will be shown in Section 5.4.

- In several applications, after data is revealed, the adjustable variables should be quickly chosen and used. This is fortunately easy to do in the multipolar robust framework since the only thing to do is to find the coefficients λ_ω^ξ and use them to combine the already computed recourse vectors $\mathbf{v}_\omega, \omega \in \Omega$.

5.3.1 Special cases

We show in this section that MRC generalizes SRC, AARC, and FARC by different settings of pole-sets and recourse actions associated with poles.

First, we show that SRC is a special case of MRC. Imposing $\mathbf{v}_\omega = \mathbf{v}_{\omega'}$ for any pair of ω and ω' belonging to Ω leads to $\mathbf{v}(\xi) = \mathbf{v}_\omega, \forall\xi \in \Xi$, which means that the recourse action is static. Another way to get SRC is to impose that $\mathbf{P}$ is a null matrix having one row (relaxing in this case the full row rank constraint related to P) and Ω contains just the zero vector.

Second, we show that FARC is a special case of MRC. Let Ω be the set of extreme points of Ξ and $\mathbf{P} = \mathbf{I}$. Then $\operatorname{conv}\Omega = \Xi$, that is for $\xi \in \Xi$, there exists $\boldsymbol{\lambda}^\xi \geq \mathbf{0}$ such that $\sum_{\omega\in\Omega} \lambda_\omega^\xi = 1$ and $\sum_{\omega\in\Omega} \lambda_\omega^\xi \omega = \xi$. By linearity of inequalities (5.5), imposing that $\mathbf{U}\mathbf{u} + \mathbf{V}\mathbf{v}_\omega \leq \mathbf{b}$ for each extreme point $\omega \in \Omega$ is necessary and sufficient to ensure the satisfaction of all inequalities (5.5) for each $\xi \in \Xi$. We get here the fully adjustable case representing the best that we can obtain for this problem since it is equivalent to assuming that $\mathbf{v}$ can vary with no restrictions. Note that if the number of extreme points of Ξ is limited, then the robust optimization counterpart can be efficiently solved. However, if the number of extreme points of Ξ is non-polynomial, the problem is generally difficult (as already mentioned in Section 5.2, see for example [30, 56, 124]).

Third, we show that AARC can also be generalized by MRC by proving the following theorem.

Theorem 9 *Let $\Omega \in \mathcal{F}_{\Xi_P}$ such that $|\Omega| = 1 + \dim(\mathbf{P}\Xi)$. Then the optimal solution of the corresponding MRC problem is exactly the best solution that is affine in $\mathbf{P}\xi$.*

Proof: Since $\mathbf{P}\Xi$ has non-empty interior, $\mathbf{P}\Xi \subset \operatorname{conv}\Omega$ and $|\Omega| = 1 + \dim(\mathbf{P}\Xi)$, the elements of Ω are affinely independent. Let $d = \dim(\mathbf{P}\Xi)$ and assume $\Omega = \{\omega^{(1)}, \dots, \omega^{(d+1)}\}$. The shadow matrix $\mathbf{P}$ is here the identity matrix. Consider matrix D obtained by taking vectors $\omega^{(i)}$ as columns and adding a final line containing only coefficients equal to 1.

$$\mathbf{D} = \begin{pmatrix} \omega_1^{(1)} & \dots & \omega_1^{(d+1)} \\ \vdots & \vdots & \vdots \\ \omega_d^{(1)} & \dots & \omega_d^{(n+1)} \\ 1 & \dots & 1 \end{pmatrix}.$$

Observe that $\mathbf{D}$ is a non-singular square matrix of size $(d+1)$.

Given any ξ, there are unique coefficients λ_ω^ξ such that $\mathbf{P}\xi = \sum_{\omega\in\Omega} \lambda_\omega^\xi \omega$ and $\sum_{\omega\in\Omega} \lambda_\omega^\xi = 1$. This can be written as $\begin{pmatrix} \mathbf{P}\xi, 1 \end{pmatrix} = \mathbf{D}\boldsymbol{\lambda}^\xi$ where $\boldsymbol{\lambda}^\xi$ is the vector whose components are the λ_ω^ξ for $\omega \in \Omega$. This immediately implies that $\boldsymbol{\lambda}^\xi = \mathbf{D}^{-1}\begin{pmatrix} \mathbf{P}\xi, 1 \end{pmatrix}$. Using $\mathbf{E}$ to denote the matrix whose columns are the recourse vectors $\mathbf{v}_\omega$, equation (5.4) becomes $\mathbf{v} = \mathbf{E}\mathbf{D}^{-1}\begin{pmatrix} \mathbf{P}\xi, 1 \end{pmatrix}$. This clearly implies that $\mathbf{v}$ affinely depends on $\mathbf{P}\xi$.

Let us now consider any affine policy $\mathbf{w} + \mathbf{W}\mathbf{P}\xi$. As shown above, the recourse vector $\mathbf{v}$ provided by the multipolar approach is given by $\mathbf{E}\mathbf{D}^{-1}\begin{pmatrix} \mathbf{P}\xi, 1 \end{pmatrix}$. By taking $\mathbf{E} = [\mathbf{W}, \mathbf{w}]\mathbf{D}$, we get $\mathbf{v} = \mathbf{w} + \mathbf{W}\mathbf{P}\xi$. In other words, any recourse policy that is affine in $\mathbf{P}\xi$ can be obtained through the multipolar approach. ■

When $\mathbf{P} = \mathbf{I}$, we get the desired result below.

Corollary 5.1 *The affinely adjustable approach is a special case of the multipolar approach. It corresponds to any set of $(\dim\Xi + 1)$ affinely independent poles, in multipolar robust optimization when $\mathbf{P} = \mathbf{I}$.*

The following corollary is also immediate.

Corollary 5.2 *If the uncertainty set Ξ is a simplex, then the affinely adjustable robust counterpart is equivalent to the fully adjustable robust counterpart in the sense that their objective values are equal.*

Proof: Taking all the vertices of the simplex uncertainty set as the set of poles in multipolar robust approach leads to the optimum of FARC. By Corollary 5.1, this pole-set corresponds to affine adjustable approach, which completes the proof. ■

Corollary 5.2 has been presented in [34] in the special case of right-hand-side uncertainty, so we may treat the result here as an alternative proof using the framework of multipolar approach.

5.4 Analysis

In this section, we first analyze the tractability of the multipolar robust counterpart MRC. Then, we show that the proposed framework can generate a monotonic sequence converging to the fully adjustable robust value of FARC. In fact, we will simultaneously generate a lower and an upper bound both converging to the optimal value of FARC under some mild assumptions.

5.4.1 Tractability

In this section, we show that MRC is computationally tractable. It can be solved either by cut generation or using a compact reformulation.

First, a cutting plane algorithm for solving MRC may be devised as follows. Assume that $|\Omega|$ is finite and has a reasonable size. Given a solution $(\mathbf{u}, \mathbf{v})$, we have to check if there exists a pair of $\xi \in \Xi$ and $\boldsymbol{\lambda}^\xi \in \Lambda_\xi$ violating the constraints of MRC. This can be done by checking the sign of the optimum of each i^{th} problem

$$\begin{aligned}
\max_{\boldsymbol{\lambda},\xi} \quad & \mathbf{U}_i\mathbf{u} + \mathbf{V}_i \sum_{\omega\in\Omega} \lambda^\xi_\omega \mathbf{v}_\omega - b_i && (5.6a)\\
\text{s.t.} \quad & \sum_{\omega\in\Omega} \lambda^\xi_\omega \omega = \mathbf{P}\xi, && (5.6b)\\
& \sum_{\omega\in\Omega} \lambda^\xi_\omega = 1, && (5.6c)\\
& \lambda^\xi_\omega \geq 0, \quad \omega \in \Omega, && (5.6d)\\
& \xi \in \Xi, && (5.6e)
\end{aligned}$$

where $\mathbf{U}_i$ and $\mathbf{V}_i$ are the i^{th} rows of $\mathbf{U}$ and $\mathbf{V}$. If it is positive, then constraint

$$\mathbf{U}_i\mathbf{u} + \mathbf{V}_i \sum_{\omega\in\Omega} \hat{\lambda}^{\hat{\xi}}_\omega \mathbf{v}_\omega \leq b_i, \tag{5.7}$$

needs to be added to the restricted problem, where $(\hat{\lambda}^{\hat{\xi}},\ \hat{\xi})$ solves (5.6). Problem (5.6) can generally be solved easily when Ξ is polyhedral or ellipsoidal. In these cases, by equivalence of separation and optimization [80], the multipolar robust optimization counterpart problem can also be solved in polynomial time if the number of poles $|\Omega|$ is polynomially bounded.

Second, we may solve MRC by duality. It is sometimes possible for several kinds of convex uncertainty sets to write a strong dual of (5.6) leading to an extended reformulation of MRC. This holds for example if Ξ is a polytope defined by a limited number of constraints, i.e., $\Xi := \{\xi \equiv [\mathbf{U}, \mathbf{b}] : \mathbf{C}\xi \leq \mathbf{d}\}$, where $\mathbf{C} = [\mathbf{C}_1, \ldots, \mathbf{C}_m]$, $\mathbf{C}_i \in \mathbb{R}^{n_d\times(n+1)}, d \in \mathbb{R}^{n_d}$ and ξ is expressed as a column vector of size $(n+1)\times m$. ξ contains m blocks of size $n+1$ vectors: the i^{th} block contains $\mathbf{U}_i^T$ followed by b_i.

By strong duality, the constraints of the multipolar robust counterpart MRC w.r.t. Ξ can be replaced with a polynomial number of inequalities. For each i, the inequalities $\mathbf{U}_i\mathbf{u} + \mathbf{V}_i \sum\limits_{\omega\in\Omega} \lambda_\omega^\xi \mathbf{v}_\omega \leq b_i, \xi \in \Xi$, $\boldsymbol{\lambda}^\xi \in \Lambda_\xi$ are replaced with

$$\begin{aligned} &\mathbf{d}^T\boldsymbol{\eta}_i + \mathbf{V}_i\mathbf{v}_\omega - \omega^T\boldsymbol{\sigma}_i \leq 0,\ \ \omega \in \Omega, \\ &\mathbf{C}_j^T\eta_i - \mathbf{P}_j^T\boldsymbol{\sigma}_i = \delta_{ij}\left(\mathbf{u}, -1\right),\ \ j = 1, \dots, m, \\ &\boldsymbol{\eta}_i \in \mathbb{R}_+^{n_d},\ \ \boldsymbol{\sigma}_i \in \mathbb{R}^{n_0}, \end{aligned} \tag{5.8}$$

where the shadow matrix $\mathbf{P} = [\mathbf{P}_1, \dots, \mathbf{P}_j, \dots, \mathbf{P}_m], \mathbf{P}_j \in \mathbb{R}^{n_0\times(n+1)}$, $j = 1, \dots, m$, δ_{ij} is Kronecker's delta function.

When Ξ is ellipsoidal, i.e., $\Xi := \{\xi : \|\mathbf{F}\xi\|_2 \leq 1\}$, the multipolar robust counterpart can be represented by a second order cone program. Then for each i, the i^{th} constraint of MRC is replaced with

$$\begin{aligned} \|\boldsymbol{\eta}_i\|_2 + \mathbf{V}_i\mathbf{v}_\omega - \omega^T\boldsymbol{\sigma}_i \leq 0, \\ \mathbf{F}^T\eta_i - \mathbf{L}_i = \mathbf{0}, \\ \boldsymbol{\eta}_i \in \mathbb{R}^{n_q},\ \ \boldsymbol{\sigma}_i \in \mathbb{R}^{n_0}. \end{aligned} \tag{5.9}$$

where $\mathbf{L}_i = \left(\mathbf{L}_{i1}, \dots, \mathbf{L}_{im}\right)$, $\mathbf{L}_{ij} = \delta_{ij}\left(\mathbf{u}, -1\right) + \mathbf{P}_j^T\sigma_i$, $j = 1, \dots, m$, n_q is number of rows of matrix $\mathbf{F}$. For sake of completeness, a proof of (5.9) is provided in Appendix.

5.4.2 Monotonicity

We show in this section that the function $\Pi_\Xi(\mathbf{P}, \cdot)$ is monotonic w.r.t. a partial order defined on $\mathcal{F}_{\Xi_P}$ when the shadow matrix $\mathbf{P}$ is fixed.

Given an uncertainty set Ξ, we now define a partial order over the collection of its pole-sets $\mathcal{F}_{\Xi_P}$ denoted by $\preceq_{\mathcal{F}_{\Xi_P}}$. We set members of $\mathcal{F}_{\Xi_P}$ ordered by the inclusion of their convex hulls, i.e., for any $\Omega', \Omega \in \mathcal{F}_{\Xi_P}$,

$$\Omega' \preceq_{\mathcal{F}_{\Xi_P}} \Omega \iff \operatorname{conv}\Omega' \subseteq \operatorname{conv}\Omega. \tag{5.10}$$

The next theorem emphasizes the fact that the function $\Pi_\Xi(\mathbf{P}, \cdot)$ is monotonic regarding the partial order $\preceq_{\mathcal{F}_{\Xi_P}}$ for each fixed $\mathbf{P} \in \mathbb{R}^{n_0\times\dim(\Xi)}$. In other words, the multipolar value gets smaller when Ω is smaller w.r.t. $\preceq_{\mathcal{F}_{\Xi_P}}$.

Theorem 10 *Given $\mathbf{P} \in \mathbb{R}^{n_0\times\dim(\Xi)}$, for any $\Omega', \Omega \in \mathcal{F}_{\Xi_P}$, if $\Omega' \preceq_{\mathcal{F}_{\Xi_P}} \Omega$, then we have $\Pi_\Xi(\mathbf{P}, \Omega') \leq \Pi_\Xi(\mathbf{P}, \Omega)$.*

Proof: If $(\mathbf{u}, (\mathbf{v}_\omega)_{\omega\in\Omega})$ is an optimal solution of MRC, then a feasible solution, when the set of poles is defined by Ω', is given as follows. Each $\omega' \in \Omega'$ writes as a convex combination of the poles of Ω: $\omega' = \sum\limits_{\omega\in\Omega} \lambda_\omega^{\omega'}\omega$. Let $v_{\omega'} = \sum\limits_{\omega\in\Omega} \lambda_\omega^{\omega'}\mathbf{v}_\omega$. The solution given by $(\mathbf{u}, (\mathbf{v}_{\omega'})_{\omega'\in\Omega'})$ is clearly feasible for MRC w.r.t. the set of poles defined by Ω', which completes the proof. ■

Theorem 10 not only implies that the smaller the Ω w.r.t. $\preceq_{\mathcal{F}_{\Xi_P}}$, the lower the multipolar robust value, but also implies that for a given $\mathbf{P}$, $\Pi_\Xi(\mathbf{P}, \Omega)$ is minimum if $\Omega = \operatorname{ext}(\Xi_P)$.

Given two pole-sets $\Omega, \Omega' \in \mathcal{F}_{\Xi_P}$, Theorem 10 also indicates that: first, for a fixed shadow matrix $\mathbf{P}$, if $|\Omega| > |\Omega'|$, then $\Pi_{\Xi}(\mathbf{P}, \Omega)$ is not necessarily less than $\Pi_{\Xi}(\mathbf{P}, \Omega')$; second, the function $\Pi_{\Xi}(\mathbf{P}, \cdot)$ is not strictly monotonically increasing. For example, let $S, S' \in \mathcal{F}_{\Xi_P}, S' \preceq_{\mathcal{F}_{\Xi_P}} S$ and their convex hulls are simplices. By Theorem 9, $\Pi_{\Xi}(\mathbf{P}, S') = \Pi_{\Xi}(\mathbf{P}, S)$ while by Theorem 10, $\Pi_{\Xi}(\mathbf{P}, S') \leq \Pi_{\Xi}(\mathbf{P}, S)$, which illustrates the second point. Now take any pole-set Ω whose cardinality is strictly greater than $1 + \dim(\Xi_P)$, such that $S' \preceq_{\mathcal{F}_{\Xi_P}} \Omega \preceq_{\mathcal{F}_{\Xi_P}} S$, then $\Pi_{\Xi}(\mathbf{P}, S') = \Pi_{\Xi}(\mathbf{P}, \Omega) = \Pi_{\Xi}(\mathbf{P}, S)$, which illustrates the first point.

Observe also that when $\mathbf{P} = \mathbf{I}$, any pole-set whose convex hull contains Ξ is contained in a simplex. This immediately implies that the optimal value of AARC represents the worst that can be obtained by the multipolar approach.

5.4.3 Convergence

The aim of this section is to show that under some mild assumptions, using the multipolar framework, one can simultaneously compute a sequence of upper bounds and a sequence of lower bounds converging to $\Pi_{\Xi}(\mathbf{I}, \text{ext}(\Xi))$, the optimal robust value of FARC. Throughout this section the shadow matrix is the identity matrix.

Definition 5.1 *Let $\Omega \in \mathcal{F}_{\Xi}$ be a pole-set of a non-empty set Ξ, the distance function between them is defined as* $d(\Omega, \Xi) = \max_{\omega \in \Omega} \min_{\xi \in \Xi} \|\omega - \xi\|_2$.

The distance function is well-defined since Ω and Ξ are closed and bounded. It characterizes the furtherest distance between pole-set Ω and the uncertainty set Ξ. This distance is nothing other than the well-known Hausdorff distance.

Let $\Omega \in \mathcal{F}_{\Xi}$ such that $d(\Omega, \Xi) = \epsilon$. For each $\omega \in \Omega$, we have $d(\omega, \Xi) \leq \epsilon$. Let $\mathbf{z}_\omega$ be the projection of ω on Ξ, i.e.,

$$\mathbf{z}_\omega = \text{argmin}_{\xi \in \Xi}\ d(\omega, \xi), \qquad \mathbf{e}_\omega = \omega - \mathbf{z}_\omega, \tag{5.11}$$

where $\|\mathbf{e}_\omega\|_2 \leq \epsilon$. For each $\xi \in \Xi$, consider convex combination coefficients $\left(\beta_\omega^\xi\right)$ such that

$$\xi = \sum_{\omega \in \Omega} \beta_\omega^\xi \omega, \qquad \text{and let} \qquad \mathbf{E} = \sum_{\omega \in \Omega} \beta_\omega^\xi \mathbf{e}_\omega. \tag{5.12}$$

Let us add subscripts to avoid confusion: $\xi \equiv [\mathbf{U}_\xi, \mathbf{b}_\xi]$, $\mathbf{E} \equiv [\mathbf{U}_E, \mathbf{b}_E]$ and $\mathbf{z}_\omega \equiv [\mathbf{U}_{\mathbf{z}_\omega}, \mathbf{b}_{\mathbf{z}_\omega}]$. We define the convex set

$$\Xi'_{z\Omega} = \text{conv}\left\{\mathbf{z}_\omega : \omega \in \Omega\right\}. \tag{5.13}$$

We obviously have $\Xi'_{z\Omega} \subseteq \Xi$. Let $(\mathbf{u}^*, (\mathbf{v}^*_{\mathbf{z}_\omega})_{\omega \in \Omega})$ be the optimal solution of the MRC problem related to $\Xi'_{z\Omega}$. Due to the definition of $\Xi'_{z\Omega}$, MRC and FARC are equivalent. Moreover, from $\Xi'_{z\Omega} \subseteq \Xi$, we get that

$$\mathbf{c}^T \mathbf{u}^* = \Pi_{\Xi'_{z\Omega}}(\mathbf{I}, \text{ext}(\Xi'_{z\Omega})) \leq \Pi_{\Xi}(\mathbf{I}, \text{ext}(\Xi)).$$

We will also assume that there is a positive number μ such that $\|(\mathbf{u}^*, 1)\|_2 \leq \mu$. This assumption generally holds. For example, if the cost vector $\mathbf{c}$ is positive and variables $\mathbf{u}$ are non-negative, then $\mathbf{c}^T\mathbf{u}^* \leq \Pi_\Xi(\mathbf{I}, \text{ext}(\Xi))$ implies that $\|(\mathbf{u}^*, 1)\|_2$ is upper-bounded. The number μ does not depend on ϵ.

Assumption 3 *There exists a constant number μ such that $\|(\mathbf{u}^*, 1)\|_2 \leq \mu$ for any $\Xi'_{z_\Omega} \subseteq \Xi$ and any optimal solution $\mathbf{u}^*$ of the FARC problem related to Ξ'.*

Lemma 5.1 *Under Assumption 3, for each $\xi \in \Xi$, $(\mathbf{U_E}\mathbf{u}^* - \mathbf{b_E})$ is bounded from above by $\epsilon\mu\mathbf{1}$, where $\mathbf{1}$ is an all-ones vector.*

Proof: The result follows from Cauchy-Schwartz inequality applied to each row of $\mathbf{E} \equiv [\mathbf{U}_E, \mathbf{b}_E]$. ■

Let δ be a small positive number and let

$$\Xi_\delta = \left\{\xi \equiv [\mathbf{U}, \mathbf{b}] : \exists \xi' \equiv [\mathbf{U}, \mathbf{b}'] \in \Xi, \left\|\mathbf{b} - \mathbf{b}'\right\|_\infty \leq \delta\right\}. \tag{5.14}$$

Observe that if $\xi \equiv [\mathbf{U}, \mathbf{b}] \in \Xi$, then $[\mathbf{U}, \mathbf{b} - \delta\mathbf{1}] \in \Xi_\delta$. We will assume that for some small number δ, the static robust counterpart problem SRC is still solvable.

Assumption 4 *There exists a static robust solution $(\mathbf{u}_\delta, \mathbf{v}_\delta)$ w.r.t. uncertainty set Ξ_δ.*

Theorem 11 *Under Assumptions 3 and 4, for each pole-set $\Omega \in \mathcal{F}_\Xi$ such that $d(\Omega, \Xi) = \epsilon \leq \frac{\delta}{\mu}$, we have*

$$\Pi_\Xi(\mathbf{I}, \Omega) \leq \left(1 - \frac{\epsilon\mu}{\delta}\right)\mathbf{c}^T\mathbf{u}^* + \frac{\epsilon\mu}{\delta}\mathbf{c}^T\mathbf{u}_\delta,$$

where $\mathbf{c}^T\mathbf{u}^*$ and $\mathbf{c}^T\mathbf{u}_\delta$ are respectively fully adjustable robust cost w.r.t. Ξ'_{z_Ω} and static cost w.r.t. Ξ_δ.

Proof: Assume that the optimal solution of FARC w.r.t. uncertainty set Ξ'_{z_Ω} is $\left(\mathbf{u}^*, (\mathbf{v}^*_{\mathbf{z}_\omega})\right)$. Consider the solution

$$\hat{\mathbf{u}} = \left(1 - \frac{\epsilon\mu}{\delta}\right)\mathbf{u}^* + \frac{\epsilon\mu}{\delta}\mathbf{u}_\delta, \qquad \hat{\mathbf{v}}_\omega = \left(1 - \frac{\epsilon\mu}{\delta}\right)\mathbf{v}^*_{\mathbf{z}_\omega} + \frac{\epsilon\mu}{\delta}\mathbf{v}_\delta. \tag{5.15}$$

Let us show that $(\hat{\mathbf{u}}, \hat{\mathbf{v}}_\omega)$ is a feasible solution of the MRC problem related to Ξ and Ω.

For any $\xi \equiv [\mathbf{U}, \mathbf{b}] \in \Xi$, by (5.12), one can write:

$$\begin{aligned}
\mathbf{U}_\xi \hat{\mathbf{u}} + \mathbf{V} \sum_{\omega \in \Omega} \beta_\omega^\xi \hat{\mathbf{v}}_\omega &= \mathbf{U}_\xi \hat{\mathbf{u}} + \left(1 - \frac{\epsilon\mu}{\delta}\right) \mathbf{V} \sum_{\omega \in \Omega} \beta_\omega^\xi \mathbf{v}^*_{\mathbf{z}_\omega} + \frac{\epsilon\mu}{\delta} \mathbf{V}\mathbf{v}_\delta \\
&\leq \mathbf{U}_\xi \hat{\mathbf{u}} + \left(1 - \frac{\epsilon\mu}{\delta}\right) \sum_{\omega \in \Omega} \beta_\omega^\xi \left(\mathbf{b}_{\mathbf{z}_\omega} - \mathbf{U}_{\mathbf{z}_\omega} \mathbf{u}^*\right) \\
&\qquad + \frac{\epsilon\mu}{\delta} \left(\mathbf{b}_\xi - \delta \mathbf{1} - \mathbf{U}_\xi \mathbf{u}_\delta\right) \qquad (5.16) \\
&= \mathbf{U}_\xi \hat{\mathbf{u}} + \left(1 - \frac{\epsilon\mu}{\delta}\right) \left(\mathbf{b}_\xi - \mathbf{b}_\mathbf{E} + \mathbf{U}_\mathbf{E} \mathbf{u}^* - \mathbf{U}_\xi \mathbf{u}^*\right) \\
&\qquad + \frac{\epsilon\mu}{\delta} \left(\mathbf{b}_\xi - \delta \mathbf{1} - \mathbf{U}_\xi \mathbf{u}_\delta\right) \qquad (5.17) \\
&= \left(1 - \frac{\epsilon\mu}{\delta}\right) \left(\mathbf{U}_\mathbf{E} \mathbf{u}^* - \mathbf{b}_\mathbf{E}\right) - \epsilon\mu\mathbf{1} + \mathbf{b}_\xi \\
&\leq \left(1 - \frac{\epsilon\mu}{\delta}\right) \epsilon\mu\mathbf{1} - \epsilon\mu\mathbf{1} + \mathbf{b}_\xi \qquad (5.18) \\
&= \mathbf{b}_\xi - \frac{\epsilon^2\mu^2}{\delta} \mathbf{1} \\
&\leq \mathbf{b}_\xi,
\end{aligned}$$

where (5.16) follows from the fact that $\left(\mathbf{u}^*, \mathbf{v}^*_{\mathbf{z}_\omega}\right)$ satisfies constraint $\mathbf{U}\mathbf{u} + \mathbf{V}\mathbf{v} \leq \mathbf{b}$ for $\mathbf{z}_\omega = \left[\mathbf{U}_{\mathbf{z}_\omega}, \mathbf{b}_{\mathbf{z}_\omega}\right]^T$ and the static solution $(\mathbf{u}_\delta, \mathbf{v}_\delta)$ satisfies $\mathbf{U}_\xi \mathbf{u} + \mathbf{V}\mathbf{v} \leq \mathbf{b}_\xi - \delta\mathbf{1}$, (5.17) follows from (5.11) and (5.12), and (5.18) is due to Lemma 5.1.

The robust cost incurred by $(\hat{\mathbf{u}}, (\hat{\mathbf{v}}_\omega))$ is $\left(1 - \frac{\epsilon\mu}{\delta}\right) \mathbf{c}^T \mathbf{u}^* + \frac{\epsilon\mu}{\delta} \mathbf{c}^T \mathbf{u}_\delta$ and is an upper bound of the optimum of MRC. ■

Corollary 5.3 *Given any sequence of pole-sets $\Omega_i \in \mathcal{F}_\Xi$ such that $\lim_{i\to\infty} d(\Omega_i, \Xi) = 0$, then under Assumptions 3 and 4,*

$$\Pi_\Xi (\mathbf{I}, \Omega_i) \geq \Pi_\Xi (\mathbf{I}, \mathrm{ext}(\Xi)) \quad \textit{and} \quad \lim_{i\to\infty} \Pi_\Xi (\mathbf{I}, \Omega_i) = \Pi_\Xi (\mathbf{I}, \mathrm{ext}(\Xi)).$$

Moreover, the corresponding sequence of sets $\Xi'_{z_{\Omega_i}}$ defined in (5.13) satisfies

$$\Pi_{\Xi'_{z_{\Omega_i}}} \left(\mathbf{I}, \mathrm{ext}\left(\Xi'_{z_{\Omega_i}}\right)\right) \leq \Pi_\Xi (\mathbf{I}, \mathrm{ext}(\Xi)) \quad \textit{and} \quad \lim_{i\to\infty} \Pi_{\Xi'_{z_{\Omega_i}}} \left(\mathbf{I}, \mathrm{ext}\left(\Xi'_{z_{\Omega_i}}\right)\right) = \Pi_\Xi (\mathbf{I}, \mathrm{ext}(\Xi)).$$

Proof: Let $\epsilon_i = d(\Omega_i, \Xi), \forall i$. From Theorem 11, we have

$$\Pi_\Xi (\mathbf{I}, \Omega_i) \leq \left(1 - \frac{\epsilon_i\mu}{\delta}\right) \Pi_{\Xi'_{z_{\Omega_i}}} \left(\mathbf{I}, \mathrm{ext}\left(\Xi'_{z_{\Omega_i}}\right)\right) + \frac{\epsilon_i\mu}{\delta} \mathbf{c}^T \mathbf{u}_\delta, \ \forall i,$$

and we know that

$$\Pi_{\Xi'_{z_{\Omega_i}}} \left(\mathbf{I}, \mathrm{ext}\left(\Xi'_{z_{\Omega_i}}\right)\right) \leq \Pi_\Xi (\mathbf{I}, \mathrm{ext}(\Xi)) \leq \Pi_\Xi (\mathbf{I}, \Omega_i).$$

Consequently,

$$\lim_{i\to\infty} \Pi_{\Xi'_{z_{\Omega_i}}} \left(\mathbf{I}, \mathrm{ext}\left(\Xi'_{z_{\Omega_i}}\right)\right) = \Pi_\Xi (\mathbf{I}, \mathrm{ext}(\Xi)) \ \text{ and } \ \lim_{i\to\infty} \Pi_\Xi (\mathbf{I}, \Omega_i) = \Pi_\Xi (\mathbf{I}, \mathrm{ext}(\Xi))$$

hold in the limit at the same time. ■

5.5 An illustrative example

To illustrate the multipolar concept, we present a simple example, which had been previously studied in [58] and is as follows:

$$\begin{array}{lll} \min & u & \\ \text{s.t.} & \forall \xi \in \mathbb{R}^n, \|\xi\|_1 \leq 1, & \exists \mathbf{v}, v_i \geq \xi_i, v_i \geq -\xi_i, i = 1, \ldots, n \\ & & u \geq \sum\limits_{i=1}^{n} v_i . \end{array} \tag{5.19}$$

Observe that u is here the unique first-stage (non adjustable) variable. On the other hand v_i, for each $i = 1, \ldots, n$, are second-stage (adjustable) variables. The uncertainty set is given by $\Xi \equiv \{\xi \in \mathbb{R}^n, \|\xi\|_1 \leq 1\}$.

As noticed in [58], an optimal fully adjustable solution is given by $u = 1$ and $v_i = \|\xi_i\|_1$, whereas the optimal affinely adjustable solution requires that $u = n$. In other words, the affine approach does not lead to any improvement compared to the static approach.

Following the paradigm of multipolar approach in Section 5.3, let us take $\mathbf{P}\xi = (\xi_1, \ldots, \xi_{n_0})$, where $n_0 \in \mathbb{N}, n_0 \leq n$. In other words, the shadow matrix $\mathbf{P}$ limits the dimension of Ξ to n_0 by leaving ξ_i as they are for $i \leq n_0$ and disregarding the other components for $i > n_0$. Let $\Omega \subseteq \mathbb{R}^{n_0}$ be the set of poles containing for $i = 1, \ldots, n_0$, vectors $\phi^i = (0, \ldots, 0, 1, 0, \ldots, 0)$ and $\overline{\phi}^i = -\phi^i$, whose components are 0 except the i^{th} component. Hence Ω contains $2n_0$ poles and $\Xi_P = \operatorname{conv} \Omega$.

Given any $\xi \subseteq \Xi$, let λ_{ϕ^i} and $\lambda_{\overline{\phi}^i}$ be the convex combination coefficients such that $\mathbf{P}\xi = \sum\limits_{i=1}^{n_0} \left(\lambda_{\phi^i}\phi^i + \lambda_{\overline{\phi}^i}\overline{\phi}^i\right)$. The equation can be transformed to $\mathbf{P}\xi = \sum\limits_{i=1}^{n_0} \phi^i(\lambda_{\phi^i} - \lambda_{\overline{\phi}^i})$; thus these coefficients should satisfy the equations $\lambda_{\phi^i} - \lambda_{\overline{\phi}^i} = \xi_i$ for $1 \leq i \leq n_0$. Let $\mathbf{v}_{\phi^i}$ (resp. $\mathbf{v}_{\overline{\phi}^i}$) be the recourse vector associated with pole ϕ^i (resp. $\overline{\phi}^i$). These vectors belong to $\mathbb{R}^n$.

In the considered example, inequalities (5.7) are equivalent to the following set of inequalities:

$$\sum_{i=1}^{n_0} \left(\lambda_{\phi^i}\mathbf{v}_{\phi^i} + \lambda_{\overline{\phi}^i}\mathbf{v}_{\overline{\phi}^i}\right) \geq (|\xi_1|, |\xi_2|, \ldots, |\xi_n|)\,, \tag{5.20a}$$

$$u \geq \left\| \sum_{i=1}^{n_0} \left(\lambda_{\phi^i}\mathbf{v}_{\phi^i} + \lambda_{\overline{\phi}^i}\mathbf{v}_{\overline{\phi}^i}\right) \right\|_1 . \tag{5.20b}$$

Let us take $\mathbf{v}_{\phi^i} = \mathbf{v}_{\overline{\phi}^i} = (0, \ldots, 0, 1, 0, \ldots, 0, 1, \ldots, 1)$, where the first n_0 components are 0 except the i^{th} component, which is equal to 1, while the last $(n - n_0)$ components are equal to 1.

Observe that the last $(n - n_0)$ components of the vector $\sum\limits_{i=1}^{n_0} \left(\lambda_{\phi^i}\mathbf{v}_{\phi^i} + \lambda_{\overline{\phi}^i}\mathbf{v}_{\overline{\phi}^i}\right)$ are equal to 1. Moreover, for $1 \leq i \leq n_0$, we have $\lambda_{\phi^i} + \lambda_{\overline{\phi}^i} \geq |\lambda_{\phi^i} - \lambda_{\overline{\phi}^i}| = |\xi_i|$. This clearly implies that inequalities (5.20a) are satisfied. In addition, inequality (5.20b) leads to $u \geq \sum\limits_{i=1}^{n_0} \left(\lambda_{\phi^i} + \lambda_{\overline{\phi}^i}\right)(1 + n - n_0)$, where $(1 + n - n_0)$ is the L^1 norm of each recourse vector $\mathbf{v}_{\phi^i}$. Consequently, $u \geq 1 + n - n_0$. Since we are minimizing u, we get $u = 1 + n - n_0$. The cost decreases when n_0 increases. When n_0 is equal to 1, we get a static solution, while the optimal fully adjustable solution is obtained when $n_0 = n$. Finally, taking $1 < n_0 < n$, we obtain a compromise between the simplicity of the static approach and the efficiency of the fully adjustable solution. As mentioned earlier, such a compromise cannot be obtained for this example with the affinely adjustable approach.

Consider now a slightly changed example with the uncertainty set being the non-polyhedral set defined by $\Xi := \{\xi \in \mathbb{R}^n : \|\xi\|_2 \leq 1\}$. The rest of the problem remains as in (5.19); thus the new problem can be formulated as follows:

$$\begin{array}{lll} \min & u & \\ \text{s.t.} & \forall \xi \in \mathbb{R}^n, ||\xi||_2 \leq 1, & \exists \mathbf{v}, v_i \geq \xi_i, v_i \geq -\xi_i, i = 1, \ldots, n, \\ & & u \geq \sum_{i=1}^{n} v_i. \end{array} \tag{5.21}$$

Observe that since the new uncertainty set Ξ contains the previous one based on L^1 norm, the optimal value of (5.21) is greater than or equal to that of (5.19). Optimal solutions based on either the static approach or the affine approach still incur a cost of n while the optimal fully adjustable solution has a cost of $\sqrt{n}$. Let us now consider the multipolar approach, where $\mathbf{P}$ is still defined by $P\xi = (\xi_1, \ldots, \xi_{n_0})$. Let us choose the following set of poles: $\Omega = \{\sqrt{n_0}\phi^i\}_{i=1}^{i=n_0} \cup \{\sqrt{n_0}\overline{\phi}^i\}_{i=1}^{i=n_0}$. One can easily show that $\Xi_P \subseteq \text{conv}(\Omega)$. Moreover, by taking $\mathbf{v}_{\sqrt{n_0}\phi^i} = \mathbf{v}_{\sqrt{n_0}\overline{\phi}^i} = (0, \ldots, 0, \sqrt{n_0}, 0, \ldots, 0, 1, \ldots, 1)$, where the first n_0 components are 0 except the i^{th} component, which is equal to $\sqrt{n_0}$, while the last $n - n_0$ components are equal to 1, we get a solution of the multipolar robust counterpart with $u = \sqrt{n_0} + n - n_0$. Similarly to the previous case, when n_0 is equal to 1, we get a static solution, while the optimal fully adjustable solution is obtained when $n_0 = n$.

5.6 The construction of pole-sets

We know from Section 5.4 that the multipolar robust value converges to a fully adjustable robust value when the distance between Ω and Ξ_P gets close to 0, and $\mathbf{P} = \mathbf{I}$. We also proved the monotonicity of multipolar robust value w.r.t. the inclusion of $\text{conv}\,\Omega$. Therefore, the objective of this section is to find a pole-set $\Omega \in \mathcal{F}_P$ as close to Ξ_P as possible, while minimizing the number of poles. This is clearly related to the theory of approximation of convex sets by polytopes.

A considerable amount of work has been done in this area. A recent survey of relevant results is given in [47]. It is proved in [46, 63] that given a convex body $\Xi_P \in \mathbb{R}^{n_0}$, there exists a polytope $F_n \in \mathbb{R}^{n_0}$ having n vertices containing Ξ_P such that $d_H(\Xi_P, F_n) \leq \frac{k(\Xi_P)}{n^{2/(n_0-1)}}$ where d_H denotes the Hausdorff distance and $k(\Xi_P)$ is a constant only depending on Ξ_P. More precise approximations are obtained in dimension 2, where we can ensure the existence of $F_n \subset \mathbb{R}^2$ such that $d_H(\Xi_P, F_n) \leq \frac{l}{2n} \sin \frac{\pi}{n}$ where l is the length of the boundary of Ξ_P. Moreover, if the boundary of Ξ_P is two-times smooth, then an explicit asymptotic result is known about the distance between Ξ_P and the set of circumscribed polytopes having n vertices: the closest polytope F_n satisfies $d_H(\Xi_P, F_n) \sim \frac{k(\Xi_P)}{n^{2/(n_0-1)}}$ where $k(\Xi_P)$ is a constant depending on n_0 and the Gaussian curvature of the boundary of Ξ_P [44].

The monotonicity of multipolar robust values w.r.t. pole-sets might suggest using minimum volume circumscribed polytopes. Considering the Nikodym distance (related to volumes) instead of the Hausdorff distance, the same kind of results can be obtained [47]. One might be interested in a minimum volume simplex containing a convex set Ξ_P. We know for example that if Ξ_P is the hypercube H_{n_0}, then a minimum volume circumscribed simplex has a volume equal to $\frac{{n_0}^{n_0}}{n_0!}$ [100]. If Ξ_P is the unit ball, then a minimum volume simplex containing the ball is a regular simplex whose volume is $\frac{n_0^{n_0/2}(n_0+1)^{(n_0+1)/2}}{n_0!}$ [47] and whose dihedral angle is $\arccos(\frac{1}{n_0})$ [134]. It is also known that a minimum volume simplex enclosing Ξ_P satisfies the centroid property: the centroid of each facet of this simplex should be in Ξ_P [96]. A polynomial-time algorithm to find such a minimum volume simplex enclosing a set of points in $\mathbb{R}^3$ is given in [155]. However, it is generally unknown how to solve the problem in higher dimensions [76].

As observed by [47], most constructive algorithms were generally proposed for low dimensional cases (2 or 3). For more general cases, constructive algorithms of circumscribed polytopes such as the algorithm of [91] are generally based on the addition of inequalities without controlling the number of vertices of the circumscribed polytope. This can hardly accommodate the need of multipolar framework since we want to control the complexity of MRC by limiting the number of poles.

Note also that we are required to construct the pole-set of Ξ_P in a reasonable time. Algorithms checking whether each extreme point of Ξ_P belongs to the convex hull of Ω fail to work, since the number of extreme points of a polytope can be exponential or even infinite.

The rest of this section is organized as follows. First, we describe a general algorithm to construct a simplex enclosing Ξ_P. The resulting simplex is guaranteed to be smallest in the sense that it cannot be shrinked. Then, a project-and-cut based tightening procedure is proposed to construct pole-sets that are closer to Ξ_P.

5.6.1 Generation of a circumscribed simplex

In this section, we describe a general algorithm for the construction of a circumscribed simplex of Ξ_P. Specifically, we first randomly generate a set of $(n_0 + 1)$ affinely independent points, whose convex hull forms a simplex S. Then, we compute the best homothetic transformation of S such that the resulting simplex contains Ξ_P.

We denote by the $\omega^{(i)}, i = 1, \dots, (n_0 + 1)$ the $(n_0 + 1)$ affinely independent points. Then the n_0-simplex set can be expressed as $\{\mathbf{x} : \ \mathbf{D}\boldsymbol{\lambda} = (\mathbf{x}, 1), \boldsymbol{\lambda} \geq \mathbf{0}\}$, where

$$\mathbf{D} = \begin{pmatrix} \omega_1^{(1)} & \dots & \omega_1^{(n_0+1)} \\ \vdots & \vdots & \vdots \\ \omega_{n_0}^{(1)} & \dots & \omega_{n_0}^{(n_0+1)} \\ 1 & \dots & 1 \end{pmatrix}.$$

Since the $(n_0 + 1)$ points are affinely independent, matrix $\mathbf{D}$ is invertible; therefore, $\lambda_j, \ j = 1, \dots, (n_0 + 1)$, can be expressed as an affine function of $\mathbf{x}$; the coefficients of the affine function λ_i are the components of the ith row of $\mathbf{D}^{-1}$, i.e.,

$$\lambda_i(x) = \sum_{j=1}^{n_0} l_{ij}x_j + l_{i(n_0+1)},\ i = 1, \ldots, (n_0+1). \tag{5.22}$$

Note that $\lambda_i(x) \geq 0,\ i = 1, \ldots, (n_0+1)$ iff $\mathbf{x}$ belongs to the n-simplex.

Let $\boldsymbol{\sigma}_\sigma, \mathbf{T}_\mathbf{t}$ be the associate matrices for the operations of scaling with factor $\sigma > 0$ and translation $\mathbf{t} \in \mathbb{R}^{n_0}$. Thus the associated matrix $\mathbf{D}_{\sigma,t}$ of the simplex with homothetic transformation on the simplex S is $\mathbf{D}_{\sigma,t} = \mathbf{T}_\mathbf{t}\boldsymbol{\sigma}_\sigma\mathbf{D}$, where

$$\mathbf{T}_\mathbf{t} = \begin{pmatrix} \mathbf{I}_{n_0} & \mathbf{t} \\ 0 & 1 \end{pmatrix},\ \boldsymbol{\sigma}_\sigma = \begin{pmatrix} \sigma\mathbf{I}_{n_0} & 0 \\ 0 & 1 \end{pmatrix}.$$

Its corresponding inverse is then

$$\begin{aligned}\mathbf{D}_{\sigma,\mathbf{t}}^{-1} &= \mathbf{D}^{-1}\begin{pmatrix} \frac{1}{\sigma}I_{n_0} & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} \mathbf{I}_{n_0} & -\mathbf{t} \\ 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{\sigma}l_{11} & \ldots & \frac{1}{\sigma}l_{1n_0} & l_{1(n_0+1)} - \frac{1}{\sigma}\sum_{k=1}^{n_0} t_k l_{1k} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{1}{\sigma}l_{n_01} & \ldots & \frac{1}{\sigma}l_{n_0n_0} & l_{n_0(n_0+1)} - \frac{1}{\sigma}\sum_{k=1}^{n_0} t_k l_{n_0k} \\ \frac{1}{\sigma}l_{(n_0+1)1} & \ldots & \frac{1}{\sigma}l_{(n_0+1)n_0} & l_{(n_0+1)(n_0+1)} - \frac{1}{\sigma}\sum_{k=1}^{n_0} t_k l_{(n_0+1)k} \end{pmatrix}.\end{aligned}$$

Let σ^* be the smallest scaling factor σ such that a translate of σS contains Ξ_P. The translate used when $\sigma = \sigma^*$ is denoted by $\mathbf{t}^*$.

Theorem 12 *σ^* and $\mathbf{t}^*$ are given by: $\mathbf{t}^* = \sum_{i=1}^{n_0+1} z_i\omega^{(i)}$ and $\sigma^* = -\sum_{i=1}^{n_0+1} z_i$, where for each $i = 1, \ldots, (n_0+1)$, $z_i = \min\{\sum_{j=1}^{n_0} l_{ij}x_j :\ \mathbf{x} \in \Xi_P\}$.*

Proof: Assume that the homothetic copy of S given by $\sigma S + \mathbf{t}$ contains Ξ_P. Then the coefficients $\lambda_i(x)$ defined in (5.22) should be nonnegative for any point $\mathbf{x} \in \Xi_P$. Considering the matrix $\mathbf{D}_{\sigma,t}^{-1}$ defined above and computing the minimum values of $\lambda_i(x), i = 1, \ldots, (n_0+1)$, we get

$$l_{i(n_0+1)} - \frac{1}{\sigma}\sum_{k=1}^{n_0} t_k l_{ik} + \frac{1}{\sigma}z_i \geq 0,\ i = 1, \ldots, (n_0+1),$$

For ease of notation, we express this as $l'_{i(n_0+1)} + z_i \geq 0,\ i = 1, \ldots, (n_0+1)$, where $l'_{i(n_0+1)} = \sigma l_{i(n_0+1)} - \sum_{k=1}^{n_0} t_k l_{ik},\ i = 1, \ldots, (n_0+1)$.
Since the matrix $(l)_{i,j=1,\ldots,(n_0+1)}$ is the inverse of matrix $\mathbf{D}$, we have

$$\sum_{i=1}^{n_0+1} l_{ij} = 0,\ j = 1, \ldots, n_0 \ \text{ and } \sum_{i=1}^{n_0+1} l_{i(n_0+1)} = 1. \tag{5.23}$$

Summing all $l'_{i(n_0+1)}$, we get

$$\sigma = \sum_{i=1}^{n_0+1} l'_{i(n_0+1)} \geq -\sum_{i=1}^{n_0+1} z_i. \tag{5.24}$$

Observe that having $l'_{i(n_0+1)}$ (and σ^* as a consequence), we can get the translate t through the linear system $\sum\limits_{k=1}^{n_0} t_k l_{ik} = \sigma l_{i(n_0+1)} - l'_{i(n_0+1)},\ i = 1, \ldots, (n_0+1)$. Multiplying by $\mathbf{D}$, we get that $(t_1, ..., t_{n_0}, 0) = \sigma(0, ..., 0, 1) - \mathbf{D}(l'_{1(n_0+1)}, ..., l'_{(n_0+1)(n_0+1)})$ which leads to $\mathbf{t} = \sum\limits_{i=1}^{n_0+1} z_i \omega^{(i)}$.

According to (5.24), the smallest σ^* is $-\sum\limits_{i=1}^{n_0+1} z_i$. We should however check if $\sigma^* \geq 0$. This holds since by considering any $x \in \Xi_P$, one can write that

$$\begin{aligned}\sum_{i=1}^{n_0+1} z_i \leq & \sum_{i=1}^{n_0+1}\sum_{j=1}^{n_0} l_{ij}x_j \\ =& \sum_{j=1}^{n_0} x_j \sum_{i=1}^{n_0+1} l_{ij} \\ =& 0,\end{aligned}$$

where the last equality is based on (5.23).

Since the matrix $\mathbf{D}^{-1}_{\sigma,t}$ does not exist when $\sigma^* = 0$, we have to study this special case. It is clear that $\sigma^* = 0$ if and only if Ξ_P is a single point. Observe that in this case, we necessarily have $\sum\limits_{i=1}^{n_0+1} z_i = 0$ since $z_i = \sum_{j=1}^{n_0} l_{ij}x_j$, where $\mathbf{x}$ is the single point of Ξ_P. Then formula $\sigma^* = -\sum\limits_{i=1}^{n_0+1} z_i$ is still valid and $\mathbf{t}^* = \mathbf{x} = \sum\limits_{i=1}^{n_0+1} z_i \omega^{(i)}$ also occurs. ■

Note that values of $z_i, i = 1, \ldots, (n_0+1)$ defined in Theorem 12 can easily be computed for any Ξ_P since we only have to minimize a linear function over a convex set.

As a special case, pole-sets of a hypercube are of great use in multipolar robust approach. First, hypercubes are one of the most common uncertainty sets in many applications. Second, general box sets of the form $\{\mathbf{x} : \mathbf{x} \in [\mathbf{l}, \mathbf{u}]^{n_0} \subseteq \mathbb{R}^{n_0}\}$ are simply affine transformations of a hypercube, so the pole-sets of a hypercube also apply to boxes with some simple transformations.

Corollary 5.4 *If Ξ_P is a hypercube, then σ^* and $\mathbf{t}^*$ are given by:*

$$\mathbf{t}^* = \sum_{i=1}^{n+1}\sum_{j=1}^{n_0} \min\{0, l_{ij}\}\omega^{(i)} \ \textit{and}\ \sigma^* = \frac{1}{2}\sum_{i=1}^{n_0+1}\sum_{j=1}^{n_0} |l_{ij}|.$$

Proof: If Ξ_P is a hypercube, by Theorem 12, we have $z_i = \sum\limits_{j=1}^{n_0} \min\{0, l_{ij}\},\ i = 1, \ldots, n_0$. By (5.23), we have $-\sum\limits_{i=1}^{n_0+1}\sum\limits_{j=1}^{n_0} \min\{0, l_{ij}\} = \frac{1}{2}\sum\limits_{i=1}^{n_0+1}\sum\limits_{j=1}^{n_0} |l_{ij}|$, which completes the proof. ■

According to Corollary 5.4, we have a closed formula for the homothetic translation for a n_0-simplex S containing the n_0-hypercube, i.e,

$$\mathbf{x}_{\sigma,t} = \frac{1}{2}\sum_{i=1}^{n_0+1}\sum_{j=1}^{n_0} |l_{ij}|\mathbf{x} + \sum_{i=1}^{n_0+1} \omega^{(i)} \sum_{j=1}^{n_0} \min\{0, l_{ij}\},\ \forall \mathbf{x} \in S. \tag{5.25}$$

Note that the value σ^* presented in Corollary 5.4 has been given in [128] but the proof here is much simpler.

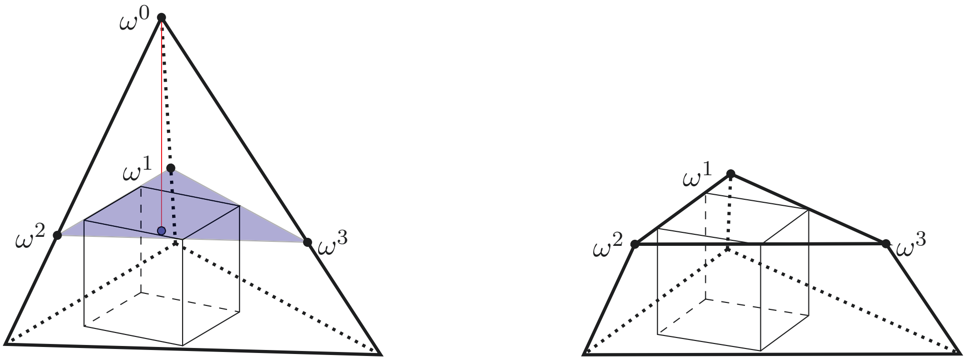

Figure 5.1: An illustration: (Left) Pole ω^0 is replaced with poles $(\omega^i)_{i=1}^3$. (Right) The updated convex hull of the new set of poles.

To sum up the foregoing, we present a general algorithm for the generation of a circumscribed simplex as follows.

1. Generate (n_0+1) affinely independent points $(\omega^{(i)})_{i=1}^{i=n_0+1}$.

2. Compute σ^* and $\mathbf{t}^*$ by Theorem 12 and output the $\sigma^*\omega^{(i)}+\mathbf{t}^*, i=1,\ldots,(n_0+1)$.

5.6.2 A tightening procedure

In this section, we propose a general procedure to construct pole-sets of good quality by tightening a given pole-set.

The procedure is the following: among the vertices of Ω select the farthest one in L^2 sense from Ξ_P and compute the projection of this vertex on Ξ_P. Then we consider the hyperplane separating this vertex from Ξ_P (containing the projection) and compute the extreme points of the intersection of this hyperplane with $\operatorname{conv}\Omega$. These extreme points are added to Ω while the vertex that has been projected is removed from Ω. Figure 5.1 illustrates a tightening procedure of a 3-D simplex covering H_3. The procedure is repeated until the cardinality of Ω reaches some fixed upper bound. Details are given below:

1. Assume $\Omega=\left\{\omega^{(k)}, k\in I\right\}$. For each $k\in I$, compute the distance between $\omega^{(k)}$ and Ξ_P. Let $\mathbf{z}^k$ be the projection of $\omega^{(k)}$ on Ξ_P. $\mathbf{z}^k$ can be usually expressed in a closed form. For example, in the ball case, we have $\mathbf{z}^k=\frac{\omega^{(k)}}{\left\|\omega^{(k)}\right\|_2}$, while in the hypercube case we get $\mathbf{z}_i^k=\omega_i^{(k)}$ if $\omega_i^{(k)}\in[0,1]$, $z_i^k=1$ if $\omega_i^{(k)}\geq 1$ and $z_i^k=0$ if not. The distance between $\omega^{(k)}$ and Ξ_P is then given by $\left\|\omega^{(k)}-\mathbf{z}^k\right\|_2$. Let $\omega^{(k_0)}$ be the vertex of Ω maximizing the distance from Ξ_P:
$$\omega^{(k_0)}=\arg\max_{\omega^{(k)}\in\Omega}\left\|\omega^{(k)}-\mathbf{z}^k\right\|_2.$$

2. Let $\boldsymbol{\alpha}=\omega^{(k_0)}-\mathbf{z}^{k_0}$ and let $B\left(\omega^{(k_0)},\|\boldsymbol{\alpha}\|_2\right)$ be the ball of radius $\|\boldsymbol{\alpha}\|_2$ centered at $\omega^{(k_0)}$. Since $B\left(\omega^{(k_0)},\|\boldsymbol{\alpha}\|_2\right)\cap\Xi_P=\{\mathbf{z}^{k_0}\}$ and $B\left(\omega^{(k_0)},\|\boldsymbol{\alpha}\|_2\right)$ and Ξ_P are convex, there is a hyperplane separating them. This hyperplane, denoted by $h(\omega^{(k_0)})$, is here uniquely defined since it contains $\mathbf{z}^{k_0}$ and is orthogonal to $\boldsymbol{\alpha}$. It is then given

by $h\left(\omega^{(k_0)}\right) = \left\{\mathbf{x} : (\mathbf{x} - \mathbf{z}^{k_0})^T \boldsymbol{\alpha} = 0\right\}$. We use $h^-\left(\omega^{(k_0)}\right) = \left\{\mathbf{x} : (\mathbf{x} - \mathbf{z}^{k_0})^T \boldsymbol{\alpha} < 0\right\}$ and $h^+\left(\omega^{(k_0)}\right) = \left\{\mathbf{x} : (\mathbf{x} - \mathbf{z}^{k_0})^T \boldsymbol{\alpha} \geq 0\right\}$ to respectively denote the inner and outer half spaces.

3. Now partition the vertices $\left(\omega^{(k)}\right)_{k \in I}$ into two disjoint sets: Ω^- and Ω^+, where $\Omega^- = \left\{\omega^{(k)} : \omega^{(k)} \in h^-\left(\omega^{(k_0)}\right)\right\}$ and $\Omega^+ = \left\{\omega^{(k)} : \omega^{(k)} \in h^+\left(\omega^{(k_0)}\right)\right\}$. Then consider the set of vertices Ω' obtained as intersections between the hyperplane $h(\omega^{(k_0)})$ and the set of lines $(\omega^{(i)}, \omega^{(j)})$ where $\omega^{(i)} \in \Omega^+$ and $\omega^{(j)} \in \Omega^-$: $\Omega' = \bigcup_{\omega^{(i)} \in \Omega^+, \omega^{(j)} \in \Omega^-} h(\omega^{(k_0)}) \cap (\omega^{(i)}, \omega^{(j)})$. The number of such intersections is of course less than $|\Omega^-| \times |\Omega^+|$. Also note that we need to remove redundant points from Ω' if they are convex combinations of other points of Ω'. Finally update Ω by deleting Ω^+ and adding Ω': $\Omega = \Omega^- \cup \Omega'$. If cardinality of Ω is still under a prescribed upper bound, the procedure is repeated.

To conclude this section, we might add that it is sometimes more efficient to start with a pole-set having more than $(n_0 + 1)$ poles. Assume, for example, that Ξ_P is the unit ball $\{\mathbf{x} \in \mathbb{R}^{n_0} : \|\mathbf{x}\|_2 \leq 1\}$. Then one can consider a *2n-pole-set* where poles are the $2n_0$ extreme points of $\left\{\mathbf{x} \in \mathbb{R}^{n_0} : \|\mathbf{x}\|_1 \leq \sqrt{n_0}\right\}$. Of course, 2n-pole-sets can also be easily generated for many other convex sets.

5.7 A numerical example: the lobbying problem

Let us consider a lobbying problem where a set of voters (for example, legislators or members of regulatory agencies) have to take some decisions. The opinion of each voter depends on the opinion of some authorities. Authority's opinions are generally uncertain. A lobby would like to ensure that an important decision will be unanimously approved by all voters. The lobby will spend some effort (energy, money, etc.) to convince each voter, while the total lobbying budget is minimized. Assume that there are m voters and n authorities. The opinion of voter i is given by $\sum_{j=1}^n Q_{ij}\xi_j$ where Q_{ij} is an estimated number belonging to $[-1, 1]$ and ξ_j represents the uncertain opinion of authority j. If Q_{ij} is close to 1, then j has a big impact on i, while $Q_{ij} = 0$ means that i does not care about j, while a negative value of Q_{ij} can be interpreted as a negative effect (i.e., when j recommends something, i is inclined to have an opposite opinion). We assume here that the lobby would be satisfied if $\sum_{j=1}^n Q_{ij}\xi_j \leq 0$ for each voter. Since this might not occur for some voters, some effort modeled here by $v_i(\xi)$ can be made by the lobby to convince them. The total effort is quantified by $\sum_{i=1}^m r_i v_i(\xi)$ where r_i is a unit effort price corresponding with voter i. The problem can be formulated as follows,

$$\begin{aligned} \min \quad & u \\ \text{s.t.} \quad & \sum_{i=1}^m r_i v_i(\xi) \leq u, \ \xi \in \Xi, \\ & \mathbf{Q}\xi \leq \mathbf{v}(\xi), \ \xi \in \Xi, \\ & \mathbf{v}(\xi) \geq \mathbf{0}, \ \xi \in \Xi, \end{aligned} \tag{5.26}$$

where $\mathbf{Q} \in [-\mathbf{1}, \mathbf{1}]^{m \times n}$, Ξ is the convex uncertainty set and u is the budget that has to be secured by the lobby. The lobby problem is related to opinion dynamics in social networks (see [1] and the references therein). Notice that interactions between voters are also possible since the set of authorities might include the set of voters as a subset.

To illustrate the multipolar robust approach, we consider here two different uncertainty sets: the hypercube H_n and a unit volume ball B_n. The numbers r_i are assumed to be equal to 1. Specializing (5.8) to H_n, we get the following formulation for MRC associated with a shadow matrix $\mathbf{P}$ and a feasible pole-set $\Omega \in \mathcal{F}_{PH_n}$,

$$
\begin{aligned}
\Pi_{H_n}(\mathbf{P}, \Omega) = \min_{u, \mathbf{v}_\omega} \quad & u \\
\text{s.t.} \quad & \sum_{i=1}^{m} \mathbf{v}_\omega^i + \sum_{j=1}^{n} \sigma_j \omega_j + \sum_{j=1}^{n} \beta_j \le u, \ \omega \in \Omega, \\
& \sum_{j=1}^{n} \eta_{ij}^T \omega_j + \sum_{j=1}^{n} \alpha_{ij} - v_\omega^i \le 0, \ \omega \in \Omega, i = 1, \ldots, m, \\
& \sum_{j=1}^{n} \mu_{ij} - \mathbf{V}_\omega^i - \sum_{j=1}^{n} \delta_{ij} \omega_j \le 0, \ \omega \in \Omega, i = 1, \ldots, m, \\
& \boldsymbol{\beta} + \mathbf{P}^T \boldsymbol{\sigma} \ge \mathbf{0}, \\
& \alpha_{ij} + \mathbf{P}^T \boldsymbol{\eta}_i \ge Q_{ij}, \ i = 1, \ldots, m, j = 1, \ldots, n, \\
& \boldsymbol{\mu}_i - \mathbf{P}^T \tau_i \ge \mathbf{0}, \ i = 1, \ldots, m, \\
& \alpha_{ij}, \beta_j, \mu_{ij} \ge 0, \ i = 1, \ldots, m, j = 1, \ldots, n, \\
& \boldsymbol{\sigma}, \boldsymbol{\eta}_i, \tau_i \in \mathbb{R}^{n_0}, \ i = 1, \ldots, m.
\end{aligned}
\tag{5.27}
$$

As said above, we also consider the case where the uncertainty set is a unit volume ball B_n, whose center is $\bar{\xi} = (\frac{1}{2}, \ldots, \frac{1}{2}, \ldots, \frac{1}{2})$ and radius $\rho = (\frac{\Gamma(n/2+1)}{\pi^{n/2}})^{\frac{1}{n}}$. According to (5.9), the dual of the multipolar robust counterpart w.r.t. B_n, writes

$$
\begin{aligned}
\Pi_{B_n}(\mathbf{P}, \Omega) = \min_{u, \mathbf{v}_\omega} \quad & u \\
\text{s.t.} \quad & \rho \left\| \mathbf{P}^T \sigma \right\|_2 + \mathbf{1}^T \mathbf{v}_\omega - \omega^T \boldsymbol{\sigma} \le u, \quad \omega \in \Omega, \\
& \rho \left\| \mathbf{Q}_i^T + \mathbf{P}^T \boldsymbol{\eta}_i \right\|_2 - \mathbf{v}_\omega^i - \omega^T \boldsymbol{\eta}_i + \mathbf{Q}_i \bar{\xi} \le 0, \quad \omega \in \Omega, i = 1, \ldots, m, \\
& \rho \left\| \mathbf{P}^T \boldsymbol{\mu}_i \right\|_2 - \mathbf{v}_\omega^i - \omega^T \boldsymbol{\mu}_i \le 0, \ \omega \in \Omega, i = 1, \ldots, m, \\
& \boldsymbol{\sigma}, \boldsymbol{\eta}_i, \boldsymbol{\mu}_i \in \mathbb{R}^{n_0}, \ i = 1, \ldots, m,
\end{aligned}
\tag{5.28}
$$

where $\Omega \in \mathcal{F}_{PB_n}$.

As stated in Section 5.3.1, the fully adjustable robust value w.r.t. hypercube H_n can be achieved by simply taking $\mathbf{P} = \mathbf{I}$ and $\Omega = \text{ext}(H_n)$. The problem looks more complex in the ball case since the number of extreme points is infinite. Assume again that $\mathbf{P}$ is identity. Given $\xi \in B_n$, the optimal solution of v_i is $\max\{0, \mathbf{Q}_i \xi\}$ for each $i = 1, \ldots, m$. Denote by $\mathcal{P}$ the power set of the index set {1,. . . ,m }. We partition the ball B_n into a family of disjoint subsets by a set valued mapping $S : \mathcal{P} \mapsto 2^{B_n}$, i.e., for each $J \in \mathcal{P}$,

$$
S(J) := \left\{ \xi \in B_n : \mathbf{Q}_i \xi \ge 0, \ i \in J, \ \mathbf{Q}_j \xi \le 0, \ j \in \bar{J} \right\},
$$

where $\bar{J} = \{1, \dots, m\} \setminus J$. Therefore, the fully adjustable robust program writes

$$\Pi^*_{B_n} = \max_{J \in \mathcal{P}: S(J) \neq \emptyset} \max_{\xi \in S(J)} \sum_{i \in J} \mathbf{Q}_i \xi. \tag{5.29}$$

Notice that (5.29) takes an exponential number (in the number of constraints m) of seconder order cone programs to obtain the fully adjustable robust value $\Pi^*_{B_n}$. We show that (5.29) is equivalent to a much simpler problem.

Lemma 5.2 *Program* (5.29) *is equivalent to*

$$\Pi^*_{B_n} = \max_{J \in \mathcal{P}} \left\{ \rho \left\| \sum_{i \in J} \mathbf{Q}_i \right\|_2 + \sum_{i \in J} \mathbf{Q}_i \bar{\xi} \right\}. \tag{5.30}$$

Proof: Observe first that $\max_{J \in \mathcal{P}} \left\{ \rho \left\| \sum_{i \in J} \mathbf{Q}_i \right\|_2 + \sum_{i \in J} \mathbf{Q}_i \bar{\xi} \right\}$ is an upper bound of $\Pi^*_{B_n}$ since it is obtained by relaxing the constraints $\xi \in S(J)$.

We show that it is also a lower bound of $\Pi^*_{B_n}$. Let $J_{max} \in \mathcal{P}$ be a subset for which the maximum is achieved:

$$J_{max} = \arg\max_{J \in \mathcal{P}} \left\{ \rho \left\| \sum_{i \in J} \mathbf{Q}_i \right\|_2 + \sum_{i \in J} \mathbf{Q}_i \bar{\xi} \right\}.$$

Take $\hat{\xi} \in B_n$ such that $\hat{\xi} = \bar{\xi} + \frac{\rho \sum_{i \in J_{max}} \mathbf{Q}_i}{\left\| \sum_{i \in J_{max}} \mathbf{Q}_i \right\|_2}$. Let $K \in \mathcal{P}$ such that

$$\begin{aligned} &\mathbf{Q}_i \hat{\xi} \geq 0, \; i \in K, \\ &\mathbf{Q}_i \hat{\xi} \leq 0, \; i \in \{1, \dots, m\} \setminus K. \end{aligned}$$

We have that

$$\begin{aligned} \rho \left\| \sum_{i \in J_{max}} \mathbf{Q}_i \right\|_2 + \sum_{i \in J_{max}} \mathbf{Q}_i \bar{\xi} &= \sum_{i \in J_{max}} \mathbf{Q}_i \hat{\xi} \\ &\leq \sum_{i \in K} \mathbf{Q}_i \hat{\xi} \\ &\leq \max_{\xi \in S(K)} \sum_{i \in K} \mathbf{Q}_i \xi \\ &\leq \Pi^*_{B_n} \end{aligned}$$

where the first equality follows from the choice of $\hat{\xi}$. The second inequality is due to $J_{max} \cap K \subseteq K$, while the third inequality is from the fact that $\hat{\xi}$ belongs to $S(K)$ by the definition of K. The last inequality is a direct consequence of (5.29). ■

Although problem (5.30) is easier than problem (5.29), it is still computationally costly when the number of constraints m is large. Notice that (5.30) can also be seen as a integer quadratic program that can be approximated by semidefinite programming and solved using standard quadratic programming tools. We will not elaborate more on this since this falls out of the scope of this thesis.

5.7.1 Numerical experiments

The problem instances are randomly generated following the rules below.

1. Let $m \in \{10, 20, 30, 40, 50\}$, $n \in \{5, 9, 10, 12, 15, 20, 30\}$.
2. Generate the components of $\mathbf{Q}$ uniformly over $[-\mathbf{1}, \mathbf{1}]$.
3. We build four different sizes of pole-sets for each considered hypercube by the circumscribed simplex generation algorithm and the tightening procedure described in Section 5.6. As a result, for a hypercube H_n, (Ω_i) is a monotonic sequence w.r.t. the set inclusion of their convex hulls, i.e., $\Omega_i \preceq_{H_n} \Omega_j$ for all $i > j$. Table 5.1 displays the cardinality of different pole-sets of H_n. The number of vertices of H_n is also provided in the last column.

Table 5.1: The pole-sets of hypercubes

Hypercube	$\lvert\Omega_0\rvert$	$\lvert\Omega_1\rvert$	$\lvert\Omega_2\rvert$	$\lvert\Omega_3\rvert$	#ext.
H_9	10	32	162	387	512
H_{10}	11	36	112	322	1,024
H_{12}	13	44	144	449	4,046
H_{15}	16	56	192	353	32,768
H_{20}	21	76	144	514	1,048,576
H_{30}	31	60	116	432	1,073,741,824

4. As an illustration of multipolar robust approach for smooth convex uncertainty sets, we generate pole-sets of ball B_n as well. In Table 5.2, Ω_0 denotes a first pole-set whose convex hull is a simplex, while Ω_1 is the $2n-$pole-set defined at the end of Section 5.6. Starting from Ω_1 and applying the tightening procedure, we get the pole-sets $\Omega_i, i = 2, 3, 4$ as outputs. The cardinality of pole-sets are shown in Table 5.2.

Table 5.2: The pole-sets of the unit volume ball

Unit ball	$\lvert\Omega_0\rvert$	$\lvert\Omega_1\rvert$	$\lvert\Omega_2\rvert$	$\lvert\Omega_3\rvert$	$\lvert\Omega_4\rvert$
B_5	6	10	118	218	308
B_9	10	18	62	152	352
B_{10}	11	20	72	132	374
B_{12}	13	24	88	164	478

The pole-sets had been readily generated before the solution procedure. Compact formulations (5.27) and (5.28) are modeled by YALMIP [115] and all the problem instances are solved by the Linux version of CPLEX 12.5 with default settings on a Dell E6400 laptop with Intel Core(TM)2 Duo CPU clocked at 2.53 GHz and with 4 GB of RAM. We evaluate our multipolar approach from different measuring: the impact of pole-sets, the impact of the shadow matrix $\mathbf{P}$, and the benefit of adaptability.

The influence of pole-sets

Recall that multipolar robust approach closes to some extend the gap between affine robust value and the fully adjustable value. To test the impact of pole-sets, we fix $\mathbf{P} = \mathbf{I}$. For relatively lower dimensional cases, we report the multipolar robust values w.r.t. different pole-sets, and compute the percentage of the closed gap induced by the multipolar robust approach $\frac{\Pi_{\Xi}(\Omega_0)-\Pi_{\Xi}(\Omega_i)}{\Pi_{\Xi}(\Omega_0)-\Pi^*_{\Xi}} \times 100$. For higher dimensional cases, where FARC can hardly be solved in a reasonable time, we report the multipolar robust values w.r.t. different pole-sets.

Results related to hypercubes are presented in Table 5.3. The closed gap percentages are given within parentheses. Overall, these results appear encouraging as indicated by the closed gaps. Observe that when the uncertainty set is fixed, the multipolar robust values in general get lower as the pole-set Ω gets smaller. Also, we report the computing time for higher dimensional instances associated with hypercube uncertainty set in Table 5.4. While the computational time compared with the affine robust approach scales in magnitude, the complexity of multipolar robust approach is controlled by the choice of pole-sets. In particular, in higher dimensional cases, where fully adjustable robust values are difficult to obtain, the robust multipolar approach brings lower cost (compared with affine approach) in a reasonable time.

Table 5.3: The multipolar robust values with different pole-sets (hypercube uncertainty sets)

(m,n)	static	affine/$\Pi_{H_n}(\Omega_0)$	$\Pi_{H_n}(\Omega_1)$	$\Pi_{H_n}(\Omega_2)$	$\Pi_{H_n}(\Omega_3)$	$\Pi^*_{H_n}$
(10,9)	24.84	12.42	12.18(9.45)	10.55(73.62)	10.16(88.98)	9.88
(10,10)	25.50	12.75	11.53(58.65)	10.96(86.06)	10.70(98.56)	10.67
(10,12)	30.66	15.33	14.63(35.18)	13.71(81.41)	13.43(95.48)	13.34
(20,9)	50.75	25.37	23.82(46.69)	22.08(99.10)	22.06(99.70)	22.05
(20,10)	50.88	25.44	23.56(16.95)	20.74(42.38)	18.58(61.86)	14.35
(20,12)	59.79	29.89	27.54(23.64)	25.40(45.17)	23.58(63.48)	19.95
(10,15)	35.81	17.90	16.91	15.98	15.29	-
(10,20)	50.88	25.44	24.82	24.35	23.33	-
(10,30)	64.32	32.16	31.70	31.14	30.22	-
(20,15)	82.49	41.25	39.36	36.20	34.87	-
(20,20)	99.28	49.64	47.17	45.66	40.80	-
(20,30)	157.20	78.60	77.82	76.83	73.77	-

A sequence of lower bounds can also be generated as stated in Corollary 5.3 of Sec-

Table 5.4: Computing time (in seconds)

(m,n)	static	affine/$\Pi_{H_n}(\Omega_0)$	$\Pi_{H_n}(\Omega_1)$	$\Pi_{H_n}(\Omega_2)$	$\Pi_{H_n}(\Omega_3)$
(10,15)	0.00	0.01	0.20	1.49	6.31
(10,20)	0.00	0.03	0.87	2.66	27.54
(10,30)	0.00	0.04	1.15	4.53	39.12
(20,15)	0.00	0.03	0.68	10.43	34.98
(20,20)	0.00	0.07	4.65	16.41	152.79
(20,30)	0.00	0.14	3.91	15.40	220.48

tion 5.4.3. All we need to do is to generate a sequence of $(\Gamma_i)_{i=0}^{i=3}$ by projecting the pole-sets $(\Omega_i)_{i=0}^{i=3}$ onto the surface of hypercubes. The obtained lower bounds are denoted by $\Pi_{H_n}(\Gamma_i)$. Note that $\operatorname{conv}\Omega' \subseteq \Omega$ does not necessarily lead to $\operatorname{conv}\Gamma' \subseteq \Gamma$ or $\operatorname{conv}\Gamma' \supseteq \Gamma$. Thus it may happen that $\Pi_{H_n}(\Gamma_i) \geq \Pi_{H_n}(\Gamma_{i+1})$. The results are summarized in Table 5.5, where the best lower bound for each problem instance is marked in bold.

Table 5.5: Lower bounds related to hypercubes

(m,n)	$\Pi_{H_n}(\Gamma_0)$	$\Pi_{H_n}(\Gamma_1)$	$\Pi_{H_n}(\Gamma_2)$	$\Pi_{H_n}(\Gamma_3)$	$\Pi^*_{H_n}$
(10,9)	6.88	8.18	9.52	**9.65**	9.88
(10,10)	7.11	8.12	**9.62**	8.34	10.67
(10,12)	**10.50**	8.88	9.48	9.57	13.34
(20,9)	20.08	16.05	18.70	**21.98**	22.05
(20,10)	11.88	11.88	12.44	**12.92**	14.35
(20,12)	14.73	16.65	16.96	**19.07**	19.95
(10,15)	7.62	7.63	8.40	**8.40**	-
(10,20)	7.86	10.20	10.20	**10.36**	-
(10,30)	7.11	9.03	9.03	**9.79**	-
(20,15)	23.49	**25.86**	23.56	23.56	-
(20,20)	15.23	16.08	16.08	**19.61**	-
(20,30)	30.53	31.12	32.53	**33.81**	-

Let us now focus on the ball case. Remember that FARC is intractable in general. However, as shown in Lemma 5.2, we can compute the optimum of FARC by solving problem (5.30) when m is small. We report the robust values obtained by solving multipolar robust counterpart with different pole-sets in Table 5.6. The results may indicate the following. First, the approximate robust values associated with balls appear lower than those associated with hypercubes although the volume and symmetric center of balls and hypercubes are the same. Second, the closed gaps by multipolar robust approach on robust problems with ball uncertainty sets might be less significant than that with hypercube uncertainty sets. As might be expected, larger poles-sets are required for balls compared to hypercubes. Third, despite the limitations, multipolar approach closes around 30% of the optimality gap. In particular, it appears compelling when the number of constraints are large, while the dimension of the uncertainty set is small.

Table 5.6: The multipolar robust values with different pole-sets (ball)

(m,n)	static	$\Pi_{B_n}(\Omega_0)$	$\Pi_{B_n}(\Omega_1)$	$\Pi_{B_n}(\Omega_2)$	$\Pi_{B_n}(\Omega_3)$	$\Pi_{B_n}(\Omega_4)$	$\Pi^*_{B_n}$
(10,9)	17.27	9.43	9.28(11.11)	9.21(16.30)	9.11(23.70)	9.01(31.11)	8.08
(10,10)	16.09	9.21	8.98(23.00)	8.95(26.00)	8.94(27.00)	8.89(32.00)	8.21
(10,12)	19.64	10.93	10.76(20.99)	10.76(20.99)	10.74(23.46)	10.70(28.40)	10.12
(20,9)	35.87	19.86	19.54(20.92)	19.51(22.88)	19.45(26.80)	19.36(32.68)	18.33
(20,10)	33.12	17.19	16.42(18.08)	15.62(36.85)	15.57(38.03)	15.36(42.96)	12.93
(20,12)	39.85	20.93	20.05(17.25)	19.96(19.02)	19.90(20.20)	19.81(21.96)	15.83
(30,5)	19.51	11.00	10.39	9.22	9.14	9.03	-
(40,5)	37.57	21.63	21.11	20.71	20.63	20.55	-
(50,5)	38.14	20.94	20.06	19.33	19.14	19.02	-

The lower bounds obtained in the ball case are reported in Table 5.7. Interestingly, the observed sequences of lower bounds associated with ball B_n are monotonically increasing and their best bounds in general are close to the fully adjustable robust value.

Table 5.7: The lower bounds in the ball case

(m,n)	$\Pi_{B_n}(\Gamma_0)$	$\Pi_{B_n}(\Gamma_1)$	$\Pi_{B_n}(\Gamma_2)$	$\Pi_{B_n}(\Gamma_3)$	$\Pi_{B_n}(\Gamma_4)$	$\Pi^*_{B_n}$
(10,9)	6.60	6.70	6.70	6.71	**7.67**	8.08
(10,10)	6.07	6.28	6.93	7.44	**7.60**	8.21
(10,12)	7.26	7.51	8.01	8.08	**8.44**	10.12
(20,9)	16.04	16.17	16.89	16.89	**16.89**	18.33
(20,10)	12.03	12.03	12.03	12.03	**12.03**	12.93
(20,12)	12.35	13.41	13.67	13.67	**13.67**	15.83
(30,5)	7.48	7.48	8.11	8.11	**8.34**	-
(40,5)	16.54	17.31	19.27	19.27	**19.87**	-
(50,5)	15.53	15.53	17.22	17.22	**17.80**	-

The impact of the shadow matrix

To investigate the impact of the shadow matrix on the robust value of the robust problem, we conduct some experiments on problem instances w.r.t. hypercube uncertainty sets.

The shadow matrices considered here are simply projection matrices on lower subspaces. The results are displayed in Table 5.8, where the uncertainty set is a hypercube and several projections are considered (on H_5, H_7, H_{10}, and H_{12}). The pole-set considered is the set of extreme points of the projected set. As might be expected, the robust value decreases as more information is employed in MRC.

Table 5.8: Impact of the shadow matrix

(m,n)	H_5	H_7	H_{10}	H_{12}
(10,30)	56.95	51.50	45.59	43.29
(10,50)	109.84	105.68	100.69	96.11
(10,70)	162.14	159.52	156.12	154.53
(10,100)	246.98	244.13	239.34	237.43
(20,30)	133.35	126.59	117.63	111.53
(20,50)	233.73	224.82	213.04	203.89
(20,70)	345.56	340.41	328.49	319.79
(20,100)	462.51	453.26	439.71	431.39

The benefit of adaptability

To illustrate the concept of *benefit of adaptability* in the framework of multipolar robust approach, we compute the multipolar robust values of problem (5.26) with different proportions of adjustable variables $\mathbf{v}$. We allow the first $\lfloor \theta m \rfloor$ components of $\mathbf{v}$ to be adaptable to the realization of ξ, while keeping the remaining $m - \lfloor \theta m \rfloor$ variables independent of the realization of ξ, where $\theta \in [0, 1]$. Note that when $\theta = 0$, we get the static case SRC. The results are summarized in Table 5.9 and we emphasize here two observations:

1. For each problem instance, as the adaptability ratio θ increases, the robust value decreases significantly, which is reasonable both in theory and practice.
2. As the adaptability ratio θ increases, the influence of pole-sets on the robust value increases. For example, when the adaptability ratio $\theta = 0.25$, the multipolar robust values of all problem instances remain the same with different pole-sets except instance (20,20). When the adaptability increases, the robust values of more instances improve as the better pole-sets are used, which can be clearly seen when $\theta = 0.75$ and $\theta = 1$.

5.8 Conclusion

In this chapter, we have presented a novel approach to handle uncertainty in optimization problems called the multipolar robust approach, which is based on a set of poles

Table 5.9: The benefit of adaptability (hypercube)

(m,n), Ω_1	$\theta = 0.25$	$\theta = 0.5$	$\theta = 0.75$	$\theta = 1$
(10,9)	23.42	19.88	16.56	12.18
(10,15)	32.64	24.83	21.83	16.91
(10,20)	46.13	37.06	32.78	24.82
(10,30)	59.72	47.14	41.59	31.70
(20,9)	45.01	36.57	28.99	23.82
(20,15)	73.99	58.66	46.80	39.36
(20,20)	85.18	73.46	59.54	47.17
(20,30)	137.78	118.51	99.94	77.82
(m,n), Ω_2	$\theta = 0.25$	$\theta = 0.5$	$\theta = 0.75$	$\theta = 1$
(10,9)	23.42	19.88	15.91	10.55
(10,15)	32.64	24.55	21.21	15.98
(10,20)	46.13	36.86	32.46	24.35
(10,30)	59.72	46.95	41.32	31.14
(20,9)	45.01	36.52	27.24	22.08
(20,15)	73.99	58.16	45.06	36.20
(20,20)	84.98	72.98	58.77	45.66
(20,30)	137.78	117.96	99.13	76.83
(m,n), Ω_3	$\theta = 0.25$	$\theta = 0.5$	$\theta = 0.75$	$\theta = 1$
(10,9)	23.42	19.88	15.88	10.16
(10,15)	32.64	24.54	20.96	15.29
(10,20)	46.13	36.60	31.81	23.33
(10,30)	59.72	46.70	40.74	30.22
(20,9)	45.01	36.52	27.06	22.06
(20,15)	73.99	58.14	44.89	34.87
(20,20)	84.78	72.42	57.23	40.80
(20,30)	137.78	116.78	96.99	73.77

that are used to approximate the fully adjustable policy by a set of associated recourse decisions at poles. The approach generalizes the static approach, the affinely adjustable approach, and the fully adjustable approach, still we can control its complexity by using the concept of the shadow matrix and considering a reasonable number of poles. Several algorithms are proposed for the construction of proper pole-sets for hypercubes and balls. Comprehensive numerical experiments are carried out to evaluate the performance of the proposed approach in terms of the robust values, the complexity, and the benefit of adaptability. In general, the results appear encouraging.

It would be interesting to investigate further the performance of the multipolar robust approach on other problems. A systematic study of good approximations of convex bodies by enclosing polytopes with a limited number of extreme points should help to alleviate overconservatism and get closer to the optimal fully adaptable robust value. One can also put more focus on the approximation of convex bodies from inside using, for example, maximum volume inscribed polytopes to get better lower bounds of the fully adjustable robust value.

While the approach was proposed in the context of a two-stage optimization problem, it can be adapted to multistage optimization. Multipolar decision rules can also be considered in stochastic programming. The multipolar approach might also be combined with finite adaptability or multi-static robustness by partitioning the uncertainty set into several subsets and considering some multipolar decision rules for each subset.

Appendix: the derivation of (5.9)

We derive the compact formulation (5.9) of Section 5.4.1 w.r.t. an ellipsoidal uncertainty set defined by $\Xi := \{\xi : \|\mathbf{F}\xi\|_2 \leq 1\}$.

For each i^{th} constraint, MRC requires the optimum of the following problem non-positive.

$$\begin{array}{lll}
\max\limits_{\xi,\mathbf{s},\boldsymbol{\lambda}\geq 0} & \mathbf{U}_i\mathbf{u} - b_i + \sum\limits_{\omega\in\Omega} \lambda^{\xi}_{\omega}\mathbf{V}_i\mathbf{v}_{\omega} & \\
\text{s.t.} & \|\mathbf{s}\|_2 \leq 1, & \\
 & \mathbf{F}\xi = \mathbf{s}, & \boldsymbol{\eta}_i \in \mathbb{R}^{n_q} \\
 & \sum\limits_{\omega\in\Omega} \lambda^{\xi}_{\omega}\omega = \mathbf{P}\xi, & \boldsymbol{\sigma}_i \in \mathbb{R}^{n_0} \\
 & \sum\limits_{\omega\in\Omega} \lambda^{\xi}_{\omega} = 1, & \tau_i \in \mathbb{R}
\end{array}$$

where $\xi \equiv [\mathbf{U}, \mathbf{b}]$ and $\boldsymbol{\eta}_i, \boldsymbol{\sigma}_i, \tau_i$ are dual multipliers corresponding to each group of constraints. Consider the corresponding Lagrangian

$$\begin{aligned}
\mathcal{L}\left(\lambda, \xi, \mathbf{s}, \boldsymbol{\eta}_i, \tau_i, \boldsymbol{\sigma}_i\right) = \mathbf{U}_i\mathbf{u} - b_i + \sum_{\omega\in\Omega} \lambda^{\xi}_{\omega}\mathbf{V}_i\mathbf{v}_{\omega} + \boldsymbol{\eta}_i^T\left(\mathbf{s} - \mathbf{F}\xi\right) \\
+ \boldsymbol{\sigma}_i^T\left(\sum_{\omega\in\Omega} \lambda^{\xi}_{\omega}\omega - \mathbf{P}\xi\right) + \tau_i\left(\sum_{\omega\in\Omega} \lambda^{\xi}_{\omega} - 1\right).
\end{aligned}$$

The dual function is then $\max\limits_{\lambda,\xi,\|\mathbf{s}\|_2\leq 1} \mathcal{L}\left(\boldsymbol{\lambda}, \xi, \boldsymbol{\eta}_i, \tau_i, \boldsymbol{\sigma}_i\right)$. Setting the derivative w.r.t. $\boldsymbol{\lambda}, \xi$ leads to the dual constrains

$$\begin{aligned}
\mathbf{V}_i\mathbf{v}_{\omega} + \tau_i - \omega^T\boldsymbol{\sigma}_i \leq 0, \\
\mathbf{F}^T\eta_i - \mathbf{L}_i = \mathbf{0},
\end{aligned} \tag{5.31}$$

where $\mathbf{P} = [\mathbf{P}_1, \ldots, \mathbf{P}_k, \ldots, \mathbf{P}_m]$, $\mathbf{L}_i = \Big(\mathbf{L}_{i1}, \ldots, \mathbf{L}_{im}\Big)$, $\mathbf{L}_{ij} = \delta_{ij}\Big(\mathbf{u}, -1\Big) + \mathbf{P}_j^T \boldsymbol{\sigma}_i,\ j = 1, \ldots, m.$ The dual objective is

$$\min_{\boldsymbol{\eta}_i, \tau_i, \boldsymbol{\sigma}_i} \quad \|\boldsymbol{\eta}_i\|_2 - \tau_i. \tag{5.32}$$

By duality, the optimum of the above dual problem is equal to the optimum of the problem. Thus restricting the non-positivity of the primal optimum can be equivalently represented as

$$\begin{aligned} \|\boldsymbol{\eta}_i\|_2 + \mathbf{V}_i \mathbf{v}_\omega - \omega^T \boldsymbol{\sigma}_i \leq 0, \\ \mathbf{F}^T \eta_i - \mathbf{L}_i = \mathbf{0}, \\ \boldsymbol{\eta}_i \in \mathbb{R}^{n_q}, \boldsymbol{\sigma}_i \in \mathbb{R}^{n_0}. \end{aligned} \tag{5.33}$$

Chapter 6

Conclusions

To conclude this thesis, we briefly summarize our main contributions, and outline some directions of future research.

6.1 Evaluation

This thesis is motivated by a mapping problem where a set of virtual machines need to be assigned to servers with multiple 0-1 quadratic constraints. Two main challenges exist. First, to the best of our knowledge, efficient exact solution procedures or approximation schemes with performance guarantees of bounds do not appear in the literature. Second, the demands of computing resources and networking resources are highly uncertain. Few investigations have been conducted to incorporate uncertainty properly in the context of robust optimization. It is well-known that both problems are difficult. Thus this thesis aims at proposing and applying novel relaxation techniques to handle MIQCPs and robust programs.

In Chapter 3, an exact model dealing with static mapping procedure is provided. Relaxations involving certain standard techniques such as RLT inequalities are proposed. To further strengthen the formulation, two types of strong cuts are proposed by exploiting the problem structure, which numerically prove quite effective. Then, we have tried a number of decompositions based on Lagrangian relaxations with a hope that the Lagrangian dual bound is much stronger (compared with the bound provided by the continuous relaxation of a strong formulation) while the solution procedure is efficient. A request based decomposition scheme is then discussed in Section 3.4. We also provided a geometric result on the strength of the decomposition scheme, which implies a novel hierarchy of relaxations by grouping the set of requests. Our numerical results show that the proposed hierarchy can handle relatively larger problem instances much more efficiently than the standard branch-and-bound routines of CPLEX12.6.3. Finally, we proposed a reformulation exploiting symmetries of virtual requests, which scales the proposed algorithms greatly.

Chapter 4 discusses a couple of relaxation approaches for bilinear optimization problems over a hypercube from the perspective of convex and concave envelopes. A connection between quadratic convexification techniques and Shor's relaxation strengthened by

RLT inequalities was established. We showed that two seemingly different formulations are in fact same, which generalizes a related result from Burer and Letchford [50]. Then we proposed a novel relaxation approach by considering a predefined tool to approximate the convex and concave envelope of f over a hypercube. This approach leads to a linear reformulation and we also established its connections with various inequalities and the RLT method proposed in [2].

To address the issues of uncertainty, we proposed a new tractable robust paradigm called multipolar robust optimization in Chapter 5. It provides a new way to model the recourse action for a fairly general linear program with uncertain parameters. It generalizes notions of static robustness, affinely adjustable robustness, fully adjustable robustness and fills the gaps in-between. The result of [34] stating that affine rules are optimal when the uncertainty set is simplex in the special case of right-hand-side uncertainty is generalized as a consequence of the multipolar approach. Moreover, we show that the multipolar robust counterpart is tractable by either a cut generation procedure or a compact formulation. Further, we prove that the multipolar approach can generate a sequence of upper bounds and a sequence of lower bounds at the same time and both sequences converge to the robust value of fully adjustable robust counterpart under some mild assumptions. The multipolar approach is based on some tools related to the uncertainty set, that we call as pole-sets. For their construction, we start with a simplex and then compute the best homothetic transformation of this simplex to allow it to enclose a given convex set. An efficient algorithm is proposed to compute such homothetic set. Several numerical experiments are carried out showing the advantages of the proposed robustness framework.

6.2 Perspectives

As future research directions, it would be interesting to investigate the following topics, which have been left due to the lack of time.

Regarding the mapping problem discussed in Chapter 3, we have the following perspectives. First, we may integrate the proposed decomposition hierarchy into the branch-and-bound procedure of CPLEX with a hope of accelerating the solution procedure further. We might also consider combining the symmetry-induced model and decomposition hierarchy together to deal with much larger problem instances. Second, it might be worthwhile developing a spatial branch-and-bound scheme to solve the extended mapping model discussed in Section 3.6. We can also consider certain convex and concave envelops over some polytopes associated with assignment constraints and location constraints. Third, a future project could be applying the multipolar robust optimization paradigm to the mapping problem with uncertain parameters associated with virtual requests.

In Chapter 4, we have proposed a perspective on approximating the convex and concave envelopes using the idea of pole-set, which covers an n-dimensional hypercube with much less extreme points. A valuable direction might be to combine the idea of pole-set and certain valid inequalities for QPB polytope to derive strong estimators for general quadratic forms rather than bilinear forms. And, it might be interesting to integrate ideas of cover-based estimators and pole-set based estimators in order to find the most expressive linear formulation to approximate the estimators of bilinear functions.

To extend the work of multipolar robust optimization, we may investigate the approximation of convex bodies from inside using, for example, maximum volume inscribed polytopes, to get better lower bounds of the fully adjustable robust value. We can also extend our work to model recourse actions that are binary or general integer variables.

All these topics will be pursued in future research.

Bibliography

[1] Acemoglu, D. and Ozdaglar, A. (2010). Opinion dynamics and learning in social networks. *Dynamic Games and Applications*, 1(1):3–49.

[2] Adams, W. P. and Sherali, H. D. (2005a). A hierarchy of relaxations leading to the convex hull representation for general discrete optimization problems. *Annals OR*, 140(1):21–47.

[3] Adams, W. P. and Sherali, H. D. (2005b). A hierarchy of relaxations leading to the convex hull representation for general discrete optimization problems. *Annals OR*, 140(1):21–47.

[4] Adjiman, C., Dallwig, S., Floudas, C., and Neumaier, A. (1998). A global optimization method, αbb, for general twice-differentiable constrained nlps — i. theoretical advances. *Computers & Chemical Engineering*, 22(9):1137 – 1158.

[5] Agra, A., Santos, M. C., Nace, D., and Poss, M. (2016). A dynamic programming approach for a class of robust optimization problems. *SIAM Journal on Optimization*, 26(3):1799–1823.

[6] Al-Khayyal, F. A. and Falk, J. E. (1983). Jointly Constrained Biconvex Programming. *Mathematics of Operations Research*, 8:273–286.

[7] Amaldi, E., Coniglio, S., Koster, A. M., and Tieves, M. (2016). On the computational complexity of the virtual network embedding problem. *Electronic Notes in Discrete Mathematics*, 52:213 – 220. INOC 2015 – 7th International Network Optimization Conference.

[8] Ameur, W. B. and Neto, J. (2008). A geometric characterization of "optimality-equivalent" relaxations. *Journal of Global Optimization*, 42(4):533–547.

[9] Androulakis, I. P., Maranas, C. D., and Floudas, C. A. (1995). αbb: A global optimization method for general constrained nonconvex problems. *Journal of Global Optimization*, 7(4):337–363.

[10] Anstreicher, K. (2009). Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. *Journal of Global Optimization*, 43(2-3):471–484.

[11] Anstreicher, K. (2012). On convex relaxations for quadratically constrained quadratic programming. *Mathematical Programming*, 136(2):233–251.

[12] Anstreicher, K. M. and Burer, S. (2010). Computable representations for convex hulls of low-dimensional quadratic forms. *Mathematical programming*, 124(1-2):33–43.

[13] Audet, C., Brimberg, J., Hansen, P., Digabel, S. L., and Mladenović, N. (2004). Pooling problem: Alternate formulations and solution methods. *Management science*, 50(6):761–776.

[14] Babonneau, F., Vial, J.-P., Klopfenstein, O., and Ouorou, A. (2013). Robust capacity assignment solutions for telecommunications networks with uncertain demands. *Networks*, 62(4):255–272.

[15] Balas, E. (1979). Disjunctive programming. *Annals of Discrete Mathematics*, 5:3–51.

[16] Balas, E. (1998). Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Applied Mathematics*, 89(1):3–44.

[17] Balas, E., Ceria, S., and Cornuéjols, G. (1993). A lift-and-project cutting plane algorithm for mixed 0–1 programs. *Mathematical programming*, 58(1-3):295–324.

[18] Bao, X., Sahinidis, N. V., and Tawarmalani, M. (2009). Multiterm polyhedral relaxations for nonconvex, quadratically constrained quadratic programs. *Optimization Methods & Software*, 24(4-5):485–504.

[19] Beale, E. M. L. and Tomlin, J. A. (1970). Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables. *OR*, 69(447-454):99.

[20] Belotti, P. (2012). Disjunctive cuts for nonconvex minlp. In *Mixed Integer Nonlinear Programming*, pages 117–144. Springer.

[21] Belotti, P., Kirches, C., Leyffer, S., Linderoth, J., Luedtke, J., and Mahajan, A. (2013). Mixed-integer nonlinear optimization. *Acta Numerica*, 22:1–131.

[22] Bemporad, A., Borrelli, F., and Morari, M. (2003). Min-max control of constrained uncertain discrete-time linear systems. *Automatic Control, IEEE Transactions on*, 48(9):1600–1606.

[23] Ben-Ameur, W. (2007). Between fully dynamic routing and robust stable routing. In *Design and Reliable Communication Networks, 2007. DRCN 2007. 6th International Workshop on*, pages 1–6. IEEE.

[24] Ben-Ameur, W. and Kerivin, H. (2005). Routing of uncertain traffic demands. *Optimization and Engineering*, 6(3):283–313.

[25] Ben-Ameur, W. and Neto, J. (2007). Acceleration of cutting-plane and column generation algorithms: Applications to network design. *Networks*, 49(1):3–17.

[26] Ben-Ameur, W., Ouorou, A., Wang, G., and Żotkiewicz, M. (2016). Multipolar robust optimization. *arXiv preprint arXiv:1604.01813*.

[27] Ben-Ameur, W., Ouorou, A., and Zotkiewicz, M. (2012). Robust routing in communication networks. *Progress in combinatorial optimization*, pages 353–390.

[28] Ben-Ameur, W. and Żotkiewicz, M. (2011). Robust routing and optimal partitioning of a traffic demand polytope. *International Transactions in Operational Research*, 18(3):307–333.

[29] Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. (2009). *Robust optimization*. Princeton University Press.

[30] Ben-Tal, A., Goryashko, A., Guslitzer, E., and Nemirovski, A. (2004). Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376.

[31] Ben-Tal, A. and Nemirovski, A. (1998). Robust convex optimization. *Mathematics of operations research*, 23(4):769–805.

[32] Bertsimas, D., Brown, D. B., and Caramanis, C. (2011a). Theory and applications of robust optimization. *SIAM review*, 53(3):464–501.

[33] Bertsimas, D. and Caramanis, C. (2010). Finite adaptability in multistage linear optimization. *Automatic Control, IEEE Transactions on*, 55(12):2751–2766.

[34] Bertsimas, D. and Goyal, V. (2012). On the power and limitations of affine policies in two-stage adaptive optimization. *Mathematical programming*, 134(2):491–531.

[35] Bertsimas, D., Goyal, V., and Sun, X. A. (2011b). A geometric characterization of the power of finite adaptability in multistage stochastic and adaptive optimization. *Mathematics of Operations Research*, 36(1):24–54.

[36] Bertsimas, D., Iancu, D. A., and Parrilo, P. A. (2010). Optimality of affine policies in multistage robust optimization. *Mathematics of Operations Research*, 35(2):363–394.

[37] Bertsimas, D., Iancu, D. A., and Parrilo, P. A. (2011c). A hierarchy of near-optimal policies for multistage adaptive optimization. *Automatic Control, IEEE Transactions on*, 56(12):2809–2824.

[38] Billionnet, A., Elloumi, S., and Lambert, A. (2012). Extending the QCR method to the case of general mixed integer programs. *Mathematical Programming*, 131(1):381–401.

[39] Billionnet, A., Elloumi, S., and Lambert, A. (2015). Exact quadratic convex reformulations of mixed-integer quadratically constrained problems. *Mathematical Programming*, pages 1–32.

[40] Billionnet, A., Elloumi, S., and Plateau, M.-C. (2009). Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: The qcr method. *Discrete Applied Mathematics*, 157(6):1185–1197.

[41] Birge, J. R. and Louveaux, F. (2011). *Introduction to stochastic programming*. Springer Science & Business Media.

[42] Bompadre, A. and Mitsos, A. (2012). Convergence rate of mccormick relaxations. *Journal of Global Optimization*, 52(1):1–28.

[43] Bonami, P., Kilinç, M., and Linderoth, J. (2012). Algorithms and software for convex mixed integer nonlinear programs. In Lee, J. and Leyffer, S., editors, *Mixed Integer Nonlinear Programming*, volume 154 of *The IMA Volumes in Mathematics and its Applications*, pages 1–39. Springer New York.

[44] Böröczky, K. (2000). Approximation of general smooth convex bodies. *Advances in Mathematics*, 153(2):325–341.

[45] Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

[46] Bronshteyn, E. M. and Ivanov, L. (1975). The approximation of convex sets by polyhedra. *Siberian Mathematical Journal*, 16(5):852–853.

[47] Bronstein, E. M. (2008). Approximation of convex sets by polytopes. *Journal of Mathematical Sciences*, 153(6):727–762.

[48] Buchheim, C. and Wiegele, A. (2013). Semidefinite relaxations for non-convex quadratic mixed-integer programming. *Mathematical Programming*, 141(1-2):435–452.

[49] Burer, S. (2009). On the copositive representation of binary and continuous nonconvex quadratic programs. *Mathematical Programming*, 120(2):479–495.

[50] Burer, S. and Letchford, A. N. (2009). On nonconvex quadratic programming with box constraints. *SIAM Journal on Optimization*, 20(2):1073–1089.

[51] Burer, S. and Letchford, A. N. (2012). Non-convex mixed-integer nonlinear programming: A survey. *Surveys in Operations Research and Management Science*, 17(2):97 – 106.

[52] Burer, S. and Saxena, A. (2012). The milp road to miqcp. In *Mixed Integer Nonlinear Programming*, pages 373–405. Springer.

[53] Cadoux, F. (2010). Computing deep facet-defining disjunctive cuts for mixed-integer programming. *Mathematical Programming*, 122(2):197–223.

[54] Chaisiri, S., Lee, B.-S., and Niyato, D. (2009). Optimal virtual machine placement across multiple cloud providers. In *Services Computing Conference, 2009. APSCC 2009. IEEE Asia-Pacific*, pages 103–110.

[55] Charnes, A. and Cooper, W. W. (1959). Chance-constrained programming. *Management science*, 6(1):73–79.

[56] Chekuri, C., Shepherd, F. B., Oriolo, G., and Scutellà, M. G. (2007). Hardness of robust network design. *Networks*, 50(1):50–54.

[57] Chen, X., Sim, M., Sun, P., and Zhang, J. (2008). A linear decision-based approximation approach to stochastic programming. *Operations Research*, 56(2):344–357.

[58] Chen, X. and Zhang, Y. (2009). Uncertain linear programs: Extended affinely adjustable robust counterparts. *Operations Research*, 57(6):1469–1482.

[59] Chowdhury, N., Rahman, M., and Boutaba, R. (2009). Virtual network embedding with coordinated node and link mapping. In *INFOCOM 2009, IEEE*, pages 783–791.

[60] Dantzig, G. B. (1955). Linear programming under uncertainty. *Management science*, 1(3-4):197–206.

[61] Deza, M. and Laurent, M. (1997). *Geometry of cuts and metric embeddings*. Springer-Verlag, New York.

[62] Dickinson, P. J. C. and Gijben, L. (2014). On the computational complexity of membership problems for the completely positive cone and its dual. *Computational Optimization and Applications*, 57(2):403–415.

[63] Dudley, R. M. (1974). Metric entropy of some classes of sets with differentiable boundaries. *Journal of Approximation Theory*, 10(3):227–236.

[64] Duffield, N. G., Goyal, P., Greenberg, A., Mishra, P., Ramakrishnan, K. K., and van der Merive, J. E. (1999). A flexible model for resource management in virtual private networks. In *ACM SIGCOMM Computer Communication Review*, volume 29, pages 95–108. ACM.

[65] Dür, M. (2010). Copositive programming–a survey. In *Recent advances in optimization and its applications in engineering*, pages 3–20. Springer.

[66] Duran, M. A. and Grossmann, I. E. (1986). An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical programming*, 36(3):307–339.

[67] EI-Ghaoui, L. and Lebret, H. (1997). Robust solutions to least-square problems to uncertain data matrices. *Sima Journal on Matrix Analysis and Applications*, 18:1035–1064.

[68] Espling, D., Larsson, L., Li, W., Tordsson, J., and Elmroth, E. Modeling and placement of structured cloud services. *Submitted to IEEE Transactions on Network and Service Management*.

[69] Fingerhut, J. A., Suri, S., and Turner, J. S. (1997). Designing least-cost nonblocking broadband networks. *Journal of Algorithms*, 24(2):287–309.

[70] Foulds, L., Haugland, D., and Jörnsten, K. (1992). A bilinear approach to the pooling problem†. *Optimization*, 24(1-2):165–180.

[71] Garstka, S. J. and Wets, R. J.-B. (1974). On decision rules in stochastic programming. *Mathematical Programming*, 7(1):117–143.

[72] Geoffrion, A. (1972). Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260.

[73] Ghaoui, L. E., Oks, M., and Oustry, F. (2003). Worst-case value-at-risk and robust portfolio optimization: A conic programming approach. *Operations Research*, 51(4):543–556.

[74] Glover, F. (1975). Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22(4):455–460.

[75] Goh, J. and Sim, M. (2010). Distributionally robust optimization and its tractable approximations. *Operations research*, 58(4-part-1):902–917.

[76] Gritzmann, P. and Klee, V. (1994). On the complexity of some basic problems in computational convexity. In *Polytopes: Abstract, Convex and Computational*, pages 373–466. Springer.

[77] Grossmann, I. E. (2002). Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and engineering*, 3(3):227–252.

[78] Grossmann, I. E. and Lee, S. (2003). Generalized convex disjunctive programming: Nonlinear convex hull relaxation. *Computational optimization and applications*, 26(1):83–100.

[79] Grossmann, I. E. and Ruiz, J. P. (2012). Generalized disjunctive programming: A framework for formulation and alternative algorithms for minlp optimization. In *Mixed Integer Nonlinear Programming*, pages 93–115. Springer.

[80] Grötschel, M., Lovász, L., and Schrijver, A. (2012). *Geometric algorithms and combinatorial optimization*, volume 2. Springer Science & Business Media.

[81] Hahn, P. and Grant, T. (1998). Lower bounds for the quadratic assignment problem based upon a dual formulation. *Operations Research*, 46(6):912–922.

[82] Hammer, P. L. and Rubin, A. A. (1970). Some remarks on quadratic programming with 0-1 variables. *RAIRO - Operations Research - Recherche Opérationnelle*, 4(V3):67–79.

[83] Headquarters, A. (2007). Cisco data center infrastructure 2.5 design guide.

[84] Hiriart-Urruty, J.-B. and Seeger, A. (2010). A variational approach to copositive matrices. *SIAM review*, 52(4):593–629.

[85] Horst, R. and Thoai, N. V. (1999). Dc programming: overview. *Journal of Optimization Theory and Applications*, 103(1):1–43.

[86] Houidi, I., Louati, W., Ben Ameur, W., and Zeghlache, D. (2011). Virtual network provisioning across multiple substrate networks. *Comput. Netw.*, 55(4):1011–1023.

[87] Iancu, D. A., Sharma, M., and Sviridenko, M. (2013). Supermodularity and affine policies in dynamic robust optimization. *Operations Research*, 61(4):941–956.

[88] Jeroslow, R. (1973). There cannot be any algorithm for integer programming with quadratic constraints. *Operations Research*, 21(1):221–224.

[89] Kaibel, V. (1997). *Polyhedral combinatorics of the quadratic assignment problem.* na.

[90] Kall, P. and Wallace, S. (1994). *Stochastic programming.* Wiley-Interscience series in systems and optimization. Wiley.

[91] Kamenov., G. (1996). Algorithm of approaching polyhedrons. *Zh. Vychisl. Mat. Fiz.*, 36:134–147.

[92] Karp, R. M. (1972). Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer.

[93] Karve, A., Kimbrel, T., Pacifici, G., Spreitzer, M., Steinder, M., Sviridenko, M., and Tantawi, A. (2006). Dynamic placement for clustered web applications. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, pages 595–604, New York, NY, USA. ACM.

[94] Kerivin, H. and Ben-Ameur, W. (2003). New economical virtual private networks. *Communications of the ACM*, 46(6):69–73.

[95] Kim, S. and Kojima, M. (2001). Second order cone programming relaxation of nonconvex quadratic optimization problems. *Optimization Methods and Software*, 15(3-4):201–224.

[96] Klee, V. (1986). Facet-centroids and volume minimization. *Studia Scientiarum Mathematicarum Hungarica*, 21:143–147.

[97] Klopfenstein, O. and Nace, D. (2008). A robust approach to the chance-constrained knapsack problem. *Operations Research Letters*, 36(5):628–632.

[98] Kouvelis, P. and Yu, G. (2013). *Robust discrete optimization and its applications*, volume 14. Springer Science & Business Media.

[99] Kuhn, D., Wiesemann, W., and Georghiou, A. (2011). Primal and dual linear decision rules in stochastic and robust optimization. *Mathematical Programming*, 130(1):177–209.

[100] Lassak, M. (1999). Parallelotopes of maximum volume in a simplex. *Discrete & Computational Geometry*, 21(3):449–462.

[101] Lasserre, J. B. (2001). Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817.

[102] Lasserre, J. B. (2002). Semidefinite programming vs. lp relaxations for polynomial programming. *Mathematics of operations research*, 27(2):347–360.

[103] Lasserre, J. B. (2009). *Moments, positive polynomials and their applications*, volume 1. World Scientific.

[104] Lasserre, J. B. (2011). On convex optimization without convex representation. *Optimization Letters*, 5(4):549–556.

[105] Lasserre, J. B. and Thanh, T. P. (2013). Convex underestimators of polynomials. *Journal of Global Optimization*, 56(1):1–25.

[106] Laurent, M. (2009). Sums of squares, moment matrices and optimization over polynomials. *Emerging applications of algebraic geometry*, pages 157–270.

[107] Lemaréchal, C. and Oustry, F. (1999). *Semidefinite relaxations and Lagrangian duality with application to combinatorial optimization*. PhD thesis, INRIA.

[108] Lemaréchal, C. and Renaud, A. (2001). A geometric study of duality gaps, with applications. *Mathematical Programming*, 90(3):399–427.

[109] Lemaréchal, C. (2001). Lagrangian relaxation. In Jünger, M. and Naddef, D., editors, *Computational Combinatorial Optimization*, volume 2241 of *Lecture Notes in Computer Science*, pages 112–156. Springer Berlin Heidelberg.

[110] Liberti, L. (2009). Reformulations in mathematical programming: Definitions and systematics. *RAIRO-Operations Research*, 43(1):55–85.

[111] Liberti, L. and Marinelli, F. (2014). Mathematical programming: Turing completeness and applications to software analysis. *Journal of Combinatorial Optimization*, 28(1):82–104.

[112] Liberti, L. and Pantelides, C. (2006). An exact reformulation algorithm for large nonconvex nlps involving bilinear terms. *Journal of Global Optimization*, 36(2):161–189.

[113] Linderoth, J. (2005). A simplicial branch-and-bound algorithm for solving quadratically constrained quadratic programs. *Mathematical Programming*, 103(2):251–282.

[114] Locatelli, M. and Schoen, F. (2014). On convex envelopes for bivariate functions over polytopes. *Mathematical Programming*, 144(1-2):65–91.

[115] Lofberg, J. (2004). Yalmip: A toolbox for modeling and optimization in matlab. In *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pages 284–289.

[116] Luedtke, J., Namazifar, M., and Linderoth, J. (2012). Some results on the strength of relaxations of multilinear functions. *Mathematical Programming*, 136(2):325–351.

[117] Malick, J. (2007). The spherical constraint in boolean quadratic programs. *Journal of Global Optimization*, 39(4):609–622.

[118] Maxfield, J. E. and Minc, H. (1962). On the matrix equation x'x= a. *Proceedings of the Edinburgh Mathematical Society (Series 2)*, 13(02):125–129.

[119] McCormick, G. (1976). Computability of global solutions to factorable nonconvex programs: Part i-convex underestimating problems. *Mathematical Programming*, 10(1):147–175.

[120] Mechtri, M. (2014). *Virtual networked infrastructure provisioning in distributed cloud environments*. PhD thesis, Institut National des Télécommunications.

[121] Meier, J. F., Clausen, U., Rostami, B., and Buchheim, C. (2016). A compact linearisation of euclidean single allocation hub location problems. *Electronic Notes in Discrete Mathematics*, 52:37 – 44. INOC 2015 – 7th International Network Optimization Conference.

[122] Meng, X., Pappas, V., and Zhang, L. (2010). Improving the scalability of data center networks with traffic-aware virtual machine placement. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE.

[123] Meyer, C. A. and Floudas, C. A. (2005). Convex envelopes for edge-concave functions. *Mathematical Programming*, 103(2):207–224.

[124] Minoux, M. (2010). Robust network optimization under polyhedral demand uncertainty is np-hard. *Discrete Applied Mathematics*, 158(5):597–603.

[125] Misener, R., Thompson, J. P., and Floudas, C. A. (2011). Apogee: Global optimization of standard, generalized, and extended pooling problems via linear and logarithmic partitioning schemes. *Computers & Chemical Engineering*, 35(5):876–892.

[126] Murat Afsar, H., Artigues, C., Bourreau, E., and Kedad-Sidhoum, S. (2016). Machine reassignment problem: the roadef/euro challenge 2012. *Annals of Operations Research*, 242(1):1–17.

[127] Murty, K. G. and Kabadi, S. N. (1987). Some np-complete problems in quadratic and nonlinear programming. *Mathematical programming*, 39(2):117–129.

[128] Nevskii, M. (2011). Properties of axial diameters of a simplex. *Discrete & Computational Geometry*, 46(2):301–312.

[129] Orlowski, S., Pióro, M., Tomaszewski, A., and Wessäly, R. (2007). SNDlib 1.0–Survivable Network Design Library. In *Proceedings of the 3rd International Network Optimization Conference (INOC 2007), Spa, Belgium.* http://sndlib.zib.de, extended version accepted in Networks, 2009.

[130] Ouorou, A. (2013). Tractable approximations to a robust capacity assignment model in telecommunications under demand uncertainty. *Computers & Operations Research*, 40(1):318–327.

[131] Ouorou, A. and Vial, J.-P. (2007). A model for robust capacity planning for telecommunications networks under demand uncertainty. In *2007 6th International Workshop on Design and Reliable Communication Networks*.

[132] Padberg, M. (1989). The boolean quadric polytope: some characteristics, facets and relatives. *Mathematical programming*, 45(1-3):139–172.

[133] Papagianni, C., Leivadeas, A., Papavassiliou, S., Maglaris, V., Cervello-Pastor, C., and Monje, A. (2013). On the optimal allocation of virtual resources in cloud computing networks. *Computers, IEEE Transactions on*, 62(6):1060–1071.

[134] Parks, H. R. and Wills, D. C. (2002). An elementary calculation of the dihedral angle of the regular n-simplex. *The American mathematical monthly*, 109(8):756–758.

[135] Parrilo, P. A. (2000). *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization.* PhD thesis, Citeseer.

[136] Parrilo, P. A. (2003). Semidefinite programming relaxations for semialgebraic problems. *Mathematical programming*, 96(2):293–320.

[137] Putinar, M. (1993). Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, 42(3):969–984.

[138] Quesada, I. and Grossmann, I. E. (1992). An lp/nlp based branch and bound algorithm for convex minlp optimization problems. *Computers & chemical engineering*, 16(10-11):937–947.

[139] Raghavan, P. and Tompson, C. (1987). Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374.

[140] Rikun, A. D. (1997). A convex envelope formula for multilinear functions. *Journal of Global Optimization*, 10(4):425–437.

[141] Rockafellar, R. T. (1993). Lagrange multipliers and optimality. *SIAM review*, 35(2):183–238.

[142] Saito, H., Fujie, T., Matsui, T., and Matuura, S. (2009). A study of the quadratic semi-assignment polytope. *Discrete Optimization*, 6(1):37–50.

[143] Saxena, A., Bonami, P., and Lee, J. (2010). Convex relaxations of non-convex mixed integer quadratically constrained programs: extended formulations. *Mathematical Programming*, 124(1-2):383–411.

[144] Saxena, A., Bonami, P., and Lee, J. (2011). Convex relaxations of non-convex mixed integer quadratically constrained programs: projected formulations. *Mathematical Programming*, 130(2):359–413.

[145] Sherali, H. and Adams, W. (1990). A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3):411–430.

[146] Sherali, H. D. (1997). Convex envelopes of multilinear functions over a unit hypercube and over special discrete sets. *Acta mathematica vietnamica*, 22(1):245–270.

[147] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905.

[148] Skjäl, A., Westerlund, T., Misener, R., and Floudas, C. (2012). A generalization of the classical αbb convex underestimation via diagonal and nondiagonal quadratic terms. *Journal of Optimization Theory and Applications*, 154(2):462–490.

[149] Soyster, A. L. (1973). Technical note—convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations research*, 21(5):1154–1157.

[150] Tawarmalani, M., Richard, J.-P. P., and Xiong, C. (2013). Explicit convex and concave envelopes through polyhedral subdivisions. *Mathematical Programming*, 138(1):531–577.

[151] Vandenberghe, L. and Boyd, S. (1996). Semidefinite programming. *SIAM review*, 38(1):49–95.

[152] Wächter, A. and Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57.

[153] Wang, G., Ben-Ameur, W., Neto, J., and Ouorou, A. (2016). Optimal mapping of cloud virtual machines. *Electronic Notes in Discrete Mathematics*, 52:93 – 100. INOC 2015 – 7th International Network Optimization Conference.

[154] Wang, G. and Ng, T. E. (2010). The impact of virtualization on network performance of amazon ec2 data center. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE.

[155] Zhou, Y. and Suri, S. (2002). Algorithms for a minimum volume enclosing simplex in three dimensions. *SIAM Journal on Computing*, 31(5):1339–1357.

[156] Zotkiewicz, M. and Ben-Ameur, W. (2009). More adaptive robust stable routing. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1–6. IEEE.

END