Variational-analysis look at combinatorial optimization, and other selected topics in optimization

Jérôme Malick

Université Grenoble Alpes

Habilitation à diriger des recherches

Spécialité : Informatique et Mathématiques appliquées

présentée le 26 janvier 2017 par

Jérôme Malick

Variational-analysis look at combinatorial optimization and other selected topics in optimization

---

Rapporteurs

Alexandre d'Aspremont	Département d'informatique, Ecole Normale Supérieure, Paris
Yurii Nesterov	Ecole Polytechnique de Louvain-la-Neuve, Belgique
Michael Overton	Courant Institute of Mathematical Sciences, New York University, USA

Jury

Jérôme Bolte	Toulouse School of Economics
Gérard Cornuéjols	Carnegie Mellon University, Pittsburgh, USA
Jean Lasserre	LAAS, Toulouse
Nabil Layaida	Inria, Grenoble

Contents

Chapter 1

Introduction

This document gives a global view on my research on mathematical optimization. The present chapter is the entry door for this document: it highlights specific aspects that constitute the originality of my research, gives an overview of my contributions, and summarizes four of my main contributions. I have chosen these contributions to give a flavour on my various interests, subjects, and developments. Each of these four contributions constitutes the topic of a chapter of this document. These contributions have opened new research perspectives, detailed in a final chapter opening to future research.

I have written this introductive chapter to be accessible to non-experts. Its presentation is increasingly technical, starting from a general discussion on my philosophy of work and going step by step to mathematical developments. I have adopted a concise style, with few references. The interested (or frustrated) reader should refer to the others chapters. The bibliographical entries of my publications are denoted by [Mal-*] and given at the end of Chapter 2; the other references are gathered in a common bibliography at the end of this document.

For simplicity, I use the first person (saying “I”, or “my”) in this chapter, but all the research results presented in this document have been obtained by team work with colleagues and students. Indeed, I have had the chance to work with great researchers and great persons (acknowledged at the end of Chapter 2) from who I have learnt a lot. Thus, except in the first two chapters of this document giving a global view and making an exhaustive summary of my activities, the “we” pronoun is mainly used throughout this document.

1.1 Overview of my research activities

My research work has multiple facets but mainly fits in mathematical optimization, which is a branch of applied mathematics[1] dealing with variational problems of the form "minimizing a quantity subject to constraints". Mathematical optimization is currently going through a period of growth and revitalization supported by an explosion of applications in data science and image/signal processing. In parallel, we have been observing a constant penetration of optimization and operation research tools in engineering in a broad sense (industry, services, and management). In this active field of research, I have got various contributions, presented briefly in section 1.2, with a special emphasis on some of them in section 1.3. I stay here at a higher level to highlight the originality of my work.

[1]Sometimes, one presents probability as "the mathematics of randomness" and statistics as "the mathematics of information". Then I would say that optimization is "the mathematics of decision-making". J.-B. Hiriart-Urruty, one the eniment researchers of optimization in France, says that it is "the mathematics of doing better" (with limited resources) [100].

1.1.1 Scientific positioning and originality

The guiding motivation of my work is to promote mathematics in action. All my research projects and scientific interests have been driven, closely or indirectly, from applications in other scientific domains or in industry and services. I have derived this "taste for usefulness" to the three leitmotivs of my research:

1. motivate theoretical developments by applications,
2. unify and clarify results,
3. make tools and results accessible to non-specialists.

This philosophy has shaped the aspects that constitute my scientific originality. I am going to point them out throughout the rest of this chapter and this document. I would like to emphasize here four essential aspects of my research and contributions in mathematical optimization:

- I have contributions of different natures, covering all the range from theoretical analysis to algorithmic developments and real-life applications. I give two illustrations of this aspect in section 1.1.3.
- My contributions concern many facets of optimization: convex, nonsmooth, conic, semidefinite, polynomial, combinatorial, Riemannian, and stochastic. They also concern various applications in mechanics, finance, energy, and computer vision. This is detailed later in section 1.2.
- I have drawn connections between different subjects, e.g. between Riemannian optimization and nonsmooth optimization ([Mal-26], [Mal-24]), between variational analysis and projection algorithms (see chapter 6) or between continuous optimization and combinatorial optimization (as developed in section 1.1.2).
- I always pay a special attention and effort to explain the main ideas behind the technical developments: in my articles, the proofs are given in a precise but intelligible way, several examples are given to illustrate ideas, and various figures help support geometric intuition. I try to illustrate this aspect throughout this document.

Below, I develop further the first and third aspects; the second is detailed in section 1.2, and the fourth, hopefully, throughout the document.

1.1.2 From continuous to combinatorial optimization

As explained above, a particularity of my work is to build connections between different aspects of optimization or different communities. I illustrate this originality with my research on the intersection of continuous and combinatorial optimization, more precisely on the use of advanced convex optimization and variational analysis on combinatorial optimization problems[2]. Let us start with a general positioning.

Historically, continuous and combinatorial optimization have followed two distinct trajectories. The study of combinatorial optimization has been intertwined with that of theoretical computer science: the foundations of computational complexity and algorithm design blossomed around the study

[2] this explains the title of this document !

of discrete optimization problems. In contrast, continuous optimization has been grounded in the mathematical theory of convex analysis and geometry. The overlap with scientific computing has led continuous optimization to become a basic tool in many areas of science that adopt continuous models to describe and understand natural phenomena. In the last decade, the interface between discrete and continuous optimization has become increasingly active, partly stimulated by complex industrial problems (Big Problems) and the proliferation of massive datasets (Big Data). The emergence of a non-linear mixed-integer optimization community clearly positioned between the continuous and combinatorial optimization is an illustration of the increasing interactions between the two topics.

An significant part of my research lies at this interface. To give bird-eye view on it, let me further specify the context. Roughly speaking, combinatorial optimization problems can be formulated as minimizing a "simple" objective function $f\colon \mathbb{R}^n \to \mathbb{R}$ (e.g. linear or quadratic) over a "complicated" constraint set $S \subset \mathbb{R}^n$ (with discrete or nonconvex aspects).

$$\min f(x) \text{ such that } x \in S \tag{1.1}$$

Most of the approaches to solve or approximate such problems are based on the two following foundational techniques of combinatorial optimization:

- bounding (or relaxing), i.e. enlarge S to a “simpler” set P;
- cutting, i.e. construct a hyperplane separating S and a point of P to tighten the relaxation.

These two paradigms turn out to be intrinsically linked to convex optimization. First, the notion of separating hyperplanes is also at the heart of convex analysis, as a frequent proof technique and in the analysis of support functions, providing theoretical advances going beyond standard linear techniques. Second, convex duality provides an automatic way (often called Lagrangian relaxation) to obtain lower bounds on the optimal value of (1.1) in the case when S can be expressed as $S = X \cap \{x : c(x) = 0\}$, where X is “easy” in terms of minimizing over it, while the constraint $c\colon \mathbb{R}^n \to \mathbb{R}^m$ is "complicating". By construction, dual problems are convex, so that dual resolution and primal-dual relationships have a natural appeal for convex analysis and optimization. When applied to binary quadratic problems for example, duality leads to optimization problems over the convex cone of positive semidefinite matrices (semidefinite optimization). I further develop and illustrate these ideas in a popularization article "convexity and combinatorics" in the French operations research community newsletter [130].

I have decided to highlight in this document two main recent accomplishments in this context:

- algorithmic research on bounding: Chapter 3 presents a bounding procedure for binary quadratic problems intersecting dynamically standard cuts and original semidefinite relaxations.
- theoretical research on cutting: Chapter 5 presents the introduction and analysis of a theory of “cut generating functions” in integer optimization, based on convex analysis.

1.1.3 From mathematics to real-life problems

Another particularity of my research, mentioned previously, is to cover all the range from theoretical analysis, to algorithmic developments and real-life applications. To illustrate this, I briefly present two examples of contributions on the two extreme sides of this range: a "pure" mathematical contribution and an application to an industrial problem.

Pure mathematics: spectral manifolds The spectral sets are subsets of the symmetric matrices space $\mathbb{S}_n$ entirely defined by their eigenvalues, that is, that are invariant under the action of the orthogonal group by conjugation. They can be equivalently defined as inverse images of subsets of $\mathbb{R}^n$ by the spectral function λ mapping a symmetric matrix X to its ordered eigenvalues

$$\lambda^{-1}(M) := \{X \in \mathbb{S}_n : \lambda(X) \in M\}, \quad \text{for some } M \subset \mathbb{R}^n.$$

For example, if M is the Euclidean unit ball of $\mathbb{R}^n$, then $\lambda^{-1}(M)$ is the Euclidean unit ball of $\mathbb{S}_n$ too.

Nice research has investigated which properties on a set M "lift" to the corresponding spectral set $\lambda^{-1}(M)$. Simple examples show that everything can happen but it turns out that invariance properties of M under permutations often correct bad behaviors. Indeed, if the set $M \subset \mathbb{R}^n$ is invariant by permutations, then M is closed, convex, or prox-regular respectively if and only if so is $\lambda^{-1}(M)$ ([122],[Mal-22]). Note that such a lifting property also holds for algebraicity: if M is a permutation-invariant algebraic manifold, then $\lambda^{-1}(M)$ is an algebraic manifold of $\mathbb{S}_n$ (see [Mal-6]).

A natural question is then the following: is it possible also to lift smooth manifolds? Hristo Sendov and Aris Daniilidis and I answered positively to the question in general by introducing the notion of *local* invariance by permutation (or local symmetry in the terminology of [Mal-6]).

Theorem 1.1.1. *Consider a closed set $\mathcal{M}$ locally symmetric around $\bar{x} = \lambda(\bar{X})$, which means that there exists a neighborhood $\mathcal{N}$ of $\bar{x}$ in M invariant by all the permutations fixing a vector $x \in \mathcal{N}$. If $\mathcal{M}$ is a C^k submanifold of $\mathbb{R}^n$ of dimension d around $\bar{x}$, then $\lambda^{-1}(\mathcal{M})$ is a C^k submanifold of $\mathbb{S}_n$ around $\bar{X}$ with dimension*

$$\dim \lambda^{-1}(M) \;=\; d \;+ \sum_{1 \le i < j \le m^*} |I_i^*||I_j^*|, \tag{1.2}$$

where $\{I_1^*, \ldots, I_{m^*}^*\}$ is the so-called characteristic partition of M defined in [Mal-6, Section 2.3].

The unit Euclidean ball of $\mathbb{S}_n$ or the manifold of matrices of rank k provide basic illustrations of this theorem. We established its proof in a long research report [58] that we split into two articles: [Mal-9] about characteristic properties of locally symmetric submanifolds and [Mal-6] about the spectral manifolds themselves. Applications can be found in matrix manifold optimization (in the construction of retractions for Riemannian optimization algorithms [Mal-12]) and in image processing applications (in alternating projections with spectral manifolds, see e.g., the discussion of the appendix of [Mal-21]).

Let me finish this discussion by mentioning that this work has opened several perspectives. First, the tools we developed to prove the above theorem could be used in principle to prove the transfer of all local properties of sets M or functions f to associated spectral sets or functions. In particular, we could revisit our previous paper [Mal-22] and refine its main result to get the transfer of the notion of "prox-regularity" *for a fixed vector* in the subdifferential. We have not built further on this line of research. Second, the above theorem is in turn the key step in proving the lifting of central notions for many active-set-type optimization algorithms (namely the partial-smoothness and the properties of "identifiability" [57]). The authors of [57] were also able to simplify the proof of our theorem by using a nice result from [148] linking the smoothness of the manifold M and the one of the squared distance function $x \mapsto \min_{z \in M} \|x - z\|_2^2$.

Real-life problem: Optimization of electricity generation at EDF Day-to-day optimization of electricity generation is an important industrial problem that faces EDF, the French electricity board. It consists in finding a minimal cost production schedule for the next day that satisfies the operational constraints and that meets customers' demand. This can be formalized as a large-scale optimization

problems having complex discrete subproblems: the problem is to minimize generation costs over (separable) operational constraints and (coupling) demand constraints. EDF uses a Lagrangian duality approach that leads to a nonsmooth convex problem that drives the solution process [74].

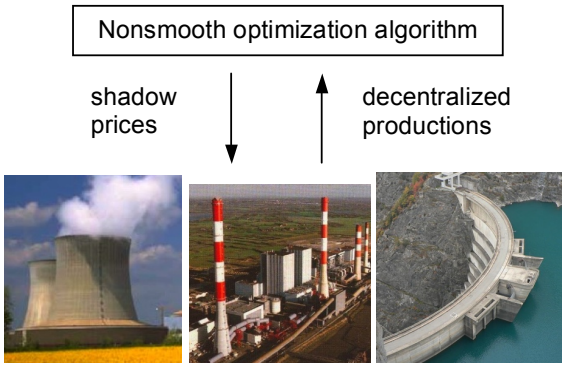


Figure 1.1: Decomposition of large-scale unit-commitment problems by duality

This problem, having a huge industrial impact, is a great playground for my research in convex analysis and optimization. I inherited this application from my PhD advisor Claude Lemaréchal; I have contributed on several aspects presented below. I am proud to help in saving money (power generating costs) and emission of pollution (CO2 and nuclear waste) everyday in France.

- Marginal price recovery (with Claude Lemaréchal). Dual variables computed by the nonsmooth optimization algorithm are interpreted as marginal prices of generation schedules. A standard convex optimization result states that dual optimal solution gives the derivative of the optimal cost with respect to a variable demand [101]. However the EDF problem is not convex and is modified by constraints penalizations. Questions then arise about the meaning of the computed prices and their links with intrinsic prices. We produced an EDF internal research report on this question in 2010. We also prepared with our EDF colleagues two popularization articles on this successful application of optimization [93], [94].

- Stabilisation of prices (with Sofia Zaourar). Several combinatorial sub-problems of the electricity generation optimization problem cannot be solved exactly within strong time constraints. The EDF nonsmooth optimization algorithm can efficiently handle these inexact computations (that was the subject of a preceding contract with EDF). The issue comes from the inherent sensibility of solutions computed by the inexact algorithm. We proposed a simple and cheap way to get rid of this computational noise by adding a penalization term to control price variations. As a result on real-life EDF problems, the new prices show 80% less of variation while staying within the error margin of the inexact solution, see more in [Mal-10].

- Stochastic unit-commitment (with Wim van Ackooij). The next big step consisted of adding uncertainty on the demand to capture in the model the impact of weather conditions on consumption and generation by renewable sources (wind, sun). In the article [Mal-5], we develop an innovative double decomposition (by units and by scenarios) of the resulting stochastic optimization problem that scales up to the large-scale heterogeneous instances of EDF (by using the same numerical sub-routines as the deterministic version in use at EDF). Our current work on this application is about investigating other stochastic models and associated algorithms, including a robust optimization approach able to deal with complex forms of uncertainty [132].

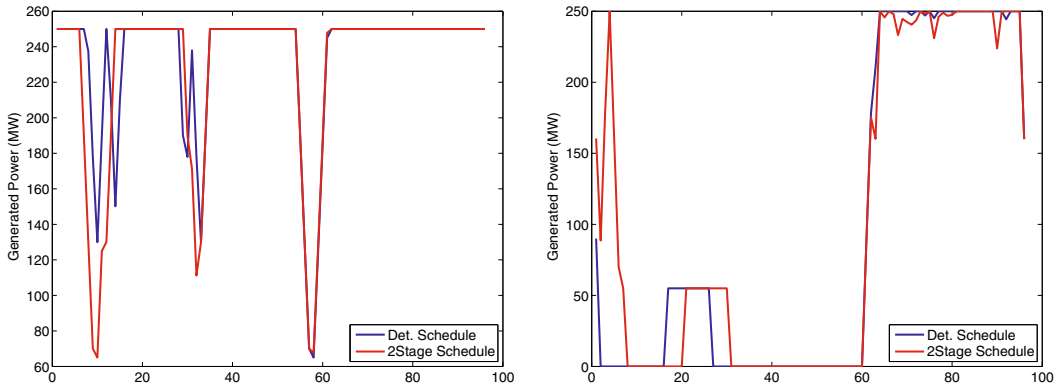


Figure 1.2: Electricity generation under uncertainty: illustration of the transfer of generation from an inflexible unit (left) to a flexible one (right) between the deterministic schedule (blue) and the stochastic one (red)

I illustrate the latter point with a numerical experiment extracted from [Mal-5] on a stochastic unit-commitment problem with real-life EDF data. The overall (two-stage) stochastic problem (with 50 scenarios) has 1,200,000 continuous variables, 700,000 binary variables, and more than 20,000,000 constraints – completely out of reach of existing (mixed-integer linear) solvers. Leveraging on its two-stage structure, our double decomposition algorithm allows us to solve this giant problem in reasonable time (i.e. with computing times comparable to those observed when solving deterministic unit-commitment problems). It is interesting to observe the impact of uncertainty on the computed schedules sent to units: some generation is transferred from inflexible but cheap units to expensive but flexible units. See more information in [Mal-5], and other applications of optimization in the field of energy in section 4.4.

1.2 Overview of my contributions

My research focus is on mathematical, algorithmic and computational aspects of optimization, driven by problems from others domains or real-life applications. This section gives a bird-eye view of my scientific contributions, grouped in three topics. It also contains some elements to evaluate the impact of these contributions, and their rupture with previous state-of-the-art.

I have tried to make this section accessible to non-experts, i.e. computer scientists or mathematicians, who have basic knowledge on operation research or optimization. For sake of brevity, I have given very few references to the state-of-the-art; more complete discussions can be found in corresponding articles.

1.2.1 Applied convex optimization

Convexity is a key property in optimization, and more generally in applied mathematics and engineering, having taken on a prominent role akin to the one previously played by linearity[3] . Convex optimization, and in particular semidefinite optimization, is also used for studying and solving non-

[3]In an article in SIAM Review [157], R. Tyrrell Rockafellar, one of the most important researchers of the domain, wrote "...in fact, the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity."

convex optimization problems. I present here my contributions in this area, with an emphasis on conic and semidefinite optimization (which were my first themes of research).

Convex analysis for cut-generating functions The article [Mal-8] introduces a formal theory of cut-generating functions, generalizing the famous Gomory's cuts (whose use within mixed-integer linear solvers reduced the computing times by a factor of 1000). Our theoretical analysis is strongly supported by convex analysis through standard techniques and less-known properties. This research was published in the top journal in theoretical operation research [Mal-8] and on of the most selective conference in combinatorial optimization [Mal-31]. It is the topic of chapter 5 of this document. Moreover, section 7.1 highlights some of the follow-up articles inspired by this work.

Conic optimization algorithms The main field of research in my early carrer was focused on a new generation of projection-type algorithms for semidefinite optimization. After six years of research, the chapter [Mal-34] of book *Handbook of semidefinite, conic, and polynomial optimization* gives a synthetic overview of my four articles on this theme ([Mal-28], [Mal-27], [Mal-19] and [Mal-16]) together with some results that they have inspired to other researchers. In this paper, the results are put in perspective, generalized to conic optimization, and illustrated by new applications in polynomial optimization. Projection-type algorithms and applications quickly became an active and competitive research theme: since my very first article [Mal-28] in 2004, more than 150 articles have been published on this topic.

Applications of semidefinite optimization I have applied semidefinite optimization algorithms to covariance matrices calibration. Such calibration problems appear in robust models for stock selection [Mal-28, Section 5], in stress testing of financial models [149], or in the construction of global risk models from local ones – this last application was explicitly called "one of the biggest current challenges in risk modeling" by [4]. The dual algorithm of [Mal-28] proposes an efficient response to this challenge. I also considered applications in control, through the test of positivity of polynoms [Mal-16], which is a generic relaxation of many linear and non-linear control problems [97]. My paper with Didier Henrion [Mal-16] received the Charles Broyden prize of the best article of the year 2011 in the journal *Optimization Methods and Software*.

Semidefinite optimization for combinatorial optimization My main contribution on this subject is the design of semidefinite optimization methods for exact resolution of combinatorial optimization problems, through a sequence of papers: from the original idea [Mal-23], a basic deployment on a special problem [Mal-13], a theoretical study of the new semidefinite relaxations in general [Mal-11], the complete deployment of the approach on the simplest but the most competitive case (max-cut) [Mal-7], a brief come-back to the special problem [Mal-4], and finally the development and distribution of the generic software [Mal-1]. The whole chapter 3 is devoted to this contribution.

1.2.2 Variational analysis: theory and algorithms

Variational analysis is a branch of applied mathematics that extends the methods arising from the classical calculus of variations and convex analysis [158]. I have got interested in several questions in nonsmooth analysis, in particular about spectral properties of matrix functions.

Structured nonsmooth optimization Despite powerful tools of variational analysis developed since the 70s (see [156] in particular), the nonsmoothness is still a major difficulty of optimization.

However a nonsmooth function often admits an underlying regularity that I exploited to establish theoretical properties or to design faster optimization algorithms [Mal-26], [Mal-24]. These two articles reveal surprising connections between nonsmooth optimization algorithms and Riemannian optimization algorithms, which were insightful for both sides of the connections.

Matrix manifold optimization Riemannian optimization has been emerging over the last decade driven by various applications in robotics, vision, or medical imaging in particular. The article [Mal-12] enlarges the field of potential applications by introducing practical tools for a key step of these methods (retraction). We developped a theoretical framework to analyze known retractions and to generate new ones by projection-like operations. The underlying ideas were inspired from recent developments in nonsmooth optimization through the bridge revealed by [Mal-26].

Analysis and geometry of spectral functions and spectral sets The so-called spectral functions and sets (defined only through properties of eigenvalues of matrices) frequently appear in engineering sciences, in particular in signal and image processing. In application articles, properties of spectral objects are often proved, by hand, for special cases. My work concerns with the automatic transfert of variational properties to spectral functions [Mal-22] and the automatic transfert of smoothness to spectral sets [Mal-9], [Mal-6] (presented in section 1.1.2). I also wrote a pedagogical article [Mal-15] on variational properties of positive semidefinite matrices and sets of positive semidefinite matrices.

Nonconvex projection algorithms Alternating projection methods are very popular in imaging sciences, but their theoretical study was restrained to convex settings. My contribution on this topic was an original variational approach to prove local convergence of these methods, by-passing the convexity-based arguments. Thus we got the first convergence analysis in a non-convex framework: first in the smooth case [Mal-21] and then in a general case [Mal-20]. This topic is developed in Chapter 6. Perspectives and related work are also discussed in section 7.1.

1.2.3 Optimization at work

My research has been driven, closely or indirectly, from applications in other scientific domains or in industry and services. In particular, I have had the following four projects in direct contact with applications. On each of these, my contributions stemming from my work and my "optimization" point of view have led to original advances and have opened new perspectives.

Applications in machine learning and computer vision Challenging optimization learning problems have appeared with Big Data area. These learning problems are of huge scale but are also strongly structured, often as a form of a composite objective function with "data-fidelity" term and a "low-complexity" regularizer [54]. For example, the regularization by trace-norm (which is the sum of the singular values of the involved matrix) enforces a low rank to optimal solutions, which is a natural target in multi-class learning or collaborative filtering. In this case, existing (proximal) methods [24] use at each iteration a singular value decomposition, too expensive in a large-scale setting.

We proposed in 2010 a first algorithm [Mal-33] that scales up to these large-scale learning problems with trace-norm regularization, by constructing low-rank iterates using only largest singular value sub-computations. The links with Frank-Wolfe algorithm were later revealed by [92]. In the meanwhile, we pushed further an application in computer vision: we improved each numerical block of the algorithm to apply it to the image categorization challenge "imagenet". The algorithm scaled up to tackle large scale problems in the three learning dimensions (number of examples, feature size,

number of categories) and we obtained experimental results significantly better than competitive approaches. This work was accepted for a presentation in the most prestigious conference in computer vision [Mal-32].

Applications in contact mechanics The simulation of mechanical dynamic systems where objects are in contact and friction requires to solve equations with nonsmooth components. For instance, when using the Coulomb's friction law, these equations involve convex cones (second-order cones, or ice-cream cones), that are treated by polyhedral approximations in the existing numerical software. We proposed in [Mal-18] an approach to solve such computational mechanical problems by isolating its part of convexity (to treat it apart by conic optimization) and reformulating the overall problem as a fixed-point problem of a nonsmooth application. The new approach provides, first, a proof of existence of a solution to the problem under natural assumptions [Mal-17] and, second, a numerical method to solve the problem [Mal-18]. Numerical experiments on computer graphics problems showed that this approach is surprisingly robust. It was embedded in the software Siconos, developed in my former research team (Bipop at Inria Grenoble) and transferred to two French companies (EDF and Schneider).

Applications in finance Practical implementation of financial mathematics reveals numerical uncertainties that are not captured in the financial models (missing or perverted data, approximation errors amplified by non-linearities...). Efficient and robust decision-making tools thus call for financial engineering developments based on optimization. Between 2007 and 2009, I helped the Grenoble company RaisePartner in their software developments on these issues. In particular, my algorithm [Mal-28] gives an efficient way to calibrate covariance matrices in the financial models, which is an important problem in computational finance (as underlined in [4]).

Applications in electricity generation planning The electricity generation in France is managed by the EDF's nonsmooth optimization algorithm [74]. I have improved this algorithm on several aspects, including a noise reducing technique [Mal-10] and a stochastic extension [Mal-5]. This important industrial application is further discussed in section 1.1.3.

1.3 Overview of this document

This goal of this document is to give a global view of my work. The present chapter briefly presents my research activities; the next chapter completes it with my other activities of researcher (supervision, teaching, projects,...). The rest of the document develops four of my main contributions. Finally, Chapter 7 concludes this work and draws some perspectives for the future.

A few criteria were used for picking a subset of contributions further detailed in the next chapters. Basically, I chose to focus on a balanced set of self-contained results that are quite representative of my research approach, while illustrating different facets and particularities of my contributions. Chapters 3 and 4 are more algorithmic, about semidefinite algorithms of combinatorial optimization and about nonsmooth optimization algorithms for energy optimization. Chapters 5 and 6 are more mathematical, about convex analysis in discrete maths, and about variational analysis of non-convex projections. I briefly review these contributions here in this section; interested readers should refer to the corresponding chapters for a complete presentation to results, related work, and references. Finally, I present follow-up research and perspectives on these topics in section 7.1.

1.3.1 Semidefinite optimization for binary quadratic problems

Chapter 3 is the achievement of an ambitious research started in 2007 with Frédéric Roupin (Paris XIII) on semidefinite optimization for quadratic problems with binary variables. These are combinatorial optimization problems modeling standard graph theory problems (e.g. max-cut, max-clique,...) and applications (e.g. in medecine [104], physics [125], computer vision [107]).

Since the 90s (see e.g. [84] and [141]), semidefinite optimization methods became an important technique in combinatorial optimization, sustained by the generalization of interior points methods from linear to convex optimization [140]. Despite the nice theory, the practical applications to solve combinatorial problems have reached only partial success, due to the burden of heavy linear algebra sub-computations in solvers. Until recent years, semidefinite programming has been mostly viewed as a nice theoretical technique not really useful in practice in the context of combinatorial optimization.

Our research has showed that semidefinite optimization can indeed be used with great practical efficiency in this context. Our contributions have been to introduce new semidefinite bounds and to embed them efficiently within an exact resolution scheme. More specifically, the sequence of contributions and the six associated papers have been the following.

- We first wanted to validate the whole approach on a specific problem. We chose the k-cluster problem which consists in finding in a graph a subgraph of k nodes with the heaviest weight. The first computational experiments (on relaxations [Mal-35] and on exact resolutions [Mal-13]) were promising, as a basic implementation was already competitive with the state-of-the-art [29].

- The article [Mal-11] presented our methodology in general: we introduced and studied a new family of semidefinite bounds for general quadratic optimization problems with binary variables. This family of bounds shows an interesting property: we can manage the ratio of tightness over computing time with a real parameter. When the parameter vanishes, the tightness tends to the one of a standard semidefinite relaxation.

- We then focused on the most competitive special case, the max-cut problem, for which it existed many results, test-problems, and an efficient semidefinite-based software [151]. This problem is the "simplest" binary quadratic problem, in the sense that it has the simple formulation as an unconstrained quadratic problem, which is a favorable situation for standard semidefinite relaxations and interior-point algorithms. With Nathan Krislock (post-doc with us in 2010-2012), we came up in [Mal-7] with an efficient implementation of our bounds having an adaptative updating rule of the tightness parameter and an automatic selection of strengthening inequalities. Embedded within a branch-and-bound platform, it provides a complete software package that outperforms the previous methods.

- We finally generalized each component of the previous max-cut algorithm to general binary quadratic problems. We came up with the first (and so far unique) software based on semidefinite optimization to solve these problems to optimality. This software, called *BiqCrunch*, is presented in Chapter 3, along with its basic ideas, its mathematics foundations, and tips to efficiently use it. It has been successfully tested on a variety of combinatorial optimization problems, such as k-cluster [Mal-4] and max-independent-set; see many results online at

  `http://lipn.univ-paris13.fr/BiqCrunch/`

  The code is publicly available online; a web interface and conversion tools are also provided.

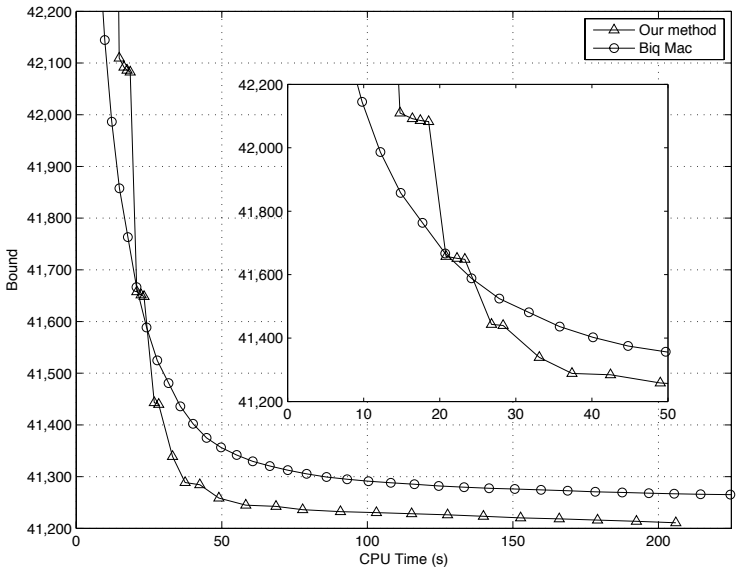


Figure 1.3: Illustration, on a max-cut problem (Beasley bqp250.6), of the semidefinite bounds produced by Biq Mac [151] vs ours. We see that our algorithm gets quickly in the zone of "tight" bounds that could help to prune parts of branch-and bound tree. Drops in the curve correspond to updates of the tightness parameter, see chapter 3.

1.3.2 Nonsmooth optimization with uncontrolled inexact information

Chapter 4 considers convex nonsmooth optimization problems where additional information with uncontrolled accuracy is readily available. It is often the case when the objective function is itself the output of an optimization solver, as for large-scale energy optimization problems tackled by decomposition. This chapter studies how to incorporate the uncontrolled linearizations into bundle algorithms in view of generating better iterates and possibly accelerating the methods. The article [Mal-2] presented in this chapter was the first to explicitly mention and study uncontrolled inexact information in a context of bundle methods.

In this chapter, a convergence analysis of two (proximal and level) bundle algorithms using uncontrolled linearizations is developed. The technical difficulty was to handle excessive inexactness within level bundle methods (in contrast with prox-bundle methods, where increasing the stepsize tackles excessive noise [113]). We proposed an original *implicit* noise attenuation rule, that gives asymptotic convergence of the algorithm (without further boundedness assumption).

Numerical illustrations show that the bundle methods incorporating uncontrolled linearizations can indeed speed up resolution for two stochastic optimization problems coming from energy optimization (two-stage linear problems and chance-constrained problems in reservoir management). As expected, the level bundle algorithm, fully exploiting the additional uncontrolled information in its lower bound, works particularly well on these problems. Let us finally note that the numerical experiments compare level and proximal bundle algorithms, whereas such comparisons are (surprisingly) rare in the literature on nonsmooth optimization.

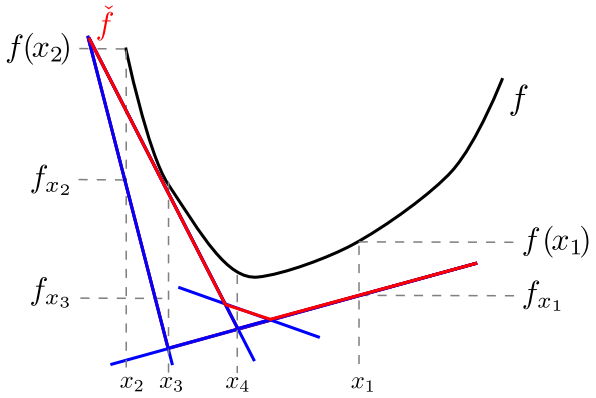


Figure 1.4: Illustration of uncontrolled inexact linearizations. The inexact sub-linearizations at x_1, x_2 and x_4 still improve the cutting-plane model (in red). We have convergence when the model incorporates enough controlled linearizations (as the exact linearization at x_3)

1.3.3 Discrete analysis of cut-generating functions

Chapter 5 introduces the concept of "cut-generating functions" and develops a formal theory for them, largely based on convex analysis. Cuts are a fundamental tool in mixed-integer programming, and they have had an enormous practical impact on solvers. The theory of cut-generating functions allows the development of a library of cuts that are independent of the structure of the problem on hand and are cheap to generate. Gomory cuts, the most used cuts in commercial solvers, fit this framework and indeed our theory is inspired by them.

More precisely, in this chapter, we consider the separation problem for sets X that are pre-images of a given set S by a linear mapping $X = \left\{x \in \mathbb{R}^n_+ : \ Rx \in S\right\}$. Classical examples occur in integer programming, as well as in other optimization problems such as complementarity. One would like to generate valid inequalities that cut off some point not lying in X, without reference to the linear mapping. To this aim, we introduce a concept "cut-generating functions" and we develop a formal theory for them, based on convex analysis.

This chapter combines convex analysis and geometry: the study of cut-generating functions is based on their intimate relation with the so-called "S-free" sets (see Figure 1.5 and a precise definition Chapter 5). The article discloses several definitions for minimal cut-generating functions and maximal S-free sets, and puts in perspective a number of existing works. The main result of the chapter (Theorem 5.5.1) generalizes four prior results published in top journals in mathematical optimization ([68] [38][17] [18]).

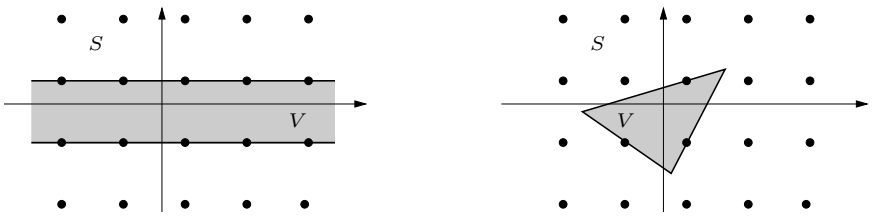


Figure 1.5: Two S-free sets V for $S = (-\frac{1}{4}, -\frac{1}{4}) + \mathbb{Z}^2$

Working out unifying results in a general framework required original mathematical proofs, largely based on geometric arguments. In particular, the chapter uses convex analysis results for

support functions in an original way: notice that section 5.3 generalizes the correspondence between support functions and gauge functions.

1.3.4 Variational analysis of alternating projections methods

The von Neumann's method of "alternating projections" and its variants provide simple but efficient tools to solve convex feasibility problems in engineering sciences. Such alternating projection methods also make sense for nonconvex feasibility problems and have been used extensively (see the references in the chapter or in [Mal-21]). In many applications, the linear convergence was observed but not explained by the existing theory, restrained to the convex case. Aiming to explain the success of the method in nonconvex settings, I studied during my post-doc at Cornell with Adrian Lewis the special case of transversal smooth manifolds [Mal-21]. We developed original proof techniques based on the variational analysis of nonsmooth coupling functions (rather than convex analysis tools as previous works). We then generalized these techniques to a more abstract nonconvex setting [Mal-20]. The convergence proof combines two ingredients (see Figure 6.1): a geometrical (transversality-like) condition of the intersection, controlling the "angle" between the two sets at a point of the intersection, together with a geometrical (convexity-like) property of one of the two sets, controlling the behaviour of projections.

The "transversality" is connected to the idea of a finite collection of closed sets having "linearly regular intersection" at a point, which is a central theoretical condition in variational analysis. We show that it also has striking algorithmic consequences for the convergence of the alternating projection method, which converges locally to a point in the intersection at a linear rate associated with the modulus of regularity of the intersection (see Theorem 6.5.1). As a consequence, in the case of several arbitrary closed sets having linearly regular intersection at some point, the method of "averaged projections" converges locally at a linear rate to a point in the intersection. Inexact versions of both algorithms also converge linearly.

This work, in addition, sheds some new light on the correspondence between several notions of conditioning for the feasibility problems, making connections between the sensitivity of solutions, the smallest perturbations destroying regularity, and the speed of convergence of basic algorithms (as illustrated on Figure 1.6).

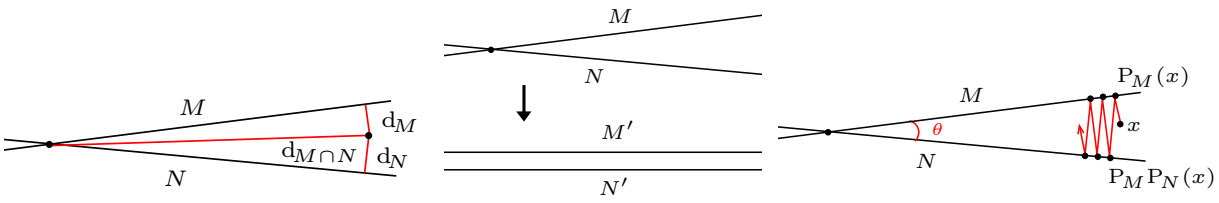


Figure 1.6: Illustration of the idea of "conditioning" for feasibility problems with two linear spaces. The drawn problem $M \cap N$ has a bad "condition number": (i) we have a weak error bound on the distance to the intersection from the distances to each set (picture on the left), (ii) small perturbations render the problem ill-posed (picture in the middle), and (iii) alternating projection converges with a slow rate $\cos\theta$ (picture on the right).

Chapter 2

Curriculum

This chapter gathers some elements of curriculum vitae, updated in summer 2016. For more information, visit my webpage:

`http://ljk.imag.fr/membres/Jerome.Malick/`

2.1 Short biography and distinctions

I am currently a CNRS "chargé de recherche" (CR1) at the Laboratoire Jean Kunztmann (LJK), on the Grenoble campus, in the research team DAO.

My trajectory to this current situation has been as follows. I was a student of the Ecole Normale Supérieure de Cachan (ENS Cachan) where I got the "agrégation" of mathematics[1], and of the engineering school of Grenoble ENSIMAG where I got a master in computer sciences and applied mathematics. After an amazing summer internship with Michael Overton at NYU, I felt in love with mathematical optimization, and I decided to make an PhD under the supervision of Claude Lemaréchal (Inria, Grenoble). I defended my PhD in 2005; the two referees were Jean-Baptiste Hiriart-Urruty (Université Paul Sabatier de Toulouse) and Adrian Lewis (Cornell University, US), and the president of the PhD committee Anatoli Iouditski (Université Joseph Fourier de Grenoble). Then I did my postdoc with Adrian Lewis at the Operation Research and Information Technologies department (ORIE) of Cornell University. At the end of 2006, I joined LJK as a junior researcher (CR2) in a joint Inria-LJK research team (Bipop). Since then, the split of my working time into the multiple facets of the researcher job is as follows: scientific research 45%, supervision & teaching 25%, management of research 20%, industrial relations & transfert 10%. The next sections describe these activities.

Let me quote below the scientific distinctions that I have received.

Second for CNRS bronze medal (2013) Every year, the committee of the "Section 6" of CNRS awards one "bronze medal" to a researcher in the broad domain of theoretical information sciences. I was proposed second for the bronze medal in 2013 by the committee.

Charles Broyden Prize (2011) Every year, the Charles Broyden prize is awarded to an outstanding article in journal *Optimization Methods and Software*. In 2011, the paper [Mal-16] with Didier Henrion was selected.

[1] The "agrégations" are ones of the most prestigious and selective civil service competitive examinations in France; the "agrégation de mathématiques" consists in a series of written and oral exams in algebra, analysis and applied maths. I have been eventually ranked 33th (over 320 admitted and out of 1584 candidates) at the agrégation of maths in 2002.

Robert Faure Prize (2009) The French association of Operations Research (ROADEF, equivalent to INFORMS in the US) awards every three years a prize to the most outstanding researcher under 35 years old. This prize has the name of Robert Faure, a pioneer of Operations Research in France; see more information on the ROADEF website. It was a great pleasure and a honor for me to receive this prize in 2009.

2.2 Supervision of young reseachers

I have (co-)supervised four post-doctoral students, four PhD students and six Master students. I was strongly committed to each of these supervisions, for the pleasure to interact with young scientists and the care of helping them to express their full abilities for research. Here I give some information about what we did together and their current position.

Supervision of PhD students

- Simon Boulmier (started in Oct. 2016) – lower bounds within a local search solver

  Simon is starting his PhD ("cifre" funding) while I am finishing this document. He is going to work on exploiting bounding within a local search solver, called LocalSolver, and developed by the company Innovation 24. I am looking forward to start this collaboration with them.

- Florent Bouchard (started in Oct. 2015) – Riemannian optimization for brainwave signals

  The goal of the Florent's PhD is to exploit brainwave signals correlated both in time and space, for example when measures are captured on several persons simultaneously. The idea is to find good formulations of this problem as riemannian optimization problems on matrix manifolds, and to develop algorithms robust to the high ratio noise/signal.

  Florent has two supervisors, an "applied" one(Marco Congedo, Gipsa-lab) and a "theoretical" one (myself). I am happy about how Florent entered in his PhD work; a first publication [Mal-29] came out from his first year.

- Federico Pierucci (about to finish) – Nonsmooth optimization for machine learning

  Typical machine learning problems, as collaborative filtering or multi-task classification, can be formulated as large-scale optimization problems with a ℓ_1-penalty on the spectrum of matrices. Conditional gradient algorithms then have good features to tackle them, but they require to formulate problems with smooth losses. The goal of the Federico thesis is to extend these algorithms to nonsmooth losses, using adapative smoothing techniques.

  Zaid Harchaoui, Anatoli Iouditski and myself have been co-supervising the Federico's PhD work. An article [Mal-30] was accepted in the French conference in machine learning, and two others are about to be submitted (to a statistics journal and to a numerical linear algebra journal).

- Sofia Zaourar (2011-2014) – Decomposition of large-scale problems

  Realistic operations research applications lead to large-scale mixed-integer optimization problems out of reach from direct approaches. Some of these problems, as in network design for example, can be treated by variable decomposition (also called Benders decomposition [83]) revealing an underlying nonsmooth optimization problem. The core of the Sofia's PhD was to design algorithmic accelerations of this decomposition, inspired from inexact level-bundle methods, able to deal with discrete variables.

I had witnessed Sofia becoming an inspired and independent researcher. Her PhD ended up with one article alone, two articles together [Mal-10] [186], and one [Mal-2] with Welington de Oliveira who was post-doc at that period. Sofia is now researcher at the Xerox European Research center.

- Florent Cadoux (2006-2009) – Applications to mechanics

  The numerical simulation of mechanical dynamical systems with contact and friction leads to equations and inclusions involving the Coulomb cone (also called the second-order cone or ice-cream cone in the optimisation community). The core of the Florent's PhD was to tackle such problem by extracting the underlying convexity and reformulate the problem as a fixed-point of a nonsmooth operator. Numerical experimentation shows that this approach (using a conic optimization engine) is suprisingly robust.

  Official PhD advisors of Florent's PhD were Claude Lemaréchal and Vincent Acary; I served like complementary advisor on many aspects: research (we have two papers together), teaching (Florent was my teaching assistant), and organisation (we organized the one-week workshop CAOA2010). Florent is now INPG-ErDF researcher in optimization for electrical networks.

Supervision of post-docs

- Welington de Oliveira (2011-2012) – Nonsmooth optimization algorithms

  The post-doc of Welington de Oliveira (student of Claudia Sagastizabal, Rio, Brazil) was funded by the French company EDF and supervised by Claude Lemaréchal (Inria) and myself. We delivered a code for their short-term electricity generation, and we were also able to conduct academic research which ended up with two articles, one with Claude [60] and the other one with Sofia and me [Mal-2]. Welington is currently assistant professor at Universidade do Estado do Rio de Janeiro.

- Nathan Krislock (2010-2012) – Semidefinite optimization

  I supervised the post-doctoral work of Nathan Krislock (student of Henry Wolkowicz, University of Waterloo, Canada) during two years. His stay allowed us to start the research and development around our software BiqCrunch. The collaboration has brought three articles [Mal-7], [Mal-4] and [Mal-1], and is still on-going: Nathan visited me 2 months in 2013 and in 2014. Nathan is now assistant professor at North Illinois University, US.

- Miroslav Dudik (fall semester in 2010) – Spectral regularizers in learning

  Zaid Harchaoui (at Inria Grenoble at this time) and myself invited Miro for a short post-doctoral stay after his PhD at Princeton and before he gets hired by Yahoo Research. This stay was the kick-off of my research on optimization algorithms for machine learning, which has led to two articles [Mal-33] et [Mal-32].

- Marc Fuentes (2008-2009) – Numerical optimization

  After his PhD with J.-B. Hiriart-Urruty (Toulouse), Marc joined Claude Lemaréchal and me to work on proximal regularization of degenerate smooth optimization problems [Mal-14]. Ideas developed during this post-doc have recently found a completely different application in learning [126]. Marc got an Inria research engineer position; he is now at Inria Bordeaux.

2.3 Scientific responsibilities

Main scientific responsibilities

- **Member of the board of GdR MOA ("Mathématiques de l'Optimisation et Applications")** since 2012. Animation of the French community of continuous optimization, organization of summer schools, and funding of workshops.
- **Head of a new team at LJK, created in 2015: DAO (optimization and learning for data science)**. I gathered 8 researchers of my lab LJK to make up this team with the goal of foster exchanges, create emulation, tackle interdisciplinary challenges, and gain visibility on these topics. The scientific positioning of the team is briefly discussed in section 7.3. More information are on the website `http://dao-ljk.imag.fr`.

Other scientific responsibilities

- Associate Editor for *Journal of Global Optimization*, since 2010
- Member of the advisory committee of the group MODE ("Maths de l'Optimisation et de l'aide à la DEcision") of the french applied maths association (SMAI, equivalent to SIAM in the US), since 2012
- Member of the Inria working group "Actions Incitatives" (2009-2011)
  (selection and evaluation of projects funded by Inria)
- Member of four hiring committees: for an assistant professor in optimization/learning at Grenoble in 2013; for the two researchers INRA in learning at Paris in 2015; for an assistant professor in optimization at Grenoble in 2015; for assistant professor in statistics at Grenoble in 2016.
- Member of PhD thesis committees
  - Jingwei Liang (Université de Caen Normandie) "Convergence rates of first-order operator splitting methods" (2016)
  - Carmen Cardozo (Supélec, Paris-Saclay) "Optimisation of power system security with high share of variable renewables" (2016)
  - Referee of the PhD of Alice Chiche (Paris VI) "Theory and algorithms for large-scale numerical optimization problems. Applications to electricity generation managment" (2012)

Seminar, conference, and workshop organization I have co-organized the following 7 events (in addition to those organized with GdR MOA). Click on the links in the pdf document to access to the dedicated webpages.

- Workshop "Optimization and Statistical Learning" (one week) first edition OSL2013 and second édition OSL2015
- Workshop TITAN "Large-scale inverse problems and optimization; Applications to image processing and astrophysics" (2015)
- Grenoble Optimization Day GOD14 (2014)
- Workshop "Advanced optimisation methods and their applications to unit commitment in energy management" at EDF (2011)

- Workshop "Convex Analysis, Optimization and Applications" CAOA2010 (I was Chair of both scientific and organization committees)
- Workshop GeoLMI on the geometry and algebra of linear matrix inequalities GeoLMI2009

I have also coordinated the following three seminars and reading groups.

- OGre Seminar (Optimization, learning and applications at Grenoble) gathering people from different Grenoble labs (LJK, LIG, Gipsa-lab, G-Scop, CEA, G2e-lab, Inria) (2015-present)
- Seminar of the teams BIPOP&CASYS of LJK (2007-2012)
- Inria reading group on optimisation and machine learning (2010-2011)

2.4 Teaching and popularization

Since my PhD, I have always taught (with pleasure!) more than 60 hours a year (except in 2006, the year of my post-doc). I also wrote a teaching book (in French) with my two friends from ENS

> *Objectif Agrégation*, V. Beck, J. Malick and G. Peyré
> (published by Editions H&K, distributed by Vuibert)
> First edition: August 2004, ISBN : 2 914010 58 3 (700 sold copies)
> Second extended edition: August 2005, ISBN : 2 914010 92 3 ($\geq$ 2200 sold copies)

I have taught courses at different levels (from undergraduate to graduate) in different universities and engineering schools. In additional to traditional applied maths courses, I have introduced and enhanced over time new courses closer to my research activities. My activities have covered the entire teaching spectrum from elaborating the courses content, to practical lab manipulations, and exams.

- **Numerical Optimisation (2007-present)** I am in charge of teaching continuous optimization at the Grenoble Engineering School ENSIMAG. The Master1 course consists of around 65 hours (lectures and exercises on tables or machines) and is taken by around 100 students at each fall semester. According to the students' assessment in 2014, the course got the best grade among comparable courses[2].
- **Mathematical programming and industrial applications (2013, 2014, and 2015)** In the Grenoble master of Computer Science, major operation research and combinatorics (around 15 students).
- **Optimization and convexity (2011)** Intensive course (40 students, 24h concentrated in one week!) in the computer science master program of ENS Lyon
- **Optimization for finance (2010)** Applied optimization course (18h) for Master 1 students in financial mathematics at ENSIMAG
- **Teaching assistant** in applied maths at Grenoble University (2003-2005). I mainly taught analysis in Licence for 64h/y.

[2]More specifically, it was ranked best course among mandatory courses by the students in financial mathematics, and best course among optional courses by students in applied maths)

Three new courses will start in fall/winter 2016 : Stochastic Optimization (in the Master of Computer Science of Grenoble), Numerical linear algebra and optimization (in the Master of Applied Maths of Grenoble), Convex and Distributed Optimization (in the major Data Science common to the Master of Applied Maths and the Master of Informatics of Grenoble).

I have also had some activities at destination to a broader audience; I have tried to promote convex optimization and convex analysis as soon as I had the opportunity :)

- Preparation of an exam for the competitive entrance to ENS Cachan in 2012 (available on line, click here on the pdf document)
- Two popularization articles on the optimization of electricity generation:
  - to the optimization community: the paper [94] explains the "unit-commitment" problem at EDF the French electricity Board; it was published in the newsletter de *Mathematical Optimization Society*.
  - to a broader audience: the paper [93] explains the "success story" of the collaboration between Inria/CNRS and EDF; it was published by *European Science Foundation* in a book *Mathematics and Industry: success stories*.
- Invited speaker at the "colloque inter-académique des inspecteurs de maths" (2009)
  I gave a tribute-lecture to convex analysis at this workshop gathering persons in charge of maths programs and teachers evaluation in french high-schools.
- Article [130] in the newsletter of the french operation research community (2009).

2.5 Projects, fundings, and industrial contacts

Research projects

- Leader of a “PGMO” project (Gaspard Monge Program for Optimization) called "advanced nonsmooth optimization methods for stochastic programming" (2016). I have gathered a task-force of 4 colleagues on the application of modern nonsmooth optimization to tackle large-scale stochastic optimization problems.
- Leader of project TITAN of the CNRS “Mastodons” Big Data Challenge (2015). The topic of the project was optimization and learning with a special interest to image reconstruction in astro-physics. I was the scientific coordinator of the project involving 36 researchers over 8 laboratories (in applied maths, computer science, and signal processing). Several follow-up projects emerged from generated interactions.
- Coordinator for my lab of the ANR Project GeoLMI (2011-2015). This project focused on polynomial optimization, and included 15 persons from 6 laboratories. I was coordinator for my lab, and in charge of the task “semidefinite programming algorithms”.
- Leader of a Math-STIC project "LMI-SDP-2" funded by University Grenoble Alpes (2009-2011). This ambitious project aimed at developing a new generation of semidefinite programming tools for control an combinatorics; it gave birth to the software biqcrunch. The funding supported the post-doctorial stays of Miro Dudik (Microsoft Research, NYC) et Nathan Krislock (Waterloo University, Canada).

- Leader of a GdR RO (Recherche Opérationnelle) project (2009, renewed in 2010). This small project with 4 persons was the start of the work on exact resolution of hard combinatorial optimization problems.

- I was an active member of the following projects:
  - Project "Gargantua" of CNRS (2013, renewed in 2014)
    Topic: Big Data Optimization
  - Khronos team of the LabEx Persyval-Lab (2014-2017)
    Topic: Learning and time series
  - "PGMO" project "Consistent Dual Signals and Optimal Primal Solutions" (2012-2015)
    Topic: decomposition of large-scale unit-commitment optimization problems
  - Project "Parsimat" funded by Grenoble University Alpes (2012-2013)
    Topic: Matrix learning with spectral sparsity
  - ANR project "SalaDYN" (2009-2011)
    Topic: Numerical simulation of mechanical dynamical systems
  - Two CNRS "PEPS" projects : "GeoLMI" (2009-2010) et "Autoélèmi" (2010-2011)
    Topic: Geometry of semidefinite optimization and applications to control
  - Project "CARESSE" funded by Grenoble University Alpes (2008-2010)
    Topic: Graphs and dynamical networks
  - ANR project "Guidage" (2005-2008)
    Topic: Optimization and control for mechanics and aeronautics

Consulting activity: robust optimization for Finance I was consultant for the start-up RaisePartner which provides software and expertise on numerical quantitative finance. I worked for them one day per week during two years between 2007 and 2009. I helped them setting up robust decision-making tools embedded in their products.

Main industrial collaboration: Optimization of EDF electricity generation park In 2009 with Claude Lemarechal getting close to retirement, I took the lead of the collaboration with EDF. I was in charge of two contracts with EDF (in 2009 and 2010). Since 2012, the collaboration fits in the Program Gaspard Monge for Optimization (PGMO) framework with two projects (in 2012 and 2016).

The EDF electricity generation optimization problem is important industrial problem whose solution involved much convex analysis and optimization [74]. I contributed on various aspects: EDF internal research (with a report on marginal prices with C. Lemaréchal in 2010); improvement of their operational algorithm (the supervision of a master student and post-doc); academic research oriented to energy applications ([Mal-10], [Mal-2], [Mal-5]). This resulted in helping EDF in saving money (power generating costs) and emission of pollution (CO2 and nuclear waste). See more in section 1.2.

2.6 Publications

Here are the list of my articles published in journals and the list of publications in books or refereed proceedings of conferences. In the two categories, the publications are ranked in inverse chronological order. These publications are available, as preprint, on the French open access archive HAL. Links to my HAL entries and my Google Scholar profile are on my webpage

`http://ljk.imag.fr/membres/Jerome.Malick/publis.html`

I briefly present in the previous chapter 1 of this document all these 35 publications in an overview of my research work. In next chapters 3, 4, 5, and 6, I discuss further only 10 of these articles.

Published papers

[Mal-1] N. Krislock, J. Malick, and F. Roupin. Biqcrunch: a semidefinite branch-and-bound method for solving binary quadratic problems. *To appear in ACM Transactions on Mathematical Software*, 2016.

[Mal-2] J. Malick, S. Zaourar, and W. Oliveira. Uncontrolled inexact information within bundle methods. *To appear in EURO Journal on Computational Optimization*, 2016.

[Mal-3] W. van Ackooij and J. Malick. Second-order differentiability of probability functions. *To appear in Optimization Letters*, 2016.

[Mal-4] N. Krislock, J. Malick, and F. Roupin. Computational results of a semidefinite branch-and-bound algorithm for k-cluster. *Computers and Operations Research*, 66:153–159, 2016.

[Mal-5] W. van Ackooij and J. Malick. Decomposition algorithm for large-scale two-stage unit-commitment. *Annals of Operations Research*, 238(1):587–613, 2016.

[Mal-6] A. Daniilidis, J. Malick, and H. Sendov. Spectral (isotropic) manifolds and their dimension. *Journal d'Analyse Mathématique*, 128(1):369–397, 2015.

[Mal-7] N. Krislock, J. Malick, and F. Roupin. Improved semidefinite bounding procedure for solving max-cut problems to optimality. *Mathematical Programming*, 143(2):61–86, 2014.

[Mal-8] M. Conforti, G. Cornuéjols, A. Daniilidis, C. Lemaréchal, and J. Malick. Cut-generating functions and s-free sets. *Mathematics of Operations Research*, 40, 2014.

[Mal-9] A. Daniilidis, J. Malick, and H. Sendov. On the structure of locally symmetric manifolds. *Journal of Convex Analysis*, 22, 2014.

[Mal-10] S. Zaourar and J. Malick. Prices stabilization for inexact unit-commitment problems. *Mathematical Methods of Operations Research*, 78(3):341–359, 2013.

[Mal-11] J. Malick and F. Roupin. On the bridge between combinatorial optimization and nonlinear optimization: a family of semidefinite bounds for 0-1 quadratic problems leading to quasi-newton methods. *Mathematical Programming B*, 140(1):99–124, 2013.

[Mal-12] P.-A. Absil and J. Malick. Projection-like retractions on matrix manifolds. *SIAM Journal on Optimization*, 22(1):135–158, 2012.

[Mal-13] J. Malick and F. Roupin. Solving k-cluster problems to optimality with semidefinite programming. *Mathematical Programming B*, 136:279–300, 2012. Special issue on Mixed-Integer Nonlinear Programming.

[Mal-14] M. Fuentes, J. Malick, and C. Lemarechal. Inexact proximal algorithms for differentiable optimization. *Computational Optimization and Applications*, 53:755–769, 2012.

[Mal-15] J.-B. Hiriart-Urruty and J. Malick. A fresh variational-analysis look at the positive semidefinite matrices world. *Journal of Optimization Theory and Applications*, 153(3):551–577, 2012.

[Mal-16] D. Henrion and J. Malick. Projection methods for conic feasibility problems; application to sum-of-squares decompositions. *Optimization Methods and Software*, 26(1):23–46, 2011.

[Mal-17] F. Cadoux and J. Malick. Existence of a fixed point of a nonsmooth function arising in numerical mechanics. *Set-Valued Analysis and Variational Analysis*, 18(4), 2010.

[Mal-18] F. Cadoux, J. Malick, V. Acary, and C. Lemarechal. A formulation of the linear discrete coulomb friction problem via convex optimization. *ZAMM (Zeitschrift für Angewandte Mathematik und Mechanik)*, 94(2), 2010.

[Mal-19] J. Malick, J. Povh, F. Rendl, and A. Wiegele. Regularization methods for semidefinite programming. *SIAM Journal on Optimization*, 20(1):336–356, 2009.

[Mal-20] A. Lewis, D. Luke, and J. Malick. Local linear convergence for alternating and averaged nonconvex projections. *Foundations of Computational Mathematics*, 9:485–513, 2009.

[Mal-21] A.S. Lewis and J. Malick. Alternating projections on manifolds. *Mathematics of Operations Research*, 33(1):216–234, 2008.

[Mal-22] A. Daniilidis, A. Lewis, J. Malick, and H. Sendov. Prox-regularity of spectral functions and spectral sets. *Journal of Convex Analysis*, 15(3), 2008.

[Mal-23] J. Malick. Spherical constraint in Boolean quadratic programming. *Journal of Global Optimization*, 39(4), 2007.

[Mal-24] A. Daniilidis, W. Hare, and J. Malick. Geometrical interpretation of proximal-type algorithms in structured nonsmooth optimization. *Optimization*, 55(5):481–503, 2006.

[Mal-25] A. Daniilidis and J. Malick. Filling the gap between lower-C^1 and lower-C^2 functions. *Journal of Convex Analysis*, 12(2), 2005.

[Mal-26] J. Malick and S.A. Miller. Newton methods for convex minimization : connection among $\mathcal{U}$-lagrangian, Riemannian Newton and SQP methods. *Mathematical Programming*, 104(3), 2005.

[Mal-27] J. Malick and H. Sendov. Clarke generalized Jacobian of the projection onto the cone of positive semidefinite matrices. *Set-Valued Analysis*, 14(3):273–293, 2006.

[Mal-28] J. Malick. A dual approach to semidefinite least-squares problems. *SIAM Journal on Matrix Analysis and Applications*, 26, Number 1:272–284, 2004.

Book chapters, conference proceedings

[Mal-29] F. Bouchard, L. Korczowski, J. Malick, and M. Congedo. Approximate Joint Diagonalization within the Riemannian Geometry Framework. *European Signal Processing Conference (Eusipco)*, 2016.

[Mal-30] F. Pierucci, Z. Harchaoui, and J. Malick. A smoothing approach for composite conditional gradient with nonsmooth loss. *Conférence française d'apprentissage CAp*, 2013. Research Report RR-8662, INRIA Grenoble, July 2014.

[Mal-31] M. Conforti, G. Cornuéjols, A. Daniilidis, C. Lemaréchal, and J. Malick. Cut-generating functions. In Michel Goemans and Jose Correa, editors, *Integer Programming and Combinatorial Optimization (IPCO)*, volume 7801 of *Lecture Notes in Computer Science*, pages 123–132. Springer Berlin Heidelberg, 2013.

[Mal-32] Z. Harchaoui, M. Douze, M. Paulin, M. Dudik, and J. Malick. Large-scale classification with trace-norm regularization. *IEEE Conference on Computer Vision & Pattern Recognition (CVPR)*, 2012.

[Mal-33] M. Dudik, Z. Harchaoui, and J. Malick. Lifted coordinate descent for learning with trace-norm regularization. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2012.

[Mal-34] D. Henrion and J. Malick Projection methods in conic optimization. *Chapter of the Handbook of semidefinite, conic and polynomial optimization*, edited by M. Anjos and J.B. Lasserre. Springer, International Series in Operations Research & Management Science, volume 166, 2012.

[Mal-35] J. Malick and F. Roupin. Numerical study of semidefinite bounds for the k-cluster problem. In *Electronics Notes of Discrete Mathematics*, pages 399–406. Elsevier, 2010. ISCO'10, International Symposium on Combinatorial Optimization.

Co-authors All the above publications are the results of a team work with colleagues or students:

- researchers in French universities or research units:
  - Vincent Acary (research associate – Inria Grenoble)
  - Vincent Beck (assistant professor – Orléan University)
  - Florent Cadoux (researcher – Grenoble)
  - Marco Congedo (researcher – CNRS Grenoble)
  - Marc Fuentes (research engineer – Inria, Bordeaux)
  - Didier Henrion (senior researcher – LAAS CNRS)
  - Jean-Baptiste Hiriart-Urruty (professor – Université de Toulouse)
  - Claude Lemaréchal (senior researcher – Inria Grenoble)
  - Gabriel Peyré (senior researcher – ENS Ulm, Paris)
  - Frédéric Roupin (professor – Paris XIII)
- researchers in foreign universities
  - Pierre-Antoine Absil (professor – Université de Louvain-la-Neuve, Belgique)
  - Michele Conforti (professor – Université de Padova, Italie)
  - Gérard Cornuéjols (professor – Carnegie Mellon University, USA)
  - Aris Daniilidis (professor – Universidad de Chile)
  - Zaid Harchaoui (professor – University of Washington, US)
  - Warren Hare (professor – University of British Columbia)
  - Nathan Krislock (assistant Professor – University of North Illinois)
  - Adrian Lewis (professor – Cornell University)
  - Russel Luke (professor – Göttinberfg Universität, Germany)

- Welington de Oliveira (assistant professor – Universidade do Estado do Rio de Janeiro)
- Janez Povh (assistant Professor – Maribor, Slovenia)
- Franz Rendl (professor – Universität Klagenfurt)
- Hristo Sendov (professor – London University)
- Angelika Wiegele (assistant Professor – Universität Klagenfurt)

- researchers in compagnies

  - Wim van Ackooij (researcher – EDF)
  - Mathijs Douze (researcher – Facebook, France)
  - Miroslav Dudik (researcher – Microsoft Research, NYC, USA)
  - Grace Hechme-Doukopoulos (researcher – EDF Research, France)
  - Sandrine Brignol-Charousset (head of PGMO, France)
  - Scott Miller (researcher – Numerica Corp., USA)
  - Sophie Volle (CEO of RaisePartner SAS)
  - Sofia Zaourar-Michel (researcher – Xerox European Research Center)

I wish to acknowledge at this occasion the passion and talent they have put in our joint work. I have learn a lot, interacting with them (and not only about science !). I am aware of this luxury of our job: we can choose those you want to work with.

Chapter 3

Semidefinite optimization for binary quadratic problems

This chapter is the achievement of a line of research started in 2007 with Frédéric Roupin (Paris XIII) on semidefinite optimization for quadratic problems with binary variables. More precisely, this chapter corresponds to the article [Mal-1] with Frédéric Roupin and Nathan Krislock (now at North Illinois University) to appear in ACM Transaction on Mathematical Software, augmented with short parts of [Mal-11], [Mal-7], and [Mal-4].

3.1 Introduction: binary quadratic problems and solvers

3.1.1 Binary quadratic optimization problems

We consider binary quadratic optimization problems, i.e., (nonconvex) optimization problems with a quadratic objective, quadratic constraints, and 0–1 variables. A binary quadratic problem with m_I inequality constraints and m_E equality constraints has the following mathematical formulation:

$$\begin{cases} \text{maximize} & z^T S_0 z + s_0^T z & \\ \text{subject to} & z^T S_i z + s_i^T z \leq a_i, & i \in \{1, \ldots, m_I\} \\ & z^T S_i z + s_i^T z = a_i, & i \in \{m_I + 1, \ldots, m_I + m_E\} \\ & z \in \{0,1\}^n & \end{cases} \tag{3.1}$$

where the S_i's are real symmetric $n \times n$ matrices (possibly $S_i = 0$), the s_i's are vectors in $\mathbb{R}^n$, and the a_i's are real numbers. Many optimization problems in the sciences, operations research, or engineering are expressed as binary quadratric problems, such as, in medicine [104], in physics [125], in space allocation [7], in computer vision [107], or in computational biology [76].

Three examples of classical combinatorial optimization problems that can be expressed as problem (3.1) are Max-Cut, Max-k-Cluster, and Max-Independent-Set. In the Max-Cut problem (see, e.g., [84, 151]), we are given an edge-weighted graph with n vertices, and the objective is to maximize the total weight of the edges between a subset of vertices and its complement; this problem can be stated as:

$$(\text{Max-Cut}) \quad \begin{array}{ll} \text{maximize} & \sum_{ij} w_{ij} z_i (1 - z_j) \\ \text{subject to} & z \in \{0,1\}^n. \end{array} \tag{3.2}$$

In the Max-k-Cluster problem, we are given an edge-weighted graph with n vertices and a natural number k, and the objective is to find a subgraph of k nodes having maximum total edge weight; this

problem can be stated as:

$$\begin{array}{rll} & \text{maximize} & \frac{1}{2}\sum_{ij} w_{ij} z_i z_j \\ (\text{Max-}k\text{-Cluster}) & \text{subject to} & \sum_{i=1}^{n} z_i = k \\ & & z \in \{0,1\}^n. \end{array} \tag{3.3}$$

In the Max-Independent-Set (MIS) problem (see, e.g., [187]), we are given a graph $G = (V, E)$ with vertex weights w_i, and the objective is to maximize the total weight of the vertices in an independent set (a set S of vertices having no two vertices joined by an edge in E); this problem can be stated as:

$$\begin{array}{rlll} & \text{maximize} & \sum_i w_i z_i & \\ (\text{MIS}) & \text{subject to} & z_i z_j = 0, & \forall (i,j) \in E \\ & & z \in \{0,1\}^n. & \end{array} \tag{3.4}$$

These three problems, and more generally binary quadratic problems, are NP-hard and are often difficult to solve in practice.

This chapter introduces *BiqCrunch*, an exact solver for general binary quadratic (*biq*) optimization problems. Extensive numerical experiments show that *BiqCrunch* is the current state-of-the-art for several difficult binary quadratic optimization problems. The source code is available online and distributed under the GNU General Public License, version 3.

The remainder of the introduction sketches the existing solvers and the contributions of *BiqCrunch*. The mathematical foundations of *BiqCrunch* are presented in Section 3.3, its algorithmic description in Section 3.4, and finally advanced techniques for improving its performance in Section 3.5. Further information is available on the *BiqCrunch* website:

`http://lipn.univ-paris13.fr/BiqCrunch/`

3.1.2 Existing solvers for binary quadratic optimization

Binary quadratic programming is included in the broader class of mixed-integer nonlinear programming [43, 44, 56]. Thus problem (3.1) could be handled directly by using mixed-integer nonlinear programming solvers, such as the commercial solvers BARON [161], LocalSolver, Gurobi, and IBM/CPLEX, as well as the noncommercial solvers SCIP [1] and Bonmin [35]. However these mixed-integer nonlinear programming solvers do not fully exploit the quadratic form of the objective function and the constraints in problem (3.1), except in preprocessing phases.

In contrast, another widely used technique for solving binary quadratic problems is to add linearization variables to formulate problem (3.1) as a binary linear programming problem; see, e.g., [167]. The advantage of this approach is the possibility of using all the available efficient tools for integer linear programming. However, for hard combinatorial problems, it is often necessary to go beyond standard linear bounds and work with tighter bounds. For example, for graph problems that are very sparse, linear-based solvers that take advantage of the sparsity and the geometric properties of underlying problems usually perform well; however for small dense problems, they can perform poorly.

The quadratic nature of the objective function and the constraints of problem (3.1) implies that we can use semidefinite relaxations of problem (3.1) to get tight bounds (see, e.g., [84, 121, 147, 168]). Currently, there are three types of semidefinite-based solvers for binary quadratic problems. The first type is semidefinite branch-and-bound methods specialized for solving specific subclasses of problems (3.1), such as the semidefinite solver of [9] for graph bisection problems, and the *Biq Mac* solver of [151] for the Max-Cut problem (3.2). The second type of semidefinite-based solvers is the

quadratic convex reformulation for mixed-integer quadratic problems [29, 30, 81] which uses semidefinite bounds at the root node to give a boost to linear programming based branch-and-bound methods.

The third type of semidefinite-based solvers are standard branch-and-bound methods replacing linear programming solvers with semidefinite programming solvers, such as SCIP-SDP [82, 131]. SCIP-SDP solves general mixed-integer semidefinite programming (MISDP) problems, which implies that it is able to solve generic binary quadratic problems after making a suitable transformation of the problem to an MISDP. SCIP-SDP uses a standard branch-and-bound approach where bounds are obtained by solving the SDP relaxation that is obtained by simply relaxing the integer constraints—this SDP relaxation is then solved by a standard SDP solver, such as an interior-point method. SCIP-SDP must use several safe-guards against failures to solve the SDP relaxation due to the loss of strict feasibility that can occur when branching, and is limited to solving only small to medium-sized problems.

3.1.3 *BiqCrunch*, a free solver for binary quadratic problems

In this chapter, we introduce *BiqCrunch*, an open-source code for solving binary quadratic optimization problems to optimality. *BiqCrunch* is a branch-and-bound algorithm using generic or specific heuristics to compute lower-bounds and an original adaptive bounding procedure to compute upper-bounds. The bounding procedure automatically adjusts several parameters to efficiently produce a wide range of tightness levels from rough bounds to tight semidefinite-quality bounds.

BiqCrunch is of particular interest for solving hard problems which are very difficult to solve using linear-bounds. *BiqCrunch* therefore complements the currently available software packages mentioned in the previous section. Generally speaking, the set of problems for which linear-bounds underperform are the problems best-suited for the *BiqCrunch* solver. Compared to other semidefinite-based solvers, *BiqCrunch* offers a flexible and efficient bounding procedure that can produce a range of bounds with a varying degree of tightness.

The *BiqCrunch* solver is available as:

- an open-source code for solving problem (3.1);
- specific versions of the software for different standard combinatorial problems;
- a simple online interface.

The *BiqCrunch* solver is written in C (and uses a Fortran library). The distribution also includes converters and heuristics written in C and Python. The code is developed using established numerical tools, namely: basic linear algebra functions in LAPACK [3] or the Intel Math Kernel Library (MKL), the nonlinear optimization routine L-BFGS-B [133, 188], and the branch-and-bound platform BOB [55].

We have conducted extensive computational tests on classical NP-hard combinatorial problems, known to be difficult to solve even for many medium-sized instances. As discussed the results in section 3.6, the computational results provide strong evidence that *BiqCrunch* is among the best solvers for solving to optimality combinatorial optimization problems.

3.1.4 Outline of the chapter

The material of this chapter accompanies the public release of the *BiqCrunch* code by providing a complete description of the solver and how to use it. We first present some basic information and

examples on how to use *BiqCrunch* in Section 3.2; a complete description is available in the user manual that is distributed with *BiqCrunch*. The mathematical foundations of *BiqCrunch* are presented in Section 3.3 where we will recall the standard strengthened semidefinite bounds for problem (3.1) and, motivated by the desire to have semidefinite quality bounds without the inherent computational cost of the standard bounds, we will describe the original semidefinite bounds that are used in *BiqCrunch*. The two main algorithmic ingredients are then described in Section 3.4: the generic heuristic for computing feasible solutions (i.e., lower bounds) and the efficient procedure for computing upper bounds. In addition, we provide an analysis of the theoretical convergence of the semidefinite bounding procedure. n In Section 3.5, we discuss the parameters of the code and the advanced use of *BiqCrunch*. Finally, we report some numerical comparisons in Section 3.6.

3.2 *BiqCrunch* in practice, examples, illustrations

The latest version of the *BiqCrunch* code is available from the *BiqCrunch* webpage. Installation instructions are included with the source code. We have made the installation straightforward, only requiring a C compiler, a Fortran compiler, and either LAPACK or the Intel MKL.

Once *BiqCrunch* has been installed, it can be run from the command-line as follows.

```
$ biqcrunch [-v 1] <INSTANCE> <PARAMETERS>
```

The optional parameter `-v` is the verbosity; `<INSTANCE>` is the input file in the *BiqCrunch* format; `<PARAMETERS>` is a parameters file which can be one of the files provided with the code, or a user's own file.

This section provides some information about the format of the input file (in Section 3.2.1) and examples on how to use *BiqCrunch* (in Sections 3.2.2 and 3.2.3). We refer to the user manual for complete information on installing and running *BiqCrunch*, and to Section 3.5 for a discussion of the parameters.

3.2.1 Matrix formulation and input file format

We describe briefly here the matrix formulation of the binary quadratic problem (3.1) on which the *BiqCrunch* input file format is based. First we introduce the usual inner product of two matrices and the associated norm (sometimes called the Frobenius norm), respectively defined by

$$\langle X, Y \rangle = \operatorname{trace}(X^T Y) = \sum_{ij} X_{ij} Y_{ij} \qquad \text{and} \qquad \|X\|_F = \sqrt{\langle X, X \rangle} = \sqrt{\sum_{ij} X_{ij}^2}.$$

Since $z^T S_i z = \langle S_i, zz^T \rangle$, this inner product allows us to rewrite the quadratic terms $z^T S_i z + s_i^T z$ of problem (3.1) as linear terms $\langle Q_i, Z \rangle$ where

$$Z = \begin{bmatrix} zz^T & z \\ z^T & 1 \end{bmatrix} \qquad \text{and} \qquad Q_i = \begin{bmatrix} S_i & \frac{1}{2} s_i \\ \frac{1}{2} s_i^T & 0 \end{bmatrix}.$$

Thus, the binary quadratic problem (3.1) can be reformulated as:

$$\left\{ \begin{array}{lll} \text{maximize} & \langle Q_0, Z \rangle & \\ \text{subject to} & \langle Q_i, Z \rangle \leq a_i, & i \in \{1, \ldots, m_I\} \\ & \langle Q_i, Z \rangle = a_i, & i \in \{m_I + 1, \ldots, m_I + m_E\} \\ & Z = \begin{bmatrix} zz^T & z \\ z^T & 1 \end{bmatrix}, \; z \in \{0,1\}^n . & \end{array} \right. \tag{3.5}$$

Note that the objective function and the constraints are now linear with respect to Z, and that the only non-convexity of the problem lies in the form of Z, which is a rank-one matrix with 0–1 entries.

BiqCrunch requires the objective value of (3.5) to be integer for any feasible solution. This corresponds to having integers on the diagonal of Q_0 and integers divided by two on the off-diagonal entries of Q_0. *BiqCrunch* takes advantage of this feature by pruning the branch-and-bound search tree when the computed bound is strictly less than $\beta + 1$, where β is the objective value of the current best feasible solution; see Section 3.4.1. To use *BiqCrunch* with fractional data, one should first multiply the coefficients by the smallest common denominator to make them integers.

The matrix formulation in problem (3.5) is used in the input format of the solver. The *BiqCrunch* format is similar to the widely used sparse `SDPA` format in semidefinite optimization; see [181]. Roughly speaking, it consists of specifying general parameters (m, n, type of constraints, etc.) and describing the matrices Q_i in a sparse matrix format. The *BiqCrunch* solver stores the input problem matrices in this sparse format in memory to keep its memory requirements small. The main difference between the *BiqCrunch* format and the sparse `SDPA` format is that the first line of a *BiqCrunch* input file indicates if the problem is a maximization problem (using $+1$) or a minimization problem (using -1). Moreover the *BiqCrunch* format uses a block of size $n + 1$ to represent the positive semidefinite matrix and a diagonal block of slack variables (for inequality constraints). The *BiqCrunch* file format is fully described and illustrated in the user manual. We also give an example in the next section.

To write a *BiqCrunch* file, a user would need to have a good understanding of the SDP relaxation and how to write it in `SDPA` format. This was a major barrier to being able to use *BiqCrunch* before we created an `lp2bc` converter. To simplify the use of *BiqCrunch*, we provide two types of converters to the *BiqCrunch* format:

1. A converter from the so-called LP format to the *BiqCrunch* format: `lp2bc`.

2. Specific converters for each problem class; for example `mc2bc` to convert Max-Cut problems and `kc2bc` to convert Max-k-Cluster problems.

These conversion tools are described in the user manual. In the next two sections, we give examples of the use of *BiqCrunch* to solve a generic problem specified in the LP format, and a Max-Cut problem specified in a standard sparse format.

Let us emphasize that *BiqCrunch* works directly on problem (3.5) given by the Q_i's and a_i's. This is in contrast with CPLEX (version 12.1) and Gurobi (version 5.6), which both preprocess the entries and in particular convexify binary quadratic problems to work with positive semidefinite matrices. Such automatic reformulations or convexifications are not always efficient as they can negatively affect the solution process. (Smarter convexifications use semidefinite optimization, as exploited in [29, 30]). In *BiqCrunch* there is no reformulation phase and no data preprocessing phase. The user has complete control over the problem and the way it is modeled. We give some advice on how to enhance the problem formulation in Section 3.5.3.

3.2.2 Example with the LP converter

We give an illustration of running *BiqCrunch* on a simple test problem, using the converter from the human-readable LP format of IBM/CPLEX to the *BiqCrunch* format. Consider the binary quadratic problem

$$\begin{cases} \text{maximize} & z_1 z_2 + 2 z_1 z_3 \\ \text{subject to} & z_1 + z_2 + z_3 \leq 2 \\ & (z_1, z_2, z_3) \in \{0,1\}^3 \end{cases}$$

whose optimal solution is $(1, 0, 1)$. We describe this problem in the LP format as a file called `example.lp` containing the following lines:

```
maximize
    z1*z2 + 2 z1*z3
subject to
    z1 + z2 + z3 <= 2
binary
    z1 z2 z3
end
```

The `lp2bc` converter is called by the command line

```
$ lp2bc.py example.lp > example.bc
```

and generates the following file in *BiqCrunch* format:

```
# List of binary variables:
#    1:  z1
#    2:  z2
#    3:  z3
1 = max problem
1 = number of constraints
2 = number of blocks
4, -1
2.0
0 1 1 2 0.5
0 1 1 3 1.0
1 1 1 4 0.5
1 1 2 4 0.5
1 1 3 4 0.5
1 2 1 1 1.0
```

The *BiqCrunch* format is similar to the sparse SDPA format and defines the problem by specifying the coefficient matrices Q_i, which constraints are inequalities, and the right-hand-side values a_i of all the constraints. For a complete description of the *BiqCrunch* format, see the *BiqCrunch* User's Guide which is available on the *BiqCrunch* website.

We solve now the problem using *BiqCrunch* (with default parameters) by executing

```
$ biqcrunch example.bc generic.param
```

We obtain the following command-line output:

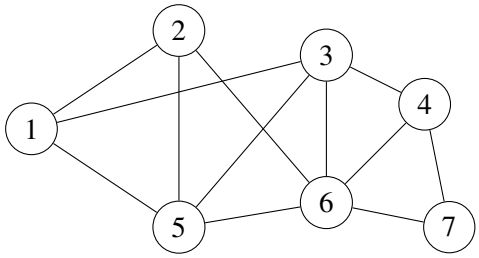
```
Output file: example.bc.output
Input file:  example.bc
Parameter file: generic.param
Node 0 Feasible solution 2
Nodes = 1
Root node bound = 2.84
Maximum value = 2
Solution = { 1 3 }
CPU time = 0.0074 s
```

This output reports the result of running of *BiqCrunch* on this instance. At the root node of the branch-and-bound search tree, the computed bound is 2.84, and, since the optimal value is integer, this gives an effective upper-bound of 2. The generated solution was $z_1 = z_3 = 1$ and $z_2 = 0$ with objective value 2, which proves that it is an optimal solution. Thus there was only one node in the branch-and-bound search. More detailed output information is given in the output file.

3.2.3 Example with the Max-Cut converter

We give an illustration of using *BiqCrunch* to solve a Max-Cut problem on a simple graph, using the converter `mc2bc` to create a *BiqCrunch* input file from a graph file. Let us consider the following graph, drawn in the figure and described in a file `graph.txt`. The first line of `graph.txt` records the number of nodes and number of edges in the graph and the following lines record the list of edges, each written as the triple $i\ j\ w_{ij}$. Note that each edge in this graph has a weight of 1.

```
7 12
1 2 1
1 3 1
1 5 1
2 5 1
2 6 1
3 4 1
3 5 1
3 6 1
4 6 1
4 7 1
5 6 1
6 7 1
```

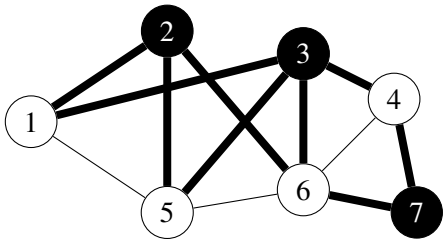


Using the `mc2bc` converter, we compute a maximal cut for this graph using *BiqCrunch* as follows:

```
$ mc2bc.py graph.txt > maxcut.bc
$ biqcrunch maxcut.bc maxcut.param
```

BiqCrunch returns the following output giving us a maximum cut which we have represented in the figure. The nine edges between the black nodes and the white nodes are a maximal cut.

```
Output file: maxcut.bc.output
Input file:  maxcut.bc
Parameter file: maxcut.param
Node 0 Feasible solution 9
Nodes = 1
Root node bound = 9.91
Maximum value = 9
Solution = { 1 4 5 6 }
CPU time = 0.0075 s
```



3.3 Mathematical foundations

In this section, we give a theoretical description of the bounds used by *BiqCrunch*. We start by recalling some basic facts about semidefinite bounds in Section 3.3.1, then we present the special semidefinite bounds used by *BiqCrunch* in Section 3.3.2.

We refer the interested reader to the books [162] and [6] for more information, including historical perspectives, about semidefinite programming in the context of combinatorial optimization.

3.3.1 Semidefinite relaxations

We first introduce some notation and briefly describe the standard semidefinite bounds for the binary quadratic problem (3.1) written as its matrix form as problem (3.5). The presentation of bounds in the next section becomes more straightforward when the problem is reformulated using $\{-1,1\}$ variables. We apply the change of variables between $z \in \{0,1\}^n$ and $x \in \{-1,1\}^n$ defined by $z = \frac{1}{2}(x+e)$, where e is the vector of all ones. This can be written as

$$\begin{bmatrix} z \\ 1 \end{bmatrix} = U \begin{bmatrix} x \\ 1 \end{bmatrix} \qquad \text{with} \qquad U = \begin{bmatrix} \frac{1}{2} I_n & \frac{1}{2} e \\ 0 & 1 \end{bmatrix}$$

and therefore, in the matrix formulation (3.5)

$$Z = UXU^T \qquad \text{where} \qquad X = \begin{bmatrix} xx^T & x \\ x^T & 1 \end{bmatrix}.$$

Since U is invertible, this transformation works the opposite way as well. Observe now that the binary constraints $x_i \in \{-1,1\}$ (or $x_i^2 = 1$) can be formulated as $\mathrm{diag}(xx^T) = e$. A formulation in $\{-1,1\}$ variables of problem (3.5) is

$$\left\{ \begin{array}{ll} \text{maximize} & \langle U^T Q_0 U, X \rangle \\ \text{subject to} & \langle U^T Q_i U, X \rangle \leq a_i, \quad i \in \{1, \ldots, m_I\} \\ & \langle U^T Q_i U, X \rangle = a_i, \quad i \in \{m_I + 1, \ldots, m_I + m_E\} \\ & \mathrm{diag}(X) = e \\ & X = \begin{bmatrix} xx^T & x \\ x^T & 1 \end{bmatrix}. \end{array} \right. \tag{3.6}$$

In *BiqCrunch*, the matrices $U^T Q_i U$ are formed directly from Q_i when reading the data, without explicitly forming the matrix U.

Let us now simplify notation by introducing $a \in \mathbb{R}^{m_I}$, $b \in \mathbb{R}^{m_E+n+1}$ and the two mappings

$$A \colon \mathbb{S}^{n+1} \to \mathbb{R}^{m_I} \quad \text{and} \quad B \colon \mathbb{S}^{n+1} \to \mathbb{R}^{m_E+n+1}$$

to gather all the inequality constraints of problem (3.6) as $A(X) \leq a$ and all the equality constraints as $B(X) = b$ (those defined by $U^T Q_i U$ together with the diagonal constraints). Defining $Q = U^T Q_0 U$, problem (3.1) can be written as the following optimization problem with respect to X positive semidefinite (denoted $X \succeq 0$) and rank-one:

$$\left\{ \begin{array}{ll} \text{maximize} & \langle Q, X \rangle \\ \text{subject to} & A(X) \leq a, B(X) = b \\ & X \succeq 0, \mathrm{rank}(X) = 1. \end{array} \right. \tag{3.7}$$

The basic semidefinite relaxation is then obtained by dropping the rank-one constraint on X:

$$\begin{cases} \text{maximize} & \langle Q, X \rangle \\ \text{subject to} & A(X) \leq a, B(X) = b \\ & X \succeq 0. \end{cases} \tag{3.8}$$

Since problem (3.1) is equivalent to problem (3.7), the optimal value of problem (3.8) provides an upper bound on the optimal value of problem (3.1). As in linear programming, we can further tighten the bound by adding valid inequalities to problem (3.7) before relaxing. There exist many problem-dependent or generic inequality families in our framework; see, e.g., the textbook [69] or early references as [16, 95].

In *BiqCrunch*, we use the triangle inequalities, defined for $1 \leq i < j < k \leq n+1$ by

$$\begin{aligned} X_{ij} + X_{ik} + X_{jk} \geq -1, \qquad & X_{ij} - X_{ik} - X_{jk} \geq -1, \\ -X_{ij} + X_{ik} - X_{jk} \geq -1, \qquad & -X_{ij} - X_{ik} + X_{jk} \geq -1. \end{aligned}$$

These inequalities correspond to the fact that for $x \in \{-1, 1\}^{n+1}$, it is not possible to have exactly one of three products $\{x_i x_j, x_i x_k, x_j x_k\}$ equal to -1, nor is it possible to have all three of the products equal to -1. These inequalities are particularly interesting in our framework, for two reasons. There are $4\binom{n+1}{3}$ of them, which is large but still manageable. Given X, we can evaluate all of them and efficiently find the most violated ones. Moreover, they are known to give good results for semidefinite relaxations in general (see, e.g., [9, 95, 159]).

Ideally we would like to add all the triangle inequalities to problem (3.8) to get the tightest possible bound of this type. However, the cost of doing so is very high, so the relaxation incorporating all the triangle inequalities is rarely used (see, e.g., the numerical comparisons of [159]). In *BiqCrunch* we iteratively identify a subset of useful inequalities, as in [151]. The idea is to select the most promising active inequalities using the current approximate solution. The management of inequalities will be precisely described in Algorithm 2.

For a subset of triangle inequalities $\mathcal{I}$, we let $A_{\mathcal{I}} \colon \mathbb{S}^n \to \mathbb{R}^{|\mathcal{I}|}$ be the corresponding linear function describing the inequalities in this subset. We end up with the following strengthened SDP relaxation of problem (3.1):

$$(\text{SDP}_{\mathcal{I}}) \qquad \begin{cases} \text{maximize} & \langle Q, X \rangle \\ \text{subject to} & A(X) \leq a, B(X) = b \\ & A_{\mathcal{I}}(X) \geq -e \\ & X \succeq 0. \end{cases} \tag{3.9}$$

Note that the maximum is finite and attained if the problem is feasible. This follows from the fact the feasible set is included in the set of correlation matrices ($X \succeq 0$ and $\operatorname{diag}(X) = e$) which is well-known to be compact (see, e.g., [Mal-23, Theorem 1] or more complete studies as [116, 117]).

3.3.2 Adjustable semidefinite bounds

In this section, we define and sketch the main properties of the semidefinite bounds that are used by *BiqCrunch*. We refer to [Mal-11] for the background, the motivation and the theory behind the family of adjustable semidefinite bounds.

Recall that the nonnegative part X_+ of a symmetric matrix X is computable via an eigenvalue decomposition $X = V \operatorname{Diag}(\sigma) V^T$ (with the vector of eigenvalues $\sigma \in \mathbb{R}^n$, and an orthogonal matrix $V \in \mathbb{R}^{n \times n}$) by

$$X_+ = V \operatorname{Diag}(\max\{\sigma, 0\}) V^T.$$

Note that X_+ is the orthogonal projection of X onto the set of positive semidefinite matrices [98]. The nonpositive part X_- is defined similarly. We will need the following property:

$$\langle X, X_+ \rangle = \|X_+\|_F^2 . \tag{3.10}$$

Let $\mathcal{I}$ be a set of inequalities and define $\Omega := \mathbb{R}_+^{m_I} \times \mathbb{R}^{m_E+n+1} \times \mathbb{R}_+^{|\mathcal{I}|}$. For any dual variables $(\lambda, \mu, \nu) \in \Omega$, we define the positive semidefinite matrix

$$X_{\mathcal{I}}(\lambda, \mu, \nu) := [Q - A^*(\lambda) - B^*(\mu) + A_{\mathcal{I}}^*(\nu)]_+ \tag{3.11}$$

where A^*, B^*, and $A_{\mathcal{I}}^*$ are the adjoints of the linear operators A, B, and $A_{\mathcal{I}}$ representing the constraints. For example, for $\lambda \in \mathbb{R}^{m_I}$,

$$A^*(\lambda) = \sum_{i=1}^{m_I} \lambda_i U^T Q_i U.$$

Furthermore, for a parameter $\alpha > 0$, we introduce $F_{\mathcal{I}}^{\alpha}(\lambda, \mu, \nu)$ defined for $(\lambda, \mu, \nu) \in \Omega$ by

$$F_{\mathcal{I}}^{\alpha}(\lambda, \mu, \nu) := \frac{1}{2\alpha} \|X_{\mathcal{I}}(\lambda, \mu, \nu)\|_F^2 + a^T\lambda + b^T\mu + e^T\nu + \frac{\alpha}{2}(n+1)^2. \tag{3.12}$$

Up to a change of sign and a slight change of notation, $F_{\mathcal{I}}^{\alpha}$ corresponds to the function Θ in [Mal-11]. Using the current notation, Theorem 3 of [Mal-11] reads as follows.

Theorem 3.3.1. *For a set of inequalities $\mathcal{I}$, a parameter $\alpha > 0$, and any $(\lambda, \mu, \nu) \in \Omega$, we have that $F_{\mathcal{I}}^{\alpha}(\lambda, \mu, \nu)$ is an upper bound on the optimal value of the semidefinite relaxation* (3.9) *and therefore on the optimal value of the binary quadratic problem* (3.1).

The question now is how to choose parameters to get these bounds $F_{\mathcal{I}}^{\alpha}(\lambda, \mu, \nu)$ as tight as possible. For fixed α and $\mathcal{I}$, the tightest bounds can be obtained by minimizing $F_{\mathcal{I}}^{\alpha}$ over $(\lambda, \mu, \nu) \in \Omega$. The smoothness of $F_{\mathcal{I}}^{\alpha}$ is the key property that allows it to be efficiently minimized. Theorem 2 of [Mal-11] states that the function $F_{\mathcal{I}}^{\alpha}$ is convex and differentiable on Ω, and we have explicit expressions of its partial gradients. In particular, if $X = \frac{1}{\alpha} X_{\mathcal{I}}(\lambda, \mu, \nu)$, then we have

$$\begin{aligned} \nabla_\lambda F_{\mathcal{I}}^{\alpha}(\lambda, \mu, \nu) &= a - A(X), \\ \nabla_\mu F_{\mathcal{I}}^{\alpha}(\lambda, \mu, \nu) &= b - B(X), \\ \nabla_\nu F_{\mathcal{I}}^{\alpha}(\lambda, \mu, \nu) &= e + A_{\mathcal{I}}(X). \end{aligned} \tag{3.13}$$

Thus, we can minimize $F_{\mathcal{I}}^{\alpha}$ using any first-order optimization algorithm that can handle nonnegativity constraints. We could also use the so-called second-order semismooth Newton method [150] since it is possible to show that $\nabla F_{\mathcal{I}}^{\alpha}$ is semismooth using properties of the projection $[\cdot]_+$ (see, e.g., [32]).

For its simplicity and robustness, *BiqCrunch* uses a quasi-Newton method (more specifically, a projected BFGS with Wolfe line-search), which lies between first-order and second-order methods. The properties guaranteeing convergence (see, e.g., [36, Theorem 4.9]) hold here, as $F_{\mathcal{I}}^{\alpha}$ is convex and differentiable with Lipschitz gradient. A simple proof of this basic result is given in [Mal-1].

Lemma 3.3.2. *For given α and $\mathcal{I}$, the gradient $\nabla F_{\mathcal{I}}^{\alpha}$, as an operator from Ω to $\mathbb{R}^{m_I+m_E+n+1}$, is Lipschitz continuous with Lipschitz constant L/α, where L is a constant that depends on the norms of A, B and $A_{\mathcal{I}}$ and their adjoints.*

In addition to minimizing $F_{\mathcal{I}}^{\alpha}$ for fixed α and $\mathcal{I}$, we have two ways to get tighter bounds. Firstly, adding violated triangle inequalities to $\mathcal{I}$ enlarges the space Ω which allows us to further minimize $F_{\mathcal{I}}^{\alpha}$. Secondly, decreasing the parameter α also yields tighter bounds. Theorem 4 of [Mal-11] shows that α controls the tightness of the bound, in that smaller values of α give tighter bounds. In practice, special attention should be paid to decreasing α, since Lemma 3.3.2 relates α to the smoothness of $F_{\mathcal{I}}^{\alpha}$, indicating that when α is small, the gradient can have a sharp behavior, and therefore minimizing $F_{\mathcal{I}}^{\alpha}$ could become ill-conditioned.

The semidefinite bounding procedure presented in the next section efficiently combines these three levers (minimizing $F_{\mathcal{I}}^{\alpha}$, adding inequalities, and decreasing α). Its convergence analysis is studied in Section 3.4.4. Later in Section 3.5.1, practical advice is given on how to adjust key parameters to compute bounds efficiently (with a good ratio of tightness to computing time).

3.4 Algorithmic description

BiqCrunch is a branch-and-bound algorithm, implemented using the branch-and-bound platform BOB [55] which automatically handles the management of subproblems. The BOB platform only requires the following functionalities to be implemented:

1. a bounding procedure (producing an upper bound),
2. a heuristic for generating a feasible solution (producing a lower bound),
3. a method for generating subproblems (branching).

In this section, we provide details about how each of these are implemented in *BiqCrunch*. We consider the binary quadratic subproblem of the current node of the branch-and-bound tree (with a slight abuse of notation, we consider it to be problem (3.1) and use the same notation as before). At iteration k of the bounding procedure, the algorithm brackets the optimal value as

$$\beta_k \leq \text{ optimal value of the binary quadratic problem } \leq F_k, \tag{3.14}$$

where β_k is the best lower bound given by the heuristics (described in Section 3.4.2), and F_k is the upper bound of the bounding procedure (described in Section 3.4.1). Using the fact that we know that the optimal value of problem (3.1) is integer, we have that

$$\text{if } F_k < \beta_k + 1, \text{ then we prune the node of the branch-and-bound tree,} \tag{3.15}$$

since all feasible solutions of the subproblem have an objective value no better than β_k. If this is not the case, we need to explore the branch-and-bound tree further. The different branching strategies that are available in *BiqCrunch* are described in Section 3.4.3. Finally, Section 3.4.4 provides a theoretical analysis of the convergence of our semidefinite bounding procedure.

3.4.1 Semidefinite bounding procedure

We first turn our attention to the bounding procedure used by *BiqCrunch* and its computational aspects. We start by emphasizing that no SDP problem is solved during the bounding procedure, which is a major difference compared to semidefinite-based procedures used by other software packages, such as *Biq Mac* or SCIP-SDP. Our bounding procedure can be very fast to run if the node is easy to prune, but is also able to provide tighter more expensive bounds if necessary.

The key numerical ingredient of the bounding procedure of *BiqCrunch* is the algorithm that minimizes the bounding function (3.12) for a given set of inequalities $\mathcal{I}$ and a given tightness parameter α. We use the projected quasi-Newton software L-BFGS-B [133, 188] (with default parameters, except for `nitermax`, `minNiter`, and `maxNiter`; see Section 3.5.1). The quasi-Newton solver calls a subroutine that computes the value of the bounding function (3.12) and its gradient (3.13) at the current point (λ, μ, ν). This computation boils down to the computation of $X_{\mathcal{I}}(\lambda, \mu, \nu)$ as defined in equation (3.11), which, in turn, reduces to computing the positive eigenvalues and corresponding eigenvectors of the symmetric matrix $[Q - A^*(\lambda) - B^*(\mu) + A_I^*(\nu)]$ for which we use the routine `DSYEVR` of the package `MKL` (or `LAPACK`, if `MKL` is not available). Note that the eigendecomposition computed here is also used in the heuristic for computing feasible solutions (see Section 3.4.2).

Algorithm 1 Semidefinite bounding algorithm of *BiqCrunch*

Data: $\texttt{alpha0} > 0$; $\texttt{tol0} > 0$; $0 < \texttt{scaleAlpha}, \texttt{scaleTol} < 1$
Initialize parameters: $k \leftarrow 1$, $\beta_1 \leftarrow -\infty$, $\varepsilon_1 \leftarrow \texttt{tol0}$, $\alpha_1 \leftarrow \texttt{alpha0}$.
Initialize variables: $\mathcal{I}_0 \leftarrow \emptyset$, $\lambda_0 \leftarrow 0 \in \mathbb{R}^{m_E+n+1}$,
$\mu_0 \leftarrow 0 \in \mathbb{R}^{m_I}$, $F_0 \leftarrow +\infty$.
while $F_k \geq \beta_k + 1$ **do**
    Minimize the function $F^{\alpha_k}_{\mathcal{I}_{k-1}}$ using a quasi-Newton method:
    Starting from $(\lambda_{k-1}, \mu_{k-1}, \nu_{k-1})$, compute $(\lambda_k, \mu_k, \hat{\nu}_k)$ such that (3.16) holds.
    if `withCuts` **then**
        Run inequality update subroutine to get $\mathcal{I}_k$ (and associated multipliers ν_k)
    end if
    Update the upper bound: $F_k \leftarrow F^{\alpha_k}_{\mathcal{I}_{k-1}}(\lambda_k, \mu_k, \hat{\nu}_k) = F^{\alpha_k}_{\mathcal{I}_k}(\lambda_k, \mu_k, \nu_k)$.
    Update the lower bound: run Algorithm 3 to get β, and update $\beta_k \leftarrow \max\{\beta_{k-1}, \beta\}$
    if $\mathrm{Card}(\mathcal{I}_k - \mathcal{I}_{k-1}) \leq \texttt{minCuts}$ **or** α_k has not changed for `maxNAiter` iterations **then**
        $\alpha_{k+1} \leftarrow \max\{\texttt{minAlpha}, \texttt{scaleAlpha} \cdot \alpha_k\}$, $\quad \varepsilon_{k+1} \leftarrow \max\{\texttt{minTol}, \texttt{scaleTol} \cdot \varepsilon_k\}$
    else
        $\alpha_{k+1} \leftarrow \alpha_k$, $\quad \varepsilon_{k+1} \leftarrow \epsilon_k$
    end if
    Run the heuristic in Algorithm 3
end while

The semidefinite bounding procedure of *BiqCrunch* is described in Algorithm 1. Given a set of inequalities $\mathcal{I}_{k-1}$ and tightness parameter α_k, *BiqCrunch* runs a quasi-Newton algorithm on $F^{\alpha_k}_{\mathcal{I}_{k-1}}$: it is warm-started from the previous $(\lambda_{k-1}, \mu_{k-1}, \nu_{k-1})$ and it computes a solution $(\lambda_k, \mu_k, \hat{\nu}_k)$ of ℓ_∞-tolerance ϵ_k:

$$\max\left\{ \left\| \left[a - A\left(X_k\right)\right]_- \right\|_\infty, \ \left\| b - B\left(X_k\right) \right\|_\infty, \ \left\| \left[e + A_{\mathcal{I}}\left(X_k\right)\right]_- \right\|_\infty \right\} < \varepsilon_k, \tag{3.16}$$

where $X_k = \frac{1}{\alpha_k} X_{\mathcal{I}_{k-1}}(\lambda_k, \mu_k, \hat{\nu}_k)$. We stop the bounding procedure when the value of the bound is less than $\beta_k + 1$; in practice, we also stop it when it is likely that a bound less than $\beta_k + 1$ is not attainable within a reasonable amount of time. It is important to note that the bounding procedure may be stopped anytime and will return a valid upper-bound for problem (3.1) (by Theorem 3.3.1).

The remainder of the bounding procedure consists of updating parameters. Algorithm 1 interlaces the decrease of α_k and ϵ_k with the management of the set of inequalities $\mathcal{I}_k$ (by Algorithm 2 that we describe below). The idea is to reduce the tightness parameter α_k when we can no longer make good progress by adding inequalities. We reduce α_k and ϵ_k when the number of violated triangle

inequalities is lower than the threshold `minCuts`, or when they have not been reduced for `maxNAiter` iterations.

Having a lot of enforced inequalities is both good and bad: the more inequalities the better the bound, but on the other hand it increases the number of dual variables that must be optimized over in the quasi-Newton method. *BiqCrunch* updates the set of enforced inequalities by Algorithm 2. First, it gets rid of ϵ_k-inactive triangle inequalities for X_k (i.e., the indices i such that $(\hat{\nu}_k)_i$ is zero and $\mathcal{A}_i(X_k) + 1 > \epsilon_k$). Second, it adds a predefined number of the most violated inequalities to improve the bound as quickly as possible. Once the set $\mathcal{I}_k$ is updated, Algorithm 2 generates ν_k such that

$$X_k = \frac{1}{\alpha_k} X_{\mathcal{I}_k}(\lambda_k, \mu_k, \nu_k)$$

and

$$F_k = F^{\alpha_k}_{\mathcal{I}_k}(\lambda_k, \mu_k, \nu_k) = \frac{\alpha_k}{2} \|X_k\|^2 + a^T\lambda_k + b^T\mu_k + e^T\nu_k + \frac{\alpha_k}{2}(n+1)^2. \tag{3.17}$$

Algorithm 2 Inequality update subroutine of the bounding procedure

1: **Data:** (at iteration k of the bounding procedure) $\mathcal{I}_{k-1}$, $\hat{\nu}_k$, X_k, ϵ_k and X_{k-1}

2: Remove the triangle inequalities that are not ε_k-active:

$$\mathcal{I}^-_{k-1} \leftarrow \{i \in \mathcal{I}_{k-1} : (\hat{\nu}_k)_i = 0 \text{ and } \mathcal{A}_i(X_k) + 1 > \varepsilon_k \}.$$

3: Add the most-violated triangle inequalities:

Let $i_1, \ldots, i_\ell$ be the indices $i \notin \mathcal{I}_{k-1}$ such that $\mathcal{A}_i(X_k) + 1 \leq \texttt{gapCuts} < 0$, ordered such that $\mathcal{A}_{i_1}(X_k) \leq \cdots \leq \mathcal{A}_{i_\ell}(X_k)$. Let

$$\mathcal{I}^+_{k-1} \leftarrow \{i_1, \ldots, i_K\}, \quad \text{where } K = \min\{\ell, \texttt{cuts}\}.$$

4: Update the set of inequalities: $\mathcal{I}_k \leftarrow \left(\mathcal{I}_{k-1} \setminus \mathcal{I}^-_{k-1}\right) \cup \mathcal{I}^+_{k-1}$.

5: Initialize the multipliers for added inequalities to zero:

$$\text{for each } i \in \mathcal{I}_k, \qquad (\nu_k)_i \leftarrow \begin{cases} (\hat{\nu}_k)_i & \text{if } i \in \mathcal{I}_{k-1}, \\ 0 & \text{if } i \notin \mathcal{I}_{k-1}. \end{cases}$$

3.4.2 Heuristics: options and generic semidefinite heuristic

BiqCrunch uses heuristics for generating feasible solutions for problem (3.1). The best feasible solution found provides the lower bound β_k in (3.14). This lower bound is used to prune parts of the branch-and-bound search tree according to the rule (3.15).

BiqCrunch allows the use of three types of heuristics:

1. root-node heuristic (called once before starting the branch-and-bound method),
2. bound heuristic (called each iteration of the bounding procedure),
3. node heuristic (called at the end of bounding procedure).

These three type of heuristics can all be the same or be completely different; they can also depend on the type of problem that is being solved. We include several specific heuristics in the *BiqCrunch* release for Max-Cut, Max-k-cluster, and Maximum-Independent-Set. Users can also specify their own heuristics for their problems of interest as explained in Section 3.5.2. By default, the generic *BiqCrunch* solver uses an empty heuristic for the root-node heuristic and a semidefinite heuristic for both the bound heuristic and the node heuristic.

The generic semidefinite heuristic of *BiqCrunch* is presented in Algorithm 3. It is based on the celebrated Goemans-Williamson heuristic for Max-Cut [84] using randomly generated hyperplans and a factorization of the optimal semidefinite solution computed by the bounding procedure, called $\hat{X}$ here. From a factorization $\hat{X} = WW^T$ with $W \in \mathbb{R}^{(p+1)\times m}$ and a random unit vector $v \in \mathbb{R}^m$, a $\{0,1\}$-vector z is generated from the sign of the inner-product of v with the i^{th} row of W. Then the feasibility of this $\{0,1\}$-vector z for problem (3.1) is tested. The best lower bound is updated if z is both feasible and improves the objective value. Note that, contrary to [84], we do not need to compute a Cholesky factorization of $\hat{X}$, since a factorization is already available from the bounding procedure (which computes an eigendecomposition of the matrix, see Section 3.4.1). Since this process is computationally inexpensive, this is repeated for several random v and different z.

At the end of the semidefinite heuristic, we also add a simple "1-opt" local-search. This local-search routine returns a solution that is locally optimal, in the sense that changing any variable from zero to one, or from one to zero, does not result in a better feasible solution. Note that for some problems (Max-k-Cluster for example), this local-search does not make sense since it cannot produce feasible points; in this case, a parameter `local_search` allows us to disable it.

Algorithm 3 Generic semidefinite heuristic for finding feasible solutions

for many iterations **do**
  Generate a random unit vector v
  for $i = 1, \ldots, n$ **do**
    if z_i is a fixed variable **then** $z_i \leftarrow$ fixed value of z_i
    else $z_i \leftarrow \begin{cases} 0, & \text{if } v^T \text{row}_i(W) < 0 \\ 1, & \text{otherwise} \end{cases}$
    end if
  end for
  Test of improvement:
  if z is feasible for problem (3.1) **and** $z^T S_0 z + s_0^T z > \beta$ **then** $\hat{z} \leftarrow z$ and $\beta \leftarrow z^T S_0 z + s_0^T z$
  end if
end for
if $\hat{z}$ is feasible for problem (3.1) **then**
  while $\hat{z}$ is not locally optimal **do** $\hat{z} \leftarrow$ a strictly better local solution
  end while
end if

3.4.3 Branching strategies

BiqCrunch provides three branching strategies, each of which can be selected by changing the value of the parameter `branchingStrategy` in the input parameter file. The branching rule uses the optimal semidefinite solution given by the semidefinite bounding procedure, as follows. First we

extract the last column $\hat{x}$ of $\hat{X}$ and define $\hat{z} = \frac{1}{2}(\hat{x} + e)$. Then we choose a variable z_i to branch on, using one of the following three strategies:

1. least-fractional: a variable z_i for which $\hat{z}_i$ is furthest from $\frac{1}{2}$ is selected;
2. most-fractional: a variable z_i for which $\hat{z}_i$ is closest to $\frac{1}{2}$ is selected;
3. closest-to-one: a variable z_i for which $\hat{z}_i$ is closest to 1 is selected.

The most-fractional branching strategy is used as the default in *BiqCrunch*.

Branching on variable z_i creates two new subproblems, one where z_i is fixed to 0 and the other where z_i is fixed to 1. These subproblems correspond to nodes in the branch-and-bound search tree. When branching occurs, two nodes are created and added to this search tree. The BOB branch-and-bound platform [55] automatically selects the subproblem having the weakest bound to be the next subproblem to branch on; in the case of a tie, BOB selects the subproblem that is lower in the search tree (i.e., having more variables fixed); if the subproblem is already near the bottom of the search tree (i.e., where all variables are fixed), BOB switches to a depth-first-search traversal of that subtree.

3.4.4 Convergence of the semidefinite bounding procedure

In this section, we study the theoretical convergence of Algorithm 1, the semidefinite bounding procedure of *BiqCrunch*. We assume that it runs an infinite number of iterations: more precisely, we set `maxNIter` $= +\infty$, `minAlpha` $= 0$, `minTol` $= 0$, and we suppose that β is small enough that the loop will not stop. In this case, the two tightness parameters vanish ($\alpha_k \to 0$ and $\varepsilon_k \to 0$). In addition, since the number of sets of inequalities is finite, there exists a set of inequalities $\mathcal{I}$ that is visited an infinite number of times. With this setting, the following theorem shows that the bounds F_k converge (Property *(i)*) and that we know the limit, under a technical boundedness assumption (Property *(ii)*). This result is related to Theorem 4 of [Mal-11] (which is a theoretical convergence of ideal bounds under a strict feasibility assumption) and Theorem 1 of [Mal-7] (which is a similar result for the specific version of *BiqCrunch* for Max-Cut).

Theorem 3.4.1. *Let $(X_k, \lambda_k, \mu_k, \nu_k, \mathcal{I}_k, F_k)_k$ be the sequence of iterates generated by the bounding algorithm. Let $\mathcal{I}$ be a set of inequalities such that there exist infinitely many $\mathcal{I}_k = \mathcal{I}$. Assume that the feasible set of the binary quadratic optimization problem* (3.1) *in nonempty, so that there exists an optimal solution to* (3.1)*. Then the following properties hold:*

(i) The sequence of the bounds $(F_k)_k$ converges; its limit $\bar{F}$ is a bound for the optimal value of (3.9)*. A subsequence of the primal iterates $(X_k)_k$ converges; its limits $\bar{X}$ is feasible for* (3.9)*.*

(ii) If moreover the sequence of dual variables $(\lambda_k, \mu_k, \nu_k)_k$ is bounded, then $\bar{F}$ is the optimal value of (3.9)*, and $\bar{X}$ is an optimal solution of* (3.9)*.*

Proof. The assumption that the feasible set of the initial problem (3.1) is nonempty yields that the feasible sets of its reformulations (3.6) and (3.7) and its relaxations (3.8) and (3.9) are nonempty as well. For this proof, we denote by K the (infinite) set of indexes such that $\mathcal{I}_k = \mathcal{I}$ for $k \in K$.

Let us start the proof of Property *(i)* by noting that, from (3.16), the diagonal entries of X_k for all k are bounded by ϵ_1:

$$\|e - \operatorname{diag}(X_k)\|_\infty \leq \|b - B(X_k)\|_\infty < \epsilon_k \leq \epsilon_1 \quad \text{for all } k. \tag{3.18}$$

Since $X_k \succeq 0$, the determinant of its submatrix with indices $\{i, j\}$ is nonnegative:

$$\det \begin{pmatrix} (X_k)_{ii} & (X_k)_{ij} \\ (X_k)_{ij} & (X_k)_{jj} \end{pmatrix} = (X_k)_{ii}(X_k)_{jj} - (X_k)_{ij}^2 \geq 0.$$

By (3.18), the diagonal entries $(X_k)_{ii}$ and $(X_k)_{jj}$ lie between $[1 - \epsilon_1, 1 + \epsilon_1]$, and therefore we have $(X_k)_{ij}^2 \leq (1 + \epsilon_1)^2$. The norm of X_k is thus bounded:

$$\|X_k\|_F^2 = \sum (X_k)_{ij}^2 \leq (1 + \epsilon_1)^2 (n + 1)^2 \quad \text{for all } k. \tag{3.19}$$

The boundedness of the subsequence $(X_k)_{k \in K}$ implies that we can further extract a converging subsequence; we denote its limit by $\bar{X}$. The closedness of the set of positive semidefinite matrices yields that $\bar{X} \succeq 0$. Notice also that (3.16) implies $B\left(\bar{X}\right) - b = 0$, $[A\left(\bar{X}\right) - a]_- = 0$, and $[e + A_{\mathcal{I}}\left(\bar{X}\right)]_- = 0$, since $\epsilon_k \to 0$. Thus we have that the limit matrix $\bar{X}$ is feasible for (3.9).

Let us now turn to the other part of Property *(i)*, that $(F_k)_k$ converges to a bound of (3.9). Fix k; we are going to show first that F_{k+1} cannot be significantly larger than F_k. Start by observing in Algorithm 2 that we have $F_{k+1} = F_{\mathcal{I}_k}^{\alpha_{k+1}}(\lambda_{k+1}, \mu_{k+1}, \hat{\nu}_{k+1})$, by definition of ν_k and $\mathcal{I}_k$. This implies that $F_{k+1} \leq F_{\mathcal{I}_k}^{\alpha_{k+1}}(\lambda_k, \mu_k, \nu_k)$, since the quasi-Newton algorithm with Wolfe line-search can only decrease the objective value (see [36, Chap.1]). Using the definition (3.12) of the bounds, we write

$$\begin{aligned} F_{k+1} &\leq \frac{1}{\alpha_{k+1}} \left\| X_{\mathcal{I}_k}(\lambda_k, \mu_k, \nu_k) \right\|^2 / 2 + a^T \lambda_k + b^T \mu_k + e^T \nu_k + \alpha_{k+1}(n+1)^2/2 \\ &= F_{\mathcal{I}_k}^{\alpha_k}(\lambda_k, \mu_k, \nu_k) + \left(\frac{1}{\alpha_{k+1}} - \frac{1}{\alpha_k} \right) \left\| X_{\mathcal{I}_k}(\lambda_k, \mu_k, \nu_k) \right\|^2 / 2 + (\alpha_{k+1} - \alpha_k)(n+1)^2/2 \\ &= F_k + \left(\frac{\alpha_k^2}{\alpha_{k+1}} - \alpha_k \right) \|X_k\|^2 / 2 + (\alpha_{k+1} - \alpha_k)(n+1)^2/2 \end{aligned}$$

If $\alpha_{k+1} = \alpha_k$, this inequality is simply $F_{k+1} \leq F_k$. If $\alpha_{k+1} = \texttt{scaleAlpha} \cdot \alpha_k$, this reads

$$F_{k+1} \leq F_k + \frac{\alpha_k}{2} \frac{(1 - \texttt{scaleAlpha})}{\texttt{scaleAlpha}} \left(\|X_k\|^2 - \texttt{scaleAlpha}(n+1)^2 \right).$$

In both cases, this yields, using again (3.19),

$$F_{k+1} \leq F_k + C\alpha_k \quad \text{with } C = \frac{1}{2} \frac{(1 - \texttt{scaleAlpha})}{\texttt{scaleAlpha}} (n+1)^2 \left((1+\epsilon_1)^2 - \texttt{scaleAlpha} \right) > 0. \tag{3.20}$$

This bound on the growth of F_k enables us to argue, as follows, that the sequence converges. Let us repeat the above bounding for $\ell > k$: let $k_1, \ldots, k_p$ be the p indices $k \leq k_i < k + \ell$ such that $\alpha_{k_i+1} = \texttt{scaleAlpha} \cdot \alpha_{k_i}$; from repeated application of inequality (3.20), and using the fact that $F_{k+1} \leq F_k$ when $\alpha_{k+1} = \alpha_k$, we obtain

$$\begin{aligned} F_{k+\ell} &\leq F_k + C \left(\alpha_{k_1} + \alpha_{k_2} + \cdots + \alpha_{k_p} \right) \\ &= F_k + C \left(\alpha_k + \texttt{scaleAlpha} \cdot \alpha_k + \cdots + \texttt{scaleAlpha}^{p-1} \cdot \alpha_k \right) \\ &\leq F_k + C \left(\frac{1}{1 - \texttt{scaleAlpha}} \right) \alpha_k. \end{aligned}$$

Taking $\ell \to +\infty$ and then $k \to +\infty$ above, we get $\limsup_{k \to +\infty} F_k \leq \liminf_{k \to +\infty} F_k$, hence the sequence $(F_k)_k$ converges; let us call its limit $\bar{F}$.

Recall now that Theorem 3.3.1 implies that F_k, for all $k \in K$, is an upper bound for (3.9) (since $\mathcal{I}_k = \mathcal{I}$ for $k \in K$). Since $\bar{F}$ is obviously also the limit of the subsequence $(F_k)_{k\in K}$, $\bar{F}$ is an upper bound for (3.9) as well. Thus we have Property *(i)*:

$$\langle Q, \bar{X}\rangle \leq \text{the optimal value of (3.9)} \leq \bar{F}. \tag{3.21}$$

We prove now Property *(ii)*. We start by observing that, for a given k, we have by (3.10)

$$\langle Q - A^*(\lambda_k) - B^*(\mu_k) + A^*_{\mathcal{I}}(\nu_k), X_k\rangle = \alpha_k \|X_k\|^2$$

which in turn yields

$$\begin{aligned}\langle Q, X_k\rangle &= \alpha_k \|X_k\|^2 + \langle A^*(\lambda_k), X_k\rangle + \langle B^*(\mu_k), X_k\rangle - \langle A^*_{\mathcal{I}}(\nu_k), X_k\rangle \\ &= \alpha_k \|X_k\|^2 + \lambda_k^T A(X_k) + \mu_k^T B(X_k) - \nu_k^T A_{\mathcal{I}}(X_k).\end{aligned}$$

Combining this equation with (3.17), we get

$$F_k - \langle Q, X_k\rangle = \frac{\alpha_k}{2}((n+1)^2 - \|X_k\|^2) + \lambda_k^T(a - A(X_k)) + \mu_k^T(b - B(X_k)) + \nu_k^T(e + A_{\mathcal{I}}(X_k)). \tag{3.22}$$

Notice that the three inner products in the above equation can be bounded with (3.16) as follows:

$$|\lambda_k^T(a - A(X_k))| \leq \|\lambda_k\|\,\epsilon_k, \quad |\mu_k^T(b - B(X_k))| \leq \|\mu_k\|\,\epsilon_k, \text{ and } |\nu_k^T(e + A_{\mathcal{I}}(X_k))| \leq \|\nu_k\|\,\epsilon_k.$$

We use now the additional assumption that the sequence $(\lambda_k, \mu_k, \nu_k)_k$ is bounded and we conclude that the three terms vanish when $k \to +\infty$. Recall (3.19) which implies that the term $\frac{\alpha_k}{2}((n+1)^2 - \|X_k\|^2)$ also goes to zero when $k \to +\infty$. Therefore, we can pass to the limit in (3.22) when $k \to +\infty$ with $k \in K$ and we get $\bar{F} = \langle Q, \bar{X}\rangle$. Therefore, by equation (3.21), $\bar{F}$ is the optimal value of (3.9) and $\bar{X}$ is optimal. ■

Property *(ii)* of this theorem says that, under a boundedness assumption, the bounding procedure eventually solves the SDP relaxation (3.9) as F_k approximates the optimal value and X_k approximates an optimal solution. Thus this result theoretically supports what we observe in practice: once a "good" set of inequalities is "identified," the algorithm solves the corresponding SDP relaxation. However, the bounding procedure is not meant to be just another SDP solver: it combines fast initial iterations (α_k large for small k) and the ability to gain more and more tightness (α_k small for large k). The bounding procedure is therefore primarily designed to compute efficient bounds inside a branch-and-bound routine; solving the SDP relaxation to optimality is not necessary.

It turns out that the bounding procedure has good observed convergence and returns high-quality bounds within a reasonable amount of time. For instance, a convergence curve on a Max-Cut problem is given in Figure 1.3 in Chapter 1. Such convergence is typical of what we have observed in our numerical experiments, for max-cut and other combinatorial problems. For another illustration, we plot the convergence of our bounding procedure in Figure 3.1 on a k-cluster problem.

The numerical comparison of [Mal-11] for k-cluster between this bounding procedure (with the triangle inequalities disabled) and two standard SDP solvers solving ($\text{SDP}_\emptyset$) shows that this bounding procedure has a better ratio of tightness to computing time. Recall also that SDP bounds are shown in [159, Sec. 4.3] to be superior to linear programming bounds for large-scale k-cluster – which confirms the fact reported in [30] that linear programming approaches are not efficient for large-scale k-cluster.

In this chapter, we do not report further numerical comparisons of our bounding procedure with standard SDP, LP or QP bounds; we refer the computational studies of previous papers (in particular [28, 30, 159, Mal-11] for k-cluster). We focus here on solving problems to optimality. The tightness and the efficiency of our bounds will be illustrated indirectly by the numerical results on exact resolution presented in section 3.6.

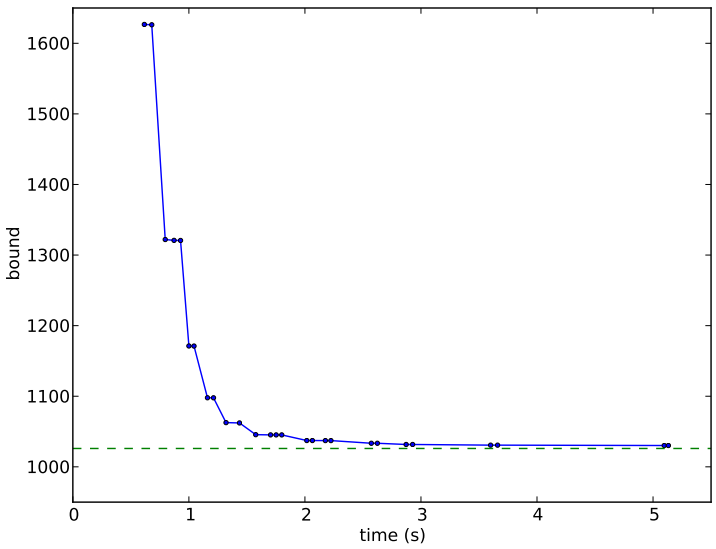


Figure 3.1: Time/bound plot of the bounding procedure on a k-cluster problem, with $n = 160$, edge density 25%, $k = 80$, which has optimal value 1026. Sharp decreases corresponds to decrease of α.

3.5 Improving the performance

As described in the previous section, *BiqCrunch* can theoretically solve any binary quadratic problem. In practice we can improve the performance of *BiqCrunch* for specific problems by:

1. adjusting the parameters of *BiqCrunch*,
2. providing specific heuristics to produce better feasible solutions,
3. strengthening the problem formulation to obtain better upper bounds.

In the *BiqCrunch* package, we have provided different versions of *BiqCrunch*, each of which has been adapted to solve specific problems with tailored heuristics and parameter files. In the rest of this section we discuss each of the above three items.

3.5.1 Parameters

The parameters of *BiqCrunch* are listed in Table 3.1. These parameters are specified in a `biq_crunch.param` file that must be provided when running the solver. *BiqCrunch* provides parameter files with the default parameters, as well as parameter files that have been adjusted for Max-Cut, Max-k-Cluster, and Max-Independent-Set.

For most problems, these parameters do not need to be modified. Nevertheless, some of them are crucial to the performance of *BiqCrunch* for specific instances. The most important parameter is `alpha0` which determines the initial value of α_k in Algorithm 1. For problems that do not require a semidefinite approach to obtain good bounds (for instance when linear programming relaxations are known to be efficient), `alpha0` could be set to a higher value to reduce the computing time when evaluating each node. For more difficult problems (when weak relaxations are not efficient), `alpha0` should be set to a lower value to have tighter initial bounds when evaluating each node.

The gapCuts and cuts parameters are also important since they can be adjusted to find the right trade-off between adding too many or too few cuts. Typically, we want to avoid adding many cuts that are only violated by a small amount. By default *BiqCrunch* only adds at most cuts $= 500$ triangle inequalities each iteration that each have a violation of at most gapCuts $= -0.05$.

We recommend to users who are looking for better performance to adjust the three key parameters (alpha0, gapCuts, and cuts) in the following way: set the "root" parameter to 1 and use the verbose command-line option ("-v 1"), then do tests with different instances of your problem and inspect the output files. A useful rule of thumb is that if the values of the parameters nitermax or cuts are reached when evaluating the root node then the three parameters should be adjusted accordingly.

parameter	**definition / role**	**default value**
alpha0	initial value of α	1e-1
scaleAlpha	scaling value of α	0.5
minAlpha	minimum value of α	5e-5
tol0	initial value of tolerance ε for L-BFGS-B	1e-1
scaleTol	scaling value of tolerance ε for L-BFGS-B	0.95
minTol	minimum value of the tolerance ε for L-BFGS-B	1e-2
nitermax	maximum number of iterations per call of L-BFGS-B	2000
minNiter	minimum number of L-BFGS-B calls	12
maxNiter	maximum number of L-BFGS-B calls	100
maxNAiter	maximum number of L-BFGS-B calls with fixed α	50
withCuts	use the triangle inequalities	1
gapCuts	minimum violation of added cuts (inequalities)	-5e-2
cuts	maximum number of cuts to add per iteration	500
minCuts	minimum number of violated cuts to reduce α and ε	50
scaling	pre-scale the constraints	1
heur_1	use the root-node heuristic	1
heur_2	use the bound heuristic	1
heur_3	use the node heuristic	1
seed	random number generator seed	2016
local_search	use the local search	1
branchingStrategy	0: Branch on least-fractional variable 1: Branch on most-fractional variable 2: Branch on variable that is closest to one	1
root	just evaluate root node (no branch-and-bound)	0
time_limit	limit on computing time (in seconds)	0 (*i.e.*, no time limit)
soln_value_provided	user is providing a known feasible solution value	0
soln_value	the value of a known feasible solution	0

Table 3.1: *BiqCrunch* main parameters

3.5.2 Problem-specific heuristics

The generic heuristic (described in Section 3.4.2) can be substituted with heuristics tailored for specific problems. In the *BiqCrunch* directory, there are several "problems/<PROBLEM>" folders for different optimization problems, and a "problems/user" directory where users can create their own heuristics. For a new heuristic to be called by *BiqCrunch*, one just has to create a directory in the problems/ directory that contains their heur.c file; upon compiling *BiqCrunch*, a biqcrunch executable will be created in the location of the heur.c file.

3.5.3 Strengthening bounds with additional constraints

BiqCrunch does not perform any reformulation or preprocessing of the input problem. The user has complete control over the formulation of their problem. This allows users to try different formulations of the same problem, such as adding redundant constraints to strengthen the semidefinite relaxation and obtain tighter bounds.

Adding linear or quadratic constraints that are redundant for the binary quadratic problem (3.1) does not change its set of optimal solutions, nor its optimal value, but may result in tighter bounds. This is because, with each additional constraint the space of dual multipliers Ω increases, resulting in possibly smaller upper bounds of problem (3.1). In this section, we discuss a set of strengthening constraints that we recommend adding to the formulation of a problem to be solved by *BiqCrunch*.

Suppose problem (3.1) has a linear constraint $s^T z = a$. For instance, the $\sum_{i=1}^n z_i = k$ constraint in the Max-k-Cluster problem is an example of such a linear constraint. The *product constraints* are the valid quadratic constraints generated from $s^T z = a$:

$$z_i s^T z - z_i a = 0, \qquad i = 1, \ldots, n.$$

Introducing quadratic constraints by multiplication is a well-known technique; see [167] for the general approach and [128] for the semidefinite case. It was shown [78] that adding any number of redundant quadratic constraints results in semidefinite bounds that are never better than the one obtained by adding these product constraints (see also [96] for an early study of this question). These product constraints therefore form an optimal set of redundant quadratic constraints.

In practice, adding these constraints to the formulation of problem (3.1) often significantly improves the tightness of the bounds computed by *BiqCrunch* and reduces the overall computing time. As an illustration, we consider solving a problem with $n = 20$ binary variables, a random quadratic objective function, and the cardinality constraint $\sum_{i=1}^n z_i = 10$. First we solve the problem without the product constraints.

```
$ tools/lp2bc.py randprob.lp > randprob.bc
$ problems/generic/biqcrunch randprob.bc biq_crunch.param
Output file: randprob.bc.output
Input file:  randprob.bc
Parameter file: biq_crunch.param
Node 0 Feasible solution 30
Node 1 Feasible solution 95
Node 1 Feasible solution 108
Node 2 Feasible solution 109
Nodes = 27
Root node bound = 113.54
Maximum value = 109
Solution = { 2 3 4 5 8 11 12 13 14 19 }
CPU time = 0.2050 s
```

Next we solve the same problem after having added the product constraints.

```
$ tools/lp2bc.py randprob_prod.lp > randprob_prod.bc
$ problems/generic/biqcrunch randprob_prod.bc biq_crunch.param
Output file: randprob_prod.bc.output
Input file:  randprob_prod.bc
Parameter file: biq_crunch.param
```

```
Node 0 Feasible solution 30
Node 1 Feasible solution 109
Nodes = 1
Root node bound = 109.96
Maximum value = 109
Solution = { 2 3 4 5 8 11 12 13 14 19 }
CPU time = 0.0169 s
```

We notice that the root node bound is much tighter with the product constraints. In this case, the root node bound was tight enough to be able to solve the problem without branching. On the other hand, without the product constraints, 27 nodes of the branch-and-bound search tree are visited before solving the problem. Including such product constraints often significantly improves the performance of *BiqCrunch*.

3.6 Numerical illustrations

We have conducted extensive computational tests on classical NP-hard combinatorial problems. Results on Max-Cut and Max-k-Cluster are available in [Mal-7] and [Mal-4], respectively, and the *BiqCrunch* website reports the latest results available on other problems. These computational results provide strong evidence that *BiqCrunch* is among the best solvers for solving to optimality combinatorial optimization problems that can be formulated using quadratic terms.

To give an idea of the performance of *BiqCrunch*, we provide in this section parts of the numerical experiments of [Mal-7] and [Mal-4], on the comparisons against the best existing methods of the two problems Max-Cut and Max-k-Cluster. We invite the interested reader to visit the BiqCrunch website:

`http://lipn.univ-paris13.fr/BiqCrunch`

to access the entire dataset of problems and the full numerical results of our tests. We will continue to post the complementary results on the website in our further developments.

3.6.1 Illustrations for Max-Cut, comparisons with Biq Mac

There exist efficient methods for solving Max-Cut and binary quadratic problems, such as the ones of [145] and [29]. However, the numerical tests of [151] show that the results of the other methods are dominated by the method called "Biq Mac" presented in this paper. Thus we only compare *BiqCrunch* to this method.

We have compiled and run both *BiqCrunch* and the Biq Mac code (kindly provided by the authors) on the same machine (Dell T-7500, using a single core, with 4 GB of memory, running a Linux OS), and have used the same libraries (i.e., MKL) and compilation flags for both codes. We furthermore consider two versions of the codes depending on the branching strategies "least-fractional" or "most-fractional", as described in section 3.4.3. For the sake of simplicity, we use the same terminology (R2) and (R3) as in [151]. We use 328 test problems in the Biq Mac Library; we refer to [151, Section 6] for a description of the data set.

We have run 328 test-problems for almost 1600 hours of computing time. We observed that 241 out of the 328 problems are solved strictly faster by using our solver, which is around 75% of the test-problems. When considering only the “hard problems” (those that Biq Mac does not solve at the root node), this percentage increases to 85% (226 out of 269). We also report aggregated results

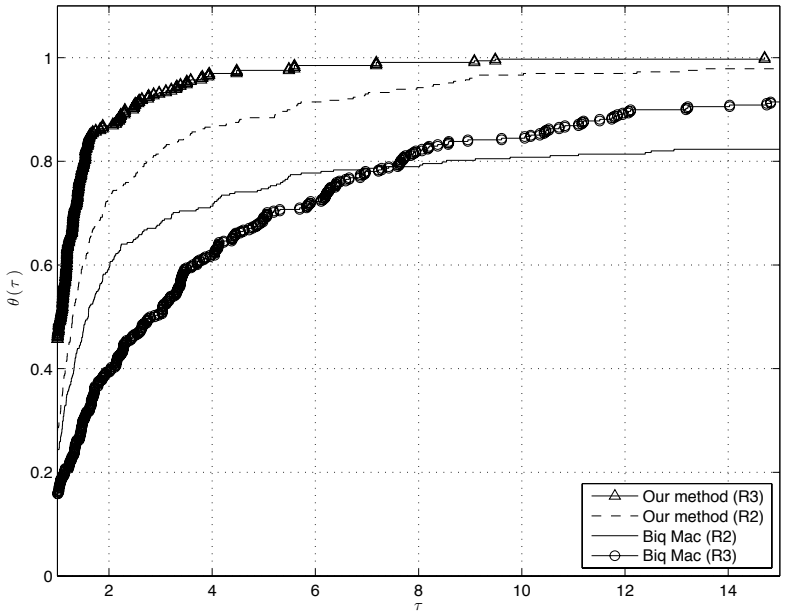


Figure 3.2: Performance profile curves for the two versions of *BiqCrunch* and two versions of Biq Mac on the 328 max-cut problems of the Biq Mac Library. The higher the better!

on Figure 3.2: we use a performance profile[1] (see [70]) to show that both of our methods dominate both of the Biq Mac methods in terms of speed, and also robustness, since the curves for our methods are constantly above the ones for Biq Mac.

3.6.2 Illustration on Max-k-Cluster, comparisons with QCR

The best existing methods to solve the Max-k-Cluster problem to optimality are

- the semidefinite method of [Mal-13], which is a precursor of the method presented here using the same branch-and-bound structure but using basic semidefinite bounds (without dynamic control of α, ε and without triangle inequalities);
- the quadratic convex reformulation (QCR) of [30] which first convexifies the objective function of the problem and then uses the state-of-the-art IBM/CPLEX mixed-integer convex quadratic solver (we use CPLEX 12.6 for the experiments). Since the best convexifying parameters of the initial problem are obtained by solving the semidefinite relaxation $(\mathrm{SDP}_{\mathcal{I}})$ (we use CSDP [37] for the experiments), the QCR and our method use the same (tight) bound at the root node of the branch-and-bound tree. However, the tightness of the QCR bounds deteriorates when going down in the search tree.

We have compared *BiqCrunch* with these two methods on the same machine (Dell Intel Xeon

[1] For each problem $p \in \mathbb{P}$ the set of problems, we define $t_p^{\min}$ as the minimum time required to solve p over all the solvers. Then, for each solver, we consider the performance profile function $\theta(\tau) = \frac{1}{|\mathbb{P}|}\left|\left\{p \in \mathbb{P} : t_p \leq \tau t_p^{\min}\right\}\right|$, for $\tau \geq 1$, where t_p is the time required for the solver to solve problem p. The function θ is therefore a cumulative distribution function, and $\theta(\tau)$ represents the probability of the solver to solve a problem from $\mathbb{P}$ within a multiple τ of the minimum time required by all solvers considered.

CPU E31270 3.40 GHz, using a single core, with 8GB of memory and running a Linux OS), after having tuned their parameters to reach the best performance. We have used the the most challenging test-problems publicly available; their parameters are the size of the graph n, the value of k, and the graph density d. We have a total of 360 instances with $n = 80, 100, 120, 140, 160$. The instances with $n = 80, 100$ have already been used in previous articles, such as [28, 30, Mal-13]; we call them the standard instances. We call the instances with $n = 120, 140, 160$ the larger instances. As far as we are aware, it is the first time that numerical results are reported for $n = 140$ and $n = 160$.

Table 3.2 reports the comparison *BiqCrunch* with the two other methods in terms of CPU time and number of nodes of the search tree to reach optimality for the standard problems. Note that the reported times for QCR are the ones of CPLEX 12.6 (single threaded). The dual variables needed to convexify the problem are obtained by CSDP [37]; but the computing times of CSDP are not considered in the reported CPU time as they are usually insignificant (about 0.4 seconds for $n = 80$ and 0.9 for $n = 100$). Table 3.2 shows that *BiqCrunch* clearly outperforms the two other methods in terms of time spent to solve to optimality and number of nodes in the branch-and-bound tree. Regarding memory issues, the two first methods use a small amount of memory (less than 4 MB) whereas the third featuring CPLEX uses up to 6 GB. In view of this first experiment comparing the best methods on the standard instances, we now focus only on our method to solve larger problems.

			(b) *BiqCrunch*		(c) [Mal-13]		(d) [30]	
n	k	$d(\%)$	nodes	time	nodes	time	nodes	time
80	20	25	3.4	3.2	11650	94.5	170658	39.9
		50	7.4	6.1	41857	323.1	536648	125.0
		75	13.8	9.5	102948	1002.4	1827452	407.4
	40	25	1.4	1.1	1544	13.1	26597	8.6
		50	1.0	0.8	2806	24.6	34148	9.6
		75	6.6	8.0	19789	195.7	231620	55.2
	60	25	1.0	1.1	148	1.3	946	0.6
		50	1.0	0.8	302	2.7	5128	3.1
		75	1.0	0.7	1123	11.2	5754	3.4
mean			4.1	3.5	20241	185.4	315439	72.5
100	25	25	20.6	31.3	127901	2207.1	5680415	1882.8
		50	35.0	41.6	303648	5543.0	19164583	6684.9
		75	30.6	32.3	1180710	19661.2	44336562	14275.8
	50	25	3.4	5.3	9328	164.9	415340	153.6
		50	25.4	46.1	211308	3923.5	5156390	2182.2
		75	1.4	2.1	27099	514.2	514822	203.1
	75	25	1.0	1.9	455	8.0	10261	5.1
		50	1.8	3.8	2018	39.7	108962	37.3
		75	1.0	1.6	1958	38.7	14956	6.5
mean			13.4	18.5	207158	3566.7	8378032	2825.7

Table 3.2: For the standard k-cluster problems: for the three best existing methods, we compare the averaged number of nodes and CPU time (s) for each triple (n, k, d).

For the second experiment, we consider the larger instances and report in Table 3.3 the average number of nodes and time required for our solver to solve each set of five problems. We emphasize that our solver does not need to visit a lot of nodes in the branch-and-bound search tree; for example, we have found that 55% of the problems with size $n \leq 120$ are solved at the root of the branch-and-

			(b) rudy instances		(c) other instances	
n	k	$d(\%)$	nodes	time (s)	nodes	time (s)
120	30	25	119.4	315.8	16.6	36.0
		50	194.2	425.1	39.4	83.1
		75	422.2	889.8	62.2	119.5
	60	25	59.8	198.5	12.2	28.3
		50	85.8	263.2	27.4	69.4
		75	43.0	143.0	41.8	93.0
	90	25	1.8	6.8	1.0	2.9
		50	22.2	96.9	1.0	2.7
		75	1.0	3.0	1.0	2.6
140	35	25	366.2	1165.8	131.0	445.1
		50	1063.4	2888.6	383.0	964.7
		75	1558.6	4079.5	485.0	1279.8
	70	25	134.2	543.0	54.2	216.5
		50	780.6	3035.3	298.6	1133.0
		75	52.2	202.9	155.8	571.7
	105	25	2.6	14.1	1.4	7.5
		50	11.0	61.5	7.4	39.8
		75	6.6	34.8	3.0	19.1
160	40	25	744.6	2856.4	235.4	1023.6
		50	11325.4	37565.2	858.6	3280.1
		75	8050.6	26302.6	1132.2	3997.8
	80	25	395.4	1835.2	73.4	401.6
		50	993.4	4654.5	479.8	1908.6
		75	3829.0	18653.9	1425.0	6288.9
	120	25	31.4	219.8	1.4	9.5
		50	17.4	143.4	2.2	16.7
		75	9.8	82.2	4.2	31.7

Table 3.3: For the larger Max-k-Cluster problems: the averaged number of nodes and CPU time for each triple (n, k, d). The first column concerns with the instances generated by the graph generator `rudy`, and the second column with the random instances generated by Amélie Lambert and available online on her webpage. Data sets and entire results are available online at `http://lipn.univ-paris13.fr/BiqCrunch/results`.

bound tree. Our algorithm is also able to solve unstructured k-cluster problems of sizes $n = 140$ and $n = 160$, for which, as far as we are aware, no numerical results have been reported in the literature.

Let us point out the significant differences in the performance of the algorithm for problems of the same size. In particular, we notice that problems are harder when k is small, and that, in this case, they are harder again when d increases from 25% to 75%, which is due to the presence of many near-optimal k-clusters for larger density graphs (consequently, even if the bound is tight, it is hard to prune nodes in the search tree since the evaluations of many nodes are almost the same).

The bottomline is that $n = 160$ is the largest size of unstructured k-cluster problems to be solved to optimality within a reasonable amount of time on a single-threaded machine. This highlights the need of a multi-threaded version of ***BiqCrunch***, which is the first point in the perspectives of section 7.1.

3.7 Conclusion

In this chapter, we have introduced *BiqCrunch*, an exact solver for general binary quadratic problems. The main feature of *BiqCrunch* is its ability to dynamically set the tightness of its bounding procedure (node by node), using adjustable semidefinite bounds. The bounding procedure automatically adjusts from cheap/poor bounds to expensive/good bounds as needed.

Since *BiqCrunch* uses high-quality bounds, the number of nodes visited throughout the branch-and-bound process is relatively small. Thus, *BiqCrunch* can perform well on problems which are difficult to solve by methods based on linear bounds. *BiqCrunch* complements other exact methods by expanding on the types of problems that we can now efficiently solve. *BiqCrunch* also complements heuristic methods by providing tight bounds that give an accurate measure of the suboptimality of the solutions generated by such methods. *BiqCrunch* can also benefit from high-quality heuristic solutions since having such solutions can further reduce the number branch-and-bound nodes visited.

The source code for *BiqCrunch* is now publicly available. We hope it is a valuable resource to those interested in solving binary quadratic problems. With feedback from the community, we look forward to continuing to improve the code and expanding the range of problems that can be efficiently solved by *BiqCrunch*.

Chapter 4

Nonsmooth optimization with uncontrolled inexact information

This chapter is about nonsmooth optimization algorithms, a research topic inherited from my PhD advisor Claude Lemaréchal. More precisely, this chapter corresponds to the article [Mal-2] on the joint work with Welington de Oliveira (now at Universidade do Estado do Rio de Janeiro) and Sofia Zaourar (now with Xerox Research). Note that this work also impacted the main work of the Sofia's PhD thesis [186] on a nonsmooth approach to a classical technique of operation research, the Benders decomposition.

4.1 Introduction: context, problem, and contributions

4.1.1 Nonsmooth minimization with an (inexact) oracle

We consider nonsmooth optimization problems of the form

$$f_* := \inf_{x \in X} \; f(x)\,, \tag{4.1}$$

with a convex function $f\colon \mathbb{R}^n \to \mathbb{R}$; and a (nonempty) polyhedral set $X \subset \mathbb{R}^n$, and we assume that the infimum is finite ($f_* > -\infty$). Typically, the nonsmoothness of f comes after a maximization, i.e. when f itself is the result of an inner optimization problem

$$f(x) = \sup_{u \in U} \; h(u, x) \tag{4.2}$$

where $h(u, \cdot)$ are convex for each u lying in a set U. Such nonsmooth objective functions appear in Lagrangian relaxation (see e.g. [119]), in stochastic optimization with recourse (see e.g. [166]), or in Benders decomposition (see e.g. [83]).

For a fixed accuracy $\eta \geq 0$, a so-called (lower) η-oracle of f provides, for a point $x \in X$ as an input, an approximate value and an approximate linearization

$$\left\{ \begin{array}{lll} f_x \in \mathbb{R} & \text{such that} & f(x) - \eta \leq f_x \leq f(x), \\ g_x \in \mathbb{R}^n & \text{such that} & f_x + g_x^\top (\cdot - x) \leq f(\cdot). \end{array} \right. \tag{4.3}$$

If the oracle error is null ($\eta = 0$), the oracle returns the exact value $f_x = f(x)$ and a subgradient $g_x \in \partial f(x)$. For some problems, as in large-scale stochastic optimization or in combinatiorial optimization, computing exact information on f is expensive, or even out-of-reach, whereas computing

some inexact information ($\eta > 0$) is still possible. For example, when f is given by (4.2), any $\bar{u} \in U$ gives an inexact value and an approximate linearization of f at a given $x \in X$. Indeed, the convexity of $h(u, \cdot)$ yields

$$h(\bar{u}, x) + g^\top (z - x) \leq h(\bar{u}, z) \leq f(z), \quad \text{for any } g \in \partial_x h(\bar{u}, x).$$

So we have inexact information on f by taking

$$f_x = h(\bar{u}, x) \quad \text{and} \quad g_x = g \in \partial_x h(\bar{u}, x). \tag{4.4}$$

In this case, an η-oracle maximizes $h(\cdot, x)$ over U up to the tolerance η, i.e., computes $\bar{u} \in U$ satisfying $f(x) - \eta \leq h(\bar{u}, x) \leq f(x)$ so that (4.4) gives the η-information (4.3).

Among the nonsmooth optimization methods to solve problems (4.1) with f known by an oracle (4.3), are the bundle-type methods: the Kelley method [102, 111], proximal bundle methods [102], level bundle methods [120], generalized bundle methods [80], and doubly stabilized bundle methods [62]. Initially devellopped for exact oracles ($\eta = 0$), these methods have been extended to handle inexact oracles ($\eta > 0$) and to solve (4.1) up to an accuracy of η. Complete convergence analysis of these methods exists; roughly speaking, under some assumptions, the iterates x_k are an η-minimizing sequence

$$f_* \leq \liminf f(x_k) \leq f_* + \eta. \tag{4.5}$$

We refer to [99] and [169] for first articles, [185] for an inexact version of the Kelley method, [77] for an inexact level method, [113] and [60] for inexact proximal bundle methods, and [59] for inexact level methods with vanishing errors.

4.1.2 Inexact oracle... and more

For some optimization problems as above with an η-oracle, there is in fact additional uncontrolled information on f, which is already available or cheap to get.

A typical example is in combinatorial optimization when f has the form (4.2), with a discrete set U and with a Lagrangian function h (see e.g. [119]). In this case, exact or approximate resolution schemes produce "good" feasible points $\bar{u} \in U$, that give, in turn, linearizations of f by (4.4) – but with uncontrolled accuracy, so that this cannot be used for an oracle with fixed accuracy η. For instance, when (4.2) is solved by a branch-and-bound algorithm, feasible solutions are generated during the exploration of the branch-and-bound tree, but only the final one, the optimal solution, is used by the oracle to generate (4.3). The (uncontrolled) information (4.4) produced by the intermediate feasible solutions is not used, whereas it is available for free and possibly fine (since nearly optimal solutions are usually obtained soon in the branch-and-bound process). It is the same situation when we have cheap heuristics computing solutions that are "good" in practice (sometimes with probabilistic guarantees) but without the (deterministic) guarantee required for an η-oracle. We will consider in section 4.4 an energy optimization problem with such an efficient specific heuristic; other examples include p-median problems [25] and unit-commitment problems (see e.g. the recent review [170]).

Another type of example of cheap uncontrolled information appears in two-stage stochastic linear problems (see e.g. [166], and applications to energy problems in [185] and [61]). In this case, the function has a form (4.2) with separable terms corresponding to linear maximization subproblems

$$f(x) = c^\top x + \sum_{i=1}^{N} \pi_i f_i(x) \qquad \text{with} \quad f_i(x) = \sup_{W^\top u \leq q} (h_i - Tx)^\top u, \tag{4.6}$$

for given N, π_i, h_i, T, W and q (details to come in section 4.4). Computing exact information on f requires to solve the N linear optimization subproblems, which is costly when N is large. Solving only a fraction of these subproblems (say 10%) still gives inexact uncontrolled information on f. Indeed if we compute $\bar{u}_i$ an optimal solution giving $f_i(x)$, then we can also use it to under-approximate other terms $f_j(x)$ (since the feasible sets are the same, we have $(h_j - Tx)^\top \bar{u}_i \leq f_j(x)$). Thus, for a given fraction of solved problems, we have an inexact linearization but with an unknown accuracy.

We formalize the situation where we can compute controlled information together with some uncontrolled inexact information by assuming that we have

$$\begin{array}{c}\text{an oracle with accuracy bounded by } \eta \geq 0, \text{ and} \\ \text{a "cutting-plane generator" adding linearizations with uncontrolled accuracy.}\end{array} \tag{4.7}$$

This abstract cutting-plane generator should be seen as an external module, having the previous bundle of linearizaions as an input, and adding other linearizations without calling the η-oracle. There is no other requirement on the generator: it can use information already available, call heuristics, or even run optimization algorithms. For example, in our numerical experiments, the cutting-plane generators will add inexact (uncontrolled) linearizations produced during a fixed number of iterations of a standard bundle method using heuristics.

Note that the situation (4.7) does not fit in the context of "on-demand accuracy oracles" of [59] where the oracle both requires and provides more information. Note also that the cutting-plane generator is different from the multi-cuts techniques used to accelerate cutting-plane methods in operation research (see e.g. [65] in "column generation", [129] for the Benders decomposition of mixed-integer programming, and [160] in stochastic programming). Contrary to our cutting-plane generator, these techniques usually add several "controlled" cuts. In this context, our approach can be seen as an uncontrolled multi-cut technique.

In the two situations mentioned in this section (Lagrangian relaxations of combinatorial optimization problems, and decompositions of stochastic optimization problems), obtaining uncontrolled bundle information and calling the cutting-plane generator are often of neglectable computational cost compared to the cost of calling the (controlled) oracle. A wise practitioner can therefore be tempted to use the uncontrolled bundle information inside of his bundle method. The goal of this chapter is to serve as an incentive to follow this meaningful practical intuition, as it establishes that incorporating uncontrolled bundle information can help in practice and is also consistent in theory.

4.1.3 Using uncontrolled linearizations in bundle methods

Assume that we are at iteration k of a bundle method solving (4.1), and that we have a family of linearizations

$$\bar{f}_i(\cdot) := f_{x_i} + g_{x_i}^\top(\cdot - x_i) \quad \big(\leq f(\cdot) \big) \tag{4.8}$$

associated to points $\{x_i\} \subset X$. In this chapter, we consider that some of these linearizations (indexed by J_k^η) were given by the oracle, (so they are inexact up to the oracle error $f(x_i) - \eta \leq f_{x_i} \leq f(x_i)$ for all $i \in J_k^\eta$), and that the others (indexed by J_k^{u}) were created by the cutting-plane generator (so we do not known and do not control their inexactness). Bundle methods use available linearizations to create the so-called cutting-plane model of f, which is

$$\check{f}_k(\cdot) := \max_{i \in J_k^\eta \cup J_k^{\mathrm{u}}} \bar{f}_i(\cdot) \quad (\leq f(\cdot)). \tag{4.9}$$

This model is used to compute the next iterate x_{k+1} by solving a convex quadratic programming problem. In proximal bundle algorithms (see e.g. [102]), x_{k+1} is the proximal point of $\check{f}_k$ given

a "prox-parameter" $t_k > 0$ and the "stability center" $\hat{x}_k$; the quadratic optimization problem is the following:

$$\min_{x\in X} \check{f}_k(x) + \frac{1}{2t_k}\|x-\hat{x}_k\|^2 \quad \Longleftrightarrow \quad \left\{ \begin{array}{ll} \min_{x,r} & r + \frac{1}{2t_k}\|x-\hat{x}_k\|^2 \\ \text{s.t.} & \bar{f}_i(x) \le r, \quad \forall\, i \in J_k^\eta \cup J_k^{\mathrm{u}} \\ & x \in X,\ r \in \mathbb{R}\,. \end{array} \right. \tag{4.10}$$

In level bundle algorithms (see e.g. [120]), x_{k+1} is the projection of the current stability center $\hat{x}_k$ onto the level set of "level parameter" f_k^{lev}

$$\mathbb{X}_k := \left\{ x \in X :\ \check{f}_k(x) \le f_k^{\mathrm{lev}} \right\} = \left\{ x \in X :\ \bar{f}_i(x) \le f_k^{\mathrm{lev}} \ \text{ for all } i \in J_k^\eta \cup J_k^{\mathrm{u}} \right\}; \tag{4.11}$$

the quadratic optimization problem is the following:

$$\min_{x\in\mathbb{X}_k} \frac{1}{2}\|x-\hat{x}_k\|^2 \quad \Longleftrightarrow \quad \left\{ \begin{array}{ll} \min_{x} & \frac{1}{2}\|x-\hat{x}_k\|^2 \\ \text{s.t.} & \bar{f}_i(x) \le f_k^{\mathrm{lev}} \quad \forall\, i \in J_k^\eta \cup J_k^{\mathrm{u}} \\ & x \in X. \end{array} \right. \tag{4.12}$$

In both cases, it is clear that using more information gives a more precise model, so would possibly lead to computation of better iterates since the model (4.9) using all the information (max on both J_k^η and J_k^{u}) is obviously always above the model of f that would restrict the max to J_k^η only. Admittedly, in practice using the complete model (4.9) rather than ignoring uncontrolled information makes quadratic programming problems (4.10) and (4.12) larger and then more difficult to solve. This is partly compensated by the ever-growing performance of (specific or even general-purpose) linear-quadratic programming solvers. Anyway, this drawback does not really hold in the case of expensive oracles – which is the situation we consider in this chapter. Thus, there is a clear practical interest to consider as much information as possible when solving (4.1) with bundle methods: richer information can accelerate numerical methods at a neglectable cost, so that the overall computing time is lower than using only the controlled information. This will be illustrated in section 4.4.

There is nevertheless a theoretical argument against using the uncontrolled information in the model. Up to our understanding, the convergence results of bundle methods do not extend in a straightforward way for handling general models (4.9). Standard proofs of convergence use indeed that iterates are computed using a cutting-plane model with "controlled" linearizations, produced by an oracle with bounded or vanishing accuracy ($\eta \to 0$); see e.g. [77], [113], and [59]. In our situation, the call of the cutting-plane generator makes us lose control on the construction of the cutting-plane model and therefore on the next iterate.

The only analysis which is generic enough to cover uncontrolled linearizations is the recent article [60] on the convergence of various forms of proximal bundle methods. So we start with considering, in section 4.2, a proximal bundle algorithm using the cutting-plane generator which is a trivial extension of the standard inexact proximal method of [113] and whose analysis is an instantiation of the generic analysis of [60]. Numerical experiments show that this proximal algorithm does accelerate the convergence with the help of uncontrolled linearizations added by a cheap cutting-plane generator. However this algorithm may not fully capture the uncontrolled information: the prox-parameter t_k in (4.10) ties the next iterate to the stability center $\hat{x}_k$, so that the step can be small even when the model is reasonably rich due to the added uncontrolled linearizations. This feature is inherent to proximal algorithms.

In contrast, level bundle method would better benefit from additional uncontrolled inexact information: richer cutting-plane models would tend to generate useful lower bounds, and then the level set $\mathbb{X}_k$ would better approximate the solution set, and the next iterate would better approximate a

solution. However, to our knowledge, there is no level bundle method which we could build on to introduce uncontrolled information: the level counterpart of [113] able to deal with bounded accuracy η-oracle in the general case has not been developed yet, because of the difficulty of setting up a noise attenuation without direct control on the step (operated by t_k in proximal methods). In particular, our situation does not fit in the recent analysis of [59] that features oracles with varying accuracy but controlling the error (and driving it to zero), nor in [173] that assumes the oracle to have uniformly bounded errors on a compact feasible set X.

4.1.4 Contributions, structure, and notation

This chapter presents two inexact bundle algorithms (proximal and level) incorporating (already available or cheap to compute) uncontrolled bundle information. We formalize the additional information as produced by an external module (the cutting-plane generator of (4.7)) producing inexact linearizations without known or bounded accuracy. In section 4.2, we consider a proximal bundle algorithm using this cutting-plane generator, which is simple (in the sense that is a trivial extension of the standard inexact proximal method of [113]) and has a simple analysis (in the sense that it is an instantiation of the generic analysis of proximal methods of [60]). We introduce in section 4.3 a new inexact level bundle algorithm using the cutting-plane generator which is an extension of the limited-memory proximal level algorithm [41] with an implicit noise attenuation step. This is the first level algorithm able to handle inexact oracles without assuming compactness of the feasible set X or a vanishing error; this is the main technical contribution of this chapter. Finally, in section 4.4, we present and discuss computational illustrations on stochastic optimization problems coming from energy optimization: two-stage linear problems (arising in the planning of hydro-electric power generation, see [185] and [61]) and joint chance-constrained optimization problems (arising in cascaded reservoir management, see e.g. [173]). For these problems, we show that the methods save computational time in using both controlled and uncontrolled information.

Before moving to these developments, we finish this introduction by recalling some notation and terminology of bundle algorithms.

– *Aggregate linearizations.* We will see that the optimality conditions of the quadratic problems (4.10) and (4.12) introduce the "aggregated subgradients" $\hat{g}_k \in \partial \check{f}_k(x_{k+1}) + N_X(x_{k+1})$, which will have a role in the stopping tests. They define in turn the so-called "aggregate linearizations", denoted with the convenient notation "$-k$" borrowed from [60],

$$\bar{f}_{-k}(\cdot) := \check{f}_k(x_{k+1}) + \hat{g}_k^\top(\cdot - x_{k+1})\,. \tag{4.13}$$

It can be proved (see e.g. [59, Prop. 3.2]) that $\bar{f}_{-k}$ is indeed a linearization

$$\bar{f}_{-k}(x) \le \check{f}_k(x) \le f(x) \qquad \text{for all } x \in X\,. \tag{4.14}$$

We also define the "aggregate linearization error" by

$$\hat{e}_k := f_{\hat{x}_k} - \bar{f}_{-k}(\hat{x}_k). \tag{4.15}$$

– *Bundle compression.* The linearizations used in $\check{f}_k$ are possibly numerous and imprecise. It is interesting to be able to work with a limited memory and to somehow extract the useful part from all linearizations. In bundle algorithm terminology, this is called "bundle compression", which is a desirable property in general [102], and thus even more in our context where it would make sense to compress uncontrolled bundle information. In theory we can compress a lot in the algorithms

presented in this chapter: as usual for bundle methods, the current controlled linearization and the aggregate linearization are sufficient to guarantee convergence.

– *Descent test.* The two bundle methods presented in this chapter have a descent step which is the technical point bringing convergence without compactness of X. The stability center $\hat{x}_k$ is updated if the observed decrease is at least a fraction of the predicted decrease

$$v_k = f_{\hat{x}_k} - \check{f}_k(x_{k+1}). \tag{4.16}$$

More specifically, we use the following descent test

$$f_{x_{k+1}} \leq f_{\hat{x}_k} - \kappa_f v_k\,, \quad \text{with} \quad \kappa_f \in (0,1)\,. \tag{4.17}$$

4.2 Proximal bundle method using uncontrolled information

This section explains how the usual inexact bundle method extends easily to deal with uncontrolled bundle information: Algorithm 4 below is a version of the inexact proximal algorithm of [113] using the cutting-plane generator to incorporate uncontrolled bundle information.

At iteration k of this algorithm, optimality conditions of the quadratic proximal subproblem (4.10) can be written with the help of the (simplicial) Lagrange multipliers α_i associated to the constraints $\bar{f}_i(x) \leq r$, as

$$-x + \hat{x}_k - t_k \sum_{i \in J_k^\eta \cup J_k^{\mathrm{u}}} \alpha_i g_i \;\in\; N_X(x).$$

The unique solution x_{k+1} can thus be written as a "subgradient step" along $\hat{g}_k \in \partial \check{f}_k(x_{k+1}) + N_X(x_{k+1})$ with specified stepsize t_k

$$x_{k+1} = \hat{x}_k - t_k \hat{g}_k\,. \tag{4.18}$$

Combined with (4.13), this yields that the aggregate linearization error $\hat{e}_k$ defined in (4.15) and the predicted decrease ν_k of (4.16) are connected by $v_k = e_k + t_k \|\hat{g}_k\|^2$.

Excessive inexactness is handled in a standard way: we employ the "noise attenuation" rule proposed by [113], consisting in increasing sharply t_k whenever $\hat{e}_k$ is overly negative. More precisely, if $\hat{e}_k < -\kappa_{\mathrm{att}} t_k \|\hat{g}_k\|^2$, we set $t_k = 10 t_k$ and solve again (4.10) to obtain another iterate. Otherwise (i.e., $\hat{e}_k \geq -\kappa_{\mathrm{att}} t_k \|\hat{g}_k\|^2$), the algorithm performs like a classical proximal bundle method. We implement this using an extra binary variable `na` indicating noise attenuation. After noise attenuation, t_k does not decrease until a new descent step is performed (see line 20). Though it deals with the coarse information ($J_k^{\mathrm{u}} \neq \emptyset$), the convergence of the algorithm still fits into the generic bundle scheme analysis of [60].

Theorem 4.2.1 (Convergence of inexact proximal bundle)**.** *Set the tolerances to zero in Algorithm 4. Then the sequences testing optimality $\{\hat{e}_k + \hat{g}_k^\top \hat{x}_k\}$ and $\{\hat{g}_k\}$ become "nonpositive", in the sense that there exists a subsequence (indexed by $\mathcal{I}$) such that:*

$$\limsup_{k \in \mathcal{I}} \; \hat{e}_k + \hat{g}_k^\top \hat{x}_k \leq 0 \;\; \textit{and} \;\; \lim_{k \in \mathcal{I}} \|\hat{g}_k\| = 0\,.$$

Furthermore, the iterates $\{\hat{x}_k\}$ generate an η-minimizing sequence, i.e. (4.22) *holds. Thus Algorithm 4 terminates after finitely many steps with an approximate solution if the tolerances tol_g, and tol_e are strictly positive.*

Algorithm 4 Usual inexact proximal bundle method using cutting plane generator

1: Choose $x_1 \in X$, and set $\hat{x}_1 \leftarrow x_1$
2: Choose stopping tolerances, $\mathrm{tol}_e \geq 0$ and $\mathrm{tol}_g \geq 0$
3: Select $\kappa_f, \kappa_{\mathrm{att}} \in (0,1)$ and $t_1 \geq \bar{t} > 0$
4: $(f_{x_1}, g_{x_1}) \leftarrow \eta$-oracle$(x_1)$, set $\hat{g}_1 \leftarrow g_{x_1}$ and $\hat{e}_1 \leftarrow 0$
5: $J_1^\eta \leftarrow \{1\}$, $J_0^{\mathrm{u}} \leftarrow \emptyset$ and $\mathtt{na} \leftarrow 0$
6: **for** $k = 1, 2, \ldots$ **do**
7: $\quad J_k^{\mathrm{u}} \leftarrow$cutting-plane-generator$(\hat{x}_k, J_k^\eta, J_{k-1}^{\mathrm{u}})$ ▷ introduction of uncontrolled linearizations
8: $\quad$ Solve (4.10) to get x_{k+1} and compute $\hat{g}_k$
9: $\quad$ Set $\hat{e}_k \leftarrow v_k - t_k \|\hat{g}_k\|^2$
10: $\quad$ **if** $\hat{e}_k + \hat{g}_k^\top \hat{x}_k \leq \mathrm{tol}_e$ and $\|\hat{g}_k\| \leq \mathrm{tol}_g$ **then** ▷ stopping test
11: $\quad\quad$ **return** $\hat{x}_k$ and $f_{\hat{x}_k}$
12: $\quad$ **end if**
13: $\quad$ **if** $\hat{e}_k < -\kappa_{\mathrm{att}} t_k \|\hat{g}_k\|^2$ **then** ▷ (noise) attenuation
14: $\quad\quad \mathtt{na} \leftarrow 1$, $t_k \leftarrow 10 t_k$, and go back to line 8
15: $\quad$ **end if**
16: $\quad (f_{x_{k+1}}, g_{x_{k+1}}) \leftarrow \eta$-oracle$(x_{k+1})$ ▷ call η-oracle
17: $\quad$ **if** $f_{x_{k+1}} \leq f_{\hat{x}_k} - \kappa_f v_k$ **then**
18: $\quad\quad \hat{x}_{k+1} \leftarrow x_{k+1}$, $\mathtt{na} \leftarrow 0$ and choose $t_{k+1} \geq \bar{t}$ ▷ descent step
19: $\quad$ **else**
20: $\quad\quad \hat{x}_{k+1} \leftarrow \hat{x}_k$ and update t_k: $\begin{cases} t_{k+1} \in [\bar{t}, t_k] & \text{if } \mathtt{na} = 0 \\ t_{k+1} = t_k & \text{if } \mathtt{na} = 1 \end{cases}$
21: $\quad$ **end if**
22: $\quad$ Choose $J_{k+1}^\eta \supset \{k+1, -k\}$ ▷ bundle compression
23: **end for**

Proof. The algorithm fits into the algorithmic pattern 4.2 of [60], and roughly speaking the convergence comes from the use of the η-oracle at descent steps. More specifically, we apply the generic convergence result of Theorems 6.11 and 4.4 of [60]; let us check their assumptions one by one:

- The oracle error is uniformly bounded by η (for iterates x_j with $j \in J_k^\eta$), and thus satisfies (6.8) of [60].
- The cutting-plane model (4.9) satisfies $\check{f}_k \leq f$, i.e., equation (4.10) in [60].
- We have (3.8) of [60] by setting $\ell_k = f_{\hat{x}_k}$.
- The prox-parameter updating rule is of the type (6.14) of [60].
- Equation (6.11) in [60] holds trivially for $f_{\hat{x}_k} - f_{x_{k+1}}$ as effective decrease.
- We have (6.16) of [60] (specifically, with $\alpha_k = 0$ and $\beta_k = \kappa_{\mathrm{att}}$ in there).

Thus Algorithm 4 satisfies all the assumptions (6.15) and (6.16) of [60, Theorem 6.11]. This opens the way to apply [60, Theorem 4.4], which in turn states that having a subsequence $\mathcal{I}$ such that $\limsup_{k \in \mathcal{I}} (\hat{e}_k + \hat{g}_k^\top \hat{x}_k) \leq 0$ and $\lim_{k \in \mathcal{I}} \hat{g}_k = 0$ gives the convergence up to η, which is the desired conclusion. ■

We report numerical illustrations of this algorithm in section 4.3.2. They show that using uncontrolled linearizations within this algorithm leads to less iterations and lower CPU time than using only controlled linearizations. However we see on (4.18) that the algorithm may not exploit completely the added uncontrolled linearizations: x_{k+1} is tied to $\hat{x}_k$ by the explicit prox-parameter t_k, which could prevent the algorithm from making big steps in case of rich cutting-plane model. Such behavior would not appear with level bundle method.

4.3 Level method using uncontrolled information

This section presents a level bundle algorithm dealing with an η-oracle and a cutting-plane generator introducing uncontrolled linearizations, as in (4.7). When disregarding the cutting-plane generator, this algorithm turns out to be the first level method able to deal with inexact η-oracles in general; in this way, it can be seen as the level counterpart of the proximal bundle method of [113]. The algorithm is presented in section 4.3.1, its convergence is stated in section 4.3.2 and analyzed in section 4.3.3. Its numerical behaviour is illustrated in section 4.4.

4.3.1 An inexact proximal-descent level bundle method

To avoid any compactness assumption, we consider a proximal-descent version of level bundle method, inspired from the one of [41]. At iteration k of this algorithm, optimality conditions of the projection problem (4.12) can be written, with the help of the Lagrange multipliers $\alpha_i \geq 0$ associated to the constraints $\bar{f}_i(x) \leq f_k^{\rm lev}$, as

$$-x + \hat{x}_k - \sum_{i \in J_k^\eta \cup J_k^{\rm u}} \alpha_i \, g_i \;\in\; N_X(x).$$

Introducing the "stepsize"

$$\mu_k := \sum_{i \in J_k^\eta \cup J_k^{\rm u}} \alpha_i,$$

we observe that x_{k+1}, the unique solution of the above optimality conditions, can be written as the "subgradient step" along a direction $\hat{g}_k \in \partial \check{f}_k(x_{k+1}) + N_X(x_{k+1})$

$$x_{k+1} = \hat{x}_k - \mu_k \hat{g}_k \qquad \text{such that} \quad \mu_k(\check{f}_k(x_{k+1}) - f_k^{\rm lev}) = 0. \tag{4.19}$$

The (inexact) upper bound is given by the η-oracle at the stability center ($f_k^{\rm up} = f_{\hat{x}_k}$). When $\mu_k > 0$, the predicted decrease (4.16) then corresponds to the level depth

$$v_k = f_k^{\rm up} - f_k^{\rm lev},$$

and the aggregate linearization error is related to it, as

$$\hat{e}_k = v_k - \mu_k \|\hat{g}_k\|^2 . \tag{4.20}$$

To see this, notice from (4.19) that $\mu_k > 0$ ensures that $\check{f}_k(x_{k+1}) = f_k^{\rm lev}$ and, therefore

$$\hat{e}_k = f_{\hat{x}_k} - (\check{f}_k(x_{k+1}) + \hat{g}_k^\top(\hat{x}_k - x_{k+1})) = f_{\hat{x}_k} - f_k^{\rm lev} - \mu_k\|\hat{g}_k\|^2 = v_k - \mu_k\|\hat{g}_k\|^2 .$$

We emphasize that we do not control the stepsize μ_k in level bundle methods, in contrast with proximal bundle methods where we can choose the prox-parameter t_k giving the stepsize. This poses a technical difficulty for handling excessive inexactness within level methods. In Algorithm 4, as in other inexact proximal bundle methods, t_k is increased when the noise is excessively large compared to $\hat{g}_k$ (see line 13 in Algorithm 4); this can not be done directly in an inexact level method. So we propose in Algorithm 5 an *implicit* noise attenuation rule, combined with the level attenuation rule. The idea is simple: we do not allow the depth v_k to decrease if the noise is excessive (see line 21). We will prove in the key proposition 4.3.7 that this simple idea makes μ_k to go to infinity in presence of noise, such that either a new descent step is generated, or the algorithm terminates.

Algorithm 5 New inexact proximal level method using cutting-plane generator

1: Choose $x_1 \in X$, $v_1 > 0$, and set $\hat{x}_1 \leftarrow x_1$
2: Choose $\mathrm{tol}_\Delta \geq 0$, $\mathrm{tol}_e \geq 0$ and $\mathrm{tol}_g \geq 0$
3: Select $\kappa_l, \kappa_f, \kappa_{\mathrm{att}} \in (0,1)$
4: Choose a threshold $\mu_{\mathrm{large}} > 0$
5: $(f_{x_1}, g_{x_1}) \leftarrow \eta$-oracle$(x_1)$, set $\hat{g}_1 \leftarrow g_{x_1}$ and $\hat{e}_1 \leftarrow 0$
6: Set $f_1^{\mathrm{up}} \leftarrow f_{x_1}$, $f_1^{\mathrm{low}} \leftarrow -\infty$, $\Delta_1 \leftarrow +\infty$, $J_1^\eta \leftarrow \{1\}$, $J_0^{\mathrm{u}} \leftarrow \emptyset$
7: **for** $k = 1, 2, \ldots$ **do**
8: $J_k^{\mathrm{u}} \leftarrow$cutting-plane-generator$(\hat{x}_k, J_k^\eta, J_{k-1}^{\mathrm{u}})$ ▷ introduction of uncontrolled linearizations
9: Update $f_k^{\mathrm{lev}} \leftarrow f_k^{\mathrm{up}} - v_k$ and $\mathbb{X}_k \leftarrow \left\{x \in X : \check{f}_k(x) \leq f_k^{\mathrm{lev}}\right\}$
10: **if** $\Delta_k \leq \mathrm{tol}_\Delta$ or ($\hat{e}_k \leq \mathrm{tol}_e$ and $\|\hat{g}_k\| \leq \mathrm{tol}_g$) **then** ▷ stopping test
11: **return** $\hat{x}_k$ and $f_{\hat{x}_k} = f_k^{\mathrm{up}}$
12: **end if**
13: Run a quadratic optimization software on problem (4.12)
14: **if** $\mathbb{X}_k$ is empty **then**
15: $f_k^{\mathrm{low}} \leftarrow f_k^{\mathrm{lev}}$, $\Delta_k \leftarrow f_k^{\mathrm{up}} - f_k^{\mathrm{low}}$, $v_k \leftarrow \min\{\nu_k, \kappa_l \Delta_k\}$ ▷ lower bound
16: Go back to line 9
17: **else**
18: Get x_{k+1} and μ_k, and compute $\hat{g}_k$ using (4.19)
19: $\hat{e}_k \leftarrow v_k - \mu_k \|\hat{g}_k\|^2$
20: **end if**
21: **if** $\mu_k > \mu_{\mathrm{large}}$ and $\hat{e}_k \geq -\kappa_{\mathrm{att}} \mu_k \|\hat{g}_k\|^2$ **then** ▷ (level+noise) attenuation
22: $v_k \leftarrow \frac{v_k}{2}$, and go back to line 9
23: **end if**
24: $(f_{x_{k+1}}, g_{x_{k+1}}) \leftarrow \eta$-oracle$(x_{k+1})$ ▷ call η-oracle
25: **if** $f_{x_{k+1}} \leq f_{\hat{x}_k} - \kappa_f v_k$ **then**
26: $\hat{x}_{k+1} \leftarrow x_{k+1}$, $f_{k+1}^{\mathrm{up}} \leftarrow f_{\hat{x}_{k+1}}$ and $f_{k+1}^{\mathrm{low}} \leftarrow f_k^{\mathrm{low}}$ ▷ descent step
27: $\Delta_{k+1} \leftarrow f_{k+1}^{\mathrm{up}} - f_{k+1}^{\mathrm{low}}$ and $v_{k+1} \leftarrow \min\{v_k, \kappa_l \Delta_{k+1}\}$
28: **else**
29: $\hat{x}_{k+1} \leftarrow \hat{x}_k$, $\Delta_{k+1} \leftarrow \Delta_k$, $v_{k+1} \leftarrow v_k$, $f_{k+1}^{\mathrm{up}} \leftarrow f_k^{\mathrm{up}}$ and $f_{k+1}^{\mathrm{low}} \leftarrow f_k^{\mathrm{low}}$
30: **end if**
31: Choose $J_{k+1}^\eta \supset \{k+1, -k\}$ ▷ bundle compression
32: **end for**

In practice, the projection onto $\mathbb{X}_k$ (problem (4.12)) is solved by a quadratic programming solver (at line 18 of Algorithm 5). If $\mathbb{X}_k$ is nonempty, the solver provides x_{k+1} and μ_k, from which we deduce $\hat{g}_k$ by (4.19). If $\mathbb{X}_k$ is empty, the solver raises a flag of infeasibility and we exploit this information by updating the lower bound for the optimal value f_*: observe indeed that when $\mathbb{X}_k$ is empty, there holds

$$f_k^{\text{lev}} < \check{f}_k(x) \leq f(x) \quad \text{for all } x \in X,$$

so that we can set $f_k^{\text{low}} = f_k^{\text{lev}}$ (see line 15). At each iteration of Algorithm 5, we thus have a lower bound f_k^{low} and an inexact upper bound f_k^{up} such that

$$f_k^{\text{low}} \leq f_* \leq f_k^{\text{up}} + \eta. \tag{4.21}$$

4.3.2 Convergence result

We have the following theorem stating the convergence of Algorithm 5, which is of the same vein as Theorem 4.2.1 for Algorithm 4.

Theorem 4.3.1 (Convergence of inexact proximal level)**.** *Set the tolerances to zero in Algorithm 5. Then the sequences testing optimality* $\{\Delta_k = f_k^{\text{up}} - f_k^{\text{low}}\}$, $\{\hat{e}_k\}$ *and* $\{\hat{g}_k\}$ *become "nonpositive", in the sense that*

- *either the sequence* $\{\Delta_k\}$ *tends to be nonpositive:* $\lim \Delta_k \leq 0$,
- *or there exists a subsequence (indexed by* $\mathcal{I}$*) such that:* $\liminf_{k\in\mathcal{I}} \hat{e}_k \leq 0$ *and* $\lim_{k\in\mathcal{I}} \|\hat{g}_k\| = 0$.

Furthermore, the iterates $\{\hat{x}_k\}$ *generate an* η*-minimizing sequence, i.e.*

$$f_* \leq \liminf f(\hat{x}_k) \leq f_* + \eta. \tag{4.22}$$

Thus Algorithm 5 terminates after finitely many steps with an approximate solution if the tolerances tol_Δ, tol_g, *and* tol_e *are strictly positive.*

The next section is devoted to the proof of this theorem. We will say that the algorithm converges up to η when (4.22) holds. Note that there are two ways to stop the algorithm (see line 10): the usual criterion based on the gap $\Delta_k = f_k^{\text{up}} - f_k^{\text{low}}$

$$\lim \Delta_k \leq 0 \quad \Longrightarrow \text{ convergence up to } \eta \tag{4.23}$$

and a second one inspired from [41] based on the aggregated error and subgradients to deal with unbounded feasible sets. The next two lemmas explain these two stopping tests and their consistency.

Lemma 4.3.2 (Nonpositivity of Δ_k and convergence)**.** *If* $\lim \Delta_k \leq 0$*, then the sequence* $\{\hat{x}_k\}$ *satisfies*

$$f_* - \eta \leq \lim f_{\hat{x}_k} \leq f_* \leq \liminf f(\hat{x}_k) \leq f_* + \eta\,. \tag{4.24}$$

Furthermore, if at some iteration k *we have* $\Delta_k \leq 0$*, then we have in fact*

$$f_* - \eta \leq f_{\hat{x}_k} \leq f_* \leq f(\hat{x}_k) \leq f_* + \eta\,. \tag{4.25}$$

Proof. Note first that the η-oracle properties imply that, for all k,

$$f_* - \eta \leq f(\hat{x}_k) - \eta \leq f_{\hat{x}_k}\,, \tag{4.26}$$

so that $f_k^{\mathrm{up}} = f_{\hat{x}_k}$ satisfies (4.21). We see that $\{f_k^{\mathrm{up}} = f_{\hat{x}_k}\}$ is nonincreasing (line 26), $\{f_k^{\mathrm{low}}\}$ is nondecreasing (line 15), and so $\{\Delta_k\}$ is nonincreasing (line 27). The nonincreasing sequence $\{f_{\hat{x}_k}\}$ is bounded from below thus converges and $\lim f_{\hat{x}_k} \geq f_* - \eta$. Similarly the nondecreasing $\{f_k^{\mathrm{low}}\}$ is bounded from above by f_*, thus it also converges and $\lim f_k^{\mathrm{low}} \leq f_*$. Writing $\lim \Delta_k \leq 0$ as $\lim f_{\hat{x}_k} - \lim f_k^{\mathrm{low}} \leq 0$ we obtain

$$f_* - \eta \leq \lim f_{\hat{x}_k} \leq f_*. \tag{4.27}$$

Now passing to the limit-inf in (4.26) and adding η, we also have

$$f_* \leq \liminf f(\hat{x}_k) \leq \lim f_{\hat{x}_k} + \eta \leq f_* + \eta.$$

Combining this inequalities with (4.27) gives the announced inequalities (4.24).

The argument leading to the second inequality (4.25) is the same as above. For a fixed k, (4.26) and $\Delta_k \leq 0$ give $f_* - \eta \leq f_{\hat{x}_k} \leq f_*$, and adding η to (4.26) yields

$$f_* \leq f(\hat{x}_k) \leq f_{\hat{x}_k} + \eta \leq f_* + \eta\,.$$

Combining the inequalities gives (4.25). ■

Lemma 4.3.3 (Vanishing aggregate errors and convergence)**.** *For the sequences $\{\hat{x}_k\}$, $\{\hat{e}_k\}$ and $\{\hat{g}_k\}$ generated by Algorithm 5, we have, for all $x \in X$,*

$$f(\hat{x}_k) \leq f(x) + \hat{e}_k + \eta - \hat{g}_k^\top (x - \hat{x}_k). \tag{4.28}$$

Assume that $\{\hat{x}_k\}$ is bounded and there exists a subsequence indexed by $\mathcal{I}$ such that

$$\liminf_{k \in \mathcal{I}} \hat{e}_k \leq 0 \quad \textit{and} \quad \lim_{k \in \mathcal{I}} \|\hat{g}_k\| = 0. \tag{4.29}$$

Then the algorithm converges up to η.

Proof. Fix $x \in X$. The inequality (4.28) comes from (4.13) as follows:

$$\begin{aligned}
f(x) &\geq \bar{f}_{-k}(x) \\
&= \check{f}_k(x_{k+1}) + \hat{g}_k^\top (x - x_{k+1}) \\
&= \check{f}_k(x_{k+1}) + \hat{g}_k^\top (\hat{x}_k - x_{k+1}) + \hat{g}_k^\top (x - \hat{x}_k) \\
&= \bar{f}_{-k}(\hat{x}_k) + \hat{g}_k^\top (x - \hat{x}_k) \\
&= f_{\hat{x}_k} - (f_{\hat{x}_k} - \bar{f}_{-k}(\hat{x}_k)) + \hat{g}_k^\top (x - \hat{x}_k) \\
&= f_{\hat{x}_k} - \hat{e}_k + \hat{g}_k^\top (x - \hat{x}_k) \\
&\geq f(\hat{x}_k) - \eta - \hat{e}_k + \hat{g}_k^\top (x - \hat{x}_k).
\end{aligned}$$

We also get the upper bound

$$f_* \leq f(\hat{x}_k) \leq f(x) + \hat{e}_k + \eta + \|\hat{g}_k\|\,\|x - \hat{x}_k\|\,.$$

Passing to the $\liminf$, (4.29) together with the boundedness of $\{\hat{x}_k\}$ yields

$$f_* \leq \liminf_{k \in \mathcal{I}} f(\hat{x}_k) \leq f(x) + \eta.$$

Taking the infimum over $x \in X$ gives (4.22). ■

4.3.3 Convergence proof

To prove Theorem 4.3.1, we adapt the usual rationale of convergence proof of bundle methods, by considering the two cases of infinitely many and finitely many descent steps (line 26). We show that in both cases one of the two stopping tests is active, which guarantees in turn that the algorithm converges up η (by (4.23) and Lemma 4.3.3). The technical challenge is to handle, first, a fixed inexactness in a level method and, second, the uncontrolled cutting-plane model. We note that this proof of convergence differs from the one of [41].

We start with a remark about the level depth v_k. Looking at lines 15, 22 and 27, we see that $\{v_k\}$ is nonincreasing, and that if $v_k \geq 0$ then $\hat{e}_k \geq -\mu_k \|\hat{g}_k\|^2$. We also notice that v_k can be negative only if so is Δ_k, and then (4.23) holds. Therefore, we consider that $v_k \geq 0$ in the remainder of the section.

We will also need the index set $\mathcal{A}$ of the iterations requiring a noise attenuation (line 22). The following lemma studies the situation of infinitely many of such attenuations. The following proposition treats the first case of infinitely many descent steps.

Lemma 4.3.4 (Infinitely many attenuations)**.** *If $\mathcal{A}$ contains infinitely many indices, then* (4.29) *holds with $\mathcal{I} = \mathcal{A}$. If the sequence $\{\hat{x}_k\}$ is furthermore bounded, then the algorithm converges up to η.*

Proof. Recall that $v_k = \hat{e}_k + \mu_k \|\hat{g}_k\|^2$ by (4.20). If $k \in \mathcal{A}$, then we have

$$v_k = \hat{e}_k + \mu_k \|\hat{g}_k\|^2 \geq (1 - \kappa_{\text{att}})\mu_k \|\hat{g}_k\|^2 \geq (1 - \kappa_{\text{att}})\mu_{\text{large}} \|\hat{g}_k\|^2 \geq 0.$$

If the set $\mathcal{A}$ is infinite, then we have $v_k \to 0$, and therefore $\|\hat{g}_k\| \to 0$ by the above inequality. By (4.20), this yields that $\hat{e}_k \to 0$ and then we have (4.29) with $\mathcal{I} = \mathcal{A}$. As a result, if the sequence $\{\hat{x}_k\}$ is bounded, we can invoke Lemma 4.3.3 and get that $\{\hat{x}_k\}$ is η-minimizing. ■

Proposition 4.3.5 (Infinitely many descent steps)**.** *Suppose there are infinitely many descent steps (line 26). Then the algorithm converges up to η.*

Proof. Let us index the descent steps by ℓ. More precisely $k(\ell)$ denotes the ℓ^{th} descent iteration, and $j(\ell) = k(\ell+1) - 1$ the last iteration before the $(\ell+1)^{\text{th}}$. Note that $\hat{x}_{k(\ell)}$ is the ℓ^{th} (different) stability center, and that $\hat{x}_{k(\ell)} = \hat{x}_{j(\ell)}$. The descent test (4.17) gives the inequality

$$f_{x_{k(\ell)}} - f_{x_{k(\ell+1)}} \geq \kappa_f v_{j(\ell)} \geq 0\,.$$

Summing over ℓ we get

$$f_{x_{k(0)}} - \lim_{\ell} f_{x_{k(\ell+1)}} \geq \kappa_f \sum_{\ell=0}^{\infty} v_{j(\ell)}\,.$$

Since $\lim_\ell f_{x_{k(\ell+1)}} \geq f_* - \eta > -\infty$, we get that the series converges and then

$$\lim_{\ell} v_{j(\ell)} = 0. \tag{4.30}$$

By monotonicity of v_k, we thus have $\lim_k v_k = 0$. Let us distinguish now three cases:

(i)$\mathcal{A}$ finite (ii)$\mathcal{A}$ infinite and $\{\hat{x}_k\}_k$ bounded (iii)$\mathcal{A}$ infinite and $\{\hat{x}_k\}_k$ unbounded

In the case (i), for k large enough, we have $v_k = \kappa_l \Delta_k$, and then $\lim_k \Delta_k = 0$. Thus, (4.23) holds and the proof is over. In the case (ii), we can use Lemma 4.3.4 which gives (4.29) and that $\{\hat{x}_k\}$ is η-minimizing. So let us focus on the case (iii), and let us prove by contradiction that $\{\hat{x}_k\}$ is still η-minimizing.

Suppose that there exists $\epsilon > 0$ such that $f(\hat{x}_k) > f_* + \eta + \epsilon$ for all k large enough. This yields that there exists $\tilde{x} \in X$ such that $f(\hat{x}_{k(\ell)}) \geq f(\tilde{x}) + \eta + \epsilon/2$ for all large ℓ. Then (4.28) applied to $k = j(\ell)$ gives

$$\hat{g}_{j(\ell)}^\top (x - \hat{x}_{k(\ell)}) \leq f(x) + \eta - f(\hat{x}_{k(\ell)}) + \hat{e}_{j(\ell)} \quad \text{for all } x \in X\,,$$

which yields

$$\hat{g}_{j(\ell)}^\top (\tilde{x} - \hat{x}_{k(\ell)}) \leq \hat{e}_{j(\ell)} - \epsilon/2\,.$$

Using this inequality and (4.20), we develop

$$\begin{aligned}
\left\|\hat{x}_{k(\ell+1)} - \tilde{x}\right\|^2 &= \left\|\hat{x}_{k(\ell)} - \mu_{j(\ell)}\hat{g}_{j(\ell)} - \tilde{x}\right\|^2 \\
&= \left\|\hat{x}_{k(\ell)} - \tilde{x}\right\|^2 + \left\|\mu_{j(\ell)}\hat{g}_{j(\ell)}\right\|^2 + 2\mu_{j(\ell)}\hat{g}_{j(\ell)}^\top(\tilde{x} - \hat{x}_{k(\ell)}) \\
&= \left\|\hat{x}_{k(\ell)} - \tilde{x}\right\|^2 + \mu_{j(\ell)}[\mu_{j(\ell)}\left\|\hat{g}_{j(\ell)}\right\|^2 + 2\hat{g}_{j(\ell)}^\top(\tilde{x} - \hat{x}_{k(\ell)})] \\
&\leq \left\|\hat{x}_{k(\ell)} - \tilde{x}\right\|^2 + \mu_{j(\ell)}[\mu_{j(\ell)}\left\|\hat{g}_{j(\ell)}\right\|^2 + 2\hat{e}_{j(\ell)} - \epsilon] \\
&\leq \left\|\hat{x}_{k(\ell)} - \tilde{x}\right\|^2 + 2\mu_{j(\ell)}[v_{j(\ell)} - \epsilon/2]\,.
\end{aligned}$$

As $\lim_\ell v_{j(\ell)} = 0$ by (4.30), we have for all ℓ large enough $v_{j(\ell)} \leq \epsilon/2$ and then

$$\left\|\hat{x}_{k(\ell+1)} - \tilde{x}\right\|^2 \leq \left\|\hat{x}_{k(\ell)} - \tilde{x}\right\|^2$$

which contradicts the fact that $\{\hat{x}_k\}$ is unbounded. Hence, (4.22) must hold. ■

We consider now the second case of finitely many descent steps. We start with a lemma stating that null iterates get further away from the last stability center.

Lemma 4.3.6 (After a last descent step)**.** *If $\hat{x}_k = \hat{x}_{k-1} = \hat{x}$, $f_k^{\text{lev}} \leq f_{k-1}^{\text{lev}}$ and $v_k = v_{k-1}$, then we have*

$$\|x_{k+1} - \hat{x}\|^2 \geq \|x_k - \hat{x}\|^2 + \frac{(1-\kappa_f)^2}{\|g_{x_k}\|^2} v_k^2.$$

Proof. The bundle management of line 31 incorporates two pieces in the model $\check{f}_k$: the k-th linearization $\bar{f}_k$ and the aggregate linearization $\bar{f}_{-k}$. Both bring some information, as follows. First, since $\bar{f}_{-(k-1)} \leq \check{f}_k$ and $f_k^{\text{lev}} \leq f_{k-1}^{\text{lev}}$, we have that the level set $\mathbb{X}_k$ is included in the "aggregate level set" $\mathbb{X}_{-(k-1)} := \{x \in X : \bar{f}_{-(k-1)}(x) \leq f_{k-1}^{\text{lev}}\}$, and therefore that $x_{k+1} \in \mathbb{X}_{-(k-1)}$. It can be proved (see e.g. [59, Prop. 3.2]) that the aggregate level-set produces the same iterate that $\mathbb{X}_{k-1}$; in other words,

$$x_k = P_{\mathbb{X}_{k-1}}(\hat{x}) = P_{\mathbb{X}_{-(k-1)}}(\hat{x}) \quad \text{and} \quad (\hat{x} - x_k)^\top (x - x_k) \leq 0 \text{ for all } x \in \mathbb{X}_{-(k-1)}\,. \tag{4.31}$$

Thus, we have $(\hat{x} - x_k)^\top (x_{k+1} - x_k) \leq 0$ and developing $\|x_{k+1} - \hat{x}\|^2 = \|x_{k+1} - x_k + (x_k - \hat{x})\|^2$, the inequality gives

$$\|x_{k+1} - \hat{x}\|^2 \geq \|x_k - \hat{x}\|^2 + \|x_k - x_{k+1}\|^2\,. \tag{4.32}$$

Now since $\bar{f}_k \leq \check{f}_k$ and $x_{k+1} \in \mathbb{X}_k$, we have $f_{x_k} + g_{x_k}^\top (x_{k+1} - x_k) \leq f_k^{\text{lev}}$, which gives

$$f_{x_k} - f_k^{\text{lev}} \leq \|g_{x_k}\|\,\|x_{k+1} - x_k\|\,. \tag{4.33}$$

Iteration k is not a descent iteration: the converse of line 25 reads $f_{x_k} \geq f_{\hat{x}} - \kappa_f v_{k-1}$. Recalling that $f_k^{\text{lev}} = f_{\hat{x}} - v_k$ and $v_k = v_{k-1}$, this yields $f_{x_k} - f_k^{\text{lev}} \geq (1 - \kappa_f) v_k$. Together with (4.33), this gives

$$\|x_{k+1} - x_k\| \geq \frac{(1-\kappa_f)}{\|g_{x_k}\|} v_k.$$

which ends the proof with (4.32). ■

Proposition 4.3.7 (Finitely many descent steps)**.** *Suppose that Algorithm 5 generates only finitely many descent steps. Then the algorithm converges up to η.*

Proof. Let us consider first two easy cases. If $\lim \Delta_k \leq 0$ then (4.23) holds, and the proof is over. If $\mathcal{A}$ has infinitely many indices, we can conclude with Lemma 4.3.4 together with the fact that the sequence $\{\hat{x}_k\}$ is constant for k large enough.

Let us focus on the case where there exists $\bar{\Delta} > 0$ such that $\Delta_k \geq \bar{\Delta}$ for all k, and there is eventually no noise attenuation ($\mathcal{A}$ has finitely many indices). For k large enough, the stability center is fixed (denoted $\hat{x}$) and the depth is also fixed (at $\bar{v} > 0$).

We claim that the sequence $\|x_{k+1} - \hat{x}\|$ is not bounded. For sake of a contradiction, suppose that it is bounded. Then the η-subgradients are bounded (by a constant Λ) by [102, Prop. XI.4.1.2]. Apply Lemma 4.3.6; since the v_k and the f_k^{lev} are fixed, the sequence $\{\|x_{k+1} - \hat{x}\|\}_k$ increases by a constant factor $(1 - \kappa_f)^2 \bar{v}^2 / \Lambda^2$ at each iteration. This contradicts the boundedness.

We claim now that $\mu_k \to \infty$. In view of a contradiction, suppose that $\{\mu_k\}$ is bounded: let $\bar{\mu} > 0$ be such that $\mu_k \leq \bar{\mu}$ for all k large enough. Using (4.20) we have that

$$\mu_k v_k = \mu_k \hat{e}_k + \mu_k^2 \|\hat{g}_k\|^2 \geq -\mu_k \eta + \mu_k^2 \|\hat{g}_k\|^2 \geq -\bar{\mu}\eta + \|x_{k+1} - \hat{x}\|^2 .$$

As $\{v_k\}$ is nonincreasing, we have that $\bar{\mu} v_0 \geq \mu_k v_k \geq -\bar{\mu}\eta + \|x_{k+1} - \hat{x}\|^2$, contradicting that $\|x_{k+1} - \hat{x}\|^2 \to \infty$. Hence, $\mu_k \to \infty$.

Since there is eventually no noise attenuation, we have (see line 21)

$$\hat{e}_k < -\kappa_{\text{att}} \mu_k \|\hat{g}_k\|^2 < 0 \qquad \text{for all } k \text{ large enough.}$$

By definition of $\hat{e}_k$ in (4.15), we have that $\hat{e}_k \geq f(\hat{x}_k) - \eta - \bar{f}_{-k}(\hat{x}_k) \geq -\eta$, from (4.3) and (4.14). This yields $\|\hat{g}_k\|^2 \leq \eta / (\kappa_{\text{att}} \mu_k)$. Since $\mu_k \to \infty$, we get that $\hat{g}_k \to 0$. Hence, (4.29) holds with $\mathcal{I}$ being all the large indices. Since the sequence $\{\hat{x}_k\}$ is finite (thus bounded), we can conclude with Lemma 4.3.3. ■

Remark 4.3.8 (More sophisticated versions)**.** *We emphasize that the important point of the above proofs was to control the linearization error at descent steps. As a consequence, we could add a test in the algorithm to stop the oracle whenever we detect that the descent test will be false. This version of the algorithm would be proved to be convergent with the exact same proof.*

We could also cover the case of "upper oracles" in the terminology of [dOSL14]. The algorithm could indeed deal with controllable linearizations overestimating the function by no more than a constant $\eta^g > 0$. The same convergence proof would result in a convergence to an $(\eta + \eta^g)$-solution.

4.4 Numerical illustration on energy optimization

We illustrate the efficiency of our approach on two classes of energy optimization problems: two-stage stochastic programming problems (with publicly available data sets) and chance-constrained optimization problems arising from cascaded reservoir management (with real-life data). Each following subsection treats one family of problems for which we consider an exact oracle and a cutting-plane generator incorporating uncontrolled linearizations. Our goal here is not to obtain the best computational results for these problems, but to show that using the uncontrolled bundle information can speed-up computations.

Specifically, we compare Algorithm 4 and Algorithm 5 using cutting-plane generators to their basic versions not using any additional uncontrolled linearizations ($J_k^{\text{u}} = \emptyset$ for all k). We have implemented these algorithms in MATLAB (using the Gurobi solver for LP and QP problems); we name them as follows

- u-P: Algorithm 4, the proximal bundle using uncontrolled information,
- P: Algorithm 4 with $J_k^{\mathrm{u}} = \emptyset$, the standard proximal bundle algorithm,
- u-L: Algorithm 5, the level bundle using uncontrolled information,
- L: Algorithm 5 with $J_k^{\mathrm{u}} = \emptyset$, the (new) level algorithm.

Notice that the comparison between level and proximal bundle algorithms are (surprisingly) rare; an exception is [62]. In particular, in section 5.1.4 of [dS15], tests are reported with tuning parameters of proximal and level bundle methods. Here we set the parameters of the algorithms according to these tests: for both algorithms, we take $\kappa_f = 0.1$ and $\kappa_{\mathrm{att}} = 0.99$; for Algorithm 4, we take $t_1 = 10$, $\bar{t} = 10^{-6}$, and the update rule of Section 5.1.2 of [dS15] (with $a = 2$) for t_k ; for Algorithm 2, we take $\kappa_l = 0.2$ and $\mu_{\mathrm{large}} = 5$.

Since the controlled oracle is exact ($\eta = 0$), the four methods converge to the exact solution. The algorithms are compared by measuring the number of calls to the exact oracle and the total CPU time to reach the stopping test. We use the relative stopping tolerance

$$\mathrm{tol}_e = \mathrm{tol}_\Delta = 10^{-5}(1 + f(\hat{x}_k)) \qquad \text{and} \qquad \mathrm{tol}_g = 10^{-4}(1 + f(\hat{x}_k)).$$

These experiments were performed on a computer with Intel(R) Core(TM), i3-3110M CPU 2.40, 4G (RAM), under Windows 8, 64Bits.

We also compare the speed and robustness of the algorithms globally on all the problems by using performance profiles [70]. For each algorithm, we plot the proportion of problems that it solved within a factor of the time required by the best algorithm. More precisely, if we denote by $t_A(p)$ the time spent by algorithm A to solve problem p and $t^*(p)$ the best time for solving problem p, then the proportion of problems solved by A within a factor τ is

$$\theta_A(\tau) = \frac{\text{number of problems } p \text{ such that } t_A(p) \leq \tau\, t^*(p)}{\text{total number of problems}}.$$

4.4.1 Two-stage stochastic linear optimization problems

Problem and instances description. Two-stage stochastic linear problems arise in the planning of hydro-electric power generation; see e.g. [185] and [61] for applications to the New Zealand and Brazilian electricity system. The problem can be formulated as (4.1) with

$$X = \{x \in \mathbb{R}^n_+ :\ Ax = b\} \quad \text{and} \quad f(x) = c^\top x + \sum_{i=1}^{N} \pi_i f_i(x)$$

where $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m_1 \times n}$ and $b \in \mathbb{R}^{m_1}$ are such that the set X is bounded. Also,

$$f_i(x) := \min_{y \in \mathbb{R}^{n_2}_+} q^\top y \ \text{ s.t. } \ Tx + Wy = h_i \tag{4.34}$$

is the so-called recourse function associated with the i-th scenario $h_i \in \mathbb{R}^{m_2}$ (which has a probability $\pi_i > 0$). In these problems, the vectors h_i are the only uncertainty parameters and are normally distributed. The dual linear problem of (4.34) is

$$f_i(x) = \sup_{W^\top u \leq q} (h_i - Tx)^\top u. \tag{4.35}$$

We use the set of two-stage stochastic linear test-problems that have been used by several authors including [63, 142, 166]). The set is available online on the webpage of István Deák[1]. The data set consists in 7 families of problems of different sizes; we call them F1 to F7. A family of problems is given by the data (c, A, b, q, T, W) along with a generator of appropriate scenarios, which takes as an input the number of scenarios N, and returns (π_i, h_i) for $i = 1, \ldots, N$. For each family, we have 7 problems corresponding to $N \in \{100, 200, 500, 800, 1000, 1200, 1500\}$.

Oracles and cutting-plane generator. Computing exact information on f requires solving the N linear optimization subproblems (4.34)-(4.35). Solving only a fraction of these subproblems still gives inexact information on f: the optimal solution $\bar{u}_i$ giving $f_i(x)$ can also be used to under-approximate other terms $f_j(x)$ (since the dual feasible sets are the same, we have $(d_j - T_i x)^\top \bar{u}_i \leq f_j(x)$). Thus we are in the situation (4.7) with

- an exact oracle providing the value $f(x)$ and a subgradient $g \in \partial f(x)$ ($\eta = 0$) by solving exactly the N subproblems (4.35);
- an uncontrolled oracle by solving 10% of the subproblems (4.35) and taking a feasible solution of the remaining subproblems. This oracle is about 90% times faster than the fine one, but we do not know its accuracy.
- a cutting-plane generator consisting in running several iterations of a bundle method using only the uncontrolled oracle (with the same stopping test and a maximum of 100 iterations).

Numerical results. Table 4.1 presents the performances of the four algorithms on the 49 test-problems. It reports the number of (exact) oracle calls and CPU time (in minutes) required to reach convergence. Each entry is the average over the seven instances of the family, except for the last line which is the grand total over the 49 instances. This table shows that adding uncontrolled linearization does speed up significantly the two algorithms: we observe 25% less oracle calls and 10% less CPU time between L and u-L, and 39.4% and 28.6% between P and u-P.

	# oracle calls				CPU time (min)			
N	L	P	u-L	u-P	L	P	u-L	u-P
F1	19	25	13	11	1.2	1.7	1.0	0.8
F2	25	39	19	25	2.9	4.6	2.5	3.2
F3	37	56	23	30	2.5	3.7	1.8	2.2
F4	39	62	31	35	3.3	5.1	2.9	3.0
F5	38	63	36	39	4.5	7.1	4.5	4.7
F6	57	81	37	51	4.8	6.2	4.2	5.4
F7	59	68	47	49	6.7	7.9	6.1	6.6
Total	1928	2768	1440	1678	3.0h	4.2h	2.7h	3.0h

Table 4.1: Comparison of the four algorithms with respect to the number of oracle calls and the global CPU time to get convergence. Each entry is the average over the seven instances of the family, except for the last line which is the grand total over the 49 instances.

The decrease of oracle calls is more important than the one of CPU time because of the additional time taken by the calls of the cutting-plane generator and by solving larger quadratic subproblems. We still observe a decrease of CPU time in all our instances, even if the uncontrolled linearizations may have a poor accuracy. We also note that the decrease is more important for proximal algorithm

[1] `http://www.uni-corvinus.hu/index.php?id=26637`

than the level one. This is due to the fact that the level algorithm without uncontrolled information (L) does already well: we see that the CPU times of L are comparable to the ones of u-P.

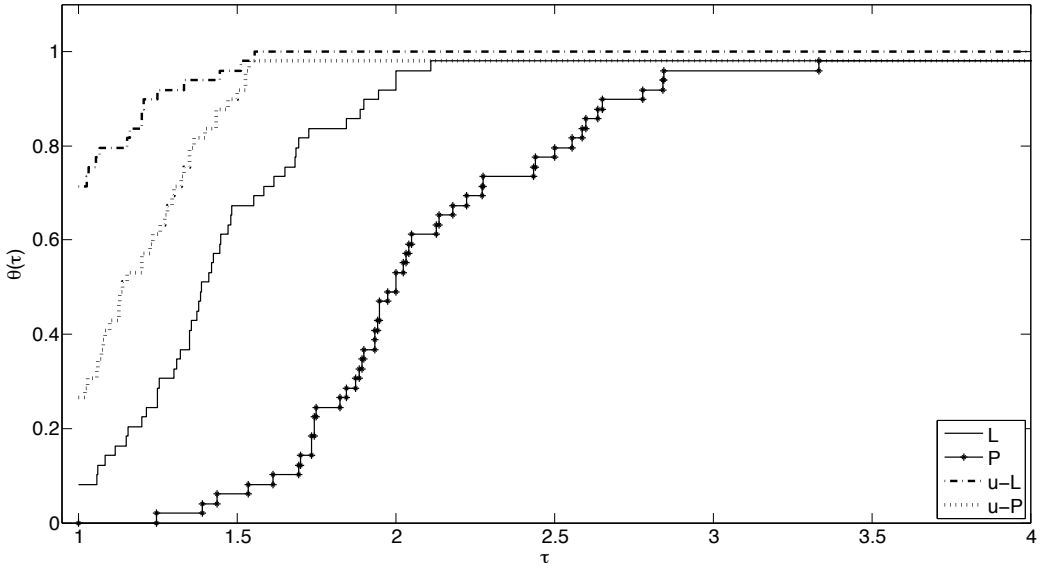


Figure 4.1: Performance profiles of the four methods over the 49 instances

For these problems, the best method in terms of both oracle calls and CPU time is u-L, the level method using the uncontrolled cutting-plane generator. Figure 4.1 confirms this by showing the performance profile of all solvers with respect to oracle calls (the plot for CPU time is similar). Since its curve is always higher, u-L clearly dominates the other methods in terms of speed and robustness. The value at $\tau = 1$ indicates that u-L is the best to solve around 70% of the 49 problems; it also solves all the problems within a factor $\tau \approx 1.5$ of the best method.

We finish with a remark about the influence of the accuracy of the uncontrolled oracle. We note indeed that we can adjust the accuracy of the uncontrolled oracle in this situation by changing the percentage of subproblems (4.35) solved. In the reported experiments, we choose to solve 10% of the subproblems because we found out that it provides a good compromise between performance of the overall algorithm and computational burden of the external module. We mention here that, during preliminary tests, we observed that solving fewer subproblems (e.g. 1% of all subproblems) tends to increase of the total number of exact oracle calls and to give higher CPU times to solve the overall problem. On the other hand, we also observed that solving more subproblems (e.g. 20% of all subproblems) tends to decrease of the number of exact oracle calls, but with higher total CPU costs (since the uncontrollable oracle becomes more expensive). In general, the efficient choice of the percentage of problems solved depends on the problem's data, such as variance of the random vectors and number of considered scenarios.

4.4.2 Chance-constraint optimization problems

Problem and instances description. Joint chance-constrained optimization problems appear in cascaded reservoir management in presence of probabilistic guarantees that volumes in the reservoirs remain within bounds see e.g. [173]. With a target probability $p \in (0, 1)$, these constraints can be expressed as $\mathrm{P}[g(y) \geq \xi] \geq p$ where $\xi \in \mathbb{R}^n$ represents the random vector of water inflows (of associate probability measure P) and $g : \mathbb{R}^m \to \mathbb{R}^n$ is an affine mapping. The associated optimization problem

can be written (see more precisely [177, Eq.(15)]) as

$$\min_{y\in Y, v\in V} q^\top y \quad \text{s.t.} \quad g(y) \geq v\,, \tag{4.36}$$

where Y is a bounded polyhedron and V the set of points satisfying the probability constraint. When considering finitely many scenarios $\{\xi^1, \ldots, \xi^N\}$ with associated probability $\{\pi_1, \ldots, \pi_N\}$ (see e.g. [166, Chap. 4], [64, Chap. 6]), V can be expressed as the following feasibility set

$$V = \big\{v \in \mathbb{R}^n \colon \exists\, z \in \{0,1\}^N,\ \pi^\top z \leq 1-p,\ (1-z_i)\xi^i \leq v - \underline{b}\, z_i,\ i = 1, \ldots, N\big\}$$

where $\underline{b} \in \mathbb{R}^n$ is defined component-wise by $\underline{b}_j := \min_{1\leq i\leq N} \xi^i_j$. Then the dual problem has the form (4.1) with

$$X = \mathbb{R}^n_+ \quad \text{and} \quad f(x) := -(h(x) + d(x))$$

where $h(x)$ is the optimal value of a mere linear programming problem (since Y is a polyedron and g is affine)

$$h(x) := \min_{y\in Y}\ q^\top y - x^\top g(y)$$

and $d(x)$ is the optimal value of a (large-scale) mixed-binary linear problem

$$d(x) := \left\{ \begin{array}{lll} \min\limits_{v\in\mathbb{R}^m,\ z\in\{0,1\}^N} & x^\top v & \\ \text{s.t.} & \xi^i(1-z_i) & \leq v - \underline{b}\, z_i,\ i = 1, \ldots, N \\ & \pi^\top z & \leq 1-p\,. \end{array} \right. \tag{4.37}$$

Here we use the instances described in [177] and [173] constructed from real-life data on the French hydro-valley Isère (provided to us by EDF, the French Electricity Board). For $N \in \{50,\ 100,\ 150,\ 200,\ 250\}$ and $p \in \{80\%,\ 90\%\}$, three different scenario samples are randomly generated, and as a result, we get thirty different associated instances.

Oracles and cutting-plane generator. The bulk of the work of an exact oracle for f is to solve the mixed-binary linear optimization problem (4.37) to optimality, which is expensive as N grows. On the other hand, we have an easy way to produce feasible solutions, as follows. To any binary point $\tilde{z} \in \{0,1\}^N$ satisfying $\pi^\top \tilde{z} \leq 1-p$ we associate the vector $\tilde{v} \in \mathbb{R}^n$ such that

$$\tilde{v}_j := \max_{i\in\{l\colon z_l = 0\}} \xi^i_j \ \text{ for all } \ j = 1, \ldots, N\,. \tag{4.38}$$

Observe then that the pair $(\tilde{v}, \tilde{z})$ is feasible for (4.37). Accordingly, $d_x := x^\top \tilde{v}$ is an upper approximation for $d(x)$, which in turn provides a (cheap but imprecise) approximation for $f(x)$

$$f_x := -(h(x) + d_x) \leq f(x)\,.$$

The recent work [172] proposes a fast heuristic (denoted *Heuristic h1* therein) to compute a good candidate $\tilde{z}$ (and therefore $\tilde{v}$ as above) to approximate a solution of problem (4.37). Thus we are in the situation (4.7) with

- an exact oracle providing the value $f(x)$ and a subgradient $g \in \partial f(x)$ ($\eta = 0$) by solving exactly the subproblem (4.37) with Gurobi;
- an uncontrolled oracle using the heuristic of [172] and (4.38);
- a cutting-plane generator consisting in running several iterations of a bundle method using only the uncontrolled oracle (with the same stopping test and a maximum of 100 iterations).

Numerical results. Table 4.2 reports the number of (exact) oracle calls and CPU time (in minutes) required to reach convergence, for the four algorithms over the 30 test-problems. Each entry is the average over the instances with same N and p, except for the last line which is the grand total over the 30 instances.

		# oracle calls				CPU time (min)			
N	p	L	P	u-L	u-P	L	P	u-L	u-P
50	0.8	18	32	13	13	0.6	0.9	0.4	0.4
50	0.9	18	18	11	9	0.4	0.4	0.3	0.2
100	0.8	19	24	12	15	2.9	3.5	1.7	1.6
100	0.9	19	19	11	21	1.3	1.1	0.7	1.2
150	0.8	19	24	11	20	12.8	12.0	6.2	7.9
150	0.9	18	24	8	18	2.4	3.6	1.2	2.1
200	0.8	20	19	13	23	24.2	20.2	12.1	12.6
200	0.9	19	22	10	6	5.6	5.7	3.2	1.4
250	0.8	18	32	15	15	48.9	45.6	29.9	28.0
250	0.9	19	36	12	40	17.0	26.1	5.3	20.5
Total		558	751	346	543	5.8h	6.0h	3.0h	3.8h

Table 4.2: Comparison of the four algorithms with respect to oracle calls and global CPU time to get convergence. Each entry is the average over the three instances, except for the last line which is the grand total over the 30 instances.

The figures show that introducing uncontrolled linearizations reduces both the number of oracle calls and the CPU time for both proximal and level algorithms. This improvement is even more significant than for the two-stage problems: the reduction of CPU times is of 47% for u-L and 36% for u-P.

We also see that u-L is more efficient u-P, both in CPU time and number of oracle calls. In fact u-L makes a better use of uncontrolled information added by the cutting-plane generator: L and P are comparable in terms of CPU time whereas u-L is faster than u-P (by more than 20%). The performance profiles of Figure 4.2 confirm that u-L is the fastest and most robust among the four methods.

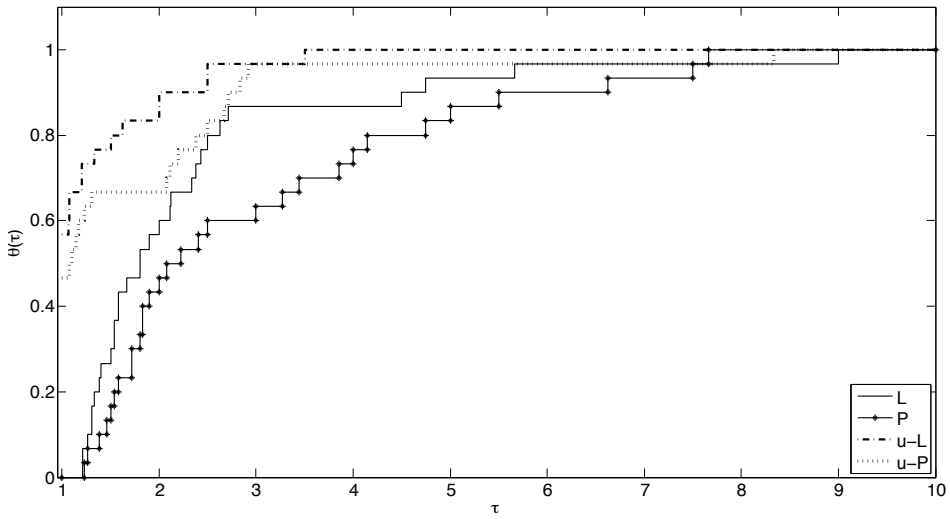


Figure 4.2: Performance profiles of the four methods on the 30 instances.

4.5 Conclusions

This chapter analyzes two bundle algorithms (a proximal one and a level one) handling cheap uncontrolled inexact linearizations, incorporated by an abstract cutting-plane generator. Beside the formalisation and the emphasis on uncontrolled bundle information, the main technical contribution of this chapter is the challenging convergence analysis of the level algorithm. This algorithm extends [41] to handle inexact η-oracles and to use the general uncontrolled cutting-plane model generator. The key feature of this algorithm is a novel noise attenuation rule, that can be seen as an implicit version of the one of [113]. Numerical experiments on two energy optimization problems show that including cheap uncontrolled information can decrease the CPU time to reach optimality, and that the level algorithm, fully exploiting the additional information, works particularly well on these problems.

To our knowledge, this chapter is the first one to consider cheap uncontrolled inexact information within bundle methods and to show the interest to use it. A recent preprint [175] builds on this line of research in a context of Benders decomposition. Note finally that we consider here an extreme case of a cutting-plane generator with no control at all on the linearizations. More sophisticated and subtle ways to incorporate cheap information should be possible, as for exemple using "adaptative oracles". Such inexact oracles would interact with the bundle algorithm (though accuracy parameters as in [59] but not only) and would be able to choose between several available approximation schemes with increasing accuracy and increasing cost (as for example the three specific heuristics of [25] for a combinatorial problem). A general study of smart and communicating oracles goes beyond the scope of this work and deserves special research and developments.

Chapter 5

Cut-generating functions

This chapter presents to the joint work with the "task force", as called by Claude Lemaréchal: we invited in Grenoble three colleagues, Aris Daniilidis (University of Chile), Gérard Cornuéjols (Carnegie Mellon University), and Michele Conforti (University of Padoue, Italy), for a long research stay in Spring 2011, with the ambitious objective to connect convex analysis and cutting theory in discrete optimization. This fruitful visit has led to two publications [Mal-8] and [Mal-31] about a formal theory of cut-generating functions, generalizing the famous Gomory's cuts

5.1 Introduction

In this chapter, we consider sets of the form

$$X = X(R,S) := \left\{x \in \mathbb{R}^n_+ \,:\, Rx \in S\right\}, \tag{5.1a}$$

$$\text{where } \begin{cases} R = [r_1 \dots r_n] \text{ is a real } q \times n \text{ matrix}, \\ S \subset \mathbb{R}^q \text{ is a nonempty closed set with } 0 \notin S. \end{cases} \tag{5.1b}$$

In other words, our set X is the intersection of a closed convex cone with a pre-image by a linear mapping. This model goes back to [106], where S was a finite set: constraints $Rx = b$ were considered for several righthand sides b. Here, we rather consider a general (possibly infinite) set S and a varying constraint matrix R. The closed convex hull of X does not contain 0 (see Lemma 5.2.1 below) and we are then interested in *separating* 0 from X: we want to generate *cuts*, i.e. inequalities that are valid for X, which we write as

$$c^\top x \geqslant 1\,, \quad \text{for all } x \in X\,. \tag{5.2}$$

5.1.1 Motivating examples

Our first motivation comes from (mixed) integer linear programming.

Example 5.1.1 (An integer linear program)**.** Let us first consider a pure integer program, which consists in optimizing a linear function over the set defined by the constraints

$$Dz = d \in \mathbb{R}^m\,, \quad z \in \mathbb{Z}^p_+\,. \tag{5.3}$$

Set $n := p - m$, assume the matrix D to have full row-rank m and select m independent columns (a *basis*). The corresponding decomposition $z = (x, y)$ into non-basic and basic variables amounts to

writing the above feasible set as the intersection of $\mathbb{Z}^n \times \mathbb{Z}^m$ with the polyhedron

$$P := \left\{(x, y) \in \mathbb{R}^n_+ \times \mathbb{R}^m_+ \,:\, Ax + y = b\right\} \tag{5.4}$$

for suitable $m \times n$ matrix A and m-vector b.

Relaxing the nonnegativity constraint on the basic variables y, we obtain the classical *corner polyhedron* [87], namely the convex hull of

$$\left\{(x, y) \in \mathbb{Z}^n_+ \times \mathbb{Z}^m \,:\, Ax + y = b\right\}.$$

This model has the form (5.1) if we set

$$q = n + m\,, \quad R = \begin{bmatrix} I \\ -A \end{bmatrix}, \quad S = \mathbb{Z}^n \times \left(\mathbb{Z}^m - \{b\}\right), \tag{5.5}$$

where $\mathbb{Z}^m - \{b\}$ denotes the translation of $\mathbb{Z}^m$ by the vector $-b$. Assuming $b \notin \mathbb{Z}^m$, the above S is a closed set not containing the origin.

For $m = 1$, (5.4) has a single constraint

$$\sum_{j=1}^{n} a_j x_j + y = b\,, \quad y \in \mathbb{Z}\,, \quad x \in \mathbb{Z}^n_+;$$

the celebrated Gomory cut [86] is

$$\sum_{j:f_j \leqslant f_0} \frac{f_j}{f_0} x_j + \sum_{j:f_j > f_0} \frac{1 - f_j}{1 - f_0} x_j \geqslant 1\,, \tag{5.6}$$

where $f_j = a_j - \lfloor a_j \rfloor$ and $f_0 = b - \lfloor b \rfloor$. Inequality (5.6) is valid for the corner polyhedron and cuts off the basic solution $(x = 0, y = b)$. In the x-space $\mathbb{R}^n$, this inequality is a cut as defined in (5.2). We will demonstrate in Example 5.2.8 how to recover such a cut from our formalism.

Except for the translation by the basic solution $(0, b)$, S is quasi instance-independent. This is actually a crucial feature; it determines the approach developed in this chapter, namely cut-generating functions to be developed below. □

Example 5.1.2 (A mixed-integer linear program)**.** In our integer program (5.3), let us now relax not only nonnegativity of the basic variables but also integrality of the non-basic variables: the corner polyhedron is further relaxed to the convex hull of

$$\left\{(x, y) \in \mathbb{R}^n_+ \times \mathbb{Z}^m \,:\, Ax + y = b\right\}.$$

We are still in the context of (5.1) with

$$q = m\,, \quad R = -A\,, \quad S = \mathbb{Z}^m - b\,;$$

this is the model considered in [2] for $m = 2$, and in [38] for general m. Other relevant references are [17, 18, 68, 88, 106].

This type of relaxation can be used when (5.3) becomes a mixed-integer linear program

$$Dz = d \in \mathbb{R}^m\,, \quad z \geqslant 0\,, \quad z_j \in \mathbb{Z}\,,\ j \in J\,,$$

where J is some subset of $\{1, \ldots, p\}$. Extract a basis as before and choose a subset of basic variables indexed in J; call $m' \leqslant m$ the number of rows in this restriction and $b' \in \mathbb{R}^{m'}$ the resulting restriction

of b (in other words, ignore a number $m - m'$ of linear constraints). Relax nonnegativity of the m' remaining basic variables, as well as integrality of the non-basic variables indexed in J. This results in (5.1), with

$$q = m' \,, \quad R = -A \,, \quad S = \mathbb{Z}^{m'} - b' \,.$$

Any cut for this set X is *a fortiori* a valid inequality for the original mixed-integer linear program.

When $m' = 1$, a classical example of such inequalities is

$$\sum_{j:a_j>0} \frac{a_j}{f_0} x_j - \sum_{j:a_j<0} \frac{a_j}{1-f_0} x_j \geqslant 1 \,. \tag{5.7}$$

Actually, Gomory's mixed-integer cuts [86] combine (5.6) for the integer non-basic variables with the above formula for the continuous ones. □

Model (5.1) occurs in other areas than integer programming and we give another example.

Example 5.1.3 (Complementarity problem)**.** Still using P of (5.4), let

$$E \subset \{1, 2, \ldots, m\} \times \{1, 2, \ldots, m\} \quad \text{and} \quad C := \{y \in \mathbb{R}^m_+ \,:\, y^i y^j = 0 \,, (i,j) \in E\}$$

(in this chapter, $\subset$ stands for inclusion and $\subsetneq$ for strict inclusion).

The set of interest is then $P \cap (\mathbb{R}^n \times C)$. It can be modeled by (5.1) where

$$q = m \,, \quad R = -A \,, \quad S = C - b \,.$$

Cuts have been used for complementarity problems of this type, for example in [108]. □

We will retain from these examples the dissymetry between S (a very particular and highly structured set) and R (an arbitrary matrix). Keeping this in mind, we will consider that (q, S) is given and fixed, while (n, R) is instance-dependent data: our cutting problem can be viewed as *parametrized* by (n, R). This point of view is natural for the last two examples; but some pre-processing (to be seen in Example 5.2.8) is needed to apply it to Example 5.1.1: by (5.5), S does depend on the data through its dimension q, which depends on n.

5.1.2 Introducing cut-generating functions

To generate cuts in the present situation, it would be convenient to have a mapping, taking instances of (5.1) as input, and producing cuts as output. What we need for this is a function

$$\mathbb{R}^q \ni r \mapsto \rho(r) \in \mathbb{R}$$

which, applied to the columns r_j of a $q \times n$ matrix R (an arbitrary matrix, with an arbitrary number of columns) will produce the n coefficients $c_j := \rho(r_j)$ of a cut (5.2). We stress the fact that ρ must assign a number $\rho(r)$ to *any* $r \in \mathbb{R}^q$: the function ρ is defined on the whole space.

Thus, we require from our ρ to satisfy

$$x \in X \quad \Longrightarrow \quad \sum_{j=1}^{n} \rho(r_j) x_j \geqslant 1 \,, \tag{5.8}$$

for every instance X of (5.1). Such a ρ can then justifiably be called a *cut-generating function* (CGF). The notation ρ refers to *representation*, which will appear in Definition 5.2.6 below. One of the most

well-known cut-generating functions in integer programming is the so-called Gomory function [86], which we presented in Examples 5.1.1 and 5.1.2. The corresponding cuts can be generated quickly, so they are a powerful tool in computations; indeed, they drastically speed up integer-programming solvers [31].

So far, a CGF is a rather abstract object, as it lies in the (vast!) set of functions from $\mathbb{R}^q$ to $\mathbb{R}$; but the following observation allows a drastic reduction of this set.

Remark 5.1.4 (Dominating cuts)**.** Consider in (5.2) a vector c' with $c'_j \leqslant c_j$ for $j = 1, \ldots, n$; then ${c'}^\top x \leqslant c^\top x$ whenever $x \geqslant 0$. If c' is a cut, it is tighter than c in the sense that it cuts a bigger portion of $\mathbb{R}^n_+$. We can impose some "minimal" character to a CGF, in order to reach some "tightness" of the resulting cuts. □

With this additional requirement, the decisive Theorem 5.2.3 below will show that a CGF can be imposed to be *convex positively homogeneous* (and defined on the whole of $\mathbb{R}^q$; positive homogeneity means $\rho(tr) = t\rho(r)$ for all $r \in \mathbb{R}^q$ and $t > 0$). This is a fairly narrow class of functions indeed, which is fundamental in convex analysis. Such functions are in correspondence with closed convex sets and in our context, this correspondence is based on the mapping $\rho \mapsto V$ defined by

$$V = V(\rho) := \left\{ r \in \mathbb{R}^q \,:\, \rho(r) \leqslant 1 \right\}, \tag{5.9}$$

which turns out to be a cornerstone: via Theorem 5.2.5 below, (5.9) establishes a correspondence between the CGF's and the so-called *S-free sets*. As a result, cut-generating functions can alternatively be studied from a geometric point of view, involving sets V instead of functions ρ. This situation, common in convex analysis, is often very fruitful. With regard to Remark 5.1.4, observe that $V(\rho)$ increases when ρ decreases: small ρ's give large V's. However the converse is not true because the mapping in (5.9) is many-to-one and therefore has no inverse. A first concern will therefore be to specify appropriate correspondences between (cut-generating) functions and (S-free) sets.

5.1.3 Scope of the chapter

The aim of the chapter is to present a formal theory of minimal cut-generating functions and maximal S-free sets, valid independently of the particular S. Such a theory would gather and synthetize a number of papers dealing with the above problem for various special forms for S: [2, 17, 18, 38, 68, 127] and references therein. For this, we use basic tools from convex analysis and geometry. Readers not familiar with this field may use [103] (especially its Chap. C) for an elementary introduction, while [101, 156] are more complete.

The chapter is organized as follows.

- Section 5.2 states more accurately the concepts of CGF's and S-free sets.
- Section 5.3 studies the mapping (5.9). We show that the pre-images of a given V (the representations of V) have a unique largest element γ_V and a unique smallest element μ_V; in view of Remark 5.1.4, the latter then appears as *the* relevant inverse of $\rho \mapsto V(\rho)$.
- In Section 5.4, we study the correspondence $V \leftrightarrow \mu_V$. We show that different concepts of minimality come into play for ρ in Remark 5.1.4. Geometrically they correspond to different concepts of maximality for V.
- We also show in Section 5.5 that these minimality concepts coincide in a number of cases.
- Finally we have a conclusion section, with some suggestions for future research.

The ideas in Sections 5.2 and 5.3 extend in a natural way the earlier works mentioned above. However, Sections 5.4 and 5.5 contain new results.

5.2 Cut-generating functions: definitions and first results

We begin with making sure that our framework is consistent. We will use $\mathrm{conv}\,(X)$ [resp. $\overline{\mathrm{conv}}\,(X)$] to denote the convex hull [resp. closed convex hull] of a set X.

Lemma 5.2.1. *With X given as in* (5.1)*, $0 \notin \overline{\mathrm{conv}}\,(X)$.*

Proof. Assume $X \neq \emptyset$, otherwise we have nothing to prove. Since 0 does not lie in the closed set S, there is $\varepsilon > 0$ such that $s \in S$ implies $\|s\|_1 \geqslant \varepsilon$; and by continuity of the mapping $x \mapsto Rx$, there is $\eta > 0$ such that $\|x\|_1 \geqslant \eta$ for all $x \in X \subset \mathbb{R}^n_+$. This means

$$\|x\|_1 = \sum_{j=1}^{n} |x_j| = \sum_{j=1}^{n} x_j \geqslant \eta\,, \quad \text{for all } x \in X\,.$$

In other words, the hyperplane $\sum_j x_j \geqslant \eta$ separates 0 from X, hence from $\overline{\mathrm{conv}}\,(X)$. ■

Remember that we are interested in functions ρ satisfying (5.8) for any (n, R) in (5.1). There are too many such functions, we now proceed to specify which exactly are relevant.

5.2.1 Sublinear cut-generating functions suffice

The following lemma, inspired from Claim 1 in the proof of [17, Lem. 23], is instrumental for our purpose.

Lemma 5.2.2. *Let ρ be a* CGF*. For all sets of K vectors $r_k \in \mathbb{R}^q$ and nonnegative coefficients α_k, the following relation holds:*

$$\sum_{k=1}^{K} \alpha_k r_k = 0 \quad \Longrightarrow \quad \sum_{k=1}^{K} \alpha_k \rho(r_k) \geqslant 0\,.$$

Proof. Call $e \in \mathbb{R}^q$ the vector of all ones and $\alpha \in \mathbb{R}^K$ the vector of α_k's; take $t \geqslant 0$ and define the vectors in $\mathbb{R}^{K+q}$

$$x := \begin{bmatrix} 0 \\ e \end{bmatrix}\,, \quad d := \begin{bmatrix} \alpha \\ 0 \end{bmatrix}\,, \quad \text{so that} \quad x + td = \begin{bmatrix} t\alpha \\ e \end{bmatrix} \in \mathbb{R}^{K+q}_+\,.$$

Then pick $s \in S$; make an instance of (5.1) with $n = K + q$ and $R := \big[r_1 \,\ldots\, r_K \,|\, D(s)\big]$, where the $q \times q$ matrix $D(s)$ is the diagonal matrix whose diagonal is the vector s. Observing that

$$R(x + td) = t \sum_k \alpha_k r_k + D(s)e = s\,,$$

$x + td$ is feasible in the resulting instance of (5.1a): (5.8) becomes

$$t \sum_{k=1}^{K} \alpha_k \rho(r_k) \geqslant 1 - z\,,$$

where z is a fixed number gathering the result of applying ρ to the columns of $D(s)$. Letting $t \to +\infty$ proves the claim. ■

Now we introduce some notation. The *domain* and *epigraph* of a function $\rho : \mathbb{R}^q \to \mathbb{R} \cup \{+\infty\}$ are

$$\operatorname{dom}\rho := \big\{r \in \mathbb{R}^q \,:\, \rho(r) < +\infty\big\} \quad \text{and} \quad \operatorname{epi}\rho := \big\{(r,z) \in \mathbb{R}^{q+1} \,:\, z \geqslant \rho(r)\big\}\,.$$

If $\operatorname{dom}\rho$ is the whole of $\mathbb{R}^q$ (i.e., $\rho(r)$ is a finite real number for all $r \in \mathbb{R}^q$), we say that ρ is *finite-valued*; a convex finite-valued function is continuous on $\mathbb{R}^q$. A function is said to be *sublinear* if it is convex and positively homogeneous; or equivalently if its epigraph is a convex cone. The *conical hull* $\operatorname{cone}(\operatorname{epi}\rho)$ of $\operatorname{epi}\rho$ is the set of nonnegative combinations of points $(r,z) \in \operatorname{epi}\rho$:

$$r = \sum_{k=1}^{K} \alpha_k r_k\,,\ z = \sum_{k=1}^{K} \alpha_k z_k\,, \quad \text{with} \quad z_k \geqslant \rho(r_k)\,,\ \alpha_k \geqslant 0\,,\ k = 1,\dots,K\,,$$

where K is an arbitrary integer. This conical hull is itself the epigraph of a sublinear function $\bar\rho$, called the *sublinear hull* of ρ. Its value at r is the smallest possible of the above z's:

$$\bar\rho(r) := \inf\Big\{\sum_{k=1}^{K} \alpha_k \rho(r_k) \,:\, \sum_{k=1}^{K} \alpha_k r_k = r,\ \alpha_k \geqslant 0\Big\}\,. \tag{5.10}$$

Of course $\bar\rho \leqslant \rho$; in the spirit of Remark 5.1.4, our next result establishes that a CGF can be improved by taking its sublinear hull.

Theorem 5.2.3. *If ρ is a* CGF*, then $\bar\rho$ of* (5.10) *is nowhere $-\infty$ and is again a* CGF.

Proof. Express every $r \in \mathbb{R}^q$ as a nonnegative combination: $\sum_k \alpha_k r_k - r = 0$, hence (Lemma 5.2.2) $\sum_{k=1}^K \alpha_k \rho(r_k) + \rho(-r) \geqslant 0$ and $\bar\rho(r) \geqslant -\rho(-r) > -\infty$.

Then take an instance $R = [r_j]_{j=1}^n$ of (5.1b). If it produces $X = \emptyset$ in (5.1a), there is nothing to prove. Otherwise fix $\bar x \in X$.

Any positive decomposition $r_j = \sum_k \alpha_{j,k} r_{j,k}$ of each column of R satisfies

$$\bar s := R\bar x = \sum_{j=1}^{n} \bar x_j r_j = \sum_{j=1}^{n} \bar x_j \sum_{k=1}^{K} \alpha_{j,k} r_{j,k} = R_+ x_+\,,$$

where $x_+ \in \mathbb{R}^{nK}$ denotes the vector with coordinates $\alpha_{j,k}\bar x_j \geqslant 0$ and R_+ the matrix whose nK columns are $r_{j,k}$. Then R_+ is a possible instance of (5.1b) and $R_+ x_+ = \bar s \in S$, so the CGF ρ separates x_+ from 0:

$$1 \leqslant \sum_{j,k} \rho(r_{j,k})\big(\alpha_{j,k}\bar x_j\big) = \sum_{j=1}^{n} \Big(\sum_{k=1}^{K} \alpha_{j,k}\rho(r_{j,k})\Big)\bar x_j\,. \tag{5.11}$$

Apply the definition of an infimum: for each $\varepsilon > 0$ we can choose our decompositions $(r_{j,k}, \alpha_{j,k})$ so that

$$\sum_{k=1}^{K} \alpha_{j,k}\rho(r_{j,k}) \leqslant \bar\rho(r_j) + \varepsilon\,, \quad \text{for } j = 1,\dots,n$$

which yields with (5.11)

$$1 \leqslant \sum_{j=1}^{n} \big(\bar\rho(r_j) + \varepsilon\big)\bar x_j = \sum_{j=1}^{n} \bar\rho(r_j)\bar x_j + \varepsilon \sum_{j=1}^{n} \bar x_j\,.$$

Since ε is arbitrarily small – while $\bar x$ is fixed – we see that $\bar\rho$ does satisfy (5.8). ∎

In view of Remark 5.1.4, Theorem 5.2.3 allows us to restrict our attention to CGF's that are sublinear; and their domain is the whole space by definition. We are now in a position to explain the use of the operation (5.9) in our context.

5.2.2 Cut-generating functions as representations

From now on, a CGF ρ will always be understood as a (finite-valued) sublinear function. By continuity and because $\rho(0) = 0$, $V(\rho)$ in (5.9) is a closed convex neighborhood of 0 in $\mathbb{R}^q$. Besides, its interior and boundary are respectively

$$\operatorname{int}(V(\rho)) = \{r \in V \,:\, \rho(r) < 1\}\,, \quad \operatorname{bd}(V(\rho)) = \{r \in V \,:\, \rho(r) = 1\}\,. \tag{5.12}$$

This follows from the Slater property $\rho(0) = 0$ (see, e.g., [103, Prop. D.1.3.3]); it can also be checked directly:
– by continuity, $\rho(\bar r) < 1$ implies $\rho(r) \leqslant 1$ for r close to $\bar r$;
– by positive homogeneity, $\rho(\bar r) = 1$ implies $\rho(r) = 1 + \varepsilon$ for $r = (1+\varepsilon)\bar r$.
The relevant neighborhoods for our purpose are the following:

Definition 5.2.4 (S-free set)**.** Given a closed set $S \subset \mathbb{R}^q$ not containing the origin, a closed convex neighborhood V of $0 \in \mathbb{R}^q$ is called S-free if its interior contains no point in S: $\operatorname{int}(V) \cap S = \emptyset$. □

Let us make clear the importance of this definition.

Theorem 5.2.5. *Let the sublinear function $\rho : \mathbb{R}^q \to \mathbb{R}$ and the closed convex neighborhood V (of $0 \in \mathbb{R}^q$) satisfy* (5.9)*. Then ρ is a* CGF *for* (5.1) *if and only if V is S-free.*

Proof. Let V be S-free; in view of (5.12), $\rho(r) \geqslant 1$ for all $r \in S$. In particular, take a $q \times n$ matrix R, $x \in X$ of (5.1a) and set $r := Rx \in S$. Then, using sublinearity,

$$1 \leqslant \rho(Rx) = \rho\Big(\sum_{j=1}^n x_j r_j\Big) \leqslant \sum_{j=1}^n x_j \rho(r_j)\,;$$

ρ is a CGF.

Conversely, suppose V is not S-free: again from (5.12), there is some $r_1 \in S$ such that $\rho(r_1) < 1$. Take in (5.1b) the instance $(n, R) = (1, [r_1])$. Then $1 \in X$ ($r_1 \in S$), so $c_1 := \rho(r_1) < 1$ cannot be a cut. ■

This allows a new definition of CGF's, much more handy than the original one:

Definition 5.2.6 (CGF as representation)**.** Let $V \subset \mathbb{R}^q$ be a closed convex neighborhood of the origin. A *representation* of V is a finite-valued sublinear function ρ such that

$$V = \big\{r \in \mathbb{R}^q \,:\, \rho(r) \leqslant 1\big\}\,. \tag{5.13}$$

We will say that ρ *represents* V.

A sublinear cut-generating function for (5.1) is a representation of an S-free set. □

A finite-valued sublinear function ρ represents a unique $V = V(\rho)$, well-defined by (5.13). One easily checks monotonicity of the mapping $V(\cdot)$:

$$\rho \leqslant \rho' \quad \Longrightarrow \quad V(\rho) \supset V(\rho')\,. \tag{5.14}$$

Conversely, one may ask whether a given closed convex neighborhood of the origin V always has a representation. In fact, (5.13) fixes via (5.12) the value $\rho(r) = 1$ on the boundary of V; whether this set of prescribed values can be extended to make a sublinear function on the whole of $\mathbb{R}^q$ is not obvious. This will be the subject of Section 5.3, where we will see that this is indeed possible; there may even be infinitely many extensions, and we are interested in the small ones. Now we illustrate the material introduced so far with some examples.

5.2.3 Examples

We start with a simple 1-dimensional example supporting the claim that the mapping $\rho \to V$ of (5.13) is many-to-one – or equivalently that a given V may have several representations.

Example 5.2.7. With $q = 1$, consider $V =]-\infty, 1]$. In $\mathbb{R}^1$, the positively homegeneous functions have the form

$$\rho(r) = \begin{cases} \alpha r & \text{for } r \geqslant 0 \\ \beta r & \text{for } r \leqslant 0\,; \end{cases}$$

they are convex when $\alpha \geqslant \beta$.

Taking $r = 1 \in V$ in (5.13) imposes $\alpha \leqslant 1$, while taking $r = 1 + \varepsilon \notin V$ ($\varepsilon > 0$) imposes $\alpha > 1/(1+\varepsilon)$. Altogether $\alpha = 1$. On the other hand, letting $r \to -\infty$, the property $\beta r \leqslant 1$ imposes $\beta \geqslant 0$.

Conversely, we easily see that, for any $\beta \in [0, 1]$, the function

$$\rho(r) = \begin{cases} r & \text{for } r \geqslant 0 \\ \beta r & \text{for } r \leqslant 0 \end{cases}$$

is sublinear and satisfies (5.13). Thus, the representations of V are exactly the functions of the form $\rho(r) = \max\{r, \beta r\}$, for $\beta \in [0, 1]$.

This example suggests – and Lemma 5.3.2 will confirm – that nonuniqueness appears when V is *unbounded*. □

Example 5.1.2 is quite suitable for illustration, Figure 5.1 visualizes it for $q = m = 2$. The dots are the set $S = \mathbb{Z}^2 - \{b\}$. The stripe V of the left part, called a *split set*, is used in the framework of disjunctive cuts. Other neighborhoods can be considered, for example triangles (right part of the picture) as in [2].

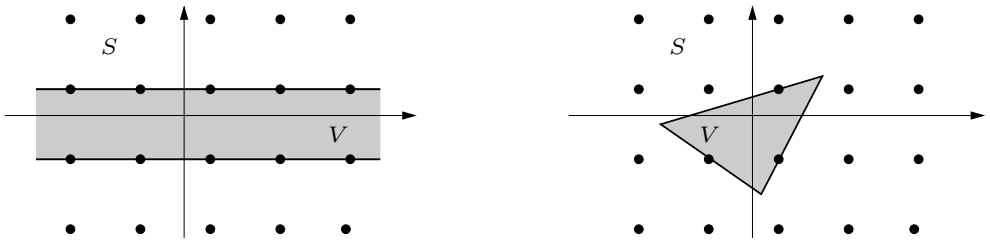


Figure 5.1: Two S-free sets for $q = 2$

With $q = 1$, no need for a picture and the calculations in Example 5.1.2 can be worked out. In this case, $X \subset \mathbb{R}^n_+$ is defined by $a^\top x \in \mathbb{Z} - b$, i.e. (5.1) with $r = -a$ and $S = \mathbb{Z} - \{b\}$. The only possible S-free neighborhoods of the origin are the segments $r \in [-r_-, r_+]$ with

$$-f_0 = \lfloor b \rfloor - b \leqslant -r_- < 0 < r_+ \leqslant \lceil b \rceil - b = 1 - f_0\,.$$

For a representation ρ of this segment, the equations $\rho(r_+) = 1$ and $\rho(-r_-) = 1$ fix in a unique way

$$\rho(r) = \begin{cases} \frac{r}{r_+} & \text{if } r \geqslant 0\,, \\ -\frac{r}{r_-} & \text{if } r \leqslant 0\,. \end{cases}$$

Choose the extreme values for r_+ and r_- to obtain

$$c_j = \rho(-a_j) = \begin{cases} \frac{a_j}{f_0} & \text{if } a_j \leqslant 0\,, \\ \frac{-a_j}{1-f_0} & \text{if } a_j \geqslant 0\,, \end{cases}$$

which is just (5.7).

Finally, let us show how Gomory cuts (5.6) can be obtained as CGF's.

Example 5.2.8. Still in Example 5.1.2, take $q = m = 1$; we want to separate the set defined by

$$\sum_{j=1}^{n} a_j x_j + y = b\,, \quad y \in \mathbb{Z}\,, \quad x \in \mathbb{Z}^n_+$$

from the origin (remember that $b \notin \mathbb{Z}$). This set has the form (5.1) with

$$q = n+1\,, \quad R = \begin{bmatrix} I \\ -a^\top \end{bmatrix}, \quad S = \mathbb{Z}^n \times \big(\mathbb{Z} - \{b\}\big)\,.$$

Introduce the vector $\pi \in \mathbb{R}^{n+1}$ defined by

$$\pi_{n+1} := 1 \text{ and, for } j = 1, \dots, n\text{: } \pi_j := \begin{cases} \lfloor a_j \rfloor & \text{if } f_j \leqslant f_0\,, \\ \lceil a_j \rceil & \text{if } f_j > f_0 \end{cases}$$

and its scalar product $\pi^\top r = \sum_{j=1}^n \pi_j x_j + y$ with $r = (x, y) \in \mathbb{R}^{n+1}$. Then define

$$V := \big\{ r \,:\, \lfloor b \rfloor - b \leqslant \pi^\top r \leqslant \lceil b \rceil - b \big\}\,. \tag{5.15}$$

The assumption $b \notin \mathbb{Z}$ implies that $(0,0) \in \operatorname{int}(V)$; therefore V is a closed convex neighborhood of the origin. Furthermore, V is S-free: in fact, $b + \pi^\top r$ is an integer for every $r = (x, y) \in S$ and therefore it cannot be strictly between the two consecutive integers $\lfloor b \rfloor$ and $\lceil b \rceil$. We claim that any representation of V produces Gomory cuts.

Call e_j the jth unit vector of $\mathbb{R}^n$, so that the n columns of R are

$$r_j = \begin{pmatrix} e_j \\ -a_j \end{pmatrix}.$$

Direct calculations give

$$\pi^\top r_j = \begin{cases} \lfloor a_j \rfloor - a_j = -f_j & \text{if } f_j \leqslant f_0\,, \\ \lceil a_j \rceil - a_j = 1 - f_j & \text{if } f_j > f_0\,. \end{cases}$$

For each $j = 1, \dots, n$, consider three cases.

(i) If $\pi^\top r_j > 0$ (which implies $f_j > f_0$), there is $t > 0$ such that $t\pi^\top r_j = \lceil b \rceil - b > 0$, namely

$$t = \frac{\lceil b \rceil - b}{\pi^\top r_j} = \frac{\lceil b \rceil - b}{\lceil a_j \rceil - a_j} = \frac{1 - f_0}{1 - f_j}\,.$$

(ii) If $\pi^\top r_j < 0$ (which implies $0 < f_j \leqslant f_0$), there exists likewise $t > 0$ such that $t\pi^\top r_j = \lfloor b \rfloor - b < 0$, therefore

$$t = \frac{f_0}{f_j}\,.$$

(iii) If $\pi^\top r_j = 0$ (which implies $a_j \in \mathbb{Z}$), $t r_j \in V$ for any $t > 0$.

In (i) and (ii), the computed value of t puts tr_j on the boundary of V. Let ρ represent V; then by (5.12) and positive homogeneity, $\rho(r_j) = \frac{1}{t}\rho(tr_j) = \frac{1}{t}$ in cases (i), (ii) and $\rho(r_j) = 0$ in case (iii). Altogether,

$$\rho(r_j) = \begin{cases} \frac{f_j}{f_0} & \text{if } f_j \leqslant f_0\,, \\ \frac{1-f_j}{1-f_0} & \text{if } f_j > f_0 \end{cases}$$

for $j = 1, \ldots, n$; we recognize Gomory's formula (5.6).

As mentioned after Definition 5.2.6, the n values $\rho(r_j)$ can be extended to make a sublinear function on the whole of $\mathbb{R}^{n+1}$. This will be confirmed in the next section but can be accepted here, thanks to the simple form (5.15) of V: a stripe orthogonal to π. Indeed, the above calculations are designed so as to construct $\rho(r) = 1$ for each r such that $\pi^\top r = \lceil b \rceil - b > 0$ as in (i) [resp. $\pi^\top r = \lfloor b \rfloor - b < 0$ as in (ii)]. Then $\rho(r)$ is given by positive homogeneity for any r such that $\pi^\top r \neq 0$; and $\rho \equiv 0$ on $\pi^\perp$. □

5.3 Largest and smallest representations

In this section, we study the representation operation introduced in Definition 5.2.6. The main result is that our closed convex neighborhood V has a largest and a smallest representation. This result was already given in [19, 47, 184], with weaker assumptions in the latter work (which came to our knowledge only after [Mal-8] was completed). Here we emphasize the geometric counterpart of the result, we put the proof of [19] in perspective, and we take advantage of our stricter assumptions to develop finer results that will be useful in sequel.

5.3.1 Some elementary convex analysis

First recall some basic theory (see, e.g., [103, Chap. C]), which will be central in our development. In what follows, V will always be a closed convex neighborhood of $0 \in \mathbb{R}^q$.

A common object in convex analysis is the *gauge*

$$\mathbb{R}^q \ni r \mapsto \gamma_V(r) := \inf\{\lambda > 0 \,:\, r \in \lambda V\}\,, \tag{5.16}$$

a (nonnegative) finite-valued sublinear function. Applying for example [103, Thm. C.1.2.5] with the notation (x, C, r) replaced by $(r, V, 1)$, we obtain the relation

$$V = \left\{r \in \mathbb{R}^q \,:\, \gamma_V(r) \leqslant 1\right\}.$$

Thus γ_V represents V; this first confirms that Definition 5.2.6 is consistent.

Another fundamental object is the *support function* of an arbitrary set $G \subset \mathbb{R}^q$, defined by

$$\mathbb{R}^q \ni r \mapsto \sigma_G(r) := \sup_{d \in G} d^\top r\,. \tag{5.17}$$

This function is easily seen to be sublinear, to grow when G grows, and to remain unchanged if G is replaced by its closed convex hull: $\sigma_G = \sigma_{\overline{\text{conv}}\,(G)}$. Besides, it is finite-valued if (and only if) G is bounded.

Conversely, every (finite-valued) sublinear function σ is the support function of a (bounded) closed convex set, unambiguously defined by

$$G = G_\sigma := \left\{d \in \mathbb{R}^q \,:\, d^\top r \leqslant \sigma(r) \text{ for all } r \in \mathbb{R}^q\right\} \tag{5.18}$$

(note: G_σ is closed and convex because it is an intersection of half-spaces; actually, G_σ is just the *subdifferential* of σ at 0). We then say that σ supports G_σ. The correspondence $\sigma \leftrightarrow G$ defines a one-to-one mapping between finite-valued sublinear functions and bounded closed convex sets (the mapping $\sigma \mapsto G$ of (5.18) extends to sublinear functions in $\mathbb{R} \cup \{+\infty\}$ but such an extension is not needed here).

Remark 5.3.1 (Primal-dual notation)**.** Equation (5.17) involves two variables, d and r, both written as column-vectors; nevertheless, they lie in two mutually dual spaces. In this chapter, we keep going back and forth between these two spaces; even though they are the same $\mathbb{R}^q$, we make a point to distinguish between the two. The notation $r, V, \dots$ [resp. $d, G, \dots$] will generally be used for primal elements [resp. for dual ones]. Most of the time, we will deal with support functions $\sigma_G(r)$ of dual sets; but we will also consider the support function $\sigma_V(d)$ of our primal neighborhood V. □

Being finite-valued sublinear, the gauge of V supports a compact convex set, obtained by replacing σ by γ_V in (5.18). Since $\gamma_V \geqslant 0$, we guess from positive homogeneity that this set is just the *polar* of V:

$$\begin{aligned} &\left\{d \in \mathbb{R}^q \,:\, d^\top r \leqslant \gamma_V(r) \text{ for all } r \in \mathbb{R}^q\right\} \;= \\ &\left\{d \in \mathbb{R}^q \,:\, d^\top r \leqslant 1 \text{ for all } r \in V\right\} \;=:\; V^\circ\,. \end{aligned} \tag{5.19}$$

Write (5.19) as $V^\circ = \left\{d \in \mathbb{R}^q \,:\, \sigma_V(d) \leqslant 1\right\}$ to see that σ_V represents V°; thus, the support function of V is the gauge of V°, so that the polar of V° is V itself: $(V^\circ)^\circ = V$. These various properties are rather classical, see for example [103, Prop. C.3.2.4, Cor. C.3.2.5], with (d, C, s) replaced by (r, V, d).

Now remember Example 5.2.7: V may have several representations. Any such representation ρ supports a set G_ρ and we will see that the polar of G_ρ is again V itself; G_ρ is a pre-image of V for the polarity mapping. We thus obtain a new concept: a *prepolar* of V is a set G such that

$$G^\circ := \left\{r \in \mathbb{R}^q \,:\, \sigma_G(r) \leqslant 1\right\} = V\,,$$

or equivalently σ_G represents V.

The property $(V^\circ)^\circ = V$ means that the standard polar V° is itself a prepolar – which is somewhat confusing; and it turns out to be the largest one (Corollary 5.3.3 below); or equivalently, its support function $\sigma_{V^\circ} = \gamma_V$ turns out to be the largest representation of V. The main result of this section states that V has also a smallest prepolar, or equivalently a smallest representation (Proposition 5.3.6 below); keeping Remark 5.1.4 in mind, this is exactly what we want. This result is actually [19, Thm. 1]; here we use elementary convex analysis and we insist more on the geometric aspect.

5.3.2 Largest representation

Introduce the recession cone V_∞ of V. Using the property $0 \in V$, it can be defined as

$$V_\infty = \{r \in \mathbb{R}^q \,:\, tr \in V \text{ for all } t > 0\} = \bigcap_{\lambda > 0} \lambda V\,,$$

and the second relation shows that V_∞ is closed; taking in particular $\lambda = 1$ shows that

$$V_\infty \subset V\,. \tag{5.20}$$

One then easily sees from (5.16) that $\gamma_V(r) = 0$ if $r \in V_\infty$. Yet, for any other representation ρ of V, (5.13) just imposes $\rho(r) \leqslant 0$ at this r and we may a priori have $\rho(r) < 0$: the possible representations of V may differ on V_∞; see Example 5.2.7 again. We make this more precise.

Lemma 5.3.2 (Representations and recession cone)**.** *For all representations ρ of the closed convex neighborhood V,*

$$\rho(r) \leqslant 0 \iff r \in V_\infty \quad \textit{and} \quad \rho(r) < 0 \implies r \in \operatorname{int}(V_\infty)\,.$$

Besides, all representations coincide on the complement of $\operatorname{int}(V_\infty)$ *in* $\mathbb{R}^q$.

Proof. By positive homogeneity, $\rho(r) \leqslant 0$ implies $\rho(tr) \leqslant 0 < 1$ (hence $tr \in V$) for all $t > 0$; this implies $r \in V_\infty$. Conversely, $\rho(r) > 0$ implies $\rho(tr) > 1$ for t large enough: using $0 \in V$ again, r cannot lie in V_∞.

To prove the second implication, invoke continuity of ρ: if $\rho(r) < 0$, ρ is still negative in a neighborhood of r, this neighborhood is contained in V_∞.

Besides, take a half-line emanating from 0 but not contained in V_∞; it certainly meets the boundary of V, at a point $\bar{r}$ which is unique (see, e.g., [103, Rem. A.2.1.7]). By (5.12), every representation ρ satisfies $\rho(\bar{r}) = 1$; and by positive homogeneity, the value of this representation is determined all along the half-line. In other words, all possible representations of V coincide on the complement W of V_∞; and by continuity, they coincide also on the closure of W, which is the complement of $\operatorname{int}(V_\infty)$. ∎

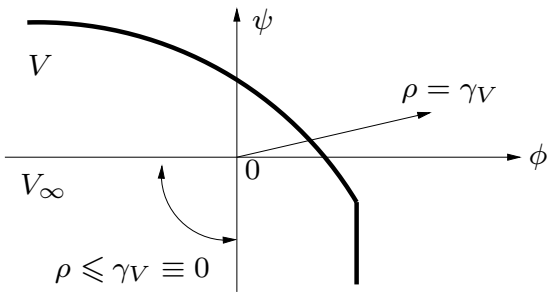


Figure 5.2: All representations coincide except in $\operatorname{int}(V_\infty)$

Figure 5.2 illustrates the difference between the recession cone (where the gauge is "maximal") and the rest of the space (where it is *the* representation). Altogether, the gauge appears as the largest representation:

Corollary 5.3.3 (Maximality of the gauge)**.** *All representations ρ of V satisfy $\rho \leqslant \gamma_V$, with equality on the complement of* $\operatorname{int}(V_\infty)$.

Geometrically, all prepolars G are contained in the polar of V:

$$G^\circ = V \quad \Longrightarrow \quad G \subset V^\circ\,.$$

In particular, V has a unique representation $\rho = \gamma_V$ (and a unique prepolar V°) whenever $\operatorname{int}(V_\infty) = \emptyset$.

Proof. Just apply Lemma 5.3.2, observing from (5.16) that the gauge is nonnegative.

Geometrically, the inequality between support functions becomes an inclusion: the set G supported by ρ is included in the set V° supported by γ_V (see, e.g., [103, Thm. C.3.3.1]). ∎

The next subsection will use the support function σ_V. It is positive on $\mathbb{R}^q \backslash \{0\}$, and even more: for some $\varepsilon > 0$, V contains the ball $B(\varepsilon)$ centered at 0 of radius ε, hence

$$\varepsilon \|d\| = \sigma_{B(\varepsilon)}(d) \leqslant \sigma_V(d) \quad \text{for all } d \in \mathbb{R}^q\,. \tag{5.21}$$

Then V° is bounded since the relation $\sigma_V(d) \leqslant 1$ implies $\|d\| \leqslant 1/\varepsilon$.

5.3.3 Smallest representation

The previous subsection dealt with polarity in the usual sense, viewing the gauge as a special representation. However, we are rather interested in *small* representations. Geometrically, we are interested in small prepolars, and the following definitions are indeed relevant:

$$\begin{cases} \widetilde{V}^\circ := \left\{ d \in V^\circ \,:\, d^\top r = \sigma_V(d) = 1 \text{ for some } r \in V \right\}, \\ \widehat{V}^\circ := \left\{ d \in V^\circ \,:\, \sigma_V(d) = 1 \right\}. \end{cases} \tag{5.22}$$

From (5.12), $\widehat{V}^\circ \neq \emptyset$ if V has a boundary, i.e. if $V \neq \mathbb{R}^q$. Obviously, $\widetilde{V}^\circ \subset \widehat{V}^\circ$. Besides, (5.21) implies that the two sets are bounded. There is a slight difference between the two, suggested by Figure 5.2 and specified on Figure 5.3, where the dashed line represents them both. We see that d_1 lies in $\widehat{V}^\circ$ but not in $\widetilde{V}^\circ$; and d_2 lies in both. On this example, $\widehat{V}^\circ$ is closed but Figure 5.5 will show that it need not be so. Although quite similar, we introduce the two sets for technical reasons, when proving that they have the same closed convex hull – which is our required smallest prepolar.

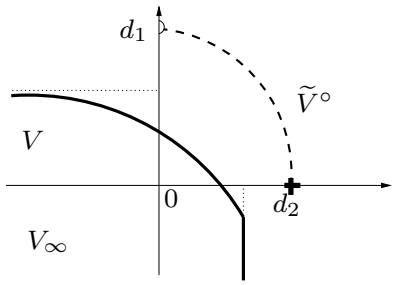


Figure 5.3: Activity in V°

Lemma 5.3.4. *The sets in* (5.22) *satisfy* $\widetilde{V}^\circ \subset \widehat{V}^\circ \subset \mathrm{cl}\left(\widetilde{V}^\circ\right)$. *It follows that* $\widehat{V}^\circ$ *and* $\widetilde{V}^\circ$ *have the same closed convex hull. In particular,* $\widetilde{V}^\circ \neq \emptyset$ *whenever* $\widehat{V}^\circ \neq \emptyset$.

Proof. The first inclusion is clear. To prove the second inclusion, recall two properties:
– the domain $\mathrm{dom}\,\partial\sigma_V$ of a subdifferential is dense in the domain $\mathrm{dom}\,\sigma_V$ of the function itself: see, e.g., [103, Thm. E.1.4.2];
– the subdifferential $\partial\sigma_V(d)$ is the face of V exposed by d: see, e.g., [103, Prop. C.3.1.4].
Thus, $d \notin \widetilde{V}^\circ$ implies $\partial\sigma_V(d) = \emptyset$; in other words, $\widetilde{V}^\circ \supset \mathrm{dom}\,\partial\sigma_V$. Taking closures,

$$\mathrm{cl}\,\widetilde{V}^\circ \supset \mathrm{cl}\left(\mathrm{dom}\,\partial\sigma_V\right) \supset \mathrm{dom}\,\sigma_V\,;$$

the required inclusion follows, since the last set obviously contains $\widehat{V}^\circ$.

It follows from the second inclusion that

$$\overline{\mathrm{conv}}\left(\widehat{V}^\circ\right) \subset \overline{\mathrm{conv}}\left(\mathrm{cl}\left(\widetilde{V}^\circ\right)\right).$$

On the other hand, the first inclusion implies that $\overline{\mathrm{conv}}\left(\widehat{V}^\circ\right)$ (a closed set) contains the closure of $\widetilde{V}^\circ$: $\mathrm{cl}\left(\widetilde{V}^\circ\right) \subset \overline{\mathrm{conv}}\left(\widehat{V}^\circ\right)$. This inclusion remains valid by taking the closed convex hulls:

$$\overline{\mathrm{conv}}\left(\mathrm{cl}\left(\widetilde{V}^\circ\right)\right) \subset \overline{\mathrm{conv}}\left(\widehat{V}^\circ\right);$$

the two sets coincide. The last statement is clear since the closure of the empty set is itself. ■

To help understand this construction, consider the polyhedral case, say $V = \mathrm{conv}\,\{p_i\}_i + \mathrm{cone}\,\{r_i\}_i$. Then the linear program defining $\sigma_V(d)$
– has no finite solution if some $d^\top r_i$ is positive, i.e. if $d \notin (V_\infty)^\circ$,
– is solved at some extreme point p_i otherwise.
In this situation, the two sets in (5.22) coincide and are closed; they are a union of hyperplanes of equation $d^\top p_i = 1$ (facets of V°), for p_i describing the extreme points of V. Besides, the polar V° is defined by

$$d^\top p_i \leqslant 1\,, \quad \text{and} \quad d^\top r_i \leqslant 0\,.$$

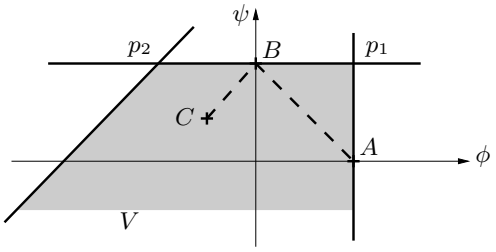


Figure 5.4: Constructing $\widetilde{V}^\circ$ or $\widehat{V}^\circ$

Example 5.3.5. For later use, we detail the calculation on a simple instance. Take for V the polyhedron of Figure 5.4, defined by the three inequalities

$$\phi \leqslant 1\,, \quad \psi \leqslant 1\,, \quad \psi \leqslant 2 + \phi\,;$$

here (ϕ, ψ) denotes a primal point in $\mathbb{R}^2$ we take row-vectors for typographical convenience). The two extreme points $p_1 = (1, 1)$ and $p_2 = (-1, 1)$ of V define the two segments (facets of V°) $[A, B]$ and $[B, C]$.

As for V°, it has first the two constraints $d^\top p_i \leqslant 1$ (yielding the above two segments). Besides, the two extreme rays $r_1 = (0, -1)$ and $r_2 = (-1, -1)$ of V_∞ make two more constraints $d^\top r_i \leqslant 0$, so that V° is the convex hull of A, B, C and 0. If V had a fourth constraint, say $\psi \geqslant -1$, then 0 would be moved down to $D = (0, -1)$ – and enter $\widetilde{V}^\circ$ and $\widehat{V}^\circ$. □

The closed convex hull thus revealed deserves a notation, as well as its support function: we set

$$V^\bullet := \overline{\mathrm{conv}}\,\big(\widetilde{V}^\circ\big) = \overline{\mathrm{conv}}\,\big(\widehat{V}^\circ\big) \quad \text{and} \quad \mu_V := \sigma_{V^\bullet} = \sigma_{\widetilde{V}^\circ} = \sigma_{\widehat{V}^\circ}\,. \tag{5.23}$$

For example in Figure 5.4, $V^\bullet$ is the triangle $\mathrm{conv}\{A, B, C\}$. In fact, the next result shows that μ_V is *the* small representation we are looking for. From now on, we assume $V \neq \mathbb{R}^q$, otherwise $V^\bullet = \emptyset$, $\mu_V \equiv -\infty$; a degenerate situation, which lacks interest anyway.

Proposition 5.3.6 (Smallest representation)**.** *Any ρ representing $V \neq \mathbb{R}^q$ satisfies $\rho \geqslant \mu_V$.*

Geometrically, $V^\bullet$ is the smallest closed convex set whose support function represents V.

Proof. Our assumption implies that neither $\widehat{V}^\circ$ nor $\widetilde{V}^\circ$ is empty (recall Lemma 5.3.4). Then take an arbitrary d in $\widetilde{V}^\circ$. We have to show that $d^\top r \leqslant \rho(r)$ for all $r \in \mathbb{R}^q$; this inequality will be transmitted to the supremum over d, which is $\mu_V(r)$.

Case 1. First let r be such that $\rho(r) > 0$. Then $\bar{r} := r/\rho(r)$ lies in V, so that $d^\top \bar{r} \leqslant \sigma_V(d) = 1$. In other words, $d^\top \bar{r} = \frac{d^\top r}{\rho(r)} \leqslant 1$, which is the required inequality.

Case 2. Let now r be such that $\rho(r) \leqslant 0$, so that $r \in V_\infty$ by Lemma 5.3.2. Since $d \in \widetilde{V}^\circ$, we can take $r_d \in V$ such that $d^\top r_d = 1$. Being exposed, r_d lies on the boundary of V: by (5.12), $\rho(r_d) = 1$.

By definition of the recession cone, $r_d + tr \in V$ for all $t > 0$ and, by continuity of ρ, $\rho(r_d + tr) > 0$ for t small enough. Apply Case 1:

$$d^\top r_d + t d^\top r = d^\top (r_d + tr) \leqslant \rho(r_d + tr) \leqslant \rho(r_d) + t\rho(r)\,,$$

where we have used sublinearity. This proves the required inequality since the first term is $1 + td^\top r$ and the last one is $1 + t\rho(r)$.

The geometric counterpart is proved just as in Corollary 5.3.3. ■

Thus, V does have a smallest representation, which is the support function of $V^\bullet$. Piecing together our results, we can now fully describe the polarity operation.

5.3.4 The set of prepolars

First of all, it is interesting to link the two extreme representations/prepolars introduced so far, and to confirm the intuition suggested by Figure 5.4:

Proposition 5.3.7. *Appending* 0 *to* $V^\bullet$ *gives the standard polar:*

$$\gamma_V = \max\{\mu_V, 0\} \quad \textit{i.e.} \quad V^\circ = \overline{\mathrm{conv}}\big(V^\bullet \cup \{0\}\big) = [0,1]V^\bullet\,.$$

Proof. For $r \in V_\infty$, $\gamma_V(r) = 0$, while $\mu_V(r) \leqslant 0$ (Proposition 5.3.6). For $r \notin V_\infty$, Lemma 5.3.2 gives $\gamma_V(r) = \mu_V(r) > 0$ because γ_V and μ_V are two particular representations.

Altogether, the first equality holds. Its geometric counterpart is [103, Thm. C.3.3.2]; and because $V^\bullet$ is convex compact, its closed convex hull with 0 is the sets of $\alpha d + (1-\alpha)0$ for $\alpha \in [0,1]$. ■

In summary, the set of representations – or of prepolars – is fully described as follows:

Theorem 5.3.8. *The representations of* V (*a closed convex neighborhood of the origin*) *are the finite-valued sublinear functions* ρ *satisfying*

$$\sigma_{V^\bullet} = \mu_V \leqslant \rho \leqslant \gamma_V = \sigma_{V^\circ} = \max\{0, \mu_V\}\,. \tag{5.24}$$

Geometrically, the prepolars of V*, i.e. the sets* G *whose support function represents* V*, are the sets sandwiched between the two extreme prepolars of* V*:*

$$G^\circ = V \quad \Longleftrightarrow \quad V^\bullet \subset \overline{\mathrm{conv}}\,(G) \subset V^\circ = \overline{\mathrm{conv}}\big(V^\bullet \cup \{0\}\big) = [0,1]V^\bullet\,.$$

Proof. In view of Corollary 5.3.3 and Propositions 5.3.6, 5.3.7, we just have to prove that a ρ satisfying (5.24) does represent V. Indeed, if $r \in V$ then $\rho(r) \leqslant \gamma_V(r) \leqslant 1$; if $r \notin V$, then $1 < \mu_V(r) \leqslant \rho(r)$. The geometric counterpart is again standard calculus with support functions. ■

We end this section with a deeper study of prepolars, which will be useful in the sequel. The next result introduces the polar cone $(V_\infty)^\circ$. When G is a cone, positive homogeneity can be used to replace the righthand side "1" in (5.19) by any positive number, or even by "0": in particular,

$$V_\infty^\circ := (V_\infty)^\circ = \{r \in \mathbb{R}^q \,:\, \sigma_{V_\infty}(r) \leqslant 0\}\,. \tag{5.25}$$

The notation V_∞° is used for simplicity, although it is somewhat informal; $(V_\infty)^\circ$ and $(V^\circ)_\infty$ differ, the latter is $\{0\}$ since V° is bounded.

Lemma 5.3.9 (Additional properties of prepolars)**.** *With the notation* (5.22)*,* (5.23)*,* (5.25)*,*

(i) V_∞° *is the closure of* $\operatorname{dom}\sigma_V$*,*

(ii) $\mathbb{R}_+\widehat{V}^\circ = \mathbb{R}_+V^\bullet = \mathbb{R}_+V^\circ = \operatorname{dom}\sigma_V$.

Proof. First of all, let $d \notin V_\infty^\circ$: there is $r \in V_\infty$ ($\mathbb{R}_+r \in V$) and $d^\top r > 0$; then $d^\top(tr) \to +\infty$ for $t \to +\infty$ and $\sigma_V(d)$ cannot be finite, i.e. $d \notin \operatorname{dom}\sigma_V$. Thus, $\operatorname{dom}\sigma_V \subset V_\infty^\circ$; hence $\operatorname{cl}(\operatorname{dom}\sigma_V) \subset V_\infty^\circ$ because V_∞° is closed.

To prove the converse inclusion, take $r \notin (\operatorname{dom}\sigma_V)^\circ$: there is d such that $\sigma_V(d) < +\infty$ and $d^\top r > 0$. Then $d^\top(tr) \to +\infty$ when $t \to +\infty$; if r were in V_∞, then tr would lie in V and $\sigma_V(d)$ would be $+\infty$, a contradiction. Thus we have proved $V_\infty \subset (\operatorname{dom}\sigma_V)^\circ$. Taking polars and knowing that $\operatorname{dom}\sigma_V$ is a cone, $V_\infty^\circ \supset (\operatorname{dom}\sigma_V)^{\circ\circ} = \operatorname{cl}(\operatorname{dom}\sigma_V)$ (see [103, Prop. A.4.2.6]). This proves (i).

To prove (ii), observe first that $\widehat{V}^\circ \subset V^\bullet \subset V^\circ \subset \operatorname{dom}\sigma_V$; and because $\operatorname{dom}\sigma_V$ is a cone,

$$\mathbb{R}_+\widehat{V}^\circ \subset \mathbb{R}_+V^\bullet \subset \mathbb{R}_+V^\circ \subset \operatorname{dom}\sigma_V\,. \tag{5.26}$$

On the other hand, take $0 \neq d \in \operatorname{dom}\sigma_V$, so that $\sigma_V(d) > 0$ by (5.21) and $\frac{1}{\sigma_V(d)}d \in \widehat{V}^\circ$: $d \in \mathbb{R}_+\widehat{V}^\circ$. Since 0 also lies in $\mathbb{R}_+\widehat{V}^\circ$, we do have $\operatorname{dom}\sigma_V \subset \mathbb{R}_+\widehat{V}^\circ$; (5.26) is actually a chain of equalities. To complete the proof, observe from Proposition 5.3.7 that $\mathbb{R}_+V^\circ = \mathbb{R}_+V^\bullet$. ■

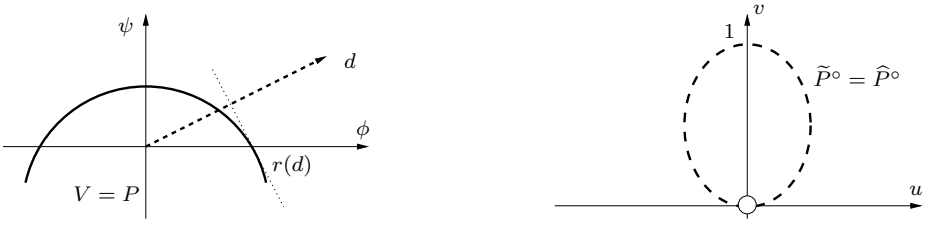


Figure 5.5: Trouble appears if the neighborhood has no asymptote

Beware that really pathological prepolars can exist, Figure 5.5 illustrates a well-known situation. Its left part displays the parabolic neighborhood $V = P \subset \mathbb{R}^2$ defined by the constraint $\psi \leqslant 1 - \frac{1}{2}\phi^2$. A direction $d = (u, v)$ with $v > 0$ exposes the point $r(d)$. When $v \downarrow 0$, the component of $r(d)$ along d (namely ϕ) goes to $+\infty$, which does bring trouble. Computing $r(d)$ is an exercise resulting in

$$\sigma_P(d) = \sigma_P(u,v) = \begin{cases} 0 & \text{if } d = 0\,, \\ v + \frac{u^2}{2v} & \text{if } v > 0\,, \\ +\infty & \text{if } v \leqslant 0\,; \end{cases} \tag{5.27}$$

two phenomena are then revealed.

– First, $\widehat{V}^\circ$ is defined by the equation

$$v + \frac{u^2}{2v} = 1\,, \quad \text{i.e.} \quad 2(v^2 - v) + u^2 = 0\,.$$

This is an ellipse passing through the origin (right part of Figure 5.5); yet 0 cannot lie in $\widehat{V}^\circ$, since $\sigma_P(0) = 0 \neq 1$. Thus, $\widehat{P}^\circ$ is not closed and, more importantly, $0 \in P^\bullet$.

– The second phenomenon is a violent discontinuity of σ_P at 0. In fact, fix $\alpha > 0$ and let $d_k = \left(\frac{\alpha}{k}, \frac{1}{k^2}\right)$; then $d_k \to 0$, while $\sigma_P(d_k) \to \frac{\alpha^2}{2}$, an arbitrary positive number.

Both phenomena are due to (local) unboundedness of σ_P on its domain, which is thus not closed; if $(u_k, v_k) \in \operatorname{dom}\sigma_P$ tends to any $(u, 0)$ with $u \neq 0$, then $\sigma_P(u_k, v_k) \to +\infty$. Ruling out such a behaviour brings additional useful properties:

Corollary 5.3.10 (Safe prepolars)**.** *If* $0 \notin V^\bullet$*, then*

$$\mathbb{R}_+\widehat{V}^\circ = \mathbb{R}_+V^\bullet = \mathbb{R}_+V^\circ = \operatorname{dom}\sigma_V = V_\infty^\circ \tag{5.28}$$

and $\operatorname{int} V_\infty \neq \emptyset$ (*the polar* V_∞° *is a so-called pointed cone*)*.*

Proof. When $0 \notin V^\bullet$, $\mathbb{R}_+V^\bullet$ is closed ([103, Prop. A.1.4.7]). Then apply Lemma 5.3.9: by (ii) $\operatorname{dom}\sigma_V$ is closed and (5.28) follows from (i).

Now we separate 0 from $V^\bullet$: there is some r such that $\sigma_{V^\bullet}(r) < 0$. By continuity of the finite-valued convex function $\sigma_{V^\bullet}$, this inequality is still valid in a neighborhood of r: $\sigma_{V^\bullet} \leqslant 0$ over some nonzero ball B around r. By Lemma 5.3.9(ii),

$$\sigma_{V_\infty^\circ}(d) = \sigma_{\mathbb{R}_+V^\bullet}(d) = \sup_{t\geqslant 0}\sup_{d\in V^\bullet} td^\top r = \sup_{t\geqslant 0} t\sigma_{V^\bullet}(d)\,,$$

so that $\sigma_{V_\infty^\circ}$ enjoys the same property: by (5.25), B is contained in $(V_\infty^\circ)^\circ$. Proposition A.4.2.6 of [103] finishes the proof. ■

Property (5.28) means closedness of $\operatorname{dom}\sigma_V$ and is rather instrumental. We mention another simple assumption implying it:

Proposition 5.3.11. *If* $V = U + V_\infty$*, where* U *is bounded, then* $\operatorname{dom}\sigma_V = V_\infty^\circ$*.*

Proof. The support function of a sum is easily seen to be the sum of support functions: $\sigma_V = \sigma_U + \sigma_{V_\infty}$. Every $d \in V_\infty^\circ$ then satisfies $\sigma_V(d) = \sigma_U(d)$, a finite number when U is bounded. ■

Let us put this section in perspective. The traditional gauge theory defines via (5.16), (5.19) the polarity correspondence $V \leftrightarrow V^\circ$ for *compact* convex neighborhoods of the origin. We generalize it to *unbounded* neighborhoods, whose standard gauge is replaced via Definition 5.2.6 by their family of representations. Each representation ρ, which may assume negative values, gives rise to $\partial\rho(0)$ – which we call a prepolar of V. Theorem 5.3.8 establishes the existence of a largest element (the usual polar V°) and of a smallest element ($V^\bullet$) in the family of (closed convex) prepolars of V. Gauge theory is further generalized in [184], in which 0 may lie on the boundary of V. Our stricter framework allows a finer analysis of the smallest prepolar; in particular, the property $0 \notin V^\bullet$ helps avoiding nasty phenomena.

5.4 Minimal CGF's, maximal S-free sets

Remembering Remark 5.1.4, our goal in this section is to study the concept of minimality for CGF's. Geometrically, we study the concept of maximality for S-free sets. In fact, the two concepts are in correpondence via (5.14); but a difficulty arises because the reverse inclusion does not hold in (5.14). As a result, several definitions of minimality and maximality are needed.

5.4.1 Minimality, maximality

In our quest for small CGF's, the following definition is natural.

Definition 5.4.1 (Minimality)**.** A CGF ρ is called minimal if the only possible CGF $\rho' \leqslant \rho$ is ρ itself. □

Knowing that a CGF ρ represents $V(\rho)$ and that $\mu_{V(\rho)} \leqslant \rho$ represents the same set, a minimal CGF is certainly a smallest representation:

$$\rho \text{ is a minimal CGF} \quad \Longrightarrow \quad \rho = \mu_{V(\rho)} = \sigma_{V(\rho)^\bullet} \,. \tag{5.29}$$

In addition, $V(\rho)$ must of course be a special S-free set when ρ is minimal. Take for example $S = \{1\} \subset \mathbb{R}$, $V = [-1, +1]$; then $\rho(r) := |r|$ is the smallest (because unique) representation of V but is not minimal: $\rho'(r) := \max\{0, r\}$ is also a CGF, representing $V' =]-\infty, +1]$.

From (5.14), a smaller ρ describes a larger V; so Definition 5.4.1 has its geometrical counterpart:

Definition 5.4.2 (Maximality)**.** An S-free set V is called *maximal* if the only possible S-free set $V' \supset V$ is V itself. □

The two objects are indeed related:

Proposition 5.4.3. *If V is a maximal S-free set, then its smallest representation μ_V is a minimal* CGF.

Proof. Take a CGF ρ', representing the S-free set $V' = V(\rho')$. If $\rho' \leqslant \mu_V$, then $V' \supset V$; and if V is maximal, $V' = V$. Then $\rho' \geqslant \mu_V = \mu_{V'}$ by Proposition 5.3.6. ■

Besides, these objects do exist:

Theorem 5.4.4. *Every S-free set is contained in a maximal S-free set. It follows that there exists a maximal S-free set and a minimal* CGF.

Proof. Let V be an S-free set. In the partially ordered family $(\mathbb{F}, \subset)$ of all S-free sets containing V, let $\{W_i\}_{i\in I}$ be a totally ordered subfamily (a chain) and define $W := \cup_{i\in I} W_i$. Clearly, W is a neighborhood of the origin; its convexity is easily established, let us show that its closure is S-free.

Remember from [103, Thm. C.3.3.2(iii)] that the support function of a union is the (closure of the) supremum of the support functions:

$$\sigma_{\operatorname{int}(W)} = \sigma_W = \operatorname{cl}\left(\sup_{i\in I} \sigma_{W_i}\right) = \operatorname{cl}\left(\sup_{i\in I} \sigma_{\operatorname{int}(W_i)}\right) = \sigma_{\cup_i \operatorname{int}(W_i)} \,.$$

Having the same support function, the two open convex sets $\operatorname{int}(W)$ and $\cup_i \operatorname{int}(W_i)$ coincide: $r \in \operatorname{int}(W)$ means $r \in \operatorname{int}(W_i)$ for some i; because W_i is S-free, $r \notin S$ and our claim is proved. Thus, the chain $\{W_i\}$ has an upper bound in $\mathbb{F}$; in view of Zorn's lemma, $\mathbb{F}$ has a maximal element.

Now (5.1b) implies that a ball centered at 0 with a small enough radius is S-free; and there exists a maximal S-free set containing it. Proposition 5.4.3 finishes the proof. ■

The maximal S-free sets can be explicitly described for some special S's: $\mathbb{Z}^q$ [127], the intersection of $\mathbb{Z}^q$ with an affine subspace [17], with a rational polyhedron [18], or with an arbitrary closed convex set [14, 134]. Unfortunately, the "duality" between minimal CGF's and maximal S-free sets is deceiving, as the two definitions do not match: the set represented by a minimal CGF need not be maximal. In fact, when ρ is linear, the property introduced in Definition 5.4.1 holds vacuously: no sublinear function can properly lie below a linear function. Thus, a linear CGF ρ is always minimal;

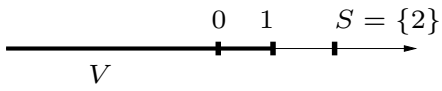


Figure 5.6: A linear CGF is always maximal

yet, a linear ρ represents a neighborhood $V(\rho)$ (a half-space) which is S-free but has no reason to be maximal. See Figure 5.6: with $n = 1$, the set $V =]-\infty, 1]$ (represented by $\rho(x) = x$) is $\{2\}$-free but is obviously not maximal.

A more elaborate example reveals the profound reason underlying the trouble: for an S-free set W containing V, μ_W need not be comparable to μ_V.

Example 5.4.5. In Example 5.3.5, take for S the union of the three lines with respective equations

$$\phi = 1\,, \quad \psi = 1\,, \quad \psi = 2 + \phi\,,$$

so that V is clearly maximal S-free.

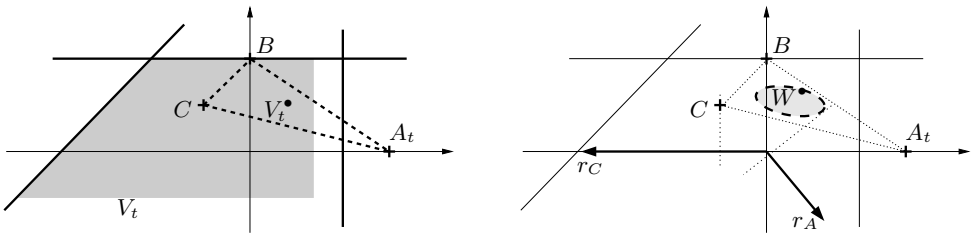


Figure 5.7: The mapping $V \mapsto V^\bullet$ is not monotonic

Now shrink V to V_t (left part of Figure 5.7) by moving its right vertical boundary to $\phi \leqslant 1 - t$. Then A is moved to $A_t = \left(\frac{1}{1-t}, 0\right)$; there is no inclusion between the new $V_t^\bullet = \mathrm{conv}\{A_t, B, C\}$ and the original $V^\bullet = \mathrm{conv}\{A, B, C\}$; this is the key to our example.

Let us show that μ_{V_t} is minimal, even though V_t is not maximal. Take for this a CGF $\rho \leqslant \mu_{V_t}$, which represents an S-free set W; by (5.14), $W \supset V_t$. We therefore have

$$\sigma_{W^\bullet} = \mu_W \leqslant \rho \leqslant \mu_{V_t} = \sigma_{V_t^\bullet}\,, \quad \text{i.e.,} \quad W^\bullet \subset V_t^\bullet$$

and we proceed to show that equality does hold, i.e. the three extreme points of $V_t^\bullet$ do lie in $W^\bullet$.

– If $A_t \notin W^\bullet$, the right part of Figure 5.7 shows that $W^\bullet$ is included in the open upper half-space. Knowing that

$$W = \left(W^\bullet\right)^\circ = \left\{r \,:\, d^\top r \leqslant 1 \text{ for all } d \in W^\bullet\right\},$$

this implies that W_∞ has a vector of the form $r_A = (\varepsilon, -1)$ ($\varepsilon > 0$); W cannot be S-free.

– If $C \notin W^\bullet$, there is $r_C \in \mathbb{R}^2$ such that $C^\top r_C > \sigma_{W^\bullet}(r_C) = \mu_W(r_C)$ (we denote also by C the 2-vector representing C). For example $r_C = (-2, 0) \in \mathrm{bd}\,(V)$ (see the right part of Figure 5.7), so that

$$C^\top r_C = 1 > \sigma_{W^\bullet}(-2, 0) = \mu_W(-2, 0)\,.$$

By continuity, $\mu_W(-2 - \varepsilon, 0) \leqslant 1$ for $\varepsilon > 0$ small enough. Since μ_W represents W, this implies that $(-2 - \varepsilon, 0) \in W$; W (which contains V_t) is not S-free.

– By the same token, we prove that $B \in W^\bullet$ (the separator $r_B = (0,1) \in \mathrm{bd}\,(V)$ does the job).

We have therefore proved that $W^\bullet = V_t^\bullet$, i.e $\mu_W = \mu_{V_t}$, i.e. μ_{V_t} is minimal. □

The next section makes a first step toward a theory relating small CGF's and large S-free sets.

5.4.2 Strong minimality, asymptotic maximality

First, let us give a name to those minimal CGF's corresponding to maximal S-free sets.

Definition 5.4.6 (Strongly minimal CGF)**.** A CGF ρ is called strongly minimal if it is the smallest representation of a maximal S-free set.

The strongly minimal CGF's can be characterized without any reference to the geometric space.

Proposition 5.4.7. *A* CGF ρ *is strongly minimal if and only if, for every* CGF ρ'*,*

$$\rho' \leqslant \max\{0,\rho\} \left[= \gamma_{V(\rho)} = \sigma_{V(\rho)^\circ}\right] \quad \Longrightarrow \quad \rho' \geqslant \rho\,. \tag{5.30}$$

Proof. Take first a maximal V. Every CGF $\rho' \leqslant \gamma_V$ represents an S-free set V', which contains V – see (5.13) – so that $V' = V$ by maximality, i.e. ρ' represents V as well; hence $\rho' \geqslant \mu_V$ by Proposition 5.3.6. Thus, $\rho(= \mu_V)$ satisfies (5.30).

Let now ρ satisfy (5.30), we have to show that $V := V(\rho)$ is maximal. Taking in particular $\rho' = \mu_V$ in (5.30) shows that ρ must equal μ_V. Let $V' \supset V$ be S-free; we have $(V')^\circ \subset V^\circ$, i.e.

$$\gamma_{V'} = \sigma_{(V')^\circ} \leqslant \sigma_{V^\circ} = \gamma_V = \max\{0,\rho\}\,.$$

Now $\rho' := \gamma_{V'}$ is a CGF, so $\rho' \geqslant \rho = \mu_V$ by (5.30); by Theorem 5.3.8, ρ' represents not only V' but also V, i.e. $V' = V$: V is maximal. ■

In Section 5.3 we have systematically developed the geometric counterpart of representations; this exercise can be continued here. In fact, the concept of minimality involves two properties from a sublinear function:

– it must be the *smallest* representation of some neighborhood V – remember (5.29),

– this neighborhood must enjoy some maximality property.

In view of the first property, a CGF can be imposed to be not only sublinear but also to support a set that is a *smallest* prepolar. Then Definition 5.4.1 has a geometric counterpart: minimality of $\rho = \mu_V = \sigma_{V^\bullet}$ means

$$\begin{array}{ccccc} G' \subset V^\bullet & \text{and} & (G')^\circ \text{ is } S\text{-free} & \Longrightarrow & G' = V^\bullet,\ \text{i.e. } (G')^\circ = V\,. \\ [\rho' = \sigma_{G'} \leqslant \rho] & & [\rho' \text{ is a CGF}] & & [\rho' = \rho] \end{array}$$

Likewise for Definition 5.4.6: strong minimality of $\rho = \gamma_V = \sigma_{V^\circ}$ means

$$\begin{array}{ccccc} G' \subset V^\circ & \text{and} & (G')^\circ \text{ is } S\text{-free} & \Longrightarrow & G' \supset V^\bullet,\ \text{i.e. } (G')^\circ \subset V\,. \\ [\rho' = \sigma_{G'} \leqslant \gamma_V] & & [\rho' \text{ is a CGF}] & & [\rho' \geqslant \rho] \end{array}$$

These observations allow some more insight into the $(\cdot)^\bullet$ operation:

Proposition 5.4.8. *Let $\rho = \mu_V = \sigma_{V^\bullet}$ be a minimal* CGF*. If an S-free set W satisfies $W^\bullet \subset V^\bullet$, then $W = V$.*

Proof. The smallest representation $\rho' := \mu_W = \sigma_{W^\bullet}$ of the S-free set W is a CGF; and from monotonicity of the support operation, $\rho' \leqslant \rho$. Then minimality of ρ implies $\rho' = \rho$, i.e. $W^\bullet = V^\bullet$, an equality transmitted to the polars: $W = (W^\bullet)^\circ = (V^\bullet)^\circ = V$. ■

This result confirms that non-equivalence between minimal CGF's and maximal S-free sets comes from non-monotonicity of the mapping $V \mapsto V^\bullet$ – or of $V \mapsto \mu_V$. To construct Example 5.4.5, we do need a $W \supset V$ such that $W^\bullet \not\subset V^\bullet$.

Then comes a natural question: how maximal are the S-free sets represented by minimal CGF's? For this, we introduce one more concept:

Definition 5.4.9. An S-free set V is called *asymptotically maximal* if every S-free set $V' \supset V$ satisfies $V'_\infty = V_\infty$.

It allows a partial answer to the question.

Theorem 5.4.10 (Minimal ⇒ asymptotically maximal)**.** *The S-free set represented by a minimal* CGF *is asymptotically maximal.*

Proof. Let μ_V be a minimal CGF and take an S-free set $V' \supset V$. Introduce the set $G := V^\bullet \cap \big(V'_\infty\big)^\circ$. Inclusions translate to inequalities between support functions:

$$\sigma_G \leqslant \sigma_{V^\bullet} = \mu_V \tag{5.31}$$

and we proceed to prove that this is actually an equality. Let us compute the set $W := G^\circ$ represented by σ_G. The support function of an intersection is obtained via an inf-convolution (formula (3.3.1) in [103, Chap. C)] for example): $\sigma_G(\cdot)$ is the closure of the function

$$r \mapsto \inf\big\{\sigma_{V^\bullet}(r_1) + \sigma_{(V'_\infty)^\circ}(r_2) \,:\, r_1 + r_2 = r\big\}.$$

In this formula, $\sigma_{V^\bullet} = \mu_V$ and the support function of the closed convex cone $\big(V'_\infty\big)^\circ$ is the indicator of its polar V'_∞: the above function is

$$r \mapsto \inf\big\{\mu_V(r_1) \,:\, r_1 + r_2 = r,\, r_2 \in V'_\infty\big\}\,.$$

Now use (5.12): because σ_G represents W, to say that $r \in \operatorname{int}(W)$ is to say that the above infimum is strictly smaller than 1, i.e. that there are r_1, r_2 such that

$$r_1 + r_2 = r,\ r_2 \in V'_\infty,\ \mu_V(r_1) < 1 \qquad \text{i.e.} \qquad r_1 + r_2 = r,\ r_2 \in V'_\infty,\ r_1 \in \operatorname{int} V\,.$$

In a word:

$$\operatorname{int}(W) = V'_\infty + \operatorname{int}(V) \supset \operatorname{int}(V) \ni 0\,,$$

where we have used the property $0 \in V'_\infty$. Remembering the inclusion $V \subset V'$ and the definition of a recession cone, we also have

$$\operatorname{int}(W) = V'_\infty + \operatorname{int}(V) \subset V'_\infty + \operatorname{int}(V') \subset V'_\infty + V' \subset V'\,.$$

Altogether, $0 \in \operatorname{int}(W) \subset \operatorname{int}(V')$. As a result, $W(= G^\circ)$ is an S-free closed convex neighborhood of the origin: its representation σ_G is a CGF and minimality of $\mu_V = \sigma_{V^\bullet}$ implies with (5.31) that $\sigma_G = \sigma_{V^\bullet}$.

By closed convexity of both sets $V^\bullet$ and $G = V^\bullet \cap \big(V'_\infty\big)^\circ$, this just means $G = V^\bullet$, i.e. $\big(V'_\infty\big)^\circ \supset V^\bullet$. By polarity, $V'_\infty \subset \big(V^\bullet\big)^\circ = V$ (invoke Theorem 5.3.8). The cone V'_∞, contained in the neighborhood V, is also contained in its recession cone: $V'_\infty \subset V_\infty$. Since the converse inclusion is clear from $V' \supset V$, we have proved $V'_\infty = V_\infty$: V is asymptotically maximal. ■

5.5 Favourable cases

Despite Example 5.4.5, a number of papers have established the equivalence between maximal S-free sets and minimal CGF's, for various forms of S. This equivalence is indeed known to hold in a number of situations:

(a) when S is a finite set of points in $\mathbb{Z}^q - b$; see [106] and more recently [68];

(b) when S is the intersection of $\mathbb{Z}^n$ with an affine space; this was considered in [38] and [17];

(c) when $S = P \cap (\mathbb{Z}^q - b)$ for some rational polyhedron P; this was considered in [18, 68].

Accordingly, we investigate in this section the question: when does minimality imply strong minimality? So we consider an S-free set V, whose smallest representation $\mu_V = \sigma_V \bullet$ is minimal, hence V is asymptoticaly maximal (Theorem 5.4.10); we want to exhibit conditions under which V is maximal. We denote by $L = (-V_\infty) \cap V_\infty$ the *lineality space* of V (the largest subspace contained in the closed convex cone V_∞) and our result is the following.

Theorem 5.5.1. *Suppose $0 \in \overline{S} := \overline{\text{conv}}\,(S)$. A minimal μ_V is strongly minimal whenever one of the following two properties (i) and (ii) holds:*

(i) $V_\infty \cap \overline{S}_\infty = \{0\}$ *(in particular S bounded),*

(ii) $(ii)_1$ $\overline{S} = U + \overline{S}_\infty$ *with U bounded, and*
$(ii)_2$ $V_\infty \cap \overline{S}_\infty = L \cap \overline{S}_\infty$.

This theorem generalizes the above-mentioned results: (i) is a weakening of (a) and (ii) weakens (b) or (c). Note that $(ii)_2$ generalizes (i) (to an unbounded $V_\infty \cap \overline{S}$); the price to pay is assumption $(ii)_1$, whose role is to exclude an asymptotic behaviour of $\overline{S}$ similar to that of P in Figure 5.5 (see Proposition 5.3.11).

However, the interesting point does not lie in the above assumptions (a) – (ii). Recalling that the whole issue lies in unboundedness of V, our proof of Theorem 5.5.1 uses Theorem 5.4.10 as follows. Starting from an S-free set V which is asymptotically maximal but not maximal, we construct a sequence of neighborhoods V^k satisfying $V^k_\infty \supsetneq V_\infty$. Then V^k is not S-free: there is some $r^k \in S \cap \text{int}\,(V^k)$; see Figure 5.8.

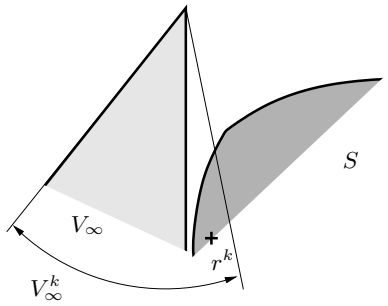


Figure 5.8: Constructing in S an unbounded sequence "tending to" V

Besides, our construction is organized in such a way that V^k "tends to" V and, by non-maximality of V, r^k is unbounded but "tends to" V. More precisely,

$$\text{the cluster points of the normalized sequence } \{r^k\} \text{ lie in } \overline{S}_\infty \cap V_\infty.$$

Decomposing $r^k = \ell^k + u^k$ along L and $L^\perp$, we also prove that u^k is unbounded but "tends to" $\overline{S} \cap L^\perp$, more precisely

the cluster points of the normalized sequence $\{u^k\}$ lie in $\overline{S}_\infty \cap L^\perp$.

We believe that these are key properties of non-maximal S-free sets. Having established them, the whole business is to find appropriate assumptions under which existence of our unbounded sequences is impossible; (a) – (ii) above are such *ad hoc* assumptions.

Obtaining r^k and u^k is a fairly complicate operation, which we divide into a series of lemmas. For a reason that will appear in (5.39) below, we may assume $0 \notin V^\bullet$. Then we enlarge V to V^k by chopping off a bit of $V^\bullet$ as follows. Take an extreme ray $\mathbb{R}_+ d_V$ of V_∞°. By (5.28), its intersection with $V^\bullet$ is a nonempty segment $[d_V, t_V d_V]$, with $1 \leqslant t_V < +\infty$. Given a positive integer k, we introduce the open neighborhood of $[d_V, t_V d_V]$:

$$N^k := [d_V, t_V d_V] + B\Big(0, \frac{1}{k}\Big) = \bigcup_{1 \leqslant t \leqslant t_V} B\Big(t d_V, \frac{1}{k}\Big)\,, \tag{5.32}$$

where $B(d, \delta)$ is the open ball of center d and radius δ. We remove N^k from $V^\bullet$, thus obtaining a set C, closed hence compact; its convex hull

$$G^k := \operatorname{conv} C\,, \quad \text{with} \quad C := V^\bullet \backslash N^k = \Big\{ d \in V^\bullet \,:\, \|d - t d_V\| \geqslant \frac{1}{k} \text{ for all } t \in [1, t_V] \Big\} \tag{5.33}$$

is convex compact. Figure 5.9 illustrates our construction.

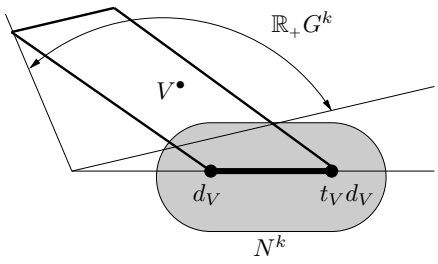


Figure 5.9: Chopping off $V^\bullet$ near an extreme ray

Note for future use that the distance from every $d \in [d_V, t_V d_V]$ to C does not exceed $1/k$; and the same holds for $G^k \supset C$. Formally:

$$\forall \bar{d} \in [d_V, t_V d_V]\,,\ \exists d_k \in G^k \ \text{ such that } \ \|d_k - \bar{d}\| \leqslant \frac{1}{k}\,. \tag{5.34}$$

Remark 5.5.2. The above construction would become substantially simpler, and N^k would reduce to the open ball $B\big(d_V, \frac{1}{k}\big)$, if $V^\bullet \cap \mathbb{R}_+ d_V$ reduced to a singleton, i.e. if $t_V = 1$; but this property need not hold when σ_V is not continuous.

To make a counterexample, start from the parabola of Figure 5.5. We already know that $\sigma_P(d_k)$ can tend to any nonnegative value when $d_k \to 0$. However $0 \in P^\bullet$; alternatively, the domain of σ_P is not closed. In fact, we need a discontinuous sublinear function which is locally bounded on its domain – and this requires three variables. Thus, we first bound σ_P by defining

$$f(d) := 1 + \begin{cases} \sigma_P(d) & \text{if } \sigma_P(d) \leqslant 1\,, \\ +\infty & \text{otherwise} \end{cases}$$

(the "1+" appearing above is just aimed at getting 0 in the interior of V). Although no longer positively homogeneous, f is still convex, its domain is the compact convex set $P^\bullet$, on which $1 \leqslant f \leqslant 2$;

when $d_k \in P^\bullet$ tends to 0, $f(d_k)$ can tend to any value in $[1,2]$. To complete the construction, we take the so-called *perspective* of f:

$$\mathbb{R}^2 \times \mathbb{R} \ni (d,w) \mapsto \sigma(d,w) := \begin{cases} w f\Big(\frac{d}{w}\Big) & \text{if } w > 0\,, \\ 0 & \text{if } (d,w) = (0,0)\,, \\ +\infty & \text{otherwise} \end{cases}$$

whose positive homogeneity is clear. Actually, σ is known to be convex and to support a closed convex set V; see [103, § B.2.2] (in particular Remark 2.2.3), where our (d,w) is called (x,u). Besides, the property $f \geqslant 1$ implies that V is a neighborhood of the origin; remember (5.21).

Now take $(d,w) \in \widehat{V}^\circ \subset \operatorname{dom}\sigma$, so that $d' := \big(\frac{d}{w}\big) \in \operatorname{dom} f$ and $w > 0$. Then use positive homogeneity:

$$1 = \sigma(d,w) \quad \Longrightarrow \quad \frac{1}{w} = \sigma(d',1) = f(d') \in [1,2] \quad \Longrightarrow \quad w \geqslant \frac{1}{2}\,.$$

Thus, $\widehat{V}^\circ$ is separated from the origin (by the hyperplane $w \geqslant \frac{1}{2}$) and this property is transmitted to its closed convex hull $V^\bullet$. On the other hand, σ inherits the discontinuities of f. In fact, choose $\alpha \in [1,2]$ and construct a sequence $\{d_k\}$ in $\operatorname{dom} f$ tending to 0, such that $f(d_k) \to \alpha$. Since $\sigma(d_k,1) = f(d_k) > 0$, positive homogeneity gives

$$\sigma\Big(\frac{d_k}{f(d_k)}, \frac{1}{f(d_k)}\Big) = 1\,, \quad \text{hence} \quad \Big(\frac{d_k}{f(d_k)}, \frac{1}{f(d_k)}\Big) \in \widehat{V}^\circ\,.$$

Pass to the limit:

$$\Big(\frac{d_k}{f(d_k)}, \frac{1}{f(d_k)}\Big) \to \Big(0, \frac{1}{\alpha}\Big) \in \operatorname{cl}\widehat{V}^\circ \subset V^\bullet\,.$$

Since α was arbitrary in $[1,2]$, the intersection of $V^\bullet$ with the ray $\{0\} \times \mathbb{R}_+$ contains the whole segment $\{0\} \times [\frac{1}{2}, 1]$. □

Viewing G^k of (5.33) as a prepolar, we set

$$V^k := (G^k)^\circ\,.$$

Of course, $V^\bullet \supset G^{k+1} \supset G^k$ and $V \subset V^{k+1} \subset V^k$. The closed convex neighborhood V^k enjoys all of the properties listed in Section 5.3, in particular those coming from $0 \notin G^k$.

Lemma 5.5.3 (Enlarging V_∞)**.** *Assume $0 \notin V^\bullet$; let $\mathbb{R}_+ d_V$ be an extreme ray of V_∞° and assume that $\mathbb{R}_+ d_V \subsetneq V_\infty^\circ$ ($\mathbb{R}_+ d_V$ is properly contained in V_∞°). Given an integer $k > 0$, construct N^k, G^k, V^k as above. Then $G^k \neq \emptyset$ for k large enough (say $k \geqslant k_0$) and*

(i) *$V_\infty \subsetneq V_\infty^k$ for $k \geqslant k_0$,*

(ii) *$\cap_{k \geqslant k_0} V^k = V$.*

Proof. If G^k were empty for all k, we would have $V^\bullet \subset N^k$ for all k, hence $V^\bullet$ would reduce to $[d_V, t_V d_V]$. In view of (5.28), this would imply $\mathbb{R}_+ d_V = V_\infty^\circ$, which our assumption rules out.

Every $d \in G^k$ is a convex combination $\sum_i \alpha_i d_i$ with each d_i in $V^\bullet \backslash N^k \subset V_\infty^\circ$. None of these d_i's can lie in $[d_V, t_V d_V] \subset N^k$, and none of their convex combinations either because of extremality of $\mathbb{R}_+ d_V$. We conclude that

$$G^k \cap [d_V, t_V d_V] = \emptyset\,. \tag{5.35}$$

Now, we see from Theorem 5.3.8 that

$$\mathbb{R}_+ (V^k)^\bullet \subset \mathbb{R}_+ G^k \subset \mathbb{R}_+ (V^k)^\circ\,;$$

but from Proposition 5.3.7, this is actually a chain of equalities:

$$\mathbb{R}_+(V^k)^\bullet = \mathbb{R}_+ G^k\,. \tag{5.36}$$

Besides, $(V^k)^\bullet \subset\ G^k \subset V^\bullet$, hence $0 \notin (V^k)^\bullet$ and we can apply (5.28) to V^k. Then we write

$$\begin{array}{rcll} \left(V^k_\infty\right)^\circ & = & \mathbb{R}_+(V^k)^\bullet & \text{[(5.28)]} \\ & = & \mathbb{R}_+ G^k & \text{[(5.36)]} \\ & \subsetneq & \mathbb{R}_+ V^\bullet & \text{[consequence of (5.35)]} \\ & = & V^\circ_\infty\,. & \text{[(5.28) again]} \end{array}$$

Thus, $\left(V^k_\infty\right)^\circ \subsetneq V^\circ_\infty$, which implies (i) since polarity is an involution between closed convex cones.

To prove (ii), take $\bar r$ in $\cap_k V^k$; we have to prove that $\bar r \in V$ (the other inclusion being obvious). If $\bar r \notin V$ there is a separating hyperplane $\bar d$: $\sigma_V(\bar d) < \bar d^\top \bar r$. Normalizing $\bar d$ via (5.28), we have altogether

$$\bar r \in \bigcap_k V^k\,, \quad \bar d \in \widehat V^\circ\,, \quad \bar d^\top \bar r > 1\,; \tag{5.37}$$

but σ_{G^k} represents V^k, so (5.37) gives

$$\sigma_{G^k}(\bar r) \leqslant 1 < \bar d^\top \bar r\,, \quad \text{hence } \bar d \notin G^k\,.$$

Then $\bar d \in V^\bullet \cap N^k$ for all k (large enough), i.e. $\bar d \in [d_V, t_V d_V]$. Introduce $d_k \in G^k$ from (5.34):

$$\|d_k - \bar d\| \leqslant \frac{1}{k} \quad \text{and} \quad d_k^\top \bar r \leqslant \sigma_{G^k}(\bar r) \leqslant 1\,.$$

Passing to the limit, $\bar d^\top \bar r \leqslant 1$; a contradiction to (5.37). Therefore $\bar r \in V$. ■

Now we assume the existence of an S-free set W containing V; it satisfies in particular

$$W^\bullet \subset W^\circ \subset V^\circ = [0,1]V^\bullet\,. \tag{5.38}$$

If $W^\bullet \subset V^\bullet$, this W is of no use to disprove maximality of V (Proposition 5.4.8). We are therefore in the situation

$$W^\bullet \not\subset V^\bullet\,, \quad \text{which implies from (5.38):} \quad 0 \notin V^\bullet\,. \tag{5.39}$$

Thus, $W^\bullet$ contains some points out of $V^\bullet$. The key argument for our analysis is that one of these points lies on an extreme ray of V°_∞ – which will be the d_V of Lemma 5.5.3, crucial to construct the unbounded sequence $\{r^k\}$ of Figure 5.8.

Lemma 5.5.4 (Constructing an appropriate extreme ray)**.** *Let $W \supset V$ satisfy* (5.39)*. There is an extreme ray $\mathbb{R}_+ d_V$ of V°_∞ such that the set N^k defined by* (5.32) *satisfies $W^\circ \cap N^k = \emptyset$ for k large enough.*

Proof. From (5.39), we are in the framework of Corollary 5.3.10; Figure 5.10 is helpful to follow the proof. If $\widehat W^\circ \subset V^\bullet$ then $W^\bullet = \overline{\text{conv}}\left(\widehat W^\circ\right) \subset V^\bullet$, contradiction. So there is $e \in \widehat W^\circ$ (hence $\sigma_W(e) = 1$) which does not lie in $V^\bullet$; because $V \subset W$, i.e. $\sigma_V \leqslant \sigma_W$, this e satisfies $\sigma_V(e) < 1$ (otherwise $\sigma_V(e) = 1$, hence $e \in \widehat V^\circ \subset V^\bullet$).

Then construct $d_e := \frac{1}{\sigma_V(e)} e \in \widehat V^\circ$ (remember (5.21): $\sigma_V(e) > 0$). For every $e' \in [0, e]$, the segment $[e', d_e]$ contains e. Being a convex set, $V^\bullet$ cannot contain such an e' (otherwise it would

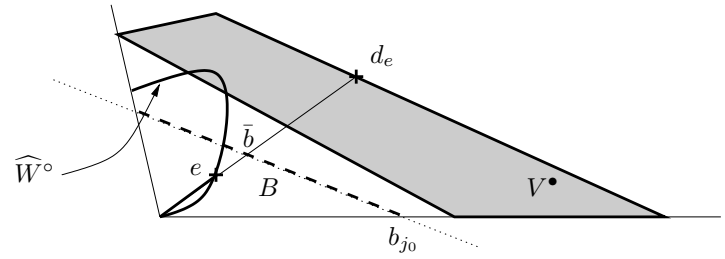


Figure 5.10: The extreme ray $\mathbb{R}_+ b_{j_0}$ contains some point in $V^\bullet \backslash W^\bullet$

contain e as well). As a result, the compact convex sets $V^\bullet$ and $[0, e]$ can be separated: there is $\ell \in \mathbb{R}^q$ (appropriately scaled) such that

$$\max\{0, e^\top \ell\} < 1 < \min_{d \in V^\bullet} d^\top \ell \,. \tag{5.40}$$

Observe that

$$1 > e^\top \ell = \sigma_V(e) d_e^\top \ell > 0 \,. \tag{5.41}$$

Now introduce the closed convex set

$$B := \{b \in V_\infty^\circ \,:\, b^\top \ell = 1\} \,.$$

Clearly, $\mathbb{R}_+ B \subset V_\infty^\circ$. Conversely, apply (5.28): every nonzero $d \in V_\infty^\circ$ can be scaled to some $td \in V^\bullet$. By (5.40), $td^\top \ell > 1$, then d can be scaled again to $td/(td^\top \ell)$, which lies in B. We have shown

$$\mathbb{R}_+ B = V_\infty^\circ \,. \tag{5.42}$$

By (5.28), every $b \in B$ can be obtained by scaling some $d \in \widehat{V}^\circ$: $b = td$; and $t = \frac{1}{d^\top \ell} \in]0, 1[$ by (5.40). This means that

$$B \subset]0, 1[\, \widehat{V}^\circ \subset V^\circ \,; \tag{5.43}$$

B is therefore bounded (and closed because V_∞° is closed), hence compact.

Using (5.41), scale e to $\bar{b} := \frac{1}{e^\top \ell} e \in B$ and express $\bar{b} = \sum_j \alpha_j b_j$ as a convex combination of extreme points b_j of B (Minkowski's Theorem). Then

$$\sigma_W(\bar{b}) = \frac{1}{e^\top \ell}\, \sigma_W(e) = \frac{1}{e^\top \ell} > 1 \,.$$

By convexity of σ_W, there is some j_0 such that $\sigma_W(b_{j_0}) > 1$ (we may have $\sigma_W(b_{j_0}) = +\infty$). Altogether, we have exhibited

$$b_{j_0} \text{ extreme in } B \text{ and satisfying } 1 < \sigma_W(b_{j_0}) \,.$$

Extremality of b_{j_0} in B implies extremality of the ray $\mathbb{R}_+ b_{j_0}$ in $\mathbb{R}_+ B$, i.e. in V_∞° because of (5.42). The intersection of W° with this extreme ray is some $[0, d_W]$ (d_W may be 0) which, by definition of a polar, does not contain b_{j_0}. Since $b_{j_0}^\top \ell = 1$ (because $b_{j_0} \in B$), $d^\top \ell < 1$ for all $d \in [0, d_W]$. Then, (5.40) shows that $[0, d_W]$ and $[d_V, t_V d_V]$ are separated.

As a result, the two compact sets W° and $[d_V, t_V d_V]$ are disjoint. If there were $d^k \in W^\circ \cap N^k$ for all k, then the bounded sequence $\{d^k\}$ would have some cluster point d^*; but W° is closed: d^* would lie in $W^\circ \cap [d_V, t_V d_V]$, contradiction. ■

The set B constructed in the above proof is a so-called basis of the pointed cone V_∞°. The case $\sigma_W(b_{j_0}) = +\infty$, $d_W = 0$ corresponds to a W as in Figure 5.5; it occurs in Figure 5.10. This latter picture is still helpful to follow the next proof. Recall that L is the lineality space of V.

Proposition 5.5.5. *Assume $0 \in \overline{S} = \overline{\mathrm{conv}}\,(S)$. If a minimal* CGF *$\rho$ represents the S-free set $V = V(\rho)$ which is not maximal, then V^k exists as described by Lemma 5.5.3. There is $r^k \in V^k \cap S$, decomposed as $r^k = \ell^k + u^k$ with $\ell^k \in L$ and $u^k \in L^\perp$, such that*

$$\text{for some } K \subset \mathbb{N}\,, \quad \lim_{k\in K} \|r^k\| = +\infty \ \text{ and } \ \lim_{k\in K} \|u^k\| = +\infty\,.$$

Proof. If all of the S-free sets W containing V satisfy $W^\bullet \subset V^\bullet$, then V is maximal (Proposition 5.4.8). Thus, there is an S-free set $W \supset V$ satisfying (5.39) and we can construct d_V as in Lemma 5.5.4.

If $\mathbb{R}_+ d_V = V_\infty^\circ$, then $\widehat{V}^\circ = V^\bullet = \{d_V\}$ and $V^\circ = [0, d_V]$ (Proposition 5.3.7): the S-free set V, represented by σ_{V°, is the half-space $\{r : d_V^\top r \leqslant 1\}$, which separates 0 from $\overline{S}$; this is ruled out by assumption.

Otherwise, $\mathbb{R}_+ d_V \subsetneq V_\infty^\circ$: we can apply Lemma 5.5.3 and construct the sequence of neighborhoods V^k. By minimality of μ_V, V^k cannot be S-free (Lemma 5.5.3(i) and Theorem 5.4.10): there exists r^k lying

– in $\operatorname{int} V^k$, hence from (5.12)

$$1 > \sigma_{G^k}(r^k)\,, \tag{5.44}$$

– and in S, hence $r^k \notin \operatorname{int} W$: $\sigma_{W^\bullet}(r^k) \geqslant 1$; since $W^\bullet$ is compact,

$$\exists e_k \in W^\bullet \text{ such that } e_k^\top r^k \geqslant 1\,. \tag{5.45}$$

Now we claim that there is $\delta > 0$ such that

$$t_k e_k \in V^\bullet \cap N^k\,, \quad \text{for some } t_k \geqslant 1+\delta \text{ and all } k \text{ large enough}\,. \tag{5.46}$$

Using (5.28), scale e_k (nonzero from its definition) to $t_k e_k \in V^\bullet$; and note from (5.38) that $t_k \geqslant 1$. Then (5.45) implies that $t_k e_k \notin G^k$: otherwise

$$1 \leqslant e_k^\top r^k \leqslant t_k e_k^\top r^k \leqslant \sigma_{G^k}(r^k)$$

by definition of a support function; this contradicts (5.44). It follows that $t_k e_k \in V^\bullet \cap N^k$, which is far from $W^\bullet$ (Lemma 5.5.4); (5.46) is proved.

Now we can conclude. First, let $\bar{d} \in [d_V, t_V d_V]$ be a cluster point of the bounded sequence $\{t_k e_k\}$. Next, use (5.46), (5.45), (5.44) to write for all $d \in G^k$

$$1+\delta \leqslant t_k \leqslant t_k e_k^\top r^k = (t_k e_k - d)^\top r^k + d^\top r^k < (t_k e_k - d)^\top r^k + 1\,.$$

This holds in particular for $d = d_k$ stated in (5.34):

$$\delta < (t_k e_k - d_k)^\top r^k\,. \tag{5.47}$$

Then we obtain with the Cauchy-Schwarz inequality

$$\delta < \|t_k e_k - \bar{d} + \bar{d} - d_k\|\,\|r^k\| \leqslant \Big(\|t_k e_k - \bar{d}\| + \frac{1}{k}\Big)\|r^k\|\,.$$

Furthermore, decompose $r^k = \ell^k + u^k$ in (5.47) and observe that both $e_k^\top \ell^k$ and $d_k^\top \ell^k$ are 0 ($\ell^k \in L$ while e^k and d^k lie in $V_\infty^\circ \subset L^\perp$). So (5.47) gives also

$$\delta < (t_k e_k - d_k)^\top u^k \leqslant \Big(\|t_k e_k - \bar{d}\| + \frac{1}{k}\Big)\,\|u^k\|\,.$$

Both statements are proved since there is $K \subset \mathbb{N}$ such that $\lim_{k\in K} \|t_k e_k - \bar{d}\| = 0$. ■

As suggested in the beginning of this section, proving Theorem 5.5.1 is now easy. An S-free set represented by a minimal CGF will be automatically maximal under any assumption contradicting the existence of our unbounded sequences.

Proof. of Theorem 5.5.1 Construct the sequences $\{r^k\}$ and $\{u^k\}$ of Proposition 5.5.5.

Case (i): Extract a cluster point $\hat{r}$ of the normalized subsequence $\{r^k\}_{k\in K}$: for some $K' \subset K$,

$$\lim_{k\in K'} \frac{r^k}{\|r^k\|} = \hat{r}\,.$$

Then take an arbitrary $M > 0$. We know that $M/\|r^k\| \leqslant 1$ if k is large enough in K' so, because both 0 and r^k lie in $V^k \cap \overline{S}$,

$$\frac{M}{\|r^k\|} r^k \in V^k \cap \overline{S}\,, \quad \text{for large enough } k \in K'\,.$$

By closedness, this implies $M\hat{r} \in \overline{S}$, hence $\hat{r} \in \overline{S}_\infty$ because M is arbitrary. The same argument using Lemma 5.5.3(ii) gives $\hat{r} \in V_\infty$.

Let us sum up. If V is not maximal, then $V_\infty \cap \overline{S}_\infty$ contains a vector $\hat{r}$ of norm 1; this contradicts (i).

Case (ii): Write $u^k = r^k - \ell^k \in V^k - L = V^k + L \subset V^k + V_\infty \subset V^k$. Then proceed as in Case (i): extract a cluster point $\hat{u}$ of $\left\{\frac{u^k}{\|u^k\|}\right\}_K$ and argue that $\frac{M}{\|u^k\|}u^k \in V^k \cap L^\perp$ to exhibit

$$\hat{u} \in V_\infty \cap L^\perp \quad \text{and} \quad \|\hat{u}\| = 1\,. \tag{5.48}$$

Besides, u^k is the projection onto $L^\perp$ (a linear operator) of $r^k \in S \subset U + \overline{S}_\infty$; hence

$$u^k \in \mathrm{Proj}_{L^\perp} U + \mathrm{Proj}_{L^\perp} \overline{S}_\infty\,.$$

By $(ii)_1$, $\mathrm{Proj}_{L^\perp} U$ is a bounded set, so our cluster direction $\hat{u}$ lies in $\mathrm{Proj}_{L^\perp} \overline{S}_\infty$:

$$\hat{u} = \hat{s} - \hat{\ell}\,, \quad \text{for some } \hat{s} \in \overline{S}_\infty \text{ and } \hat{\ell} \in L\,.$$

Use (5.48):

$$\overline{S}_\infty \ni \hat{s} = \hat{u} + \hat{\ell} \in V_\infty + L = V_\infty\,;$$

then use $(ii)_2$:

$$\hat{s} \in V_\infty \cap \overline{S}_\infty = L \cap \overline{S}_\infty\,.$$

As a result, $\hat{u} = \hat{s} - \hat{\ell}$ lies in L; use (5.48) again: $\hat{u} \in L \cap L^\perp$ cannot have norm 1.

Thus, in this case also, V has to be maximal. ∎

Let us insist once more: the core of our proof is Proposition 5.5.5. Then (i) and (ii) appear as *ad hoc* assumptions to contradict the existence of the stated unbounded sequences; other similar assumptions might be designed.

5.6 Conclusion and perspectives

In this chapter, we have laid down some basic theory toward studying the cutting paradigm for sets of the form (5.1). We have introduced for this the concept of cut-generating functions, which allowed us to put in perspective an abundant literature devoted to S-free sets. We have revealed the discrepancy between minimality and maximal S-freeness; and we have recovered existing theorems [17, 18, 38, 68, 106], dealing with mere minimality, exhibiting the intrinsic arguments allowing their proofs. Our theory necessitated a generalization of the polarity correspondence to certain unbounded sets; we have conducted it via a systematic exploitation of the correspondence between sublinear functions and closed convex sets.

This theoretical work opens a crucial question: do CGF's do generate all possible cuts ? (i.e., is (5.8) able to produce all possible c's satisfying (5.2)). This turns out to be a tough nut to crack, we conclude with some considerations for future research concerning it. The following counter-example shows that the answer is no in general.

Example 5.6.1 (CGF's need not generate all cuts)**.** In $\mathbb{R}^2$, take $S = (0,1) \cup \big\{(\mathbb{Z}, -1)\big\}$. The left part of Figure 5.11, drawn in the S-space, clearly shows that, if the unit-vector $(1,0)$ lies in the recession cone of an S-free set V, then it lies on the boundary of this cone.

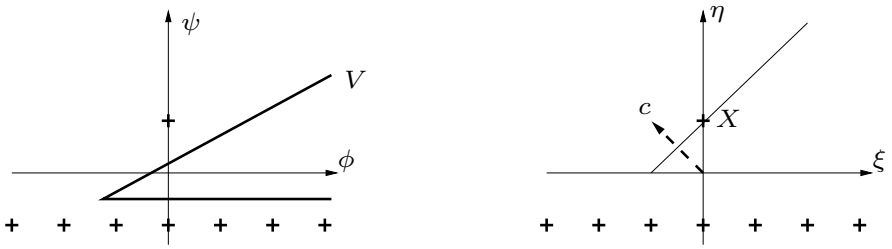


Figure 5.11: Not all cuts are obtained from a CGF

Now take the identity matrix for R: in the $x = (\xi, \eta)$-space, X reduces to the singleton $(0,1)$ in $\mathbb{R}^2$ (right part of Figure 5.11). It can be separated from the origin by the cut $\eta \geqslant \xi + 1$, obtained with $c = (-1,1)^\top$. Knowing that the first column of R is $r_1 = (1,0)^\top$, a CGF ρ producing this c must therefore have $\rho(r_1) = -1$. In view of Lemma 5.3.2, $(1,0)$ lies in the interior of V_∞; but we have seen that no V can satisfy this. □

Negative c_j's are therefore troublesome, a general sufficiency theorem is out of reach. To eliminate $c_j < 0$, we can restrict the class of instances:

Proposition 5.6.2. *If the recession cone of $\overline{\text{conv}}\,(X)$ is the whole of $\mathbb{R}^n_+$, then every cut c lies in $\mathbb{R}^n_+$.*

Proof. Each basis vector e_j of $\mathbb{R}^n$ lies in $\big[\overline{\text{conv}}\,(X)\big]_\infty$: picking some $x \in X$,

$$c^\top(x + te_j) = c^\top x + tc_j \geqslant 1 \quad \text{for all } t \geqslant 0\,;$$

let $t \to +\infty$ to see that $c_j \geqslant 0$. ■

This result might suggest that the trouble in Example 5.6.1 is due to the difference between the recession cones of $\overline{\text{conv}}\,(X)$ and of the ground set $\mathbb{R}^n_+$ in (5.1a). However, the assumption introduced in Proposition 5.6.2 does not suffice, as even $c_j = 0$ brings trouble. In fact, make a "more nonlinear"

variant of Example 5.6.1: instead of the horizontal line $\psi = -1$, take for S the curve $\psi = -1/|\phi|$ ($\phi \neq 0$). This leaves $X = \{(0,1)\}$ unchanged; $c = (0,1)^\top$ is a cut and a CGF ρ generating it has $\rho(r_1) = 0$; this ρ represents a set $V(\rho)$ which has $(\mathbb{R}_+, 0)$ in its recession cone. Being a neighborhood of the origin, $V(\rho)$ contains $A := (0, -\varepsilon)$ for small enough $\varepsilon > 0$; also, $B := (r, 0) \in V(\rho)_\infty \subset V(\rho)$ for all $r > 0$ (see Figure 5.12); by convexity, the whole segment $[A, B]$ lies in $V(\rho)$, which therefore cannot be S-free.

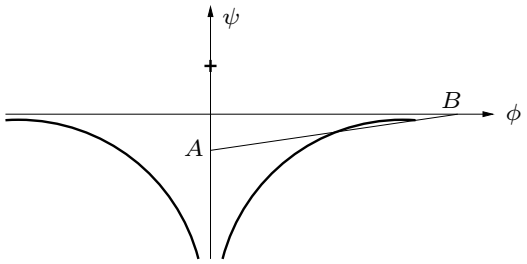


Figure 5.12: Trouble appears when V_∞ is an asymptote of S

In these two examples, the conical hull of the r_j's does not cover the whole of S. In fact, S contains points that can be reached by no $x \in \mathbb{R}^n_+$; these points have nothing to do with the problem, so forcing V not to contain them is unduly demanding. Then one may ask:

whether CGF's are able to describe all possible cuts,
for all possible instances such that $S \subset \operatorname{cone}(r_1, \ldots, r_n)$?

Here we prove this result in the reasonably simple case, limiting ourselves to the use of a "comfortable" assumption; this result motivated the generalization obtained recently in [53].

Theorem 5.6.3. *Let an instance of* (5.1) *be as described by Proposition* 5.6.2 *and assume*

$$\operatorname{cone}(r_1, \ldots, r_n) := \Big\{ \sum_{j=1}^{n} \lambda_j r_j \ : \ \lambda_j \geqslant 0,\ j = 1, \ldots, n \Big\} = \mathbb{R}^q .$$

Then every cut can be obtained from a CGF.

Proof. Let $c \in \mathbb{R}^n_+$ and set

$$J_+ := \big\{ j \in \{1, \ldots, n\} \ : \ c_j > 0 \big\} , \quad J_0 := \big\{ j \in \{1, \ldots, n\} \ : \ c_j = 0 \big\} .$$

Then introduce in $\mathbb{R}^q$ the vectors

$$r'_j := \frac{r_j}{c_j} , \quad \text{for } j \in J_+$$

and the polyhedron

$$V := G + K , \quad \text{with } \begin{cases} G := \operatorname{conv}\{r'_j \ : \ j \in J_+\} , \\ K := \operatorname{cone}\{r_j \ : \ j \in J_0\} . \end{cases}$$

Claim 1: V is a neighborhood of the origin. In fact, our assumption means that $\mathbb{R}^q = \operatorname{cone}(G) + K$: every $\bar{d} \in \mathbb{R}^n$ has the form

$$\bar{d} = \bar{t}\bar{g} + \bar{k} , \quad \text{with} \quad \bar{t} \geqslant 0 ,\ \bar{g} \in G ,\ \bar{k} \in K .$$

Then compute $\sigma_V(\bar{d})$ for nonzero $\bar{d}$.

– Case 1: $\bar{t} = 0$. Fixing $g \in G$ so that $g + t\bar{k} \in V$ for all $t \geqslant 0$, we have

$$\sigma_V(\bar{d}) = \sigma_V(\bar{k}) \geqslant \bar{k}^\top(g + t\bar{k}) = \bar{k}^\top g + t\|\bar{k}\|^2\,, \quad \text{for all } t > 0\,;$$

let $t \to +\infty$ to see that $\sigma_V(\bar{d}) = +\infty$.
– Case 2: $\bar{t} > 0$. Scale $\bar{d}$ to $\bar{t}^{-1}\bar{d} \in G + K = V$ to obtain $\sigma_V(\bar{d}) \geqslant \bar{t}^{-1}\|\bar{d}\|^2 > 0$.
Altogether, we have proved that $\sigma_V(\bar{d}) > 0$ for all $\bar{d} \neq 0$, i.e. $0 \in \mathrm{int}\,(V)$.
Claim 2: V is S-free. Take $\bar{r} \in \mathrm{int}\,(V)$. For $\varepsilon > 0$ small enough, $\bar{r} + \varepsilon\bar{r} \in V$:

$$(1+\varepsilon)\bar{r} = \sum_{j\in J_+} \beta_j r'_j + \sum_{j\in J_0} \mu_j r_j\,, \quad \text{with } \beta_j, \mu_j \geqslant 0, \sum_{j\in J_+} \beta_j = 1\,.$$

Divide by $1+\varepsilon$ and set $\alpha_j = \beta_j/(1+\varepsilon)$, $\lambda_j = \mu_j/(1+\varepsilon)$ to get

$$\bar{r} = \sum_{j\in J_+} \alpha_j r'_j + \sum_{j\in J_0} \lambda_j r_j,\,, \quad \text{for } \alpha_j, \lambda_j \geqslant 0\,, \sum_{j=1}^{n} \alpha_j < 1\,.$$

Introduce the vector $\bar{x} \in \mathbb{R}^n$ whose coordinates are

$$\bar{x}_j := \begin{cases} \dfrac{\alpha_j}{c_j} & \text{if } j \in J_+\,, \\ \lambda_j & \text{if } j \in J_0,\,. \end{cases}$$

Observe that $\bar{x} \geqslant 0$ and that

$$R\bar{x} = \sum_{j=1}^{n} \bar{x}_j r_j = \sum_{j\in J_+} \frac{\alpha_j}{c_j} r_j + \sum_{j\in J_0} \lambda_j r_j = \bar{r}\,.$$

If $\bar{r} \in S$ then $x \in X$ by definition (5.1a); but

$$c^\top \bar{x} = \sum_{j\in J_+} c_j \frac{\alpha_j}{c_j} = \sum_{j\in J_+} \alpha_j \leqslant \sum_{j=1}^{n} \alpha_j < 1$$

and x cannot lie in X if c is a cut. We have proved that $\mathrm{int}\,(V) \cap S = \emptyset$, i.e. that V is S-free.
Conclusion: We have proved that the gauge γ_V is a CGF; besides
– for $j \in J_0$, r_j is a direction of recession of V: $\gamma_V(r_j) = 0 = c_j$;
– for $j \in J_+$, the property $r'_j \in V$ gives

$$1 \geqslant \gamma_V(r'_j) = \frac{1}{c_j}\gamma_V(r_j)\,, \quad \text{hence } \gamma_V(r_j) \leqslant c_j\,.$$

In summary, γ_V is a CGF dominating the cut c. ■

The argument of this theorem is quite rudimentary. A few months after [Mal-8], the above comfortable assumption was dropped in [53] which proved the consistency under the assumption that S is included in the conic hull of R. This completely answers to one of the main open question raised here. Several other questions and perpectives are discussed in Chapter 7.

Chapter 6

Variational analysis of alternating projections methods

This chapter corresponds to the article [Mal-20] in collaboration with A. Lewis (Cornell University) et R. Luke (Universitat Gottingen), published in Foundations of Computational Mathematics (which is one of the best journal in applied maths).

6.1 Introduction

An important theme in computational mathematics is the relationship between "conditioning" of a problem instance and speed of convergence of iterative solution algorithms on that instance. A classical example is the method of conjugate gradients for a positive definite system of linear equations: the relative condition number of the associated matrix gives a bound on the linear convergence rate. More generally, Renegar [152, 153] showed that the rate of convergence of interior-point methods for conic convex programming can be bounded in terms of the "distance to ill-posedness" of the program.

In studying the convergence of iterative algorithms for nonconvex minimization problems or nonmonotone variational inequalities, we must content ourselves with a local theory. A suitable analogue of the distance to ill-posedness is then the notion of "metric regularity", fundamental in variational analysis. Loosely speaking, a constraint system, such as a system of inequalities, for example, is metrically regular when, locally, we can bound the distance from a trial solution to an exact solution by a constant multiple of the error in the equation generated by the trial solution. The constant needed is called the "regularity modulus", and its reciprocal has a natural interpretation as a distance to ill-posedness for the equation [72]. While not appropriate as a universal condition on general variational systems [138], metric regularity is often a reasonable assumption for constraint systems.

This philosophy suggests understanding the speed of convergence of algorithms for solving constraint systems in terms of the regularity modulus at a solution. Recent literature focuses in particular on the proximal point algorithm (see for example [79, 105]). A unified approach to the relationship between metric regularity and the linear convergence of a family of conceptual algorithms appears in [114].

We here study a very basic algorithm for a very basic problem. We consider the problem of finding a point in the intersection of several closed sets, using the *method of averaged projections*: at each step, we project the current iterate onto each set, and average the results to obtain the next iterate. Global convergence of this method for convex sets was proved in 1969 in [12]. Here we show, in complete generality, that this method converges locally to a point in the intersection of the sets, at a

linear rate governed by an associated regularity modulus. Our linear convergence proof is elementary: although we use the idea of the normal cone, we apply only the definition, and we discuss metric regularity only to illuminate the rate of convergence.

Finding a point in the intersection of several sets is a problem of fundamental computational significance. In the case of closed halfspaces, for example, the problem is equivalent to linear programming. We mention some nonconvex examples below.

Our approach to the convergence of the method of averaged projections is standard, see e.g. [22]: we identify the method with von Neumann's *alternating* projections algorithm [179] on two closed sets (one of which is a linear subspace) in a suitable product space. A nice development of the classical method of alternating projections in the convex case may be found in [66]. The convergence of the method for two intersecting closed convex sets was proved in [42], and *linear* convergence under a regular intersection assumption was proved in [22], strengthening a classical result of [91]. Our algorithmic contribution is to show that, assuming linear regularity, *local* linear convergence does not depend on convexity of both sets, but rather on a good geometric property (such as convexity, smoothness, or more generally, "amenability " or "prox-regularity") of just *one* of the two.

One consequence of our convergence proof is an algorithmic demonstration for the "exact extremal principle" of [137] (see also [136, Theorem 2.8]). This result, a unifying theme in [136], asserts that if several sets have linearly regular intersection at a point, then that point is not "locally extremal": that is, translating the sets by sufficiently small vectors cannot render the intersection empty locally. To prove this result, we simply apply the method of averaged projections, starting from the point of regular intersection. In a further section, we show that inexact versions of the method of averaged projections, closer to practical implementations, also converge linearly.

The method of averaged projections is a conceptual algorithm that might appear hard to implement on concrete nonconvex problems. However, the projection problem for some nonconvex sets is relatively easy. A good example is the set of matrices of some fixed rank: given a singular value decomposition of a matrix, projecting it onto this set is immediate. Furthermore, nonconvex alternating projection algorithms and analogous heuristics are quite popular in practice, in areas such as inverse eigenvalue problems [48, 49], pole placement [143, 182], information theory [171], low-order control design [89, 90, 144] and image processing [23, 180]. Previous convergence results on nonconvex alternating projection algorithms have been uncommon, and have either focussed on a very special case (see for example [48, Mal-21]), or have been much weaker than for the convex case [52, 171]. For more discussion, see [Mal-21].

Our results primarily concern R-linear convergence: we show that our sequences of iterates converge, with error bounded by a geometric sequence. In a final section, we employ a completely different approach to show that the method of averaged projections, for prox-regular sets with regular intersection, has a Q-linear convergence property: each iteration guarantees a fixed rate of improvement. In a final section, we illustrate these theoretical results with an elementary numerical example coming from signal processing.

Our interest here is not in the development of practical numerical methods. Notwithstanding linear convergence proofs, basic alternating and averaged projection schemes may be slow in practice. Rather we aim to study the interplay between a simple, popular, fundamental algorithm and a variety of central ideas from variational analysis. Whether such an approach can help in the design and analysis of more practical algorithms remains to be seen.

6.2 Notation and definitions

We fix some notation and definitions. Our underlying setting throughout this work is a Euclidean space $\mathbb{E}$ with corresponding closed unit ball B. For any point $x \in \mathbb{E}$ and radius $\rho > 0$, we write $B_\rho(x)$ for the set $x + \rho B$.

Consider first two sets $F, G \subset \mathbb{E}$. A point $\bar{x} \in F \cap G$ is *locally extremal* [136] for this pair of sets if there exists a constant $\rho > 0$ and a sequence of vectors $z_r \to 0$ in $\mathbb{E}$ such that $(F+z_r)\cap G \cap B_\rho(\bar{x}) = \varnothing$ for all $r = 1, 2, \ldots$. In other words, restricting to a neighborhood of $\bar{x}$ and then translating the sets by arbitrarily small distances can render their intersection empty. Clearly $\bar{x}$ is not locally extremal if and only if

$$0 \in \text{int}\Big(((F - \bar{x}) \cap \rho B) - ((G - \bar{x}) \cap \rho B)\Big) \text{ for all } \rho > 0.$$

For recognition purposes, it is easier to study a weaker property than local extremality. We say that two sets $F, G \subset \mathbb{E}$ have *linearly regular intersection* at the point $\bar{x} \in F \cap G$ if there exist constants $\alpha, \delta > 0$ such that for all points $x \in F \cap B_\delta(\bar{x})$ and $z \in G \cap B_\delta(\bar{x})$, and all $\rho \in (0, \delta]$, we have

$$\alpha\rho B \ \subset \ ((F - x) \cap \rho B) - ((G - z) \cap \rho B).$$

(In [115] this property is called "strong regularity".) By considering the case $x = z = \bar{x}$, we see that linear regularity implies that $\bar{x}$ is not locally extremal. This "primal" definition of linear regularity is often not the most convenient way to handle linear regularity, either conceptually or theoretically. By contrast, a "dual" approach, using normal cones, is very helpful.

Given a set $F \subset \mathbb{E}$, we define the *distance function* and (multivalued) *projection* for F by

$$\begin{aligned} d_F(x) &= d(x, F) = \inf\{\|z - x\| : z \in F\} \\ P_F(x) &= \text{argmin}\{\|z - x\| : z \in F\}. \end{aligned}$$

The *normal cone* to a closed set $F \subset \mathbb{E}$ at a point $\bar{x} \in F$ is

$$N_F(\bar{x}) \ = \ \Big\{ \lim_i t_i(x_i - z_i) : t_i \geq 0, \ x_i \to \bar{x}, \ z_i \in P_F(x_i) \Big\}.$$

The centrality of this idea in variational analysis is described at length in [50, 136, 158]). This construction dates back to [137]: see [158, Chapter 6 Commentary] and [136, Chapter 1 Commentary] for a discussion of the equivalence between this definition and that of [158, p. 199]. Notice two properties in particular. First,

$$z \in P_F(x) \ \Rightarrow \ x - z \in N_F(z). \tag{6.1}$$

Secondly, the normal cone is a "closed" multifunction: for any sequence of points $x_r \to \bar{x}$ in F, any limit of a sequence of normals $y_r \in N_F(x_r)$ must lie in $N_F(\bar{x})$. Indeed, the normal cone is the smallest cone satisfying the two properties. Note also $N_F(x) = \{0\} \iff x \in \text{int}\, F$.

Normal cones provide an elegant alternative approach to defining linear regularity. A family of closed sets $F_1, F_2, \ldots F_m \subset \mathbb{E}$ has *linearly regular intersection* at a point $\bar{x} \in \cap_i F_i$, when the only solution to the system

$$\sum_{i=1}^{m} y_i = 0, \ \text{ with } \ y_i \in N_{F_i}(\bar{x}) \ \ (i = 1, 2, \ldots, m)$$

is $y_i = 0$ for $i = 1, 2, \ldots, m$ (cf. the "exact extremal principle" of [136, Theorem 2.8]). In the case $m = 2$, this condition can be written

$$N_{F_1}(\bar{x}) \cap -N_{F_2}(\bar{x}) = \{0\}, \tag{6.2}$$

and it is equivalent to our previous definition (see [115, Corollary 2], for example). We also note that this condition appears throughout variational-analytic theory. For example, it guarantees the crucial inclusion (see [135, Theorem 1] and also [158, Theorem 6.42])

$$N_{F_1 \cap \ldots \cap F_m}(\bar{x}) \;\subset\; N_{F_1}(\bar{x}) + \cdots + N_{F_m}(\bar{x}).$$

For convex F_1 and F_2, condition (6.2) asserts the nonexistence of a separating hyperplane. More generally, linear regularity was introduced in [135] as the "generalized nonseparation property". The notion of a "linear regular" family of convex sets [20] is also related, though the definition we use here is local.

We will find it helpful to quantify the notion of linear regularity (cf. [115]). A straightforward compactness argument shows the following result.

Proposition 6.2.1 (quantifying linear regularity)**.** *A collection of closed sets $F_1, F_2, \ldots, F_m \subset \mathbb{E}$ have linearly regular intersection at a point $\bar{x} \in \cap F_i$ if and only if there exists a constant $k > 0$ such that the following condition holds:*

$$y_i \in N_{F_i}(\bar{x}) \;\; (i = 1, 2, \ldots, m) \;\; \Rightarrow \;\; \sqrt{\sum_i \|y_i\|^2} \leq k \Big\| \sum_i y_i \Big\|. \tag{6.3}$$

We define the *condition modulus* $\mathrm{cond}(F_1, F_2, \ldots, F_m | \bar{x})$ to be the infimum of all constants $k > 0$ such that property (6.3) holds. Since $\|\cdot\|^2$ is convex, we notice that vectors $y_1, y_2, \ldots, y_m \in \mathbb{E}$ always satisfy the inequality

$$\sum_i \|y_i\|^2 \geq \frac{1}{m} \Big\| \sum_i y_i \Big\|^2, \tag{6.4}$$

which yields

$$\mathrm{cond}(F_1, F_2, \ldots, F_m | \bar{x}) \geq \frac{1}{\sqrt{m}}\,, \tag{6.5}$$

except in the special case when $N_{F_i}(\bar{x}) = \{0\}$ (or equivalently $\bar{x} \in \mathrm{int}\, F_i$) for all $i = 1, 2, \ldots, m$; in this case the condition modulus is zero.

One goal of this chapter is to show that, far from being of purely analytic significance, linear regularity has central algorithmic consequences, specifically for the method of averaged projections for finding a point in the intersection $\cap_i F_i$. Given any initial point $x_0 \in \mathbb{E}$, the algorithm proceeds iteratively as follows:

$$\begin{aligned} z_n^i \;\; &\in \;\; P_{F_i}(x_n) \;\; (i = 1, 2, \ldots, m) \\ x_{n+1} \;\; &= \;\; \frac{1}{m}(z_n^1 + z_n^2 + \cdots + z_n^m). \end{aligned}$$

Our main result shows, assuming only linear regularity, that providing the initial point x_0 is sufficiently near $\bar{x}$, any sequence $x_1, x_2, x_3, \ldots$ generated by the method of averaged projections converges linearly to a point in the intersection $\cap_i F_i$, at a rate governed by the condition modulus.

6.3 Linear and metric regularity

The notion of linear regularity is well-known to be closely related to another central idea in variational analysis: "metric regularity". A concise summary of the relationships between a variety of

regular intersection properties and metric regularity appears in [115]. We summarize the relevant ideas here.

Consider a set-valued mapping $\Phi\colon \mathbb{E} \rightrightarrows \mathbf{Y}$, where $\mathbf{Y}$ is a second Euclidean space. The inverse mapping $\Phi^{-1}\colon \mathbf{Y} \rightrightarrows \mathbb{E}$ is defined by $x \in \Phi^{-1}(y) \Leftrightarrow y \in \Phi(x)$, for $x \in \mathbb{E}$ and $y \in \mathbf{Y}$. For vectors $\bar{x} \in \mathbb{E}$ and $\bar{y} \in \Phi(\bar{x})$, we say Φ is *metrically regular* at $\bar{x}$ for $\bar{y}$ if there exists a constant $\kappa > 0$ such that all pairs $(x, y) \in \mathbb{E} \times \mathbf{Y}$ sufficiently near $(\bar{x}, \bar{y})$ satisfy the inequality

$$d(x, \Phi^{-1}(y)) \leq \kappa d(y, \Phi(x)).$$

The infimum of all such constants κ is called the *modulus of metric regularity* of Φ at $\bar{x}$ for $\bar{y}$, denoted $\operatorname{reg} \Phi(\bar{x}|\bar{y})$. See [158, Chapter 9G] for a discussion.

Intuitively, metric regularity gives a local linear bound for the distance to a solution of the constraint system $y \in \Phi(x)$ (where the vector y is given and we seek the unknown vector x), in terms of the the distance from y to the set $\Phi(x)$. The modulus is a measure of the sensitivity or "conditioning" of the constraint system $y \in \Phi(x)$. To take one simple example, if Φ is a single-valued linear map, the modulus of regularity is the reciprocal of its smallest singular value. In general, variational analysis provides a powerful calculus for computing the regularity modulus. In particular, we have the following formula (see [135, Theorem 8] and [158, Theorem 9.43]):

$$\frac{1}{\operatorname{reg} \Phi(\bar{x}|\bar{y})} \;=\; \min\Big\{d(0, D^*\Phi(\bar{x}|\bar{y})(w)) : w \in \mathbf{Y},\ \|w\| = 1\Big\}, \tag{6.6}$$

where D^* denotes the "coderivative".

We now study these ideas for a particular mapping, highlighting the connections between metric and linear regularity. As in the previous section, consider closed sets $F_1, F_2, \ldots, F_m \subset \mathbb{E}$ and a point $\bar{x} \in \cap_i F_i$. We endow the space $\mathbb{E}^m$ with the inner product

$$\Big\langle (x_1, x_2, \ldots, x_m), (y_1, y_2, \ldots, y_m) \Big\rangle \;=\; \sum_i \langle x_i, y_i \rangle,$$

and define set-valued mapping $\Phi\colon \mathbb{E} \rightrightarrows \mathbb{E}^m$ by

$$\Phi(x) \;=\; (F_1 - x) \times (F_2 - x) \times \cdots \times (F_m - x).$$

Then the inverse mapping is given by $\Phi^{-1}(y) = \cap_i (F_i - y_i)$, for $y \in \mathbb{E}^m$, and finding a point in the intersection $\cap_i F_i$ is equivalent to finding a solution of the constraint system $0 \in \Phi(x)$. By definition, the mapping Φ is metrically regular at $\bar{x}$ for 0 if and only if there is a constant $\kappa > 0$ such that the following *strong metric inequality* holds:

$$d\Big(x, \bigcap_i (F_i - z_i)\Big) \;\leq\; \kappa \sqrt{\sum_i d^2(x, F_i - z_i)} \;\text{ for all } (x, z) \text{ near } (\bar{x}, 0). \tag{6.7}$$

Furthermore, the regularity modulus $\operatorname{reg} \Phi(\bar{x}|0)$ is just the infimum of those constants $\kappa > 0$ such that inequality (6.7) holds.

To compute the coderivative $D^*\Phi(\bar{x}|0)$, we decompose the mapping Φ as $\Psi - A$, where, for points $x \in \mathbb{E}$, we define $\Psi(x) = F_1 \times F_2 \times \cdots \times F_m$ and $Ax = (x, x, \ldots, x)$. The calculus rule [158, 10.43] yields $D^*\Phi(\bar{x}|0) = D^*\Psi(\bar{x}|A\bar{x}) - A^*$. Then, by definition,

$$v \in D^*\Psi(\bar{x}|A\bar{x})(w) \;\Leftrightarrow\; (v, -w) \in N_{\operatorname{gph} \Psi}(\bar{x}, A\bar{x}),$$

and since $\operatorname{gph}\Psi = \mathbb{E} \times F_1 \times F_2 \times \cdots \times F_m$, we deduce

$$D^*\Psi(\bar{x}|A\bar{x})(w) = \begin{cases} \{0\} & \text{if } w_i \in -N_{F_i}(\bar{x})\ \forall i \\ \varnothing & \text{otherwise} \end{cases}$$

and hence

$$D^*\Phi(\bar{x}|0)(w) = \begin{cases} -\sum_i w_i & \text{if } w_i \in -N_{F_i}(\bar{x})\ \forall i \\ \varnothing & \text{otherwise.} \end{cases}$$

From the coderivative formula (6.6) we now obtain

$$\frac{1}{\operatorname{reg}\Phi(\bar{x}|0)} \;=\; \min\Big\{\Big\|\sum_i y_i\Big\| : \sum_i \|y_i\|^2 = 1,\ y_i \in N_{F_i}(\bar{x})\Big\}, \tag{6.8}$$

where, as usual, we interpret the right-hand side as $+\infty$ if $N_{F_i}(\bar{x}) = \{0\}$ (or equivalently $\bar{x} \in \operatorname{int} F_i$) for all $i = 1, 2, \ldots, m$. Thus the regularity modulus agrees exactly with the condition modulus that we defined in the previous section: $\operatorname{reg}\Phi(\bar{x}|0) = \operatorname{cond}(F_1, F_2, \ldots, F_m|\bar{x})$. It is well-known [115] that linear regularity is equivalent to the strong metric inequality (6.7).

6.4 Clarke regularity and refinements

"Clarke regularity" is a basic variation-geometric property of sets, shared in particular by closed convex sets and smooth manifolds. We next study a slight refinement, crucial for our development. In the interest of maintaining as elementary approach as possible, we use the following definition of Clarke regularity, easy to interpret geometrically in terms of certain angles.

Definition 6.4.1 (Clarke regularity)**.** A closed set $C \subset \mathbf{R}^n$ is *Clarke regular* at a point $\bar{x} \in C$ if, for all $\delta > 0$, any two points u, z sufficiently near $\bar{x}$ with $z \in C$, and any point $y \in P_C(u)$, satisfy $\langle z - \bar{x}, u - y\rangle \;\le\; \delta\|z - \bar{x}\| \cdot \|u - y\|$.

Remark 6.4.2. This property is equivalent to the standard notion of Clarke regularity [158, Definition 6.4]. To see this, suppose the property in the definition holds. Consider any unit vector $v \in N_C(\bar{x})$, and any unit "tangent direction" w to C at $\bar{x}$. By definition, there exists a sequences $u_r \to \bar{x}$, $y_r \in P_C(u_r)$, and $z_r \to \bar{x}$ with $z_r \in C$, such that

$$v_r \;=\; \frac{u_r - y_r}{\|u_r - y_r\|} \quad\to\quad v$$

$$w_r \;=\; \frac{z_r - \bar{x}}{\|z_r - \bar{x}\|} \quad\to\quad w.$$

By assumption, given any $\delta > 0$, for all sufficiently large r we have $\langle v_r, w_r\rangle \le \delta$, and hence $\langle v, w\rangle \le \delta$. Thus $\langle v, w\rangle \le 0$, so Clarke regularity follows, by [158, Corollary 6.29]. Conversely, if the property described in the definition fails, then for some $\delta > 0$ and some sequences $u_r \to \bar{x}$, $y_r \in P_C(u_r)$, and $z_r \to \bar{x}$ with $z_r \in C$, we have

$$\Big\langle \frac{u_r - y_r}{\|u_r - y_r\|}\ ,\ \frac{z_r - \bar{x}}{\|z_r - \bar{x}\|}\Big\rangle \;\ge\; \delta \quad \text{for all } r.$$

Then any cluster points v and w of the two sequences of unit vectors defining the above inner product are respectively an element of $N_C(\bar{x})$ and a tangent direction to C at $\bar{x}$, and satisfy $\langle v, w\rangle > 0$, contradicting Clarke regularity.

The property we need for our development is an apparently-slight modification of Clarke regularity, again easy to interpret geometrically.

Definition 6.4.3 (super-regularity)**.** A closed set $C \subset \mathbf{R}^n$ is *super-regular* at a point $\bar{x} \in C$ if, for all $\delta > 0$, any two points u, z sufficiently near $\bar{x}$ with $z \in C$, and any point $y \in P_C(u)$, satisfy $\langle z - y, u - y \rangle \leq \delta \|z - y\| \cdot \|u - y\|$.

An equivalent statement involves the normal cone.

Proposition 6.4.4 (super-regularity and normal angles)**.** *A closed set $C \subset \mathbf{R}^n$ is super-regular at a point $\bar{x} \in C$ if and only if, for all $\delta > 0$, the inequality $\langle v, z - y \rangle \leq \delta \|v\| \cdot \|z - y\|$ holds for all points $y, z \in C$ sufficiently near $\bar{x}$ and all vectors $v \in N_C(y)$.*

Proof Super-regularity follows immediately from the normal cone property described in the proposition, by property (6.1). Conversely, suppose the normal cone property fails, so for some $\delta > 0$ and sequences of distinct points $y_r, z_r \in C$ approaching $\bar{x}$ and unit normal vectors $v_r \in N_C(y_r)$, we have, for all $r = 1, 2, \ldots$,

$$\left\langle v_r, \frac{z_r - y_r}{\|z_r - y_r\|} \right\rangle > \delta.$$

Fix an index r. By definition of the normal cone, there exist sequences of distinct points $u_r^j \to y_r$ and $y_r^j \in P_C(u_r^j)$ such that

$$\lim_{j \to \infty} \frac{u_r^j - y_r^j}{\|u_r^j - y_r^j\|} = v_r.$$

Since $\lim_j y_r^j = y_r$, we must have, for all sufficiently large j,

$$\left\langle \frac{u_r^j - y_r^j}{\|u_r^j - y_r^j\|}, \frac{z_r - y_r^j}{\|z_r - y_r^j\|} \right\rangle > \delta.$$

Choose j sufficiently large to ensure both the above inequality and the inequality $\|u_r^j - y_r\| < \frac{1}{r}$, and then define points $u_r' = u_r^j$ and $y_r' = y_r^j$.

We now have sequences of points u_r', z_r approaching $\bar{x}$ with $z_r \in C$, and $y_r' \in P_C(u_r')$, and satisfying

$$\left\langle \frac{u_r' - y_r'}{\|u_r' - y_r'\|}, \frac{z_r - y_r'}{\|z_r - y_r'\|} \right\rangle > \delta.$$

Hence C is not super-regular at $\bar{x}$. □

Super-regularity is a strictly stronger property than Clarke regularity, as the following result and example make clear.

Corollary 6.4.5 (super-regularity implies Clarke regularity)**.** *At any point in a closed set $C \subset \mathbf{R}^n$, super regularity implies Clarke regularity.*

Proof Suppose the point in question is $\bar{x}$. Fix any $\delta > 0$, and set $y = \bar{x}$ in Proposition 6.4.4. Then clearly any unit tangent direction d to C at $\bar{x}$ and any unit normal vector $v \in N_C(\bar{x})$ satisfy $\langle v, d \rangle \leq \delta$. Since δ was arbitrary, in fact $\langle v, d \rangle \leq 0$, so Clarke regularity follows by [158, Cor 6.29]. □

Example 6.4.6. Consider the following function $f\colon \mathbb{R} \to (-\infty, +\infty]$, taken from an example in [164]:

$$f(t) = \begin{cases} 2^r(t - 2^r) & (2^r \leq t < 2^{r+1},\ r \in \mathbf{Z}) \\ 0 & (t = 0) \\ +\infty & (t < 0). \end{cases}$$

This function has Clarke-regular epigraph at $(0,0)$, but an exercise shows it is not super-regular there. Indeed, a minor refinement of this example (smoothing the set slightly close to the nonsmooth points $(2^r, 0)$ and $(2^r, 4^{r-1})$) shows that a set can be everywhere Clarke regular, and yet not super-regular.

Super-regularity is a common property: indeed, it is implied by two well-known properties, that we discuss next. Following [158], we say that a set $C \subset \mathbf{R}^n$ is *amenable* at a point $\bar{x} \in C$ when there exists a neighborhood U of $\bar{x}$, a C^1 mapping $G\colon U \to \mathbb{R}^\ell$, and a closed convex set $D \subset \mathbb{R}^\ell$ containing $G(\bar{x})$, and satisfying the constraint qualification

$$N_D(G(\bar{x})) \cap \ker(\nabla G(\bar{x})^*) = \{0\}, \tag{6.9}$$

such that

$$C \cap U \;=\; \{x \in U : G(x) \in D\}.$$

In particular, if C is defined by C^1 equality and inequality constraints and the Mangasarian-Fromovitz constraint qualification holds, then C is amenable.

Proposition 6.4.7 (amenable implies super-regular)**.** *If a closed set $C \subset \mathbf{R}^n$ is amenable at a point in C, then it is super-regular there.*

Proof Suppose the result fails at some point $\bar{x} \in C$. Assume as in the definition of amenability that, in a neighborhood of $\bar{x}$, the set C is identical with the inverse image $G^{-1}(D)$, where the C^1 map G and the closed convex set D satisfy the condition (6.9). Then by definition, for some $\delta > 0$, there are sequences of points $y_r, z_r \in C$ converging to $\bar{x}$ and unit normal vectors $v_r \in N_C(y_r)$ satisfying $\langle v_r, z_r - y_r\rangle > \delta\|z_r - y_r\|$ for all $r = 1, 2, \ldots$. Since the normal cone mapping N_D is outer semicontinuous relative to D [158, Proposition 6.6], it is easy to check the condition

$$N_D(G(y_r)) \cap \ker(\nabla G(y_r)^*) = \{0\},$$

for all sufficiently large r, since otherwise we contradict assumption (6.9). Consequently, using the standard chain rule [158, Exercise 10.26(d)], we deduce $N_C(y_r) = \nabla G(y_r)^* N_D(G(y_r))$, so there are vectors $u_r \in N_D(G(y_r))$ such that $\nabla G(y_r)^* u_r = v_r$. The sequence (u_r) must be bounded, since otherwise, by taking a subsequence, we could suppose $\|u_r\| \to \infty$ and $\|u_r\|^{-1} u_r$ approaches some unit vector $\hat{u}$, leading to the contradiction

$$\hat{u} \in N_D(G(\bar{x})) \cap \ker(\nabla G(\bar{x})^*) = \{0\}.$$

For all sufficiently large r, we now have $\langle \nabla G(y_r)^* u_r, z_r - y_r\rangle > \delta\|z_r - y_r\|$, and by convexity of D, since $u_r \in N_D(G(y_r))$, we have $\langle u_r, G(z_r) - G(y_r)\rangle \leq 0$. Adding these two inequalities gives

$$\langle u_r, G(z_r) - G(y_r) - \nabla G(y_r)(z_r - y_r)\rangle < -\delta\|z_r - y_r\|.$$

But as $r \to \infty$, the left-hand side is $o(\|z_r - y_r\|)$, since the sequence (u_r) is bounded and G is C^1. This contradiction completes the proof. □

A rather different refinement of Clarke regularity is the notion of "prox-regularity". Following [146, Thm 1.3], we call a set $C \subset \mathbb{E}$ is *prox-regular* at a point $\bar{x} \in C$ if the projection mapping P_C is single-valued around $\bar{x}$. (In this case, clearly C must be locally closed around $\bar{x}$.) For example, if, in the definition of an amenable set that we gave earlier, we strengthen our assumption on the map G to be C^2 rather than just C^1, the resulting set must be prox-regular. Without this strengthening, however, notice the set $\{(s,t) \in \mathbb{R}^2 : t = |s|^{3/2}\}$ is amenable at the point $(0,0)$ (and hence super-regular there), but is not prox-regular there.

Proposition 6.4.8 (prox-regular implies super-regular)**.** *If a closed set $C \subset \mathbf{R}^n$ is prox-regular at a point in C, then it is super-regular there.*

Proof If the results fails at $\bar{x} \in C$, then for some $\delta > 0$, there exist sequences of points $y_r, z_r \in C$ converging to the point $\bar{x}$, and a sequence of normal vectors $v_r \in N_C(y_r)$ satisfying the inequality $\langle v_r, z_r - y_r \rangle > \delta \|v_r\| \cdot \|z_r - y_r\|$. By [146, Proposition 1.2], there exist constants $\epsilon, \rho > 0$ such that

$$\left\langle \frac{\epsilon}{2\|v_r\|} v_r, z_r - y_r \right\rangle \leq \frac{\rho}{2} \|z_r - y_r\|^2$$

for all large r. This gives a contradiction, since $\|z_r - y_r\| \leq \frac{\delta\epsilon}{\rho}$ eventually. □

We digress briefly to discuss relationships between super-regularity and other notions in the literature. First note the following equivalent definition, which is an immediate consequence of Proposition 6.4.4, and which gives an alternate proof of Proposition 6.4.8 via "hypomonotonicity" of the truncated normal cone mapping $x \mapsto N_C(x) \cap B$ for prox-regular sets C [146, Thm 1.3].

Corollary 6.4.9 (approximate monotonicity)**.** *A closed set $C \subset \mathbf{R}^n$ is super-regular at a point $\bar{x} \in C$ if and only if, for all $\delta > 0$, the inequality $\langle v - w, y - z \rangle \geq -\delta \|y - z\|$ holds for all points $y, z \in C$ sufficiently near $\bar{x}$ and all normal vectors $v \in N_C(y) \cap B$ and $w \in N_C(z) \cap B$.*

If we replace the normal cone N_C in the property described in the result above by its convex hull, the "Clarke normal cone", we obtain a stronger property, called "subsmoothness" in [13]. Similar proofs to those above show that, like super-regularity, subsmoothness is a consequence of either amenability or prox-regularity. However, subsmoothness is strictly stronger than super-regularity. To see this, consider the graph of the function $f\colon \mathbb{R} \to \mathbb{R}$ defined by the following properties: $f(0) = 0$, $f(2^r) = 4^r$ for all integers r, f is linear on each interval $[2^r, 2^{r+1}]$, and $f(t) = f(-t)$ for all $t \in \mathbb{R}$. The graph of f is super-regular at $(0, 0)$, but is not subsmooth there.

In a certain sense, however, the distinction between subsmoothness and super-regularity is slight. Suppose the set F is super-regular at every point in $F \cap U$, for some open set $U \subset \mathbf{R}^n$. Since super-regularity implies Clarke regularity, the normal cone and Clarke normal cone coincide throughout $F \cap U$, and hence F is also subsmooth throughout $F \cap U$. In other words, "local" super regularity coincides with "local" subsmoothness, which in turn, by [13, Thm 3.16] coincides with the "first order Shapiro property" [163] (also called "near convexity" in [165]) holding locally.

6.5 Alternating projections with nonconvexity

Having reviewed or developed over the last few sections the key variational-analytic properties that we need, we now turn to projection algorithms. In this section we develop our convergence analysis of the method of alternating projections. The following result is our basic tool, guaranteeing conditions under which the method of alternating projections converges linearly. For flexibility, we state it in a rather technical manner. For clarity, we point out afterward that the two main conditions, (6.11) and (6.12), are guaranteed in applications via assumptions of linear regularity and super-regularity (or in particular, amenability or prox-regularity) respectively.

Given any sets $F, C \subset \mathbb{E}$, an *alternating projection sequence* is any sequence of points $\{x_j\}$ in $\mathbb{E}$ satisfying the condition

$$x_{2n+1} \in P_F(x_{2n}) \text{ and } x_{2n+2} \in P_C(x_{2n+1}) \quad (n = 0, 1, 2, \ldots), \tag{6.10}$$

or the same property with F and C interchanged.

Theorem 6.5.1 (linear convergence of alternating projections)**.** *Consider the closed sets $F, C \subset \mathbb{E}$, and a point $\bar{x} \in F$. Fix any constant $\epsilon > 0$. Suppose for some constant $c' \in (0,1)$, the following condition holds:*

$$\left.\begin{array}{ll} x \in F \cap (\bar{x} + \epsilon B), & u \in -N_F(x) \cap B \\ y \in C \cap (\bar{x} + \epsilon B), & v \in N_C(y) \cap B \end{array}\right\} \;\Rightarrow\; \langle u, v\rangle \leq c'. \tag{6.11}$$

Suppose for some constant $\delta \in [0, \frac{1-c'}{2})$ the following condition holds:

$$\left.\begin{array}{rcl} y, z & \in & C \cap (\bar{x} + \epsilon B) \\ v & \in & N_C(y) \cap B \end{array}\right\} \;\Rightarrow\; \langle v, z - y\rangle \leq \delta \|z - y\|. \tag{6.12}$$

Define a constant $c = c' + 2\delta < 1$. Then for any initial point $x_0 \in C$ satisfying $\|x_0 - \bar{x}\| \leq \frac{1-c}{4}\epsilon$, any alternating projection sequence $\{x_j\}$ for the sets F and C must converge with R-linear rate $\sqrt{c}$ to a point $\hat{x} \in F \cap C$ satisfying the inequality $\|\hat{x} - x_0\| \leq \frac{1+c}{1-c}\|x_0 - \bar{x}\|$.

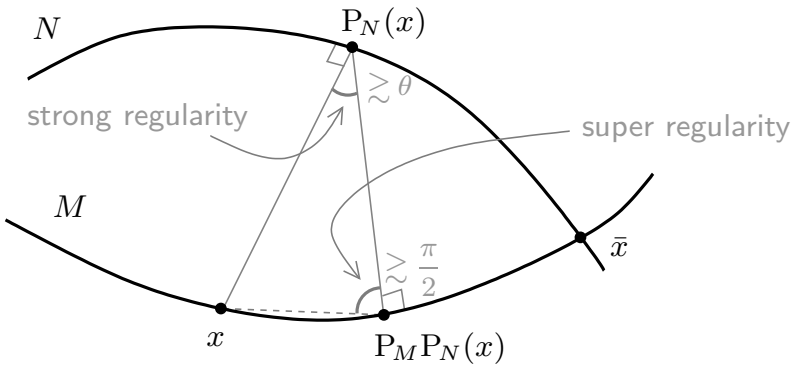


Figure 6.1: Geometry controls convergence: illustration in the case of two manifolds M and N

Proof Assume property (6.10). By the definition of the projections we have

$$\|x_{2n+3} - x_{2n+2}\| \;\leq\; \|x_{2n+2} - x_{2n+1}\| \;\leq\; \|x_{2n+1} - x_{2n}\|. \tag{6.13}$$

Clearly we therefore have

$$\|x_{2n+2} - x_{2n}\| \leq 2\|x_{2n+1} - x_{2n}\|. \tag{6.14}$$

We next claim

$$\left.\begin{array}{l} \|x_{2n+1} - \bar{x}\| \leq \frac{\epsilon}{2} \text{ and} \\ \|x_{2n+1} - x_{2n}\| \leq \frac{\epsilon}{2} \end{array}\right\} \;\Rightarrow\; \|x_{2n+2} - x_{2n+1}\| \leq c\|x_{2n+1} - x_{2n}\|. \tag{6.15}$$

To see this, note that if $x_{2n+2} = x_{2n+1}$, the result is trivial, and if $x_{2n+1} = x_{2n}$ then $x_{2n+2} = x_{2n+1}$ so again the result is trivial. Otherwise, we have

$$\frac{x_{2n} - x_{2n+1}}{\|x_{2n} - x_{2n+1}\|} \in N_F(x_{2n+1}) \cap B$$

while

$$\frac{x_{2n+2} - x_{2n+1}}{\|x_{2n+2} - x_{2n+1}\|} \in -N_C(x_{2n+2}) \cap B.$$

Furthermore, using inequality (6.13), the left-hand side of the implication (6.15) ensures

$$\begin{aligned}\|x_{2n+2} - \bar{x}\| &\leq \|x_{2n+2} - x_{2n+1}\| + \|x_{2n+1} - \bar{x}\| \\ &\leq \|x_{2n+1} - x_{2n}\| + \|x_{2n+1} - \bar{x}\| \leq \epsilon.\end{aligned}$$

Hence, by assumption (6.11) we deduce

$$\left\langle \frac{x_{2n} - x_{2n+1}}{\|x_{2n} - x_{2n+1}\|}, \frac{x_{2n+2} - x_{2n+1}}{\|x_{2n+2} - x_{2n+1}\|} \right\rangle \leq c',$$

so

$$\langle x_{2n} - x_{2n+1}, x_{2n+2} - x_{2n+1} \rangle \leq c' \|x_{2n} - x_{2n+1}\| \cdot \|x_{2n+2} - x_{2n+1}\|.$$

On the other hand, by assumption (6.12) we know

$$\begin{aligned}\langle x_{2n} - x_{2n+2}, x_{2n+1} - x_{2n+2} \rangle &\leq \delta \|x_{2n} - x_{2n+2}\| \cdot \|x_{2n+1} - x_{2n+2}\| \\ &\leq 2\delta \|x_{2n} - x_{2n+1}\| \cdot \|x_{2n+2} - x_{2n+1}\|,\end{aligned}$$

using inequality (6.14). Adding this inequality to the previous inequality then gives the right-hand side of (6.15), as desired.

Now let $\alpha = \|x_0 - \bar{x}\|$. We will show by induction the inequalities

$$\|x_{2n+1} - \bar{x}\| \leq 2\alpha \frac{1 - c^{n+1}}{1 - c} < \frac{\epsilon}{2} \tag{6.16}$$

$$\|x_{2n+1} - x_{2n}\| \leq \alpha c^n < \frac{\epsilon}{2} \tag{6.17}$$

$$\|x_{2n+2} - x_{2n+1}\| \leq \alpha c^{n+1}. \tag{6.18}$$

Consider first the case $n = 0$. Since $x_1 \in P_F(x_0)$ and $\bar{x} \in F$, we deduce $\|x_1 - x_0\| \leq \|\bar{x} - x_0\| = \alpha < \epsilon/2$, which is inequality (6.17). Furthermore,

$$\|x_1 - \bar{x}\| \leq \|x_1 - x_0\| + \|x_0 - \bar{x}\| \leq 2\alpha < \frac{\epsilon}{2},$$

which shows inequality (6.16). Finally, since $\|x_1 - x_0\| < \epsilon/2$ and $\|x_1 - \bar{x}\| < \epsilon/2$, the implication (6.15) shows

$$\|x_2 - x_1\| \leq c\|x_1 - x_0\| \leq c\|\bar{x} - x_0\| = c\alpha,$$

which is inequality (6.18).

For the induction step, suppose inequalities (6.16), (6.17), and (6.18) all hold for some n. Inequalities (6.13) and (6.18) imply

$$\|x_{2n+3} - x_{2n+2}\| \leq \alpha c^{n+1} < \frac{\epsilon}{2}. \tag{6.19}$$

We also have, using inequalities (6.19), (6.18), and (6.16)

$$\begin{aligned}\|x_{2n+3} - \bar{x}\| &\leq \|x_{2n+3} - x_{2n+2}\| + \|x_{2n+2} - x_{2n+1}\| + \|x_{2n+1} - \bar{x}\| \\ &\leq \alpha c^{n+1} + \alpha c^{n+1} + 2\alpha \frac{1 - c^{n+1}}{1 - c},\end{aligned}$$

so

$$\|x_{2n+3} - \bar{x}\| \leq 2\alpha \frac{1 - c^{n+2}}{1 - c} < \frac{\epsilon}{2}. \tag{6.20}$$

Now implication (6.15) with n replaced by $n+1$ implies $\|x_{2n+4} - x_{2n+3}\| \leq c\|x_{2n+3} - x_{2n+2}\|$, and using inequality (6.19) we deduce

$$\|x_{2n+4} - x_{2n+3}\| \leq \alpha c^{n+2}. \tag{6.21}$$

Since inequalities (6.20), (6.19), and (6.21) are exactly inequalities (6.16), (6.17), and (6.18) with n replaced by $n+1$, the induction step is complete and our claim follows.

We can now easily check that the sequence (x_k) is Cauchy and therefore converges. To see this, note for any integer $n = 0, 1, 2, \ldots$ and any integer $k > 2n$, we have

$$\begin{aligned} \|x_k - x_{2n}\| &\leq \sum_{j=2n}^{k-1} \|x_{j+1} - x_j\| \\ &\leq \alpha(c^n + c^{n+1} + c^{n+1} + c^{n+2} + c^{n+2} + \cdots) \end{aligned}$$

so

$$\|x_k - x_{2n}\| \leq \alpha c^n \frac{1+c}{1-c},$$

and a similar argument shows

$$\|x_{k+1} - x_{2n+1}\| \leq \frac{2\alpha c^{n+1}}{1-c}. \tag{6.22}$$

Hence x_k converges to some point $\hat{x} \in \mathbb{E}$, and for all $n = 0, 1, 2, \ldots$ we have

$$\|\hat{x} - x_{2n}\| \leq \alpha c^n \frac{1+c}{1-c} \quad \text{and} \quad \|\hat{x} - x_{2n+1}\| \leq \frac{2\alpha c^{n+1}}{1-c}. \tag{6.23}$$

We deduce that the limit $\hat{x}$ lies in the intersection $F \cap C$ and satisfies the inequality $\|\hat{x} - x_0\| \leq \alpha \frac{1+c}{1-c}$, and furthermore that the inequality

$$\|\hat{x} - x_r\| \leq \alpha(\sqrt{c})^r \frac{1+c}{1-c}$$

holds for all $r = 0, 1, 2, \ldots$, so the convergence is R-linear with rate $\sqrt{c}$. □

We can now prove our key result. To apply Theorem 6.5.1 to alternating projections between a closed and a super-regular set, we make use of the key geometric property of super-regular sets (Proposition 6.4.4).

Theorem 6.5.2 (alternating projections with a super-regular set)**.** *Consider closed sets $F, C \subset \mathbb{E}$ and a point $\bar{x} \in F \cap C$. Suppose C is super-regular at $\bar{x}$ (as holds, for example, if it is amenable or prox-regular there). Suppose furthermore that F and C have linearly regular intersection at $\bar{x}$: that is, $N_F(\bar{x}) \cap -N_C(\bar{x}) = \{0\}$, or equivalently, the constant*

$$\bar{c} \;=\; \max\Big\{\langle u, v\rangle : u \in N_F(\bar{x}) \cap B,\; v \in -N_C(\bar{x}) \cap B\Big\} \tag{6.24}$$

is strictly less than one. Fix any constant $c \in (\bar{c}, 1)$. Then any alternating projection sequence with initial point sufficiently near $\bar{x}$ must converge to a point in $F \cap C$ with R-linear rate $\sqrt{c}$.

Proof Let us show first the equivalence between $\bar{c} < 1$ and linear regularity. The compactness of the intersections between normal cones and the unit ball guarantees the existence of u and v achieving

the maximum in (6.24). Observe then that $\langle u, v\rangle \leq \|u\|\,\|v\| \leq 1$. The cases of equality in the Cauchy-Schwarz inequality permits to write

$$\bar{c} = 1 \iff u \text{ and } v \text{ are colinear} \iff N_F(\bar{x}) \cap -N_C(\bar{x}) \neq \{0\},$$

which corresponds to the desired equivalence.

Denote the alternating sequence $\{x_j\}$. We can suppose $x_0 \in C$. Fix any constant $c' \in (\bar{c}, c)$ and define $\delta = \frac{c-c'}{2}$. To apply Theorem 6.5.1, we just need to check the existence of a constant $\epsilon > 0$ such that conditions (6.11) and (6.12) hold. Condition (6.12) holds for all sufficiently small $\epsilon > 0$, by Proposition 6.4.4. On the other hand, if condition (6.11) fails for all sufficiently small $\epsilon > 0$, then there exist sequences of points $x_r \to \bar{x}$ in the set F and $y_r \to \bar{x}$ in the set C, and sequences of vectors $u_r \in -N_F(x_r) \cap B$ and $v_r \in N_C(y_r) \cap B$, satisfying $\langle u_r, v_r\rangle > c'$. After taking subsequences, we can suppose u_r approaches some vector $u \in -N_F(\bar{x}) \cap B$ and v_r approaches some vector $v \in N_C(\bar{x}) \cap B$, and then $\langle u, v\rangle \geq c' > \bar{c}$, contradicting the definition of the constant $\bar{c}$. □

Corollary 6.5.3 (improved convergence rate)**.** *With the assumptions of Theorem 6.5.2, suppose the set F is also super-regular at $\bar{x}$. Then the alternating projection sequence converges with R-linear rate c.*

Proof Inequality (6.15), and its analog when the roles of F and C are interchanged, together show $\|x_{k+1} - x_k\| \leq c\|x_k - x_{k-1}\|$ for all sufficiently large k, and the result then follows easily, using an argument analogous to that at the end of the proof of Theorem 6.5.1. □

In the light of our discussion in the previous section, the linear regularity assumption of Theorem 6.5.2 is equivalent to the metric regularity at $\bar{x}$ for 0 of the set-valued mapping $\Psi\colon \mathbb{E} \rightrightarrows \mathbb{E}^2$ defined by $\Psi(x) = (F - x) \times (C - x)$, for $x \in \mathbb{E}$. Using equation (6.8), the regularity modulus is determined by

$$\frac{1}{\operatorname{reg} \Psi(\bar{x}|0)} \;=\; \min\Big\{\|u + v\| : u \in N_F(\bar{x}),\; v \in N_C(\bar{x}),\; \|u\|^2 + \|v\|^2 = 1\Big\},$$

and a short calculation then shows

$$\operatorname{reg} \Psi(\bar{x}|0) = \frac{1}{\sqrt{1 - \bar{c}}}. \tag{6.25}$$

The closer the constant $\bar{c}$ is to one, the larger the regularity modulus. We have shown that $\bar{c}$ also controls the speed of linear convergence for the method of alternating projections applied to the sets F and C.

Inevitably, Theorem 6.5.2 concerns *local* convergence: it relies on finding an initial point x_0 sufficiently close to a point of linearly regular intersection. How might we find such a point? One natural context in which to pose this question is that of sensitivity analysis. Suppose we already know a point of linearly regular intersection of two closed sets, but now want to find a point in the intersection of two slight translations of these sets. The following result shows that, starting from the original point of intersection, the method of alternating projections will converge linearly to the new intersection.

Theorem 6.5.4 (perturbed intersection)**.** *Given any closed sets $F, C \subset \mathbb{E}$ and any point $\bar{x} \in F \cap C$, suppose the assumptions of Theorem 6.5.2 hold. Then for any sufficiently small vector $d \in \mathbb{E}$, any alternating projection sequence for the sets $d + F$ and C, with the initial point $\bar{x}$, must converge with R-linear rate $\sqrt{c}$ to a point in the set $(d + F) \cap C \cap B_\rho(\bar{x})$, where $\rho = \frac{1+c}{1-c}\|d\|$.*

Proof As in the proof of Theorem 6.5.2, if we fix any constant $c' \in (\bar{c}, c)$ and define $\delta = \frac{c-c'}{2}$, then there exists a constant $\epsilon > 0$ such that conditions (6.11) and (6.12) hold. Suppose the vector d satisfies

$$\|d\| \leq \frac{(1-c)\epsilon}{8} < \frac{\epsilon}{2}.$$

Since

$$\begin{aligned} y \in (C-d) \cap (\bar{x} + \frac{\epsilon}{2}B) \text{ and } v \in N_{C-d}(y) \\ \Rightarrow \quad y + d \in C \cap (\bar{x} + \epsilon B) \text{ and } v \in N_C(y+d), \end{aligned}$$

we deduce from condition (6.11) the implication

$$\left.\begin{array}{ll} x \in F \cap (\bar{x} + \frac{\epsilon}{2}B), & u \in -N_F(x) \cap B \\ y \in (C-d) \cap (\bar{x} + \frac{\epsilon}{2}B), & v \in N_{C-d}(y) \cap B \end{array}\right\} \quad \Rightarrow \quad \langle u, v \rangle \leq c'.$$

Furthermore, using condition (6.12) we deduce the implication

$$\begin{aligned} y, z \in (C-d) \cap (\bar{x} + \frac{\epsilon}{2}B) \text{ and } v \in N_{C-d}(y) \cap B \\ \Rightarrow \quad y + d, z + d \in C \cap (\bar{x} + \epsilon B) \text{ and } v \in N_C(y+d) \cap B, \\ \Rightarrow \quad \langle v, z - y \rangle \leq \delta \|z - y\|. \end{aligned}$$

We now apply Theorem 6.5.1 with the set C replaced by $C - d$ and ϵ replaced by $\frac{\epsilon}{2}$. We deduce that any alternating projection sequence for the sets F and $C-d$, starting at the point $x_0 = \bar{x} - d \in C - d$, converges with R-linear rate $\sqrt{c}$ to a point $\hat{x} \in F \cap (C-d)$ satisfying the inequality $\|\hat{x} - x_0\| \leq \frac{1+c}{1-c}\|x_0 - \bar{x}\|$. The theorem statement then follows by translation. □

Lack of convexity notwithstanding, more structure sometimes implies that the method of alternating projections converges Q-linearly, rather than just R-linearly, on a neighborhood of point of linearly regular intersection of two closed sets. One example is the case of two manifolds [Mal-21].

6.6 Inexact alternating projections

Our basic tool, the method of alternating projections for a super-regular set C and an arbitrary closed set F, is a conceptual algorithm that may be challenging to realize in practice. We might reasonably consider the case of exact projection on the super-regular set C: for example, in the next section, for the method of averaged projections, C is a subspace and computing projections is trivial. However, projecting onto the set F may be much harder, so a more realistic analysis allows relaxed projections.

We sketch one approach. Given two iterates $x_{2n-1} \in F$ and $x_{2n} \in C$, a necessary condition for the new iterate x_{2n+1} to be an exact projection on F, that is $x_{2n+1} \in P_F(x_{2n})$, is

$$\|x_{2n+1} - x_{2n}\| \leq \|x_{2n} - x_{2n-1}\| \text{ and } x_{2n} - x_{2n+1} \in N_F(x_{2n+1}).$$

In the following result we assume only that we choose the iterate x_{2n+1} to satisfy a relaxed version of this condition, where we replace the second part by the assumption that the distance

$$d_{N_F(x_{2n+1})}\Big(\frac{x_{2n} - x_{2n+1}}{\|x_{2n} - x_{2n+1}\|}\Big)$$

from the normal cone at the iterate to the normalized direction of the last step is *sufficiently small*.

Theorem 6.6.1 (inexact alternating projections)**.** *With the assumptions of Theorem 6.5.2, fix any constant $\gamma < \sqrt{1-c^2}$, and consider the following inexact alternating projection iteration. Given any initial points $x_0 \in C$ and $x_1 \in F$, for $n = 1, 2, 3, \ldots$ suppose $x_{2n} \in P_C(x_{2n-1})$ and $x_{2n+1} \in F$ satisfies*

$$\|x_{2n+1} - x_{2n}\| \leq \|x_{2n} - x_{2n-1}\| \text{ and } d_{N_F(x_{2n+1})}\Big(\frac{x_{2n} - x_{2n+1}}{\|x_{2n} - x_{2n+1}\|}\Big) \leq \gamma.$$

Then, providing x_0 and x_1 are sufficiently close to $\bar{x}$, the iterates converge to a point in $F \cap C$ with R-linear rate

$$\sqrt{c\sqrt{1-\gamma^2} + \gamma\sqrt{1-c^2}} \ < \ 1.$$

Sketch proof. Once again as in the proof of Theorem 6.5.2, we fix any constant $c' \in (\bar{c}, c)$ and define $\delta = \frac{c-c'}{2}$, so there exists a constant $\epsilon > 0$ such that conditions (6.11) and (6.12) hold. Define a vector

$$z = \frac{x_{2n} - x_{2n+1}}{\|x_{2n} - x_{2n+1}\|}.$$

By assumption, there exists a vector $w \in N_F(x_{2n+1})$ satisfying $\|w - z\| \leq \gamma$. Easy manipulation then shows that the unit vector $\hat{w} = \|w\|^{-1}w$ satisfies $\langle \hat{w}, z \rangle \geq \sqrt{1-\gamma^2}$. As in the proof of Theorem 6.5.1, assuming inductively that x_{2n+1} is sufficiently close to both $\bar{x}$ and x_{2n}, since $\hat{w} \in N_F(x_{2n+1})$, and

$$u = \frac{x_{2n+2} - x_{2n+1}}{\|x_{2n+2} - x_{2n+1}\|} \in -N_C(x_{2n+2}) \cap B,$$

we deduce $\langle \hat{w}, u \rangle \leq c'$.

We now see that, on the unit sphere, the arc distance between the unit vectors $\hat{w}$ and z is no more than $\arccos(\sqrt{1-\gamma^2})$, whereas the arc distance between $\hat{w}$ and the unit vector u is at least $\arccos c'$. Hence by the triangle inequality, the arc distance between z and u is at least

$$\arccos c' - \arccos(\sqrt{1-\gamma^2}),$$

so

$$\langle z, u \rangle \ \leq \ \cos\Big(\arccos c' - \arccos(\sqrt{1-\gamma^2})\Big) \ = \ c'\sqrt{1-\gamma^2} + \gamma\sqrt{1-c'^2}.$$

Some elementary calculus shows that the quantity on the right-hand side is strictly less than one. Again as in the proof of Theorem 6.5.1, this inequality shows, providing x_0 is sufficiently close to $\bar{x}$, the inequality

$$\|x_{2n+2} - x_{2n+1}\| \ \leq \ \Big(c\sqrt{1-\gamma^2} + \gamma\sqrt{1-c^2}\Big)\|x_{2n+1} - x_{2n}\|,$$

and in conjunction with the inequality $\|x_{2n+1} - x_{2n}\| \leq \|x_{2n} - x_{2n-1}\|$, this suffices to complete the proof by induction. □

6.7 Local convergence for averaged projections

We return to the problem of finding a point in the intersection of several closed sets via averaged projections. Given sets $F_1, F_2, \ldots, F_m \subset \mathbb{E}$, an *averaged projection sequence* is any sequence of

points $\{x_j\}$ in $\mathbb{E}$ satisfying

$$x_{j+1} \in \frac{1}{m}\sum_{i=1}^{m} P_{F_i}(x_j) \quad (j = 0, 1, 2, \ldots).$$

We apply our previous results to the method of averaged projections via the well-known reformulation of the algorithm as alternating projections on a product space. This leads to the main result of this section, Theorem 6.7.2, which shows linear convergence in a neighborhood of any point of linearly regular intersection, at a rate governed by the associated regularity modulus.

We begin with a characterization of linearly regular intersection, relating the condition modulus with a generalized notion of angle for several sets. Such notions, for collections of convex sets, have also been studied recently in the context of projection algorithms in [67].

Proposition 6.7.1 (characterization of linear regularity)**.** *Closed sets $F_1, F_2, \ldots, F_m \subset \mathbb{E}$ have linearly regular intersection at a point $\bar{x} \in \cap_i F_i$ if and only if the optimal value $\bar{c}$ of the optimization problem*

$$\begin{array}{ll} \textit{maximize} & \sum_i \langle u_i, v_i \rangle \\ \textit{subject to} & \sum_i \|u_i\|^2 \leq 1, \quad \sum_i \|v_i\|^2 \leq 1 \\ & \sum_i u_i = 0 \\ & u_i \in \mathbb{E},\ v_i \in N_{F_i}(\bar{x}) \quad (i = 1, 2, \ldots, m) \end{array}$$

is strictly less than one. Indeed, we have

$$\bar{c}^2 \;=\; \begin{cases} 0 & (\bar{x} \in \cap_i \text{int}\, F_i) \\ 1 - \dfrac{1}{m \cdot \text{cond}^2(F_1, F_2, \ldots, F_m | \bar{x})} & (\textit{otherwise}). \end{cases} \tag{6.26}$$

Proof When $\bar{x} \in \cap_i \text{int}\, F_i$, the result follows by definition. Henceforth, we therefore rule out that case.

For any vectors $u_i, v_i \in \mathbb{E}$ $(i = 1, 2, \ldots, m)$, by Lagrangian duality and differ entiation we obtain

$$\begin{aligned} &\max_{u_i} \Big\{ \sum_i \langle u_i, v_i \rangle : \sum_i \|u_i\|^2 \leq 1,\ \sum_i u_i = 0 \Big\} \\ &\quad = \min_{\lambda \in \mathbb{R}_+,\, z \in \mathbb{E}} \max_{u_i} \Big\{ \sum_i \langle u_i, v_i \rangle + \frac{\lambda}{2}\Big(1 - \sum_i \|u_i\|^2\Big) + \langle z, \sum_i u_i \rangle \Big\} \\ &\quad = \min_{\lambda \in \mathbb{R}_+,\, z \in \mathbb{E}} \Big\{ \frac{\lambda}{2} + \sum_i \max_{u_i} \Big\{ \langle u_i, v_i + z \rangle - \frac{\lambda}{2}\|u_i\|^2 \Big\} \Big\} \\ &\quad = \min_{\lambda > 0,\, z \in \mathbb{E}} \Big\{ \frac{\lambda}{2} + \frac{1}{2\lambda} \sum_i \|v_i + z\|^2 \Big\} rockafellar - wets - 1998 \;=\; \min_{z \in \mathbb{E}} \sqrt{\sum_i \|v_i + z\|^2} \\ &\quad = \sqrt{\sum_{i=1}^{m} \Big\| v_i - \frac{1}{m} \sum_j v_j \Big\|^2} \;=\; \sqrt{\sum_i \|v_i\|^2 - \frac{1}{m} \Big\| \sum_i v_i \Big\|^2}. \end{aligned}$$

Consequently, $\bar{c}^2$ is the optimal value of the optimization problem

$$\begin{aligned} \text{maximize} \quad & \sum_i \|v_i\|^2 - \frac{1}{m}\Big\|\sum_i v_i\Big\|^2 \\ \text{subject to} \quad & \sum_i \|v_i\|^2 \le 1 \\ & v_i \in N_{F_i}(\bar{x}) \quad (i = 1, 2, \ldots, m). \end{aligned}$$

By homogeneity, the optimal solution must occur when the inequality constraint is active, so we obtain an equivalent problem by replacing that constraint by the corresponding equation. By equation (6.8) and the definition of the condition modulus, the optimal value of this new problem is

$$1 - \frac{1}{m \cdot \text{cond}^2(F_1, F_2, \ldots, F_m|\bar{x})}$$

as required. □

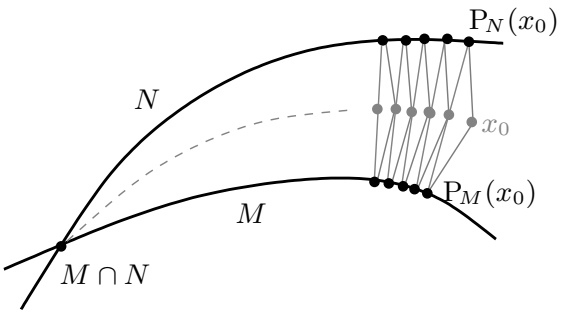


Figure 6.2: Convergence of averaged projections for two sets M and N

Theorem 6.7.2 (linear convergence of averaged projections)**.** *Suppose closed sets $F_1, F_2, \ldots, F_m \subset \mathbb{E}$ have linearly regular intersection at a point $\bar{x} \in \cap_i F_i$. Define a constant $\bar{c} \in [0, 1)$ by equation (6.26), and fix any constant $c \in (\bar{c}, 1)$. Then any averaged projection sequence with initial point sufficiently near $\bar{x}$ converges to a point in the intersection $\cap_i F_i$, with R-linear rate c (and if each set F_i is super-regular at $\bar{x}$, or in particular, prox-regular or amenable there, then the convergence rate is c^2). Furthermore, for any sufficiently small perturbations $d_i \in \mathbb{E}$ for $i = 1, 2, \ldots, m$, any averaged projection sequence for the sets $d_i + F_i$ with the initial point $\bar{x}$ converges linearly to a nearby point in the intersection, with R-linear rate c.*

Proof In the product space $\mathbb{E}^m$ with the inner product

$$\langle (u_1, u_2, \ldots, u_m), (v_1, v_2, \ldots, v_m) \rangle \;=\; \sum_i \langle u_i, v_i \rangle,$$

we consider the closed set $F = \prod_i F_i$ and the subspace $L = \{Ax : x \in \mathbb{E}\}$, where the linear map $A\colon \mathbb{E} \to \mathbb{E}^m$ is defined by $Ax = (x, x, \ldots, x)$. Notice $A\bar{x} \in F \cap L$, and it is easy to check $N_F(A\bar{x}) = \prod_i N_{F_i}(\bar{x})$ and

$$L^\perp \;=\; \Big\{(u_1, u_2, \ldots, u_m) : \sum_i u_i = 0\Big\}.$$

Hence $F_1, F_2, \ldots, F_m$ have linearly regular intersection at $\bar{x}$ if and only if F and L have linearly regular intersection at the point $A\bar{x}$. This latter property is equivalent to the constant $\bar{c}$ in Theorem 6.5.2 (with $C = L$) being strictly less than one. But that constant agrees exactly with that defined by equation (6.26), so we show next that we can apply Theorem 6.5.2 and Theorem 6.5.4.

To see this note that, for any point $x \in \mathbb{E}$, we have the equivalence

$$(z_1, z_2, \ldots, z_m) \in P_F(Ax) \;\; \Leftrightarrow \;\; z_i \in P_{F_i}(x) \;\; (i = 1, 2, \ldots, m).$$

Furthermore a quick calculation shows, for any $z_1, z_2, \ldots, z_m \in \mathbb{E}$,

$$P_L(z_1, z_2, \ldots, z_m) \;=\; \frac{1}{m}(z_1 + z_2 + \cdots + z_m).$$

Hence in fact the method of averaged projections for the sets $F_1, F_2, \ldots, F_m$, starting at an initial point x_0, is essentially identical with the method of alternating projections for the sets F and L, starting at the initial point Ax_0. If $x_0, x_1, x_2, \ldots$ is a possible sequence of iterates for the former method, then a possible sequence of even iterates for the latter method is $Ax_0, Ax_1, Ax_2, \ldots$. For x_0 sufficiently close to $\bar{x}$, this latter sequence must converge to a point $A\hat{x} \in F \cap L$ with R-linear rate c, by Theorem 6.5.2 and its corollary. Thus the sequence $x_0, x_1, x_2, \ldots$ converges to $\hat{x} \in \cap_i F_i$ at the same linear rate. When each of the sets F_i is super-regular at $\bar{x}$, it is easy to check that the Cartesian product F is super-regular at $A\bar{x}$, so the rate is c^2. The last part of the theorem follows from Theorem 6.5.4. $\square$

Applying Theorem 6.6.1 to the product-space formulation of averaged projections shows in a similar fashion that an inexact variant of the method of averaged projections will also converge linearly.

Remark 6.7.3 (linear regularity and local extremality)**.** In the language of [136], that we have proved algorithmically that if closed sets have linearly regular intersection at a point, then that point is not "locally extremal".

Remark 6.7.4 (alternating versus averaged projections)**.** For a feasibility problem for two super-regular sets F_1 and F_2, assume that linear regularity holds at $\bar{x} \in F_1 \cap F_2$ and set $\kappa = \text{cond}(F_1, F_2|\bar{x})$. Theorem 6.7.2 gives a bound on the rate of convergence of the method of averaged projections as

$$r_{\text{av}} \leq 1 - \frac{1}{2\kappa^2} \; .$$

Notice that each iteration involves two projections: one onto each of the sets F_1 and F_2. On the other hand, Corollary 6.5.3 and (6.25) give a bound on the rate of convergence of the method of alternating projections as

$$r_{\text{alt}} \leq 1 - \frac{1}{\kappa^2} \; ,$$

and each iteration involves just one projection. Thus we note that our bound on the rate of alternating projections r_{alt} is always better than the bound on the rate of averaged projections r_{av}. From the perspective of this analysis, averaged projections seems to have no advantage over alternating projections, although our proof of linear convergence for alternating projections needs a super-regularity assumption not necessary in the case of averaged projections.

6.8 Prox-regularity and averaged projections

If we assume that the sets $F_1, F_2, \ldots, F_m$ are prox-regular, then we can refine our understanding of local convergence for the method of averaged projections using a completely different approach, explored in this section.

Proposition 6.8.1. *Around any point $\bar{x}$ at which the set $F \subset \mathbb{E}$ is prox-regular, the squared distance to F is continuously differentiable, and its gradient $\nabla d_F^2 = 2(I - P_F)$ has Lipschitz constant 2.*

Proof This result corresponds essentially to [146, Prop 3.1], which yields the smoothness of d_F^2 together with the gradient formula. This proof of this proposition also shows that for any sufficiently small $\delta > 0$, all points $x_1, x_2 \in \mathbb{E}$ near $\bar{x}$ satisfy the inequality

$$\langle x_1 - x_2, P_F(x_1) - P_F(x_2) \rangle \geq (1 - \delta)\|P_F(x_1) - P_F(x_2)\|^2$$

(see "Claim" in [146, p. 5239]). Consequently we have

$$\begin{aligned}
&\|(I - P_F)(x_1) - (I - P_F)(x_2)\|^2 - \|x_1 - x_2\|^2 \\
&\quad = \|(x_1 - x_2) - (P_F(x_1) - P_F(x_2))\|^2 - \|x_1 - x_2\|^2 \\
&\quad = -2\langle x_1 - x_2, P_F(x_1) - P_F(x_2) \rangle + \|P_F(x_1) - P_F(x_2)\|^2 \\
&\quad \leq (2\delta - 1)\|P_F(x_1) - P_F(x_2)\|^2 \ \leq \ 0,
\end{aligned}$$

provided we choose $\delta \leq 1/2$. □

As before, consider sets $F_1, F_2, \ldots, F_m \subset \mathbb{E}$ and a point $\bar{x} \in \cap_i F_i$, but now let us suppose moreover that each set F_i is prox-regular at $\bar{x}$. Define a function $f \colon \mathbb{E} \to \mathbb{R}$ by

$$f = \frac{1}{2m} \sum_{i=1}^{m} d_{F_i}^2. \tag{6.27}$$

This function is half the *mean-squared-distance* from the point x to the set system $\{F_i\}$. According to the preceding result, f is continuously differentiable around $\bar{x}$, and its gradient

$$\nabla f = \frac{1}{m} \sum_{i=1}^{m} (I - P_{F_i}) = I - \frac{1}{m} \sum_{i=1}^{m} P_{F_i} \tag{6.28}$$

is Lipschitz continuous with constant 1 on a neighborhood of $\bar{x}$. The method of averaged projections constructs the new iterate $x_+ \in \mathbb{E}$ from the old iterate $x \in \mathbb{E}$ via the update

$$x_+ = \frac{1}{m} \sum_{i=1}^{m} P_{F_i}(x) = x - \nabla f(x), \tag{6.29}$$

so we can interpret it as the method of steepest descent with a step size of one when the sets F_i are all prox-regular. To understand its convergence, we return to our linear regularity assumption.

The condition modulus controls the behavior of normal vectors not just at the point $\bar{x}$ but also at nearby points.

Proposition 6.8.2 (local effect of condition modulus)**.** *Consider closed sets $F_1, F_2, \ldots, F_m \subset \mathbb{E}$ having linearly regular intersection at a point $\bar{x} \in \cap F_i$, and any constant $k > \text{cond}(F_1, F_2, \ldots, F_m | \bar{x})$. Then for any points $x_i \in F_i$ sufficiently near $\bar{x}$, any vectors $y_i \in N_{F_i}(x_i)$ (for $i = 1, 2, \ldots, m$) satisfy the inequality*

$$\sqrt{\sum_i \|y_i\|^2} \leq k \Big\| \sum_i y_i \Big\|.$$

Proof If the result fails, then we can find sequences of points $x_i^r \to \bar{x}$ in F_i and sequences of vectors $y_i^r \in N_{F_i}(x_i^r)$ (for $i = 1, 2, \ldots, m$) satisfying

$$\sqrt{\sum_i \|y_i^r\|^2} > k \Big\| \sum_i y_i^r \Big\|$$

for all $r = 1, 2, \ldots$. Define new vectors

$$u_i^r = \frac{1}{\sqrt{\sum_j \|y_j^r\|^2}}\, y_i^r \in N_{F_i}(x_i^r)$$

for each index $j = 1, 2, \ldots, m$ and r. Notice $\sum_i \|u_i^r\|^2 = 1$ and $\| \sum_i u_i^r \| < \frac{1}{k}$. For each $i = 1, 2, \ldots,$ the sequence $u_i^1, u_i^2, \ldots$ is bounded, so after taking subsequences we can suppose it converges to some vector $u_i \in \mathbb{E}$, and since the normal cone N_{F_i} is closed as a set-valued mapping from F_i to $\mathbb{E}$, we deduce $u_i \in N_{F_i}(\bar{x})$. But then we have $\sum_i \|u_i\|^2 = 1$ and $\| \sum_i u_i \Big\| \leq \frac{1}{k}$, contradicting the definition of the modulus $\text{cond}(F_1, F_2, \ldots, F_m|\bar{x})$. □

The size of the gradient of the mean-squared-distance function f, defined by equation (6.27), is closely related to the value of the function near a point of linearly regular intersection. To be precise, we have the following result.

Proposition 6.8.3 (gradient of mean-squared-distance)**.** *Consider prox-regular sets $F_1, F_2, \ldots, F_m \subset \mathbb{E}$ having linearly regular intersection at a point $\bar{x} \in \cap F_i$, and any constant $k > \text{cond}(F_1, F_2, \ldots, F_m|\bar{x})$. Then on a neighborhood of $\bar{x}$, the mean-squared-distance function*

$$f = \frac{1}{2m} \sum_{i=1}^m d_{F_i}^2$$

satisfies the inequalities

$$\frac{1}{2}\|\nabla f\|^2 \leq f \leq \frac{k^2 m}{2}\|\nabla f\|^2. \tag{6.30}$$

Proof Consider any point $x \in \mathbb{E}$ sufficiently near $\bar{x}$. Equation (6.28) implies $\nabla f(x) = \frac{1}{m}\sum_i y_i$, where $y_i = x - P_{F_i}(x) \in N_{F_i}(P_{F_i}(x))$ for each $i = 1, 2, \ldots, m$. By definition, we have $f(x) = \frac{1}{2m}\sum_i \|y_i\|^2$. Using inequality (6.4), we obtain

$$m^2\|\nabla f(x)\|^2 = \Big\| \sum_{i=1}^m y_i \Big\|^2 \leq m \sum_{i=1}^m \|y_i\|^2 = 2m^2 f(x)$$

But since x is sufficiently near $\bar{x}$, so are the projections $P_{F_i}(x)$, so

$$2mf(x) = \sum_i \|y_i\|^2 \leq k^2 \Big\| \sum_i y_i \Big\|^2 = k^2 m^2 \|\nabla f(x)\|^2.$$

by Proposition 6.8.2. The result now follows. □

A standard argument now gives the main result of this section.

Theorem 6.8.4 (Q-linear convergence for averaged projections)**.** *Consider prox-regular sets $F_1, F_2, \ldots, F_m \subset \mathbb{E}$ having linearly regular intersection at a point $\bar{x} \in \cap F_i$, and any constant $k >$ cond$(F_1, F_2, \ldots, F_m|\bar{x})$. Then, for any averaged projection sequence $\{x_j\}$ with initial point x_0 sufficiently near $\bar{x}$, the mean-squared-distance*

$$f = \frac{1}{2m} \sum_{i=1}^{m} d_{F_i}^2$$

is reduced by at least a constant factor at each iteration:

$$f(x_{j+1}) \leq \Big(1 - \frac{1}{k^2 m}\Big) f(x_j) \quad (j = 0, 1, 2, \ldots).$$

Proof Consider any point $x \in \mathbb{E}$ near $\bar{x}$. The function f is continuously differentiable around the minimizer $\bar{x}$, so the gradient $\nabla f(x)$ must be small, and hence the new iterate $x_+ = x - \nabla f(x)$ must also be near $\bar{x}$. Hence, as we observed after equation (6.28), the gradient ∇f has Lipschitz constant one on a neighborhood of the line segment $[x, x_+]$. Consequently, a standard argument in optimization (see [139]) leads to

$$f(x_+) - f(x) \leq -\frac{1}{2}\|\nabla f(x)\|^2 \leq -\frac{1}{k^2 m} f(x),$$

using Proposition 6.8.3. □

A simple induction argument now gives an independent proof in the prox-regular case that the method of averaged projections converges linearly to a point in the intersection of the given sets. Specifically, the result above shows that mean-squared-distance $f(x_k)$ decreases by at least a constant factor at each iteration, and Proposition 6.8.3 shows that the size of the step $\|\nabla f(x_k)\|$ also decreases by a constant factor. Hence the sequence (x_k) must converge R-linearly to a point in the intersection.

Comparing this result to Theorem 6.7.2 (linear convergence of averaged projections), we see that the predicted rates of linear convergence are the same. Theorem 6.7.2 guarantees that the squared distance to the intersection converges to zero with R-linear rate c^2 (for any constant $c \in (\bar{c}, 1)$). The argument gives no guarantee about improvements in a particular iteration: it only describes the asymptotic behavior of the iterates. By contrast, the argument of Theorem 6.8.4, with the added assumption of prox-regularity, guarantees the same behavior but with the stronger information that the mean-squared-distance decreases monotonically to zero with Q-linear rate c^2. In particular, each iteration must decrease the mean-squared-distance.

6.9 Numerical example

In this final section, we give a numerical illustration showing the linear convergence of alternating and averaged projections algorithms. Some major problems in signal or image processing come down to reconstructing an object from as few linear measurements as possible. Several recovery procedures from randomly sampled signals have been proved to be effective when combined with sparsity constraints (see for instance the recent developments of compressed sensing [45],[71]). These optimization problems can be cast as linear programs. However for extremely large and/or nonlinear problems, projection methods become attractive alternatives. In the spirit of compressive sampling we use projection algorithms to optimize the compression matrix. This speculative example is meant simply to illustrate the theory rather than make any claim on real applications.

We consider the decomposition of images $x \in R^n$ as $x = Wz$ where $W \in \mathbb{R}^{n \times m}$ ($n < m$) is a "dictionary" (that is, a redundant collection of basis vectors). Compressed sensing consists in linearly reducing x to $y = Px = PWz$ with the help of a compression matrix $P \in \mathbb{R}^{d \times n}$ (with $d \ll n$); the inverse operation is to recover x (or z) from y. Compressed sensing theory gives sparsity conditions on z to ensure exact recovery [45],[71]. Reference [45] in fact proposes a recovery algorithm based on alternating projections (on two convex sets). In general, we might want to design a specific sensing matrix P adapted to W, to ease this recovery process. An initial investigation on this question is [75]; we suggest here another direction, inspired by [46], where averaged projections naturally appear.

Candes and Romberg [46] showed that, under orthogonality conditions, sparse recovery is more efficient when the entries $|(PW)_{ij}|$ are small. One could thus use the componentwise ℓ_∞ norm of PW as a measure of quality of P. This leads to the following feasibility problem: to find $U = PW$ such that $UU^\top = I$ and with the infinity norm constraint $\|U\|_\infty \leq \alpha$ (for a fixed tolerance α). The sets corresponding to these constraints are given by

$$\begin{aligned} L &= \{U \in \mathbb{R}^{d \times m} : U = PW\}, \\ M &= \{U \in \mathbb{R}^{d \times m} : UU^\top = I\}, \\ C &= \{U \in \mathbb{R}^{d \times m} : \|U\|_\infty \leq \alpha\}. \end{aligned}$$

The first set L is a subspace, the second set M is a smooth manifold while the third C is convex; hence the three are prox-regular. Moreover we can easily compute the projections. The projection onto the linear subspace L can be computed with a pseudo-inverse. The manifold M corresponds to the set of matrices U whose singular values are all ones; it turns out that naturally the projection onto M is obtained by computing the singular value decomposition of U, and setting singular values to 1 (apply for example Theorem 27 of [Mal-21]). Finally the projection onto C comes by shrinking entries of U (specifically, we operate $\min\{\max\{u_{ij}, -\alpha\}, \alpha\}$ for each entry u_{ij}). This feasibility problem can thus be treated by projection algorithms, and hopefully a matrix $U \in L \cap M \cap C$ will correspond to a good compression matrix P.

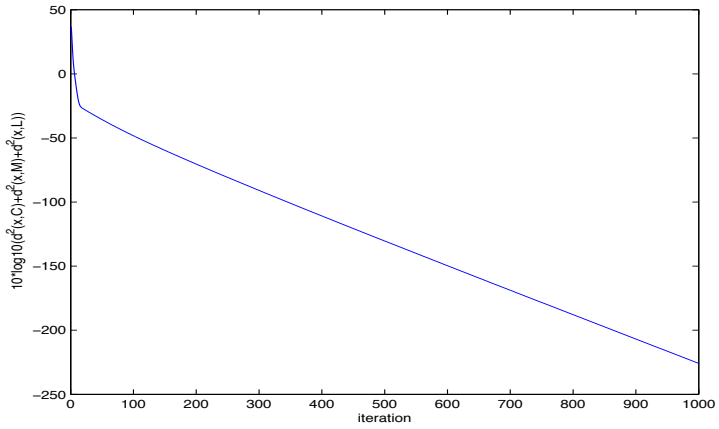


Figure 6.3: Convergence of averaged projection algorithm for designing compression matrix: plot of $10 \log_{10} f(U_k)$ vs the iteration number k.

To illustrate, we generate random entries (normally distributed) of the dictionary W (size 128×512, redundancy factor 4) and of an initial iterate $U_0 \in L$. (In practice, since the theory only guarantees *local convergence*, we would need a heuristic to find an initial iterate.) We fix $\alpha = 0.1$

and run the averaged projection algorithm, thereby computing a sequence of U_k that appear to be converging, as hoped, to a feasible solution to our problem. Furthermore the convergence appears linear: Figure 6.9 shows

$$10 \log_{10} f(U_k) \qquad \text{with} \qquad f(U) = d_L^2(U) + d_M^2(U) + d_C^2(U)$$

for each iteration k. We observe $f(U_{k+1})/f(U_k) < 0.9627$ for all k, suggesting the expected local Q-linear convergence. Random examples are interesting for our simple test of averaged projections: the challenging question of checking a priori the linear regularity of the intersection of the three sets is open, but randomness seems to prevent irregular solutions, providing α is not too small. So in this situation, we would hope that the algorithm will converge locally linearly; this is indeed what the numerical results in Figure 6.9 suggest. We note furthermore that we tested iterated projections on this problem (involving three sets, so not explicitly covered by Theorem 6.5.2). We observed that the method still appears locally linearly convergent in practice, and again, that the rate is better than for averaged projections.

Chapter 7

Conclusion, perspectives

7.1 Summary of presented contributions and perspectives

This document presents an excerpt from the research results that I have obtained since I received my PhD in November 2005. Chapter 1 gives an overview of my research activities, and explains their originality. Chapter 2 lists all my activities as a researcher, in the form of an extended curriculum vitæ. In the following chapters, I have chosen to focus on four contributions, each one illustrating a different aspect of my research. These contributions concern four subtopics: (i) semidefinite relaxations of combinatorial optimization problems, (ii) nonsmooth optimization algorithms for energy optimization, (iii) cut-generating functions in discrete optimization, and (iv) variational analysis of alternating projections. To conclude on these contributions, I would like to briefly put them here in a more general perspective and discuss some further research to which they gave birth.

Semidefinite optimization for binary quadratic problems Chapter 3 presents a semidefinite-based algorithm for solving binary quadratic optimization problems to optimality, which is the achievement of our research on practical use of semidefinite optimization for combinatorial problems. The resulting software called *BiqCrunch* is presented in this chapter, along with its basic ideas, its mathematical foundations, tips to efficiently use it, and illustrative numerical comparisons. *BiqCrunch* is freely available through the GNU Public License, version 3.0, as open source software available for non-commercial use. Distributing our code was a important milestone, but not the end of the story: several projects have started to improve it and enlarge its potential applications. I mention here four of them, ranked in an increasing degree of difficulty.

- Parallelization. Splitting the computation on N nodes would bring much more than a division by N of computing times: by generating more feasible solutions, we have more chances to improve the upper bound and then to prune parts of the branch-and-bound tree. Parallelization is possible with the branch-and-bound platform [55] used by *BiqCrunch*, but a number a technical details need to be addressed (in our own implementation and in the interface of the two codes).

- Specific version for QAP. We aim at attacking a "monster" of combinatorial optimization: the quadratic assignment problem [8]. This is a (NP-hard) problem which is hard to solve in practice; getting good results on it would definitely draw much attention and interest to our solver. Preliminary experiments show that computed semidefinite bounds are tight but that our solver has difficulties to find the optimal solution. We have identified two issues: 1) the binary search tree is unbalanced and 2) the used heuristics are not efficient enough.

- Integrating general cuts. The strengthening cuts manipulated by *BiqCrunch* are the so-called triangular inequalities that have a theoretical meaning from copositive optimization and a practical one as they are known to be very efficient for semidefinite relaxations in general. It would be great if an additional family of cuts could be specified by an advanced user (in the same way as already possible for heuristics). This brings however difficult questions on the selection of cuts and their dynamic management.

- Treating continuous variables. Semidefinite relaxation make a fundamental use of (the quadratic nature of the constraints and) the binary nature of the variables [147]. How to efficiently handle continuous variables in *BiqCrunch* is not clear; this would open a large field of applications.

Nonsmooth optimization with uncontrolled inexact information Chapter 4 presents convex nonsmooth optimization algorithms (prox- or level-bundle methods) able to take advantage of readily available additional linearizations with uncontrolled accuracy. There has been an active research on the use of inexact information within bundle methods and nonsmooth optimization; we refer for example to the foundational paper [60] unifying the convergence analysis of inexact proximal bundle methods. However all existing articles considered a bounded or vanishing inexactness. Our article [Mal-2] presented in this chapter was the first to explicitly mention and study uncontrolled inexact oracles in a context of bundle methods. It led to further reflexions by colleagues about uncontrolled oracles in discrete cases [175] and more recently to the continuous case with upper-oracles [174]. To me, the most exciting perspective opened by the development of this chapter is the design and analysis of *asynchronous* algorithms for stochastic optimization; I briefly described it below.

In parallel computing, a computational problem is solved by multiple agents that concurrently solve simpler subproblems and exchange information. In asynchronous computing, each agent can compute with the information it has, even if the latest information from other agents has not yet arrived. Asynchronism is extremely important to the efficiency and resilience of parallel computing. Today, the majority of optimization algorithms are still singled-threaded, and most of the already-parallelized algorithms are synchronous. For example, big problems in stochastic optimization are heavily structured, often amenable to parallel computing by standard decomposition schemes (e.g. by scenarios [155], by production units [74], or even both [Mal-5]). However existing optimization algorithms exploiting this decomposability are all synchronous. We have here a vast playground for distributed computing; this will be further developed in the next section.

Bundle methods are particularly well-suited for asynchronous generalizations: outdated information provided by a late agent can indeed be considered as uncontrolled linearization and treated as such by similar techniques as the one developed here. I have started investigating these questions with Welington de Oliveira. We are also considering with Franck Iutzeler an asynchronous version of the so-called progressive hedging algorithm [155] for multistage stochastic optimization. We believe that there is a bright future for applications of such distributed optimization algorithms to stochastic energy optimization problems (electricity generation, transport, and distribution, where uncertainty is due to intermittent renewable energy sources).

Cut-generating functions Chapter 5 introduces the theory of cut-generating functions, unifying a number of existing works on S-free sets, and recovering the celebrated Gomory cuts. This is a unique research, with developments both in discrete mathematics and in convex analysis, with a beautiful synergy between the two. Our ambition was to write a foundational paper in the intersection of discrete and continuous optimization with exciting potential use in mixed-integer software. This work opened several new perspectives which has already inspired strong follow-up papers, including [53], [112], and [183]. A number of theoretical questions indeed arise; let us mention some of them.

- Consistency. An important question is whether cut-generating functions do generate all possible cuts. This was formalized as open problem in [Mal-8], and a few months after its publication, the authors of [53] proved the conjecture. This is discussed in the conclusion of Chapter 5.

- Generalizations. One might want to consider a more general set X with a similar structure as (5.1a). For example, the first option is to replace the "ground set" $\mathbb{R}^n_+$ by some other closed convex cone, opening the way toward cutting mixed-integer conic problems. The recent article [112] goes in this direction for second-order cones. Another generalization would be inspired by the situation of Example 5.1.1: there, X has the form $\left\{x \in \mathbb{Z}^n_+ : Ax \in \mathbb{Z}^m - b\right\}$; the set $S = \mathbb{Z}^m - b$ lies in a smaller space but the ground set $\mathbb{Z}^n_+$ is no longer convex, so sublinear cut-generating functions are now ruled out (see section 5.2). Instead, cut-generating functions in this context are subadditive, periodic, and satisfy a certain symmetry condition [88]. The recent article [183] studies the case of integers variables x; there is still work to do to cover the general case of cut-generating functions in this framework.

- Related convex analysis questions. A couple of questions come naturally from section 5.3. Given a convex compact set G, can we detect whether it is the minimal prepolar of $V := G^\circ$? Could we link our "prepolar" (defined by (5.23)) with the sets studied in [184] ? These questions are limited to pure convex analysis and are basic enough to possibly find corollaries in other branches of mathematics.

Variational analysis of alternating projections methods Chapter 6 gives new insight to the study of alternating projection algorithms and establishes the first (linear) convergence guarantees in absence of convexity. The original proof techniques are based on the variational analysis of nonsmooth coupling functions (rather than convex analysis tools as all the previous works). The convergence proofs combine two ingredients:

(i) a geometrical (transversality-like) condition of the intersection, controlling the "angle" between the two sets at a point of the intersection;

(ii) a geometrical (convexity-like) property of one of the two sets, controlling the behaviour of projections.

The corresponding article [Mal-20] has opened a new line of research on non-convex projections methods; it has got around 100 citations, seven years later. In particular, the local convergence results have been further refined and extended in several papers (see e.g. [10], [21]) by lightening one of the two assumptions (i) and (ii) above. On one hand, the idea is to strengthen (ii) to get rid of the limitation of the transversality (i): for example, [5] proposes a new concept called non-tangential intersection point for manifolds; [73] shows that local linear convergence holds with no transversality assumption when the two sets are semi-algebraic and bounded. On the other hand, the opposite idea is to remove (ii): for example, [73] proves convergence under only the transversality assumption using the slope of a natural nonsmooth coupling function; [34] studied converge of a broad family of alternating algorithms (proximal alternating linearized minimization algorithm for solving a class of nonconvex and nonsmooth minimization problems) using another non-degeneracy assumption (building on [11]).

The above-mentionned assumption is called Kurdyka-Lojasiewicz: roughly speaking, it allows to turn estimates on the norm of (sub)gradients of a function f into estimates on the values of f itself. Thus the assumption (or its variants [110] or extensions [33]) plays a key role in the analysis of first-order methods in non-convex setting. I would be curious to know if there exist connections between them and the metric regularity (for minimization problems), which is the assumption that I have used

so far in my research on non-convex algorithms. Another research direction would be to combine these assumptions and related techniques with standard convex complexity analysis of algorithms (see e.g. [139]) in order to get rates of convergence towards stationary points or local minima. This topic and, more generally nonconvex optimization[1], are becoming an active research area in the machine learning community, see e.g. [123], [124] and [110].

7.2 Personal research perspectives

All my publications contains openings and perspectives for future research; some of them, related to the material presented in this document, are presented in the previous section. There is work there for several years! Besides, some new research directions have recently drawn my curiosity and attention, in view of their huge potential usefulness. I would like to put an emphasis in this section on two of them, that will attract my work and attention in a near future. Their presentation goes in two stages: a contextual positioning and a brief description of my related projects.

7.2.1 Probability-constrained optimization in action

Context: Facing uncertainty in optimization Data uncertainty is an inherent feature of optimization problems and has to be taken into account in optimization and decision-making tools. Uncertainty can come from noisy data (e.g. from high-volatility financial markets), unreliable data (e.g. imprecise ratings in collaborative filtering), or uncertain predictions (e.g. different scenarios in forecasting). A typical example is the recent trend in electricity generation: the growing incorporation of renewable energy sources (wind, solar) in the electricity park has dramatically increased the level of uncertainty when optimizing electricity generation [170]. The two main ways to model uncertainty are:

- Robust optimization (see e.g. [26]). Assuming that the uncertain variable ξ has a given uncertainty set ($\xi \in \Xi$), we optimize the worst possible situation over Ξ:

$$\begin{cases} \min_x \quad \max\limits_{\xi \in \Xi} \; f(x,\xi) \\ \quad g(x,\xi) \leq 0 \quad \text{for all } \xi \in \Xi \end{cases} \tag{7.1}$$

- Stochastic optimization (see e.g. [160]): Assuming that the random variable ξ has a given probability law ($\xi \sim \mathbb{P}$ known, or partly known up to parameters or from observations), we want a solution with a good expected objective value. This gives typically problems of the form:

$$\begin{cases} \min_x \quad \mathbb{E}[f(x,\xi)] \\ \quad \mathbb{P}[g(x,\xi) \leq 0] \geq p \end{cases} \tag{7.2}$$

  for a defined safety level $p \in [0,1]$. For example, $p = 1$ means that the constraint should be satisfied almost-surely.

When facing an uncertain optimization problem, modeling is as important as solving the resulting optimization problem. In recent years, considerable progress has been made due to a better understanding of modeling issues and the development of new algorithms, as those based on randomization

[1] see also the recent NIPS workshops on optimization, e.g. in 2014 `http://opt-ml.org/oldopt/opt14/invited.html` or in 2016 `https://sites.google.com/site/nonconvexnips2016/home`

techniques or those based on bundle methods. But the gap between theoretical promises and numerical tractability is still large.

For example, modeling uncertainty with probability constraints as in (7.2) is attractive, as attested by the applications in fields as diverse as energy, telecommunications, or chemical engineering; see references in the textbook [166]. However optimization problems having such constraints are very hard to solve in general, for both theoretical and computational reasons. First, the analytical properties of the probability function $x \mapsto \mathbb{P}[g(x,\xi) \leq 0]$ such as differentiability or convexity are not immediately derived from nominal properties of g. Second, numerical approximations of the values (and the gradients, when available) of these functions require heavy noisy computations.

Project: theory and algorithms of probability-constrained optimization The aim of this research project, with my colleague from EDF Wim van Ackooij, is to reduce the gap between what we can model and what we can solve in stochastic optimization. With a special interest to probability constraints, we will consider several theoretical and algorithmic questions (e.g getting out of the convex setting of bundle algorithms for probability-constrainted problems [173, 178]) and application-oriented theoretical questions (e.g. expand the differentiability or convexity properties of probability functions [172, 176], [Mal-5]).

Let us detail one of these mathematical questions with practical impact when solving (7.2): the possible convexity of the the feasible set of (7.2). All existing results about the convexity of such a set are restricted to special separable problems or under restrictive assumptions. For example, a celebrated result by András Prékopa[2] asserts convexity of the feasible set provided g is jointly quasi-convex and ξ admits a density with generalized concavity properties. The restrictive assumption here is that the quasi-convexity of g, which fails many situation of interests (as a linear constraint $A(\xi)x \leq b$). It would be useful to have a result of "eventual convexity" guaranteeing that the set $\{x : \mathbb{P}[g(x,\xi) \leq 0] \geq p\}$ is convex for all $p > p^*$, where p^* is a threshold that could be estimated or even computed. We are working to establish such a result without restrictive assumptions.

7.2.2 Distributed optimisation, from decomposable algorithms to efficient systems

Context: enter in the Big Data era The explosion of data collection and processing systems has triggered a lot of research and developments towards the exploitation of huge amounts of information. Because of the data size, every elementary operation such as storage, communication, and, most importantly, processing has to be looked into again in the light of the physical limitations and bottlenecks of computing systems. An effort has been conducted both in the industry and academia to redefine paradigms and good practices concerning large-scale data processing, for example with Google's MapReduce (2004), Apache's Hadoop (2009) and Spark (2014) systems. Parallel computing for numerical linear algebra have a (relatively!) long history and distributed computing for machine learning is an emerging hot topic. For example, Spark's MLlib library already contains ready to use machine learning algorithms, including some basic optimization algorithms (like stochastic gradient descent[3]).

Mathematical optimization is experiencing a similar shift of paradigm to face big data challenges. Optimization algorithms indeed face new challenges raised by the explosion in size and complexity of optimization problems; notably when dealing with large-scale inverse problems that arise in signal processing, medical imaging, and machine learning. Distributed optimization algorithms have re-

[2]While working on this conclusion chapter, I heard that András Prékopa passed away. I would like to pay a tribute here to his pioneer work on probability functions and his special interest to applications.

[3]https://spark.apache.org/docs/1.1.0/mllib-guide.html

cently emerged to decompose computation in a tractable, parallel, or distributed manner over clusters (see e.g ADMM [40] or the random coordinate descent methods for big data optimization [154]).

Project: theory and algorithms for distributed computing in optimization I will put parts of my work and attention on scalable optimization algorithms taking advantage of the development of distributed computing. The goal is to reduce the gap between mathematical optimization algorithms and efficient distribution of computation. Among the many numerical difficulties, let us mention:

- The distribution of optimization problems is obtained at the expense of an important increase in the problem size, which directly reduces the convergence speed of the associated algorithm.
- Parallel and distributed algorithms require variables (if not data) transmission between workers and an eventual coordinator; these exchanges can be costly in terms of time delay or storage and usually present an important practical bottleneck. Also, it is highly likely that some transmissions and/or computations fail at some point (e.g. hardware failures in the cluster).
- Synchronism can be a burden in such systems with many potentially heterogeneous workers: waiting for everyone of them to finish computing before assigning new tasks is limiting. Algorithms should also be adapted to deal with computations made using outdated data.

These questions should be looked at with a special attention to properly leverage on the specificities of optimization (e.g., optimization problems themselves are strongly structured, and advanced optimization algorithms have the ability to self-heal from one iteration to another, or the ability to handle inexact computations, see e.g. [60]).

I will team up with passionate colleagues (and good friends) from Grenoble to contribute on this multidisciplinary domain. In particular, I am very happy to have Nabil Layaida in my committee, perfect allegory of this research project. We all aim at investigating design, complexity analysis, high-performing implementations, and real deployment of distributed advanced optimization algorithms.

7.3 Team research perspectives

My personal research projects, presented in the previous two sections, are part of the more general research direction of my team. Last year indeed, I created a new team in my lab LJK, DAO "optimization and learning for data science", gathering brilliant researchers on optimization, machine learning, statistics and their interplay. In this section, I present DAO and our research perspectives.

DAO research team Data science aims at extracting information from heterogeneous, dynamical, or massive data-bases. The scientific challenges cover all the processing chain from data collection to analysis and interpretation. Thus a domain "data science" has recently emerged as a unifying scientific discipline blending techniques and theories from many scientific fields including information theory, computer science and mathematics. Expected impacts on science, economy and society are of paramount importance. LJK has a top-level research on mathematical methods of data science as well as expertise in some applications (e.g. computer vision, or oceanic flow modeling).

The objective of the new LJK team is to structure the activity on mathematical methods for data science on the interplay between mathematical optimization and machine learning. We want to gather researchers, foster exchanges, attract students, solidify collaborations, and highlight successful research. Recent publications, as well as a selection of research projects are given on the team's website `http://dao-ljk.imag.fr`.

Figure 7.1: Cloud of words from the document of creation the research team DAO at LJK

DAO research perspectives As a team, DAO has several research projects on the interplay between mathematical optimization and statistical learning, ranging from theoretical analysis of problems and algorithms to their numerical applications on real-life data. I highlight three main directions on which my personal projects take part.

- **Stochastic algorithms in force.** Stochastic optimization is a key subject in DAO on various aspects: e.g. stochastic optimal control and approximation [118], efficient algorithms to stochastic two-stage optimization problems [Mal-5], modelisation of dependency in temporal series [51], as well as adaptative monte-carlo methods [15]. Our first research direction is thus to capitalize on this rich background to push further the develop of stochastic methods for data science applications. My personal project on probability-constrained optimization fits into this dynamic.

- **Turning large-scale data sets into interpretable statistical models.** This research direction is twofold. First, we aim at proposing new theoretically and computationally solid techniques for a wide class of statistical problems. This is to be done by building a new framework based on computable risk bounds for analysis [85, 109] and efficient numerical implementation of convex optimization methods. Second, we will work on improving the scalability of optimization algorithms to solve large-scale statistical inference problems. We will pursue the development of advanced, computationally-cheap first-order optimization techniques [92] or distributed primal-dual algorithms [27] capable of solving corresponding large-scale data science problems.

- **Understand and optimize over networks.** People, devices, and data are getting more or more connected. These networks have different scales and tightness: from micro and dense (clusters), meso and more loose (grid computing), up to macro and complex (social networks). In view of the increasing complexity of the data and interplay between the agents, computing and processing over these networks has to be performed by adapted decentralized algorithms. In addition, it is often interesting to consider stochastic versions of distributed algorithms; for exemple, splitting large-scale datasets and processing randomly chosen batches is a popular technique in machine learning [39]. The goal of this research is to develop and analyze efficient methods retrieving and computing information over graphs, or performing learning from data scattered over a network [27], let it be physical (databases spread over different physical places) or virtual (in the case of high performance computing). My personal project on efficient distributed optimization fits into this more general project.

Bibliography

[1] T. Achterberg. SCIP: Solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41, 2009.

[2] K. Andersen, Q. Louveaux, R. Weismantel, and L. Wolsey. Cutting planes from two rows of a simplex tableau. In M. Fischetti and D. Williamson, editors, *Integer Programming and Combinatorial Optimization*, volume 4513 of *Lecture Notes in Computer Science*, pages 1–15. Springer Verlag, 2007.

[3] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.

[4] G. Anderson, L. Goldberg, A.N. Kercheval, G. Miller, and K. Sorge. On the aggregation of local risk models for global risk management. *Journal of Risk*, 8:25–40, 2005.

[5] F. Andersson and M. Carlsson. Alternating projections on nontangential manifolds. *Constructive Approximation*, 38(3):489–525, 2013.

[6] M. Anjos and J.B. Lasserre. *Handbook of semidefinite, conic and polynomial optimization*. Springer, 2012.

[7] M. Anjos and A. Vannelli. Computing globally optimal solutions for single-row layout problems using semidefinite programming and cutting planes. *INFORMS Journal on Computing*, 20(4):611–617, 2008.

[8] K. Anstreicher. Recent advances in the solution of quadratic assignment problems. *Mathematical Programming*, 97(1-2):27–42, 2003.

[9] M. Armbruster, M. Fügenschuh, C. Helmberg, and A. Martin. Lp and sdp branch-and-cut algorithms for the minimum graph bisection problem: a computational comparison. *Mathematical Programming Computation*, 4(3):275–306, 2012.

[10] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the kurdyka-lojasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010.

[11] H. Attouch, J. Bolte, and B. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods. *Mathematical Programming*, 137(1-2):91–129, 2013.

[12] A. Auslender. *Méthodes Numériques pour la Résolution des Problèmes d'Optimisation avec Constraintes*. PhD thesis, Faculté des Sciences, Grenoble, 1969.

[13] D. Aussel, A. Daniilidis, and L. Thibault. Subsmooth sets: functional characterizations and related concepts. *Trans. Amer. Math. Soc.*, 357:1275–1301, 2005.

[14] G. Averkov. On maximal S-free sets and the Helly number of the family of S-convex sets. *SIAM J. on Disc. Math.*, 27:1610–1624, 2013.

[15] L. Badouraly Kassim, J. Lelong, and I. Loumrhari. Importance sampling for jump processes and applications to finance. *Journal of Computational Finance*, to appear, 2014.

[16] F. Barahona, M. Jünger, and G. Reinelt. Experiments in quadratic 0–1 programming. *Mathematical Programming*, 44(1):127–137, 1989.

[17] A. Basu, M. Conforti, G. Cornuéjols, and G. Zambelli. Maximal lattice-free convex sets in linear subspaces. *Math. Oper. Res.*, 35(3):704–720, 2010.

[18] A. Basu, M. Conforti, G. Cornuéjols, and G. Zambelli. Minimal inequalities for an infinite relaxation of integer programs. *SIAM J. on Disc. Math.*, 24(1):158–168, 2010.

[19] A. Basu, G. Cornuéjols, and G. Zambelli. Convex sets and minimal sublinear functions. *Journal of Convex Analysis*, 18(2):427–432, 2011.

[20] H. Bauschke and J. Borwein. On projection algorithms for solving convex feasibility problems. *SIAM review*, 38(3):367–426, 1996.

[21] H. Bauschke, D. R. Luke, H. Phan, and X. Wang. Restricted normal cones and the method of alternating projections: theory. *Set-Valued and Variational Analysis*, 21(3):431–473, 2013.

[22] H.H. Bauschke and J.M. Borwein. On the convergence of von Neumann's alternating projection algorithm for two sets. *Set Valued Analysis*, 1(2):185–212, 1993.

[23] H.H. Bauschke, P.L. Combettes, and D.R. Luke. Phase retrieval, error reduction algorithm, and Fienup variants: A view from convex optimization. *Journal of the Optical Society of America*, 19(7), 2002.

[24] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[25] C. Beltran, C. Tadonki, and J.Ph. Vial. Solving the p-median problem with a semi-lagrangian relaxation. *Computational Optimization and Applications*, 35(2), 2006.

[26] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton University Press, 2011.

[27] P. Bianchi, W. Hachem, and F. Iutzeler. A stochastic coordinate descent primal-dual algorithm and applications to distributed optimization. *to appear in IEEE Transactions on Automatic Control*, 2015.

[28] A. Billionnet. Different formulations for solving the heaviest k-subgraph problem. *Information Systems and Operational Res.*, 43(3):171–186, 2005.

[29] A. Billionnet and S. Elloumi. Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem. *Mathematical Programming*, 109(1):55–68, 2007.

[30] A. Billionnet, S. Elloumi, and M.-C. Plateau. Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: The QCR method. *Discrete Applied Mathematics*, 157(6):1185–1197, 2009.

[31] R. Bixby and E. Rothberg. Progress in computational mixed integer programming – a look back from the other side of the tipping point. *Ann. Op. Res.*, 149(1):37–41, 2007.

[32] J. Bolte, A. Daniilidis, and A.S. Lewis. Tame functions are semismooth. *Math. Program.*, 117(1):5–19, 2008.

[33] J. Bolte, A. Daniilidis, O. Ley, and L. Mazet. Characterizations of lojasiewicz inequalities: subgradient flows, talweg, convexity. *Transactions of the American Mathematical Society*, 362(6):3319–3363, 2010.

[34] J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2):459–494, 2014.

[35] P. Bonami, L. Biegler, A. Conn, G. Cornuéjols, I. Grossmann, C. Laird, J. Lee, A. Lodi, F. Margot, and N. Sawaya. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5(2):186–204, 2008.

[36] J.F. Bonnans, J.Ch. Gilbert, C. Lemaréchal, and C. Sagastizábal. *Numerical Optimization*. Springer Verlag, 2003.

[37] B. Borchers. CSDP, a C library for semidefinite programming. *Optimization Methods and Software*, 11(1):613–623, 1999.

[38] V. Borozan and G. Cornuéjols. Minimal valid inequalities for integer constraints. *Math. Oper. Res.*, 34(3):538–546, 2009.

[39] L. Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9):25, 1998.

[40] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundation and Trends in Machine Learning*, 3:1–122, 2011.

[41] U. Brannlund, K. C. Kiwiel, and P. O. Lindberg. A descent proximal level bundle method for convex nondifferentiable optimization. *Operations Research Letters*, 17(3):121 – 126, 1995.

[42] L.M. Bregman. The method of successive projection for finding a common point of convex sets. *Soviet Math. Dokl.*, 6:688–692, 1965.

[43] S. Burer and A. Letchford. Non-convex mixed-integer nonlinear programming: A survey. *Surveys in Operations Research and Management Science*, 17(2):97 – 106, 2012.

[44] M. Bussieck, S. Vigerske, J. Cochran, L. Cox, P. Keskinocak, J. Kharoufeh, and J. Smith. *MINLP Solver Software*. John Wiley, Inc., 2010. Updated Feb 21, 2012.

[45] E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006.

[46] E. J. Candès and J. Romberg. Sparsity and incoherence in compressive sampling. *Inv. Prob.*, 23(3):969–986, 2007.

[47] E. Caprari and A. Zaffaroni. Conically equivalent convex sets and applications. *Pac. J. Optim.*, 6:281–303, 2010.

[48] X. Chen and M. Chu. On the least squares solution of inverse eigenvalue problems. *SIAM Journal on Numerical Analysis*, 33(6):2417–2430, 1996.

[49] M. Chu. Constructing a hermitian matrix from its diagonal entries and eigenvalues. *SIAM Journal on Matrix Analysis and Applications*, 16(1):207–217, 1995.

[50] F.H. Clarke, Yu.S. Ledyaev, R.J. Stern, and P.R. Wolenski. *Nonsmooth Analysis and Control Theory*. Springer-Verlag, New York, 1998.

[51] M. Clausel, J.F. Coeurjolly, and J. Lelong. Stein estimation of the intensity of a spatial homogeneous poisson point process. *To appear in Annals of Applied Probability (2015)*, 2015.

[52] P.L. Combettes and H.J. Trussell. Method of successive projections for finding a common point of sets in metric spaces. *Journal of Optimization Theory and Applications*, 67(3):487–507, 1990.

[53] Gérard Cornuéjols, Laurence Wolsey, and Sercan Yıldız. Sufficiency of cut-generating functions. *Mathematical Programming*, 152(1-2):643–651, 2015.

[54] F. Cucker and D. Zhou. *Learning Theory: An Approximation Theory Viewpoint*. Cambridge University Press, 2007.

[55] B.L. Cun and C. Roucairol. BOB: A unified platform for implementing branch-and-bound like algorithms. Technical report, University of Versailles Saint-Quentin-en-Yvelines, 1995.

[56] C. D'Ambrosio and A. Lodi. Mixed integer nonlinear programming tools: a practical overview. *4OR*, 9(4):329–349, 2011.

[57] A. Daniilidis, D. Drusvyatskiy, and A. Lewis. Orthogonal invariance and identifiability. *SIAM Journal on Matrix Analysis and Applications*, 35(2):580–598, 2014.

[58] Aris Daniilidis, Jérôme Malick, and Hristo Sendov. Locally symmetric submanifolds lift to spectral manifolds. Technical report, December 2012.

[59] W. de Oliveira and C. Sagastizàbal. Level bundle methods for oracles with on-demand accuracy. *Optimization Methods and Software*, 29(6):1180–1209, 2014.

[60] W. de Oliveira, C. Sagastizabal, and C. Lemaréchal. Convex proximal bundle methods in depth: a unified analysis for inexact oracles. *Mathematical Programming*, 148(1-2):241–277, 2014.

[61] W. de Oliveira, C. Sagastizábal, D. Penna, M. Maceira, and J. Damázio. Optimal scenario tree reduction for stochastic streamflows in power generation planning problems. *Optimization Methods and Software*, 25(6):917–936, 2010.

[62] W. de Oliveira and M. Solodov. A doubly stabilized bundle method for nonsmooth convex optimization. *Mathematical Programming*, pages 1–35, 2015.

[63] I. Deák. Two-stage stochastic problems with correlated normal variables: computational experiences. *Annals OR*, 142(1):79–97, 2006.

[64] D. Dentcheva and G. Martinez. Regularization methods for optimization problems with probabilistic constraints. *Math. Programming (series A)*, 138(1-2):223–251, 2013.

[65] Jacques Desrosiers and Marco Lobbecke. A primer in column generation. In G. Desaulniers, J. Desrosiers, and M.. Solomon, editors, *Column Generation*, pages 1–32. Springer US, 2005.

[66] F. Deutsch. *Best Approximation in Inner Product Spaces*. Springer, New York, 2001.

[67] F. Deutsch and H. Hundal. The rate of convergence for the cyclic projections algorithm i: angles between convex sets. *Journal of Approximation Theory*, 142(1):36–55, 2006.

[68] S. S. Dey and L. A. Wolsey. Constrained infinite group relaxations of mips. *SIAM J. on Opt.*, 20(6):2890–2912, September 2010.

[69] M. Deza and M. Laurent. *Geometry of Cuts and Metrics*, volume 15 of *Algorithms and Combinatorics*. Springer, Berlin, 1997.

[70] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91:201–213, 2002.

[71] D. Donoho. Compressed sensing. *IEEE Trans. Inform. Theory*, 52:1289–1306, 2006.

[72] A.L. Dontchev, A.S. Lewis, and T. Rockafellar. The radius of metric regularity. *Transactions of the American Mathematical Society*, 355(2):493–517, 2002.

[73] D Drusvyatskiy, AD Ioffe, and AS Lewis. Alternating projections and coupling slope. *arXiv preprint arXiv:1401.7569*, 2014.

[74] L. Dubost, R. Gonzalez, and C. Lemaréchal. A primal-proximal heuristic applied to the french unit-commitment problem. *Mathematical Programming*, 104:129–151, 2005.

[75] M. Elad. Optimized projections for compressed-sensing. *to appear in IEEE Trans. on Signal Processing*, 2006.

[76] A. Engau, M. Anjos, and A. Vannelli. On handling cutting planes in interior-point methods for solving semi-definite relaxations of binary quadratic optimization problems. *Optimization Methods and Software*, pages 1–21, 2012.

[77] C. Fábián. Bundle-type methods for inexact data. *Central European Journal of Operations Research*, 8:35–55, 2000.

[78] A. Faye and F. Roupin. Partial Lagrangian for general quadratic programming. *4'OR, A Quarterly Journal of Operations Research*, 5(1):75–88, 2007.

[79] A.L. Dontchev F.J. Aragón Artacho and M.H. Geoffroy. Convergence of the proximal point method for metrically regular mappings. *ESAIM Proceedings*, 17:1–8, 2007.

[80] A. Frangioni. Generalized bundle methods. *SIAM Journal on Optimization*, 13(1):117–156, 2002.

[81] L. Galli and A. Letchford. A compact variant of the qcr method for quadratically constrained quadratic 0–1 programs. *Optimization Letters*, 8(4):1213–1224, 2014.

[82] T. Gally, M. Pfetsch, and Stefan U. A framework for solving mixed-integer semidefinite programs. Technical report, Technische Universität Darmstadt, 2016.

[83] A.M. Geoffrion. Generalized Benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 1972.

[84] M. Goemans and D. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 6:1115–1145, 1995.

[85] A. Goldenshluger, A. Juditsky, and A. Nemirovski. Rejoinder of hypothesis testing by convex optimization. *Electronic Journal of Statistics*, 9(2):1744–1748, 2015.

[86] R. E. Gomory. An algorithm for integer solutions to linear programs. In R. L. Graves and P. Wolfe, editors, *Recent Advances in Mathematical Programming*, pages 269–302. McGraw-Hill, 1963.

[87] R. E. Gomory. Some polyhedra related to combinatorial problems. *Lin. Alg. Appl.*, 2(4):451–558, 1969.

[88] R. E. Gomory and E. L. Johnson. Some continuous functions related to corner polyhedra i. *Mathematical Programming*, 3:23–85, 1972.

[89] K.M. Grigoriadis and E. Beran. Alternating projection algorithm for linear matrix inequalities problems with rank constraints. In *Advances in Linear Matrix Inequality Methods in Control*. SIAM, 2000.

[90] K.M. Grigoriadis and R.E. Skelton. Low-order control design for LMI problems using alternating projection methods. *Automatica*, 32:1117–1125, 1996.

[91] L. Gubin, B. Polyak, and E. Raik. The method of projections for finding the common point of convex sets. *U.S.S.R. Computational Mathematics and Mathematical Physics*, 7:1–24, 1967.

[92] Z. Harchaoui, A. Juditsky, and A. Nemirovski. Conditional gradient algorithms for norm-regularized smooth convex optimization. *Mathematical Programming*, 152(1-2):75–112, 2015.

[93] G. Hechme-Doukopoulos, S. Brignol-Charousset, J. Malick, and C. Lemaréchal. Optimization of electricity production. *European Science Foundation, Mathematics and Industry: success stories*, 2010.

[94] G. Hechme-Doukopoulos, S. Brignol-Charousset, J. Malick, and C. Lemaréchal. The short-term electricity production management problem at EDF. *Optima, Newsletter of Mathematical Optimization Society*, 84, 2010.

[95] C. Helmberg and F. Rendl. Solving quadratic (0,1)-problems by semidefinite programs and cutting planes. *Mathematical Programming*, 82(3):291–315, 1998.

[96] C. Helmberg, F. Rendl, and R. Weismantel. A semidefinite programming approach to the quadratic knapsack problem. *Journal of Combinatorial Optimization*, 4(2):197–215, 2000.

[97] D. Henrion and A. Garulli. *Positive polynomials in control*. LNCIS, Springer Verlag, 2005.

[98] N. Higham. Computing a nearest symmetric positive semidefinite matrix. *Linear Algebra and its Applications*, 103:103–118, 1988.

[99] M. Hintermüller. A proximal bundle method based on approximate subgradients. *Computational Optimization and Applications*, 20:245–266, 2001. 10.1023/A:1011259017643.

[100] J.-B. Hiriart-Urruty. Les mathématiques du mieux faire, vol. 1: Premiers pas en optimisation. *Collection Opuscules, Ellipses*, 2007.

[101] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms*. Springer Verlag, Heidelberg, 1993. Two volumes.

[102] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms*. Springer Verlag, Heidelberg, 1993. Two volumes.

[103] J.-B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of Convex Analysis*. Springer Verlag, Heidelberg, 2001.

[104] L.D. Iasemidis, P. Pardalos, J.C. Sackellares, and D.-S. Shiau. Quadratic binary programming and dynamical system approach to determine the predictability of epileptic seizures. *Journal of Combinatorial Optimization*, 5:9–26, 2001.

[105] A.N. Iusem, T. Pennanen, and B.F. Svaiter. Inexact versions of the proximal point algorithm without monotonicity. *SIAM Journal on Optimization*, 13:1080–1097, 2003.

[106] E. L. Johnson. Characterization of facets for multiple right-hand side choice linear programs. *Math. Program. Study 14*, pages 112–142, 1981.

[107] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1943–1950, 2010.

[108] J. J. Júdice, H. Sherali, I. M. Ribeiro, and A. M. Faustino. A complementarity-based partioning and disjujnctive cut algorithm for mathematical programming problems with equilibrium constraints. *J. Global Opt.*, 136:89–114, 2006.

[109] A. Juditsky and A. Nemirovski. Nonparametric estimation by convex programming. *Annals of Statistics*, 37(5A):2278–2300, 10 2009.

[110] H. Karimi, J. Nutini, and M. Schmidt. *Linear Convergence of Gradient and Proximal-Gradient Methods Under the Polyak-Lojasiewicz Condition*, pages 795–811. Springer International Publishing, Cham, 2016.

[111] J. E. Kelley. The cutting plane method for solving convex programs. *J. Soc. Indust. Appl. Math.*, 8:703–712, 1960.

[112] F. Kılınç-Karzan. On minimal valid inequalities for mixed integer conic programs. *Mathematics of Operations Research*, 41(2):477–510, 2015.

[113] K. C. Kiwiel. A proximal bundle method with approximate subgradient linearizations. *SIAM Journal on Optimization*, 16(4):1007–1023, 2006.

[114] D. Klatte and B. Kummer. Optimization methods and stability of inclusions in banach spaces. *Mathematical Programming*, 117(1):305–330, 2009.

[115] A. Kruger. About regularity of collections of sets. *Set-Valued Analysis*, 14(2):187–206, 2006.

[116] M. Laurent and S. Poljak. On a positive semidefinite relaxation of the cut polytope. *Linear Algebra and its Applications*, 223–224:439 – 461, 1995.

[117] M. Laurent and S. Poljak. On the facial structure of the set of correlation matrices. *SIAM Journal on Matrix Analysis and Applications*, 17(3):530–547, 1996.

[118] J. Lelong. Asymptotic normality of randomly truncated stochastic algorithms. *ESAIM. Probability and Statistics*, 17:105–119, 2013.

[119] C. Lemaréchal. Lagrangian relaxation. In M. Jünger and D. Naddef, editors, *Computational Combinatorial Optimization*, pages 112–156. Springer Verlag, Heidelberg, 2001.

[120] C. Lemaréchal, A. Nemirovskii, and Y. Nesterov. New variants of bundle methods. *Math. Program.*, 69(1):111–147, 1995.

[121] C. Lemaréchal and F. Oustry. Semidefinite relaxations and Lagrangian duality with application to combinatorial optimization. Rapport de Recherche 3710, INRIA, 1999.

[122] A. Lewis. Convex analysis on the Hermitian matrices. *SIAM Journal on Optimization*, 6(1):164–177, 1996.

[123] G. Li and T. Pong. Global convergence of splitting methods for nonconvex composite optimization. *SIAM Journal on Optimization*, 25(4):2434–2460, 2015.

[124] G. Li and T. Pong. Calculus of the exponent of kurdyka-lojasiewicz inequality and its applications to linear convergence of first-order methods. *arXiv preprint arXiv:1602.02915*, 2016.

[125] F. Liers, M. Jünger, G. Reinelt, and G. Rinaldi. *Computing Exact Ground States of Hard Ising Spin Glass Problems by Branch-and-Cut*, pages 47–69. Wiley-VCH Verlag GmbH & Co. KGaA, 2005.

[126] H. Lin, J. Mairal, and Z. Harchaoui. A universal catalyst for first-order optimization. *NIPS*, 2015.

[127] L. Lovász. Geometry of numbers and integer programming. In M. Iri and K. Tanabe, editors, *Mathematical Programming: Recent Developements and Applications*, pages 177–210. Kluwer, 1989.

[128] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1(2):166–190, 1991.

[129] T. L. Magnanti and R. T. Wong. Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3):464–484, 1981.

[130] J. Malick. Convexité et combinatoire. *Bulletin de la ROADEF*, 22, 2009. available online at `https://hal.archives-ouvertes.fr/hal-00804111`.

[131] S. Mars. *Mixed-Integer Semidefinite Programming with an Application to Truss Topology Design*. PhD thesis, Technische Universitat Darmstadt, Germany, 2013.

[132] M. Minoux. Two-stage robust optimization, state-space representable uncertainty and applications. *RAIRO-Operations Research*, 48:455–475, 2014.

[133] J. Morales and J. Nocedal. Remark on "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization". *ACM Trans. Math. Softw.*, 38(1):1–4, 2011.

[134] D.A. Morán R. and S.S. Dey. On maximal S-free convex sets. *SIAM J. on Disc. Math.*, 25:379–393, 2011.

[135] B Sh Mordukhovich. Nonsmooth analysis with nonconvex generalized differentials and conjugate mappings. In *Doklady Akademii Nauk Belarusi*, volume 28, pages 976–979. ACADEMII NAUK BELARUSI F SCORINA PR 66, ROOM 403, MINSK, BYELARUS 220072, 1984.

[136] B.N. Mordukhovich. *Variational analysis and generalized differentiation*, volume 330 of *Grundlehren der mathematischen Wissenschaften*. Springer, 2006. (2 volumes).

[137] B.S. Mordukhovich. Maximum principle in the problem of time optimal response with nonsmooth constraints. *Journal of Applied Mathematics and Mechanics*, 40:960–969, 1976.

[138] B.S. Mordukhovich. Failure of metric regularity for major classes of variational systems. *Nonlinear Analysis*, 69:918–924, 2008.

[139] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[140] Y. Nesterov and A.S. Nemirovski. *Interior-Point Polynomial Algorithms in Convex Programming*. Number 13 in SIAM Studies in Applied Mathematics. SIAM, Philadelphia, 1994.

[141] Yu Nesterov. Semidefinite relaxation and nonconvex quadratic optimization. *Optimization methods and software*, 9(1-3):141–160, 1998.

[142] W. Oliveira, C. Sagastizábal, and S. Scheimberg. Inexact bundle methods for two-stage stochastic programming. *SIAM Journal on Optimization*, 21(2):517–544, 2011.

[143] R. Orsi. Numerical methods for solving inverse eigenvalue problems for nonnegative matrices. *SIAM Journal on Matrix Analysis and Applications*, 28:190–212, 2006.

[144] R. Orsi, U. Helmke, and J. Moore. A Newton-like method for solving rank constrained linear matrix inequalities. *Automatica*, 42:1875–1882, 2006.

[145] P. Pardalos and G.P. Rodgers. Computational aspects of a branch and bound algorithm for quadratic zero-one programming. *Computing*, 45:134–144, 1990.

[146] R.A. Poliquin, R.T. Rockafellar, and L. Thibault. Local differentiability of distance functions. *Transactions of the American Mathematical Society*, 352:5231–5249, 2000.

[147] S. Poljak, F. Rendl, and H. Wolkowicz. A recipe for semidefinite relaxation for (0,1)-quadratic programming. *Journal of Global Optimization*, 7:51–73, 1995.

[148] J.-B. Poly and G. Raby. Fonction distance et singularités. *Bull. Sci. Math*, 108(2):187–195, 1984.

[149] H. Qi and D. Sun. Correlation stress testing for value-at-risk: an unconstrained convex optimization approach. *Computational Optimization and Applications*, 45(2):427–462, 2010.

[150] L.Q. Qi and J. Sun. A nonsmooth version of Newton's method. *Mathematical Programming*, 58(3):353–367, 1993.

[151] F. Rendl, G. Rinaldi, and A. Wiegele. Solving max-cut to optimality by intersecting semidefinite and polyedral relaxations. *Math. Programming*, 121:307–335, 2010.

[152] J. Renegar. Linear programming, complexity theory and elementary functionnal analysis. *Mathematical Programming*, 70:279–351, 1995.

[153] J. Renegar. Condition numbers, the barrier method, and the conjugate gradient method. *SIAM Journal on Optimization*, 6:879–912, 1996.

[154] P. Richtarik and M. Takac. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.

[155] R Tyrrell Rockafellar and Roger J-B Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of operations research*, 16(1):119–147, 1991.

[156] R.T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[157] R.T. Rockafellar. Lagrange multipliers and optimality. *SIAM Review*, 35:183–238, 1993.

[158] R.T. Rockafellar and R.J.-B. Wets. *Variational Analysis*. Springer Verlag, Heidelberg, 1998.

[159] F. Roupin. From linear to semidefinite programming: an algorithm to obtain semidefinite relaxations for bivalent quadratic problems. *Journal of Combinatorial Optimization*, 8(4):469–493, 2004.

[160] A. Ruszczyński and A. Shapiro. *Stochastic Programming*, volume 10 of *Handbooks in Operations Research and Management Science*. Elsevier, Amsterdam, 2003.

[161] N. V. Sahinidis. *BARON 12.1.0: Global Optimization of Mixed-Integer Nonlinear Programs,* User's Manual, 2013.

[162] R. Saigal, L. Vandenberghe, and H. Wolkowicz. *Handbook of Semidefinite Programming*. Kluwer, 2000.

[163] A. Shapiro. Existence and differentiability of metric projections in Hilbert space. *SIAM Journal on Optimization*, 4:130–141, 1994.

[164] A. Shapiro. On the asymptotics of constrained local m-estimators. *Annals of statistics*, pages 948–960, 2000.

[165] A. Shapiro and F. Al-Khayyal. First-order conditions for isolated locally optimal solutions. *Journal of Optimization Theory and Applications*, 77:189–196, 1993.

[166] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming. Modeling and Theory*, volume 9 of *MPS-SIAM series on optimization*. SIAM and MPS, Philadelphia, 2009.

[167] H. Sherali and W. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3):411–430, 1990.

[168] N.Z. Shor. Class of global minimum bounds of polynomial functions. *Cybernetics*, 23(6):731–734, 1987.

[169] M.V. Solodov. On approximations with finite precision in bundle methods for nonsmooth optimization. *Journal of Optimization Theory and Applications*, 119(1):151–165, 2003.

[170] M. Tahanan, W. van Ackooij, A. Frangioni, and F. Lacalandra. Large-scale unit commitment under uncertainty. *4OR*, pages 1–57, 2015.

[171] J.A. Tropp, I.S. Dhillon, R.W. Heath, and T. Strohmer. Designing structured tight frames via in alternating projection method. *IEEE Transactions on Information Theory*, 51:188–209, 2005.

[172] W. van Ackooij, V. Berge, W. de Oliveira, and C. Sagastizábal. Probabilistic optimization via approximate p-efficient points and bundle methods. *Computers & Operations Research*, 77:177–193, 2017.

[173] W. van Ackooij and W. de Oliveira. Level bundle methods for constrained convex optimization with various oracles. *Computation Optimization and Applications*, 57(3):555–597, 2014.

[174] W. van Ackooij and A. Frangioni. Incremental bundle methods using upper models. Technical report, University of Pisa, 2016.

[175] W. van Ackooij, A. Frangioni, and W. de Oliveira. Inexact stabilized benders' decomposition approaches with application to chance-constrained problems with finite support. *Computational Optimization and Applications*, pages 1–33, 2016.

[176] W. van Ackooij and R. Henrion. Gradient formulae for nonlinear probabilistic constraints with Gaussian and Gaussian-like distributions. *SIAM Journal on Optimization*, 24(4):1864–1889, 2014.

[177] W. van Ackooij, R. Henrion, A. Moller, and R. Zorgati. Joint chance constrained programming for hydro reservoir management. *Optimization and Engineering*, 15(2):509–531, 2014.

[178] W. van Ackooij and C. Sagastizábal. Constrained bundle methods for upper inexact oracles with application to joint chance constrained energy problems. *SIAM Journal on Optimization*, 24(2):733–765, 2014.

[179] J. von Neumann. The geometry of orthogonal spaces, functional operators- vol. ii. *Annals of Math. Studies*, 22(1950), This is a reprint of mimeographed lecture notes, first distributed in 1933.

[180] C.A. Weber and J.P. Allebach. Reconstruction of frequency-offset Fourier data by alternating projection on constraint sets. In *Proceedings of the 24th Allerton Conference on Communication, Control and Computing*, pages 194–201. Urbana-Champaign, IL, 1986.

[181] M. Yamashita, K. Fujisawa, M. Fukuda, K. Kobayashi, K. Nakata, and M. Nakata. Latest developments in the SDPA family for solving large-scale SDPs. In Miguel F. Anjos and Jean B. Lasserre, editors, *Handbook on Semidefinite, Conic and Polynomial Optimization*, volume 166 of *International Series in Operations Research & Management Science*, pages 687–713. Springer US, 2012.

[182] K. Yang and R. Orsi. Generalized pole placement via static output feedback: a methodology based on projections. *Automatica*, 42:2143–2150, 2006.

[183] S. Yıldız and G. Cornuéjols. Cut-generating functions for integer variables, 2014.

[184] A. Zaffaroni. Convex radiant costarshaped sets and the least sublinear gauge. *Journal of Convex Analysis*, 20(2):307–328, 2013.

[185] G. Zakeri, A. Philpott, and D. Ryan. Inexact cuts in benders decomposition. *SIAM Journal on Optimization*, 10(3):643–657, 2000.

[186] S. Zaourar and J. Malick. Quadratic stabilization of benders decomposition. *Submitted*, 2014. preprint hal-01181273.

[187] Q. Zhao, S. Karisch, F. Rendl, and H. Wolkowicz. Semidefinite programming relaxations for the quadratic assignment problem. *Journal of Combinatorial Optimization*, 2(1):71–109, 1998.

[188] C. Zhu, R. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Softw.*, 23(4):550–560, December 1997.