UNIVERSITÀ
DEGLI STUDI
FIRENZE

PhD Program in Smart Computing
Dipartimento di Ingegneria dell'Informazione (DINFO)

Algorithms for ℓ_0-norm Optimization Problems

Tommaso Levato

Dissertation presented in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Smart Computing

PhD Program in Smart Computing
University of Florence, University of Pisa, University of Siena

Algorithms for ℓ_0-norm Optimization Problems

Tommaso Levato

Advisors:

Prof. Marco Sciandrone

Prof. Fabio Schoen

Head of the PhD Program:

Prof. Paolo Frasconi

Evaluation Committee:
Prof. Manlio Gaudioso, *Università della Calabria*
Prof. Fabio Tardella, *Sapienza Università di Roma*

XXXII ciclo — October 2019

A Mamma e Babbo

Acknowledgments

First, I would like to thank Professors Marco Sciandrone and Fabio Schoen: you have been way better advisors than what I have been as a PhD student. I also would like to thank Giampaolo Liuzzi and Francesco Rinaldi, as coauthors and supervisors, for their support and advice.

Finally, thanks to my friends and colleagues of the Global Optimization Laboratory (LabGOL), who have shared this journey with me: Francesco, Enrico, Guido, Leonardo D.G., Leonardo G., Alessandro, Giulio, Matteo, Federica, Roberta, Alessio, Luca. I am fortunate to have had you by my side.

Abstract

This thesis concerns ℓ_0-norm optimization problems, where we are interested in finding solutions with few nonzero components. This characteristic is of interest in a number of different fields, such as portfolio optimization, signal processing and machine learning. In this thesis, we propose three methods for the solution of ℓ_0-norm optimization problems, satisfying suitable convergence guarantees.

In the first part, we propose a mixed-integer approach for dealing with cardinality constrained optimization problems, under additional convex constraints and with a nonconvex objective function. We state theoretical convergence results stronger than those recently presented in the literature. We also compare the numerical performance of our approach with two state-of-the-art algorithms, and the results obtained show that our method is competitive.

The second contribution is an inexact version of a popular penalty decomposition algorithm known in literature. We prove that this inexact variant satisfies the same convergence guarantees as the exact algorithm, which makes our method suitable to deal with nonconvex objective functions. We also provide a derivative-free extension for the case where first order information are not available, and we show that this adaptation maintains the same convergence guarantees as the original smooth version.

Finally, we propose an optimization method for sparse multiobjective problems where one of the objectives is the ℓ_0-norm. The approach is based on concave approximations of the ℓ_0-norm, and the obtained approximated problem is shown to be equivalent, in some sense, to the original one. We propose an algorithm based on line searches along suitable descent directions to provide a good approximation of the Pareto front of the original multiobjective problem. We show that our method is competitive against different state-of-the-art multiobjective algorithms, in both the single point and Pareto front cases.

Contents

Chapter 1

Introduction

Sparse solutions to optimization problems are desirable for different reasons: they are more robust to noise, they are simpler to realize and, most importantly, they are simpler to understand and to explain. Parsimonious models are, therefore, desirable in various practical settings: portfolio optimization (Deng et al. (2012); Mutunge and Haugland (2018)), compressed sensing (Eldar and Kutyniok (2012)), machine learning (Miller (2002); John et al. (1994)) and so on. However, this feat is hard to obtain: in general, retrieving the sparsest solution to an optimization problem is NP-Hard (Natarajan (1995)).

Essentially, sparsity can be either imposed or encouraged. In other words, we can either constrain our solutions to be sparse, or we can express a preference towards sparse solutions. More formally, given a generic feasible set $X \subseteq \mathbb{R}^n$, we consider two different formulations: the first is the cardinality constrained problem

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & x \in X, \\ & \|x\|_0 \leq s, \end{aligned} \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function, $0 < s < n$ and $\|\cdot\|_0$ is the so-called "ℓ_0-norm", i.e., the cardinality of the support of x:

$$\|x\|_0 = |\{i \mid x_i \neq 0\}|. \tag{1.2}$$

Here, each feasible point cannot exceed a predefined cardinality, specified by the parameter s. The second formulation is the ℓ_0-penalty problem

$$\begin{aligned} \min_x \quad & f(x) + \lambda\|x\|_0 \\ \text{s.t.} \quad & x \in X, \end{aligned} \tag{1.3}$$

where $\lambda > 0$ is a fixed parameter. Here, instead, sparsity of the solution is not guaranteed.

These two problems are strictly related, though not equivalent (Nikolova (2016); Soubies et al. (2017)). The choice between these two variants is not trivial and offers different solution strategies, which have to be carefully considered for the application at hand.

A common thread, or threat, runs through the previous formulations: the ℓ_0-norm is not continuous, let alone differentiable or convex. Standard optimization techniques are therefore generally inapplicable in this context. In general, solution strategies can be roughly categorized in two kinds of approaches: build new optimization methods specifically designed to work with the ℓ_0-norm, or recast the problem at hand into a problem where standard optimization techniques are applicable, trying to ensure that the resulting optimization program is somewhat equivalent to the original one. In reference to the relevant literature of the field, we have a variety of different classes of related approaches, comprising (but not limited to):

- convex relaxations (Tibshirani (1996); Efron et al. (2004); Zou and Hastie (2005)),
- global optimization approaches (Bertsimas and Shioda (2009); Bertsimas et al. (2016); Bienstock (1996); Boudt and Wan (2019)),
- decomposition methods (Bai et al. (2016); Di Gangi et al. (2019); Lu and Zhang (2013)),
- concave approximations (Rinaldi et al. (2010); Di Lorenzo et al. (2012)),
- iterative thresholding (Blumensath and Davies (2008); Xu et al. (2012)),
- multiobjective approaches (Anagnostopoulos and Mamanis (2010); Brito and Vicente (2014)),

and so on.

The line between the two main kinds of approaches described earlier can be blurred at times, but there are a number of methods that strongly fit this categorization. For example, the Lasso approach (Tibshirani (1996)) is entirely based on the use of the convex ℓ_1-norm in place of the ℓ_0-norm, while the iterative thresholding algorithms are instead designed taking into account the specific characteristics of the ℓ_0-norm.

Summing up, there are a lot of different ways to tackle the daunting task of minimizing ℓ_0-norm optimization problems. In this regard, this thesis proposed three novel methods for the resolution of ℓ_0-norm optimization problems. In our view, the methods we propose fit into the "reformulation" category, meaning that we try to bridge the gap between the inherently nonsmooth nature of the ℓ_0-norm term with standard smooth optimization techniques. The reason is simple: this choice opens up a lot of opportunities in the way we approach the problem and in the solution

strategies we can pick from, and this results in a huge amount of research directions that can be followed.

Another important thing to note is the fact that our methods do not require the convexity of the objective function, unlike some important state-of-the-art methods. In fact, although in the literature the convexity assumption is not always explicitly stated, it is often required in practice, typically through exact minimizations of subproblems in an iterative scheme (see, e.g., Beck and Eldar (2013); Lu and Zhang (2013)). This observation, moreover, does not only hold in practical settings, but also in the convergence analysis, which assumes that global minima can be found at each iteration. In case of a nonconvex objective, this can be prohibitive. The methods we propose, instead, only rely on the differentiability of the objective function, without any convexity assumptions. This feat is achieved by replacing exact minimizations with line search procedures along suitable descent directions, typically related to the gradient of the objective function.

We now briefly describe the contributions of this thesis. In Chapter 2, taking inspiration from Burdakov et al. (2016), we first recast problem (1.1) into a mixed integer formulation, where added binary variables are used to express implicitly the original cardinality constraint. Essentially, each binary variable indicates whether the related original variable is constrained to be 0 or not. Then, we apply an optimization strategy based on local searches and the exploration of different "discrete neighborhoods" of the current iterate, following the strategy proposed in Lucidi et al. (2005) for general mixed integer problems. Informally, at each iteration, the idea is to first look for a set of promising variables, fixing the others at 0, and then to improve the chosen free variables by using standard local search techniques. In this way, we separate the task of finding a small subset of variables from the true optimization phase, which in principle can be carried out with any standard optimization technique.

In Chapter 3 we propose a penalty decomposition method for problem (1.3). Following the approach in Lu and Zhang (2013), the idea is to duplicate the variables of the problem to split the constraints between the original and the new variables. In particular, the cardinality constraint is assigned to the added variables, while the original variables are free to move. As in the previous work, the idea is to separate the two tasks of choosing a small subset of variables and of optimizing the selected ones. This is done by employing an alternate minimization scheme between the two sets of variables. To ensure feasibility of the limit points, the distance between the original and the new variables is penalized in the objective function, constructing a series of optimization problems where this penalty gets increasingly bigger.

Lastly, considering problem (1.3), it can be argued that the nature of this problem is inherently multiobjective: we have to deal with the (usually) conflicting goals of reducing some kind of cost, embodied by $f(x)$, trying to do so with as little com-

ponents as possible. Roughly speaking, the λ parameter encodes a priority between the two objectives, resulting in different solutions for different values of λ. In other words, we can argue that problem (1.3) can be well represented by the multiobjective program

$$\begin{aligned} \min_{x} \quad & f(x), ||x||_0 \\ s.t. \quad & x \in X \end{aligned}$$

or, more generally, by the multiobjective problem

$$\begin{aligned} \min_{x} \quad & f_1(x), f_2(x), \ldots, f_{m-1}(x), ||x||_0 \\ s.t. \quad & x \in X, \end{aligned} \tag{1.4}$$

for some continuously differentiable functions $f_1(x), \ldots, f_{m-1}(x) : \mathbb{R}^n \to \mathbb{R}$. In Chapter 4 we propose to reformulate problem (1.4) by replacing the ℓ_0-norm objective with a smooth concave approximation function, as proposed in Rinaldi et al. (2010) for single objective problems. This allows the use of gradient-descent related algorithms as a solution strategy. The approximated problem obtained has been shown to be equivalent, in some sense, to the original multiobjective problem. To compute an approximation of the Pareto front, we propose an algorithm that iteratively improves the current list of non-dominated points using a set of descent directions, as in Fliege and Vaz (2016).

Chapter 2

A mixed-integer algorithm for nonconvex problems with cardinality constraint

In this chapter, we consider the following cardinality constrained problem:

$$\begin{array}{ll} \min\limits_{x} & f(x) \\ \text{s.t.} & \|x\|_0 \leq s, \\ & x \in X, \end{array} \tag{2.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function, nonconvex in general, and $X \subseteq \mathbb{R}^n$ is a closed convex set. As shown in, e.g., Burdakov et al. (2016), problem (2.1) can be stated as a mixed integer program equivalent, in a certain sense, to the original one. Essentially, we insert new binary variables $y \in \{0,1\}^n$ to express which components x_i must be equal to zero, ensuring the satisfaction of the cardinality constraint.

Here, we tackle the mixed integer reformulation by applying a resolution strategy adapted from the MIVAM approach defined in Lucidi et al. (2005). The idea is to explore different "discrete neighborhoods" of the current solution, which are configurations of the integer variables defining the chosen components that we can set different than zero. The associated continuous variables are then optimized with a standard line search technique.

Another work that guided us towards the choice of a mixed integer approach is the Greedy Sparse-Simplex (GSS) approach proposed in Beck and Eldar (2013). The GSS method iteratively refines the current point by optimally moving a single component. In our view, the iterative exact minimization of each single component is a combinatorial approach in disguise, strongly related to the definition of our discrete neighborhoods.

Our neighborhood, in fact, consists of points obtained by swapping pairs of components: one currently set to zero, one currently different from zero. This explicit combinatorial approach is, in principle, relatively similar to the GSS approach, while it does not require exact minimizations. Moreover, we also adapt our approach to the case where the convex set X is a strict subset of $\mathbb{R}^n$, which GSS does not treat out of the box and has not yet been theoretically analyzed.

2.1 Preliminaries

In this section we give a brief summary of the two approaches that inspired our work. First, we consider the Greedy Sparse-Simplex method proposed in Beck and Eldar (2013), suited for the resolution of problem (2.1) in the case where $X = \mathbb{R}^n$, i.e. when the only constraint is the cardinality constraint. Then, we summarize the Regularization method proposed in Burdakov et al. (2016), which can be applied when X is a general subset of $\mathbb{R}^n$. In what follows, the support of x is defined by

$$I_1(x) = \{i \mid x_i \neq 0\}$$

while its complement is

$$I_0(x) = \{i \mid x_i = 0\}.$$

We denote by x_I the subvector of x identified by the components contained in an index set I.

Greedy Sparse-Simplex Method

The Greedy Sparse-Simplex method (GSS) is an iterative method that, at each iteration, tries to improve the current point moving a single component. Depending on the cardinality of the current point x^k, two things can happen:

- if $\|x^k\|_0 < s$, the next iterate is the best point among the n different points x^i obtained by moving in an optimal way each single component while keeping fixed the others:

$$x^i = x^k + t^i e_i \text{ where } t_i = \arg\min_t f(x^k + te_i);$$

- if $\|x^k\|_0 = s$, the next iterate is the best point among the $n \cdot s$ different points $x^{i,j}$ defined as:

$$x^{i,j} = x^k - x_i^k e_i + t^j e_j \text{ where } t^j = \arg\min_t f(x^k - x_i^k e_i + te_j),$$

where $i \in I_1(x^k)$ and $j = 1, \ldots, n$.

The method stops when the current iterate is kept fixed after an iteration. It can be shown that the method converges to a *CW-minimum* (CW = component-wise), defined as follows.

Definition 2.1.1 (CW-minimum)**.** A vector x^* is a CW-minimum if:

$\|x^*\|_0 < s$ and

$$f(x^*) = \min_{t \in \mathbb{R}} f(x^* + te_i) \quad \forall i = 1, \ldots, n;$$

or

$\|x^*\|_0 = s$ and

$$f(x^*) \leq \min_{t \in \mathbb{R}} f(x^* - x_i^* e_i + te_j) \quad \forall i \in I_1(x^*), \forall j = 1, \ldots, n.$$

This algorithm is simple, intuitive, and its theoretical analysis is quite elegant. Moreover, the notion of CW-minimum is an optimality condition which is quite strong and usually identifies quality points. However, the need for exact minimizations makes the method impractical when the objective function is nonconvex. Also, as already stated, the method cannot be employed when $X \subset \mathbb{R}^n$.

Regularization Method

Problem (2.1) can be reformulated into the following mixed-integer program:

$$\begin{array}{lll} \min\limits_{x,y} & f(x) & \\ \text{s.t.} & e^\top y \geq n - s, & \\ & x_i y_i = 0, & \forall i = 1, \ldots, n, \\ & x \in X, & \\ & y \in \{0,1\}^n. & \end{array} \tag{2.2}$$

As proposed by Burdakov et al. (2016), we could relax the mixed-integer program 2.2 into the following smooth problem:

$$\begin{array}{lll} \min\limits_{x,y} & f(x) & \\ \text{s.t.} & e^\top y \geq n - s, & \\ & x_i y_i = 0, & \forall i = 1, \ldots, n, \\ & x \in X, & \\ & 0 \leq y_i \leq 1, & \forall i = 1, \ldots, n. \end{array} \tag{2.3}$$

It can be shown (ref. Burdakov et al. (2016)) that, if X is polyhedral convex, then the feasible set of problem 2.3 satisfies a suitable constraint qualification. In other words, in this case, KKT stationarity is indeed a necessary optimality condition. In general,

however, the feasible set of problem 2.3 may violate each standard constraint qualification.

In any case, the authors base their discussion and their method on the following definitions of stationarity. In the sequel, for notational purposes, we will sometimes replace the functions defining X with functions $h : \mathbb{R}^n \to \mathbb{R}^m$ and $g : \mathbb{R}^n \to \mathbb{R}^p$.

Definition 2.1.2. Let (x^*, y^*) be feasible for the relaxed problem 2.3. Then (x^*, y^*) is called

(a) *S-stationary* (S = strong) if there exist multipliers $\lambda \in \mathbb{R}^m$, $\mu \in \mathbb{R}^p$ and $\gamma \in \mathbb{R}^n$ such that the following conditions hold:

$$\begin{aligned} \nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*) + \sum_{i=1}^{p} \mu_i \nabla h_i(x^*) + \sum_{i=1}^{n} \gamma_i e_i = 0, \\ \lambda_i \geq 0,\ \lambda_i g_i(x^*) = 0,\ \forall i = 1, \ldots, m, \\ \gamma_i = 0,\ \forall i \text{ such that } y_i^* = 0. \end{aligned} \tag{2.4}$$

(b) *M-stationary* (M = Mordukhovich) if there exist multipliers $\lambda \in \mathbb{R}^m, \mu \in \mathbb{R}^p$ and $\gamma \in \mathbb{R}^n$ such that the following conditions hold:

$$\begin{aligned} \nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*) + \sum_{i=1}^{p} \mu_i \nabla h_i(x^*) + \sum_{i=1}^{n} \gamma_i e_i = 0, \\ \lambda_i \geq 0,\ \lambda_i g_i(x^*) = 0,\ \forall i = 1, \ldots, m, \\ \gamma_i = 0,\ \forall i \text{ such that } x_i^* \neq 0. \end{aligned} \tag{2.5}$$

Note that M-stationarity is a weaker condition than S-stationarity, as it does not impose conditions on the components for which both x_i^* and y_i^* are equal to 0.

The following proposition, proven in Burdakov et al. (2016), shows that S-stationarity is equivalent to KKT stationarity. As a consequence, M-stationarity is a weaker concept than KKT stationarity.

Proposition 2.1.1. *Let (x^*, y^*) be feasible for the relaxed problem 2.3. Then (x^*, y^*) is a usual KKT point if and only if (x^*, y^*) is an S-stationary point.*

M-stationarity, instead, can be understood realizing that x^* is M-stationary if and only if it is a KKT point for the problem

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & x \in X, \\ & x_i = 0, \quad \forall i : x_i^* = 0. \end{aligned} \tag{2.6}$$

Note that, while S-stationarity depends on y^*, M-stationarity is a condition that only depends on the original variable x^*. In other words, a "wrong" vector y^* can destroy S-stationarity, while an M-stationary point x^* remains M-stationary independently of the vector y^* that is associated to it, as long as (x^*, y^*) is feasible. Moreover,

even though M-stationarity is a condition that is relatively weak, it is a necessary optimality condition under mild regularity conditions on the feasible set X, usually satisfied in practice.

Our interest in the concept of S-stationarity and M-stationarity is twofold:

1. the regularization method proposed by Burdakov et al. (2016) converges to an M-stationary point;

2. S-stationarity is implied by the concept of stationarity proposed in Lucidi et al. (2005), as we will show in the next section.

2.2 The Mixed-Integer Approach

Following Lucidi and Piccialli (2004); Lucidi et al. (2005), a different way to characterize problem (2.2) is by defining notions of local minima and stationary points suited for mixed-integer problems. From here onwards, we will use the following notation:

$$\begin{aligned} \mathcal{Y} &= \{y \mid y \in \{0,1\}^n, e^\top y \geq n - s\}, \\ \mathcal{X}(y) &= \{x \in X \mid x_i y_i = 0 \; \forall i = 1, \ldots, n\}. \end{aligned} \tag{2.7}$$

The definitions of local minimizer and stationary point are based on a suitable definition of neighborhood of a feasible point, which has to take into account the mixed-integer nature of the problem. For the moment, we refer to this notion of "discrete neighborhood" with the notation $\mathcal{N}(x^*, y^*)$, that will be formally defined in the next sections.

Definition 2.2.1. A feasible point (x^*, y^*) is said to be a *local minimizer* of problem (2.2) with respect to the feasible discrete neighborhood $\mathcal{N}(x^*, y^*)$ if there exists an $\epsilon > 0$ such that for all $(\hat{x}, \hat{y}) \in \mathcal{N}(x^*, y^*)$

$$f(x^*) \leq f(x) \quad \forall x \in \mathcal{B}(\hat{x}, \epsilon) \cap \mathcal{X}(\hat{y}), \tag{2.8}$$

where $\mathcal{N}(x^*, y^*)$ is a finite set of feasible points.

We refer to the notion of stationarity we employ within this chapter as MIVAM-stationarity, formally defined next.

Definition 2.2.2. A feasible point (x^*, y^*) is said to be a *MIVAM-stationary point* of problem (2.2) with respect to the feasible discrete neighborhood $\mathcal{N}(x^*, y^*)$ if

(i) the point x^* is a stationary point of the continuous problem

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in \mathcal{X}(y^*); \end{aligned} \tag{2.9}$$

(ii) every $(\hat{x}, \hat{y}) \in \mathcal{N}(x^*, y^*)$ satisfies $f(\hat{x}) \geq f(x^*)$;

(iii) for every $(\hat{x}, \hat{y}) \in \mathcal{N}(x^*, y^*)$ such that $f(\hat{x}) = f(x^*)$, the point $\hat{x}$ is a stationary point of the continuous problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in \mathcal{X}(\hat{y}). \end{array} \tag{2.10}$$

Now, points (i) and (iii) of MIVAM-stationarity depend on an underlying notion of stationarity that has to be specified. Here, we concentrate on KKT stationarity. Under these premises, it is fairly easy to see that MIVAM-stationarity implies S-stationarity.

In fact, we now show a couple of results that clarify the relationship between the different optimality conditions we have defined, namely, CW-minima (Definition 2.1.1), S-stationarity (Definition 2.1.2) and MIVAM-stationarity.

Relationships between optimality conditions

We start by proving that MIVAM-stationarity is a stronger optimality condition than S-stationarity.

Proposition 2.2.1. *Let (x^*, y^*) be a MIVAM-stationary point of problem 2.2. Then, it is an S-stationary point.*

Proof. If (x^*, y^*) is MIVAM-stationary, in particular we must have (point (i)) that x^* is stationary w.r.t. the following problem:

$$\begin{array}{lll} \min\limits_{x} & f(x) & \\ \text{s.t.} & h_i(x) = 0, & \forall i = 1, \ldots, m, \\ & g_i(x) \leq 0, & \forall i = 1, \ldots, p, \\ & x_i y_i^* = 0, & \forall i = 1, \ldots, n. \end{array} \tag{2.11}$$

Rearranging, the previous problem can be rewritten as

$$\begin{array}{lll} \min\limits_{x} & f(x) & \\ \text{s.t.} & h_i(x) = 0, & \forall i = 1, \ldots, m, \\ & g_i(x) \leq 0, & \forall i = 1, \ldots, p, \\ & x_i = 0, & \forall i : y_i^* = 1. \end{array} \tag{2.12}$$

This means that there exist multipliers $\lambda \in \mathbb{R}^m$, $\mu \in \mathbb{R}^p$ and $\gamma \in \mathbb{R}^n$ such that the following conditions hold:

$$\begin{aligned} \nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*) + \sum_{i=1}^{p} \mu_i \nabla h_i(x^*) + \sum_{\{i | y_i^* = 1\}} \gamma_i e_i = 0, \\ \lambda_i \geq 0,\ \lambda_i g_i(x^*) = 0,\ \forall i = 1, \ldots, m. \end{aligned} \tag{2.13}$$

That is:

$$\begin{aligned} \nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*) + \sum_{i=1}^{p} \mu_i \nabla h_i(x^*) + \sum_{i=1}^{n} \gamma_i e_i = 0, \\ \lambda_i \geq 0,\ \lambda_i g_i(x^*) = 0,\ \forall i = 1, \dots, m, \\ \gamma_i = 0,\ \forall i \text{ such that } y_i^* = 0. \end{aligned} \tag{2.14}$$

Therefore, (x^*, y^*) is an S-stationary point. □

CW-minima, instead, satisfy a stronger optimality condition than MIVAM-stationarity, as shown in the following example.

Example 2.2.1. *Consider the following problem:*

$$\begin{aligned} \min_{x} \quad & f(x) = \tfrac{1}{4}x_1^4 + \tfrac{2}{3}x_1^3 - \tfrac{3}{2}x_1^2 \\ s.t. \quad & \|x\|_0 \leq 1, \end{aligned}$$

where $x \in \mathbb{R}^2$. The objective $f(x_1)$, reduced for simplicity to the only meaningful component x_1, has a local minimum in $\bar{x}_1 = 1$ and a global minimum in $x_1^* = -3$, with $f(\bar{x}_1) = -\frac{7}{12}$ and $f(x_1^*) = -\frac{45}{4}$.

Letting $\bar{x} = [\bar{x}_1, 0]$ and $\bar{y} = [0, 1]$, we have that $(\bar{x}, \bar{y})$ is a stationary point of

$$\begin{aligned} \min_{x,y} \quad & f(x) = \tfrac{1}{4}x_1^4 + \tfrac{2}{3}x_1^3 - \tfrac{3}{2}x_1^2 \\ s.t. \quad & y_1 + y_2 \geq 1, \\ & x_1 y_1 = 0, \\ & x_2 y_2 = 0, \\ & y_1, y_2 \in \{0, 1\}. \end{aligned}$$

In fact, since $\bar{x}_1$ is a local minimum for $f(x_1)$, point (i) of Definition 2.2.2 is automatically satisfied. Moreover, every $(x, y) \in \mathcal{N}(\bar{x}, \bar{y})$ is such that $x_1 = 0$, which means that $f(x) = 0 > f(\bar{x})$, so also point (ii) of Definition 2.2.2 is satisfied (and point (iii) is irrelevant).

However, we have that $f(\bar{x}) = -\frac{7}{12} > \min_{t \in \mathbb{R}} f(\bar{x} - \bar{x}_1 e_1 + t e_1) = f(x_1^*) = -45/4$. Therefore, $\bar{x}$ is not a CW-minimum.

In the case where $X = \mathbb{R}^n$, another optimality condition is given by the concept of BF-point.

Definition 2.2.3 (BF-point)**.** A point x^* is a BF-point if:

- $\|x^*\|_0 < s$ implies that $\nabla_i f(x^*) = 0$ for all $i = 1, \dots, n$;
- $\|x^*\|_0 = s$ implies that $\nabla_i f(x^*) = 0$ for all $i \in I_1(x^*)$.

It is quite easy to realize that the CW-minimum condition is stronger than the concept of BF-point, as shown in Beck and Eldar (2013). It also holds that MIVAM-stationary points are automatically BF-points, as we prove in the following proposition.

Proposition 2.2.2. *Let (x^*, y^*) be a MIVAM-stationary point. Then, x^* is a BF-point.*

Proof. There are two cases:

1. $\|x^*\|_0 = s$. By point (i) of Definition 2.2.2, there exist multipliers $\gamma \in \mathbb{R}^n$ such that
$$\nabla f(x^*) + \sum_{\{i|y_i^*=1\}} \gamma_i e_i = 0.$$
Therefore, we must have that $\nabla_i f(x^*) = 0$ for all $i \in \{i \mid y_i^* = 0\} \supseteq I_1(x^*)$.

2. $\|x^*\|_0 < s$. As in the previous case, we get that $\nabla_i f(x^*) = 0$ for all $i \in \{i \mid y_i^* = 0\}$. Consider then an index j such that $y_j^* = 1$, which implies that $j \in I_0(x^*)$. By definition of $\mathcal{N}(x^*, y^*)$, there exists a point $(\hat{x}, \hat{y}) \in \mathcal{N}(x^*, y^*)$ such that $\hat{x} = x^*$ and $\hat{y}_j = 0$. Then, by point (iii) of Definition 2.2.2, we must have that $\nabla_j f(\hat{x}) = \nabla_j f(x^*) = 0$. In other words, $\nabla_i f(x^*) = 0$ for all $i = 1, \ldots, n$.

□

The BF optimality condition is, in fact, weaker than MIVAM-stationarity, as shown in the next example.

Example 2.2.2. *Consider the following problem:*

$$\begin{array}{ll} \min\limits_{x} & (x_1 - 2)^4 + (x_2 - 1)^2 \\ s.t. & \|x\|_0 \leq 1, \end{array}$$

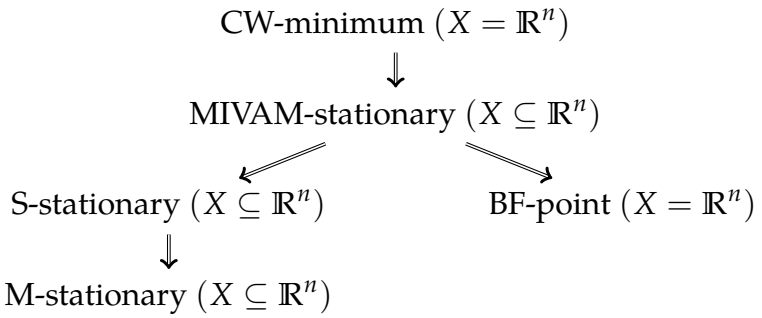
and let $x^* = [0, 1]^\top$. It is easy to see that x^* is a BF-point. Moreover, let $y^* = [1, 0]^\top$. Note that (x^*, y^*) is feasible for the mixed integer reformulation

$$\begin{array}{ll} \min\limits_{x,y} & (x_1 - 2)^4 + (x_2 - 1)^2 \\ s.t. & y_1 + y_2 \geq 1, \\ & x_1 y_1 = 0, \\ & x_2 y_2 = 0, \\ & y_1, y_2 \in \{0, 1\} \end{array}$$

and also note that there does not exist a vector $y \neq y^*$ such that $y \in \mathcal{Y}, x^* \in \mathcal{X}(y)$. In other words, y^* is the only possible vector such that the pair (x^*, y^*) is feasible for the mixed integer reformulation.

Let $\hat{x} = [1, 0]^\top$ and $\hat{y} = [0, 1]^\top$. By definition, $(\hat{x}, \hat{y}) \in \mathcal{N}(x^*, y^*)$. Moreover, $f(\hat{x}) = 2 < f(x^*) = 16$. Therefore, point (ii) of Definition 2.2.2 is not satisfied and so (x^*, y^*) is not a stationary point.

To summarize, we have the following chains of implications:



2.3 Discrete neighborhoods

The crucial point in the mixed-integer approach is choosing suitable discrete neighborhoods. Before we define a concrete discrete neighborhood, however, we tweak the usual notion of convergence to a point, taking into account the discrete nature of the problem. In particular, we have the following definition.

Definition 2.3.1. A sequence $\{(x^k, y^k)\}$ converges to a point $(\bar{x}, \bar{y})$ if for any $\epsilon > 0$ there exists an index k_ϵ such that for all $k \geq k_\epsilon$ we have that $y^k = \bar{y}$ and $\|x^k - \bar{x}\| < \epsilon$.

We now give a formal definition of discrete neighborhood of a feasible point, denoted by $\mathcal{N}(x, y)$.

Definition 2.3.2. Let (x, y) be a feasible point for problem (2.2). A discrete neighborhood $\mathcal{N}(x, y)$ is a finite set of feasible points different from (x, y).

We also state the following "continuity" assumption on the discrete neighborhoods we explore, equivalent to the lower semicontinuity of a point-to-set function as defined in Berge (1963).

Assumption 2.3.1. *Let $\{(x^k, y^k)\}$ be a sequence converging to $(\bar{x}, \bar{y})$. Then, for any $(\hat{x}, \hat{y}) \in \mathcal{N}(\bar{x}, \bar{y})$, there exists a sequence $\{(\hat{x}^k, \hat{y}^k)\}$ converging to $(\hat{x}, \hat{y})$ such that $(\hat{x}^k, \hat{y}^k) \in \mathcal{N}(x^k, y^k)$.*

The assumption is, in its essence and its consequences, a definition of the possible discrete neighborhoods. To keep matters concrete, we give an example of a possible discrete neighborhood.

Example 2.3.1. *A $n \times n$ permutation matrix is a square matrix obtained from the $n \times n$ identity matrix by a permutation of rows. Let H be a permutation matrix obtained, for instance, by interchanging two rows, say i and j. The point*

$$\hat{x} = Hx$$

is such that

$$\hat{x}_h = x_h \; h \neq i, j \qquad\qquad \hat{x}_i = x_j, \;\; \hat{x}_j = x_i,$$

so that $\|\hat{x}\|_0 = \|x\|_0$. *Let* $\Gamma = \{H_1, H_2, \ldots, H_p\}$ *be a given set of permutation matrices obtained by interchanging two rows. Note that the maximum cardinality* p *of* Γ *is* $\frac{n(n-1)}{2}$. *Given a feasible point* (x, y), *i.e. point such that* $y \in \mathcal{Y}$ *and* $x \in \mathcal{X}(y)$, *we define* $\mathcal{N}(x, y)$ *as follows:*

$$\mathcal{N}(x, y) = \{(\hat{x}^l, \hat{y}^l) : \; \hat{x}^l = H_l x, \hat{y}^l = H_l y, \; l = 1, \ldots, p\},$$

i.e., $\mathcal{N}(x, y)$ *is obtained by swapping pairs of variables* (*both continuous and binary*). *Note that the swap produces the same point* (x, y) *in the case that it is performed by two null components of* x (*corresponding to two components of* y *equal to 1*).

Now, a discrete neighborhood, by definition, is a set of feasible points. In the case where $X \subset \mathbb{R}^n$, swapping variables, as done in Example 2.3.1, may result in points that are not feasible. For this reason, we initially consider an easier version of problem (2.2) where $X = \mathbb{R}^n$.

Proposition 2.3.1. *The point-to-set map* $\mathcal{N}(x, y)$ *defined in Example 2.3.1 satisfies assumption 2.3.1.*

Proof. If $\{(x^k, y^k)\}$ converges to $(\bar{x}, \bar{y})$, then for any $\epsilon > 0$ there exists an index k_ϵ such that for all $k \geq k_\epsilon$ we have that $y^k = \bar{y}$ and $\|x^k - \bar{x}\| < \epsilon$. Let $(\hat{x}, \hat{y}) \in \mathcal{N}(\bar{x}, \bar{y})$, i.e., for some $l \in \{1, \ldots, p\}$ we have

$$\hat{x} = H_l \bar{x}, \qquad \hat{y} = H_l \bar{y}.$$

Let $\{(\hat{x}^k, \hat{y}^k)\}$ be the sequence such that $(\hat{x}^k, \hat{y}^k) = (H_l x^k, H_l y^k)$ for all k. Note that $(\hat{x}^k, \hat{y}^k) \in \mathcal{N}(x^k, y^k)$ for all k since $H_l \in \Gamma$.

For k sufficiently large we have $y^k = \bar{y}$. This implies that $\hat{y}^k = H_l y^k = H_l \bar{y} = \hat{y}$. Moreover we can write

$$\lim_{k \to \infty} \hat{x}^k = \lim_{k \to \infty} H_l x^k = H_l \bar{x} = \hat{x},$$

and hence we may conclude that $(\hat{x}^k, \hat{y}^k) \in \mathcal{N}(x^k, y^k)$ and $\{(\hat{x}^k, \hat{y}^k)\}$ converges to $(\hat{x}, \hat{y})$. □

To generalize the previous proposition to the case where $X \subset \mathbb{R}^n$, we replace each $(\tilde{x}, \tilde{y}) \in N(x, y)$ with the point $(\hat{x}, \hat{y})$, where $\hat{y} = \tilde{y}$ and $\hat{x} = \Pi_{\mathcal{X}(\hat{y})}(\tilde{x})$. In other words, first we "permute" (x, y) as in Example 2.3.1, then we project the x part onto X with the projection operator $\Pi_{\mathcal{X}(\hat{y})}(\cdot)$. In the following, we will refer to this new discrete neighborhood as $\mathcal{N}_C(x, y)$.

Proposition 2.3.2. *Let $\{(x^k, y^k)\}$ be a sequence converging to $(\bar{x}, \bar{y})$. Then, $\mathcal{N}_C(\bar{x}, \bar{y})$ satisfies assumption 2.3.1.*

Proof. Let $(\hat{x}, \hat{y}) \in \mathcal{N}_C(\bar{x}, \bar{y})$. Similarly to Proposition 2.3.1, let $(\tilde{x}, \tilde{y}) = p(\bar{x}, \bar{y})$, where p is the permutation associated to $(\hat{x}, \hat{y})$. Note that $\hat{y} = \tilde{y}$. For each k, let $(\tilde{x}^k, \tilde{y}^k) = p(x^k, y^k)$. Then, consider the point $(\hat{x}^k, \hat{y}^k)$ such that $\hat{y}^k = \tilde{y}^k$ and $\hat{x}^k = \Pi_{\mathcal{X}(\hat{y}^k)}(\tilde{x}^k)$. By definition, $(\hat{x}^k, \hat{y}^k) \in \mathcal{N}_C(x^k, y^k)$. Moreover, as in Proposition 2.3.1, for k sufficiently large we get that $\hat{y}^k = \hat{y}$.

Following the same arguments of the proof of Proposition 2.3.1, we get that $\tilde{x}^k \to \tilde{x}$. Since $\Pi_{\mathcal{X}(\hat{y})}(\cdot)$ is continuous, this entails that $\Pi^{\hat{y}}_{\mathcal{X}}(\tilde{x}^k) \to \Pi^{\hat{y}}_{\mathcal{X}}(\tilde{x})$, which is saying that $\hat{x}^k \to \hat{x}$. The claim holds. □

2.4 Algorithmic framework

Here, we discuss an algorithmic framework for the resolution of problem (2.2). Roughly speaking, the approach can be described as an alternate minimization between the continuous and the discrete variables, ensuring that each iterate remains feasible.

In the first phase, we try to improve the continuous part of our solution by doing a local search, of some sorts, around the current iterate x^k, producing a new candidate $\tilde{x}^k$. Here, we consider an Armijo-type line search applied to the projected gradient descent direction, formalized in Algorithm 1.

Algorithm 1: Projected-Gradient Line Search (PGLS)

1 **input**: $y \in \mathcal{Y}, x \in \mathcal{X}(y), \gamma \in (0, \frac{1}{2}), \delta \in (0,1), \alpha = 1$.
2 **Step 1**: Set $\hat{x} = \Pi_{\mathcal{X}(y)}\left[x - \nabla f(x)\right]$, $d = \hat{x} - x$.
3 **Step 2**: If

$$f(x + \alpha d) \leq f(x) + \gamma \alpha \nabla f(x)^\top d,$$

set $\tilde{x} = x + \alpha d$ and exit.
4 **Step 3**: Set $\alpha = \delta\alpha$ and go to Step 2.

The second phase, instead, tries to explore the discrete neighborhood $N_C(\tilde{x}^k, y^k)$. If one of the points in the discrete neighborhood ensures a sufficient decrease of the objective function, we pick it as our new point (x^{k+1}, y^{k+1}) and move on to the next iteration. Otherwise, for any point $(\hat{x}^k, \hat{y}^k)$ in $N_C(\tilde{x}^k, y^k)$ which is not significantly worse (in terms of the objective value) than the current candidate, we perform a con-

tinuous local search around $\hat{x}^k$. The algorithm is formally defined in Algorithm 2.

Algorithm 2: MISO

input: $y^0 \in \mathcal{Y}, x^0 \in \mathcal{X}(y^0), \xi \geq 0, \theta \in (0,1), \eta_0 > 0, \mu_0 > 0, \delta \in (0,1)$.
Step 0: Set $k = 0$.
Step 1: Compute $\tilde{x}^k$ by PGLS(x^k, y^k).
Step 2: If there exists $(\hat{x}^k, \hat{y}^k) \in \mathcal{N}_C(\tilde{x}^k, y^k)$ such that

$$f(\hat{x}^k) \leq f(\tilde{x}^k) - \eta_k,$$

set $x^{k+1} = \hat{x}^k, y^{k+1} = \hat{y}^k, \eta_{k+1} = \eta_k$ and go to Step 5.
Step 3: Define $W_k = \{(x,y) \in \mathcal{N}_C(\tilde{x}^k, y^k) \mid f(x) \leq f(\tilde{x}^k) + \xi\}$.
3.1: If $W_k \neq \emptyset$, choose $(x', y') \in W_k$, set $j = 1, x^j = x'$.
Otherwise, go to Step 4.
3.2: Compute x^{j+1} by PGLS(x^j, y').
3.3: If $f(x^{j+1}) \leq f(\tilde{x}^k) - \eta_k$, set $x^{k+1} = x^{j+1}, y^{k+1} = y', \eta_{k+1} = \eta_k$ and go to Step 5.
3.4: If $\left\| x^j - \Pi_{\mathcal{X}(y')}\left[x^j - \nabla f(x^j)\right] \right\| > \left\| x^k - \Pi_{\mathcal{X}(y^k)}\left[x^k - \nabla f(x^k)\right] \right\| + \mu_k$, set $j = j + 1$ and go to 3.2. Otherwise, set $W_k = W_k \setminus \{(x', y')\}$ and go to 3.1.
Step 4: Set $x^{k+1} = \tilde{x}^k, y^{k+1} = y^k$. If $f(x^{k+1}) \leq f(x^k) - \eta_k$, set $\eta_{k+1} = \eta_k$. Otherwise set $\eta_{k+1} = \theta\eta_k$.
Step 5: Set $\mu_{k+1} = \delta\mu_k, k = k + 1$ and go to Step 1.

2.5 Convergence analysis

In this section, we prove a set of theoretical results concerning the properties of the sequences produced by Algorithm 2. The analysis that follows is strongly related to the ones carried out in Lucidi and Piccialli (2004); Lucidi et al. (2005), with small modifications. First, we state some suitable assumptions.

Assumptions

A1. The gradient $\nabla f(x)$ is Lipschitz-continuous, i.e., there exists a constant $L > 0$ such that

$$\|\nabla f(x) - \nabla f(\bar{x})\| \leq L \|x - \bar{x}\|$$

for all $x, \bar{x} \in \mathbb{R}^n$.

A2. Given $y^0 \in \mathcal{Y}$, $x^0 \in \mathcal{X}(y^0)$ and a scalar $\xi > 0$, the level set

$$\mathcal{L}(x^0, y^0) = \{(x,y) \in \mathcal{X}(y) \times \mathcal{Y} \mid f(x) \leq f(x^0) + \xi\}$$

is compact.

First, we prove a property of Algorithm 1 that will play an important role in the convergence analysis of Algorithm 2.

Proposition 2.5.1. *Given a feasible point (x, y), Algorithm 1 produces a feasible point $(\tilde{x}, y)$ such that*

$$f(\tilde{x}) \leq f(x) - \sigma\left(\left\|x - \Pi_{\mathcal{X}(y)}\left[x - \nabla f(x)\right]\right\|\right),$$

where the function $\sigma(\cdot) \geq 0$ is such that if $\sigma\left(t^h\right) \to 0$ then $t^h \to 0$.

Proof. By definition, $d = \hat{x} - x$, where $\hat{x} = \Pi_{\mathcal{X}(y)}\left[x - \nabla f(x)\right]$. By the properties of the projection operator, we can write

$$(x - \nabla f(x) - \hat{x})^\top (x - \hat{x}) \leq 0,$$

which, with simple manipulations, implies that

$$\nabla f(x)^\top d \leq -\|d\|^2 = -\left\|x - \Pi_{\mathcal{X}(y)}\left[x - \nabla f(x)\right]\right\|^2. \tag{2.15}$$

By the instructions of the algorithm, either $\alpha = 1$ or $\alpha < 1$. If $\alpha = 1$, then $\tilde{x} = x + d$ satisfies

$$f(\tilde{x}) \leq f(x) + \gamma \nabla f(x)^\top d \leq f(x) - \gamma\left\|x - \Pi_{\mathcal{X}(y)}\left[x - \nabla f(x)\right]\right\|^2. \tag{2.16}$$

If $\alpha < 1$, we must have that

$$f(x + \alpha d) \leq f(x) + \gamma \alpha \nabla f(x)^\top d, \tag{2.17}$$

$$f\left(x + \frac{\alpha}{\delta} d\right) > f(x) + \gamma \frac{\alpha}{\delta} \nabla f(x)^\top d. \tag{2.18}$$

Applying the mean value theorem to equation (2.18), we get

$$\nabla f\left(x + \theta \frac{\alpha}{\delta} d\right)^\top d > \gamma \nabla f(x)^\top d,$$

where $\theta \in (0, 1)$. Adding and subtracting $\nabla f(x)^\top d$, and rearranging, we get

$$(1 - \gamma) \nabla f(x)^\top d > \left[\nabla f(x) - \nabla f\left(x + \theta \frac{\alpha}{\delta} d\right)\right]^\top d.$$

By the Lipschitz-continuity of $\nabla f(x)$, we can write

$$\left[\nabla f(x) - \nabla f\left(x + \theta \frac{\alpha}{\delta} d\right)\right]^\top d \geq -L \frac{\alpha}{\delta} \|d\|^2,$$

which means that

$$(1 - \gamma) \nabla f(x)^\top d > -L \frac{\alpha}{\delta} \|d\|^2,$$

Rearranging, we get

$$\frac{\delta}{L}(1-\gamma)\nabla f(x)^\top d > -\alpha \|d\|^2. \tag{2.19}$$

This last inequality, together with (2.15), yields

$$\frac{\delta}{L}(1-\gamma)\nabla f(x)^\top d > \alpha \nabla f(x)^\top d, \tag{2.20}$$

and substituting in equation (2.17) we finally get

$$f(\tilde{x}) < f(x) + \gamma\frac{\delta}{L}(1-\gamma)\nabla f(x)^\top d \leq f(x) - \gamma\frac{\delta}{L}(1-\gamma)\left\|x - \Pi_{\mathcal{X}(y)}\left[x - \nabla f(x)\right]\right\|^2.$$

This last inequality, together with (2.16), implies that

$$f(\tilde{x}) \leq f(x) - \sigma\left(\left\|x - \Pi_{\mathcal{X}(y)}\left[x - \nabla f(x)\right]\right\|\right)$$

where

$$\sigma(t) = \gamma \min\left\{1, \frac{\delta}{L}(1-\gamma)\right\} t^2.$$

□

We can now state a couple of preliminary theoretical results. We first show that Algorithm 2 is well-posed.

Proposition 2.5.2. *For each iteration k, Step 3 of Algorithm 2 terminates in a finite number of steps.*

Proof. Suppose by contradiction that Steps 3.1-3.4 generate an infinite loop, so that an infinite sequence of points $\{x^j\}$ is produced for which

$$\left\|x^j - \Pi_{\mathcal{X}(y')}\left[x^j - \nabla f(x^j)\right]\right\| > \left\|x^k - \Pi_{\mathcal{X}(y^k)}\left[x^k - \nabla f(x^k)\right]\right\| + \mu_k > 0 \quad \forall j. \tag{2.21}$$

By proposition 2.5.1, for each j we have that

$$f(x^{j+1}) - f(x^j) \leq -\sigma\left(\left\|x^j - \Pi_{\mathcal{X}(y')}\left[x^j - \nabla f(x^j)\right]\right\|\right), \tag{2.22}$$

where $\sigma(\cdot) \geq 0$. The sequence $\{f(x^j)\}$ is therefore nonincreasing. Moreover, (2.22) implies that

$$\left|f(x^{j+1}) - f(x^j)\right| \geq \sigma\left(\left\|x^j - \Pi_{\mathcal{X}(y')}\left[x^j - \nabla f(x^j)\right]\right\|\right). \tag{2.23}$$

By assumption A2, $\{f(x^j)\}$ is lower bounded. Therefore, recalling that $\{f(x^j)\}$ is nonincreasing, we get that $\{f(x^j)\}$ converges, which implies that

$$\left|f(x^{j+1}) - f(x^j)\right| \to 0.$$

By (2.23), we get that $\sigma\left(\left\|x^j - \Pi_{\mathcal{X}(y')}\left[x^j - \nabla f(x^j)\right]\right\|\right) \to 0$, and, by the properties of $\sigma(\cdot)$, we finally get that $\left\|x^j - \Pi_{\mathcal{X}(y')}\left[x^j - \nabla f(x^j)\right]\right\| \to 0$, and this contradicts (2.21). □

The next proposition shows some properties of the sequences generated by the algorithm, which will play an important role in the subsequent analysis.

Proposition 2.5.3. *Let $\{(x^k, y^k)\}$, μ_k and η_k be the sequences produced by the algorithm. Then:*

(i) *the sequence $\{(x^k, y^k)\}$ is bounded;*

(ii) *the sequence $\{f(x^k)\}$ is nonincreasing and convergent;*

(iii) *the set $K_u = \{k \mid \eta_k < \eta_{k-1}\}$ of unsuccessful iterates is infinite;*

(iv) $\lim_{k\to\infty} \mu_k = 0$;

(v) $\lim_{k\to\infty} \eta_k = 0$;

(vi) $\lim_{k\to\infty} \left\|x^k - \Pi_{\mathcal{X}(y^k)}\left[x^k - \nabla f(x^k)\right]\right\| = 0$.

Proof. (i) The instructions of the algorithm imply that each point (x^k, y^k) belongs to the level set $\mathcal{L}(x^0, y^0)$, which is compact by assumption A2. Therefore, $\{(x^k, y^k)\}$ is bounded.

(ii) The instructions of the algorithm and proposition 2.5.1 imply that $\{f(x^k)\}$ is nonincreasing, and assumption A2 implies that $\{f(x^k)\}$ is lower bounded. Hence, $\{f(x^k)\}$ converges.

(iii) Suppose that K_u is finite. Then there exists $\bar{k} > 0$ such that all iterates satisfying $k > \bar{k}$ are successful, i.e.,

$$f(x^k) \leq f(x^{k-1}) - \eta_{k-1},$$

and $\eta_k = \eta_{k-1} = \eta > 0$ for all $k \geq \bar{k}$. Since $\eta > 0$, this implies that $\{f(x^k)\}$ diverges to $-\infty$, in contradiction with point (ii).

(iv) If $k \in K_u$, then $\eta_{k+1} = \theta\eta_k$, where $\theta \in (0, 1)$. Since K_u is infinite, the claim holds.

(v) Since, for all k, $\mu_{k+1} = \delta\mu_k$, where $\delta \in (0, 1)$, the claim holds.

(vi) By proposition 2.5.1, we have that

$$f(\tilde{x}^k) - f(x^k) \leq -\sigma\left(\left\|x^k - \Pi_{\mathcal{X}(y^k)}\left[x^k - \nabla f(x^k)\right]\right\|\right).$$

By the instructions of the algorithm, $f(x^{k+1}) \leq f(\tilde{x}^k)$, and so we can write

$$f(x^{k+1}) - f(x^k) \leq -\sigma\left(\left\|x^k - \Pi_{\mathcal{X}(y^k)}\left[x^k - \nabla f(x^k)\right]\right\|\right),$$

i.e.,

$$\left|f(x^{k+1}) - f(x^k)\right| \geq \sigma\left(\left\|x^k - \Pi_{\mathcal{X}(y^k)}\left[x^k - \nabla f(x^k)\right]\right\|\right).$$

Since $\{f(x^k)\}$ converges, we get that $\sigma\left(\left\|x^k - \Pi_{\mathcal{X}(y^k)}\left[x^k - \nabla f(x^k)\right]\right\|\right) \to 0$. By the properties of $\sigma(\cdot)$, we get that $\left\|x^k - \Pi_{\mathcal{X}(y^k)}\left[x^k - \nabla f(x^k)\right]\right\| \to 0$.

□

Before stating the main theorem of this section, it is useful to summarize some theoretical properties of the subsequence $\{(x^k, y^k)\}_{K_u}$ of the unsuccessful iterates.

Proposition 2.5.4. *Let $\{(x^k, y^k)\}$ be the sequence of iterates generated by Algorithm 2, and let $K_u = \{k \mid \eta_k < \eta_{k-1}\}$. Then:*

(i) *$\{(x^k, y^k)\}_{K_u}$ admits accumulation points;*

(ii) *for any accumulation point (x^*, y^*) of the sequence $\{(x^k, y^k)\}_{K_u}$, every $(\hat{x}, \hat{y}) \in \mathcal{N}_C(x^*, y^*)$ is an accumulation point of a sequence $\{(\hat{x}^k, \hat{y}^k)\}_{K_u}$ where $(\hat{x}^k, \hat{y}^k) \in \mathcal{N}_C(x^k, y^k)$.*

Proof. (i) By proposition 2.5.3, $\{(x^k, y^k)\}$ is bounded. Therefore, $\{(x^k, y^k)\}_{K_u}$ is also bounded, and so it admits accumulation points.

(ii) proposition 2.3.2 implies that every $(\hat{x}, \hat{y}) \in \mathcal{N}_C(x^*, y^*)$ is an accumulation point of a sequence $\{(\hat{x}^k, \hat{y}^k)\}_{K_u}$, where $(\hat{x}^k, \hat{y}^k) \in \mathcal{N}_C(x^k, y^k)$.

□

We can now prove the main theoretical result of this section.

Theorem 2.5.1. *Let $\{(x^k, y^k)\}$ be the sequence of iterates generated by Algorithm 2. Every accumulation point of $\{(x^k, y^k)\}_{K_u}$ is a stationary point of problem* (2.2).

Proof. Let (x^*, y^*) be an accumulation point of $\{(x^k, y^k)\}_{K_u}$. We must show that conditions (i)-(iii) of definition 2.2.2 are satisfied.

(i) The result follows from propositions 2.5.3, (vi).

(ii) Since K_u is an infinite subset of unsuccessful iterations, recalling that $x^k = \tilde{x}^{k-1}$, $y^k = y^{k-1}$, and setting $\hat{x}^k = \hat{x}^{k-1}$, $\hat{y}^k = \hat{y}^{k-1}$ for all $(\hat{x}^{k-1}, \hat{y}^{k-1}) \in \mathcal{N}_C(\tilde{x}^{k-1}, y^{k-1})$, the tests at Step 2 and 4 fail at iteration k, and therefore

$$f(\hat{x}^k) > f(x^k) - \eta_{k-1} \tag{2.24}$$

for all $(\hat{x}^k, \hat{y}^k) \in \mathcal{N}_C(x^k, y^k)$. Since the sequence $\{f(x^k)\}$ is nonincreasing (propositions 2.5.3, (ii)), we can write

$$f(x^*) \leq f(x^k) < f(\hat{x}^k) + \eta_{k-1}. \tag{2.25}$$

for all $(\hat{x}^k, \hat{y}^k) \in \mathcal{N}_C(x^k, y^k)$. Taking limits, the result follows from propositions 2.5.3(v), 2.3.1, and from the continuity of f.

(iii) Note that point (ii) of 2.5.3 ensures the existence of $f^* \in \mathbb{R}$ satisfying

$$\lim_{k\to\infty} f(x^k) = f(x^*) = f^*. \tag{2.26}$$

Consider any $(\hat{x}, \hat{y}) \in \mathcal{N}_C(x^*, y^*)$ such that

$$f(\hat{x}) = f^*. \tag{2.27}$$

Proposition 2.5.4 implies that $(\hat{x}, \hat{y})$ is an accumulation point of a sequence $\{(\hat{x}^k, \hat{y}^k)\}_{K_u}$, where $(\hat{x}^k, \hat{y}^k) \in \mathcal{N}_C(x^k, y^k)$. Since $k \in K_u$, we have that $x^k = \tilde{x}^{k-1}, y^k = y^{k-1}$. Setting $\hat{x}^k = \hat{x}^{k-1}, \hat{y}^k = \hat{y}^{k-1}$ for all $(\hat{x}^{k-1}, \hat{y}^{k-1}) \in \mathcal{N}_C(\tilde{x}^{k-1}, y^{k-1})$, by (2.26) and (2.27) we get, for k sufficiently large,

$$f(\hat{x}^k) < f(x^k) + \xi.$$

Therefore, for such values of k, $(\hat{x}^k, \hat{y}^k) \in W_k$, and Steps 3.2-3.4 produce the points $x_k^2, \ldots, x_k^{j_k^*}$ (where j_k^* is the finite number of iterations of Steps 3.2-3.4 until the test at Step 3.4 fails), which, by the instructions at Step 3.2 and by proposition 2.5.1, satisfy

$$f(\hat{x}^k) \geq f(x_k^2) \geq \ldots \geq f(x_k^{j_k^*}). \tag{2.28}$$

Since $k \in K_u$, Step 3.3 fails, and we can write

$$f(x_k^{j_k^*}) > f(\tilde{x}^k) - \eta_k \geq f(x^k) - \eta_{k-1}. \tag{2.29}$$

Moreover, as the sequence $\{(\hat{x}^k, \hat{y}^k)\}_{K_u}$ converges to the point $(\hat{x}, \hat{y})$, by (2.26), (2.27), (2.28), (2.29), and by point (v) of (2.5.3), we obtain

$$f^* = \lim_{k\to\infty, k\in K_u} f(\hat{x}^k) = \lim_{k\to\infty, k\in K_u} f(x_k^2) = \lim_{k\to\infty, k\in K_u} f(x^k) = f^*. \tag{2.30}$$

By proposition 2.5.1, we have that

$$f(x_k^2) \leq f(\hat{x}^k) - \sigma\left(\left\|\hat{x}^k - \Pi_{\mathcal{X}(\hat{y}^k)}\left[\hat{x}^k - \nabla f(\hat{x}^k)\right]\right\|\right), \tag{2.31}$$

which can be rewritten as

$$\left|f(x_k^2) - f(\hat{x}^k)\right| \geq \sigma\left(\left\|\hat{x}^k - \Pi_{\mathcal{X}(\hat{y}^k)}\left[\hat{x}^k - \nabla f(\hat{x}^k)\right]\right\|\right). \tag{2.32}$$

Taking limits for $k \to \infty, k \in K_u$, we finally get

$$\left\|\hat{x} - \Pi_{\mathcal{X}(\hat{y})}\left[\hat{x} - \nabla f(\hat{x})\right]\right\| = 0, \tag{2.33}$$

and the claim holds.

□

2.6 Numerical results

To assess the performance of the proposed approach, we compared Algorithm 2 with the Greedy Sparse-Simplex method proposed in Beck and Eldar (2013) and with the Penalty Decomposition approach by Lu and Zhang (2013) on a set of cardinality constrained least squares problems of the form

$$\begin{array}{ll} \min\limits_{x} & \|Ax - b\|^2 \\ \text{s.t.} & \|x\|_0 \leq s, \end{array} \tag{2.34}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^n$. We considered three different values of n, namely, $n \in \{20, 100, 1000\}$, and for a fixed n we considered the following values of m and s:

- $m \in \{n * 0.25, n * 0.5, n * 0.75\}$,
- $s \in \{n * 0.1, n * 0.2, n * 0.3, n * 0.4, n * 0.5\}$.

Therefore, we considered 45 different configurations of n, m and s. For each given configuration, we randomly generated 100 instances of problem (2.34). In particular, matrix A and vector b are generated as in Beck and Eldar (2013), while the optimal solution x^* was generated by setting $2 \cdot s$ randomly chosen components x_i^* to either 1 or -1.

Exact minimization

One of the characteristics of the least squares problems we consider in this section is the fact that the subproblem w.r.t. any given subset of components admits a closed-form solution. In other words, after choosing a subset of s potentially nonzero components, the corresponding candidate solution of problem (2.34) is easy to obtain.

In the context of Algorithm 2, this means that we can replace the local search w.r.t. the continuous variables with the exact minimization w.r.t. the chosen components. Therefore, alongside the "vanilla" implementation of Algorithm 2, we also report the result obtained by the "exact minimization" implementation. In the following, we will refer to these two implementations as MISO_{LS} and $\text{MISO}_{\text{EXACT}}$, respectively.

Implementation details

The parameters of Algorithm 2 have been set as follows:

- $\xi = 10^{-2}$,
- $\theta = 0.5$,
- $\eta_0 = 10^{-5}$.

For what concerns μ_0 and δ, we actually keep the value of μ fixed to 10^{-6}.

For the Greedy Sparse-Simplex method we used the implementation available at `https://sites.google.com/site/amirbeck314/software`. The implementation of the Penalty Decomposition method can be found at `http://people.math.sfu.ca/~zhaosong/`. We used the default values of the parameters for each method.

As for the stopping criterion, we run each algorithm for a maximum of 200 iterations.

Results

Here, we show the performance profiles (Dolan and Moré (2002)) w.r.t. the objective function values obtained by the different algorithms. In particular, for each problem and each algorithm, we consider the objective function $f(\bar{x})$, where $\bar{x}$ is the retrieved solution[1].

The results are depicted in figures 2.1-2.3. While we stress the fact that the scope of this work is mainly theoretical, we feel that MISO stands its ground against the two state-of-the-art methods we compare it against. This is especially true for $n = 20$ and $n = 100$, while the PD approach seems to be quite strong for the large scale case $n = 1000$, particularly for large values of m.

The comparison against GSS instead, shows that MISO outperforms it numerous times across the different configurations, especially for higher values of n. This is particularly interesting since the GSS approach, with its iterative exploration of single components, embodies a philosophy which is not too distant from the ideas at the foundation of MISO.

[1]Note that this metric is adequate, since it is always nonnegative and a lower value is better. To avoid numerical problems when the best objective value obtained is 0, we add a small constant (0.1) to each objective value computed.

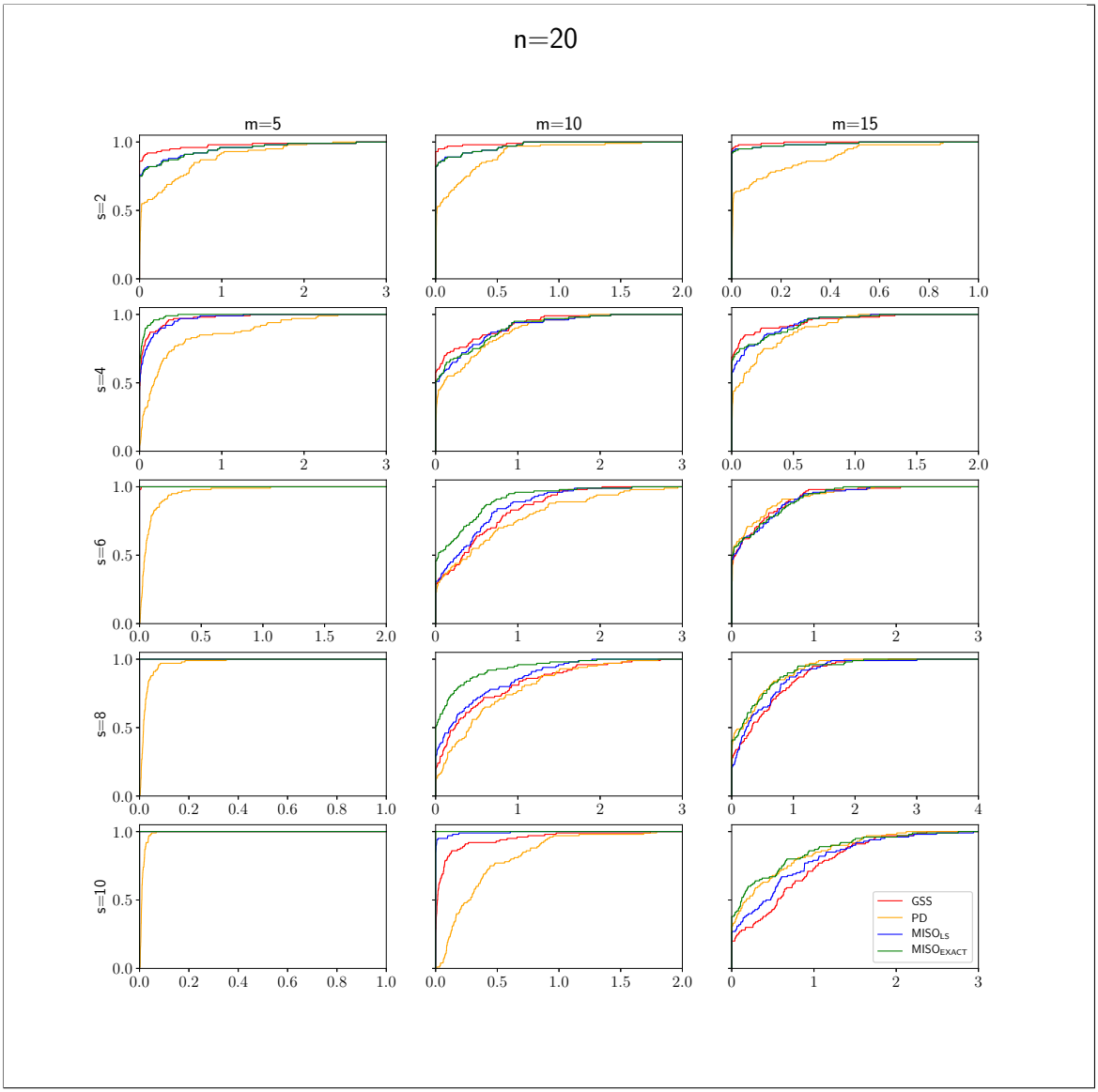


Figure 2.1: Performance profiles computed over the 15 configurations of m and s for $n = 20$.

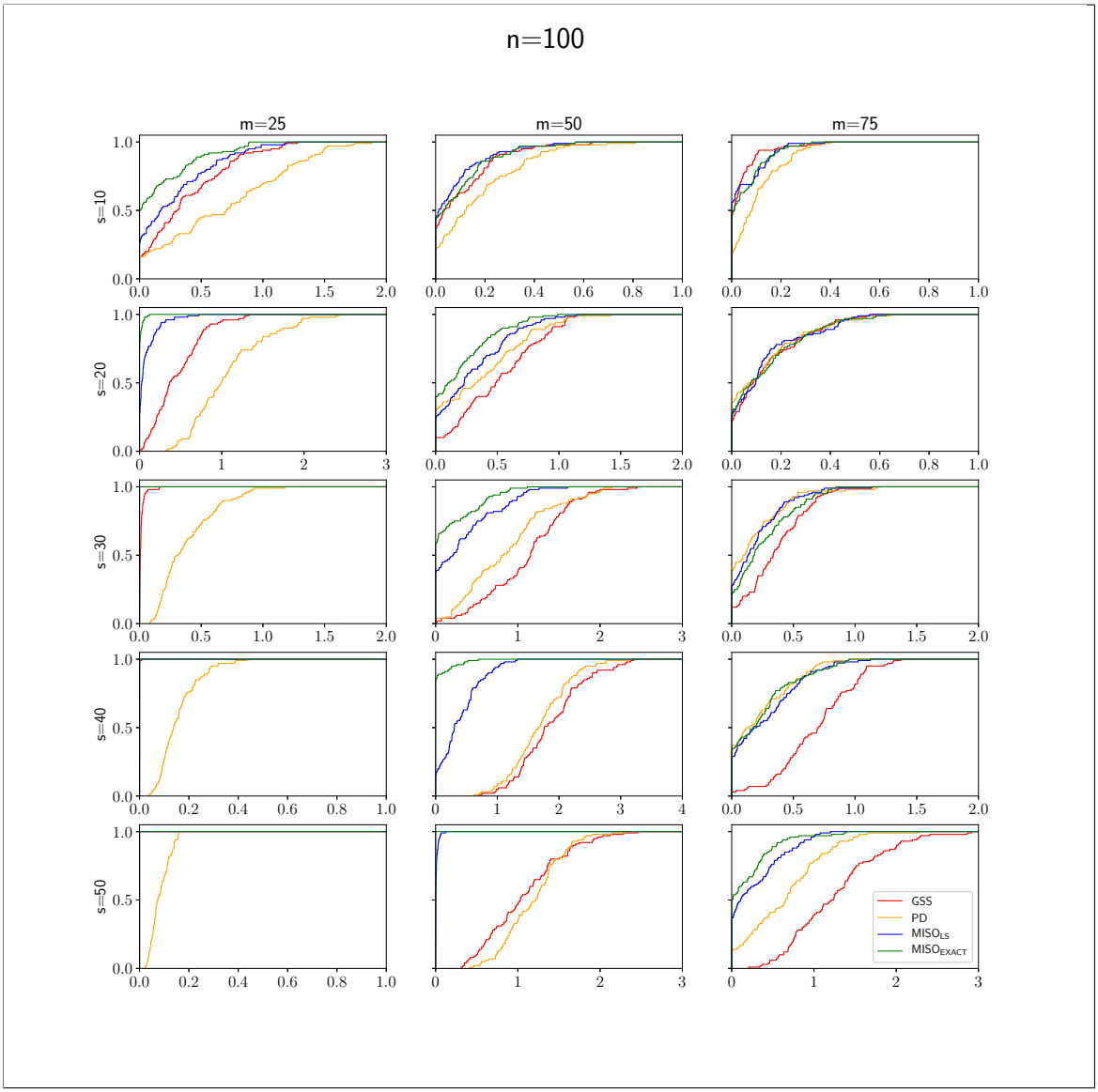


Figure 2.2: Performance profiles computed over the 15 configurations of m and s for $n = 100$.

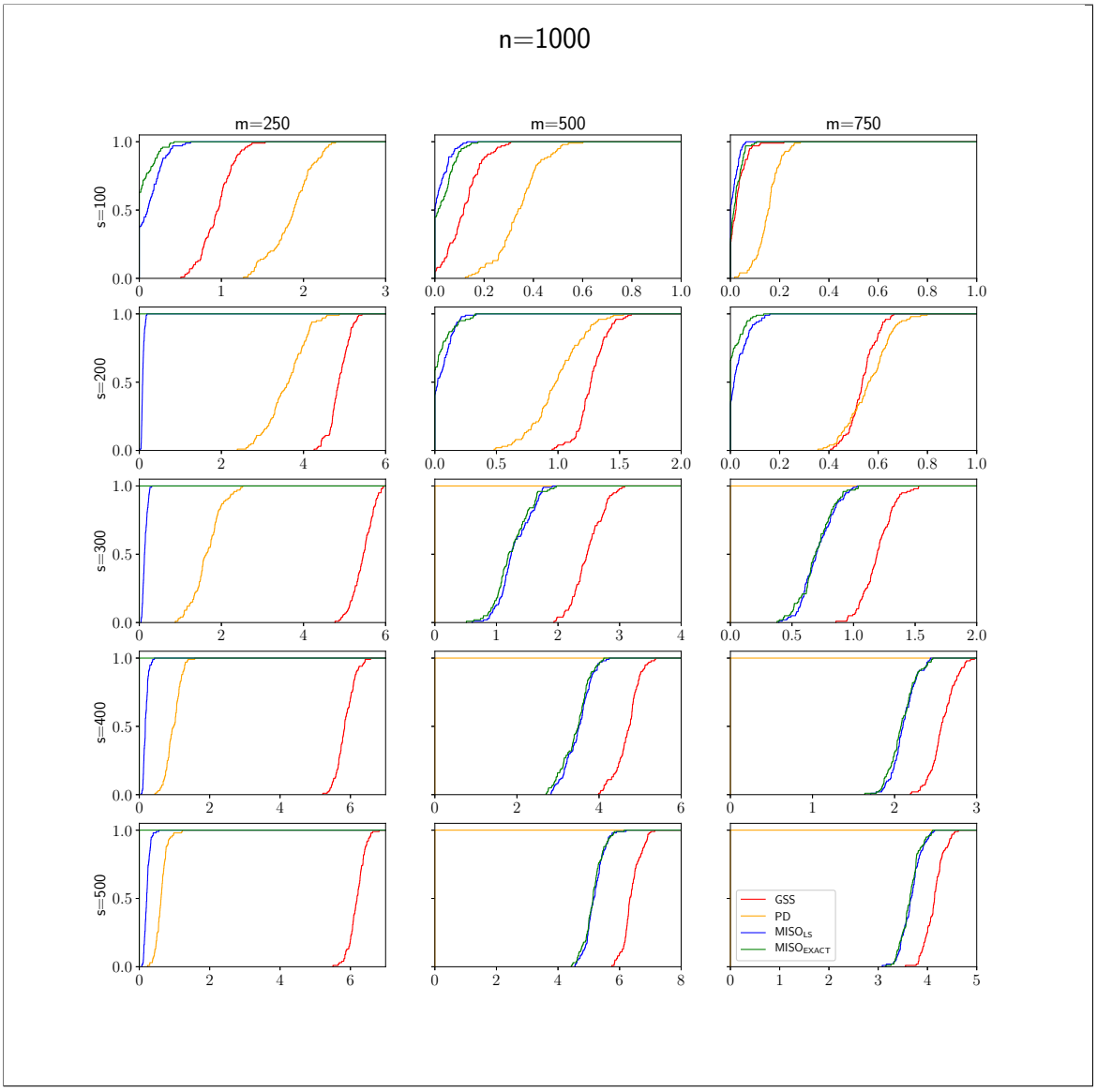


Figure 2.3: Performance profiles computed over the 15 configurations of m and s for $n = 1000$.

2.7 Conclusions

In this chapter we introduced a mixed integer approach for cardinality constrained optimization problems with nonconvex objective function, under additional convex constraints. The method is based on the exploration of discrete neighborhoods of the current iterate, making sure the cardinality constraint is satisfied, which is followed by an optimization phase based on line searches on the chosen free variables.

Under mild assumptions, we proved the convergence of this iterative scheme to points satisfying a suitable optimality condition, stronger than many optimality conditions commonly used in the literature.

We tested the effectiveness of our approach on a set of cardinality constrained least squares problem, and compared the results against two state-of-the-art algorithms, namely GSS and PD. The results show that our algorithm is competitive, and even better than our competitors in a decent number of cases.

Chapter 3

A penalty decomposition approach for nonconvex cardinality constrained optimization

Another popular approach for dealing with cardinality constrained problems is the Penalty Decomposition method proposed in Lu and Zhang (2013), informally introduced in the previous chapter. The method deals with problems like

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & x \in X, \\ & h(x) = 0, \\ & g(x) \leq 0, \\ & \|x\|_0 \leq s, \end{aligned} \tag{3.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^m$, $h : \mathbb{R}^n \to \mathbb{R}^p$ are continuously differentiable functions and $X \subseteq \mathbb{R}^n$ is a closed convex set.

The approach is based on an equivalent formulation that introduces auxiliary variables y with the added constraint $x = y$. Then, we replace the original cardinality constraint with the term $\|y\|_0 \leq s$. The equality constraint is then moved into the objective function in a suitable penalization term, resulting in a series of subproblems in which the penalty gets increasingly bigger. Each subproblem is minimized in an alternating fashion between x and y, where the x step is a the minimization of a smooth nonlinear problem and the y step is a simple projection on the cardinality constraint. This "variable splitting" scheme, as it is also known in literature, is then guaranteed to converge to a feasible point of problem (3.1) which satisfies a suitable necessary optimality condition.

This approach, although powerful and efficient, suffers from an important theoretical shortcoming: the algorithm, and the following convergence analysis, assumes that in the minimization phase w.r.t. x we are able to find a global solution

of a general nonconvex problem, which can be prohibitive, even in the case where $X = \mathbb{R}^n$ and the additional constraints $h(x), g(x)$ are not present.

In this chapter, we consider an "unconstrained" version of problem (3.1), i.e., problems of the form

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & \|x\|_0 \leq s, \end{aligned} \tag{3.2}$$

and we propose a variant of the algorithm that replaces the original exact minimization step with an Armijo-type line search, more suited to deal with nonconvex objective functions.

We prove that this adaptation enjoys the same theoretical guarantees as the original algorithm, and we also provide additional conditions ensuring the convergence to stronger optimality conditions. Moreover, we also provide a derivative free extension, performing the x minimization step with line searches along the coordinate axes.

In the next section, we provide some details about the original Penalty Decomposition approach proposed in Lu and Zhang (2013).

3.1 Preliminaries

Problem (3.2) can be equivalently stated as

$$\begin{aligned} \min_{x,y \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & \|y\|_0 \leq s, \\ & x = y. \end{aligned} \tag{3.3}$$

For simplicity, in the following, we will refer to the set $\{y \in \mathbb{R}^n \mid \|y\|_0 \leq s\}$ as Y. Defining, for any $\tau > 0$, the function

$$q_\tau(x, y) = f(x) + \frac{\tau}{2}\|x - y\|^2, \tag{3.4}$$

we can then consider the sequence of subproblems

$$\begin{aligned} \min_{x,y \in \mathbb{R}^n} \quad & q_\tau(x, y) \\ \text{s.t.} \quad & \|y\|_0 \leq s. \end{aligned} \tag{3.5}$$

This idea leads to the original Penalty Decomposition algorithm, formalized in Algorithm 3.

Algorithm 3: Penalty Decomposition

Input: $\tau_0 > 0$, $\theta > 1$, $x^0 = y^0 \in \mathbb{R}^n$ s.t. $\|x^0\|_0 \leq s$, a sequence $\{\varepsilon_k\}$ s.t. $\varepsilon_k \to 0$, $\Gamma \geq \max\{f(x^0), \min_x q_{\tau_0}(x, y^0)\}$.
for $k = 0, 1, \ldots$ **do**
  $\ell = 0$
  $u^0 = x^k$
  if $\min_x q_{\tau_k}(x, y^k) \leq \Gamma$ **then**
    $v^0 = y^k$
  end
  else
    $v^0 = y^0$
  end
  while $\|\nabla_x q_{\tau_k}(u^\ell, v^\ell)\| > \varepsilon_k$ **do**
    $u^{\ell+1} = \arg\min_u q_{\tau_k}(u, v^\ell)$
    $v^{\ell+1} = \arg\min_{v \in Y} q_{\tau_k}(u^{\ell+1}, v)$
    $\ell = \ell + 1$
  end
  $\tau_{k+1} = \theta \tau_k$
  x^{k+1}, $y^{k+1} = u^\ell$, v^ℓ
end
Output: The sequence $\{x^k\}$.

Under suitable assumptions, it can be shown that the Penalty Decomposition method converges to points satisfying the following necessary optimality condition, namely, the Lu-Zhang conditions, introduced in Lu and Zhang (2013)[1].

Definition 3.1.1. We say that a point $\bar{x} \in \mathbb{R}^n$ satisfies Lu-Zhang first order optimality conditions if there exists a set $I \subseteq \{1, \ldots, n\}$ such that $|I| = s$, $\bar{x}_i = 0$ for all $i \in \bar{I} = \{1, \ldots, n\} \setminus I$ and $\nabla_i f(\bar{x}) = 0$ for all $i \in I$.

It is trivial to realize that the concept of BF-stationarity introduced earlier (Definition 2.2.3) implies Lu-Zhang conditions. The converse is not necessarily true, i.e., Lu-Zhang conditions are weaker than BF-stationarity. We show this with the following example.

Example 3.1.1. *Let*

$$f(x) = (x_1 - 1)^2 + x_2^2 + (x_3 - 1)^2$$

and $s = 2$. *The point* $\bar{x} = [1\ 0\ 0]$ *satisfies Lu-Zhang conditions but it is not BF-stationary. Indeed, let* $J = \{1, 2\}$. *We have that* $\bar{x}_j = 0$ *for all* $j \in \bar{J}$ *and* $\nabla_i f(\bar{x}) = 0$ *for all* $i \in J$.

[1]The original definition given in Lu and Zhang (2013) takes into account additional constraints $h(x) = 0$, $g(x) \leq 0$. Here, we restate the original condition into an equivalent but simplified form when $X = \mathbb{R}^n$.

Thus $\bar{x}$ is Lu-Zhang stationary. On the other hand, $\|\bar{x}\|_0 < 2$, and $\nabla_3 f(\bar{x}) \neq 0$, i.e., it is not BF-stationary.

Before delving into the technical details of this work, we state the following assumption on the objective function $f(x)$.

Assumption 3.1.1. *The objective function $f(x)$ is coercive.*

As we now show, under Assumption 3.1.1, it also holds that the penalized function $q_\tau(x, y)$ is coercive.

Lemma 3.1.1. *The function $q_\tau(x, y)$ defined in (3.4) is coercive on $\mathbb{R}^n \times \mathbb{R}^n$.*

Proof. Suppose that $\{(x^k, y^k)\}$ is a sequence such that either $\|x^k\| \to \infty$ or $\|y^k\| \to \infty$.

If $\|x^k\| \to \infty$, then $f(x^k) \to \infty$ as $f(x)$ is coercive, and so does $q_\tau(x^k, y^k)$.

If $\|y^k\| \to \infty$, then either $\|x^k\| \to \infty$ and we are in the previous case, or $\|x^k - y^k\| \to \infty$. In the latter case, $q_\tau(x^k, y^k) \to \infty$ as $f(x)$ is coercive and, consequently, bounded below. □

3.2 Inexact Penalty Decomposition approach

Algorithm 3 requires the exact solution of a potentially nonconvex optimization problem at steps 5 and 12, which can be prohibitive in practice. To address this issue, we propose a modification of the algorithm for nonconvex optimization.

The proposed procedure is described in Algorithm 4. The global optimization step of a possibly nonconvex function has been replaced by a descent step along the gradient of q_{τ_k} w.r.t. the x block. The stepsize is computed by the classical Armijo backtracking line search.

We now address the properties of the Inexact Penalty Decomposition method. First, we show that Algorithm 4 is indeed well posed.

Proposition 3.2.1. *Algorithm 4 does not loop infinitely between steps 11-14.*

Proof. Suppose by contradiction that, at a certain iteration k, the sequence $\{u^\ell, v^\ell\}$ is infinite. From the instructions of the algorithm, we have

$$q_{\tau_k}(u^{\ell+1}, v^{\ell+1}) \leq q_{\tau_k}(u^0, v^0).$$

Hence, $\{u^\ell, v^\ell\} \subset \mathcal{L}_0(q_{\tau_k})$, which is compact from the coercivity of q_{τ_k}.

Therefore, there exists $K \subseteq \{0, 1, \ldots\}$ such that $(u^\ell, v^\ell) \to_K (\bar{u}, \bar{v})$. From the instructions of the algorithm we have

$$q_{\tau_k}(u^{\ell+1}, v^{\ell+1}) \leq q_{\tau_k}(u^{\ell+1}, v^\ell) = q_{\tau_k}(u^\ell - \alpha_\ell \nabla_x q_{\tau_k}(u^\ell, v^\ell), v^\ell) < q_{\tau_k}(u^\ell, v^\ell).$$

Algorithm 4: Inexact Penalty Decomposition

Input: $\tau_0 > 0$, $\theta > 1$, $x^0 = y^0 \in \mathbb{R}^n$ s.t. $\|x^0\|_0 \leq s$, a sequence $\{\varepsilon_k\}$ s.t. $\varepsilon_k \to 0$.
for $k = 0, 1, \ldots$ **do**
  $\ell = 0$
  $x_{\text{trial}} = x^k - \alpha \nabla_x q_{\tau_k}(x^k, y^k)$ with α computed by Armijo line search
  if $q_{\tau_k}(x_{trial}, y^k) \leq f(x^0)$ **then**
    $u^0, v^0 = x^k, y^k$
  end
  else
    $u^0, v^0 = x^0, y^0$
  end
  while $\|\nabla_x q_{\tau_k}(u^\ell, v^\ell)\| > \varepsilon_k$ **do**
    $u^{\ell+1} = u^\ell - \alpha_\ell \nabla_x q_{\tau_k}(u^\ell, v^\ell)$ with α_ℓ computed by Armijo line search
    $v^{\ell+1} = \arg\min_{v \in Y} q_{\tau_k}(u^{\ell+1}, v)$
    $\ell = \ell + 1$
  end
  $\tau_{k+1} = \theta \tau_k$
  $x^{k+1} = u^\ell$
  $y^{k+1} = v^\ell$
end
Output: The sequence $\{x^k\}$.

Since f is continuously differentiable, $\nabla_x q_{\tau_k}(u, v) = \nabla f(u) + \tau(u - v)$ is continuous, therefore $\nabla_x q_{\tau_k}(u^\ell, v^\ell) \to_K \nabla_x q_{\tau_k}(\bar{u}, \bar{v})$.

The sequence $\{q_{\tau_k}(u^\ell, v^\ell)\}$ is decreasing, so it admits a limit, which may be $-\infty$. But, recalling the continuity of q_{τ_k}, we have that $q_{\tau_k}(u^\ell, v^\ell) \to_K q_{\tau_k}(\bar{u}, \bar{v})$. Hence, the whole sequence converges to $q_{\tau_k}(\bar{u}, \bar{v})$.

Then we have

$$\lim_{\ell \to \infty} q_{\tau_k}(u^\ell, v^\ell) - q_{\tau_k}(u^\ell - \alpha_\ell \nabla_x q_{\tau_k}(u^\ell, v^\ell), v^\ell) = 0. \tag{3.6}$$

Therefore, from the properties of Armijo line search we have

$$\lim_{\substack{\ell \in K \\ \ell \to \infty}} \left\| \nabla_x q_{\tau_k}(u^\ell, v^\ell) \right\| = 0,$$

a contradiction. □

Now, we can turn to the convergence analysis of the algorithm. We start by proving the existence of limit points of the generated sequence.

Proposition 3.2.2. *Let $\{x^k, y^k\}$ be the sequence generated by Algorithm 4. Then $\{x^k, y^k\}$ admits cluster points.*

Proof. Consider a generic iteration k. Since instructions 12-13 of the algorithm both do not increase the value of q_{τ_k}, we have that

$$q_{\tau_k}(x^{k+1}, y^{k+1}) \leq ... \leq q_{\tau_k}(u^1, v^0) = q_{\tau_k}(u^0 - \alpha_0 \nabla_x q_{\tau_k}(u^0, v^0), v^0). \tag{3.7}$$

From the definition of (u^0, v^0), we either have $(u^0, v^0) = (x^k, y^k)$ or $(u^0, v^0) = (x^0, y^0)$. In the first case, we have, by the definition of x_{trial}, that

$$q_{\tau_k}(u^0 - \alpha_0 \nabla_x q_{\tau_k}(u^0, v^0), v^0) = q_{\tau_k}(x_{\text{trial}}, y^k) \leq f(x^0),$$

where the last inequality holds, as in this case the condition at line 5 is satisfied. In the second case we have

$$\begin{aligned} q_{\tau_k}(u^0 - \alpha_0 \nabla_x q_{\tau_k}(u^0, v^0), v^0) &\leq q_{\tau_k}(u^0, v^0) \\ &= q_{\tau_k}(x^0, y^0) \\ &= f(x^0) + \frac{\tau_k}{2}\|x^0 - y^0\|^2 = f(x^0). \end{aligned}$$

So, putting everything back together in (3.7) we get

$$q_{\tau_k}(x^{k+1}, y^{k+1}) \leq f(x^0). \tag{3.8}$$

But, by the definition of q_{τ_k} it also holds

$$f(x^{k+1}) \leq q_{\tau_k}(x^{k+1}, y^{k+1}).$$

Then $f(x^{k+1}) \leq f(x^0)$. As k is arbitrary, it follows that $\{x^{k+1}\} \subset \mathcal{L}_0$, which is a compact set by the coercivity assumptions; hence $\{x^k\}$ is bounded.

From equation (3.8), we also have

$$q_{\tau_k}(x^{k+1}, y^{k+1}) = f(x^{k+1}) + \frac{\tau_k}{2}\|x^{k+1} - y^{k+1}\|^2 \leq f(x^0).$$

Dividing by τ_k we get

$$\|x^{k+1} - y^{k+1}\|^2 \leq 2\frac{f(x^0) - f(x^{k+1})}{\tau_k}.$$

Taking the limits for $k \to \infty$, recalling the boundedness of $\{x^{k+1}\}$ and that $\tau_k \to \infty$, we have that $\{y^{k+1}\}$ is also a bounded sequence. Hence, the sequence $\{x^k, y^k\}$ is bounded and therefore admits limit points, as we wanted to prove. □

Proposition 3.2.3. *Let $\{x^k, y^k\}$ be the sequence generated by Algorithm 4. Suppose that $(\bar{x}, \bar{y})$ is a limit point of $\{x^k, y^k\}$, i.e. $(x^k, y^k) \to_K (\bar{x}, \bar{y})$. Then, $(\bar{x}, \bar{y})$ is a feasible point for problem (3.3), and $\bar{x}$ is feasible for problem (3.2).*

Proof. From the instructions of the algorithm, for each k we have that

$$\|\nabla f(x^{k+1}) + \tau_k(x^{k+1} - y^{k+1})\| \leq \epsilon_k.$$

Dividing both sides by τ_k, we get

$$\left\| \frac{\nabla f(x^{k+1})}{\tau_k} + (x^{k+1} - y^{k+1}) \right\| \leq \frac{\epsilon_k}{\tau_k}.$$

Taking limits for $k \to \infty$, $k \in K$, recalling that $\tau_k \to \infty$, $\epsilon_k \to 0$ and that $\{\nabla f(x^k)\}_K$ is bounded as ∇f is continuous, we get

$$\|\bar{x} - \bar{y}\| = \lim_{\substack{k \in K \\ k \to \infty}} \|x^k - y^k\| \leq 0,$$

and the statement holds recalling that $\|\bar{y}\|_0 \leq s$ by the lower semi-continuity of $\|\cdot\|_0$. □

We now analyze the stationarity properties of the limit points generated by the algorithm. To this aim, let us define the index set $I(x)$ of the largest active variables (in absolute value) at a generic point $x \in \mathbb{R}^n$, satisfying the following properties:

$$\begin{aligned} I(x) \in \underset{S \subseteq \{1,\dots,n\}}{\arg\max} & |S| \\ \text{s.t.} \quad & |S| \leq s, \\ & i \in S \Rightarrow x_i \neq 0, \\ & |x_i| \geq |x_j| \quad \forall\, i \in S, \forall\, j \notin S. \end{aligned} \tag{3.9}$$

For simplicity, in the following we will assume that problem (3.9) provides a unique definition of I. In practice, this assumption can be assured by explicitly defining how ambiguous cases should be handled. Also, note that $I(x) = \{i \in \{1,\dots,n\} \mid x_i \neq 0\}$ if $\|x\|_0 \leq s$.

Proposition 3.2.4. *Let $\{x^k, y^k\}$ be the sequence generated by Algorithm 4. Suppose that $(\bar{x}, \bar{y})$ is a limit point of $\{x^{k+1}, y^{k+1}\}$, i.e. $(x^{k+1}, y^{k+1}) \to_K (\bar{x}, \bar{y})$. Then $\bar{x}$ satisfies Lu-Zhang conditions for problem (3.2). Moreover, if there exists $K_1 \subset K$ s.t. $\|x^{k+1}\|_0 = \|\bar{x}\|_0$ for all $k \in K_1$ or $\|x^{k+1}\|_0 < s$ for all $k \in K_1$, $\bar{x}$ is BF-stationary for problem (3.2).*

Proof. From Proposition 3.2.3, $\bar{x} = \bar{y}$. Moreover, from the instructions of the algorithm, at each iteration k we have that

$$\|\nabla f(x^{k+1}) + \tau_k(x^{k+1} - y^{k+1})\| \leq \varepsilon_k$$

and

$$y^{k+1} = \underset{y \in Y}{\arg\min}\, q_{\tau_k}(x^{k+1}, y).$$

It is easy to see that

$$y_i^{k+1} = x_i^{k+1} \quad \text{for } i \in I(x^{k+1}),$$
$$y_i^{k+1} = 0 \quad \text{for } i \in \bar{I}(x^{k+1}).$$

Recalling the BF necessary optimality conditions, we have that

$$\nabla_{y_{I(x^{k+1})}} q_{\tau_k}(x^{k+1}, y^{k+1}) = 0 \quad \text{if } |I(x^{k+1})| = s,$$
$$\nabla_y q_{\tau_k}(x^{k+1}, y^{k+1}) = 0 \quad \text{if } |I(x^{k+1})| < s,$$

i.e.,

$$-\tau_k(x_i^{k+1} - y_i^{k+1}) = 0 \begin{cases} \forall\, i \in I(x^{k+1}) & \text{if } |I(x^{k+1})| = s, \\ \forall\, i \in \{1, \dots, n\} & \text{if } |I(x^{k+1})| < s. \end{cases} \tag{3.10}$$

There are finitely many possible sets $I(x^{k+1})$, therefore at least one of them is repeated infinitely on K. Thus, let us assume that $K_1 \subseteq K$ is such that $I(x^{k+1}) = I$ for all $k \in K_1$. From the instructions of the algorithm we have that

$$\|\nabla f(x^{k+1}) + \tau_k(x^{k+1} - y^{k+1})\| \to 0. \tag{3.11}$$

Now, let $I^\star = I(\bar{x})$ and let us address the relation between I and $I^\star$. The case $|I| < |I^\star|$ is impossible, since $(x^{k+1}, y^{k+1}) \to_{K_1} (\bar{x}, \bar{y})$ and so, if $|\bar{y}_i| > 0$, then $|y_i^{k+1}| > 0$ for k sufficiently large. Therefore, $|I| \geq |I^\star|$. Moreover, $I \supseteq I^\star$, otherwise there would exist i such that $y_i^{k+1} = 0$ for all $k \in K$ and $\bar{y}_i \neq 0$, which is absurd by similar reasonings as above.

Hence, we have the following possible cases:

(i) $|I| = s$, $I = I^\star$;

(ii) $|I| < s$;

(iii) $|I| = s$, $I \supset I^\star$.

We will address these three cases one at a time:

(i) Let $i \in I = I^\star$; from (3.11) we have

$$\nabla_i f(x^{k+1}) + \tau_k(x_i^{k+1} - y_i^{k+1}) \to_{K_1} 0,$$

while from (3.10) it follows that

$$\tau_k(x_i^{k+1} - y_i^{k+1}) = 0 \quad \forall\, k \in K_1.$$

Therefore,

$$\nabla_i f(x^{k+1}) \to_{K_1} \nabla_i f(\bar{x}) = 0 \quad \forall\, i \in I^\star,$$

i.e., BF conditions hold and hence Lu-Zhang hold too.

(ii) Let $i \in \{1, \dots, n\}$; similarly to the previous case we have that

$$\nabla_i f(x^{k+1}) + \tau_k (x^{k+1} - y^{k+1}) \to_{K_1} 0$$

and

$$\tau_k (x_i^{k+1} - y_i^{k+1}) = 0 \quad \forall\, k \in K_1,$$

from which we can derive

$$\nabla_i f(x^{k+1}) \to_{K_1} \nabla_i f(\bar{x}) = 0 \quad \forall\, i \in \{1, \dots, n\},$$

i.e., BF conditions, and consequently Lu-Zhang conditions, hold.

(iii) Let $i \in I$. By similar reasonings as in case 1., we can derive

$$\nabla_i f(x^{k+1}) \to_{K_1} \nabla_i f(\bar{x}) = 0.$$

Therefore Lu-Zhang conditions are satisfied selecting the index set I.

Putting everything together, we have from (i), (ii) and (iii) that Lu-Zhang conditions are always satisfied, while from (i) and (ii) we can assert BF-stationarity under the assumption of the proposition. □

As we can see, the proposed inexact version of the algorithm enjoys the same convergence properties as the original exact scheme. We also provided a better characterization of the algorithm, showing that the limit points are BF-stationary if none of the components goes to zero asymptotically.

A real case where the PD method is employed in practice on cardinality constrained problems with nonconvex objective is that of Neural Networks compression. In fact, it has been shown in Carreira-Perpinan and Idelbayev (2018) that the PD method provides in practice great results even if the x minimization step is performed by a surrogate of the arg min operator. There is however no theoretical analysis supporting the quality of the reported results. Employing our inexact procedure would yield such theoretical support. Moreover, the convergence analysis we conducted somehow hints why local minimization was sufficient in practice for the PD algorithm to work.

3.3 A Derivative-Free extension

First order information about the objective function is fundamental for the PD methods we have considered thus far. However, there are applications where gradients are not available. Such lack of information has an impact for the applicability of Algorithm 4, particularly w.r.t. the x update step and the inner stopping criterion.

In this section, we propose a derivative-free modification of Algorithm 4 that, similarly to Fasano et al. (2014); Liuzzi et al. (2016); Lucidi and Sciandrone (2002), updates x by line search steps along the coordinate axes and employs a stopping criterion based on the length of such steps.

The derivative-free PD method is described in Algorithm 5. In the x update step, we employ as search directions the coordinate and anti-coordinate directions. A tentative step length $\tilde{\alpha}_i$ is associated to each of these directions. At every iteration, all search directions are considered one at a time; a derivative-free line search is performed along each direction, according to Algorithm 6. If the tentative step size does not provide a sufficient decrease, il will be reduced for the next iteration. If on the other hand the tentative step size is of sufficient decrease, an extrapolation procedure is carried out; the tentative step size for that same direction at the successive iteration will be the longest one tried in the extrapolation phase that provides a sufficient decrease. That same step length is also used to move along the considered direction, provided that it is at least as large as ε_k; otherwise, no movement is done along the direction. The inner loop then stops when all tentative step sizes have become smaller than ε_k.

Hereafter, we show that Algorithm 5 enjoys the same convergence properties as its "smooth" counterpart. First, we prove that the line search procedure does not loop infinitely inside our procedure.

Proposition 3.3.1. *Assume the objective function f is coercive. Then, Algorithm 6 terminates in a finite number of iterations.*

Proof. In order to prove the thesis, we have to show that Algorithm 6 does not loop infinitely between steps 5-8. Assume by contradiction that this is not the case. Then, from the instructions of the algorithm, we have

$$f(x + \sigma^k \alpha_0 d) \leq f(x) - \gamma \sigma^{2k} \alpha_0^2 \|d\|^2$$

for any $k \in \mathbb{N}$. From the continuity and the coercivity of f, we know that it is bounded below. We also have $\gamma \sigma^{2k} \alpha_0^2 \to \infty$. Thus

$$f(x + \sigma^k \alpha_0 d) \leq f(x) - \gamma \sigma^{2k} \alpha_0^2 \|d\|^2 \to -\infty,$$

which violates the boundedness of f. □

Note that q_{τ_k} satisfies the assumptions of Proposition 3.3.1 for every τ_k. We now show that the inner loop of Algorithm 5 also terminates in finite time.

Proposition 3.3.2. *Algorithm 5 does not loop infinitely between steps 18-37.*

Algorithm 5: Derivative-Free Inexact Penalty Decomposition

Input: $\tau_0 > 0, \theta > 1, \delta \in (0,1), x^0 = y^0 \in \mathbb{R}^n$ s.t. $\|x^0\|_0 \leq s$, a sequence $\{\varepsilon_k\}$ s.t. $\varepsilon_k \to 0$, $\mathcal{D} = \{d_1, \ldots, d_{2n}\} = \{e_1, \ldots, e_n, -e_1, \ldots, -e_n\}$.
for $k = 0, 1, \ldots$ **do**
  $\tilde{\alpha}^0 = e \in \mathbb{R}^{2n}$
  $\ell = 0$
  for $i = 1, \ldots, 2n$ **do**
    $\hat{\alpha}_i = \texttt{LineSearch}(q_{\tau_k}(x, y^k), d_i, 1, x^k)$
    if $\hat{\alpha}_i > \varepsilon_k$ **then**
      $x_{\text{trial}} = x^k + \hat{\alpha}_i d_i$
      break
    end
  end
  if $q_{\tau_k}(x_{trial}, y^k) \leq f(x^0)$ **then**
    $u^0, v^0 = x^k, y^k$
  end
  else
    $u^0, v^0 = x^0, y^0$
  end
  while $\max_{i=1,\ldots,2n} \{\tilde{\alpha}_i^\ell\} > \varepsilon_k$ **do**
    $u^\ell(0) = u^\ell$
    for $i = 1, \ldots, 2n$ **do**
      $\alpha_i^\ell = \texttt{LineSearch}(q_{\tau_k}(u, v^\ell), d_i, \tilde{\alpha}_i^\ell, u^\ell(i-1))$
      if $\alpha_i^\ell = 0$ **then**
        $\tilde{\alpha}_i^{\ell+1} = \delta \tilde{\alpha}_i^\ell$
      end
      else
        $\tilde{\alpha}_i^{\ell+1} = \alpha_i^\ell$
      end
      if $\alpha_i^\ell > \varepsilon_k$ **then**
        $u^\ell(i) = u^\ell(i-1) + \alpha_i^\ell d_i^\ell$
      end
      else
        $u^\ell(i) = u^\ell(i-1)$
      end
    end
    $u^{\ell+1} = u^\ell(2n)$
    $v^{\ell+1} = \arg\min_{v \in Y} q_{\tau_k}(u^{\ell+1}, v)$
    $\ell = \ell + 1$
  end
  $\tau_{k+1} = \theta \tau_k$
  $x^{k+1} = u^\ell$
  $y^{k+1} = v^\ell$
end
Output: The sequence $\{x^k\}$.

Algorithm 6: LineSearch

Input: $f : \mathbb{R}^n \to \mathbb{R}$, $d \in \mathbb{R}^n$, $\alpha_0 \in \mathbb{R}^+$, $x \in \mathbb{R}^n$, $\gamma \in (0,1)$, $\sigma > 1$.
$\alpha = \alpha_0$
if $f(x + \alpha d) \leq f(x) - \gamma\alpha^2\|d\|^2$ **then**
  Let $\beta = \alpha$;
  repeat
    Set $\alpha = \beta$;
    Set $\beta = \sigma\alpha$;
  until $f(x + \beta d) > f(x) - \gamma\beta^2\|d\|^2$;
  return α;
end
Set $\alpha = 0$;
return α

Proof. Assume by contradiction that the algorithm loops infinitely. Then, for all $\ell = 0, 1, \ldots$, there exists $i \in \{1, \ldots, n\}$ such that $\tilde{\alpha}_i^\ell > \varepsilon_k$. Moreover, from the instructions of the algorithm, we have

$$q_{\tau_k}(u^{\ell+1}, v^{\ell+1}) \leq q_{\tau_k}(u^{\ell+1}, v^\ell) \leq q_{\tau_k}(u^\ell(i), v^\ell) \leq q_{\tau_k}(u^\ell(i-1), v^\ell) \leq q_{\tau_k}(u^\ell, v^\ell),$$

i.e., $\{q_{\tau_k}(u^\ell, v^\ell)\}$ is a non-increasing sequence. It thus has limit $\bar{q}$, which is finite from the coercivity of q_{τ_k}.

Consider now the sequence $\{\tilde{\alpha}_i^\ell\}$ for $i \in \{1, \ldots, n\}$. We can split the sequence of iterations $\{0, 1, \ldots\}$ into two subsequences K_1 and K_2, where for all $\ell \in K_1$, recalling that $\|d_i\| = \| \pm e_i\| = 1$, it holds

$$\begin{gathered} q_{\tau_k}(u^\ell(i-1) + \tilde{\alpha}_i^\ell \sigma^s d_i^\ell, v^\ell) \leq q_{\tau_k}(u^\ell(i-1), v^\ell) - \gamma(\tilde{\alpha}_i^\ell \sigma^s)^2, \\ \tilde{\alpha}_i^{\ell+1} = \alpha_i^\ell = \tilde{\alpha}_i^\ell \sigma^s, \end{gathered}$$

for some $s \in \mathbb{N}$, while for all $\ell \in K_2$ we have

$$\tilde{\alpha}_i^{\ell+1} = \delta \tilde{\alpha}_i^\ell.$$

Consider the sequence K_1. For all $\ell \in K_1$ we have:

$$\begin{aligned} q_{\tau_k}(u^{\ell+1}, v^{\ell+1}) &\leq q_{\tau_k}(u^{\ell+1}, v^\ell) \\ &\leq q_{\tau_k}(u^\ell(i), v^\ell) \\ &\leq q_{\tau_k}(u^\ell(i-1), v^\ell) - \gamma(\tilde{\alpha}_i^\ell \sigma^s)^2 \\ &\leq q_{\tau_k}(u^\ell(0), v^\ell) - \gamma(\tilde{\alpha}_i^\ell)^2 \\ &= q_{\tau_k}(u^\ell, v^\ell) - \gamma(\tilde{\alpha}_i^\ell)^2. \end{aligned}$$

Assume that K_1 is infinite. Then, taking the limits for $\ell \to \infty$, $\ell \in K_1$, recalling that $q_{\tau_k}(u^\ell, v^\ell) \to \bar{q}$, we get

$$\bar{q} \leq \bar{q} - \gamma \lim_{\substack{\ell \to \infty \\ \ell \in K_1}} (\tilde{\alpha}_i^\ell)^2 \leq \bar{q},$$

which implies that

$$\tilde{\alpha}_i^\ell \to_{K_{11}} 0. \tag{3.12}$$

We have therefore that for $\ell \in K_1$ sufficiently large, $\tilde{\alpha}_i^\ell \leq \varepsilon_k$.

Consider now K_2 and assume it is infinite. If K_1 is finite, then there exists $\bar{\ell}$ such that $\ell \in K_2$ for all $\ell \geq \bar{\ell}$, which implies $\tilde{\alpha}_i^\ell = \delta^{\ell-\bar{\ell}} \tilde{\alpha}_i^{\bar{\ell}}$, i.e., $\tilde{\alpha}_i^\ell \leq \varepsilon_k$ for ℓ sufficiently large.

So, let us assume K_1 is also infinite. For all $\ell \in K_2$, let m_ℓ be the maximum index on $\{0, 1, \ldots\}$ such that $m_\ell \in K_1$, $m_\ell < \ell$ (m_ℓ is the index of the last iteration in K_1 preceding ℓ). Then it holds

$$\tilde{\alpha}_i^\ell = \delta^{\ell - m_\ell} \alpha_j^{m_\ell}. \tag{3.13}$$

Assume by contradiction that there exist infinite $\hat{\ell} \in K_1$ such that $\alpha_i^{\hat{\ell}} > \varepsilon_k$ (call this subsequence $\hat{K}$). For any of such indices, recalling previous reasonings, it holds

$$q_{\tau_k}\left(u^{\hat{\ell}+1}, v^{\hat{\ell}+1}\right) \leq q_{\tau_k}\left(u^{\hat{\ell}}, v^{\hat{\ell}}\right) - \gamma \left(\alpha_i^{\hat{\ell}}\right)^2 < q_{\tau_k}\left(u^{\hat{\ell}}, v^{\hat{\ell}}\right) - \gamma \varepsilon_k^2.$$

Taking the limits for $\hat{\ell} \in \hat{K}$, recalling that $q_{\tau_k}(u^\ell, v^\ell)$ has limit $\bar{q}$, we get

$$\bar{q} \leq \bar{q} - \gamma \varepsilon_k^2 < \bar{q},$$

which is absurd.

Then, for $\ell \in K_2$ sufficiently large, we have that $\alpha_i^{m_\ell} \leq \varepsilon_k$, which, together with (3.13), implies again that $\tilde{\alpha}_i^\ell \leq \varepsilon_k$.

Combining the results obtained on K_1 and K_2 and recalling that i is arbitrary we get

$$\max_{i=1,\ldots,n} \{\tilde{\alpha}_i^\ell\} \leq \varepsilon_k$$

for ℓ sufficiently large, a contradiction. □

Next, we prove a technical result concerning the directional derivatives of q_{τ_k} with respect to the variables x_i.

Proposition 3.3.3. *Assume for all k that $\tilde{\alpha}_i^0 > \varepsilon_k$ for all $i = 1, \ldots, 2n$. Then, for all $k = 1, 2, \ldots$ and for all $i = 1, \ldots, 2n$ there exists $\rho_i^k \in (0, c\varepsilon_k)$ such that*

$$\nabla_x q_{\tau_k}(x^{k+1} + \rho_i^k d_i, y^{k+1})^T d_i > -c\varepsilon_k,$$

with $c = \max\{\sigma, 1/\delta\}$.

Proof. Let ℓ be the index of the last inner iteration. By definition of ℓ, we must have that $\tilde{\alpha}_i^{\ell+1} \leq \varepsilon_k$ for all $i = 1, \ldots, n$. From the instructions of the algorithm this implies that we have $u^{\ell+1} = u^{\ell}(2n) = \ldots = u^{\ell}(0) = u^{\ell}$ and consequently $v^{\ell+1} = v^{\ell}$. Consider any $i \in \{1, \ldots, 2n\}$. We have to cases:

- $\tilde{\alpha}_i^{\ell+1} = \delta\tilde{\alpha}_i^{\ell}$; in this case, $\tilde{\alpha}_i^{\ell}$ did not satisfy the sufficient decrease condition in the `LineSearch` procedure, i.e.

$$q_{\tau_k}(u^{\ell} + \tilde{\alpha}_i^{\ell} d_i, v^{\ell}) - q_{\tau_k}(u^{\ell}, v^{\ell}) > -\gamma(\tilde{\alpha}_i^{\ell})^2. \tag{3.14}$$

  From the Mean Value Theorem, we can write:

$$q_{\tau_k}(u^{\ell} + \tilde{\alpha}_i^{\ell} d_i, v^{\ell}) - q_{\tau_k}(u^{\ell}, v^{\ell}) = \tilde{\alpha}_i^{\ell} \nabla_x q_{\tau_k}(u^{\ell} + \rho_i^{\ell} d_i, v^{\ell})^T d_i, \tag{3.15}$$

  for some $\rho_i^{\ell} \in (0, \tilde{\alpha}_i^{\ell})$. From (3.14), (3.15), it follows:

$$\nabla_x q_{\tau_k}(u^{\ell} + \rho_i^{\ell} d_i, v^{\ell})^T d_i > -\gamma\tilde{\alpha}_i^{\ell} = -\frac{\gamma}{\delta}\tilde{\alpha}_i^{\ell+1} \geq -\frac{\gamma}{\delta}\varepsilon_k,$$

  where $\rho_i^{\ell} \in (0, \varepsilon_k/\delta)$.

- $\tilde{\alpha}_i^{\ell+1} = \alpha_i^{\ell}$; we know, from the instructions of the LineSearch procedure, that

$$q_{\tau_k}(u^{\ell} + \sigma\alpha_i^{\ell} d_i, v^{\ell}) - q_{\tau_k}(u^{\ell}, v^{\ell}) > -\gamma(\sigma\alpha_i^{\ell})^2. \tag{3.16}$$

  From the Mean Value Theorem, we can write:

$$q_{\tau_k}(u^{\ell} + \sigma\alpha_i^{\ell} d_i, v^{\ell}) - q_{\tau_k}(u^{\ell}, v^{\ell}) = \sigma\alpha_i^{\ell} \nabla_x q_{\tau_k}(u^{\ell} + \rho_i^{\ell} d_i, v^{\ell})^T d_i, \tag{3.17}$$

  for some $\rho_i^{\ell} \in (0, \sigma\alpha_i^{\ell})$. From (3.16),(3.17), it follows:

$$\nabla_x q_{\tau_k}(u^{\ell} + \rho_i^{\ell} d_i, v^{\ell})^T d_i > -\gamma\sigma\alpha_i^{\ell} = -\gamma\sigma\tilde{\alpha}_i^{\ell+1} \geq -\gamma\sigma\varepsilon_k,$$

  where $\rho_i^{\ell} \in (0, \sigma\varepsilon_k)$.

In both cases we thus have

$$\nabla_x q_{\tau_k}(u^{\ell} + \rho_i^{\ell} d_i, v^{\ell})^T d_i > -c\varepsilon_k \tag{3.18}$$

for some $\rho_i^{\ell} \in (0, c\varepsilon_k)$, $c = \max\{\sigma, 1/\delta\}$.

Since $\tilde{\alpha}_i^{\ell+1} \leq \varepsilon_k$ for all $i = 1, \ldots, 2n$, from the instructions of the algorithm we have $u^{\ell+1} = u^{\ell}$ and consequently $v^{\ell+1} = v^{\ell}$. Hence, equation (3.18) holds for (x^{k+1}, y^{k+1}). □

We are finally able to show that the gradients of the penalty function w.r.t. x go to zero in the limit.

Proposition 3.3.4. *Let $\{x^{k+1}, y^{k+1}\}$ be the sequence generated by Algorithm 5. Then*

$$\lim_{k\to\infty} \|\nabla_x q_{\tau_k}(x^{k+1}, y^{k+1})\| = 0. \tag{3.19}$$

Proof. From Proposition 3.3.3, we know that for all $k = 1, 2, \ldots$ and for all $i = 1, \ldots, 2n$ there exists $\rho_i^k, \in (0, c\varepsilon_k)$ such that

$$\nabla_x q_{\tau_k}(x^{k+1} + \rho_i^k d_i, y^{k+1})^T d_i > -c\varepsilon_k,$$

with $c = \max\{\sigma, 1/\delta\}$. Since $\mathcal{D} = \{d_1, \ldots, d_{2n}\} = \{e_1, \ldots e_n, -e_1, \ldots, -e_n\}$, we have in particular that, for all $i = 1, \ldots, n$

$$\begin{aligned}\nabla_x q_{\tau_k}(x^{k+1} + \rho_i^k e_i, y^{k+1})^T e_i &> -c\varepsilon_k,\\ -\nabla_x q_{\tau_k}(x^{k+1} - \rho_{i+n}^k e_i, y^{k+1})^T e_i &> -c\varepsilon_k\end{aligned}$$

Taking limits for $k \to \infty$, recalling that $\varepsilon_k \to 0$ and $\rho_i^k \in (0, c\varepsilon_k)$, we get

$$\begin{aligned}\lim_{k\to\infty} \nabla_x q_{\tau_k}(x^{k+1} + \rho_i^k e_i, y^{k+1})^T e_i &= \lim_{k\to\infty} \nabla_x q_{\tau_k}(x^{k+1}, y^{k+1})^T e_i \geq 0\\ \lim_{k\to\infty} \nabla_x q_{\tau_k}(x^{k+1} - \rho_{i+n}^k e_i, y^{k+1})^T e_i &= \lim_{k\to\infty} \nabla_x q_{\tau_k}(x^{k+1}, y^{k+1})^T e_i \leq 0\end{aligned}$$

from which we get the thesis. □

With Proposition 3.3.4 at hand, the rest of the convergence analysis can be derived almost exactly following the same reasonings as in the "smooth" case. We have therefore that the following statements hold.

Proposition 3.3.5. *Let $\{x^k, y^k\}$ be the sequence generated by Algorithm 5. Then $\{x^k, y^k\}$ admits cluster points.*

Proposition 3.3.6. *Let $\{x^k, y^k\}$ be the sequence generated by Algorithm 5. Suppose that $(\bar{x}, \bar{y})$ is a limit point of $\{x^k, y^k\}$, i.e. $(x^k, y^k) \to_K (\bar{x}, \bar{y})$. Then, $(\bar{x}, \bar{y})$ is a feasible point for problem* (3.3)*, and $\bar{x}$ is feasible for problem* (3.2)*.*

Proposition 3.3.7. *Let $\{x^k, y^k\}$ be the sequence generated by Algorithm 5. Suppose that $(\bar{x}, \bar{y})$ is a limit point of $\{x^{k+1}, y^{k+1}\}$, i.e. $(x^{k+1}, y^{k+1}) \to_K (\bar{x}, \bar{y})$. Then $\bar{x}$ satisfies Lu-Zhang conditions for problem* (3.2)*. Moreover, if there exists $K_1 \subset K$ s.t. $\|x^{k+1}\|_0 = \|\bar{x}\|_0$ for all $k \in K_1$ or $\|x^{k+1}\|_0 < s$ for all $k \in K_1$, $\bar{x}$ is BF-stationary for problem* (3.2)*.*

3.4 Conclusions

In this chapter we introduced an inexact version of the Penalty Decomposition method, based on line searches, for cardinality constrained problems which is applicable in

practice even when the objective function is nonconvex. We proved that the proposed procedure has the same convergence guarantees as the original scheme that requires exact minimization steps, and we also provide additional conditions to guarantee the convergence to a stronger necessary optimality condition.

We also proposed a derivative-free extension, aimed at solving optimization problems without the use of first order information. The approach is based on a suitable search scheme along the coordinate axes, using standard derivative-free line searches. The convergence analysis carried out shows that this extension enjoys the same convergence guarantees as the smooth case.

Chapter 4

A concave optimization-based approach for sparse multiobjective programming

In this chapter we consider sparse multiobjective optimization problems, i.e., problems where one of the objectives is the ℓ_0-norm. Formally, we consider problems of the form

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} (f_1(x), f_2(x), \ldots, f_{m-1}(x), ||x||_0) \\ &s.t.\ x \in X \end{aligned} \tag{SMOP}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$, $m \geq 2$ and $X \subset \mathbb{R}^n$ is a compact convex set.

To deal with the difficulties brought by the use of the ℓ_0-norm, following the approach of Rinaldi et al. (2010), we propose to replace the ℓ_0-norm objective by means of smooth concave approximating functions. In this way, we are able to convert (SMOP) into a smooth problem. Further, we prove that such an approximating problem enjoys some nice equivalence properties with respect to the original combinatorial one. Then, we define an optimization algorithm based on the steepest descent framework for smooth multiobjective optimization Fliege and Svaiter (2000); Fukuda and Drummond (2014).

In the following, we will denote by $F : \mathbb{R}^n \to \mathbb{R}^m$ the vector-valued function defined by

$$F(x) \stackrel{\triangle}{=} (f_1(x), f_2(x), \ldots, f_{m-1}(x), \|x\|_0).$$

Given any two vectors $u, v \in \mathbb{R}^p$, we write

$$\begin{aligned} u < v &\Leftrightarrow u_i < v_i, \text{ for all } i = 1, \ldots, p \\ u \leqq v &\Leftrightarrow u_i \leq v_i, \text{ for all } i = 1, \ldots, p \\ u \leq v &\Leftrightarrow u \leqq v \text{ and } u \neq v. \end{aligned}$$

4.1 Preliminaries

In the context of multiobjective optimization, the notion of optimality used in the single objective case is not adequate. To this end, we define the concepts of Pareto dominance and Pareto optimality, as stated below.

Definition 4.1.1 (Pareto dominance)**.** Given two vectors $x, y \in \mathbb{R}^n$, we say that x (strictly) Pareto dominates y when

$$F(x) \leqq F(y) \ \big(F(x) \leq F(y)\big).$$

Ideally, a solution $x^* \in X$ of problem (SMOP) would be a point such that x^* Pareto dominates each other feasible point $x \in X$, i.e.,

$$F(x^*) \leqq F(x), \qquad \text{for all } x \in X.$$

Unfortunately, such a point x^* very seldom exists. Therefore, we usually consider the following definitions of optimality.

Definition 4.1.2 (Weak Pareto optimality)**.** A point $x^* \in X$ is a weak Pareto optimal point for problem (SMOP), if there does not exist any $x \in X$ such that

$$F(x) < F(x^*).$$

Definition 4.1.3 (Pareto optimality)**.** A point $x^* \in X$ is a Pareto optimal point for problem (SMOP), if there does not exist any $x \in X$ such that

$$F(x) \leq F(x^*).$$

Using these definitions, our goal is to identify the so-called Pareto front of the problem, i.e., the set of non-dominated points.

The steepest descent framework for smooth multiobjective optimization

Optimization algorithms are usually defined with reference to some definition of stationarity, i.e, necessary optimality conditions useful to identify "good" points. In the context of multiobjective optimization, a common stationarity condition employed in practice is the concept of *Pareto-stationarity*. First, let us consider the problem

$$\begin{aligned} &\min \ F_s(x) = (f_1(x), \ldots, f_m(x)) \\ &s.t. \ x \in X, \end{aligned} \tag{4.1}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$ are continuously differentiable functions and X is a convex, compact set.

Definition 4.1.4 (Pareto stationarity)**.** A point $x^* \in X$ is Pareto stationary for problem (4.1) if, for all $y \in X$, an index $j \in \{1, \ldots, m\}$ exists such that

$$\nabla f_j(x^*)^\top (y - x^*) \geq 0.$$

It can be easily shown that, if x^* is a Pareto optimal point for problem (4.1), then x^* is Pareto stationary for problem (4.1). In other words, Pareto-stationarity is in fact a necessary optimality condition. The inverse implication does not hold in general, but it can be proven for particular cases like, e.g., when F_s is a strictly convex continuously differentiable map.

Thanks to Definition 4.1.4, when $\bar{x} \in X$ is not a Pareto stationary point, we know that a $y \in X$ must exist such that $v = y - \bar{x}$ is a descent direction for all the objective functions f_i, $i = 1, \ldots, m$, at $\bar{x}$. This observation leads quite naturally to the definition of an iterative optimization scheme, as we now explain in detail.

First, for any given $x \in X$, we define the function $g_x : X \to \mathbb{R}$ by

$$g_x(y) = \max_{i=1,\ldots,m} \nabla f_i(x)^\top (y - x).$$

Note that g_x is continuous, piecewise linear, and convex. By the compactness of X, g_x admits a global minimum on X; hence, we denote respectively by $\theta(x)$ and $y(x)$ the global minimum value and a global minimum point of g_x over X, i.e.,

$$\theta(x) = \min_{y \in X} g_x(y) \tag{4.2}$$

$$y(x) \in \arg\min_{y \in X} g_x(y). \tag{4.3}$$

Problem (4.2) is a finite *minimax* problem with linear component functions. It can thus be trivially restated as the following problem:

$$\begin{aligned} &\min_{y,\beta} \ \beta \\ &\nabla f_i(x)^\top (y - x) \leq \beta, \quad i = 1, \ldots, m, \\ &y \in X. \end{aligned} \tag{4.4}$$

Furthermore, from Fliege and Svaiter (2000), we report the following proposition.

Proposition 4.1.1. *Given problem (4.1), let $\theta : X \to \mathbb{R}$ be defined as in (4.2). Then the following statements hold:*

- *θ is a continuous function;*
- *$\theta(x) \leq 0$, for all $x \in X$;*
- *$x^* \in X$ is Pareto stationary for problem (4.1) if and only if $\theta(x^*) = 0$.*

We now introduce the steepest descent direction for the vector valued mapping F_s at x.

Definition 4.1.5 (Steepest descent direction)**.** Given any point $x \in X$, the steepest descent direction for F_s at x is

$$v(x) = y(x) - x,$$

where $y(x)$ is given by (4.3).

The idea now is quite simple: as long as our current point is not stationary, we compute the steepest descent direction and we search for a better point along this direction. The algorithm, which we formally define next, was originally proposed and analyzed in Fliege and Svaiter (2000) ($J(x)$ denotes the Jacobian of the vector of objective functions).

Algorithm 7: Steepest Descent Algorithm (SDA)

Data: $x_0 \in X$, $\gamma \in (0,1)$
Compute $\theta(x_0)$, $v(x_0)$ and set $k = 0$
while $\theta(x_k) < 0$ **do**
  Compute $\alpha_k = 2^{-\beta_k}$ with β_k the smallest non-negative integer s.t.

$$F_s(x_k + \alpha_k v(x_k)) \leq F_s(x_k) + \gamma \alpha_k J(x_k) v(x_k).$$

  Set $x_{k+1} = x_k + \alpha_k v(x_k)$, and $k = k + 1$
  Compute $\theta(x_k)$ and $v(x_k)$
end while

4.2 Concave approximations

The steepest descent framework we defined in the previous section is based on the computation of the gradients of the objective functions. In other words, we need a smooth problem to be able to use it.

To this end, we consider smooth concave approximating functions to convert (SMOP) into a smooth problem. This approach is motivated by the fact that

$$\|x\|_0 = \sum_{i=1}^{n} s(|x_i|),$$

where $s : \mathbb{R} \to \mathbb{R}^+$ is the step function such that $s(t) = 1$ for $t > 0$ and $s(t) = 0$ for $t \leq 0$. The idea is then to replace the discontinuous step function with a continuously differentiable concave function.

The basic idea of replacing the ℓ_0-norm with a concave functions is well known in the literature, and can be traced back to Mangasarian (1996). Here, as done by Rinaldi et al. (2010); Di Lorenzo et al. (2012) in the single objective case, we study the equivalence between the concave smooth formulation and the original nonsmooth problem.

More in detail, we first rewrite problem (SMOP) into the following equivalent program:

$$\begin{aligned} \min_{x,y} \quad & (f_1(x), f_2(x), \ldots, f_{m-1}(x), ||y||_0) \\ s.t. \quad & x \in X, \\ & -y \leq x \leq y, \end{aligned} \tag{4.5}$$

where inequality $-y \leq x \leq y$ is intended component-wise, i.e. $-y_i \leq x_i \leq y_i$, $i = 1, \ldots, n$. Secondly, we prove a set of equivalence results between problem (4.5) and a problem of the form

$$\begin{aligned} \min \; & F(x,y) = \left[f_1(x), \ldots, f_{m-1}(x), \sum_{i=1}^{n} f^u(y_i) \right] \\ s.t. \; & x \in X, \\ & -y \leq x \leq y. \end{aligned} \tag{4.6}$$

where $f^u : \mathbb{R}^+ \to \mathbb{R}$ is a smooth function depending on a parameter $u \in U \subseteq \mathbb{R}$.

Note that the added variables y and the related constraint $-y \leq x \leq y$ is a modeling trick to remove the absolute value from the original concave approximation term $\sum_{i=1}^{n} f^u(|x_i|)$.

We now introduce the following assumption on the parameterized function f^u.

Assumption 4.2.1. *There exists $\bar{u} \in U$ such that, for any infinite sequence $\{u_k\} \to \bar{u}$ we have that:*

(i) for each $y_i \geq 0$, $\lim_{k\to\infty} f^{u_k}(y_i)$ is well defined;

(ii) for each $y_i > 0$, it follows $f^{u_k}(0) < f^{u_k}(y_i)$ and

$$\lim_{k\to\infty} f^{u_k}(0) < \lim_{k\to\infty} f^{u_k}(y_i) < \infty;$$

(iii) for any $\bar{y}_i > 0$, and for any sequence $\{y_i^k\} \to \bar{y}_i$, we have

$$\lim_{k\to\infty} f^{u_k}(y_i^k) = \lim_{k\to\infty} f^{u_k}(\bar{y}_i);$$

(iv) for each $y_i \geq 0$, one of the following conditions holds: either

$$\lim_{k\to\infty} f^{u_k}(y_i) = \begin{cases} 1 & \text{if } y_i > 0, \\ 0 & \text{if } y_i = 0, \end{cases} \tag{4.7}$$

or

$$\lim_{k\to\infty} f^{u_k}(0) = -\infty. \tag{4.8}$$

It can be shown that, when $U = \mathbb{R}^+$, Assumption 4.2.1 is satisfied, for instance:

- by $f^u(y_i) = 1 - e^{-uy_i}$, with $\bar{u} = +\infty$, which satisfies condition (4.7);
- by $f^u(y_i) = \log(u + y_i)$, with $\bar{u} = 0$, which satisfies condition (4.8).

In particular, we note that, whenever condition (4.7) holds, it results

$$\lim_{k\to\infty} \sum_{i=1}^{n} f^{u_k}(y_i) = \|y\|_0. \tag{4.9}$$

Now, concerning the connections between problem (4.6) and problem (SMOP), the following statement holds.

Proposition 4.2.1. *Let $\{u_k\}$ be a sequence such that $\lim_{k\to\infty} u_k = \bar{u}$ and let $\{(x^k, y^k)\}$ be a sequence such that (x^k, y^k) is weakly Pareto optimal for problem (4.6) with $u = u^k$, and $y^k = |x^k|$. Then, $\{(x^k, y^k)\}$ has limit points and every limit point $(\bar{x}, \bar{y})$ is such that $\bar{x}$ is weakly Pareto optimal for problem (SMOP).*

Proof. By the assumptions, for all k it results $x^k \in X$ and $y^k = |x^k|$. Hence, as X is compact, the sequence $\{(x^k, y^k)\}$ admits limit points. We proceed by contradiction and assume that there exists an infinite subset $K \subseteq \{1, 2, \ldots\}$ such that

$$\lim_{k\in K, k\to\infty} (x^k, y^k) = (\bar{x}, \bar{y}),$$

and $\bar{x}$ is not weakly Pareto optimal for problem (SMOP). Then, there must exist a point $v \in X$ such that

$$F(v) < F(\bar{x}). \tag{4.10}$$

Recalling the continuity of functions $f_1, \ldots, f_{m-1}$, we get that, for $k \in K$ sufficiently large, the following inequalities hold:

$$f_i(v) < f_i(x^k) \qquad i = 1, \ldots, m-1. \tag{4.11}$$

Moreover, from (4.10), recalling that $\bar{y} = |\bar{x}|$, it follows that

$$||v||_0 < ||\bar{x}||_0 = ||\bar{y}||_0. \tag{4.12}$$

Let y^v be such that $y^v = |v|$. Then, (v, y^v) is feasible for problem (4.6). Therefore, recalling that (x^k, y^k) is weakly Pareto optimal for problem (4.6) and taking (4.11) into account, for $k \in K$ sufficiently large we must have that

$$\sum_{i=1}^{n} f^{u_k}(y_i^k) \leq \sum_{i=1}^{n} f^{u_k}(y_i^v) = \sum_{i=1}^{n} f^{u_k}(|v_i|). \tag{4.13}$$

Consider any $i \in \{1, \dots, n\}$ such that $\bar{y}_i > 0$. From assumption (iii), it follows that

$$\lim_{k\to\infty} f^{u_k}(\bar{y}_i) = \lim_{k\to\infty} f^{u_k}(y_i^k) = l_i. \tag{4.14}$$

Then, given any positive ϵ such that $n\epsilon < 1$, two positive integers $k_1(\epsilon)$ and $k_2(\epsilon)$ exist such that

$$f^{u_k}(\bar{y}_i) \le l_i + \frac{\epsilon}{2}, \quad \text{for all } k \ge k_1(\epsilon),$$
$$f^{u_k}(y_i^k) \ge l_i - \frac{\epsilon}{2}, \quad \text{for all } k \ge k_2(\epsilon).$$

Thus, for k sufficiently large, we obtain

$$f^{u_k}(\bar{y}_i) \le f^{u_k}(y_i^k) + \epsilon. \tag{4.15}$$

Now, let us consider any index $i \in \{1, \dots, n\}$ such that $\bar{y}_i = 0$. Using assumption (ii), we have, for all k,

$$f^{u_k}(\bar{y}_i) \le f^{u_k}(y_i^k). \tag{4.16}$$

From (4.15) and (4.16), we get that for k sufficiently large, we can write

$$\sum_{i=1}^{n} f^{u_k}(\bar{y}_i) \le \sum_{i=1}^{n} f^{u_k}(y_i^k) + n\epsilon. \tag{4.17}$$

Condition (4.13) implies that

$$\sum_{i=1}^{n} f^{u_k}(\bar{y}_i) \le \sum_{i=1}^{n} f^{u_k}(y_i^v) + n\epsilon = \sum_{i=1}^{n} f^{u_k}(|v_i|) + n\epsilon. \tag{4.18}$$

Let us now distinguish two cases.

Case I: Suppose that condition (4.7) holds. Using (4.9), we have

$$\lim_{k\to\infty} \sum_{i=1}^{n} f^{u_k}(\bar{y}_i) = \|\bar{y}\|_0 = \|\bar{x}\|_0,$$

$$\lim_{k\to\infty} \sum_{i=1}^{n} f^{u_k}(y_i^v) = \|y^v\|_0 = \|v\|_0.$$

Hence, taking limits for $k \to \infty$ in (4.18), we obtain

$$\|\bar{y}\|_0 \le \|v\|_0 + n\epsilon.$$

From the above relation and (4.12), it follows

$$\|v\|_0 + 1 \le \|\bar{y}\|_0 \le \|v\|_0 + n\epsilon,$$

which contradicts the fact that $n\epsilon < 1$.

Case II: Suppose that condition (4.8) holds. First, we rewrite relation (4.18) as follows:

$$\sum_{\bar{y}_i>0} f^{u_k}(\bar{y}_i) + (n - \|\bar{y}\|_0) f^{u_k}(0) \leq \sum_{y_i^v>0} f^{u_k}(y_i^v) + (n - \|y^v\|_0) f^{u_k}(0) + n\epsilon,$$

from which we obtain

$$(\|y^v\|_0 - \|\bar{y}\|_0) f^{u_k}(0) \leq \sum_{y_i^v>0} f^{u_k}(y_i^v) - \sum_{\bar{y}_i>0} f^{u_k}(\bar{y}_i) + n\epsilon.$$

Taking limits for $k \to \infty$, using (4.12), the fact that $\|y^v\|_0 = \|v\|_0$ and condition (4.8), we get that the left member of the above relation tends to $+\infty$, while the right member tends to a finite value (see assumption (ii)), a contradiction. □

Remark 4.2.1. *In general, it turns out that weak Pareto optimality is the best we can aim for, even if we consider a sequence $\{(x^k, y^k)\}$ of Pareto points for the approximated problem, instead of the weak Pareto optimality used in Proposition* (4.2.1)*. To have a better understanding of this point, we provide the following example.*

Example 4.2.1. *Consider the following multiobjective optimization problem:*

$$\begin{aligned} &\min_{x \in \mathbb{R}} (-x, \|x\|_0) \\ &s.t. -1 \leq x \leq 1. \end{aligned}$$

It is easy to see that the only Pareto optimal points are $x = 0$ and $x = 1$, whose objective vectors are $(0, 0)$ and $(-1, 1)$, respectively.

Consider now the sequence of smooth approximating problems

$$\begin{aligned} &\min_{x,y} (-x, f^{u_k}(y)) \\ &s.t. -1 \leq x \leq 1, \\ &\quad -y \leq x \leq y, \end{aligned} \tag{P_{u^k}}$$

indexed by the sequence $\{u_k\}$. Consider, for example, the function $f^{u_k}(y) = 1 - e^{-u_k y}$. It is easy to see that, for any chosen value of $u_k > 0$, the point $(\frac{1}{2}, \frac{1}{2})$ is Pareto optimal for problem (P_{u^k}). Then, consider the sequence $\{(x^k, y^k)\}$ such that $x^k = \frac{1}{2}$ and $y^k = \frac{1}{2}$ for every k. Taking the limits for $k \to \infty$, $x^k \to \bar{x} = \frac{1}{2}$, which is not Pareto optimal for the original problem, since its objective vector is $F(\bar{x}) = (-\frac{1}{2}, 1)$. However, $\bar{x}$ is a weak Pareto optimal point for the original problem.

The same reasoning still holds if we consider the function $f^{u_k}(y) = \log(u_k + y)$.

Polyhedral sets

When X is a polyhedral set, the approximated problem is characterized by stronger equivalence results, as we now show. First, we state the following assumption.

Assumption 4.2.2. *The polyhedral set X has at least a vertex.*

Denoting by T the feasible set of (4.6), i.e.,

$$T = \{x, y \in \mathbb{R}^n \mid x \in X, -y \leq x \leq y\},$$

We now show that the equivalence in terms of weak Pareto points does not only hold asymptotically, but also "finitely". We first prove this property for the exponential approximation.

Proposition 4.2.2. *Suppose that X is a non-empty polyhedral set, and let $f^u(y) = 1 - e^{-uy}$. There exists a value $\bar{u} > 0$ such that, for any $u \geq \bar{u}$, problem (4.6) admits a weak Pareto point which is also a weak Pareto point for the original problem (SMOP).*

Proof. As shown in (Rinaldi et al., 2010, prop. 5), for each $u \geq \bar{u}$, there exists a solution $(x(u), y(u))$ to the problem

$$\begin{array}{ll} \min & \sum_{i=1}^{n} f^u(y_i) \\ s.t. & x \in X, \\ & -y \leq x \leq y \end{array} \tag{4.19}$$

which is also a solution to the sparse problem

$$\begin{array}{ll} \min & \|y\|_0 \\ s.t. & x \in X, \\ & -y \leq x \leq y. \end{array} \tag{4.20}$$

Since $(x(u), y(u))$ minimizes one of the objectives of both problem (4.6) and problem (SMOP), it must be at least a weak Pareto point for both problems. □

The previous result, with minor modifications, also holds for the logarithmic approximating function.

Proposition 4.2.3. *Suppose that X is a non-empty polyhedral set, and let $f^u(y) = \log(u + y)$. Assume that problem (4.20) admits a solution y^* such that $\|y^*\|_0 < n$. There exists a value $\bar{u} > 0$ such that, for any $u \in (0, \bar{u}]$, problem (4.6) admits a weak Pareto point which is also a weak Pareto point for the original problem (SMOP).*

Proof. As in the previous proposition, the result follows from (Rinaldi et al., 2010, prop. 6). □

In case problem (4.20) admits a single solution, the point $(x(u), y(u))$ defined in the previous propositions is a Pareto point for the original problem, since there cannot be another feasible point that improves or matches the zero norm objective value $\|y(u)\|_0$. This observation is formally stated in the following corollary.

Corollary 4.2.1. *Suppose that problem* (4.20) *admits a single solution* y^**. Then:*

1. *if* $f^u(y) = 1 - e^{-uy}$*, there exists a value* $\bar{u} > 0$ *such that, for any* $u \geq \bar{u}$*, problem* (4.6) *admits a weak Pareto point which is also a Pareto point for the original problem* (SMOP)*;*

2. *if* $\|y^*\|_0 < n$ *and* $f^u(y) = \log(u + y)$*, there exists a value* $\bar{u} > 0$ *such that, for any* $u \in (0, \bar{u}]$*, problem* (4.6) *admits a weak Pareto point which is also a Pareto point for the original problem* (SMOP)*.*

We now provide a couple of equivalence results in terms of Pareto points. First, when the first $m - 1$ objective functions are concave, there exists a finite parameter $\bar{u}$ such that every u "better" than $\bar{u}$ (i.e., $u \geq \bar{u}$ for property (4.7) or $u \leq \bar{u}$ for property (4.8)) yields a Pareto point of the approximated problem which is a Pareto point of the original problem.

Proposition 4.2.4. *Assume that the first* $m - 1$ *functions are concave functions bounded below on the polyhedral set* T*. Let* $\{u_k\}$ *be a sequence such that* $\lim_{k\to\infty} u_k = \bar{u}$ *and let* $f^u(y)$ *be a function satisfying condition* (4.9)*. Then, there exists an index* $\bar{k}$ *such that for all* $k \geq \bar{k}$ *there exists a Pareto point of the approximated problem which is a Pareto point of the original problem.*

Proof. Consider the scalarized functions

$$F(x, y) = w_1 f_1(x) + w_2 f_2(x) + \ldots + w_{m-1} f_{m-1}(x) + w_m \|y\|_0$$

$$F_u(x, y) = w_1 f_1(x) + w_2 f_2(x) + \ldots + w_{m-1} f_{m-1}(x) + w_m \sum_{i=1}^{n} f^u(y_i),$$

with $w_h > 0$ for $h = 1, \ldots, m$.

We know that any global minimum point of $F(x, y)$ over T is a Pareto point for the original problem, and any global minimum point of $F_u(x, y)$ over T is a Pareto point for the approximated problem.

Function $F_u(x, y)$ is concave and bounded below on T and hence admits a minimum point which is a vertex of T.

Now, in order to prove the thesis, by contradiction assume that there exists a sequence $\{u^k\}$ such that $u^k \to \bar{u}$ and, denoted by (x^k, y^k) the optimal vertex of $F_{u^k}(x, y)$, we have that (x^k, y^k) is not a Pareto point of the original problem.

The number of vertices is finite, so there exists an infinite subset K such that $(x^k, y^k) = (\bar{x}, \bar{y})$ for all $k \in K$. Since $(\bar{x}, \bar{y})$ is not a Pareto point for the original problem it can not be a global minimum point of $F(x, y)$ over T, and hence there exists a point $(\hat{x}, \hat{y}) \in T$ such that

$$F(\hat{x}, \hat{y}) < F(\bar{x}, \bar{y}). \tag{4.21}$$

As $(\bar{x}, \bar{y})$ is an optimal vertex for $F_{u^k}(x, y)$, we can write

$$F_{u^k}(\bar{x}, \bar{y}) \leq F_{u^k}(\hat{x}, \hat{y}).$$

Taking the limits for $k \in K$ and $k \to \infty$, recalling from condition (4.9) that

$$\lim_{k \in K, k \to \infty} \sum_{i=1}^{n} f^{u_k}(y_i) = \|y\|_0,$$

we obtain

$$F(\bar{x}, \bar{y}) \leq F(\hat{x}, \hat{y}),$$

which contradicts (4.21).

□

To conclude this section, we show that under some technical assumptions on the approximating function, the equivalence in terms of Pareto points also holds asymptotically without requiring the concavity of all the objective functions. It is worth noting that these technical conditions are mild and are satisfied by, for example, the exponential approximating function: $f^u(y_i) = 1 - e^{-uy_i}$.

Proposition 4.2.5. *Assume that the first $m - 1$ functions are bounded below on the polyhedral set T. Let $\{u_k\}$ be a sequence such that $\lim_{k \to \infty} u_k = \bar{u}$ and let $f^u(y)$ be a function satisfying Assumption 4.2.1 and condition (4.7). Assume also that the following conditions hold for each $u \geq 0$:*

1. $\sum_{i=1}^{n} f^u(|y_i|) \leq \|y\|_0$ *for all* $y \in \mathbb{R}^n$,
2. $f^u(y) \geq 0$ *for all* $y \geq 0$,
3. $f^u(y) \leq f^u(z)$ *for all* $0 \leq y \leq z$.

Let $w_i > 0$, $i = 1, \ldots, m$ be a set of positive weights and let $\{(x^k, y^k)\}$ be a sequence such that (x^k, y^k) is a solution of the scalar problem

$$\begin{aligned} \min \; & \sum_{i=1}^{m-1} w_i f_i(x) + w_m \sum_{i=1}^{n} f^{u_k}(y_i) \\ s.t. \; & x \in X, \\ & -y \leq x \leq y. \end{aligned} \tag{4.22}$$

Then, each accumulation point $\{(\bar{x}, \bar{y})\}$ *of the sequence* $\{(x^k, y^k)\}$ *is such that* $\bar{x}$ *is a Pareto point for the original problem* (SMOP).

Proof. By condition 3, we can assume without loss of generality that $y^k = |x^k|$, and let $(\bar{x}, \bar{y})$ be an accumulation point of the sequence $\{(x^k, y^k)\}$, i.e. there exists a subsequence K such that $\lim_{k \in K, k \to \infty} (x^k, y^k) = (\bar{x}, \bar{y})$. By definition, we have that $\bar{y} = |\bar{x}|$.

Suppose by contradiction that $\bar{x}$ is not a Pareto point for problem (SMOP). This implies that $\bar{x}$ is not a solution of the scalar problem

$$\begin{aligned} &\min \sum_{i=1}^{m-1} w_i f_i(x) + w_m \|x\|_0 \\ &s.t.\ x \in X, \end{aligned} \tag{4.23}$$

i.e. there exists a $z \in X$ such that

$$\sum_{i=1}^{m-1} w_i f_i(z) + w_m \|z\|_0 < \sum_{i=1}^{m-1} w_i f_i(\bar{x}) + w_m \|\bar{x}\|_0. \tag{4.24}$$

By definition of (x^k, y^k), for all $k \in K$ we have that

$$\sum_{i=1}^{m-1} w_i f_i(x^k) + w_m \sum_{i=1}^{n} f^{u_k}(y_i^k) \leq \sum_{i=1}^{m-1} w_i f_i(z) + w_m \sum_{i=1}^{n} f^{u_k}(|z_i|). \tag{4.25}$$

By condition 1, $\sum_{i=1}^{n} f^{u_k}(|z_i|) \leq \|z\|_0$ for all $u_k \geq 0$, and so, from (4.24) and (4.25), we can write:

$$\sum_{i=1}^{m-1} w_i f_i(x^k) + w_m \sum_{i=1}^{n} f^{u_k}(y_i^k) \leq \sum_{i=1}^{m-1} w_i f_i(z) + w_m \|z\|_0 < \sum_{i=1}^{m-1} w_i f_i(\bar{x}) + w_m \|\bar{x}\|_0. \tag{4.26}$$

Define the index set $I = \{i \mid \bar{y}_i \neq 0\}$. Then, from condition 2 it follows that

$$\sum_{i=1}^{m-1} w_i f_i(x^k) + w_m \sum_{i \in I} f^{u_k}(y_i^k) \leq \sum_{i=1}^{m-1} w_i f_i(z) + w_m \|z\|_0 < \sum_{i=1}^{m-1} w_i f_i(\bar{x}) + w_m \|\bar{x}\|_0. \tag{4.27}$$

Taking limits for $k \in K, k \to \infty$, recalling condition (iii) of Assumption 4.2.1 and the fact that $\bar{y} = |\bar{x}|$, we get that

$$\sum_{i=1}^{m-1} w_i f_i(\bar{x}) + w_m \|\bar{x}\|_0 < \sum_{i=1}^{m-1} w_i f_i(\bar{x}) + w_m \|\bar{x}\|_0, \tag{4.28}$$

a contradiction. □

4.3 The algorithm

Inspired by the work in Fliege and Vaz (2016), in this section we propose an algorithm (see Algorithm 8) to approximate the Pareto front of problem (SMOP). The algorithm is based on the iterative refinement of a set of non-dominated points, using a set of descent directions.

First, let us denote

$$f_m(y) = \sum_{i=1}^{n} f^u(y_i),$$

so that problem (4.6) can be rewritten as

$$\begin{aligned} &\min\ F(x,y) = (f_1(x), \ldots, f_{m-1}(x), f_m(y)) \\ &s.t.\ x \in X, \\ &\quad -y \leq x \leq y. \end{aligned} \tag{4.29}$$

From now on, we consider $z = (x,y)^\top$ as the set of optimization variables.

The skeleton of our algorithm is composed of three fundamental phases: an initialization phase, a search phase and a refining phase.

Initialization phase: the list is initialized with a set of non-dominated solutions. For simplicity, we consider the case in which the list is initialized with a singleton. Hence, given a feasible point z_0 and a stepsize $\alpha_0 > 0$, we initialize the list $Z_0 = \{(z_0, \alpha_0)\}$.

Search phase: we try to improve the set of non-dominated points, iterating over the list Z_k. At every iteration, we select a pair (z, α) and we generate the following set of points:

$$S(z,\alpha) = \{(z + \alpha d_i, \alpha_0) \mid \nabla f_i(z)^T d_i < 0, i \in 1, \ldots, m\} \tag{4.30}$$

where each direction d_i is a feasible descent direction in z for the corresponding objective function f_i. Then, the list Z_{k+1} is updated only considering non-dominated solutions of the set $Z_k \cup S(z,\alpha)$. If $Z_{k+1} = Z_k$, i.e. no new points are added to the list, then the iteration is considered unsuccessful and the stepsize related to the point z is decreased by a factor $\delta < 1$. The rationale is that of spreading a set of initial points by separately considering the single objective functions, aiming to obtain an approximation of the Pareto front that is "uniform", intuitively, w.r.t. each objective function.

Refining phase: in order to drive the obtained non-dominated points towards the Pareto front, the multiobjective steepest descent strategy is applied for each point of the list Z_k.

A few comments are in order:

Algorithm 8: MultiObjective Sparse Optimization (MOSO)

input: a nonempty, finite set of pairs $Z_0 = \{(z_0, \alpha_0)\}$, where every z_0 is a non-dominated point, $\alpha_0 > 0$, $\delta < 1$ and $\gamma < 1$.
ouput: a Pareto front approximation Z^*.
Set $k \leftarrow 0$.
`// Start Search phase`
while *stopping criterion not satisfied* **do**
  Select a pair $(z, \alpha) \in Z_k$.
  Compute the set of points $S(z, \alpha)$ as in (4.30).
  Set `iter_success` $\leftarrow$ *false*.
  if $S(z, \alpha) \neq \emptyset$ **then**
    Set $L \leftarrow Z_k \cup S(z, \alpha)$.
    Set $L_{tmp} \leftarrow \{(w, \alpha_w) \in L \mid \nexists \bar{z} \in L \; s.t. \; F(\bar{z}) \leq F(w)\}$.
    if $L_{tmp} \neq Z_k$ **then**
      Set $Z_{k+1} \leftarrow L_{tmp}$.
      `iter_success` $\leftarrow$ *true*.
    end
  else
    $L_{tmp} = Z_k$.
  end
  if *not* `iter_success` **then**
    Set $Z_{k+1} \leftarrow L_{tmp} \setminus \{(z, \alpha)\} \cup \{(z, \delta \cdot \alpha)\}$.
  end
  Set $k \leftarrow k + 1$.
end
`// End Search phase`
Set $Z_f \leftarrow \emptyset$.
`// Start Refining phase`
for *each* $z \in Z_k$ **do**
  $z^* \leftarrow$ `steepest_descent_algorithm`(z, γ).
  $Z_f \leftarrow Z_f \cup \{z^*\}$.
end
`// End Refining phase`

- The search phase stops when one of the following criteria is satisfied: all the stepsizes related to the list Z_k are lower than a positive tolerance or a maximum number of function evaluations is reached[1].

- In the search phase we consider two different feasible descent directions per objective, computed with the Projected Gradient and Frank-Wolfe method re-

[1] A computation of the vector $F(x)$ counts as a single function evaluation. A computation of the gradient vector $\nabla F(x)$ counts as n function evaluations.

spectively. Moreover, we also include the common steepest descent direction, which, as described in Section 4.1, can be computed by retrieving the solution (τ^*, x^*, y^*) of the problem

$$\begin{aligned} &\min_{\tau,x,y} \ \tau \\ &s.t. \ \nabla f_i(\bar{x})^\top (x - \bar{x}) - \tau \leq 0, \quad i = 1, \dots, m-1, \\ &\quad \nabla f_m(\bar{y})^\top (y - \bar{y}) - \tau \leq 0, \\ &\quad x \in X, \\ &\quad -y \leq x \leq y, \end{aligned} \tag{4.31}$$

and setting $d_z = (d_x, d_y) = (x^* - \bar{x}, y^* - \bar{y})$.

- The notion of dominated point is referred to the original objective vector $(f_1(x), f_2(x), \dots, f_{m-1}(x), ||x||_0)$, i.e., we consider the actual ℓ_0-norm, not the approximation.

4.4 Numerical results

Implementation details

For numerical experiments we use a concave approximation of the ℓ_0-norm using the logarithmic approximation:

$$||x||_0 \approx \sum_{i=1}^{n} \log(\varepsilon + x_i). \tag{4.32}$$

We made this choice since, in our experiments, the logarithmic approximation obtained better results than the exponential approximation.

Parameters of algorithm `MOSO` have been set as follows:

$$\begin{aligned} &\varepsilon = 10^{-5}, \quad &&\delta = 0.5, \\ &\alpha_0 = 1, \quad &&\alpha_{min} = 10^{-7} \\ &\gamma = 10^{-6}, \end{aligned}$$

where α_{min} represents the tolerance on the stepsizes for the search phase.

Numerical results on sparse portfolio selection problems

To assess the effectiveness of Algorithm 8, we consider the following portfolio selection problem:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} & -\frac{\mu^\top x}{x^\top Q x}, \|x\|_0 \\ s.t.\ & e^\top x = 1, \\ & 0 \leq x \leq b, \end{aligned} \tag{4.33}$$

where $\mu \in \mathbb{R}^n$ and $Q \in \mathbb{R}^{n\times n}$ are the vector of expected returns and covariance matrix, respectively, and b is a resource constraint. This problem is obtained from the standard Markowitz model

$$\begin{aligned} \min_{x\in\mathbb{R}^n} & x^\top Q x \\ s.t.\ & e^\top x = 1, \\ & \mu^\top x \geq R, \\ & 0 \leq x \leq b \end{aligned} \tag{4.34}$$

replacing the objective function $x^\top Qx$ with the so-called "Sharpe ratio" (which takes into account both the variance $x^\top Qx$ and the expected return $\mu^\top x$), and by adding $\|x\|_0$ as a further objective. In our experiments, we set $b = \mathbf{1}$.

The data used in the following experiments consists of daily data for securities from the FTSE 100 index, from 01/2003 to 12/2007. Such data is public and available from the website `http://www.bolsapt.com`. The three datasets are referred to as DTS1, DTS2, and DTS3, and are formed by 12, 24, and 48 securities, respectively. The assets we considered for the generation of the datasets are those used in Brito and Vicente (2014). Moreover, we also included three datasets from the Fama/French benchmark collection (FF10, FF17, and FF48, with cardinalities 10, 17, and 48), using the monthly returns from 07/1971 to 06/2011 (forty years) given there for a number of industry security sectors. More information about these last three datasets can be found at `http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html`. For further details on these 6 dataset, see Brito and Vicente (2014).

We mainly compare the performance of `MOSO` with the Direct Multisearch (`DMS`) algorithm (Custódio et al. (2011)) with its setup defined in Brito and Vicente (2014), where a custom approach to deal with the equality constraint is proposed. Moreover, we set the maximum number of function evaluations to 200,000 and we consider the variables with absolute value lower than 10^{-8} as zero. For completeness, in the final part of the section we also considered the Pareto front approximations obtained by three other multiobjective algorithms, namely `NSGA-II` Deb et al. (2002) (via the `gamultiobj` function included in the MATLAB© Global Optimization Tool-

box MATLAB Optimization Toolbox (2018)), MOFEPSO Sinan Hasanoglu and Dolen (2018) and NOMAD Le Digabel (2011).

The spirit of these experiments is to compare the "non-smooth" derivative-free approach, on which DMS, NSGA-II, MOFEPSO and NOMAD are based, with the steepest descent approach we employ in our approach. Note, however, that the results we report in the following sections are with respect to the original vector of objectives of problem (SMOP), i.e., considering the actual $\|x\|_0$ objective.

Single point comparison

Here, we first show a comparison on the "single point" scenario, i.e. when only one Pareto point has to be retrieved, between MOSO and DMS[2]. Since the goal is to obtain a single Pareto point, the list Z_k is always composed of a single solution. To maintain a single point, we modify the search phase in MOSO as follows: we update the current solution only if the algorithm is able to find a point that strictly dominates it in the set L. More formally, lines 9-14 of Algorithm 8 are replaced by the following:

Set $L \leftarrow S(z, \alpha)$.
for each $\hat{z} \in L$ **do**
  if $\hat{z}$ strictly dominates Z_k **then**
    $Z_k \leftarrow \hat{z}$.
    iter_success $\leftarrow$ *true*.
  end if
end for

In Table 4.1, we report the results obtained on the DTS1, DTS2, DTS3, FF10, FF17 and FF48 datasets. We run both MOSO and DMS starting from 100 different initial points (each time, we use the same initial point for both algorithms). The list L_{init} of initial points is the following:

$$L_{init} = \{e_1, \ldots, e_n, x_{n+1}, \ldots, x_{100}\},$$

where $e_i \in \mathbb{R}^n$ is the i-th unit vector[3] and all the $x_i \in \mathbb{R}^n$ are random dense feasible points. For each dataset, we report:

- the number of times the point obtained by MOSO strictly dominates the one obtained by DMS (column MOSO-Wins);
- the number of times the point obtained by DMS strictly dominates the one obtained by MOSO (column DMS-Wins).

[2]The other algorithms do not provide this functionality.
[3]For the DTS1, DTS2 and DTS3, n is equal to 12, 24, 48, respectively. For the FF datasets, n is equal to 10, 17 and 48, respectively.

	MOSO-Wins	DMS-Wins
DTS1	**36**	3
DTS2	**95**	1
DTS3	**70**	6
FF10	**92**	0
FF17	**94**	0
FF48	**90**	10

Table 4.1: Single point comparison for DTS and FF problems.

The results show the effectiveness of `MOSO`. In fact, `MOSO` is able to outperform `DMS` on every dataset, dominating the `DMS` solution considerably often.

Pareto front comparison

In order to assess the performance of our approach, we compute the purity and spread metrics defined in Custódio et al. (2011) for each solver. We recall that the purity metric measures the quality of the generated front, i.e. how good the non-dominated points computed by a solver are with respect to those computed by any other solver. Here, a higher value is a better value. The spread metrics, instead, measure the uniformity of the generated front in the objectives space. In particular, the spread Γ metric is defined as the maximum ℓ_∞ distance between adjacent points in the retrieved Pareto front. The spread Δ metric is related to the standard deviation of the ℓ_∞ distances between adjacent points in the retrieved Pareto front. Here, instead, a lower value is a better value.

First, the solvers were initialized with the singleton $\{x_0\}$[4] such that:

$$x_0 = \begin{bmatrix} 0, & \ldots, & 0, & 1 \end{bmatrix}^\top. \tag{4.35}$$

In Table 4.2, purity and spread metrics scores are reported for the considered problems. Note that we do not consider the `NSGA-II` and `MOFEPSO` solvers here, since they are evolutionary algorithms that thrive when started with a list of points and for which an initialization with a single point would be unfair.

The obtained results show the effectiveness of our solver with respect to the purity metric. With respect to the spread Γ metric, it is easy to see that, generally, when `MOSO` is the best algorithm, the performance ratios of the other solvers are very high, while when it is outperformed the performance ratio is low. With respect to the spread Δ metric, a degenerate case happens for both the DTS2 and the FF48 datasets. In fact, `DMS` is not able to extend the Pareto front. Since the computation of spread Δ requires the mean distance between consecutive points in the front, it can be set

[4]This initialization is suggested in Brito and Vicente (2014).

Purity			
Prob/Alg	MOSO	DMS	NOMAD
DTS1	**0.71**	0.17	0.2
DTS2	**1**	**1**	0
DTS3	**0.94**	0.22	0.06
FF10	**1**	0.5	0.25
FF17	**1**	0.5	0.25
FF48	**1**	**1**	0

Spread Γ			
Prob/Alg	MOSO	DMS	NOMAD
DTS1	2.66	**2.11**	5
DTS2	**3.81**	12	11
DTS3	3.29	9	**2**
FF10	**1**	**1**	**1**
FF17	**1**	**1**	2
FF48	**1**	5	3.43

Spread Δ			
Prob/Alg	MOSO	DMS	NOMAD
DTS1	0.78	**0.62**	1.14
DTS2	**0.70**	∞	0.98
DTS3	0.85	0.78	**0.70**
FF10	0.89	**0.52**	1.16
FF17	0.68	**0.5**	0.92
FF48	**0.93**	∞	1.04

Table 4.2: Purity and Spread tables for DTS and FF problems starting with a singleton.

to ∞ when only one point is retrieved. In the other problems, `DMS` seems to be the most robust choice w.r.t. this metric.

To sum up, `MOSO` is clearly the winner w.r.t. the purity metric, and is a good choice also for the spread Γ metric; while it may suffer a little bit w.r.t. the spread Δ metric, overall it shows a good level of performance.

In the second case, the five algorithms where compared when initialized with a list of points. More precisely:

- we tested the solvers starting from a list of 5 random points for each feasible cardinality (e.g., with $n = 12$ variables the starting list contains $12 \cdot 5 = 60$ points), and we executed the algorithms with 5 different random seeds obtaining 30 total instances;

- we compute the performance profiles Dolan and Moré (2002) of the five solvers considering all the 30 instances of the problems.

Concerning the comparison between MOSO and the other 4 approaches, the profiles confirm the results obtained with the singleton initialization. In fact, our solver outperforms the others in terms of purity and spread Γ as in the singleton case. For the spread Δ metric, MOSO is outperformed, but maintains a reasonable level of performance.

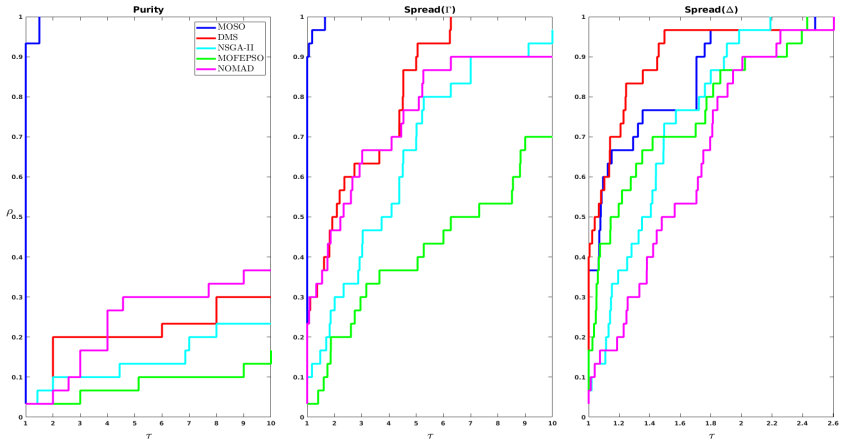


Figure 4.1: Performance profiles computed over the 30 instances of portfolio problems starting from a random list of points.

Numerical results on 3-objective optimizazion problems

To further assess the performance of our approach, we also considered a set of box constrained problems with 3 objective functions, where the third objective is the ℓ_0-norm. The problems are of the form

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} (f_1(x), f_2(x), \|x\|_0) \\ &s.t.\ 0 \le x \le b, \end{aligned} \tag{4.36}$$

where $b = 10 \cdot \mathbf{1}$. In particular, we consider the following three problems (BK1, MOP2, QV1) from the dataset used in Cocchi et al. (2018); Custódio et al. (2011):

1. **BK1**:

$$\begin{aligned} f_1(x) &= \|x\|^2 \\ f_2(x) &= \|x - 5 \cdot \mathbf{1}\|^2 \end{aligned} \tag{4.37}$$

2. **MOP2**

$$\begin{aligned} f_1(x) &= 1 - e^{-\sum_{i=1}^n (x_i + 1/\sqrt{n})^2} \\ f_2(x) &= 1 - e^{-\sum_{i=1}^n (x_i - 1/\sqrt{n})^2} \end{aligned} \tag{4.38}$$

3. **QV1**[5]:

$$\begin{aligned} f_1(x) &= \frac{1}{n}\sum_{i=1}^n \left(x_i^2 - 10\cos(2\pi x_i) + 10\right) \\ f_2(x) &= \frac{1}{n}\sum_{i=1}^n \left((x_i - 1.5)^2 - 10\cos(2\pi(x_i - 1.5)) + 10\right). \end{aligned} \tag{4.39}$$

[5]The functions are made continuously differentiable by removing the 1/4 roots.

in which the $||\cdot||_0$ function represents the third objective function.

Single point comparison

As in the previous case, we begin our analysis with the "single point" scenario, as explained in section 4.4. The results are reported in Table 4.3.

	MOSO-Wins	DMS-Wins
BK1	**22**	0
MOP2	**27**	0
QV1	**20**	0

Table 4.3: Single point comparison for the 3 objectives problems.

As in the previous case, `MOSO` outperforms `DMS` in each problem.

Pareto front comparison

We now report the results in terms of Pareto front approximations obtained on the 3 objectives datasets[6]. First, we report the results obtained by `MOSO` and `DMS` with the singleton initialization in Table 4.4. In general, `MOSO` works reasonably well when compared to `DMS`, in terms of Purity and Spread Δ. The Spread Γ metric, instead, sees `DMS` as the clear winner.

The results obtained with the "list" initialization are reported in Figure 4.2. In particular, the results are obtained by executing the algorithms on the 3 problems with five different list initializations, as explained in section 4.4.

[6]The NOMAD algorithm is not reported since the available implementation does not support problems with more than 2 objectives.

Purity		
Prob/Alg	MOSO	DMS
BK1	**1**	0.91
MOP2	0.81	**0.88**
QV1	**1**	0.95

Spread Δ		
Prob/Alg	MOSO	DMS
BK1	**1.25**	1.99
MOP2	**1.26**	1.96
QV1	2	**1.96**

Spread Γ		
Prob/Alg	MOSO	DMS
BK1	25	**1.25**
MOP2	2	**1.26**
QV1	26	**1**

Table 4.4: Purity and Spread tables for the BK1, MOP2 and QV1 problems starting with a singleton.

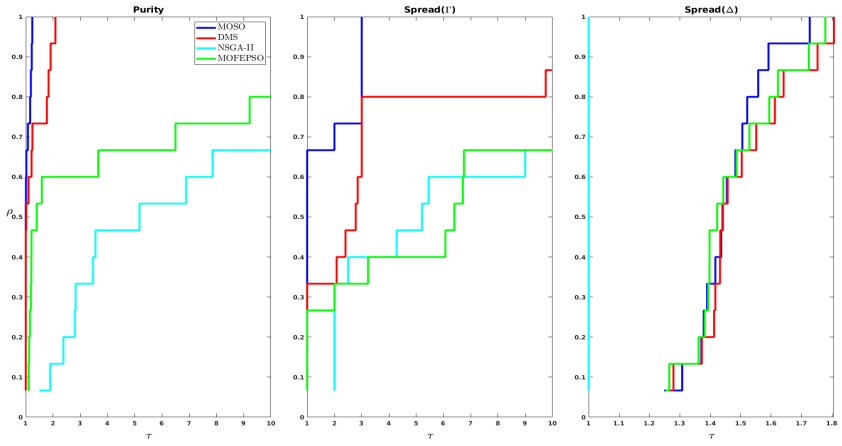


Figure 4.2: Performance profiles computed overall the 15 instances of 3 objective problems starting from a random list of points.

The numerical results show the effectiveness of the proposed algorithm also on 3-objective optimization problems in terms of the Purity and the Spread Γ metrics as in the portfolio case, while with respect the Spread Δ it is outperformed by `NSGA-II` but it is comparable with the other three solvers.

4.5 Conclusions

In this chapter, we have proposed a smooth formulation for constrained sparse multiobjective optimization problems on a compact convex set, by replacing the original discontinuous ℓ_0-norm objective with concave approximation functions. A series of equivalence results between the smooth reformulation and the original problem were stated.

We have also proposed an algorithm for solving sparse multiobjective optimization problems. The proposed approach computes an approximation of the true Pareto front exploiting first order information (multiobjective steepest descent algorithm) in order to improve the set of non dominated solutions.

We have compared our method against a state-of-the-art algorithm, namely DMS, on a set of sparse multiobjective optimization problems. The obtained numerical results show the effectiveness of the proposed algorithm both in computing a single Pareto solution and in computing an approximation of the entire Pareto front. For what concerns the Pareto front approximation, a further comparison with three other algorithms for multiobjective optimization, namely NSGA-II, MOFEPSO and NOMAD, confirmed the effectiveness of the proposed approach.

Bibliography

Anagnostopoulos, K. and Mamanis, G. (2010). A portfolio optimization model with three objectives and discrete variables. *Computers & Operations Research*, 37(7):1285–1297.

Bai, Y., Liang, R., and Yang, Z. (2016). Splitting augmented lagrangian method for optimization problems with a cardinality constraint and semicontinuous variables. *Optimization Methods and Software*, 31(5):1089–1109.

Beck, A. and Eldar, Y. (2013). Sparsity constrained nonlinear optimization: Optimality conditions and algorithms. *SIAM Journal on Optimization*, 23(3):1480–1509.

Berge, C. (1963). *Topological Spaces: Including a Treatment of Multi-valued Functions, Vector Spaces and Convexity*. Macmillan.

Bertsimas, D., King, A., and Mazumder, R. (2016). Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44(2):813–852.

Bertsimas, D. and Shioda, R. (2009). Algorithm for cardinality-constrained quadratic optimization. *Computational Optimization and Applications*, 43(1):1–22.

Bienstock, D. (1996). Computational study of a family of mixed-integer quadratic programming problems. *Mathematical Programming*, 74(2):121–140.

Blumensath, T. and Davies, M. E. (2008). Iterative thresholding for sparse approximations. *Journal of Fourier Analysis and Applications*, 14(5):629–654.

Boudt, K. and Wan, C. (2019). The effect of velocity sparsity on the performance of cardinality constrained particle swarm optimization. *Optimization Letters*.

Brito, R. P. and Vicente, L. N. (2014). Efficient cardinality/mean-variance portfolios. In *System Modeling and Optimization*, Springer series IFIP Advances in Information and Communication Technology, pages 52–73. Springer, Berlin, Heidelberg.

Burdakov, O., Kanzow, C., and Schwartz, A. (2016). Mathematical Programs with Cardinality Constraints: Reformulation by Complementarity-Type Conditions and a Regularization Method. *SIAM Journal on Optimization*, 26(1):397–425.

Carreira-Perpinan, M. A. and Idelbayev, Y. (2018). "learning-compression" algorithms for neural net pruning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8532–8541.

Cocchi, G., Liuzzi, G., Papini, A., and Sciandrone, M. (2018). An implicit filtering algorithm for derivative-free multiobjective optimization with box constraints. *Computational Optimization and Applications*, 69(2):267–296.

Custódio, A. L., Madeira, J. F. A., Vaz, A. I. F., and Vicente, L. N. (2011). Direct multisearch for multiobjective optimization. *SIAM Journal on Optimization*, 21(3):1109–1140.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Deng, G.-F., Lin, W.-T., and Lo, C.-C. (2012). Markowitz-based portfolio selection with cardinality constraints using improved particle swarm optimization. *Expert Systems with Applications*, 39(4):4558–4566.

Di Gangi, L., Lapucci, M., Schoen, F., and Sortino, A. (2019). An efficient optimization approach for best subset selection in linear regression, with application to model selection and fitting in autoregressive time-series. *Computational Optimization and Applications*.

Di Lorenzo, D., Liuzzi, G., Rinaldi, F., Schoen, F., and Sciandrone, M. (2012). A concave optimization-based approach for sparse portfolio selection. *Optimization Methods and Software*, 27(6):983–1000.

Dolan, E. D. and Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, 32(2):407–499.

Eldar, Y. C. and Kutyniok, G. (2012). *Compressed sensing: theory and applications*. Cambridge university press.

Fasano, G., Liuzzi, G., Lucidi, S., and Rinaldi, F. (2014). A linesearch-based derivative-free approach for nonsmooth constrained optimization. *SIAM Journal on Optimization*, 24(3):959–992.

Fliege, J. and Svaiter, B. (2000). Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494.

Fliege, J. and Vaz, A. (2016). A Method for Constrained Multiobjective Optimization Based on SQP Techniques. *SIAM Journal on Optimization*, 26(4):2091–2119.

Fukuda, E. H. and Drummond, L. M. G. (2014). A survey on multiobjective descent methods. *Pesquisa Operacional*, 34(3):585–620.

John, G. H., Kohavi, R., and Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *Machine Learning Proceedings 1994*, pages 121–129. Elsevier.

Le Digabel, S. (2011). Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm. *ACM Transactions on Mathematical Software*, 37(4):1–15.

Liuzzi, G., Lucidi, S., and Rinaldi, F. (2016). A derivative-free approach to constrained multiobjective nonsmooth optimization. *SIAM Journal on Optimization*, 26(4):2744–2774.

Lu, Z. and Zhang, Y. (2013). Sparse Approximation via Penalty Decomposition Methods. *SIAM Journal on Optimization*, 23(4):2448–2478.

Lucidi, S. and Piccialli, V. (2004). A derivative-based algorithm for a particular class of mixed variable optimization problems. *Optimization Methods and Software*, 19(3-4):371–387.

Lucidi, S., Piccialli, V., and Sciandrone, M. (2005). An Algorithm Model for Mixed Variable Programming. *SIAM Journal on Optimization*, 15(4):1057–1084.

Lucidi, S. and Sciandrone, M. (2002). A derivative-free algorithm for bound constrained optimization. *Computational Optimization and Applications*, 21(2):119–142.

Mangasarian, O. L. (1996). *Machine Learning via Polyhedral Concave Minimization*, pages 175–188. Physica-Verlag HD, Heidelberg".

MATLAB Optimization Toolbox (2018). MATLAB R2018b Global Optimization Toolbox v4.0. The MathWorks, Natick, MA, USA.

Miller, A. (2002). *Subset Selection in Regression*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. CRC Press.

Mutunge, P. and Haugland, D. (2018). Minimizing the tracking error of cardinality constrained portfolios. *Computers & Operations Research*, 90:33–41.

Natarajan, B. K. (1995). Sparse Approximate Solutions to Linear Systems. *SIAM Journal on Computing*, 24(2):227–234.

Nikolova, M. (2016). Relationship between the optimal solutions of least squares regularized with ℓ_0-norm and constrained by k-sparsity. *Applied and Computational Harmonic Analysis*, 41(1):237–265.

Rinaldi, F., Schoen, F., and Sciandrone, M. (2010). Concave programming for minimizing the zero-norm over polyhedral sets. *Computational Optimization and Applications*, 46(3):467–486.

Sinan Hasanoglu, M. and Dolen, M. (2018). Multi-objective feasibility enhanced particle swarm optimization. *Engineering Optimization*, 50(12):2013–2037.

Soubies, E., Blanc-Féraud, L., and Aubert, G. (2017). A Unified View of Exact Continuous Penalties for ℓ_2-ℓ_0 Minimization. *SIAM Journal on Optimization*, 27(3):2034–2060.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

Xu, Z., Chang, X., F., X., and Zhang, H. (2012). $l_{1/2}$ regularization: A thresholding representation theory and a fast solver. *IEEE Transactions on Neural Networks and Learning Systems*, 23(7):1013–1027.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.

Appendix A

Publications

Journal papers

1. F. Ceccarelli, M. Sciandrone, C. Perricone, G. Galvan, E. Cipriano, A. Galligari, **T. Levato**, T. Colasanti, L. Massaro, F. Natalucci, F. R. Spinelli, C. Alessandri, G. Valesini, F. Conti, "Biomarkers of erosive arthritis in systemic lupus erythematosus: Application of machine learning models", *PLOS ONE*, volume:13, pages:1–12, 2018. **Candidate's contributions**: participated in designing the application, in the implementation of the algorithm and in carrying out numerical experiments.

2. G. Galvan, M. Lapucci, **T. Levato**, M. Sciandrone, "An Alternating Augmented Lagrangian method for constrained nonconvex optimization", *Optimization Methods and Software*, to appear, pages:1–19, 2019. **Candidate's contributions**: participated in the theoretical analysis, in the implementation of the algorithm and in carrying out numerical experiments.

3. G. Cocchi, **T. Levato**, G. Liuzzi, M. Sciandrone, "A concave optimization-based approach for sparse multiobjective programming", *Optimization letters*, to appear, pages:1–22, 2019. **Candidate's contributions**: participated in the theoretical analysis and in carrying out numerical experiments.

Papers under review

- L. Galli, **T. Levato**, F. Schoen, L. Tigli, "Prescriptive Analytics for Inventory Management in Health Care", *Journal of the Operational Research Society*. **Candidate's contributions**: participated in the literature review, in designing the algorithmic framework and in carrying out numerical experiments.

- M. Lapucci, **T. Levato**, M. Sciandrone, "A Class of Convergent Inexact Penalty Decomposition Methods for Cardinality Constrained Problems", *Journal of Optimization Theory and Applications*. **Candidate's contributions**: participated in the theoretical analysis.

Other

- **T. Levato**, F. Rinaldi, M. Sciandrone, "A Novel Method for Cardinality Constrained Optimization", to be submitted. **Candidate's contributions**: participated in the theoretical analysis, implemented the algorithm and carried out numerical experiments.