École polytechnique - École doctorale IP Paris

Laboratoire d'Informatique de l'École polytechnique

With the support of École des ponts and Réseau de transport d'électricité

Global optimization of nonlinear semi-infinite programming problems: Applications in power systems and control.

By Antoine Oustry

PhD Thesis in Computer Science

Under the supervision of Claudia D'Ambrosio and Leo Liberti (CNRS)

Abstract

This thesis deals with the computation of global optima of nonlinear semi-infinite programming problems. These mathematical programming problems are particularly difficult because of the infinite number of constraints and the potential nonconvexity of the objective and the constraints: computing an optimum and certifying its global optimality is a scientific challenge. This thesis makes theoretical and practical contributions to address this challenge, with a focus on applications to the optimization of electrical grids and the control of dynamical systems.

The first part of the thesis is devoted to convex semi-infinite programming. First, we exhibit a convergence rate for the cutting-plane algorithm, also called the adaptive discretization algorithm, when the objective function is strongly convex. This result holds even though one uses an oracle that approximately solves the separation problem, i.e., the problem of finding the most violated constraint, with a given relative accuracy. Second, we tackle a limitation of the cutting-plane algorithm: feasibility is only achieved asymptotically. Focusing on semi-infinite programs with a quadratically constrained quadratic programming separation problem, we propose an iterative inner-outer approximation algorithm that generates a feasible point at each iteration. This sequence of feasible points converges to an optimum of the semi-infinite program, and the convergence also occurs even when the separation problem is solved approximately. We give two sufficient conditions for this algorithm to converge in one iteration to a global optimum. We benchmark this algorithm with three competing approaches on two different applications. The third chapter of this part consists of an original approach to minimal-time nonlinear control based on convex semi-infinite programming. The applicability of our approach is illustrated by numerical experiments on three different non-polynomial dynamical systems.

The second part is concerned with the global optimization of finite and semi-infinite nonconvex optimization problems related to the dispatch of electricity in a power system. First, we tackle the standard AC optimal power flow problem, a quadratically constrained quadratic programming problem with a finite number of constraints. We propose a global optimization algorithm based on a strengthened semidefinite programming relaxation and piecewise linear approximations. This results in an iterative algorithm where the master problems are mixed-integer linear programs. Extensive numerical experiments on a reference benchmark of instances show that this algorithm achieves state-of-the-art performance. Second, we address the numerical issues raised by the semidefinite programming relaxation, the building block of our global optimization algorithm, for instances at a larger scale. We use an unconstrained dual formulation to obtain certified lower bounds. To improve the dual solution returned by an interior-point method, we apply a structured bundle method as a post-processing step. Our numerical experiments on large-scale instances show that this post-processing considerably improves the tightness of the certified dual bounds.

Finally, in the last chapter, we integrate some uncertainties regarding loads and fluctuating generation sources in the AC power flow model: we address the adjustable robust AC optimal power flow problem. We reformulate this adjustable robust optimization problem as an equivalent semi-infinite program. We solve this reformulation with an adaptive discretization algorithm based on a deterministic sampling and a homotopy method. Local and global convergence guarantees are given, depending on whether the master problems are solved locally or globally at each iteration. For any positive feasibility tolerance, finite termination is guaranteed. We provide extensive numerical experiments on small- to large-scale instances.

Keywords: global optimization, semi-infinite programming, semidefinite programming, AC optimal power flow, adjustable robust optimization.

Résumé

Cette thèse traite de la résolution exacte de problèmes de programmation non linéaire semi-infinie. Ces problèmes sont particulièrement difficiles en raison du nombre infini de contraintes et de la potentielle non convexité de la fonction objectif et des contraintes : calculer un optimum et certifier son optimalité globale constituent un défi scientifique. Cette thèse apporte des contributions théoriques et pratiques pour relever ce défi, et met l'accent sur les applications de la programmation semi-infinie à l'optimisation des réseaux électriques et au contrôle des systèmes dynamiques.

La première partie est consacrée à la programmation semi-infinie convexe. Nous démontrons tout d'abord un taux de convergence pour l'algorithme des plans coupants lorsque la fonction objectif est fortement convexe. Ce résultat est valable même si l'oracle utilisé ne résout le problème de séparation, c'est-à-dire le problème de la recherche de la contrainte la plus enfreinte, que de façon approchée avec une précision relative donnée. Dans un deuxième temps, nous cherchons à résoudre un inconvénient majeur de l'algorithme des plans coupants : les points générés par l'algorithme ne sont réalisables qu'asymptotiquement. En nous restreignant aux programmes semi-infinis avec un problème de séparation quadratique sous contraintes quadratiques, nous proposons un algorithme itératif d'approximation interne-externe qui génère une séquence de points réalisables convergeant vers un optimum du problème semi-infini. Nous donnons deux conditions suffisantes pour que l'algorithme converge en une seule itération. Nous comparons cet algorithme à trois approches concurrentes, sur des exemples provenant de deux applications différentes. Dans un chapitre dédié, nous appliquons l'optimisation convexe semi-infinie à la résolution de problèmes de contrôle temps-optimal non linéaires.

La deuxième partie se concentre sur la résolution exacte de problèmes d'optimisation non convexe finis et semi-infinis liés à la répartition des flux de puissance dans un réseau électrique. Premièrement, nous abordons le problème d'optimisation des flux de puissance en courant alternatif (problème ACOPF, pour *AC Optimal Power Flow*), un problème avec un nombre fini de contraintes quadratiques non convexes. Nous proposons un algorithme d'optimisation globale fondé sur la relaxation semi-définie, renforcée par de nouvelles coupes et par des approximations linéaires par morceaux des contraintes non convexes. Il en résulte un algorithme itératif dont le problème maître résolu à chaque itération est un programme linéaire en variables mixtes. L'algorithme obtient des résultats à l'état de l'art sur une librairie d'instances de référence du problème ACOPF. Deuxièmement, nous traitons les difficultés numériques rencontrées par les algorithmes de points intérieurs résolvant la relaxation semi-définie pour des instances de grande taille. Nous proposons une formulation duale sans contrainte pour obtenir des bornes inférieures certifiées. Pour améliorer la solution duale calculée par l'algorithme de points intérieurs, nous exécutons une méthode de faisceau structurée en post-traitement. Nos expériences numériques montrent que ce post-traitement améliore sensiblement la précision des bornes duales obtenues pour des problèmes de grande taille.

Dans le dernier chapitre, nous intégrons au problème ACOPF des incertitudes liées aux consommations et aux productions intermittentes. Le problème d'optimisation robuste avec recours qui en découle est reformulé en un programme semi-infini. Nous résolvons cette reformulation avec un algorithme de discrétisation adaptatif fondé sur un échantillonnage des scénarios et une méthode d'homotopie. Des garanties de convergence locales et globales sont données, selon que les problèmes maîtres sont résolus localement ou globalement à chaque itération. Nous présentons des expériences numériques sur des instances de grande taille.

Mots-clefs: optimisation globale, programmation semi-infinie, programmation semi-définie, optimisation des flux de puissance AC, optimisation robuste avec recours.

Notations

Sets

We denote by $\mathbb{N}$ (respectively $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$) the set of natural numbers (resp. integers, rationals and real numbers). Real intervals are written $[a, b)$, where a bracket means that the bound is included, and a parenthesis means that the bound is excluded. As an example, we define the set of nonnegative numbers $\mathbb{R}_+ = [0, +\infty)$, and the set of positive numbers $\mathbb{R}_{++} = (0, +\infty)$. We also extend the real line by defining $\bar{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$. Integer intervals are written $[\![i, j]\!]$, where both bounds are always included. The Cartesian product of the sets A and B is denoted $A \times B$. By induction, we define $A^k = A \times A^{k-1}$ for all $k \geq 2$, given that $A^1 = A$.

Real and complex numbers

Given a real number $a \in \mathbb{R}$, we denote by $|a|$ its absolute value, $a^+ = \max\{a, 0\}$ its positive part, and $a^- = \max\{-a, 0\}$ its negative part. Given a complex number $a \in \mathbb{C}$, we denote by $|a|$ its modulus, $\mathsf{Re}(a)$ its real part, and $\mathsf{Im}(a)$ its imaginary part, so that $a = \mathsf{Re}(a) + \mathbf{i}\,\mathsf{Im}(a)$. The complex conjugate number of a is $a^* = \mathsf{Re}(a) - \mathbf{i}\,\mathsf{Im}(a)$. We define the partial order $\leq$ for complex numbers as follows: $a \leq b$ if and only $\mathsf{Re}(a) \leq \mathsf{Re}(b)$ and $\mathsf{Im}(a) \leq \mathsf{Im}(b)$.

Linear Algebra

For $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$, a vector $x \in \mathbb{K}^p$ with components x_i is written $x = (x_1, \ldots, x_p)$. We denote by $\|x\|_2$ its Euclidean norm, by $\|x\|_1$ its ℓ_1 norm, by $\|x\|_\infty$ its ℓ_∞ norm. The notation e_j stands for the basis vector with a 1 at position j and zeros elsewhere.
A square matrix $M \in \mathbb{K}^{d \times d}$ is defined by its coefficients $(M_{ij})_{1 \leq i,j \leq d}$. We denote by $M^\top$ its transpose, by M^{H} its transpose conjugate, and M^{-1} its inverse. Its trace is denoted $\mathsf{Tr}(M)$, and its determinant $\det(M)$. A matrix $M \in \mathbb{K}^{d \times d}$ is symmetric if and only if $M = M^{\mathsf{H}}$. In the real case, this means $M = M^\top$, and we denote by $\mathbb{S}_p$ the set of real symmetric matrices. In the complex case, we denote by $\mathbb{H}_p$ the set of symmetric (Hermitian) matrices. We notice that $\mathbb{S}_p \subset \mathbb{H}_p$. For any $M \in \mathbb{H}_p$, we denote by $\lambda_1(M), \lambda_2(M), \ldots \lambda_p(M)$ its eigenvalues, which are real, ordered as follows: $\lambda_{\max}(M) = \lambda_1(M) \geq \lambda_2(M) \geq \cdots \geq \lambda_p(M) = \lambda_{\min}(M)$. For any $M \in \mathbb{H}_p$, we write $M \succeq 0$ if and only if $\lambda_{\min}(M) \geq 0$, i.e., M is Positive Semidefinite (PSD).

We denote by E_{ij} the element of the canonical basis of $\mathbb{K}^{d\times d}$. For $A, B \in \mathbb{K}^{d\times d}$, the Froebenius product is $\langle A, B\rangle = \sum_{1\leq i,j\leq d} A_{ij}B_{ij}^* = \mathsf{Tr}(AB^{\mathsf{H}})$.
We write I_p for the identity matrix of size $p \times p$. For $x \in \mathbb{K}^p$, the notation $\mathsf{diag}(x)$ stands for the diagonal matrix with coefficients $x_1, \dots, x_p$.

Analysis

The set $\Delta_p = \{x \in \mathbb{R}_+^p \colon \sum_{j=1}^p x_j = 1\}$ is the unit simplex of dimension p, and $B(x, r)$ the Euclidean ball of center $x \in \mathbb{R}^p$ and radius $r \in \mathbb{R}_+$. If $S \subset \mathbb{R}^p$ is a set, we define its convex hull as $\mathsf{conv}(S) = \{\sum_{k=1}^r x_k s^k \colon r \in \mathbb{N}, (s^1, \dots, s^r) \in S^r, x \in \Delta_r\}$, and its conic hull as $\mathsf{cone}(S) = \{\sum_{k=1}^r x_k s^k \colon r \in \mathbb{N}, (s^1, \dots, s^r) \in S^r, x \in \mathbb{R}_+^r\}$. For any nonempty set A, and any $x \in \mathbb{R}^p$, we denote $d(x, A) = \inf_{a\in A} \|x - a\|_2$ the distance between the set A and the point x. The Hausdorff distance between two sets A and B is $d_H(A, B) = \max\{\sup_{b\in B} d(b, A), \sup_{a\in A} d(a, B)\}$.
For any nonempty set A, the contingent cone to A at $x \in A$, denoted $T_A(x)$, is the set of directions $d \in \mathbb{R}^p$, such that there exist a sequence $(t_k) \in \mathbb{R}_{++}^{\mathbb{N}}$, and a sequence $(d_k) \in (\mathbb{R}^p)^{\mathbb{N}}$, satisfying $t_k \to 0$, $d_k \to d$, and $x + t_k d_k \in A$, for all $k \in \mathbb{N}$.
A property P holds "almost everywhere" (a.e.) on A, or equivalently "for almost all $x \in A$", if there exists a set N of Lebesgue measure zero such that the property P holds for all $x \in A \setminus N$.
For any $p \in \mathbb{N}^*$, and $k \in \mathbb{N} \cup \{\infty\}$, we denote by $C^k(\mathbb{R}^p) = C^k(\mathbb{R}^p, \mathbb{R})$ the vector space of real-valued functions with k continuous derivatives over $\mathbb{R}^p$. For any differentiable function $f \colon \mathbb{R}^p \to \mathbb{R}^q$, we denote by $\nabla f(x) \in \mathbb{R}^{p\times q}$ its Jacobian matrix (its gradient, if $q = 1$). We denote by $\partial_i f$ the partial derivative with respect to x_i. In the function f is convex, the notation $\partial f(x)$ designates the subdifferential of f at x. For any locally Lipschitz function f, we denote by $\partial^c f$ its Clarke's generalized derivative [51]. If the function is convex, this coincides with its subdifferential [51].
For a given set $A \subset \mathbb{R}^p$, for any function $f \in C^k(\mathbb{R}^p)$, we denote by $f_{|A}$ the restriction of f on A; moreover, we define the vector space $C^k(\mathbb{R}^p|A) = \{f_{|A} \colon f \in C^k(\mathbb{R}^p)\}$ of restrictions on A of C^k functions.
For any two functions f, g Lebesgue integrable over $\mathbb{R}^p$, we define the convolution product $f \star g$ as $f \star g(x) = \int_{\mathbb{R}^p} f(x)g(x - h)dh$. We emphasize that this convolution product is also well defined if f is supported on a compact set, and g is locally integrable.
We denote by $\mathbb{R}[x_1, \dots x_p]$ the vector space of real multivariate polynomials with variables $x_1, \dots, x_p$, and $\mathbb{R}_d[x_1, \dots x_p]$ the vector space of real multivariate polynomials with degree at most d.

Glossary

ACOPF Alternating Current Optimal Power Flow. 1, 5–12, 75–78, 83, 85, 86, 92, 93, 97, 99, 102–104, 108, 116, 119, 121, 124, 136, 144, 146–148, 173, 177

ADMM Alternating Direction Method of Multipliers. 103

ARO Adjustable Robust Optimization. 5, 12, 121–123

AR-OPF Adjustable Robust AC Optimal Power Flow. 10–12, 122–125, 136, 143, 144, 146–149

B&B Branch-and-Bound. 9, 77

CC-OPF Chance-Constrained AC Optimal Power Flow. 10

CPA Cutting-plane algorithm. 3, 11, 15–17, 20–22, 25, 27–29, 40–48, 57, 114, 115

ESIP Existence-constrained Semi-Infinite Programming. 121–124

FBBT Feasibility-Based Bound Tightening. 76, 85, 86, 92, 93, 96

FCFW Fully Corrective Frank–Wolfe. 21, 22

GO Global Optimization. 6, 8–11

HJB Hamilton-Jacobi-Bellman. 47–52, 54, 55, 58, 64–66, 72, 73

IOA Inner-Outer Approximation algorithm. 11, 27, 29, 35, 36, 40–46

IPM Interior-Point Methods. 4, 5, 8, 9, 76, 103, 116

KKT Karush–Kuhn–Tucker. 4, 8, 9, 28, 41, 122

LMI Linear Matrix Inequalities. 107

LP Linear Programming. 9, 28, 48, 76, 86–89, 92, 93

MILP Mixed-Integer Linear Programming. 11, 77, 86, 89–100, 146, 148, 149

MINLP Mixed-Integer Nonlinear Programming. 2, 122

MIP Mixed-Integer Programming. 77

MIQCP Mixed-Integer Quadratically Constrained Programming. 77

NLP Nonlinear Programming. 6, 76, 85, 91–93, 95

NSO Nonsmooth Optimization. 112, 114

OBBT Optimization-Based Bound Tightening. 76, 85, 86, 92, 93, 96, 97
OCP Optimal Control Problems. 47, 48, 50
ODE Ordinary Differential Equation. 128–130

PBM Proximal Bundle Methods. 112, 114, 116, 118, 119
PDE Partial Differential Equation. 50, 51
PSD Positive Semidefinite. v, 29, 30, 32, 33, 43, 45, 79, 105, 106, 116

QCQP Quadratically Constrained Quadratic Programming. 7, 9, 11, 27–30, 42, 46, 76, 77, 87, 97, 136, 146, 177
QP Quadratic Programming. 57, 115, 116

RLT Reformulation-Linearization Technique. 76, 97
RO Robust Optimization. 5

SC-OPF Security-Constrained AC Optimal Power Flow. 10, 123
SDP Semidefinite Programming. 5, 9, 11, 12, 29–32, 34, 46, 76–79, 82, 83, 87, 97, 99, 102, 103, 106, 107, 112, 116
SOCP Second-Order Cone Programming. 9, 76, 77, 87, 97
SoS Sum-of-Squares. 9, 28, 48, 68, 69, 76, 77
SQP Sequential Quadratic Programming. 4, 9

Table of Contents

Introduction

Semi-infinite programming is a branch of mathematical programming interested in optimization problems with an infinite number of constraints. Computing a vector that is a solution to an infinite system of inequations and, subsequently, certifying that the inequations are indeed satisfied, are challenging tasks. Despite this difficulty and the potential nonconvexity of these semi-infinite problems, our goal is to produce algorithms capable of computing globally optimal solutions. After an introduction to semi-infinite programming, we present the Alternating Current Optimal Power Flow (ACOPF) problem, an optimization problem arising from electrical engineering that is the main application we focus on in this dissertation. After addressing the standard ACOPF formulation with a finite number of constraints, we also study an application-driven variant with an infinite number of constraints.

A. Introduction to semi-infinite programming

First, we introduce the typical formulation in semi-infinite programming, and we define some central concepts for this field of study. We present the adaptive discretization algorithm, one of the standard semi-infinite programming algorithms, and we introduce the reader to several applications. For a comprehensive introduction to semi-infinite programming, the reader is referred to [84, 104].

The semi-infinite formulation and the oracle framework

Problem definition. Given two integers $m, n \in \mathbb{N}^*$, we consider two continuous functions $F\colon \mathbb{R}^m \to \mathbb{R}$ and $G\colon \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$. We also consider two nonempty and compact sets $\mathcal{X} \subset \mathbb{R}^m$, and $\mathcal{Y} \subset \mathbb{R}^n$. We define the semi-infinite programming problem (**SIP**) as

$$\left.\begin{array}{ll} \min\limits_{x \in \mathcal{X}} & F(x) \\ \text{s.t.} & \forall y \in \mathcal{Y},\ G(x,y) \leq 0. \end{array}\right\} \qquad (\textbf{SIP})$$

In this formulation, we refer to x as the *decision vector* and to y as the *parameter vector*. Using a terminology from bilevel optimization [57], we may also refer to x as the vector of *upper-level variables* and to y as the vector of *lower-level variables*. We now detail the connection between semi-infinite and bilevel programming.

Lower-level problem and value function. For a given $x \in \mathcal{X}$, the problem of finding the parameter vector $y \in \mathcal{Y}$ that maximizes $G(x, y)$ is called the *lower-level problem*, or equivalently, the *separation problem*. We introduce $\phi(x)$ the value function of the lower-level problem:

$$\phi(x) = \max_{y \in \mathcal{Y}} G(x, y). \tag{1}$$

Proposition 0.1. *The function $\phi(x)$ is continuous over $\mathbb{R}^m$.*

Proof. This is a direct application of the Maximum Theorem [7, Th. 2.1.6], in so far as (i) G is continuous, therefore lower and upper semicontinuous, (ii) the set-valued map $\tilde{\mathcal{Y}}(x) = \mathcal{Y}$ is compact valued and lower and upper semicontinuous (since it is constant). We deduce that $\phi(x)$ is lower and upper semicontinuous, therefore continuous. □

Having defined the function $\phi(x)$, the semi-infinite programming problem (**SIP**) also reads

$$\left.\begin{array}{ll} \min\limits_{x \in \mathcal{X}} & F(x) \\ \text{s.t.} & \phi(x) \leq 0. \end{array}\right\} \tag{2}$$

However, it should be noted that in most cases there is no closed-form expression for the function $\phi(x)$, this is why an optimization algorithm is necessary to compute it.

Separation oracle. We call *separation oracle* any optimization algorithm that solves the lower-level problem (1). Given that this optimization problem can be computationally hard, it is not necessarily realistic to assume that the oracle solve it exactly. Therefore, we consider a separation oracle that computes a feasible solution to the problem (1) with a relative optimality gap of $\delta \in [0, 1)$. More precisely, for any $x \in \mathcal{X}$, this black-box algorithm computes $\hat{y}(x) \in \mathcal{Y}$, and an upper bound $\hat{v}(x) \geq \phi(x)$ such that

$$\hat{v}(x) - G(x, \hat{y}(x)) \leq \delta\, |\phi(x)|. \tag{3}$$

Consequently, the following inequalities hold:

$$\phi(x) - \delta\, |\phi(x)| \leq G(x, \hat{y}(x)) \leq \phi(x) \leq \hat{v}(x) \leq \phi(x) + \delta\, |\phi(x)|. \tag{4}$$

Throughout the thesis, we will use the term of "δ-oracle" to designate such a separation oracle with *limited relative accuracy*. Alternatively, we will also say that the oracle has a parameter $\delta \in [0, 1)$. We point out that $\delta = 0$ corresponds to the case of a perfectly accurate oracle. In practice, a δ-oracle can be implemented by different types of methods depending on the nature of the problem: local or global optimization solvers for Mixed-Integer Nonlinear Programming (MINLP), positivity certificates for polynomials, combinatorial optimization algorithms, interval arithmetics, constraint programming, or sampling for example, as long as we obtain not only an approximate solution $\hat{y}(x)$, but also the upper bound $\hat{v}(x)$. In this dissertation, we consider the oracle as a black box, without making any assumptions about $\hat{y}(x)$ and $\hat{v}(x)$ other than those in Eqs. (3)-(4).

A standard algorithm: the adaptive discretization

We present the adaptive discretization algorithm because it is a standard approach to semi-infinite programming and because it recurs throughout this dissertation in different contexts and with several variants. This algorithm was introduced in 1976 by Blankenship and Falk as the *generalized cutting-plane algorithm* [31]. The term "generalized" is used insofar as this algorithm can be seen as an extension of the Cutting-plane algorithm (CPA) for convex optimization [118]; we will prefer here the term "adaptive discretization" to make a clear distinction. This algorithm (see Algorithm 1) successfully solves relaxations obtained by imposing only finitely many constraints. At iteration k of this algorithm, the infinite set of constraints $\mathcal{Y}$ is replaced by a finite subset $\mathcal{Y}^k \subset \mathcal{Y}$, that is progressively enlarged with the output of the separation oracle. We recall that this oracle may have limited relative accuracy.

Algorithm 1 Adaptive discretization algorithm for (**SIP**) with limited-accuracy oracle

```
    **Input:** Oracle with parameter $\delta \in [0,1)$, tolerance $\epsilon \in \mathbb{R}_+$, finite set $\mathcal{Y}^0 \subset \mathcal{Y}$
0:  Let $k \leftarrow 0$, $\nu_0 \leftarrow \infty$.
1:  **while** $\nu_k > \epsilon$ **do**
2:      Compute $x^k$ an optimal solution of the relaxation
            $\min_{x \in \mathcal{X}} F(x)$ s.t. $\forall y \in \mathcal{Y}^k,\ G(x,y) \le 0.$      $(\mathbf{R_k})$
3:      Call the $\delta$-oracle to compute $y^k = \hat{y}(x^k)$ an approximate solution of
            $\max_{y \in \mathcal{Y}} G(x^k, y).$
4:      $\mathcal{Y}^{k+1} \leftarrow \mathcal{Y}^k \cup \{y^k\}$, $\nu_{k+1} \leftarrow G(x^k, y^k)$, $k \leftarrow k+1$.
5:  **end while**
6:  Return $x^k$.
```

We present some convergence results, the proof of which are presented in Appendix A. The first lemma states the asymptotic feasibility of the iterates generated by the Algorithm 1.

Lemma 0.1. *Consider a parameter $\delta \in [0,1)$, infinite sequences $(x^k)_{k\in\mathbb{N}} \subset \mathcal{X}$ and $(y^k)_{k\in\mathbb{N}} \subset \mathcal{Y}$, where $y^k = \hat{y}(x^k)$ is the output of the δ-oracle evaluated at point x^k. If, for any $k \in \mathbb{N}$, for any $\ell \in [\![0, k-1]\!]$, $G(x^k, y^\ell) \le 0$, then, the feasibility error $\phi(x^k)^+$ vanishes: $\phi(x^k)^+ \to 0$.*

The following theorem states the finite convergence of the algorithm, if the tolerance ϵ is positive, and the asymptotic convergence towards a global minimum, if not.

Theorem 0.1. *If $\epsilon \in \mathbb{R}_{++}$, Algorithm 1 stops after a finite number of iterations. On the contrary, if $\epsilon = 0$ and Algorithm 1 generates an infinite sequence of iterates $(x^k)_{k\in\mathbb{N}}$, then any limit value $x \in \mathbb{R}^m$ of $(x^k)_{k\in\mathbb{N}}$ is an optimal solution in* (**SIP**).

This last theorem states the optimality and feasibility errors of the output of the algorithm, depending on the tolerance ϵ and the accuracy parameter δ of the oracle.

Theorem 0.2. *If Algorithm 1 terminates after K iterations, the iterate $x^K \in \mathcal{X}$, satisfies $G(x^K, y) \leq \frac{\epsilon}{1-\delta}$, for all $y \in \mathcal{Y}$, and has value $F(x^K) \leq \mathsf{val}(\textbf{SIP})$.*

Overview of the semi-infinite programming literature

After this description of the adaptive discretization algorithm, we propose an overview of other algorithms for semi-infinite programming, leaving a more detailed presentation of the relevant chapters. Note that these algorithms are not stand-alone numerical methods but rely on the ability to solve some finite optimization problems.

Discretization methods. The above mentioned “adaptive” discretization exists by opposition with “fixed” discretization approaches [103, 195, 207], where $\mathcal{Y}$ is approximated by a predefined grid $\hat{\mathcal{Y}}$, yielding relaxations of the original problem. The convergence of the solution of the relaxation is proven as a function of the Haussdorff distance between $\mathcal{Y}$ and $\hat{\mathcal{Y}}$ [207]. Back to the adaptive discretization, a limitation of Algorithm 1 is that the feasibility of the iterates is obtained asymptotically only; some variant of the algorithm were introduced to compute feasible points after a finite number of iterations [65, 161, 218]. In the setting of convex semi-infinite programming, some central cutting-plane algorithms were proposed [24, 89, 104], to improve the stability and the speed of convergence of the adaptive discretization. In this convex setting, the *exchange* algorithms [32, 243], are a bounded-memory variants of the cutting-plane algorithm.

Overestimation methods. These methods replace the lower-level value function $\phi(x)$ by an overestimation to obtain a finite restriction of the original semi-infinite program. The overestimation may be obtained through a convex relaxation of the lower-level problem. The relaxed lower-level problem is, then, equivalently replaced by its Karush–Kuhn–Tucker (KKT) conditions, resulting in a restriction of the original problem that is nonconvex problem, with the complementarity constraints [74, 205]. The overestimation may also be obtained by dualization of the lower-level problem [18, 64, 132, 142], or based on interval arithmetics [25, 157].

Other methods. The KKT approach directly replaces the lower-level problem by its KKT conditions to obtain a finite formulation [2]. Again, the main drawback of this approach is to result in a nonconvex formulation, even if the original semi-infinite program is convex. *Local reduction* methods exploit the sensitivity of the value and the solution of the lower-level problem (1) with respect to x, to apply nonlinear differentiable optimization algorithms such as Sequential Quadratic Programming (SQP), Interior-Point Methods (IPM) or the projected augmented Lagrangian algorithm [58, 104, 186, 214]. An obstacle to the scalability of the local reduction methods is the need to compute all the (globally) optimal solutions of the lower-level problem (1), which can be highly costly if this problem is nonconvex.

Applications of semi-infinite programming

The study of semi-infinite programming is motivated by its various applications. From a theoretical point of view, we first consider the connections with other classes of mathematical optimization problems. The framework of semi-infinite programming includes min-max optimization problems, also known as zero-sum games in game theory [242]. In Chapter 2, we present one such application. We can also think of Semidefinite Programming (SDP) as a particular case of semi-infinite programming [222]. Certainly, specialized methods for SDP, mainly IPM, are particularly efficient [1, 100, 168] and do not rely on semi-infinite programming. However, Chapter 5 illustrates the interest of the semi-infinite programming view for large-scale SDP problems. Robust Optimization (RO), which models optimization problems under non-probabilistic uncertainty [16], is also covered by the semi-infinite programming framework [67]. In Chapter 6, we apply semi-infinite programming to solve an Adjustable Robust Optimization (ARO) problem [239], where the decision variables are separated into "here-and-now" decisions (before the uncertainty realization) and "wait-and-see" decisions (after the uncertainty realization). One of the advantages of semi-infinite programming is the ability to combine the different paradigms mentioned above: this framework makes it possible, for example, to model the robust solution of SDP problems [69].

From a more practical point of view, we can envision several applications of semi-infinite programming in various fields arising from situations where an infinite number of constraints must be satisfied. One of the first problems that motivated the study of semi-infinite optimization is the Chebyshev approximation problem [153], where the best approximation of a given continuous function is searched among a parameterized family of functions. In chemical engineering, model reductions are approximation problems that can be formulated as semi-infinite programs [26]. Another standard application is the task of design centering [190, 204], where a parameterized body should be inscribed into a container. Trajectory planning, in robotics [98] or in air traffic control [44], is another area of application, as it involves a continuous description of time. More generally, semi-infinite programming can be used in optimal control, as illustrated by Chapter 3 of this dissertation. We can mention further applications in chemical engineering [210] again, in statistics [90], in economics [228], in finance [128], in industrial process optimization [235], and in structural design [3], only to mention a few.

Due to the scientific, economic, and societal challenges associated with the energy transition, electrical engineering is one of the most crucial areas of application for mathematical optimization and, more specifically, for robust and semi-infinite optimization. We now present the ACOPF problem in more detail, an optimization problem that we focused on during this thesis.

B. Introduction to AC power flow optimization

The main application we consider in this dissertation is the Alternating-Current Optimal Power Flow (ACOPF). This is a widely studied optimization problem related to dispatching electricity in a power network. The authorship of this problem is commonly attributed to Carpentier, who introduced it in 1962 in a seminal paper [42] as "Economic Dispatch". Since then, this problem has not only interested the electrical engineering community [37, 43], but also the community of operations research and mathematical programming [27, 117, 165, 233]. Indeed, ACOPF was identified as a challenging application of Nonlinear Programming (NLP) and Global Optimization (GO).

The standard ACOPF problem, presented in this section, is a deterministic NLP problem with a finite number of constraints: as such, it does not fit directly into the category of semi-infinite programming problems. Nonetheless, our developments show that semi-infinite programming provides a fruitful approach to solving the ACOPF problem (see Chapter 4). Moreover, we also employ the semi-infinite programming framework to define and solve a variant of this problem with robust feasibility guarantees in the presence of uncertainties (see Chapter 6).

Mathematical programming formulation of the ACOPF problem

Graph, sets and parameters. An electrical grid is a network of buses interconnected by lines. It can be modeled as an oriented graph $\mathcal{N} = (\mathcal{B}, \mathcal{L})$. A line $\ell \in \mathcal{L}$ is described by a couple (b, a) such that $b \in \mathcal{B}$ is the "from" bus (denoted by f), $a \in \mathcal{B}$ is the "to" bus (denoted by t). The set $\mathcal{L}$ is such that $\mathcal{L} \cap \mathcal{L}^R = \emptyset$, where $\mathcal{L}^R$ is the set of couples (b, a) such that $(a, b) \in \mathcal{L}$. Electricity generating units are located at several buses in the network. We denote by $\mathcal{G}_b$ the set of generators located at bus $b \in \mathcal{B}$. The set of all generators is $\mathcal{G} = \cup_{b\in\mathcal{B}} \mathcal{G}_b$. The parameters of the ACOPF problem are described in Table 0.1. More details about their physical meaning are given in Appendix B.

Parameters	Index set	Meaning
$c_{0g}, c_{1g} \in \mathbb{R}, c_{2g} \in \mathbb{R}_+$	$g \in \mathcal{G}$	Generator's cost parameters
$\underline{s}_g, \overline{s}_g \in \mathbb{C}$	$g \in \mathcal{G}$	Generator's domain bounds
$\underline{v}_b, \overline{v}_b \in \mathbb{R}_+$	$b \in \mathcal{B}$	Voltage magnitude bounds
$S^{\mathsf{d}}_b \in \mathbb{C}$	$b \in \mathcal{B}$	Power demand
$Y^s_b \in \mathbb{C}$	$b \in \mathcal{B}$	Shunt admittance
$Y^{\mathsf{ff}}_{ba}, Y^{\mathsf{ft}}_{ba}, Y^{\mathsf{tf}}_{ba}, Y^{\mathsf{tt}}_{ba} \in \mathbb{C}$	$(b, a) \in \mathcal{L}$	Line impedance matrix

Table 0.1: Main parameters of the ACOPF problem

Regarding the parameter $\underline{s}_g$ (resp. $\overline{s}_g$), we denote by $\underline{P}_g$ and $\underline{Q}_g$ (resp. $\overline{P}_g$ and $\overline{Q}_g$) its real and imaginary part. We denote by P^{d}_b and Q^{d}_b the real and imaginary part of S^{d}_b.

Problem definition. Depending on the authors, the ACOPF problem is introduced with different notations and formulations in polar or cartesian coordinates. We privilege here the quadratic formulation in complex numbers, yielding a compact presentation.

$$\left.\begin{array}{lll}\min\limits_{V\in\mathbb{C}^{\mathcal{B}},\, S\in\mathbb{C}^{\mathcal{G}}} & \sum\limits_{g\in\mathcal{G}}\Big(c_{0g}+c_{1g}\,\mathsf{Re}(S_g)+c_{2g}\,\mathsf{Re}(S_g)^2\Big) & \\ \text{s.t. } \forall b\in\mathcal{B}, & \underline{v}_b\leq|V_b|\leq\overline{v}_b & \\ \quad\forall g\in\mathcal{G}, & \underline{s}_g\leq S_g\leq\overline{s}_g & \\ \quad\forall b\in\mathcal{B}, & \sum\limits_{g\in\mathcal{G}_b}S_g-S^{\mathsf{d}}_b=(Y^s_b)^*|V_b|^2 & +\sum\limits_{a:(b,a)\in\mathcal{L}}\Big((Y^{\mathsf{ff}}_{ba})^*|V_b|^2+(Y^{\mathsf{ft}}_{ba})^*V_bV_a^*\Big) \\ & & +\sum\limits_{a:(b,a)\in\mathcal{L}^R}\Big((Y^{\mathsf{tt}}_{ab})^*|V_b|^2+(Y^{\mathsf{tf}}_{ab})^*V_bV_a^*\Big)\end{array}\right\}\quad\text{(ACOPF)}$$

In this Quadratically Constrained Quadratic Programming (QCQP) with complex variables, the decision vector $V\in\mathbb{C}^{\mathcal{B}}$ corresponds to the voltages of the buses; the decision vector $S\in\mathbb{C}^{\mathcal{G}}$ corresponds to the power injections of the generators. The constraints of the first type are the voltage magnitude limits at each bus. The constraints of the second type are the generator power limits: note that the order $z_1\leq z_2$ means $\mathsf{Re}(z_1)\leq\mathsf{Re}(z_2)$ and $\mathsf{Im}(z_1)\leq\mathsf{Im}(z_2)$. Most importantly, the third type of constraint, named "power flow equations", corresponds to power conservation. In this dissertation, we use the following matrix notation: for each bus $b\in\mathcal{B}$, we define

$$M_b=Y^s_bE_{bb}+\sum_{a:(b,a)\in\mathcal{L}}\left(Y^{\mathsf{ff}}_{ba}E_{bb}+Y^{\mathsf{ft}}_{ba}E_{ba}\right)+\sum_{a:(b,a)\in\mathcal{L}^R}\left(Y^{\mathsf{tt}}_{ab}E_{bb}+Y^{\mathsf{tf}}_{ab}E_{ba}\right), \tag{5}$$

recalling that the matrices E_{ba} are the elements of the canonical basis of $\mathbb{C}^{\mathcal{B}\times\mathcal{B}}$. Then, the power conservation at $b\in\mathcal{B}$ reads

$$\sum_{g\in\mathcal{G}_b}S_g-S^{\mathsf{d}}_b=\left\langle M_b,VV^{\mathsf{H}}\right\rangle. \tag{6}$$

Appendix B details how we derive these mathematical equations from Ohm's and Kirchhoff's laws. In particular, we also explain why these power flow equations involve complex numbers, which is unusual for a mathematical optimization problem.

Additional constraints. In addition, some authors consider capacity constraints regarding the electrical lines. We exhibit the corresponding parameters in Table 0.2.

Parameters	Index set	Meaning
$\overline{S}_{ba}\in\mathbb{R}_+$	$(b,a)\in\mathcal{L}\cup\mathcal{L}^R$	Power magnitude limit
$\overline{I}_{ba}\in\mathbb{R}_+$	$(b,a)\in\mathcal{L}\cup\mathcal{L}^R$	Current magnitude limit
$\underline{\theta}_{ba},\overline{\theta}_{ba}\in[-\frac{\pi}{2},\frac{\pi}{2}]$	$(b,a)\in\mathcal{L}\cup\mathcal{L}^R$	Angle difference limits

Table 0.2: Additional parameters for the ACOPF problem

- **Line capacities**, either in power magnitude, or in current magnitude:
  - Limits in power magnitude:

$$\forall (b,a) \in \mathcal{L},\ |(Y^{\mathsf{ff}}_{ba})^*|V_b|^2 + (Y^{\mathsf{ft}}_{ba})^* V_b V_a^*| \leq \overline{S}_{ba} \tag{7}$$
$$\forall (b,a) \in \mathcal{L}^R,\ |(Y^{\mathsf{tt}}_{ab})^*|V_b|^2 + (Y^{\mathsf{tf}}_{ab})^* V_b V_a^*| \leq \overline{S}_{ba} \tag{8}$$

  - Limits in current magnitude:

$$\forall (b,a) \in \mathcal{L},\ |Y^{\mathsf{ff}}_{ba} V_b + Y^{\mathsf{ft}}_{ba} V_a| \leq \bar{I}_{ba}, \tag{7bis}$$
$$\forall (b,a) \in \mathcal{L}^R,\ |Y^{\mathsf{tt}}_{ab} V_b + Y^{\mathsf{tf}}_{ab} V_a| \leq \bar{I}_{ba}. \tag{8bis}$$

- **Angle difference limits**:

$$\forall (b,a) \in \mathcal{L} \cup \mathcal{L}^R,\ \underline{\theta}_{ba} \leq \angle V_b - \angle V_a \leq \overline{\theta}_{ba} \tag{9}$$

Complexity. The feasibility problem associated with ACOPF has been proven to be strongly NP-Hard [30], which underlines the computational difficulty of ACOPF as a GO problem. That said, it has also been shown that polynomial optimization problems with a constraints system of bounded tree-width can be solved in polynomial time to any desired accuracy [29]. In practice, ACOPF instances indeed have low tree-widths.

Overview of the ACOPF literature

We present succinctly the main categories of algorithms to solve the ACOPF problem, keeping a more detailed presentation and comparison of these methods for the relevant chapters.

Power flow algorithms. In the power flow equation (6), fixing at each bus the real and imaginary parts of the power injection (PQ), or the real part of the power injection and the voltage magnitude (PV), results in a quadratic system of nonlinear equations. In the power system community, the problem of solving these power flow equations with known injections is called the "load flow." Several methods can solve it numerically, mainly based on the Newton-Raphson algorithm [78, 159, 208].

Local optimization algorithms. Some algorithms are designed to efficiently compute feasible points satisfying local optimality conditions for large-scale nonlinear optimization problems. On the one hand, an IPM solves, by means of Newton's algorithm, a sequence of KKT systems of optimization problems where the inequalities are replaced by a logarithmic barrier with a penalty factor. This barrier parameter is progressively decreased to converge to a solution of the KKT system of the original problem: this is a homotopy approach. General differentiable optimization solvers such as `Ipopt` [238] or `Artelys Knitro` [36], which are the state-of-the-art local solvers for the ACOPF problem [116], implement such an IPM. On the other hand, SQP methods iteratively use quadratic approximations of the objective function and constraints to obtain a subproblem, which is then solved using a quadratic programming solver. Under mild assumptions, such an algorithm also converges to a feasible point that satisfies the KKT conditions. SQP implementations include `Artelys Knitro` [36], `SNOPT` [82], or `NLOpt` [114]. In practice, these methods often converge to a global optimum without guaranteeing that it is [85, 116].

Formulating and solving convex relaxations. To evaluate the quality of the feasible solutions computed with local optimization algorithms, one needs lower bounds on the ACOPF optimal value. This is possible through the formulation and the solution of convex relaxations of the ACOPF problem [165]. The main families of relaxation are Second-Order Cone Programming (SOCP) relaxations [109, 122], convex QCQP relaxations [54, 55, 85], and SDP relaxations, which include the rank relaxation [135] (also known as Schor relaxation) and the Sum-of-Squares (SoS) hierarchy [117, 133]. In the SoS hierarchy, the values of the relaxations converge towards the value of the ACOPF problem at a computational price that drastically increases with the number of variables and with the degree of the relaxation. Several strategies exist to exploit the sparse structure of the optimization problem (correlative sparsity, term sparsity) and obtain a more tractable relaxation [131, 234, 232]. Most of the convex relaxation approaches leverage efficient and accurate conic optimization solvers, such as `MOSEK` [168] or `SeDuMi` [211], although these solvers may encounter numerical troubles for large-scale problems, as illustrated in Chapter 5. The reader is referred to this chapter for a more detailed overview of the state-of-the-art to solve SDP relaxations for the ACOPF problem.

Global optimization algorithms. Apart from the SoS hierarchy [117], none of the above relaxations is proven to be tight in general, i.e., to have the same value as the ACOPF problem. Except in some specific cases [77, 151], these convex relaxations may not provide a feasible solution to the ACOPF problem. Similarly, IPM or SQP may only provide a local stationary point [35]. Therefore, computing a global minimum may require subdividing and exploring the space of variables. Except for the Moment-SoS hierarchy, most of the algorithms with global convergence guarantees implement a spatial Branch-and-Bound (B&B) algorithm built upon a SOCP, convex QCQP, or SDP relaxation. General purpose solvers for nonconvex GO like `SCIP` [225], or for nonconvex QCQP like `Gurobi`[94], which can handle the ACOPF formulation presented above, are mainly based on Linear Programming (LP) relaxations. A detailed description of the state-of-the-art global optimization algorithms for ACOPF is given in Chapter 4.

Variants of the optimal power flow problem

The formulation ACOPF presented in this introduction is a stylized description of a decision problem faced by a power system operator. It is a rich scientific challenge in terms of GO, but it needs to be refined to be more relevant from an operational point of view.

A critical refinement of the ACOPF problem is the consideration of uncertainties [197]. In power systems, optimization is used for real-time operation and long-term planning, but decision-makers rely on uncertain input data to make optimal decisions. For example, these uncertain parameters include load and renewable energy forecasts. Taking into account this uncertainty on loads in the ACOPF problem leads to the Chance-Constrained AC Optimal Power Flow (CC-OPF) problem [28, 62, 169, 223] if the existence of a solution is to be guaranteed with a certain probability. This leads to the Adjustable Robust AC Optimal Power Flow (AR-OPF) [129, 137, 150], if the feasibility has to be guaranteed, thanks to some recourse variables arising in a second stage, for all the possible realizations of the uncertainty: this is a two-stage optimization problem. In addition to the uncertain parameters, taking into account some contingencies, such as the loss of a line or a generator, is crucial to avoid cascading failures leading to blackouts. The Security-Constrained AC Optimal Power Flow (SC-OPF) problem, still in a two-stage optimization approach, aims to find an optimal operating point that can survive, with limited disturbances, to one or more contingencies thanks to recourse variables [72, 237]. If the recourse variables correspond only to state variables of the network, we speak of *preventive* SC-OPF. If they also correspond to corrective actions, we speak of *corrective* SC-OPF [120]. The purpose of Chapter 6, is indeed to address the preventive AR-OPF and SC-OPF problems using the semi-infinite programming framework.

Another axis of refinement of the ACOPF problem is the consideration of the underlying dynamics of generators, which, coupled by the electrical network, form a large-scale dynamic system. In the standard (static) ACOPF problem, we search for an equilibrium point that minimizes a certain cost without considering the dynamics of the network. However, it would be relevant to consider stability constraints, i.e., to look for an equilibrium point that is stable even in the presence of disturbances. Moreover, it is also necessary to address the target equilibrium point's reachability given the network's initial state. Although there is some preliminary work to consider stability criteria in the ACOPF, their applicability is still limited to small instances [61, 215, 230, 236]. Regarding the study of reachable sets of dynamical systems in power networks, the state-of-the-art is also limited to the study of small systems [174, 181, 213], and the inclusion of such constraints in the ACOPF problem is still to be done, to the best of our knowledge.

C. Thesis structure and summary of results

Part I of the thesis is devoted to convex semi-infinite programming and its applications.

Chapter 1. We study the application of Algorithm 1 to convex semi-infinite programs. In this context, this algorithm is equivalent to the CPA, also known as Kelley's algorithm. Assuming the strong convexity of the objective and the existence of a strictly feasible point, we establish that the CPA algorithm and some variants converge at a rate of $O(\frac{1}{k})$ for the objective and for the feasibility error, where k is the number of calls to the separation oracle.

Chapter 2. This chapter deals with a special type of convex semi-infinite programs, where the lower-level problem (1) is a QCQP, which is potentially nonconvex. This chapter addresses a limitation of the CPA algorithm, which is that feasibility occurs only asymptotically. We design an algorithm called Inner-Outer Approximation algorithm (IOA), that generates a sequence of feasible iterations that converge to the optimal value of the semi-infinite problem. We demonstrate the interest of this algorithm by experimental comparison with three other algorithmic approaches for several instances stemming from two different applications.

Chapter 3. As an application, we solve minimum-time nonlinear control problems with a semi-infinite programming approach. We propose a hierarchy of convex semi-infinite programming restrictions of the dual problem with convergence guarantees. Based on this hierarchy, we derive an algorithmic approach to compute guaranteed lower and upper bounds for minimum time control problems and test it on three non-polynomial control problems.

Part II deals with nonconvex optimization problems arising in power systems. This part is dedicated to the global solution of the ACOPF and the AR-OPF problems. The latter problem fits the setting of nonconvex semi-infinite programming.

Chapter 4. We propose a method for solving the AC Optimal Power Flow problem globally. The approach involves adding novel valid inequalities to strengthen the SDP relaxation, resulting in a conic programming relaxation. These inequalities dominate other inequalities from the literature used for the same purpose. Additionally, a GO algorithm is proposed that employs a sequence of Mixed-Integer Linear Programming (MILP) relaxations, with linear cuts stemming from the conic constraints. The integration of these cuts, derived from the conic relaxation, may be seen as a semi-infinite discretization scheme, as presented in the Introduction. The algorithm presents global convergence guarantees. We apply it to the IEEE PES PGLib benchmark [9] and compare it with other recent GO methods.

Chapter 5. For large-scale ACOPF problems, the conic programming solvers, such as the one used in the previous chapter, may encounter numerical difficulties in solving the SDP relaxation. To address these numerical problems, this chapter introduces a novel approach to computing SDP bounds of the ACOPF problem. This approach involves an unconstrained formulation for the Lagrangian dual of the problem. Using this formulation, we show how to derive a certified lower bound from any dual vector, feasible in the classical dual problem or not. This unconstrained optimization problem is solved using a polyhedral bundle method that exploits the structure of the problem. The experiments on the IEEE PES PGLib benchmark [9] illustrate the advantages of this approach in terms of the accuracy of the computed certified lower bounds.

Chapter 6. This chapter deals with nonlinear ARO problems, where the number of recourse variables equals the number of nonlinear equality constraints in the second stage. This formulation includes the AR-OPF problem. We reformulate the ARO problem as a semi-infinite program. However, this semi-infinite program is difficult to solve with the adaptive discretization algorithm (Algorithm 1) since the separation problem is a difficult max-min problem that we cannot solve to a guaranteed optimality gap. We propose an alternative discretization algorithm based on a deterministic sampling of the uncertainty set and a homotopy method to solve the nonlinear system locally. Depending on whether the master problems are solved locally or globally, the algorithm converges to a local or global minimum of the ARO problem. Interestingly, global convergence occurs even though the homotopy method is only a local equation solver. For a positive feasibility tolerance, finite convergence occurs. We apply the algorithm to AR-OPF instances with up to 1354 buses built from the IEEE PES PGLib benchmark.

D. Publications

The journal and conference publications, and preprints on which this thesis is built are listed below. We also list the corresponding chapters of the thesis.

Journal papers

[46] M. Cerulli, A. Oustry, C. D'Ambrosio, L. Liberti. *Convergent algorithms for a class of convex Semi-Infinite programs*, SIAM Journal on Optimization, 2022. [Chapter 1 and Chapter 2]
[173] A. Oustry. *AC Optimal Power Flow: a Conic Programming relaxation and an iterative MILP scheme for Global Optimization*, Open Journal of Mathematical Optimization, 3(6):1–19, 2022. [Chapter 4] This paper received the "Best student paper" award of the ROADEF in 2022.

International conference paper

[176] A. Oustry, C. D'Ambrosio, L. Liberti, M. Ruiz, *Certified and accurate SDP bounds for the ACOPF problem*, Proceedings of the 22nd PSCC conference, Porto, Portugal, 2022. [Chapter 5]

Preprints

[175] A. Oustry, M. Cerulli. *Semi-infinite programming solution algorithms with inexact separation oracles*, Submitted, 2023. [Chapter 1 and Chapter 2]

[180] A. Oustry, M. Tacchi. *Minimum time control via semi-infinite programming*, `arXiv:2307.00857`, Submitted, 2023. [Chapter 3]

[179] A. Oustry, F. Pacaud, M. Anitescu. *Adjustable robust nonlinear optimization via semi-infinite programming*, Submitted. [Chapter 6]

In addition, publications on other topics have been achieved during the thesis.

Journal papers

[178] A. Oustry, M. Le Tilly, T. Clausen, C. D'Ambrosio, L. Liberti. *Optimal deployment of indoor wireless local area networks*, Networks, 81:23-50, 2023.

[177] A. Oustry, B. Erkan, R. Svartzman, P.-F. Weber. *Risques climatiques et politique de collatéral des banques centrales : une expérience méthodologique*, Revue économique, 73(2):173-218, 2022.

National conference paper

[145] L. Liberti, B. Manca, A. Oustry, and P.-L. Poirion. *Random projections for semidefinite programming*, Proceedings of the AIRO-ODS 2022 conference, Florence, Italy, 2022.

Encyclopedia articles

[45] M. Cerulli, D. Delle Donne, M. Escobar, L. Liberti, A. Oustry *Optimal Power Flow*, 3$^{\text{rd}}$ edition of the Encyclopedia of Optimization, Springer, 2023.

[182] A. Oustry, L. Xu, S. Haddad-Vanier, J.-A. Cordero, T. Clausen. *Optimization in Wireless Networks*, 3$^{\text{rd}}$ edition of the Encyclopedia of Optimization, Springer, 2023.

Part I

Contributions to convex semi-infinite programming and applications

Chapter 1

A convergence rate for the cutting-plane algorithm

Kelley's algorithm, also known as the cutting-plane algorithm (CPA) for convex programming, is a classic optimization algorithm dating back from 1960 [118]. This chapter introduces a convergence rate for this algorithm, and some variants, in the special case of convex semi-infinite programming. As presented in the Introduction, given two continuous functions $F\colon \mathbb{R}^m \to \mathbb{R}$ and $G\colon \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$, given two nonempty and compact sets $\mathcal{X} \subset \mathbb{R}^m$, and $\mathcal{Y} \subset \mathbb{R}^n$, we consider the semi-infinite programming formulation (**SIP**), defined as

$$\left.\begin{array}{ll} \min\limits_{x\in\mathcal{X}} & F(x) \\ \text{s.t.} & \forall y \in \mathcal{Y},\ G(x,y) \leq 0. \end{array}\right\} \qquad (\textbf{SIP})$$

In the present chapter, we study the formulation (**SIP**) under the assumption that it is convex with respect to the decision vector x.

Assumptions 1.1. *The set $\mathcal{X}$ is convex, the function $F(x)$ is convex and the function $G(\cdot, y)$ is linear for any $y \in \mathcal{Y}$.*

We highlight that we make no convexity assumptions with respect to the parameter y: neither the set $\mathcal{Y}$ is assumed to be convex, nor the function $G(x, \cdot)$, for a fixed $x \in \mathcal{X}$, is assumed to be convex/concave. Without loss of generality, the setting of Assumption 1.1 also covers the case where the function $G(\cdot, y)$ is affine for any $y \in \mathcal{Y}$, since we can add a decision variable x_0 and include the constraint $x_0 = 1$ in the definition of the convex and compact set $\mathcal{X}$.

As explained in the Introduction, one can write the semi-infinite programming problem (**SIP**) as

$$\left.\begin{array}{ll} \min\limits_{x\in\mathcal{X}} & F(x) \\ \text{s.t.} & \phi(x) \leq 0, \end{array}\right\} \tag{1.1}$$

where $\phi(x) = \max_{y\in\mathcal{Y}} G(x,y)$ is the value function of the lower-level problem. Under Assumption 1.1, this function is convex, and therefore, problem (1.1) is a constrained convex optimization problem. A *separation oracle* that computes $\hat{y}(x) \in \mathsf{argmax}_{y\in\mathcal{Y}} G(x,y)$, as defined in the Introduction, allows one to compute the value of $\phi(x)$, and a subgradient $s \in \partial\phi(x)$. In this setting of convex semi-infinite programming, the adaptive discretization algorithm [31] presented in the Introduction (Algorithm 1), is exactly the Kelley's cutting-plane algorithm (CPA). In this chapter, we prove an $O(\frac{1}{k})$ convergence rate for this algorithm, where k is the number of calls to the oracle, in the case where the objective function is strongly convex, and under the assumption that there exists a strictly feasible point.

Related works

On the one hand, we can think of the problem (**SIP**) as a generic convex optimization problem (1.1). Therefore, we review the algorithms for constrained convex optimization and their respective convergence rates. We focus on first-order algorithms because we are in the framework of the separation oracle: instead of a closed-form expression for $\phi(x)$, we only have a black-box algorithm that returns an (approximate) value of $\phi(x)$, and an (approximate) subgradient. More powerful structured optimization techniques, based on self-concordant functions, for example, are not appropriate here because they require more structure. The first algorithm for convex programs was derived independently by Kelley [118] and Cheney and Goldstein [50] in 1959-1960: it is the cutting-plane algorithm (CPA), the convergence rate of which is studied in this chapter. The ellipsoid algorithm was proposed by the Russian mathematician Shor in 1977 for convex optimization problems [201] before being used by Khachyan in 1979 to prove the polynomial-time complexity of linear programming [119]. Although this algorithm has a linear convergence rate [170], i.e., a convergence rate in $O(\exp(-\alpha k))$ with $\alpha \in \mathbb{R}_{++}$, it is not used in practice because it is known to be numerically unstable. In the unconstrained setting, where the oracle is used to compute the objective function, the celebrated primal-dual subgradient algorithm by Nesterov [171] has a sublinear rate of convergence in $O(\frac{1}{\sqrt{k}})$, even when the oracle is inexact [63]. This algorithm was also adapted to convex constrained optimization with the same rate [160]. Bundle methods [121, 138, 140, 183], also popular in practice, do not have a convergence rate in general. A convergence rate in $O(\frac{1}{\sqrt{k}})$ is known for the level-bundle method [139]. In the unconstrained case, with a strongly convex objective function, the standard bundle method [68] has an iteration complexity in $O(\epsilon^{-1}\log(\epsilon^{-1}))$, i.e., a convergence rate close to but strictly worse than $O(\frac{1}{k})$. As regards the analytic center cutting-plane method, we know an iteration complexity in $O(\frac{1}{\epsilon^2})$, i.e., a convergence rate $O(\frac{1}{\sqrt{k}})$ [240]. The Elzinga-Moore algorithm is another central cutting plane method [71]. The authors prove the linear convergence of the objective values of the feasible iterates generated by this algorithm. However, this is not a real convergence rate with respect to k, the total number of calls to the oracle, since the number of calls required to generate the j-th feasible iterate is not bounded.

On the other hand, one can also look at algorithms specifically designed for semi-infinite programming [83, 191, 196]. Among the works devoted to linear semi-infinite programming, one counts several variants of the central cutting-plane algorithm [24, 89, 104]. The same property of the "pseudo" linear convergence rate is stated for these algorithms, as for the Elzinga-Moore algorithm: neither the objective error nor the feasibility error is bounded by an explicit function of k. For fixed discretization approaches [103, 195, 207], where $\mathcal{Y}$ is approximated by a fixed grid $\mathcal{Y}^d$, there is a convergence rate of the approximation error of the solution as a function of the Haussdorff distance between $\mathcal{Y}$ and $\mathcal{Y}^d$ [207]. However, this does not provide a convergence rate in terms of the number of calls to the separation oracle. The *exchange* algorithms [32, 243], which are a refinement of the cutting-plane algorithm with constraints dropping, do not have a proven convergence rate, to the best of our knowledge. For the global optimization algorithms proposed by Mitsos et al. for general semi-infinite programming [65, 161], the convergence is proven, but no rate is stated.

Contributions and organization of the chapter

In this context of convex semi-infinite programming (Assumption 1.1), under the assumptions that the objective function is strongly convex and that a strictly feasible point exists, we prove an $O(\frac{1}{k})$ convergence rate for the algorithm CPA and several variants. This convergence rate is proven for the objective value of the iterates and their feasibility with respect to the semi-infinite constraint. This convergence rate is guaranteed despite the limited relative accuracy of the separation oracle, in line with the framework of the δ-oracle presented in the Introduction. Indeed, as the lower-level problem may be computationally hard, we relax the assumption that the oracle computes a point in $\mathsf{argmax}_{y\in\mathcal{Y}} G(x,y)$, and we just assume that there exists $\delta \in [0,1)$ such that for all $x \in \mathcal{X}$, the solution $\hat{y}(x) \in \mathcal{Y}$ and the upper bound $\hat{v}(x) \in \mathbb{R}$ on $\phi(x)$ computed by the oracle satisfy $\hat{v}(x) - G(x,\hat{y}(x)) \leq \delta\,|\phi(x)|$. The case of the "perfect" oracle is covered by the value $\delta = 0$. We introduce the Lagrangian dual problem of the convex semi-infinite program in Section 1.1. We provide the pseudocode of CPA, some variants with constraints dropping and aggregation, and the dual interpretation of these algorithms in Section 1.2. Based on this dual interpretation, we introduce the $O(\frac{1}{k})$ convergence rates for CPA in Section 1.3.

1.1 The Lagrangian dual of the convex semi-infinite program

1.1.1 Derivation of the dual problem

In line with Assumption 1.1, for any $y \in \mathcal{F}$, we define $a(y) \in \mathbb{R}^m$ such that $G(x,y) = x^\top a(y)$. From the continuity of G, we deduce the continuity of $a(\cdot)$. We also define the set $\mathcal{M} = \{a(y) \colon y \in \mathcal{Y}\}$ and $\mathcal{K} = \mathsf{cone}(\mathcal{M})$ the convex cone generated by $\mathcal{M}$. With this notation, we observe that the semi-infinite program (**SIP**) also reads

$$\left.\begin{aligned} &\min_{x\in\mathcal{X}} && F(x) \\ &\text{s.t.} && \forall z \in \mathcal{M},\ x^\top z \leq 0. \end{aligned}\right\} \qquad (\textbf{SIP}')$$

We introduce the Lagrangian function $\mathcal{L}(x,z) = F(x) + x^\top z$, defined over $\mathcal{X} \times \mathcal{K}$, and we notice that $\mathsf{val}(\textbf{SIP}) = \min_{x\in\mathcal{X}} \sup_{z\in\mathcal{K}} \mathcal{L}(x,z)$. We underline that the Lagrangian is convex with respect to x, and linear with respect to z. Since the set $\mathcal{X}$ is compact and convex (Assumption 1.1) and the set $\mathcal{K}$ is convex, Sion's minimax theorem [202] is applicable and the following holds:

$$\min_{x\in\mathcal{X}} \sup_{z\in\mathcal{K}} \mathcal{L}(x,z) = \sup_{z\in\mathcal{K}} \min_{x\in\mathcal{X}} \mathcal{L}(x,z). \tag{1.2}$$

We introduce the dual function

$$D(z) = \min_{x\in\mathcal{X}} \mathcal{L}(x,z), \tag{1.3}$$

and the dual optimization problem

$$\sup_{z\in\mathcal{K}} D(z). \qquad (\textbf{DSIP})$$

With this definition, Eq. (1.2) means the absence of duality gap between the dual problems (**SIP**) and (**DSIP**), i.e., $\mathsf{val}(\textbf{SIP}) = \mathsf{val}(\textbf{DSIP})$.

1.1.2 Properties of the dual problem

We make further assumptions to guarantee the regularity of the dual problem.

Assumptions 1.2. *The function $F(x)$ is μ-strongly convex.*

The following proposition states that under this assumption, the dual function is differentiable, with a Lipschitz continuous gradient.

Proposition 1.1. *Under Assumptions 1.1-1.2, the dual function $D(z)$ is differentiable, with gradient $\nabla D(z) = \arg\min_{x\in\mathcal{X}} \mathcal{L}(x,z)$. The gradient $\nabla D(z)$ is $\frac{1}{\mu}$-Lipschitz continuous.*

Proof. We apply [106, Cor. VI.4.4.5] to the function $D(z) = -(\sup_{x\in\mathcal{X}} -\mathcal{L}(x,z))$. The set $\mathcal{X}$ is compact. The function $-\mathcal{L}(\cdot,z)$ is continuous for all $z \in \mathcal{K}$. The function $-\mathcal{L}(x,\cdot)$ is convex and differentiable for all $x \in \mathcal{X}$. The function $\sup_{x\in\mathcal{X}} -\mathcal{L}(x,\cdot)$ is finite-valued over $\mathbb{R}^m$. Finally, due to Assumptions 1.1 and 1.2, the function $-\mathcal{L}(\cdot,z)$ is strongly concave and the set $\mathcal{X}$ is compact and convex, therefore the supremum $\sup_{x\in\mathcal{X}} -\mathcal{L}(x,z)$ is attained for a unique $x(z)$. We deduce from [106, Cor. VI.4.4.5] that $D(z)$ is differentiable over $\mathbb{R}^m$, with gradient $\nabla_z\mathcal{L}(x(z),z) = x(z)$. We now take $z, z' \in \mathbb{R}^m$, and prove that $\|\nabla D(z) - \nabla D(z')\| \leq \frac{1}{\mu}\|z - z'\|$. We define the functions $w(u) = \mathcal{L}(u,z) + i_{\mathcal{X}}(u)$ and $w'(u) = \mathcal{L}(u,z') + i_{\mathcal{X}}(u)$, where $i_{\mathcal{X}}(\cdot)$ is the characteristic function of $\mathcal{X}$. We introduce x (resp. x') the unique minimum of w (resp. w'). From the first part of the proof, we know that $\nabla D(z) = x$ and $\nabla D(z') = x'$. The first-order optimality condition for the convex functions w and w' reads

$$0 \in \partial w(x) \tag{1.4}$$

$$0 \in \partial w'(x'). \tag{1.5}$$

We notice that the function $(F + i_{\mathcal{X}})(u)$ is convex due to Assumption 1.1, and the function $\ell(u) = z^\top u$ is linear, thus convex. The intersection of the relative interiors of the domains of these convex functions is $\mathsf{ri}(\mathcal{X})$. With $\mathcal{X}$ a finite-dimensional convex set, $\mathsf{ri}(\mathcal{X}) \neq \emptyset$, according to [219, Proposition 1.9]. Hence, the subdifferential of the sum is the sum of the subdifferentials [198, Theorem 2.1], i.e., $\partial w(x) = \partial(F + i_{\mathcal{X}})(x) + \partial\ell(x) = \partial(F + i_{\mathcal{X}})(x) + \{z\}$. Similarly, $\partial w'(x') = \partial(F + i_{\mathcal{X}})(x') + \{z'\}$. Therefore, Eqs. (1.4)-(1.5) may be reformulated as the existence of $s \in \partial(F + i_{\mathcal{X}})(x)$ and $s' \in \partial(F + i_{\mathcal{X}})(x')$ such that

$$0 = s + z \tag{1.6}$$

$$0 = s' + z'. \tag{1.7}$$

Due to Assumptions 1.1-1.2, the function $F + i_{\mathcal{X}}$ is μ-strongly convex. Applying [106, Theorem VI.6.1.2], the μ-strong convexity of $F + i_{\mathcal{X}}$ gives that $(s - s')^\top(x - x') \geq \mu\|x - x'\|^2$, since $s \in \partial(F + i_{\mathcal{X}})(x)$ and $s' \in \partial(F + i_{\mathcal{X}})(x')$. Using the Cauchy-Schwartz inequality and Eqs. (1.6)-(1.7), we deduce that $\|z - z'\|\,\|x - x'\| \geq \mu\|x - x'\|^2$. Noting that $\nabla D(z) = x$ and $\nabla D(z') = x'$, we deduce

$$\|z - z'\|\,\|\nabla D(z) - \nabla D(z')\| \geq \mu\|\nabla D(z) - \nabla D(z')\|^2. \tag{1.8}$$

From Eq. (1.8), we deduce that $\frac{1}{\mu}\|z - z'\| \geq \|\nabla D(z) - \nabla D(z')\|$, if $\|\nabla D(z) - \nabla D(z')\| > 0$. If $\|\nabla D(z) - \nabla D(z')\| = 0$, this inequality is also trivially true. $\square$

Proposition 1.2. *Under Assumptions 1.1-1.2, for any $y, z \in \mathcal{K}$, for any $\gamma \geq 0$,*

$$D(z + \gamma y) \geq D(z) + (\nabla D(z)^\top y)\gamma - \frac{\|y\|^2}{2\mu}\gamma^2. \tag{1.9}$$

Proof. For any $y, z \in \mathcal{K}$ and $\gamma \geq 0$, we obtain by integration that

$$D(z+\gamma y) - D(z) = \int_0^\gamma \nabla D(z+ty)^\top y \, dt = \gamma \nabla D(z)^\top y + \int_0^\gamma (\nabla D(z+ty) - \nabla D(z))^\top y \, dt. \tag{1.10}$$

Using the Cauchy-Schwarz inequality and the $\frac{1}{\mu}$-smoothness of D, according to Proposition 1.1, we know that

$$(\nabla D(z+ty) - \nabla D(z))^\top y \geq -\|\nabla D(z+ty) - \nabla D(z)\|_2 \, \|y\|_2 \geq -\frac{t}{\mu}\|y\|_2^2. \tag{1.11}$$

Combining this with Eq. (1.10), we deduce that $D(z+ty) - D(z) \geq \gamma(\nabla D(z))^\top y - \int_{t=0}^{\gamma} \frac{t}{\mu}\|y\|_2^2 dt$, which yields that $D(z+\gamma y) - D(z) \geq \gamma(\nabla D(z))^\top y - \frac{\|y\|^2}{2\mu}\gamma^2$. ☐

We prove now that we can replace the sup operator with the max operator in the formulation (**DSIP**), under the following additional assumption.

Assumptions 1.3. *There exists $\tilde{x} \in \mathcal{X}$, such that $\tilde{x}^\top a(y) < 0$ for all $y \in \mathcal{Y}$.*

Proposition 1.3. *Under Assumptions 1.1 and 1.3, problem (**DSIP**) admits an optimal solution.*

Proof. According to Assumption 1.3, we use $\tilde{x} \in \mathcal{X}$ such that $\tilde{x}^\top a(y) < 0$ for all $y \in \mathcal{Y}$. By continuity of G and compactness of $\mathcal{Y}$, we know that there exists a constant $c \in \mathbb{R}_{++}$ such that $\tilde{x}^\top a(y) \leq -c$ for all $y \in \mathcal{Y}$. For any $z \in \mathcal{K}$, there exists $r \in \mathbb{N}$, $y_1, \ldots, y_r \in \mathcal{Y}$, and $\rho_1, \ldots, \rho_r \in \mathbb{R}_{++}$ such that $z = \sum_{i=1}^p \rho_i a(y_i)$. By definition of $D(z)$, $D(z) \leq \mathcal{L}(\tilde{x}, z)$. We deduce that $D(z) \leq F(\tilde{x}) - c\sum_{i=1}^p \rho_i$, which also reads $\sum_{i=1}^p \rho_i \leq c^{-1}(F(\tilde{x}) - D(z))$. From this equation, we deduce that for any solution of (**DSIP**) such that $D(z) \geq V - 1$ where $V = \mathsf{val}(\textbf{SIP}) = \mathsf{val}(\textbf{DSIP})$, we have $\sum_{i=1}^p \rho_i \leq c^{-1}(F(\tilde{x}) - V + 1)$, and this for any decomposition $z = \sum_{i=1}^p \rho_i a(y_i)$. This implies that the set of elements $z \in \mathcal{K}$ such that $D(z) \geq V - 1$, is included in the compact set $q\mathsf{conv}(\mathcal{M})$, where $q = c^{-1}(F(\tilde{x}) - V + 1)$. Over this compact set, the continuous function $D(z)$ reaches its maximum. ☐

1.2 The cutting-plane algorithm and its dual interpretation

1.2.1 Description of the algorithm with constraints management

In the setting of convex semi-infinite programming defined by Assumption 1.1, we instantiate Algorithm 1, which is, from a convex optimization perspective, the cutting-plane algorithm (CPA). In fact, Algorithm 2 is not exactly Algorithm 1, but a variant that offers more flexibility with respect to finite subset of constraints kept in the master problem ($\mathbf{R_k}$), as detailed in step 6 of the pseudo-code. We insist on three particular cases regarding the set $\mathcal{S}^k$:

- The set $\mathcal{S}^k$ may be equal to $\mathcal{M}^k$ at every step, in which case the algorithm coincides with Algorithm 1 with the correspondence $\mathcal{M}^k = a(\mathcal{Y}^k)$

- The set $\mathcal{S}^k$ may be the set of atoms $z \in \mathcal{M}^k$ such that $\rho_z > 0$, in which case step 6 consists in dropping all the inactive constraints.
- The set $\mathcal{S}^k$ may be a subset of $\mathsf{conv}(\mathcal{M}^k)$ of size at most M, and such that $z^k \in \mathsf{cone}(\mathcal{S}^k)$. For instance for $M = 1$, we can take $\mathcal{S}^k = \{\frac{1}{\sum_z \rho_z} \sum_z \rho_z z\}$, where the sums go for $z \in \mathcal{M}^k$. In this case, known as "subgradient aggregation" in the bundle methods literature [95, 121], we obtain a bounded-memory algorithm.

We can also think about intermediate approaches where we do not drop every inactive constraint, but only the ones that have been staying inactive for a given number of iterations. Such strategies are also included in the framework of Algorithm 2.

Algorithm 2 CPA algorithm for (**SIP**) with limited-accuracy oracle and constraints management

```
    Input: Oracle with parameter $\delta \in [0,1)$, tolerance $\epsilon \in \mathbb{R}_+$, finite set $\mathcal{M}^0 \subset \mathcal{M}$
 0: Let $k \leftarrow 0$.
 1: $\nu_0 \leftarrow \infty$.
 2: while $\nu_k > \epsilon$ do
 3:     Compute $x^k$ an optimal solution of the relaxation
            $$\left.\begin{array}{ll} \min\limits_{x \in \mathcal{X}} & F(x) \\ \text{s.t.} & \forall z \in \mathcal{M}^k,\ x^\top z \leq 0, \end{array}\right\} \qquad (\mathbf{R_k})$$
        and compute $z^k = \sum_{z \in \mathcal{M}^k} \rho_z z$, where $\rho_z \in \mathbb{R}_+$ is the dual variable of $x^\top z \leq 0$.
 4:     Call the $\delta$-oracle to compute $y^k = \hat{y}(x^k)$ an approximate solution of $\max\limits_{y \in \mathcal{Y}} a(y)^\top x^k$.
 5:     $\xi^k \leftarrow a(y^k)$.
 6:     $\mathcal{M}^{k+1} \leftarrow \mathcal{S}^k \cup \{\xi^k\}$, where $\mathcal{S}^k$ is any finite subset of $\mathsf{conv}(\mathcal{M}^k)$ such that $z^k \in \mathsf{cone}(\mathcal{S}^k)$
 7:     $\nu_{k+1} \leftarrow (\xi^k)^\top x^k$
 8:     $k \leftarrow k + 1$
 9: end while
10: Return $x^k$.
```

1.2.2 Dual interpretation as a Frank-Wolfe algorithm

We show that Algorithm 2, can be interpreted, from a dual perspective, as a cone constrained Fully Corrective Frank–Wolfe (FCFW) algorithm [147] solving the dual problem (**DSIP**). We prove that, during the execution of Algorithm 2, the dual vectors z^k instantiate the iterates of an FCFW algorithm. The generic iteration k is described in Table 1.1. We also give more details for the two first steps:

- *Step 1*: At iteration k, the dual problem of ($\mathbf{R_k}$) is in fact a restriction of (**DSIP**) on

	Primal perspective: Algorithm 2 (CPA)	Link	Dual perspective: FCFW
Step 1	Solve ($\mathbf{R_k}$), store the solution x^k, and the dual vector z^k	Strong duality	Solve the dual problem $\max_{z\in\mathsf{cone}(\mathcal{M}^k)} D(z)$, compute the solution z^k, and the gradient $\nabla D(z^k) = x^k$
Step 2	Call the δ-oracle to solve $\max_{y\in\mathcal{Y}} a(y)^\top x^k$, store the solution y^k, and set $\xi^k = a(y^k)$	$\xi^k = a(y^k)$ $x^k = \nabla D(z^k)$	Call the δ-oracle to solve $\max_{y\in\mathcal{M}} \xi^\top \nabla D(z^k)$, and store the solution ξ^k
Step 3	$\mathcal{M}^{k+1} \leftarrow \mathcal{S}^k \cup \{\xi^k\}$		$\mathcal{M}^{k+1} \leftarrow \mathcal{S}^k \cup \{\xi^k\}$
Stopping criterion	$(\xi^k)^\top x^k \le \epsilon$	$x^k = \nabla D(z^k)$	$(\xi^k)^\top \nabla D(z^k) \le \epsilon$

Table 1.1: The k-th iteration of the algorithms CPA (Algorithm 2) and FCFW

$\mathsf{cone}(\mathcal{M}^k)$, which is a polyhedral subcone of $\mathcal{K}$. Indeed, the following holds:

$$\begin{aligned}\max_{z\in\mathsf{cone}(\mathcal{M}^k)} D(z) &= \max_{z\in\mathsf{cone}(\mathcal{M}^k)} \min_{x\in\mathcal{X}} F(x) + x^\top z\\ &= \min_{x\in\mathcal{X}} \max_{z\in\mathsf{cone}(\mathcal{M}^k)} F(x) + x^\top z\\ &= \min_{x\in\mathcal{X}}\{F(x)\colon \forall z\in\mathcal{M}^k, x^\top z \le 0\},\end{aligned}$$

which we recognize being the master problem ($\mathbf{R_k}$). The absence of duality gap is, also in this case, a direct application of Sion's Theorem [202]. The new dual solution z^k is obtained solving this restriction of (**DSIP**) on $\mathsf{cone}(\mathcal{M}^k)$, and the primal solution $x^k = \arg\min_{x\in\mathcal{X}} \mathcal{L}(x, z^k)$ gives the gradient of the dual function in z^k, i.e., $\nabla D(z^k) = x^k$ (see Proposition 1.1).

- *Step 2*: we notice that $\max_{y\in\mathcal{Y}} a(y)^\top x^k = \max_{\xi\in\mathcal{M}} \xi^\top x^k = \max_{\xi\in\mathcal{M}} \xi^\top \nabla D(z^k)$, since $\mathcal{M} = a(\mathcal{Y})$, and $\nabla D(z^k) = x^k$.

We notice that the dual iterates of the algorithm satisfies an orthogonality property.

Lemma 1.1. *For any $k\in\mathbb{N}$, $\nabla D(z^k)^\top z^k = 0$.*

Proof. This property follows from the first order optimality condition at 1 of the differentiable function $\alpha(t) = D(tz^k)$. Indeed, $\alpha'(1) = (\nabla D(z^k))^\top z^k = 0$, because (i) 1 is optimal for w since $z^k \in \operatorname*{argmax}_{z\in\mathsf{cone}(\mathcal{M}^k)} D(z)$, (ii) 1 lies in the interior of the definition domain of α. □

1.3 Convergence rate for the cutting-plane algorithm

1.3.1 Convergence rate for the objective

We define the constant $R = \sup_{z \in \mathcal{M}} \|z\| = \sup_{z \in \mathsf{conv}(\mathcal{M})} \|z\|$. According to Proposition 1.3, problem (**DSIP**) admits an optimal solution $z^* \in \mathcal{K}$. We define $\tau = \inf\{t \geq 0 \colon z^* \in t\,\mathsf{conv}(\mathcal{M})\}$. This scalar plays a central role in the convergence rate analysis of Algorithm 2, conducted in the following theorem.

Theorem 1.1. *Under Assumptions 1.1-1.3, if Algorithm 2 executes iteration $k \in \mathbb{N}$, then*

$$F(x^*) - F(x^k) \leq \frac{2\,R^2\tau^2}{\mu\,(1-\delta)^2}\,\frac{1}{k+2}. \tag{1.12}$$

In this convergence rate, some constants are, on the one hand, characteristics of the problem (**SIP**): μ is the strong convexity parameter of the objective function, R is the radius of $\mathcal{X}$, and τ is a scaling factor of a dual optimal solution. On the other hand, $\delta \in [0,1)$ is not a property of the problem (**SIP**) but of the separation oracle used: this is its relative accuracy.

Proof. We define the optimality gap $\Delta_k = F(x^*) - F(x^k) = \mathsf{val}(\textbf{SIP}) - F(x^k)$. We emphasize, that at each iteration k, $D(z^k) = F(x^k)$, thus Δ_k is also the optimality gap in the dual problem (**DSIP**): $\Delta_k = \mathsf{val}(\textbf{DSIP}) - F(x^k) = D(z^*) - D(z^k)$. We prove the inequality (1.12) by induction. We exclude the trivial case where $\tau = 0$ (i.e. where $\mathbf{0}$ is a dual optimal solution), in which case Eq. (1.12) holds, since $F(x^k) = F(x^0) = \mathsf{val}(\textbf{SIP})$. Therefore, we assume that $\tau \in \mathbb{R}_{++}$.

Base case ($k = 0$). Since $0 \in \mathsf{cone}(\mathcal{M}^0)$, and since D is concave, $\Delta_0 = D(z^*) - D(z^0) \leq D(z^*) - D(0) \leq \nabla D(0)^\top (z^* - 0) = \nabla D(0)^\top z^*$. We remark that $\nabla D(0)^\top z^* = (\nabla D(0) - \nabla D(z^*))^\top z^*$ since $\nabla D(z^*)^\top z^* = 0$ by optimality of z^*. Hence,

$$\Delta_0 \leq (\nabla D(0) - \nabla D(z^*))^\top z^* \leq \|\nabla D(0) - \nabla D(z^*)\|\,\|z^*\|,$$

where the last inequality is the Cauchy-Schwartz inequality. Using the $\frac{1}{\mu}$-Lipschitzness of ∇D (Proposition 1.1), we know that $\|\nabla D(0) - \nabla D(z^*)\| \leq \frac{1}{\mu}\|z^*\|$. Since $z^* \in \tau\mathsf{conv}(\mathcal{M})$, $\Delta_0 \leq \frac{1}{\mu}\|z^*\|^2 \leq \frac{(R\tau)^2}{\mu} \leq \frac{(R\tau)^2}{(1-\delta)^2\mu}$ as $1 - \delta \in (0,1]$.

Induction. We suppose that the algorithm runs $k+1$ iterations and does not meet the stopping condition; we assume that the property (1.12) is true for k. Since $z^k \in \mathsf{cone}(\mathcal{S}^k)$, and $\mathcal{M}^{k+1} = \mathcal{S}^k \cup \{\xi^k\}$, we deduce that $z^k + \gamma\xi^k \in \mathsf{cone}(\mathcal{M}^{k+1})$, for any $\gamma \geq 0$, implying

$D(z^{k+1}) \geq D(z^k + \gamma \xi^k)$. Moreover, Proposition 1.2 yields a lower bound on the progress made during the iteration $k+1$:

$$D(z^{k+1}) \geq D(z^k + \gamma \xi^k) \geq D(z^k) + \gamma \, \nabla D(z^k)^\top \xi^k - \frac{\|\xi^k\|^2}{2\mu}\gamma^2, \tag{1.13}$$

for any $\gamma \geq 0$. Multiplying by -1, adding $D(z^*)$ to both left and right hand sides of the above inequality, and using $\|\xi^k\| \leq R$, we have that

$$\Delta_{k+1} \leq \Delta_k - \gamma \, \nabla D(z^k)^\top \xi^k + \frac{R^2}{2\mu}\gamma^2, \tag{1.14}$$

for any $\gamma \geq 0$. In addition, by concavity of D, $\Delta_k = D(z^*) - D(z^k) \leq \nabla D(z^k)^\top (z^* - z^k)$. By Lemma 1.1, we have $\nabla D(z^k)^\top z^k = 0$. Thus, $\Delta_k \leq \nabla D(z^k)^\top z^*$. As $z^* \in \tau \, \mathsf{conv}(\mathcal{M})$,

$$\Delta_k \leq \max_{z \in \tau \mathsf{conv}(\mathcal{M})} \nabla D(z^k)^\top z = \tau \max_{z \in \mathcal{M}} \nabla D(z^k)^\top z = \tau \phi(x^k), \tag{1.15}$$

where the last equality follows from $\nabla D(z^k) = x$, and from the definition of the value function $\phi(x) = \max_{y \in \mathcal{Y}} x^\top a(y)$. The stopping criterion $a(y^k)^\top x^k = (\xi^k)^\top x^k \leq \epsilon$ is not met at the end of iteration k, as iteration $k+1$ is executed. Therefore, $\phi(x) \geq a(y^k)^\top x^k > \epsilon \geq 0$. The property of the δ-oracle described by Eq. (4) yields $\phi(x^k) - G(x^k, y^k) \leq \delta \phi(x^k)$, i.e., $(1-\delta)\phi(x^k) \leq G(x^k, y^k) = (x^k)^\top a(y^k) = \nabla D(z^k)^\top \xi^k$. Therefore, as we have $\tau \in \mathbb{R}_{++}$ (see beginning of the proof), we deduce from Eq. (1.15) that

$$\frac{1-\delta}{\tau}\Delta_k \leq \nabla D(z^k)^\top \xi^k. \tag{1.16}$$

Combining Eqs. (1.14) and (1.16), we obtain $\Delta_{k+1} \leq \Delta_k - \gamma \frac{1-\delta}{\tau}\Delta_k + \frac{R^2}{2\mu}\gamma^2$, for every $\gamma \geq 0$. Factoring and setting $\tilde{\gamma} = \gamma \frac{1-\delta}{\tau}$ (for any $\tilde{\gamma} \geq 0$) yields

$$\Delta_{k+1} \leq (1 - \tilde{\gamma})\Delta_k + \frac{R^2 \tau^2}{2\mu(1-\delta)^2}\tilde{\gamma}^2. \tag{1.17}$$

Applying Eq. (1.17) with $\tilde{\gamma} = \frac{2}{k+2}$, and defining $C = \frac{2R^2\tau^2}{\mu(1-\delta)^2}$ we obtain

$$\Delta_{k+1} \leq (1 - \frac{2}{k+2})\Delta_k + \frac{C}{(k+2)^2} \leq \frac{k}{k+2}\frac{C}{k+2} + \frac{C}{(k+2)^2},$$

with the second inequality coming from the application of (1.12), which holds for k by the induction hypothesis. Finally, we deduce that

$$\Delta_{k+1} \leq \frac{C}{k+2}(\frac{k}{k+2} + \frac{1}{k+2}) \leq \frac{C}{k+2}\frac{k+1}{k+2} \leq \frac{C}{k+2}\frac{k+2}{k+3} = \frac{C}{k+3},$$

where the third inequality follows from the observation that $\frac{k+1}{k+2} \leq \frac{k+2}{k+3}$. Hence, the property (1.12) is true for $k+1$ as well. This concludes the proof. $\square$

1.3.2 Convergence rate for the feasibility error

The following theorem states that the lowest feasibility error of the iterates generated by the algorithm CPA (with a limited-accuracy oracle) also follows a $O(\frac{1}{k})$ convergence rate.

Theorem 1.2. *Under Assumptions 1.1-1.3, if Algorithm 2 executes iteration k, for $k \geq 2$, then*

$$\min_{0\leq \ell\leq k} \phi(x^\ell) \leq \frac{27\, R^2\tau}{4\mu(1-\delta)^2}\,\frac{1}{k+2}. \tag{1.18}$$

The following proof is inspired by previous work on the Frank-Wolfe algorithm [111], with some adaptations to the dual problem (**DSIP**), a convex optimization problem on a cone instead of on a compact set for the tradition Frank-Wolfe algorithm. We also adapt the proof to our framework of limited-accuracy oracle.

Proof. We keep the definition of the constant $C = \frac{2R^2\tau^2}{\mu\,(1-\delta)^2}$, we define the constants $\alpha = \frac{2}{3}$, $\beta = \frac{27}{8\tau}$, and $H = k+2$. We suppose that

$$\phi(x^\ell) > \frac{\beta C}{H}, \forall \ell \in [\![0, k]\!], \tag{1.19}$$

and we will show a contradiction. If iteration $k+1$ is not executed because the algorithm stopped at iteration k, we still define $z^{k+1} = \underset{z=z^k+\gamma\zeta^k,\gamma\geq 0}{\operatorname{argmax}} D(z)$, and $\Delta_{k+1} = D(z^*) - D(z^{k+1}) \geq 0$. Hence, regardless whether the iteration $k+1$ is executed or not, Δ_ℓ and $\Delta_{\ell+1}$ are well defined for all $\ell \in [\![0, k]\!]$, and we can apply Eq. (1.14) to deduce $\Delta_{\ell+1} \leq \Delta_\ell - \gamma\nabla D(z^\ell)^\top\xi^\ell + \frac{R^2}{2\mu}\gamma^2$, for any $\gamma \geq 0$. Eq. (1.19) implies that $\phi(x^\ell) > 0$, and from Eq. (4), we deduce that $\phi(x^\ell) - G(x^\ell, y^\ell) \leq \delta\phi(x^\ell)$, i.e., $(1-\delta)\phi(x^\ell) \leq G(x^\ell, y^\ell) = (x^\ell)^\top a(y^\ell) = \nabla D(z^\ell)^\top\xi^\ell$. Combining this with Eq. (1.19), we deduce that $\nabla D(z^\ell)^\top\xi^\ell > \frac{(1-\delta)\beta C}{H}$, and therefore, for any $\gamma \geq 0$, $\Delta_{\ell+1} < \Delta_\ell - \gamma\,\frac{(1-\delta)\beta C}{H} + \frac{R^2}{2\mu}\gamma^2$. Applying this inequality for $\gamma = \frac{\tau}{2(1-\delta)(\ell+2)} \geq 0$, we obtain

$$\Delta_{\ell+1} < \Delta_\ell - \frac{2\tau\beta C}{(\ell+2)H} + \frac{2R^2\tau^2}{\mu(1-\delta)^2}\frac{1}{(\ell+2)^2} \tag{1.20}$$

$$= \Delta_\ell - \frac{2\tau\beta C}{(\ell+2)H} + \frac{C}{(\ell+2)^2}. \tag{1.21}$$

We define $k_{\min} = \lceil\alpha H\rceil - 2$, and we notice that $k_{\min} \geq 0$, since $H \geq 4$. Note also that for any $\ell \in [\![k_{\min}, k]\!]$, $\alpha H \leq \ell + 2 \leq H$. Combining this with Eq. (1.21), we know that for any $\ell \in [\![k_{\min}, k]\!]$,

$$\Delta_{\ell+1} < \Delta_\ell - \frac{2\tau\beta C}{H^2} + \frac{C}{\alpha^2 H^2} = \Delta_\ell + \frac{C}{H^2}(\frac{1}{\alpha^2} - 2\tau\beta). \tag{1.22}$$

Summing these inequalities for $\ell \in [\![k_{\min}, k]\!]$, we obtain

$$\Delta_{k+1} < \Delta_{k_{\min}} + \frac{C(k+1-k_{\min})}{H^2}(\frac{1}{\alpha^2} - 2\tau\beta), \tag{1.23}$$

and using the bound on the objective gap at iteration $k_{\min}$ given by Theorem 1.1, we have

$$\Delta_{k+1} < \frac{C}{k_{\min}+2} + \frac{C(k+1-k_{\min})}{H^2}(\frac{1}{\alpha^2} - 2\tau\beta). \tag{1.24}$$

We notice that $k_{\min}+2 \geq \alpha H$, and $k+1-k_{\min} \geq (1-\alpha)H$. By definition of α and β, $\frac{1}{\alpha^2} - 2\tau\beta = \frac{9}{4} - \frac{27}{4} \leq 0$, and we deduce that

$$\Delta_{k+1} < \frac{C}{\alpha H} + \frac{C(1-\alpha)}{H}(\frac{1}{\alpha^2} - 2\tau\beta) = \frac{C}{\alpha H}(1 + \frac{1-\alpha}{\alpha} - 2\alpha(1-\alpha)\tau\beta). \tag{1.25}$$

Using again that $\alpha = \frac{2}{3}$, we deduce that $\Delta_{k+1} < \frac{C}{\alpha H}(\frac{3}{2} - \frac{4}{9}\tau\beta)$. Since $\beta = \frac{27}{8\tau}$, $(\frac{3}{2} - \frac{4}{9}\tau\beta) = 0$. Hence, we obtain $\Delta_{k+1} < 0$, which contradicts the definition of Δ_{k+1}. Therefore, the assumption at Eq. (1.19) cannot hold, and there exists $\ell \in [\![0, k]\!]$ such that $\phi(x^\ell) \leq \frac{\beta C}{H} = \frac{27}{4\mu(1-\delta)^2}\frac{R^2\tau}{}\frac{1}{k+2}$. $\square$

1.4 Conclusion

The cutting-plane algorithm is very standard in convex optimization, but its performance may be pretty poor in some cases. Nevertheless, this chapter highlights a particular case where its performance is guaranteed since the objective error and the feasibility error decrease in $O(1/k)$. This particular case is the following: the semi-infinite problem is convex with respect to x, the objective function is strongly convex, and a strictly feasible point exists. We specify that there is no assumption of convexity or of a particular structure concerning the semi-infinite constraint parameter y. Note also that the convergence rate is guaranteed even if the separation oracle has limited relative accuracy.

It could be interesting to pursue this line of research by relaxing some of the assumptions we make here. For example, one could relax the convexity assumption with respect to x by assuming that some variables x_i for $i \in I \subset [\![1, n]\!]$ are subject to integrity constraints: this would give an extension to mixed-integer convex semi-infinite programming. We conjecture that an $O(1/k)$ asymptotic convergence rate could be obtained using similar reasoning for fixed values of the integer variables. It is likely that the number of different possible (optimal) values for $(x_i)_{i\in I}$, a number that is potentially exponential with the size of the problem, would appear in one such convergence rate. Another extension would be to remain in continuous variables but to relax the strong convexity assumption for the objective and look for lighter assumptions that allow us to preserve the smoothness of the dual function, which plays a central role in our analysis.

CHAPTER 2

Inner-outer approximation algorithm for a class of convex semi-infinite programs

Semi-infinite programming is challenging from a numerical optimization perspective since feasibility requires the satisfaction of an infinite number of constraints. In Chapter 1, we presented and analyzed the cutting-plane algorithm (CPA) for convex semi-infinite programming. Although this algorithm presents interesting properties, such as a guaranteed convergence rate if the objective function is strongly convex, it suffers a significant drawback: until the algorithm converges, the generated iterates are not feasible, i.e., they do not satisfy all the constraints. Feasibility in the semi-infinite program is obtained only asymptotically. This may be problematic in the case of an early termination of the algorithm if the computational time budget is limited. To overcome this limitation and compute fast feasible solutions, the present chapter introduces a finite reformulation and an IOA for the problem (**SIP**). This chapter addresses the case where the lower-level problem is a QCQP problem, potentially nonconvex, as described by the following assumption.

Assumptions 2.1. *The parameterization of the semi-infinite constraints satisfies the following:*

- *There exist linear mappings $x \mapsto Q(x) \in \mathbb{S}_n$, $x \mapsto q(x) \in \mathbb{R}^n$ and $x \mapsto b(x) \in \mathbb{R}$ such that*

$$G(x,y) = -\frac{1}{2}y^\top Q(x)y + q(x)^\top y + b(x).$$

- *There exist $Q^1, \dots, Q^r \in \mathbb{S}_n$, $q^1, \dots, q^r \in \mathbb{R}^n$ and $b_1, \dots, b_r \in \mathbb{R}$ such that*

$$\mathcal{Y} = \left\{ y \in \mathbb{R}^n : \forall j \in [\![1, r]\!], \frac{1}{2}y^\top Q^j y + (q^j)^\top y + b_j \leq 0 \right\}.$$

We also assume to know $\rho \geq 0$ such that $\mathcal{Y} \subset B(0, \rho)$.

Note that this chapter is also in line with Assumption 1.1, which guarantees that the semi-infinite problem (**SIP**) is convex with respect to the decision vector x. Therefore, this chapter addresses a certain class of convex semi-infinite programs with lower-level problems that are (potentially nonconvex) QCQP.

Related works

We present here state-of-the-art methods for efficiently computing feasible solutions of semi-infinite programs. We distinguish several families of methods for this purpose.

Some adaptive discretization methods are designed to find feasible points before the final convergence of the algorithm, unlike the standard CPA [31, 118]. One can think of the algorithm of Mitsos [161, 162], that discretizes the constraints and plays on the right-hand-side term to obtain a feasible solution: instead of imposing $G(x, y) \leq 0$ for all $y \in \mathcal{Y}^k$ (where $\mathcal{Y}^k$ is a finite subset of $\mathcal{Y}$), the constraints $G(x, y) \leq -\epsilon^g$ are imposed for $\epsilon^g \in \mathbb{R}_{++}$; the value of ϵ^g is progressively decreased each time a feasible solution of the semi-infinite programs is found. The algorithm by Tsoukalas and Rustem [218] is centered around the auxiliary problem $\min_{x\in\mathcal{X}}\{F(x) - \hat{F}, \max_{y\in\mathcal{Y}^k} G(x, y)\}\}$, where $\hat{F}$ is a target value for the objective function, which is updated following a binary search. In [65], Djelassi and Mitsos propose hybridizing both algorithms.

Another approach, applicable to bilevel programming in general, is called the KKT approach and consists in replacing the lower-level problem by its KKT conditions to obtain a finite formulation [2]. If the lower-level problem is convex and satisfies some constraint qualification properties, the resulting finite formulation is an equivalent reformulation; if not, this is a relaxation. The main drawback of this approach is to result in a nonconvex formulation, even if the original semi-infinite program is convex.

Another family of methods called "overestimation methods", replaces the lower-level value function $\phi(x) = \max_{y\in\mathcal{Y}} G(x, y)$ by an overestimation, to obtain a finite restriction of the original semi-infinite program, and, therefore, a feasible point. On the one hand, convexification methods adaptively construct convex relaxations of the lower-level problem, replace the relaxed lower-level problem equivalently by its KKT conditions, and solve the resulting problem with complementarity constraints [74, 205], which is a restriction of the original problem. Once again, the resulting problem to solve is nonconvex, therefore, numerically challenging. On the other hand, some approaches dualize the lower-level problem. In linear robust optimization, the lower-level problem is dualized to obtain a finite LP formulation that is equivalent to the robust LP [18]. In robust polynomial optimization and polynomial min-max games [132], the polynomial programming lower-level problem is replaced by a moment relaxation, which is, then, dualized into a SoS problem. This way, a finite formulation is obtained that is a restriction of the original problem: convergence is proven in terms of value and solution when the degree of the moment SoS relaxation increases, i.e., at the price of increasing size. In [64], several strategies are used to reformulate generalized semi-infinite programs into finite nonconvex minimization problems by exploiting Wolfe duality for the convex lower-level problems. In [142], the authors tackle generalized semi-infinite problems where the convex quadratic lower-level problem has a fixed Hessian matrix Q, which does not depend on the variable x. Instead, in this chapter, we consider standard semi-infinite programs with a linear function $Q(x)$ for the lower-level Hessian (as stated in Assumption 2.1). Back to [142], the authors use the Lagrangian dual of the lower level to obtain a nonconvex restriction with a finite number of variables and constraints.

Contributions and organization of the chapter

First, we obtain a tractable restriction with a finite number of constraints by dualizing, using Lagrangian and SDP duality, the lower-level problem $\max_{y\in\mathcal{Y}} G(x,y)$. If $G(x,y)$ is concave in y, the obtained formulation is not a restriction but an exact reformulation of (**SIP**). The dualization technique has been used in the semi-infinite programming literature [64, 142], but, contrary to these approaches, the finite restriction we obtain with this approach is convex. Second, we present *a priori* and *a posteriori* conditions for this restriction to have the same value as the semi-infinite program (**SIP**). Even if these conditions are not met, we still have convergence guarantees. Indeed, we introduce the algorithm IOA, which is a hybrid algorithm mixing adaptative discretization and overestimation techniques. It progressively enlarges the restriction set to generate a sequence of feasible points, the values of which converge to the value of (**SIP**). This algorithm converges even with a separation oracle of limited relative accuracy δ.

We start by introducing the SDP relaxation of the lower-level problem (or a SDP reformulation if the latter is convex), and its strong SDP dual in Section 2.1. In Section 2.2, we present the finite formulation (**SIPR**), obtained by replacing the lower-level problem with its dual. This formulation is a reformulation of (**SIP**) if $Q^1,\dots,Q^r$ are PSD and $Q(x)$ is PSD for any $x\in\mathcal{X}$. Otherwise, an *a posteriori* sufficient condition on a computed solution $\bar{x}$ of (**SIPR**) introduced in Section 2.2.2 can be verified. If $\bar{x}$ satisfies such a condition, one can state that it is an optimal solution of (**SIP**). If not, the Inner-Outer Approximation (IOA) algorithm introduced in Section 2.3, generates a sequence of converging feasible solutions of (**SIPR**). Section 2.4 introduces numerical experiments, including a comparison with CPA (Algorithm 2), with Mitsos' algorithm [161], and with the KKT-approach [2].

2.1 Semidefinite relaxation of the lower-level problem

In this section, we reason for any fixed value of the decision vector $x\in\mathcal{X}$. The corresponding lower-level problem $\max_{y\in\mathcal{Y}} G(x,y)$ is the following QCQP problem

$$\left.\begin{array}{ll}\max\limits_{y\in\mathbb{R}^n} & -\frac{1}{2}y^\top Q(x)y + q(x)^\top y + b(x)\\ \text{s.t.} & \forall j\in[\![1,r]\!],\ \frac{1}{2}y^\top Q^j y + (q^j)^\top y + b_j \le 0.\end{array}\right\} \qquad (\mathbf{P}_x)$$

Note that this problem is potentially nonconvex. We introduce now the SDP relaxation of the QCQP problem ($\mathbf{P}_x$), and its Lagrangian dual problem.

2.1.1 Primal SDP relaxation of the lower-level problem

We define the linear matrix operator $\mathcal{Q}(x) = \frac{1}{2}\begin{pmatrix} -Q(x) & q(x) \\ q(x)^\top & 2b(x) \end{pmatrix} \in \mathbb{S}_{n+1}$, and the matrices $\mathcal{Q}^j = \frac{1}{2}\begin{pmatrix} Q^j & q^j \\ (q^j)^\top & 2b_j \end{pmatrix} \in \mathbb{S}_{n+1}$, for $j \in [\![1, r]\!]$, the identity matrix $I_{n+1} \in \mathbb{S}_{n+1}$, as well as $E \in \mathbb{S}_{n+1}$ defined as the matrix of the canonical basis associated with the indices $(n+1, n+1)$. With this notation, we introduce the SDP problem

$$\left.\begin{array}{ll} \max\limits_{Y \in \mathbb{S}_{n+1}} & \langle \mathcal{Q}(x), Y \rangle \\ \text{s.t.} & \forall j \in [\![1, r]\!],\ \langle \mathcal{Q}^j, Y \rangle \leq 0 \\ & \langle I_{n+1}, Y \rangle \leq 1 + \rho^2 \\ & \langle E, Y \rangle = 1 \\ & Y \succeq 0. \end{array}\right\} \qquad (\mathbf{SDP}_x)$$

As proven in Lemma 2.1, this problem is a relaxation of ($\mathbf{P}_x$). If the latter problem is convex, Lemma 2.1 states that ($\mathbf{SDP}_x$) has the same optimal objective value.

Lemma 2.1. *Under Assumption 2.1,* $\mathsf{val}(\mathbf{SDP}_x) \geq \mathsf{val}(\mathbf{P}_x)$*. If, moreover, $Q(x), Q^1, \dots, Q^r$ are PSD, then* $\mathsf{val}(\mathbf{SDP}_x) = \mathsf{val}(\mathbf{P}_x)$*.*

Proof. First, we underline that due to Assumption 2.1, the constraint $\|y\|^2 + 1 \leq 1 + \rho^2$ is redundant in the problem ($\mathbf{P}_x$), therefore, adding it does not change the value of this problem. Consequently, we recognize that ($\mathbf{SDP}_x$) is the standard SDP relaxation of the problem ($\mathbf{P}_x$) augmented with this redundant constraint, this is why $\mathsf{val}(\mathbf{SDP}_x) \geq \mathsf{val}(\mathbf{P}_x)$.
Second, we assume now that $Q(x), Q^1, \dots, Q^r$ are PSD. Given a matrix Y feasible for ($\mathbf{SDP}_x$), we denote by $u_1, \dots, u_{n+1} \in \mathbb{R}^{n+1}$ a basis of eigenvectors of Y (which is PSD) and their respective eigenvalues $v_1, \dots, v_{n+1} \in \mathbb{R}_+$. Let us introduce the two following index sets: $I = \{i \in [\![1, n+1]\!] \colon (u_i)_{n+1} \neq 0\}$ and $J = \{i \in [\![1, n+1]\!] : (u_i)_{n+1} = 0\}$, which gives $I \cup J = [\![1, n+1]\!]$. Moreover,

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

With this notation, we have that $Y = \sum\limits_{i=1}^{n+1} v_i u_i u_i^\top = \sum\limits_{i \in I} \mu_i \begin{pmatrix} y_i y_i^\top & y_i \\ y_i^\top & 1 \end{pmatrix} + \sum\limits_{i \in J} \nu_i \begin{pmatrix} z_i z_i^\top & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{pmatrix}$, where $\mathbf{0}$ is the null n-dimensional vector. Let us define the vector $\bar{y} = \sum\limits_{i \in I} \mu_i y_i$. Its objective value in

($\mathbf{P}_x$) is larger than the objective value of Y in ($\mathbf{SDP}_x$), since

$$\langle \mathcal{Q}(x), Y\rangle = \sum_{i\in I} \mu_i G(x, y_i) - \frac{1}{2}\sum_{i\in J} \nu_i z_i^\top Q(x) z_i \tag{2.1}$$

$$\leq \sum_{i\in I} \mu_i G(x, y_i) \tag{2.2}$$

$$\leq G(x, \sum_{i\in I} \mu_i y_i) = G(x, \bar{y}). \tag{2.3}$$

The first inequality is due to $Q(x) \succeq 0$ and $\nu_i \geq 0$. The second inequality derives from $\sum_{i\in I} \mu_i = Y_{n+1,n+1} = 1$, and from the concavity of the function $G(x, \,.\,)$ (Jensen inequality). Similarly, knowing that Q^j is PSD and that Y is feasible in ($\mathbf{SDP}_x$), we can show that $\frac{1}{2}\bar{y}^\top Q^j \bar{y} + (q^j)^\top \bar{y} + b_j \leq \langle \mathcal{Q}^j, Y\rangle \leq 0$, which means that $\bar{y}$ is feasible in ($\mathbf{P}_x$). This implies that $\langle \mathcal{Q}(x), Y\rangle \leq G(x, \bar{y}) \leq \mathsf{val}(\mathbf{P}_x)$. This being true for any matrix Y feasible in ($\mathbf{SDP}_x$), we conclude that $\mathsf{val}(\mathbf{SDP}_x) \geq \mathsf{val}(\mathbf{P}_x)$, hence, $\mathsf{val}(\mathbf{SDP}_x) = \mathsf{val}(\mathbf{P}_x)$. □

2.1.2 Dual SDP problem

The following SDP problem

$$\left.\begin{array}{ll} \min\limits_{\lambda,\alpha,\beta} & \alpha(1+\rho^2) + \beta \\ \text{s.t.} & \sum\limits_{j=1}^{r} \lambda_j \mathcal{Q}^j + \alpha I_{n+1} + \beta E \succeq \mathcal{Q}(x) \\ & \lambda \in \mathbb{R}^r_+,\ \alpha \in \mathbb{R}_+, \beta \in \mathbb{R}, \end{array}\right\} \tag{$\mathbf{DSDP}_x$}$$

is the dual of the problem ($\mathbf{SDP}_x$), as the following lemma states.

Lemma 2.2. *Formulations* ($\mathbf{SDP}_x$) *and* ($\mathbf{DSDP}_x$) *are a primal-dual pair of SDP problems and strong duality holds, thus* $\mathsf{val}(\mathbf{SDP}_x) = \mathsf{val}(\mathbf{DSDP}_x)$.

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

$$L_x(Y, \lambda, \alpha, \beta) = \langle \mathcal{Q}(x), Y\rangle - \sum_{j=1}^{r} \lambda_j \langle \mathcal{Q}^j, Y\rangle + \alpha(1 + \rho^2 - \langle I_{n+1}, Y\rangle) + \beta(1 - \langle E, Y\rangle) \tag{2.4}$$

$$= \alpha(1+\rho^2) + \beta + \langle \mathcal{Q}(x) - \sum_{j=1}^{r} \lambda_j \mathcal{Q}^j - \alpha I_{n+1} - \beta E, Y\rangle. \tag{2.5}$$

The Lagrangian dual problem of ($\mathbf{SDP}_x$) is $\inf\limits_{\lambda,\alpha,\beta} \sup\limits_{Y} L_x(Y, \lambda, \alpha, \beta)$ and be written as

$$\inf_{\substack{\lambda\in\mathbb{R}^r_+\\ \alpha\in\mathbb{R}_+\\ \beta\in\mathbb{R}}} \Big(\alpha(1+\rho^2) + \beta + \sup_{Y\in\mathbb{S}^+_{n+1}} \langle \mathcal{Q}(x) - \sum_{j=1}^{r} \lambda_j \mathcal{Q}^j - \alpha I_{n+1} - \beta E, Y\rangle\Big). \tag{2.6}$$

We recognize that the supremum equals to $+\infty$, unless $\sum\limits_{j=1}^{r} \lambda_j \mathcal{Q}^j + \alpha I_{n+1} + \beta E \succeq \mathcal{Q}(x)$, in which case it is zero. This proves that the dual problem of ($\mathbf{SDP}_x$) can be formulated as ($\mathbf{DSDP}_x$). To prove that $\mathsf{val}(\mathbf{SDP}_x) = \mathsf{val}(\mathbf{DSDP}_x)$, we prove that the Slater condition, which is a sufficient condition for strong duality [221], holds for the dual problem ($\mathbf{DSDP}_x$). We denote by m_x the maximum eigenvalue of $\mathcal{Q}(x)$, and we notice that $(\lambda, \alpha, \beta) = (0, \dots, 0, \max\{1 + m_x, 0\}, 0)$ is a strictly feasible point of ($\mathbf{DSDP}_x$). Hence, the Slater condition holds. □

2.2 Finite restriction of the semi-infinite program

2.2.1 Finite restriction based on lower-level duality

Leveraging on Section 2.1, which focus on the lower-level problem ($\mathbf{P}_x$), its SDP relaxation ($\mathbf{SDP}_x$) and the respective dual problem ($\mathbf{DSDP}_x$), we propose a finite restriction of problem (**SIP**). It is a reformulation of (**SIP**) if $Q^1, \dots, Q^r$ are PSD, and $Q(x)$ is PSD for any $x \in \mathcal{X}$.

Theorem 2.1. *Under Assumption 2.1, the finite formulation*

$$\left.\begin{aligned} \min_{x,\lambda,\alpha,\beta} \quad & F(x) \\ s.t. \quad & \alpha(1+\rho^2) + \beta \leq 0 \\ & \sum_{j=1}^{r} \lambda_j \mathcal{Q}^j + \alpha I_{n+1} + \beta E - \mathcal{Q}(x) \succeq 0 \\ & x \in \mathcal{X}, \lambda \in \mathbb{R}^r_+,\ \alpha \in \mathbb{R}_+, \beta \in \mathbb{R}, \end{aligned}\right\} \qquad (\mathbf{SIPR})$$

is a restriction of problem (**SIP**)*. If $Q^1, \dots, Q^r$ are PSD, and if $Q(x)$ is PSD for any $x \in \mathcal{X}$, then the finite formulation* (**SIPR**) *is an exact reformulation of* (**SIP**)*.*

Proof. Let $\mathsf{Feas}(\mathbf{SIP})$ and $\mathsf{Feas}(\mathbf{SIPR})$ be the feasible sets of (**SIP**) and (**SIPR**), respectively. Since (**SIP**) and (**SIPR**) share the same objective function, proving for any $x \in \mathcal{X}$ the implication

$$(\exists\, \lambda \in \mathbb{R}^r_+,\ \alpha \in \mathbb{R}_+,\ \beta \in \mathbb{R} :\ (x, \lambda, \alpha, \beta) \in \mathsf{Feas}(\mathbf{SIPR})) \implies x \in \mathsf{Feas}(\mathbf{SIP}), \tag{2.7}$$

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

$$\mathsf{val}(\mathbf{SDP}_x) \leq 0 \implies \mathsf{val}(\mathbf{P}_x) \leq 0 \iff x \in \mathsf{Feas}(\mathbf{SIP}), \tag{2.8}$$

where the implication stems from Lemma 2.1, which stipulates that $\mathsf{val}(\mathbf{P}_x) \leq \mathsf{val}(\mathbf{SDP}_x)$. Applying the strong duality lemma, Lemma 2.2, we obtain that for any $x \in \mathcal{X}$,

$$\mathsf{val}(\mathbf{SDP}_x) \leq 0 \iff \mathsf{val}(\mathbf{SDP}_x) \leq 0 \tag{2.9}$$

$$\iff \exists\, \lambda \in \mathbb{R}^r_+,\ \alpha \in \mathbb{R}_+,\ \beta \in \mathbb{R} : \begin{cases} \alpha(1+\rho^2) + \beta \leq 0 \\ \sum_{j=1}^{r} \lambda_j \mathcal{Q}^j + \alpha I_{n+1} + \beta E - \mathcal{Q}(x) \succeq 0 \end{cases} \tag{2.10}$$

$$\iff \exists\, \lambda \in \mathbb{R}^r_+,\ \alpha \in \mathbb{R}_+,\ \beta \in \mathbb{R},\ (x, \lambda, \alpha, \beta) \in \mathsf{Feas}(\mathbf{SIPR}). \tag{2.11}$$

The equivalence (2.11), together with implication (2.8), proves the implication (2.7).
If $Q^1, \dots, Q^r$ are PSD, and, if $Q(x)$ is PSD for any $x \in \mathcal{X}$, we can replace the implication (2.8)

by the equivalence $\mathsf{val}(\mathbf{SDP}_x) \leq 0 \iff \mathsf{val}(\mathbf{P}_x) \leq 0 \iff x \in \mathsf{Feas}(\mathbf{SIP})$. This, together with equivalence (2.11), proves that

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

meaning that (**SIPR**) is a reformulation of (**SIP**), having the same objective function. □

Note that under Assumptions 1.1 and 2.1, the single-level and finite formulation (**SIPR**) is convex.

2.2.2 Optimality of the restriction: an *a posteriori* sufficient condition

Theorem 2.1 states that the finite formulation (**SIPR**) is an exact reformulation of the problem (**SIP**), if $Q^1, \dots Q^r$ are PSD, and $Q(x)$ is PSD for all $x \in \mathcal{X}$. Even if this *a priori* condition is not satisfied for all $x \in \mathcal{X}$, an *a posteriori* condition on the computed optimal solution $\bar{x}$ of (**SIPR**) enables us to state if $\bar{x}$ is an optimal solution of (**SIP**).

Theorem 2.2. *We consider that Assumptions 1.1 and 2.1 hold, and that $Q^1, \dots, Q^r$ are PSD. Let $\bar{x}$ be an optimal solution of the formulation (**SIPR**). If $Q(\bar{x}) \succ 0$, then $\bar{x}$ is optimal in (**SIP**).*

Proof. Given a closed convex set S, according to Def. 5.1.1 in [106, Chapter III], the tangent cone to S at x (denoted by $T_S(x)$) is the set of directions $u \in \mathbb{R}^m$ such that there exist a sequence $(x_k)_{k\in\mathbb{N}}$ in S, and a positive sequence $(t_k)_{k\in\mathbb{N}}$ such that $t_k \to 0$ and $\frac{x_k - x}{t_k} \to u$. Moreover, according to Def. 5.2.4 in [106, Chapter III], the normal cone $N_S(x)$ to S at x is the polar cone of the tangent cone $T_S(x)$, i.e., $N_S(x) = T_S(x)^\circ$. We define the closed convex set C (resp. $\hat{C}$) as the feasible set of formulation (**SIP**) (resp. (**SIPR**)).

Since $Q(\bar{x}) \succ 0$, being the set of positive definite matrices open and $Q(x)$ continuous, there exists $r \in \mathbb{R}_{++}$ such that, for all x in the open ball of radius r with center $\bar{x}$ (denoted by $B(\bar{x}, r)$), $Q(x) \succeq 0$. According to Lemma 2.1, this means that, for all x in $\mathcal{X} \cap B(\bar{x}, r)$, $\mathsf{val}(\mathbf{P}_x) = \mathsf{val}(\mathbf{SDP}_x)$. Hence, we deduce that, for any $x \in \mathcal{X} \cap B(\bar{x}, r)$, x is feasible in (**SIP**) if and only if x is feasible in (**SIPR**). In other words, $C \cap B(\bar{x}, r) = \hat{C} \cap B(\bar{x}, r)$. According to the aforementioned definition of the tangent and normal cones, we further deduce that $T_C(\bar{x}) = T_{\hat{C}}(\bar{x})$, and $N_C(\bar{x}) = T_C(\bar{x})^\circ = T_{\hat{C}}(\bar{x})^\circ = N_{\hat{C}}(\bar{x})$.

We know that $\bar{x}$ is optimal in (**SIPR**), i.e., $\bar{x} \in \arg\min_{x\in\hat{C}} F(x)$. Since F is a finite-valued convex function, and $\hat{C}$ is a closed and convex set, the assumptions of Theorem 1.1.1 in [106, Chapter VII] hold, and we can deduce that $0 \in \partial F(\bar{x}) + N_{\hat{C}}(\bar{x})$. Using the equality $N_C(\bar{x}) = N_{\hat{C}}(\bar{x})$, we have that $0 \in \partial F(\bar{x}) + N_C(\bar{x})$ too. Applying the same theorem with the closed and convex set C, we know that $0 \in \partial F(\bar{x}) + N_C(\bar{x})$ implies that $\bar{x} \in \arg\min\limits_{x\in C} F(x)$, meaning that $\bar{x}$ is optimal in (**SIP**). □

2.3 Inner-outer approximation algorithm

If neither the lower-level problem is convex, nor the sufficient optimality condition in Theorem 2.2 is satisfied, we do not directly obtain an optimal solution of problem (**SIP**) by solving the finite formulation (**SIPR**). To address this issue, we design an algorithm based on the *lower-level dualization approach* and on a separation oracle, that allows us to construct a minimizing sequence of feasible solutions of problem (**SIP**).

2.3.1 Using the separation oracle to improve the lower-level dualization

The existence of a duality gap for the lower-level problem at $\bar{x}$, a solution of the finite reformulation (**SIPR**), is an obstacle to prove the optimality of $\bar{x}$ in the original problem (**SIP**). Using the output of the separation oracle at some points can nevertheless help providing a tighter relaxation and, therefore dualization of the lower-level problem. For $k \in \mathbb{N}^*$, we consider two finite sequences $x^1, \ldots, x^{k-1} \in \mathcal{X}$, and $v_1, \ldots, v_{k-1} \in \mathbb{R}$ such that v_ℓ is an upper bound on $\mathsf{val}(\mathsf{P}_{x^\ell})$, given by a δ-oracle. Since, for all $\ell \in [\![1, k-1]\!]$, the inequality

$$-\frac{1}{2} y^\top Q(x^\ell) y + q(x^\ell)^\top y + b(x^\ell) \leq v_\ell, \tag{2.12}$$

i.e., $\langle \mathcal{Q}(x^\ell), Y \rangle \leq v_\ell$ holds for any $y \in \mathcal{Y}$, the following SDP problem is still a relaxation of ($\mathbf{P}_x$), for any $x \in \mathcal{X}$:

$$\left.\begin{array}{ll} \max\limits_{Y \in \mathbb{S}_{n+1}} & \langle \mathcal{Q}(x), Y \rangle \\ \text{s.t.} & \forall j \in [\![1, r]\!],\ \langle \mathcal{Q}^j, Y \rangle \leq 0 \\ & \forall \ell \in [\![1, k-1]\!],\ \langle \mathcal{Q}(x^\ell), Y \rangle \leq v_\ell \\ & \langle I_{n+1}, Y \rangle \leq 1 + \rho^2 \\ & \langle E, Y \rangle = 1 \\ & Y \succeq 0. \end{array}\right\} \qquad (\mathbf{SDP}_x^k)$$

We recall that the value of ($\mathbf{P}_x$) is $\phi(x) = \max_{y \in \mathcal{Y}} G(x, y)$. We also denote by $\phi_{\mathsf{SDP}}(x)$ the value of ($\mathbf{SDP}_x$), and by $\phi^k_{\mathsf{SDP}}(x)$ the value of ($\mathbf{SDP}^k_x$). Note that this latter function implicitly depends on the sequences $(x^\ell)_{1 \leq \ell \leq k}$ and $(v_\ell)_{1 \leq \ell \leq k}$. Being ζ_ℓ the Lagrangian multiplier associated to the constraint $\langle \mathcal{Q}(x^\ell), Y \rangle \leq v_\ell$, the strong SDP dual of problem ($\mathbf{SDP}^k_x$) is

$$\left.\begin{array}{ll} \min\limits_{\lambda, \alpha, \beta, \zeta} & \alpha(1 + \rho^2) + \beta + \sum_{\ell=1}^{k-1} \zeta_\ell v_\ell \\ \text{s.t.} & \sum\limits_{j=1}^{r} \lambda_j \mathcal{Q}^j + \sum_{\ell=1}^{k-1} \zeta_\ell \mathcal{Q}(x^\ell) + \alpha I_{n+1} + \beta E \succeq \mathcal{Q}(x) \\ & \lambda \in \mathbb{R}^r_+,\ \alpha \in \mathbb{R}_+, \beta \in \mathbb{R}, \zeta \in \mathbb{R}^{k-1}_+. \end{array}\right\} \qquad (\mathbf{DSDP}^k_x)$$

Hence, for any $\hat{x} \in \mathcal{X}$, $\phi^k_{\mathsf{SDP}}(\hat{x}) \leq 0$ holds if and only if $\hat{x} \in \mathcal{R}^k$, where $\mathcal{R}^k$ is defined as

$$\mathcal{R}^k = \Bigg\{ \hat{x} \in \mathbb{R}^m : \exists(\lambda, \alpha, \zeta) \in \mathbb{R}^{r+k}_+, \exists \beta \in \mathbb{R}, \left(\alpha(1+\rho^2) + \beta + \sum_{\ell=1}^{k-1} \zeta_\ell v_\ell \leq 0 \right) \\ \wedge \left(\sum_{j=1}^{r} \lambda_j \mathcal{Q}^j + \sum_{\ell=1}^{k-1} \zeta_\ell \mathcal{Q}(x^\ell) + \alpha I_{n+1} + \beta E \succeq \mathcal{Q}(\hat{x}) \right) \Bigg\}.$$

Proposition 2.1. *Under Assumption 2.1, for any two finite sequences $x^1, \ldots, x^{k-1} \in \mathcal{X}$, and $v_1, \ldots, v_{k-1} \in \mathbb{R}$ such that $v_\ell \geq \phi(x^\ell)$, the resulting set $\mathcal{X} \cap \mathcal{R}^k$ is included in the feasible set of* (**SIP**).

Proof. We can apply Theorem 2.1 to the lower-level problem modified with the additional valid cuts defined by Eq. (2.12), which do not change $\phi(x)$ the value of the nonconvex lower-level problem, and, therefore, do not change the feasible set of the problem (**SIP**). The term $\alpha(1+\rho^2) + \beta \leq 0$ becomes $\alpha(1+\rho^2) + \beta + \sum_{\ell=1}^{k-1} \zeta_\ell v_\ell \leq 0$ due to the apparition of a the right-hand side constants v_ℓ in the quadratic inequalities. We deduce that $\mathcal{X} \cap \mathcal{R}^k$ is a restriction of the feasible set of the problem (**SIP**). □

2.3.2 Description of the algorithm

We now present the algorithm IOA (see Algorithm 3). It starts by solving the restriction (**SIPR**) and checks whether the condition presented in Theorem 2.2 is satisfied or not. If so, the algorithm stops returning the solution which is optimal for both (**SIPR**) and (**SIP**). Otherwise, it performs a sequence of iterations, until the stopping criteria are satisfied, i.e., $G(x^k, y^k) \leq \epsilon_1$ and $\|x^k - \hat{x}^k\| \leq \epsilon_2$. At each iteration the convex optimization problem (2.13) is solved. This problem is a *coupling* between the minimization of $F(x)$ on a relaxed set and the minimization of $F(\hat{x})$ on a restricted set. Indeed, x belongs to an outer-approximation (relaxation), whereas $\hat{x}$ belongs to an inner-approximation (restriction) of (**SIP**) feasible set (see Figure 2.1). The minimization of F over these two sets is coupled by a proximal term that penalizes the distance between x and $\hat{x}$. After solving the master problem (2.13), the lower-level problem ($\mathbf{P}_x$) is solved for $x = x^k$. The solution of this problem is used to restrict the outer-approximation, and to enlarge the inner-approximation.

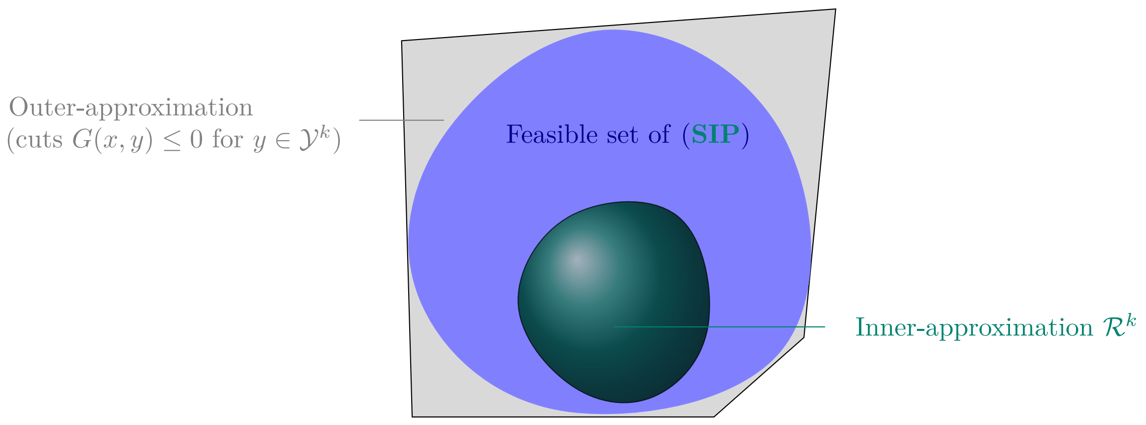

Outer-approximation
(cuts $G(x,y) \leq 0$ for $y \in \mathcal{Y}^k$)
Feasible set of (SIP)
Inner-approximation $\mathcal{R}^k$

Figure 2.1: Inner- and outer-approximations of the feasible set

Algorithm 3 Inner-outer approximation algorithm (IOA) for (**SIP**)

```
    Input: Oracle with parameter $\delta \in [0,1)$, tolerances $(\epsilon_1,\epsilon_2) \in \mathbb{R}^2_+$, bounds $\underline{\mu},\overline{\mu} \in \mathbb{R}_{++}$
 0: Solve the restriction (SIPR), to obtain a solution $\hat{x}^0$.
 1: if $Q(\hat{x}^0) \succ 0$ and $Q^1,\dots,Q^r \succeq 0$ then
 2:     Return $\hat{x}^0$.
 3: end if
 4: $k \leftarrow 0$, $\mathcal{Y}^0 \leftarrow \emptyset$, $(\nu_1^0,\nu_2^0) \leftarrow (\infty,\infty)$.
 5: while $\nu_1^k > \epsilon_1$ or $\nu_2^k > \epsilon_2$ do
 6:     Choose $\mu_k \in [\underline{\mu},\overline{\mu}]$, and compute $(x^k,\hat{x}^k)$ an optimal solution of the problem
```

$$\left.\begin{array}{rl} \min\limits_{x,\hat{x}\in\mathcal{X}} & F(x)+F(\hat{x})+\frac{\mu_k}{2}\|x-\hat{x}\|^2 \\ \text{s.t.} & \forall y \in \mathcal{Y}^k,\ G(x,y) \leq 0 \\ & \hat{x} \in \mathcal{R}^k. \end{array}\right\} \tag{2.13}$$

```
 7:     Call the $\delta$-oracle to compute an approximate solution $y^k \in \mathcal{Y} = \hat{y}(x^k)$, and an upper
        bound $v_k = \hat{v}(x^k)$ on $\phi(x^k) = \max_{y\in\mathcal{Y}} G(x^k,y)$.
 8:     Based on $x^k$ and $v_k$, update the set $\mathcal{R}^k$ in $\mathcal{R}^{k+1}$.
 9:     $\mathcal{Y}^{k+1} \leftarrow \mathcal{Y}^k \cup \{y^k\}$
10:     $(\nu_1^{k+1},\nu_2^{k+1}) \leftarrow (G(x^k,y^k), \|x^k-\hat{x}^k\|)$
11:     $k \leftarrow k+1$
12: end while
13: Return $(x^k,\hat{x}^k)$.
```

2.3.3 Convergence of the algorithm

Before proving the termination and the convergence of Algorithm 3, we introduce a technical lemma.

Lemma 2.3. *Under Assumptions 1.1 and 2.1, let us denote by x^* an optimal solution of* (**SIP**)*. If Algorithm 3 runs iteration k, $F(x^k) \leq F(x^*) + \mu_k(x^k-\hat{x}^k)^\top(x^*-x^k)$.*

Proof. We analyze the variation of the objective function with respect to the decision vector x. Since $x^* \in \mathcal{X}$ is a feasible value for x in the problem (2.13), the direction $(h, 0)$ for $h = x^* - x^k$ is admissible at $(x^k, \hat{x}^k)$ in the convex problem (2.13). As $F(x)$ is convex over $\mathbb{R}^n$, the directional derivative $F'(x^k, h) = \lim_{t \to 0^+} \frac{F(x^k+th)-F(x^k)}{t}$ is well-defined. By optimality of x^k, the directional derivative of the function $F(x)+\frac{\mu_k}{2}\|x-\hat{x}^k\|^2$ in the direction h is nonnegative, i.e., $F'(x^k, h)+\mu_k(x^k-\hat{x}^k)^\top h \geq 0$. By convexity of $F(x)$, we also have that $F(x^*)-F(x^k) \geq F'(x^k, h)$. Combining this with the previous inequality yields $F(x^k) \leq F(x^*) + \mu_k(x^k - \hat{x}^k)^\top(x^* - x^k)$. $\square$

Before proving Theorem 2.4, we assume in the following that F is Lipschitz continuous, and that the Slater condition holds for the restriction (**SIPR**). Yet, we do not require to compute the corresponding Slater point to run Algorithm 3.

Assumptions 2.2. *The function F is C_F-Lipschitz, and there exists $x^S \in \mathcal{X}$ such that $\Phi_{\mathsf{SDP}}(x^S) < 0$.*

Theorem 2.3 studies the termination of the algorithm. It states the asymptotic convergence in the case where the algorithm does not terminate. On the contrary, Theorem 2.4 studies the suboptimality of the returned solution in the case of a finite termination.

Theorem 2.3. *We consider the setting of Assumptions 1.1, 2.1, and 2.2. If $\epsilon_1, \epsilon_2 \in \mathbb{R}_{++}$, then Algorithm 3 stops after a finite number of iterations. On the contrary, if Algorithm 3 generates an infinite number of iterations, then $(\hat{x}^k)$ is a sequence of feasible solutions such that $F(\hat{x}^k) \to \mathsf{val}($**SIP**$)$.*

Proof. We reason by contrapositive: we suppose that Algorithm 3 does not stop, i.e., generates infinite sequences $(x^k)_{k\in\mathbb{N}^*}$ and $(\hat{x}^k)_{k\in\mathbb{N}^*}$, and we show that $\epsilon_1 = 0$ or $\epsilon_2 = 0$. Thanks to Lemma 0.1, we claim that

$$\phi(x^k)^+ \to 0. \tag{2.14}$$

We prove, then, that $\phi^k_{\mathsf{SDP}}(x^k)^+ \to 0$. Since $\phi^k_{\mathsf{SDP}}(x^k)^+$ is bounded, there exists at least one accumulation value ℓ for this sequence. We show that, necessarily, $\ell = 0$. We take $\psi : \mathbb{N} \to \mathbb{N}$, such that $\phi^{\psi(k)}_{\mathsf{SDP}}(x^{\psi(k)})^+ \to \ell$. Up to the extraction of a subsequence, we can assume, by compactness of $\mathcal{X}$, that $x^{\psi(k)} \to x \in \mathcal{X}$. For $k \in \mathbb{N}$, we define $j = \psi(k)$ and $\ell = \psi(k-1)$.

$$\begin{aligned}\phi^j_{\mathsf{SDP}}(x^j) =& \phi^j_{\mathsf{SDP}}(x^\ell) + \phi^j_{\mathsf{SDP}}(x^j) - \phi^j_{\mathsf{SDP}}(x^\ell) && (2.15)\\ \leq& v_\ell + \phi^j_{\mathsf{SDP}}(x^j) - \phi^j_{\mathsf{SDP}}(x^\ell), && (2.16)\end{aligned}$$

as the constraint $\langle \mathcal{Q}(x^\ell), Y\rangle \leq v_\ell$ is enforced in the problem (SDP^j_x), since $\ell \leq j-1$. We also introduce $\tilde{Y}$, the solution of SDP^j_x at $x = x^\ell$ so that, $\langle \mathcal{Q}(x^\ell), \tilde{Y}\rangle = \phi^j_{\mathsf{SDP}}(x^\ell)$; we also have that $\langle \mathcal{Q}(x^j), \tilde{Y}\rangle \leq \phi^j_{\mathsf{SDP}}(x^j)$, since $\tilde{Y}$ is feasible in this problem. Therefore, by linearity of Q, and due to the Cauchy-Schwartz inequality, $\phi^j_{\mathsf{SDP}}(x^j) - \phi^j_{\mathsf{SDP}}(x^\ell) \leq \langle \mathcal{Q}(x^j - x^\ell), \tilde{Y}\rangle \leq$

$\|\mathcal{Q}(x^j - x^\ell)\|_F\, B$ where $B = \max_{Y \in \mathsf{Feas}(\mathbf{SDP}_x)} \|Y\|_F$, which is independent from j, ℓ. Indeed, $\tilde{Y}$ is feasible in ($\mathbf{SDP}_x$). We also define $\|\mathcal{Q}\|_{op}$, the operator norm of $x \mapsto \mathcal{Q}(x)$, and we obtain that $\phi^j_{\mathsf{SDP}}(x^j) - \phi^j_{\mathsf{SDP}}(x^\ell) \le \|x^j - x^\ell\|\, \|\mathcal{Q}\|_{op}\, B$. We combine this with Eq. (2.16), using the fact that the positive part is nondecreasing, to obtain

$$\phi^j_{\mathsf{SDP}}(x^j)^+ \le v^+_\ell + \|x^j - x^\ell\|\, \|\mathcal{Q}\|_{op}\, B \tag{2.17}$$

$$\le \phi(x^\ell)^+(1+\delta) + \|x^j - x^\ell\|\, \|\mathcal{Q}\|_{op}\, B, \tag{2.18}$$

the second inequality coming from the property of the δ-oracle given by Eq. (4). Using the definition of ℓ and j, we obtain

$$0 \le \phi^{\psi(k)}_{\mathsf{SDP}}(x^{\psi(k)})^+ \le \phi(x^{\psi(k-1)})^+(1+\delta) + \|x^{\psi(k)} - x^{\psi(k-1)}\|\, \|\mathcal{Q}\|_{op}\, B. \tag{2.19}$$

Using the fact that $\phi(x^k)^+ \to 0$, as proven above, and that $x^{\psi(k)}$ is converging, we deduce by taking the limit that $\ell = 0$. We conclude that $\phi^k_{\mathsf{SDP}}(x^k)^+ \to 0$.

Using Assumption 2.2, we introduce a Slater point $x^S \in \mathcal{X}$ such that $\phi_{\mathsf{SDP}}(x^S) = -c$, for $c \in \mathbb{R}_{++}$. We also introduce $\omega_k = \phi^k_{\mathsf{SDP}}(x^k)^+/(c + \phi^k_{\mathsf{SDP}}(x^k)^+)$. We notice that $\omega_k \to 0$, since $\phi^k_{\mathsf{SDP}}(x^k)^+ \to 0$. We define the convex combination $\bar{x}_k = (1-\omega_k)x^k + \omega_k x^S$. We emphasize that $(\bar{x}_k, \bar{x}_k)$ is feasible in problem (2.13) at iteration k since $\mathcal{X}$ is convex, and

- $\bar{x}_k$ satisfies the constraints on x, because both x^k and x^S satisfy the convex constraints $G(x,y) \le 0$ for $y \in \mathcal{Y}^k$: by convex combination, so does $\bar{x}_k$. We remind that the convexity of the constraint $G(x,y) \le 0$ for each y follows from Assumption 1.1.
- $\bar{x}_k$ satisfies the constraints on $\hat{x}$; indeed $\phi^k_{\mathsf{SDP}}(x^k) \le 0$, and $\phi^k_{\mathsf{SDP}}(x^S) \le \phi_{\mathsf{SDP}}(x^S) \le 0$; by convexity of $\phi^k_{\mathsf{SDP}}(x^k)$ (as a max. of linear functions), $\phi^k_{\mathsf{SDP}}(\bar{x}_k) \le 0$, i.e., $\bar{x}_k \in \mathcal{R}^k$.

Since the objective value of $(\bar{x}_k, \bar{x}_k)$ in the problem (2.13) is $2F(\bar{x}_k)$, by optimality of $(x^k, \hat{x}^k)$: $F(x^k) + F(\hat{x}^k) + \frac{\mu_k}{2}\|x^k - \hat{x}^k\|^2 \le 2F((1-\omega_k)x^k + \omega_k x^S)$, which means by convexity of F

$$F(x^k) + F(\hat{x}^k) + \frac{\mu_k}{2}\|x^k - \hat{x}^k\|^2 \le 2(1-\omega_k)F(x^k) + 2\omega_k F(x^S). \tag{2.20}$$

We also notice that $(\hat{x}^k, \hat{x}^k)$ is feasible in the problem (2.13) at iteration k, thus $F(x^k) + F(\hat{x}^k) + \frac{\mu_k}{2}\|x^k - \hat{x}^k\|^2 \le 2F(\hat{x}^k)$, which means

$$F(x^k) + \frac{\mu_k}{2}\|x^k - \hat{x}^k\|^2 \le F(\hat{x}^k). \tag{2.21}$$

Summing Eq. (2.20) with Eq. (2.21), we obtain that $2F(x^k) + \mu_k\|x^k - \hat{x}^k\|^2 \le 2(1-\omega_k)F(x^k) + 2\omega_k F(x^S)$, and thus $\mu_k\|x^k - \hat{x}^k\|^2 \le 2\omega_k \left(F(x^S) - F(x^k)\right)$. Using that $0 < \underline{\mu} \le \mu_k$,

$$\|x^k - \hat{x}^k\| \le \sqrt{\underline{\mu}^{-1}(2\omega_k\,(F(x^S) - F(x^k)))} \tag{2.22}$$

holds. Since $\omega_k \to 0$, and $F(x^S) - F(x^k)$ is bounded, we deduce from Eq. (2.22) that

$$\|x^k - \hat{x}^k\| \to 0. \tag{2.23}$$

As the stopping criterion is not met, this means that for all $k \in \mathbb{N}$, $G(x^k, y^k) > \epsilon_1$ (and therefore $G(x^k, y^k)^+ > \epsilon_1$), or $\|x^k - \hat{x}^k\| > \epsilon_2$. As $G(x^k, y^k)^+ \leq \phi(x^k)^+ \to 0$ (see Eq. (2.14)) and $\|x^k - \hat{x}^k\| \to 0$, this means that at least one number among ϵ_1 and ϵ_2 is zero.

As $\hat{x}^k \in \mathcal{R}^k$, it is feasible in (**SIP**), as stated in Prop 2.1. Therefore, and as F is C_F-Lipschitz, we have $F(x^*) \leq F(\hat{x}^k) \leq F(x^k) + J\|x^k - \hat{x}^k\|$. According to Lemma 2.3, we know that $F(x^k) \leq F(x^*) + \mu_k(x^k - \hat{x}^k)^\top(x^* - x^k)$, which implies, according to the Cauchy-Schwartz inequality, that $F(x^*) \leq F(\hat{x}^k) \leq F(x^*) + \mu_k\|x^k - \hat{x}^k\|\|x^* - x^k\| + C_F\|x^k - \hat{x}^k\|$. Since $\|x^* - x^k\|$ is bounded, we deduce from Eq. (2.23) that $F(x^*) + \mu_k\|x^k - \hat{x}^k\|\|x^* - x^k\| + C_F\|x^k - \hat{x}^k\| \to F(x^*)$, and thus, $F(\hat{x}^k) \to F(x^*) = \mathsf{val}(\textbf{SIP})$. □

Theorem 2.4. *Under Assumptions 1.1, 2.1, and 2.2, if Algorithm 3 terminates after iteration K, it returns an iterate $\hat{x}^K \in \mathcal{X}$ feasible in* (**SIP**)*, and such that $F(\hat{x}^K) \leq \mathsf{val}(\textbf{SIP}) + \epsilon_2(\mu_K \mathsf{diam}(\mathcal{X}) + C_F)$.*

Proof. Due to Proposition 2.1, we know that $\hat{x}^K$ is feasible in (**SIP**). From Lemma 2.3, and from Cauchy-Schwartz inequality, we know that $F(x^K) \leq F(x^*) + \mu_k(x^K - \hat{x}^K)^\top(x^* - x^K) \leq F(x^*) + \mu_k\|x^K - \hat{x}^K\|\,\|x^* - x^K\|$. Using also that F is C_F-Lipschitz, we obtain $F(\hat{x}^K) \leq F(x^*) + \mu_K\|x^K - \hat{x}^K\|\,\|x^* - x^K\| + C_F\|x^K - \hat{x}^K\|$. Finally, we notice that $\|x^K - \hat{x}^K\| \leq \epsilon_2$ and $\|x^* - x^K\| \leq \mathsf{diam}(\mathcal{X})$ to conclude. □

2.4 Applications and numerical experiments

In this section, we present two problems that fit in the setting of formulation (**SIP**), and the corresponding numerical experiments.

2.4.1 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:

$$z = \frac{1}{2}w^\top \bar{Q} w + \bar{q}^\top w + \bar{b} + \epsilon, \text{ where } \bar{Q} \in \mathbb{S}_n,\ \bar{q} \in \mathbb{R}^n,\ \bar{b} \in \mathbb{R} \text{ and } \epsilon \sim \mathcal{N}(0, \sigma^2).$$

Let us suppose that the parameters of this model are unknown, except an *a priori* bound $\rho' \in \mathbb{R}_+$ on their magnitude. Given a dataset $(w_i, z_i)_{1 \leq i \leq P} \in (\mathbb{R}^n \times \mathbb{R})^P$, the problem of finding the maximum likelihood estimator for $\bar{Q} \in \mathbb{R}^{n \times n}$, $\bar{q} \in \mathbb{R}^n$, $\bar{b} \in \mathbb{R}$ consists in computing the triplet $(Q, q, b) \in \mathbb{R}^{n \times n} \times \mathbb{R}^n \times \mathbb{R}$ that minimizes the least-squares error $\sum_{i=1}^{P}(z_i - \frac{1}{2}w_i^\top Q w_i - q^\top w_i - b)^2$. We consider that

1. the features vector belongs to a polytope $\mathcal{Y} = \{y \in \mathbb{R}^n : Ty \leq u\}$, for a given matrix $T \in \mathbb{R}^{r \times n}$ and vector $u \in \mathbb{R}^r$, and we know $\rho \geq 0$ such that $\mathcal{Y} \subset B(0, \rho)$,
2. the noiseless value $\frac{1}{2}y^\top \bar{Q} y + \bar{q}^\top y + \bar{b}$ is nonnegative for any $y \in \mathcal{Y}$.

Hence, this inverse problem is a "constrained quadratic regression problem" that reads

$$\left.\begin{array}{ll}\min\limits_{Q,q,b} & \sum\limits_{i=1}^{P}(z_i - \frac{1}{2}w_i^\top Q w_i - q^\top w_i - b)^2 \\ \text{s.t.} & \forall y \in \mathcal{Y},\ \frac{1}{2}y^\top Q y + q^\top y + b \geq 0 \\ & Q = Q^\top \\ & \|Q\|_\infty \leq \rho',\ \|q\|_\infty \leq \rho',\ |b| \leq \rho' \\ & Q \in \mathbb{R}^{n\times n},\ q \in \mathbb{R}^n,\ b \in \mathbb{R}. \end{array}\right\} \tag{2.24}$$

Formulation (2.24) is a semi-infinite program, that fits in the general setting of the formulation (**SIP**), with the decision variable is the triplet $x = (Q, q, b) \in \mathbb{R}^{n\times n} \times \mathbb{R}^n \times \mathbb{R}$, and satisfies Assumptions 1.1, 2.1, and 2.2. In the numerical experiments, we compare the CPA and IOA approaches with the global optimization algorithm proposed in [161]. We also compare these algorithms with the solution of the relaxation/reformulation (depending on the convexity of the lower-level problem) obtained by replacing the lower-level problem by its KKT conditions [2]:

$$\left.\begin{array}{ll}\min\limits_{Q,q,b,y,\gamma} & \sum\limits_{i=1}^{P}(z_i - \frac{1}{2}w_i^\top Q w_i - q^\top w_i - b)^2 \\ \text{s.t.} & \frac{1}{2}y^\top Q y + q^\top y + b \geq 0 \\ & Q = Q^\top \\ & Ty \leq u \\ & Qy + q + T^\top \gamma = 0 \\ & \gamma^\top (Ty - u) = 0 \\ & \|Q\|_\infty \leq \rho',\ \|q\|_\infty \leq \rho',\ |b| \leq \rho' \\ & Q \in \mathbb{R}^{n\times n},\ q \in \mathbb{R}^n,\ b \in \mathbb{R},\ y \in \mathbb{R}^n,\ \gamma \in \mathbb{R}^r_+, \end{array}\right\} \tag{2.25}$$

where γ is the KKT multiplier vector associated to the lower-level constraints $Ty \leq u$. Problem (2.25) is a nonconvex polynomial problem involving polynomials of degree up to three.

2.4.2 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 $\mathcal{G} = (V, E)$. 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 such that, 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 $P_i(x, y)$ as 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 a matrix having $M_{ij} = 1$ 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: $c_1(x) = \frac{1}{2}x^\top Hx + h_1^\top x$, with $H \in \mathbb{S}_n^+$ and $h_1 \in \mathbb{R}^n$,
- 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: $c_2(x, y) = \frac{1}{2}y^\top Q(x)y + h_2^\top y$, where $Q(x) \in \mathbb{R}^{n\times n}$ is a linear mapping and $h_2 \in \mathbb{R}^n$ is a vector.

Hence, this zero-sum game can be written as $\min\limits_{x\in\Delta_n}\max\limits_{y\in\Delta_n} \; -x^\top My + c_1(x) - c_2(x, y)$. 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 semi-infinite programming formulation:

$$\left.\begin{array}{ll} \min\limits_{x,z} & \frac{1}{2}x^\top Hx + h_1^\top x + z \\ \text{s.t.} & \forall y \in \Delta_n,\ \frac{1}{2}y^\top Q(x)y + (h_2 + M^\top x)^\top y + z \geq 0 \\ & x \in \Delta_n,\ z \in \mathbb{R}. \end{array}\right\} \tag{2.26}$$

This formulation clearly fits in the general setting of formulation (**SIP**), and satisfies Assumptions 1.1, 2.1, and 2.2. As for the first application, we benchmark the CPA and IOA approaches with the algorithm proposed by Mitsos in [161], and the KKT relaxation approach [2]. Given the KKT multipliers γ_1 and γ_2 associated respectively to the lower-level constraints $\sum\limits_{i=1}^{n} y_i = 1$ and $y \geq 0$, the finite formulation obtained by replacing the lower level of (2.26) by its KKT conditions, is

$$\left.\begin{array}{rl} \min\limits_{x,z,y,\gamma_1,\gamma_2} & z + \frac{1}{2}x^\top Hx + h_1^\top x \\ \text{s.t.} & \frac{1}{2}y^\top Q(x)y + (h_2 + M^\top x)^\top y + z \geq 0 \\ & Q(x)y + q(x) + M^\top x + \gamma_1\mathbf{1} - I_n\gamma_2 = 0 \\ & -\gamma_2^\top (I_n y) = 0 \\ & x \in \Delta_n,\ y \in \Delta_n,\ z \in \mathbb{R},\ \gamma_1 \in \mathbb{R},\ \gamma_2 \in \mathbb{R}_+^n. \end{array}\right\} \tag{2.27}$$

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 (2.27). This relaxation/reformulation of problem (2.26) is a nonconvex polynomial optimization problem involving multivariate polynomials of degree up to three.

2.4.3 Experimental protocol

Implementation of the algorithms The global solutions of the semi-infinite programs (2.24) and (2.26) are computed using the algorithms CPA and IOA, and the algorithm proposed in [161], that we call "Mitsos Algorithm". We also benchmark these algorithms with the KKT relaxation/reformulation approach [2].

- The algorithm CPA is implemented using the programming language `python 3`. Both the master problem (**$\mathbf{R_k}$**) and the lower-level problem (**$\mathbf{P}_x$**) are solved using the global QP solver `Gurobi 9.1` [94]. The tolerance for the feasibility error ϵ is set to 10^{-6}.
- The algorithm IOA is also implemented in Python 3. We use the conic programming solver `MOSEK 9.3` [168] to solve (**SIPR**) at step 1, as well as the master problem (2.13) at step . The nonconvex QCQP solver in `Gurobi 9.1` [94] is used to implement the oracle, i.e., to solve the problem (**$\mathbf{P}_x$**) with relative optimality gap $\delta = 10^{-4}$. The tolerances ϵ_1 and ϵ_2, used in the stopping criteria, are set to 10^{-6}. As previously seen, an *a priori* knowledge on the convex nature of the lower-level problem would give the guarantee that the formulation (**SIPR**) has the same value as the formulation (**SIP**): in one such case, solving (**SIPR**) yields an optimal solution of (**SIP**). Yet, this prior knowledge is not common to all the possible instances considered here, and we decide to treat all in the same way. For this reason, we run the sequence of instructions described in Algorithm 3, without any *a priori* information about the convexity of the lower-level problem.
- Mitsos algorithm [161] is implemented using `python 3`. This general algorithm, proposed for the global solution of semi-infinite programs without convexity assumptions, generates a lower and an upper bound of the optimal value of (**SIP**) at each iteration. It stops when the gap is closed, or if the relaxation (solved to get the lower bound) yields a feasible solution. This relaxation is obtained by approximating the infinite set $\mathcal{Y}$ by a progressively finer finite subset, as in the CPA algorithm. The formulation solved to get an upper bound is obtained by restricting the infinite constraints right hand side by $\epsilon^g \in \mathbb{R}_{++}$ and considering a successively finer discretization of $\mathcal{Y}$. For arbitrary combinations of the discretized parameter set and ϵ^g, this formulation is neither a restriction, nor a relaxation of the semi-infinite program. However, the existence of a strictly feasible point ensures that the algorithm finitely generates feasible iterates, the objective value of which converges to the optimal value. At each iteration, both the relaxation, and the restriction of the semi-infinite program, as well as the lower-level problem are solved using `Gurobi 9.1`.
- We implement the KKT relaxation/reformulation approach [2], presented at Eq. (2.25) and Eq. (2.27), with the `AMPL` modeling language [75], and solve them using `Gurobi 9.1`. These formulations are particularly hard to solve, mainly because of the complementarity constraints. Indeed, for most of the tested instances, `Gurobi 9.1` does not terminate within the time limit. For these instances, we just display, in italic font, the lower bound given by the optimal value of the best relaxation of the KKT formulation found by `Gurobi` within the time limit.

For all the approaches, we run `Gurobi 9.1` with its default settings. The tests were performed on a computer with a 2.70GHz Intel(R) Core(TM) i7 quad-core and with 16 GB of RAM. For all the approaches we set a time limit (t.l.) of 18,000 seconds (5 hours).

Data generation For the constrained quadratic regression (Section 2.4.1), we solve twenty randomly generated instances. Each of these instances is generated by choosing the statistical parameters $\bar{Q}, \bar{q}, \bar{c}$ at random, drawing $P = 4,000$ 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. The data $(w_i, z_i)_{1 \leq i \leq P}$ are produced with a PSD matrix $\bar{Q}$ for ten instances, named *PSD_inst#* in Table 2.1, and are produced with an indefinite $\bar{Q}$ for ten instances, named *notPSD_inst#* in Table 2.1. For the zero-sum game with cubic payoff application (Section 2.4.2), we test twenty-two instances where the matrix M is taken from the DIMACS graph coloring challenge[1]. We randomly generate H in a way such that it is PSD, as well as the coefficients of the linear function $Q(x)$ such that $Q(x)$ is PSD for all feasible x in the instances named *#_PSD* in Table 2.2. Regarding the instances named *#_notPSD* in Table 2.2, no particular precaution is taken to enforce that $Q(x)$ is PSD. Hence, the sign of the eigenvalues of $Q(x)$ depends on x. The data, the data generation code, the implementation of the algorithms, and the numerical results are available online at the public repository `https://github.com/aoustry/SIP-with-QP-LL`.

2.4.4 Numerical results

The results for both applications are reported in Tables 2.1 and 2.2, respectively. The headings are the following:

- "m" is the number of decision variables;"n" is the number of semi-infinite constraint parameters, i.e., lower-level variables; "time(s)" is the computing time in seconds; "it" is the number of iterations; for the IOA algorithm, this number is 0 when the condition of Theorem 2.2, checked at step 1, is satisfied and the algorithm does not enter the loop;
- for CPA and Mitsos algorithms, "obj/*LB*-*UB*" is, respectively, either the value of the solution obtained at termination, or a pair of values corresponding to: the best lower bound (*LB*) and the best feasible solution, i.e., upper bound (*UB*), found within the time limit;
- for IOA, "obj/*UB*" is, respectively, either the value of the solution obtained at termination, or the best value $F(\hat{x}^k)$ found by the algorithm within the time limit;
- for CPA and IOA algorithms "% oracle" is the share of the total computing time dedicated to the oracle, i.e., used to solve the lower-level problem ($\mathbf{P}_x$);
- for the KKT approach, "obj/*LB*" is, respectively, either the optimal value of the KKT relaxations (Eq. (2.25) for the first application and Eq. (2.27) for the second), or the best lower bound of such value found by the solver `Gurobi 9.1` within the time limit, which also is a lower bound for $\mathsf{val}(\mathbf{SIP})$.

In Tables 2.1 and 2.2, the minimum computing times are reported in bold for each instance.

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

Instances		Algorithm 2 (CPA)				Algorithm 3 (IOA)				Mitsos algorithm			KKT
Name	m/n	obj/LB–UB	time(s)	it	% oracle	obj/UB	time(s)	it	% oracle	obj/LB–UB	time(s)	it	obj/LB
PSD_inst1	31/5	358.64	0.70	6	3.4	358.64	**0.27**	0	-	358.64	1.26	6	*355.78*
PSD_inst2	31/5	365.60	0.32	3	3.6	365.60	**0.23**	0	-	365.60	0.56	3	*363.85*
PSD_inst3	31/5	363.43	0.91	8	3.4	363.43	**0.22**	0	-	363.43	1.78	8	*359.16*
PSD_inst4	31/5	353.90	0.54	5	3.6	353.90	**0.22**	0	-	353.90	0.97	5	*353.19*
PSD_inst5	111/10	391.21	4.81	17	1.1	391.21	**0.60**	0	-	391.21	10.14	17	*359.48*
PSD_inst6	111/10	397.59	4.92	17	1.0	397.59	**0.63**	0	-	397.59	10.45	17	*353.55*
PSD_inst7	183/13	440.84	8.70	19	0.7	440.84	**1.01**	0	-	440.84	25.1	19	*358.19*
PSD_inst8	183/13	382.17	**2581**	17	99.7	382.17	2734	15	98.6	382.17	6193	17	*345.52*
PSD_inst9	241/15	*564.84 – 622.88*	t.l.	5	100	572.77	**1.62**	0	-	*564.84 – inf*	t.l.	5	*351.95*
PSD_inst10	241/15	*526.22 – 545.34*	t.l.	8	100	528.93	**1.44**	0	-	*526.22 – inf*	t.l.	8	*346.43*
notPSD_inst1	31/5	358.47	**0.22**	2	4.4	358.47	2.03	4	0.8	358.47	0.28	2	*345.12*
notPSD_inst2	31/5	378.28	**0.22**	2	4.95	378.28	2.04	4	0.5	378.28	0.28	2	370.89
notPSD_inst3	31/5	345.81	**0.12**	1	3.5	345.81	0.66	1	0.3	345.81	0.18	1	345.81
notPSD_inst4	31/5	353.25	**0.11**	1	4.3	353.25	1.10	2	0.3	353.25	0.14	1	353.25
notPSD_inst5	111/10	503.88	**5.17**	18	8.4	503.88	32.2	18	1.3	503.88	11.3	18	*360.42*
notPSD_inst6	111/10	482.96	**31.6**	36	68.0	482.96	84.2	35	32.4	482.96	65.4	36	*357.48*
notPSD_inst7	183/13	647.08	**119**	61	77.2	647.08	211	54	37.8	647.08	263	61	*351.31*
notPSD_inst8	183/13	588.19	**566**	77	92.8	588.19	700	74	73.6	588.19	977	77	*358.28*
notPSD_inst9	241/15	1126.44	**687**	97	89.9	1126.44	922	92	63.4	1126.44	1356	97	*345.44*
notPSD_inst10	241/15	580.60	**595**	64	92.0	580.60	711	60	70.9	580.60	1047	64	*350.60*

Table 2.1: Numerical results of the 1st application (constrained quadratic regression)

Instances		Algorithm 2 (CPA)				Algorithm 3 (IOA)				Mitsos algorithm			KKT
Name	$m = n$	obj/LB–UB	time(s)	it	% oracle	obj/UB	time(s)	it	% oracle	obj/LB–UB	time(s)	it	obj/LB
jean_PSD	80	-0.0760	49.5	183	23.5	-0.0760	**19.9**	0	-	-0.0760	128	171	*-1.0121*
myciel4_PSD	23	-0.3643	4.81	390	31.0	-0.3643	**0.09**	0	-	-0.3643	139	371	*-1.0154*
myciel5_PSD	47	-0.3164	21.6	684	13.1	-0.3164	**1.51**	0	-	-0.3164	580	633	*-1.0171*
myciel6_PSD	95	-0.2841	399	2203	2.8	-0.2841	**42.0**	0	-	-0.2841	6738	2008	*-1.0207*
myciel7_PSD	191	-0.2608	7498	5586	0.5	-0.2608	**3452**	0	-	*-0.2608 – -0.2608*	t.l.	3268	*-1.9246*
queen5_5_PSD	25	-0.5536	1.73	165	39.4	-0.5536	**0.12**	0	-	-0.5536	19.7	151	*-1.0163*
queen6_6_PSD	36	-0.4619	8.98	511	22.3	-0.4619	**0.37**	0	-	-0.4619	168	458	*-1.0185*
queen7_7_PSD	49	-0.4054	31.0	937	12.1	-0.4054	**1.60**	0	-	-0.4054	602	863	*-1.0204*
queen8_8_PSD	64	-0.3614	97.0	1578	7.0	-0.3614	**4.43**	0	-	-0.3614	1662	1416	*-1.0215*
queen8_12_PSD	96	-0.3000	1194	4138	1.9	-0.3000	**36.4**	0	-	-0.3000	14153	3570	*-1.0217*
queen9_9_PSD	81	-0.3247	351	2637	3.4	-0.3247	**14.9**	0	-	-0.3247	5027	2357	*-1.0216*
jean_notPSD	80	2.3979	**6.82**	7	99.5	2.3979	195	8	4.0	2.3979	16.4	7	*1.4095*
myciel4_notPSD	23	0.5198	**43.5**	40	99.8	0.5198	52.8	41	82.9	0.5198	102.6	40	*-0.2441*
myciel5_notPSD	47	1.2779	**42.4**	37	99.7	1.2779	86.3	33	35.3	1.2779	103	37	*0.3167*
myciel6_notPSD	95	2.9378	**236**	35	99.9	2.9378	2223	38	11.8	2.9378	615	35	*1.7319*
myciel7_notPSD	191	6.2486	**773**	23	99.9	*6.2932*	t.l.	4	0.06	6.2486	1320	23	*-9.2171*
queen5_5_notPSD	25	0.3800	**21.2**	51	99.4	0.3800	29.5	44	57.2	0.3800	42.4	51	*-0.3318*
queen6_6_notPSD	36	0.8511	**293**	73	99.9	0.8511	350	68	81.5	0.8511	751	73	*-0.0377*
queen7_7_notPSD	49	1.3510	**69.8**	44	99.7	1.3510	161	40	40.7	1.3510	174	44	*0.3615*
queen8_8_notPSD	64	1.8122	**543**	33	100	1.8122	1001	42	70.1	1.8122	1113	33	*0.7866*
queen8_12_notPSD	96	2.8102	**1049**	34	100	2.8102	2525	32	32.4	2.8102	1935	34	*1.6273*
queen9_9_notPSD	81	2.2979	**2424**	46	100	2.2979	2613	39	69.9	2.2979	4545	46	*1.2042*

Table 2.2: Numerical results of the 2nd application (zero-sum game)

A first satisfactory observation is that CPA, IOA, and the Mitsos algorithm, when they converge before the time limit, effectively return the same value for the problem (**SIP**). As for the KKT approach, we observe that it does provide a lower bound, which is tight for only two instances. For most cases, `Gurobi 9.1` does not manage to solve the KKT reformulation, which is particularly numerically challenging mainly because of the complementarity constraints. This results in poor lower bounds.

In terms of computational time, IOA is more efficient than the other approaches for all the instances where the restriction is proven to be optimal during the preliminary step of this algorithm: for 14 (resp. 17) instances, IOA is at least 5 times faster than CPA (resp. Mitsos algorithm), and up to 12,500 times faster (PSD_inst10). When solving the other instances, CPA shows the best performance, although the number of iterations needed by the three methods is always comparable. Indeed, IOA iterations are, in average, more time consuming than the other algorithms and hence, for these instances, the computational time for IOA is larger even if the number of iterations of IOA is often less with respect to CPA and Mitsos algorithm. As regards Mitsos algorithm, it turns out to be slower than CPA. Comparing Mitsos algorithm and IOA on these instances with nonconvex lower-level problem, Table 2.3 shows that the performance of both algorithms are balanced. We recall that, as the algorithm IOA, Mitsos algorithm computes a sequence of feasible solutions, the value of which converges to the optimal value, whereas the iterates of CPA are only asymptotically feasible.

Instances		Algorithm 3 (IOA)	Mitsos algorithm
First application	PSD_inst#	100%	0%
	notPSD_inst#	40%	60%
Second application	#_PSD	100%	0%
	#_notPSD	60%	40%

Table 2.3: Percentage of instances for which IOA or Mitsos algorithm is faster

The instance "PSD_inst8" is of particular interest, since the restriction (**SIPR**) is obviously optimal (its value is also 382.17), but IOA is not able to detect it at step 1, since the matrix $Q(\hat{x}^0)$ is not positive definite, but only positive semidefinite. Therefore, the algorithm needs to enter the loop and runs 15 iterations before stopping.

To understand the computational time required by CPA and IOA, we can look at "% oracle" columns of Table 2.1 and 2.2. As regards CPA, for the first application, the time required to solve the lower-level problem ($\mathbf{P}_x$) is longer than the time required to solve the master problem only for the biggest instances. Indeed, when n grows, more time is needed to solve a possibly nonconvex quadratic programming problem of size n, rather than a convex master problem of size $m = n^2 + n + 1$. When n is small, this is different: even if the inner problem is quadratic nonconvex, it has a small size so it is not harder to solve than the convex master problem. For the second application, the time required to solve the lower-level problem is longer than the time required to solve the master problem only for the instances having a nonconvex lower-level problem, i.e., the second half of Table 2.2. Indeed, when $Q(x^k)$ is not PSD, problem ($\mathbf{P}_x$) is possibly nonconvex and it becomes harder to solve than the master problem. As regards IOA, we see that the percentage of time required to solve problem ($\mathbf{P}_x$) depends on the instance. Actually, the computational difficulty of the lower-level problem may also vary, for a same instance, between iterates, depending, e.g., on the number of the negative eigenvalues of $Q(x^k)$. In general, the value in the column "% oracle" for the IOA is always less than the corresponding value in the column of CPA. As expected, this comes from the fact that the master problem solved at each iteration is larger for IOA than for CPA.

2.5 Conclusion

This chapter addresses a special case of semi-infinite problems where the lower-level problem is a QCQP, potentially nonconvex. We propose a new algorithm that exploits this structure to generate a (globally) minimizing sequence of feasible iterates. A particular feature of this algorithm is that it uses the output of the separation oracle to tighten the outer approximation and extend the inner approximation. Another feature of this algorithm is the ability to detect *a posteriori* the convexity of the lower-level problem ($\mathbf{P}_x$), at $x = \hat{x}^0$ being the first iterate. This condition is sufficient for an early stop of the algorithm: in this case, it has found the optimal solution of the semi-infinite problem — even if the lower-level problem is nonconvex for other values of x. If this condition is met, this yields a clear advantage in terms of computation time, compared to the cutting-plane algorithm [31, 118] and Mitsos' algorithm [161].

An avenue of research consistent with this chapter and with Chapter 1 is to study the convergence speed of this algorithm. The same assumptions as in Chapter 1, or less stringent ones, could be used. We can wonder whether the algorithm IOA may have a convergence rate in $O(1/k)$ without assuming the strong convexity of the objective function as we did in Chapter 1 for the algorithm CPA.

From a practical point of view, we could study the possibility of warm-starting the solution of the master problem, which is a SDP problem, at each iteration. Indeed, the computational cost of the master problem has been identified as a limitation of the IOA algorithm for large instances, and a warm start could reduce this cost.

Chapter 3

Minimal time nonlinear control via convex semi-infinite programming

This chapter deals with the control of a deterministic dynamical system. We consider the general case of a time-dependent nonlinear system under nonlinear state constraints:

$$\dot{x}(t) = f(t, x(t), u(t)), (x(t), u(t)) \in X \times U, \quad \text{a. e. on } [0, T]. \tag{3.1}$$

The objective of the control is to reach a target set $K \subset X$ in a minimum time. Several applications in various fields, such as robotics [113], aerospace [216], maritime routing [156] or medicine [241], can be formulated as minimal time control problems. The approach proposed here is based on convex semi-infinite programming and, more particularly, on the developments of Chapter 1 about the algorithm CPA (Algorithm 2). In addition to introducing a hierarchy of semi-infinite programs for solving the dual problem and computing approximate value functions, this chapter studies the existence and the performance of closed-loop trajectories following a feedback controller based on an approximate value function.

Related works

Minimal time control, also known as time optimal control, can be seen as a special case of the general framework of Optimal Control Problems (OCP). Solving an OCP for such generic dynamics and constraints is a difficult challenge, although deep theoretical tools are available such as the Pontryagin Maximum Principle [34, 52, 193] and the Hamilton-Jacobi-Bellman (HJB) equation [60, 76]. Those theoretical tools, initially developed in the unconstrained setting, have been extended to the case of state constraints [41, 203]. From a numerical point of view, the *multiple shooting* techniques [187, 227] are based on the Pontryagin Maximum Principle and reduce to the solution of a two-point boundary value problem. The *direct methods* reduce to the solution a nonlinear programming problem after discretizing the time space, or parameterizing the control $u(t)$ in a finite-dimensional subspace [216, 227]. The celebrated Model Predictive Control approach belongs to the category of direct methods [38]. Another approach is to compute the value function of the problem as a maximal subsolution of the HJB equation [226]. This approach is related to the weak formulation of the OCP, an infinite-dimensional LP involving occupation measures. The dual problem of this LP is exactly the problem of finding a maximal subsolution of the HJB equation [102, 134]. In [134], the Moment SoS hierarchy is used to approximate the solution of the resulting infinite dimensional LPs, in the case where the dynamics and the constraints of the OCP are defined by polynomials. The convergence rate of this numerical scheme is studied in [127] for infinite-time discounted polynomial control problems. Still in the context of polynomial control problems, a work [115] based on the dual LP and the SoS hierarchy also studies the design of a closed-loop controller based on the approximate value function that is computed. In [19], an extension of the SoS hierarchy based on Kernel methods is employed to extend this computation to a general nonlinear system. Regarding the methods specifically dedicated to time optimal control, we find the same categories: direct methods such as Model Predictive Control [224], indirect methods based on the Pontryagin Maximum Principle and the bang-bang property [146, 172] or methods based on convex optimization [141].

Contributions and organization of the chapter

In this chapter, we focus on the problem of computing a control to reach a target set in a minimal time. We follow the line of works that use convex optimization to solve the dual problem of the nonlinear control problem over the subsolutions of the HJB equation [102, 127, 134, 226]. In contrast to several works using the Moment SoS hierarchy [115, 127, 134, 181, 199], the dynamical system and the state constraints considered here are generic and, in particular, are not assumed to be defined by polynomials. Instead of using polynomial optimization theory and the associated positivity certificates, our approach relies on the existence of a *separation oracle* capable of returning, for a given differentiable function V, a point (t, x) where the function V does not satisfy the HJB inequality. Such an oracle can be provided by a global optimization solver or by a sampling scheme in a black-box optimization approach. In particular, our approach is compatible with the sampled-data control paradigm [19, 34, 126]. Our contribution is manifold

- We introduce a hierarchy of linear semi-infinite programs, the values of which converge to the value of the control problem. After regularization, we solve these semi-infinite programs using the algorithm CPA with a convergence rate in $O(\frac{1}{k})$, where k is the number of calls to the oracle. This yields subsolutions of the HJB equation that lower-approximate the value function and provides a certified lower bound on the minimum time.

- It is known that one can leverage any function $V(t,x)$ approximating the value function, to design a closed-loop, *i.e.*, feedback controller [101, 115]. In this chapter, we study the existence of trajectories generated by such a controller.
- We study the performance of such a closed-loop controller, depending on how well $V(t,x)$ approximates the value function in a way distinct from the analysis in [115]. In particular, this novel analysis enables us to give a sufficient condition for the controller to effectively generate a trajectory reaching the target set within the considered time horizon.
- We perform numerical experiments on three non-polynomial controlled systems and compute lower and upper bounds on the minimum time.

In Section 3.1, we state the minimal time control problem and its dual infinite-dimensional linear programming formulation(s). Section 3.2 introduces the hierarchy of convex semi-infinite programming problems and the solution algorithm to obtain near-optimal HJB subsolutions. Based on these approximate value functions, we obtain a feedback controller and study its trajectories' existence and performance in Section 3.3. We provide numerical experiments on three different non-polynomial systems in Section 3.4.

3.1 Problem statement and LP formulations

3.1.1 Definition of the minimal time control problem

Let n and m be nonzero integers. We consider on $\mathbb{R}^n$ the control system

$$\dot{x}(t) = f(t, x(t), u(t)), \tag{3.2}$$

where $f\colon \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is Lipschitz continuous, and where the controls are bounded measurable functions, defined on intervals $[t_0, t_1] \subset [0, T]$, and taking their values in a compact set U of $\mathbb{R}^m$. Let X and $K \subset X$ be compact sets of $\mathbb{R}^n$ and $x_0 \in \mathbb{R}$. For $t_0, t_1 \geq 0$, a control u is said admissible on $[t_0, t_1]$ whenever the solution $x(.)$ of (3.2), such that $x(t_0) = x_0$, is well defined on $[t_0, t_1]$ and satisfies the constraints

$$(x(t), u(t)) \in X \times U, \quad \text{a. e. on } [t_0, t_1], \tag{3.3}$$

and satisfies the terminal state constraint

$$x(t_1) \in K. \tag{3.4}$$

We denote by $\mathcal{U}(t_0, t_1, x_0)$ the set of admissible controls on $[t_0, t_1]$. We consider the question of the minimal time problem from x_0 to K,

$$V^*(t_0, x_0) = \inf_{\substack{t_1 \in [t_0, T] \\ u(\cdot) \in \mathcal{U}(t_0, t_1, x_0)}} t_1 - t_0. \tag{3.5}$$

This is a particular case of the OCP with free final time [134], associated with the cost $\int_{t_0}^{t_1} \ell(t, x(t), u(t))dt$ for $\ell(t, x(t), u(t)) = 1$. The function V^* is called the value function of this minimal time control problem: this describes the smallest time to reach the target set K, starting from x_0 at time t_0.

Assumptions 3.1. *For any $(t, x) \in [0, T] \times X$, the set $f(t, x, U)$ is convex.*

We underline that we do not have any convexity assumption on sets X and K.

Remark 3.1. *Even if the dynamical system of interest does not satisfy Assumption 3.1, we can apply the present analysis to the convexified inclusion $\dot{x}(t) \in \mathsf{conv}\, f(t, x(t), U)$. According to the Filippov-Ważewski relaxation Theorem [8, Th. 10.4.4], the trajectories of the original control problem are dense in the set of trajectories of the convexified inclusion. The trajectories of the convexified inclusion may be seen as the limit of chattering trajectories, i.e., when the control oscillates infinitely fast and where the constraint set is infinitesimally dilated.*

Theorem 3.1. *Under Assumption 3.1, the minimal time control problem* (3.2)-(3.5) *associated with a starting point $(t_0, x_0) \in [0, T] \times X$ is either infeasible or admits an optimal trajectory.*

Proof. We consider the case where a feasible trajectory exists. This is a direct application of [226, Th. 2.1], which, among others, characterizes the existence of an optimal trajectory for a control problem over a differential inclusion. To emphasize the correspondence with the notation of [226], we highlight that we apply the theorem with: the running cost function $\ell(t, x, p) = 1$, the terminal cost function $g(t, x) = 0$, the set-valued map $F(t, x) = f(t, x, U)$, the constraint set $A = [0, T] \times X$ and the target set $C = [0, T] \times K$. We underline that the assumptions (H1)-(H5) in [226] are satisfied here; more precisely, we highlight that our Assumption 3.1 enforces (H2) and the hypothesis that a feasible trajectory exists enforces (H4). □

3.1.2 Hamilton-Jacobi-Bellman equation and subsolutions

In optimal control theory, a well-known sufficient condition for a function V to be the value function V^* is to satisfy the HJB Partial Differential Equation (PDE). This PDE can be seen as a continuous time generalization of Bellman's dynamic programming optimality principle in discrete time [11]. In our minimal time control setting, the HJB equation reads

$$\forall (t, x) \in [0, T] \times X, \quad \partial_t V(t, x) + \min_{u \in U}\{1 + \nabla_x V(t, x)^\top f(t, x, u)\} = 0 \tag{3.6}$$

$$\forall (t, x) \in [0, T] \times K, \quad V(t, x) = 0. \tag{3.7}$$

In general, differentiable solutions of this PDE may not exist, so the concept of viscosity solutions is typically used [60]. Another approach to get around the lack of a differentiable solution to the HJB equation consists in leveraging the concept of subsolutions [226], *i.e.*, functions $V \in C^1(\mathbb{R}^{n+1})$ satisfying the following inequalities:

$$\forall(t,x) \in [0,T] \times X, \quad \partial_t V(t,x) + \min_{u \in U}\{1 + \nabla_x V(t,x)^\top f(t,x,u)\} \geq 0 \tag{3.8}$$

$$\forall(t,x) \in [0,T] \times K, \quad V(t,x) \leq 0. \tag{3.9}$$

The following lemma states a standard result: any subsolution of the HJB PDE is an under-approximation of the value function.

Lemma 3.1. *For any $V \in C^1(\mathbb{R}^{n+1})$ satisfying Eqs. (3.8)-(3.9), the following holds:*

$$\forall(t,x) \in [0,T] \times X, \qquad V(t,x) \leq V^*(t,x).$$

Proof. We take any $(t,x) \in [0,T] \times X$ and we consider that $V^*(t,x) < \infty$, since the case $V^*(t,x) = \infty$ is trivial. Hence, according to Theorem 3.1, there exists an admissible control $\bar{u}(\cdot) \in \mathcal{U}(t,t_1,x)$ for $t_1 \in [t,T]$ such that $V^*(t,x) = t_1 - t$, and an associate trajectory $\bar{x}(t)$ such that $\bar{x}(t) = x$ and $\bar{x}(t_1) \in K$. We observe that $\frac{d}{dt}[V(t,\bar{x}(t))] = \partial_t V(t,\bar{x}(t)) + \nabla_x V(t,\bar{x}(t))^\top f(t,\bar{x}(t),\bar{u}(t)) \geq -1$ a.e. on $[t,t_1]$, the inequality holding since V satisfies Eq. (3.8). By integration, we observe that $V(t_1,\bar{x}(t_1)) - V(t,x) \geq t - t_1 = -V^*(t,x)$, *i.e.*, $V(t_1,\bar{x}(t_1)) + V^*(t,x) \geq V(t,x)$. As V satisfies Eq. (3.9) and as $\bar{x}(t_1) \in K$, we observe that $0 \geq V(t_1,\bar{x}(t_1))$, and therefore $V^*(t,x) \geq V(t,x)$. $\square$

3.1.3 Infinite dimensional LP formulations

In the rest of the chapter, we consider a given point $x_0 \in X$, and we raise the issue of computing the minimal time from x_0 to K, and the associated control. We make the following assumption:

Assumptions 3.2. *There exists an admissible control $u \in \mathcal{U}(0,t_1,x_0)$ associated with $t_1 \in [0,T]$. In other words, $V^*(0,x_0) \leq t_1 < \infty$.*

We consider the optimization problem of finding the subsolution of the HJB PDE that maximizes the evaluation in $(0,x_0)$. This problem may be cast as an infinite dimensional linear program:

$$\begin{array}{rl} \sup\limits_{V \in \mathcal{F}} & V(0,x_0) \\ \text{s.t.} & \\ \forall(t,x,u) \in [0,T] \times X \times U, & \partial_t V(t,x) + 1 + \nabla_x V(t,x)^\top f(t,x,u) \geq 0 \\ \forall(t,x) \in [0,T] \times K, & V(t,x) \leq 0, \end{array} \tag{$D_{\mathcal{F}}$}$$

with $\mathcal{F} \in \{C^1(\mathbb{R}^{n+1}), C^\infty(\mathbb{R}^{n+1}), \mathbb{R}[t,x_1,\ldots,x_n]\}$. For a given $V \in \mathcal{F}$, the feasibility in ($D_{\mathcal{F}}$) is clearly equivalent to the satisfaction of Eqs. (3.8)-(3.9). We also note that this infinite dimensional LP formulation corresponds to the dual LP formulation in [134]; in fact, this is the dual problem of an infinite dimensional LP formulation of the control problem based on occupation measures. According to the next theorem, the problem ($D_{\mathcal{F}}$) on C^1 functions has the same value as the minimal time control problem.

Theorem 3.2. *Under Assumption 3.1 and Assumption 3.2, and for $\mathcal{F} = C^1(\mathbb{R}^{n+1})$, the value of the LP formulation ($D_{\mathcal{F}}$) equals $V^*(0, x_0)$.*

Proof. As for the proof of Theorem 3.1, this is a direct application of [226, Th. 2.1], which also states the absence of duality gap between a control problem over a differential inclusion and a maximization problem over subsolutions of the HJB equation. We underline that the assumptions (H1)-(H5) in [226] are satisfied here; more precisely, our Assumption 3.1 enforces (H2) and our Assumption 3.2 enforces (H4). □

Theorem 3.3 extends this result by stating that we can require the subsolutions of the HJB equation to be in $C^\infty(\mathbb{R}^{n+1})$, while preserving the value of ($D_{\mathcal{F}}$). Before stating this theorem, we introduce an auxiliary lemma.

Lemma 3.2. *For any $V \in C^1(\mathbb{R}^{n+1})$ satisfying Eqs. (3.8)-(3.9) with feasibility error less or equal than $\eta \geq 0$, $V(t,x) + \eta(t - 1 - T)$ satisfies Eqs. (3.8)-(3.9).*

Proof. We introduce $\tilde{V}(t,x) = V(t,x) + \eta(t - 1 - T)$. By assumption on $V(t,x)$, we have $\partial_t V(t,x) + 1 + \nabla_x V(t,x)^\top f(t,x,u) \geq -\eta$, for all $(t,x,u) \in [0,T] \times X \times U$. By linearity, and since $\partial_t(t - 1 - T) = 1$ and $\nabla_x(t - 1 - T) = 0$, $\partial_t \tilde{V}(t,x) + 1 + \nabla_x \tilde{V}(t,x)^\top f(t,x,u) \geq 0$. By assumption on $V(t,x)$, we have $V(t,x) \leq \eta$, for all $(t,x) \in [0,T] \times K$. Hence, $\tilde{V}(t,x) \leq \eta + \eta(t - 1 - T) \leq \eta + \eta(T - 1 - T) = 0$ for all $(t,x) \in [0,T] \times K$. □

Theorem 3.3. *Under Assumption 3.1 and Assumption 3.2, and for $\mathcal{F} = C^\infty(\mathbb{R}^{n+1})$, the value of the LP formulation ($D_{\mathcal{F}}$) equals $V^*(0, x_0)$.*

Proof. We consider $\mathcal{F} = C^\infty(\mathbb{R}^{n+1})$ and we use the notation $\Sigma = [0,T] \times X$. We fix $\epsilon \in \mathbb{R}_{++}$, and we will prove that there exists $V \in C^\infty(\mathbb{R}^{n+1})$ that is feasible in ($D_{\mathcal{F}}$) and such that $V(0,x_0) \geq V^*(0,x_0) - \epsilon$. According to Theorem 3.2, there exists $V_1 \in C^1(\mathbb{R}^{n+1})$ that is feasible in ($D_{\mathcal{F}}$) and such that $V_1(0,x_0) \geq V^*(0,x_0) - \frac{\epsilon}{2}$. For any $\sigma \in (0,1]$, we introduce the mollified function $V_{1\sigma} = V_1 * \phi_\sigma \in C^\infty(\mathbb{R}^{n+1})$, where ω_σ is the standard mollifier defined as $\omega_\sigma(z) = \frac{1}{\sigma^{n+1}}\omega(z/\sigma)$, where $\omega(z) = \begin{cases} \xi e^{-\frac{1}{1-\|z\|^2}} & \text{if } \|z\| < 1 \\ 0 & \text{if } \|z\| \geq 1 \end{cases}$ for a given constant $\xi \in \mathbb{R}_{++}$ such that $\int_{\mathbb{R}^{n+1}} \omega(z)dz = 1$. Hence, a simple change of variable shows that $\int_{\mathbb{R}^{n+1}} \omega_\sigma(z)dz = 1$. We also underline that ω_σ is nonnegative and supported on the ball $B(0,\sigma)$. For any $z = (t,x) \in \Sigma$ and any $\sigma \in (0,1]$, we have that $|V_1(z) - V_{1\sigma}(z)| = |V_1(z) - \int_{B(0,\sigma)} V_1(z-h)\omega_\sigma(h)dh| = |\int_{B(0,\sigma)}(V_1(z) - V_1(z-h))\omega_\sigma(h)dh|$ as $\int_{B(0,\sigma)} \omega_\sigma(h)dh = 1$. We denote by L_V an upper bound for the continuous function $\|\nabla V_1(z)\|_2$ over the compact set $\Sigma_1 = \{z \in \mathbb{R}^{n+1} \colon d(z,\Sigma) \leq 1\}$, which is a Lipschitz constant for the function V_1. We deduce, by triangular inequality and non-negativity of ω_σ that for any $z \in \Sigma$,

$$\begin{aligned}
|V_1(z) - V_{1\sigma}(z)| &\leq \int_{B(0,\sigma)} |V_1(z) - V_1(z-h)|\omega_\sigma(h)dh && (3.10)\\
&\leq \int_{B(0,\sigma)} L_V \|h\| \omega_\sigma(h)dh && (3.11)\\
&\leq L_V \sigma \int_{B(0,\sigma)} \|h/\sigma\| \omega(h/\sigma) \frac{1}{\sigma^{n+1}} dh && (3.12)\\
&\leq L_V \sigma \underbrace{\int_{B(0,1)} \|\tilde{h}\| \omega(\tilde{h}) d\tilde{h}.}_{\text{constant, denoted } \mathcal{I}.} && (3.13)
\end{aligned}$$

By property of the mollifiers [107], we have $\partial_i V_{1\sigma}(z) = \partial_i (V_1 * \omega_\sigma) = (\partial_i V_1 * \omega_\sigma)$ for any $i \in \{t, x_1, \dots, x_n\}$. Therefore, $\partial_t V_{1\sigma}(z) = \int_{B(0,\sigma)} \partial_t V_1(z-h)\omega_\sigma(h)dh$ and $\nabla_x V_{1\sigma}(z) = \int_{B(0,\sigma)} \nabla_x V_1(z-h)\omega_\sigma(h)dh$. Using the equality $\int_{B(0,\sigma)} \omega_\sigma(h)dh = 1$, we deduce that for any $z \in \Sigma$,

$$\begin{aligned}
\partial_t V_{1\sigma}(z) + 1 + (\nabla_x V_{1\sigma}(z))^\top f(z,u) &= \int_{B(0,\sigma)} (\partial_t V_1(z-h) + 1 + (\nabla_x V_1(z-h))^\top f(z,u))\omega_\sigma(h)dh && (3.14)\\
&= \int_{B(0,\sigma)} (\partial_t V_1(z-h) + 1 + (\nabla_x V_1(z-h))^\top f(z-h,u))\omega_\sigma(h)dh && (3.15)\\
&\quad + \int_{B(0,\sigma)} \nabla_x V_1(z-h)^\top (f(z,u) - f(z-h,u))\omega_\sigma(h)dh. && (3.16)
\end{aligned}$$

We compute lower bounds for the two terms of the sum. We start with the second term: using Cauchy-Schwartz inequality, we notice that $\int_{B(0,\sigma)} \nabla_x V_1(z-h)^\top (f(z,u) - f(z-h,u))\omega_\sigma(h)dh \geq -\int_{B(0,\sigma)} \|V_1(z-h)\| \|f(z,u) - f(z-h,u)\| \omega_\sigma(h)dh$. Noticing that $\|\nabla_x V_1(z-h)\| \leq L_V$, since $z - h \in \Sigma_1$ for any $h \in B(0,\sigma) \subset B(0,1)$, and introducing the Lipschitz constant L_f for f, we have

$$\int_{B(0,\sigma)} \nabla_x V_1(z-h)^\top (f(z,u) - f(z-h,u))\omega_\sigma(h)dh \geq -L_V L_f \int_{B(0,\sigma)} \|h\| \omega_\sigma(h)dh = -L_V L_f \sigma \mathcal{I}. \tag{3.17}$$

We define $\eta = \frac{\epsilon}{2(T+2)}$. We introduce the compact set $S = [0,T] \times X \times U$ and the family of compact sets $S_\delta = \{s \in \mathbb{R}^{1+n+m} : d(s,S) \leq \delta\}$ for $\delta \in (0,1]$. For any $s = (z,u) \in S_1$, we introduce $\psi(s) = \partial_t V_1(z) + 1 + (\nabla_x V_1(z))^\top f(z,u)$. The function $\psi(s)$ is continuous and according to Lemma 0.7, there exists $\sigma_1 \in \mathbb{R}_{++}$ such that $\min_{s \in S_\sigma} \psi(s) \geq \min_{s \in S} \psi(s) - \frac{\eta}{2}$ for any $\sigma \in (0, \sigma_1]$. By feasibility of V_1 in ($D_\mathcal{F}$), we know that $\min_{s \in S} \psi(s) \geq 0$, which yields that $\psi(s) \geq -\frac{\eta}{2}$ for any $s \in S_{\sigma_1}$. We deduce that for any $\sigma \in (0, \sigma_1]$,

$$\int_{B(0,\sigma)} (\partial_t V_1(z-h) + 1 + (\nabla_x V_1(z-h))^\top f(z-h,u))\omega_\sigma(h)dh \geq -\int_{B(0,\sigma)} \frac{\eta}{2} \omega_\sigma(h)dh = -\frac{\eta}{2}, \tag{3.18}$$

since $(z-h,u) \in S_\sigma$ for any $h \in B(0,\sigma)$. Combining the decomposition of Eqs. (3.14)-(3.16), with the lower bounds of Eq. (3.17) and (3.18), we deduce that

$$\partial_t V_{1\sigma}(z) + 1 + (\nabla_x V_{1\sigma}(z))^\top f(z,u) \geq -(L_V L_f \sigma \mathcal{I} + \frac{\eta}{2}), \tag{3.19}$$

for any $(z,u) = (t,x,u) \in S$ and $\sigma \in (0,\sigma_1]$. We define $\tilde{\sigma} = \min\{\sigma_1, \frac{\eta}{2L_V L_f \mathcal{I}}, \frac{\eta}{L_V \mathcal{I}}\}$. From Eq. (3.13) and Eq. (3.19), we deduce that

$$V_{1\tilde{\sigma}}(0,x_0) \geq V_1(0,x_0) - \eta \geq V^*(0,x_0) - \frac{\epsilon}{2} - \eta \tag{3.20}$$

$$\forall (t,x) \in [0,T] \times K \quad V_{1\tilde{\sigma}}(t,x) \leq V_1(t,x) + \eta \leq \eta \tag{3.21}$$

$$\forall (t,x,u) \in S \quad \partial_t V_{1\tilde{\sigma}}(t,x) + 1 + \nabla_x V_{1\tilde{\sigma}}(t,x)^\top f(t,x,u) \geq -\eta. \tag{3.22}$$

From Lemma 3.2, we deduce that $V(t,x) = V_{1\tilde{\sigma}}(t,x) + \eta(t-1-T) \in C^\infty(\mathbb{R}^{n+1})$ is feasible in $(D_\mathcal{F})$. From Eq. (3.20), we deduce that $V(0,x_0) \geq V^*(0,x_0) - \frac{\epsilon}{2} - \eta - (1+T)\eta$, and by definition of η, $V(0,x_0) \geq V^*(0,x_0) - \epsilon$. □

The next theorem underlies the convergence proof of the hierarchy of semi-infinite problems in Section 3.2: if we restrict to polynomials HJB subsolutions, the value of the problem $(D_\mathcal{F})$ remains unchanged.

Theorem 3.4. *Under Assumption 3.1 and Assumption 3.2, and for $\mathcal{F} = \mathbb{R}[t,x_1,\dots,x_n]$, the value of the LP formulation $(D_\mathcal{F})$ equals $V^*(0,x_0)$.*

Proof. We consider $\mathcal{F} = \mathbb{R}[t,x_1,\dots,x_n]$. For a given $\epsilon \in \mathbb{R}_{++}$, and we will prove that there exists $V \in \mathbb{R}[t,x_1,\dots,x_n]$ that is feasible in $(D_\mathcal{F})$ and such that $V(0,x_0) \geq V^*(0,x_0) - \epsilon$. According to Theorem 3.3, there exists a function $Q \in C^\infty(\mathbb{R}^{n+1})$ which is a subsolution of the HJB equation and such that $Q(0,x_0) \geq V^*(0,x_0) - \frac{\epsilon}{2}$. We notice that Q has a locally Lipschitz gradient. Therefore, we can apply Lemma 0.8. This yields, in particular, that for any $\nu \in \mathbb{R}_{++}$, there exists a polynomial $w \in \mathbb{R}[t,x_1,\dots,x_n]$ such that for all $(t,x) \in [0,T] \times X$, $|w(z) - w(z)| \leq \nu$ and $|\partial_i w(t,x) - \partial_i V(t,x)| \leq \nu$, $i \in \{t,x_1,\dots,x_N\}$. We deduce that $|\partial_t Q(t,x) + \nabla_x Q(t,x)^\top f(t,x,u) - \partial_t w(t,x) + \nabla_x w(t,x)^\top f(t,x,u)| \leq |\partial_t Q(t,x) - \partial_t w(t,x)| + \sum_{i=1}^n |\partial_{x_i} Q(t,x) - \partial_{x_i} w(t,x)| M_i \leq \nu(1 + \sum_{i=1}^n M_i)$, where $M_i = \max_{(t,x,u)\in S} |f_i(t,x,u)|$. Therefore, for all $(t,x,u) \in S$,

$$\partial_t w(t,x) + 1 + \nabla_x w(t,x)^\top f(t,x,u) \geq \partial_t Q(t,x) + 1 + \nabla_x Q(t,x)^\top f(t,x,u) - \nu(1 + \sum_{i=1}^n M_i) \tag{3.23}$$

$$\geq -\nu(1 + \sum_{i=1}^n M_i), \tag{3.24}$$

as Q is a subsolution of the HJB equation. In summary, for $\nu = \eta(1 + \sum_{i=1}^n M_i)^{-1} \leq \eta$,

$$w(0,x_0) \geq Q(0,x_0) - \eta \geq V^*(0,x_0) - \frac{\epsilon}{2} - \eta \tag{3.25}$$

$$\forall (t,x) \in [0,T] \times K \quad w(t,x) \leq Q(t,x) + \eta \leq \eta \tag{3.26}$$

$$\forall (t,x,u) \in S \quad \partial_t w(t,x) + 1 + \nabla_x w(t,x)^\top f(t,x,u) \geq -\eta, \tag{3.27}$$

the last inequality following from Eq. (3.24). Based on Eqs. (3.26)-(3.27) and Lemma 3.2, we notice that the polynomial $V(t,x) = w(t,x) + \eta(t - T - 1) \in \mathbb{R}[t, x_1, \dots, x_n]$ is feasible in the problem ($D_\mathcal{F}$). Having defined $\eta = \frac{\epsilon}{2(T+2)}$, we see, based on Eq. (3.25), that it satisfies $V(0,x_0) \geq V^*(0,x_0) - \frac{\epsilon}{2} - \eta - \eta(T+1) = V^*(0,x_0) - \epsilon$. $\square$

3.2 Convex semi-infinite programming to compute near-optimal HJB subsolutions

For $\mathcal{F}$ being either $C^1(\mathbb{R}^{n+1})$, $C^\infty(\mathbb{R}^{n+1})$, or $\mathbb{R}[t, x_1, \dots, x_n]\}$, the linear program ($D_\mathcal{F}$) is infinite dimensional, and thus, not tractable as it stands. We next present a hierarchy of convex SIP problems that are solvable with a dedicated algorithm, to compute subsolutions to the HJB equation that are near-optimal in the problem ($D_\mathcal{F}$).

3.2.1 A hierarchy of linear semi-infinite programs

Instead of having an optimization space $\mathcal{F}$ that is infinite dimensional, we suggest to restrict to the finite dimensional subspaces $\mathbb{R}_d[t, x_1, \dots, x_n]$ of polynomials of degree bounded by d. This restricted dual problem is:

$$\begin{aligned} \sup_{V \in \mathbb{R}_d[t,x_1,\dots,x_n]} \quad & V(0,x_0) \\ \text{s.t.} \quad & \\ \forall (t,x,u) \in [0,T] \times X \times U, \quad & \partial_t V(t,x) + 1 + \nabla_x V(t,x)^\top f(t,x,u) \geq 0 \\ \forall (t,x) \in [0,T] \times K, \quad & V(t,x) \leq 0. \end{aligned} \tag{$\mathbf{R_d}$}$$

In the rest of the chapter, we will denote by N the dimension of the vector space $\mathbb{R}_d[t, x_1, \dots, x_n]$ and $\Phi(t,x) \in \mathbb{R}^N$ a basis of this space. For both objects, there is indeed a dependence of d, that is implicit here for readability reasons. For any $V \in \mathbb{R}_d[t, x_1, \dots, x_n]$, we introduce the vector θ of the coordinates of V in the basis Φ. Hence, we have the relation

$$V(t,x) = \theta^\top \Phi(t,x) \in \mathbb{R}_d[t, x_1, \dots, x_n]. \tag{3.28}$$

Expressing problem ($\mathbf{R_d}$) as an optimization problem over the vector of coefficients, it appears clearly that this is a linear semi-infinite program.

Proposition 3.1. *For $d \in \mathbb{N}^*$, problem ($\mathbf{R_d}$) is a linear semi-infinite program, i.e, a linear program with a finite number of variables and an infinite number of constraints. More precisely, there exist a vector $c \in \mathbb{R}^N$, and a compact set $\mathcal{Y} \subset \mathbb{R}^{N+1}$ such that ($\mathbf{R_d}$) reads*

$$\begin{aligned} \sup_{\theta \in \mathbb{R}^N} \quad & c^\top \theta \\ s.t. \quad & \forall (a,b) \in \mathcal{Y},\ a^\top \theta + b \leq 0. \end{aligned} \tag{SIP}$$

Proof. We define the vector $c = \Phi(0, x_0)$, and the compact sets

$$\mathcal{Y}_1 = \{(-\partial_t \Phi(t,x) - \nabla_x \Phi(t,x)^\top f(t,x,u), -1), (t,x,u) \in [0,T] \times X \times U\} \tag{3.29}$$

$$\mathcal{Y}_2 = \{(\Phi(t,x), 0), (t,x) \in [0,T] \times K\} \tag{3.30}$$

$$\mathcal{Y} = \mathcal{Y}_1 \cup \mathcal{Y}_2. \tag{3.31}$$

We see that for any $V_\theta(t,x) = \theta^\top \Phi(t,x) \in \mathbb{R}_d[t, x_1, \dots, x_n]$, $V_\theta(0,x_0) = c^\top \theta$, and $V_\theta(t,x)$ is feasible in ($\mathbf{R_d}$) if and only if $a^\top \theta + b \leq 0$, for all $(a,b) \in \mathcal{Y}$. $\square$

We will see in the next section how to efficiently solve those semi-infinite programs. Prior to that, we state the convergence of this hierarchy of semi-infinite programs.

Theorem 3.5. *The sequence* $\mathsf{val}(\mathbf{R_d})$ *converges to* $V^*(0,x_0)$ *when* $d \to \infty$.

Proof. On the one hand, we introduce the notation $v_d = \mathsf{val}(\mathbf{R_d})$. This sequence is obviously an increasing sequence, bounded above by $V^*(0,x_0)$. Hence, it converges to a value $\ell \leq V^*(0,x_0)$, and any subsequence converges to ℓ. On the other hand, Theorem 3.4 guarantees that there exists a sequence of polynomials $w_k \in \mathbb{R}[t, x_1, \dots, x_n]$ feasible in ($D_\mathcal{F}$) and such that $w_k(0,x_0) \to_k V^*(0,x_0)$. By definition, we have $v_{d_k} \geq w_k(0,x_0)$, where $d_k = \mathsf{deg}(w_k)$. Up to the extraction of a subsequence of (w_k), we can assume that the sequence d_k increasing, therefore $(v_{d_k})_{k \in \mathbb{N}}$ is a subsequence of $(v_d)_{d \in \mathbb{N}}$. As $v_{d_k} \to \ell$ and $w_k(0,x_0) \to_k V^*(0,x_0)$, we deduce that $\ell \geq V^*(0,x_0)$, which yields the equality $\ell = V^*(0,x_0)$. $\square$

3.2.2 Regularization and solution of the semi-infinite programs

We introduce a quadratic regularization in the semi-infinite program (**SIP**), yielding the following formulation depending on $\mu \in \mathbb{R}_{++}$:

$$\begin{aligned} \max_{\theta \in \mathbb{R}^N} \quad & c^\top \theta - \tfrac{\mu}{2}\|\theta\|^2 \\ \text{s.t.} \quad & \forall (a,b) \in \mathcal{Y},\ a^\top \theta + b \leq 0. \end{aligned} \tag{$\mathbf{SIP}_\mu$}$$

Proposition 3.2. *For any* $\mu \in \mathbb{R}_{++}$, *the semi-infinite program* ($\mathbf{SIP}_\mu$) *has a unique optimal solution with value* $\mathsf{val}(\mathbf{SIP}_\mu) \leq V^*(0,x_0)$. *Moreover,* $\mathsf{val}(\mathbf{SIP}_\mu) \underset{\mu \to 0}{\to} \mathsf{val}(\mathbf{SIP})$.

Proof. The feasible set of ($\mathbf{SIP}_\mu$) being convex, and the objective function being strongly concave, this optimization problem admits a unique maximum θ. By definition, $\mathsf{val}(\mathbf{SIP}_\mu) = c^\top \theta - \frac{\mu}{2}\|\theta\|^2 \leq c^\top \theta \leq \mathsf{val}(\mathbf{SIP})$, since θ is also feasible in the maximization problem (**SIP**). Additionally, $\mathsf{val}(\mathbf{SIP}) \leq V^*(0,x_0)$, since any function V feasible in ($\mathbf{R_d}$) satisfies $V(0,x_0) \leq V^*(0,x_0)$. We also notice that the function $\mu \mapsto \mathsf{val}(\mathbf{SIP}_\mu)$ is decreasing, so it admits a limit ℓ at 0^+, due to the aforementioned inequalities, $\ell \leq \mathsf{val}(\mathbf{SIP})$. For any $\mu, \epsilon \in \mathbb{R}_{++}$, if we take θ_ϵ an ϵ-optimal solution in the problem (**SIP**), we see that $\mathsf{val}(\mathbf{SIP}) - \epsilon - \frac{\mu}{2}\|\theta_\epsilon\|^2 \leq c^\top \theta_\epsilon - \frac{\mu}{2}\|\theta_\epsilon\|^2 \leq \mathsf{val}(\mathbf{SIP}_\mu)$. For a fixed ϵ, and taking $\mu \to 0^+$, we obtain $\mathsf{val}(\mathbf{SIP}) - \epsilon \leq \ell$. This being true for any $\epsilon \in \mathbb{R}_{++}$, we deduce that $\mathsf{val}(\mathbf{SIP}) \leq \ell$, which proves the equality. $\square$

Setting the regularization parameter μ in practice implies a trade-off between the computational tractability of the semi-infinite program (**SIP**$_\mu$) and the accuracy of the approximation of the original problem (**SIP**). To solve the formulation (**SIP**$_\mu$), we use the algorithm CPA (Algorithm 2). To that extent, we need a *separation oracle* computing, for any $\theta \in \mathbb{R}^N$,

$$\phi(\theta) = \max_{(a,b)\in\mathcal{Y}} a^\top \theta + b, \tag{3.32}$$

and an associate argmaximum. Solving the optimization problem in Eq. (3.32) may be computationally intensive, since the compact set $\mathcal{Y}$ may not be convex. Therefore, we only assume to have a δ-oracle, as defined in the Introduction: an algorithm computing, for any θ, $(a, b) \in \mathcal{Y}$, such that $\phi(\theta) - (a^\top\theta + b) \leq \delta|\phi(\theta)|$. We treat this oracle as a black box, regardless of its implementation, via global optimization, gridding, interval arithmetics or sampling for instance. The instantiation of Algorithm 2 for the problem (**SIP**$_\mu$) is Algorithm 4.

Algorithm 4 Cutting-plane algorithm for (**SIP**$_\mu$)

```
    Input: An oracle with parameter $\delta \in [0, 1)$, a tolerance $\epsilon \in \mathbb{R}_+$, a finite set $\mathcal{Y}^0 \subset \mathcal{Y}$, $k \leftarrow 0$
1:  $\nu_0 \leftarrow \infty$
0:  while $\nu_k > \epsilon$ do
1:      Compute $\theta^k$ the optimal solution of the convex Quadratic Programming (QP) problem
            $\max_{\theta\in\mathbb{R}^N} \; c^\top\theta - \frac{\mu}{2}\|\theta\|^2$
            s.t. $\forall (a, b) \in \mathcal{Y}^k,\; a^\top\theta + b \leq 0.$          (3.33)
2:      Call the $\delta$-oracle to compute $(a^k, b^k)$ an approximate solution of (3.32).
3:      $\nu_k \leftarrow (a^k)^\top\theta^k + b^k$.
4:      $\mathcal{Y}^{k+1} \leftarrow \mathcal{Y}^k \cup \{(a^k, b^k)\}$.
5:      $k \leftarrow k + 1$.
6:  end while
7:  Return $\theta^k$.
```

Before stating the termination and the convergence of Algorithm 4, we introduce the vector $\hat{\theta} \in \mathbb{R}^N$ of coordinates of the polynomial $\hat{v}(t, x) = t - 1 - T$ in the basis $\Phi(t, x)$. Since $\phi(\hat{\theta}) = -1$, this helps obtaining feasible solutions: due to Lemma 3.2, we observe that if θ has a feasibility error less or equal than $\eta \geq 0$ in (**SIP**) and (**SIP**$_\mu$) then, $\theta + \eta\hat{\theta}$ is feasible in (**SIP**) and (**SIP**$_\mu$). For any $\mu \in \mathbb{R}_{++}$, we define the convex and compact set $\mathcal{X}_\mu = \{\theta \in \mathbb{R}^N : c^\top\theta - \frac{\mu}{2}\|\theta\|^2 \geq c^\top\hat{\theta} - \frac{\mu}{2}\|\hat{\theta}\|^2\}$, and we define $R_\mu = \sup_{\theta\in\mathcal{X}_\mu}\|\theta\|$. Finally, we define the function $r_\mu(e) = e(1 + T + \mu R_\mu^2(1 + \frac{e}{2}))$. Note that $r_\mu(e) \underset{e\to 0}{\to} 0$.

Theorem 3.6. *If $\epsilon \in \mathbb{R}_{++}$, Algorithm 4 stops after a finite number K of iterations, and $\theta^K + \frac{\epsilon}{1-\delta}\hat{\theta}$ is a feasible and $r_\mu(\frac{\epsilon}{1-\delta})$-optimal in (**SIP**$_\mu$). If $\epsilon = 0$, the alternative holds: (a) Algorithm 4 either stops after a finite number of iterations, and the last iterate is the optimal solution of (**SIP**$_\mu$), (b) Or it generates an infinite sequence, and the optimality gap and the feasibility error converge towards zero with an asymptotic rate in $O(\frac{1}{k})$.*

Proof. First of all, we notice that during the execution of Algorithm 4, we necessarily have $\theta^k \in \mathcal{X}_\mu$, since $\hat{\theta}$ is a feasible solution in (3.33) with value $c^\top \hat{\theta} - \frac{\mu}{2}\|\hat{\theta}\|^2$, therefore by optimality of θ^k in (3.33), $c^\top \theta^k - \frac{\mu}{2}\|\theta^k\|^2 \geq c^\top \hat{\theta} - \frac{\mu}{2}\|\hat{\theta}\|^2$. We recall that Algorithm 4 is in fact Algorithm 2 applied to the problem for (**SIP**$_\mu$). The finite convergence if $\epsilon \in \mathbb{R}_{++}$, and the convergence rate in the case $\epsilon = 0$ (if no finite convergence) follows from Theorems 1.1-1.2 in Chapter 1: we apply these theorems to the problem (**SIP**$_\mu$) with the additional constraint set $\mathcal{X} = \mathcal{X}_\mu$. As previously explained, this additional constraint does not change the execution of the algorithm, but it enables us to match the setting of Theorems 1.1-1.2, with a compact constraint set $\mathcal{X}$. We also note that the semi-infinite constraint is indeed linear in θ, the objective function is μ-strongly concave, and that $\hat{\theta} \in \mathcal{X}_\mu$ is a strictly feasible point with respect to the semi-infinite constraints: Assumptions 1.1-1.3 are indeed satisfied.

We finish the proof by showing that if Algorithm 4 stops at iteration K, then $\tilde{\theta} = \theta^K + \frac{\epsilon}{1-\delta}\hat{\theta}$ is feasible and $r_\mu(\frac{\epsilon}{1-\delta})$-optimal in (**SIP**$_\mu$). If Algorithm 4 stops at iteration K, this means that $(a^K)^\top \theta^K + b^K \leq \epsilon$. If $\phi(\theta^K) \leq 0$, then θ^K is feasible in (**SIP**$_\mu$), and so is $\tilde{\theta}$ due to Lemma 3.2. If $\phi(\theta^K) > 0$, then by property of the δ-oracle, $(1-\delta)\phi(\theta^K) \leq (a^K)^\top \theta^K + b^K \leq \epsilon$, and we deduce that the feasibility error is $\phi(\theta^K) \leq \frac{\epsilon}{1-\delta}$. With Lemma 3.2, we deduce that $\tilde{\theta}$ is feasible in (**SIP**$_\mu$). We also note that

$$c^\top \tilde{\theta} - \frac{\mu}{2}\|\tilde{\theta}\|^2 = c^\top \theta^K + \frac{\epsilon}{1-\delta} c^\top \hat{\theta} - \frac{\mu}{2}\|\theta^K + \frac{\epsilon}{1-\delta}\hat{\theta}\|^2 \tag{3.34}$$

$$\geq c^\top \theta^K - \frac{\epsilon}{1-\delta}(1+T) - \frac{\mu}{2}\left(\|\theta^K\|^2 + \frac{2\epsilon}{1-\delta}\|\theta^K\|\,\|\hat{\theta}\| + \frac{\epsilon^2}{(1-\delta)^2}\|\hat{\theta}\|^2\right), \tag{3.35}$$

since $c^\top \hat{\theta} = V_{\hat{\theta}}(0, x_0) = -(1+T)$, and due to the Cauchy-Schwartz inequality. By optimality of θ^K in (3.33), which is a relaxation of (**SIP**$_\mu$), we know that $\mathsf{val}(\textbf{SIP}_\mu) \leq c^\top \theta^K - \frac{\mu}{2}\|\theta^K\|^2$. We deduce from Eq. (3.35) that

$$c^\top \tilde{\theta} - \frac{\mu}{2}\|\tilde{\theta}\|^2 \geq \mathsf{val}(\textbf{SIP}_\mu) - \frac{\epsilon}{1-\delta}(1+T) - \frac{\mu}{2}\left(\frac{2\epsilon}{1-\delta}\|\theta^K\|\,\|\hat{\theta}\| + \frac{\epsilon^2}{(1-\delta)^2}\|\hat{\theta}\|^2\right) \tag{3.36}$$

$$\geq \mathsf{val}(\textbf{SIP}_\mu) - \frac{\epsilon}{1-\delta}(1+T) - \mu R_\mu^2\left(\frac{\epsilon}{1-\delta} + \frac{\epsilon^2}{2(1-\delta)^2}\right) \tag{3.37}$$

$$\geq \mathsf{val}(\textbf{SIP}_\mu) - r_\mu(\frac{\epsilon}{1-\delta}), \tag{3.38}$$

the second inequality following from the fact that $\|\hat{\theta}\| \leq R_\mu$ and $\|\theta^K\| \leq R_\mu$, as $\hat{\theta}, \theta^K \in \mathcal{X}_\mu$. ☐

3.3 Feedback control based on approximate value functions

In the previous section, we have seen how to compute subsolutions of the HJB equation based on convex semi-infinite programming, and how to deduce a lower bound on the minimal travel time. In this section, we focus on how subsolutions of the HJB equation, which approximate the value function V^*, enable one to recover a near-optimal control for the minimal time control problem (3.2)-(3.5).

3.3.1 Controller design and existence of trajectories

For a given continuously differentiable function $V \in C^1(\mathbb{R}^{n+1})$, we define the set-valued maps

$$\mathcal{U}_V(t,x) = \operatorname*{argmin}_{u \in U} \nabla_x V(t,x)^\top f(t,x,u) \tag{3.39}$$

$$\mathcal{I}_V(t,x) = \{u \in \mathcal{U}_V(t,x) \colon f(t,x,u) \in T_X(x)\}, \tag{3.40}$$

where $T_X(x)$ is the contingent cone to X at point x (see Introduction). In line with previous works designing feedback controllers based on approximate value functions [101, 115], we are interested in the trajectories satisfying the following differential inclusion depending on the function $V \in C^1(\mathbb{R}^{n+1})$:

$$\dot{x}_V(t) = f(t, x_V(t), u_V(t)) \text{ with } u_V(t) \in \mathcal{U}_V(t, x_V(t)). \tag{CL_V}$$

Intuitively, such a feedback control pushes the system towards the descent direction of the function V. The following proposition confirms that, should the function $V \in C^1(\mathbb{R}^{n+1})$ be optimal in problem ($D_{\mathcal{F}}$), then any minimal time trajectory satisfies the differential inclusion (CL_V) with respect to V.

Proposition 3.3. *Under Assumptions 3.1-3.2, we consider an optimal trajectory $(x^*(\cdot), u^*(\cdot))$ of the minimal time control problem (3.2)-(3.5) starting from $(0, x_0)$, with hitting time $\tau^* = V^*(0, x_0)$. If the linear program ($D_{\mathcal{F}}$), for $\mathcal{F} = C^1(\mathbb{R}^{n+1})$, admits an optimal solution V, then, for almost every $t \in [0, \tau^*]$,*

$$u^*(t) \in \mathcal{I}_V(t, x^*(t)) \subset \mathcal{U}_V(t, x^*(t)). \tag{3.41}$$

In particular, the trajectory $(x^*(\cdot), u^*(\cdot))$ satisfies the differential inclusion (CL_V).

Proof. We define the function $\alpha(t) = V(t, x^*(t)) + t$, which is differentiable. We have that $\alpha'(t) = \partial_t V(t, x^*(t)) + 1 + \nabla_x V(t, x^*(t))^\top f(t, x^*(t), u^*(t))$, for almost all $t \in [0, \tau^*]$. Since V is feasible in ($D_{\mathcal{F}}$), therefore satisfies Eq. (3.8), and since $(x^*(t), u^*(t)) \in X \times U$ a. e. on $[0, \tau^*]$, we know that $\alpha'(t) \geq 0$ a. e. on $[0, \tau^*]$. This proves that the differentiable function $\alpha(t)$ is non-decreasing function over $[0, \tau^*]$. By optimality of V in ($D_{\mathcal{F}}$), and due to Theorem 3.2 (Assumptions 3.1-3.2 are satisfied), $\alpha(0) = V(0, x_0) = \mathsf{val}(D_{\mathcal{F}}) = \tau^*$. Moreover, $\alpha(\tau^*) = \tau^* + V(\tau^*, x^*(\tau^*)) = \tau^*$, since V satisfies Eq. (3.9) and $x^*(\tau^*) \in K$. From $\alpha(\tau^*) \leq \alpha(0)$, we obtain that $\alpha(t)$ is constant. Hence, $\partial_t V(t, x^*(t)) + 1 + \nabla_x V(t, x^*(t))^\top f(t, x^*(t), u^*(t)) = 0$, meaning

$$\nabla_x V(t, x^*(t))^\top f(t, x^*(t), u^*(t)) = -(\partial_t V(t, x^*(t)) + 1), \text{ a.e. on } [0, \tau^*]. \tag{3.42}$$

As V satisfies Eq. (3.8), we have that $\nabla_x V(t, x^*(t))^\top f(t, x^*(t), u) \geq -(\partial_t V(t, x^*(t)) + 1)$ for all $t \in [0, \tau^*]$ and for all $u \in U$. Together with Eq. (3.42), we deduce that $u^*(t) \in \mathcal{U}_V(t, x^*(t))$ for almost all $t \in [0, \tau^*]$. Based on this fact, Lemma 0.9 yields that for almost all $t \in [0, \tau^*]$, $f(t, x^*(t), u^*(t)) \in T_X(x^*(t))$. Therefore, for almost all $t \in [0, \tau^*]$, $u^*(t) \in \mathcal{I}_V(t, x^*(t))$. $\square$

We just saw that whenever $V \in C^1(\mathbb{R}^{n+1})$ is optimal in the linear program $(D_{\mathcal{F}})$, any minimal time trajectory is a solution of the differential inclusion (CL_V) associated with the function V. However, we may not be able to compute exactly such an optimal function in practice, especially because it may not exist. The next theorem states the existence of closed-loop trajectories following (CL_V), for any function $V \in C^1(\mathbb{R}^{n+1})$.

Theorem 3.7. *Under Assumptions 3.1-3.2, if $V \in C^1(\mathbb{R}^{n+1})$ is such that for any $(t,x) \in \mathbb{R}_+ \times X, \mathcal{I}_V(t,x) \neq \emptyset$, then there exists a trajectory $(x_V(\cdot), u_V(\cdot))$ starting at $(0,x_0)$, satisfying the differential inclusion (CL_V) over $[0,\infty)$ and such that $x_V(t) \in X$ for almost all $t \in [0,\infty)$.*

Proof. We introduce an auxiliary control system to reduce to a time-invariant system with a convex control set, so as to fit in the setting of [7, Th. 6.6.6]. In what follows, we use the notation $z = (t,x)$ again. We introduce two objects: the set-valued map $\hat{U}(z) = \{f(z,u), u \in U\}$ and the function $\hat{f}(z,v) = \begin{pmatrix} 1 \\ v \end{pmatrix}$ for $z \in \mathbb{R}^{n+1}$ and $v \in \mathbb{R}^n$. According to the terminology introduced in [7, Def. 6.1.3], $(\hat{U},\hat{f})$ is a Marchaud control system, as (i) $\{(z,v) \in \mathbb{R}^{2n+1} : v \in \hat{U}(z)\}$ is closed, (ii) $\hat{f}$ is continuous, (iii) the velocity set $\{1\} \times f(z,U)$ is convex according to Assumption 3.1 and (iv) $\hat{f}$ has a linear growth, and so has $\hat{U}$ due to the fact that f is Lipschitz continuous and $\mathcal{U}$ is bounded. We introduce $\mathcal{C} = \mathbb{R}_+ \times X$ and define the regulation map $REG(z) = \{v \in \hat{U}(z) \colon \hat{f}(z,v) \in T_{\mathcal{C}}(z)\}$. We also introduce the set-valued map $SEL(z) = \underset{v \in \hat{U}(z)}{\operatorname{argmin}} \nabla_x V(z)^\top v$. We prove now that the graph of SEL is closed. For any converging sequence $(z_k, v_k) \to (\bar{z}, \bar{v})$ with $v_k \in SEL(z_k)$, we see that for all $k \in \mathbb{N}$, $v_k = f(z_k, u_k)$ for a given $u_k \in U$ and $\nabla_x V(z_k)^\top f(z_k,u_k) = h(z_k)$, where $h(z_k) = \min_{u \in U} \nabla_x V(z_k)^\top f(z_k,u)$. Up to extracting a subsequence of u_k, we can assume that $u_k \to \bar{u}$, as U is compact. Note that h is continuous, by application of the Maximum Theorem [7, Th. 2.1.6], in so far as (i) $(z,u) \mapsto \nabla_x V(z)^\top f(z,u)$ is continuous, therefore lower and upper semicontinuous, (ii) the set-valued map $M(z) = U$ is compact-valued, and lower and upper semicontinuous since it is constant. By continuity of h, ∇V and f, we conclude that $\nabla_x V(\bar{z})^\top \bar{v} = \nabla_x V(\bar{z})^\top f(\bar{z},\bar{u}) = h(\bar{z}) = \min_{v \in \hat{U}(\bar{z})} \nabla_x V(\bar{z})^\top v$, meaning that $\bar{v} = f(\bar{z},\bar{u}) \in SEL(\bar{z})$.

We notice that if $u \in \mathcal{I}_V(z)$, then $v = f(z,u) \in REG(z) \cap SEL(z)$. As $\mathcal{I}_V(z) \neq \emptyset$ for all $z \in \mathcal{C}$ (by assumption), $REG(z) \cap SEL(z) \neq \emptyset$. Together with the closedness of the graph of SEL, this means SEL is a selection procedure of REG, according to the terminology of [7, Def. 6.5.2], and has convex values. We underline that $REG(z) \neq \emptyset$, for all $z \in \mathcal{C}$, *i.e.*, $\mathcal{C}$ is a viability domain for $(\hat{U},\hat{f})$. As $(0,x_0) \in \mathcal{C}$, [7, Th. 6.6.6] yields the existence of a solution $(z(\cdot), v(\cdot))$ such that $z(t) \in \mathcal{C}$, $v(t) \in R_{\mathcal{C}}(z(t))$ and

$$v(t) \in SEL(z(t)) \cap REG(z(t)), \tag{3.43}$$

for almost all $t \in [0,\infty)$. We notice first that $z_1(0) = 0$ and $\dot{z}_1(t) = 1$ for almost all $t \geq 0$, thus $z_1(t) = t$. Hence, we can indeed see $z(t)$ as $(t, x(t))$, with $x(0) = x_0$ and $\dot{x}(t) = v(t)$. Moreover, $v(t) = f(t,x(t),u(t))$ for a given $u(t) \in U$, since $v(t) \in \hat{U}(z(t)) = f(t,x(t),U)$ a.e. on $[0,\infty)$.

We deduce from $v(t) \in SEL(z(t))$, which comes from (3.43), that $u(t) \in \mathcal{U}_V(t, x(t))$. Moreover, we deduce from $z(t) \in \mathcal{C}$ that $x(t) \in X$ a.e. on $[0, \infty)$. $\square$

Remark 3.2. *The condition $\mathcal{I}_V(t, x) \neq \emptyset$ in Theorem 3.7 may appear restrictive, because it is not evident why a vector $f(t, x, u_V)$ minimizing $\nabla_x V(t, x)^\top f(t, x, u)$ over $u \in U$ would belong to $T_X(x)$. However, we have seen that under the hypotheses of Proposition 3.3, Eq. (3.41) yields $\mathcal{I}_V(t, x) \neq \emptyset$. Moreover, should the condition $\mathcal{I}_V(t, x) \neq \emptyset$ not be satisfied, we could enlarge the definition of $\mathcal{U}_V(t, x)$ in $\mathcal{U}_{V,\epsilon}(t, x) = \operatorname{argmin}_{\epsilon, u \in U} \nabla_x V(t, x)^\top f(t, x, u)$, so that for $\epsilon \in \mathbb{R}_{++}$ large enough, $\mathcal{I}_{V,\epsilon}(t, x) = \{u \in \mathcal{U}_{V,\epsilon}(t, x) \colon f(t, x, u) \in T_X(x)\} \neq \emptyset$.*

3.3.2 Performance of the feedback controller depending on the value function approximation error

Previously, we studied closed-loop trajectories satisfying the differential inclusion (CL_V) with respect to a function $V \in C^1(\mathbb{R}^{n+1})$. We state now some performance guarantees on those trajectories, depending on some approximation properties of V. In the following, we assume that, up to an enlargement of the time horizon, the system can reach the target set starting from any initial condition $(t, x) \in [0, T] \times X$, and that the associated value function is Lipschitz.

Assumptions 3.3. *There exists a time $T^\sharp \geq T$ such that the minimal time control problem (3.2)-(3.5) defined over $[0, T^\sharp]$ has a value function $V^\sharp$ which takes finite values over $\Sigma = [0, T] \times X$, and is Lipschitz continuous.*

We emphasize that, under Assumption 3.3, $V^*(t, x) < \infty$ implies $V^*(t, x) = V^\sharp(t, x)$ for any $(t, x) \in \Sigma$. Since $V^\sharp$ is Lipschitz continuous over $\Sigma \subset \mathbb{R}^{n+1}$, it admits a Lipschitz continuous extension over $\mathbb{R}^{n+1}$ [79, Chap. 3, Th. 1]. We assimilate the value function and its extension on $\mathbb{R}^{n+1}$, such that we can speak about the Clarke's generalized derivative $\partial^c V^\sharp(z)$ of $V^\sharp$ at $z \in \Sigma$. For any $V \in C^1(\mathbb{R}^{n+1})$, we introduce the notation

$$\|\nabla V - \nabla V^\sharp\|_\infty = \sup_{z \in \Sigma} \sup_{g \in \partial^c V^\sharp(z)} \|\nabla V(z) - g\|_2. \tag{3.44}$$

We also define the constant $C_f = \sup_{(t,x,u) \in \Sigma \times U} \|f(t, x, u)\| < \infty$.

Theorem 3.8. *Let $V \in C^1(\mathbb{R}^{n+1})$ be a continuously differentiable function, and let $(x_V(\cdot), u_V(\cdot))$ be a closed-loop trajectory starting at $(0, x_0)$ satisfying the differential inclusion (CL_V) and the state constraints over $[0, T]$. We define $t_V = \sup\{t \in [0, T] \colon x_V([0, t]) \subset X \setminus K\}$. Then, under Assumptions 3.1-3.3,*

$$V^\sharp(t, x_V(t)) \leq (\tau^* - t) + t\, 2(1 + C_f)\|\nabla V - \nabla V^\sharp\|_\infty \quad \forall t \in [0, t_V], \tag{3.45}$$

where $\tau^* = V^*(0, x_0) = V^\sharp(0, x_0) \leq t_V$. In particular, we notice that

$$V^\sharp(\tau^*, x_V(\tau^*)) \leq 2\tau^*(1 + C_f)\|\nabla V - \nabla V^\sharp\|_\infty. \tag{3.46}$$

In Eq. (3.46), $V^\sharp(\tau^*, x_V(\tau^*))$ measures how far the closed-loop trajectory $(x_V(\cdot), u_V(\cdot))$ is from the target set K at the time when the optimal trajectory reaches K. As a corollary, we give a condition for the closed-loop trajectory $(x_V(\cdot), u_V(\cdot))$ to effectively reach the target set K, with a bounded delay compared to the time-optimal trajectory.

Corollary 3.1. *Under the same hypotheses as Theorem 3.8, if* $\|\nabla V - \nabla V^\sharp\|_\infty \leq \frac{1-\tau^*/T}{2(1+C_f)}$, *then*

$$x_V(t_V) \in K \text{ with } t_V \in [\tau^*, \frac{1}{1 - 2(1+C_f)\|\nabla V - \nabla V^\sharp\|_\infty}\tau^*]. \tag{3.47}$$

We underline that the hitting time $t_V \geq \tau^*$ converges to the minimal time τ^*, when the approximation error $\|\nabla V - \nabla V^\sharp\|_\infty$ vanishes.

Proof of Th 3.8 and Cor. 3.1. For any Lipschitz continuous function $F : \mathbb{R}^{n+1} \to \mathbb{R}$, we recall that $\partial^c F(z)$ denote the Clarke's generalized derivative at z, and we define H_F as

$$H_F(z) = 1 + \min_{\substack{u \in U \\ g \in \partial^c F(z)}} \{g^\top \begin{pmatrix} 1 \\ f(z,u) \end{pmatrix}\}. \tag{3.48}$$

The minimum is attained by continuity of the objective, and by compactness of U and $\partial^c F(z)$ (see [51]). Note also that for any $V \in C^1(\mathbb{R}^{n+1})$, for any $z = (t,x) \in \mathbb{R}^{n+1}$, $H_V(t,x) = 1 + \partial_t V(t,x) + \min_{u\in U} \nabla_x V(t,x)^\top f(t,x,u)$, and the argmin is $\mathcal{U}_V(t,x)$. By application of the Maximum Theorem [7, Th. 2.1.6], we know that H_F is lower semi-continuous, since (i) $\partial^c F(z)$ is a compact-valued and upper semi-continuous set-valued map [51], therefore so is $z \mapsto U \times \partial^c F(z)$, and (ii) $(z,u,g) \mapsto g^\top \begin{pmatrix} 1 \\ f(z,u) \end{pmatrix}$ is continuous.

First, we take any $z_1 = (t_1, x_1) \in [0,T] \times X \setminus K$, and we prove that $H_{V^\sharp}(t_1,x_1) \leq 0$. According to Assumption 3.3, $V^\sharp(t_1,x_1) < \infty$, and according to Theorem 3.1 applied to the control system (3.2)-(3.5) on the interval $[0, T^\sharp]$, there exists an optimal trajectory $(x(\cdot), u(\cdot))$ over $[t_1,t_2]$ (with $t_2 > t_1$ since $x_0 \notin K$) starting from (t_1,x_1). By definition, $V^\sharp(t_1,x_1) = t_2 - t_1$. We can also prove that for all $t \in [t_1,t_2]$, $V^\sharp(t,x(t)) = t_2 - t$: (i) the trajectory restricted to $[t,t_2]$, yields an admissible trajectory starting from $(t,x(t))$, therefore $V^\sharp(t,x(t)) \leq t_2 - t$, and (ii) for an optimal trajectory $(\tilde{x}(\cdot), \tilde{u}(\cdot))$ starting from $(t,x(t))$ over $[t,t_3]$, the trajectory following $(x(\cdot), u(\cdot))$ over $[t_1,t]$ and $(\tilde{x}(\cdot), \tilde{u}(\cdot))$ over $[t,t_3]$ is admissible and starting from (t_1,x_1), therefore, $V^\sharp(t,x(t)) + (t - t_1) \geq V^\sharp(t_1,x_1) = t_2 - t_1$, giving $V^\sharp(t,x(t)) \geq t_2 - t$. As $\alpha(t) = V^\sharp(t,x(t)) = t_2 - t$ for all $t \in [t_1,t_2]$, we deduce that

$$\alpha'(t) = -1 \text{ a. e. on } [t_1,t_2]. \tag{3.49}$$

Moreover, since $V^\sharp$ is Lipschitz continuous by assumption, and $t \mapsto (t,x(t))$ is Lipschitz continuous as $x(t)$ is differentiable a.e. with a bounded derivative, Lemma 0.10 gives: $\alpha'(t) =$

$\frac{d(V^\sharp(t,x(t)))}{dt} \geq \min_{g\in\partial^c V^\sharp(t,x(t))} g^\top \begin{pmatrix} 1 \\ \dot{x}(t) \end{pmatrix}$ a.e. on $[t_1,t_2]$. Using that $\dot{x}(t) = f(t,x(t),u(t))$ a.e. on $[t_1,t_2]$, and the definition of $H_{V^\sharp}$:

$$\alpha'(t) \geq \min_{g\in\partial^c V^\sharp(t,x(t))} g^\top \begin{pmatrix} 1 \\ f(t,x(t),u(t)) \end{pmatrix} \geq H_{V^\sharp}(t,x(t)) - 1, \tag{3.50}$$

a.e. on $[t_1,t_2]$. Combining this with Eq. (3.49), we deduce that for almost all $t \in [t_1,t_2]$, $H_{V^\sharp}(t,x(t)) \leq 0$. By lower semi-continuity of $H_{V^\sharp}$ (see above), and by continuity of $x(\cdot)$

$$H_{V^\sharp}(t_1,x_1) \leq 0. \tag{3.51}$$

Second, still for any $(t_1,x_1) \in [0,T]\times X\setminus K$, we observe that there exists $(g,u_1) \in \partial^c V^\sharp(t_1,x_1)\times U$ such that $H_{V^\sharp}(t_1,x_1) = 1 + g^\top \begin{pmatrix} 1 \\ f(t_1,x_1,u_1) \end{pmatrix}$; indeed, we already mentioned that the minimum in (3.48) is attained. Therefore, for any $V \in C^1(\mathbb{R}^{n+1})$

$$1 + \partial_t V(t_1,x_1) + \nabla_x V(t_1,x_1)^\top f(t_1,x_1,u_1) = H_{V^\sharp}(t_1,x_1) + (\nabla V(t_1,x_1) - g)^\top \begin{pmatrix} 1 \\ f(t_1,x_1,u_1) \end{pmatrix} \tag{3.52}$$

$$\leq H_{V^\sharp}(t_1,x_1) + \|\nabla V - \nabla V^\sharp\|(1+C_f), \tag{3.53}$$

the inequality being due to Cauchy-Schwartz inequality, and the definition of $\|\nabla V - \nabla V^\sharp\|$. We know that $H_V(t_1,x_1) \leq \partial_t V(t_1,x_1) + 1 + \nabla_x V(t_1,x_1)^\top f(t_1,x_1,u_1)$ by definition of $H_V(t_1,x_1)$ (as $u_1 \in U$), therefore Eq. (3.53) gives $H_V(t_1,x_1) \leq H_{V^\sharp}(t_1,x_1) + (1+C_f)\|\nabla V - \nabla V^\sharp\|$. Using this inequality and Eq. (3.51), we deduce that for all $(t_1,x_1) \in [0,T]\times X\setminus K$,

$$H_V(t_1,x_1) \leq (1+C_f)\|\nabla V - \nabla V^\sharp\|. \tag{3.54}$$

Third, according to the hypotheses of the theorem, we take any $V \in C^1(\mathbb{R}^{n+1})$, and any closed-loop trajectory $(x_V(\cdot),u_V(\cdot))$ starting at $(0,x_0)$ satisfying the differential inclusion (CL_V) and the state constraints over $[0,T]$. We, then, study the evolution of $V^\sharp$ over this trajectory. As $x_V(t)$ is Lipschitz continuous, Lemma 0.10 yields the existence of $g(t) \in \partial^c V^\sharp(t,x_V(t))$ for almost all $t\in[0,T]$, such that

$$\frac{d}{dt}\left(V^\sharp(t,x_V(t))\right) \leq g(t)^\top \begin{pmatrix} 1 \\ f(t,x_V(t),u_V(t)) \end{pmatrix} \text{ a.e. on } [0,T]. \tag{3.55}$$

As $u_V(t) \in \mathcal{U}_V(t,x_V(t))$, we know that $H_V(t,x_V(t)) = 1 + \nabla V(t,x_V(t))^\top \begin{pmatrix} 1 \\ f(t,x_V(t),u_V(t)), \end{pmatrix}$, and therefore,

$$\frac{d}{dt}\left(V^\sharp(t,x_V(t))\right) \leq -1 + H_V(t,x_V(t)) + (g(t) - \nabla V(t,x_V(t)))^\top \begin{pmatrix} 1 \\ f(t,x_V(t),u_V(t)), \end{pmatrix} \tag{3.56}$$

We deduce, using Cauchy-Schwartz inequality and the definition of $\|\nabla V - \nabla V^\sharp\|$,

$$\frac{d}{dt}\left(V^\sharp(t, x_V(t))\right) \leq -1 + H_V(t, x_V(t)) + (1 + C_f)\|\nabla V - \nabla V^\sharp\|, \tag{3.57}$$

for almost all $[0, T]$. Moreover, for all $t \in [0, t_V)$, $x_V(t) \notin K$. Therefore, we can apply Eq. (3.54) to deduce, in combination with Eq. (3.57), that for almost all $[0, t_V]$, $\frac{d}{dt}\left(V^\sharp(t, x_V(t))\right) \leq -1 + 2(1 + C_f)\|\nabla V - \nabla V^\sharp\|$. By integration, we deduce that for all $t \in [0, t_V]$, $V^\sharp(t, x_V(t)) - \tau^* \leq -t + 2t(1 + C_f)\|\nabla V - \nabla V^\sharp\|$, as $V^\sharp(0, x_V(0)) = V^\sharp(0, x_0) = \tau^*$. This proves Eq. (3.45).

Fourth and finally, we prove the corollary. Due to the definition of t_V, the following (non-exclusive) alternative holds: either $x_V(t_V) \in K$ or $t_V = T$. Moreover, if $\|\nabla V - \nabla V^\sharp\|_\infty \leq \frac{1-\tau^*/T}{2(1+C_f)}$, then $V^\sharp(t, x_V(t)) - \tau^* \leq -t + t(1 - \tau^*/T)$ and $V^\sharp(t, x_V(t)) \leq \tau^*(1 - t/T)$ for all $t \in [0, t_V]$. We notice that if $t_V = T$, then $V^\sharp(t_V, x_V(t_V)) \leq 0$, *i.e.*, $x_V(t_V) \in K$. Coming to the aforementioned alternative, we deduce that $x_V(t_V) \in K$. Moreover, this fact combined with Eq. (3.45) gives us that $0 \leq (\tau^* - t_V) + t_V\, 2(1 + C_f)\|\nabla V - \nabla V^\sharp\|_\infty$, hence

$$t_V\left(1 - 2(1 + C_f)\|\nabla V - \nabla V^\sharp\|_\infty\right) \leq \tau^*. \tag{3.58}$$

By assumption, $1 - 2(1 + C_f)\|\nabla V - \nabla V^\sharp\|_\infty \geq \tau^*/T > 0$, we can thus divide Eq. (3.58) by this quantity to obtain the result of the corollary: $t_V \leq \tau^*/\left(1 - 2(1 + C_f)\|\nabla V - \nabla V^\sharp\|_\infty\right)$. □

In the previous theorem and the corollary, we saw that the suboptimality, in terms of hitting time, of a closed-loop trajectory $(x_V(\cdot), u_V(\cdot))$ satisfying the differential inclusion (CL_V) decreases as the approximation error $\|\nabla V - \nabla V^\sharp\|_\infty$ decreases. Furthermore, we see that the closed-loop trajectory is near-optimal when the approximation error vanishes. We now study a sufficient condition under which the approximation $\|\nabla V_d - \nabla V^\sharp\|_\infty$ can be made arbitrarily small, using a polynomial $V_d(t, x)$ of sufficiently large degree $d \in \mathbb{N}$.

3.3.3 A sufficient regularity condition for the existence of near-optimal controllers based on polynomials

In the case where the value function is twice differentiable, there exist polynomials V_d with such a vanishing approximation error $\|\nabla V_d - \nabla V^\sharp\|_\infty$, and that are near optimal solutions in the hierarchy of semi-infinite programs ($\mathbf{R_d}$).

Theorem 3.9. *Under Assumptions 3.1-3.3, if the value function $V^\sharp$ belongs to $C^2(\mathbb{R}^p|\Sigma)$, and is a subsolution to the HJB equation, then there exist a sequence of polynomials $(V_d(t, x))_{d \in \mathbb{N}^*}$, with $V_d(t, x) \in \mathbb{R}_d[t, x_1, \ldots, x_n]$, and two constants $c_1, c_2 \in \mathbb{R}_{++}$, such that for all $d \in \mathbb{N}^*$,*

- *The polynomial $V_d(t, x)$ is feasible, and $\frac{c_1}{d}$-optimal in the problems ($D_\mathcal{F}$) and ($\mathbf{R_d}$),*
- *The following inequality holds: $\|\nabla V_d - \nabla V^\sharp\|_\infty \leq \frac{c_2}{d}$.*

Under these hypotheses, the polynomials $V_d(t, x)$ are subsolutions to the HJB equation, and form a maximizing sequence of the problem ($D_\mathcal{F}$); we also notice that the hierarchy of semi-infinite programs ($\mathbf{R_d}$) converges in $O(\frac{1}{d})$ in terms of objective value. Moreover, according to Cor. 3.1, for any sequence of closed-loop trajectories $(x_{V_d}(\cdot), u_{V_d}(\cdot))$, the associated hitting times converge to the minimal time τ^*: this is a minimizing sequence of trajectories for the optimal time control problem (3.2)-(3.5).

Proof. By definition of $C^2(\mathbb{R}^{n+1}|\Sigma)$, there exists a function $Q \in C^2(\mathbb{R}^{n+1})$ such that $V^\sharp(z) = Q(z)$ and $\nabla V^\sharp(z) = \nabla Q(z)$ for all $z \in \Sigma$. In application of Lemma 0.8, as Q has a locally Lipschitz gradient since it is twice differentiable, there exists a constant $A \in \mathbb{R}_{++}$, and a sequence of polynomials $(w_d(t,x))_{d\in\mathbb{N}^*}$ with $w_d(t,x) \in \mathbb{R}_d[t, x_1, \ldots, x_n]$ and such that for all $(t,x) \in \Sigma$, $|w_d(t,x) - Q(t,x)| \leq \frac{A}{d}$ and $\|\nabla w_d(t,x) - \nabla Q(t,x)\|_2 \leq \frac{A}{d}$. With $\alpha_d = \frac{A(1+C_f)}{d}$, and $\beta_d = \frac{A}{d}(1+T+TC_f)$, we define the polynomial $V_d(t,x) = w_d(t,x) + \alpha_d t - \beta_d \in \mathbb{R}_d[t, x_1, \ldots, x_n]$. First, we notice that $\|\nabla V_d - \nabla V^\sharp\|_\infty \leq \|\nabla w_d - \nabla V^\sharp\|_\infty + \alpha_d \leq \frac{A(2+C_f)}{d}$ for all $d \geq 1$. This proves the second point of the theorem, having defined the constant $c_2 = A(2+C_f)$, which is independent from d. We prove now the first point. For all $d \geq 1$, and $(t,x,u) \in [0,T] \times X \times U$, $\partial_t V_d(t,x) + 1 + \nabla_x V_d(t,x)^\top f(t,x,u) = \alpha_d + \partial_t V^\sharp(t,x) + 1 + \nabla_x V^\sharp(t,x)^\top f(t,x,u) + (\nabla w_d(t,x) - \nabla V^\sharp(t,x))^\top \begin{pmatrix} 1 \\ f(t,x,u) \end{pmatrix} \geq \alpha_d + (\nabla w_d(t,x) - \nabla V^\sharp(t,x))^\top \begin{pmatrix} 1 \\ f(t,x,u) \end{pmatrix}$, as $V^\sharp$ is a subsolution to the HJB equation, hence satisfies Eq. (3.8). Using the Cauchy-Schwartz inequality, we obtain $\partial_t V_d(t,x) + 1 + \nabla_x V_d(t,x)^\top f(t,x,u) \geq \alpha_d - \|\nabla w_d(z) - \nabla V^\sharp(z)\|_2(1+C_f) \geq \alpha_d - \frac{A}{d}(1+C_f) = 0$. This proves that V_d satisfies Eq. (3.8). It also satisfies Eq. (3.9), because for any $(t,x) \in [0,T] \times K$,

$$\begin{aligned} V_d(t,x) &= w_d(t,x) + \alpha_d t - \beta_d && (3.59)\\ &\leq V^\sharp(t,x) + \frac{A}{d} + \alpha_d t - \beta_d && (3.60)\\ &\leq V^\sharp(t,x) + \frac{A}{d} + \alpha_d T - \beta_d && (3.61)\\ &\leq V^\sharp(t,x) = 0. && (3.62) \end{aligned}$$

since $\frac{A}{d} + \alpha_d T - \beta_d = 0$ by definition of β_d, and since $x \in K$. We deduce that V_d is feasible in ($\mathbf{R_d}$). Its objective value is $V_d(0,x_0) \geq w_d(0,x_0) - \beta_d \geq V^\sharp(0,x_0) - \frac{A}{d} - \beta_d = V^\sharp(0,x_0) - \frac{c_1}{d}$, where $c_1 = A(2+T+TC_f)$. As $V^\sharp(0,x_0) = V^*(0,x_0)$ due to Assumption 3.2, $V_d(0,x_0) \geq V^*(0,x_0) - \frac{c_1}{d} \geq \mathsf{val}(D_{\mathcal{F}}) - \frac{c_1}{d} \geq \mathsf{val}(\mathbf{R_d}) - \frac{c_1}{d}$, and we therefore conclude that V_d is $\frac{c_1}{d}$-optimal in ($D_{\mathcal{F}}$) and ($\mathbf{R_d}$). $\square$

Remark 3.3. *Admittedly, the hypothesis in Theorem 3.9 that the value function $V^\sharp$ belongs to $C^2(\mathbb{R}^p|\Sigma)$ is stringent. It is worth noting, however, that there exist systems that satisfy this hypothesis. Here is an example: $\dot{x}(t) = u(t)$, $x(t) \in X = [0,1]^2$, $\|u(t)\| \leq 1$ and $K = \{0\} \times [0,1]$. The value function associated with the horizon $T = \infty$ is $V^\sharp(t,x) = x_1$.*

3.4 Illustrative examples

We implemented and tested the proposed methodology on three Minimal Time Control Problems: a generalization of the Zermelo problem, a regatta problem and a generalization of the Brockett integrator. The numerical examples in this section were implemented with our `Julia` package `MinTimeControl.jl`[1], and run on a 64-bit Ubuntu computer with 32 Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz and 64 GB RAM. In our implementation of Algorithm 4, the master problem (3.33) is solved with the simplex algorithm of the commercial solver `Gurobi 10.0` [94]. The separation oracle (3.32) is implemented with a random sampling scheme (with 500,000 samples at each iteration to detect a violated constraint), and with the global optimization solver `SCIP 8` [23], for the certification at the last iterate. This solver is used with a relative tolerance $\delta = 10^{-4}$, and with a time limit of $10,000s$. At each iteration, we add a maximum of 100 points to the set $\mathcal{Y}^k$. We also precise that we compute a heuristic trajectory based on the particularities of each problem; this heuristic is not optimal, but provides an upper bound T on the minimum time, and therefore, a relevant time horizon $[0, T]$. The trajectory resulting from the heuristic is used to initialize the set $\mathcal{Y}^0$, in the sense that we enforce the HJB inequalities for some points of this trajectory. During the iterations of the algorithm, we obtain functions $V_{\theta^k}(t, x)$ and we simulate the associate feedback trajectory defined by the differential inclusion (CL_V); if the obtained trajectory reaches the target set, it gives us an upper bound. Those trajectories are also used to enrich the set $\mathcal{Y}^k$. For all the experiments, the regularization parameter is $\mu = 10^{-5}$, and the tolerance $\epsilon = 10^{-3}$.

Tables 3.1, 3.3 and 3.4 present the numerical results for three different applications. The different columns of these tables are the following:

- "$d \in \mathbb{N}$" is the degree of the polynomial basis used.
- "Estimated value of ($\mathbf{R_d}$)" stands for the value $V_\theta(0, x_0)$, where θ is the output of Algorithm 4, using the sampling oracle. This estimated value of ($\mathbf{R_d}$) is not an exact lower bound, since this sampling oracle does not provide the guarantee that θ is indeed feasible in (**SIP**).
- "Certified lower bound for ($\mathbf{R_d}$)" stands for $V_\theta(0, x_0) - \hat{\phi}(\theta)(1 + T)$, where θ is as defined above and $\hat{\phi}(\theta)$ is a guaranteed upper bound on $\phi(\theta)$, the feasibility error of θ in (**SIP**), computed by the global optimization solver `SCIP 8`, playing the role of δ-oracle. As $V_\theta(t, x) + \hat{\phi}(\theta)(t - 1 - T)$ is therefore feasible in ($\mathbf{R_d}$), the value $V_\theta(0, x_0) - \hat{\phi}(\theta)(1 + T)$ is a guaranteed lower bound on $\mathsf{val}(\mathbf{R_d})$, and, therefore, on $V^*(0, x_0)$.
- "Value feedback control (CL_V)" is the hitting time of the best feasible control generated along the iterations: either with the heuristic control at the first iteration, or the closed-loop controlled trajectory defined by (CL_V) associated with $V = V_{\theta^k}$ at iteration k of Algorithm 4.

[1]This package is available at `github.com/aoustry/MinTimeControl.jl`

- "Solution time (in s)" is the total computational time of the heuristic control, of the iterations of Algorithm 4 including the sampling oracle, and of the closed-loop trajectory simulation. Therefore, this is the computational time needed to obtain the estimated value of ($\mathbf{R_d}$) (second column), and the best feasible control (fourth column).
- "Iterations number" is the total number of iterations of Algorithm 4.
- "Certification time (in s)" is the computational time of the global optimization solver `SCIP 8`, playing the role of δ-oracle, to compute the aforementioned bound $\hat{\phi}(\theta)$, and deduce the certified lower bound (third column).

3.4.1 A time-dependent Zermelo problem

We consider a time-dependent nonlinear system with $n = 2$ and $m = 2$, defined by

$$\dot{x}_1(t) = u_1(t) + \frac{1}{2}(1+t)\ \sin(\pi x_2(t)) \tag{3.63}$$

$$\dot{x}_2(t) = u_2(t), \tag{3.64}$$

with the state constraint set $X = [-1, 1] \times [-1, 0]$, the control set $U = B(0, 1)$. This is the celebrated Zermelo problem, but with a river flow gaining in intensity over time. Figure 3.1 gives a representation of this flow. The initial condition is $x(0) = (0, -1)$, and the target set is $K = B(0, r)$, for $r = 0.05$. The travel time associated with the heuristic control, consisting in following a straight trajectory, is 1.261 (see Figure 3.2). Table 3.1 presents the numerical results for different values of d. We see that the value of the linear semi-infinite program ($\mathbf{R_d}$) quickly converges as d increases: starting from $d = 6$, the 4 first digits of the estimated value (second column) reach a plateau which corresponds to the value (1.100) of the best feasible trajectory we generate with our feedback control. As regards the certified lower bound, the best value (1.092) is obtained for $d = 5$. For greater d, we see that increasing d deteriorates the tightness of the best certified bound. This is due to the fact that the separation problem becomes more difficult, with two consequences: (i) the sampling fails to detect unsatisfied constraints, so Algorithm 4 stops with a solution that has a real infeasibility $\phi(\theta)$ larger than ϵ (targeted tolerance), and (ii) the global optimization solver called afterwards does not manage to solve the separation problem to global optimality within the time limit (case $d \in \{7, 8\}$), given only a large upper bound $\hat{\phi}(\theta)$ on the true infeasibility $\phi(\theta)$. We notice that as soon as $d \geq 3$, the feedback control defined by (CL_V) (see Section 3.3) yields a trajectory that is 13% faster than the heuristic trajectory. In summary, we obtain a certified optimization gap of 0.7% for this minimal time control problem.

In the special case of this non-polynomial controlled system, a polynomial reformulation

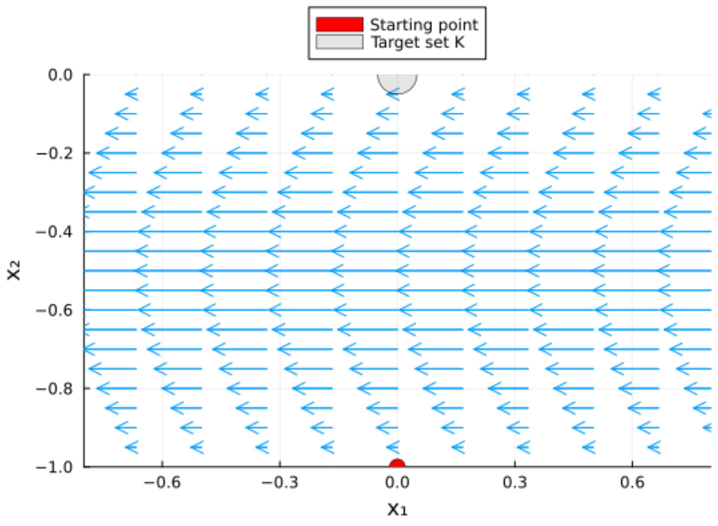


Figure 3.1: Representation of the water flow in the Zermelo problem

$d \in \mathbb{N}$	**Estimated val. of** $(\mathbf{R_d})$	**Certified LB for** $(\mathbf{R_d})$	**Val. feedback control** (CL_V)	**Solution time (in** s**)**	**Iter. nbr.**	**Certification time (in** s**)**
2	0.952	0.945	1.261	2	4	1
3	1.064	1.044	1.101	12	14	12
4	1.096	1.084	1.100	22	17	1530
5	1.099	1.092	1.100	54	22	4000
6	1.100	1.059	1.100	60	18	2330
7	1.100	1.051	1.100	105	24	TL
8	1.100	0.690	1.100	215	31	TL

Table 3.1: Time-dependent Zermelo problem: lower and upper bounds, and computational times for various degrees of the semi-infinite hierarchy $(\mathbf{R_d})$

exists, at the price of increasing the dimension of the system to $n = 4$:

$$\dot{x}_1(t) = u_1(t) + \frac{1}{2}(1+t)\ \sin(\pi x_2(t)) \tag{3.65}$$

$$\dot{x}_2(t) = u_2(t) \tag{3.66}$$

$$\dot{x}_3(t) = -\pi x_4(t) u_2(t) \tag{3.67}$$

$$\dot{x}_4(t) = \pi x_3(t) u_2(t), \tag{3.68}$$

with the state constraint set $\hat{X} = [-1,1] \times [-1,0] \times [-1,1] \times [-1,0]$, the control set $\hat{U} = B(0,1)$, the terminal set $\hat{K} = B(0,r) \times [-1,1] \times [-1,0]$, and the initial condition $x_0 = (0,-1,-1,0)$. The dynamics maintain the equalities $x_3(t) = \cos(\pi x_2(t))$ and $x_4(t) = \sin(\pi x_2(t))$. We are therefore able to compare our approach with the SoS hierarchy, which consists of replacing semi-infinite inequalities in $(\mathbf{R_d})$ with SoS positivity certificates. For each order k of the hierarchy, i.e., for a maximal degree $d = 2k$ of the polynomial basis, this yields a semidefinite programming problem that we solve with the solver `MOSEK`, used with the package `SumOfSquares.jl`. We obtain a

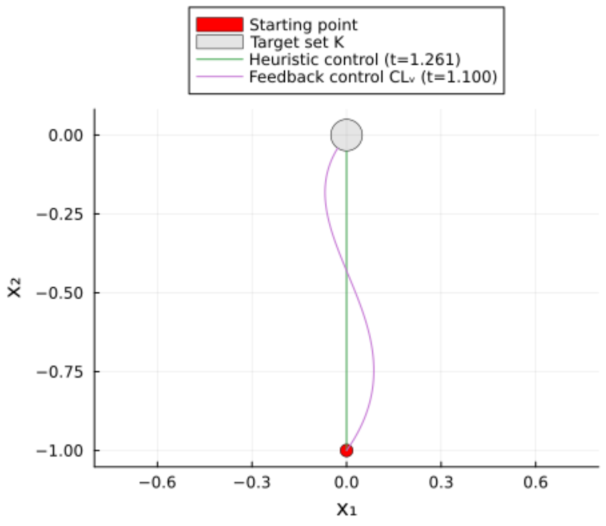


Figure 3.2: Time-dependent Zermelo problem: heuristic control and feedback control ($d = 6$)

polynomial $V(t, x_1, x_2, x_3, x_4)$ that is solution of the corresponding relaxation. Based on this polynomial, we can also generate a feedback controlled trajectory solution of the differential inclusion (CL_V). Table 3.2 compares the performance of the semi-infinite programming and

	SIP hierarchy			**SOS hierarchy**		
$d \in \mathbb{N}$	**Est./Cert. LB**	**Val. feedback control** (CL_V)	**Sol./Cert. time (in** s**)**	**Cert. LB**	**Val. feedback control** (CL_V)	**Sol. time (in** s**)**
2	0.952/0.945	1.261	2/1	0.533	1.261	≤ 1
4	1.096/1.084	1.101	22/1530	1.064	1.105	1
6	1.100/1.059	1.100	60/2330	1.099	1.100	12
8	1.100/0.690	1.100	215/TL	1.100	1.100	190

Table 3.2: Time-dependent Zermelo problem: comparing the SIP and the SOS approaches

the SoS approaches. We see that for low-degree polynomials ($d \leq 4$), the semi-infinite hierarchy gives better lower bounds than the SoS hierarchy, although at a higher computational time in the case $d = 4$. For $d \in \{6, 8\}$, the lower bound of the SoS hierarchy is tight, while only the estimated lower-bound of the semi-infinite hierarchy is tight: to obtain a certified lower bound, the SoS hierarchy performs better. For these values of the degree d, this optional certification step (calling to the global optimization solver) is costly in the proposed semi-infinite approach. For this first example, where the SoS hierarchy is applicable since a polynomial reformulation of the dynamical system exists, the semi-infinite approach is slower than the SoS hierarchy.

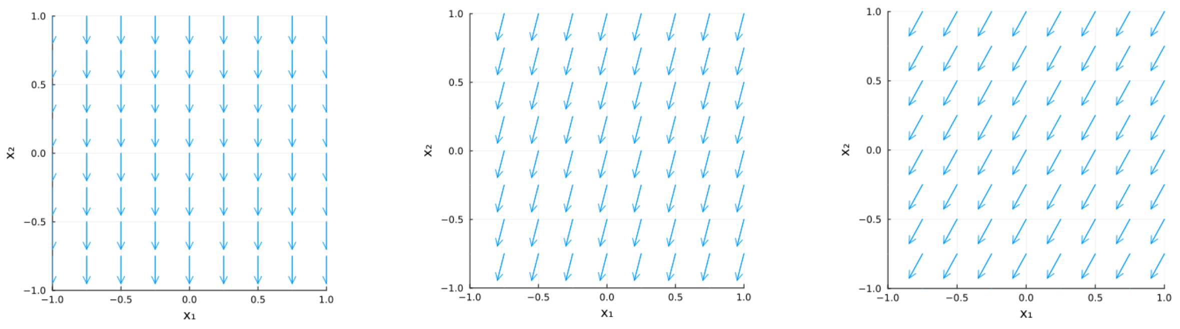
3.4.2 A regatta toy-model

We consider a time-dependent nonlinear (and non-polynomial) system with $n = 2$ and $m = 1$, defined by

$$\dot{x}_1(t) = \mathsf{windspeed}(t)\ \mathsf{polar}\,[u(t)]\ \cos(u(t) + \mathsf{windangle}(t)) \tag{3.69}$$

$$\dot{x}_2(t) = \mathsf{windspeed}(t)\ \mathsf{polar}\,[u(t)]\ \sin(u(t) + \mathsf{windangle}(t)), \tag{3.70}$$

where $\mathsf{windspeed}(t) = 2 + t$, $\mathsf{windangle}(t) = \frac{\pi}{2}(1 - 0.4t)$ and $\mathsf{polar}[u] = |\sin(\frac{2u}{3})|$. In this model, the control $u(t)$ represents the relative angle between the heading of the boat and the (origin) direction of the wind. The evolution of the wind direction over time is depicted in Figure 3.3. The polar curve of this toy model of a sailing boat is represented in Figure 3.4; this figure clearly shows that this model does not satisfy Assumption 3.1. Although the absence of duality gap between the control problem and the LP problem $(D_{\mathcal{F}})$ is, therefore, not guaranteed, we see in Table 3.3 that if this gap exists in this case, it is low (below 1.6%). The state constraint set is $X = [-1, 1]^2$, and the control set $U = [-\pi, \pi]$. The initial condition is $x(0) = (0, -1)$, and the target set is $K = B(0, r)$, for $r = 0.05$. The travel time associated with the heuristic control, consisting in following a straight trajectory, is 1.278 (see Figure 3.5).



(a) $t = 0$ (b) $t = 0.4$ (c) $t = 0.8$

Figure 3.3: Regatta problem: wind direction at different times

$d \in \mathbb{N}$	**Estimated val. of** $(\mathbf{R_d})$	**Certified LB for** $(\mathbf{R_d})$	**Val. feedback control** (CL_V)	**Solution time (in** s**)**	**Iter. nbr.**	**Certification time (in** s**)**
2	0.834	0.829	1.278	6	6.0	2
3	0.896	0.880	0.912	16	10.0	56
4	0.904	0.896	0.915	31	13.0	498
5	0.907	0.774	0.912	52	16.0	1020
6	0.907	0.799	0.912	100	22.0	1930
7	0.908	0.691	0.912	190	29.0	7600
8	0.908	0.000	0.911	312	33.0	TL

Table 3.3: Regatta problem: lower and upper bounds, and computational times for various degrees of the semi-infinite hierarchy $(\mathbf{R_d})$

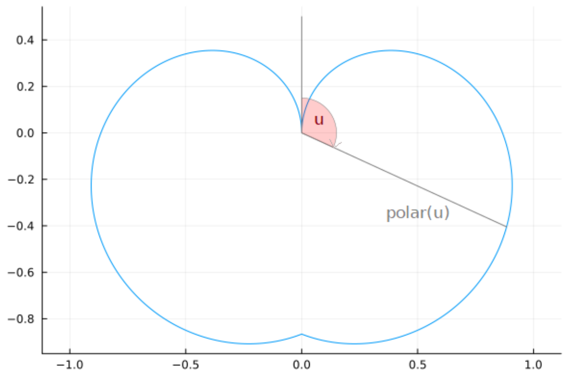


Figure 3.4: Regatta problem: the polar curve of the sailing boat

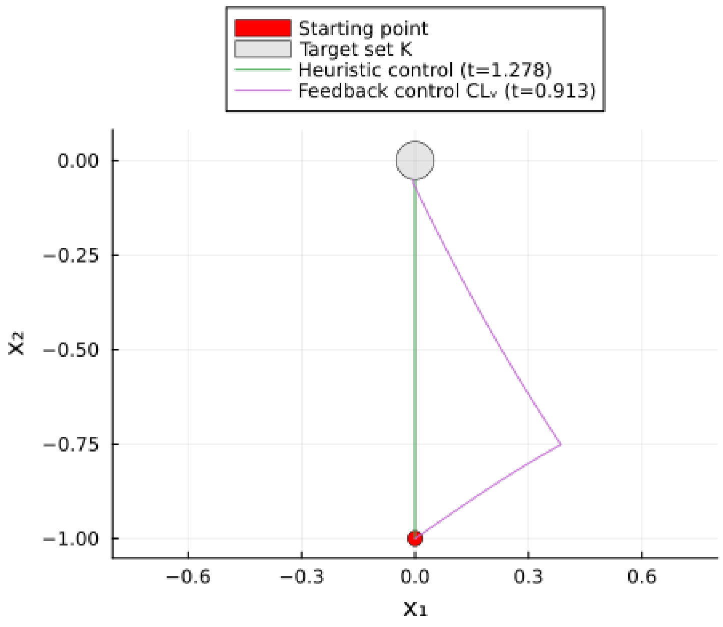


Figure 3.5: Regatta problem: heuristic control and feedback control ($d = 6$)

We see that the highest estimated value of ($\mathbf{R_d}$), for $d = 7$ and $d = 8$, is 0.5% lower than the value (0.913) of the best feasible trajectory obtained with the feedback controller for $d = 6$. This feedback controller yields a trajectory which is 29% faster than the heuristic trajectory. As regards the certified lower bound, $d = 4$ yields the best result (0.896), at a price of a running time of 498s for the exact oracle (`SCIP 8`). For the same reasons as in the previous application, a larger d does not necessarily mean a better certified lower bound obtained within the time limit. In summary, we obtain a certified optimization gap of 1.6% for this minimal time control problem.

3.4.3 A generalized Brockett integrator

For $n \in \mathbb{N}^*$ and $m = n - 1$ and given a continuous mapping $q\colon \mathbb{R}^n \to \mathbb{R}^m$, we consider the following generalization of the Brockett integrator [148],

$$\dot{x}_i(t) = u_i(t) \quad \forall i \in \{1, \dots, n-1\} \tag{3.71}$$

$$\dot{x}_n(t) = q(x(t))^\top u(t). \tag{3.72}$$

In particular, we study this system for $n = 6$, and

$$q(x) = \Big(2/(2+x_4), -x_1, -\cos(x_1 x_3), \exp(x_2), x_1 x_2 x_6\Big). \tag{3.73}$$

The state constraint set is $X = [-1,1]^5$, and the control set is $U = B(0,1)$. The initial condition is $(x_0) = \frac{1}{2}\mathbf{1}$, and the target set is $K = B(0,r)$, for $r = 0.05$. The travel time associated with the heuristic control is 1.377.

$d \in \mathbb{N}$	**Estimated val. of** $(\mathbf{R_d})$	**Certified LB for** $(\mathbf{R_d})$	**Val. feedback control** (CL_V)	**Solution time (in** s**)**	**Iter. nbr.**	**Certification time (in** s**)**
2	1.071	0.763	1.071	220	28	73
3	1.072	0.000	1.071	5630	133	TL
4	1.072	0.000	1.070	147400	319	TL

Table 3.4: Generalized Brockett integrator: lower and upper bounds, and computational times for various degrees of the semi-infinite hierarchy $(\mathbf{R_d})$

Since this system has a larger dimension than the other two examples, we see that the computation times are longer for the same degree d. Already for $d = 2$, we obtain an estimated value of $(\mathbf{R_d})$ that is within 0.1% of the value of the feedback control (1.070). This feedback control yields an improvement of 22% over the heuristic trajectory. Note that the estimated values of $(\mathbf{R_d})$ computed by Algorithm 4 with the (inexact) sampling oracle are slightly larger than the value of the best trajectory we computed: thus, these estimates are not valid lower bounds, but only estimates of the value of the minimum time control problem. Regarding the certification of lower bounds, the global optimization solver `SCIP 8` fails to produce tight upper and lower bounds on $\phi(\theta)$, the infeasibility of the solution θ returned by Algorithm 4. Therefore, the resulting certified lower bounds are not tight either. In summary, we obtain a certified optimization gap of 29% for this minimal time control problem.

3.5 Conclusion

We apply the dual approach in minimal time control, which consists in searching for maximal subsolutions of the HJB partial differential equation to generic nonlinear, even non-polynomial, controlled systems. The basis functions used to generate these subsolutions are polynomials that are subject to semi-infinite constraints. We prove the theoretical convergence of the resulting hierarchy of semi-infinite linear programs, and our numerical tests on three different systems show good convergence properties in practice. These results show that using a random sampling oracle allows a good approximation of the value of the control problem. It is possible for small systems to obtain tight and certified lower bounds based on a global optimization solver. Finally, the numerical experiments also show that the computed subsolutions of the HJB equation help to recover near-optimal controls in a closed-loop form. As illustrated in these numerical experiments, the advantage of our approach based on semi-infinite programming, compared to the sum-of-squares approach, is the ability to handle non-polynomial systems. In the numerical example where a polynomial reformulation of the system was possible, the sum-of-squares approach was, however, faster.

A promising avenue for continuing this work is to investigate using another basis of functions to search for an approximate value function, resulting in other semi-infinite programming hierarchies with convergence guarantees. In particular, it would be relevant to use nondifferentiable functions in the basis to improve approximation capabilities for nondifferentiable value functions. We could investigate whether using nondifferentiable functions for the basis could help extend the approximation result (Theorem 3.9) to the case of nondifferentiable value functions. Another avenue of research is to extend the algorithmic approach and the theoretical results to a generic optimal control problem.

Part II

Global optimization of finite and semi-infinite power flow problems

Chapter 4

Conic programming and MILP scheme for global optimization of AC power flows

The Alternating Current Optimal Power Flow (ACOPF) problem is a foundational optimization problem related to the dispatching of electricity in a power network. This nonconvex optimization problem has been of interest to the electrical engineering and mathematical optimization communities for several decades. The ACOPF problem with power magnitude limits on lines reads as follows

$$\left.\begin{array}{rl}
\min\limits_{V\in\mathbb{C}^{\mathcal{B}},\, S\in\mathbb{C}^{\mathcal{G}}} & \sum\limits_{g\in\mathcal{G}}\Big(c_{0g}+c_{1g}\,\mathsf{Re}(S_g)+c_{2g}\,\mathsf{Re}(S_g)^2\Big)\\
\text{s.t. } \forall b\in\mathcal{B} & \underline{v}_b\le |V_b|\le \overline{v}_b\\
\forall g\in\mathcal{G} & \underline{s}_g\le S_g\le \overline{s}_g\\
\forall b\in\mathcal{B} & \sum\limits_{g\in\mathcal{G}_b} S_g - S^{\mathsf{d}}_b = \langle M_b, VV^{\mathbf{H}}\rangle\\
\forall (b,a)\in\mathcal{L} & |(Y^{\mathsf{ff}}_{ba})^*|V_b|^2+(Y^{\mathsf{ft}}_{ba})^*V_bV_a^*|\le \overline{S}_{ba}\\
\forall (a,b)\in\mathcal{L} & |(Y^{\mathsf{tt}}_{ab})^*|V_b|^2+(Y^{\mathsf{tf}}_{ab})^*V_bV_a^*|\le \overline{S}_{ab}\\
\forall (b,a)\in\mathcal{L}\cup\mathcal{L}^R & \underline{\theta}_{ba}\le \angle V_b-\angle V_a\le\overline{\theta}_{ba}.
\end{array}\right\} \qquad (\textbf{ACOPF})$$

As defined in the Introduction, the oriented graph $\mathcal{N}=(\mathcal{B},\mathcal{L})$ represents the electrical network. The need for an orientation of the edges (the electrical lines) results from the definition of the orientation of the electricity flows. The set $\mathcal{G}_b$ is the set of generators installed at the bus b, and the total set of generators is $\mathcal{G}=\cup_{b\in\mathcal{B}}\mathcal{G}_b$. The different parameters of this problem, presented in the Introduction, are recalled in Table 4.1. The matrix M_b, for $b\in\mathcal{B}$, is defined as

$$M_b = Y^s_b E_{bb} + \sum_{a:(b,a)\in\mathcal{L}}\Big(Y^{\mathsf{ff}}_{ba}E_{bb}+Y^{\mathsf{ft}}_{ba}E_{ba}\Big)+\sum_{a:(b,a)\in\mathcal{L}^R}\Big(Y^{\mathsf{tt}}_{ab}E_{bb}+Y^{\mathsf{tf}}_{ab}E_{ba}\Big).$$

Parameters	Index set	Meaning
$c_{0g}, c_{1g} \in \mathbb{R}, c_{2g} \in \mathbb{R}_+$	$g \in \mathcal{G}$	Generator's cost parameters
$\underline{s}_g, \overline{s}_g \in \mathbb{C}$	$g \in \mathcal{G}$	Generator's domain bounds
$\underline{v}_b, \overline{v}_b \in \mathbb{R}_+$	$b \in \mathcal{B}$	Voltage magnitude bounds
$S^{\mathsf{d}}_b \in \mathbb{C}$	$b \in \mathcal{B}$	Power demand
$Y^s_b \in \mathbb{C}$	$b \in \mathcal{B}$	Shunt admittance
$Y^{\mathsf{ff}}_{ba}, Y^{\mathsf{ft}}_{ba}, Y^{\mathsf{tf}}_{ba}, Y^{\mathsf{tt}}_{ba} \in \mathbb{C}$	$(b,a) \in \mathcal{L}$	Line impedance matrix
$\overline{S}_{ba} \in \mathbb{R}_+$	$(b,a) \in \mathcal{L} \cup \mathcal{L}^R$	Power magnitude limit
$\underline{\theta}_{ba}, \overline{\theta}_{ba} \in [-\frac{\pi}{2}, \frac{\pi}{2}]$	$(b,a) \in \mathcal{L} \cup \mathcal{L}^R$	Angle difference limits

Table 4.1: Main parameters of the ACOPF problem

Related works

Thanks to interior-point methods (IPM) for nonlinear programming (NLP) [36, 238], developed since the 1990s, the computation of ACOPF feasible solutions and local minimizers is possible, even for instances of several thousand nodes [9, 116]. However, since the ACOPF problem is nonconvex, there is no guarantee that a local minimizer found by a NLP algorithm is a global minimizer. To bound the optimality gap of such local minimizers, several convex relaxations have been introduced in the last decade, such as LP, SOCP, convex QCQP, or SDP relaxations. A review of the numerous relaxation techniques for the ACOPF problem is available in [165]. Leveraging NLP algorithms and convex relaxations techniques, several approaches emerged to solve the ACOPF problem to global optimality. We gather these works for globally solving the ACOPF problem in four different categories.

Relaxation strengthening and bound tightening. Strengthening the classical convex relaxations [165], such as the rank relaxation, helps improve the corresponding lower bounds. This strengthening is possible through additional valid inequalities coming from the polar formulation of the ACOPF problem [56, 122, 124] or derived from the Reformulation-Linearization Technique (RLT) [200]. Feasibility-Based Bound Tightening (FBBT) and Optimization-Based Bound Tightening (OBBT) techniques [13], the latter being based on the value of a known feasible solution, are also known to be particularly efficient for the ACOPF problem [56, 212]. Even if these methods do not have a guarantee of convergence toward a global solution in the general case, these papers report that they significantly reduce the optimality gap and even close the gap in some instances.

Moment-Sum-of-Squares hierarchy. The moment sum-of-squares (SoS) hierarchy of relaxations for polynomial optimization problems [130] has been applied to the ACOPF problem in several works [88, 87, 163, 167]. The convergence of the relaxation values towards the optimal value of the ACOPF problem is proven, at the price of the rapidly increasing size and computational cost of the resulting convex relaxations. In practice, only the first and second-order relaxations are solvable for medium-scale ACOPF instances, using the sparse variant of the moment SoS hierarchy [131].

Spatial branch-and-bound. Other global optimization approaches for the ACOPF problem follow spatial B&B schemes [14]. To obtain a lower bound at each node of the exploration tree, these algorithms may use a SOCP relaxation [123], a convex QCQP relaxation [85, 86] or a SDP relaxation [48].

Piecewise convex relaxations. Rather than implementing a spatial B&B algorithm from scratch, an alternative is to encode branching decisions via binary variables and use an off-the-shelf Mixed-Integer Programming (MIP) solver. This leads to piecewise convex relaxations, which can be iteratively refined. This is the approach followed in [152], leading to convex Mixed-Integer Quadratically Constrained Programming (MIQCP) problems.

Contributions and organization of the chapter

In this quest towards global optimization, the contributions of this chapter are manifold. First, we add valid inequalities to strengthen the SDP relaxation, which yields a conic programming relaxation. These valid inequalities dominate the *lifted nonlinear cuts* introduced in [56] for the same purpose. Second, leveraging the conic programming constraints, we propose a global optimization algorithm that solves a sequence of dynamically generated piecewise linear relaxations, *i.e.*, MILP problems. Contrary to [152], a previous paper using piecewise relaxations for the ACOPF problem, we do not use MIQCP but MILP models, which integrate cuts from the conic relaxation. The integration of linear cuts derived from the conic relaxation may be seen as a semi-infinite discretization scheme, as presented in this dissertation's Introduction. Third, we apply this algorithm to the IEEE PES PGLib [9] benchmark and compare the optimality gaps with three recent global optimization approaches [87, 94, 212]. In Section 4.1, we present the ACOPF problem and an equivalent reformulation of it. Section 4.2 introduces our valid inequalities, the resulting conic programming relaxation, and our bound tightening procedure. The iterative MILP scheme is presented in Section 4.3 and the numerical experiments in Section 4.4.

4.1 Extended-variables formulation for the ACOPF problem

The global optimization algorithm proposed in this chapter is based on an equivalent formulation of the ACOPF problem. This formulation includes extended variables W_{ba} representing $V_b V_a^*$. Classically, we do not add a variable W_{ba} for all pairs $(b, a) \in \mathcal{B} \times \mathcal{B}$, but only for certain pairs built from a *tree decomposition* of the graph $\mathcal{N}$.

Definition 4.1. *A tree decomposition $\mathcal{T}$ of the graph $\mathcal{N} = (\mathcal{B}, \mathcal{L})$ is a tree where each node $k \in \mathcal{T}$ is associated with a set $\mathcal{B}_k \subset \mathcal{B}$, and satisfying the following properties*

- *The union of the subsets $\mathcal{B}_k$ equals the set $\mathcal{B}$: $\bigcup_{k \in \mathcal{T}} \mathcal{B}_k = \mathcal{B}$,*
- *For every $(b, a) \in \mathcal{L}$, there exists $k \in \mathcal{T}$ s.t $\{b, a\} \subset \mathcal{B}_k$,*
- *If $b \in \mathcal{B}_k \cap \mathcal{B}_\ell$ for any $k, \ell \in \mathcal{T}$, then $b \in \mathcal{B}_j$ for all nodes j of the tree $\mathcal{T}$ in the unique path between k and ℓ.*

We consider a given tree decomposition $\mathcal{T}$ of the graph $\mathcal{N}$, and we introduce the symmetric set $\mathcal{E} \subset \mathcal{B} \times \mathcal{B}$ of arcs defined as $\mathcal{E} = \cup_{k \in \mathcal{T}} \mathcal{B}_k \times \mathcal{B}_k$. As a matter of fact, the sets $\mathcal{B}_k$ are cliques of the undirected graph induced by $(\mathcal{B}, \mathcal{E})$. In this respect, the sets $\mathcal{B}_k$ are called *cliques*. We define $\mathbb{H}(\mathcal{E}) = \{W \in \mathbb{C}^{\mathcal{E}} : \forall (b, a) \in \mathcal{E}, W_{ba} = W^*_{ab}\}$, the set of Hermitian matrices with sparsity pattern $\mathcal{E}$. For any $k \in \mathcal{T}$, we denote by $W_{\mathcal{B}_k, \mathcal{B}_k}$ the matrix $(W_{ba})_{(b,a) \in \mathcal{B}_k^2}$. With this notation, we reformulate (**ACOPF**) as

$$\left.\begin{aligned}
\min_{S \in \mathbb{C}^{\mathcal{G}},\, W \in \mathbb{H}(\mathcal{E})} \quad & \sum_{g \in \mathcal{G}} \left(c_{0g} + c_{1g}\,\mathsf{Re}(S_g) + c_{2g}\,\mathsf{Re}(S_g)^2\right) \\
\text{s.t. } \forall g \in \mathcal{G} \quad & \underline{s}_g \leq S_g \leq \overline{s}_g \\
\forall b \in \mathcal{B} \quad & \underline{v}_b^2 \leq W_{bb} \leq \overline{v}_b^2 \\
\forall b \in \mathcal{B} \quad & \sum_{g \in \mathcal{G}_b} S_g - S_b^{\mathsf{d}} = \langle M_b, W \rangle \\
\forall (b,a) \in \mathcal{L} \quad & |(Y_{ba}^{\mathsf{ff}})^* W_{bb} + (Y_{ba}^{\mathsf{ft}})^* W_{ba}| \leq \overline{S}_{ba} \\
\forall (b,a) \in \mathcal{L}^R \quad & |(Y_{ab}^{\mathsf{tt}})^* W_{bb} + (Y_{ab}^{\mathsf{tf}})^* W_{ba}| \leq \overline{S}_{ba} \\
\forall (b,a) \in \mathcal{L} \cup \mathcal{L}^R \quad & \tan(\underline{\theta}_{ba})\mathsf{Re}(W_{ba}) \leq \mathsf{Im}(W_{ba}) \leq \tan(\overline{\theta}_{ba})\mathsf{Re}(W_{ba}) \\
\forall (b,a) \in \mathcal{E} \quad & |W_{ba}|^2 = W_{bb} W_{aa} \qquad\qquad (\star) \\
\forall k \in \mathcal{T} \quad & W_{\mathcal{B}_k, \mathcal{B}_k} \succeq 0.
\end{aligned}\right\} \quad (\mathbf{ACOPF}_W)$$

While the clique-based SDP relaxation is well known, this clique-based reformulation of the ACOPF problem itself is not stated in the literature, as far as we know. Yet, we acknowledge that the proof of Theorem 4.1 is closely related to the developments presented in [48].

Theorem 4.1. *A pair (S, W) is feasible (resp. optimal) in (**ACOPF$_W$**) if and only if there exists $V \in \mathbb{C}^{\mathcal{B}}$ such that (S, V) is feasible (resp. optimal) in (**ACOPF**) and $W_{ba} = V_b V_a^*$ for all $(b, a) \in \mathcal{E}$.*

Proof. We prove the equivalence for the feasibility, which also proves the equivalence for the optimality since both problems share the same objective value. We take (S, V) a feasible solution in (**ACOPF**) and we define $W \in \mathbb{H}(\mathcal{E})$ as $W_{ba} = V_b V_a^*$ for any $(b, a) \in \mathcal{E}$. For any $b \in \mathcal{B}$, we make the following observations:

- Since $\underline{v}_b \leq |V_b| \leq \overline{v}_b$, the inequalities $\underline{v}_b^2 \leq |V_b|^2 \leq \overline{v}_b^2$ and $\underline{v}_b^2 \leq W_{bb} \leq \overline{v}_b^2$ hold.

- Since $\sum_{g\in\mathcal{G}_b} S_g - S_b^{\mathsf{d}} = \left\langle M_b, VV^{\mathsf{H}} \right\rangle$ and since $M_b \in \mathbb{H}_n(\mathcal{E})$, we deduce by substitution that $\sum_{g\in\mathcal{G}_b} S_g - S_b^{\mathsf{d}} = \langle M_b, W\rangle$.

Similarly by direct substitution, we deduce that $|(Y_{ba}^{\mathsf{ff}})^* W_{bb} + (Y_{ba}^{\mathsf{ft}})^* W_{ba}| \le \overline{S}_{ba}$ for all $(b,a) \in \mathcal{L}$ and $|(Y_{ab}^{\mathsf{tt}})^* W_{bb} + (Y_{ab}^{\mathsf{tf}})^* W_{ba}| \le \overline{S}_{ba}$ for all $(b,a) \in \mathcal{L}^R$. For any $(b,a) \in \mathcal{L} \cup \mathcal{L}^R$, since $\angle W_{ba} = \angle V_b V_a^* = \angle V_b - \angle V_a$, we have $\underline{\theta}_{ba} \le \angle W_{ba} \le \overline{\theta}_{ba}$. Using that $\underline{\theta}_{ba}, \overline{\theta}_{ba} \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, we deduce that $\tan(\underline{\theta}_{ba})\mathsf{Re}(W_{ba}) \le \mathsf{Im}(W_{ba}) \le \tan(\overline{\theta}_{ba})\mathsf{Re}(W_{ba})$. To conclude about the feasibility of (S,W) in (**ACOPF**$_W$), we state that $W_{\mathcal{B}_k,\mathcal{B}_k} = (V_b V_a^*)_{(b,a)\in\mathcal{B}_k^2} \succeq 0$ for all $k \in \mathcal{T}$, and that $|W_{ba}|^2 = |V_b|^2|V_a|^2 = W_{bb}W_{aa}$ for all $(b,a) \in \mathcal{E}$.

Conversely, we consider any (S,W) feasible in (**ACOPF**$_W$). Since $W_{\mathcal{B}_k,\mathcal{B}_k} \succeq 0$ and $|W_{ba}|^2 = W_{bb}W_{aa}$ for all $(b,a) \in \mathcal{B}_k^2$, we can apply [48, Proposition 6] to state that $\mathsf{rank}\, W_{\mathcal{B}_k,\mathcal{B}_k} = 1$ for all $k \in \mathcal{T}$. By induction on the tree decomposition $\mathcal{T}$, we prove that there exists $V \in \mathbb{C}^{\mathcal{B}}$ such that $W_{ba} = V_b(V_a)^*$ for all $(b,a) \in \mathcal{E}$. The case $|\mathcal{T}| = 1$ is trivial, since any rank-one PSD matrix W can be written as $W = VV^{\mathsf{H}}$. We now assume that the induction hypothesis is true for any graph with a tree decomposition of size less or equal than $p \in \mathbb{N}^*$, and we consider a graph $\mathcal{N}$ with a tree decomposition $\mathcal{T}$ of size $p+1$. We consider a leaf k of $\mathcal{T}$, $\mathcal{B}_k$ the corresponding clique, $\tilde{\mathcal{B}} = \cup_{\ell\neq k}\mathcal{B}_\ell$ the union of the other cliques, and $\mathcal{C}_k = \mathcal{B}_k \setminus \tilde{\mathcal{B}}$. By property of a tree decomposition, since k is a leaf of $\mathcal{T}$, $\mathcal{T} \setminus \{k\}$ is a tree decomposition of the graph $(\tilde{\mathcal{B}}, \tilde{\mathcal{E}})$, where $\tilde{\mathcal{E}}$ denotes the edges in $\mathcal{E}$ that are not adjacent to $\mathcal{C}_k$. Applying the induction hypothesis, since $\mathcal{T}\setminus\{k\}$ has size p, there exists a complex vector $V \in \mathbb{C}^{\tilde{\mathcal{B}}}$ such that $W_{ba} = V_b V_a^*$ for all $(b,a) \in \tilde{\mathcal{E}}$. Additionally, since $W_{\mathcal{B}_k,\mathcal{B}_k}$ is a rank-one PSD matrix, there exists $U \in \mathbb{C}^{\mathcal{B}_k}$ such that $W_{ba} = U_b U_a^*$ for all $(b,a) \in \mathcal{B}_k \times \mathcal{B}_k$. For all $b \in \mathcal{B}_k \setminus \mathcal{C}_k$, $|V_b|^2 = W_{bb} = |U_b|^2$, since $b \in \tilde{\mathcal{B}} \cap \mathcal{B}_k$. Hence, $|V_b| = |U_b|$ by nonnegativity of the module. Moreover, for all $(b,a) \in (\mathcal{B}_k \setminus \mathcal{C}_k)^2$, $\angle V_b - \angle V_a = \angle W_{ba} = \angle U_b - \angle U_a$, and hence, $\angle V_b - \angle U_b = \angle V_a - \angle U_a$. Defining $\mu = \angle V_b - \angle U_b$ for any $b \in \mathcal{B}_k \setminus \mathcal{C}_k$, we define $U' = e^{\mathbf{i}\mu}U$, which satisfies $U'_b = V_b$ for all $b \in \mathcal{B}_k \setminus \mathcal{C}_k$. Hence, the vector $V' \in \mathbb{C}^{\mathcal{B}}$ defined as $V'_b = U'_b$ if $b \in \mathcal{B}_k$ and $V'_b = V_b$ if $b \in \tilde{\mathcal{B}}$, is well-defined and satisfies $W_{ba} = V'_b(V'_a)^*$ for all $(b,a) \in \mathcal{E}$. By induction, this proves that there exists a vector $V \in \mathbb{C}^{\mathcal{B}}$ such that $W_{ba} = V_b V_a^*$ for all $(b,a) \in \mathcal{E}$. The feasibility of (S,V) in (**ACOPF**) follows by substituting W_{ba} by $V_b V_a^*$ in the constraints of (**ACOPF**$_W$). □

4.2 Strengthened conic programming relaxation

In the formulation (**ACOPF**$_W$), the constraints ($\star$) are the only nonconvex constraints. Removing them leads to the clique-based SDP relaxation [165, 176] (see Chapter 5). Instead of merely deleting constraints ($\star$), we investigate different ways of approximating these constraints, exploiting the potentially tight voltage magnitude and phase angle difference bounds, so as to strengthen the SDP relaxation.

4.2.1 Outer-approximation of the nonconvex constraints

For all $b \in \mathcal{B}$, we introduce a variable $L_b \in [\underline{v}_b, \overline{v}_b]$ that represents the voltage magnitude $|V_b|$. For all $(b,a) \in \mathcal{E}$, we introduce a variable $R_{ba} \in [\underline{v}_b\underline{v}_a, \overline{v}_b\overline{v}_a]$ that stands for $|V_b||V_a|$ and is subject to

$$R_{ba} \geq \underline{v}_b L_a + \underline{v}_a L_b - \underline{v}_b\underline{v}_a \qquad R_{ba} \geq \overline{v}_b L_a + \overline{v}_a L_b - \overline{v}_b\overline{v}_a \tag{4.1}$$

$$R_{ba} \leq \overline{v}_b L_a + \underline{v}_a L_b - \underline{v}_a\overline{v}_b \qquad R_{ba} \leq \overline{v}_a L_b + \underline{v}_b L_a - \overline{v}_a\underline{v}_b. \tag{4.2}$$

For all $b \in \mathcal{B}$, we also define the following constraints

$$L_b^2 \leq R_{bb} \qquad R_{bb} = W_{bb} \tag{4.3}$$

$$R_{bb} + \underline{v}_b\overline{v}_b \leq (\underline{v}_b + \overline{v}_b)L_b. \tag{4.4}$$

Whereas constraints (4.1)-(4.4) approximate the equality $R_{ba}^2 = W_{bb}W_{aa}$, we also need to approximate $|W_{ba}| = R_{ba}$. For this purpose, we impose for all $(b,a) \in \mathcal{E}$,

$$|W_{ba}| \leq R_{ba}. \tag{4.5}$$

For all $(b,a) \in \mathcal{E}\backslash(\mathcal{L}\cup\mathcal{L}^R)$, we define $\underline{\theta}_{ba} = -2\pi$ and $\overline{\theta}_{ba} = 2\pi$. In fact, we present in Section 4.2.3 how these phase angle difference bounds may be tightened based on a shortest path algorithm. Then, we can define the angles $\omega_{ba} = \frac{\underline{\theta}_{ba}+\overline{\theta}_{ba}}{2}$ and $\delta_{ba} = \frac{\overline{\theta}_{ba}-\underline{\theta}_{ba}}{2}$ for any $(b,a) \in \mathcal{E}$. With this notation, the following constraints are valid for any $(b,a) \in \mathcal{E}$ such that $\delta_{ba} \leq \frac{\pi}{2}$:

$$\cos(\omega_{ba})\mathsf{Re}(W_{ba}) + \sin(\omega_{ba})\mathsf{Im}(W_{ba}) \geq R_{ba}\cos(\delta_{ba}). \tag{4.6}$$

Constraints (4.5)-(4.6) are illustrated in Figure 4.1. Finally, for every $k \in \mathcal{T}$, we require that

$$R_{\mathcal{B}_k\mathcal{B}_k} = (R_{\mathcal{B}_k\mathcal{B}_k})^{\mathsf{H}} \qquad \begin{pmatrix} 1 & L_{\mathcal{B}_k}^{\mathsf{H}} \\ L_{\mathcal{B}_k} & R_{\mathcal{B}_k\mathcal{B}_k} \end{pmatrix} \succeq 0, \tag{4.7}$$

where $R_{\mathcal{B}_k\mathcal{B}_k}$ denotes the matrix $(R_{ba})_{(b,a)\in\mathcal{B}_k^2}$ and $L_{\mathcal{B}_k}$ denotes the vector $(L_b)_{b\in\mathcal{B}_k}$. Adding the decision vectors $L \in \mathbb{R}^{\mathcal{B}}$ and $R \in \mathbb{R}^{\mathcal{E}}$ to the problem (**ACOPF**$_W$) and replacing constraints $(\star)$ by constraints (4.1)-(4.7), we obtain a conic programming problem, that we denote (**ACOPF**$_C$).

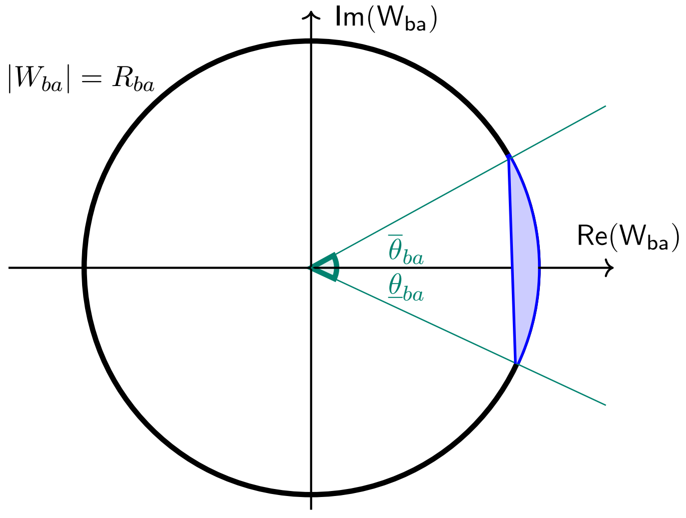


Figure 4.1: Convex approximation (in blue) of the equality $|W_{ba}| = R_{ba}$, performed by the constraints (4.5) and (4.6)

$$\left.\begin{array}{rl}
\min\limits_{\substack{S\in\mathbb{C}^{\mathcal{G}},\, W\in\mathbb{H}(\mathcal{E})\\ L\in\mathbb{R}^{\mathcal{B}}, R\in\mathbb{R}^{\mathcal{E}}}} & \sum\limits_{g\in\mathcal{G}} \Big(c_{0g} + c_{1g}\,\mathsf{Re}(S_g) + c_{2g}\,\mathsf{Re}(S_g)^2\Big) \\
\text{s.t. } \forall g \in \mathcal{G} & \underline{s}_g \le S_g \le \overline{s}_g \\
\forall b \in \mathcal{B} & \underline{v}_b^2 \le W_{bb} \le \overline{v}_b^2 \\
\forall b \in \mathcal{B} & \sum\limits_{g\in\mathcal{G}_b} S_g - S_b^{\mathsf{d}} = \langle M_b, W\rangle \\
\forall (b,a) \in \mathcal{L} & |(Y_{ba}^{\mathsf{ff}})^* W_{bb} + (Y_{ba}^{\mathsf{ft}})^* W_{ba}| \le \overline{S}_{ba} \\
\forall (b,a) \in \mathcal{L}^R & |(Y_{ab}^{\mathsf{tt}})^* W_{bb} + (Y_{ab}^{\mathsf{tf}})^* W_{ba}| \le \overline{S}_{ba} \\
\forall (b,a) \in \mathcal{L}\cup\mathcal{L}^R & \tan(\underline{\theta}_{ba})\mathsf{Re}(W_{ba}) \le \mathsf{Im}(W_{ba}) \le \tan(\overline{\theta}_{ba})\mathsf{Re}(W_{ba}) \\
\forall k \in \mathcal{T} & W_{\mathcal{B}_k,\mathcal{B}_k} \succeq 0 \\
\forall (b,a) \in \mathcal{E} & (4.1) - (4.2) \\
\forall b \in \mathcal{B} & (4.3) - (4.4) \\
\forall (b,a) \in \mathcal{E}\colon \delta_{ba} \le \frac{\pi}{2} & (4.6) \\
\forall k \in \mathcal{T} & (4.7).
\end{array}\right\} \quad (\textbf{ACOPF}_C)$$

Proposition 4.1. *The conic programming problem* (**ACOPF**$_C$) *is a relaxation of* (**ACOPF**$_W$).

Proof. We prove the validity of the constraints (4.1)-(4.7). More specifically, we prove that for any couple $(S, W) \in \mathbb{C}^{\mathcal{G}} \times \mathbb{H}(\mathcal{E})$ feasible in (**ACOPF**$_W$), the quadruplet (S, W, L, R) is feasible in (**ACOPF**$_C$), where L and R are defined as $L_b = \sqrt{W_{bb}}$ and $R_{ba} = |W_{ba}|$ for all $(b, a) \in \mathcal{E}$. Since the objective function is the same in (**ACOPF**$_C$) and (**ACOPF**$_W$), this will prove that (**ACOPF**$_C$) is a relaxation of (**ACOPF**$_W$). Since $R_{ba} = L_b L_a$ and $(L_b, L_a) \in [\underline{v}_b, \overline{v}_b] \times [\underline{v}_a, \overline{v}_a]$, the triplet (R_{ba}, L_b, L_a) satisfies the McCormick inequalities [158] with respect to these bounds, i.e., constraints (4.1)-(4.2). Constraint (4.3) is satisfied since $W_{bb} \in \mathbb{R}$, as (S, W) is feasible in (**ACOPF**$_W$), yielding $R_{bb} = |W_{bb}| = W_{bb} = L_b^2$. Constraint (4.4) also being a McCormick constraint (for $b = a$), it is satisfied by (R_{bb}, L_b), as $R_{bb} = L_b^2$. Constraint (4.5) just follows from the definition of $R_{ba} = |W_{ba}|$. For any $(b, a) \in \mathcal{E}$, we define $\theta_{ba} = \angle W_{ba}$; considering the definition of ω_{ba} and δ_{ba}, we notice that $|\theta_{ba} - \omega_{ba}| \le \delta_{ba}$. For this reason, if $\delta_{ba} \le \frac{\pi}{2}$, we obtain $\cos(|\theta_{ba} - \omega_{ba}|) \ge \cos(\delta_{ba})$, as cos is decreasing over $[0, \frac{\pi}{2}]$. Using the parity of cos, and multiplying by $R_{ba} \ge 0$, we obtain $R_{ba} \cos(\theta_{ba} - \omega_{ba}) \ge R_{ba} \cos(\delta_{ba})$ Moreover, $R_{ba} \cos(\theta_{ba} - \omega_{ba}) = |W_{ba}|(\cos(\omega_{ba}) \cos(\theta_{ba}) + \sin(\omega_{ba}) \sin(\theta_{ba})) = \cos(\omega_{ba})\mathsf{Re}(W_{ba}) + \sin(\omega_{ba})\mathsf{Im}(W_{ba})$, explaining that (R_{ba}, W_{ba}) satisfies constraint (4.6), whenever $\delta_{ba} \le \frac{\pi}{2}$. Finally, constraint (4.7) just follows from the equalities $R_{ba} = |W_{ba}| = |W_{ab}| = R_{ab}$ and $\begin{pmatrix} 1 & L_{\mathcal{B}_k}^{\mathsf{H}} \\ L_{\mathcal{B}_k} & R_{\mathcal{B}_k\mathcal{B}_k} \end{pmatrix} = \begin{pmatrix} 1 \\ L_{\mathcal{B}_k} \end{pmatrix} \begin{pmatrix} 1 & L_{\mathcal{B}_k}^{\mathsf{H}} \end{pmatrix} \succeq 0.$ □

By construction, the relaxation (**ACOPF**$_C$) is tighter than the clique-based SDP relaxation, the value of which equals the value of the standard SDP relaxation [91], also known as rank relaxation. The following theorem shows how constraints (4.1)-(4.6) help having $|W_{ba}|^2 \approx W_{bb}W_{aa}$ when the voltage magnitude and phase angle difference intervals are small. We recall the notation $\Delta_b = \overline{v}_b - \underline{v}_b$ and that we assume $\Delta_b \le 1$ throughout the paper.

Theorem 4.2. *For any $(b, a) \in \mathcal{E}$, the following statements hold*

- *Under constraints (4.1)-(4.2), we have $|R_{ba}^2 - L_a^2 L_b^2| \le 9\Delta_b\Delta_a$.*
- *Under constraints (4.3)-(4.4), we have $|W_{bb}W_{aa} - L_b^2 L_a^2| \le (\Delta_b + \Delta_a)^2$.*
- *Under constraints (4.5)-(4.6), if $\delta_{ba} \le \frac{\pi}{2}$, we have $|\,|W_{ba}|^2 - R_{ba}^2| \le 16\delta_{ba}^2$.*

Therefore, if constraints (4.1)-(4.6) are satisfied and $\delta_{ba} \le \frac{\pi}{2}$, then $|\ |W_{ba}|^2 - W_{bb}W_{aa}| \le 9\Delta_b\Delta_a + (\Delta_b + \Delta_a)^2 + 16\delta_{ba}^2$.

Proof. First, we take any $(b, a) \in \mathcal{E}$ and we define a triplet (W, L, R) satisfying constraints (4.1)-(4.2). We define $a_1 = \underline{v}_b L_a + \underline{v}_a L_b - \underline{v}_b \underline{v}_a$ and we notice that $L_b L_a - a_1 = (L_b - \underline{v}_b)(L_a - \underline{v}_a) \in [0, \Delta_b\Delta_a]$, since $L_b - \underline{v}_b \in [0, \Delta_b]$ and $L_a - \underline{v}_a \in [0, \Delta_a]$. Hence, $a_1 \in [L_b L_a - \Delta_b\Delta_a, L_b L_a]$. Similarly, defining $a_2 = \overline{v}_b L_a + \overline{v}_a L_b - \overline{v}_b \overline{v}_a$, $a_3 = \overline{v}_b L_a + \underline{v}_a L_b - \underline{v}_a \overline{v}_b$ and $a_4 = \overline{v}_a L_b + \underline{v}_b L_a - \overline{v}_a \underline{v}_b$, we can prove that $a_2 \in [L_b L_a - \Delta_b\Delta_a, L_b L_a]$, $a_3 \in [L_b L_a, L_b L_a + \Delta_b\Delta_a]$ and $a_4 \in [L_b L_a, L_b L_a + \Delta_b\Delta_a]$. According to constraints (4.1)-(4.2), $R_{ba} \in [\max(a_1, a_2), \min(a_3, a_4)]$, which proves that $R_{ba} \in [L_b L_a - \Delta_b\Delta_a, L_b L_a + \Delta_b\Delta_a]$. We square the inequalities $0 \le R_{ba} \le L_b L_a + \Delta_b\Delta_a$ to obtain

$$R_{ba}^2 \le L_b^2 L_a^2 + \Delta_b\Delta_a(2L_b L_a + \Delta_b\Delta_a) \le L_b^2 L_a^2 + 9\Delta_b\Delta_a, \tag{4.8}$$

the last inequality following from $L_b \le \overline{v}_b \le 2$, $L_a \le \overline{v}_a \le 2$ and $0 \le \Delta_b\Delta_a \le 1$. Squaring the inequalities $0 \le L_b L_a \le R_{ba} + \Delta_b\Delta_a$, we deduce that $L_b^2 L_a^2 \le R_{ba}^2 + \Delta_b\Delta_a(2R_{ba} + \Delta_b\Delta_a) \le R_{ba}^2 + 9\Delta_b\Delta_a$ since $R_{ba} \le \overline{v}_b \overline{v}_a \le 4$. Consequently,

$$|R_{ba}^2 - L_a^2 L_b^2| \le 9\Delta_b\Delta_a. \tag{4.9}$$

Second, we take any triplet (W, L, R) satisfying constraints (4.3)-(4.4) for b and a. We notice that the maximum of the quadratic form $(\underline{v}_b + \overline{v}_b)X - X^2 - \underline{v}_b\overline{v}_b$ is attained for $X = \frac{\underline{v}_b+\overline{v}_b}{2}$ with value $\frac{(\underline{v}_b+\overline{v}_b)^2}{4} - \underline{v}_b\overline{v}_b = \frac{\Delta_b^2}{4}$. Hence, $(\underline{v}_b + \overline{v}_b)L_b - L_b^2 - \underline{v}_b\overline{v}_b \leq \frac{\Delta_b^2}{4}$. Constraint (4.4) yielding $R_{bb} + \underline{v}_b\overline{v}_b \leq (\underline{v}_b + \overline{v}_b)L_b$, we deduce that $R_{bb} - L_b^2 \leq \Delta_b^2/4$. As $R_{bb} \geq 0$, we have $0 \leq R_{bb} \leq L_b^2 + \Delta_b^2/4$. Applying the same reasoning for a, we have $0 \leq R_{aa} \leq L_a^2 + \Delta_a^2/4$. Multiplying both sets of inequalities together, we obtain

$$0 \leq R_{bb}R_{aa} \leq L_b^2L_a^2 + L_b^2\frac{\Delta_a^2}{4} + L_a^2\frac{\Delta_b^2}{4} + \frac{\Delta_b^2}{4}\frac{\Delta_a^2}{4} \leq L_b^2L_a^2 + \Delta_a^2 + \Delta_b^2 + 2\Delta_b\Delta_a \tag{4.10}$$

$$\leq L_b^2L_a^2 + (\Delta_b + \Delta_a)^2, \tag{4.11}$$

using that $L_b, L_a \in [0, 2]$ and $\Delta_b, \Delta_a \in [0, 1]$. As constraint (4.3) yields $L_b^2 \leq R_{bb}$ and $L_a^2 \leq R_{aa}$, we deduce that $L_b^2L_a^2 \leq R_{bb}R_{aa}$ and finally, since $R_{bb} = W_{bb}$ and $R_{aa} = W_{aa}$,

$$|W_{bb}W_{aa} - L_b^2L_a^2| \leq (\Delta_b + \Delta_a)^2. \tag{4.12}$$

Third, we take any triplet (W, L, R) satisfying constraints (4.5)-(4.6). We consider $(b, a) \in \mathcal{E}$ such that $\delta_{ba} \leq \frac{\pi}{2}$. We write W_{ba} as $|W_{ba}|e^{i\theta}$ and constraint (4.6), which is applicable since $\delta_{ba} \leq \frac{\pi}{2}$, yields $|W_{ba}|(\cos(\omega_{ba})\cos(\theta) + \sin(\omega_{ba})\sin(\theta)) \geq R_{ba}\cos(\delta_{ba})$. This may be written as $|W_{ba}|\cos(\omega_{ba} - \theta) \geq R_{ba}\cos(\delta_{ba})$. This implies that $|W_{ba}| \geq R_{ba}\cos(\delta_{ba})$, and thus $|W_{ba}|^2 \geq R_{ba}^2\cos(\delta_{ba})^2$. As $|W_{ba}|^2 \leq R_{ba}^2$, according to constraint (4.5), we have

$$0 \leq R_{ba}^2 - |W_{ba}|^2 \leq R_{ba}^2(1 - \cos(\delta_{ba})^2) = R_{ba}^2\sin(\delta_{ba})^2.$$

Using that $R_{ba} \leq 4$ and that $\sin(\delta_{ba})^2 \leq \delta_{ba}^2$, we obtain

$$|\,|W_{ba}|^2 - R_{ba}^2| \leq 16\delta_{ba}^2. \tag{4.13}$$

As a conclusion, for any triplet (W, L, R) satisfying constraints (4.1)-(4.6), we deduce from Eqs. (4.9), (4.12) and (4.13) that

$$|\,|W_{ba}|^2 - W_{bb}W_{aa}| \leq |\,|W_{ba}|^2 - R_{ba}^2| + |R_{ba}^2 - L_b^2L_a^2| + |L_b^2L_a^2 - W_{bb}W_{aa}| \tag{4.14}$$

$$\leq 9\Delta_b\Delta_a + (\Delta_b + \Delta_a)^2 + 16\delta_{ba}^2. \tag{4.15}$$

□

4.2.2 Connections to previous works

Previous works proposed valid inequalities to strengthen the SDP relaxation of the ACOPF problem [56, 122, 124]. In [56], the authors show that these valid inequalities are all dominated by the inequalities [56, (36a) and (36b)]. Using the parameter $v_b^\sigma = \underline{v}_b + \overline{v}_b$, the inequalities [56, (36a) and (36b)] read, with our notation,

$$v_b^\sigma v_a^\sigma\,(\cos(\omega_{ba})\mathsf{Re}(W_{ba}) + \sin(\omega_{ba})\mathsf{Im}(W_{ba})) - \overline{v}_a\cos(\delta_{ba})v_a^\sigma W_{bb} - \overline{v}_b\cos(\delta_{ba})v_b^\sigma W_{aa} \geq \overline{v}_b\overline{v}_a\cos(\delta_{ba})(\underline{v}_b\underline{v}_a - \overline{v}_b\overline{v}_a) \tag{†}$$

$$v_b^\sigma v_a^\sigma\,(\cos(\omega_{ba})\mathsf{Re}(W_{ba}) + \sin(\omega_{ba})\mathsf{Im}(W_{ba})) - \underline{v}_a\cos(\delta_{ba})v_a^\sigma W_{bb} - \underline{v}_b\cos(\delta_{ba})v_b^\sigma W_{aa} \geq -\underline{v}_b\underline{v}_a\cos(\delta_{ba})(\underline{v}_b\underline{v}_a - \overline{v}_b\overline{v}_a). \tag{‡}$$

The following proposition states that constraints (4.1)-(4.6), that we introduce here to strengthen the SDP relaxation, dominate Eqs. (†)-(‡).

Proposition 4.2. *For any* $(b,a) \in \mathcal{L} \cup \mathcal{L}^R$*, for any quadruplet* $(\mathsf{Re}(W_{ba}), \mathsf{Im}(W_{ba}), W_{bb}, W_{aa})$ *such that there exist* $L_b, L_a, R_{ba} \in \mathbb{R}_+$ *that satisfy* (4.1)-(4.6)*, then the quadruplet* $(\mathsf{Re}(W_{ba}),$ $\mathsf{Im}(W_{ba}),$ $W_{bb},$ $W_{aa})$ *satisfies* (†)-(‡).

Proof. We take any $(b,a) \in \mathcal{L} \cup \mathcal{L}^R$ and any quadruplet $(\mathsf{Re}(W_{ba}), \mathsf{Im}(W_{ba}), W_{bb}, W_{aa})$ such that there exists $L_b, L_a, R_{ba} \in \mathbb{R}_+$ such that constraints (4.1)-(4.6) are satisfied. Constraints (4.3)-(4.4) applied for b and a yields

$$v_b^\sigma L_b \geq W_{bb} + \overline{v}_b \underline{v}_b \tag{4.16}$$

$$v_a^\sigma L_a \geq W_{aa} + \overline{v}_a \underline{v}_a. \tag{4.17}$$

First, we combine Eqs. (4.16)-(4.17) with $R_{ba} \geq \overline{v}_a L_b + \overline{v}_b L_a - \overline{v}_b \overline{v}_a$ from constraint (4.1), that we multiply by $v_b^\sigma v_a^\sigma \geq 0$, to deduce that $v_b^\sigma v_a^\sigma R_{ba} \geq \overline{v}_a v_a^\sigma W_{bb} + \overline{v}_b v_b^\sigma W_{aa} + \overline{v}_a v_a^\sigma \overline{v}_b \underline{v}_b + \overline{v}_b v_b^\sigma \overline{v}_a \underline{v}_a - v_b^\sigma v_a^\sigma \overline{v}_b \overline{v}_a$ and, thus,

$$v_b^\sigma v_a^\sigma R_{ba} - \overline{v}_a v_a^\sigma W_{bb} - \overline{v}_b v_b^\sigma W_{aa} \geq \overline{v}_a v_a^\sigma \overline{v}_b \underline{v}_b + \overline{v}_b v_b^\sigma \overline{v}_a \underline{v}_a - v_b^\sigma v_a^\sigma \overline{v}_b \overline{v}_a = \overline{v}_b \overline{v}_a (\underline{v}_b \underline{v}_a - \overline{v}_b \overline{v}_a), \tag{4.18}$$

as $\overline{v}_a v_a^\sigma \overline{v}_b \underline{v}_b + \overline{v}_b v_b^\sigma \overline{v}_a \underline{v}_a - v_b^\sigma v_a^\sigma \overline{v}_b \overline{v}_a = \overline{v}_b \overline{v}_a (\underline{v}_b \underline{v}_a + \underline{v}_b \overline{v}_a + \underline{v}_b \underline{v}_a + \overline{v}_b \underline{v}_a - \underline{v}_b \underline{v}_a - \underline{v}_b \overline{v}_a - \overline{v}_b \underline{v}_a - \overline{v}_b \overline{v}_a) = \overline{v}_b \overline{v}_a (\underline{v}_b \underline{v}_a - \overline{v}_b \overline{v}_a)$. Multiplying Eq. (4.18) by $\cos(\delta_{ba}) \geq 0$, we have

$$v_b^\sigma v_a^\sigma \cos(\delta_{ba}) R_{ba} - \overline{v}_a \cos(\delta_{ba}) v_a^\sigma W_{bb} - \overline{v}_b \cos(\delta_{ba}) v_b^\sigma W_{aa} \geq \overline{v}_b \overline{v}_a \cos(\delta_{ba}) (\underline{v}_b \underline{v}_a - \overline{v}_b \overline{v}_a). \tag{4.19}$$

Multiplying constraint (4.6) by $v_b^\sigma v_a^\sigma \geq 0$ yields $v_b^\sigma v_a^\sigma (\cos(\omega_{ba}) \mathsf{Re}(W_{ba}) + \sin(\omega_{ba}) \mathsf{Im}(W_{ba})) \geq v_b^\sigma v_a^\sigma \cos(\delta_{ba}) R_{ba}$; combining this with (4.19), we deduce Eq. (†). We underline that constraint (4.6) is indeed applicable since $\delta_{ba} \leq \frac{\pi}{2}$, as $(b,a) \in \mathcal{L} \cap \mathcal{L}^R$ (see Table 4.1).
Second, we combine the Eqs. (4.16)-(4.17) with $R_{ba} \geq \underline{v}_a L_b + \underline{v}_b L_a - \underline{v}_b \underline{v}_a$ from constraint (4.1) that we multiply by $v_b^\sigma v_a^\sigma \geq 0$, to obtain $v_b^\sigma v_a^\sigma R_{ba} \geq \underline{v}_a v_a^\sigma W_{bb} + \underline{v}_b v_b^\sigma W_{aa} + \underline{v}_a v_a^\sigma \overline{v}_b \underline{v}_b + \underline{v}_b v_b^\sigma \overline{v}_a \underline{v}_a - v_b^\sigma v_a^\sigma \underline{v}_b \underline{v}_a$. As $\underline{v}_a v_a^\sigma \overline{v}_b \underline{v}_b + \underline{v}_b v_b^\sigma \overline{v}_a \underline{v}_a - v_b^\sigma v_a^\sigma \underline{v}_b \underline{v}_a = \underline{v}_b \underline{v}_a (\overline{v}_b \underline{v}_a + \overline{v}_b \overline{v}_a + \underline{v}_b \overline{v}_a + \overline{v}_b \overline{v}_a - \underline{v}_b \underline{v}_a - \underline{v}_b \overline{v}_a - \overline{v}_b \underline{v}_a - \overline{v}_b \overline{v}_a) = -\underline{v}_b \underline{v}_a (\underline{v}_b \underline{v}_a - \overline{v}_b \overline{v}_a)$, we deduce that

$$v_b^\sigma v_a^\sigma R_{ba} - \underline{v}_a v_a^\sigma W_{bb} - \underline{v}_b v_b^\sigma W_{aa} \geq -\underline{v}_b \underline{v}_a (\underline{v}_b \underline{v}_a - \overline{v}_b \overline{v}_a). \tag{4.20}$$

Multiplying Eq. (4.20) by $\cos(\delta_{ba}) \geq 0$, we obtain

$$v_b^\sigma v_a^\sigma \cos(\delta_{ba}) R_{ba} - \underline{v}_a \cos(\delta_{ba}) v_a^\sigma W_{bb} - \underline{v}_b \cos(\delta_{ba}) v_b^\sigma W_{aa} \geq -\underline{v}_b \underline{v}_a \cos(\delta_{ba}) (\underline{v}_b \underline{v}_a - \overline{v}_b \overline{v}_a). \tag{4.21}$$

Multiplying constraint (4.6) by $v_b^\sigma v_a^\sigma \geq 0$ yields $v_b^\sigma v_a^\sigma (\cos(\omega_{ba}) \mathsf{Re}(W_{ba}) + \sin(\omega_{ba}) \mathsf{Im}(W_{ba})) \geq v_b^\sigma v_a^\sigma \cos(\delta_{ba}) R_{ba}$; combining this with (4.21), we deduce Eq. (‡). □

The advantage of constraints (4.1)-(4.6), compared to Eqs. (†)-(‡), is to enforce a coupling between the convex envelopes of the quadruplets $(\mathsf{Re}(W_{ba}), \mathsf{Im}(W_{ba}), W_{bb}, W_{aa})$ involving a same index b. This coupling is realized by the additional decision vectors L and R. In Appendix C, we present an illustrative example of two quadruplets $(\mathsf{Re}(W_{ba}), \mathsf{Im}(W_{ba}), W_{bb}, W_{aa})$ and $(\mathsf{Re}(W_{bc}), \mathsf{Im}(W_{bc}), W_{bb}, W_{cc})$ satisfying Eqs. (†)-(‡) introduced in [56], but for which there is no vector L and R such that constraints (4.1)-(4.6) are satisfied. In this respect, we can state that constraints (4.1)-(4.6) strictly dominate Eqs. (†)-(‡).

4.2.3 Bound tightening procedures

We use three bound tightening procedures to reduce the interval lengths Δ_b and δ_{ba} and, thus, reduce the error bound in Theorem 4.2.

Feasibility-based bound tightening (FBBT) The power flow limit for the line $(b,a) \in \mathcal{L}$ implicitly restricts the phase $\angle V_b V_a^*$ and, consequently, can help reduce the length of the interval $[\underline{\theta}_{ba}, \overline{\theta}_{ba}]$. Dividing the inequality $|(Y^{\mathsf{ft}}_{ba})^* V_b V_a^* + (Y^{\mathsf{ff}}_{ba})^* |V_b|^2| \leq \overline{S}_{ba}$ by $|Y^{\mathsf{ft}}_{ba} V_b V_a|$, we deduce that $|\frac{V_b V_a^*}{|V_a||V_b|} - z\frac{|V_b|}{|V_a|}| \leq R$, where $z = \frac{(Y^{\mathsf{ff}}_{ba})^*}{(Y^{\mathsf{ft}}_{ba})^*}$ and $\rho = \frac{\overline{S}_{ba}}{|Y^{\mathsf{ft}}_{ba} V_b V_a|}$. We notice that $u = \frac{V_b V_a^*}{|V_a||V_b|}$ is a unit complex number and has a nonnegative real part since $\angle V_b - \angle V_a \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. Representing the ratio $\frac{|V_b|}{|V_a|}$ by a variable λ, we can formulate the following convex optimization problem

$$\begin{cases} \max_{u,\lambda} & \mathsf{Im}(u) \\ \text{s.t.} & |u - z\lambda| \leq \rho \\ & \mathsf{Re}(u) \geq 0 \\ & |u| \leq 1 \\ & u \in \mathbb{C}, \lambda \in [\frac{\underline{v}_b}{\overline{v}_a}, \frac{\overline{v}_b}{\underline{v}_a}]. \end{cases} \tag{4.22}$$

Denoting by $\overline{h}$ its value, we deduce that $\arcsin(\overline{h})$ is an upper-bound on $\angle V_b - \angle V_a$. Hence, we can set $\overline{\theta}_{ba} \leftarrow \min(\overline{\theta}_{ba}, \arcsin(\overline{h}))$ without changing the value of (**ACOPF**). If we minimize $\mathsf{Im}(u)$ under the same constraints to get a value $\underline{h}$, we can set $\underline{\theta}_{ba} \leftarrow \max(\underline{\theta}_{ba}, \arcsin(\underline{h}))$.

Similarly for any $(b,a) \in \mathcal{L}^R$, leveraging the inequality $|(Y^{\mathsf{tf}}_{ab})^* V_b V_a^* + (Y^{\mathsf{tt}}_{ab})^* |V_b|^2| \leq \overline{S}_{ba}$, we use the same procedure with $z = \frac{(Y^{\mathsf{tt}}_{ab})^*}{(Y^{\mathsf{tf}}_{ab})^*}$ and $\rho = \frac{S_{ba}}{|Y^{\mathsf{tf}}_{ab} V_b V_a|}$ to tighten $\underline{\theta}_{ba}$ and $\overline{\theta}_{ba}$. This type of bound tightening is cheap, since one only solves a 2-variable convex optimization problem for each bound.

Optimization-based bound tightening (OBBT) We also apply a OBBT procedure to the relaxation (**ACOPF$_C$**). We use any NLP algorithm to find an ACOPF feasible solution. With the corresponding upper-bound denoted $\overline{\mathsf{obj}}$, we add the constraint $\sum_{g\in\mathcal{G}} c_{1g} \mathsf{Re}(S_g) + c_{2g} \mathsf{Re}(S_g)^2 \leq \overline{\mathsf{obj}}$ to the problem (**ACOPF$_C$**). We denote by $\mathcal{F}$ the resulting convex feasible set. Then, we update the following bounds:

- For the **voltage magnitude** at bus $b \in \mathcal{B}$, we set

$$\overline{v}_b \leftarrow \max_{(S,W,L,R)\in\mathcal{F}} L_b \tag{4.23}$$

$$\underline{v}_b \leftarrow \min_{(S,W,L,R)\in\mathcal{F}} L_b. \tag{4.24}$$

- For the **angle difference** on line $(b,a) \in \mathcal{L}$, we compute $\overline{h}_{ba} = \max_{(S,W,L,R)\in\mathcal{F}} \mathsf{Im}(W_{ba})$ and $\underline{h}_{ba} = \min_{(S,W,L,R)\in\mathcal{F}} \mathsf{Im}(W_{ba})$ and set

$$\overline{\theta}_{ba} \leftarrow \min(\overline{\theta}_{ba}, \arcsin(\max(\frac{\overline{h}_{ba}}{\overline{v}_b\overline{v}_a}, \frac{\overline{h}_{ba}}{\underline{v}_b\underline{v}_a}))) \tag{4.25}$$

$$\underline{\theta}_{ba} \leftarrow \max(\underline{\theta}_{ba}, \arcsin(\min(\frac{\underline{h}_{ba}}{\overline{v}_b\overline{v}_a}, \frac{\underline{h}_{ba}}{\underline{v}_b\underline{v}_a}))). \tag{4.26}$$

Shortest path algorithm to tighten phase angle difference bounds Through FBBT and OBBT, we may individually improve the bounds $\underline{\theta}_{ba}$ and $\overline{\theta}_{ba}$ for any $(b,a) \in \mathcal{E}$. To propagate the reduction of the phase angle difference domains, we apply a shortest path algorithm. Indeed we notice that, for any $(b_0, b_t) \in \mathcal{B} \times \mathcal{B}$, for any path $b_0, b_1, \ldots, b_t$ in the graph $(\mathcal{B}, \mathcal{E})$, for any feasible solution (S,V) in (**ACOPF**), we have $\angle V_{b_t} - \angle V_{b_0} = \sum_{s=0}^{t-1} \angle V_{b_{s+1}} - \angle V_{b_s} \leq \sum_{s=0}^{t-1} \overline{\theta}_{b_{s+1}b_s}$. The shortest path between b_0 and b_t in the directed weighted graph $(\mathcal{B}, \mathcal{E}, \overline{\theta})$ helps finding the lowest sum $\sum_{s=0}^{t-1} \overline{\theta}_{b_{s+1}b_s}$ to update $\overline{\theta}_{b_t b_0}$. Symmetrically, we have that $\angle V_{b_t} - \angle V_{b_0} \geq \sum_{s=0}^{t-1} \underline{\theta}_{b_{s+1}b_s}$. The shortest path between b_0 and b_t in the directed weighted graph $(\mathcal{B}, \mathcal{E}, -\underline{\theta})$ helps improving the lower bound on $\angle V_{b_t} - \angle V_{b_0}$ to update $\underline{\theta}_{b_t b_0}$. To compute shortest paths, we apply the Floyd-Warshall algorithm [59], which fits the context of a weighted directed graph, with weights of unspecified sign. Should the Floyd-Warshall algorithm find a cycle with negative weight in the directed weighted graph $(\mathcal{B}, \mathcal{E}, \overline{\theta})$, it would certify the infeasibility of (**ACOPF**), since it would give a path $b_0, b_1, \ldots, b_t$ with $b_t = b_0$ and $0 = \angle V_{b_t} - \angle V_{b_0} = \sum_{s=0}^{t-1} \angle V_{b_{s+1}} - \angle V_{b_s} \leq \sum_{s=0}^{t-1} \overline{\theta}_{b_{s+1}b_s} < 0$. Similarly, finding a cycle of negative weight in $(\mathcal{B}, \mathcal{E}, -\underline{\theta})$ certifies the infeasibility of (**ACOPF**).

4.3 A MILP-based scheme for global optimization

In this section, leveraging the conic relaxation (**ACOPF**$_C$), we generate a sequence of MILP problems, the values of which converge to the ACOPF value. We achieve this using binary variables to encode piecewise linear approximations of the feasible sets. The resulting Mixed-integer conic programming problems are then approximated as MILP. From a semi-infinite programming perspective, this approximation is an adaptive discretization approach, as presented in the Introduction. We start by showing that we can "discretize" the conic constraints (in the sense of semi-infinite programming) to obtain a LP relaxation as tight as (**ACOPF**$_C$).

4.3.1 Linear programming outer-approximations

The disadvantage of solving the conic problem (**ACOPF**$_C$) is its computational cost, that is higher, due to SDP constraints, than that of solving a LP, SOCP, or a convex QCQP relaxation. Hence, it may not be computationally efficient to solve such a relaxation at every node of an exploration tree. The idea of our approach is to solve the relaxation (**ACOPF**$_C$) at the root node only, and use it to generate a LP relaxation with same value. We denote by $x \in \mathbb{R}^\Xi$ the decision vector $(\mathsf{Re}(S), \mathsf{Im}(S), \mathsf{Re}(W), \mathsf{Im}(W), L, R)$. The problem (**ACOPF**$_C$) may be seen as

$$\left.\begin{array}{ll} \min\limits_{x\in\mathcal{P}} & \phi_0(x) \\ \forall j \in [\![1,r]\!] & \phi_j(x) \leq 0, \end{array}\right\} \qquad (\mathbf{ACOPF}_C\text{'})$$

with $\mathcal{P} \subset \mathbb{R}^\Xi$ being a polytope and $\phi_0(x), \phi_1(x), \dots, \phi_r(x)$ continuous and convex functions. In Appendix C, we detail this polytope, the functions $\phi_j(x)$, and show that they share a common structure: for any $j \in [\![0,r]\!]$, there exists an affine application $\pi_j : \mathbb{R}^\Xi \mapsto \mathbb{R}^{p_j}$ and a compact and convex set $\mathcal{Y}_j \subset \mathbb{R}^{p_j}$ such that for all $x \in \mathcal{P}$, $\phi_j(x) = \max_{y\in\mathcal{Y}_j} y^\top \pi_j(x)$. Hence the constraint $\phi_j(x) \leq 0$ can be cast as the following semi-infinite programming constraint: $\forall y \in \mathcal{Y}_j, y^\top\pi_j(x) \leq 0$. In a semi-infinite discretization approach, for any family of finite subsets $\check{\mathcal{Y}}_j \subset \mathcal{Y}_j$, we obtain a LP relaxation of (**ACOPF**$_C$)

$$\left.\begin{array}{ll} \min\limits_{(x,\lambda)\in\mathcal{P}\times\mathbb{R}} & \lambda \\ \forall y \in \check{\mathcal{Y}}_0 & y^\top\pi_0(x) \leq \lambda \\ \forall j \in [\![1,r]\!], \forall y \in \check{\mathcal{Y}}_j & y^\top\pi_j(x) \leq 0. \end{array}\right\} \qquad (\mathbf{ACOPF}_L)$$

We show that based on a primal-dual solution of (**ACOPF**$_C$), we can compute finite sets $\check{\mathcal{Y}}_0, \dots, \check{\mathcal{Y}}_r$ such that $\mathsf{val}(\mathbf{ACOPF}_C) = \mathsf{val}(\mathbf{ACOPF}_L)$. For $j \in [\![1,r]\!]$ we define $\mathcal{K}_j$ as the convex cone generated by $\mathcal{Y}_j$. We define $\mathcal{K}_0$ as the convex cone generated by $\{1\} \times \mathcal{Y}_0$. We also define $\underline{\lambda}$ and $\overline{\lambda}$ as *a priori* lower and upper bounds on the value of (**ACOPF**$_C$), that may be very rough estimates. We introduce a Lagrangian function L for the conic program (**ACOPF**$_C$), defined for any $(x,\lambda) \in \mathcal{P} \times [\underline{\lambda}, \overline{\lambda}], \mathbf{z} = (z_0, z_1, \dots, z_r) \in \mathcal{K}_0 \times \mathcal{K}_1 \times \dots \times \mathcal{K}_r$ as $L(x,\lambda,\mathbf{z}) = \lambda + z_0^\top \begin{pmatrix} -\lambda \\ \pi_0(x) \end{pmatrix} + \sum_{j=1}^r z_j^\top \pi_j(x)$. With this definition, we see that the formulation (**ACOPF**$_C$) is the min-max problem $\inf_{x\in\mathcal{P},\lambda\in[\underline{\lambda},\overline{\lambda}]} \; \sup_{\mathbf{z}\in\mathcal{K}_0\times\mathcal{K}_1\times\dots\times\mathcal{K}_r} L(x,\lambda,\mathbf{z})$. We define the concave function $D(\mathbf{z}) = \inf_{x\in\mathcal{P},\lambda\in[\underline{\lambda},\overline{\lambda}]} L(x,\lambda,\mathbf{z}) \in \mathbb{R} \cup \{-\infty\}$, and the dual optimization problem

$$\sup_{\mathbf{z}\in\mathcal{K}_0\times\mathcal{K}_1\times\dots\times\mathcal{K}_r} D(\mathbf{z}). \tag{4.27}$$

Proposition 4.3. *There is no duality gap between Problem (**ACOPF**$_C$) and Problem (4.27), i.e., they share the same value. Moreover, if Problem (4.27) has an optimal solution $z^* \in \mathcal{K}_0 \times \mathcal{K}_1 \times \dots \times \mathcal{K}_r$, written as $z^* = (\eta_0 v_0^*, \eta_1 y_1^*, \dots \eta_r y_r^*)$ with*

- $\eta_j \in \mathbb{R}_+$ *for all* $j \in [\![0, r]\!]$,
- $v_0^* = (1, y_0^*)$ *with* $y_0^* \in \mathcal{Y}_0$*, and* $y_j^* \in \mathcal{Y}_j$ *for all* $j \in [\![0, r]\!]$,

then the definition of the finite sets $\check{\mathcal{Y}}_j = \{y_j^*\}$ *yields a* LP *relaxation* ($\mathbf{ACOPF}_L$) *that satisfies* $\mathsf{val}(\mathbf{ACOPF}_L) = \mathsf{val}(\mathbf{ACOPF}_C)$.

Proof. We use a similar approach as in Section 1.1 for duality in convex semi-infinite programming. The absence of duality gap follows from the Sion min-max theorem [125], since

- The primal optimization set $\mathcal{P} \times [\underline{\lambda}, \overline{\lambda}]$ is convex and compact,
- The dual optimization set $\mathcal{K}_0 \times \mathcal{K}_1 \times \cdots \times \mathcal{K}_r$ is convex,
- L is continuous and convex with respect to (x, λ) for any $\mathbf{z} \in \mathcal{K}_0 \times \mathcal{K}_1 \times \cdots \times \mathcal{K}_r$, and,
- L is continuous and concave with respect to $\mathbf{z}$ for any $(x, \lambda) \in \mathcal{P} \times [\underline{\lambda}, \overline{\lambda}]$.

Then, we assume that Problem (4.27) has an optimal solution $\mathbf{z}^* \in \mathcal{K}_0 \times \mathcal{K}_1 \times \cdots \times \mathcal{K}_r$. Due to the absence of duality gap, we know that $D(\mathbf{z}^*) = \mathsf{val}(\mathbf{ACOPF}_C)$. Writing $\mathbf{z}^* = (\eta_0(1, y_0^*), \eta_1 y_1^*, \ldots \eta_r y_r^*)$ as indicated above, we define $\check{\mathcal{Y}}_j = \{y_j^*\} \subset \mathcal{Y}_j$ and $\check{\mathcal{K}}_j = \mathsf{cone}(\check{\mathcal{Y}}_j)$, for all $j \in [\![0, r]\!]$. With this definition, ($\mathbf{ACOPF}_L$) reads

$$\inf_{x \in \mathcal{P}, \lambda \in [\underline{\lambda}, \overline{\lambda}]} \quad \sup_{\mathbf{z} \in \check{\mathcal{K}}_0 \times \check{\mathcal{K}}_1 \times \cdots \times \check{\mathcal{K}}_r} \quad L(x, \lambda, \mathbf{z}),$$

and its dual problem is $\sup_{\mathbf{z} \in \check{\mathcal{K}}_0 \times \check{\mathcal{K}}_1 \times \cdots \times \check{\mathcal{K}}_r} D(\mathbf{u})$. As $z^* \in \check{\mathcal{K}}_0 \times \check{\mathcal{K}}_1 \times \cdots \times \check{\mathcal{K}}_r$, we can write by weak duality that $\mathsf{val}(\mathbf{ACOPF}_L) \geq D(z^*) = \mathsf{val}(\mathbf{ACOPF}_C)$. As ($\mathbf{ACOPF}_L$) is a relaxation of ($\mathbf{ACOPF}_C$), we conclude that $\mathsf{val}(\mathbf{ACOPF}_L) = \mathsf{val}(\mathbf{ACOPF}_C)$. $\square$

At the price of finding an optimal primal-dual solution (x^*, λ^*, z^*) of ($\mathbf{ACOPF}_C$)-(4.27), we can build a LP relaxation with the same value as the conic programming relaxation ($\mathbf{ACOPF}_C$). In practice, we obtain such a primal-dual solution for every tested instance.

4.3.2 Binary variables to encode piecewise linear constraints

To tighten the relaxations ($\mathbf{ACOPF}_C$) and ($\mathbf{ACOPF}_L$), we introduce binary variables to encode piecewise linear outer-approximation of the nonconvex feasible set of ($\mathbf{ACOPF}_W$).

Partitioning voltage magnitude intervals For any $b \in \mathcal{B}$, we may want to split the interval $[\underline{v}_b, \overline{v}_b]$ in sub-intervals. We introduce a tree $\mathcal{J}_b$ and pairs $(\underline{v}_{bj}, \overline{v}_{bj})$ such that $\underline{v}_{bj} \leq \overline{v}_{bj}$ for all $j \in \mathcal{J}_b$. For r being the root node of $\mathcal{J}_b$, we have $(\underline{v}_{br}, \overline{v}_{br}) = (\underline{v}_b, \overline{v}_b)$. Denoting $\mathcal{J}_b^+(i)$ the set of the child nodes of i, the partition $[\underline{v}_{bi}, \overline{v}_{bi}] = \bigcup_{j \in \mathcal{J}_b^+(i)} [\underline{v}_{bj}, \overline{v}_{bj}]$ holds for any $i \in \mathcal{J}_b$. For any $j \in \mathcal{J}_b$, we introduce a variable $\alpha_{bj} \in \{0, 1\}$. To encode the equivalence $(\alpha_{bj} = 1) \iff (L_b \in [\underline{v}_{bj}, \overline{v}_{bj}])$ for any $j \in \mathcal{J}_b$, we impose $\alpha_{br} = 1$, and for any $j \in \mathcal{J}_b$

$$\underline{v}_{bj} \alpha_{bj} + (1 - \alpha_{bj}) \underline{v}_b \leq L_b \leq \overline{v}_{bj} \alpha_{bj} + (1 - \alpha_{bj}) \overline{v}_b, \tag{4.28}$$

and for any $i \in \mathcal{J}_b$ such that $\mathcal{J}_b^+(i) \neq \emptyset$,

$$\sum_{j \in \mathcal{J}_b^+(i)} \alpha_{bj} = \alpha_{bi}. \tag{4.29}$$

Moreover we add the following constraint for every $j \in \mathcal{J}_b$,

$$R_{bb} + \underline{v}_{bj}\overline{v}_{bj} \leq (\underline{v}_{bj} + \overline{v}_{bj})L_b + (\overline{v}_b^2 + \underline{v}_{bj}\overline{v}_{bj})(1 - \alpha_{bj}). \tag{4.30}$$

For every $j \in \mathcal{J}_b$ and for all $a \in \mathcal{B}$ such that $(b, a) \in \mathcal{E}$, we add the inequalities

$$R_{ba} \geq \underline{v}_{bj}L_a + \underline{v}_a L_b - \underline{v}_{bj}\underline{v}_a + \overline{v}_b\overline{v}_a(\alpha_{bj} - 1) \qquad R_{ba} \geq \overline{v}_{bj}L_a + \overline{v}_a L_b - \overline{v}_{bj}\overline{v}_a + \overline{v}_b\overline{v}_a(\alpha_{bj} - 1) \tag{4.31}$$

$$R_{ba} \leq \overline{v}_{bj}L_a + \underline{v}_a L_b - \underline{v}_a\overline{v}_{bj} + \overline{v}_b\overline{v}_a(1 - \alpha_{bj}) \qquad R_{ba} \leq \overline{v}_a L_b + \underline{v}_{bj}L_a - \overline{v}_a\underline{v}_{bj} + \overline{v}_b\overline{v}_a(1 - \alpha_{bj}). \tag{4.32}$$

Partitioning phase angle difference intervals For any $(b, a) \in \mathcal{E}$, we may want to split the interval $[\underline{\theta}_{ba}, \overline{\theta}_{ba}]$ in subintervals. We introduce a tree $\mathcal{J}_{ba}$ and pairs $(\underline{\theta}_{baj}, \overline{\theta}_{baj})$ such that $\underline{\theta}_{baj} \leq \overline{\theta}_{baj}$ for all $j \in \mathcal{J}_{ba}$. For r being the root node of $\mathcal{J}_{ba}$, we have $(\underline{\theta}_{bar}, \overline{\theta}_{bar}) = (\underline{\theta}_{ba}, \overline{\theta}_{ba})$. Denoting $\mathcal{J}_{ba}^+(i)$ the set of child nodes of i, the partition $[\underline{\theta}_{bai}, \overline{\theta}_{bai}] = \bigcup_{j \in \mathcal{J}_{ba}^+(i)} [\underline{\theta}_{baj}, \overline{\theta}_{baj}]$ holds for any $i \in \mathcal{J}_{ba}$. For $j \in \mathcal{J}_{ba}$, we introduce a variable $\beta_{baj} \in \{0, 1\}$. To encode the equivalence $(\beta_{baj} = 1) \iff (\angle W_{ba} \in [\underline{\theta}_{baj}, \overline{\theta}_{baj}])$, we impose $\beta_{bar} = 1$ and for any $j \in \mathcal{J}_{ba}$

$$\tan(\underline{\theta}_{ba})\mathsf{Re}(W_{ba}) + (\beta_{baj} - 1)\overline{v}_b\overline{v}_a \leq \mathsf{Im}(W_{ba}) \leq \tan(\overline{\theta}_{ba})\mathsf{Re}(W_{ba}) + (1 - \beta_{baj})\overline{v}_b\overline{v}_a, \tag{4.33}$$

and for any $i \in \mathcal{J}_{ba}$ such that $\mathcal{J}_{ba}^+(i) \neq \emptyset$

$$\sum_{j \in \mathcal{J}_{ba}^+(i)} \beta_{baj} = \beta_{bai}. \tag{4.34}$$

Moreover, for all $j \in \mathcal{J}_{ba}$, we define the angles $\omega_{baj} = \frac{\underline{\theta}_{baj} + \overline{\theta}_{baj}}{2}$ and $\delta_{baj} = \frac{\overline{\theta}_{baj} - \underline{\theta}_{baj}}{2}$, and if $\delta_{baj} \leq \frac{\pi}{2}$, we impose

$$\cos(\omega_{baj})\mathsf{Re}(W_{ba}) + \sin(\omega_{baj})\mathsf{Im}(W_{ba}) \geq R_{ba}\cos(\delta_{baj}) + (\beta_{baj} - 1)\overline{v}_b\overline{v}_a. \tag{4.35}$$

For fixed sets $\mathcal{J}_b, \mathcal{J}_{ba}$ of binary variables α_{bj}, β_{baj}, the relaxation obtained by adding these binary variables and the corresponding constraints (4.28)-(4.35) to the conic problem (**ACOPF**$_C$) (resp. the LP problem (**ACOPF**$_L$)) is a Mixed-integer conic programming problem (resp. a MILP problem).

4.3.3 Updating the partitions of the intervals

During the algorithm presented in section 4.3.4, the partitions of the intervals $[\underline{v}_b, \overline{v}_b]$ and $[\underline{\theta}_{ba}, \overline{\theta}_{ba}]$ are dynamically updated. The partition trees are all initialized as single-node graphs,

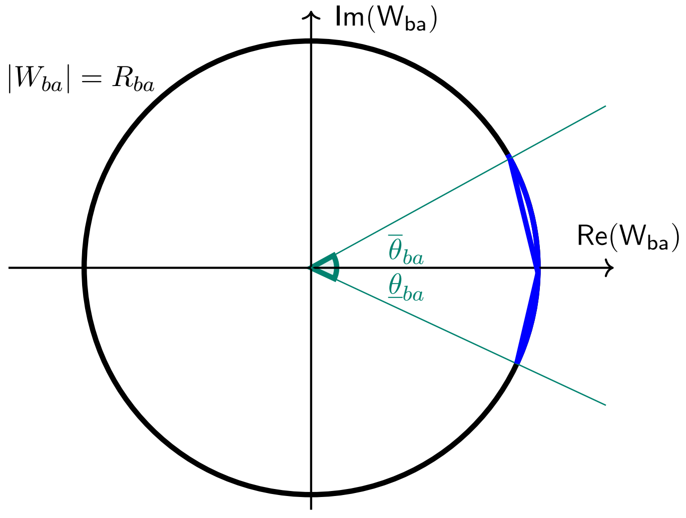


Figure 4.2: Piecewise convex approximation (in blue) of the equality $|W_{ba}| = R_{ba}$, performed by the constraints (4.33)-(4.35) (case $|\mathcal{J}_{ba}| = 2$)

and are then updated over the course of the algorithm. We discuss now how these trees are updated, at any iteration t of the algorithm, where the current iterate is (S^t, W^t, L^t, R^t).

For a given bus $b \in \mathcal{B}$, we update the partition tree $\mathcal{J}_b$ by selecting the active leaf j, i.e. the only leaf j of $\mathcal{J}_b$ such that $L_b^t \in [\underline{v}_{bj}, \overline{v}_{bj}]$. We create three new leaves j_1, j_2, j_3 in the tree, which are attached to node j, and we partition the interval $[\underline{v}_{bj}, \overline{v}_{bj}]$ as follows:

- We define $\underline{v}_{bj_1} = \underline{v}_{bj}$ and $\overline{v}_{bj_3} = \overline{v}_{bj}$;
- If $L_b^t \leq \frac{\underline{v}_{bj}+\overline{v}_{bj}}{2}$, we define $\overline{v}_{bj_1} = \underline{v}_{bj_2} = L_b^t$ and $\overline{v}_{bj_2} = \underline{v}_{bj_3} = \frac{L_b^t+\overline{v}_{bj}}{2}$;
- Else, we define $\overline{v}_{bj_1} = \underline{v}_{bj_2} = \frac{\underline{v}_{bj}+L_b^t}{2}$ and $\overline{v}_{bj_2} = \underline{v}_{bj_3} = L_b^t$.

For a given pair $(b, a) \in \mathcal{E}$, we update the partition tree $\mathcal{J}_{ba}$ by selecting the active leaf j, i.e. the only leaf j of $\mathcal{J}_{ba}$ such that $\angle W_{ba}^t \in [\underline{\theta}_{baj}, \overline{\theta}_{baj}]$. We create three new leaves j_1, j_2, j_3 in the tree, which are attached to node j, and we partition the interval $[\underline{\theta}_{baj}, \overline{\theta}_{baj}]$ as follows:

- We define $\underline{\theta}_{baj_1} = \underline{\theta}_{baj}$ and $\overline{\theta}_{baj_3} = \overline{\theta}_{baj}$;
- If $\angle W_{ba}^t \leq \frac{\underline{\theta}_{baj}+\overline{\theta}_{baj}}{2}$, we define $\overline{\theta}_{baj_1} = \underline{\theta}_{baj_2} = \angle W_{ba}^t$ and $\overline{\theta}_{baj_2} = \underline{\theta}_{baj_3} = \frac{\angle W_{ba}^t+\overline{\theta}_{baj}}{2}$;
- Else, we define $\overline{\theta}_{baj_1} = \underline{\theta}_{baj_2} = \frac{\underline{\theta}_{baj}+\angle W_{ba}^t}{2}$ and $\overline{\theta}_{baj_2} = \underline{\theta}_{baj_3} = \angle W_{ba}^t$.

The construction procedure of the trees $\mathcal{J}_b$ and $\mathcal{J}_{ba}$ guarantees that (i) $\overline{v}_{bj} - \underline{v}_{bj}$, the length of the interval associated with a node $j \in \mathcal{J}_b$ of depth $\Lambda(j)$, is less than $\frac{\Delta_b}{2^{\Lambda(j)}}$, (ii) the coefficient δ_{baj} associated with a node $j \in \mathcal{J}_{ba}$ is less than $\frac{\delta_{ba}}{2^{\Lambda(j)}}$.

Proposition 4.4. *We assume that the convex constraints* (4.1)-(4.6) *and the MILP constraints* (4.28)-(4.35) *are satisfied, but with a tolerance* $\epsilon \in [0, 1]$ *for the nonlinear constraints* (4.3) *and* (4.5)*. Then, for any nodes $j_b \in \mathcal{J}_b, j_a \in \mathcal{J}_a$ and $j_{ba} \in \mathcal{J}_{ba}$ that are active, i.e., such that $\alpha_{bj_b} = \alpha_{aj_a} = \beta_{baj_{ba}} = 1$, we have*

$$| (R_{ba})^2 - W_{bb}W_{aa}| \leq \frac{9\Delta_b\Delta_a}{2^{\max\{\Lambda(j_b),\Lambda(j_a)\}}} + \max\{9\epsilon, (\frac{\Delta_b}{2^{\Lambda(j_b)}} + \frac{\Delta_a}{2^{\Lambda(j_a)}})^2\}, \tag{4.36}$$

$$\left(\Lambda(j_{ba}) \geq \log_2(\frac{2\delta_{ba}}{\pi})\right) \implies \left(| |W_{ba}|^2 - (R_{ba})^2| \leq \max\{9\epsilon, \frac{16\delta_{ba}^2}{4^{\Lambda(j_{ba})}}\}\right). \tag{4.37}$$

We underline that the implication is still valid if $\delta_{ba} = 0$ and $\log_2(\frac{2\delta_{ba}}{\pi}) = -\infty$.

Proof. Since $\alpha_{bj_b} = 1$, constraints (4.31)-(4.32) yield constraints (4.1)-(4.2), but with $\underline{v}_b$, $\overline{v}_b$ and Δ_b replaced by $\underline{v}_{bj_b}$, $\overline{v}_{bj_b}$ and $\tilde{\Delta}_b = \overline{v}_{bj_b} - \underline{v}_{bj_b} \leq \frac{\Delta_b}{2^{\Lambda(j_b)}}$. Applying the first point of Theorem 4.2 with these parameters, we deduce that $|(R_{ba})^2 - L_a^2L_b^2| \leq 9\tilde{\Delta}_b\Delta_a \leq 9\frac{\Delta_b}{2^{\Lambda(j_b)}}\Delta_a$. Similarly, since $\alpha_{bj_a} = 1$ and since $R_{ba} = R_{ab}$, we also deduce from constraint (4.31)-(4.32) that $|(R_{ba})^2 - L_a^2L_b^2| \leq 9\frac{\Delta_a}{2^{\Lambda(j_a)}}\Delta_b$. Hence, we obtain

$$|(R_{ba})^2 - L_a^2L_b^2| \leq \frac{9\Delta_b\Delta_a}{2^{\max\{\Lambda(j_b),\Lambda(j_a)\}}}. \tag{4.38}$$

Since $\alpha_{bj_b} = 1$ (resp. $\alpha_{aj_a} = 1$), constraint (4.30) yields constraint (4.4) for b (resp. a) with $\underline{v}_b$, $\overline{v}_b$ and Δ_b (resp. $\underline{v}_a$, $\overline{v}_a$ and Δ_a) replaced by $\underline{v}_{bj_b}$, $\overline{v}_{bj_b}$ and $\tilde{\Delta}_b$ (resp. $\underline{v}_{aj_a}$, $\overline{v}_{bj_a}$ and $\tilde{\Delta}_a$). Applying Eq. (4.11) in the proof of Theorem 4.2, that follows only from constraint (4.4), we deduce that $R_{bb}R_{aa} - L_b^2L_a^2 \leq (\tilde{\Delta}_b + \tilde{\Delta}_a)^2 \leq (\frac{\Delta_b}{2^{\Lambda(j_b)}} + \frac{\Delta_a}{2^{\Lambda(j_a)}})^2$. Since constraint (4.3) is satisfied with tolerance $\epsilon \in [0, 1]$, we have that $L_b^2 \leq R_{bb} + \epsilon$ and $L_a^2 \leq R_{aa} + \epsilon$. Multiplying both inequalities, we deduce that $L_b^2L_a^2 \leq R_{bb}R_{aa} + \epsilon(R_{bb} + R_{aa}) + \epsilon^2 \leq R_{bb}R_{aa} + 9\epsilon$, since $R_{bb}, R_{aa} \in [0, 4]$ and $\epsilon^2 \leq \epsilon$. Hence, $|W_{bb}W_{aa} - L_b^2L_a^2| = |R_{bb}R_{aa} - L_b^2L_a^2| \leq \max\{9\epsilon, (\frac{\Delta_b}{2^{\Lambda(j_b)}} + \frac{\Delta_a}{2^{\Lambda(j_a)}})^2\}$. Combining this with Eq. (4.38), we deduce Eq. (4.36) due to the triangle inequality. We assume now that $\Lambda(j_{ba}) \geq \log_2(\frac{2\delta_{ba}}{\pi})$, implying that $\delta_{baj_{ba}} \leq \frac{\delta_{ba}}{2^{\Lambda(j_{ba})}} \leq \frac{\pi}{2}$. Since $\beta_{baj_{ba}} = 1$, constraint (4.35) yields constraint (4.6) with ω_{ba}, δ_{ba} replaced by $\omega_{baj_{ba}}, \delta_{baj_{ba}}$. Writing W_{ba} as $|W_{ba}|e^{i\theta}$ we thus have $|W_{ba}|(\cos(\omega_{baj_{ba}})\cos(\theta) + \sin(\omega_{baj_{ba}})\sin(\theta)) \geq R_{ba}\cos(\delta_{baj_{ba}})$. This also reads $|W_{ba}|\cos(\omega_{baj_{ba}} - \theta) \geq R_{ba}\cos(\delta_{baj_{ba}})$. This implies that $|W_{ba}| \geq R_{ba}\cos(\delta_{baj_{ba}})$, and thus $|W_{ba}|^2 \geq R_{ba}^2\cos(\delta_{baj_{ba}})^2$. Noticing that constraint (4.5) is satisfied with tolerance ϵ, we have that $|W_{ba}| \leq R_{ba} + \epsilon$ and $|W_{ba}|^2 \leq R_{ba}^2 + 2R_{ba}\epsilon + \epsilon^2 \leq R_{ba}^2 + 9\epsilon$. In summary, we have $-9\epsilon \leq R_{ba}^2 - |W_{ba}|^2 \leq R_{ba}^2(1 - \cos(\delta_{baj_{ba}})^2) \leq R_{ba}^2\sin(\delta_{baj_{ba}})^2 \leq 16(\delta_{baj_{ba}})^2 \leq 16(\frac{\delta_{ba}}{2^{\Lambda(j_{ba})}})^2$. □

4.3.4 The MILP-based iterative scheme

The following global optimization algorithm is executed based on (i) a local NLP solver (ii) a conic programming solver and (iii) a MILP solver. In this pseudocode, we use the function $\mathsf{err}(W) = \max_{(b,a)\in\mathcal{E}} | |W_{ba}|^2 - W_{bb}W_{aa}|$, which denotes the feasibility error in constraint ($\star$).

0. **Input:** A target optimality gap $\mathsf{targetOptGap} \geq 0$, a tolerance $\mathsf{feasTol} \geq 0$, integers $N_1, N_2 \in \mathbb{N}^*$ and a sequence $(\epsilon_t)_{t\in\mathbb{N}}$ with $\epsilon_t > 0$.

1. **Initialization:** Compute an ACOPF feasible solution with a NLP solver and denote by $\overline{\mathsf{obj}}$ its value (if the NLP solver fails, $\overline{\mathsf{obj}} \leftarrow +\infty$). Solve the conic programming relaxation ($\mathbf{ACOPF}_C$). If the gap is greater than $\mathsf{targetGap}$, apply FBBT and OBBT to ($\mathbf{ACOPF}_C$). Based on the optimal solution of Problem ($\mathbf{ACOPF}_C$), generate the LP relaxation ($\mathbf{R_L}$) with same value as ($\mathbf{ACOPF}_C$) (see section 4.3.1). Set $t \leftarrow 0$ and $\mathsf{LB}_t \leftarrow \mathsf{val}(\mathbf{ACOPF}_C)$.
2. **Outer-iterations:** While (i) $\overline{\mathsf{obj}} - \mathsf{LB}_t > \mathsf{targetGap}$ and (ii) $\mathsf{err}(W) > \mathsf{feasTol}$, do:
   - 2.1 For N_1 couples $(b, a) \in \mathcal{E}$ with largest violation $|\ R_{ba}^2 - W_{bb}W_{aa}|$, update the partition trees $\mathcal{J}_b$ and $\mathcal{J}_a$ according to section 4.3.3, and add the corresponding constraint (4.28)-(4.32) to the MILP relaxation.
   - 2.2 For N_2 couples $(b, a) \in \mathcal{E}$ with largest violation $|\ |W_{ba}|^2 - R_{ba}^2|$, update the partition tree $\mathcal{J}_{ba}$ according to section 4.3.3 and add the corresponding constraint (4.33)-(4.35) to the MILP relaxation.
   - 2.3 Solve the resulting MILP relaxation to global optimality to get

     $$x = (\mathsf{Re}(S), \mathsf{Im}(S), \mathsf{Re}(W), \mathsf{Im}(W), L, R).$$

     If the resulting lower bound has a gap with $\overline{\mathsf{obj}}$ lower than $\mathsf{targetGap}$, return; else enter the inner loop (step 3.). After the end of the inner loop, set LB_{t+1} as the value of the MILP relaxation and set $t \leftarrow t + 1$.
3. **Inner-iterations:** While x does not satisfy the convex constraints in ($\mathbf{ACOPF}_C$') within tolerance ϵ_t, i.e., while $\max_{j\in[\![1,r]\!]} \phi_j(x) > \epsilon_t$ and $\phi_0(x) - \max_{y\in\check{\mathcal{Y}}_0} y^\top \pi_0(x) > \epsilon_t$,
   - 3.1 Add the corresponding cuts: $\check{\mathcal{Y}}_j \leftarrow \check{\mathcal{Y}}_j \cup \{y\}$, for all $j \in [\![0, r]\!]$ and for $y \in \mathcal{Y}_j$ such that $\phi_j(x) = y^\top \pi_j(x)$.
   - 3.2 Solve the resulting MILP relaxation to global optimality to compute

     $$x = (\mathsf{Re}(S), \mathsf{Im}(S), \mathsf{Re}(W), \mathsf{Im}(W), L, R).$$

     If the resulting lower bound has a gap with $\overline{\mathsf{obj}}$ lower than $\mathsf{targetGap}$, return.

Figure 4.3 is a block diagram representing this pseudocode. During the inner-iterations (step 3.), for fixed sets $\mathcal{J}_b, \mathcal{J}_{ba}$ of binary variables α_{bj}, β_{baj}, the Mixed-integer conic programming problem obtained by adding these binary variables and the corresponding constraints (4.28)-(4.35) to the conic problem ($\mathbf{ACOPF}_C$), is approximated by MILP problems. From a semi-infinite programming perspective, this is an adaptive discretization algorithm, as presented in the Introduction, to solve this Mixed-integer conic programming relaxation of ($\mathbf{ACOPF}_W$). The next proposition, which relies on the Lemma 0.1, states the finite termination of these inner-loops.

Proposition 4.5. *For any outer-iteration index $t \in \mathbb{N}^*$, for any tolerance $\epsilon_t \in \mathbb{R}_{++}$, the inner-loop has a finite number of iterations.*

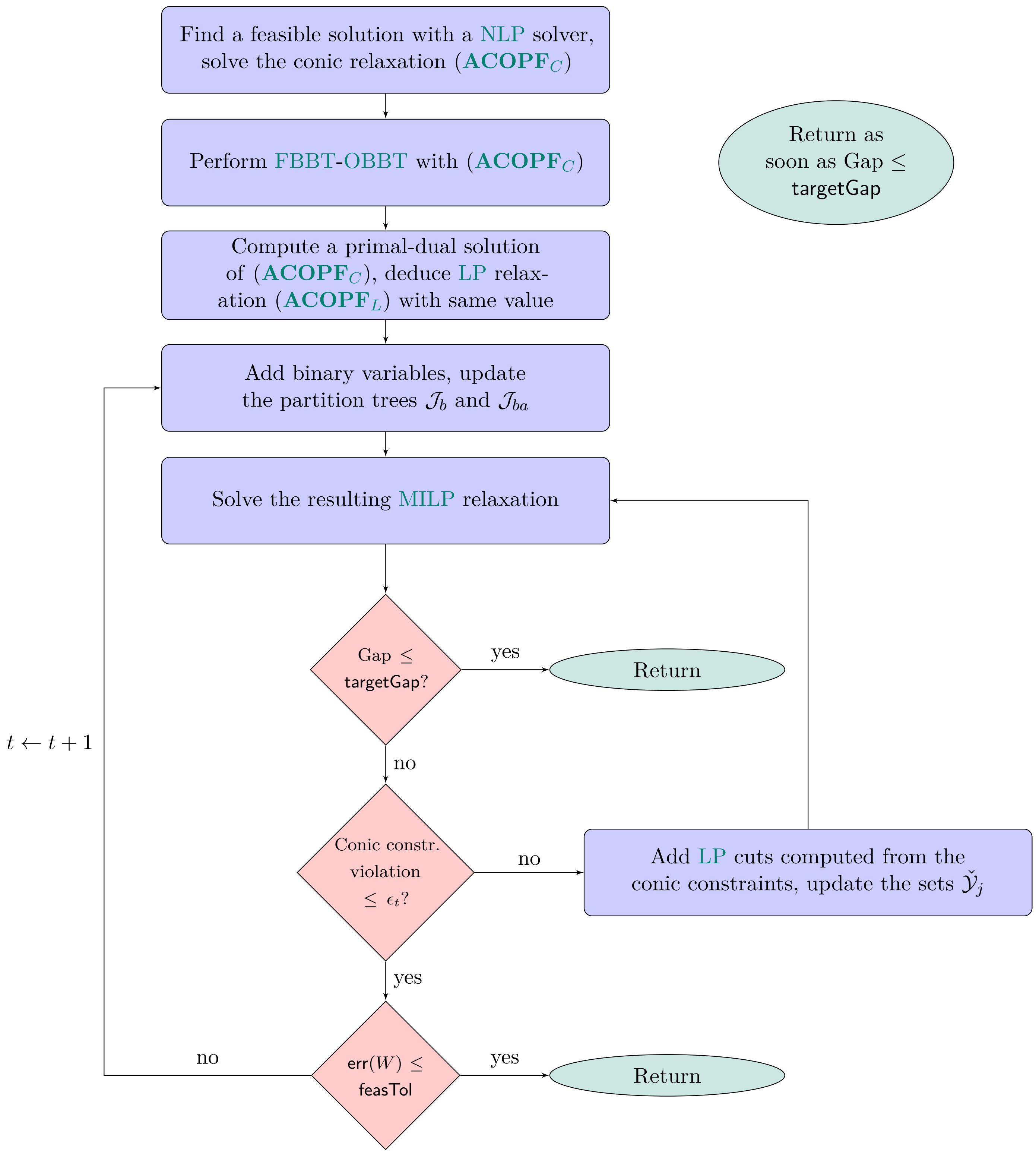

Find a feasible solution with a NLP solver, solve the conic relaxation ($\mathbf{ACOPF}_C$)
Perform FBBT-OBBT with ($\mathbf{ACOPF}_C$)
Compute a primal-dual solution of ($\mathbf{ACOPF}_C$), deduce LP relaxation ($\mathbf{ACOPF}_L$) with same value
Add binary variables, update the partition trees $\mathcal{J}_b$ and $\mathcal{J}_{ba}$
Solve the resulting MILP relaxation
Return as soon as Gap $\leq$ targetGap
Gap $\leq$ targetGap?
yes
Return
no
Conic constr. violation $\leq \epsilon_t$?
no
Add LP cuts computed from the conic constraints, update the sets $\check{\mathcal{Y}}_j$
yes
$\mathrm{err}(W) \leq$ feasTol
yes
Return
no
$t \leftarrow t+1$

Figure 4.3: MILP-based global optimization algorithm for ACOPF

Proof. During outer-iteration $t \in \mathbb{N}^*$ and the previous iterations, several auxiliary binary variables and associated linear constraints have been added to the relaxation (**ACOPF**$_C$). From the perspective of the decision vector $x = (\mathsf{Re}(S), \mathsf{Im}(S), \mathsf{Re}(W), \mathsf{Im}(W), L, R)$, these constraints together with the constraint of being in the polytope $\mathcal{P}$ yield a feasible set $\mathcal{X}^t \subset \mathbb{R}^\Xi$ that is nonconvex but compact. We also inherit finite sets $(\check{\mathcal{Y}}_{j0})_{j\in[\![0,r]\!]}$, the subscript 0 denoting the inner-iteration of index $s = 0$. The inner-iteration $s \in \mathbb{N}$ consists in solving

$$\begin{cases} \min\limits_{x\in\mathcal{X}^t,\lambda\in\mathbb{R}} & \lambda \\ \forall y \in \check{\mathcal{Y}}_{0s} & y^\top \pi_0(x) \leq \lambda \\ \forall j \in [\![1,r]\!], \forall y \in \check{\mathcal{Y}}_{js} & y^\top \pi_j(x) \leq 0, \end{cases} \tag{4.39}$$

to obtain a solution (x_s, λ_s), and in defining $\mathcal{Y}_{j(s+1)} = \{y_{js}\} \cup \mathcal{Y}_{js}$ with $y_{js} \in \operatorname{argmax}_{y\in\mathcal{Y}_j} y^\top \pi_j(x_s)$ for all $j \in [\![0,r]\!]$. By optimality in (4.39), we have that $\lambda_s = \max_{y\in\check{\mathcal{Y}}_{0s}} y^\top \pi_0(x)$.
We reason by contrapositive and assume now that the inner-loop is not terminating in finite time, meaning that the generated MILP relaxation is feasible at each inner-iteration and the stopping condition of the inner-loop is never met. For any $j \in [\![1,r]\!]$, we notice that the sequences $(x_s)_{s\in\mathbb{N}}$ and $(y_{js})_{s\in\mathbb{N}}$ satisfies $\phi_j(x_s) = y_{js}^\top \pi_j(x_s)$ (meaning that y_{js} is the output of a perfect separation oracle), and for all $\ell \in [\![0, s-1]\!]$, $y_{j\ell}^\top \pi_j(x_s) \leq 0$. Moreover, the sets $\mathcal{X}^t$ and $\mathcal{Y}_j$ are compact, and the mapping $(x,y) \mapsto y^\top \pi_j(x)$ is continuous. In summary, the hypotheses of the Lemma 0.1, about adaptive discretization for nonconvex semi-infinite programming, are met. This proves that $(\phi_j(x_s))^+ \to_s 0$. Applying the same result for $\tilde{x}_s = (x_s, \lambda_s)$ and $(\tilde{x}, y) \mapsto y^\top \pi_0(x) - \lambda$, we see that $(\phi_0(x_s) - \lambda_s)^+ \to_s 0$. As the stopping criterion is not met (we reason by contrapositive), we know that for all $s \in \mathbb{N}$, $\max\{(\phi_0(x_s) - \lambda_s)^+, \max_{j\in[\![1,r]\!]}(\phi_j(x_s))^+\} > \epsilon_t \geq 0$. As we just proved that the left term vanishes, this proves that $\epsilon_t = 0$. This is not true since we assumed $\epsilon_t \in \mathbb{R}_{++}$ in the hypotheses of the proposition. Therefore, the inner-loop terminates in finite time. □

The next theorem states the convergence of the whole algorithm presented in this section.

Theorem 4.3. *If* $\mathsf{targetOptGap} = \mathsf{feasTol} = 0$ *and* $\epsilon_t \to_t 0$*, then*

- *Either the algorithm stops due to the stopping criterion, and yields a global minimizer of* (**ACOPF**$_W$)*,*
- *Or the algorithm stops due to the infeasibility of a relaxation, certifying the infeasibility of* (**ACOPF**$_W$)*,*
- *Or the algorithm generates an infinite sequence of iterates* (S^t, W^t, L^t, R^t)*, and*
  - *The sequence* LB_t *monotonously converges to* $\mathsf{val}(\textbf{ACOPF}_W) = \mathsf{val}(\textbf{ACOPF})$*,*
  - *The limit points of the sequence* $(S^t, W^t)_{t\in\mathbb{N}}$ *are global minimizers of* (**ACOPF**$_W$)*.*

Proof. We consider the first case where the algorithm meets the stopping criterion at the beginning of the outer-iteration t, meaning that either (a) $\overline{\mathsf{obj}} = \mathsf{LB}_t$, proving that the solution

(S, V) found by the NLP solver at step 1. is globally optimal in (**ACOPF**) and yields (S, VV^{H}) globally optimal in ($\mathbf{ACOPF}_W$), or (b) the solution (S^t, W^t, L^t, R^t) of the current MILP relaxation of ($\mathbf{ACOPF}_W$) satisfies $\mathsf{err}(W^t) = 0$, i.e., (S^t, W^t) is in fact feasible in ($\mathbf{ACOPF}_W$) and thus optimal in ($\mathbf{ACOPF}_W$) since it is the optimal solution of a relaxation.

The second case is trivial: if the relaxation ($\mathbf{ACOPF}_C$) or any MILP relaxation during the iterations is infeasible, this implies that ($\mathbf{ACOPF}_W$) is also infeasible.

We consider now the third case, where the algorithm does not terminate. We invoke Proposition 4.5 to claim that for any outer-iteration $t \in \mathbb{N}$, the inner-loop terminates in finite time. Hence, there is an infinite number of outer-iterations and we define the infinite sequence $x_t = (S^t, W^t, L^t, R^t)_{t\in\mathbb{N}}$, where x_t is the solution of the MILP relaxation at the beginning of the outer-iteration t. For any $(b, a) \in \mathcal{E}$, we define $\chi^t_{ba} = |\,(R^t_{ba})^2 - W^t_{bb}W^t_{aa}|$ and $\xi^t_{ba} = |\,|W^t_{ba}|^2 - (R^t_{ba})^2|$. We let $\mathcal{J}^t_b$ (resp. $\mathcal{J}^t_{ba}$) denote the state of the tree $\mathcal{J}_b$ (resp. $\mathcal{J}_{ba}$) at the beginning of iteration t, and $\overline{\mathcal{J}}_b$ (resp. $\overline{\mathcal{J}}_{ba}$) the (potentially infinite) limit tree $\bigcup_t \mathcal{J}^t_b$ (resp. $\bigcup_t \mathcal{J}^t_{ba}$). We first show that $\chi^t_{ba} \to_t 0$ for any $(b, a) \in \mathcal{E}$. For $t \in \mathbb{N}$, we define $(b_t, a_t) \in \mathcal{E}$ such that $\chi^t_{b_ta_t} = \max_{ba} \chi^t_{ba}$, and we define $j_b(t) \in \mathcal{J}^t_{b_t}$ and $j_a(t) \in \mathcal{J}^t_{a_t}$ the active leaves to which three child nodes are attached during step 2.1 since (b_t, a_t) presents the largest violation. For any $j \in \bigcup_b \overline{\mathcal{J}}_b$, we recall that $\Lambda(j)$ is the depth of j in the (unique) tree $\overline{\mathcal{J}}_b$ it belongs to. As x_t is the output of the outer-iteration $t-1$, it satisfies constraint (4.3) and (4.5) with tolerance ϵ_{t-1}, and we can apply Proposition 4.4 with $\epsilon = \epsilon_{t-1}$. This yields

$$\chi^t_{b_ta_t} = |\,(R^t_{b_ta_t})^2 - W^t_{b_tb_t}W^t_{a_ta_t}| \le \frac{9\Delta_{b_t}\Delta_{a_t}}{2^{\max\{\Lambda(j_b(t)),\Lambda(j_a(t))\}}} + \max\{9\epsilon_{t-1}, (\frac{\Delta_{b_t}}{2^{\Lambda(j_b(t))}} + \frac{\Delta_{a_t}}{2^{\Lambda(j_a(t))}})^2\}. \tag{4.40}$$

We notice that the sequence $(j_b(t))_{t\in\mathbb{N}}$ is injective, since each $j_b(t)$ is a leaf in $\mathcal{J}^t_{b_t}$, but not in the trees $\mathcal{J}^s_{b_t}$ for $s \ge t+1$. We deduce that $\Lambda(j_b(t)) \to_t \infty$, otherwise by contrapositive, there would exist $H \in \mathbb{N}$ such that an infinite number of nodes of depth less or equal than H are created in the union of ternary trees $\bigcup_b \overline{\mathcal{J}}_b$; This is false since the number of nodes with depth less or equal than H is bounded by $|\mathcal{B}| \sum_{\ell=0}^{H} 3^\ell$. By the same argument, we have $\Lambda(j_a(t)) \to_t \infty$. Combined with (4.40), we deduce that $\chi^t_{b_ta_t} \to_t 0$ since $\epsilon_t \to_t 0$ and because $\Delta_{b_t}, \Delta_{a_t}$ are bounded. For any $t \in \mathbb{N}$ and $(b, a) \in \mathcal{E}$, we have $0 \le \chi^t_{ba} \le \chi^t_{b_ta_t}$ by definition of (b_t, a_t), implying $\chi^t_{ba} \to_t 0$.

We apply the same approach to prove that $\xi^t_{ba} \to_t 0$ for any $(b, a) \in \mathcal{E}$. For $t \in \mathbb{N}$, we define $(\tilde{b}_t, \tilde{a}_t) \in \mathcal{E}$ such that $\xi^t_{\tilde{b}_t\tilde{a}_t} = \max_{ba} \xi^t_{ba}$, and we define $\tilde{j}(t) \in \mathcal{J}^t_{\tilde{b}_t\tilde{a}_t}$ the active leaf to which three child nodes are attached during step 2.2. We also define $\Lambda(\tilde{j}(t))$ as the depth of $\tilde{j}(t)$ in $\mathcal{J}^t_{\tilde{b}_t\tilde{a}_t}$, which satisfies $\Lambda(\tilde{j}(t)) \to_t \infty$ by injectivity of $(\tilde{j}(t))_{n\in\mathbb{N}}$ and since the number of nodes in $\bigcup_{(b,a)\in\mathcal{E}} \overline{\mathcal{J}}_{ba}$ with depth less or equal than M is bounded by $|\mathcal{E}| \sum_{\ell=0}^{r} 3^\ell$. As $\Lambda(\tilde{j}(t)) \to_t \infty$, we know that it exists $t_0 \in \mathbb{N}$ such that $\Lambda(\tilde{j}(t)) \ge 2$ for all $t \ge t_0$. Hence, for all $t \ge 0$, $\Lambda(\tilde{j}(t)) \ge \log_2(\frac{4\pi}{\pi}) \ge \log_2(\frac{2\delta_{\tilde{b}_t\tilde{a}_t}}{\pi})$, since $\delta_{b_ta_t} \in [0, 2\pi]$. Applying Proposition 4.4, we know that for any $t \geq t_0$,

$$\xi^t_{\tilde{b}_t\tilde{a}_t} = |\,|W^t_{\tilde{b}_t\tilde{a}_t}|^2 - (R^t_{\tilde{b}_t\tilde{a}_t})^2| \leq \max\{9\epsilon_{t-1}, \frac{16(\delta_{\tilde{b}_t\tilde{a}_t})^2}{4^{\Lambda(\tilde{j}(t))}}\} \leq \max\{9\epsilon_{t-1}, \frac{64\pi^2}{4^{\Lambda(\tilde{j}(t))}}\}, \tag{4.41}$$

Combined with $\epsilon_t \rightarrow_t 0$ and $\Lambda(\tilde{j}(t)) \rightarrow_t \infty$, we deduce that $\xi^t_{\tilde{b}_t\tilde{a}_t} \rightarrow_t 0$. Additionally, since $\xi^t_{\tilde{b}_t\tilde{a}_t} = \max_{ba} \xi^t_{ba}$, we have $0 \leq \xi^t_{ba} \leq \xi^t_{\tilde{b}_t\tilde{a}_t}$ and, thus, $\xi^t_{ba} \rightarrow_t 0$ for any $(b,a) \in \mathcal{E}$.

We deduce that $\mathsf{err}(W^t) \rightarrow_t 0$, since $\mathsf{err}(W^t) = \max_{(b,a)\in\mathcal{E}} ||W^t_{ba}|^2 - W^t_{bb}W^t_{aa}| \leq \sum_{(b,a)\in\mathcal{E}} ||W^t_{ba}|^2 - W^t_{bb}W^t_{aa}| \leq \sum_{(b,a)\in\mathcal{E}} \chi^t_{ba} + \xi^t_{ba}$, due to the triangle inequality. Hence, for any limit point $(\bar{S}, \bar{W})$ of (S^t, W^t), we thus have $\mathsf{err}(\bar{W}) = 0$. As $\epsilon_t \rightarrow_t 0$, this also proves that $(\bar{S}, \bar{W})$ satisfies the nonlinear convex constraints in (**ACOPF**$_C$). Hence, $(\bar{S}, \bar{W})$ is feasible in (**ACOPF**$_W$). We denote by $\check{\phi}^t_0(x) = \max_{y\in\check{\mathcal{Y}}_0} y^\top \pi_0(x)$ the cutting-plane model of the objective function at the beginning of iteration t (the set $\check{\mathcal{Y}}_0$ depending implicitly on t). As this function only depends on the variable S and not on the other decision variables, we write $\check{\phi}^t_0(S)$. As the successive MILP relaxations over the iterations have nonincreasing feasible sets with respect to variables (S, W, L, R) and have nondecreasing sequence $\check{\phi}^t_0(S)$ as objective functions, the sequence $\check{\phi}^t_0(S^t) = \mathsf{LB}_t$ is nondecreasing. It is also bounded above by val(**ACOPF**$_W$) and, thus, converges to a value $v^* \leq \mathsf{val}$(**ACOPF**$_W$). Since $\epsilon_t \rightarrow_t 0$, $\check{\phi}^t_0(S^t) \rightarrow_t \phi_0(\bar{S}) = \sum_{g\in\mathcal{G}} c_{1g}\mathsf{Re}(\bar{S}_g) + c_{2g}\mathsf{Re}(\bar{S}_g)^2$, for any limit point $(\bar{S}, \bar{W})$. By uniqueness of the limit of $\check{\phi}^t_0(S^t)$, and since $(\bar{S}, \bar{W})$ is feasible in (**ACOPF**$_W$), we deduce that $v^* = \sum_{g\in\mathcal{G}} \left(c_{0g} + c_{1g}\,\mathsf{Re}(\bar{S}_g) + c_{2g}\,\mathsf{Re}(\bar{S}_g)^2\right) \geq \mathsf{val}$(**ACOPF**$_W$). We conclude that $v^* = \mathsf{val}$(**ACOPF**$_W$) $= \mathsf{val}$(**ACOPF**) and that $(\bar{S}, \bar{W})$ is optimal in (**ACOPF**). □

4.4 Numerical experiments

4.4.1 Experimental setting

For all experiments, we use a 64-bit Ubuntu computer with 32 Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz and 64 GB RAM. Along our algorithm, we use the commercial solvers `MOSEK` [168] and `CPLEX` [108] called through their `python` APIs, as well as the academic solver `Ipopt` [238] called through the `Pyomo` interface [97]. We compute the tree decomposition with the approximate minimum degree ordering routine of the `chompack` package. We consider a relative optimality gap of 0.01% for global optimality (**GO**) and use the parameters $(N_1, N_2, \bar{\epsilon}) = (4, 4, 10^{-6})$. The FBBT and OBBT procedures are applied for all variables a maximum of 4 times, and with a time limit of 10 hours. After each pass of FBBT and OBBT, we apply the Floyd-Warshall algorithm and we check whether the gap of the tightened conic relaxation reaches the target optimality gap. If the maximum number of bound tightening iterations or time limit is reached, we enter the MILP iterative scheme with a time limit of 5 hours. Hence, our global time limit is 15 hours. Our code is available at `github.com/aoustry/SDP-MILP4OPF`.

This study focuses on the network instances from the IEEE PES PGLib AC-OPF v21.07 library [9] with less than 500 buses. As shown in Table 4.2, the instances of this benchmark are split in three categories depending on their characteristics: Typical Operating Condition (TYP) instances correspond to a reference scenario, Congested Operating Condition (API) correspond to situations with greater Power Demands, and Small Angle Condition (SAD) correspond to tighter constraints for the phase angle difference.

We compare our approach with the standard SOCP and SDP relaxations [165], and with three other global optimization approaches [87, 94, 212]. We performed these comparative experiments on the same PGLib v21.07 instances, with the same hardware, the same global time limit (15 hours), and the same relative optimality gap tolerance (0.01%). The concurrent approach from [212] is an OBBT algorithm based on a strengthened convex QCQP relaxation. We ran the `Julia` implementation of this algorithm provided in the `PowerModels.jl` package [53]. The convex quadratic relaxations are solved with `Ipopt`. The concurrent approach from [87] consists of an OBBT algorithm, based on a Determinant SDP relaxation strengthened with RLT constraints. We ran a `C++` implementation of this algorithm, that is based on the Mathematical Modeling Language `Gravity` [105], and is available at the link indicated in [87]. The corresponding relaxations are also solved with `Ipopt`. Finally, we also tested the global optimization algorithm for generic QCQP problems that is built into the solver `Gurobi 10.0` [94]. We highlight that this algorithm is not specific to the ACOPF problem. This solver is called through the `PowerModels.jl` interface.

4.4.2 Numerical results

Table 4.2 presents the optimality gap (in %) and the computational times obtained by the several approaches for the considered list of instances. For lack of space, we do not detail the computation time of the SOCP and SDP relaxations. To give an idea, the computation time of the SOCP relaxation is below 2s for all instances; for the SDP relaxation, the computation time goes from 0.2s for the smallest cases to about 40s for the largest cases. As regards the column "This work", in the optimality gap section of the table: the entry (**GO**) means that the bound tightening procedure based on the conic programming relaxation closed the gap; else, the entry a/b represents the gap after the bound tightening procedure (a) and the gap after the iterative MILP scheme (b). Figure 4.4 shows the histograms of the optimality gap for the four global optimization methods compared. Regarding Figure 4.5, it compares these algorithms by showing for how many instances it reaches global optimality (in minimum time), and the best gap among the four algorithms.

In Fig 4.4, we see that for the four considered optimality gap thresholds, this work obtains the largest number of instances with a gap below the threshold. Our algorithm reaches global optimality for 35 instances over 51. The optimality gap is below 1% for 47 instances over 51. For 4 instances only, the optimality gap at the end of the algorithm is above 2%. As regards the instances with less than 57 buses, they are all solved to global optimality in less than 212 seconds. For all these instances with less than 57 buses, except case5_pjm and case30_as_api, the

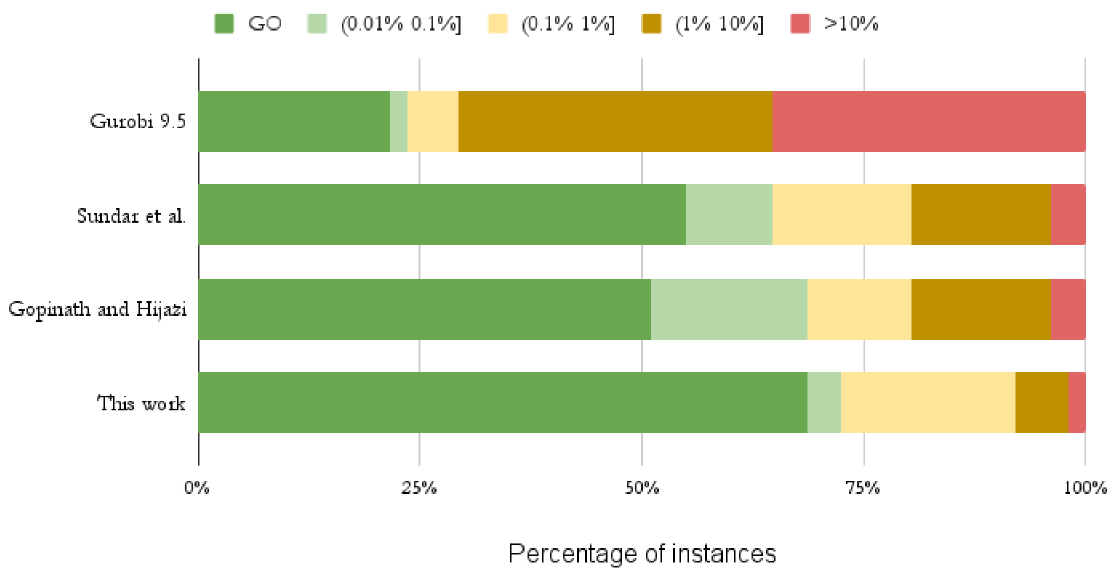


Figure 4.4: Optimization gap histograms

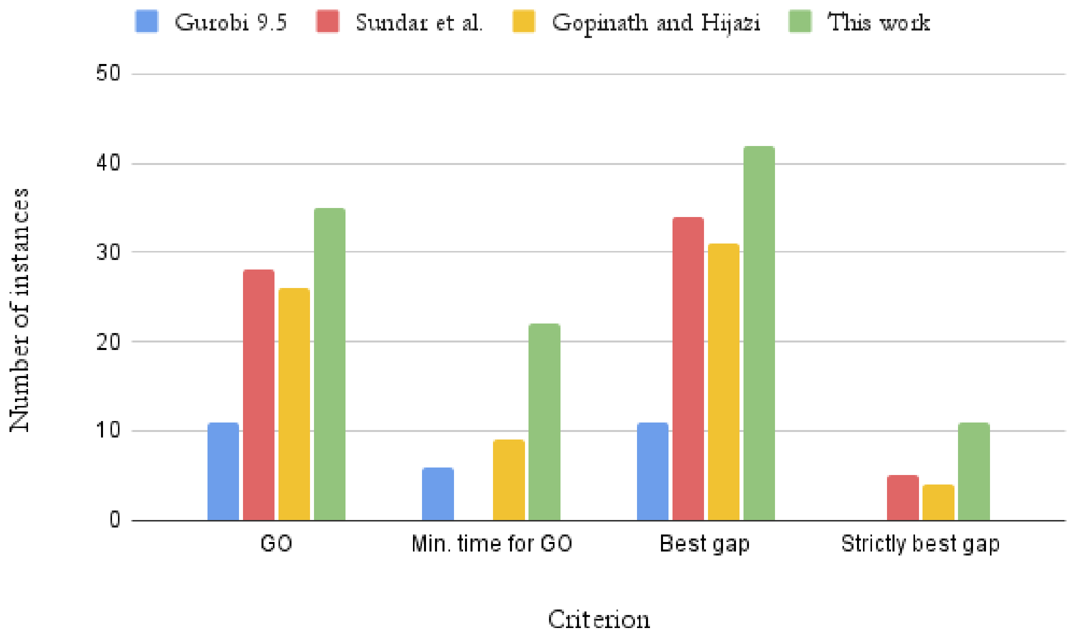


Figure 4.5: Number of instances for which each algorithm performs best, for multiple criteria

bound tightening procedure based on the conic programming relaxation (**ACOPF**$_C$) manages to close the gap. For the instances case5_pjm and case30_as_api, the gap is closed by the iterative MILP scheme. For all the instances with more than 57 nodes where the MILP scheme is executed, the gap is admittedly not closed within the time limit, but it is reduced, except for case500_goc_api. Figure 4.5 shows that for 42 over these 51 instances, our algorithm has the lowest gap; for 11 instances over 51, it has a strictly lower gap than the others approaches. For 5 instances (among those 11 instances), our approach is the only one to reach global optimality. Regarding the 9 instances where our approach has not the best gap: for 5 instances, the QC relaxation-based bound tightening algorithm [212] yields the lowest gap ; for the 4 other instances, the Determinant-SDP relaxation-based bound tightening algorithm [87] yields the lowest gap. Still in Figure 4.5, we see that our approach is the fastest to certify global optimality for 22 instances; this is the case for only 6 instances for `Gurobi 10.0`, and for only 9 instances for [87]. In summary, for all the criterion presented in Figure 4.4 and Figure 4.5, our algorithm presents the best performance.

4.5 Conclusion

We introduce a conic programming relaxation for the ACOPF problem. This relaxation is a strengthening of the classical SDP relaxation with additional variables and valid inequalities. These inequalities dominate previously introduced nonlinear cuts used to strengthen convex relaxations. Our numerical experiments on a reference benchmark illustrate that this conic programming relaxation is particularly suitable for a bound tightening procedure: it closes the gap in many cases where a bound tightening based on a quadratic convex relaxation does not. We also introduce an iterative scheme based on MILP, that asymptotically converges to global minimizers of the ACOPF problem. For cases where the bound tightening procedure does not close the gap, this iterative scheme significantly reduces the gap in most cases. Our numerical experiments on a standard benchmark for the ACOPF problem show that our algorithm outperforms three other global optimization approaches in terms of the number of instances solved to global optimality or below a certain optimization gap and in terms of the computational time to global optimality. For a timeout of 15h, our algorithm obtains the lowest gap among the compared methods in 82% of the cases.

A first line of research concerns the scalability of conic programming relaxations with semidefinite constraints, such as the one proposed in this chapter. So far, the scalability of semidefinite programming is not as good as that of linear programming or second-order conic programming. This challenge is the main topic of Chapter 5. This chapter addresses the numerical issues that arise when solving semidefinite relaxations of large ACOPF instances.

A future line of research will be to improve the scalability of the optimization-based bound tightening: parallelizing this procedure or targeting the bounds to tighten based on the graph structure. Another avenue to explore is the possibility of speeding up the solution of the MILP problem at a given iteration by reusing the branch-and-bound trees of the problems solved during the previous iterations. From an applicative point of view, our method could be extended to the Optimal Transmission Switching problem, where the network topology can be modified by switching electrical lines. We can also foresee an extension to Unit Commitment problems with AC power flow equations, where generators can be switched on or off. Indeed, the proposed MILP scheme could easily accommodate additional binary variables to describe these switches.

	Optimality gap (%)						Time (s)			
	Benchmark					This work	Benchmark			This
Case	SOCP	SDP	[94]	[212]	[87]	BT/MILP	[94]	[212]	[87]	work
Typical Operating Condition (TYP)										
3lmbd	1.32	0.39	**GO**	**GO**	**GO**	**GO**	**<1**	1	**<1**	1
5pjm	14.6	5.21	**GO**	5.76	**GO**	*5.01*/**GO**	**2**	44	21	205
14ieee	0.11	**GO**	**GO**	**GO**	**GO**	**GO**	1,380	12	**<1**	3
24ieeerts	0.02	**GO**	0.48	**GO**	**GO**	**GO**	TL	41	8	**7**
30as	0.06	**GO**	34.0	**GO**	**GO**	**GO**	TL	107	**2**	10
30ieee	18.8	**GO**	1.14	**GO**	**GO**	**GO**	TL	226	**2**	8
39epri	0.56	**GO**	0.34	**GO**	**GO**	**GO**	TL	272	**2**	9
57ieee	0.16	**GO**	3.66	**GO**	**GO**	**GO**	TL	437	178	**13**
73ieeerts	0.04	**GO**	3.43	**GO**	**GO**	**GO**	TL	399	22	**21**
89pegase	0.75	0.37	4.15	0.22	**0.08**	*0.27*/0.18	TL	TL	TL	TL
118ieee	0.91	0.07	4.66	**GO**	**GO**	**GO**	TL	8,440	TL	**783**
162ieeedtc	5.95	1.78	10.8	**0.02**	1.57	*0.59*/0.53	TL	TL	TL	TL
179goc	0.16	0.07	0.43	**0.03**	**0.03**	*0.04*/**0.03**	TL	TL	TL	TL
200activ	0.01	**GO**	5.46	**GO**	**GO**	**GO**	TL	2,250	1,570	**54**
240pserc	2.78	1.43	4.44	2.71	1.20	*1.02*/**0.93**	TL	TL	TL	TL
300ieee	2.63	1.03	12.5	2.55	0.05	**GO**	TL	TL	TL	**5,480**
500goc	0.25	**GO**	112	0.13	**GO**	**GO**	TL	TL	10,100	**139**
Congested Operating Condition (API)										
3lmbd__api	9.27	7.10	**GO**	**GO**	**GO**	**GO**	**<1**	2	3	2
5pjm__api	4.09	0.26	**GO**	**GO**	**GO**	**GO**	**1**	10	51	3
14ieee__api	5.13	**GO**	**GO**	**GO**	**GO**	**GO**	612	55	**1**	4
24ieeerts__api	17.9	2.07	0.10	**GO**	**GO**	**GO**	TL	1,310	1,950	**97**
30as__api	44.6	10.9	30.1	36.1	0.63	*0.13*/**GO**	TL	3,330	3,330	**216**
30ieee__api	5.46	**GO**	25.3	**GO**	0.02	**GO**	TL	860	1,670	**8**
39epri__api	1.72	0.20	2.54	**GO**	**GO**	**GO**	TL	652	2,080	**96**
57ieee__api	0.08	**GO**	1.88	**GO**	**GO**	**GO**	TL	487	**10**	13
73ieeerts__api	12.9	2.90	21.7	0.28	0.23	*0.28*/**0.08**	TL	TL	TL	TL
89pegase__api	23.1	22.0	33.5	**17.2**	17.6	*21.7*/19.3	TL	TL	TL	TL
118ieee__api	30.0	11.7	50.2	3.16	1.44	*1.26*/**0.90**	TL	21,350	TL	TL
162ieeedtc__api	4.36	1.44	10.3	**0.18**	1.31	*0.29*/0.25	TL	TL	TL	TL
179goc__api	9.88	0.59	4.25	0.54	**0.38**	*0.54*/0.53	TL	TL	TL	TL
200activ__api	0.03	1.49	77.8	**GO**	**GO**	**GO**	TL	3,410	1,240	**55**
240pserc__api	0.67	0.28	1.96	0.62	*err.*	*0.12*/**0.11**	TL	TL	*err.*	TL
300ieee__api	0.85	0.09	4.98	0.81	0.07	**GO**	TL	TL	TL	**7,840**
500goc__api	3.44	2.36	581	3.00	**2.12**	*2.19*/2.19	TL	TL	TL	TL
Small Angle Difference (SAD)										
3lmbd__sad	3.75	1.86	**GO**	**GO**	**GO**	**GO**	<1	1	3	1
5pjm__sad	3.62	**GO**	**GO**	**GO**	**GO**	**GO**	**<1**	4	**<1**	1
14ieee__sad	21.5	0.09	**GO**	**GO**	0.11	**GO**	110	135	500	**17**
24ieeerts__sad	9.55	4.35	**GO**	**GO**	**GO**	**GO**	5,420	140	1,450	**98**
30as__sad	7.88	0.24	18.1	**GO**	0.09	**GO**	TL	168	2,600	**71**
30ieee__sad	9.70	**GO**	1.39	**GO**	**GO**	**GO**	TL	197	**5**	9
39epri__sad	0.67	0.02	**GO**	**GO**	**GO**	**GO**	41,600	193	1,500	**90**
57ieee__sad	0.71	0.05	34.9	**GO**	**GO**	**GO**	TL	674	6,500	**212**
73ieeerts__sad	6.73	2.74	7.17	**GO**	0.05	**GO**	TL	7,650	TL	**1,470**
89pegase__sad	0.73	0.37	3.55	0.34	**0.07**	*0.28*/0.19	TL	TL	TL	TL
118ieee__sad	8.17	3.25	12.1	0.02	0.19	**GO**	TL	TL	TL	**1,690**
162ieeedtc__sad	6.48	2.07	11.0	**0.02**	1.35	*0.51*/0.48	TL	TL	TL	TL
179goc__sad	1.12	0.95	1.46	**0.03**	0.71	*0.66*/0.42	TL	TL	TL	TL
200activ__sad	0.03	**GO**	5.46	**GO**	**GO**	**GO**	TL	2,230	460	**53**
240pserc__sad	4.93	3.42	6.73	4.34	3.16	*2.63*/**2.61**	TL	TL	TL	TL
300ieee__sad	2.61	0.67	12.8	2.33	0.05	**GO**	TL	TL	TL	**5,650**
500goc__sad	6.67	5.68	127	5.29	5.59	*5.21*/**5,18**	TL	TL	TL	TL

Table 4.2: Detailed results for the instances from IEEE PES PGLib AC-OPF v21.07 with less than 500 buses

Chapter 5

Certified and accurate spectral bounds for the ACOPF problem

Convex relaxations are fundamental tools for designing global optimization algorithms for nonlinear problems such as the AC Optimal Power Flow (ACOPF) problem since they provide lower bounds on the problem value [164]. Among them, the SDP relaxation is of strong interest for both theoretical and practical reasons: indeed, this is the ACOPF's bidual (the dual of the dual) [135], and this relaxation is known to provide tight lower bounds [116]. In Chapter 4, we use the SDP relaxation as a starting point for our global optimization algorithm. As noted in this chapter, compared to other convex relaxations, the computational cost of the SDP relaxation is a drawback that limits its applicability to large-scale instances. Another drawback is the numerical difficulties encountered by SDP solvers for large-scale instances. [116, 231]. This chapter addresses these limitations to improve the scalability of the SDP relaxation and, in fine, of any global optimization algorithm based on this relaxation.

State-of-the-art algorithms for the SDP relaxation

The main drawback of the standard SDP relaxation (also known as Schor relaxation, or rank relaxation) is that it involves a dense $n \times n$ matrix as a decision variable, where n is the number of buses. This becomes intractable when n exceeds magnitudes of around 10^3. To overcome this computational burden, state-of-the-art approaches to solve the SDP relaxation [70, 166] exploit the sparse structure of the power grid by using a tree decomposition technique, as presented in Chapter 4. Thanks to a semidefinite completion theorem [91] for matrices with chordal sparsity pattern, it is possible to find an equivalent formulation for the semidefinite relaxation, with many small semidefinite blocks instead of a single and large $n \times n$ matrix. Each of these blocks corresponds to a clique of the tree decomposition. This clique-based semidefinite relaxation can be solved with a symmetric IPM solver [168, 211], with a non-symmetric IPM solver [4], or with a first-order method like the Alternating Direction Method of Multipliers (ADMM) [154, 184, 244]. Eltved et al. [70] report the solution of the clique-based ACOPF's semidefinite relaxation for test cases with up to $82,000$ buses in a few hours. Despite these advances, the clique-based semidefinite relaxation remains difficult to solve for large-scale instances: numerical instabilities may arise when solving this convex optimization problem. In [116] and [231], the authors report numerical instabilities with the commercial solver `MOSEK` [168], which raises a warning for many tested instances. In [70], the authors report that the academic IPM solver `SeDuMi` fails to solve the clique-based relaxation in 50% of the test cases; and that the ADMM-based solver `CDCS` [244] often terminates with sizable dual residuals.

Two categories of troubles may arise due to numerical difficulties in solving the semidefinite relaxation. First, they may limit the accuracy of the calculation of the relaxation value. Second, obtaining a solution with nonzero primal and/or dual feasibility errors implies that the calculated relaxation value is not *certified* [229]. In this case, one obtains an *approximated* value of the relaxation without knowing if it is an *exact* lower bound on the ACOPF's value.

Contributions and organization of the chapter

This chapter tackles both aforementioned numerical issues with an original approach. For this purpose, we propose a novel formulation for the Lagrangian dual of the ACOPF, the value of which equals the value of the semidefinite relaxation. Our formulation is a concave maximization problem with the following interesting properties: (a) it is unconstrained (b) the objective function is partially separable. This formulation is qualified as *spectral* because it involves the minimum eigenvalue $\lambda_{\min}$ of a matrix operator. Based on this formulation, we present how to obtain a certified lower bound from any dual vector, whether feasible or not, in the classical dual semidefinite relaxation. We solve this unconstrained dual problem with a structure-exploiting polyhedral bundle method. We use this algorithm as a *post-processing step*, after solving the clique-based semidefinite relaxation with the commercial IPM solver `MOSEK` [168], which is the state-of-the-art solver for ACOPF's semidefinite relaxation according to [70, 164]. Our numerical experiments on instances from IEEE PES PGLib AC-OPF v21.07 [9] show that this post-processing considerably improves the tightness of the dual bounds.

This work is not the first to take the path of spectral reformulations and nonsmooth optimization to solve a SDP problem [99, 183]. We mention, however, two original contributions of the present chapter with respect to [99, 183]: (i) we combine this spectral approach with the clique decomposition technique for semidefinite programming, leading to a structure-exploiting nonsmooth optimization algorithm, and (ii) we solve a spectral problem involving Hermitian complex matrix operators, whereas the mentioned works for spectral bundle methods addressed real symmetric matrix operators only. Compared to [189], a previous work applying Lagrangian duality for the ACOPF and solving a nonsmooth dual, our structure-exploiting algorithm based on clique decomposition enables us to address ACOPF instances of a much larger scale.

The chapter is organized as follows. Section 5.1 reviews the formulations of the ACOPF problem, the semidefinite relaxation, and its dual. Section 5.2 introduces the unconstrained dual formulation and illustrates its advantage in computing certified lower bounds. The bundle method to solve this unconstrained problem is presented in Section 5.3, and the numerical experiments in Section 5.4.

5.1 ACOPF, semidefinite relaxation and dual problem

This section reviews the ACOPF formulation solved here, with line limits in terms of current (see Eqs. (7bis)-(8bis)), the semidefinite relaxation based on the tree decomposition, and the standard dual problem.

5.1.1 Problem formulation

As explained in the Introduction, the ACOPF problem involves a direct graph $\mathcal{G} = (\mathcal{B}, \mathcal{L})$ describing the power grid. The set of generators connected to the bus $b \in \mathcal{B}$ is denoted by $\mathcal{G}_b$, and the set of all the generators is $\mathcal{G} = \bigcup_b \mathcal{G}_b$. The parameters involved in the ACOPF formulation were given in the Introduction and in the Chapter 4 of this manuscript (see Table 4.1 for instance). Additionally, we introduce here some matrices related to the line limits. For any line $(b, a) \in \mathcal{L}$, we define the Hermitian matrix

$$N_{ba} = |Y^{\mathsf{ff}}_{ba}|^2 E_{bb} + Y^{\mathsf{ff}}_{ba}(Y^{\mathsf{ft}}_{ba})^* E_{ab} + (Y^{\mathsf{ff}}_{ba})^* Y^{\mathsf{ft}}_{ba} E_{ba} + |Y^{\mathsf{ft}}_{\ell}|^2 E_{aa}, \tag{5.1}$$

and for any $(b, a) \in \mathcal{L}^R$, we define the Hermitian matrix

$$N_{ba} = |Y^{\mathsf{tt}}_{ab}|^2 E_{bb} + Y^{\mathsf{tt}}_{ab}(Y^{\mathsf{tf}}_{ab})^* E_{ab} + (Y^{\mathsf{tt}}_{ab})^* Y^{\mathsf{tf}}_{ab} E_{ba} + |Y^{\mathsf{tf}}_{ab}|^2 E_{aa}, \tag{5.2}$$

recalling that the matrices E_{ba} are the elements of the canonical basis of $\mathbb{C}^{\mathcal{B}\times\mathcal{B}}$. Hence, constraints (7bis) and (8bis) read $\left\langle N_{ba}, VV^{\mathsf{H}} \right\rangle \leq (\bar{I}_{ba})^2$ for all $(b, a) \in \mathcal{L} \cup \mathcal{L}^R$. In summary, the ACOPF formulation with current line limits is

$$\left.\begin{array}{rl}
\displaystyle\min_{V\in\mathbb{C}^{\mathcal{B}},\, S\in\mathbb{C}^{\mathcal{G}}} & \displaystyle\sum_{g\in\mathcal{G}} \left(c_{0g} + c_{1g}\,\mathsf{Re}(S_g) + c_{2g}\,\mathsf{Re}(S_g)^2\right) \\
\text{s.t. } \forall b \in \mathcal{B}, & \underline{v}_b \leq |V_b| \leq \overline{v}_b \\
\forall g \in \mathcal{G}, & \underline{s}_g \leq S_g \leq \overline{s}_g \\
\forall b \in \mathcal{B}, & \displaystyle\sum_{g\in\mathcal{G}_b} S_g - S^{\mathsf{d}}_b = \langle M_b, VV^{\mathbf{H}} \rangle \\
\forall \ell \in \mathcal{L} \cup \mathcal{L}^R & \left\langle N_\ell, VV^{\mathsf{H}} \right\rangle \leq (\bar{I}_\ell)^2.
\end{array}\right\} \tag{ACOPF}$$

For the sake of readability, we work here with current line limits, but the approach of this chapter could also be applied to power magnitude limits (7)-(8), since they can be formulated as convex constraints with respect to W (see for example (**ACOPF**$_W$)). In the following, we do not show the offsets c_{0g} for the sake of readability, but we do take them into account in the numerical results for the values displayed.

5.1.2 The (clique-based) semidefinite relaxation

The standard semidefinite relaxation, also known as rank relaxation, is classically derived by replacing the rank-one matrix VV^{H} in (ACOPF) by a Hermitian and PSD matrix W of unspecified rank. Hence, this is the following convex optimization problem

$$\left.\begin{array}{cl} \min\limits_{W\succeq 0,\, S\in\mathbb{C}^{\mathcal{G}}} & \sum\limits_{g\in\mathcal{G}}\Big(c_{1g}\,\mathsf{Re}(S_g)+c_{2g}\,\mathsf{Re}(S_g)^2\Big) \\ \forall b\in\mathcal{B} & \underline{v}_b^2\le W_{bb}\le\overline{v}_b^2 \\ \forall g\in\mathcal{G} & \underline{s}_g\le S_g\le\overline{s}_g \\ \forall b\in\mathcal{B} & \sum\limits_{g\in\mathcal{G}_b}S_g-S_b^{\mathsf{d}}=\langle M_b,W\rangle \\ \forall \ell\in\mathcal{L}\cup\mathcal{L}^R & \langle N_\ell,W\rangle\le(\bar{I}_\ell)^2 \end{array}\right\} \qquad (\textbf{SDR})$$

This formulation becomes intractable for large-scale instances [164]. A standard approach [220] to avoid this computational burden consists in the following. We consider a given tree decomposition $\mathcal{T}$ of the graph $\mathcal{N}$ (see Definition 4.1), and the associated set $\mathcal{B}_k\subset\mathcal{B}$ for any index $k\in\mathcal{T}$, the size of which is denoted $\mathbf{n}_k$. We introduce the symmetric set $\mathcal{E}\subset\mathcal{B}\times\mathcal{B}$ of arcs defined as $\mathcal{E}=\bigcup_{k\in\mathcal{T}}\mathcal{B}_k\times\mathcal{B}_k$. As a matter of fact, the sets $\mathcal{B}_k$ are cliques of the undirected graph $(\mathcal{B},\mathcal{E})$. The sets $\mathcal{B}_k$ are therefore simply called *cliques*. For any symmetric set $\tilde{\mathcal{E}}\subset\mathcal{E}$, we define $\mathbb{H}(\tilde{\mathcal{E}})=\{W\in\mathbb{C}^{\tilde{\mathcal{E}}}:\forall(b,a)\in\tilde{\mathcal{E}},W_{ba}=W_{ab}^*\}$, the set of Hermitian matrices with sparsity pattern $\tilde{\mathcal{E}}$. For any $k\in\mathcal{T}$, we denote by $W_{\mathcal{B}_k,\mathcal{B}_k}$ the matrix $(W_{ba})_{(b,a)\in\mathcal{B}_k^2}$. The clique-based semidefinite relaxation, also known as chordal relaxation, reads

$$\left.\begin{array}{cl} \min\limits_{W\in\mathbb{H}(\mathcal{E}),\, S\in\mathbb{C}^{\mathcal{G}}} & \sum_{g\in\mathcal{G}}\Big(c_{1g}\,\mathsf{Re}(S_g)+c_{2g}\,\mathsf{Re}(S_g)^2\Big) \\ \forall b\in\mathcal{B} & \underline{v}_b^2\le W_{bb}\le\overline{v}_b^2 \\ \forall g\in\mathcal{G} & \underline{s}_g\le S_g\le\overline{s}_g \\ \forall b\in\mathcal{B} & \sum\limits_{g\in\mathcal{G}_b}S_g-S_b^{\mathsf{d}}=\ \langle M_b,W\rangle \\ \forall \ell\in\mathcal{L}\cup\mathcal{L}^R & \langle N_\ell,W\rangle\le(\bar{I}_\ell)^2 \\ \forall k\in\mathcal{T} & W_{\mathcal{B}_k,\mathcal{B}_k}\succeq 0. \end{array}\right\} \qquad (\textbf{cSDR})$$

The advantage of formulation (**cSDR**) is that the total number of coefficients in the matrix W is $\sum_{k\in\mathcal{T}}\mathbf{n}_k^2$, whereas the matrix variable W in (**SDR**) involves $|\mathcal{B}|^2$ non-zero coefficients. If the cliques are of limited size ($\mathbf{n}_k\ll|\mathcal{B}|$), then this decomposition is particularly relevant. This is often the case, since power grids are known to have a low tree-width [164]. Since $|\mathcal{T}|\le|\mathcal{B}|$ by property of chordal graphs, we deduce that for graphs with bounded tree-width, $\sum_{k\in\mathcal{T}}\mathbf{n}_k^2$ scales in $O(|\mathcal{B}|)$ rather than $O(|\mathcal{B}|^2)$. The following proposition is a standard result for sparse semidefinite programming [220], that follows from a PSD completion theorem for Hermitian matrices with chordal sparsity pattern [91].

Proposition 5.1. *The SDP problems* (**SDR**) *and* (**cSDR**) *share the same value.*

The formulation (**SDR**) is a conic programming problem on the cone $\mathcal{K} = \{W \in \mathbb{H}(\mathcal{E}) \colon \forall k \in \mathcal{T}, W_{\mathcal{B}_k,\mathcal{B}_k} \succeq 0\}$, which is not a symmetric cone [220]. For this reason, this is not a formulation that the state-of-the-art conic programming solver can handle [168]: in practice, one have to introduce a square Hermitian matrix $W_k \in \mathbb{H}(\mathcal{B}_k \times \mathcal{B}_k)$ for each $k \in \mathcal{T}$ to describe $W_{\mathcal{B}_k\mathcal{B}_k}$, and impose so-called *overlap constraints* (▲) to make sure that these matrices coincides for the pair of indices belonging to the cliques' intersections, so that the family $(W_k)_{k\in\mathcal{T}}$ indeed describes a matrix $W \in \mathbb{H}(\mathcal{E})$. This block reformulation with overlap constraints is also interesting to obtain a sparse dual problem, therefore we discuss it here.

We use additional notation regarding the tree decomposition $(\mathcal{B}_k)_{k\in\mathcal{T}}$ of the graph $\mathcal{N} = (\mathcal{B}, \mathcal{L})$. We consider a given rooting of the tree $\mathcal{T}$, and for any node k, we denote by $\mathcal{C}(k) \subset \mathcal{T}$ the set of children of k in this tree. For any $j \in \mathcal{T}$, we define the overlap $\mathcal{J}_j = \mathcal{B}_k \cap \mathcal{B}_j$, between k and its parent node k in the tree $\mathcal{T}$. By definition of a tree decomposition, for all $\ell \in \mathcal{L}$, we can find at least one $k \in \mathcal{T}$ such that $\ell \in \mathcal{B}_k \times \mathcal{B}_k$: therefore, we can partition $\mathcal{L}$ as a disjoint union $\bigcup_{k\in\mathcal{T}} \mathcal{L}_k$ such that $\mathcal{L}_k \subset \mathcal{B}_k \times \mathcal{B}_k$; We also define $\mathcal{L}_k^R$ as the reverse arcs of $\mathcal{L}_k$. Since $M_b \in \mathbb{C}^{\mathcal{E}}$, i.e. $(M_b)_{ij} = 0$ if $(i,j) \notin \mathcal{E}$, there exists a family of matrices $(M_{bk})_{k\in\mathcal{T}}$ such that $M_{bk} \in \mathbb{C}^{\mathcal{B}_k\times\mathcal{B}_k}$, and

$$M_b = \sum_{k\in\mathcal{T}} M_{bk}, \tag{5.3}$$

where M_{bk} is seen as a matrix indexed by $\mathcal{B} \times \mathcal{B}$, with zero coefficients for the indices outside $\mathcal{B}_k \times \mathcal{B}_k$. With this notation, an equivalent reformulation of (**cSDR**) is

$$\left.\begin{array}{rl}
\underset{W_k\in\mathbb{H}(\mathcal{B}_k\times\mathcal{B}_k),\, S\in\mathbb{C}^{\mathcal{G}}}{\min} & \sum_{g\in\mathcal{G}}\Big(c_{1g}\,\mathsf{Re}(S_g) + c_{2g}\,\mathsf{Re}(S_g)^2\Big) \\
\forall k \in \mathcal{T}\, \forall b \in \mathcal{B}_k & \underline{v}_b^2 \le \langle E_{bb}, W_k\rangle \le \overline{v}_b^2 \\
\forall g \in \mathcal{G} & \underline{s}_g \le S_g \le \overline{s}_g \\
\forall b \in \mathcal{B} & \sum\limits_{g\in\mathcal{G}_b} S_g - S_b^{\mathrm{d}} = \sum_{k\in\mathcal{T}} \langle M_{bk}, W_k\rangle \\
\forall k \in \mathcal{T}\, \forall \ell \in \mathcal{L}_k \cup \mathcal{L}_k^R & \langle N_\ell, W_k\rangle \le (\bar{I}_\ell)^2 \\
\forall k \in \mathcal{T}\, \forall j \in C(k) & (W_k)_{\mathcal{J}_j\mathcal{J}_j} = (W_j)_{\mathcal{J}_j\mathcal{J}_j} \quad (\blacktriangle) \\
\forall k \in \mathcal{T} & W_k \succeq 0.
\end{array}\right\} \tag{$\widetilde{\textbf{cSDR}}$}$$

The overlap constraint (▲) means that the submatrices of W_k and W_j with respect to the indices $(b,a) \in \mathcal{J}_j \times \mathcal{J}_j$ should be equal.

5.1.3 The conic programming dual of the semidefinite relaxation

In this section, we present the dual problem of the relaxation ($\widetilde{\textbf{cSDR}}$), obtained by standard SDP dualization, as opposed to the dual formulation that we propose in Section 5.2. We define the following integers

$$K_1 = \sum_{k \in \mathcal{T}} \mathbf{n}_k \qquad K_2 = \sum_{k \in \mathcal{T}} |\mathcal{J}_j| \qquad \Xi = K_1 + 2(|\mathcal{B}| + |\mathcal{L}| + K_2). \tag{5.4}$$

The standard SDP dual formulation contains Linear Matrix Inequalities (LMI) involving a family of $\mathbb{R}$-linear matrix operators $\mathcal{A}_k : \mathbb{R}^{\Xi} \to \mathbb{H}(\mathcal{B}_k \times \mathcal{B}_k)$ for $k \in \mathcal{T}$. The operator $\mathcal{A}_k$ is defined such that for all $\theta = (\alpha, \beta, \gamma, \eta, \nu, \mu) \in \mathbb{R}^{\Xi}$,

$$\mathcal{A}_k(\theta) = \sum_{b \in \mathcal{B}_k} \alpha_{bk} E_{bb} + \beta_b \mathcal{H}(M_{bk}) + \mathbf{i}\gamma_b \mathcal{Z}(M_{bk}) + \sum_{\ell \in \mathcal{L}_k \cup \mathcal{L}_k^R} \eta_\ell N_\ell \tag{5.5}$$

$$+ \sum_{(b,a) \in (\mathcal{J}_k)^2} \nu_{bak} \mathcal{H}(E_{ba}) + \mathbf{i}\mu_{bak} \mathcal{Z}(E_{ba}) - \left(\sum_{j \in \mathcal{C}(k)} \sum_{(b,a) \in (\mathcal{J}_j)^2} \nu_{baj} \mathcal{H}(E_{ba}) + \mathbf{i}\mu_{baj} \mathcal{Z}(E_{ba}) \right), \tag{5.6}$$

with $\mathcal{H}(M) = \frac{M + M^{\mathsf{H}}}{2}$ (Hermitian part) and $\mathcal{Z}(M) = \frac{M - M^{\mathsf{H}}}{2}$ (skew-Hermitian part). We define the sets $\mathcal{G}_1 = \{g \in \mathcal{G} \colon c_{2g} = 0\}$ and $\mathcal{G}_2 = \mathcal{G} \setminus \mathcal{G}_1$. For any $g \in \mathcal{G}$, $b(g) \in \mathcal{B}$ is the bus where the generator g is located. We can now introduce the formulation

$$\left.\begin{array}{ll}
\max & \sum\limits_{k \in \mathcal{T},\, b \in \mathcal{B}_k} \underline{v}_b^2 \underline{\alpha}_{bk} - \overline{v}_b^2 \overline{\alpha}_{bk} + \sum\limits_{b \in \mathcal{B}} P_b^d \beta_b + Q_b^d \gamma_b - \sum\limits_{\ell \in \mathcal{L} \cup \mathcal{L}^R} \overline{I}_\ell^2 \eta_\ell \\
& \quad + \sum\limits_{g \in \mathcal{G}} \underline{P}_g \underline{y}_g - \overline{P}_g \overline{y}_g + \underline{Q}_g \underline{z}_g - \overline{Q}_g \overline{z}_g - \sum\limits_{g \in \mathcal{G}_2} \frac{(\overline{y}_g - \underline{y}_g - \beta_{b(g)} + c_{1g})^2}{4c_{2g}} \\
\text{s.t.} & \\
\forall g \in \mathcal{G}_1 & \overline{y}_g - \underline{y}_g = \beta_{b(g)} - c_{1g} \\
\forall g \in \mathcal{G} & \overline{z}_g - \underline{z}_g = \gamma_{b(g)} \\
\forall k \in \mathcal{T} & \mathcal{A}_k(\overline{\alpha} - \underline{\alpha}, \beta, \gamma, \eta, \nu, \mu) \succeq 0 \\
& (\underline{y}, \overline{y}, \underline{z}, \overline{z}) \in \mathbb{R}_+^{4|\mathcal{G}|} \\
& (\underline{\alpha}, \overline{\alpha}) \in \mathbb{R}_+^{2K_1} \\
& (\beta, \gamma, \eta, \nu, \mu) \in \mathbb{R}^{2(|\mathcal{B}| + |\mathcal{L}| + K_2)}.
\end{array}\right\} \quad (\textbf{dualcSDR})$$

We recall that $\underline{P}_g$ and $\underline{Q}_g$ (resp. $\overline{P}_g$ and $\overline{Q}_g$) were introduced as the real and imaginary parts of $\underline{s}_g$ (resp. $\overline{s}_g$), and P_b^{d} and Q_b^{d} as the real and imaginary parts of S_b^{d}.

Proposition 5.2. *The constrained maximization problem* (**dualcSDR**) *is the dual conic programming problem of the relaxation* ($\widetilde{\textbf{cSDR}}$).

Proof. We start by writing the formulation ($\widetilde{\textbf{cSDR}}$) only with real equalities and inequalities, so has to introduce the corresponding dual variables. To do so, we notice that for $A, M \in \mathbb{C}^{\mathcal{B} \times \mathcal{B}}$ such that $M = M^{\mathsf{H}}$, the following holds [81]

$$\mathsf{Re}(\langle A, M \rangle) = \langle \mathcal{H}(A), M \rangle \tag{5.7}$$

$$\mathsf{Im}(\langle A, M \rangle) = \langle \mathbf{i}\mathcal{Z}(A), M \rangle \,. \tag{5.8}$$

We underline that the matrices $\mathcal{H}(A)$ and $\mathbf{i}\mathcal{Z}(A)$ are indeed Hermitian, so the scalar products $\langle \mathcal{H}(A), M\rangle$ and $\langle \mathbf{i}\mathcal{Z}(A), M\rangle$ are indeed real. Based on this observation, we observe that the following problem is a reformulation of ($\mathbf{c\widetilde{SDR}}$)

$$\left.\begin{array}{lll}
\min\limits_{W_k\in\mathbb{H}(\mathcal{B}_k\times\mathcal{B}_k),\ P,Q\in\mathbb{R}^{\mathcal{G}}} & \sum_{g\in\mathcal{G}}\left(c_{1g}\,P_g + c_{2g}\,P_g^2\right) & \\
\forall g\in\mathcal{G} & \underline{P}_g \le P_g \le \overline{P}_g & (\underline{y}_g, \overline{y_g}) \\
\forall g\in\mathcal{G} & \underline{Q}_g \le Q_g \le \overline{Q}_g & (\underline{z}_g, \overline{z_g}) \\
\forall k\in\mathcal{T}\,\forall b\in\mathcal{B}_k & \underline{v}_b^2 \le \langle E_{bb}, W_k\rangle \le \overline{v}_b^2 & (\underline{\alpha}_{bk}, \overline{\alpha}_{bk}) \\
\forall b\in\mathcal{B} & \sum\limits_{g\in\mathcal{G}_b} P_g - P_b^{\mathsf{d}} = \sum_{k\in\mathcal{T}}\langle \mathcal{H}(M_{bk}), W_k\rangle & (\beta_b) \\
\forall b\in\mathcal{B} & \sum\limits_{g\in\mathcal{G}_b} Q_g - Q_b^{\mathsf{d}} = \sum_{k\in\mathcal{T}}\langle \mathbf{i}\mathcal{Z}(M_{bk}), W_k\rangle & (\gamma_b) \\
\forall k\in\mathcal{T}\,\forall \ell\in\mathcal{L}_k\cup\mathcal{L}_k^R & \langle N_\ell, W_k\rangle \le (\bar{I}_\ell)^2 & (\eta_\ell) \\
\forall k\in\mathcal{T}\,\forall j\in C(k)\,\forall (b,a)\in(\mathcal{J}_j)^2 & \langle \mathcal{H}(E_{ba}), W_k\rangle = \langle \mathcal{H}(E_{ba}), W_j\rangle & (\nu_{baj}) \\
\forall k\in\mathcal{T}\,\forall j\in C(k)\,\forall (b,a)\in(\mathcal{J}_j)^2 & \langle \mathbf{i}\mathcal{Z}(E_{ba}), W_k\rangle = \langle \mathbf{i}\mathcal{Z}(E_{ba}), W_j\rangle & (\mu_{baj}) \\
\forall k\in\mathcal{T} & W_k \succeq 0. &
\end{array}\right\} \quad (5.9)$$

In Eq. (5.9), we display the dual variable of each constraint. The formulation (**dualcSDR**) is then obtained by standard conic programming dualization. In (**dualcSDR**) the constraint $\overline{y}_g - \underline{y}_g = \beta_{b(g)} - c_{1g}$ corresponds to the primal variable P_g, the constraint $\overline{z}_g - \underline{z}_g = \gamma_{b(g)}$ to the primal variable Q_g, and $\mathcal{A}_k(\overline{\alpha} - \underline{\alpha}, \beta, \gamma, \eta, \nu, \mu) \succeq 0$ to the primal variable W_k. □

5.2 Unconstrained spectral formulation for the dual problem

This section introduces the central element of our approach, i.e. an unconstrained formulation for the dual problem of the ACOPF.

5.2.1 Some concave functions of interest

To reformulate (**dualcSDR**) as an unconstrained problem, we introduce some basic functions that are terms in the objective function of (**dualcSDR**) or penalizations of the constraints: For each generator $g\in\mathcal{G}$, we introduce the concave functions p_g, q_g such that for all $x\in\mathbb{R}$,

$$p_g(x) = \min_{P\in[\underline{P}_g, \overline{P}_g]} (c_{1g} - x)P + c_{2g}\,P^2 \quad (5.10)$$

$$q_g(x) = \min_{Q\in[\underline{Q}_g, \overline{Q}_g]} -xQ = \underline{Q}_g\, x^- - \overline{Q}_g\, x^+. \quad (5.11)$$

To obtain an explicit formula for the function $p_g(x)$, we have to distinguish two cases:

- If $g\in\mathcal{G}_1$, then $p_g(x) = \underline{P}_g(x - c_{1g})^- - \overline{P}_g(x - c_{1g})^+$,
- If $g\in\mathcal{G}_2$, then $p_g(x) = \begin{cases} (c_{1g} - x)\underline{P}_g + c_{2g}\,\underline{P}_g^2 & \text{if } x - c_{1g} \le 2c_{2g}\underline{P}_g, \\ -\frac{(x-c_{1g})^2}{4c_{2g}} & \text{if } x - c_{1g} \in [\,2c_{2g}\underline{P}_g, 2c_{2g}\overline{P}_g\,], \\ (c_{1g} - x)\overline{P}_g + c_{2g}\,\overline{P}_g^2 & \text{if } x - c_{1g} \ge 2c_{2g}\overline{P}_g. \end{cases}$

For each bus $b \in \mathcal{B}$, we define the concave functions v_b, p_b, q_b such that for all $x \in \mathbb{R}$,

$$v_b(x) = \min_{v \in [\underline{v}_b^2, \overline{v}_b^2]} -xv = \underline{v}_b^2\, x^- - \overline{v_b}^2\, x^+, \tag{5.12}$$

$$p_b(x) = P_b^{\mathsf{d}}\, x + \sum_{g \in \mathcal{G}_b} p_g(x), \tag{5.13}$$

$$q_b(x) = Q_b^{\mathsf{d}}\, x + \sum_{g \in \mathcal{G}_b} q_g(x). \tag{5.14}$$

For each $\ell \in \mathcal{L} \cup \mathcal{L}^R$, we define the function h_ℓ such that for all $x \in \mathbb{R}$,

$$h_\ell(x) = \min_{I \in [0, \overline{I}_\ell^2]} -Ix = -\overline{I}_\ell^2\, x^+. \tag{5.15}$$

Finally, we introduce for each $k \in \mathcal{T}$ the concave multivariate function f_k, such that for all $\theta = (\alpha, \beta, \gamma, \eta, \nu, \mu) \in \mathbb{R}^\Xi$,

$$f_k(\theta) = \min_{\substack{Z \in \mathbb{H}(\mathcal{B}_k \times \mathcal{B}_k) \\ Z \succeq 0, \mathsf{Tr}(Z) \leq \rho_k}} \langle \mathcal{A}_k(\theta), Z \rangle = \min \left\{ \rho_k\, \lambda_{\min}(\mathcal{A}_k(\theta)), 0 \right\}, \tag{5.16}$$

where $\rho_k = \sum_{b \in \mathcal{B}_k} \overline{v_b}^2$. The linear operator $\mathcal{A}_k$, defined in Eqs. (5.5)-(5.6), only depends on variables related to the clique index k. Therefore, so does function f_k, as shown Table 5.2.

5.2.2 Our dual formulation

We define the function $F : \mathbb{R}^\Xi \to \mathbb{R}$, such that for all $\theta = (\alpha, \beta, \gamma, \eta, \nu, \mu) \in \mathbb{R}^\Xi$

$$F(\theta) = \sum_{k \in \mathcal{T}} f_k(\theta) + \sum_{k \in \mathcal{T}} \sum_{b \in \mathcal{B}_k} v_b(\alpha_{bk}) + \sum_{b \in \mathcal{B}} p_b(\beta_b) + \sum_{b \in \mathcal{B}} q_b(\gamma_b) + \sum_{\ell \in \mathcal{L} \cup \mathcal{L}^R} h_\ell(\eta_\ell), \tag{5.17}$$

and we propose the following formulation:

$$\begin{aligned} \max \quad & F(\theta) \\ \text{s.t.} \quad & \theta \in \mathbb{R}^\Xi. \end{aligned} \tag{dualOPF}$$

Since the objective function F is a sum of functions, each of them involving just one or a limited number of variables, problem (**dualOPF**) is said partially separable.

Theorem 5.1. *The unconstrained maximization problem* (**dualOPF**) *has the same value as the constrained problems* (**SDR**), (**cSDR**),($\widetilde{\textbf{cSDR}}$) *and* (**dualcSDR**).

Proof. We define $\mathbb{X} = \{(W_k)_{k \in \mathcal{T}} \mid \forall k \in \mathcal{T},\ (W_k \in \mathbb{H}(\mathcal{B}_k \times \mathcal{B}_k)) \wedge (W_k \succeq 0) \wedge (\mathsf{Tr}(W_k) \leq \rho_k)\}$. We underline that the constraint $\mathsf{Tr}(W_k) \leq \rho_k$ is redundant in ($\widetilde{\textbf{cSDR}}$) since it follows from the constraints $\forall b \in \mathcal{B}_k, \langle E_{bb}, W_k \rangle \leq \overline{v_b}^2$, but it plays a role in the derivation of the dual formulation. The dualization of ($\widetilde{\textbf{cSDR}}$) introduced here differs from the standard conic programming dualization presented in Proposition 5.2 as: (i) we keep explictly the constraint $\mathbf{W} \in \mathbb{X}$ (ii) we do not introduce dual variables for the bound constraints on S_g, but we also keep the constraint explicitly (iii) as regards the inequality constraints $\underline{v}_b^2 \leq \langle E_{bb}, W_k \rangle \leq \overline{v}_b^2$, we encode it as $\langle E_{bb}, W_k \rangle = u_{bk}$, with u_{bk} constrained in $[\underline{v}_b^2, \overline{v}_b^2]$, and we introduce a dual variable α_{bk} for the equality constraint but not for the bound constraints, that are kept explicit. We proceed similarly for the line constraints, introducing the variable $u_\ell \in [0, \bar{I}_\ell^2]$. Therefore, we define the Lagrangian L such that for any primal vectors $S \in \mathbb{C}^{\mathcal{G}}$, $\mathbf{W} \in \mathbb{X}$, $u \in \mathbb{R}^{K_1+2|\mathcal{L}|}$ and for any dual vector $\theta = (\alpha, \beta, \gamma, \eta, \nu, \mu) \in \mathbb{R}^{\Xi}$,

$$\begin{aligned}
L(S, \mathbf{W}, u, \theta) = &\sum_{g\in\mathcal{G}} c_{1g}\,\mathsf{Re}(S_g) + c_{2g}\,\mathsf{Re}(S_g)^2 \\
&+ \sum_{k\in\mathcal{T}} \sum_{b\in\mathcal{B}_k} \alpha_{bk}\big(\langle E_{bb}, W_k\rangle - u_{bk}\big) \\
&+ \sum_{b\in\mathcal{B}} \beta_b\big(P_b^{\mathsf{d}} + \sum_{k\in\mathcal{T}} \langle \mathcal{H}(M_{bk}), W_k\rangle - \sum_{g\in\mathcal{G}_b} \mathsf{Re}(S_g)\big) + \sum_{b\in\mathcal{B}} \gamma_b\big(Q_b^{\mathsf{d}} + \sum_{k\in\mathcal{T}} \langle \mathbf{i}\mathcal{Z}(M_{bk}), W_k\rangle - \sum_{g\in\mathcal{G}_b} \mathsf{Im}(S_g)\big) \\
&+ \sum_{k\in\mathcal{T}} \sum_{\ell\in\mathcal{L}_k\cup\mathcal{L}_k^R} \eta_\ell\big(\langle N_\ell, W_k\rangle - u_\ell\big) \\
&+ \sum_{k\in\mathcal{T}} \sum_{j\in\mathcal{C}(k)} \sum_{(b,a)\in(\mathcal{J}_j)^2} \nu_{baj}\big(\langle \mathcal{H}(E_{ba}), W_j\rangle - \langle \mathcal{H}(E_{ba}), W_k\rangle\big) \\
&+ \sum_{k\in\mathcal{T}} \sum_{j\in\mathcal{C}(k)} \sum_{(b,a)\in(\mathcal{J}_j)^2} \mu_{baj}\big(\langle \mathbf{i}\mathcal{Z}(E_{ba}), W_j\rangle - \langle \mathbf{i}\mathcal{Z}(E_{ba}), W_k\rangle\big).
\end{aligned}$$

Then, the problem ($\widetilde{\mathbf{cSDR}}$) reads as follows:

$$\min_{S,\mathbf{W},u} \ \max_{\theta} \ L(S, \mathbf{W}, u, \theta),$$

where the minimization is over the constraints $S \in \prod_g \, [\underline{s}_g, \overline{s}_g]_{\mathbb{C}}$, $\mathbf{W} \in \mathbb{X}$, $u \in \prod_{bk}[\underline{v}_b^2, \overline{v_b}^2] \times \prod_\ell [0, \bar{I}_\ell^2]$, and the maximization over $\theta \in \mathbb{R}^{\Xi}$. Then, we reorder the terms of the Lagrangian L:

$$\begin{aligned}
L(S, \mathbf{W}, u, \theta) = &-\sum_{k\in\mathcal{T}} \sum_{b\in\mathcal{B}_k} \alpha_{bk} u_{bk} + \sum_{b\in\mathcal{B}} \beta_b P_b^{\mathsf{d}} + \sum_{b\in\mathcal{B}} \gamma_b Q_b^{\mathsf{d}} \\
&+ \sum_{b\in\mathcal{B}} \sum_{g\in\mathcal{G}_b} (c_{1g} - \beta_b)\mathsf{Re}(S_g) + c_{2g}\,\mathsf{Re}(S_g)^2 \\
&- \sum_{b\in\mathcal{B}} \sum_{g\in\mathcal{G}_b} \gamma_b \mathsf{Im}(S_g) \\
&- \sum_{k\in\mathcal{T}} \sum_{\ell\in\mathcal{L}_k\cup\mathcal{L}_k^R} \eta_\ell u_\ell \\
&+ \sum_{k\in\mathcal{T}} \langle \mathcal{A}_k(\theta), W_k\rangle\,.
\end{aligned}$$

Hence, for fixed $\theta \in \mathbb{R}^{\Xi}$, the minimum over $S \in \prod_g \, [\underline{s}_g, \overline{s}_g]_{\mathbb{C}}$, $\mathbf{W} \in \mathbb{X}$, $u \in \prod_{bk}[\underline{v}_b^2, \overline{v_b}^2] \times \prod_\ell [0, \bar{I}_\ell^2]$

is

$$\begin{aligned}
\min_{S,\mathbf{W},u} L(S,\mathbf{W},u,\theta) &= \sum_{k\in\mathcal{T}}\sum_{b\in\mathcal{B}_k} \min_{v\in[\underline{v}_b^2,\overline{v_b}^2]} -\alpha_{bk}v + \sum_{b\in\mathcal{B}}\beta_b P_b^{\mathsf{d}} + \sum_{b\in\mathcal{B}}\gamma_b Q_b^{\mathsf{d}} \\
&\quad + \sum_{b\in\mathcal{B}}\sum_{g\in\mathcal{G}_b} \min_{p\in[\underline{P}_g,\overline{P}_g]} (c_{1g}-\beta_b)p + c_{2g}\,p^2 + \min_{q\in[\underline{Q}_g,\overline{Q}_g]} -\gamma_b q \\
&\quad + \sum_{k\in\mathcal{T}}\sum_{\ell\in\mathcal{L}_k\cup\mathcal{L}_k^R} \min_{I\in[0,\bar{I}_\ell^2]} -\eta_\ell I + \sum_{k\in\mathcal{T}} \min_{\substack{Z\in\mathbb{H}(\mathcal{B}_k\times\mathcal{B}_k)\\ Z\succeq 0,\mathsf{Tr}(Z)\le\rho_k}} \langle\mathcal{A}_k(\theta),Z\rangle \\
&= \sum_{k\in\mathcal{T}}\sum_{b\in\mathcal{B}_k} v_b(\alpha_{bk}) + \sum_{b\in\mathcal{B}}\beta_b P_b^{\mathsf{d}} + \sum_{b\in\mathcal{B}}\gamma_b Q_b^{\mathsf{d}} \\
&\quad + \sum_{b\in\mathcal{B}}\sum_{g\in\mathcal{G}_b} p_g(\beta_b) + q_g(\gamma_b) \\
&\quad + \sum_{k\in\mathcal{T}}\sum_{\ell\in\mathcal{L}_k\cup\mathcal{L}_k^R} h_\ell(\eta_l) + \sum_{k\in\mathcal{T}} f_k(\theta) \\
&= F(\theta).
\end{aligned}$$

We deduce that the max-min problem $\max_{\theta}\min_{S,\mathbf{W},u} L(S,\mathbf{W},u,\theta)$ is exactly the formulation (**dualOPF**). Since (i) the Lagrangian L is convex with respect to primal variables and linear with respect to dual variables (ii) both minimization and maximization sets are convex (iii) the minimization set is compact; we can apply Sion min-max theorem [202] and deduce that the two values are equals, hence

$$\mathsf{val}(\mathbf{c}\widetilde{\mathbf{SDR}}) = \mathsf{val}(\mathbf{dualOPF}). \tag{5.18}$$

As ($\mathbf{c}\widetilde{\mathbf{SDR}}$) is a reformulation of (**cSDR**), which has the same value as (**SDR**) (see Proposition 5.1), we know that (**dualOPF**) has the same value as these primal relaxations. It remains to prove that (**dualcSDR**) also share this value. By duality, we know that

$$\mathsf{val}(\mathbf{c}\widetilde{\mathbf{SDR}}) \geq \mathsf{val}(\mathbf{dualcSDR}). \tag{5.19}$$

Moreover, we notice that from any solution $\theta = (\alpha,\beta,\gamma,\eta,\nu,\mu)$ with value $F(\theta)$ in (**dualOPF**), we build a solution $Z = (\underline{y},\overline{y},\underline{z},\overline{z},\theta)$ of (**dualcSDR**) with the same value, by defining

- for all $g \in \mathcal{G}$, $\underline{z}_g = (\gamma_{b(g)})^-$, $\overline{z}_g = (\gamma_{b(g)})^+$,
- for all $g \in \mathcal{G}_1$, $\underline{y}_g = (\beta_{b(g)} - c_{1g})^-$, $\overline{y}_g = (\beta_{b(g)} - c_{1g})^+$,
- for all $g \in \mathcal{G}_2$, $\underline{y}_g = \max\{0, c_{1g} - \beta + 2c_{2g}\underline{P}_g\}$ and $\overline{y}_g = \max\{0, \beta - c_{1g} - 2c_{2g}\overline{P}_g\}$.

Therefore, val(**dualcSDR**) $\geq$ val(**dualOPF**). Together with Eqs. (5.18)-(5.19), we deduce that (**dualcSDR**) $=$ val(**dualOPF**). □

5.2.3 Lower bound certification

In many applications, computing a lower bound on an optimization problem is the first motivation of solving a relaxation of this problem. We will now present how our dual formulation (**dualOPF**) enables us to compute certified lower bounds on the value of (ACOPF). We consider a realistic computational framework in rational numbers. The parameters of problem (ACOPF) are thus supposed to have rational real and imaginary parts. We consider the question of computing a certified lower bound, that is, computing a value $v \in \mathbb{Q}$ such that $\mathsf{val}(\text{ACOPF}) \geq v$. A disadvantage of solving the primal-dual pair (**cSDR**), (**dualcSDR**) of SDP problems is that obtaining a certified dual bound requires the exact feasibility of the dual solution. In practice, one may obtain a slightly infeasible solution of (**dualcSDR**) which invalidates the certification.

The unconstrained formulation (**dualOPF**) helps addressing this issue. We take any vector $\theta \in \mathbb{Q}^N$, produced by any algorithm maximizing $F(\theta)$. The inequality $\mathsf{val}(\text{ACOPF}) \geq F(\theta)$ holds. Nevertheless, we have to acknowledge that, because of the eigenvalue functions in $f_k(\theta)$ terms, the value of $F(\theta)$ can only be approximated. Despite this, we need to compute an exact lower bound on $f_k(\theta)$, so as to compute an exact lower bound on $F(\theta)$. For any $M \in \mathbb{H}_n$ with rational coefficients, we define $\lambda_{\mathsf{LB}}(M) = \min_{1\leq i\leq n} M_{ii} - \sum_{j\neq i} |M_{ij}|$, which is a lower bound on $\lambda_{\min}(M)$ due to Gershgorin's circle theorem. We point out that computing the magnitude $|M_{ij}| = \sqrt{\mathsf{Re}(M_{ij})^2 + \mathsf{Im}(M_{ij})^2}$ involves computing a square root, which is not necessarily rational. Yet, we can always round up the output of the square root algorithm to an arbitrary precision, to obtain $s \in \mathbb{Q}$, and check that $s^2 \geq \mathsf{Re}(M_{ij})^2 + \mathsf{Im}(M_{ij})^2$. Such an upper bound on the square root will guarantee that the computed value $\underline{\lambda}_{\mathsf{LB}}(M) \in \mathbb{Q}$ is lower than $\lambda_{\mathsf{LB}}(M)$ and thus than $\lambda_{\min}(M)$. By eigendecomposition, we compute a matrix U and a diagonal matrix D, both with rational coefficients and such that $\mathcal{A}_k(\theta) \approx UDU^{\mathsf{H}}$. We define then $\underline{f_k}(\theta|U,D) = \rho_k \min\{\min_i(D_{ii}) + \underline{\lambda}_{\mathsf{LB}}(\mathcal{A}_k(\theta) - UDU^{\mathsf{H}}), 0\}$, which is computable through basic arithmetic operations in rational numbers and

$$f_k(\theta) \geq \underline{f_k}(\theta|U,D). \tag{5.20}$$

Replacing $f_k(\theta)$ by $\underline{f_k}(\theta|U,D)$ in the expression of $F(\theta)$, we can compute an *exact* lower bound on $\mathsf{val}(\text{ACOPF})$. Indeed, the functions p_b, q_b, v_b, h_ℓ are computable in rational numbers since they involve elementary arithmetic operations.

5.3 Nonsmooth optimization algorithm

The objective function of (**dualOPF**) being nonsmooth, we solve this new formulation for the dual ACOPF with a tailored Nonsmooth Optimization (NSO) algorithm belonging to the family of Proximal Bundle Methods (PBM) [12].

5.3.1 Abstract formulation

In the definition of the dual objective function $F(\theta)$, introduced in Eq. (5.17), it appears clearly that this is a sum of functions

$$F(\theta) = \sum_{j=1}^{r} F_j(\theta), \tag{5.21}$$

where $F_j(\theta) = \min_{(a,b)\in\mathcal{Y}_j} a + b^\top\theta$ for a given compact set $\mathcal{Y}_j \subset \mathbb{R}^{1+\Xi}$. In Table (5.1), we detail what are these sets for the different terms appearing in the sum Eq. (5.17). For the understanding of this table, we remind that the variables $\alpha_{bk}, \beta_b, \gamma_b$, etc., must be understood as specific coordinates of θ; we denote $e_{\alpha bk} \in \mathbb{R}^\Xi$ (resp. $e_{\beta b}$, $e_{\gamma b}$, etc.) the element of the canonical basis such that $\theta^\top e_{\alpha bk} = \alpha_{bk}$ (resp. $\theta^\top e_{\beta b} = \beta_b$, $\theta^\top e_{\gamma b} = \gamma_b$, etc.). The notation $\mathcal{A}_k^\dagger$ designates the adjoint operator to $\mathcal{A}_k$, and is made explicit in Table 5.2.

Function $F_j(\theta)$	Set $\mathcal{Y}_j$
$v_b(\alpha_{bk})$	$\{(0, -ve_{\alpha bk}) : v \in [\underline{v}_b^2, \overline{v}_b^2]\}$
$p_g(\beta_b)$	$\{(c_{1g}p + c_{2g}p^2, -pe_{\beta b}) : p \in [\underline{P}_g, \overline{P}_g]\}$
$q_g(\gamma_b)$	$\{(0, -qe_{\gamma b}) : q \in [\underline{Q}_g, \overline{Q}_g]\}$
$p_b(\beta_b)$	$\{(0, P_b^{\mathsf{d}} e_{\beta b})\}$
$q_b(\gamma_b)$	$\{(0, Q_b^{\mathsf{d}} e_{\gamma b})\}$
$h_\ell(\eta_\ell)$	$\{(0, -Ie_{\eta\ell}) : I \in [0, \overline{I}_\ell^2]\}$
$f_k(\theta)$	$\{(0, \mathcal{A}_k^\dagger(W)) : U \succeq 0, \mathsf{Tr}(W) \le \rho_k\}$

Table 5.1: Description of the functions as $F_j(\theta) = \min_{(a,b)\in\mathcal{Y}_j} a + b^\top\theta$

Note that for all function types except $f_k(\theta)$, the set $\mathcal{Y}_j$ is of dimension 0, 1, or 2. The computation of $\min_{(a,b)\in\mathcal{Y}_j} a + b^\top\theta$ can be done in closed form: the separation oracle is simple and exact. For the functions of the type $f_k(\theta)$, an exact separation oracle is also affordable, since the minimum is computed as follows: first, we compute the minimum eigenvalue λ of $\mathcal{A}_k(\theta)$ and an associated eigenvector U; if $\lambda > 0$, $f_k(\theta) = 0$ and the null vector is a solution to this minimization problem; otherwise $f_k(\theta) = \rho_k\lambda$ and $(0, \mathcal{A}_k^\dagger(W))$ is solution, where $W = \rho_k UU^{\mathsf{H}}$. We emphasize that the computational cost of this operation is limited since

- computing (λ, U) amounts to compute a minimum eigenpair of a $\mathbf{n}_k \times \mathbf{n}_k$ matrix, since $\mathcal{A}_k(\theta) \in \mathbb{H}(\mathcal{B}_k \times \mathcal{B}_k)$,
- as shown in Table 5.2, computing the real vector $\mathcal{A}_k^\dagger(W)$ amounts to compute Froebenius products involving matrices with very few non-zero coefficients, all related to clique $\mathcal{B}_k$.

Remark 5.1. *The formulation (**dualOPF**) has an equivalent semi-infinite reformulation:*

$$\left.\begin{array}{rl} \max\limits_{\theta\in\mathbb{R}^\Xi, s\in\mathbb{R}^r} & \sum_{j=1}^r s_j \\ s.t. & \forall j \in [\![1, r]\!],\ \forall (a,b) \in \mathcal{Y}_j,\ s_j \le a + b^\top\theta. \end{array}\right\} \tag{5.22}$$

Variable	Corresponding coordinate of $\mathcal{A}_k^\dagger(W)$
α_{bk}, for $b \in \mathcal{B}_k$	$\langle E_{bb}, W\rangle$
β_b, γ_b, for $b \in \mathcal{B}_k$	$\langle \mathcal{H}(M_{bk}), W\rangle$, $\langle \mathbf{i}\mathcal{Z}(M_{bk}), W\rangle$
η_ℓ, for $\ell \in \mathcal{L}_k \cup \mathcal{L}_k^R$	$\langle N_\ell, W\rangle$
ν_{bak}, μ_{bak} for $(b,a) \in (\mathcal{J}_k)^2$	$\langle \mathcal{H}(E_{ba}), W\rangle$,$\langle \mathbf{i}\mathcal{Z}(E_{ba}), W\rangle$
ν_{bad}, μ_{bad} for $d \in \mathcal{C}(k)$ and for $(b,a) \in (\mathcal{J}_d)^2$	$-\langle \mathcal{H}(E_{ba}), W\rangle$,$-\langle \mathbf{i}\mathcal{Z}(E_{ba}), W\rangle$

Table 5.2: Description of the nonzero coordinates of $\mathcal{A}_k^\dagger(W)$

5.3.2 A structured cutting-plane model

When designing a PBM, a key choice is the cutting-plane models used to approximate the objective function. In the present case, the objective function $F(\theta)$ of (**dualOPF**) has a partially separable structure. We exploit this property to produce a rich and structured cutting-plane model, sometimes referred as *disaggregated* cutting-plane model in the NSO literature. Instead of using a cutting-plane model for the whole function $F(\theta)$, we use one model for each single function $F_j(\theta)$. For a given finite subset $\check{\mathcal{Y}}_j \subset \mathcal{Y}_j$, we define the polyhedral function

$$\check{F}_j(\theta) = \min_{(a,b)\in\check{\mathcal{Y}}_j} a + b^\top\theta. \tag{5.23}$$

During an iterative algorithm, the set $\check{\mathcal{Y}}_j$ might evolve: therefore, we denote by $\check{\mathcal{Y}}_j^t$ at the iteration t, and by $\check{F}_j(\theta)$ the resulting function. Then, we define the disaggregated polyhedral model

$$\check{F}^t(\theta) = \sum_{j=1}^{r} \check{F}_j^t(\theta). \tag{5.24}$$

The algorithm CPA, studied in Chapter 1, and also known as Kelley's algorithm [118], could be a possible approach here to solve the formulation (**dualOPF**) based on the model (5.24): in other words, this would correspond to applying CPA to the semi-infinite programming formulation (5.22). However, the assumptions of Chapter 1, in the setting of which we proved that the algorithm CPA performs well, are not satisfied here: (i) we do not have an *a priori* set $\mathcal{X}$ that is compact to bound θ, therefore $\max_\theta \check{F}^t(\theta)$ may not be finite (ii) the objective function $F(\theta)$ is not strongly concave. Therefore, the performance of CPA may be poor. Hence, we prefer to use a proximal bundle method.

5.3.3 The proximal bundle method

A proximal bundle method (PBM) may be seen as a stabilized version of the algorithm CPA: based on stability centers $\bar{\theta}^t$, a quadratic regularization is added to the objective function to make it strongly concave. Algorithm 5 presents the PBM framework used to solve (**dualOPF**).

Algorithm 5 Proximal bundle method.

```
    Input: $\bar{\theta}^0 \in \mathbb{R}^d$; finite sets $\check{F}_j^0 \subset F_j$; $m \in [0,1]$; $tol, \kappa_0 \in \mathbb{R}_{++}$.
 0: $t \leftarrow 0$, $\delta_0 \leftarrow \infty$
 1: while $\delta_t > tol$ do
 2:     Let $\theta^t$ be a solution of
            $$\max_{\theta \in \mathbb{R}^N} \check{F}^t(\theta) - \frac{1}{2}\kappa_t \|\theta - \bar{\theta}^t\|^2. \qquad (\mathbf{QP}_t)$$
 3:     For all $j \in [\![1,r]\!]$, compute $F_j(\theta^t)$ and $(a_j, b_j) \in \mathcal{Y}_j$ such that $F_j(\theta^t) = a_j + (b_j)^\top \theta^t$.
 4:     $F(\theta^t) \leftarrow \sum_{j=1}^r F_j(\theta^t)$
 5:     $\delta_t \leftarrow F(\bar{\theta}^t) - \check{F}^t(\theta^t) + \frac{1}{2}\kappa_t \|\theta^t - \bar{\theta}^t\|^2$.
 6:     if $F(\theta^t) \leq F(\bar{\theta}^t) - m\delta_t$ then          ▷ Serious step
 7:         $\bar{\theta}^{t+1} \leftarrow \theta^t$.
 8:     else                                                              ▷ Null step
 9:         $\bar{\theta}^{t+1} \leftarrow \bar{\theta}^t$.
10:     end if
11:     For all $j \in [\![1,r]\!]$, $\check{\mathcal{Y}}_j^{t+1} \leftarrow \check{\mathcal{Y}}_j^t \cup \{(a_j, b_j)\}$. Create the model $\check{F}^{t+1}$ accordingly.
12:     Update $\kappa_t$ to build $\kappa_{t+1}$.
13:     $t \leftarrow t + 1$.
14: end while
```

The function $\check{F}^t$ being polyhedral; the subproblem solved at each iteration is a convex QP problem. An explicit formulation of this QP problem is

$$\left.\begin{array}{rl} \max\limits_{\theta \in \mathbb{R}^{\Xi}, s \in \mathbb{R}^r} & \sum_{j=1}^r s_j - \frac{1}{2}\kappa_t \|\theta - \bar{\theta}^t\|^2 \\ \text{s.t.} & \forall j \in [\![1,r]\!],\ \forall (a,b) \in \check{\mathcal{Y}}_j^t,\ s_j \leq a + b^\top \theta, \end{array}\right\} \qquad (\mathbf{QP}_t)$$

with the sets $\check{\mathcal{Y}}_j^t$ being finite. The framework presented in Algorithm 5 must be specified in different ways. First, we stop the algorithm after a maximal number of iterations $K_{\max}$ or a maximal number of consecutive null steps $K_{\max}^{\mathsf{null}}$, or whenever $F(\theta^t)$ is greater than a known upper bound on val(ACOPF) (primal infeasibility). Second, we update the proximal parameter $\kappa \in [\kappa_{\min}, \kappa_{\max}]$, only when: (a) 2 serious steps are consecutive or separated by at most 1 null step, then $\kappa_{t+1} \leftarrow \max\{r_{\mathsf{down}}\kappa_t, \kappa_{\min}\}$ with $r_{\mathsf{down}} \in]0,1[$; (b) every batch of 10 consecutive null steps, $\kappa_{t+1} \leftarrow \min\{r_{\mathsf{up}}\kappa_t, \kappa_{\max}\}$ with $r_{\mathsf{up}} > 1$. Third, we keep bounded-size cutting-plane models, as explained in Section 1.2, by (i) deleting inactive cutting-plane every 5 serious steps (ii) aggregating each individual cutting-plane model $\check{F}_j^t(\theta)$ based on the QP dual solution every 10 serious steps.

Remark 5.2. *We notice that during a sequence of consecutive null steps, the stability center $\bar{\theta}^t$ is constant. We denote it $\bar{\theta}$. If we were to choose a constant parameter $\kappa_t = \kappa$, we notice that these consecutive null steps are performing the Algorithm 1 (CPA) to solve the proximal point problem $\max_{\theta \in \mathbb{R}^{\Xi}} F(\theta) - \frac{\kappa}{2}\|\theta - \bar{\theta}\|^2$. However, it is known that it is often more effective in practice to adapt the proximal parameter κ dynamically [140].*

5.4 Numerical experiments

In this section we illustrate that the proposed formulation and the PBM can be used to obtain accurate and certified dual bounds on val(ACOPF). We do not use this algorithm as a standalone solver but as a *post-processing* step after calling the IP solver MOSEK [168]; Tables 5.3 and 5.4 show that this post-processing enables substantial accuracy improvements for several instances from the library IEEE PES PGLib AC-OPF v21.07 [9].

5.4.1 Experimental setting

For all experiments, we used a 64-bit Ubuntu computer with 32 Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz and 32 GB RAM. Algorithm 1 was implemented in `Python 3.6`. We compute the clique decomposition thanks to the `chompack` package. To manipulate sparse matrices and solve eigenproblems, we use the scientific packages `Scipy` and `numpy`. At each iteration, we solve the QP problem in its dual form, using the open-source solver `OSQP` [206]. We execute Algorithm 5 with an Armijo parameter $m = 0.01$ and a relative tolerance parameter of $tol = 10^{-6}$. We also set $K_{\max} = 500$ and $K^{\mathsf{null}}_{\max} = 50$. Based on the isometry between $n \times n$ complex PSD matrices and $2n \times 2n$ real PSD matrices [81], we formulate problem (**c$\widetilde{\text{SDR}}$**)-(5.9) as a SDP problem in real numbers, and solve it with the IPM solver `MOSEK`. We initialize the PBM with a dual solution given by `MOSEK`; this is why we do not evaluate here the PBM as a standalone solver, but as a post-processing for `MOSEK`. We point out that (i) `MOSEK` is considered as the state-of-the-art solver for the ACOPF's semidefinite relaxation [70, 164] (ii) we call `MOSEK` with primal and dual feasibility tolerances (10^{-10}) smaller than tolerance by default (10^{-8}). Therefore, the examples of `MOSEK`'s inaccuracy presented here cannot be attributed to an early stop of `MOSEK` due to high-tolerance configuration. The considered ACOPF instances are taken from the reference library PGLib-OPF v21.07 [9], with typical operating conditions (TYP) and less than 7,000 buses. The code is available in the following repository: `github.com/aoustry/dualACOPFsolver`. The full results tables and logs are also available at this link.

5.4.2 Numerical results

For cases up to 1,000 buses, `MOSEK` is very accurate and hence, no significant accuracy improvement is made by the post-processing. We focus now on the instances with more than 1,000 buses. For case4019_goc, case4020_goc and case4661_sdet, we obtained an out-of-memory error during `MOSEK` execution. The results for other instances are presented in Table 5.3. The *estimated* lower bound ($\mathsf{ELB_M}$) is the value of the (potentially slightly infeasible) dual solution $Z_{\mathsf{M}} = (\underline{y}, \overline{y}, \underline{z}, \overline{z}, \underline{\alpha}, \overline{\alpha}, \beta, \gamma, \eta, \nu, \mu)$ computed by `MOSEK`. The certified lower bound given by `MOSEK` ($\mathsf{CLB_M}$) is the value $F\theta_{\mathsf{M}})$ with $\theta_{\mathsf{M}} = (\overline{\alpha} - \underline{\alpha}, \beta, \gamma, \eta, \nu, \mu)$. We point out that if Z_{M} is feasible and optimal in (**dualcSDR**), then both values $\mathsf{ELB_M}$ and $\mathsf{CLB_M}$ are equal; case2853_sdet is an example of such a situation. On the contrary, the spread between $\mathsf{ELB_M}$ and $\mathsf{CLB_M}$, observed for most of the instances in Table 5.3, is due to the slight infeasibility of Z_{M} in (**dualcSDR**). The certified lower bound given by the post-processing ($\mathsf{CLB_P}$) is the best value obtained by running Algorithm 5 initialized with $\hat{\theta}^0 = \theta_{\mathsf{M}}$. For brevity, these absolute values are expressed in scientific notation without mentioning the scale (e+5 or e+6). The progress of the estimated (resp. certified) lower bound is $\frac{\mathsf{CLB_P}-\mathsf{ELB_M}}{\mathsf{ELB_M}}$ (resp. $\frac{\mathsf{CLB_P}-\mathsf{CLB_M}}{\mathsf{CLB_M}}$), expressed in %. These figures should be compared with the targeted optimality gap in global optimization, that is typically 0.01%. Table 5.3 shows that, whether we talk about the progress with respect

Instance (case_###)	MOSEK status	Estim. LB MOSEK	Certif. LB MOSEK	Certif. LB post-proc.	Progress estim. LB	Progress certif. LB
3012wp_k	unknown	2.56486	2.54175	2.57994	0.59%	1.5%
6495_rte	unknown	2.65137	2.61727	2.64956	-0.07%	1.2%
3120sp_k	unknown	2.13237	2.09093	2.11465	-0.83%	1.1%
2736sp_k	unknown	1.30380	1.29815	1.30739	0.28%	0.71%
2737sop_k	unknown	7.75459	7.72214	7.77551	0.27%	0.69%
1354_pegase	unknown	1.23135	1.23066	1.23738	0.49%	0.55%
6515_rte	unknown	2.66299	2.65224	2.66667	0.14%	0.54%
2746wp_k	unknown	1.62822	1.62361	1.63139	0.19%	0.48%
6468_rte	unknown	2.05831	2.05267	2.06084	0.12%	0.40%
2746wop_k	unknown	1.20676	1.20469	1.20806	0.11%	0.28%
6470_rte	unknown	2.21458	2.21198	2.21576	0.053%	0.17%
1951_rte	unknown	2.08388	2.08152	2.08489	0.049%	0.16%
1888_rte	unknown	1.37253	1.37087	1.37296	0.03%	0.15%
2383wp_k	unknown	1.86230	1.86129	1.8629	0.03%	0.09%
4917_goc	unknown	1.37226	1.37153	1.37239	0.01%	0.06%
2869_pegase	unknown	2.44727	2.44645	2.44793	0.03%	0.06%
3022_goc	unknown	5.95436	5.95183	5.95505	0.01%	0.05%
2848_rte	unknown	1.28496	1.28476	1.28507	0.01%	0.02%
2312_goc	unknown	4.35665	4.35582	4.35683	0.00%	0.02%
4601_goc	optimal	8.26171	8.26110	8.26216	0.01%	0.01%
2868_rte	unknown	2.00945	2.00933	2.00952	0.00%	0.01%
2000_goc	optimal	9.72929	9.72904	9.72956	0.00%	0.01%
2742_goc	optimal	2.75607	2.75598	2.75607	0.00%	0.00%
4837_goc	optimal	8.71895	8.71883	8.71891	0.00%	0.00%
2853_sdet	optimal	2.03604	2.03604	2.03604	0.00%	0.00%
3970_goc	unknown	9.60853	9.60741	9.60741	-0.01%	0.00%
3375wp_k	unknown	7.40602	7.39211	7.39211	-0.19%	0.00%

Table 5.3: Estimated and certified lower bounds computed by MOSEK and by our post-processing algorithm (PBM)

to $\mathsf{ELB_M}$ or $\mathsf{CLB_M}$, the post-processing step yields an improvement in most cases: the best certified lower bound gets higher than both values by more than 0.01% for all 27 cases (in Table 5.3) but 5. We see that the progress with respect to `MOSEK`'s output value is more than 0.1% in 8 cases; it is more than 0.5% in one case. If we ask for *certification*, then the progress made by the post-processing step are even better: it is more than 0.1% for 13 cases and more than 0.5% in 7 cases. Figure 5.1 shows MOSEK's lower bound certification penalty (the ratio $(\mathsf{ELB}_\mathsf{M} - \mathsf{CLB}_\mathsf{M})/\mathsf{CLB}_\mathsf{M}$ in x-axis) versus the progress made by the PBM in terms of certified lower bound (the ratio $(\mathsf{CLB}_\mathsf{P} - \mathsf{CLB}_\mathsf{M})/\mathsf{CLB}_\mathsf{M}$, in y-axis). A point is above the $x = y$ line if the lower bound certified by the PBM is better than MOSEK's estimated value; if it is below, this means that the post-processing could not confirm that MOSEK's estimated value is indeed a valid lower bound.

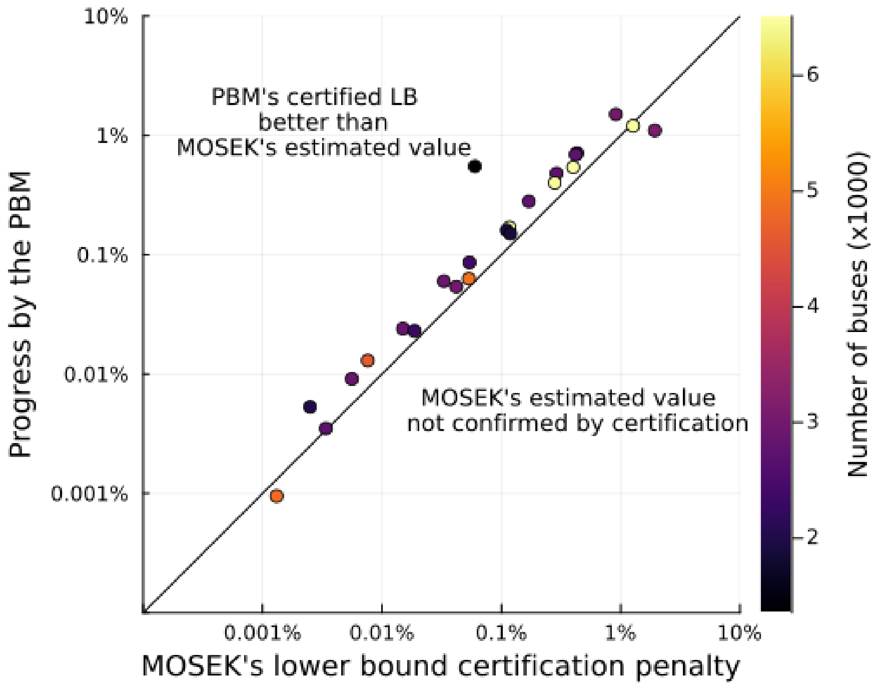


Figure 5.1: Certification penalty for MOSEK versus progress of the PBM

Unsurprisingly, the accuracy gains offered by the post-processing have a computational cost: running the PBM after calling MOSEK induces a time overhead. Table 5.4 presents this time overhead, expressed as a percentage of MOSEK's computational time. We observe that this relative overhead is very variable depending on the instance, ranging from 2% to 3300%. To confirm that this computational overhead is worth it, we measured how the increase of the lower bound reduces the optimality gap. Thanks to the solver Ipopt [238], we compute an upper-bound (UB) on val(ACOPF) (see Table 5.4). Note that for all the instances present in Table 5.3 but not in Table 5.4, Ipopt failed to converge, hence we could not compute an optimality gap. Table 5.4 displays MOSEK's estimated gap ($\frac{\mathsf{UB}-\mathsf{ELB}_\mathsf{M}}{\mathsf{UB}}$), MOSEK's certified gap ($\frac{\mathsf{UB}-\mathsf{CLB}_\mathsf{M}}{\mathsf{UB}}$) and the best certified gap after the post-processing ($\frac{\mathsf{UB}-\mathsf{CLB}_\mathsf{P}}{\mathsf{UB}}$). Table 5.4 also displays the relative reduction of the estimated gap $\frac{(\mathsf{UB}-\mathsf{ELB}_\mathsf{M})-(\mathsf{UB}-\mathsf{CLB}_\mathsf{P})}{\mathsf{UB}-\mathsf{ELB}_\mathsf{M}} = \frac{\mathsf{CLB}_\mathsf{P}-\mathsf{ELB}_\mathsf{M}}{\mathsf{UB}-\mathsf{ELB}_\mathsf{M}}$ and of the certified gap $\frac{\mathsf{CLB}_\mathsf{P}-\mathsf{CLB}_\mathsf{M}}{\mathsf{UB}-\mathsf{CLB}_\mathsf{M}}$. Table 5.4 shows that the accuracy gains allowed by the post-processing are significant, since they are of the order of magnitude of the optimality gap: the post-processing reduces the

Instance (case_###)	UB (IPOPT)	MOSEK's estim./cert. gap	Post-proc cert. gap	Estim./certif. gap reduction	Time overhead
2737sop_k	7.77719	0.291%/0.708%	0.022%	93%/ 97%	549%
2746wp_k	1.63171	0.214%/0.496%	0.020%	91%/ 96%	272%
2746wop_k	1.20826	0.124%/0.295%	0.017%	87%/ 94%	61%
2736sp_k	1.30800	0.321%/0.753%	0.047%	85%/ 94%	465%
4601_goc	8.26223	0.006%/0.014%	0.001%	87%/ 93%	4%
3012wp_k	2.59001	0.971%/1.863%	0.389%	60%/ 79%	444%
2383wp_k	1.86360	0.070%/0.124%	0.038%	46%/ 70%	814%
2868_rte	2.00961	0.008%/0.014%	0.004%	43%/ 66%	86%
1354_pegase	1.24250	0.897%/0.953%	0.412%	54% / 57%	2200%
2742_goc	2.75616	0.003%/0.007%	0.003%	0.0% / 50%	5%
3120sp_k	2.14655	0.661%/2.591%	1.486%	-125% / 43%	346%
2869_pegase	2.45053	0.133%/0.166%	0.106%	20% / 36%	444%
2000_goc	9.73324	0.041%/0.043%	0.038%	6.8% / 12%	978%
3022_goc	5.98838	0.568%/0.610%	0.557%	2.0% / 8.8%	805%
4917_goc	1.38152	0.670%/0.723%	0.661%	1.4% / 8.6%	529%
1888_rte	1.40530	2.331%/2.450%	2.301%	1.3% / 6.1%	3300%
4837_goc	8.72020	0.014%/0.016%	0.015%	-3.2% / 5.8%	24%
2312_goc	4.38452	0.636%/0.655%	0.632%	0.6% / 3.5%	240%
2853_sdet	2.04037	0.212%/0.212%	0.212%	0.0% / 0.0%	36%
3375wp_k	7.42469	0.251%/0.439%	0.439%	-75% / 0.0%	65%
3970_goc	9.60985	0.014%/0.025%	0.025%	-85% / 0.0%	2%

Table 5.4: Optimality gap reduction thanks to the post-processing step (PBM)

certified optimality gap by more than 50% for 10 instances. For 5 instances of them, the certified optimality gap is reduced by more than 90%. For these instances, the geometric average of the time overhead is 111%. Regarding the instances with a negative estimated gap reduction, they corresponds to the instances below the $x = y$ line in Figure 5.1: the best certified lower bound after the post-processing is lower than `MOSEK`'s estimated value.

5.5 Conclusion

This work shows that for many ACOPF instances from a reference benchmark, the dual bound computed by the state-of-the-art semidefinite programming interior-point solver `MOSEK` can be improved by a nonsmooth optimization algorithm used as a post-processing step, to the extent that the ACOPF's optimality gap is significantly reduced. This somehow illustrates a complementarity between these two algorithms: a first-order method may provide dual progress in a situation where the employed interior-point solver stalls. This work also shows the difficulty of achieving a given precision of the objective function value and, thus, the importance of manipulating certified lower bounds, which the proposed approach allows. In the context of a global optimization algorithm, the bounds exactness is critical to the correctness of the master (e.g. branch-and-bound) algorithm.

Further research could focus on speeding up this algorithm by parallelizing the spectral oracles and using a GPU implementation of the `OSQP` solver. Further works on the stopping criterion of the bundle method would help to identify cases where the improvement in the lower bound is small and where it is not worth spending additional computational time on post-processing. Finally, further steps could concern the integration of this approach into a global optimization algorithm such as the one we proposed in Chapter 4, or such as the spectral Moment-SOS hierarchy [155].

Chapter 6

Adjustable robust nonlinear optimization via semi-infinite programming and homotopy continuation

The power systems of the XXI$^{\text{st}}$ century are subject to uncertainties, mainly related to loads and wind and solar generation units, that depend on random factors. This last chapter addresses an electricity dispatch optimization problem under uncertainty [197]. We chose to refer to a generic framework to present our methodological contributions, but the numerical experiments focus on the Adjustable-Robust ACOPF problem. Adjustable Robust Optimization (ARO) models such situations where one has to make some decisions in the presence of uncertain parameters y_i. The decision variables are divided into “here-and-now” decisions x, before the uncertainty is realized, and “wait-and-see” decisions z, after the uncertainty is realized. Given four integers $n_x, n_y, n_z, n_h \in \mathbb{N} \setminus \{0\}$, two nonempty compact sets $\mathcal{X} \subset \mathbb{R}^{n_x}$ and $\mathcal{Y} \subset \mathbb{R}^{n_y}$, and three locally Lipschitz continuous functions $f\colon \mathbb{R}^{n_x} \to \mathbb{R}$, $g\colon \mathbb{R}^{n_x+n_y+n_z} \to \mathbb{R}^{n_z}$ and $h\colon \mathbb{R}^{n_x+n_y+n_z} \to \mathbb{R}^{n_h}$, we are interested in solving

$$\left.\begin{array}{lll} \min\limits_{x\in\mathcal{X}} & f(x) & \\ \text{s.t. } \forall y \in \mathcal{Y}, & \exists z \in \mathbb{R}^{n_z}, & g(x,y,z) = 0 \\ & & h(x,y,z) \leq 0. \end{array}\right\} \tag{ARO/ESIP}$$

As a matter of fact, this problem belongs to the class of Adjustable Robust Optimization [239]. In contrast to two-stage stochastic programs, ARO problems consider the worst case with respect to uncertainty rather than a statistical measure based on probability distributions. Since uncertainty and recourse only appear in the constraints of the present formulation, this problem can also be seen as an Existence-constrained Semi-Infinite Programming (ESIP), a term recently coined by Djelassi and Mitsos [66]. We acknowledge that this work deals with the special case where the number of equality constraints equates the dimension of the vector z: for a fixed $(x, y) \in \mathcal{X} \times \mathcal{Y}$, the nonlinear system of equations $g(x, y, z) = 0$ with unknown z is a square system. This is indeed the case in the AR-OPF formulation presented in Section 6.3.

Related works

Flexibility index of complex systems. The study of the flexibility index of complex systems predates the definition of the ESIP framework but can be seen as one particular case of the formulation (ARO/ESIP). Grossmann, Halemane, and Swaley [92, 93, 96] consider the problem of designing flexible chemical plants based on a formulation that is an ESIP. The solution techniques for the flexibility index problem mainly involve the reformulation as a bilevel optimization problem, and the convexification of the lower-level problem (then replaced by its KKT conditions) [73, 194]. The resulting nonconvex MINLP formulation, a relaxation of the original flexibility index problem, is solved with a global optimization method such as the mixed-integer nonlinear α-BB algorithm [5].

Min-max-min programs. A related class of problems that has received attention in the literature is the min-max-min programs. The connection with the formulation (ARO/ESIP) is the following: looking for a feasible point in (ARO/ESIP) amounts to solve the problem $\min_{x\in\mathcal{X}} \max_{y\in\mathcal{Y}} \min_{z\colon g(x,y,z)=0} \mathcal{H}(x, y, z)$, where $\mathcal{H}(x, y, z) = \max_i h_i(x, y, z)$. Polak and Royset [192] study min-max-min programs without assuming convexity, where the inner min operator operates over a finite set. To address these problems, they use a smoothing technique to replace the inner min operator with a smoothed approximation, effectively transforming the min-max-min program into a min-max approximation. This approximation is iteratively used and improved to solve min-max-min programs with finite or infinite max to local optimality. Tsoukalas et al. [217] extend this approach by applying the smoothing technique to the max operator in finite min-max-min programs as well. ESIP problems, however, may involve infinite sets. It remains unclear whether the smoothing technique can be extended to handle sets with infinite cardinality.

Adjustable Robust Optimization. Regarding ARO, a standard approach consists in restricting the recourse z to be a function of the uncertainty y. To obtain a finite-dimensional problem, the functional decision rule is optimized among a parametric family of functions $z(y, \theta)$, parameterized by a vector θ incorporated in the "here-and-now" decisions. In several works [15, 20, 21], this parametrization is chosen to be affine in y for tractability issue. This leads to a restriction of the original problem. In contrast, in the present work, we do not restrict z to be an explicit function of y and solve the formulation (ARO/ESIP) exactly. In some specific cases, the structure of the optimal decision rule may be deduced. Typically, in linear ARO problems where uncertainties lie solely on the right-hand side, it has been observed in [17] that there exist optimal piecewise linear decision rules. Expanding on this observation, Bertsimas and Goyal [21] proved that, when considering a linear ARO problem with right-hand side uncertainties and a simplex uncertainty set, affine decision rules are optimal. Also in [22], the authors proved that affine decision rules are optimal for ARO problems with a specific objective function (convex in the uncertain parameters and adjustable variables), with box constraints for the variables, and with a box uncertainty set. In the present case, we do not make such convexity assumptions.

Existence Constrained Semi-Infinite Programming. We emphasize that our formulation (ARO/ESIP) encompasses both equality and inequality constraints. In the aforementioned work by Djelassi and Mitsos [66], where the notion of ESIP is coined, the authors consider only inequality constraints in the lower-level problem and assume the existence of a point that strictly satisfies these inequalities: thus, unlike the present approach, [66] does not cover the case of equality constraints. Previous works by Stuber and Barton consider semi-infinite programming formulations with implicit functions [209, 210]: this corresponds to the case of the formulation (ARO/ESIP) where there exists a unique z_y such that $g(x, y, z_y) = 0$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$. We make no such assumption here. Moreover, this work by Stuber and Barton [209, 210] requires the use of exact computational methods based on interval arithmetics, untractable on large-scale problems. In contrast, we propose an algorithm based on scalable nonlinear programming methods.

Power systems applications. The Security-Constrained AC Optimal Power Flow (SC-OPF) and Adjustable-Robust AC Optimal Power Flow (AR-OPF) problems are ESIP that have attracted much attention in the operations research and power system communities [197]. In the SC-OPF, the uncertainty set $\mathcal{Y}$ representing possible line trippings on a subset of the electrical lines is finite [6, 40, 39]. In this case, the problem can be formulated as a finite nonlinear programming problem with a block structure suitable for decomposition techniques to handle large-scale problems with many contingencies [6, 80, 120, 185, 188]. In contrast, the AR-OPF involves continuous uncertainties regarding the power consumption (or the fluctuating power generation by wind or solar power plants) [129, 137, 149, 150]. The latter is more in line with the literature on ARO. In [110], Jabr applies the affinely adjustable robust optimization framework to the optimal power flow problem with the linear model of DC power flow. In [149], Louca and Bittar apply the affinely adjustable robust optimization framework to the nonconvex AC power flow equations. Another work by Louca and Bittar [150] applies the affinely adjustable robust optimization framework to the semidefinite programming relaxation of the AC power flow equations, and uses standard robust convex optimization techniques to obtain an inner approximation of the ARO feasible set. A limitation of this work is that it requires a flexible power injection at each bus, which is not a realistic assumption. Some other works, like our approach, tackle the nonconvex AC power flow model without restricting to affine recourse policies. Chamanbaz et al. [47] propose to work with a semidefinite relaxation of the AC power flow equations and proceed with a fixed sampling of the uncertainty set. The recent work by Lee et al. [137] uses a method to generate convex inner approximations of the AC power flow feasible set [136] based on Brouwer's fixed point theorem and inner convexification techniques. However, this approach may not be optimal due to the conservativeness of the robust convex constraint. Since the robust convex constraint is an inner approximation, it may only cover a subset of the robustly feasible points. Finally, in [129], the authors compute feasible solutions of a formulation similar to (ARO/ESIP) by designing a piecewise affine recourse policy. Thus, they come up with a robust polynomial programming problem (or semi-infinite polynomial programming problem) that is a restriction of the original problem. Based on sum-of-squares positivity certificates, they design a tractable reformulation of this restriction, the solution of which provides a feasible solution of the original adjustable robust optimization problem. An iterative scheme is proposed to improve the piecewise affine policy, but no convergence to a global optimum of the original problem is stated.

Contributions and organization of the chapter

This work looks at achieving computational tractability in the solution of generic (ARO/ESIP) formulation — without assuming convexity. To this end, we use a parsimonious adaptive discretization algorithm based on a deterministic sampling of the uncertainty set. The discretization algorithm is *parsimonious* in the sense that it does not require solving a difficult separation problem up to global optimality (or even to δ-optimality) at each iteration, contrary to standard adaptive discretization algorithms and their refinements [31, 161] (see Introduction). Instead, we employ the homotopy continuation method, a scalable algorithm for solving *locally* the system of nonlinear equations. Under a standard constraint qualification condition, we prove the convergence of this algorithm to either a global optimum or a local optimum depending, on whether the master problems are solved globally or locally. This flexibility pledges for scalability: even if only local optimality is achievable when solving the master problems, we keep a convergence guarantee to a local optimum of the ESIP problem. The proposed algorithm may also benefit from a parallel infrastructure of computation.

We apply the proposed algorithm to the AR-OPF problem. We provide extensive numerical results on small, middle, and large scale ACOPF instances from the IEEE PES PGLib benchmark. Whereas previous works on AR-OPF deals with instances with up to 118 buses [129, 137], we deal with instances of over 1000 buses.

The chapter is organized as follows: In Section 6.1, definitions and assumptions are collected, and an equivalent semi-infinite formulation is presented. In Section 6.2, the proposed adaptive discretization based on homotopy continuation and deterministic sampling is introduced; its convergence is proven under the Linear Independence Constraint Qualification (LICQ) condition.

Finite termination is proven for any positive feasibility tolerance. In Section 6.3, the numerical results for the AR-OPF problem are presented. Finally, Section 6.4 briefly concludes and gives an outlook for future work.

6.1 Semi-infinite programming reformulation and computational challenges

In this section, we aim to reformulate the existence-constrained semi-infinite program (ARO/ESIP) as a standard semi-infinite program. We also discuss the difficulties of solving this reformulation with the standard methods for semi-infinite programming.

6.1.1 Semi-infinite programming formulation

We start with an assumption guaranteeing the compactness of the set of feasible recourses.

Assumptions 6.1. *There exists a positive scalar ρ_z and an index $j_0 \in [\![1, r]\!]$, such that $h_{j_0}(x, y, z) = h_{j_0}(z) = \|z\| - \rho_z$.*

Remark 6.1. *In presence of bounds constraints $\underline{z}_i \leq z_i \leq \overline{z}_i$ among the inequalities $h(x, y, z) \leq 0$, we can always add a redundant constraint $\|z\| \leq \rho_z$ associated with $\rho_z = \sqrt{\sum_i \max(\underline{z}_i^2, \overline{z}_i^2)}$, so that Assumption 6.1 is satisfied. This is the case in the application considered in Section 6.3.*

We start by defining a function to quantify the infeasibility of the inequality constraints.

Definition 6.1. *We define the function $\mathcal{H} \colon \mathbb{R}^{n_x+n_y+n_z} \to \mathbb{R}$ as*

$$\mathcal{H}(x, y, z) = \max_{j \in [\![1, n_h]\!]} h_j(x, y, z). \tag{6.1}$$

Lemma 6.1. *The function $(x, y, z) \mapsto \mathcal{H}(x, y, z)$ is locally Lipschitz continuous. Under Assumption 6.1, for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$, the function $z \mapsto \mathcal{H}(x, y, z)$ is coercive.*

Proof. The function $(x, y, z) \mapsto \mathcal{H}(x, y, z)$ is locally Lipschitz continuous as the composition of locally Lipschitz continuous functions (max operator and h). For fixed $(x, y) \in \mathcal{X} \times \mathcal{Y}$, we consider a sequence $(z^k)_{k \in \mathbb{N}}$ such that $\|z^k\| \to +\infty$. Due to Assumption 6.1, $\mathcal{H}(x, y, z^k) \geq \|z^k\| - \rho_z$, therefore $\mathcal{H}(x, y, z^k) \to +\infty$. This proves that $z \mapsto \mathcal{H}(x, y, z)$ is coercive. □

Definition 6.2. *We define the set-valued mapping $\mathcal{Z}(x, y) = \{z \in \mathbb{R}^{n_z} \colon g(x, y, z) = 0\}$, and the extended real-valued function $G \colon \mathcal{X} \times \mathcal{Y} \mapsto \overline{\mathbb{R}}$ as*

$$G(x, y) = \inf_{z \in \mathbb{R}^{n_z}} \{\mathcal{H}(x, y, z) \text{ s.t. } g(x, y, z) = 0\} = \inf_{z \in \mathcal{Z}(x,y)} \mathcal{H}(x, y, z). \tag{6.2}$$

Lemma 6.2. *Under Assumption 6.1, for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$, $G(x, y) \in \mathbb{R} \cup \{+\infty\}$. Moreover, if $G(x, y) \in \mathbb{R}$, then there exists $z \in \mathcal{Z}(x, y)$ such that $G(x, y) = \mathcal{H}(x, y, z)$.*

Proof. For $(x, y) \in \mathcal{X} \times \mathcal{Y}$, we assume that $G(x, y) < +\infty$, i.e. $\mathcal{Z}(x, y)$ is not empty. By continuity of g, the nonempty set $\mathcal{Z}(x, y)$ is closed. The function $z \mapsto \mathcal{H}(x, y, z)$ is continuous, and coercive (Lemma 6.1, due to Assumption 6.1). Thus, problem (6.2) admits a minimizer z. □

Lemma 6.3. *Under Assumption 6.1, for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$, the following equivalence holds*

$$\exists z \in \mathcal{Z}(x, y), h(x, y, z) \leq 0 \iff G(x, y) \leq 0. \tag{6.3}$$

Proof. If there exists $z \in \mathcal{Z}(x, y)$ such that, for all $j \in [\![1, r]\!]$, $h_j(x, y, z) \leq 0$, then $\mathcal{H}(x, y, z) \leq 0$. As $z \in \mathcal{Z}(x, y)$, this implies that $G(x, y) \leq \mathcal{H}(x, y, z) \leq 0$. Conversely, if $G(x, y) \leq 0$, Lemma 6.2 yields the existence of $z \in \mathcal{Z}(x, y)$ such that $\mathcal{H}(x, y, z) = G(x, y) \leq 0$, i.e., for all $j \in [\![1, r]\!]$, $h_j(x, y, z) \leq 0$. □

We introduce the following semi-infinite programming problem:

$$\left.\begin{array}{rl} \min_{x \in \mathcal{X}} & f(x) \\ \text{s.t. } \forall y \in \mathcal{Y} & G(x, y) \leq 0. \end{array}\right\} \tag{SIP}$$

Proposition 6.1. *Under Assumption 6.1, the problems (SIP) and (ARO/ESIP) are equivalent.*

Proof. The objective function $f(x)$ and the constraint $x \in \mathcal{X}$ are identical in both formulations. The constraint $\forall y \in \mathcal{Y}, \exists z \in \mathbb{R}^{n_z}, g(x, y, z) = 0 \wedge h(x, y, z) \leq 0$ in formulation (ARO/ESIP) may be written as $\forall y \in \mathcal{Y}, \exists z \in \mathcal{Z}(x, y), h(x, y, z) \leq 0$. Lemma 6.3, under Assumption 6.1, yields the equivalence with the constraint $\forall y \in \mathcal{Y}, G(x, y) \leq 0$, which appears in formulation (SIP). This proves that both formulations are equivalent. □

6.1.2 The challenges of solving the semi-infinite reformulation

Under Assumption 6.1, we reformulate the problem (ARO/ESIP) as an equivalent semi-infinite program (SIP). The formulation (SIP) seems more approachable for optimization methods: it seems natural to address this formulation with an adaptive discretization method, such as Blankenship and Falk's algorithm [31] or one of its refinements. Nevertheless, various difficulties arise in the application of this algorithm in the present case:

- First, evaluating the function $G(x, y)$ involves computing a global minimizer of the nonconvex NLP problem (6.2), which can be out of reach at large scale,
- Second, the separation problem $\sup_{y \in \mathcal{Y}} G(x, y)$ is a difficult sup-inf nonlinear problem,

$$\sup_{y \in \mathcal{Y}} \inf_{z \in \mathcal{Z}(x,y)} \mathcal{H}(x, y, z), \tag{6.4}$$

- Third, the function $G(x, y)$ can be discontinuous and infinite-valued, whereas the convergence guarantee of the semi-infinite programming algorithms requires the continuity of the function [31, 46, 161].

Regarding the first limitation, we remind that problem (6.2) involves a square system of nonlinear equations $g(x, y, z) = 0$ with a set of solutions $\mathcal{Z}(x, y)$ that is potentially difficult to exhaustively enumerate. Often, we have at our disposal only a (potentially empty) subset of the solution. We name *local equation solver* any algorithm returning a subset of $\mathcal{Z}(x, y)$.

Definition 6.3. *We define a local equation solver as an algorithm R computing, for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$, a finite subset $\mathcal{Z}_R(x, y) \subset \mathcal{Z}(x, y)$ in finite time. We define*

$$G_R(x, y) = \inf_{z \in \mathcal{Z}_R(x,y)} \mathcal{H}(x, y, z). \tag{6.5}$$

Remark 6.2. *Note that the following holds: (i) $G(x, y) \leq G_R(x, y)$ for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$, (ii) the trivial algorithm returning the empty set is a local equation solver.*

In Section 6.2.1, we show that a homotopy continuation method provides a local equation solver with Lipschitzness properties. To overcome the second aforementioned limitation, i.e., the intractable sup-inf problem, we introduce in Section 6.2.2 a sampling-based separation oracle replacing the optimization-based oracle. We show in Section 6.2.3, that we can build on these objects to obtain an adaptive discretization scheme with convergence properties.

6.2 Discretization algorithm based on homotopy continuation and sampling

6.2.1 Homotopy continuation method for the nonlinear system of equation

Newton's method and its refinements, such as the homotopy continuation method [49], are standard algorithms to solve square systems of nonlinear equations. These algorithms rely on the assumption that the Jacobian matrix of the system of equations is locally invertible.

Assumptions 6.2. *(LICQ condition) The function g is twice continuously differentiable and, for all $(x, y, z) \in \mathcal{X} \times \mathcal{Y} \times B(0, \rho_z)$, $g(x, y, z) = 0 \implies \det(\nabla_z g(x, y, z)) \neq 0$.*

Under this assumption, we can derive a lower bound for the minimum singular value of the Jacobian, as specified in the following lemma. For the purpose of this lemma, we introduce the notation $\sigma_{\min}$ (resp. $\sigma_{\max}$) for the minimum (resp; maximum) singular value. We also introduce the set $\mathcal{Y}_\Delta = \{y \in \mathbb{R}^{n_y} : d(y, \mathcal{Y}) \leq \Delta\}$, where $d(y, \mathcal{Y})$ is the distance between y and the closed set $\mathcal{Y}$.

Lemma 6.4. *Under Assumption 6.2, there exist $\Delta, \sigma, M \in \mathbb{R}_{++}$ satisfying the following property: for all $(x, y, z) \in \mathcal{X} \times \mathcal{Y}_\Delta \times B(0, \rho_z + \Delta)$ such that $g(x, y, z) = 0$,*

$$\sigma_{\min}(\nabla_z g(x, y, z)) \geq \sigma \tag{6.6}$$

$$\sigma_{\max}(\nabla_z g(x, y, z)) \leq M. \tag{6.7}$$

In the rest of the chapter, we will refer to the scalars Δ, σ, M as introduced in this lemma, and to the scalar $\rho_y = \min\{\frac{\sigma\Delta}{2M}, \Delta\}$. In addition, we introduce $L_{\mathcal{H}}$, a Lipschitz constant for $\mathcal{H}$ over the compact set $(x, y, z) \in \mathcal{X} \times \mathcal{Y}_\Delta \times B(0, \rho_z + \Delta)$ (as detailed in Lemma 6.1, this function is indeed locally Lipschitz continuous).

Proof. We introduce the function $\psi(x, y, z) = \sigma_{\min}(\nabla_z g(x, y, z))$, and the set-valued mapping $A_\epsilon = \{(x, y, z) \in \mathcal{X} \times \mathcal{Y}_\epsilon \times B(0, \rho_z + \epsilon) \colon g(x, y, z) = 0\}$. We notice that the function $\Psi \colon \epsilon \mapsto \inf\{\psi(x, y, z) \colon (x, y, z) \in A_\epsilon\}$ is well-defined and takes its values in $\mathbb{R} \cup \{+\infty\}$, since ψ is continuous and A_ϵ is compact. As $A_{\epsilon_1} \subset A_{\epsilon_2}$ for any $\epsilon_1 \leq \epsilon_2$, the function Ψ is nonincreasing, which proves that the following limit exists:

$$\lim_{\epsilon \to 0^+} \Psi(\epsilon) = \Psi(0^+) \leq \Psi(0). \tag{6.8}$$

We take a positive sequence $\{\epsilon_k\} \in \mathbb{R}_{++}^{\mathbb{N}}$ such that $\epsilon_k \to 0$. Hence, $\Psi(\epsilon_k) \to \Psi(0^+)$ by definition of the right-limit. We consider a first case where there exists $k \in \mathbb{N}$ such that $A_{\epsilon_k} = \emptyset$: then, defining $\Delta = \epsilon_k$, and σ and M being any positive number, the statement of the lemma is true (empty set property). We consider the second case, where for all $k \in \mathbb{N}$, $A_{\epsilon_k} \neq \emptyset$: by compactness of A_{ϵ_k}, we can define $(x^k, y^k, z^k) \in A_{\epsilon_k}$ such that $\psi(x^k, y^k, z^k) = \Psi(\epsilon_k)$. The sequence $\{\epsilon_k\}$ being bounded, we can introduce an upper bound $\bar{\epsilon}$. Hence, any element of the sequence (x^k, y^k, z^k) belongs to the compact set $A_{\bar{\epsilon}}$, and up to the extraction of a subsequence, converges to a point (x, y, z) being such that $\psi(x, y, z) = \Psi(0^+)$ by continuity of ψ and uniqueness of the limit. By continuity of the distance, we know that $(x, y, z) \in A_0$, and $\Psi(0^+) = \psi(z) \geq \Psi(0)$. Together with Eq. (6.8) , this yields $\Psi(0^+) = \Psi(0) > 0$, the positivity being due to Assumption 6.2. We define $\sigma = \Psi(0)/2 > 0$, and we can therefore deduce that there exists $\Delta > 0$ such that $\Psi(\Delta) \geq \sigma$: therefore, for all $(x, y, z) \in A_\Delta$, we have $\psi(x, y, z) \geq \sigma$. Since the set A_Δ is compact and $(x, y, z) \to \sigma_{\max}(\nabla_y g(x, y, z))$ is continuous, this function admits a maximum M on this set. □

In our algorithm, the homotopy continuation is used to compute, starting from a triplet $(\bar{x}, \bar{y}, \bar{z})$ such that $g(\bar{x}, \bar{y}, \bar{z}) = 0$, a recourse z associated with another value $\hat{y} \in \mathcal{Y}$ of the uncertainty vector. Denoting $\bar{q} = \hat{y} - \bar{y}$, we define the homotopy

$$\Lambda(t, z) = g(\bar{x}, \bar{y} + t\bar{q}, z). \tag{6.9}$$

From the solution $\bar{z}$ such that $\Lambda(0, \bar{z}) = 0$, one can deduce by continuation a solution to the system $\Lambda(t, z) = 0$ for all $t \in [0, 1]$. To properly introduce the resulting solution $z = \pi(t)$, we define an Ordinary Differential Equation (ODE) related to this homotopy. For any triplet $(\bar{x}, \bar{y}, \bar{q})$, we define the vector field

$$F_{\bar{x},\bar{y},\bar{q}}(t, z) = -(\nabla_z g(\bar{x}, \bar{y} + t\bar{q}, z))^{-1} \nabla_y g(\bar{x}, \bar{y} + t\bar{q}, z)\bar{q}. \tag{6.10}$$

Then, any quadruplet $(\bar{x}, \bar{y}, \bar{z}, \bar{q})$ defines an ODE

$$\begin{cases} \dot{\pi}(t) = F_{\bar{x},\bar{y},\bar{q}}(t, \pi(t)) \\ \pi(0) = \bar{z}. \end{cases} \tag{6.11}$$

Proposition 6.2. *We consider $(\bar{x}, \bar{y}, \bar{z}) \in \mathcal{X} \times \mathcal{Y} \times B(0, \rho_z + \frac{\Delta}{2})$ such that $g(\bar{x}, \bar{y}, \bar{z}) = 0$. We take any $y \in B(\bar{y}, \rho_y)$ and define $\bar{q} = y - \bar{y}$. Under Assumption 6.2, the ODE (6.11) associated with $(\bar{x}, \bar{y}, \bar{z}, \bar{q})$ admits a solution $\pi(t)$ defined over $[0, 1]$, and $z = \pi(1)$ satisfies*

$$g(\bar{x}, y, z) = 0 \tag{6.12}$$

$$\|z - \bar{z}\| \leq \frac{M}{\sigma}\|y - \bar{y}\|. \tag{6.13}$$

Proof. We define the open set $U = \{(t, z) \in \mathbb{R}^{n_z+1} \colon \det(\nabla_z g(\bar{x}, \bar{y} + t\bar{q}, z)) \neq 0\}$. The vector field $(t, z) \mapsto F_{\bar{x},\bar{y},\bar{q}}(t, z)$ is well-defined and locally Lipschitz over U (we recall that $\nabla_z g$ is continuously differentiable). Hence, the Cauchy-Lipschitz theorem states that the corresponding differential equation admits an unique maximal solution $\pi(t)$ defined over an open interval $I = [0, T)$ $(T \in \mathbb{R} \cup \{\infty\})$. We make a first observation:

$$\forall t \in I \quad \frac{d}{dt}[\nabla g(\bar{x}, \bar{y} + t\bar{q}, \pi(t)] = 0. \tag{6.14}$$

Therefore, as $g(\bar{x}, \bar{y}, \pi(0)) = 0$, we deduce by integration

$$\forall t \in I \quad g(\bar{x}, \bar{y} + t\bar{q}, \pi(t)) = 0. \tag{6.15}$$

We define $t_s = \sup\{t \geq 0 \colon \pi([0, t]) \subset B(\bar{z}, \frac{\Delta}{2})\}$. By continuity of $\pi(t)$, we notice that $t_s > 0$. Note that, for all $t \in [0, t_s)$, $g(\bar{x}, \bar{y} + t\bar{q}, \pi(t)) = 0$ and $(\bar{x}, \bar{y} + t\bar{q}, \pi(t)) \in \mathcal{X} \times \mathcal{Y}_\Delta \times B(0, \rho_z + \Delta)$, therefore $\sigma_{\min}(\nabla_z g(\bar{x}, \bar{y} + t\bar{q}, z)) \geq \sigma$, and $\sigma_{\max}(\nabla_y g(\bar{x}, \bar{y} + t\bar{q}, z)) \leq M$ (see Lemma 6.4), and

$$\|\dot{\pi}(t)\| \leq \|{-(\nabla_z g(\bar{x}, \bar{y} + t\bar{q}, \pi(t)))^{-1}} \nabla_y g(\bar{x}, \bar{y} + t\bar{q}, \pi(t))\bar{q}\| \leq \frac{M\|\bar{q}\|}{\sigma}. \tag{6.16}$$

As $\|\bar{q}\| \leq \rho_y$, $\|\dot{\pi}(t)\| \leq \frac{M\rho_y}{\sigma} = \rho_z$. We deduce by integration, and triangle inequality that $\|\pi(t) - \bar{z}\| \leq t\rho_z$ for all $t \in [0, t_s)$. We assume that $t_s < 1$, and we show a contradiction. In this case, we know that $\pi(t_s)$ is well-defined by continuity of π and compactness of $B(\bar{z}, \frac{\Delta}{2})$, and therefore $\|\pi(t_s) - \bar{z}\| \leq t_s\frac{\Delta}{2}$. Since $(\bar{x}, \bar{y} + t_s\bar{q}, \pi(t_s)) \in \mathcal{X} \times \mathcal{Y}_\Delta \times B(0, \rho_z + \Delta)$, we deduce from Lemma 6.4 (under Assumption 6.2) that $\nabla_z g(\bar{x}, \bar{y} + t_s\bar{q}, \pi(t_s))$ is invertible, hence, $(\bar{x}, \bar{y} + t_s\bar{q}, \pi(t_s)) \in U$ and therefore $t_s < T$, by maximality of the interval $[0, T]$. As $\|\pi(t_s) - \bar{z}\| \leq t_s\frac{\Delta}{2} < \frac{\Delta}{2}$, we see that, by continuity of π, there exists $\epsilon \in \mathbb{R}_{++}$ such that $\pi([0, t_s + \epsilon]) \subset B(\bar{z}, \frac{\Delta}{2})$, which contradicts the definition of t_s. As a consequence $t_s \geq 1$, yielding that $\pi(t)$ is well-defined over $[0, 1]$, and $\pi([0, 1]) \subset B(\bar{z}, \frac{\Delta}{2})$. Due to Eq. (6.16), we deduce by integration, and triangle inequality that $\|\pi(1) - \bar{z}\| \leq \frac{M\|\bar{q}\|}{\sigma}$, i.e., $\|z - \bar{z}\| \leq \frac{M\|y - \bar{y}\|}{\sigma}$. □

A Newton homotopy solver computes a trajectory $\pi(t)$ defined over $[0, 1]$, and the resulting solution $z = \pi(1)$ satisfying Eqs. (6.12)-(6.13) by simulating the ODE (6.11). This simulation can be done by time discretization, using a predictor-corrector scheme, starting from $\pi(0) = \bar{z}$:

- Euler predictor: $\hat{\pi}(t_{i+1}) = \pi(t_i) + (t_{i+1} - t_i)F_{\bar{x},\bar{y},\bar{q}}(t_i, \pi(t_i))$.
- Newton corrector: applying a Newton method, initialized with $z = \hat{\pi}(t_{i+1})$, to compute $\pi(t_{i+1})$ the solution of the homotopy equation $\Lambda(t_{i+1}, z) = 0$.

Lemma 6.5. *Under Assumption 6.2, if the Newton homotopy continuation solver R receives $(\bar{x},\bar{y},\bar{z}) \in \mathcal{X} \times \mathcal{Y} \times B(0, \rho_z + \frac{\Delta}{2})$ such that $g(\bar{x},\bar{y},\bar{z}) = 0$ in input, then, the following holds:*

$$\forall y \in B(\bar{y}, \rho_y) \quad G_R(\bar{x}, y) \le \mathcal{H}(\bar{x},\bar{y},\bar{z}) + L_{\mathcal{H}}(1 + \frac{M}{\sigma})\|y - \bar{y}\|. \tag{6.17}$$

Proof. We take any $y \in B(\bar{y}, \rho_y)$, and we define $\bar{q} = y - \bar{y}$. We provide the point $(\bar{x},\bar{y},\bar{z}) \in \mathcal{X} \times \mathcal{Y} \times B(0, \rho_z + \frac{\Delta}{2})$ such that $g(\bar{x},\bar{y},\bar{z}) = 0$ to the homotopy continuation solver R simulating the ODE (6.11). Then, simulating the ODE (6.11) with parameters $(\bar{x},\bar{y},\bar{z},\bar{q})$, the solver can compute, according to the Proposition 6.2, the trajectory $\pi(t)$ and its final point $z = \pi(1)$ that is such that $g(\bar{x}, y, z) = 0$ and $\|z - \bar{z}\| \le \frac{M}{\sigma}\|y - \bar{y}\|$. Therefore $z \in \mathcal{Z}_R(\bar{x}, y)$, and $G_R(\bar{x}, y) \le \mathcal{H}(\bar{x}, y, z)$. As $\mathcal{H}$ is $L_{\mathcal{H}}$-Lipschitz continuous over $\mathcal{X} \times \mathcal{Y}_\Delta \times B(0, \rho_z + \Delta)$,

$$\begin{aligned} G_R(\bar{x}, y) \le \mathcal{H}(\bar{x}, y, z) &\le \mathcal{H}(\bar{x},\bar{y},\bar{z}) + L_{\mathcal{H}}\|(\bar{x},\bar{y},\bar{z}) - (\bar{x}, y, z)\| \\ &\le \mathcal{H}(\bar{x},\bar{y},\bar{z}) + L_{\mathcal{H}}(\|\bar{y} - y\| + \|\bar{z} - z\|) \\ &\le \mathcal{H}(\bar{x},\bar{y},\bar{z}) + L_{\mathcal{H}}(1 + \frac{M}{\sigma})\|y - \bar{y}\|. \end{aligned}$$

□

6.2.2 Sampling-based oracle

As mentioned in Section 6.1.2, the separation problem in the semi-infinite formulation (SIP) may be computationally intractable to solve to global optimality. We propose to overcome this hurdle by using an oracle that is not based on optimization but on a deterministic sampling scheme. In the following definition, we use the notation $d_H(A, B)$ for the Haussdorff distance between two sets. We underline that if $A \subset B$, $d_H(A, B) = \sup_{b \in B} d(b, A)$.

Definition 6.4. *We say that a family $(\mathcal{W}^k)_{k \in \mathbb{N}}$ of finite sets is a deterministic sampling of $\mathcal{Y}$ if $d_H(\mathcal{W}^k, \mathcal{Y}) \to 0$.*

In practice, such a sampling can be obtained, for example, by a succession of increasingly finer meshes, or by the use of a pseudo-random sequence $(w^t)_{t \in \mathbb{N}}$, defining $\mathcal{W}^k = (w^t)_{t \in [\![1,k]\!]}$. Leveraging this sampling and a local equation solver R, we present the sampling-based sorting oracle.

Definition 6.5. *The sorting oracle associated with a finite set $\mathcal{W} \subset \mathcal{Y}$, a local equation solver R, and an integer $D \in [\![1, |\mathcal{W}|]\!]$ computes the D elements $w_1, w_2, \dots, w_D$ of $\mathcal{W}$ with greatest value $G_R(x, w)$.*

6.2.3 Adaptive discretization algorithm

In this section, we present a method to solve the semi-infinite program (SIP) through an adaptive discretization algorithm, using the sampling-based oracle presented in Section 6.2.2. Given the equivalence between the formulations, computing an optimal (resp. locally optimal, feasible) solution of (SIP) yields an optimal (resp. locally optimal, feasible) solution of (ARO/ESIP). A standard way to relax a semi-infinite programming problem is to consider a finite number of constraints among the infinite set, giving a discretization of the semi-infinite problem. For a given finite set $\mathcal{Y}^k \subset \mathcal{Y}$, we consider the following finite nonlinear programming formulation

$$\begin{cases} \min_{x\in\mathcal{X}} & f(x) \\ \quad\text{s.t.} & \forall y \in \mathcal{Y}^k, G(x,y) \leq 0. \end{cases} \tag{P^k}$$

Proposition 6.3. *The finite problem (P^k) is a relaxation of the semi-infinite problem (SIP). An equivalent writing of (P^k) is*

$$\begin{cases} \min_{x,\boldsymbol{z}} & f(x) \\ \quad s.t. & \forall y \in \mathcal{Y}^k, g(x,y,\boldsymbol{z}_y) = 0. \\ & \forall y \in \mathcal{Y}^k, h(x,y,\boldsymbol{z}_y) \leq 0. \\ & x \in \mathcal{X}, \boldsymbol{z} \in (\mathbb{R}^{n_z})^{\mathcal{Y}^k}, \end{cases} \tag{Q^k}$$

where $\mathbf{z}_y \in \mathbb{R}^{n_z}$ represents the recourse vector in the uncertainty scenario y. We highlight that the number of variables in the formulation (Q^k) is $n_x + n_z \times |\mathcal{Y}^k|$: its grows linearly with the cardinality of the finite set $\mathcal{Y}^k$.

Proof. For any $x \in \mathcal{X}$ feasible in (SIP), we have that $G(x,y) \leq 0$ for all $y \in \mathcal{Y}$, and, in particular, for all $y \in \mathcal{Y}^k$. Hence, any $x \in \mathcal{X}$ feasible in (SIP) is also feasible in (P^k). The objective function between formulation being the same, we see that (P^k) is a relaxation of (SIP). Regarding the equivalence between (P^k) and (Q^k), this is a consequence from the following equivalence, due to Lemma 6.3: for any $x \in \mathcal{X}$

$$\begin{aligned} \forall y \in \mathcal{Y}^k\ G(x,y) \leq 0 &\iff \forall y \in \mathcal{Y}^k\ \exists z_y \in \mathcal{Z}(x,y)\ h(x,y,z_y) \leq 0, \\ &\iff \forall y \in \mathcal{Y}^k\ \exists z_y \in \mathbb{R}^{n_z}\ (g(x,y,z_y) = 0) \wedge (h(x,y,z_y) \leq 0). \end{aligned}$$

□

With all the building blocks being introduced, we can present the discretization algorithm with a sampling-based sorting oracle. Note the interest of the sorting oracle: rather than adding the constraint $G(x,w) \leq 0$ for all $w \in \mathcal{W}^k$ at iteration k, we add only the D constraints returned by the oracle, to prevent the size of the master problem from increasing too fast.

Algorithm 6 Discretization algorithm with a sampling-based sorting oracle

```
Input: A deterministic sampling $(\mathcal{W}^k)_{k\in\mathbb{N}}$ of $\mathcal{Y}$, integer $D \in \mathbb{N}^*$, integer $K \in \mathbb{N} \cup \{\infty\}$, finite set $\mathcal{Y}^1 \subset \mathcal{Y}$
1: for k = 1,. . . , K do
2:     Compute $(x^k, \mathbf{z}^k) \in \mathcal{X} \times (\mathbb{R}^{n_z})^{\mathcal{Y}^k}$, a solution to the relaxation $(Q^k)$. If this step fails, stop.
3:     Let $R^k$ be the Newton homotopy continuation solver based on the pairs $(y, \mathbf{z}^k_y)$ for $y \in \mathcal{Y}^k$.
4:     Let $w_1, w_2, \dots, w_D$ be the elements of $\mathcal{W}^k$ with greatest values $G_{R^k}(x^k, w)$ (sampling-based sorting oracle).
5:     $\mathcal{Y}^{k+1} \leftarrow \mathcal{Y}^k \cup \{w_1, w_2, \dots, w_D\}$.
6: end for
7: Return $x^K$.
```

Theorem 6.1. *Under Assumptions 6.1-6.2, if $K = \infty$, then Algorithm 6 either stops after a finite number of iterations due to the failure of step 2, or generates an infinite sequence $(x^k)_{k\in\mathbb{N}}$. In this case, the infeasibility error vanishes along the iterations:*

$$\left(\sup_{y\in\mathcal{Y}} G(x^k, y)\right)^+ \underset{k\to\infty}{\to} 0 \tag{6.18}$$

This result implies that, for any tolerance $\epsilon \in \mathbb{R}_{++}$, the iterate x^k gets feasible within tolerance ϵ after a finite number of iterations, if no failure occurred at step 2 of Algorithm 6.

Proof. We assume that the algorithm does not terminate, i.e. that step 2 never fails and that a solution $x^k \in \mathcal{X}$, $\mathbf{z}^k \in (\mathbb{R}^{n_z})^{\mathcal{Y}^k}$ feasible in (Q^k) is computed at every iteration k. We define the following sequences

$$\alpha_k = \sup_{w\in\mathcal{W}^k} G(x^k, w) \qquad \alpha_k^R = \sup_{w\in\mathcal{W}^k} G_{R^k}(x^k, w) \tag{6.19}$$

$$\beta_k = \sup_{y\in\mathcal{Y}} G(x^k, y) \qquad \beta_k^R = \sup_{y\in\mathcal{Y}} G_{R^k}(x^k, y). \tag{6.20}$$

In the definition of α_k^R, the homotopy solver R^k only use the information provided at step 3 regarding pairs (y, z_y^k) for $y \in \mathcal{Y}^k$, such that $g(x^k, y, z_y^k) = 0$. In the definition of β_k^R, we consider the state of the homotopy solver R^k after being evaluated at all points $w \in \mathcal{W}^k$ during step 4: hence, for the computation of β_k^R, the homotopy solver use the pairs (y, z_y^k) for $y \in \mathcal{Y}^k$ as initialization points, but also the pairs (w, z_w) such that $g(x^k, w, z_w) = 0$ for the sampled parameters $w \in \mathcal{W}^k$ for which the solver found a recourse z_w.

We start by proving that $\limsup_{k\to\infty} \alpha_k^R \leq 0$. We define $(\alpha_{k(\ell)}^R)_{\ell\in\mathbb{N}}$ a subsequence converging to $\limsup_{k\to\infty} \alpha_k^R$. In the definition of $\alpha_{k(\ell)}^R$ in Eq. (6.19), this supremum is over a finite set, therefore there exists $w^{k(\ell)} \in \mathcal{W}^{k(\ell)}$ such that $\alpha_{k(\ell)}^R = G_{R^{k(\ell)}}(x^{k(\ell)}, w^{k(\ell)}) \in \mathbb{R} \cup \{\infty\}$, and such that

$w^{k(\ell)}$ is returned by the sampling-based sorting oracle at the iteration $k(\ell)$. As $(w^{k(\ell)})_{\ell\in\mathbb{N}}$ is a bounded sequence, up to the extraction of a subsequence, we can assume here that it is converging. Therefore, there exists $\ell_0 \in \mathbb{N}$ such that, for all $\ell \geq \ell_0$, $\|w^{k(\ell+1)} - w^{k(\ell)}\| \leq \rho_y$. For any $\ell \geq \ell_0$, we define $\bar{x} = x^{k(\ell+1)}$, $\bar{y} = w^{k(\ell)}$. As $k(\ell+1) \geq k(\ell)+1$, $\bar{y} \in \mathcal{Y}^{k(\ell+1)}$ by definition of the algorithm, since $\bar{y}$ is returned by the oracle at iteration $k(\ell)$: therefore, we can also define $\bar{z} = z_{\bar{y}}^{k(\ell+1)}$. By feasibility of $x^{k(\ell+1)}, \mathbf{z}^{k(\ell+1)}$ in the master problem at iteration $k(\ell+1)$, $\mathcal{H}(\bar{x},\bar{y},\bar{z}) = \mathcal{H}(x^{k(\ell+1)}, \bar{y}, z_y^{k(\ell+1)}) \leq 0$, therefore Assumption 6.1 yields $\bar{z} \in B(0,\rho_z)$. Based on Assumption 6.2, Lemma 6.5 applied for $(\bar{x},\bar{y},\bar{z}) \in \mathcal{X} \times \mathcal{Y} \times B(0,\rho_z)$ (such that $g(\bar{x},\bar{y},\bar{z}) = 0$), and, for $w^{k(\ell+1)} \in B(\bar{y},\rho_y)$, gives

$$\alpha^R_{k(\ell+1)} = G_{R^{k(\ell+1)}}(x^{k(\ell+1)}, w^{k(\ell+1)}) \tag{6.21}$$

$$\leq L_{\mathcal{H}}(1+\frac{M}{\sigma})\|w^{k(\ell+1)} - w^{k(\ell)}\| + \mathcal{H}(x^{k(\ell+1)}, \bar{y}, z_{\bar{y}}^{k(\ell+1)}), \tag{6.22}$$

as the input $(\bar{x},\bar{y},\bar{z})$ was given to the local equation solver at step 3. By feasibility of $x^{k(\ell+1)}, \mathbf{z}^{k(\ell+1)}$ in the master problem at iteration $k(\ell+1)$, $\mathcal{H}(x^{k(\ell+1)}, \bar{y}, z_y^{k(\ell+1)}) \leq 0$ and, hence,

$$\alpha^R_{k(\ell+1)} \leq L_{\mathcal{H}}(1+\frac{M}{\sigma})\|w^{k(\ell+1)} - w^{k(\ell)}\|. \tag{6.23}$$

Using that $(w^{k(\ell)})_{\ell\in\mathbb{N}}$ is convergent, and that $\lim\limits_{\ell\to\infty} \alpha^R_{k(\ell)} = \limsup\limits_{k\to\infty} \alpha^R_k$, Eq. (6.23) gives $\limsup\limits_{k\to\infty} \alpha^R_k \leq 0$. As $\alpha_k \leq \alpha^R_k$, we know that $\limsup\limits_{k\to\infty} \alpha^R_k \leq 0$. We continue by proving that $\limsup\limits_{k\to\infty} \beta^R_k \leq 0$. As $d_H(\mathcal{W}^k,\mathcal{Y}) \underset{k\to\infty}{\to} 0$, and $\alpha^R_k \underset{k\to\infty}{\to} 0$ there exists $k_0 \in \mathbb{N}$ such that for all $k \geq k_0$, $\alpha^R_k \leq \frac{\Delta}{2}$, and $d_H(\mathcal{W}^k,\mathcal{Y}) \leq \rho_y$. For $k \geq k_0$, we define $\bar{x} = x^k$. We take any $y \in \mathcal{Y}$, and we define $\bar{y} \in \mathcal{W}^k$ such that $\|\bar{y} - y\| = d(y,\mathcal{W}^k) \leq d_H(\mathcal{W}^k,\mathcal{Y}) \leq \rho_y$. As $\bar{y} \in \mathcal{W}^k$, $G_{R^k}(\bar{x},\bar{y}) \leq \alpha^R_k \leq \frac{\Delta}{2}$, there exists $\bar{z} \in \mathcal{Z}_{R^k}(\bar{x},\bar{y})$ such that $\mathcal{H}(\bar{x},\bar{y},\bar{z}) \leq \frac{\Delta}{2}$, and that is successfully computed by the homotopy solver R^k, by definition of $G_{R^k}(\bar{x},\bar{y})$. In particular, this proves, due to Assumption 6.1, that $\bar{z} \in B(0,\rho_z + \frac{\Delta}{2})$. The triplet $(\bar{x},\bar{y},\bar{z})$ satisfies $g(\bar{x},\bar{y},\bar{z}) = 0$, $y \in B(\bar{y},\rho_y)$, and since the homotopy solver R^k has the knowledge of this triplet at the end of step 4, Lemma 6.5 yields

$$G_{R^k}(x^k, y) \leq L_{\mathcal{H}}(1+\frac{M}{\sigma})\|y-\bar{y}\| + \mathcal{H}(x^k,\bar{y},\bar{z}) \tag{6.24}$$

$$\leq L_{\mathcal{H}}(1+\frac{M}{\sigma})d_H(\mathcal{W}^k,\mathcal{Y}) + \alpha^R_k. \tag{6.25}$$

This being true for any $y \in \mathcal{Y}$, we obtain by taking the supremum over $\mathcal{Y}$ that

$$\beta^R_k \leq L_{\mathcal{H}}(1+\frac{M}{\sigma})d_H(\mathcal{W}^k,\mathcal{Y}) + \alpha^R_k. \tag{6.26}$$

As $d_H(\mathcal{W}^k,\mathcal{Y}) \underset{k\to\infty}{\to} 0$ and $\alpha^R_k \underset{k\to\infty}{\to} 0$, we have $\limsup\limits_{k\to\infty} \beta^R_k \leq 0$, and we also have $\limsup\limits_{k\to\infty} \beta_k \leq 0$ since $\beta_k \leq \beta^R_k$ for all k. We deduce that $\limsup\limits_{k\to\infty} \beta^+_k = 0$. □

Definition 6.6. *A point $\bar{x} \in \mathcal{X}$ is said r-locally optimal in a formulation (P) if (i) it is feasible in (P), and (ii) for all $x \in B(\bar{x}, r)$, "x feasible in (P)" implies $f(\bar{x}) \leq f(x)$.*

Theorem 6.2. *Under Assumptions 6.1-6.2, if $K = \infty$, and if Algorithm 6 generates an infinite sequence $(x^k)_{k\in\mathbb{N}}$, then it admits a limit point $\bar{x} \in \mathcal{X}$. For any such a limit point, if x^k is optimal (resp. r-locally optimal, feasible) in problem (P^k) for all the iterations k starting from a certain rank, then $\bar{x}$ is optimal (resp. r'-locally optimal for any $r' \in [0, r)$, feasible) in problem (SIP).*

Proof. We consider such a case where $K = \infty$, and Algorithm 6 generates an infinite sequence $(x^k)_{k\in\mathbb{N}} \in \mathcal{X}^{\mathbb{N}}$. As $\mathcal{X}$ is compact, this sequence admits a limit point $\bar{x} \in X$. We denote by $x^{k(\ell)}$ a subsequence converging to $\bar{x}$. We take any $y \in \mathcal{Y}$, and we observe that $G(x^{k(\ell)}, y) \leq \beta_k$ (see the definition in Eq. (6.20)). In the proof of Theorem 6.1, we showed that $\limsup_k \beta_k \leq 0$, therefore there exists $\ell_0 \in \mathbb{N}$, such that for all $\ell \geq \ell_0$, there exists $z^{k(\ell)} \in \mathcal{Z}(x^{k(\ell)}, y)$ such that $\mathcal{H}(x^{k(\ell)}, y, z^{k(\ell)}) \leq \beta_{k(\ell)} \leq 1$. Due to Assumption 6.1, we see that $z^{k(\ell)} \in B(0, \rho_z + 1)$. By compactness of $B(0, \rho_z + 1)$, and by continuity of g and $\mathcal{H}$, this means that there exists $z \in B(0, \rho_z + 1)$ a limit point of $(z^{k(\ell)})_{\ell\in\mathbb{N}}$ such that $g(\bar{x}, y, z) = 0$, and $\mathcal{H}(\bar{x}, y, z) \leq \limsup_k \beta_k \leq 0$, yielding $G(\bar{x}, y) \leq 0$. We conclude that $\bar{x}$ is feasible in (SIP).
We consider now the case where x^k is r-locally optimal in the formulation (P^k) at each iteration k, starting from a certain rank. We take any radius $r' \in [0, r)$. Since $x^{k(\ell)} \underset{\ell\to\infty}{\rightarrow} \bar{x}$, there exists ℓ_0 such that for all $\ell \geq \ell_0$, $\|\bar{x} - x^{k(\ell)}\| \leq r - r'$. Therefore, by triangle inequality, for all $\ell \geq \ell_0$ and $x \in B(\bar{x}, r')$, $\|x - x^{k(\ell)}\| \leq r$. By r-local optimality of $x^{k(\ell)}$ in (P^k), we have for all $\ell \geq \ell_0$

$$\forall x \in B(\bar{x}, r') \cap \mathsf{Feas}(P^k), f(x^{k(\ell)}) \leq f(x). \tag{6.27}$$

Since the problem (P^k) is a relaxation of (SIP), we have $\mathsf{Feas}(\text{SIP}) \subset \mathsf{Feas}(P^k)$, and we deduce that $\forall x \in B(\bar{x}, r') \cap \mathsf{Feas}(\text{SIP}), f(x^{k(\ell)}) \leq f(x)$, for all $\ell \geq \ell_0$. Taking the limit, since $x^{k(\ell)} \underset{\ell\to\infty}{\rightarrow} \bar{x}$, we have by continuity of $f(x)$ that

$$\forall x \in B(\bar{x}, r') \cap \mathsf{Feas}(\text{SIP}), f(\bar{x}) \leq f(x). \tag{6.28}$$

Recalling that $\bar{x}$ was proven to be feasible in (SIP), this proves that it is r'-optimal in (SIP).

Note that the case where x^k is globally optimal in (P^k) at each iteration can be covered by the r-local optimality case, for $r = 2\max_{x,x'\in\mathcal{X}}\|x - x'\|$. Indeed, x^k is then r-locally optimal in (P^k) at each iteration. Applying the previous result for $r' = \max_{x,x'\in\mathcal{X}}\|x - x'\| < r$, we deduce that $\bar{x}$ is r'-locally optimal in (SIP), therefore globally optimal in (SIP) since any point in $\mathcal{X}$ has a distance to x less or equal than r'. □

Remark 6.3. *If the problems (P^k) are solved to global optimality, then there is convergence to a global optimum of (SIP), even if the algorithm does not use a separation oracle based on global optimization. The fact that only a local solver (the homotopy continuation method) is used to solve the NLP subproblem (6.2) is not an issue, since the challenge of finding a globally optimal recourse for a given uncertainty $y \in \mathcal{Y}$ is addressed when solving (P^k).*

Corollary 6.1. *Under Assumptions 6.1-6.2, if the problem (SIP) is infeasible, then Algorithm 6 parameterized with $K = \infty$ terminates in a finite number of iterations due to the failure of step 2. Conversely, if Algorithm 6 terminates in a finite number of iterations due to the failure of step 2 with certified infeasibility, then problem (SIP) is infeasible.*

Proof. In Theorem 6.2, we proved that under Assumptions 6.1-6.2, if the algorithm generates an infinite sequence of iterates $(x^k)_{k\in\mathbb{N}}$, this implies the existence of a feasible point $\bar{x} \in \mathsf{Feas}(\text{SIP})$: by contrapositive, if there is not such $\bar{x} \in \mathsf{Feas}(\text{SIP})$ (infeasibility of (SIP)), this means that the algorithm terminates in finite time. Moreover, if $K = \infty$, this means that the algorithm terminates due to the failure of step 2. Regarding the second assertion: as (P^k) is a relaxation of (SIP), certifying the infeasibility of problem (P^k) proves that the problem (SIP) is infeasible. □

6.2.4 Stopping criterion in probability

The disadvantage of the robust optimization formulation, compared with the stochastic optimization one, is that it may be excessively conservative and therefore have a strong impact in terms of cost. We can imagine that a certain probability constraint may be sufficient in practice, i.e. we want to guarantee the existence of a feasible recourse with a sufficiently high probability.

Assumptions 6.3. *We consider a probability measure $\mathbb{P}_y$ over $\mathcal{Y}$, for which we can obtain independent and identically distributed samples. For any set $A \subset \mathcal{Y}$ such that the membership $y \in A$ can be efficiently tested, we can compute $\mathbb{P}_y(A)$.*

For a given local equation solver R, a tolerance $\epsilon \in \mathbb{R}_{++}$, and a point $x \in X$, we can efficiently compute whether $G_R(x, y) > \epsilon$ or not. Therefore, in the setting of Assumption 6.3, we are able to efficiently evaluate the probability $\mathbb{P}_y(\{G_R(x, y) > \epsilon\})$. The following theorem states that this probability falls below any threshold $p \in [0, 1]$ after a finite number of iterations.

Proposition 6.4. *For $\epsilon \in \mathbb{R}_{++}$, and $p \in [0, 1]$, we consider Algorithm 6, but with an additional stopping criterion*

$$\mathbb{P}_y(\{G_{R^k}(x, y) > \epsilon\}) \leq p. \tag{6.29}$$

Under Assumptions 6.1-6.3, the modified algorithm terminates in finite time.

Proof. If $K < \infty$, the finite termination is trivial. We consider, therefore, the case $K = \infty$. We assume that the modified algorithm does not stop in finite time, and we show a contradiction. If the modified algorithm does not stop in finite time, this means that it generates an infinite sequence $(x^k)_{k\in\mathbb{N}}$ as generated by Algorithm 6: indeed, the only difference between both algorithms is the stopping criterion Eq. (6.29), that is not met. In the proof of Theorem 6.1, we showed that $\limsup_k (\sup_{y\in\mathcal{Y}} G_{R^k}(x^k, y)) \leq 0$, therefore, there exists $k_0 \in \mathbb{N}$ such that for all $k \geq k_0$, $\sup_{y\in\mathcal{Y}} G_{R^k}(x^k, y) \leq \epsilon$, and, therefore $\mathbb{P}_y(\{G_{R^k}(x, y) > \epsilon\}) = 0 \leq p$. This shows that the stopping criterion is met in finite time. This is in contradiction with the hypothesis that the modified algorithm does not stop. □

6.3 Application to the adjustable robust ACOPF problem

We apply our methodology to the AR-OPF problem, a standard formulation of the ACOPF problem under load uncertainty. The model adopted here represents the dispatch of electricity in a power system, in the steady state regime and with automatic generation control (AGC) (see [112]), that proportionally adjusts the power injections of the generator to match the loads. This model is used in many robust and stochastic ACOPF formulations, such as those of [129, 137, 223]. Note that the formulation used here is not the QCQP formulation in complex numbers used in the previous chapters, but the polar formulation (see Appendix B), favored to find a local optimum only. In this formulation, the complex variable $V_b \in \mathbb{C}$ is represented by $v_b e^{\mathbf{i}\theta_b}$, with $v_b, \theta_b \in \mathbb{R}$.

6.3.1 Formulation

We first show how this model fits into the framework of the formulation (ARO/ESIP). We consider an oriented graph $\mathcal{N} = (\mathcal{B}, \mathcal{L})$ representing the power grid, as in the Introduction and Chapters 4 and 5. We also consider a set $\mathcal{G}$ of power generators, located at some buses $\mathcal{G} \subset \mathcal{B}$ (PV and REF buses); we assume here that there is at most one generator per bus, therefore we can see $\mathcal{G}$ as a subset of $\mathcal{B}$.

Decision variables. The set of buses without generator (PQ buses) is $\mathcal{C} = \mathcal{B} \backslash \mathcal{G}$. The decision vector x represents the dispatch variables, i.e., the voltage and the active power production setpoint at each bus in $\mathcal{G}$:

$$x = (v_g, P_g^{set})_{g \in \mathcal{G}}, \tag{6.30}$$

with the associated constraint set $\mathcal{X} = \prod_{g \in \mathcal{G}} [\underline{v}_g, \overline{v}_g] \times [\underline{P}_g, \overline{P}_g]$.

Uncertainties. We model the active p^{d} and reactive power loads q^{d} as random uncertainties, parameterized by a random vector $y \in \mathbb{R}^{|\mathcal{B}|}$ such that at each bus $i \in \mathcal{N}$,

$$p_i^{\mathsf{d}}(y) = (1 + y_i) P_i^{\mathsf{d}} \ , \quad q_i^{\mathsf{d}}(y) = (1 + y_i) Q_i^{\mathsf{d}}, \tag{6.31}$$

where P_i^{d} and Q_i^{d} are the active and reactive power in the reference scenario. To represent some correlations between the loads, the compact uncertainty set $\mathcal{Y} \subset \mathbb{R}^{|\mathcal{B}|}$ is built upon a matrix $A \in \mathbb{R}^{|\mathcal{B}| \times r}$ such that $A^\top A = I_r$, and a scaling factor $\lambda \in \mathbb{R}$:

$$\mathcal{Y} = \{\lambda A u \colon u \in \mathbb{R}^r, \|u\| \le \sqrt{2r}\}. \tag{6.32}$$

When speaking about probability distribution, the vector u is a standard normal random vector, truncated on $B(0, \sqrt{2r})$ (conditional probability law). This results in a certain distribution law for the vector y over $\mathcal{Y}$. The scaling variable λ is adapted to match a certain level of standard deviation for the parameters y_i (see Section 6.3.2).

Recourse variables. The recourse vector z corresponds to the variables describing the internal state of the power network:

$$z = \{(v_i)_{i\in\mathcal{C}}, (\theta_i)_{i\in\mathcal{B}}, \Delta\}, \tag{6.33}$$

where v_i and θ_i are the voltage magnitude and angle at the bus i, and Δ is a slack variable representing the automatic adjustment of the generators. We underline that $n_z = 2|\mathcal{B}| - |\mathcal{G}| + 1$. After the uncertainty occurs, the power generation at each generator $g \in \mathcal{G}$ follows the response

$$R_g(P_g^{set}, \Delta) = \min\{\max\{P_g^{set} + \alpha_g\Delta, \underline{P}^g\}, \overline{P}^g\}. \tag{6.34}$$

The α_i are proportional response parameters (fixed to $\alpha_i = \frac{1}{|\mathcal{G}|}$ in our numerical experiments). Note that the function inherits the nonsmoothness of the min and max operators. In practice, we replace these operators by a smooth approximation: for a given smoothing parameter η, the expression $\max(t_1, t_2)$ is replaced by $\eta\ln(\exp(\frac{t_1}{\eta}) + \exp(\frac{t_2}{\eta}))$, and $\min(t_1, t_2)$ is replaced by $-\eta\ln(\exp(-\frac{t_1}{\eta}) + \exp(-\frac{t_2}{\eta}))$, in concordance with [80]. Then, we can introduce the function $g(x, y, z)$ that describes different kind of expressions. In the following, the coefficients $\mathfrak{b}_i, \mathfrak{g}_i, \mathfrak{b}_{ij}$ and $\mathfrak{g}_{ij}$ are physical parameters describing the shunts and the lines. In Appendix B, we explain how they relate to the previously introduced admittances $Y_i^s, Y_{ij}^{\mathsf{ff}}, Y_{ij}^{\mathsf{ft}}, Y_{ij}^{\mathsf{tf}}, Y_{ij}^{\mathsf{tt}} \in \mathbb{C}$.

- Angle at the reference bus: $g_0(x, y, z) = \theta_r$, where $r \in \mathcal{B}$ is the reference (REF) bus.
- Active power conservation at bus $i \in \mathcal{G}$:
$$g_{p:i}(x,y,z) = R_i(P_i^{set}, \Delta) - p_i^{\mathsf{d}}(y) - \mathfrak{g}_i v_i^2 - \sum_{j:(i,j)\in\mathcal{L}\cup\mathcal{L}^R} v_i v_j\big(\mathfrak{g}_{ij}\cos(\theta_i - \theta_j) + \mathfrak{b}_{ij}\sin(\theta_i - \theta_j)\big).$$
- Active power conservation at bus $i \in \mathcal{C}$:
$$g_{p:i}(x,y,z) = -p_i^{\mathsf{d}}(y) - \mathfrak{g}_i v_i^2 - \sum_{j:(i,j)\in\mathcal{L}\cup\mathcal{L}^R} v_i v_j\big(\mathfrak{g}_{ij}\cos(\theta_i - \theta_j) + \mathfrak{b}_{ij}\sin(\theta_i - \theta_j)\big).$$
- Reactive power conservation at bus $i \in \mathcal{C}$:
$$g_{q:i}(x,y,z) = -q_i^{\mathsf{d}}(y) + \mathfrak{b}_i v_i^2 - \sum_{j:(i,j)\in\mathcal{L}\cup\mathcal{L}^R} v_i v_j\big(\mathfrak{g}_{ij}\sin(\theta_i - \theta_j) - \mathfrak{b}_{ij}\cos(\theta_i - \theta_j)\big).$$

We underline that the dimension of $g(x, y, z)$ is indeed $2|\mathcal{B}| - |\mathcal{G}| + 1 = n_z$. Finally, the function $h(x, y, z)$ is also defined with different kind of expressions:

- Voltage magnitude limit at the bus $i \in \mathcal{C}$:
$$h_{v1:i}(x,y,z) = v_i - \overline{v}_i$$
$$h_{v2:i}(x,y,z) = \underline{v}_i - v_i.$$
- Reactive power limit at the bus $i \in \mathcal{G}$:
$$h_{q1:i}(x,y,z) = q_i^{\mathsf{d}}(y) - \mathfrak{b}_i v_i^2 + \sum_{j:(i,j)\in\mathcal{L}\cup\mathcal{L}^R} v_i v_j\big(\mathfrak{g}_{ij}\sin(\theta_i - \theta_j) - \mathfrak{b}_{ij}\cos(\theta_i - \theta_j)\big) - \overline{Q}_i$$
$$h_{q2:i}(x,y,z) = \underline{Q}_i - q_i^{\mathsf{d}}(y) + \mathfrak{b}_i v_i^2 - \sum_{j:(i,j)\in\mathcal{L}\cup\mathcal{L}^R} v_i v_j\big(\mathfrak{g}_{ij}\sin(\theta_i - \theta_j) - \mathfrak{b}_{ij}\cos(\theta_i - \theta_j)\big).$$

6.3.2 Experimental protocol

We implemented Algorithm 6 in the `Julia` programming language, using the package `ExaPF.jl`. The relaxation (P^k) are solved to local optimality, with the nonlinear programming solver `Ipopt` [238]. The solver is warm-started at each iteration with the solution of the previous iteration. The minimum value for the barrier parameter is set to 10^{-4} (`Ipopt` parameter: `mu_target`), and the smoothing parameter for the approximate active power response function is $\eta = 10^{-3}$. For the homotopy continuation method, we use the Newton solver implemented in the `ExaPF` framework. At iteration k, the set $\mathcal{W}^k$ is split into 12 batches, corresponding to 12 threads executed in parallel: each thread applies the homotopy continuation method for each of the 4000 uncertainty values in its batch. The number D of scenarios returned by the sorting oracle is $D = 5$. We set the maximal number of iterations to $K = 25$, and a timeout of 18,000s.

As explained in Section 6.3.1, we consider a probability distribution for $y = \lambda A u$, with u being a truncated multivariate Gaussian vector of dimension $r = \min\{8, |\mathcal{B}|\}$ (see Eq. (6.32)) and A a randomly generated matrix satisfying $A^\top A = I_r$. We calibrate λ so that the standard deviation of y_i matches a certain level $\bar{s}$ with $\bar{s} \in \{0.5\%, 1\%, 2\%\}$ in average over $i \in [\![1, |\mathcal{B}|]\!]$. Based on the resulting probability distribution $\mathbb{P}_y$ for y, we apply the stopping criterion

$$\mathbb{P}_y(G(x,y) > \epsilon) \leq p, \tag{6.35}$$

with $\epsilon = 10^{-4}$ and $p = 10^{-4}$. This statistical test to evaluate this probability is performed with a 99% confidence level. We emphasize that the point x returned is not necessarily feasible in the robust sense of the formulation (SIP), but in the sense of the probabilistic constraint Eq. (6.29): the probability $\mathbb{P}_y(G(x,y) \leq \epsilon)$ is estimated to be greater than 99.99%. The algorithm is applied to the instances of the IEEE PES PGLib benchmark with a size between 3 and 1354 buses, under standard and congested conditions (API), and is run on a 64-bit Ubuntu computer with 32 Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz and 64 GB RAM. The code is available at the following repository: `github.com/aoustry/robustOPF`.

6.3.3 Numerical results

Table 6.1 gathers the numerical results of Algorithm 6, under the stopping criterion Eq. (6.35) . The different columns are the following:

- "Instance" is the considered instance from the IEEE PES PGLib benchmark. The number in the name corresponds to the number of buses $|\mathcal{B}|$.
- "Std. dev. (%)" corresponds to $\bar{s}$, the aforementioned standard deviation level of the coordinates y_i (in percentage). This value implicitly determines the magnitude parameter λ (see Eq. (6.32)).
- "Termin. status" describes the reason of termination of the algorithm: "success" means that the stopping criterion $\mathbb{P}_y(\{G_{R^k}(x,y) > \epsilon\}) \leq p$ was met, "MaxIt" that the maximal

number of iterations was met, “TimeOut” that the time limit was reached, “Infeasible” that the solution of the master problem (i.e. the relaxation (P^k)) failed at some point. “Sing.” means singularity: the Newton solver failed due to the singularity of the nonlinear system, i.e., a point where Assumption 6.2 is not satisfied was found.

- “Uncertainty cost (%)” is the increase (in percentage) of the objective value, from the value of the deterministic ACOPF problem (reference scenario) to the value of the solution ouput by the algorithm.
- “It. nb.” is the number of iterations.
- “$|\mathcal{Y}^k|$” is the cardinality of $\mathcal{Y}^k$ at the last iteration k.
- “Master time (s)” is the total time spent in solving the relaxations (P^k).
- “Oracle time (s)” is the total time spent in the sampling-based sorting oracle..
- “Total time (s)” is the total computation time.

Table 6.1: Performance of the proposed adaptive discretization algorithm

Instance	**Std. dev. (%)**	**Termin. status**	**Uncertainty cost (%)**	**It. nb.**	$\lvert\mathcal{Y}^k\rvert$	**Master time (s)**	**Oracle time (s)**	**Total time (s)**
3_lmbd	0.5	success	0.00	1	1	≤ 1	1	1
3_lmbd	1	success	0.00	1	1	≤ 1	2	2
3_lmbd	2	success	0.00	1	1	1	1	2
3_lmbd_api	0.5	success	0.00	1	1	≤ 1	1	1
3_lmbd_api	1	success	0.00	1	1	≤ 1	1	1
3_lmbd_api	2	success	0.00	1	1	≤ 1	1	1
5_pjm	0.5	success	0.00	1	1	7	4	11
5_pjm	1	success	0.01	2	6	6	7	13
5_pjm	2	success	0.01	2	6	6	7	14
5_pjm_api	0.5	success	0.00	1	1	≤ 1	1	1
5_pjm_api	1	success	0.00	1	1	≤ 1	1	2
5_pjm_api	2	success	0.00	1	1	≤ 1	1	1
14_ieee	0.5	success	0.00	1	1	4	12	16
14_ieee	1	success	0.01	2	5	4	21	25
14_ieee	2	success	0.01	2	6	5	20	25
14_ieee_api	0.5	success	0.01	2	6	≤ 1	18	18
14_ieee_api	1	success	0.01	2	6	≤ 1	17	17
14_ieee_api	2	success	0.03	2	6	≤ 1	16	16
24_ieee_rts	0.5	success	0.01	3	11	7	41	48
24_ieee_rts	1	success	0.01	3	11	6	39	46
24_ieee_rts	2	success	0.02	3	11	7	39	45
24_ieee_rts_api	0.5	success	0.02	4	14	1	50	51
24_ieee_rts_api	1	success	0.18	5	15	1	59	59
24_ieee_rts_api	2	success	1.83	5	16	3	51	54

Instance	**Std. dev. (%)**	**Termin. status**	**Uncertainty cost (%)**	**It. nb.**	$\lvert\mathcal{Y}^k\rvert$	**Master time (s)**	**Oracle time (s)**	**Total time (s)**
30_as	0.5	success	0.00	1	1	4	18	22
30_as	1	success	0.00	1	1	5	19	23
30_as	2	success	0.00	1	1	4	19	24
30_as_api	0.5	success	0.00	2	6	2	42	44
30_as_api	1	success	0.01	3	11	3	60	63
30_as_api	2	success	0.22	3	11	3	58	61
30_ieee	0.5	success	0.01	2	6	1	40	41
30_ieee	1	success	0.01	2	6	1	39	40
30_ieee	2	success	0.03	3	11	1	58	59
30_ieee_api	0.5	success	0.02	2	6	1	40	41
30_ieee_api	1	success	0.03	2	6	1	38	39
30_ieee_api	2	success	0.07	2	6	1	38	39
39_epri	0.5	success	0.00	1	1	1	37	38
39_epri	1	success	12.37	2	6	2	63	64
39_epri	2	success	13.96	2	6	2	61	62
39_epri_api	0.5	success	1.50	7	31	21	96	117
39_epri_api	1	success	1.54	7	31	16	76	92
39_epri_api	2	success	4.06	4	16	6	19	26
57_ieee	0.5	success	0.02	2	6	4	47	51
57_ieee	1	success	0.02	2	6	1	46	47
57_ieee	2	success	0.20	3	11	3	68	71
57_ieee_api	0.5	MaxIt	*	25	126	692	453	1145
57_ieee_api	1	Infeas	*	5	26	15	150	165
57_ieee_api	2	Infeas	*	4	21	3	163	167
73_ieee_rts	0.5	success	0.01	3	11	18	115	133
73_ieee_rts	1	success	0.02	3	11	45	117	162
73_ieee_rts	2	success	0.03	4	16	66	154	220
73_ieee_rts_api	0.5	success	0.08	3	11	3	118	121
73_ieee_rts_api	1	success	0.30	3	11	6	114	120
73_ieee_rts_api	2	success	1.72	5	21	47	178	226
89_pegase	0.5	success	0.00	1	1	2	57	59
89_pegase	1	success	0.10	2	4	2	114	116
89_pegase	2	MaxIt	*	25	126	856	461	1317
89_pegase_api	0.5	success	0.00	1	1	2	52	54
89_pegase_api	1	success	0.05	2	6	6	130	136
89_pegase_api	2	success	0.06	2	6	4	108	112
118_ieee	0.5	success	0.00	1	1	2	29	31
118_ieee	1	success	0.08	2	6	2	64	66
118_ieee	2	success	0.1	3	8	3	91	94

Instance	**Std. dev. (%)**	**Termin. status**	**Uncertainty cost (%)**	**It. nb.**	$\lvert\mathcal{Y}^k\rvert$	**Master time (s)**	**Oracle time (s)**	**Total time (s)**
118_ieee_api	0.5	success	0.05	2	6	4	63	67
118_ieee_api	1	success	0.05	2	6	4	64	69
118_ieee_api	2	success	0.98	5	18	33	144	177
179_goc	0.5	success	0.02	3	11	8	410	418
179_goc	1	success	0.02	3	11	8	276	284
179_goc	2	success	0.03	4	13	10	292	302
179_goc_api	0.5	success	2.58	8	36	341	625	967
179_goc_api	1	sing.	*	1	1	5	≤ 1	5
179_goc_api	2	sing.	*	1	1	5	≤ 1	5
200_activ	0.5	success	0.00	1	1	6	155	161
200_activ	1	success	0.00	1	1	6	142	148
200_activ	2	success	0.02	2	6	10	284	294
200_activ_api	0.5	success	0.00	1	1	2	136	138
200_activ_api	1	success	0.01	2	6	4	277	281
200_activ_api	2	success	0.01	2	6	4	271	275
240_pserc	0.5	success	0.03	4	16	216	448	664
240_pserc	1	success	0.04	7	23	743	910	1653
240_pserc	2	success	0.06	9	33	1295	1032	2327
240_pserc_api	0.5	success	0.04	4	16	239	680	919
240_pserc_api	1	success	0.24	6	25	661	1004	1664
240_pserc_api	2	success	0.12	7	23	752	1033	1785
300_ieee	0.5	success	2.42	6	26	363	992	1355
300_ieee	1	success	3.92	9	41	896	1438	2334
300_ieee	2	MaxIt	*	25	126	7634	1749	9382
300_ieee_api	0.5	success	1.68	2	6	26	400	426
300_ieee_api	1	success	1.63	4	13	166	878	1043
300_ieee_api	2	success	17.35	9	38	689	1603	2291
588_sdet	0.5	success	0.00	1	1	6	133	139
588_sdet	1	success	0.00	1	1	6	170	176
588_sdet	2	success	1.12	2	6	44	393	437
588_sdet_api	0.5	success	0.00	1	1	4	172	176
588_sdet_api	1	success	0.00	1	1	4	172	176
588_sdet_api	2	success	0.00	1	1	4	173	177
1354_pegase	0.5	success	0.28	4	13	743	3435	4178
1354_pegase	1	TimeOut	*	12	61	11436	6735	18171
1354_pegase	2	TimeOut	*	15	76	18251	113	18363
1354_pegase_api	0.5	success	0.00	1	1	23	1194	1217
1354_pegase_api	1	success	0.14	2	6	177	2374	2551
1354_pegase_api	2	success	0.14	2	6	146	961	1107

Over the 102 instances, the algorithm terminates with the satisfaction of the stopping

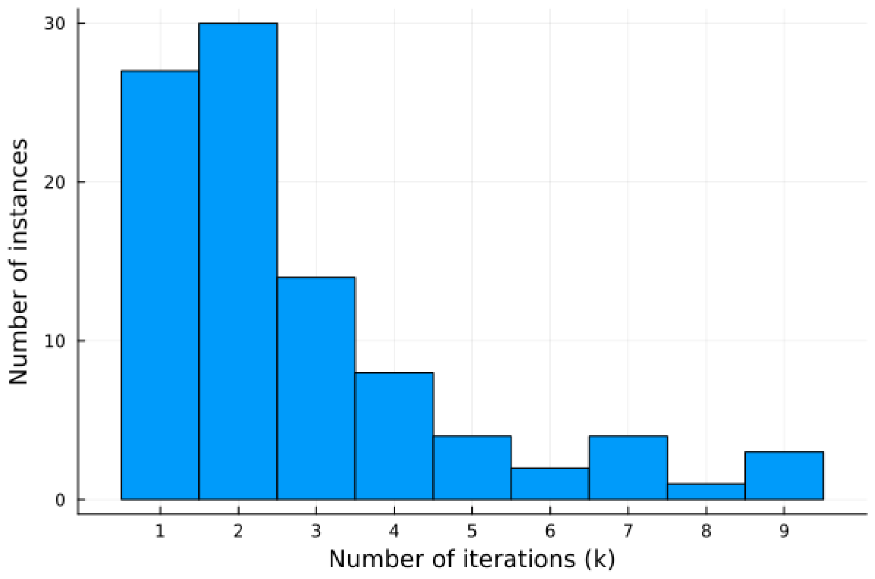


Figure 6.1: Number of instances for which the sampling-based discretization algorithm succeeds after k iterations

criterion in 93 cases. For 2 cases, it stops because `Ipopt` returns a "locally infeasible" status. In 5 cases, the maximum number of iterations or the time limit is reached: in these cases, there is a suspicion of infeasibility, in the sense that `Ipopt` does not manage to find a local optimum of the master problem at each iteration — but do not state local infeasibility. Finally, for 2 cases (related to the same network 179_goc_api), the Newton solver failed due to the singularity of the nonlinear system, i.e., a point where Assumption 6.2 is not satisfied was found: in most of the cases, no counterexample (x, y, z) violating Assumption 6.2 was found.

The histogram in Figure 6.1 represents the number of instances (in y-axis) of the benchmark for which our algorithm succeeds after k iterations (in x-axis). We see that a limited number of iterations of the algorithm are necessary to obtain a point feasible in the sense of the probabilistic constraint (6.29). We see different explanations for this: (i) the uncertainty set is low-dimensional, compared to the decision set (ii) the selection oracle looks for the most adverse uncertainties, which somehow, dominate the others. We have to acknowledge that in some cases (3_lmbd,3_lmbd_api, 5_pjm, 5_pjm_api, 30_as or 588_sdet_api), even for the largest uncertainty level, the point x^1 obtained at the first iteration with $\mathcal{Y}^1 = \{\mathbf{0}\}$ is already feasible in the sense of the probabilistic constraint (6.29). The probability $\mathbb{P}_y(G(x^1, y) > 10^{-4})$ is already below 0.01%: the load variations seem to have little effect on the constraints that are tight in the reference scenario.

Figure 6.2 represents, for the three different values of the standard deviation $\bar{s}$ of the uncertainty, the distribution of the cost of uncertainty (4th column in Table 6.1) over all test cases for which the algorithm succeeded. We observe in Table 6.1 and in Figure 6.2 that the objective value of the computed point x consistently increases with the magnitude of

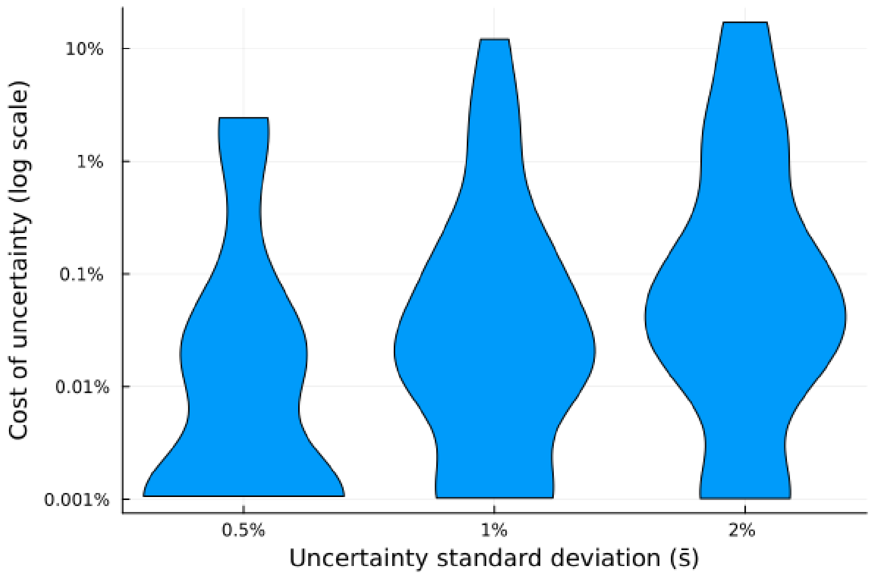


Figure 6.2: Distributions of the cost of uncertainty, depending on the uncertainty level

the uncertainties for most of the instances: logically, the cost of uncertainty is higher for higher uncertainty levels. However, one should acknowledge that 2 cases make exceptions: 240_pserc_api, where the situation $\bar{s} = 1$ is costlier than the situation $\bar{s} = 2$, and 300_ieee_api, where the situation $\bar{s} = 0.5$ is costlier than the situation $\bar{s} = 1$. This anomaly seems to occur because `Ipopt` sometimes fails to compute a locally optimal point and just returns a feasible point. This indeed occurs for cases where the number of scenarios added to the master problem is above 10: those problems are more difficult to solve for `Ipopt`.

6.4 Conclusion

In this chapter, we focus on nonlinear Adjustable Robust Optimization problems, specifically those with the same number of recourse variables as the number of nonlinear equality constraints in the second stage. This encompasses problems like the AR-OPF problem. To address these Adjustable Robust Optimization problems, we propose a reformulation as a semi-infinite programming problem. Since solving this semi-infinite programming reformulation using the adaptive discretization algorithm (Algorithm 1) is not scalable due to a very difficult separation problem, we propose an alternative discretization algorithm that relies on a deterministic sampling of the uncertainty set and uses a homotopy method to solve the nonlinear system of equations locally. The proposed algorithm exhibits convergence to either a local or a global minimum of the Adjustable Robust Optimization problem, depending on whether the master problems are solved locally or globally. In particular, although the homotopy method serves as a local equation solver during the screening of new uncertainty scenarios to incorporate, the algorithm achieves global convergence as long as the master problems are solved globally. In addition, finite convergence occurs for any positive feasibility tolerance.

To demonstrate the algorithm's effectiveness, we apply it to AR-OPF instances with up to 1354 buses to compute local solutions. This problem size is larger than similar work on the subject [129, 137] and is closer to the actual size of national-scale grids with thousands of buses. Note, however, that the master problems here were solved only locally with an interior-point method; one avenue to explore would be to combine this work with global optimization methods for ACOPF, such as the one proposed in Chapter 4, to compute global optimizers of the AR-OPF problem. This would likely come at a high computational cost. To perform such a global optimization, it could be interesting to opt for a single-tree method: the spatial branch-and-bound tree used to solve the master problem (P^k) could be continued for solving the master problem (P^{k+1}) at the next iteration. This would result in a sort of branch-and-bound-and-cut for solving the problem (SIP).

Our algorithm is based on a qualification condition for equality constraints (Assumption 6.2), which seems to be satisfied in practice on the vast majority of the instances we have used. However, we did find one network structure where this condition was not satisfied. A possible solution to address this singularity issue could be to incorporate an additional inequality constraint to the problem formulation to force the singular value of the Jacobian matrix to be strictly positive. The resulting research avenue would be to be able to handle such a singular value constraint into the optimization problem, possibly based on smooth spectral optimization techniques [144, 143].

A relevant research direction is to employ our approach in practical use cases incorporating uncertainty sets derived from real statistical data obtained from measurements. One possible application for a Transmission System Operator like Réseau de Transport d'Électricité could be the optimization of the primary and secondary reserves for frequency control.

Conclusion and perspectives

Conclusion

This thesis proposed several theoretical and practical contributions to the development of reliable algorithms for nonlinear semi-infinite optimization, especially for some problems arising in power systems optimization and optimal control. Various properties contribute to the reliability of an optimization algorithm, and we have sought to address these aspects in the thesis: the convergence to a global optimum, or at least the ability to certify an optimality gap; the ability to compute feasible points with respect to the infinite set of constraints; and the computational efficiency of the algorithm.

Contributions of Part I

Chapter 1 makes a theoretical contribution to analyzing the cutting-plane algorithm for convex semi-infinite optimization. We show that, if the objective function is strongly convex, the objective and feasibility error converges in $O(1/k)$ over the iterations of this algorithm, where k is the number of calls to the separation oracle. As it is known that this algorithm may have poor convergence properties in some (other) cases, the interest of our contribution is to give a sufficient condition (the strong convexity of the objective) for this algorithm to be efficient. Note that this convergence rate is valid even if the separation problem is nonconvex, even if the oracle solves the separation problem not to global optimality, but up to an optimality gap $\delta \in [0, 1)$, and even if the constraints in the master problems are aggregated to keep the master problems of constant size over the iterations of the algorithm. Chapter 3 may be seen as an application of convex semi-infinite programming: we introduce and solve a converging hierarchy of convex semi-infinite programs to compute approximate value functions of a minimum time control problem. We apply the cutting-plane algorithm to solve these semi-infinite programs after regularizing the objective function to have strong convexity. Based on the resulting approximate value function, we obtain certified lower bounds on the control problem and generate closed-loop controlled trajectories, yielding upper bounds. The originality of this approach, based on semi-infinite programming, is its ability to handle non-polynomial controlled dynamical systems.

Chapter 2 focuses on convex semi-infinite programs, where the separation problem is a

QCQP, that is potentially nonconvex. More specifically, we address the problem of computing a feasible point of this semi-infinite problem, i.e., a solution that satisfies an infinite number of inequalities. This issue is indeed a weak point of the cutting-plane algorithm, where feasibility is achieved only asymptotically. We propose a new algorithm, named Inner-Outer Approximation algorithm, that exploits the primal-dual structure of the separation problem to generate an inner approximation of the feasible set of the semi-infinite problem. We obtain a (globally) minimizing sequence of feasible iterates, as our algorithm uses the separation oracle output to tighten the outer approximation and extend the inner approximation of the feasible set of the semi-infinite problem. Another feature of this algorithm is the ability to detect *a posteriori* the convexity of the separation problem, at $x = \hat{x}^0$, the first iterate. This condition is sufficient for an early stop of the algorithm: in this case, it has found the optimal solution of the semi-infinite problem (even if the separation problem is nonconvex for other values of x). If this condition is met, this yields a clear advantage in computation time compared to three competing algorithms.

Contributions of Part II

Part II deals with nonconvex optimization problems arising in power systems operations. This part is dedicated to the global solution of the ACOPF, and, then, the solution of the AR-OPF problem, which fits the setting of nonconvex semi-infinite programming.

In Chapter 4, we address the standard ACOPF problem, a QCQP with a finite number of constraints. We propose a global optimization algorithm based on sparse semidefinite programming, novel nonlinear cuts, bound tightening, and adaptive piecewise linear relaxations: this leads to Mixed-Integer Conic Programming relaxations of the ACOPF problem that are progressively refined. In a semi-infinite discretization approach, these Mixed-Integer Conic Programming problems are approximated by MILP problems. The solutions of these MILP problems converge to a global optimum of the ACOPF. Our numerical experiments on a standard benchmark for the ACOPF, for small- to middle-scale instances, show that our algorithm outperforms three other global optimization approaches in terms of the number of instances solved to global optimality or below a certain optimization gap, and in terms of the computational time to global optimality.

Note that our global optimization algorithm for ACOPF relies on the ability to solve semidefinite programming relaxations many times during the bound tightening procedure. In Chapter 5, we address some numerical issues that arise when solving such semidefinite relaxations for large-scale ACOPF problems. Indeed, it is known that state-of-the-art interior-point methods often fail to converge when solving large-scale semidefinite relaxations. This causes two issues: the accuracy of the computed relaxation value and, therefore, the validity of this lower bound. We propose an unconstrained dual formulation that enables us to compute a certified lower bound from any dual vector, feasible or not, in the classical dual semidefinite problem. Using a structure-exploiting bundle method, we solve this unconstrained dual formulation starting from the dual vector given by the state-of-the-art interior-point solver MOSEK. Our extensive numerical experiments show that this post-processing significantly improves the computed lower bounds' quality, demonstrating a certain complementarity between this interior-point solver and the implemented bundle method.

In the last chapter of the dissertation, we study an Adjustable-Robust variant of the ACOPF problem to account for uncertainties. We address the AR-OPF problem by reformulating it as a nonlinear semi-infinite programming problem. Due to the challenging separation problem, solving this semi-infinite program with the standard adaptive discretization algorithm (Algorithm 1) is not scalable. As an alternative, we propose a discretization algorithm that employs deterministic sampling of the uncertainty set and utilizes a homotopy method to solve the nonlinear system of equations locally. The proposed algorithm converges to either a local or a global minimum for the adjustable robust optimization problem, depending on whether the master problems are solved locally or globally during the algorithm. Furthermore, finite convergence is guaranteed for any positive feasibility tolerance. To demonstrate the algorithm's scalability, we apply our algorithm to compute local solutions of AR-OPF instances with a maximum of 1354 buses. This problem size surpasses previous work in this area.

Perspectives

Perspectives for Part I

Regarding the first part of the thesis on convex semi-infinite programming, we identify several limitations that may lead to further work.

As regards the $O(1/k)$ convergence rate of the cutting-plane algorithm proved in Chapter 1, a limitation is the strong convexity assumption for the objective function. It could be interesting to study if one can relax it and derive weaker assumptions, also guaranteeing $O(1/k)$ convergence rate for the cutting-plane algorithm or for a variant. An extension of the current work could also be to prove this convergence rate for the cutting-plane algorithms applied to mixed-integer convex semi-infinite programming problems with a strongly convex objective function.

Another line of research concerns the rate of convergence of the Inner-Outer Approximation algorithm proposed in Chapter 2. From a theoretical point of view, a limitation of this chapter is indeed the absence of results regarding the speed of convergence. Could we prove that the objective value of the feasible iterates generated by the Inner-Outer Approximation algorithm converges to the optimal value of the semi-infinite program at a guaranteed rate? From a practical point of view, a limitation of the Inner-Outer Approximation algorithm is the computational cost of the master problem solved at each iteration, which is higher than the computational cost of the master problem of the standard cutting-plane algorithm due to the additional semidefinite constraint. To reduce the computational cost at each iteration of the algorithm, one can exploit the sparsity of the separation problem to obtain several small semidefinite constraints instead of one large one. One could also explore the possibility of doing a warm-start when solving the master problem at each iteration.

The semi-infinite programming approach for minimum time control could be improved by using other bases of functions to approximate the value function. This could improve the quality of approximation and, therefore, the quality of the lower bound and of the generated closed-loop trajectory. In particular, we could investigate using nonsmooth basis functions to approximate a potentially nonsmooth value function better.

Perspectives for Part II

We terminate with different limitations and perspectives regarding our work for the global optimization of the ACOPF problem and its Adjustable-Robust variant.

A first limitation of the state-of-the-art global optimization algorithm for the ACOPF problem that we propose in Chapter 4 is the scalability of the conic programming relaxation used in the bound tightening. This relaxation involves semidefinite constraints. For this algorithm to scale to problems with thousands of buses, we would rely on the progress of semidefinite solvers. For the bound tightening, it would also be relevant to study to what extent this task could be parallelized in practice, or one could target the bounds to tighten based on the graph structure. Concerning the sequence of MILP relaxations that are solved, the main limitation is that the branch-and-bound tree generated by the MILP solver to solve the k-th relaxations not re-used to solve the $k+1$-th relaxation, whereas this is a tightening of the previous relaxation. One could study the possibility of exploiting the information of the branch-and-bound tree at iteration k, to avoid some computations at iteration $k+1$. From an applicative point of view, our method could be extended to the Optimal Transmission Switching problem, where the network topology can be modified by switching electrical lines. We can also foresee an extension to Unit Commitment problems with AC power flow equations, where generators can be switched on or off. Indeed, the proposed MILP scheme could easily accommodate additional binary variables to describe these switches.

Regarding Chapter 5, the future lines of research are mainly practical. One could speed up the algorithm by parallelizing the evaluation of the spectral functions or by using a GPU implementation of the Quadratic Programming solver called at each iteration. The long-term objective is to obtain a large-scale, warm-startable, and reliable semidefinite programming solver, which could be used, for example, in the bound tightening procedure performed during our global optimization algorithm (Chapter 4). While we demonstrated that this bundle method can be warm-started and improves `MOSEK`'s accuracy in post-processing, it remains to be proven that such a first-order method could help save computational time.

Finally, the work on AR-OPF problem conducted in Chapter 6 could be improved by implementing a global optimization approach, such as this of Chapter 4, to solve the master problem at each iteration. This would enable one to compute global optimizers of the AR-

OPF problem instead of local minimizers as we did in Chapter 6. As aforementioned, for the MILP approximation scheme, it would probably be relevant to store the branch-and-bound tree generated when solving the master problem (P^k) to exploit this information (mainly the lower bound at each node) for the solution of the master problem (P^{k+1}) at the following iteration. This would result in a sort of branch-and-bound-and-cut for solving the semi-infinite programming problem. The proposed discretization algorithm based on deterministic sampling and homotopy continuation converges under a qualification condition for equality constraints (Assumption 6.2), which seems to be satisfied in practice on most of the instances we have used. However, we did find one example of a network where this condition was not satisfied. A possible solution to overcome this limitation of depending on this assumption, and to address such numerically difficult cases is to add the explicit constraint that the singular value of the Jacobian matrix must be strictly positive. The resulting research avenue would be to handle such a singular value constraint in the optimization problem based on smooth spectral optimization techniques. A last and necessary avenue of research is to calibrate this adaptive discretization algorithm for the AR-OPF problem on real test cases, such as reserve management for frequency control, with loads and renewables uncertainty datasets obtained from measurements.

Bibliography

[1] Farid Alizadeh.
Interior Point Methods in Semidefinite Programming with Applications to Combinatorial Optimization.
SIAM Journal on Optimization, 5(1):13–51, February 1995.
Publisher: Society for Industrial and Applied Mathematics.
(page 5)

[2] Gemayqzel Bouza Allende and Georg Still.
Solving bilevel programs with the KKT-approach.
Mathematical Programming, 138(1):309–332, April 2013.
(pages 4, 28, 29, 40, 41, and 42)

[3] Zahia Amrouchi, Frederic Messine, Clement Nadal, and Mohand Ouanes.
A deterministic semi-infinite global optimization method to design slotless permanent magnet machines.
COMPEL - The international journal for computation and mathematics in electrical and electronic engineering, 40(2):84–94, January 2021.
Publisher: Emerald Publishing Limited.
(page 5)

[4] Martin S. Andersen, Joachim Dahl, and Lieven Vandenberghe.
Implementation of nonsymmetric interior-point methods for linear optimization over sparse matrix cones.
Mathematical Programming Computation, 2(3-4):167–201, December 2010.
(page 103)

[5] Ioannis Androulakis, Costas Maranas, and Christodoulos Floudas.
BB: A global optimization method for general constrained nonconvex problems.
Journal of Global Optimization, 7(4):337–363, December 1995.
(page 122)

[6] Ignacio Aravena, Daniel K. Molzahn, Shixuan Zhang, Cosmin G. Petra, Frank E. Curtis, Shenyinying Tu, Andreas Wächter, Ermin Wei, Elizabeth Wong, Amin Gholami, Kaizhao Sun, Xu Andy Sun, Stephen T. Elbert, Jesse T. Holzer, and Arun Veeramany.
Recent Developments in Security-Constrained AC Optimal Power Flow: Overview of Challenge 1 in the ARPA-E Grid Optimization Competition, June 2022.
arXiv:2206.07843 [math].
(page 123)

[7] Jean-Pierre Aubin.
Viability Theory.
Systems & control: foundations & applications. Springer Science+Business Media, New York, 1991.
(pages 2, 60, 62, and 182)

[8] Jean-Pierre Aubin and Hélène Frankowska.
Set-Valued Analysis.
Birkhäuser, Boston, 2009.
(page 50)

[9] Sogol Babaeinejadsarookolaee, Adam Birchfield, Richard D. Christie, Carleton Coffrin, Christopher DeMarco, Ruisheng Diao, Michael Ferris, Stephane Fliscounakis, Scott Greene, Renke Huang, Cedric Josz, Roman Korab,

Bernard Lesieutre, Jean Maeght, Terrence W. K. Mak, Daniel K. Molzahn, Thomas J. Overbye, Patrick Panciatici, Byungkwon Park, Jonathan Snodgrass, Ahmad Tbaileh, Pascal Van Hentenryck, and Ray Zimmerman.
The Power Grid Library for Benchmarking AC Optimal Power Flow Algorithms.
arXiv:1908.02788 [math], January 2021.
arXiv: 1908.02788.
(pages 11, 12, 76, 77, 96, 103, and 116)

[10] Thomas Bagby, Len Bos, and Norman Levenberg.
Multivariate simultaneous approximation.
Constructive Approximation, 18(4):569–577, December 2002.
(page 181)

[11] Richard Bellman.
Dynamic programming.
Science, 153(3731):34–37, 1966.
Publisher: American Association for the Advancement of Science.
(page 50)

[12] Alexandre Belloni.
Lecture Notes for IAP 2005 : Course Introduction to Bundle Methods, 2005.
(page 112)

[13] Pietro Belotti, Sonia Cafieri, Jon Lee, and Leo Liberti.
Feasibility-Based Bounds Tightening via Fixed Points.
In Weili Wu and Ovidiu Daescu, editors, *Combinatorial Optimization and Applications*, pages 65–76, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
(page 76)

[14] Pietro Belotti, Jon Lee, Leo Liberti, François Margot, and Andreas Wächter.
Branching and bounds tighteningtechniques for non-convex MINLP.
Optimization Methods and Software, 24(4-5):597–634, October 2009.
Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10556780903087124.
(page 77)

[15] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski.
Adjustable robust solutions of uncertain linear programs.
Mathematical Programming, 99(2):351–376, March 2004.
(page 122)

[16] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovskiĭ.
Robust optimization.
Princeton series in applied mathematics. Princeton University Press, Princeton, 2009.
OCLC: ocn318672208.
(page 5)

[17] Aharon Ben-Tal, Omar El Housni, and Vineet Goyal.
A tractable approach for designing piecewise affine policies in two-stage adjustable robust optimization.
Mathematical Programming, 182(1):57–102, July 2020.
(page 123)

[18] Aharon Ben-Tal and Aharon Nemirovski.
Robust solutions of uncertain linear programs.
Operations Research Letters, 25(1):1–13, August 1999.
(pages 4 and 28)

[19] Eloïse Berthier, Justin Carpentier, Alessandro Rudi, and Francis Bach.
Infinite-dimensional sums-of-squares for optimal control.
In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 577–582. IEEE, 2022.
(page 48)

[20] Dimitris Bertsimas and Hoda Bidkhori.
On the performance of affine policies for two-stage adaptive optimization: a geometric perspective.

Mathematical Programming, 153(2):577–594, November 2015.
(page 122)

[21] Dimitris Bertsimas and Vineet Goyal.
On the power and limitations of affine policies in two-stage adaptive optimization.
Mathematical Programming, 134(2):491–531, September 2012.
(pages 122 and 123)

[22] Dimitris Bertsimas, Dan A. Iancu, and Pablo A. Parrilo.
Optimality of Affine Policies in Multistage Robust Optimization.
Mathematics of Operations Research, 35(2):363–394, May 2010.
Publisher: INFORMS.
(page 123)

[23] Ksenia Bestuzheva, Mathieu Besançon, Wei-Kun Chen, Antonia Chmiela, Tim Donkiewicz, Jasper van Doornmalen, Leon Eifler, Oliver Gaul, Gerald Gamrath, Ambros Gleixner, Leona Gottwald, Christoph Graczyk, Katrin Halbig, Alexander Hoen, Christopher Hojny, Rolf van der Hulst, Thorsten Koch, Marco Lübbecke, Stephen J. Maher, Frederic Matter, Erik Mühmer, Benjamin Müller, Marc E. Pfetsch, Daniel Rehfeldt, Steffan Schlein, Franziska Schlösser, Felipe Serrano, Yuji Shinano, Boro Sofranac, Mark Turner, Stefan Vigerske, Fabian Wegscheider, Philipp Wellner, Dieter Weninger, and Jakob Witzig.
The SCIP Optimization Suite 8.0.
Technical Report, Optimization Online, December 2021.
(page 66)

[24] Bruno Betro.
An accelerated central cutting plane algorithm for linear semi-infinite programming.
Mathematical Programming, 101(3):479–495, December 2004.
(pages 4 and 17)

[25] B. Bhattacharjee, P. Lemonidis, W.H. Green Jr., and P.I. Barton.
Global solution of semi-infinite programs.
Mathematical Programming, 103(2):283–307, June 2005.
(page 4)

[26] Binita Bhattacharjee.
Kinetic model reduction using integer and semi-infinite programming.
PhD Thesis, Massachusetts Institute of Technology, 2003.
(page 5)

[27] Dan Bienstock, Mauro Escobar, Claudio Gentile, and Leo Liberti.
Mathematical programming formulations for the alternating current optimal power flow problem.
4OR, 18(3):249–292, September 2020.
(pages 6 and 176)

[28] Daniel Bienstock, Michael Chertkov, and Sean Harnett.
Chance-Constrained Optimal Power Flow: Risk-Aware Network Control under Uncertainty.
SIAM Review, 56(3):461–495, January 2014.
(page 10)

[29] Daniel Bienstock and Gonzalo Munoz.
LP Formulations for Polynomial Optimization Problems.
SIAM Journal on Optimization, 28(2):1121–1150, January 2018.
Publisher: Society for Industrial and Applied Mathematics.
(page 8)

[30] Daniel Bienstock and Abhinav Verma.
Strong NP-hardness of AC power flows feasibility.
Operations Research Letters, 47(6):494–501, November 2019.
(page 8)

[31] Jerry Blankenship and James Falk.
Infinitely constrained optimization problems.

Journal of Optimization Theory and Applications, 19(2):261–281, June 1976.
(pages 3, 16, 28, 46, 124, and 126)

[32] Hans-Peter Blatt, Ulrich Kaiser, and B. Ruffer-Beedgen.
A Multiple Exchange Algorithm in Convex Programming.
In *Optimization*. CRC Press, 1983.
Num Pages: 18.
(pages 4 and 17)

[33] Jérôme Bolte and Edouard Pauwels.
Conservative set valued fields, automatic differentiation, stochastic gradient methods and deep learning.
Mathematical Programming, 188(1):19–51, July 2021.
(page 183)

[34] Loïc Bourdin and Emmanuel Trélat.
Pontryagin maximum principle for optimal sampled-data control problems.
IFAC-PapersOnLine, 48(25):80–84, January 2015.
(pages 47 and 48)

[35] Waqquas A. Bukhsh, Andreas Grothey, Ken I. M. McKinnon, and Paul A. Trodden.
Local Solutions of the Optimal Power Flow Problem.
IEEE Transactions on Power Systems, 28(4):4780–4788, November 2013.
Conference Name: IEEE Transactions on Power Systems.
(page 9)

[36] Richard H. Byrd, Jorge Nocedal, and Richard A. Waltz.
Knitro: An Integrated Package for Nonlinear Optimization.
In G. Di Pillo and M. Roma, editors, *Large-Scale Nonlinear Optimization*, Nonconvex Optimization and Its Applications, pages 35–59. Springer US, Boston, MA, 2006.
(pages 8, 9, and 76)

[37] Mary B Cain, Richard P O'neill, Anya Castillo, and others.
History of optimal power flow and formulations.
Federal Energy Regulatory Commission, 1:1–36, 2012.
Publisher: Citeseer.
(page 6)

[38] Eduardo F. Camacho and Carlos Bordons Alba.
Model predictive control.
Springer science & business media, 2013.
(page 48)

[39] Florin Capitanescu.
Critical review of recent advances and further developments needed in AC optimal power flow.
Electric Power Systems Research, 136:57–68, July 2016.
(page 123)

[40] Florin Capitanescu, Jose Luis Martinez Ramos, Patrick Panciatici, Daniel Kirschen, Alejandro Marano Marcolini, Ludovic Platbrood, and Louis Wehenkel.
State-of-the-art, challenges, and future trends in security constrained optimal power flow.
Electric Power Systems Research, 81(8):1731–1741, August 2011.
(page 123)

[41] Italo Capuzzo-Dolcetta.
Hamilton-Jacobi equations with state constraints.
Transactions of the American Mathematical Society, 318(2):643–683, 1990.
(page 47)

[42] Jacques Carpentier.
Contribution a l'étude du dispatching économique.
Bulletin de la Société Francaise des électriciens, 3(1):431–447, 1962.
(page 6)

[43] Jacques Carpentier.
Optimal power flows.
International Journal of Electrical Power & Energy Systems, 1(1):3–15, 1979.
Publisher: Elsevier.
(page 6)

[44] Martina Cerulli.
Bilevel optimization and applications.
Theses, Institut Polytechnique de Paris, December 2021.
Issue: 2021IPPAX108.
(page 5)

[45] Martina Cerulli, Diego Delle Donne, Mauro Escobar, Antoine Oustry, and Leo Liberti.
Optimal Power Flow, 2023.
(page 13)

[46] Martina Cerulli, Antoine Oustry, Claudia D'Ambrosio, and Leo Liberti.
Convergent Algorithms for a Class of Convex Semi-infinite Programs.
SIAM Journal on Optimization, 32(4):2493–2526, December 2022.
Publisher: Society for Industrial and Applied Mathematics.
(pages 12 and 126)

[47] Mohammadreza Chamanbaz, Fabrizio Dabbene, and Constantino M. Lagoa.
Probabilistically Robust AC Optimal Power Flow.
IEEE Transactions on Control of Network Systems, 6(3):1135–1147, September 2019.
Conference Name: IEEE Transactions on Control of Network Systems.
(page 124)

[48] Chen Chen, Alper Atamtürk, and Shmuel S. Oren.
A spatial branch-and-cut method for nonconvex QCQP with bounded complex variables.
Mathematical Programming, 165(2):549–577, October 2017.
(pages 77, 78, and 79)

[49] Tianran Chen and Tien-Yien Li.
Homotopy continuation method for solving systems of nonlinear and polynomial equations.
Communications in Information and Systems, 15(2):119–307, 2015.
Publisher: International Press of Boston.
(page 127)

[50] Elliott W. Cheney and Allen A. Goldstein.
Newton's method for convex programming and Tchebycheff approximation.
Numerische Mathematik, 1(1):253–268, December 1959.
(page 16)

[51] Frank H. Clarke.
Generalized gradients and applications.
Transactions of the American Mathematical Society, 205:247–262, 1975.
(pages vi, 62, and 183)

[52] Frank H. Clarke and Richard B. Vinter.
The Relationship between the Maximum Principle and Dynamic Programming.
SIAM Journal on Control and Optimization, 25(5):1291–1311, September 1987.
Publisher: Society for Industrial and Applied Mathematics.
(page 47)

[53] Carleton Coffrin, Russell Bent, Kaarthik Sundar, Yeesian Ng, and Miles Lubin.
PowerModels. JL: An Open-Source Framework for Exploring Power Flow Formulations.
In *2018 Power Systems Computation Conference (PSCC)*, pages 1–8, June 2018.
(page 97)

[54] Carleton Coffrin, Hassan L. Hijazi, and Pascal Van Hentenryck.
Strengthening Convex Relaxations with Bound Tightening for Power Network Optimization.

In Gilles Pesant, editor, *Principles and Practice of Constraint Programming*, Lecture Notes in Computer Science, pages 39–57, Cham, 2015. Springer International Publishing.
(page 9)

[55] Carleton Coffrin, Hassan L. Hijazi, and Pascal Van Hentenryck.
The QC Relaxation: A Theoretical and Computational Study on Optimal Power Flow.
IEEE Transactions on Power Systems, 31(4):3008–3018, 2016.
(page 9)

[56] Carleton Coffrin, Hassan L. Hijazi, and Pascal Van Hentenryck.
Strengthening the SDP Relaxation of AC Power Flows With Convex Envelopes, Bound Tightening, and Valid Inequalities.
IEEE Transactions on Power Systems, 32(5):3549–3558, September 2017.
Conference Name: IEEE Transactions on Power Systems.
(pages 76, 77, 83, 85, and 178)

[57] Benoît Colson, Patrice Marcotte, and Gilles Savard.
An overview of bilevel optimization.
Annals of operations research, 153:235–256, 2007.
Publisher: Springer.
(page 1)

[58] Ian D. Coope and Georges A. Watson.
A projected lagrangian algorithm for semi-infinite programming.
Mathematical Programming, 32(3):337–356, July 1985.
(page 4)

[59] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.
Introduction to algorithms.
MIT Press, Cambridge, Massachusetts London, England, third edition edition, 2009.
(page 86)

[60] Michael G. Crandall and Pierre-Louis Lions.
Viscosity Solutions of Hamilton-Jacobi Equations.
Transactions of the American Mathematical Society, 277(1):1–42, 1983.
Publisher: American Mathematical Society.
(pages 47 and 50)

[61] Bai Cui and Xu Andy Sun.
A New Voltage Stability-Constrained Optimal Power Flow Model: Sufficient Condition, SOCP Representation, and Relaxation.
IEEE Transactions on Power Systems, 33(5):5092–5102, September 2018.
arXiv: 1705.10372.
(page 10)

[62] Emiliano Dall'Anese, Kyri Baker, and Tyler Summers.
Chance-Constrained AC Optimal Power Flow for Distribution Systems With Renewables.
IEEE Transactions on Power Systems, 32(5):3427–3438, September 2017.
Conference Name: IEEE Transactions on Power Systems.
(page 10)

[63] Olivier Devolder, François Glineur, and Yurii Nesterov.
First-order methods of smooth convex optimization with inexact oracle.
Mathematical Programming, 146(1-2):37–75, August 2014.
(page 16)

[64] Moritz Diehl, Boris Houska, Oliver Stein, and Paul Steuermann.
A lifting method for generalized semi-infinite programs based on lower level Wolfe duality.
Computational Optimization and Applications, 54(1):189–210, January 2013.
(pages 4, 28, and 29)

[65] Hatim Djelassi and Alexander Mitsos.
A hybrid discretization algorithm with guaranteed feasibility for the global solution of semi-infinite programs.
Journal of Global Optimization, 68(2):227–253, June 2017.
(pages 4, 17, and 28)

[66] Hatim Djelassi and Alexander Mitsos.
Global Solution of Semi-infinite Programs with Existence Constraints.
Journal of Optimization Theory and Applications, 188(3):863–881, March 2021.
(pages 122 and 123)

[67] Hatim Djelassi, Alexander Mitsos, and Oliver Stein.
Recent advances in nonconvex semi-infinite programming: Applications and algorithms.
EURO Journal on Computational Optimization, 9:100006, 2021.
(page 5)

[68] Yu Du and Andrzej Ruszczyński.
Rate of convergence of the bundle method.
Journal of Optimization Theory and Applications, 173:908–922, 2017.
Publisher: Springer.
(page 16)

[69] Laurent El Ghaoui, Francois Oustry, and Hervé Lebret.
Robust Solutions to Uncertain Semidefinite Programs.
SIAM Journal on Optimization, 9(1):33–52, January 1998.
(page 5)

[70] Anders Eltved, Joachim Dahl, and Martin S. Andersen.
On the Robustness and Scalability of Semidefinite Relaxation for Optimal Power Flow Problems.
Optimization and Engineering, 21(2):375–392, June 2020.
arXiv: 1806.08620.
(pages 102, 103, and 116)

[71] Jack Elzinga and Thomas G. Moore.
A central cutting plane algorithm for the convex programming problem.
Mathematical Programming, 8(1):134–145, December 1975.
(page 17)

[72] Stéphane Fliscounakis, Patrick Panciatici, Florin Capitanescu, and Louis Wehenkel.
Contingency Ranking With Respect to Overloads in Very Large Power Systems Taking Into Account Uncertainty, Preventive, and Corrective Actions.
IEEE Transactions on Power Systems, 28(4):4909–4917, November 2013.
Conference Name: IEEE Transactions on Power Systems.
(page 10)

[73] Christodoulos A. Floudas, Zeynep H. Gümüş, and Marianthi G. Ierapetritou.
Global Optimization in Design under Uncertainty: Feasibility Test and Flexibility Index Problems.
Industrial & Engineering Chemistry Research, 40(20):4267–4282, October 2001.
(page 122)

[74] Christodoulos A. Floudas and Oliver Stein.
The Adaptive Convexification Algorithm: A Feasible Point Method for Semi-Infinite Programming.
SIAM Journal on Optimization, 18(4):1187–1208, January 2008.
Publisher: Society for Industrial and Applied Mathematics.
(pages 4 and 28)

[75] Robert Fourer, David M Gay, and Brian W Kernighan.
AMPL: A mathematical programming language.
AT & T Bell Laboratories Murray Hill, NJ, 1987.
(page 42)

[76] Halina Frankowska.
Optimal trajectories associated with a solution of the contingent Hamilton-Jacobi equation.

Applied Mathematics and Optimization, 19(1):291–311, January 1989.
(page 47)

[77] Lingwen Gan, Na Li, Ufuk Topcu, and Steven H. Low.
Exact Convex Relaxation of Optimal Power Flow in Radial Networks.
IEEE Transactions on Automatic Control, 60(1):72–87, January 2015.
arXiv: 1311.7170.
(page 9)

[78] Carlos B. Garcia and Willard J. Zangwill.
Pathways to Solutions, Fixed Points, and Equilibria.
SIAM Review, 26(3):445–446, July 1984.
Publisher: Society for Industrial and Applied Mathematics.
(page 8)

[79] Lawrence Craig Evans Gariepy, Ronald F.
Measure Theory and Fine Properties of Functions, Revised Edition.
Chapman and Hall/CRC, New York, April 2015.
(pages 61 and 183)

[80] Amin Gholami, Kaizhao Sun, Shixuan Zhang, and Xu Andy Sun.
An ADMM-based Distributed Optimization Method for Solving Security-Constrained AC Optimal Power Flow, August 2022.
arXiv:2202.06787 [math].
(pages 123 and 137)

[81] Jean Charles Gilbert and Cédric Josz.
Plea for a semidefinite optimization solver in complex numbers.
HAL Archives-Ouvertes, page 36, 2017.
(pages 107 and 116)

[82] Philip E. Gill, Walter Murray, and Michael A. Saunders.
SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization.
SIAM Review, 47(1):99–131, January 2005.
Publisher: Society for Industrial and Applied Mathematics.
(page 9)

[83] Miguel A. Goberna and Marco A. López.
Recent contributions to linear semi-infinite optimization: an update.
Annals of Operations Research, 271(1):237–278, December 2018.
(page 17)

[84] Miguel Á. Goberna, Marco A. López, and Panos Pardalos, editors.
Semi-Infinite Programming, volume 57 of *Nonconvex Optimization and Its Applications.*
Springer US, Boston, MA, 2001.
(page 1)

[85] Hadrien Godard.
Résolution exacte du problème de l'optimisation des flux de puissance.
PhD thesis, Conservatoire National des Arts et Métiers, Paris, 2019.
(pages 9 and 77)

[86] Hadrien Godard, Sourour Elloumi, Amélie Lambert, Jean Maeght, and Manuel Ruiz.
Novel Approach Towards Global Optimality of Optimal Power Flow Using Quadratic Convex Optimization.
In *Proceedings of the 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, Paris, 2019.
(page 77)

[87] Smitha Gopinath and Hassan L. Hijazi.
Benchmarking Large-Scale ACOPF Solutions and Optimality Bounds.
In *2022 IEEE Power & Energy Society General Meeting (PESGM)*, pages 1–5, July 2022.
ISSN: 1944-9933.

(pages 76, 77, 97, 99, and 101)

[88] Smitha Gopinath, Hassan L. Hijazi, Tillmann Weisser, Harsha Nagarajan, Mertcan Yetkin, Kaarthik Sundar, and Russel W. Bent.
Proving global optimality of ACOPF solutions.
Electric Power Systems Research, 189:106688, December 2020.
(page 76)

[89] Paul R. Gribik.
A central-cutting-plane algorithm for semi-infinite programming problems.
In Rainer Hettich, editor, *Semi-Infinite Programming*, volume 15, pages 66–82. Springer-Verlag, Berlin/Heidelberg, 1979.
Series Title: Lecture Notes in Control and Information Sciences.
(pages 4 and 17)

[90] Paul R. Gribik and Kenneth O. Kortanek.
Equivalence Theorems and Cutting Plane Algorithms for a Class of Experimental Design Problems.
SIAM Journal on Applied Mathematics, 32(1):232–259, 1977.
_eprint: https://doi.org/10.1137/0132021.
(page 5)

[91] Robert Grone, Charles R. Johnson, Eduardo M. Sá, and Henry Wolkowicz.
Positive definite completions of partial Hermitian matrices.
Linear Algebra and its Applications, 58:109–124, April 1984.
(pages 82, 102, and 106)

[92] Ignacio E. Grossmann and Keshava P. Halemane.
Decomposition strategy for designing flexible chemical plants.
AIChE Journal, 28(4):686–694, 1982.
_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/aic.690280422.
(page 122)

[93] Ignacio E. Grossmann, Keshava P. Halemane, and Ross E. Swaney.
Optimization strategies for flexible chemical processes.
Computers & Chemical Engineering, 7(4):439–462, January 1983.
(page 122)

[94] LLC Gurobi Optimization.
Gurobi Optimizer Reference Manual, 2021.
(pages 9, 42, 66, 77, 97, and 101)

[95] Napsu Haarala, Kaisa Miettinen, and Marko M. Mäkelä.
New limited memory bundle method for large-scale nonsmooth optimization.
Optimization Methods and Software, 19(6):673–692, December 2004.
Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10556780410001689225.
(page 21)

[96] Keshava P. Halemane and Ignacio E. Grossmann.
Optimal process design under uncertainty.
AIChE Journal, 29(3):425–433, 1983.
_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/aic.690290312.
(page 122)

[97] William E. Hart, Jean-Paul Watson, and David L. Woodruff.
Pyomo: modeling and solving mathematical programs in Python.
Mathematical Programming Computation, 3(3):219, August 2011.
(page 96)

[98] Kris Hauser.
Semi-infinite programming for trajectory optimization with non-convex obstacles.
The International Journal of Robotics Research, 40(10-11):1106–1122, September 2021.
Publisher: SAGE Publications Ltd STM.

(page 5)

[99] Christoph Helmberg and Franz Rendl.
A Spectral Bundle Method for Semidefinite Programming.
SIAM Journal on Optimization, 10(3):673–696, January 2000.
(page 103)

[100] Christoph Helmberg, Franz Rendl, Robert J. Vanderbei, and Henry Wolkowicz.
An Interior-Point Method for Semidefinite Programming.
SIAM Journal on Optimization, 6(2):342–361, May 1996.
Publisher: Society for Industrial and Applied Mathematics.
(page 5)

[101] Didier Henrion, Jean-Bernard Lasserre, and Carlo Savorgnan.
Nonlinear optimal control synthesis via occupation measures.
In *2008 47th IEEE Conference on Decision and Control*, pages 4749–4754, December 2008.
ISSN: 0191-2216.
(pages 49 and 59)

[102] Daniel Hernández-Hernández, Onésimo Hernández-Lerma, and Michael Taksar.
The linear programming approach to deterministic optimal control problems.
Applicationes Mathematicae, 24(1):17–33, 1996.
(page 48)

[103] Rainer Hettich.
An implementation of a discretization method for semi-infinite programming.
Mathematical Programming, 34(3):354–361, April 1986.
(pages 4 and 17)

[104] Rainer Hettich and Kenneth O. Kortanek.
Semi-Infinite Programming: Theory, Methods, and Applications.
SIAM Review, 35(3):380–429, September 1993.
(pages 1, 4, and 17)

[105] Hassan Hijazi, Guanglei Wang, and Carleton Coffrin.
Gravity: A Mathematical Modeling Language for Optimization and Machine Learning.
In *Proceedings of NIPS 2018 Workshop MLOSS*, Montreal, Canada, October 2018.
(page 97)

[106] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal.
Convex Analysis and Minimization Algorithms I, volume 305 of *Grundlehren der mathematischen Wissenschaften*.
Springer Berlin Heidelberg, Berlin, Heidelberg, 1993.
(pages 19 and 33)

[107] Lars Hörmander.
The Analysis of Linear Partial Differential Operators I.
Classics in Mathematics. Springer, Berlin, Heidelberg, 2003.
(page 53)

[108] IBM.
IBM ILOG CPLEX 12.7 User's Manual, 2017.
(page 96)

[109] Rabih A. Jabr.
Radial distribution load flow using conic programming.
IEEE Transactions on Power Systems, 21(3):1458–1459, August 2006.
Conference Name: IEEE Transactions on Power Systems.
(page 9)

[110] Rabih A. Jabr.
Adjustable Robust OPF With Renewable Energy Sources.
IEEE Transactions on Power Systems, 28(4):4742–4751, November 2013.

Conference Name: IEEE Transactions on Power Systems.
(page 123)

[111] Martin Jaggi.
Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization.
In *International Conference on Machine Learning*, pages 427–435. PMLR, February 2013.
ISSN: 1938-7228.
(page 25)

[112] Nasser Jaleeli, Louis S. VanSlyck, Donald N. Ewart, Lester H. Fink, and A. G. Hoffmann.
Understanding automatic generation control.
IEEE Transactions on Power Systems, 7(3):1106–1122, August 1992.
Conference Name: IEEE Transactions on Power Systems.
(page 136)

[113] Reza Jazar.
Theory of Applied Robotics: Kinematics, Dynamics, and Control (2nd Edition).
Springer, New York, NY, 2010.
(page 47)

[114] Steven G. Johnson and Julien Schueller.
NLopt: Nonlinear optimization library, November 2021.
ADS Bibcode: 2021ascl.soft11004J.
(page 9)

[115] Morgan Jones and Matthew M. Peet.
Polynomial Approximation of Value Functions and Nonlinear Controller Design with Performance Bounds, January 2023.
arXiv:2010.06828 [cs, eess, math].
(pages 48, 49, and 59)

[116] Cédric Josz, Stéphane Fliscounakis, Jean Maeght, and Patrick Panciatici.
AC Power Flow Data in MATPOWER and QCQP Format: iTesla, RTE Snapshots, and PEGASE.
arXiv:1603.01533 [math], March 2016.
arXiv: 1603.01533.
(pages 8, 9, 76, 102, and 103)

[117] Cédric Josz and Daniel K. Molzahn.
Lasserre Hierarchy for Large Scale Polynomial Optimization in Real and Complex Variables.
SIAM Journal on Optimization, 28(2):1017–1048, 2018.
_eprint: https://doi.org/10.1137/15M1034386.
(pages 6 and 9)

[118] James E. Kelley.
The Cutting-Plane Method for Solving Convex Programs.
Journal of the Society for Industrial and Applied Mathematics, 8(4):703–712, 1960.
Publisher: Society for Industrial and Applied Mathematics.
(pages 3, 15, 16, 28, 46, and 114)

[119] Leonid Genrikhovich Khachiyan.
A polynomial algorithm in linear programming.
In *Doklady Akademii Nauk*, volume 244, pages 1093–1096. Russian Academy of Sciences, 1979.
Issue: 5.
(page 16)

[120] Kibaek Kim, Youngdae Kim, Daniel A. Maldonado, Michel Schanen, Victor M. Zavala, and Nai-Yuan Chiang.
A Scalable Mixed-Integer Decomposition Method for Security-Constrained Optimal Power Flow with Complementarity Constraints.
Technical Report ANL-21/66, Argonne National Lab. (ANL), Argonne, IL (United States), November 2021.
(pages 10 and 123)

[121] Krzysztof Czesław Kiwiel.
An aggregate subgradient method for nonsmooth convex minimization.
Mathematical Programming, 27(3):320–341, October 1983.
(pages 16 and 21)

[122] Burak Kocuk, Santanu S. Dey, and X. Andy Sun.
Strong SOCP Relaxations for the Optimal Power Flow Problem.
Operations Research, 64(6):1177–1196, December 2016.
(pages 9, 76, and 83)

[123] Burak Kocuk, Santanu S. Dey, and X. Andy Sun.
Matrix minor reformulation and SOCP-based spatial branch-and-cut method for the AC optimal power flow problem.
Mathematical Programming Computation, 10(4):557–596, December 2018.
(page 77)

[124] Burak Kocuk, Santanu S. Dey, and Xu Andy Sun.
Inexactness of SDP Relaxation and Valid Inequalities for Optimal Power Flow.
IEEE Transactions on Power Systems, 31(1):642–651, January 2016.
Conference Name: IEEE Transactions on Power Systems.
(pages 76 and 83)

[125] Hidetoshi Komiya.
Elementary proof for Sion's minimax theorem.
Kodai Mathematical Journal, 11(1), January 1988.
(page 88)

[126] Milan Korda.
Computing controlled invariant sets from data using convex optimization.
SIAM Journal on Control and Optimization, 58(5):2871–2899, January 2020.
arXiv: 1912.03256.
(page 48)

[127] Milan Korda, Didier Henrion, and Colin N. Jones.
Convergence rates of moment-sum-of-squares hierarchies for optimal control problems.
Systems & Control Letters, 100:1–5, February 2017.
(page 48)

[128] Kenneth O. Kortanek and Vladimir G. Medvedev.
Semi-infinite Programming and Applications in Finance.
In Christodoulos A. Floudas and Panos M. Pardalos, editors, *Encyclopedia of Optimization*, pages 3396–3404. Springer US, Boston, MA, 2009.
(page 5)

[129] Olga Kuryatnikova, Bissan Ghaddar, and Daniel K. Molzahn.
Two-Stage Robust Quadratic Optimization with Equalities and its Application to Optimal Power Flow, January 2023.
arXiv:2104.03107 [math].
(pages 10, 123, 124, 136, and 144)

[130] Jean-Bernard Lasserre.
Global Optimization with Polynomials and the Problem of Moments.
SIAM Journal on Optimization, 11(3):796–817, January 2001.
(page 76)

[131] Jean-Bernard Lasserre.
Convergent SDP-Relaxations in Polynomial Optimization with Sparsity.
SIAM Journal on Optimization, 17(3):822–843, January 2006.
(pages 9 and 77)

[132] Jean-Bernard Lasserre.
Min-max and robust polynomial optimization.
Journal of Global Optimization, 51(1):1–10, September 2011.

(pages 4 and 28)

[133] Jean-Bernard Lasserre.
An introduction to polynomial and semi-algebraic optimization, volume 52.
Cambridge University Press, 2015.
(page 9)

[134] Jean-Bernard Lasserre, Didier Henrion, Christophe Prieur, and Emmanuel Trélat.
Nonlinear Optimal Control via Occupation Measures and LMI-Relaxations.
SIAM Journal on Control and Optimization, 47(4):1643–1666, January 2008.
(pages 48, 50, and 52)

[135] Javad Lavaei and Steven H. Low.
Zero Duality Gap in Optimal Power Flow Problem.
IEEE Transactions on Power Systems, 27(1):92–107, February 2012.
(pages 9 and 102)

[136] Dongchan Lee, Hung D. Nguyen, Krishnamurthy Dvijotham, and Konstantin Turitsyn.
Convex Restriction of Power Flow Feasibility Sets.
IEEE Transactions on Control of Network Systems, 6(3):1235–1245, September 2019.
Conference Name: IEEE Transactions on Control of Network Systems.
(page 124)

[137] Dongchan Lee, Konstantin Turitsyn, Daniel Kenneth Molzahn, and Line Roald.
Robust AC Optimal Power Flow with Robust Convex Restriction.
IEEE Transactions on Power Systems, pages 1–1, 2021.
(pages 10, 123, 124, 136, and 144)

[138] Claude Lemaréchal.
Nonsmooth Optimization and Descent Methods, March 1978.
Num Pages: 25 Place: IIASA, Laxenburg, Austria Publisher: RR-78-004.
(page 16)

[139] Claude Lemaréchal, Arkadii Nemirovskii, and Yurii Nesterov.
New variants of bundle methods.
Mathematical Programming, 69(1):111–147, July 1995.
(page 16)

[140] Claude Lemaréchal and Claudia Sagastizábal.
Variable metric bundle methods: From conceptual to implementable forms.
Mathematical Programming, 76(3):393–410, March 1997.
(pages 16 and 115)

[141] Mirko Leomanni, Gabriele Costante, and Francesco Ferrante.
Time-Optimal Control of a Multidimensional Integrator Chain With Applications.
IEEE Control Systems Letters, 6:2371–2376, 2022.
Conference Name: IEEE Control Systems Letters.
(page 48)

[142] Evgeni Levitin and Rainer Tichatschke.
A Branch-and-Bound Approach for Solving a Class of Generalized Semi-infinite Programming Problems.
Journal of Global Optimization, 13(3):299–315, October 1998.
(pages 4 and 29)

[143] Adrian S. Lewis.
The mathematics of eigenvalue optimization.
Mathematical Programming, 97(1):155–176, July 2003.
(page 144)

[144] Adrian S. Lewis and Hristo S. Sendov.
Twice Differentiable Spectral Functions.
SIAM Journal on Matrix Analysis and Applications, 23(2):368–386, January 2001.

Publisher: Society for Industrial and Applied Mathematics.
(page 144)

[145] Leo Liberti, Benedetto Manca, Antoine Oustry, and Pierre-Louis Poirion.
Random projections for semidefinite programming *.
In *AIRO-ODS 2022*, Florence, Italy, August 2022.
(page 13)

[146] Daniel Liberzon.
Calculus of Variations and Optimal Control Theor – A Concise Introduction.
Princeton University Press, Princeton ; Oxford, January 2012.
(page 48)

[147] Francesco Locatello, Michael Tschannen, Gunnar Rätsch, and Martin Jaggi.
Greedy algorithms for cone constrained optimization with convergence guarantees.
In *Advances in Neural Information Processing Systems*, volume 30, 2017.
(page 21)

[148] Jérôme Lohéac and Jean-François Scheid.
Time-optimal control for a perturbed Brockett integrator.
In *Variatonal methods*, pages 454–472. De Gruyter, January 2017.
(page 72)

[149] Raphael Louca and Eilyan Bitar.
Stochastic AC optimal power flow with affine recourse.
In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 2431–2436, Las Vegas, NV, USA, December 2016. IEEE.
(page 123)

[150] Raphael Louca and Eilyan Bitar.
Robust AC Optimal Power Flow.
IEEE Transactions on Power Systems, 34(3):1669–1681, May 2019.
Conference Name: IEEE Transactions on Power Systems.
(pages 10 and 123)

[151] Steven H. Low.
Convex Relaxation of Optimal Power Flow—Part II: Exactness.
IEEE Transactions on Control of Network Systems, 1(2):177–189, June 2014.
Conference Name: IEEE Transactions on Control of Network Systems.
(page 9)

[152] Mowen Lu, Harsha Nagarajan, Russell Bent, Sandra D. Eksioglu, and Scott J. Mason.
Tight Piecewise Convex Relaxations for Global Optimization of Optimal Power Flow.
In *2018 Power Systems Computation Conference (PSCC)*, pages 1–7, Dublin, Ireland, June 2018. IEEE.
(page 77)

[153] Marco López and Georg Still.
Semi-infinite programming.
European Journal of Operational Research, 180(2):491–518, July 2007.
(page 5)

[154] Ramtin Madani, Abdulrahman Kalbat, and Javad Lavaei.
ADMM for sparse semidefinite programming with applications to optimal power flow problem.
In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 5932–5939, Osaka, December 2015. IEEE.
(page 103)

[155] Ngoc Hoang Anh Mai, Jean-Bernard Lasserre, and Victor Magron.
A hierarchy of spectral relaxations for polynomial optimization.
Mathematical Programming Computation, May 2023.
(page 120)

[156] Gianandrea Mannarini, Deepak N. Subramani, Pierre F. J. Lermusiaux, and Nadia Pinardi.

Graph-Search and Differential Equations for Time-Optimal Vessel Route Planning in Dynamic Ocean Waves.
IEEE Transactions on Intelligent Transportation Systems, 21(8):3581–3593, August 2020.
(page 47)

[157] Antoine Marendet, Alexandre Goldsztejn, Gilles Chabert, and Christophe Jermann.
A standard branch-and-bound approach for nonlinear semi-infinite problems.
European Journal of Operational Research, 282(2):438–452, April 2020.
(page 4)

[158] Garth P. McCormick.
Computability of global solutions to factorable nonconvex programs: Part I — Convex underestimating problems.
Mathematical Programming, 10(1):147–175, December 1976.
(page 81)

[159] Dhagash Mehta, Daniel K. Molzahn, and Konstantin Turitsyn.
Recent advances in computational methods for the power flow equations.
In *2016 American Control Conference (ACC)*, pages 1753–1765, Boston, MA, USA, July 2016. IEEE.
(page 8)

[160] Michael R. Metel and Akiko Takeda.
Primal-dual subgradient method for constrained convex optimization problems.
Optimization Letters, 15(4):1491–1504, June 2021.
(page 16)

[161] Alexander Mitsos.
Global optimization of semi-infinite programs via restriction of the right-hand side.
Optimization, 60(10-11):1291–1308, October 2011.
(pages 4, 17, 28, 29, 40, 41, 42, 46, 124, and 126)

[162] Alexander Mitsos and Angelos Tsoukalas.
Global optimization of generalized semi-infinite programs via restriction of the right hand side.
Journal of Global Optimization, 61(1):1–17, January 2015.
(page 28)

[163] Daniel K. Molzahn and Ian A. Hiskens.
Sparsity-Exploiting Moment-Based Relaxations of the Optimal Power Flow Problem.
IEEE Transactions on Power Systems, 30(6):3168–3180, November 2015.
Conference Name: IEEE Transactions on Power Systems.
(page 76)

[164] Daniel K. Molzahn and Ian A. Hiskens.
Convex Relaxations of Optimal Power Flow Problems: An Illustrative Example.
IEEE Transactions on Circuits and Systems I: Regular Papers, 63(5):650–660, May 2016.
(pages 102, 103, 105, and 116)

[165] Daniel K. Molzahn and Ian A. Hiskens.
A Survey of Relaxations and Approximations of the Power Flow Equations.
Foundations and Trends® in Electric Energy Systems, 4(1-2):1–221, 2019.
(pages 6, 9, 76, 79, and 97)

[166] Daniel K. Molzahn, Jesse T. Holzer, Bernard C. Lesieutre, and Christopher L. DeMarco.
Implementation of a Large-Scale Optimal Power Flow Solver Based on Semidefinite Programming.
IEEE Transactions on Power Systems, 28(4):3987–3998, November 2013.
(page 102)

[167] Daniel K. Molzahn, Cedric Josz, and Ian A. Hiskens.
Moment relaxations of optimal power flow problems: Beyond the convex hull.
In *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 856–860, December 2016.
(page 76)

[168] Mosek.
The mosek optimization software, 2019.
(pages 5, 9, 42, 96, 103, 106, and 116)

[169] T. Mühlpfordt, L. Roald, V. Hagenmeyer, T. Faulwasser, and S. Misra.
Chance-Constrained AC Optimal Power Flow – A Polynomial Chaos Approach.
arXiv:1903.11337 [cs], April 2019.
arXiv: 1903.11337.
(page 10)

[170] Yurii Nesterov.
Introductory Lectures on Convex Optimization: A Basic Course.
Springer Science & Business Media, December 2003.
(page 16)

[171] Yurii Nesterov.
Primal-dual subgradient methods for convex problems.
Mathematical Programming, 120(1):221–259, August 2009.
(page 16)

[172] Geert J. Olsder.
Time-optimal control of multivariable systems near the origin.
Journal of optimization theory and applications, 16:497–517, 1975.
Publisher: Springer.
(page 48)

[173] Antoine Oustry.
AC Optimal Power Flow: a Conic Programming relaxation and an iterative MILP scheme for Global Optimization.
Open Journal of Mathematical Optimization, 3:1–19, 2022.
(page 12)

[174] Antoine Oustry, Carmen Cardozo, Patrick Pantiatici, and Didier Henrion.
Maximal Positively Invariant Set Determination for Transient Stability Assessment in Power Systems.
In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 6572–6577, December 2019.
ISSN: 2576-2370.
(page 10)

[175] Antoine Oustry and Martina Cerulli.
Semi-infinite programming solution algorithms with inexact separation oracles, 2023.
(page 13)

[176] Antoine Oustry, Claudia D'Ambrosio, Leo Liberti, and Manuel Ruiz.
Certified and accurate SDP bounds for the ACOPF problem.
Electric Power Systems Research, 212:108278, November 2022.
(pages 12 and 79)

[177] Antoine Oustry, Bünyamin Erkan, Romain Svartzman, and Pierre-François Weber.
Climate-related risks and central banks' collateral policy: a methodological experiment.
Revue economique, 73(2):173–218, April 2022.
Bibliographie_available: 1 Cairndomain: www.cairn-int.info Cite Par_available: 0 Publisher: Presses de Sciences Po.
(page 13)

[178] Antoine Oustry, Marion Le Tilly, Thomas Clausen, Claudia D'Ambrosio, and Leo Liberti.
Optimal deployment of indoor wireless local area networks.
Networks, 81(1):23–50, 2023.
_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/net.22116.
(page 13)

[179] Antoine Oustry, François Pacaud, and Mihai Anitescu.
Adjustable robust nonlinear optimization via semi-infinite programming, 2023.
(page 13)

[180] Antoine Oustry and Matteo Tacchi.
Minimal-time nonlinear control via semi-infinite programming, July 2023.
arXiv:2307.00857 [math].
(page 13)

[181] Antoine Oustry, Matteo Tacchi, and Didier Henrion.
Inner Approximations of the Maximal Positively Invariant Set for Polynomial Dynamical Systems.
IEEE Control Systems Letters, 3(3):733–738, July 2019.
Conference Name: IEEE Control Systems Letters.
(pages 10 and 48)

[182] Antoine Oustry, Liding Xu, Sonia Haddad-Vanier, Juan-Antonio Cordero, and Thomas Clausen.
Optimization in Wireless Networks, 2023.
(page 13)

[183] François Oustry.
A second-order bundle method to minimize the maximum eigenvalue function.
Mathematical Programming, 89(1):1–33, November 2000.
(pages 16 and 103)

[184] Brendan O'Donoghue, Eric Chu, Neal Parikh, and Stephen Boyd.
Conic Optimization via Operator Splitting and Homogeneous Self-Dual Embedding.
Journal of Optimization Theory and Applications, 169(3):1042–1068, June 2016.
(page 103)

[185] François Pacaud, Michel Schanen, Sungho Shin, Daniel Adrian Maldonado, and Mihai Anitescu.
Parallel Interior-Point Solver for Block-Structured Nonlinear Programs on SIMD/GPU Architectures, January 2023.
arXiv:2301.04869 [math].
(page 123)

[186] Ana I. Pereira, M. Fernanda P. Costa, and Edite M.G.P. Fernandes.
Interior point filter method for semi-infinite programming problems.
Optimization, 60(10-11):1309–1338, October 2011.
(page 4)

[187] Hans Josef Pesch.
A Practical Guide To The Solution Of Real-Life Optimal Control Problems.
Control and Cybernetics, 23, July 1996.
(page 47)

[188] Cosmin G. Petra and Ignacio Aravena.
Solving realistic security-constrained optimal power flow problems, October 2021.
arXiv:2110.01669 [math].
(page 123)

[189] Dzung T. Phan.
Lagrangian Duality and Branch-and-Bound Algorithms for Optimal Power Flow.
Operations Research, 60(2):275–285, April 2012.
(page 103)

[190] Elijah Polak.
An implementable algorithm for the optimal design centering, tolerancing, and tuning problem.
Journal of Optimization Theory and Applications, 37(1):45–67, May 1982.
(page 5)

[191] Elijah Polak.
Optimization: algorithms and consistent approximations, volume 124.
Springer Science & Business Media, 2012.
(page 17)

[192] Elijah Polak and Johannes O. Royset.
Algorithms for Finite and Semi-Infinite Min–Max–Min Problems Using Adaptive Smoothing Techniques.
Journal of Optimization Theory and Applications, 119(3):421–457, December 2003.
(page 122)

[193] Lev S. Pontryagin.
Mathematical Theory of Optimal Processes.

CRC Press, March 1987.
Google-Books-ID: kwzq0F4cBVAC.
(page 47)

[194] Joshua L. Pulsipher, Daniel Rios, and Victor M. Zavala.
A Computational Framework for Quantifying and Analyzing System Flexibility.
Computers & Chemical Engineering, 126:342–355, July 2019.
arXiv:2106.12706 [cs, eess, math].
(page 122)

[195] Rembert Reemtsen.
Discretization methods for the solution of semi-infinite programming problems.
Journal of Optimization Theory and Applications, 71(1):85–103, October 1991.
(pages 4 and 17)

[196] Rembert Reemtsen and Stephan Görner.
Numerical Methods for Semi-Infinite Programming: A Survey.
In Rembert Reemtsen and Jan-J. Rückmann, editors, *Semi-Infinite Programming*, Nonconvex Optimization and Its Applications, pages 195–275. Springer US, Boston, MA, 1998.
(page 17)

[197] Line A. Roald, David Pozo, Anthony Papavasiliou, Daniel K. Molzahn, Jalal Kazempour, and Antonio Conejo.
Power systems optimization under uncertainty: A review of methods and applications.
Electric Power Systems Research, 214:108725, January 2023.
(pages 10, 121, and 123)

[198] Gianpaolo Romano.
New results in subdifferential calculus with applications to convex optimization.
Applied Mathematics & Optimization, 32(3):213–234, November 1995.
(page 19)

[199] Sebastian Sager, Mathieu Claeys, and Frédéric Messine.
Efficient upper and lower bounds for global mixed-integer optimal control.
Journal of Global Optimization, 61(4):721–743, April 2015.
(page 48)

[200] Hanif D. Sherali and Amine Alameddine.
A new reformulation-linearization technique for bilinear programming problems.
Journal of Global Optimization, 2(4):379–410, December 1992.
(page 76)

[201] Naum Z. Shor.
Cut-off method with space extension in convex programming problems.
Cybernetics, 13(1):94–96, January 1977.
(page 16)

[202] Maurice Sion.
On general minimax theorems.
Pacific Journal of Mathematics, 8(1):171–176, March 1958.
Publisher: Mathematical Sciences Publishers.
(pages 18, 22, and 111)

[203] Halil Mete Soner.
Optimal Control with State-Space Constraint I.
SIAM Journal on Control and Optimization, 24(3):552–561, May 1986.
Publisher: Society for Industrial and Applied Mathematics.
(page 47)

[204] Oliver Stein.
A semi-infinite approach to design centering.
Optimization with Multivalued Mappings: Theory, Applications, and Algorithms, pages 209–228, 2006.
Publisher: Springer.

(page 5)

[205] Oliver Stein and Paul Steuermann.
The adaptive convexification algorithm for semi-infinite programming with arbitrary index sets.
Mathematical Programming, 136(1):183–207, December 2012.
(pages 4 and 28)

[206] Bartolomeo Stellato, Goran Banjac, Paul Goulart, Alberto Bemporad, and Stephen Boyd.
OSQP: an operator splitting solver for quadratic programs.
Mathematical Programming Computation, 12(4):637–672, December 2020.
(page 116)

[207] Georg Still.
Discretization in semi-infinite programming: the rate of convergence.
Mathematical Programming, 91(1):53–69, October 2001.
(pages 4 and 17)

[208] Brian Stott.
Review of load-flow calculation methods.
Proceedings of the IEEE, 62(7):916–929, July 1974.
Conference Name: Proceedings of the IEEE.
(page 8)

[209] Matthew D Stuber and Paul I Barton.
Robust simulation and design using semi-infinite programs with implicit functions.
International Journal of Reliability and Safety, 5(3-4):378–397, 2011.
Publisher: Inderscience Publishers.
(page 123)

[210] Matthew D Stuber and Paul I Barton.
Semi-infinite optimization with implicit functions.
Industrial & Engineering Chemistry Research, 54(1):307–317, 2015.
Publisher: ACS Publications.
(pages 5 and 123)

[211] Jos F. Sturm.
Using SeDuMi 1.02, A Matlab toolbox for optimization over symmetric cones.
Optimization Methods and Software, 11(1-4):625–653, January 1999.
(pages 9 and 103)

[212] Kaarthik Sundar, Harsha Nagarajan, Sidhant Misra, Mowen Lu, Carleton Coffrin, and Russell Bent.
Optimization-Based Bound Tightening using a Strengthened QC-Relaxation of the Optimal Power Flow Problem.
arXiv:1809.04565 [cs, math], January 2019.
arXiv: 1809.04565.
(pages 76, 77, 97, 99, and 101)

[213] Matteo Tacchi, Bogdan Marinescu, Marian Anghel, Soumya Kundu, Sifeddine Benahmed, and Carmen Cardozo.
Power System Transient Stability Analysis Using Sum of Squares Programming.
In *2018 Power Systems Computation Conference (PSCC)*, pages 1–7, June 2018.
(page 10)

[214] Yoshihiro Tanaka, Masao Fukushima, and Toshihide Ibaraki.
A globally convergent SQP method for semi-infinite nonlinear optimization.
Journal of Computational and Applied Mathematics, 23(2):141–153, August 1988.
(page 4)

[215] Xiaojiao Tong, Liqun Qi, Soon-Yi Wu, and Felix F. Wu.
A smoothing SQP method for nonlinear programs with stability constraints arising from power systems.
Computational Optimization and Applications, 51(1):175–197, January 2012.
(page 10)

[216] Emmanuel Trélat.

Optimal Control and Applications to Aerospace: Some Results and Challenges.
Journal of Optimization Theory and Applications, 154(3):713–758, September 2012.
(pages 47 and 48)

[217] Angelos Tsoukalas, Panos Parpas, and Berç Rustem.
A smoothing algorithm for finite min–max–min problems.
Optimization Letters, 3(1):49–62, January 2009.
(page 122)

[218] Angelos Tsoukalas and Berç Rustem.
A feasible point adaptation of the Blankenship and Falk algorithm for semi-infinite programming.
Optimization Letters, 5(4):705–716, November 2011.
(pages 4 and 28)

[219] Hoang Tuy.
Convex Analysis and Global Optimization, volume 110 of *Springer Optimization and Its Applications*.
Springer International Publishing, Cham, 2016.
(page 19)

[220] Lieven Vandenberghe and Martin S Andersen.
Chordal Graphs and Semidefinite Optimization, volume 1 of *Foundations and Trends in Optimization*.
now Publishers Inc., Boston, MA, now publishers inc. edition, 2014.
(pages 105 and 106)

[221] Lieven Vandenberghe and Stephen Boyd.
Semidefinite programming.
SIAM review, 38(1):49–95, 1996.
Publisher: SIAM.
(page 32)

[222] Lieven Vandenberghe and Stephen Boyd.
Connections between Semi-Infinite and Semidefinite Programming.
In Rembert Reemtsen and Jan-J. Rückmann, editors, *Semi-Infinite Programming*, Nonconvex Optimization and Its Applications, pages 277–294. Springer US, Boston, MA, 1998.
(page 5)

[223] Andreas Venzke, Lejla Halilbasic, Uros Markovic, Gabriela Hug, and Spyros Chatzivasileiadis.
Convex Relaxations of Chance Constrained AC Optimal Power Flow.
IEEE Transactions on Power Systems, 33(3):2829–2841, May 2018.
arXiv: 1702.08372.
(pages 10 and 136)

[224] Robin Verschueren, Hans Joachim Ferreau, Alessandro Zanarini, Mehmet Mercangöz, and Moritz Diehl.
A stabilizing nonlinear model predictive control scheme for time-optimal point-to-point motions.
In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2525–2530, December 2017.
(page 48)

[225] Stefan Vigerske and Ambros Gleixner.
SCIP: global optimization of mixed-integer nonlinear programs in a branch-and-cut framework.
Optimization Methods and Software, 33(3):563–593, May 2018.
(page 9)

[226] Richard Vinter.
Convex Duality and Nonlinear Optimal Control.
SIAM Journal on Control and Optimization, 31(2):518–538, March 1993.
Publisher: Society for Industrial and Applied Mathematics.
(pages 48, 50, 51, and 52)

[227] Oskar von Stryk and Roland Bulirsch.
Direct and indirect methods for trajectory optimization.
Annals of Operations Research, 37(1):357–373, December 1992.
(pages 47 and 48)

[228] Francisco Guerra Vázquez and Jan-J. Rückmann.
Semi-Infinite Programming: Properties and Applications to Economics.
In Jacek Leskow, Lionello F. Punzo, and Martín Puchet Anyul, editors, *New Tools of Economic Dynamics*, Lecture Notes in Economics and Mathematical Systems, pages 373–393. Springer, Berlin, Heidelberg, 2005.
(page 5)

[229] Hayato Waki, Maho Nakata, and Masakazu Muramatsu.
Strange behaviors of interior-point methods for solving semidefinite programming problems in polynomial optimization.
Computational Optimization and Applications, 53(3):823–844, December 2012.
(page 103)

[230] Chong Wang, Bai Cui, Zhaoyu Wang, and Chenghong Gu.
SDP-Based Optimal Power Flow With Steady-State Voltage Stability Constraints.
IEEE Transactions on Smart Grid, 10(4):4637–4647, July 2019.
(page 10)

[231] Jie Wang, Victor Magron, Jean B. Lasserre, and Ngoc Hoang Anh Mai.
CS-TSSOS: Correlative and term sparsity for large-scale polynomial optimization.
arXiv:2005.02828 [cs, math], May 2020.
arXiv: 2005.02828.
(pages 102 and 103)

[232] Jie Wang, Victor Magron, and Jean-Bernard Lasserre.
Chordal-TSSOS: A Moment-SOS Hierarchy That Exploits Term Sparsity with Chordal Extension.
SIAM Journal on Optimization, 31(1):114–141, January 2021.
Publisher: Society for Industrial and Applied Mathematics.
(page 9)

[233] Jie Wang, Victor Magron, and Jean-Bernard Lasserre.
Certifying global optimality of AC-OPF solutions via sparse polynomial optimization.
Electric Power Systems Research, 213:108683, December 2022.
(page 6)

[234] Jie Wang, Victor Magron, Jean-Bernard Lasserre, and Ngoc Hoang Anh Mai.
CS-TSSOS: Correlative and Term Sparsity for Large-Scale Polynomial Optimization.
ACM Transactions on Mathematical Software, 48(4):42:1–42:26, December 2022.
(page 9)

[235] Anton Winterfeld.
Application of general semi-infinite programming to lapidary cutting problems.
European Journal of Operational Research, 191(3):838–854, December 2008.
(page 5)

[236] Felix Wu and Sadatoschi Kumagai.
Steady-State Security Regions of Power Systems.
IEEE Transactions on Circuits and Systems, 29(11):703–711, November 1982.
Conference Name: IEEE Transactions on Circuits and Systems.
(page 10)

[237] Xuan Wu, Antonio J. Conejo, and Nima Amjady.
Robust Security Constrained ACOPF via Conic Programming: Identifying the Worst Contingencies.
IEEE Transactions on Power Systems, 33(6):5884–5891, November 2018.
Conference Name: IEEE Transactions on Power Systems.
(page 10)

[238] Andreas Wächter and Lorenz T. Biegler.
On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming.
Mathematical Programming, 106(1):25–57, March 2006.
(pages 8, 76, 96, 118, and 138)

[239] İhsan Yanıkoğlu, Bram L. Gorissen, and Dick den Hertog.
A survey of adjustable robust optimization.
European Journal of Operational Research, 277(3):799–813, September 2019.
(pages 5 and 121)

[240] Yinyu Ye.
Complexity analysis of the analytic center cutting plane method that uses multiple cuts.
Mathematical Programming, 78(1):85–104, July 1996.
(page 17)

[241] Saïd Zabi, Isabelle Queinnec, Germain Garcia, and Michel Mazerolles.
Time-optimal control for the induction phase of anesthesia.
In *Proceedings of the 20th IFAC World Congress*, page 12708, July 2017.
(page 47)

[242] Stanislav Zakovic and Berç Rustem.
Semi-Infinite Programming and Applications to Minimax Problems.
Annals of Operations Research, 124:81–110, January 2003.
(page 5)

[243] Liping Zhang, Soon-Yi Wu, and Marco A. López.
A New Exchange Method for Convex Semi-Infinite Programming.
SIAM Journal on Optimization, 20(6):2959–2977, January 2010.
(pages 4 and 17)

[244] Yang Zheng, Giovanni Fantuzzi, Antonis Papachristodoulou, Paul Goulart, and Andrew Wynn.
Chordal decomposition in operator-splitting methods for sparse semidefinite programs.
Mathematical Programming, 180(1):489–532, March 2020.
(page 103)

[245] Ray Daniel Zimmerman, Carlos Edmundo Murillo-Sánchez, and Robert John Thomas.
MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education.
IEEE Transactions on power systems, 26(1):12–19, 2010.
Publisher: IEEE.
(page 176)

Appendices

A. Convergence proof of the adaptive discretization algorithm

We present the proofs of some convergence results for the adaptive discretization algorithm (Algorithm 1) presented in the Introduction, a standard algorithm for semi-infinite programming.

Lemma 0.6. *Consider a parameter $\delta \in [0,1)$, infinite sequences $(x^k)_{k\in\mathbb{N}} \subset \mathcal{X}$ and $(y^k)_{k\in\mathbb{N}} \subset \mathcal{Y}$, where $y^k = \hat{y}(x^k)$ is the output of the δ-oracle evaluated at point x^k. If, for any $k \in \mathbb{N}$, for any $\ell \in [\![0, k-1]\!]$, $G(x^k, y^\ell) \leq 0$, then, the feasibility error $\phi(x^k)^+$ vanishes: $\phi(x^k)^+ \to 0$.*

Proof. We notice that the sequence $\phi(x^k)^+$ is bounded, so it admits at least one accumulation value ℓ. We are going to prove that, necessarily, $\ell = 0$. We define $\psi : \mathbb{N} \to \mathbb{N}$ being an increasing function, such that $\phi(x^{\psi(k)})^+ \to \ell$. By compactness of $\mathcal{X}$ (resp. $\mathcal{Y}$), up to the extraction of a subsequence of $x^{\psi(k)}$ (resp. $y^{\psi(k)}$), we can assume that $x^{\psi(k)} \to x \in \mathcal{X}$ (resp. $y^{\psi(k)} \to y \in \mathcal{Y}$). By assumption, $G(x^{\psi(k)}, y^j) \leq 0$ for all $j \leq \psi(k) - 1$, in particular, $G(x^{\psi(k)}, y^{\psi(k-1)}) \leq 0$. We deduce that $G(x^{\psi(k)}, y^{\psi(k)}) \leq G(x^{\psi(k)}, y^{\psi(k)}) - G(x^{\psi(k)}, y^{\psi(k-1)})$, and therefore, since the positive part $t^+ = \max\{t, 0\}$ is nondecreasing,

$$\left(G(x^{\psi(k)}, y^{\psi(k)})\right)^+ \leq \left(G(x^{\psi(k)}, y^{\psi(k)}) - G(x^{\psi(k)}, y^{\psi(k-1)})\right)^+. \tag{36}$$

By definition of the δ-oracle, Eq. (4) yields $\phi(x^{\psi(k)}) - G(x^{\psi(k)}, y^{\psi(k)}) \leq \delta|\phi(x^{\psi(k)})|$. If $\phi(x^{\psi(k)}) \geq 0$, this means that $\phi(x^{\psi(k)}) - G(x^{\psi(k)}, y^{\psi(k)}) \leq \delta\, \phi(x^{\psi(k)})$, i.e., $\phi(x^{\psi(k)}) \leq \frac{1}{1-\delta} G(x^{\psi(k)}, y^{\psi(k)})$. To also cover the case $\phi(x^{\psi(k)}) < 0$, we write $\phi(x^{\psi(k)})^+ \leq \frac{1}{1-\delta} G(x^{\psi(k)}, y^{\psi(k)})^+$. As $\frac{1}{1-\delta} \geq 0$, we obtain from Eq. (36) that $\phi(x^{\psi(k)})^+ \leq \frac{1}{1-\delta}\left(G(x^{\psi(k)}, y^{\psi(k)}) - G(x^{\psi(k)}, y^{\psi(k-1)})\right)^+$. By continuity of the functions G, ϕ (Proposition 0.1), and $(\cdot)^+$, we deduce that $\ell = \phi(x)^+ \leq \frac{1}{1-\delta}\left(G(x,y) - G(x,y)\right)^+ = 0$, since $(x^{\psi(k)}, y^{\psi(k)})$ and $(x^{\psi(k)}, y^{\psi(k-1)})$ both converges to (x, y). Hence, $\phi(x^k)^+ \to 0$. $\square$

Theorem 0.3. *If $\epsilon \in \mathbb{R}_{++}$, Algorithm 1 stops after a finite number of iterations. On the contrary, if $\epsilon = 0$ and Algorithm 1 generates an infinite sequence of iterates $(x^k)_{k\in\mathbb{N}}$, then any limit value $x \in \mathbb{R}^m$ of $(x^k)_{k\in\mathbb{N}}$ is an optimal solution in (**SIP**).*

Proof. We consider fixed parameters $(\delta, \epsilon) \in [0,1) \times \mathbb{R}_+$, and we reason by contrapositive: we assume that Algorithm 1 generates infinite sequence $(x^k)_{k\in\mathbb{N}}$ and we show that $\epsilon = 0$.

The generated sequences $(x^k)_{k\in\mathbb{N}} \subset \mathcal{X}$ and $(y^k)_{k\in\mathbb{N}} \subset \mathcal{Y}$, satisfies $y^k = \hat{y}(x^k)$, and for any $\ell \in [\![0, k-1]\!]$, $G(x^k, y^\ell) \leq 0$. Hence, Lemma 0.1 yields that $\phi(x^k)^+ \to 0$. We take any limit value $x \in \mathbb{R}^m$ of $(x^k)_{k\in\mathbb{N}}$, and we define $(x^{\psi(k)})_{k\in\mathbb{N}}$ a subsequence converging to x. Since $\phi(x^{\psi(k)})^+$ is a subsequence of the vanishing sequence $\phi(x^k)^+$, and since ϕ is continuous (see Proposition 0.1), we deduce that $\phi(x)^+ = 0$. Since $\mathcal{X}$ is closed, $x \in \mathcal{X}$: in summary, x is feasible in (**SIP**). Since the stopping criterion is not met, $G(x^{\psi(k)}, y^{\psi(k)})^+ > \epsilon$ for all $k \in \mathbb{N}$. As $0 \leq G(x^{\psi(k)}, y^{\psi(k)})^+$, $\leq \phi(x^{\psi(k)})^+$, we deduce by continuity of ϕ that $G(x^{\psi(k)}, y^{\psi(k)})^+ \to 0$, and, therefore, $\epsilon = 0$. We conclude the proof by proving that $F(x) = \mathsf{val}(\mathbf{SIP})$. As x is feasible in (**SIP**), we have $F(x) \geq \mathsf{val}(\mathbf{SIP})$. However, as $x^{\psi(k)}$ is the optimal solution of a relaxation of (**SIP**), we know that $F(x^{\psi(k)}) \leq \mathsf{val}(\mathbf{SIP})$, and by continuity of F, $F(x) \leq \mathsf{val}(\mathbf{SIP})$. □

Theorem 0.4. *If Algorithm 1 terminates after K iterations, the iterate $x^K \in \mathcal{X}$, satisfies $G(x^K, y) \leq \frac{\epsilon}{1-\delta}$, for all $y \in \mathcal{Y}$, and has value $F(x^K) \leq \mathsf{val}(\mathbf{SIP})$.*

Proof. First, note that for any iterate x^k of the algorithm, $F(x^k) \leq \mathsf{val}(\mathbf{SIP})$ since it x^k is an optimal solution of a relaxation of the problem (**SIP**). If Algorithm 1 terminates after K iterations, this means that $\nu^{K+1} \leq \epsilon$, i.e., $G(x^K, y^K) \leq \epsilon$. If we are in the case $\phi(x^K) \geq 0$, then by property of the δ-oracle, we deduce that $\phi(x^K) - G(x^K, y^K) \leq \delta\, \phi(x^K)$, i.e., $\phi(x^K) \leq \frac{1}{1-\delta} G(x^K, y^K) \leq \frac{\epsilon}{1-\delta}$. If we are in the case $\phi(x^K) \leq 0$, the inequality $\phi(x^K) \leq \frac{\epsilon}{1-\delta}$ also holds. □

B. ACOPF: from the physics to the optimization problem

In this Appendix, we provide the minimal knowledge in electrical engineering to understand the constraints involved in the ACOPF problem.

Algebraic representation of the physical values in an AC circuit

We start with a brief introduction to the use of complex numbers to represent the physical quantities in an Alternating Current (AC) electrical circuit at steady-state. In an AC circuit, the physical values such as current, voltage and power values are sine signals that oscillate with the same pulse $\omega = 2\pi f \in \mathbb{R}$. In the European electrical grid, $f = 50$Hz.

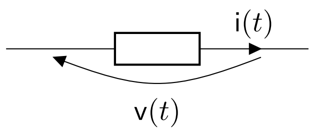


Figure 1: Dipole in an AC electrical circuit.

For a certain dipole in the circuit (see Figure 1), voltage (in V), current (in A) and power (in W) are functions of time:

$$\mathsf{v}(t) = V^{\max}\cos(\omega t + \theta_{\mathsf{v}}) \tag{37}$$

$$\mathsf{i}(t) = I^{\max}\cos(\omega t + \theta_{\mathsf{i}}) \tag{38}$$

$$\mathsf{s}(t) = \mathsf{v}(t)\mathsf{i}(t) = \frac{1}{2}V^{\max}I^{\max}(\cos(\theta_{\mathsf{v}} - \theta_{\mathsf{i}}) + \cos(2\omega t + \theta_{\mathsf{v}} + \theta_{\mathsf{i}})), \tag{39}$$

where $V^{\max}, I^{\max}$ are amplitude, and $\theta_{\mathsf{v}}, \theta_{\mathsf{i}}$ are phases. The equality for the power expression follows from basic trigonometric relations. Note that the average of the power $\mathsf{s}(t)$ over a period $2\pi/\omega = 1/f$ is

$$\int_0^{2\pi/\omega} \mathsf{s}(t)dt = \frac{1}{2}V^{\max}I^{\max}\cos(\theta_{\mathsf{v}} - \theta_{\mathsf{i}}). \tag{40}$$

The representation of the signals $\mathsf{v}(t), \mathsf{i}(t)$ by an amplitude and a phase, and the integral analysis of the power (see Eq. (40)) motivate the use of complex numbers to model these physical values.

$$V = \frac{V^{\max}}{\sqrt{2}}e^{\mathbf{i}\theta_{\mathsf{v}}} \tag{41}$$

$$I = \frac{I^{\max}}{\sqrt{2}}e^{\mathbf{i}\theta_{\mathsf{i}}} \tag{42}$$

$$S = VI^*. \tag{43}$$

With this definition, we can reformulate Eq. (40) as

$$\int_0^{2\pi/\omega} \mathsf{s}(t)dt = \mathsf{Re}(S). \tag{44}$$

In the power system literature, the real part of S is known as the *active power* (in W). This quantity is subject to some power balance equation (see Eq. 51). The imaginary part of S is an abstract quantity that is also subject to this power balance equation, and that also plays a crucial role in the representation of the system:

$$\mathsf{Im}(S) = \frac{1}{2}V^{\max}I^{\max}\sin(\theta_{\mathsf{v}} - \theta_{\mathsf{i}}). \tag{45}$$

In the power grid literature, the imaginary part of S is known as the *reactive power* (in VAR). We can understand it as a power that is consumed by the dipole, and then returned to the circuit over one period $2\pi/\omega$: in average the power consumption/production is null, but this oscillating behavior of the dipole (between power consumption and injection) is has a major impact on the circuit.

Description of the physical components of a power network

The grid A high voltage power network is made of buses interconnected by lines. It can be modeled as an oriented graph $\mathcal{N} = (\mathcal{B}, \mathcal{L})$. A line $\ell \in \mathcal{L}$ is described by a couple (b, a) such that $b \in \mathcal{B}$ is the "from" bus (denoted by f), $a \in \mathcal{B}$ is the "to" bus (denoted by t). The set $\mathcal{L}$ is such that $\mathcal{L} \cap \mathcal{L}^R = \emptyset$, where $\mathcal{L}^R$ is the set of couples (b, a) such that $(a, b) \in \mathcal{L}$.

Bus model A node $b \in \mathcal{B}$ of the graph $\mathcal{N}$ represents an electrical bus, i.e., a point of the network where are connected:

1. A subnetwork containing power generators and loads. This subnetwork is not detailed in the graph $\mathcal{N}$: it is represented as an exogenous flow I_b (see Figure 2) of current incoming at bus b. This exogenous flow results from
   - Some generators connected to this bus. We denote by $\mathcal{G}_b$ the set of generators located at bus $b \in \mathcal{B}$. It may be empty.
   - A load representing the power consumption in this subnetwork. This load may be, for instance, an industrial electricity consumer, or a local distribution network for residential consumers.

   The resulting flow I_b is the current injection at the bus b.
2. A shunt, of impedance $Z_b^s = 1/Y_b^s$, and with a current I_b^s flowing in it: this represents a loss of current to the ground at this bus.
3. Some lines $\ell = (b, a) \in \mathcal{L} \cup \mathcal{L}^R$ connecting the bus b to other buses, with current flows I_{ba}.

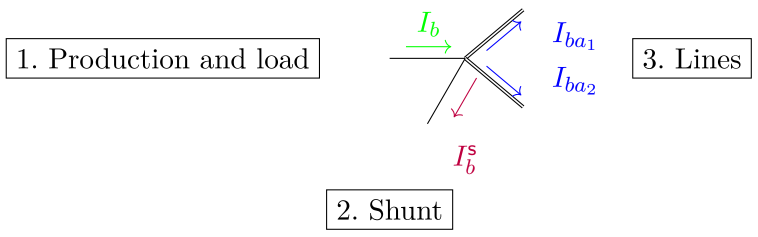


Figure 2: A bus $b \in \mathcal{B}$, with incident lines (b, a_1) and (b, a_2) (the impedances are not drawn).

The voltage of this electrical node $b \in \mathcal{B}$ with respect to the ground is V_b (see Figure 3).

Line model An electrical line $\ell = (b, a) \in \mathcal{L}$ is modeled as a quadripole, represented by the schema in Figure 3. The parameter $y_{ba} = r_{ba} + ix_{ba}$ is the line series impedance, and $\mathfrak{s}_{ba}$ is the line charging susceptance. The pair of vertical parallel coils on the left represents a transformer installed at b: the parameter $N_{ba} \in \mathbb{C}$, the "ratio" of the AC transformation, is not necessarily an integer; in general, it is a complex number. It is usually expressed in its polar representation

$N_{ba} = \tau_{ba} e^{\mathbf{i}\nu_{ba}}$. From these parameters, that are provided in the MATPOWER data [245] format for OPF instances, we derive the expression of the 2×2 admittance matrix for the line:

$$\mathbf{Y}_{ba} = \begin{pmatrix} Y_{ba}^{\mathsf{ff}} & Y_{ba}^{\mathsf{ft}} \\ Y_{ba}^{\mathsf{tf}} & Y_{ba}^{\mathsf{tt}} \end{pmatrix} = \begin{pmatrix} (\frac{1}{r_{ba}+\mathbf{i}x_{ba}} + \mathbf{i}\frac{\mathfrak{s}_{ba}}{2})/\tau_{ba}^2 & -\frac{1}{(r_{ba}+\mathbf{i}x_{ba})\tau_{ba}e^{-i\nu_{ba}}} \\ -\frac{1}{(r_{ba}+\mathbf{i}x_{ba})\tau_{ba}e^{i\nu_{ba}}} & \frac{1}{r_{ba}+\mathbf{i}x_{ba}} + \mathbf{i}\frac{\mathfrak{s}_{ba}}{2} \end{pmatrix}. \tag{46}$$

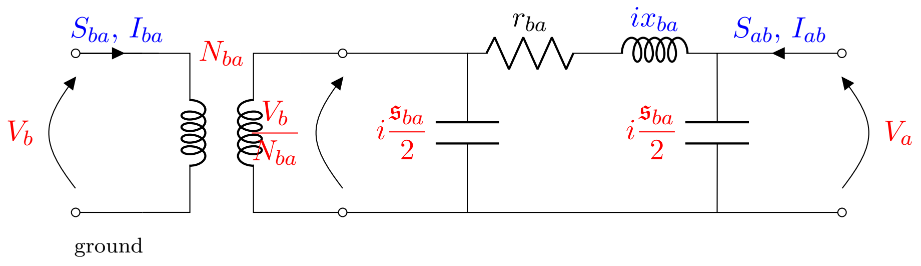


Figure 3: The π-model of a line $\ell = (b, a) \in \mathcal{L}$ (from [27], courtesy of L. Liberti)

Derivation of the power flow equations from Kirchoff's and Ohm's laws

We now state the two main principles that enable us to derive the power flow equation.

Kirchoff's Law This physical principle designates the conservation of current at each node $b \in \mathcal{B}$. Using the notation introduced in Figure 2, the Kirchoff's law is

$$I_b = I_b^s + \sum_{a\colon (b,a)\in\mathcal{L}\cup\mathcal{L}^R} I_{ba}. \tag{47}$$

Ohm's Law This physical principle links the current flowing through an electrical device, its voltage, and its admittance. For a given line $\ell = (b, a) \in \mathcal{L}$, the Ohm's law applied to this quadripole, represented by the matrix $\mathbf{Y}_{ba}$, reads as follows

$$\begin{pmatrix} I_{ba} \\ I_{ab} \end{pmatrix} = \mathbf{Y}_{ba} \begin{pmatrix} V_b \\ V_a \end{pmatrix} = \begin{pmatrix} Y_{ba}^{\mathsf{ff}} V_b + Y_{ba}^{\mathsf{ft}} V_a \\ Y_{ba}^{\mathsf{tf}} V_b + Y_{ba}^{\mathsf{tt}} V_a \end{pmatrix}. \tag{48}$$

Applying Ohm's law for the shunt at bus $b \in \mathcal{B}$ gives

$$I_b^s = Y_b^s V_b. \tag{49}$$

Power balance On the one hand, the net power injection at node b through the current flow I_b is

$$S_b = V_b I_b^* = V_b(I_b^s + \sum_{a\colon (b,a)\in\mathcal{L}\cup\mathcal{L}^R} I_{ba})^*, \tag{50}$$

the second inequality following from Kirchhoff's law (Eq. 47). On the other hand, the net power injection is also simply the difference between power generation $\sum_{g\in\mathcal{G}_b} S_g$ and the power consumption S_b^{d}. In summary, using Ohm's law we deduce the power flow equation:

$$\begin{aligned}\sum_{g\in\mathcal{G}_b} S_g - S_b^{\mathsf{d}} &= V_b(I_b^s + \sum_{a\colon (b,a)\in\mathcal{L}\cup\mathcal{L}^R} I_{ba})^* \\ &= (Y_b^s)^*|V_b|^2 + \sum_{a\colon (b,a)\in\mathcal{L}} ((Y_{ba}^{\mathsf{ff}})^*|V_b|^2 + (Y_{ba}^{\mathsf{ft}})^* V_b) + \sum_{a\colon (b,a)\in\mathcal{L}^R} ((Y_{ab}^{\mathsf{tt}})^*|V_b|^2 + (Y_{ab}^{\mathsf{tf}})^* V_b V_a^*).\end{aligned}$$

Operational limits The components of the electrical grid are designed to work within operational limits, that results in some inequality constraints in the ACOPF formulation:

- The operational limits of the generators are usually defined by a rectangle in the space $\mathbb{C}$

$$\underline{P}_g \leq \mathsf{Re}(S_g) \leq \overline{P}_g \tag{51}$$
$$\underline{Q}_g \leq \mathsf{Im}(S_g) \leq \overline{Q}_g, \tag{52}$$

  that can be summarized as $\underline{s}_g \leq S_g \leq \overline{s}_g$, with $\underline{s}_g = \underline{P}_g + \mathbf{i}\underline{Q}_g$, and $\overline{s}_g = \overline{P}_g + \mathbf{i}\overline{Q}_g$.
- The buses are designed for a certain range of voltage magnitude:

$$\underline{v}_b \leq |V_b| \leq \overline{v}_b. \tag{53}$$

- The electrical lines also have operational limits. Depending on the authors, these limits are either expressed in current magnitude ($|I_{ba}| \leq \bar{I}_{ba}$) or in apparent power ($|V_b I_{ba}| \leq \bar{S}_{ba}$).

Polar formulation

In Chapter 6, we use the polar formulation of the ACOPF problem, instead of the QCQP formulation: the complex variable $V_b \in \mathbb{C}$ (resp. $S_g \in \mathbb{C}$) is represented by $v_b e^{\mathbf{i}\theta_b}$ (resp. $P_g + \mathbf{i}Q_g$), with $v_b, \theta_b, P_g, Q_g \in \mathbb{R}$ being real variables. We define the following parameters, for any $b \in \mathcal{B}$:

$$\mathfrak{g}_b = \mathsf{Re}(Y_b^s) + \sum_{a\colon (b,a)\in\mathcal{L}} \mathsf{Re}(Y_{ba}^{\mathsf{ff}}) + \sum_{a\colon (b,a)\in\mathcal{L}^R} \mathsf{Re}(Y_{ab}^{\mathsf{tt}}) \tag{54}$$
$$\mathfrak{b}_b = \mathsf{Im}(Y_b^s) + \sum_{a\colon (b,a)\in\mathcal{L}} \mathsf{Im}(Y_{ba}^{\mathsf{ff}}) + \sum_{a\colon (b,a)\in\mathcal{L}^R} \mathsf{Im}(Y_{ab}^{\mathsf{tt}}). \tag{55}$$

For any $(b,a) \in \mathcal{L}$, we also define $\mathfrak{g}_{ba} = \mathsf{Re}(Y_{ba}^{\mathsf{ft}})$ and $\mathfrak{b}_{ba} = \mathsf{Im}(Y_{ba}^{\mathsf{ft}})$. For any $(b,a) \in \mathcal{L}^R$, we also define $\mathfrak{g}_{ba} = \mathsf{Re}(Y_{ab}^{\mathsf{tf}})$ and $\mathfrak{b}_{ba} = \mathsf{Im}(Y_{ab}^{\mathsf{tf}})$. With these parameters, the power flow equations in polar form reads, for all $b \in \mathcal{B}$,

$$\sum_{g\in\mathcal{G}_b} P_g - P_b^{\mathsf{d}} = \mathfrak{g}_b v_b^2 + \sum_{a:(b,a)\in\mathcal{L}\cup\mathcal{L}^R} v_b v_a(\mathfrak{g}_{ba}\cos(\theta_b - \theta_a) + \mathfrak{b}_{ba}\sin(\theta_b - \theta_a)) \tag{56}$$
$$\sum_{g\in\mathcal{G}_b} Q_g - Q_b^{\mathsf{d}} = -\mathfrak{b}_b v_b^2 + \sum_{a:(b,a)\in\mathcal{L}\cup\mathcal{L}^R} v_b v_a(\mathfrak{g}_{ba}\sin(\theta_b - \theta_a) - \mathfrak{b}_{ba}\cos(\theta_b - \theta_a)). \tag{57}$$

C. ACOPF: complements on the conic relaxation

Strict dominance of Constraints (4.1)-(4.6) with respect to Constraints (†)-(‡)

In Section 4.2.2, Proposition 4.2 states that Constraints (4.1)-(4.6) dominate Constraints (†)-(‡). The advantage of Constraints (4.1)-(4.6) is to enforce a coupling between the convex envelopes of the quadruplets $(\mathsf{Re}(W_{ba}), \mathsf{Im}(W_{ba}), W_{bb}, W_{aa})$ involving the same index b. In this Appendix, we present an illustrative example of two quadruplets $(\mathsf{Re}(W_{ba}), \mathsf{Im}(W_{ba}), W_{bb}, W_{aa})$ and $(\mathsf{Re}(W_{bc}), \mathsf{Im}(W_{bc}), W_{bb}, W_{cc})$ satisfying Constraints (†)-(‡) introduced in [56], but for which there are no vectors L and R such that Constraints (4.1)-(4.6) are simultaneously satisfied for (b, a) and (b, c). This illustrates the interest of using the trigonometric cuts (4.6) with a variable radius R_{ba}, whereas previous works, to our knowledge, only use cuts associated to extreme values of R_{ba}.
We consider any $(b, a, c) \in \mathcal{B}^3$ with the following realistic data:

- Voltage Magnitude Bounds: $\underline{v}_b = \underline{v}_a = \underline{v}_c = 0.9$ and $\overline{v}_b = \overline{v}_a = \overline{v}_c = 1.1$,
- Phase Angle Difference Bounds: $\overline{\theta}_{ba} = \overline{\theta}_{bc} = -\underline{\theta}_{ba} = -\underline{\theta}_{bc} = \arccos(0.99) \simeq 8.11°$.

As a consequence, we have $v_b^\sigma = v_a^\sigma = v_c^\sigma = 2$ and $\omega_{ba} = \omega_{bc} = 0$ and $\delta_{ba} = \delta_{bc} = \arccos(0.99)$. We consider the quadruplets

$$(\mathsf{Re}(W_{ba}), \mathsf{Im}(W_{ba}), W_{bb}, W_{aa}) = (1.1, 0, 1, 1.21) \tag{58}$$

$$(\mathsf{Re}(W_{bc}), \mathsf{Im}(W_{bc}), W_{bb}, W_{cc}) = (1.085, 0, 1, 1.21), \tag{59}$$

noticing that W_{bb} has indeed the same value in both quadruplets. These quadruplets both satisfy Equations (†)-(‡).

- First quadruplet: It satisfies (†), since

$$\begin{aligned} &v_b^\sigma v_a^\sigma \left(\cos(\omega_{ba})\mathsf{Re}(W_{ba}) + \sin(\omega_{ba})\mathsf{Im}(W_{ba})\right) - \overline{v}_a \cos(\delta_{ba}) v_a^\sigma W_{bb} - \overline{v}_b \cos(\delta_{ba}) v_b^\sigma W_{aa} \\ &\quad = 2 \times 2 \times 1 \times 1.1 + 0 - 1.1 \times 0.99 \times 2 \times 1 - 1.1 \times 0.99 \times 2 \times 1.21 = -0.41338, \end{aligned}$$

  which is greater than

$$\overline{v}_b \overline{v}_a \cos(\delta_{ba})(\underline{v}_b \underline{v}_a - \overline{v}_b \overline{v}_a) = 1.1 \times 1.1 \times 0.99 \times (0.9 \times 0.9 - 1.1 \times 1.1) = -0.47916.$$

  It satisfies (‡), since

$$\begin{aligned} &v_b^\sigma v_a^\sigma \left(\cos(\omega_{ba})\mathsf{Re}(W_{ba}) + \sin(\omega_{ba})\mathsf{Im}(W_{ba})\right) - \underline{v}_a \cos(\delta_{ba}) v_a^\sigma W_{bb} - \underline{v}_b \cos(\delta_{ba}) v_b^\sigma W_{aa} \\ &\quad = 2 \times 2 \times 1 \times 1.1 + 0 - 0.9 \times 0.99 \times 2 \times 1 - 0.9 \times 0.99 \times 2 \times 1.21 = 0.46178, \end{aligned}$$

  is greater than

$$-\underline{v}_b \underline{v}_a \cos(\delta_{ba})(\underline{v}_b \underline{v}_a - \overline{v}_b \overline{v}_a) = -0.9 \times 0.9 \times 0.99 \times (0.9 \times 0.9 - 1.1 \times 1.1) = 0.32076.$$

- Second quadruplet: It satisfies (†), since

$$\begin{aligned}&v_b^\sigma v_c^\sigma\left(\cos(\omega_{bc})\mathsf{Re}(W_{bc})+\sin(\omega_{bc})\mathsf{Im}(W_{bc})\right)-\overline{v}_c\cos(\delta_{bc})v_c^\sigma W_{bb}-\overline{v}_b\cos(\delta_{bc})v_b^\sigma W_{cc}\\ &=2\times 2\times 1\times 1.085+0-1.1\times 0.99\times 2\times 1-1.1\times 0.99\times 2\times 1.21=-0.47338,\end{aligned}$$

  which is greater than

$$\overline{v}_b\overline{v}_c\cos(\delta_{bc})(\underline{v}_b\underline{v}_c-\overline{v}_b\overline{v}_c)=1.1\times 1.1\times 0.99\times(0.9\times 0.9-1.1\times 1.1)=-0.47916.$$

  It satisfies (‡), since

$$\begin{aligned}&v_b^\sigma v_c^\sigma\left(\cos(\omega_{bc})\mathsf{Re}(W_{bc})+\sin(\omega_{bc})\mathsf{Im}(W_{bc})\right)-\underline{v}_c\cos(\delta_{bc})v_c^\sigma W_{bb}-\underline{v}_b\cos(\delta_{bc})v_b^\sigma W_{cc}\\ &=2\times 2\times 1\times 1.085+0-0.9\times 0.99\times 2\times 1-0.9\times 0.99\times 2\times 1.21=0.40178\end{aligned}$$

  is greater than

$$-\underline{v}_b\underline{v}_c\cos(\delta_{bc})(\underline{v}_b\underline{v}_c-\overline{v}_b\overline{v}_c)=-0.9\times 0.9\times 0.99:\times(0.9\times 0.9-1.1\times 1.1)=0.32076.$$

We assume now that there exists $L_b\in[\underline{v}_b,\overline{v}_b], L_a\in[\underline{v}_a,\overline{v}_a], L_c\in[\underline{v}_c,\overline{v}_c], R_{ba}\in[\underline{v}_b\underline{v}_a,\overline{v}_b\overline{v}_a]$, $R_{bc}\in[\underline{v}_b\underline{v}_c,\overline{v}_b\overline{v}_c]$ such that Constraints (4.1)-(4.6) are satisfied. Since $W_{aa}=\overline{v}_a^2$, we deduce from Constraint (4.4) that $\overline{v}_a^2+\underline{v}_a\overline{v}_a=R_{aa}+\underline{v}_a\overline{v}_a\leq(\underline{v}_a+\overline{v}_a)L_a$, i.e., that $(\underline{v}_a+\overline{v}_a)\overline{v}_a\leq(\underline{v}_a+\overline{v}_a)L_a$, and, thus, $\overline{v}_a\leq L_a$ since $(\underline{v}_a+\overline{v}_a)>0$. As $\overline{v}_a\geq L_a$ by definition of L_a, we observe that $L_a=\overline{v}_a$. We use then $R_{ba}\leq\overline{v}_aL_b+\underline{v}_bL_a-\overline{v}_a\underline{v}_b$ from Constraint (4.2) to deduce that $R_{ba}\leq\overline{v}_aL_b$. Constraint (4.5) gives $|W_{ba}|\leq R_{ba}$, meaning that $L_b\geq\frac{|W_{ba}|}{\overline{v}_a}=1$. As $L_b^2\leq W_{bb}=1$ according to Constraint (4.3), $L_b=1$. As we did for a, we deduce from $W_{aa}=\overline{v}_a^2$ and Constraint (4.4) that $L_c=\overline{v}_c$. We use $R_{bc}\leq\overline{v}_cL_b+\underline{v}_bL_c-\overline{v}_c\underline{v}_b$ from Constraint (4.2) to state that $R_{bc}\leq\overline{v}_cL_b$, and we use $R_{bc}\geq\overline{v}_bL_c+\overline{v}_cL_b-\overline{v}_b\overline{v}_c$ to state that $R_{bc}\geq\overline{v}_cL_b$. Hence, $R_{bc}=\overline{v}_cL_b=1.1$. As Constraint (4.6) imposes $\cos(\omega_{bc})\mathsf{Re}(W_{bc})+\sin(\omega_{bc})\mathsf{Im}(W_{bc})\geq R_{bc}\cos(\delta_{bc})$, we deduce that $(\omega_{bc})\mathsf{Re}(W_{bc})+\sin(\omega_{bc})\mathsf{Im}(W_{bc})\geq 1.089$. This is contradictory with the fact that $(\omega_{bc})\mathsf{Re}(W_{bc})+\sin(\omega_{bc})\mathsf{Im}(W_{bc})=\mathsf{Re}(W_{bc})=1.085$. As a conclusion, there does not exist $L_b\in[\underline{v}_b,\overline{v}_b], L_a\in[\underline{v}_a,\overline{v}_a], L_c\in[\underline{v}_c,\overline{v}_c], R_{ba}\in[\underline{v}_b\underline{v}_a,\overline{v}_b\overline{v}_a], R_{bc}\in[\underline{v}_b\underline{v}_c,\overline{v}_b\overline{v}_c]$ such that Constraints (4.1)-(4.6) are satisfied for the pairs (b,a) and (b,c).

Nonlinear but convex objective and constraints in the relaxation (ACOPF$_C$)

The decision vector in (ACOPF$_C$) is $x=(\mathsf{Re}(S),\mathsf{Im}(S),\mathsf{Re}(W),\mathsf{Im}(W),L,R)\in\mathbb{R}^\Xi$. First, we denote by $\mathcal{P}\subset\mathbb{R}^\Xi$ the polytope defined by the following box constraints:

- For all $g\in\mathcal{G}$, $\mathsf{Re}(S_g)\in[\mathsf{Re}(\underline{s}_g),\mathsf{Re}(\overline{s}_g)]$ and $\mathsf{Im}(S_g)\in[\mathsf{Im}(\underline{s}_g),\mathsf{Im}(\overline{s}_g)]$,
- For all $(b,a)\in\mathcal{E}$, $\mathsf{Re}(W_{ba})\in[0,\overline{v}_b\overline{v}_a]$, $\mathsf{Im}(W_{ba})\in[-\overline{v}_b\overline{v}_a,\overline{v}_b\overline{v}_a]$ and $R_{ba}\in[\underline{v}_b\underline{v}_a,\overline{v}_b\overline{v}_a]$,
- For all $b\in\mathcal{B}$, $L_b\in[\underline{v}_b,\overline{v}_b]$.

Now, we review the nonlinear terms in the objective and in the constraints of relaxation ($\mathbf{ACOPF}_C$), as functions of x. We show that all these functions have the form $\phi(x) = \max_{y\in\mathcal{Y}} y^\top \pi(x)$ for all $x \in \mathcal{P}$, with a given affine application $\pi : \mathbb{R}^\Xi \mapsto \mathbb{R}^p$ and a compact and convex set $\mathcal{Y} \subset \mathbb{R}^p$.

- **The objective function** is $\sum\limits_{g\in\mathcal{G}} \left(c_{0g} + c_{1g}\,\mathsf{Re}(S_g) + \sum\limits_{g\in\mathcal{G}_2} c_{2g}\,\mathsf{Re}(S_g)^2 \right)$, where $\mathcal{G}_2$ is the set of generators $g \in \mathcal{G}$ such that $c_{2g} > 0$. This function reads $\max_{y\in\mathcal{Y}} y^\top \pi(x)$ for all $x \in \mathcal{P}$ with
  - The compact and convex set $\mathcal{Y} = \{1\} \times \prod_{g\in\mathcal{G}_2} \{(z_1, -z_2) : z_1^2 \le z_2, z_1 \in [\mathsf{Re}(\underline{s}_g), \mathsf{Re}(\overline{s}_g)]\}$,
  - The affine application $\pi(x) = \left(\sum\limits_{g\in\mathcal{G}} (c_{0g} + c_{1g}\mathsf{Re}(S_g)), (2c_{2g}\mathsf{Re}(S_g), c_{2g})_{g\in\mathcal{G}_2} \right)$.
- **Thermal limits for lines** yield constraints with the form $|y_1^* W_{bb} + y_2^* W_{ba}| - \overline{S}_{ba} \le 0$, with $(y_1, y_2) = (Y_{ba}^{\mathsf{ff}}, Y_{ba}^{\mathsf{ft}})$ if $(b, a) \in \mathcal{L}$ or $(y_1, y_2) = (Y_{ab}^{\mathsf{tt}}, Y_{ab}^{\mathsf{tf}})$ if $(b, a) \in \mathcal{L}^R$. Introducing $(r_1, h_1, r_2, h_2) = (\mathsf{Re}(y_1), \mathsf{Im}(y_1), \mathsf{Re}(y_2), \mathsf{Im}(y_2))$, this constraint is

$$\sqrt{(r_1\mathsf{Re}(W_{bb}) + r_2\mathsf{Re}(W_{ba}) + h_2\mathsf{Im}(W_{ba}))^2 + (-h_1\mathsf{Re}(W_{bb}) + r_2\mathsf{Im}(W_{ba}) - h_2\mathsf{Re}(W_{ba}))^2} - \overline{S}_{ba} \le 0.$$

  This is $\max_{y\in\mathcal{Y}} y^\top \pi(x) \le 0$ with
  - The compact and convex set $\mathcal{Y} = \{(-1, z_1, z_2) \in \mathbb{R}^3 \ : \ z_1^2 + z_2^2 \le 1\}$,
  - The affine application $\pi(x) = (\overline{S}_{ba}, r_1\mathsf{Re}(W_{bb}) + r_2\mathsf{Re}(W_{ba}) + h_2\mathsf{Im}(W_{ba}), -h_1\mathsf{Re}(W_{bb}) + r_2\mathsf{Im}(W_{ba}) - h_2\mathsf{Re}(W_{ba}))$.
- **Constraint** (4.3), i.e., $L_b^2 - R_{bb} \le 0$ for any $b \in \mathcal{B}$, has the form $\max_{y\in\mathcal{Y}} y^\top \pi(x) \le 0$, with
  - The compact and convex set $\mathcal{Y} = \{(-1, z_1, -z_2) \in \mathbb{R}^2 \ : \ z_1^2 \le z_2 \in [\underline{v}_b, \overline{v}_b]\}$,
  - The affine application $\pi(x) = (R_{bb}, 2L_b, 1)$.
- **Constraint** (4.5), i.e., $|W_{ba}| - R_{ba} \le 0$ for any $(b, a) \in \mathcal{E}$, has the form $\max_{y\in\mathcal{Y}} y^\top \pi(x) \le 0$ with
  - The compact and convex set $\mathcal{Y} = \{(-1, z_1, z_2) \in \mathbb{R}^3 \ : \ z_1^2 + z_2^2 \le 1\}$,
  - The affine application $\pi(x) = (R_{ba}, \mathsf{Re}(W_{ba}), \mathsf{Im}(W_{ba}))$.
- The relaxation ($\mathbf{ACOPF}_C$) includes several **SDP constraints** $\mathcal{A}(x) \succeq 0$, where $\mathcal{A}$ is a linear matrix operator. Such a constraint amounts to $\max_{y\in\mathcal{Y}} y^\top \pi(x) \le 0$ with
  - The compact and convex set $\mathcal{Y} = \{M \in \mathbb{H}_p \ : \ (\mathsf{Tr}(M) = 1) \wedge (M \succeq 0)\}$,
  - The linear application $\pi(x) = -\mathcal{A}(x)$,

  and seeing $p \times p$ Hermitian matrices as real vectors of length $2p^2$.

D. Other technical lemmatas

Lemma 0.7. *We consider a compact set $S \subset \mathbb{R}^p$, and the family of compact sets $S_\delta = \{z \in \mathbb{R}^p : d(z, S) \leq \delta\}$ for any $\delta \geq 0$ and a continuous function $\psi \in C(\mathbb{R}^p)$. Then, the function $\Psi(\delta) = \min_{z \in S_\delta} \psi(z)$ is continuous at* 0.

Proof. First of all, we notice that the function $\delta \mapsto \min_{z \in S_\delta} \psi(z)$ is well-defined, since ψ is continuous and S_δ is compact. As $S_{\delta_1} \subset S_{\delta_2}$ for any $\delta_1 \leq \delta_2$, the function Ψ is non-increasing, which proves that the following limit exists:

$$\lim_{\delta \to 0^+} \Psi(\delta) = \Psi(0^+) \leq \Psi(0). \tag{60}$$

We take a positive sequence $(\delta_k) \in \mathbb{R}_{++}^{\mathbb{N}}$ such that $\delta_k \to 0$. Hence, $\Psi(\delta_k) \to \Psi(0^+)$ by definition of the right-limit. For any $k \in \mathbb{N}$, we define $z_k \in S_{\delta_k}$ such that $\psi(z_k) = \Psi(\delta_k)$. The sequence (δ_k) being bounded, we can introduce an upper bound $\bar{\delta}$. Hence, any element of the sequence (z_n) belongs to the compact set $S_{\bar{\delta}}$, and up to the extraction of a subsequence, converges to a point z being such that $\psi(z) = \Psi(0^+)$ by continuity of ψ and uniqueness of the limit. As $d(z_k, S)$, the distance between z_k and the compact set S, is bounded above by δ_k and is nonnegative, it converges to 0. By continuity of the distance, we know that $d(z, S) = 0$ and, thus, $\Psi(0^+) = \psi(z) \geq \Psi(0)$. Together with Eq. (60), this yields $\Psi(0^+) = \Psi(0)$. □

Lemma 0.8. *Let $Q \in C^1(\mathbb{R}^p)$ be a continuously differentiable function, with a locally Lipschitz gradient. Let $Z \subset \mathbb{R}^p$ be a compact set. Then, there exists a constant $A \in \mathbb{R}_{++}$, and a sequence of polynomials $(w_d(x))_{d \in \mathbb{N}^*}$ such that for all $d \in \mathbb{N}^*$, $w_d \in \mathbb{R}_d[x_1, \dots, x_p]$ and*

$$\sup_{x \in Z} |w_d(x) - Q(x)| \leq \frac{A}{d} \tag{61}$$

$$\sup_{x \in Z} \|\nabla w_d(x) - \nabla Q(x)\| \leq \frac{A}{d}. \tag{62}$$

We underline that the constant A implicitly depends on p, Q and Z, but not on the polynomial $w_d(x)$, nor on its degree d.

Proof. We introduce a constant $R \in \mathbb{R}_{++}$ such that $Z \subset B(0, R)$, and the function $\tilde{\omega} = \omega * \mathbf{1}_{B(0,R+1)}$, where ω is the mollifier introduced in the proof of Theorem 3.3. We notice that $\tilde{\omega} \in C^\infty(\mathbb{R}^p)$ is supported on $\tilde{Z} = B(0, R+2)$ and constant equal to 1 over $B(0, R)$. We define $\tilde{Q}(x) = Q(x)\tilde{\omega}(x)$, and we notice that (i) $\tilde{Q}$ is supported on the compact set $\tilde{Z}$, which contains Z, (ii) for all $x \in Z$, $\tilde{Q}(x) = Q(x)$ and $\nabla \tilde{Q}(x) = \nabla Q(x)$. Applying [10, Th. 1] to the compactly supported function $\tilde{Q}$, we know that there exists a constant C such that for any $d \geq 1$, there exists a polynomial $w_d(x)$ of degree at most d such that

$$\sup_{x\in Z} |w_d(x) - \tilde{Q}(x)| \leq \frac{C}{d}\kappa(\frac{1}{d}) \leq C\kappa(\frac{1}{d}), \tag{63}$$

$$\sup_{x\in Z} |\partial_i(w_d - \tilde{Q})(x)| \leq C\kappa(\frac{1}{d}) \tag{64}$$

where $\kappa(\delta) = \sup_{1\leq i\leq p} \sup_{\substack{(x,x')\in\mathbb{R}^p\times\mathbb{R}^p\\ |x-x'|\leq\delta}} |\partial_i\tilde{Q}(x) - \partial_i\tilde{Q}(x')|$. We define $\tilde{Z} = \{x \in \mathbb{R}^p \colon d(x, Z) \leq 1\}$. Since $\partial\tilde{Q}$ is uniformly null outside $\tilde{Z}$, and assuming that $\delta \leq 1$, we notice that

$$\kappa(\delta) = \sup_{1\leq i\leq p} \sup_{\substack{x,x'\in\tilde{Z}\times\mathbb{R}^p\\ |x-x'|\leq\delta}} |\partial_i\tilde{Q}(x) - \partial_i\tilde{Q}(x')| = \sup_{1\leq i\leq p} \sup_{\substack{x,x'\in\tilde{Z}\times\tilde{Z}\\ |x-x'|\leq\delta}} |\partial_i\tilde{Q}(x) - \partial_i\tilde{Q}(x')|.$$

We note that $\partial_i\tilde{Q}(x) = \tilde{\omega}(x)\partial_iQ(x)+Q(x)\partial_i\tilde{\omega}(x)$, and therefore, $|\partial_i\tilde{Q}(x)-\partial_i\tilde{Q}(x')| = |\tilde{\omega}(x)(\partial_iQ(x)-\partial_iQ(x')) + \partial_iQ(x')(\tilde{\omega}(x) - \tilde{\omega}(x')) + Q(x)(\partial_i\tilde{\omega}(x) - \partial_i\tilde{\omega}(x')) + \partial_i\tilde{\omega}(x')(Q(x) - Q(x'))|$. We use, then, the triangle inequality and the facts that (i) $\tilde{\omega}$ is C^∞, therefore bounded, Lipschitz continuous, and with a Lipschitz-continuous gradient over $\tilde{Z}$ and (ii) Q is continuously differentiable, therefore bounded, and Lipschitz continuous over $\tilde{Z}$; by assumption it has a Lipschitz continuous gradient over the compact set $\tilde{Z}$. We deduce that $\partial_i\tilde{Q}$ is Lipschitz continuous over $\tilde{Z}$: there exists $L_i > 0$ such that $\sup_{\substack{x,x'\in\tilde{Z}\times\tilde{Z}\\ |x-x'|\leq\delta}} |\partial_i\tilde{Q}(x) - \partial_i\tilde{Q}(x')| \leq L_i\delta$, for all $\delta \in [0, 1]$. Defining $L = \max_i L_i$, we deduce $\kappa(\delta) \leq L\delta$. Then Eq. (63) reads $\sup_{x\in Z} |w_d(x) - \tilde{Q}(x)| \leq \frac{CL}{d}$, and Eq. (64) reads $\sup_{x\in Z} |\partial_i(w_d - \tilde{Q})(x)| \leq \frac{CL}{d}$ for all $i \in \{1, \dots, p\}$. We also deduce that $\sup_{x\in Z}\|\nabla w_d(x) - \nabla\tilde{Q}(x)\|\leq \frac{CLp}{d}$. Defining $A = CLp$, and noticing that for all $x \in Z$, $\tilde{Q}(x) = Q(x)$ and $\nabla\tilde{Q}(x) = \nabla Q(x)$, one obtains the claimed statement. □

Lemma 0.9. *Under Assumption 3.1 and Assumption 3.2, we consider an admissible trajectory* $(x(\cdot), u(\cdot))$ *over* $[0, t_1]$ *of the minimal time control problem* (3.2)-(3.5) *starting from* $(0, x_0)$*. Then, for almost all* $t \in [0, t_1]$*,* $f(t, x(t), u(t)) \in T_X(x(t))$*.*

Proof. We reduce to a time-invariant controlled system: we define, for any $z = (t, x) \in \mathbb{R}^{n+1}$ and $u \in \mathbb{R}^m$, $\tilde{f}(z, u) = \begin{pmatrix} 1 \\ f(z, u) \end{pmatrix}$, and the constant set-valued map $\tilde{U}(z) = U$. The control system $(\tilde{f}, \tilde{U})$ is a Marchaud control system [7, Def. 6.1.3], since: (i) the graph of $\tilde{U}$ is closed (ii) $\tilde{f}$ is continuous (iii) the velocity subsets $\{\tilde{f}(z, u) \colon u \in \tilde{U}(z)\}$ are convex due to Assumption 3.1, and (iv) the function f has a linear growth since it is Lipschitz continuous, and the set-valued map are bounded, thus also has a linear growth. We define the set $\mathcal{C} = \mathbb{R}_+ \times X$ and notice that the control $u(\cdot)$ regulates a trajectory $z(t) = \begin{pmatrix} t \\ x(t) \end{pmatrix}$ that remains in $\mathcal{C}$, therefore according to [7, Th. 6.1.4], for all most all $t \in [0, t_1]$, $u(t) \in \{u \in \tilde{U}(z(t)) \colon \tilde{f}(z(t), u(t)) \in T_{\mathcal{C}}(z(t))\}$. We notice that $T_{\mathcal{C}}(z(t)) \subset \mathbb{R} \times T_X(x(t))$, implying that $f(z(t), u(t)) \in T_X(x(t))$. □

Lemma 0.10. *For any locally Lipschitz continuous function $F\colon \mathbb{R}^p \to \mathbb{R}$, and for any Lipschitz continuous curve $z\colon [0,T] \to \mathbb{R}^p$, the function $t \mapsto F(z(t))$ is differentiable a.e. and satisfies, for almost all $t \in [0,T]$,*

$$\min_{g\in\partial^c F(z(t))} g^\top \dot z(t) \le \frac{d}{dt}(F(z(t))) \le \max_{g\in\partial^c F(z(t))} g^\top \dot z(t). \tag{65}$$

The particular functions F for which these three quantities are equal are called path-differentiable in [33].

Proof. First, we notice that the functions $t \mapsto z(t)$ and $t \mapsto F(z(t))$ are Lipschitz continuous, therefore differentiable a.e. on $[0,T]$ due to the Rademacher theorem [79]. Hence, for almost all $t \in [0,T]$, both $z(t)$ and $F(z(t))$ are differentiable at t. We consider such a t, and we show that (65) holds for this particular t, which we prove the Lemma. Since z is differentiable at t, $r(h) = z(t+h) - z(t) - h\dot z(t)$ is in $o_{h\to 0}(h)$. Since $s \mapsto F(z(s))$ is differentiable at t, the following holds

$$\frac{d}{dt}(F(z(t))) = \lim_{h\to 0, h>0} \frac{F(z(t+h)) - F(z(t))}{h} \tag{66}$$
$$= \lim_{h\to 0, h>0} \frac{F(z(t) + h\dot z(t) + r(h)) - F(z(t))}{h} \tag{67}$$

Since $r(h) = o_{h\to 0}(h)$ and F is locally Lipschitz, we know that $\lim\limits_{h\to 0, h>0} \frac{F(z(t)) - F(z(t)+r(h))}{h} = 0$. Summing this with Eq. (67), we deduce that

$$\frac{d}{dt}(F(z(t))) = \lim_{h\to 0, h>0} \frac{F(z(t) + r(h) + h\dot z(t)) - F(z(t) + r(h))}{h} \tag{68}$$
$$\le \limsup_{\substack{z'\to z(t)\\ h\to 0, h>0}} \frac{F(z' + h\dot z(t)) - F(z')}{h} = F^\circ(z(t), \dot z(t)), \tag{69}$$

where $F^\circ(z; v) = \limsup\limits_{\substack{z'\to z\\ h\to 0, h>0}} \frac{F(z'+hv) - F(z')}{h}$ is the $F^\circ(z; h)$ Clarke's directional derivative at $z \in \mathbb{R}^p$ in the direction $v \in \mathbb{R}^p$. The inequality follows from the fact that $z(t) + r(h) \to z(t)$. By property of the Clarke subdifferential [51], we also know that $F^\circ(z; v) = \max_{g\in\partial^c F(z)} g^\top v$. Hence, in particular,

$$\frac{d}{dt}(F(z(t))) \le \max_{g\in\partial^c F(z(t))} g^\top \dot z(t). \tag{70}$$

The reasoning that proved Eq. (70) is also applicable to $-F$, that is also locally Lipschitz, and such that $s \mapsto (-F)(s)$ is differentiable at t. Therefore,

$$\frac{d}{dt}(-F(z(t))) \le \max_{g\in\partial^c (-F)(z(t))} g^\top \dot z(t). \tag{71}$$

As $\partial^c(-F)(z(t)) = -\partial^c F(z(t))$ by property of the Clarke subdifferential, we deduce that

$$-\frac{d}{dt}(F(z(t))) \leq \max_{g \in \partial^c F(z(t))} -g^\top \dot{z}(t) = -\min_{g \in \partial^c F(z(t))} g^\top \dot{z}(t), \tag{72}$$

and therefore $\frac{d}{dt}(F(z(t))) \geq \min_{g \in \partial^c F(z(t))} g^\top \dot{z}(t)$. $\square$