

Efficient methods for structured nonconvex optimization and optimization under inexact oracles

Guillaume Van Dessel

Thesis submitted in partial fulfillment
of the requirements for the degree of
Docteur en sciences de l'ingénieur et technologie

Dissertation committee:

Pr. François Glineur (UCLouvain, advisor)
Pr. Yuri Nesterov (UCLouvain)
Pr. Pierre-Antoine Absil (UCLouvain)
Pr. Hoai-An Le Thi (Université de Lorraine)
Dr. Wellington de Oliveira (Mines Paris PSL)
Pr. Jean-Charles Delvenne (UCLouvain, chair)

Abstract

This thesis revolves around three main optimization frameworks, each accounting for one thesis part. In each of them, contributions are spread among theoretical results and fully implementable algorithms. Notably, a particular emphasis has been put throughout this work to either propose parameter-free methods or to specify how to fix the values of an algorithms' parameters.

The first part about *Tunable Oracle Methods* addresses the problem of optimally coping with inexact updates/information within traditional methods when inexactness is controllable and its (worst-case) impact predictable. Based on an assumed relationship linking the computational cost invested at a given iteration and the level of inexactness incurred, we derive optimal inexactness schedules for a given total computational budget.

In the second part about *Difference-of-Convex Methods*, we highlight various substructures of generic Difference-of-Convex (DC) objectives (e.g. Sum of Minimum-of-Convex) to which individual chapters are dedicated. (DC) objectives represent a broad class of nonconvex and generally nonsmooth functions that enhance and extend the modelling power of convex ones. Their optimization is thereby practically very attractive but unfortunately computationally really difficult. Within each of the aforementioned chapters, we endow the substructure with a specific minimization oracle. Computing these oracles always amount to solving some convex optimization problem. Then, we analyze what can be achieved with the oracle at hand and how it can be best exploited to harvest the best objective values in reasonable computational times.

In the last part about *Chance-Constrained Programs* (CCP), we present pre-solve techniques for optimization problems that exhibit a single probabilistic constraint. We focus on the finite-support case, i.e. the random

parameter involved in the formulation can take any value among several possible scenarios. Such (CCP) are usually reformulated as mixed-integer convex programs involving binary variables. Under the assumption of quasi-convexity of the probabilistic constraint with respect to the random parameter, we use some geometric arguments in the parameter-space leading to sufficient conditions to reduce the number of binary variables used. These conditions dramatically speed-up the solving of the aforementioned mixed-integer convex programs.

Acknowledgments

J'attendais le moment d'écrire ces quelques lignes depuis bien longtemps... Synonymes de complétude et d'accomplissement, ce sont elles qui ponctuent le parcours du doctorat. Le doctorat a été pour moi une découverte pleine. Je la qualifierais d'expérience de vie tant les apprentissages qui en découlent sont précieux et abondants. Lorsque vous suivez la voie de *Teaching Assistant* (TA), une certaine liberté de recherche vous est accordée. Cela comprend ses avantages et ses inconvénients. Cette liberté est une saine garante d'une recherche intrinsèquement portée par l'intérêt. Cependant, elle implique que le chercheur se pose lui-même les bonnes questions, prenne des initiatives et, surtout, apprenne à éviter de s'égarer *de trop* dans ses investigations. Ceci étant dit, à mon humble avis, les meilleures découvertes surviennent souvent là où ne les soupçonnait pas. Les travaux dont je suis le plus fier durant ma thèse sont nés d'une sortie de route, de résultats décevants obtenus d'un premier algorithme que j'avais imaginé. C'est en comprenant pourquoi nous pouvions nous attendre à de pareils résultats que nous avons pu rebondir plus haut et prendre réellement de la hauteur. Cela restera pour moi un fait marquant que je garderai en tête lorsque je serai confronté à nouveau à des difficultés imprévues...

J'aimerais désormais prendre le temps de remercier les principaux protagonistes m'ayant accompagné, de près ou de loin, dans la palpitante aventure du doctorat. J'annonce à l'avance, je risque d'en oublier certain(e)s et je m'en excuse.

Tout d'abord, je tiens à remercier mon promoteur, François Glineur, grâce à qui cette aventure fut sereine, passionnante et rondement menée. François, fort de ses dizaines d'années d'expérience, m'a aidé à naviguer

sans écueil au travers des remous du monde académique. Il m'en appris les lois, les coutumes et les limites. Doté d'une grande curiosité ainsi que d'une bonne dose d'intuition, il a pu m'écouter avec attention pendant presque 6 ans et sans cesse questionner mes choix afin de les rendre plus robustes aux critiques. Ayant été son assistant pour quelques de ses cours, je salue également son sens de la justice et de l'équité, allant bien au delà de ce qui se fait en général. C'est une qualité très inspirante.

Ensuite, naturellement, je pense à ma femme Sophie (et récemment mon fils Gaspard, malgré lui je crois) qui m'a soutenu tout du long. Le doctorat, uniquement de façon très ponctuelle, génère du stress et du travail de dernière minute. Ces pics (i.e. *deadlines*) ont constitué des moments où la compréhension du partenaire (heureusement pour moi, Sophie est chercheuse également) est essentielle. J'ai reçu son soutien inconditionnel.

Je remercie fort mes parents et par extension ma famille au sens large. Ils m'ont supporté de tout leur coeur et n'ont jamais remis en question que je travaille parfois (par passion) à la place de regarder la télévision avec eux. Ils m'auront fait aussi comprendre à quel point notre recherche leur paraît niche et déconnectée de leur réalité. D'une certaine manière, sans le vouloir, ils m'ont permis de rester terre-à-terre par moment. Inconsciemment, ce sont peut-être eux qui ont fait naître en moi le désir de rendre chacun de mes algorithmes les plus applicables possibles. Ils ont suscité en moi le besoin de produire une recherche qui puisse (ou tout du moins essayer de) s'ancrer dans leur réalité.

Finalement, comment ne pas remercier mes collègues ? L'ambiance de travail à l'Euler est sans pareil, je pense. Les blablas de cafet' me manqueront à coup sûr, légères et sans tabou, elles m'ont permis de me sentir intégré au sein de notre institut. Je n'oublierai pas les discussions de bureau avec Bastien et Timothée, qui m'ont rejoint ces deux dernières années. Au delà, j'aimerais aussi saluer mes copains/collègues plus anciens (Henri, Briec, Charles-Henry, Victoria, Antoine). Avec eux, j'ai partagé tant de repas de midi à parler *gossips* ICTEAM ou à explorer un problème de mathématiques ardu, au choix ! Merci pour ces bons moments.

Notations

Operators

$\lambda_{\max}(\mathbf{V})$ _____ Maximal (real) eigenvalue of $\mathbf{V}$ (p. 11)

$\lambda_{\min}(\mathbf{V})$ _____ Minimal (real) eigenvalue of $\mathbf{V}$ (p. 11)

$|\cdot|$ _____ Elementwise absolute value (p. 11)

$\|\cdot\|_p$ _____ p -norm of a vector/matrix (p. 11)

$\mathbb{I}_S$ _____ Indicator function of a set S (p. 17)

$\text{spec}(\mathbf{V})$ _____ Spectrum of matrix $\mathbf{V}$ (p. 11)

$\text{dom } h$ _____ Domain of an operator h (p. 12)

$F'(x; u^\circ)$ _____ Directional derivative of F at x in the direction u° (p. 17)

$\text{card}(S)$ _____ Cardinality of set S , i.e. the number of elements in S (p. 10)

$\text{id}_{\mathbb{E}}$ _____ Identity operator on $\mathbb{E}$ (p. 30)

$\text{Nor}(v, \Sigma)$ _____ Normal distribution with mean v and variance Σ (p. 54)

$\text{Uni}(S)$ _____ Uniform distribution over a set S (p. 55)

Comparison & Logic

$\geq, >$ _____ Generalized inequalities: greater or equal, strictly greater (p. 13)

$\wedge$ _____ Logical "AND" (p. 14)

Vectors & Matrices

$\mathbf{I}_d$ _____ Identity matrix of order d (p. 11)

$\text{vec}(v)$ _____ Matrix of zeroes with diagonal filled with v (p. 11)

$(v)_S$ _____ Subvector of v with values taken at positions $S \subseteq \{1, \dots, d\}$ (p. 10)

$\mathbf{0}_d$ _____ All-zeroes vector of size d (p. 9)

$\mathbf{1}_d$ _____ All-ones vector of size d (p. 9)

$\mathbf{e}_S$ _____ Vector of zeroes but ones only at positions in $S \subseteq \{1, \dots, d\}$ (p. 9)

$\mathbf{U} \odot \mathbf{V}$ _____ Elementwise product between $\mathbf{U}$ and $\mathbf{V}$ (p. 10)

$\mathbf{V}$ _____ Matrix represented by a bold capital letter (p. 10)

$\mathbf{V}^T$ _____ Transpose of matrix $\mathbf{V}$ (p. 10)

$\mathbf{V}_{:,k}$ _____ k -th row of matrix $\mathbf{V}$ (p. 10)

$\mathbf{V}_{s,:}$ _____ s -th row of matrix $\mathbf{V}$ (p. 10)

$\mathbf{V}_{s,k}$ _____ Element at s -th row and k -th column of matrix $\mathbf{V}$ (p. 10)

Sets

$[x, y]$ _____ Closed segment connecting point x to point y (p. 13)

Δ^n _____ n -standard simplex (p. 14)

$\emptyset$ _____ Empty set (p. 15)

$\mathbb{B}_{\|\cdot\|}(\hat{x}; R)$ _____ $\|\cdot\|$ -norm ball centered at $\hat{x}$ of radius R (p. 13)

$\mathbb{B}_p(\hat{x}; R)$ _____ $\|\cdot\|_p$ -norm ball centered at $\hat{x}$ of radius R (p. 13)

$\partial h(x)$ _____ Subdifferential set of convex function h at point x (p. 16)

$\partial_C h(x)$ _____ Clarke subdifferential set of h at point x (p. 16)

$\text{aff}(S)$ _____ Affine hull of a collection of elements/set S (p. 14)

$\text{cone}(S)$ _____ Canonical hull of a collection of elements/set S (p. 14)

$\text{conv}(S)$ _____ Convex hull of a collection of elements/set S (p. 14)

$\text{int}(S)$	Interior of a set S (p. 14)
$\text{rint}(S)$	Relative interior of a set S (p. 14)
$]x, y[$	Open segment connecting point x to point y (p. 13)
$[N] = \{1, \dots, N\}$	N first counting numbers (p. 9)
$\bar{\mathbb{R}}$	Extended real number line (p. 9)
$\mathbb{N}$	Natural numbers (p. 9)
$\mathbb{R}$	Real numbers (p. 9)
$\mathbb{R}_+$	Nonnegative real numbers (p. 9)
$\mathbb{R}_-$	Nonpositive real numbers (p. 9)
$\mathbb{R}_{++}$	Strictly positive real numbers (p. 9)
$\mathbb{R}_{--}$	Strictly negative real numbers (p. 9)
$\ker(\mathbf{V})$	Kernel of a matrix $\mathbf{V}$ (p. 88)
S^d	Cartesian product of d copies of a set S (p. 9)
$S_2 \setminus S_1$	Set S_2 but with elements of set S_1 excluded (p. 9)

Acronyms

CCP	Chance-Constrained Program (p. 197)
DC	Difference-of-Convex (p. 4)
DMsC	Difference-of-Max-of-smooth Convex (p. 73)
DSMsC	Difference-of-Sum-of-Max-of-smooth Convex (p. 73)
FGM	Fast Gradient Method (p. 33)
GM	Gradient Method (p. 24)
KKT	Karush-Kuhn-Tucker (p. 3)
MC	Minimum-of-Convex (p. 71)
MM	Majorization-Minimization (p. 74)
SMC	Sum-of-Minimum-of-Convex (p. 72)

Contents

	Abstract	i
	Acknowledgments	iii
	Notations	v
1	Introduction	1
	1.1 Objectives	5
	1.2 Content	5
2	Preliminary Material	9
	2.1 Generalities	9
	2.2 Geometry	13
	2.3 Optimality concepts	15
I	Tunable Oracle Methods	
3	Optimal Inexactness Schedules	23
	3.1 Inexactness in numerical optimization	23
	3.2 Tunable (Inexact) Oracles	28
	3.3 Optimization of inexactness schedules	35
	3.4 Numerical Experiments	53
	3.5 Conclusion	64
II	Difference-of-Convex (DC) Methods	
4	Guided Tour of (DC) Structure	67

4.1	Intuition	67
4.2	Diversity	69
4.3	Hierarchy	70
4.4	Oracles	74
5	Adaptive Sequential (DC) algorithm with extrapolation	77
5.1	Introduction	77
5.2	Regularized Difference-of-Bundles optimization	88
5.3	ASDCe	92
5.4	Numerical Experiments	98
5.5	Conclusion	104
6	Upper-Lower Optimization method for (MC)	107
6.1	Introduction	107
6.2	Minimum structured optimization	114
6.3	ULO	121
6.4	Numerical Experiments	136
6.5	Conclusion	144
7	Relaxed Alternating Minimization for (SMC)	147
7.1	Introduction	147
7.2	Equivalent representations of (SMC)	163
7.3	Relaxed Alternating Minimization ($\mathbf{x}$ -AM)	175
7.4	Numerical Experiments	183
7.5	Conclusion	192

III Chance-Constrained Programs (CCP)

8	Presolve techniques for finite-support (CCP)	195
8.1	Introduction	195
8.2	Enumerative complexity	199
8.3	Presolve techniques	203
8.4	Numerical Experiments	207
8.5	Conclusion	212
★	Discussion	213
	Bibliography	216

1

Introduction

OPTIMIZING decisions is a longstanding need and endeavour. Actually, many quantitative questions one is eager to tackle rationally translate into optimization problems. Presumably due to its importance from early on, the *Travelling Salesman Problem* represents one of such situations that can seamlessly be written as a mathematical optimization problem. Given a list of waypoints together with lengths of paths connecting (some of) these latter, the goal is to retrieve the shortest path (if any) to visit every waypoint at least once. Equally as iconic is the *Stigler Diet Problem*. There, one needs to determine the amounts of several nutritive products one should buy (and eat/drink) in order to meet the recommended dietary daily allowances while minimizing the costs of feeding. A third very well-known example resides in *Supervised Learning*, e.g. in breast cancer detection [NPD22]. Based on labeled mammographies (some of them tagged as *sane* and others presenting signs of *tumors*), one aims to tune the parameters of a deep-learning model. Ultimately, this model should predict the most accurately possible whether a provided mammography is *healthy* or not. In order to make their study and their solving more systematic, mathematicians gradually drew the nomenclature of optimization problems. Nowadays, it is widely accepted that they admit a formal definition (see e.g. [BV04]) involving three key ingredients listed hereafter.

Variables: the degrees of freedom at hand, i.e. quantities (either continuous or discrete) that encode the decisions to be taken.

Objective: a measure of (dis)satisfaction that can be computed by assigning values to the variables.

Constraints: conditions that should be fulfilled under the decisions that are made, restricting thereby the values variables can take.

For example, recalling the *Stigler Diet Problem* above, one identifies that the amounts of each food to buy represents the variables. Unless stated otherwise, one could assume that such amounts are continuous and positive quantities. Then, it is easy to observe that the total price one must spend for the diet corresponds to the objective. Obviously, it appears for the customer as a dissatisfaction to be minimized. Finally, the problem imposes some constraints in the sense that a nutritional equilibrium should be attained. Overall, the goods in their respective decided quantities should ensure that daily dietary recommendations are satisfied.

Over the decades, the scientific community adopted the following abstraction of an optimization problem

$$F^* = \min_{x \in \mathcal{X}} F(x) \quad (1.1)$$

where F stems as the **objective** function and $F(x)$ represents its value for the decision **variables** x . The requirement telling that x must be included in the feasible set $\mathcal{X}$, i.e. $x \in \mathcal{X}$, implements the **constraints** aforementioned. Finally, $F^* \in \mathbb{R}$ represents the *optimal (minimal) value* of the problem. Without loss of generality, it is very often possible to rephrase (1.1) in an equivalent form where F is continuously differentiable since sources of nonsmoothness can be rewritten as additional constraints with, eventually, extra variables. As a mathematical riddle, optimization problems can first be tackled analytically. Back to the 18th century, Joseph-Louis Lagrange came up with optimality conditions known today as the *method of Lagrange multipliers* [BR14]. In other words, he described necessary conditions depending on F and $\mathcal{X}$ that any x achieving optimality, i.e. $F^* = F(x)$, must satisfy. As a byproduct, a lot of potential decision variables are usually discarded because they do not meet the conditions. Lagrange's pioneer work focused on equality constraints, i.e. the feasible set turns out to be the roots that a couple of functions share in common. Later on, more refined conditions also taking into account inequality constraints were devised. These are re-

ferred to as KKT conditions in the literature (see e.g. [BMS13]), acronym of their founders. Indeed, they were independently stated by William Karush in his master's thesis (1939) and by Harold W. Kuhn & Albert W. Tucker in their celebrated paper (1951) [KT51]. In what concerns continuous optimization, i.e. problems whose variables are allowed to take continuous values, KKT conditions serve as cornerstone. They form systems of equations for which one can, in easy cases, find all the solutions analytically. Then, solving (1.1) boils down to evaluate the objective F at all these solutions and elect one that leads to a minimal value F^* . Nevertheless, closed-form solutions are far from being ubiquitous in optimization and very few problems can be treated that way. More often, one needs to resort to numerical methods that automatically computes (approximate) KKT points, i.e. variables x (approximately) fulfilling KKT conditions. Such points are valid candidates as optimal decisions for problem (1.1).

Reading between lines, since KKT equations are not meant to be sufficient conditions in general, KKT points might fail to be optimal solutions of (1.1). Fortunately enough, for some problem structures, every KKT point is optimal. For example, this is the case for what are called **convex** problems, i.e. F is convex on $\mathcal{X}$ convex set¹. In addition, KKT conditions are necessary for **convex** problems under additional constraint qualifications, e.g. Slater's condition. In such circumstances, it is clearly worth investing resources to develop efficient numerical methods that can yield (approximate) KKT points in a reasonable amount of time. In this spirit, a panacea of research works were undertaken to deal with convex problems. Some families of optimization algorithms emerged, i.e. zero-order (no use of derivatives, see e.g. [DJWW13]), first-order (use of the gradient of F , see e.g. [Bec17] for an exhaustive course), second-order and higher-order methods (use of, if they exist, Hessian matrix and/or higher-order derivative of F , see e.g. [JHZ21]), etc. Depending on the dimensionality of the problem, i.e. the number of variables, but also on the cost of their iterations, one picks up a method from a suitable family. There is some *trade-off* to be considered. Methods involving lower-order derivatives require more iterations but these iterations are considerably cheaper in terms of CPU time.

Until recently, research in numerical optimization largely revolved around

¹To be accurate: assuming a formulation where F and $c^{(j)}$ ($j = 1, \dots, m$) are convex and belong to $\mathcal{C}^1(\mathbb{R}^d)$ and $\mathcal{X}$ is described by inequalities $c^{(j)}(x) \leq 0$ for $j = 1, \dots, m$.

methods that assumed, under the hat, access to exact information about the objective and exact computations within their respective iterations. Authors [Kiw95] (respectively [SRB11]) initiated an analysis of classical non-smooth (respectively smooth) first-order methods alleviating this second assumption and thereby tolerating inexactness (1995) (respectively 2011). Since then, plenty of other works (see e.g. the doctoral dissertation [Dev13] or papers [DN20, STG⁺21, Xu21] and the references therein) were devoted to understand how flexible the standard optimization methods are with respect to their usual and original exact framework.

As tacitly teased above, a substantial amount of tools were developed for **convex** optimization. Yet, tons of problems of great practical interest can not be cast as convex, i.e. **nonconvex** problems, making these tools useless for direct use. Nevertheless, many of them exhibit a structure where convexity partially arises. For instance, some **objectives** can be expressed as the **difference of two convex** functions (DC). Aware of such structure, scholars managed to construct algorithms tackling (DC) based on ingredients that proved efficient in the convex world (see e.g. [DT97, LP18]).

Finally, we highlight that **constraints** $x \in \mathcal{X}$ are implicitly understood as *deterministic* in the sense that the geometry of $\mathcal{X}$ is fixed and known beforehand. In some circumstances, in addition to *deterministic* constraints, it also makes sense to introduce so-called *probabilistic* constraints, i.e.

$$\mathbb{P}_{\xi}[c(x, \xi) \leq 0] \geq 1 - \tau,$$

translating the wish of having x that belongs to $\mathcal{R}(\xi) := \{x \mid c(x, \xi) \leq 0\}$ for a proportion of at least $1 - \tau$ of the realizations of a random event ξ . With $\tau > 0$, unlike *deterministic* constraints, one potentially allows to consider decisions x such that $c(x, \xi) > 0$ for some outcomes $\xi \in \Xi$.

To conclude this introductory paragraph, we emphasize that difficult optimization problems, whether convex or nonconvex, are solved **iteratively**. That is, problem's data as well as other user predefined specificities are first fed in a black-box algorithm. Then, an initial guess on the optimal value of decision **variables** is taken accordingly. Lastly, the algorithm will update this guess a couple of times, each update comes at the end of some computations encapsulated in what is called an *iteration*, thereby the name. Eventually, an *exit condition* triggers and the current guess is returned as the (approximate) solution of the optimization problem.

1.1 Objectives

In the light of the above, one can argue that the *comfort zone* of numerical optimization resides in **convex** problems (i) under **exact** computations (ii) and **deterministic** constraints (iii). In this thesis, we aim at going beyond considerations (i), (ii), or (iii) by creating new methods. We put quite an emphasis on their implementability, i.e. our goal was to design algorithms that practitioners could easily try for themselves without a deep understanding of the underlying convergence theory. In that spirit, we tried to limit as much as possible the tuning of algorithms' hyperparameters.

1.2 Content

Chapter 2: Preliminary Material

What's in? In this chapter, we review standard notions of optimization. In addition, some notations are introduced. This knowledge base is mobilized throughout the whole thesis. Some notations are introduced.

Part I. Tunable Oracle (TO) Methods

Chapter 3: Optimal Inexactness Schedules

What's in? This chapter addresses consideration (ii) from the above section about thesis' objectives. It studies how to cope *optimally* with **inexactness** when this latter can be controlled in minimization algorithms.

This chapter has been published in [Optimization Methods and Software \[DG24b\]](#).

G. Van Dessel & F. Glineur, **Optimal inexactness schedules for tunable oracle-based methods**, In *Optimization Methods and Software* (39), pages 664-698, 2024.

Part 2. Difference-of-Convex (DC) Methods

Chapter 4: Guided Tour of (DC) Structure

What's in? This chapter introduces the realm of *difference-of-convex* (DC) optimization problems, core of the second part of this thesis wherein **convexity** (i) is dropped. In (DC) problems, the objective F from (1.1) is decomposed as $F = f_1 - f_2 + \Psi$ with f_1 , f_2 and Ψ convex. We present in this chapter subclasses of (DC) problems with more structure, i.e. assumptions on f_1 , f_2 and/or Ψ . In the remaining chapters of Part 2, we aim at taking advantage of these additional assumptions.

Chapter 5: Adaptive Sequential (DC) algorithm with extrapolation

What's in? We study proximal type algorithms tackling (DC) problems wherein f_1 has Lipschitz continuous gradient over the set $\mathcal{X}$. We combine adaptivity to unknown parameters, extrapolation and bundles all at once in a single new algorithm called ASDCe. We also discuss how and when using these features turns out to be efficient.

Two previous versions of this chapter were presented respectively at SigOPT23 (international conference) and the 42nd [Benelux Meeting](#).

Chapter 6: Upper-Lower Optimization method for (MC)

What's in? We propose a global optimization algorithm, UL0, to deal with the case where F can be expressed as the pointwise minimum of a finite collection of functions, called *pieces*. Specifically, we assume that $\mathcal{X}$ is defined by a large number m of constraints but the cardinality n of the collection is reasonable in the following sense: optimizing each *piece* individually subject to a subset of the m constraints is computationally tractable.

This chapter has been presented at NOPTA24 as poster and is currently going through the publication process of the [Journal of Global Optimization](#). A [preprint \[DG24a\]](#) is available on ArXiv.

Chapter 7: Relaxed Alternating Minimization method for (SMC)

What's in? In this chapter, we assume that F can be represented as a sum involving a big number N of terms, each of which being the minimum of

a finite collection of convex functions. Objective F from Chapter 7 can be recast as an objective from Chapter 6 whose number of *pieces* (i.e. n) grows exponentially with N . Thereby, Algorithm ULO is not applicable anymore since minimizing an exponential number of *pieces*, even when there is no additional constraint (i.e. $m = 0$), is too expensive. Global optimization being out of reach, we resort to local methods. First, we reformulate the problem of minimizing F as an equivalent mixed-integer convex program that can possibly be solved on (small) subregions of $\mathcal{X}$, providing local solutions. Second, we restrict ourselves to use continuous variables only. Accordingly, we reformulate the problem as a bi-convex one and apply new (approximate) alternating minimization methods, r-AM.

This chapter has been presented at EUROPT24 and is currently going through the publication process of the [Optimization](#) journal.

A [preprint \[DG25a\]](#) is available on ArXiv.

Part 3. Chance-Constrained Programs (CCP)

Chapter 8: Presolve techniques for finite-support (CCP)

What's in? Finally, this last chapter is dedicated to problems for which **deterministic** constraints (iii) are replaced by *probabilistic* counterparts.

We study a specific subclass of the generic Chance-Constrained Programs (CCP) framework wherein uncertainty is modeled by a random vector ξ of dimension p whose support is finite. After having reformulated (CCP) as a mixed-integer convex program (MI-CCP), we propose presolve techniques that aim at reducing the number of binary variables and the overall size of the search space in (MI-CCP). When $p \in \{2, 3\}$, i.e. the dimensionality of the uncertainty vector is low, our most efficient techniques rely on the assumption that the probabilistic constraint is quasi-convex in ξ .

This chapter is accepted for publication in [Lecture Notes in Networks and Systems](#) (proceedings of MCO25) and will be presented at EUROPT25.

A [preprint \[DG25b\]](#) is available on ArXiv.

2

Preliminary Material

THERE will be a substantial overlap regarding the notations employed throughout the three part of this thesis. We take the space of this preliminary chapter to clarify things from the start. We lay down central definitions as well as a couple of properties inherent to mathematical optimization. We will build upon the concepts introduced hereafter.

2.1 Generalities

Sets $\mathbb{N} = \{1, \dots, \infty\}$ will refer to the set of strictly positive integers. $\mathbb{R}_+$ (respectively $\mathbb{R}_-$) contains all the nonnegative (respectively nonpositive) real numbers. We denote $\mathbb{R}_{++} := \mathbb{R}_+ \setminus \{0\}$ and $\mathbb{R}_{--} := \mathbb{R}_- \setminus \{0\}$. Moreover, the extended real number line will be written as $\bar{\mathbb{R}} := [-\infty, \infty]$. Let $N \in \mathbb{N}$, we define $[N] := \{1, \dots, N\}$. Let S depict a set and $d \in \mathbb{N}$, S^d denotes the cartesian product involving d copies of S .

Elements $\mathbf{1}_d \in \mathbb{R}^d$ stands as the vector of size d with all elements equal to 1. For $S \subseteq [d]$, $\mathbf{e}_S$ represents the vector admitting, at position $k \in [d]$, the value 1 if $k \in S$ and 0 otherwise. Thus, $\mathbf{e}_{[d]} = \mathbf{1}_d$. Moreover, $\mathbf{e}_d$ will be understood as $\mathbf{e}_{\{d\}}$. On the other hand, $\mathbf{0}_d \in \mathbb{R}^d$ stands as the vector of size d with all elements equal to 0, i.e. $\mathbf{0}_d = \mathbf{e}_\emptyset$.

2 | Preliminary Material

For $v \in \mathbb{R}^d$, we define the subvector $(v)_S$ of size $\text{card}(S) \leq d$ whose values are taken from v at indices in S . Again, the shorthand notation $(v)_k := (v)_{\{k\}}$ will provide access to the k -th entry of object v .

Vectors & Matrices Let S be a set and $(N, d) \in \mathbb{N}^2$. Unless stated otherwise, an element from S^d is called a vector and decomposes as follows

$$v = (v_1, \dots, v_d)$$

where for each $k \in [d]$, $v_k \in S$. A matrix $\mathbf{V} \in S^{N \times d}$ can be seen as

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_{1,:} \\ \vdots \\ \mathbf{V}_{N,:} \end{bmatrix} = \begin{bmatrix} \mathbf{V}_{1,1} & \dots & \mathbf{V}_{1,d} \\ \vdots & \ddots & \vdots \\ \mathbf{V}_{N,1} & \dots & \mathbf{V}_{N,d} \end{bmatrix} = [\mathbf{V}_{:,1} \quad \dots \quad \mathbf{V}_{:,d}]$$

where for each $s \in [N]$, $\mathbf{V}_{s,:} \in S^d$ and for each $k \in [d]$, $\mathbf{V}_{:,k} \in S^N$ and, by transitivity, $\mathbf{V}_{s,k} \in S$. A vector $v \in S^N$ is also understood as a matrix $\mathbf{V} \in S^{N \times 1}$ with $\mathbf{V}_{:,1} = v$. The transpose of matrix $\mathbf{V}$ reads $\mathbf{V}^T \in S^{d \times N}$ with

$$\mathbf{V}^T = \begin{bmatrix} \mathbf{V}_{1,1} & \dots & \mathbf{V}_{N,1} \\ \vdots & \ddots & \vdots \\ \mathbf{V}_{1,d} & \dots & \mathbf{V}_{N,d} \end{bmatrix}.$$

Assuming that S is a subset of a real vector space $\mathbb{E}$ wherein operations $+$ and $\cdot$ are well-defined, for $\mathbf{U} \in S^{N \times B}$ and $\mathbf{V} \in S^{N \times d}$, the matrix product $\mathbf{W} \in S^{B \times d}$ reads

$$\mathbf{W} = \mathbf{U}^T \mathbf{V} = \begin{bmatrix} \sum_{s=1}^N \mathbf{U}_{s,1} \cdot \mathbf{V}_{s,1} & \dots & \sum_{s=1}^N \mathbf{U}_{s,1} \cdot \mathbf{V}_{s,d} \\ \vdots & \ddots & \vdots \\ \sum_{s=1}^N \mathbf{U}_{s,B} \cdot \mathbf{V}_{s,1} & \dots & \sum_{s=1}^N \mathbf{U}_{s,B} \cdot \mathbf{V}_{s,d} \end{bmatrix}.$$

Furthermore, if $B = d$, the elementwise product $\mathbf{X} \in S^{N \times d}$ reads

$$\mathbf{X} = \mathbf{U} \odot \mathbf{V} = \mathbf{V} \odot \mathbf{U} = \begin{bmatrix} \mathbf{U}_{1,1} \cdot \mathbf{V}_{1,1} & \dots & \mathbf{U}_{1,d} \cdot \mathbf{V}_{1,d} \\ \vdots & \ddots & \vdots \\ \mathbf{U}_{N,1} \cdot \mathbf{V}_{N,1} & \dots & \mathbf{U}_{N,d} \cdot \mathbf{V}_{N,d} \end{bmatrix}.$$

The spectrum of $\mathbf{V} = \mathbf{V}^T \in \mathbb{R}^{d \times d}$, written $\mathbf{spec}(\mathbf{V})$, collects the eigenvalues of $\mathbf{V}$

$$\mathbf{spec}(\mathbf{V}) := \left\{ \lambda \in \mathbb{R} \mid \exists v \in \mathbb{R}^d \setminus \{\mathbf{0}_d\}, \mathbf{V}v = \lambda \cdot v \right\}. \quad (2.1)$$

Eigenvalues rank from the smallest, $\lambda_{\min}(\mathbf{V})$, to the biggest, $\lambda_{\max}(\mathbf{V})$. The vectors $v \neq \mathbf{0}_d$ achieving $\mathbf{V}v = \lambda \cdot v$ for $\lambda \in \mathbf{vec}(\mathbf{V})$ are called eigenvectors.

More generally, applying any operator $o : S \rightarrow S$ in an elementwise fashion on a vector $v \in S^d$ or a matrix $\mathbf{V} \in S^{N \times d}$ is authorized, i.e. $o(v) = (o(v_1), \dots, o(v_d))$ or $\mathbf{Z} = o(\mathbf{V})$ with $\mathbf{Z}_{s,k} = o(\mathbf{V}_{s,k})$ for any $(s, k) \in [N] \times [d]$.

The notation $\mathbf{vec}(v) \in S^{d \times d}$ constructs a matrix full of zeroes but with a main diagonal filled with elements of $v \in S^d$ (0 should belong to S).

The identity matrix of order d , written $\mathbf{I}_d$, corresponds to the matrix $\mathbf{vec}(\mathbf{1}_d)$.

Sequences Let S be a set and $d \in \mathbb{N}$. A sequence of vectors u from S^d will be denoted as $\{u_k\}_{k \in [N]}$. Tacitly, it assigns to any integer $k \in \mathbb{N}$, a vector $u_k \in S^d$. Given an unbounded index set $\mathcal{K} \subseteq \mathbb{N}$, i.e. $\mathbf{card}(\mathcal{K}) = \infty$, $\{u_k\}_{k \in \mathcal{K}}$ is called a subsequence of $\{u_k\}_{k \in \mathbb{N}}$.

p -norms For any $p \geq 1$ and any $v \in \mathbb{R}^N$, the p -norm of v reads

$$\|v\|_p = \left(\sum_{s=1}^N |v_s|^p \right)^{\frac{1}{p}}.$$

We extend this notation with $p = \infty$ and set $\|v\|_\infty \triangleq \max\{|v_s| \mid s \in [N]\}$. In what concerns matrices, the p -norm of any $\mathbf{V} \in \mathbb{R}^{N \times d}$ translates to

$$\|\mathbf{V}\|_p = \max_{u \neq \mathbf{0}_p} \frac{\|\mathbf{V}u\|_p}{\|u\|_p}.$$

Frobenius norm In the spirit of the Euclidean norm (2-norm regarding p -norms above), the Frobenius norm of a matrix $\mathbf{V} \in S^{N \times d}$ amounts to

$$\|\mathbf{V}\|_F := \sqrt{\left(\sum_{s=1}^N \sum_{k=1}^d \mathbf{v}_{s,k}^2 \right)}.$$

2 | Preliminary Material

Dual spaces Let $d \in \mathbb{N}$ and $\mathbb{E}$ denote a d -dimensional real vector space endowed with a norm $\|\cdot\|_{\mathbb{E}}$. $\mathbb{E}$ is bound with its dual space, noted $\mathbb{E}^*$, via the bilinear *dual pairing* operator $\langle \cdot, \cdot \rangle : \mathbb{E}^* \times \mathbb{E} \rightarrow \mathbb{R}$. The induced (dual) norm of any element $g \in \mathbb{E}^*$ reads

$$\|g\|_{\mathbb{E}^*} := \max_{\|x\|_{\mathbb{E}} \leq 1} \langle g, x \rangle \quad (2.2)$$

Remark 2.1. Let $p \in \mathbb{N}$ and $(\mathbb{E}, \mathbb{E}^*) = (\mathbb{R}^d, \mathbb{R}^d)$ with $\langle g, x \rangle = \sum_{k=1}^d g_k \cdot x_k$ with $\|\cdot\|_{\mathbb{E}} = \|\cdot\|_p$. It comes $\|\cdot\|_{\mathbb{E}^*} = \|\cdot\|_{p^*}$ with p^* defined as

$$p^* = \begin{cases} \infty & p = 1 \\ p \cdot (p-1)^{-1} & p \in (1, \infty) \\ 1 & p = \infty. \end{cases} \quad (2.3)$$

Domain Let $(\mathbb{E}^{(1)}, \|\cdot\|^{(1)})$ and $(\mathbb{E}^{(2)}, \|\cdot\|^{(2)})$ be two normed vector spaces as defined above and let $h : S \subseteq \mathbb{E}^{(1)} \rightarrow \mathbb{E}^{(2)} \cup \{-\infty, \infty\}$. The (natural) *domain* of h , noted $\text{dom } h$, is defined as

$$\text{dom } h := \{x \in S \mid \|h(x)\|^{(2)} < \infty\}. \quad (2.4)$$

Lipschitz continuity Again, let $(\mathbb{E}^{(1)}, \|\cdot\|^{(1)})$ and $(\mathbb{E}^{(2)}, \|\cdot\|^{(2)})$ be two normed vector spaces and $L \in \mathbb{R}_+$. The function $h : \mathcal{X} \subseteq \mathbb{E}^{(1)} \rightarrow \mathbb{E}^{(2)}$ is said L Lipschitz continuous on $\mathcal{X}$ if for any $y, x \in \mathcal{X} \supseteq \text{dom } h$,

$$\|h(y) - h(x)\|^{(2)} \leq L \|y - x\|^{(1)}. \quad (2.5)$$

Also, h is called locally Lipschitz (continuous) if for every $x \in \mathcal{X}$ there exists a neighbourhood $\mathcal{S}$ around x such that h is $L_{\mathcal{S}}$ Lipschitz on $\mathcal{S}$.

Inequalities Let $\tilde{S} \subseteq \mathbb{E}$. Let $h_1 : \tilde{S} \rightarrow \mathbb{R} \cup \{\infty\}$, $h_2 : \tilde{S} \rightarrow \mathbb{R} \cup \{\infty\}$ be two functions with $\text{dom } h_1 = S = \text{dom } h_2$, we consider:

$$\begin{aligned} h_1 \geq h_2 &\Leftrightarrow h_1(x) \geq h_2(x), \quad \forall x \in \tilde{S} \\ h_1 > h_2 &\Leftrightarrow h_1(x) > h_2(x), \quad \forall x \in S. \end{aligned}$$

Let $N \in \mathbb{N}$. For any $u, v \in \mathbb{R}^N$, we also note:

$$\begin{aligned} u \geq v &\Leftrightarrow u_k \geq v_k, \quad \forall k \in [N] \\ u > v &\Leftrightarrow u_k > v_k, \quad \forall k \in [N]. \end{aligned}$$

Matrix inequalities work as follows. Let us consider that $\mathbb{E} = \mathbb{R}^d$ with $d \in \mathbb{N}$ and $\mathbf{V} \in (\mathbb{E}^*)^d$ is a matrix with d vectors of $\mathbb{E}^*$ on top of each other.

$$x \in \mathbb{R}^d \rightarrow \mathbf{V}x = (\langle \mathbf{V}_{1,:}, x \rangle, \dots, \langle \mathbf{V}_{d,:}, x \rangle) \in \mathbb{R}^d$$

The dominance of $\mathbf{U} \in (\mathbb{E}^*)^d$ with respect to $\mathbf{V}$ entails one of the following

$$\begin{aligned} \mathbf{U} \geq \mathbf{V} &\Leftrightarrow \langle \mathbf{U}x, x \rangle \geq \langle \mathbf{V}x, x \rangle \quad \forall x \in \mathbb{E} \\ \mathbf{U} > \mathbf{V} &\Leftrightarrow \langle \mathbf{U}x, x \rangle > \langle \mathbf{V}x, x \rangle \quad \forall x \in \mathbb{E}. \end{aligned}$$

Remark 2.2. Unless stated otherwise, $\mathbb{E} = \mathbb{R}^d$ for a certain $d \in \mathbb{N}$ clear from the context. The associated norms are Euclidean (set-up $p = 2$ within Remark 2.1), i.e. $\|v\| = \|v\|_2$, $\|\mathbf{V}\| = \|\mathbf{V}\|_2$. The distance between vectors u and v reads $\|u - v\|$ or between matrices $\mathbf{U}$ and $\mathbf{V}$ equals $\|\mathbf{U} - \mathbf{V}\|$.

2.2 Geometry

Let $\mathbb{E}$ be a real vector space as depicted on the previous page.

Definition 2.3 (Convex set). A set $\mathcal{X} \subseteq \mathbb{E}$ is *convex* if for any $x, y \in \mathcal{X}$,

$$[x, y] := \zeta y + (1 - \zeta)x \in \mathcal{X} \quad \forall \zeta \in [0, 1]. \quad (2.6)$$

Any segment connecting two points of $\mathcal{X}$ must entirely belong to $\mathcal{X}$.

Useful to define *local* concepts later, we recall the definition of norm balls.

Definition 2.4 (Norm ball). Let $\|\cdot\|$ be a norm on $\mathbb{E}$, $\hat{x} \in \mathbb{E}$ and $R \geq 0$. The *norm ball* of radius R centered at $\hat{x}$ is defined as

$$\mathbb{B}_{\|\cdot\|}(\hat{x}; R) := \{x \in \mathbb{E} \mid \|x - \hat{x}\| \leq R\}. \quad (2.7)$$

Moreover, we use the shorthand $\mathbb{B}_p(\hat{x}; R)$ when referring to $\mathbb{B}_{\|\cdot\|_p}(\hat{x}; R)$.

2 | Preliminary Material

Another geometrical set of utmost importance is the standard simplex.

Definition 2.5 (Standard simplex). Let $n \in \mathbb{N}$. The n -standard simplex reads

$$\Delta^n := \{q \in \mathbb{R}^n \mid \langle \mathbf{1}_n, q \rangle = 1, q \geq \mathbf{0}_n\} \quad (2.8)$$

The *convex hull* of a set $S \subseteq \mathbb{R}^d$ written $\mathbf{conv}(S)$ represents the smallest convex set containing S . This set is described as

$$\mathbf{conv}(S) := \left\{ \sum_{s=1}^N q_s \cdot g^{(s)} \mid N \in \mathbb{N}, g^{(s)} \in S \forall s \in [N] \wedge q \in \Delta^N \right\}. \quad (2.9)$$

Moreover, the *conical hull* generated by elements of G reads

$$\mathbf{cone}(S) := \left\{ \sum_{s=1}^N q_s \cdot g^{(s)} \mid N \in \mathbb{N}, g^{(s)} \in S \forall s \in [N] \wedge q \in \mathbb{R}_+^N \right\}. \quad (2.10)$$

Likewise, we present for completeness the *affine hull* spanned by G

$$\mathbf{aff}(S) := \left\{ \sum_{s=1}^N q_s \cdot g^{(s)} \mid N \in \mathbb{N}, g^{(s)} \in S \forall s \in [N] \wedge q \in \mathbb{R}^N, \langle \mathbf{1}_N, q \rangle = 1 \right\}.$$

Finally, we recall two different notions of *interior*. The second, namely the *relative interior*, is useful when dealing with *relatively smooth* functions.

Definition 2.6 (Interior). Let $S \subseteq \mathbb{E}$ and let $\|\cdot\|$ define the norm. Then, the interior of S is

$$\mathbf{int}(S) := \left\{ x \in S \mid \exists \alpha > 0, \mathbb{B}_{\|\cdot\|}(x; \alpha) \subseteq S \right\}. \quad (2.11)$$

Definition 2.7 (Relative interior). Let $S \subseteq \mathbb{E}$ and let $\|\cdot\|$ define the norm. Then, the relative interior of S is

$$\mathbf{rint}(S) := \left\{ x \in S \mid \exists \alpha > 0, \mathbb{B}_{\|\cdot\|}(x; \alpha) \cap \mathbf{aff}(S) \subseteq S \right\}. \quad (2.12)$$

In the case where $S = \mathcal{X}$ is *convex*, it comes ([Roc70], p. 47)

$$\mathbf{rint}(\mathcal{X}) = \left\{ x \in \mathcal{X} \mid \forall y \neq x \in \mathcal{X}, \exists z \in \mathcal{X}, x \in]y, z[\right\}. \quad (2.13)$$

Example 2.8. A typical representative example requiring the care of *relative interior*'s refinement compared to the usual *interior* works as follows.

Let $x, x_+ \in \mathbb{R}^2$ be two points of the plane and $S = [x, x_+]$, as in (2.6).

According to (2.11) and (2.12), the set $\text{int}(S) = \emptyset$ whereas $\text{rint}(S) =]x, x_+[$.

2.3 Optimality concepts

We recall here the abstract formulation of an optimization problem (1.1). Provided a finite d -dimensional real vector space $\mathbb{E}$ whose zero vector is $\mathbf{0}$, we consider a feasible set $\mathcal{X} \subseteq \mathbb{E}$ and an objective function $F : \mathbb{E} \rightarrow \bar{\mathbb{R}}$.

$$F^* = \min_{x \in \mathcal{X}} F(x) > -\infty. \quad (2.14)$$

Definition 2.9 (Local/Global optimality). A point $\hat{x} \in \mathcal{X}$ is called a *local minimum* of (2.14) if $\hat{x} \in \text{dom } F$ and there exists a radius $\alpha > 0$ such that

$$F(\hat{x}) = \min_{x \in \mathcal{X} \cap \mathbb{B}_2(\hat{x}; \alpha)} F(x) \quad (2.15)$$

whereas $\hat{x}$ is called a *global minimum* of (2.14) if and only if $F(\hat{x}) = F^*$.

Definition 2.10 (Convex function). Let $\mathcal{X} \subseteq \mathbb{E}$ and $h : \mathcal{X} \rightarrow \mathbb{R} \cup \{\infty\}$. The function h is called *convex* on $\mathcal{X}$, if $\mathcal{X}$ is convex and for every $x, y \in \mathcal{X}$,

$$h(\zeta \cdot y + (1 - \zeta) \cdot x) \leq \zeta \cdot h(y) + (1 - \zeta) \cdot h(x) \quad \forall \zeta \in [0, 1]. \quad (2.16)$$

Let $\mu \geq 0$. h is called μ -strongly convex on $\mathcal{X}$ if $h - \frac{1}{2}\|\cdot\|^2$ is convex on $\mathcal{X}$. Conversely, h is μ -strongly concave if and only if $-h$ is μ -strongly convex.

Remark 2.11. Note that a mere convex function will be 0-strongly convex.

Proposition 2.12 (Local to Global, [Ber09] Proposition 3.1.1). Let F be convex on a convex subset $\mathcal{X} \subseteq \mathbb{E}$. Every local optimum for (2.14) is global.

As we shall see, outside the convex realm, developing efficient algorithms provably returning *global* and even *local minima* is often out of reach. As a sequel, a whole bunch of relaxed notions of optimality, more accessible, where invented (see e.g. [LSM20a]). Regarding the present thesis, we will focus in particular on notions that mobilize *Clarke subdifferential sets*, defined immediately hereafter. The rest of this subsection is inspired from the optimality concepts evoked in papers [LSM20a, AAV20], thesis [Dev13].

2 | Preliminary Material

Definition 2.13 (Clarke subdifferential set). Let $h : \mathbb{E} \rightarrow \mathbb{R}$ be locally Lipschitz continuous (differentiable almost everywhere [LSM20a]), its *Clarke subdifferential set* at any $x \in \mathbb{E}$ corresponds to

$$\partial_C h(x) = \mathbf{conv}(D_h(x)) \quad (2.17)$$

with

$$D_h(x) := \left\{ g \in \mathbb{E}^* : \begin{array}{l} \exists \{x_k\}_{k \in \mathbb{N}}, \lim_{k \rightarrow \infty} x_k = x \text{ such that} \\ \nabla h(x_k) \text{ exists } \forall k \in \mathbb{N} \wedge \lim_{k \rightarrow \infty} \nabla h(x_k) = g \end{array} \right\}.$$

Remark 2.14. If $\partial_C h(x)$ is a singleton then h is differentiable at x .

Example 2.15 (Subdifferential set of a nondifferentiable concave function). Let $h : \mathbb{R} \rightarrow \mathbb{R}$ be defined by the mapping

$$x \mapsto h(x) = -(x^2 + |x|).$$

Away from $x = 0$, it is easy to observe that h is differentiable. It comes $\partial_C h(x) = \{-2x + 1\}$ for $x < 0$ and $\partial_C h(x) = \{-2x - 1\}$ for $x > 0$. Precisely at $x = 0$, either one produces a convergent sequence approaching $x = 0$ by the left so that $g = -(0 - 1) = 1$ in (2.17) or by the right so that $g = -(0 + 1) = -1$. In the end, one has $\partial_C h(0) = \mathbf{conv}(\{-1, 1\}) = [-1, 1]$.

Proposition 2.16 ([Roc70] Theorem 17.2). Let h be convex in Definition 2.13. The set $\partial_C h(x)$ reduces to the non-empty and compact set $\partial h(x)$,

$$\partial h(x) := \left\{ g \in \mathbb{E}^* \mid h(y) \geq h(x) + \langle g, y - x \rangle \quad \forall y \in \mathbb{E} \right\}. \quad (2.18)$$

Definition 2.17 (Quadratic lower/upper bounds). Let h be μ -strongly convex in Definition 2.13, it holds at any $x \in \mathbb{E}$,

$$h(y) \geq h(x) + \langle g, y - x \rangle + \frac{\mu}{2} \|y - x\|^2 \quad \forall y \in \mathbb{E}, \forall g \in \partial h(x). \quad (\text{QLB})$$

If h in Definition 2.13 is differentiable then for every $x \in \mathbb{E}$, $\partial_C h(x) = \{\nabla h(x)\}$ and if ∇h is L Lipschitz continuous then it also comes

$$h(y) \leq h(x) + \langle \nabla h(x), y - x \rangle + \frac{L}{2} \|y - x\|^2 \quad \forall y \in \mathbb{E}. \quad (\text{QUB})$$

Remark 2.18. We denote by L -smooth function exclusively functions that exhibit a L Lipschitz continuous gradient hence satisfying (QUB).

In order to clarify the interest for (Clarke) subdifferential sets as tractable tools, we need first to rewrite (2.14) into an equivalent form using *indicator functions* and, second, to introduce the concept of *regular function*.

Definition 2.19 (Indicator function). The *indicator function* of a set $\mathcal{X} \subseteq \mathbb{E}$, noted $\mathbb{I}_{\mathcal{X}} : \mathbb{E} \rightarrow \mathbb{R} \cup \{\infty\}$, is defined by

$$x \rightarrow \mathbb{I}_{\mathcal{X}}(x) := \begin{cases} 0 & x \in \mathcal{X} \\ \infty & x \notin \mathcal{X}. \end{cases} \quad (2.19)$$

Remark 2.20. If $\mathcal{X}$ above represents a convex set, $\mathbb{I}_{\mathcal{X}}$ becomes a convex function on $\mathcal{X}$ ($S = \mathbb{E}$ regarding the formal Definition 2.10). In that case, despite not being locally Lipschitz continuous, the indicator function admits a stronger subdifferential set (as in Proposition 2.16). Let $x \in \mathcal{X}$,

$$\partial \mathbb{I}_{\mathcal{X}}(x) = \mathcal{N}_{\mathcal{X}}(x) := \{g \in \mathbb{E}^* \mid \langle g, y - x \rangle \leq 0 \quad \forall y \in \mathcal{X}\},$$

known as the *normal cone* to $\mathcal{X}$ at x .

At the light of the above, without loss of generality, one can switch from the constrained optimization problem (2.14) to its unconstrained equivalent (2.20) involving the extended objective function $\mathcal{F} := \mathbb{I}_{\mathcal{X}} + F$,

$$F^* = \min_{x \in \mathbb{E}} \mathcal{F}(x). \quad (2.20)$$

Prior to unveiling the definition of *regular function*, *directional derivatives* of a function must be recalled. We point out that *directional derivatives* might fail to exist in some cases, even for finite-valued functions.

Definition 2.21 (Directional derivative). Let $F : \mathbb{E} \rightarrow \mathbb{R}$ denote a function, $x \in \mathbb{E}$ a point and $v \in \mathbb{E} \setminus \{0\}$ a direction. The *directional derivative* of F at x following u° , noted $F'(x; u^\circ)$, is the following limiting value (if it exists):

$$F'(x; u^\circ) := \lim_{\zeta \searrow 0} \frac{F(x + \zeta \cdot u^\circ) - F(x)}{\zeta}. \quad (2.21)$$

Proposition 2.22 (Strict decrease, following from [AAV20] Proposition 3.1). If $F'(x; u^\circ) \leq -2\epsilon < 0$ then there exists $\xi > 0$ such that

$$F(x + \zeta \cdot u^\circ) \leq F(x) - \zeta \epsilon \quad \forall \zeta \in [0, \xi]. \quad (2.22)$$

2 | Preliminary Material

Definition 2.23 (Regular, inspired from [LSM20a]: Definition 2 and Fact 3). A function $F : \mathbb{E} \rightarrow \mathbb{R}$ is called *regular* at x if for every $u^\circ \in \mathbb{E}$,

$$F'(x; u^\circ) = F^\circ(x; u^\circ) := \max_{g \in \partial_C F(x)} \langle g, u^\circ \rangle. \quad (2.23)$$

A function *regular* at every point of a set $S \subseteq \mathbb{E}$ is *regular* on S .

Remark 2.24 ([LSM20a]). Every finite-valued convex function h is regular.

Remark 2.25. If F is differentiable at x then $F'(x; u^\circ) = \langle \nabla F(x), u^\circ \rangle$.

Proposition 2.26 (Direct addition rule, building upon [RW04] Corollary 10.9).

Let $F = f_1 - f_2$ with $f_1 : \mathbb{E} \rightarrow \mathbb{R}$ and $f_2 : \mathbb{E} \rightarrow \mathbb{R}$, both of which are regular. For every $x \in \mathbb{E}$,

$$\partial_C F(x) \subseteq \partial_C f_1(x) + \partial_C(-f_2)(x) = \partial_C f_1(x) - \partial_C f_2(x). \quad (2.24)$$

We are now in position to state a central proposition for this thesis that justifies the algorithmic search for *critical points*, also defined hereafter.

Proposition 2.27 (Criticality condition, according to [Bal09] Theorem 3.1).

Let $\mathcal{X} \subseteq \mathbb{E}$ be a convex set and let $\hat{x} \in \mathcal{X}$. If the point $\hat{x}$ is a local minimum for problem (2.14) with F locally Lipschitz continuous then the following inclusion must hold

$$\mathbf{0} \in \mathcal{N}_{\mathcal{X}}(\hat{x}) + \partial_C F(\hat{x}). \quad (\text{Clarke-criticality})$$

Moreover, if $F = f_1 - f_2$ fulfills the requirements of Proposition 2.26 with f_1 and f_2 being both convex, the above necessary (local) optimality condition implies

$$\mathbf{0} \in \mathcal{N}_{\mathcal{X}}(\hat{x}) + \partial f_1(\hat{x}) - \partial f_2(\hat{x}). \quad (\text{DC-criticality})$$

Remark 2.28. A point $\hat{x} \in \mathcal{X}$ satisfying either (Clarke-criticality) or simply (DC-criticality) will be called a *critical point* in both contexts. Note that there might be *critical points* that are not *local minima* for (2.14). Recalling Example 2.15 with $\mathcal{X} = \mathbb{R}$, $\mathbf{0} \in \mathcal{N}_{\mathcal{X}}(0) + \partial_C h(0) = [-1, 1]$ yet $x = 0$ is actually a *strict local maximum*. We underline that when F is smooth on $\mathcal{X} = \mathbb{E}$, *Clarke-criticality* simply boils down to the well-known necessary first-order optimality condition

$$\mathbf{0} \in \partial_C F(\hat{x}) = \{\nabla F(\hat{x})\} \Rightarrow \nabla F(\hat{x}) = \mathbf{0}.$$

A stronger notion than *Clarke-criticality*, strictly weaker than *local optimality*, requires the positivity of the directional derivatives along *feasible directions*.

Definition 2.29 (Feasible direction). Let $\mathcal{X} \subseteq \mathbb{E}$ be convex and $u^\circ \in \mathbb{E} \setminus \{0\}$. We call u° a *feasible direction* at x (with respect to $\mathcal{X}$) if there exists $\varsigma > 0$ such that $x + \varsigma \cdot u^\circ \in \mathcal{X}$. Equivalently, u° must belong to the *polar* of $\mathcal{N}_{\mathcal{X}}(x)$,

$$\mathcal{N}_{\mathcal{X}}^\circ(x) := \left\{ \tilde{u}^\circ \in \mathbb{E} \mid \langle \tilde{x}, \tilde{u}^\circ \rangle \leq 0 \quad \forall \tilde{x} \in \mathcal{N}_{\mathcal{X}}(x) \right\}. \quad (2.25)$$

Proposition 2.30 (Stationarity condition, building on [RW04] Theorem 8.15). Let $\mathcal{X} \subseteq \mathbb{E}$ be a convex set and let $\hat{x} \in \mathcal{X}$. If the point $\hat{x}$ is a local minimum for problem (2.14) with F directionally differentiable on an open set containing $\text{rint}(\mathcal{X})$ then the following inclusion must hold

$$F'(\hat{x}; u^\circ) \geq 0 \quad \forall u^\circ \in \mathcal{N}_{\mathcal{X}}^\circ(\hat{x}) \setminus \{0\}. \quad (\text{stationarity})$$

Moreover, if $F = f_1 - f_2$ fulfills the requirements of Proposition 2.26 with f_1 and f_2 being both convex, the above necessary (local) optimality condition reads

$$\partial f_2(\hat{x}) \subseteq \mathcal{N}_{\mathcal{X}}(\hat{x}) + \partial f_1(\hat{x}). \quad (\text{DC-stationarity})$$

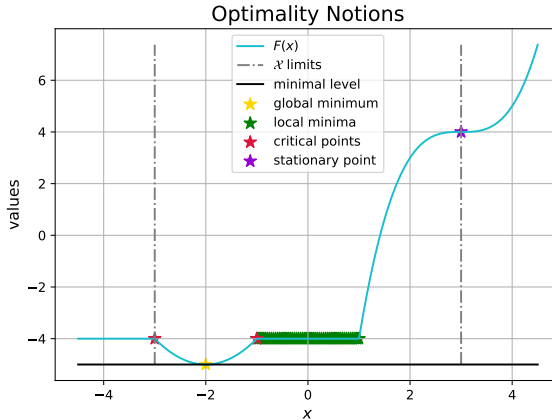


Fig. 2.1 Global optimality $\subsetneq$ Local Optimality $\subsetneq$ Stationarity $\subsetneq$ Criticality

Remark 2.31. Let $x \rightarrow F(x) = \min\{x^2, \max\{(x-3)^3 + 3, -5\}\} + \min\{(x+2)^2, 1\}$ and let $\mathcal{X} = [-3, 3]$. One can check that mere *critical points* are found at $\{-3, -1\}$ whereas *local minima* spread on $] -1, 1]$. Precisely at $x = 3$, $\mathcal{N}_{\mathcal{X}}(3) = [0, \infty[$. Thereby, $\mathcal{N}_{\mathcal{X}}^\circ(3) = \mathbb{R}_-$. For any $u^\circ \in \mathbb{R}_- \setminus \{0\}$, one computes $F'(0; u^\circ) = 3(3-3)^2 \cdot u^\circ = 0 \geq 0$. Hence, $\hat{x} = 3$ achieves (stationarity). Yet, $\hat{x}$ is a *strict global maximum* hence not locally minimal.

PART I

Tunable Oracle Methods

3

Optimal Inexactness Schedules

3.1 Inexactness in numerical optimization

Typical iterative optimization schemes rely on key ingredients often referred to as *oracles* [HT20]. In what concerns continuous optimization, with respect to both convex and nonconvex realms, the vast majority of the papers use two distinctive types of oracles at each iteration, namely the *informative* and *computational* ones [TJNO20].

(I) **Informative:** one assumes the possibility to obtain differential information about the objective (zero, first, $\dots$, higher-order) at successive query points.

(II) **Computational:** one assumes the ability to update sequences of iterates following some rules, often involving the resolution of easy or, at least, not too complicated, subproblems.

By default, one considers implicitly that both oracles yield *error-free* outputs. Nevertheless there exist cases for which such commodity appears unreasonable, e.g. when the objective value F in problem (1.1) itself stands

as the result of a non-trivial optimization problem (I) [ZYG⁺21] or when one cannot solve exactly subproblems in (II) [MOJ18]. Unfortunately, more and more problems of practical interest exhibit a structure that does not allow for *exact* oracles. Therefore there has been put a lot of efforts over the past years in dealing with inexactness about *informative* and *computational* oracles in order for widely spread algorithms (Bregman Proximal Gradient, Prox-Linear, their accelerated variants, etc.) [DP19, STG⁺21, YT21] to remain applicable in such scenarios. As argued above, a common source of inexactness is due from the necessity of numerically solving auxiliary non-trivial optimization problems to produce oracles' outputs.

Example 3.1 (Saddle-point problems). For instance, [DGN13] analyzed the Gradient Method (GM) that allowed for what they defined as (δ, L, μ) inexact oracles. Consider saddle-point problems of the type

$$F^* = \min_{x \in \mathbb{R}^d} \left\{ F(x) := \max_{u \in \mathbb{R}^B} \mathcal{P}(u) + \langle \mathbf{A}u, x \rangle \right\} > -\infty \quad (3.1)$$

where $\mathcal{P}$ is continuously differentiable with a gradient $L(\mathcal{P})$ Lipschitz continuous and $\mu(\mathcal{P}) > 0$ -strongly concave and $\mathbf{A} \in \mathbb{R}^{d \times B}$ is a matrix. They showed that F is $L(F) = \frac{\lambda_{\max}(\mathbf{A}\mathbf{A}^T)}{\mu(\mathcal{P})}$ smooth, $\mu(F) = \frac{\lambda_{\min}(\mathbf{A}\mathbf{A}^T)}{L(\mathcal{P})}$ -strongly convex and its gradient is given as $\nabla F(x) = \mathbf{A}u^*(x)$ where $u^*(x)$ is the exact minimizer of

$$\max_{u \in \mathbb{R}^B} \mathcal{P}(u) + \langle \mathbf{A}u, x \rangle. \quad (3.2)$$

In the general case, (3.2) cannot be solved exactly. Nevertheless, it can be approximately solved up to $\delta > 0$ global accuracy quite efficiently, e.g. by dedicated accelerated first-order methods [MF22]. That is, instead of $u^*(x)$, one can provide $u_x \in \mathbb{R}^B$ such that

$$\mathcal{P}(u^*(x)) + \langle \mathbf{A}u^*(x), x \rangle - (\mathcal{P}(u_x) + \langle \mathbf{A}u_x, x \rangle) \leq \delta \quad (3.3)$$

and use approximate gradient information $\nabla \tilde{F}(x) = \mathbf{A}u_x$ in GM as follows. Starting from $x_0 \in \mathbb{R}^d$, we compute the next iterate x_{k+1} of GM for $k \geq 0$ as

$$x_{k+1} = x_k - \frac{1}{L(F)} \nabla \tilde{F}(x_k) = x_k - \frac{1}{L(F)} \mathbf{A}u_{x_k}$$

Thereby and luckily enough, it happens that one can *tune* the quality of oracles by adequately choosing the amount of computational time spent on these problems. Looking back at the above example, it is well-known

that the number ω of iterations one must undertake to solve (3.2) up to δ accuracy (3.3) scales¹ as $\varphi(\mathcal{P})^{-\frac{1}{2}} \log(\delta^{-1})$ where $\varphi(\mathcal{P}) = \mu(\mathcal{P})/L(\mathcal{P})$. After K steps of GM involving a sequence of such inexact gradients with parameters $\{\delta_k\}_{k=0}^{K-1}$ at iteration $k = 0, \dots, K-1$ while using a constant stepsize $L^{-1} = (2L(F))^{-1}$, it is proven in [DGN13] that for

$$\hat{x}_K = \frac{\sum_{k=0}^{K-1} (1 - \varphi)^{K-1-k} x_k}{\sum_{k=0}^{K-1} (1 - \varphi)^{K-1-k}}$$

the following guarantees held

$$F(\hat{x}_K) - F^* \leq \frac{2^{-1} LD^2 (1 - \varphi)^K + 3 \sum_{k=0}^{K-1} (1 - \varphi)^{K-1-k} \delta_k}{\sum_{k=0}^{K-1} (1 - \varphi)^{K-1-k}} \quad (3.4)$$

with $\varphi = 4^{-1}\varphi(\mathcal{P})$ and D initial distance to a minimizer of (3.1). One observes the additive impact of such inexactness on GM's convergence.

It is worth noticing that inexactness also naturally occurs in the context of stochastic gradient methods. Within such framework one can tune stochastic gradient and/or Hessian's bias by averaging more or less sample gradients and/or Hessians [KLBC19, KL17]. Despite the fact that convergence guarantees under a different level of inexactness at each iteration are well-established in the literature, very few works devise specific inexactness schedules as in [Xu21, DN20]. This can provably constitute a miss of opportunity. Akin to our introductory example, let us assume that we dispose of relations linking the quality of oracles (δ) with the computational efforts invested in the creation of their output (ω). It is possible to retrieve optimal inexactness schedules taking into account the *trade-off* between the computational price and the harm in terms of convergence guarantees, e.g. (3.4), of the prescribed oracle precision.

We detail our optimality criteria in Section 3.3. Informally, we aim in this chapter at answering the question:

*How can we make the most of a computational budget
when using optimization methods dealing with oracle inexactness?*

¹In practice, one may resort to *warm-start* techniques for the inner maximization problem reducing the oracle cost. Since the impact of such hack is not predictable, our cost model stays conservative and agnostic to this regard.

3.1.1 Related work

At first sight and as is often the case in mathematical optimization research, the goal of this work can be summed up simply as an improvement of a worst-case convergence bound. Indeed, under the models we motivate in Section 3.2 and the assumptions we introduce in Section 3.3, we solve a specific instance of *non-linear allocation* problems to provide our enhanced inexactness schedules in terms of convergence guarantees per computational cost unit.

How straightforward might it sound, to the best of our knowledge there actually exists not so many previous works on that subject, i.e. *trade-off* optimality for iterative algorithms dealing with controllable inexactness. We contrast however this last sentence. First, a handful of works (e.g. papers [Kiw06, dOS14] or book [dOS20] Chapter 12) were dedicated to solve nonsmooth convex problems in which approximate first-order information is used. The quality of this latter is *on-demand* depending on the needs during algorithm execution. Similar to the present framework, more accurate oracles come with more computations. However, the inexactness defined therein is less general and they do not study explicitly the optimization of overall oracle cost. Second, by reassuring that a bunch of new papers propose criteria involving parameter fixed *relative* inexactness (see e.g. [WRDC21, PSJ22]). In opposition with the framework of *absolute* inexactness, computational costs to achieve *relative* inexactness are by nature prone to uncertainty and worst-case guarantees dependencies on this parameter often appears quite opaque. One usually leaves the relative inexactness parameter as fixed to a conservatively low wished terminal accuracy. Authors of [MF22] studied a mix between *relative* and *absolute* inexactness, highlighting the positive impact of incorporating *absolute* oracle accuracy.

With a similar approach to ours, [MAB12] focuses on (Accelerated) Proximal Gradient algorithms as analyzed by [SRB11] in their seminal paper. Unlike us, they do not show explicit closed-forms, i.e. analytical value for δ_k , method's level of inexactness at iteration $k = 0, \dots, K - 1$ as in (3.4), and their results remain mainly theoretical.

More recently, [Xu21] comes up with specific schedules for the inexactness in the auxiliary subproblems they deal with in order to lower the over-

all computational load after $K \in \mathbb{N}$ iterations. Our more general results encompass theirs if we were concerned with the same context of inexact Augmented Lagrangian. As a byproduct of an asymptotical analysis, i.e. $K \rightarrow \infty$, [VSBV13] whose follow-up resides in [MF22], suggests to use inexactness schedules that decrease sufficiently fast in order to maintain (up to a logarithmic factor) the rate of convergence of the *error-free*, i.e. *exact*, counterpart of the algorithms at scope. In [DN20] were tried *online* (adaptive) as well as purely *offline* schedules in what concerns the accuracy of the inexact higher-order tensor steps. Both last works provide *non-constant* inexactness schedules but none of them brings into balance (inexact) oracles' computational complexities.

3.1.2 Contributions

Let us consider an iterative algorithm involving controllable inexactness, e.g GM as depicted in the previous paragraph. We can state the main contributions of this chapter as follows.

- Firstly, considering fixed the number $K \in \mathbb{N}$ of inexact oracles calls, we propose a systematic *offline* procedure to devise the amount δ_k of inexactness to adopt at each iteration $k \in \{0, \dots, K-1\}$ based on an oracle cost model and algorithm's guarantees. To that end, we solve a rather general *non-linear assignment* problem, result of independent interest. Building upon this, we present closed-forms results about the optimal inexactness schedules for a broad class of oracle cost models, directly inspired from practical scenarios.
- Secondly, we propose an *online* heuristic extension in which neither K nor the overall computational budget allocated must be fixed beforehand.
- Thirdly, we conduct numerical experiments that sustain the validity of our approach, either *offline* or *online*, by comparing it against (a) *constant* inexactness schedules and (b) *non-constant* inexactness schedules from the literature [VSBV13, MF22].

We put emphasis on the fact that our strategies are fully implementable.

3.1.3 Outline

In Section 3.2 are motivated and then explained the concept of Tunable (Inexact) Oracles, i.e. we develop the models of costful (inexact) oracles and

convergence under an inexactness schedule which we illustrate thanks to three examples serving as guidelines. We then proceed to Section 3.3 in which we show all the contributions teased just above. This section represents the main content of this chapter. We conclude by quantifying the computational savings of our approach on the aforementioned guideline examples in Section 3.4.

3.2 Tunable (Inexact) Oracles

Here below we aim at defining the class of Tunable Oracles Methods (TOM) from which one can benefit by adapting the amount of inexactness in the oracles involved at each iteration. Prior to this goal, we recall some findings about methods incorporating inexactness and we introduce our assumed oracle cost model. Finally, we substantiate the concept of Tunable Oracles Methods with three complete examples serving as common thread.

3.2.1 Impact of Inexact Oracles

The worst-case behaviour analysis of iterative methods relying on inexact oracles displays some favorable structure. Let $K \in \mathbb{N}$ be the number of performed iterations, one defines the amount of inexactness at iteration $k \in \{0, \dots, K-1\}$ as δ_k . Researchers come up with convergence models

$$\mathcal{L}(K) \leq \mathcal{E}(\{\delta_k\}_{k=0}^{K-1}) \quad (3.5)$$

where

- $\mathcal{L}(K)$ acts as a positive *gauge* one aims to minimize.
- $\mathcal{E} : \mathbb{R}_+^K \rightarrow \mathbb{R}_+$ informs about a method's convergence guarantee, with respect to $\mathcal{L}(K)$, under the impact of a sequence $\{\delta_k\}_{k=0}^{K-1}$.

$\mathcal{L}$ gauges include gradient mapping norms [DP19], functional gaps [STG⁺21, DGN13]. As in Example 3.1 inequality (3.4), a possible instance for our model could show up as

$$\underbrace{F(\hat{x}_K) - F^*}_{\mathcal{L}(K)} \leq \underbrace{\frac{2^{-1} L D^2 (1 - \varphi)^K + 3 \sum_{k=0}^{K-1} (1 - \varphi)^{K-1-k} \delta_k}{\sum_{k=0}^{K-1} (1 - \varphi)^{K-1-k}}}_{\mathcal{E}(\{\delta_k\}_{k=0}^{K-1})}.$$

Remark 3.2. Within the *error-free* framework, i.e. $\delta_k = 0$ for every integer k ,

$$\lim_{K \rightarrow \infty} \mathcal{E}(\{0\}_{k=0}^{K-1}) = 0$$

When accounting for errors, one theoretically observes convergence only up to $\mathcal{E}(\{\delta_k\}_{k=0}^{K-1})$ accuracy. It can happen that the schedule $\{\delta_k\}_{k=0}^{K-1}$ does not decrease sufficiently fast leading to a well-known phenomenon referred to as *error-accumulation* in the literature [Dev13]. This latter translates to

$$\lim_{K \rightarrow \infty} \mathcal{E}(\{\delta_k\}_{k=0}^{K-1}) = \infty.$$

3.2.2 Cost of Inexact Oracles

Without loss of generality, one can define a *reference* inexactness $\bar{\delta} > 0$ together with constants $0 \leq M^- < 1 < M^+ < \infty$ such that the inexactness of the oracles falls in the segment $\Xi := [M^- \bar{\delta}, M^+ \bar{\delta}]$. This means that at each iteration $k \in \{0, \dots, K-1\}$, one allows the user to freely pick up any $\delta_k \in \Xi$. As previously unveiled, such request costs a computational tribute, namely $\mathcal{B}_k(\delta_k) \geq 0$. Suitable for a variety of applications [MF22, DP19, KLBC19], we suggest the oracle cost model

$$\mathcal{B}_k(\delta_k) = b_k \chi(\delta_k) \tag{3.6}$$

where

- $b_k > 0$ denotes the cost distortion of iteration k .
- $\chi : \Xi \rightarrow \mathbb{R}_+$ dictates how the cost fluctuates with δ_k .

Recalling once again Example 3.1, one identifies the *a priori* number ω_k of inner-iterations to obtain δ_k inexact information about F as a multiple of

$$\mathcal{B}_k(\delta_k) = \underbrace{\sqrt{\varphi(\mathcal{P})^{-1}}}_{b_k} \underbrace{\log(\delta_k^{-1})}_{\chi(\delta_k)}.$$

Remark 3.3. When unknown or when no argument justifies that any iteration turns out to be less expensive than another, we arbitrarily set $b_k = 1$ for any integer $k \geq 0$. Let us also point out that in the absence exogenous indication, $M^- = 0$ and $M^+ = \infty$.

3.2.3 Tunable Oracle Method

An instance of Tunable Oracle Methods (TOM) stands as a K steps *iterative* algorithm $\mathcal{A}$ whose iterations involve a notion of *controllable* δ -inexactness as in (3.6) such that $\mathcal{A}$ converges in the sense of (3.5). Furthermore, in order to truly exploit such *controllable* feature, we require the explicit knowledge of $\{b_k\}_{k=0}^{K-1}$, $\mathcal{E}$ and χ , up to a multiplicative constant factor. Such desire emphasizes the *offline* nature focus of this work at this stage. We make use of *a priori* information about the behaviour of $\mathcal{A}$ and the expected cost one should encounter while requesting a schedule of $\{\delta_k\}_{k=0}^{K-1}$ inexact oracles. We explain later on how to take advantage of TOM in an *online* setting.

3.2.4 Guideline examples

Throughout the sequel, we make use of the following:

- $\Psi : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{\infty\}$ proper closed convex function (not necessarily smooth), i.e. $\text{dom } \Psi \neq \emptyset$ and Ψ is lower semi-continuous.
- $\ell : \mathbb{R}^p \rightarrow \mathbb{R}$ closed convex function, $L_\ell < \infty$ Lipschitz continuous and simple, i.e. its proximal operator

$$\text{prox}_\ell : \mathbb{R}^d \rightarrow \mathbb{R}^d, \bar{x} \rightarrow \text{prox}_\ell(\bar{x}) := \arg \min_{\tilde{x} \in \mathbb{R}^d} \ell(\tilde{x}) + \frac{1}{2} \|\tilde{x} - \bar{x}\|^2$$

can be computed at negligible cost².

- $c : \mathbb{R}^d \rightarrow \mathbb{R}^p$ with $L_{\nabla c} < \infty$ Lipschitz continuous Jacobian ∇c .

Combining these ingredients, we introduce $f(\cdot) = \ell(c(\cdot))$ and $\mathcal{F}(\cdot) = f(\cdot) + \Psi(\cdot)$ nonconvex, nonsmooth in general. We will assume that

$$F^* = \min_{x \in \mathbb{R}^d} \left\{ \mathcal{F}(x) := f(x) + \Psi(x) \right\} > -\infty. \quad (3.7)$$

Example 3.4 (Composite optimization under inexact proximal operator). The inexact Accelerated Forward-Backward algorithm (iAFB) from [MF22] tackles so called convex composite problems, ubiquitous in image processing [CP16]. The specificities of this class read: $p = 1$, $\ell = \text{id}_{\mathbb{R}}$ thus $f = c$, convexity (respectively $\mu \geq 0$ strong convexity) of c (respectively Ψ) and ∇f is $L_f = L_{\nabla c}$ Lipschitz continuous. Among other appealing features,

²Existence of a closed-form or at the expense of an easy one dimensional segment-search.

iAFB presented therein allows for Ψ whose proximal mapping is not simple, e.g. in *sparse overlapping groups regularization* [RJGF10]. At $z \in \mathbb{R}^d$, $\lambda > 0$, the primal-dual pair of problems related to the proximal step of a function $\mathcal{P}$ translates to

$$\begin{aligned} \min_{x \in \mathbb{R}^d} \left\{ F_{\downarrow}(x; \lambda, z, \mathcal{P}) &:= \frac{1}{2} \|x - z\|^2 + \lambda \mathcal{P}(x) \right\} & (\text{Primal}) \\ \max_{u \in \mathbb{R}^d} \left\{ F_{\uparrow}(u; \lambda, z, \mathcal{P}) &:= \frac{1}{2} (\|z\|^2 - \|z - \lambda u\|^2) - \lambda \mathcal{P}^*(u) \right\} & (\text{Dual}) \end{aligned} \quad (3.8)$$

where $\mathcal{P}^*$ here depicts the Fenchel conjugate of $\mathcal{P}$, i.e. the operator $\mathcal{P}^* : \mathbb{R}^d \rightarrow \bar{\mathbb{R}}$, $u \rightarrow \mathcal{P}^*(u) = \sup_{x \in \mathbb{R}^d} \langle u, x \rangle - \mathcal{P}(x)$. iAFB produces iterates x_k , y_k , z_k , u_k thanks to coefficients $\lambda_k \in \mathcal{O}(L_f)$ and $A_k \in \Omega(k^2)$ for any integer $k \geq 0$. Either iAFB employs predefined sequence of stepsizes $\{\lambda_k^{-1}\}_{k \in \mathbb{N}}$ and then A_k , accounting as certificate sequence, can be computed in advance, i.e. *offline*, or it adopts Armijo line-search to adapt to local smoothness λ_k and one only has access to A_k for iterations $k' \geq k$, in an *online* fashion. Following authors' notations, (x_{k+1}, u_{k+1}) stems as a δ_k inexact output from the proximal step oracle at iteration $k \geq 0$ if

$$\text{PD}_{\lambda \mathcal{P}}(x, u; z) := F_{\downarrow}(x; \lambda, z, \mathcal{P}) - F_{\uparrow}(u; \lambda, z, \mathcal{P}) \leq \delta_k \quad (3.9)$$

with $x = x_{k+1}$, $u = u_{k+1} - \mu x_{k+1}$, $\lambda = \frac{\lambda_k}{1 + \lambda_k \mu}$, $\mathcal{P} = \Psi - \frac{\mu}{2} \|\cdot\|^2$ and $z = \frac{y_k - \lambda_k \nabla f(y_k)}{1 + \lambda_k \mu}$. Under this notion of inexactness, after $K \in \mathbb{N}$ steps, the following guarantees hold

$$\underbrace{\mathcal{F}(x_K) - F^*}_{\mathcal{L}(K)} \leq \underbrace{\frac{4D^2 + \sum_{k=0}^{K-1} (A_{k+1} \cdot (1 + \mu \lambda_k)^2 \cdot \lambda_k^{-1}) \delta_k}{A_K}}_{\mathcal{E}(\{\delta_k\}_{k=0}^{K-1})} \quad (3.10)$$

where $D < \infty$ denotes the distance from x_0 to the set of global minimizers. The presumed oracle cost, the work $\mathcal{B}_k(\delta_k)$ needed to produce (x_{k+1}, v_{k+1}) fulfilling (3.9) remains problem dependent. It is also highly influenced by the technique employed to solve the pair (Primal) / (Dual). In [MF22], authors deal with CUR factorization problem with *sparse overlapping groups regularization*. They simply use FISTA, a *fast* first-order method, to solve (Dual) yielding a linked sequence of primal recovered iterates (Primal).

3 | Optimal Inexactness Schedules

In the present setting, it is shown in [LJ16b] that the primal-dual gap decreases sublinearly as ω_k^{-1} where ω_k denotes the number of FISTA steps in (Dual). Then, for an arbitrary $b_k = 1$ (see Remark 3.3), we model the cost as

$$\mathcal{B}_k(\delta_k) = \underbrace{\delta_k^{-1}}_{\chi(\delta_k)}. \quad (3.11)$$

Example 3.5 (Composition of convex functions). The paper [DP19] considers the most general setting for which global optimality is, obviously, out of reach. Let us focus on a single method analyzed therein, the inexact Prox-Linear algorithm, iPL in short terms. Given $z \in \text{dom}(\Psi)$ and $t > 0$, one defines the functional

$$\mathcal{F}_t(\cdot; z) = \Psi(\cdot) + \ell(c(z) + \nabla c(z)(\cdot - z)) + \frac{t^{-1}}{2} \|\cdot - z\|^2.$$

We call δ inexact solution (x_+, ξ) of the minimization of $\mathcal{F}_t(\cdot; z)$ a pair fulfilling the properties: $\|\xi\| \leq \delta$ and

$$x_+ \in \arg \min_{x \in \mathbb{R}^d} \Psi(x) + \ell(\xi + c(z) + \nabla c(z)(x - z)) + \frac{t^{-1}}{2} \|x - z\|^2.$$

In the case of an exact $(x_+, 0)$ (i.e. $\delta = 0$ inexact) solution, the proximal gradient mapping $\mathcal{G}_t(x)$ at z with stepsize $t \leq (L_\ell L_{\nabla c})^{-1}$ stands as the vector $\mathcal{G}_t(x) = t^{-1}(x - x_+)$ if and only if $\mathcal{F}(x_+) \leq \mathcal{F}_t(x_+; z)$. Its norm constitutes a measure of stationarity [DP19] that generalizes the gradient norm in the smooth unconstrained optimization framework. When calling up $\{\delta_k\}_{k=0}^{K-1}$ inexact solutions at successive iterates $z = x_k \in \text{dom}(\Psi)$ and stepsizes $t = t_k \in]0, (L_\ell L_{\nabla c})^{-1}]$, iPL produces iterates $\{x_k\}_{k \geq 0}$ such that:

$$\underbrace{\min_{k' \in \{1, \dots, K\}} \|\mathcal{G}_{t_k}(x_{k'})\|^2}_{\mathcal{L}(K)} \leq \underbrace{\frac{\sum_{k=0}^{K-1} 4 t_k^{-1} (\mathcal{F}(x_k) - \mathcal{F}(x_{k+1}) + 4 L_\ell \delta_k)}{N}}_{\mathcal{E}(\{\delta_k\}_{k=0}^{K-1})}. \quad (3.12)$$

Again, the oracle cost should be derived from the complexity of the inner method used to obtain the sequence of $\{\delta_k\}_{k=0}^{K-1}$ inexact solutions at pairs $\{(x_k, t_k)\}_{k=0}^{K-1}$. When no further specific structure is taken as granted, excepted an easy proximal mapping of Ψ (unlike the previous example (3.10)), [DP19] suggests to exploit duality. At any iteration $k \geq 0$, one can

compute a δ_k inexact pair (x_{k+1}, ξ_{k+1}) as previously explained by finding a subgradient ξ_{k+1} of the (negated) Fenchel conjugate function of $\mathcal{F}_{t_k}(\cdot; x_k)$ whose norm does not exceed δ_k . With the oracles at hand and accounting for the fact that this conjugate boils down to a sum of a smooth convex term and a proximable nonsmooth convex function, A-HPE [MS13] ensures a minimal subgradient norm of order $\mathcal{O}(t_k \|\nabla c(x_k)\|^2 \omega_k^{-3/2})$ after ω_k iterations.

Hence, it comes that the number of A-HPE iterations required to get a δ_k inexact pair scales as $t_k^{2/3} \|\nabla c(x_k)\|^{4/3} \delta_k^{-2/3}$. Obviously, one does not know in advance the value $\|\nabla c(x_k)\|$ since it relies on x_k , only computable from iteration $k \geq 0$. Remark 3.3 then suggests that any $C > 0$, e.g. $C = 1$, can serve as artificial upper-bound for $\sup_{k \in \mathbb{N}} \|\nabla c(x_k)\|$ so that either iPL uses predefined stepsizes $\{t_k\}_{k \in \mathbb{N}}$ (*offline*) or adjusted stepsizes (*online*).

$$\mathcal{B}_k(\delta_k) = \underbrace{t_k^{2/3}}_{b_k} \underbrace{\delta_k^{-2/3}}_{\chi(\delta_k)}. \quad (3.13)$$

Note that if $\text{dom}(\Psi)$ is bounded with known diameter, one can use regularization and apply Nesterov fast gradient method on strongly convex composite objectives in order to obtain an enhanced $t_k^{1/2} \|\nabla c(x_k)\| \delta_k^{-1/2}$ complexity [Nes13]. We emphasize that *stopping criteria* are readily available to check whether δ_k accuracy has been reached.

Example 3.6 (Robust optimization on convex hull). Related to [MF22], [STG⁺21] extends the analysis of Fast Gradient Method (FGM) under the presence of inexactness by allowing f to be smooth and convex relative to some Legendre kernel function (see [LJ16b]), covering the so-called Bregman setting. Therefore, their results also encompass the Euclidean setting which we stick to for the sake of simplicity. We assume that Ψ represents the indicator function of a convex subset $\mathcal{X} \subseteq \mathbb{R}^d$, $p = 1$, $\ell = \text{id}_{\mathbb{R}}$ and c is $\mu \geq 0$ strongly convex. That is, f is $\mu \geq 0$ strongly convex and $\nabla f = \nabla c$ is $L_f = L_{\nabla c}$ Lipschitz continuous. Let $\Theta = \{\vartheta_i\}_{i=1}^n$ be a collection of $n \in \mathbb{N}$ vectors from $\mathbb{R}^d$ dubbed as *scenarios* and let $\omega > 0$. In *robust optimization*, one might be interested in minimizing the function

$$f : \mathcal{X} \rightarrow \mathbb{R}, x \rightarrow f(x) = \frac{\mu}{2} \|x\|^2 + \max_{\vartheta \in \text{conv}(\Theta)} \langle \vartheta, x \rangle - \frac{\omega}{2} \|\vartheta - \bar{\vartheta}\|^2$$

3 | Optimal Inexactness Schedules

for some *anchor scenario* $\bar{\vartheta}$, e.g. $\bar{\vartheta} = n^{-1} \sum_{i=1}^n \vartheta_i$. In other words, one would like to minimize a (regularized) linear objective taking into account that the cost vector could be any convex combination of previously encountered costs $\{\vartheta_i\}_{i=1}^n$. Akin to Example 3.1, we deduce that the *exact* gradient of f at any $x \in \mathcal{X}$ reads $\nabla f(x) = \mu x + \vartheta^*(x)$ where $\vartheta^*(x) = \arg \max_{\vartheta \in \text{conv}(\Theta)} \langle \vartheta, x \rangle - \frac{\omega}{2} \|\vartheta - \bar{\vartheta}\|^2$ and gradient's Lipschitz constant $L_f = \omega^{-1}$. An approximate maximizer $\vartheta_x \in \text{conv}(\Theta)$ of the problem defining f at x that verifies

$$\langle (\vartheta^*(x) - \vartheta_x), x \rangle + \frac{\omega}{2} (\|\vartheta_x - \bar{\vartheta}\|^2 - \|\vartheta^*(x) - \bar{\vartheta}\|^2) \leq \delta \quad (3.14)$$

can be used to construct $\tilde{f}(x) = \frac{\mu}{2} \|x\|^2 + \langle \vartheta_x, x \rangle - \frac{\omega}{2} \|\vartheta_x - \bar{\vartheta}\|^2 \simeq f(x)$ and $\nabla \tilde{f}(x) = \mu x + \vartheta_x \simeq \nabla f(x)$, providing $(2\delta, 2L_f + \mu, \mu)$ inexact information as originally understood in [Dev13]. [STG⁺21] shows that FGM ([STG⁺21], Algorithm 2) involving a sequence of $\{\delta_k\}_{k=0}^{K-1}$ inexact information and fixed stepsizes produces a final iterate x_K such that

$$\underbrace{\mathcal{F}(x_K) - F^*}_{\mathcal{L}(K)} \leq \underbrace{\frac{D^2 + 2 \sum_{k=0}^{K-1} A_{k+1} \delta_k}{A_K}}_{\mathcal{E}(\{\delta_k\}_{k=0}^{K-1})} \quad (3.15)$$

where, once again, $D < \infty$ denotes the distance from x_0 to the set of global minimizers of F . As for iAFB, the coefficients A_k for $k \geq 0$ serve as convergence certificates and are involved in subtle convex combinations of iterates within FGM. In the present setting, A_k 's value can be determined beforehand, $A_k \in O(\max\{k^2, (1 + \frac{1}{4}\sqrt{\mu/(\mu + 2L_f)})^{2k}\})$.

Let $\mathbf{O} = [\vartheta_1, \dots, \vartheta_n]^T$ and let $\hat{\varphi} := \frac{\lambda_{\min}(\mathbf{O}\mathbf{O}^T)}{\lambda_{\max}(\mathbf{O}\mathbf{O}^T)}$. By using another version of FISTA described in [Van22], one can take advantage of possible strong convexity, i.e. $\hat{\varphi} > 0$, of the usual reformulation of the inner-problem :

$$\max_{\vartheta \in \text{conv}(\Theta)} \langle \vartheta, x \rangle - \frac{\omega}{2} \|\vartheta - \bar{\vartheta}\|^2 = \max_{q \in \Delta^n} \langle \mathbf{O}^T q, x \rangle - \frac{\omega}{2} \|\mathbf{O}^T q - \bar{\vartheta}\|^2.$$

Assuming that $\hat{\varphi} > 0$, just as in Example 3.1, one can link the work ω_k to obtain δ_k accurate ϑ_{x_k} at iteration $k \geq 0$ by writing

$$\mathcal{B}_k(\delta_k) = \underbrace{\sqrt{\hat{\varphi}^{-1}}}_{b_k} \underbrace{\log(\delta_k^{-1})}_{\chi(\delta_k)} \quad (3.16)$$

whereas in the absence of strong convexity, i.e. $\hat{\varphi} = 0$, one would rather set $\mathcal{B}_k(\delta_k) = \sqrt{\lambda_{\max}(\mathbf{O}\mathbf{O}^T)} \delta_k^{-1/2}$ as common for smooth convex optimization [Nes13]. We explain in Section 3.4 how one can easily monitor the quality of an approximate solution for (3.14).

Remark 3.7. Usually, the workload $\omega \propto \mathcal{B}(\delta)$ to obtain (the output) of δ inexact oracles stays *by nature* an integer quantity, e.g. a number of inner-iterations (Example 3.4, 3.5 and 3.6). For the sake of simplicity however, we will accept that it varies continuously as in (3.11), (3.13) and (3.16).

3.3 Optimization of inexactness schedules

Now that the impact and the cost of inexact oracles have been introduced, we can elaborate about the main objective of this chapter. We assume for the time being that $K \in \mathbb{N}$ is fixed. One can try to suffer the smallest possible effect from inexact oracles in order to ensure the best worst-case convergence upper-bound. Obviously, one would like each δ_k to match its best value $M^- \bar{\delta}$. However, sometimes one cannot afford a schedule $\{\delta_k = M^- \bar{\delta}\}_{k=0}^{K-1}$ if one limits the overall computational budget. On the other hand, when not obliged to, it is not advisable to ask for the worst oracle accuracy at each iteration, i.e. $M^+ \bar{\delta}$ yielding the schedule $\{\delta_k = M^+ \bar{\delta}\}_{k=0}^{K-1}$.

Thereby, we propose to solve a master problem that aims at devising the optimal *trade-off* between the costs of oracles and worst-case guarantee harms due to their inexactness. From now on we will make a little abuse of notation about the *reference* inexactness $\bar{\delta} > 0$. Depending on the context, this latter will either depict a real value, a K steps inexactness schedule $\{\delta_k = \bar{\delta}\}_{k=0}^{K-1}$ or even the corresponding vector in $\mathbb{R}_+^K$, i.e. $\bar{\delta} \mathbf{1}_K$.

We start by specifying the structure of $\mathcal{E}$ in (3.5).

Assumption 3.8. We have access to $a > 0_K, \exists \tilde{\mathcal{E}} : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ strictly increasing

$$\mathcal{E}(\delta) := \mathcal{E}(\{\delta_k\}_{k=0}^{K-1}) = \tilde{\mathcal{E}}\left(\sum_{k=0}^{K-1} a_k \delta_k\right). \quad (\text{A})$$

Then, we invoke Assumption (B) suggesting that one is able to predict the

3 | Optimal Inexactness Schedules

overall cost $\mathcal{B}(\delta)$ of a schedule of inexactness δ . Furthermore, some technicalities about the function χ from (3.6) are stated.

Assumption 3.9. We have access to $b > \mathbf{0}_K$ and $\chi : \Xi =]M^-\bar{\delta}, M^+\bar{\delta}] \rightarrow \mathbb{R}_+$ differentiable (derivative noted χ'), invertible, strictly decreasing with

$$\mathcal{B}(\delta) := \sum_{k=0}^{K-1} \mathcal{B}_k(\delta_k) = \sum_{k=0}^{K-1} b_k \chi(\delta_k). \quad (\text{B})$$

$\chi' : \text{dom}(\chi') \supseteq \text{int}(\Xi) \rightarrow \mathbb{R}_-$ must be invertible and strictly increasing.

Remark 3.10. Assumption (A) tells us that on any subset of $\mathbb{R}_+^K$, minimizing $\mathcal{E}(\{\delta_k\}_{k=0}^{K-1})$ boils down to minimizing $\sum_{k=0}^{K-1} a_k \delta_k$, i.e. a minimizer of $a^T \delta$ stays optimal for $\mathcal{E}(\delta)$. Moreover, in what concerns our examples, one can identify problem dependent constants $C_1, C_2 > 0$ such that $\tilde{\mathcal{E}}$ admits a shared structure $\delta_{\text{weighted}} \rightarrow \tilde{\mathcal{E}}(\delta_{\text{weighted}}) = C_1 + C_2 \delta_{\text{weighted}}$.

Table 3.1 Illustration of Assumptions (A) and (B).

	$\chi(\delta_k)$	$\propto a_k$	$\propto b_k$
Example 3.4	δ_k^{-1}	$A_{k+1} \cdot (1 + \mu \lambda_k)^2 \cdot \lambda_k^{-1}$	1
Example 3.5	$\delta_k^{-2/3}$	t_k^{-1}	$t_k^{2/3}$
Example 3.6	$\log(\delta_k^{-1}) / \delta_k^{-1/2}$	A_{k+1}	1

In their adaptive *online* versions, sometimes practically more attractive, parameters λ_k (iAFB) and t_k (iPL) are not known in advance since they rely on line-searches at iteration $k \geq 0$. Unfortunately, they influence the values of a_k/b_k 's. Thenceforth, Example 3.4 and 3.5 would fail to satisfy Assumption (A) and/or (B). Nevertheless, as already argued, one can fix $\lambda_k \leq L_{\nabla c}^{-1}$ (iAFB), $t_k^{-1} \geq L_\ell \cdot L_{\nabla c}$ (iPL) for every $k \geq 0$ and avoid the line-searches. In such *offline* circumstances, both $\{a_k\}_{k \geq 0}$ and $\{b_k\}_{k \geq 0}$ become accessible and the assumptions are fulfilled. If taken constants, i.e. for every $k \geq 0$, $\lambda_k = \lambda$, $t_k = t$ for well-chosen $\lambda, t \in \mathbb{R}_+$, then $A_k \in \mathcal{O}(\max\{k^2, (1 - \sqrt{\mu/L_f})^{-k}\})$ for iAFB, $A_k \in \mathcal{O}(\max\{k^2, (1 + \frac{1}{4}\sqrt{\mu/(\mu + 2L_f)})^{2k}\})$ in FGD and $t_k \in \mathcal{O}(1)$ in iPL.

3.3.1 Master Problem

In its most standard version, we design a master problem $\delta^*(K, \bar{\delta}, M^-, M^+)$ for which the degrees of freedom reside in the accuracies of the K iterations of our TOM, i.e $\delta \in \mathbb{R}_+^K$. Given a *reference* $\bar{\delta} \in \mathbb{R}_+$ precision, we implicitly deduce the total allocated computational budget as $\sum_{k=0}^{K-1} \mathcal{B}_k(\bar{\delta}) = (\sum_{k=0}^{K-1} b_k) \chi(\bar{\delta})$. Controlling $\delta \in [M^-\bar{\delta}, M^+\bar{\delta}]^K$, we minimize the bound $\mathcal{E}(\delta)$ under the budget constraint $\mathcal{B}(\delta) = \mathcal{B}(\bar{\delta})$. Taking into account Assumption (A), (B) and Remark 3.10, this amounts to

$$\begin{aligned} \delta^*(K, \bar{\delta}, M^-, M^+) \in \arg \min_{M^-\bar{\delta} \leq \delta \leq M^+\bar{\delta}} & \sum_{k=0}^{K-1} a_k \delta_k \\ \text{s.t.} & \sum_{k=0}^{K-1} b_k \chi(\delta_k) = \mathcal{B}(\bar{\delta}). \end{aligned} \quad (3.17)$$

Remark 3.11. If $\lim_{\delta \rightarrow M^-\bar{\delta}} \chi(\delta) < \infty$ then we implicitly extend Ξ in Assumption (B) as $[M^-\bar{\delta}, M^+\bar{\delta}]$. Otherwise, lower-bounds $M^-\bar{\delta} \leq \delta$ become useless in problem (3.17) (basic feasible domain) and must be removed.

3.3.2 General solutions

We present a first theorem which will be proved to be useful to guarantee some consistency in the optimal schedules. Its scope encompasses more general structured *non-linear* resource allocation problems. Related results can be found deeply rooted in the literature (see e.g. [KL98]).

Theorem 3.12 (Workload consistency). Let $K \in \mathbb{N}$ and $\Xi =]\bar{\delta}^-, \bar{\delta}^+] \subseteq \mathbb{R}$. Let $\{q_{\uparrow,k}\}_{k=0}^{K-1}$ and $\{q_{\downarrow,k}\}_{k=0}^{K-1}$ be two sequences of continuously differentiable functions on Ξ . $\{q_{\uparrow,k}\}_{k=0}^{K-1}$ elements are furthermore strictly increasing on Ξ . If for every $k \in \{0, \dots, K-1\}$, the ratio $\mathcal{I}_k = q'_{\uparrow,k}/q'_{\downarrow,k}$ is strictly increasing and strictly negative on Ξ then for $\bar{\mathcal{B}} > 0$, any solution δ^* of

$$\min_{\delta \in [\bar{\delta}^-, \bar{\delta}^+]^K} \sum_{k=0}^{K-1} q_{\uparrow,k}(\delta_k) \quad \text{s.t.} \quad \sum_{k=0}^{K-1} q_{\downarrow,k}(\delta_k) = \bar{\mathcal{B}} \quad (3.18)$$

admits as a property that for all pairs of indices $(k_1, k_2) \in \{0, \dots, K-1\}^2$,

$$\mathcal{I}_{k_1} \geq \mathcal{I}_{k_2} \quad \Rightarrow \quad \delta_{k_1}^* \geq \delta_{k_2}^*. \quad (3.19)$$

3 | Optimal Inexactness Schedules

$$\mathcal{I}_{k_1} > \mathcal{I}_{k_2} \wedge \delta_{k_1}^*, \delta_{k_2}^* \in]\bar{\delta}^-, \bar{\delta}^+[\Rightarrow \delta_{k_1}^* > \delta_{k_2}^*. \quad (3.20)$$

Proof. We proceed in two sequential steps: proving (3.19) then (3.20) with the help of (3.19). Let us proceed by *absurdum* by assuming that there exists a pair of indices $(k_1, k_2) \in \{0, \dots, K-1\}^2$ such that

$$\mathcal{I}_{k_1} \geq \mathcal{I}_{k_2} \wedge \delta_{k_1}^* < \delta_{k_2}^*.$$

We consider the function

$$(\delta_{k_1}, \delta_{k_2}) \rightarrow f(\delta_{k_1}, \delta_{k_2}) := q_{\uparrow, k_1}(\delta_{k_1}) + q_{\uparrow, k_2}(\delta_{k_2}) + \sum_{k=0, k \neq k_1, k \neq k_2}^{K-1} q_{\uparrow, k}(\delta_k^*).$$

Trivially, the entries $\delta_{k_1}^*$ and $\delta_{k_2}^*$ introduced before form a solution of the two dimensional problem

$$\min_{(\delta_{k_1}, \delta_{k_2}) \in \Xi^2} f(\delta_{k_1}, \delta_{k_2}) \quad \text{s.t.} \quad q_{\downarrow, k_1}(\delta_{k_1}) + q_{\downarrow, k_2}(\delta_{k_2}) = \bar{B} - \sum_{k=0, k \neq k_1, k \neq k_2}^{K-1} q_{\downarrow, k}(\delta_k^*).$$

Let us show that $\delta_{k_1}^* < \delta_{k_2}^*$ is impossible under the hypotheses above.

KKT necessary conditions ensure the existence of $\tilde{\lambda} \in \mathbb{R}$, $\tilde{\mu}_{k_1, \oplus}, \tilde{\mu}_{k_1, \ominus} \geq 0$ and $\tilde{\mu}_{k_2, \oplus}, \tilde{\mu}_{k_2, \ominus} \geq 0$ with

$$\begin{aligned} q'_{\uparrow, k_1}(\delta_{k_1}^*) + \tilde{\lambda} q'_{\downarrow, k_1}(\delta_{k_1}^*) &= (\tilde{\mu}_{k_1, \ominus} - \tilde{\mu}_{k_1, \oplus}) \\ q'_{\uparrow, k_2}(\delta_{k_2}^*) + \tilde{\lambda} q'_{\downarrow, k_2}(\delta_{k_2}^*) &= (\tilde{\mu}_{k_2, \ominus} - \tilde{\mu}_{k_2, \oplus}) \\ (\bar{\delta}^- - \delta_{k_1}^*) \cdot \tilde{\mu}_{k_1, \ominus} &= 0 \\ (\bar{\delta}^- - \delta_{k_2}^*) \cdot \tilde{\mu}_{k_2, \ominus} &= 0 \\ (\bar{\delta}^+ - \delta_{k_1}^*) \cdot \tilde{\mu}_{k_1, \oplus} &= 0 \\ (\bar{\delta}^+ - \delta_{k_2}^*) \cdot \tilde{\mu}_{k_2, \oplus} &= 0. \end{aligned}$$

The only possible scenarios leading to $\bar{\delta}^- \leq \delta_{k_1}^* < \delta_{k_2}^* \leq u$ are achieved when $\tilde{\mu}_{k_1, \oplus} = \tilde{\mu}_{k_2, \ominus} = 0$ implying that

$$-\frac{q'_{\uparrow, k_2}(\delta_{k_2}^*) + \tilde{\mu}_{k_2, \oplus}}{q'_{\downarrow, k_2}(\delta_{k_2}^*)} = \tilde{\lambda} = -\frac{q'_{\uparrow, k_1}(\delta_{k_1}^*) - \tilde{\mu}_{k_1, \ominus}}{q'_{\downarrow, k_1}(\delta_{k_1}^*)}. \quad (3.21)$$

After elementary calculations, it must follow that

$$\frac{1 + \frac{\tilde{\mu}_{k_2, \oplus}}{q'_{\uparrow, k_2}(\delta_{k_2}^*)}}{1 - \frac{\tilde{\mu}_{k_1, \ominus}}{q'_{\uparrow, k_1}(\delta_{k_1}^*)}} = \frac{\mathcal{I}_{k_1}(\delta_{k_1}^*)}{\mathcal{I}_{k_2}(\delta_{k_2}^*)} < 1.$$

The inequality comes from the fact that $\{\mathcal{I}_k\}_{k=0}^{K-1}$ functions are strictly increasing and strictly negative on Ξ while $\delta_{k_1}^* < \delta_{k_2}^*$. Indeed, our inductive step leads to

$$\mathcal{I}_{k_1}(\delta_{k_1}^*) - \mathcal{I}_{k_2}(\delta_{k_1}^*) \geq 0 \quad \Rightarrow \quad 0 < \mathcal{I}_{k_1}(\delta_{k_1}^*)/\mathcal{I}_{k_2}(\delta_{k_1}^*) \leq 1$$

because of the strict negativity of $\{\mathcal{I}_k\}_{k=0}^{K-1}$. One then notices $\mathcal{I}_{k_2}(\delta_{k_1}^*) < \mathcal{I}_{k_2}(\delta_{k_2}^*)$. Therefore, one should observe that

$$0 \leq 1 + \frac{\tilde{\mu}_{k_2, \oplus}}{q'_{\uparrow, k_2}(\delta_{k_2}^*)} < 1 - \frac{\tilde{\mu}_{k_1, \ominus}}{q'_{\uparrow, k_1}(\delta_{k_1}^*)}$$

which turns out to be impossible since $\{q_{\uparrow, k}\}_{k=0}^{K-1}$ functions are strictly increasing on Ξ . We conclude that (3.19) is correct. Let us prove (3.20).

On the other hand, let us imagine this time that there exists a pair of indices $(k_1, k_2) \in \{0, \dots, K-1\}^2$ such that

$$\mathcal{I}_{k_1} > \mathcal{I}_{k_2} \wedge \bar{\delta}^- < \delta_{k_1}^* \leq \delta_{k_2}^* < \bar{\delta}^+.$$

Clearly, (3.19) already shows that $\delta_{k_1}^* \not\leq \delta_{k_2}^*$. The only plausible scenario left resumes in $\bar{\delta}^- < \delta_{k_1}^* = \delta^* = \delta_{k_2}^* < \bar{\delta}^+$. Setting $\tilde{\mu}_{k_1, \oplus} = 0 = \tilde{\mu}_{k_2, \ominus}$ in (3.21) yields

$$-\mathcal{I}_{k_2}(\delta^*) = \tilde{\lambda} = -\mathcal{I}_{k_2}(\delta^*).$$

Again, this contradicts our assumption that $\mathcal{I}_{k_1} > \mathcal{I}_{k_2}$. □

Example 3.13. The reader can get more insight thanks to the following example. For every $k \in \{0, \dots, K-1\}$, let $\delta \rightarrow q_{\uparrow, k}(\delta) = (k+1)^3 \delta$, $\delta \rightarrow q_{\downarrow, k}(\delta) = -\log(\delta)$, $\bar{\delta}^- = 1$ and $\bar{\delta}^+ = 10^5$. The functional ratios $\mathcal{I}_k$ are given by

$$\mathcal{I}_k :]\bar{\delta}^-, \bar{\delta}^+[\rightarrow \mathbb{R}, \delta \rightarrow \mathcal{I}_k(\delta) = \frac{q'_{\uparrow, k}(\delta)}{q'_{\downarrow, k}(\delta)} = \frac{(k+1)^3}{-\delta^{-1}} = -(k+1)^3 \delta^{-1}.$$

On $\Xi = [\bar{\delta}^-, \bar{\delta}^+]$, $q_{\uparrow, k}$'s and $q_{\downarrow, k}$'s are continuously differentiable and $q_{\uparrow, k}$'s are increasing while $\mathcal{I}_k$'s are strictly increasing and strictly negative.

3 | Optimal Inexactness Schedules

In addition, if $k_1 \leq k_2$,

$$\mathcal{I}_{k_1} \geq \mathcal{I}_{k_2}$$

since for every $\delta \in \Xi$, $\mathcal{I}_{k_1}(\delta) = -(k_1 + 1)^3 \delta^{-1} \geq -(k_2 + 1)^3 \delta^{-1} = \mathcal{I}_{k_2}(\delta)$.

Prior to displaying optimal inexactness schedules, let us define the concept of *descending ranks*, convenient mapping that will allow to rewrite solutions of our master problem (3.17) in a more reader friendly fashion.

Definition 3.14 (Descending rank). Let $v = (v_0, \dots, v_{K-1}) \in \mathbb{R}^K$ and let $\psi : \{0, \dots, K-1\} \rightarrow \{0, \dots, K-1\}$ be a bijection setting the vector $\hat{v} = (v_{\psi(-1)(0)}, \dots, v_{\psi(-1)(K-1)})$ in *descending mode*. For any $k \in \{0, \dots, K-1\}$, we call $\psi(k)$ the *descending rank* of v_k (with respect to ψ).

Remark 3.15. It comes that if $\psi(k) = \tilde{k}$ then v_k is, according to the sorting induced by ψ , the $(\tilde{k} + 1)$ -th biggest element of v .

Now, in the case where all the ratios $\{\mathcal{I}_k\}_{k=0}^{K-1}$ can be ordered, i.e. there exists a bijective mapping $\mathcal{M} : \{0, \dots, K-1\} \rightarrow \{0, \dots, K-1\}$ such that for all pair of indices $(k_1, k_2) \in \{0, \dots, K-1\}^2$,

$$\mathcal{M}(k_1) < \mathcal{M}(k_2) \quad \Rightarrow \quad \mathcal{I}_{k_1} \geq \mathcal{I}_{k_2}, \quad (3.22)$$

Theorem 3.12 suggests that the entries $\{\delta_k^*\}_{k=0}^{K-1}$ of an optimal solution δ^* of (3.18) can be *ranked* as well. Provided a suitable comparison vector $v \in \mathbb{R}^K$, we advocate the usefulness of ψ , *descending rank* function based on v to act like a $\mathcal{M}$ function above, i.e. for any $k \in \{0, \dots, K-1\}$, one has

$$\psi(k) = \mathcal{M}(k). \quad (3.23)$$

As displayed in Theorem 3.16, Theorem 3.12 applies verbatim to problem (3.17) by picking for every $k \in \{0, \dots, K-1\}$, $q_{\uparrow,k}(\delta_k) = a_k \delta_k$, $q_{\downarrow,k}(\delta_k) = b_k \chi(\delta_k)$ and $\bar{\mathcal{B}} = \mathcal{B}(\bar{\delta})$. One valid comparison vector that fulfills (3.22) and (3.23) would be $v_k = b_k/a_k$.

Let δ^* be an optimal solution for (3.17). We summarize last paragraph's key content by underlining the fact that the biggest the value of b_k/a_k will be, the biggest the optimal inexactness at iteration $k \in \{0, \dots, K-1\}$, according to our master problem, will be as well.

$$\psi(k_1) < \psi(k_2) \quad \Rightarrow \quad v_{k_1} = \frac{b_{k_1}}{a_{k_1}} \geq \frac{b_{k_2}}{a_{k_2}} = v_{k_2} \quad \Rightarrow \quad \delta_{k_1}^* \geq \delta_{k_2}^*.$$

Theorem 3.16. Let Assumptions (A) and (B) hold with $(a, b) \in \mathbb{R}_{++}^{K \times 2}$, $M^- < 1 < M^+$ and $\chi :]M^- \bar{\delta}, M^+ \bar{\delta}] \rightarrow \mathbb{R}_+$ convex³. $\exists K_\oplus, K_\ominus \in \{0, \dots, K-1\}$, $\lambda^* \in \mathbb{R}$ such that $\forall k \in \{0, \dots, K-1\}$,

$$\delta_k^* = \begin{cases} M^+ \bar{\delta} & k \in \oplus := \{\tilde{k} \mid \psi(\tilde{k}) < K_\oplus\} \\ (\chi')^{(-1)}\left(\frac{a_k \lambda^*}{b_k}\right) & k \in \mathcal{T} := \{\tilde{k} \mid K_\oplus \leq \psi(\tilde{k}) \leq K-1-K_\ominus\} \\ M^- \bar{\delta} & k \in \ominus := \{\tilde{k} \mid \psi(\tilde{k}) > K-1-K_\ominus\} \end{cases} \quad (3.24)$$

where $\psi(k)$ depicts the descending rank of $v_k = b_k/a_k$, λ^* satisfies the equality

$$\begin{aligned} & \sum_{k=0}^{K-1} b_k \chi(\bar{\delta}) - \left[\chi(M^+ \bar{\delta}) \left(\sum_{k \in \oplus} b_k \right) + \chi(M^- \bar{\delta}) \left(\sum_{k \in \ominus} b_k \right) \right] \\ & = \\ & \sum_{k \in \mathcal{T}} b_k \chi\left((\chi')^{(-1)}\left(\frac{a_k \lambda^*}{b_k}\right)\right). \end{aligned} \quad (3.25)$$

and δ^* stands as a solution of (3.17).

Proof. A neutral schedule $\{\delta_k = \bar{\delta}\}_{k=0}^{K-1}$ certifies the feasibility of problem (3.17) whose basic domain Ξ^K is compact. Since $\bar{\delta} \in \text{int}(\Xi) =]M^- \bar{\delta}, M^+ \bar{\delta}[$ is fixed, according χ definition within Assumption (B), we can write

$$0 < \sum_{k=0}^{K-1} b_k \chi(\bar{\delta}) = \bar{B} < \infty.$$

Before diving into KKT conditions, let us point out two observations:

* With $M^- < 1$, $\{\delta_k = M^- \bar{\delta}\}_{k=0}^{K-1}$ is clearly not feasible since χ is assumed to be strictly decreasing on Ξ and $b > \mathbf{0}_K$ (Assumption (B)).

* With $M^+ > 1$, $\{\delta_k = M^+ \bar{\delta}\}_{k=0}^{K-1}$ is clearly not optimal since the neutral schedule above is feasible and

$$\sum_{k=0}^{K-1} a_k \bar{\delta} = \|a\|_1 \bar{\delta} < M^+ \|a\|_1 \bar{\delta} = \sum_{k=0}^{K-1} a_k M^+ \bar{\delta}$$

as $a > \mathbf{0}_K$ according to Assumption (A).

³If χ is bounded at $M^- \bar{\delta}$, the closed interval is considered (see Remark 3.11).

3 | Optimal Inexactness Schedules

Assumptions of Theorem 3.16 satisfy the ones required by Theorem 3.12 so that this latter applies verbatim to problem (3.17) with $\varrho_{\uparrow,k'}(\cdot) = a_k(\cdot)$ and $\varrho_{\downarrow,k}(\cdot) = b_k \chi(\cdot)$. The negative sign of χ' ensures the soundness of our ranking (cfr. (3.22)).

$$\psi(k_1) < \psi(k_2) \Rightarrow v_{k_1} = b_{k_1}/a_{k_1} \geq b_{k_2}/a_{k_2} = v_{k_2}$$

leading to

$$\mathcal{I}_{k_1}(\cdot) = a_{k_1}/(b_{k_1} \chi'(\cdot)) \geq a_{k_2}/(b_{k_2} \chi'(\cdot)) = \mathcal{I}_{k_2}(\cdot).$$

Since $\varrho'_{\uparrow,k}/\varrho'_{\downarrow,k}$ ratios are strictly increasing and negative, we can claim that there exists $K_{\oplus}, K_{\ominus} \in \{0, \dots, K-1\}$ defining two subsets $\oplus = \{k \mid \psi(k) < K_{\oplus}\}, \ominus = \{k \mid \psi(k) > K-1-K_{\ominus}\}$ of $\{0, \dots, K-1\}$ such that δ_k^* will be affected to the upper-bound $\bar{\delta}^+ = M^+ \bar{\delta}$ if $k \in \oplus$ whereas it will be assigned the lower-bound $\bar{\delta}^- = M^- \bar{\delta}$ if $k \in \ominus$. By means of our two previous observations,

$$K_{\oplus}, K_{\ominus} < K.$$

If $(\oplus, \ominus)$ forms a partition of $\{0, \dots, K-1\}$ with

$$\sum_{k \in \oplus} b_k \chi(M^+ \bar{\delta}) + \sum_{k \in \ominus} b_k \chi(M^- \bar{\delta}) = \bar{\mathcal{B}} \quad (3.26)$$

then the inherent schedule $\delta_k = M^+ \bar{\delta}$ for every $k \in \oplus$, $\delta_k = M^- \bar{\delta}$ for every $k \in \ominus$ might be a solution of (3.17). The structure of such schedule is identical to the one expected by (3.24). We consider now other possible solutions. KKT necessary conditions (see LICQ below) imply that if δ^* is a local optimum of (3.17), then there exist $\tilde{\lambda} \in \mathbb{R}$, $\tilde{\mu}_{\ominus}, \tilde{\mu}_{\oplus} \in \mathbb{R}_+^K$ (dual feasibility) such that equalities and inequalities of Table 3.2 are jointly satisfied. When the context is not misleading, we make the abuse of notation for any $\delta \in \Xi^K$, $\chi'(\delta) = (\chi'(\delta_0), \dots, \chi'(\delta_{K-1}))$ and $\chi(\delta) = (\chi(\delta_0), \dots, \chi(\delta_{K-1}))$. Indeed, (3.17) enjoys the linear independence constraint qualification (LICQ) at any $\delta \in \Xi^K$ not matching the degenerate case (3.26). At any feasible $\delta \in \Xi^K$, the span of active constraints' gradients always involve $\chi'(\delta)$ whose entries are all non-zero, excepted eventually at $k_+ \in \oplus$ where $\delta_{k_+} = M^+ \bar{\delta}$ if $\chi'(M^+ \bar{\delta}) = 0$ (due to the convexity of χ and strict monotonicity of χ'). Gradients of other active constraints are vectors $\mathbf{e}_{k_+}$ for $k_+ \in \oplus$ and $\mathbf{e}_{k_-}$ for $k_- \in \ominus$ that are all linearly independent. Since we explicitly discard schedules of (3.26) type, it must exist an entry $k \in \{0, \dots, K-1\}$ where

Table 3.2 KKT conditions for (3.17)

$a + \tilde{\lambda} b \odot \chi'(\delta^*) = (\tilde{\mu}_\ominus - \tilde{\mu}_\oplus)$	stationarity
$M^- \bar{\delta} \leq \delta^* \leq M^+ \bar{\delta}$	primal feasibility (bounds)
$\langle b, \chi(\delta^*) \rangle = \bar{\mathcal{B}}$	primal feasibility (budget)
$(M^- \bar{\delta} - \delta^*) \odot \tilde{\mu}_\ominus = \mathbf{0}_K$	complementary (-)
$(\delta^* - M^+ \bar{\delta}) \odot \tilde{\mu}_\oplus = \mathbf{0}_K$	complementary (+)

bound constraints are inactive, hence $\chi'(\delta_k) < 0$. In order to obtain the $\mathbf{0}_K$ vector from a linear combination of active constraints' gradients, one should then take zero times $\chi'(\delta)$. If no bound constraint is active at δ then the induced linear combination immediately turns out to be the trivial one. Conversely, if some bound constraints are active then their associated coefficients in the linear combination must also be zero as shown previously, giving rise again to the null combination.

KKT conditions are sufficient since the relaxation (convex) of (3.17)

$$\min_{M^- \bar{\delta} \leq \delta \leq M^+ \bar{\delta}} \sum_{k=0}^{K-1} a_k \delta_k \quad \text{s.t.} \quad \sum_{k=0}^{K-1} b_k \chi(\delta_k) \leq \bar{\mathcal{B}} \quad (3.27)$$

would always lead to an optimal solution that tightens the budget constraint as $a > \mathbf{0}_K$ and χ is strictly decreasing. For writing ease, let us now denote

$$\mathcal{T} = \{0, \dots, K-1\} \setminus (\oplus \cup \ominus)$$

In line with the comments made so far, we look after a schedule $\{\delta_k^*\}_{k=0}^{K-1}$ of the form $\delta_k^* = M^+ \bar{\delta}$ if $k \in \oplus$, $\delta_k^* \in]M^- \bar{\delta}, M^+ \bar{\delta}[$ if $k \in \mathcal{T}$ and $\delta_k^* = M^- \bar{\delta}$ otherwise if $k \in \ominus$, for two positive integers $K_\oplus, K_\ominus$ with $K_\oplus + K_\ominus \leq K-1$. For any $k \in \mathcal{T}$, stationarity requirement compels

$$\begin{aligned} a_k + \tilde{\lambda} b_k \chi'(\delta_k^*) = 0 &\Rightarrow \chi'(\delta_k^*) = -(\tilde{\lambda} a_k^{-1} b_k)^{-1} = \frac{a_k \lambda^*}{b_k} \\ &\Rightarrow \delta_k^* = (\chi')^{(-1)} \left(\frac{a_k \lambda^*}{b_k} \right) \end{aligned}$$

for $\lambda^* = -(\tilde{\lambda})^{-1} \in \mathbb{R}$. Primal feasibility (budget) row of Table 3.2 is achieved by (3.25). $\square$

3 | Optimal Inexactness Schedules

Proposition 3.17 (Uniqueness). *If all the entries of $v \in \mathbb{R}_{++}^K$ differ, their ordering is unique, so becomes δ^* in problem (3.17) hence (3.24).*

Proof. We prove here the uniqueness of δ^* . For ease of writing, let us reuse the notation $\sum_{k=0}^{K-1} b_k \chi(\delta) = \bar{\mathcal{B}}$. Aside from the fact that all the entries v_k of v supposedly differ, Assumption (B) about χ certifies that for any fixed $K_\oplus$, $K_\ominus$, there can be at most one $\lambda^* \in \mathbb{R}$ ensuring (3.25). Seen from a different angle, for any couple $(K_\oplus, K_\ominus)$ there can be a single $\{\delta_k^*(K_\oplus, K_\ominus)\}_{k=0}^{K-1}$ respecting both (3.24) and (3.25). Let us assume by *absurdum* the optimality of two such schedules. Then, there must exist two couples $(K_{\oplus,1}, K_{\ominus,1})$ and $(K_{\oplus,2}, K_{\ominus,2})$ such that

$$\sum_{k=0}^{K-1} a_k \delta_k^*(K_{\oplus,1}, K_{\ominus,1}) = O^* = \sum_{k=0}^{K-1} a_k \delta_k^*(K_{\oplus,2}, K_{\ominus,2}).$$

Any schedule $\hat{\delta}^{(q)}$ in between for $q \in]0, 1[$ with

$$\hat{\delta}_k^{(q)} = q \cdot \delta_k^*(K_{\oplus,1}, K_{\ominus,1}) + (1 - q) \cdot \delta_k^*(K_{\oplus,2}, K_{\ominus,2})$$

for every $k \in \{0, \dots, K-1\}$ would trivially also yield the optimal cost O^* . By convexity of the set $S \subset \Xi^K$,

$$S = \left\{ \delta \in \Xi^K \left| \sum_{k=0}^{K-1} b_k \chi(\delta_k) \leq \bar{\mathcal{B}} \right. \right\}$$

we have that $\delta^*(K_{\oplus,1}, K_{\ominus,1}) \in S \wedge \delta^*(K_{\oplus,2}, K_{\ominus,2}) \in S$ implies $\hat{\delta}^{(q)} \in S$.

We examine two mutually exclusive situations.

* $\sum_{k=0}^{K-1} b_k \chi(\hat{\delta}_k^{(q)}) < \bar{\mathcal{B}}$. Then, it is possible to decrease at least one entry of $\hat{\delta}^{(q)}$ and obtain a strictly better objective value meaning that $\delta^*(K_{\oplus,1}, K_{\ominus,1})$ and $\delta^*(K_{\oplus,2}, K_{\ominus,2})$ were not jointly optimal: contradiction.

* $\sum_{k=0}^{K-1} b_k \chi(\hat{\delta}_k^{(q)}) = \bar{\mathcal{B}}$. In such scenario, the entire segment $[\delta^{(0)}, \delta^{(1)}]$ must be an edge of S . We can quickly invoke continuity arguments suggesting the existence of multiple schedules on that segment sharing both a common number of $\delta_k^* = M^+ \bar{\delta}$ (same $K_\oplus$ value) and a common number of $\delta_k^* = M^- \bar{\delta}$ (same $K_\ominus$ value). Yet by linearity their objective value would remain equal to O^* . As written above, this is impossible because there exists at most one valid schedule $\delta^*(K_\oplus, K_\ominus)$ by pair $(K_\oplus, K_\ominus)$: contradiction. $\square$

Example 3.18 (Toy example). We want here to give a first glimpse about the upcoming optimal schedules that will apply for (among others) situations reflected in Example 3.4, 3.5 and 3.6. To this purpose, we clarify the above notations by writing explicitly what they entail given an academic toy example rightfully meeting Assumptions (A) and (B). We consider that the *impact coefficients* a and *relative costs* b of oracles are given for any $k \in \{0, \dots, K-1 = 79\}$ by

$$a_k = (k+1), \quad b_k = \begin{cases} \frac{3}{420} & 0 \leq k < 20 \\ \frac{2}{420} & 20 \leq k < 40 \\ \frac{8}{420} & 40 \leq k < 80. \end{cases}$$

The *reference* inexactness parameter is chosen as $\bar{\delta} = 10^{-4}$ and $M^- = 0 < 1 < M^+ = 2$. The oracle cost $\chi : (0, 1] \rightarrow [0, \infty)$ is convex, differentiable and fluctuates poly-logarithmically with $0 \leq \delta \leq 1$ as $\delta \rightarrow \chi(\delta) = \log^2(\delta^{-1})$. χ' is negative and strictly increasing from $(0, 1]$ to $(-\infty, 0]$, its inverse $(\chi')^{(-1)} : (-\infty, 0] \rightarrow (0, 1]$ exists for $-\omega < 0$

$$-\omega \rightarrow (\chi')^{(-1)}(-\omega) = 2\omega^{-1} \mathcal{W}_0(\omega/2)$$

where $\mathcal{W}_0$ depicts the [Lambert](#) $\mathcal{W}$ function on its 0-branch. Here, Theorem 3.16 applies. Obviously, $\lim_{\delta \rightarrow 0} \chi(\delta) = \infty$ translates to $\Theta = \emptyset \Leftrightarrow K_\Theta = 0$. From that point, we can efficiently solve KKT conditions. They inform that for our present problem instance, $K_\oplus = 10$. Fortunately, the indices $k \in \{0, \dots, 79\}$ for which v_k values are the $K_\oplus$ biggest fall in $\oplus = \{0, \dots, 9\}$. We can conclude that the set $\mathcal{T}$ contains the indices $\{10, \dots, 79\}$.

We summarize the calculated optimal schedules for (3.17) in (3.28) bearing in mind that $\lambda = -\lambda^* \simeq 60.998423$. Figure 3.1 graphs the optimal inexactness schedule in (3.28).

$$\delta_k^* = \begin{cases} 2 \cdot 10^{-4} & k \in \{0, \dots, 9\} \\ \left(\frac{2(3/420)}{(k+1)\lambda}\right) \mathcal{W}_0\left(\frac{(k+1)\lambda}{2(3/420)}\right) & k \in \{10, \dots, 19\} \\ \left(\frac{2(2/420)}{(k+1)\lambda}\right) \mathcal{W}_0\left(\frac{(k+1)\lambda}{2(2/420)}\right) & k \in \{20, \dots, 39\} \\ \left(\frac{2(8/420)}{(k+1)\lambda}\right) \mathcal{W}_0\left(\frac{(k+1)\lambda}{2(8/420)}\right) & k \in \{40, \dots, 79\} \end{cases} \quad (3.28)$$

as well its sorted version in *descending* mode, thanks to the *descending ranks*.

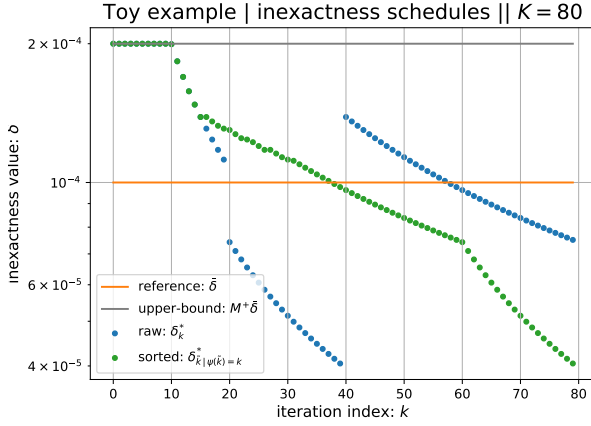


Fig. 3.1 Inexactness schedules: *reference*, *raw* optimal and *sorted* optimal

Remark 3.19. We underline that one only need to know a_k (respectively b_k) up to a common factor $C_a > 0$ (respectively $C_b > 0$). That is, it is sufficient to know $\tilde{a}_k$ (respectively $\tilde{b}_k$) such that for any $k \in \{0, \dots, K-1\}$, $a_k = C_a \tilde{a}_k$ (respectively $b_k = C_b \tilde{b}_k$). In Theorem 3.16, instead of looking for λ^* , one then searches for $\tilde{\lambda}^* = \lambda^* C_a / C_b$ that would act like $\tilde{\lambda}^* \tilde{a}_k / \tilde{b}_k = \lambda^* a_k / b_k$.

3.3.3 Closed-form solutions

Let us focus now on closed-form solutions we can obtain from the previous theorems when specifying a certain type of χ function, related to the practical examples motivating our oracle cost model (3.6). Yet, it usually remains to seize the correct values of $K_{\oplus}$ (number of iterations either performed at worst oracle accuracy or involving the least computational efforts) and $K_{\ominus}$ (number of iterations either achieved with the best oracle accuracy or demanding the heaviest computational efforts).

To circumvent a cautious search for the right pair $(K_{\oplus}, K_{\ominus}) \in \{0, \dots, K-1\}^2$, one can immediately detect whether $K_{\oplus} = 0 = K_{\ominus}$ with a simple trial in constant time. It is essentially what we achieve in Corollary 3.20. In such circumstances, $\oplus = \emptyset = \ominus$ (Theorem 3.16), i.e. the *transient set* $\mathcal{T}$ that normally collects the k -indices of iterations associated with v_k values smaller than the $K_{\oplus}$ biggest and bigger than the $K_{\ominus}$ smallest involves here all the iterations, i.e. $\mathcal{T} = \{0, \dots, K-1\}$.

Sticking to our current goal, we formally declare a meaningful functional family of χ functions that captures our applications of interest. For any $r > 0$, we define the convex function $\chi_r : \mathbb{R}_+ \rightarrow \mathbb{R}_{++}$ and the inverse of its derivative $(\chi'_r)^{(-1)} : \mathbb{R}_{--} \rightarrow \mathbb{R}_+$ ($\delta > 0$ and $\omega > 0$ here below),

$$\delta \rightarrow \chi_r(\delta) = \delta^{-r} \quad \Rightarrow \quad -\omega \rightarrow (\chi'_r)^{(-1)}(-\omega) = \left(\frac{\omega}{r}\right)^{-\frac{1}{r+1}}. \quad (3.29)$$

Corollary 3.20. Let Assumptions (A) and (B) hold with $(a, b) \in \mathbb{R}_{++}^{K \times 2}$, $M^- < 1 < M^+$ and $\chi = \chi_r :]M^- \bar{\delta}, M^+ \bar{\delta}] \rightarrow \mathbb{R}_+$. If $M^- \bar{\delta} < \delta^\circ < M^+ \bar{\delta}$ with

$$\delta^\circ = \bar{\delta} \cdot \left(\frac{\sum_{k=0}^{K-1} (b_k a_k^r)^{\frac{1}{r+1}}}{\sum_{k=0}^{K-1} b_k} \right)^{\frac{1}{r}} \cdot (b \odot a^{-1})^{\frac{1}{r+1}} \quad (3.30)$$

then δ° is optimal for (3.17).

For the sake of completeness, we also provide closed-form schedules in what concerns an extended family of χ_r functions. It includes the logarithmic model $\chi_0(\delta) = \log(\delta^{-1})$ from our introductory Example 3.1 as a pathological case $r \rightarrow 0$ and we allow that $K_\oplus, K_\ominus$ are not necessarily zero. Theorem 3.21 readily follows from Theorem 3.16, its proof is thus omitted.

Theorem 3.21. Let Assumptions (A) and (B) hold with $(a, b) \in \mathbb{R}_{++}^{K \times 2}$, $M^- < 1 < M^+$ and $\chi = \chi_{r \geq 0} :]M^- \bar{\delta}, M^+ \bar{\delta}] \rightarrow \mathbb{R}_+$. $\exists K_\oplus, K_\ominus \in \{0, \dots, K-1\}$, $\hat{\lambda} \in \mathbb{R}$ such that $\forall k \in \{0, \dots, K-1\}$

$$\delta_k^* = \begin{cases} M^+ \bar{\delta} & k \in \oplus := \{\tilde{k} \mid \psi(\tilde{k}) < K_\oplus\} \\ \hat{\lambda} \left(\frac{b_k}{a_k}\right)^{\frac{1}{r+1}} & k \in \mathcal{T} := \{\tilde{k} \mid N_\oplus \leq \psi(\tilde{k}) \leq K-1-K_\ominus\} \\ M^- \bar{\delta} & k \in \ominus := \{\tilde{k} \mid \psi(\tilde{k}) > K-1-K_\ominus\} \end{cases} \quad (3.31)$$

where $\psi(k)$ depicts the descending rank of $v_k = b_k/a_k$, $\hat{\lambda} > 0$ is defined as

$$\hat{\lambda} = \begin{cases} \bar{\delta} \cdot \left[\frac{(M^+)^{\sum_{k \in \oplus} b_k} \cdot (M^-)^{\sum_{k \in \ominus} b_k}}{\prod_{k \in \mathcal{T}} (b_k/a_k)^{-b_k}} \right]^{-\left(\sum_{k \in \mathcal{T}} b_k\right)^{-1}} & r = 0 \\ \bar{\delta} \cdot \left[\frac{\left(\sum_{k=0}^{K-1} b_k\right) - \left(\sum_{k \in \oplus} b_k\right) (M^+)^{-r} - \left(\sum_{k \in \ominus} b_k\right) (M^-)^{-r}}{\sum_{k \in \mathcal{T}} (a_k^r b_k)^{\frac{1}{r+1}}} \right]^{-\frac{1}{r}} & r > 0 \end{cases} \quad (3.32)$$

and δ^* stands as a solution of (3.17).

Interpretation Rather intuitively, an iteration whose *impact coefficient* a_k is bigger harms more the upper-bound $\mathcal{E}(\{\delta_k\}_{k=0}^{K-1})$ on the objective gauge $\mathcal{L}(K)$ (cfr. (3.5)) thus requires more precision, i.e. a small δ_k . However its associated cost model $B_k(\cdot) = b_k \chi(\cdot)$ (cfr. (3.6)) eventually reweights the accuracy of the oracle through b_k according to its relative computational burden with respect to the other iterations. Hence, δ_k^* are governed by *compound impacts* $v_k = b_k/a_k$,

$$\delta_k^* \propto \left(\frac{b_k}{a_k} \right)^{\frac{1}{r+1}}. \quad (3.33)$$

At fixed a, b , as r approaches 0, the oracles appear cheap and one can gain a lot by experiencing large deviations from $\bar{\delta}$. Conversely, we observe that if r tends to infinity then the cost of oracles spikes so that the constant schedule at $\bar{\delta}$ inexactness level, feasible, becomes optimal. Indeed, at any iteration, requesting an oracle accuracy even slightly better than $\bar{\delta}$ turns out to be extremely expensive and not affordable. Thus, we must observe $\mathcal{T} = \{0, \dots, K-1\}$ if r is large enough. Another situation in which one can easily predict that $\mathcal{T} = \{0, \dots, K-1\}$ or, equivalently, $K_{\ominus} = 0 = K_{\oplus}$, arises when $M^- = 0$ and $M^+ = \infty$. Indeed, akin to Example 3.18, the oracle cost blows up for $\delta \rightarrow M^- \bar{\delta} = 0$. In addition, if not obliged to, one has no advantage to choose an arbitrarily large $\delta \rightarrow M^+ \bar{\delta} = \infty$.

Remark 3.22. For didactical purposes, we express here below a possible asymptotical trend of optimal schedules of inexactness concerning Example 3.4 ($\mu = 0$, constant λ_k) based on (3.33) and Table 3.1

$$\delta_k^* \propto A_{k+1}^{-\frac{1}{2}} \in \mathcal{O}(k^{-1}).$$

Illustration We illustrate now the application of our previous theorems on an instance closely related to Example 3.4. Indeed, when $\mu = 0$ and $\lambda_k \in \mathcal{O}(1)$ for any integer $k \geq 0$, it is known that $a_k \in \mathcal{O}(k^2)$, see e.g. [STG⁺21]. We display the evolution of quantities $K_{\oplus}$ (Figure 3.2) and δ_k^* (Figure 3.3) with oracle's complexity parameter r and the factor of maximal tolerated inaccuracy M^+ . Let $M^- = 0$ and let the *reference* inexactness tolerated be $\bar{\delta} = 10^{-4}$. In what follows, $K_{\ominus} = 0$ in any situation since the oracle cost model $\chi_r(\delta) = \delta^{-r}$ explodes as $\delta \rightarrow 0 = M^- \bar{\delta}$. Accordingly, we assume for any $k \geq 0$ that $b_k = 1$ and $a_k = (k+1)^2$. We highlight two observations of interest.

■ About Figure 3.2. $K_{\oplus} > 0$ if K itself is big enough and $M^+ \bar{\delta}$ constrains our master problem (3.17), i.e. $M^+ \rightarrow 1$. In this latter case, the computational savings usually earned by investing in late iterations, requiring more care, must be spread out on earlier iterations since the biggest δ_k from any optimal schedule $\{\delta_k^*\}_{k=0}^{K-1}$ cannot take a value that falls way above the reference $\bar{\delta}$. In other words, $M^+ \bar{\delta}$ does not allow one to save a lot of efforts in the iterations linked with the smallest *impact factors*.

■ About Figure 3.3. The variability in the optimal schedules for inexactness $\{\delta_k^*\}_{k=0}^{K-1}$ heavily depends on the oracle cost parameter r , as emphasized by the power $(r+1)^{-1}$ in the relationship (3.33). Our findings discussed in the previous paragraph are graphically confirmed, e.g. when $r = 50$, δ_k^* is roughly equal to $\bar{\delta}$ for every $k \in \{0, \dots, K-1\}$.

3.3.4 Practical extensions

This chapter part is dedicated to direct extensions of the results unveiled so far. We adapt them to practical scenarios beyond the initial scope of TOM. Firstly, we investigate the modifications one should undertake to apply the concept of tunable oracle when, instead of the oracles' accuracies one would like to monitor the computational work invested in producing their outputs. Secondly, we propose an *online* strategy which preserves the structure of optimal inexactness schedules without the knowledge of K .

Work controlled

In some cases, one would like to manually specify the time spent at each iteration. In other words, instead of deciding δ_k that would *a priori* lead to a cost of $\mathcal{B}_k(\delta_k)$, we process the other way around. We choose the amount of computations $\omega_k \in [\omega_{M^+}, \omega_{M^-}]$ and we expect to incur a level of inexactness $\delta_k \sim \mathcal{B}_k^{(-1)}(\omega_k)$, i.e. $M^+ \bar{\delta}$ for a work of ω_{M^+} . Therefore, we can equivalently fix a *reference* total work $\bar{\omega}$ that acts as a surrogate for the budget term $\sum_{k=0}^{K-1} \mathcal{B}_k(\bar{\delta})$. We rewrite the objective from problem (3.17) as

$$\sum_{k=0}^{K-1} a_k \delta_k \sim \sum_{k=0}^{K-1} a_k \mathcal{B}_k^{(-1)}(\omega_k) = \sum_{k=0}^{K-1} a_k \chi_r^{(-1)}(\omega_k/b_k).$$

The homogeneity of χ_r allows to write for any $C > 0$, $r > 0$ and $S \subseteq \mathbb{R}_+^K$,

$$\arg \min_{\omega \in S} \sum_{k=0}^{K-1} \chi_r^{(-1)}(\omega_k \cdot C/b_k) = \arg \min_{\omega \in S} \sum_{k=0}^{K-1} \chi_r^{(-1)}(\omega_k/b_k) \quad (3.34)$$

3 | Optimal Inexactness Schedules

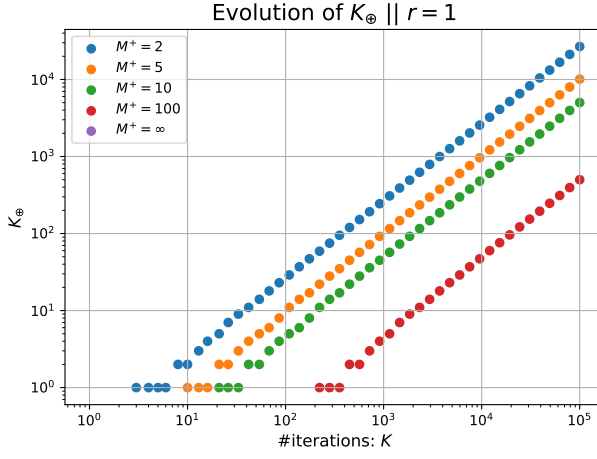


Fig. 3.2 $K_{\oplus}$ grows with *selected* K , this effect starts earlier when (3.17) is more constrained, i.e. $M^+ \rightarrow 1$.

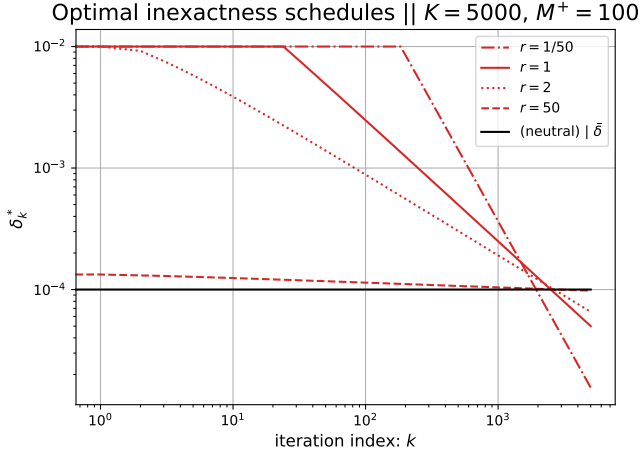


Fig. 3.3 K fixed, $r \rightarrow 0$ means cheaper oracles and thus more aggressive schedules (see $r = 50^{-1}$).

This ensures that b_k must only be known up to a common constant multi-

plicative factor, as previously assumed (see Remark 3.19). We formulate the work controlled counterpart of (3.17) in (3.35), assuming that $\chi = \chi_r$

$$\begin{aligned} \omega^*(K, \bar{\omega}, \omega_{M^-}, \omega_{M^+}) \in \arg \min_{\omega_{M^+} \leq \omega \leq \omega_{M^-}} & \sum_{k=0}^{K-1} a_k \chi^{(-1)}\left(\frac{\omega_k}{b_k}\right) \\ \text{s.t.} & \sum_{k=0}^{K-1} \omega_k = \bar{\omega}. \end{aligned} \quad (3.35)$$

Let us state the analogous version of Corollary 3.20 regarding the work controlled framework.

Corollary 3.23. *Let Assumptions (A) and (B) hold with $(a, b) \in \mathbb{R}_{++}^{K \times 2}$, $\omega_{M^+} < \bar{\omega}/K < \omega_{M^-}$ and $\chi = \chi_r : \mathbb{R}_+ \rightarrow \mathbb{R}_+$. If $\omega_{M^+} < \dot{\omega} < \omega_{M^-}$ with*

$$\dot{\omega} = \bar{\omega} \cdot \frac{(b \odot a^r)^{\frac{1}{r+1}}}{\sum_{k=0}^{K-1} (b_k a_k^r)^{\frac{1}{(r+1)}}} \quad (3.36)$$

then $\dot{\omega}$ is optimal for (3.35).

Remark 3.24. Despite being very similar, problems (3.17) and (3.35) are not perfectly equivalent in general. They do share the common goal of minimizing $\mathcal{E}$ subject to a budget constraint $\mathcal{B}$. Nevertheless, one can see (3.35) as a version of (3.17) where the bounds on the achievable δ_k at iteration $k \in \{0, \dots, K-1\}$ vary. Indeed, we have the following bounds:

$$\delta_k \in \left[\mathcal{B}_k^{(-1)}(\omega_{M^-}), \mathcal{B}_k^{(-1)}(\omega_{M^+}) \right]. \quad (3.37)$$

However, in the neutral case where $b \propto \mathbf{1}_K$, these bounds are constant and can be written as $[M^- \bar{\delta}, M^+ \bar{\delta}]$ for well-chosen $\bar{\delta} > 0$ and $0 < M^- \leq M^+$.

Interpretation Just as in (3.33), it is not difficult to construe results from Corollary 3.23 by advocating that

$$\omega_k^* \propto \left(b_k a_k^r \right)^{\frac{1}{r+1}}. \quad (3.38)$$

We have already seen that when a_k grows, linked oracle's accuracy δ_k should evolve inversely proportional. As a consequence, the computational cost ω_k rise accordingly. One should pay attention to the role of b_k in (3.38).

3 | Optimal Inexactness Schedules

It seems like a bigger b_k implies a bigger ω_k , which turns out to be true. Meanwhile, a bigger b_k is prone to curb the demand for highly accurate oracles in (3.33) hence intuitively reducing oracle's work ω_k . Yet, there is no contradiction. Combining (3.33) and Assumption (B), we can capture the overall marginal effect of b_k in

$$\omega_k = b_k \chi(\delta_k) \propto b_k \left((b_k/a_k)^{\frac{1}{r+1}} \right)^{-r} = b_k^{\frac{1}{r+1}} a_k^{\frac{r}{r+1}}.$$

Finally, let us assert that b_k 's *weighting impact* diminishes as $r \rightarrow \infty$, ω_k^* becoming asymptotically proportional to the *impact coefficient* a_k . Conversely when $r \rightarrow 0$, the suggested oracle cost somehow flattens. It ultimately depends only on b_k , constant in the absence of further knowledge (see Example 3.4 and 3.6).

Online version

So far, we have assumed that the choice of K was exogenous and well thought. Unfortunately, there is no *one fits all* approach to adequately fix K . Usually, one runs an optimization algorithm and stops it as soon as a relative tolerance, tracked alongside the iterations, is observed. Let δ^* (respectively ω^*) be an optimal solution of (3.17) (respectively (3.35)) for a specified $K \in \mathbb{N}$ and $\chi = \chi_r$. Hidden behind equation (3.30) (respectively (3.36)) and emphasized by (3.33) (respectively (3.38)), one can retrieve a recursion rule linking the optimal inexactness parameters of two distinctive iterations, say $\hat{k}$ and k . Let $\delta_k^* \in]M^-\bar{\delta}, M^+\bar{\delta}[$ then for any $\hat{k} \in \{0, \dots, K-1\}$, one can recover

$$\delta_{\hat{k}}^* = \max \left\{ M^-\bar{\delta}, \min \left\{ M^+\bar{\delta}, \left(\frac{b_{\hat{k}}}{a_{\hat{k}}} \cdot \frac{a_k}{b_k} \right)^{\frac{1}{(r+1)}} \delta_k^* \right\} \right\}. \quad (3.39)$$

Following the same logic, a recursion exists for $\omega_k^* \in]\omega_{M+}, \omega_{M-}[$,

$$\omega_{\hat{k}}^* = \max \left\{ \omega_{M+}, \min \left\{ \omega_{M-}, \left(\frac{b_{\hat{k}} a_{\hat{k}}^r}{b_k a_k^r} \right)^{\frac{1}{(r+1)}} \omega_k^* \right\} \right\}. \quad (3.40)$$

Therefore, in practice, one can adopt a *reference* situation accounting for a lower-bound of K_r iterations then compute its inherent *offline* optimal schedule. For any iteration $\hat{k} \geq K_r$, one can extrapolate using the recursion rules explained right above.

Rationale The extrapolated schedules obtained with (3.39), (3.40) present the advantage to preserve the right ratios δ_k^*/δ_k^* and ω_k^*/ω_k^* as if we knew the number of iterations performed by our Tunable Oracles Method (TOM) when exited. In addition, it does not break our practical assumption of relative knowledge of a_k 's and b_k 's since we only involve ratios wiping out any common multiplicative constant. Finally, such *online* schedule allows for b_k/a_k 's that are not necessarily pre-computed. As mentioned earlier, in various adaptive methods such coefficients are defined by the final result of local line-search techniques [MF22, STG⁺21, DP19]. This feature of our *online* strategy dramatically extends the applicability of our approach, endowing it with local information exploitation, allowing for possible *non-monotonicity* in the parameters δ_k or ω_k used.

We end this subsection with the equivalent of Theorem 3.21, adapted to the current framework of *work-controlled* schedules. For the same reasons, its detailed proof is therefore skipped since redundant with Theorem 3.16's.

Theorem 3.25. Let Assumptions (A) and (B) hold with $(a, b) \in \mathbb{R}_{++}^{K \times 2}$, $\omega_{M^+} < \bar{\omega}/K < \omega_{M^-}$ and $\chi = \chi_{r>0} : \mathbb{R}_+ \rightarrow \mathbb{R}_+$. $\exists K_\oplus, K_\ominus \in \{0, \dots, K-1\}$, $\hat{\lambda} \in \mathbb{R}$ such that $\forall k \in \{0, \dots, K-1\}$

$$\omega_k^* = \begin{cases} \omega_{M^+} & k \in \oplus := \{\tilde{k} \mid \psi(\tilde{k}) < K_\oplus\} \\ \hat{\lambda} (a_k^r b_k)^{\frac{1}{r+1}} & k \in \mathcal{T} := \{\tilde{k} \mid K_\oplus \leq \psi(\tilde{k}) \leq K-1-K_\ominus\} \\ \omega_{M^-} & k \in \ominus := \{\tilde{k} \mid \psi(\tilde{k}) > K-1-K_\ominus\} \end{cases} \quad (3.41)$$

where $\psi(k)$ depicts the descending rank of $v_k = (a_k^r b_k)^{-1}$, $\hat{\lambda} > 0$ is defined as

$$\hat{\lambda} = \left[\frac{\bar{\omega} - K_\oplus \omega_{M^+} - K_\ominus \omega_{M^-}}{\sum_{k \in \mathcal{T}} (a_k^r b_k)^{\frac{1}{r+1}}} \right] \quad (3.42)$$

and ω^* stands as a solution of (3.35).

3.4 Numerical Experiments

We present three experiments that showcase both our elaborated *offline* and *online* techniques for Tunable Oracle Methods (TOM). We emphasize that

we are primarily concerned with showing that our approach effectively improves the efficiency of known existing methods such as FGM [STG⁺21]. The first experiment serves to validate our theoretical optimal *offline* schedules for various levels of parameters, i.e. r (oracle cost parameter), $\bar{\delta}$ (*reference* inexactness) and K (number of performed iterations). In order to stick to our theoretical framework as much as possible, we generate inexact oracle outputs by adding artificial noise to the gradients used in FGM. The noise is chosen according to a simulated oracle cost one would invest to control the level of inexactness. Confirming that our optimal schedules also perform better in a practical setting is important, as numerical optimization methods (with or without inexactness) typically perform better than their worst-case guarantees in practice. Within this setting, we show the superiority of our schedules compared to a *constant* schedule approach with matching overall computational cost. In the second and third experiments, we investigate how the *offline* and our heuristic *online* approach behave within a real case where inexactness naturally emerges. Experiments 1,2 and 3 rely on the problem motivated in Example 3.6 but Experiment 3 allows for line-search within FGM hence turning it into an *online* method. Our code is freely available on [GitHub](#) so that one can observe that the reported results are representative of the usual performances.

Test Problem & Data Generation

We describe a *robust optimization* task involving a regularization parameter $\mu \geq 0$ and data vectors $\Theta = \{\vartheta_i\}_{i=1}^n$,

$$\vartheta_i \sim \text{Nor}(\mathbf{0}_d, \mathbf{I}_d/P) \quad \forall i \in \{1, \dots, n\}$$

a collection of $n \in \mathbb{N}$ *scenarios* from $\mathbb{R}^d$ ($P > 0$). We consider the classical problem where one minimizes the worst outcome of the regularized linear objective only over the previously seen *scenarios*, i.e. one wishes to minimize a regularized objective $\frac{\mu}{2}\|x\|^2 + \langle \vartheta, x \rangle$ over the unit simplex, i.e. $\Delta^d := \{x \in \mathbb{R}^d \mid x \geq \mathbf{0}, \|x\|_1 = 1\}$ for a meaningful cost ϑ based on historical data about its value, stored in $\mathbf{O} = [\vartheta_1, \dots, \vartheta_n]^T$,

$$\min_{x \in \Delta^d} \max_{\vartheta \in \Theta} \langle \vartheta, x \rangle + \frac{\mu}{2}\|x\|^2 = \min_{x \in \Delta^d} \max_{\vartheta \in \text{conv}(\Theta)} \langle \vartheta, x \rangle + \frac{\mu}{2}\|x\|^2. \quad (3.43)$$

3.4.1 Experiment 1 | Softmax Optimization under Synthetic Noise

In a large scale setting, solving (3.43) using the standard epigraph reformulation might be prohibitive and is sometimes replaced by a smoothed version of the robust objective, leaving the feasible set intact, i.e. no extra constraint. Let $\tilde{\omega} > 0$, the problem becomes⁴

$$F^* = \min_{x \in \mathbb{R}^d} \underbrace{\tilde{\omega}^{-1} \log \left(n^{-1} \sum_{i=1}^n \exp(\tilde{\omega} \cdot \langle \vartheta_i, x \rangle) \right)}_{f(x)} + \underbrace{\frac{\mu}{2} \|x\|^2 + \mathbb{I}_{\Delta^d}(x)}_{\Psi(x)}. \quad (3.44)$$

Solving (3.44) using FGM requires the ability to compute (in-)exact information about f at any query point $x \in \mathcal{X} = \Delta^d$, i.e. $(\tilde{f}(x), \nabla \tilde{f}(x)) \simeq (f(x), \nabla f(x))$. Here, the value of $f(x)$ and its gradient $\nabla f(x)$ are available in closed-form for (3.44) hence discarding the need for schedules of inexactness. However, one could think about a game where one has to pay δ^{-r} ($r > 0$) or $-\log(\delta)$ ($r = 0$) to obtain $\nabla f(x)$ corrupted by a noise of limited radius $R\delta$ depending on parameter $r \geq 0$. I.e. one gets $\nabla \tilde{f}(x)$ such that

$$\nabla \tilde{f}(x) = \nabla f(x) + R\delta \cdot u, \quad (3.45)$$

with $u \sim \text{Uni}(\mathbb{B}_2(\mathbf{0}_d; 1))$. From [DGN13], one deduces $(f(x), \nabla \tilde{f}(x))$ as defined above provides a $(4R\delta, \tilde{\omega} \cdot \|\mathbf{O}\|^2 + \mu, \mu)$ inexact information.

Test instances As suggested by Example 3.6, let us now try out FGM on problem (3.44) with fixed stepsize $L^{-1} = (\tilde{\omega} \cdot \|\mathbf{O}\|^2 + \mu)^{-1}$ and different values of μ , r , K , $\bar{\delta}$ while using sequences of inexact information tuples parametrized by schedules $\{\delta_k\}_{k=0}^{K-1}$. For each instance, $r \in \{0, 1\}$ and

$$\begin{aligned} (\mu/L, K, \bar{\delta}) &\in \{0\} \times \{10, 500, 1000, 5000, 10000\} \times \{10^{-1}, 10^{-3}, 10^{-5}\} \\ &\cup \{1.5 \cdot 10^{-5}\} \times \{10, 500, 1000, 5000, 10000\} \times \{9 \cdot 10^{-3}, 10^{-4}, 10^{-6}\}, \end{aligned}$$

we compare our *offline* optimized approach $\{\delta_k = \delta_k^*\}_{k=0}^{K-1}$ where δ^* solves (3.17) with vectors (a, b) specified in Table 3.1, $M^- = 0$ and $M^+ = 100$ (we allow the inexactness from our *tunable* approach to be a hundred times worse than the *reference* $\bar{\delta}$) with the *constant* schedule $\{\delta_k = \bar{\delta}\}_{k=0}^{K-1}$. We

⁴Note that the value of (3.44) differs at most by $\epsilon = \mu + n/\tilde{\omega}$ from (3.43).

3 | Optimal Inexactness Schedules

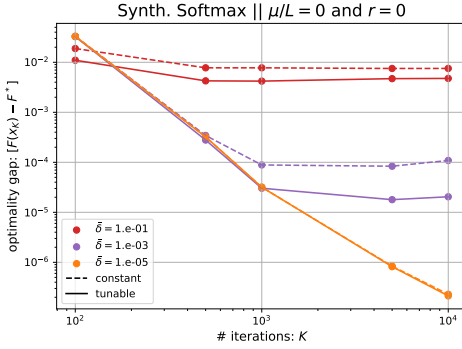
define the simulated cost of inexact oracles as $K/\bar{\delta} = \sum_{k=0}^{K-1} (\delta_k^*)^{-1}$ ($r = 1$) and $-K \log(\bar{\delta}) = -\sum_{k=0}^{K-1} \log(\delta_k^*)$ ($r = 0$)⁵ and match it to the total cost of the constant $\bar{\delta}$ reference schedule. Finally, in order to tame the effects of the randomness inherent to our synthetic inexact information production, we aggregate the obtained results from FGM over 5 runs with 5 different starting points drawn as $x_0 \sim \text{Uni}(\Delta^d)$. Here below, the dimensions read $(d, n, R, P) = (100, 500, 100, 10)$.

Estimation of F^* Regarding this simple example, F^* is estimated by the value of F at the output from a noise-free FGM applied for $3 \cdot 10^5$ iterations.

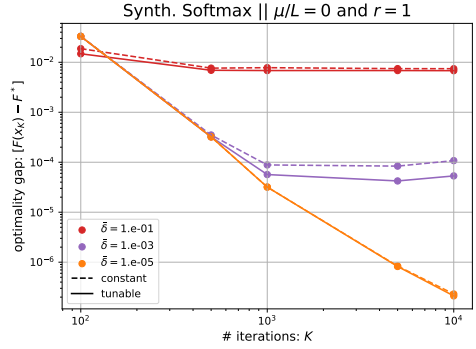
Results Figure 3.4 and 3.5 show the evolution of the terminal primal optimality gap $\mathcal{F}(x_K) - F^* = F(x_K) - F^* = f(x_K) - F^*$ after K iterations of FGM (output iterate x_K is in Δ^d) using the variants of inexactness schedules described above. Each dot plotted on these graphs represents the averaged performance over the $25 = 5 \cdot 5$ independent runs of either our *tunable* or the *constant* approach for a single instance $(\mu, r, K, \bar{\delta})$. As predicted by the theory, the *tunable* inexactness schedule reached a better terminal primal optimality gap for every setting tried. Not surprisingly, the gains were dramatically more impressive when the discrepancy within *impact coefficients* $\{a_k\}_{k=0}^{K-1}$ was high. Intuitively, when such coefficients do not vary at all, our optimized *tunable* approach boils down to the *constant* schedule and no gain is to expect at all. Table 3.1 recalls that $a_k = A_{k+1}$ for any $k \in \{0, \dots, K-1\}$ and Remark 3.22 suggests that $\{A_k\}_{k \in \mathbb{N}}$ grows sublinearly (respectively linearly) when $\mu = 0$ (respectively $\mu > 0$). Therefore, settings in which $\mu > 0$ are the most promising in terms of gains in favor of the *tunable* approach, thanks to the inherent high variance of *impact coefficients*. Figure 3.5 undoubtedly confirmed this hope with substantial gains for our *tunable* approach although problem's conditioning stays relatively small, i.e. $\mu/L \simeq 10^{-5}$. Both Figure 3.4 and 3.5 show that the benefits of high discrepancy in *impact coefficients* (hence in optimal inexactness schedules, see (3.33)) are enhanced by cheap oracles, i.e. $r \rightarrow 0$. We point out as a generic comment that for a small number of iterations, the additive effect of inexactness does not dominate the *error-free* convergence bound and one can barely observe any difference between our optimized approach and other inexactness schedules. Finally, one can observe that some of the results linked to more accurate *reference* inexactness of $\bar{\delta} \leq 10^{-5}$ led to poorer

⁵ $M^+ \bar{\delta} < 1$ in order to ensure that any iteration costs a strictly positive amount of work.

primal optimality gaps for $K = 10^2$. This unexpected phenomenon can be explained by the fact that for bigger values of δ , (3.45) might actually yield more aggressive and successful descent directions, compared to plain accurate gradients, again in a regime where the additive impact of inexactness is of secondary importance.

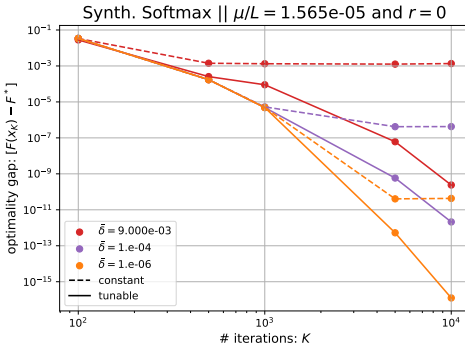


(a) logarithmic oracle cost ($r = 0$)

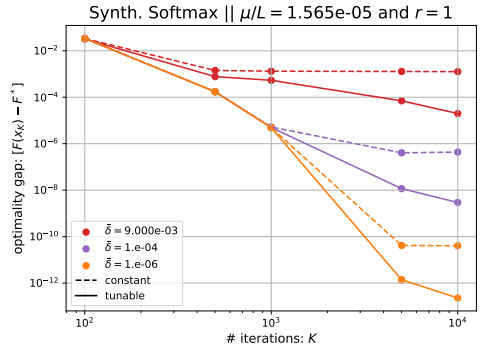


(b) inversely proportional oracle cost ($r = 1$)

Fig. 3.4 (no regularization, $\mu = 0$) | Softmax with Synthetic Noise



(a) logarithmic oracle cost ($r = 0$)



(b) inversely proportional oracle cost ($r = 1$)

Fig. 3.5 (regularization, $\mu > 0$) | Softmax with Synthetic Noise

3.4.2 Experiment 2 & 3 | Robust Optimization over Convex Hull

As depicted in Example 3.6, one may choose an *anchor scenario* $\bar{\vartheta} \in \mathbf{conv}(\Theta)$ and a parameter $\omega > 0$ in order to regularize the inner maximization problem in (3.43) (right-hand side) yielding a smoothed outer level objective [Nes05]. Then, a challenging task consists of minimizing the worst possible outcome of the inner-regularized objective over the convex hull of all the *scenarios*, i.e.

$$\min_{x \in \Delta^d} \max_{\vartheta \in \mathbf{conv}(\Theta)} \langle \vartheta, x \rangle - \frac{\omega}{2} \|\vartheta - \bar{\vartheta}\|^2 + \frac{\mu}{2} \|x\|^2. \quad (3.46)$$

To that end, we adopt the *anchor* $\bar{\vartheta} = n^{-1} \sum_{i=1}^n \vartheta_i$ and construct

$$F^* = \min_{x \in \mathbb{R}^d} \underbrace{\frac{\mu}{2} \|x\|^2 + \max_{q \in \Delta^n} \underbrace{\langle \mathbf{O}^T q, x \rangle - \frac{\omega}{2} \|\mathbf{O}^T q - \bar{\vartheta}\|^2}_{\check{f}(q; x)}}_{f(x)} + \underbrace{\mathbb{I}_{\Delta^d}(x)}_{\Psi(x)}. \quad (3.47)$$

Again, as motivated in Example 3.6, we use FISTA from [Van22] with a momentum depending on the number $\hat{\varphi} = \frac{\lambda_{\min}(\mathbf{O}\mathbf{O}^T)}{\lambda_{\max}(\mathbf{O}\mathbf{O}^T)}$ in order to get $q_x \geq \mathbf{0}_n$, $\|q_x\|_1 = 1$ ensuring

$$f(x) - \left(\frac{\mu}{2} \|x\|^2 + \check{f}(q_x; x) \right) \leq \delta. \quad (3.48)$$

Then $(\check{f}(x), \nabla \check{f}(x)) = (\frac{\mu}{2} \|x\|^2 + \check{f}(q_x; x), \mu x + \mathbf{O}^T q_x)$ provides $(2\delta, 2 \cdot \omega^{-1} + \mu, \mu)$ inexact information about f , tuple used within FGM to solve problem (3.46). We monitor the quality of a candidate $q_x^{(\omega)}$ at any point $x \in \Delta^d$ after ω inner-iterations thanks to the Frank-Wolfe gap $\text{FW}(\omega)$, which is defined as follows

$$\text{FW}(\omega) = \max_{k_{\text{in}} \in [\omega]} \max_{q \in \Delta^n} \check{f}(q_x^{(k_{\text{in}})}; x) + \langle \nabla \check{f}(q_x^{(k_{\text{in}})}; x), q - q_x^{(k_{\text{in}})} \rangle - \check{f}(q_x^{(\omega)}; x). \quad (3.49)$$

As soon as the Frank-Wolfe gap satisfies $\text{FW}(\omega) \leq \delta$ we have that the quality $q_x^{(\omega)}$ fulfills (3.48). In order to speed-up computations to obtain approximate solutions of the inner problem, we give as starting point of FISTA the solution obtained at the last oracle call.

Estimation of F^* Let $\hat{x}^* \in \Delta^d$ be the best known solution to (3.47). Then, by convexity of the objective, we estimate a lower-bound for F^* as follows

$$\begin{aligned} F^* &= \min_{x \in \Delta^d} f(x) \\ &\geq \min_{x \in \Delta^d} \underbrace{\frac{\mu}{2} \|\hat{x}^*\|^2 + \check{f}(q_{\hat{x}^*}; \hat{x}^*)}_{\check{f}(\hat{x}^*)} + \underbrace{\langle \mu \hat{x}^* + \mathbf{O}^T q_{\hat{x}^*}, x - \hat{x}^* \rangle}_{\nabla \check{f}(\hat{x}^*)} + \frac{\mu}{2} \|x - \hat{x}^*\|^2, \end{aligned} \quad (3.50)$$

where $q_{\hat{x}^*}$ represents the q candidate that achieved 10^{-10} precision when computing an approximate pair $(\check{f}(\hat{x}^*), \nabla \check{f}(\hat{x}^*)) \simeq (f(\hat{x}^*), \nabla f(\hat{x}^*))$ with respect to criterion (3.48).

Offline Schedules (Experiment 2)

Test instances Here, we run FGM with stepsize $L^{-1} = (2\omega^{-1} + \mu)^{-1}$ for a predefined number of iterations K . We have tested the two following settings in terms of scaling and dimension; $(d, n, P, \omega) = (100, 500, 5^{-1}, 10^{-3})$ and $(d, n, P, \omega) = (1000, 500, 5^{-1}, 10^{-3})$. The first setting led to $\hat{\varphi} = 0$ hence implying that the oracle cost parameter r equals $\frac{1}{2}$ whereas under the second setting $\hat{\varphi} > 0$ and r is taken as 0, recalling arguments of Example 3.6. For each instance,

$$\begin{aligned} (r, K, \bar{\delta}) &\in \{1/2\} \times \{10, 500, 1000, 5000\} \times \{10^{-1}, 10^{-3}, 10^{-5}\} \\ &\cup \{0\} \times \{10, 500, 1000, 5000\} \times \{9 \cdot 10^{-3}, 10^{-4}, 10^{-6}\}, \end{aligned}$$

with $\mu \in \{0, 10^{-1}\}$ and we obtain our *tunable* δ^* by solving problem (3.17) in the very same fashion as Experiment 1. Although the overall computational cost of our *tunable* $\{\delta_k = \delta_k^*\}_{k=0}^{K-1}$ schedule is meant to coincide with the computational cost of the *constant* schedule $\{\delta_k = \bar{\delta}\}_{k=0}^{K-1}$, the oracle cost model does not always perfectly fit the true encountered computational costs required to produce the inexact information tuples. Therefore, we record the total number of inner-iterations $\mathcal{B}_K = \sum_{k=0}^{K-1} \omega_k$ as well as the final primal accuracy about (3.47) for fair comparisons. This latter entails the computation of $f(x_K) = F(x_K)$, also performed at 10^{-10} precision with respect to criterion (3.48) (see above). Here, $\hat{x}^*$ stems as the output of FGM for $8 \cdot 10^3$ iterations with *constant* inexactness level of 10^{-8} when $r = 0$. Because of much more expensive oracles, we execute only $5 \cdot 10^3$ iterations at inexactness level 10^{-7} to produce $\hat{x}^*$ when $r = \frac{1}{2}$. To reduce the initialization bias $x_0 \sim \text{Uni}(\Delta^d)$, we aggregated results of 10 repetitions.

Results Here, we have plotted the terminal primal optimality gaps against the overall oracle workloads $\sum_{k=0}^{K-1} \omega_k$ after K (outer-)iterations of FGM. Since the wall-clock times strongly correlate with the workloads, we have chosen not to display duplicate figures other than Figure 3.6 and 3.7. Again, each dot symbolizes the averaged performances tracked for one specific inexactness schedule for one specific instance defined by the vector $(\mu, r, K, \bar{\delta})$. One can observe that most of the trends highlighted in the paragraph about the results of our synthetic first experiment still hold for this real second experiment. Notably, it looks like that regularization plays a key role in the substantial superiority of our optimized *tunable* approach. Again, this is understood by a variability of possibly several orders of magnitude in the value of *impact coefficients* $\{a_k\}_{k=0}^{K-1}$ when $\mu > 0$. In every case, our *tunable* approach was more efficient, i.e. it dominated the *constant* approach in the Pareto sense in the plane of reached primal optimality versus unit of workload spent in the inner problems. On Figure 3.6 (a) and 3.7 (a), the *tunable* schedule based on a *reference* inexactness $\bar{\delta}$ fixed to 10^{-6} (yellow) did take more inner-iterations for $K = 500$ than $K = 1000$. Although not intuitive, this behaviour can be explained. It happens that oracles cost way less than expected, especially for low requested accuracies. In what concerns Experiment 2, it could even happen that FISTA's starting iterate at outer-iteration k , $q_{x_k}^{(0)} = q_{x_{k-1}}^{(\omega_{k-1})}$, already fulfills δ_k inexactness for free, i.e. $\omega_k = 0$. It turns out that our *tunable* approach for $K = 1000$ benefited more of such *cheap-meal* phenomenons since its linked schedule demands less accurate oracles than the one with $N = 500$ in early iterations. Overall, a *reference* inexactness $\bar{\delta} \in [10^{-3}, 10^{-2}]$ provided the best *trade-off* to reach a given primal accuracy target within this *offline* setting.

Online Schedules (Experiment 3)

Test instances Finally, we run FGM with adaptive stepsizes $L_k^{-1} > 0$ for any $k \geq 0$. A stepsize is validated at iteration k as soon as the inequality

$$\tilde{f}(x_{k+1}) \leq \tilde{f}(y_k) + \langle \nabla \tilde{f}(y_k), x_{k+1} - y_k \rangle + \frac{L_{k+1}}{2} \|x_{k+1} - y_k\|^2 + 2\delta_k \quad (3.51)$$

holds for $(2\delta_k, \omega^{-1} + \mu, \mu)$ inexact information tuples at feasible points x_{k+1} and y_k respectively. FGM increases by $3/2$ the stepsize when an iteration succeeds, i.e. (3.51) passes, and decreases it by a factor 2 otherwise. Note that (3.51) is always satisfied for $L_{k+1} \geq \omega^{-1} + \mu$. Convergence guarantees like (3.15) are conserved [STG⁺21] but now the *impact coefficients* $a_k = A_{k+1}$

depend on L_{k+1} via the recursion ($A_0 = 0$)

$$A_{k+1}(1 + \mu A_k) = L_{k+1}(A_{k+1} - A_k)^2 \quad (3.52)$$

and are therefore not predictable in advance. This paves the way for the application of our *online* approach (3.39) with $K_r = 50$. K_r is kept small in practice, its sole interest being to specify the very first inexactness parameters before the *online* approach takes over. Yet, one must simulate the K_r first coefficients $\{a_k\}_{k=0}^{K_r-1}$ required to devise the *offline* optimized schedule $\{\delta_k^*\}_{k=0}^{K_r-1}$ from (3.17). To that purpose, one fakes the value of L_{k+1} fixing it to $L = \omega^{-1} + \mu$ within (3.52) and uses the computed $\{a_k = A_{k+1}\}_{k=0}^{K_r-1}$. For each level of *reference* inexactness $\bar{\delta} \in \{9 \cdot 10^{-3}, 10^{-4}, 10^{-6}\}$, we tried 4 different inexactness schedules. I.e. for any iteration index $k \geq 0$, the *constant* one $\delta_k = \bar{\delta}$, the heuristic *online tunable* (as described above) with $a_k = A_{k+1}$, the fully sublinear monotonically decreasing tuned as in [VSBV13] (*poly-3*), $\delta_k = \bar{\delta}(k+1)^{-3}$ and the fully linear one [MF22] (*linear*) with $\delta_k = \bar{\delta}(1 - \sqrt{\mu/L})^{-k}$. For this experiment, $\hat{x}^*$ (see (3.50)) is taken as the output of adaptive FGM after at $3 \cdot 10^3$ iterations with *constant* inexactness of 10^{-12} for $r = 0$ and 10^{-8} otherwise for $r = \frac{1}{2}$. We have conducted 5 random initializations of the settings; $(d, n, P, \omega) = (50, 100, 5^{-1}, 10^{-3})$ which, akin to Experiment 2, led to $\hat{\phi} = 0$ and thus $r = \frac{1}{2}$ and $(d, n, P, \omega) = (200, 100, 5^{-1}, 10^{-3})$ that gave $\hat{\phi} > 0$ and, accordingly, $r = 0$. Finally, we set the regularization to $\mu = 10^{-1}$.

Results We decided to keep track of the primal objective value every 10 outer-iteration, i.e. when $k \bmod 10 = 0$, we did compute $\tilde{f}(x_k) \simeq f(x_k)$ with precision 10^{-10} . After the runs, we turned these primal objective values into primal optimality gaps thanks to our estimation of F^* . We can observe on Figure 3.8 (top to bottom) that our *online* heuristic *tunable* inexactness schedule adapts well to the *error-free* speed of convergence of FGM, symbolized by the value of the main sequence $\{A_k\}_{k \in \mathbb{N}}$. We recall that its starting growth is of order $\mathcal{O}(k^2)$ but eventually, since $\mu > 0$ in the present experiment, a linear growth rate shows up lately [STG⁺21, MF22]. It is worth noticing that (*linear*) schedule was the best in the low *reference* inexactness regime. Nevertheless, overall, neither (*poly-3*) or (*linear*) adapts as well as our schedule. Indeed, although comparable to *tunable* when $r = \frac{1}{2}$ on Figure 3.8 (right) (top to bottom), *poly-3* does not take advantage of the regularization whereas *linear* appears way too conservative on Figure 3.8 (middle and bottom), perhaps in order to preserve as much

3 | Optimal Inexactness Schedules

as possible the asymptotical rate of convergence of the *error-free* counterpart of FGM. The reference *constant* schedule for inexactness turns out to be competitive until *error-accumulation* (see Remark 3.2) undermines further improvements.

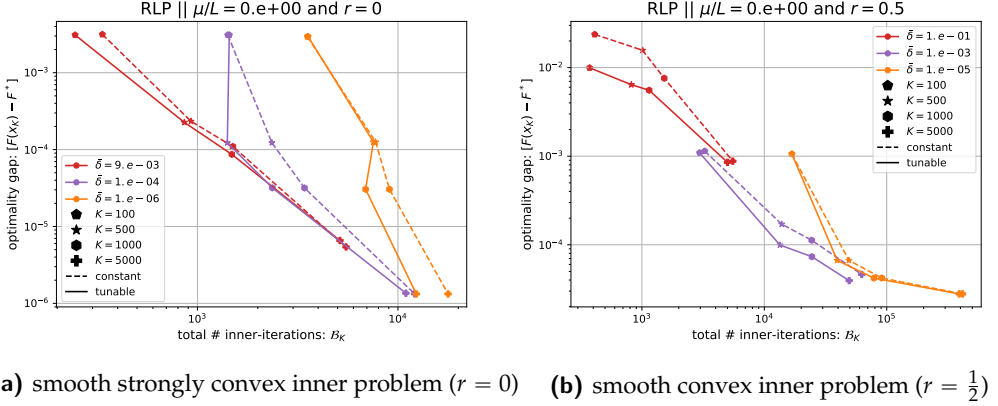


Fig. 3.6 (no regularization, $\mu = 0$) | R.O. *offline* with multiple $\bar{\delta}$

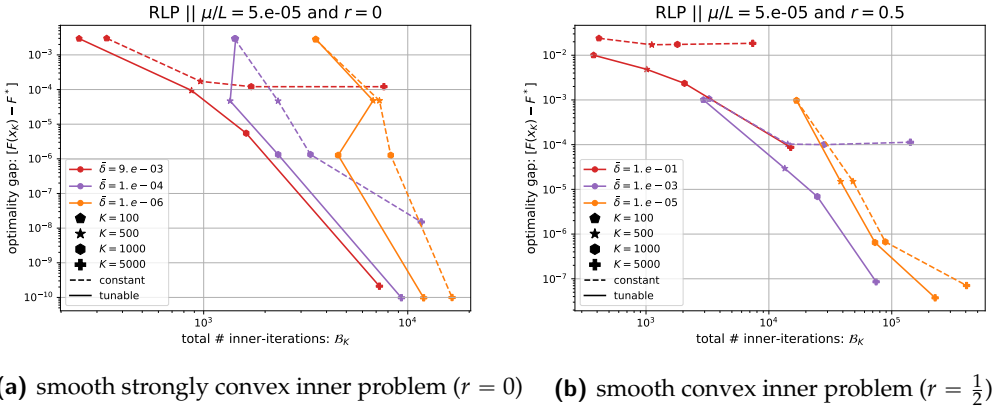


Fig. 3.7 (regularization, $\mu > 0$) | R.O. *offline* with multiple $\bar{\delta}$

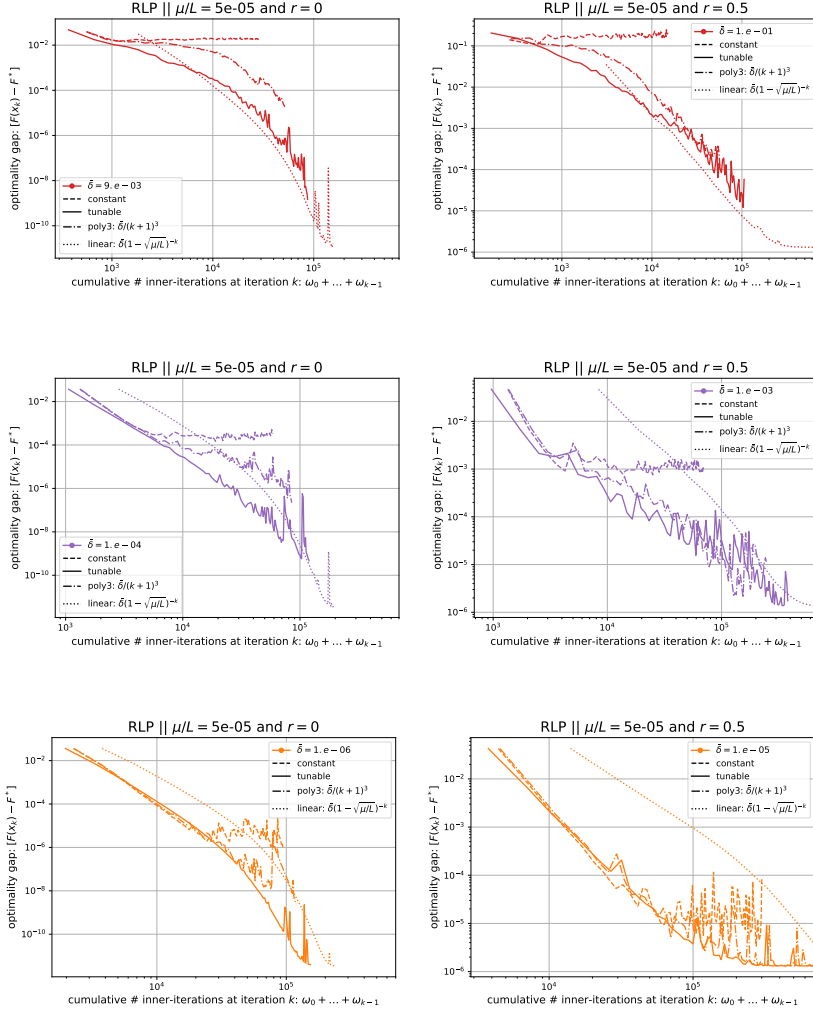


Fig. 3.8 (regularization, $\mu > 0$) | R.O. *online* with low then medium then high accuracy $\bar{\delta}$, smooth strongly convex inner problem $r = 0$ (left) and smooth convex inner problem $r = 1$ (right)

3.5 Conclusion

In this chapter we considered a class of iterative algorithms, namely Tunable Oracle Methods (TOM), for which one can take advantage from the combined knowledge of both the computational cost associated with the oracle calls and the impact of associated inexactness on the convergence. We have shown how to choose optimally the level of inexactness that should be requested at each iteration. Our numerical experiments confirm the superiority of these optimal schedules over the use of constant inexactness, for a given total computational budget, and also compare favorably with existing baselines from the literature. Future work may include a *tight analysis* of more iterative methods that involve controllable inexactness, e.g. bilevel learning, for which we can also hope that optimal inexactness schedules enhance the practical performances. Another direction for future research would be to gain insight about *random inexactness* rather than the (possibly) adversarial model considered in this chapter. In this context, the goal could be to choose the appropriate amount of work to drive the distribution of inexactness towards the local needs of the iterative algorithm. Of course, there would exist a clear connection with previous works in stochastic optimization wherein using *mini-batches* involving more *sample gradients* (hence computational work) provides a more accurate gradient estimate. Yet, other techniques including *biased* stochastic gradients of tunable quality might be further explored.

PART II
Difference-of-Convex (DC)
Methods

4

Guided Tour of (DC) Structure

4.1 Intuition

As previously teased, the second part of this thesis is dedicated to the analysis and the creation of new optimization methods that tackle problems falling outside the usual convex comfort zone. That is, we aim at identifying structured objective functions F in (1.1) that relax convexity requirements, yet allowing to reuse (e.g. in a sequential manner) most of the efficient optimization tools developed for convex optimization. Moreover, an appreciable feature of these *close-to* convex objectives could be that, indeed, a degenerated case might turn out be convex. Interestingly, for some of these structured nonconvex objectives, the problem of globally minimizing them amounts to solve a finite number of (tractable) convex problems. In a sense, this allows to quantify how hard these nonconvex problems are. Ultimately, the easiest would then be a convex problem.

Recalling generic notations of (3.7), we propose now to study problems where $f = f_1 - f_2$ matches the aforementioned description.

$$F^* = \min_{x \in \mathbb{R}^d} \left\{ \mathcal{F}(x) := f_1(x) - f_2(x) + \Psi(x) \right\} > -\infty. \quad (\nu_{\text{DC}})$$

4 | Guided Tour of (DC) Structure

Throughout the remaining chapters of this thesis, we denote:

- $\Psi : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{\infty\}$ proper closed convex function (not necessarily smooth), i.e. $\text{dom } \Psi \neq \emptyset$ and Ψ is lower semi-continuous.
- $f_1 : \mathbb{R}^d \rightarrow \mathbb{R}$ and $f_2 : \mathbb{R}^d \rightarrow \mathbb{R}$ proper and convex, i.e.

$$\max\{|f_1(x)|, |f_2(x)|\} < \infty \quad \forall x \in \mathbb{R}^d \supseteq \text{dom } \Psi.$$

The objective F is described as the difference between two convex functions $\Psi + f_1$ and f_2 , or equivalently, the sum of a *convex component* $\Psi + f_1$ and a *concave component* $-f_2$. Thereby, problems like (v_{DC}) are referred to as difference-of-convex (DC). Although nonconvex and intrinsically hard, the (DC) framework allows for successive global approximations either convex or nonconvex but very well structured, that are easily solvable.

Remark 4.1. It comes that if $x \rightarrow f_2(x) = 0$ for every $x \in \mathbb{R}^d$ (null application) then F becomes convex. The nonconvexity of F solely depends on where in the ambient space f_2 brings *enough* concavity to destroy the convexity brought by $f_1 + \Psi$. Let $x \rightarrow \Psi(x) = 0$ with f_1 and f_2 twice continuously differentiable. Then one can detect nonconvexity at *near* $x \in \mathbb{R}^d$, if there exists $v \in \mathbb{B}_2(\mathbf{0}_d; 1)$ such that

$$\langle v, \nabla^2 f_1(x) v \rangle < \langle v, \nabla^2 f_2(x) v \rangle.$$

Nonconvexity arises when the curvature induced by f_2 leads. There might be large portions of $\text{dom } F$ where F is actually convex under the hood.

At the light of the above, we see that (v_{DC}) suits well as straightforward relaxation of a convex problem. That being stated, (DC) turns out to be quite general ! Every lower- $\mathcal{C}^2$ function, every finite maxima of smooth functions and every polynomial is (DC) representable (see [dO20]). Nevertheless, when existing, the (DC) representation of a function is **never** unique.

Proposition 4.2 (Convexity shift). Let $f = f_1 - f_2$ be a (DC) function fulfilling the assumptions stated above where f_1 (respectively f_2) is μ_1 -strongly convex (respectively μ_2 -strongly convex). Then, for every $\mu \geq -\min\{\mu_1, \mu_2\}$,

$$f = \underbrace{f_1 + \frac{\mu}{2} \|\cdot\|^2}_{f_{1,\mu}} - \underbrace{\left(f_2 + \frac{\mu}{2} \|\cdot\|^2\right)}_{f_{2,\mu}} \quad (4.1)$$

where $f_{1,\mu}$ and $f_{2,\mu}$ are both still convex.

4.2 Diversity

The goal of this section is to give a first glimpse of (DC) programs broad scope and, possibly, the inherent associated difficulty. Thanks to the first example we highlight some intrinsic limits of the (DC) framework and the usual strive for (DC-criticality) therein thanks to a pathological example crafted by Pr. Y. Nesterov. Then, moving on with the second example, we recall the very well-known *clustering* problem arising in machine-learning. This latter is a structured (DC) program, i.e. an instance of (SMC) that we introduce in Section 4.3.1 and thoroughly study in Chapter 7.

Infamous Let $a \in \mathbb{Z}^d$ and $\varsigma \in]0, 1[$. The goal is to solve

$$F^* = \min_{x \in \mathbb{R}^d} \left\{ \mathcal{F}(x) := (1 - \varsigma) \|x\|_\infty - \min_{e \in [d]} |x_e| + |\langle a, x \rangle| \right\}. \quad (\text{Challenge})$$

$\mathcal{F}$ is a homogenous (DC) function with no Ψ term and

$$f_1(x) = (1 - \varsigma) \|x\|_\infty + \max_{e' \in [d]} \sum_{e=1, e \neq e'}^d |x_e| + |\langle a, x \rangle|, \quad f_2(x) = \sum_{e=1}^d |x_e|.$$

The point $x = \mathbf{0}_d$ is a *critical* one with $F(\mathbf{0}_d) = 0$. However, it is NP-hard to decide whether there exists $x \in \mathbb{R}^d$ (*critical* or not) such that $F(x) < 0$.

Clustering Let $\{\beta^{(s)}\}_{s=1}^N$ denote N vectors from $\mathbb{R}^p$. The *squared L_2 clustering* task with B centroids amounts to solve

$$F^* = \min_{x=(x_1, \dots, x_B) \in (\mathbb{R}^p)^B} \left\{ \mathcal{F}(x) := \frac{1}{N} \sum_{s=1}^N \min_{l \in [B]} \|x_l - \beta^{(s)}\|_2^2 \right\}. \quad (\text{Clustering})$$

The objective $\mathcal{F}$ can be decomposed as

$$f_1(x) = \frac{1}{N} \sum_{s=1}^N \sum_{l=1}^B \|x_l - \beta^{(s)}\|_2^2, \quad f_2(x) = \frac{1}{N} \sum_{s=1}^N \max_{l' \in [B]} \sum_{l=1, l \neq l'}^B \|x_l - \beta^{(s)}\|_2^2.$$

For the record, the celebrated Lloyd-algorithm [Llo82] implements an alternating-minimization scheme that tackles problem (Clustering) under the hat.

4.3 Hierarchy

We aim in this section at underpinning interesting subclasses of (DC) problems exhibiting even more valuable structure. We recall that, as such, (DC) problems involve objectives that are nonsmooth and nonconvex.

In order to ease the reading of the upcoming content, we provide well-known facts about (DC) functions. Aside from $f = f_1 - f_2$, we introduce another of such (DC) functions, i.e. $\tilde{f} := \tilde{f}_1 - \tilde{f}_2$. Here, $\tilde{f}_1$ (respectively $\tilde{f}_2$) is also proper and convex as assumed above for f_1 (respectively f_2).

Proposition 4.3 (Closure properties of (DC), [Jem20] Proposition 3.1 & 3.2). *With $f = f_1 - f_2$, $\tilde{f} = \tilde{f}_1 - \tilde{f}_2$ as previously defined and $C \in \mathbb{R}_+$, it comes that*

$$f \cdot C = C \cdot f = (C \cdot f_1) - (C \cdot f_2) \quad (4.2)$$

$$f_+ = f + \tilde{f} = (f_1 + \tilde{f}_1) - (f_2 + \tilde{f}_2) \quad (4.3)$$

$$f_{\max} = \max\{f, \tilde{f}\} = \max\{f_1 + \tilde{f}_2, \tilde{f}_1 + f_2\} - (f_2 + \tilde{f}_2) \quad (4.4)$$

$$f_{\min} = \min\{f, \tilde{f}\} = (f_1 + \tilde{f}_1) - \max\{f_2 + \tilde{f}_1, \tilde{f}_2 + f_1\} \quad (4.5)$$

are all (DC) functions.

Remark 4.4. Of course, one can iterate multiple times the operations (4.2)-(4.5) over (DC) functions and still end up with a (DC) function at the end.

Our goal in Chapter 6-7 resides in exploiting the additional assumptions that define the two subclasses, i.e. (MC) and (SMC), detailed in Section 4.3.1. We will extensively use Proposition 4.3 and Remark 4.4 to derive the (DC) representation of the objectives involved in these subclasses.

Akin to existing (DC) optimization methods (see e.g. [LZ19, MFU23]), we study in Chapter 5 another (DC) subclass defined by the assumption of f_1 being smooth. In addition, we also present structured nonsmoothness of f_1 (and f_2) within problem (ν_{DC}) as (DC) subclasses (DMsC) and (DSMsC) in Section 4.3.2. We tackle instances (DSMsC) at the end of Chapter 8.

We already mention at this point that most of the (DC) problems under our scope of interest are *data* driven. We will detail situations wherein a dataset of size $N \in \mathbb{N}$ appears explicitly. Thereby, the aforementioned subclasses are subdivided in two cases.

(i) **Exhaustively enumerable.** The objective (or the set of constraints) involves a number of terms depending on (N, d) small enough (e.g. polynomial) so that a scan over all the terms is computationally tractable.

(ii) **Non-Exhaustively enumerable.** The objective involves a number of terms that appears too large for linear time scanning procedures.

4.3.1 Minimum-of-Convex (MC)

Let $n \in \mathbb{N}$ and let $\{f^{(i)}\}_{i \in [n]}$ be a family of continuous convex functions. The pointwise minimum F_{MC} of the n convex functions read as follows

$$F_{\text{MC}} : \mathbb{R}^d \rightarrow \mathbb{R}, \quad x \rightarrow F_{\text{MC}}(x) := \min_{i \in [n]} f^{(i)}(x). \quad (4.6)$$

Proposition 4.5 ((MC) $\subsetneq$ (DC)). F_{MC} is always (DC) representable but there exists a (DC) function that does not result as the pointwise minimum of a finite family of n continuous convex functions.

Proof. It suffices to notice that f_1 and f_2 are convex in (4.7) since convex functions are closed under addition and pointwise maximum operators.

$$x \rightarrow F_{\text{MC}}(x) = \underbrace{\left(\sum_{i \in [n]} f^{(i)}(x) \right)}_{f_1(x)} - \underbrace{\left(\max_{i^* \in [n]} \sum_{i'=1, i' \neq i^*}^n f^{(i')}(x) \right)}_{f_2(x)}. \quad (4.7)$$

Let $\bar{x} \in \mathbb{R}^d$. By definition, we are assured that there exists $\mathcal{A}(\bar{x}) \subseteq [n]$ such that $f^{(i^*)}(\bar{x}) = F_{\text{MC}}(\bar{x})$ for every $i \in \mathcal{A}(\bar{x})$ and by continuity arguments, there must exist a radius $R > 0$ so that on $\mathbb{B}_2(\bar{x}; R)$, F_{MC} and $\min_{i \in \mathcal{A}(\bar{x})} f^{(i^*)}$ coincide (Proposition 7.7). In any case, given the finiteness of $|\mathcal{A}(\bar{x})| \leq n < \infty$ and the compactness of $\mathbb{B}_2(\bar{x}; R)$, there must be compact subregions of $\mathbb{B}_2(\bar{x}; R)$ wherein F_{MC} is expressed solely by a single $i^* \in \mathcal{A}(\bar{x})$. We denote by S one of these subregions. Let us now take

$$x \rightarrow F(x) = -x^2 = -\max_{\bar{x} \in \mathbb{R}} \bar{x}^2 + 2\bar{x} \cdot (x - \bar{x}) = \min_{\bar{x} \in \mathbb{R}} \bar{x}^2 - 2\bar{x} \cdot x.$$

F is clearly (DC) with f_1 being the null application and $f_2(x) = x^2$. Yet, as pointed out, it is represented by an infinite amount of continuous convex (affine) functions. It is concave and everywhere, $\nabla^2 F(x) = -2 \cdot \mathbf{I}_1$. Therefore, on S , it is impossible that $F = f^{(i^*)} = F_{\text{MC}}$. $\square$

4 | Guided Tour of (DC) Structure

As previously stated (cfr. (ii)), it might happen that the number n of terms involved in F_{MC} can be prohibitively large. We expose one such situation that arises a lot in practice. We recall that $N \in \mathbb{N}$ is *data* related and can rapidly scale as $10^2 \rightarrow 10^5$ or even more... For every index $s \in [N]$, we consider $n_s \in \mathbb{N}$ continuous functions $\{h_l^{(s)}\}_{l \in [n_s]}$ called *component functions*. Together with $\bar{h}$ lower semicontinuous, dubbed as the *main function*, they are all assumed to be proper and convex on $\mathbb{R}^d$. In Chapter 7, we are concerned with objectives F_{SMC} (Sum-of-Minimum-of-Convex),

$$\begin{aligned} F_{\text{SMC}} : \mathbb{R}^d \rightarrow \mathbb{R}, \quad x \rightarrow F_{\text{SMC}}(x) &:= \bar{h}(x) + \frac{1}{N} \sum_{s=1}^N \underbrace{\min_{l=1, \dots, n_s} h_l^{(s)}(x)}_{h^{(s)}(x)} \quad (4.8) \\ &= \sum_{s=1}^N \min_{l=1, \dots, n_s} N^{-1} (\bar{h}(x) + h_l^{(s)}(x)). \end{aligned}$$

Remark 4.6. Immediately, since Proposition 4.5 proves that every $h^{(s)}$ is (DC) itself, Proposition 4.3 applied N times consecutively (the first application serving to show that $\bar{h} + N^{-1}h^{(1)}$ is (DC)) allows to conclude that F_{SMC} is (DC) representable. Then, without loss of generality, the function F_{SMC} might be rewritten as

$$x \rightarrow F_{\text{SMC}}(x) = \min_{i \in [n]} \left\{ f^{(i)}(x) := \bar{h}(x) + \frac{1}{N} \sum_{s=1}^N h_{\sigma_s(i)}^{(s)}(x) \right\}. \quad (4.9)$$

where $n = \prod_{s=1}^N n_s$ and $i \in [n]$ encodes a $\sigma(i) = (\sigma_1(i), \dots, \sigma_N(i)) \in \times_{s=1}^N [n_s]$.

$$n_0 := 1, \quad \sigma_s(i) = 1 + ([i / (\prod_{\bar{s}=0}^{s-1} n_{\bar{s}})] - 1) \bmod n_s \quad \forall s \in \{1, \dots, N\}.$$

Conversely, we can retrieve the index $i \in [n]$ from a σ -selection as

$$n_0 := 1, \quad i(\sigma_1, \dots, \sigma_N) = 1 + \sum_{s=1}^N (\sigma_s - 1) \cdot \prod_{\bar{s}=0}^{s-1} n_{\bar{s}}.$$

When $N = 1$, $n = n_1$ and minimizing F_{SMC} might fall in the category (i) if n is small enough. More generally, if $n_s \geq 2$ for every $s \in [N]$, $n \in \Omega(2^N)$. In such case, even for moderate N , n becomes quite large and (ii) occurs.

4.3.2 Difference-of-Max-of-smooth Convex (DMsC)

Finally, we consider the following setting wherein the nonsmoothness of both f_1 and f_2 stems explicitly from $\max$ operators. In Chapter 8 (Section 8.4.1), we will need to solve instances of the present framework. Here, $f_\star$ is the sum for $s = 1$ to $s = N_\star$ of pointwise maxima of $n_{\star,s}$ smooth convex functions, i.e. $\{h_{\star,l}^{(s)}\}_{l=1}^{n_{\star,s}}$ for $\star \in \{1, 2\}$. Then, F_{DSMsC} (Difference-of-Sum-of-Max-of-smooth Convex) objectives read

$$F_{\text{DSMsC}} : \mathbb{R}^d \rightarrow \mathbb{R}, \quad x \rightarrow F_{\text{DSMsC}}(x) := \underbrace{\sum_{s=1}^{N_1} \max_{l=1, \dots, n_{1,s}} h_{1,l}^{(s)}(x)}_{f_1(x)} - \underbrace{\sum_{s=1}^{N_2} \max_{l=1, \dots, n_{2,s}} h_{2,l}^{(s)}(x)}_{f_2(x)}. \quad (4.10)$$

Of course, akin to the relationship linking (SMC) and (MC) in Section 4.3.1, any (DSMsC) function can be rewritten into a form where $N_1 = 1 = N_2$ (noted (DMsC) accordingly) by choosing adequate $n_{1,1}$ and $n_{2,1}$.

Remark 4.7. We emphasize that if $\bar{h}$ (*main function*) is a pointwise maximum of $\bar{n} \in \mathbb{N}$ smooth convex functions and furthermore, $\{h_l^{(s)}\}_{l \in [n_s]}$ (*component functions*) are all smooth convex for every $s \in [N]$ then it turns out that (SMC) is (DSMsC). Indeed, with $N_1 = N + 1$, $n_{1,s} = 1$ for every $s \in [N]$ and $n_{1,N+1} = \bar{n}$ as well as $N_2 = N$ and $n_{2,s} = n_s$ for every $s \in [N]$, one chooses

$$\begin{aligned} x \rightarrow h_{1,1}^{(s)}(x) &= \frac{1}{N} \sum_{l=1}^{n_s} h_l^{(s)}(x) & \forall s \in [N], \\ x \rightarrow h_{1,l}^{(N+1)}(x) &= \bar{h}_l(x) & \forall l \in [\bar{n}], \\ x \rightarrow h_{2,l}^{(s)}(x) &= \frac{1}{N} \sum_{l'=1, l' \neq l}^{n_s} h_{l'}^{(s)}(x) & \forall s \in [N_2], \forall l \in [n_{2,s}]. \end{aligned} \quad (4.11)$$

Proposition 4.8 ((DMsC) $\subsetneq$ (DC)). *There exists a function F that is (DC) yet impossible to write as a (DMsC) function.*

Proof. Let $x \rightarrow F(x) = \|x\|_2 + \|x\|_2^4$. F is convex hence (DC) but neither $\|\cdot\|_2$ nor $\|\cdot\|_2^4$ can be formulated as pointwise maximum of a finite number differentiable convex functions, each of which globally bounded from above by an affine term plus a positively defined quadratic form. $\square$

4 | Guided Tour of (DC) Structure

This remark leads us to draw a *Venn diagram* of (DC) functions with the presented subclasses where $(MC) \equiv (SMC)$ and $(DMsC) \equiv (DSMsC)$ share a non-empty intersection and are both strictly contained in (DC).

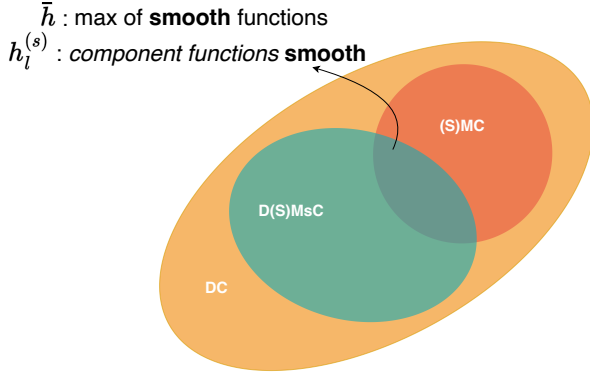


Fig. 4.1 Some subclasses from the (DC) realm.

4.4

Oracles

Excepted in Chapter 6 wherein *global* optimization of F_{MC} is pursued, we will restrict ourselves to find (the best possible) *local* solutions (see Definition 2.9) or, to the least, good *critical points* (see condition DC-criticality). In such case, a well-established approach to deal with problem (ν_{DC}) , i.e. adopted by the continuous optimization community, resides in the framework of *Majorization-Minimization* (MM) (see [Str23] and references therein).

At each iteration of a prescribed (MM) method, one needs to construct an upper-model $\mathcal{F}^+$ of the objective function $\mathcal{F}$ over a well-chosen subset $\mathcal{D} \subseteq \mathbb{R}^d$; this represents the **majorization** phase. Most of the time, this latter model is *tight* at some *stability center* [dO19] (often taken as the current iterate) $\bar{x} \in \text{dom } \mathcal{F}$, i.e.

$$\mathcal{F}^+(u) \geq \mathcal{F}(u) \quad \forall u \in \mathcal{D} \quad \wedge \quad \mathcal{F}^+(\bar{x}) = \mathcal{F}(\bar{x}). \quad (4.12)$$

The **minimization** phase consists then in optimizing $\mathcal{F}^+$ over $\mathcal{D}$ (by default, $\mathcal{D} = \mathbb{R}^d$) and in updating the *stability center* in order to proceed to the next iteration. By executing these two phases repeatedly, under properly stated assumptions, one usually shows the (sub)sequential convergence of a main sequence of iterates towards a *critical point* of the problem (ν_{DC}).

Both these phases define the **computational** (also called **minimization**) oracles involved in the methods we will develop. As in (2.20), we use the notation $\mathcal{F} = f_1 - f_2 + \Psi = F + \mathbb{I}_{\mathcal{X}}$ for some convex subset $\mathcal{X} \subseteq \mathbb{R}^d$. Note that $g_2(\bar{x})$ will be selected among the set $\partial f_2(\bar{x})$ whereas $(\gamma^{(e)}, \beta^{(e)})$ ($e \in [B]$) are built by mixing first-order information about f_1 and f_2 . On Figure 4.2, provided a (SMC) objective $\mathcal{F}$, we show two possible examples of upper-models $\mathcal{F}^+$. This figure suggests that, when available, (SMC) structure and its dedicated oracle (Chapter 7) should be the ones considered.

$\mathcal{F}^+$	notes	chapter
$\max_{e \in [B]} \{\gamma^{(e)} + \langle \beta^{(e)}, u \rangle\} + \hat{L} \ u\ _2^2 + \Psi(u)$	$\hat{L} > 0, B \in \mathbb{N}$	5
$f^{(i)}(u) + \mathbb{I}_{\mathcal{X}}(u)$	$F(\bar{x}) = f^{(i)}(\bar{x})$	6
$\bar{h}(u) + \frac{1}{N} \sum_{s=1}^N \sum_{l \in [n_s]} q_l^{(s)} h_l^{(s)}(u) + \mathbb{I}_{\mathcal{X}}(u)$	$q^{(s)} \in \Delta^{n_s} \forall s \in [N]$	7

Table 4.1 Informal description of **minimization** oracles used in Part II.

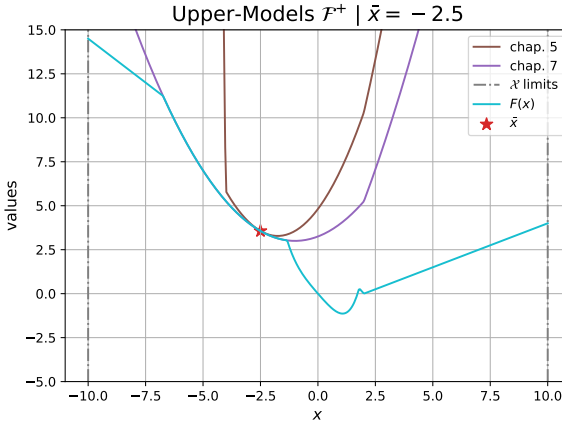


Fig. 4.2 $\mathcal{F}(x) = |x - 2| + \frac{1}{4} \min\{x^4 + \frac{1}{2}(x - 4)^2 - 10, 10, -2x + 4, (x + 3)^2 - 4\}$.

5

Adaptive Sequential (DC) algorithm with extrapolation

5.1 Introduction

During the last decades, many techniques were developed to speed-up the solving of convex optimization problems. Among most successful ones we find Nesterov's *extrapolation*, the *nonmonotone backtracking line-search* and the use of *cutting-plane models*. Each of those relies on exogenous information about the structure at hand. In this chapter, we propose to exploit these techniques all at once to tackle (ν_{DC}) from Chapter 4, i.e.

$$F^* = \min_{x \in \mathbb{R}^d} \left\{ \mathcal{F}(x) := f_1(x) - f_2(x) + \Psi(x) \right\} > -\infty \quad (\nu_{\text{DC}})$$

where $\mathcal{X} = \text{dom } \Psi$. We start by thoroughly detailing these techniques. Then, we provide a sufficient descent lemma tailored for (DC) regularized problems. By solving a sequence of such subproblems, we propose an approach, *Adaptive Sequential (DC) with extrapolation* (ASDCe), involving all the aforementioned features and provably subsequentially converging to a critical point of (ν_{DC}) . Finally, we illustrate the performance of our procedure on two nonconvex problems defined over the L_1 -norm ball, involving quadratic forms and smoothed pointwise maxima of affine terms.

Local smoothness Among the generic (DC) class, without accounting for extra structure like (MC) or (DMsC) (see Chapter 4), there is still a clear distinction to be made when the *convex component* f_1 is smooth (see (QUB)). Indeed, assuming that ∇f_1 is L -Lipschitz continuous on an open set containing $\mathcal{X} \subseteq \mathbb{R}^d$, it is immediately possible to derive an upper-model (see (4.12)) $\mathcal{F}^+$ of the objective $\mathcal{F}$, *tight* at fixed $\bar{x} \in \mathcal{X}$, valid at any $u \in \mathcal{D} = \mathcal{X}$,

$$\begin{aligned} \mathcal{F}(u) &\leq f_1(\bar{x}) - f_2(\bar{x}) + \langle \nabla f_1(\bar{x}) - g_2(\bar{x}), u - \bar{x} \rangle + \frac{L}{2} \|u - \bar{x}\|_2^2 + \Psi(u) \\ &= \gamma(\bar{x}, L) + \langle \beta(\bar{x}, L), u \rangle + \frac{L}{2} \|u\|_2^2 + \Psi(u) \end{aligned} \quad (5.1)$$

where $g_2(\bar{x})$ is any element of $\partial f_2(\bar{x})$ and

$$\gamma(\bar{x}, L) := f_1(\bar{x}) - f_2(\bar{x}) - \langle \nabla f_1(\bar{x}) - g_2(\bar{x}), \bar{x} \rangle + \frac{L}{2} \|\bar{x}\|_2^2 \quad (5.2)$$

$$\beta(\bar{x}, L) := \nabla f_1(\bar{x}) - g_2(\bar{x}) - L\bar{x}. \quad (5.3)$$

This comes at handy since (5.1) matches the **minimization** oracle we consider in the present chapter (see first row of Table 4.1, $B = 1$). It has been long studied that minimizing models like (5.1) to compute a next iterate $x_+ \in \mathcal{X}$ and then setting $\bar{x} = x_+$ and repeating this process all over again asymptotically leads to (DC-criticality) (see e.g. [TFT22] under the Euclidean Bregman kernel and no inertia). Starting from $\bar{x} = x_0 \in \mathcal{X}$, let us denote by $\{x_k\}_{k \in \mathbb{N}}$ the sequence that would be obtained by such procedure. The rate at which the iterates are getting closer from each other scales as

$$\min_{0 \leq k \leq K-1} \|x_{k+1} - x_k\|_2^2 \in \mathcal{O}\left(\frac{L(\mathcal{F}(x_0) - F^*)}{K}\right), \quad (5.4)$$

where $K \in \mathbb{N}$ represents the number of performed iterations.

We highlight that if $x_{k+1} = x_k$ for any $k \in \mathbb{N}$, $\hat{x} = x_k$ must be a *critical point*. One can observe that L plays a direct role in the speed of convergence, i.e. the lower the better. However, one or both of the following issues concerning the (true) underlying global L constant frequently happen.

- This constant is sometimes not known (e.g. too demanding to compute in (very) large empirical risk minimization problems) or, more often, largely overestimated (e.g. based on theoretical properties of a smoothing).
- Although it might be correctly computed from a global perspective, it might appear overly big on substantial portions of $\mathcal{X}$.

Actually, a careful reading of the proof leading to (5.4) shows that one does not really require the L -Lipschitz continuity of ∇f_1 over $\mathcal{X}$. Ultimately, instead of (QUB), it suffices to ensure that for any integer $k \geq 0$,

$$f_1(x_{k+1}) \leq f_1(x_k) + \langle \nabla f_1(x_k), x_{k+1} - x_k \rangle + \frac{L_k}{2} \|x_{k+1} - x_k\|_2^2 \quad (5.5)$$

where $L_k > 0$ and

$$x_{k+1} = \arg \min_{u \in \mathbb{R}^d} \gamma(x_k, L_k) + \langle \beta(x_k, L_k), u \rangle + \frac{L_k}{2} \|u\|_2^2 + \Psi(u). \quad (5.6)$$

Obviously, (5.5) holds for $L_k \geq L$. Nevertheless, depending on the local geometry of f_1 , it could also be satisfied for much smaller values of L_k . Using a nonconstant schedule of (local) smoothness parameters $\{L_k\}_{k \geq 0}$, fulfilling (5.5) at each iteration, allows to make (5.4) more precise as

$$\min_{0 \leq k \leq K-1} \|x_{k+1} - x_k\|_2^2 \in \mathcal{O} \left(\frac{\mathcal{F}(x_0) - F^*}{\sum_{k=0}^{K-1} L_k^{-1}} \right). \quad (5.7)$$

To that end, scholars have developed methods endowed with a (*non*)*monotone backtracking line-search*, i.e. *trial-and-error* adjustment of L_k (increasing or decreasing its estimate on the fly). This accelerates empirically (and theoretically) the convergence since employing small $L_k \leq L$ decreases the expression on the right-hand side of (5.7). This *line-search* circumvents both the possible pitfalls aforementioned. It is worth mentioning that, sometimes, an acceptance rule like (5.5) might be triggered for $L_k > 0$, yet the function

$$u \rightarrow \mathcal{H}_{x_k, L_k}(u) = \gamma(x_k, L_k) + \langle \beta(x_k, L_k), u \rangle + \frac{L_k}{2} \|u\|_2^2 + \Psi(u) \quad (5.8)$$

might fail to represent a global upper-model $\mathcal{F}^+$ of $\mathcal{F}$ on $\mathcal{X}$. Thereby, the methods involving such a *line-search* fall out of the scope of the *Majorization-Minimization* (MM) framework (see Chapter 4) in the strict sense.

Example 5.1. We consider the following scenario to illustrate the above comments. Let $d = 1$, $\Psi(x) = \mathbb{I}_{[-10, 10]}(x)$, $f_2(x) = \max\{-2x - 4, 0, x - 2\}$,

$$f_1(x) = \omega \log \left(B^{-1} \sum_{e=1}^B \exp \left(\frac{\gamma^{(e)} + \langle \beta^{(e)}, x \rangle}{\omega} \right) \right) \quad (5.9)$$

for $\omega = 1/2$, $\beta = (-4, -2, -1/2, 0, 1, 6)$, $\gamma = (0, 5, 5, 4, 1, -20)$, $B = 6$.

5 | Adaptive Sequential (DC) algorithm with extrapolation

It is known (see e.g. [Nes05] or [BT12] and references therein) that f_1 expressed as the log-sum-exp above (5.9) admits a gradient that is L -Lipschitz continuous over $\mathbb{R}$ with $L = \|\beta\|_2^2/\omega$. It is easy to derive that given $\bar{x} \in \mathcal{X} = [-10, 10]$ and $\hat{L} > 0$, one computes the minimizer of $\hat{H}_{\bar{x}, \hat{L}}$ (5.8) as

$$\begin{aligned} x_+ &= \arg \min_{u \in \mathbb{R}} \mathcal{H}_{\bar{x}, \hat{L}}(u) = \mathbf{proj}_{\mathcal{X}} \left(\bar{x} - \hat{L}^{-1} (\nabla f_1(\bar{x}) - g_2(\bar{x})) \right) \\ &= \max \{ -10, \min \{ 10, \bar{x} - \hat{L}^{-1} (\nabla f_1(\bar{x}) - g_2(\bar{x})) \} \}. \end{aligned} \quad (5.10)$$

On Figure 5.1 (a), one observes that with $\hat{L} = L/15$, condition (5.5) is verified, i.e. the green dashed-dot trace (QB1) falls above plain blue one at x_+ . We also observe that the dashed-dot cyan trace is everywhere above the plain cyan one meaning that, in this context, even if $\hat{L} < L$, $\mathcal{H}_{\bar{x}, \hat{L}}$ is a global upper-model ($\mathcal{F}^+$) of $\mathcal{F}$, *tight* at $x = \bar{x} = -3$. Conversely, on Figure 5.1 (b), at $\bar{x} = 5$, condition (5.5) is far from being fulfilled with a local smoothness estimate of $\hat{L} = L/15$. Moreover, for the majority of the interval $[-10, 10]$, the model $\mathcal{H}_{\bar{x}, \hat{L}}$ falls in this case below the curve of $\mathcal{F}(x)$. Thereby, in this context, $\mathcal{H}_{\bar{x}, \hat{L}}$ is **not** an upper-model of $\mathcal{F}$ on $\mathcal{X}$. Nevertheless, we observe that $\mathcal{F}(x_+)$ still managed to decrease (by a tiny margin) the objective value with respect to $\mathcal{F}(\bar{x})$. Actually, as we will showcase in a while, iterate acceptance rules based on *potential function* decrease (e.g. eventually only based on $\mathcal{F}$) can be used instead of (5.5). These rules represent even more relaxed conditions than (5.5) compared to the strong assumption of L -Lipschitz continuity of f_1 's gradient over $\mathcal{X}$.

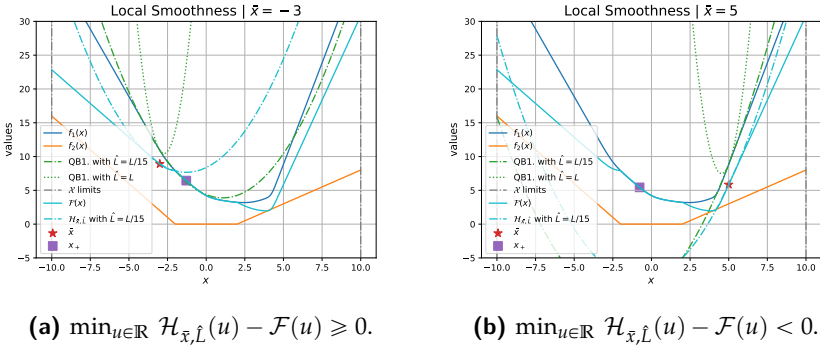


Fig. 5.1 Two different outcomes occurring when *line-searching*.

Nesterov extrapolation Let us step back for a while considering the special case of convex optimization, i.e. $f_2(x) = 0$ for every $x \in \text{dom } \Psi$. It is in this context that both *momentum* and *extrapolation* mechanisms were born to enhance the practical performance of numerical iterative optimization methods. We emphasize that, at first, these methods employed constant stepsizes, i.e. $L_k^{-1} = \hat{L}^{-1}$ was taken for every integer $k \geq 0$, regarding the notations above (5.6). Polyak initiated in 1964 [Pol64] the idea to add *momentum* to tame wild moves from one iterate to the next, an efficient strategy to converge faster especially when iterates get close from the optimum. Slightly different, Nesterov came up in 1983 with its celebrated paper [Nes83] (applicable under our setting for f_1 smooth, $\Psi(x) = 0$ for every $x \in \text{dom } \mathcal{F} = \mathbb{R}^d$) that Beck & Teboulle [BT09] further exploited in 2009, allowing for a more general Ψ . In their works, as soon as two consecutive iterates x_{k-1} and x_k were computed (e.g. as in (5.6)), one extrapolates that the direction $(x_k - x_{k-1})$ that drove x_{k-1} towards x_k (supposedly a better iterate), should be further employed to place the next *stability center*,

$$y_k = x_k + \varsigma_k(x_k - x_{k-1})$$

where $\varsigma_k \in (0, 1)$. That is, instead of minimizing a model of $\mathcal{F} = f_1 + \Psi = f + \Psi$ that will be *tight* at x_k to produce x_{k+1} , one constructs a model based on first-order **information** collected at y_k . The intuition is that y_k represents a promising new iterate in regard of x_k . Starting from $x_0 = x_{-1} \in \text{dom } \Psi$, both [Nes83] ($\Psi = 0$) and [BT09] showed that with carefully crafted *extrapolation* coefficients $\{\varsigma_k\}_{k \in \mathbb{N}}$, alternating the steps

$$\begin{aligned} y_k &= x_k + \varsigma_k(x_k - x_{k-1}) && \text{(Extrapolation)} \\ x_{k+1} &= \arg \min_{u \in \mathbb{R}^d} f(y_k) + \langle \nabla f(y_k), u - y_k \rangle + \frac{\hat{L}}{2} \|u - y_k\|_2^2 + \Psi(u) && \text{(MO)} \end{aligned}$$

with $\hat{L} \geq L$ leads to x_K after $K \in \mathbb{N}$ iterations ensuring

$$\mathcal{F}(x_K) - F^* \in \mathcal{O}\left(\frac{L \|x_0 - x^*\|_2^2}{K^2}\right).$$

The point x^* minimizes the distance to x_0 under the constraint $\mathcal{F}(x^*) = F^*$. As we will see in Section 5.1.1, this substantial improvement in convergence speed, at least in what concerns the sequence of achieved functional values, attracted many researchers to extend the application of *extrapolation* in nonconvex settings, i.e. $f_2 \neq 0$ with respect to our framework.

5 | Adaptive Sequential (DC) algorithm with extrapolation

However, since global optimality is (in general) out of reach when dealing with nonconvex objectives, one can not hope to observe such $\mathcal{O}(1/K^2)$ decay towards F^* . Usually, one compares the rate at which the main iterate sequence divergence converges to 0, as in (5.7). Under the convex setting described above (i.e. L -Lipschitz continuity of ∇f_1 and $f_2 = \mathbf{0}$), it is known [Nes13] that *extrapolation* provably accelerates convergence and one enjoys

$$\min_{k=0,\dots,K-1} \|x_{k+1} - x_k\|_2^2 \in \mathcal{O}\left(\frac{L^2 \|x_0 - x^*\|_2^2}{K^3}\right) \quad (5.11)$$

whereas when $\Psi = \mathbf{0}$ and $f = f_1 - f_2$ is twice continuously differentiable as a whole (possibly nonconvex) with ∇f being L -Lipschitz and $\nabla^2 f$ being C -Lipschitz continuous then [LL22] proved that a fine tuned (and restarted) version of the steps (Extrapolation) & (MO) led to an equivalent rate of

$$\min_{k=0,\dots,K-1} \|x_{k+1} - x_k\|_2^2 \in \mathcal{O}\left(\frac{L^{\frac{4}{7}} C^{\frac{2}{7}} (\mathcal{F}(x_0) - F^*)^{\frac{8}{7}}}{K^{\frac{8}{7}}}\right). \quad (5.12)$$

To the best of our knowledge, until recently (see e.g. [LL22, MT24]), very few papers supported the use of *extrapolation* when tackling nonconvex problems with real theoretical evidence. Under more restrictive assumptions and thanks to a far more involved proof, (5.12) barely improves on the standard (5.4). Notably, one enjoys an enhanced dependency on L and, this time marginally, on the number of performed iterations K . Yet, the additional factor $(C^2 \cdot (\mathcal{F}(x_0) - F^*))^{\frac{1}{7}}$ might mitigate these potential gains. Nevertheless, it is recognized by practitioners to work well in practice (see e.g. [WCP18, TFT22, Yan24]). Within our framework, it is especially true when the positive curvature brought in by f_1 dominates the negative curvature induced by $-f_2$. This will be confirmed by our numerical experiments. We propose now to illustrate how *extrapolation* is usually [WCP18, TFT22] implemented for the particular nonconvex (DC) setting wherein objectives are sometimes not differentiable due to a lack of smoothness about f_2 .

Example 5.2. Mobilizing the same functions (i.e. f_1 , f_2 and Ψ) as in Example 5.1, we display on Figure 5.2 (a) & (b) a scenario wherein *extrapolation* turned out to be quite successful. One proceeds as follows. Starting from an initial iterate $\hat{x}_- = \hat{x}_{--} = -5/2 \in \mathcal{X}$, we computed the *stability center* $y_- = \hat{x}_- + \varsigma_- (\hat{x}_- - \hat{x}_{--}) = \hat{x}_-$ for any $\varsigma_- \in [0, 1]$. Then, to get the iterate $\hat{x}$, we applied accordingly $\hat{x} = \text{prox}_{\hat{L}^{-1}\Psi}(\hat{x}_- - \hat{L}^{-1}(\nabla f_1(\hat{x}_-) - g_2(\hat{x}_-))) = \text{prox}_{\hat{L}^{-1}\Psi}(y_- - \hat{L}^{-1}(\nabla f_1(y_-) - g_2(\hat{x}_-)))$.

On Figure 5.2 (a), we observe that if the *extrapolation parameter* ς was set to $9/10$, $y = \hat{x} + \varsigma(\hat{x} - \hat{x}_-)$ (olive triangle) would achieve a better functional value than $\hat{x}$ (purple square), free of any additional first-order **information** oracle call. Now, on Figure 5.2 (b), we examine the quality of the next iterate $\hat{x}_+$ depending on ς . We can see that fixing the *stability center* at $\hat{x}$ (i.e. **no** extrapolation with $\varsigma = 0$) or y (i.e. extrapolation with $\varsigma = 9/10$) yields two different points $\hat{x}_+$ (purple and olive diamonds respectively). The latter would still be associated with a better objective value but the difference between extrapolation and **no** extrapolation is less pronounced. Under continuity assumptions regarding f_2 's subdifferential set, this difference might vanish as $\{x_k\}_{k \in \mathbb{N}}$ converges.

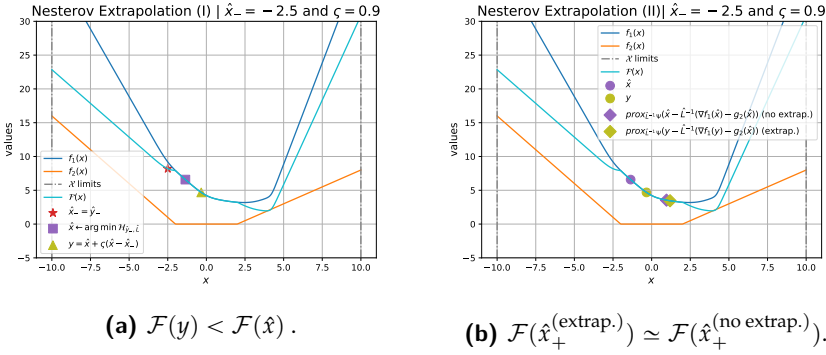


Fig. 5.2 Favorable *extrapolation* in Proximal (DC) methods.

Cutting-planes Let $z_\star \in \mathbb{R}^d$, $g_\star \in \partial f_\star(z_\star)$ and $\tilde{f}_\star = f_\star(z_\star) - \langle g_\star, z_\star \rangle$ for $\star \in \{1, 2\}$. So far, the update steps producing new iterates $x_+ \in \text{dom } \Psi$ involved simple models of $f_\star$,

$$f_\star \simeq \tilde{f}_\star + \langle g_\star, \cdot \rangle.$$

For instance in (5.6), we count a single linearization of f_1 and a single linearization of f_2 around well thought $z_1 = x_k$ and $z_2 = x_k$ respectively. The quadratic term serves as implicit *trust-region* radius. That is, with $\hat{L} = L_k$ big enough, $x_+ = x_{k+1}$ would end up not too far away from z_1 and/or z_2 so that these simple models still approximate $f_\star$ well enough to guarantee improvements regarding the optimization of the objective $\mathcal{F}$. Back in the 70s, initiated (among others) by [Lem74], optimization methods dedicated to the nonsmooth convex case ($f_2 = 0$ and f_1 not necessarily smooth) exploited the convexity and the continuity of functions like $f_\star$ to the fullest.

5 | Adaptive Sequential (DC) algorithm with extrapolation

One can naturally approximate $f_\star$ with polyhedral models, named *cutting-planes* models. These *cutting-planes* models better express the behaviour of $f_\star$ further away from $z_\star$. By iteratively minimizing a strengthened model of the objective, required $\hat{L}$ values to pass iterations tend to diminish and promote, as seen in the first paragraph, faster convergence. Let us now briefly elaborate on these models.

Definition 5.3 (Bundle). A collection $\mathcal{Y} = \{(\tilde{f}^{(e)}, g^{(e)}, z^{(e)}) \in \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}^d\}_{e=1}^B$ is a *bundle* for $f : \mathbb{R}^d \rightarrow \bar{\mathbb{R}}$ on $S \subseteq \mathbb{R}^d$ if for every $e \in [B]$,

$$f(u) \geq f(z^{(e)}) + \langle g^{(e)}, u - z^{(e)} \rangle \geq \tilde{f}^{(e)} + \langle g^{(e)}, u \rangle \quad \forall u \in S. \quad (5.13)$$

Inequality (5.14) follows, involving the *cutting-planes* model $\ell(\cdot; \mathcal{Y})$

$$f(u) \geq \ell(u; \mathcal{Y}) := \max_{e=1, \dots, B} \tilde{f}^{(e)} + \langle g^{(e)}, u \rangle \quad \forall u \in S. \quad (5.14)$$

Bundles may admit a certain number of *anchor points*, i.e. points at which (5.14) becomes tight.

Definition 5.4 (Anchor point). A point $y \in \mathbb{R}^d$ is an *anchor point* of a bundle $\mathcal{Y}$ for f on S if and only if there exists $e \in \{1, \dots, B\}$ such that

$$f(y) = \tilde{f}^{(e)} + \langle g^{(e)}, y \rangle. \quad (5.15)$$

Remark 5.5. Let f be convex and finite at $y \in S$, i.e. $-\infty < f(y) < \infty$. Taking any $g(y) \in \partial f(y)$, the tuple $(\tilde{f} = f(y) - \langle g(y), y \rangle, g = g(y), z = y)$ satisfies inequalities (5.13) and any bundle $\mathcal{Y}$ for f on S containing $(\tilde{f}, g, z)$ trivially admits y as anchor point. Furthermore, when f is differentiable, the unicity of $\nabla f(y)$ as subgradient of f at y implies that

$$f(y) = \tilde{f} + \langle g, y \rangle \Rightarrow \tilde{f} = f(y) - \langle \nabla f(y), y \rangle, \quad g = \nabla f(y). \quad (5.16)$$

Remark 5.6. Let $\mathcal{Y}$ be a bundle about f on S containing $(f(z) - \langle g, z \rangle, g, z)$ with $z \in S$ and $g \in \partial f(z)$. Then, we have

$$f(u) - \ell(u; \mathcal{Y}) \leq f(u) - f(z) - \langle g, u - z \rangle \quad \forall u \in S.$$

In other words, when the bundle $\mathcal{Y}$ stores multiple linearizations $\tilde{f}^{(e)} + \langle g^{(e)}, \cdot \rangle$ ($e \in [B]$) of a function f then $\ell(\cdot; \mathcal{Y})$ becomes a tighter model with respect to f than any single linearization $\tilde{f}^{(e)} + \langle g^{(e)}, \cdot \rangle$.

In this chapter, we tackle the following open questions.

How can we exploit the convex optimization tricks (i.e. extrapolation, line-search and bundles) all at once in the context of (DC) programs (v_{DC}) ? Is it worth it?

5.1.1 Related work

Our work draws its inspiration from a couple of papers, each of which investigating how one (or more) of the above optimization speed-up *tricks* can be employed in the (DC) framework. We describe the most closely related ones. Overall, one can interpret our work as a specific instance of the generic *Sequential (DC)* approach (SDC) introduced in [dO20], yet including *extrapolation* as an extended feature. In SDC, one needs to define a convex model $f_{1,\mathcal{M}}$ of f_1 , *tight* at the current iterate x , and approximately solve the (DC) supproblem of minimizing $f_{1,\mathcal{M}} + \Psi - f_2$ to obtain the next iterate x_+ . To that end, we resort to the local method DCA [LP18], with starting point $x^{(0)} = x$. Say DCA halts after $K_{\text{inner}} \in \mathbb{N}$ iterations. Then,

$$x^{(k_{\text{inner}}+1)} \in \arg \min_{u \in \mathbb{R}^d} f_{1,\mathcal{M}}(u) + \Psi(u) - \langle g_2(x^{(k_{\text{inner}})}), u \rangle \quad (5.17)$$

for every $k_{\text{inner}} = 0, \dots, K_{\text{inner}} - 1$ and $x_+ = x^{(K_{\text{inner}})}$. Actually, such x_+ solves an **implicit** model of $\mathcal{F}$ involving a *cutting-planes* model of f_2 , i.e.

$$x_+ = \arg \min_{u \in \mathbb{R}^d} f_{1,\mathcal{M}}(u) + \Psi(u) - \ell(u; \mathcal{Y}_2), \quad (5.18)$$

where $\mathcal{Y}_2$ stands as a bundle for f_2 on $\mathbb{R}^d$ containing all its successive linearizations within DCA, i.e. $\mathcal{Y}_2$ contains $(f_2(x^{(e)}) - \langle g_2(x^{(e)}), x^{(e)} \rangle, g_2(x^{(e)}), x^{(e)})$ for every $e = 0, \dots, K_{\text{inner}} - 1$. Our method ASDCe produces its main sequence of iterates by operating just like SDC (Eq. (5.17) & (5.18)), excepted that our model of f_1 is not necessarily *tight* at x due to the (*Extrapolation*) step. Akin to [dO19] and [GTBK23], we construct a model $f_{1,\mathcal{M}}$ of f_1 that is a regularized *cutting-planes* model, i.e.

$$f_{1,\mathcal{M}}(u) = \ell(u; \mathcal{Y}_1) + \frac{\hat{L}}{2} \|u - y\|_2^2 \quad \forall u \in \mathbb{R}^d \quad (5.19)$$

for some $\hat{L} > 0$, stability center $y \in \mathbb{R}^d$ and bundle $\mathcal{Y}_1$ of f_1 *tight* at y . However, our motivations are different. Both cited papers do not assume f_1 's smoothness. The bundles therein serve to pass iterations and, consequently, prove the asymptotical convergence of their respective algorithms.

In the spirit of [NF22], we want our bundle about f_1 to harvest local smoothness for accelerating convergence. Using the smoothness of f_1 , our convergence analysis is made simpler and allows that the size of $\mathcal{Y}_1$ is kept to 1 throughout all the optimization process, unlike [dO19] ($|\mathcal{Y}_1|$ can be kept as small as 3) and [GTBK23]. Our *extrapolation* coefficients follow the mechanics of [WCP18] and [TFT22]. Yet, since we incorporate *line-search* to adapt the local smoothness parameter $\hat{L}$, we cut-off iteratively the maximal value ς can take in (Extrapolation). To conclude, we list several papers together with their own features. The symbol / in Table 5.1 signifies "not applicable" due to other types of **minimization** oracles (e.g. DCA-like algorithms use the proximal mapping of $f_1 + \Psi$ as a whole). In addition, the symbol $\circ$ means that the feature is suggested by the author(s) but not explicitly defined in an algorithm.

$\downarrow$ name \ features $\rightarrow$	<i>extrapolation</i>	<i>adaptive $\hat{L}$</i>	<i>bundles</i>
PBM [dO19]	No	Yes	mostly f_1
PDCe [WCP18]	Yes	No	No
BEM-DC [GTBK23]	No	Yes	f_1 / f_2
BPGe [TFT22]	Yes	No	No
nm-EPDCA-LS [LZ19]	No	Yes	No
sPDCAe [YN23]	Yes	Yes, monotonic	No
BDCA [ACV22]	Yes	/	No
HDCA-In [Niu23]	Yes	/	No
DME [THP20]	No	No	No
DPAC ₃ [PU23]	No	/	f_1 / f_2 (implicit)
SDC [dO20]	No	$\circ$	f_1 / f_2 (implicit)
ASDCe (this work)	Yes	Yes	f_1 / f_2 (implicit)

Table 5.1 A short literature review of recent (DC) methods.

5.1.2 Contributions

- We propose Lemma 5.12 (Section 5.2) that characterizes the guarantees one obtains by minimizing regularized *difference-of-bundles* models of the objective $\mathcal{F}$, i.e. (5.18) with $f_{1,\mathcal{M}}$ as described in (5.19).
- We combine *extrapolation*, *non-monotone backtracking line-search* and possibly bundles about both f_1 and f_2 all at once in a single algorithm (Section 5.3), i.e. ASDCe. To the best of our knowledge, we are the first to do so. We prove in Theorem 5.16 that some divergence between successive iterates of ASDCe vanishes as $\mathcal{O}(K^{-1})$ with K the number of passed iterations.

- We conduct numerical experiments (Section 5.4) that showcase instances of (DC) programs for which combining from two up to three of the ingredients (i.e. *extrapolation*, *line-search* and *bundles*) is beneficial.

5.1.3 Motivation

We introduce now two classes of (DC) problems to motivate our framework. We will build upon the first one (and a variant) to create benchmark problems to assess the utility of each of the optimization *tricks* detailed in this section, i.e. *extrapolation*, *line-search* and *bundles*. Here below $p \in \{1, \infty\}$.

L_p -ball Trust Region Let $\mathbf{U}_1 = \mathbf{U}_1^T \geq \mathbf{0}_{d \times d}$ and $\mathbf{U}_2 = \mathbf{U}_2^T \geq \mathbf{0}_{d \times d}$, $\bar{\mathbf{u}} \in \mathbb{R}^d$, $R > 0$. A p -norm ball Trust Region problem from [ACV22] reads, for $\mathbf{U} = \mathbf{U}_1 - \mathbf{U}_2$ and $\mathcal{X} = \mathbb{B}_p(\mathbf{0}_d; R)$,

$$\min_{x \in \mathbb{B}_p(\mathbf{0}_d; R)} \underbrace{\frac{1}{2} \langle \mathbf{U}_1 x, x \rangle + \langle \bar{\mathbf{u}}, x \rangle}_{f_1(x)} - \underbrace{\frac{1}{2} \langle \mathbf{U}_2 x, x \rangle}_{f_2(x)}. \quad (L_p\text{-TRS})$$

Here, regarding (v_{DC}) , we let $\Psi = \mathbb{I}_{\mathcal{X}}$. We notice that f_1 is smooth with ∇f_1 being $L = \|\mathbf{U}_1\|_2$ Lipschitz continuous and that $\mathcal{X}$ is bounded polyhedron. We also note that any $\mathbf{U} \in \mathbb{R}^{d \times d}$ can be decomposed as $\mathbf{U} = \bar{\mathbf{U}}_1 - \bar{\mathbf{U}}_2 = (\|\mathbf{U}\|_2 \mathbf{I}_d) - (\|\mathbf{U}\|_2 \mathbf{I}_d - \mathbf{U})$ and that Euclidean projections on an L_1 and L_∞ -balls are achieved in $O(d)$ operations.

p -norm Generalized Eigenvalue Let $\mathbf{U} = \mathbf{U}^T \geq \mathbf{0}_{d \times d}$, $\mathbf{V} \in \mathbb{R}^{N \times d}$. A p -norm generalized (leading) eigenvector of V reads

$$\tilde{x}^* \leftarrow \arg \max_{\tilde{x} \in \mathbb{R}^d} \|\mathbf{V} \tilde{x}\|_p \quad \text{s.t.} \quad \tilde{x}^T \mathbf{U} \tilde{x} = 1.$$

Let $C > 0$. According to [Yua23], one can obtain $\tilde{x}^*$ by solving the following unconstrained (DC) problem (i.e. $\Psi = \mathbf{0}$ and $\text{dom } \mathcal{F} = \mathbb{R}^d$)

$$\min_{x \in \mathbb{R}^d} \underbrace{(C/2) x^T \mathbf{U} x}_{f_1(x)} - \underbrace{\|\mathbf{V} x\|_p}_{f_2(x)}. \quad (\text{GEP})$$

and then setting $\tilde{x}^* = \pm x^* / \sqrt{(x^*)^T \mathbf{U} x^*}$ with x^* a global minimizer of (GEP). This time, the gradient of f_1 appears $C \cdot \|\mathbf{U}\|_2$ Lipschitz continuous. The f_2 component is piecewise-linear and not differentiable on the subspaces $\ker(\mathbf{V}) = \{x \in \mathbb{R}^d \mid \mathbf{V}x = \mathbf{0}_N\}$ and $\{x \in \mathbb{R}^d \mid \exists s \in [N], (\mathbf{V}x)_s = 0\}$ for respectively $p = \infty$ and $p = 1$.

5.2 Regularized Difference-of-Bundles optimization

Aside from the specifications regarding f_1 , f_2 , Ψ and the boundedness of (ν_{DC}) at the beginning of Chapter 4, we now state a few other assumptions.

Assumption 5.7. Function f_1 is continuously differentiable on an open set containing $\text{dom } \Psi$ and there exists a constant $L > 0$ so that at any $y \in \text{dom } \Psi$, we have access to $(f_1(y), \nabla f_1(y))$ and it holds for any $x_+ \in \text{dom } \Psi$,

$$f_1(x_+) \leq f_1(y) + \langle \nabla f_1(y), x_+ - y \rangle + \frac{L}{2} \|x_+ - y\|_2^2. \quad (\text{A})$$

Assumption 5.8. f_2 is proper convex with full domain, i.e. $\text{dom } f_2 = \mathbb{R}^d$. At any $\bar{x} \in \mathbb{R}^d$, we have access to $(f_2(\bar{x}), g_2(\bar{x}))$ where $g_2 \in \partial f_2(\bar{x})$. Also, f_2 is closed so that the next implicit formulation is valid:

$$f_2(u) = \sup_{\tilde{u} \in \mathbb{R}^d} \max_{g_2(\tilde{u}) \in \partial f_2(\tilde{u})} f_2(\tilde{u}) + \langle g_2(u), u - \tilde{u} \rangle. \quad (\text{B})$$

Moreover, for some theoretical properties, we need a third assumption.

Assumption 5.9. For any $\nu \in \mathbb{R}$, the sublevel set $S_{\mathcal{F}, \nu}$ is bounded.

$$S_{\mathcal{F}, \nu} := \{x \in \mathbb{R}^d \mid \mathcal{F}(x) \leq \nu\}. \quad (\text{C})$$

Finally, our **minimization** oracle (MO) reads as follows and suffices to minimize regularized *difference-of-bundles* models of $\mathcal{F}$.

Assumption 5.10. Let $B \in \mathbb{N}$ and $\hat{L} > 0$. Considering a collection $\Phi = \{\tilde{\gamma}^{(e)}, \tilde{\beta}^{(e)}\}_{e \in [B]}$ of elements in $\mathbb{R} \times \mathbb{R}^d$, it is *cheap* to compute an element of

$$\arg \min_{u \in \mathbb{R}^d} \max_{e \in [B]} \tilde{\gamma}^{(e)} + \langle \tilde{\beta}^{(e)}, u \rangle + \frac{\hat{L}}{2} \|u\|_2^2 + \Psi(u). \quad (\text{D})$$

Remark 5.11. Since Ψ is convex, as soon as Assumption (D) is fulfilled with $B = 1$, it is possible to build a more powerful oracle for $B > 2$ up to arbitrary accuracy (see e.g. [NF22]). We assume throughout this chapter that the computations are *exact*.

Descent guarantees Cornerstone of this chapter, we prove Lemma 5.12 that provides a decrease property tailored for regularized *difference-of-bundles* (DOB) optimization. It will serve to prove the convergence of ASDCe.

Lemma 5.12. Let $\mathcal{Y}_\star = \{\tilde{f}_\star^{(e)}, g_\star^{(e)}, z_\star^{(e)}\}_{e=1}^{B_\star}$ represent a bundle for $f_\star$ ($\star \in \{1, 2\}$) on $\text{dom } \Psi$. Let $y \in \text{dom } \Psi$ (respectively x) be an anchor point for $\mathcal{Y}_1$ (respectively $\mathcal{Y}_2$) and let Assumption (A), (B) & (D) hold. Then, there exists $e^* \in [B_2]$ such that $f_2(x) = \tilde{f}_2^{(e^*)} + \langle g_2^{(e^*)}, x \rangle$ and for any $\hat{L} > 0$,

$$\mathcal{F}(x_+) \leq \mathcal{F}(x) + \frac{\hat{L}}{2} \|y - x\|_2^2 - \frac{\hat{L}}{2} \left(\|x - (x)_+\|_2^2 + \left(1 - \frac{L}{\hat{L}}\right) \|x_+ - y\|_2^2 \right)$$

with

$$(x)_+ = \arg \min_{u \in \mathbb{R}^d} \ell(u; \mathcal{Y}_1) - \langle g_2^{(e^*)}, u \rangle + \frac{\hat{L}}{2} \|u - y\|_2^2 + \Psi(u), \quad (5.20)$$

$$x_+ \in \arg \min_{u \in \mathbb{R}^d} \mathcal{H}(u) := \ell(u; \mathcal{Y}_1) - \ell(u; \mathcal{Y}_2) + \frac{\hat{L}}{2} \|u - y\|_2^2 + \Psi(u). \quad (5.21)$$

Proof. We first notice that

$$\begin{aligned} \mathcal{H}(x_+) &= \min_{e \in [B_2]} v(e) := \min_{u \in \mathbb{R}^d} \ell(u; \mathcal{Y}_1) - [\tilde{f}_2^{(e)} + \langle g_2^{(e)}, u \rangle] + \frac{\hat{L}}{2} \|u - y\|_2^2 + \Psi(u) \\ &= \min_{e \in [B_2]} \ell(x_+(\Phi(e), \hat{L}); \mathcal{Y}_1) - [\tilde{f}_2^{(e)} + \langle g_2^{(e)}, x_+(\Phi(e), \hat{L}) \rangle] \\ &\quad + \frac{\hat{L}}{2} \|x_+(\Phi(e), \hat{L}) - y\|_2^2 + \Psi(x_+(\Phi(e), \hat{L})) \end{aligned}$$

where $x_+(\Phi(e), \hat{L})$ is computed thanks to the **minimization** oracle (D) with

$$\Phi(e) = \left\{ \tilde{f}_1^{(e')} - \tilde{f}_2^{(e)} + \frac{\hat{L}}{2} \|y\|_2^2, g_1^{(e')} - g_2^{(e)} - \hat{L}y \right\}_{e'=1}^{B_1}.$$

Accordingly, we can write

$$\mathcal{H}(x_+) \leq v(e^*) = \min_{u \in \mathbb{R}^d} \ell(u; \mathcal{Y}_1) - [\tilde{f}_2^{(e^*)} + \langle g_2^{(e^*)}, u \rangle] + \frac{\hat{L}}{2} \|u - y\|_2^2 + \Psi(u).$$

5 | Adaptive Sequential (DC) algorithm with extrapolation

Let $(x)_+$ minimize the above right-hand side. First-order optimality conditions at $(x)_+$ imply the existence of $q \in \Delta^{B_1}$ and $g_\Psi \in \partial\Psi((x)_+)$ such that

$$\left\langle \sum_{e'=1}^{B_1} (q)_{e'} g_1^{(e')} - g_2^{(e^*)} + g_\Psi + \hat{L}((x)_+ - y), u - (x)_+ \right\rangle \geq 0 \quad \forall u \in \mathbb{R}^d. \quad (5.22)$$

Then, the following identity holds for any points $(x)_+, y$ and x ,

$$2 \langle (x)_+ - y, -x + (x)_+ \rangle = \|(x)_+ - y\|_2^2 + \|x - (x)_+\|_2^2 - \|x - y\|_2^2.$$

Combining inequality (5.22) for $u = x$ and the last equality yields

$$\left\langle \sum_{e'=1}^{B_1} (q)_{e'} g_1^{(e')} - g_2^{(e^*)} + g_\Psi, (x)_+ - x \right\rangle \leq \frac{\hat{L}}{2} \left(-\|(x)_+ - y\|_2^2 - \|x - (x)_+\|_2^2 + \|x - y\|_2^2 \right).$$

Rewriting this inequality as

$$\begin{aligned} & \left\langle \sum_{e'=1}^{B_1} (q)_{e'} g_1^{(e')} - g_2^{(e^*)}, (x)_+ \right\rangle + \frac{\hat{L}}{2} \|(x)_+ - y\|_2^2 + \langle g_\Psi, (x)_+ - x \rangle \\ & \leq \left\langle \sum_{e'=1}^{B_1} (q)_{e'} g_1^{(e')} - g_2^{(e^*)}, x \right\rangle + \frac{\hat{L}}{2} (\|x - y\|_2^2 - \|x - (x)_+\|_2^2) \end{aligned}$$

and then adding $\Psi((x)_+) + \langle g_\Psi, x - (x)_+ \rangle - \tilde{f}_2^{(e^*)} + \sum_{e'=1}^{B_1} (q)_{e'} \tilde{f}_1^{(e')}$ on both sides allows to write (thanks to *bundle* properties, i.e. Eq. (5.14))

$$\begin{aligned} \nu(e^*) & \leq \ell((x)_+; \mathcal{Y}_1) - [\tilde{f}_2^{(e^*)} + \langle g_2^{(e^*)}, (x)_+ \rangle] + \frac{\hat{L}}{2} \|(x)_+ - y\|_2^2 + \Psi((x)_+) \\ & \leq \mathcal{F}(x) + \frac{\hat{L}}{2} (\|x - y\|_2^2 - \|x - (x)_+\|_2^2). \end{aligned} \quad (5.23)$$

Indeed, $(q)_{e'} > 0$ if and only if $\tilde{f}_1^{(e')} + \langle g_1^{(e')}, (x)_+ \rangle = \ell((x)_+; \mathcal{Y}_1)$ and, as usual for proper convex functions, $\Psi((x)_+) + \langle g_\Psi, x - (x)_+ \rangle \leq \Psi(x)$.

We also used the fact that $f_2(x) = \tilde{f}_2^{(e^*)} + \langle g_2^{(e^*)}, x \rangle$. Since y is an anchor point for $\mathcal{Y}_1$, the differentiability of f_1 and Remark 5.5 lead us to conclude that there exists $\bar{e} \in \{1, \dots, B_1\}$ such that

$$\tilde{f}_1^{(\bar{e})} = f_1(y) - \langle \nabla f_1(y), y \rangle, \quad g_1^{(\bar{e})} = \nabla f_1(y).$$

Recalling f_1 's L -smoothness (Assumption 5.7), we write sucessively that

$$\begin{aligned}
 v(e^*) &\geq \mathcal{H}(x_+) \\
 &= \ell(x_+; \mathcal{Y}_1) - \ell(x_+; \mathcal{Y}_2) + \frac{\hat{L}}{2} \|x_+ - y\|_2^2 + \Psi(x_+) \\
 &\geq \ell(x_+; \mathcal{Y}_1) - f_2(x_+) + \frac{\hat{L}}{2} \|x_+ - y\|_2^2 + \Psi(x_+) \\
 &= f_1(y) + \langle \nabla f_1(y), x_+ - y \rangle - f_2(x_+) + \frac{\hat{L}}{2} \|x_+ - y\|_2^2 + \Psi(x_+) \\
 &\geq f_1(x_+) - \frac{L}{2} \|x_+ - y\|_2^2 - f_2(x_+) + \frac{\hat{L}}{2} \|x_+ - y\|_2^2 + \Psi(x_+) \\
 &= \mathcal{F}(x_+) - \frac{(L - \hat{L})}{2} \|x_+ - y\|_2^2.
 \end{aligned} \tag{5.24}$$

Combining (5.23) and (5.24) yields the desired result. $\square$

Reading between lines, Lemma 5.12 tells that when the stepsize $\hat{L}^{-1}$ is sufficiently small and when first-order **information** about f_1 and f_2 is collected around points y and x close enough from each other then the global minimizer of the model $\mathcal{H}$ in (5.21) will decrease the value of $\mathcal{F}$.

Corollary 5.13 extends and summarizes this last statement formally.

Corollary 5.13. Let the requirements of Lemma 5.12 hold with $g_2^{(e^*)} \in \partial f_2(x)$, $f_2^{(e^*)} = f_2(x) - \langle g_2^{(e^*)}, x \rangle$, $y = x + \varsigma((x_-)_+ - x_-)$ for any x_- , $(x_-)_+ \in \text{dom } \Psi$ and $\varsigma \in [0, 1]$. Then, $(x)_+$ and x_+ as computed in (5.20) and (5.21) jointly satisfy

$$\begin{aligned}
 \mathcal{F}(x_+) + \frac{\hat{L}}{2} \|x - (x)_+\|_2^2 &\leq \mathcal{F}(x) + \frac{\hat{L}}{2} \varsigma^2 \|x_- - (x_-)_+\|_2^2 \\
 &\quad + \frac{\hat{L} - L}{2} \|x_+ - y\|_2^2
 \end{aligned} \tag{5.25}$$

Proof. Inequality (5.25) follows verbatim from the application of Lemma 5.12. $\square$

Remark 5.14. Let $\hat{L} > L$. If $\varsigma = 0$ (hence $y = x$ in Lemma 5.12) then (5.25) implies that $\mathcal{F}(x_+) < \mathcal{F}(x)$. Otherwise, for any $\varsigma < 1$, we will observe in Section 5.3.1 that another key quantity (i.e. not the objective value) is ensured to decrease. This latter quantity is referred to as a *Lyapunov potential*.

5.3 ASDCe

In this section, we present and analyze ASDCe, our iterative algorithm involving the sequential optimization of regularized *cutting-plane* models (as depicted in the previous section) to obtain new iterates. Highlighted in Remark 5.14 about Corollary 5.13, given all its features (e.g. *extrapolation*), the minimizer of our models (see E.q. (5.21)) does not necessarily decrease the objective value. To perform a proper study of ASDCe, we need to specify a *Lyapunov potential* that will also serve in the iteration acceptance rule.

In Algorithm 1 (ASDCe), some iterate sequences are implicitly defined, e.g. $\{x_k\}_{k=0}^\infty$ and $\{y_k\}_{k=0}^\infty$ for any integer k . In addition, we also denote by $\mathcal{Y}_{1,k}$ (respectively $\mathcal{Y}_{2,k}$) the bundle for f_1 (respectively f_2) on $\text{dom } \Psi$ and by L_k the parameter of estimated local smoothness used at iteration k .

5.3.1 Lyapunov potential

Crucial to build our convergence gauge is the sequence $\{(x_k)_+\}_{k=0}^\infty$, whose elements are defined for every $k \in \mathbb{N} \cup \{0\}$ as

$$(x_k)_+ := \arg \min_{u \in \mathbb{R}^d} \ell(u; \mathcal{Y}_{1,k}) - (f_2(x_k) + \langle g_2(x_k), u - x_k \rangle) + \frac{L_k}{2} \|u - y_k\|_2^2 + \Psi(u)$$

with $g_2(x_k) \in \partial f_2(x_k)$. We are now ready to formulate our new *Lyapunov potential*. We fix $P_0 := \mathcal{F}(x_0)$ and for any iteration $k \geq 0$,

$$P_{k+1} := \mathcal{F}(x_{k+1}) + \frac{L_k}{2} \|x_k - (x_k)_+\|_2^2. \quad (5.26)$$

We will show in Section 5.3.3 that $\{P_k\}_{k=0}^\infty$ is a non-increasing sequence. It is lower-bounded by the optimal value of the problem, i.e. $F^* > -\infty$.

Furthermore, our iteration acceptance rule will imply that if it is not possible to get $P_{k+1} < P_k$ then x_{k+1} must be a *critical point*.

Remark 5.15. Let $B_{2,k} := |\mathcal{Y}_{2,k}| = 1$ for every $k \geq 0$, i.e. the second *bundle* always involves a single tuple that is the first-order **information** about f_2 at x_k . Then x_{k+1} turns out to be $(x_k)_+$ and P_{k+1} boils down to a more *casual* form where the squared distance between the two last successive iterates from $\{x_{k'}\}_{k'=0}^{k+1}$ comes at play, i.e. $\mathcal{F}(x_{k+1}) + \frac{L_k}{2} \|x_k - x_{k+1}\|_2^2$ as in [YN23].

5.3.2 Main algorithm

Algorithm 1 (ASDCe) presented below includes some hyper-parameters that we discuss now. It also assumes, under the hat, some user prescribed policies in what concerns the line-search methodology (e.g. fixing $\hat{L}$), the constitution of bundles and the extrapolation mechanism. We will come back to this matter when specifying our implementation details for ASDCe.

First of all, one needs a lower limit on the (local) smoothness parameter $\hat{L}$ that will be used. This latter will be noted $L_{\min}$. Then, $\tau_{\text{ext}} \in (0, 1)$ will ensure that $\sup_{k \in \mathbb{N}} L_k / L_{k-1} \zeta_k^2 < 1$ (technicality useful in Theorem 5.16). To adjust the *on-the-go* (local) smoothness parameter, we employ the parameter $\tau_{\text{loc}} \in (0, 1)$. That is, when the estimate should grow, it comes $L_k \leftarrow \tau_{\text{loc}}^{-1} L_k$. Lastly, $\tau_{\text{exc}} \in (0, 1/2)$ comes in handy to endow our acceptance rule (i.e. exit condition of the while loop) with some flexibility.

Algorithm 1 ASDCe(ϵ)

Require: $(x_{-1})_+ = x_0 = x_{-1} \in \text{dom } \Psi, 0 < \tau_{\text{ext}} < 1, 0 < \tau_{\text{exc}} < 1/2,$
 $0 < \tau_{\text{loc}} < 1, \text{ counter } k = 0, 0 < L_{\min} \leq L_{-1}$
while $k > 0$ **and** $\|x_{k+1} - x_k\|_2^2 \geq \epsilon$ **do**
 $\hat{L} \in [L_{\min}, L_{k-1}]$
 $(L_k, E_k) = (\hat{L}, 0)$
 while $E_k < 0$ **do**
 $\zeta_k \in [0, \sqrt{L_{k-1} \tau_{\text{ext}} / L_k}]$
 $(x^{(0)}, y_k) = (x_k, x_k + \zeta_k ((x_{k-1})_+ - x_{k-1}))$
 $\hat{f}_{1,k}(\cdot) = \ell(\cdot; \mathcal{Y}_{1,k}) + \frac{L_k}{2} \|\cdot - y_k\|_2^2$
 for $k_{\text{inner}} = 0, \dots, K_{\text{inner}} - 1$ **do** $\triangleright$ DCA (cfr. (5.17))
 $g_2(x^{(k_{\text{inner}})}) \in \partial f_2(x^{(k_{\text{inner}})})$
 $x^{(k_{\text{inner}}+1)} = \arg \min_{u \in \mathbb{R}^d} \hat{f}_{1,k}(u) + \Psi(u) - \langle g_2(x^{(k_{\text{inner}})}), u \rangle$
 end for
 $(x_{k+1}, (x_k)_+) = (x^{(K_{\text{inner}})}, x^{(1)})$
 $P_{k+1} = \mathcal{F}(x_{k+1}) + \frac{L_k}{2} \|(x_k)_+ - x_k\|_2^2$
 $E_k = \mathcal{F}(x_k) + \frac{L_k}{2} \|y_k - x_k\|_2^2 - P_{k+1} - L_{\min} \tau_{\text{exc}} \|x_{k+1} - y_k\|_2^2$
 if $E_k < 0$ **then**
 $L_k \leftarrow \tau_{\text{loc}}^{-1} L_k$ $\triangleright$ backtracking required
 else
 $k \leftarrow k + 1$ $\triangleright$ successful iteration occurred
 end if
 end while
end while

5 | Adaptive Sequential (DC) algorithm with extrapolation

Well-definiteness

Prior to give more insights about our algorithm, we show that it will never be stuck during iteration $k \geq 0$, i.e. after a finite number of *line-search backtracks*, we must have $E_k \geq 0$. Indeed, Corollary 5.13 applies in a straightforward fashion with $\hat{L} = L_k$ and $g_2^{(e^*)} \in \partial f_2(x_k)$. We see that our iteration acceptance rule $E_k \geq 0$ will be triggered if $\frac{L_k - L}{2} \geq \tau_{\text{exc}} L_{\min}$. Thereby, if $L_k \geq L + 2\tau_{\text{exc}} L_{\min}$ then $E_k \geq 0$. This would happen after no more than $1 + \lceil \log_{\tau_{\text{loc}}^{-1}}(L/L_{\min} + 2\tau_{\text{exc}}) \rceil$ *backtracks* during any iteration.

Implementation details

Line-search At the beginning of every iteration $k \geq 0$, parameter $\hat{L}$ either takes the value (i) L_{k-1} or (ii) $\max\{L_{\min}, \tau_{\text{loc}} L_{k-1}\}$. We allow $\hat{L}$ to decrease (i.e. option (ii)) with respect to L_{k-1} if and only if, for $K_{\text{trig}} \in \mathbb{N}$ iterations in a row, no *backtracking* occurred. In other words, this happens if $\hat{L}$ estimate of local-smoothness was good enough for K_{trig} iterations straight so that, perhaps, our iteration acceptance rule might pass with a smaller L_k than $\hat{L}$.

Bundles We initiate $\mathcal{Y}_{1,0}$ as $\emptyset$. Let $k \geq 0$ be the iteration counter. Any time the bundle $\mathcal{Y}_{1,k}$ must be involved in the optimization of a model, i.e. through $\hat{f}_{1,k}$ within the DCA steps, we remove (if necessary) its oldest elements so that its size is at most $B_{1,\max} - 1 \geq 0$. Then, the tuple $(f_1(y_k) - \langle \nabla f_1(y_k), y_k \rangle, \nabla f_1(y_k), y_k)$ is stored in $\mathcal{Y}_{1,k}$ so that $|\mathcal{Y}_{1,k}| \leq B_{1,\max}$. At the end of the model optimization, i.e. after the last DCA inner-step

$$x^{(K_{\text{inner}})} = \arg \min_{u \in \mathbb{R}^d} \hat{f}_{1,k}(u) + \Psi(u) - \langle g_2(x^{(K_{\text{inner}}-1)}), u \rangle, \quad (5.27)$$

two more tuples are added to $\mathcal{Y}_{1,k}$ if *backtracking* occurs. Similar to [dO19], based on the multipliers $q \in \Delta^{B_1}$ (cfr. (5.22)) linked to (5.27), we store

$$\left(\sum_{e'=1}^{B_1} (q)_{e'} \tilde{f}_1^{(e')}, \sum_{e'=1}^{B_1} (q)_{e'} g_1^{(e')}, x^{(K_{\text{inner}})} \right), \\ (f_1(x^{(K_{\text{inner}})}) - \langle \nabla f_1(x^{(K_{\text{inner}})}), x^{(K_{\text{inner}})} \rangle, \nabla f_1(x^{(K_{\text{inner}})}), x^{(K_{\text{inner}})}).$$

As previously written, $\mathcal{Y}_{2,k}$ is **implicit**/not known in advance. The size $|\mathcal{Y}_{2,k}|$ can be limited to a specified number $B_{2,\max}$ by stopping DCA earlier, accordingly. In the end, $\mathcal{Y}_{2,k}$ includes **information** about $x^{(0)} = x_k$ and

$$(f_2(x^{(k_{\text{inner}})}) - \langle g_2(x^{(k_{\text{inner}})}), x^{(k_{\text{inner}})} \rangle, g_2(x^{(k_{\text{inner}})}), x^{(k_{\text{inner}})}) \quad \forall k_{\text{inner}} \in [K_{\text{inner}} - 1].$$

Extrapolation Akin to [BT09], an auxiliary sequence $\{o_k\}_{k \in \mathbb{N}}$ with $o_0 = 1$ first suggests an *extrapolation parameter* at iteration $k \geq 0$, i.e. $(o_k - 1)/o_{k+1}$. For convergence properties (see Theorem 5.16), we also need to specify a *maximal extrapolation* $\hat{\tau} \in [0, 1]$. Let $\varsigma_0 = 0$. For any $k \geq 1$, we define

$$o_{k+1} = (1/2) \cdot (1 + \sqrt{1 + 4o_k^2}), \quad \varsigma_k = \min \left\{ \sqrt{L_{k-1} \tau_{\text{ext}} / L_k}, (o_k - 1)/o_{k+1}, \hat{\tau} \right\}.$$

5.3.3 Convergence

We proceed now to show, step-by-step, that any accumulation point of the main sequence $\{x_k\}_{k=0}^\infty$ produced by ASDCe must be a *critical point* of (v_{DC}) .

Step 1. We start by showing that $\{P_k\}_{k=0}^\infty$ must decrease each time the iteration counter increases, i.e. a *successful iteration* occurred.

Step 2. Thanks to Assumption (A) & (C), we will then manage to prove that $\lim_{k \rightarrow \infty} \|(x_{k-1})_+ - x_{k-1}\|_2 = 0$ and $\lim_{k \rightarrow \infty} \|x_{k+1} - y_k\|_2 = 0$ which in turn will imply $\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\|_2 = 0$.

Step 3. Finally, the judicious use of a converging subsequence of $\{x_k\}_{k=0}^\infty$, i.e. $\{x_{\bar{k}}\}_{\bar{k} \in \mathcal{K}}$ defined by an infinite subsequence of indices $\mathcal{K} \subseteq \mathbb{N}$ with $\lim_{\bar{k} \rightarrow \infty, \bar{k} \in \mathcal{K}} x_{\bar{k}} = x^* \in \text{dom } \Psi$, together with first-order optimality conditions at $(x_{\bar{k}})_+$ will imply that x^* satisfies (DC-criticality).

Theorem 5.16. Let Assumption (A) & (C) hold and let $\delta x_k := (x_{k-1})_+ - x_{k-1}$ for every $k \in \mathbb{N}$ where $\{x_k\}_{k=0}^\infty$ is main sequence produced by Algorithm 1. Then,

- (i). $\{P_k\}_{k=0}^\infty$ is non-increasing.
- (ii). $\lim_{k \rightarrow \infty} \|\delta x_k\|_2 = 0$, $\lim_{k \rightarrow \infty} \|x_{k+1} - y_k\|_2 = 0$ and thus

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\|_2 = 0.$$

- (iii). $\min_{0 \leq k \leq K-1} \frac{\|\delta x_k\|_2^2}{2} + \frac{\tau_{\text{exc}}}{(1 - \tau_{\text{ext}})} \|x_{k+1} - y_k\|_2^2 \leq \frac{L_{\min}^{-1} (\mathcal{F}(x_0) - F^*)}{K(1 - \tau_{\text{ext}})}.$

Proof. (i). For any integer $k \geq 0$, according to our iteration acceptance rule (i.e. the exit condition of the `while` loop) and the fact that $L_k \varsigma_k^2 \leq L_{k-1}$,

$$E_k \geq 0 \Rightarrow P_{k+1} \leq \mathcal{F}(x_k) + \frac{L_k \varsigma_k^2}{2} \|(x_{k-1})_+ - x_{k-1}\|_2^2 \leq P_k.$$

Since k does not increase until $E_k \geq 0$ then $P_{k+1} \leq P_k$.

5 | Adaptive Sequential (DC) algorithm with extrapolation

(ii). Let us develop the following inequality involving series

$$\begin{aligned} \sum_{k=0}^{\infty} (P_k - P_{k+1}) &\geq \sum_{k=0}^{\infty} \frac{L_{k-1}}{2} \|\delta x_k\|_2^2 + L_{\min} \tau_{\text{exc}} \|x_{k+1} - y_k\|_2^2 - \frac{L_k \zeta_k^2}{2} \|\delta x_k\|_2^2 \\ &\geq (1 - \tau_{\text{ext}}) L_{\min} \sum_{k=0}^{\infty} \frac{\|\delta x_k\|_2^2}{2} + \tau_{\text{exc}} (1 - \tau_{\text{ext}})^{-1} \|x_{k+1} - y_k\|_2^2. \end{aligned} \quad (5.28)$$

Since $P_0 = \mathcal{F}(x_0)$ and $F^* \leq \liminf_{k \rightarrow \infty} \mathcal{F}(x_{k+1}) \leq \liminf_{k \rightarrow \infty} P_{k+1}$, the left-hand side is bounded by $(\mathcal{F}(x_0) - F^*) < \infty$. Therefore, it immediately comes that both

$$\lim_{k \rightarrow \infty} \|\delta x_k\|_2 = 0 \quad \text{and} \quad \lim_{k \rightarrow \infty} \|x_{k+1} - y_k\|_2 = 0.$$

Noticing that $\|x_{k+1} - x_k\|_2 = \|x_{k+1} - y_k + y_k - x_k\|_2 \leq \|x_{k+1} - y_k\|_2 + \zeta_k \|\delta x_k\|_2$ and $\sup_{k \geq 0} \zeta_k \leq 1$ allows us to state that iterates get closer, i.e.

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\|_2 = 0.$$

(iii) Same reasoning as (ii) but considering the partial sum of (5.28) up to $K - 1$ where $K \in \mathbb{N}$ denotes the number of performed outer iterations. $\square$

Remark 5.17. Bullet (iii) above provides asymptotical guarantees, i.e.

$$\lim_{k \rightarrow \infty} \frac{\|\delta x_k\|_2^2}{2} + \tau_{\text{exc}} (1 - \tau_{\text{ext}})^{-1} \|x_{k+1} - y_k\|_2^2 = 0$$

However, if it happens that $\|\delta x_{k^*}\|_2 = \|x_{k^*+1} - y_{k^*}\|_2 = 0$ for some $k^* \geq 0$ then Lemma 5.18 below applies and claims (DC-criticality).

Lemma 5.18. Let $\delta x_k := (x_{k-1})_+ - x_{k-1}$ for every $k \in \mathbb{N}$ and let an integer $k^* \geq 0$ be such that $\zeta_{k^*} \delta x_{k^*} = \mathbf{0}_d$ and $x_{k^*+1} = y_{k^*}$ within Algorithm 1 then x_{k^*+1} is a critical point for problem (v_{DC}) , namely according to Assumption (A), there exists $g_2(x_{k^*+1}) \in \partial f_2(x_{k^*+1})$ such that

$$-\left(\nabla f_1(x_{k^*+1}) - g_2(x_{k^*+1})\right) \in \partial \Psi(x_{k^*+1}).$$

Proof. We extend the demonstration of ([dO19], Lemma 2) to the difference-of-bundles (DOB) framework. We first notice that $x_{k^*+1} = x_{k^*}$ since $y_{k^*} =$

$x_{k*} + \zeta_{k*} \delta x_{k*} = x_{k*} + \mathbf{0}_d$ and $x_{k*+1} = y_{k*}$ by assumption. Then it comes

$$\begin{aligned}
 \mathcal{F}(x_{k*}) &= \mathcal{F}(x_{k*+1}) \\
 &= \ell(x_{k*+1}; \mathcal{Y}_{1,k*}) - \ell(x_{k*+1}; \mathcal{Y}_{2,k*}) + \Psi(x_{k*+1}) \\
 &= \min_{u \in \mathbb{R}^d} \ell(u; \mathcal{Y}_{1,k*}) - \ell(u; \mathcal{Y}_{2,k*}) + \frac{L_{k*}}{2} \|u - y_{k*}^*\|_2^2 + \Psi(u) \\
 &\leq \min_{u \in \mathbb{R}^d} f_1(u) - \ell(u; \mathcal{Y}_{2,k*}) + \frac{L_{k*}}{2} \|u - x_{k*}\|^2 + \Psi(u) \quad (5.29) \\
 &\leq f_1(x_{k*}) - f_2(x_{k*}) + 0 + \Psi(x_{k*}) = \mathcal{F}(x_{k*}).
 \end{aligned}$$

The second equality is justified by the fact that $x_{k*+1} = x_{k*}$ represents, by construction, an anchor for both $\mathcal{Y}_{1,k*}$ and $\mathcal{Y}_{2,k*}$ whereas the first inequality is due to bundles' properties. Finally, the last inequality simply comes from x_{k*} 's feasibility for problem (5.29). One can deduce that $x_{k*} = x_{k*+1}$ is actually a global minimizer of (5.29) and therefore

$$\partial \ell(x_{k*}; \mathcal{Y}_{2,k*}) = \partial \ell(x_{k*+1}; \mathcal{Y}_{2,k*}) \subseteq \nabla f_1(x_{k*+1}) + \partial \Psi(x_{k*+1}).$$

Subdifferential set calculus together with the fact that for any $k \geq 0$, $\partial f_2(x_k) \cap \partial \ell(x_k; \mathcal{Y}_{2,k}) \supseteq \{g_2^\circ(x_k)\} \neq \emptyset$ allows to conclude. $\square$

Steps (1-2) are covered. We proceed with (3), akin to [TFT22] Theorem 6.

Theorem 5.19. Let Assumption (A) & (C) hold. Algorithm 1 produces a sequence $\{x_k\}_{k=0}^\infty$ such that (i) $\{x_k\}_{k=0}^\infty$ is bounded and (ii) any accumulation point of $\{x_k\}_{k=0}^\infty$ that belongs to $\text{dom } \Psi$ must be critical.

Proof. (i). By definition, $P_0 = \mathcal{F}(x_0)$. Furthermore, Theorem 5.16 states that for any integer $k \geq 0$, $P_{k+1} \leq P_k$ thus $P_{k+1} \leq \mathcal{F}(x_0)$. Again, noticing that $\mathcal{F}(x_{k+1}) \leq P_{k+1}$ allows to conclude that $\mathcal{F}(x_{k+1}) \leq \mathcal{F}(x_0)$ hence $x_{k+1} \in S_{\mathcal{F}, P_0}$ (Assumption (C)). The boundedness of this latter sublevel set implies the boundedness of $\{x_k\}_{k=0}^\infty$. (ii). Combining fact (i) above, point (ii) from Theorem 5.16 and the arguments of [TFT22] (Theorem 6), for any accumulation point $x^* \in \text{dom } \Psi$ of $\{x_k\}_{k=0}^\infty$, there must be an infinite countable subset of indices $\mathcal{K} \subseteq \mathbb{N}$ such that for the sequence $\{g_{2,k}\}_{k=0}^\infty$ of collected subgradients of f_2 at x_k and some $g_2(x^*) \in \partial f_2(x^*)$,

$$\lim_{\bar{k} \rightarrow \infty, \bar{k} \in \mathcal{K}} x_{\bar{k}} = x^* \quad \text{and} \quad \lim_{\bar{k} \rightarrow \infty, \bar{k} \in \mathcal{K}} g_{2,\bar{k}} = g_2(x^*).$$

5 | Adaptive Sequential (DC) algorithm with extrapolation

For any $\bar{k} \in \mathcal{K}$, the optimality of $(x_{\bar{k}})_+$ as a solution of

$$\min_{u \in \mathbb{R}^d} \ell(u; \mathcal{Y}_{1,\bar{k}}) - (f_2(x_{\bar{k}}) + \langle g_{2,\bar{k}}, u - x_{\bar{k}} \rangle) + \frac{L_{\bar{k}}}{2} \|u - y_{\bar{k}}\|^2 + \Psi(u)$$

implies the existence of $q \in \Delta^{|\mathcal{Y}_{1,k}|}$ and $\hat{g}_{\bar{k}} \in \partial\Psi((x_{\bar{k}})_+)$ such that

$$\begin{aligned} \hat{g}_{\bar{k}} + \sum_{e=1}^{|\mathcal{Y}_{1,k}|} (q)_e g_{1,\bar{k}}^{(e)} &= g_{2,k} + L_{\bar{k}}(y_{\bar{k}} - (x_{\bar{k}})_+) \\ &= g_{2,k} + L_{\bar{k}} \left(\underbrace{x_{\bar{k}} - (x_{\bar{k}})_+}_{-\delta x_{\bar{k}+1}} + \underbrace{\varsigma_{\bar{k}}((x_{\bar{k}-1})_+ - x_{\bar{k}-1})}_{\delta x_{\bar{k}}} \right). \end{aligned} \quad (5.30)$$

Taking the limit as $\bar{k} \rightarrow \infty$ and recalling the boundedness of $\{x_k\}_{k=0}^\infty$, $\{L_k\}_{k=0}^\infty$ and $\{\varsigma_k\}_{k=0}^\infty$ together with Theorem 5.19 (ii), we state that $\hat{g}_{\bar{k}} + \sum_{e=1}^{|\mathcal{Y}_{1,k}|} (q)_e g_{1,\bar{k}}^{(e)}$ converges to an element of $\nabla f_1(x^*) + \partial\Psi(x^*)$ ([dO19], Lemma 3 and Proposition 2) and (5.30) converges to $g_2(x^*) \in \partial f_2(x^*)$. Therefore, x^* turns out to be a *critical point* of problem (ν_{DC}) . $\square$

5.4 Numerical Experiments

In this section, we aim at analyzing the performances of ASDCe under different settings, i.e. by enabling/disabling its features, i.e. *line-search*, *cutting-planes* models for f_1 and/or f_2 and *extrapolation*. We investigate when and how combining some (or all) of these features yields either a better (average/median) terminal objective value or a speed-up in the average computational time taken before the associated algorithm exits. For the sake of transparency, our numerical experiments are accessible on [GitHub](#).

Prior to describing how the benchmark instances of (ν_{DC}) were created, we list the different contenders. Each of which stems as a special version of ASDCe, defined by parameters $(B_{1,\max}, B_{2,\max})$, τ_{loc} and $\hat{\tau}$. When $\tau_{\text{loc}} = 1$, $L_k = L_{-1}$ for every $k \geq 0$ so that *line-search* is deactivated. In this case, L_{-1} must then be taken at least as large as $L + 2\tau_{\text{exc}}L_{\min}$ to ensure convergence. Otherwise, we pick $\tau_{\text{loc}} = 2/3$. Fixing $\hat{\tau} = 0$ skips *extrapolation*, otherwise it is simply set to 1. When $B_{2,\max} > 1$, we set it to 200 whereas if $B_{1,\max} > 1$ we chose it as $\max\{\min\{d + 5, 1000\}, 100\}$. Finally, we fix $(\epsilon, K_{\text{trig}}) = (10^{-9}, 3)$.

We also clarify how the integer K_{inner} is decided during each iteration of ASDCe. We impose that $B_{2,\text{max}} \geq K_{\text{inner}}$. It is only when $B_{2,\text{max}} > 1$ that at least a second DCA step is undertaken. Every time a new iterate $x^{(k_{\text{inner}}+1)}$ is returned we compute the model decrease $\hat{f}_{1,k}(x^{(k_{\text{inner}})}) + \Psi(x^{(k_{\text{inner}})}) - f_2(x^{(k_{\text{inner}})}) - (\hat{f}_{1,k}(x^{(k_{\text{inner}}+1)}) + \Psi(x^{(k_{\text{inner}}+1)}) - f_2(x^{(k_{\text{inner}}+1)}))$. DCA stops when this decrease falls below $5 \cdot 10^{-4}$ or when $k_{\text{inner}} \geq B_{2,\text{max}}$.

$\downarrow \text{name} \setminus \text{features} \rightarrow$	<i>extrapolation</i>	<i>line-search</i>	<i>bundle</i>	experiment
ASDCe	Yes	Yes	f_1/f_2	2
ASDC	No	Yes	f_1/f_2	2
SDCe-LS	Yes	Yes	f_2	1 & 2
SDC-LS	No	Yes	f_2	1 & 2
SDCe	Yes	No	f_2	1 & 2
SDC	No	No	f_2	1 & 2
SCe-LS	Yes	Yes	No	1 & 2
SC-LS	No	Yes	No	1 & 2
SCe	Yes	No	No	1
SC	No	No	No	1

Table 5.2 List of contenders || ASDCe variants.

5.4.1 Experiment 1 | Quadratic - Quadratic Minimization over L_1 -ball

In this first experiment, we tackle problem (L_p -TRS) for $p = 1$, $d = 10^3$ and $R = 10$. That is, with $\mathbf{U} = \mathbf{U}^T \geq \mathbf{0}_{d \times d}$,

$$F^* = \min_{x \in \mathbb{B}_1(\mathbf{0}_d; R)} \frac{1}{2} \langle \mathbf{U}x, x \rangle = \min_{x \in \mathbb{B}_1(\mathbf{0}_d; R)} \frac{1}{2} \langle \mathbf{U}_1x, x \rangle - \frac{1}{2} \langle \mathbf{U}_2x, x \rangle$$

We sampled 10 different matrices $\mathbf{U}_{\text{start}} \sim \text{Uni}([-2, 2]^{d \times d})$. Then, from the eigendecomposition of $(\mathbf{U}_{\text{start}} + \mathbf{U}_{\text{start}}^T)/2 = [\mathbf{V}_1, \mathbf{V}_2] \text{diag}(\Sigma_1, -\Sigma_2) [\mathbf{V}_1, \mathbf{V}_2]^T$ where $\mathbf{V} = [\mathbf{V}_1, \mathbf{V}_2]$ is orthogonal and $\Sigma_\star \geq \mathbf{0}_{d \times d}$ for $\star \in \{1, 2\}$, we derive $\hat{\mathbf{U}}_\star = \mathbf{V}_\star \Sigma_\star \mathbf{V}_\star^T$. In order to generate $\mathbf{U}$ with a variable level of convexity/concavity, we build three matrices $\mathbf{U}_{\text{cvx}} = \hat{\mathbf{U}}_1 - (1/3) \cdot \hat{\mathbf{U}}_2$, $\mathbf{U}_{\text{neutral}} = \hat{\mathbf{U}}_1 - \hat{\mathbf{U}}_2$ and $\mathbf{U}_{\text{ccv}} = \hat{\mathbf{U}}_1 - (5/3) \cdot \hat{\mathbf{U}}_2$ leading to three times as many problem instances. In any case, Ψ is the indicator function on $\mathbb{B}_1(\mathbf{0}_d; R)$, f_1 is the quadratic form defined by matrix $\mathbf{U}_1 = \hat{\mathbf{U}}_1$ and f_2 is also a quadratic form either defined by $\mathbf{U}_2 = \mathbf{U}_{2,\text{cvx}} = (1/3) \cdot \hat{\mathbf{U}}_2$, $\mathbf{U}_2 = \mathbf{U}_{2,\text{neutral}} = \hat{\mathbf{U}}_2$ or $\mathbf{U}_2 = \mathbf{U}_{2,\text{ccv}} = (5/3) \cdot \hat{\mathbf{U}}_2$. The smoothness parameter of f_1 is $L = \|\mathbf{U}_1\|_2$. For every problem instance, we drew 10 random starting points x_0 to initialize contenders of Table 5.2; $\hat{x}_0 \sim \text{Uni}([-3, 3]^d)$ and $x_0 = \text{proj}_{\mathbb{B}_1(\mathbf{0}_d; R)}(\hat{x}_0)$.

Results Figure 5.3 displays the outcome of every single run, i.e. objective value reached and time to exit, of any of the contenders for Experiment 1. Tables 5.3-5.5 aggregate, for every problem type (more convex (cvx), more concave (ccv) or (neutral)), all the results of the benchmarked methods over the $N_{\text{prob}} = 10$ problems under the $N_{\text{rep}} = 10$ different initializations.

As observed on Figure 5.3, overall, *extrapolation* promotes accelerated convergence for the tested algorithms. Looking more carefully at Tables 5.3-5.5, it turns out that for algorithms not exploiting *line-search*, i.e. SDC / SDCe and SC / SCe, not incorporating *extrapolation* is really detrimental in what concerns their average exit time. Conversely, when *line-search* is turned on, enabling *extrapolation* seems to slow down the empirical speed of convergence. Yet, this latter effect looks way less pronounced for the (cvx) case. It is also in the (cvx) case that the averaged objective value obtained is significantly better for the methods using *extrapolation*. When the concavity of the problem takes over the lead, *extrapolation* starts to harm the final objective value. As always, this last statement should be taken with caution, i.e. it applies in the context of this preliminary experiment about quadratic form constrained optimization. That being said, this result is not totally counter-intuitive. We recall that *extrapolation* is crafted to take advantage of *convexity* patterns. Finally, there is clear evidence that (DC) updates methods, i.e. SDCe-LS / SDCe-LS and SDC / SDCe for which $B_{2,\text{max}} = 200$, provide better objective values than their counterparts ($B_{2,\text{max}} = 1$) referred to as (C). For every setting (*extrapolation* yes/no, *line-search* yes/no), enabling (DC) updates (i.e. implicitly minimizing a regularized *difference-of-bundles* model at each iteration by taking multiple DCA inner-steps) was beneficial to reach lower objective values. However, this improvement upon (C) methods comes at a price, as (DC) methods take more time to exit due to their increased amount of exploration. About problems of type (cvx), the overall average value obtained by (DC) methods is -7.26 compared to -7.24 for (C). When it comes to the (neutral) type, the overall average value for (DC) is -90.18 vs. -89.77 for (C). Finally, for the (ccv) problems, the difference was substantial, i.e. -462.01 (DC) vs. -457.29 (C). Not surprisingly, (DC) methods were the most efficient when the concavity of the objective prevails since a *cutting-planes* model about f_2 would, in this case, better capture the behaviour of $f = f_1 - f_2 \simeq -f_2$ further away from the *stability center*.

$\downarrow name \setminus metrics \rightarrow$	avg. time [s]	avg. value	med. value	avg. iters
SDCe-LS	2.19	-7.26	-7.25	213.7
SDC-LS	1.93	-7.28	-7.24	187.9
SDCe	2.35	-7.28	-7.26	349.8
SDC	10.73	-7.23	-7.21	1810.2
SCe-LS	1.58	-7.25	-7.24	259.7
SC-LS	1.41	-7.24	-7.26	251.3
SCe	1.59	-7.27	-7.26	342.7
SC	7.88	-7.22	-7.19	1811.9

Table 5.3 Experiment 1 | $\mathbf{U}_2 = \hat{\mathbf{U}}_{2,cvx}$

$\downarrow name \setminus metrics \rightarrow$	avg. time [s]	avg. value	med. value	avg. iters
SDCe-LS	2.18	-90.50	-92.18	115.5
SDC-LS	1.08	-90.60	-92.22	42.18
SDCe	2.93	-89.71	-91.11	304.2
SDC	10.76	-89.93	-91.45	1234.4
SCe-LS	1.72	-89.96	-91.84	258.3
SC-LS	0.22	-89.58	-91.72	43.2
SCe	1.55	-89.90	-91.84	301.4
SC	6.64	-89.65	-91.11	1334.1

Table 5.4 Experiment 1 | $\mathbf{U}_2 = \hat{\mathbf{U}}_{2,neutral}$

$\downarrow name \setminus metrics \rightarrow$	avg. time [s]	avg. value	med. value	avg. iters
SDCe-LS	0.37	-462.98	-472.67	7.84
SDC-LS	0.37	-466.08	-473.26	7.93
SDCe	0.52	-457.10	-467.93	21.37
SDC	0.85	-461.87	-471.56	34.95
SCe-LS	0.09	-457.82	-469.25	12.71
SC-LS	0.08	-455.48	-467.71	12.11
SCe	0.18	-456.22	-467.71	26.03
SC	0.35	-459.65	-469.25	48.02

Table 5.5 Experiment 1 | $\mathbf{U}_2 = \hat{\mathbf{U}}_{2,ccv}$

5 | Adaptive Sequential (DC) algorithm with extrapolation

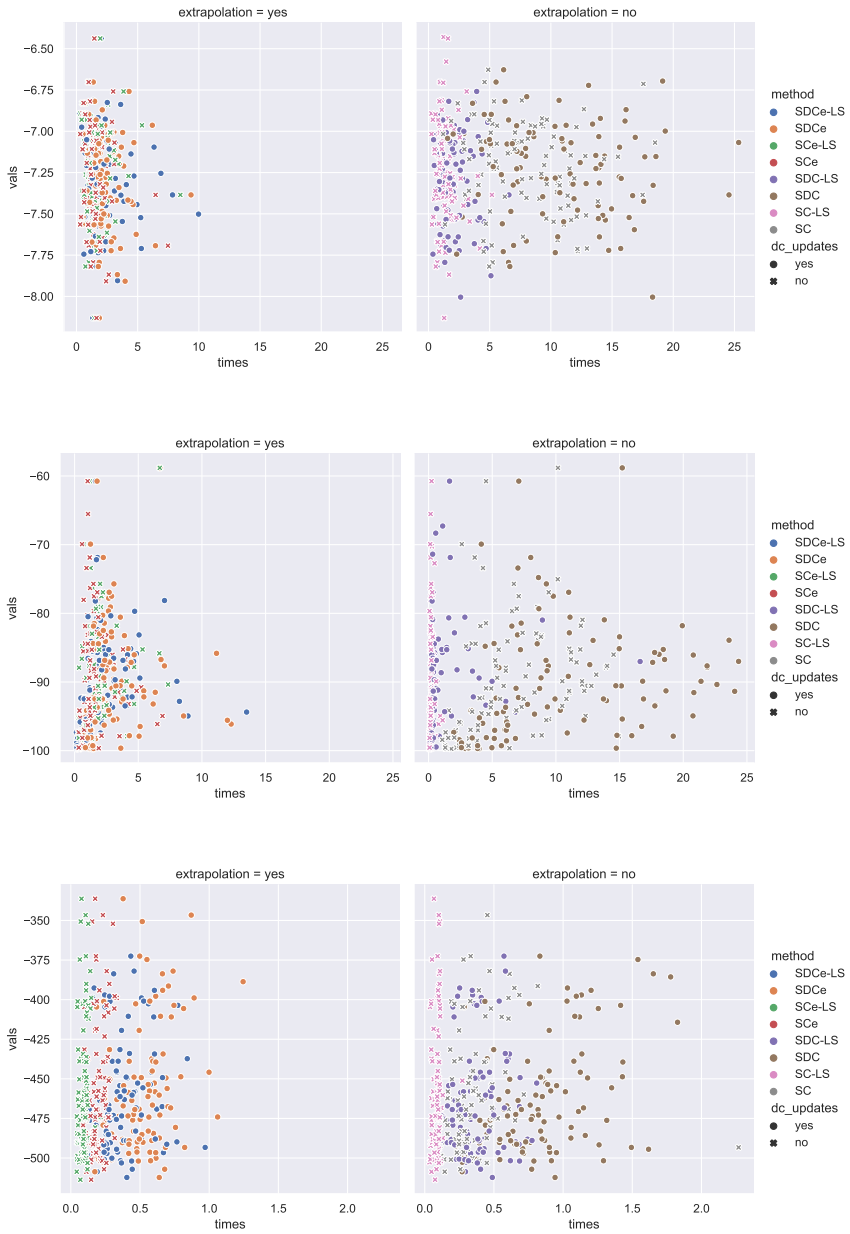


Fig. 5.3 Experiment 1: exit times and best objective values for every run l
(up) $U_2 = \hat{U}_{2, cvx}$, (mid) $U_2 = \hat{U}_{2, neutral}$, (down) $U_2 = \hat{U}_{2, ccv}$

5.4.2 Experiment 2 | LSE - Quadratic Minimization over L_1 -ball

In this second experiment, we consider the same data generation process as in Experiment 1 to build matrices $\mathbf{U}_2 = 2 \cdot \hat{\mathbf{U}}_{2,\text{neutral}}$. However, we replace the quadratic form that implemented f_1 with a Log-Sum-Exp (LSE) expression. Let $N_{\text{LSE}} = 5 \cdot 10^4$. We sampled $\beta^{(s)} \sim \text{Uni}(\{-5, 5\}^d)$ for every $s \in [N_{\text{LSE}}]$ so that $f_1(x) = \omega \cdot \log(\sum_{s=1}^{N_{\text{LSE}}} \exp(\langle \beta^{(s)}, x \rangle / \omega))$. Here, due to data's high-dimensionality, i.e. $[\beta^{(1)}, \dots, \beta^{(N_{\text{LSE}})}] \in \{-5, 5\}^{10^3 \times 5 \cdot 10^4}$, any theoretical L parameter (cfr. Example 5.1) is so large that methods from Experiment 1 not using the *line-search* feature are not competitive anymore. The *smooth maximum* structure of f_1 suggests that *cutting-planes* models, allowing to further harvest *local-smoothness*, should be tried. Therefore, ASDCe / ASDC are benchmarked. For every tested method, we set arbitrarily $L_{-1} = 10^3$. We drew x_0 as in Experiment 1 but this time, $N_{\text{prob}} = 5 = N_{\text{rep}}$.

Results According to Table 5.6, ASDCe / ASDC that mobilized bundles about f_1 (i.e. $B_{1,\max} > 1$) were the most competitive methods in terms of objective value obtained and number of iterations performed. ASDCe was the fastest with an average time of 100 [s] but, surprisingly, ASDC lagged quite a bit with its average exit time of 720 [s]. Digging into performances' data, we noticed the following. On two particular runs out of the $N_{\text{prob}} \cdot N_{\text{rep}} = 25$ in total, abnormal times were recorded, among which, for instance, an extreme $\sim 4 \cdot 10^4$ [s] time mark for one run of ASDC.¹ Looking instead at the median exit times for ASDCe / ASDC, we find respectively at 74.66 [s] and 68.33 [s] which are in phase. These times fall way below others, i.e. 145.92 [s] for SDCe-LS, 522.22 [s] for SDC-LS, 163.51 [s] for SCe-LS and, finally, 527.06 [s] for SC-LS. Here, both (DC) updates and *extrapolation* seem to have been of added value.

$\downarrow \text{name} \setminus \text{metrics} \rightarrow$	avg. time [s]	avg. value	med. value	avg. iters
ASDCe	100	-1536.3	-1549.2	37.6
ASDC	720	-1541.9	-1549.2	41.6
SDCe-LS	266	-1532.0	-1557.5	76.4
SDC-LS	2913	-1534.5	-1549.2	283.5
SCe-LS	465	-1524.2	-1544.2	79.9
SC-LS	1685	-1534.6	-1549.2	283.6

Table 5.6 Experiment 2

¹The time outliers are either due to numerical issues preventing to achieve the $\epsilon = 10^{-9}$ accuracy targeted or a CPU *pause* during the (long) computations of this second experiment.

5.4.3 Recommendations

In the light of the above results, we formulate the following recommendations. First, based on both experiments, we empirically validate the use of *line-search*. In the worst case scenario, *backtracking* could slow down the convergence of a specific variant of ASDCe if the parameter K_{trig} is too aggressive, i.e. $K_{\text{trig}} = 1$. That is, one decreases the *local-smoothness* estimate $\hat{L}$ after any successful iteration. This might lead to a high number of *backtracking* steps hence unsuccessful iterations and wasted time. However, this potential issue is mild compared to the alternative option of using an overly big theoretical L parameter (cfr. Experiment 2). In addition to faster convergence, *line-search* sometimes promotes exploration. By trying out $\hat{L} < L$ in equation (5.21), one optimizes models $\mathcal{H}$ of $\mathcal{F}$ that are not necessarily upper-models (4.12). Sometimes, this allows to visit neighbouring local convex valleys that exhibit enhanced local optima. Also, if one really looks after better solutions and is ready to spend extra computational time on the problem, (DC) updates methods (i.e. $B_{2,\max} > 1$) should be considered. This is especially true if the problem turns out to be more concave, i.e. when component f_2 's curvature dominates component f_1 's on substantial parts of the basic feasible domain $\mathcal{X}$. Whether or not setting $B_{1,\max} > 1$ is problem dependent and relies on the structure of f_1 . That being said, we certainly advise to do so when parameter L appears so big that methods with no *line-search* feature are *a priori* discarded (cfr. Experiment 2). Finally, *extrapolation* is a great feature to incorporate when tackling (v_{DC}) but its benefits are uncertain. For example, it dramatically accelerates convergence when f_1 brings *enough* convexity in the problem. Yet, *enough* is difficult to quantify beforehand. It also promotes exploration but this induces more variance in the terminal objective values obtained. We would recommend practitioners to split in two the total amount of repetitions they intend to invest in the solving of a problem. Half of the repetitions should be undertaken with *extrapolation* enabled, the other *half* should be executed without *extrapolation*.

5.5 Conclusion

In this chapter we have motivated, analyzed (including providing a convergence proof) and benchmarked our ASDCe algorithm.

To that end, we have shown that it is possible to mix together three ingredients from the convex optimization realm, i.e. *nonmonotone backtracking line-search*, *extrapolation* and *cutting-planes* models, so to use them together in a single algorithm, ASDCe. We have explained in which sense applying the celebrated DCA on a specific model $f_{1,\mathcal{M}} + \Psi - f_2$ of our objective $\mathcal{F}$ is equivalent to exactly minimizing a *regularized difference-of-bundles* model $\ell(\cdot; \mathcal{Y}_1) + \frac{\hat{L}}{2} \|\cdot - y\|_2^2 + \Psi - \ell(\cdot; \mathcal{Y}_2)$. Then, after having specified our working assumptions, we detail in Lemma 5.12 the descent guarantees that come with the minimization of such latter models. Based upon these guarantees, we further establish a sublinear rate of convergence for Algorithm 1 in Theorem 5.16. Finally, we have conducted preliminary numerical experiments validating the ASDCe approach. Based on the observed results, we gave recommendations to follow in order to best take advantage of one, two and even sometimes three (i.e. all) of the aforementioned features.

6

Upper-Lower Optimization method for (MC)

6.1 Introduction

Section 4.3.1 was meant to introduce so-called (MC) objectives, i.e. point-wise minima of finite collections of $n \in \mathbb{N}$ continuous convex functions $f^{(i)}$ for every $i \in [n]$. In this chapter, we study the minimization of such objectives but we relax the convexity assumption on the $\{f^{(i)}\}_{i \in [n]}$, called *pieces* throughout the sequel. We also consider a collection of $m \in \mathbb{N}$ functional *constraints* $\{c^{(j)}\}_{j \in [m]}$. We are concerned with *minimum structured* (MS) optimization problems of the form

$$\begin{aligned} F^* &:= \min_{x \in \mathcal{X}} \min_{i=1, \dots, n} f^{(i)}(x) \\ \text{s.t. } & c^{(j)}(x) \leq 0 \quad \forall j \in [m] \end{aligned} \tag{MS}$$

for which we assume that the optimal value is finite, i.e. $-\infty < F^* < \infty$. In (MS), one always need to select subsets of *pieces* / *constraints*. This suggests the following two notations. For any subset $S \subseteq [m]$, we introduce

$$\mathcal{D}^{(S)} := \left\{ x \in \mathbb{R}^d \mid c^{(j)}(x) \leq 0 \quad \forall j \in S \right\}, \tag{6.1}$$

the implicit domain defined by the intersection of the constraints indexed in S . Building upon this notation, we define for every $H \subseteq [n]$ the function

$$x \rightarrow F^{(H,S)}(x) := \begin{cases} \min_{i \in H} f^{(i)}(x) & x \in \mathcal{D}^{(S)} \\ \infty & x \notin \mathcal{D}^{(S)}. \end{cases} \quad (6.2)$$

$F^{(H,S)}$ represents a simplified version of the whole objective F where, instead of considering the minimum over all the n *pieces* while satisfying each of the m *constraints*, one only cares about the minimum over *pieces* indexed in H while only taking into account *constraints* in S . We can now rewrite

$$F^* = \min_{x \in \mathcal{X}} F(x) := F^{([n],[m])}(x). \quad (6.3)$$

Although non-convex and intrinsically hard, the (MS) framework allows for building successive global approximations that are either upper models or lower models of the objective function. We will assume throughout this chapter the availability of an oracle that can perform exact global minimization of any the aforementioned models $F^{(H,S)}$. More specifically, the oracle can solve any problem

$$\nu(H, S) := \min_{x \in \mathcal{X}} F^{(H,S)}(x). \quad (\text{Oracle})$$

Note that it is sufficient to assume a black-box computing for any $i \in [n]$,

$$\nu(\{i\}, S) = \min_{x \in \mathcal{X}} f^{(i)}(x) \quad \text{s.t.} \quad \max_{j \in S} c^{(j)}(x) \leq 0 \quad (6.4)$$

since it trivially implies $\nu(H, S) = \min_{i \in H} \nu(\{i\}, S)$. Obviously, calculating F^* can be achieved by calling n times this black-box as in

$$F^* = \min_{i \in [n]} \underbrace{\nu(\{i\}, [m])}_{:= \nu^{(i)}}. \quad (6.5)$$

In the absence of any further simplification, the time complexity of the above *enumeration* strategy (6.5) to solve (MS) inevitably scales linearly with n . This might become prohibitive if n is big and the efforts taken by the black-box to obtain values $\nu^{(i)}$ for every $i \in [n]$ are not negligible. For instance, when *pieces* belong to the same class and so do *functional constraints*, every black-box call is expected to spend a computational time that essentially depends on the number of constraints involved, i.e. $|S|$.

In the practical context of conic programming [NT08], this usually amounts to a polynomial of d and $|S|$ with a leading term of order $\text{poly}(d) \cdot |S|^r$ for a suitable $r > 0$, e.g. $r \in \{\frac{1}{2}, \frac{3}{2}\}$.

Naturally, one may question the possibility of avoiding the n black-box calls needed to get all the $v^{(i)}$'s while sticking to a global optimization approach for (MS). In this chapter, our aim is to satisfy the following demand:

Can we exploit the structure in (MS) to devise an exact algorithm that can certify (approximate) global optimality while requiring in practice much fewer than n computations $\{v^{(i)}\}_{i \in [n]}$ required in the baseline enumeration strategy?

We answer this question positively by proposing a new algorithm dubbed UL0 (Upper-Lower Optimization) that alternates between (a) local-search steps, which minimize upper models, and (b) suggestion steps, which optimize lower models of F . By carefully choosing which upper models to optimize in step (a), we can ensure the *local optimality* of our final output iterate. On the other hand, within our step (b), the lower models optimizations either improve global certifications about the candidates and/or suggest a new restart for step (a). In order to illustrate our framework, we detail a first example for which we point out which functions are the *pieces* and which are the *constraints*.

Example 6.1 (Optimistic Robust LP). A nominal *optimistic* program (see [SMD16] for more examples) reads as follows. Suppose one can choose an option $(\beta, \gamma) \in \mathbb{R}^p \times \mathbb{R}$ among a dataset $\{(\beta^{(i)}, \gamma^{(i)})\}_{i \in [n]}$ of possibilities. Then, depending on the retained option, one optimizes an affine cost $\langle \beta, u \rangle + \gamma$ over a polyhedral set of feasible actions $\mathcal{U} \subseteq \mathbb{R}^p$. Given a positive semidefinite matrix $\mathbf{B}$ and an allowance $R > 0$, one also needs to satisfy a quadratic budget constraint $\langle \mathbf{B}u, u \rangle \leq R$. As such, the nominal problem, already nonconvex in general, boils down to

$$\min_{i \in [n]} \min_{u \in \mathcal{U}} \langle \beta^{(i)}, u \rangle + \gamma^{(i)} \quad \text{s.t.} \quad \langle \mathbf{B}u, u \rangle \leq R. \quad (6.6)$$

One can tackle an even more challenging problem. Like [BNK09] (Section 2), we take into account possible implementation errors $\Delta_u \in \mathbb{R}^p$, i.e. instead of the chosen u vector, $u + \Delta_u$ is implemented so that the budget constraint is possibly exceeded. Let $\bar{\alpha} > 0$ describe the radius of maximal ∞ -norm perturbation so that $\Delta_u \sim \text{Uni}(\mathbb{B}_\infty(\mathbf{0}_p; \bar{\alpha}))$. We would like the budget constraint to be satisfied in any case while optimizing the expected

perturbed cost. At any fixed (β, γ) , the expected value of the objective, i.e. $\mathbb{E}_{\Delta_u}[\langle \beta, u + \Delta_u \rangle + \gamma]$, still equals $\langle \beta, u \rangle + \gamma$. The robust counterpart of (6.6) becomes

$$\min_{i \in [n]} \min_{u \in \mathcal{U}} \langle \beta^{(i)}, u \rangle + \gamma^{(i)} \quad \text{s.t.} \quad \max_{\|\Delta_u\|_\infty \leq \bar{\alpha}} \langle \mathbf{B}(u + \Delta_u), (u + \Delta_u) \rangle \leq R. \quad (6.7)$$

At fixed $u \in \mathcal{U}$, [SBTBdH22] proposes a sophisticated adjustable robust optimization (ARO) mixed-integer convex reformulation of the right-hand side of (6.7) to compute the perturbation $\Delta_u^*(u)$ effectively achieving the worst-case scenario in terms of budget. Nevertheless, since u itself is a variable, bilinear terms $\langle \mathbf{B}u, \Delta_u \rangle$ together with possible high dimensionality p prevent from establishing an efficient convex reformulation in (u, Δ_u) of the robust budget constraint. To keep things tractable, one possible approximation of (6.7) is to sample $m \in \mathbb{N}$ different scenarios $\{\Delta_u^{(j)}\}_{j \in [m]}$ among the 2^p vertices (at least one of these achieves the worst perturbation) of $\mathbb{B}_\infty(\mathbf{0}_p; \bar{\alpha})$, and to require

$$\max_{j \in [m]} \langle \mathbf{B}(u + \Delta_u^{(j)}), (u + \Delta_u^{(j)}) \rangle = \max_{j \in [m]} \langle \mathbf{B}u, u \rangle + \langle 2\mathbf{B}\Delta_u^{(j)}, u \rangle + \langle \mathbf{B}\Delta_u^{(j)}, \Delta_u^{(j)} \rangle \leq R.$$

Note that for every $m \leq 2^p$ (equality ensured if $m = 2^p$) and every $u \in \mathbb{R}^p$,

$$\max_{j \in [m]} \langle \mathbf{B}(u + \Delta_u^{(j)}), (u + \Delta_u^{(j)}) \rangle \leq \max_{\|\Delta_u\|_\infty \leq \bar{\alpha}} \langle \mathbf{B}(u + \Delta_u), (u + \Delta_u) \rangle.$$

Hence, the following (MS) problem with the extra variable $\eta \geq \langle \mathbf{B}u, u \rangle$ adequately represents the situation,

$$\begin{aligned} \min_{u \in \mathcal{U}, \langle \mathbf{B}u, u \rangle \leq \eta} \quad & \min_{i \in [n]} \langle \beta^{(i)}, u \rangle + \gamma^{(i)} \\ \text{s.t.} \quad & \eta + \langle 2\mathbf{B}\Delta_u^{(j)}, u \rangle + \langle \mathbf{B}\Delta_u^{(j)}, \Delta_u^{(j)} \rangle - R \leq 0 \quad \forall j \in [m] \end{aligned} \quad (6.8)$$

elements	expression	typical structure
d	$p + 1$	$10^2 \rightarrow 10^3$
$\mathcal{X}$	$\{(u, \eta) \in \mathcal{U} \times \mathbb{R} \mid \langle \mathbf{B}u, u \rangle \leq \eta\}$	convex
$f^{(i)}$	$\langle \beta^{(i)}, u \rangle + \gamma^{(i)}$	linear in u
$c^{(j)}$	$\eta + \langle 2\mathbf{B}\Delta_u^{(j)}, u \rangle + \langle \mathbf{B}\Delta_u^{(j)}, \Delta_u^{(j)} \rangle - R$	

Table 6.1 Optimistic Robust LP translated as an instance of (MS).

6.1.1 Related work

We emphasize that, from a practical point of view, most of the problems we can solve mobilize *pieces* and *functional constraints* that are both convex, turning the problem solved by our black-box (6.4) into a convex one. In this case, one identifies the (DC) framework of (ν_{DC}) with

$$F(x) = F_{\text{MS}}(x) = F_{\text{MC}}(x) = f_1(x) - f_2(x) = \min_{i \in [n]} f^{(i)}(x)$$

$$\Psi(x) = \begin{cases} 0 & x \in \mathcal{X} \cap \mathcal{D}^{([m])} \\ \infty & \text{otherwise.} \end{cases}$$

Some interesting machine-learning problems are (SMC) representable (cfr. Eq. (4.8)), e.g. *empirical risk minimization with clipped convex losses* [BAB20]. Such problems have been receiving increasing attention over the past years and a few related papers addressing their solving already exist. We present these works and their applications in our dedicated Chapter 7.

Although (SMC) falls in the applicability scope of (MC) hence (MS), as highlighted in Section 4.3.1, we clarify that our algorithm UL0 is not the method of choice for a very large number n of *pieces*, i.e. scaling exponentially with the number N of data observations as in machine-learning. UL0 solves problems that are **(i) exhaustively enumerable** according to the nomenclature introduced in Section 4.3. Consequently to this exponential dependency, (SMC) problems are meant to be tackled locally. On the contrary, we are focused on globality results with UL0. To the best of our knowledge, we are not aware of a comparable previous work making use of (Oracle). Yet, we will compare UL0 against two other intuitive baselines both of which also (only) relying on (Oracle). Our concern in (MS) is driven by the numerous *functional constraints* that render the evaluation of $\nu^{(i)}$ for every piece $i \in [n]$ costful, like in (6.8). By computing values $\nu(\{i\}, S)$ as in (6.4) for small-sized $S \subseteq [m]$, we produce lower bounds $\check{F}$ at reduced time expenses, possibly leading to a satisfying optimality certificate.

6.1.2 Contributions

- First, we give a valuable insight (Section 6.2.1) into (MS) together with meaningful examples (Section 6.2.2). Furthermore, we provide sufficient local optimality conditions (Section 6.2.4), tailored for (MS). We present (Section 6.3.1) and prove the exactness (Section 6.3.2) of our new algorithm

ULO, using the above conditions to elect our output iterate among visited candidates. That is, if ULO is stopped before exact global optimality, it will return a local minimum of (MS), usually hard to certify in a nonconvex nonsmooth setting.

- Second, assuming a time complexity model when using our black-box (6.4), we conduct a theoretical study (Section 6.3.3) of the overall computational cost of both the baseline Enumeration Scheme (6.5), and our algorithm ULO. To that end, we develop a stochastic directed acyclic graph (DAG) abstraction of (MS) instances whose difficulty relies on controllable parameters. By simulating ULO many times on those graph abstract problems, we identify regimes, defined by the control parameters, for which ULO is expected to outperform the baseline.
- Third, we demonstrate the validity of our approach with some numerical experiments on practical tasks (Section 6.4). We compare (emphasis put on effective wall-clock time) ULO against a pure Enumeration Scheme but also against a Restarted Alternating Minimization algorithm, based on an extension of the bi-convex formulation in [BAB20] (Section 3. § Alternating minimization).

6.1.3 Technicalities

Let $x \in \mathcal{X}$. We will often need to collect all the indices of *pieces* that achieve the minimal value among all the *pieces*, i.e. $\min_{i \in [n]} f^{(i)}(x)$. Since numerical calculations are prone to (small) computational errors, we set a tolerance $\rho \geq 0$ and look after the *pieces* whose value at x does not exceed $\min_{i \in [n]} f^{(i)}(x) + \rho$, as suggested below.

Definition 6.2 (Active sets). Let $\rho \geq 0$. The $(\rho -)$ active set at $x \in \mathcal{X}$ is

$$\mathcal{A}_\rho(x) := \left\{ i^* \in [n] \mid f^{(i^*)}(x) \leq \min_{i \in [n]} f^{(i)}(x) + \rho \right\}. \quad (6.9)$$

By convention, $\mathcal{A}(x) := \mathcal{A}_0(x)$, simply called *active set* at x .

When *pieces* are continuous on $\mathcal{X}$ (which we assume), an important feature of *active sets* resides in their nonexpansiveness in the direct neighborhood of a point $\hat{x}$, as highlighted by the next proposition.

Proposition 6.3. Let $\hat{x} \in \mathcal{X}$. There exists $\alpha > 0$ such that for any $\rho \geq 0$,

$$\mathcal{A}(x) \subseteq \mathcal{A}_\rho(\hat{x}) \quad \forall x \in \mathcal{X} \cap \mathbb{B}_2(\hat{x}; \alpha).$$

Proof. Let $\bar{h} = \min_{i \in \mathcal{A}(\hat{x})} h^{(i)}$, $\bar{h}$ is continuous since $h^{(i)}$ is continuous for every $i \in [n]$. By definition, for every $\tilde{i} \in [n] \setminus \mathcal{A}(\hat{x})$, there must exist a slack $\delta(\tilde{i}) > 0$ such that

$$h^{(\tilde{i})}(\hat{x}) = \bar{h}(\hat{x}) + \delta(\tilde{i}).$$

Noticing that the function $q^{(\tilde{i})} := h^{(\tilde{i})} - \bar{h}$ must also be continuous, it comes that there exists a strictly positive radius $R(\tilde{i}) > 0$ such that

$$q^{(\tilde{i})}(x) > 0 \quad \forall x \in \mathcal{X} \cap \mathbb{B}_2(\hat{x}; R(\tilde{i})).$$

Indeed, $q^{(\tilde{i})}(\hat{x}) = \delta(\tilde{i}) > 0$ by construction. Taking $\alpha = \min_{\tilde{i} \in ([n] \setminus \mathcal{A}(\hat{x}))} R(\tilde{i})$ leads to a neighborhood of $\hat{x}$, i.e. $\mathcal{X} \cap \mathbb{B}_2(\hat{x}; \alpha)$, on which $q^{(\tilde{i})}$ functions are all strictly positive. Hence, no index $\tilde{i} \in [n] \setminus \mathcal{A}(\hat{x})$ is such that $h^{(\tilde{i})}(x) < \bar{h}(x)$ for any x of that neighborhood. It comes then that

$$\mathcal{A}(x) \subseteq \mathcal{A}(\hat{x}) \quad \forall x \in \mathcal{X} \cap \mathbb{B}_2(\hat{x}; \alpha).$$

Finally, for any $\rho \geq 0$, it always holds $\mathcal{A}(\hat{x}) = \mathcal{A}_0(\hat{x}) \subseteq \bar{\mathcal{A}}_\rho(\hat{x})$ by definition. $\square$

In other words, there exists a ball around $\hat{x}$ over which no other *piece* than those already *active* at $\hat{x}$ (i.e. contained in $\mathcal{A}(\hat{x})$) become *active*. Subsequently, the analytical expression of F on such ball relies on the (probably reduced number of) *pieces* in $\mathcal{A}(\hat{x})$. We will formally lay down and use this fact in the proof of Lemma 6.7.

The following assumptions hold tacitly for the remaining of this chapter.

Assumption 6.4. Let $(H, S) \in [n] \times [m]$.

- One has access to an oracle computing the value

$$v(H, S) = \min_{x \in \mathcal{X}} \min_{i \in H} f^{(i)}(x) \quad \text{s.t.} \quad \max_{j \in S} c^{(j)}(x) \leq 0. \quad (\text{A})$$

6 | Upper-Lower Optimization method for (MC)

- In addition, we further assume the existence and the availability of a (finite) minimizer $x^*(H, S) \in \mathcal{X} \cap \mathcal{D}^{(S)}$ such that,

$$\min_{i \in H} f^{(i)}(x^*(H, S)) = v(H, S). \quad (\text{B})$$

Moreover, the oracle deterministically outputs $x^*(H, S)$. That is, it will always return the same $x^*(H, S)$ when provided with (H, S) as arguments.

- For every $i \in [n]$, $f^{(i)}$ is continuous and convex on its domain with

$$\mathcal{X} \subseteq \text{dom } f^{(i)}. \quad (\text{C})$$

6.2 Minimum structured optimization

We start by discussing what causes (MS) problems to be hard. We emphasize that even when *pieces* $f^{(i)}$ and *constraints* $c^{(j)}$ are smooth convex, our objective may be nevertheless nonconvex and nonsmooth in general. However, in this latter case, the source of nonconvexity is entirely explained and understood through the $\min$ operator. We recall at this stage that the objective F amounts to $F^{([n],[m])}$ as expressed in (6.3).

6.2.1 Main features

According to (6.2) and (6.3), the (natural) *domain* of F comes as follows

$$\text{dom } F = \bigcup_{i=1}^n \left(\mathcal{D}^{([m])} \cap \text{dom } f^{(i)} \right) = \mathcal{D}^{([m])} \cap \bigcup_{i=1}^n \text{dom } f^{(i)}. \quad (6.10)$$

It represents the set of points that satisfy all the *functional constraints* of (MS) and for which at least one *piece* takes a finite value. When every *piece* is continuous over the whole ambient space $\mathbb{R}^d$, we observe that $\text{dom } F = \mathcal{D}^{([m])}$. Looking back to Assumption (C), from the fact that $\mathcal{X} \subseteq \text{dom } f^{(i)}$ for every $i \in [n]$, we deduce that

$$F(x) < \infty \quad \forall x \in \mathcal{X} \cap \mathcal{D}^{([m])}. \quad (6.11)$$

We notice that (MS) can also be rewritten compactly as

$$F^* = \min_{x \in \mathcal{X} \cap \mathcal{D}^{([m])}} \min_{i \in [n]} f^{(i)}(x). \quad (6.12)$$

Equation (6.12) underlines that our problem is nothing else than a constrained optimization problem for which the feasible set $\mathcal{X} \cap \mathcal{D}^{([m])}$ might be convex but for which the objective $\min_{i \in [n]} f^{(i)}$ is usually not smooth nor convex. Even in the simple case where *pieces* and *constraints* are all affine, $\mathcal{X} \cap \mathcal{D}^{([m])}$ is convex if $\mathcal{X}$ is convex (e.g. a polyhedron if $\mathcal{X}$ is a polyhedron) but $\min_{i \in [n]} f^{(i)}$ would be piecewise-linear concave.

Without loss of generality, there exists a partition of the feasible set of (6.12) into $Y \in \mathbb{N}$ distinct regions $\{\mathcal{R}^{(e)}\}_{e \in [Y]}$ such that for every $e \in [Y]$, there is $i^*(e) \in [n]$ with

$$F(x) = \min_{i \in [n]} f^{(i)}(x) = f^{(i^*(e))}(x) \quad \forall x \in \mathcal{R}^{(e)}. \quad (6.13)$$

Nonsmoothness. Every $x \in \text{dom } F$ for which $|\mathcal{A}(x)| > 1$ represents a point where the objective is not differentiable unless active *pieces'* subdifferential sets boil down to a common singleton. When *pieces* are smooth, the *Clarke subdifferential set* reads [LSM20b]

$$\partial_C F(x) = \text{conv}(\{\nabla f^{(i)}(x) \mid i \in \mathcal{A}(x)\}). \quad (6.14)$$

If $\nabla f^{(\star)}(x) \neq \nabla f^{(\times)}(x)$ for $\star \neq \times$ indices in $\mathcal{A}(x)$ then $\partial_C F(x)$ contains more than one element thus implying the non-differentiability of F at x .

Nonconvexity. Let $v \in \mathbb{R}$, $I \subseteq [n]$, the v -sublevel set of $\min_{i \in I} f^{(i)}$ reads

$$\mathcal{S}_{I,v} = \bigcup_{i \in I} \{x \in \mathbb{R}^d \mid f^{(i)}(x) \leq v\}.$$

Thereby, the term $\min_{i \in [n]} f^{(i)}$ fails to be convex as soon as $\mathcal{S}_{[n],v}$ is not convex for a fixed value of v . For example, this happens if there exists a partition $(I_\star, I_\times)$ of $[n]$ so that $\mathcal{S}_{I_\star,v}$ and $\mathcal{S}_{I_\times,v}$ are non-empty and $\mathcal{S}_{I_\star,v} \setminus \mathcal{S}_{I_\times,v} \neq \emptyset$ is not convex. From a geometrical point of view, this would imply that there are points $x_\star \in \mathcal{S}_{I_\star,v} \subset \mathcal{S}_{[n],v}$ and $x_\times \in \mathcal{S}_{I_\star,v} \setminus \mathcal{S}_{I_\times,v} \subset \mathcal{S}_{[n],v}$ such that the segment $[x_\star, x_\times]$ is not contained in $\mathcal{S}_{[n],v}$.

6 | Upper-Lower Optimization method for (MC)

In order to clarify our notations, we illustrate on Figure 6.1 a toy example, relatively general case of (MS). Its global optimum is located at $x^* = \sqrt{3/2}$.

$$\begin{aligned} \min_{x \in \mathbb{R}} \quad & \min\{10^{-1}(x-1)^2, \max\{x-2, -2x\} - 1, 1 + \exp(x/5), 10(x+4)^2\} \\ \text{s.t.} \quad & \max\{(3/2) - x^2, x - 5, -x - 5\} \leq 0 \end{aligned} \quad (6.15)$$

One identifies $\mathcal{X} = \mathbb{R}, n = 4$ (every *piece* is convex), $m = 3$ (the two last *functional constraints* are convex but not the first one). Overall, $\text{dom } F = [-5, -\sqrt{3/2}] \cup [\sqrt{3/2}, 5]$, set that is not convex nor connected.

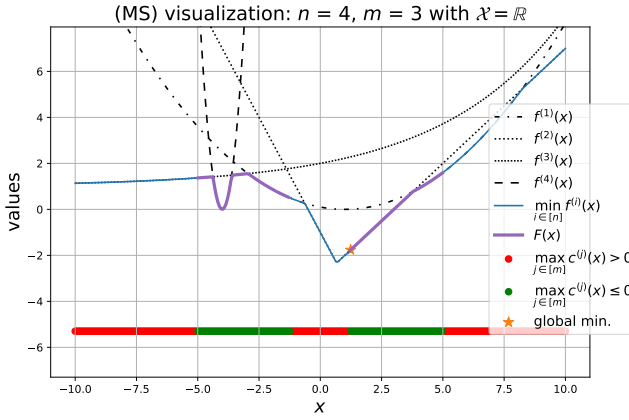


Fig. 6.1 Graph and illustration of problem (6.15).

Let us now detail two more examples that motivate the framework of (MS) and the further use of our algorithm UL0 as an efficient tool to tackle them.

6.2.2 Further examples

Every piecewise-linear objective can be rewritten as a difference between two convex piecewise affine functions [KS87]. Among others, these functions appear in recent works [SGKS20, HLP21, BTK⁺22] to fit models as cheap surrogates to complicated and potentially nonconvex general objectives. It is straightforward to reformulate the optimization of piecewise-linear objectives as instances of (MS) as shown in the next example, very similar to our introductory Example 6.1.

Example 6.5 (Piecewise-Linear objectives). Let $\{(\beta_1^{(j)}, \gamma_1^{(j)})\}_{j \in [m]}, \{(\beta_2^{(i)}, \gamma_2^{(i)})\}_{i \in [n]}$ be two collections of tuples from $\mathbb{R}^p \times \mathbb{R}$ defining a difference-of-convex (piecewise-linear) function $f : \mathbb{R}^p \rightarrow \mathbb{R}$,

$$u \rightarrow f(u) = \left[\max_{j \in [m]} \langle \beta_1^{(j)}, u \rangle + \gamma_1^{(j)} \right] - \left[\max_{i \in [n]} \langle \beta_2^{(i)}, u \rangle + \gamma_2^{(i)} \right]. \quad (6.16)$$

Using a standard epigraph technique, minimizing f on any polyhedral set $\mathcal{U} \subseteq \mathbb{R}^p$ can be equivalently stated as

$$\begin{aligned} \min_{u \in \mathcal{U}} f(u) &\equiv \min_{(u, \eta) \in \mathcal{U} \times \mathbb{R}} \min_{i \in [n]} \eta - \gamma_2^{(i)} - \langle \beta_2^{(i)}, u \rangle \\ \text{s.t.} \quad &\max_{j \in [m]} -\eta + \gamma_1^{(j)} + \langle \beta_1^{(j)}, u \rangle \leq 0. \end{aligned} \quad (6.17)$$

elements	expression	typical value
d	$p + 1$	$10^2 \rightarrow 10^4$
$\mathcal{X}$	$\mathcal{U} \times \mathbb{R}$	polyhedron
$f^{(i)}$	$\eta - \gamma_2^{(i)} - \langle \beta_2^{(i)}, u \rangle$	linear in u
$c^{(j)}$	$-\eta + \gamma_1^{(j)} + \langle \beta_1^{(j)}, u \rangle$	

Table 6.2 Piecewise-linear program translated as an instance of (MS).

A different example is the *continuous relaxation* of difficult *disjunctive programming* problems [Bal18]. That is, one lifts and strongly penalizes the least violated constraint among the set of disjunctive constraints as explained in the following example.

Example 6.6 (Relaxed Disjunctive Programming). Let $(\mathbf{B}, g) \in \mathbb{R}^{m \times p} \times \mathbb{R}^m$ be a data matrix together with linked noisy measurements. The typical L_∞ regression task asks to recover coefficients $\bar{u} \in \mathbb{R}^p$ such that $\mathbf{B}\bar{u} \simeq g$ in the sense of minimal worst deviation, i.e. with the smallest residual $\|\mathbf{B}\bar{u} - g\|_\infty$. In certain circumstances, one knows prior results based on previous regressions with large amounts of data. That is, thanks to the results of $n \in \mathbb{N}$ previous fits, one obtained a collection $\{u^{(i)}\}_{i \in [n]}$ of n *a-priori* plausible values for $\bar{u}$. The goal is then to update one of these previous models at the light of the new data $(\mathbf{B}, g)$ at hand.

6 | Upper-Lower Optimization method for (MC)

More precisely, one looks for $\bar{u}$ that best explains the aforementioned new data while keeping $\bar{u}$ at a maximal distance of $R > 0$ from at least one of the previous models $u^{(i)}$. Hence, the disjunctive model

$$\begin{aligned} \min_{u \in \mathbb{R}^p} \quad & \|Bu - g\|_\infty \\ \text{s.t.} \quad & u \in \bigcup_{i \in [n]} \mathbb{B}_2(u^{(i)}; R) \end{aligned} \quad (6.18)$$

A possible relaxation of the above allows to search for u outside the feasible set of (6.18) but penalizes the excess distance from u to the closest *a-priori* explanation $u^{(i)}$, i.e. $\min_{i \in [n]} \max\{\|u - u^{(i)}\|_2 - R, 0\}$. If there exists $i \in [n]$ such that $\|u - u^{(i)}\|_2 \leq R$ then there is no penalty. Otherwise, it amounts to $\min_{i \in [n]} \|u - u^{(i)}\|_2 - R$. To strengthen the relaxation, we multiply the penalty by a large constant $C_{\text{pen}} > 0$. For any $j \in [m]$, let $(\beta^{(j)}, \gamma^{(j)})$ denote (B, g) 's j -th observation, the problem becomes

$$\begin{aligned} \min_{(u, \eta) \in \mathbb{R}^p \times \mathbb{R}} \quad & \min_{i \in [n]} \eta + C_{\text{pen}} \cdot \max\{\|u^{(i)} - u\|_2 - R, 0\} \\ \text{s.t.} \quad & \max_{j \in [m]} -\eta + |\langle \beta^{(j)}, u \rangle - \gamma^{(j)}| \leq 0 \end{aligned} \quad (6.19)$$

elements	expression	typical value
d	$p + 1$	$10^2 \rightarrow 10^4$
$\mathcal{X}$	$\mathbb{R}^p \times \mathbb{R}$	whole domain
$f^{(i)}$	$\eta + C_{\text{pen}} \cdot \max\{\ u^{(i)} - u\ _2 - R, 0\}$	convex
$c^{(j)}$	$-\eta + \langle \beta^{(j)}, u \rangle - \gamma^{(j)} $	polyhedral convex

Table 6.3 Relaxed disjunctive program translated as an instance of (MS).

6.2.3 Upper and lower models

We derive now global models of the objective F that ultimately lead to subproblems of (MS), either yielding upper bounds $\hat{F}$ or lower bounds $\check{F}$ on the optimal value F^* . These models are not necessarily tight in the sense that there might not exist a point in the set $\mathcal{X} \cap \mathcal{D}^{([m])}$ such that the value of the models and the value of F coincide.

Upper models Let $H \subseteq [n]$. For both $F^{(H,[m])}$ and $F = F^{([n],[m])}$, all the *constraints* are included in the model. In general, $\text{dom } F \supseteq \text{dom } F^{(H,[m])}$ but equality holds, for example, if all the *pieces* in H are full-domain. In any case, at fixed $x \in \mathcal{D}^{([m])}$, one only chooses a minimal value among *pieces* in $H \subseteq [n]$ for $F^{(H,[m])}$, hence the inequality

$$F^{(H,[m])}(x) \geq F(x) \quad \forall x \in \text{dom } F^{(H,[m])}. \quad (6.20)$$

If $\mathcal{A}(x) \cap H = \emptyset$ for every $x \in \mathcal{X} \cap \mathcal{D}^{([m])}$, $F^{(H,[m])}$ is not tight since *pieces* from H are nowhere *active* on the feasible set. Minimizing both sides of (6.20) on $\mathcal{X}$ yields

$$\hat{F}(H) := \min_{x \in \mathcal{X}} F^{(H,[m])}(x) = \min_{i \in H} \nu^{(i)} \geq F^*. \quad (6.21)$$

Lower models Symmetrically to (6.20) where all the *constraints* were considered in S in the expression $F^{(H,S)}$, here one keeps all the *pieces* in H . For any $S \subseteq [m]$,

$$F^{([n],S)}(x) \leq F(x) \quad \forall x \in \text{dom } F^{([n],S)}. \quad (6.22)$$

When $x \in \mathcal{D}^{([m])}$, $F^{([n],S)}$ coincide with F in the sense that either $x \notin \cup_{i \in [n]} \text{dom } f^{(i)}$ (6.10) and $F(x) = F^{([n],S)}(x) = \infty$ or $F(x) = F^{([n],S)}(x) < \infty$ otherwise. Conversely, if $x \in \text{dom } F^{([n],S)} \cap (\mathbb{R}^d \setminus \mathcal{D}^{([m])})$, $F(x) = \infty$ whereas $F^{([n],S)}$ takes a finite value. One concludes that $\text{dom } F^{([n],S)} \supseteq \text{dom } F$. Again, minimizing both sides of (6.22) on $\mathcal{X}$ yields

$$\check{F}(S) := \min_{x \in \mathcal{X}} F^{([n],S)}(x) \leq F^*. \quad (6.23)$$

6.2.4 Optimality conditions

We end this section by proving a lemma providing a convenient sufficient condition (6.24) to deduce *local optimality* as understood in Definition 2.9.

Lemma 6.7. Let $i^* \in [n]$, $\rho \geq 0$ and $x^* \in \arg \min_{x \in \mathcal{X}} F^{(\{i^*\},[m])}(x)$. If it holds

$$\min_{x \in \mathcal{X}} F^{(\{i\},[m])}(x) = \nu^{(i)} \geq \nu^{(i^*)} = F^{(\{i^*\},[m])}(x^*) \quad \forall i \in \mathcal{A}_\rho(x^*) \quad (6.24)$$

then x^* is a local optimum of (MS).

Although sharing similarities with [dO23] (Th. 10), the above lemma differs. In [dO23] the absence of a sufficient descent direction among ρ -active

6 | Upper-Lower Optimization method for (MC)

pieces together with their local convexity is used to prove *local optimality*. Here, since we can optimize globally each *piece* individually, we require that no ρ -active neighbouring *piece* achieves a better objective than i^* 's.

Proof. Let $\rho \geq 0$ be fixed, Proposition 6.3 implies the existence of a radius $\alpha > 0$ such that $\mathcal{A}(x) \subseteq \mathcal{A}_\rho(x^*)$ for every $x \in \mathcal{X} \cap \mathbb{B}_2(x^*; \alpha)$. By Definition 6.2 of *active set*,

$$F(x) = \min_{i \in [n]} f^{(i)}(x) = \min_{i \in \mathcal{A}(x)} f^{(i)}(x) \quad \forall x \in \mathcal{X} \cap \mathcal{D}^{([m])}.$$

Combining both last statements, we can write that

$$F(x) = \min_{i \in \mathcal{A}_\rho(x^*)} f^{(i)}(x) \quad \forall x \in \mathcal{X} \cap \mathbb{B}_2(x^*; \alpha) \cap \mathcal{D}^{([m])}.$$

Moreover, we notice that

$$\begin{aligned} \min_{x \in \mathcal{X} \cap \mathbb{B}_2(x^*; \alpha)} F(x) &= \min_{x \in \mathcal{X} \cap \mathbb{B}_2(x^*; \alpha) \cap \mathcal{D}^{([m])}} F(x) \\ &= \min_{i \in \mathcal{A}_\rho(x^*)} \min_{x \in \mathcal{X} \cap \mathbb{B}_2(x^*; \alpha) \cap \mathcal{D}^{([m])}} f^{(i)}(x) \\ &= \min_{i \in \mathcal{A}_\rho(x^*)} \min_{x \in \mathcal{X} \cap \mathbb{B}_2(x^*; \alpha)} F^{(\{i\}, [m])}(x) \\ &\geq \min_{i \in \mathcal{A}_\rho(x^*)} \min_{x \in \mathcal{X}} F^{(\{i\}, [m])}(x) = \min_{i \in \mathcal{A}_\rho(x^*)} \nu^{(i)}. \end{aligned}$$

Condition (6.24) finally tells us that

$$\min_{x \in \mathcal{X} \cap \mathbb{B}_2(x^*; \alpha)} F(x) \geq F^{(\{i^*\}, [m])}(x^*) \geq F(x^*).$$

The last inequality comes from the fact that $x^* \in \text{dom } F^{(\{i^*\}, [m])}$ hence $\text{dom } F$. Since $x^* \in \mathcal{X} \cap \mathbb{B}_2(x^*; \alpha)$, we also have that

$$\min_{x \in \mathcal{X} \cap \mathbb{B}_2(x^*; \alpha)} F(x) \leq F(x^*).$$

We can conclude that x^* locally minimizes F over $\mathcal{X} \cap \mathbb{B}_2(x^*; \alpha)$. $\square$

Remark 6.8. The *piece* i^* that generated x^* actually belongs itself to $\mathcal{A}(x^*)$. Let us assume it does not, i.e. $i^* \notin \mathcal{A}(x^*)$ but the assumptions of Lemma 6.7 are satisfied. Immediately, a contradiction arises since then, in this case,

$$\nu^{(i^*)} = f^{(i^*)}(x^*) > F(x^*) = \min_{i \in \mathcal{A}(x^*)} f^{(i)}(x^*) \geq \min_{i \in \mathcal{A}(x^*)} \nu^{(i)}.$$

6.3 ULO

Prior to describe ULO, we first explain what are the observations that led us out to devise it as it is.

6.3.1 Rationale

Looking back at the motivating examples and, more broadly, at the core definition of (MS), one could ask which *functional constraints* really influence the value of F^* . Recalling Assumption (B), the minimization of any piece $f^{(i)}$ on $\mathcal{X} \cap \mathcal{D}^{([m])}$ yields a well-defined $x^*(i) := x^*({i}, [m])$. The set of *active constraints* at $x^*(i)$ is

$$\mathcal{C}(i) = \{j \in [m] \mid c^{(j)}(x^*(i)) = 0\}. \quad (6.25)$$

Removing the *passive constraints* does not alter the optimization, i.e.

$$\min_{x \in \mathcal{X} \cap \mathcal{D}^{([m])}} f^{(i)}(x) = v^{(i)} = \min_{x \in \mathcal{X} \cap \mathcal{D}^{(\mathcal{C}(i))}} f^{(i)}(x). \quad (6.26)$$

The naive enumeration visits the candidates $x^*(i)$ for $i = 1, \dots, n$ without keeping in memory which *constraints* were active at previously computed $x^*(1), \dots, x^*(i-1)$. A global minimum $x^* = x^*(i^*)$ of (MS) is found among $\{x^*(i)\}_{i=1}^n$ for a certain $i^* \in [n]$. Two favourable situations can occur at this stage. When constraints are fairly different from each other, e.g. affine and linearly independent, most likely only a small subset of them will be active at a given $x^*(i)$. For instance, if the problem defining $x^*(i)$ is an LP, usually $|\mathcal{C}(i)| \in \Theta(d)$ [LY21], which can be significantly smaller than m . It could also happen that sets $\mathcal{C}(i)$ share a substantial part of their indices in common, lowering considerably the cardinality of their union (6.27). Based on both these statements, if one was able to detect the overall union of *active constraints* sets

$$\bar{\mathcal{C}} := \bigcup_{i=1}^n \mathcal{C}(i), \quad (6.27)$$

then even the baseline enumeration scheme efficiency could be enhanced in the sense that every $x^*(i)$ could be obtained for less computations

$$x^*(i) = \arg \min_{x \in \mathcal{X} \cap \mathcal{D}^{(\bar{\mathcal{C}})}} f^{(i)}(x) = x^*({i}, \bar{\mathcal{C}}).$$

In this work, we intend to iteratively build a set $S \subseteq [m]$ of indices that will serve as surrogate to $\bar{\mathcal{C}}$. Roughly speaking, the minimization of upper models (6.20) will give us new candidates $x^*(\bar{i})$ with new active indices $\mathcal{C}(\bar{i})$ to be added to our set S . Furthermore, we also keep track of visited indices in $H \subseteq [n]$ to avoid testing a *piece* $i \in [n]$ twice. The minimization of lower models (6.22) includes a minimization pass over every *piece* not in H but only taking into account constraints belonging to set S , ideally close from set $\bar{\mathcal{C}}$. By minimizing both upper and lower models, we obtain accordingly upper and lower bounds on F^* that hopefully allow us to skip testing most of the *pieces* in the enumeration phase.

6.3.2 Correctness

Now that the rationale behind UL0 has been enlightened, we detail our algorithm. We start by presenting its pseudo-code in Algorithm 2. It takes as arguments a *piece* index $\hat{i} \in [n]$ to start with as well as two parameters $\tilde{\epsilon} \geq 0$ and $\epsilon \geq 0$, respectively relative and absolute approximate globality tolerances. As previously announced, each outer-iteration of UL0 involves two successive distinct steps, namely phase (a) (Algorithm 3) and phase (b) (Algorithm 4), that we thoroughly explain and analyze. Each time phase (a) is exited, $\hat{F}$ (smallest upper bound computed so far) and the output iterate x are updated whereas at the end of phase (b), $\check{F}$ (biggest lower bound computed so far) is revised and, possibly, approximate global optimality is detected according to the gap $\hat{F} - \check{F}$. After a maximum of n (respectively m) updates, $\hat{F}$ (respectively $\check{F}$) must be equal to F^* , ultimately pushing the gap $\hat{F} - \check{F}$ towards 0.

Algorithm 2 UL0($\hat{i}, \tilde{\epsilon}, \epsilon$)

Require: $\rho \geq 0, k = 0, H_0 = \emptyset = S_0, (\hat{F}_0, \check{F}_0) = (-\infty, \infty), \tilde{\epsilon} \geq 0, \epsilon \geq 0$.

Ensure: x_k local minimum s.t. $F(x_k) - F^* \leq \max\{\epsilon, \max\{1, |F(x_k)|\} \cdot \tilde{\epsilon}\}$.

```

1: while  $\hat{F}_k - \check{F}_k > \max\{\epsilon, \max\{1, |\hat{F}_k|\} \cdot \tilde{\epsilon}\}$  do
2:    $(i_k^*, H_{k+1}, \hat{F}_{k+1}, x_{k+1}) \leftarrow \text{phase-(a)}(\hat{i}_k, H_k, \hat{F}_k, x_k)$ 
3:    $(\hat{i}_{k+1}, S_{k+1}, \check{F}_{k+1}) \leftarrow \text{phase-(b)}(i_k^*, S_k, \check{F}_k, H_k)$ 
4:   if  $\check{F}_{k+1} = \hat{F}_{k+1}$  then return  $x_{k+1}$  ▷ exact global optimality
5:   else
6:      $k \leftarrow k + 1$ 
7:   end if
8: end while
9: return  $x_k, \hat{F}_k, \check{F}_k$ 

```

We dig now into phases (a) & (b) specifically. For the sake of clarity, k subscripts (the outer-iteration count) were dropped within pseudo-codes.

Phase (a) From input $V \subseteq [n] \setminus H$, one selects uniformly at random the index $\bar{i}$ of a *piece* not tested yet. According to Assumption 6.4 (B), this leads to a well-defined new candidate $\tilde{x} = x^*(\bar{i}) = x^*(\{\bar{i}\}, [m])$ as the result of the minimization of $F^{\{\bar{i}\}, [m]}$ over $\mathcal{X}$ or, equivalently, the minimization of $f^{(\bar{i})}$ over $\mathcal{X} \cap \mathcal{D}^{([m])}$. The set of tested pieces, H , is then updated accordingly and we start over this process until there is **no new** reachable *piece* from $\bar{x}$ (output iterate of phase (a)), i.e. $\mathcal{A}_\rho(\bar{x}) \setminus H = \emptyset$. We accept a new candidate if $\nu^{(\bar{i})}$ (cfr. (6.5)) is smaller than R (obj. value achieved by the last accepted candidate $\bar{x}$). R is initialized at ∞ to make the first inner-iteration pass.

Algorithm 3 | phase-(a) ($\hat{i}, H, \hat{F}, x$)

Require: set of visited *pieces* H , starting *piece* $\hat{i} \notin H$, x best *local minimum* encountered so far (or initial iterate), $\hat{F}$ upper bound on F^* .

Ensure: i^* s.t. no $\bar{i} \in \mathcal{A}_\rho(\bar{x})$ with $\nu^{(\bar{i})} < F(\bar{x})$, the sound update of $(H, \hat{F}, x)$.

```

1:  $(H_+, V, R) \leftarrow (H, \{\hat{i}\}, \infty)$  ▷ init.
2: while  $V \neq \emptyset$  do
3:    $\bar{i} \sim \text{Uni}(V)$ 
4:    $(\tilde{x}, H_+) \leftarrow (x^*(\bar{i}), H_+ \cup \{\bar{i}\})$ 
5:   if  $\nu^{(\bar{i})} \geq R$  then  $V \leftarrow V \setminus \{\bar{i}\}$ 
6:   else
7:      $(\bar{x}, V, R, i^*) \leftarrow (\tilde{x}, \mathcal{A}_\rho(\tilde{x}) \setminus H_+, F(\tilde{x}), \bar{i})$ 
8:   end if
9: end while
10: if  $F(\bar{x}) < \hat{F}$  then  $(\hat{F}, x) \leftarrow (F(\bar{x}), \bar{x})$  ▷ update of  $\hat{F}$ 
11: end if
12: return  $(i^*, H_+, \hat{F}, x)$ 
```

When a new step is taken within the **while** loop of phase (a), a *piece* $\bar{i} \in \mathcal{A}_\rho(\bar{x})$ is tested. We show in Lemma 6.9 that its value $\nu^{(\bar{i})}$ cannot exceed $F(\bar{x})$ by more than ρ .

Lemma 6.9. Let $\bar{i} \in \mathcal{A}_\rho(\bar{x})$ for any $\bar{x} \in \mathcal{X} \cap \mathcal{D}^{([m])}$ and $\rho \geq 0$. Then we have

$$F(x^*(\bar{i})) \leq \nu^{(\bar{i})} = f^{(\bar{i})}(x^*(\bar{i})) \leq f^{(\bar{i})}(\bar{x}) \leq F(\bar{x}) + \rho. \quad (6.28)$$

Proof. By definition of ρ -active set at $\bar{x}$, $f^{(\bar{i})}(\bar{x})$ is at most ρ above the value

6 | Upper-Lower Optimization method for (MC)

$\min_{i \in [n]} f^{(i)}(\bar{x}) = F(\bar{x})$. Then, since $x^*(\tilde{i})$ realizes the minimum of *piece* $\tilde{i}$ over $\mathcal{X} \cap \mathcal{D}^{([m])}$, one must have $\nu^{(\tilde{i})} = f^{(\tilde{i})}(x^*(\tilde{i})) \leq f^{(\tilde{i})}(\bar{x})$. Finally, one observes that $f^{(\tilde{i})}(x^*(\tilde{i})) \geq \min_{i \in [n]} f^{(i)}(x^*(\tilde{i})) = F(x^*(\tilde{i}))$ to conclude. $\square$

Exiting `while` loop of phase (a), one among two possible outcomes occurs:

- $F(\bar{x}) < \hat{F}$. In this case, we can conclude that $\bar{x}$ is a *local minimum* of (MS). Indeed, we know that $\bar{x} = x^*(i^*)$ for some $i^* \in H_+ \setminus H$. Let's assume *ad absurdum* that $\bar{x}$ is not a *local minimizer*. As a consequence of Proposition 6.3, there exists $\alpha > 0$ such that on $\mathcal{X} \cap \mathbb{B}_2(\bar{x}; \alpha) \cap \mathcal{D}^{([m])}$, $F = \min_{i \in \mathcal{A}_\rho(\bar{x})} f^{(i)}$. Since $\bar{x}$ is not *locally optimal*, there must be a *piece* $\tilde{i} \in \mathcal{A}_\rho(\bar{x})$ whose minimum over any $\mathcal{X} \cap \mathbb{B}_2(\bar{x}; \alpha) \cap \mathcal{D}^{([m])}$ strictly falls below $F(\bar{x})$. Thereby, it would also mean that $\nu^{(\tilde{i})} < F(\bar{x})$ since $\nu^{(\tilde{i})}$ is the optimal value of $f^{(\tilde{i})}$ over $\mathcal{X} \cap \mathcal{D}^{([m])}$. Then, either $\tilde{i} \in H$ and, by construction, one would have had $\hat{F}_k \leq F(x^*(\tilde{i})) < F(\bar{x})$ (contradiction) or $\tilde{i} \notin H$ but then $x^*(\tilde{i})$ should have been visited ($\tilde{i} \in \mathcal{A}_\rho(\bar{x})$) and selected ($\nu^{(\tilde{i})} < F(\bar{x})$) so that $\bar{x}$ could not be the output iterate of phase (a) (contradiction). As pointed out in Remark 6.8, it also holds $\nu^{(i^*)} = F(\bar{x})$.

- $F(\bar{x}) \geq \hat{F}$. In this case, $\bar{x}$ might possibly be a *local minimum* of (MS) but we already recorded a candidate x associated with a better F value.

To summarize, prior to entering phase (b), UL0 ensures that output iterate x_k is the best *local minimum* visited so far and $\hat{F}_k = F(x_k) = \min_{i \in H_k} \nu^{(i)}$. Obviously, since $|H_{k+1} \setminus H_k| \geq 1$, there can be at most n passes through phase (a) in total during an execution of Algorithm 2. Hence there must be exist $\hat{k} \in [n]$ such that $\hat{F}_{\hat{k}} = F^*$.

Phase (b) Once the set of considered *constraints* S_+ is enriched with *active constraints* $\mathcal{C}(i^*)$ at the output iterate $\bar{x}$ of phase (a) as well as with a random $\tilde{j} \in [m] \setminus S$ (hence $S \subsetneq S_+$), we look after a new *piece* $\hat{i}$ to restart phase (a). To that end, we compute the values $\check{\nu}^{(\hat{i})}$ for every $i \in [n]$, each of which representing a lower bound on $\nu^{(i)}$, as highlighted hereafter:

$$\nu^{(i)} \geq \check{\nu}^{(i)} = \begin{cases} \nu^{(i)} & i \in H_+ \\ \nu(\{i\}, S_+) & i \notin H_+. \end{cases} \quad (6.29)$$

Then, we select $\hat{i} \in [n]$ as the most promising *piece*, i.e. $\hat{i} = \arg \min_{i \in [n]} \check{\nu}^{(i)}$, achieving the lower bound $\min_{i \in [n]} \check{\nu}^{(i)} \leq F^*$.

Algorithm 4 | phase-(b) ($i^*, S, \check{F}, H_+$)

Require: i^* output *piece* from phase (a), S set of previously stored *active constraints*, $\check{F}$ lower bound on F^* , H_+ set of visited *pieces* (i.e. $v^{(i)}$ is known for every $i \in H_+$).

Ensure: $\hat{i} \notin H_+$ such that

- 1: $\tilde{j} \sim \text{Uni}([m] \setminus S)$
 - 2: $S_+ \leftarrow S \cup \mathcal{C}(i^*) \cup \{\tilde{j}\}$ $\triangleright$ **init.**
 - 3: $\hat{i} = \arg \min_{i \in [n]} \check{v}^{(i)} = \begin{cases} v^{(i)} & i \in H_+ \\ v(\{i\}, S_+) & i \notin H_+ \end{cases}$
 - 4: $\check{F} \leftarrow \max\{\check{F}, \check{v}^{(\hat{i})}\}$ $\triangleright$ update of $\check{F}$
 - 5: **return** $(\hat{i}, S_+, \check{F})$
-

At the end of phase (b), one can face the following scenario: $\hat{i}_{k+1} \in H_{k+1}$. Recalling that $\hat{F}_{k+1} = \min_{i \in H_{k+1}} v^{(i)}$, we can write

$$F^* \geq \min_{i \in [n]} \check{v}^{(i)} = \check{v}^{(\hat{i}_{k+1})} = v^{(\hat{i}_{k+1})} \geq \hat{F}_{k+1} \geq F^* \Rightarrow x_{k+1} \in \arg \min_{x \in \mathcal{X}} F(x).$$

Note that for every $\hat{i}_{k+1} \in [n]$, one can safely update $\check{F}_{k+1} = \max\{\check{F}_k, \check{v}^{(\hat{i}_{k+1})}\}$.

In summary, either ULO certifies that x_{k+1} is *globally optimal* or phase (a) is launched again with starting *piece* $\hat{i}$. $\check{F}_{k+1}$ is monotonically increasing. Since $|S_{k+1} \setminus S_k| \geq 1$, there can be at most m phases (b). That is, there exists $\check{k} \in [m]$ such that $S_{\check{k}+1} = [m]$ and

$$\check{F}_{\check{k}+1} = \min_{i \in [n]} \check{v}^{(i)} = \min_{i \in [n]} v(\{i\}, S_{\check{k}+1}) = \min_{i \in [n]} v^{(i)} = F^*.$$

Illustration Let us illustrate the behaviour of ULO on an example, closely related to Example 6.5. The original problem is a Tikhonov-regularized piecewise-linear program

$$\min_{|u| \leq 5} \frac{u^2}{2} + \left[\max_{j \in [6]} \beta_1^{(j)} \cdot u + \gamma_1^{(j)} \right] - \left[\max_{i \in [6]} \beta_2^{(i)} \cdot u + \gamma_2^{(i)} \right] \quad (6.30)$$

with $(\beta_1^{(1)}, \dots, \beta_1^{(6)}) = (\frac{1}{4}, -\frac{1}{2}, \frac{1}{3}, 2, 0, 3)$, $(\gamma_1^{(1)}, \dots, \gamma_1^{(6)}) = (-2, -1, 0, -2, -\frac{1}{4}, -4)$ and $(\beta_2^{(1)}, \dots, \beta_2^{(6)}) = (\frac{3}{2}, 1, -1, 4, -2, 0)$, $(\gamma_2^{(1)}, \dots, \gamma_2^{(6)}) = (0, 2, 1, -1, 1, 2)$.

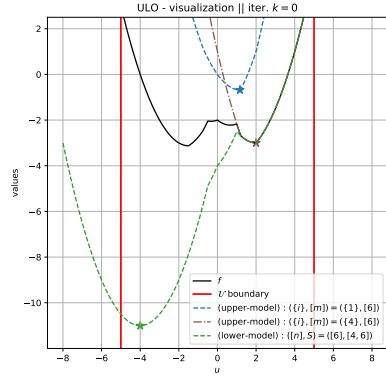
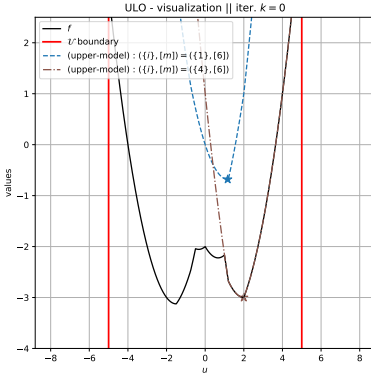
Translated into the (MS) framework, the problem now reads

6 | Upper-Lower Optimization method for (MC)

$$\min_{(u,\eta) \in \mathbb{B}_\infty(0;5) \times \mathbb{R}} \min_{i \in [6]} \left\{ \frac{u^2}{2} + \eta - \beta_2^{(i)} \cdot u - \gamma_2^{(i)} \right\} \text{ s.t. } \max_{j \in [6]} \left\{ \beta_1^{(j)} \cdot u + \gamma_1^{(j)} - \eta \right\} \leq 0$$

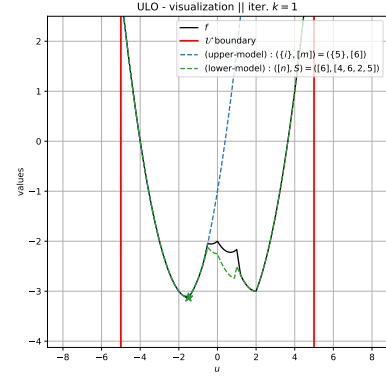
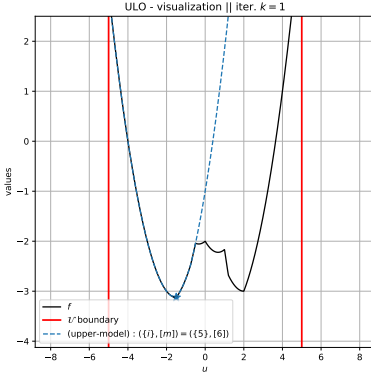
We display on Figure 6.3 two iterations of ULO applied on the above problem. The values of different models are computed on

$$\mathcal{R} = \{(u, \eta) \mid \eta = \max_{j \in [6]} \beta_1^{(j)} \cdot u + \gamma_1^{(j)}, |u| \leq 8\}.$$



(a) $H_1 = \{1, 4\}, \bar{x}_1 = (2, 2), \hat{F}_1 = -3$

(b) $S_1 = \{4, 6\}, \check{x}_1 = (-4, -10), \check{F}_1 = -11$



(a) $H_2 = \{1, 4, 5\}, \bar{x}_2 = (-3/2, -1/4), \hat{F}_2 = -25/8$

(b) $S_2 = \{2, 4, 5, 6\}, \check{x}_2 = (-3/2, -1/4), \check{F}_2 = -25/8$

Fig. 6.3 Representation of $f(u) = F(u, \eta)$ and various iterates/models for every $(u, \eta) \in \mathcal{R}$. $\hat{F}_2 - \check{F}_2 = 0$ certifies that $x_2 = \bar{x}_2$ is globally optimal.

We now state the main theorem characterizing the convergence of ULO.

Theorem 6.10. Let $\tilde{\epsilon} \geq 0$, $\epsilon \geq 0$ be prescribed accuracies and let $\hat{i} \in [n]$ be any starting piece to test. $\text{ULO}(\hat{i}, \tilde{\epsilon}, \epsilon)$ returns a point $x_K \in \mathcal{X}$ such that

$$F(x_K) - F^* \leq \max\{\epsilon, \max\{1, |F(x_K)|\} \cdot \tilde{\epsilon}\} \quad (6.31)$$

in $K \leq \min\{n, m + 1\}$ iterations, each of which involving a finite number of oracle calls. Moreover, x_K is a local optimum of (MS).

Proof. We have already proven that x_k is maintained as the best *local minimum* encountered across the iterations and therefore, as suggested by Remark 6.8, $\hat{F}_k = F(x_k)$. By construction, $\check{F}_k \leq F^*$ for any $k \in \mathbb{N}$ and thereby,

$$F(x_k) - F^* = \hat{F}_k - F^* \leq \hat{F}_k - \check{F}_k.$$

We have shown that sequence $\{\hat{F}_k\}_{k \in \mathbb{N}}$ (respectively $\{\check{F}_k\}_{k \in \mathbb{N}}$) was monotonically decreasing (respectively increasing) so that the optimality gap is monotonically decreasing with $k \in \mathbb{N}$. Moreover, we noticed that in any case, $\hat{F}_k = F^*$ for every $k \geq n$ and $\check{F}_k = F^*$ for every $k \geq m$. Considering the first case, if $k = n$, it is certain that $\check{F}_n = F^* = \min_{i \in [n]} \nu^{(i)}$ since $H_n = [n]$. On the other hand, as previously mentioned, it is sure that $S_k = [m]$ for $k \geq m$. In such circumstance, $\check{F}_k = F^*$ since for any $i \notin H_k$, $\check{\nu}^{(i)} = \nu(\{i\}, S_k) = \nu(\{i\}, [m]) = \nu^{(i)}$ (and, by definition, $\check{\nu}^{(i)} = \nu^{(i)}$ for every $i \in H_k$) thus $\check{F}_k = \min_{i \in [n]} \check{\nu}^{(i)} = \min_{i \in [n]} \nu^{(i)} = F^*$. Yet, is still possible that $F(x_k) = \hat{F}_k$ prevents ULO to exit at this iteration. Nevertheless, the suggested piece $\hat{i} = \arg \min_{i \in [n]} \check{\nu}^{(i)} \notin H_k$ will lead to the next iterate x_{k+1} being *globally optimal*. Hence the closure of the optimality gap is guaranteed after $\min\{n, m + 1\}$ iterations. The estimated gap $\hat{F}_k - \check{F}_k$ can fall below any level $\delta = \max\{\epsilon, \min\{1, |F(x_k)|\} \cdot \tilde{\epsilon}\}$ of approximate global optimality in $K < \min\{n, m + 1\}$ iterations. $\square$

6.3.3 DAG interpretation

Let $\rho \geq 0$ be fixed. Throughout this section, we assume for simplicity

$$i_+ \in \mathcal{A}_\rho(x^*(i)) \wedge i_+ \neq i \Rightarrow \nu^{(i_+)} < F(x^*(i)) \leq \nu^{(i)} \quad \forall (i, i_+) \in [n]^2. \quad (6.32)$$

We remind that Lemma 6.9 implies $\nu^{(i_+)} \leq f^{(i_+)}(x^*(i)) \leq F(x^*(i)) + \rho \leq \nu^{(i)} + \rho$. That is, we consider that a piece i_+ (nearly) achieving the minimum value among the n pieces at $x^*(i)$ (i.e. $F(x^*(i))$), when optimized itself on the feasible set $\mathcal{X} \cap \mathcal{D}([m])$, yields $\nu^{(i_+)}$ strictly improving with respect

to $F(x^*(i))$. Bearing this in mind, one can build a simple *directed acyclic graph* $G([n], E)$ that summarizes what are the possible transitions from one candidate i to another i_+ within UL0. Its vertices are the *pieces* $i \in [n]$ with values $\nu^{(i)}$ and its set of edges E is defined as

$$E := \{(i, i_+) \in [n]^2 \mid i_+ \in \mathcal{A}_\rho(x^*(i)) \wedge i_+ \neq i\}. \quad (6.33)$$

One can observe that, indeed, (6.32) and (6.33) prevent G from containing cycles. If a node i_+ is accessible from i in G then $F(x^*(i_+)) \leq \nu^{(i_+)} < F(x^*(i))$. Then, it is impossible to trace back a path from i_+ to i following vertices $\tilde{i}$ with values $F(x^*(\tilde{i})) < F(x^*(i_+))$. Let $W \subseteq [n]$. Let $G[W]$ depict the subgraph of G induced by vertices in W . For every node i of graph $G[W]$, we denote $\text{outdeg}(i) = |\{i_+ \in W \mid (i, i_+) \in E\}|$.

Starting from a node $\hat{i} \in [n]$, phase (a) performs a random walk in $G[[n] \setminus H]$, until a **sink node** i^* is reached (i.e. $i^* \in [n]$ with $\text{outdeg}(i^*) = 0$), see Figure 6.4 & 6.5. At the end of phase (a), H is augmented and now includes all the nodes visited alongside the directed path $(\hat{i} \rightarrow i^*)$. After phase (b), a new random walk is undertaken with the elected $\hat{i} \in [n] \setminus H$ and the overall procedure is run again.

Example 6.11. In order to substantiate the present section, we expose the situation below with $n = 9$ and ρ fixed in $[0, 1/10]$. As usual, $\nu^{(i)} = f^{(i)}(x^*(i))$ for every $i \in [n]$.

nodes i	$\mathcal{A}_\rho(x^*(i))$	values of <i>active pieces</i> at $x^* = x^*(i)$	$\nu^{(i)}$
1	$\{1\}$	$f^{(1)}(x^*) = 5/2$	$5/2$
2	$\{4\}$	$f^{(4)}(x^*) = 5/2$	$16/5$
3	$\{3\}$	$f^{(3)}(x^*) = 2$	2
4	$\{3, 5\}$	$f^{(3)}(x^*) = 8/3, f^{(5)}(x^*) = 8/3$	3
5	$\{5\}$	$f^{(5)}(x^*) = 7/3$	$7/3$
6	$\{4, 9\}$	$f^{(4)}(x^*) = 9/2, f^{(9)}(x^*) = 9/2 - \rho/3$	6
7	$\{1, 2, 3\}$	$f^{(1)}(x^*) = 4, f^{(2)}(x^*) = 4, f^{(3)}(x^*) = 4$	6
8	$\{8\}$	$f^{(8)}(x^*) = 5/2$	$5/2$
9	$\{9\}$	$f^{(9)}(x^*) = 4$	4

Table 6.4 Key characteristics of a possible (MS) problem | $F^* = \nu^{(3)} = 2$

From the above table, we can draw diagrams of Figure 6.4 & 6.5.

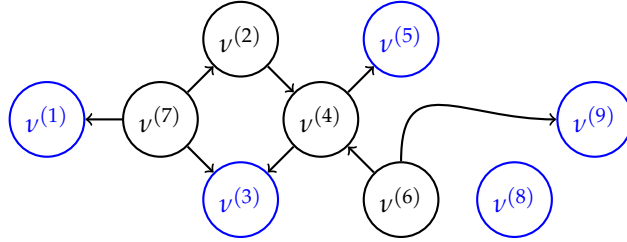


Fig. 6.4 initial graph $\mid G[[n] \setminus \emptyset] = G([n], E)$.

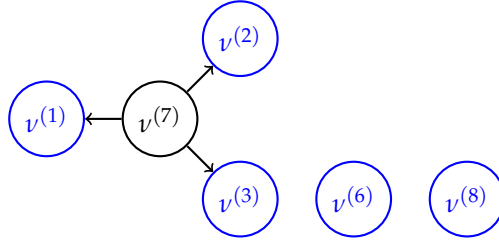


Fig. 6.5 graph after two walks: $\{4, 5\}, \{9\} \mid G[[n] \setminus H]$ with $H = \{4, 5, 9\}$.

Remark 6.12. No edge $(1, 8)$ is present since $v^{(8)} = f^{(8)}(x^*(8)) = 5/2 \nless F(x^*(1)) = 5/2$. By (6.32), $f^{(8)}(x^*(1)) > 5/2 + \rho$ because otherwise, $8 \in \mathcal{A}_\rho(x^*(1))$ but then one would have had $v^{(8)} < 5/2$.

6.3.4 Efficiency simulations

As previously teased in our introduction (Section 6.1), the purpose of this section is to analyze, *a priori*, what are the parameters, characterizing the structure and inherent difficulty of an instance of (MS), for which ULO is expected to outperform the baseline, i.e. the pure enumeration strategy as in (6.5). Therefore, we need first to state another assumption regarding our oracle time complexity. This latter will enable us to describe, at least from an abstract theoretical point of view, what are the total times taken by both algorithms since they essentially rely on successive oracle calls.

Assumption 6.13. Let $i \in [n]$ and $S \subseteq [m]$.

There exists a strictly increasing function

$$T^{(i)} : [m] \rightarrow \mathbb{R}_+ \quad (\text{D})$$

that outputs the wall-clock time $T^{(i)}(|S|)$ taken by our black-box oracle (6.4) to return $x^*(\{i\}, S)$.

Remark 6.14. Through Assumption (D), we implicitly consider that the computational burden to compute $x^*(\{i\}, S)$ is only influenced by the selected *piece* $i \in [n]$ and the number of *constraints* kept in $S \subseteq [m]$. The value of $T^{(i)}$ definitely depends on the solvers at hand, e.g. active-set methods [Won11], simplex methods [ST04], interior-point methods [NT08], etc. Yet, the nature of *pieces*, e.g. whether $\{(x, \nu) \in \mathcal{X} \times \mathbb{R} \mid f^{(i)}(x) \leq \nu\}$ is a polyhedral cone, second-order cone, exponential cone, etc., plays the most significant role in the definition of $T^{(i)}$'s once the solver has been chosen.

At this stage, the total time taken by UL0 (stopping after $K \in \mathbb{N}$ iterations) and the `enumeration` strategy, solely based on Assumption (D), read:

Algorithm	Enumeration	UL0
Running time	$\underbrace{\sum_{i \in [n]} T^{(i)}(m)}_{\text{full enumeration}}$	$\underbrace{\sum_{i \in H_K} T^{(i)}(m)}_{\text{phase (a)}} + \sum_{k=1}^K \underbrace{\sum_{i \in [n] \setminus H_k} T^{(i)}(S_k)}_{\text{phase (b)}}$
Guarantees	<i>global</i> optimality	ϵ <i>global</i> or $\tilde{\epsilon}$ <i>relative</i> optimality gap (6.31)

Table 6.5 General time complexities of UL0 and `enumeration` strategy.

For every $k \in \{0, \dots, K-1\}$, let $\Delta H_{k+1} := H_{k+1} \setminus H_k$ contain the new *pieces* tested at iteration k and let i_k^* (respectively $\tilde{j}_k$) be the output *piece* of phase (a) (respectively the drawn *constraint* index $\tilde{j}$, see line 1 of Algorithm 4) during that same iteration. As such, without further assumptions on functions $\{T^{(i)}\}_{i \in [n]}$, Table 6.5 is not yet informative enough to predict situations, symbolized by ΔH_k and $S_k = \bigcup_{\tilde{k}=0}^{k-1} \mathcal{C}(i_{\tilde{k}}^*) \cup \{\tilde{j}_{\tilde{k}}\}$ for every $k \in [K]$, for which the expected time complexity of UL0 would be the smallest.

Therefore, we present Assumption (E), narrowing down the scope of possible interpretations for $\{T^{(i)}\}_{i \in [n]}$. As suggested in our introduction and supported by [NN94], we introduce a simple complexity model mirroring interior-point methods efficiency [NT08] since the use of these latter prevails in our numerical experiments.

Assumption 6.15. There exists constants $r > 0$ and $C > 0$ such that

$$T^{(i)}(|S|) = T(|S|) = C + |S|^r \quad \forall i \in [n]. \quad (\text{E})$$

Remark 6.16. Constant C serves to better capture the practical behaviour of numerical solvers, usually involving a problem-dependent fixed computational cost to store data in memory and/or encode high-level problem's specification as workable data structures (e.g. matrices of a standard conic formulation, see *matrix stuffing* [WDHB16]).

Practical considerations Although not mentioned textually, for the sake of clarity, within the pseudo-code of Algorithm 2, it is in practice worth allowing UL0 to exit earlier as soon as the update of $\hat{F}$ yields a certificate that $\hat{F} - \tilde{F} \leq \max\{\epsilon, \max\{1, |\hat{F}|\} \cdot \tilde{\epsilon}\}$ holds for the user prescribed accuracies $\tilde{\epsilon}, \epsilon \geq 0$. That is, UL0 can be stopped since the output iterate satisfies the (approximate) optimality requirements and no more computational work should be undertaken. Another possible practical modification resides in storing, if the memory space allows it, subsequent oracle outputs $x^*(\{i\}, S_{k+1})$ computed at iteration k for every $i \in [n] \setminus H_{k+1}$. These vectors will serve to *warm-start* problems $v(\{i\}, S_{k+2})$ at the next iteration, i.e. $k + 1$, for every $i \in [n] \setminus H_{k+2}$, reducing in practice the time cost of phase (b), theoretically $(n - |H_{k+2}|) \cdot (C + |S_{k+2}|^r)$.

Simulator It order to simulate UL0 on various (possible) instances of (MS) and then draw lessons about its efficiency, one must build models that describe relationships between all its constitutive elements. For instance, one must quantify how crude lower bounds $v(\{i\}, S)$ on *pieces'* values $v^{(i)}$ are. Another important feature to fix is deciding which and how many constraints are *active* at points $\{x^*(i)\}_{i \in [n]}$. Through the lens of the DAG interpretation and according to (6.32) and (6.33), it amounts to elect outgoing edges in the graph abstraction $G([n], E)$. We specify our model in Table 6.6.

As usual, $i \in [n]$, $S \subseteq [m]$ and $(\bar{i}, \bar{i}_+) \in E$.

expressions	parameters
$ \mathcal{C}(i) = \varsigma_{\text{act}} \cdot m$	$\varsigma_{\text{act}} \in (0, 1)$
$\mathbb{P}[\mathcal{C}(i)] \propto \sum_{j \in \mathcal{C}(i)} j^v$	$v > 0$
$\nu^{(\bar{i}+)} \leq \nu^{(\bar{i})} - \text{Exp}(\theta)$	$\theta > 0$
$\nu^{(i)} - \nu(\{i\}, S) = \min \left\{ \left(1 - \frac{ \mathcal{C}(i) \cap S }{ \mathcal{C}(i) } \right), \left(1 - \frac{ S }{m} \right)^{3/2} \right\} \cdot \text{Exp}(\bar{\theta})$	$\bar{\theta} > 0$
$\sum_{i=1}^n \text{outdeg}(i) = \varsigma_{\text{deg}} \cdot \frac{n \cdot (n-1)}{2}$	$\varsigma_{\text{deg}} \in (0, 1)$

Table 6.6 Summary of simulator's parametrized stochastic models.

Remark 6.17. Viewing $\varsigma_{\text{act}} \cdot m$ as the largest size among the *active constraints* sets $\{\mathcal{C}(i)\}_{i \in [n]}$, our model is intentionally adversarial so that $\bar{\mathcal{C}}$ (see (6.27)) is the biggest possible, hence rendering phase (b) of UL0 more expensive. We want our simulated results to be robust against such bad situations so that if UL0 wins (i.e. takes less computational time units to exit), it would largely win in more realistic situations.

- Parameter v induces a selection bias towards certain constraints. The biggest v is, the less diversity there will be in the $\varsigma_{\text{act}} \cdot m$ constraints of sets $\{\mathcal{C}(i)\}_{i \in [n]}$. Obviously the size of $\bar{\mathcal{C}}$ will likely drop with ς_{act} getting smaller and v getting bigger. This intuition will be confirmed on Figure 6.6 (a).
- Parameter θ serves as a reference quality difference between two adjacent *pieces* in the sense of the DAG interpretation, it can also be seen as the expected functional value gain one will incur by moving from *piece* $\bar{i}$ to $\bar{i}+$.
- Parameter $\bar{\theta}$ describes how far, on average, is the worst lower bound $\nu(\{i\}, \emptyset)$ from $\nu^{(i)}$. In addition, as underlined in (6.26), if $S \supseteq \mathcal{C}(i)$, $\nu(\{i\}, S)$ must equal $\nu^{(i)}$. Also, the more constraints were added to S , the tightest should the lower bounds be. Based on these observations, we derived the third model of Table 6.6. We can expect an abstract instance of (MS) to get more difficult as $\bar{\theta}$ grows.
- Parameter ς_{deg} represents the fraction of (directed) edges present in $G([n], E)$ with respect to the complete (undirected) graph with $n \cdot (n-1)/2$ edges. Parameter ς_{deg} plays a crucial role in the time spent within phase (a) of UL0 since the number of edges influences the number of paths within $G([n], E)$ hence the number of consecutive *pieces* one visits at each iteration k of Algorithm 2.

To create a single (abstract) instance of (MS) in accordance with the specifications of Table 6.6, we proceed as follows (implementation on [GitHub](#)).

We start from scratch, i.e. $E = \emptyset$, and iteratively add new edges to E . To that purpose, we sample one edge candidate uniformly at random among remaining possibilities, initially corresponding to the whole set $[n]^2$. That edge is accepted if no cycle is created or removed from the remaining possibilities otherwise. We loop until $\varsigma_{\text{deg}} \cdot n(n-1)/2$ edges are added.

For every $i \in [n]$ and $(\bar{i}, \bar{i}_+) \in E$, we draw independently $\theta_{(\bar{i}, \bar{i}_+)} \sim \text{Exp}(\theta)$, $\bar{\theta}_i \sim \text{Exp}(\bar{\theta})$ and $\mathcal{C}(i)$ of size $\varsigma_{\text{act}} \cdot m$. This latter is achieved without replacement and following the probability distribution of the first row of Table 6.6. We point out that this distribution is non-uniform, unless $v = 0$. Then, we solve the LP (6.34) to obtain $\mathbf{F}^{(i)} = F(x^*(i))$ and $\nu^{(i)}$ for every $i \in [n]$. Without loss of generality, we fixed a reference upper bound $\Lambda = 10$. Finally, we opt for a threshold $\xi \simeq 10^{-7} > 0$ for strict inequalities (6.33).

$$\begin{aligned}
 \max_{\mathbf{F} \leq \nu \leq \Lambda \cdot \mathbf{1}_n} \quad & \langle \mathbf{1}_n, \nu \rangle \\
 \text{s.t.} \quad & \nu^{(\bar{i}_+)} \leq \nu^{(\bar{i})} - \theta_{(\bar{i}, \bar{i}_+)} & \forall (\bar{i}, \bar{i}_+) \in E \\
 & \nu^{(\bar{i}_+)} \leq \mathbf{F}^{(i)} - \xi & \forall (\bar{i}, \bar{i}_+) \in E \\
 & \nu^{(i)} = \mathbf{F}^{(i)} & \forall i \in [n], \text{ outdeg}(i) = 0.
 \end{aligned} \tag{6.34}$$

Once an abstraction is generated, we can simulate Algorithm 2 without having to wait for the computations of $x^*(i)$ or $x^*(\{i\}, S)$ for numerous values of $i \in [n]$ and $S \subseteq [m]$ as all we actually care about are the quantities $\nu^{(i)}$, $\nu(\{i\}, S)$, $F(x^*(i))$ and sets $\mathcal{C}(i)$. Each time a call to the black-box (6.4) would have been performed in practice to compute $x^*(\{i\}, S)$, we increment a counter by $T(|S|) = C + |S|^r$ units, per Assumption (E). The total time taken by enumeration scheme and by UL0 after $k \in \mathbb{N}$ iterations are

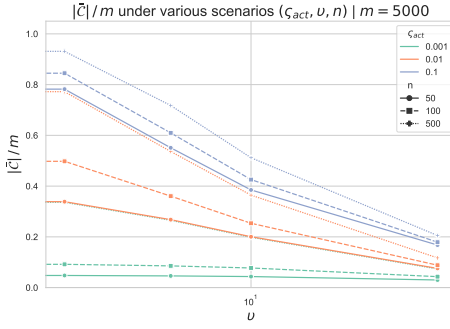
$$\Gamma_{\text{ES}}(k) = k \cdot (C + m^r), \tag{6.35}$$

$$\Gamma_{\text{UL0}}(k) = |H_k| \cdot (C + m^r) + \sum_{\tilde{k}=1}^k (n - |H_{\tilde{k}}|) \cdot (C + |S_{\tilde{k}}|^r). \tag{6.36}$$

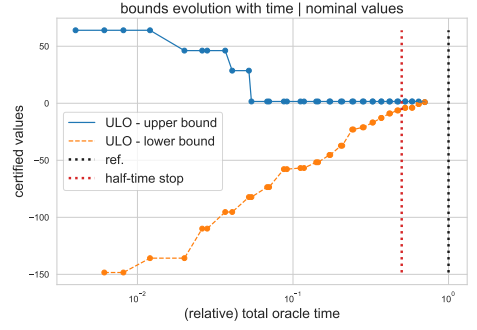
where one recalls the identities of Table 6.5. Let $K \in \mathbb{N}$ be the number of iterations performed in a run of UL0, the ratio between total times reads

$$Y = \Gamma_{\text{UL0}}(K) / \Gamma_{\text{ES}}(n). \tag{6.37}$$

6 | Upper-Lower Optimization method for (MC)

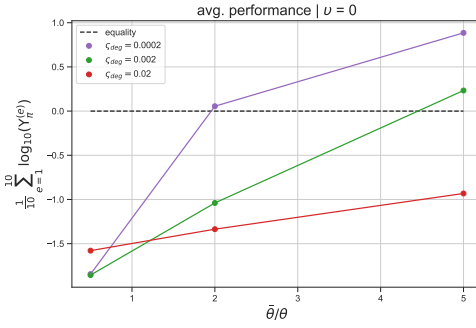


(a)

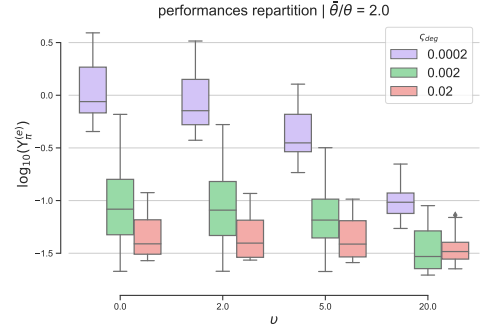


(b)

Fig. 6.6 Typical trends.



(a)



(b)

Fig. 6.7 Macroscopic insights.

dimensions / parameters / tolerances	nominal values
(n, m)	$(5 \cdot 10^2, 10^4)$
ζ_{act}	10^{-2}
(C, r)	$(1118, 3/2)$
θ	10
$(\tilde{\epsilon}, \epsilon)$	$(5 \cdot 10^{-2}, 10^{-3})$

Table 6.7 Nominal values of the parameters of our abstract model.

Results To obtain Figure 6.6 (b) and Figure 6.7 (a) & (b), we used the dimensions and parameters prescribed in Table 6.7. Ranges for other parameters are given below. Although there were several possible values for (C, r) , we stuck to $r = 3/2$ as it was motivated in our introductory section and led to clear interpretable results. Once r was fixed, we decided that C should match the cost of a thousandth of a full oracle call, i.e. $T(m)$ (see Assumption (E)). Note that the smaller the value of C is, the best for UL0 since it requires at least n calls to (Oracle) like the enumeration. However, if UL0 is executed for at least $K \geq 2$ (outer-)iterations, UL0 usually calls much more than n times the (Oracle).

A single run of UL0 is displayed on Figure 6.6 (b) where one can observe a typical trend. The upper bound $\hat{F}$ reaches $F^* = 1$ quite rapidly, i.e. after $\sim 10^{-1} \Gamma_{\text{ES}}(n)$ units of time. This behaviour will also be observed in practice in our real numerical experiments (Section 6.4). Here, the triplet $(\bar{\theta}, v, \zeta_{\text{deg}})$ was fixed to $(3, 5, 2 \cdot 10^{-3})$.

For graphs in Figure 6.7, we generated 10 abstract instances of (MS) for each triplet $(\bar{\theta}, v, \zeta_{\text{deg}}) \in \{5, 20, 50\} \times \{0, 2, 5, 20\} \times \{2 \cdot 10^{-4}, 2 \cdot 10^{-3}, 2 \cdot 10^{-2}\}$ and we averaged the overall (theoretical) times of UL0 over 250 random starting pieces $\hat{i} \in [n]$. That is, for each triplet $\pi = (\bar{\theta}, v, \zeta_{\text{deg}})$ and each instance $e \in [10]$, we sampled uniformly at random a 250-subset of $[n]$, namely $P_{\pi}^{(e)}$, and ran the simulation for each $\hat{i} \in P_{\pi}^{(e)}$ to obtain $\Gamma_{\text{UL0}}(K_{\pi}^{(e)}(\hat{i}))$ where $K_{\pi}^{(e)}(\hat{i})$ is the number of iterations UL0 took to complete when started from $\hat{i}$ under scenario π for the e -th abstract problem. We finally aggregate and normalize the performances associated with each abstract problem in the following way,

$$Y_{\pi}^{(e)} = \Gamma_{\text{UL0}, \pi}^{(e)} / \Gamma_{\text{ES}}(n) = (\Gamma_{\text{ES}}(n))^{-1} \sum_{\hat{i} \in P_{\pi}^{(e)}} |P_{\pi}^{(e)}|^{-1} \Gamma_{\text{UL0}}(K_{\pi}^{(e)}(\hat{i})).$$

The exit time of UL0 crucially depends on the quality of lower bounds (tuned by $\bar{\theta}$) with respect to the expected difference θ between two adjacent pieces (cfr. Table 6.6). Hence, we fixed θ arbitrarily and analyzed different regimes for $\bar{\theta}$, i.e. $\bar{\theta}/\theta \in \{1/2, 2, 5\}$. Not surprisingly, for lower values of $\bar{\theta}$, UL0 clearly outperforms the enumeration scheme; $10^{-1} \sum_{e=1}^{10} \log_{10}(Y_{\pi}^{(e)}) < -1$ for $\bar{\theta}/\theta = 1/2$ as depicted on Figure 6.7 (a) with $v = 0$. This is because one obtains a decent $\check{F} \leq F^*$ even when the set of considered constraints S is small, at little computational expense in phase (b) of UL0.

How impressive it might already look, we note that this setting (i.e. $v = 0$) translates an uniform sampling among the *constraints* to construct the sets of *active constraints* $\mathcal{C}(i)$ for every $i \in [n]$ so that $|\bar{\mathcal{C}}|$ (size of their union) is the biggest (cfr. Figure 6.6 (a)), usually detrimental for UL0. Therefore, the actual benefits of UL0 grow with the value of $v > 0$ as one can observe on Figure 6.7 (b). Nevertheless, on that same Figure 6.7 (b), we expose results that are more contrasted as we consider a larger ratio $\bar{\theta}/\theta = 2$. When ζ_{deg} is small, $|E|$ drops and the graph $G([n], E)$ is not well connected. Consequently, there exists a fair number of *local-minima* (scaling with nodes $i^* \in [n]$ such that $\text{outdeg}(i^*) = 0$ in $G([n], E)$) and phase (a) of UL0 will encounter difficulties in decreasing $\hat{F}$ multiple times in a row since long paths within $G([n], E)$ are limited by the small quantity of edges. Thereby, in general, UL0 takes advantage of a good connectivity between *pieces* of a (MS) problem, represented by a graph $G([n], E)$ with more edges. Interestingly enough when $\bar{\theta}/\theta = 1/2$, on Figure 6.7 (a), the relative performance of UL0 with respect to enumeration is predicted to be worst for the largest value of ζ_{deg} . We explain it as follows. Looking closely at the recorded data, for almost every simulation with $\pi = (5, 0, \zeta_{\text{deg}})$, there were only $K = 2$ outer-iterations meaning that problems were really easy for UL0. Phase (b) likely provided a good $\hat{i} \in [n]$ at the end of the first outer-iteration based on which the second call to phase (a) led from $\hat{i}$ to i^* with i^* being an optimal *piece*. However, both calls to phase (a) were presumably more expensive when ζ_{deg} was the largest since $G([n], E)$ is more connected.

Takeaway message A clear correlation between $|\bar{\mathcal{C}}|$ and the efficiency of UL0 can be established and is driven by parameters (ζ_{act}, v) : the smaller $|\bar{\mathcal{C}}|$, the better UL0 is expected to perform. Moreover, UL0's efficiency heavily depends on the quality of lower bounds $\{v(\{i\}, S)\}_{i \in [n]}$ on *pieces'* values $\{v^{(i)}\}_{i \in [n]}$ for sets $S \subseteq [m]$. When $v^{(i)} - v(\{i\}, S)$ vanishes relatively quickly to 0 for $|S| \ll m$ then UL0 should beat the enumeration strategy.

6.4 Numerical Experiments

In this last section, we benchmark three algorithms to solve (generalized) problem instances based on Example 6.1. Aside from UL0 and the Enumeration Scheme (ES), we also put into test a third procedure, Restart Alternating Minimization (RAM), which is a mix between UL0 and ES.

Restarted Alternating Minimization As developed in the previous section, starting from a piece $i \in [n]$, one can jump to another piece $i_+ \in \mathcal{A}_\rho(x^*(i))$ while expecting a functional decrease or, at least, a functional increase of a most ρ (see Lemma 6.9).

We have observed that iterating these jumps until we are stuck amounts to a random walk in a particular directed (sub)graph made of all the nodes matched with *pieces* that were not visited yet. This corresponds to phase (a) of UL0. Actually, when $\rho \rightarrow 0$, it can be seen as an alternating minimization scheme. To see this, we reformulate the discrete minimum over the n *pieces* using variables $q \in \Delta^n$ and rewrite (MS) as

$$F^* = \min_{x \in \mathcal{X} \cap \mathcal{D}([m]), q \in \Delta^n} \sum_{i=1}^n q^{(i)} f^{(i)}(x) \quad (6.38)$$

Every *piece* $i \in [n]$ is uniquely encoded by $q^{(i)} = 1$ and $q^{(i')} = 0$ for every $i' \in [n] \setminus \{i\}$. Let $x \in \text{dom } F$ be fixed. If $i_+ \in \mathcal{A}_0(x)$, the vector q_+ that encodes i_+ (as described in the last sentence) must satisfy

$$q_+ \in \arg \min_{q \in \Delta^n} \sum_{i=1}^n q^{(i)} f^{(i)}(x).$$

Conversely, fixing $q \leftarrow q_+$ and minimizing over x this time, $x_{++} = x^*(i_+)$ satisfies

$$x_{++} \in \arg \min_{x \in \mathcal{X} \cap \mathcal{D}([m])} \sum_{i=1}^n q_+^{(i)} f^{(i)}(x).$$

Phase (b) has been devised to produce meaningful bounds on F^* and provide a clever restart of phase (a), which are the main principles behind UL0. One could adopt a strategy that, instead of executing phase (b) as in UL0, randomly restart phase (a) when reaching a sink. Unlike the Enumeration Scheme that tries every piece $i \in [n]$ in a totally random order (i.e. computes the values $\{v^{(i)}\}_{i \in [n]}$ one-by-one according to a given permutation of $[n]$ to solve (MS)), one can expect to initialize phase (a) with a piece $\hat{i} \in [n]$ and follow a directed path leading, in just a few alternating minimization steps, to an optimal *piece* i^* . Therefore, we name this intermediate strategy Restarted Alternating Minimization (RAM). We suspect that it will reach, on average, an upper bound $\hat{F} = F^*$ quicker than (ES). To point out the eventual benefits of phase (b) alone, we also account for $\rho = 10^{-3}$

in phase (a) of Restarted Alternating Minimization, just like for UL0. They only differ by the way they restart their local-search phase. That being stated, the effect of a small ρ remains, however, limited and we do not observe substantial variations when letting $\rho \rightarrow 0$.

6.4.1 Pessimistic-Optimistic Piecewise-Linear Programming

We consider piecewise-linear optimization problems. Our problems originate from the continuous interpolation between *optimistic* [NTM17] and *pessimistic* perturbed linear programming arising in market investment optimization. One disposes of n market scenarios $\{(\beta^{(i)}, \gamma^{(i)})\}_{i \in [n]}$ stemming from possible deviations with respect to a *nominal market* $(\bar{\beta}, \bar{\gamma}) \in \mathbb{R}_+^p \times \mathbb{R}_+$. From each tuple $(\beta^{(i)}, \gamma^{(i)})$, $\beta^{(i)} = (\beta_1^{(i)}, \dots, \beta_p^{(i)})$ describes the expected returns for assets 1 to p of market $i \in [n]$, expressed for one unit of investment whereas $\gamma^{(i)}$ denotes a fixed fee cost entering that specific market. In a *pessimistic* setting, one would like to maximize the expected profit assuming that the worst market scenario would occur. Opposite in sense, the *optimistic* setting considers that one can choose which market conditions will apply and then optimize the investments accordingly. Let $R > 0$ be the amount of investment units at disposal. We also take into account m maximal investment conditions $\{(v^{(j)}, W^{(j)})\}_{j \in [m]}$, i.e. $\langle v^{(j)}, u \rangle \leq W^{(j)}$ for every $j \in [m]$. Finally, it is allowed for these constraints to be violated by a positive amount of at most η . This latter will be (heavily) penalized in the objective by $-C_{\text{pen}} \cdot \eta$ ($C_{\text{pen}} \sim 5 \cdot 10^4$ in our experiments).

The full problem reads

$$\begin{aligned} \max_{(u, \eta) \in \mathcal{U} \times \mathbb{R}_+} \quad & -C_{\text{pen}} \cdot \eta + \zeta_{\text{cvx}} \cdot \left(\min_{\tilde{i} \in [n]} \langle \beta^{(\tilde{i})}, u \rangle - \gamma^{(\tilde{i})} \right) \\ & + (1 - \zeta_{\text{cvx}}) \cdot \left(\max_{i \in [n]} \langle \beta^{(i)}, u \rangle - \gamma^{(i)} \right) \quad (\text{POPLP}) \\ \text{s.t.} \quad & \max_{j \in [m]} \langle v^{(j)}, u \rangle - W^{(j)} - \eta \leq 0 \end{aligned}$$

with parameter $\zeta_{\text{cvx}} \in [0, 1]$ allowing a trade-off between *pessimistic* ($\zeta_{\text{cvx}} = 1$) and *optimistic* ($\zeta_{\text{cvx}} = 0$) settings. For instance, one could optimize u under the hypothesis that there is an equal chance that the *pessimistic* or the *optimistic* scenario occurs, i.e. $\zeta_{\text{cvx}} = 1/2$. Note that if $\zeta_{\text{cvx}} = 1$, (POPLP) becomes convex. Thus, we focus on $\zeta_{\text{cvx}} \in [0, 1)$.

Remark 6.18. One can identify (POPLP) as (MS) instance with

elements	expression
d	$p + 1$
$\mathcal{X}$	$\mathcal{U} \times \mathbb{R}_+$
$f^{(i)}$	$C_{\text{pen}} \cdot \eta + \zeta_{\text{cvx}} \cdot (\max_{\tilde{i} \in [n]} \gamma^{(\tilde{i})} - \langle \beta^{(\tilde{i})}, u \rangle)$ $+ (1 - \zeta_{\text{cvx}}) \cdot (\gamma^{(i)} - \langle \beta^{(i)}, u \rangle)$
$c^{(j)}$	$\langle v^{(j)}, u \rangle - W^{(j)} - \eta$

Table 6.8 (POPLP) translated as an instance of (MS).

Data Generation

We precise now the procedure that produced our (synthetic) data for (POPLP).

- For every $i \in [n]$, the data has been generated identically and independently as follows. The *nominal scenario* is elected as follows. We set $\bar{\gamma} = 0$ and for every $l \in [p]$, $\bar{\beta}_l \sim \text{Uni}([15])$. Then, $\Delta_l^{(i)}$ is drawn with respect to

$$\mathbb{P}[\Delta_l^{(i)} = \delta] = \begin{cases} 3/5 & \delta = 0 \\ 1/10 & \delta \in \{\pm 1, \pm 2\} \\ 0 & \text{otherwise.} \end{cases}$$

Moreover, let $I \in \{1, 10\}$ be a scaling parameter, we set $\beta^{(i)} = \max\{\mathbf{0}_p, \bar{\beta} + I \cdot \Delta^{(i)}\}$ and $\gamma^{(i)} \sim \text{Uni}([0, \gamma_{\max}])$ with

$$\gamma_{\max} := \frac{15}{100} \cdot \left(\max_{\tilde{i} \in [n]} \|\beta^{(\tilde{i})}\|_1 - \min_{i \in [n]} \|\beta^{(i)}\|_1 \right).$$

- For every $j \in [m]$, $v^{(j)} \geq \mathbf{0}_p$ is a ι -sparse vector with $\iota \sim \text{Uni}(\{\lceil \frac{p}{10} \rceil, \dots, p\})$. Positions at which $v_l^{(j)} > 0$ were sampled uniformly at random on $[p]$ and were assigned, uniformly at random, a value from $[10]$. Introducing parameter $\zeta > 0$, we decided to set

$$W^{(j)} = \zeta \cdot \|v^{(j)}\|_1 / p > 0. \quad (6.39)$$

6 | Upper-Lower Optimization method for (MC)

- According to the classical investment framework, the search space $\mathcal{U}$ takes the form of the p -dimensional simplex of radius $R > 0$,

$$\mathcal{U} := \{u \in \mathbb{R}^p \mid \langle \mathbf{1}_p, u \rangle = R, u \geq \mathbf{0}_p\}.$$

We note that problem (POPLP) fulfills Assumption (B) since $\mathcal{U}$ is bounded.

Our (POPLP) problems can be described using seven parameters. We list them as (n, m) , I , ζ , (p, R) and ς_{cvx} . We have investigated three different settings, namely *nominal (hard/easy)* and *big- n* , for which we detail the tested values of the parameters in Table 6.9.

dimensions / parameters	nominal (hard)	nominal (easy)	big- n
(n, m)	$(5 \cdot 10^2, 10^4)$		$(10^3, 10^3)$
I	1	10	1
ζ	$\{10^{-1}, 1, 5\}$		
(p, R)	$(3 \cdot 10^2, 10^2)$	$(10^2, 10)$	
ζ_{cvx}	$\{0, 1/2\}$		

Table 6.9 Nominal values of the (POPLP) parameters.

Benchmark procedure Throughout the sequel, as we benchmark the three studied algorithms (ULO, RAM, ES) on a new instance problem, we run times methods ULO and RAM $N_{\text{rep}} = 15$ with common N_{rep} initial pieces $(\hat{i}_1, \dots, \hat{i}_{N_{\text{rep}}})$ of $[n]^{N_{\text{rep}}}$. We have fixed a time limit of 4 minutes for every repetition. We have implemented a log routine within the algorithms to record tuples $(t, \hat{F}, \check{F})$ anytime an upper/lower bound candidate is produced at elapsed time $t > 0$. We let ES finish until it solves (POPLP) by computing the values of the n convex subproblems $\{\nu^{(i)}\}_{i \in [n]}$, as in (6.5). For any $i \in [n]$, let $t^{(i)}$ depict the time taken to get $\nu^{(i)}$. Then, it is easy to simulate runs of ES without any further numerical experiment by simply considering any bijection of $\varrho : [n] \rightarrow [n]$. Each of these bijections induces a permutation of $[n]$ for which we can collect tuples

$$(t, \hat{F}) = \left(\sum_{k'=1}^k t^{(\varrho(k'))}, \min_{k'=1, \dots, k} \nu^{(\varrho(k'))} \right) \quad (6.40)$$

at each iteration $k \in [n]$. Therefore, in what concerns algorithm ES, we could afford to sample 10^3 permutations acting like as many repetitions.

Aggregation In order to compare the algorithms, we defined $N_{ts} \in \mathbb{N}$ timestamps $0 < \tau^{(1)} < \dots < \tau^{(N_{ts})}$ corresponding to elapsed wall-clock times. Instead of the aforementioned $(t, \hat{F}, \check{F})$ involving various unpredictable times t , we search for new values $(\tau, \hat{F}_{\text{new}}, \check{F}_{\text{new}})$ that will resemble the most the actual recorded tuples, so that they share a common temporal basis. Thereby, for every setting tested and for every recorded repetition of a given algorithm, we collected the values $(\hat{F}, \check{F})$ obtained as late as possible before each timestamp. That is, we picked values of $\hat{F}$ and $\check{F}$ obtained during that repetition corresponding to the last true time $t \leq \tau$. Accordingly, we set $\hat{F}_{\text{new}} \leftarrow \hat{F}$ (respectively $\check{F}_{\text{new}} \leftarrow \check{F}$). We recall that, for algorithms ES and RAM, $\check{F} = -\infty$ unless every piece has been visited, i.e. $k = n$ in (6.40). Finally, we take the mean of these new value traces $(\tau, \hat{F}_{\text{new}}, \check{F}_{\text{new}})$ to represent the expected behaviour of UL0, RAM or ES.

Remark 6.19. We point out one minor drawback of the above procedure. For early timestamps, it could happen that for some repetitions, no tuples were recorded yet. Hence, an averaged performance measure (e.g. a primal gap) plotted for a given algorithm would be based on less than N_{rep} . In such case, we may observe a deterioration of that measure at the next timestamp if ever the new repetitions coming into play present worse $(\hat{F}, \check{F})$ values than the average of the previous timestamp.

Results We subdivide the analysis of results in two parts. In the first part, we make generic comments based on raw pictures. In the second part, we document UL0's behaviour and the effects of parameter ζ_{cvx} .

Global analysis. Graphs in Figure 6.10 clearly showcase the advantages of phase (a) within Algorithm 2 (UL0), especially in the most constrained setting, i.e. $\zeta = 10^{-1}$. Indeed, for every investment condition $(v^{(j)}, W^{(j)})$ with $j \in [m]$, inequality $\langle v^{(j)}, u \rangle \geq 0$ holds for any $u \in \mathcal{U}$. Therefore, with small values of ζ in (6.39), we bound the quantities $\langle v^{(j)}, u \rangle$ from above more strictly, reducing the effective feasible domain for variable u and, as a byproduct, the minimizers $x^*(i)$ of several *pieces* might become identical. Yet, their values $v^{(i)}$ presumably differ. As a consequence, *pieces* tend to be more connected in the sense of (6.33). Algorithm RAM can fully exploit that structure since it skips phase (b) from UL0 to immediately relaunch phase (a) at a new *piece* $\hat{i} \in [n]$ as soon as it gets stuck (possibly at a *local optimum*). In these favorable situations, many starting *pieces* $\hat{i}$ would

be on a connected path, according to the underlying DAG $G([n], E)$ representation of problem (6.17), towards a globally optimizing *piece* $i^* \in [n]$, i.e. $F^* = v^{(i^*)}$. UL0 also profits from this phenomenon but spends a substantial time in phase (b) to obtain lower bounds certificates. Yet, in the *nominal (hard/easy)* experiments, despite its global optimization focus (unlike RAM), UL0 sometimes beats RAM and largely outperforms ES. More precisely, RAM is the fastest to reach any level $0 < \epsilon \leq 5 \cdot 10^{-2}$ of primal gap $\hat{F} - F^* \leq \epsilon$ for the *nominal (hard)* experiment but UL0 takes the win in the *nominal (easy)* experiment when $\zeta = 1$ and $\zeta = 5$. Phase (b) sometimes turns out to be a burden. Considering settings like the *big-n* experiment wherein $n = m = 10^3$, phase (b)'s relative cost becomes prohibitive if several iterations of UL0 ($k = 1, \dots, K$) are run with substantial sizes for S_k in Algorithm 2. This is unfortunately the case on the two last graphs of Figure 6.10 but as we explain hereafter, UL0 still performs relatively well.

UL0 detailed analysis. On Figure 6.12, one can observe the evolution of lower bounds $\check{F}$ produced by UL0 on the go. The two last graphs allow us to nuance our last statements about UL0's relative failure on experiment *big-n*. Indeed, we can see that if $\tilde{\epsilon} = 5 \cdot 10^{-2}$, i.e. 5% relative accuracy is tolerated in Algorithm 2, then UL0 exits with optimality guarantees after 63[s] on average when $(\zeta_{\text{cvx}}, \zeta) = (0, 1)$ and exits with even better 1% relative accuracy after roughly 90[s] when $(\zeta_{\text{cvx}}, \zeta) = (1/2, 5)$. In comparison, ES terminates with its (exact) optimality guarantees after 76[s] and 150[s] respectively for equivalent settings. In what concerns the *nominal* experiments, upper and lower bounds both progress relatively smoothly with time making it quite predictable to know when UL0 will reach a given level of $\tilde{\epsilon}$ relative accuracy. We end this discussion by commenting on the value of (POPLP) with hyper-parameter ζ_{cvx} . Although not displayed on Figure 6.10, one can notice on the first two graphs of Figure 6.12 that the value of F^* on the right-hand side ($\zeta_{\text{cvx}} = 1/2$) is larger than left-hand side's ($\zeta_{\text{cvx}} = 0$). This was of course expected since problem's data was common to both situations and the parameters (n, m, I, ζ, p, R) were fixed. More interestingly, as ζ_{cvx} grows up to 1, the weight put on the common max-affine (convex) part of *pieces* (see description of $f^{(i)}$ in Table 6.8) becomes more significant. Eventually, (POPLP) becomes a simple linear program and its intrinsic enumerative complexity with n vanishes. We finally underline that UL0 did, on average, take less time on the more convex case ($\zeta_{\text{cvx}} = 1/2$) to make its upper bound $\hat{F}$ meet F^* . It took around 220[s] for the case $\zeta_{\text{cvx}} = 0$ against 175[s] for $\zeta_{\text{cvx}} = 1/2$.

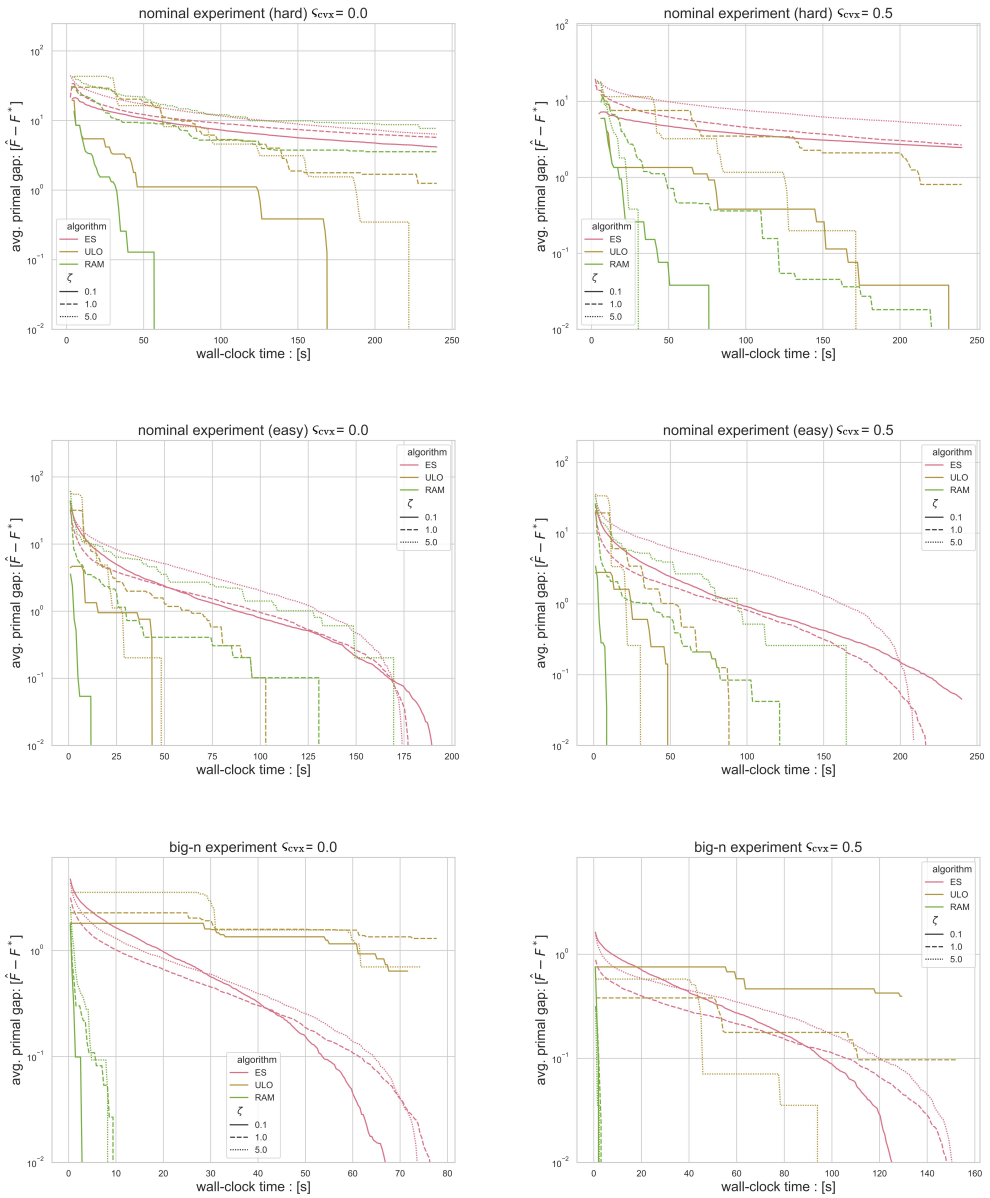


Fig. 6.10 (POPLP) experiments: benchmarks of ULO, RAM & ES strategies.

6 | Upper-Lower Optimization method for (MC)

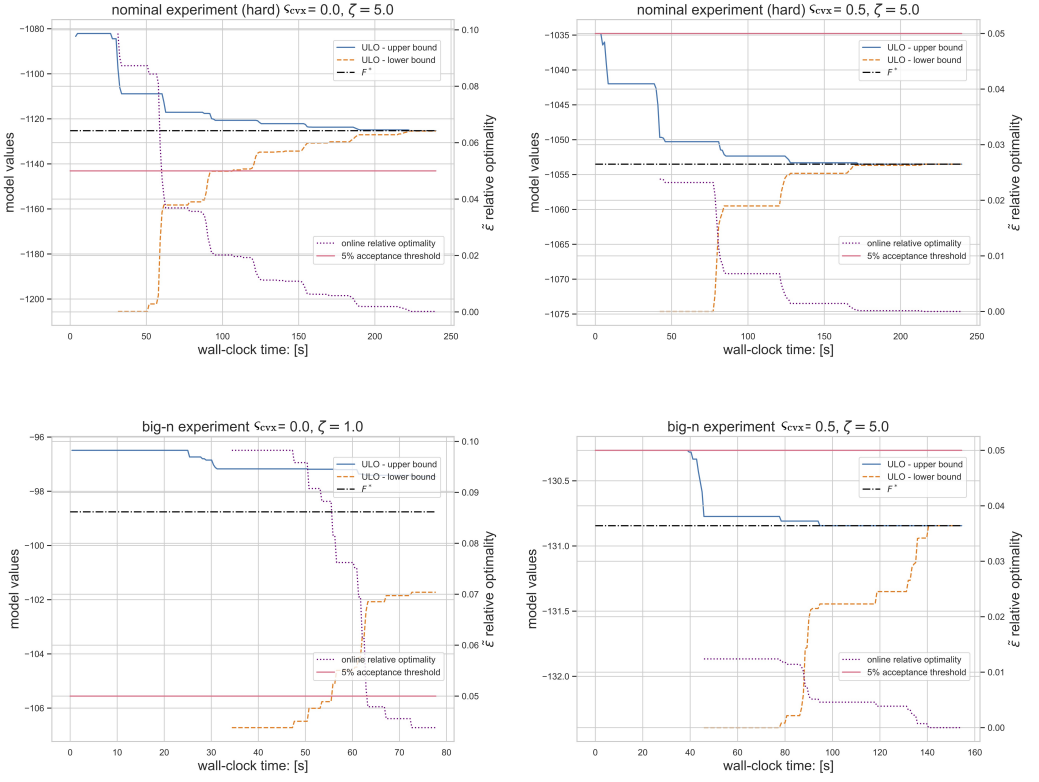


Fig. 6.12 (POPLP) experiments: details of ULO optimality certificates.

6.5 Conclusion

In this chapter, we have introduced and motivated the framework of *minimum structured* optimization problems, dubbed as (MS), whose objectives generalize (MC) functions from Section 4.3.1. Although solvable in a straightforward fashion by a simple enumeration scheme, we have shown how global upper/lower models could be of help to significantly reduce its computational complexity. Based on these models, we have devised and analyzed a new global optimization algorithm, ULO, for which exactness guarantees are provided. We have first conducted theoretical experiments to assess when ULO should exhibit better performances than enumeration.

Then we have also tested UL0 and another local-search heuristic, on a practical optimization task. The results are very convincing, as UL0 is definitely the algorithm of choice for instances of (MS) linked to bigger ratios m/n , i.e. the problem (MS) involves far more *functional constraints* than *pieces*. This is typically the case for *optimistic robust* linear programs (Example 6.1). We leave the possible parallelization of phase (a) and (b) in UL0 with asynchronous communication steps for further research, which could enhance even more its practical performance.

7

Relaxed Alternating Minimization for (SMC)

7.1 Introduction

As previously announced in Chapter 4 (Section 4.3.1) & 6 (Section 6.1.1), we address now (constrained) optimization problems of the form

$$F^* = \min_{x \in \mathcal{X}} \left\{ F(x) := \bar{h}(x) + \frac{1}{N} \sum_{s=1}^N \underbrace{\min_{l=1, \dots, n_s} h_l^{(s)}(x)}_{h^{(s)}(x)} \right\} \quad (\text{SMC})$$

where $N \in \mathbb{N}$ and $\mathcal{X} \subseteq \mathbb{R}^d$. We remind from Section 4.3.1 that (SMC) is nothing else than a structured instance of (MC) being **(ii) non-exhaustively enumerable**, in stark contrast with problems tackled in Chapter 6.

For every index $s \in [N]$, we consider $n_s \in \mathbb{N}$ continuous functions $h_1^{(s)}, \dots, h_{n_s}^{(s)}$ called *component functions*. Together with $\bar{h}$ lower semicontinuous, dubbed as the *main function*, they are assumed to be closed, proper and convex. Throughout the sequel, we further assume that problem (SMC) is *feasible* and *bounded* (i.e. $F^* > -\infty$ and $\mathcal{X} \cap \text{dom } F \neq \emptyset$ hence $F^* < \infty$).

7 | Relaxed Alternating Minimization for (SMC)

Problems of the form (SMC) frequently arise in applications such as clustering [PSW22], minimization of truncated convex functions [LJ19] (e.g. regression with capped polyhedral norm penalty [OA13] and regression robust to outliers [BAB20]). More generally, it encompasses the wide class of Piecewise-Linear-Quadratic optimization programs [CCHP20].

Goals The intrinsic nature of (SMC) is combinatorial, as already unveiled in equation (4.9) (Chapter 4). From (4.9), it was possible to enumerate all the underlying *pieces* $\{f^{(i)}\}_{i \in [n]}$ of objective F in (SMC) so to recast (SMC) problem as a specific instance of (MS) (Chapter 6). Thereby, one could apply the Enumeration Strategy (ES)¹ and solve (deterministically) (SMC), as we detail now. Let $\sigma = (\sigma_1, \dots, \sigma_N) \in \times_{s=1}^N [n_s]$, we define

$$\nu(\sigma) := \min_{x \in \mathcal{X}} \left\{ F_\sigma(x) := \bar{h}(x) + \frac{1}{N} \sum_{s=1}^N h_{\sigma_s}^{(s)}(x) \right\}. \quad (\text{Oracle})$$

Note that the convexity of F_σ ensures the convexity of the (sub)problems mobilized in (Oracle). We consider that $\nu(\sigma)$ is computed *exactly*, setting numerical accuracy concerns aside. By feeding (Oracle) every possible σ -selection, one is guaranteed to solve (SMC) as

$$F^* = \min_{\sigma \in \times_{s=1}^N [n_s]} \nu(\sigma) = \min_{x \in \mathcal{X}} \min_{\sigma \in \times_{s=1}^N [n_s]} F_\sigma(x). \quad (7.1)$$

Obviously, this enumerative approach quickly becomes impractical since

$$n := \text{card} \left(\times_{s=1}^N [n_s] \right) = \prod_{s=1}^N n_s \in \mathcal{O} \left(\left(\max_{s=1, \dots, N} n_s \right)^N \right) \quad (7.2)$$

grows exponentially with N , e.g. $\bar{n}^N$ if $n_s = \bar{n}$ for every $s \in [N]$. Unfortunately, our motivation essentially comes from problems exhibiting a large N , e.g. $N \sim 10^2 \rightarrow 10^5$. In the absence of simplifications, there always exists a problem instance that requires a *scan* of the n possible *pieces* F_σ when it comes to global optimization, as we clarify in the paragraph about (SMC)'s difficulty. In other words, the ES is optimal in the worst case. Actually, [BAB20] points out that finding F^* is at least as hard as the (NP-hard) subset sum problem (SSP), since (SSP) can be written as a (SMC) problem. Global optimality out of reach, we are mainly driven by empirical results.

¹Our Upper-Lower Optimization (ULO) method should be employed if $\mathcal{X}$ entails a large number m of functional constraints $c^{(j)}(x) \leq 0$ for every $j \in [m]$.

Therefore, our primary goal is to develop heuristic methods, yet covered by sound convergence guarantees towards *critical* points of (SMC). To that end, we will use a slightly stronger tool than (Oracle) (i.e. Assumption (A)), hopefully obtaining the best solutions possible for a prescribed amount of computational time. For most of the works in nonconvex nonsmooth optimization, *criticality* represents the standard satisfaction requirement. Nevertheless, with all the structure at hand regarding the objective function, one can hope for better. As a secondary goal, we also aim at filtering out mere *critical* points, ultimately striving for *local-minima*.

Data driven Aside from purely academic toy examples, (SMC) problems are typically data driven. Starting from a collection of $N \in \mathbb{N}$ elements from $\mathbb{R}^p$, i.e. $\{\beta^{(s)}\}_{s=1}^N$, one tries to minimize a (regularized) average loss over the dataset, i.e.

$$\min_{x \in \mathcal{X}} \bar{h}(x) + \frac{1}{N} \sum_{s=1}^N h(x | \beta^{(s)}) \quad (\text{data-SMC})$$

where $h(\cdot | \beta)$ is the pointwise minimum among $\bar{n} \in \mathbb{N}$ fixed proper convex individual losses $\{h_l(\cdot | \beta)\}_{l=1}^{\bar{n}}$ that only depend on $\beta \in \mathbb{R}^p$, i.e.

$$h(\cdot | \beta) := \min_{l \in [\bar{n}]} h_l(\cdot | \beta). \quad (7.3)$$

In other words, about (data-SMC), $n_s = \bar{n}$ for every $s \in [N]$. Moreover, a term $h(\cdot | \beta^{(s)})$ matches with $h^{(s)}$ whereas for every $l \in [\bar{n}]$, $h_l(\cdot | \beta^{(s)})$ corresponds to the *component* $h_l^{(s)}$. Two main types of situations, both of which we illustrate by a meaningful example, lead to $h(\cdot | \beta)$ as described in (7.3).

Multiple choices.

In multifacility location problems [HN09], one needs to decide where to install $\bar{n} \in \mathbb{N}$ new *facilities* $x_l \in \mathbb{R}^2$ for every $l \in [\bar{n}]$, taking into account territory constraints, i.e. $x_l \in \mathcal{X}_l$ for every $l \in [\bar{n}]$. The overall decision variable is $x = (x_1, \dots, x_{\bar{n}})$ and $\mathcal{X} = \mathcal{P} \cap \times_{l=1}^{\bar{n}} \mathcal{X}_l$ where $\mathcal{P} \subseteq \mathbb{R}^{2 \cdot \bar{n}}$ implements binding constraints for *facilities'* locations. Here, $\beta \in \mathbb{R}^2$ ($p = 2$) represents a customer's position. This customer incurs a proximity loss regarding the l -th *facility* set by

$$h_l(x | \beta) = \|x_l - \beta\|. \quad (7.4)$$

7 | Relaxed Alternating Minimization for (SMC)

where $\|\cdot\|$ depicts a metric over $\mathbb{R}^2$. Hence, his loss becomes the minimum over all the installed *facilities*, i.e. the loss induced by the closest one,

$$h(x | \beta) = \min_{l \in [\bar{n}]} \|x_l - \beta\|.$$

Range cuts.

In SVM problems, each element $\beta = (\gamma, \bar{\beta})$ includes a class $\gamma \in \{-1, 1\}$ and features $\bar{\beta} \in \mathbb{R}^{p-1}$ ($p > 1$ here). The goal is to find a vector $x \in \mathbb{R}^{p-1}$ ($d = p - 1$) that correlates positively with $\bar{\beta}$ when $\gamma = 1$ and negatively otherwise. Hence, a sound loss can be $\ell_\beta(x) = \max\{0, 1 - \gamma \langle \bar{\beta}, x \rangle\}$ with lower values of ℓ_β indicating a good classification of the data point β . However, in the presence of outliers, the decision vector x can be highly influenced by non-representative data. Therefore, in order to tame the impact of outliers, it is common to *truncate* [LJ19] or *clip* [BAB20] the loss ℓ_β when it exceeds a fixed threshold quantity $\lambda > 0$. With $\bar{n} = 2$, $\mathcal{X} = \mathbb{R}^{p-1}$, it comes

$$h(x | \beta) = \min\{\max\{0, 1 - \gamma \langle \bar{\beta}, x \rangle\}, \lambda\},$$

$$h_1(x | \beta) = \max\{0, 1 - \gamma \langle \bar{\beta}, x \rangle\}, \quad h_2(x | \beta) = \lambda. \quad (7.5)$$

Remark 7.1. Whether regarding (data-SMC) or the more general (SMC), the *main function* $\bar{h}$ primarily serves as a regularizing term promoting some desired structure about the decision variable x . Also, as it will be the case for our motivational Example 7.4, it can capture the common convex part to all the *components*. Indeed, let us consider a (SMC) problem. If for every $s \in [N]$ and $l \in [n_s]$, $h_l^{(s)} = \bar{h}^{(s)} + \Delta h_l^{(s)}$ with both $\bar{h}^{(s)}$ and $\Delta h_l^{(s)}$ proper convex then we can reformulate

$$F^* = \min_{x \in \mathcal{X}} \underbrace{\bar{h}(x) + \frac{1}{N} \sum_{s=1}^N \bar{h}^{(s)}(x)}_{\text{new main term}} + \frac{1}{N} \sum_{s=1}^N \min_{l \in [n_s]} \underbrace{\Delta h_l^{(s)}(x)}_{\text{new components}}. \quad (7.6)$$

Formulation (7.6) is preferred to (SMC) since it only retains the discriminative part of the *components*, i.e. $\Delta h_l^{(s)}$ for every $s \in [N]$ and $l \in [n_s]$. Thereby, it becomes more obvious where on $\mathcal{X}$ a *component's* value $h_{l_+}^{(s)}(x)$ is smaller than its alternatives, i.e. $h_l^{(s)}(x)$ for $l \in [n_s] \setminus \{l_+\}$.

Difficulty By default, problem (SMC) is nonconvex. However, consciously or not, sometimes it degenerates into a convex program. It is trivially convex if $n_s = 1$ for every term $s \in [N]$. More subtly, it will also be convex if there exists a selection $\sigma \in \times_{s=1}^N [n_s]$ such that $F(x) = F_\sigma(x)$ for every $x \in \mathcal{X}$. Otherwise, the objective F , restricted to $\mathcal{X}$, must be expressed as the pointwise minimum of a number $\hat{n} > 1$ (and $\hat{n} \leq n$) of convex functions, turning (SMC) into a nonconvex program for sure. We note that in some (very) favorable settings with $d \leq 2$, it is shown in [LJ19] that one can cope globally with problem (SMC) in polynomial-time by solving $\mathcal{O}(N^d)$ convex subproblems, i.e. (Oracle), even though $n_s = 2$ for every $s \in [N]$. However, as suggested, their methodology is not applicable for any convex *components* and especially not suited for $d > 2$. Even if it were applicable more broadly, we highlight the exponential complexity in the dimension d , corroborating the NP-hardness result stated previously in the paragraph about our **goals**. Actually, (SMC) objectives can exhibit an exponential number of *local minima*. To prove this, we propose a family of worst-case instances (Proposition 7.2). They embrace the full complexity of our problem in the sense that for any selection $\sigma \in \times_{s=1}^N [n_s]$, there exists a non-empty region $\mathcal{R}$ of $\mathbb{R}^d$ over which σ is the unique selection leading to $F(x) = F_\sigma(x)$ for every $x \in \mathcal{R}$. We call these objectives *fully-active* since every piece F_σ is useful in F 's enumerative description (7.1).

Proposition 7.2 (Fully-active (SMC) instance). Let $N \in \mathbb{N}$ be fixed and the number of components $n_s \in \mathbb{N}$ be chosen for every $s \in [N]$. For any $d \geq N$, let $\bar{h} : \mathbb{R}^d \rightarrow \mathbb{R}$ be a proper convex function. The function $\mathcal{W}$ defined for every $x \in \mathbb{R}^d$ by

$$\mathcal{W}(x) = \bar{h}(x) + \frac{1}{N} \sum_{s=1}^N \min_{l \in [n_s]} \left\{ h_l^{(s)}(x) := l \cdot x_s + \frac{l \cdot (l-1)}{2} \right\} \quad (7.7)$$

is fully-active. I.e., for every $\sigma \in \times_{s=1}^N [n_s]$, σ is the unique selection leading to

$$\mathcal{W}(x) = \mathcal{W}_\sigma(x) := \bar{h}(x) + \frac{1}{N} \sum_{s=1}^N \left(\sigma_s \cdot x_s + \frac{\sigma_s \cdot (\sigma_s - 1)}{2} \right)$$

for every $x \in \times_{s=1}^N]-\sigma_s, -\sigma_s + 1[$.

7 | Relaxed Alternating Minimization for (SMC)

Proof. We notice the independence of terms $h^{(s)} = \min_{l \in [n_s]} h_l^{(s)}$ for every $s \in [N]$. At any $x \in \mathbb{R}^d$ and $(s_1, s_2) \in [N]^2$, this implies that an *active component* $h_{\sigma_{s_1}}^{(s_1)}$, i.e.

$$h_{\sigma_{s_1}}^{(s_1)}(x) \leq \min_{l_1 \in [n_{s_1}]} h_{l_1}^{(s_1)}(x)$$

has no influence on the possible activation of $h_{\sigma_{s_2}}^{(s_2)}$ regarding the term $h^{(s_2)}$. Indeed, the s -th dimension of vector x defines on its own which *component* among $h_1^{(s)}, \dots, h_{n_s}^{(s)}$ will be *active*. Hence, every combination $\sigma \in \times_{s=1}^N [n_s]$ can achieve $\mathcal{W}(x) = \mathcal{W}_\sigma(x)$, conditional on the fact that each $h_{\sigma_s}^{(s)}$ might become *active* for some $x_s \in \mathbb{R}$. One can observe that this is the case, i.e. $h_{\sigma_s}^{(s)}$ is *active* for every $x_s \in [-\sigma_s, -\sigma_s + 1]$. Finally, on $]-\sigma_s, -\sigma_s + 1[$, $h_{\sigma_s}^{(s)}$ is the only *active component* for the term $h^{(s)}$. $\square$

To summarize, in what concerns *fully-active* objectives, globally minimizing F using (Oracle) suggests that, indeed, one should solve n convex sub-problems, possibly leading to as many *local-minima* among which up to $n - 1$ are spurious, i.e. non-global.

Geometry & Structure In Section 6.2, we have already shown that objective functions F in (MS) problems are nonconvex and nonsmooth even when every *piece* $f^{(i)}$ is proper convex, i.e. F is also a valid objective for (MC) $\equiv$ (SMC) in this latter case. In this paragraph, we continue to elaborate on geometrical aspects and provide some refinements for (SMC)'s structure. We start by underlining a feature shared with convex problems. Let $x \in \mathbb{R}^d$ and let us denote for every $s \in [N]$

$$\mathcal{A}^{(s)} : \mathbb{R}^d \rightrightarrows \{1, \dots, n_s\}, \quad x \rightarrow \mathcal{A}^{(s)}(x) := \{l \mid h^{(s)}(x) = h_l^{(s)}(x)\}. \quad (7.8)$$

$\mathcal{A}^{(s)}(x)$ collects which *components* $h_l^{(s)}$ are *active* at x for the term $h^{(s)}$.

If we assume that the *main function* and *component functions* are twice continuously differentiable on their respective domains, then for every $x \in \text{int}(\text{dom } \bar{h})$ fulfilling

$$\mathcal{A}^{(s)}(x) = \{\sigma_s\} \quad \forall s \in [N],$$

the matrices $\nabla^2 h^{(s)}(x) = \nabla^2 h_{\sigma_s}(x)$ are well-defined so that

$$\nabla^2 F(x) = \nabla^2 \bar{h}(x) + \frac{1}{N} \sum_{s=1}^N \nabla^2 h_{\sigma_s}^{(s)}(x) \geq \mathbf{0}_{d \times d}. \quad (7.9)$$

In other words, just like convex objectives, there exists no point in the effective domain of F where, when existing, its Hessian matrix admits a strictly negative eigenvalue (no negative curvature). Generally, even when the *components* or the *main function* fail to be differentiable on their domains, (SMC) objectives are convex on subregions of $\mathcal{X}$ that we define now.

Every σ -selection induces two (possibly nonconvex or empty) regions

$$\mathcal{R}(\sigma) := \{x \in \mathcal{X} \mid h_{\sigma_s}^{(s)}(x) \leq h_l^{(s)}(x) \quad \forall s \in [N], \forall l \in [n_s]\}, \quad (7.10)$$

$$\overset{\circ}{\mathcal{R}}(\sigma) := \{x \in \mathcal{X} \mid h_{\sigma_s}^{(s)}(x) < h_l^{(s)}(x) \quad \forall s \in [N], \forall l \in [n_s] \setminus \{\sigma_s\}\}. \quad (7.11)$$

On $\mathcal{R}(\sigma)$, $F = F_\sigma$ and every *critical point* $\hat{x} \in \overset{\circ}{\mathcal{R}}(\sigma)$, i.e. $\mathbf{0}_d \in \partial F_\sigma(\hat{x}) + \mathcal{N}_{\mathcal{X}}(\hat{x})$, must be a *local-minimum* of F on the subregion $\mathcal{R}(\sigma)$, there is no interior *local-maximum*. Indeed, by convexity of F_σ , $\hat{x}$ would then be a global minimum of F_σ over any subset $\mathcal{S} \subseteq \mathbb{R}^d$ in which $\mathcal{R}(\sigma)$ is contained. Moreover, by continuity of the *components*, there must exist a radius $\hat{\alpha} > 0$ for which every $x \in \mathcal{X} \cap \mathbb{B}_2(\hat{x}; \hat{\alpha})$ satisfies $h_{\sigma_s}^{(s)}(x) \leq h_l^{(s)}(x)$ for every $s \in [N]$ and $l \in [n_s]$. Therefore, on $\mathcal{S} = \mathcal{X} \cap \mathbb{B}_2(\hat{x}; \hat{\alpha})$, $F = F_\sigma$ is minimized at $\hat{x}$.

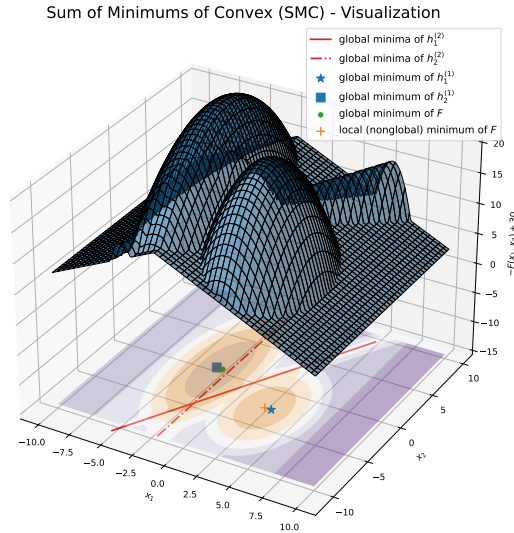


Fig. 7.1 $F(x_1, x_2) = \min\{(x_1 - 3)^2 + \frac{1}{3}(x_2 + 3)^2, (x_1 + 3)^2 + \frac{1}{6}x_2^2, 15\} + \min\{(x_2 - 2x_1 + 1)^2, |x_1 + 2|\}$

7 | Relaxed Alternating Minimization for (SMC)

Remark 7.3. On Figure 7.1, we have displayed the graph of $-F$ as well as contour lines for an example with $d = 2$, $N = 2$ ($n_1 = 3$, $n_2 = 2$), $\mathcal{X} = [-10, 10]^2$ and $\bar{h}(x) = 0$. F is nonconvex on substantial portions of the basic feasible domain $\mathcal{X}$, i.e. there exists $x, x_+ \in \mathcal{X}$ and $q \in [0, 1]$ such that $F(q \cdot x + (1 - q) \cdot x_+) > q \cdot F(x) + (1 - q) \cdot F(x_+)$. However, one can spot a global optimum of F as being the global optimum of a convex function $(1/2) \cdot (h_2^{(1)} + h_2^{(2)})$, i.e. $F^* = \min_{x \in \mathcal{X}} F_{(2,2)}(x)$ and $\mathcal{X}^* = \{-(\frac{5}{2}, 0)\}$.

To conclude this paragraph, building upon the (DC) representation of (MC) objectives (4.7), we also specify a possible (DC) representation of (SMC). For any $x \in \mathbb{R}^d$,

$$F(x) = \underbrace{\bar{h}(x) + \frac{1}{N} \sum_{s=1}^N \sum_{l \in [n_s]} h_l^{(s)}(x)}_{f_1(x)} - \underbrace{\frac{1}{N} \max_{\bar{l} \in [n_s]} \sum_{l \in [n_s] \setminus \{\bar{l}\}} h_l^{(s)}(x)}_{f_2(x)}. \quad (7.12)$$

We understand as *critical point* of (SMC) a *critical point* (cfr. (DC-criticality)) of the equivalent (DC) program. Moreover, the decomposition (7.12) paves the way for the use of (DC) algorithms (e.g. DCA [LP18]) as a baseline for our numerical experiments.

Motivation Since the brute force enumeration approach (7.1) quickly becomes impractical as N grows ($n = \prod_{s=1}^N n_s$, recalling Eq. (7.2)), one question naturally emerges.

Can we exploit the structure in (SMC) to (i) implement an efficient/scalable local-search heuristic, (ii) possibly endowed with some local optimality guarantees ?

About (i), we answer by the affirmative. Rewriting (SMC) as a specific bi-convex problem, the possibility to apply alternating minimization (AM) procedures is unveiled. One only requires *off-the-shelf* softwares, e.g. CVXPY [DB16], to solve convex subproblems, e.g. (Oracle), and the ability to evaluate *component functions* at any produced iterate. Relying on both these tools, we come up with a family of new local-search methods, which we call *relaxed alternating minimization* (RAM), empirically providing better solutions than AM on our benchmark tests. RAM, but also AM as a limiting case, scale well with N in the sense that we prove their convergence in $O(\delta^{-1})$

iterations, each of which involving one call to both oracles, to reach any level $\delta > 0$ of approximate optimality measure. This latter coincides with *criticality* for $\delta = 0$.

About (ii), we nuance our answer. We leverage a new lemma tailored for (SMC) (Lemma 7.17), from which we deduce sufficient conditions (Corollary 7.18) for a candidate $\hat{x} \in \mathcal{X}$ to be *locally optimal*. We show that it amounts to solving a reduced scale instance of the original (SMC) problem on a local neighbourhood of $\hat{x}$. This scale relates to $\Pi_{s=1}^N |\mathcal{A}^{(s)}(\hat{x})|$ (recall Eq. (7.8)), an integer called *degeneracy factor*.

- If the *degeneracy factor* is small enough, one can implement a direct exhaustive scanning procedure to check for *local optimality*.
- Otherwise, we propose to use a mixed-integer formulation for (SMC) problems whose continuous relaxations are convex. Hopefully, the existence of dedicated *branch-and-cut* algorithms (see e.g. [KBLG18]) might help in practice to solve the aforementioned reduced scale instance, also allowing to check for *local optimality*.

That being stated, the reduced scale problem might be as hard to solve as the original one in some unlucky circumstances, e.g. the order of the *degeneracy factor* is $\mathcal{O}(n)$. Without further assumptions on the feasible set $\mathcal{X}$ and objective F , guaranteeing *local optimality* remains difficult from a theoretical point of view.

To get further insight, we present hereafter our most exciting example, instance of (data-SMC), closely related to **multiple choices** type.

Example 7.4. (Piecewise-Linear L_1 -Regression, [BTK⁺22]) Due to their appealing universal approximator property [Hua20], piecewise-linear models provide an excellent trade-off between complexity and performance. Considering a L_1 -regression task, given a dataset of training examples from $\mathbb{R}^{p+1}$, i.e. $\{(\gamma^{(s)}, \bar{\beta}^{(s)})\}_{s=1}^N$, one would like build a piece-wise linear model $\Delta\ell(\cdot; x)$ that predicts the most accurately possible $\gamma^{(s)} \in \mathbb{R}$ as $\Delta\ell(\bar{\beta}^{(s)}; x)$ by minimizing its *mean absolute deviation* regarding the examples, i.e.

$$\min_{x \in \mathcal{X}} \frac{1}{N} \sum_{s=1}^N |\gamma^{(s)} - \Delta\ell(\bar{\beta}^{(s)}; x)|. \quad (7.13)$$

7 | Relaxed Alternating Minimization for (SMC)

The prediction of the model $\Delta\ell(\cdot; x)$ is defined for every $\bar{\beta} \in \mathbb{R}^p$ as

$$\Delta\ell(\bar{\beta}; x) = \underbrace{\max_{e_1 \in [B_1]} \{\langle \bar{\beta}, x_{e_1}^{(1)} \rangle\}}_{\ell_1(\bar{\beta}; x)} - \underbrace{\max_{e_2 \in [B_2]} \{\langle \bar{\beta}, x_{e_2}^{(2)} \rangle\}}_{\ell_2(\bar{\beta}; x)} \quad (7.14)$$

The model is parametrized by a vector of coefficients x of size $p \cdot (B_1 + B_2)$,

$$x = (x_1^{(1)}, \dots, x_{B_1}^{(1)}, x_1^{(2)}, \dots, x_{B_2}^{(2)}).$$

Usually, one sets $\mathcal{X} = \{u \in \mathbb{R}^p \mid \|u\|_\infty \leq L\}^{B_1+B_2}$ to ensure the $2L$ Lipschitz continuity of $\Delta\ell(\cdot; x)$. As we prove right after, problem (7.13) fits in (data-SMC) with

$$\begin{aligned} \bar{h}(x) = \frac{1}{N} \sum_{s=1}^N \max \{ & \gamma^{(s)} + \ell_2(\bar{\beta}^{(s)}; x), \ell_1(\bar{\beta}^{(s)}; x) \} \\ & + \max \{ -\gamma^{(s)} + \ell_1(\bar{\beta}^{(s)}; x), \ell_2(\bar{\beta}^{(s)}; x) \}, \end{aligned} \quad (7.15)$$

$$h_l(x \mid \bar{\beta}) = -\langle \bar{\beta}, x_{e_1(l)}^{(1)} + x_{e_2(l)}^{(2)} \rangle, \quad (7.16)$$

as well as $\bar{n} = B_1 \cdot B_2$, $e_1(l) = \lceil l/B_2 \rceil$ and finally $e_2(l) = 1 + (l-1) \bmod B_2$. We derive now the above formulas. Let us consider a tuple $(\gamma, \bar{\beta}) \in \mathbb{R} \times \mathbb{R}^p$. Now, let $\epsilon = |\gamma - \Delta\ell(\bar{\beta}; x)|$. Two scenarios are possible, either $\epsilon = \epsilon_+$ or $\epsilon = \epsilon_-$ with

$$\begin{cases} \epsilon_+ = \gamma + \ell_2(\bar{\beta}; x) - \ell_1(\bar{\beta}; x) \geq 0, \\ \epsilon_- = -\gamma + \ell_1(\bar{\beta}; x) - \ell_2(\bar{\beta}; x) \geq 0. \end{cases}$$

and

$$\ell_\star(\cdot; x) := \max_{e_\star \in [B_\star]} \langle \cdot, x_{e_\star}^{(\star)} \rangle \quad \forall \star \in \{1, 2\}.$$

It is easy to see that $\epsilon = \max\{0, \epsilon_+\} + \max\{0, \epsilon_-\}$, the sum of two polyhedral (DC) functions in x hence itself (DC). Let us rewrite $\max\{0, \epsilon_+\}$ as

$$\begin{aligned} \max\{0, \epsilon_+\} &= \max \{ \gamma + \ell_2(\bar{\beta}; x), \ell_1(\bar{\beta}; x) \} - \ell_1(\bar{\beta}; x) \\ &= \max \{ \gamma + \ell_2(\bar{\beta}; x), \ell_1(\bar{\beta}; x) \} + \min_{e_1 \in [B_1]} -\langle \bar{\beta}, x_{e_1}^{(1)} \rangle \end{aligned}$$

and, similarly, $\max\{0, \epsilon_{-}\}$ as

$$\begin{aligned}\max\{0, \epsilon_{-}\} &= \max \left\{ -\gamma + \ell_1(\bar{\beta}; x), \ell_2(\bar{\beta}; x) \right\} - \ell_2(\bar{\beta}; x) \\ &= \max \left\{ -\gamma + \ell_1(\bar{\beta}; x), \ell_2(\bar{\beta}; x) \right\} + \min_{e_2 \in [B_2]} -\langle \bar{\beta}, x_{e_2}^{(2)} \rangle\end{aligned}$$

Applying the above decomposition on every term of objective's sum yields

$$\begin{aligned}F(x) &= \frac{1}{N} \sum_{s=1}^N \epsilon^{(s)} \\ &= \frac{1}{N} \sum_{s=1}^N \max\{0, \epsilon_{+}^{(s)}\} + \max\{0, \epsilon_{-}^{(s)}\} \\ &= \bar{h}(x) + \frac{1}{N} \sum_{s=1}^N \min_{(e_1, e_2) \in [B_1] \times [B_2]} -\langle \bar{\beta}^{(s)}, x_{e_1}^{(1)} + x_{e_2}^{(2)} \rangle \\ &= \bar{h}(x) + \frac{1}{N} \sum_{s=1}^N \min_{l \in [\bar{n}]} h_l(x \mid \bar{\beta}^{(s)}).\end{aligned}$$

with $\bar{h}$ as described in equation (7.15).

7.1.1 Related work

The present framework entails many optimization problems of interest. Among them, the **multiple choices** type of (**data-SMC**), seems to have attracted the most attention. To the best of our knowledge, authors of [RSU05] first used the designation *sum of minima of convex* functions. The scope of their paper is to run the (zero-order) discrete gradient method (see e.g. [BKB08] for an updated version) on tasks like generalized clustering as well as Bradley-Mangasarian approximation of finite sets [RGZ05]. They also tried the hybrid discrete gradient/simulated annealing method from [BZ03] as heuristic to escape from bad sub-optimal valleys. Roughly speaking, this approach only relies on function evaluations and does not fully take advantage of (DC)'s structure nor (SMC)'s. In contrast, [ACV22], and later [TFV23], devised algorithms fundamentally based on (DC)'s decomposition (7.12) to tackle clustering with constraints. Both papers displayed good empirical performances of their algorithms and proved that every converging subsequence of their generated iterates leads to an accumulation point that is *critical*. Closer to our framework, authors in [LJ19]

address (SMC) problems of **range cuts** type, e.g. signal-restoration, clipped linear regression or more generally, statistical learning subject to outlier detection. Their methodology aims at solving (SMC) globally thanks to a spatial segmentation of the ambient space $\mathcal{X}$ into regions $\mathcal{R}$, as in (7.10). However, as soon as $d \geq 3$, this method hardly remains useful in practice. Moreover, one needs an efficient tool to enumerate all the intersections of $\mathcal{R}$ -boundaries, a very complicated task in the absence of closed-forms, even when $d = 2$.

The terminology of *min-convex* also appears in [DT19] where it is shown that the proximal operator of a (SMC) objective, which can be cast into a (SMC) problem, satisfies a property of *union averaged nonexpansiveness*. Nevertheless, authors do not provide specific algorithms² to compute the associated proximal mapping and their findings remain theoretical in what concerns our applications, e.g. Example 7.4. Very recently, [DCWY24] proposed a new inertial variant of a generalized Lloyd's algorithm³, named `inLloyd`, applicable on (SMC) instances that are of **multiple choices** type,

$$x = (x_1, \dots, x_{\bar{n}}) \rightarrow h^{(s)}(x) = \min_{l \in [\bar{n}]} H^{(s)}(x_l) \quad \forall s \in [N],$$

with $H^{(s)}$ strongly convex. In this precise situation, in the spirit of `kmeans++` [AV07], they come up with a randomized strategy to set up the initial values of the replicas $\{x_l\}_{l=1}^{\bar{n}}$ with proven quality guarantees. Unfortunately, we were not able to extend their methodology to initialize instances of (SMC) in the general case.

The present work places itself in the direct continuity of [BAB20]. Just like [LJ19], the paper [BAB20] is devoted to tackle (SMC) instances for which every term in the sum is the pointwise minimum between a convex function and a simple constant value λ (i.e. $n_s = 2$ for every $s \in [N]$). Therein, authors call such terms *clipped convex functions*.

$$F_{\text{clipped}}^* = \min_{x \in \mathcal{X}} \bar{h}(x) + \frac{1}{N} \sum_{s=1}^N \min\{h_1^{(s)}(x), \lambda\}. \quad (\text{SCC})$$

²Other than the naive `enumeration` (7.1), not suitable when N is medium to large.

³For clustering tasks, Lloyd's algorithm corresponds to alternating minimization `AM`.

Their contribution is twofold. First, they highlight a bi-convex reformulation of (SCC). Let $Q = Q = (q^{(1)}, \dots, q^{(N)})$, one writes equivalently

$$F_{\text{clipped}}^* = \min_{Q \in [0,1]^N} \min_{x \in \mathcal{X}} \underbrace{\bar{h}(x) + \frac{1}{N} \sum_{s=1}^N q^{(s)} \cdot h_1^{(s)}(x) + (1 - q^{(s)}) \cdot \lambda}_{\bar{F}^*(Q)} \quad (7.17)$$

Out of (7.17), one usually alternates between an exact minimization with respect to Q and x . This represents the methodology of AM. However, authors of [BAB20] suggest to perform inexact Q -updates, i.e.

$$q_+^{(s)} = \Pi_{[0,1]^N}(q^{(s)} - \kappa \cdot \text{sign}(h_1^{(s)}(x) - \lambda)) \quad \forall s \in [N], \quad (7.18)$$

for a fixed $\kappa \in (0, 1]$ ($\kappa = 1$ implementing usual AM). As a side note, we point out the following fact. Where $\bar{F}^*(Q)$ is locally differentiable with respect to Q , one can interpret (7.18) as projected signed gradient steps [MLP19].

Second, they devise a mixed-integer convex program equivalent to (SCC),

$$\begin{aligned} F_{\text{clipped}}^* = \min_{x \in \mathcal{X}} \quad & \frac{1}{N} \sum_{s=1}^N (\bar{h})^\dagger(x - z^{(s)}, 1 - q^{(s)}) + (\bar{h})^\dagger(z^{(s)}, q^{(s)}) \quad (7.19) \\ & + \frac{1}{N} \sum_{s=1}^N (h_1^{(s)})^\dagger(z^{(s)}, q^{(s)}) + (1 - q^{(s)}) \cdot \lambda \\ \text{s.t.} \quad & q^{(s)} \in \{0, 1\}, z^{(s)} \in \mathcal{X} \quad \forall s \in [N] \end{aligned}$$

where a (convex) perspective function $(h)^\dagger$ is obtained from any closed convex⁴ h as,

$$(h)^\dagger : \mathbb{R}^d \times \mathbb{R}_+, \quad (x, q) \rightarrow (h)^\dagger(x, q) = \begin{cases} q \cdot h(x/q) & q > 0 \\ 0 & q = 0, x = \mathbf{0}_d \\ +\infty & \text{otherwise.} \end{cases} \quad (7.20)$$

To obtain (7.20) and the property that $(h)^\dagger$ is well-defined and convex, it is assumed in [BAB20] that $\bar{h}$ is superlinear and $\mathbf{0}_d \in \text{dom}(\bar{h})$. Note these requirements slightly decrease the generality of the approach. As the authors noticed, aside from being intrinsically hard as a mixed-integer convex pro-

⁴We recall that *components* as well as the *main function* are proper closed and convex.

gram (MICP), formulation (7.19) exhibits $N \cdot d$ extra continuous variables $\{z^{(s)}\}_{s=1}^N$ (i.e. compared to the raw problem (7.17)) in addition to the seemingly unavoidable N binary variables $\{q^{(s)}\}_{s=1}^N$. From a practical aspect, this renders the resolution of (7.19) quite time consuming (or impossible) even if the problem is only restricted to a smaller domain portion $\mathcal{S} \subset \mathcal{X}$. As argued in [BAB20], solving (7.19) while relaxing integrality constraints on Q yields a lower-bound on F_{clipped}^* and a possible starting iterate in the x -coordinates. In this case, to circumvent the dimensionality drawback, it is possible to apply ADMM procedures [BPC⁺11] to deal with a *consensus form* of the continuous relaxation of (7.19). Both the aforementioned contributions initiated the present work. Indeed, our first goal had been to provide extensions of models (7.17) and (7.19) to fit in the general (SMC) picture.

Finally, we also mention that a last related paper [ZJ23] appeared in the literature. Therein, a first step towards more versatility is undertaken. Indeed, the authors consider again a (SMC) instance for which $n_s = 2$ for every $s \in [N]$ but this time, $h_2^{(s)}$ needs not to be a constant. However, they implicitly assume that $\bar{h} \in \mathcal{C}^1(\mathbb{R}^d)$. Their methodology works as follows. By approximating each pointwise minimum in the (SMC) sum by its own smooth minimum surrogate, they leverage the power of (L-)BFGS (unconstrained case) or any black-box method dedicated to constrained smooth nonconvex optimization. Overall, according to the numerical experiments conducted in [ZJ23], this new approach performed equally as well as [BAB20] on similar problems.

7.1.2 Contributions

Now that the framework of (SMC) has been thoroughly introduced, we summarize below our own specific contributions. We emphasize that all our numerical experiments (and, soon, a free to use new dedicated CVXPY bundle called `smc`, improving upon `sccf` from [BAB20]), are available on our [GitHub](#) repository. Focusing on mathematical optimization contributions, we point out the following elements:

- First, we detail how (SMC) can be rewritten as a *big-M* (MICP) (Section 7.2). In stark contrast with the perspective reformulation (7.19) of [BAB20], for which we also provide an extension with arbitrary $n_s \in \mathbb{N}$ for every $s \in [N]$ in (7.38), our (MICP) does not involve $\sum_{s=1}^N n_s$ copies of the decision vector x , hence d (i.e. dimension of x) times as many extra variables.

- Second, building upon (Proposition 1.4) from [DG24a], we restrict our global (MICP) formulation on local neighbourhoods (Section 7.2.2). As a byproduct of our local (MICP) formulation, we state *equivalent local optimality* conditions (Lemma 7.17). From these latter, useful *sufficient* conditions (Corollary 7.18) are derived to possibly go beyond *criticality*, the usual standard in (DC) optimization.
- Third, we expose the bi-convex (BIC) formulation of general (SMC) (Section 7.2.3). We review plain alternating minimization (AM) applied on this formulation to underpin another new lemma (Lemma 7.22) describing *criticality* sufficient conditions. This lemma together with sufficient descent conditions will serve as backbones to develop our new family of heuristics, i.e. relaxed alternating minimization (r-AM) methods. We prove that every accumulation point of the main sequence of r-AM iterates is *critical* for problem (SMC) (Theorem 7.30).
- Finally, we display the results of numerical experiments (Section 7.4), demonstrating the practical efficiency of two r-AM methods of ours when tackling (SMC) on *piecewise-linear regression* or *facility location* problems. In addition, thanks to the aforementioned *sufficient* conditions (Corollary 7.18), we show that we were sometimes successful in certifying *local optimality* of r-AM iterates (Section 7.4.3).

7.1.3 Preliminaries

Akin to Definition 6.2 from the previous chapter, we extend the statement (7.8) to capture not only active *components* at $x \in \mathcal{X}$ but also ρ -close to active ones. The straightforward translation of Definition 6.2 applied on every term $h^{(s)}$ of objective's F sum in (SMC) would be as follows:

Definition 7.5 (ρ - active sets). Let $\rho \geq 0$. For every $s \in [N]$, we call the set

$$\mathcal{A}_\rho^{(s)}(x) := \left\{ \tilde{l} \mid h_{\tilde{l}}^{(s)}(x) - h^{(s)}(x) \leq \rho \right\} \quad (7.21)$$

the (ρ -) *active set* at $x \in \mathcal{X}$ of the s -th term.

By convention, $\mathcal{A}^{(s)}(x) = \mathcal{A}_0^{(s)}(x)$ is called *active set* at x of the s -th term.

7 | Relaxed Alternating Minimization for (SMC)

We also advocate another relaxation of *active sets* (Definition 7.5), understood in a relative sense. We found it convenient to use in Section 7.4.3.

Definition 7.6 (ρ - relative active sets). Let $\rho \in [0, 1]$. For every $s \in [N]$, we call the set

$$\bar{\mathcal{A}}_\rho^{(s)}(x) := \left\{ \tilde{l} \mid h_{\tilde{l}}^{(s)}(x) - h^{(s)}(x) \leq \rho \cdot \left(\max_{l \in [n_s]} h_l^{(s)}(x) - h^{(s)}(x) \right) \right\} \quad (7.22)$$

the (ρ -)relative active set at $x \in \mathcal{X}$ of the s -th term.

Proposition 7.7. Let $\hat{x} \in \mathcal{X}$. There exists $\alpha > 0$ such that for every $\rho \geq 0$,

$$\mathcal{A}^{(s)}(x) \subseteq \mathcal{A}_\rho^{(s)}(\hat{x}) \quad \forall x \in \mathcal{X} \cap \mathbb{B}_2(\hat{x}; \alpha), \quad \forall s \in [N]. \quad (7.23)$$

Proof. According to Proposition 6.3 applying verbatim, for every $s \in [N]$ there exists $\alpha^{(s)} > 0$ such that $\mathcal{A}^{(s)}(x) \subseteq \mathcal{A}_\rho^{(s)}(\hat{x})$ for every $x \in \mathcal{X} \cap \mathbb{B}_2(\hat{x}; \alpha^{(s)})$. Then, (7.23) simply holds by taking $\alpha = \min_{s \in [n]} \alpha^{(s)}$. $\square$

Assumption 7.8. We end this section with the following assumption.

- One has access to a deterministic black-box algorithm that will return for any fixed $Q = (q^{(1)}, \dots, q^{(N)}) \in \times_{s=1}^N \Delta^{(n_s)}$ the same output

$$x_{|Q}^* \in \arg \min_{x \in \mathcal{X}} \bar{h}(x) + \frac{1}{N} \sum_{s=1}^N \sum_{l=1}^{n_s} q_l^{(s)} h_l^{(s)}(x). \quad (\text{A})$$

Remark 7.9. Note that if every $q^{(s)}$ is a vertex of $\Delta^{(n_s)}$, i.e. there exists $\sigma_s \in [n_s]$ such that $q_{\sigma_s}^{(s)} = 1$ and $q_l^{(s)} = 0$ for every $l \in [n_s] \setminus \{\sigma_s\}$, Assumption (A) allows to solve (Oracle) subproblem with $\sigma = (\sigma_1, \dots, \sigma_N)$. In that sense, it is stronger than simply requiring the ability to solve (Oracle) for any σ -selection. Yet, we emphasize that many (SMC) problems share the feature that *components* belong to the same optimization class, e.g. linear in Example 7.4. Therefore, if this class is closed under convex combinations (through weights $q^{(s)}$ for every $s \in [N]$), Assumption (A) and (Oracle) represent the same level of optimization difficulty.

7.2 Equivalent representations of (SMC)

In this section we aim at explaining the core procedure to reformulate problem (SMC) as a *big-M* (MILP). We first need to introduce a localized definition of *big-M* bounds in what concerns (SMC). That is, given a zone $\mathcal{S} \subseteq \mathbb{R}^d$, one needs a finite bound on the maximal difference of value (on $\mathcal{S}$) achievable by every pair of *components*. These bounds will come into play to provide an epigraph representation of objective F , restricted to $\mathcal{S}$, using convex constraints and binary variables. To construct an equivalent formulation of (SMC), one needs a representation of F that is, *at least*, valid on the basic feasible domain $\mathcal{X}$, hence the necessity of Assumption (B). However, when one only wants a local representation, computing these bounds on a small $\mathcal{S} \not\supseteq \mathcal{X}$ will prove useful (see Section 7.4.3). That being said, this new assumption is satisfied for our applications, including the interesting *piecewise-linear regression* (Example 7.4).

Definition 7.10 ($\mathcal{S}$ -bounds). Let $\mathcal{S} \subseteq \mathbb{R}^d$. We call $\mathcal{S}$ -bounds values $M_{l+,l}^{(s)} < \infty$ for every $s \in [N]$ and $(l+, l) \in [n_s]^2$ such that $M_{l+,l}^{(s)} = 0 = M_{l+,l+}^{(s)}$ and

$$M_{l+,l}^{(s)} \geq \max_{u \in \mathcal{S}} h_{l+}^{(s)}(u) - h_l^{(s)}(u) \quad \text{s.t.} \quad h_l^{(s)}(u) = h^{(s)}(u). \quad (7.24)$$

By convention, if $\{u \in \mathcal{S} \mid h_l^{(s)}(u) = h^{(s)}(u)\} = \emptyset$, i.e. when *component* $h_l^{(s)}$ is nowhere *active* on $\mathcal{S}$ regarding the term $h^{(s)}$, one should set $M_{l+,l}^{(s)} = -\infty$.

Assumption 7.11. Accordingly, we state a second core assumption.

We dispose of $\mathcal{S}$ -bounds for problem (SMC) with $\mathcal{S} \supseteq \mathcal{X}$. (B)

Remark 7.12. Definition 7.10 fixes which properties our own so-called *big-M* constants must satisfy. As we will see in Section 7.2.1, akin to [TG15], their role is to model a selection between multiple incentives. Here, if l is not the index of the selected *component* for the s -th term ($s \in [N]$), then it must be some $l' \in [n_s] \setminus \{l\}$ and the inequality $h_l^{(s)}(x) \leq \nu^{(s)} + M_{l,l'}^{(s)}$ should hold at any $x \in \mathcal{X}$ (Assumption (B)) for $\nu^{(s)} = h_{l'}^{(s)}(x)$.

Computability of $\mathcal{S}$ -bounds As such, even if we relax the nonconvex constraints on the right-hand side of (7.24), the problem defining our *big-M* bounds remains of (DC) nature. If the dimension d is (very) small and the topology of $\mathcal{X}$ polyhedral, one might resort to global optimization techniques [PU23] to get a tight value of the corresponding *big-M* constant. Otherwise, the general problem is not tractable. We identify however generic cases for which one can compute upper-bounds with $\mathcal{S} = \mathcal{X}$.

- $h_l^{(s)}(\cdot)$ has L -Lipschitz continuous gradient with respect to a norm $\|\cdot\|$ and $\mathcal{X}$ is bounded with diameter $D = \sup_{(x,y) \in \mathcal{X}^2} \|x - y\| < \infty$.

$$M_{l+,l}^{(s)} \leftarrow \max_{u \in \mathcal{X}} h_{l+}^{(s)}(\bar{x}) + \langle \nabla h_{l+}^{(s)}(\bar{x}), u - \bar{x} \rangle + \frac{L}{2} D^2 - h_l^{(s)}(u) \quad (7.25)$$

for any reference $\bar{x} \in \mathcal{X}$ chosen. We emphasize that problem (7.25) is convex. This latter can be solved for more than one reference $\bar{x}$, the *big-M* constant would then be chosen as the smallest encountered.

- $h_l^{(s)}(\cdot) = \max_{e \in [B]} \gamma_{l,e}^{(s)} + \langle \beta_{l,e}^{(s)}, \cdot \rangle$ and $\mathcal{X}$ is bounded.

$$M_{l+,l}^{(s)} \leftarrow \max_{e \in [B]} \left\{ \max_{u \in \mathcal{X}} \gamma_{l,e}^{(s)} + \langle \beta_{l,e}^{(s)}, u \rangle - h_l^{(s)}(u) \right\}. \quad (7.26)$$

One gets $M_{l+,l}^{(s)}$ by solving B convex programs, e.g. LPs if $\mathcal{X}$ is polyhedral.

- Any Trust Region Subproblem (TRS) like setup $(h_{l+}^{(s)}(\cdot), h_l^{(s)}(\cdot), \mathcal{X})$ matching requirements of [HNKK17]. For example, let us describe the classical (TRS) case. Let $R > 0$ and $\mathcal{X} = \mathbb{B}_2(\mathbf{0}_d; R)$. If for every $x \in \mathcal{X}$,

$$h_{l+}^{(s)}(x) - h_l^{(s)}(x) = \frac{1}{2} \langle \mathbf{V}_{l+,l}^{(s)} x, x \rangle + \langle \bar{v}_{l+,l}^{(s)}, x \rangle + \gamma_{l+,l}^{(s)}$$

for a symmetric matrix $\mathbf{V}_{l+,l}^{(s)} \in \mathbb{R}^{d \times d}$, vector $\bar{v}_{l+,l}^{(s)} \in \mathbb{R}^d$ and $\gamma_{l+,l}^{(s)} \in \mathbb{R}$ then

$$-M_{l+,l}^{(s)} \leftarrow \min_{u \in \mathcal{X}} - \left(\frac{1}{2} \langle \mathbf{V}_{l+,l}^{(s)} u, u \rangle + \langle \bar{v}_{l+,l}^{(s)}, u \rangle + \gamma_{l+,l}^{(s)} \right) + \min \left\{ \frac{\lambda_{l+,l}^{(s)}}{2}, 0 \right\} (R^2 - \|u\|_2^2). \quad (7.27)$$

where $\lambda_{l+,l}^{(s)} = \lambda_{\min}(-\mathbf{V}_{l+,l}^{(s)})$. The resulting problem (7.27) is convex [SXZ22].

- $h_l^{(s)}(\cdot)$ is bounded from below by $\check{H}_l^{(s)} > -\infty$ and $h_{l+}^{(s)}(\cdot)$ is bounded from above by $\hat{H}_{l+}^{(s)} < \infty$. Then, it trivially comes

$$M_{l+,l}^{(s)} \leftarrow \hat{H}_{l+}^{(s)} - \check{H}_l^{(s)}. \quad (7.28)$$

Remark 7.13. If $n_s = \bar{n}$ for every $s \in [N]$, as it is the case for e.g. (data-SMC), there are $\mathcal{O}(N\bar{n}^2)$ *big-M* parameters to compute to obtain the whole collection of $\mathcal{S}$ -bounds.

7.2.1 Global (MICP) Model

The adequate tools have been introduced, we can recast (SMC) as a (MICP).

big-M formulation For the purpose of this formulation, we apply the standard *epigraph* technique on every term $h^{(s)}$ of (SMC)'s sum. Specifically, for every $s \in [N]$, we introduce a continuous variable $\eta^{(s)}$ and we need to rewrite

$$h^{(s)}(x) \leq \eta^{(s)} \quad (7.29)$$

as an equivalent set of constraints, eventually including binary variables, whose continuous relaxations are convex. Then, instead of the initial objective, we can minimize

$$\bar{h}(x) + \frac{1}{N} \sum_{s=1}^N \eta^{(s)} \quad (7.30)$$

on the new feasible set, i.e. constituted of the aforementioned constraints as well as the initial feasible set $\mathcal{X}$. To encode (7.29), we resort to n_s new binary variables $t^{(s)} = (t_1^{(s)}, \dots, t_{n_s}^{(s)}) \in \{0, 1\}^{n_s}$ and impose that exactly one of them equals 1, i.e.

$$\sum_{l=1}^{n_s} t_l^{(s)} = 1. \quad (7.31)$$

Let $t_\sigma^{(s)} = 1$ for some $\sigma \in [n_s]$. We recall that, by definition, if $h_l^{(s)}(x) \leq \eta^{(s)}$ for any $l \in [n_s]$ then $h^{(s)}(x) \leq \eta^{(s)}$. We consider the following inequalities for every $l_+ \in [n_s]$,

$$h_{l+}^{(s)}(x) \leq \eta^{(s)} + \sum_{l=1}^{n_s} t_l^{(s)} M_{l+,l}^{(s)}. \quad (7.32)$$

7 | Relaxed Alternating Minimization for (SMC)

Among inequalities (7.32) is hidden the required (7.29) since

$$h_{\sigma}^{(s)}(x) \leq \eta^{(s)} + M_{\sigma,\sigma}^{(s)} = \eta^{(s)} \Rightarrow h^{(s)}(x) \leq \eta^{(s)} \quad (7.33)$$

where we used (7.31) and the fact that, by definition, $M_{l,l}^{(s)} = 0$ for every $l \in [n_s]$. For every $l_+ \in [n_s] \setminus \{\sigma\}$, we verify now that inequalities (7.32) are not restrictive. We start by noticing that for any $l \in \mathcal{A}^{(s)}(x)$, it **always** holds

$$\begin{aligned} h_{l_+}^{(s)}(x) - h_{\sigma}^{(s)}(x) &\leq h_{l_+}^{(s)}(x) - h_l^{(s)}(x) \\ &\leq \max_{u \in \mathcal{X}} h_{l_+}^{(s)}(u) - h_l^{(s)}(u) \quad \text{s.t.} \quad h_l^{(s)}(u) = h^{(s)}(u) \\ &\leq M_{l_+,l}^{(s)}. \end{aligned} \quad (7.34)$$

Then, it readily follows from (7.33) and (7.34) that

$$h_{l_+}^{(s)}(x) \leq h_{\sigma}^{(s)}(x) + M_{l_+,l}^{(s)} \leq \eta^{(s)} + M_{l_+,l}^{(s)}. \quad (7.35)$$

In the mean time, at any minimizer x of (7.30), it is **impossible** that $t_{\sigma}^{(s)} = 1$ and $\sigma \notin \mathcal{A}^{(s)}(x)$ (which would mean $h_l^{(s)}(x) < h_{\sigma}^{(s)}(x)$) because choosing $t_l^{(s)} = 1$ instead would allow $\eta^{(s)}$ to decrease, achieving thereby a strictly better objective value. This alternative choice is possible since x variable would remain feasible. Indeed, for every $l_+ \in [n_s] \setminus \{l\}$, inequality (7.32) boils down to (7.35) that is **always** satisfied under these conditions (i.e. $t_l^{(s)} = 1$ with $l \in \mathcal{A}^{(s)}(x)$) and (7.33) simply turns into $h_l^{(s)}(x) \leq \eta^{(s)}$.

Thus, we can conclude that $t_{\sigma}^{(s)} = 1$ enforces $\sigma \in \mathcal{A}^{(s)}(x)$ at the optimum and, thereby, we can safely lay down the equivalence

$$h^{(s)}(x) \leq \eta^{(s)} \Leftrightarrow \sum_{l=1}^{n_s} t_l^{(s)} = 1, \quad h_{l_+}^{(s)}(x) \leq \eta^{(s)} + \sum_{l=1}^{n_s} t_l^{(s)} M_{l_+,l}^{(s)} \quad \forall l_+ \in [n_s]. \quad (7.36)$$

We have now a formulation that completely avoids the nonconvex term $h^{(s)}$ in (7.29) but at the expense of additional binary variables.

Based on (7.36), we can write the following parametric mixed-integer convex program, i.e. $F^*(C)$ for every $C \in [0, 1]$. It involves *value functions* V_C as depicted in Figure 7.2. For every $C \in [0, 1]$, $F^*(C) = \min_{x \in \mathcal{X}} V_C(x)$ where

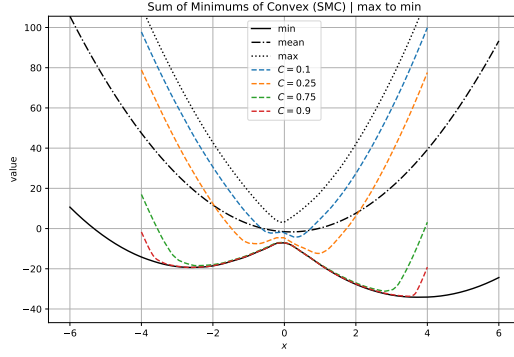


Fig. 7.2 value functions | $C = 0 \rightarrow$ sum of max., $C = 1 \rightarrow$ sum of min.

$$\begin{aligned}
 V_C(x) &:= \min_{(\eta^{(1)}, \dots, \eta^{(N)}) \in \mathbb{R}^N} \bar{h}(x) + \frac{1}{N} \sum_{s=1}^N \eta^{(s)} & (7.37) \\
 \text{s.t. } & t^{(s)} \in \{0, 1\}^{n_s}, \quad \sum_{l=1}^{n_s} t_l^{(s)} = 1 & \forall s \in [N] \\
 & h_{l+}^{(s)}(x) \leq \eta^{(s)} + C \cdot \sum_{l=1}^{n_s} t_l^{(s)} M_{l+,l}^{(s)} \quad \forall s \in [N], \quad \forall l_+ \in [n_s].
 \end{aligned}$$

One gets the (MICP) formulation of (SMC) by fixing $C = 1$, i.e. $F^*(1) = F^*$.

Continuum from max to min One can interpret the goal of minimizing the sum of maximums of convex functions (i.e. **max** hereafter) as $F^*(0)$.

$$\begin{aligned}
 \mathbf{max} &\equiv \min_{x \in \mathcal{X}} \bar{h}(x) + \frac{1}{N} \sum_{s=1}^N \max_{l \in [n_s]} h_l^{(s)}(x) \quad (\text{convex}) \\
 \mathbf{mean} &\equiv \min_{x \in \mathcal{X}} \bar{h}(x) + \frac{1}{N} \sum_{s=1}^N \sum_{l=1}^{n_s} \frac{1}{n_s} h_l^{(s)}(x) \quad (\text{convex}) \\
 \mathbf{min} &\equiv \min_{x \in \mathcal{X}} \bar{h}(x) + \frac{1}{N} \sum_{s=1}^N \min_{l \in [n_s]} h_l^{(s)}(x) \quad (\text{nonconvex})
 \end{aligned}$$

Remark 7.14. On Figure 7.2, all the component functions were chosen to be 1D convex quadratics. In our computations of the value function V_C , we used the non-uniform big- M constants as devised in the previous section at bullet point (3). Here, $\mathcal{X} = [-4, 4]$. In multiple colors, we have displayed the interpolating (from **max** to **min**) V_C functions for $C \in \{10^{-1}, 2.5 \cdot 10^{-1}, 7.5 \cdot 10^{-1}, 9 \cdot 10^{-1}\}$, all nonconvex in this case.

Perspective formulation For the sake of completeness, as a minor contribution, we extend the *perspective* eformulation [BAB20], summarized in (7.19), when $n_s \in \mathbb{N}$ for every $s \in [N]$. Again, we resort to n_s -dimensional vectors of binary variables, i.e.

$$q^{(s)} = (q_1^{(s)}, \dots, q_{n_s}^{(s)}) \in \{0, 1\}^{n_s} \quad \forall s \in [N].$$

We also need n_s copies $\{z_l^{(s)}\}_{l=1}^{n_s}$ of the main variable x . According to (7.20), imposing

$$\sum_{l=1}^{n_s} q_l^{(s)} = 1 \quad \forall s \in [N]$$

will force, for every $(s, \sigma_s) \in [N] \times [n_s]$ such that $q_{\sigma_s}^{(s)} = 1$, the equality $x = z_{\sigma_s}^{(s)}$. On the other hand, $q_l^{(s)} = 0$ implies that $z_l^{(s)} = \mathbf{0}_d$. Therefore, we write in a compact form

$$\begin{aligned} F^* = \min_{x \in \mathcal{X}} \quad & \frac{1}{N} \sum_{s=1}^N \sum_{l=1}^{n_s} (\bar{h})^\dagger(x - z_l^{(s)}, 1 - q_l^{(s)}) + (\bar{h})^\dagger(z_l^{(s)}, q_l^{(s)}) \\ & + \frac{1}{N} \sum_{s=1}^N \sum_{l=1}^{n_s} (h_l^{(s)})^\dagger(x - z_l^{(s)}, 1 - q_l^{(s)}) + (h_l^{(s)})^\dagger(z_l^{(s)}, q_l^{(s)}) \\ \text{s.t.} \quad & q^{(s)} \in \{0, 1\}^{n_s}, \quad \sum_{l=1}^{n_s} q_l^{(s)} = 1, \quad z^{(s)} \in \mathcal{X}^{n_s} \quad \forall s \in [N]. \end{aligned} \quad (7.38)$$

7.2.2 Local (MICP) Model

In this subsection, we outline how *simpler* the description of the objective F becomes locally. Using continuity arguments, we will show that only terms $h^{(s)}$ for which s belongs to the *degeneracy set* (see Definition 7.15) need to be expressed using binary variables as in (7.36). This consideration in mind, checking the *local optimality* of a candidate $\hat{x} \in \mathcal{X}$ comes at a (hopefully) reduced computational expense.

Definition 7.15 (degeneracy factor & degeneracy set). Let $\hat{x} \in \mathcal{X}$. We call the *degeneracy factor* at $\hat{x}$, the size of the set $\mathcal{A}^{(1)}(\hat{x}) \times \dots \times \mathcal{A}^{(N)}(\hat{x})$, i.e. $\prod_{s=1}^N |\mathcal{A}^{(s)}(\hat{x})|$. In addition, we call *degeneracy set* $I(\hat{x})$ at $\hat{x}$ the set of indices $s \in [N]$ such that $|\mathcal{A}^{(s)}(\hat{x})| \geq 2$, i.e. more than one *component function* is active at $\hat{x}$ with respect to (7.8).

Remark 7.16. According to Proposition 7.7, there exists $\alpha > 0$ such that for every $s \in [N]$, $\mathcal{A}^{(s)}(x) \subseteq \mathcal{A}^{(s)}(\hat{x})$ for all x belonging to the neighbourhood $\mathcal{X} \cap \mathbb{B}_2(\hat{x}; \alpha)$. Then, instead of involving N terms each of which being the pointwise minimum of n_s components, the analytical expression of objective F requires only $|I(\hat{x})|$ of them to be pointwise minima of $|\mathcal{A}^{(s)}(\hat{x})|$ components. For the other indices, i.e. $s \notin I(\hat{x})$, $\mathcal{A}^{(s)}(\hat{x}) = \{\sigma_s\}$ for some $\sigma_s \in [n_s]$. Indeed, in light of these considerations, we develop

$$\begin{aligned} F(x) &= \bar{h}(x) + \frac{1}{N} \sum_{s \in [N]} \min_{l \in \mathcal{A}^{(s)}(x)} h_l^{(s)}(x) \\ &= \bar{h}(x) + \frac{1}{N} \left(\sum_{s \in [N] \setminus I(\hat{x})} h_{\sigma_s}^{(s)}(x) \right) + \frac{1}{N} \left(\sum_{s \in I(\hat{x})} \min_{l \in \mathcal{A}^{(s)}(\hat{x})} h_l^{(s)}(x) \right). \end{aligned} \quad (7.39)$$

Building upon our *big-M* (MICP) reformulation (7.37) with $C = 1$ and (7.39), we prove a first lemma.

Lemma 7.17 (local optimality: equivalent conditions). Let $\hat{\mathcal{X}}(\hat{\alpha}) := \mathcal{X} \cap \mathbb{B}_2(\hat{x}; \hat{\alpha})$ for any $\hat{\alpha} > 0$ and $\hat{x} \in \mathcal{X}$. The point $\hat{x}$ is a local minimizer of (SMC) if and only if there exists $\hat{\alpha} > 0$ such that $F(\hat{x}) = \min_{x \in \hat{\mathcal{X}}(\hat{\alpha})} \hat{F}(x|\hat{x})$ where

$$\begin{aligned} \hat{F}(x|\hat{x}) &:= \min_{\eta \in \mathbb{R}^N} \bar{h}(x) + \frac{1}{N} \sum_{s \in [N] \setminus I(\hat{x})} h_{\sigma_s}^{(s)}(x) + \frac{1}{N} \sum_{s \in I(\hat{x})} \eta^{(s)} \quad (7.40) \\ \forall s \in I(\hat{x}), \quad t^{(s)} &\in \{0, 1\}^{n_s}, \quad \sum_{l=1}^{n_s} t_l^{(s)} = 1 \\ h_{l_+}^{(s)}(x) &\leq \eta^{(s)} + \sum_{l=1}^{n_s} t_l^{(s)} M_{l_+, l}^{(s)} \quad \forall l_+ \in \mathcal{A}^{(s)}(\hat{x}) \\ t_{l_+}^{(s)} &= 0 \quad \forall l_+ \notin \mathcal{A}^{(s)}(\hat{x}) \end{aligned}$$

where $\mathcal{A}^{(s)}(\hat{x}) = \{\sigma_s\}$ for every $s \in [N] \setminus I(\hat{x})$.

Proof. Remark 7.16 states that there exists $\alpha > 0$ so that F can be expressed as in (7.39) for any $0 < \hat{\alpha} \leq \alpha$, locally on $\hat{\mathcal{X}}(\hat{\alpha})$. To obtain the formulation at the right hand side of (7.40), we apply our *epigraph* reformulation trick (7.36) to every term $s \in I(\hat{x})$ in order to rewrite the constraints

$$\min_{l \in \mathcal{A}^{(s)}(\hat{x})} h_l^{(s)}(x) \leq \eta^{(s)}$$

7 | Relaxed Alternating Minimization for (SMC)

For every $s \notin I(\hat{x})$, one just sets $h^{(s)}(x) = h_{\sigma_s}^{(s)}(x)$ since $\mathcal{A}^{(s)}(\hat{x}) = \{\sigma_s\}$.

($\Rightarrow$) Let $\hat{x}$ be a *local minimizer* of F then there exists $\bar{\alpha} > 0$ such that $F(\hat{x}) = \min_{x \in \hat{\mathcal{X}}(\bar{\alpha})} F(x)$. Let $\hat{\alpha} = \min\{\bar{\alpha}, \alpha\}$. Thereby, $\hat{x}$ is also the minimum of F over $\hat{\mathcal{X}}(\hat{\alpha})$. On $\hat{\mathcal{X}}(\hat{\alpha})$, $\hat{F}(\cdot|\hat{x})$ and F coincide so that equality (7.40) holds.

($\Leftarrow$) If $\hat{\alpha} > 0$ exists so that $F(\hat{x}) = \min_{x \in \hat{\mathcal{X}}(\hat{\alpha})} \hat{F}(x|\hat{x})$ then for any $\bar{\alpha} \in (0, \min\{\alpha, \hat{\alpha}\}]$, $\hat{x}$ will remain a *global minimum* of $\hat{F}(\cdot|\hat{x}) = F$ over $\hat{\mathcal{X}}(\hat{\alpha})$. $\square$

Although interesting theoretically speaking, one can not try every $\hat{\alpha} > 0$ in Lemma 7.17 to prove or disprove the *local optimality* of a candidate $\hat{x} \in \mathcal{X}$. Nevertheless, fixing a suitable neighbourhood $\mathcal{S}$ around $\hat{x}$, depending on the geometry of the problem, and solving the problem at the right hand side of (7.40) will span two different outcomes both from which one can learn something. We detail this formally in the next corollary.

Corollary 7.18. Let $\hat{\alpha} > 0$ be fixed in Lemma 7.17 and $\mathcal{S} \supseteq \mathcal{X} \cap \mathbb{B}_2(\hat{x}; \hat{\alpha})$ and

$$\hat{F}_{\hat{x}, \mathcal{S}}^* = \min_{x \in \mathcal{S}} \hat{F}(x|\hat{x}). \quad (7.41)$$

Let $x^*(\hat{x}, \mathcal{S})$ denote a minimizer of (7.41). Either $F(\hat{x}) = \hat{F}_{\hat{x}, \mathcal{S}}^*$ and $\hat{x}$ is a *local minimizer* of (SMC) or $F(x^*(\hat{x}, \mathcal{S})) \leq \hat{F}_{\hat{x}, \mathcal{S}}^* < F(\hat{x})$.

Proof. In any case, $\hat{F}(\hat{x}|\hat{x}) = F(\hat{x})$ and for every $x \in \mathcal{S}$,

$$\hat{F}(x|\hat{x}) = \min_{\sigma \in \times_{s=1}^N \mathcal{A}^{(s)}(\hat{x})} F_{\sigma}(x) \geq F(x) = \min_{\sigma \in \times_{s=1}^N [n_s]} F_{\sigma}(x). \quad (7.42)$$

If $\hat{F}_{\hat{x}, \mathcal{S}}^* = F(\hat{x})$ then it is clear that $F(\hat{x}) = \min_{x \in \mathcal{X} \cap \mathbb{B}_2(\hat{x}; \hat{\alpha})} \hat{F}(x|\hat{x})$ and Lemma 7.17 applies. Otherwise, $\hat{F}(\hat{x}|\hat{x}) < F(\hat{x})$ and $F(x^*(\hat{x}, \mathcal{S})) \leq \hat{F}_{\hat{x}, \mathcal{S}}^*$ according to (7.42). $\square$

Remark 7.19. The problem in (7.41) involves $Z \in \mathbb{N}$ binary variables⁵,

$$Z = \sum_{s \in I(\hat{x})} |\mathcal{A}^{(s)}(\hat{x})|. \quad (7.43)$$

Yet, among the *possible* assignments for these variables only $\prod_{s=1}^N |\mathcal{A}^{(s)}(\hat{x})| < 2^Z$ are *feasible* and worth consideration. $\prod_{s=1}^N |\mathcal{A}^{(s)}(\hat{x})|$ is nothing else than the *degeneracy factor* at $\hat{x}$, showing the intrinsic link between the hardness

⁵Dummy variables set to 0, i.e. $t_{l+}^{(s)}$ for every $l+ \notin \mathcal{A}^{(s)}(\hat{x})$ with $s \in I(\hat{x})$, are not counted.

of certifying *local optimality* and this specific number. In the worst-case, this latter could be as large as $\prod_{s=1}^N n_s$. As we observed in our experiments, this is far from being true in practice.

Our true motivation to express $\hat{F}(\cdot|\hat{x})$ as above (7.40) is to take advantage of state-of-the-art (MICP) solvers, e.g. MOSEK [MA24], BARON [TOF24], that might compute a minimizer of (7.41) faster than a full combinatorial enumeration that we recall now

$$\hat{F}_{\hat{x},\mathcal{S}}^* = \min_{\sigma \in \times_{s=1}^N \mathcal{A}^{(s)}(\hat{x})} \left\{ \min_{x \in \mathcal{S}} \bar{h}(x) + \frac{1}{N} \sum_{s=1}^N h_{\sigma_s}^{(s)}(x) \right\} \quad (7.44)$$

Yet, if the *degeneracy factor* at $\hat{x}$ is small enough, it might be more convenient to get $\hat{F}_{\hat{x},\mathcal{S}}^*$ in a straightforward enumerative fashion since (MICP) formulations require, under the hood, additional computational work (e.g. *canonicalization*, *factorizations*, etc.). This is especially true when every σ -selection problem defined by

$$v_{|\mathcal{S}}(\sigma) := \min_{x \in \mathcal{S}} \bar{h}(x) + \frac{1}{N} \sum_{s=1}^N h_{\sigma_s}^{(s)}(x)$$

admits a closed-form expression. Therefore, when resorting to Corollary 7.18 in practice, one should first question the time it would take to adopt the approach of (7.44). If it is presumably too long then one should launch a global solver to compute $\hat{F}_{\hat{x},\mathcal{S}}^*$.

Remark 7.20. Let $\hat{\mathcal{S}} \supseteq \mathbb{B}_2(\hat{x}; \hat{\alpha})$ so that $\mathcal{S} = \mathcal{X} \cap \hat{\mathcal{S}} \supseteq \mathcal{X} \cap \mathbb{B}_2(\hat{x}; \hat{\alpha})$ in Corollary 7.18. One should, if possible, use refined $\hat{\mathcal{S}}$ -bounds (Definition 7.10) instead of the assumed $\mathcal{X}$ -bounds (Assumption (B)), when building the model of $\hat{F}(\cdot|\hat{x})$, i.e. (7.40), involved in the problem (7.41). The choice of $\hat{\mathcal{S}}$ significantly affects the practical solvability of $F_{\hat{x},\mathcal{S}}^*$. We illustrate this procedure in Section 7.4.3 with two numerical examples for which, fortunately, $\hat{\mathcal{S}}$ -bounds are available in closed-form.

7.2.3 Global (BIC) Model

We proceed to the second global reformulation of (SMC), this latter being of utmost practical interest. Every discrete minimum among $\bar{n} \in \mathbb{N}$ elements from a vector $\mathbf{h} = (h_1, \dots, h_{\bar{n}})$ satisfies the relationship

$$\min_{l \in [\bar{n}]} h_l = \min_{q \in \Delta^{\bar{n}}} q^T \mathbf{h}. \quad (7.45)$$

7 | Relaxed Alternating Minimization for (SMC)

Based on (7.45), looking through another lens, (SMC) is equivalent to

$$F^* = \min_{x \in \mathcal{X}, Q \in \mathcal{Q}} \left\{ \bar{F}(x, Q) := \bar{h}(x) + \frac{1}{N} \sum_{s=1}^N \sum_{l=1}^{n_s} q_l^{(s)} h_l^{(s)}(x) \right\} \quad (\text{BIC-SMC})$$

with $\mathcal{Q} = \Delta^{n_1} \times \dots \times \Delta^{n_N}$ and $Q = (q^{(1)}, \dots, q^{(N)})$. Fixing the values of *component weights* $\{q^{(s)}\}_{s=1}^N$ in their respective simplices, the problem (7.46) becomes convex

$$\bar{F}^*(Q) := \min_{\tilde{x} \in \mathcal{X}} \bar{F}(\tilde{x}, Q). \quad (7.46)$$

Considering the notations above, we can write

$$F^* = \min_{Q \in \mathcal{Q}} \bar{F}^*(Q). \quad (7.47)$$

It is worth noticing that if $\bar{F}(\cdot, Q)$ is strongly-convex then $\bar{F}^*(Q)$ is continuously differentiable in Q and one can tackle (7.47) with standard Frank-Wolfe algorithms [LJ16a, Pok24], highlighting already that one can expect some convergence. Together, equations (7.46) and (7.47) undoubtedly suggest a two-step alternating minimization (AM) [GS00] map $\mathcal{M} : \mathcal{X} \times \mathcal{Q} \rightarrow \mathcal{X} \times \mathcal{Q}$, $(x, Q) \rightarrow \mathcal{M}(x, Q) = (x_+, Q_+)$

$$\begin{cases} x_+ & \leftarrow \arg \min_{\tilde{x} \in \mathcal{X}} \bar{F}(\tilde{x}, Q) \\ Q_+ & \leftarrow \arg \min_{\tilde{Q} \in \mathcal{Q}} \bar{F}(x_+, \tilde{Q}) \end{cases} \quad (\text{AM})$$

for which one looks after *fixed-value pairs*, i.e. $x \in \mathcal{X}$ and $Q \in \mathcal{Q}$ such that $\bar{F}(x, Q) = \bar{F}(\mathcal{M}(x, Q))$. Let x^* be a minimizer of (SMC) and let $Q^* \in \arg \min_{\tilde{Q} \in \mathcal{Q}} \bar{F}(x^*, \tilde{Q})$, then (x^*, Q^*) must be a *fixed-value pair* of $\mathcal{M}$, easily deduced from the fact that $\mathcal{M}$ decreases $\bar{F}$ monotonically. Actually, we will see in a while that being a *fixed-value pair* (x, Q) is sufficient to prove that x is *critical* for (SMC). However, as we expose in Proposition 7.22, *criticality* requires even less stringent conditions in general. To that end, we need to define our notion of *gain* (when optimizing the *weights*).

Definition 7.21 (Gain). Let $Q \in \mathcal{Q}$ and $x_+ \in \arg \min_{\tilde{x} \in \mathcal{X}} \bar{F}(\tilde{x}, Q)$. The *gain* at x_+ is the positive quantity defined as

$$\mathcal{G}_{(x_+, Q)}^* = \bar{F}(x_+, Q) - \min_{\tilde{Q} \in \mathcal{Q}} \bar{F}(x_+, \tilde{Q}) \geq 0. \quad (7.48)$$

Proposition 7.22. If $\mathcal{G}_{(x_+, Q)}^* = 0$ in (7.48) then x_+ is a critical point for (SMC).

Proof. In the absence of gain, it comes $Q \in \arg \min_{\tilde{Q} \in Q} \bar{F}(x_+, \tilde{Q})$.

We first notice that for every $s \in [N]$ and $l \in [n_s]$, $q_l^{(s)} > 0 \Rightarrow l \in \mathcal{A}^{(s)}(x_+)$.

Otherwise, there would exist $\bar{s} \in [N]$, $\bar{l} \in [n_{\bar{s}}]$ such that $q_{\bar{l}}^{(\bar{s})} > 0$ with $\bar{l}$ not in $\mathcal{A}^{(\bar{s})}(\hat{x})$. One could then spread the weight of $q_{\bar{l}}^{(\bar{s})}$ on indices of $\mathcal{A}^{(\bar{s})}(\hat{x})$ and achieve a strictly better objective, contradicting the optimality of Q . Moreover, first-order necessary optimality conditions for the convex subproblem defining x_+ imply the existence of $\hat{v} \in \mathcal{N}_{\mathcal{X}}(x_+)$, $\bar{g} \in \partial \bar{h}(x_+)$, $g_l^{(s)}(x_+) \in \partial h_l^{(s)}(x_+)$ for every $s \in [N]$, $l \in [n_s]$ such that

$$\mathbf{0}_d = \hat{v} + \bar{g} + \frac{1}{N} \sum_{s=1}^N \sum_{l \in \mathcal{A}^{(s)}(x_+)} q_l^{(s)} g_l^{(s)} = \hat{v} + g_1 - g_2$$

where $g_1 \in \partial f_1(x_+)$ and $g_2 \in \partial f_2(x_+)$ with

$$g_1 = \bar{g} + \frac{1}{N} \sum_{s=1}^N \sum_{l \in [n_s]} g_l^{(s)}, \quad g_2 = \frac{1}{N} \sum_{s=1}^N \sum_{\bar{l} \in \mathcal{A}^{(s)}(x_+)} q_{\bar{l}}^{(s)} \sum_{l \in [n_s] \setminus \{\bar{l}\}} g_l^{(s)}$$

□

Remark 7.23. We crafted a simple toy example ($N = 1$, $n_1 = 3$, $d = 1$) showing the converse is not true. Consider the function $x \rightarrow F(x) = |x| + \min\{x - 1/8, x^2, 2x - 1/16\}$, weights $Q = (0, 1, 0)$ (e.g. encoding *active components* at $x = 1/2$) and $\mathcal{X} = [-2, 2]$. One easily obtains that $x_+ = 0 = \arg \min_{\tilde{x} \in [-2, 2]} |\tilde{x}| + \tilde{x}^2$ is *critical* in this context. According to the (DC) decomposition (7.12), $\partial f_1(x_+) = [2, 4]$ and $\partial f_2(x_+) = \{2\}$.

Yet, the *gain* is strictly positive with respect to $\bar{F}(x_+, Q) = 0$ in the sense

$$F(x_+) = \min_{\tilde{Q} \in \Delta^3} \bar{F}(x_+, \tilde{Q}) = -1/8 < 0 \Rightarrow \bar{F}(x_+, Q) - \bar{F}(x_+, \tilde{Q}) = 1/8 > 0$$

achieved for $\tilde{Q} = (1, 0, 0)$. Note x_+ is not globally optimal as $F^* = -33/16$.

To conclude about (BIC) models, we point out the following equation.

It explicitly shows that *fixed-value pairs* of $\mathcal{M}$ produce *critical points* x_+ ,

$$\bar{F}(x, Q) - \bar{F}(\mathcal{M}(x, Q)) = \underbrace{(\bar{F}(x, Q) - \bar{F}(x_+, Q))}_{\geq 0} + \underbrace{(\bar{F}(x_+, Q) - \bar{F}(x_+, Q_+))}_{=\mathcal{G}_{(x_+, Q)}^*}$$

7 | Relaxed Alternating Minimization for (SMC)

due to the implication $\bar{F}(x, Q) - \bar{F}(\mathcal{M}(x, Q)) = 0 \Rightarrow \mathcal{G}_{(x_+, Q)}^* = 0$ and Proposition 7.22. In Section 7.3 hereafter, we will ensure the sufficient decrease of our relaxed alternating minimization approaches (r-AM) with respect to the monitorable quantity $\mathcal{G}_{(x_+, Q)}^*$.

7.2.4 Problem specific knowledge encoding

We end this section with a word on *valid* and/or *symmetry breaking* constraints, also playing a determinant role in (MICP)'s solvability [Lib12, DL21]. In some situations, one is aware, *a priori*, of some regularity conditions or specific structure that holds at any optimal solution of (SMC). General *valid* constraints implement cuts in the original search space whereas *symmetry breaking* constraints essentially induce an ordering of the solutions and prevent the possibility of having multiple optimal solutions that are permutations of each other [SSUT24]. We illustrate now both types of constraints hereafter.

Example 7.24 (L_2 -clustering). Let $\{\beta^{(s)}\}_{s \in [N]}$ be N points in $\mathbb{R}^p$. A clustering task with $B \in \mathbb{N}$ centroids might ask to minimize the average distance to the nearest centroid among $\{x_l\}_{l \in [B]}$,

$$x = (x_1, \dots, x_B) \rightarrow F(x) = \frac{1}{N} \sum_{s=1}^N \min_{l \in [B]} \|x_l - \beta^{(s)}\|_2.$$

- For any $l \in [B]$, it can be shown that an optimally located centroid x_l^* is contained in the convex hull of $\{\beta^{(s)}\}_{s=1}^N$. The following constraints are *valid cuts*

$$x_l^* \in \mathbf{conv}(\{\beta^{(1)}, \dots, \beta^{(N)}\}) \quad \forall l \in [B]. \quad (7.49)$$

Also, without loss of generality, an optimal solution can be assumed to satisfy the following conditions:

$$x_{l_1}^* \neq x_{l_2}^* \quad \forall (l_1, l_2) \in [B]^2, l_1 \neq l_2. \quad (7.50)$$

Unfortunately, these constraint are nonconvex. If such hypothesis applies, a more restrictive constraint that entails (7.50) would be, for a chosen $\delta \geq 0$,

$$\langle \mathbf{1}_d, x_{l_1} - x_{l_2} \rangle \geq \delta \quad \forall (l_1, l_2) \in [B]^2, l_1 < l_2. \quad (7.51)$$

These *symmetry breaking* constraints rank subvectors $\{x_l\}_{l \in [B]}$ according to their mean and impose a separation of δ/d between two successive means.

- For any $l \in [B]$, there should be at least one data point $\beta^{(s)}$ such that x_l^* is the closest among $(x_1^*, \dots, x_B^*)$ from $\beta^{(s)}$. We write the *valid* constraints (7.52) and (7.53) for our (MICP) and (BIC) reformulations of (SMC).

$$\sum_{s \in [N]} t_l^{(s)} \geq 1 \quad \forall l \in [B], \quad (7.52)$$

$$\sum_{s \in [N]} q_l^{(s)} \geq 1 \quad \forall l \in [B]. \quad (7.53)$$

Remark 7.25. If *a priori* information is encoded through constraints such as (7.49), we implicitly assume that feasible sets $\mathcal{X}$, $\mathcal{T} = \times_{s=1}^N \{0, 1\}^{n_s}$ and $\mathcal{Q} = \times_{s=1}^N \Delta^{n_s}$ for variable x , $T = (t^{(1)}, \dots, t^{(N)})$ and $Q = (q^{(1)}, \dots, q^{(N)})$ incorporate these constraints.

7.3 Relaxed Alternating Minimization (r-AM)

Although alternating minimization applied on (BIC-SMC) is intuitive and works relatively well in practice (see e.g. [JT15, CHW22]), we emphasize a major drawback in Section 7.3.1. Fortunately, this drawback can be somehow mitigated and this is what we propose to explore in Section 7.3.3.

7.3.1 Rationale

By default, the update of the weights in AM, i.e. the minimization of $\bar{F}(x_+, \tilde{Q})$ with respect to $\tilde{Q}$ at fixed $x_+ \in \mathcal{X}$, assigns systematically for every $s \in [N]$ and $l \in [n_s]$,

$$(q_+^{(s)})_l = \begin{cases} 1 & \text{when } l \text{ is the smallest element of } \mathcal{A}^{(s)}(x_+) \\ 0 & \text{otherwise.} \end{cases} \quad (7.54)$$

In other words, the information about other *component's* values, i.e. $h_l^{(s)}(x_+)$ for every $s \in [N]$ and $l \neq \min \mathcal{A}^{(s)}(x_+)$, is totally disregarded. Lemma 7.17 underlines that directly next to x_+ , the representation of objective F only relies on indices belonging to active sets $\mathcal{A}^{(s)}(x_+)$. However, *components* close to active at x_+ play a key role when it comes to represent the landscape of each loss $h^{(s)}$ in a neighbouring valley, i.e. regions $\mathcal{R}(\sigma)$ (recall Eq. (7.10)) for which there exists a $\tilde{s} \in [N]$ such that $\sigma_{\tilde{s}} \notin \mathcal{A}^{(\tilde{s})}(x_+)$.

7 | Relaxed Alternating Minimization for (SMC)

Explained differently, AM proceeds as follows. Given $x_+ \in \mathcal{X}$, it sets $\sigma = (\sigma_1, \dots, \sigma_N)$ where, for every $s \in [N]$, $\sigma_s = l$ if and only if $(q_+^{(s)})_l = 1$ according to (7.54), achieving

$$F_\sigma(x_+) = \bar{h}(x_+) + \frac{1}{N} \sum_{s=1}^N h_{\sigma_s}^{(s)}(x_+) = F(x_+).$$

Then, x_{++} is chosen as a global minimizer of F_σ . If it occurs that $F_\sigma(x_{++}) = F(x_{++})$ then AM might already be stuck. Indeed, this means that σ_s belongs to $\mathcal{A}^{(s)}(x_{++})$ for every $s \in [N]$. Furthermore if σ_s is the smallest element⁶ of $\mathcal{A}^{(s)}(x_{++})$ for every $s \in [N]$ then the procedure detailed above will loop forever. This suggests that AM is biased in the sense that x -updates only take into account the very local geometry of F and, presumably, tends to converge fast but towards *critical points* not far away from the initial guess. Based on this observation, one may attempt to put strictly positive weights not only on indices from $\mathcal{A}^{(s)}(x_+)$ but also on other promising indices linked to close to active *components*. As mentioned earlier, they better describe the surroundings of x_+ and, possibly, allow to escape from the current convex valley wherein AM will perhaps stay until convergence. We initiated this work with the belief that coupled, with a multi-start strategy, this approach would promote more exploration in the search space and lead to better final objective values.

To end this subsection, we show a 1D toy example clearly displaying the valley escape phenomenon we strive for when relaxing plain alternating minimization.

Example 7.26 (Valley escape phenomenon). On Figure 7.3, we depicted a (SMC) objective defined for every $x \in \mathcal{X} = [-5, 5]$ by $F(x) = \frac{1}{2} \min \{x^2 - 3x - 2, x^2 + 2, x^2 + x - 2, x^2 + 4x + 2\} + \frac{1}{2} \min \{\frac{1}{2}x^2 + 2x - 2, x^2 + 4x + 2, x^2 - 1\}$. Its (unique) global minimizer is located at $x^* = -4$. We sampled 10^2 equidistant starting points along $[-5, 5]$ and applied plain alternating minimization AM subject to the election rule (7.54). We also tried an alternative version wherein for every other iteration, instead of (7.54), we set up the weights as

$$q_+^{(s)} = \text{softmin} \left(\kappa \cdot (h_1^{(s)}(x_+), \dots, h_{n_s}^{(s)}(x_+)) \right) \quad \forall s \in [N]. \quad (7.55)$$

⁶Note that it was the smallest element of $\mathcal{A}^{(s)}(x_+)$ according to (7.54) and the fact that $\sigma_s = l \Leftrightarrow (q_+^{(s)})_l = 1$.

where the **softmin** transformation of $u = (u_1, \dots, u_{\bar{n}}) \in \mathbb{R}^{\bar{n}}$ reads

$$(v_1, \dots, v_{\bar{n}}) = \mathbf{softmin}(u) \quad \text{with} \quad v_l = e^{-u_l} / \sum_{l' \in [\bar{n}]} e^{-u_{l'}} \quad \forall l \in [\bar{n}]. \quad (7.56)$$

Here, $N = 2$ with $n_1 = 4$ and $n_2 = 3$. This alternative method with $\kappa = 1/4$ is dubbed as ALTER on Figure 7.3. Note that $\kappa \rightarrow \infty$ in (7.55) would have led to AM back again. Hence, using the **softmin** operator above relaxes exact optimization of the weights at fixed $x_+ \in \mathcal{X}$. One can observe that for every starting point, it allowed ALTER to converge towards x^* although the relaxation mechanism (7.55) was employed for only half of the iterations. It allowed jumping over a *local minimum* near $x \simeq 3/2$ while AM inevitably got trapped therein.

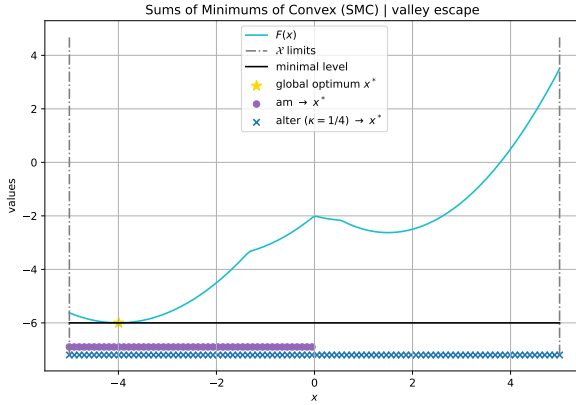


Fig. 7.3 A first glimpse of *relaxed Alternating Minimization* methods

7.3.2 x -updates

We emphasize that, compared to AM, the *relaxed alternating minimization* (r-AM) methods we will describe only differ in the way they handle Q -updates. Let $Q \in \mathcal{Q}$, the minimizer $x_{|Q}^*$ of $F_{|Q}^*$ (recall Eq. (7.46)) returned by our black-box (Assumption (A)) is chosen deterministically.

Remark 7.27. When the mapping $Q \rightarrow x_{|Q}^*$ is Lipschitz continuous, any algorithm (e.g. AM) alternating between x -updates and Q -updates enjoys the desirable feature that converging Q also imply converging x iterates.

Nevertheless, one cannot usually assess the Lipschitz continuity of such a mapping beforehand. However, it might happen in practice that two successive iterates x and x_+ define the exact same *active sets*, i.e. $\mathcal{A}^{(s)}(x) = \mathcal{A}^{(s)}(x_+)$ for every $s \in [N]$. When Q -updates do not allow for symmetry, e.g. in (7.54), we note that this would induce equal weights Q and Q_+ . Then, as highlighted above, two consecutive calls to the black-box with inputs Q and Q_+ where $Q = Q_+$ will provide $x_{|Q}^* = x_{|Q_+}^*$ so that a stopping criteria based on the distance between successive iterates will halt.

7.3.3 Q -updates

We explain now the threefold weight update mechanism in r -AM methods. Let $x_+ \in \mathcal{X}$, we will extensively use the shorthand

$$\mathbf{h}^{(s)}(x_+) := (h_1^{(s)}(x_+), \dots, h_{n_s}^{(s)}(x_+)) \in \mathbb{R}^{(n_s)} \quad \forall s \in [N].$$

Informally, Q -updates in r -AM are simply convex combinations between the weights associated with AM, i.e. (7.54), and sound **candidate** weights that take into account the value of *all* the *components*, e.g. (7.55), as motivated in Section 7.3.1. We get $Q_+ = (q_+^{(1)}, \dots, q_+^{(N)}) \in \mathcal{Q}$ thanks to the following election procedure applied for every $s \in [N]$:

$$\begin{cases} (q^*)^{(s)} & \leftarrow \arg \min_{\tilde{q} \in \Delta^{n_s}} \langle \tilde{q}, \mathbf{h}^{(s)}(x_+) \rangle & // \text{ AM weights} \\ \hat{q}_+^{(s)} & \leftarrow \text{candidate from } \Delta^{n_s} & // \text{ exploration} \\ q_+^{(s)} & = \varepsilon^{(s)} \cdot \hat{q}_+^{(s)} + (1 - \varepsilon^{(s)}) \cdot (q^*)^{(s)} & // \text{ convex combination} \end{cases} \quad (7.57)$$

where $\varepsilon^{(s)} \in [0, 1]$ depicts the *exploration ratio* of the update. If $\varepsilon^{(s)} = 1$ (full exploration) then $q_+^{(s)} = \hat{q}_+^{(s)}$ whereas if $\varepsilon^{(s)} = 0$ (greedy) then $q_+^{(s)} = (q^*)^{(s)}$ like AM. Let us illustrate (7.57) before moving on. On Figure 7.3, remembering its description from Section 7.3.1, ALTER can be seen as a r -AM method with $\varepsilon^{(1)} = \varepsilon^{(2)} = 0$ every odd iteration and $\varepsilon^{(1)} = \varepsilon^{(2)} = 1$ every pair iteration with $\hat{q}_+^{(1)}$ and $\hat{q}_+^{(2)}$ computed as (7.55).

We prove now Lemma 7.28 that provides a rule to set the parameters $\varepsilon = (\varepsilon^{(1)}, \dots, \varepsilon^{(N)})$ so that a sufficient decrease of $\bar{F}(x_+, \cdot)$ is guaranteed as long as the *gain* $\mathcal{G}_{(x_+, Q)}^* > 0$, i.e. the weight update in AM itself decreases $\bar{F}(x_+, \cdot)$.

Lemma 7.28 (Exploration Bound). Let $C \in [0, 1]$. Imposing, for every $s \in [N]$,

$$\varepsilon^{(s)} = \min \{1, C \cdot \langle q^{(s)} - (q^*)^{(s)}, \mathbf{h}^{(s)}(x_+) \rangle / \langle \hat{q}_+^{(s)} - (q^*)^{(s)}, \mathbf{h}^{(s)}(x_+) \rangle\} \quad (7.58)$$

in (7.57) ensures⁷ that $\bar{F}(x_+, Q) - \bar{F}(x_+, Q_+) \geq (1 - C) \cdot \mathcal{G}_{(x_+, Q)}^*$.

Proof. Let $Q_+(\varepsilon) = (q_+^{(1)}(\varepsilon^{(1)}), \dots, q_+^{(N)}(\varepsilon^{(N)}))$ with, for every $s \in [N]$,

$$q_+^{(s)}(\varepsilon^{(s)}) = \varepsilon^{(s)} \cdot \hat{q}_+^{(s)} + (1 - \varepsilon^{(s)}) \cdot (q^*)^{(s)}$$

as suggested by (7.57). We start by recalling that

$$\begin{aligned} \mathcal{G}_{(x_+, Q)}^* &= \frac{1}{N} \sum_{s=1}^N \langle q^{(s)} - (q^*)^{(s)}, \mathbf{h}^{(s)}(x_+) \rangle \\ \bar{F}(x_+, Q) - \bar{F}(x_+, Q_+) &= \frac{1}{N} \sum_{s=1}^N \langle q^{(s)} - q_+^{(s)}, \mathbf{h}^{(s)}(x_+) \rangle \end{aligned}$$

Based upon this decomposition, if for every $s \in [N]$, we show that

$$\langle q^{(s)} - q_+^{(s)}, \mathbf{h}^{(s)}(x_+) \rangle \geq (1 - C) \cdot \langle q^{(s)} - (q^*)^{(s)}, \mathbf{h}^{(s)}(x_+) \rangle \quad (7.59)$$

then the proof is completed. We need to consider two cases. First, let $\varepsilon^{(s)} =$

1. In such circumstances, $q_+^{(s)} = \hat{q}_+^{(s)}$ and according to (7.58),

$$\begin{aligned} \langle \hat{q}_+^{(s)} - (q^*)^{(s)}, \mathbf{h}^{(s)}(x_+) \rangle &= \langle q_+^{(s)} - (q^*)^{(s)}, \mathbf{h}^{(s)}(x_+) \rangle \\ &\leq C \cdot \langle q^{(s)} - (q^*)^{(s)}, \mathbf{h}^{(s)}(x_+) \rangle \end{aligned}$$

Multiplying the above inequality by -1 and adding $\langle q^{(s)} - (q^*)^{(s)}, \mathbf{h}^{(s)}(x_+) \rangle$ on both sides yields (7.59). Now, let $\varepsilon^{(s)} < 1$. We notice that $\varepsilon^{(s)} \geq 0$ by definition of $(q^*)^{(s)}$.

$$\begin{aligned} \langle q_+^{(s)}(\varepsilon^{(s)}), \mathbf{h}^{(s)}(x_+) \rangle &= \varepsilon^{(s)} \cdot \langle \hat{q}_+^{(s)}, \mathbf{h}^{(s)}(x_+) \rangle + (1 - \varepsilon^{(s)}) \cdot \langle (q^*)^{(s)}, \mathbf{h}^{(s)}(x_+) \rangle \\ &= \varepsilon^{(s)} \cdot \langle \hat{q}_+^{(s)} - (q^*)^{(s)}, \mathbf{h}^{(s)}(x_+) \rangle + \langle (q^*)^{(s)}, \mathbf{h}^{(s)}(x_+) \rangle \\ &= C \cdot \langle q^{(s)} - (q^*)^{(s)}, \mathbf{h}^{(s)}(x_+) \rangle + \langle (q^*)^{(s)}, \mathbf{h}^{(s)}(x_+) \rangle \end{aligned}$$

multiplying both sides by -1 and adding $\langle q^{(s)}, \mathbf{h}^{(s)}(x_+) \rangle$ yields (7.59). $\square$

⁷If $\hat{q}_+^{(s)} = (q^*)^{(s)}$ then $\varepsilon^{(s)} \leftarrow 1$.

Candidates Although the possibilities to choose **candidates** in (7.57) are limitless, we propose three different options that appear to work well in practice, compared to the alternating minimization AM baseline. Of course, some fine tuning has been attempted to set their hyper-parameter $\kappa \geq 0$ right. Nevertheless, to stay fair with respect to the parameter-free AM, throughout all our numerical experiments, we will adopt a fixed schedule $\{\kappa_k\}_{k \in \mathbb{N}}$ for the hyper-parameter value (see Table 7.1).

<i>method</i>	$\hat{q}_+^{(s)}$ expr.	κ - schedule
Barratt-Boyd (BB)	$u_l^{(s)} = \begin{cases} \text{proj}_{\Delta^{n_s}}(q^{(s)} + \kappa \cdot u^{(s)}) & \text{if } l = \min \mathcal{A}^{(s)}(x_+) \\ 1 & \text{otherwise} \end{cases}$	$\kappa_k = 10^{-1}$
(Norm.) Softmin (SM)	$\text{softmin}\left(\kappa \cdot \frac{\mathbf{h}^{(s)}(x_+) + u^{(s)}}{\max\{10^{-4}, \langle \mathbf{1}_{n_s}, \mathbf{h}^{(s)}(x_+) \rangle \}}\right)$ $u^{(s)} \sim \text{Uni}([-5 \cdot 10^{-7}, 5 \cdot 10^{-7}]^{n_s})$	$\kappa_k = \left(\frac{3}{2}\right)^{k^{\frac{3}{4}}}$
Max-Min (MM)	$\text{proj}_{\Delta^{n_s}}\left(\kappa \cdot \frac{\mathbf{1}_{n_s} \cdot \max_{l \in [n_s]} h_l^{(s)}(x_+) - \mathbf{h}^{(s)}(x_+)}{\max_{l \in [n_s]} h_l^{(s)}(x_+) - \min_{l \in [n_s]} h_l^{(s)}(x_+)}\right)$	$\kappa_k = k^{\frac{2}{3}}$

Table 7.1 Candidates for *exploration* in Q -updates.

Remark 7.29. The Barratt-Boyd (BB) candidate above stands as our extension of the inexact Q -update introduced in [BAB20], i.e. equation (7.18) when $n_s = 2$ for every $s \in [N]$. Unlike (SM) and (MM), (BB) candidate might fail to preserve the order of the values of the *components*, i.e. the following property for every $s \in [N]$:

$$h_{l_2}^{(s)}(x_+) \leq h_{l_1}^{(s)}(x_+) \Rightarrow (\hat{q}_+^{(s)})_{l_2} \geq (\hat{q}_+^{(s)})_{l_1} \quad \forall (l_1, l_2) \in [n_s]^2. \quad (7.60)$$

This is due to the fact that (BB) is not memoryless, its update involves the last Q used. It might happen, especially in early iterations when iterates are still far from convergence, that there exists $s \in [N]$ such that $(\hat{q}_+^{(s)})_l = q_l^{(s)} + \kappa < (\hat{q}_+^{(s)})_{\tilde{l}} = q_{\tilde{l}}^{(s)} - \kappa$ although l would be the smallest element of $\mathcal{A}^{(s)}(x_+)$. Indeed, it would occur if the weight previously put on $\tilde{l}$, i.e. $q_{\tilde{l}}^{(s)}$, was significant enough, i.e. $q_{\tilde{l}}^{(s)} > q_l^{(s)} + 2\kappa$.

Max-Min (MM) maps $\mathbf{h}^{(s)}(x_+)$ to $v^{(s)} \in [0, 1]^{n_s}$ where $v_l^{(s)} = 1$ (respectively 0) indicates that $h_l^{(s)}(x_+)$ achieves the minimal (respectively maximal) value of $\mathbf{h}^{(s)}(x_+)$. Then, the resulting $\kappa \cdot v^{(s)}$ is projected back on the simplex Δ^{n_s} .

Normalized Softmin (SM) is (7.55) but the input argument is divided by the (absolute of the) sum of the values of the *components*, i.e. $|\langle \mathbf{1}_{n_s}, \mathbf{h}^{(s)}(x_+) \rangle|$, if not too small. We underline that the small stochastic perturbations $\{u^{(s)}\}_{s \in [N]}$ aim at breaking close ties among the values of the *components*.

This promotes candidates fulfilling

$$\min_{q \in \{0,1\}^{n_s}} \|\hat{q}_+^{(s)} - q\| \rightarrow 0 \quad \text{as } \kappa \rightarrow \infty. \quad (7.61)$$

7.3.4 Convergence

We start by providing the pseudocode of r-AM methods in Algorithm 5.

Algorithm 5 r-AM (Q_{init})

Require: $\{C_k\}_{k \in \mathbb{N}}, \{\kappa_k\}_{k \in \mathbb{N}}$, tolerance $\delta > 0$, $Q_1 = Q_{\text{init}} \in \mathcal{Q}$, $v = \infty$, $K_{\text{max}} \in \mathbb{N}$

```

for  $k = 1, \dots, K_{\text{max}}$  do
     $x_k \leftarrow \arg \min_{\tilde{x} \in \mathcal{X}} \bar{F}(\tilde{x}, Q_k)$  // x-update
    for  $s = 1, \dots, N$  do // Q-update
         $(q^*)^{(s)} \leftarrow \arg \min_{\tilde{q} \in \Delta^{n_s}} \langle \tilde{q}, \mathbf{h}^{(s)}(x_+) \rangle$ 
         $\hat{q}_+^{(s)}(\kappa_k) \leftarrow \text{candidate from } \Delta^{n_s}$  // cfr. Table 7.1
         $q_+^{(s)} = \varepsilon^{(s)} \cdot \hat{q}_+^{(s)} + (1 - \varepsilon^{(s)}) \cdot (q^*)^{(s)}$  //  $\varepsilon^{(s)}(C_k, Q_k, \hat{Q}_+, Q^*)$ 
    end for
    if  $v - F(x_k) < \delta$  then return  $x$  such that  $F(x) = \min_{\tilde{k}=1, \dots, k} F(x_{\tilde{k}})$ 
    end if
     $(Q_{k+1}, v) \leftarrow (Q_+, \bar{F}(x_k, Q_k))$ 
end for
return  $x$  such that  $F(x) = \min_{\tilde{k}=1, \dots, k} F(x_{\tilde{k}})$ 

```

By choosing $(K_{\text{max}}, \delta) = (\infty, 0)$ in Algorithm 5, one would like to (asymptotically) converge towards *critical points*. Under such setting, using the values for $\varepsilon = (\varepsilon^{(1)}, \dots, \varepsilon^{(N)})$ prescribed in Lemma 7.28, we state in Theorem 7.30 sufficient conditions for r-AM methods, independently from the choice of **candidate**, to converge in objective value. We also give a characterization of cluster points.

7 | Relaxed Alternating Minimization for (SMC)

Theorem 7.30 (r-AM - convergence). Let $Q_{\text{init}} \in \mathcal{Q}$, $\delta = 0$ and $K_{\text{max}} = \infty$ within Algorithm 5. Let the values $\varepsilon^{(s)}$ follow the rule (7.58) for every $s \in [N]$ with $q^{(s)} = q_k^{(s)}$ and parameter C chosen to be C_k for every iteration $k \in \mathbb{N}$, let $\bar{C} := \sup_{k \in \mathbb{N}} C_k < 1$ and let $\hat{F} := \bar{F}(x_1, Q_1) < \infty$. The sequence $\{(x_k, Q_k)\}_{k \in \mathbb{N}}$ produced by r-AM satisfies

- (i). $\{\bar{F}(x_k, Q_k)\}_{k \in \mathbb{N}}$ is non-increasing.
- (ii). Every cluster point of $\{x_k\}_{k \in \mathbb{N}}$ is a critical point and for every $K \geq 2$

$$\min_{k=1, \dots, K-1} \mathcal{G}_{(x_k, Q_k)}^* \leq \frac{\hat{F} - F^*}{\sum_{k=1}^{K-1} (1 - C_k)} \leq \frac{\hat{F} - F^*}{(1 - \bar{C}) \cdot (K - 1)} \quad (7.62)$$

Proof. (i). Let $k \in \mathbb{N}$ be fixed. We show that $\bar{F}(x_k, Q_k) - \bar{F}(x_{k+1}, Q_{k+1}) \geq 0$.

$$\begin{aligned} \bar{F}(x_k, Q_k) - \bar{F}(x_{k+1}, Q_{k+1}) &= \underbrace{\bar{F}(x_k, Q_k) - \bar{F}(x_k, Q_{k+1})}_{\geq (1 - C_k) \cdot \mathcal{G}_{(x_k, Q_k)}^* \text{ (Lemma 7.28)}} + \underbrace{\bar{F}(x_k, Q_{k+1}) - \bar{F}(x_{k+1}, Q_{k+1})}_{\max_{\tilde{x} \in \mathcal{X}} \bar{F}(x_k, Q_{k+1}) - \bar{F}(\tilde{x}, Q_{k+1})} \end{aligned}$$

Since $x_k \in \mathcal{X}$ by construction, the two terms above are both nonnegative.

(ii). For any $(x, Q) \in \mathcal{X} \times \mathcal{Q}$, we have $\bar{F}(x, Q) \geq F(x) \geq F^* > -\infty$ hence one deduces that $\bar{F}$ is bounded from below. Summing the above equation for $k = 1$ to $K - 1$ yields

$$\begin{aligned} \bar{F}(x_1, Q_1) - \bar{F}(x_K, Q_K) &= \sum_{k=1}^{K-1} \bar{F}(x_k, Q_k) - \bar{F}(x_{k+1}, Q_{k+1}) \\ &\geq \sum_{k=1}^{K-1} (1 - C_k) \cdot \mathcal{G}_{(x_k, Q_k)}^* \end{aligned}$$

from which (7.62) readily emerges as $\bar{C} < 1$. Now, let us denote by $(x^\perp, Q^\perp)$ an arbitrary cluster point of the sequence $\{(x_k, Q_k)\}_{k \in \mathbb{N}}$. We would like to show that $x^\perp$ must be *critical*. Let us assume by absurdum that $\mathcal{G}_{(x^\perp, Q^\perp)}^* \geq 2\iota > 0$. By continuity of $\mathcal{G}_{(x, Q)}^*$ as a function with input arguments (x, Q) and the definition of cluster point in first-countable sets, there exists an infinite subset of indices $\mathcal{K} \subseteq \mathbb{N}$ generating the subsequence $\{(x_{\tilde{k}}, Q_{\tilde{k}})\}_{\tilde{k} \in \mathcal{K}}$ along which elements are close enough from $(x^\perp, Q^\perp)$ such that $\mathcal{G}_{(x_{\tilde{k}}, Q_{\tilde{k}})}^* \geq \iota$. Hence, one would conclude that $\lim_{\tilde{k} \rightarrow \infty, \tilde{k} \in \mathcal{K}} \bar{F}(x_{\tilde{k}}, Q_{\tilde{k}}) = -\infty$ (by subtracting at least $(1 - \bar{C}) \cdot \iota > 0$ infinitely many times from $\hat{F} = \bar{F}(x_1, Q_1)$) which is impossible since $\bar{F}$ is bounded from below. Therefore, it holds that $\mathcal{G}_{(x^\perp, Q^\perp)}^* = 0$, showing by Proposition 7.22 that $x^\perp$ must be *critical*. $\square$

Stopping criterion Let $\delta > 0$ for Algorithm 5. The main for-loop exits if quantity $\bar{F}(x_k, Q_k) - F(x_{k+1})$ falls below the threshold δ , which in turn would imply that $\bar{F}(x_k, Q_k) - \bar{F}(x_{k+1}, Q_{k+1}) \leq \delta$. In such circumstances, the guarantees at x_k , in terms of *gains* (cfr. Proposition 7.22), become

$$\mathcal{G}_{(x_k, Q_k)}^* \leq \delta / (1 - C_k).$$

7.4 Numerical Experiments

In this last section, we benchmark different local approaches on *Piecewise-Linear Regression* (PLR), exactly as detailed in Example 7.4, as well as on *Restricted Facility Location* (RFL) problems somehow related to Example 7.24. More specifically, we try out alternating minimization AM and make it compete against two of our r-AM methods, i.e. Algorithm 5 with **candidates** chosen as (Norm.) Softmin (SM) and Max-Min (MM) (see Table 7.1), and against our generalization (BB) of Barratt & Boyd’s work [BAB20]. Since (SMC) belongs to the less structured (DC) programming world, we also apply the standard (DC) algorithm, i.e. DCA (see e.g. [LP18]), for comparison. Except for DCA, every approach can be seen as a r-AM method with its own specifications in terms of ε policies and $\{C_k\}_{k \in \mathbb{N}}$ schedules.

<i>method</i>	ε rule	C-schedule
Barratt-Boyd (BB)	$\varepsilon = \mathbf{1}_N$	/
Alternating-Minimization (AM)	$\varepsilon = \mathbf{0}_N$	/
Max-Min (MM) (Norm.) Softmin (SM)	Lemma 7.28	$C_k = \frac{2}{\sqrt{k-1}+3}$

Table 7.2 Specifications of the methods.

For the (PLR) benchmark (respectively (RFL)), 50 (respectively 150) weights Q_{init} were sampled uniformly at random on $\mathcal{Q}$ and fed as starting Q iterates. We tracked the times taken by each method to terminate for $\delta = 10^{-8}$ and $K_{\text{max}} = 400$, and recorded the best objective value obtained during each run. Interestingly enough, it turns out that usual DCA iterations for (PLR) are the same as AM. However, in what concerns (RFL), the two methods are no longer equivalent.

7.4.1 Piecewise-Linear Regression

We chose the common datasets [abalone](#), [winequality](#) and [insurance](#) for this regression task. The raw data has first been worked out to transform the few categorical features into numerical ones and then to increase its dimensionality by incorporating second-order interactions between explanatory features. That is, let $\tilde{\beta} \in \mathbb{R}^{\tilde{p}}$ be an observation in the original feature space, then we computed $\bar{\beta} \in \mathbb{R}^p$ with $p = \tilde{p} \cdot (3 + \tilde{p})/2$ as

$$\tilde{\beta} \rightarrow \bar{\beta} = (\tilde{\beta}, \tilde{\beta}_1 \cdot \tilde{\beta}_{1:p}, \tilde{\beta}_2 \cdot \tilde{\beta}_{2:p}, \dots, \tilde{\beta}_p \cdot \tilde{\beta}_{p:p}) \in \mathbb{R}^{\tilde{p} \cdot (3 + \tilde{p})/2}$$

	insurance	winequality	abalone
p	27	77	44

In the end, we dispose of a dataset of tuples $(\bar{\beta}^{(s)}, \gamma^{(s)}) \in \mathbb{R}^{p+1}$ for $s = 1, \dots, N = 750$. To predict γ from $\bar{\beta}$, we fit the coefficients x of a piecewise-linear model in $\bar{\beta}$, i.e.

$$\Delta\ell(\bar{\beta}; x) = \max_{e_1 \in [B_1]} \{\langle \bar{\beta}, x_{e_1}^{(1)} \rangle\} - \max_{e_2 \in [B_2]} \{\langle \bar{\beta}, x_{e_2}^{(2)} \rangle\}$$

where $B_1 = 6$, $B_2 = 5$ and $x = (x_1^{(1)}, \dots, x_{B_1}^{(1)}, x_1^{(2)}, \dots, x_{B_2}^{(2)})$, by trying to minimize the averaged- L_1 prediction loss (7.63) on $\mathcal{X} = \mathbb{B}_\infty(\mathbf{0}_p; 10^2)^{B_1+B_2}$,

$$\min_{x \in \mathcal{X}} \frac{1}{N} \sum_{s=1}^N |\gamma^{(s)} - \Delta\ell(\bar{\beta}^{(s)}; x)|. \quad (7.63)$$

Results As expected (cfr. Section 7.3.1), one can clearly observe on Figure 7.6 that AM usually finished the fastest but, most of the time, with the second worst average objective value after (BB). The latter was quite slow and did not manage to explore as much as the other contenders. We partially explain this by the number of Q -updates it takes (BB) to move from one vertex of $\mathcal{Q}$ to another when n_s is big (here $n_s = B_1 \cdot B_2 = 30$ for every $s \in [N]$, see Example 7.4). On the other hand, this ability to try intermediate weights Q , i.e. not only vertices of $\mathcal{Q}$, also represents the exploration strength of the method compared to AM. Hence, a *trade-off* symbolized by κ should exist although our trials with $\kappa = 1/10^6$ for $\varsigma \in \{1/4, 1/2, 1, 2\}$ were not more successful. Based on Table 7.3, we can safely state that our methods (MM) and (SM) were the most competitive on the (PLR). One either chooses rapidity (MM) or better objective values (SM).

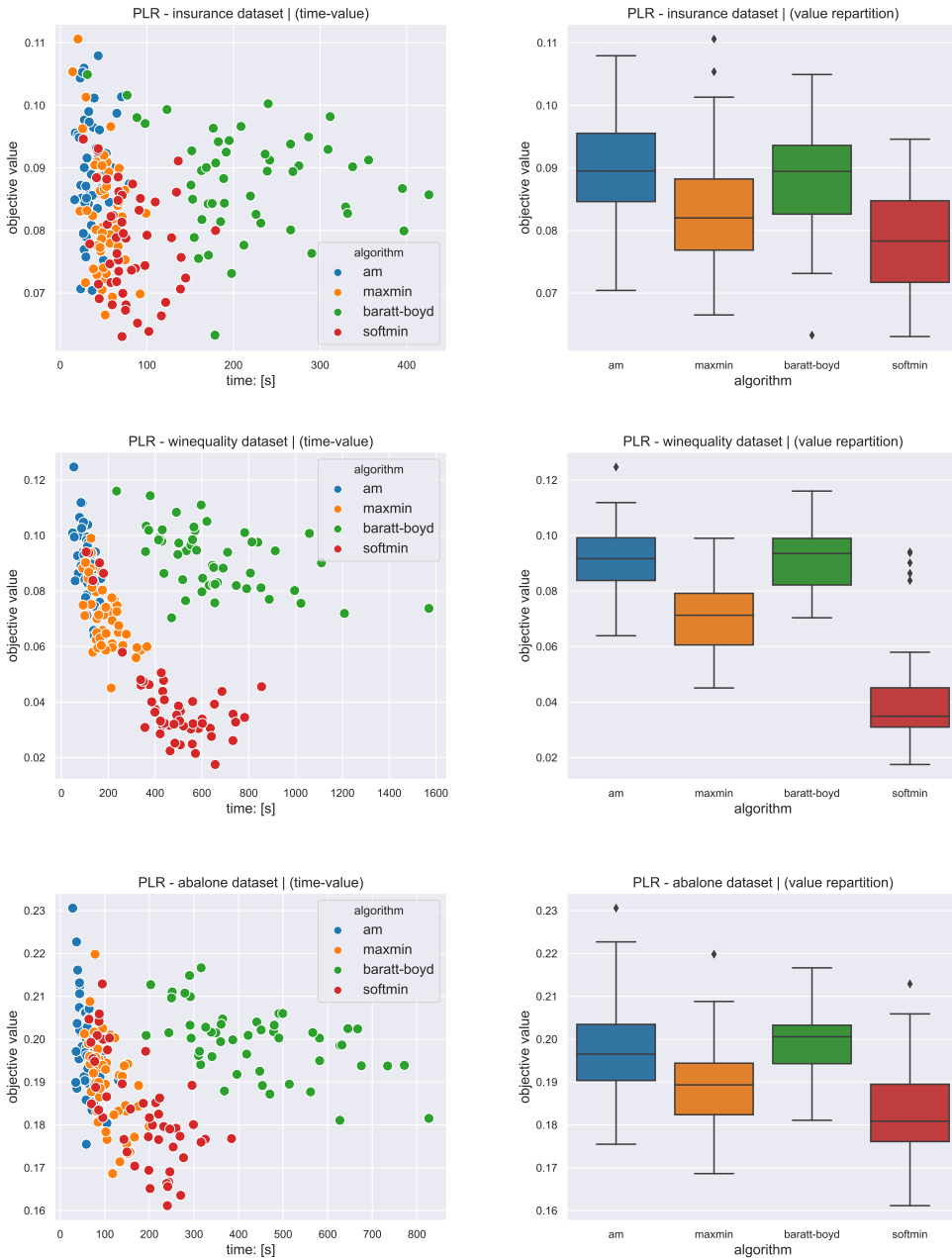


Fig. 7.6 (PFL): (time) and (value distribution)

7 | Relaxed Alternating Minimization for (SMC)

dataset	method	avg. time [s]	min. value	avg. value	med. value
insurance	(MM)	51.47	0.0664	0.0825	0.0820
	AM / DCA	37.26	0.0704	0.0895	0.0895
	(BB)	219.34	0.0632	0.0879	0.0894
	(SM)	80.77	0.0630	0.0779	0.0783
winequality	(MM)	186.85	0.0450	0.0712	0.0712
	AM / DCA	108.43	0.0638	0.0906	0.0916
	(BB)	659.53	0.0703	0.0913	0.0935
	(SM)	479.71	0.0175	0.0407	0.0348
abalone	(MM)	112.41	0.1686	0.1886	0.1893
	AM / DCA	63.28	0.1754	0.1976	0.1965
	(BB)	439.67	0.1810	0.1995	0.2005
	(SM)	186.03	0.1611	0.1830	0.1808

Table 7.3 Piecewise-Linear Regression: results

7.4.2 Restricted Facility Location

We picked up the datasets [austria](#), [belgium](#) and [costarica](#) for this facility location task. A dataset is composed of N cities, each of which identified by an index $s \in [N]$ with respective population $P^{(s)} \in \mathbb{N}$ and coordinates $\beta^{(s)} \in \mathbb{R}^2$ (latitude, longitude).

	austria	belgium	costarica
N	183	53	140

One plans to build $B = 10$ stores, namely $\{x_l\}_{l \in [B]}$, and one distribution center, $x_0 \in \mathbb{R}^2$, so that considering the distance induced by $\|\cdot\|_1$ on $\mathbb{R}^2$,

★ the population weighted averaged distance to the closest store is minimized, i.e. the *components* read

$$h_l^{(s)}(x) = \frac{P^{(s)}}{\sum_{\bar{s}=1}^N P^{(\bar{s})}} \|\beta^{(s)} - x_l\|_1 \quad \forall s \in [N], \quad \forall l \in [B].$$

◇ the stores are located within a $R_{\text{ref}} > 0$ distance from the center x_0

We set a cost by unit of excess distance $\Lambda > 0$ to be incurred if one chooses to relax the proximity constraint above. In the spirit of [HN09], the problem

is formulated as follows

$$\min_{x \in \mathcal{X}} \Lambda \cdot \max\{R - R_{\text{ref}}, 0\} + \left(\sum_{\bar{s}=1}^N P^{(\bar{s})} \right)^{-1} \sum_{s=1}^N \min_{l \in [B]} P^{(s)} \|\beta^{(s)} - x_l\|_1 \quad (7.64)$$

where the description of the basic feasible set is as follows,

$$\mathcal{X} = \left\{ (R, x_0, x_1, \dots, x_B) \in \mathbb{R} \times \mathbb{R}^2 \times (\mathbb{R}^2)^B \mid R > 0, \|x_0 - x_l\|_1 \leq R \ \forall l \in [B] \right\}.$$

Results Table 7.4 displays the results obtained when $\Lambda = 0$, i.e. when we do not penalize the placement of stores outside the R_{ref} distance window from the distribution center x_0 . This time, performances are more spread out. Depending on the dataset, a different winner arises, e.g. AM dominated for the case austria. Nevertheless, DCA clearly fell short suggesting it is worthwhile to use the structure behind (BIC-SMC) in r-AM methods. Looking closer to Table 7.4, we point out that for datasets belgium and costarica, our (SM) and (MM) were again the best in terms of average/median objective value returned. Within these smaller scale problems (compared to (PLR)), (BB) slightly outperforms AM. Initially devised as $n_s = \bar{n} = 2$ for every term $s \in [N]$ and beating AM in such case according to the authors of [BAB20], it is not totally surprising to observe such behaviour as $\bar{n}$ drops from 30 (PLR) to 10 here.

Interpreting Table 7.5 for which we picked $\Lambda = 10$, one draws relatively similar conclusions. The solutions returned by all methods were satisfying the implicit constraint $R \leq R_{\text{ref}}$, meaning that the penalty implemented by parameter Λ was prohibitive enough. The median and minimal values obtained by each method among the 150 trials were bigger for $\Lambda = 10$ than for $\Lambda = 0$, as anticipated. A single exception exists when it comes to the performances of AM on costarica's dataset. We noticed that constraints dramatically changed the solutions about austria and belgium datasets. In the case of austria, methods performed equally well and their respective performance variability diminished.

Overall, unlike for (PLR), one can observe that AM provided satisfactory results on (RFL) tasks taking into account its speed. The method with the most variability, (BB), provided the best one-shot solution in the majority of the cases.

dataset	method	avg. time [s]	min. value	avg. value	med. value
austria	(MM)	1.0837	0.1279	0.3223	0.2862
	AM	0.2821	0.1008	0.3078	0.3024
	DCA	2.7869	0.1807	0.3268	0.3241
	(BB)	1.3152	0.1049	0.3280	0.3148
	(SM)	1.9238	0.1755	0.4144	0.3275
belgium	(MM)	0.6201	0.0924	0.1992	0.1901
	AM	0.1787	0.0781	0.2257	0.2382
	DCA	3.3046	0.1131	0.2333	0.2438
	(BB)	0.8764	0.0795	0.2237	0.2367
	(SM)	1.0632	0.0950	0.1754	0.1678
costarica	(MM)	0.9845	0.1232	0.1624	0.1614
	AM	0.3744	0.1242	0.1805	0.1829
	DCA	6.6098	0.1474	0.2067	0.2113
	(BB)	2.1687	0.1197	0.1699	0.1747
	(SM)	2.1629	0.1259	0.1670	0.1677

Table 7.4 Restricted Facility Location: results for $\Lambda = 0$

dataset	method	avg. time [s]	min. value	avg. value	med. value
austria	(MM)	1.0598	0.5463	0.5943	0.6041
	AM	0.3171	0.5556	0.6028	0.6054
	DCA	4.6697	0.5671	0.6473	0.6094
	(BB)	1.3908	0.5548	0.5997	0.6058
	(SM)	1.8707	0.5540	0.6073	0.6079
belgium	(MM)	0.5360	0.1301	0.2188	0.2331
	AM	0.2018	0.1545	0.2294	0.2357
	DCA	3.1733	0.1659	0.2349	0.2439
	(BB)	0.9186	0.1197	0.2313	0.2367
	(SM)	0.9724	0.1304	0.1894	0.1856
costarica	(MM)	1.0066	0.1255	0.1680	0.1694
	AM	0.4917	0.1311	0.1724	0.1746
	DCA	7.7209	0.1499	0.2109	0.2137
	(BB)	2.9921	0.1240	0.1710	0.1726
	(SM)	1.8744	0.1267	0.1707	0.1703

Table 7.5 Restricted Facility Location: results for $\Lambda = 10$

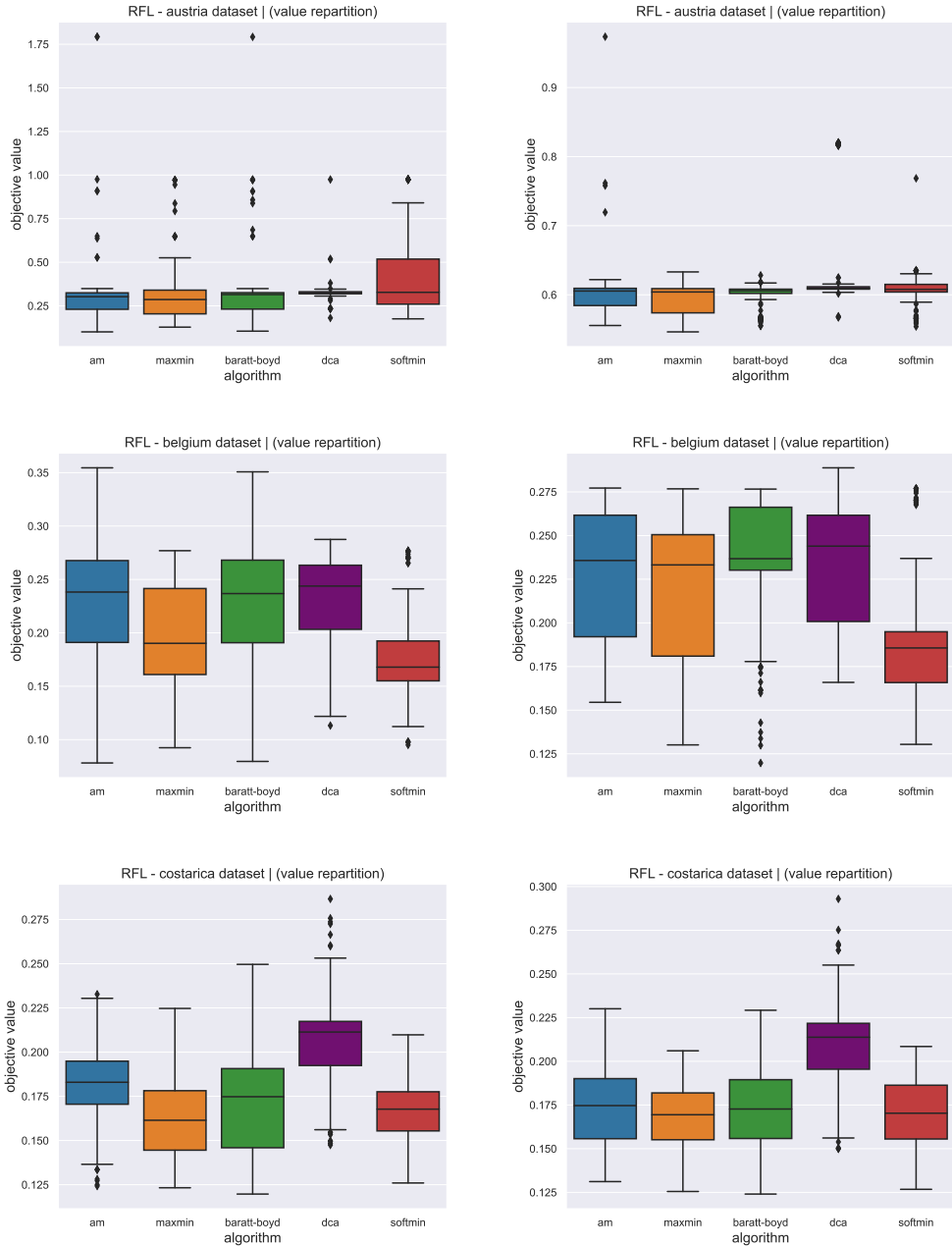


Fig. 7.9 (RFL): (value distribution) $\Lambda = 0$ (left) and $\Lambda = 10$ (right)

7.4.3 Local Optimality Certifications

Let $\hat{x} \in \mathcal{X}$ be the output iterate of a single run of any of the methods tested above. Fortunately enough for both (PLR) and (RFL), we can define a suitable neighbourhood $\hat{\mathcal{S}}$ around $\hat{x}$ so that the computation of $\hat{\mathcal{S}}$ -bounds (see Definition 7.10) becomes closed-form. Then, according to Remark 7.20, we can try to solve problem (7.41) with $\mathcal{S} = \mathcal{X} \cap \hat{\mathcal{S}}$ from Corollary 7.18 using these local $\hat{\mathcal{S}}$ -bounds as the *big-M* values. For numerical stability purposes, we used ρ -relative active sets at $\hat{x}$ (see Definition 7.6 with $\rho = 10^{-12}$) instead of mere active sets in formulation (7.40).

We fixed a time limit of 120 [s] and a sufficient decrease threshold of $\delta_{\text{glob}} = 5 \cdot 10^{-7}$. From there, two possible outcomes may happen:

- the solver outputs a point $x_{\text{new}} \in \mathcal{S}$ such that $F(x_{\text{new}}) \leq F(\hat{x}) - \delta_{\text{glob}}$
- the solver did not find such point

When (1) occurs, we restart the method with $q_{\text{init}}^{(s)} \in \arg \min_{\tilde{q} \in \Delta^{n_s}} \langle \tilde{q}, \mathbf{h}^{(s)}(x_{\text{new}}) \rangle$ for every $s \in [N]$, i.e. AM weights. We proceed all over again until the method reaches the next $\hat{x}$ approximate *critical* point. When (2) happens, we are done. Sometimes, it would mean that the solver found the optimal solution $x^*(\hat{x}, \mathcal{S})$ of (7.41) and based on the value $F(x^*(\hat{x}, \mathcal{S}))$, the *local optimality* of $\hat{x}$ might be certified (see Corollary 7.18).

$\hat{\mathcal{S}}$ -bounds for (PLR) Let $R > 0$ and let $(\|\cdot\|, \|\cdot\|_*)$ denote a norm and its dual defined on $\mathbb{R}^p$. Let $\hat{x} = (\hat{x}_1^{(1)}, \dots, \hat{x}_{B_1}^{(1)}, \hat{x}_1^{(2)}, \dots, \hat{x}_{B_2}^{(2)})$ under the notations of Section 7.4.1. If one takes

$$\hat{\mathcal{S}} = \left(\bigtimes_{e_1 \in [B_1]} \mathbb{B}_{\|\cdot\|}(\hat{x}_{e_1}^{(1)}; R) \right) \times \left(\bigtimes_{e_2 \in [B_2]} \mathbb{B}_{\|\cdot\|}(\hat{x}_{e_2}^{(2)}; R) \right),$$

then valid $\hat{\mathcal{S}}$ -bounds for (PLR) are given by

$$\begin{aligned} M_{l_+, l}^{(s)} &\leftarrow h_{l_+}^{(s)}(\hat{x}) - h_l^{(s)}(\hat{x}) \\ &\quad + 2 \cdot \left(\mathbb{1}_{\{e_1(l_+) \neq e_1(l)\}} + \mathbb{1}_{\{e_2(l_+) \neq e_2(l)\}} \right) \cdot (R \cdot \|\tilde{\beta}^{(s)}\|_*) \end{aligned} \quad (7.65)$$

for every $(l_+, l) \in [n_s]^2$ and $s \in [N]$. Here above, $e_1(l) = \lfloor l/B_2 \rfloor$ and $e_2(l) = 1 + (l-1) \bmod B_2$.

$\hat{\mathcal{S}}$ -bounds for (RFL) Let $R_\infty > 0$. Sticking to the notations used in Section 7.4.2, we can write $\hat{x} = (\hat{R}, \hat{x}_0, \hat{x}_1, \dots, \hat{x}_B)$. If one takes

$$\hat{\mathcal{S}} = [\hat{R} - R_\infty, \hat{R} + R_\infty] \times \left(\bigotimes_{l=0}^B \mathbb{B}_{\|\cdot\|_\infty}(\hat{x}_l; R_\infty) \right) \quad (7.66)$$

then valid $\hat{\mathcal{S}}$ -bounds for (PLR) are given by

$$\begin{aligned} M_{l_+, l}^{(s)} \leftarrow & N \cdot P^{(s)} / \bar{P} \cdot \left(\|\beta^{(s)} - \hat{x}_{l_+} + R_\infty \cdot \mathbf{sign}(\beta^{(s)} - \hat{x}_{l_+})\|_1 \right. \\ & \left. - \|\beta^{(s)} - \hat{x}_l - \mathbf{min}(R_\infty \cdot \mathbf{1}_2, \mathbf{abs}(\beta^{(s)} - \hat{x}_l)) \cdot \mathbf{sign}(\beta^{(s)} - \hat{x}_l)\|_1 \right) \end{aligned} \quad (7.67)$$

for every $(l_+, l) \in [n_s]^2$ and $s \in [N]$. Therein, $\bar{P} = \sum_{s=1}^N P^{(s)}$ and $\mathbf{sign}$ represents the vectorized version of $\text{sign} : \mathbb{R} \rightarrow \{-1, 1\}$ defined for every scalar u as $\text{sign}(u) = 1$ if $u \geq 0$ or $\text{sign}(u) = -1$ otherwise.

As a final contribution, paving the way for further investigations, we tried the procedure described at the beginning of this subsection using only (SM), our most promising local-search technique. We conducted 5 independent trials represented by 5 initial weights Q_{init} for every experiment. For each trial, we recorded the difference between the final best value obtained at the end of a minimization of (7.41) with the prescribed $\hat{\mathcal{S}}$ -bounds (either (7.65) for (PLR) or (7.67) for (RFL)) and the objective value obtained after the first halt of (SM) (i.e. after the initial r-AM run) within that trial.

This difference is a metric of performance and will be called *value enhancement*. Note that if the *value enhancement* is strictly positive, (SM) has been restarted at least once, i.e. it must have restarted after its first halt and the associated encounter of (7.41).

Using $R = 10^{-1}$, and $(\|\cdot\|, \|\cdot\|_*) = (\|\cdot\|_1, \|\cdot\|_\infty)$, we present in the table below the averaged performances for the (PLR) experiments.

dataset	avg. val. enhancement [%]	# local optimality certifications
insurance	4.79	5/5
winequality	0	0/5
abalone	2.79	2/5

7 | Relaxed Alternating Minimization for (SMC)

In what concerns (RFL) experiments, we fixed $R_\infty = 2 \cdot 10^{-1}$.

dataset	avg. val. enhancement [%]	# local optimality certifications
austria ($\Lambda = 0$)	8.76	5/5
austria ($\Lambda = 10$)	0	5/5
belgium ($\Lambda = 0$)	1.03	5/5
belgium ($\Lambda = 10$)	0.7	5/5
costarica ($\Lambda = 0$)	1.52	5/5
costarica ($\Lambda = 10$)	0	5/5

Results The ability to certify *local optimality* relies on the radii used for defining the neighbourhood $\hat{\mathcal{S}}$, e.g. R_∞ , as well as the capabilities of the (MICP) invoked. Also, it can happen that r-AM methods return points $\hat{x} \in \mathcal{X}$ that are not *local minima*. Arguably, our restart methodology did not provide substantial gains in terms of objective value (i.e. *value enhancement*) but, unexpectedly, managed to certify *local optimality* for many instances, especially in lower-dimensional settings such as in (RFL).

7.5 Conclusion

In this chapter, we studied a class of nonconvex nonsmooth optimization problems involving the sum of pointwise minima of convex functions. We derived (MICP) reformulations of the original problem as well as new strong *local optimality* conditions. To circumvent the computational difficulty associated with (MICP), we devised a second, simple, bi-convex reformulation of the problem. This latter allowed us to successfully develop a new framework for local-search methods that can be applied as good heuristics, beating most of the time baseline alternating minimization. On small neighbourhoods, our (MICP) formulation can be alleviated of many binary variables and proved to be useful to certify *local optimality* in certain scenarios. It remains unclear to us how and when to use these reduced scale (MICP) formulations. A possible future research direction is to investigate a smoothing of the objective as a tool to render substantial portions of its landscape benign. In addition, this would also make the problem amenable for second-order methods, not exploited in this context so far.

PART III
Chance-Constrained
Programs (CCP)

8

Presolve techniques for finite-support (CCP)

8.1 Introduction

Let ξ be a p -dimensional random vector with finite support of cardinality N , i.e. $\text{supp}(\xi) = \{\xi^{(1)}, \dots, \xi^{(N)}\}$. We are concerned in this final chapter with optimization problems of the form

$$F_{\tau}^* := \min_{x \in \mathcal{X}} F(x) \quad \text{s.t.} \quad \mathbb{P}_{\xi}[c(x, \xi) \leq 0] \geq 1 - \tau \quad (\text{CCP})$$

where F represents the objective function to be minimized, ξ parametrizes the probabilistic constraint, $\mathcal{X} \subseteq \mathbb{R}^d$ encodes deterministic constraints that must be satisfied (i.e. independently from ξ) and $\tau \in (0, 1)$ is a *relaxation tolerance* with respect to the (fully deterministic) robust counterpart of (CCP),

$$F_0^* := \min_{x \in \mathcal{X}} F(x) \quad \text{s.t.} \quad c(x, \xi^{(s)}) \leq 0 \quad \forall s \in \{1, \dots, N\}. \quad (\text{RO})$$

Given the finite support of ξ , one usually rewrites (CCP) as follows

$$F_{\tau}^* = \min_{x \in \mathcal{X}} F(x) \quad \text{s.t.} \quad \sum_{s=1}^N \pi^{(s)} \cdot \mathbb{1}(c(x, \xi^{(s)}) \leq 0) \geq 1 - \tau. \quad (8.1)$$

with $\pi^{(s)} := \mathbb{P}[\xi = \xi^{(s)}]$ for every $s \in [N] := \{1, \dots, N\}$.

Problems like (CCP) arise in various fields, e.g. in finance [SO23], supply-chain management [CK01], optimal vaccination strategies [TSN08] or even microgrid scheduling [LSX⁺17]. While robustness is arguably a desirable feature of solutions of optimization problems, formulations like (RO) tend to be overly conservative. Indeed, their optimal solutions are affected by every outcome of the random parameters ξ , even by the rarest scenarios that, by definition, do not occur often in practice.

Note that it can happen that (RO) is not even feasible, especially in the case of infinite supports. Setting the *relaxation tolerance* $\tau > 0$ (sometimes referred to as *risk tolerance* in the literature, see e.g. [ALSX17]) big enough can turn problem F_τ^* into a feasible one yet keeping a fair amount of robustness by considering most of constraints induced by the empirical scenarios. Solutions of (CCP) can dramatically improve the cost, i.e. $F_\tau^* \ll F_0^*$, but at the price of solving a nonconvex problem in general (see Section 8.2).

Big-M approach

In this context, one usually invokes the possibility to compute, for any realization of ξ , a big-M bound. For every $s \in [N]$, one has access to

$$M^{(s)} \geq \max_{x \in \mathcal{X}} c(x, \xi^{(s)}). \quad (8.2)$$

Based upon (8.1) and the above big-M constants (8.2), one can introduce N *switching* binary variables $z = (z^{(1)}, \dots, z^{(N)})$ to reformulate (CCP) as a mixed-integer program (MI-CCP) (see e.g. [ALSX17, KJ22, RFV23] and references therein)

$$\begin{aligned} F_\tau^* = \min_{x \in \mathcal{X}, z \in \{0,1\}^N} & F(x) \\ \text{s.t.} \quad & \sum_{s=1}^N \pi^{(s)} z_s \geq 1 - \tau \\ & c(x, \xi^{(s)}) \leq M^{(s)} \cdot (1 - z_s) \quad \forall s \in [N]. \end{aligned} \quad (\text{MI-CCP})$$

Note. Functions F and c do not need to be convex nor smooth. All we actually assume is that (MI-CCP) can be provided (as such) to some black-box global optimization tool that would solve it directly. Yet, this may be computationally very demanding. Notably, it is well-known that the large number of possibly quite conservative big-M constants leads to poor continuous relaxations [CRT90]. Yet, for many applications of interest, additional structure (e.g. [dO25] wherein $c(x, \xi)$ is linear in ξ) is available.

This leverages heuristics that allow to efficiently compute lower-bounds $\hat{F}_\tau^* \leq F_\tau^*$ and upper-bounds $\hat{F}_\tau^* \geq F_\tau^*$ as well as valid inequalities, substantially cutting down the effective search-space of (MI-CCP) during a pre-solve stage. We investigate this approach under the hypothesis that $c(x, \xi)$ is quasi-convex in ξ .

Goals As already claimed, (MI-CCP) tractability crucially depends on how large are the big- M values and how many scenarios (hence binary variables) are considered in the model. Unfortunately, even when the inequalities (8.2) are tight, the structure of c as well as the diameter of $\mathcal{X}$ can both lead to arbitrarily large bounds. Thereby, in order to speed-up the solving of (CCP) (which is our main goal), one can proceed as follows.

– **Logic encoding.** First, based on topological arguments in the ξ -space, we devise logical constraints involving variables z that induce a partition of the scenarios into a *safe set* $\oplus \subseteq [N]$, a *pruned set* $\ominus \subseteq [N]$ and a *selectable set* $[N] \setminus (\oplus \cup \ominus)$. Let F_τ^* be finite and let (x^*, z^*) be an optimal solution of (MI-CCP). If $s \in \oplus$ then either $z_s^* = 1$ or z_s^* can be set to 1 without changing the optimal value F_τ^* . If $s \in \ominus$, z_s^* must be equal to 0, i.e. the scenario $\xi^{(s)}$ can be ignored. Finally, inspired by [Rus02], we provide valid inequalities that couple the values z_s^* for $s \in [N] \setminus (\oplus \cup \ominus)$ so that at least one of them must be 0.

– **big- M tightening.** Second, akin to [RFV23], we tighten the big- M values based on $\oplus$ as well as, possibly, an upper-bound $\hat{F}_\tau^* \geq F_\tau^*$ (e.g. $\hat{F}_\tau^* = F_0^*$ if (RO) is *feasible*). For every $s \in [N] \setminus (\oplus \cup \ominus)$, we compute

$$M^{(s)}(\oplus, \hat{F}_\tau^*) \geq \max_{x \in \mathcal{X}, F(x) \leq \hat{F}_\tau^*} c(x, \xi^{(s)}) \quad \text{s.t.} \quad c(x, \xi^{(\tilde{s})}) \leq 0 \quad \forall \tilde{s} \in \oplus. \quad (8.3)$$

Note that if the problem at the right-hand side of (8.3) admits an optimal value less or equal than 0, the scenario s can be added to the *safe set* $\oplus$.

Regarding (8.2), $M^{(s)}$ is a shorthand for $M^{(s)}(\emptyset, +\infty)$ as defined in (8.3).

Notations When an optimization problem is *infeasible*, i.e. its *feasible set* is empty, we assign it the optimal value $+\infty$. Conversely, when it is *unbounded*, e.g. there exists a *recession direction* along which the objective is not lower-bounded, we set the optimal value to $-\infty$. Let $\bar{\xi} \in \mathbb{R}^p$ denote a realization of ξ , we introduce

$$\mathcal{R}(\bar{\xi}) := \{x \in \mathbb{R}^d \mid c(x, \bar{\xi}) \leq 0\} \quad (8.4)$$

8 | Presolve techniques for finite-support (CCP)

as the set of admissible decision vectors for the probabilistic constraint under ξ . For every subset $S \subseteq [N]$, we further define the subproblem

$$\nu(S) := \min_{x \in \mathcal{X}} F(x) \quad \text{s.t.} \quad c(x, \xi^{(s)}) \leq 0 \quad \forall s \in S. \quad (\text{S-subproblem})$$

Assumption 8.1. We lay down two useful assumptions holding throughout.

- For every $s \in [N]$,

$$\pi^{(s)} > 0 \quad \wedge \quad \nu(\{s\}) > -\infty. \quad (\text{A})$$

- For every $x \in \mathcal{X}$,

$$\text{the function } c(x, \cdot) \text{ is quasi-convex.} \quad (\text{B})$$

Remark 8.2. As pointed out in [Lue14], Assumption (A) is without loss of generality. Indeed, if either $\pi^{(s)} = 0$ or $\mathcal{X} \cap \mathcal{R}(\xi^{(s)}) = \emptyset$, $\xi^{(s)}$ should be removed from the dataset of considered scenarios and the problem becomes F_τ^* with $\tilde{N} = N - 1$ scenarios and $\tilde{\tau} = \tau - \pi^{(s)}$. Note that this assumption implies the *boundedness* of F_τ^* when $\tau < 1$ since there must be at least one $s \in [N]$ such that $z_s^* = 1$ at an optimal solution (x^*, z^*) . On the other hand, *feasibility* might be quite difficult to check *a priori*. Assumption (B) represents the building block of our presolve heuristics, i.e. we extensively use the convexity of the sublevel-sets of $c(x, \cdot)$ to add scenarios into $\oplus$ and $\ominus$, see **logic-encoding**.

We propose now a family of problems that satisfies our assumptions. From there, we detail two illustrative examples, i.e. Example 8.3 & 8.4.

Probabilistic Ball Projection

We are provided N different spatial locations, i.e. the rows of the data matrix $\mathbf{D} = (\xi^{(1)}, \dots, \xi^{(N)})^T \in \mathbb{R}^{N \times p}$, a reference point in space $\bar{x} \in \mathbb{R}^p$ and a pair of norms $(\|\cdot\|_o, \|\cdot\|_{\tilde{o}})$ with $o, \tilde{o} \in \{1, 2, \infty\}$. The goal is to find the closest point $x \in \mathbb{R}^p$ from $\bar{x}$, i.e. minimizing the distance $\|x - \bar{x}\|_o$ while ensuring that it belongs to $\mathbb{B}_{\tilde{o}}(\xi, R)$ with probability of at least $1 - \tau$. Assuming that $\mathbb{P}[\xi = \xi^{(s)}] = \pi^{(s)} > 0$ for every $s \in [N]$, the problem reads

$$F_\tau^* = \min_{x \in \mathcal{X}} \underbrace{\|x - \bar{x}\|_o}_{F(x)} \quad \text{s.t.} \quad \mathbb{P}_\xi[\underbrace{\|x - \xi^{(s)}\|_{\tilde{o}} - R}_{c(x, \xi)} \leq 0] \geq 1 - \tau. \quad (\text{PBP-}(p, o, \tilde{o}))$$

Here, $p = d$ and $\mathcal{X} = \mathbb{B}_\infty(\mathbf{0}_d, \bar{R})$ is as uniform box in $\mathbb{R}^d$ with $\bar{R} > 0$. For any $s \in [N]$, $\mathcal{X}(\{s\}) := \mathcal{X} \cap \mathcal{R}(\zeta^{(s)}) = \mathbb{B}_{\bar{\theta}}(\zeta^{(s)}, R)$ is convex and compact. Assumption (A) is satisfied with

$$\nu(\{s\}) = \|\mathbf{proj}_{\mathcal{X}(\{s\})}(\bar{x}) - \bar{x}\|_0 \geq 0 > -\infty$$

and $c(x, \zeta)$ is convex (thus quasi-convex) in ζ so that Assumption (B) is also verified. We present hereafter an instance of PBP-(2, 2, 1) (Example 8.3) with real application and one instance of PBP-(3, 2, ∞) (Example 8.4).

Example 8.3 (Probabilistic Facility Location). One must find where to install a new heliport facility $x \in \mathbb{R}^2$ so to minimize its flying L_2 -distance with respect to a reference hospital $\bar{x} \in \mathbb{R}^2$. The concerned population living in the surroundings is aggregated at N different locations, i.e. $(\zeta^{(1)}, \dots, \zeta^{(N)})$. A proportion $\pi^{(s)} > 0$ of the population is located at position $\zeta^{(s)}$ for every $s \in [N]$. The design constraint requires that an emergency happening at random among the population should be located within a radius of $R > 0$ in L_1 -distance from the heliport with probability of a least $1 - \tau$.

Example 8.4 (Optimal Watch Spot). In the same spirit, one can think about the following problem. One would like to be closest from $\bar{x} \in \mathbb{R}^3$ to watch an event happening there while being in a safe place with probability of at least $1 - \tau$. For every $s \in [S]$, there is a $\pi^{(s)} \in (0, 1)$ chance that the outcome of a random meteorological phenomenon implies the following: $\zeta^{(s)}$ depicts the center of 3D uniform box of radius $R > 0$ wherein one needs to stay to safely watch the aforementioned event.

8.2 Enumerative complexity

In this section, we show that (MI-CCP) can be posed as the global minimization of an objective function $\mathcal{F}_\tau$ stemming as the minimum of a finite collection of functions. As we will unveil, the domain of these functions differ, yielding an objective $\mathcal{F}_\tau$ not everywhere continuous on $\mathcal{X}$. We recall that both Chapter 6 & 7 of the present thesis were devoted to tackling the global minimization of a minimum of a finite collection of functions. Unfortunately, the methods presented therein are not applicable since they crucially depend on the continuity of the objective as a whole. To obtain a formulation wherein the size of the aforementioned collection is minimal, we underpin the following fact.

Without loss of generality, there will always exist an optimal solution (x^*, z^*) to (MI-CCP), called *minimal solution*, so that

$$\sum_{s \in [N]} \pi^{(s)} z_s^* - \min_{\tilde{s} \in \{s \in [N] \mid z_s^* = 1\}} \pi^{(\tilde{s})} < 1 - \tau. \quad (8.5)$$

Indeed, if (8.5) is not satisfied, $(x^*, z^* - \mathbf{e}_{\tilde{s}})$ becomes admissible and optimal for (MI-CCP) and we can proceed recursively until (8.5) holds. We note that such x^* is closely related to the concept of $(1 - \tau)$ efficient point [DPR00, Lej12]. Every optimal solution (x^*, z^*) induces a *selection* $\hat{S}(z^*) = \{s \in [N] \mid z_s^* = 1\} \subseteq [N]$ of scenarios so that $F_\tau^* = F(x^*) = \nu(\hat{S}(z^*))$, recalling (*S-subproblem*).

We introduce now Definition 8.5 describing the structure of the induced *selections* by *minimal solutions*.

Definition 8.5 (Minimal subset for (MI-CCP)). Let $S \subseteq [N]$. S is called a *minimal subset* for (MI-CCP) if and only if

$$1 - \tau \leq \sum_{s \in S} \pi^{(s)} < 1 - \tau + \min_{\tilde{s} \in S} \pi^{(\tilde{s})}. \quad (8.6)$$

Remark 8.6. Obviously, if (x^*, z^*) is a *minimal solution* then $\hat{S}(z^*)$ is a *minimal subset*. However, if S is a *minimal subset* then the minimizer $x^*(S)$ of $\nu(S)$ in (*S-subproblem*) is not necessarily a global minimizer of (MI-CCP).

We depict by $\mathbf{S}_\tau$ the collection containing every *minimal subset* for (MI-CCP). We highlight in Example 8.7 a first clue of the combinatorial nature of (MI-CCP), i.e. $\mathbf{S}_\tau$ is explicitly given in the case of equiprobable scenarios.

Example 8.7 (Equiprobable scenarios). Let $\pi^{(s)} = 1/N$ for every $s \in [N]$. It comes that (8.6) can be rephrased as

$$N \cdot (1 - \tau) \leq |S| < N \cdot (1 - \tau) + 1. \quad (8.7)$$

From (8.7), one can deduce that $\mathbf{S}_\tau = \{S \subseteq [N] \mid |S| = \lceil N \cdot (1 - \tau) \rceil\}$ so that

$$|\mathbf{S}_\tau| = \binom{N}{\lceil N \cdot (1 - \tau) \rceil}. \quad (8.8)$$

Since (at least) one global minimizer (x^*, z^*) of (MI-CCP) is *minimal* and can be obtained by setting x^* as the minimizer of $\nu(\hat{S}(z^*))$ with $\hat{S}(z^*) \in \mathbf{S}_\tau$

then a strategy to solve (MI-CCP) can work as follows. One iterates over $\mathbf{S}_\tau$, trying one *minimal subset* S at the time by computing $\nu(S)$. Ultimately, it comes

$$F_\tau^* = \min_{S \in \mathbf{S}_\tau} \nu(S). \quad (8.9)$$

Remark 8.8. When all the scenarios are equally as likely (Example 8.7), the exhaustive enumeration requested in (8.9) is straightforward to implement. Under more general distributions, one can solve an incremental sequence of feasibility problems. Starting with an empty collection $\hat{\mathbf{S}}_\tau = \emptyset$, we compute the next *minimal subset* S to review in (8.9) (and to include $\hat{\mathbf{S}}_\tau$ afterwards) as $\hat{S}(z) = \{s \in [N] \mid z_s = 1\}$ where z is a feasible solution of the system

$$\sum_{s=1}^N \pi^{(s)} z_s \geq 1 - \tau \quad (8.10)$$

$$\sum_{s \in S} \pi^{(s)} z_s \leq 1 - \tau - \check{\epsilon} + \pi^{(\check{s})} + (1 - z_{\check{s}}) \quad \forall \check{s} \in [N] \quad (8.11)$$

$$\sum_{s \in S} z_s \leq |S| - 1 \quad \forall S \in \hat{\mathbf{S}}_\tau. \quad (8.12)$$

First (8.10) and middle inequalities (8.11) above implement the requirements of *minimal subsets* with $\check{\epsilon} \in (0, \min_{s \in [N]} \pi^{(s)}/2]$ ensuring the strict inequality of the right-hand side of (8.6). One can notice that, indeed, if $z_s = 0$ (hence $s \notin S = \hat{S}(z)$) the right-hand side of (8.11) is bigger than 1 and the constraint is always valid. The last inequalities (8.12) discard previously seen *minimal subsets* S , stored in $\hat{\mathbf{S}}_\tau$. That being stated, the enumerative approaches described above quickly become impractical, even for moderate values of N . Noteworthy, when $\mathbf{S}_\tau$ is explicit (Example 8.7), the enumerative task becomes parallelizable across $B \geq 1$ units. Yet, unless $B = \Theta(|\mathbf{S}_\tau|)$, the minimal number of (*S-subproblem*) solved in series (hence the wall-clock elapsed time) grows exponentially with N .

As previously announced, we close this section by posing (MI-CCP) as the global minimization of $\mathcal{F}_\tau$, i.e. pointwise minimum of a finite collection of functions $f_S : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{\infty\}$ for every $S \in \mathbf{S}_\tau$. The respective effective domains read

$$\text{dom } f_S = \mathcal{X}(S) := \bigcap_{s \in S} \left(\mathcal{X} \cap \mathcal{R}(\xi^{(s)}) \right).$$

8 | Presolve techniques for finite-support (CCP)

Thereby, one simply writes $F_\tau^* = \min_{x \in \mathbb{R}^d} \mathcal{F}_\tau(x)$ where, for every $x \in \mathbb{R}^d$,

$$\mathcal{F}_\tau(x) := \min_{S \in \mathbf{S}_\tau} F(x) + \mathbb{I}_{\mathcal{X}(S)}(x) = \min_{S \in \mathbf{S}_\tau} f_S(x). \quad (8.13)$$

Remark 8.9. In any situation, $\text{dom } \mathcal{F}$ is an union of $|\mathbf{S}_\tau|$ subsets, i.e.

$$\text{dom } \mathcal{F}_\tau = \bigcup_{S \in \mathbf{S}_\tau} \mathcal{X}(S).$$

Let $\mathcal{X}$ be convex and $c(\cdot, \xi^{(s)})$ be quasi-convex for every $s \in [N]$. The domain of $\mathcal{F}_\tau$ stems as the union of finitely many convex sets. As analyzed in [VAM19] in the case where ξ admits a log-concave continuous distribution, $\text{dom } \mathcal{F}_\tau$ might be convex as whole, depending on parameter τ . Obviously, $\text{dom } \mathcal{F}_0$ is convex. In the continuous case, [VAM19] found that there exists a threshold $\hat{\tau}$ below which every $\tau \leq \hat{\tau}$ leads to a convex $\text{dom } \mathcal{F}_\tau$. It is sometimes true for discrete distributions. We illustrate this in Figure 8.2 where we corroborate the findings of [VAM19] with an empirical $\hat{\tau} \simeq 2 \cdot 10^{-1}$. An **open question** remains assuming we know that $\text{dom } \mathcal{F}_\tau$ is convex so that problem (8.1) turns out to be convex. Relying on a **membership** oracle, i.e. a black-box returning for any $x \in \mathcal{X}$ whether or not the probabilistic constraint is satisfied, a *feasible* starting point x_0 , the possibility to evaluate F at any $x \in \mathcal{X}$ and a **minimization** oracle, i.e. a black-box returning the optimal solution of (*S-subproblem*) for any $S \subseteq [N]$, is it possible to solve problem (8.1) with polynomial complexity in N and d ?

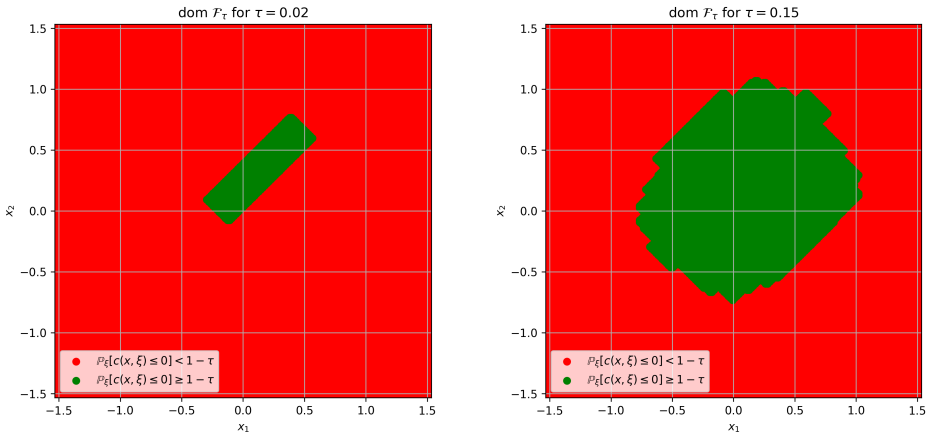


Fig. 8.1 Probabilistic Facility Location: convexity analysis of $\text{dom } \mathcal{F}_\tau$ | convex for $\tau \leq 2 \cdot 10^{-1}$ (left) and nonconvex under $\tau = 1.5 \cdot 10^{-1}$ (right).

8.3 Presolve techniques

Now that the background is all set, we can dive into our presolve techniques. We emphasize the importance of Assumption (B) based on which we derive the useful Lemma 8.10. We recall that for any $z \in \{0, 1\}^N$,

$$\hat{S}(z) = \{s \in [N] \mid z_s = 1\}.$$

Lemma 8.10. Let (x, z) be feasible for (MI-CCP) and $\Xi = \text{conv}(\{\zeta^{(\tilde{s})} \mid \tilde{s} \in \hat{S}(z)\})$. For every $s \in [N]$ such that $z_s = 0$ and $\zeta^{(s)} \in \Xi$, $(x, z + \mathbf{e}_s)$ stays feasible.

Proof. By construction, $c(x, \zeta^{(\tilde{s})}) \leq 0$ for every $\tilde{s} \in [N]$ such that $z_{\tilde{s}} = 1$. Let $\zeta^{(s)} \in \Xi$ so that there exists weights $q^{(\tilde{s})} \in [0, 1]$ summing up to 1 with $\sum_{\tilde{s} \in \hat{S}(z)} q^{(\tilde{s})} \zeta^{(\tilde{s})} = \zeta^{(s)}$. By quasi-convexity (Assumption (B)) of $c(x, \cdot)$,

$$c(x, \zeta^{(s)}) \leq \max_{\tilde{s} \in \hat{S}(z)} c(x, \zeta^{(\tilde{s})}) \leq 0.$$

It comes that if $z_s = 0$ then it is possible to set $z_s = 1$ without changing the value of x hence certifying that $(x, z + \mathbf{e}_s)$ is a feasible solution of (MI-CCP). $\square$

Thus, one should consider feasible (x, z) of *minimal volume* in the sense that there is no *minimal subset* $S \subset \{s \in [N] \mid \zeta^{(s)} \in \text{conv}(\{\zeta^{(\tilde{s})} \mid \tilde{s} \in \hat{S}(z)\})\}$,

$$\text{conv}(\{\zeta^{(s)} \mid s \in S\}) \subsetneq \text{conv}(\{\zeta^{(\tilde{s})} \mid \tilde{s} \in \hat{S}(z)\}). \quad (8.14)$$

Otherwise, $(x, \hat{z})$ with $\hat{z}_s = 1$ for every $s \in S$ and $\hat{z}_s = 0$ is also feasible and furthermore, one has $F_\tau^* \leq \nu(S) \leq \nu(\hat{S}(z))$. When scenarios are equiprobable, for *minimal volume* solutions, *minimal subsets* $S = \hat{S}(z)$ must satisfy

$$s \in S \Leftrightarrow \zeta^{(s)} \in \text{conv}(\{\zeta^{(\tilde{s})} \mid \tilde{s} \in S\}). \quad (8.15)$$

This is not true for non-uniform distributions as shown in Example 8.11.

Example 8.11. Let $N = 5$, $d = 2$ and $\tau = 15/100$. For $s \in [4]$, $\pi^{(s)} = 22/100$ and $\{\zeta^{(s)} \mid s \in [4]\} = \text{vert}(\mathbb{B}_\infty(\mathbf{0}_2, 1))$ so that $\pi^{(5)} = 1 - 4 \cdot 22/100 = 12/100$. The only possible *minimal subset* is $S = [4] = \hat{S}(\mathbf{1}_5 - \mathbf{e}_5)$. Then, if $\zeta^{(5)} = \mathbf{0}_2$, $\zeta^{(5)} \in \text{int}(\mathbb{B}_\infty(\mathbf{0}_2, 1))$ yet $5 \notin S$.

8 | Presolve techniques for finite-support (CCP)

We define now a pair of subsets for the scenarios, i.e. *safe* and *pruned sets* (see Definition 8.12), each of which serving as backbone of one or more techniques and allowing to compute upper/lower bounds.

Definition 8.12 (Safe & Pruned sets). $\oplus, \ominus \subseteq [N]$ is a *safe / pruned sets* pair if including the equalities $z_s = 1$ for every $s \in \oplus$ and $z_s = 0$ for every $s \in \ominus$ in (MI-CCP) does not alter its value F_τ^* .

Note. Throughout the sequel, when we were about to add *scenarios* in $\oplus$ or $\ominus$, we ensured that the refreshed $(\oplus, \ominus)$ stays a valid *safe / pruned sets* pair.

Remark 8.13. As a direct consequence of this definition, if one looks after a sound upper-bound like $\hat{F}_\tau^* = \nu(S) \geq F_\tau^*$, the subset $S \subseteq [N]$ should be such that

$$S \in \mathbf{T}(\oplus, \ominus) := \left\{ S \subseteq [N] \setminus \ominus \mid \oplus \subseteq S \wedge \sum_{s \in S} \pi^{(s)} \geq 1 - \tau \right\}. \quad (8.16)$$

Similar to (8.9), one derives $F_\tau^* = \min_{S \in \mathbf{T}(\oplus, \ominus)} \nu(S)$. Moreover, minimizing F considering only constraints induced by $\oplus$ yields a lower-bound $\check{F}_\tau^* = \nu(\oplus) \leq F_\tau^*$.

8.3.1 Safing techniques

In this subsection, we present sufficient conditions that help to build-up *safe sets* incrementally. That is, starting from $\oplus = \emptyset$, *selectable* indices $s \in [N] \setminus (\oplus \cup \ominus)$ can be tried and if one of the conditions below is triggered, the *safe set* is updated as $\oplus \leftarrow \oplus \cup \{s\}$. We start by presenting a generic condition (Proposition 8.14) applying independently of Assumption (B). The second one (Proposition 8.15), more involved, is really specific to this work. Note however that, from a practical point of view, it can only be efficiently implemented when $p = 2$ or $p = 3$.

First, it happens that based on valid inequalities for (MI-CCP), the remaining feasible domain (or some relaxation of it) is small enough so that a constraint $c(\cdot, \zeta^{(s)})$ becomes non-positive everywhere therein. If so, $s \in \oplus$.

Proposition 8.14 (Non-positivity selection). Let $\hat{F}_\tau^* \geq F_\tau^*$. For any $S \subseteq \oplus$,

$$0 \geq \max_{x \in \mathcal{X}, F(x) \leq \hat{F}_\tau^*} c(x, \zeta^{(s)}) \quad \text{s.t.} \quad c(x, \zeta^{(\tilde{s})}) \leq 0 \quad \forall \tilde{s} \in S$$

implies that s belongs to $\oplus$.

Second, conditional to the current states of sets $\oplus$ and $\ominus$, if s belongs to every *sound* subset $S \in \mathbf{T}(\oplus, \ominus)$ (cfr. Eq. (8.14)), one can conclude that $s \in \oplus$. Through the lens of Lemma 8.10, this statement is equivalent to the following condition.

Proposition 8.15 (Non-separability induction). *If there exists no $S \in \mathbf{T}(\oplus, \ominus)$ such that $\zeta^{(s)}$ is **not** contained in $\Xi = \text{conv}(\zeta^{(\tilde{s})} \mid \tilde{s} \in S)$ then $s \in \oplus$.*

Proof. We need to consider two outcomes. Either problem (MI-CCP) is *infeasible* and $F_\tau^* = \infty$ or there exists an optimal solution (x^*, z^*) with $\hat{S}(z^*) \in \mathbf{T}(\oplus, \ominus)$. In the first case, it comes immediately that $s \in \oplus$ since, by definition, this equality will not affect the optimal value of the problem. Otherwise, it follows that either $z_s^* = 1$ already or $z_s^* = 0$ and $\zeta^{(s)} \in \text{conv}(\zeta^{(\tilde{s})} \mid \tilde{s} \in \hat{S}(z^*))$. In this latter situation, Lemma 8.10 ensures that $(x^*, z^* + \mathbf{e}_s)$ stays *feasible* hence optimal by hypothesis. Thereby, one can include s in $\oplus$. $\square$

Finally, we invoke Corollary 8.16 that allows to expand a *safe set* $\oplus$ with all the *selectable* scenarios contained in the convex hull made of $\oplus$'s scenarios.

Corollary 8.16. *The set $\{s \in [N] \mid \zeta^{(s)} \in \text{conv}(\{\zeta^{(\tilde{s})} \mid \tilde{s} \in \oplus\})\}$ is (also) *safe*.*

Proof. The proof readily follows from Lemma 8.10. $\square$

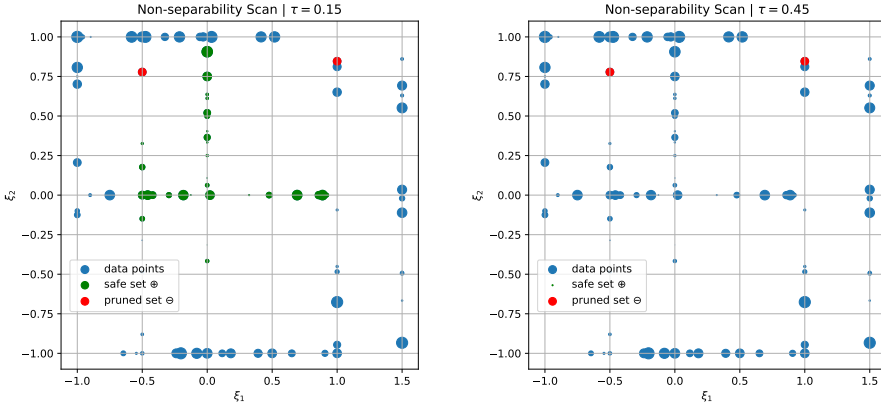


Fig. 8.2 Probabilistic Facility Location: *safe set* incremental building based on Proposition 8.15 ; starting with $\oplus = \emptyset$ and $\ominus = \{11, 74\}$ (toy example) | Initially $|\mathbf{T}(\oplus, \ominus)|$ smaller (left) than (right) hence less conditions to satisfy in order to become certified as non-separable indice s .

8.3.2 Pruning techniques

In the same spirit as Section 8.3.1, yet diametrically opposite in goal, we aim now at discarding scenarios. Again, if one of the conditions below is satisfied, the *pruned set* is updated as $\ominus \leftarrow \ominus \cup \{s\}$.

Akin to Proposition 8.14, it can happen, *at contrario*, that the remaining feasible domain (or some relaxation of it) is small enough so that a constraint $c(\cdot, \zeta^{(s)})$ becomes strictly positive everywhere therein. If so, $s \in \ominus$.

Proposition 8.17 (Strict-positivity exclusion). Let $\hat{F}_\tau^* \geq F_\tau^*$. For any $S \subseteq \oplus$,

$$0 < \min_{x \in \mathcal{X}, F(x) \leq F_\tau^*} c(x, \zeta^{(s)}) \quad \text{s.t.} \quad c(x, \zeta^{(\tilde{s})}) \leq 0 \quad \forall \tilde{s} \in S$$

implies that s belongs to $\ominus$.

It is easy to observe that if a *safe set* $\oplus$ entails enough probability weight, i.e. $\sum_{s \in \oplus} \pi^{(s)} \geq 1 - \tau$ (e.g. a *minimal subset*), it is optimal in the sense $v(\oplus) = F_\tau^*$. Otherwise, $\oplus$ must include (at least) one more index $s \in [N] \setminus (\oplus \cup \ominus)$. If optimizing (MI-CCP) conditional to $z_{\tilde{s}} = 1$ for every $\tilde{s} \in \oplus \cup \{s\}$ yields a value, i.e. $v(\oplus \cup \{s\})$, falling strictly above $\hat{F}_\tau^* \geq F_\tau^*$ then the hypothesis that $z_s = 1$ would not deteriorate the optimal value of (MI-CCP) is rejected. Thereby, the indice s should be *pruned*, as stated in Proposition 8.18.

Proposition 8.18 (Sub-optimality exclusion). Let $\hat{F}_\tau^* \geq F_\tau^*$ and let $\oplus \subseteq [N]$ be a non-optimal safe set. If $v(\oplus \cup \{s\}) > \hat{F}_\tau^*$ then $s \in \ominus$. Moreover, it holds

$$\check{F}_\tau^* = \min_{s \in [N] \setminus (\oplus \cup \ominus)} v(\oplus \cup \{s\}) \leq F_\tau^*. \quad (8.17)$$

8.3.3 Valid inequalities

In some situations, e.g. constraints are decomposable as $c(x, \zeta) = \bar{c}(x) - \zeta$, it is possible to define so-called (*generalized*) *precedence constraints* (see [Rus02] and references therein). That is, one can define a partial order $\leq$ in the ζ -space (e.g. elementwise comparison $\leq$) so that from $\zeta^{(s_1)} \leq \zeta^{(s_2)}$, $z_{s_1} \leq z_{s_2}$ becomes a valid inequality for (MI-CCP). Here, we extend the concept of *precedence constraints* without accounting for the decomposability of $c(x, \zeta)$. Instead of a simple comparison between two vectors in the ζ -space, we build our valid inequalities by answering the question:

Is $\zeta^{(s)}$ strictly contained in a convex hull formed by scenarios from $\tilde{S} \subseteq [N]$?

If so, our first set of inequalities is an equivalent of $\min_{\tilde{s} \in \tilde{S}} z_{\tilde{s}} \leq z_s$.

Under the assumption of equiprobable scenarios, our second set of valid inequalities state that if there is provably *too much* probability mass associated with scenarios contained in a prescribed convex hull, at least one of its scenario vertices $\zeta^{(\tilde{s})}$ should not be selected (i.e. $z_{\tilde{s}} = 0$).

Let $S \subseteq [N]$ and $\Xi = \text{conv}(\zeta^{(\tilde{s})} \mid \tilde{s} \in S)$. We denote by $\mathbf{V}(S)$ indices of scenarios that are vertices for Ξ , i.e. $\mathbf{V}(S) := \{s \in S \mid \zeta^{(s)} \in \text{vert}(\Xi)\}$ and we depict by $\mathbf{N}(S)$ indices of all the scenarios included in Ξ , i.e. $\mathbf{N}(S) := \{s \in S \mid \zeta^{(s)} \in \Xi\}$. We are now ready to formulate two sets of valid inequalities.

Proposition 8.19 (Convex-hull inductions). The following inequalities are valid

$$z_s \geq \sum_{\tilde{s} \in \mathbf{V}(S)} z_{\tilde{s}} - |\mathbf{V}(S)| + 1 \quad \forall s \in \mathbf{N}(S) \setminus \mathbf{V}(S). \quad (8.18)$$

If scenarios are equally as likely and $\sum_{s \in \mathbf{N}(S)} \pi^{(s)} \geq (1 - \tau) + N^{-1}$ then

$$\sum_{\tilde{s} \in \mathbf{V}(S)} z_{\tilde{s}} \leq |\mathbf{V}(S)| - 1. \quad (8.19)$$

Proof. Deduced from Lemma 8.10, if $\zeta^{(s)}$ is contained in $\mathbf{N}(S)$ without being one of its vertices, it can be selected (i.e. $z_s = 1$) without loss of generality if all the vertices are. Next, let us assume that scenarios are equiprobable and that $\sum_{s \in \mathbf{N}(S)} \pi^{(s)} \geq (1 - \tau) + N^{-1}$. In this case, we know from (8.13) & (8.15) that there exists an optimal solution (x^*, z^*) to (MI-CCP) such that $\hat{S}(z^*) = \mathbf{N}(\hat{S}(z^*))$ and $\sum_{s \in \hat{S}(z^*)} \pi^{(s)} < (1 - \tau) + N^{-1}$. Thereby, we can discard solutions inducing vertices of a convex hull in the ζ -space that entails *too much* probability mass, i.e. $(1 - \tau) + N^{-1}$. $\square$

8.4 Numerical Experiments

We focus here on the problem of Example 8.3 (and its straightforward generalization for $p = 3$) to showcase the benefits of the following presolve routine. We initialize $(\oplus, \ominus) = (\emptyset, \emptyset)$ and $(\check{F}_\tau^*, \hat{F}_\tau^*) = (\infty, \infty)$.

- For every $s \in [N]$, we solve problem $\nu(\{s\})$, record its minimizer $x^*(\{s\})$, update $\check{F}_\tau^* \leftarrow \min\{\check{F}_\tau^*, \nu(\{s\})\}$ and, if $x^*(\{s\}) \in \text{dom } \mathcal{F}_\tau$, $\hat{F}_\tau^* \leftarrow \min\{\hat{F}_\tau^*, \nu(\{s\})\}$. At the end, $\check{F}_\tau^* \leq F_\tau^* \leq \hat{F}_\tau^*$. If $\hat{F}_\tau^* < \infty$, we set $\ominus \leftarrow \{s \in [N] \mid \nu(\{s\}) > \hat{F}_\tau^*\}$.

<i>method</i>	p	τ	avg. time [s]	# insts. solved	avg. $(\mathbf{UB} - F_\tau^*)/F_\tau^*$
presolve direct	2	5%	211.75 321.07	5/5 3/5	0% 1.15%
presolve direct	2	15%	157.41 (TL)	5/5 0/5	0% 110.3% (excl. 1 outlier)
presolve direct	3	5%	438.91 (TL)	4/5 0/5	$\leq 0.8\%$ 14.31%
presolve direct	3	15%	363.24 (TL)	3/5 0/5	$\leq 7.49\%$ 122.8%

Table 8.1 Comparison between use of presolve and direct resolution

- For every $s \in [N] \setminus (\oplus \cup \ominus)$, condition of Proposition 8.15 is checked to possibly add s in $\oplus$. It is advised to set a time limit after which one stops the checks. Indeed, our implementation of the *non-seperability induction* check is of complexity $\mathcal{O}(N^3)$ for $p = 3$ compared to $\mathcal{O}(N \log N)$ for $p = 2$. Thus, we chose 60[s] (never reached in practice) for $p = 2$ and 120[s] for $p = 3$.
- We apply Corollary 8.16 (possibly extending $\oplus$) and then implement *sub-opti. exclusion* checks for every $s \in [N] \setminus (\oplus \cup \ominus)$ (possibly extending $\ominus$).
- We finish by tightening big- M bounds as specified in (8.3).

Benchmarks All the details about our implementations and the data generation are freely available on [GitHub](#). Parameters in $(\text{PBP}-(p, o, \delta))$ were taken as $N = 500$, $p \in \{2, 3\}$, $o = 2$, $\delta = 1$ and $R = \frac{23}{25} \cdot p \cdot \max_{s \in [N]} \|\zeta^{(s)}\|_\infty$.

For each level $\tau \in \{0.05, 0.15\}$, 5 independent random datasets $\mathbf{D}$ were drawn and fed the model. An overall time limit (TL) was set to 500[s] ($p = 2$) and 720[s] ($p = 3$) to solve the problem. We report on Table 8.1 the results: **UB** (respectively **LB**) stands as the best upper-bound (respectively lower-bound) found during a solve.

Comments The fourth column represents the mean time taken by an algorithm, either *presolve* or *direct*, when it did success in providing the optimal solution (i.e. $\mathbf{UB} = \mathbf{LB} = F_\tau^*$). The last column represents the average *a posteriori* relative optimality gap achieved. If ever the optimal value F_τ^* of a given instance was not found within time limit for both *methods*, we launched again *presolve* for a long time period until we obtained it.

As clearly shown in the results reported above, our preliminary numerical experiments demonstrate that the use of our presolve techniques is computationally beneficial. We leave as future research the possibility to incorporate our valid inequalities (8.18) and (8.19) in (MI-CCP) and to extend the applicability of our methodology to higher dimensions $p > 3$.

8.4.1 Bisection Heuristic

Property 8.14, 8.17 and 8.18 rely, among others, on a good upper-bound $\hat{F}_\tau^* \geq F_\tau^*$. Unlike our core and most effective tool, Proposition 8.15 (only applicable (in practice) for $p \in \{2, 3\}$), the other conditions can still be considered when $p > 3$. Therefore, in this last section, we propose a *bisection heuristic* (Algorithm 6) tackling problem (8.1) directly. Hopefully, our algorithm returns a feasible point $\tilde{x} \in \mathcal{X}$ of the problem and, as a byproduct, the upper-bound $\hat{F}_\tau^* = F(\tilde{x})$ we are striving for.

(DC) reformulation of a probabilistic constraints

Inspired from [dO19] (Section 6.2), we start by approximating the probabilistic constraint. Let $\iota > 0$ be a small parameter. For every $x \in \mathbb{R}^d$,

$$1 - \mathbb{P}_\xi[c(x, \xi) \leq 0] \simeq \iota^{-1} (\mathbb{E}_\xi[\max\{0, c(x, \xi) + \iota\}] - \mathbb{E}_\xi[\max\{0, c(x, \xi)\}]).$$

Then, given that ξ has finite support in (8.1), we write

$$\begin{aligned} \mathbb{E}_\xi[\max\{0, c(x, \xi) + \iota\}] &= \varrho_\iota(x) = \sum_{s=1}^N \pi^{(s)} \cdot \max\{0, c(x, \xi^{(s)}) + \iota\}, \\ \mathbb{E}_\xi[\max\{0, c(x, \xi)\}] &= \varrho_0(x) = \sum_{s=1}^N \pi^{(s)} \max\{0, c(x, \xi^{(s)})\}. \end{aligned}$$

Our probabilistic constraint can be simplified as $\varrho_\iota(x) - \varrho_0(x) \leq \tau \cdot \iota$. However, this latter is still nonconvex in x . It is (DC) in general and might be (DSMsC) if c is smooth (cfr. Section 4.3.2).

Feasibility maximization paradigm

Let $\hat{v} \in \mathbb{R}$. One would like to test whether $\hat{v}$ is a valid upper-bound, i.e. if $\hat{v} \geq F_\tau^*$ is a correct inequality. The inequality is correct if and only if one can find some *feasible* $x_{\hat{v}} \in \mathcal{X}$ such that $F(x_{\hat{v}}) \leq \hat{v}$. If such a $x_{\hat{v}}$ does not exist

8 | Presolve techniques for finite-support (CCP)

then $\hat{\nu} < F_{\tau}^*$, otherwise it must hold

$$\mathbb{P}_{\xi}[c(x_{\hat{\nu}}, \xi) \leq 0] = \sum_{s=1}^N \pi^{(s)} \cdot \mathbb{1}(c(x_{\hat{\nu}}, \xi^{(s)}) \leq 0) \geq 1 - \tau. \quad (8.20)$$

As shown in [HYZ11], for every $(x, \xi) \in \mathbb{R}^d \times \mathbb{R}^p$,

$$\mathbb{1}(c(x, \xi) \geq 0) \leq \iota^{-1} (\max\{0, c(x, \xi) + \iota\} - \max\{0, c(x, \xi)\})$$

hence

$$\iota^{-1} (\varrho_{\iota}(x) - \varrho_0(x)) \geq 1 - \mathbb{P}_{\xi}[c(x, \xi) \leq 0]. \quad (8.21)$$

If (8.20) is satisfied then it must also hold $\varrho_{\iota}(x) - \varrho_0(x) \leq \tau \cdot \iota$.

Therefore, assuming that computing a minimizer $x_{\hat{\nu}}^*$ of the (DC) program

$$\min_{x \in \mathcal{X}} \underbrace{\varrho_{\iota}(x)}_{f_1(x)} - \underbrace{\varrho_0(x)}_{f_2(x)} \quad \text{s.t.} \quad F(x) \leq \hat{\nu}. \quad (\text{biseq-Oracle})$$

is possible, we could use the proxy $\varrho_{\iota}(x_{\hat{\nu}}^*) - \varrho_0(x_{\hat{\nu}}^*) > \tau \cdot \iota$ to decide that $\hat{\nu}$ is not a valid upper-bound, i.e. $\hat{\nu} < F_{\tau}^*$. The goal of (biseq-Oracle) is to *maximize the feasibility* of decision vector x with respect to the probabilistic constraint on the sub-level set $\{x \in \mathcal{X} \mid F(x) \leq \hat{\nu}\}$. Unfortunately, solving (biseq-Oracle) exactly is a difficult task. Yet, one can apply DCA (see [LP18]) and obtain an (approximate) *critical point* $\tilde{x}_{\hat{\nu}} \in \mathcal{X}$. We note that if c is smooth, it is sometimes possible to chase stronger *stationary points*. Let $I_+ \subseteq [N]$ (respectively I_-) denotes the indices $s \in [N]$ such that $c(x, \xi^{(s)}) > 0$ (respectively $c(x, \xi^{(s)}) = 0$). Any subgradient $g_2(x)$ of f_2 at x in (biseq-Oracle) can be written as

$$g_2(x) = \sum_{s \in I_+} \pi^{(s)} \cdot \nabla_x c(x, \xi^{(s)}) + \sum_{s \in I_-} q_s \cdot \pi^{(s)} \cdot \nabla_x c(x, \xi^{(s)})$$

where $q_s \in [0, 1]$ for every $s \in I_-$. It comes that $\partial f_2(x)$ is a bounded polyhedron from which one could theoretically enumerate the vertices. Let $\tilde{x}_{\hat{\nu}} \in \mathcal{X}$ be such that $F(\tilde{x}_{\hat{\nu}}) \leq \hat{\nu}$. The point $\tilde{x}_{\hat{\nu}}$ can be certified *stationary* if using the DCA **minimization** oracle fed with every $v_2 \in \mathbf{vert}(\partial f_2(\tilde{x}_{\hat{\nu}}))$,

$$x_+(v_2) \leftarrow \arg \min_{x \in \mathcal{X}} \varrho_{\iota}(x) - \langle v_2, x \rangle \quad \text{s.t.} \quad F(x) \leq \hat{\nu}$$

does not improve the objective; $f_1(x_+(v_2)) - f_2(x_+(v_2)) \geq f_1(\tilde{x}_{\hat{\nu}}) - f_2(\tilde{x}_{\hat{\nu}})$.

In Algorithm 6, we tacitly assume that $\tilde{x}_{\hat{v}}$ (either a *critical* point, a *stationary* point or even, perhaps, a *global minimum* of (bisec-Oracle)) *maximizes* the aforementioned *feasibility*. Then, based on the value of $\mathbb{P}_{\tilde{\zeta}}[c(\tilde{x}_{\hat{v}}, \tilde{\zeta}) \leq 0] = \sum_{s=1}^N \pi^{(s)} \cdot \mathbb{1}(c(\tilde{x}_{\hat{v}}, \tilde{\zeta}^{(s)}) \leq 0)$, we will decide whether $\hat{v}$ is an upper-bound or not. This decision tool constitutes the backbone of our algorithm.

Remark 8.20. Although difficult to prove rigorously, it has been empirically observed that $q_1 - q_0$ involved in (bisec-Oracle) is highly nonsmooth and nonconvex when the number of *scenarios* N is small... Fortunately enough, $q_1 - q_0$ becomes smoother as N grows and easier to optimize.

Bisection

As for any bisection algorithm, we need two *crude* bounds, i.e. a lower-bound $\mathbf{LB} \leq F_{\tau}^*$ and an upper-bound $\mathbf{UB} \geq F_{\tau}^*$, to initialize Algorithm 6. Then, according to the methodology detailed in the previous paragraph (*feasibility maximization paradigm*), we test whether $\hat{v} = (\mathbf{UB} + \mathbf{LB})/2$ is a valid upper-bound. If it is, $\mathbf{UB}$ is updated with the value $\hat{v}$ otherwise, we set $\mathbf{LB} = \hat{v}$. This procedure is executed iteratively until $\mathbf{UB} - \mathbf{LB} < \delta$.

Algorithm 6 DCA-bisec (LB, UB)

Require: $\mathbf{LB} \leq F_{\tau}^* \leq \mathbf{UB}$, tolerance $\delta \geq 0$

```

while  $\mathbf{UB} - \mathbf{LB} \geq \delta$  do
     $\hat{v} = (\mathbf{UB} + \mathbf{LB})/2$ 
     $\tilde{x}_{\hat{v}} \sim \arg \min_{x \in \mathcal{X}} q_i(x) - q_0(x) \quad \text{s.t.} \quad F(x) \leq \hat{v} \quad \triangleright (\text{bisec-Oracle})$ 
    if  $\mathbb{P}_{\tilde{\zeta}}[c(\tilde{x}_{\hat{v}}, \tilde{\zeta}) \leq 0] \geq 1 - \tau$  then
         $\mathbf{UB} \leftarrow \hat{v}$ 
         $\tilde{x} = \tilde{x}_{\hat{v}}$ 
    else
         $\mathbf{LB} \leftarrow \hat{v}$ 
    end if
end while
return  $\tilde{x}$ 

```

Remark 8.21. Algorithm 6 display similarities with [ALSX17] (Algorithm 1). Moreover, both algorithms share the same goal. For the sake of completeness, we present here below how Algorithm 1 in [ALSX17] elects $\tilde{x}_{\hat{v}}$ when a candidate upper-bound $\hat{v}$ is tried out.

$$\tilde{x}_{\hat{v}} \in \arg \min_{x \in \mathcal{X}, u \in \mathbb{R}_+^N} \frac{1}{N} \sum_{s=1}^N \max\{c(x, \tilde{\zeta}^{(s)}), u_s\} \quad \text{s.t.} \quad F(x) \leq \hat{v} \quad (8.22)$$

Assuming that $c(x, \xi)$ and $F(x)$ are convex in x , we note that problem (8.22) is convex (C) whereas ours is (DC). The respective solving of (bisec-Oracle) and (8.22) possibly leads to different $\tilde{x}_\varphi$ and, therefore, different final solutions $\tilde{x}$ returned by Algorithm 6 and [ALSX17] (Algorithm 1) respectively.

8.5 Conclusion

In this chapter, we were concerned with optimization problems involving *probabilistic constraints*, relaxing the *robust optimization* framework and its *deterministic* constraints. We have developed tools (presolve techniques) aiming at cutting down the complexity of finding a global optimal solution to the problem. Among these tools, some leveraged the quasi-convexity of the probabilistic constraint with respect to its defining random vector ξ . We have shown that our presolve techniques, especially those taking advantage of the aforementioned quasi-convexity, were very effective in reducing the overall computational time to solve our chance-constrained programs when p (dimension of ξ) is either 2 or 3. Unfortunately, their practical implementability as well as their efficiency will inevitably decline with higher p . Nevertheless, the other presolve techniques that we introduced do not suffer the *curse of dimensionality* and could still be used with $p > 3$. These rely on a good upper-bound on the optimal objective value of the problem. As a minor contribution, we suggest a new approach to find such upper-bound. We leave for further research the task of choosing which (DC) method provides the best (empirical) solutions to (bisec-Oracle).



Discussion

At the end of this thesis, it is time to draw some general conclusions about the research topics investigated. This final section also represents a good opportunity to provide more personal insights about the results we have been obtaining for more than 5 years now. In other words,

What is my take on that matter ?

Moreover, we would like to suggest a couple of perspectives for future research. What are the challenges for tomorrow ?

Tunable (Inexact) Oracles From the theoretical point of view, there is no ambiguity. One should definitely adapt the amount δ_k of inexactness within iteration k of a Tunable Oracle Method (TOM) when there exists some *a priori* discrepancy regarding the respective impacts of $\delta_0, \dots, \delta_{K-1}$ in its convergence guarantees. As prescribed in equation (3.33), a sound rule of thumb suggests to fix the inexactness δ_k as

$$\delta_k \propto \left(\frac{b_k}{a_k} \right)^{\frac{1}{r+1}}.$$

We recall that coefficients a_k represent *impact factors*, i.e. δ_k inexactness at iteration k will harm a convergence bound by an amount of $a_k \cdot \delta_k$. Coefficients b_k and r stand for *relative oracle costs* and *oracle complexity parameter*,

i.e. requesting δ_k inexactness at iteration k costs b_k/δ_k^r . We note that when $r \rightarrow 0$, the oracle cost appears almost constant, i.e. independent of the accuracy requested. Typically, iterations become more and more accurate as k grows, i.e. $\delta_k \rightarrow 0$. Although working well in practice, our *offline* and *online* optimal inexactness schedules are constrained to be constant schedules, i.e. $\delta_k = \bar{\delta}$ for every $k \geq 0$, if the ratios b_k/a_k are constant themselves. It turns out that some TOM (e.g. iPL from Example 3.5 with t_k constant for every $k \geq 0$) exhibit no *a priori* discrepancy, i.e. b_k/a_k is constant for them. Our intuition is that these TOM could benefit from decaying inexactness schedules, however our theory does not predict any *a priori* computational gains doing so. It remains unclear how to deal with inexactness in these cases.

Future research directions include answering the above question and studying optimal inexactness schedules for other types of impact models. Indeed, we considered in this thesis the *additive impact* of inexactness in a given convergence upper-bound. It could be interesting to analyze models with *multiplicative impacts*.

Optimization of DC & CCP After years of readings and numerical practices, I would say to anyone asking that *true* nonconvex optimization, as in (DC) or (CCP), is hard... no matter what. This statement should be understood as follows. If your task is to find and/or certify that a given solution x of a nonconvex problem is *globally* optimal or even *locally* optimal, there will always be pathological problem instances for which (if even possible) this would take a computational time scaling badly (e.g. exponentially) with some dimensions of the problem. We gave a glimpse of this in Proposition 7.2. Yet, as we did throughout this thesis, we can often exploit a lot of structure to obtain good local solutions. About (DC) programming, my takeaway message is unequivocal: when there is more structure than the usual (DC) decomposition (e.g. (SMC)) one should use it and expect better solutions. Similarly, we have shown that if the structure at hand allows it, reducing (CCP)'s size is a must do to save computational time.

In summary, despite poor guarantees concerning the quality of the obtained solutions, the modelization power associated with nonconvex problems is worth consideration. In my humble opinion, I think that striving for the best solution possible of a nonconvex or mixed-integer convex formulation often yields more satisfactory results than expected.

In what concerns mixed-integer convex programs, it happens that a global solver finds a global minimum of a given problem quite fast. Yet, it would take ages and so many *cuts* to certify the global optimality... Certified or not, the returned solution could look just as fine to the user.

I finish by recalling Remark 8.9, raising another research direction in what concerns (CCP). Indeed, it turns out that (CCP) is convex sometimes. Yet, even if the practitioner was aware of that fact (an unrealistic assumption), it is unclear what can be done to take advantage of this information. Finally, the (DC) world being so broad, I would be surprised if there exists no more subclasses with exploitable structure. Both these points could be addressed in future research.

Bibliography

- [AAV20] Francisco J. Aragón Artacho and Phan T. Vuong. The Boosted Difference of Convex functions algorithm for nonsmooth functions. *SIAM Journal on Optimization*, 30(1):980–1006, 2020.
- [ACV22] Francisco J. Aragón Artacho, Rubén Campoy, and Phan Tu Vuong. The Boosted DC algorithm for linearly constrained DC programming. *Set-Valued and Variational Analysis*, 30:1265–1289, 2022.
- [ALSX17] Shabbir Ahmed, James Luedtke, Yongjia Song, and Weijun Xie. Nonanticipative duality, relaxations, and formulations for chance-constrained stochastic programs. *Math. Prog.*, 162:51–81, 2017.
- [AV07] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07*, pages 1027–1035, USA, 2007. Society for Industrial and Applied Mathematics.
- [BAB20] S. Barratt, G. Angeris, and S. Boyd. Minimizing a sum of clipped convex functions. *Optimization Letters*, 14:2443–2459, 2020.
- [Bal09] Erik Balder. On generalized gradients and optimization. 11 2009. URL: https://www.researchgate.net/publication/239803129_On_generalized_gradients_and_optimization.

- [Bal18] Egon Balas. *Disjunctive Programming*. Springer Nature Switzerland AG, 2018.
- [Bec17] Amir Beck. *First-Order Methods in Optimization*, chapter 10. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017. doi:10.1137/1.9781611974997.
- [Ber09] D. Bertsekas. *Convex Optimization Theory*. Athena Scientific optimization and computation series. Athena Scientific, 2009. URL: https://web.mit.edu/dimitrib/www/Convex_Theory_Entire_Book.pdf.
- [BKB08] A. Bagirov and Sezer M. Karasozen B. Discrete gradient method: Derivative-free method for nonsmooth optimization. *Journal of Optimization Theory and Applications*, 137(2):317–334, 2008.
- [BMS13] V. Boltyanski, H. Martini, and V. Soltan. *Geometric Methods and Optimization Problems*. Combinatorial Optimization. Springer US, 2013.
- [BNK09] D. Bertsimas, O. Nohadani, and M. T. Kwong. Robust optimization for unconstrained simulation-based problems. *Acta Numerica*, 58(1):161–178, 2009.
- [BPC⁺11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3:1–122, 2011.
- [BR14] D.P. Bertsekas and W. Rheinboldt. *Constrained Optimization and Lagrange Multiplier Methods*. Computer science and applied mathematics. Elsevier Science, 2014.
- [BT09] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.
- [BT12] Amir Beck and Marc Teboulle. Smoothing and first order methods: A unified framework. *SIAM Journal on Optimization*, 22(2):557–580, 2012.

- [BTK⁺22] A.M. Bagirov, S. Taheri, N. Karmitsa, N. Sultanova, and S. Asadi. Robust piecewise linear L1-regression via nonsmooth DC optimization. *Optimization Methods and Software*, 37(4):1289–1309, 2022.
- [BV04] Stephen Boyd and Lieven Vandenbergh. *Convex optimization*. Cambridge university press, 2004.
- [BZ03] A.M. Bagirov and J Zhang. Hybrid simulating annealing method and discrete gradient method for global optimization. *Proceedings of Industrial Mathematics Symposium*, 2003.
- [CCHP20] Y. Cui, TH. Chang, M. Hong, and Jong-Shi Pang. A study of piecewise linear-quadratic programs. *Journal of Optimization Theory and Applications*, 186:523–553, 2020.
- [CHW22] H. Chen, H. He, and Y. et al. Wang. An efficient alternating minimization method for fourth degree polynomial optimization. *Journal of Global Optimization*, 82:82–103, 2022.
- [CK01] Frank Y. Chen and Dmitry Krass. Inventory models with minimal service level constraints. *European Journal of Operational Research*, 134(1):120–140, 2001.
- [CP16] Antonin Chambolle and Thomas Pock. An introduction to continuous optimization for imaging. In *Cambridge University Press (CUP)*, volume 25, pages 161–319. Acta Numerica, 2016.
- [CRT90] Jeffrey D. Camm, Amitabh S. Raturi, and Shigeru Tsubakitani. Cutting big M down to size. *Interfaces*, 20(5):61–66, 1990.
- [DB16] Steven Diamond and Stephen P. Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of machine learning research : JMLR*, 17, 2016.
- [DCWY24] Lisang Ding, Ziang Chen, Xinshang Wang, and Wotao Yin. Efficient algorithms for sum-of-minimum optimization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 10927–10959. PMLR, 21–27 Jul 2024.
- [Dev13] O. Devolder. Exactness, inexactness and stochasticity in first-order methods for large-scale convex optimization. In *Dctoral Thesis*, 2013.

- [DG24a] Guillaume Van Dessel and François Glineur. Global minimization of a pointwise minimum of a finite collection of functions. *arXiv:2412.04625*, 2024.
- [DG24b] Guillaume Van Dessel and François Glineur. Optimal inexactness schedules for tunable oracle-based methods. *Optimization Methods and Software*, 39(3):664–698, 2024. [arXiv:https://doi.org/10.1080/10556788.2023.2296982](https://doi.org/10.1080/10556788.2023.2296982), doi: 10.1080/10556788.2023.2296982.
- [DG25a] Guillaume Van Dessel and François Glineur. Global and local approaches for the minimization of a sum of pointwise minima of convex functions. *arXiv:2501.19372*, 2025.
- [DG25b] Guillaume Van Dessel and François Glineur. Presolve techniques for quasi-convex chance constraints with finite-support low-dimensional uncertainty. *arXiv:2504.05785*, 2025.
- [DGN13] Olivier Devolder, François Glineur, and Yurii Nesterov. First-order methods with inexact oracle: the strongly convex case. LIDAM Discussion Papers CORE 16, 2013. URL: <https://EconPapers.repec.org/RePEc:cor:louvco:2013016>.
- [DJWW13] John C. Duchi, Michael I. Jordan, Martin J. Wainwright, and Andre Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61:2788–2806, 2013.
- [DL21] G. Dias and L. Liberti. Exploiting symmetries in mathematical programming via orbital independence. *Ann. Oper. Res.*, 298:149–182, 2021.
- [DN20] Nikita Doikov and Yurii Nesterov. Inexact tensor methods with dynamic accuracies. In *ICML*, 2020.
- [dO19] W. de Oliveira. Proximal bundle methods for nonsmooth DC programming. *Journal of Global Optim.*, 75:523–563, 2019.
- [dO20] Welington de Oliveira. Sequential Difference-of-Convex Programming. *Journal of Optimization Theory and Applications*, 186, 09 2020.

- [dO23] W. de Oliveira. A note on the Frank-Wolfe algorithm for a class of nonconvex and nonsmooth optimization problems. *Open Journal of Mathematical Optimization*, 4:1–10, 2023.
- [dO25] W. de Oliveira. Derivative-free approaches for chance-constrained problems with right-hand side uncertainty. *SIAM Journal on Optimization*, 35(1):1–27, 2025.
- [dOS14] Welington de Oliveira and Claudia Sagastizábal. Level bundle methods for oracles with on-demand accuracy. *Optimization Methods and Software*, 29(6):1180–1209, 2014.
- [dOS20] Welington de Oliveira and Mikhail Solodov. *Bundle Methods for Inexact Data*. Springer International Publishing, 2020.
- [DP19] D Drusvyatskiy and C Paquette. Efficiency of minimizing compositions of convex functions and smooth maps. In *Mathematical Programming*, volume 178, pages 503–558, 2019.
- [DPR00] Darinka Dentcheva, András Prékopa, and Andrzej Ruszczyński. Concavity and efficient points of discrete distributions in probabilistic programming. *Math. Prog.*, 89:55–77, 2000.
- [DT97] Tao Pham Dinh and Hoai An Le Thi. Convex analysis approach to D.C. programming: Theory, algorithm and applications. 1997. URL: <https://api.semanticscholar.org/CorpusID:15546259>.
- [DT19] M.N. Dao and M.K. Tam. Union averaged operators with applications to proximal algorithms for min-convex functions. *Journal of Optimization Theory and Applications*, 181:61–94, 2019.
- [GS00] Luigi Grippo and Marco Sciandrone. On the convergence of the block nonlinear gauss-seidel method under convex constraints. *Oper. Res. Lett.*, 26:127–136, 2000.
- [GTBK23] Manlio Gaudioso, Sona Taheri, Adil M. Bagirov, and Napsu Karmitsa. Bundle enrichment method for nonsmooth difference of convex programming problems. *Algorithms*, 16(8), 2023.

- [HLP21] Vinh Thanh Ho, Hoai An Le Thi, and Tao Pham Dinh. DCA-based algorithms for DC fitting. *Journal of Computational and Applied Mathematics*, 389:113353, 2021.
- [HN09] Horst W. Hamacher and Stefan Nickel. *Multifacility and Restricted Location Problems*, pages 2409–2413. Springer US, Boston, MA, 2009.
- [HNKK17] Nam Ho-Nguyen and Fatma Kiliç-Karzan. A second-order cone based approach for solving the Trust-Region Subproblem and its variants. *SIAM Journal on Optimization*, 27(3):1485–1512, 2017.
- [HT20] Nadav Hallak and Marc Teboulle. Finding second-order stationary points in constrained minimization: A feasible direction approach. In *Journal of Optimization Theory and Applications*, volume 186, pages 480–503, 2020.
- [Hua20] C. Huang. Relu networks are universal approximators via piecewise linear or constant functions. *Neural Computations.*, 32(11):2249–2278, 2020.
- [HYZ11] L. Jeff Hong, Yi Yang, and Liwei Zhang. Sequential convex approximations to joint chance constrained programs: A Monte Carlo approach. *Operations Research*, 59(3):617–630, 2011.
- [Jem20] L. Jemison. On dc and local dc functions. *University Honors Theses*, 910, 2020.
- [JHZ21] Bo Jiang, Wang Haoyue, and Shuzhong Zhang. An optimal high-order tensor method for convex optimization. *Mathematics of Operational Research*, 46(4):1390–1412, 2021.
- [JT15] Prateek Jain and Ambuj Tewari. Alternating minimization for regression problems with vector-valued outputs. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.
- [KBLG18] Jan Kronqvist, David E. Bernal, Andreas Lundell, and Ignacio E. Grossmann. A review and comparison of solvers for convex MINLP. *Optimization and Engineering*, 20:397–455, 2018.

- [Kiw95] Krzysztof C. Kiwiel. Approximations in proximal bundle methods and decomposition of convex programs. *Journal of Optimization Theory and Applications*, 84:529–548, 1995.
- [Kiw06] Krzysztof C. Kiwiel. A proximal bundle method with approximate subgradient linearizations. *SIAM Journal on optimization*, 16(4):1007–1023, 2006.
- [KJ22] Simge Kucukyavuz and Ruiwei Jiang. Chance-constrained optimization under limited distributional information: A review of reformulations based on sampling and distributional robustness. *EURO Journal on Computational Optimization*, 10, 2022.
- [KL98] Muralidharan S. Kodialam and Hanan Luss. Algorithms for separable nonlinear resource allocation problems. *Operations Research*, 46(2):272–284, 1998.
- [KL17] Jonas Moritz Kohler and Aurélien Lucchi. Sub-sampled cubic regularization for non-convex optimization. In *ICML*, 2017.
- [KLBC19] Ali Kavis, Kfir Yehuda Levy, Francis R. Bach, and Volkan Cevher. Unixgrad: A universal, adaptive algorithm with optimal guarantees for constrained optimization. In *NeurIPS*, 2019.
- [KS87] Anita Kripfanz and R. Schulze. Piecewise affine functions as a difference of two convex functions. *Optimization*, 18(1):23–29, 1987.
- [KT51] H. W. Kuhn and A.W. Tucker. Nonlinear programming. *Berkeley Symp. on Math. Statist. and Prob.*, pages 481–492, 1951.
- [Lej12] Miguel A Lejeune. Pattern-based modeling and solution of probabilistically constrained optimization problems. *Op. Research*, 60(6):1356–1372, 2012.
- [Lem74] C. Lemarechal. An algorithm for minimizing convex functions. *Inf. Process.*, 1:552–556, 1974.
- [Lib12] Leo Liberti. Symmetry in mathematical programming. In *Mixed Integer Nonlinear Programming*, pages 263–283, New York, NY, 2012. Springer New York.

- [LJ16a] Simon Lacoste-Julien. Convergence rate of Frank-Wolfe for non-convex objectives. *arXiv:1607.00345*, 2016.
- [LJ16b] J. Lu and M. Johansson. Convergence analysis of approximate primal solutions in dual first-order methods. In *SIAM J. Optim.*, volume 26(4), pages 2430–2467, 2016.
- [LJ19] Tzu-Ying Liu and Hui Jiang. Minimizing sum of truncated convex functions and its applications. *Journal of Computational and Graphical Statistics*, 28(1):1–10, 2019.
- [LL22] Huan Li and Zhouchen Lin. Restarted nonconvex accelerated gradient descent: No more polylogarithmic factor in the $\mathcal{O}(\epsilon^{-7/4})$ complexity. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 12901–12916. PMLR, 17–23 Jul 2022.
- [Llo82] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. doi:10.1109/TIT.1982.1056489.
- [LP18] Hoai An Le Thi and Tao Pham Dinh. DC programming and DCA: thirty years of developments. *Math. Program.*, 169:5–68, 2018.
- [LSM20a] Jiajin Li, Anthony Man-Cho So, and Wing-Kin Ma. Understanding notions of stationarity in nonsmooth optimization: A guided tour of various constructions of subdifferential for nonsmooth functions. *IEEE Signal Processing Magazine*, 37(5):18–31, 2020.
- [LSM20b] Jiajin Li, Anthony Man-Cho So, and Wing-Kin Ma. Understanding notions of stationarity in nonsmooth optimization: A guided tour of various constructions of subdifferential for nonsmooth functions. *IEEE Signal Processing Magazine*, 37(5):18–31, 2020.
- [LSX⁺17] Guodong Liu, M Starke, B Xiao, Xiaohu Zhang, and Kevin Tomsovic. Microgrid optimal scheduling with chance-constrained islanding capability. *Electric Power Systems Research*, 145:197–206, 2017.

- [Lue14] James Luedtke. A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support. *Math. Prog.*, 146(1):219–244, 2014.
- [LY21] David G. Luenberger and Yinyu Ye. *Linear and Nonlinear Programming*. Springer, 2021.
- [LZ19] Zhaosong Lu and Zirui Zhou. Nonmonotone enhanced Proximal DC algorithms for a class of structured nonsmooth DC programming. *SIAM Journal on Optimization*, 29(4):2725–2752, 2019.
- [MA24] . MOSEK Aps. Mosek modeling cookbook v.3.3.0. <https://docs.mosek.com/MOSEKModelingCookbook-a4paper.pdf/>, 2024. [Online; accessed 29-June-2024].
- [MAB12] Pierre Machart, Sandrine Anthoine, and Luca Baldassarre. Optimal computational trade-off of inexact proximal methods. *arXiv:1210.5034*, 2012.
- [MF22] Taylor Mathieu, Barréand Adrien and Bach Francis. A note on approximate accelerated forward-backward methods with absolute and relative errors, and possibly strongly convex objectives. In *Open Journal of Mathematical Optimization*, volume 3, 2022.
- [MFU23] R Díaz Millán, Orizon P Ferreira, and Julien Ugon. Frank-Wolfe algorithm for DC optimization problem. *arXiv:2308.16444*, 2023.
- [MLP19] Emmanuel Moulay, Vincent Léchappé, and Franck Plestan. Properties of the sign gradient descent algorithms. *Inf. Sci.*, 492(C):29–39, aug 2019. doi:10.1016/j.ins.2019.04.012.
- [MOJ18] Aryan Mokhtari, Asuman Ozdaglar, and Ali Jadbabaie. Escaping saddle points in constrained optimization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS’18*, pages 3633–3643, Red Hook, NY, USA, 2018. Curran Associates Inc.
- [MS13] Renato D. C. Monteiro and B. F. Svaiter. An accelerated hybrid proximal extragradient method for convex optimization

- and its implications to second-order methods. *SIAM Journal on Optimization*, 23(2):1092–1125, 2013.
- [MT24] Naoki Marumo and Akiko Takeda. Parameter-free accelerated gradient descent for nonconvex minimization. *SIAM Journal on Optimization*, 34(2):2093–2120, 2024.
- [Nes83] Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$. 1983. URL: <https://api.semanticscholar.org/CorpusID:202149403>.
- [Nes05] Yurii Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103:127–152, 2005.
- [Nes13] Yuri Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1, Ser. B):125–161, 2013.
- [NF22] Yurii Nesterov and Mihai I. Florea. Gradient methods with memory. *Optimization Methods and Software*, 37(3):936–953, 2022.
- [Niu23] Yi-Shuai Niu. Accelerated DC algorithms for the asymmetric eigenvalue complementarity problem. *arXiv:2305.12076*, 2023.
- [NN94] Yurii Nesterov and Arkadii Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics, 1994.
- [NPD22] Varsha Nemade, Sunil Pathak, and Ashutosh Kumar Dubey. A systematic literature review of breast cancer diagnosis using machine intelligence techniques. *Archives of Computational Methods in Engineering*, 29(6):4401–4430, 2022.
- [NT08] Arkadi S. Nemirovski and Michael J. Todd. Interior-point methods for optimization. *Acta Numerica*, 17:191–234, 2008.
- [NTM17] Matthew Norton, Akiko Takeda, and Alexander Mafusalov. Optimistic robust optimization with applications to machine learning. *arXiv:1711.07511*, 2017.

- [OA13] C. Ong and L. An. Learning sparse classifiers with difference of convex functions algorithms. *Optimization Methods and Software*, 28(4):830–854, 2013.
- [Pok24] Sebastian Pokutta. The frank-wolfe algorithm: a short introduction. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 126(1):3–35, 2024.
- [Pol64] B.T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.
- [PS]22] Bastian Pötzl, Anton Schiela, and Patrick Jaap. Inexact proximal newton methods in hilbert spaces. In <https://spp1962.wias-berlin.de/preprints/192.pdf>, 2022. pre-print.
- [PSW22] Veronica Piccialli, Antonio M. Sudoso, and Angelika Wiegele. SOS-SDP: An exact solver for minimum sum-of-squares clustering. *INFORMS Journal on Computing*, 34(4):2144–2162, 2022.
- [PU23] Fahaar Mansoor Pirani and Firdevs Ulus. Algorithms for DC programming via polyhedral approximations of convex functions. *arXiv:2309.05487*, 2023.
- [RFV23] Marius Roland, Alexandre Forel, and Thibaut Vidal. Adaptive partitioning for chance-constrained problems with finite support. *arXiv:2312.13180*, 2023.
- [RGZ05] A. Rubinov R. Ghosh and J. Zhang. Optimization approach for clustering datasets with weights. *Optimization Methods and Software*, 20(2-3):335–351, 2005.
- [RJGF10] Jenatton Rodolphe, Mairal Julien, Obozinski Guillaume, and Bach Francis. Proximal methods for sparse hierarchical dictionary learning. In *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML)*, pages 487–494. Association for Computing Machinery, 2010.
- [Roc70] Ralph Tyrell Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, 1970.

- [RSU05] Alexander Rubinov, Nadejda Soukhoroukova, and Julien Ugon. *Minimization of the Sum of Minima of Convex Functions and Its Application to Clustering*, pages 409–434. Springer US, Boston, MA, 2005.
- [Rus02] Andrzej Ruszczyński. Probabilistic programming with discrete distributions and precedence constrained knapsack polyhedra. *Math. Prog.*, 93, 2002.
- [RW04] Ralph Tyrell Rockafellar and Roger J-B Wets. *Variational Analysis*. Number 317. Springer, 2 edition, 2004.
- [SBTBdH22] Aras Selvi, Aharon Ben-Tal, Ruud Brekelmans, and Dick den Hertog. Convex maximization via adjustable robust optimization. *INFORMS Journal on Computing*, 34(4):2091–2105, 2022.
- [SGKS20] Ali Siahkamari, Aditya Gangrade, Brian Kulis, and Venkatesh Saligrama. Piecewise linear regression via a difference of convex functions. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8895–8904. PMLR, 13–18 Jul 2020.
- [SMD16] A. Sinha, P. Malo, and K. Deb. Solving optimistic bilevel programs by iteratively approximating lower level optimal value function. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 1877–1884, 2016.
- [SO23] Pulak Swain and Akshay Kumar Ojha. Chance constrained robust portfolio optimization when the perturbations follow normal and exponential distributions. *arXiv:2311.04577*, 2023.
- [SRB11] Mark W. Schmidt, Nicolas Le Roux, and Francis R. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. In *NIPS*, 2011.
- [SSUT24] Kristian Sabo, Rudolf Scitovski, Šime Ungar, and Zoran Tomljanović. A method for searching for a globally optimal k -partition of higher-dimensional datasets. *Journal of Global Optimization*, 89:633–653, 2024.

- [ST04] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *J. ACM*, 51(3):385–463, may 2004.
- [STG⁺21] Fedor S. Stonyakin, Alexander Tyurin, Alexander V. Gasnikov, Pavel E. Dvurechensky, Artem Agafonov, Darina Dvinskikh, Mohammad S. Alkousa, Dmitry Pasechnyuk, Sergei Artamonov, and Victorya Piskunova. Inexact relative smoothness and strong convexity for optimization and variational inequalities by inexact model. *arXiv:2001.09013*, 2021.
- [Str23] Matthew Streeter. Universal majorization-minimization algorithms. *arXiv:2308.00190*, 2023.
- [SXZ22] Mengmeng Song, Yong Xia, and Jinyiang Zheng. On globally solving nonconvex trust region subproblem via projected gradient method. *arXiv:2207.05597*, 2022.
- [TFT22] S. Takahashi, M. Fukuda, and M. Tanaka. New Bregman proximal type algorithms for solving DC optimization problems. *Comput Optim Appl*, 83:893–931, 2022.
- [TFV23] Tuyen Tran, Kate Figenschou, and Phan-Tu Vuong. The Boosted DC algorithm for clustering with constraints. *arXiv:2310.14148*, 2023.
- [TG15] Francisco Trespalcios and Ignacio E. Grossmann. Improved big-M reformulation for generalized disjunctive programs. *Computers & Chemical Engineering*, 76:98–103, 2015.
- [THP20] Andreas Themelis, Ben Hermans, and Panagiotis Patrinos. A new envelope function for nonsmooth DC optimization. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 4697–4702, 2020.
- [TJNO20] Kiran Koshy Thekumparampil, Prateek Jain, Praneeth Netrapalli, and Sewoong Oh. Projection efficient subgradient method and optimal nonsmooth Frank-Wolfe method. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS’20*, Red Hook, NY, USA, 2020. Curran Associates Inc.

- [TOF24] . The Optimization Firm. Baron user manual v.2024.5.8. <https://minlp.com/downloads/docs/baron%20manual.pdf/>, 2024. [Online; accessed 29-June-2024].
- [TSN08] Matthew W. Tanner, Lisa Sattenspiel, and Lewis Ntaimo. Finding optimal vaccination strategies under parameter uncertainty using stochastic programming. *Mathematical Biosciences*, 215(2):144–151, 2008.
- [VAM19] Wim Van Ackooij and Jérôme Malick. Eventual convexity of probability constraints with elliptical distributions. *Math. Prog.*, 175(1):1–27, 2019.
- [Van22] L. Vandenbergh. Accelerated proximal gradient methods. <https://www.seas.ucla.edu/~vandenbe/236C/lectures/fgrad.pdf>, 2022.
- [VSBV13] Silvia Villa, Saverio Salzo, Luca Baldassarre, and Alessandro Verri. Accelerated and inexact forward-backward algorithms. In *SIAM Journal on Optimization*, volume 23(3), pages 1607–1633, 2013.
- [WCP18] B. Wen, X. Chen, and T.K. Pong. A proximal difference-of-convex algorithm with extrapolation. *Comput. Optim. Appl.*, 69:297–324, 2018.
- [WDHB16] Matt Wytock, Steven Diamond, Felix Heide, and Stephen Boyd. A new architecture for optimization modeling frameworks. In *2016 6th Workshop on Python for High-Performance and Scientific Computing (PyHPC)*, pages 36–44, 2016.
- [Won11] Elizabeth Wong. *Active-Set Methods for Quadratic Programming*. Doctoral Thesis, University of California, 2011.
- [WRDC21] Kong Weiwei and Monteiro Renato D. C. An accelerated inexact dampened augmented lagrangian method for linearly-constrained nonconvex composite optimization problems. 2021. pre-print.
- [Xu21] Y. Xu. Iteration complexity of inexact augmented lagrangian methods for constrained convex programming. In *Mathematical Programming*, volume 185, pages 199–244, 2021.

- [Yan24] L. Yang. Proximal gradient method with extrapolation and line search for a class of non-convex and non-smooth problems. *Journal of Optim. Theory Appl.*, 200:68–103, 2024.
- [YN23] Yu You and Yi-Shuai Niu. A variable metric and nesterov extrapolated proximal DCA with backtracking for a composite DC program. *Journal of Industrial and Management Optimization*, 19(10):7716–7734, 2023.
- [YT21] Lei Yang and Kim-Chuan Toh. An inexact Bregman proximal gradient method and its inertial variant. *arXiv:2109.05690*, September 2021.
- [Yua23] Ganzhao Yuan. Coordinate descent methods for DC minimization: optimality conditions and global convergence. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*. AAAI Press, 2023.
- [ZJ23] X. Zuo and Y. Jiang. Solution methodologies for minimizing a sum of pointwise minima of two functions. *Optimization Letters*, 17:75–87, 2023.
- [ZYG⁺21] Siqi Zhang, Junchi Yang, Cristóbal Guzmán, Negar Kiyavash, and Niao He. The complexity of nonconvex-strongly-concave minimax optimization. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 482–492. PMLR, 27–30 Jul 2021.