INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2021IPPAX108

Thèse de doctorat

MINOA
MIXED-INTEGER NON-LINEAR OPTIMISATION: ALGORITHMS AND APPLICATIONS

ÉCOLE POLYTECHNIQUE

IP PARIS

Bilevel optimization and applications

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à l'École Polytechnique

École doctorale n°626 École doctorale de l'Institut Polytechnique de Paris (ED IP Paris)
Spécialité de doctorat : Informatique

Thèse présentée et soutenue à Palaiseau, le 13.12.2021, par

MARTINA CERULLI

Composition du Jury :

Luce Brotcorne Directeur de Recherche, INRIA Lille-Nord Europe	President
Martine Labbé Retired Full Professor, Université Libre de Bruxelles (Faculty of Sciences)	Rapporteur
Ivana Ljubic Full Professor, ESSEC Business School (Department of Information Systems, Decision Sciences and Statistics)	Rapporteur
Fabio Furini Associate Professor, Sapienza Università di Roma (DIAG)	Examinateur
Leo Liberti Directeur de Recherche, CNRS - École Polytechnique (LIX)	Directeur de thèse
Claudia D'Ambrosio Directeur de Recherche, CNRS - École Polytechnique (LIX)	Co-directeur de thèse

Acknowledgments

This PhD thesis is part of a project funded by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement MINOA No. 764759. I do want to acknowledge the MINOA consortium not only for having provided funding for my PhD, but for having offered me a unique and unforgettable doctoral training experience.

I owe my greatest gratitude to my thesis advisors, Profs. Leo Liberti and Claudia D'Ambrosio. It has been a privilege learning from them, and their enlightening advice has been priceless for me. All the stimulating mathematical discussions we had are the foundations on which this whole thesis lies. They allowed me to follow my path in the world of research on my own, being always ready to guide me when my compass got mad. They taught me what one should learn during the PhD besides pure mathematics, and made sure I didn't miss the right events, giving me uncountably many possibilities to increase my network of scientific colleagues. I am very grateful to Prof. Leo Liberti for his trustworthy frankness, for all his valuable suggestions, for having often helped me to overcome my fears, simply believing in myself. Thanks to Prof. Claudia D'Ambrosio for having shared with me her great scientific experience, for her patience and support. If I think of the woman I would like to become, I can only imagine myself similar to her. She guided me into the world of research, as well as that of the city of Paris. The award for recommending me the best restaurants and the most characteristic places to visit goes to her.

I had the honor to spend some months working with Prof. Martin Schmidt at the Department of Mathematics of the University of Trier. The period spent there has been essential to open up new scenarios of study and to understand more in depth several important points of my thesis. Prof. Martin Schmidt, according to the MINOA project, is my "mentor" and working with him I could really feel him as such. Collaborating with him has been a privilege not only for his great experience in the field of optimization, but also for the beautiful person he is. In Trier, I also enjoyed the nice atmosphere of his *Nonlinear Optimization* group, who made me feel welcomed and part of the team immediately. I sincerely thank them for the nice moments spent together.

Thanks to the MINOA project I could experience working in a great company like Siemens AG, in Munich. There, with Dr. Kurt Majewski's guidance, I tiptoed into a project that was already underway, managing to enter dynamics other than purely academic ones and learning many challenging things.

I want to express my gratitude to all the colleagues of the OptimiX team that I have had the chance to meet over the years. During the confinement we experienced, I really realized how precious it was to be able to share my working days with them, complain (me more than them) about the food at the *magnan*, write our ideas on the blackboards of the LIX open spaces and daydream about our future. It has been a pleasure working directly with Antoine, Mercedes, Mauro, Diego, and our complementing collaborations have yielded some great papers. I also want to thank all LIX secretaries, who were always there to help me and who thought about things when I would have forgotten them. From conferences and doctoral courses, I also met a lot of good friends. I am indebted to them for making these three years the best ever. Special thanks to Matteo, Felipe, Filippo, and the rest of our crew for always knowing how to cheer me up, for discussing the problems of being a PhD student, for sharing the (infinitely many) bureaucratic issues, and for our adventures in Paris (if Violin Dingue walls could talk..), but also in the unforgettable Angers, Provence, Como. I am looking forward to welcoming them to Naples. Thanks to Alice, for her infinite patience, for our complicity, simply for having been there from beginning to end. Fate has brought us together in that room of Fontainebleau, and I'll make sure we never get lost.

This dissertation could not have happened without the continuous support of my family and my friends. Their tremendous understanding and encouragement in the past few years have been the fuel that helped me to complete this journey. I am immensely grateful to my mother for her support and tolerance in the most stressful periods, for her smiles, for being able to understand me even from a video call. My father was the first who believed in me, who introduced me to the world of research. I thank him for letting me free to make my choices without any conditioning, while always making me feel with my back covered. And finally, I am grateful to Fabiana because if it is easy to find people who console you in the dark times, it is really difficult to find someone like her who celebrates your goals as if they were hers. Last, but definitely not least, thanks to my boyfriend, Antonio. I thank him for being my greatest source of inspiration, for giving me the serenity that I was missing, for leaving me free to follow my dreams, with the awareness that my heart will always and only be his.

Abstract

A bilevel problem is an optimization problem where a subset of variables is constrained to be optimal for another given optimization problem parameterized by the remaining variables. The outer problem is commonly referred to as the upper-level problem, the inner one as the lower-level problem.

The first part of this dissertation concerns the key definitions, the solution approaches and the complexity of bilevel problems, as well as the study of a particular class of bilevel programs. Such bilevel problems have a lower level with a quadratic objective function, the value of which is contained into an upper-level inequality constraint. They can be obtained by reformulating semi-infinite programming problems with an infinite number of quadratically parametrized constraints. We propose an approach to solve this class of bilevel programs, based on the dualization of the lower level. This approach is compared with a new cutting plane algorithm, which we prove to be convergent. The rate of convergence of this algorithm is derived under stricter assumptions and is directly related to the iteration index, which is something new w.r.t. what is usually proved in semi-infinite programming literature. We successfully test the proposed methods on two applications: the constrained quadratic regression and a zero-sum game with cubic payoff.

The second part of the thesis is devoted to practical applications. A chapter is dedicated to the aircraft conflict resolution problem. This problem essentially consists in enforcing a minimum distance between flying aircraft to avoid conflicts, using different strategies. We focus on two of them: speed regulations and heading angles changes. We present a natural semi-infinite formulation of the problem via speed regulation strategy in k dimensions. To deal with the issue of uncountably many constraints of this formulation, we reformulate it, firstly, using polynomial programming, and, secondly, using bilevel programming. Then, we present a bilevel formulation of conflict resolution problem via heading angle changes in two dimensions (i.e., aircraft flying at the same altitude). In both bilevel formulations, the convexity of the lower level allows us to derive three different single-level reformulations, using KKT conditions, Dorn's duality, and Wolfe's duality, respectively. The resulting single-level reformulations of both problems are solved by using state-of-the-art solvers. Alternatively, we propose a cut generation algorithm to solve the bilevel problems, which fits in the general framework of the cutting plane algorithm presented in the first part. This algorithm obtains the best results in terms of computational time for most of the tested instances. Another application studied in this dissertation involves the Alternating Current (AC) Optimal Power Flow (ACOPF) problem at the lower level. The idea comes from the possibility for power generation in private households. In this scenario, we derive a bilevel problem to model the interaction between a retailer and several prosumers (consumers who can also produce, store, and sell power), who interact with each other through an AC network. When, together with the ACOPF, one wants to optimally design a power transportation network with respect to line activity, an ACOPF with on/off variables on lines can be used, which yields a nonconvex mixed-integer nonlinear problem in complex numbers. We, then, propose two convex relaxations, compared with the famous Jabr's second-order cone relaxation.

Résumé

Un problème de programmation à deux niveaux est un problème d'optimisation où un sous-ensemble de variables est contraint de prendre la valeur d'une solution optimale d'un autre problème d'optimisation, paramétré par les variables restantes. En termes mathématiques, un problème à deux niveaux peut être écrit comme suit :

$$\left.\begin{array}{ll} \min\limits_{x} & F(x,y) \\ s.t. & G(x,y) \leqslant 0 \\ & y \in \arg\min\limits_{y' \in Y}\{f(x,y') \mid g(x,y') \leqslant 0\} \\ & x \in X. \end{array}\right\} \tag{PB}$$

Le problème d'optimisation externe est communément appelé problème de niveau supérieur dans les variables x, et le problème interne problème de niveau inférieur dans les variables y'. Historiquement, les premières études liées à l'optimisation à deux niveaux se trouvent dans les travaux de l'économiste allemand von Stackelberg [196] (1934) sur la théorie des jeux, dans lesquels deux joueurs interagissent successivement. En fait, le problème (PB) peut être interprété comme un jeu hiérarchique, impliquant un *leader*, qui décide du problème de niveau supérieur, et un *follower*, qui décide du problème de niveau inférieur. Le leader a une connaissance complète du problème de niveau inférieur, tandis que le follower ne fait qu'observer les décisions du leader et optimiser par conséquent ses propres stratégies. Si le follower a plus d'une réponse optimale à une certaine sélection du leader, la meilleure ou la pire solution du follower par rapport à la fonction objective du leader est supposée. Le problème à deux niveaux qui en résulte est appelé problème de programmation à deux niveaux *optimiste* ou *pessimiste*, respectivement. Dans cette thèse, lorsque la solution de niveau inférieur n'est pas unique, nous ne considérons que l'approche optimiste.

Selon l'état actuel de l'art sur les problèmes de programmation biniveau, leurs formulations de programmation mathématique peuvent être très utiles pour aborder des situations complexes impliquant plusieurs décideurs avec des objectifs différents. En effet, la programmation à deux niveaux a été utilisée pour formuler de nombreux problèmes hiérarchiques du monde réel dans le domaine de la planification de la production et de la capacité [101, 142], du trafic et du transport [144, 35, 147, 2], chimie [63, 62, 72], sciences de la gestion [31, 169, 69, 68], défense des infrastructures critiques [17, 91, 80], ainsi que réseaux et marché de l'énergie [99, 205]. En raison de leur capacité à représenter l'interaction entre deux acteurs autonomes, les problèmes de programmation à deux niveaux sont intrinsèquement difficiles à résoudre. Déjà leur version la plus simple avec seulement des fonctions linéaires et des variables continues est fortement NP-hard [109].

La contribution de cette thèse est double. Sa première partie est composée de trois

chapitres. Les deux premiers ont pour but d'introduire les lecteurs dans le contexte de la programmation biniveau, tandis que le troisième présente deux approches théoriques pour traiter une classe particulière de programmes à deux niveaux .

Le chapitre 1 est consacré aux notions fondamentales associées aux problèmes de programmation biniveau. Premièrement, pour formaliser les concepts de base de la programmation à deux niveaux qui sont utilisés tout au long de la thèse, nous fournissons quelques définitions clés et examinons les différences entre les formulations optimistes et pessimistes des problèmes à deux niveaux . Ensuite, nous dérivons des reformulations classiques à un seul niveau des problèmes biniveau: l'une basée sur la soi-disant *fonction de valeur*, les autres sur les conditions de Karush-Kuhn-Tucker (KKT) ou sur la dualité forte du niveau inférieur, supposé être convexe. Enfin, nous offrons un aperçu des algorithmes qui ont été proposés pour l'optimisation biniveau: méthodes des points extrêmes, techniques de *branch-and-bound* et de *branch-and-cut*, méthodes de descente, algorithmes de fonction de pénalité, méthodes de région de confiance, entre autres.

Le chapitre 2 est complémentaire du chapitre 1. En effet, étant génériquement non-convexes et non-différentiables, les problèmes à deux niveaux sont intrinsèquement difficiles. Dans le chapitre 2, nous discutons de la complexité computationnelle des problèmes à deux niveaux. Après une introduction sur la hiérarchie polynomiale, nous utilisons le concept des formulations à double quantificateur pour dériver des considérations sur les problèmes de programmation à deux niveaux. Nous présentons ensuite des résultats pour les problèmes à deux niveaux linéaires et linéaires en nombres entiers, ainsi que des cas particuliers dans lesquels les problèmes à deux niveaux peuvent être résolus en temps polynomial. Enfin, la complexité de ce que l'on appelle les *problems à deux niveaux indépendants* est abordée, puisque les formulations de ce type sont considérées dans le chapitre suivant.

Le chapitre 3 conclut la première partie de la thèse. Il concerne l'étude d'une classe particulière de problèmes à deux niveaux, où le vecteur de décision du niveau inférieur y n'apparaît ni dans les contraintes de couplage du niveau supérieur ni dans la fonction objectif du niveau supérieur. Le niveau inférieur est modélisé comme un problème de programmation quadratique, avec un ensemble admissible $\mathcal{F}$ qui est un polyèdre ne pas dépendent des variables du niveau supérieur x. La valeur de ce niveau inférieur se traduit par une contrainte de niveau supérieur de la forme $h(x) \leqslant \min_{y \in \mathcal{F}} f(x, y)$. Comme les valeurs des variables de niveau inférieur ne sont pas utilisées dans le niveau supérieur, l'opérateur *argmin* présent dans (PB) est redondant. Ainsi, le problème des solutions optimales équivalentes du problème de niveau inférieur n'existe pas. De telles formulations à deux niveaux peuvent être considérées comme des reformulations de problèmes de Programmation Semi-Infinie (PSI), c'est-à-dire des problèmes d'optimisation ayant un nombre infini de contraintes paramétrées (quadratiquement) par y du type $\forall y \in Y, \ ;0 \leqslant f(x, y)$. En fait, dans ce case, il suffit d'imposer que cette inégalité est vraie pour le minimum sur tous les $y \in Y$ de $f(x, y)$ de la manière suivante $0 \leqslant \min_{y \in Y} f(x, y)$, reformulant ainsi le programme PSI en un problème à deux niveaux. Pour résoudre les problèmes PSI, des méthodes de discrétisation, des méthodes de plans coupants et d'autres méthodes hybrides sont utilisées dans la littérature. Nous explorons une approche d'un autre type, qui procède par la résolution d'une formulation à un seul niveau avec un nombre fini de contraintes, obtenue par la dualisation du problème de niveau inférieur $\min_{y \in \mathcal{F}} f(x,y)$, en utilisant la programmation semi-définie. Si la dépendance quadratique de $f(x,y)$ à le vecteur paramètre y est convexe, cette formulation à un seul niveau est une reformulation équivalente du programme original à deux niveaux. Dans le formalisme de l'optimisation robuste, il s'agit de traiter des contraintes quadratiquement perturbées sous un ensemble d'incertitudes polytopique. Les principes de base pour traiter les perturbations non linéaires sont brièvement exposés dans [36] par Ben-Tal et al. mais le cas des perturbations quadratiques avec un ensemble d'incertitudes polytopique n'a pas encore été abordé, à notre connaissance. Afin de comparer cette approche avec une approche plus traditionnelle, nous introduisons également un algorithme de plans coupants adapté, dont nous avons prouvé la convergence. Un nouveau taux de convergence est donné lorsque la fonction objectif de niveau supérieur est fortement convexe, et sous une hypothèse stricte de faisabilité. Ce taux de convergence est directement lié à l'indice d'itération, ce qui est nouveau par rapport à ce qui est généralement prouvé dans la littérature PSI. La validité de ces approches est démontrée par des résultats d'implémentation et de calcul sur deux applications différentes : un jeu à somme nulle avec un gain cubique, et une régression quadratique contrainte.

La deuxième partie de la thèse est consacrée aux applications. En particulier, un chapitre exploite le contenu du chapitre 3, et est dédié au problème de résolution de conflits d'avions (PRC) des avions. Un autre chapitre traite du problème de flux de puissance optimal en courant alternatif (ACOPF), à la fois dans un cadre à deux niveaux et à un seul niveau.

Le chapitre 4 aborde le PRC via différentes approches. Dans le domaine de la gestion du trafic aérien, le terme *résolution des conflits entre aéronefs* — également appelé *déconfliction* — désigne l'ensemble des stratégies utilisées pour détecter et résoudre les conflits potentiels entre aéronefs partageant la même portion d'espace aérien, deux aéronefs étant dits potentiellement en conflit si leur distance relative est inférieure à un seuil de sécurité donné [8]. La résolution des conflits consiste alors à fournir des configurations d'aéronefs sans conflit, en modifiant les trajectoires des aéronefs. La résolution centralisée des conflits suppose qu'une autorité de contrôle du trafic aérien est chargée de surveiller les trajectoires des aéronefs pour résoudre les conflits potentiels entre eux. Dans ce contexte, la déconfliction des aéronefs peut être modélisée comme un problème d'optimisation dans lequel les trajectoires des aéronefs sont modifiées pour assurer la distance de sécurité entre les aéronefs, en minimisant l'impact de ces ajustements de trajectoire. Parmi les différentes manœuvres qui peuvent être utilisées pour prévenir les conflits, nous nous concentrons sur les régulations de vitesse, et les changements d'angle de cap. Alors que la PRC via la régulation de vitesse est modélisée en k dimensions, la PRC via les modifications d'angles est formulée en deux dimensions (c'est-à-dire lorsque les avions volent à la même altitude). La PRC consiste essentiellement à imposer une distance minimale entre les avions en vol sur un horizon temporel donné, ce qui conduit naturellement à une formulation PSI. Nous utilisons d'abord la programmation polynomiale pour reformuler la formulation PSI du PRC via des régulations de vitesse, en partant d'un résultat présenté dans [208]. Ensuite, nous reformulons à la fois la formulation PSI du PRC via la vitesse et via les changements d'angle de cap en formulations à deux niveaux avec un problème de niveau inférieur pour chaque paire d'avions. Dans les deux cas, la convexité des problèmes de niveau inférieur nous permet de dériver trois problèmes différents de niveau unique, en utilisant les conditions KKT, la dualité de Dorn et la dualité de Wolfe. Les reformulations à un seul niveau des deux problèmes sont résolues à l'aide de solveurs de pointe, qui fournissent de bonnes solutions en un temps de calcul raisonnable. Alternativement, nous proposons un algorithme de génération de coupes pour résoudre les problèmes à deux niveaux dans la même veine que la méthode de plans coupants présentée dans le chapitre 3. Cet algorithme, comparé aux solveurs de l'état de l'art, obtient les meilleurs résultats en termes de temps de calcul pour la plupart des instances testées. Les résultats numériques, comparés à d'autres dans la littérature, sont encourageants et soulignent le potentiel des approches proposées.

Dans le chapitre 5, nous étudions le problème ACOPF. Tout d'abord, nous utilisons un problème à deux niveaux avec ACOPF au niveau inférieur pour modéliser l'interaction entre un détaillant et plusieurs *prosommateurs*, c'est-à-dire des consommateurs qui peuvent également produire, stocker et vendre de l'énergie. Ces prosommateurs interagissent les uns avec les autres par le biais d'un réseau électrique alternatif (chaque prosommateur est un bus) et visent à maximiser leurs revenus totaux provenant de la vente d'électricité/minimiser le coût payé au détaillant lorsque leur production ne satisfait pas leur besoin en électricité. Au niveau supérieur, le détaillant, qui fixe le prix de l'électricité pour l'ensemble des prosommateurs, vise à maximiser son propre profit. Après la formulation du problème via la programmation à deux niveaux, le niveau inférieur est convexé grâce à la programmation conique du second ordre. Lorsqu'on considère, avec l'ACOPF, le problème de la conception optimale d'un réseau de transport d'énergie en fonction de l'activité des lignes, on peut utiliser un ACOPF avec des variables on/off sur les lignes. Cela donne un problème non-convexe non linéaire mixte en nombres complexes. Dans ce scénario, nous proposons deux relaxations convexes, comparées à la relaxation conique de second ordre bien connue formulée par Jabr dans [115].

Contents

List of acronyms

The following table describes the meaning of all the abbreviations and acronyms used throughout the dissertation.

Acronym	Definition
AC	Alternating Current
ACOPF	Alternating Current Optimal Power Flow
BLP	Bilevel Linear Problem
BQP	Bilevel Quadratic Problem
CRP	Conflict Resolution Problem
CP	Cutting Plane
DC	Direct Current
DD	Diagonally Dominant
DDP	Diagonally Dominant Programming
DCOPF	Direct Current Optimal Power Flow
FCFW	Fully Corrective Frank-Wolfe
GLCP	Generalized Linear Complementary Problem
HAC	Heading Angle Changes
HPR	High Point Relaxation
IR	Inducible (or Induced) Region
KKT	Karush-Kuhn-Tucker
LMP	Locational Marginal Prices
LP	Linear Programming
MIBLP	Bilevel Mixed-Integer Linear Problem
MILP	Mixed-Integer Linear Programming
MINLP	Mixed-Integer Nonlinear Programming
MPCC	Mathematical Problem with Complementarity Constraints
NLP	NonLinear Programming
OPF	Optimal Power Flow
PP	Polynomial Programming
PSD	Positive Semidefinite
SDP	Semidefinite Programming
SDr	Semidefinite Representable
SIP	Semi-Infinite Programming
SIP	Semi-Infinite Programming
SOC	Second-Order Cone
SOCP	Second-Order Cone Programming
SR	Speed Regulation

Table 1: List of acronyms

Introduction

A bilevel programming problem is an optimization problem where a subset of the variables is constrained to take the value of an optimal solution of another given optimization problem parameterized by the remaining variables. In mathematical terms, a bilevel problem can be written as follows:

$$\left.\begin{array}{ll} \min\limits_{x} & F(x,y) \\ s.t. & G(x,y) \leqslant 0 \\ & y \in \arg\min\limits_{y' \in Y}\{f(x,y') \mid g(x,y') \leqslant 0\} \\ & x \in X. \end{array}\right\} \qquad (P)$$

The outer optimization problem is commonly referred to as the upper-level problem in the variables x, and the inner one as the lower-level problem in the variables y'. From now on, with a slight abuse of notation which we hope will simplify readability, we dispense with the distinction between y and y', and adopt the widespread usage of using the y both outside and inside the lower level, i.e., the bilevel constraint will read $y \in \arg\min\limits_{y \in Y}\{f(x,y) \mid g(x,y) \leqslant 0\}$.

Historically, the first studies related to bilevel optimization can be found in the works of the German economist von Stackelberg [196] (1934) in the field of game theory, in which two players interact successively. Indeed, problem (P) can be interpreted as a hierarchical game, involving a leader, who decides on the upper-level problem, and a follower, deciding on the lower-level problem. The leader has complete knowledge of the lower level-problem, while the follower only observes the decisions of the leader and consequently optimizes his/her own strategies. If the follower has more than one optimal response to a certain selection of the leader, either the best or the worst follower's solution with respect to the leader's objective function is assumed. The resulting bilevel problem is called *optimistic* or *pessimistic* bilevel programming problem, respectively. In this dissertation, whenever the lower-level solution is not unique, we only consider the optimistic approach.

According to the current state of the art on bilevel programming problems, their mathematical programming formulations can be quite useful in tackling complex situations involving multiple decision makers with different goals. Indeed, bilevel programming has been used to formulate many real-world hierarchical problems in the field of production and capacity planning [101, 142], traffic and transportation [144, 35, 147, 2], chemistry [63, 62, 72], management science [31, 169, 69, 68], defense of critical infrastructure [17, 91, 80], as well as energy networks and market [99, 205]. Because of the ability to represent the interaction among two autonomous players, bilevel programming problems are intrinsically difficult to solve. Already their simplest version with only linear functions and continuous variables is strongly NP-hard [109].

The contribution of this dissertation is twofold. Its first part is made of three chapters. The first two of them aim at introducing the readers to the context of bilevel programming, while the third one presents two theoretical approaches to deal with a particular class of bilevel programs.

Chapter 1 is devoted to the fundamental notions associated with bilevel programming problems. Firstly, to formalize the basic concepts of bilevel programming that are used throughout the dissertation, we provide some key definitions and examine the differences between optimistic and pessimistic formulations of bilevel problems. Secondly, we derive classical single-level reformulations of the bilevel problems: one based on the so-called *value function*, the others on the Karush-Kuhn-Tucker (KKT) conditions or on strong duality of the lower level, assumed to be convex. Finally, we offer an overview on the algorithms that have been proposed for bilevel optimization: extreme point methods, branch-and-bound and branch-and-cut techniques, descent methods, penalty function algorithms, trust region methods, among others.

Chapter 2 is complementary to Chapter 1. Indeed, being generically nonconvex and non-differentiable, bilevel problems are intrinsically difficult. In Chapter 2 we discuss the computational complexity of bilevel problems. After an introduction about the polynomial hierarchy, we use the concept of double quantifier formulations to derive considerations about bilevel programming problems. We, then, present results for linear and mixed-integer linear bilevel problems, as well as special cases in which bilevel problems can be solved in polynomial time. Finally, the complexity of the so-called *independent bilevel problems* is tackled, since formulations of this sort are considered in the subsequent chapter.

Chapter 3 concludes the first part of the dissertation. It concerns the study of a particular class of bilevel problems, where the lower-level decision vector y appears neither in the upper-level coupling constraints nor in the upper-level objective function. The lower level is modeled as a quadratic programming problem, with a feasible set $\mathcal{F}$ which is a polyhedron assumed not to depend on the upper-level variables x. The value of such lower level occurs into an upper-level constraint of the form $h(x) \leqslant \min_{y \in \mathcal{F}} f(x, y)$. Since the values of the lower-level variables are not used in the upper level, the *argmin* operator present in (P) is redundant.

Thus, the problem of equivalent optimal solutions of the lower-level problem does not exist. Such bilevel formulations can be seen as reformulations of Semi-Infinite Programming (SIP) problems, i.e., optimization problems having an infinite number of (quadratically) parametrized constraints of the type $\forall y \in Y,\ 0 \leqslant f(x, y)$, where y is the parameter. In fact, if this is the case, it is sufficient to impose that this inequality holds for the minimum over all $y \in Y$ of $f(x, y)$ in the following way $0 \leqslant \min_{y \in Y} f(x, y)$, thus reformulating the SIP program into a bilevel problem. To solve SIP problems, discretization methods, cutting plane methods, and other hybrid methods are used in the literature. We explore an approach of a different kind, that proceeds by solving a single-level formulation with a finite number of constraints, obtained by dualizing the lower-level problem $\min_{y \in \mathcal{F}} f(x, y)$, using semidefinite programming. If the quadratic dependence of $f(x, y)$ on the parameter vector y is convex, this single-level formulation is an equivalent reformulation of the original bilevel program. In the formalism of Robust Optimization, it is about dealing with quadratically perturbed constraints under a polytopic uncertainty set. The main principles to tackle nonlinear perturbation are briefly outlined in [36] by Ben-Tal et al., but the case of quadratic perturbations with a polytopic uncertainty set have not been addressed yet to the best of our knowledge. In order to compare this approach with a more traditional one, we also introduce a tailored cutting plane algorithm, which we proved to be convergent. A new rate of convergence is given when the upper-level objective function is strongly convex, and under a strict feasibility assumption. Such convergence rate is directly related to the iteration index, which is something new with respect to what is usually proved in SIP literature. The validity of these approaches is shown through implementation and computational results on two different applications: a zero-sum game with a cubic payoff, and a constrained quadratic regression.

The second part of the thesis is devoted to applications. In particular, a chapter leverages the contents of Chapter 3, and is dedicated to the aircraft Conflict Resolution Problem (CRP). Another chapter addresses the Alternating Current (AC) Optimal Power Flow Problem (ACOPF), both in a bilevel and in a single-level framework.

Chapter 4 addresses the CRP via different approaches. In air traffic management, the term *aircraft conflict resolution* — also known as aircraft *deconfliction* — designates the set of strategies used to detect and solve potential conflicts among aircraft sharing the same portion of airspace, where two aircraft are said to be potentially in conflict if their relative distance is less than a given safety threshold [8]. Conflict resolution consists then in providing conflict-free aircraft configurations, by modifying aircraft trajectories. Centralized conflict resolution assumes that an Air Traffic Control authority is in charge of monitoring aircraft trajectories to resolve potential conflicts among aircraft. In this context, aircraft deconfliction can be modeled as an optimization problem in which aircraft trajectories are modified to ensure the safety distance among aircraft, while minimizing the impact of these trajectory adjustments. Among the different maneuvers which can be used to prevent conflicts, we focus on speed regulations, and heading angle changes. While CRP via speed regulation is modeled in k dimensions, the CRP via angles modifications is formulated in two dimensions (i.e., when aircraft fly at the same altitude). The CRP essentially consists in enforcing a minimum distance between flying aircraft over a given time horizon, which naturally results in a SIP formulation. We first use polynomial programming to reformulate the SIP formulation of CRP via speed regulations, starting from a result presented in [208]. Then, we reformulate both the SIP formulation of the CRP via speed and via heading angle changes into bilevel formulations with a lower-level problem for each pair of aircraft. In both cases, the convexity of the lower-level problems allows us to derive three different single-level problems, using KKT conditions, Dorn's duality, and Wolfe's duality. The single level reformulations of both problems are solved by using state-of-the-art solvers, which provide good solutions in reasonable computing time. Alternatively, we propose a cut generation algorithm in the same vein of the cutting plane method presented in Chapter 3 to solve the bilevel problems. This algorithm, compared with state-of-the-art solvers, obtains the best results in terms of computational time for most of the tested instances. Numerical results, when compared with others in the literature, are encouraging and stress the potential of the proposed approaches.

In Chapter 5, we study the ACOPF problem. Firstly, we use a bilevel problem with ACOPF at the lower level to model the interplay among a retailer and several *prosumers*, i.e., consumers who can also produce, store, and sell power. Such prosumers interact with each other through an AC power network (each prosumer is a bus) and aim at maximizing their total revenues from selling power/minimizing the cost paid to the retailer when their production does not satisfy their power need. At the upper level, the retailer, who sets the price of power for the set of prosumers, aims at maximizing his/her own profit. After the formulation of the problem via bilevel programming, the lower level is convexified thanks to second-order cone programming. When, together with the ACOPF, the problem of optimally designing a power transportation network with respect to line activity is considered, an ACOPF with on/off variables on lines can be used. It yields a nonconvex mixed-integer nonlinear problem in complex numbers. In this scenario, we propose two convex relaxations, compared with the well-known second-order cone relaxation formulated by Jabr in [115].

Publications and contributions

The present thesis is the result of different peer-reviewed journal articles, submitted preprints, as well as ongoing works, which are the outcome of past and current scientific collaborations with different co-authors. The context in which these works have been produced and my contributions to each of them are clearly stated in the following.

1. During the first part of my PhD, I studied the state of the art of bilevel programming. My research focused on the aircraft deconfliction problem, to which Chapter 4 is dedicated. Three works emerged from this first phase. A conference paper "Flying Safely by Bilevel Programming" was published in the AIRO Springer Series *Advances in Optimization and Decision Science for Society, Services and Enterprises* book [1]. Two papers were published on international journals: "Detecting and solving aircraft conflicts using bilevel programming" appeared in *Journal Of Global Optimization* [2], and "Polynomial programming prevents aircraft (and other) conflicts" in *Operation Research Letters* [3]. Paper [1] is a joint work with my two thesis advisors Profs. Leo Liberti and Claudia D'Ambrosio. Paper [2] is the result of a collaboration with my advisors and Mercedes Pelegrín, a postdoctoral researcher in the OptimiX team at LIX (CNRS - École Polytechnique). Finally, paper [3] is a collaboration between prof. Leo Liberti and myself. In order to test our approaches, I proposed some new 3D instances for the conflict resolution problem, publicy available at [4]. During the last months, Mercedes Pelegrín and I worked at a benchmark generator [5] that allows generating benchmarks of the problem of different complexity levels. In particular, I took care of the 3D (spheric, polyhedral, cubic and random) instances generator.

2. At the end of 2019, I visited Columbia University, where my thesis advisor Prof. Leo Liberti had already started a collaboration with Prof. Daniel Bienstock. I had the opportunity to work with them and Mauro Escobar, a postdoctoral researcher at LIX at that time, on a Power Network design problem. The result of this collaboration is the submitted paper "Power network design with line activity" [6], which is part of Chapter 5.

3. A few months after the beginning of my second year of PhD, in January 2020 I started my research stay at the Department of Mathematics of the University of Trier (Germany), which lasted almost three months. There, I collaborated with Prof. Martin Schmidt, who welcomed me within his *Nonlinear Optimization* team. Preliminary results of this ongoing collaboration are given in the first part of Chapter 5.

4. During the last phase of my doctoral studies I collaborated with another PhD student of OptimiX team, Antoine Oustry. Together with my thesis advisors, we worked on two solution approaches for a particular class of

bilevel programs, presented in detail in Chapter 3. As an outcome of this phase, the manuscript "Solving a class of bilevel programs with quadratic lower level" was submitted to the *SIAM Journal on Optimization* [7].

5. In March 2021, some members of my team at LIX (OptimiX team) were invited to contribute to the 3rd edition of the *Springer Encyclopedia of Optimization*, edited by Panos M. Pardalos and Oleg A. Prokopyev. In particular I contributed to two entries of the Encyclopedia, which have been submitted in the beginning of September: "Optimal Power Flow" (authors: Cerulli M., Delle Donne D., Escobar M., Liberti L., Oustry A.) [9], and "Aircraft Conflict Resolution" (authors: Cerulli M., Pelegrín M., Cafieri S., D'Ambrosio C., Rey D.) [8].

Part I

Bilevel optimization

Chapter 1

Bilevel programming

Bilevel programming is a field of mathematical programming in which some variables are constrained to be the optimal solution of another optimization problem. Therefore, bilevel optimization can be used to model the real-world hierarchical relationship between two autonomous, and possibly conflictual, decision makers. In this chapter, we discuss the key concepts of bilevel optimization. In Section 1.1, we introduce the basic concepts of bilevel optimization. We present some of the well-known single-level reformulation approaches in Section 1.2. In Section 1.3, some solution methods for the bilevel problems are described.

In the following, given a formulation (P) of an optimization problem, we use the term *reformulation* to describe a formulation having the same set of optima of (P), i.e., what is defined as *exact reformulation* in [131, Definition 10]. With the term *relaxation*, we refer to a formulation having a feasible set which contains the feasible set of (P) [131, Definition 13]. Finally, we use the term *restriction* when referring to a formulation having a feasible set which is included in the feasible set of (P).

1.1 Introduction

From a historical point of view, bilevel optimization is closely related to the economic problem of Stackelberg (1934) in the field of game theory [196, 197], which was used to model the interaction among two firms competing sequentially on the quantity of output they produce of a homogeneous good. A formal definition of bilevel programming problems was first introduced by Bracken and McGill (1973) [46] as “mathematical programs with optimization problems in the constraints”, but the designation bilevel and multilevel were only later introduced by Candler and Norton (1977) [52]. For a general overview of bilevel programming, we refer to the thorough surveys [195, 65, 74, 73], and to the books by Bard [27] and Dempe [72, 77].

A bilevel programming problem can be seen as a hierarchical game, where two players (a leader and a follower) make their decisions following a hierarchical order. Firstly, the leader makes his/her choice and communicates this to the follower, who will select a response taking into account the selection of the leader, and give it back to the leader. Thus, the leader's task is to determine the best decision optimizing, together with the *reaction* of the follower, his/her own objective function.

The mathematical programming formulation of a classical (continuous) bilevel problem is the following:

$$\min_{x} F(x,y) \tag{1.1}$$

$$s.t.\ G(x,y) \leqslant 0 \tag{1.2}$$

$$y \in \arg\min_{y\in Y}\{f(x,y) \mid g(x,y) \leqslant 0\} \tag{1.3}$$

$$x \in X, \tag{1.4}$$

where $X \subseteq \mathbb{R}^n$, $Y \subseteq \mathbb{R}^m$, F and $f : X \times Y \to \mathbb{R}$, $G : X \times Y \to \mathbb{R}^p$, and $g : X \times Y \to \mathbb{R}^l$. We define $y \in \mathbb{R}^m$ the lower-level decision variable and refer to the embedded minimization problem as the *lower-level problem* (or follower's problem), respectively. For the abuse of notation regarding the use of y both inside and outside the lower level (1.3), see the note after the formulation (P) in the introduction. We call the outer problem in the variables x and y *upper-level problem* (or leader's problem), and $x \in \mathbb{R}^n$ is the upper-level decision variable. Upper-level constraints $G(x,y) \leqslant 0$ are defined as *coupling constraints* if they depend on y.

The feasible set of the so-called *High Point Relaxation* (HPR) is defined as the set of points (x,y) satisfying upper and lower-level constraints (but (1.3)), i.e., for formulation (1.1)–(1.4) is the set

$$\mathcal{H} = \{(x,y) \in X \times Y \mid G(x,y) \leqslant 0 \wedge g(x,y) \leqslant 0\}. \tag{1.5}$$

If the HPR is infeasible then the original bilevel problem is infeasible. However, if the HPR is unbounded, the bilevel problem can either be unbounded too, or still infeasible, or admit a finite-valued solution (see [207, Example 3]). The *inducible* (or *induced*) *region* (IR) is the "feasible set of the bilevel problem", i.e., for formulation (1.1)–(1.4) is the set:

$$\text{IR} = \{(x,y) \in \mathcal{H} : y \text{ satisfies } (1.3)\}.$$

It is usually nonconvex and, in presence of upper-level constraints, can be disconnected or even empty.

The definition of the bilevel programming problem (1.1)–(1.4) does not consider the ambiguity in the definition of the problem in case of multiple optimal solutions in the follower's problem for some x values. To overcome this ambiguity two approaches have been suggested [74]. What we consider in the rest of this dissertation is the so-called *optimistic approach*, which consists in assuming that, if the follower has more than one optimal solution, he/she selects the one which is the best for the leader (in terms of upper-level objective function). To reflect this, the leader minimizes the function $\min_y F(x,y)$ over x. Thus, the optimistic bilevel problem is formulated as:

$$\min_{x,y} F(x,y) \tag{1.6}$$
$$s.t.\ G(x,y) \leqslant 0 \tag{1.7}$$
$$y \in \arg\min_{y \in Y}\{f(x,y) \mid g(x,y) \leqslant 0\} \tag{1.8}$$
$$x \in X, \tag{1.9}$$

Most of the literature on bilevel programming consider the optimistic problem. One of the reasons can be that this problem has an optimal solution under quite reasonable assumptions (see [72]). When cooperation of the leader and the follower is not allowed, or if the leader wants to bound the "damage" resulting from an undesirable selection by the follower, the *pessimistic approach* [203, 135] must be used: the leader assumes that the follower selects the optimal solution corresponding to the worst upper-level objective function value. To model this, the leader minimizes the function $\max_y F(x,y)$ over x. Thus, the pessimistic bilevel problem is formulated as:

$$\min_x \max_y F(x,y) \tag{1.10}$$
$$s.t.\ G(x,y) \leqslant 0 \tag{1.11}$$
$$y \in \arg\min_{y \in Y}\{f(x,y) \mid g(x,y) \leqslant 0\} \tag{1.12}$$
$$x \in X, \tag{1.13}$$

Essentially, both optimistic and pessimistic bilevel problems possess a structure involving three interrelated optimization problems [130]. In the optimistic version, this structure gives rise to a two-level problem, since the successive minimization of F over x and y, i.e., $\min_x \min_y F(x,y)$, is combined to form a joint minimization of F over the points $(x,y) \in \mathrm{IR}$, i.e., $\min_{x,y} F(x,y)$. As already written, in this dissertation, we only study the optimistic approach.

Most decisions in application areas such as energy, security, production planning, or revenue management are of a bilevel nature, in the sense that they impact systems with some degree of autonomy and conflicting objectives. The capability to model hierarchical decision processes also makes bilevel optimization problems notoriously hard to solve. For instance, already the easiest case of linear upper and lower level is strongly NP-hard; see Chapter 2 for details.

1.2 Single-level reformulations

To investigate bilevel problems, their transformation into one-level optimization problems may be necessary, and different approaches can be used towards this aim [75]. First of all, instead of using the *point-to-set mapping*, linking the leader's selection to the set of optimal solutions of the follower's problem, one can also use the *value function*

$$\varphi(x) = \min_{y \in Y} \{f(x, y) \mid g(x, y) \leqslant 0\}, \tag{1.14}$$

and reformulate the optimistic bilevel problem (1.6)–(1.9) as

$$\min_{x,\, y} F(x, y) \tag{1.15}$$

$$s.t.\ G(x, y) \leqslant 0 \tag{1.16}$$

$$g(x, y) \leqslant 0 \tag{1.17}$$

$$f(x, y) \leqslant \varphi(x) \tag{1.18}$$

$$x \in X,\ \ y \in Y \tag{1.19}$$

which is the so-called *value function reformulation*. Problems (1.6)–(1.9) and (1.15)–(1.19) are equivalent (both w.r.t. local and global optima). The function $\varphi(x)$ is in general not differentiable, even if all the constraints and the objective function in the lower level are smooth [76]. Using nonsmooth analysis, optimality conditions for the optimal value transformation can be obtained (see, e.g., [76]).

When the lower-level problem is convex for each feasible upper-level variable, and satisfies some regularity conditions (e.g., Slater's constraint qualification), it is possible to reformulate the bilevel problem into a single-level problem using either KKT conditions of the lower level or strong duality applied to the lower-level problem. Indeed, in this case, the lower-level problem can be replaced by its necessary and sufficient KKT conditions, obtaining the following single-level reformulation of problem (1.6)–(1.9):

$$\min_{x,y,\lambda} F(x, y) \tag{1.20}$$

$$s.t.\ G(x, y) \leqslant 0 \tag{1.21}$$

$$0 \in \partial_y f(x, y) + \lambda^\top \partial_y g(x, y) = 0 \tag{1.22}$$

$$g(x, y) \leqslant 0, \quad \lambda \geqslant 0 \tag{1.23}$$

$$\lambda^\top g(x, y) = 0, \tag{1.24}$$

$$x \in X, y \in Y, \lambda \in \mathbb{R}^l \tag{1.25}$$

where $\lambda \in \mathbb{R}^l$ is the Lagrangian multiplier associated to the lower-level convex inequality constraint $g(x, y) \leqslant 0$. Eq. (1.22) (setting the gradient of the lower-level Lagrangian function equal to zero) corresponds to lower-level stationarity conditions, Eq. (1.23) to lower-level primal and dual feasibility, and Eq. (1.24) to complementarity slackness. If the lower level is nonconvex, the formulation (1.20)–(1.25) is a relaxation of the original bilevel problem, since the set of bilevel feasible solutions is enlarged by adding local optima as well as stationary solutions of the lower-level problem to it.

We can claim that the above presented reformulation is the most often used approach to deal with bilevel problems, despite the nonconvexities that occur in the complementarity constraint (1.24). Indeed, problem (1.20)–(1.25) is a Mathematical Problem with Complementarity Constraints (MPCC) [139, 157]. In [178], it is shown that the Mangasarian-Fromovitz constraint qualification is violated at every feasible point of the problem (1.20)–(1.25). Thus, the solution of the problem and also the definition of (necessary and sufficient) optimality conditions is difficult.

For bilevel problems with convex lower level, another approach consists in exploiting strong duality of such lower level. Given the lower level

$$\min_y \{ f(x, y) \mid y \in \mathcal{F}(x, y) \}$$

with $\mathcal{F}(x, y) = \{ y \in Y \mid g(x, y) \leqslant 0 \}$ the lower-level feasible set, and the dual variable $z \in \mathbb{R}^m$, we can write the lower-level dual problem as

$$\max_z \{ f_d(x, z) \mid z \in \mathcal{F}_d(x, z) \},$$

where f_d is the dual objective function and $\mathcal{F}_d(x, z)$ the dual feasible set. By weak duality, for all $(x, y, z) \in X \times \mathcal{F}(x, y) \times \mathcal{F}_d(x, y)$ we have:

$$f_d(x, z) \leqslant f(x, y).$$

Lower-level optimality of y can, thus, be ensured by imposing that a lower-level dual feasible z exists such that:

$$f_d(x, z) \geqslant f(x, y),$$

i.e., strong duality holds for the pair (y, z).

We can, thus, obtain the so-called *strong duality reformulation* of problem (1.6)–(1.9) (assuming the convexity of the lower level) by

$$\min_{x,y,z} F(x, y) \tag{1.26}$$

$$s.t.\ G(x, y) \leqslant 0 \tag{1.27}$$

$$f_d(x, z) \geqslant f(x, y) \tag{1.28}$$

$$x \in X,\ y \in \mathcal{F}(x, y),\ z \in \mathcal{F}_d(x, z). \tag{1.29}$$

Also with this approach, constraint (1.28) may introduce some nonconvexities even when suitable convexity assumptions are made on all the objectives and constraints of the bilevel formulation, but, still, the overall bilevel optimization problem is reduced to a single-level (constrained) optimization problem.

1.3 Solution approaches

In this section we introduce the main methods from the literature to tackle bilevel problems, following the structure of surveys [65, 180, 123, 194]. Being bilevel programming problems inherently complex (see Chapter 2), it is not surprising that most algorithmic literature has focused on the simplest case of the continuous Bilevel Linear Problems (BLPs) having only continuous variables and linear functions in both the levels [33]. They can be formulated as:

$$\begin{aligned} \min_{x,y}\ & c^\top x + d^\top y \\ \text{s.t. } & Ax + By \geqslant a \\ & y \in \arg\min_{y}\ e^\top x + f^\top y \\ & \quad \text{s.t. } \quad Cx + Dy \geqslant b, \end{aligned} \tag{1.30}$$

where $c, e \in \mathbb{R}^n$, $d, f \in \mathbb{R}^m$, $A \in \mathbb{R}^{p\times n}$, $B \in \mathbb{R}^{p\times m}$, $C \in \mathbb{R}^{l\times n}$, $D \in \mathbb{R}^{l\times m}$, $a \in \mathbb{R}^p$, and $b \in \mathbb{R}^l$. An important feature of BLPs is that their solution set, whenever it is nonempty, contains at least one vertex of the constraint region (1.5). Therefore, many methods for solving BLPs are based on vertex enumeration. The first approach of this kind is presented in [53] for solving BLPs with no upper-level constraints. Afterwards in [41], under the assumption that $\mathcal{H}$ is bounded, a so-called "K^{th} method" is proposed. While the former algorithm enumerates vertices of the lower-level problem, the latter, enumerates vertices of the HPR. Other vertex enumeration approaches for BLPs have been proposed by [59, 70, 159, 190].

When the bilevel problem is a BLP, it is often transformed into a MPCC through the KKT reformulation (see Section 1.2). The most common approach in the literature is to use Mixed-Integer Linear Programming (MILP) techniques to linearize the KKT complementarity constraints $\lambda^\top(Cx + Dy - b) = 0$ by replacing them with the constraints

$$\lambda \leqslant M_D z \quad Cx + Dy \leqslant b + M_P(1 - z),$$

where $z \in \{0,1\}^m$ is a new binary variable, and M_D and M_P are sufficiently large big-M constants. On the one hand, existing powerful solvers can be used to solve the obtained MILP problem. On the other hand, the choice of the big-M constants can significantly influence the performance and correctness of the reformulation.

There may be context-specific ways to choose a safe value for big-Ms, but this is not always the case. In [124], it is proved that finding a correct big-M cannot be done efficiently, i.e., in polynomial-time, unless $\mathsf{P} = \mathsf{NP}$. If the selected big-Ms is too small, the solution of the MILP problem could be not bilevel optimal. Otherwise, the choice of too large big-Ms can lead to weak relaxations, which, in turn, may make the model hard for the solvers.

Several branch-and-bound approaches exploit the complementarity between the multipliers and the slack variables of the lower-level constraints in the condition (1.24). This is discussed in [28] and [94], where a general branch-and-bound method is applied to the KKT reformulation of a BLP. The approach was adapted to linear-quadratic in [29] and to quadratic-quadratic bilevel problems in [16, 26, 86], with some additional assumptions.

Novel branching rules, different from most-violated complementarity, to solve medium-size BLPs have been studied in [109].

What can be seen as a combination of the methods already presented (i.e., vertex enumeration and branch-and-bound) is the so-called *sequential linear complementarity problem* for solving linear and linear-quadratic bilevel programs, introduced in [118, 119]. This approach seems quite efficient for the solution of medium-scale problems.

Several descent methods have been proposed in literature for solving bilevel programming problems. A descent direction in bilevel optimization is a direction $d \in \mathbb{R}^n$ along which the upper-level function value decreases while keeping bilevel feasibility. Given that a point is considered bilevel feasible only if it is lower-level optimal, finding a descent direction can be challenging. Indeed, a major issue is the availability of the gradient (or a sub-gradient) of the upper-level objective function at a feasible point. To tackle this problem, researchers have investigated ways to approximate the gradient of the upper-level objective [128] as well as considered formulation of auxiliary programs, like in [177] where the computation of the steepest descent direction for a bilevel problem without upper-level constraints is done with the help of a linear-quadratic bilevel problem. The sequential linear complementarity problem algorithm proposed in [119] is used to solve such linear-quadratic bilevel problem . The same approach of [177] is applied to convex bilevel programs, where the upper level is quadratic and the lower level is strictly convex quadratic programs, in [194].

Part of the literature is about penalty methods, which consist in solving a series of unconstrained problems, obtained by adding a term, called a penalty function, to the objective function measuring the violation of the constraints while relaxing them. The measure of violation is nonzero when the constraints are violated and is zero when constraints are not violated. Despite the large number of works focused on such methods, they are generally limited to computing stationary points and local minima. While in [14, 179, 15] only the lower level is penalized, and bilevel hierarchy is maintained, in [114] both upper and lower-level objective functions are penalized. In [202], a BLP is converted into a penalized bilinear optimization problem, and an exact penalty algorithm is proposed to find the BLP optimal solution by solving a series of bilinear problems. In a number of works [39, 40, 56], the lower-level problem is replaced by its KKT conditions and, then, a penalized approach is used to solve the single-level problem. Recently, in [140], the authors use the KKT approach and, then, append the complementarity condition to the upper-level objective function with a penalty. The penalized problem is handled using a series of linear programs.

Some papers propose trust-region algorithms to solve bilevel programs. In trust-region methods a certain region of the objective function is approximated using a model function. Such region is expanded if the approximation is adequate, otherwise it is contracted. The first trust-region technique to solve bilevel problems was presented in [136], when the lower-level problem has a strongly convex objective function and linear constraints, and no upper-level constraints are considered. Later, in [143] a bilevel problem is solved with a trust-region approach where the model is itself a bilevel problem with linear upper level and a linear variational inequality at the lower level. Similarly, in [64], the authors consider a bilevel program with no coupling constraints, and the model solved is a linear-quadratic bilevel problem, then reformulated using the KKT approach.

In many real systems, the leader or the follower may have to make discrete decisions. This type of decisions can be described by considering formulations where some variables are restricted to be integer [151]. Branch-and-bound algorithms for solving Bilevel Mixed-Integer Linear Problems (MIBLPs), and mixed-integer quadratic bilevel problems with different assumptions are proposed in [30, 201, 92] and [87], respectively.

To enhance the performance of their basic branch-and-bound method, the authors of [92] also introduce intersection cuts to cut off bilevel infeasible points, thus obtaining a branch-and-cut approach for MIBLPs. Another branch-and-cut algorithm is proposed in [81], for MIBLPs with only integer variables in both levels and no coupling constraints. An extension of the former method that allows for a mixed-integer setting at both levels is given by [185].

A cutting plane method using the Chvátal-Gomory cut for solving a bilevel program with continuous upper level and discrete lower level was proposed in [71]; the algorithm approximates the feasible set of the lower level.

In [107], two deterministic global optimization methods that solve mixed-integer nonlinear bilevel problems are proposed. The first addresses problems in which the upper level is mixed-integer nonlinear and the lower level continuous nonlinear. The second solves problems in which the upper level involves mixed-integer nonlinear functions, and the lower level is mixed-integer nonlinear in upper-level variables, linear polynomial in lower integer variables, and linear in lower continuous variables. This second approach is based on the reformulation of the lower problem as continuous via its convex hull and solving the resulting nonlinear bilevel problem by a novel deterministic global optimization framework.

For the solution of bilevel problems with mixed-integer variables, also Benders decomposition techniques have been employed. Decomposition techniques exploit the decomposable structure of the problems in order to facilitate their solution through the resolution of a series of smaller sub-problems. In [173], the sub-problem is the *slave problem* which is obtained by fixing a number of integer variables of the initial MIBLP to a feasible value. The *master problem*, instead, gives the optimal solution after the addition of Benders cuts. Fixing the integer values makes the slave problem a BLP, which is solved by KKT reformulation techniques, and its solution is used to add a Benders cut to the master problem. The algorithm iterates between master and slave problems until optimality criteria are met. In [45], a Benders-like decomposition is viewed as a procedure for iterative refinement of dual functions associated with the value function of a MIBLP involving integer variables also at the lower level.

In [211], a single-level reformulation and a decomposition algorithm based on a column-and-constraint generation scheme are proposed for MIBLPs with continuous and integer variables in both upper and lower-level programs. In [210], upper-level constraints involving lower-level variables are also considered, and the approach proposed in [211] is enhanced projecting the constraint region on the space of lower-level integer variables and working with KKT conditions of the remaining continuous lower-level problem.

Authors of [89] proposed a global solution approach using a parametric programming theory to address BLPs, Bilevel Quadratic Problems (BQPs) and MIBLPs with binary variables. Indeed, the follower's problem can be solved as a multi-parametric programming problem, with parameters being the upper-level variables. By inserting the resulting exact parametric solutions into the upper level, the overall problem is transformed into a set of independent single-level problems, which can be solved to global optimality. The approach is extended to mixed-integer convex quadratic bilevel programs in [23]. The same authors provide a computational study for MIBLPs and mixed-integer BQPs in [170], where B-POP is presented, a MATLAB toolbox for bilevel optimization through multi-parametric programming. Very recently, in [48] another parametric approach is proposed for linearly constrained bilevel problems in which the upper-level objective function depends on both the lower-level primal and dual optimal solutions. When the upper-level objective is affine in the lower-level primal optimal solution, the parametric function is piece-wise linear. This property facilitates the application of parametric programming and allows for decomposition of a separable lower-level problem. When the upper-level objective is bilinear in the lower-level primal and dual optimal solutions, the authors also provide an exact linearization method that reduces the bilevel problem to a single-level MILP problem.

Obviously, the entire field of bilevel optimization solution methods is broader and we, thus, are not able to cover it entirely in this dissertation. The readers may refer to [79] which contains the largest up-to-date list of references in the field.

1.4 Conclusion

In this first chapter we introduced the field of bilevel programming, which is a useful tool to model real-world problems involving a hierarchical relationship between two decision levels. Indeed bilevel programming has been applied to: military problems, traffic and transportation applications (see, e.g., Chapter 4), management science, production planning, security, as well as energy networks and market (see Chapter 5). We presented the main approaches in this area that deal with theoretical issues, reformulations and algorithms. To close the discussion, in Chapter 2 we discuss the computational complexity of bilevel problems.

Chapter 2

Computational foundations of bilevel programming

In most cases, bilevel problems are Σ_2^{P}-hard. There are classes of strongly NP-hard bilevel problems, but only special cases are proved to be solvable in polynomial time. In this chapter we discuss the computational complexity of bilevel problems. In Section 2.1 we introduce the concepts of polynomial hierarchy, as well as existential and universal quantifiers, used to discuss bilevel programming computational complexity. In Section 2.2 we focus on BLPs, which are strongly NP-hard, and MIBLPs, which are instead Σ_2^{P}-hard. In Section 2.3 we present BLPs solvable in polynomial time. Finally, in Section 2.4 we illustrate the results obtained for the so-called "Independent bilevel problems".

2.1 Polynomial hierarchy and formulations with two classifiers

In order to classify bilevel problems, we must introduce the so-called polynomial hierarchy, a classifying scheme including several levels of complexity. Level zero Σ_0^{P} contains problems solvable in polynomial time, i.e., problems in the well-known class P. Level one includes $\Sigma_1^{\mathsf{P}} = \mathsf{NP}$ and $\Pi_1^{\mathsf{P}} = \text{CO-}\mathsf{NP}$ problems. The second level is the one on which we focus, and contains Σ_2^{P} problems. The generic k^{th} level of the hierarchy includes Σ_k^{P} problems, i.e., problems solvable in nondeterministic polynomial time with an oracle for $\Sigma_{k-1}^{\mathsf{P}}$ problems [146].

We recall here the discussion in [204] on the computational complexity of bilevel programs, which stems from the difference between formulations with a single existential quantifier and formulations including also an universal quantifier.

A problem with a single quantifier in the complexity class NP (or Σ_1^{P}) asks whether

$$\exists x \in \mathcal{X} : P(I, x), \tag{2.1}$$

where I is an instance taken in input, $\mathcal{X}$ denotes the set of potential solutions for instance I, and $P(I, x)$ is a Boolean predicate depending on I and x. We assume that the encoding length of every object x is polynomially bounded in the encoding length of the instance, $|I|$, (i.e., an integer corresponding to the number of symbols required to describe I under some encoding scheme) and that the predicate $P(I, x)$ can be evaluated in an amount of time that is a polynomial function of $|I|$. A problem is NP-hard when every problem in NP can be reduced to it in polynomial time, and is NP-complete if it is NP and NP-hard. A problem is strongly NP-complete if it remains NP-complete even when all of its numerical parameters are bounded by a polynomial in the length of the input problem and is strongly NP-hard if a strongly NP-complete problem has a polynomial reduction to it.

If, for a problem (2.1), the cardinality of set $\mathcal{X}$ is polynomially bounded in $|I|$, all objects $x \in \mathcal{X}$ can be enumerated, checking whether $P(I, x)$ is satisfied, and therefore the problem is solved in polynomial time. Thus, if it is possible to write problem (2.1) as a simpler equivalent one of the form

$$P'(I), \tag{2.2}$$

with $P'(I)$ a property of I testable in polynomial time, such problem belongs to the complexity class P.

The complexity class CO-NP (or Π_1^{P}) contains problems asking whether

$$\forall y \in \mathcal{Y} : P''(I, y), \tag{2.3}$$

where $\mathcal{Y}$ denotes the set of potential solutions for instance I, and $P''(i, y)$ is again a Boolean predicate depending on I and x, that can be evaluated in polynomial time.

If we look at formulations with two quantifiers, we can define Σ_2^{P} complexity class [182] as the class of decision problems having the YES-instances I characterized by a formula of the form

$$\exists x \in \mathcal{X} : \forall y \in \mathcal{Y} \; P(I, x, y). \tag{2.4}$$

Similarly to NP-hard problems, a problem is Σ_2^{P}-hard if every problem in the complexity class Σ_2^{P} can be reduced to it in polynomial time. The most difficult problems in class Σ_2^{P} are the Σ_2^{P}-complete problems: every problem in Σ_2^{P} can be reduced in polynomial time to every Σ_2^{P}-complete problem.

The negated version of a Σ_2^{P} problem is a Π_2^{P} problem, i.e., a problem having YES-instances I characterized by the formulation

$$\forall y \in \mathcal{Y} \; \exists x \in \mathcal{X} : P(I, x, y). \tag{2.5}$$

The class Σ_k^{P} contains the problems that can be expressed by a formula having a sequence of k existential or universal quantifiers, followed by a predicate depending on the variables and on the given instance and that can be evaluated in polynomial time. Looking at formulations (2.4),(2.1),(2.2), we can deduce that:

$$\mathsf{P} \subseteq \mathsf{NP} \subseteq \Sigma_2^{\mathsf{P}} \subseteq \Sigma_3^{\mathsf{P}} \subseteq \cdots \subseteq \Sigma_k^{\mathsf{P}} \subseteq \Sigma_{k+1}^{\mathsf{P}} \subseteq \ldots$$

It is an open question whether these inclusions are strict, but it is widely believed that they all are. If any $\Sigma_k^{\mathsf{P}} = \Sigma_{k+1}^{\mathsf{P}}$, then the hierarchy *collapses* to level k, i.e., for all $j > k$, $\Sigma_j^{\mathsf{P}} = \Sigma_k^{\mathsf{P}}$.

General bilevel problems include both an existing quantifier and an universal quantifier, and aim at finding if it *exists* a solution that is optimal for the leader *for all* the decisions of the followers. Assuming that the encoding length of every potential decision of the leader is polynomially bounded, as well as every potential decision of the follower, and that good final solutions can be recognized in polynomial time, bilevel problems of the described form are contained in class Σ_2^{P}.

2.2 Complexity of BLPs and MIBLPs

In this section, we focus on bilevel optimization problems having only linear functions. Firstly, we discuss the computational results whenever all the variables are continuous, secondly, we consider the problems involving integer variables too.

BLPs are proved to be NP-hard in [117, 34, 27]. Later, in [109] a reduction from the strongly NP-complete Kernel problem [61] is used to show that BLPs are indeed strongly NP-hard. These results are strengthened in [194] (Theorems 5.1 and 5.2), where it is proved that checking strict or local optimality in BLP is also NP-hard, through reductions from 3-SAT problem. Indeed, many combinatorial optimization problems can be reduced to bilevel programs. In [22], it is shown that MILP is a special case of bilevel linear programming, based on the results of [94], where a reformulation of a BLP as a Generalized Linear Complementary Problem (GLCP) is presented, using the KKT conditions of the lower level. A big-M linearization of the KKT complementarity conditions is used to reformulate the GLCP as a MILP problem. In [22], it is shown that this two-stage reformulation has its converse, since the GLCP obtained as a reformulation of a MILP problem in [120] can be further reformulated as a BLP. In fact, every MILP program can be formulated as a BLP using the following equivalence:

$$x \in \{0, 1\} \iff y = 0 \text{ and } y = \arg\max_{w}\{w : w \leqslant x, w \leqslant 1 - x, w \geqslant 0\}.$$

The method used in [94] to reformulate the GLCP (in turn obtained through the KKT reformulation of the BLP) into a MILP problem is based on the linearization of every complementarity constraint (see Section 1.3). It requires the determination of some big-M constant providing a valid bound to the KKT multipliers. Finding a correct big-M cannot be done efficiently, i.e., in polynomial-time, unless $\mathsf{P} = \mathsf{NP}$ [124].

For multilevel linear problems, Jeroslow [117] has shown that the optimal value of a $(k+1)$-level linear program is Σ_k^{P}-hard.

If a subset of either upper or lower-level variables of problem (1.30) are constrained to be integer, the resulting problem is a MIBLP. For such bilevel programs, already checking feasibility of a given vector $(\bar{x}, \bar{y})$ requires solving a MILP formulation corresponding to the lower-level problem for $x = \bar{x}$, i.e., solving a NP-hard problem. As a consequence, using branch-and-bound algorithms to solve a MIBLP will require solving at each node a NP-hard problem (at least for checking feasibility).

The authors of [138] proved that MIBLPs are Σ_2^{P}-hard: there is no way of formulating a MIBLP as a MILP program of polynomial size unless the polynomial hierarchy collapses.

2.3 Polynomially solvable special cases

Despite the results presented in the previous sections, there are special cases in which bilevel problems could be solved in polynomial time. The authors of [136] introduced a polynomial-time algorithm for a BLP where the number of the lower-level control variables is bounded by a constant. In [82] a simpler proof for the above result is proposed, showing that it can be extended to the more general case of multilevel problems where the leader controls all but a constant number of follower's variables.

In [54], the three bilevel variants of knapsack problem presented respectively by Dempe–Richter [78], Mansi–Alves–de-Carvalho–Hanaf [141], and DeNegre [80] are studied. Firstly, they are proved to be Σ_2^{P}-complete under the standard binary encoding of the input. Secondly, the three bilevel formulations are studied under the so-called unary encodings (when an integer n is represented as a string of n ones). The reason behind this approach is that the classical single-level formulation of knapsack problem (known to be NP-complete) is polynomially solvable if the input is encoded in unary. Indeed, the first two bilevel knapsack variants become in this way solvable in polynomial time, whereas the third becomes NP-complete.

2.4 Independent bilevel problems

In this section we focus on the so-called *independent bilevel problems* [203], which are characterized by a feasible set $\mathcal{F}$ independent of the upper-level variables x, that is, for any y, $\mathcal{F}(x,y) = \mathcal{F}(x',y) = \mathcal{F}$ for all $x, x' \in X$. Therefore, these problems are formulated as

$$\begin{aligned} \min_{x,\,y}\ & F(x,y) \\ s.t.\ & G(x,y) \leqslant 0 \\ & y \in \arg\min_y \{f(x,y) \mid y \in \mathcal{F}\} \\ & x \in X. \end{aligned}$$

Note that the lower-level problem of an independent bilevel problem still depends on x because of the lower-level objective function f.

In [203], firstly, it is proved that, if one assumes that X and $\mathcal{F}$ are compact, the independent optimistic and independent pessimistic formulation of a BLP can be solved in polynomial time. The proof is based on the reformulation (using duality theory) of such problems into linear problems of polynomial size in the length of the input data, which can be solved in polynomial time using interior point methods. Secondly, it is shown that even the best-behaved independent formulation of the nonlinear bilevel problem, i.e., linear upper-level problem, X and Y polyhedral, and strictly convex quadratic lower-level objective function, is already strongly NP-hard. The proof of this result is based on a polynomial time reduction to the NP-complete Kernel problem [61].

2.5 Conclusion

The goal of this chapter was to review the most important results on bilevel programming computational complexity. The general bilevel problem is Σ_2^{P}-hard. To arrive to this conclusion we started from the difference between formulations with one quantifier and formulations with two quantifiers, as in [204]. We also presented the main results for both bilevel linear problems, and mixed-integer bilevel linear problems. Finally, we consider the bilevel problems having a lower-level feasible set not depending on the upper-level decision variables, i.e., the *independent bilevel problems*, which we consider in Chapter 3, as well as in the aircraft conflict resolution problem studied in Chapter 4. For the sake of completeness, we also presented some polynomially solvable special bilevel programs.

Chapter 3

Solving a class of bilevel programs with quadratic lower level

In this chapter, we present two theoretical approaches to deal with a particular class of bilevel programs with a quadratic lower-level problem, which can be obtained by reformulating Semi-Infinite Programming (SIP) problems with an infinite number of quadratically parametrized constraints. The problems considered in our analysis are what we defined in Chapter 2 "independent bilevel problems", i.e., problems having a lower-level feasible set not depending on the upper-level variables. We propose a new approach to solve this class of bilevel programs, based on the dual of the lower-level problem, which can lead to a convex or a semidefinite programming problem, depending on the parametrization of the lower level with respect to the upper-level variables. This approach is compared with a new tailored Cutting Plane (CP) algorithm, which is proved to be convergent. The rate of convergence of this CP algorithm, directly related to the iteration index, is derived when the upper-level objective function is strongly convex, and under a strict feasibility assumption. We successfully test the two proposed methods on two applications: the constrained quadratic regression and a zero-sum game with cubic payoff.

The results presented in this chapter are contained in the submitted paper [7].

3.1 Introduction

We consider bilevel problems where the upper-level problem has a continuous convex objective function $F(x)$ (where x is an array of upper-level decision variables), and a convex feasible set $\mathcal{X} \subset \mathbb{R}^m$ depending only on x. The lower-level problem is a quadratic problem in the lower-level decision variables y, with a possibly nonconvex objective function, but with a feasible set consisting of the polytope

$$\mathcal{F} = \{y \in \mathbb{R}^n : Ay \leqslant b\} = \{y \in \mathbb{R}^n : \forall j \leqslant r,\ (a_j^\top y \leqslant b_j)\},$$

where a_j is the j-th row of the matrix A, and r is the number of lower-level constraints.

We make two overarching assumptions on the bilevel class of interest: (i) $\mathcal{F}$ does not depend on x; (ii) the upper-level problem depends only on the optimal objective function value of the lower-level problem, rather than its optimal solutions.

Thus, the Mathematical Programming formulation we study is as follows:

$$\left\{\begin{array}{rl} \min\limits_{x\in\mathbb{R}^m} & F(x) \\ \text{s.t.} & x \in \mathcal{X} \\ & d(x) \leqslant \min\limits_{y\in\mathbb{R}^n}\{\frac{1}{2}y^\top Q(x)y + q(x)^\top y \mid Ay \leqslant b\}, \end{array}\right. \tag{BP}$$

where F, and d, are continuous convex functions in the upper-level variables x, both the $n \times n$ matrix $Q(x)$ and the n-dimensional vector $q(x)$ depend linearly on x, A a $r \times n$ matrix, and b a r-dimensional vector.

Here are the technical assumptions we make on (BP).

Assumption 1. *The upper-level objective function $x \mapsto F(x)$ is convex.*

Assumption 2. *$\mathcal{X}$ is convex.*

Assumption 3. *The functions $x \mapsto q(x)$ and $x \mapsto Q(x)$ are linear.*

Assumption 4. *The function $x \mapsto d(x)$ is convex and Lipschitz continuous.*

Assumption 5. *The set $\mathcal{F}$ is compact, and a scalar $\rho > 0$ is known such that (s.t.) the set $\mathcal{F}$ is included in the centered ℓ_2-ball with radius ρ.*

As mentioned above, (BP) does not consider the optimal solutions of the lower-level problem, but only its optimal objective function value. This renders "pessimistic" or "optimistic" interpretations of (BP) meaningless. The bilevel class (BP) arises in many applications requiring SIP problems, i.e., optimization problems with a finite number of variables, and an infinite number of parametrized constraints of the type $\forall y \in Y,\ g(x, y) \geqslant 0$, where y is the parameter. Indeed, this is equivalent to:

$$0 \leqslant \min_{y\in Y} g(x, y),$$

which allows the reformulation of the SIP constraints into a lower level of a bilevel problem in the class (BP), as long as $g(x, y) = \frac{1}{2}y^\top Q(x)y + q(x)^\top y - d(x)$ and $Y = \mathcal{F}$. We remark that, in a bilevel context, the function $\phi(x) = \min\limits_{y\in Y} g(x, y)$ is called *optimal value function.*

Our first contribution is an analysis of (BP) which yields a single-level formulation with a finite number of constraints. This single-level formulation is obtained by dualizing, using Semidefinite Programming (SDP), the problem $\min_{y \in Y} g(x, y)$, i.e., the problem of finding the most violated constraint among the infinite number of constraints of the corresponding SIP problem. If $g(x, y)$ is convex in y, i.e., if $Q(x)$ is Positive Semidefinite (PSD), our single-level is a reformulation of (BP). This analysis yields a new solution approach, consisting in solving the single-level formulation. We note that, if $g(x, y)$ were linear in y, our reformulation would be the same as the one mentioned in [36, Section 1.3]. Although an extension to nonlinear perturbations is briefly outlined in [36, Section 1.4], the specific case of quadratic perturbations over an uncertainty polytope is not considered.

Our second contribution is a tailored CP algorithm. While such algorithms are well known in SIP, we prove its convergence and derive a new convergence rate in terms of the number of iterations, under the additional assumptions that F is strongly convex and that there exists an upper-level solution strictly satisfying the constraint involving the lower-level problem.

The rest of the chapter is organized as follows. We review the relevant literature in Section 3.2. A single-level restriction/reformulation of problem (BP) is introduced and discussed in Section 3.3. A tailored CP algorithm for solving formulation (BP) directly is presented in Section 3.4. We successfully test the two proposed methods on two applications: the constrained quadratic regression and a zero-sum game with cubic payoff. Such applications are introduced in Section 3.5. Numerical results, obtained by applying both solution approaches to these applications, are presented in Section 3.6: our results illustrate the interest of the proposed method. Some concluding comments are given in Section 3.7.

3.2 Literature review

BQPs are bilevel problems having either one or both the objective functions which can be expressed as quadratic functions. In [29] a BQP having a linear upper-level problem and a convex quadratic lower level is considered, and a branch-and-bound algorithm to solve it is presented. In [209], an ergodic branch-and-bound method is introduced to solve mixed-integer BQPs, having a convex lower-level problem, which is thus replaced by its KKT optimality conditions. In [186], a more general class of BQPs is considered, by allowing some (not necessarily convex) quadratic upper-level constraints and some convex quadratic functions in lower-level constraints. After the reformulation of the problem into a nonconvex quadratic single-level problem by replacing its lower level by its KKT conditions (which is possible as they assume to know a sufficiently large number that bounds the Lagrange multipliers) the authors adopt the successive convex relaxation method given by Kojima and Tunçel in [127] for approximating the nonconvex feasible region. Then, they present two types of techniques to enhance the efficiency of the method used.

A part of the literature focuses on general nonlinear bilevel problems. For example, in [145], the authors aim at solving bilevel mixed-integer optimization problems with lower-level integer variables and including nonlinear terms. They assume that, for any fixed upper-level variables, and lower-level integer variables, the lower-level problem is convex and satisfies the Slater condition. In order to solve these bilevel problems, the authors consider an approximate projection-based algorithm for mixed-integer linear bilevel programming problems introduced by Yue et al. [210] and propose a way of making it exact under the additional assumption that continuous upper-level variables do not appear in lower-level constraints.

A nonconvex lower-level problem is considered in both [132, 148], as well as in [25]. In particular, in [132] a bilevel problem having closed convex feasible sets both in the upper and in the lower level (the lower-level one assumed not dependent on the upper-level variables), but eventually nonconvex objective functions in both levels is reformulated into a single-level problem, using the so-called optimal value function transformation. To deal with the non-smoothness introduced by the optimal value function, a smoothing projected gradient algorithm is proposed and used to solve the bilevel problem if a calmness condition holds, which is a strong assumption, and an approximate bilevel program otherwise.

In [148], a bounding algorithm for the global solution of nonlinear bilevel programs involving nonconvex functions in both the upper and lower levels is presented. The algorithm is rigorous and terminates finitely to a point that satisfies ϵ-optimality in both upper and lower-level problems. This is possible using the optimal value function of the lower-level problem and a piecewise, yet discontinuous, approximation of it. Previously, Bard [25] proposed an algorithm (not guaranteed to be convergent) based on a grid search between a lower and an upper bound of the optimal value of a bilevel problem (max-max) without upper-level constraints. The upper bound is found by solving a relaxation obtained replacing the lower level with its KKT conditions. The lower bound is obtained solving the lower level for a fixed value of the upper-level variables (i.e., $x = x_0$), and then computing the value of the upper-level function in the point $(x_0, \phi(x_0))$.

This chapter focuses on a particular class of bilevel problems, where there is no *argmin* operator, but a constraint in the upper level involving the lower-level problem's value. As mentioned before, such bilevel programs can be obtained by reformulating SIP problems having an infinite number of quadratically parametrized constraints. To solve SIP problems, discretization methods, CP methods, and other hybrid methods are used in the literature. The discretization approach [110, 181] consists in replacing the infinite constraint parameter set by a finite subset which samples it finely: this leads to a relaxation of the original problem, the value of which converges towards the value of the original problem when the mesh gets finer. This method is commonly used for parameters sets of low dimensions, but deals with the curse of dimensionality when the number of parameters increases. Instead of using a fixed subset of constraints, the CP approach [121] consists in iteratively generating and adding constraints. The CP algorithm and its refined variants, as the accelerated central CP algorithm for instance, are major techniques used for solving linear, quadratic, and convex SIP problems [129, 88, 38].

In this chapter, we introduce a tailored CP algorithm which directly solve formulation (BP), and we prove that it is convergent. We also do a step further, by proving a rate of convergence for CP valid for a specific setting. Our convergence rate is directly related to the iteration index k, which is something new w.r.t. what is usually proved in SIP literature, where the linear rate of convergence is related to an index which is not *controlled* by the index k (see [162, Theorem 4.3]).

Another class of algorithms for SIP is based on Lagrangian penalty functions and Trust-Region methods [67, 187]. However, in the context of problem (BP), they would require to compute the set of all local minima of problem $\min_{y \in Y} g(x, y)$. In the case where g is not convex with respect to variables y, the enumeration of all local minima is intractable even for medium-scale instances.

3.3 Single-level formulation via the dual approach

A possible way to deal with the bilevel problem (BP) is what we call *dual approach*, which consists in replacing the constraint involving the quadratic lower-level problem with one involving its dual. We obtain a strong dual from an SDP relaxation of the lower-level problem (or a reformulation if the latter is convex). We recall that the lower-level problem of (BP), for any $x \in \mathcal{X}$, reads:

$$\begin{cases} \min\limits_{y \in \mathbb{R}^n} & \frac{1}{2} y^\top Q(x) y + q(x)^\top y \\ \text{s.t.} & a_j^\top y \leqslant b_j, \quad \forall j \in \{1, \dots, r\}, \end{cases} \tag{P_x}$$

where the objective function $f(x, y) = \frac{1}{2} y^\top Q(x) y + q(x)^\top y$ is convex if $Q(x)$ is PSD. In Section 3.3.1, we introduce the classical SDP relaxation (reformulation, if the lower level is convex) of the lower-level problem regularized by a ball constraint and then, in Section 3.3.2, we introduce the SDP dual of this relaxation (reformulation resp.). Finally, in Section 3.3.3 we present a single-level formulation obtained applying the so-called dual approach to the bilevel problem (BP). This formulation is a reformulation of (BP) if $Q(x)$ is PSD for any $x \in \mathcal{X}$. Otherwise, it is a restriction.

3.3.1 SDP relaxation/reformulation of the lower-level problem

In this section, we reason for any fixed value of the upper-level decision vector $x \in \mathcal{X}$. Let us define the following matrices:

- $\mathcal{Q}(x) = \frac{1}{2}\begin{pmatrix} Q(x) & q(x) \\ q(x)^\top & 0 \end{pmatrix}$,
- $\mathcal{A}_j = \frac{1}{2}\begin{pmatrix} 0_n & a_j \\ a_j^\top & 0 \end{pmatrix}, \quad \forall j \in \{1, \dots, r\}$,

where 0_n is the $n \times n$ null matrix. We denote by $\langle A, B\rangle = \mathsf{Tr}(A^\top B)$ the Froebenius product of two square matrices A and B with same size. With this notation, under Assumption 5, the problem

$$\begin{cases} \min\limits_{Y \in \mathbb{R}^{(n+1)\times(n+1)}} & \langle \mathcal{Q}(x), Y\rangle \\ \text{s.t.} & \langle \mathcal{A}_j, Y\rangle \leqslant b_j \quad \forall j \in \{1, \dots, r\} \\ & \mathsf{Tr}(Y) \leqslant 1 + \rho^2 \\ & Y_{n+1,n+1} = 1 \\ & Y \succeq 0 \\ & \mathsf{rank}(Y) = 1, \end{cases} \tag{3.1}$$

is a reformulation of (P_x), because any feasible matrix Y has the form $Y = \begin{pmatrix} y \\ 1 \end{pmatrix}\begin{pmatrix} y \\ 1 \end{pmatrix}^\top$ with $y \in \mathcal{F}$, and, therefore, $\langle \mathcal{Q}(x), Y\rangle = f(x, y)$. The constraint $\mathsf{Tr}(Y) \leqslant 1 + \rho^2$, derives from Assumption 5 as follows:

$$\|y\|_2^2 \leqslant \rho^2 \Leftrightarrow \mathsf{Tr}(yy^\top) \leqslant \rho^2 \Leftrightarrow \mathsf{Tr}(Y) \leqslant \rho^2 + 1,$$

being $\mathsf{Tr}(Y) = \mathsf{Tr}(yy^\top) + 1$. This constraint does not play any role at this point, but will be useful thereafter to come up with a dual SDP problem with no duality gap (see Section 3.3.2). If we relax the nonconvex constraint $\mathsf{rank}(Y) = 1$ in (3.1), we obtain:

$$\begin{cases} \min\limits_{Y \in \mathbb{R}^{(n+1)\times(n+1)}} & \langle \mathcal{Q}(x), Y\rangle \\ \text{s.t.} & \langle \mathcal{A}_j, Y\rangle \leqslant b_j \quad \forall j \in \{1, \dots, r\} \\ & \mathsf{Tr}(Y) \leqslant 1 + \rho^2 \\ & Y_{n+1,n+1} = 1 \\ & Y \succeq 0, \end{cases} \tag{SDP_x}$$

which is a SDP relaxation of (P_x), as proved in the following Lemma 1. If $Q(x)$ is PSD, Lemma 1 states that (SDP_x) is a reformulation of (P_x), the rank-constraint relaxation notwithstanding.

Lemma 1. *Under Assumption 5,* $\mathsf{val}(\mathsf{SDP}_x) \leqslant \mathsf{val}(\mathsf{P}_x)$*. If* $Q(x)$ *is PSD, then* $\mathsf{val}(\mathsf{SDP}_x) = \mathsf{val}(\mathsf{P}_x)$*.*

For a sake of completeness, we give a proof of this standard lemma.

Proof. The inequality $\mathsf{val}(\mathsf{SDP}_x) \leqslant \mathsf{val}(\mathsf{P}_x)$ follows from the relaxation of the rank-constraint. We now assume that $Q(x)$ is PSD and prove that $\mathsf{val}(\mathsf{SDP}_x) \geqslant \mathsf{val}(\mathsf{P}_x)$ holds. Given a matrix Y feasible for (SDP_x), we denote by $u_1, \ldots, u_{n+1} \in \mathbb{R}^{n+1}$ a basis of eigenvectors of Y (which is PSD) and their respective eigenvalues $v_1, \ldots, v_{n+1} \in \mathbb{R}_+$. Let us introduce the two following index sets:

$$I = \{i \in \{1, \ldots, n+1\} : (u_i)_{n+1} \neq 0\} \text{ and } J = \{i \in \{1, \ldots, n+1\} : (u_i)_{n+1} = 0\}.$$

We have then: $I \cup J = \{1, \ldots, n+1\}$. Moreover,

- if $i \in I$: we define the nonnegative scalar $\mu_i = v_i\,(u_i)_{n+1}^2$ and $y_i \in \mathbb{R}^n$ s.t. $u_i = (u_i)_{n+1} \begin{pmatrix} y_i \\ 1 \end{pmatrix}$
- if $i \in J$: we define the nonnegative scalar $\nu_i = v_i$ and $z_i \in \mathbb{R}^n$ s.t. $u_i = \begin{pmatrix} z_i \\ 0 \end{pmatrix}$.

With this notation, we have that

$$\begin{aligned} Y = \sum_{i=1}^{n+1} v_i u_i u_i^\top &= \sum_{i \in I} v_i (u_i)_{n+1}^2 \begin{pmatrix} y_i \\ 1 \end{pmatrix} \begin{pmatrix} y_i \\ 1 \end{pmatrix}^\top + \sum_{i \in J} v_i \begin{pmatrix} z_i \\ 0 \end{pmatrix} \begin{pmatrix} z_i \\ 0 \end{pmatrix}^\top \\ &= \sum_{i \in I} \mu_i \begin{pmatrix} y_i y_i^\top & y_i \\ y_i^\top & 1 \end{pmatrix} + \sum_{i \in J} \nu_i \begin{pmatrix} z_i z_i^\top & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{pmatrix}, \end{aligned}$$

where $\mathbf{0}$ is the null n-dimensional vector, not to be confused with 0_n, the $n \times n$ null matrix. Let us define the vector $\bar{y} = \sum_{i \in I} \mu_i y_i$. Its objective value in (P_x) is smaller than the objective value of Y in (SDP_x). In fact:

$$\langle \mathcal{Q}(x), Y \rangle = \sum_{i \in I} \mu_i f(x, y_i) + \frac{1}{2} \sum_{i \in J} \nu_i z_i^\top Q(x) z_i \geqslant \sum_{i \in I} \mu_i f(x, y_i) \geqslant f(x, \sum_{i \in I} \mu_i y_i) = f(x, \bar{y}). \tag{3.2}$$

The first inequality is due to $Q(x) \succeq 0$ and $\nu_i \geqslant 0$. The second inequality derives from $\sum_{i \in I} \mu_i = Y_{n+1,n+1} = 1$, and from the convexity of function f_x (Jensen inequality). Moreover, since Y is feasible in (SDP_x), for each $j \in \{1, \ldots, r\}$ we have $b_j \geqslant \langle \mathcal{A}_j, Y \rangle = \sum_{i \in I} \mu_i a_j^\top y_i = a_j^\top \bar{y}$, which means that $\bar{y}$ is feasible in (P_x) too. This implies that $f(x, \bar{y}) \geqslant \mathsf{val}(\mathsf{P}_x)$ and together with (3.2), that $\langle \mathcal{Q}(x), Y \rangle \geqslant \mathsf{val}(\mathsf{P}_x)$. This being true for any matrix Y feasible in (SDP_x), we conclude that $\mathsf{val}(\mathsf{SDP}_x) \geqslant \mathsf{val}(\mathsf{P}_x)$. This proves that $\mathsf{val}(\mathsf{SDP}_x) = \mathsf{val}(\mathsf{P}_x)$. □

3.3.2 Dual SDP problem

As already done in Section 3.3.1, also in this section we reason for any fixed value of $x \in \mathcal{X}$. Let E be a $(n+1)\times(n+1)$ matrix s.t. $E_{n+1,n+1} = 1$ and $E_{ij} = 0$ everywhere else. Let I_{n+1} be the $(n+1)\times(n+1)$ identity matrix. The following SDP problem

$$\begin{cases} \max\limits_{\lambda \in \mathbb{R}^r_+,\, \alpha \in \mathbb{R}_+,\, \beta \in \mathbb{R}} & -b^\top \lambda - \alpha(1+\rho^2) - \beta \\ \text{s.t.} & \mathcal{Q}(x) + \sum\limits_{j=1}^r \lambda_j \mathcal{A}_j + \alpha I_{n+1} + \beta E \succeq 0, \end{cases} \tag{DSDP_x}$$

is the dual of problem (SDP_x), as the following proposition states.

Proposition 1. *Formulations* (SDP_x) *and* (DSDP_x) *are a primal-dual pair of SDP problems and strong duality holds, i.e.,* $\mathsf{val}(\mathsf{SDP}_x) = \mathsf{val}(\mathsf{DSDP}_x)$.

Proof. The Lagrangian of problem (SDP_x) is defined over $Y \in S^+_{n+1}(\mathbb{R})$, $\lambda \in \mathbb{R}^r_+$, $\alpha \in \mathbb{R}_+$, $\beta \in \mathbb{R}$ and reads

$$\begin{aligned} L_x(Y,\lambda,\alpha,\beta) &= \langle \mathcal{Q}(x), Y\rangle + \sum_{j=1}^r \left[\lambda_j\left(\langle \mathcal{A}_j, Y\rangle - b_j\right)\right] + \alpha(\mathsf{Tr}(Y) - 1 - \rho^2) + \beta(Y_{n+1,n+1} - 1) \\ &= -\sum_{j=1}^r \lambda_j b_j - \alpha(1+\rho^2) - \beta + \left\langle \mathcal{Q}(x) + \sum_{j=1}^r \lambda_j \mathcal{A}_j + \alpha I_{n+1} + \beta E,\, Y \right\rangle. \end{aligned}$$

The Lagrangian dual problem of (SDP_x) is:

$$\max_{\substack{\lambda \in \mathbb{R}^r_+ \\ \alpha \in \mathbb{R}_+ \\ \beta \in \mathbb{R}}} \; \min_{Y \in S^+_{n+1}(\mathbb{R})} L_x(Y,\lambda,\alpha,\beta).$$

According to equality above, it can thus be written as

$$\max_{\substack{\lambda \in \mathbb{R}^r_+ \\ \alpha \in \mathbb{R}_+ \\ \beta \in \mathbb{R}}} \left(-\left(\sum_{j=1}^r \lambda_j b_j + \alpha(1+\rho^2) + \beta \right) + \min_{Y \in S^+_{n+1}(\mathbb{R})} \left\langle \mathcal{Q}(x) + \sum_{j=1}^r \lambda_j \mathcal{A}_j + \alpha I_{n+1} + \beta E, Y \right\rangle \right).$$

We notice that

$$\min_{Y \in S^+_{n+1}(\mathbb{R})} \left\langle \mathcal{Q}(x) + \sum_{j=1}^r \lambda_j \mathcal{A}_j + \alpha I_{n+1} + \beta E,\, Y \right\rangle = \begin{cases} 0 & \text{if } \left(\mathcal{Q}(x) + \sum\limits_{j=1}^r \lambda_j \mathcal{A}_j + \alpha I_{n+1} + \beta E\right) \succeq 0 \\ -\infty & \text{otherwise.} \end{cases}$$

This proves that the dual problem of (SDP_x) reads

$$\begin{cases} \max\limits_{\lambda \in \mathbb{R}^r_+,\, \alpha \in \mathbb{R}_+,\, \beta \in \mathbb{R}} & -b^\top \lambda - \alpha(1+\rho^2) - \beta \\ \text{s.t.} & \mathcal{Q}(x) + \sum\limits_{j=1}^r \lambda_j \mathcal{A}_j + \alpha I_{n+1} + \beta E \succeq 0, \end{cases}$$

which is the formulation (DSDP_x). To prove that $\mathsf{val}(\mathsf{SDP}_x) = \mathsf{val}(\mathsf{DSDP}_x)$, we prove that the Slater condition holds for the dual problem (DSDP_x), exploiting the Lagrangian multiplier associated to the constraint $\mathsf{Tr}(Y) \leqslant 1+\rho^2$. In fact, the Slater condition is a sufficient condition for strong duality [192]. We denote by m_x the minimum eigenvalue of $\mathcal{Q}(x)$. By definition of m_x, the matrix $\mathcal{Q}(x) + (1-m_x)I_{n+1}$ is positive definite. This is why $(\lambda, \alpha, \beta) = (0, \ldots, 0, 1-m_x, 0)$ is a strictly feasible point of (DSDP_x). Hence, the Slater condition holds. □

3.3.3 SDP restriction/reformulation of the bilevel problem

Leveraging on Section 3.3.1 and Section 3.3.2, which focus on the lower-level problem (P_x), its SDP relaxation (SDP_x) and the respective dual problem (DSDP_x), we propose a single-level restriction of the bilevel programming problem (BP). It is a reformulation of (BP) if $Q(x)$ is PSD for any $x \in \mathcal{X}$.

Theorem 1. *The single-level formulation*

$$\begin{cases} \min\limits_{x,\lambda,\alpha,\beta} & F(x) \\ s.t. & x \in \mathcal{X} \\ & d(x) \leqslant -\lambda^\top b - \alpha(1+\rho^2) - \beta \\ & \mathcal{Q}(x) + \sum_j \lambda_j \mathcal{A}_j + \alpha I_{n+1} + \beta E \succeq 0 \\ & x \in \mathbb{R}^m, \lambda \in \mathbb{R}^r_+,\ \alpha \in \mathbb{R}_+, \beta \in \mathbb{R}, \end{cases} \tag{BPR}$$

is a restriction of the bilevel programming problem (BP)*. If $Q(x)$ is PSD for any $x \in \mathcal{X}$,* (BPR) *is a reformulation of* (BP)*.*

Proof. Being $\mathsf{Feas}(\mathsf{BP})$ and $\mathsf{Feas}(\mathsf{BPR})$ the feasible sets of (BP) and (BPR) respectively, since (BP) and (BPR) share the same objective function, proving the following implication for any $x \in \mathbb{R}^m$

$$\big(\exists\ \lambda \in \mathbb{R}^r_+,\ \alpha \in \mathbb{R}_+,\ \beta \in \mathbb{R} :\ (x, \lambda, \alpha, \beta) \in \mathsf{Feas}(\mathsf{BPR})\big) \implies x \in \mathsf{Feas}(\mathsf{BP}), \tag{3.3}$$

will prove the first part of the theorem. For any $x \in \mathcal{X}$, we have:

$$d(x) \leqslant \mathsf{val}(\mathsf{SDP}_x) \Longrightarrow d(x) \leqslant \mathsf{val}(\mathsf{P}_x) \iff x \in \mathsf{Feas}(\mathsf{BP}), \tag{3.4}$$

where the first implication stems from Lemma 1, which stipulates that $\mathsf{val}(\mathsf{SDP}_x) \leqslant \mathsf{val}(\mathsf{P}_x)$. Applying Proposition 1, we obtain that:

$$d(x) \leqslant \mathsf{val}(\mathsf{SDP}_x) \iff d(x) \leqslant \mathsf{val}(\mathsf{DSDP}_x). \tag{3.5}$$

For any $x \in \mathcal{X}$, we have that

$$d(x) \leqslant \mathsf{val}(\mathsf{DSDP}_x) \iff \exists\, \lambda \in \mathbb{R}^r_+,\ \alpha \in \mathbb{R}_+,\ \beta \in \mathbb{R} : \begin{cases} d(x) \leqslant -\lambda^\top b - \alpha(1+\rho^2) - \beta \\ \mathcal{Q}(x) + \sum_{j=1}^{r} \lambda_j \mathcal{A}_j + \alpha I_{n+1} + \beta E \succeq 0 \end{cases} \tag{3.6}$$

The equivalence (3.6) just expresses the fact that the maximization problem (DSDP_x) has a value exceeding $d(x)$ if and only if it has a feasible solution with value exceeding $d(x)$. Hence, from (3.5), and (3.6), the following equivalences hold:

$$d(x) \leqslant \mathsf{val}(\mathsf{SDP}_x) \iff \exists\, \lambda \in \mathbb{R}^r_+,\ \alpha \in \mathbb{R}_+,\ \beta \in \mathbb{R} : \begin{cases} d(x) \leqslant -\lambda^\top b - \alpha(1+\rho^2) - \beta \\ \mathcal{Q}(x) + \sum_{j=1}^{r} \lambda_j \mathcal{A}_j + \alpha I_{n+1} + \beta E \succeq 0 \end{cases} \tag{3.7}$$

$$\iff \exists\, \lambda \in \mathbb{R}^r_+,\ \alpha \in \mathbb{R}_+,\ \beta \in \mathbb{R},\ (x, \lambda, \alpha, \beta) \in \mathsf{Feas}(\mathsf{BPR}).$$

The equivalence (3.7), together with implication (3.4), proves the implication (3.3).

If $Q(x)$ is PSD for any $x \in \mathcal{X}$, we can replace the implication (3.4) by the equivalence

$$d(x) \leqslant \mathsf{val}(\mathsf{SDP}_x) \iff d(x) \leqslant \mathsf{val}(\mathsf{P}_x) \iff x \in \mathsf{Feas}(\mathsf{BP}). \tag{3.8}$$

This, together with equivalence (3.7), proves that

$$\exists\, \lambda \in \mathbb{R}^r_+,\ \alpha \in \mathbb{R}_+,\ \beta \in \mathbb{R} :\ (x, \lambda, \alpha, \beta) \in \mathsf{Feas}(\mathsf{BPR}) \iff x \in \mathsf{Feas}(\mathsf{BP}),$$

meaning that (BPR) is a reformulation of (BP), since the objective function is the same. □

Assumptions 1, 2, 3, and 4 implies that the single-level problem (BPR) is convex. Let us recall the following definition of *semidefinite representable* (SDr) functions.

Definition 1 ([172]). *A convex (resp. concave) function f is SDr if and only if its epigraph, i.e., $(t, x) : f(x) \leqslant t$ (resp. the hypograph $(t, x) : t \leqslant f(x)$), is SDr [37].*

Thus, we further remark that formulation (BPR) is a SDP problem if set $\mathcal{X}$ is SDr, as well as functions $F(x)$, and $d(x)$.

3.4 Cutting plane algorithm

In order to benchmark the results and the performance of the single-level approach proposed in Section 3.3, we introduce in this section a CP algorithm for solving the bilevel formulation (BP) directly. We also include a proof of convergence for this algorithm in Section 3.4.1 for sake of completeness, even if the convergence of these methods is well-known in SIP literature. We prove a convergence rate in Section 3.4.2, obtained by introducing a dual view of the CP algorithm. We make the following further assumption on the set $\mathcal{X}$:

Assumption 6. *The set $\mathcal{X}$ is compact.*

Algorithm 1 CP algorithm for (BP)

1: Let $h = 0$. Initialize the relaxation R_h of the bilevel problem (BP), obtained by considering the upper-level problem only.
2: **while** true **do**
3: Solve R_h, obtaining an optimal solution x^h.
4: Compute an optimal solution y^h of the lower-level problem for $x = x^h$.
5: **if** $d(x^h) \leqslant \frac{1}{2}(y^h)^\top Q(x^h)y^h + q(x^h)^\top y^h$ **then**
6: Return (x^h, y^h).
7: **else**
8: Define R_{h+1} as R_h with the additional inequality:

$$d(x) \leqslant \frac{1}{2}(y^h)^\top Q(x)y^h + q(x)^\top y^h. \tag{3.9}$$

9: $h := h + 1$
10: **end if**
11: **end while**

At the first iteration of Algorithm 1, the relaxed problem R_0 is given by:

$$\min_{x \in \mathcal{X}} \quad F(x), \tag{3.10}$$

which considers minimizing the upper-level objective function subject to the upper-level constraints only. This problem has a finite value according to the compactness of set $\mathcal{X}$.

At each iteration, Algorithm 1 defines the feasible set of the upper-level problem by means of cuts in the upper-level variables x. The resulting R_h problems are relaxations of (BP), and their feasible sets are decreasing in the sense of the inclusion, bounded, because included in the feasible set of R_0, and closed as intersections of closed sets. Thus, each problem R_h admits a minimum. Moreover, the sequence $(F(x^h))$ is increasing, and $F(x^h) \leqslant \mathsf{val}(\text{BP})$ holds for any h. At step 4, the problem solved to find a new cutting plane is

$$\min_{y \in \mathbb{R}^n} \{\frac{1}{2} y^\top Q(x^h) y + q(x^h)^\top y \mid Ay \leqslant b\}. \tag{P_{x^h}}$$

This problem is a quadratic program that is either convex or nonconvex depending on the positive semi-definiteness of the constant matrix $Q(x^h)$. In order to find global optima of (P_{x^h}), regardless of the definiteness of $Q(x^h)$ (in turn depending on the value of x^h), a global optimization algorithm should be employed. Step 6 returns the optimal solution of the bilevel formulation (BP).

3.4.1 Convergence proof

In this section, a convergence proof for Algorithm 1 is given. First of all, let us define the negative part of a function f as $f^- := \max(0, -f)$. Since $Q(x)$ and $q(x)$ are linear w.r.t. x, the function $f : (x, y) \mapsto \frac{1}{2} y^\top Q(x) y + q(x)^\top y$ is continuously differentiable, and therefore Lipschitz-continuous on the compact set $\mathcal{X} \times \mathcal{F}$ (see Assumption 5 and 6), with $L > 0$ an associated Lipschitz constant.

Moreover, $x \mapsto \mathsf{val}(P_x)$ is continuous. To show this, let us consider any $\omega > 0$ and any pair $(x, \tilde{x}) \in \mathcal{X}^2$ s.t. $\|x - \tilde{x}\| \leqslant \frac{\omega}{L}$. We define $y \in \mathcal{F}$ an optimal solution of (P_x), i.e., $\mathsf{val}(P_x) = f(x, y)$, and $\tilde{y} \in \mathcal{F}$ an optimal solution of $(P_{\tilde{x}})$, i.e., $\mathsf{val}(P_{\tilde{x}}) = f(\tilde{x}, \tilde{y})$. By definition of $\mathsf{val}(P_{\tilde{x}})$ and using the Lipschitz continuity of f, we know that

$$\mathsf{val}(P_{\tilde{x}}) \leqslant f(\tilde{x}, y) \leqslant f(x, y) + L \left\| \begin{pmatrix} x - \tilde{x} \\ y - y \end{pmatrix} \right\| \leqslant \mathsf{val}(P_x) + L \|x - \tilde{x}\| \leqslant \mathsf{val}(P_x) + \omega,$$

and, symmetrically, that

$$\mathsf{val}(P_x) \leqslant f(x, \tilde{y}) \leqslant f(\tilde{x}, \tilde{y}) + L \left\| \begin{pmatrix} x - \tilde{x} \\ \tilde{y} - \tilde{y} \end{pmatrix} \right\| \leqslant \mathsf{val}(P_{\tilde{x}}) + L \|x - \tilde{x}\| \leqslant \mathsf{val}(P_{\tilde{x}}) + \omega.$$

Thus, $|\mathsf{val}(P_x) - \mathsf{val}(P_{\tilde{x})}| \leqslant \omega$, which proves that the value function $x \mapsto \mathsf{val}(P_x)$ is continuous at any $x \in \mathcal{X}$. Based on these observations, we prove the convergence of the algorithm.

Theorem 2. *Under Assumptions 5 and 6 Algorithm 1 either terminates in $H \in \mathbb{N}^\star$ iterations, in which case x^h is the solution of* (BP), *or generates an infinite sequence $(x^h)_{h \in \mathbb{N}^\star}$ with the following convergence guarantees:*

- *feasibility error: $\epsilon_h = \left(\mathsf{val}(P_{x^h}) - d(x^h)\right)^- \to 0$,*
- *objective error: $\delta_h = \mathsf{val}(\mathsf{BP}) - F(x^h) \to 0$.*

Proof. If Algorithm 1 terminates at iteration $H \in \mathbb{N}^\star$, x^H is feasible in (BP), i.e., $x^H \in \mathcal{X}$ and $\mathsf{val}(P_{x^H}) \geqslant d(x^H)$, which implies that $F(x^H) \geqslant \mathsf{val}(\mathsf{BP})$. At the same time $F(x^H) = \mathsf{val}(R_H) \leqslant \mathsf{val}(\mathsf{BP})$, being R_H a relaxation of (BP) by definition. Thus, $F(x^H) = \mathsf{val}(\mathsf{BP})$, and x^H is an optimal solution of (BP).

Let us suppose now that the stopping test is never satisfied. In this context, we prove first the convergence of the feasibility error ϵ_h towards 0. For any $h \in \mathbb{N}^\star$, we have that $\mathsf{val}(P_{x^h}) = \frac{1}{2} y^{h\top} Q(x^h) y^h + q(x^h)^\top y^h = f(x^h, y^h)$, thus $\epsilon_h = \left(f(x^h, y^h) - d(x^h)\right)^-$. Since f, d and the negative part function are continuous, and since both x^h and y^h are bounded, the sequence ϵ_h is also bounded. According to Bolzano-Weierstrass theorem [10], this bounded sequence has at least a convergent sub-sequence. In the following, we define any convergent sub-sequence extracted from ϵ_h as $\epsilon_{\psi_0(h)}$, where $\psi_0 : \mathbb{N}^\star \mapsto \mathbb{N}^\star$ is an increasing application. Defining as $\epsilon_* \in \mathbb{R}$ the limit of this convergent sub-sequence, we will show that this limit value is in fact 0.

The sequence $\left(y^{\psi_0(h)}, \epsilon_{\psi_0(h)}\right)$ is a sub-sequence of the bounded sequence (y^h, ϵ_h), therefore it is bounded. According to the Bolzano-Weierstrass theorem, the sequence $\left(y^{\psi_0(h)}, \epsilon_{\psi_0(h)}\right)$ has thus a convergent sub-sequence $\left(y^{\psi(h)}, \epsilon_{\psi(h)}\right)$. Since $\epsilon_{\psi(h)}$ is a convergent sub-sequence of $\epsilon_{\psi_0(h)}$, $\epsilon_{\psi(h)} \to \epsilon_*$ holds. Because $\psi(h-1) < \psi(h)$ by definition of ψ, the cut related to $y^{\psi(h-1)}$ is a constraint of problem $R_{\psi(h)}$ (added by Algorithm 1 at iteration $h-1$). Thus, $f(x^{\psi(h)}, y^{\psi(h-1)}) - d(x^{\psi(h)}) \geqslant 0$, and

$$\begin{array}{rcl} f(x^{\psi(h)}, y^{\psi(h)}) - d(x^{\psi(h)}) & = & f(x^{\psi(h)}, y^{\psi(h)}) - f(x^{\psi(h)}, y^{\psi(h-1)}) + f(x^{\psi(h)}, y^{\psi(h-1)}) - d(x^{\psi(h)}) \\ & \geqslant & f(x^{\psi(h)}, y^{\psi(h)}) - f(x^{\psi(h)}, y^{\psi(h-1)}). \end{array}$$

Being the negative part function decreasing,

$$\epsilon_{\psi(h)} = \left(f(x^{\psi(h)}, y^{\psi(h)}) - d(x^{\psi(h)})\right)^- \leqslant \left(f(x^{\psi(h)}, y^{\psi(h)}) - f(x^{\psi(h)}, y^{\psi(h-1)})\right)^- .$$

Therefore

$$\epsilon_{\psi(h)} \leqslant \left| f(x^{\psi(h)}, y^{\psi(h)}) - f(x^{\psi(h)}, y^{\psi(h-1)}) \right| . \tag{3.11}$$

From the fact that f is L-Lipschitz continuous, and Eq. (3.11) we deduce that

$$\epsilon_{\psi(h)} \leqslant L \left\| \begin{pmatrix} x^{\psi(h)} \\ y^{\psi(h)} \end{pmatrix} - \begin{pmatrix} x^{\psi(h)} \\ y^{\psi(h-1)} \end{pmatrix} \right\| = L \, \|y^{\psi(h)} - y^{\psi(h-1)}\|. \tag{3.12}$$

As $y^{\psi(h)}$ is convergent, we know that $\|y^{\psi(h)} - y^{\psi(h-1)}\| \to 0$. Being $\epsilon_{\psi(h)}$ nonnegative, we deduce from Eq. (3.12) that $\epsilon_{\psi(h)} \to 0$, and thus, $\epsilon_\star = 0$.

We proved that the sequence ϵ_h is bounded, and that any converging sub-sequence converge towards 0, thus we can conclude that ϵ_h converges towards 0 itself, according to a well-known result in analysis [10]. Based on this first result, we are now going to prove the second point, i.e., the convergence of objective error. We know that

$$\forall h \in \mathbb{N}^\star \quad F(x^h) \in \left[F(x^1), \mathsf{val}(\mathrm{BP})\right], \tag{3.13}$$

therefore the increasing sequence $F(x^h)$ is bounded, and thus, converging. Since x^h is bounded, we can derive a converging sub-sequence $x^{\phi(h)} \to x^\star$ with $\phi : \mathbb{N}^\star \mapsto \mathbb{N}^\star$ being an increasing function. The associated feasibility error is

$$\epsilon_{\phi(h)} = \left(\mathsf{val}(P_{x^{\phi(h)}}) - d(x^{\phi(h)})\right)^- .$$

On the one hand, being $\epsilon_{\phi(h)}$ a sub-sequence of ϵ_h which has been proven to converge towards zero, $\epsilon_{\phi(h)} \to 0$. On the other hand, $\epsilon_{\phi(h)} \to (\mathsf{val}(P_{x^\star}) - d(x^\star))^-$ holds by continuity of $x \mapsto \mathsf{val}(P_x)$ and d. By uniqueness of the limit, $(\mathsf{val}(P_{x^\star}) - d(x^\star))^- = 0$. Therefore, $x^\star \in \mathcal{X}$ is feasible in (BP) and $F(x^\star) \geqslant \mathsf{val}(\text{BP})$. From (3.13) we also know that $F(x^\star) \leqslant \mathsf{val}(\text{BP})$, and thus $F(x^\star) = \mathsf{val}(\text{BP})$. We can conclude that $F(x^h)$ is bounded and admits a unique limit point which is $\mathsf{val}(\text{BP})$. Hence, $\delta_h \to 0$. □

3.4.2 A convergence rate for the CP algorithm

In this section, we give a convergence rate of the CP algorithm 1, under two additional assumptions on the bilevel problem. First of all, let us reformulate the bilevel problem, by moving the function $d(x)$ within the lower-level problem:

$$\left\{ \begin{array}{ll} \min\limits_{x \in \mathcal{X}} & F(x) \\ \text{s.t.} & 0 \leqslant \min\limits_{y \in \mathbb{R}^n} \{ \frac{1}{2} y^\top Q(x) y + q(x)^\top y - d(x) \mid y \in \mathcal{F} \}. \end{array} \right. \tag{BP}$$

We introduce then the matrix $\mathcal{G}(x) = \frac{1}{2}\begin{pmatrix} Q(x) & q(x) \\ q(x)^\top & -2d(x) \end{pmatrix} = \mathcal{Q}(x) - \begin{pmatrix} 0_n & 0 \\ 0 & d(x) \end{pmatrix}$ and we define the set

$$\mathcal{P} = \left\{ M(y) = \begin{pmatrix} yy^\top & y \\ y^\top & 1 \end{pmatrix} : y \in \mathcal{F} \right\} \subset \mathbb{R}^{(n+1)\times(n+1)}.$$

With this notation, we acknowledge that (BP) can be formulated as

$$\left\{ \begin{array}{ll} \min\limits_{x \in \mathcal{X}} & F(x) \\ \text{s.t.} & 0 \leqslant \langle \mathcal{G}(x), Y \rangle, \ \forall\, Y \in \mathcal{P}. \end{array} \right. \tag{SIP}$$

We define as $\mathcal{K} = \mathsf{cone}(\mathcal{P}) \subset \mathbb{R}^{(n+1)\times(n+1)}$ the convex cone generated by $\mathcal{P}$, and $\mathcal{L}(x, Y) = F(x) - \langle \mathcal{G}(x), Y \rangle$ the Lagrangian function defined over $\mathcal{X} \times \mathcal{K}$. We remark that for any $x \in \mathcal{X}$, the following equality holds

$$\sup_{Y \in \mathcal{K}} \mathcal{L}(x, Y) = \left\{ \begin{array}{ll} F(x) & \text{if } 0 \leqslant \langle \mathcal{G}(x), Y \rangle, \ \forall\, Y \in \mathcal{P} \\ +\infty & \text{else.} \end{array} \right.$$

Hence, problem (SIP) can be expressed as the saddle-point problem $\min\limits_{x \in \mathcal{X}} \sup\limits_{Y \in \mathcal{K}} \mathcal{L}(x, Y)$. At this point, we do the following further assumption.

Assumption 7. *The upper-level objective function $F(x)$ is μ-strongly-convex.*

Assumptions 7 is quite strong, but we remark that, if the original objective function is just convex, it is always possible to enforce this assumption by "regularizing" the bilevel problem adding a ℓ_2 penalty to the primal objective function, i.e., minimizing $F(x) + \frac{\mu}{2}\|x\|^2$ instead of $F(x)$.

The Lagrangian function $\mathcal{L}(x, Y)$ is linear (thus continuous and concave) w.r.t. Y for all $x \in \mathcal{X}$ and is continuous and convex w.r.t. x for all $Y \in \mathcal{K}$. The convexity w.r.t. x follows from Assumptions 3 and 4 and from the fact that $Y_{n+1,n+1} \geqslant 0$ for any $Y \in \mathcal{K}$. Since the set $\mathcal{X}$ is convex (Assumption 2) and the set $\mathcal{K}$ is convex too, the Sion's minimax theorem is applicable and the following holds:

$$\min_{x \in \mathcal{X}} \sup_{Y \in \mathcal{K}} \mathcal{L}(x, Y) = \sup_{Y \in \mathcal{K}} \min_{x \in \mathcal{X}} \mathcal{L}(x, Y).$$

Defining the dual function $\theta(Y) = \min_{x \in \mathcal{X}} \mathcal{L}(x, Y)$, we know that

$$\mathsf{val}(\mathsf{SIP}) = \sup_{Y \in \mathcal{K}} \theta(Y). \tag{3.14}$$

Notice that the dual function $\theta(Y)$ is concave, as a minimum of linear functions in Y. As a direct application of [111, Corollary VI.4.4.5], the dual function $\theta(Y)$ is differentiable because of the uniqueness of $\arg\min_{x \in \mathcal{X}} \mathcal{L}(x, Y)$, which is, in turn, a consequence of the strong convexity of $x \mapsto \mathcal{L}(x, Y)$ that follows from Assumption 7. Moreover, the gradient of the dual function is $\nabla\theta(Y) = -\mathcal{G}(x)$, where $x = \arg\min_{x \in \mathcal{X}} \mathcal{L}(x, Y)$. The differentiability of θ implies, in particular, that θ is continuous. We prove now that we can replace the sup operator with the max operator in the formulation (3.14), under the following assumption.

Assumption 8. *There exists $\hat{x} \in \mathcal{X}$, s.t., for all $y \in \mathcal{F}$, $g(\hat{x}, y) = \frac{1}{2} y^\top Q(\hat{x}) y + q(\hat{x})^\top y - d(\hat{x}) > 0$.*

Lemma 2. *Under Assumption 8, the dual problem of (SIP) has an optimal solution Y^*.*

Proof. We denote by $\hat{x} \in \mathcal{X}$ the primal feasible solution s.t. $g(\hat{x}, y) = \frac{1}{2} y^\top Q(\hat{x}) y + q(\hat{x})^\top y - d(\hat{x}) > 0$ for all $y \in \mathcal{F}$. Since the set $\mathcal{F}$ is compact and the function $y \mapsto g(\hat{x}, y)$ is continuous and positive, it exists $c > 0$ s.t. $g(\hat{x}, y) \geqslant c$ for all $y \in \mathcal{F}$. For any $Y \in \mathcal{K}$, we have that $Y = \sum_{k=1}^{p} \lambda_k M(y^k)$, for an integer $p \in \mathbb{N}$, vectors $y^1, \ldots, y^p \in \mathcal{F}$ and nonnegative scalars $\lambda_1, \ldots, \lambda_p \in \mathbb{R}_+$. Since $\langle \mathcal{G}(\hat{x}), M(y) \rangle = \frac{1}{2} y^\top Q(\hat{x}) y + q(\hat{x})^\top y - d(\hat{x})$ for any $y \in \mathcal{F}$, the following holds by linearity:

$$\langle \mathcal{G}(\hat{x}), Y \rangle = \left\langle \mathcal{G}(\hat{x}), \sum_{k=1}^{p} \lambda_k M(y^k) \right\rangle = \sum_{k=1}^{p} \lambda_k \left\langle \mathcal{G}(\hat{x}), M(y^k) \right\rangle \geqslant \sum_{k=1}^{p} \lambda_k c = Y_{n+1,n+1} c.$$

Moreover, by definition of θ:

$$\theta(Y) = \min_{x\in\mathcal{X}} F(x) - \langle \mathcal{G}(x), Y\rangle \leqslant F(\hat{x}) - \langle \mathcal{G}(\hat{x}), Y\rangle \leqslant F(\hat{x}) - Y_{n+1,n+1}c,$$

this for any $Y \in \mathcal{K}$. We take then a maximizing sequence $(Y^h)_{h\in\mathbb{N}}$ of problem (3.14). Defining $V = \mathsf{val}(\mathsf{SIP})$, we know that $\theta(Y^h) \to V$ and hence, it exists $j \in \mathbb{N}$ s.t. for all $h \geqslant j$, $\theta(Y^h) \geqslant V - 1$. This implies that, for all $h \geqslant j$,

$$0 \leqslant Y^h_{n+1,n+1} \leqslant \frac{F(\hat{x}) - V + 1}{c}.$$

Defining $D = \frac{F(\hat{x})-V+1}{c}$, we deduce that $\forall h \geqslant j$, Y^h belongs to $D\,\mathsf{conv}(\mathcal{P})$, which is compact. Thus, the sequence $(Y^h)_{h\in\mathbb{N}}$ admits an accumulation point Y^*, s.t. $\theta(Y^*) = V$ by continuity of θ. $\square$

According to this lemma, the dual version of problem (SIP) thus reads

$$\max_{Y\in\mathcal{K}} \theta(Y). \tag{DSIP}$$

This concave maximization problem on the convex cone $\mathcal{K}$ is the Lagrangian dual of the problem (SIP) i.e., of the bilevel program (BP). Indeed, in this section, we are dualizing the whole bilevel problem (BP), contrary to Section 3.3, where we dualize the lower-level problem only. We are now going to see that the CP algorithm 1 can be interpreted, from a dual perspective, as a cone constrained Fully Corrective Frank-Wolfe (FCFW) algorithm [137] solving the dual problem (DSIP). We prove that during the execution of the CP algorithm 1, the dual variables obtained when solving the relaxation R_h instantiate the iterates of a FCFW algorithm. In the following, the sets $B_h \subset \mathbb{R}^{n+1\times n+1}$ are finite sets, composed of rank-one matrices of the form $M(y)$.

First, the initialization of the CP can be seen, in the dual perspective, as the initialization of a Frank-Wolfe type algorithm, with $B_0 \leftarrow \varnothing$, and $Y^0 = 0$. Second, the generic iteration h is described in Table 3.1. The different steps summarized in Table 3.1 can be explicated as follows:

- *Step 1*: At iteration h, set B_h represents, from a dual perspective, the set of CPs in the primal relaxation R_h. The dual problem of R_h is in fact a restriction of (DSIP) on $\mathsf{cone}(B_h)$, which is a polyhedral subcone of $\mathcal{K}$, since the following holds:

$$\begin{aligned}\max_{Y\in\mathsf{cone}(B_h)} \theta(Y) &= \max_{Y\in\mathsf{cone}(B_h)} \min_{x\in\mathcal{X}} \left(F(x) - \langle \mathcal{G}(x), Y\rangle\right)\\ &= \min_{x\in\mathcal{X}} \max_{Y\in\mathsf{cone}(B_h)} \left(F(x) - \langle \mathcal{G}(x), Y\rangle\right)\\ &= \min_{x\in\mathcal{X}} \{F(x) \text{ s.t. } 0 \leqslant \langle \mathcal{G}(x), Z\rangle,\ \forall Z \in B_h\},\end{aligned}$$

	Primal perspective: CP	Link	Dual perspective: FCFW
Step 1	Solve R_h and store the solution x^h	Duality	Solve the dual problem on $\mathsf{cone}(B_h)$, i.e., $\max_{Y \in \mathsf{cone}(B_h)} \theta(Y)$, store the solution Y^h, the associated x^h and the gradient $\nabla\theta(Y^h) = -\mathcal{G}(x^h)$
Step 2	Solve the lower-level problem P_{x^h} $\min_{y \in \mathcal{F}} \frac{1}{2} y^\top Q(x^h) y + q(x^h)^\top y$ and store the solution y^h	$Z^h = M(y^h)$	Solve the problem $\max_{Z \in \mathcal{P}} \langle \nabla\theta(Y^h), Z \rangle$ and store the solution Z^h
Step 3a	If $d(x^h) \leqslant \frac{1}{2}(y^h)^\top Q(x^h) y^h + q(x^h)^\top y^h$, (x^h, y^h) is the optimal solution of (BP)	Reformulation	If $\langle \nabla\theta(Y^h), Z^h \rangle \leqslant 0$, Y^h is the optimal solution of (DSIP), x^h is the optimal solution of (SIP)
Step 3b	If $d(x^h) > \frac{1}{2}(y^h)^\top Q(x^h) y^h + q(x^h)^\top y^h$, build R_{h+1} as R_h with the additional ineq. $d(x) \leqslant \frac{1}{2}(y^h)^\top Q(x) y^h + q(x)^\top y^h$	Reformulation	If $\langle \nabla\theta(Y^h), Z^h \rangle > 0$, set $B_{h+1} \leftarrow B_h \cup \{Z^h\}$.

Table 3.1: The h-th iteration of the CP (Algorithm 1), and of the FCFW algorithm

which we recognize being the master problem R_h. The absence of duality gap is, also in this case, a direct application of Sion's Theorem. The new dual solution Y^h is obtained solving this restriction of (DSIP) on $\mathsf{cone}(B_h)$, and the primal solution $x^h = \arg\min_{x \in \mathcal{X}} \mathcal{L}(x, Y^h)$ gives the gradient of the dual function in Y^h, i.e., $\nabla\theta(Y^h) = -\mathcal{G}(x^h)$.

- *Step 2*: Finding the *bilevel* constraint that is the most violated by x^h is equivalent to finding the furthest point of $\mathcal{P}$ in the direction $\nabla\theta(Y^h)$. Indeed, the following equality holds:

$$\max_{Z \in \mathcal{P}} \langle \nabla\theta(Y^h), Z \rangle = -\min_{Z \in \mathcal{P}} \langle \mathcal{G}(x^h), Z \rangle \tag{3.15}$$

$$= -\min_{y \in \mathcal{F}} \{ \frac{1}{2} y^\top Q(x^h) y + q(x^h)^\top y - d(x^h) \}, \tag{3.16}$$

  and any optimal solution Z^h in problem (3.15) has the form $Z^h = M(y^h)$, with y^h optimal in problem (3.16).

- *Step 3a*: The CP feasibility test $\frac{1}{2}(y^h)^\top \ Q(x^h) y^h + q(x^h)^\top \ y^h \geqslant d(x^h)$, is equivalent to the dual optimality condition $\langle \nabla\theta(Y^h), Z^h \rangle \leqslant 0$, according to the equality $\nabla\theta(Y^h) = -\mathcal{G}(x^h)$.

- *Step 3b*: Increasing the set of atoms $B_{h+1} \leftarrow B_h \cup \{Z^h\}$ is the dual point of view of adding the corresponding CP (with y^h s.t. $Z^h = M(y^h)$) to R_h, which creates the relaxation R_{h+1}.

The following lemma states a property of the iterates Y^h.

Lemma 3. *For any $h \in \mathbb{N}$, $\langle \nabla\theta(Y^h), Y^h \rangle = 0$.*

Proof. This property follows directly from the first order optimality condition at 1 of the differentiable function $g : \begin{cases} \mathbb{R}_+ \to \mathbb{R} \\ t \mapsto \theta(tY^h) \end{cases}$. Indeed, $g'(1) = \langle \nabla\theta(Y^h), Y^h \rangle = 0$, because (i) 1 is optimal for g since $Y^h \in \arg\max_{Y \in \mathsf{cone}(B_h)} \theta(Y)$, (ii) 1 lies in the interior of the definition domain of g. $\square$

Based on the dual interpretation of the CP algorithm, we are now going to state a convergence rate for this algorithm. We begin with two technical lemmas.

Lemma 4. *It exists $L > 0$ s.t. function θ is L-smooth, i.e., for all $Y, Y' \in \mathcal{K}$,*

$$\|\nabla\theta(Y) - \nabla\theta(Y')\|_2 \leqslant L\|Y - Y'\|_2.$$

Proof. For the purpose of this proof, we introduce the linear operator $\mathcal{Q}^\star$, defined as the adjoint operator of the linear (by Assumption 3) operator $x \mapsto \mathcal{Q}(x)$. With this notation, we have that $\langle \mathcal{Q}(x), Y \rangle = x^\top(\mathcal{Q}^\star Y)$. We also denote by $\|\mathcal{Q}^\star\|_{\mathsf{op}}$ the operator norm of $\mathcal{Q}^\star$. We notice that the image of the bounded set $\mathcal{X}$ by the subdifferential mapping $\partial d(\mathcal{X}) = \bigcup_{x \in \mathcal{X}} \partial d(x)$ is bounded according to Theorem 6.2.2 in [111, Chapter VI]. Hence it exists $D \geqslant 0$ such that

$$\forall x \in \mathcal{X},\ \forall s \in \partial d(x), \quad \|s\|_2 \leqslant D. \tag{3.17}$$

Given $Y, Y' \in \mathcal{K}$, we are now going to prove that $\|\nabla\theta(Y) - \nabla\theta(Y')\|_2 \leqslant L\|Y - Y'\|_2$ for a constant L that is independent from Y and Y'. Being $i_{\mathcal{X}}(x)$ the indicator function of the set $\mathcal{X}$, we introduce the applications $w : x \mapsto \mathcal{L}(x, Y) + i_{\mathcal{X}}(x)$ and $w' : x \mapsto \mathcal{L}(x, Y') + i_{\mathcal{X}}(x)$. According to Assumptions 7, as well as 2, 3, and 4 we remark that application w (resp. w') is μ-strongly convex because it is the sum of the μ-strongly convex function F and the convex function $x \mapsto -\langle \mathcal{G}(x), Y \rangle + i_{\mathcal{X}}(x)$ (resp. $x \mapsto -\langle \mathcal{G}(x), Y' \rangle + i_{\mathcal{X}}(x)$). Being u (resp. u') the unique minimum of function w (resp. w'), the uniqueness following from the strong convexity, the optimality conditions of function w, and w' respectively read

$$0 \in \partial w(u), \tag{3.18}$$
$$0 \in \partial w'(u'). \tag{3.19}$$

We remark that $w'(x) = F(x) + i_{\mathcal{X}}(x) + Y'_{n+1,n+1}d(x) - x^\top(\mathcal{Q}^\star Y')$. The function $x \mapsto F(x) + i_{\mathcal{X}}(x)$ is convex as a sum of convex functions; the function $x \mapsto Y'_{n+1,n+1}d(x)$ is convex since d is convex and $Y'_{n+1,n+1} \geqslant 0$ by definition of cone $\mathcal{K}$; $x \mapsto -x^\top(\mathcal{Q}^\star Y')$ is linear and thus convex. The intersection of the relative interiors of the domains of these convex functions is $\mathsf{ri}(\mathcal{X})$. Since $\mathcal{X}$ is a finite-dimensional convex set, $\mathsf{ri}(\mathcal{X}) \neq \varnothing$ [189, Proposition 1.9]. Hence the subdifferential of the sum is the sum of the subdifferentials [167, Theorem 2.1]. In this respect, the subdifferential of function w' at u' reads

$$\partial w'(u') = \partial(F + i_{\mathcal{X}})(u') - \mathcal{Q}^\star Y' + Y'_{n+1,n+1}\partial d(u').$$

Based on this decomposition, it follows from (3.19) that $\exists\, g_0 \in \partial(F + i_{\mathcal{X}})(u')$, $g_1 \in \partial d(u')$ such that

$$g_0 - \mathcal{Q}^\star Y' + Y'_{n+1,n+1}g_1 = 0. \tag{3.20}$$

Additionally, we have that

$$g_0 - \mathcal{Q}^\star Y + Y_{n+1,n+1}g_1 \in \partial w(u'), \tag{3.21}$$

since $w(x) = F(x) + i_{\mathcal{X}}(x) - x^\top(\mathcal{Q}^\star Y) + Y_{n+1,n+1}d(x)$, and $g_0 \in \partial(F + i_{\mathcal{X}})(u')$, $g_1 \in \partial d(u')$. Combining Eq. (3.20) with Eq. (3.21), we deduce:

$$\mathcal{Q}^\star(Y' - Y) + (Y_{n+1,n+1} - Y'_{n+1,n+1})g_1 \in \partial w(u'). \tag{3.22}$$

Applying Theorem 6.1.2 in [111, Chapter VI], the μ-strong convexity of w gives that, for any $s_1 \in \partial w(u)$ and $s_2 \in \partial w(u')$, $\langle s_2 - s_1, u' - u\rangle \geqslant \mu\|u - u'\|_2^2$. Moreover, due to the Cauchy-Schwartz inequality, $\|s_1 - s_2\|_2\|u - u'\|_2 \geqslant \langle s_2 - s_1, u' - u\rangle$. Therefore, $\|s_2 - s_1\|_2 \geqslant \mu\|u - u'\|_2$ holds for any $s_1 \in \partial w(u)$ and $s_2 \in \partial w(u')$. Since $0 \in \partial w(u)$ according to (3.18), and $\mathcal{Q}^\star(Y' - Y) + (Y_{n+1,n+1} - Y'_{n+1,n+1})g_1 \in \partial w(u')$ according to (3.22), we deduce that

$$\left\|\mathcal{Q}^\star(Y' - Y) + (Y_{n+1,n+1} - Y'_{n+1,n+1})g_1 - 0\right\|_2 \geqslant \mu\|u - u'\|_2.$$

According to the triangle inequality

$$\|\mathcal{Q}^\star(Y' - Y)\|_2 + |Y_{n+1,n+1} - Y'_{n+1,n+1}|\,\|g_1\|_2 \geqslant \mu\|u - u'\|_2,$$

and thus, since $\|Y - Y'\|_2 \geqslant |Y_{n+1,n+1} - Y'_{n+1,n+1}|$,

$$\|\mathcal{Q}^\star\|_{\mathsf{op}}\|Y - Y'\|_2 + \|Y - Y'\|_2\,\|g_1\|_2 \geqslant \mu\|u - u'\|_2.$$

Defining $B = \|\mathcal{Q}^\star\|_{\mathsf{op}} + D$ and using the inequality $\|g_1\|_2 \leqslant D$, which holds according to (3.17), we know that

$$B\|Y - Y'\|_2 \geqslant \mu\|u - u'\|_2.$$

According to Assumption 4, d is Lipschitz continuous and so are q and Q by the linearity Assumption 3. Hence, it exists a constant $K > 0$ such that $x \mapsto \mathcal{G}(x)$ is K-Lipschitz continuous. We deduce that $K\|u - u'\|_2 \geqslant \|\mathcal{G}(u) - \mathcal{G}(u')\|_2$, and, consequently, $\|Y - Y'\|_2 \geqslant \frac{\mu}{BK}\|\mathcal{G}(u) - \mathcal{G}(u')\|_2$. We define the constant $L = \frac{BK}{\mu}$, which is clearly independent from Y, Y', u and u'. Since $\nabla\theta(Y) = -\mathcal{G}(u)$ and $\nabla\theta(Y') = -\mathcal{G}(u')$, we deduce that

$$L\|Y - Y'\|_2 \geqslant \|\nabla\theta(Y) - \nabla\theta(Y')\|_2,$$

which concludes the proof. □

The following lemma is a consequence of the L-smoothness θ.

Lemma 5. *Let L denote the smoothness constant associated with θ. For any $Y, Z \in \mathcal{K}$ and for any $\gamma \geqslant 0$,*

$$\theta(Y + \gamma Z) \geqslant \theta(Y) + \gamma\langle\nabla\theta(Y), Z\rangle - \frac{L\|Z\|^2}{2}\gamma^2.$$

Proof. For any $Y, Z \in \mathcal{K}$ and $\gamma > 0$, it holds by integration that

$$\theta(Y+\gamma Z)-\theta(Y) = \int_{t=0}^{\gamma}\langle\nabla\theta(Y+tZ), Z\rangle dt = \gamma\langle\nabla\theta(Y), Z\rangle+\int_{t=0}^{\gamma}\langle\nabla\theta(Y+tZ)-\nabla\theta(Y), Z\rangle dt. \tag{3.23}$$

Since $\langle\nabla\theta(Y + tZ) - \nabla\theta(Y), Z\rangle \geqslant -|\langle\nabla\theta(Y + tZ) - \nabla\theta(Y), Z\rangle|$, using Cauchy-Schwartz inequality and L-smoothness of θ, we know that

$$\langle\nabla\theta(Y + tZ) - \nabla\theta(Y), Z\rangle \geqslant -\|\nabla\theta(Y + tZ) - \nabla\theta(Y)\|_2\,\|Z\|_2 \geqslant -tL\|Z\|_2^2. \tag{3.24}$$

Combining Eq. (3.23) with Eq. (3.24), we deduce that

$$\theta(Y + \gamma Z) - \theta(Y) \geqslant \gamma\langle\nabla\theta(Y), Z\rangle - \int_{t=0}^{\gamma} tL\|Z\|_2^2 dt,$$

which yields finally that $\theta(Y + \gamma Z) - \theta(Y) \geqslant \gamma\langle\nabla\theta(Y), Z\rangle - \frac{L\|Z\|^2}{2}\gamma^2$. □

We define the constant $T = \max_{Z\in\mathcal{P}} \|Z\|^2$, which is finite by compactness of $\mathcal{F}$, and thus of $\mathcal{P}$. According to Lemma 2, (DSIP) admits an optimal solution Y^*. We remark that the dual optimality gap at h-th iteration is $\delta_h = \theta(Y^*) - \theta(Y^h) \geqslant 0$, where δ_h is the objective error defined in Theorem 2. We define τ as the last element of the optimal dual solution Y^*, i.e., $\tau = Y^*_{n+1,n+1}$. This scalar plays a central role in the convergence rate analysis, conducted in the following theorem.

Theorem 3. *Under Assumptions 1-8: if Algorithm 1 executes the iteration of index $h \in \mathbb{N}$, then*

$$\delta_h \leqslant \frac{2LT\tau^2}{h+2}. \tag{3.25}$$

Otherwise, it exists an index $j \leqslant h$ s.t. Y^j is optimal for (DSIP), and $x^j = \arg\min_{x \in \mathcal{X}} \mathcal{L}(x, Y^j)$ is optimal for (SIP).

Proof. If the algorithm terminates at iteration $j \in \mathbb{N}$, this means that

$$\max_{Z \in \mathcal{P}} \langle \nabla\theta(Y^j), Z \rangle \leqslant 0. \tag{3.26}$$

Defining $x^j = \arg\min_{x \in \mathcal{X}} \mathcal{L}(x, Y^j)$, we have that $\nabla\theta(Y^j) = -\mathcal{G}(x^j)$. Eq. (3.26) is thus equivalent to $\min_{Z \in \mathcal{P}} \langle \mathcal{G}(x^j), Z \rangle \geqslant 0$. This proves that x^j is feasible in (SIP). Moreover $\langle \mathcal{G}(x^j), Y^j \rangle = \langle \nabla\theta(Y^j), Y^j \rangle = 0$, according to Lemma 3, and, therefore, $F(x^j) = \mathcal{L}(x^j, Y^j) = \theta(Y^j)$. Hence x^j and Y^j are feasible solutions in the primal (SIP) and the dual (DSIP) respectively, and have the same value. Therefore, x^j is optimal for (SIP), and Y^j is optimal for (DSIP).

We focus now on the case where Algorithm 1 does not terminate, and prove (3.25) by induction.

Base case: $h = 0$ Since θ is concave, we have that

$$\delta_0 = \theta(Y^*) - \theta(Y^0) \leqslant \langle \nabla\theta(Y^0), Y^* - Y^0 \rangle = \langle \nabla\theta(Y^0), Y^* \rangle,$$

the last equality coming from $Y^0 = 0$. We remark that $\langle \nabla\theta(Y^0), Y^* \rangle = \langle \nabla\theta(Y^0) - \nabla\theta(Y^*), Y^* \rangle$ since $\langle \nabla\theta(Y^*), Y^* \rangle = 0$ by optimality of Y^*. Hence,

$$\delta_0 \leqslant \langle \nabla\theta(Y^0) - \nabla\theta(Y^*), Y^* \rangle \leqslant \|\nabla\theta(Y^0) - \nabla\theta(Y^*)\| \, \|Y^*\|,$$

where the last inequality is the Cauchy-Schwarz inequality. Using the L-Lipschitzness of $\nabla\theta$, we know that $\|\nabla\theta(Y^0) - \nabla\theta(Y^*)\| \leqslant L\|Y^0 - Y^*\| = L\|Y^*\|$. Finally, we deduce that, since $Y^* \in \tau\mathcal{P}$,

$$\delta_0 \leqslant L\|Y^*\|^2 \leqslant LT\tau^2.$$

Induction We suppose that the algorithm runs $h+1$ iterations, and that the property (3.25) is true for h. Using Lemma 5, we can compute a lower bound on the progress made during the iteration of index $h+1$:

$$\theta(Y^{h+1}) \geqslant \theta(Y^h + \gamma Z^h) \geqslant \theta(Y^h) + \gamma\langle \nabla\theta(Y^h), Z^h \rangle - \frac{L\|Z^h\|^2}{2}\gamma^2,$$

for any $\gamma \geqslant 0$. Multiplying by -1, and adding $\theta(Y^*)$ to both left and right hand sides of the above inequality, and using $\|Z^h\|^2 \leqslant T$, we have that

$$\delta_{h+1} \leqslant \delta_h - \gamma\langle\nabla\theta(Y^h), Z^h\rangle + \frac{LT}{2}\gamma^2, \tag{3.27}$$

for any $\gamma \geqslant 0$. We remark that the value T is independent from h. By concavity of θ, it also holds that $\delta_h = \theta(Y^*) - \theta(Y^h) \leqslant \langle\nabla\theta(Y^h), Y^* - Y^h\rangle$. We notice that $\langle\nabla\theta(Y^h), Y^h\rangle = 0$, according to Lemma 3. Thus, $\delta_h \leqslant \langle\nabla\theta(Y^h), Y^*\rangle$. As $Y^*_{n+1,n+1} = \tau$, we know that $Y^* \in \tau\mathsf{conv}(\mathcal{P})$, and, therefore,

$$\delta_h \leqslant \max_{Z\in\tau\mathcal{K}}\langle\nabla\theta(Y^h), Z\rangle = \max_{Z\in\tau\mathcal{P}}\langle\nabla\theta(Y^h), Z\rangle = \tau\langle\nabla\theta(Y^h), Z^h\rangle, \tag{3.28}$$

the last equality following from the definition of Z^h. Combining Eq. (3.27) and (3.28), it holds that

$$\delta_{h+1} \leqslant \delta_h - \gamma\tau^{-1}\delta_h + \frac{LT}{2}\gamma^2,$$

for every $\gamma \geqslant 0$. Factorizing and doing a change of variable $\eta = \gamma\tau^{-1}$, for any $\eta \geqslant 0$:

$$\delta_{h+1} \leqslant (1-\eta)\delta_h + \frac{LT\tau^2}{2}\eta^2. \tag{3.29}$$

We have derived a lower bound on optimality gap at iteration h. We apply then (3.29) with $\eta = \frac{2}{h+2}$:

$$\delta_{h+1} \leqslant (1 - \frac{2}{h+2})\delta_h + \frac{LT\tau^2}{2}\frac{4}{(h+2)^2} \leqslant \frac{h}{h+2}\frac{2LT\tau^2}{h+2} + \frac{LT\tau^2}{2}\frac{4}{(h+2)^2},$$

the second inequality coming from the application of (3.25) for h, which is true by induction hypothesis. Finally, we deduce that

$$\delta_{h+1} \leqslant \frac{2LT\tau^2}{h+2}(\frac{h}{h+2} + \frac{1}{h+2}) \leqslant \frac{2LT\tau^2}{h+2}\frac{h+1}{h+2} \leqslant \frac{2LT\tau^2}{h+2}\frac{h+2}{h+3} = \frac{2LT\tau^2}{h+3},$$

the third inequality coming from the observation that $\frac{h+1}{h+2} \leqslant \frac{h+2}{h+3}$. Hence, the property (3.25) is true for $h+1$ as well. This concludes the proof by induction. □

We remark that the convergence rate defined in (3.25) is directly related to the iteration index h, which is something different w.r.t. what is usually proved for existing CP algorithms solving SIP problems [38, 129, 162], where the rate of convergence is not directly controlled by h.

3.5 Applications

In this section, we present two problems that can be modeled as (BP). For each of these, we present both the bilevel formulation, and the corresponding single-level formulation (BPR).

3.5.1 Constrained quadratic regression

We consider a quadratic statistical model with Gaussian noise linking a vector $w \in \mathbb{R}^n$ of explanatory variables, i.e., the features vector, and an output $z \in \mathbb{R}$ as follows:

$$z = \frac{1}{2} w^\top \bar{Q} w + \bar{q}^\top w + \bar{c} + \epsilon,$$

where $\bar{Q} \in \mathbb{R}^{n\times n}$ s.t. $\bar{Q} = \bar{Q}^\top$, $\bar{q} \in \mathbb{R}^n$, $\bar{c} \in \mathbb{R}$ and $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Let us suppose that the parameters of this model are unknown, but we are given a dataset $(w_i, z_i)_{1\leqslant i\leqslant P} \in (\mathbb{R}^n \times \mathbb{R})^P$. The problem of finding the maximum likelihood estimator for $\bar{Q} \in \mathbb{R}^{n\times n}$, $\bar{q} \in \mathbb{R}^n$, $\bar{c} \in \mathbb{R}$ just consists in computing the triplet $(Q, q, c) \in \mathbb{R}^{n\times n} \times \mathbb{R}^n \times \mathbb{R}$ that minimizes the least-squares error $\sum_{i=1}^{P} (z_i - \frac{1}{2} w_i^\top Q w_i - q^\top w_i - c)^2$. We consider that (i) the features vector belongs to a given polytope $\mathcal{F} \subset \mathbb{R}^n$, (ii) the noiseless value $\frac{1}{2} y^\top \bar{Q} y + \bar{q}^\top y + \bar{c}$ is nonnegative for any $y \in \mathcal{F}$. Hence, this inverse problem is a "constrained quadratic regression problem" that may be written as:

$$\begin{cases} \min\limits_{Q,q,c} & \sum\limits_{i=1}^{P} (z_i - \frac{1}{2} w_i^\top Q w_i - q^\top w_i - c)^2 \\ \text{s.t.} & Q = Q^\top \\ & \frac{1}{2} y^\top Q y + q^\top y + c \geqslant 0 \quad \forall y \in \mathcal{F} \\ & Q \in \mathbb{R}^{n\times n},\ q \in \mathbb{R}^n,\ c \in \mathbb{R}. \end{cases} \tag{3.30}$$

Formulation (3.30) is a SIP problem, having uncountably many constraints, which are parametrized by $y \in \mathcal{F}$. We can reformulate this SIP problem as a bilevel problem just replacing the SIP constraint $\frac{1}{2} y^\top Q y + q^\top y + c \geqslant 0\ \forall y \in \mathcal{F}$ with the bilevel constraint $\min\limits_{y\in\mathcal{F}} \{\frac{1}{2} y^\top Q y + q^\top y\} \geqslant -c$.

This model fits in the general setting of formulation (BP), where the matrix Q is itself the upper-level variable of dimensions $n \times n$. As in Section 3.3, we assume that $\mathcal{F} = \{y \in \mathbb{R}^n : a_j^\top y \leqslant b_j, \forall j = 1, \dots, r\}$ is included in the centered ℓ_2-ball with radius $\rho > 0$, and we use the notation $\mathcal{A}_j = \begin{pmatrix} 0_n & \frac{a_j}{2} \\ \frac{a_j^\top}{2} & 0 \end{pmatrix}$ for all $j \in \{1, \dots, r\}$. Then, the (BPR) formulation corresponding to (3.30) reads:

$$\begin{cases} \min\limits_{Q,q,c,\lambda,\alpha,\beta} & \sum\limits_{i=1}^{P} (z_i - \frac{1}{2} w_i^\top Q w_i - q^\top w_i - c)^2 \\ \text{s.t.} & Q = Q^\top \\ & -\lambda^\top b - \alpha(1+\rho^2) - \beta \geqslant -c \\ & \frac{1}{2} \begin{pmatrix} Q + 2\alpha I_n & q \\ q^\top & 2(\beta+\alpha) \end{pmatrix} + \sum\limits_{j=1}^{r} \lambda_j \mathcal{A}_j \succeq 0 \\ & Q \in \mathbb{R}^{n\times n},\ q \in \mathbb{R}^n,\ c \in \mathbb{R} \\ & \lambda \in \mathbb{R}_+^r,\ \alpha \in \mathbb{R}_+, \beta \in \mathbb{R}. \end{cases} \tag{3.31}$$

Formulation (3.31) is feasible, because the all-zero solution satisfies every constraint. In general, (3.31) is a restriction of (3.30) since Q may not necessarily be PSD. In order to benchmark our approaches, we can solve the following relaxation of (3.30) — it is be a reformulation if Q is PSD — obtained by replacing the lower-level problem by its KKT conditions:

$$\left\{\begin{array}{rl} \min\limits_{Q,q,c,y,\gamma} & \sum\limits_{i=1}^{P}(z_i - \frac{1}{2}w_i^\top Q w_i - q^\top w_i - c)^2 \\ \text{s.t.} & Q = Q^\top \\ & \frac{1}{2}y^\top Q y + q^\top y \geqslant -c \\ & Ay \leqslant b \\ & Qy + q + A^\top \gamma = 0 \\ & \gamma^\top (Ay - b) = 0 \\ & Q \in \mathbb{R}^{n\times n},\ q \in \mathbb{R}^n,\ c \in \mathbb{R},\ y \in \mathbb{R}^n,\ \gamma \in \mathbb{R}^r_+, \end{array}\right. \tag{3.32}$$

where γ is the KKT multiplier vector associated to the lower-level constraints $Ay \leqslant b$. This relaxation/reformulation of problem (3.30) is a nonconvex polynomial optimization problem involving multivariate polynomials of degree up to three.

3.5.2 Zero-sum game with cubic payoff

In this section, we are interested in solving a two-player zero-sum game that is related to an undirected graph $\mathcal{G} = (V, E)$. A two-player game, in general, is zero-sum if one player's gains are the other player's losses. Zero-sum games have applications in sports, finance, politics, economics, and so on. Here, we consider a zero-sum game involving a typical optimization structure: a graph. We assume that player 1 benefits from a strategical advantage on player 2, which will be explained more precisely later. We let n denote the cardinality of V. Each player positions a resource on each node $i \in V$. After normalization, we can consider that the action set of both players is $\Delta_n = \{x \in \mathbb{R}^n_+ \ :\ \sum\limits_{i=1}^{n} x_i = 1\}$. A two-player zero-sum game is a two-player game s.t., for every strategy $x \in \Delta_n$ of player 1, and for every strategy $y \in \Delta_n$ of player 2, the payoffs of the two players sum to zero. If we define $P_i(x, y)$ the payoff of player i related to the strategy pair (x, y), we thus have that $P_1(x, y) = -P_2(x, y)$. Since the payoffs sum to zero, we can write the zero-sum game by specifying only one *game payoff*. Player 1 wishes to minimize it, and player 2 wishes to maximize it. The game payoff $P(x, y)$ related to the pair of strategies $(x, y) \in \Delta_n \times \Delta_n$ is the sum of:

- the opposite of a term describing the "proximity" between x and y in the graph, $x^\top M y$, where $M \in \mathbb{R}^{n\times n}$ is the matrix defined as $M_{ij} = 1$ if $i = j$ or $\{i, j\} \in E$, and $M_{ij} = 0$ otherwise,

- the quadratic costs that player 1 has to pay to deploy his resources on the graph: $c_1(x) = \frac{1}{2}x^\top Q_1 x + q_1^\top x$,
- the opposite of the quadratic costs that player 2 has to pay to deploy her resources on the graph, and that is influenced by player 1 strategy: $c_2(x,y) = \frac{1}{2}y^\top Q_2(x)y + q_2^\top y$. In this sense, player 1 has a strategic advantage over player 2.

Hence, this zero-sum game can then be written as $\min_{x\in\Delta_n}\max_{y\in\Delta_n} -x^\top My + c_1(x) - c_2(x,y)$. Loosely speaking, player 1 trades off his costs for placing his resource where player 2's one is (i.e., maximizing the proximity) and for augmenting player 2's costs. In the meantime, player 2 tries to *avoid* player 1, while minimizing her own costs. From player 1's perspective, this problem can be cast as the following bilevel formulation:

$$\begin{cases} \min\limits_{x,v} & \frac{1}{2}x^\top Q_1 x + q_1^\top x + v \\ \text{s.t.} & -v \leqslant \min\limits_{y\in\Delta_n} \frac{1}{2}y^\top Q_2(x)y + (q_2 + M^\top x)^\top y \\ & x \in \Delta_n,\ v \in \mathbb{R}. \end{cases} \tag{3.33}$$

This latter formulation clearly fits in the general setting of formulation (BP). Hence, we apply the methodology of Section 3.3 with $r = n+2$, and

- $a_1 = \mathbf{1}$ and $b_1 = 1$,
- $a_2 = -\mathbf{1}$ and $b_2 = 0$,
- $\forall j \in \{1,\dots,n\} \quad a_{j+2} = -e_j$ and $b_j = 0$,
- $\rho = 1$,

where e_j is the j-th vector of the standard basis in $\mathbb{R}^n$ and $\mathbf{1}$ the all-ones n-dimensional vector. The dual variable is $\lambda \in \mathbb{R}_+^{n+2}$. In this application, the single-level formulation (BPR) reads

$$\begin{cases} \min\limits_{x,v,\lambda,\alpha,\beta} & v + \frac{1}{2}x^\top Q_1 x + q_1^\top x \\ \text{s.t.} & -v \leqslant -\lambda_1 + \lambda_2 - 2\alpha - \beta \\ & \frac{1}{2}\begin{pmatrix} Q_2(x) + 2\alpha I_n & W(x,\lambda) \\ W(x,\lambda)^\top & 2\beta + 2\alpha \end{pmatrix} \succeq 0 \\ & x \in \Delta_n,\ v \in \mathbb{R} \\ & \lambda \in \mathbb{R}_+^{n+2}, \alpha \in \mathbb{R}_+, \beta \in \mathbb{R}, \end{cases} \tag{3.34}$$

where $W(x,\lambda) = q_2 + M^\top x - \sum\limits_{j=1}^{n} \lambda_{j+2} e_j + (\lambda_1 - \lambda_2)\mathbf{1}$. If $Q_2(x) \succeq 0$ is PSD for any $x \in \Delta_n$, formulation (3.34) is a reformulation of (3.33). Otherwise, it is just a restriction of (3.33). In any case, such formulation is feasible, because for given vectors $x \in \Delta_n$, $\lambda \in \mathbb{R}_+^{n+2}$ and scalar $\beta \in \mathbb{R}$, taking arbitrary large scalars α and v, the two constraints are satisfied.

As for the first application, we benchmark our two approaches with the KKT-based relaxation/reformulation (depending on the convexity of the lower-level problem). Given the KKT multipliers γ_1 and γ_2 associated respectively to the lower-level constraint $\sum_{i=1}^{n} y_i = 1$, and the nonnegativity constraint $y \geqslant 0$, the single-level formulation obtained by replacing the lower level of (3.33) by its KKT conditions, is

$$\left\{ \begin{array}{rl} \min\limits_{x,v,y,\gamma_1,\gamma_2} & v + \frac{1}{2}x^\top Q_1 x + q_1^\top x \\ \text{s.t.} & -v \leqslant \frac{1}{2}y^\top Q_2(x) y + (q_2 + M^\top x)^\top y \\ & Q_2(x)y + q_2 + M^\top x + \gamma_1 \mathbf{1} - I_n \gamma_2 = 0 \\ & -\gamma_2^\top (I_n y) = 0 \\ & x \in \Delta_n,\ y \in \Delta_n,\ v \in \mathbb{R},\ \gamma_1 \in \mathbb{R},\ \gamma_2 \in \mathbb{R}_+^n. \end{array} \right. \tag{3.35}$$

The KKT multiplier γ_1 is associated to an equality constraint, hence it can be either nonnegative or negative, and we have no complementarity constraint involving it in formulation (3.35). This relaxation/reformulation of problem (3.33), as well as (3.35), is a nonconvex polynomial optimization problem involving multivariate polynomials of degree up to three.

3.6 Numerical results

In this section we present the numerical results obtained by testing several instances of the two applications presented in Section 3.5, available online at the public repository https://github.com/aoustry/Bilevel-programs-with-QP-as-LL.

For the constrained quadratic regression (Section 3.5.1), we solved twenty randomly generated instances. Each of these instances was generated by choosing the statistical parameters $\bar{Q}, \bar{q}, \bar{c}$ at random, drawing $P = 4000$ random features vectors $w_i \in \mathbb{R}^n$, and then computing the associated outputs $z_i \in \mathbb{R}$ with a centered Gaussian noise. Ten instances — named *PSD_inst#* in Table 3.2 — were produced with $\bar{Q}$ PSD and ten instances — named *notPSD_inst#* in Table 3.2 — with an indefinite $\bar{Q}$.

For the zero-sum game with cubic payoff application (Section 3.5.2), we tested twenty-two instances where the matrix M is taken from the DIMACS graph coloring challenge[1]. We randomly generated Q_1 in a way such that it is PSD, as well as the coefficients of the linear mapping $x \mapsto Q_2(x)$ such that $Q_2(x)$ is

[1] https://mat.tepper.cmu.edu/COLOR/instances.html

PSD for all feasible x in the instances named *#_PSD* in Table 3.3. Regarding the instances named *#_notPSD* in Table 3.3, no particular precaution was taken to enforce that $Q_2(x)$ is PSD. Hence, the sign of the eigenvalues of $Q_2(x)$ depends on x. The code that generated all the instances is available online.

We implemented the single-level formulations based on the *dual approach* using the Python programming language [191] and solve them with the conic optimization solver MOSEK [152]. The bilevel formulations were solved using the CP algorithm (Algorithm 1 presented in Section 3.4) and implemented using the AMPL modeling language [95]. Both the master problem R_h and the lower level problem P_{x^h} were solved using the global optimization solver GUROBI [108]. The tolerance for the feasibility error $\epsilon_h = (d(x^h) - \mathsf{val}(P_{x^h}))^+$ is set to 10^{-6}. With AMPL, we also implemented the traditional relaxation/reformulation approach based on the KKT conditions of the lower-level problem. We solved the KKT-based formulations using the global optimization solver COUENNE [32], chosen after some preliminary computational experiments. These formulations are particularly hard to solve for COUENNE, mainly because of the complementarity constraints. Indeed, for all the tested instances, COUENNE does not terminate within the time limit, and we just display, in italic font, the LB given by the optimal value of the best relaxation of the KKT formulation found by COUENNE within the time limit. All the solvers were run with their default settings. The tests were performed on a computer with 24 2.53GHz Intel(R) Xeon(R) CPUs and with 49.4 GB of RAM. For all the approaches we set a time limit (t.l.) of 18000 seconds (5 hours).

The results for Application 1 and Application 2 are reported in Table 3.2 and Table 3.3 respectively. The headings are the following: "n" is the dimension of the lower-level variable y (or, equivalently, for Application 1 of the matrix Q, for Application 2 of the upper-level variable x); for the single-level formulation approach "obj" is the optimal value found by MOSEK (i.e., either the bilevel optimal value, or an upper bound of it); for the KKT approach, "LB", reported in italics, is the best LB of the KKT formulation value found by the solver COUENNE within the time limit, which is a lower bound for the bilevel optimal value too; for the CP approach "obj/(LB, UB)" is, respectively, either the optimal value of the bilevel formulation, or a pair of values corresponding to: the best lower bound (LB) and the best feasible solution, i.e., upper bound (UB), found by the algorithm within the time limit; "time(s)" is the computing time in seconds; "it" is the number of CP iterations, i.e., the number of times R_h and (P_{x^h}) are solved; "% time (P_{x^h})" is the percentage of the total computing time, i.e., time(s), used to solve (P_{x^h}). In Table 3.2, the "Avg LSE", which is the average least-squares error of the regression, is reported as well. In Table 3.2 and Table 3.3, the best objective values and minimum required times are reported in bold for each instance.

As expected, the *dual approach* leads to a single-level formulation which is a restriction for most of the bilevel problems with a nonconvex lower level, but

Instances		Single-level formulation			KKT approach	CP approach				
Name	n	obj	Avg LSE	time(s)	LB	obj/(LB, UB)	Avg LSE	time(s)	it	% time (P_{x^h})
PSD_inst1	5	358.64	0.08966	**0.19**	*355.78*	358.64	0.08966	1.21	6	3.9
PSD_inst2	5	365.60	0.09140	**0.26**	*363.85*	365.60	0.09140	0.63	3	4.1
PSD_inst3	5	363.43	0.09086	**0.07**	*359.16*	363.43	0.09086	2.62	8	18.0
PSD_inst4	5	353.90	0.08847	**0.07**	*353.19*	353.90	0.08847	1.93	5	32.2
PSD_inst5	10	391.21	0.09780	**0.37**	*359.48*	391.21	0.09780	23.5	17	0.7
PSD_inst6	10	397.59	0.09940	**0.41**	*353.55*	397.59	0.09940	24.2	17	0.7
PSD_inst7	13	440.84	0.11021	**0.36**	*358.19*	440.84	0.11021	64.3	19	0.3
PSD_inst8	13	382.22	0.09555	**0.34**	*345.52*	*381.81, 383.34*	0.09545	t.l.	5	99.9
PSD_inst9	15	572.77	0.14319	**0.92**	*351.95*	*557.71, 1362.6*	0.13943	t.l.	4	100.0
PSD_inst10	15	528.93	0.13223	**1.37**	*346.43*	*526.22, 544.90*	0.13156	t.l.	8	100.0
notPSD_inst1	5	493.19	0.12330	**0.14**	*345.12*	**358.47**	0.08962	0.38	2	5.8
notPSD_inst2	5	425.14	0.10628	**0.15**	*370.89*	**378.28**	0.09457	0.39	2	5.7
notPSD_inst3	5	345.81	0.08645	**0.06**	*345.81*	345.81	0.08645	0.33	1	4.0
notPSD_inst4	5	353.25	0.08831	**0.07**	*353.25*	353.25	0.08831	0.19	1	3.6
notPSD_inst5	10	743.81	0.18595	**0.55**	*360.42*	**503.88**	0.12597	28.3	19	12.9
notPSD_inst6	10	637.62	0.15940	**0.28**	*357.48*	**482.96**	0.12074	412	41	86.6
notPSD_inst7	13	903.44	0.22586	**0.35**	*351.31*	**647.08**	0.16177	657	57	69.7
notPSD_inst8	13	932.21	0.23305	**0.30**	*358.28*	**588.19**	0.14705	3825	77	92.9
notPSD_inst9	15	1592.60	0.39815	**0.99**	*345.44*	**1126.44**	0.28161	15002	99	95.5
notPSD_inst10	15	897.89	0.22447	**0.83**	*350.60*	**580.60**	0.14515	2537	56	87.0

Table 3.2: Numerical results of the first application

Instances		Single-level formulation		KKT approach	CP approach			
Name	n	obj	time(s)	LB	obj/(LB, UB)	time(s)	it	% time (P_{x^h})
jean_PSD	80	-0.0760	18.4	*-4.5808*	-0.0760	**4.68**	186	38.5
myciel4_PSD	23	-0.3643	**0.06**	*-1.9429*	-0.3643	14.3	422	26.8
myciel5_PSD	47	-0.3164	**1.45**	*-4.0081*	-0.3164	85.4	752	9.2
myciel6_PSD	95	-0.2841	**41.4**	*-9.1222*	-0.2841	2781	2323	1.0
myciel7_PSD	191	-0.2608	**4359**	*-14.9495*	*-0.2608, -0.2608*	t.l.	3565	0.4
queen5_5_PSD	25	-0.5536	**0.10**	*-5.6076*	-0.5536	4.16	161	44.3
queen6_6_PSD	36	-0.4619	**0.38**	*-5.6353*	-0.4619	34.4	512	18.3
queen7_7_PSD	49	-0.4054	**1.47**	*-7.8210*	-0.4054	155	969	7.8
queen8_8_PSD	64	-0.3614	**4.22**	*-12.7220*	-0.3614	742	1651	3.1
queen8_12_PSD	96	-0.3000	**34.8**	*-16.0606*	*-0.3000, -0.3000*	t.l.	4082	0.4
queen9_9_PSD	81	-0.3247	**14.4**	*-14.5807*	-0.3247	3544	2578	0.8
jean_notPSD	80	3.2708	**17.4**	*-8.5541*	**2.3979**	37.6	6	99.7
myciel4_notPSD	23	0.8668	**0.07**	*-2.5166*	**0.5198**	466	44	99.9
myciel5_notPSD	47	1.9571	**1.27**	*-7.4343*	**1.2779**	315	32	99.8
myciel6_notPSD	95	3.9171	**39.2**	*-13.9108*	**2.9378**	2735	38	100
myciel7_notPSD	191	7.8030	**3419**	$-\infty$	*6.2486, 6.2486*	t.l.	19	100
queen5_5_notPSD	25	0.8112	**0.08**	*-4.7699*	**0.3800**	326	53	99.8
queen6_6_notPSD	36	1.3876	**0.37**	*-9.7370*	**0.8511**	15872	71	100.0
queen7_7_notPSD	49	1.9740	**1.56**	*-12.4690*	**1.3510**	852	42	99.9
queen8_8_notPSD	64	2.6032	**5.79**	*-15.0751*	**1.8123**	10410	42	100
queen8_12_notPSD	96	3.8131	**41.0**	*-31.4660*	**2.8102**	7035	30	100
queen9_9_notPSD	81	3.2449	**17.3**	*-17.4348*	*2.2975, 2.2996*	t.l.	23	100

Table 3.3: Numerical results of the second application

for the instances *notPSD_inst3* and *notPSD_inst4* of Table 3.2, where the bilevel global optimal solution is attained using both the two approaches, despite the matrix Q is indefinite. It is clear that, in terms of computational time, the *dual approach* is more efficient than the CP approach, not only when MOSEK deals with a restriction of the original bilevel formulation but also when a reformulation is solved. This is the main reason why the *dual approach* is promising, even if a restriction of the original bilevel program is solved. In fact, it let us compute either the bilevel optimal solution or an upper bound of such solution within a small CPU time. As concerns the computation of lower bounds, we see that the CP algorithm provides much tighter lower-bounds than the best lower bound of the KKT relaxation computed by COUENNE within the time limit. Indeed, this formulation is particularly hard to solve mainly because of the complementarity constraints. To understand the causes of the long computational time required by the CP algorithm, we can look at the last column of Table 3.2 and 3.3. For the first application, the time required to perform step 4 of the CP algorithm (i.e., to solve P_{x^h}) is longer than the time required to perform step 3 (i.e., to solve R_h) only for the bigger instances ($n \geqslant 13$ for instances with a convex lower level and $n \geqslant 10$ for instances with a nonconvex lower level). In fact, when n grows, more time is needed to solve a possibly nonconvex QP having Q and q as coefficients, rather than a convex QP having Q and q as variables. When n is small, it is different: even if the inner problem is quadratic nonconvex, it has a small size so it is not harder to solve than the master problem. For the second application, the time required to solve the lower-level problem is longer than the time required to solve the outer relaxation only for the instances having a nonconvex lower level, i.e., the second half of the Table 3.3 rows. In fact, problem R_h has a convex quadratic objective function, since the matrix Q_1 is always PSD, while the inner problem has a convex quadratic objective function only when the matrix $Q_2(x^h)$ is PSD. When $Q_2(x^h)$ is not PSD, problem P_{x^h} is possibly nonconvex and it becomes harder to solve than the master problem.

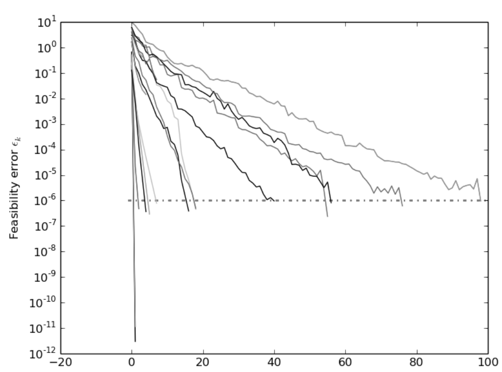

Figure 3.1: Constrained quadratic regression

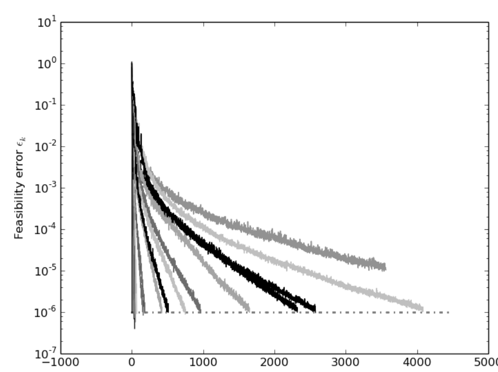

Figure 3.2: Zero-sum game with cubic payoff

Figures 3.1 and 3.2 are aggregated plots showing, for all the tested instances, the trend of the feasibility error ϵ_h over the iterations of the CP algorithm indexed by k. As already said, we set a tolerance of 10^{-6}: for most of the instances, the algorithm stops when ϵ_h reaches or is less than such value. For the instances where the algorithm reaches the time limit, the curve ends at a value of ϵ_h greater than 10^{-6}. For all the instances, anyhow, we can see that the sequence of ϵ_h converges towards 0, as proved in Theorem 2.

3.7 Conclusion

In this chapter we focused on a class of bilevel programs having a possibly nonconvex quadratic problem at the lower level. These bilevel programs are, in fact, linear semi-infinite programming problems with an infinite number of quadratically parameterized constraints. From the point of view of Robust Optimization, it is about handling constraints with quadratic perturbations and a polytopic uncertainty set. We proposed two independent approaches to deal with such bilevel problems. First, a convex single-level formulation obtained via the *dual approach* provides a feasible solution, which is optimal in the case where the quadratic lower-level problem is convex. Second, a cutting plane algorithm enables one to solve directly the bilevel formulation with a guaranteed convergence rate, at the price of solving possibly nonconvex quadratic inner problems. At each iteration, such algorithm provides a lower bound on the value of the bilevel program, which allows one to bound the optimality gap of the feasible solution obtained with the dual approach. Our computational experiments on small and medium-scale instances showed the superiority, in terms of solution time, of the dual approach for the instances with a convex lower level. As concerns the cases with a nonconvex lower level, the two approaches are complementary: the dual approach was faster but provides “only” a feasible solution, the cutting plane approach was slower, but solved the bilevel problem to optimality with good accuracy.

The formulation and the solution of the Conflict Resolution Problem, which is the focus of Chapter 4, will leverage on the results of this chapter.

Part II

Applications

Chapter 4

Aircraft conflict resolution

In this chapter we present different formulations of the aircraft conflict resolution problem (CRP), based on two of the available conflict resolution maneuvers: speed and heading angle changes. As regards the speed regulation based CRP, we present a natural SIP formulation, a polynomial programming formulation, a bilevel formulation and three single-level formulations. For the CRP via heading angle changes, we restrict our analysis to the 2D scenario, where aircraft fly at the same altitude. For this problem, we present a bilevel formulation and its single-level reformulations. We will see that the bilevel formulations, which are much easier to read with respect to other existing formulations, fit in the general framework of (BP) presented in Chapter 3. Therefore we propose a tailored CP algorithm for CRP based on Algorithm 1. The approaches are compared using some benchmarking instances, and a benchmark generator is presented.

The results described in this chapter have been published in [3, 1, 2]. The benchmark generator is detailed in [5], which is currently under review.

4.1 Introduction

Many strategies can be used to detect and solve potential conflicts among aircraft sharing the same portion of airspace. In this chapter we focus on two of them: speed and heading angle modifications. While Heading Angle Changes (HAC) are often used in practice to prevent collisions, speed changes are almost never performed in practice because of the tight speed modification restrictions imposed by air travel regulations. There are several reasons for the strict bounds, which include aircraft dynamics, passengers' comfort and the real-time nature of the decision process needed to make this maneuver efficient. In 2004, however, the concept of *Subliminal Control* was introduced in the context of the European project ERASMUS [66]. Subliminal speed control consists in allowing minor speed adjustments that have to be small enough to remain imperceptible to controllers, thus reducing their workloads.

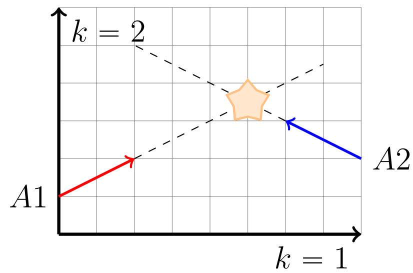

Figure 4.1: Two conflicting aircraft in 2 dimensions

When conflict resolution is applied to aircraft flying at the same altitude level, the optimization takes place in the plane (see Figure 4.1).

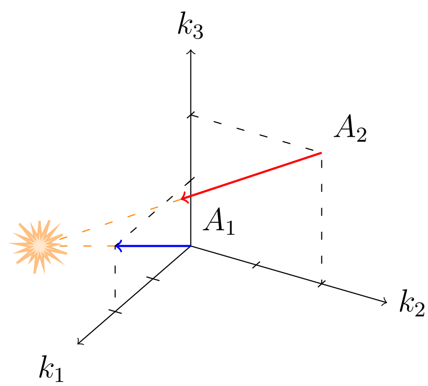

Figure 4.2: Two conflicting aircraft in 3 dimensions

When we consider aircraft moving in a three-dimensional space (see Figure 4.2 as an example), the need for subliminal speed changes becomes less relevant: Speed Regulation (SR) is not realistically performed while changing altitude, but only when aircraft are flying within a fixed altitude layer. This does not apply to Unmanned Aerial Vehicles, however, which have different dynamics, and the development of urban air mobility could benefit from advances in aircraft conflict resolution optimization.

We review the relevant literature in Section 4.2. In Section 4.3 we introduce the SR based CRP formulations: a natural formulation, a polynomial programming formulation, and a bilevel formulation, with three single-level reformulations. In Section 4.4 we present a bilevel formulation of the CRP via HAC in two dimensions (aircraft fly at the same altitude level), and its reformulations. A tailored CP solution algorithm for CRP via both SR and HAC is presented in Section 4.5, based on Algorithm 1 (Chapter 3). In Section 4.6 we discuss computational results. In Section 4.7, we introduce the aircraft deconfliction benchmark generator presented in [5] that allows to generate benchmarks of different complexity, which can be used to test the different approaches. Section 4.8 concludes the chapter.

4.2 Literature review

There exists a wide range of approaches for modeling and solving the CRP. Most of the works consider either only speed regulation (see, e.g., [158, 49, 171, 165, 51] and [2]) or heading angles changes (see, e.g., [158, 161, 19] and [2]). A combination of both maneuvers is considered too (see, e.g., [98, 20, 155, 154]), involving alternative sets of decision variables.

As regards SR based works, in [165] and [166], scalar speed regulation is converted into travel time control in order to minimize the total cost of the conflicts, defined as a function of the time a pair of aircraft spends travelling below the separation threshold in the neighborhood of an intersection point. The variables of the proposed MILP problem are thus the time instants at which aircraft cross each intersection point of the trajectories in the considered time horizon. An equity-oriented deconfliction MILP model, based on these same variables, is introduced in [166]. It proposes an innovative aircraft collision avoidance model promoting the fairness of the solutions (airlines are equally affected by the trajectory adjustments). To this goal, three optimization stages, which are formulated as MILP programs and solved using Cplex, are combined aiming at: maximizing the number of solved conflicts; solving conflicts in the most equitable way possible; reducing the delay induced by the trajectory adjustments. A different kind of approach is proposed in [49, 171], where Mixed-Integer Nonlinear Programming (MINLP) formulations for the CRP via SR in the plane are considered. The formulation proposed in [49] is solved using the general-purpose solver for nonconvex MINLP problems Couenne [32]. In order to deal with the computational difficulties of globally solving the proposed MINLP formulation, a heuristic procedure is introduced too, where the problem is decomposed into problems involving only a small subset of aircraft. All the local solutions obtained by solving each of these problems are combined, returning a feasible solution of the original problem. In [171], another heuristic is proposed to solve the same problem. It builds two sequences of points: one consisting of points that are feasible w.r.t. nonlinear constraints, and the other consisting of points satisfying the integrality conditions. The algorithm iterates until the two sequences converge to a feasible solution of the MINLP formulation.

SR fails to solve frontal conflicts; moreover, it may not be sufficient to ensure safety if speed bounds are tight. Consequently, it is usually combined with other maneuvers, such as HAC. In [50], SR and HAC are applied sequentially. The

MINLP model proposed in [49] is the starting point to present another MINLP formulation where aircraft speeds are fixed, while heading angles can change. Such a model is combined with another MINLP formulation, that maximizes the number of aircraft conflicts solved via speed regulation. These two MINLP formulations are solved using COUENNE. Furthermore, within a two-step methodology, the solution of the second one is used as a pre-processing step for the first one. This methodology significantly speeds-up the resolution process.

SR and HAC are sometimes combined in the same formulation. In [98], the planar CRP is formulated as a nonconvex Quadratically Constrained Quadratic Programming (QCQP) problem where the objective function minimizes the deviations from the original velocity vector. If the solution of the "natural" SDP relaxation of this QCQP formulation has rank one, then the problem is solved; otherwise, a locally optimal and conflict-free solution with a certain crossing pattern can be obtained via a stochastic rank reduction procedure. A different approach, which also combines SR and HAC to find optimal aircraft maneuvers, is proposed in [164]. In this case, a formulation in complex numbers with disjunctive constraints is introduced; speed bounds are translated into nonconvex quadratic constraints by considering the Euclidean norm of the vectors of velocities; different relaxations of the resulting MINLP problem are then proposed, solved, and compared.

Some works consider only HAC strategy. This is, for example, the case of [161], where trajectories are modeled with B-splines and a SIP formulation of CRP via HAC is presented, reformulated with an exact penalty function, and solved using local optimization methods. A two-step approach is introduced in [19]. The first step consists of a nonconvex MINLP model based on geometric constructions, which aims to minimize the total HAC cost (potentially the angle variation of each aircraft has a cost, even if in the experiments reported in the paper such costs are set to 1) to obtain the new conflict-free flight configuration. The second step consists of a set of unconstrained quadratic optimization models solved as a post-processing step to return each aircraft to its original flight plan as soon as possible after conflict resolution.

As for the first approach in [19], part of the literature focuses on the geometric characterization of conflicts, used in SR or HAC based models. In [44], for example, the geometric characteristics of aircraft trajectories are used in order to obtain closed-form expressions for single planar conflicts, based on SR and HAC alone, as well as closed-form expressions yielding minimum deviations from the original trajectories with combined SR and HAC. The authors of [158] present a geometric analysis of the conflicts leading to two MILP formulations: one for SR and another for HAC. The resulting separation constraints are linear on speeds and heading angles, respectively.

Some approaches consider more than one flight level. In this case, the corresponding optimization problem includes both flight-level allocation and conflict resolution within each level. For instance, the authors of [193] present a MILP formulation where conflict situations are avoided by performing both speed and altitude changes over predefined routes. The objective is to minimize the expected fuel costs of the aircraft. Binary variables are used to assign flight levels, which indicate whether two aircraft fly at different altitudes, as well as allowing deconfliction of aircraft traversing the same flight level. A multi-objective MILP approach in a similar vein, based on both maneuver types, and aiming at an equitable distribution of the maneuvers over the aircraft, is proposed in [18]. In [164], two disjunctive formulations are proposed for the CRP based on speed and altitude changes. Their objective functions penalize the number of changes linearly or quadratically, giving rise to a MILP program or a Mixed Integer Quadratic Program, respectively.

In order to test the different approaches, several benchmarking instances have been used, either generated accordingly to predefined scenarios or randomly. In particular, *circle instances* (presented for the first time in [98], under the name "symmetric encounter pattern") are characterized by an unrealistic highly symmetric configuration, with aircraft placed on a circumference and flying exactly or almost exactly towards its center; *random circle instances*, which are more realistic than the previous ones, include aircraft having trajectories with a starting angle deviation with respect to the diameter of the circle. The last predefined scenarios commonly used in the literature are the so called *grid scenario*, firstly presented in [153] as "Perpendicular crossing stream", and the more general *rhomboidal scenario*, introduced in [98] with the name "Crossing aircraft stream". These kinds of instances consider aircraft moving along several crossing straight trajectories. If the trails intersect at right angles, we speak of a grid scenario, otherwise, this kind of scenario has been called rhomboidal [160] and flow [163]. The public GitHub repository [163] gathers benchmarking instances of the circle, random circle, grid, and rhomboidal problems. There are also benchmarks that represent random configurations, for which both initial positions and aircraft velocity vectors are randomly generated in a squared region [20, 51]. If we consider instances in $\mathbb{R}^3$, i.e., aircraft while changing their altitude, *spherical* and *polyhedral* instances, which are generalization of circle and rhomboidal instances respectively, are available in the public repository [4].

4.3 Conflict resolution via speed regulation

Given a set of aircraft $\mathcal{A} = \{1, \dots, n\}$ sharing the same airspace, the goal of the approach presented in this section is to minimize the total speed changes needed to satisfy the minimum safety distance [1] d for each pair of aircraft $(i, j) \in \mathcal{A} \times \mathcal{A}$

[1]Expressed in Nautical Miles (NM), where 1 NM = 1852 m

in the given time horizon $[0, T]$. An important assumption is that changes occur instantaneously (i.e., when $t = 0$) and that the new speeds remain constant in the time horizon. Specifically, given a constant planned speed for every aircraft v_i, our formulation decides new optimal constant speeds satisfying the safety constraints. Such new speed is given by the product $q_i v_i$, where q_i is the variable of our formulations expressing the ratio of the implemented speed w.r.t. the initially planned speed of aircraft i. Variable $q_i = 1$ if the speed is equal to the initially planned one, $q_i > 1$ if it is increased, $q_i < 1$ if it is decreased, $\forall i \in \mathcal{A}$. Scale factor q_i is considered to range between q_i^{min} and q_i^{max}, with $q_i^{\mathsf{min}} < 1 < q_i^{\mathsf{max}}$. These bounds are often quite strict, in order to have speed variations barely perceived by human air traffic controllers. In this section, the vector of velocity direction components of each aircraft, u_i, is constant through the time horizon taken into account.

The set of dimension indices is $K = \{1, \ldots, k_{\max}\}$. In our implementation, we consider either $k_{\max} = 2$ (2D instances, Figure 4.1) or $k_{\max} = 3$ (3D instances, Figure 4.2). In this framework, the k-th component of the position vector of aircraft i at time t is defined as $x_{ik}(t) = x_{ik}^0 + t q_i v_i u_{ik}$, where $x_{ik}^0 = x_{ik}(0)$ is the k-th component of the initial position of aircraft i.

4.3.1 Natural formulation

The following provides a "natural" way to formulate CRP via SR:

$$\min_{q} \quad \sum_{i \in \mathcal{A}} (q_i - 1)^2 \tag{4.1a}$$

$$\forall i \in \mathcal{A} \quad q_i^{\mathsf{min}} \leqslant q_i \leqslant q_i^{\mathsf{max}} \tag{4.1b}$$

$$\forall i < j \in \mathcal{A},\ t \in [0, T] \quad \sum_{k \in K} \left[(x_{ik}^0 - x_{jk}^0) + t(q_i v_i u_{ik} - q_j v_j u_{jk}) \right]^2 \geqslant d^2. \tag{4.1c}$$

Formulation (4.1a)–(4.1c) is a SIP program, where the last set of inequalities (Eq. (4.1c)) contains uncountably many constraints, which ensure aircraft separation. Specifically, Eq. (4.1c) requires the squared Euclidean distance between each two aircraft i and j to be greater than or equal to d^2 at each instant t in the time window $[0, T]$.

The (convex) objective function is the sum of squared aircraft speed changes. This is equivalent to finding the feasible solution with the minimum speed change, which must be in $[q_i^{\mathsf{min}}, q_i^{\mathsf{max}}]$ for every aircraft i. As mentioned earlier, each aircraft i will start flying with the implemented speed, which is equal to $v_i q_i$.

4.3.2 Polynomial programming formulation

In order to address the issue of uncountably many constraints (4.1c) of the natural formulation, in this section we reformulate it using Polynomial Programming (PP). The fact that such SIP problems could be reformulated to PP ones (either via Lasserre-type semidefinite relaxations [199] or kinetic distance matrices [184]) was previously known. Previous results, however, only offered relaxations, because of the large size and polynomial degree of the corresponding (nonconvex) formulations. We provide here a reasonably small PP formulation having the same polynomial degree as the original SIP problem, which can be solved in practice to derive feasible solutions.

Polishing the polynomial

For each $i < j \in \mathcal{A}$, we define the polynomial:

$$p_{ij}(t) := \sum_{k\in K} \left[\left(x^0_{ik} - x^0_{jk}\right) + t\left(q_i v_i u_{ik} - q_j v_j u_{jk}\right)\right]^2 - d^2$$

in function of t. We have:

$$\begin{aligned}
p_{ij}(t) &= \sum_{k\in K} \left[\left(x^0_{ik} - x^0_{jk}\right) + t\left(q_i v_i u_{ik} - q_j v_j u_{jk}\right)\right]^2 - d^2 \\
&= \sum_{k\in K} \Big[\left(x^0_{ik} - x^0_{jk}\right)^2 + t^2 q_i^2 \left(v_i u_{ik}\right)^2 + t^2 q_j^2 (v_j u_{jk})^2 \\
&\quad -2t^2\left(v_i u_{ik} v_j u_{jk}\right) q_i q_j + 2t\left(x^0_{ik} - x^0_{jk}\right)\left(v_i u_{ik}\right) q_i \\
&\quad -2t\left(x^0_{ik} - x^0_{jk}\right)\left(v_j u_{jk}\right) q_j\Big] - d^2 \\
&= \sum_{k\in K}\left(x^0_{ik} - x^0_{jk}\right)^2 + t^2 q_i^2 \sum_{k\in K}\left(v_i u_{ik}\right)^2 + t^2 q_j^2 \sum_{k\in K}(v_j u_{jk})^2 \\
&\quad -2t^2 q_i q_j \sum_{k\in K}\left(v_i u_{ik} v_j u_{jk}\right) + 2t q_i \sum_{k\in K}\left(x^0_{ik} - x^0_{jk}\right)\left(v_i u_{ik}\right) \\
&\quad -2t q_j \sum_{k\in K}\left(x^0_{ik} - x^0_{jk}\right)\left(v_j u_{jk}\right) - d^2 \\
&= \left(B_i q_i^2 + B_j q_j^2 - 2C_{ij} q_i q_j\right) t^2 + 2\left(D^i_{ij} q_i - D^j_{ij} q_j\right) t \\
&\quad + A_{ij} - d^2,
\end{aligned}$$

where $A_{ij}, B_i, B_j, C_{ij}, D^i_{ij}, D^j_{ij}$ are constant (w.r.t. t) defined as follows:

$$\begin{aligned}
A_{ij} &:= \sum_{k\in K}\left(x^0_{ik} - x^0_{jk}\right)^2 & C_{ij} &:= \sum_{k\in K} v_i u_{ik} v_j u_{jk} \\
B_i &:= \sum_{k\in K}(v_i u_{ik})^2 & B_j &:= \sum_{k\in K}(v_j u_{jk})^2 \\
D^i_{ij} &:= \sum_{k\in K}\left(x^0_{ik} - x^0_{jk}\right)\left(v_i u_{ik}\right) & D^j_{ij} &:= \sum_{k\in K}\left(x^0_{ik} - x^0_{jk}\right)\left(v_j u_{jk}\right).
\end{aligned}$$

Thus,

$$p_{ij}(t) = (B_i q_i^2 + B_j q_j^2 - 2C_{ij} q_i q_j)t^2 + 2(D_{ij}^i q_i - D_{ij}^j q_j)t + A_{ij} - d^2 \tag{4.2}$$

is a polynomial of second degree in t.

We can now rewrite the SIP formulation (4.1a)–(4.1c) as:

$$\left.\begin{array}{rl} \min\limits_{q} & \sum\limits_{i \in \mathcal{A}} (q_i - 1)^2 \\ \forall i \in \mathcal{A} & q_i^{\mathsf{min}} \leqslant q_i \leqslant q_i^{\mathsf{max}} \\ \forall i < j \in A,\ \forall t \in [0, T] & p_{ij}(t) \geqslant 0. \end{array}\right\} \tag{4.3}$$

Problem (4.3) is the minimization of $\sum\limits_{i \in A} (q_i - 1)^2$ subject to the second degree polynomial $p_{ij}(t)$ being non-negative on $t \in [0, T]$, and q_i being in $[q_i^{\mathsf{min}}, q_i^{\mathsf{max}}]$ for each aircraft i.

Reformulation to polynomial programming

We introduce now a reformulation of (4.1a)–(4.1c) based on a result from [208]. This allows us to obtain a (finite) PP problem of the same degree of the original SIP formulation.

In particular, the following proposition is an immediate corollary of [208, Lemma 2.1].

Proposition 2 (corollary of Lemma 2.1 from [208])**.** *For any $i < j \in \mathcal{A}$, the polynomial $p_{ij}(t)$ is non-negative on $[0, T]$ iff there is a 2×2 positive semidefinite matrix*

$$M_{ij} = \begin{pmatrix} m_{ij} & r_{ij} \\ r_{ij} & g_{ij} \end{pmatrix} \succeq 0$$

and a nonnegative scalar $\mu_{ij} \geqslant 0$ such that:

$$p_{ij}(t) = (1\ t) M_{ij} \begin{pmatrix} 1 \\ t \end{pmatrix} + (T - t)t\mu_{ij}. \tag{4.4}$$

We use Proposition 2 to introduce an exact reformulation of the SIP problem (4.3), as shown in Theorem 4.

Theorem 4. *The following formulation:*

$$\left.\begin{array}{rrcl} \min\limits_{q,M,\mu} & \sum\limits_{i \in \mathcal{A}} (q_i - 1)^2 & & \\ \forall i < j \in \mathcal{A} & g_{ij} - \mu_{ij} & = & B_i q_i^2 + B_j q_j^2 - 2C_{ij} q_i q_j \\ \forall i < j \in \mathcal{A} & 2r_{ij} + T\mu_{ij} & = & 2\left(D_{ij}^i q_i - D_{ij}^j q_j\right) \\ \forall i < j \in \mathcal{A} & m_{ij} & = & A_{ij} - d^2 \\ \forall i < j \in \mathcal{A} & (r_{ij})^2 & \leqslant & m_{ij} g_{ij} \\ \forall i < j \in \mathcal{A} & m_{ij}, g_{ij}, \mu_{ij} & \geqslant & 0 \\ \forall i \in \mathcal{A} & q_i^{\mathsf{min}} \leqslant & q_i & \leqslant q_i^{\mathsf{max}} \end{array}\right\} \tag{4.5}$$

is an exact reformulation of (4.1a)–(4.1c).

Proof. Note that $p_{ij}(t)$ is given in two different forms in Eq. (4.2) and Eq. (4.4). We can therefore match coefficients of terms in t. This yields the following system:

$$\begin{array}{rcll} g_{ij} - \mu_{ij} & = & B_i q_i^2 + B_j q_j^2 - 2C_{ij} q_i q_j & \forall i < j \in \mathcal{A} \\ 2r_{ij} + T\mu_{ij} & = & 2(D^i_{ij} q_i - D^j_{ij} q_j) & \forall i < j \in \mathcal{A} \\ m_{ij} & = & A_{ij} - d^2 & \forall i < j \in \mathcal{A}, \end{array}$$

which is independent of t by construction. We now have to impose the constraints $M_{ij} \succeq 0$ and $\mu_{ij} \geqslant 0$ given in the statement of Prop. 2. For the former, we observe that the 2×2 matrix M_{ij} is positive semidefinite iff $(r_{ij})^2 \leqslant m_{ij} g_{ij}$ and $m_{ij}, g_{ij} \geqslant 0$, which yields the corresponding constraints in formulation (4.5). The latter is simply copied from formulation (4.3) to (4.5). □

We observe that problem (4.5) is a quadratic PP problem, and the degree is the same as in the original formulation (4.1a)–(4.1c). We also observe that formulation (4.5) is nonconvex in q because of the constraints $\forall i < j \in \mathcal{A}$

$$g_{ij} - \mu_{ij} = B_i q_i^2 + B_j q_j^2 - 2C_{ij} q_i q_j = \sum_{k \in K} (v_i u_{ik} q_i - v_j u_{jk} q_j)^2. \tag{4.6}$$

We remark that a convex relaxation can be readily obtained by relaxing Eq. (4.6) to

$$g_{ij} - \mu_{ij} \geqslant B_i q_i^2 + B_j q_j^2 - 2C_{ij} q_i q_j \quad \forall i < j \in \mathcal{A}. \tag{4.7}$$

4.3.3 Bilevel formulation

Another way to deal with the uncountably many constraints (4.1c) of the natural formulation is reformulating the CRP via SR into a bilevel formulation with multiple lower-level problems, one for each pair of aircraft $i < j \in \mathcal{A}$:

$$\min_{q,t} \quad \sum_{i \in \mathcal{A}} (q_i - 1)^2 \tag{4.8a}$$

$$\forall i \in \mathcal{A} \quad q_i^{\mathsf{min}} \leqslant q_i \leqslant q_i^{\mathsf{max}} \tag{4.8b}$$

$$\forall i < j \in \mathcal{A} \quad \min_{t_{ij} \in [0,T]} \sum_{k \in K} \left[(x^0_{ik} - x^0_{jk}) + t_{ij}(q_i v_i u_{ik} - q_j v_j u_{jk}) \right]^2 \geqslant d^2. \tag{4.8c}$$

Note that each lower-level problem is an optimization problem in the lower-level variables t_{ij}, parametrized by the upper-level variables q_i and q_j. An optimal solution of each lower-level problem, denoted by τ_{ij}, corresponds to the time instant at which aircraft i and j are closest. Formulation (4.8a)–(4.8c) is equivalent to (4.1a)–(4.1c) because, if aircraft pairwise separation constraints (constraints (4.8c)) hold at the time instant τ_{ij} which is a minimizer of aircraft relative distance (constraints (4.8c)), it will be true for each time instant in the time horizon $[0,T]$. It is evident that formulation (4.8a)–(4.8c) fits in the general setting of formulation (BP), with $d(x) = d^2$.

4.3.4 KKT reformulation

Since the presented bilevel formulation (4.8a)–(4.8c) has multiple convex lower-level problems for which Slater's condition holds, it can be easily reformulated to single-level by replacing the lower-level problem by its KKT conditions – see [77, Sec. 3.5] and [22] for the specific case of linear bilevel programs. This yields a single-level mathematical program with complementarity constraints.

Given the lower-level problem for each (i,j)

$$\begin{aligned} \min_{t_{ij}} \quad & \sum_{k\in K} \left[(x^0_{ik} - x^0_{jk}) + t_{ij}(q_i v_i u_{ik} - q_j v_j u_{jk})\right]^2 \\ s.t. \quad & -t_{ij} \leqslant 0 \ \wedge \ t_{ij} \leqslant T, \end{aligned} \tag{$SRLL_{ij}$}$$

and the KKT multipliers μ_{ij} and λ_{ij} defined for each pair of lower-level constraints $-t_{ij} \leqslant 0$ and $t_{ij} \leqslant T$ respectively, we have the following single-level reformulation of problem (4.8a)–(4.8c):

$$\min_{q,t,\mu,\lambda} \quad \sum_{i\in\mathcal{A}} (q_i - 1)^2 \tag{4.9a}$$
$$\forall i \in \mathcal{A} \quad q_i^{\mathsf{min}} \leqslant q_i \leqslant q_i^{\mathsf{max}} \tag{4.9b}$$
$$\forall i < j \in \mathcal{A} \quad \sum_{k\in K} \left[(x^0_{ik} - x^0_{jk}) + t_{ij}\psi_{ijk}\right]^2 \geqslant d^2 \tag{4.9c}$$
$$\forall i < j \in \mathcal{A} \quad \sum_{k\in K} \left[2t_{ij}\psi^2_{ijk} + 2(x^0_{ik} - x^0_{jk})\psi_{ijk}\right] - \mu_{ij} + \lambda_{ij} = 0 \tag{4.9d}$$
$$\forall i < j \in \mathcal{A} \quad \mu_{ij},\ \lambda_{ij} \geqslant 0 \tag{4.9e}$$
$$\forall i < j \in \mathcal{A} \quad \mu_{ij}\, t_{ij} = 0 \tag{4.9f}$$
$$\forall i < j \in \mathcal{A} \quad \lambda_{ij}\, t_{ij} - \lambda_{ij}\, T = 0 \tag{4.9g}$$
$$\forall i < j \in \mathcal{A} \quad 0 \leqslant t_{ij} \leqslant T, \tag{4.9h}$$

where the symbol ψ_{ijk} appearing in Eq. (4.9c) and (4.9d) is defined as:

$$\psi_{ijk} := q_i v_i u_{ik} - q_j v_j u_{jk}, \tag{4.10}$$

used throughout the chapter as short-hand for its definition in the right hand side.

Constraints (4.9d) (setting the gradient of the lower-level Lagrangian function equal to zero) correspond to the stationary condition of problem ($SRLL_{ij}$), Eq. (4.9e) and (4.9h) to dual and primal feasibility conditions respectively, and

Eq. (4.9f)–(4.9g) to complementary slackness. Eq. (4.9c) enforce the safety distance for each KKT solution. No dual variable is introduced for Eq. (4.9c) since they are upper-level constraints.

We remark that the complementarity constraints (4.9f)–(4.9g) involve products of continuous decision variables, and, therefore, define nonconvex feasible sets in general. A possible reformulation based on MILP modeling may define mixed-integer linear feasible sets instead, but also requires the determination of some big-M constant providing a valid bound to μ, λ, which cannot be done in polynomial time. This particular reformulation, moreover, would not dispose of the nonconvexities in constraints (4.9c) and (4.9d). We therefore propose to solve the formulation above by means of global optimization techniques (see Section 4.6).

4.3.5 Dual reformulations

We propose another reformulation of the bilevel problem (4.8a)–(4.8c), which arises because the lower-level problems ($SRLL_{ij}$) are convex quadratic, for which strong duality holds. We can therefore apply the *dual approach* presented in Section 3.3 of Chapter 3 to our bilevel formulation.

We observe that we can consider two different duals of the lower-level problems: Dorn's dual [85, 84], and Wolfe's dual [206].

Dorn's dual reformulation

Given the dual variables g_{ij} and z_{ij} of each lower-level problem in the left hand side of Eq. (4.8c) (defined for constraints $-t_{ij} \leqslant 0$ and $t_{ij} \leqslant T$ respectively), using Dorn's dual [85, 84], the following reformulation of (4.8a)–(4.8c) follows:

$$\min_{q,g,z} \quad \sum_{i \in \mathcal{A}} (q_i - 1)^2 \tag{4.11a}$$

$$\forall i \in \mathcal{A} \quad q_i^{\mathsf{min}} \leqslant q_i \leqslant q_i^{\mathsf{max}} \tag{4.11b}$$

$$\forall i < j \in \mathcal{A} \quad -\sum_{k \in K} \psi_{ijk}^2 \, g_{ij}^2 - T z_{ij} \geqslant d^2 - \sum_{k \in K} (x_{ik}^0 - x_{jk}^0)^2 \tag{4.11c}$$

$$\forall i < j \in \mathcal{A} \quad -\frac{z_{ij}}{2} - \sum_{k \in K} \psi_{ijk}^2 \, g_{ij} \leqslant \sum_{k \in K} (x_{ik}^0 - x_{jk}^0) \psi_{ijk} \tag{4.11d}$$

$$\forall i < j \in \mathcal{A} \quad z_{ij} \geqslant 0, \tag{4.11e}$$

obtained replacing the lower-level problems of Eq. (4.8a)–(4.8c) by their Dorn duals in the variables g_{ij}, z_{ij} for each aircraft pair $i < j \in \mathcal{A}$. This yields Eq. (4.11c)–(4.11d). Note that the primal lower-level variable t_{ij} does not appear in (4.11a)–(4.11e). This is not an issue because we just want to know the new aircraft speeds such that each potential conflict is avoided.

Proposition 3. *Eq. (4.11a)–(4.11e) is an exact reformulation of (4.8a)–(4.8c).*

Proof. By Dorn's duality theory [85], (D) is a dual problem of (P):

$$\left.\begin{array}{rrcl} \min\limits_{y} & \frac{1}{2}y^\top Q y + p^\top y & & \\ & Ay & \geqslant & b \\ & y & \geqslant & 0 \end{array}\right\} \quad (P) \qquad \left.\begin{array}{rrcl} \max\limits_{g,z} & -\frac{1}{2}g^\top Q g + b^\top z & & \\ & A^\top z - Qg & \leqslant & p \\ & z & \geqslant & 0 \end{array}\right\} \quad (D)$$

In our case, we have:

- $y := t_{ij}$,
- $Q := 2 \sum\limits_{k \in K} \psi_{ijk}^2$,
- $p := 2 \sum\limits_{k \in K} (x_{ik}^0 - x_{jk}^0)\psi_{ijk}$,
- $A := -1$,
- $b := -T$.

Recall that ψ_{ijk} is constant in the lower level since, by Eq. (4.10), it only depends on the upper-level variables q_i and q_j. By easy replacements, the formulation (4.11a)–(4.11e) follows. □

Wolfe's dual reformulation

Another single-level reformulation can be obtained using Wolfe's dual [206] of the convex lower-level problems in Eq. (4.8c). The lower-level dual objective function is the Lagrangian of the lower-level primal problem in Eq. (4.8c)

$$\sum_{k \in K} \left[(x_{ik}^0 - x_{jk}^0) + t_{ij}\psi_{ijk}\right]^2 + \alpha_{ij}(t_{ij} - T) - \beta_{ij} t_{ij},$$

where α_{ij} and β_{ij} are the Lagrangian multipliers associated to the constraints $-t_{ij} \leqslant 0$ and $t_{ij} \leqslant T$, respectively. Therefore, we obtain the following reformulation of Eq. (4.8a)–(4.8c):

$$\min_{q,t,\alpha,\beta} \quad \sum_{i \in \mathcal{A}} (q_i - 1)^2 \tag{4.12a}$$

$$\forall i \in \mathcal{A} \quad q_i^{\mathsf{min}} \leqslant q_i \leqslant q_i^{\mathsf{max}} \tag{4.12b}$$

$$\forall i < j \in \mathcal{A} \quad \sum_{k \in K} \left[(x_{ik}^0 - x_{jk}^0) + t_{ij}\psi_{ijk}\right]^2 + \alpha_{ij}(t_{ij} - T) - \beta_{ij} t_{ij} \geqslant d^2 \tag{4.12c}$$

$$\forall i < j \in \mathcal{A} \quad \sum_{k \in K} \left[2t_{ij}\psi_{ijk}^2 + 2(x_{ik}^0 - x_{jk}^0)\psi_{ijk} + \alpha_{ij} - \beta_{ij}\right] = 0 \tag{4.12d}$$

$$\forall i < j \in \mathcal{A} \quad \alpha_{ij}, \beta_{ij} \geqslant 0. \tag{4.12e}$$

We note that the single-level reformulation presented above involves some of the KKT conditions as constraints: the stationarity condition Eq. (4.12d) and the nonnegativity of the Lagrangian multipliers Eq. (4.12e). The (nonlinear) complementarity constraints, however, are not needed in Wolfe's duality [206]. The obtained reformulation (4.12a)–(4.12e) is exact.

Proposition 4. *Eq.* (4.12a)–(4.12e) *is an exact reformulation of Eq.* (4.8a)–(4.8c).

Proof. By Wolfe's duality theory [206], (D) is a dual problem of (P):

$$\left.\begin{array}{lrcl} \min\limits_{y} & \frac{1}{2}y^\top Qy + p^\top y + c & & \\ & Ay & \geqslant & b \\ & y & \geqslant & 0 \end{array}\right\} (P) \qquad \left.\begin{array}{lrcl} \max\limits_{\alpha,\beta} & \mathcal{L}(y,\alpha,\beta) & & \\ & \frac{\partial \mathcal{L}}{\partial y} & = & 0 \\ & \alpha,\beta & \geqslant & 0 \end{array}\right\} (D)$$

with $\mathcal{L}(y,\alpha,\beta) = \frac{1}{2}y^\top Qy + p^\top y + c + \alpha(b - Ay) - \beta y$, and $\frac{\partial \mathcal{L}}{\partial y} = Qy + p + \alpha - \beta$. In our case, we have:

- $y := t_{ij}$,
- $Q := 2\sum\limits_{k\in K} \psi_{ijk}^2$,
- $p := 2\sum\limits_{k\in K} (x_{ik}^0 - x_{jk}^0)\psi_{ijk}$,
- $c := \sum\limits_{k\in K} (x_{ik}^0 - x_{jk}^0)^2$,
- $A := -1$,
- $b := -T$.

Again, we recall that ψ_{ijk} is constant in the lower level because, by Eq. (4.10), it only depends on the upper level variables q_i and q_j. By easy replacements, the formulation (4.12a)–(4.12e) follows. □

4.4 Conflict resolution via heading angle changes

In this section, we present several formulations to model the CRP via HAC in two dimensions. The goal is again to satisfy the minimum safety distance d for each pair of aircraft while minimizing the total deviations with respect to the original flight plan. Given the initial heading angle ϕ_i of each aircraft i, the outcome of the HAC based CRP will be the set of new heading angles $\theta_i + \phi_i$ of the aircraft (see Figure 4.3), where the heading angle variations θ_i are the variables of the

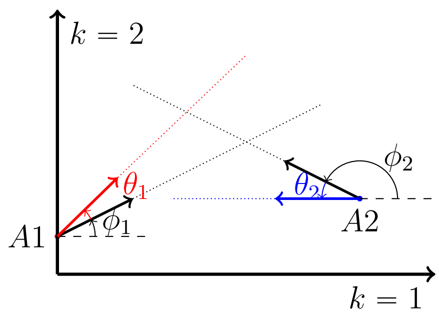

Figure 4.3: Heading angles of two aircraft (in black before deconfliction)

formulations introduced in this section. As well as speed modifications, also HAC are bounded, i.e., $\forall i \in \mathcal{A}\ \theta_i \in [\theta_i^{\mathsf{min}}, \theta_i^{\mathsf{max}}]$, with $\theta_i^{\mathsf{min}} < 0$ and $\theta_i^{\mathsf{max}} > 0$.

Since we consider aircraft at the same altitude level, the position of aircraft i at time t has only two components:

$$x_i(t) = x_i^0 + \cos(\phi_i + \theta_i)v_i t \quad \text{and} \quad y_i(t) = y_i^0 + \sin(\phi_i + \theta_i)v_i t,$$

where $x_i^0 = x_i(0)$ and $y_i^0 = y_i(0)$ are the first and the second component of the initial position of aircraft i, respectively.

4.4.1 Bilevel formulation

We introduce a bilevel formulation where the upper-level decision variables are the heading angle variations θ_i (for all $i \in \mathcal{A}$) and the lower-level decision variables are t_{ij} (for all $i < j \in \mathcal{A}$). Each lower-level problem is parametrized by the upper-level variables θ_i, θ_j.

$$\min_{\theta,t} \quad \sum_{i\in\mathcal{A}} \theta_i^2 \tag{4.13a}$$

$$\forall i \in \mathcal{A} \quad \theta_i^{\mathsf{min}} \leqslant \theta_i \leqslant \theta_i^{\mathsf{max}} \tag{4.13b}$$

$$\begin{aligned}\forall i < j \in \mathcal{A} \quad & \min_{t_{ij}\in[0,T]} \left[(x_i^0 - x_j^0) + t_{ij}(\cos(\phi_i+\theta_i)v_i - \cos(\phi_j+\theta_j)v_j)\right]^2 \\ & + \left[(y_i^0 - y_j^0) + t_{ij}(\sin(\phi_i+\theta_i)v_i - \sin(\phi_j+\theta_j)v_j)\right]^2 \geqslant d^2. \end{aligned} \tag{4.13c}$$

The convex objective function of the upper level (4.13a) is the sum of squared heading angle changes, which are bounded by $[\theta_i^{\mathsf{min}}, \theta_i^{\mathsf{max}}]$. Each lower-level objective is to minimize the squared Euclidean distance between aircraft i and j over $t_{ij} \in [0, T]$. Note that the lower-level objective function is also convex in t_{ij}. Similarly to Eq. (4.8c) of the previous section, Eq. (4.13c) guarantees that the minimum squared distance between each pair within the time horizon is at least d^2.

4.4.2 KKT reformulation

We derive the KKT reformulation of Eq. (4.13a)–(4.13c), based on KKT multipliers λ_{ij} (resp. μ_{ij}) associated to constraints $t_{ij} \leqslant T$ (resp. $-t_{ij} \leqslant 0$) of each lower level problem

$$\begin{aligned} \min_{t_{ij}} \quad & \Big[(x_i^0 - x_j^0) + t_{ij}(\cos(\phi_i + \theta_i)v_i - \cos(\phi_j + \theta_j)v_j)\Big]^2 \\ & + \Big[(y_i^0 - y_j^0) + t_{ij}(\sin(\phi_i + \theta_i)v_i - \sin(\phi_j + \theta_j)v_j)\Big]^2 \\ s.t. \quad & -t_{ij} \leqslant 0 \ \wedge \ t_{ij} \leqslant T. \end{aligned} \tag{$HACLL_{ij}$}$$

Since Slater's condition holds, and the lower level is convex in the variable t_{ij}, the following reformulation is exact [77, Sec. 3.5].

$$\min_{\theta,t,\lambda,\mu} \quad \sum_{i \in \mathcal{A}} \theta_i^2 \tag{4.14a}$$
$$\forall i \in \mathcal{A} \quad \theta_i^{\mathsf{min}} \leqslant \theta_i \leqslant \theta_i^{\mathsf{max}} \tag{4.14b}$$
$$\forall i < j \in \mathcal{A} \quad \big[(x_i^0 - x_j^0) + t_{ij}c_{ij}\big]^2 + \big[(y_i^0 - y_j^0) + t_{ij}s_{ij}\big]^2 \geqslant d^2 \tag{4.14c}$$
$$\forall i < j \in \mathcal{A} \quad 2t_{ij}(c_{ij}^2 + s_{ij}^2) + 2(x_i^0 - x_j^0)c_{ij} + 2(y_i^0 - y_j^0)s_{ij} + \lambda_{ij} - \mu_{ij} = 0 \tag{4.14d}$$
$$\forall i < j \in \mathcal{A} \quad \lambda_{ij},\ \mu_{ij} \geqslant 0 \tag{4.14e}$$
$$\forall i < j \in \mathcal{A} \quad \lambda_{ij}\, t_{ij} - \lambda_{ij}\, T = 0 \tag{4.14f}$$
$$\forall i < j \in \mathcal{A} \quad \mu_{ij}\, t_{ij} = 0 \tag{4.14g}$$
$$\forall i < j \in \mathcal{A} \quad 0 \leqslant t_{ij} \leqslant T, \tag{4.14h}$$

where the symbols c_{ij} and s_{ij} are shorthand for the nonlinear expressions

$$c_{ij} := \cos(\phi_i + \theta_i)v_i - \cos(\phi_j + \theta_j)v_j, \quad \text{and} \quad s_{ij} := \sin(\phi_i + \theta_i)v_i - \sin(\phi_j + \theta_j)v_j.$$

The formulation in Eq. (4.14a)–(4.14c) is a single-level NonLinear Programming (NLP) problem in the variables θ, t, λ, and μ. Constraints (4.14d) correspond to stationarity conditions of the lower-level problems, Eq. (4.14h) to primal feasibility, Eq. (4.14e) to dual feasibility, Eq. (4.14f) and Eq. (4.14g) to complementarity conditions. We require that the safety distance is satisfied for each pair of aircraft and for each t_{ij} satisfying the KKT conditions imposed in (4.14d)–(4.14h), with constraints (4.14c).

4.4.3 Dual reformulations

We follow the procedure discussed in Section 4.3.5 for the SR based CRP, and in Section 3.3 of Chapter 3 in order to obtain two *dual reformulations* of Eq. (4.13a)–(4.13c). The first one involves Dorn's dual of the lower-level problems, while the second one involves Wolfe's dual.

Dorn's dual reformulation

Given the dual variables g_{ij} and z_{ij} of the lower-level problems, the following reformulation of Eq. (4.13a)–(4.13c) is obtained:

$$\min_{\theta,g,z} \quad \sum_{i\in\mathcal{A}} \theta_i^2 \tag{4.15a}$$

$$\forall i \in \mathcal{A} \quad \theta_i^{\mathsf{min}} \leqslant \theta_i \leqslant \theta_i^{\mathsf{max}} \tag{4.15b}$$

$$\forall i < j \in \mathcal{A} \quad -g_{ij}^2(c_{ij}^2 + s_{ij}^2) - Tz_{ij} \geqslant d^2 - (x_i^0 - x_j^0)^2 - (y_i^0 - y_j^0)^2 \tag{4.15c}$$

$$\forall i < j \in \mathcal{A} \quad -\frac{z_{ij}}{2} - (c_{ij}^2 + s_{ij}^2)g_{ij} \leqslant (x_i^0 - x_j^0)c_{ij} + (y_i^0 - y_j^0)s_{ij} \tag{4.15d}$$

$$\forall i < j \in \mathcal{A} \quad z_{ij} \geqslant 0. \tag{4.15e}$$

The formulation in Eq. (4.15a)–(4.15e) is a single-level problem in the variables θ, g, and z, the exactness of which can be proved in a way analogous to what is done for Proposition 3. Note that the primal lower-level variable t_{ij} does not appear in (4.15a)–(4.15e), but we just want to know the new heading angles such that each potential conflict is avoided.

Wolfe's dual reformulation

Using Wolfe's dual of each lower-level problem in the variables α_{ij} and β_{ij}, we obtain the following single-level reformulation of Eq. (4.13a)–(4.13c):

$$\min_{\theta,t,\alpha,\beta} \quad \sum_{i\in\mathcal{A}} \theta_i^2 \tag{4.16a}$$

$$\forall i \in \mathcal{A} \quad \theta_i^{\mathsf{min}} \leqslant \theta_i \leqslant \theta_i^{\mathsf{max}} \tag{4.16b}$$

$$\forall i < j \in \mathcal{A} \quad \left[(x_i^0 - x_j^0) + t_{ij}c_{ij}\right]^2 + \left[(y_i^0 - y_j^0) + t_{ij}s_{ij}\right]^2 + \alpha_{ij}(t_{ij} - T) - \beta_{ij}t_{ij} \geqslant d^2 \tag{4.16c}$$

$$\forall i < j \in \mathcal{A} \quad 2t_{ij}(c_{ij}^2 + s_{ij}^2) + 2(x_i^0 - x_j^0)c_{ij} + 2(y_i^0 - y_j^0)s_{ij} + \alpha_{ij} - \beta_{ij} = 0 \tag{4.16d}$$

$$\forall i < j \in \mathcal{A} \quad \alpha_{ij}, \beta_{ij} \geqslant 0. \tag{4.16e}$$

With Eq. (4.16c), the Lagrangian of each lower-level problem is required to exceed the minimum required safety distance. The stationarity KKT condition (gradient of the Lagrangian equal to zero) corresponds to (4.16d). Constraints (4.16e) impose the nonnegativity of the dual variables α_{ij} and β_{ij}. The exactness of formulation (4.16a)–(4.16e) can be proven as done for SR based CRP in Proposition 4.

4.5 Cutting plane algorithm

In Section 3.4 of Chapter 3 we have introduced a Cutting Plane algorithm to solve formulation (BP). In this section we tailor such CP algorithm 1 for the bilevel formulations (4.8a)–(4.8c) of SR based CRP and (4.13a)–(4.13c) of HAC based CRP.

The problem solved at each iteration of the CP the algorithm is nonconvex. In our implementation, its solution is obtained either with global solvers or, in the interest of efficiency, by executing a local NLP solver several times within a multistart procedure that starts from randomly chosen points (not necessarily returning the global optimum).

We assume that aircraft are separated at the beginning of the time horizon considered, otherwise the problem is infeasible.

Cutting plane algorithm for CRP via speed regulation

Algorithm 2 is a solution algorithm for the bilevel formulation (4.8a)–(4.8c), which iteratively defines the feasible set of the upper-level problem by means of quadratic cuts in the upper-level variables q. At each iteration h, the relaxation R_h of the original bilevel problem, obtained by considering the upper-level problem together with the cuts added in previous iterations, is solved. At the outset, R_0 is:

$$\begin{aligned} \min_{q} \quad & \sum_{i \in \mathcal{A}} (q_i - 1)^2 \\ \forall i \in \mathcal{A} \quad & q_i^{\mathsf{min}} \leqslant q_i \leqslant q_i^{\mathsf{max}}. \end{aligned}$$

The problem R_h, solved at each iteration of Algorithm 2, is nonconvex since constraints (4.17) are of the form $f(q_i, q_j) \geqslant d^2$ with $f(q_i, q_j)$ convex. Therefore, in order to find global optima of R_h, a global optimization algorithm should be employed. This, however, would make the algorithm excessively slow. In our implementation (see Section 4.6) we chose to heuristically solve R_h using a multistart algorithm calling a local NLP solver, from randomly chosen starting points, when global optimization solvers are too slow.

Note that τ_{ij}^h is the instant for which the distance between i and j is minimum. If this distance is greater than or equal to the safety value for each pair of aircraft, the algorithm terminates at Step 12, as q^h must be an optimal solution of the bilevel formulation. Note that, in Step 8, τ_{ij}^h, easily computed in closed form, is set to 0 or T if

$$-\frac{\sum_{k \in K} (x_{ik}^0 - x_{jk}^0)(q_i^h v_i u_{ik} - q_j^h v_j u_{jk})}{\sum_{k \in K} (q_i^h v_i u_{ik} - q_j^h v_j u_{jk})^2}$$

is negative or greater than T respectively.

Algorithm 2 CP algorithm for CRP via SR

1: Let $h = 0$. Initialize the relaxation R_h of the bilevel program (4.8a)–(4.8c), obtained by considering the upper-level problem only.
2: **while** true **do**
3: Solve R_h to obtain the optimal solution q^h.
4: **for** each aircraft pair (i, j) **do**
5: **if** $\forall\ k \in K : (q_i^h v_i u_{ik} - q_j^h v_j u_{jk}) = 0$ **then**
6: Set $\tau_{ij}^h = \frac{T}{2}$.
7: **else**
8: Compute the instant $\tau_{ij}^h \in [0, T]$ as

$$\tau_{ij}^h = \min\left\{T, \max\left\{0, -\frac{\sum_{k\in K}(x_{ik}^0 - x_{jk}^0)(q_i^h v_i u_{ik} - q_j^h v_j u_{jk})}{\sum_{k\in K}(q_i^h v_i u_{ik} - q_j^h v_j u_{jk})^2}\right\}\right\}.$$

9: **end if**
10: **end for**
11: **if** $\sum\limits_{k\in K}((x_{ik}^0 - x_{jk}^0) + \tau_{ij}^h(q_i^h v_i u_{ik} - q_j^h v_j u_{jk}))^2 \geqslant d^2 \quad \forall i < j \in \mathcal{A}$ **then**
12: The algorithm terminates and q^h is the optimal solution of the bilevel formulation.
13: **else**
14: For each pair (i, j) violating the inequality, define R_{h+1} as R_h with the adjoined inequality:

$$\sum_{k\in K}((x_{ik}^0 - x_{jk}^0) + \tau_{ij}^h(q_i v_i u_{ik} - q_j v_j u_{jk}))^2 \geqslant d^2. \tag{4.17}$$

15: $h := h + 1$
16: **end if**
17: **end while**

In Step 6, τ_{ij}^h is set to $\frac{T}{2}$ if $(q_i^h v_i u_{ik} - q_j^h v_j u_{jk}) = 0$, for all $k \in K$, i.e., if aircraft i and j fly on parallel trajectories with the same speed. Having assumed that aircraft are separated at the beginning (namely $\sum\limits_{k\in K}(x_{ik}^0 - x_{jk}^0)^2 \geqslant d^2$), no cut will be added in the next steps of the algorithm.

Cutting plane algorithm for CRP via HAC

We propose a tailored version of the CP algorithm 1 for the bilevel formulation (4.13a)–(4.13c), which models the HAC based CRP. In this case, the nonconvex problem R_0 solved at the first iteration is

$$\begin{aligned} \min_{\theta} \quad & \sum_{i \in \mathcal{A}} \theta_i^2 \\ \forall i \in \mathcal{A} \quad & \theta_i^{\mathsf{min}} \leqslant \theta_i \leqslant \theta_i^{\mathsf{max}}. \end{aligned}$$

Algorithm 3 CP algorithm for CRP via HAC

Let $h = 0$. Initialize the relaxation R_h of the bilevel program, obtained by considering the upper-level problem only.
while true **do**
  Solve R_h to obtain the optimal solution θ^h.
  for each aircraft pair (i, j) **do**
    if $c_{ij}^h = 0$ and $s_{ij}^h = 0$ **then**
      Set $\tau_{ij}^h = \frac{T}{2}$.
    else
      Compute the instant $\tau_{ij}^h \in [0, T]$ as

$$\tau_{ij}^h = \min\left\{T, \max\left\{0, -\frac{(x_i^0 - x_j^0)c_{ij}^h + (y_i^0 - y_j^0)s_{ij}^h}{(c_{ij}^h)^2 + (s_{ij}^h)^2}\right\}\right\},$$

      with $c_{ij}^h = \cos(\phi_i + \theta_i^h)v_i - \cos(\phi_j + \theta_j^h)v_j$ and $s_{ij}^h = \sin(\phi_i + \theta_i^h)v_i - \sin(\phi_j + \theta_j^h)v_j$.
    end if
  end for
  if $\left[(x_i^0 - x_j^0) + \tau_{ij}^h c_{ij}^h\right]^2 + \left[(y_i^0 - y_j^0) + \tau_{ij}^h s_{ij}^h\right]^2 \geqslant d^2 \quad \forall i < j \in \mathcal{A}$ **then**
    The algorithm terminates and θ^h is the optimal solution of the bilevel formulation.
  else
    For each pair (i, j) violating the inequality, define R_{h+1} as R_h with the adjoined inequality:

$$\begin{aligned} &\left[(x_i^0 - x_j^0) + \tau_{ij}^h(\cos(\phi_i + \theta_i)v_i - \cos(\phi_j + \theta_j)v_j)\right]^2 \\ &+ \left[(y_i^0 - y_j^0) + \tau_{ij}^h(\sin(\phi_i + \theta_i)v_i - \sin(\phi_j + \theta_j)v_j)\right]^2 \geqslant d^2. \end{aligned} \tag{4.18}$$

    $h := h + 1$
  end if
end while

Again, the problem R_h, solved at each iteration of the algorithm, is nonconvex since the constraints (4.18) are of the form $f(\theta_i, \theta_j) \geqslant d^2$ with $f(\theta_i, \theta_j)$ convex. We find θ^h in Step 3 using a global NLP solver or, when the time limit is exceeded, with a local NLP solver within a multistart procedure from randomly chosen starting points. As in Algorithm 2, $\tau_{ij}^h \in [0, T]$ indicates when the distance between i and j is minimized and it is always computed in closed form in Step 8. If this distance satisfies the safety threshold for each pair of aircraft, the algorithm terminates at Step 12. Again, in Step 6 we discarded the case in which τ_{ij}^h is not well defined whenever aircraft i and j share the same direction and speed.

4.6 Computational experiments

For the CRP via SR in 2 dimensions, we test our approach using the set of instances used in [171]. It consists of *circle* —see Figure 4.4— and *rhomboidal instances*, where aircraft move along crossing directed trails intersecting in n_c conflict points.

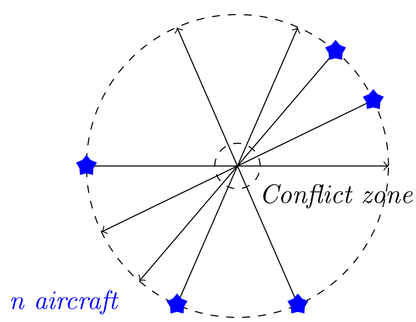

Figure 4.4: n conflicting aircraft flying towards the center of a circle

For the CRP via SR in 3 dimensions we use a 3D generalization of circle instances (named *spherical instances* in Table 4.1), where n aircraft are placed on a sphere of a given radius r — see Figure 4.5. We consider also *polyhedral instances* in which aircraft move along straight 3D trajectories, which intersect in at least $\frac{n}{2}$ conflict points. These instances are available online at the public repository [4].

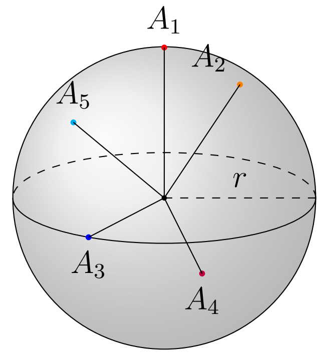

Figure 4.5: n conflicting aircraft flying towards the center of a sphere

A trajectory is defined by two angles: the so-called pitch angle γ_i (angle that the vector of the direction u_i forms with the axis k_3) and the heading angle ϕ_i (angle between the projection of u_i onto the k_1k_2-plane and the axis k_1) — see Figure 4.6.

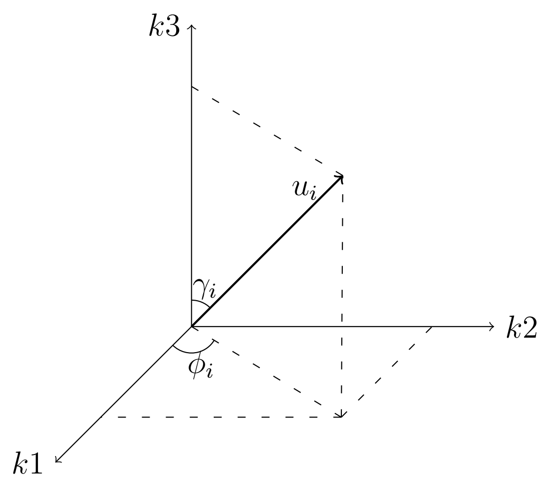

Figure 4.6: The 3-dimensional airspace

Finally, we test our approaches for the CRP via HAC in 2 dimensions using the set of instances proposed in [50], consisting of both *circle instances* and *random circle instances*. In such instances aircraft are placed around a circle and have trajectories with a starting heading angle ϕ_i randomly chosen in $[-\frac{\pi}{6}, \frac{\pi}{6}]$ with respect to the diameter of the circle. Moreover, we test some *rhomboidal instances* from [171] in which aircraft move along straight trajectories intersecting in n_c conflict points.

In all the experiments, we consider standard safety distance $d = 5\,\text{NM}$ and a time horizon of $T = 2$ hours. We implement the formulations, as well as the CP algorithms, using the AMPL modeling language [95]. All the solvers are run with their default settings. The tests are performed on a 2.53GHz Intel(R) Xeon(R) CPU with 49.4 GB RAM.

CRP via speed regulation

For the *circle* and *spherical* instances the initial speed is $v_i = 400\,\mathrm{NM/h}$ for each $i \in \mathcal{A}$ and the angles γ_i and ϕ_i are randomly generated ($\gamma_i = \frac{\pi}{2}\ \forall i \in \mathcal{A}$ for circle instances). Parameters x_{ik}^0 and u_{ik} are given by

$$u_{i1} = \cos(\phi_i)\sin(\gamma_i), \quad u_{i2} = \sin(\phi_i)\sin(\gamma_i), \quad u_{i3} = \cos(\gamma_i), \quad x_{ik}^0 = -r\,u_{ik}$$

where the radius r is chosen in $\{100, 200, \ldots, 700\}$. The bounds q_i^{min} and q_i^{max} are set to 0.94 and 1.03 respectively, following the weaker bounds proposed by the ERASMUS project. We decided to stick to these well-known bounds on q_i also for the 3D cases, having in mind the CRP application. In the context of urban air mobility, vehicles might have less strict bounds on admissible deceleration or acceleration. However, urban air mobility concepts are still being developed and estimations on these control parameters are not yet available.

We solve the single-level reformulations (4.9a)–(4.9h), (4.11a)–(4.11e), and (4.12a)–(4.12e) with the global optimization solver Baron [174] (B in the Table 4.1). When Baron exceeds the time-limit (set to 3600 seconds), we use a Multistart algorithm (MS in the tables), which performs 1000 calls to the local NLP solver Snopt [104] from randomly sampled starting points.

We solve the bilevel formulation (4.8a)–(4.8c) using the CP algorithm in Section 4.5 (CP in Table 4.1) with maximum iteration number set to 1000; at each iteration we solve the relaxed formulation R_h using Snopt called 50 times within a Multistart procedure from randomly chosen points.

We solve the PP formulation (4.5) with the global optimization solver Gurobi [108] (G in Table 4.1). For cases in which Gurobi exceeds its time limit set to 3600 seconds, we use a Multistart algorithm, which performs 1000 calls to the Ipopt local NLP solver [198] from randomly sampled starting points.

All the results are reported in Table 4.1. The headings are the following: n number of aircraft; r radius of the sphere in NM; *obj* best objective value found by each model; *cpu* computing time in seconds; *slv* solver used (for the CP algorithm the solver used to solve the inner problem R_h); *UB* and *LB* respectively upper and lower bound on the optimal solution value, determined by the global solver (Baron or Gurobi) when time limit is reached (we write 0 whenever a bound is less than 10^{-6}); *it* number) of CP iterations, i.e., number of times R_h is solved. The results on *circle* and *rhomboidal* instances are also compared with those that are obtained by solving the MINLP formulation proposed in [171] with Baron, with default options but again with a time limit of 3600 seconds –see last column of Table 4.1. Whenever this time limit is reached, we report in italics the best feasible solution found by the solver within the time limit.

The value of the objective function is always very small, given the nature of the problem (q must be in $[0.94, 1.03]$). The tight speed variation bounds imposed by ERASMUS project lead to an additional complication since instances are not guaranteed to be feasible. Best objective values and minimum required time are reported in bold for each instance. The best formulation in terms of solution quality is the PP one in Eq. (4.5). In terms of computational efficiency, for most of the instances the CP Algorithm 2 is the best. Since we are in the context of the *tactical* conflict resolution, which deals with potential conflicts that are about to happen within a few (i.e., from five to thirty) minutes, the proposed approaches are concretely promising.

CRP in 2 dimensions via heading angles changes

As mentioned above, the HAC based CRP instances are taken from [50, 171]. The authors set $v_i = 400\,\mathrm{NM/h}$ for each $i \in \mathcal{A}$ for all the instances. For the *circle instances* the angles ϕ_i are randomly generated and parameters x_i^0 and y_i^0 are given by

$$x_i^0 = -r\cos(\phi_i), \quad y_i^0 = -r\sin(\phi_i).$$

For the *random circle instances* both the angles ϕ_i and the parameters x_i^0 and y_i^0 are randomly generated. The bounds θ_i^{min} and θ_i^{max} are set to $-\pi/6$ and $\pi/6$ respectively.

We solve the formulations with the global optimization solver COUENNE [32] (C in the Table 4.2). We do not use BARON because it cannot handle the trigonometric functions sine and cosine. When COUENNE exceeds the time-limit (set to 3600 seconds), we use a Multistart algorithm (MS in Table 4.2). It performs 1000 calls to SNOPT for KKT reformulation (4.14a)–(4.14c) and Wolfe's dual reformulation (4.16a)–(4.16e), and 1000 calls to IPOPT [198] in the case of Dorn's dual reformulation (4.15a)–(4.15e). In all of the calls, starting points are randomly chosen.

We solve the bilevel formulation using the CP algorithm in Section 4.5 (CP in Table 4.2) with maximum iteration number set to 1000; at each iteration we solve the relaxed formulation R_h using COUENNE, or, when the CP exceed the time-limit of 3600 seconds, the local solver SNOPT called 50 times within a Multistart procedure from randomly chosen points.

Our results are reported in Tables 4.2 and 4.3. The headings are the following: name of the instance; n number of aircraft; n_c number of potential conflicts; *obj* best objective value found by each model; *cpu* computing time in seconds; *slv* solver used (in the last column, the solver used to solve the inner problem R_h of the CP algorithm); *UB* and *LB* respectively upper and lower bound on the optimal solution value, determined by COUENNE when time limit is reached (we write 0 whenever a bound is less than 10^{-6}); *it* number of CP iterations, i.e., number of times R_h is solved. Table 4.2 includes Circle instances, while Table 4.3 is divided into two blocks, which correspond to Random Circle and Rhomboidal instances. Both Table 4.2 and each block of Table 4.3 are followed by a row that shows the percentage of instances for which each approach outperforms (or is as good as) the rest. Only instances solved with the same method (namely C or MS) were used for the average. In Tables 4.2 and 4.3 the results on *circle* and *random circle* instances are also compared with those that are obtained using HAC only (without pre-processing) in [50]. Best objective values and minimum required time are reported in bold for each instance.

Among the models proposed, for most of the instances the CP algorithm is the best in terms of objective function and computational time, even when the inner problem R_h is solved using COUENNE. Looking at the comparison of our results with the ones obtained in [50], it appears that they are comparable.

4.7 Conflict resolution problem: a benchmark generator

Despite CRP being a very challenging optimization problem, the question of testing the different mathematical optimization approaches against each other is still open. In Section 4.6 we used the literature instances to compare our approaches against previously proposed methods. However, these instances are not always representative of the real-world problems as all the flights move towards a common point or are randomly generated. There is a lack of a common set of test instances that allows comparison of the available methods under a variety of heterogeneous and representative scenarios. Starting from the procedure used to generate the 3D instances we used to test CRP via SR in three dimensions (see Section 4.6), in [5] we present a flight deconfliction benchmark generator that allows the user to choose between (i) different predefined scenario inspired on existing benchmarks in the literature; (ii) pseudo-random traffic meeting some user-predefined congestion; (iii) and randomly generated traffic. While (i) and (iii) aim at standardizing existing benchmarks, to the best of our knowledge, (ii) provides a novel framework that allows to tune the resulting instance intricacy. Scenarios in (i) and (iii) typically stand for overcrowded traffic layouts which are unrealistic, and completely random traffic with no other congestion indicator than the space window and number of aircraft, respectively. Conversely, (ii) allows the user to decide on congestion indicator such as the number of conflicts, probability of conflict, and max number of conflict per aircraft of the generated instance.

In particular, we develop a generator in which six predefined scenarios are available, corresponding to the 2D and 3D implementations of three different layouts that exist in the literature: circle, rhomboidal, grid, spherical, polyhedral, cubic instances. In each scenario, both the initial positions and nominal vectors of velocity of the aircraft are generated according to a predefined configuration. Moreover, we provide a random variant for each of the six scenarios, which consists in adding a random deviation to the predefined nominal vectors of velocity. The range of such random deviation can be selected by the user. Each scenario is characterized by certain parameters, which can be tuned by the user. For further technical details on the options and parameters of the generator, we refer the reader to the user manual.

In contrast to scenario-based instances, we also consider free-flight instances generation. That is, we do not assume the aircraft flying under an specific layout. Instead of that, we consider a rectangle/parallelepiped (air sector) in which the initial positions and nominal vectors are generated randomly or pseudo-randomly.

On the one hand, randomly generated instances have been used in previous literature [20, 51, 98]. However, the lack of a common reference makes the different studies not comparable. We provide the option of randomly generating both 2D and 3D instances in order to offer such common reference.

On the other hand, we address pseudo-random instances for the first time (to the best of our knowledge). This consists in generating aircraft configurations that meet a particular level of congestion, which is determined by the user through different parameters. Namely, we define the following parameters related to traffic congestion:

- n_c: total number of pairs of aircraft that are in conflict;
- p_c: probability that one aircraft is in conflict with at least another aircraft;
- m_c: maximum number of aircraft that a fixed aircraft can be in conflict with.

The expected number of conflicts is given by:

$$E(n_c) = n \cdot p_c \cdot \left(\frac{1 + m_c}{2}\right) \cdot \frac{1}{2}. \tag{4.19}$$

Therefore, the user would only need to introduce two of these three parameters (the remaining one would be calculated according to (4.19)). The maximum number of conflicts that can be generated is given by $(n \cdot m_c/2)$, thus, we discourage the use of values of n_c and m_c such that $m_c < 2 \cdot n_c/n$. On the other hand, the average number of conflict is at most $(n \cdot (m_c + 1)/4)$ (if $p_c = 1$). Then, if both n_c and m_c are input by the user, we recommend to use values such that $m_c \geqslant 4 \cdot n_c/n - 1$. The user may decide to input just some of these three parameters or none of them. If no parameter is specified, then $p_c = 0.5$, $m_c = n - 1$ by default, and n_c is fixed to the closer integer to $E(n_c)$.

4.8 Conclusion

In this chapter we proposed both polynomial and bilevel programming as suitable approaches to model the well-known aircraft conflict resolution problem or CRP. In particular, we presented two bilevel formulations of the CRP: one based on speed regulation in k dimensions and another where potential conflicts are avoided via heading angle changes in two dimensions. In both cases, the convexity of the lower-level problems allowed us to derive three different single-level problems respectively, using KKT conditions, Dorn's duality, and Wolfe's duality. We presented a single-level polynomial programming formulation of the speed regulation based CRP as well.

The single-level formulations of both problems were solved by using state-of-the-art solvers, which provided good solutions in reasonable computing time. Alternatively, we proposed a tailored cut generation algorithm (in the same vein of CP Algorithm 1 of Chapter 3) to solve the bilevel formulations. This algorithm, compared with state-of-the-art solvers, outperformed the other approaches in terms of efficiencies. Numerical results, when compared with other approaches in the literature, are encouraging and stress the potential of the proposed approach.

Table 4.1: Results obtained solving 4 different formulations of CRP via SR in 2 and 3 dimensions. The results on 2D instances are compared with those obtained in [171].

Instances		KKT reformulation					Dorn's dual reformulation					Wolfe's dual reformulation					CP				PP formulation					[171]
n	*r*	*obj*	*cpu*	slv	UB	LB	*obj*	*cpu*	slv	UB	LB	*obj*	*cpu*	slv	UB	LB	*obj*	*cpu*	slv	it	*obj*	*cpu*	slv	UB	LB	*obj*
Circle																										
2	100	0.002527	0.61	B	-	-	**0.002526**	0.52	B	-	-	0.002530	0.75	B	-	-	0.002530	1.45	B	8	0.002531	**0.10**	G	-	-	0.002531
3	200	0.001665	**2.04**	B	-	-	**0.001659**	19.91	B	-	-	0.001665	3.46	B	-	-	0.001665	99.16	B	59	0.001663	**1.05**	G	-	-	0.001666
4	200	0.004028	18.59	B	-	-	0.004022	300.4	B	-	-	0.004023	839.7	B	-	-	0.004027	**15.06**	B	11	**0.004021**	19.5	G	-	-	0.004026
5	300	0.003056	6.20	MS	0.000698	0.003055	0.003056	4.70	MS	0.000961	0.003049	0.003056	9.76	MS	0.000107	0.003053	0.003056	**3.30**	MS	17	**0.003049**	46.1	G	-	-	0.003054
6	300	0.006058	9.10	MS	0.000113	0.00605	0.006058	17.41	MS	0.000214	0.006414	0.006058	13.11	MS	0	0.006794	0.006058	**8.24**	MS	30	**0.006044**	89.1	G	-	-	*0.006088*
Rhomboidal																										
6	-	0.001254	21.04	MS	0	0.001254	0.001254	28.01	MS	0	0.001254	0.001254	33.97	MS	0	0.001252	0.001254	**1.09**	MS	8	**0.001253**	12.9	G	-	-	*0.001253*
7	-	**0.001591**	24.37	MS	0	0.001591	**0.001591**	50.01	MS	0	0.001591	**0.001591**	109.3	MS	0	0.001591	**0.001591**	**1.71**	MS	9	**0.001591**	3.18	G	-	-	*0.002140*
7	-	0.001566	20.86	MS	0	0.001623	0.001566	51.04	MS	0	0.001566	0.001566	89.68	MS	0	0.001566	0.001566	**1.54**	MS	9	**0.001565**	5.55	G	-	-	*0.001865*
8	-	**0.002384**	37.08	MS	0	0.002384	**0.002384**	62.05	MS	0	0.002384	**0.002384**	41.81	MS	0	0.002384	**0.002384**	**1.22**	MS	9	**0.002384**	112	G	-	-	*0.002384*
10	-	0.001400	41.54	MS	0	0.001465	**0.001397**	150.2	MS	0	0.001397	**0.001397**	89.49	MS	0	0.001397	**0.001397**	**1.41**	MS	7	0.001398	536	G	-	-	*0.001465*
Spherical																										
2	100	**0.002220**	0.63	B	-	-	0.002223	0.86	B	-	-	0.002226	0.62	B	-	-	0.002226	0.68	B	5	0.002226	**0.19**	G	-	-	-
3	200	0.001406	9.64	B	-	-	**0.001404**	5.86	B	-	-	0.001407	11.4	B	-	-	0.001408	**0.66**	MS	10	0.001405	1.78	G	-	-	-
4	200	0.003713	312	B	-	-	**0.003703**	270	B	-	-	0.003709	3582	B	-	-	0.003714	**0.86**	MS	11	0.003708	5.60	G	-	-	-
5	300	0.002959	5.90	MS	0.002959	0.000070	0.002959	4.53	MS	0.002959	0.000473	0.002959	11.7	MS	0.002959	0	0.002959	**1.19**	MS	13	**0.002946**	25.5	G	-	-	-
6	300	0.005847	9.69	MS	0.005847	0.000001	0.005847	7.41	MS	0.005847	0.000188	0.005847	12.6	MS	0.005847	0	0.005916	**3.05**	MS	24	**0.005817**	251	G	-	-	-
7	500	**0.002855**	15.3	MS	0.002903	0	**0.002855**	20.2	MS	0.002855	0	0.002856	26.4	MS	0.002906	0	0.002887	**12.2**	MS	41	**0.002855**	209	MS	0.002831	0	-
8	500	0.004549	**16.8**	MS	0.004557	0	0.004605	35.6	MS	0.004641	0	0.004572	23.1	MS	0.004744	0	0.004521	64.09	MS	55	**0.004513**	104	MS	0.004616	0	-
9	500	**0.006987**	**20.4**	MS	0.007491	0	0.007137	53.0	MS	0.007188	0	0.007109	28.1	MS	0.007380	0	0.007135	78.9	MS	64	**0.006987**	127	MS	0.007352	0	-
10	600	0.006410	**32.2**	MS	0.006484	0	0.006436	70.7	MS	0.006420	0	**0.006402**	34.7	MS	0.006408	0	0.006497	204	MS	121	0.006405	180	MS	0.007319	0	-
12	700	0.008511	79.4	MS	0.008713	0	0.008685	**57.4**	MS	0.008963	0	**0.008404**	81.4	MS	0.008800	0	0.008676	624	MS	172	0.008440	137	MS	0.009456	0	-
Polyhedral																										
2	-	**0.000305**	**0.14**	B	-	-	**0.000305**	0.17	B	-	-	**0.000305**	0.26	B	-	-	**0.000305**	1.05	B	10	**0.000305**	0.54	G	-	-	-
4	-	**0.003278**	54.9	B	-	-	0.003283	5.19	MS	0.003278	0.001648	0.003709	3584	B	-	-	0.003283	**0.46**	MS	6	0.003282	1.72	G	-	-	-
6	-	0.006004	14.5	MS	0.006001	0.000288	0.006004	10.2	MS	0.006004	0.000003	0.006004	16.2	MS	0.006004	0	**0.006003**	**0.45**	MS	7	**0.006003**	5.67	G	-	-	-
8	-	0.011705	19.0	MS	0.011705	0.000043	0.011705	20.9	MS	0.011705	0	0.011705	17.9	MS	0.011705	0	0.011704	**0.41**	MS	6	**0.011703**	3.91	G	-	-	-
10	-	**0.015025**	34.5	MS	0.015025	0.000174	**0.015025**	67.3	MS	0.015025	0	**0.015025**	30.0	MS	0.015025	0	**0.015025**	**0.63**	MS	8	**0.015025**	15.6	G	-	-	-

Table 4.2: Results obtained solving 4 different formulations of CRP via HAC compared with those obtained in [50], *Circle instances*

Instances			KKT reformulation					Dorn's dual reformulation					Wolfe's dual reformulation					CP				[50]
Name	n	n_c	*obj*	*cpu*	*slv*	UB	LB	*obj*	*cpu*	*slv*	UB	LB	*obj*	*cpu*	*slv*	UB	LB	*obj*	*cpu*	*slv*	it	*obj*
Circle																						
pb_n2	2	1	**0.001250**	0.15	C	-	-	**0.001250**	0.14	C	-	-	**0.001250**	0.18	C	-	-	**0.001250**	**0.11**	C	2	**0.001250**
pb_n3_1	3	3	**0.002501**	0.83	C	-	-	**0.002501**	1.27	C	-	-	**0.002501**	0.83	C	-	-	**0.002501**	**0.74**	C	3	**0.002501**
pb_n3_2	3	3	0.006672	7.83	C	-	-	0.006672	5.82	C	-	-	**0.006657**	7.89	C	-	-	0.006671	**2.00**	MS	20	0.006665
pb_n3_3	3	3	**0.000950**	**1.03**	C	-	-	**0.000950**	1.16	C	-	-	**0.000950**	1.26	C	-	-	**0.000950**	1.67	C	6	**0.000950**
pb_n4_1	4	6	0.007240	**416**	C	-	-	0.007240	529.65	C	-	-	**0.007228**	1410	C	-	-	0.007240	**1.64**	MS	15	0.007240
pb_n4_2	4	6	**0.017061**	2497	C	-	-	0.017094	38.2	MS	0.017095	0.017074	0.017095	21.8	MS	0.017065	0.0169614	0.017098	**3.25**	MS	27	0.017065
pb_n4_3	4	6	**0.001318**	**25.3**	C	-	-	**0.001318**	85.2	C	-	-	**0.001318**	33.5	C	-	-	**0.001318**	57.1	C	10	**0.001318**
pb_n5_1	5	10	0.011629	25.3	MS	0.011629	0.004543	0.011629	38.4	MS	0.011629	0.011615	0.011629	27.4	MS	0.011629	0.004583	**0.011627**	**2.16**	MS	17	0.011629
pb_n5_2	5	10	**0.018467**	28.1	MS	0.018467	0.003903	**0.018467**	37.7	MS	0.018467	0.010120	**0.018467**	30.0	MS	0.018467	0.003660	0.018483	**5.74**	MS	40	0.018468
pb_n5_3	5	10	0.017122	23.6	MS	0.017122	0.006728	0.017121	40.8	MS	0.017122	0.015603	0.017122	27.7	MS	0.017122	0.004560	0.017123	**4.68**	MS	29	**0.017100**
pb_n5_4	5	10	**0.014750**	22.9	MS	0.014750	0.014624	**0.014750**	40.5	MS	0.014750	0.009179	**0.014750**	27.6	MS	0.014750	0.003219	**0.014750**	**3.42**	MS	24	**0.014750**
pb_n5_5	5	10	0.012163	33.0	MS	0.012163	0.012096	0.012163	53.4	MS	0.012163	0.012148	0.012163	35.0	MS	0.012163	0.005287	0.012162	**6.09**	MS	24	**0.012149**
pb_n5_6	5	10	0.011235	36.7	MS	0.011235	0.005837	0.011235	111	MS	0.011235	0.011217	0.011235	38.2	MS	0.011235	0.008566	0.011235	**1.71**	MS	15	**0.011225**
pb_n5_7	5	10	0.012273	31.6	MS	0.012273	0.003918	0.012273	42.3	MS	0.012273	0.012257	0.012273	33.3	MS	0.012273	0.009218	0.012273	**2.46**	MS	20	**0.012262**
pb_n5_8	5	10	0.017556	36.0	MS	0.017556	0.002707	0.017556	42.0	MS	0.017556	0.008683	0.017556	39.5	MS	0.017556	0.006314	**0.017555**	**3.47**	MS	26	0.017556
pb_n5_9	5	10	0.019140	29.1	MS	0.019140	0.007309	0.019140	46.2	MS	0.019140	0.015657	0.019140	32.9	MS	0.019140	0.007695	0.019138	**5.59**	MS	35	**0.019119**
pb_n5_10	5	10	0.026022	20.9	MS	0.026022	0.006197	0.026022	42.3	MS	0.026022	0.014691	0.026022	23.6	MS	0.026022	0.005229	0.026051	**5.18**	MS	32	**0.025960**
pb_n5_11	5	10	0.011202	42.4	MS	0.011202	0.011167	0.011199	38.2	MS	0.011201	0.011188	0.011202	37.4	MS	0.011201	0.011110	0.011199	**2.30**	MS	19	**0.011190**
pb_n5_12	5	10	0.012122	23.7	MS	0.012122	0.009387	0.012121	39.0	MS	0.012122	0.007619	0.012122	25.3	MS	0.012122	0.007107	0.012121	**2.55**	MS	19	**0.012111**
pb_n5_13	5	10	0.023265	32.1	MS	0.023265	0.005860	0.023265	43.9	MS	0.023265	0.009149	0.023265	33.4	MS	0.023265	0.004473	**0.023264**	**5.82**	MS	40	0.023265
pb_n5_14	5	10	**0.013789**	28.1	MS	0.013789	0.006102	**0.013789**	40.6	MS	0.013789	0.013385	**0.013789**	32.7	MS	0.013789	0.005103	**0.013789**	**3.55**	MS	25	0.013790
pb_n5_15	5	10	0.014578	32.9	MS	0.014578	0.006508	0.014578	51.4	MS	0.014578	0.007278	0.014578	35.3	MS	0.014578	0.005682	0.014577	**3.28**	MS	25	**0.014551**
pb_n5_16	5	10	0.010378	31.9	MS	0.010378	0.006824	0.010378	93.8	MS	0.010378	0.010367	0.010378	35.8	MS	0.010378	0.007561	0.010375	**1.73**	MS	15	**0.010367**
pb_n5_17	5	10	0.011956	34.9	MS	0.011956	0.008863	0.011956	38.8	MS	0.011956	0.008161	0.011956	39.2	MS	0.011956	0.005857	0.011956	**2.05**	MS	17	**0.011940**
pb_n5_18	5	10	0.011166	24.6	MS	0.011166	0.007224	0.011166	32.1	MS	0.011166	0.011154	0.011166	27.5	MS	0.011166	0.011106	0.011166	**2.41**	MS	18	**0.011153**
pb_n5_19	5	10	0.009920	2498	C	-	-	0.009920	1613	C	-	-	0.009920	36.9	MS	0.009920	0.008954	**0.009919**	**0.75**	MS	8	0.009920
pb_n5_20	5	10	0.019774	27.6	MS	0.019774	0.006149	0.019773	430	MS	0.019773	0.010206	0.019774	29.2	MS	0.019773	0.004218	0.019773	**5.34**	MS	34	**0.019739**
pb_n5_21	5	10	**0.009050**	1402	C	-	-	**0.009050**	1503	C	-	-	0.009051	61.8	MS	0.009050	0.009022	0.009051	**0.30**	MS	4	0.009051
pb_n5_22	5	10	0.030611	30.5	MS	0.030610	0.003752	0.030610	49.3	MS	0.030610	0.013442	0.030611	33.0	MS	0.030610	0.003154	0.030638	**12.4**	MS	53	**0.030577**
pb_n5_23	5	10	0.001543	**27.4**	MS	0.001543	0.001530	0.001543	38.9	MS	0.001550	0.001523	0.001543	40.1	MS	0.001543	0.001528	**0.001542**	2655	C	26	0.001543
pb_n6_1	6	15	0.001667	49.9	MS	0.001667	0.000080	0.001667	43.2	MS	0.001949	0.000273	0.001667	71.3	MS	0.001667	0.000095	0.001667	**2.44**	MS	17	**0.001649**
pb_n6_2	6	15	0.001667	51.5	MS	0.001667	0.000078	0.001667	44.0	MS	0.007084	0.000164	0.001667	75.8	MS	0.001667	0.000197	0.001667	**2.46**	MS	17	**0.001661**
% best *cpu*			7.69					0					0					92.31				

Table 4.3: Results obtained solving 4 different formulations of HACCRP compared with those obtained in [50], *Random Circle* and *Rhomboidal instances*

Instances			KKT reformulation					Dorn's dual reformulation					Wolfe's dual reformulation					CP				[50]
Name	n	n_c	*obj*	*cpu*	*slv*	UB	LB	*obj*	*cpu*	*slv*	UB	LB	*obj*	*cpu*	*slv*	UB	LB	*obj*	*cpu*	*slv*	it	*obj*
Random Circle																						
*rpb*2_1	2	1	**0.000141**	0.07	C	-	-	**0.000141**	**0.06**	C	-	-	**0.000141**	**0.06**	C	-	-	**0.000141**	0.07	C	2	**0.000141**
*rpb*2_2	2	1	**0.000795**	0.07	C	-	-	**0.000795**	**0.05**	C	-	-	**0.000795**	**0.06**	C	-	-	**0.000795**	**0.06**	C	2	**0.000795**
*rpb*3_1	3	2	**0.000078**	0.39	C	-	-	**0.000078**	2946	C	-	-	**0.000078**	**0.29**	C	-	-	**0.000078**	0.45	C	6	**0.000078**
*rpb*3_2	3	2	0.000515	0.63	C	-	-	**0.000509**	9.9	C	-	-	0.000515	0.85	C	-	-	0.000515	**0.62**	C	6	0.000513
*rpb*3_3	3	1	0.000114	0.11	C	-	-	0.000114	234	C	-	-	0.000114	0.11	C	-	-	0.000114	**0.06**	C	2	**0.000113**
*rpb*4_1	4	1	**0.000156**	4.36	C	-	-	**0.000156**	4.70	C	-	-	**0.000156**	3.65	C	-	-	**0.000156**	**0.06**	C	2	**0.000156**
*rpb*4_2	4	3	0.001177	13.3	C	-	-	0.001177	24.7	MS	0.001206	0.001172	0.001177	28.6	C	-	-	0.001177	**1.28**	C	7	**0.001175**
*rpb*4_3	4	2	**0.000202**	1.55	C	-	-	**0.000202**	3.39	C	-	-	**0.000202**	6.39	C	-	-	**0.000202**	**0.17**	C	2	**0.000202**
*rpb*5_1	5	6	0.000413	356	C	-	-	0.000413	39.3	MS	0.000413	0.000407	0.000413	319	C	-	-	0.000413	**2.60**	C	6	**0.000408**
*rpb*5_2	5	3	0.000454	1323	C	-	-	0.000454	40.7	MS	0.004249	0.000327	0.000454	36.4	MS	0.000454	0.000430	0.000454	**14.6**	C	11	**0.000450**
*rpb*5_3	5	8	0.000612	213	C	-	-	0.000612	39.2	MS	0.001263	0.000602	0.000612	171	C	-	-	0.000612	**6.77**	C	7	0.000613
*rpb*6_1	6	5	0.000965	**44.2**	MS	0.000965	0.000053	0.000965	48.1	MS	0.009621	0.000146	0.000965	58.3	MS	0.000965	0.000012	0.000964	794	C	16	**0.000955**
*rpb*6_2	6	9	0.000858	50.7	MS	0.001133	0.000103	0.000858	45.4	MS	0.018710	0.000094	0.000858	58.8	MS	0.000858	0.000071	0.000858	**5.60**	C	6	**0.000855**
*rpb*6_3	6	4	0.000694	**44.0**	MS	0.000694	0.000032	0.000694	49.5	MS	0.006783	0.000070	0.000694	57.9	MS	0.000694	0.000008	0.000694	61.5	C	11	**0.000693**
*rpb*7_1	7	2	0.002199	50.7	MS	0.000213	0	0.000213	60.6	MS	0.000313	0.000106	0.000213	96.0	MS	0.000213	0.000053	0.000213	**0.49**	C	6	**0.000210**
*rpb*7_2	7	7	**0.001161**	64.5	MS	0.001161	0	**0.001161**	64.8	MS	0.003693	0.000046	**0.001161**	89.0	MS	0.001161	0	**0.001161**	**2.46**	MS	17	0.001162
*rpb*7_3	7	9	0.002794	66.4	MS	0.002418	0	**0.001914**	58.1	MS	0.003851	0.000040	**0.001914**	99.8	MS	0.001914	0	**0.001914**	**2.52**	MS	13	0.002637
*rpb*8_1	8	12	0.001174	99.1	MS	0.001237	0	0.001174	82.9	MS	0.006447	0.000007	0.001174	93.8	MS	0.001174	0	**0.001173**	**3.07**	MS	15	0.001189
*rpb*8_2	8	5	**0.000370**	102	MS	0.000370	0	**0.000370**	**79.6**	MS	0.015749	0	**0.000370**	97.2	MS	0.000370	0	**0.000370**	150	C	14	0.000373
*rpb*8_3	8	10	**0.001006**	97.8	MS	0.001006	0.000002	0.001007	84.3	MS	0.006782	0	**0.001006**	98.7	MS	0.001006	0	**0.001006**	**7.50**	MS	28	0.001019
% best *cpu*			0					18.18					18.18					72.73				
Rhomboidal																						
*pb*6_5	6	5	0.128250	59.3	MS	0.128250	0	0.128249	40.4	MS	0.133770	0	0.128250	146	MS	0.128249	0	**0.128233**	**1.50**	MS	8	-
*pb*7_4	7	4	0.001628	219	MS	0.001672	0	**0.001602**	71.7	MS	0.001602	0	**0.001602**	327	MS	0.001602	0	**0.001602**	**2.89**	MS	12	-
*pb*7_6	7	6	0.001567	222	MS	0.001672	0	0.001567	75.6	MS	0.009484	0	0.001567	324	MS	0.001567	0	0.001566	**1.85**	MS	8	-
*pb*8_4	8	4	0.002401	108	MS	0.002548	0	**0.002384**	81.6	MS	0.003082	0	**0.002384**	261	MS	0.002384	0	**0.002384**	**1.73**	MS	8	-
*pb*10_10	10	10	0.127695	166	MS	0.669673	0	0.127693	90.9	MS	0.127693	0	0.127695	270	MS	0.127693	0	**0.127677**	**0.53**	MS	8	-
% best *cpu*			0					0					0					100				

Chapter 5

ACOPF in a bilevel framework and the network design ACOPF

In the first part of this chapter, we present a bilevel formulation which can be used to determine the optimal interaction between a retailer, who sets the price of power, and a network of consumers, who decide on storing, consuming, buying power from, or selling power to the grid. The main assumption is that the network routes Alternating Current (AC), i.e., the network routing problem solved at the lower level is the AC Optimal Power Flow (ACOPF). The main difficulties in this model come from the nonlinearities inherent in the physical laws governing the electrical equilibrium of the considered AC power network. To deal with these issues, we relax the lower level, using Second-Order Cone Programming (SOCP). Such modelization approach is the starting point of an ongoing work with Claudia D'Ambrosio, Leo Liberti, and Martin Schmidt. Some preliminary results on tiny instances, not reported in this dissertation, are encouraging, and lead us to believe in the interest of this approach.

Secondly, we consider the problem of optimally designing a power transportation network of this type with respect to line activity. In this framework, we formulate the ACOPF with on/off variables on lines as a nonconvex MINLP problem in complex numbers. Then we propose different convex MINLP relaxations. The results presented in this second part of the chapter are presented in [6], currently under review.

5.1 Introduction

The possibility for generation in private households has extended the activities of electricity customers from simply purchasing electricity, to taking a more active role within the production and storage of electricity, transforming themselves from purely power consumers to so-called "prosumers" (a fusion of the terms "producers" and "consumers", first introduced in [188]). In this scenario, we study an AC power grid, defined by a set of buses, and a set of lines, in a discretized time horizon. Specifically, we derive a bilevel formulation to model the interaction between a retailer and several prosumers. The retailer is not part of the grid, but is considered as an entity external to the set of prosumers who interact with each other through the network (each prosumer is a bus). The prosumers can produce power thanks to rooftop PV panels, and store it in a battery. Both these two devices are connected via inverters to the grid. Each prosumer can potentially decide to sell some of the produced/stored power, receiving money from the government, which pays a fixed *feed-in tariff* for each unit of power sold (we will later assume that the feed-in tariff is equal to the price of the power sold by the retailer). At the upper level, the retailer, who sets the price of power for the set of prosumers, aims to maximize his/her profit. At the lower level, prosumers (considered as acting together) want to maximize the revenues from selling power/minimize what is paid to the retailer when their production does not satisfy their power need.

In the second part of this chapter, we focus on the problem solved at the lower level, i.e., the ACOPF problem. We detail its formulation and consider the additional problem of network design. While this is similar to the optimal switching problem [93], here we start from the ACOPF formulation rather than its Direct Current (DC) counterpart. We present a nonconvex MINLP formulation of the network design problem derived from the ACOPF, and several convex MINLP relaxations. While there is little hope of solving even the tiniest network design ACOPF instances with the nonconvex MINLP formulation, we show that some results for small ACOPF instances can be obtained using the convex MINLP relaxations.

The chapter is organized as follows. In Section 5.2, we briefly discuss the related literature. In Section 5.3 the bilevel model is presented, and in Subsection 5.3.1 a convex relaxation based on Low's approach in [100] of the lower level is introduced. We present the nonconvex MINLP ACOPF formulation for the corresponding network design problem in Section 5.4. We, then, propose some new Mixed-Integer SOCP relaxations of this problem in Subsection 5.4.1. We test our single-level formulations of the power network design problem with some literature instances in Section 5.4.3.

5.2 Literature review

The literature on Optimal Power Flow (OPF) problem is rich and spans over half a century. In the literature, various reformulations, decomposition methods and algorithms have been proposed. We refer to [9] for an introduction to OPF and to

[58, 97, 96] for detailed surveys.

When considering the DC, a small angle-linearization of the ACOPF problem is solved, where losses and reactive power are ignored and a flat voltage profile is considered. The approach is widely used in practice, since the DC Optimal Power Flow (DCOPF) problem is a Linear Programming (LP) problem efficiently solvable even for large-sized networks (see [42, Sec. 1.2.4], [149, Eq. (5.48)]).

Solving the ACOPF problem is a more challenging task. Indeed, such problem can be naturally formulated in many ways, e.g., QCQP, PP, and general NLP, all of which involve nonconvexities [42]. The variables (voltage, current, power) are naturally defined on continuous domains. A very interesting feature of the ACOPF is that its variables range in complex numbers (see Appendix A for a recall on complex numbers). While a separation in real and imaginary parts is always possible, matrix formulations and relaxations generally take up twice the amount of storage w.r.t. working directly in complex numbers [103].

Several algorithms, summarized in [58], tackle the nonconvex ACOPF problem using NLP solution techniques which can only guarantee local optimality. Among these approaches we can find:

- gradient or reduced gradient methods [83, 21];
- sequential quadratic programming methods [47], exploiting the fact that the quadratic behaviour is sufficiently accurate for small deviations;
- Newton [183] or quasi-Newton methods [105], characterized by a fast convergence, but very sensitive for the initial conditions;
- interior-point methods [150, 116], usually applied for large scale OPF problem;
- derivative free optimization techniques [11, 112], typically applied when first and second-derivatives are too expensive to compute.

Empirically, interior-point methods work well on large-scale instances and find very good locally-optimal solutions. However, certifying a tight optimality gap is challenging. For this purpose, convex relaxations of the ACOPF problem (i.e., formulations obtained enlarging the ACOPF feasible region to a convex set) have been developed in the literature. Common relaxations are LP, SOCP, SDP [149].

In [57] a multi-period ACOPF problem with charge and discharge dynamics for energy storage devices is modeled. A convex relaxation based on SDP is proposed and the Lagrangian dual is derived to investigate the relationship between the storage variables and the Locational Marginal Prices (LMPs), i.e., price signals reflecting the marginal impact of an additional unit of power generation or demand at each location (bus) on the grid [55]. They are defined by the dual variables associated with the nodal active power flow balance constraints. Later on, in [200] a successive linear programming approach is used to solve the ACOPF problem, called SLP IV-ACOPF algorithm. Starting from the canonical formulation of the ACOPF in rectangular coordinates, a combination of linearization and reduction techniques is applied to the problem constraints. Real and reactive power power dispatch are iteratively co-optimized within the algorithm. The computational performance and convergence quality of the SLP IV-ACOPF are proved comparing it to an interior-point algorithm for solving the nonlinear ACOPF. The dual of the SLP IV-ACOPF inner problem (which is a linear problem itself) is studied in [133]. In [100] the ACOPF problem in distribution networks is studied. The nonconvex formulation is relaxed obtaining an SOCP exact relaxation, which optimum is also the global optimum of ACOPF, under a certain condition that can be checked prior to solving the SOCP relaxation itself.

A bilevel model with ACOPF in the lower level is proposed in [60], where the market equilibrium of an integrated heat-power distribution system with strategic providers and demand elasticity is studied. In the upper level, a provider bids offering prices of electricity and heat to the respective markets. In the lower level, the system operators solve the (both heat and power) market clearing problems, and determine the dispatch of their generation devices, energy contracts with the provider in case of need, as well as LMPs. The ACOPF-based electricity market clearing lower-level problem is nonlinear. To overcome such issue, first, convex relaxation is performed on the "branch flow based OPF model", replacing a nonconvex equality constraint with a rotated Second-Order Cone (SOC). Then, each rotated SOC is approximated via a polyhedral set, and the power market clearing finally gives rise to an LP. Bilevel programming is used also in [90] to model a different problem: demand response in organized energy markets. Demand response is a modification in the power consumption of a prosumer to better match the demand for power with the supply. In the proposed bilevel model, the lower level performs the economic dispatch of energy and generates the price (LMP) while the upper level minimizes the total amount of demand response subject to a net benefit requirement. In [122] the pessimistic variant of the same problem is considered, where the retailer prepares for the least favorable optimal responses from the consumers. It is indeed demonstrated that the set of optimal consumption schedules typically contains various responses that are equal for the follower, but bring radically different profits for the leader. The main contribution of the paper is an exact procedure for solving the pessimistic variant of the problem taken into account.

Network design problems defined on the OPF in DC can be readily formulated as MILP problems [125, 12]. To the best of our knowledge, the first paper exhibiting computational results for the ACOPF with binary variables used for design purposes is [168], where binary variables are used to switch generators and shunts on and off: a local NLP solver is deployed on a well-known continuous NLP reformulation of the corresponding nonconvex MINLP formulation. A perspective cut based relaxation of an ACOPF formulation with binary variables for switching generators on and off was proposed in [175]. Another possible approach for working with ACOPF involving binary variables is to apply network design modeling techniques involving binary variables to an LP or SOCP relaxation of the ACOPF. This was done in [176], which proposed inner and outer mixed-integer Diagonally Dominant Programming (DDP) formulations.

In the first part of this chapter we use bilevel programming to model the same interaction tackled in [106], but we assume that prosumers are connected to an AC power grid. We separate imaginary and real parts of the variables, in order to avoid dealing with complex numbers in the formulation. We, then, relax the lower level using SOCP. In the second part of the chapter, we move a step towards solving a "network design ACOPF" by integrating binary variables that control whether a line is active or not. Our objective is to decrease the number of active lines, while still satisfying demand.

5.3 The bilevel formulation

In this section we model the interaction among a retailer and a set of prosumers which are placed at the buses of an AC power network, in a given time horizon $[t_1, t_k]$, discretized as $T = \{t_1, t_2, \ldots, t_k\}$. The considered AC power grid is defined by the set of buses $N = \{1, \ldots, n\}$, and the set of lines L. We assume $|N| = n$ and $|L| = m$.

The amount of charged and discharged power in the battery by the prosumer $i \in N$ at period $t \in T$ is $c_i^t \geqslant 0$ and $d_i^t \geqslant 0$ respectively. During time period t, a rooftop PV plant generates power v_i^t that the prosumer i can either consume directly or store in a battery. Since PV production mainly depends on solar radiation, it cannot be controlled by the prosumer. Thus, v_i^t is a given parameter. We suppose that also the demand s_i^t is known for each prosumer i, and for each period t. Both the parameters v_i^t and s_i^t, as well as the variables c_i^t and d_i^t are complex numbers because of the cyclic nature of alternating current, and because of the inverters. The real part is the *active power*, while the imaginary part is *reactive*.

Assume that there is a reference bus, indexed by r, which has fixed voltage (with imaginary part set to zero) and flexible power injection for power balance. We abbreviate $(i, j) \in L$ (i.e., (i, j) is a line of the grid) by $i \to j$. If $i \to j$ or $j \to i$ we write $i \sim j$; otherwise we write $i \nsim j$.

For each period $t \in T$ and each bus $i \in N$, let V_i^t be its voltage, I_i^t its current injection, and S_i^t its power injection. In particular, for each period t,

$V_r^t = \Re(V_r^t)$ is the fixed voltage at the reference bus, and S_r^t is the power that it draws from the distribution network for power balance. For each line $i \sim j$, let $y_{ij} = \Re(y_{ij}) - \boldsymbol{i}\Im(y_{ij})$ denote its admittance.

All the sets, parameters, and variables used in our mathematical formulations, some of which already introduced before, are listed below.

Sets

- $T = \{t_1, t_2, \ldots, t_k\}$ set of periods taken into account for the analysis
- $N = \{1, \ldots, n\}$ set of buses of the grid
- $L \subseteq \{(i, j) : i, j \in N\}$ set of lines of the grid

Parameters

- $\widehat{\gamma}$ feed-in tariff that the government pays to the prosumer for each unit of active power sold
- $\check{\gamma}$ feed-in tariff that the government pays to the prosumer for each unit of reactive power sold
- s_i^t complex demand of the prosumer $i \in N$ during time period $t \in T$ (where $s_r^t = 0$)
- v_i^t complex power generated by the PV station at bus $i \in N$ during time period $t \in T$ (where $v_r^t = 0$)
- y_{ij} admittance of line $(i, j) \in L$ with $m_{ij} = \Re(y_{ij})$, and $n_{ij} = \Im(y_{ij})$
- $\overline{V}_i$, and $\underline{V}_i$ bounds on the voltage magnitude at bus $i \in N$
- $V_r^t = \Re(V_r^t)$ is the fixed voltage at the reference bus r at each period $t \in T$
- $\overline{B_i}$ capacity of the battery at bus $i \in N$
- $\alpha \in (0, 1]$ self-discharge rate of the battery, which is the same for each bus w.l.o.g.
- $\beta \in (0, 1]$ efficiency of the battery charging/discharging process, which is the same for each bus w.l.o.g.
- $\overline{c}_i$, and $\overline{d}_i$ the bounds on the absolute value of the amount of charged/discharged power in/from the battery by prosumer $i \in N$

Variables The upper-level variables, set by the retailer, and fixed for each period, are:

- price $\widehat{p}$ of active power,
- price $\check{p}$ of reactive power.

The lower-level variables are:

- the voltage V_i^t at bus $i \in N$ at period $t \in T$

– x_+^t the amount of power that the network of the prosumers buys from the retailer during period $t \in T$

– x_-^t the amount of power that the network of the prosumers sells, and puts into the grid during time $t \in T$

– c_i^t the amount of charged power in the battery, by prosumer $i \in N$ at period $t \in T$

– d_i^t the amount of discharged power from the battery, by prosumer $i \in N$ at period $t \in T$

– B_i^t the active power that is stored into the battery by prosumer $i \in N$ at period $t \in T$

The following equalities, derived by physical laws, hold at each period $t \in T$:

- Current balance and Ohm's law:

$$I_i^t = \sum_{j:j\sim i} y_{ij}(V_i^t - V_j^t),\ i \in N$$

- Power balance:

$$S_i^t = V_i^t(I_i^t)^*,\ i \in N$$

- Power flow equation:

$$S_i^t = -(s_i^t - v_i^t + c_i^t - d_i^t),\ i \in N$$

The three sets of equations can be combined into a single one for each $i \in N$:

$$v_i^t + d_i^t - c_i^t - s_i^t = V_i^t \sum_{j:j\sim i} [(V_i^t)^* - (V_j^t)^*]y_{ij}^*.$$

For the reference bus, the first side of the equation is $S_r^t = v_r^t + d_r^t - c_r^t - s_r^t = 0 + d_r^t - c_r^t + 0 = d_r^t - c_r^t$, and at the right side V_r^t is fixed and given.

The bilevel formulation of the problem, involving p as upper-level and c, d, B, V, x_+ and x_- as lower-level variables, is the following:

$$\max_{\widehat{p},\breve{p}} \widehat{p} \sum_{t\in T} \Re(x_+^t) + \breve{p} \sum_{t\in T} \Im(x_+^t) \tag{5.1a}$$

$$\text{s.t. } \widehat{p}, \breve{p} \geqslant 0 \text{ and } (c, d, B, V, x_+, x_-) \text{ solves (5.2)} \tag{5.1b}$$

with the lower level (5.2)

$$\max_{c,d,B,V,x_+,x_-} \sum_{t\in T}(\widehat{\gamma}\,\Re(x_-^t)+\breve{\gamma}\,\Im(x_-^t)-\widehat{p}\,\Re(x_+^t)-\breve{p}\,\Im(x_+^t)) \tag{5.2a}$$

$$\text{s.t. } \forall t\in T:\ \Re(x_+^t)-\Re(x_-^t)=\sum_{i\in N}(\Re(s_i^t)-\Re(v_i^t)+\Re(c_i^t)-\Re(d_i^t)) \tag{5.2b}$$

$$\forall t\in T:\ \Im(x_+^t)-\Im(x_-^t)=\sum_{i\in N}(\Im(s_i^t)-\Im(v_i^t)+\Im(c_i^t)-\Im(d_i^t)) \tag{5.2c}$$

$$\forall t\in T:\ \Re(x_+^t)\Re(x_-^t)=0 \tag{5.2d}$$

$$\forall t\in T:\ \Im(x_+^t)\Im(x_-^t)=0 \tag{5.2e}$$

$$\forall t\in T:\ \Re(x_+^t)\geqslant 0,\ \Re(x_-^t)\geqslant 0 \tag{5.2f}$$

$$\forall t\in T:\ \Im(x_+^t)\geqslant 0,\ \Im(x_-^t)\geqslant 0 \tag{5.2g}$$

$$\forall t\in T\backslash\{t_n\}\wedge\forall i\in N:\ B_i^{t+1}=\alpha B_i^t+\beta\,\Re(c_i^t)-\frac{\Re(d_i^t)}{\beta} \tag{5.2h}$$

$$\forall t\in T\wedge\forall i\in N:\ 0\leqslant B_i^t\leqslant \overline{B}_i \tag{5.2i}$$

$$\forall t\in T\wedge\forall i\in N:\ v_i^t+d_i^t-c_i^t-s_i^t=V_i^t\sum_{j:j\sim i}[(V_i^t)^*-(V_j^t)^*]y_{ij}^* \tag{5.2j}$$

$$\forall t\in T\wedge\forall i\in N:\ |c_i^t|^2\leqslant \overline{c}_i^2 \tag{5.2k}$$

$$\forall t\in T\wedge\forall i\in N:\ |d_i^t|^2\leqslant \overline{d}_i^2 \tag{5.2l}$$

$$\forall t\in T\wedge\forall i\in N:\ \underline{V}_i^2\leqslant |V_i^t|^2\leqslant \overline{V}_i^2. \tag{5.2m}$$

Constraints (5.2b) and (5.2c) define the amount of active and reactive power that the network of prosumers buys from/sells to the retailer respectively. At each period the prosumers either buy or sell active/reactive power, which is stated by the bilinear constraints (5.2d)–(5.2e). The amount of bought or sold active/reactive power must be nonnegative as ensured by (5.2f)–(5.2g). Constraints (5.2h)–(5.2i) model the storage of the active power into the battery. In particular, from (5.2h), the amount of active power that is in the battery at period $t+1$ is given by the amount of active power in the battery at period t minus the amount of active power charged ($\Re(c_i^t)\geqslant 0$) or discharged ($\Re(d_i^t)\geqslant 0$) at period t. Finally, constraints (5.2j)–(5.2m) model the power flow in the AC network of prosumers (the power balance equation (5.2j), for the reference bus r, becomes $-c_r^t+d_r^t=V_r^t\sum_{j:j\sim r}[(V_r^t)^*-(V_j^t)^*]y_{rj}^*$, where V_r^t is fixed and given for all t). Note that $|c_i^t|$ in (5.2k), $|d_i^t|$ in (5.2l), and $|V_i^t|$ in (5.2m) are the moduli of the complex numbers c_i^t, d_i^t, and V_i^t, respectively, and their square is a real number (see Appendix A for details).

Since it is difficult to deal with complementarity constraints (5.2d)–(5.2e) in the lower level, we make the assumption that the power is sold by the prosumers at the same prices $\widehat{p}$, and $\breve{p}$ at which it is sold by the retailer. In this case the upper-level, as well as the lower-level objective function reads:

$$\sum_{t\in T}\sum_{i\in N}\left(\widehat{p}\left(\Re(s_i^t)-\Re(d_i^t)+\Re(c_i^t)-\Re(d_i^t)\right)+\breve{p}\left(\Im(s_i^t)-\Im(v_i^t)+\Im(c_i^t)-\Im(d_i^t)\right)\right). \tag{5.3}$$

It is optimized w.r.t. $\widehat{p},\breve{p}\in\mathbb{R}_+$ in the upper level, and w.r.t. c,d,B,V in the lower level.

5.3.1 Convex relaxation of the lower level

The challenge in solving a problem involving constraints (5.2j)–(5.2m) mainly comes from the nonconvex quadratic equality constraints (5.2j). To overcome this challenge and obtain in the end a convex relaxation, as done in [100], first define

$$W_{ij}^t := V_i^t(V_j^t)^*,\ i\sim j \text{ or } i=j,$$

and

$$W^t\{i,j\} := \begin{pmatrix} W_{ii}^t & W_{ij}^t \\ W_{ji}^t & W_{jj}^t \end{pmatrix}.$$

$W^t\{i,j\}$ is an Hermitian matrix, i.e., $W_{ij}^t=(W_{ji}^t)^*$. Accordingly, since $W^t\{i,j\}$ is a PSD rank-one matrix, the power flow constraints (5.2j)–(5.2m) can be equivalently formulated as:

$$\forall t\in T \wedge \forall i\in N: v_i^t+d_i^t-c_i^t-s_i^t=\sum_{j:j\sim i}\left(W_{ii}^t-W_{ij}^t\right)y_{ij}^*, \tag{5.4a}$$
$$\forall t\in T \wedge \forall i\in N: |c_i^t|^2\leqslant \bar{c}_i^2, \tag{5.4b}$$
$$\forall t\in T \wedge \forall i\in N: |d_i^t|^2\leqslant \bar{d}_i^2, \tag{5.4c}$$
$$\forall t\in T \wedge \forall i\in N: \underline{V}_i^2\leqslant W_{ii}^t\leqslant \overline{V}_i^2, \tag{5.4d}$$
$$\forall t\in T \wedge \forall i\to j: W^t\{i,j\}\succeq 0, \tag{5.4e}$$
$$\forall t\in T \wedge \forall i\to j:\ \operatorname{rank}(W^t\{i,j\})=1. \tag{5.4f}$$

Let us recall that, as well as V_r^t, W_{rr}^t is constant and given. At this point we can relax the set defined by Eqs. (5.4a)–(5.4f) to a convex one, by relaxing the rank constraints in (5.4f).

Therefore, the lower-level feasible set that we consider in the following is defined by the following constraints:

$$\forall t \in T\backslash\{t_n\} \wedge \forall i \in N : B_i^{t+1} = \alpha B_i^t + \beta\, \Re(c_i^t) - \frac{\Re(d_i^t)}{\beta} \tag{5.5a}$$

$$\forall t \in T \wedge \forall i \in N : 0 \leqslant B_i^t \leqslant \overline{B}_i \tag{5.5b}$$

$$\forall t \in T \wedge \forall i \in N : v_i^t + d_i^t - c_i^t - s_i^t = \sum_{j:j\sim i} \left(W_{ii}^t - W_{ij}^t\right) y_{ij}^*, \tag{5.5c}$$

$$\forall t \in T \wedge \forall i \in N : |c_i^t|^2 \leqslant \overline{c}_i^2, \tag{5.5d}$$

$$\forall t \in T \wedge \forall i \in N : |d_i^t|^2 \leqslant \overline{d}_i^2, \tag{5.5e}$$

$$\forall t \in T \wedge \forall i \in N : \underline{V}_i^2 \leqslant W_{ii}^t \leqslant \overline{V}_i^2, \tag{5.5f}$$

$$\forall t \in T \wedge \forall i \rightarrow j : W^t\{i,j\} \succeq 0. \tag{5.5g}$$

As recalled in [126], using the Sylvester criterion, (5.4e) (i.e., (5.5g)) is equivalent to $W_{ij}^t W_{ji}^t \leqslant W_{ii}^t W_{jj}^t$, and $W_{ii}^t, W_{jj}^t \geqslant 0$, where the first inequality can be written as

$$\Re(W_{ij}^t)^2 + \Im(W_{ij}^t)^2 + \left(\frac{W_{ii}^t - W_{jj}^t}{2}\right)^2 \leqslant \left(\frac{W_{ii}^t + W_{jj}^t}{2}\right)^2,$$

which is a SOCP constraint.

5.3.2 Bilevel formulation with a convex lower-level problem

To avoid dealing with a model over complex numbers, we reformulate constraints (5.5c)

$$v_i^t + d_i^t - c_i^t - s_i^t = \sum_{j:j\sim i} \left(W_{ii}^t - W_{ij}^t\right) y_{ij}^*$$

by separating real and imaginary parts.

While $W_{ii}^t = V_i^t (V_i^t)^* = \Re(V_i^t)^2 + \Im(V_i^t)^2$ is a real number, $W_{ij}^t = V_i^t (V_j^t)^*$ with $i \neq j$ has a real $\Re(W_{ij}^t)$ and an imaginary part $\Im(W_{ij}^t)$. Define

- $a_{ij}^t = \Re(W_{ij}^t) = \Re(V_i^t)\Re(V_j^t) + \Im(V_i^t)\Im(V_j^t)$,
- $q_{ij}^t = \Im(W_{ij}^t) = -\Re(V_i^t)\Im(V_j^t) + \Re(V_j^t)\Im(V_i^t)$.

We note that $W_{ii}^t = a_{ii}^t$ for all i and t, and, in particular, $W_{rr}^t = a_{rr}^t = \Re(V_r^t)^2$, with $\Re(V_r^t)$ fixed.

The left-hand side of constraint (5.5c) is

$$v_i^t + d_i^t - c_i^t - s_i^t = \left[\Re(v_i^t) + \Re(d_i^t) - \Re(c_i^t) - \Re(s_i^t)\right] + \boldsymbol{i}\left[\Im(v_i^t) + \Im(d_i^t) - \Im(c_i^t) - \Im(s_i^t)\right].$$

Its right-hand side is

$$\sum_{j:j\sim i} \left(a^t_{ii} - a^t_{ij} - \boldsymbol{i} q^t_{ij}\right)\left(m_{ij} - \boldsymbol{i} n_{ij}\right) = \sum_{j:j\sim i} \left[\left(a^t_{ii} m_{ij} - a^t_{ij} m_{ij} - q^t_{ij} n_{ij}\right) + \boldsymbol{i}\left(-a^t_{ii} n_{ij} + a^t_{ij} n_{ij} - q^t_{ij} m_{ij}\right)\right]$$

We can then rewrite constraint (5.5c) in the following way:

$$\begin{aligned} &\left[\Re(v^t_i) + \Re(d^t_i) - \Re(c^t_i) - \Re(s^t_i)\right] + \boldsymbol{i}\left[\Im(v^t_i) + \Im(d^t_i) - \Im(c^t_i) - \Im(s^t_i)\right] \\ &= \sum_{j:j\sim i} \left[\left(a^t_{ii} m_{ij} - a^t_{ij} m_{ij} - q^t_{ij} n_{ij}\right) + \boldsymbol{i}\left(-a^t_{ii} n_{ij} + a^t_{ij} n_{ij} - q^t_{ij} m_{ij}\right)\right]. \end{aligned} \tag{5.6}$$

Since two complex numbers are equal if and only if their real and imaginary parts are equal respectively, Eq. (5.6) can be split into two equalities:

$$\Re(v^t_i) + \Re(d^t_i) - \Re(c^t_i) - \Re(s^t_i) = \sum_{j:j\sim i} \left(a^t_{ii} m_{ij} - a^t_{ij} m_{ij} - q^t_{ij} n_{ij}\right),$$

and

$$\Im(v^t_i) + \Im(d^t_i) - \Im(c^t_i) - \Im(s^t_i) = \sum_{j:j\sim i} \left(-a^t_{ii} n_{ij} + a^t_{ij} n_{ij} - q^t_{ij} m_{ij}\right).$$

The *final* bilevel formulation we obtain is

$$\max_{\widehat{p},\breve{p}} \sum_{t\in T}\sum_{i\in N} \left(\widehat{p}\left(\Re(v^t_i) + \Re(d^t_i) - \Re(c^t_i) - \Re(s^t_i)\right) + \breve{p}\left(\Im(v^t_i) + \Im(d^t_i) - \Im(c^t_i) - \Im(s^t_i)\right)\right) \tag{5.7a}$$

$$\text{s.t. } \widehat{p},\breve{p} \geqslant 0 \text{ and } (c,d,B,a,q) \text{ solves (5.8)} \tag{5.7b}$$

with the lower level

$$\min_{c,d,B,a,q} \sum_{t\in T}\sum_{i\in N} \left(\widehat{p}\left(\Re(v^t_i) + \Re(d^t_i) - \Re(c^t_i) - \Re(s^t_i)\right) + \breve{p}\left(\Im(v^t_i) + \Im(d^t_i) - \Im(c^t_i) - \Im(s^t_i)\right)\right) \tag{5.8a}$$

$$\text{s.t. } \forall t \in T\backslash\{t_n\} \wedge \forall i \in N:\ B^{t+1}_i = \alpha B^t_i + \beta\,\Re(c^t_i) - \frac{\Re(d^t_i)}{\beta} \tag{5.8b}$$

$$\forall t \in T \wedge \forall i \in N:\ 0 \leqslant B^t_i \leqslant \overline{B}_i \tag{5.8c}$$

$$\forall t \in T \wedge \forall i \in N:\ \Re(v^t_i) + \Re(d^t_i) - \Re(c^t_i) - \Re(s^t_i) = \sum_{j:j\sim i} \left(a^t_{ii} m_{ij} - a^t_{ij} m_{ij} - q^t_{ij} n_{ij}\right) \tag{5.8d}$$

$$\forall t \in T \wedge \forall i \in N:\ \Im(v^t_i) + \Im(d^t_i) - \Im(c^t_i) - \Im(s^t_i) = \sum_{j:j\sim i} \left(-a^t_{ii} n_{ij} + a^t_{ij} n_{ij} - q^t_{ij} m_{ij}\right) \tag{5.8e}$$

$$\forall t \in T \wedge \forall i \in N:\ \underline{V}^2_i \leqslant a^t_{ii} \leqslant \overline{V}^2_i \tag{5.8f}$$

$$\forall t \in T \wedge \forall i \in N:\ \Re(c^t_i)^2 + \Im(c^t_i)^2 \leqslant \overline{c}^2_i, \tag{5.8g}$$

$$\forall t \in T \wedge \forall i \in N:\ \Re(d^t_i)^2 + \Im(d^t_i)^2 \leqslant \overline{d}^2_i, \tag{5.8h}$$

$$\forall t \in T \wedge \forall i \to j:\ (a^t_{ij})^2 + (q^t_{ij})^2 + \left(\frac{a^t_{ii} - a^t_{jj}}{2}\right)^2 \leqslant \left(\frac{a^t_{ii} + a^t_{jj}}{2}\right)^2 \tag{5.8i}$$

The above presented formulation has a linear upper-level problem in the variable $\widehat{p}, \breve{p}$ and a lower-level problem in the variables $c, d, B, a,$ and q. In particular, all the constraints are linear, but the last one (5.8i) which is a SOCP constraint, and (5.8g)–(5.8h) which are convex quadratic, and therefore special cases of SOCP constraints. We refer the readers to Appendix B for two single-level reformulations of the bilevel problem with convex lower level (5.7)–(5.8): one is the strong duality reformulation and the other the KKT reformulation (see Section 1.2 of Chapter 1).

5.4 The network design ACOPF

In this section, we consider the power network design problem, explaining in more detail the characteristics of the network (using different notation as well). We include in our formulation parallel arcs, which occur whenever parallel cables are deployed on connections that must transport excessive amounts of power for a single cable. The h-th line ℓ_{ijh} joining two buses i and j is represented by a pair of anti-parallel arcs $\ell_{ijh} = \{(i,j,h),(j,i,h)\}$. We assume that L is partitioned in two sets L_0, L_1 with $|L_0| = |L_1|$: for each pair of antiparallel arcs, one is in L_0 and the other in L_1, according to the asymmetry of the branch admittance matrix $\mathbf{Y}_{ijh}$ matrix described below.

Ohm's law expresses the current I_{ijh} injected on a line ℓ_{ijh} in function of the voltages V_i, V_j at the endpoints i and j, and of the physical properties of the line. The fundamental difference with Ohm's law in DC is that AC yields an asymmetry. While in DC we have $I_{ijh} = -I_{jih}$, in AC we instead have:

$$\forall (i,j,h) \in L_0 \quad I_{ijh} = Y^{\mathsf{ff}}_{ijh} V_i + Y^{\mathsf{ft}}_{ijh} V_j \quad \wedge \quad I_{jih} = Y^{\mathsf{tf}}_{ijh} V_i + Y^{\mathsf{tt}}_{ijh} V_j. \tag{5.9}$$

The Y constants in the above equations are defined as follows [42, 43]:

$$\mathbf{Y}_{ijh} = \begin{pmatrix} Y^{\mathsf{ff}}_{ijh} & Y^{\mathsf{ft}}_{ijh} \\ Y^{\mathsf{tf}}_{ijh} & Y^{\mathsf{tt}}_{ijh} \end{pmatrix} = \begin{pmatrix} \left(\frac{1}{r_{ijh}+ix_{ijh}} + i\frac{\mathfrak{b}_{ijh}}{2}\right)/\tau^2_{ijh} & -\frac{1}{(r_{ijh}+ix_{ijh})\tau_{ijh}e^{-i\nu_{ijh}}} \\ -\frac{1}{(r_{ijh}+ix_{ijh})\tau_{ijh}e^{i\nu_{ijh}}} & \frac{1}{r_{ijh}+ix_{ijh}} + i\frac{\mathfrak{b}_{ijh}}{2} \end{pmatrix}, \tag{5.10}$$

where $r, x, \mathfrak{b}, \tau, \nu$ measure some physical properties of the line, and are given as part of the instance. The suffixes ff, ft, tf, tt to Y stand for "from-from", "from-to", "to-from", and "to-to": they are a reminder of the direction of the routed quantities w.r.t. the line ℓ_{ijh}.

We can now introduce sets, parameters, and decision variables of the ACOPF considered in this section of the chapter. We also recall elements already introduced in Section 5.3, for sake of completeness.

Sets N, L, and a set $\mathscr{G}$ of generators partitioned as $\{\mathscr{G}_i \mid i \in N\}$, where $\mathscr{G}_i$ contains the generators attached to bus b.

Parameters power demand (or *load*) $\tilde{S}_i$, *shunt admittance* A_i; voltage magnitude bounds $\underline{V}_i, \overline{V}_i$ at each bus $i \in N$; *admittance matrix* $\mathbf{Y}_{ijh}$; upper bound $\overline{S}_{ijh}$ to injected power magnitude; lower/upper bounds $\underline{\eta}_{ijh}, \overline{\eta}_{ijh}$ to phase difference at each line $(i,j,h) \in L$; cost coefficients C_{g2}, C_{g1}, C_{g0}; lower/upper bounds $\underline{\mathscr{S}}_g, \overline{\mathscr{S}}_g$ to power generated at $g \in \mathscr{G}$; a *reference bus* $r \in N$.

Variables voltage V_i at bus $i \in N$, injected current I_{ijh}, injected power S_{ijh} at each line $(i,j,h) \in L$, and generated power $\mathscr{S}_g$ at each generator $g \in \mathscr{G}$.

All variables range in $\mathbb{C}$. Among the parameters, the power magnitude, voltage magnitude, phase difference bounds, cost coefficients are in $\mathbb{R}$; r ranges in the bus set; the generated power bounds are in $\mathbb{C}$.

We present now objective function and constraints of what we call here the (S,I,V)-formulation of the ACOPF.

Objective function $\min \sum_{g \in \mathscr{G}} (C_{g2}(\Re(\mathscr{S}_g))^2 + C_{g1}\Re(\mathscr{S}_g) + C_{g0})$, which is quadratic and separable in generated power.

Constraints Bounds constraints are: on voltage magnitude $\underline{V}_i^2 \leqslant |V_i|^2 \leqslant \overline{V}_i^2$ for each $i \in N$; on power magnitude $|S_{ijh}|^2 \leqslant \overline{S}_{ijh}^2$ for each $(i,j,h) \in L$; on phase difference $\tan(\underline{\eta}_{ijh})\Re(V_i V_j^*) \leqslant \Im(V_i V_j^*) \leqslant \tan(\overline{\eta}_{ijh})\Re(V_i V_j^*)$ together with $\Re(V_i V_j^*) \geqslant 0$ for $(i,j,h) \in L_0$; on generated power $\underline{\mathscr{S}}_g \leqslant \mathscr{S}_g \leqslant \overline{\mathscr{S}}_g$ for each $g \in \mathscr{G}$. Moreover, we have $\Im(V_r) = 0$ and $\Re(V_r) \geqslant 0$ on the reference bus.

Functional constraints, with the new parameters and variables, are:

- Power flow equations:

$$\forall i \in N \qquad \sum_{(i,j,h) \in L} S_{ijh} + \tilde{S}_i = -A_i^* |V_i|^2 + \sum_{g \in \mathscr{G}_i} \mathscr{S}_g. \tag{5.11}$$

- The relationship between S, V, I (power balance):

$$\forall (i,j,h) \in L \quad S_{ijh} = V_i I_{ijh}^*. \tag{5.12}$$

- Ohm's law Eq. (5.9), which we write equivalently as:

$$\forall (i,j,h) \in L_0 \quad I_{ijh} = Y_{ijh}^{\mathsf{ff}} V_i + Y_{ijh}^{\mathsf{ft}} V_j \tag{5.13}$$

$$\forall (i,j,h) \in L_1 \quad I_{ijh} = Y_{jih}^{\mathsf{tf}} V_j + Y_{jih}^{\mathsf{tt}} V_i. \tag{5.14}$$

We now introduce a binary variable y_{ijh} for each (i,j,h) in L. We have $y_{ijh} = 1$ iff the corresponding line is active, and we must ensure that both antiparallel arcs are active/inactive at the same time by $y_{ijh} = y_{jih}$. We control activation/deactivation of a line by limiting the injected power magnitude bound:

$$\forall (i,j,h) \in L \quad |S_{ijh}|^2 \leqslant \overline{S}^2_{ijh} y_{ijh}. \tag{5.15}$$

In order to ensure that Eq. (5.15) does not impose constraints on V_i and V_j when the line (i,j,h) is not active, we introduce a new complex variable z_{ijh} in Eq. (5.12), such that:

$$\forall (i,j,h) \in L \quad S_{ijh} = V_i \, I_{ijh}{}^* + z_{ijh}, \tag{5.16}$$

and

$$\forall (i,j,h) \in L \quad |z_{ijh}|^2 \leqslant M^2_{ijh}(1 - y_{ijh}), \tag{5.17}$$

where M_{ijh} is a large enough constant. Note that Eqs. (5.15)–(5.17) do not cut the global optima of the ACOPF: it suffices to set $y_{ijh} = 1$ for each $(i,j,h) \in L$ to see this. Instead, we add an objective function $\min \sum\limits_{(i,j,h) \in L_0} y_{ijh}$. We can tackle this bi-objective MINLP problem by scalarization approaches. We could add a constraint $\sum\limits_{(i,j,h) \in L_0} y_{ijh} \leqslant \xi$ and letting ξ vary in $\{1, \ldots, m/2\}$. Here we consider another scalarization approach, so that the objective function becomes:

$$\min \sum_{g \in \mathscr{G}} (C_{g2}(\Re(\mathscr{S}_g))^2 + C_{g1}\Re(\mathscr{S}_g) + C_{g0}) + \rho \sum_{(i,j,h) \in L_0} y_{ijh}, \tag{5.18}$$

where $\rho > 0$ is a scalar weight which we set to 1 for testing purposes. We denote the network design ACOPF problem with binary variables on lines by ACOPF$_{\mathsf{L}}$.

5.4.1 ACOPF relaxations

The material in this section is motivated by the solution difficulty posed by the nonconvex MINLP formulation of the ACOPF$_{\mathsf{L}}$. First of all, we propose some valid relaxation for network design ACOPF problem.

The decision variables I for current can be eliminated from the (S, I, V)-formulation by replacing them in Eq. (5.12) with their expressions in Eqs. (5.13)-(5.14), as done in Section 5.3. This yields the (S, V)-formulation, which is still a nonconvex NLP. In turn, using Eqs. (5.15)-(5.18), this NLP yields a nonconvex MINLP formulation for the ACOPF$_{\mathsf{L}}$.

(S, V, W)-relaxation

Following the same approach used in Section 5.3.1, we can linearize the only nonlinear terms appearing in the nonconvex constraints of the ACOPF (S, V)-formulation, i.e., the products $V_i \, V_j{}^*$ for some $i, j \in N$. Every such product term can be linearized, i.e., replaced by a new (complex) variable W_{ij} for $i, j \in N$ (we do not include the corresponding defining constraint $W_{ij} = W_i W_j{}^*$). Let us call this the (S, V, W)-relaxation. This turns out to be a SOCP. The quadratic terms are: $\mathscr{S}_g^2$ in the minimizing objective and $|S_{ijh}|^2$ in the LHS of the power magnitude bound constraints.

(S, V, W)-SDP

Note that, as stated in Section 5.3.1 the (S, V, W)-relaxation is an exact reformulation if we enforce $W = VV^{\mathsf{H}}$, where the apex stands for "Hermitian transpose", i.e., the transpose of the componentwise complex conjugate. Accordingly, since W is a PSD rank-one matrix, we get a stronger relaxation w.r.t. the (S, V, W)-relaxation presented in Sec. 5.4.1, if we replace $W = VV^{\mathsf{H}}$ by $W \succeq 0$, which yields a complex SDP relaxation called (S, V, W)-SDP.

(S, V, W)-$\frac{1}{2}$DDP

Given the scarcity of off-the-shelf mixed-integer SDP solvers, we consider a DDP approximation of the PSD cone [13]: since every Diagonally Dominant (DD) matrix is also PSD [102], the constraint "W is DD" yields an inner approximation (i.e., a restriction) of the complex SDP.

Writing the DDP constraints corresponding to $W \succeq 0$ requires splitting W into real and imaginary parts, which yields $\overline{W} = \begin{pmatrix} W^{\mathsf{rr}} & W^{\mathsf{rc}} \\ W^{\mathsf{cr}} & W^{\mathsf{cc}} \end{pmatrix} \in \mathbb{R}^{2n \times 2n}$, where $W^{\mathsf{rr}} = (\Re(W_{ij}))$, $W^{\mathsf{cc}} = (\Im(W_{ij}))$, W^{rc} linearizes the matrix $(\Re(V_i)\Im(V_j))$, and W^{cr} linearizes the matrix $(\Im(V_i)\Re(V_j))$. We remark that $W^{\mathsf{rr}}, W^{\mathsf{cc}}$ are symmetric matrices, while $W^{\mathsf{rc}}, W^{\mathsf{cr}}$ are not; on the other hand, $W^{\mathsf{rc}}_{ij} = W^{\mathsf{cr}}_{ji}$ for each $i, j \in N$.

Now the DDP inner approximation of $\overline{W} \succeq 0$ states that any diagonal component of $\overline{W}$ is greater than or equal to the sum of the absolute values of the components in the same row. This corresponds to:

$$\forall i \in N \quad W^{\mathsf{rr}}_{ii} \geqslant \sum_{\substack{a \in N \\ a \neq b}} T^{\mathsf{rr}}_{ij} + \sum_{a \in N} T^{\mathsf{rc}}_{ij} \tag{5.19}$$

$$\forall i \in N \quad W^{\mathsf{cc}}_{ii} \geqslant \sum_{\substack{a \in N \\ a \neq b}} T^{\mathsf{cc}}_{ij} + \sum_{a \in N} T^{\mathsf{cr}}_{ij}, \tag{5.20}$$

where $\overline{T} = \begin{pmatrix} T^{\mathsf{rr}} & T^{\mathsf{rc}} \\ T^{\mathsf{cr}} & T^{\mathsf{cc}} \end{pmatrix}$ is a real variable matrix such that $-\overline{T} \leqslant \overline{W} \leqslant \overline{T}$ [13].

The issue with inner DDP approximations is that they may be infeasible even if the corresponding SDP is feasible. Experimentally, this was verified to be the case in every ACOPF instance we tested. This issue can be addressed algorithmically

[13], but this would require solving a sequence of DDPs, which would in turn take excessive time. Instead, we chose to only impose Eq. (5.19), which yielded feasible "half-DDP" relaxations (which we refer to as (S, V, W)-$\frac{1}{2}$DDP relaxation) of the tested ACOPF instances. Note that we do not have a general feasibility proof for $\frac{1}{2}$DDP relaxations. So far, we have not found any counterexamples yet, either.

`Jabr` relaxation

Another SOCP relaxation of the ACOPF, called "Jabr relaxation", was proposed in [115]. It can be constructed from the (S, V)-formulation as follows:

1. transform cartesian coordinates $\Re(V), \Im(V)$ to polar coordinates v, θ by replacing $\Re(V) = v\cos\theta$ and $\Im(V) = v\sin\theta$: this will result with nonlinear terms in $v_i v_j \cos(\theta_i - \theta_j)$ and $v_i v_j \sin(\theta_i - \theta_j)$;

2. define an index set $R = \{(i,i) \mid i \in N\} \cup \{(i,j) \mid (i,j,1) \in L\}$;

3. linearize (replace) the nonlinear terms with new variables $c_{ij} = v_i v_j \cos(\theta_i - \theta_j)$ and $s_{ij} = v_i v_j \sin(\theta_i - \theta_j)$ for all $(i,j) \in R$: this also yields $c_{ij} = c_{ji}$, $s_{ij} = -s_{ji}$, $c_{ij}^2 + s_{ij}^2 = v_i^2 v_j^2$ $(\star)$ for all $(i,j,1) \in L_0$, as well as $s_{ii} = 0$ and $c_{ii} = v_i^2$ $(\dagger)$ for each $i \in N$;

4. replace v_i^2, v_j^2 in $(\star)$ with c_{ii}, c_{jj} by means of $(\dagger)$, and relax $(\star)$ to a convex (conic) constraint $c_{ij}^2 + s_{ij}^2 \leqslant c_{ii}c_{jj}$;

5. replace $|V_i|^2$ in the voltage magnitude bounds with c_{ii};

6. remark that $V_i V_j^* = c_{ij} + i s_{ij}$, and infer the phase difference bounds as $c_{ij} \geqslant 0$ and $\tan(\underline{\eta}_{ijh})c_{ij} \leqslant s_{ij} \leqslant \tan(\overline{\eta}_{ijh})c_{ij}$ for each $(i,j,h) \in L_0$;

7. the injected power variables S_{ijh} satisfy the linear equations:

$$\begin{array}{llcl}
\forall (i,j,h) \in L_0 & \Re(S_{ijh}) & = & \Re(Y^{\mathsf{ff}}_{ijh})c_{ii} + \Re(Y^{\mathsf{ft}}_{ijh})c_{ij} + \Im(Y^{\mathsf{ft}}_{ijh})s_{ij} \\
\forall (i,j,h) \in L_0 & \Im(S_{ijh}) & = & -\Im(Y^{\mathsf{ff}}_{ijh})c_{ii} + \Re(Y^{\mathsf{ft}}_{ijh})s_{ij} - \Im(Y^{\mathsf{ft}}_{ijh})c_{ij} \\
\forall (i,j,h) \in L_1 & \Re(S_{ijh}) & = & \Re(Y^{\mathsf{tt}}_{jih})c_{ii} + \Re(Y^{\mathsf{tf}}_{jih})c_{ij} + \Im(Y^{\mathsf{tf}}_{jih})s_{ij} \\
\forall (i,j,h) \in L_1 & \Im(S_{ijh}) & = & -\Im(Y^{\mathsf{tt}}_{jih})c_{ii} + \Re(Y^{\mathsf{tf}}_{jih})s_{ij} - \Im(Y^{\mathsf{tf}}_{jih})c_{ij}.
\end{array}$$

5.4.2 ACOPF$_{\mathsf{L}}$ relaxations

We derive ACOPF$_{\mathsf{L}}$ relaxations from the (S, V, W)-relaxation, the (S, V, W)-$\frac{1}{2}$DDP and Jabr relaxations of the ACOPF, by employing the binary variables y as in Section 5.4, i.e., by imposing Eqs. (5.15)-(5.17) and minimizing Eq. (5.18). A few preliminary results showed that the active lines do not form a connected set at the optimum. In order to enforce connectivity, we therefore also added a set of multicommodity flow constraints on added variables f^{ij}_{deh}, defined for each distinct pair $i, j \in N$ and line $(d, e, h) \in L$:

$$\begin{array}{rrcl} \forall i < j \in N & \sum_{(i,d,h)\in L} f^{ij}_{idh} - \sum_{(d,i,h)\in L} f^{ij}_{dih} & = & 1 \\ \forall i < j \in N & \sum_{(d,j,h)\in L} f^{ij}_{djh} - \sum_{(j,d,h)\in L} f^{ij}_{jdh} & = & 1 \\ \forall i < j \in N, d \in N \smallsetminus \{i,j\} & \sum_{(e,d,h)\in L} f^{ij}_{edh} - \sum_{(d,e,h)\in L} f^{ij}_{deh} & = & 0, \end{array}$$

as well as the linking constraints: $\forall i < j \in N, (d, e, h) \in L \quad f^{ij}_{deh} \leqslant y_{deh}$.

In Table 5.1, we refer to the ACOPF$_{\mathsf{L}}$relaxations from (S, V, W)-$\frac{1}{2}$DDP, and Jabr as "`ddp`", and "`Jabr`" respectively.

5.4.3 Computational experiments

The standard reference testbed for computational assessments in ACOPF is the PGLib library [24], which also includes "case files" from MatPower [212]. We compare performances of the two convex MINLP relaxations of the ACOPF$_{\mathsf{L}}$(`ddp` and `Jabr`) on the small case instances `case`i for $i \in \{5, 9, 14, 18, 22, 24, 30\}$. Our implementation is carried out in AMPL [95]. We solve both formulations, which are of the Mixed-Integer SOCP sort, with CPLEX 12.9 [113], which is given 300s as maximum CPU time. Only instance "`case5`" is solved using Baron, because AMPL failed to successfully pass it to CPLEX.

The results in Table 5.1 are obtained on a 2.53GHz Intel(R) Xeon(R) CPU with 49.4 GB RAM. They show that 300s are only sufficient to obtain meaningful results for small instances.

name	lines	known	opt1	opt2	act1	act2	stat1	stat2	cpu1	cpu2
`case5`	6	17551.89	0.00	**15169.08**	4	4	solved	limit	**2.03**	300
`case9`	9	5296.67	2244.81	**5296.67**	9	9	solved	solved	**3.58**	3.98
`case14`	20	8081.52	0.00	**1786.93**	13	14	limit	limit	300	300
`case18`	17	11.85	-0.00	**11.85**	17	17	solved	solved	**0.36**	0.49
`case22`	21	0.068	0.00	**0.068**	21	21	solved	solved	**0.56**	6.01
`case24`	38	63352.20	47320.2	**63345.20**	38	38	limit	limit	300	300
`case30`	41	576.89	0.00	**568.86**	29	30	limit	limit	300	300

Table 5.1: Numerical results limited to 300s using a single CPU processor. We report instance name, number of lines, known optimal value; then, for each relaxation type $(1, 2) =$(`ddp`,`Jabr`), we report obtained optimal value, number of active lines, solver status, CPU time. Best results are in boldface.

An encouraging feature of the results in Table 5.1 is that the slacker `ddp` relaxation takes less time to solve than `Jabr`, provides a worst bound, but still identifies a valid connectivity for active lines for all the tested instances. In Fig. 5.1, e.g., we report two solutions found by solving the `ddp` relaxation, which appear to be the same found by `Jabr` relaxation, as well as the two different solutions obtained by solving the same instance "`case30`" with `ddp`, and `Jabr`.

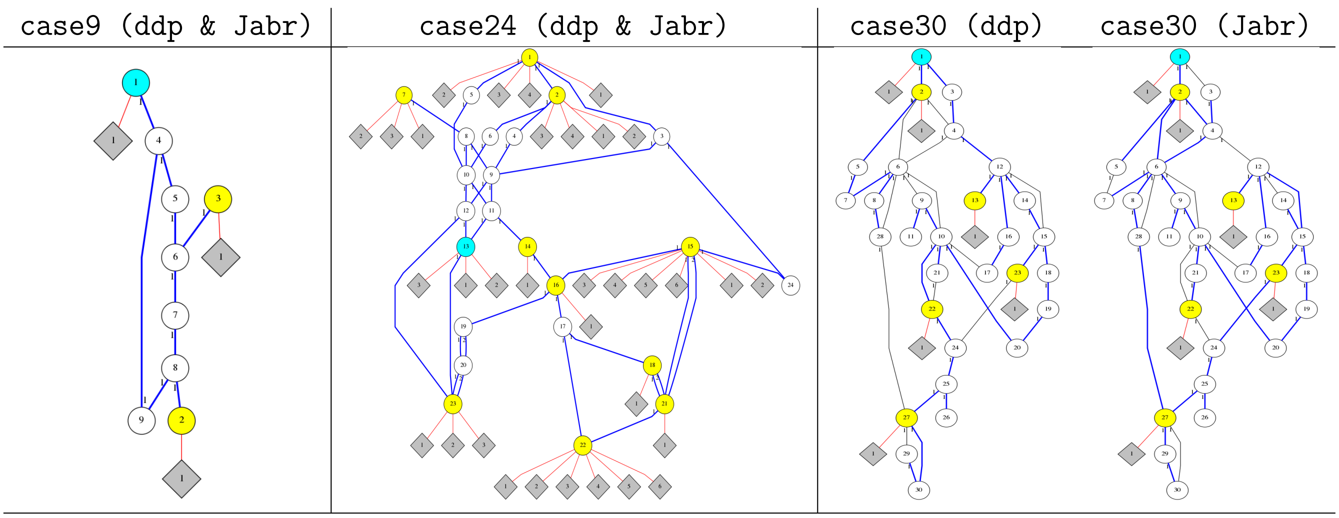

Figure 5.1: The solutions of three instances from results in Table 5.1. Buses in circles, generators in parallelograms (buses with generators are colored, reference bus is colored differently); active lines are thick and colored.

In Table 5.2 we report results from the (S, V, W)-relaxation of ACOPF$_{\mathsf{L}}$on slightly larger instances, solved using CPLEX limited to 7200s. When solutions are found atypically quickly (e.g., `case69`, `case85`), it is because the networks have no cycles. The solution found for `case 30` is shown in Figure 5.2.

name	lines	known	opt	act	stat	cpu
`case24`	38	63352.20	47320.20	38	limit	7200
`case30`	41	576.89	0.00	29	limit	7200
`ieee30`	41	9974.99	0.00	29	limit	7200
`case39`	46	41864.17	27417.26	46	limit	7200
`case69`	68	0.39	0.00	68	solved	9.74
`case85`	84	0.00	0.00	84	solved	21.50

Table 5.2: Computational results on the (S, V, W)-relaxation limited to 7200s.

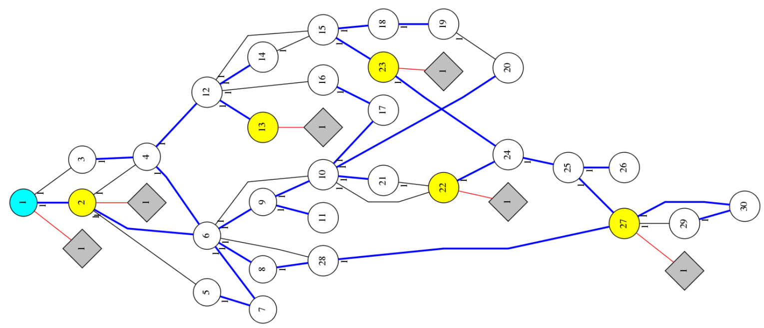

Figure 5.2: Nontrivial solution for `case30`.

5.5 Conclusion

Firstly, in this chapter we used bilevel programming to model the interplay among a retailer and a set of prosumers, i.e., consumers who can produce (thanks to rooftop PV panels), store (in a battery), or buy (from the retailer or the other prosumers) power. Such prosumers are connected to an alternating current power network. The retailer is assumed to operate in the upper level and the set of prosumers in the lower level. Since the ACOPF constraints considered at the lower level are nonconvex, we proposed a relaxation of the bilevel problem obtained by relaxing the lower-level problem using SOCP.

Secondly, we focused on the network design ACOPF problem, i.e., the AC power network design with respect to line activity. For such single-level problem we proposed several convex relaxations, which are compared using some benchmarking instances.

Conclusion

A bilevel programming problem is defined as an optimization program containing another optimization program in the constraints. Their particular structure allows the formulation of a huge number of real-world problems that involve hierarchical decision making processes, i.e., processes where the outcome of any choice made by an upper-level authority (leader) is influenced by the reaction of lower-level subjects (followers). Despite the fact that most real-world choices may be argued to be bilevel in the sense that they influence systems with autonomous and competing agents, until a few decades ago, still few studies had used this promising modelization tool to formulate practical problems. The reason may be found in the inherent complexity of bilevel problems, which turn out to be particularly difficult to handle mathematically. Even their simplest version, with only linear functions and continuous variables, is proved to be strongly NP-hard, and merely evaluating a solution for optimality is a NP-hard task. More recently, a growing number mathematical programming community members has shown interest in bilevel optimization, proposing direct solution procedures, which may be a valid alternative to the usual approach of transforming bilevel problems into simpler single-level optimization programs, which are solved eventually giving up on finding an optimal solution. This dissertation may be seen as a step forward in this direction.

We started with two introductory chapters. Central definitions, the differences between optimistic and pessimistic bilevel formulations, and an overview on reformulation approaches and algorithms that have been proposed for bilevel optimization have been presented in Chapter 1. We moved on to discuss the computational complexity of bilevel programming problems in Chapter 2. We focused on (mixed-integer) linear bilevel problems, and on special cases which are surprisingly polynomially solvable. In Chapter 3, we presented the results contained in the submitted paper [7]. A particular class of bilevel problem with quadratic lower level has been studied, which can be obtained reformulating linear SIP problems with an infinite number of quadratically parameterized constraints. We proposed two approaches to solve such bilevel problems. Firstly, a convex single-level formulation with a finite number of constraints has been obtained by dualizing the lower level, using SDP optimization. Solving it using state-of-the-art global solvers provides a feasible solution, which is optimal if the quadratic lower-level problem is convex. Secondly, a cutting plane algorithm has been proposed to solve directly the bilevel formulation. Such algorithm has been proved to be convergent, and a new rate of convergence has been given when the upper-level objective function is strongly convex, under a strict feasibility assumption. This convergence rate is directly related to the iteration index, which is something new with respect to what is usually proved in SIP literature. Our computational experiments on small and medium-scale instances of two interesting applications demonstrated that the single-level formulation approach outperforms, in terms of solution time, the cutting plane algorithm for instances with a convex lower level. As for the cases with a nonconvex lower level, the two techniques are complementary: the dual approach was faster, but only gives a feasible solution, whereas the cutting plane approach was slower, but solved the bilevel problem to optimality.

The second part of the dissertation was dedicated to two applications. In Chapter 4 we studied the aircraft conflict resolution problem via two strategies. This problem essentially consists in enforcing a minimum distance between flying aircraft, which naturally results in a SIP model. When considering the speed regulation strategy, we reformulated such SIP formulation using first, polynomial programming, and second, bilevel programming. Then, we considered the heading angle change strategy for aircraft flying at the same altitude level, directly formulating the conflict resolution problem in two dimensions as a bilevel program. Both bilevel formulations, which fit in the setting of the formulation studied in Chapter 3, have been reformulated into single-level problems using KKT conditions and strong duality of the convex lower level. Moreover, a tailored cutting plane algorithm has been proposed, which outperforms in terms of efficiency the other approaches presented. The second practical application studied in Chapter 5 is the ACOPF problem. When considering the interplay among a retailer and several prosumers (consumers who can also produce, store and sell power) operating in an ACOPF network, we obtain a bilevel programming formulation. Since the ACOPF constraints are nonconvex, we proposed a relaxation of the lower level, using SOCP. When dealing with the AC network design problem, the ACOPF formulation must include some binary variables, becoming a nonconvex MINLP problem in complex numbers, which we relaxed in different ways. The proposed relaxation have been tested and the results turned out to be comparable with the ones obtained using other well-known relaxations.

Overall, we have made contributions in bilevel optimization worthy of attention. We proposed two approaches to solve particular bilevel problems with a possibly nonconvex lower level. We studied two challenging applications: the conflict resolution problem and the ACOPF. Nonetheless, there remain important open questions and challenges. A possible extension of the content of Chapter 3 could be implementing a cutting plane algorithm with the lower-level problem solved with an "on-demand" accuracy at each iteration. Regarding the *dual approach*, the sparse structure of the lower-level problem would be worth exploiting with the celebrated cliques decomposition technique. A direct extention of both proposed approaches would be an algorithm combining them, which may solve at each iteration a relaxation of the restriction obtained via the dual approach. Bigger instances should be tested, in particular for the first considered application, which may be relevant also in a machine learning context. As far as it concerns the aircraft conflict resolution problem discussed in Chapter 4, some limitations related to the simplification assumptions (such as rectilinear trajectories, uniform motion, maneuvers performed at the beginning of the considered time horizon) still remain to be addressed. Some more recent emerging applications could benefit from advances in aircraft conflict resolution optimization, such as autonomous vehicles on the ground (see, e.g., [134]), or new vehicles in urban air mobility (see, e.g., [156]). In this context, the performances of the proposed methods should be evaluated with less strict bounds on admissible speed modifications w.r.t. the ones proposed within ERASMUS project [66]. For the bilevel formulation with ACOPF at the lower level presented in Chapter 5, we should test at least medium-scale instances to prove its applicability to real situations. In order to represent more realistic scenarios, we could assume feed-in tariffs different from the prices, other renewable energy sources besides the solar panels could be considered, as well as the uncertainty related to such uncontrollable energy sources. As regards the network design ACOPF, the proposed reformulations could be tested on larger instances. Finally, we can also look at the bigger picture. In this thesis, we mostly discussed problems involving only continuous variables. Considering mixed-integer bilevel problem is definitely challenging and is a very relevant future research direction.

We conclude that, although there is still a lot to be learned in bilevel optimization, the theoretical results and solution approaches presented in this dissertation can be helpful in future studies in their relative fields.

Bibliography

[1] M. Cerulli, C. D'Ambrosio, and L. Liberti. Flying safely by bilevel programming. In M. Paolucci, A. Sciomachen, and P. Uberti, editors, *Advances in Optimization and Decision Science for Society, Services and Enterprises: ODS, Genoa, Italy, September 4-7, 2019*, pages 197–206. Springer International Publishing, Cham, 2019. doi:10.1007/978-3-030-34960-8_18.

[2] M. Cerulli, C. D'Ambrosio, L. Liberti, and M. Pelegrín. Detecting and solving aircraft conflicts using bilevel programming. *Journal of Global Optimization*, 81(2):529–557, 02 2021. doi:10.1007/s10898-021-00997-1.

[3] M. Cerulli and L. Liberti. Polynomial programming prevents aircraft (and other) conflicts. *Operations Research Letters*, 49(4):447–451, 2021. doi:10.1016/j.orl.2021.05.001.

[4] M. Cerulli. Instances of aircraft deconfliction problem via speed regulation in 3 dimensions. https://github.com/MartinaCerulli/SRADP, 2020.

[5] M. Pelegrín and M. Cerulli. Aircraft conflict resolution: A benchmark generator. Submitted paper, 2021.

[6] D. Bienstock, M. Cerulli, M. Escobar, and L. Liberti. Power network design with line activity. Submitted paper available at: https://hal.archives-ouvertes.fr/hal-03263306v1, 2021.

[7] M. Cerulli, Oustry A., C. D'Ambrosio, and L. Liberti. Solving a class of bilevel programs with quadratic lower level. Submitted paper available at: https://hal.archives-ouvertes.fr/hal-03339887v1, 2021.

[8] M. Cerulli, M. Pelegrín, S. Cafieri, C. D'Ambrosio, and D. Rey. Aircraft conflict resolution. Submitted paper, 2021.

[9] M. Cerulli, D. Delle Donne, M. Escobar, and A. Oustry. Optimal Power Flow. Submitted paper, 2021.

[10] S. Abbott. *Understanding Analysis*. Undergraduate Texts in Mathematics. Springer New York, 2016. doi:10.1007/978-1-4939-2712-8.

[11] M. A. Abido. Optimal power flow using particle swarm optimization. *International Journal of Electrical Power & Energy Systems*, 24(7):563–571, 2002. doi:10.1016/S0142-0615(01)00067-9.

[12] R.N. Adams and M. A. Laughton. Optimal planning of power networks using mixed-integer programming. Part 1: Static and time-phased network synthesis. Proceedings of IEE, 121(2):139–147, 1974. doi:10.1049/piee.1974.0024.

[13] A. A. Ahmadi and A. Majumdar. DSOS and SDSOS optimization: More tractable alternatives to sum of squares and semidefinite optimization. *SIAM Journal on Applied Algebra and Geometry*, 3(2):193–230, 2019. doi:10.1137/18M118935X.

[14] E. Aiyoshi and K. Shimizu. Hierarchical decentralized systems and its new solution by a barrier method. *IEEE Transactions on Systems, Man and Cybernetics*, 11(6):444–449, 1981. doi:10.1109/tsmc.1981.4308712.

[15] E. Aiyoshi and K. Shimizu. A solution method for the static constrained stackelberg problem via penalty method. *IEEE Transactions on Automatic Control*, 29(12):1111–1114, 1984. doi:10.1109/TAC.1984.1103455.

[16] F. A. Al-Khayyal, R. Horst, and P. M. Pardalos. Global optimization of concave functions subject to quadratic constraints: an application in nonlinear bilevel programming. *Annals of Operations Research*, 34(1):125–147, 1992. doi:10.1007/BF02098176.

[17] N. Alguacil, A. Delgadillo, and J. Arroyo. A trilevel programming approach for electric grid defense planning. *Computers & Operations Research*, 41:282–290, 2014. doi:10.1016/j.cor.2013.06.009.

[18] A. Alonso-Ayuso, L. Escudero, and F. J. Martín-Campo. Collision avoidance in air traffic management: A mixed-integer linear optimization approach. *Intelligent Transportation Systems, IEEE Transactions on*, 12:47–57, 04 2011. doi:10.1109/TITS.2010.2061971.

[19] A. Alonso-Ayuso, L. Escudero, and F. J. Martín-Campo. Exact and approximate solving of the aircraft collision resolution problem via turn changes. *Transportation Science*, 50(1):263–274, 2014. doi:10.1287/trsc.2014.0557.

[20] A. Alonso-Ayuso, L. Escudero, and F. J. Martín-Campo. An exact multi-objective mixed integer nonlinear optimization approach for aircraft conflict resolution. *Top*, 24(2):381–408, 2016. doi:10.1007/s11750-015-0402-z.

[21] O. Alsac and B. Stott. Optimal Load Flow with Steady-State Security. *IEEE Transactions on Power Apparatus and Systems*, PAS-93(3):745–751, 1974. doi:10.1109/TPAS.1974.293972.

[22] C. Audet, P. Hansen, B. Jaumard, and G. Savard. Links between linear bilevel and mixed 0–1 programming problems. *Journal of Optimization Theory and Applications*, 93:273—300, 1997. doi:10.1023/A:1022645805569.

[23] S. Avraamidou and E. N. Pistikopoulos. A multi-parametric optimization approach for bilevel mixed-integer linear and quadratic programming problems. *Computers & Chemical Engineering*, 125:98–113, 2019. doi:10.1016/j.compchemeng.2019.01.021.

[24] S. Babaeinejadsarookolaee, A. Birchfield, R. D. Christie, C. Coffrin, C. DeMarco, R. Diao, M. Ferris, S. Fliscounakis, S. Greene, R. Huang, et al. The power grid library for benchmarking AC optimal power flow algorithms. Technical report, arXiv:1908.02788, 2019.

[25] J. F. Bard. An algorithm for solving the general bilevel programming problem. *Mathematics of Operations Research*, 8(2):260–272, 1983. doi:10.1287/moor.8.2.260.

[26] J. F. Bard. Convex two-level optimization. *Mathematical programming*, 40(1):15–27, 1988. doi:10.1007/BF01580720.

[27] J. F. Bard. Some properties of the bilevel programming problem. *Journal of Optimization Theory and Applications volume*, 68:371–378, 1991. doi:10.1007/BF00941574.

[28] J. F. Bard and J. E. Falk. An explicit solution to the multi-level programming problem. *Computers & Operations Research*, 9(1):77–100, 1982. doi:10.1016/0305-0548(82)90007-7.

[29] J. F. Bard and J. Moore. A branch and bound algorithm for the bilevel programming problem. *Siam Journal on Scientific and Statistical Computing*, 11:281–292, 1990. doi:10.1137/0911017.

[30] J. F. Bard and J. T. Moore. An algorithm for the discrete bilevel programming problem. *Naval Research Logistics (NRL)*, 39(3):419–435, 1992. doi:10.1002/1520-6750(199204).

[31] J.F. Bard. Coordination of a multidivisional organization through two levels of management. *Omega*, 11(5):457–468, 1983. doi:10.1016/0305-0483(83)90038-5.

[32] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, 24(4):597–634, 2009. doi:10.1080/10556780903087124.

[33] O. Ben-Ayed. Bilevel linear programming. *Computers & Operations Research*, 20(5):485–501, 1993. doi:10.1016/0305-0548(93)90013-9.

[34] O. Ben-Ayed and C. E. Blair. Computational difficulties of bilevel linear programming. *Operations Research*, 38(3):556–560, 1990. doi:10.1287/opre.38.3.556.

[35] O. Ben-Ayed, D.E. Boyce, and C. E. Blair. A general bilevel linear programming formulation of the network design problem. *Transportation Research Part B: Methodological*, 22(4):311–318, 1988. doi:10.1016/0191-2615(88)90006-9.

[36] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust optimization*. Princeton university press, 2009.

[37] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization*. Society for Industrial and Applied Mathematics, 2001. doi:10.1137/1.9780898718829.

[38] B. Betró. An accelerated central cutting plane algorithm for linear semi-infinite programming. *Mathematical Programming*, 101(3):479–495, 2004. doi:10.1007/s10107-003-0492-5.

[39] Z. Bi, P. Calamai, and A. Conn. Technical report 167-o-310789: An exact penalty function approach for the linear bilevel programming problem. Technical report, Department of Systems Design Engineering, University of Waterloo, Canada, 1989.

[40] Z. Bi, P. Calamai, and A. Conn. Technical report 180-o-170591: An exact penalty function approach for the nonlinear bilevel programming problem. Technical report, Department of Systems Design Engineering, University of Waterloo, Canada, 1989.

[41] W. F. Bialas and M. H. Karwan. Two-level linear programming. *Management Science*, 30(8):1004–1020, 1984. doi:10.1287/mnsc.30.8.1004.

[42] D. Bienstock. Electrical Transmission System Cascades and Vulnerability: an Operations Research Viewpoint. Number 22 in MOS-SIAM Optimization. SIAM, Philadelphia, 2016. doi:10.1137/1.9781611974164.

[43] D. Bienstock, M. Escobar, C. Gentile, and L. Liberti. Mathematical programming formulations for the alternating current optimal power flow problem. 4OR, 18(3):249–292, 2020. doi:10.1007/s10288-020-00455-w.

[44] K.D. Bilimoria. A geometric optimization approach to aircraft conflict resolution. In *18th Applied aerodynamics conference*, pages 42–65, 2000. doi:10.2514/6.2000-4265.

[45] S. Bolusani and T. K. Ralphs. A framework for generalized benders' decomposition and its application to multilevel optimization. *arXiv preprint arXiv:2104.06496*, 2021.

[46] J. Bracken and J. T. McGill. Mathematical programs with optimization problems in the constraints. *Operations Research*, 21(1):37–44, 1973. doi:10.1287/opre.21.1.37.

[47] R. C. Burchett, H. H. Happ, and D. R. Vierath. Quadratically Convergent Optimal Power Flow. *IEEE Transactions on Power Apparatus and Systems*, PAS-103(11):3267–3275, 1984. doi:10.1109/TPAS.1984.318568.

[48] H. C. Bylling, S. A. Gabriel, and T. K. Boomsma. A parametric programming approach to bilevel optimisation with lower-level variables in the upper level. *Journal of the Operational Research Society*, 71(5):846–865, 2020. doi:10.1080/01605682.2019.1590132.

[49] S. Cafieri and N. Durand. Aircraft deconfliction with speed regulation: new models from mixed-integer optimization. *Journal of Global Optimization*, 58(4):613–629, 2014. doi:10.1007/s10898-013-0070-1.

[50] S. Cafieri and R. Omheni. Mixed-integer nonlinear programming for aircraft conflict avoidance by sequentially applying velocity and heading angle changes. *European Journal of Operational Research*, 1(260):283–290, 2017. doi:10.1016/j.ejor.2016.12.010.

[51] S. Cafieri and D. Rey. Maximizing the number of conflict-free aircraft using mixed-integer nonlinear programming. *Computers and Operations Research*, 80:147–158, 2017. doi:10.1016/j.cor.2016.12.002.

[52] W. Candler and R. Norton. Multi-level programming and development policy. Technical report, The World Bank Development Research Center, Washington D.C., 1977.

[53] W. Candler and R. Townsley. A linear two-level programming problem. *Computers & Operations Research*, 9(1):59–76, 1982. doi:10.1016/0305-0548(82)90006-5.

[54] A. Caprara, M. Carvalho, A. Lodi, and G. J. Woeginger. A study on the computational complexity of the bilevel knapsack problem. *SIAM Journal on Optimization*, 24(2):823–838, 2014. doi:10.1137/130906593.

[55] M. C. Caramanis, R. E. Bohn, and F. C. Schweppe. Optimal spot pricing: Practice and theory. *IEEE Transactions on Power Apparatus and Systems*, PAS-101(9):3234–3245, 1982. doi:10.1109/TPAS.1982.317507.

[56] L. M. Case. *An l_1 penalty function approach to the nonlinear bilevel programming problem*. PhD thesis, University of Waterloo, 1997.

[57] A. Castillo and D. F. Gayme. Profit maximizing storage allocation in power grids. In *52nd IEEE Conference on Decision and Control*, pages 429–435, 2013. doi:10.1109/CDC.2013.6759919.

[58] A. Castillo and R. P. O'Neill. Survey of approaches to solving the ACOPF (OPF paper 4). Technical report, US FERC, 2013.

[59] Y. Chen, M. Florian, and S. Wu. Technical report crt-866: A descent dual approach for linear bilevel program. Technical Report 866, Centre De Recherche Sur Les Transports Publication, 1992.

[60] Y. Chen, W. Wei, F. Liu, E. E. Sauma, and S. Mei. Energy trading and market equilibrium in integrated heat-power distribution systems. *IEEE Transactions on Smart Grid*, 10(4):4080–4094, 2019. doi:10.1109/TSG.2018.2849227.

[61] V. Chvátal. On the computational complexity of finding a kernel. *Report CRM-300, Centre de Recherches Mathématiques, Université de Montréal*, 592, 1973. URL: http://users.encs.concordia.ca/~chvatal/kernel.html.

[62] P.A. Clark. Bilevel programming for steady-state chemical process design—II. performance study for nondegenerate problems. *Computers & Chemical Engineering*, 14(1):99–109, 1990. doi:10.1016/0098-1354(90)87008-D.

[63] P.A. Clark and A.W. Westerberg. Bilevel programming for steady-state chemical process design—I. Fundamentals and algorithms. *Computers & Chemical Engineering*, 14(1):87–97, 1990. doi:10.1016/0098-1354(90)87007-C.

[64] B. Colson, P. Marcotte, and G. Savard. A trust-region method for nonlinear bilevel programming: algorithm and computational experience. *Computational Optimization and Applications*, 30(3):211–227, 2005. doi:10.1007/s10589-005-4612-4.

[65] P. Colson, B.and Marcotte and G. Savard. An overview of bilevel optimization. *Annals of Operations Research*, 153:235–256, 2007. doi:10.1007/s10479-007-0176-2.

[66] ERASMUS Consortium. Technical report proj. tren/06/fp6ae/s07, deliv. d4.6, en route air traffic soft management ultimate system. Technical report, European Commission, 2009. An optional note.

[67] I. D. Coope and G. A. Watson. A projected lagrangian algorithm for semi-infinite programming. *Mathematical Programming*, 32(3):337–356, 1985. doi:10.1007/BF01582053.

[68] T. Dan, A. Lodi, and P. Marcotte. Joint location and pricing within a user-optimized environment. *EURO Journal on Computational Optimization*, 8(1):61–84, 2020. doi:10.1007/s13675-019-00120-w.

[69] T. Dan and P. Marcotte. Competitive facility location with selfish users and queues. *Operations Research*, 67(2):479–497, 2019. doi:10.1287/opre.2018.1781.

[70] S. Dempe. A simple algorithm for the-linear bilevel programming problem. *Optimization*, 18(3):373–385, 1987. doi:10.1080/02331938708843247.

[71] S. Dempe. Discrete bilevel optimization problems. Technical report, Citeseerx, 2001.

[72] S. Dempe. *Foundations of Bi-Level Programming*. Nonconvex Optimization and Its Applications. Springer US, 1 edition, 2002. doi:10.1007/b101970.

[73] S. Dempe. Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints. *Optimization*, 52(3):333–359, 2003. doi:10.1080/0233193031000149894.

[74] S. Dempe. Bilevel programming: A survey. Technical report, Dekan der Fak. für Mathematik und Informatik, 2003.

[75] S. Dempe. *Bilevel Optimization: Reformulation and First Optimality Conditions*, pages 1–20. Springer, Singapore, 01 2017. doi:10.1007/978-981-10-4774-9_1.

[76] S. Dempe, J. Dutta, and B. S. Mordukhovich. New necessary optimality conditions in optimistic bilevel programming. *Optimization*, 56(5-6):577–604, 2007. doi:10.1080/02331930701617551.

[77] S. Dempe, V. Kalashnikov, G. A. Pérez-Valdés, and N. Kalashnykova. *Bilevel Programming Problems: Theory, Algorithms and Applications to Energy Networks*. Springer, Berlin, 2015. doi:10.1007/978-3-662-45827-3.

[78] S. Dempe and K. Richter. Bilevel programming with knapsack constraints. *CEJOR. Central European Journal of Operations Research*, 8:93–107, 2000.

[79] Stephan Dempe. Bilevel optimization: theory, algorithms, applications and a bibliography. In *Bilevel Optimization*, pages 581–672. Springer, 2020. doi:10.1007/978-3-030-52119-6_20.

[80] S. DeNegre. *Interdiction and Discrete Bilevel Linear Programming*. PhD thesis, Lehigh University, TBethlehem, PA, 2011.

[81] S. T. DeNegre and T. K. Ralphs. A branch-and-cut algorithm for integer bilevel linear programs. In J.W. Chinneck and M.J. Kristjansson, B. Saltzman, editors, *Operations research and cyber-infrastructure*, volume 47, pages 65–78. Springer, Boston, MA, 2009. doi:10.1007/978-0-387-88843-9_4.

[82] X. Deng. Complexity issues in bilevel linear programming. In A. Migdalas, P. M. Pardalos, and P. Värbrand, editors, *Multilevel Optimization: Algorithms and Applications*, pages 149–164. Springer US, Boston, MA, 1998. doi:10.1007/978-1-4613-0307-7_6.

[83] H. W. Dommel and W. F. Tinney. Optimal Power Flow Solutions. *IEEE Transactions on Power Apparatus and Systems*, PAS-87(10):1866–1876, 1968. doi:10.1109/TPAS.1968.292150.

[84] W. S. Dorn. Duality in Quadratic Programming. *Quarterly of Applied Mathematics*, 18(2):155–162, 1960. URL: https://www.jstor.org/stable/43636320.

[85] W. S. Dorn. Self-dual quadratic programs. *Journal of the Society for Industrial and Applied Mathematics*, 9(1):51–54, 1961. doi:10.1137/0109006.

[86] T. A. Edmunds and J. F. Bard. Algorithms for nonlinear bilevel mathematical programs. *IEEE transactions on Systems, Man, and Cybernetics*, 21(1):83–89, 1991. doi:10.1109/21.101139.

[87] T. A. Edmunds and J. F. Bard. An algorithm for the mixed-integer nonlinear bilevel programming problem. *Annals of Operations Research*, 34(1):149–162, 1992. doi:10.1007/BF02098177.

[88] S.C. Fang, C.J. Lin, and S.Y. Wu. Solving quadratic semi-infinite programming problems by using relaxed cutting-plane scheme. *Journal of Computational and Applied Mathematics*, 129(1):89–104, 2001. doi:10.1016/S0377-0427(00)00544-6.

[89] N. Faísca, V. Dua, B. Rustem, P. Saraiva, and E. Pistikopoulos. Parametric global optimisation for bilevel programming. *Journal of Global Optimization*, 38:609–623, 08 2007. doi:10.1007/s10898-006-9100-6.

[90] M. C. Ferris and Y. Liu. Modelling demand response in organized wholesale energy markets. *Optimization Methods and Software*, 31(5):1064–1088, 2016. doi:10.1080/10556788.2016.1177527.

[91] F. Fioretto, T. W.K. Mak, and P. Van Hentenryck. Privacy-preserving obfuscation of critical infrastructure networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 1086–1092. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi:10.24963/ijcai.2019/152.

[92] M. Fischetti, I. Ljubić, M. Monaci, and M. Sinnl. On the use of intersection cuts for bilevel optimization. *Mathematical Programming*, 172(1):77–103, 2018. doi:10.1007/s10107-017-1189-5.

[93] E. B. Fisher, R. P. O'Neill, and M. C. Ferris. Optimal Transmission Switching. IEEE Transactions on Power Systems, 23(3):1346–1355, 2008. doi:10.1109/TPWRS.2008.922256.

[94] J. Fortuny-Amat and B. McCarl. A representation and economic interpretation of a two-level programming problem. *The Journal of the Operational Research Society*, 32(9):783–792, 1981. doi:10.2307/2581394.

[95] R. Fourer, D. M. Gay, and B. W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. Cengage Learning, Boston, MA, 2002.

[96] S. Frank, I. Steponavice, and S. Rebennack. Optimal power flow: a bibliographic survey II. *Energy Systems*, 3(3):259 – 289, 2012. doi:10.1007/s12667-012-0057-x.

[97] S. M. Frank, I. Steponavice, and S. Rebennack. Optimal power flow: a bibliographic survey I formulations and deterministic methods. *Energy Systems*, 3(3):221 – 258, 2012. doi:10.1007/s12667-012-0056-y.

[98] E. Frazzoli, Z. Mao, J. Oh, and E. Feron. Resolution of conflicts involving many aircraft via semidefinite programming. *Journal of Guidance, Control and Dynamic*, 24(1):79–86, 2001. doi:10.2514/2.4678.

[99] S. A. Gabriel, A. J. Conejo, J. D. Fuller, B. F. Hobbs, and C. Ruiz. *Complementarity modeling in energy markets*, volume 180. Springer Science & Business Media, 2012. doi:10.1007/978-1-4419-6123-5.

[100] L. Gan, N. Li, S. Low, and U. Topcu. Exact convex relaxation for optimal power flow in distribution networks. *ACM SIGMETRICS Performance Evaluation Review*, 41, 2013. doi:10.1145/2465529.2465535.

[101] P. Garcia-Herreros, L. Zhang, P. Misra, E. Arslan, S. Mehta, and I.E. Grossmann. Mixed-integer bilevel optimization for capacity planning with rational markets. *Computers & Chemical Engineering*, 86:33–47, 2016. doi:10.1016/j.compchemeng.2015.12.007.

[102] S.A. Gerschgorin. Über die abgrenzung der eigenwerte einer matrix. Izvestia Akademii Nauk USSR, 6:749–754, 1931.

[103] J. C. Gilbert and C. Josz. Plea for a semidefinite optimization solver in complex numbers. Technical report, HAL Archives-Ouvertes: hal-01422932, 2017. URL: https://hal.inria.fr/hal-01422932v2/document.

[104] P.E. Gill. *User's Guide for SNOPT 5.3*. Systems Optimization Laboratory, Department of EESOR, Stanford University, California, February 1999.

[105] T. C. Giras and S. N. Talukdar. Quasi-Newton method for optimal power flows. *International Journal of Electrical Power & Energy Systems*, 3(2):59–64, 1981. doi:10.1016/0142-0615(81)90010-7.

[106] V. Grimm, G. Orlinskaya, L. Schewe, M. Schmidt, and G. Zöttl. Optimal design of retailer-prosumer electricity tariffs using bilevel optimization. *Omega*, 102(C), 2021. URL: https://ideas.repec.org/a/eee/jomega/v102y2021ics0305048320306812.html, doi:10.1016/j.omega.2020.1023.

[107] Z.H. Gümüş and A.F. Christodoulos. Global optimization of mixed-integer bilevel programming problems. *Computational Management Science*, 2:181–212, 2005. doi:10.1007/s10287-005-0025-1.

[108] LLC Gurobi Optimization. Gurobi optimizer reference manual, 2021. URL: http://www.gurobi.com.

[109] P. Hansen, B. Jaumard, and G. Savard. New branch-and-bound rules for bilevel linear programming. *SIAM Journal on Scientific and Statistical Computing*, 13(5):1194–1217, 1992. doi:10.1137/0913069.

[110] R. Hettich. An implementation of a discretization method for semi-infinite programming. *Mathematical Programming*, 34(3):354–361, 1986. doi:10.1007/BF01582235.

[111] J. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms I: Fundamentals*, volume 305. Springer-Verlag Berlin Heidelberg, 2013. doi:10.1007/978-3-662-02796-7.

[112] K. Iba. Reactive power optimization by genetic algorithm. *IEEE Transactions on Power Systems*, 9(2):685–692, 1994. doi:10.1109/59.317674.

[113] IBM. *ILOG CPLEX 12.10* User's Manual. IBM, 2020.

[114] Y. Ishizuka and E. Aiyoshi. Double penalty method for bilevel optimization problems. *Annals of Operations Research*, 34(1):73–88, 1992. doi:10.1007/BF02098173.

[115] R. A. Jabr. Radial distribution load flow using conic programming. IEEE Transactions on Power Systems, 21(3):1458–1459, 2006. doi:10.1109/TPWRS.2006.879234.

[116] R. A. Jabr, A. H. Coonick, and B. J. Cory. A primal-dual interior point method for optimal power flow dispatching. *IEEE Transactions on Power Systems*, 17(3):654–662, 2002. doi:10.1109/TPWRS.2002.800870.

[117] R. G. Jeroslow. The polynomial hierarchy and a simple model for competitive analysis. *Mathematical Programming*, 32:146–164, 1985. doi:10.1007/BF01586088.

[118] J. J. Júdice and A. Faustino. A sequential lcp method for bilevel linear programming. *Annals of Operations Research*, 34(1):89–106, 1992. doi:10.1007/BF02098174.

[119] J. J. Júdice and A. Faustino. The linear-quadratic bilevel programming problem. *INFOR: Information Systems and Operational Research*, 32(2):87–98, 1994. doi:10.1080/03155986.1994.11732240.

[120] J. J. Júdice and G. Mitra. Reformulation of mathematical programming problems as linear complementarity problems and investigation of their solution methods. *Journal of Optimization Theory and Applications*, 57:123–149, 1988. doi:10.1007/BF00939332.

[121] J.E. Kelley, Jr. The cutting-plane method for solving convex programs. *Journal of the society for Industrial and Applied Mathematics*, 8(4):703–712, 1960. URL: https://www.jstor.org/stable/2099058.

[122] T. Kis, A. Kovács, and C. Mészáros. On optimistic and pessimistic bilevel optimization models for demand response management. *Energies*, 14(8), 2021. doi:10.3390/en14082095.

[123] T. Kleinert, M. Labbé, I. Ljubić, and M. Schmidt. A survey on mixed-integer programming techniques in bilevel optimization. *preprint*, 2021. URL: https://hal.inria.fr/hal-03095139/file/bilevel_survey-04012021%20%281%29.pdf.

[124] T. Kleinert, M. Labbé, F. Plein, and M. Schmidt. Technical Note – There's no free lunch: On the hardness of choosing a correct big-M in bilevel optimization. *Operations Research*, 68(6):1716–1721, 2020. doi:10.1287/opre.2019.1944.

[125] U.G.W. Knight. The logical design of electrical networks using linear programming methods. Proceedings of IEE, 107(33):306–314, 1960. doi:10.1049/pi-a.1960.0063.

[126] B. Kocuk, S. Dey, and X. Sun. Strong SOCP relaxations for the optimal power flow problem. *Operations Research*, 64, 05 2016. doi:10.1287/opre.2016.1489.

[127] M. Kojima and L. Tunçel. Cones of matrices and successive convex relaxations of nonconvex sets. *SIAM Journal on Optimization*, 10:750–778, 2000. doi:10.1137/S1052623498336450.

[128] C. D. Kolstad and L. S. Lasdon. Derivative evaluation and computational experience with large bilevel mathematical programs. *Journal of optimization theory and applications*, 65(3):485–499, 1990. doi:10.1007/BF00939562.

[129] K. O. Kortanek and H. No. A central cutting plane algorithm for convex semi-infinite programming problems. *SIAM Journal on optimization*, 3(4):901–918, 1993. doi:10.1137/0803047.

[130] L. Lampariello, S. Sagratella, and O. Stein. The standard pessimistic bilevel problem. *SIAM Journal on Optimization*, 29(2):1634–1656, 2019. doi:10.1137/18M119759X.

[131] L. Liberti, S. Cafieri, and F. Tarissan. Reformulations in mathematical programming: A computational approach. In A. Abraham et al., editors,

Foundations of Computational Intelligence Volume 3: Global Optimization, pages 153–234. Springer, Berlin, Heidelberg, 2009. doi:10.1007/978-3-642-01085-9_7.

[132] G. Lin, M. Xu, and J. Ye. On solving simple bilevel programs with a nonconvex lower level program. *Mathematical Programming*, 144:277–305, 2014. doi:10.1007/s10107-013-0633-4.

[133] P. Lipka, S. S. Oren, R. P. O'Neill, and A. Castillo. Running a more complete market with the SLP-IV-ACOPF. *IEEE Transactions on Power Systems*, 32(2):1139–1148, 2017. doi:10.1109/TPWRS.2016.2569596.

[134] C. Liu, C. Lin, S. Shiraishi, and . Tomizuka. Distributed conflict resolution for connected autonomous vehicles. *IEEE Transactions on Intelligent Vehicles*, 3(1):18–29, 2018. doi:10.1109/TIV.2017.2788209.

[135] J. Liu, Y. Fan, Z. Chen, and Y. Zheng. Pessimistic bilevel optimization: A survey. *International Journal of Computational Intelligence Systems*, 11:725–736, 03 2018. doi:10.2991/ijcis.11.1.56.

[136] Y. Liu and T. H. Spencer. Solving a bilevel linear program when the inner decision maker controls few variables. *European Journal of Operational Research*, 81(3):644–651, 1995. doi:10.1016/0377-2217(94)00005-W.

[137] F. Locatello, M. Tschannen, G. Rätsch, and M. Jaggi. Greedy algorithms for cone constrained optimization with convergence guarantees. *arXiv preprint*, 2017. URL: http://arxiv.org/abs/1705.11041.

[138] A. Lodi, T. K. Ralphs, and G. J. Woeginger. Bilevel programming and the separation problem. *Mathematical Programming*, 146:437–458, 2014. doi:10.1007/s10107-013-0700-x.

[139] Z. Luo, J. Pang, and D. Ralph. *Mathematical programs with equilibrium constraints*. Cambridge University Press, 1996. doi:10.1017/CBO9780511983658.

[140] Y. Lv, T. Hu, G. Wang, and Z. Wan. A penalty function method based on kuhn-tucker condition for solving linear bilevel programming. *Applied Mathematics and Computation*, 188(1):808–813, 2007. doi:10.1016/j.amc.2006.10.045.

[141] R. Mansi, C. Alves, V. J. M. de Carvalho, and S. Hanafi. An exact algorithm for bilevel 0-1 knapsack problems. *Mathematical Problems in Engineering*, 2012. doi:10.1155/2012/504713.

[142] H. Maravillo, J. Camacho-Vallejo, J. Puerto, and M. Labbé. A market regulation bilevel problem: A case study of the mexican petrochemical industry. *Omega*, 97:102–105, 2020. doi:10.1016/j.omega.2019.08.012.

[143] P. Marcotte, G. Savard, and D. Zhu. A trust region algorithm for nonlinear bilevel programming. *Operations research letters*, 29(4):171–179, 2001. doi:10.1016/S0167-6377(01)00092-X.

[144] Patrice Marcotte. Network design problem with congestion effects: A case of bilevel programming. *Mathematical programming*, 34(2):142–162, 1986. doi:10.1007/BF01580580.

[145] M. Merkert, G. Orlinskaya, and D. Weninger. An exact projection-based algorithm for bilevel mixed-integer problems with nonlinearities. *Optimization Online Preprint*, 2020. URL: http://www.optimization-online.org/DB_FILE/2020/12/8153.pdf.

[146] A. R. Meyer and L. J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *13th Annual Symposium on Switching and Automata Theory (swat 1972)*, pages 125–129, 1972. doi:10.1109/SWAT.1972.29.

[147] A. Migdalas. Bilevel programming in traffic planning: Models, methods and challenge. *Journal of global optimization*, 7(4):381–405, 1995. doi:10.1007/BF01099649.

[148] A. Mitsos, P. Lemonidis, and P. Barton. Global solution of bilevel programs with a nonconvex inner program. *J. Global Optimization*, 42:475–513, 2008. doi:10.1007/s10898-007-9260-z.

[149] D. K. Molzahn and I. A. Hiskens. A Survey of Relaxations and Approximations of the Power Flow Equations. Foundations and Trends in Electric Energy Systems. Now Publishers, Hanover, MA, 2019. doi:10.1561/3100000012.

[150] J. A. Momoh and J. Z. Zhu. Improved interior point method for OPF problems. *IEEE Transactions on Power Systems*, 14(3):1114–1120, 1999. doi:10.1109/59.780938.

[151] J. T. Moore and J. F. Bard. The mixed integer linear bilevel programming problem. *Operations research*, 38(5):911–921, 1990. doi:10.1287/opre.38.5.911.

[152] MOSEK ApS. *The MOSEK python optimizer API manual. Version 9.2.36*, 2021. URL: https://docs.mosek.com/9.2/pythonapi/index.html.

[153] W.P. Niedringhaus. Stream Option Manager (SOM): automated integration of aircraft separation, merging, stream management, and other air traffic control functions. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(9):1269–1280, 1995. doi:10.1109/21.400505.

[154] J. Omer. A space-discretized mixed-integer linear model for air-conflict resolution with speed and heading maneuvers. *Computers and Operations Research*, 58:75–86, 2015. doi:10.1016/j.cor.2014.12.012.

[155] J. Omer and J.L. Farges. Hybridization of Nonlinear and Mixed-Integer Linear Programming for aircraft separation with trajectory recovery. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1218–1230, 2013. doi:10.1109/TITS.2013.2257758.

[156] Alena Otto, Niels Agatz, James Campbell, Bruce Golden, and Erwin Pesch. Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. *Networks*, 72(4):411–458, 2018. doi:10.1002/net.21818.

[157] J. Outrata, M. Kocvara, and J. Zowe. *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints*. Nonconvex Optimization and Its Applications. Springer, Boston, MA, 1 edition, 1998. doi:10.1007/978-1-4757-2825-5.

[158] L. Pallottino, E. Feron, and A. Bicchi. Conflict resolution problems for air traffic management systems solved with mixed integer programming. *IEEE transactions on intelligent transportation systems*, 3(1):3–11, 2002. doi:10.1109/6979.994791.

[159] G. P. Papavassilopoulos. Algorithms for static stackelberg games with linear costs and polyhedra constraints. In *1982 21st IEEE Conference on Decision and Control*, pages 647–652. IEEE, 1982. doi:10.1109/CDC.1982.268221.

[160] M. Pelegrín and C. D'Ambrosio. Aircraft deconfliction via mathematical programming: Review and insights. *Transportation Science*, to appear. URL: https://hal.archives-ouvertes.fr/hal-02902566v3/file/survey-HAL-v3.pdf.

[161] C. Peyronne, A. R. Conn, M. Mongeau, and D. Delahaye. Solving air traffic conflict problems via local continuous optimization. *European Journal of Operational Research*, 241(2):502–512, 2015. doi:10.1016/j.ejor.2014.08.045.

[162] R. Reemtsen and S. Görner. Numerical methods for semi-infinite programming: A survey. In R. Reemtsen and J. Rückmann, editors, *Semi-Infinite Programming*, pages 195–275. Springer, Boston, 1998. doi:10.1007/978-1-4757-2868-2_7.

[163] D. Rey, F.H. Dias, and H. Hijazi. Instances library for the Aircraft Conflict Resolution Problem. https://github.com/acrp-lib/acrp-lib, 2020.

[164] D. Rey and H. Hijazi. Complex number formulation and convex relaxations for aircraft conflict resolution. *IEEE 56th Annual Conference on Decision and Control*, pages 88–93, 2017. doi:10.1109/CDC.2017.8263648.

[165] D. Rey, C. Rapine, S. Constans, and R. Fondacci. A mixed integer linear model for potential conflict minimization by speed modulations. In *ICRAT 2010. Fourth International Conference on Research in Air Transportation*, pages 513–515, 06 2010. URL: https://hal.archives-ouvertes.fr/hal-00520052/file/David_Rey_-_Doctoral_Symposium.pdf.

[166] D. Rey, C. Rapine, V.V. Dixit, and S.T. Waller. Equity-oriented aircraft collision avoidance model. *IEEE Transactions on Intelligent Transportation Systems*, 16(1):172–183, 2014. doi:10.1109/TITS.2014.2329012.

[167] G. Romano. New results in subdifferential calculus with applications to convex optimization. *Applied Mathematics and Optimization*, 32:213–234, 1995. doi:10.1007/BF01187900.

[168] M. Ruiz, J. Maeght, A. Marié, P. Panciatici, and A. Renaud. A progressive method to solve large-scale AC optimal power flow with discrete variables and control of the feasibility. In 2014 Power Systems Computation Conference, pages 1–7, 2014. doi:10.1109/PSCC.2014.7038395.

[169] J. Ryu, V. Dua, and E.N. Pistikopoulos. A bilevel programming framework for enterprise-wide process networks under uncertainty. *Computers & Chemical Engineering*, 28(6):1121–1129, 2004. FOCAPO 2003 Special issue. doi:10.1016/j.compchemeng.2003.09.021.

[170] Avraamidou S. and E. N. Pistikopoulos. B-POP: Bi-level parametric optimization toolbox. *Computers & Chemical Engineering*, 122:193–202, 2019. 2017 Edition of the European Symposium on Computer Aided Process Engineering (ESCAPE-27). doi:10.1016/j.compchemeng.2018.07.007.

[171] Cafieri S. and C. D'Ambrosio. Feasibility pump for aircraft deconfliction with speed regulation. *Journal of Global Optimization*, 71(3):501–515, 2018. doi:10.1007/s10898-017-0560-7.

[172] G. Sagnol. On the semidefinite representation of real functions applied to symmetric matrices. *Linear Algebra and its Applications*, 439:2829–2843, 2013. doi:10.1016/j.laa.2013.08.021.

[173] G. K. Saharidis and M. G. Ierapetritou. Resolution method for mixed integer bi-level linear problems based on decomposition technique. *Journal of Global Optimization*, 44(1):29–51, 2009. doi:10.1007/s10898-008-9291-0.

[174] N. V. Sahinidis. *BARON 17.8.9: Global Optimization of Mixed-Integer Nonlinear Programs,* User's Manual, 2017. URL: https://www.minlp.com/downloads/docs/baron%20manual.pdf.

[175] E. Salgado, C. Gentile, and L. Liberti. Perspective Cuts for the ACOPF with Generators. In P. Daniele and L. Scrimali, editors, New Trends in Emerging Complex Real Life Problems, volume 1 of *AIRO Series*, pages 451–461. Springer, Cham, 2018. doi:10.1007/978-3-030-00473-6_48.

[176] E. Salgado, A. Scozzari, F. Tardella, and L. Liberti. Alternating Current Optimal Power Flow with Generator Selection. In J. Lee, G. Rinaldi, and R. Mahjoub, editors, Combinatorial Optimization ISCO 2018, volume 10856 of *LNCS*, pages 364–375, 2018. doi:10.1007/978-3-319-96151-4_31.

[177] G. Savard and J. Gauvin. The steepest descent direction for the nonlinear bilevel programming problem. *Operations Research Letters*, 15(5):265–272, 1994. doi:10.1016/0167-6377(94)90086-8.

[178] H. Scheel and S. Scholtes. Mathematical programs with complementarity constraints: Stationarity, optimality, and sensitivity. *Mathematics of Operations Research*, 25(1):1–22, 2000. doi:10.1287/moor.25.1.1.15213.

[179] K. Shimizu and E. Aiyoshi. A new computational method for stackelberg and min-max problems by use of a penalty method. *IEEE Transactions on Automatic Control*, 26(2):460–466, 1981. doi:10.1109/TAC.1981.1102607.

[180] A. Sinha, P. Malo, and K. Deb. A review on bilevel optimization: From classical to evolutionary approaches and applications. *IEEE Transactions on Evolutionary Computation*, 22(2):276–295, 2018. doi:10.1109/TEVC.2017.2712906.

[181] G. Still. Discretization in semi-infinite programming: the rate of convergence. *Mathematical programming*, 91(1):53–69, 2001. doi:10.1007/s101070100239.

[182] L. J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976. doi:10.1016/0304-3975(76)90061-X.

[183] D. I. Sun, B. Ashley, B. Brewer, A. Hughes, and W. F. Tinney. Optimal Power Flow By Newton Approach. *IEEE Transactions on Power Apparatus and Systems*, PAS-103(10):2864–2880, 1984. doi:10.1109/TPAS.1984.318284.

[184] P. Tabaghi, I. Dokmanić, and M. Vetterli. Kinetic Euclidean distance matrices. *IEEE Transactions on Signal Processing*, 68:452–465, 2020. doi:10.1109/TSP.2019.2959260.

[185] S. Tahernejad, T. K. Ralphs, and S. T. DeNegre. A branch-and-cut algorithm for mixed integer bilevel linear optimization problems and its implementation. *Mathematical Programming Com putation*, 12(4):529–568, 2020. doi:10.1007/s12532-020-00183-6.

[186] A. Takeda and M. Kojima. Successive convex relaxation approach to bilevel quadratic optimization problems. In M. C. Ferris et al., editors, *Complementarity: Applications, Algorithms and Extensions*, pages 317–340. Springer-Boston, 2000. doi:10.1007/978-1-4757-3279-5_15.

[187] Y. Tanaka, M. Fukushima, and T. Ibaraki. A globally convergent SQP method for semi-infinite nonlinear optimization. *Journal of Computational and Applied Mathematics*, 23(2):141–153, 1988. doi:10.1016/0377-0427(88)90276-2.

[188] A. Toffler. *The third wave*. Morrow, 1980.

[189] H. Tuy. *Convex Analysis and Global Optimization*, volume 22. Springer Boston, MA, 2 edition, 1998. doi:10.1007/978-3-319-31484-6.

[190] H. Tuy, A. Migdalas, and P. Värbrand. A global optimization approach for the linear two-level program. *Journal of Global Optimization*, 3(1):1–23, 1993. doi:10.1007/BF01100237.

[191] G. Van Rossum and F. L. Drake, Jr. *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995. URL: https://books.google.it/books?id=hHMVPwAACAAJ.

[192] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM review*, 38(1):49–95, 1996. doi:10.1137/1038003.

[193] A. Vela, S. Solak, W. Singhose, and J. Clarke. A mixed integer program for flight-level assignment and speed control for conflict resolution. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 5219–5226, 2009. doi:10.1109/CDC.2009.5400520.

[194] L. Vicente, G. Savard, and J. Júdice. Descent approaches for quadratic bilevel programming. *Journal of Optimization Theory and Applications*, 81(2):379–399, 1994. doi:10.1007/bf02191670.

[195] L. N. Vicente and P. H. Calamai. Bilevel and multilevel programming: A bibliography review. *Journal of Global optimization*, 5(3):291–306, 1994. doi:10.1007/BF01096458.

[196] H. Von Stackelberg. *Marktform und gleichgewicht*. J. springer, 1934.

[197] H. Von Stackelberg et al. *Theory of the market economy*. Oxford University Press, 1952.

[198] A. Wächter and L. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25–57, 2006. doi:10.1007/s10107-004-0559-y.

[199] L. Wang and F. Guo. Semidefinite relaxations for semi-infinite polynomial programming. *Computational Optimization and Applications*, 58:133–159, 2014. doi:10.1007/s10589-013-9612-1.

[200] J. Watson, A. Castillo, P. Lipka, S. Oren, and R. O'Neill. A current-voltage successive linear programming approach to solving the ACOPF. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2014. doi:10.1109/TPWRS.2016.1234567.

[201] U.P. Wen and Y.H. Yang. Algorithms for solving the mixed integer two-level linear programming problem. *Computers & Operations Research*, 17(2):133–142, 1990. doi:10.1016/0305-0548(90)90037-8.

[202] D. J. White and G. Anandalingam. A penalty function approach for solving bi-level linear programs. *Journal of Global Optimization*, 3(4):397–419, 1993. doi:10.1007/BF01096412.

[203] W. Wiesemann, A. Tsoukalas, P. Kleniati, and B. Rustem. Pessimistic bilevel optimization. *SIAM Journal on Optimization*, 23(1):353–380, 2013. doi:10.1137/120864015.

[204] G. J. Woeginger. The trouble with the second quantifier. *4OR*, 19(2):157–181, 2021. doi:10.1007/s10288-021-00477-y.

[205] S. Wogrin, S. Pineda, and D. A. Tejada-Arango. Applications of bilevel optimization in energy and electricity markets. In S. Dempe and A. Zemkoho, editors, *Bilevel Optimization*, pages 139–168. Springer, 2020. doi:10.1007/978-3-030-52119-6_5.

[206] P. Wolfe. A duality theorem for non-linear programming. *Quarterly of applied mathematics*, 19(3):239–244, 1961. doi:10.1090/QAM/135625.

[207] P. Xu. *Three essays on bilevel optimization algorithms and applications*. PhD thesis, PhD thesis. Iowa State University, 2012. URL: https://lib.dr.iastate.edu/etd/12964, doi:10.31274/etd-180810-47.

[208] Y. Xu, W. Sun, and L. Qi. On solving a class of linear semi-infinite programming by SDP method. *Optimization*, 64(3):603–616, 2015. doi:10.1080/02331934.2013.793325.

[209] S.B. Yaakob and J. Watada. Solving bilevel quadratic programming problems and its application. In *Proceedings of the 15th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems - Volume Part III*, pages 187–196. Springer-Verlag, 2011. URL: https://dl.acm.org/doi/10.5555/2041420.2041443.

[210] D. Yue, J. Gao, and F. You. A projection-based reformulation and decomposition algorithm for global optimization of mixed integer bilevel linear programs. *Journal of Global Optimization*, 73:27–57, 2019. doi:10.1007/s10898-018-0679-1.

[211] B. Zeng and Y. An. Solving bilevel mixed integer program by reformulations and decomposition. *Optimization online*, pages 1–34, 2014. URL: http://www.optimization-online.org/DB_HTML/2014/07/4455.html.

[212] R. D. Zimmerman and C. E. Murillo-Sánchez. *Matpower 7.0b1 User's Manual*. Power Systems Engineering Research Center, 2018.

Appendix A

Recall on complex numbers

A complex number x is a number which can be expressed (using the Cartesian coordinates) in the form

$$x = \Re(x) + \boldsymbol{i}\Im(x),$$

where $\Re(x)$ and $\Im(x)$ are real numbers defined as the real and imaginary parts of x respectively, and $\boldsymbol{i}$ is the imaginary unit, i.e., the square root of -1. Its *modulus* or *absolute value* $|x|$ is defined as

$$|x| = \sqrt{[\Re(x)]^2 + [\Im(x)]^2}$$

The complex conjugate of the complex number x is denoted by x^*, and is given by

$$\Re(x) - \boldsymbol{i}\Im(x).$$

An alternative option for Cartesian representation of x is the polar representation, which is

$$x = \alpha e^{\boldsymbol{i}\theta} = \alpha \cos\theta + \boldsymbol{i}\alpha \sin\theta,$$

where α is the magnitude, and θ is known as the *angle* or *phase* of the complex number. The relationship among Cartesian and polar coordinates is the following:

$$\begin{array}{rr} \Re(x) = \alpha\cos\theta & \alpha = \sqrt{[\Re(x)]^2 + [\Im(x)]^2} \\ \Im(x) = \alpha\sin\theta & \theta = \arccos(\frac{\Re x}{\alpha}) = \arcsin(\frac{\Im x}{\alpha}) \end{array}$$

Appendix B

Reformulating the bilevel problem with ACOPF as lower level

In this appendix, we propose two reformulations of the bilevel problem (5.7)–(5.8), based on strong duality and KKT conditions of the lower level (5.8) respectively. First of all, let us rewrite the lower level problem (5.8) in a different form:

- each linear equality constraint as $c^\top x = b_r$;
- each linear inequality constraint as $a^\top x \geqslant b_1$;
- the SOCP last constraint (5.8i) as $A_3 x - b_3 \geqslant_{\boldsymbol{L}^4} 0$, where $\boldsymbol{L}^4$ is the so called 4-dimensional Lorentz or second-order cone, and, given any $x \in \mathbb{R}^4$, $x \geqslant_{\boldsymbol{L}^4} 0$ is equivalent to $x_1^2 + x_2^2 + x_3^2 \leqslant x_4^2$;
- the quadratic inequality contraints (5.8g)–(5.8h) as SOCP constraints in the form $A_2 x - b_2 \geqslant_{\boldsymbol{L}^3} 0$, by introducing two additional linear constraints involving the slack variables zc and $zd \in \mathbb{R}^{|N||T|}$. Indeed each constraint (5.8g) is equivalent to $(\Re(c_i^t))^2 + (\Im(c_i^t))^2 \leqslant (z_i^t)^2 \ \wedge \ z_i^t = \overline{c}_i$, and each constraint (5.8h) to $(\Re(d_i^t))^2 + (\Im(d_i^t))^2 \leqslant (u_i^t)^2 \ \wedge \ u_i^t = \overline{d}_i$.

In our case we have, for constraint (5.8i):

$$A_3 := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{-1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}, \ b_3 := \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \text{ and } x := \begin{pmatrix} a_{ij}^t \\ q_{ij}^t \\ a_{ii}^t \\ a_{jj}^t \end{pmatrix},$$

while for constraints (5.8g), and (5.8h):

$$A_2 := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},\ b_2 := \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$

and $x := \begin{pmatrix} \Re(c_i^t) \\ \Im(c_i^t) \\ z_i^t \end{pmatrix}$, or $x := \begin{pmatrix} \Re(d_i^t) \\ \Im(d_i^t) \\ u_i^t \end{pmatrix}$, respectively.

The lower-level problem in this form thus reads:

$$\min_{c,d,B,a,q} \sum_{t\in T}\sum_{i\in N} \left(\hat{p}\left(\Re(v_i^t) + \Re(d_i^t) - \Re(c_i^t) - \Re(s_i^t)\right) + \check{p}\left(\Im(v_i^t) + \Im(d_i^t) - \Im(c_i^t) - \Im(s_i^t)\right)\right) \tag{B.1a}$$

$$\text{s.t.}\quad \forall t \in T\backslash\{t_n\} \wedge \forall i \in N:\ B_i^{t+1} - \alpha B_i^t + \beta\Re(c_i^t) - \frac{\Re(d_i^t)}{\beta} = 0 \tag{B.1b}$$

$$\forall t \in T \wedge \forall i \in N:\ B_i^t \geqslant 0 \tag{B.1c}$$

$$\forall t \in T \wedge \forall i \in N:\ -B_i^t \geqslant -\overline{B}_i \tag{B.1d}$$

$$\forall t \in T \wedge \forall i \in N:\ \Re(d_i^t) - \Re(c_i^t) - \sum_{j:j\sim i}\left(a_{ii}^t m_{ij} - a_{ij}^t m_{ij} - q_{ij}^t n_{ij})\right) = \Re(s_i^t) - \Re(v_i^t) \tag{B.1e}$$

$$\forall t \in T \wedge \forall i \in N:\ \Im(d_i^t) - \Im(c_i^t) - \sum_{j:j\sim i}\left(-a_{ii}^t n_{ij} + a_{ij}^t n_{ij} - q_{ij}^t m_{ij}\right) = \Im(s_i^t) - \Im(v_i^t) \tag{B.1f}$$

$$\forall t \in T \wedge \forall i \in N:\ a_{ii}^t \geqslant \underline{V}_i^2 \tag{B.1g}$$

$$\forall t \in T \wedge \forall i \in N:\ -a_{ii}^t \geqslant -\overline{V}_i^2 \tag{B.1h}$$

$$\forall t \in T \wedge \forall i \in N:\ z_i^t = \overline{c}_i^2 \tag{B.1i}$$

$$\forall t \in T \wedge \forall i \in N:\ u_i^t = \overline{d}_i^2 \tag{B.1j}$$

$$\forall t \in T \wedge \forall i \in N: \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \Re(c_i^t) \\ \Im(c_i^t) \\ z_i^t \end{pmatrix} \geqslant_{\boldsymbol{L}^3} 0 \tag{B.1k}$$

$$\forall t \in T \wedge \forall i \in N: \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \Re(d_i^t) \\ \Im(d_i^t) \\ u_i^t \end{pmatrix} \geqslant_{\boldsymbol{L}^3} 0 \tag{B.1l}$$

$$\forall t \in T \wedge \forall i \to j: \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{-1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}\begin{pmatrix} a_{ij}^t \\ q_{ij}^t \\ a_{ii}^t \\ a_{jj}^t \end{pmatrix} \geqslant_{\boldsymbol{L}^4} 0. \tag{B.1m}$$

The dual multipliers associated to the constraints (B.1b)–(B.1m) are $\lambda_b \in \mathbb{R}^{|T-1||N|}$, $\lambda_c, \lambda_d, \lambda_g, \lambda_h, \lambda_i, \lambda_j$, as well as $\lambda_{k1}, \lambda_{k2}, \lambda_{k3}, \lambda_{l1}, \lambda_{l2}, \lambda_{l3} \in \mathbb{R}^{|T||N|}$, and $\lambda_{m1}, \lambda_{m2}, \lambda_{m3}, \lambda_{m4} \in \mathbb{R}^{|T||L|}$.

B.1 Strong duality reformulation

By applying the strong duality theorem to the lower-level problem (B.1), one can equivalently reformulate the bilevel problem (5.7)–(5.8) to a single-level problem. Before introducing the reformulation of our bilevel problem (5.7)–(5.8), let us consider a general SOCP problem (see [37]):

$$\min_{x \in \mathbb{R}^n} \{c^\top x \mid A_i x - b_i \geqslant_{\boldsymbol{L}^{n_i}} 0, i = 1, \ldots, k\}, \tag{$SOCP$}$$

where $c \in \mathbb{R}^n$, $A_i \in \mathbb{R}^{n_i \times n}$, $b_i \in \mathbb{R}^{n_i}$, and $\boldsymbol{L}^{n_i}$ is the n_i-dimensional second-order cone. We recall that, for a vector $z \in \mathbb{R}^{n_i}$, the inequality $z \geqslant_{\boldsymbol{L}^{n_i}} 0$ means that

$$z_{n_i} \geqslant \left\| \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_{n_i - 1} \end{pmatrix} \right\|_2 .$$

Denoting the dual variable

$$\lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_k \end{pmatrix}$$

with $\lambda_i \in \mathbb{R}^{n_i}$ for all $i = 1, \ldots, k$, the dual of ($SOCP$) is the following:

$$\max_{\lambda} \{\sum_{i=1}^{k} b_i^\top \lambda_i \mid \sum_{i=1}^{k} A_i^\top \lambda_i = c, \lambda_i \geqslant_{\boldsymbol{L}_i^n} 0, i = 1, \ldots, k\}.$$

Our lower-level problem (B.1), is in the form of ($SOCP$) with:

- $n_i = 2$, $\forall i = 1, \ldots, h$, since the first constraints are linear. In Eqs. (B.1) there are also equality constraints, but they can easily reformulated into inequality constraints of the considered type.
- $n_i = n_j = 3$, since constraints (B.1k) and (B.1l) are convex quadratic
- $n_k \geqslant 4$, but, since the entries of A_k are all zero but the one of the submatrix A, and b_k is the zero-vector, we can reduce this last constraint to the SOCP constraint $Ax \geqslant_{\boldsymbol{L}^4} 0$ in (B.1m).

Therefore, we can reformulate the bilevel problem (5.7)–(5.8) by adding to the upper level: the original primal lower-level constraints (B.1b)–(B.1m), the dual lower-level constraints:

$$\left.\begin{array}{r}
\forall t \in T \wedge \forall i \in N: \ (\lambda_c)_i^t, (\lambda_d)_i^t, (\lambda_g)_i^t, (\lambda_h)_i^t \geqslant 0 \\
\forall t \in T \wedge \forall i \in N: \ (\lambda_k)_i^t \geqslant_{\boldsymbol{L}^3} 0 \\
\forall t \in T \wedge \forall i \in N: \ (\lambda_l)_i^t \geqslant_{\boldsymbol{L}^3} 0 \\
\forall t \in T \wedge \forall i \to j: \ (\lambda_m)_{ij}^t \geqslant_{\boldsymbol{L}^4} 0 \\
\forall t \in T \backslash t_n \wedge \forall i \in N: \ \beta(\lambda_b)_i^t - (\lambda_e)_i^t + (\lambda_{k1})_i^t = \widehat{p} \\
\forall i \in N: \ -(\lambda_e)_i^{t_n} + (\lambda_{k1})_i^{t_n} = \widehat{p} \\
\forall t \in T \backslash t_n \wedge \forall i \in N: \ -\frac{(\lambda_b)_i^t}{\beta} + (\lambda_e)_i^t + (\lambda_{l1})_i^t = -\widehat{p} \\
\forall i \in N: \ (\lambda_e)_i^{t_n} + (\lambda_{l1})_i^{t_n} = -\widehat{p} \\
\forall t \in T \wedge \forall i \in N: \ -(\lambda_f)_i^t + (\lambda_{k2})_i^t = \check{p} \\
\forall t \in T \wedge \forall i \in N: \ (\lambda_f)_i^t + (\lambda_{l2})_i^t = -\check{p} \\
\forall t \in T \wedge \forall i \in N: \ (\lambda_i)_i^t + (\lambda_{k3})_i^t = 0 \\
\forall t \in T \wedge \forall i \in N: \ (\lambda_j)_i^t + (\lambda_{l3})_i^t = 0 \\
\forall t \in T \backslash t_1 \wedge \forall i \in N: \ (\lambda_b)_i^{t-1} - \alpha(\lambda_b)_i^t + (\lambda_c)_i^t - (\lambda_d)_i^t = 0 \\
\forall i \in N: \ -\alpha(\lambda_b)_i^{t_1} + (\lambda_c)_i^{t_1} - (\lambda_d)_i^{t_1} = 0 \\
\forall i \in N: \ (\lambda_b)_i^{t_n-1} + (\lambda_c)_i^{t_n} - (\lambda_d)_i^{t_n} = 0 \\
\forall t \in T \wedge \forall i \in N: \ \sum_{j:j\sim i} \left[-m_{ij}(\lambda_e)_i^t + n_{ij}(\lambda_f)_i^t\right] + (\lambda_g)_i^t - (\lambda_h)_i^t \qquad\qquad \\
+\frac{1}{2}\sum_{j:i\to j}\left[(\lambda_{m3})_{ij}^t + (\lambda_{m4})_{ij}^t\right] + \frac{1}{2}\sum_{j:j\to i}\left[(-\lambda_{m3})_{ji}^t + (\lambda_{m4})_{ji}^t\right] = 0 \\
\forall t \in T \wedge \forall i \to j \text{ s.t. } i,j \in N: \ m_{ij}\left[(\lambda_e)_i^t + (\lambda_e)_j^t\right] - n_{ij}\left[(\lambda_f)_i^t + (\lambda_f)_j^t\right] + (\lambda_{m1})_{ij}^t = 0 \\
\forall t \in T \wedge \forall i \to j \text{ s.t. } i,j \in N: \ n_{ij}\left[(\lambda_e)_i^t + (\lambda_e)_j^t\right] + m_{ij}\left[(\lambda_f)_i^t + (\lambda_f)_j^t\right] + (\lambda_{m2})_{ij}^t = 0
\end{array}\right\} \tag{B.2}$$

and

$$\sum_{t\in T}\sum_{i\in N}\left(\widehat{p}\left(\Re(v_i^t) + \Re(d_i^t) - \Re(c_i^t) - \Re(s_i^t)\right) + \check{p}\left(\Im(v_i^t) + \Im(d_i^t) - \Im(c_i^t) - \Im(s_i^t)\right)\right) \leqslant \psi(\lambda_b, \ldots, \lambda_k), \tag{B.3}$$

where

$$\begin{aligned}
\psi(\lambda_b, \ldots, \lambda_k) \quad = \quad & \sum_{t\in T}\sum_{i\in N}\left[-\overline{B}_i(\lambda_d)_i^t + (\Re(s_i^t) - \Re(v_i^t))(\lambda_e)_i^t + (\Im(s_i^t) - \Im(v_i^t))(\lambda_f)_i^t\right. \\
& \left. + \underline{V}_i^2(\lambda_g)_i^t - \overline{V}_i^2(\lambda_h)_i^t - \overline{c}_i(\lambda_i)_i^t - \overline{d}_i(\lambda_j)_i^t\right]
\end{aligned}$$

is the dual lower-level objective function. Eq. (B.3) is needed to ensure strong duality among primal and dual lower-level problems. By weak duality, indeed,

$$\sum_{t\in T}\sum_{i\in N}\left(\widehat{p}\left(\Re(v_i^t) + \Re(d_i^t) - \Re(c_i^t) - \Re(s_i^t)\right) + \check{p}\left(\Im(v_i^t) + \Im(d_i^t) - \Im(c_i^t) - \Im(s_i^t)\right)\right) \geqslant \psi(\lambda_b, \ldots, \lambda_k)$$

is satisfied for each solution that satisfies (B.1b)–(B.1m), and (B.2).

The single-level formulation obtained using this approach is, therefore:

$$\max_{\widehat{p},\check{p},c,d,B,a,q,\lambda} \sum_{t\in T}\sum_{i\in N} \left(\widehat{p}\left(\Re(v_i^t)+\Re(d_i^t)-\Re(c_i^t)-\Re(s_i^t)\right)+\check{p}\left(\Im(v_i^t)+\Im(d_i^t)-\Im(c_i^t)-\Im(s_i^t)\right)\right) \tag{B.4}$$

$$\text{s.t. } \widehat{p},\check{p} \geqslant 0 \tag{B.5}$$

$$(\widehat{p},\check{p},c,d,B,a,q,\lambda) \text{ satisfies (B.1b)–(B.1m), (B.2), and (B.3).} \tag{B.6}$$

B.2 KKT reformulation

A classical reformulation of a bilevel problem with a convex lower level is obtained replacing the lower-level by its KKT conditions.

First of all, let us recall that for a conic program in the form

$$\begin{aligned}
&\min_x \ f(x) \\
&\text{s.t. } h(x) = 0 && (\gamma)\\
&\quad\ g(x) \geqslant 0 && (\mu)\\
&\quad\ Ax \geqslant_{\boldsymbol{L}^4} 0, && (\lambda)
\end{aligned}$$

given the KKT multipliers γ, μ and λ, and the Lagrangian $\mathcal{L}$, the KKT stationarity condition is:

$$\nabla\mathcal{L} = \nabla f(x) - \gamma\nabla h(x) - \mu\nabla g(x) - A^\top\lambda = 0.$$

For our bilevel formulation the term $A^\top\lambda$ – corresponding to the SOCP constraint (B.1m) – is:

$$A_3^\top(\lambda_m)_{ij}^t + A_2^\top\left((\lambda_k)_i^t+(\lambda_l)_i^t\right) = \begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&\frac{1}{2}&\frac{1}{2}\\0&0&\frac{-1}{2}&\frac{1}{2}\end{pmatrix}\begin{pmatrix}\lambda_{m1}\\\lambda_{m2}\\\lambda_{m3}\\\lambda_{m4}\end{pmatrix}_{ij}^t + \begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}\begin{pmatrix}\lambda_{k1}\\\lambda_{k2}\\\lambda_{k3}\end{pmatrix}_i^t$$

$$+\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}\begin{pmatrix}\lambda_{l1}\\\lambda_{l2}\\\lambda_{l3}\end{pmatrix}_i^t = \begin{pmatrix}\lambda_{m1}\\\lambda_{m2}\\\frac{\lambda_{m3}+\lambda_{m4}}{2}\\\frac{-\lambda_{m3}+\lambda_{m4}}{2}\end{pmatrix}_{ij}^t + \begin{pmatrix}\lambda_{k1}\\\lambda_{k2}\\\lambda_{k3}\end{pmatrix}_i^t + \begin{pmatrix}\lambda_{l1}\\\lambda_{l2}\\\lambda_{l3}\end{pmatrix}_i^t$$

The KKT conditions of the lower-level problem (B.1) are:

- Primal feasibility: constraints (B.1b)–(B.1m);

- Dual feasibility:

$$\forall t \in T \wedge \forall i \in N : \ (\lambda_c)_i^t, (\lambda_d)_i^t, (\lambda_g)_i^t, (\lambda_h)_i^t \geqslant 0,$$

$$\forall t \in T \wedge \forall i \in N : \ (\lambda_k)_i^t \geqslant_{\boldsymbol{L}^3} 0,$$

$$\forall t \in T \wedge \forall i \in N : \ (\lambda_l)_i^t \geqslant_{\boldsymbol{L}^3} 0,$$

and

$$\forall t \in T \wedge \forall i \to j : \ (\lambda_m)_{ij}^t \geqslant_{\boldsymbol{L}^4} 0;$$

- Stationarity conditions, given the Lagrangian $\mathcal{L}$:

  - $\frac{\partial \mathcal{L}}{\partial \Re(c_i^t)} = \widehat{p} - \beta(\lambda_b)_i^t + (\lambda_e)_i^t - (\lambda_{k1})_i^t = 0 \qquad \forall t \in T \backslash \{t_n\} \wedge \forall i \in N$
    $\frac{\partial \mathcal{L}}{\partial \Re(c_i^{t_n})} = \widehat{p} + (\lambda_e)_i^{t_n} - (\lambda_{k1})_i^{t_n} = 0 \qquad \forall i \in N$
  - $\frac{\partial \mathcal{L}}{\partial \Re(d_i^t)} = -\widehat{p} + \frac{(\lambda_b)_i^t}{\beta} - (\lambda_e)_i^t - (\lambda_{l1})_i^t = 0 \qquad \forall t \in T \backslash \{t_n\} \wedge \forall i \in N$
    $\frac{\partial \mathcal{L}}{\partial \Re(d_i^{t_n})} = -\widehat{p} - (\lambda_e)_i^{t_n} - (\lambda_{l1})_i^{t_n} = 0 \qquad \forall i \in N$
  - $\frac{\partial \mathcal{L}}{\partial \Im(c_i^t)} = \breve{p} + (\lambda_f)_i^t - (\lambda_{k2})_i^t = 0 \qquad \forall t \in T \wedge i \in N$
  - $\frac{\partial \mathcal{L}}{\partial \Im(d_i^t)} = -\breve{p} - (\lambda_f)_i^t - (\lambda_{l2})_i^t = 0 \qquad \forall t \in T \wedge i \in N$
  - $\frac{\partial \mathcal{L}}{\partial z_i^t} = -(\lambda_i)_i^t - (\lambda_{k3})_i^t = 0 \qquad \forall t \in T \wedge i \in N$
  - $\frac{\partial \mathcal{L}}{\partial u_i^t} = -(\lambda_j)_i^t - (\lambda_{l3})_i^t = 0 \qquad \forall t \in T \wedge i \in N$
  - $\frac{\partial \mathcal{L}}{\partial B_i^t} = -(\lambda_b)_i^{t-1} + \alpha(\lambda_b)_i^t - (\lambda_c)_i^t + (\lambda_d)_i^t = 0 \qquad \forall t \in T \backslash \{t_1, t_n\} \wedge \forall i \in N$
    $\frac{\partial \mathcal{L}}{\partial B_i^{t_1}} = \alpha(\lambda_b)_i^{t_1} - (\lambda_c)_i^{t_1} + (\lambda_d)_i^{t_1} = 0 \qquad \forall i \in N$
    $\frac{\partial \mathcal{L}}{\partial B_i^{t_n}} = -(\lambda_b)_i^{t_n-1} - (\lambda_c)_i^{t_n} + (\lambda_d)_i^{t_n} = 0 \qquad \forall i \in N$
  - $\frac{\partial \mathcal{L}}{\partial a_{ij}^t} = -m_{ij}[(\lambda_e)_i^t + (\lambda_e)_j^t] + n_{ij}[(\lambda_f)_i^t + (\lambda_f)_j^t] - (\lambda_{m1})_{ij}^t = 0 \qquad \forall t \in T \wedge \forall i : i \to j$
  - $\frac{\partial \mathcal{L}}{\partial q_{ij}^t} = -n_{ij}[(\lambda_e)_i^t + (\lambda_e)_j^t] - m_{ij}[(\lambda_f)_i^t + (\lambda_f)_j^t] - (\lambda_{m2})_{ij}^t = 0 \qquad \forall t \in T \wedge \forall i : i \to j$
  - $\frac{\partial \mathcal{L}}{\partial a_{ii}^t} = \sum_{j:i\sim j} [(\lambda_e)_i^t m_{ij} - (\lambda_f)_i^t n_{ij}] - (\lambda_g)_i^t + (\lambda_h)_i^t$
    $-\frac{1}{2} \sum_{j:i\to j} \left[(\lambda_{m3})_{ij}^t + (\lambda_{m4})_{ij}^t\right] - \frac{1}{2} \sum_{j:j\to i} \left[(-\lambda_{m3})_{ji}^t + (\lambda_{m4})_{ji}^t\right] = 0 \qquad \forall t \in T \wedge \forall i \in N$

- Complementarity conditions:

  - $(\lambda_c)_i^t\,(-B_i^t) = 0 \qquad \forall t \in T \wedge \forall i \in N$
  - $(\lambda_d)_i^t\,\left(B_i^t - \overline{B}_i\right) = 0 \qquad \forall t \in T \wedge \forall i \in N$
  - $(\lambda_g)_i^t\,\left(-a_{ii}^t + \underline{V}_i^2\right) = 0 \qquad \forall t \in T \wedge \forall i \in N$
  - $(\lambda_h)_i^t\,\left(a_{ii}^t - \overline{V}_i^2\right) = 0 \qquad \forall t \in T \wedge \forall i \in N$
  - $\begin{pmatrix}(\lambda_{k1})_{ij}^t \\ (\lambda_{k2})_{ij}^t \\ (\lambda_{k3})_{ij}^t\end{pmatrix}^\top \begin{pmatrix}1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1\end{pmatrix} \begin{pmatrix}\Re(c_i^t) \\ \Im(c_i^t) \\ z_i^t\end{pmatrix} = 0 \qquad \forall t \in T \wedge \forall i \in N$
  - $\begin{pmatrix}(\lambda_{l1})_{ij}^t \\ (\lambda_{l2})_{ij}^t \\ (\lambda_{l3})_{ij}^t\end{pmatrix}^\top \begin{pmatrix}1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1\end{pmatrix} \begin{pmatrix}\Re(d_i^t) \\ \Im(d_i^t) \\ u_i^t\end{pmatrix} = 0 \qquad \forall t \in T \wedge \forall i \in N$
  - $\begin{pmatrix}(\lambda_{m1})_{ij}^t \\ (\lambda_{m2})_{ij}^t \\ (\lambda_{m3})_{ij}^t \\ (\lambda_{m4})_{ij}^t\end{pmatrix}^\top \begin{pmatrix}1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{-1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2}\end{pmatrix} \begin{pmatrix}a_{ij}^t \\ q_{ij}^t \\ a_{ii}^t \\ a_{jj}^t\end{pmatrix} = 0 \qquad \forall t \in T \wedge \forall i \to j$

DOCTORAL SCHOOL

Titre : Optimisation biniveau et applications

Mots clés : programmation mathématique, programmation biniveau, résolution de conflits d'avions, ACOPF, programmation semi-infinie

Résumé : Un problème biniveau est un problème où un sous-ensemble des variables est contraint d'être optimal pour un autre problème paramétré par les variables restantes. Le problème externe est appelé problème de niveau supérieur, le problème interne le problème de niveau inférieur. La première partie de cette thèse concerne les définitions clés, les approches de solution et la complexité des problèmes biniveaux, et l'étude d'une classe particulière de problèmes biniveaux, ayant un niveau inférieur quadratique, dont la valeur est contenue dans une contrainte de niveau supérieur. Nous proposons une approche pour résoudre cette classe de problèmes, basée sur la dualisation du niveau inférieur. Cette approche est comparée à un algorithme de plans coupants, dont nous prouvons la convergence. La validité de ces deux approches est démontrée par les résultats de calcul sur deux applications: un jeu à somme nulle avec un gain cubique et une régression quadratique contrainte. La deuxième partie de la thèse est consacrée aux applications pratiques. Un chapitre est dédié au problème de résolution de conflits d'aéronefs (PRC). Ce problème consiste essentiellement à imposer une distance minimale entre les avions en vol pour éviter les conflits, en utilisant différentes stratégies. Nous nous concentrons sur deux d'entre eux: les régulations de vitesse et les changements d'angle de cap. Nous présentons une formulation de programmation semi-infinie du PRC via régulation de vitesse en k dimensions. Nous la reformulons d'une part en utilisant la programmation polynomiale et d'autre part en utilisant la programmation biniveau. Ensuite, nous présentons une formulation biniveau du PRC via changements d'angle de cap en deux dimensions. Dans les deux formulations biniveau, la convexité des niveaux inférieurs nous permet de proposer trois reformulations différentes à un seul niveau, en utilisant les conditions KKT, la dualité de Dorn et la dualité de Wolfe. Les reformulations à un seul niveau des deux problèmes sont résolues en utilisant des solveurs de l'état de l'art. Alternativement, nous proposons un algorithme de génération de coupes pour résoudre les problèmes biniveau, qui s'inscrit dans le cadre général de l'algorithme de plans coupants présenté dans la première partie. Cet algorithme obtient les meilleurs résultats en terme de temps pour la plupart des instances testées. Une autre application étudiée dans cette thèse concerne le Alternating Current (AC) Optimal Power Flow (ACOPF) au niveau inférieur. Dans un horizon temporel discrétisé fixe, un problème biniveau est derivé pour modéliser l'interaction entre un fournisseur et des prosommateurs (consommateurs qui peuvent également produire, stocker et vendre de l'électricité), qui interagissent entre eux via un réseau à courant alternatif. Lorsque, avec l'ACOPF, on veut concevoir de manière optimale un réseau de transport d'électricité par rapport à l'activité des lignes, un ACOPF avec des variables on/off sur les lignes peut être utilisé, en obtenant un problème non linéaire en variables mixtes non convexe en nombres complexes. Dans ce scénario, nous proposons deux relaxations convexes, comparées à la célèbre relaxation conique du second ordre de Jabr.

Title : Bilevel optimization and applications

Keywords : mathematical programming, bilevel programming, aircraft conflict resolution, ACOPF, semi-infinite programming

Abstract : A bilevel problem is an optimization problem where a subset of variables is constrained to be optimal for another given problem parameterized by the remaining variables. The outer problem is commonly referred to as the upper-level problem, the inner one as the lower-level problem. The first part of this dissertation concerns the key definitions, the solution approaches and the complexity of bilevel problems, as well as the study of a particular class of bilevel programs, having a quadratic lower level, the value of which is contained into an upper-level inequality constraint. Such bilevel problems can be obtained by reformulating semi-infinite programming problems with an infinite number of quadratically parametrized constraints. We propose an approach to solve this class of bilevel programs, based on the dualization of the lower-level. This approach is compared with a new cutting plane algorithm, which we prove to be convergent. The rate of convergence of this algorithm is derived under stricter assumptions and is directly related to the iteration index, which is something new w.r.t. what is usually proved in semi-infinite programming literature. We successfully test the two proposed methods on two applications: the constrained quadratic regression and a zero-sum game with cubic payoff. The second part of the thesis is devoted to practical applications. A chapter is dedicated to the aircraft conflict resolution problem. This problem essentially consists in enforcing a minimum distance between flying aircraft to avoid conflicts, using different strategies. We focus on two of them: speed regulations and heading angles changes. We present a natural semi-infinite formulation of the problem via speed regulation strategy in k dimensions. To deal with the issue of uncountably many constraints of this formulation, we reformulate it, firstly, using polynomial programming, and secondly, using bilevel programming. Then we also present a bilevel formulation of conflict resolution problem via heading angle changes in two dimensions (i.e. aircraft flying at the same altitude). In both bilevel formulations, the convexity of the lower levels allows us to derive three different single-level reformulations, using KKT conditions, Dorn's duality, and Wolfe's duality respectively. The single-level formulations of both problems are solved by using state-of-the-art solvers. Alternatively, we propose a cut generation algorithm to solve the bilevel problems, which fits in the general framework of the cutting plane algorithm presented in the first part. This algorithm obtains the best results in terms of computational time for most of the tested instances. Another application studied in this dissertation involves the Alternating Current (AC) Optimal Power Flow (ACOPF) problem at the lower level. The idea comes from the possibility for power generation in private households. In this scenario, we derive a bilevel problem to model the interaction between a retailer and several prosumers (consumers who can also produce, store and sell power), who interact with each other through an AC network. When, together with the ACOPF, one wants to optimally design a power transportation network with respect to line activity, an ACOPF with on/off variables on lines can be used, which yields a nonconvex mixed-integer nonlinear problem in complex numbers. We propose two convex relaxations, compared with the famous Jabr's second-order cone relaxation.

Institut Polytechnique de Paris
91120 Palaiseau, France