SAPIENZA
UNIVERSITÀ DI ROMA

Large-scale optimization: new active-set methods and application in unsupervised learning

Dipartimento di Ingegneria Informatica, Automatica e Gestionale "Antonio Ruberti"

Dottorato di Ricerca in Automatica e Ricerca Operativa – XXIX Ciclo

Candidate

Andrea Cristofari
ID number 1185246

Thesis Advisor

Prof. Stefano Lucidi

A thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Automatica and Operations Research

January 2017

Large-scale optimization: new active-set methods and application in unsupervised learning

Ph.D. thesis. Sapienza – University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: cristofari@dis.uniroma1.it

Contents

Chapter 1

Introduction

Mathematical programming (or optimization) is a discipline of applied mathematics. It deals with solving decision problems where a real-valued function $f(x)$ must be minimized (or maximized) over a feasible region $\mathcal{F}$. Optimization problems frequently arise in many different fields, e.g., engineering, economics, artificial intelligence, mechanical design and many others.

A typical mathematical programming problem can be written as

$$\begin{aligned} &\min_x f(x) \\ &x \in \mathcal{F} \subseteq X, \end{aligned}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$ is called *objective function*, x is the n-dimensional variable vector, X is the variable space and $\mathcal{F}$ is called *feasible set* (or *feasible region*).

It is possible to define different classes of optimization problems, depending on the features of the variable space, the objective function and the feasible set. For example, we distinguish between *continuous* and *integer* problems, *differentiable* and *non-differentiable* problems, *unconstrained* and *constrained* problems.

Numerical methods for solving optimization problems have been widely studied in the literature, from both a theoretical and a computational point of view. Lately, the growing technological development has made possible to collect and store large amounts of data, and then, efficient methods are strongly needed for their manipulation. This need also affects the field of numerical optimization. For example, many popular machine learning models are based on solving specific optimization problems and the use of appropriate methods is crucial to compute a solution in a reasonable amount of time.

Hence, defining algorithms that are able to efficiently deal with a large number of variables is an appealing and challenging task of current research. Here, new methods are proposed for some different classes of optimization problems, giving a particular attention to the large-scale case.

The guideline of this thesis is to provide mathematical arguments for the proposed algorithms, investigate their theoretical properties and show their behaviour in practice, also in comparison with other existing methods. The rest of the thesis is organized as follows.

In Chapter 2, we provide an overview of algorithms for unconstrained optimization, with a particular attention for the theoretical aspects and the methods that

will be exploited in the subsequent chapters.

In Chapter 3, we focus on box-constrained problems and we propose a two-stage active-set method, embedded into a non-monotone stabilization framework. The distinguishing feature of this algorithm is to exploit a theoretical property of the active-set estimate that guarantees a sufficient decrease in the objective function by setting the estimated active variables to the bounds. In this way, we can separately update the variables estimated active and those estimated non-active at each iteration. An in-depth convergence analysis of the algorithm is presented and numerical results are provided.

In Chapter 4, we extend the above described approach to minimize a function over the unit simplex. Also in this case, the main feature of the proposed algorithmic framework is to separately update the variables estimated active and those estimated non-active. In particular, following the idea of setting the estimated active variables to the bounds at each iteration, the presence of the linear constraint requires a further effort to maintain feasibility, while ensuring a sufficient decrease in the objective function. The convergence of the algorithm is proved when different variants of the Frank-Wolfe direction are employed. Then, we consider the problem of minimizing a function over the ℓ_1-ball, showing how the proposed method can be suitably adapted for this purpose. Preliminary numerical results are given.

In Chapter 5, an optimization-based strategy for cluster analysis is presented. In particular, a filtering data method is proposed, based on minimizing a least squares function with a weighted ℓ_0-norm penalty. To overcome the discontinuity of the objective function, smooth non-convex functions are employed to approximate the ℓ_0-norm, extending some approaches previously proposed in the literature. Theoretical properties of the model are investigated and numerical results are finally provided.

Two appendices are included: in Appendix A we report the fundamental concepts needed for this thesis and in Appendix B we report the details of the numerical experience that is discussed in Chapter 3.

This thesis includes material from three papers (published or submitted for publication) by the author. In particular, Chapter 3 and Chapter 4 are respectively based on [13] and [14], both co-authored with Marianna De Santis, Stefano Lucidi and Francesco Rinaldi. Finally, Chapter 5 is based on [12].

Chapter 2

Algorithms for unconstrained optimization: an overview

In this chapter, we focus on algorithms for solving unconstrained optimization problems where the objective function is (at lest) continuously differentiable.

Besides their own interest, such methods play a crucial role also in the field of constrained optimization, since many popular approaches (not investigated here) are based on reformulating a constrained problem as an unconstrained one.

In particular, the scope of this chapter is introducing some concepts that will be used in the rest of the thesis. For this reason, we only focus on the methods that are needed for our purposes.

Since the results presented in this chapter are well known, they are reported without proofs. For more details, the interested reader can see [4, 38, 61, 62, 66] and the references therein.

The rest of the chapter is organized as follows. In Section 2.1, we recall some general results on algorithms for unconstrained optimization. In Section 2.2, we introduce the general scheme of globally convergent methods that exploit a line search technique. In Section 2.3, we review some well-known line search methods proposed in the literature. Finally, in Section 2.4, we briefly describe some popular algorithms.

2.1 Generalities

Let us start by considering the following unconstrained minimization problem:

$$\begin{aligned} &\min_x f(x) \\ &x \in \mathbb{R}^n, \end{aligned}$$

where $f \in C^1(\mathbb{R}^n)$.

Generally, optimization algorithms produce a sequence of points $\{x^k\}$, according to the scheme reported in Algorithm 1.

When analyzing the properties of an algorithm for unconstrained optimization, a standard assumption is that $\mathcal{L}(f(x^0))$ is a compact set. This ensures that the

Algorithm 1 `General algorithm for unconstrained optimization`

0 Choose $x^0 \in \mathbb{R}^n$, set $k = 0$
1 While $\nabla f(x^k) \neq 0$
2 Compute $s^k \in \mathbb{R}^n$
3 Set $x^{k+1} = x^k + s^k$
4 Set $k = k + 1$
5 End while

problem attains optimal solutions (see Proposition 24 in Subsection A.2.2). Moreover, if $\{x^k\}$ is forced to remain in $\mathcal{L}(f(x^0))$, then the compactness of $\mathcal{L}(f(x^0))$ guarantees that $\{x^k\}$ attains limit points and each of them belongs to $\mathcal{L}(f(x^0))$.

Here, we focus on computational methods for unconstrained local optimization, that aim to find stationary points. Hence, a crucial property that these algorithms should guarantee is the convergence, in some sense, to a point x^* such that $\nabla f(x^*) = 0$.

In particular, different types of convergence can be defined. For example, in some cases (e.g., when the objective function is convex quadratic), it is possible to prove that there exists a finite index ν such that x^ν is stationary. In other cases, it can be established that the whole sequence $\{x^k\}$ converges to a stationary point, that is,

$$\lim_{k\to\infty} x^k = x^* \quad \text{and} \quad \nabla f(x^*) = 0.$$

In other cases, it is possible to guarantee a weaker condition, i.e.,

$$\lim_{k\to\infty} \nabla f(x^k) = 0$$

(if all points of $\{x^k\}$ are in a compact set, then the above relation is equivalent to saying that every limit point of $\{x^k\}$ is stationary). In other cases, it is only possible to ensure that

$$\liminf_{k\to\infty} \nabla f(x^k) = \lim_{k\to\infty} \Big\{ \inf_{m\geq k} \{\nabla f(x^m)\} \Big\} = 0.$$

In practice, algorithms are usually stopped when

$$\|\nabla f(x^k)\|_p \leq \epsilon,$$

where ϵ is a positive scalar (typical values are $\epsilon = 10^{-5}$, or $\epsilon = 10^{-6}$) and $\|\cdot\|_p$ is a suitable norm (e.g., the Euclidean norm, or the sup-norm).

Another feature that characterizes optimization algorithms is whether the sequence $\{x^k\}$ converges to a stationary point independently from the starting point x^0. We give the following definition.

Definition 1. *An optimization algorithm is said to be*

- *globally convergent if it converges to a stationary point x^* independently from the starting point,*

- *locally convergent if it converges to a stationary point x^* when the starting point belongs to a specific neighborhood of x^*.*

Moreover, it can be interesting to analyze the *convergence rate* of an algorithm, that is, how "fast" $\{x^k\}$ converges to a stationary point. More formally, given a sequence $\{x^k\} \subseteq \mathbb{R}^n$ such that

$$\lim_{k\to\infty} x^k = x^*,$$

we are interested in estimating the asymptotic rate at which the error e^k converges to zero, where e^k is a measure of the distance between x^k and x^* (for example $e^k = \|x^k - x^*\|$).

Two main criteria are generally considered in the literature:

- Q-rate, which concerns the analysis of the sequence $\{e^{k+1}/e^k\}$;
- R-rate, which concerns the analysis of the sequence of the roots of the error.

We give the following definitions.

Definition 2. *Given a sequence $\{x^k\} \subseteq \mathbb{R}^n$ converging to $x^* \in \mathbb{R}^n$, we say that*

- *$\{x^k\}$ converges to x^* (at least) Q-linearly if there exists $C \in [0, 1)$ such that, for sufficiently large k, we have*

$$\|x^{k+1} - x^*\| \leq C\|x^k - x^*\|;$$

- *$\{x^k\}$ converges to x^* (at least) Q-quadratically if there exists $C > 0$ such that, for sufficiently large k, we have*

$$\|x^{k+1} - x^*\| \leq C\|x^k - x^*\|^2;$$

- *$\{x^k\}$ converges to x^* (at least) Q-superlinearly if*

$$\lim_{k\to\infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} = 0;$$

- *$\{x^k\}$ converges to x^* Q-superlinearly with Q-rate (at least) p if there exist $p > 0$ and $C > 0$ such that, for sufficiently large k, we have*

$$\|x^{k+1} - x^*\| \leq C\|x^k - x^*\|^p.$$

Definition 3. *Given a sequence $\{x^k\} \subseteq \mathbb{R}^n$ converging to $x^* \in \mathbb{R}^n$, we say that*

- *$\{x^k\}$ converges to x^* (at least) R-linearly if there exists $C \in [0, 1)$ such that, for sufficiently large k, we have*

$$\|x^{k+1} - x^*\|^{1/k} \leq \sigma;$$

- *$\{x^k\}$ converges to x^* (at least) R-superlinearly if*

$$\lim_{k\to\infty} \|x^{k+1} - x^*\|^{1/k} = 0;$$

- $\{x^k\}$ *converges to* x^* *R-superlinearly with R-rate (at least)* p *if there exist* $p > 1$ *and* $C \in [0, 1)$ *such that, for sufficiently large* k*, we have*

$$\|x^{k+1} - x^*\|^{1/p^k} \leq C.$$

Sometimes, authors define the convergence rate of an algorithm with respect to the error $e^k = \{f(x^k) - f(x^*)\}$, instead of $e^k = \|x^k - x^*\|$.

Hereinafter, when we analyze the convergence rate of an algorithm, we refer to Q-rate (otherwise explicitly indicated).

2.2 Globally convergent methods with line search

To guarantee global convergence of an algorithm, three main strategies are usually considered in the literature. In particular, we can distinguish the following classes of methods:

- methods that use a *line search* technique, characterized by a suitable choice of a *search direction* $d^k \in \mathbb{R}^n$ and a positive *stepsize* (or *step length*) α^k at every iteration k, in order to compute the next iterate as

$$x^{k+1} = x^k + \alpha^k d^k;$$

- methods that use a *trust-region* technique, where every iterate is obtained by approximately minimizing a quadratic model of the objective function in a specific region around the current point;

- methods that use a *direct search* technique (usually applied in derivative-free optimization), which sample the objective function in a proper neighborhood of the current point.

Here, we focus on algorithms that use a line search technique and we briefly describe some theoretical properties of this class of methods. In Algorithm 2, we provide the scheme of an optimization method with line search.

Algorithm 2 `General optimization algorithm with line search`

0 Choose $x^0 \in \mathbb{R}^n$, set $k = 0$
1 While $\nabla f(x^k) \neq 0$
2 Compute a search direction $d^k \in \mathbb{R}^n$
3 Compute a stepsize $\alpha^k > 0$
4 Set $x^{k+1} = x^k + \alpha^k d^k$
5 Set $k = k + 1$
6 End while

Optimization algorithms with line search are characterized by the choice of the search direction d^k, which can strongly affect the convergence rate of the method.

Generally, d^k must satisfy the condition

$$\nabla f(x^k)^T d^k < 0,$$

that is, d^k must be a descent direction. This ensures that there exists a stepsize $\alpha^k > 0$ that makes the objective function decrease. More formally, there exists $\bar{\alpha} > 0$ such that

$$f(x^k + \alpha d^k) < f(x^k), \quad \forall \alpha \in (0, \bar{\alpha}].$$

The choice of the step length α^k plays a crucial role to ensure global convergence of the algorithm. As to be shown, line search methods must guarantee that α^k satisfy specific theoretical properties.

Before analyzing some general conditions that guarantee global convergence of Algorithm 2, we need to give the definition of *forcing function*.

Definition 4. *A function $\sigma\colon [0,\infty) \to [0,\infty)$ is called forcing function if for every sequence $\{t^h\}$, with $t^h \in [0,\infty)$, such that $\lim_{h\to\infty} \sigma(t^h) = 0$, we have that $\lim_{h\to\infty} t^h = 0$.*

Now, we can report the following result.

Theorem 1. *Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function and let $\{x^k\}$ be the sequence generated by Algorithm 2. Let us assume that $\mathcal{L}(f(x^0))$ is compact, $d^k \neq 0$ if $\nabla f(x^k) \neq 0$ and the following hypotheses are satisfied:*

(i) $f(x^{k+1}) \leq f(x^k)$ for all k;

(ii) if $\nabla f(x^k) \neq 0$ for all k, then

$$\lim_{k\to\infty} \frac{\nabla f(x^k)^T d^k}{\|d^k\|} = 0;$$

(iii) there exists a forcing function σ such that

$$\frac{|\nabla f(x^k)^T d^k|}{\|d^k\|} \geq \sigma(\|\nabla f(x^k)\|), \quad \forall d^k \neq 0.$$

Then, either there exists an index $\nu \geq 0$ such that $x^\nu \in \mathcal{L}(f(x^0))$ and $\nabla f(x^\nu) = 0$, or $\{x^k\}$ is an infinite sequence such that

- *$x^k \in \mathcal{L}(f(x^0))$ for all k,*
- *$\{x^k\}$ attains limit points and every limit point is a stationary point belonging to $\mathcal{L}(f(x^0))$,*
- *the sequence $\{f(x^k)\}$ converges,*
- *$\lim_{k\to\infty} \nabla f(x^k) = 0$.*

Condition (iii) of Theorem 1 can be satisfied by a suitable choice of the search direction. For example, if

$$\nabla f(x)^T d^k \leq -c\|d^k\|\|\nabla f(x^k)\|, \quad c > 0, \tag{2.1}$$

then condition (iii) is fulfilled by using the forcing function $\sigma(t) = ct$.

More in general, if $D^k \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix with bounded eigenvalues, then $d^k = -D^k \nabla f(x^k)$ satisfies (2.1).

Conditions (i)–(ii) of Theorem 1 can be fulfilled by employing a proper line search technique to choose the stepsize α^k.

We also observe that condition (i) imposes a non-increasing monotonicity of $\{f(x^k)\}$. This requirement can be relaxed, allowing the algorithm to produce points that make the objective function increase, in a controlled way. Such methods are called *non-monotone algorithms*. This approach can be convenient, from a computational point of view, especially when particular search directions are employed (e.g., Newton-type directions, described below).

As to be discussed in the next subsection, to guarantee the global convergence of a non-monotone method it is crucial to ensure that $\lim_{k\to\infty} \|x^{k+1} - x^k\| = 0$. In particular, the following result can be stated.

Theorem 2. *Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function and let $\{x^k\}$ be the sequence generated by Algorithm 2. Let us assume that $\mathcal{L}(f(x^0))$ is compact, $d^k \neq 0$ if $\nabla f(x^k) \neq 0$ and the following hypotheses are satisfied*

(i) $\lim_{k\to\infty} \|x^{k+1} - x^k\| = 0$;

(ii) there exists a subsequence $\{x^k\}_K$, with $0 \in K$, and an integer $\bar{M}$ such that

(ii-a) for every k, there exists $j(k) \in K$ such that

$$0 < j(k) - k \leq \bar{M};$$

(ii-b) $\{f(x^k)\}_K$ is a non-increasing monotone sequence;

(ii-c) if $\nabla f(x^k) \neq 0$ for all k, $k \in K$, then

$$\lim_{k\in K,\, k\to\infty} \frac{\nabla f(x^k)^T d^k}{\|d^k\|} = 0;$$

(ii-d) there exists a forcing function σ such that

$$\frac{|\nabla f(x^k)^T d^k|}{\|d^k\|} \geq \sigma(\|\nabla f(x^k)\|), \quad \forall d^k \neq 0,\, k \in K.$$

Then, either there exists an index $\nu \geq 0$ such that $x^\nu \in \mathcal{L}(f(x^0))$ and $\nabla f(x^\nu) = 0$, or $\{x^k\}$ is an infinite sequence such that

- *$\{x^k\}$ attains limit points and every limit point is a stationary point belonging to $\mathcal{L}(f(x^0))$,*
- *the sequence $\{f(x^k)\}$ converges,*
- $\lim_{k\to\infty} \nabla f(x^k) = 0$.

Similarly as for the monotone case, point (ii-d) of Theorem 2 can be fulfilled by a suitable choice of the search direction, while points (i) and (ii-a)–(ii-c) can be satisfied by employing a proper line search technique.

2.3 Line search techniques

In this section, we recall some well-known line search methods that are used to choose the stepsize α^k.

We will show how these techniques, when embedded within the general scheme of Algorithm 2, allow to satisfy points (i)–(ii) of Theorem 1 for the monotone case, and points (i) and (ii-a)–(ii-c) of Theorem 2 for the non-monotone case.

Let us start by considering problems with a strictly convex quadratic objective function. Namely,

$$f(x) = \frac{1}{2}x^T Qx + c^T x,$$

where $Q \in R^{n\times n}$, Q is symmetric, $Q \succ 0$, $c \in \mathbb{R}^n$.

In this case, given $x^k \in \mathbb{R}^n$ and a descent direction $d^k \in \mathbb{R}^n$, it is possible to get the *exact stepsize*, that is, the value α^* that minimizes the objective function along d^k. More formally, $\alpha^* \in \operatorname{Argmin} \varphi(\alpha) = f(x^k + \alpha d^k)$.

Such a stepsize can be analytically computed by imposing $\dfrac{d\varphi(\alpha)}{d\alpha} = 0$. Namely,

$$\alpha^* = -\frac{\nabla f(x^k)^T d^k}{(d^k)^T Q d^k} = -\frac{(Qx^k + c)^T d^k}{(d^k)^T Q d^k}.$$

When the objective function is not strictly convex quadratic, it is not possible to compute the exact stepsize, because this would require to solve a global optimization problem (with 1 variable). Even when the objective function is convex, that approach is not feasible in practice, because the computation of α^* would require an iterative optimization method that converges only asymptotically and then the whole algorithm would not be well defined.

Hence, *inexact line search* methods must be introduced for the general case. They are iterative procedures that must be able to return a value of α^k in a finite number of iterations.

Let us first analyze line search techniques for the monotone case. Roughly speaking, these methods must guarantee that α^k leads to a sufficient decrease in the objective function, while preventing α^k from being too small.

One of the most popular line search algorithms is the *Armijo method*, described in Algorithm 3.

Algorithm 3 `Armijo line search method`

0 Given $\Delta^k > 0$, $\gamma \in (0,1)$, $\delta \in (0,1)$
1 Set $\alpha = \Delta^k$
2 While $f(x^k + \alpha d^k) > f(x^k) + \gamma\alpha\nabla f(x^k)^T d^k$
3 Set $\alpha = \delta\alpha$
4 End while
5 Set $\alpha^k = \alpha$

The Armijo method terminates in a finite number of iterations under the assumption that d^k is a descent direction. The condition

$$f(x^k + \alpha^k d^k) \leq f(x^k) + \gamma\alpha^k \nabla f(x^k)^T d^k \tag{2.2}$$

is often called *Armijo condition* and in practice ensures a sufficient decrease in the objective function. Anyway, we mentioned above that this is not enough by itself to fulfill both points (i) and (ii) of Theorem 1. A further condition to control α^k from below needs to be introduced, as shown in the following theorem.

Theorem 3. *Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function. Let $\{x^k\}$ be the sequence generated by Algorithm 2, where α^k is computed at Step 3 by the Armijo line search method. Let us assume that $\mathcal{L}(f(x^0))$ is compact, $\nabla f(x^k)^T d^k < 0$ for all k and*

$$\Delta^k \geq \frac{1}{\|d^k\|}\sigma\left(\frac{|\nabla f(x^k)^T d^k|}{\|d^k\|}\right), \quad \forall k, \tag{2.3}$$

where $\sigma(\cdot)$ is a forcing function.

Then, at every iteration k, a value $\alpha^k > 0$ is computed in a finite number of steps by the Armijo line search method. Moreover, the sequence $\{x^k\}$ generated by Algorithm 2 is such that

- $f(x^{k+1}) < f(x^k), \quad \forall k;$
- $\lim\limits_{k\to\infty} \dfrac{\nabla f(x^k)^T d^k}{\|d^k\|} = 0.$

It follows that, under the hypotheses of the above theorem, points (i)–(ii) of Theorem 1 are satisfied by employing the Armijo line search.

Now, we briefly discuss condition (2.3), which ensures that α^k is sufficiently large. It is worth noticing that (2.3) can be easily satisfied at every iteration k, by setting $\Delta^k = \dfrac{\rho}{\|d^k\|}$, with $\rho > 0$. In this way, (2.3) trivially holds by using as forcing function the constant function $\sigma(t) = \rho$. Anyway, this choice can be in practice not particularly convenient.

For some algorithms that employ an Armijo-type line search, it is crucial to guarantee that $\Delta^k = 1$ satisfies (2.3) for sufficiently large k, in order to meet some requirements related to the convergence rate (e.g., in Newton-type methods, described below). It is easy to show that a constant value $\Delta^k = \Delta > 0$ satisfies (2.3) if the search direction d^k is chosen such that

$$\nabla f(x^k)^T d^k \leq -c\|\nabla f(x^k)\|^p, \quad c, p > 0.$$

In fact, from the above expression, it follows that

$$\left(\frac{|\nabla f(x^k)^T d^k|}{\|d^k\|}\right)^p \leq \|\nabla f(x^k)^T\|^p \leq \frac{1}{c}|\nabla f(x^k)^T d^k| \leq \frac{M}{c}\|d^k\|,$$

where M is the largest value assumed by $\|\nabla f(x^k)\|$ over $\mathcal{L}(f(x^0))$. Then, (2.3) holds by using as forcing function $\sigma(t) = \dfrac{c\,\Delta}{M}t^p$.

More in general, it is possible to define a set of directions d^k, called *gradient-related directions*, that satisfy

$$\nabla f(x^k)^T d^k \leq -c_1\|\nabla f(x^k)\|^2, \tag{2.4}$$

$$\|d^k\| \leq c_2\|\nabla f(x^k)\|. \tag{2.5}$$

.

The use of gradient-related directions allows to fulfill

- condition (2.3) of Theorem 3 with a constant value $\Delta^k = \Delta > 0$,
- point (iii) of Theorem 1.

In fact, reasoning as above, assumption (2.3) of Theorem 3 is satisfied by using the forcing function $\sigma(t) = \dfrac{c_1 \Delta}{M} t^2$. Moreover, we can write

$$|\nabla f(x^k)^T d^k| \geq c_1 \|\nabla f(x^k)\|^2 \geq \frac{c_1}{c_2} \|\nabla f(x^k)\| \|d^k\|,$$

and then, point (iii) of Theorem 1 is satisfied by using the forcing function $\sigma(t) = \dfrac{c_1}{c_2} t$.

Furthermore, the convergence results stated in Theorem 3 can be easily made more general. In particular, it is possible to show that $\lim\limits_{k\to\infty} \dfrac{\nabla f(x^k)^T d^k}{\|d^k\|} = 0$ still holds for any sequence $\{x^k\}$ such that

$$f(x^{k+1}) \leq f(x^k + \alpha_A^k d^k),$$

where α_A^k is a stepsize computed by the Armijo method under assumption (2.3). Naturally, this choice of the step length satisfies $f(x^{k+1}) < f(x^k)$ and then conditions (i)–(ii) of Theorem 1 are still fulfilled. As a consequence, when $f(x)$ is a strictly convex quadratic function, it is possible to employ the exact stepsize $\alpha^k = \alpha^*$ in Algorithm 2. Moreover, for strictly convex quadratic objective functions, it is easy to show that α^* satisfies the Armijo condition (2.2) if and only if $\gamma \leq 1/2$.

Besides the Armijo method, there are other popular line search techniques that control the step length from below by replacing (2.3) with other requirements. For example, *Goldstein conditions* are based on satisfying

$$\begin{cases} f(x^k + \alpha^k d^k) \leq f(x^k) + \gamma_1 \alpha \nabla f(x^k)^T d^k, \\ f(x^k + \alpha^k d^k) \geq f(x^k) + \gamma_2 \alpha \nabla f(x^k)^T d^k, \end{cases}$$

where $0 < \gamma_1 < \gamma_2 < 1/2$.

Similarly, the *Wolfe conditions* impose some requirements on the curvature of $\varphi(\alpha) = f(x^k + \alpha d^k)$. In particular, we have the *weak Wolfe conditions*:

$$\begin{aligned} f(x^k + \alpha^k d^k) &\leq f(x^k) + \gamma \alpha \nabla f(x^k)^T d^k, \\ \nabla f(x^k + \alpha^k d^k)^T d^k &\geq \rho \nabla f(x^k)^T d^k, \end{aligned}$$

where $0 < \gamma < \rho < 1$; and the *strong Wolfe conditions*:

$$\begin{aligned} f(x^k + \alpha^k d^k) &\leq f(x^k) + \gamma \alpha \nabla f(x^k)^T d^k, \\ |\nabla f(x^k + \alpha^k d^k)^T d^k| &\leq \rho |\nabla f(x^k)^T d^k|, \end{aligned}$$

where $0 < \gamma < \rho < 1$.

In the above relations, we observe that a sufficient decrease in the objective function is guaranteed by the Armijo condition.

Just as for the Armijo method, it is possible to define iterative procedures that, in a finite number of iterations, compute a value α^k satisfying Goldstein, weak Wolfe or strong Wolfe conditions, under the hypothesis that the function $\varphi(\alpha) = f(x^k + \alpha d^k)$ is bounded from below for $\alpha > 0$.

We conclude this section by describing an Armijo-type line search technique for non-monotone methods that satisfies the hypothesis of Theorem 2.

As pointed out in the previous section, a crucial requirement for non-monotone algorithms is that

$$\lim_{k\to\infty} \|x^{k+1} - x^k\| = 0. \tag{2.6}$$

In general, the above relation is not satisfied by the Armijo method described in Algorithm 3, unless further conditions on Δ^k are added, or if the search direction d^k satisfies specific properties. In particular, it easy to show that a gradient-related direction is able to guarantee that (2.6) holds with a fixed $\Delta^k = \Delta > 0$. In fact, assuming that (2.4)–(2.5) are satisfied, we can write

$$|\nabla f(x^k)^T d^k| \geq c_1 \|\nabla f(x^k)\|^2 \geq \frac{c_1}{c_2^2} \|d^k\|^2$$

and then

$$\|d^k\| \leq \frac{c_2^2}{c_1} \frac{|\nabla f(x^k)^T d^k|}{\|d^k\|}.$$

Exploiting the fact that $0 < \alpha^k \leq \Delta$, we obtain

$$\|x^{k+1} - x^k\| = \alpha^k \|d^k\| \leq \frac{c_2^2}{c_1} \Delta \frac{|\nabla f(x^k)^T d^k|}{\|d^k\|}.$$

Recalling that $\lim_{k\to\infty} \dfrac{\nabla f(x^k)^T d^k}{\|d^k\|} = 0$, we get (2.6).

Since non-monotone methods do not require $f(x^k)$ to decrease monotonically, the line search procedure must guarantee a sufficient decrease in the objective function with respect to a *reference value* f_R^k that can be greater than $f(x^k)$.

Different choices of the reference value have been proposed in the literature. A popular way to define f_R^k is the following one:

$$f_R^k = \max_{0 \leq i \leq \min\{k, M\}} \{f(x^{k-i})\},$$

where $M \geq 0$. According to the above formula, f_R^k is the largest value assumed by the objective function in the last M iterations.

In Algorithm 4 we report the scheme of a non-monotone version of the Armijo method, where we have set Δ^k equal to a constant value.

The convergence of the non-monotone Armijo method is stated in the following theorem.

Theorem 4. *Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function. Let $\{x^k\}$ be the sequence generated by Algorithm 2, where α^k is computed at Step 3 by the*

Algorithm 4 `Non-monotone Armijo line search method`

0 Given $\Delta > 0$, $\gamma \in (0,1)$, $\delta \in (0,1)$, $M \geq 0$
1 Let $f_R^k = \max_{0 \leq i \leq \min\{k,M\}} \{f(x^{k-i})\}$
2 Set $\alpha = \Delta$
3 While $f(x^k + \alpha d^k) > f_R^k + \gamma \alpha \nabla f(x^k)^T d^k$
4 Set $\alpha = \delta \alpha$
5 End while
6 Set $\alpha^k = \alpha$

non-monotone Armijo line search method. Let us assume that $\mathcal{L}(f(x^0))$ is compact, $\nabla f(x^k) \neq 0$ for all k and the search direction d^k satisfies

$$\nabla f(x^k)^T d^k \leq -c_1 \|\nabla f(x^k)\|^2,$$
$$\|d(x^k)\| \leq c_2 \|\nabla f(x^k)\|,$$

for all k, with $c_1, c_2 > 0$. Let $K \subseteq \{0, 1, \ldots\}$ be the subset of iteration indices such that

$$\bar{k} \in K \Leftrightarrow f(x^{\bar{k}}) = f_R^k, \text{ for any } k.$$

Then, at every iteration k, a value $\alpha^k > 0$ is computed in a finite number of steps by the non-monotone Armijo line search method. Moreover, the sequence $\{x^k\}$ generated by Algorithm 2 is such that

- $x^k \in \mathcal{L}(f(x^0))$ *for all k;*
- $\{f(x^k)\}_K$ *is monotonically decreasing;*
- *for every k, there exists $j(k) \in K$ such that $0 < j(k) - k \leq M + 1$;*
- $\{f(x^k)\}$ *converges;*
- $\lim_{k \to \infty} \|x^{k+1} - x^k\| = 0$;
- $\lim_{k \to \infty} \dfrac{\nabla f(x^k)^T d^k}{\|d^k\|} = 0.$

Theorem 4 shows that points (i) and (ii-a)–(ii-c) of Theorem 2 are satisfied when α^k is chosen by the above non-monotone Armijo method, if the search direction is gradient-related. Moreover, again from the fact that the search direction is gradient-related, it follows that also point (ii-d) is satisfied and then the global convergence of Algorithm 2 is guaranteed under the hypothesis that $\mathcal{L}(f(x^0))$ is compact.

2.4 Some popular algorithms

In this section, we review some well-known algorithms for unconstrained optimization. In particular, we are concerned with problems of the form

$$\min_x f(x)$$
$$x \in \mathbb{R}^n,$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$.

In the following, we only focus on the methods that will be used in the subsequent chapters. In Subsection 2.4.1, we review the gradient method. In Subsection 2.4.2, we describe the conjugate gradient method to minimize a quadratic function. In Subsection 2.4.3, we focus on Newton's method. Finally, in Subsection 2.4.4 we are concerned with the truncated-Newton method.

2.4.1 Gradient method and extensions

In this subsection we describe the *gradient method* and we mention some extensions of this algorithm. Assuming that the objective function $f(x)$ is continuously differentiable, the gradient method is characterized by the use of the anti-gradient of the objective function as search direction. This is formalized in Algorithm 5.

Algorithm 5 `Gradient method`

0 Choose $x^0 \in \mathbb{R}^n$, set $k = 0$
1 While $\nabla f(x^k) \neq 0$
2 Set $d^k = -\nabla f(x^k)$
3 Compute $\alpha^k > 0$
4 Set $x^{k+1} = x^k + \alpha^k d^k$
5 Set $k = k + 1$
6 End while

The gradient method is also known as the *steepest descent method*, because the (normalized) anti-gradient is the direction that minimizes the derivative directional of f at x^k among all the directions with the same norm. In other words, $d = -\nabla f(x^k)$ is the solution of the following problem:

$$\min_{d \in \mathbb{R}^n} \nabla f(x^k)^T d$$
$$\|d\| = 1.$$

By employing a proper line search technique to choose α^k, the convergence of Algorithm 5 follows from Theorem 1, just observing that the condition expressed in point (iii) of Theorem 1 is satisfied by using the forcing function $\sigma(t) = t$. Formally, we can state the following result.

Theorem 5. *Let $f\colon R^n \to \mathbb{R}$ be a continuously differentiable function and let $\{x^k\}$ be the sequence generated by the gradient method. Let us assume that $\mathcal{L}(f(x^0))$ is compact and that α^k is computed by a line search technique ensuring that*

- $f(x^{k+1}) < f(x^k), \quad$ *if* $\nabla f(x^k) \neq 0$;
- *if* $\nabla f(x^k) \neq 0$ *for all* k, *then* $\lim_{k\to\infty} \dfrac{\nabla f(x^k)^T d^k}{\|d^k\|} = 0$.

Then, either there exists an index $\nu \geq 0$ such that $x^\nu \in \mathcal{L}(f(x^0))$ and $\nabla f(x^\nu) = 0$, or $\{x^k\}$ is an infinite sequence such that every limit point $\bar{x}$ belongs to $\mathcal{L}(f(x^0))$ and $\nabla f(\bar{x}) = 0$.

If $\nabla f(x)$ is Lipschitz continuous, the global convergence of the gradient method is guaranteed even with a constant stepsize, as reported in the following theorem.

Theorem 6. *Let $f\colon R^n \to \mathbb{R}$ be a continuously differentiable function and let $\{x^k\}$ be the sequence generated by the gradient method. Let us assume that $\mathcal{L}(f(x^0))$ is compact and that there exists $L > 0$ such that*

$$\|\nabla f(u) - \nabla f(v)\| \leq L\|u - v\|, \quad \forall u, v \in \mathbb{R}^n.$$

Moreover, let us assume that there exists $\epsilon > 0$ such that the sequence $\{\alpha^k\}$ satisfies

$$\epsilon \leq \alpha^k \leq \frac{2-\epsilon}{L}, \quad \text{for all } k.$$

Then, either there exists an index $\nu \geq 0$ such that $x^\nu \in \mathcal{L}(f(x^0))$ and $\nabla f(x^\nu) = 0$, or $\{x^k\}$ is an infinite sequence such that every limit point $\bar{x}$ belongs to $\mathcal{L}(f(x^0))$ and $\nabla f(\bar{x}) = 0$.

The gradient method is often combined with other algorithms to ensure global convergence (e.g., with Newton-type methods, see below). In particular, if the gradient method is periodically applied to an infinite subsequence of points produced by any descent method, it can be proved that there exist limit points that satisfy stationary conditions.

Despite its simplicity, the gradient method is very slow in practice. To get an estimate of its convergence rate, we can consider the strictly convex quadratic function

$$f(x) = \frac{1}{2}x^T Q x,$$

where $Q \in \mathbb{R}^{n\times n}$ is a symmetric matrix, $Q \succ 0$. Employing the exact stepsize $\alpha^k = -\dfrac{\nabla f(x^k)^T d^k}{(d^k)^T Q (d^k)}$, and indicating with $x^* = 0$ the minimum point of the above function, it is possible to show that

$$\|x^{k+1} - x^*\| \leq \left(\frac{\lambda_M}{\lambda_m}\right)^{1/2} \left(\frac{\lambda_M - \lambda_m}{\lambda_M + \lambda_m}\right) \|x^k - x^*\|,$$

where λ_m and λ_M are the smallest and the largest eigenvalue of Q, respectively,

Then, the convergence rate of the gradient method depends on the ratio $\dfrac{\lambda_M}{\lambda_m}$, that is, performances deteriorate when the Hessian matrix is ill-conditioned.

Anyway, it is possible to show that the gradient method converges to the minimum point of a strictly convex quadratic function in at most n iterations by setting, at every iteration k, the stepsize $\alpha^k = \dfrac{1}{\lambda_k}$, where $\lambda_0, \ldots, \lambda_{n-1}$ are the eigenvalues of the Hessian matrix.

We conclude this subsection by mentioning some extensions of the gradient method. In particular, some authors define *gradient methods* those algorithms that employ a search direction d^k of the following form:

$$d^k = -D^k \nabla f(x^k),$$

where $D^k \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix. From the positive definiteness of D^k, it immediately follows that d^k is a descent direction, since $\nabla f(x^k)^T d^k = -\nabla f(x^k)^T D^k \nabla f(x^k) < 0$.

In Section 2.2, we pointed out that, if D^k has bounded eigenvalues, then such a direction d^k satisfies condition (iii) of Theorem 1.

Finally, among these methods, it is worth to consider the particular case where the search direction is given by

$$d^k = -\frac{1}{\mu^k}\nabla f(x^k),$$

where

$$\mu^k = \frac{(x^k - x^{k-1})^T(\nabla f(x^k) - \nabla f(x^{k-1}))}{\|x^k - x^{k-1}\|^2},$$

or

$$\mu^k = \frac{\|\nabla f(x^k) - \nabla f(x^{k-1})\|^2}{(x^k - x^{k-1})^T(\nabla f(x^k) - \nabla f(x^{k-1}))}.$$

This algorithm is called *Barzilai–Borwein method* (or *spectral gradient method*) and was introduced in [2]. It is widely used for large-scale problems, because it maintains the simplicity of the gradient method, but it is faster in practice.

When the objective function is strictly convex quadratic, it can be shown that μ^k is well defined and positive, and the algorithm converges to the minimum point by employing the stepsize $\alpha^k = 1$ at every iteration k.

In the general case, μ^k may assume unacceptable values and then a check must be introduced to guarantee that $\epsilon \leq \mu^k \leq \frac{1}{\epsilon}$, with $\epsilon > 0$. Moreover, a suitable line search technique must be employed to ensure the convergence of the method.

2.4.2 Conjugate gradient method for quadratic problems

In this subsection, we report one the most popular algorithms to minimize a convex quadratic function, namely the *conjugate gradient method* [44].

Let us start by considering a strictly convex quadratic objective function. Namely, we focus on the following problem:

$$\begin{aligned} &\min_x f(x) = \frac{1}{2}x^T Q x + c^T x \\ &x \in \mathbb{R}^n, \end{aligned} \tag{2.7}$$

where $Q \in R^{n\times n}$, Q is symmetric, $Q \succ 0$, $c \in \mathbb{R}^n$.

First, we give the definition of *conjugacy*.

Definition 5. *Given a symmetric matrix $Q \in R^{n\times n}$, we say that two vectors $d^i, d^j \in \mathbb{R}^n$, $d^i, d^j \neq 0$, are conjugate with respect to Q if*

$$(d^i)^T Q d^j = 0.$$

A fundamental result for the conjugate gradient method is expressed in the following proposition.

Proposition 1. *Given a symmetric matrix $Q \in R^{n\times n}$, $Q \succ 0$, let the vectors $d^0, d^1, \ldots, d^m \in \mathbb{R}^n$ be mutually conjugate with respect to Q. Then, $d^0, d^1, \ldots, d^m$ are linearly independent.*

From the above result, if we have n conjugate vectors $d^0, d^1, \ldots, d^{n-1} \in \mathbb{R}^n$, then we can express the minimum point x^* of problem (2.7) as a linear combination of $d^0, d^1, \ldots, d^{n-1}$, because $\{d^0, d^1, \ldots, d^{n-1}\}$ is a basis of $\mathbb{R}^n$.

In particular, the next proposition claims that, starting from any point $x^0 \in \mathbb{R}^n$, the optimal solution x^* of problem (2.7) can be computed in a finite number of iterations by moving along the directions $d^0, d^1, \ldots, d^{n-1}$ and employing the exact stepsize at each iteration.

Proposition 2. *Let us consider problem* (2.7) *and let x^* be the optimal solution. Let the vectors $d^0, d^1, \ldots, d^{n-1} \in \mathbb{R}^n$ be mutually conjugate with respect to Q. Let $\{x^k\}$ be the sequence defined as*

$$x^{k+1} = x^k + \alpha^k d^k,$$

where x^0 is any point of $\in \mathbb{R}^n$ and

$$\alpha^k = -\frac{\nabla f(x^k)^T d^k}{(d^k)^T Q d^k} = -\frac{(Qx^k + c)^T d^k}{(d^k)^T Q d^k}, \quad k = 0, \ldots, n-1.$$

Then, there exists $m \leq n-1$ such that $\nabla f(x^{m+1}) = 0$.

In general, a set of conjugate directions is not a priori available. The idea behind the conjugate gradient method is to iteratively build conjugate directions by suitably modifying the anti-gradient of the objective function at every iteration. The formal scheme is reported in Algorithm 6.

Algorithm 6 `Conjugate gradient method for a quadratic function`

0 Choose $x^0 \in \mathbb{R}^n$, set $k = 0$, $d^0 = -\nabla f(x^0)$
1 While $\nabla f(x^k) \neq 0$
2 Set

$$\alpha^k = -\frac{\nabla f(x^k)^T d^k}{(d^k)^T Q d^k}$$
$$x^{k+1} = x^k + \alpha d^k$$
$$\beta^{k+1} = \frac{\nabla f(x^{k+1})^T Q d^k}{(d^k)^T Q d^k}$$
$$d^{k+1} = -\nabla f(x^{k+1}) + \beta^{k+1} d^k$$

3 Set $k = k + 1$
4 End while

The finite convergence of the algorithm is stated in the following proposition.

Proposition 3. *Let us consider problem* (2.7) *and let x^* be the optimal solution. Let $\{x^k\}$ be the sequence of points generated by the conjugate gradient method.*

Then,

- $d^k = 0$ *if and only if* $\nabla f(x^k) = 0$*;*
- $\alpha^k = 0$ *if and only if* $\nabla f(x^k) = 0$*;*
- *there exists* $m \leq n-1$ *such that:*
  - *for all* $i = 1, \ldots, m$ *we have*

$$(d^i)^T Q d^j = 0, \quad j = 0, \ldots, i-1;$$

  - $\nabla f(x^{m+1}) = 0$.

Also when $Q \succeq 0$, it can be proved that the conjugate gradient method converges to the minimum point of the objective function (if any) in a finite number of iterations. The algorithm is still well defined, because it is possible to show that $\nabla f(x^k) = 0$ if $(d^k)^T Q d^k = 0$. In particular, the following result holds.

Proposition 4. *Let* $f = \frac{1}{2} x^T Q x + c^T x$*, where* $Q \in \mathbb{R}^{n \times n}$ *is a symmetric matrix,* $Q \succeq 0$ *and* $c \in \mathbb{R}^n$*. Let* $\{x^k\}$ *be the sequence of points generated by the conjugate gradient method. Let us assume that there exists* x^* *such that* $\nabla f(x^*) = 0$*.*

Then,

- $\nabla f(x^k) = 0$ *if* $(d^k)^T Q d^k = 0$*,*
- *there exists* $m \leq n-1$ *such that* $\nabla f(x^{m+1}) = 0$*.*

When Q is not positive semidefinite, the conjugate gradient method can still find a stationary point (if exists) under additional assumption on the curvature of the directions produced by the algorithm, as stated in the next proposition.

Proposition 5. *Let* $f = \frac{1}{2} x^T Q x + c^T x$*, where* $Q \in \mathbb{R}^{n \times n}$ *is a symmetric matrix and* $c \in \mathbb{R}^n$*. Let* $\{x^k\}$ *be the sequence of points generated by the conjugate gradient method. Let us assume that there exists* x^* *such that* $\nabla f(x^*) = 0$ *and every direction* d^k *satisfies*

$$(d^k)^T Q d^k > 0.$$

Then, there exists $m \leq n-1$ *such that* $\nabla f(x^{m+1}) = 0$*.*

Finally, the conjugate gradient method has been also extended to the minimization of non-quadratic functions, by properly modifying the expression of β^{k+1} and introducing a suitable line search technique. In particular, different versions have been proposed, which do not require the computation of the Hessian matrix. A survey of these methods can be found in [42].

2.4.3 Newton's method

In this subsection, we recall *Newton's method* to solve unconstrained optimization problems. This algorithm requires the objective function $f(x)$ to be twice continuously differentiable.

Newton's method produces a sequence of points by minimizing, at every iteration, a quadratic approximation of the objective functions around the current point.

In particular, by a second-order Taylor expansion, for any given $x^k \in \mathbb{R}^n$ we can write

$$f(x^k + s) = f(x^k) + \nabla f(x^k)^T s + \frac{1}{2} s^T \nabla^2 f(x^k) s + r(x^k, s), \quad \forall s \in \mathbb{R}^n,$$

where $\lim_{s \to 0} \dfrac{r(x^k, s)}{\|s\|^2} = 0$.

If $\|s\|$ is sufficiently small, we have that $f(x^k + s)$ can be approximated by the quadratic function

$$q_k(s) = f(x^k) + \nabla f(x^k)^T s + \frac{1}{2} s^T \nabla^2 f(x^k) s.$$

Then, the idea is to calculate the vector s^k that minimizes $q_k(s)$. Assuming that $\nabla^2 f(x^k) \succ 0$, this is equivalent to computing s^k such that

$$\nabla q_k(s^k) = \nabla f(x^k) + \nabla^2 f(x^k) s^k = 0. \tag{2.8}$$

Namely, we can set

$$s^k = -[\nabla^2 f(x^k)]^{-1} \nabla f(x^k)$$

and then

$$x^{k+1} = x^k + s^k.$$

We report the formal scheme of Newton's method in Algorithm 7.

Algorithm 7 `Newton's method`

0 Choose $x^0 \in \mathbb{R}^n$, set $k = 0$
1 While $\nabla f(x^k) \neq 0$
2 Set $s^k = -[\nabla^2 f(x^k)]^{-1} \nabla f(x^k)$
3 Set $x^{k+1} = x^k + s^k$
4 Set $k = k + 1$
5 End while

Under suitable conditions, the local convergence of Newton's method to a stationary point x^* where $\nabla^2 f(x^*)$ is invertible can be proved. So, Newton's method is not globally convergent (according to Definition 1) and, in the general case, the algorithm may not converge and it may not even attain limit points. Moreover, the iterations may not be well defined (if $\nabla^2 f(x^k)$ is singular for some x^k). It is also worth noticing that Newton's method may converge to a maximum point x^* (if $\nabla f(x^*) \prec 0$).

Anyway, the interest in this algorithm is motivated by the fact that it guarantees a superlinear convergence rate, and even a quadratic convergence rate if the Hessian matrix of the objective function is Lipschitz continuous.

These results are formalized in the next theorem.

Theorem 7. *Let $f\colon R^n \to \mathbb{R}$ be a twice continuously differentiable function on an open set $D \subseteq \mathbb{R}^n$. Let us assume that there exists $x^* \in D$ such that $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is invertible.*

Then, there exists an open ball with center x^* and radius $\rho > 0$, denoted with $\mathcal{B}(x^*, \rho)$, such that $\mathcal{B}(x^*, \rho) \subseteq D$ and, if $x^0 \in \mathcal{B}(x^*, \rho)$, then the sequence $\{x^k\}$ generated by Newton's method is well defined, remains in $\mathcal{B}(x^*, \rho)$ and

- $\lim_{k\to\infty} x^k = x^*$;
- $\{x^k\}$ *converges to* x^* *superlinearly;*
- $\{x^k\}$ *converges to* x^* *quadratically if there exists a scalar* $L > 0$ *such that*

$$\|\nabla^2 f(u) - \nabla^2 f(v)\| \leq L\|u - v\|, \quad \forall u, v \in D.$$

Many modifications of Newton's method have been proposed in the literature, in order to make the algorithm globally convergent, while maintaining the same convergence rate. In particular, we can distinguish between globalization methods that use a trust-region strategy and those that employ a line search technique.

For what concerns the latter, a possible approach is to use the anti-gradient of the objective function as search direction if (2.8) does not attain solutions, or if the Newton direction s^k does not satisfy suitable descent conditions. Other approaches properly modify the Hessian matrix of the objective function by factorization methods (e.g., *Cholesky factorization*), in order to obtain a descent direction.

It is worth noticing that Newton's method, as described in Algorithm 7, does not require the use of a stepsize. This is equivalent to saying that it employs a step length $\alpha^k = 1$ at every iteration k. For Newton-type methods that use a line search technique, it is crucial to guarantee that, if the sequence converges to a stationary point where the Hessian matrix of the objective function is positive definite, then the unitary stepsize is accepted by the line search procedure for sufficiently large k. Roughly speaking, this makes such algorithms able to closely mimic Newton's method when the sequence approaches the stationary point, and then, a superlinear convergence rate is still guaranteed.

2.4.4 Truncated-Newton method

In this subsection, we describe one of the most widely used algorithms for large-scale unconstrained optimization, namely, the *truncated-Newton method* [18].

Assuming that the objective function $f(x)$ is twice continuously differentiable, the basic idea behind this method is to solve the Newton system (2.8) approximately, with a growing precision. In particular, at every iteration k, the truncated-Newton method tries to compute a search direction d^k such that

$$\|\nabla f(x^k) + \nabla^2 f(x^k) d^k\| \leq \eta^k \|\nabla f(x^k)\|, \tag{2.9}$$

where $\{\eta^k\}$ is a suitable sequence of positive real numbers.

To highlight the strict connection between Newton's method and the truncated-Newton method, we first report the following result, which ensures local convergence of the truncated-Newton method, under the same hypotheses of Theorem 7.

Theorem 8. *Let $f\colon R^n \to \mathbb{R}$ be a twice continuously differentiable function on an open set $D \subseteq \mathbb{R}^n$. Let us assume that there exists $x^* \in D$ such that $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is invertible. Let $\{x^k\}$ be the sequence defined as*

$$x^{k+1} = x^k + d^k,$$

where d^k satisfies

$$\|\nabla f(x^k) + \nabla^2 f(x^k) d^k\| \leq \eta^k \|\nabla f(x^k)\|.$$

Then, there exist a scalar $\eta > 0$ and an open ball with center x^* and radius $\rho > 0$, denoted with $\mathcal{B}(x^*, \rho)$, such that, if $x^0 \in \mathcal{B}(x^*, \rho)$ and $\eta^k \in [0, \eta]$ for all k, then the sequence $\{x^k\}$ remains in $\mathcal{B}(x^*, \rho)$ and

- $\lim_{k\to\infty} x^k = x^*$;
- *$\{x^k\}$ converges to x^* superlinearly if*

$$\lim_{k\to\infty} \eta^k = 0;$$

- *$\{x^k\}$ converges to x^* quadratically if there exist two scalars $C, L > 0$ such that*

$$\|\nabla^2 f(u) - \nabla^2 f(v)\| \leq L\|u - v\|, \quad \forall u, v \in D,$$
$$\eta^k \leq C\|\nabla f(x^k)\|, \quad \forall k.$$

In order to guarantee global convergence, the truncated-Newton method must be combined with a suitable globalization strategy. Two main approaches are generally followed: one based on the use of a line search technique and one based on the use of a trust-region approach. Here, we report the truncated-Newton method with a line search technique.

First, we observe that Theorem 8 ensures a superlinear convergence rate if $\eta^k \to 0$. A popular choice of η^k is the following:

$$\eta^k = \bar{\eta} \min\{1/(k+1), \|\nabla f(x^k)\|\},$$

where $\bar{\eta}$ is a positive constant.

Moreover, at every iteration k, the search direction d^k can be computed by employing an inner method for minimizing a quadratic function. In fact, by setting

$$q_k(d) = f(x^k) + \nabla f(x^k)^T d + \frac{1}{2} d^T \nabla^2 f(x^k) d, \tag{2.10}$$

solving (2.9) is equivalent to finding a vector d^k such that

$$\|\nabla q(d^k)\| \leq \eta^k \|\nabla f(x^k)\|. \tag{2.11}$$

Namely, d^k is an approximation of a stationary point of the function $q_k(d)$.

Usually, the conjugate gradient method is employed as inner method to compute such a direction d^k.

In Subsection 2.4.2, we pointed out that the conjugate gradient method can find a stationary point of a quadratic function in a finite number of iterations if the algorithm generates all positive curvature directions. In this case, a vector d^k satisfying (2.11) can thus be computed in a finite number of steps.

When the Hessian matrix of the objective function is not positive definite, then even negative curvature directions can be produced by the conjugate gradient method. Therefore, a check is included in the scheme, so that we quit the conjugate gradient method at the first inner iteration i where $(p^{(i)})^T \nabla^2 f(x^k) p^{(i)}$ is not sufficiently large, being $p^{(i)}$ the i-th point produced by the conjugate gradient method to minimize q_k with respect to d.

The whole scheme of the truncated-Newton method is reported in Algorithm 8. In that scheme, at every iteration k, we indicate with $p^{(0)}, p^{(1)}, \ldots, p^{(i)} \in \mathbb{R}^n$, $i < n$, the inner directions produced by the conjugate gradient method, using the origin as starting point.

Assuming that $\mathcal{L}(f(x^0))$ is compact, it can be proved that the search direction d^k produced by Algorithm 8 is gradient-related. In particular, (2.4)–(2.5) are satisfied by setting $c_1 = \min\{1, 1/M\}$, where M is the largest value assumed by $\|\nabla^2 f(x^k)\|$ on $\mathcal{L}(f(x^0))$ and $c_2 = \max\{1, n/\epsilon\}$.

We also notice that the stepsize α^k is computed at Step 3 by performing the Armijo method with the starting stepsize equal to 1. Recalling what has been discussed in Section 2.3, the fact that d^k is gradient-related enables us to guarantee global convergence of the algorithm. In particular, the following result holds.

Theorem 9. *Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be a twice continuously differentiable function, let $\{x^k\}$ be the sequence generated by the truncated-Newton method and let us assume that $\mathcal{L}(f(x^0))$ is compact.*

Then, either there exists an index $\nu \geq 0$ such that $x^\nu \in \mathcal{L}(f(x^0))$ and $\nabla f(x^\nu) = 0$, or $\{x^k\}$ is an infinite sequence such that every limit point $\bar{x}$ belongs to $\mathcal{L}(f(x^0))$ and $\nabla f(\bar{x}) = 0$.

The following theorem states that, under additional assumptions, the truncated-Newton method converges to a stationary point at a superlinear rate, or even at a quadratic rate.

Theorem 10. *Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be a twice continuously differentiable function, let $\{x^k\}$ be the sequence generated by the truncated-Newton method and let us assume that $\{x^k\}$ converges to a point x^* such that $\nabla f(x^*)$ and $\nabla^2 f(x^*) \succ 0$.*

Then,

- *$\{x^k\}$ converges to x^* superlinearly;*
- *$\{x^k\}$ converges to x^* quadratically if there exist two scalars $L, \rho > 0$ such that*

$$\|\nabla^2 f(u) - \nabla^2 f(v)\| \leq L\|u - v\|, \quad \forall u, v \in \mathcal{B}(x^*, \rho).$$

It is worth highlighting that the truncated-Newton method does not require the storage of the entire Hessian matrix, because $\nabla^2 f(x^k)$ always premultiplies a vector $p(i)$ in Algorithm 8. Then, in the practical implementation of the method, it can be very convenient to define a subroutine to compute such matrix-vector products,

Algorithm 8 `Truncated-Newton method with line search`

0 Choose $x^0 \in \mathbb{R}^n$, $\bar{\eta} > 0$, $\epsilon > 0$, $\gamma \in (0,1)$, $\delta \in (0,1)$, set $k = 0$
1 While $\nabla f(x^k) \neq 0$
2 Compute the search direction d^k as follows:
2(*a*) Set $i = 0$, $d^{(0)} = 0$, $p^{(0)} = -\nabla q(d^{(0)}) = -\nabla f(x^k)$
2(*b*) Do
2(*c*) If $(p^{(i)})^T \nabla^2 f(x^k) p^{(i)} \leq \epsilon \|p^{(i)}\|^2$, then set

$$d^k = \begin{cases} -\nabla f(x^k), & \text{if } i = 0, \\ d^{(i)}, & \text{otherwise,} \end{cases}$$

and go to Step 3
2(*d*) Else set

$$\alpha^{(i)} = -\frac{\nabla q(d^{(i)})^T p^{(i)}}{(p^{(i)})^T \nabla^2 f(x^k) p^{(i)}}$$
$$d^{(i+1)} = d^{(i)} + \alpha^{(i)} p^{(i)}$$
$$\nabla q(d^{(i+1)}) = \nabla q(d^{(i)}) + \alpha^{(i)} \nabla^2 f(x^k) p^{(i)}$$

2(*e*) End if
2(*f*) If $\|\nabla q(d^{(i+1)})\| \leq \bar{\eta} \|\nabla f(x^k)\| \min\left\{\frac{1}{k+1}, \|\nabla f(x^k)\|\right\}$, then set

$$d^k = d^{(i+1)}$$

and go to Step 3
2(*g*) Else set

$$\beta^{(i+1)} = \frac{\nabla q(d^{(i+1)})^T \nabla^2 f(x^k) p^{(i)}}{(p^{(i)})^T \nabla^2 f(x^k) p^{(i)}}$$
$$p^{(i+1)} = -\nabla q(d^{(i+1)}) + \beta^{(i+1)} p^{(i)}$$

2(*h*) End if
2(*i*) Set $i = i + 1$
2(*j*) End do
3 Set $\alpha^k = \delta^\nu$, where ν is the smallest nonnegative integer for which

$$f(x^k + \delta^\nu d^k) \leq f(x^k) + \gamma \delta^\nu \nabla f(x^k)^T d^k$$

4 Set $x^k = x^k + \alpha^k d^k$
5 Set $k = k + 1$
6 End while

instead of storing the whole Hessian matrix. This feature makes the truncated-Newton method well suited for large scale problems, where the storage of matrices

should be avoided for computational reasons.

Other variants of the truncated-Newton method have been also proposed in the literature. For example, different termination criteria for the conjugate gradient method can be employed [60] and negative curvature directions can be exploited to be used within a proper line search framework [28, 34, 53]. Moreover, the convergence of the truncated-Newton method was also proved when a non-monotone line search procedure is employed [39, 40]. Finally, a survey on truncated-Newton methods can be found in [59].

Chapter 3

An active-set algorithm for bound-constrained optimization

In this chapter, we describe a two-stage method for solving non-linear optimization problems with bound constraints. The proposed algorithm exploits a suitable technique to estimate the constraints that are active at the stationary point, combining it with a non-monotone line search framework. An in-depth convergence analysis is carried out and numerical results are presented.

The rest of the chapter is organized as follows. In Section 3.1, we introduce the problem, providing a brief overview of the related works. In Section 3.2, we describe our active-set estimate and its theoretical properties. In Section 3.3, we present our two-stage active-set algorithm and carry out the convergence analysis. In Section 3.4, we describe our numerical experience. Finally, some conclusions are drawn in Section 3.5.

3.1 Introduction

We address the solution of bound-constrained (or box-constrained) minimization problems of the form:

$$\begin{aligned} &\min_x f(x) \\ &l \leq x \leq u, \end{aligned} \tag{3.1}$$

where $f \in C^2(\mathbb{R}^n)$; $x, l, u \in \mathbb{R}^n$, and $l < u$ (all the inequalities are valid componentwise).

In the literature, different approaches have been proposed for solving such problems. Among them, we recall trust-region methods (see, e.g., [11, 51]), interior-point methods (see, e.g., [20, 43, 47]), active-set methods (see, e.g., [3, 9, 25, 26, 42, 71]) and second-order methods (see, e.g., [1, 6]).

Even though a large number of different methods is available, there is still a strong interest in developing efficient methods to solve box-constrained problems. This is mainly due to the fact that many real-world applications can be modeled as large-scale problems with bound constraints: for example, in the field of machine learning, linear Support Vector Machines and Logistic Regression models are

very popular tools and they both can be formulated as box-constrained problems. Furthermore, those methods are used as building blocks in many algorithmic frameworks for nonlinearly constrained problems (e.g. in penalty-based approaches).

In particular, active-set algorithms are characterized by the use of suitable techniques to estimate the active constraints at a stationary point. Namely, they try to identify the following set:

$$\{i \in \{1, \dots, n\} : \; x_i^* = l_i, \text{ or } x_i^* = u_i\},$$

where x^* is a stationary point.

Recently, an active-set method, namely the `NMBC` algorithm, was proposed in [16]. `NMBC` algorithm has three main features: it makes use of the technique described in [24] to identify active constraints; it builds up search directions by combining a truncated-Newton strategy (used in the subspace of the non-active constraints) with a Barzilai–Borwein strategy [2] (used in the subspace of the active constraints); it generates a new iterate by means of a non-monotone line search procedure with backtracking.

Even though numerical results reported in [16] were promising, the method has a drawback that might affect its performance in some cases. Indeed, due to the fact that the search direction is given by two different subvectors (the one generated by means of the truncated-Newton strategy and the one obtained by means of the Barzilai–Borwein strategy), we might end up with a badly scaled direction. When dealing with such a direction, finding a good starting stepsize can become pretty hard.

Here, we give a twofold contribution. On the one hand, we describe and analyze an important theoretical feature of the active-set estimate proposed by Facchinei and Lucidi in [24]. In particular, we prove that, under suitable assumptions, a significant reduction in the objective function can be obtained when setting to the bounds all those variables estimated active. In this way, we extend to box-constrained nonlinear problems a similar result already proved in [17] for ℓ_1-regularized least squares problems, and in [8] for quadratic problems with non-negativity constraints.

On the other hand, thanks to the descent property of the active-set estimate, we are able to define a new algorithmic scheme that overcomes the issues described above for the `NMBC` algorithm. More specifically, we define a two-stage algorithmic framework that suitably combines the active-set estimate proposed in [24] with the non-monotone line search procedure described in [16]. In the first stage of our framework, we set the estimated active variables to the corresponding bounds. Then, in the second stage, we generate a search direction in the subspace of the non-active variables (employing a suitably modified truncated-Newton step) to get a new iterate.

There are three main differences between the method we propose here and the one in [16]:

(i) thanks to the two stages, we can get rid of the Barzilai–Borwein step for the active variables, thus avoiding the generation of badly scaled search directions;

(ii) the search direction is computed only in the subspace of the non-active variables, allowing savings in terms of CPU time, especially when dealing with large-scale problems;

(iii) a specific proximity check is included in order to guarantee global convergence of the method. This is crucial, from a theoretical point of view, since we embed the two stages described above within a non-monotone stabilization framework.

Regarding the theoretical properties of the algorithm, we prove that a non-monotone strategy is able to guarantee global convergence to stationary points even if at each iteration a gradient-related direction is generated only in the subspace of the non-active variables. Furthermore, we prove that under standard additional assumptions, the algorithm converges at a superlinear rate.

The notation used in this chapter is that reported in Appendix A. Moreover, in the following we denote by $g(x)$ and $H(x)$ the n gradient vector and the $n \times n$ Hessian matrix of $f(x)$, respectively. Given a vector $v \in \mathbb{R}^n$ and an index set $I \subseteq \{1, \ldots, n\}$, we denote by v_I the subvector with components v_i, $i \in I$. Given a matrix $H \in \mathbb{R}^{n \times n}$, we denote by $H_{I\,I}$ the submatrix with components h_{ij} with $i, j \in I$, and by $\lambda_{\max}$ its largest eigenvalue. Finally, given $x \in \mathbb{R}^n$, we indicate with $[x]^\sharp$ the projection of x onto $[l, u]$, where $l, u \in \mathbb{R}^n$ define the feasible region of problem (3.1).

3.2 Active-set estimate: preliminary results and properties

In this section, we analyze in depth the theoretical properties of the active-set estimate proposed in [24], adapted for the box-constrained case. As we will see later on, the use of a technique to estimate the active constraints plays a crucial role in the development of a theoretically sound and computationally efficient algorithmic framework.

First, we formally provide the definition of stationary point for problem (3.1).

Definition 6. *A point $x^* \in [l, u]$ is called stationary point of problem* (3.1) *if and only if it satisfies the following first-order necessary optimality conditions:*

$$g(x^*) - \lambda^* + \mu^* = 0, \tag{3.2}$$
$$(l_i - x_i^*)\,\lambda_i^* = 0, \qquad i = 1, \ldots, n, \tag{3.3}$$
$$(x_i^* - u_i)\,\mu_i^* = 0, \qquad i = 1, \ldots, n, \tag{3.4}$$
$$\lambda_i^* \geq 0, \quad \mu_i^* \geq 0, \qquad i = 1, \ldots, n, \tag{3.5}$$

where $\lambda^*, \mu^* \in \mathbb{R}^n$ are the KKT multipliers.

These conditions can be equivalently written as:

$$g_i(x^*) \geq 0, \quad \text{if } x_i^* = l_i, \tag{3.6}$$
$$g_i(x^*) \leq 0, \quad \text{if } x_i^* = u_i, \tag{3.7}$$
$$g_i(x^*) = 0, \quad \text{if } l_i < x_i^* < u_i. \tag{3.8}$$

The active-set estimate we consider here takes inspiration from the approach first proposed in [22], and further studied in [24], based on the use of some approximations of KKT multipliers.

Let x be any feasible point, and $\lambda(x)$, $\mu(x)$ be some appropriate approximations of the KKT multipliers λ^* and μ^*. We define the following index subsets:

$$A_l(x) = \{i \in \{1,\dots,n\} : \; l_i \le x_i \le l_i + \epsilon\lambda_i(x),\; g_i(x) > 0\}, \tag{3.9}$$

$$A_u(x) = \{i \in \{1,\dots,n\} : \; u_i - \epsilon\mu_i(x) \le x_i \le u_i,\; g_i(x) < 0\}, \tag{3.10}$$

$$N(x) = \{i \in \{1,\dots,n\} : \; i \notin A_l(x) \cup A_u(x)\}, \tag{3.11}$$

where $\epsilon > 0$.

In particular, $A_l(x)$ and $A_u(x)$ contain the indices of the variables estimated active at the lower bound and the upper bound, respectively. The set $N(x)$ includes the indices of the variables estimated non-active.

In order to explain how we compute $\lambda(x)$ and $\mu(x)$, we need to provide the definition of *multiplier function* for a generic constrained problem.

Definition 7. *Let us consider the following problem:*

$$\begin{aligned} &\min_x f(x) \\ &c_i(x) \le 0, \quad i = 1,\dots,m, \\ &h_j(x) = 0, \quad i = 1,\dots,p, \end{aligned}$$

with $f \in C^1(\mathbb{R}^n)$, $c_i \in C^1(\mathbb{R}^n)$, $i = 1,\dots,m$, and $h_j \in C^1(\mathbb{R}^n)$, $j = 1,\dots,p$. Let x^* be a stationary point, let $\lambda^* \in \mathbb{R}^{m+p}$ be the corresponding KKT multiplier vector and let $\Omega \subseteq \mathbb{R}^n$ be a neighborhood of x^*.

Then, a function $\lambda\colon \Omega \to \mathbb{R}^{m+p}$ is called multiplier function if

- *$\lambda(\cdot)$ is continuous in x^*,*
- *$\lambda(x^*) = \lambda^*$.*

Here, $\lambda(x)$ and $\mu(x)$ are defined as the multiplier functions introduced in [37]. For any feasible point x, the basic idea of this approach is first to solve exactly (3.2), by expressing λ in function of μ:

$$\lambda(\mu(x)) = g(x) + \mu(x).$$

Then, we minimize the error over (3.3)–(3.4), formulating the following least squares problem:

$$\min_{\mu(x)} \sum_{i=1}^{n} \big[(l_i - x_i)\lambda_i(\mu(x))\big]^2 + \sum_{i=1}^{n} \big[(x_i - u_i)\mu_i(x)\big]^2.$$

By solving the above problem, it is possible to compute the functions $\lambda\colon \mathbb{R}^n \to \mathbb{R}^n$ and $\mu\colon \mathbb{R}^n \to \mathbb{R}^n$ as:

$$\lambda_i(x) = \frac{(u_i - x_i)^2}{(l_i - x_i)^2 + (u_i - x_i)^2} g_i(x), \qquad i = 1,\dots,n, \tag{3.12}$$

$$\mu_i(x) = -\frac{(l_i - x_i)^2}{(l_i - x_i)^2 + (u_i - x_i)^2} g_i(x), \quad i = 1,\dots,n. \tag{3.13}$$

By adapting the results shown in [24], we can state the following proposition.

Proposition 6. *If (x^*, λ^*, μ^*) satisfies KKT conditions for problem (3.1), then there exists a neighborhood $\mathcal{B}(x^*, \rho)$ such that*

$$\{i : x_i^* = l_i,\, \lambda_i^* > 0\} \subseteq A_l(x) \subseteq \{i : x_i^* = l_i\},$$

$$\{i : x_i^* = u_i,\, \mu_i^* > 0\} \subseteq A_u(x) \subseteq \{i : x_i^* = u_i\},$$

for each $x \in \mathcal{B}(x^*, \rho)$.

Furthermore, if strict complementarity holds, then

$$\{i : x_i^* = l_i,\, \lambda_i^* > 0\} = A_l(x) = \{i : x_i^* = l_i\},$$

$$\{i : x_i^* = u_i,\, \mu_i^* > 0\} = A_u(x) = \{i : x_i^* = u_i\},$$

for each $x \in \mathcal{B}(x^*, \rho)$.

We notice that stationary points can be characterized by using the active-set estimate, as shown in the next propositions.

Proposition 7. *A point $\bar{x} \in [l, u]$ is a stationary point of problem (3.1) if and only if the following conditions hold:*

$$\max\{l_i - \bar{x}_i,\, -g_i(\bar{x})\} = 0, \quad i \in A_l(\bar{x}), \tag{3.14}$$

$$\max\{\bar{x}_i - u_i,\, g_i(\bar{x})\} = 0, \quad i \in A_u(\bar{x}), \tag{3.15}$$

$$g_i(\bar{x}) = 0, \quad i \in N(\bar{x}). \tag{3.16}$$

Proof. Assume that $\bar{x}$ satisfies (3.14)–(3.16). First, we show that

$$\bar{x}_i = l_i, \quad \text{if } i \in A_l(\bar{x}), \tag{3.17}$$

$$\bar{x}_i = u_i, \quad \text{if } i \in A_u(\bar{x}). \tag{3.18}$$

In order to prove (3.17), assume by contradiction that there exists an index $i \in A_l(\bar{x})$ such that $l_i < \bar{x} \leq l_i + \epsilon\lambda_i(\bar{x})$. It follows that $\lambda_i(\bar{x}) > 0$, and, from (3.12), that $g_i(\bar{x}) > 0$, contradicting (3.14). Then, (3.17) holds. The same reasoning applies to prove (3.18).

Recalling (3.9), we have that $g_i(\bar{x}) > 0$ for all $i \in A_l(\bar{x})$. Combined with (3.17), it means that $\bar{x}_i$ satisfies (3.6) for all $i \in A_l(\bar{x})$. Similarly, since $g_i(\bar{x}) < 0$ for all $i \in A_u(\bar{x})$ and (3.18) holds, then $\bar{x}_i$ satisfies (3.7) for all $i \in A_u(\bar{x})$.

From (3.16), we also have that $\bar{x}_i$ satisfies optimality conditions for all $i \in N(\bar{x})$. Then, $\bar{x}$ is a stationary point.

Now, assume that $\bar{x}$ is a stationary point. First, we consider a generic index i such that $\bar{x}_i = l_i$. For such an index, from (3.6) we get $g_i(\bar{x}) \geq 0$. If $g_i(\bar{x}) > 0$, then, from (3.9), it follows that $i \in A_l(\bar{x})$ and (3.14) is satisfied. Vice versa, if $g_i(\bar{x}) = 0$, then we have that i belongs to $N(\bar{x})$, satisfying (3.16). The same reasoning applies for a generic index i such that $\bar{x}_i = u_i$.

Finally, for every index i such that $l_i < \bar{x}_i < u_i$, from (3.8) we have that $g_i(\bar{x}) = 0$. Then, $\bar{x}$ satisfies (3.14)–(3.16). □

Proposition 8. *Given $\bar{x} \in [l, u]$, assume that*

$$\{i \in A_l(\bar{x}) : \bar{x}_i > l_i\} \cup \{i \in A_u(\bar{x}) : \bar{x}_i < u_i\} = \emptyset. \tag{3.19}$$

Then, $\bar{x}$ is a stationary point of problem (3.1) *if and only if*

$$g_i(\bar{x}) = 0 \text{ for all } i \in N(\bar{x}).$$

Proof. From (3.19) we have

$$\begin{aligned} \bar{x}_i &= l_i, \quad \text{if } i \in A_l(\bar{x}), \\ \bar{x}_i &= u_i, \quad \text{if } i \in A_u(\bar{x}). \end{aligned}$$

Recalling the definition of $A_l(\bar{x})$ and $A_u(\bar{x})$, the previous relations imply that (3.14) and (3.15) are verified. Then, from Proposition 7, $\bar{x}$ is a stationary point if and only if $g_i(\bar{x}) = 0$ for all $i \in N(\bar{x})$. □

Proposition 9. *Given $\bar{x} \in [l, u]$, assume that*

$$g_i(\bar{x}) = 0 \text{ for all } i \in N(\bar{x}). \tag{3.20}$$

Then, $\bar{x}$ is a stationary point of problem (3.1) *if and only if*

$$\{i \in A_l(\bar{x}) : \bar{x}_i > l_i\} \cup \{i \in A_u(\bar{x}) : \bar{x}_i < u_i\} = \emptyset.$$

Proof. If $\{i \in A_l(\bar{x}) \,|\, \bar{x}_i > l_i\} \cup \{i \in A_u(\bar{x}) \,|\, \bar{x}_i < u_i\} = \emptyset$, from the definition of $A_l(\bar{x})$ and $A_u(\bar{x})$ it follows that

$$\begin{aligned} \bar{x}_i = l_i \text{ and } g_i(\bar{x}) > 0, \quad i \in A_l(\bar{x}), \\ \bar{x}_i = u_i \text{ and } g_i(\bar{x}) < 0, \quad i \in A_u(\bar{x}). \end{aligned}$$

Then, conditions (3.6)–(3.8) are verified, and $\bar{x}$ is a stationary point.
Conversely, if $\bar{x}$ is a stationary point, we proceed by contradiction and assume that there exists $\bar{x}_i \in (l_i, u_i)$ such that $i \in A_l(\bar{x}) \cup A_u(\bar{x})$. From the definition of $A_l(\bar{x})$ and $A_u(\bar{x})$, it follows that $g_i(\bar{x}) \neq 0$, violating (3.8) and thus contradicting the fact that $\bar{x}$ is a stationary point. □

3.2.1 Descent property of the active-set

In this subsection, we show that the active-set estimate can be used for computing a point that ensures a sufficient decrease in the objective function simply by fixing the estimated active variables at the bounds.

First, we give an assumption on the parameter ϵ appearing in the definition of the active-set estimates $A_l(x)$ and $A_u(x)$ that will be used to prove the main result in this subsection.

Assumption 1. *Assume that the parameter ϵ appearing in* (3.9) *and* (3.10) *satisfies the following conditions:*

$$\begin{cases} 0 < \epsilon \leq \dfrac{1}{\bar{\lambda}}, & \text{if } \bar{\lambda} > 0, \\ \epsilon > 0, & \text{otherwise} \end{cases} \tag{3.21}$$

where

$$\bar{\lambda} = \max_{x \in [l,u]} \Big\{ \lambda_{max}\big(H(x)\big) \Big\}.$$

Now, we state the main result of the subsection.

Proposition 10. *Let Assumption 1 hold. Let $x \in [l, u]$ be such that*

$$A_l(x) \cup A_u(x) \neq \emptyset,$$

and let $\tilde{x}$ be the point defined as

$$\begin{aligned} \tilde{x}_i &= l_i, \quad i \in A_l(x), \\ \tilde{x}_i &= u_i, \quad i \in A_u(x), \\ \tilde{x}_i &= x_i, \quad i \in N(x), \end{aligned}$$

where $A_l(x)$, $A_u(x)$ and $N(x)$ are the index subsets defined as in (3.9), (3.10) *and* (3.11)*, respectively.*

Then,

$$f(\tilde{x}) - f(x) \leq -\frac{1}{2\epsilon}\|x - \tilde{x}\|^2.$$

Proof. By the second-order mean value theorem, we have

$$f(\tilde{x}) = f(x) + g(x)^T(\tilde{x} - x) + \frac{1}{2}(\tilde{x} - x)^T H(z)(\tilde{x} - x),$$

where $z = x + \xi(\tilde{x} - x)$ for a $\xi \in (0, 1)$. Therefore,

$$f(\tilde{x}) - f(x) \leq g(x)^T(\tilde{x} - x) + \frac{1}{2}\bar{\lambda}\|x - \tilde{x}\|^2. \tag{3.22}$$

Recalling the definition of $\tilde{x}$, we can also write

$$g(x)^T(\tilde{x} - x) = \sum_{i \in A_l(x)} g_i(x)(l_i - x_i) + \sum_{i \in A_u(x)} g_i(x)(u_i - x_i). \tag{3.23}$$

From the definitions of $A_l(x)$ and $A_u(x)$, and recalling (3.12) and (3.13), we have

$$\begin{aligned} g_i(x) &\geq \frac{(x_i - l_i)}{\epsilon}\left[\frac{(l_i - x_i)^2 + (u_i - x_i)^2}{(u_i - x_i)^2}\right], \quad i \in A_l(x), \\ g_i(x) &\leq \frac{(x_i - u_i)}{\epsilon}\left[\frac{(l_i - x_i)^2 + (u_i - x_i)^2}{(l_i - x_i)^2}\right], \quad i \in A_u(x), \end{aligned}$$

and we can write

$$\begin{aligned} g_i(x)(l_i - x_i) &\leq -\frac{1}{\epsilon}(x_i - l_i)^2\left[\frac{(l_i - x_i)^2 + (u_i - x_i)^2}{(u_i - x_i)^2}\right] \leq -\frac{1}{\epsilon}(l_i - x_i)^2, \quad i \in A_l(x), \\ g_i(x)(u_i - x_i) &\leq -\frac{1}{\epsilon}(u_i - x_i)^2\left[\frac{(l_i - x_i)^2 + (u_i - x_i)^2}{(u_i - x_i)^2}\right] \leq -\frac{1}{\epsilon}(u_i - x_i)^2, \quad i \in A_u(x). \end{aligned}$$

Hence, from (3.23), it follows that

$$g(x)^T(\tilde{x} - x) \leq -\frac{1}{\epsilon}\left[\sum_{i \in A_l(x)}(l_i - x_i)^2 + \sum_{i \in A_u(x)}(u_i - x_i)^2\right] = -\frac{1}{\epsilon}\|x - \tilde{x}\|^2. \tag{3.24}$$

Finally, from (3.22) and (3.24), we have

$$f(\tilde{x}) - f(x) \leq \frac{1}{2}\left(\bar{\lambda} - \frac{1}{\epsilon}\right)\|x - \tilde{x}\|^2 - \frac{1}{2\epsilon}\|x - \tilde{x}\|^2 \leq -\frac{1}{2\epsilon}\|x - \tilde{x}\|^2,$$

where the last inequality follows from equation (3.21) in Assumption 1. □

As we already highlighted in the Introduction, Proposition 10 is a non-trivial extension of similar results already proved in the literature.

In particular, here we deal with problems having a general non-convex objective function, while in [17, 8], where a similar analysis was carried out, the authors only considered convex quadratic optimization problems.

3.2.2 Descent property of the non-active set

In this subsection, we show that, thanks to the theoretical properties of the active-set estimate, a sufficient decrease in the objective function can also be obtained by suitably choosing a direction in the subspace of the non-active variables only. Let us consider a search direction satisfying the following conditions:

$$d_i = 0, \quad \forall i \in A_l(x) \cup A_u(x), \tag{3.25}$$

$$d_{N(x)}^T g_{N(x)}(x) \leq -\sigma_1 \|g_{N(x)}(x)\|^2, \tag{3.26}$$

$$\|d_{N(x)}\| \leq \sigma_2 \|g_{N(x)}(x)\|, \tag{3.27}$$

where $\sigma_1, \sigma_2 > 0$. Condition (3.25) ensures that the estimated active variables are not updated when moving along such a direction, while (3.26) and (3.27) imply that d is gradient-related with respect to only the estimated non-active variables.

Given a direction d satisfying (3.25)–(3.27), the following proposition shows that a sufficient decrease in the objective function can be guaranteed by projecting suitable points obtained along d.

Proposition 11. *Given $\bar{x} \in [l, u]$, let us assume that $N(\bar{x}) \neq \emptyset$ and that $g_{N(\bar{x})}(\bar{x}) \neq 0$. Let $\gamma \in (0, 1)$. Then, there exists $\bar{\alpha} > 0$ such that*

$$f(\bar{x}(\alpha)) - f(\bar{x}) \leq \gamma \alpha g(\bar{x})^T d, \quad \forall \alpha \in]0, \bar{\alpha}], \tag{3.28}$$

where $\bar{x}(\alpha) = [\bar{x} + \alpha d]^\sharp$, and d satisfies (3.25)–(3.27) in $\bar{x}$.

Proof. Since the gradient is Lipschitz continuous over $[l, u]$, there exists $L < \infty$ such that for all $s \in [0, 1]$ and for all $\alpha \geq 0$:

$$\|g(\bar{x}) - g(\bar{x} - s[\bar{x} - \bar{x}(\alpha)])\| \leq sL\|\bar{x} - \bar{x}(\alpha)\|, \quad \forall \bar{x} \in [l, u].$$

By the mean value theorem, we have:

$$\begin{aligned} f(\bar{x}(\alpha)) - f(\bar{x}) &= g(\bar{x})^T(\bar{x}(\alpha) - x) + \int_0^1 \left(g(\bar{x} - s[\bar{x} - \bar{x}(\alpha)]) - g(\bar{x})\right)^T (\bar{x}(\alpha) - \bar{x})\, ds \\ &\leq g(\bar{x})^T(\bar{x}(\alpha) - \bar{x}) + \|\bar{x}(\alpha) - \bar{x}\| \int_0^1 sL\|\bar{x}(\alpha) - \bar{x}\|\, ds \\ &= g(\bar{x})^T(\bar{x}(\alpha) - \bar{x}) + \frac{L}{2}\|\bar{x}(\alpha) - \bar{x}\|^2, \quad \forall \alpha \geq 0. \end{aligned} \tag{3.29}$$

Moreover, as the gradient is continuous and the feasible set is compact, there exists $M > 0$ such that

$$\|g(\bar{x})\| \leq M, \quad \forall \bar{x} \in [l, u]. \tag{3.30}$$

From (3.25), (3.27) and (3.30), we can write

$$d_i \leq \|d\| \leq \sigma_2 \|g(\bar{x})\| \leq \sigma_2 M, \quad \forall \bar{x} \in [l, u], \quad \forall i = 1, \ldots, n.$$

Now, let us define $\theta_1, \ldots, \theta_n$ as:

$$\theta_i = \begin{cases} \min\{\bar{x}_i - l_i, u_i - \bar{x}_i\} & \text{if } l_i < \bar{x}_i < u_i, \\ u_i - l_i & \text{otherwise,} \end{cases} \qquad i = 1, \ldots, n.$$

We set

$$\tilde{\theta} = \min_{i=1,\ldots,n} \frac{\theta_i}{2},$$

and define $\hat{\alpha}$ as follows:

$$\hat{\alpha} = \frac{\tilde{\theta}}{\sigma_2 M}.$$

In the following, we want to majorize the right-hand-side term of (3.29). First, we consider the term $g(\bar{x})^T(\bar{x}(\alpha) - x)$. We distinguish three cases:

(i) $i \in N(\bar{x})$ such that $l_i < \bar{x}_i < u_i$. We distinguish two subcases:

- if $d_i \geq 0$:

$$l_i < \bar{x}_i + \alpha d_i \leq \bar{x}_i + \frac{\tilde{\theta}}{\sigma_2 M} d_i \leq x_i + \tilde{\theta} < u_i, \qquad \forall \alpha \in (0, \hat{\alpha}];$$

- else, if $d_i < 0$:

$$u_i > \bar{x}_i + \alpha d_i \geq \bar{x}_i + \frac{\tilde{\theta}}{\sigma_2 M} d_i \geq \bar{x}_i - \tilde{\theta} > l_i, \qquad \forall \alpha \in (0, \hat{\alpha}].$$

So, we have

$$\bar{x}_i(\alpha) = \bar{x}_i + \alpha d_i, \quad \forall \alpha \in (0, \hat{\alpha}],$$

which implies

$$g_i(\bar{x})(\bar{x}_i(\alpha) - \bar{x}_i) = \alpha g_i(\bar{x}) d_i, \quad \forall \alpha \in (0, \hat{\alpha}]. \tag{3.31}$$

(ii) $i \in N(\bar{x})$ such that $\bar{x}_i = l_i$. Recalling the definition of $N(x)$, it follows that $g_i(\bar{x}) \leq 0$. We distinguish two subcases:

- if $d_i \geq 0$:

$$l_i \leq \bar{x}_i + \alpha d_i \leq \bar{x}_i + \frac{\tilde{\theta}}{\sigma_2 M} d_i \leq \bar{x}_i + \tilde{\theta} < u_i, \qquad \forall \alpha \in (0, \hat{\alpha}],$$

and then

$$\bar{x}_i(\alpha) = \bar{x}_i + \alpha d_i, \quad \forall \alpha \in (0, \hat{\alpha}],$$

which implies

$$g_i(\bar{x})(\bar{x}_i(\alpha) - \bar{x}_i) = \alpha g_i(\bar{x}) d_i, \quad \forall \alpha \in (0, \hat{\alpha}]; \tag{3.32}$$

- else, if $d_i < 0$:

$$\bar{x}_i(\alpha) = \bar{x}_i, \quad \forall \alpha > 0,$$

and then

$$0 = g_i(\bar{x})(\bar{x}_i(\alpha) - \bar{x}_i) \leq \alpha g_i(\bar{x}) d_i, \quad \forall \alpha > 0. \tag{3.33}$$

(iii) $i \in N(\bar{x})$ such that $\bar{x}_i = u_i$. Following the same reasonings done in the previous step, we have that

- if $d_i \leq 0$:

$$g_i(\bar{x})(\bar{x}_i(\alpha) - \bar{x}_i) = \alpha g_i(\bar{x}) d_i, \quad \forall \alpha \in (0, \hat{\alpha}]; \tag{3.34}$$

- else, if $d_i > 0$:

$$0 = g_i(\bar{x})(\bar{x}_i(\alpha) - \bar{x}_i) \leq \alpha g_i(\bar{x}) d_i, \quad \forall \alpha > 0. \tag{3.35}$$

From (3.25), (3.31), (3.32), (3.33), (3.34) and (3.35), we obtain

$$\begin{aligned} g(\bar{x})^T(\bar{x}(\alpha) - \bar{x}) &= \sum_{i \in N(\bar{x})} g_i(\bar{x})(\bar{x}_i(\alpha) - \bar{x}) \\ &\leq \alpha \sum_{i \in N(\bar{x})} g_i(\bar{x}) d_i = \alpha g_{N(\bar{x})}(\bar{x})^T d_{N(\bar{x})}, \quad \forall \alpha \in (0, \hat{\alpha}]. \end{aligned} \tag{3.36}$$

Now, we consider the term $\frac{L}{2}\|\bar{x}(\alpha) - \bar{x}\|^2$. For every $i \in N(\bar{x})$ such that $d_i \leq 0$, we have that $0 \leq \bar{x}_i - \bar{x}_i(\alpha) \leq -\alpha d_i$ holds for all $\alpha > 0$. Therefore,

$$(\bar{x}_i - \bar{x}_i(\alpha))^2 \leq \alpha^2 d_i^2, \quad \forall \alpha > 0. \tag{3.37}$$

Else, for every $i \in N(\bar{x})$ such that $d_i > 0$, we have that $0 \leq \bar{x}_i(\alpha) - \bar{x}_i \leq \alpha d_i$ holds for all $\alpha > 0$. Therefore,

$$0 \leq (\bar{x}_i(\alpha) - \bar{x}_i)^2 \leq \alpha^2 d_i^2, \quad \forall \alpha > 0. \tag{3.38}$$

Recalling (3.25), from (3.37) and (3.38) we obtain

$$\|\bar{x}(\alpha) - \bar{x}\|^2 \leq \alpha^2 \|d_{N(\bar{x})}\|^2, \quad \forall \alpha > 0.$$

Using (3.26) and (3.27), we get

$$\begin{aligned} \|\bar{x}(\alpha) - \bar{x}\|^2 &\leq \alpha^2 \|d_{N(\bar{x})}\|^2 \leq \alpha^2 \sigma_2^2 \|g_{N(\bar{x})}(\bar{x})\|^2 \\ &\leq -\alpha^2 \frac{\sigma_2^2}{\sigma_1} g_{N(\bar{x})}(\bar{x})^T d_{N(\bar{x})}, \quad \forall \alpha > 0. \end{aligned} \tag{3.39}$$

From (3.25), (3.29), (3.36) and (3.39), we can write

$$\begin{aligned} f(\bar{x}(\alpha)) - f(\bar{x}) &\leq \alpha \left(1 - \alpha \frac{L\sigma_2^2}{2\sigma_1}\right) g_{N(\bar{x})}(\bar{x})^T d_{N(\bar{x})} \\ &= \alpha \left(1 - \alpha \frac{L\sigma_2^2}{2\sigma_1}\right) g(\bar{x})^T d, \quad \forall \alpha \in (0, \hat{\alpha}]. \end{aligned}$$

It follows that (3.28) is satisfied by choosing $\bar{\alpha}$ such that

$$1 - \bar{\alpha}\frac{L\sigma_2^2}{2\sigma_1} \geq \gamma,$$
$$\bar{\alpha} \in (0, \hat{\alpha}].$$

Thus, the proof is completed defining

$$\bar{\alpha} = \min\left\{\hat{\alpha}, \frac{2\sigma_1(1-\gamma)}{L\sigma_2^2}\right\}.$$

□

3.3 ASA-BCP: the proposed active-set algorithm

In this section, we describe a new algorithmic framework for box-constrained problems, that we call *Active-Set Algorithm for Box-Constrained Problems* (ASA-BCP). Its distinguishing feature is the presence of two stages at each iteration, which enable us to separately handle active and non-active variables.

In particular, in the first stage the estimated active variables can be set at the corresponding bounds, thus producing a new point. After updating the active-set estimates, in the second stage a search direction can be computed in the subspace of the estimated non-active variables and a new iterate can be computed by performing a suitable line search.

Since a monotone line search could prevent from fully exploiting the efficiency of particular search directions, a non-monotone stabilization technique (see e.g. [40] and [76]) could be used in the second stage of our framework. Such a non-monotone procedure, in practice, usually guarantees significant savings in terms of functions evaluations and CPU time.

Here, we employ the approach proposed in [40], characterized by two main features: the use of a non-monotone line search and the possibility to accept the unitary stepsize without computing the objective function if the search direction satisfies a specific test. In particular, the presence of iterations at which the objective function is not computed seems, on the one hand, to be crucial in terms of efficiency (see [40]), but, on the other hand, it makes more difficult guaranteeing the global convergence of the method. This is why proper tools, i.e. suitable tests on the search direction, are included in the algorithmic scheme.

The same non-monotone strategy was also used in NMBC [16] to enforce global convergence. As mentioned above, the main difference from NMBC is the two-stage nature of the algorithm. Then, a non-trivial effort is required to combine it with non-monotonicity. In particular, the estimated active variables are updated simply by fixing them at the bounds, that is, they are not moved along a gradient-related direction. Following the idea of not computing the objective function at every iteration, the first stage needs a further control on the proximity of the generated point.

Before reporting the formal algorithmic scheme of ASA-BCP, we give a sketch of it, indicating with f_R a reference value of the objective function that is updated throughout the procedure. Different criteria were proposed in the literature to

choose this value (see e.g. [76]). Here, we take f_R as the maximum among the last M function evaluations, where M is a nonnegative parameter.

- At every iteration k, starting from the non-stationary point x^k, the algorithm fixes the estimated active variables at the corresponding bounds, thus producing the new point $\tilde{x}^k$. In particular, the sets

$$A_l^k = A_l(x^k), \quad A_u^k = A_u(x^k) \quad \text{and} \quad N^k = N(x^k) \tag{3.40}$$

  are computed and the point $\tilde{x}^k$ is produced by setting

$$\tilde{x}^k_{A_l^k} = l_{A_l^k}, \quad \tilde{x}^k_{A_u^k} = u_{A_u^k} \quad \text{and} \quad \tilde{x}^k_{N^k} = x^k_{N^k}.$$

- Afterward, a check is executed to verify if the new point $\tilde{x}^k$ is sufficiently close to x^k. If this is the case, the point $\tilde{x}^k$ is accepted. Otherwise, an objective function check is executed and two further cases are possible: if the objective function is lower than the reference value f_R, then we accept the point $\tilde{x}^k$; otherwise the algorithm sets $\tilde{x}^k$ by backtracking to the last good point (i.e., the point $\tilde{x}$ that produced the last f_R).

- At this point, the active and non-active sets are updated considering the information related to $\tilde{x}^k$, i.e., we build

$$\tilde{A}_l^k = A_l(\tilde{x}^k), \quad \tilde{A}_u^k = A_u(\tilde{x}^k) \quad \text{and} \quad \tilde{N}^k = N(\tilde{x}^k). \tag{3.41}$$

  A search direction d^k is then computed: we set $d^k_{\tilde{A}^k} = 0$, with $\tilde{A}^k = \tilde{A}_l^k \cup \tilde{A}_u^k$, and calculate $d^k_{\tilde{N}^k}$ by means of a modified truncated-Newton step (see e.g. [18] for further details on truncated-Newton approaches).

- Once d^k is computed, a non-monotone stabilization strategy, inspired by the one proposed in [40], is used to generate the new iterate. In particular, the algorithm first checks if $\|d^k\|$ is sufficiently small. If this is the case, the unitary stepsize is accepted, and we set

$$x^{k+1} = [\tilde{x}^k + d^k]^\sharp$$

  without computing the related objective function value and start a new iteration.
  Otherwise, an objective function check is executed and two further cases are possible: if the objective function is greater than or equal to the reference value f_R, then we backtrack to the last good point and take the related search direction; otherwise we continue with the current point. Finally, a non-monotone line search is performed in order to get a stepsize α^k and generate

$$x^{k+1} = [\tilde{x}^k + \alpha^k d^k]^\sharp.$$

- After a prefixed number of iterations without calculating the objective function, a check is executed to verify if the objective function is lower than the reference value f_R. If this is not the case, a backtracking and a non-monotone line search are executed.

The *non-monotone line search* used in the algorithm is the same as the one described in e.g. [16]. It sets $\alpha^k = \delta^\nu$, where ν is the smallest nonnegative integer for which

$$f([\tilde{x}^k + \delta^\nu d^k]^\sharp) \leq f_R + \gamma \delta^\nu g(\tilde{x}^k)^T d^k, \tag{3.42}$$

with $\delta \in (0,1)$ and $\gamma \in (0, \frac{1}{2})$.

The formal scheme of ASA-BCP is reported in Algorithm 9. In particular, At Step 10, 17 and 25 there is the update of the reference value of the non-monotone line search f_R^j: we set $j = j+1$, $l^j = k$ and the reference value is updated according to the formula

$$f_R^j = \max_{0 \leq i \leq \min\{j,M\}} \{f^{j-i}\}.$$

Remark 1. *From Proposition 8 and 9 it follows that ASA-BCP is well defined, in the sense that at the k-th iteration it produces a new point $x^{k+1} \neq x^k$ if and only if x^k is non-stationary.*

Hereinafter, we indicate the active-set estimates in x^k and $\tilde{x}^k$ with the notation used in (3.40) and in (3.41), respectively.

3.3.1 Analysis of convergence

In this subsection, we analyze the convergence properties of ASA-BCP, to show that the method either computes a stationary point in a finite number of iterations, or every limit point produced by the algorithm is stationary.

To this aim, we need to establish some intermediate results. We start by stating the following lemmas.

Lemma 1. *Let Assumption 1 hold. Suppose that ASA-BCP produces an infinite sequence $\{x^k\}$, then*

(i) $\{f_R^j\}$ is non-increasing and converges to a value $\bar{f}_R$;

(ii) for any fixed $j \geq 0$ we have:

$$f_R^h < f_R^j, \quad \forall h > j + M.$$

Proof. From the instructions of the algorithm we have that, for every $j \geq 0$, $f_R^j = \max_{0 \leq i \leq m^j} f^{j-i}$. Since $m^{j+1} \leq m^j + 1$ we can write

$$\begin{aligned} f_R^{j+1} &= \max_{0 \leq i \leq m^{j+1}} f(\tilde{x}^{l^{(j+1-i)}}) \leq \max_{0 \leq i \leq m^j+1} f(\tilde{x}^{l^{(j+1-i)}}) \\ &= \max\left\{ f(\tilde{x}^{l^{(j+1)}}), \max_{0 \leq i \leq m^j} f^{j-i} \right\} = \max\left\{ f(\tilde{x}^{l^{(j+1)}}), f_R^j \right\}. \end{aligned}$$

As $f(\tilde{x}^{l^{(j+1)}}) < f_R^j$, the above relation implies that

$$f_R^{j+1} \leq f_R^j \leq f_R^0 \leq f(x^0),$$

Algorithm 9 `ASA-BCP`

0 Choose $x^0 \in [l,u]$, fix $Z \geq 1$, $M \geq 0$, $\Delta_0 \geq 0$, $\beta \in (0,1)$, $\delta \in (0,1)$, $\gamma \in (0, \frac{1}{2})$, $k = 0$, $j = -1$, $l^{-1} = -1$, $f_R^{-1} = f^{-1} = f(x^0)$, $\Delta = \tilde{\Delta} = \Delta_0$, *checkpoint* $= true$
1 While x^k is a non-stationary point for problem (3.1)
2 Compute $A_l^k := A_l(x^k)$, $A_u^k := A_u(x^k)$ and $N^k := N(x^k)$
3 Set $\tilde{x}^k_{A_l^k} = l_{A_l^k}$, $\tilde{x}^k_{A_u^k} = u_{A_u^k}$ and $\tilde{x}^k_{N^k} = x^k_{N^k}$
4 If $\|\tilde{x}^k - x^k\| \leq \tilde{\Delta}$, then set $\tilde{\Delta} = \beta\tilde{\Delta}$
5 Else compute $f(x^k)$
6 If $f(x^k) \geq f_R^j$, then backtrack to $\tilde{x}^{l^j}$, set $k = l^j$ and go to Step 28
7 End if
8 Compute $\tilde{A}_l^k := A_l(\tilde{x}^k)$, $\tilde{A}_u^k := A_u(\tilde{x}^k)$ and $\tilde{N}^k := N(\tilde{x}^k)$
9 If $\tilde{N}^k \neq \emptyset$ and $g_{\tilde{N}^k}(\tilde{x}^k) \neq 0$
10 If *checkpoint* $= true$, then compute $f(\tilde{x}^k)$ and update f_R^j
11 Set *checkpoint* $= false$
12 End if
13 Set $d^k_{\tilde{A}_l^k} = 0$, $d^k_{\tilde{A}_u^k} = 0$, compute a gradient-related direction $d^k_{\tilde{N}^k}$ in $\tilde{x}^k$
14 If $k \geq l^j + Z$, then compute $f(\tilde{x}^k)$
15 If $f(\tilde{x}^k) \geq f_R^j$
16 Backtrack to $\tilde{x}^{l^j}$, set $d^k = d^{l^j}$, $k = l^j$ and go to Step 28
17 Else update f_R^j
18 End if
19 End if
20 If $\|d^k_{\tilde{N}^k}\| \leq \Delta$
21 Set $\alpha^k = 1$, $x^{k+1} = [\tilde{x}^k + \alpha^k d^k]^\sharp$, $\Delta = \beta\Delta$, $k = k+1$
22 Else if $k \neq l^j$, then compute $f(\tilde{x}^k)$
23 If $f(\tilde{x}^k) \geq f_R^j$
24 Backtrack to $\tilde{x}^{l^j}$, set $d^k = d^{l^j}$, $k = l^j$ and go to Step 28
25 Else update f_R^j
26 End if
27 End if
28 Set $\alpha^k = \delta^\nu$, where ν is the smallest nonnegative integer for which

$$f([\tilde{x}^k + \delta^\nu d^k]^\sharp) \leq f_R^j + \gamma\delta^\nu g(\tilde{x}^k)^T d^k$$

29 Set $x^{k+1} = [\tilde{x}^k + \alpha^k d^k]^\sharp$, $k = k+1$, *checkpoint* $= true$
30 Else
31 Set $\alpha^k = 0$, $d^k = 0$, $x^{k+1} = [\tilde{x}^k + \alpha^k d^k]^\sharp = \tilde{x}^k$, $k = k+1$
32 End if
33 End while

which proves that $\{f_R^j\}$ is non-increasing. Since $\tilde{x}^{l^j}$ is feasible, the non-increasing sequence $\{f_R^j\}$ is bounded from below, hence

$$\lim_{j\to\infty} f_R^j = \bar{f}_R,$$

which proves (i). Point (ii) follows from the relation $f(\tilde{x}^{l^{(h+1)}}) < f_R^h$, and from the fact that in the algorithm f_R^h is calculated considering at most $M+1$ values of the objective function. □

Lemma 2. *Let Assumption 1 hold. Suppose that ASA-BCP produces an infinite sequence $\{x^k\}$ and an infinite sequence $\{\tilde{x}^k\}$. For any given value of k, let $q(k)$ be the index such that*

$$q(k) = \max\{j : l^j \le k\}.$$

Then, there exists a sequence $\{\tilde{x}^{s(j)}\}$ and an integer L satisfying the following conditions:

(i) $f_R^j = f(\tilde{x}^{s(j)})$

(ii) for any integer k, there exist an index h^k and an index j^k such that:

$$0 < h^k - k \le L, \qquad h^k = s(j^k),$$
$$f_R^{j^k} = f(\tilde{x}^{h^k}) < f_R^{q(k)}.$$

Proof. The proof follows from Lemma 2 in [40] taking into account that for any iteration index k, there exists an integer $\tilde{L}$ such that the condition of Step 9 is satisfied within the $(k+\tilde{L})$-th iteration. In fact, assume by contradiction that it is not true. If Step 9 is not satisfied at a generic iteration k, then $x^{k+1} = \tilde{x}^k$. Since the sequences $\{x^k\}$ and $\{\tilde{x}^k\}$ are infinite, Proposition 9 implies that $\tilde{x}^{k+1} \neq x^{k+1}$ and that the objective function strictly decreases. Repeating this procedure for an infinite number of steps, an infinite sequence of distinct points $\{x^{k+1}, x^{k+2}, \ldots\}$ is produced, where these points differ from each other only for the values of the variables at the bounds. Since the number of variables is finite, this produces a contradiction. □

Lemma 3. *Let Assumption 1 hold. Suppose that ASA-BCP produces an infinite sequence $\{x^k\}$ and an infinite sequence $\{\tilde{x}^k\}$. Then,*

$$\lim_{k\to\infty} f(x^k) = \lim_{k\to\infty} f(\tilde{x}^k) = \lim_{j\to\infty} f_R^j = \bar{f}_R, \tag{3.43}$$

$$\lim_{k\to\infty} \|x^{k+1} - \tilde{x}^k\| = \lim_{k\to\infty} \alpha^k \|d^k\| = 0, \tag{3.44}$$

$$\lim_{k\to\infty} \|\tilde{x}^k - x^k\| = 0. \tag{3.45}$$

Proof. We build two different partitions of the iterations indices to analyze the computation of x^{k+1} from $\tilde{x}^k$ and that of $\tilde{x}^k$ from x^k, respectively. From the instructions of the algorithm, it follows that x^{k+1} can be computed at Step 21, Step 29

or Step 31. Let us consider the following subset of iteration indices:

$$\mathcal{K}_1 = \{k\colon x^{k+1} \text{ is computed at Step 21}\},$$
$$\mathcal{K}_2 = \{k\colon x^{k+1} \text{ is computed at Step 29}\},$$
$$\mathcal{K}_3 = \{k\colon x^{k+1} \text{ is computed at Step 31}\}.$$

Then, we have

$$\mathcal{K}_1 \cup \mathcal{K}_2 \cup \mathcal{K}_3 = \{0, 1, \ldots\}.$$

As regards the computation of $\tilde{x}^k$, we distinguish two further subsets of iterations indices:

$$\mathcal{K}_4 = \{k\colon \tilde{x}^k \text{ satisfies the test at Step 4}\},$$
$$\mathcal{K}_5 = \{k\colon \tilde{x}^k \text{ does not satisfy the test at Step 4}\}.$$

Then, we have

$$\mathcal{K}_4 \cup \mathcal{K}_5 = \{0, 1, \ldots\}.$$

Preliminarily, we point out some properties of the above subsequences. The subsequence $\{\tilde{x}^k\}_{\mathcal{K}_1}$ satisfies

$$\|x^{k+1} - \tilde{x}^k\| = \alpha^k \|d^k\| = \|d^k\| \leq \beta^t \Delta_0, \quad k \in \mathcal{K}_1,$$

where the integer t increases with $k \in \mathcal{K}_1$. Since $\beta \in (0, 1)$, if $\mathcal{K}_1$ is infinite we have

$$\lim_{k\to\infty,\, k\in\mathcal{K}_1} \|x^{k+1} - \tilde{x}^k\| = \lim_{k\to\infty,\, k\in\mathcal{K}_1} \alpha^k \|d^k\| = 0. \tag{3.46}$$

Moreover, since $\alpha^k = 0$ and $d^k = 0$ for all $k \in \mathcal{K}_3$, if $\mathcal{K}_3$ is infinite we have

$$\lim_{k\to\infty,\, k\in\mathcal{K}_3} \|x^{k+1} - \tilde{x}^k\| = \lim_{k\to\infty,\, k\in\mathcal{K}_3} \alpha^k \|d^k\| = 0. \tag{3.47}$$

The subsequence $\{\tilde{x}^k\}_{\mathcal{K}_4}$ satisfies

$$\|\tilde{x}^k - x^k\| \leq \beta^t \tilde{\Delta}_0, \quad k \in \mathcal{K}_4,$$

where the integer t increases with $k \in \mathcal{K}_4$. Since $\beta \in (0, 1)$, if $\mathcal{K}_4$ is infinite we have

$$\lim_{k\to\infty,\, k\in\mathcal{K}_4} \|\tilde{x}^k - x^k\| = 0. \tag{3.48}$$

Now we prove (3.43). Let $s(j)$, h^k and q^k be the indices defined in Lemma 2. We show that for any fixed integer $i \geq 1$, the following relations hold:

$$\lim_{j\to\infty} \|\tilde{x}^{s(j)-i+1} - x^{s(j)-i+1}\| = 0, \tag{3.49}$$

$$\lim_{j\to\infty} \|x^{s(j)-i+1} - \tilde{x}^{s(j)-i}\| = \lim_{j\to\infty} \alpha^{s(j)-i} \|d^{s(j)-i}\| = 0, \tag{3.50}$$

$$\lim_{j\to\infty} f(x^{s(j)-i+1}) = \bar{f}_R, \tag{3.51}$$

$$\lim_{j\to\infty} f(\tilde{x}^{s(j)-i}) = \bar{f}_R. \tag{3.52}$$

Without loss of generality, we assume that j is large enough to avoid the occurrence of negative apices. We proceed by induction and first show that (3.49)–(3.52) hold for $i = 1$. If $s(j) \in \mathcal{K}_4$, relations (3.49) and (3.51) follow from (3.48) and the continuity of the objective function. If $s(j) \in \mathcal{K}_5$, from the instructions of the algorithm, and taking into account Proposition 10, we get

$$f_R^j = f(\tilde{x}^{s(j)}) \leq f(x^{s(j)}) - \frac{1}{2\epsilon}\|x^{s(j)} - \tilde{x}^{s(j)}\|^2 < f_R^{j-1},$$

from which we get

$$f_R^j = f(\tilde{x}^{s(j)}) \leq f(x^{s(j)}) < f_R^{j-1}$$

and then, from point (i) of Lemma 1, it follows that

$$\lim_{j\to\infty} f(\tilde{x}^{s(j)}) = \lim_{j\to\infty} f(x^{s(j)}) = \bar{f}_R, \tag{3.53}$$

which proves (3.51) for $i = 1$. From the above relation, and by exploiting Proposition 10 again, we have that

$$\lim_{j\to\infty} \left(f(\tilde{x}^{s(j)}) - f(x^{s(j)})\right) \leq \lim_{j\to\infty} -\frac{1}{2\epsilon}\|x^{s(j)} - \tilde{x}^{s(j)}\|^2.$$

and then (3.49) holds for $i = 1$.
If $s(j)-1 \in \mathcal{K}_1 \cup \mathcal{K}_3$, from (3.46) and (3.47) it is straightforward to verify that (3.50) holds for $i = 1$. By exploiting the continuity of the objective function, since (3.50) and (3.51) hold for $i = 1$, then also (3.52) is verified for $i = 1$. If $s(j) - 1 \in \mathcal{K}_2$, from the instruction of the algorithm we obtain

$$f(x^{s(j)}) = f(\tilde{x}^{s(j)-1} + \alpha^{s(j)-1}d^{s(j)-1}) \leq f_R^{q(s(j)-1)} + \gamma\alpha^{s(j)-1}g(\tilde{x}^{s(j)-1})^T d^{s(j)-1}$$

and then

$$f(x^{s(j)}) - f_R^{q(s(j)-1)} \leq \gamma\alpha^{s(j)-1}g(\tilde{x}^{s(j)-1})^T d^{s(j)-1}.$$

From (3.53), point (i) of Lemma 1, and recalling (3.25)–(3.27), we have that

$$\lim_{j\to\infty} \alpha^{s(j)-1}\|d^{s(j)-1}\| = \lim_{j\to\infty} \|x^{s(j)} - \tilde{x}^{s(j)-1}\| = 0$$

for every subsequence such that $s(j) - 1 \in \mathcal{K}_2$. Therefore, (3.50) holds for $i = 1$. Recalling that $f(x^{s(j)}) = f(\tilde{x}^{s(j)-1}+\alpha^{s(j)-1}d^{s(j)-1})$, and since (3.49) and (3.50) hold for $i = 1$, from the continuity of the objective function it follows that also (3.52) holds for $i = 1$.
Now we assume that (3.49)–(3.52) hold for a given fixed $i \geq 1$, and show that these relations must hold for $i + 1$ as well. If $s(j) - i \in \mathcal{K}_4$, by using (3.48) it is straightforward to verify that (3.49) is verified replacing i with $i + 1$. Taking into account (3.52), this implies that

$$\lim_{j\to\infty} f(x^{s(j)-(i+1)+1}) = \lim_{j\to\infty} f(x^{s(j)-i}) = \lim_{j\to\infty} f(\tilde{x}^{s(j)-i}) = \bar{f}_R$$

and then (3.51) holds for $i + 1$. If $s(j) - i \in \mathcal{K}_5$, from the instructions of the algorithm, and taking into account Proposition 10, we get

$$f(\tilde{x}^{s(j)-i}) \leq f(x^{s(j)-i}) - \frac{1}{2\epsilon}\|x^{s(j)-i} - \tilde{x}^{s(j)-i}\|^2 < f_R^{q(s(j)-i)-1}.$$

Exploiting (3.52) and point (i) of Lemma 1, we have that

$$\lim_{j\to\infty} f(\tilde{x}^{s(j)-i}) = \lim_{j\to\infty} f(x^{s(j)-i}) = \bar{f}_R, \tag{3.54}$$

which proves (3.51) for $i+1$. From the above relation, and by exploiting Proposition 10 again, we can also write

$$\lim_{j\to\infty} \left(f(\tilde{x}^{s(j)-i}) - f(x^{s(j)} - i)\right) \leq \lim_{j\to\infty} -\frac{1}{2\epsilon}\|x^{s(j)-i} - \tilde{x}^{s(j)-i}\|^2.$$

and then (3.49) holds for $i+1$.
If $s(j) - i - 1 \in \mathcal{K}_1 \cup \mathcal{K}_3$, from (3.46) and (3.47) we obtain that (3.50) holds for $i = i+1$. Since $x^{s(j)-i} = \tilde{x}^{s(j)-i-1} + \alpha^{s(j)-i-1} d^{s(j)-i-1}$, exploiting (3.46), (3.47), (3.54) and the continuity of the objective function, we obtain that (3.52) holds replacing i with $i+1$.
If $s(j) - i - 1 \in \mathcal{K}_2$, from the instruction of the algorithm we obtain

$$\begin{aligned} f(x^{s(j)-i}) &= f(\tilde{x}^{s(j)-i-1} + \alpha^{s(j)-i-1} d^{s(j)-i-1}) \\ &\leq f_R^{q(s(j)-i-1)} + \gamma\alpha^{s(j)-i-1} g(\tilde{x}^{s(j)-i-1})^T d^{s(j)-i-1} \end{aligned}$$

and then

$$f(x^{s(j)-i}) - f_R^{q(s(j)-i-1)} \leq \gamma\alpha^{s(j)-i-1} g(\tilde{x}^{s(j)-i-1})^T d^{s(j)-i-1}.$$

From (3.54), point (i) of Lemma 1, and recalling (3.25)–(3.27), we have that

$$\lim_{j\to\infty} \alpha^{s(j)-i-1}\|d^{s(j)-i-1}\| = \lim_{j\to\infty} \|x^{s(j)-i} - \tilde{x}^{s(j)-i-1}\| = 0$$

for every subsequence such that $s(j) - 1 \in \mathcal{K}_2$. Therefore, (3.50) holds for $i+1$.
Recalling that $f(x^{s(j)-i}) = f(\tilde{x}^{s(j)-i-1} + \alpha^{s(j)-i-1} d^{s(j)-i-1})$, and since (3.49)–(3.50) hold replacing i with $i+1$, exploiting the continuity of the objective function we have

$$\lim_{j\to\infty} f(\tilde{x}^{s(j)-i-1} + \alpha^{s(j)-i-1} d^{s(j)-i-1}) = \lim_{j\to\infty} f(x^{s(j)-i}) = \lim_{j\to\infty} f(\tilde{x}^{s(j)-i}).$$

Therefore, if (3.52) holds at a generic $i \geq 1$, it must hold for $i+1$ as well. This completes the induction.
Now, for any iteration index $k > 0$ we can write

$$\begin{aligned} \tilde{x}^{h^k} &= x^k + \sum_{i=0}^{h^k-k} (\tilde{x}^{h^k-i} - x^{h^k-i}) + \sum_{i=1}^{h^k-k} \alpha^{h^k-i} d^{h^k-i}, \\ \tilde{x}^{h^k} &= \tilde{x}^k + \sum_{i=0}^{h^k-k-1} (\tilde{x}^{h^k-i} - x^{h^k-i}) + \sum_{i=1}^{h^k-k} \alpha^{h^k-i} d^{h^k-i}. \end{aligned}$$

From (3.49) and (3.50) we obtain

$$\lim_{k\to\infty} \|x^k - \tilde{x}^{h^k}\| = \lim_{k\to\infty} \|\tilde{x}^k - \tilde{x}^{h^k}\| = 0.$$

By exploiting the continuity of the objective function, and taking into account point (i) of Lemma 1, the above relation implies that

$$\lim_{k\to\infty} f(x^k) = \lim_{k\to\infty} f(\tilde{x}^k) = \lim_{k\to\infty} f(\tilde{x}^{h^k}) = \lim_{k\to\infty} f_R^j = \bar{f}_R,$$

which proves (3.43).
To prove (3.44), if $k \in \mathcal{K}_1 \cup \mathcal{K}_3$, then from (3.46) and (3.47) we obtain

$$\lim_{k\to\infty,\, k\in\mathcal{K}_1\cup\mathcal{K}_3} \|x^{k+1} - \tilde{x}^k\| = \lim_{k\to\infty,\, k\in\mathcal{K}_1\cup\mathcal{K}_3} \alpha^k \|d^k\| = 0. \tag{3.55}$$

If $k \in \mathcal{K}_2$, from the instruction of the algorithm we get

$$f(x^{k+1}) \le f(\tilde{x}^{q(k)}) + \gamma\alpha^k g(\tilde{x}^k)^T d^k$$

and then, recalling conditions (3.25)–(3.27) and (3.43), we can write

$$\lim_{k\to\infty,\, k\in\mathcal{K}_2} \|x^{k+1} - \tilde{x}^k\| = \lim_{k\to\infty,\, k\in\mathcal{K}_2} \alpha^k \|d^k\| = 0. \tag{3.56}$$

From (3.55) and (3.56) it follows that (3.44) holds.
To prove (3.45), if $k \in \mathcal{K}_4$, then from (3.48) we obtain

$$\lim_{k\to\infty,\, k\in\mathcal{K}_4} \|\tilde{x}^k - x^k\| = 0. \tag{3.57}$$

If $k \in \mathcal{K}_5$, from the instruction of the algorithm, and recalling Proposition 10, we get

$$f(\tilde{x}^k) \le f(x^k) - \frac{1}{2\epsilon}\|x^k - \tilde{x}^k\|^2 < f_R^{q(k)-1}.$$

From (3.43) and point (i) of Lemma 1, we have that

$$\lim_{k\to\infty,\, k\in\mathcal{K}_5} f(\tilde{x}^k) = \lim_{k\to\infty,\, k\in\mathcal{K}_5} f(x^k) = \bar{f}_R.$$

By exploiting Proposition 10 again, we can write

$$\lim_{k\to\infty,\, k\in\mathcal{K}_5} \big(f(\tilde{x}^k) - f(x^k)\big) \le \lim_{k\to\infty,\, k\in\mathcal{K}_5} -\frac{1}{2\epsilon}\|x^k - \tilde{x}^k\|^2 = 0$$

and then

$$\lim_{k\to\infty,\, k\in\mathcal{K}_5} \|\tilde{x}^k - x^k\| = 0. \tag{3.58}$$

From (3.57) and (3.58), it follows that (3.45) holds. □

The following theorem extends a known result from unconstrained optimization, guaranteeing that the sequence of the directional derivatives along the search direction converges to zero.

Theorem 11. *Let Assumption 1 hold. Assume that* ASA-BCP *does not terminate in a finite number of iterations, and let* $\{x^k\}$, $\{\tilde{x}^k\}$ *and* $\{d^k\}$ *be the sequences produced by the algorithm. Then,*

$$\lim_{k\to\infty} g(\tilde{x}^k)^T d^k = 0. \tag{3.59}$$

Proof. We can identify two iteration index subsets $H, K \subseteq \{1, 2, \ldots\}$, such that:

- $N(\tilde{x}^k) \neq \emptyset$ and $g_N(\tilde{x}^k) \neq 0$, for all $k \in K$;
- $H = \{1, 2, \ldots\} \setminus K$.

By assumption, the algorithm does not terminate in a finite number of iterations, and then, at least one of the above sets is infinite. Since we are interested in the asymptotic behavior of the sequence produced by `ASA-BCP`, we assume without loss of generality that both H and K are infinite sets.
Taking into account Step 31 in Algorithm 9, it is straightforward to verify that

$$\lim_{k\to\infty,\, k\in H} g(\tilde{x}^k)^T d^k = 0.$$

Therefore, we limit our analysis to consider the subsequence $\{x^k\}_K$. Let $\bar{x}$ be any limit point of $\{x^k\}_K$. By contradiction, we assume that (3.59) does not hold. Using (3.45) of Lemma 3, since $\{x^k\}$, $\{\tilde{x}^k\}$ and $\{d^k\}$ are limited, and taking into account that $A_l(x^k)$, $A_u(x^k)$ and $N(x^k)$ are subsets of a finite set of indices, without loss of generality we redefine $\{x^k\}_K$ the subsequence such that

$$\lim_{k\to\infty,\, k\in K} x^k = \lim_{k\to\infty,\, k\in K} \tilde{x}^k = \bar{x},$$

and

$$N^k := \hat{N}, \quad A_l^k := \hat{A}_l, \quad A_u^k := \hat{A}_u, \quad \forall k \in K,$$
$$\lim_{k\to\infty,\, k\in K} d^k = \hat{d}.$$

Since we have assumed that (3.59) does not hold, the above relations, combined with (3.26) and the continuity of the gradient, imply that

$$\lim_{k\to\infty,\, k\in K} g(\tilde{x}^k)^T d^k = g(\bar{x})^T \hat{d} = -\eta < 0. \tag{3.60}$$

It follows that

$$\lim_{k\to\infty,\, k\in K} \hat{d} \neq 0$$

and then, recalling (3.44) of Lemma 3, we get

$$\lim_{k\to\infty,\, k\in K} \alpha^k = 0. \tag{3.61}$$

Consequently, from the instructions of the algorithm, there must exist a subsequence (renamed K again) such that the line search procedure at Step 28 is performed and $\alpha^k < 1$ for sufficiently large k. Namely,

$$\begin{aligned} f([\tilde{x}^k + \frac{\alpha^k}{\delta} d^k]^\sharp) &> f_R^{q(k)} + \gamma \frac{\alpha^k}{\delta} g(\tilde{x}^k)^T d^k \\ &\geq f(\tilde{x}^k) + \gamma \frac{\alpha^k}{\delta} g(\tilde{x}^k)^T d^k, \quad \forall k \geq \bar{k},\, k \in K, \end{aligned} \tag{3.62}$$

where $q(k) := \max\{j : l^j \leq k\}$. We can write the point $[\tilde{x}^k + \frac{\alpha^k}{\delta} d^k]^\sharp$ as follows:

$$[\tilde{x}^k + \frac{\alpha^k}{\delta} d^k]^\sharp = \tilde{x}^k + \frac{\alpha^k}{\delta} d^k - y^k, \tag{3.63}$$

where

$$y_i^k = \max\{0, (\tilde{x}^k + \frac{\alpha^k}{\delta} d^k)_i - u_i\} - \max\{0, l_i - (\tilde{x}^k + \frac{\alpha^k}{\delta} d^k)_i\}, \quad i = 1, \ldots, n.$$

As $\{\tilde{x}^k\}$ is a sequence of feasible points, $\{\alpha^k\}$ converges to zero and $\{d^k\}$ is limited, we get

$$\lim_{k \to \infty, k \in K} y^k = 0. \tag{3.64}$$

From (3.62) and (3.63) we can write

$$f(\tilde{x}^k + \frac{\alpha^k}{\delta} d^k - y^k) - f(\tilde{x}^k) > \gamma \frac{\alpha^k}{\delta} g(\tilde{x}^k)^T d^k, \quad \forall k \geq \bar{k},\, k \in K. \tag{3.65}$$

By the mean value theorem, we have

$$f(\tilde{x}^k + \frac{\alpha^k}{\delta} d^k - y^k) = f(\tilde{x}^k) + \frac{\alpha^k}{\delta} g(z^k)^T d^k - g(z^k)^T y^k, \tag{3.66}$$

where

$$z^k = \tilde{x}^k + \theta^k (\frac{\alpha^k}{\delta} d^k - y^k), \quad \theta^k \in (0, 1). \tag{3.67}$$

From (3.61) and (3.64), and since $\{d^k\}$ is limited, we obtain

$$\lim_{k \to \infty, k \in K} z^k = \bar{x}. \tag{3.68}$$

Substituting (3.66) into (3.65), and multiplying each term by $\dfrac{\delta}{\alpha^k}$, we get

$$g(z^k)^T d^k - \frac{\delta}{\alpha^k} g(z^k)^T y^k > \gamma g(\tilde{x}^k)^T d^k, \quad \forall k \geq \bar{k},\, k \in K. \tag{3.69}$$

From the definition of y^k, it follows that

$$y_i^k = \begin{cases} 0, & \text{if } l_i \leq \tilde{x}_i^k + \frac{\alpha^k}{\delta} d_i^k \leq u_i, \\ \tilde{x}_i^k + \frac{\alpha^k}{\delta} d_i^k - u_i > 0, & \text{if } \tilde{x}_i^k + \frac{\alpha^k}{\delta} d_i^k > u_i, \\ \tilde{x}_i^k + \frac{\alpha^k}{\delta} d_i^k - l_i < 0, & \text{if } l_i > \tilde{x}_i^k + \frac{\alpha^k}{\delta} d_i^k. \end{cases} \tag{3.70}$$

In particular, we have

$$y_i^k \begin{cases} = 0, & \text{if } d_i^k = 0, \\ \in [0, \frac{\alpha^k}{\delta} d_i^k], & \text{if } d_i^k > 0, \\ \in [-\frac{\alpha^k}{\delta} d_i^k, 0], & \text{if } d_i^k < 0. \end{cases} \tag{3.71}$$

From the above relation, it is straightforward to verify that

$$|y_i^k| \leq \frac{\alpha^k}{\delta} |d_i^k|, \quad i = 1, \ldots, n. \tag{3.72}$$

In the following, we want to majorize the left-hand side of (3.69) by showing that $\{\frac{\delta}{\alpha^k} g(z^k)^T y^k\}$ converges to a nonnegative value. To this aim, we analyze three different cases, depending on whether $\bar{x}_i$ is at the bounds or is strictly feasible:

(i) $i \in \hat{N}$ such that $l_i < \bar{x}_i < u_i$. As $\{\tilde{x}^k\}$ converges to $\bar{x}$, there exists $\tau > 0$ such that

$$l_i + \tau \leq \tilde{x}^k \leq u_i - \tau, \qquad k \in K,\ k \text{ sufficiently large.}$$

Since $\{\alpha^k\}$ converges to zero and $\{d^k\}$ is limited, it follows that $\frac{\alpha^k}{\delta}|d_i^k| < \tau$, for $k \in K$, k sufficiently large. Then,

$$l_i < \tilde{x}_i^k + \frac{\alpha^k}{\delta} d_i^k < u_i, \qquad k \in K,\ k \text{ sufficiently large,}$$

which implies, from (3.70), that

$$y_i^k = 0, \qquad k \in K,\ k \text{ sufficiently large.} \tag{3.73}$$

(ii) $i \in \hat{N}$ such that $\bar{x}_i = l_i$. First, we show that

$$g_i(\bar{x}) \leq 0, \tag{3.74}$$

$$y_i^k \leq 0, \qquad k \in K,\ k \text{ sufficiently large.} \tag{3.75}$$

To show (3.74), we assume by contradiction that $g_i(\bar{x}) > 0$. From (3.12) and recalling that $\|\tilde{x}^k - x^k\|$ converges to zero from (3.45) of Lemma 3, it follows that

$$\lim_{k\to\infty,\, k\in K} \lambda_i(\tilde{x}^k) = \lim_{k\to\infty,\, k\in K} g_i(\tilde{x}^k) = g_i(\bar{x}) > 0.$$

Then, there exist an iteration index $\hat{k}$ and a scalar $\xi > 0$ such that $\lambda_i(\tilde{x}^k) \geq \xi > 0$, for all $k \geq \hat{k}$, $k \in K$. As $\{\tilde{x}_i^k\}$ converges to l_i, there also exists $\tilde{k} \geq \hat{k}$ such that

$$\begin{aligned} l_i \leq \tilde{x}_i^k \leq l_i + \epsilon\xi \leq l_i + \epsilon\lambda_i(\tilde{x}^k), \qquad & k \in K,\ k \geq \tilde{k}, \\ g_i(\tilde{x}^k) > 0, \qquad & k \in K,\ k \geq \tilde{k}, \end{aligned}$$

which contradicts the fact that $i \in N(\tilde{x}^k)$ for k sufficiently large. To show (3.75), we observe that since $\{\tilde{x}_i^k\}$ converges to l_i, there exists $\tau \in (0, u_i - l_i]$ such that

$$l_i \leq \tilde{x}_i^k \leq u_i - \tau, \qquad k \in K,\ k \text{ sufficiently large.}$$

Moreover, since $\{\alpha^k\}$ converges to zero and $\{d^k\}$ is limited, it follows that $\frac{\alpha^k}{\delta} d_i^k \leq \tau$, for $k \in K$, k sufficiently large. Then,

$$\tilde{x}^k + \frac{\alpha^k}{\delta} d_i^k \leq u_i, \qquad k \in K,\ k \text{ sufficiently large.}$$

The above relation, combined with (3.70), proves (3.75). Now, we distinguish two subcases, depending on the sign of d_i^k:

- for every subsequence $\bar{K} \subseteq K$ such that $d_i^k \geq 0$, from (3.71) it follows that $y_i^k \geq 0$. Consequently, from (3.75) we can write

$$y_i^k = 0, \qquad k \in \bar{K},\ k \text{ sufficiently large.} \tag{3.76}$$

- for every subsequence $\bar{K} \subseteq K$ such that $d_i^k < 0$, we have two further possible situations, according to (3.74):
  (a) $g_i(\bar{x}) < 0$. As $\{z^k\}$ converges to $\bar{x}$, then $g_i(z^k) \leq 0$ for $k \in \bar{K}$, k sufficiently large. From (3.75), we obtain

$$\frac{\delta}{\alpha^k} g_i(z^k) y_i^k \geq 0, \qquad k \in \bar{K},\ k \text{ sufficiently large.} \tag{3.77}$$

  (b) $g_i(\bar{x}) = 0$. From (3.72), we get

$$\frac{\delta}{\alpha^k} |g_i(z^k) y_i^k| \leq \frac{\delta}{\alpha^k} |g_i(z^k)||y_i^k| \leq |g_i(z^k)||d_i^k|.$$

  Since $\{d^k\}$ is limited, $\{z^k\}$ converges to $\bar{x}$, and $g_i(\bar{x}) = 0$, from the continuity of the gradient we get

$$\lim_{k \to \infty,\, k \in \bar{K}} \frac{\delta}{\alpha^k} g_i(z^k) d_i^k = 0. \tag{3.78}$$

(iii) $i \in \hat{N}$ such that $\bar{x}_i = u_i$. Reasoning as in the previous case, we obtain

$$\lim_{k \to \infty,\, k \in K} \frac{\delta}{\alpha^k} g_i(z^k) y_i^k \geq 0. \tag{3.79}$$

Finally, from (3.73), (3.76), (3.77), (3.78) and (3.79), we have

$$\lim_{k \to \infty,\, k \in K} \frac{\delta}{\alpha^k} g(z^k)^T y^k \geq 0, \tag{3.80}$$

and, from (3.60), (3.68), (3.69), (3.79) and (3.80), we obtain

$$\begin{aligned} -\eta &= \lim_{\substack{k \to \infty \\ k \in K}} g(\tilde{x}^k)^T d^k = \lim_{\substack{k \to \infty \\ k \in K}} g(z^k)^T d^k \geq \lim_{\substack{k \to \infty \\ k \in K}} g(z^k)^T d^k - \lim_{\substack{k \to \infty \\ k \in K}} \frac{\delta}{\alpha^k} g(z^k)^T y^k \\ &\geq \lim_{\substack{k \to \infty \\ k \in K}} \gamma g(\tilde{x}^k)^T d^k = -\gamma\eta. \end{aligned}$$

This contradicts the fact that we set $\gamma < 1$ in ASA-BCP. □

Finally, we state the main theoretical result ensuring the global convergence of ASA-BCP.

Theorem 12. *Let Assumption 1 hold. Then, ASA-BCP either produces a stationary point for problem* (3.1) *in a finite number of iterations, or produces an infinite sequence* $\{x^k\}$ *and every limit point* x^* *of the sequence is a stationary point for problem* (3.1).

Proof. Let x^* be any limit point of the sequence $\{x^k\}$, and let $\{x^k\}_K$ be the subsequence converging to x^*. From (3.45) of Lemma 3 we can write

$$\lim_{k \to \infty,\, k \in K} \tilde{x}^k = x^*, \tag{3.81}$$

and, thanks to the fact that $A_l(x^k)$, $A_u(x^k)$ and $N(x^k)$ are subsets of a finite set of indices, we can define a further subsequence $\hat{K} \subseteq K$ such that

$$N^k = \hat{N}, \quad A_l^k = \hat{A}_l, \quad A_u^k = \hat{A}_u,$$

for all $k \in \hat{K}$. Recalling Proposition 7, we define the following function that measures the violation of the optimality conditions for feasible points:

$$\phi(x_i) = \min \left\{ \max\{l_i - x_i, -g_i(x)\}^2, \max\{x_i - u_i, g_i(x)\}^2 \right\}.$$

By contradiction, we assume that x^* is a non-stationary point for problem (3.1). Then, there exists an index i such that $\phi(x_i^*) > 0$. From (3.81) and the continuity of ϕ, there exists an index $\tilde{k}$ such that

$$\phi(\tilde{x}_i^k) \geq \Delta > 0, \quad \forall k \geq \tilde{k}. \tag{3.82}$$

Now, we consider three cases:

(i) $i \in \hat{A}_l$. Then, $\tilde{x}_i^k = l_i$. From (3.12) and (3.9), we get $g_i(x^k) > 0$, $\forall k \in \hat{K}$. By continuity of the gradient, and since both $\{\tilde{x}^k\}_{\hat{K}}$ and $\{x^k\}_{\hat{K}}$ converge to x^*, we obtain

$$g_i(\tilde{x}^k) \geq -\frac{\Delta}{2},$$

for $k \in \hat{K}$, k sufficiently large. Then, we have $\phi(\tilde{x}_i^k) \leq \frac{\Delta^2}{4} < \Delta$ for $k \in \hat{K}$, k sufficiently large. This contradicts (3.82).

(ii) $i \in \hat{A}_u$. Then, $\tilde{x}_i^k = u_i$. The proof of this case is a verbatim repetition of the previous case.

(iii) $i \in \hat{N}$. As $\phi(x_i^*) > 0$, then $g_i(x^*) \neq 0$. From Theorem 11, we have

$$\lim_{k \to \infty,\, k \in \hat{K}} g(\tilde{x}^k)^T d^k = 0.$$

From (3.26), it follows that

$$\lim_{k \to \infty,\, k \in \hat{K}} \|g_{\hat{N}}(\tilde{x}^k)\| = \|g_{\hat{N}}(x^*)\| = 0,$$

leading to a contradiction.

□

3.3.2 Analysis of convergence rate

In this subsection we show that superlinear convergence of the method can be proved under standard additional assumptions. First, we need to state the following lemma.

Lemma 4. *Let Assumption 1 hold and assume that $\{x^k\}$ is an infinite sequence generated by `ASA-BCP`. Then, there exists an iteration index $\bar{k}$ such that $N(x^k) \neq \emptyset$ for all $k \geq \bar{k}$.*

Proof. By contradiction, we assume that there exists an infinite index subset $\bar{K} \subseteq \{1, 2, \ldots\}$ such that $N(x^k) = \emptyset$ for all $k \in \bar{K}$. Let x^* be a limit point of $\{x\}_{\bar{K}}$, that is,

$$\lim_{k\to\infty,\, k\in K} x^k = x^*,$$

where $K \subseteq \bar{K}$. Theorem 12 ensures that x^* is a stationary point. From (3.45) of Lemma 3 we can write

$$\lim_{k\to\infty,\, k\in K} x^k = \lim_{k\to\infty,\, k\in K} \tilde{x}^k = x^*.$$

Moreover, from Proposition 6, there exists an index $\hat{k}$ such that

$$\{i\colon x_i^* = l_i, \lambda_i^* > 0\} \subseteq A_l(x^k) \subseteq \{i\colon x_i^* = l_i\}, \quad \forall\, k \geq \hat{k}, \quad k \in K, \tag{3.83}$$

$$\{i\colon x_i^* = u_i, \mu_i^* > 0\} \subseteq A_u(x^k) \subseteq \{i\colon x_i^* = u_i\}, \quad \forall\, k \geq \hat{k}, \quad k \in K. \tag{3.84}$$

Let $\tilde{k}$ be the smallest integer such that $\tilde{k} \geq \hat{k}$ and $\tilde{k} \in K$. From (3.83) and (3.84) we can write

$$\tilde{x}_i^{\tilde{k}} = l_i = x_i^*, \quad \text{if } i \in A_l(x^{\tilde{k}}),$$
$$\tilde{x}_i^{\tilde{k}} = u_i = x_i^*, \quad \text{if } i \in A_u(x^{\tilde{k}}).$$

Since $N(x^k)$ is empty for all $k \in K$, we also have

$$A_l(x^k) \cup A_u(x^k) = \{1, \ldots, n\}, \quad \forall\, k \in K.$$

Consequently, $\tilde{x}^{\tilde{k}} = x^*$, contradicting the hypothesis that the sequence $\{x^k\}$ is infinite. □

Now, we are ready to establish the superlinear rate of convergence of ASA-BCP, as formalized in the following theorem.

Theorem 13. *Assume that $\{x^k\}$ is a sequence generated by ASA-BCP converging to a point x^* satisfying the strict complementarity condition and such that*

$$H_{N^*N^*}(x^*) \succ 0,$$

where $N^* = \{i\colon l_i < x_i^* < u_i\}$. Assume that the sequence $\{d^k\}$ of directions satisfies the following condition:

$$\lim_{k\to\infty} \frac{\|H_{\tilde{N}^k\tilde{N}^k}(\tilde{x}^k) d^k_{\tilde{N}^k} + g_{\tilde{N}^k}(\tilde{x}^k)\|}{\|g_{\tilde{N}^k}(\tilde{x}^k)\|} = 0. \tag{3.85}$$

Then, the sequence $\{x^k\}$ converges to x^* superlinearly.

Proof. From Proposition 6, exploiting the fact the sequence $\{x^k\}$ converges to x^* and that strict complementarity holds, we have that for sufficiently large k,

$$N(x^k) = N(\tilde{x}^k) = N^*,$$
$$A_l(x^k) = A_l(\tilde{x}^k) = \{i\colon x_i^* = l_i\},$$
$$A_u(x^k) = A_u(\tilde{x}^k) = \{i\colon x_i^* = u_i\}.$$

From the instructions of the algorithm, it follows that $\tilde{x}^k = x^k$ for sufficiently large k, and then, the minimization is restricted on $N(\tilde{x}^k)$. From Lemma 4, we have that $N(\tilde{x}^k) = N(x^k) = N^* \neq \emptyset$ for sufficiently large k. Furthermore, from (3.85) we have that $d^k_{N(\tilde{x}^k)}$ is a Newton-truncated direction, and then, the assertion follows from standard results on unconstrained minimization. □

3.4 Numerical experience

In this section, we describe the details of our computational experience.

In Subsection 3.4.1, we describe the implementation details related to our algorithmic scheme. In Subsection 3.4.2, we compare `ASA-BCP` with the following codes:

- `NMBC` [16] (in particular, we considered the version named $\texttt{NMBC}_2$ in [16]);
- `ALGENCAN` [6]: an active-set method using spectral projected gradient steps for leaving faces, downloaded from the TANGO web page (`http://www.ime.usp.br/~egbirgin/tango`);
- `LANCELOT B` [35]: a Newton method based on a trust-region strategy, downloaded from the GALAHAD web page (`http://www.galahad.rl.ac.uk`).

All computations have been run on an Intel Xeon(R), CPU E5-1650 v2 3.50 GHz. The test set consisted of 140 bound-constrained problems from the CUTEst collection [36], with dimension up to 10^5. The stopping condition for all codes was

$$\|x - [x - g(x)]^\sharp\|_\infty < 10^{-5},$$

where $\|\cdot\|_\infty$ denotes the sup-norm of a vector.

In order to compare the performances of the algorithms, we make use of the performance profiles proposed in [23].

Following the analysis suggested in [5], we preliminarily checked whether the codes find different stationary points: the comparison is thus restricted to problems for which all codes find the same stationary point (with a tolerance of 10^{-3}). Furthermore, we do not consider in the analysis those problems for which all methods find a stationary point in less than 1 second.

In `ASA-BCP`, we set the algorithm parameters to the following values: $Z = 20$ and $M = 99$ (so that the last 100 objective function values are included in the computation of the reference value).

In running the other methods considered in the comparisons, default values were used for all parameters (but those related to the stopping condition). More specifically:

- In `NMBC`, a non-monotone strategy was employed (it is similar to the one described for `ASA-BCP`), where the parameters Z and M were equal to 20 and 100, respectively. Moreover, the parameter ϵ used in the active-set estimate was equal to 10^{-4}.

- In LANCELOT B, a band preconditioner was employed for the conjugate gradient method, with a semi-bandwidth equal to 5. Moreover, a non-monotone strategy was used with a history-length equal to 1.
- In ALGENCAN, the truncated-Newton method was used as inner solver method, the scaling feature was disabled and the parameter η was equal to 0.1 (a face of the feasible set is abandoned by the algorithm when the norm of the internal components of the continuous projected gradient is smaller than η times the norm of the continuous projected gradient).

C++ and Fortran 90 implementations (with CUTEst interface) of ASA-BCP can be found at the following web page: https://sites.google.com/a/dis.uniroma1.it/asa-bcp. The details related to the experiments are reported in Appendix B.

3.4.1 Implementation issues

In this subsection, we describe how we set the ϵ parameter appearing in the definition of the active-set estimates (3.9)–(3.10), and how we compute the search direction for the minimization with respect to the estimated non-active variables.

3.4.1.1 Updating of the ϵ parameter

A feature that characterizes ASA-BCP (and differentiates it from NMBC) is the use of the active-set estimate: once we have an ϵ satisfying Assumption 1, a decrease in the objective function is guaranteed by fixing the estimated active variables to the values at the bounds (as it is shown in Proposition 10).

In general, such an ϵ cannot be a priori computed. This is why in our implementation we use the following updating rule. Starting from the value $\epsilon = \min\{10^{-6}, \|x^0 - [x^0 - g(x^0)]^\sharp\|^{-3}\}$, at every iteration k we compute A_l^k, A_u^k and N^k. Then, we get the point

$$\tilde{x}^k_{A_l^k} = l_{A_l^k} \quad \tilde{x}^k_{A_u^k} = u_{A_u^k} \quad \tilde{x}^k_{N^k} = x^k_{N^k}.$$

If $\tilde{x}^k$ satisfies

$$f(\tilde{x}^k) - f(x^k) \leq -\frac{1}{2\epsilon}\|\tilde{x}^k - x^k\|^2,$$

then we accept $\tilde{x}^k$ and we do not change the value of ϵ. Otherwise, we do not accept $\tilde{x}^k$, we reduce ϵ and we estimate the active variables again, repeating this procedure until the above relation is satisfied.

3.4.1.2 Calculation of the gradient-related direction

At every iteration k, in order to compute the search direction with respect to $\tilde{N}^k$, ASA-BCP approximately solves the Newton equation

$$H_{\tilde{N}^k\tilde{N}^k}(\tilde{x}^k)d^k_{\tilde{N}^k} + g_{\tilde{N}^k}(\tilde{x}^k) = 0$$

by using the conjugate gradient strategy considered in [18] (and briefly described in Subsection 2.4.4). In particular, let $m = |\tilde{N}^k|$, the scheme produces a finite

sequence of conjugate directions $p_0, p_1, \ldots, p_i \in \mathbb{R}^m$, $i < m$, and the approximated Newton direction $d^k_{\tilde{N}^k}$ is computed as $d^k_{\tilde{N}^k} = d_{i+1}$, where

$$d_{i+1} = -\sum_{j=0}^{i} \frac{g_{\tilde{N}^k}(\tilde{x}^k)^T p_j}{p_j^T H_{\tilde{N}^k\tilde{N}^k}(\tilde{x}^k) p_j} p_j.$$

In our implementation of `ASA-BCP`, we employed the following stopping criterion for the conjugate gradient method:

$$\left| \frac{[q(d_{i+1}) - q(d_i)] - \frac{3}{2}[g_{\tilde{N}^k}(\tilde{x}^k)^T d_{i+1} - g_{\tilde{N}^k}(\tilde{x}^k)^T d_i]}{q(d_{i+1}) - \frac{3}{2} g_{\tilde{N}^k}(\tilde{x}^k)^T d_{i+1}} \right| (i+1) \leq \gamma_k,$$

where

$$q(d) = \frac{1}{2} d^T H_{\tilde{N}^k\tilde{N}^k}(\tilde{x}^k) d + g_{\tilde{N}^k}(\tilde{x}^k)^T d,$$

and γ_k is taken as

$$\begin{cases} \eta_k, & \text{if } \|d_{i+1}\| \geq \|g_{\tilde{N}^k}(\tilde{x}^k)\|, \\ \eta_k \min\{1, \|d_{i+1}\|\}, & \text{otherwise.} \end{cases}$$

In `ASA-BCP`, we set $\eta_k = \min\{1,\ 0.1 + e^{-0.001\,k}\}$. This represents a further difference from `NMBC`, where η_k was chosen constant and equal to 0.1.

3.4.2 Comparison on CUTEst problems

In this subsection, we first compare `ASA-BCP` with the `NMBC` algorithm presented in [16]. Then, we report the comparison of `ASA-BCP` with other two solvers for bound-constrained problems, namely `ALGENCAN` [6] and `LANCELOT B` [35]. All the codes are implemented in Fortran 90.

Recalling how we selected the relevant test problems, the analysis was restricted to 43 problems for the comparison between `ASA-BCP` and `NMBC` and to 62 problems for the comparison between `ASA-BCP`, `ALGENCAN` and `LANCELOT B`.

In particular, in the comparison between `ASA-BCP` and `NMBC`, 96 problems were discarded because they were solved in less than 1 second by both algorithms. A further problem (namely *SCOND1LS* with 5002 variables) was removed because `ASA-BCP` and `NMBC` found two different stationary points (`NMBC` found the worst one).

In the comparison between `ASA-BCP`, `ALGENCAN` and `LANCELOT B`, 75 problems were discarded because they were solved in less than 1 second by all the considered algorithms. Other 3 problems were removed as the methods stopped at different stationary points. Namely, *NCVXBQP3* with 10^5 variables, *POWELLBC* with 10^3 variables and *SCOND1LS* with 5002 variables were discarded in our comparison. The worst stationary points were found by `ASA-BCP`, `LANCELOT B` and `ASA-BCP`, respectively.

In Figure 3.1, we report the performance profiles of `ASA-BCP` and `NMBC`. These profiles show that `ASA-BCP` outperforms `NMBC` in terms of CPU time, number of objective function evaluations and number of conjugate gradient iterations. This

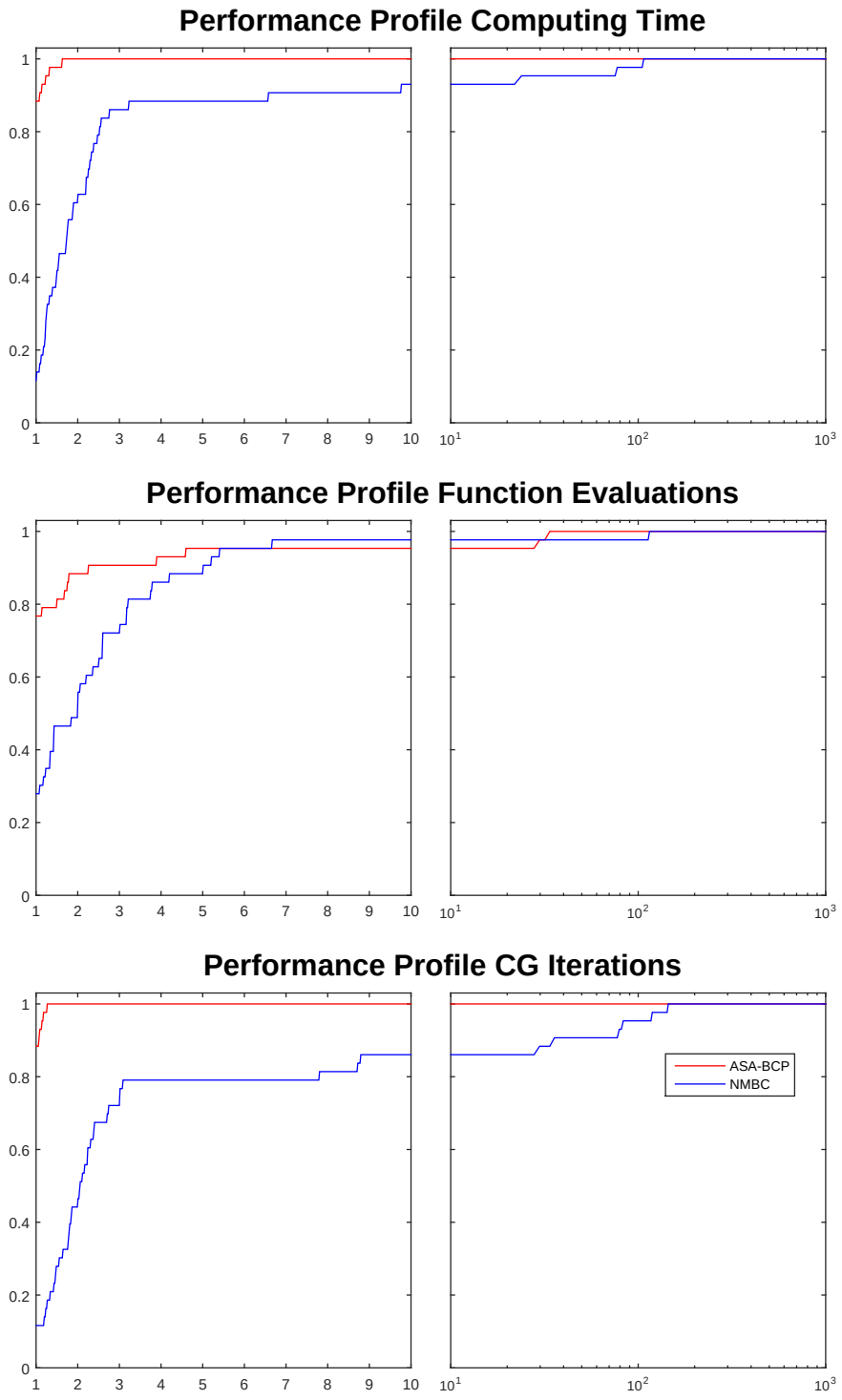

Figure 3.1. Comparison between `ASA-BCP` and `NMBC`: performance profiles on CPU time, number of objective function evaluations and number of conjugate gradient iterations. The x axis is in linear scale in the left panel and in logarithmic scale in the right panel.

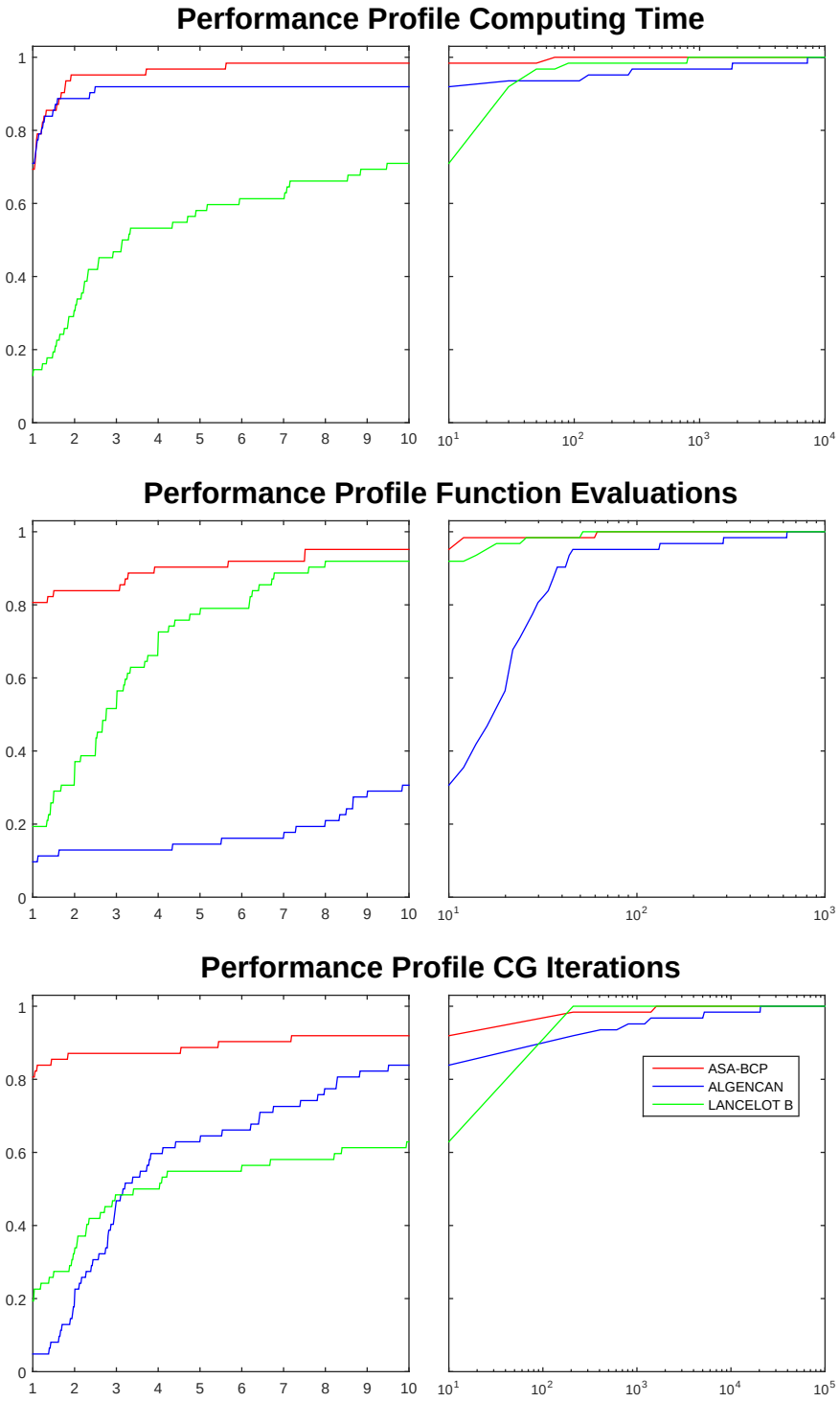

Figure 3.2. Comparison among `ASA-BCP`, `ALGENCAN` and `LANCELOT B`: performance profiles on CPU time, number of objective function evaluations and number of conjugate gradient iterations. The x axis is in linear scale in the left panel and in logarithmic scale in the right panel.

confirms the effectiveness of our two-stage approach when compared to the NMBC algorithm.

These results seem to confirm that on the one hand, computing the search direction only in the subspace of the non-active variables guarantees some savings in terms of CPU time, and on the other hand, getting rid of the Barzilai–Borwein step (used in NMBC) avoids the generation of badly scaled search directions.

In Figure 3.2, we show the performance profiles of ASA-BCP, ALGENCAN and LANCELOT B. By taking a look at the performance profiles related to CPU time, we can easily see that ASA-BCP and ALGENCAN are comparable in terms of efficiency and are both better than LANCELOT B. As regards robustness, we can see that ASA-BCP outperforms both ALGENCAN and LANCELOT B. More specifically, when τ is equal to 2, ASA-BCP solves about 95% of the problems, while ALGENCAN and LANCELOT B respectively, solve about 90% and 30% of the problems. Furthermore, ASA-BCP is able to solve all the problems when τ is about 70, while ALGENCAN and LANCELOT B get to solve all the problems for significantly larger values of τ.

For what concerns the number of objective function evaluations, ASA-BCP is the best in terms of efficiency and is competitive with LANCELOT B in terms of robustness. In particular, when τ is equal to 2, ASA-BCP solves about 85% of the problems, while ALGENCAN and LANCELOT B respectively, solve about 15% and 30% of the problems. Moreover, ASA-BCP and LANCELOT B solve all the problem when τ is about 60 and 50, respectively, while ALGENCAN gets to solve all the problems when τ is about 600.

Finally, as regards the number of conjugate gradient iterations, ASA-BCP outperforms the other two codes in terms of efficiency, while LANCELOT B is the best in terms of robustness. More in detail, when τ is equal to 2, ASA-BCP solves about 85% of the problems, while ALGENCAN and LANCELOT B respectively, solve about 20% and 35% of the problems. LANCELOT B is able to solve all the problems when τ is about 200, while ASA-BCP and ALGENCAN need larger values of τ.

3.5 Conclusions

In this chapter, a two-stage active-set algorithm for box-constrained non-linear programming problems has been devised. In the first stage, we get a significant reduction in the objective function simply by setting to the bounds the estimated active variables. In the second stage, we employ a truncated-Newton direction computed in the subspace of the estimated non-active variables. These two stages are inserted in a non-monotone framework and the convergence of the resulting algorithm ASA-BCP is proved. Experimental results have shown that our implementation of ASA-BCP is competitive with other widely used codes for bound-constrained minimization problems.

Chapter 4

An active-set method for minimization over the simplex and the ℓ_1-ball

In this chapter, we are concerned with minimization problems over the unit simplex. As to be shown, this problem is pretty general, since every minimization problem over a polytope can be reformulated as a minimization problem over the unit simplex. Here, we propose an active-set estimate that enables us to define an algorithmic framework where the variables estimated active and those estimated non-active are updated separately at each iteration, extending the approach of the previous chapter. In particular, we consider different variants of the Frank-Wolfe direction to be combined with the proposed active-set strategy. Then, we focus on the problem of minimizing a function over the ℓ_1-ball, showing how our algorithmic framework can be efficiently adapted to this problem. Preliminary numerical results are provided.

The notation used in this chapter is that reported in Appendix A. We recall that we indicate with e_i the i-th unit vector (all components are 0 except for the i-th component, which is 1) and with $e \in \mathbb{R}^n$ the n-dimensional vector with all components equal to 1. Moreover, given a vector $v \in \mathbb{R}^n$ and an index set $I \subseteq \{1, \ldots, n\}$, we denote by v_I the subvector with components v_i, $i \in I$.

The rest of the chapter is organized as follows. In Section 4.1, we introduce the minimization problem over the unit simplex. In Section 4.2, we describe our active-set estimate. In Section 4.3, we present our algorithmic framework. In Section 4.4, we carry out the convergence analysis of the algorithm for different choices of the search direction. In Section 4.5, we show how our algorithm can be easily extended to minimization problems over the ℓ_1-ball. In Section 4.6, we provide some preliminary numerical results. Finally, in Section 4.7, we draw some conclusions.

4.1 Introduction

We focus on the following minimization problem:

$$\begin{aligned} &\min_x f(x) \\ &\sum_{i=1}^{n} x_i = 1 \\ &x_i \geq 0, \quad i = 1, \dots, n, \end{aligned} \tag{4.1}$$

where $f \in C^1(\mathbb{R}^n)$. Throughout the chapter, we also make the assumption that $\nabla f(x)$ is Lipschitz continuous on the feasible set.

The problem of optimizing a continuously differentiable function over the simplex arises from several applications: e.g., finding maximum stable sets in graphs, portfolio optimization, game theory and population dynamics problems (see [15] and references therein).

We also note that every minimization problem whose feasible region is a polytope can be rewritten as a minimization problem over the unit simplex. In fact, let us consider a problem of the following form:

$$\begin{aligned} &\min_x h(x) \\ &x \in D, \end{aligned}$$

where $h\colon R^n \to R$ and $D = \operatorname{conv}\{v_1, \dots, v_m\}$, with $v_i \in \mathbb{R}^n$, $i = 1, \dots, m$. Every feasible point x can be expressed as a convex combination of $v_1, \dots, v_m$, that is, there exist $\alpha_1, \dots, \alpha_m$ such that

$$\begin{aligned} &x = \sum_{i=1}^{m} \alpha_i v_i, \\ &\sum_{i=1}^{m} \alpha_i = 1, \\ &\alpha_i \geq 0, \quad i = 1, \dots, m. \end{aligned}$$

By introducing the matrix $V = \begin{bmatrix} v_1 & \dots & v_m \end{bmatrix} \in \mathbb{R}^{n \times m}$, we can then rewrite the problem as

$$\begin{aligned} &\min_\alpha f(\alpha) \\ &\sum_{i=1}^{m} \alpha_i = 1 \\ &\alpha_i \geq 0, \quad i = 1, \dots, m, \end{aligned}$$

where

$$f(\alpha) = h(V\alpha).$$

An interesting example into this context is given by the minimization of a function over the ℓ_1-ball:

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} h(x) \\ &\|x\|_1 \leq \tau, \end{aligned}$$

with $\tau > 0$. Since the feasible set of this problem can be expressed as

$$\text{conv}\{\pm\tau e_1, \ldots, \pm\tau e_n\},$$

by performing a suitable variable transformation we can obtain an equivalent minimization problem whose feasible set is the unit simplex in $\mathbb{R}^{2n}$. This problem will be analyzed more in detail in Section 4.5, where a specific algorithmic framework for its solution will be described.

4.2 Active-set estimate

As discussed in the Chapter 3, using an active-set strategy to solve a constrained problem can be crucial from both a computational and theoretical point of view. In this section, we present an active-set estimate for problem (4.1), pointing out its theoretical properties.

First, we provide the definition of stationary point for problem (4.1).

Definition 8. *A feasible point x^* of problem* (4.1) *is a stationary point if and only if it satisfies the following first order necessary optimality conditions:*

$$\nabla f(x^*) - \lambda^* e - \mu^* = 0, \tag{4.2}$$

$$(\mu^*)^T x^* = 0, \tag{4.3}$$

$$\mu^* \geq 0. \tag{4.4}$$

where $\lambda^* \in \mathbb{R}$ and $\mu^* \in \mathbb{R}^n$ are the KKT multipliers.

Now, we can define the active-set for a stationary point x^* as the subset of zero-components of x^*. In particular, we provide the following definition.

Definition 9. *Let $x^* \in \mathbb{R}^n$ be a stationary point of problem* (4.1)*. We define as active-set the following set:*

$$\bar{A}(x^*) = \{i \in \{1, \ldots, n\} \colon x_i^* = 0\}. \tag{4.5}$$

We further define the non-active set $\bar{N}(x^*)$ as the complementary set of $\bar{A}(x^*)$:

$$\bar{N}(x^*) = \{1, \ldots, n\} \setminus \bar{A}(x^*) = \{i \in \{1, \ldots, n\} \colon x_i^* > 0\}. \tag{4.6}$$

Here, similarly as for the box-constrained case addressed in the Chapter 3, the active-set estimate is computed by following the approach proposed in [22, 24], which requires proper approximation of the KKT multipliers. The latter are obtained by means of multiplier functions (see the definition of multiplier function in Section 3.2).

In particular, let us describe how to compute the multiplier functions $\lambda \colon \mathbb{R}^n \to \mathbb{R}$ and $\mu \colon \mathbb{R}^n \to \mathbb{R}^n$. Given a stationary point x^* of (4.1), let (λ^*, μ^*) be the KKT multipliers associated to x^*. By (4.2), we have

$$\mu^* = \nabla f(x^*) - \lambda^* e,$$

then, multiplying by x^* and taking into account complementarity condition (4.3), we get

$$0 = (\mu^*)^T x^* = (\nabla f(x^*) - \lambda^* e)^T x^*.$$

From the feasibility of x^*, we obtain the following expression for the multipliers:

$$\lambda^* = \nabla f(x^*)^T x^*, \tag{4.7}$$
$$\mu^* = \nabla f(x^*) - \lambda^* e. \tag{4.8}$$

From (4.7)–(4.8), we can introduce the following two multiplier functions:

$$\lambda(x) = \nabla f(x)^T x, \tag{4.9}$$
$$\mu_i(x) = \nabla_i f(x) - \lambda(x), \quad i = 1, \ldots, n. \tag{4.10}$$

Now, for every feasible point x, we define the active-set estimate $A(x)$ and the non-active set estimate $N(x)$ as

$$A(x) = \{i \colon x_i \leq \epsilon \mu_i(x)\} = \{i \colon x_i \leq \epsilon \nabla f(x)^T (e_i - x)\}, \tag{4.11}$$
$$N(x) = \{i \colon x_i > \epsilon \mu_i(x)\} = \{i \colon x_i > \epsilon \nabla f(x)^T (e_i - x)\}, \tag{4.12}$$

where ϵ is a positive scalar.

As for the box-constrained case, this estimate of the active-set enables us to adapt the results shown in [24] and state the following theorem.

Theorem 14. *If (x^*, λ^*, μ^*) satisfies KKT conditions for problem (4.1), then there exists a neighborhood $\mathcal{B}(x^*, \rho)$ such that, for each x in this neighborhood, we have*

$$\{i \colon x_i^* = 0,\ \mu_i(x^*) > 0\} \subseteq A(x) \subseteq \bar{A}(x^*).$$

Furthermore, if strict complementarity holds, then

$$\{i \colon x_i^* = 0,\ \mu_i(x^*) > 0\} = A(x) = \bar{A}(x^*),$$

for each $x \in \mathcal{B}(x^*, \rho)$.

4.2.1 Characterization of stationary points

In this subsection, we provide a characterization of stationary points based on the proposed active-set estimate. First, we need the following proposition to state necessary and sufficient conditions for stationarity.

Proposition 12. *Let x^* be a feasible point of problem (4.1). Then, x^* is a stationary point if and only if*

$$\min_{i=1,\ldots,n} \Big\{ \nabla_i f(x^*) \Big\} = \max_{j \in \bar{N}(x^*)} \Big\{ \nabla_j f(x^*) \Big\}. \tag{4.13}$$

Proof. First, we prove necessary. From KKT conditions (4.2)–(4.4), we can write

$$\nabla f(x^*) - \lambda^* e = \mu^* \geq 0. \tag{4.14}$$

From (4.3), we have that $\mu_j^* = 0$ for every index $j \in \bar{N}(x^*)$. It follows that

$$\nabla_j f(x^*) = \lambda^*, \quad \forall j \in \bar{N}(x^*).$$

The above relation, combined with (4.14), implies that

$$\nabla_i f(x^*) \geq \lambda^* = \max_{j \in \bar{N}(x^*)} \Big\{ \nabla_j f(x^*) \Big\}, \quad \forall i = 1, \ldots, n,$$

and then (4.13) holds.

Now, assume that (4.13) is satisfied. It follows that

$$\min_{j \in \bar{N}(x^*)} \Big\{ \nabla_j f(x^*) \Big\} \geq \max_{j \in \bar{N}(x^*)} \Big\{ \nabla_j f(x^*) \Big\},$$

implying that all $\nabla_j f(x^*)$, with $j \in \bar{N}(x^*)$, have the same value. Therefore, there exists a scalar λ^* such that

$$\nabla_j f(x^*) = \lambda^*, \quad \forall j \in \bar{N}(x^*). \tag{4.15}$$

From (4.13) and (4.15), there exist $\mu_1^*, \ldots, \mu_n^* \in \mathbb{R}$ such that

$$\begin{aligned} \nabla_i f(x^*) - \lambda^* = \mu_i^* \geq 0, \quad i = 1, \ldots, n, \\ \mu_j^* = 0, \quad j \in \bar{N}(x^*). \end{aligned}$$

Then, conditions (4.2)–(4.4) hold. □

Moreover, from the previous proposition, we can easily state the following corollary.

Corollary 1. *Let x^* be a feasible point of problem (4.1). Then, x^* is a stationary point if and only if there exists a scalar ξ such that*

$$\begin{aligned} \nabla_j f(x^*) = \xi, \quad \forall j \in \bar{N}(x^*), \\ \nabla_i f(x^*) \geq \xi, \quad \forall i \in \bar{A}(x^*). \end{aligned}$$

Proof. The proof follows from that of Proposition 12, observing that for every index $j \in \bar{N}(x^*)$, we have that $\nabla_j f(x^*) = \lambda^*$, where λ^* is the KKT multiplier appearing in (4.2)–(4.4). □

The following two propositions show how our active-set estimate can be used to characterize stationary points.

Proposition 13. *Given a feasible point $\bar{x}$ of problem (4.1), assume that*

$$\{ i \in A(\bar{x}) \colon \bar{x}_i > 0 \} = \emptyset. \tag{4.16}$$

Then, $\bar{x}$ is a stationary point of problem (4.1) if only if

$$\min_{i \in N(\bar{x})} \Big\{ \nabla_i f(\bar{x}) \Big\} = \max_{\substack{j \in N(\bar{x}) \\ \bar{x}_j > 0}} \Big\{ \nabla_j f(\bar{x}) \Big\}. \tag{4.17}$$

Proof. First, we observe that, under the hypothesis (4.16), we have that

$$\{j \in N(\bar{x})\colon \bar{x}_j > 0\} = \{j \in \{1,\dots,n\}\colon \bar{x}_j > 0\}. \tag{4.18}$$

Let us assume that (4.17) holds. Then,

$$\nabla_j f(\bar{x}) = \bar{\xi}, \quad j \in N(\bar{x}),\, \bar{x}_j > 0. \tag{4.19}$$

Recalling (4.11)–(4.12), and using (4.18) and (4.19), for every index $i \in A(\bar{x})$ we can write

$$\begin{aligned} 0 = \bar{x}_i \le \epsilon\big(\nabla_i f(\bar{x}) - \nabla f(\bar{x})^T \bar{x}\big) &= \epsilon\Big(\nabla_i f(\bar{x}) - \sum_{\bar{x}_j>0} \nabla_j f(\bar{x})\, \bar{x}_j\Big) \\ &= \epsilon\big(\nabla_i f(\bar{x}) - \bar{\xi} e^T x\big) = \epsilon\big(\nabla_i f(\bar{x}) - \bar{\xi}\,\big). \end{aligned}$$

Thus, we have

$$\nabla_i f(\bar{x}) \ge \bar{\xi}, \quad \forall i \in A(\bar{x}).$$

It follows that (4.13) holds.

Now, let us assume that $\bar{x}$ is a stationary point, that is,

$$\min_{i=1,\dots,n} \Big\{\nabla_i f(\bar{x})\Big\} = \max_{\substack{j=1,\dots,n \\ \bar{x}_j>0}} \Big\{\nabla_j f(\bar{x})\Big\}. \tag{4.20}$$

It is easy to see that the following relations hold:

$$\min_{i=1,\dots,n} \Big\{\nabla_i f(\bar{x})\Big\} \le \min_{j\in N(\bar{x})} \Big\{\nabla_j f(\bar{x})\Big\} \le \max_{\substack{j\in N(\bar{x}) \\ \bar{x}_j>0}} \Big\{\nabla_j f(\bar{x})\Big\} \le \max_{\substack{j=1,\dots,n \\ \bar{x}_j>0}} \Big\{\nabla_j f(\bar{x})\Big\}.$$

From (4.20), it follows that (4.17) holds. □

Proposition 14. *Given a feasible point $\bar{x}$ of problem* (4.1)*, assume that*

$$\min_{i\in N(\bar{x})} \Big\{\nabla_i f(\bar{x})\Big\} = \max_{\substack{j\in N(\bar{x}) \\ \bar{x}_j>0}} \Big\{\nabla_j f(\bar{x})\Big\}. \tag{4.21}$$

Then, $\bar{x}$ is a stationary point of problem (4.1) *if only if*

$$\{i \in A(\bar{x})\colon \bar{x}_i > 0\} = \emptyset. \tag{4.22}$$

Proof. First, from (4.21) we observe that

$$\nabla_j f(\bar{x}) = \bar{\xi}, \quad j \in N(\bar{x}),\, \bar{x}_j > 0. \tag{4.23}$$

Now, let us assume that (4.22) holds. Recalling (4.11)–(4.12), and using (4.23), for every index $i \in A(\bar{x})$ we can write

$$\begin{aligned} 0 = \bar{x}_i \le \epsilon\big(\nabla_i f(\bar{x}) - \nabla f(\bar{x})^T \bar{x}\big) &= \epsilon\Big(\nabla_i f(\bar{x}) - \sum_{\bar{x}_j>0} \nabla_j f(\bar{x})\, \bar{x}_j\Big) \\ &= \epsilon\big(\nabla_i f(\bar{x}) - \bar{\xi} e^T x\big) = \epsilon\big(\nabla_i f(\bar{x}) - \bar{\xi}\,\big). \end{aligned}$$

Thus, we have

$$\nabla_i f(\bar{x}) \geq \bar{\xi}, \quad \forall i \in A(\bar{x}).$$

It follows that (4.13) holds.

Now, let us assume that $\bar{x}$ is a stationary point. From Corollary 1, we have that

$$\nabla_j f(\bar{x}) = \bar{\xi}, \quad j \in N(\bar{x}).$$

By contradiction, we assume that there exists $i \in A(\bar{x})$ such that $\bar{x}_i > 0$. It follows that

$$\begin{aligned} 0 < \bar{x}_i \leq \epsilon(\nabla_i f(\bar{x}) - \nabla f(\bar{x})^T \bar{x}) &= \epsilon\Big(\nabla_i f(\bar{x}) - \sum_{\bar{x}_j > 0} \nabla_j f(\bar{x})\, \bar{x}_j\Big) \\ &= \epsilon(\nabla_i f(\bar{x}) - \bar{\xi} e^T x) = \epsilon(\nabla_i f(\bar{x}) - \bar{\xi}\,) \end{aligned}$$

and then

$$\nabla_i f(\bar{x}) > \bar{\xi},$$

thus contradicting (4.13). □

4.2.2 Descent property of the active-set

In this subsection, we show how, given a feasible point x, we can obtain a sufficient decrease in the objective function by setting the estimated active variables to zero.

Here, differently from the box-constrained case, we have a linear constraint, imposing that all variables sum up to 1. Therefore, in order to maintain feasibility, at least one non-active variable must be updated as well.

In particular, let us define the following index set:

$$J(x) = \Big\{ j \colon \; j \in \operatorname*{Argmin}_{i=1,\dots,n} \{\nabla_i f(x)\} \Big\}. \tag{4.24}$$

As to be shown, after setting the estimated active variables to zero, we can guarantee that the objective function decreases and that the new point is still feasible by suitably updating only one variable x_j, with j chosen in $N(x) \cap J(x)$.

First, we need to prove that $N(x) \cap J(x)$ is non-empty for every feasible point x. This is stated in the following proposition.

Proposition 15. *For every feasible point x of problem* (4.1)*, we have*

$$N(x) \cap J(x) \neq \emptyset.$$

Proof. We distinguish two different cases.

(1) $|J(x)| = n$. For every $j \in J(x)$, we have

$$\nabla f(x)^T x = \nabla_j f(x) e^T x = \nabla_j f(x), \tag{4.25}$$

Exploiting the feasibility of x, we can choose an index $\nu \in J(x)$ such that $x_\nu > 0$ and, recalling definition of multipliers (4.9) and equation (4.25), we can write

$$\mu_\nu(x) = \nabla_\nu f(x) - \lambda(x) = \nabla_\nu f(x) - \nabla f(x)^T x = \nabla_\nu f(x) - \nabla_\nu f(x) = 0 < x_\nu.$$

Since $x_\nu > 0$ and $\mu_\nu(x) = 0$, we have that $x_\nu > \epsilon \mu_\nu(x)$ and then $\nu \in N(x)$.

(2) $|J(x)| < n$. We consider two subcases.

- We first assume that for every h such that $\nabla_h f(x) > \nabla_j f(x)$, $j \in J(x)$, we have $x_h = 0$. It follows that

$$\sum_{j \in J(x)} x_j = 1$$

and, reasoning as in the previous case, we get that (4.25) holds for all $j \in J(x)$. Using the fact that x is a feasible solution for problem (4.1), we can choose an index $\nu \in J(x)$ such that $x_\nu > 0$ and, recalling definition of multipliers (4.9) and equation (4.25), we can write

$$\mu_\nu(x) = \nabla_\nu f(x) - \lambda(x) = \nabla_\nu f(x) - \nabla f(x)^T x = \nabla_\nu f(x) - \nabla_\nu f(x) = 0 < x_\nu.$$

Since $x_\nu > 0$ and $\mu(x) = 0$, we have that $x_\nu > \epsilon \mu_\nu(x)$ and then $\nu \in N(x)$.

- Now we consider the case when there exists h such that $\nabla_h f(x) > \nabla_j f(x)$, $j \in J(x)$, and $x_h > 0$. It follows that

$$\nabla f(x)^T x > \nabla_j f(x) e^T x = \nabla_j f(x).$$

Choosing $\nu = j$, for any $j \in J(x)$, and reasoning as before, we can write

$$\mu_\nu(x) = \nabla_\nu f(x) - \lambda(x) = \nabla_\nu f(x) - \nabla f(x)^T x < \nabla_\nu f(x) - \nabla_\nu f(x) = 0 \leq x_\nu.$$

Since $x_\nu \geq 0$ and $\mu_\nu(x) < 0$, we have that $x_\nu > \epsilon \mu_\nu(x)$ and then $\nu \in N(x)$.

□

Remark 2. *Proposition 15 implies that for every feasible point x, the set $N(x)$ is non-empty.*

Before stating the main result of this section, we need an assumption on the parameter ϵ appearing in the definition of the active-set.

Assumption 2. *Assume that the parameter ϵ appearing in the estimates* (4.11)–(4.12) *satisfies the following conditions:*

$$0 < \epsilon \leq \frac{1}{2Ln}, \tag{4.26}$$

where L is the Lipschitz constant of $\nabla f(x)$ over the unit simplex.

Now, we are ready to state the main theoretical result of this section.

Proposition 16. *Let Assumption 2 hold. Given a feasible point x of problem* (4.1)*, let $j \in N(x) \cap J(x)$ and $I = \{1, \ldots, n\} \setminus \{j\}$. Let $\hat{A}(x)$ be a set of indices such that*

$$\hat{A}(x) \subseteq A(x).$$

Let $\tilde{x}$ be the feasible point defined as follows:

$$\tilde{x}_i = \begin{cases} 0, & i \in \hat{A}(x), \\ x_i, & i \in I \setminus \hat{A}(x), \\ x_i + \displaystyle\sum_{h \in \hat{A}(x)} x_h, & i = j. \end{cases}$$

Then,

$$f(\tilde{x}) - f(x) \leq -L\|\tilde{x} - x\|^2.$$

where L is the Lipschitz constant of $\nabla f(x)$ over the unit simplex.

Proof. We first define the following set

$$\hat{A} = \hat{A}(x).$$

Using the mean value theorem, we can write:

$$f(\tilde{x}) = f(x) + \nabla f(w)^T(\tilde{x} - x),$$

where $w = x + \xi(\tilde{x} - x)$, $\xi \in (0, 1)$.
From the Lipschitz continuity of the gradient, we have that

$$\begin{aligned} f(\tilde{x}) &= f(x) + \nabla f(x)^T(\tilde{x} - x) + \left[\nabla f(w) - \nabla f(x)\right]^T(\tilde{x} - x) \\ &\leq f(x) + \nabla f(x)^T(\tilde{x} - x) + \|\nabla f(w) - \nabla f(x)\|\|\tilde{x} - x\| \\ &\leq f(x) + \nabla f(x)^T(\tilde{x} - x) + L\|\tilde{x} - x\|^2 \end{aligned}$$

and, by adding and removing $L\|\tilde{x} - x\|^2$, we get

$$f(\tilde{x}) \leq f(x) + \nabla f(x)^T(\tilde{x} - x) + 2L\|\tilde{x} - x\|^2 - L\|\tilde{x} - x\|^2. \tag{4.27}$$

In order to prove the proposition, we need to show that

$$\nabla f(x)^T(\tilde{x} - x) + 2L\|\tilde{x} - x\|^2 \leq 0. \tag{4.28}$$

From the definition of the components $\tilde{x}_i$, $i = 1, \ldots, n$, we can write

$$\tilde{x}_i - x_i = \begin{cases} -x_i, & i \in \hat{A}, \\ 0, & i \in I \setminus \hat{A}, \\ \sum\limits_{h \in \hat{A}} x_h, & i = j, \end{cases}$$

so that

$$\|\tilde{x} - x\|^2 = \sum_{i \in \hat{A}} (x_i)^2 + \left(\sum_{i \in \hat{A}} x_i\right)^2 \leq \sum_{i \in \hat{A}} (x_i)^2 + |\hat{A}| \sum_{i \in \hat{A}} (x_i)^2 = (|\hat{A}| + 1) x_{\hat{A}}^T x_{\hat{A}} \tag{4.29}$$

and

$$\nabla f(x)^T(\tilde{x} - x) = -\nabla_{\hat{A}} f(x)^T x_{\hat{A}} + \nabla_j f(x) \sum_{i \in \hat{A}} x_i = x_{\hat{A}}^T \Big(\nabla_j f(x) e_{\hat{A}} - \nabla_{\hat{A}} f(x)\Big). \tag{4.30}$$

From the definition of the index j, we have that $\nabla_i f(x) \geq \nabla_j f(x)$ for all $i \in \{1, \ldots, n\}$. Then, we can write

$$\sum_{i=1}^{n} \nabla_i f(x) x_i \geq \sum_{i=1}^{n} \nabla_j f(x) x_i = \nabla_j f(x) \sum_{i=1}^{n} x_i = \nabla_j f(x). \tag{4.31}$$

Recalling the active-set estimation, we have that

$$x_i \leq \epsilon\Big(\nabla_i f(x) - \sum_{i=1}^{n} \nabla_i f(x) x_i\Big) \leq \epsilon(\nabla_i f(x) - \nabla_j f(x)), \quad \forall i \in \hat{A}, \tag{4.32}$$

where the last inequality follows from (4.31). Using (4.29) and (4.32), we can write

$$\|\tilde{x} - x\|^2 \leq \epsilon(|A| + 1)\, x_{\hat{A}}^T\Big(\nabla_{\hat{A}} f(x) - \nabla_j f(x) e_{\hat{A}}\Big). \tag{4.33}$$

From (4.30) and (4.33), we get

$$\begin{aligned}
\nabla f(x)^T(\tilde{x} - x) + 2L\|\tilde{x} - x\|^2 &\leq x_{\hat{A}}^T[\nabla_j f(x) e_{\hat{A}} - \nabla_{\hat{A}} f(x)] + \\
&\quad + 2L(|A| + 1)\epsilon\, x_{\hat{A}}^T\Big(\nabla_{\hat{A}} f(x) - \nabla_j f(x) e_{\hat{A}}\Big) \\
&= \left[2L(|A| + 1)\epsilon - 1\right] x_{\hat{A}}^T\Big(\nabla_{\hat{A}} f(x) - \nabla_j f(x) e_{\hat{A}}\Big) \\
&\leq (2Ln\epsilon - 1)\, x_{\hat{A}}^T\Big(\nabla_{\hat{A}} f(x) - \nabla_j f(x) e_{\hat{A}}\Big),
\end{aligned}$$

where the last inequality follows from the non-negativity of $x_{\hat{A}}^T\Big(\nabla_{\hat{A}} f(x) - \nabla_j f(x) e_{\hat{A}}\Big)$ (implied by (4.33)) and from and the fact that $|A| \leq n - 1$ (implied by Proposition 15). Then, inequality (4.28) follows from the assumption we made on ϵ. □

4.3 Algorithmic framework

In this section, we describe an algorithmic framework to minimize a function over the unit simplex, called Active-Set framework for optimization over the Simplex (`AS-SIMPLEX`).

The proposed approach exploits the active-set strategy described in the previous section and it is based on performing two minimization steps at each iteration: one for updating the estimated active variables and one for updating the estimated non-active variables.

In particular, let x^k be the point produced at a generic iteration k and assume that we have computed the active and non-active set estimates $A(x^k)$, $N(x^k)$. Then,

- first, we get a reduction in the objective function by generating the feasible point $\tilde{x}^k$, obtained by setting $x_{A(x^k)}$ to zero and updating the variable x_j^k, $j \in J(x^k)$, as indicated in Proposition 16, being $J(x^k)$ the index set defined as in (4.24);
- afterwards, we compute the next iterate x^{k+1} by moving the estimated non-active variables along a suitable search direction.

The formal scheme of the proposed algorithmic framework is reported in Algorithm 10.

Remark 3. *In Algorithm 10, we allow for directions d^k such that $\nabla f(\tilde{x}^k)^T d^k = 0$. This is due to the fact that only the estimated non-active variables are updated by moving $\tilde{x}^k$ along d^k (since $d_{A^k}^k = 0$). But, at a generic iteration k, it can happen that all the estimated non-active variables satisfy stationarity conditions over N^k, and then, $d^k = 0$.*

Algorithm 10 `Active-Set algorithmic framework for minimization over the Simplex (AS-SIMPLEX)`

Choose a feasible point x^0
For $k = 0, 1, \ldots$
  If x^k is a stationary point, then STOP
  Active-Set Estimation:
  Compute $A^k := A(x^k)$ and $N^k := N(x^k)$
  Minimization procedure over A^k:
  Compute $J^k := J(x^k)$, choose $j \in N^k \cap J^k$ and define $\tilde{N}^k = N^k \setminus \{j\}$
  Set $\tilde{x}^k_{A^k} = 0\,,\ \tilde{x}^k_{\tilde{N}^k} = x^k_{\tilde{N}^k}$ and $\tilde{x}^k_j = x^k_j + \sum_{h \in A^k} x^k_h$
  Minimization procedure over N^k:
  Set $d^k_{A^k} = 0$
  Compute a feasible direction $d^k_{N^k}$ in $\tilde{x}^k$ and a maximum stepsize $\alpha^k_{\max}$
  If $\nabla f(\tilde{x}^k)^T d^k < 0$ then
    Compute a stepsize $\alpha^k \in (0, \alpha^k_{\max}]$ by means of the Armijo line search (Algorithm 11)
  Else
    Set $\alpha^k = 0$
  End if
  Set $x^{k+1} = \tilde{x}^k + \alpha^k d^k$
End for

Algorithm 11 `Armijo line search within Algorithm 10`

Choose $\delta \in (0,1)$, $\gamma \in (0, \frac{1}{2})$
Choose initial stepsize $\alpha \in (0, \alpha^k_{\max}]$
While $f(\tilde{x}^k + \alpha d^k) > f(\tilde{x}^k) + \gamma\, \alpha\, \nabla f(\tilde{x}^k)^T d^k$
  Set $\alpha = \delta\alpha$
End while

Now, we describe some possible choices of the search directions d^k to be used in Algorithm 10. In particular, we consider different Frank-Wolfe (FW) type directions. For the sake of completeness, in the next subsection we preliminarily recall the classical Frank-Wolfe method and some of its variants.

4.3.1 The Frank-Wolfe method and its variants

The *Frank-Wolfe method* [30] (also known as *conditional gradient method*) is a popular algorithm to solve constrained problems of the following form:

$$\begin{aligned} &\min f(x) \\ &x \in D, \end{aligned} \tag{4.34}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function and D is a compact convex set.

At each iteration, the Frank-Wolfe method computes a feasible search direction by minimizing a linear approximation of the objective function. Then, a stepsize is obtained by the Armijo line search. The method is reported in Algorithm 12.

Algorithm 12 `Frank-Wolfe method to solve problem` (4.34)

0 Choose $x^0 \in D$
1 For $k = 0, 1, \ldots$
2 If x^k is a stationary point, then STOP
3 Compute $y^k \in \operatorname*{Argmin}_{x \in D}\{\nabla f(x^k)^T(x - x^k)\}$ and set $d^k = y^k - x^k$
4 Set $\alpha^k_{\max} = 1$ and compute α^k by the Armijo line search (Algorithm 13)
5 Set $x^{k+1} = x^k + \alpha^k d^k$
6 Set $k = k + 1$
7 End for

Algorithm 13 `Armijo line search within the Frank-Wolfe method`

0 Given $\gamma \in (0, 1)$, $\delta \in (0, 1)$, $\alpha^k_{\max} > 0$
1 Set $\alpha = \alpha^k_{\max}$
2 While $f(x^k + \alpha d^k) > f(x^k) + \gamma\alpha\nabla f(x^k)^T d^k$
3 Set $\alpha = \delta\alpha$
4 End while
5 Set $\alpha^k = \alpha$

It can be proved that, if x^k is non-stationary, then a descent direction d^k is produced at every iteration k. Moreover, the convexity of D implies that $x^k + \alpha d^k \in D$ for all $\alpha \in [0, 1]$.

A key point in the Frank-Wolfe method is the computation of the search direction d^k, because it requires to minimize the linear function $\nabla f(x^k)^T(x - x^k)$ over the feasible set D. If D is a polyhedron, this means solving a linear problem at every iteration, which can be made efficiently. Moreover, since x^k is fixed, we can simply write $y^k \in \operatorname*{Argmin}_{x \in D}\{\nabla f(x^k)^T x\}$.

The Frank-Wolfe method converges to points satisfying first-order optimality conditions.

For what concerns the convergence rate, assuming that D is a polytope, that the objective function is strongly convex on D and that $\nabla f(x)$ is Lipschitz continuous on D, we have that the sequence $\{f(x^k)\}$ converges to the optimal value $f(x^*)$ linearly if the optimal solution x^* lies in the relative interior of the feasible set [41]. Otherwise, the convergence rate is sublinear, due to the zigzagging effect. To guarantee a linear convergence rate also when the solution lies on the boundary of the feasible set, some variants of the Frank-Wolfe algorithm were proposed in the literature (described below) [48].

Now, let us analyze the properties of the Frank-Wolfe method when applied to solve problem (4.1). Computing the vector y^k at Step 3 is extremely simple, as

Algorithm 14 `Frank-Wolfe method to solve problem` (4.1)

0 Choose a feasible point x^0
1 For $k = 0, 1, \ldots$
2 If x^k is a stationary point, then STOP
3 Compute $\hat{\imath} \in \underset{h=1,\ldots,n}{\text{Argmin}}\{\nabla_h f(x^k)\}$ and set $d^k = e_{\hat{\imath}} - x^k$
4 Set $\alpha^k_{\max} = 1$ and compute α^k by the Armijo line search (Algorithm 13)
5 Set $x^{k+1} = x^k + \alpha^k d^k$
6 Set $k = k + 1$
7 End for

the feasible set is a polytope and, from known results on linear programming, there exists a vertex of the feasible set that minimizes the function $\nabla f(x^k)^T x$.

Moreover, the set of vertices of the unit simplex is $\{e_1, \ldots, e_n\}$ and, for every $x = e_h$, $h = 1, \ldots, n$, we have that $\nabla f(x^k)^T x = \nabla_h f(x^k)$. Consequently, we can set $y^k = e_{\hat{\imath}}$, where $\hat{\imath} \in \underset{h=1,\ldots,n}{\text{Argmin}}\{\nabla_h f(x^k)\}$.

For what concerns the choice of the step length α^k, we can perform the Armijo line search by using, as the largest acceptable stepsize, the value $\alpha^k_{\max} = 1$, since $x^k + \alpha d^k$ is feasible for every $\alpha \in [0, 1]$.

For the sake of completeness, we report in Algorithm 14 the Frank-Wolfe method to solve problem (4.1).

As mentioned above, some variants of the Frank-Wolfe method were studied in the literature. Here, two versions are considered: the *Away-Step Frank-Wolfe method* (AFW) [75, 41] and the *Pairwise Frank-Wolfe method* (PFW)[58]. In particular, we only focus on minimization problems over the unit simplex, but the considerations that will be presented below can be easily extended to minimization problems over a polytope.

The distinguishing feature of the AFW method is to compute, at every iteration k, two potential search directions: the first one is the Frank-Wolfe direction described above, which will be referred to as d^{FW}; and the second one is the so called *Away-Step* direction d^{A}. The search direction d^{AFW} employed by the algorithm is the one, between d^{FW} and d^{A}, that minimizes $\nabla f(x^k)^T d$.

In particular, d^{A} is computed as $d^{\text{A}} = x^k - e_{\hat{\jmath}}$, where $e_{\hat{\jmath}}$ is the vertex that maximizes $\nabla f(x^k)^T e_h = \nabla_h f(x^k)$ with respect to the vertices e_h such that $x^k_h > 0$. In other words, $\hat{\jmath} \in \underset{h:\, x^k_h > 0}{\text{Argmax}}\{\nabla_h f(x^k)\}$.

It easy to verify that, if the search direction selected by the algorithm is d^{A} (i.e., if $\nabla f(x^k)^T d^{\text{A}} < \nabla f(x^k)^T d^{\text{FW}}$), then the largest acceptable stepsize that produces a feasible point is equal to $\dfrac{x^k_{\hat{\jmath}}}{1 - x^k_{\hat{\jmath}}}$, where $\hat{\jmath} \in \underset{h:\, x^k_h > 0}{\text{Argmax}}\{\nabla_h f(x^k)\}$.

We report in Algorithm 15 the Away-Step Frank-Wolfe algorithm to solve problem (4.1).

Now, we briefly examine the PFW method. At every iteration k, the method employs the *Pairwise Frank-Wolfe* direction d^{PFW}, computed as $d^{\text{PFW}} = d^{\text{FW}} + d^{\text{A}} =$

Algorithm 15 `Away-Step Frank-Wolfe method to solve problem` (4.1)

0 Choose a feasible point x^0
1 For $k = 0, 1, \ldots$
2 If x^k is a stationary point, then STOP
3 Compute $\hat{\imath} \in \underset{h=1,\ldots,n}{\text{Argmin}}\{\nabla_h f(x^k)\}$ and set $d^{\text{FW}} = e_{\hat{\imath}} - x^k$
4 Compute $\hat{\jmath} \in \underset{h:\, x_h^k > 0}{\text{Argmax}}\{\nabla_h f(x^k)\}$ and set $d^{\text{A}} = x^k - e_{\hat{\jmath}}$
5 If $\nabla f(x^k)^T d^{\text{FW}} \leq \nabla f(x^k)^T d^{\text{A}}$ then
6 Set $d^k = d^{\text{FW}}$ and $\alpha_{\max}^k = 1$
7 Else
8 Set $d^k = d^{\text{A}}$ and $\alpha_{\max}^k = \dfrac{x_{\hat{\jmath}}^k}{1 - x_{\hat{\jmath}}^k}$
9 End if
10 Compute α^k by the Armijo line search (Algorithm 13)
11 Set $x^{k+1} = x^k + \alpha^k d^k$
12 Set $k = k + 1$
13 End for

$e_{\hat{\imath}} - e_{\hat{\jmath}}$, where $\hat{\imath} \in \underset{h=1,\ldots,n}{\text{Argmin}}\{\nabla_h f(x^k)\}$ and $\hat{\jmath} \in \underset{h:\, x_h^k > 0}{\text{Argmax}}\{\nabla_h f(x^k)\}$. Using this search direction, it is easy to verify that the largest step length that guarantees feasibility is equal to $x_{\hat{\jmath}}^k$.

We report in Algorithm 16 the Pairwise Frank-Wolfe method to solve problem (4.1).

We notice that only two variables are updated by the PFW method at each iteration. It is also worth mentioning that such a framework is employed in some popular algorithmic schemes for training support vector machines [50, 65].

Finally, an in-depth analysis on the convergence rate of Frank-Wolfe variants for minimization problems over a polytope can be found in [48].

4.3.2 Combining FW variants with the active-set strategy

According to Step 7–8 of Algorithm 10, at every iteration k we have to compute a search direction d^k such that

$$d_{A^k}^k = 0,$$
$$\nabla f(\tilde{x}^k)^T d^k = \nabla_{N^k} f(\tilde{x}^k)^T d_{N^k}^k \leq 0.$$

Recalling Remark 3, the above conditions in practice imply that only the estimated non-active variables are moved along $d_{N^k}^k$, provided that $\tilde{x}^k$ does not satisfy stationarity conditions over N^k.

In this subsection, we focus on the case where $d_{N^k}^k$ is either the Frank-Wolfe direction, or one of its variants. Then, at the iteration k, two feasible directions d_{N^k} can be computed (in the subspace N^k):

Algorithm 16 `Pairwise Frank-Wolfe method to solve problem` (4.1)

0 Choose a feasible point x^0
1 For $k = 0, 1, \ldots$
2 If x^k is a stationary point, then STOP
3 Compute $\hat{\imath} \in \underset{h=1,\ldots,n}{\operatorname{Argmin}}\{\nabla_h f(x^k)\}$
4 Compute $\hat{\jmath} \in \underset{h:\, x_h^k>0}{\operatorname{Argmax}}\{\nabla_h f(x^k)\}$
5 Set $d^{\mathrm{A}} = e_{\hat{\imath}} - e_{\hat{\jmath}}$ and $\alpha^k_{\max} = x^k_{\hat{\jmath}}$
6 Compute α^k by the Armijo line search (Algorithm 13)
7 Set $x^{k+1} = x^k + \alpha^k d^k$
8 Set $k = k + 1$
9 End for

- the Frank-Wolfe direction

$$d^{\mathrm{FW}}_{N^k} = e_{\hat{\imath}} - \tilde{x}^k_{N^k}, \quad \hat{\imath} \in \underset{i\in N^k}{\operatorname{Argmin}}\{\nabla_i f(\tilde{x}^k)\}; \tag{4.35}$$

- the Away-Step direction

$$d^{\mathrm{A}}_{N^k} = \tilde{x}^k_{N^k} - e_{\hat{\jmath}}, \quad \hat{\jmath} \in \underset{j\in N_0^k}{\operatorname{Argmax}}\{\nabla_j f(\tilde{x}^k)\}, \tag{4.36}$$

  where $N_0^k = \{j \in N^k \colon \tilde{x}^k_j > 0\}$.

So, we can compute the final search direction d^k according to one of the following three rules:

- (FW) rule: $d^k_{N^k}$ is chosen as the Frank-Wolfe direction, that is,

$$\begin{aligned} d^k_{A^k} &= 0, \\ d^k_{N^k} &= d^{\mathrm{FW}}_{N^k}. \end{aligned}$$

  In this case, we simply write

$$d^k = d^{\mathrm{FW}}.$$

- (AFW) rule: $d^k_{N^k}$ is chosen as the Away-Step Frank-Wolfe direction, that is,

$$\begin{aligned} &d^k_{A^k} = 0, \\ &d^k_{N^k} = d^{\mathrm{AFW}}_{N^k} = \begin{cases} d^{\mathrm{FW}}_{N^k}, & \text{if } \nabla_{N^k} f(\tilde{x}^k)^T d^{\mathrm{FW}}_{N^k} \le \nabla_{N^k} f(\tilde{x}^k)^T d^{\mathrm{A}}_{N^k}, \\ d^{\mathrm{A}}_{N^k}, & \text{otherwise.} \end{cases} \end{aligned}$$

  In this case, we simply write

$$d^k = d^{\mathrm{AFW}}.$$

- (PFW) rule: $d^k_{N^k}$ is chosen as the Pairwise Frank-Wolfe direction, that is,

$$d^k_{A^k} = 0,$$
$$d^k_{N^k} = d^{\mathrm{PFW}}_{N^k} = d^{\mathrm{FW}}_{N^k} + d^{\mathrm{A}}_{N^k} = e_{\hat{\imath}} - e_{\hat{\jmath}},$$

  where $\hat{\imath}$ and $\hat{\jmath}$ are defined as in (4.35) and (4.36), respectively. In this case, we simply write

$$d^k = d^{\mathrm{PFW}}.$$

The following lemma claims that all the above three search directions are non-ascent directions.

Lemma 5. *Let $\tilde{x}^k$ be a feasible point generated by* `AS-SIMPLEX` *(Step 6) at iteration k. Let d^k be a search direction computed according to one among (FW), (AFW) and (PFW) rule. Then,*

$$\nabla f(\tilde{x}^k)^T d^k \leq 0.$$

Proof. First, we consider $d^k = d^{\mathrm{FW}}$. We can write

$$\nabla f(\tilde{x}^k)^T d^k = -\nabla f(\tilde{x}^k)^T \tilde{x} + \nabla_{\hat{\imath}} f(\tilde{x}^k),$$

where $\hat{\imath}$ is defined as in (4.35). Since $\tilde{x}_{A^k} = 0$, we also have that $\nabla f(\tilde{x}^k)^T \tilde{x}^k = \sum_{h \in N^k} \nabla_h f(\tilde{x}^k)\, \tilde{x}^k_h$. Consequently, we obtain

$$\nabla f(\tilde{x}^k)^T d^k \leq -\nabla_{\hat{\imath}} f(\tilde{x}^k) \sum_{h \in N^k} \tilde{x}^k_h + \nabla_{\hat{\imath}} f(\tilde{x}^k) = 0,$$

where the first inequality follows from the definition of $\hat{\imath}$ and the feasibility of $\tilde{x}^k$ and the last equality follows from the fact that $\tilde{x}_{A^k} = 0$.

Now, we consider $d^k = d^{\mathrm{AFW}}$. As we have already shown that the assertion holds when $d^k_{N^k} = d^{\mathrm{FW}}_{N^k}$, we only have to prove that $\nabla f(\tilde{x}^k)^T d^k \leq 0$ when $d^k_{N^k} = d^{\mathrm{A}}_{N^k}$. In this case, we can write

$$\nabla f(\tilde{x}^k)^T d^k = \nabla f(\tilde{x}^k)^T \tilde{x}^k - \nabla_{\hat{\jmath}} f(\tilde{x}^k),$$

where $\hat{\jmath}$ is defined as in (4.36). Reasoning as before, since $\tilde{x}_{A^k} = 0$, we have that $\nabla f(\tilde{x}^k)^T \tilde{x}^k = \sum_{h \in N^k} \nabla_h f(\tilde{x}^k)\, \tilde{x}^k_h$. Consequently, we obtain

$$\nabla f(\tilde{x}^k)^T d^k \leq \nabla_{\hat{\jmath}} f(\tilde{x}^k) \sum_{h \in N^k} \tilde{x}^k_h - \nabla_{\hat{\jmath}} f(\tilde{x}^k) = 0,$$

where the first inequality follows from the definition of $\hat{\jmath}$ and the feasibility of $\tilde{x}^k$ and the last equality follows from the fact that $\tilde{x}_{A^k} = 0$.

Finally, it is easy to see that the assertion is true when $d^k = d^{\mathrm{PFW}}$ as well, since $d^{\mathrm{PFW}} = d^{\mathrm{FW}} + d^{\mathrm{AFW}}$. □

In the next lemma, we show that at every point $\tilde{x}^k$ produced by `AS-SIMPLEX`, the directional derivative along d^{PFW} is not larger than the directional derivative along d^{AFW}. This fact will play a crucial role for proving the convergence of the algorithm for all the considered variants of the Frank-Wolfe direction.

Lemma 6. *Let $\tilde{x}^k$ be a feasible point generated by* `AS-SIMPLEX` *at iteration k. Then,*

$$\nabla f(\tilde{x}^k)^T d^{PFW} \leq \nabla f(\tilde{x}^k)^T d^{AFW}.$$

Proof. In the following, we indicate with $\hat{\imath}$ and $\hat{\jmath}$ the indices defined as in (4.35)–(4.36), respectively. Exploiting the feasibility of $\tilde{x}^k$, and the fact that $\tilde{x}_{A^k} = 0$, we can write

$$\begin{aligned}
\nabla f(\tilde{x}^k)^T d^{\mathrm{FW}} &= \nabla_{\hat{\imath}} f(\tilde{x}^k) - \nabla f(\tilde{x}^k)^T \tilde{x}^k = \nabla_{\hat{\imath}} f(\tilde{x}^k) - \sum_{h \in N^k} \nabla_h f(\tilde{x}^k)\, \tilde{x}_h^k \\
&\geq \nabla_{\hat{\imath}} f(\tilde{x}^k) - \nabla_{\hat{\jmath}} f(\tilde{x}^k) \sum_{h \in N^k} \tilde{x}_h^k = \nabla_{\hat{\imath}} f(\tilde{x}^k) - \nabla_{\hat{\jmath}} f(\tilde{x}^k) \\
&= \nabla f(\tilde{x}^k)^T d^{\mathrm{PFW}}.
\end{aligned}$$

Similarly, we have that

$$\begin{aligned}
\nabla f(\tilde{x}^k)^T d^{\mathrm{A}} &= \nabla f(\tilde{x}^k)^T \tilde{x}^k - \nabla_{\hat{\jmath}} f(\tilde{x}^k) = \sum_{h \in N^k} \nabla_h f(\tilde{x}^k)\, \tilde{x}_h^k - \nabla_{\hat{\jmath}} f(\tilde{x}^k) \\
&\geq \nabla_{\hat{\imath}} f(\tilde{x}^k) \sum_{h \in N^k} \tilde{x}_h^k - \nabla_{\hat{\jmath}} f(\tilde{x}^k) = \nabla_{\hat{\imath}} f(\tilde{x}^k) - \nabla_{\hat{\jmath}} f(\tilde{x}^k) \\
&= \nabla f(\tilde{x}^k)^T d^{\mathrm{PFW}}.
\end{aligned}$$

From the above relations, we get

$$\nabla f(\tilde{x}^k)^T d^{\mathrm{PFW}} \leq \min\{\nabla f(\tilde{x}^k)^T d^{\mathrm{FW}}, \nabla f(\tilde{x}^k)^T d^{\mathrm{A}}\} = \nabla f(\tilde{x}^k)^T d^{\mathrm{AFW}}.$$

□

4.3.3 Computation of the stepsize

Now, we can show the convergence results of the line search procedure for all the considered variants of the Frank-Wolfe direction, which will enable us to prove the global convergence of `AS-SIMPLEX`.

Depending on the direction d^k used in the line search procedure, the maximum stepsize $\alpha_{\max}^k$ is set as follows:

(i) Frank-Wolfe direction: $\alpha_{\max}^k = 1$;

(ii) Away-Step Frank-Wolfe direction:

- if $d_{N^k}^k = d_{N^k}^{\mathrm{FW}}$, then $\alpha_{\max}^k = 1$;
- if $d_{N^k}^k = d_{N^k}^{\mathrm{A}}$, then $\alpha_{\max}^k = \tilde{x}_{\hat{\jmath}}^k / (1 - \tilde{x}_{\hat{\jmath}}^k)$ where $\hat{\jmath}$ is the index defined as in (4.36);

(iii) Pairwise direction: $\alpha_{\max}^k = \tilde{x}_{\hat{\jmath}}^k$, where $\hat{\jmath}$ is the index defined as in (4.36).

It is easy to verify that, for every considered search direction d^k, this choice guarantees that $\tilde{x}^k + \alpha d^k$ is feasible for all $\alpha \in (0, \alpha_{\max}^k]$.

Now, we prove a theorem that follows from classical results on the Armijo line search. It guarantees that $\|\tilde{x}^k - x^k\|$ converges to zero and that the sequence of the directional derivatives along the search direction converges to zero as well, for all the considered search directions.

Theorem 15. *Let Assumption 2 hold. Let $\{x^k\}$, $\{\tilde{x}^k\}$ and $\{d^k\}$ be the sequences produced by* `AS-SIMPLEX`*, where d^k is computed at Step 7–8 according to one among (FW), (AFW) and (PFW) rule. If* `AS-SIMPLEX` *does not terminate in a finite number of iterations, then*

$$\lim_{k\to\infty} \|\tilde{x}^k - x^k\| = 0, \tag{4.37}$$

$$\lim_{k\to\infty} \nabla f(\tilde{x}^k)^T d^k = 0. \tag{4.38}$$

Proof. First, we observe that, from standard results on the Armijo line search, Algorithm 11 computes α^k in a finite number of steps at every iteration k for which $\nabla f(\tilde{x}^k)^T d^k < 0$.

Now, we prove (4.37). From the instructions of the algorithm and Proposition 16, we can write

$$f(x^{k+1}) \leq f(\tilde{x}^k) \leq f(x^k) - L\|\tilde{x}^k - x^k\|^2. \tag{4.39}$$

From the continuity of the objective function and the compactness of the feasible set, it follows that

$$\lim_{k\to\infty} [f(x^{k+1}) - f(x^k)] = 0. \tag{4.40}$$

The above relation, combined with (4.39), proves (4.37).

To prove (4.38), we consider separately the iterations in which $\nabla f(\tilde{x}^k)^T d^k < 0$ from those in which $\nabla f(\tilde{x}^k)^T d^k = 0$. Namely, we identify two iteration index subsets $H, K \subseteq \{1, 2, \dots\}$, such that:

- $\nabla f(\tilde{x}^k)^T d^k < 0$, for all $k \in K$;
- $H = \{1, 2, \dots\} \setminus K$.

By assumption, the algorithm does not terminate in a finite number of iterations, and then, at least one of the above sets is infinite. Since we are interested in the asymptotic behavior of the sequence produced by `AS-SIMPLEX`, we assume without loss of generality that both H and K are infinite sets.

From the instructions of the algorithm, it is straightforward to verify that

$$\lim_{k\to\infty,\, k\in H} \nabla f(\tilde{x}^k)^T d^k = 0.$$

Therefore, we limit our analysis to consider the subsequence $\{x^k\}_K$. For all $k \in K$, since $\nabla f(\tilde{x}^k)^T d^k < 0$, the line search procedure computes a value $\alpha^k \in (0, 1]$ in a finite number of iterations, such that

$$f(x^{k+1}) \leq f(\tilde{x}^k) + \gamma\, \alpha^k\, \nabla f(\tilde{x}^k)^T d^k, \quad \forall k \in K,$$

or equivalently,

$$f(\tilde{x}^k) - f(x^{k+1}) \geq \gamma\, \alpha^k\, |\nabla f(\tilde{x}^k)^T d^k|, \quad \forall k \in K,$$

From (4.37) and (4.40), we get that the left-hand side of the above inequality converges to zero, hence

$$\lim_{k\to\infty} \alpha^k\, |\nabla f(\tilde{x}^k)^T d^k| = 0. \tag{4.41}$$

Now, proceeding by contradiction, we assume that (4.38) does not hold. From the compactness of the feasible set, $\{x^k\}_K$ attains limit points. Let $\bar{x}$ be any limit point of $\{x^k\}_K$. Using (4.37), since $\{x^k\}$, $\{\tilde{x}^k\}$ and $\{d^k\}$ are bounded, and taking into account that A^k and N^k are subsets of a finite set of indices, without loss of generality we redefine $\{x^k\}_K$ the subsequence such that

$$\lim_{k\to\infty,\, k\in K} x^k = \lim_{k\to\infty,\, k\in K} \tilde{x}^k = \bar{x}, \tag{4.42}$$

and

$$A^k = \hat{A}, \quad N^k = \hat{N}, \quad \forall k \in K, \tag{4.43}$$

$$\lim_{k\to\infty,\, k\in K} d^k = \bar{d}. \tag{4.44}$$

As we have assumed that (4.38) does not hold, then the above relations, combined with the continuity of the gradient, imply that

$$\lim_{k\to\infty,\, k\in K} \nabla f(\tilde{x}^k)^T d^k = \nabla f(\bar{x})^T \bar{d} = -\eta < 0. \tag{4.45}$$

We first prove that, if (4.45) holds, then $M > 0$ exists such that

$$\alpha_{\max}^k \geq M, \quad \forall k \in K. \tag{4.46}$$

By contradiction, let us assume that an infinite subset of K (that we denote with K for simplicity) exists such that

$$\lim_{k\to\infty,\, k\in K} \alpha_{\max}^k = 0. \tag{4.47}$$

We distinguish three different cases, depending on the strategy used for computing the direction d^k at Step 7–8 in Algorithm 10:

- Case (FW): it is easy to see that we get a contradiction since $\alpha_{\max}^k$ has a constant value equal to 1.
- Case (AFW): recalling the definition of d^{AFW}, the case we need to analyze is the one where we get an infinite subsequence of Away-Step directions in N^k. So, we assume that an infinite subset $\tilde{K} \subseteq K$ exists such that

  $$d_{N^k}^k = d_{N^k}^{\mathrm{A}}, \quad \forall k \in \tilde{K}.$$

  We have that $\alpha_{\max}^k = \dfrac{\tilde{x}_{\hat{\jmath}}^k}{1 - \tilde{x}_{\hat{\jmath}}^k}$, for all $k \in \tilde{K}$, where $\hat{\jmath}$ is the index computed according to (4.36). Since the number of indices in $\hat{N}$ is finite, we can consider a further subsequence (that we denote with $\tilde{K}$ for simplicity), where the index $\hat{\jmath}$ is fixed. Taking into account (4.47), it is easy to see that

  $$\lim_{k\to\infty,\, k\in\tilde{K}} \tilde{x}_{\hat{\jmath}}^k = 0. \tag{4.48}$$

Now, from (4.45), (4.48) and the continuity of $\nabla f(x)$, it follows that an index $\tilde{k} \in \tilde{K}$ exists such that, for all $k \geq \tilde{k}$, $k \in \tilde{K}$, we have that

$$\nabla f(\tilde{x}^k)^T d^k = \nabla f(\tilde{x}^k)^T (\tilde{x}^k - e_{\hat{\jmath}}) \leq -\frac{\eta}{2},$$
$$\tilde{x}^k_{\hat{\jmath}} \leq \epsilon \frac{\eta}{2}.$$

Therefore, we obtain

$$\tilde{x}^k_{\hat{\jmath}} \leq \epsilon \nabla f(x^k)^T (e_{\hat{\jmath}} - \tilde{x}^k), \quad \forall k \geq \tilde{k}, k \in \tilde{K}.$$

Recalling (4.11), this implies that $\hat{\jmath} \in \hat{A}$ and, considering the definition of $\hat{\jmath}$ in (4.36), we get a contradiction.

- Case (PFW): in this case $\alpha^k_{\max} = \tilde{x}^k_{\hat{\jmath}}$ and the contradiction follows from the same reasoning done above for (AFW).

So, (4.46) holds.

Now, from (4.41) and (4.45), we get

$$\lim_{k \to \infty, k \in K} \alpha^k = 0. \tag{4.49}$$

Taking into account (4.46), it follows that a value $\bar{k} \in K$ exists such that

$$\alpha^k < \alpha^k_{\max}, \quad \forall k \geq \bar{k}, k \in K.$$

In other words, for $k \geq \bar{k}$, $k \in K$, the stepsize α^k cannot be set equal to the maximum stepsize and, taking into account the line search procedure, we can write

$$f\Big(\tilde{x}^k + \frac{\alpha^k}{\delta} d^k\Big) > f(\tilde{x}^k) + \gamma \frac{\alpha^k}{\delta} \nabla f(\tilde{x}^k)^T d^k, \quad \forall k \geq \bar{k},\ k \in K. \tag{4.50}$$

We can apply the mean value theorem and we have that $\xi_k \in (0,1)$ exists such that

$$f\Big(\tilde{x}^k + \frac{\alpha^k}{\delta} d^k\Big) = f(\tilde{x}^k) + \frac{\alpha^k}{\delta} \nabla f\Big(\tilde{x}^k + \xi_k \frac{\alpha^k}{\delta} d^k\Big)^T d^k, \quad \forall k \geq \bar{k},\ k \in K. \tag{4.51}$$

By substituting (4.51) within (4.50), we have

$$\nabla f\Big(\tilde{x}^k + \xi_k \frac{\alpha^k}{\delta} d^k\Big)^T d^k > \gamma \nabla f(\tilde{x}^k)^T d^k. \tag{4.52}$$

From (4.49), (4.37), and exploiting the fact that $\{\xi_k\}$ and $\{d^k\}$ are bounded, we also get

$$\lim_{k \to \infty, k \in K} \tilde{x}^k + \xi_k \frac{\alpha^k}{\delta} d^k = \lim_{k \to \infty, k \in K} \tilde{x}^k = \bar{x}. \tag{4.53}$$

Finally, from (4.45), (4.52) and (4.53), we obtain

$$-\eta = \nabla f(\bar{x})^T \bar{d} \geq \gamma \nabla f(\bar{x})^T \bar{d} = -\gamma\eta,$$

which is a contradiction, since we set $\gamma < 1$ in `AS-SIMPLEX`. □

4.4 Global convergence analysis

In this section, we prove the global convergence of `AS-SIMPLEX` to stationary points, for every considered choice of the direction d^k.

Remark 4. *From Proposition 13 and 14, it follows that Algorithm 10 is well defined by employing any direction d^k among d^{FW}, d^{AFW} and d^{PFW}, in the sense that a new point x^{k+1} is computed if and only if x^k is non-stationary.*

Theorem 16. *Let Assumption 2 hold and let $\{x^k\}$ be the sequence of points produced by `AS-SIMPLEX`. Let us assume that the search direction d^k is computed according to one among (FW), (AFW) and (PFW) rule.*

Then, either an integer $\bar{k} \geq 0$ exists such that $x^{\bar{k}}$ is a stationary point for problem (4.1)*, or the sequence $\{x^k\}$ is infinite and every limit point x^* of the sequence is a stationary point for problem* (4.1)*.*

Proof. First, we consider the case where d^k is computed according to (FW) rule, that is, $d^k = d^{\mathrm{FW}}$. Then, we will prove the remaining two cases.

Let $\{x^k\}$ be the sequence produced by Algorithm 10 and let us assume that a stationary point is not produced in a finite number of iterations. Since the feasible set is compact, then the sequence $\{x^k\}$ attains a limit point x^* and, recalling (4.37) of Theorem 15, there exists $K \subseteq \mathbb{N}$ such that

$$\lim_{k\to\infty,\, k\in K} x^k = \lim_{k\to\infty,\, k\in K} \tilde{x}^k = x^*. \tag{4.54}$$

Let $\Phi_i(x)$ be the continuous function defined as

$$\Phi_i(x) = \max\{0, -\nabla f(x)^T(e_i - x)\},$$

that measures the violation of the optimality conditions for a variable x_i, $i = 1, \ldots, n$.

By contradiction, we assume that x^* is not a stationary point, so that an index $\nu \in \{1, \ldots, n\}$ exists such that

$$|\Phi_\nu(x^*)| > 0. \tag{4.55}$$

Taking into account that the number of possible different choices of A^k and N^k is finite, we can find a subset of iteration indices $\bar{K} \subseteq K$ such that $A^k = \hat{A}$ and $N^k = \hat{N}$ for all $k \in \bar{K}$.

First, suppose that $\nu \in \hat{A}$. Then, by definition, we can write

$$0 \leq x^k_\nu \leq \epsilon \nabla f(x^k)^T(e_\nu - x^k),$$

so that

$$\Phi_\nu(x^k) = \max\{0, -\nabla f(x^k)^T(e_\nu - x^k)\} = 0,$$

for all $k \in \bar{K}$. Thus, from (4.54) and the continuity of the function $\Phi_i(\cdot)$, we get a contradiction with (4.55).

Now, suppose that $\nu \in \hat{N}$. We can choose an index $\bar{\nu} \in \{1, \ldots, n\}$ and a further subset of iteration indices $\hat{K} \subseteq \bar{K}$ such that

$$\Phi_{\bar{\nu}}(\tilde{x}^k) = \max_{i\in\hat{N}}\{\Phi_i(\tilde{x}^k)\}, \quad \forall\, k \in \hat{K}.$$

Hence,

$$\Phi_{\bar{\nu}}(\tilde{x}^k) \geq \Phi_\nu(\tilde{x}^k) \geq 0, \quad \forall\ k \in \hat{K},$$

which, by continuity of $\Phi_i(\cdot)$, implies that

$$\Phi_{\bar{\nu}}(x^*) \geq \Phi_\nu(x^*) > 0. \tag{4.56}$$

From the definition of $\Phi_i(x)$ and $\bar{\nu}$, for all $k \in \hat{K}$ we can write

$$\begin{aligned} \Phi_{\bar{\nu}}(\tilde{x}^k) &= \max_{i \in \hat{N}} \{\max\{0, -\nabla f(\tilde{x}^k)^T(e_i - \tilde{x}^k)\}\} \\ &= -\min_{i \in \hat{N}} \{\nabla f(\tilde{x}^k)^T(e_i - \tilde{x}^k)\} \\ &= -\nabla f(\tilde{x}^k)^T d^{\mathrm{FW}}. \end{aligned} \tag{4.57}$$

Since we are considering the case where $d^k = d^{\mathrm{FW}}$, from (4.54), (4.38) of Theorem 15 and the continuity of $\Phi_i(\cdot)$, we have that

$$0 = \lim_{\substack{k \to \infty \\ k \in \hat{K}}} \nabla f(\tilde{x}^k)^T d^k = \lim_{\substack{k \to \infty \\ k \in \hat{K}}} -\Phi_{\bar{\nu}}(\tilde{x}^k) = -\Phi_{\bar{\nu}}(x^*),$$

which, combined with (4.56), implies that $\Phi_\nu(x^*) = 0$, thus contradicting (4.55). Then, the assertion is proved for $d^k = d^{\mathrm{FW}}$.

Now, we consider together the cases where $d^k = d^{\mathrm{AFW}}$ and $d^k = d^{\mathrm{PFW}}$. In both cases, we can apply the same reasoning made before and we obtain (4.57) again. Recalling the definition of d^{AFW} and Lemma 6, we can write

$$-\nabla f(\tilde{x}^k)^T d^k \geq -\nabla f(\tilde{x}^k)^T d^{\mathrm{FW}} = \Phi_{\bar{\nu}}(\tilde{x}^k),$$

for both $d^k = d^{\mathrm{AFW}}$ and $d^k = d^{\mathrm{PFW}}$. Consequently,

$$0 = \lim_{\substack{k \to \infty \\ k \in \hat{K}}} \nabla f(\tilde{x}^k)^T d^k \leq \lim_{\substack{k \to \infty \\ k \in \hat{K}}} -\Phi_{\bar{\nu}}(\tilde{x}^k) = -\Phi_{\bar{\nu}}(x^*),$$

which, combined with (4.56), implies that $\Phi_\nu(x^*) = 0$, thus contradicting (4.55). Then, the assertion is also proved for $d^k = d^{\mathrm{AFW}}$ and $d^k = d^{\mathrm{PFW}}$. □

4.5 Extension to minimization problems over the ℓ_1-ball

In this section, we focus on the following problem:

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} \ h(x) \\ &\|x\|_1 \leq \tau, \end{aligned} \tag{4.58}$$

with $h \in C^1(\mathbb{R}^n)$, $\nabla h(x)$ Lipschitz continuous on the feasible set and $\tau > 0$.

First, let us briefly analyze some theoretical properties concerning the equivalence between a minimization problem over a polytope and a minimization problem over the unit simplex. As mentioned in Section 4.1, we can rewrite as a minimization problem over the unit simplex every problem of the form

$$\begin{aligned} &\min_{x} \ h(x) \\ &x \in D, \end{aligned} \tag{4.59}$$

where $h\colon R^n \to R$ and D is the convex hull of m vectors. Namely,

$$D = \text{conv}\{v_1, \ldots, v_m\},$$

with $v_i \in \mathbb{R}^n$, $i = 1, \ldots, m$. Introducing the variable vector $\alpha \in \mathbb{R}^m$, we obtain the following equivalent problem:

$$\begin{aligned} &\min_{\alpha} f(\alpha) = h(V\alpha) \\ &\sum_{i=1}^{m} \alpha_i = 1 \\ &\alpha_i \geq 0, \quad i = 1, \ldots, m, \end{aligned} \tag{4.60}$$

where $V = \begin{bmatrix} v_1 & \ldots & v_m \end{bmatrix} \in \mathbb{R}^{n \times m}$. So, given a feasible point α of problem (4.60), we have that $x = V\alpha$ is a feasible point of problem (4.59). If $h(x)$ is continuously differentiable, we get the gradient of $f(\alpha)$ as

$$\nabla f(\alpha) = V^T \nabla h(V\alpha).$$

Moreover, the next proposition shows a correspondence between the stationary points of (4.59) and those of (4.60).

Proposition 17. *Let us consider problems* (4.59) *and* (4.60)*. A point α^* is stationary for* (4.60) *if and only if the point the $x^* = V\alpha^*$ satisfies*

$$\nabla h(x^*)^T (x - x^*) \geq 0, \quad \forall x \in D.$$

Proof. Let us denote the unitary simplex by Δ. Fist, we prove that α^* satisfies KKT conditions for (4.60) if and only if

$$\nabla f(\alpha^*)^T (\alpha - \alpha^*) \geq 0, \quad \forall \alpha \in \Delta. \tag{4.61}$$

Assuming that (4.61) holds, let i be an index such that $\alpha_i^* > 0$ and let j be any index in $\{1, \ldots, m\}$, with $j \neq i$. Then, we can consider the point α such that $\alpha_i = 0$, $\alpha_j = \alpha_j^* + \alpha_i^*$ and $\alpha_h = \alpha_h^*$, $h \in \{1, \ldots, m\} \setminus \{i, j\}$. It follows that $\nabla f(\alpha^*)^T (\alpha - \alpha^*) = [-\nabla_i f(\alpha^*) + \nabla_j f(\alpha^*)]\alpha_i^* \geq 0$, implying that $\nabla_j f(\alpha^*) \geq \nabla_i f(\alpha^*)$. Consequently, recalling Proposition 12, we have that α^* satisfies KKT conditions.
Now, assume that α^* satisfies KKT conditions (4.2)–(4.4) (with x^* replaced by α^*), with multipliers $\mu^* \in \mathbb{R}^m$ and $\lambda^* \in \mathbb{R}$. Exploiting Corollary 1, we have that

$$\nabla_j f(\alpha^*) = \lambda^*, \quad \forall j\colon \alpha_j^* > 0, \tag{4.62}$$

$$\nabla_i f(\alpha^*) \geq \lambda^*, \quad \forall i\colon \alpha_i^* = 0. \tag{4.63}$$

Moreover, using (4.7) (with x^* replaced by α^*), we also have that

$$\nabla f(\alpha^*)^T \alpha^* = \lambda^*. \tag{4.64}$$

From (4.62), (4.63) and (4.64), for every $\alpha \in \Delta$ we can write

$$\nabla f(\alpha^*)^T (\alpha - \alpha^*) = \sum_{i=1}^{m} \nabla_i f(\alpha^*)\, \alpha_i - \nabla f(\alpha^*)^T \alpha^* \geq \lambda^* \sum_{i=1}^{m} \alpha_i - \lambda^* = 0$$

and then (4.61) holds.
Finally, since $x^* = V\alpha^*$, we can write

$$\begin{aligned}
&\nabla f(\alpha^*)^T(\alpha - \alpha^*) \geq 0, \quad \forall \alpha \in \Delta \quad \Leftrightarrow \\
&\left[V^T \nabla h(V\alpha^*)\right]^T(\alpha - \alpha^*) \geq 0, \quad \forall \alpha \in \Delta \Leftrightarrow \\
&\nabla h(V\alpha^*)^T[V(\alpha - \alpha^*)] \geq 0, \quad \forall \alpha \in \Delta \Leftrightarrow \\
&\nabla h(x^*)^T(x - x^*) \geq 0, \quad \forall x \in D.
\end{aligned}$$

□

Now, let us consider problem (4.58). Every feasible point can be expressed as a convex combination of the vertices $\{\pm\tau e_1, \ldots, \pm\tau e_n\}$ and, consequently, we could rewrite (4.58) as a minimization problem over the unit simplex and then apply `AS-SIMPLEX` to solve it. Unfortunately, the number of variables would become $2n$.

So, in this section we describe how we can adapt our algorithm to solve (4.58) in the original space $\mathbb{R}^n$, avoiding to double the number of variables. As to be shown, this can be achieved thanks to the properties of the active-set estimate, which enable us to link the variables that are estimated active for the transformed problem with those that are estimated active for the original problem (4.58) (i.e., those variables x_i that are estimated to be zero at the stationary point).

We start by introducing the slack variable $z \in \mathbb{R}$, in order to reformulate (4.58) as follows:

$$\begin{aligned}
&\min_{x \in \mathbb{R}^n,\, z \in \mathbb{R}} \ \bar{h}\left(\begin{bmatrix} x \\ z \end{bmatrix}\right) \\
&\|x\|_1 + z = \tau \\
&z \geq 0,
\end{aligned} \tag{4.65}$$

where $\bar{h}\colon \mathbb{R}^{n+1} \to \mathbb{R}$ is defined such that $\bar{h}\left(\begin{bmatrix} x \\ z \end{bmatrix}\right) = h(x)$, for every $\begin{bmatrix} x \\ z \end{bmatrix} \in \mathbb{R}^{n+1}$.

Exploiting the fact that every feasible point of (4.65) can be expressed as a convex combination of the vertices, we can write the following equivalent problem:

$$\begin{aligned}
&\min_{\alpha \in \mathbb{R}^{2n+1}} \ f(\alpha) = \bar{h}(M\alpha) \\
&e^T\alpha = 1 \\
&\alpha \geq 0,
\end{aligned} \tag{4.66}$$

by performing the following variable transformation:

$$\begin{bmatrix} x \\ z \end{bmatrix} = M\alpha,$$

where

$$M = \tau \left[\begin{array}{c|c|c} I & -I & \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} \\ \hline 0 \ \ldots \ 0 & 0 \ \ldots \ 0 & 1 \end{array}\right] \in \mathbb{R}^{(n+1)\times(2n+1)}. \tag{4.67}$$

Equivalently,

$$\begin{aligned} x_i &= \tau\,(\alpha_i - \alpha_{n+i}), \quad i = 1, \dots, n, \\ z &= \tau\,\alpha_{2n+1}. \end{aligned} \tag{4.68}$$

Now, we associate to every feasible point of (4.65) a particular point $\alpha \in \mathbb{R}^{2n+1}$ that satisfies system (4.68):

$$\begin{aligned} \alpha_i &= \frac{1}{\tau}\max\{0, x_i\}, \quad i = 1, \dots, n, \\ \alpha_{n+i} &= \frac{1}{\tau}\max\{0, -x_i\}, \quad i = 1, \dots, n, \\ \alpha_{2n+1} &= \frac{1}{\tau}\,z. \end{aligned} \tag{4.69}$$

As mentioned above, we want to show that for every feasible point x of problem (4.58), there exists a correspondence between the estimated active variables x_i (i.e., those variables that are estimated to be zero at the stationary point) and the variables α_i (obtained by (4.69)) that are estimated active for problem (4.66).

In particular, for every feasible point x of problem (4.58), we can build the following index sets:

- $A_{\ell_1}(x)$, which contains the indices of the estimated active variables;
- $N_{\ell_1}(x)$, which contains the indices of the estimated non-active variables.

Using (4.69), and recalling that α_i are nonnegative, it follows that

$$x_i = 0 \Leftrightarrow \alpha_i = \alpha_{n+i} = 0.$$

Then, we estimate x_i active for (4.58) if both α_i and α_{n+i} are estimated active for (4.66). Consequently, we define $A_{\ell_1}(x)$ and $N_{\ell_1}(x)$ as follows:

$$A_{\ell_1}(x) = \{i \in \{1, \dots, n\} \colon i \in A(\alpha) \text{ and } (i+n) \in A(\alpha)\}, \tag{4.70}$$

$$N_{\ell_1}(x) = \{i \in \{1, \dots, n\} \colon i \in N(\alpha) \text{ or } (i+n) \in N(\alpha)\}. \tag{4.71}$$

Now, we describe how to build the above index sets without explicitly dealing with problem (4.66). Let x be a feasible point of (4.58). We can easily set $z = \tau - \|x\|_1$ and then compute α by (4.69). Moreover, we can write

$$\nabla f(\alpha) = M^T \nabla \bar{h}(x) = M^T \begin{bmatrix} \nabla h(x) \\ 0 \end{bmatrix} = \tau \begin{bmatrix} \nabla_1 h(x) \\ \vdots \\ \nabla_n h(x) \\ -\nabla_1 h(x) \\ \vdots \\ -\nabla_n h(x) \\ 0 \end{bmatrix} \tag{4.72}$$

and then

$$\nabla f(\alpha)^T \alpha = \begin{bmatrix} \nabla h(x)^T & 0 \end{bmatrix} M\alpha = \nabla h(x)^T x. \tag{4.73}$$

For each index $i = 1, \dots, n$, we can distinguish two cases:

(i) $x_i \geq 0$. From (4.69), we have

$$\begin{cases} \alpha_i = \dfrac{1}{\tau} x_i \geq 0, \\ \alpha_{n+i} = 0. \end{cases}$$

Recalling (4.11)–(4.12), and using (4.69), (4.72) and (4.73), we can write

$$\begin{aligned} i \in A(\alpha) \Leftrightarrow 0 \leq \frac{1}{\tau} x_i = \alpha_i &\leq \epsilon \nabla f(\alpha)^T (e_i - \alpha) \\ &= \epsilon (\nabla_i f(\alpha) - \nabla f(\alpha)^T \alpha) \\ &= \epsilon (\tau \nabla_i h(x) - \nabla h(x)^T x) \\ &= \epsilon \nabla h(x)^T (\tau e_i - x) \end{aligned} \tag{4.74}$$

and

$$\begin{aligned} n + i \in A(\alpha) \Leftrightarrow -\frac{1}{\tau} x_i \leq 0 = \alpha_{n+i} &\leq \epsilon \nabla f(\alpha)^T (e_{n+i} - \alpha) \\ &= \epsilon (\nabla_{n+i} f(\alpha) - \nabla f(\alpha)^T \alpha) \\ &= \epsilon (-\tau \nabla_i h(x) - \nabla h(x)^T x) \\ &= -\epsilon \nabla h(x)^T (\tau e_i + x). \end{aligned} \tag{4.75}$$

(ii) $x_i < 0$. From (4.69), we have

$$\begin{cases} \alpha_i = 0, \\ \alpha_{n+i} = -\dfrac{1}{\tau} x_i > 0. \end{cases}$$

Similarly to the previous case, we can write

$$\begin{aligned} i \in A(\alpha) \Leftrightarrow \frac{1}{\tau} x_i < 0 = \alpha_i &\leq \epsilon \nabla f(\alpha)^T (e_i - \alpha) \\ &= \epsilon (\nabla_i f(\alpha) - \nabla f(\alpha)^T \alpha) \\ &= \epsilon (\tau \nabla_i h(x) - \nabla h(x)^T x) \\ &= \epsilon \nabla h(x)^T (\tau e_i - x) \end{aligned} \tag{4.76}$$

and

$$\begin{aligned} (n + i) \in A(\alpha) \Leftrightarrow 0 < -\frac{1}{\tau} x_i = \alpha_{n+i} &\leq \epsilon \nabla f(\alpha)^T (e_{n+i} - \alpha) \\ &= \epsilon (\nabla_{n+i} f(\alpha) - \nabla f(\alpha)^T \alpha) \\ &= \epsilon (-\tau \nabla_i h(x) - \nabla h(x)^T x) \\ &= -\epsilon \nabla h(x)^T (\tau e_i + x). \end{aligned} \tag{4.77}$$

From (4.74), (4.75), (4.76), (4.77), and recalling (4.70)–(4.71), we obtain

$$A_{\ell_1}(x) = \{i \colon \epsilon \tau \nabla h(x)^T (\tau e_i + x) \leq x_i \leq \epsilon \tau \nabla h(x)^T (\tau e_i - x)\}, \tag{4.78}$$

$$N_{\ell_1}(x) = \{1, \ldots, n\} \setminus A_{\ell_1}(x). \tag{4.79}$$

Now, we show how the algorithmic framework described in the previous section can be easily adapted to problem (4.58), using the active and non-active estimates (4.78)–(4.79). We first need to define the following index set:

$$J_{\ell_1}(x) = \Big\{ j \in \{1, \dots, n\} \colon j \in \operatorname*{Argmax}_{i=1,\dots,n} \{|\nabla_i h(x)|\} \Big\}. \tag{4.80}$$

We can show that for every non-stationary point x of problem (4.58), the set $J_{\ell_1}(x) \cap N_{\ell_1}(x)$ is non-empty and we are able to get a sufficient reduction in the objective function by setting to zero the variables belonging to $A_{\ell_1}(x)$ and updating a variable x_j, with $j \in J_{\ell_1}(x) \cap N_{\ell_1}(x)$.

Proposition 18. *Let x be a feasible point of problem* (4.58) *and assume that x is non-stationary. Then,*

$$N_{\ell_1}(x) \cap J_{\ell_1}(x) \neq \emptyset.$$

Proof. Let α be the point given by (4.69). Considering problem (4.66), we can compute the active and non-active set estimates $A(\alpha)$, $N(\alpha)$.

From (4.72), and exploiting the hypothesis that x is non-stationary, it follows that

$$\min_{i=1,\dots,2n+1} \{\nabla_i f(\alpha)\} < 0.$$

In particular, this implies that

$$(2n+1) \notin \operatorname*{Argmin}_{i=1,\dots,2n+1} \{\nabla_i f(\alpha)\}.$$

Hence, exploiting Proposition 15, there exists $\nu \in \{1, \dots, 2n\}$ such that

$$\nu \in \operatorname*{Argmin}_{i=1,\dots,2n} \{\nabla_i f(\alpha)\}, \tag{4.81}$$

$$\nu \in N(\alpha). \tag{4.82}$$

Recalling (4.72), we can rewrite (4.81) as

$$\nabla_\nu f(\alpha) \leq \min_{i=1,\dots,n} \{\tau \nabla_1 h(x), \dots, \tau \nabla_n h(x), -\tau \nabla_1 h(x), \dots, -\tau \nabla_n h(x)\}, \tag{4.83}$$

that is,

$$-|\nabla_\nu f(\alpha)| \leq -\tau |\nabla_i h(x)|, \quad \forall i = 1, \dots, n. \tag{4.84}$$

Now, we can define the index $j \in \{1, \dots, n\}$ as

$$j = \begin{cases} \nu, & \text{if } \nu \in \{1, \dots, n\}, \\ \nu - n, & \text{if } \nu \in \{n+1, \dots, 2n\}. \end{cases} \tag{4.85}$$

Using (4.72) again, we have

$$|\nabla_\nu f(\alpha)| = |\nabla_j f(\alpha)| = \tau |\nabla_j h(x)|.$$

The previous relation, combined with (4.84), implies that

$$j \in \operatorname*{Argmax}_{i=1,\dots,n} \{|\nabla_i h(x)|\}.$$

Using (4.82) and (4.85), it follows that either $j \in N(\alpha)$, or $(j+n) \in N(\alpha)$. Recalling (4.71), it follows that

$$j \in N_{\ell_1}(x)$$

and then the assertion is proved. □

Now, we need an assumption on the parameter ϵ used in the active and non-active set estimates.

Assumption 3. *Assume that the parameter ϵ appearing in the estimates* (4.78)–(4.79) *satisfies the following conditions:*

$$0 < \epsilon \leq \frac{1}{4\tau^2 L(2n+1)}, \tag{4.86}$$

where L is the Lipschitz constant of $\nabla h(x)$ over the feasible set of (4.66).

We are ready to show how a sufficient decrease in the objective function of (4.58) can be obtained by setting the estimated active variables to zero and properly updating one estimated non-active variable.

Proposition 19. *Let Assumption 3 hold. Given a feasible point x of problem* (4.58)*, assume that x is non-stationary. Let $j \in N_{\ell_1}(x) \cap J_{\ell_1}(x)$, $I = \{1, \ldots, n\} \setminus \{j\}$ and let $\hat{A}_{\ell_1}(x)$ be a set of indices such that*

$$\hat{A}_{\ell_1}(x) \subseteq A_{\ell_1}(x).$$

Let $\tilde{x}$ be the feasible point defined as follows:

$$\tilde{x}_i = \begin{cases} 0, & i \in \hat{A}_{\ell_1}(x), \\ x_i, & i \in I \setminus \hat{A}_{\ell_1}(x), \\ x_i = x_i - \operatorname{sgn}(\nabla_j h(x_j)) \displaystyle\sum_{h \in \hat{A}_{\ell_1}(x)} |x_h|, & i = j, \end{cases} \tag{4.87}$$

Then,

$$h(\tilde{x}) - h(x) \leq -2\tau^2 L \|\tilde{x} - x\|^2.$$

where L is the Lipschitz constant of $\nabla h(x)$ over the feasible set of (4.58).

Proof. First, we show that $\nabla f(\alpha)$ is Lipschitz continuous over the unit simplex with constant $2\tau^2 L$, where L is the Lipschitz constant of $\nabla h(x)$ over the feasible region of (4.58). To prove it, let $\bar{\alpha}$ and $\hat{\alpha}$ be two feasible points of (4.66), and let $\bar{x}$ and $\hat{x}$ be the points obtained by applying (4.68) with $\alpha = \bar{\alpha}$ and $\alpha = \hat{\alpha}$, respectively. Taking into account (4.72), we have that

$$\|\nabla f(\bar{\alpha}) - \nabla f(\hat{\alpha})\|^2 = \tau^2 \left\| \begin{bmatrix} \nabla h(\bar{x}) - \nabla h(\hat{x}) \\ \nabla h(\hat{x}) - \nabla h(\bar{x}) \end{bmatrix} \right\|^2 = 2\tau^2 \|\nabla h(\bar{x}) - \nabla h(\hat{x})\|^2$$

and then, exploiting the Lipschitz continuity of $\nabla h(x)$, we obtain

$$\|\nabla f(\bar{\alpha}) - \nabla f(\hat{\alpha})\| \leq \sqrt{2}\,\tau L \|\bar{x} - \hat{x}\|. \tag{4.88}$$

Using (4.68), we get

$$\|\bar{x} - \hat{x}\| \leq \|M\bar{\alpha} - M\hat{\alpha}\| \leq \|M\|\|\bar{\alpha} - \hat{\alpha}\|, \tag{4.89}$$

where the first inequality is due to the presence of the slack variable z in (4.68). Exploiting known properties of real matrices, we can write

$$\|M\|_2 \leq \sqrt{\left(\max_{1\leq j\leq 2n+1}\left\{\sum_{i=1}^{n+1}|M_{i,j}|\right\}\right)\left(\max_{1\leq i\leq n+1}\left\{\sum_{j=1}^{2n+1}|M_{i,j}|\right\}\right)} = \sqrt{2}\,\tau,$$

where $M_{i,j}$ is the entry of M in position (i, j). Consequently, we have that

$$\|\bar{x} - \hat{x}\| \leq \sqrt{2}\tau\|\bar{\alpha} - \hat{\alpha}\|. \tag{4.90}$$

From (4.88) and (4.90), we get

$$\|\nabla f(\bar{\alpha}) - \nabla f(\hat{\alpha})\| \leq 2\tau^2 L\|\bar{\alpha} - \hat{\alpha}\|,$$

that is, $2\tau^2 L$ is the Lipschitz constant of $\nabla f(\alpha)$ over the unit simplex.

Now, we show that the assertion is true. Let α be the point given by (4.69) and let us consider the sets $A(\alpha)$, $N(\alpha)$ and $J(\alpha)$. From (4.70), there exists $\hat{A}(\alpha) \subseteq A(\alpha)$ such that

$$i \in \hat{A}_{\ell_1}(x) \Leftrightarrow i, i+n \in \hat{A}(\alpha).$$

We distinguish three cases:

(i) $\nabla_j h(x) = 0$. From (4.72) and the definition of $J_{\ell_1}(x)$, it follows that $\nabla h(x) = 0$, thus contradicting the hypothesis that x is non-stationary. So, we can exclude this case from our analysis.

(ii) $\nabla_j h(x) < 0$. We first observe that, from (4.72), $j \neq 2n+1$. Moreover, using (4.72) and the definition of $J_{\ell_1}(x)$, we can write

$$\tau\nabla_j h(x) \leq \tau\min\{\nabla_i h(x), -\nabla_i h(x)\} < -\tau\nabla_j h(x), \quad \forall i = 1, \ldots, n.$$

Exploiting again (4.72), and recalling the definition of $J(\alpha)$, the above relation implies that

$$j \in J(\alpha).$$

So, we can compute the vector $\tilde{\alpha}$ as

$$\tilde{\alpha}_i = \begin{cases} 0, & i \in \hat{A}(\alpha), \\ \alpha_i, & i \in \{1, \ldots, 2n+1\} \setminus \{j\} \setminus \hat{A}(\alpha), \\ \alpha_i + \displaystyle\sum_{i\in\hat{A}(\alpha)} \alpha_i, & i = j. \end{cases}$$

Then, by applying (4.68), we obtain $\tilde{x}$ defined as in (4.87). In particular, let us observe that $\tilde{x}_j \geq x_j$ because $\tilde{\alpha}_j \geq \alpha_j$ and $\tilde{\alpha}_{j+n} = \alpha_{j+n}$.

Now, taking into account that $\nabla f(\alpha)$ is Lipschitz continuous over the unit simple with constant $2\tau^2 L$, the assumption we made on ϵ and the fact that problem (4.66) has $2n+1$ variables, then the assertion follows from Proposition 16.

(iii) $\nabla_j h(x) > 0$. We can repeat the same reasons made for the previous case, with the difference that now we obtain

$$j + n \in J(\alpha).$$

So, we can compute the vector $\tilde{\alpha}$ as

$$\tilde{\alpha}_i = \begin{cases} 0, & i \in \hat{A}(\alpha), \\ \alpha_i, & i \in \{1, \ldots, 2n+1\} \setminus \{j\} \setminus \hat{A}(\alpha), \\ \alpha_i + \sum_{i \in \hat{A}(\alpha)} \alpha_i, & i = j + n. \end{cases}$$

Applying (4.68), we obtain $\tilde{x}$ defined as in (4.87). In particular, we have that $\tilde{x}_j \leq x_j$, since $\tilde{\alpha}_j = \alpha_j$ and $\tilde{\alpha}_{j+n} \geq \alpha_{j+n}$.

□

From the previous results, we can define an algorithmic framework to solve problem (4.58), reported in Algorithm 17.

Algorithm 17 `Active-Set algorithmic framework for minimization over the` ℓ_1`-Ball (AS-`ℓ_1`-BALL)`

Choose a feasible point x^0
For $k = 0, 1, \ldots$
  If x^k is a stationary point, then STOP

  Active-Set Estimation:

  Compute $A^k_{\ell_1} = A_{\ell_1}(x^k)$ and $N^k_{\ell_1} = N_{\ell_1}(x^k)$

  Minimization step over $A^k_{\ell_1}$:

  Compute $J^k_{\ell_1} = J_{\ell_1}(x^k)$, choose $j \in N^k_{\ell_1} \cap J^k_{\ell_1}$ and define $\tilde{N}^k_{\ell_1} = N^k_{\ell_1} \setminus \{j\}$
  Set $\tilde{x}^k_{A^k} = 0$, $\tilde{x}^k_{\tilde{N}^k} = x^k_{\tilde{N}^k}$, $\tilde{x}^k_j = x^k_j - \text{sgn}(\nabla_j h(x^k_j)) \sum_{h \in A^k_{\ell_1}} |x^k_h|$

  Minimization step over $N^k_{\ell_1}$:

  Set $d^k_{A^k_{\ell_1}} = 0$
  Compute a feasible direction $d^k_{N^k_{\ell_1}}$ in $\tilde{x}^k$ and a maximum stepsize $\alpha^k_{\max}$
  If $\nabla h(\tilde{x}^k)^T d^k < 0$ then
    Compute a stepsize $\alpha^k \in (0, \alpha^k_{\max}]$ by means of the Armijo line search (Algorithm 11)
  Else
    Set $\alpha^k = 0$
  End if
  Set $x^{k+1} = \tilde{x}^k + \alpha^k d^k$
End for

Also in this case, for every point $\tilde{x}^k$ generated at Step 6 of Algorithm 17, the search direction d^k is computed in order to update only the estimated non-active variables, that is,

$$d^k_{A^k_{\ell_1}} = 0.$$

To compute $d^k_{N^k_{\ell_1}}$, we can employ either the standard Frank-Wolfe direction, or one of its variants. In particular, exploiting the relations between problem (4.58) and (4.66), we can easily compute, in the subspace $N^k_{\ell_1}$, every variant of the Frank-Wolfe direction that has been considered in Subsection 4.3.2.

For the sake of completeness, we describe how to compute such directions. At every iteration k, two feasible search directions $d_{N^k_{\ell_1}}$ can be computed (in the subspace N^k):

- the Frank-Wolfe direction

$$d^{\mathrm{FW}}_{N^k_{\ell_1}} = -\tau \operatorname{sgn}(\nabla_{\hat{\imath}} h(\tilde{x}^k))\, e_{\hat{\imath}} - \tilde{x}^k_{N^k_{\ell_1}}, \quad \hat{\imath} = \operatorname*{Argmax}_{i \in N^k_{\ell_1}} \{|\nabla_i h(\tilde{x}^k)|\};$$

- the Away-Step direction

$$d^{\mathrm{A}}_{N^k_{\ell_1}} = \tilde{x}^k_{N^k_{\ell_1}} - \tau \operatorname{sgn}(\nabla_{\hat{\jmath}} h(\tilde{x}^k))\, e_{\hat{\jmath}}, \quad \hat{\jmath} = \operatorname*{Argmax}_{j \in N^k_{\ell_1} :\, \tilde{x}^k_j > 0} \{|\nabla_j h(\tilde{x}^k)|\}.$$

The final search direction d^k can thus be computed according to one of the following three rules:

- (FW) rule: $d^k_{N^k}$ is chosen as the Frank-Wolfe direction, that is,

$$\begin{aligned} d^k_{A^k_{\ell_1}} &= 0, \\ d^k_{N^k_{\ell_1}} &= d^{\mathrm{FW}}_{N^k_{\ell_1}}. \end{aligned}$$

  In this case, we simply write

$$d^k = d^{\mathrm{FW}}.$$

- (AFW) rule: $d^k_{N^k}$ is chosen as the Away-Step Frank-Wolfe direction, that is,

$$\begin{aligned} d^k_{A^k_{\ell_1}} &= 0, \\ d^k_{N^k_{\ell_1}} &= d^{\mathrm{AFW}}_{N^k_{\ell_1}} = \begin{cases} d^{\mathrm{FW}}_{N^k_{\ell_1}}, & \text{if } \nabla_{N^k_{\ell_1}} h(\tilde{x}^k)^T d^{\mathrm{FW}}_{N^k_{\ell_1}} \le \nabla_{N^k_{\ell_1}} h(\tilde{x}^k)^T d^{\mathrm{A}}_{N^k_{\ell_1}}, \\ d^{\mathrm{A}}_{N^k_{\ell_1}}, & \text{otherwise.} \end{cases} \end{aligned}$$

  In this case, we simply write

$$d^k = d^{\mathrm{AFW}}.$$

- (PFW) rule: $d^k_{N^k}$ is chosen as the Pairwise Frank-Wolfe direction, that is,

$$\begin{aligned} d^k_{A^k_{\ell_1}} &= 0, \\ d^k_{N^k_{\ell_1}} &= d^{\mathrm{PFW}}_{N^k_{\ell_1}} = d^{\mathrm{FW}}_{N^k_{\ell_1}} + d^{\mathrm{A}}_{N^k_{\ell_1}}. \end{aligned}$$

  In this case, we simply write

$$d^k = d^{\mathrm{PFW}}.$$

Finally, exploiting the relations between problem (4.58) and problem (4.66), recalling Proposition 17 and 19, and taking into account how we compute d^k, we can state the convergence of `AS-`ℓ_1`-BALL` for every considered variant of the Frank-Wolfe direction, which follows from the convergence results of `AS-SIMPLEX`.

Theorem 17. *Let Assumption 3 hold and let $\{x^k\}$ be the sequence of points produced by `AS-`ℓ_1`-BALL`. Let us assume that the search direction d^k is computed according to one among (FW), (AFW) and (PFW) rule.*

Then, either an integer $\bar{k} \geq 0$ exists such that $\nabla h(x^{\bar{k}})^T(x - x^{\bar{k}}) \geq 0$ for all x: $\|x\|_1 \leq \tau$, or the sequence $\{x^k\}$ is infinite and every limit point x^* satisfies $\nabla h(x^*)^T(x - x^*) \geq 0$ for all x: $\|x\|_1 \leq \tau$.

4.6 Preliminary numerical results

In this section, we report some preliminary numerical results obtained by applying `AS-`ℓ_1`-BALL` on problems of the form:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \ & \frac{1}{2}\|Ax - b\|^2 \\ & \|x\|_1 \leq \tau, \end{aligned} \tag{4.91}$$

with $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and $\tau > 0$.

The testing problems were generating following the analysis suggested in [17, 74, 29]. In particular, several artificial signals were generated, with

- dimension $n \in \{2^{11}, 2^{12}, 2^{13}\}$;
- number of observations $m = n/4$;
- number of nonzeros $T = \mathrm{round}(\rho m)$, with $\rho \in \{0.01, 0.03, 0.05, 0.07, 0.1\}$.

Matrix A was obtained by generating a matrix with $m \times n$ independent and identically distributed elements from the Normal distribution $N(0, 1)$, and then normalizing the columns.

Once matrix A has been built, the "true" signal $x^* \in \mathbb{R}^n$ was generated as a vector with all components equal to 0, except for T randomly placed ± 1 spikes. Vector b was build as $Ax^* + \eta$, with η drawn from a normal distribution with mean 0 and variance 10^{-3}. Finally, we set $\tau = 1.1\|x^*\|_1$.

For each instance, we first run the Frank-Wolfe variants without using the active-set estimate, i.e., the Frank-Wolfe (`FW`), the Away-Step Frank-Wolfe (`AFW`) and the

Pairwise Frank-Wolfe (PFW) method. Then, we run the corresponding active-set versions.

In particular, considering the framework reported in Algorithm 17, we call AS-FW-ℓ_1-BALL, AS-AFW-ℓ_1-BALL and AS-PFW-ℓ_1-BALL the methods where the search direction d^k is computed according to (FW), (AFW) and (PFW) rule, respectively.

All the considered algorithms employed the origin as starting point and they were terminated at the fist iteration k satisfying

$$\nabla h(x^k)^T x \geq -10^{-9}, \quad \forall x \colon \|x\|_1 \leq \tau,$$

where h is the objective function of (4.91). Moreover, we arrested an algorithm when the number of iterations exceeded $200\,T$.

For what concerns the value of the parameter ϵ to use in the active-set estimate, we have the same issues discussed in Chapter 3 about the impossibility of knowing an acceptable value a priori. Then, starting from the value $\epsilon = 1$, at every iteration k we compute $\tilde{x}^k$ as indicated at Step 6 of Algorithm 17. If a sufficient decrease in the objective function is obtained, then we accept $\tilde{x}^k$ and we do not change the value of ϵ. Otherwise, we do not accept $\tilde{x}^k$, we reduce ϵ and we estimate the active-set again, continuing until we get a sufficient decrease in the objective function.

All the codes were implemented in Matlab R2014b and the experiments were run on an Intel Xeon(R), CPU E5-1650 v2 3.50 GHz. For every fixed n and ρ, the results have been averaged over 10 runs.

First, we analyze the effect of using the proposed active-set estimate in every Frank-Wolfe variant.

For what concerns FW, we actually do not observe significant differences when the active-set estimate is employed. Namely, AS-FW-ℓ_1-BALL and FW perform quite similarly.

Vice versa, for both AFW and PFW, the use of the active-set estimate leads to remarkable improvements. In particular, in Figure 4.1 we compare AS-AFW-ℓ_1-BALL with AFW, and in Figure 4.2 we compare AS-PFW-ℓ_1-BALL with PFW. In both figures, we plot the objective function versus the computational time. It is clear that the objective function decreases much faster when the active-set estimate is employed, for every considered dimension n and sparsity level ρ.

Finally, in Figure 4.3 we compare all the considered active-set variants, i.e., AS-FW-ℓ_1-BALL, AS-AFW-ℓ_1-BALL and AS-PFW-ℓ_1-BALL. Also in this figure, the objective function versus the computational time is plotted. We can easily see that both AS-AFW-ℓ_1-BALL and AS-PFW-ℓ_1-BALL significantly outperform AS-FW-ℓ_1-BALL. Moreover, AS-PFW-ℓ_1-BALL performs better than and AS-AFW-ℓ_1-BALL for all the considered problems.

4.7 Conclusions

We have presented an algorithmic framework to solve minimization problems over the unit simplex. The proposed approach is based on the use of an active-set estimate to identify those variables that are equal to zero at the stationary point. This technique enables us to perform two steps at each iteration: first, we set the estimated active variables to zero and we update one estimated non-active variable,

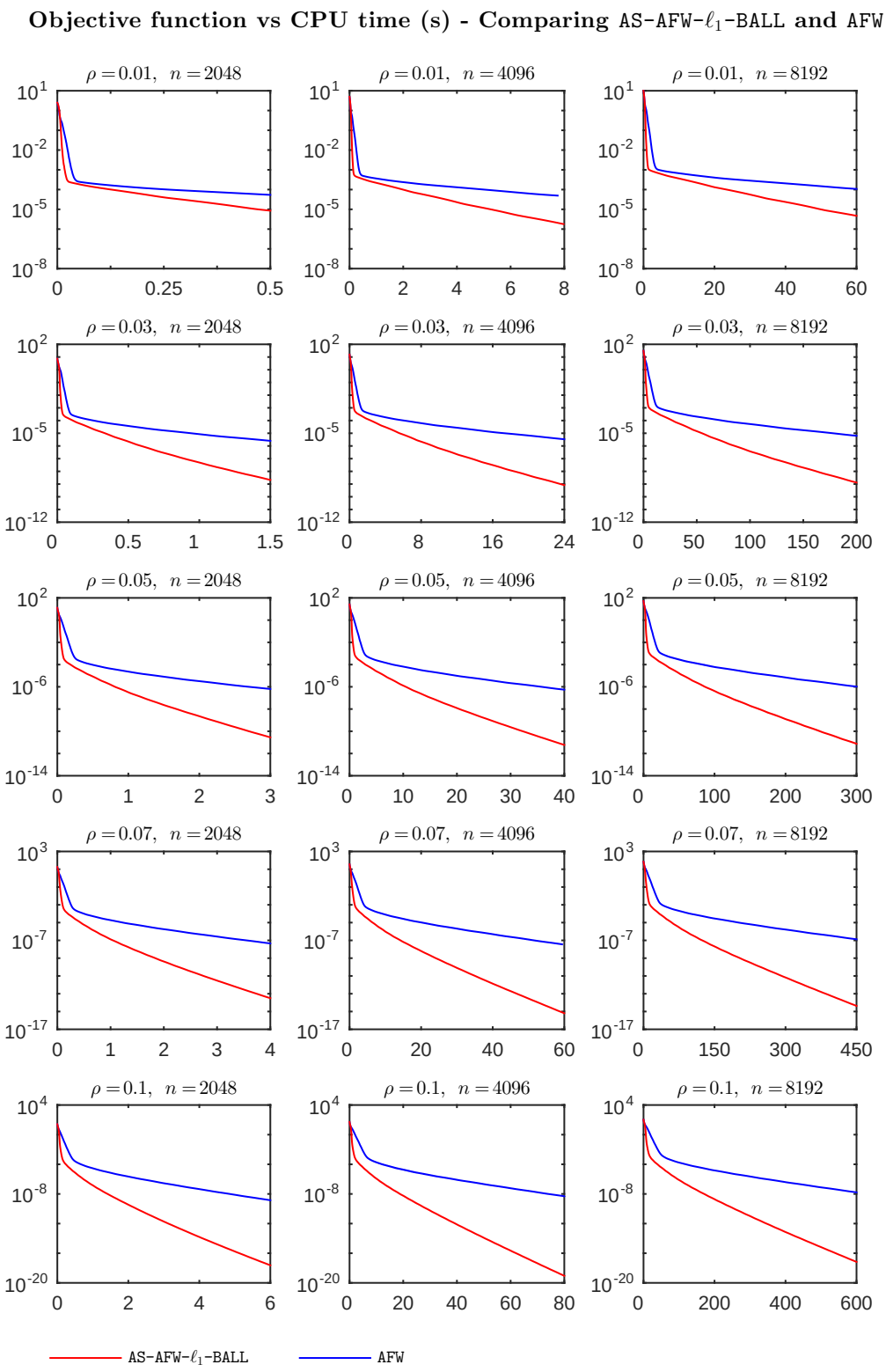

Figure 4.1. Objective function vs CPU time (in seconds). Comparison between AS-AFW-ℓ_1-BALL and AFW. The y axis is in logarithmic scale.

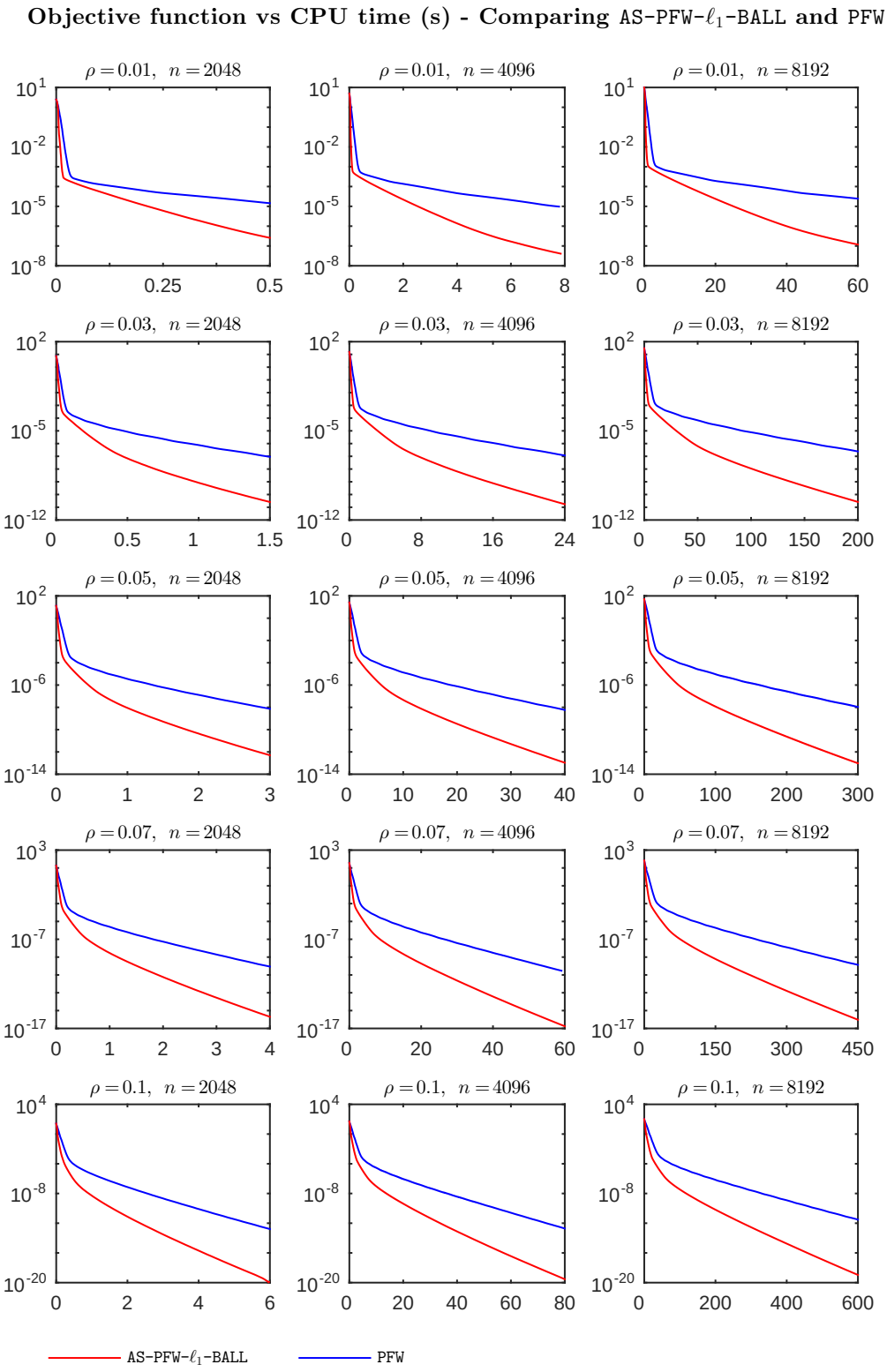

Figure 4.2. Objective function vs CPU time (in seconds). Comparison between AS-PFW-ℓ_1-BALL and PFW. The y axis is in logarithmic scale.

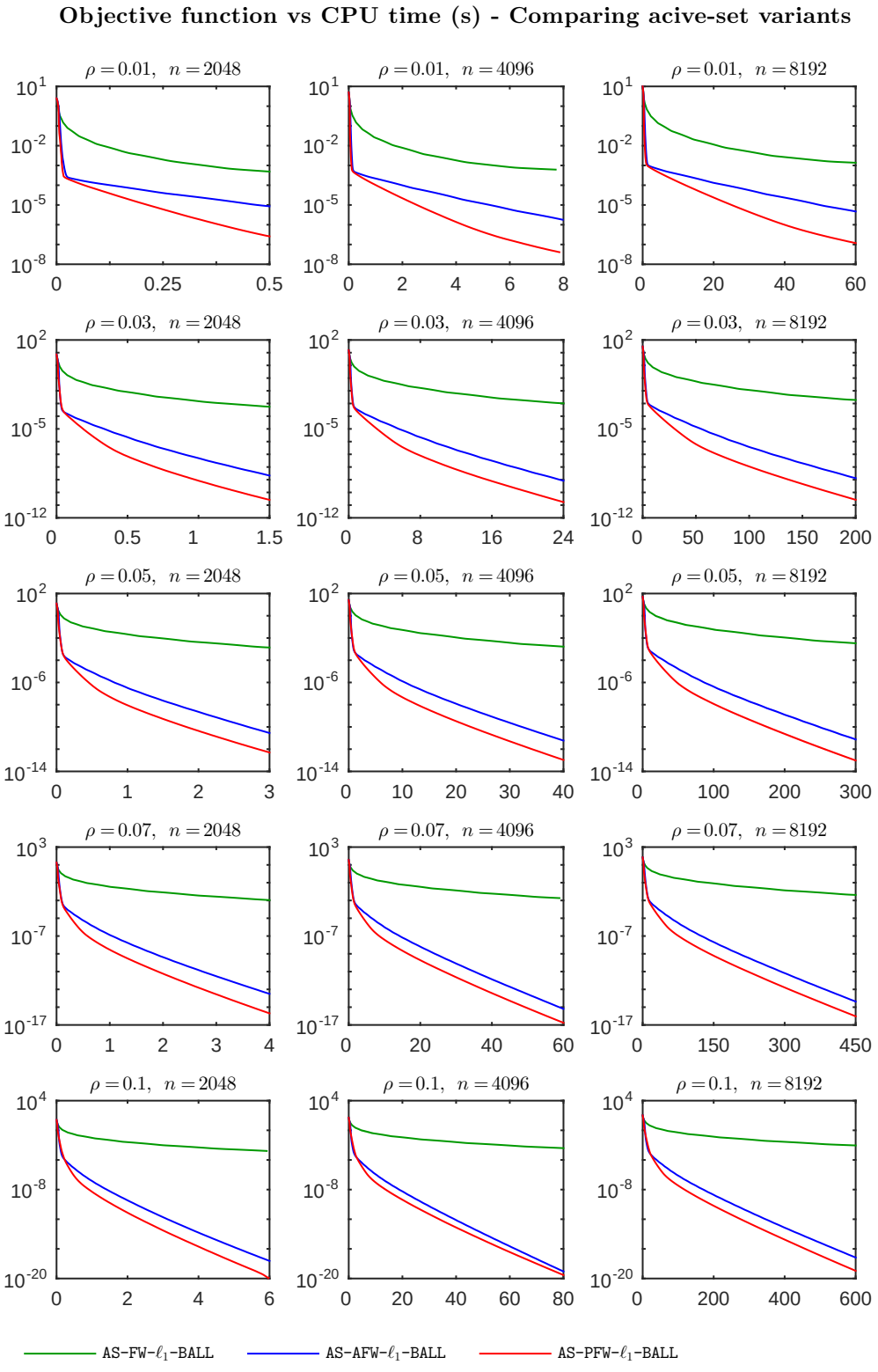

Figure 4.3. Objective function vs CPU time (in seconds). Comparison between `AS-FW-`ℓ_1`-BALL`, `AS-AFW-`ℓ_1`-BALL` and `AS-PFW-`ℓ_1`-BALL`. The y axis is in logarithmic scale.

so that a new feasible point is generated and a sufficient decrease in the objective function is obtained. Then, we move the estimated non-active variables along a suitable search direction. The convergence of the algorithm has been proved when different variants of the Frank-Wolfe direction are employed.

Finally, we have considered the problem of minimizing a function over the ℓ_1-ball. We have shown how our algorithmic framework can be adapted to this problem without any explicit variable transformation. Preliminary numerical result, obtained on some quadratic problems over the ℓ_1-ball, have shown the effectiveness of the proposed active-set strategy.

Chapter 5

Data filtering for cluster analysis by ℓ_0-norm regularization

Optimization algorithm are widely employed in the field of *machine learning*, usually with the aim of minimizing a specific loss function that measures the "error" between the output predicted by a model and its actual value.

Here, we consider the problem of *cluster analysis*, which consists in classifying a given set of unlabeled objects by similarity.

In particular, a data filtering method is proposed, based on minimizing a least squares function with a weighted ℓ_0-norm penalty, which is approximated with smooth parametric functions. After proving the convergence of the global minimum points of the approximating problems towards global minimum points of the original problem, a technique is described to choose a suitable value of the penalty parameter. Numerical results are finally provided.

The rest of the chapter is organized as follows. In Section 5.1, we introduce the problem and describe some approaches recently proposed in the literature. In Section 5.2, we present the ℓ_0-norm penalty clustering model and its smooth approximation, discussing some theoretical aspects. In Section 5.3, we present the data filtering method. In Section 5.4, we show the numerical results. Finally, in Section 5.5, we draw some conclusions.

5.1 Introduction

Cluster analysis is a branch of unsupervised learning, arising in many real-world applications and in different fields, e.g., biology, medicine, marketing, document retrieval, image segmentation and many others. It deals with grouping objects so that "alike" data are in the same clusters and "unlike" data are in different clusters. More formally, given a finite set of vectors $X = \{x_1, \dots, x_m\} \subset \mathbb{R}^n$, we want to divide X into k groups (clusters), according to a defined measure of similarity, where k can be either known or unknown.

Clustering models and algorithms can be classified on the basis of different features. For example, in *hard* clustering each object must belong to exactly one

cluster, whereas in *fuzzy* clustering each object can belong to more than one cluster with a certain degree. Another distinction can be done between hierarchical and partitioning clustering: while in the first case algorithms build a hierarchy of clusters (following either a top-down or a bottom-up approach) and return a structure called *dendrogram*, partitioning clustering methods group data as a result of an optimization problem. In partitioning clustering, further differences derive from the choice of the distance function and of the optimality criterion. Besides the mentioned approaches, other popular methods are based on cluster densities and on statistical models.

Partitioning X into a fixed number of clusters is known to be an NP-hard problem [32] and many existing clustering models are formulated as non-convex optimization problems. As a result, algorithms can generally find only approximate solutions. Moreover, there is no objectively "right" clustering model and the choice of the most suitable algorithm can strongly depend on the specific data set. So, there is still a great interest in developing new strategies for cluster analysis, also in the field of numerical optimization.

Here, we propose a data filtering method based on combining two different techniques.

The first one is a reformulation of the clustering problem as a penalized regression problem, proposed in [64, 45, 52], and further studied in [63, 10, 56]. Assuming that the number of clusters is unknown, this approach is based on introducing for each observation x_i a centroid $z_i \in \mathbb{R}^n$, representing the cluster which x_i belongs to. The problem consists in minimizing the distances between x_i and z_i, trying at the same time to group centroids. This is obtained by adding to the objective function a term to penalize each pair (i, j) such that $z_i \neq z_j$. The problem can be formulated as

$$\min_{z \in \mathbb{R}^{mn}} \sum_{i=1}^{m} \|x_i - z_i\|^2 + \lambda \sum_{j=2}^{m} \sum_{i=1}^{j-1} w_{ij} P(z_i - z_j), \tag{5.1}$$

where we indicate with z the vector $\begin{bmatrix} z_1^T & \dots & z_m^T \end{bmatrix}^T \in \mathbb{R}^{mn}$, λ is a nonnegative penalty parameter, w_{ij} are nonnegative fixed parameters, and $P\colon \mathbb{R}^n \to \mathbb{R}$ is a (symmetric) penalty function such that

$$P(y) \begin{cases} = 0, & \text{if } y = 0, \\ > 0, & \text{otherwise.} \end{cases}$$

The centroids provided by the solution $z^* = \begin{bmatrix} (z_1^*)^T & \dots & (z_m^*)^T \end{bmatrix}^T$ of (5.1) represent the final clusters. Namely, x_i and x_j are in the same cluster if $z_i^* = z_j^*$.

The basic idea behind model (5.1) is that a major number of centroids can be grouped simply by increasing the penalty parameter λ.

Anyway, when a fixed number of clusters is required, choosing a proper value of λ can be a very hard issue. In fact, by increasing λ, we can have a larger number of pairs of coinciding centroids in the optimal solution, that is, a larger number of pairs of points that belong to the same cluster. But this does not provide information on the number of clusters we obtain. Consequently, a value of λ that produces the desired number of clusters may not even exist.

Here, addressing the case in which a fixed number of clusters is required, we reinterpret model (5.1) as a method to map each sample x_i by a vector z_i that is representative of the local density of the samples in its neighborhood.

The proposed strategy also exploits a suitable technique to choose λ, based on minimizing a further optimality criterion that considers the distances within and between clusters.

As regards the penalty functions in (5.1), most authors focused on using convex ℓ_q-norms (e.g., the ℓ_1-norm, or the ℓ_2-norm), so that problem (5.1) is convex. In order to avoid the bias generated by convex penalties [27, 72], some non-convex ones were proposed in [63, 56]. On the other hand, the latter have the disadvantage not to make possible to reach the global minimum.

Here, we start from the following observation: since the penalty term in (5.1) has only the goal to force some pairs of centroids to coincide, then $P(z_i - z_j)$ should assume a constant value if $z_i \neq z_j$ (i.e., if x_i and x_j are in different clusters), regardless how far z_i and z_j are from each other. Furthermore, the penalty associated with each pair (i, j) should be weighted by taking into account the distance (i.e., the similarity) between the samples x_i and x_j, so that close pairs of points are encouraged to be in the same cluster.

Therefore, weighted ℓ_0-norm penalties are employed in this chapter. To overcome the non-continuity of the objective function, the ℓ_0-norm is then approximated with a sequence of smooth non-convex functions that converges to the ℓ_0-norm pointwise. As to be shown, the convergence of the global optimal solutions of the approximating problems towards global optimal solutions of the original problem can be proved.

The notation used in this chapter is that reported in Appendix A. Moreover, given $v \in \mathbb{R}^n$, we indicate with $(v)_h$ the h-th component of v

5.2 The model

In this section, we introduce the clustering model with ℓ_0-norm regularization and its smooth approximation, pointing out the relations between them. Since this is only the starting point for the proposed data filtering method, we do not address the issues concerning the choice of the penalty parameter, that will be discussed in Section 5.3.

5.2.1 The ℓ_0-regularized least squares problem

Let $X = \{x_1, \ldots, x_m\} \subset \mathbb{R}^n$ be a finite set of vectors, and let us consider problem (5.1). As discussed in the previous section, our goal is to employ a penalty function satisfying the following condition for each pair (i, j):

$$P(z_i - z_j) = \begin{cases} 0, & \text{if } z_i = z_j, \\ 1, & \text{otherwise,} \end{cases}$$

that is, $P(z_i - z_j)$ must not depend on the distance between the centroids z_i and z_j. We also want to weigh $P(z_i - z_j)$ by a parameter w_{ij} that takes into account the proximity of the samples x_i and x_j. In particular, w_{ij} should be large if the

samples x_i and x_j are near each other, so that close pairs of points are more strongly encouraged to be in the same cluster.

In other words, we want that the penalty value associated with each pair (i, j) depends on the distance between the samples x_i and x_j, but not on the distance between the centroids z_i and z_j. This leads to formulate the problem as follows:

$$\min_{z\in\mathbb{R}^{mn}} \sum_{i=1}^{m} \|x_i - z_i\|^2 + \lambda \sum_{j=2}^{m} \sum_{i=1}^{j-1} w_{ij}\, s(\|z_i - z_j\|), \tag{5.2}$$

where $s\colon \mathbb{R} \to \{0, 1\}$ is the step function defined as

$$s(u) = \begin{cases} 0, & \text{if } u = 0, \\ 1, & \text{otherwise,} \end{cases} \tag{5.3}$$

and w_{ij} are inversely proportional to the distance between x_i and x_j.

We observe that the penalty term can be seen as a weighted ℓ_0-norm of the vector with components $\|z_i - z_j\|$. Namely, we seek a solution z^* minimizing $\sum_{i=1}^{m} \|x_i - z_i\|^2$, such that the vector $\Big[\|z_i - z_j\|\Big]_{i<j}$ is sufficiently sparse.

Remark 5. *Problem* (5.2) *is well defined, in the sense that it attains a minimizer, since the objective function is lower semicontinuous and coercive [68].*

5.2.2 The smooth approximating problem

Minimizing a non-continuous function is hard, then it is reasonable trying to approximate (5.2) with a continuous and smooth problem.

Indicating with $\phi(z)$ the objective function of (5.2), we seek a smooth function $g(z; \alpha)$, depending on a parameter α, that converges to $\phi(z)$ pointwise. Namely, there must exist a sequence $\{\alpha^t\}$ such that

$$\lim_{t\to\infty} g(z; \alpha^t) = \phi(z), \quad \forall z \in \mathbb{R}^{mn}. \tag{5.4}$$

Roughly speaking, we expect that the minimum points of $g(z; \alpha^t)$ are "similar" to those of $\phi(z)$ for suitable values of the index t.

Many smooth approximations of the ℓ_0-norm were proposed in the literature. In particular, since the ℓ_0-norm of a vector is given by the sum of step functions, in [54, 7] the authors approximated the step function (5.3) with the following concave parametric function:

$$\beta(u; \alpha) = 1 - e^{-\alpha u}, \quad u \geq 0, \quad \alpha > 0. \tag{5.5}$$

This approach can be convenient when minimizing the ℓ_0-norm of a vector over a polyhedral set admitting a vertex. Exploiting the concavity of (5.5), it can be proved that there exists a finite index $\bar{t}$ such that, for every $t \geq \bar{t}$, the optimal solutions of the approximating problem also solve the original problem [67].

In our case, we are not interested in approximating (5.3) with a concave function, because the least squares term would make the approximating problem non-concave anyway. So, we slightly adapt the above described approach and we approximate

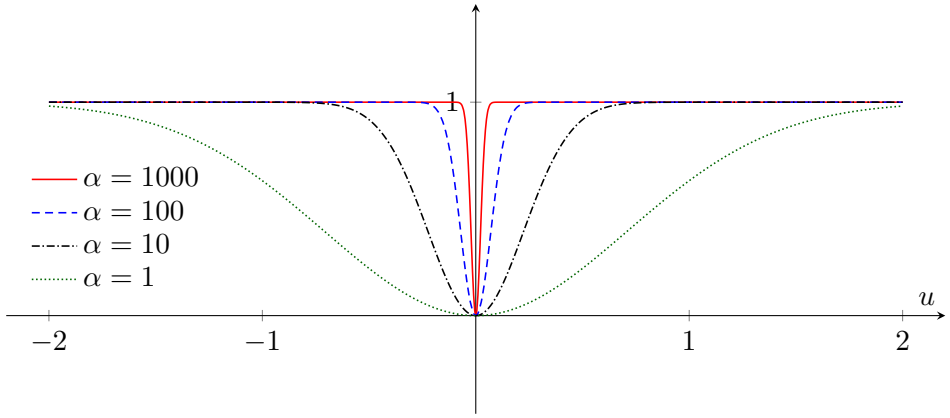

Figure 5.1. Function (5.6) approximating the step function, using different values of the parameter α.

$\sum_{j=2}^{m}\sum_{i=1}^{j-1} s(\|z_i - z_j\|)$ with the following smooth parametric function (depicted in Figure 5.1):

$$\gamma(z;\alpha) = \sum_{j=2}^{m}\sum_{i=1}^{j-1}\left(1 - e^{-\alpha\|z_i - z_j\|^2}\right), \quad \alpha > 0. \tag{5.6}$$

We finally write the problem approximating (5.2) as

$$\min_{z\in\mathbb{R}^{mn}} \sum_{i=1}^{m}\|x_i - z_i\|^2 + \lambda\sum_{j=2}^{m}\sum_{i=1}^{j-1} w_{ij}\left(1 - e^{-\alpha\|z_i - z_j\|^2}\right). \tag{5.7}$$

Indicating with $g(z;\alpha)$ the objective function of (5.7), it is straightforward to verify that (5.4) holds for every sequence $\{\alpha^t\}$ such that $\lim_{t\to\infty}\alpha^t = +\infty$. Then, we expect that the larger α is, the better (5.7) approximates (5.2).

Finally, let us remark that our approximation does not require slack variables and feasibility constraints.

5.2.3 Properties of the approximating problem

In this subsection, we investigate some theoretical properties of problem (5.7), pointing out the relations between its optimal solutions and those of (5.2). To this aim, we briefly recall the definition of the projection operator and we state some preliminary lemmas.

Definition 10. *Let $C \subseteq \mathbb{R}^n$ be a non-empty closed convex set. Given $x \in \mathbb{R}^n$, we call projection of x on C the unique solution $p(x)$ of the problem*

$$\min\{\|x - y\| : y \in C\}.$$

Lemma 7. *Let $C \subseteq \mathbb{R}^n$ be a non-empty closed convex set.*

- *For any $x \in \mathbb{R}^n$, $p(x)$ is the projection of x on C if and only if*
$$(x - p(x))^T (y - p(x)) \leq 0, \quad \forall y \in C. \tag{5.8}$$
- *For any $x, y \in \mathbb{R}^n$, let $p(x)$ and $p(y)$ be the projections of x and y on C, respectively. Then,*
$$\|p(x) - p(y)\| \leq \|x - y\|. \tag{5.9}$$

Proof. See [4][Proposition 2.1.3]. □

Lemma 8. *Let $C \subset \mathbb{R}^n$ be a non-empty closed convex set. Given $x \in C$, $y \in \mathbb{R}^n \setminus C$, let $p(y)$ be the projection of y on C. Then,*
$$\|x - (y + \xi(p(y) - y))\| < \|x - y\|, \quad \forall \xi \in (0, 1]. \tag{5.10}$$

Proof. Let $\tilde{y} = y + \xi(p(y) - y)$, where $\xi \in (0, 1]$. We can write:
$$x - y = (x - p(y)) + (p(y) - y),$$
$$x - \tilde{y} = (x - p(y)) + (p(y) - \tilde{y}) = (x - p(y)) + (1 - \xi)(p(y) - y).$$
From the above relations, it follows that
$$\|x - y\|^2 = \|x - p(y)\|^2 + \|p(y) - y\|^2 + 2(x - p(y))^T (p(y) - y),$$
$$\|x - \tilde{y}\|^2 = \|x - p(y)\|^2 + (1 - \xi)^2 \|p(y) - y\|^2 + 2(1 - \xi)(x - p(y))^T (p(y) - y).$$
Consequently,
$$\|x - y\|^2 - \|x - \tilde{y}\|^2 = (1 - (1 - \xi)^2)\|p(y) - y\|^2 + 2\xi(x - p(y))^T (p(y) - y).$$
Since $y \notin C$, $\xi \in (0, 1]$, and taking into account (5.8) of Lemma 7, we obtain that $\|x - y\|^2 - \|x - \tilde{y}\|^2 > 0$. □

Lemma 9. *Let $C \subseteq \mathbb{R}^n$ be a non-empty closed convex set. Given $x, y \in \mathbb{R}^n$, let $p(x)$ and $p(y)$ be the projections of x and y on C, respectively. Then,*
$$\|(x + \xi(p(x) - x)) - (y + \xi(p(y) - y))\| \leq \|x - y\|, \quad \forall \xi \in [0, 1]. \tag{5.11}$$

Proof. Let us consider the function
$$\omega\left(\begin{bmatrix} u \\ v \end{bmatrix}\right) = \|u - v\|,$$
where $u, v \in \mathbb{R}^n$. Since ω is convex in $\mathbb{R}^{2n}$, for all $\xi \in [0, 1]$ we can write
$$\begin{aligned}
\|(x + \xi(p(x) - x)) - (y + \xi(p(y) - y))\| &= \omega\left(\begin{bmatrix} x \\ y \end{bmatrix} + \xi \begin{bmatrix} p(x) - x \\ p(y) - y \end{bmatrix}\right) \\
&= \omega\left((1 - \xi)\begin{bmatrix} x \\ y \end{bmatrix} + \xi \begin{bmatrix} p(x) \\ p(y) \end{bmatrix}\right) \\
&\leq (1 - \xi)\,\omega\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) + \xi\,\omega\left(\begin{bmatrix} p(x) \\ p(y) \end{bmatrix}\right) \\
&= (1 - \xi)\|x - y\| + \xi\|p(x) - p(y)\| \\
&\leq (1 - \xi)\|x - y\| + \xi\|x - y\| \\
&= \|x - y\|,
\end{aligned}$$

where the last inequality follows from (5.9) of Lemma 7. □

Now, we can start analyzing some properties of problem (5.7). First, it attains optimal solutions, since the objective function is coercive. Moreover, the next proposition claims that all the local optimal solutions of (5.7) are contained in a compact set, which does not depend on λ and α.

Proposition 20. *Given a finite set of vectors $X = \{x_1, \dots, x_m\} \subset \mathbb{R}^n$, $\alpha > 0$, $\lambda \geq 0$, $w_{ij} \geq 0$, $j = 2, \dots, m$, $i = 1, \dots, j-1$, let $z^* = \begin{bmatrix}(z_1^*)^T & \dots & (z_m^*)^T\end{bmatrix}^T$ be a local optimal solution of (5.7). Then, $z_1^*, \dots, z_m^*$ are in the convex hull of X.*

Proof. Let $g(z; \alpha)$ be the objective function of problem (5.7) for every parameter $\alpha > 0$. Proceeding by contradiction, we assume that $z^* = \begin{bmatrix}(z_1^*)^T & \dots & (z_m^*)^T\end{bmatrix}^T$ is a local optimal solution of (5.7) and the following index subset is non-empty:

$$I = \{h \in \{1, \dots, m\} \colon z_h^* \notin \operatorname{conv}(X)\},$$

where $\operatorname{conv}(X)$ is the convex hull of $\{x_1, \dots, x_m\}$. Without loss of generality, we assume that

$$I = \{1, \dots, |I|\}.$$

Every vector $z \in \mathbb{R}^{mn}$ can be written as

$$z = \begin{bmatrix} z(I) \\ z(N) \end{bmatrix},$$

where

$$z(I) = \begin{bmatrix} z_1^T & \dots & z_{|I|}^T \end{bmatrix}^T \quad \text{and} \quad z(N) = \begin{bmatrix} z_{|I|+1}^T & \dots & z_m^T \end{bmatrix}^T.$$

So, in the following we indicate with $z^*(I)$ the vector $\begin{bmatrix}(z_1^*)^T & \dots & (z_{|I|}^*)^T\end{bmatrix}^T$ and with $z^*(N)$ the vector $\begin{bmatrix}(z_{|I|+1}^*)^T & \dots & (z_m^*)^T\end{bmatrix}^T$.
For each $i = 1, \dots, m$, we compute $p(z_i^*)$ as the projection of z_i^* onto $\operatorname{conv}(X)$. Now, we define the vector $\bar{d} = \begin{bmatrix}(\bar{d}_1)^T & \dots & (\bar{d}_m)^T\end{bmatrix}^T \in \mathbb{R}^{mn}$ such that

$$\bar{d}_i = p(z_i^*) - z_i^*, \quad i = 1, \dots, m.$$

From the definition of I, it follows that

$$\begin{aligned} \bar{d}_i &\neq 0, \quad i = 1, \dots, |I|, \\ \bar{d}_i &= 0, \quad i = |I| + 1, \dots, m, \end{aligned}$$

then $\bar{d} \neq 0$. The vector $\bar{d}$ can also be rewritten as

$$\bar{d} = \begin{bmatrix} \bar{d}(I) \\ \bar{d}(N) \end{bmatrix},$$

where

$$\bar{d}(I) = \begin{bmatrix} \bar{d}_1 \\ \vdots \\ \bar{d}_{|I|} \end{bmatrix} = \begin{bmatrix} p(z_1^*) - z_1^* \\ \vdots \\ p(z_{|I|}^*) - z_{|I|}^* \end{bmatrix},$$
$$\bar{d}(N) = \begin{bmatrix} \bar{d}_{|I|+1} \\ \vdots \\ \bar{d}_m \end{bmatrix} = \begin{bmatrix} p(z_{|I|+1}^*) - z_{|I|+1}^* \\ \vdots \\ p(z_m^*) - z_m^* \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}.$$

We show that $\bar{d}$ is a descent direction for $g(z;\alpha)$ at z^*, namely, that there exists a scalar $\bar{\xi} > 0$ such that

$$g(z^* + \xi\bar{d};\alpha) < g(z^*;\alpha), \quad \forall \xi \in (0, \bar{\xi}]. \tag{5.12}$$

To this aim, we rewrite $g(z;\alpha) = g_1\big(z(I)\big) + g_2\big(z(N)\big) + g_3(z;\alpha)$, where

$$g_1\big(z(I)\big) = \sum_{j=1}^{|I|} \|x_j - z_j\|^2,$$
$$g_2\big(z(N)\big) = \sum_{j=|I|+1}^{m} \|x_j - z_j\|^2,$$
$$g_3(z;\alpha) = \lambda \sum_{j=2}^{m} \sum_{i=1}^{j-1} w_{ij}\Big(1 - e^{-\alpha\|z_i - z_j\|^2}\Big).$$

We consider $g_1\big(z(I)\big)$, $g_2\big(z(N)\big)$ and $g_3(z;\alpha)$ separately.

- First, we consider $g_1\big(z(I)\big)$. From Lemma 8, for all $j \in I$ we can write

  $$\|x_j - (z_j^* + \xi\bar{d}_j)\|^2 < \|x_j - z_j^*\|^2, \quad \forall \xi \in (0, 1],$$

  and then

  $$g_1\big(z^*(I) + \xi\bar{d}(I)\big) < g_1\big(z^*(I)\big), \quad \forall \xi \in (0, 1]. \tag{5.13}$$

- Now, we consider $g_2\big(z(I)\big)$. Since $\bar{d}(N) = 0$, we simply have

  $$g_2\big(z^*(N) + \xi\bar{d}(N)\big) = g_2\big(z^*(N)\big), \quad \forall \xi > 0. \tag{5.14}$$

- Finally, we consider $g_3(z;\alpha)$. From Lemma 9, for all pairs (z_i^*, z_j^*) we can write

  $$\|(z_i^* + \xi\bar{d}_i) - (z_j^* + \xi\bar{d}_j)\|^2 \leq \|z_i^* - z_j^*\|^2, \quad \forall \xi \in (0, 1],$$

  from which we get

  $$1 - e^{-\alpha\|(z_i^* + \xi\bar{d}_i) - (z_j^* + \xi\bar{d}_j)\|^2} \leq 1 - e^{-\alpha\|z_i^* - z_j^*\|^2}, \quad \forall \xi \in (0, 1],$$

  and then

  $$g_3(z^* + \xi\bar{d};\alpha) \leq g_3(z^*;\alpha), \quad \forall \xi \in (0, 1]. \tag{5.15}$$

From (5.13), (5.14) and (5.15), we conclude that (5.12) holds with $\bar{\xi} = 1$. This contradicts the fact the z^* is a local optimal solution of (5.7). □

In the previous subsection, we pointed out that for large values of the parameter α, problem (5.7) is a good approximation of (5.2). The next theorem establishes the convergence of the global optimal solutions of problem (5.7) towards global optimal solutions of problem (5.2) for $\alpha \to +\infty$.

Theorem 18. *Given a finite set of vectors $X = \{x_1, \ldots, x_m\} \subset \mathbb{R}^n$, $\lambda \geq 0$, $w_{ij} \geq 0$, $j = 2, \ldots, m$, $i = 1, \ldots, j-1$, let $\{\alpha^t\}$ be a sequence of positive scalars such that $\alpha^{t+1} > \alpha^t$ and $\lim_{t\to\infty} \alpha^t = +\infty$. For any given parameter $\alpha \in \mathbb{R}^+$, let $g(z;\alpha)$ be the objective function of* (5.7)*, and $z(\alpha) = \begin{bmatrix} z_1(\alpha)^T & \ldots & z_m(\alpha)^T \end{bmatrix}^T$ be a global optimal solution of* (5.7)*. Then,*

(i) the sequence $\{g(z(\alpha^t);\alpha^t)\}$ converges,

(ii) the sequence $\{z(\alpha^t)\}$ attains limit points,

(iii) every limit point of $\{z(\alpha^t)\}$ is a global optimal solution of (5.2)*.*

Proof. Let $\phi(z)$ be the objective function of problem (5.2). Moreover, we indicate with $z^* = \begin{bmatrix} (z_1^*)^T & \ldots & (z_m^*)^T \end{bmatrix}^T$ a global optimal solution of problem (5.2).

From Proposition 20, it follows that the sequence $\{z(\alpha^t)\}$ remains in a compact set, thus it attains limit points, which proves (ii).

Now we show that, for all $t = 1, 2, \ldots$, the following relations hold:

$$g(z;\alpha^t) \leq g(z;\alpha^{t+1}) \leq \phi(z), \quad \forall z \in \mathbb{R}^{mn}, \tag{5.16}$$

$$g(z(\alpha^t);\alpha^t) \leq g(z^*;\alpha^t) \leq \phi(z^*), \tag{5.17}$$

$$g(z(\alpha^t);\alpha^t) \leq g(z(\alpha^{t+1});\alpha^{t+1}). \tag{5.18}$$

Relation (5.16) follows from the fact that $\alpha^{t+1} > \alpha^t > 0$. In fact, for every index pair (i, j), we have

$$1 - e^{-\alpha^t \|z_i - z_j\|^2} \leq 1 - e^{-\alpha^{t+1}\|z_i - z_j\|^2} \leq s(\|z_i - z_j\|).$$

The first inequality of (5.17) follows from the fact that $z(\alpha^t)$ minimizes $g(z;\alpha^t)$ with respect to z. The second inequality of (5.17) follows from (5.16). To prove (5.18), assume by contradiction that it does not hold. Then there exists an index t such that $g(z(\alpha^{t+1});\alpha^{t+1}) < g(z(\alpha^t);\alpha^t)$. Using (5.16), we can write

$$g(z(\alpha^{t+1});\alpha^t) \leq g(z(\alpha^{t+1});\alpha^{t+1}) < g(z(\alpha^t);\alpha^t),$$

which contradicts the fact that $z(\alpha^t)$ minimizes $g(z;\alpha^t)$ with respect to z. Then, (5.18) must hold.

From (5.17) and (5.18), it follows that the sequence $\{g(z(\alpha^t);\alpha^t)\}$ is monotonically non-decreasing and bounded from above. Thus it converges, proving (i).

Now, let $\bar{z}$ be a limit point of $\{z(\alpha^t)\}$, that is, there exists a subsequence $\{z(\alpha^t)\}_{\mathcal{T}}$ such that

$$\lim_{t\to\infty,\, t\in\mathcal{T}} z(\alpha^t) = \bar{z}. \tag{5.19}$$

To prove (iii), we assume by contradiction that $\bar{z}$ is not a global optimal solution of (5.2). Then, there exists $\epsilon > 0$ such that

$$\phi(z^*) \leq \phi(\bar{z}) - \epsilon. \tag{5.20}$$

Since $\lim_{t\to\infty} g(z;\alpha^t) = \phi(z)$ for all $z \in \mathbb{R}^{mn}$, there exists an index $\bar{t}$ such that

$$\left|\phi(\bar{z}) - g(\bar{z};\alpha^t)\right| < \epsilon, \qquad \forall t \geq \bar{t}. \tag{5.21}$$

Using (5.20) and (5.21), we have that $g(\bar{z};\alpha^t) > \phi(\bar{z}) - \epsilon \geq \phi(z^*)$, for all $t \geq \bar{t}$. Since $g(z;\alpha^t)$ is continuous with respect to z, there exists $\bar{\rho} > 0$ such that

$$g\Big(z;\alpha^{\bar{t}}\Big) > \phi(z^*), \qquad \forall z \in \mathcal{B}(\bar{z},\bar{\rho}). \tag{5.22}$$

From (5.16), (5.17) and (5.22), we can write

$$g(z;\alpha^t) \geq g\Big(z;\alpha^{\bar{t}}\Big) > \phi(z^*) \geq g(z^*;\alpha^t), \qquad \forall z \in \mathcal{B}(\bar{z},\bar{\rho}), \quad \forall t \geq \bar{t}. \tag{5.23}$$

From (5.19), there exists an index $\hat{t} \geq \bar{t}$ such that

$$z(\alpha^t) \in \mathcal{B}(\bar{z},\bar{\rho}), \qquad \forall t \geq \hat{t} \geq \bar{t}, \quad t \in \mathcal{T}. \tag{5.24}$$

Finally, from (5.23) and (5.24) we get

$$g(z(\alpha^t);\alpha^t) > g(z^*;\alpha^t), \qquad \forall t \geq \hat{t}, \quad t \in \mathcal{T},$$

which contradicts the fact that $z(\alpha^t)$ minimizes $g(z;\alpha^t)$ with respect to z for sufficiently large t. This proves (iii). □

5.3 The data filtering method

Assuming that a fixed number of clusters is required, in this section we present a data filtering strategy that combines model (5.7) with a technique to select a suitable value of the penalty parameter λ.

In particular, let $X = \{x_1, \ldots, x_m\} \subset \mathbb{R}^n$ be a finite set of vectors and assume that X must be partitioned into k clusters, with k fixed. Let $\mathcal{A}$ be a generic clustering algorithm. Our goal is to filter data to improve the performances of $\mathcal{A}$.

As discussed above, for any λ, an approximate solution $z_1^*, \ldots, z_m^*$ of (5.2) can be computed by solving (5.7) with suitable values of α. We observe that, independently of the obtained number of clusters, the points $z_1^*, \ldots, z_m^*$ provide important information, because they are grouped on the basis of local densities of the samples $x_1, \ldots, x_m$. Therefore, each z_i^* is representative of the behavior of X in the neighborhood of x_i. Consequently, after solving (5.7), a partition P of X can be computed by applying $\mathcal{A}$ to the points $z_1^*, \ldots, z_m^*$, instead of $x_1, \ldots, x_m$. Furthermore, some centroids should coincide and then the geometry of $z_1^*, \ldots, z_m^*$ is expected to be more regular than that of $x_1, \ldots, x_m$. This can make the vectors $z_1^*, \ldots, z_m^*$ easier to be clustered than $x_1, \ldots, x_m$.

In other words, the (approximated) ℓ_0-norm penalty model can be seen as a filtering method that maps each sample x_i by a vector z_i^* which is representative of the local density of X in the neighborhood of x_i.

Naturally, the solutions of problem (5.7) are sensitive to the value of λ, that is, different filters can be obtained by varying that parameter. So, a strategy to choose a proper value of λ must be introduced.

To this aim, we also have to take into account that the most suitable way to filter data can depend on the algorithm we apply later. This is why we introduce a criterion to evaluate the partitions produced by $\mathcal{A}$ after filtering data with a certain λ. The idea is to try different values of λ and finally choose the best one in terms of our criterion, as usually done in cross validation. The whole filtering method is then summarized in **Algorithm 18**:

1. Given $\{x_1, \ldots, x_m\}$, $1 \leq k \leq m$, $\{\lambda_1, \ldots, \lambda_N\} \in \mathbb{R}_0^+$, and an algorithm $\mathcal{A}$
2. For $t = 1, \ldots, N$
3. Set $\lambda = \lambda_t$ and compute $\{z_1^*, \ldots, z_m^*\}$ by solving problem (5.7)
4. Compute a partition $\tilde{P}^t = \{\tilde{C}_1, \ldots, \tilde{C}_k\}$ of $\{z_1^*, \ldots, z_m^*\}$ by algorithm $\mathcal{A}$
5. Let $P^t = \{C_1, \ldots, C_k\}$ such that $x_u \in C_i$ if $z_u^* \in \tilde{C}_i$, $u = 1, \ldots, m$, $i = 1, \ldots, k$
6. Evaluate P^t by assigning it a value $c(P^t)$
7. End for
8. Select P^* as the best among $\{P^1, \ldots, P^N\}$ in terms of $c(P^t)$

We remark that the above strategy allows adapting the filtering to the specific clustering algorithm $\mathcal{A}$. Namely, different filters can be obtained for the same data set, according to the clustering algorithm to apply later.

We conclude this section by explaining how we compute $c(P^t)$ at Step 6. Although the most proper way to evaluate a partition can strongly depend on the features of the specific data set (not known a priori), the criterion we propose comes from a natural interpretation of clusters as subsets of similar points, where similarity is measured by the distance. Basically, we encourage partitions with small distances within clusters and large distances between clusters.

More formally, given a partition $P^t = \{C_1, \ldots, C_k\}$, where $C_1, \ldots, C_k$ are disjoint subsets of X, we compute $c(P^t)$ at Step 6 as

$$c(P^t) = \frac{1}{d_b} \sum_{i=1}^{k} \frac{d_w(i)}{n_p(i)},$$

where $d_w(i)$ is the sum of the distances within cluster C_i, $n_p(i)$ is the number of pairs of points belonging to cluster C_i and d_b is the sum of the distances between all the pairs of points belonging to different clusters.

In order to operate in high-dimensional spaces, we also use *kernel functions* to compute distances between points. To this aim, we provide the following definition of kernel function, according to [69].

Definition 11. *Let $\mathcal{X}$ be some set. A kernel function is a symmetric function $K\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that*

$$K(x_u, x_v) = \langle \varphi(x_u), \varphi(x_v) \rangle, \quad x_u, x_v \in \mathcal{X},$$

where $\varphi\colon \mathcal{X} \to \mathcal{H}$, and $\mathcal{H}$ is a dot product space. Moreover, $\mathcal{H}$ is called feature space.

Then, the distance between two vectors $x_u, x_v \in R^n$ can be computed as

$$K(x_u, x_u) - 2K(x_u, x_v) + K(x_v, x_v),$$

where $K(\cdot, \cdot)$ is the chosen kernel function. In particular, in our simulations we used a Gaussian kernel. Given $x_u, x_v \in \mathbb{R}^n$, the Gaussian kernel is defined as

$$K(x_u, x_v) = e^{-\gamma \|x_u - x_v\|^2}, \quad x_u, x_v \in \mathbb{R}^n. \tag{5.25}$$

In our experiments, we set $\gamma = 0.1$.

5.4 Numerical experience

In this section, we report our numerical experience. In Subsection 5.4.1, we describe how we set up the experiments. In Subsection 5.4.2, we show how we solved problem (5.7). Finally, in Subsection 5.4.3, we report and discuss the numerical results.

5.4.1 Experimental set-up

First, we compare the performances of three well-known clustering methods when they are applied to the original data and when they are applied to the data filtered by Algorithm 18. The considered methods are the following:

- Single-Linkage (SL) method, which is a hierarchical clustering algorithm that iteratively merges the two clusters containing the closest pair of points (see [31] for further details);
- Expectation-Maximization for Gaussian mixture Models (EMGM), which tries to estimate the parameters of the probability density functions generating the samples (see [19, 57] for further details);
- Kernel K-Means (KKM), which is an extension of the standard k-means method exploiting kernel functions to compute distances (see [70, 33, 21] for further details).

Since KKM and EMGM aim to solve non-convex optimization problems, they were executed 1000 times, choosing randomly the starting parameters, and finally taking the solution providing the best objective value. In particular, to run KKM, we used a Gaussian kernel, defined as in (5.25), with $\gamma = 0.1$.

In addition, to show the effect of the ℓ_0-norm penalty, we also tried to filter data using a different regularization. In particular, we tested the squared ℓ_2-norm

regularization (also known as ridge regularization), which typically does not induce sparsity. In this case, we applied Algorithm 18 replacing problem (5.7) at Step 3 with the following problem:

$$\min_{z\in\mathbb{R}^{mn}} \sum_{i=1}^{m}\|x_i - z_i\|^2 + \lambda \sum_{j=2}^{m}\sum_{i=1}^{j-1} w_{ij}\|z_i - z_j\|^2. \tag{5.26}$$

For both filters (i.e., the one obtained with the ℓ_0-norm regularization and the one obtained with the squared ℓ_2-norm regularization), 150 increasing values of λ were used, chosen such that $\lambda_1 = 0$ and λ_{150} provides a solution of problem (5.7) (respectively, problem (5.26)) that collapses to a single centroid.

Moreover, for both problem (5.7) and problem (5.26), the weight parameters w_{ij} were set as

$$w_{ij} = e^{-0.1\|x_i - x_j\|^2}, \quad j = 2,\dots,m, \quad i = 1,\dots,j-1.$$

The experiments were conducted on some synthetic and real data sets, covering different scenarios[1]:

Case (i): two spherical clusters in two dimensions, with equal volumes and the same cardinality (Figure 5.2(a)). The first cluster has 50 points, generated from a bivariate Normal distribution with mean vector $\begin{pmatrix}0 & 0\end{pmatrix}^T$ and covariance matrix $0.33^2 I$. The second cluster has 50 points, drawn from a bivariate Normal distribution $N\Big(\begin{pmatrix}1 & 1\end{pmatrix}^T, 0.33^2 I\Big)$.

Case (ii): two elongated clusters in two dimensions, with different cardinalities (Figure 5.2(b)). The first cluster has 500 points, generated from a bivariate Normal distribution with mean vector $\begin{pmatrix}0 & 5\end{pmatrix}^T$ and covariance matrix $\begin{pmatrix}0.05 & 0\\ 0 & 5\end{pmatrix}$. The second cluster has 50 points, drawn from a bivariate Normal distribution $N\Big(\begin{pmatrix}2.5\\ 0\end{pmatrix}, \begin{pmatrix}0.3 & 0\\ 0 & 0.05\end{pmatrix}\Big)$.

Case (iii): two spherical clusters in two dimensions, with different volumes and cardinalities (Figure 5.2(c)). The first cluster has 500 points, generated from a bivariate Normal distribution $N\Big(\begin{pmatrix}0 & 0\end{pmatrix}^T, 4I\Big)$. The second cluster has 50 points, generated from a bivariate Normal distribution $N\Big(\begin{pmatrix}7 & 0\end{pmatrix}^T, 0.5I\Big)$.

Case (iv): four clusters in three dimensions. The centers $\mu_1, \mu_2, \mu_3, \mu_4 \in \mathbb{R}^3$ were drawn from a multivariate distribution $N\Big(\begin{pmatrix}0 & 0 & 0\end{pmatrix}^T, 5I\Big)$. When generating the centers, if two of them had an Euclidean distance smaller than 1, the simulation was aborted and then started again. After fixing the centers, the number of elements for each cluster was randomly chosen in the range $[10, 100]$.

[1] All data were scaled in $[-1, 1]$.

Finally, for each cluster i, the points were generated from a multivariate distribution $N(\mu_i, I)$. This is similar to case IV in [63], and scenario (c) in [73], but here clusters are more imbalanced.

Case (v): the Ecoli data set from the UCI repository [49]. There are 336 samples characterized by 7 features and divided into 8 clusters, which contain 143, 77, 52, 35, 20, 5, 2 and 2 elements, respectively.

Case (vi): the Fisher's Iris data from the UCI repository [49]. The points are in four dimensions and divided into 3 clusters of 50 elements each. The second and the third cluster are partially overlapped, whereas the first cluster is linearly separable from the other two.

Case (vii): the wine data set from the UCI repository [49], with 178 samples of 3 kinds of wine. The clusters contain 59, 71 and 48 elements, respectively, and each sample is characterized by 13 features.

Case (viii): the Wisconsin breast cancer data set from the UCI repository [49, 55]. There are 683 samples of 9 features each[2], divided in two groups: 444 benign and 239 malignant.

The partitions are finally evaluated by the Adjusted Rand Index (ARI) [46], which takes 1 as maximum value (ARI can also assume negative values).

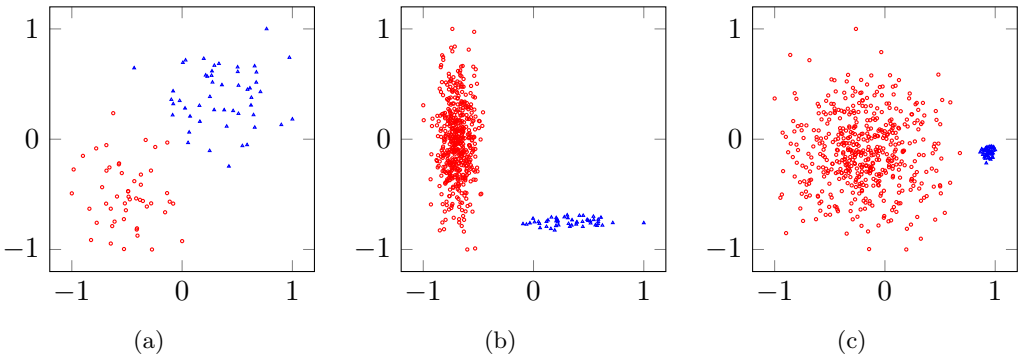

Figure 5.2. Data sets for case (i) (a), case (ii) (b) and case (iii) (c). Data were scaled in $[-1, 1]$.

5.4.2 Solving the approximating problem

Taking into account Theorem 18, solving (5.7) with large values of α can be a practical solution to get a good approximation of the optimal solutions of (5.2). Theorem 18 would also require to compute a global solution of the approximating problem, so, a global algorithm should be used, to be in line with the theory. But global algorithms are in general computationally expensive, especially when dealing with large-scale problems, as in our case. Moreover, since model (5.7) is employed

[2]Originally, there were 699 samples, but 16 of them had missing values and were removed.

to filter data, then (i) solving the problem should not be too expensive, and (ii) it could be sufficient to compute "good" solutions of problem (5.7), even if not global optima. Thus, it can be reasonable to employ a local algorithm that, on the one hand, can provide non-global minimizers, but, on the other hand, is cheaper than a global method.

After all, many clustering models are formulated as non-convex problems, and several algorithms that are widely used in practice are based on local strategies (e.g., the aforementioned KKM and EMGM). Anyway, defining efficient global methods to solve (5.7) can be a challenging task for future research.

For the above reasons, we solved problem (5.7) by employing a non-monotone version of the truncated-Newton method which exploits negative curvature directions (so, it is well suited for non-convex problems), proposed in [28].

Finally, another computational issue is that problem (5.2) becomes ill-conditioned when α and λ get large. Then, we employed a warm start strategy, gradually increasing α up to a prefixed value (this approach was also proposed in [7], but not attempted in practice). In particular, starting with $\alpha^t = 1$, $t = 1$, we employed the following updating rule: $\alpha^{t+1} = \min\{10^3, (1 + e^{-0.07t})\alpha^t\}$, $t = t + 1$, stopping the algorithm when α^t reaches 10^3. For every α^t, we solved the problem with a growing precision, terminating the minimization when the sup-norm of the gradient of the objective function was less than or equal to $\epsilon^t = \max\{10^{-5}, 10^{-2}/\alpha^t\}$.

5.4.3 Results

The final results are summarized in Table 5.1. The filter based on the ℓ_0-norm regularization and the one based on the squared ℓ_2-norm regularization are indicated as ℓ_0 *filter* and *ridge filter*, respectively.

For the ridge filter, problem (5.26) was solved by employing the truncated-Newton method reported in [28], terminating the algorithm when the sup-norm of the gradient of the objective function was less than or equal to 10^{-5}.

All computations were run on an Intel(R) Core(TM) i7-3770 CPU 3.40 GHz and the codes were implemented in Fortran 90.

First, let us discuss the results achieved by the ℓ_0 filter. Overall, the performances of the considered clustering methods improve by using this data filtering process.

In particular, in six data sets, the results obtained by SL are unsatisfactory by applying the algorithm to the original data, whereas performances remarkably increase when data are filtered. Only for the wine data set (case (vii)), the filtering does not lead to better results.

As regards EMGM, the data filtering strategy allows to improve the performances on all the real data sets (case (v)–(viii)). Only for case (iv), better partitions are obtained by applying the algorithm to the original data.

Also for KKM, the better partitions are those computed on the filtered data, except for case (vii) (and excluding case (i), where the right clusters are recognized also without filtering). A significant result is obtained for case (iii), where the clusters to detect have remarkably different volumes. This is known to be a hard case for centroid-based methods, but that issue has been overcome by the filtering strategy.

Table 5.1. Comparison between the values of the Adjusted Rand Index obtained by applying Single Linkage (SL), Expectation-Maximization for Gaussian mixture Models (EMGM) and Kernel K-Means (KKM) on the original data and on the filtered data. Two different filters are considered: the ℓ_0 filter is the one reported in Algorithm 18, whereas the ridge filter differs from the previous one in that problem (5.7) is replaced with problem (5.26) at Step 3 of Algorithm 18.

Method	**Dataset**							
	(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)	(viii)
SL	0.0000	1.0000	-0.0032	0.0057	0.0399	0.5584	-0.0038	0.0025
ℓ_0 filter + SL	1.0000	1.0000	1.0000	0.4022	0.4155	0.5657	-0.0068	0.8685
ridge filter + SL	0.0008	1.0000	-0.0032	0.0057	0.0399	0.5584	-0.0038	0.0025
EMGM	0.9600	1.0000	1.0000	0.5930	0.5843	0.4414	0.4778	0.5547
ℓ_0 filter + EMGM	1.0000	1.0000	1.0000	0.5697	0.6752	0.5657	0.7032	0.8798
ridge filter + EMGM	0.9600	1.0000	1.0000	0.5841	0.5768	0.9039	0.8154	0.8798
KKM	1.0000	0.9741	0.3977	0.4894	0.4538	0.7163	0.8992	0.8686
ℓ_0 filter + KKM	1.0000	1.0000	1.0000	0.6865	0.6977	0.7445	0.8820	0.8742
ridge filter + KKM	1.0000	0.9741	0.3977	0.4894	0.4730	0.7302	0.8992	0.8686

For what concerns the computational time, we plot in Figure 5.3 the CPU time (in seconds) needed to solve problem (5.7) versus the value of the penalty parameter λ.

We observe that each minimization required less than 3 seconds for case (i), (iv), (vi) and (vii). For case (ii), (iii) and (v), every minimizations took less than 15 seconds, except for a single value of λ in case (ii), which required 385 seconds.

As regards the largest data set considered in the experiments, i.e., case (viii), the minimizations took between 40 and 60 seconds for three values of λ. For the remaining values of λ, every minimization required less than 30 seconds. Overall, the average time needed to solve problem (5.7) is about 15 seconds.

Recalling that we solved (5.7) with a warm start strategy (by employing increasing values of α and solving the problem with a growing precision), it is also interesting to analyze the amount of time needed in the minimization procedure for every value of α. We report these times (in seconds) in Figure 5.4.

In particular, for every considered α, in Figure 5.4 is depicted the average time over the 150 considered values of the penalty parameter λ, needed to solve (5.7) with the related precision. In almost all data sets, the computational time increases when α becomes large, as expected. Only for case (viii), we have that small values of α required more time.

However, the computational time needed to solve (5.7) remains, on average, below 5 seconds for every considered α.

Now, we discuss the results obtained by applying the ridge filter. On the one hand, a clear advantage of using this filter is the low computational time needed to solve the optimization problem. In particular, less than 0.25 seconds were required to solve (5.26), for every data set and for every value of λ. Furthermore, problem (5.26) is smooth and convex, and then a global optimal solution can be

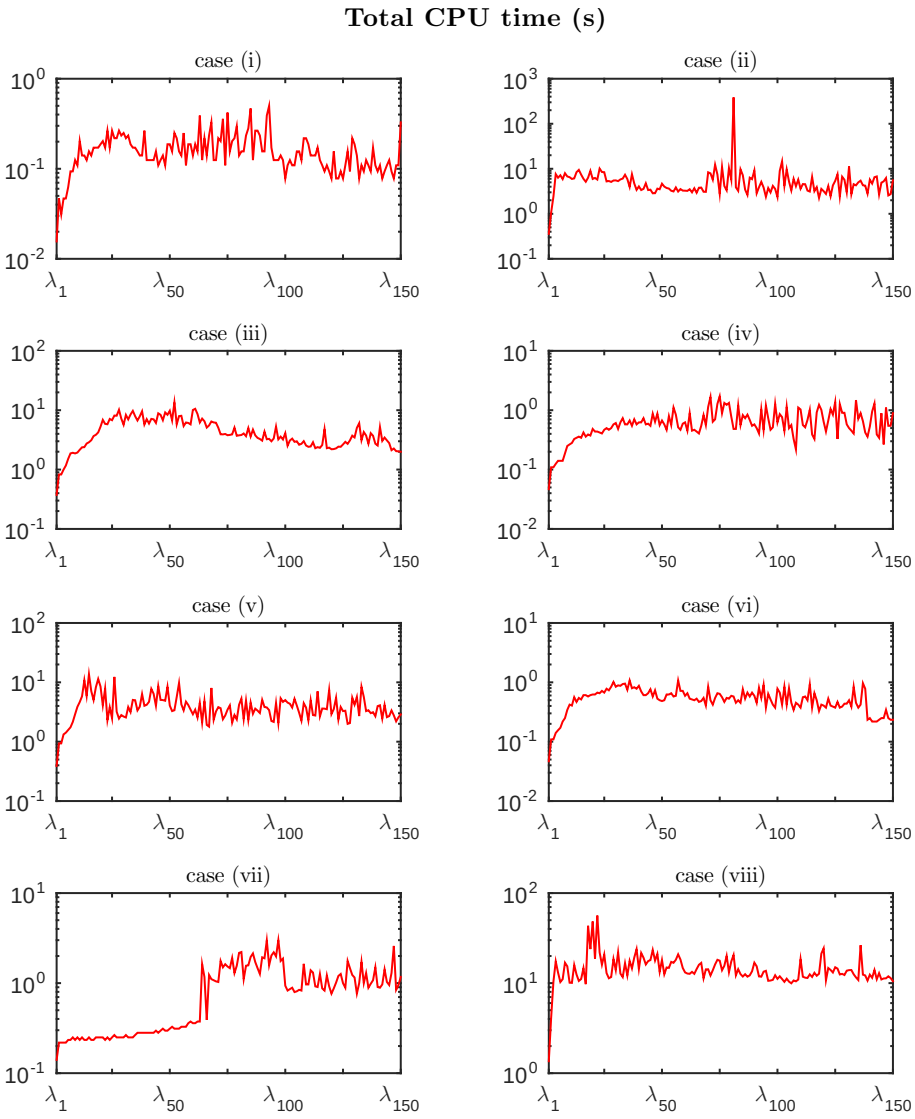

Figure 5.3. CPU time in seconds needed to solve problem (5.7) versus the value of λ. The y axis is in logarithmic scale.

computed by a local algorithm.

On the other hand, the numerical results seem worse than those achieved by the ℓ_0-norm regularization.

In particular, the ridge filter has essentially no effect on SL. Similarly, it does not provide relevant effects on KKM either. Looking at the results more in detail, we also observe that this filter is not able to improve the performance of KKM for case (iii), which is a known problematic data set for centroid-based methods, as discussed above.

For what concerns EMGM, the partitions obtained by employing the ridge filter are better than those computed on the raw data for case (vi), (vii) and (viii). In particular, very good results are achieved on the Iris data set. In comparison with the ℓ_0-norm regularization, we observe that the ridge regularization provides better

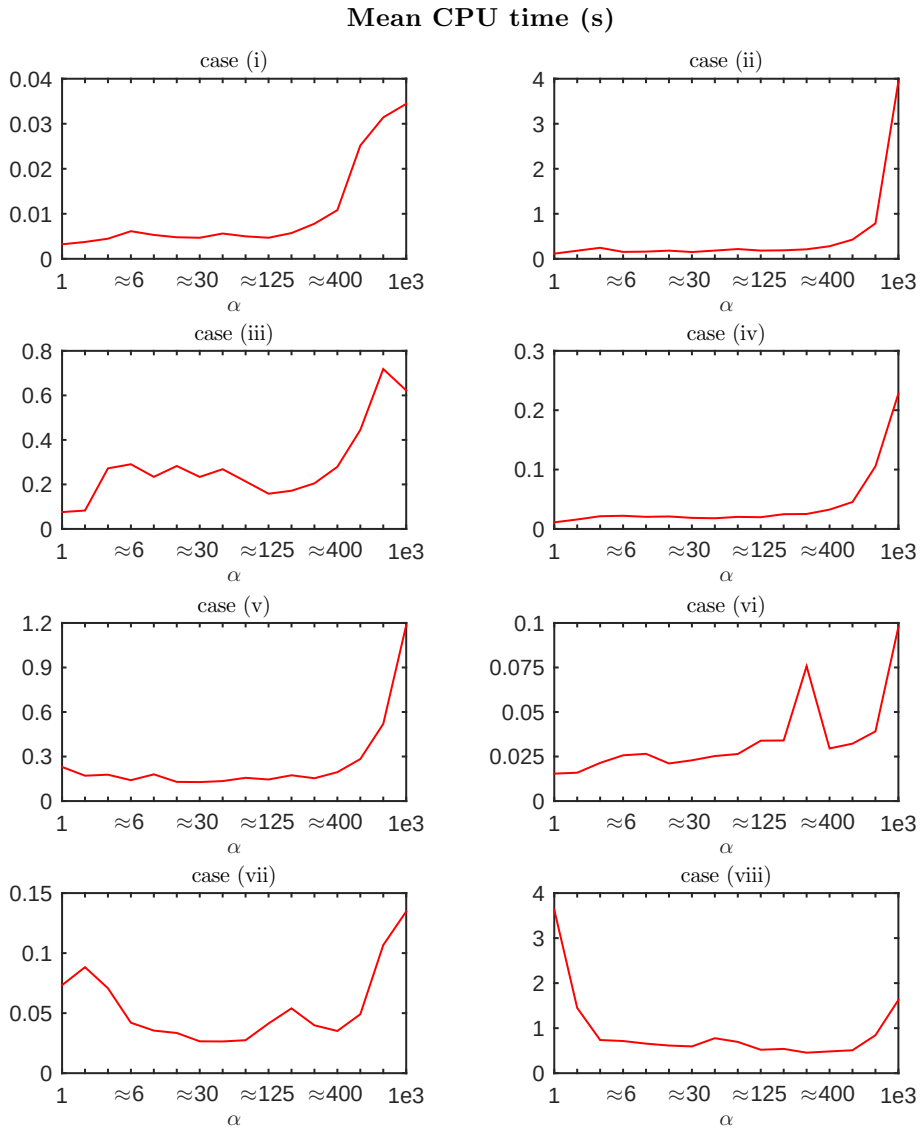

Figure 5.4. CPU time in seconds needed to solve problem (5.7) versus the value of α. In each plot, the computational time is averaged over the 150 considered values of the penalty parameter λ.

results for case (iv) (even though they are still worse than those obtained on the raw data), case (vi) and case (vii), whereas the ℓ_0-norm regularization provides better results for case (i), even if slightly, and case (v).

Summarizing, the ℓ_0-norm regularization based filter seems overall able to benefit different clustering algorithms, and it seems more flexible than the ridge regularization based filter. From a computational point of view, using the squared ℓ_2-norm turns out to be more efficient; however, also the computational time needed by the ℓ_0-norm regularization remains, on average, below an acceptable threshold, for all the considered data sets.

5.5 Conclusions

We have presented a data filtering method for cluster analysis, based on combining two strategies: the first one is the minimization of a least squares function with a weighted ℓ_0-norm penalty, approximated by smooth parametric functions; the second one is choosing the penalty parameter by minimizing a suitable criterion that considers the distances within and between clusters. Numerical results have shown that some existing and widely used clustering algorithms can take advantages from the proposed filtering strategy.

Appendix A

Fundamental notions

In this appendix, we report the fundamental concepts that are needed in this thesis. In Section A.1 we provide the mathematical background and in Section A.2 we recall some basic properties of non-linear programming.

Since the results presented in this appendix are well known, they are reported without proofs.

A.1 Mathematical background

In this section, we recall the mathematical notions that are needed for our purposes. In particular, in Subsection A.1.1 we introduce the notation and provide some basic definitions, in Subsection A.1.2 we report some fundamental concepts of linear algebra and analysis, in Subsection A.1.3 we review differentiability and integrability of functions and in Subsection A.1.4 we focus on convexity and concavity.

A.1.1 Notation and basic definitions

We start by giving the definition of *vector space*.

Definition 12. *A vector space (or linear space) X over a field F is a non-empty set equipped with the operations of addition and scalar multiplication*[1]*, that must satisfy, respectively,*

$$(u+v) \in X, \quad \forall u, v \in X,$$
$$\alpha x \in X, \quad \forall x \in X, \forall \alpha \in F.$$

Every element $x \in X$ is called vector and every element of F is called scalar.

Moreover, if X and W are two vector spaces defined over the same field and with the same operations of addition and scalar multiplication, if $W \subseteq X$, then we say that W is a *vector subspace* (or *linear subspace*) of X.

In this thesis, we are concerned with continuous optimization problems, that is, the problem variables are real numbers. Fot this reason, all the definitions we will

[1]The operations of addition and scalar multiplication must also satisfy specific axioms. The interested reader can refer to any advanced textbook on linear algebra.

provide are limited to the case where the vector space is defined over the field of real numbers.

We use the following notation.

- We indicate with $\mathbb{R}$ the field of real numbers (referred to as *scalars*).

  We indicate with $\mathbb{R}^+$ the set of positive real numbers.

  We indicate with $\mathbb{R}_0^+$ the set of nonnegative real numbers.

  We indicate with $\mathbb{R}^-$ the set of negative real numbers.

  We indicate with $\mathbb{R}_0^-$ the set of nonpositive real numbers.

- We indicate with $\mathbb{R}^n$ the vector space in which each element x (represented as a column vector) is a tuple of n real numbers, where n is a positive integer.

  Given a vector $x \in \mathbb{R}^n$, we indicate with x_i the i-th entry of x.

  Given $x, y \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$, the operations of addition and scalar multiplication in $\mathbb{R}^n$ are defined as

$$x + y = \begin{bmatrix} x_1 + y_1 \\ x_2 + y_2 \\ \vdots \\ x_n + y_n \end{bmatrix}, \qquad \alpha x = \begin{bmatrix} \alpha x_1 \\ \alpha x_2 \\ \vdots \\ \alpha x_n \end{bmatrix}.$$

- We indicate with $e_i \in \mathbb{R}^n$ the i-th unit vector (all components are 0, except for the i-th component which is 1).

- We indicate with $\mathbb{R}^{m \times n}$ the vector space in which each element A is a real matrix with m rows and n columns, where m and n are positive integers.

  Given a matrix $A \in \mathbb{R}^{m \times n}$, we indicate with $A_{i,j}$ the entry in position (i, j).

  Given $A, B \in \mathbb{R}^{m \times n}$ and $\alpha \in \mathbb{R}$, the operations of addition and scalar multiplication in $\mathbb{R}^{m \times n}$ are defined as

$$A + B = \begin{bmatrix} A_{1,1} + B_{1,1} & A_{1,2} + B_{1,2} & \dots & A_{1,n} + B_{1,n} \\ A_{2,1} + B_{2,1} & A_{2,2} + B_{2,2} & \dots & A_{2,n} + B_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m,1} + B_{m,1} & A_{m,2} + B_{m,2} & \dots & A_{m,n} + B_{m,n} \end{bmatrix},$$

$$\alpha A = \begin{bmatrix} \alpha A_{1,1} & \alpha A_{1,2} & \dots & \alpha A_{1,n} \\ \alpha A_{2,1} & \alpha A_{2,2} & \dots & \alpha A_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha A_{m,1} & \alpha A_{m,2} & \dots & \alpha A_{m,n} \end{bmatrix}.$$

- A *square* matrix is a matrix with the same number of rows and columns.

  Given a square matrix $A \in \mathbb{R}^{n \times n}$, the *diagonal* of A is the set of elements $\{A_{i,i}\}$, $i = 1, \dots, n$.

  Given a square matrix $A \in \mathbb{R}^{n \times n}$, we say that A is *symmetric* if $A_{i,j} = A_{j,i}$, for all $i, j = 1, \dots, n$.

- We indicate with $I \in \mathbb{R}^{n \times n}$ the identity matrix (all components are 0, except for the diagonal ones which are 1).
- Given a non-empty set $A \subseteq \mathbb{R}$,
  - the *minimum* of A is the smallest value of A and it is indicated with $\min(A)$;
  - the *maximum* of A is the largest value of A and it is indicated with $\max(A)$;
  - the *infimum* of A is the largest value of the set $\{x \in \mathbb{R} \colon x \leq y, \, \forall y \in A\}$ (which may not belong to A) and it is indicated with $\inf(A)$;
  - the *supremum* of A is the smallest value of the set $\{x \in \mathbb{R} \colon x \geq y, \, \forall y \in A\}$ (which may not belong to A) and it is indicated with $\sup(A)$.
- We indicate with $f \colon X \to Y$ a function that maps every element in the set X (called *domain*) to an element in the set Y (called *codomain*).
- Given a set A, we indicate the *cardinality* of A (that is, the number of elements in A) with $|A|$, or $\text{card}(A)$.
- Given two sets A and B,
  - we indicate with $A \setminus B$ the *set difference* between A and B, that is, the set $\{x \in A \colon x \notin B\}$;
  - we indicate with $A \cup B$ the *union* of A and B, that is, the set $\{x \colon x \in A \text{ or } x \in B\}$;
  - we indicate with $A \cap B$ the *intersection* of A and B, that is, the set $\{x \colon x \in A \text{ and } x \in B\}$;
  - we indicate with $A \times B$ the *Cartesian* product on A and B, that is, the set $\{(a, b) \colon a \in A, \, b \in B\}$.
- We indicate the empty set with $\emptyset$.
- Given two real numbers a and b, with $a \leq b$,
  - we indicate with $[a, b]$ the set $\{x \in \mathbb{R} \colon a \leq x \leq b\}$,
  - we indicate with (a, b) the set $\{x \in \mathbb{R} \colon a < x < b\}$,
  - we indicate with $[a, b)$ the set $\{x \in \mathbb{R} \colon a \leq x < b\}$,
  - we indicate with $(a, b]$ the set $\{x \in \mathbb{R} \colon a < x \leq b\}$.

Given a vector $x \in \mathbb{R}^n$, we indicate with x^T the *transpose* of x, that is, the row vector $\begin{bmatrix} x_1 & \dots & x_n \end{bmatrix} \in \mathbb{R}^{1 \times n}$. Given a matrix $A \in \mathbb{R}^{m \times n}$, we indicate with A^T the transpose of A, that is, the $(n \times m)$ matrix whose entry in position (i, j) is equal to $A_{j,i}$.

Given $x, y \in \mathbb{R}^n$, we indicate with $x^T y$, or with $\langle x, y \rangle$, the standard inner product in $\mathbb{R}^n$, given by the scalar $\sum_{i=1}^{n} x_i \, y_i$. Given $A \in \mathbb{R}^{m \times n}$ and $x \in \mathbb{R}^n$, we indicate with

Ax the standard matrix-vector product, given by the m-dimensional vector whose i-th entry is equal to $A_i x$, being $A_i \in \mathbb{R}^{1\times n}$ the i-th row vector of A. Given $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times p}$, we indicate with AB the standard matrix product, given by the $(m \times p)$ matrix whose entry in position (i, j) is equal to $A_i B^j$, being $A_i \in \mathbb{R}^{1\times n}$ and $B^j \in \mathbb{R}^n$ the i-th row vector of A and the j-th column vector of B, respectively.

Given a vector $x \in \mathbb{R}^n$, the notation $x = 0$ means that all components of x are zero. Similarly, given a matrix A, the notation $A = 0$ means that all components of A are zero.

We also define the set of *(affinely) extended real numbers* $\overline{\mathbb{R}}$ as

$$\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty\} \cup \{+\infty\}.$$

Although the elements $-\infty$ and $+\infty$ are not real numbers, the set $\bar{R}$ can be made ordered by defining

$$-\infty \leq x \leq +\infty, \quad \forall x \in \overline{\mathbb{R}}.$$

A.1.2 Concepts of linear algebra and analysis

We start this subsection recalling the definition of *linear combination*, *linear hull* and *linear dependency* of a set of vectors in $\mathbb{R}^n$.

Definition 13. *Given $x_1, \ldots, x_m \subseteq \mathbb{R}^n$, we say that $\bar{x} \in \mathbb{R}^n$ is a linear combination of $x_1, \ldots, x_m \subseteq \mathbb{R}^n$ if there exist $\alpha_1, \ldots, \alpha_m \in \mathbb{R}$ such that*

$$\bar{x} = \sum_{i=1}^{m} \alpha_i x_i.$$

Definition 14. *Given a set $S \subseteq \mathbb{R}^n$, we call linear hull of S the set containing all linear combinations of the elements of S.*
Moreover, we indicate the linear hull of S with $\mathrm{lin}(S)$.

Definition 15. *Given $x_1, \ldots, x_m \subseteq \mathbb{R}^n$, we say that $x_1, \ldots, x_m$ are linearly dependent if there exist $\alpha_1, \ldots, \alpha_m \in \mathbb{R}$ that are not all zero and such that*

$$\sum_{i=1}^{m} \alpha_i x_i = 0.$$

Otherwise, $x_1, \ldots, x_m$ are said to be linearly independent.

Moreover, a set $S \subseteq \mathbb{R}^n$ is said to be linearly dependent if there exists a finite set of elements in S that are linearly dependent. Otherwise, S is said to be linearly independent.

Now, we introduce the concepts of *rank* and *basis* of a set.

Definition 16. *Given $S \subseteq \mathbb{R}^n$, the rank of S as the maximum number of linearly independent vectors contained in S. We indicate the rank of S with* $\mathrm{rk}(S)$.

Definition 17. *Let W be a vector subspace of $\mathbb{R}^n$. A basis of W is a linearly independent set $B \subseteq W$, such that every element of W can be expressed as a linear combination of the elements of B.*

The following properties hold.

Theorem 19. *Let W be a vector subspace of $\mathbb{R}^n$. Then,*

- *there exists a basis of W;*
- *if B is a basis of W, then every element of W can be univocally expressed as a linear combination of the elements of B;*
- *all bases of W have the same cardinality.*

Now, we can give the following definition of *dimension* of a vector subspace.

Definition 18. *Let W be a vector subspace of $\mathbb{R}^n$ and let B be a basis of W. Then, the cardinality of B is called dimension of W.*

In particular, we say that a vector subspace W has *finite dimension* if its dimension is finite, otherwise we say that W has *infinite dimension*. The vector space $\mathbb{R}^n$ has finite dimension, which is equal to n.

Now, we introduce the notions of *rank* and *basis* of a matrix.

Definition 19. *Given $A \in \mathbb{R}^{m\times n}$, the rank of A is the maximum number of linearly independent column vectors of A. We indicate the rank of A with* $\mathrm{rk}(A)$.

It can be proved that the maximum number of linearly independent column vectors of a real matrix A is equal to the maximum number of linearly independent row vectors of A.

It follows that, for every matrix $A \in \mathbb{R}^{m\times n}$, we have $\mathrm{rk}(A) \leq \min\{m, n\}$. In particular, if $\mathrm{rk}(A) = \min\{m, n\}$, then A is said to have *full rank*, otherwise it is said to be *rank deficient*.

From known results, we have that a square matrix $A \in \mathbb{R}^{n\times n}$ has full rank if and only if there exists a matrix $B \in \mathbb{R}^{n\times n}$ such that $AB = BA = I$. In this case, the matrix A is said to be *non-singular* (or *invertible*), the matrix B is said to be the *inverse* of A and it is usually indicated by A^{-1}. Conversely, if a square matrix $A \in \mathbb{R}^{n\times n}$ is rank deficient, then A is said to be *singular* (or *non-invertible*).

In the following result, we provide a well-known sufficient and necessary condition to ensure that the column vectors of a real matrix are linearly independent.

Theorem 20. *Given $A \in \mathbb{R}^{m\times n}$, the column vectors of A are linearly independent (that is,* $\mathrm{rk}(A) = n$*) if and only if*

$$Ax = 0 \Rightarrow x = 0.$$

We can also state the following necessary and sufficient condition to guarantee the existence of solutions of a linear system of equations.

Theorem 21. *Given $A \in \mathbb{R}^{m\times n}$ and a vector $b \in \mathbb{R}^n$, then there exists $x \in \mathbb{R}^n$ such that $Ax = b$ if and only if*

$$\mathrm{rk}(A) = \mathrm{rk}\Big(\begin{bmatrix} A & b \end{bmatrix}\Big).$$

If such a vector x exists, it is unique if and only if $\mathrm{rk}(A) = n$.

Now, we recall the definition of *norm* of a vector, which is usually employed to measure the "length" of a vector.

Definition 20. *A function* $\|\cdot\|\colon \mathbb{R}^n \to \mathbb{R}$ *is a norm if*

- $\|x\| \geq 0, \quad \forall x \in \mathbb{R}^n$;
- $\|x\| = 0 \Leftrightarrow x = 0$;
- $\|x + y\| \leq \|x\| + \|y\|, \quad \forall x, y \in \mathbb{R}^n$;
- $\|\alpha x\| = |\alpha| \|x\|, \quad \forall x \in \mathbb{R}^n, \alpha \in \mathbb{R}$.

In $\mathbb{R}^n$, we define the ℓ_p-norm as:

$$\|x\|_p = \left(\sum_{i=1}^{n} |x|^p \right)^{\frac{1}{p}}, \quad p \geq 1.$$

It can be shown that, for all $1 \leq p \leq \infty$, the functions $\|\cdot\|_p$ fulfill all the points of Definition 20.

In particular, when $p = 2$ we have the *Euclidean norm*. The space $\mathbb{R}^n$ equipped with the Euclidean norm is called *Euclidean space*. Moreover, the Euclidean norm satisfies the *Cauchy-Schwarz* inequality:

$$|x^T y| \leq \|x\|_2 \|y\|_2, \quad \forall x, y \in \mathbb{R}^n.$$

We indicate the Euclidean norm also with $\|\cdot\|$ (thus omitting the subscript).

When $p = \infty$, we have the *supremum norm* (or *sup-norm*, or *Chebyshev norm*):

$$\|x\|_\infty = \max_{i=1,\dots,n} \{|x_i|\}.$$

Sometimes, authors also define the l_0-norm as the number of non-zero entries of a vector $x \in \mathbb{R}^n$, that is

$$\|x\|_0 = \text{card}\{i \in \{1, \dots, n\} \colon x_i \neq 0\}. \tag{A.1}$$

Actually, this function is not a norm (because it does not satisfy the last point of Definition 20) and, rigorously, this terminology should be avoided (for instance, some authors put the word *norm* between quotation marks). Nevertheless, this notation is commonly used, especially in computer sciences, machine learning and signal processing. Keeping this in mind, in Chapter 5 we address problems in which (A.1) must be minimized and, with an abuse of terminology, we refer to (A.1) as l_0-norm.

Now, we recall the definition of *metric*, which is usually employed to measure the distance between two vectors.

Definition 21. *A function* $d\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ *is a metric (or distance function) if*

- $d(x, y) \geq 0, \quad \forall x, y \in \mathbb{R}^n$;
- $d(x, y) = 0 \Leftrightarrow x = y$;

- $d(x,y) \leq d(x,z) + d(z,y), \quad \forall x,y,z \in \mathbb{R}^n$;
- $d(x,y) = d(y,x), \quad \forall x,y \in \mathbb{R}^n$.

From the above definitions, it follows that the norm can be used to define a metric on $\mathbb{R}^n$ by setting

$$d(x,y) = \|x - y\|.$$

Now, we recall some useful properties of real matrices. Similarly as for vectors, it is possible to define the *norm* of a matrix as follows.

Definition 22. *A function* $\|\cdot\| \colon \mathbb{R}^{m\times n} \to \mathbb{R}$ *is a norm if*

- $\|A\| \geq 0, \quad \forall A \in \mathbb{R}^{m\times n}$;
- $\|A\| = 0 \Leftrightarrow A = 0$;
- $\|A + B\| \leq \|A\| + \|B\|, \quad \forall A, B \in \mathbb{R}^{m\times n}$;
- $\|\alpha A\| = |\alpha|\|A\|, \quad \forall A \in \mathbb{R}^{m\times n}, \alpha \in \mathbb{R}$.

Some widely used matrix norms are the *Frobenius norm*, defined as

$$\|A\|_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} A_{i,j}^2 \right)^{\frac{1}{2}},$$

and those induced by a vector norm:

$$\|A\|_p = \sup_{x\in\mathbb{R}^n,\, x\neq 0} \frac{\|Ax\|_p}{\|x\|_p}, \quad p \geq 1.$$

All the above norms also satisfy the *consistency property*

$$\|AB\| \leq \|A\|\|B\|, \quad \forall A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{n\times q}.$$

Now, we focus on square matrices and we report the definition of *eigenvalue* and *eigenvector*.

Definition 23. *Given a square matrix* $A \in \mathbb{R}^{n\times n}$*, a scalar* λ *is said to be an eigenvalue of* A *if there exists a vector* $u \in \mathbb{R}^n$*,* $u \neq 0$*, such that*

$$A\lambda = \lambda u.$$

Moreover, u *is called eigenvector of* A*.*

For symmetric real matrices, the following result holds.

Theorem 22. *Let* $A \in \mathbb{R}^{n\times n}$ *be a symmetric matrix. Then, there exist* n *real eigenvalues of* A*. Moreover, there exist* n *orthonormal eigenvectors of* A*, that is, there exist* $u_1, \ldots, u_n \in \mathbb{R}^n$ *that are eigenvectors of* A *and satisfy*

$$u_i^T u_j = 0, \quad i,j = 1,\ldots,n, \quad i \neq j,$$
$$\|u_i\| = 1, \quad i = 1,\ldots,n.$$

Now, we can give the following definitions.

Definition 24. *Given a square matrix $A \in \mathbb{R}^{n\times n}$, we say that*

- *A is positive definite if $d^T Ad > 0$ for all $d \in \mathbb{R}^n$, $d \neq 0$, or equivalently, if all its eigenvalues are positive. We indicate that A is positive definite by $A \succ 0$.*
- *A is positive semidefinite if $d^T Ad \geq 0$ for all $d \in \mathbb{R}^n$, or equivalently, if all its eigenvalues are nonnegative. We indicate that A is positive semidefinite by $A \succeq 0$.*
- *A is negative definite if $d^T Ad < 0$ for all $d \in \mathbb{R}^n$, $d \neq 0$, or equivalently, if all its eigenvalues are negative. We indicate that A is negative definite by $A \prec 0$.*
- *A is negative semidefinite if $d^T Ad \leq 0$ for all $d \in \mathbb{R}^n$, or equivalently, if all its eigenvalues are nonpositive. We indicate that A is negative semidefinite by $A \preceq 0$.*

Moreover, indicating with λ_m and λ_M the smallest and largest eigenvalue of $A \in \mathbb{R}^{n\times n}$, respectively, we have

$$\lambda_m = \min_{x\in\mathbb{R}^n,\, x\neq 0} \frac{x^T Ax}{\|x\|^2}, \qquad \lambda_M = \max_{x\in\mathbb{R}^n,\, x\neq 0} \frac{x^T Ax}{\|x\|^2}.$$

It can also be shown that a square matrix is non-singular if and only it all its eigenvalues are non-zero. It follows that both positive definite matrices and negative definite matrices are non-singular.

Now, we recall some properties of a sequence of vectors in $\mathbb{R}^n$. A sequence of vectors in $\mathbb{R}^n$ is usually indicated by $\{x^k\}$ and it can be defined as a countably infinite ordered set, where every element is a vector of $\mathbb{R}^n$ which is univocally associated to a nonnegative integer k (referred to as *index*).

First, we report the definition of *limit* of a sequence, that is useful to study the behavior of the sequence when the index k becomes infinitely large.

Definition 25. *Given a sequence $\{x^k\} \subseteq \mathbb{R}^n$, we say that*

- *$\{x^k\}$ is convergent if there exists a finite value $x^* \in \mathbb{R}^n$ such that, for every scalar $\epsilon > 0$, there exists an index k satisfying $\|x^k - x^*\| < \epsilon$ for all $k \geq k$.*

  In this case, we say that $\{x^k\}$ converges to x^*, or equivalently, that $\{x^k\}$ has limit x^*, and we write

  $$\lim_{k\to\infty} x^k = x^*.$$

- *$\{x^k\}$ is divergent if it is not convergent.*

  Moreover, if $\{x^k\}$ is a sequence of scalars, we can distinguish three further cases:

  - *if, for every $M > 0$, there exists an index k such that $x^k > M$, then we write*

    $$\lim_{k\to\infty} x^k = +\infty;$$

– *if, for every $M > 0$, there exists an index k such that $x^k < -M$, then we write*

$$\lim_{k\to\infty} x^k = -\infty;$$

– *otherwise, the sequence is said to be oscillating.*

From known results, we have that if the limit of a sequence $\{x^k\} \subseteq \mathbb{R}^n$ exists, then it is unique.

Given a sequence $\{x^k\} \subseteq \mathbb{R}^n$, we indicate with $\{x^k\}_K$ a subsequence of $\{x^k\}$, where $K \subseteq \{0, 1, \ldots\}$ is an infinite index subset.

Now, we report the definition of *limit point*.

Definition 26. *Given a sequence $\{x^k\} \subseteq \mathbb{R}^n$, we say that x^* is a limit point (or equivalently, an accumulation point) of $\{x^k\}$ if there exists a subsequence of $\{x^k\}$ converging to x^*.*

From known results, we also have that if $\{x^k\}$ converges to x^*, then every subsequence of $\{x^k\}$ converges to x^*.

Now, we give the definition of *bounded* and *unbounded* sequence.

Definition 27. *Given a sequence $\{x^k\} \subseteq \mathbb{R}^n$, we say that*

- *$\{x^k\}$ is bounded if there exists $M > 0$ such that $\|x^k\| \leq M$ for all k;*
- *$\{x^k\}$ is unbounded if, for every $M > 0$, there exists an index k such that $\|x\|^k > M$.*

It can be shown that every convergent sequence in $\mathbb{R}^n$ is bounded. Furthermore, from *Bolzano-Weierstrass theorem*, we have that every bounded sequence $\{x^k\} \subseteq \mathbb{R}^n$ has limit points.

Considering a sequence of scalars, we can also give the following definition of *monotonicity*.

Definition 28. *Given a sequence $\{x^k\} \subseteq \mathbb{R}$, we say that*

- *$\{x^k\}$ is monotonically decreasing (respectively non-increasing) if $x^{k+1} < x^k$ (respectively, $x^{k+1} \leq x^k$) for all k,*
- *$\{x^k\}$ is monotonically increasing (respectively non-decreasing) if $x^{k+1} > x^k$ (respectively, $x^{k+1} \geq x^k$) for all k,*
- *$\{x^k\}$ is bounded from belove if there exists $M \in \mathbb{R}$ such that $x^k \geq M$ for all k,*
- *$\{x^k\}$ is bounded from above if there exists $M \in \mathbb{R}$ such that $x^k \leq M$ for all k.*

It can also be shown that, if a sequence of scalars $\{a^k\}$ is monotonically non-increasing (respectively non-decreasing) and bounded from below (respectively from above), then $\{a^k\}$ converges.

Now, we focus on the properties of sets in $\mathbb{R}^n$. We first report the following definitions.

Definition 29. *Given a set $S \subseteq \mathbb{R}^n$, we say that*

- *$x \in S$ is an interior point of S if there exists $\rho > 0$ such that*

$$\{y \in \mathbb{R}^n \colon \|x - y\| < \rho\} \subseteq S;$$

- *$x \in \mathbb{R}^n$ is a closure point of S if*

$$\{y \in \mathbb{R}^n \colon \|x - y\| < \rho\} \cap S \neq \emptyset, \quad \forall \rho > 0.$$

Definition 30. *Given a set $S \subseteq \mathbb{R}^n$,*

- *the interior of S is the set containing all interior points of S;*
- *the closure of S is the set containing all closure points of S;*
- *the boundary of S is the closure of S without the interior of S;*
- *$x \in \mathbb{R}^n$ is a boundary point of S if x belongs to the boundary of S;*
- *S is closed if it coincides with its closure;*
- *S is open if $\mathbb{R}^n \setminus S$ is closed;*
- *S is bounded if there exists $M > 0$ such that $\|x\| \leq M$ for all $x \in S$;*
- *S is unbounded if, for every $M > 0$, there exists $x_M \in S$ such that $\|x_M\| \geq M$;*
- *S is compact if it is closed and bounded.*

Moreover, we define the *relative interior* of a set $S \subseteq \mathbb{R}^n$ as the interior of S within the linear hull of S and we indicate it with $\text{relint}(S)$. Namely,

$$\text{relint}(S) = \{x \in S \colon \exists \rho > 0, \mathcal{B}(x, \rho) \cap \text{lin}\,(S) \subseteq S\}.$$

We can also give a characterization of closed, bounded and compact sets by analyzing the behaviour of specific sequences, as shown in the next proposition.

Proposition 21. *Given a set $S \subseteq \mathbb{R}^n$, we have that*

- *S is closed if and only if, for every sequence $\{x^k\} \subseteq S$ that has limit points, every limit point of $\{x^k\}$ belongs to S;*
- *S is bounded if and only if every sequence $\{x^k\} \subseteq S$ has limit points (possibly not belonging to S);*
- *S is compact if and only if every sequence $\{x^k\} \subseteq S$ has limit points and every limit point of $\{x^k\}$ belongs to S.*

Now, we provide the following definitions of *half-space*, *hyperplane*, *polyhedron*, *vertex* and *polytope*.

Definition 31. *Given $a \in \mathbb{R}^n$, $a \neq 0$, and $b \in \mathbb{R}$, we say that*

- *$\{x \in \mathbb{R}^n \colon a^T x - b \geq 0\}$ is a closed half-space,*

- $\{x \in \mathbb{R}^n : a^T x - b > 0\}$ *is an open half-space,*
- $\{x \in \mathbb{R}^n : a^T x - b = 0\}$ *is a hyperplane.*

Definition 32. *A polyhedron $P \subseteq \mathbb{R}^n$ is a set given by the intersection of a finite number of closed half-spaces and hyperplanes.*

Definition 33. *Given a polyhedron $P \subseteq \mathbb{R}^n$, we say that $x \in \mathbb{R}^n$ is a vertex of P if there do not exist $x_1, x_2 \in \mathbb{R}^n$, $x_1 \neq x_2$, such that $x = \lambda x_1 + (1-\lambda)x_2$, $\lambda \in (0,1)$.*

Definition 34. *A polytope is a bounded polyhedron.*

Now, we focus on functions, recalling some concepts that are useful for our purposes.

For the sake of simplicity, we will only consider functions whose domain is $\mathbb{R}^n$ (or $\mathbb{R}$), but the following definitions can be properly adapted to the case in which the domain is a subset of $\mathbb{R}^n$ (or $\mathbb{R}$).

In the following, we will also distinguish between univariate (i.e., the domain is $\mathbb{R}$) and multivariate (i.e., the domain is $\mathbb{R}^n$) functions and between real-valued (i.e., the codomain is $\mathbb{R}$) and vector-valued (i.e., the codomain is $\mathbb{R}^m$) functions. When convenient, the different cases will be considered separately. Anyway, we recall that a vector-valued function $f : \mathbb{R}^n \to \mathbb{R}^m$ can be represented as a vector of m functions, namely, $f(x) = \begin{bmatrix} f_1(x) & \dots & f_m(x) \end{bmatrix}^T$.

We start by providing the following definitions for a real-valued function.

Definition 35. *Given a function $f : \mathbb{R}^n \to \mathbb{R}$, we say that*

- *f is affine if there exist $\alpha_0, \alpha_1, \dots, \alpha_n \in \mathbb{R}$ such that*

$$f(x) = \alpha_0 + \sum_{i=1}^{n} \alpha_i x_i, \quad \forall x \in \mathbb{R}^n;$$

- *f is linear if there exist $\alpha_1, \dots, \alpha_n \in \mathbb{R}$ such that*

$$f(x) = \sum_{i=1}^{n} \alpha_i x_i, \quad \forall x \in \mathbb{R}^n;$$

- *f is bounded if there exists $M > 0$ such that*

$$\|f(x)\| \leq M, \quad \forall x \in \mathbb{R}^n;$$

- *f is bounded from belove if there exists $M \in \mathbb{R}$ such that*

$$f(x) \geq M, \quad \forall x \in \mathbb{R}^n;$$

- *f is bounded from above if there exists $M \in \mathbb{R}$ such that*

$$f(x) \leq M, \quad \forall x \in \mathbb{R}^n.$$

For what concerns vector-valued functions $f\colon \mathbb{R}^n \to \mathbb{R}^m$, we say that f is *affine*, *linear*, or *bounded*, if each f_i, $i = 1, \dots, m$, is affine, linear, or bounded, respectively.

Now, we focus on the *limit of a function*. First, we need to give the definition of *neighborhood*.

Definition 36. *Given a point $x \in \mathbb{R}^n$ and a scalar $\rho > 0$, we indicate with $\mathcal{B}(x, \rho)$ the (open) neighborhood of x with radius ρ, (also called (open) ball centered at x with radius ρ), defined as*

$$\mathcal{B}(x, \rho) = \{y \in \mathbb{R}^n : \|x - y\| < \rho\}.$$

We can also extend the notion of neighborhood to the set $\bar{R}$ by defining the neighborhoods of $-\infty$ and $+\infty$ as follows:

- a neighborhood of $-\infty$ is any interval $[-\infty, a)$, with $a \in \overline{\mathbb{R}} \setminus \{-\infty\}$,
- a neighborhood of $+\infty$ is any interval $(b, +\infty]$, with $b \in \overline{\mathbb{R}} \setminus \{+\infty\}$.

Now, we introduce the notion of *limit* for a multivariate real-valued function $f(x)$. First, we consider the case in which x approaches a finite point $x^* \in \mathbb{R}^n$.

Definition 37. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}$, a point $x^* \in \mathbb{R}^n$ and $\ell \in \overline{\mathbb{R}}$, we write*

$$\lim_{x \to x^*} f(x) = \ell, \quad \textit{or equivalently,} \quad f(x) \to \ell \textit{ for } x \to x^*,$$

if, for every neighborhood $\mathcal{V}$ of ℓ, there exists a neighborhood $\mathcal{U}$ of x^* such that $f(x) \in \mathcal{V}$ for all $x \in \mathcal{U}$.

Moreover, if $\ell \in \mathbb{R}$, then we say that f has (finite) limit ℓ as x approaches x^*.

Now, we introduce the concept of limit for $\|x\| \to +\infty$, in order to describe the behaviour of a function when $\|x\|$ becomes infinitely large.

Definition 38. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}$ and $\ell \in \overline{\mathbb{R}}$, we write*

$$\lim_{\|x\| \to +\infty} f(x) = \ell, \quad \textit{or equivalently,} \quad f(x) \to \ell \textit{ for } \|x\| \to +\infty,$$

if, for every neighborhood $\mathcal{V}$ of ℓ, there exists $M > 0$ such that $f(x) \in \mathcal{V}$ for all $x \in \mathbb{R}^n$ satisfying $\|x\| > M$.

For what concerns vector-valued functions $f\colon \mathbb{R}^n \to \mathbb{R}^m$, the calculation of the limits of f boils down to computing the limits of the real-valued functions $f_1, \dots, f_m$. Namely, the limit of f can be taken component-wise.

Now, we recall the definitions of *continuity*, *Lipschitz continuity* and *uniform continuity* (we only consider multivariate vector-valued functions to provide general definitions).

Definition 39. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}^m$ and a point $x \in \mathbb{R}^n$, we say that f is continuous at x if*

$$\lim_{y \to x} f(y) = f(x).$$

Moreover, we say that f is continuous on $S \subseteq \mathbb{R}^n$ if f is continuous at every point $x \in S$. We indicate that f is continuous on $S \subseteq \mathbb{R}^n$ with $f \in C(S)$.

Definition 40. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}^m$, we say that f is Lipschitz continuous on $S \subseteq \mathbb{R}^n$ if there exists a constant $L > 0$ (called Lipschitz constant) such that*

$$\|f(x) - f(y)\| \leq L\|x - y\|, \quad \forall x, y \in S.$$

Definition 41. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}^m$, we say that f is uniformly continuous on $S \subseteq \mathbb{R}^n$ if, for every $\epsilon > 0$, there exists $\rho > 0$ such that*

$$\|f(x) - f(y)\| < \epsilon, \quad \forall x, y \in S\colon \|x - y\| < \rho.$$

Given a function $f\colon \mathbb{R}^n \to \mathbb{R}^m$ and a set $S \subseteq \mathbb{R}^n$, it is straightforward to verify that

- if f is Lipschitz continuous on S, then f is uniformly continuous on S;
- if f is uniformly continuous on S, then f is continuous on S.

Moreover, from *Heine-Cantor theorem*, we have that if a function $f\colon \mathbb{R}^n \to \mathbb{R}^m$ is continuous on a compact set $S \subset \mathbb{R}^n$, then f is uniformly continuous on S.

For real-valued functions, we can also provide the definition of *lower semicontinuity* and *upper semicontinuity*.

Definition 42. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}$ and a point $x \in \mathbb{R}^n$, we say that*

- *f is lower semicontinuous at x if, for every $\epsilon > 0$, there exists $\rho > 0$ such that $f(y) \geq f(x) - \epsilon$ for all $y \in \mathcal{B}(x, \rho)$;*
- *f is upper semicontinuous at x if, for every $\epsilon > 0$, there exists $\rho > 0$ such that $f(y) \leq f(x) + \epsilon$ for all $y \in \mathcal{B}(x, \rho)$.*

Moreover, we say that f is lower semicontinuous (respectively, upper semicontinuous) on $S \subseteq \mathbb{R}^n$ if f is lower semicontinuous (respectively, upper semicontinuous) at every point $x \in S$.

It is straightforward to verify that, if a function $f\colon \mathbb{R}^n \to \mathbb{R}$ is continuous at x, then f is both lower semicontinuous and upper semicontinuous at x.

Now, we provide a fundamental result of non-linear programming, that is, the *Weierstrass theorem*. It states some sufficient conditions to guarantee that a function attains a minimizer on a subset of $\mathbb{R}^n$.

Theorem 23 (Weierstrass theorem)**.** *Let $S \subset \mathbb{R}^n$ be a non-empty compact set and let $f\colon \mathbb{R}^n \to \mathbb{R}$ be a continuous function on S. Then, there exists $x^* \in S$ such that*

$$f(x^*) \leq f(x), \quad \forall x \in S.$$

Actually, the hypotheses of the above theorem can be relaxed. In particular, it can be shown that if $f\colon \mathbb{R}^n \to \mathbb{R}$ is lower semicontinuous (respectively, upper semicontinuous) on a non-empty compact set $S \subset \mathbb{R}^n$, then there exists $x^* \in S$ such that $f(x^*) \leq f(x)$ (respectively, $f(x^*) \geq f(x)$) for all $x \in S$.

A.1.3 Differentiability and integrability

In this subsection, we review the concepts of differential calculus and integral calculus that are needed in this thesis.

We only focus on functions whose domain is $\mathbb{R}^n$ (or $\mathbb{R}$), since it is sufficient for our purposes. For functions defined on subsets of $\mathbb{R}^n$ (or $\mathbb{R}$), the definitions we provide here need to be suitably adapted.

First, we consider univariate real-valued functions, then we extend our analysis to the other cases. We start by recalling the definition of *derivative* and *differentiability*.

Definition 43. *Given a function $f\colon \mathbb{R} \to \mathbb{R}$ and a point $x \in \mathbb{R}$, if*

$$\lim_{\delta \to 0} \frac{f(x+\delta) - f(x)}{\delta}$$

exists and is finite, then

- *the above limit is called derivative of f at x and it is indicated with*

$$\frac{df(x)}{dx}, \quad \text{or equivalently,} \quad f'(x);$$

- *f is said to be differentiable at x.*

If f is differentiable at every point of a set $S \subseteq \mathbb{R}$, then we say that f is *differentiable on S*.

Moreover, the differentiability of a function implies its continuity, as formalized in the next theorem.

Theorem 24. *Given a function $f\colon \mathbb{R} \to \mathbb{R}$, if f is differentiable at $x \in \mathbb{R}$, then f is continuous at x.*

One of the most important properties that follow from the differentiability of an univariate function f is the possibility to approximate f in a neighborhood of a given point by a linear function. In particular, the next result holds.

Theorem 25. *Given a function $f\colon \mathbb{R} \to \mathbb{R}$, if f is differentiable at $x \in \mathbb{R}$, then*

$$f(x+d) = f(x) + f'(x)d + \omega(d), \quad \forall d \in \mathbb{R},$$

where $\omega\colon \mathbb{R} \to \mathbb{R}$ is a function such that

$$\lim_{u \to 0} \frac{\omega(u)}{u} = 0.$$

Moreover, we can compute the derivative of a composite function by the *chain rule*. In particular, given $\phi\colon \mathbb{R} \to \mathbb{R}$ and $g\colon \mathbb{R} \to \mathbb{R}$, assume that $f(x) = \phi(g(x))$. Let $\bar{x} \in \mathbb{R}$, if g is differentiable at $\bar{x}$ and ϕ is differentiable at $g(\bar{x})$, then f is differentiable at $\bar{x}$ and

$$f'(\bar{x}) = \varphi'(g(\bar{x}))\, g'(\bar{x}).$$

By considering the derivative of a function as a function itself, we can also define higher order derivatives. In particular, we define the *second-order derivative* as follows.

Definition 44. *Given a function $f\colon \mathbb{R} \to \mathbb{R}$ and a point $x \in \mathbb{R}$, if f is differentiable on an open set containing x and*

$$\lim_{\delta \to 0} \frac{f'(x+\delta) - f'(x)}{\delta}$$

exists and is finite, then

- *the above limit is called second-order derivative of f at x and it is indicated with*

$$\frac{d^2 f(x)}{dx^2}, \quad \text{or equivalently,} \quad f''(x);$$

- *f is said to be twice differentiable at x.*

If f is twice differentiable at every point of a set $S \subseteq \mathbb{R}$, then we say that f is *twice differentiable on S*.

The following result points out how *second-order derivatives* can be used to approximate a function in the neighborhood of a given point with a second-order polynomial.

Theorem 26. *Given a function $f\colon \mathbb{R} \to \mathbb{R}$, if f is twice differentiable at $x \in \mathbb{R}$, then*

$$f(x+d) = f(x) + f'(x)d + \frac{1}{2} f''(x)d^2 + \omega(d), \quad \forall d \in \mathbb{R},$$

where $\omega\colon \mathbb{R} \to \mathbb{R}$ is a function such that

$$\lim_{u \to 0} \frac{\omega(u)}{u^2} = 0.$$

Now, we can give the definition of continuous differentiability.

Definition 45. *Given a function $f\colon \mathbb{R} \to \mathbb{R}$ and a set $S \subseteq \mathbb{R}$, if $f'(x)$ exists and is continuous at every $x \in S$, then f is said to be continuously differentiable on S. We indicate that f is continuous differentiable on S with $f \in C^1(S)$.*

Definition 46. *Given a function $f\colon \mathbb{R} \to \mathbb{R}$ and a set $S \subseteq \mathbb{R}$, if $f''(x)$ exists and is continuous at every $x \in S$, then f is said to be twice continuously differentiable on S. We indicate that f is twice continuous differentiable on S with $f \in C^2(S)$.*

Now, we introduce some concepts on integral calculus. First, we need the following definitions.

Definition 47. *Given a non-empty interval $[a,b] \subset \mathbb{R}$, we say that the finite set $D = \{x_1, \ldots, x_q\} \subseteq [a,b]$ is a partition of $[a,b]$ if $a = x_0 < x_1 < \ldots < x_q = b$.*

Definition 48. *Given a limited function $f\colon \mathbb{R} \to \mathbb{R}$, a non-empty interval $[a,b] \subset \mathbb{R}$ and a partition $D = \{x_1, \ldots, x_q\}$ of $[a,b]$, we say that*

- *$s(D,f) = \sum_{i=1}^{q} (x_i - x_{i-1}) \inf\{f(x_{i-1}), f(x_i)\}$ is the lower sum of f with respect to D;*

- $S(D,f)=\sum_{i=1}^{q}(x_i-x_{i-1})\sup\{f(x_{i-1}),f(x_i)\}$ *is the upper sum of* f *with respect to* D*.*

Now, we can give the definition of *Riemann integral*.

Definition 49. *Given a limited function* $f\colon\mathbb{R}\to\mathbb{R}$ *and a non-empty interval* $[a,b]\subset\mathbb{R}$*, if*

$$\sup\{s(D,f)\colon D \text{ is a partition of } [a,b]\}=\inf\{S(D,f)\colon D \text{ is a partition of } [a,b]\},$$

then

- *the above value is called Riemann integral of* f *in* $[a,b]$ *and it is indicated with*

$$\int_a^b f(s)\,ds,$$

- f *is said to be Riemann integrable in* $[a,b]$*,*
- $f(x)$ *is called integrand function.*

Given a function $f\colon\mathbb{R}\to\mathbb{R}$, an interval $[a,b]\subset\mathbb{R}$ and a scalar $c\in[a,b]$, if f is Riemann integrable on $[a,b]$, then we define the *integral function of* f *with respect to* c as

$$F_c(x)=\int_c^x f(s)\,ds,\quad \forall x\in[a,b].$$

Now, we can state the *fundamental theorem of calculus*.

Theorem 27 (Fundamental theorem of calculus)**.** *Given a function* $f\colon\mathbb{R}\to\mathbb{R}$*, an interval* $[a,b]\subset\mathbb{R}$ *and a scalar* $c\in[a,b]$*, let us assume that* f *is Riemann integrable on* $[a,b]$*. If* $f\in C([a,b])$*, then the function* $F_c(x)=\int_c^x f(s)\,ds$ *is differentiable on* $[a,b]$ *and*

$$F_c'(x)=f(x),\quad \forall x\in[a,b].$$

Given a function $f\colon\mathbb{R}\to\mathbb{R}$, we say that the function $F\colon\mathbb{R}\to\mathbb{R}$ is an *antiderivative* (or a *primitive function*) of f if $F'(x)=f(x)$ for all $x\in\mathbb{R}$.

From the fundamental theorem of calculus, it follows that every continuous function f has an antiderivative F. Moreover, if such an F exists, then it is unique up to a constant. We denote an antiderivative of the function f (if any) with the symbol $\int f(s)\,ds$.

Another consequence of the above theorem is that, if $f\in C([a,b])$, then

$$\int_a^b f(s)\,ds=F(b)-F(a),$$

where F is an antiderivative of f.

A known property of Riemann integral is that, if f is Riemann integrable on $[a,b]$, then $\left|\int_a^b f(s)\,ds\right|\le\int_a^b|f(s)|\,ds$.

Now, we are ready to state the following properties.

Theorem 28 (Mean value theorem)**.** *Let the function $f\colon \mathbb{R} \to \mathbb{R}$ be differentiable at x. Then, for every $d \in \mathbb{R}$, we have*

$$f(x+d) = f(x) + \int_0^1 f'(x+sd)d \, ds,$$

or equivalently,

$$f(x+d) = f(x) + f'(z)d,$$

where $z = x + \xi d$, $\xi \in (0,1)$.

Theorem 29 (Second-order mean value theorem)**.** *Let the function $f\colon \mathbb{R} \to \mathbb{R}$ be twice differentiable at x. Then, for every $d \in \mathbb{R}$, we have*

$$f(x+d) = f(x) + f'(x)d + \int_0^1 (1-s)f''(x+sd)d^2 \, ds,$$

or equivalently,

$$f(x+d) = f(x) + f'(x)d + \frac{1}{2}f''(z)d^2,$$

where $z = x + \xi d$, $\xi \in (0,1)$.

Now, we consider multivariate and vector-valued functions and we extend the concept of differentiability. First, we give the definition of *partial derivative*, *directional derivative*, *gradient*, *Jacobian matrix* and *Hessian matrix*.

Definition 50. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}$ and a point $x \in \mathbb{R}^n$, if*

$$\lim_{\delta \to 0} \frac{f(x + \delta e_i) - f(x)}{\delta}$$

exists and is finite for an index $i \in \{1, \ldots, n\}$, then the above limit is called i-th partial derivative of f at x and it is indicated with

$$\frac{\partial f(x)}{\partial x_i}.$$

Definition 51. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}$, a point $x \in \mathbb{R}^n$ and a vector $d \in \mathbb{R}^n$, if*

$$\lim_{\delta \to 0} \frac{f(x + \delta d) - f(x)}{\delta}$$

exists and is finite, then the above limit is called directional derivative of f at x along d.

Definition 52. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}$ and a point $x \in \mathbb{R}^n$, if $\dfrac{\partial f(x)}{\partial x_i}$ exists and is finite for every $i = 1, \ldots, n$, then the gradient of f at x is defined as the n-column vector*

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix}.$$

Moreover, we indicate with $\nabla_i f(x)$ *the* i*-th component of* $\nabla f(x)$*, that is,*

$$\nabla_i f(x) = \frac{\partial f(x)}{\partial x_i}.$$

Definition 53. *Given a function* $f\colon \mathbb{R}^n \to \mathbb{R}^m$ *and a point* $x \in \mathbb{R}^n$*, if* $\dfrac{\partial f_i(x)}{\partial x_j}$ *exists and is finite for every* $i = 1, \dots, m$*,* $j = 1, \dots, n$*, then the Jacobian matrix of* f *at* x *is defined as the* $(m \times n)$*-matrix*

$$Jf(x) = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_1(x)}{\partial x_2} & \cdots & \frac{\partial f_1(x)}{\partial x_n} \\ \frac{\partial f_2(x)}{\partial x_1} & \frac{\partial f_2(x)}{\partial x_2} & \cdots & \frac{\partial f_2(x)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m(x)}{\partial x_1} & \frac{\partial f_m(x)}{\partial x_2} & \cdots & \frac{\partial f_m(x)}{\partial x_n} \end{bmatrix}.$$

Moreover, we indicate with $J_{i,j} f(x)$ *the element of* $Jf(x)$ *in position* (i, j)*, that is,*

$$J_{i,j} f(x) = \frac{\partial f_i(x)}{\partial x_j}.$$

Definition 54. *Given a function* $f\colon \mathbb{R}^n \to \mathbb{R}$ *and a point* $x \in \mathbb{R}^n$*, if* $\dfrac{\partial}{\partial x_j}\dfrac{\partial f(x)}{\partial x_i} = \dfrac{\partial^2 f(x)}{\partial x_j \partial x_i}$ *exists and is finite for every* $i, j = 1, \dots, n$*, then the Hessian matrix of* f *at* x *is defined as the* $(n \times n)$*-matrix*

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}.$$

Moreover, we indicate with $\nabla^2_{i,j} f(x)$ *the element of* $\nabla^2 f(x)$ *in position* (i, j)*, that is,*

$$\nabla^2_{i,j} f(x) = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}.$$

Now, we introduce the notion of *differentiability* in the sense of *Fréchet*, that enable us to approximate a multivariate function in the neighborhood of a given point with a linear function.

We first provide the definition of Fréchet differentiability for a multivariate real-valued function.

Definition 55. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}$ and a point $x \in \mathbb{R}^n$, we say that f is Fréchet differentiable at x if there exists $g \in \mathbb{R}^n$ such that*

$$f(x+d) = f(x) + g^T x + \omega(\|d\|), \quad \forall d \in \mathbb{R}^n,$$

where $\omega\colon \mathbb{R} \to \mathbb{R}$ is a function such that

$$\lim_{u \to 0} \frac{\omega(u)}{u} = 0.$$

Moreover, if f is Fréchet differentiable at x, then g is called Fréchet derivative of f at x.

If f is Fréchet differentiable at every point of a set $S \subseteq \mathbb{R}$, then we say that f is *Fréchet differentiable on S*.

Moreover, the following result holds.

Theorem 30. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}$ and a point $x \in \mathbb{R}^n$, if f is Fréchet differentiable at x, then*

- *f is continuous at x,*
- *$\nabla f(x)$ coincides with the Fréchet derivative of f at x,*
- *for every $d \in \mathbb{R}^n$, the directional derivative of f at x along d exists and is equal to $\nabla f(x)^T d$.*

We can also extend the chain rule to the multivariate case. In particular, given $\varphi\colon \mathbb{R} \to \mathbb{R}$ and $g\colon \mathbb{R}^n \to \mathbb{R}$, assume that $f(x) = \varphi(g(x))$. Let $\bar{x} \in \mathbb{R}^n$, if there exists $\nabla g(\bar{x})$ and φ is differentiable at $g(\bar{x})$, then f is Fréchet differentiable at $\bar{x}$ and

$$\nabla f(\bar{x}) = \varphi'(g(\bar{x}))\, \nabla g(\bar{x}).$$

Now, we give the definition of Fréchet differentiability for a vector-valued multivariate function.

Definition 56. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}^m$ and a point $x \in \mathbb{R}^n$, we say that f is Fréchet differentiable at x if there exists $M \in \mathbb{R}^{m \times n}$ such that*

$$f(x+d) = f(x) + Md + \omega(\|d\|), \quad \forall d \in \mathbb{R}^n,$$

where $\omega\colon \mathbb{R} \to \mathbb{R}$ is a function such that

$$\lim_{u \to 0} \frac{\omega(u)}{u} = 0.$$

Moreover, if f is Fréchet differentiable at x, then M is called Fréchet derivative of f at x.

If $f\colon \mathbb{R}^n \to \mathbb{R}^m$ is Fréchet differentiable at every point of a set $S \subseteq \mathbb{R}$, then we say that f is *Fréchet differentiable on S*.

Moreover, the following result holds.

Theorem 31. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}^m$ and a point $x \in \mathbb{R}^n$, if f is Fréchet differentiable at x, then*

- *f is continuous at x,*
- *$Jf(x)$ coincides with the Fréchet derivative of f at x.*

Furthermore, we have that a function $f\colon \mathbb{R}^n \to \mathbb{R}^m$ is Fréchet differentiable at $x \in \mathbb{R}^n$ if and only if $f_1, \ldots, f_m$ are Fréchet differentiable at x.

We can also extend the chain rule to the vector-valued case. In particular, given $\phi\colon \mathbb{R}^p \to \mathbb{R}^m$ and $g\colon \mathbb{R}^n \to \mathbb{R}^p$, assume that $f(x) = \varphi(g(x))$. Let $\bar{x} \in \mathbb{R}^n$, if g is Fréchet differentiable at $\bar{x}$ and φ is Fréchet differentiable at $g(\bar{x})$, then f is Fréchet differentiable at $\bar{x}$ and

$$Jf(\bar{x}) = J\varphi(g(\bar{x}))\, Jg(\bar{x}).$$

Considering the Fréchet derivative as a function itself, it is also possible to define higher order Fréchet derivatives. In particular, we report the definition of the *second-order Fréchet derivative* for a multivariate real-valued function.

Definition 57. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}$ and a point $x \in \mathbb{R}^n$, we say that f is twice Fréchet differentiable at x if f if Fréchet differentiable on an open neighborhood of x and ∇f is Fréchet differentiable at x.*

We can state the following result.

Theorem 32. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}$ and a point $x \in \mathbb{R}^n$, if f is twice Fréchet differentiable at x, then*

- *$\nabla^2 f(x)$ is symmetric,*
- *$\nabla^2 f(x)$ coincides with the second-order Fréchet derivative of f at x,*
- *$f(x+d) = f(x) + \nabla f(x)^T d + \frac{1}{2} d^T \nabla^2 f(x) d + \omega(\|d\|)$, for all $d \in \mathbb{R}^n$, where $\omega\colon \mathbb{R} \to \mathbb{R}$ is a function such that $\lim_{u \to 0} \frac{\omega(u)}{u^2} = 0$.*

While in the univariate case the differentiability of a function is guaranteed by the existence of the derivate of the function, in the multivariate case a function may not be Fréchet differentiable even if all its partial derivatives exist.

Moreover, even if the directional derivatives of a function f at a given point x exist along every direction d, the function may not be Fréchet differentiable at x.

Anyway, if all partial derivatives of a function f exist and are continuous, then it can be proved that f is Fréchet differentiable.

Before formalizing this result, we need to give the following definitions.

Definition 58. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}$ and a set $S \subseteq \mathbb{R}^n$, if $\nabla f(x)$ exists and is continuous at every $x \in S$, then f is said to be continuously differentiable on S. We indicate that f is continuous differentiable on S with $f \in C^1(S)$.*

Definition 59. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}^m$ and a set $S \subseteq \mathbb{R}^n$, if $\nabla Jf(x)$ exists and is continuous at every $x \in S$, then f is said to be continuously differentiable on S. We indicate that f is continuous differentiable on S with $f \in C^1(S)$.*

Definition 60. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}$ and a set $S \subseteq \mathbb{R}^n$, if $\nabla^2 f(x)$ exists and is continuous at every $x \in S$, then f is said to be twice continuously differentiable on S. We indicate that f is twice continuous differentiable on S with $f \in C^2(S)$.*

Now, we can report the following results.

Theorem 33. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}$ and a point $x \in \mathbb{R}^n$, if there exists $\rho > 0$ such that $f \in C^1(\mathcal{B}(x, \rho))$, then f is Fréchet differentiable at x.*

Theorem 34. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}^m$ and a point $x \in \mathbb{R}^n$, if there exists $\rho > 0$ such that $f \in C^1(\mathcal{B}(x, \rho))$, then f is Fréchet differentiable at x.*

Theorem 35. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}$ and a point $x \in \mathbb{R}^n$, if there exists $\rho > 0$ such that $f \in C^2(\mathcal{B}(x, \rho))$, then f is twice Fréchet differentiable at x.*

Moreover, continuous differentiability of a function on a compact set implies Lipschitz-continuity of the function on that set, as formalized in the following theorem.

Theorem 36. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}^m$ and a compact set $S \subset \mathbb{R}^n$, if f is continuously differentiable on S, then f is Lipschitz continuous on S.*

It is also possible to extend the mean value theorems to multivariate and vector-valued functions.

Theorem 37 (Mean value theorem for a multivariate real-valued function)**.** *Let the function $f\colon \mathbb{R}^n \to \mathbb{R}$ be Fréchet differentiable at x. Then, for every $d \in \mathbb{R}^n$, we have*

$$f(x+d) = f(x) + \int_0^1 \nabla f(x+sd)^T d \, ds,$$

or equivalently,

$$f(x+d) = f(x) + \nabla f(z)^T d,$$

where $z = x + \xi d$, $\xi \in (0,1)$.

Before reporting the mean value theorem for a function $f\colon \mathbb{R}^n \to \mathbb{R}^m$, we need to suitably extend the Riemann integral to the vector-valued case.

Definition 61. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}^m$ and a non-empty interval $[a, b]$, if every f_i, $i = 1, \ldots, m$, is Riemann integrable in $[a, b]$, then f is said to be Riemann integrable in $[a, b]$ and we define the Riemann integral of f in $[a, b]$ as*

$$\int_a^b f(s)\, ds = \left[\int_a^b f_1(s)\, ds \quad \ldots \quad \int_a^b f_m(s)\, ds \right]^T.$$

It can also be shown that, if the function $f\colon \mathbb{R}^n \to \mathbb{R}^m$ is Riemann integrable in the interval $[a, b]$, then $\left\| \int_a^b f(s)\, ds \right\| \leq \int_a^b \|f(s)\|\, ds$.

Now, we can state the following result.

Theorem 38 (Mean value theorem for a multivariate vector-valued function)**.** *Let $f\colon \mathbb{R}^n \to \mathbb{R}^m$ be Fréchet differentiable at x. Then, for every $d \in \mathbb{R}^n$, we have*

$$f(x+d) = f(x) + \int_0^1 Jf(x+sd)d \, ds.$$

Now, we report the second-order mean value theorem for a multivariate real-valued function.

Theorem 39 (Second-order mean value theorem for a multivariate real-valued function)**.** *Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be twice Fréchet differentiable at x. Then, for every $d \in \mathbb{R}^n$, we have*

$$f(x+d) = f(x) + \nabla f(x)^T d + \int_0^1 (1-s) d^T \nabla^2 f(x+sd) d \, ds,$$

or equivalently,

$$f(x+d) = f(x) + \nabla f(x)^T d + \frac{1}{2} d^T \nabla^2 f(z) d,$$

where $z = x + \xi d$, $\xi \in (0,1)$.

A.1.4 Convexity and concavity

We conclude this appendix by reporting the definitions and the fundamental properties of *convexity* and *concavity*.

First, we provide the definition of *convex set*.

Definition 62. *A set $S \subseteq \mathbb{R}^n$ is said to be convex if, for all $x_1, x_2 \in S$,*

$$\lambda x_1 + (1-\lambda) x_2 \in S, \quad \forall \lambda \in [0,1].$$

Now, we can define the *convex combination* and the *convex hull* of a set of vectors in $\mathbb{R}^n$.

Definition 63. *Given $x_1, \ldots, x_m \subseteq \mathbb{R}^n$, we say that $\bar{x} \in \mathbb{R}^n$ is a convex combination of $x_1, \ldots, x_m \subseteq \mathbb{R}^n$ if there exist $\alpha_1, \ldots, \alpha_m \in \mathbb{R}$ such that*

$$\begin{aligned} \bar{x} &= \sum_{i=1}^m \alpha_i x_i, \\ \sum_{i=1}^m \alpha_i &= 1, \\ \alpha_i &\geq 0. \end{aligned}$$

Definition 64. *Given a set $S \subseteq \mathbb{R}^n$, we call convex hull of S the set containing all convex combinations of the elements of S.*
Moreover, we indicate the convex hull of S with $\mathrm{conv}(S)$.

The following properties hold:

- the intersection of two convex sets is a convex set,

- a set $S \subseteq \mathbb{R}^n$ is convex if and only if every convex combination of the element of S belong to S,
- the convex hull of a set $S \subseteq \mathbb{R}^n$ is a convex set containing S,
- the convex hull of a set $S \subseteq \mathbb{R}^n$ is the intersection of all the convex sets containing S.

Now, we provide the following definitions of convex and concave functions.

Definition 65. *Let $S \subseteq \mathbb{R}^n$ be a convex set. A function $f\colon \mathbb{R}^n \to \mathbb{R}$ is said to be*

- *convex on S if, for all $x_1, x_2 \in S$,*

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda x_1 + (1-\lambda)x_2, \quad \forall \lambda \in [0,1];$$

- *strictly convex on S if, for all $x_1 \neq x_2 \in S$,*

$$f(\lambda x_1 + (1-\lambda)x_2) < \lambda x_1 + (1-\lambda)x_2, \quad \forall \lambda \in (0,1);$$

- *μ-strongly convex on S if there exists $\mu > 0$ such that, for all $x_1, x_2 \in S$,*

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2) - \frac{1}{2}\mu\lambda(1-\lambda)\|x_1 - x_2\|^2, \quad \forall \lambda \in [0,1].$$

Definition 66. *Let $S \subseteq \mathbb{R}^n$ be a convex set. A function $f\colon \mathbb{R}^n \to \mathbb{R}$ is said to be*

- *concave on S if, for all $x_1, x_2 \in S$,*

$$f(\lambda x_1 + (1-\lambda)x_2) \geq \lambda x_1 + (1-\lambda)x_2, \quad \forall \lambda \in [0,1].$$

- *strictly concave on S if, for all $x_1 \neq x_2 \in S$,*

$$f(\lambda x_1 + (1-\lambda)x_2) > \lambda x_1 + (1-\lambda)x_2, \quad \forall \lambda \in (0,1);$$

- *μ-strongly concave on S if there exists $\mu > 0$ such that, for all $x_1, x_2 \in S$,*

$$f(\lambda x_1 + (1-\lambda)x_2) \geq \lambda f(x_1) + (1-\lambda) f(x_2) + \frac{1}{2}\mu\lambda(1-\lambda)\|x_1 - x_2\|^2, \quad \forall \lambda \in [0,1].$$

It is straightforward to verify that a function f is convex (respectively, concave) if and only if $g(x) = -f(x)$ is concave (respectively, convex).

Moreover, from the above definitions, it is easy to see that, if a function f is strongly convex (respectively, strongly concave), then f is strictly convex (respectively, strictly concave); and if f is strictly convex (respectively, strictly concave), then f is convex (respectively, concave).

The following result holds.

Proposition 22. *Given $f\colon \mathbb{R}^n \to \mathbb{R}$ and an open convex set $S \subseteq \mathbb{R}^n$, let us assume that f is convex or concave on S. We have that*

- $f \in C(S)$,

- *if all partial derivatives of f on S exist, then $f \in C^1(S)$.*

Moreover, we can give the following characterizations of convexity and concavity.

Proposition 23. *Given $f\colon \mathbb{R}^n \to \mathbb{R}$, let us assume that $S \subseteq \mathbb{R}^n$ is an open convex set.*

- *If f is Fréchet differentiable on S, then*
  - *f is convex on S if and only if*
$$f(x_2) \geq f(x_1) + \nabla f(x_1)^T (x_2 - x_1), \quad \forall x_1, x_2 \in S;$$
  - *f is strictly convex on S if and only if*
$$f(x_2) > f(x_1) + \nabla f(x_1)^T (x_2 - x_1), \quad \forall x_1 \neq x_2 \in S;$$
  - *f is μ-strongly convex on S if and only if there exists $\mu > 0$ such that*
$$f(x_2) \geq f(x_1) + \nabla f(x_1)^T (x_2 - x_1) + \frac{1}{2}\mu \|x_2 - x_1\|^2, \quad \forall x_1, x_2 \in S;$$
  - *f is concave on S if and only if*
$$f(x_2) \leq f(x_1) + \nabla f(x_1)^T (x_2 - x_1), \quad \forall x_1, x_2 \in S;$$
  - *f is strictly concave on S if and only if*
$$f(x_2) < f(x_1) + \nabla f(x_1)^T (x_2 - x_1), \quad \forall x_1 \neq x_2 \in S;$$
  - *f is μ-strongly concave on S if and only if there exists $\mu > 0$ such that*
$$f(x_2) \leq f(x_1) + \nabla f(x_1)^T (x_2 - x_1) - \frac{1}{2}\mu \|x_2 - x_1\|^2, \quad \forall x_1, x_2 \in S.$$
- *If f is twice Fréchet differentiable on S, then*
  - *f is convex on S if and only if*
$$\nabla^2 f(x) \succeq 0, \quad \forall x \in S;$$
  - *f is strictly convex on S if*
$$\nabla^2 f(x) \succ 0, \quad \forall x \in S;$$
  - *f is μ-strongly convex on S if and only if there exists $\mu > 0$ such that*
$$\nabla^2 f(x) - \mu I \succeq 0, \quad \forall x \in S;$$
  - *f is concave on S if and only if*
$$\nabla^2 f(x) \preceq 0, \quad \forall x \in S;$$
  - *f is strictly concave on S if*
$$\nabla^2 f(x) \prec 0, \quad \forall x \in S;$$
  - *f is μ-strongly concave on S if and only if there exists $\mu > 0$ such that*
$$\nabla^2 f(x) + \mu I \preceq 0, \quad \forall x \in S.$$

A.2 Basics of continuous optimization

In this section, we recall some fundamental concepts of continuous optimization. In particular, in Subsection A.2.1 we provide basic definitions, in Subsection A.2.2 we focus on the conditions that guarantee the existence of optimal solutions and on stationarity and, finally, in Subsection A.2.3 we are concerned with the projection operator onto a closed convex set.

A.2.1 Definitions

In *continuous optimization*, the variables space is $\mathbb{R}^n$. Namely, we are concerned with the following optimization problem:

$$\begin{aligned} &\min_x f(x) \\ &x \in \mathcal{F} \subseteq \mathbb{R}^n. \end{aligned} \tag{A.2}$$

We provide the following definitions.

- f is called *objective function*.
- $\mathcal{F}$ is called *feasible set* (or *feasible region*).
- Every point $x \in \mathcal{F}$ is called *feasible solution*.
- The problem is said to be *unconstrained* if

  $$\mathcal{F} = X.$$

  Otherwise, the problem is said to be *constrained*.
- The problem is said to be *feasible* if

  $$\mathcal{F} \neq \emptyset.$$

  Otherwise, the problem is said to be *infeasible*.
- The problem is said to be *unbounded* if, for every $M > 0$, there exists a point $x \in \mathcal{F}$ such that $f(x) < -M$.
- A point $x^* \in \mathcal{F}$ such that

  $$f(x^*) \leq f(x), \quad \forall x \in \mathcal{F},$$

  is said to be a *global optimal solution* and the value $f(x^*)$ is called *optimum value*. Moreover, x^* is said to be a *strict global optimal solution* if

  $$f(x^*) < f(x), \quad \forall x \in \mathcal{F}.$$

- A point $x^* \in \mathcal{F}$ is said to be a *local optimal solution* if there exists $\rho > 0$ such that

  $$f(x^*) \leq f(x), \quad \forall x \in \mathcal{F} \cap \mathcal{B}(x^*, \rho).$$

  Moreover, x^* is said to be a *strict local optimal solution* if

  $$f(x^*) < f(x), \quad \forall x \in \mathcal{F} \cap \mathcal{B}(x^*, \rho).$$

- Given $\alpha \in \mathbb{R}$, every non-empty set defined as

$$L(\alpha) = \{x \in \mathcal{F} \colon f(x) \leq \alpha\}$$

  is called *level set*.

Usually, the feasible region $\mathcal{F}$ is defined by a set of inequality and equality constraints. Namely,

$$\mathcal{F} = \{x \in \mathbb{R}^n : \ g_i(x) \leq 0,\ i = 1, \ldots, m,\ h_j(x) = 0,\ j = 1, \ldots, p\},$$

where $g_i \colon \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$, and $h_j \colon \mathbb{R}^n \to \mathbb{R}$, $j = 1, \ldots, p$.

Given a point $x \in \mathbb{R}^n$, we say that a constraint g_i, $i = 1, \ldots, m$, is *active* at x if

$$g_i(x) = 0.$$

A.2.2 Existence of optimal solutions and their characterization

In this subsection, we first analyze some conditions that ensure the existence of optimal solutions for problem (A.2) and then we provide some characterizations of optimality, introducing the concept of *stationarity*.

A first condition that guarantees the existence of optimal solutions of (A.2) is the Weierstrass theorem (see Theorem 23). By exploiting that result, it is possible to obtain other conditions for the case where the feasible set is not compact. In particular, we state the following propositions.

Proposition 24. *Let us consider problem* (A.2) *and assume that f is continuous on $\mathcal{F}$. If there exists a compact level set, then there exists $x^* \in \mathcal{F}$ such that*

$$f(x^*) \leq f(x), \quad \forall x \in \mathcal{F}.$$

Proposition 25. *Let us consider problem* (A.2) *and assume that f is continuous on $\mathcal{F}$. Then, every level set is compact if and only if both the following conditions are satisfied:*

(i) for every sequence $\{x^k\} \subseteq \mathcal{F}$ such that $\lim_{k\to\infty} \|x^k\| = +\infty$, we have

$$\lim_{k\to\infty} f(x^k) = +\infty;$$

(ii) for every sequence $\{x^k\} \subseteq \mathcal{F}$ converging to $\bar{x} \notin \mathcal{F}$, we have

$$\lim_{k\to\infty} f(x^k) = +\infty.$$

When $\mathcal{F} = \mathbb{R}^n$, the hypotheses of Proposition 25 boil down to require f to be *coercive*. Formally, we give the following definition of *coercivity*.

Definition 67. *Given $f \colon \mathbb{R}^n \to \mathbb{R}$, we say that f is coercive if, for every sequence $\{x^k\} \subseteq \mathbb{R}^n$ such that $\lim_{k\to\infty} \|x^k\| = +\infty$, we have*

$$\lim_{k\to\infty} f(x^k) = +\infty.$$

It is worth mentioning that a quadratic function $f(x) = \frac{1}{2}x^TQx + c^Tx$, with $Q \in \mathbb{R}^{n\times n}$, Q symmetric and $c \in \mathbb{R}^n$, is coercive if and only if $Q \succ 0$.

Now, we need to provide the definitions of *feasible directions*, *descent directions* and *ascent directions*.

Definition 68. *Given $\mathcal{F} \subseteq \mathbb{R}^n$ and $\bar{x} \in \mathcal{F}$, we say that $d \in \mathbb{R}^n$ is a feasible direction for $\mathcal{F}$ at x if there exists $\bar{\alpha} > 0$ such that*

$$\bar{x} + \alpha d \in \mathcal{F}, \quad \forall \alpha \in [0, \bar{\alpha}].$$

Definition 69. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}$ and a point $x \in \mathbb{R}^n$, we say that*

- *$d \in \mathbb{R}^n$ is a descent direction for f at x if there exists $\bar{\alpha} > 0$ such that*

$$f(x + \alpha d) < f(x), \quad \forall \alpha \in (0, \bar{\alpha}];$$

- *$d \in \mathbb{R}^n$ is an ascent direction for f at x if there exists $\bar{\alpha} > 0$ such that*

$$f(x + \alpha d) > f(x), \quad \forall \alpha \in (0, \bar{\alpha}].$$

Descent directions play a crucial role in defining both optimality conditions and algorithms for minimization problems. We show how descent directions can be recognized and employed to characterize optimal solutions (in the following, we only focus on descent directions, since we are concerned with minimization problems, but a characterization of ascent directions can be provided as well).

Proposition 26. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}$ and a point $x \in \mathbb{R}^n$, let us assume that f is continuously differentiable on a neighborhood of x. We have that*

- *$d \in \mathbb{R}^n$ is a descent direction for f at x if*

$$\nabla f(x)^T d < 0; \tag{A.3}$$

- *if f is convex on a neighborhood of x and $d \in \mathbb{R}^n$ is a descent direction, then* (A.3) *holds.*

Assuming that the objective function is twice continuously differentiable, we can also define *negative curvature directions* and *positive curvature directions* as follows.

Definition 70. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}$ and a point $x \in \mathbb{R}^n$, let us assume that f is twice continuously differentiable on a neighborhood of x. We say that*

- *$d \in \mathbb{R}^n$ is a negative curvature direction for f at x if*

$$d^T \nabla^2 f(x)\, d < 0,$$

- *$d \in \mathbb{R}^n$ is a positive curvature direction for f at x if*

$$d^T \nabla^2 f(x)\, d > 0.$$

Moreover, the following result holds.

Proposition 27. *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}$ and a point $x \in \mathbb{R}^n$, let us assume that f is twice continuously differentiable on a neighborhood of x. If $d \in \mathbb{R}^n$ is such that $\nabla f(x)^T d = 0$ and $d^T \nabla^2 f(x)\, d < 0$, then d is a descent direction for f at x.*

Now, we can give the following necessary optimality conditions, which state that there do not exist feasible descent directions at a local optimal solution.

Proposition 28. *Let us consider the following problem:*

$$\begin{aligned} &\min_x f(x) \\ &x \in \mathcal{F} \subseteq \mathbb{R}^n \end{aligned}$$

and assume that x^* is a local optimal solution.

We have that

- *if f is continuously differentiable on a neighborhood of x^*, then*

$$\nabla f(x^*)^T d \geq 0, \quad \text{for every feasible direction } d \text{ at } x^*;$$

- *if f is twice continuously differentiable on a neighborhood of x^*, then every feasible direction $d \in \mathbb{R}^n$ at x^* such that $\nabla f(x^*)^T d = 0$ satisfies*

$$d^T \nabla^2 f(x^*) d \geq 0.$$

Now, we provide other characterizations of local optimal solutions by considering separately the unconstrained and the constrained case.

First, we focus on an unconstrained problem, that is,

$$\begin{aligned} &\min_x f(x) \\ &x \in \mathbb{R}^n, \end{aligned} \tag{A.4}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$.

A well-known characterization of local optimal solutions of problem (A.4) is reported in the following theorem.

Theorem 40. *Let us consider problem* (A.4) *and assume that $x^* \in \mathbb{R}^n$ is a local optimal solution. If f is continuously differentiable on a neighborhood of x^*, then*

$$\nabla f(x^*) = 0. \tag{A.5}$$

We can also provide the following second-order necessary optimality conditions for problem (A.4).

Theorem 41. *Let us consider problem* (A.4) *and assume that $x^* \in \mathbb{R}^n$ is a local optimal solution. If f is twice continuously differentiable on a neighborhood of x^*, then*

$$\nabla f(x^*) = 0, \tag{A.6}$$

$$\nabla^2 f(x^*) \succeq 0. \tag{A.7}$$

Now, we are ready to give the definition of *stationary point* for an unconstrained problem.

Definition 71. *Let us consider problem* (A.4)*. A point $x^* \in \mathbb{R}^n$ is said to be*

- *a (first-order) stationary point if it satisfies* (A.5)*,*
- *a second-order stationary point if it satisfies* (A.6)–(A.7)*.*

In the following theorem, we provide sufficient optimality conditions for an unconstrained problem.

Theorem 42. *Let us consider problem* (A.4) *and assume that $f \in C^2(\mathbb{R}^n)$. If there exist $x^* \in \mathbb{R}^n$ and $\rho > 0$ such that*

$$\begin{aligned} \nabla f(x^*) &= 0, \\ \nabla^2 f(x) &\succeq 0, \quad \forall x \in \mathcal{B}(x^*, \rho), \end{aligned}$$

then x^* is a local optimal solution.

Moreover, if $\nabla^2 f(x^*) \succ 0$, then x^* is a strict local optimal solution.

For what concerns constrained optimization, let us consider a problem of the following form:

$$\begin{aligned} &\min_x f(x) \\ &g_i(x) \leq 0, \quad i = 1, \dots, m, \\ &h_j(x) = 0, \quad j = 1, \dots, p, \end{aligned} \tag{A.8}$$

where $f \colon \mathbb{R}^n \to \mathbb{R}$, $g_i \colon \mathbb{R}^n \to \mathbb{R}$, $i = 1, \dots, m$, and $h_j \colon \mathbb{R}^n \to \mathbb{R}$, $j = 1, \dots, p$.

We can introduce the *Lagrangian function* $\mathcal{L} \colon \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ for problem (A.8), defined as

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{j=1}^{p} \mu_j h_j(x).$$

Moreover, we use the following notation to indicate the set of inequality constraints that are active at a feasible point x:

$$I(x) = \{i \in \{1, \dots, m\} : g_i(x) = 0\}.$$

Now, we report some necessary optimality conditions for problem (A.8), known as *Karush-Kuhn-Tucker* (*KKT*) conditions. They require that a *constraint qualification* is satisfied, as shown in the next theorem (where only some of the most popular constraint qualifications are considered).

Theorem 43. *Let us consider problem* (A.8) *and assume that $x^* \in \mathbb{R}^n$ is a local optimal solution. If f, g_i, $i = 1, \dots, m$, and h_j, $j = 1, \dots, p$, are continuously*

differentiable on a neighborhood of x^**, then there exist* $\lambda_1^*, \dots, \lambda_m^*, \mu_1^*, \dots, \mu_p^* \in \mathbb{R}$ *(called KKT multipliers) such that*

$$\nabla_x \mathcal{L}(x^*, \lambda^*, \mu^*) = \nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^{p} \mu_j^* \nabla h_j(x^*) = 0, \tag{A.9}$$

$$\lambda_i^* \, g_i(x^*) = 0, \qquad i = 1, \dots, m, \tag{A.10}$$

$$\lambda_i^* \geq 0, \qquad i = 1, \dots, m, \tag{A.11}$$

if one of the following constraint qualifications holds:

- h_j*,* $j = 1, \dots, p$*, are affine functions and* g_i*,* $i = 1, \dots, m$*, are concave functions;*
- *there do not exist* $\alpha_i \geq 0$*,* $i \in I(x^*)$*,* β_j*,* $j = 1, \dots, p$*, that are not all zero and such that*
$$\nabla f(x^*) + \sum_{i \in I(x^*)} \alpha_i \nabla g_i(x^*) + \sum_{j=1}^{p} \beta_j \nabla h_j(x^*) = 0;$$
- *the vectors* $\nabla g_i(x^*)$*,* $i \in I(x^*)$*,* $\nabla h_j(x^*)$*,* $j = 1, \dots, p$*, are linearly independent (in this case,* $\lambda_1^*, \dots, \lambda_m^*, \mu_1^*, \dots, \mu_p^*$ *are unique);*
- h_j*,* $j = 1, \dots, p$*, are affine functions,* g_i*,* $i = 1, \dots, m$*, are convex functions and there exists a feasible point* $\bar{x}$ *such that*
$$g_i(\bar{x}) < 0, \quad \forall i \in I(x^*).$$

We can also provide the following second-order necessary optimality conditions, which require linear independence of the gradients of the active constraints.

Theorem 44. *Let us consider problem* (A.8) *and assume that* $x^* \in \mathbb{R}^n$ *is a local optimal solution. If* f*,* g_i*,* $i = 1, \dots, m$*, and* h_j*,* $j = 1, \dots, p$*, are twice continuously differentiable on a neighborhood of* x^* *and the vectors* $\nabla g_i(x^*)$*,* $i \in I(x^*)$*,* $\nabla h_j(x^*)$*,* $j = 1, \dots, p$*, are linearly independent, then there exist and are unique* $\lambda_1^*, \dots, \lambda_m^*, \mu_1^*, \dots, \mu_p^* \in \mathbb{R}$ *such that*

$$\nabla_x \mathcal{L}(x^*, \lambda^*, \mu^*) = \nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^{p} \mu_j^* \nabla h_j(x^*) = 0, \tag{A.12}$$

$$\lambda_i^* \, g_i(x^*) = 0, \qquad i = 1, \dots, m, \tag{A.13}$$

$$\lambda_i^* \geq 0, \qquad i = 1, \dots, m, \tag{A.14}$$

$$d^T \nabla_{xx}^2 \mathcal{L}(x^*, \lambda^*, \mu^*) d \geq 0, \quad \forall d \in T(x^*, \lambda^*), \tag{A.15}$$

where

$$\begin{aligned} T(x^*, \lambda^*) = \{ d \in \mathbb{R}^n \colon\ & d^T \nabla g_i(x^*) = 0,\ i \in I(x^*) \text{ and } \lambda_i^* > 0, \\ & d^T \nabla g_i(x^*) \leq 0,\ i \in I(x^*) \text{ and } \lambda_i^* = 0, \\ & d^T \nabla h_j(x^*) = 0,\ j = 1, \dots, p \}. \end{aligned}$$

Now, we are ready to give the definition of *stationary point* for a constrained problem.

Definition 72. *Let us consider problem* (A.8)*. A feasible point x^* is said to be*

- *a (first-order) stationary point if it satisfies* (A.9)–(A.11)*,*
- *a second-order stationary point if it satisfies* (A.12)–(A.15)*.*

In the following theorem, we provide sufficient optimality conditions for problem (A.8) (which do not require any constraint qualification).

Theorem 45. *Let us consider problem* (A.8) *and assume that f, g_i, $i = 1, \dots, m$, h_j, $j = 1, \dots, p$, are twice continuously differentiable on an open set containing the feasible set. If there exist a feasible point $x^* \in \mathbb{R}^n$ and $\lambda_1^*, \dots, \lambda_m^*, \mu_1^*, \dots, \mu_p^* \in \mathbb{R}$ such that*

$$\begin{aligned}
&\nabla_x \mathcal{L}(x^*, \lambda^*, \mu^*) = \nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^{p} \mu_j^* \nabla h_j(x^*) = 0 \\
&\lambda_i^* \, g_i(x^*) = 0, \qquad i = 1, \dots, m, \\
&\lambda_i^* \geq 0, \qquad i = 1, \dots, m, \\
&d^T \nabla_{xx}^2 \mathcal{L}(x^*, \lambda^*, \mu^*) d > 0, \quad \forall d \in T(x^*, \lambda^*), d \neq 0,
\end{aligned}$$

where

$$\begin{aligned}
T(x^*, \lambda^*) = \{d \in \mathbb{R}^n \colon\ & d^T \nabla g_i(x^*) = 0,\ i \in I(x^*) \text{ and } \lambda_i^* > 0, \\
& d^T \nabla g_i(x^*) \leq 0,\ i \in I(x^*) \text{ and } \lambda_i^* = 0, \\
& d^T \nabla h_j(x^*) = 0,\ j = 1, \dots, p\},
\end{aligned}$$

then x^* is a strict local optimal solution.

We also provide the following definition of *strict complementarity* for a stationary point of a constrained problem.

Definition 73. *Let us consider problem* (A.8)*. Let x^* be a stationary point and let $\lambda^* \in \mathbb{R}^m$, $\mu^* \in \mathbb{R}^p$ be the corresponding KKT multiplier vectors. We say that strict complementarity holds at x^* if*

$$\lambda_i^* > 0, \quad \forall i \in \{1, \dots, m\} \colon g_i(x^*) = 0.$$

Now, we recall some useful properties of convex and concave problems. We first need to give the following definitions.

Definition 74. *A minimization problem*

$$\begin{aligned}
&\min_x f(x) \\
&x \in \mathcal{F} \subseteq \mathbb{R}^n
\end{aligned}$$

is said to be convex if $\mathcal{F}$ is a convex set and f is a convex function on $\mathcal{F}$.

A maximization problem

$$\begin{aligned}
&\max_x f(x) \\
&x \in \mathcal{F} \subseteq \mathbb{R}^n
\end{aligned}$$

is said to be convex if $\mathcal{F}$ is a convex set and f is a concave function on $\mathcal{F}$.

Definition 75. *A minimization problem*

$$\min_x f(x)$$
$$x \in \mathcal{F} \subseteq \mathbb{R}^n$$

is said to be concave if $\mathcal{F}$ is a convex set and f is a concave function on $\mathcal{F}$.

A maximization problem

$$\max_x f(x)$$
$$x \in \mathcal{F} \subseteq \mathbb{R}^n$$

is said to be concave if $\mathcal{F}$ is a convex set and f is a convex function on $\mathcal{F}$.

A fundamental property of convex problems is that every local optimal solution is also a global optimal solution. Moreover, stationarity conditions are necessary and sufficient for global optimality. These results are summarized in the following proposition.

Proposition 29. *Let us consider the following problem:*

$$\min_x f(x)$$
$$x \in \mathcal{F} \subseteq \mathbb{R}^n,$$

where $\mathcal{F}$ is a convex set and f is convex function on $\mathcal{F}$.

We have that

- *every local optimal solution is a global optimal solution;*
- *if f is strictly convex, then the problem attains at most one global optimal solution;*
- *every stationary point is a global optimal solution.*

For what concerns concave problems, the following result ensures that there do exist local optimal solutions in the interior of the feasible set $\mathcal{F}$ if the objective function is not constant on $\mathcal{F}$.

Proposition 30. *Let us consider the following problem:*

$$\min_x f(x)$$
$$x \in \mathcal{F} \subseteq \mathbb{R}^n,$$

where $\mathcal{F}$ is a convex set and f is a concave function on $\mathcal{F}$.

If f is not constant on $\mathcal{F}$, then every local optimal solution is on the boundary of $\mathcal{F}$.

Now, we provide further optimality conditions for problems with a convex feasible set, extending those reported in Proposition 28. First, we state the following result.

Proposition 31. *Let $\mathcal{F} \subseteq \mathbb{R}^n$ be a convex set and let $\bar{x} \in \mathcal{F}$. If $\mathcal{F} \neq \{\bar{x}\}$, then the direction $x - \bar{x}$, with $x \in \mathcal{F}$, $x \neq \bar{x}$, is a feasible direction for $\mathcal{F}$ at $\bar{x}$. In particular, we have*

$$\bar{x} + \alpha(x - \bar{x}) \in \mathcal{F}, \quad \forall \alpha \in [0, 1].$$

We can provide the following necessary optimality conditions.

Proposition 32. *Let us consider the following problem:*

$$\begin{aligned} &\min_x f(x) \\ &x \in \mathcal{F} \subseteq \mathbb{R}^n, \end{aligned}$$

where $\mathcal{F}$ is a convex set. Let us assume that x^* is a local optimal solution.

We have that

- *if f is continuously differentiable on a neighborhood of x^*, then*

$$\nabla f(x^*)^T (x - x^*) \geq 0, \quad \forall x \in \mathcal{F};$$

- *if f is twice continuously differentiable on a neighborhood of x^*, then*

$$(x - x^*)^T \nabla^2 f(x^*)(x - x^*) \geq 0, \quad \forall x \in \mathcal{F} \colon \nabla f(x^*)^T (x - x^*) = 0.$$

When the objective function is convex, we can state necessary and sufficient conditions for global optimality.

Proposition 33. *Let us consider the following problem:*

$$\begin{aligned} &\min_x f(x) \\ &x \in \mathcal{F} \subseteq \mathbb{R}^n, \end{aligned}$$

where $\mathcal{F}$ is a convex set and f is a convex and continuously differentiable function on an open set containing $\mathcal{F}$. Then, x^* is a global optimal solution if and only if

$$\nabla f(x^*)^T (x - x^*) \geq 0, \quad \forall x \in \mathcal{F}.$$

Finally, we give the following sufficient conditions to ensure that the feasible region of a constrained problem is a closed convex set.

Proposition 34. *Let $\mathcal{F} \subseteq \mathbb{R}^n$ be defined as*

$$\mathcal{F} = \{x \in \mathbb{R}^n : \ g_i(x) \leq 0, \, i = 1, \ldots, m, \ h_j(x) = 0, \, j = 1, \ldots, p\},$$

where $g_i \colon \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$, are convex functions and $h_j \colon \mathbb{R}^n \to \mathbb{R}$, $j = 1, \ldots, p$, are affine functions.

Then, $\mathcal{F}$ is a closed convex set.

A.2.3 Projection operator onto a closed convex set

In this subsection, we first define the *projection operator* onto a closed convex set and then we recall some useful properties.

Definition 76. *Let $\mathcal{F} \subseteq \mathbb{R}^n$ be a non-empty closed convex set. Given $x \in \mathbb{R}^n$, we call projection of x onto $\mathcal{F}$ the optimal solution $p(x)$ of the following problem:*

$$\begin{aligned} &\min_y \ \|x - y\| \\ &y \in \mathcal{F}. \end{aligned}$$

Exploiting the results provided above for coercive and strictly convex functions, it follows that projecting a vector onto a non-empty closed convex set consists in solving a convex optimization problem which attains one and only one optimal solution.

The projection operator can be employed to provide a further characterization of optimal solutions for problems with a closed convex feasible set, as stated in the next proposition.

Proposition 35. *Let us consider the following problem:*

$$\begin{aligned} &\min_x f(x) \\ &x \in \mathcal{F} \subseteq \mathbb{R}^n, \end{aligned}$$

where $\mathcal{F}$ is a closed convex set. Let $\bar{x} \in \mathcal{F}$ be such that f is continuously differentiable on a neighborhood of $\bar{x}$. Then

$$\bar{x} - p(\bar{x} - \alpha \nabla f(\bar{x})) = 0 \quad \text{if and only if} \quad \nabla f(\bar{x})^T (x - \bar{x}) \geq 0, \quad \forall x \in \mathcal{F},$$

where $p(\cdot)$ is the projection operator onto $\mathcal{F}$ and α is any positive real number.

Hence, recalling Proposition 32 and 33, we can state the following results.

Proposition 36. *Let us consider the following problem:*

$$\begin{aligned} &\min_x f(x) \\ &x \in \mathcal{F} \subseteq \mathbb{R}^n, \end{aligned}$$

where $\mathcal{F}$ is a closed convex set. Let us assume that x^* is a local optimal solution. If f is continuously differentiable on a neighborhood of x^*, then

$$x^* - p(x^* - \alpha \nabla f(x^*)) = 0,$$

where $p(\cdot)$ is the projection operator onto $\mathcal{F}$ and α is any positive real number.

Proposition 37. *Let us consider the following problem:*

$$\begin{aligned} &\min_x f(x) \\ &x \in \mathcal{F} \subseteq \mathbb{R}^n, \end{aligned}$$

where $\mathcal{F}$ is a non-empty closed convex set and f is a convex and continuously differentiable function on an open set containing $\mathcal{F}$. Then, x^* is a global optimal solution if and only if

$$x^* - p(x^* - \alpha \nabla f(x^*)) = 0,$$

where $p(\cdot)$ is the projection operator onto $\mathcal{F}$ and α is any positive real number.

Finally, it is worth to consider the case where $\mathcal{F} \subseteq \mathbb{R}^n$ is a set of the following form:

$$\mathcal{F} = \{x \in \mathbb{R}^n \colon l_i \leq x_i \leq u_i,\ i = 1, \ldots, n\},$$

with $l_i \leq u_i$, $i = 1, \ldots, n$. In this case, it is easy to verify that the projection $p(x)$ of a point $x \in \mathbb{R}^n$ onto $\mathcal{F}$ has components $p_i(x)$, $i = 1, \ldots, n$, defined as

$$p_i(x) = \begin{cases} l_i, & \text{if } x_i < l_i, \\ x_i, & \text{if } l_i \leq x_i \leq u_i, \\ u_i, & \text{if } x_i > u_i, \end{cases}$$

or equivalently,

$$p_i(x) = \max\{l_i, \min\{x_i, u_i\}\}.$$

Appendix B

Numerical results of ASA-BCP

In this appendix, we report the details of the numerical experience described in Section 3.4.

In particular, in Section B.1 we report the numerical results related to the comparison between ASA-BCP and NMBC, and in Section B.2 we report the numerical results related to the comparison among ASA-BCP, ALGENCAN and LANCELOT B.

B.1 Comparison with NMBC

In Table B.1, we report the numerical results obtained by ASA-BCP and NMBC on 140 problems from the CUTEst collection.

In Table B.2, we report the 43 problems that have been considered in the performance profiles between ASA-BCP and NMBC (Figure 3.1). In particular, these problems have been solved in more than 1 second by at least one algorithm and both algorithms have found the same stationary point (with a tolerance of 10^{-3}). To be more specific, let f_{A} and f_{N} be the objective function values of the stationary points found, respectively, by ASA-BCP and NMBC when applied to a particular problem. Let $f_{\min} = \min\{f_{\mathrm{A}}, f_{\mathrm{N}}\}$. We consider that ASA-BCP and NMBC have found the same stationary point if

$$\frac{|f_{\mathrm{A}} - f_{\min}|}{\max\{1, f_{\min}\}} < 10^{-3} \quad \text{and} \quad \frac{|f_{\mathrm{N}} - f_{\min}|}{\max\{1, f_{\min}\}} < 10^{-3}.$$

In Table B.3, we report the problems that have been solved in more than 1 second by ASA-BCP or NMBC and such that both algorithms have found different stationary points (with a tolerance of 10^{-3}).

All the tables include the following data: the name (Problem) and the dimension (n) of the problems considered, the objective function value of the stationary point found (obj), the number of function evaluations (f-eval), the total number of conjugate gradient iterations (cg-it), the computational time in seconds (time).

Table B.1. Comparison between ASA-BCP and NMBC on 140 problems from the CUTEst collection.

Problem	n	**ASA-BCP**				**NMBC**			
		obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)
BDEXP	1000	3.953e-04	11	10	0.0	3.940e-04	3	20	0.0
BDEXP	5000	1.967e-03	11	10	0.0	1.966e-03	3	20	0.1
BIGGSB1	1000	1.500e-02	198	2272	0.1	1.500e-02	5	1288	0.1
BIGGSB1	5000	1.595e-02	199	2442	0.6	1.500e-02	6	3974	0.9
BIGGSB1	10000	1.595e-02	199	2442	1.3	1.500e-02	6	6598	2.9
BIGGSB1	20000	1.595e-02	199	2442	2.5	1.500e-02	7	19025	16.6
BQPGAUSS	2003	-3.626e-01	9	9040	1.8	-3.626e-01	60	7674	1.6
CHARDIS0	2000	6.384e-22	2	1	0.1	2.260e-19	2	2	0.1
CHARDIS0	4000	4.648e-20	2	1	0.5	4.398e-18	2	2	0.5
CHARDIS0	10000	1.246e-19	2	1	2.8	1.526e-16	2	2	3.3
CHENHARK	1000	-2.000e+00	107	1953	0.1	-2.000e+00	107	56216	2.2
CHENHARK	5000	-2.000e+00	73	1556	0.4	-2.000e+00	89	129087	23.0
CHENHARK	10000	-2.000e+00	108	1935	0.9	-2.000e+00	117	278775	105.7
CHENHARK	50000	-2.000e+00	137	2643	4.5	-2.000e+00	61	207468	348.6
CVXBQP1	1000	2.252e+04	10	9	0.0	2.252e+04	4	16	0.0
CVXBQP1	10000	2.250e+06	13	11	0.0	2.250e+06	4	19	0.0
CVXBQP1	100000	2.250e+08	15	12	0.2	2.250e+08	4	22	0.2
EXPLIN	1200	-7.193e+07	108	279	0.0	-7.193e+07	119	279	0.0
EXPLIN2	1200	-7.200e+07	41	69	0.0	-7.200e+07	106	205	0.0
EXPQUAD	1200	-3.685e+09	146	225	0.0	-3.685e+09	106	320	0.1
JNLBRNG1	1024	-1.803e-01	4	135	0.0	-1.803e-01	3	171	0.1
JNLBRNG1	1156	-1.803e-01	3	128	0.0	-1.803e-01	3	184	0.1
JNLBRNG1	5625	-1.805e-01	4	278	0.4	-1.805e-01	8	483	0.8
JNLBRNG1	10000	-1.806e-01	5	391	1.1	-1.806e-01	11	699	1.9

continued on next page

Table B.1 – continued from previous page

Problem	n	**ASA-BCP**				**NMBC**			
		obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)
JNLBRNG1	15625	-1.806e-01	5	429	1.6	-1.806e-01	13	955	3.7
JNLBRNG2	1024	-4.125e+00	3	158	0.0	-4.125e+00	3	167	0.1
JNLBRNG2	1156	-4.128e+00	3	191	0.1	-4.128e+00	3	146	0.1
JNLBRNG2	5625	-4.147e+00	5	407	0.5	-4.147e+00	6	441	0.7
JNLBRNG2	10000	-4.149e+00	6	527	1.3	-4.149e+00	8	752	1.9
JNLBRNG2	15625	-4.150e+00	7	659	2.3	-4.150e+00	10	969	3.4
JNLBRNGA	1024	-2.954e-01	3	99	0.0	-2.954e-01	3	125	0.0
JNLBRNGA	1156	-2.935e-01	3	120	0.0	-2.935e-01	3	142	0.1
JNLBRNGA	5625	-2.753e-01	5	340	0.5	-2.753e-01	8	405	0.6
JNLBRNGA	10000	-2.711e-01	6	418	1.1	-2.711e-01	11	650	1.7
JNLBRNGA	15625	-2.685e-01	5	446	1.8	-2.685e-01	13	907	3.4
JNLBRNGB	1024	-6.440e+00	4	596	0.1	-6.440e+00	4	449	0.1
JNLBRNGB	1156	-6.429e+00	4	593	0.1	-6.429e+00	3	524	0.1
JNLBRNGB	5625	-6.330e+00	6	1463	1.7	-6.330e+00	4	1388	1.7
JNLBRNGB	10000	-6.301e+00	7	1962	3.7	-6.301e+00	4	2333	4.6
JNLBRNGB	15625	-6.281e+00	8	2361	7.1	-6.281e+00	7	2902	8.9
LINVERSE	1999	6.810e+02	21	241	0.1	6.810e+02	31	422	0.2
MCCORMCK	1000	-9.137e+02	7	20	0.0	-9.137e+02	38	1019	0.2
MCCORMCK	5000	-4.567e+03	13	22	0.0	-4.567e+03	39	765	0.8
MCCORMCK	10000	-9.133e+03	12	16	0.1	-9.133e+03	38	547	1.1
MCCORMCK	50000	-4.566e+04	18	29	0.5	-4.566e+04	37	255	2.5
MINSURFO	2706	2.515e+00	5	274	0.3	2.515e+00	26	2390	2.6
MINSURFO	5931	2.485e+00	105	647	1.8	2.485e+00	27	1994	4.2
NCVXBQP1	1000	-1.987e+08	10	9	0.0	-1.987e+08	10	10	0.0
NCVXBQP1	10000	-1.986e+10	11	10	0.0	-1.986e+10	11	12	0.0

continued on next page

Table B.1 – continued from previous page

Problem	n	**ASA-BCP**				**NMBC**			
		obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)
NCVXBQP1	100000	-1.985e+12	15	11	0.2	-1.985e+12	11	12	0.2
NCVXBQP2	1000	-1.334e+08	15	24	0.0	-1.334e+08	13	28	0.0
NCVXBQP2	10000	-1.334e+10	31	63	0.1	-1.334e+10	30	70	0.1
NCVXBQP2	100000	-1.334e+12	41	100	0.7	-1.334e+12	39	101	0.7
NCVXBQP3	1000	-6.530e+07	20	34	0.0	-6.530e+07	20	34	0.0
NCVXBQP3	10000	-6.506e+09	54	129	0.1	-6.506e+09	6164	15326	9.8
NCVXBQP3	100000	-6.505e+11	287	1024	6.6	-6.505e+11	679	3076	16.7
NOBNDTOR	5476	-4.499e-01	4	235	0.4	-4.499e-01	10	492	0.8
NOBNDTOR	10000	-4.438e-01	5	318	0.9	-4.438e-01	16	763	2.3
NOBNDTOR	14884	-4.405e-01	6	396	1.7	-4.405e-01	19	914	3.8
NOBNDTOR	40000	-4.347e-01	9	659	7.6	-4.347e-01	34	1981	20.9
NONSCOMP	1000	8.661e-16	7	37	0.0	1.869e-12	3	29	0.0
NONSCOMP	5000	5.054e-13	7	34	0.0	1.038e-19	3	42	0.0
NONSCOMP	10000	1.680e-12	7	37	0.0	7.380e-19	3	41	0.0
OBSTCLAE	5625	1.863e+00	9	221	0.4	1.863e+00	3	293	0.4
OBSTCLAE	10000	1.886e+00	23	438	1.6	1.886e+00	5	384	0.9
OBSTCLAE	15625	1.901e+00	16	406	1.9	1.901e+00	9	541	1.8
OBSTCLAL	5625	1.863e+00	4	124	0.2	1.863e+00	3	215	0.2
OBSTCLAL	10000	1.886e+00	4	159	0.3	1.886e+00	5	317	0.6
OBSTCLAL	15625	1.901e+00	4	211	0.7	1.901e+00	8	433	1.2
OBSTCLBL	5625	7.231e+00	4	116	0.2	7.231e+00	5	183	0.3
OBSTCLBL	10000	7.272e+00	4	155	0.5	7.272e+00	6	273	0.7
OBSTCLBL	15625	7.296e+00	4	201	0.9	7.296e+00	10	370	1.6
OBSTCLBM	5625	7.231e+00	3	88	0.1	7.231e+00	5	167	0.3
OBSTCLBM	10000	7.272e+00	4	119	0.4	7.272e+00	5	299	0.8

continued on next page

Table B.1 – continued from previous page

Problem	n	**ASA-BCP**				**NMBC**			
		obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)
OBSTCLBM	15625	7.296e+00	4	164	0.8	7.296e+00	4	157	0.7
OBSTCLBU	5625	7.231e+00	5	211	0.4	7.231e+00	4	266	0.4
OBSTCLBU	10000	7.272e+00	4	163	0.5	7.272e+00	7	293	0.7
OBSTCLBU	15625	7.296e+00	5	231	1.1	7.296e+00	10	380	1.5
ODNAMUR	11130	9.237e+03	722	40213	24.5	9.237e+03	842	58579	41.7
PENTDI	1000	-7.500e-01	3	11	0.0	-7.500e-01	3	14	0.0
PENTDI	5000	-7.500e-01	3	11	0.0	-7.500e-01	3	14	0.1
PENTDI	10000	-7.500e-01	3	11	0.0	-7.500e-01	3	14	0.2
PENTDI	50000	-7.500e-01	3	11	0.1	-7.500e-01	3	14	0.8
POWELLBC	1000	3.103e+05	483	4169	59.3	3.103e+05	287	3328	48.7
QRTQUAD	1200	-3.685e+09	118	638	0.1	-3.685e+09	201	660	0.1
QRTQUAD	5000	-2.649e+11	716	3078	1.1	-2.649e+11	1024	3908	1.8
QUDLIN	1200	-7.200e+07	4	1	0.0	-7.200e+07	4	2	0.0
QUDLIN	5000	-1.250e+09	7	2	0.0	-1.250e+09	7	4	0.0
QUDLIN	10000	-5.000e+09	8	2	0.0	-5.000e+09	8	4	0.0
SCOND1LS	1002	2.967e-04	3657	944441	57.3	8.954e-04	4850	2576566	184.4
SCOND1LS	5002	1.278e-02	1874	451965	133.3	2.032e-02	1051	995227	317.3
SINEALI	1000	-9.990e+04	16	44	0.0	-9.990e+04	14	83	0.0
TORSION1	1024	-4.450e-01	3	77	0.0	-4.450e-01	5	113	0.0
TORSION1	5476	-4.303e-01	4	213	0.3	-4.303e-01	15	333	0.4
TORSION1	10000	-4.273e-01	4	265	0.6	-4.273e-01	20	556	1.3
TORSION1	14884	-4.257e-01	5	338	1.2	-4.257e-01	27	753	2.4
TORSION2	1024	-4.450e-01	3	67	0.0	-4.450e-01	5	123	0.0
TORSION2	5476	-4.303e-01	4	151	0.3	-4.303e-01	10	334	0.5
TORSION2	10000	-4.273e-01	4	264	0.8	-4.273e-01	12	466	1.1

continued on next page

Table B.1 – continued from previous page

Problem	n	**ASA-BCP**				**NMBC**			
		obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)
TORSION2	14884	-4.257e-01	4	260	1.1	-4.257e-01	15	616	2.0
TORSION3	1024	-1.232e+00	3	41	0.0	-1.232e+00	4	37	0.0
TORSION3	5476	-1.217e+00	3	84	0.1	-1.217e+00	6	127	0.1
TORSION3	10000	-1.214e+00	4	143	0.3	-1.214e+00	7	205	0.3
TORSION3	14884	-1.212e+00	4	156	0.4	-1.212e+00	9	245	0.6
TORSION4	1024	-1.232e+00	3	37	0.0	-1.232e+00	4	52	0.0
TORSION4	5476	-1.217e+00	3	102	0.1	-1.217e+00	6	165	0.2
TORSION4	10000	-1.214e+00	4	131	0.3	-1.214e+00	7	249	0.5
TORSION4	14884	-1.212e+00	5	221	0.8	-1.212e+00	10	308	0.9
TORSION5	1024	-2.876e+00	3	15	0.0	-2.876e+00	3	21	0.0
TORSION5	5476	-2.863e+00	3	39	0.0	-2.863e+00	4	53	0.0
TORSION5	10000	-2.860e+00	3	62	0.1	-2.860e+00	4	79	0.1
TORSION5	14884	-2.859e+00	3	68	0.2	-2.859e+00	5	92	0.2
TORSION6	1024	-2.876e+00	3	20	0.0	-2.876e+00	3	36	0.0
TORSION6	5476	-2.863e+00	3	47	0.0	-2.863e+00	4	91	0.1
TORSION6	10000	-2.860e+00	3	65	0.1	-2.860e+00	4	130	0.3
TORSION6	14884	-2.859e+00	4	116	0.3	-2.859e+00	5	173	0.5
TORSIONA	1024	-4.174e-01	3	78	0.0	-4.174e-01	5	115	0.0
TORSIONA	5476	-4.183e-01	4	222	0.3	-4.183e-01	15	330	0.4
TORSIONA	10000	-4.184e-01	5	290	0.8	-4.184e-01	21	536	1.2
TORSIONA	5476	-4.183e-01	4	222	0.3	-4.183e-01	15	330	0.4
TORSIONB	1024	-4.174e-01	3	78	0.0	-4.174e-01	4	100	0.0
TORSIONB	5476	-4.183e-01	4	164	0.3	-4.183e-01	9	318	0.5
TORSIONB	10000	-4.184e-01	5	261	0.8	-4.184e-01	10	467	1.2
TORSIONB	14884	-4.184e-01	5	284	1.2	-4.184e-01	13	615	2.2

continued on next page

Table B.1 – continued from previous page

Problem	n	**ASA-BCP**				**NMBC**			
		obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)
TORSIONC	1024	-1.202e+00	3	42	0.0	-1.202e+00	4	33	0.0
TORSIONC	5476	-1.204e+00	3	84	0.1	-1.204e+00	5	127	0.1
TORSIONC	10000	-1.204e+00	4	139	0.3	-1.204e+00	7	205	0.3
TORSIONC	14884	-1.204e+00	4	156	0.4	-1.204e+00	9	246	0.6
TORSIOND	1024	-1.202e+00	3	49	0.0	-1.202e+00	5	50	0.0
TORSIOND	5476	-1.204e+00	4	123	0.2	-1.204e+00	6	167	0.2
TORSIOND	10000	-1.204e+00	4	142	0.3	-1.204e+00	9	234	0.5
TORSIOND	14884	-1.204e+00	7	339	1.3	-1.204e+00	10	312	0.9
TORSIONE	1024	-2.846e+00	3	15	0.0	-2.846e+00	3	21	0.0
TORSIONE	5476	-2.850e+00	3	35	0.0	-2.850e+00	4	53	0.0
TORSIONE	10000	-2.851e+00	3	56	0.1	-2.851e+00	4	79	0.1
TORSIONE	14884	-2.851e+00	3	68	0.2	-2.851e+00	5	92	0.2
TORSIONF	1024	-2.846e+00	4	23	0.0	-2.846e+00	4	37	0.0
TORSIONF	5476	-2.850e+00	3	42	0.0	-2.850e+00	5	93	0.1
TORSIONF	10000	-2.851e+00	3	66	0.1	-2.851e+00	4	131	0.3
TORSIONF	14884	-2.851e+00	3	106	0.3	-2.851e+00	5	211	0.6

Table B.2. Comparison between `ASA-BCP` and `NMBC` on the 43 problems considered in the performance profiles.

Problem	n	**ASA-BCP**				**NMBC**			
		obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)
BIGGSB1	10000	1.595e-02	199	2442	1.3	1.500e-02	6	6598	2.9
BIGGSB1	20000	1.595e-02	199	2442	2.5	1.500e-02	7	19025	16.6
BQPGAUSS	2003	-3.626e-01	9	9040	1.8	-3.626e-01	60	7674	1.6
CHARDIS0	10000	1.246e-19	2	1	2.8	1.526e-16	2	2	3.3

continued on next page

Table B.2 – continued from previous page

Problem	n	**ASA-BCP**				**NMBC**			
		obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)
CHENHARK	1000	-2.000e+00	107	1953	0.1	-2.000e+00	107	56216	2.2
CHENHARK	5000	-2.000e+00	73	1556	0.4	-2.000e+00	89	129087	23.0
CHENHARK	10000	-2.000e+00	108	1935	0.9	-2.000e+00	117	278775	105.7
CHENHARK	50000	-2.000e+00	137	2643	4.5	-2.000e+00	61	207468	348.6
JNLBRNG1	10000	-1.806e-01	5	391	1.1	-1.806e-01	11	699	1.9
JNLBRNG1	15625	-1.806e-01	5	429	1.6	-1.806e-01	13	955	3.7
JNLBRNG2	10000	-4.149e+00	6	527	1.3	-4.149e+00	8	752	1.9
JNLBRNG2	15625	-4.150e+00	7	659	2.3	-4.150e+00	10	969	3.4
JNLBRNGA	10000	-2.711e-01	6	418	1.1	-2.711e-01	11	650	1.7
JNLBRNGA	15625	-2.685e-01	5	446	1.8	-2.685e-01	13	907	3.4
JNLBRNGB	5625	-6.330e+00	6	1463	1.7	-6.330e+00	4	1388	1.7
JNLBRNGB	10000	-6.301e+00	7	1962	3.7	-6.301e+00	4	2333	4.6
JNLBRNGB	15625	-6.281e+00	8	2361	7.1	-6.281e+00	7	2902	8.9
MCCORMCK	10000	-9.133e+03	12	16	0.1	-9.133e+03	38	547	1.1
MCCORMCK	50000	-4.566e+04	18	29	0.5	-4.566e+04	37	255	2.5
MINSURFO	2706	2.515e+00	5	274	0.3	2.515e+00	26	2390	2.6
MINSURFO	5931	2.485e+00	105	647	1.8	2.485e+00	27	1994	4.2
NCVXBQP3	10000	-6.506e+09	54	129	0.1	-6.506e+09	6164	15326	9.8
NCVXBQP3	100000	-6.505e+11	287	1024	6.6	-6.505e+11	679	3076	16.7
NOBNDTOR	10000	-4.438e-01	5	318	0.9	-4.438e-01	16	763	2.3
NOBNDTOR	14884	-4.405e-01	6	396	1.7	-4.405e-01	19	914	3.8
NOBNDTOR	40000	-4.347e-01	9	659	7.6	-4.347e-01	34	1981	20.9
OBSTCLAE	10000	1.886e+00	23	438	1.6	1.886e+00	5	384	0.9
OBSTCLAE	15625	1.901e+00	16	406	1.9	1.901e+00	9	541	1.8
OBSTCLAL	15625	1.901e+00	4	211	0.7	1.901e+00	8	433	1.2

continued on next page

Table B.2 – continued from previous page

Problem	n	ASA-BCP				NMBC			
		obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)
OBSTCLBL	15625	7.296e+00	4	201	0.9	7.296e+00	10	370	1.6
OBSTCLBU	15625	7.296e+00	5	231	1.1	7.296e+00	10	380	1.5
ODNAMUR	11130	9.237e+03	722	40213	24.5	9.237e+03	842	58579	41.7
POWELLBC	1000	3.103e+05	483	4169	59.3	3.103e+05	287	3328	48.7
QRTQUAD	5000	-2.649e+11	716	3078	1.1	-2.649e+11	1024	3908	1.8
SCOND1LS	1002	2.967e-04	3657	944441	57.3	8.954e-04	4850	2576566	184.4
TORSION1	10000	-4.273e-01	4	265	0.6	-4.273e-01	20	556	1.3
TORSION1	14884	-4.257e-01	5	338	1.2	-4.257e-01	27	753	2.4
TORSION2	10000	-4.273e-01	4	264	0.8	-4.273e-01	12	466	1.1
TORSION2	14884	-4.257e-01	4	260	1.1	-4.257e-01	15	616	2.0
TORSIONA	10000	-4.184e-01	5	290	0.8	-4.184e-01	21	536	1.2
TORSIONB	10000	-4.184e-01	5	261	0.8	-4.184e-01	10	467	1.2
TORSIONB	14884	-4.184e-01	5	284	1.2	-4.184e-01	13	615	2.2
TORSIOND	14884	-1.204e+00	7	339	1.3	-1.204e+00	10	312	0.9

Table B.3. Comparison between `ASA-BCP` and `NMBC`: problems solved in more than 1 second by at least one algorithm and such that both algorithms find different stationary points.

Problem	n	ASA-BCP				NMBC			
		obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)
SCOND1LS	5002	1.278e-02	1874	451965	133.3	2.032e-02	1051	995227	317.3

B.2 Comparison with ALGENCAN and LANCELOT B

In Table B.4, we report the numerical results obtained by ASA-BCP, ALGENCAN and LANCELOT B on 140 problems from the CUTEst collection.

In Table B.5, we report the 62 problems that have been considered in the performance profiles between ASA-BCP, ALGENCAN and LANCELOT B (Figure 3.2). In particular, these problems have been solved in more than 1 second by at least one algorithm and all the algorithms have found the same stationary point (with a tolerance of 10^{-3}). To be more specific, let f_{AS}, f_{AL} and f_{LB} be the objective function values of the stationary points found, respectively, by ASA-BCP, ALGENCAN and LANCELOT B when applied to a particular problem. Let $f_{\min} = \min\{f_{\mathrm{AS}}, f_{\mathrm{AL}}, f_{\mathrm{LB}}\}$. We consider that ASA-BCP, ALGENCAN and LANCELOT B find the same stationary point if

$$\frac{|f_{\mathrm{AS}} - f_{\min}|}{\max\{1, f_{\min}\}} < 10^{-3}, \quad \frac{|f_{\mathrm{AL}} - f_{\min}|}{\max\{1, f_{\min}\}} < 10^{-3} \quad \text{and} \quad \frac{|f_{\mathrm{LB}} - f_{\min}|}{\max\{1, f_{\min}\}} < 10^{-3}.$$

In Table B.6, we report the problems that have been solved in more than 1 second by at least one algorithm among ASA-BCP, ALGENCAN and LANCELOT B and such that all the algorithms have found different stationary points (with a tolerance of 10^{-3}).

All the tables include the following data: the name (Problem) and the dimension (n) of the problems considered, the objective function value of the stationary point found (obj), the number of function evaluations (f-eval), the total number of conjugate gradient iterations (cg-it), the computational time in seconds (time).

Table B.4. Comparison among ASA-BCP, ALGENCAN and LANCELOT B on 140 problems from the CUTEst collection.

Problem	n	ASA-BCP				ALGENCAN				LANCELOT B			
		obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)
BDEXP	1000	3.953e-04	11	10	0.0	3.919e-04	11	69	0.0	3.919e-04	11	19	0.0
BDEXP	5000	1.967e-03	11	10	0.0	1.964e-03	11	59	0.0	1.964e-03	11	26	0.1
BIGGSB1	1000	1.500e-02	198	2272	0.1	1.500e-02	4034	$> 10^5$	2.5	1.500e-02	502	500	0.1
BIGGSB1	5000	1.595e-02	199	2442	0.6	1.500e-02	26150	$> 10^6$	283.3	1.500e-02	2502	2500	2.1
BIGGSB1	10000	1.595e-02	199	2442	1.3	1.500e-02	57308	$> 10^7$	2274.7	1.500e-02	5002	5000	7.4
BIGGSB1	20000	1.595e-02	199	2442	2.5	1.500e-02	$> 10^5$	$> 10^7$	18287.8	1.500e-02	10002	10000	32.6
BQPGAUSS	2003	-3.626e-01	9	9040	1.8	-3.626e-01	189	20144	1.3	-3.626e-01	12	4903	1.1
CHARDIS0	2000	6.384e-22	2	1	0.1	1.618e-21	2	2	0.3	7.275e-22	2	0	7.2
CHARDIS0	4000	4.648e-20	2	1	0.5	1.668e-20	2	2	1.1	5.438e-20	2	0	76.7
CHARDIS0	10000	1.246e-19	2	1	2.8	1.793e-19	2	2	6.6	1.799e-18	2	0	2226.3
CHENHARK	1000	-2.000e+00	107	1953	0.1	-2.000e+00	64	4670	0.1	-2.000e+00	206	484	0.1
CHENHARK	5000	-2.000e+00	73	1556	0.4	-2.000e+00	36	6091	0.6	-2.000e+00	611	1641	0.9
CHENHARK	10000	-2.000e+00	108	1935	0.9	-2.000e+00	35	6092	1.3	-2.000e+00	615	2294	2.0
CHENHARK	50000	-2.000e+00	137	2643	4.5	-2.000e+00	35	10094	7.2	-2.000e+00	589	3953	7.9
CVXBQP1	1000	2.252e+04	10	9	0.0	2.252e+04	11	5	0.0	2.252e+04	2	0	0.0
CVXBQP1	10000	2.250e+06	13	11	0.0	2.250e+06	14	5	0.0	2.250e+06	2	1	0.1
CVXBQP1	100000	2.250e+08	15	12	0.2	2.250e+08	18	5	0.1	2.250e+08	2	0	16.0
EXPLIN	1200	-7.193e+07	108	279	0.0	-7.192e+07	91	292	0.0	-7.193e+07	14	285	0.0
EXPLIN2	1200	-7.200e+07	41	69	0.0	-7.200e+07	109	129	0.0	-7.200e+07	10	70	0.0
EXPQUAD	1200	-3.685e+09	146	225	0.0	-3.685e+09	185	484	0.0	-3.685e+09	100	373	0.1
JNLBRNG1	1024	-1.803e-01	4	135	0.0	-1.803e-01	28	374	0.0	-1.803e-01	7	93	0.0
JNLBRNG1	1156	-1.803e-01	3	128	0.0	-1.803e-01	35	439	0.0	-1.803e-01	8	117	0.0
JNLBRNG1	5625	-1.805e-01	4	278	0.4	-1.805e-01	92	1788	0.3	-1.805e-01	15	635	1.0
JNLBRNG1	10000	-1.806e-01	5	391	1.1	-1.806e-01	121	3239	0.9	-1.806e-01	20	1135	2.9
JNLBRNG1	15625	-1.806e-01	5	429	1.6	-1.806e-01	163	4889	2.0	-1.806e-01	25	1810	7.1
JNLBRNG2	1024	-4.125e+00	3	158	0.0	-4.125e+00	12	342	0.0	-4.125e+00	5	68	0.0
JNLBRNG2	1156	-4.128e+00	3	191	0.1	-4.128e+00	13	362	0.0	-4.128e+00	5	68	0.0
JNLBRNG2	5625	-4.147e+00	5	407	0.5	-4.147e+00	44	2155	0.3	-4.147e+00	10	319	0.5

continued on next page

Table B.4 – continued from previous page

Problem	n	**ASA-BCP**				**ALGENCAN**				**LANCELOT B**			
		obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)
JNLBRNG2	10000	-4.149e+00	6	527	1.3	-4.149e+00	59	3376	0.8	-4.149e+00	12	537	1.3
JNLBRNG2	15625	-4.150e+00	7	659	2.3	-4.150e+00	79	5154	1.9	-4.150e+00	15	912	3.4
JNLBRNGA	1024	-2.954e-01	3	99	0.0	-2.954e-01	28	253	0.0	-2.954e-01	8	81	0.0
JNLBRNGA	1156	-2.935e-01	3	120	0.0	-2.935e-01	29	253	0.0	-2.935e-01	8	85	0.0
JNLBRNGA	5625	-2.753e-01	5	340	0.5	-2.753e-01	75	734	0.2	-2.753e-01	15	441	0.7
JNLBRNGA	10000	-2.711e-01	6	418	1.1	-2.711e-01	106	1199	0.5	-2.711e-01	19	824	2.2
JNLBRNGA	15625	-2.685e-01	5	446	1.8	-2.685e-01	129	1508	0.8	-2.685e-01	22	1327	5.2
JNLBRNGB	1024	-6.440e+00	4	596	0.1	-6.440e+00	11	841	0.0	-6.440e+00	8	48	0.0
JNLBRNGB	1156	-6.429e+00	4	593	0.1	-6.429e+00	17	1137	0.0	-6.429e+00	8	52	0.0
JNLBRNGB	5625	-6.330e+00	6	1463	1.7	-6.330e+00	26	2744	0.4	-6.330e+00	10	106	0.2
JNLBRNGB	10000	-6.301e+00	7	1962	3.7	-6.301e+00	51	3418	0.9	-6.301e+00	10	156	0.5
JNLBRNGB	15625	-6.281e+00	8	2361	7.1	-6.281e+00	56	5074	1.9	-6.281e+00	11	329	1.3
LINVERSE	1999	6.810e+02	21	241	0.1	6.820e+02	25	71	0.0	6.810e+02	23	1645	0.5
MCCORMCK	1000	-9.137e+02	7	20	0.0	-9.137e+02	11	27	0.0	-9.137e+02	8	4	0.0
MCCORMCK	5000	-4.567e+03	13	22	0.0	-4.567e+03	10	29	0.0	-4.567e+03	7	5	0.0
MCCORMCK	10000	-9.133e+03	12	16	0.1	-9.133e+03	10	29	0.1	-9.133e+03	7	4	0.1
MCCORMCK	50000	-4.566e+04	18	29	0.5	-4.566e+04	10	29	0.3	-4.566e+04	8	6	0.3
MINSURFO	2706	2.515e+00	5	274	0.3	2.515e+00	8	525	0.1	2.515e+00	7	53	0.1
MINSURFO	5931	2.485e+00	105	647	1.8	2.485e+00	10	948	0.2	2.485e+00	9	119	0.3
NCVXBQP1	1000	-1.987e+08	10	9	0.0	-1.987e+08	4	1	0.0	-1.987e+08	2	0	0.0
NCVXBQP1	10000	-1.986e+10	11	10	0.0	-1.986e+10	8	1	0.0	-1.986e+10	2	0	0.1
NCVXBQP1	100000	-1.985e+12	15	11	0.2	-1.985e+12	11	1	0.1	-1.985e+12	2	6	16.1
NCVXBQP2	1000	-1.334e+08	15	24	0.0	-1.334e+08	26	24	0.0	-1.334e+08	3	39	0.0
NCVXBQP2	10000	-1.334e+10	31	63	0.1	-1.334e+10	84	102	0.1	-1.334e+10	4	407	0.2
NCVXBQP2	100000	-1.334e+12	41	100	0.7	-1.334e+12	87	142	1.2	-1.334e+12	4	4013	18.5
NCVXBQP3	1000	-6.530e+07	20	34	0.0	-6.577e+07	13	18	0.0	-6.579e+07	4	34	0.0
NCVXBQP3	10000	-6.506e+09	54	129	0.1	-6.557e+09	29	50	0.0	-6.558e+09	6	360	0.2
NCVXBQP3	100000	-6.505e+11	287	1024	6.6	-6.556e+11	69	185	1.1	-6.557e+11	6	3441	20.3
NOBNDTOR	5476	-4.499e-01	4	235	0.4	-4.499e-01	126	1725	0.4	-4.499e-01	24	363	0.7

continued on next page

Table B.4 – continued from previous page

Problem	n	**ASA-BCP**				**ALGENCAN**				**LANCELOT B**			
		obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)
NOBNDTOR	10000	-4.438e-01	5	318	0.9	-4.438e-01	173	2630	0.9	-4.438e-01	31	596	2.0
NOBNDTOR	14884	-4.405e-01	6	396	1.7	-4.405e-01	224	3763	1.9	-4.405e-01	37	790	4.0
NOBNDTOR	40000	-4.347e-01	9	659	7.6	-4.347e-01	408	8922	11.7	-4.347e-01	61	1791	23.9
NONSCOMP	1000	8.661e-16	7	37	0.0	2.981e-15	9	62	0.0	2.986e-15	9	8	0.0
NONSCOMP	5000	5.054e-13	7	34	0.0	1.529e-14	9	59	0.0	1.523e-14	9	9	0.0
NONSCOMP	10000	1.680e-12	7	37	0.0	3.062e-14	9	56	0.0	3.055e-14	9	9	0.0
OBSTCLAE	5625	1.863e+00	9	221	0.4	1.863e+00	50	788	0.2	1.863e+00	6	2488	4.7
OBSTCLAE	10000	1.886e+00	23	438	1.6	1.886e+00	85	945	0.6	1.886e+00	7	4646	15.1
OBSTCLAE	15625	1.901e+00	16	406	1.9	1.901e+00	105	2244	1.0	1.901e+00	5	7409	37.5
OBSTCLAL	5625	1.863e+00	4	124	0.2	1.863e+00	63	789	0.2	1.863e+00	17	196	0.3
OBSTCLAL	10000	1.886e+00	4	159	0.3	1.886e+00	83	1224	0.4	1.886e+00	20	336	0.7
OBSTCLAL	15625	1.901e+00	4	211	0.7	1.901e+00	112	1862	0.9	1.901e+00	25	480	1.6
OBSTCLBL	5625	7.231e+00	4	116	0.2	7.231e+00	51	323	0.2	7.231e+00	13	953	1.5
OBSTCLBL	10000	7.272e+00	4	155	0.5	7.272e+00	94	435	0.5	7.272e+00	16	1797	4.9
OBSTCLBL	15625	7.296e+00	4	201	0.9	7.296e+00	110	516	1.1	7.296e+00	19	2761	11.8
OBSTCLBM	5625	7.231e+00	3	88	0.1	7.231e+00	20	181	0.1	7.231e+00	6	485	0.9
OBSTCLBM	10000	7.272e+00	4	119	0.4	7.272e+00	32	285	0.2	7.272e+00	6	794	2.6
OBSTCLBM	15625	7.296e+00	4	164	0.8	7.296e+00	34	310	0.3	7.296e+00	6	1377	7.1
OBSTCLBU	5625	7.231e+00	5	211	0.4	7.231e+00	43	371	0.1	7.231e+00	14	230	0.4
OBSTCLBU	10000	7.272e+00	4	163	0.5	7.272e+00	60	478	0.3	7.272e+00	17	425	1.2
OBSTCLBU	15625	7.296e+00	5	231	1.1	7.296e+00	81	687	0.7	7.296e+00	20	787	3.3
ODNAMUR	11130	9.237e+03	722	40213	24.5	9.237e+03	368	$> 10^5$	3037.8	9.237e+03	12	27886	23.6
PENTDI	1000	-7.500e-01	3	11	0.0	-7.500e-01	2	0	0.0	-7.500e-01	2	0	0.0
PENTDI	5000	-7.500e-01	3	11	0.0	-7.500e-01	2	0	0.0	-7.500e-01	2	0	0.0
PENTDI	10000	-7.500e-01	3	11	0.0	-7.500e-01	2	0	0.0	-7.500e-01	2	0	0.0
PENTDI	50000	-7.500e-01	3	11	0.1	-7.500e-01	2	0	0.0	-7.500e-01	2	0	0.0
POWELLBC	1000	3.103e+05	483	4169	59.3	3.103e+05	361	2971	17.7	3.198e+05	635	3296	79.7
QRTQUAD	1200	-3.685e+09	118	638	0.1	-3.685e+09	206	851	0.0	-3.685e+09	230	435	0.3
QRTQUAD	5000	-2.649e+11	716	3078	1.1	-2.649e+11	531	2822	0.5	-2.649e+11	3568	11356	34.0

continued on next page

Table B.4 – continued from previous page

Problem	n	**ASA-BCP**				**ALGENCAN**				**LANCELOT B**			
		obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)
QUDLIN	1200	-7.200e+07	4	1	0.0	-7.200e+07	2	0	0.0	-7.200e+07	2	0	0.0
QUDLIN	5000	-1.250e+09	7	2	0.0	-1.250e+09	2	0	0.0	-1.250e+09	2	0	0.0
QUDLIN	10000	-5.000e+09	8	2	0.0	-5.000e+09	2	0	0.0	-5.000e+09	2	0	0.1
SCOND1LS	1002	2.967e-04	3657	$> 10^5$	57.3	8.938e-06	1045	$> 10^5$	12.3	1.158e-10	645	618	0.3
SCOND1LS	5002	1.278e-02	1874	$> 10^5$	133.3	2.391e-04	1060	$> 10^6$	318.7	3.418e-05	740	716	1.8
SINEALI	1000	-9.990e+04	16	44	0.0	-9.990e+04	31	64	0.0	-9.990e+04	14	10	0.0
TORSION1	1024	-4.450e-01	3	77	0.0	-4.450e-01	33	172	0.0	-4.450e-01	11	73	0.0
TORSION1	5476	-4.303e-01	4	213	0.3	-4.303e-01	116	1027	0.3	-4.303e-01	24	321	0.5
TORSION1	10000	-4.273e-01	4	265	0.6	-4.273e-01	173	1791	0.7	-4.273e-01	32	549	1.5
TORSION1	14884	-4.257e-01	5	338	1.2	-4.257e-01	217	2498	1.3	-4.257e-01	38	793	3.1
TORSION2	1024	-4.450e-01	3	67	0.0	-4.450e-01	30	129	0.0	-4.450e-01	7	301	0.1
TORSION2	5476	-4.303e-01	4	151	0.3	-4.303e-01	87	562	0.2	-4.303e-01	10	1640	3.1
TORSION2	10000	-4.273e-01	4	264	0.8	-4.273e-01	117	817	0.4	-4.273e-01	10	3138	11.0
TORSION2	14884	-4.257e-01	4	260	1.1	-4.257e-01	143	983	0.7	-4.257e-01	10	4326	22.6
TORSION3	1024	-1.232e+00	3	41	0.0	-1.232e+00	12	45	0.0	-1.232e+00	6	23	0.0
TORSION3	5476	-1.217e+00	3	84	0.1	-1.217e+00	43	292	0.1	-1.217e+00	12	98	0.1
TORSION3	10000	-1.214e+00	4	143	0.3	-1.214e+00	66	525	0.2	-1.214e+00	16	171	0.4
TORSION3	14884	-1.212e+00	4	156	0.4	-1.212e+00	92	807	0.5	-1.212e+00	20	241	0.7
TORSION4	1024	-1.232e+00	3	37	0.0	-1.232e+00	18	71	0.0	-1.232e+00	7	256	0.1
TORSION4	5476	-1.217e+00	3	102	0.1	-1.217e+00	50	201	0.1	-1.217e+00	10	1387	2.3
TORSION4	10000	-1.214e+00	4	131	0.3	-1.214e+00	72	319	0.2	-1.214e+00	12	4572	14.0
TORSION4	14884	-1.212e+00	5	221	0.8	-1.212e+00	102	464	0.4	-1.212e+00	16	5646	27.5
TORSION5	1024	-2.876e+00	3	15	0.0	-2.876e+00	6	21	0.0	-2.876e+00	4	7	0.0
TORSION5	5476	-2.863e+00	3	39	0.0	-2.863e+00	18	103	0.0	-2.863e+00	7	33	0.0
TORSION5	10000	-2.860e+00	3	62	0.1	-2.860e+00	27	171	0.1	-2.860e+00	9	57	0.1
TORSION5	14884	-2.859e+00	3	68	0.2	-2.859e+00	33	215	0.2	-2.859e+00	10	72	0.2
TORSION6	1024	-2.876e+00	3	20	0.0	-2.876e+00	11	24	0.0	-2.876e+00	6	171	0.0
TORSION6	5476	-2.863e+00	3	47	0.0	-2.863e+00	26	76	0.1	-2.863e+00	8	2002	2.2
TORSION6	10000	-2.860e+00	3	65	0.1	-2.860e+00	37	110	0.1	-2.860e+00	9	4400	8.5

continued on next page

Table B.4 – continued from previous page

Problem	n	ASA-BCP				ALGENCAN				LANCELOT B			
		obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)
TORSION6	14884	-2.859e+00	4	116	0.3	-2.859e+00	43	111	0.2	-2.859e+00	11	4933	16.8
TORSIONA	1024	-4.174e-01	3	78	0.0	-4.174e-01	33	172	0.0	-4.174e-01	11	77	0.0
TORSIONA	5476	-4.183e-01	4	222	0.3	-4.183e-01	116	1034	0.3	-4.183e-01	24	320	0.6
TORSIONA	10000	-4.184e-01	5	290	0.8	-4.184e-01	173	1802	0.7	-4.184e-01	32	558	1.6
TORSIONA	5476	-4.183e-01	4	222	0.3	-4.183e-01	116	1034	0.3	-4.183e-01	24	320	0.5
TORSIONB	1024	-4.174e-01	3	78	0.0	-4.174e-01	25	126	0.0	-4.174e-01	8	149	0.1
TORSIONB	5476	-4.183e-01	4	164	0.3	-4.183e-01	73	488	0.1	-4.183e-01	8	1677	3.3
TORSIONB	10000	-4.184e-01	5	261	0.8	-4.184e-01	101	714	0.4	-4.184e-01	10	2593	9.5
TORSIONB	14884	-4.184e-01	5	284	1.2	-4.184e-01	127	909	0.7	-4.184e-01	10	4447	24.0
TORSIONC	1024	-1.202e+00	3	42	0.0	-1.202e+00	12	44	0.0	-1.202e+00	6	23	0.0
TORSIONC	5476	-1.204e+00	3	84	0.1	-1.204e+00	43	291	0.1	-1.204e+00	12	98	0.1
TORSIONC	10000	-1.204e+00	4	139	0.3	-1.204e+00	66	526	0.3	-1.204e+00	16	171	0.4
TORSIONC	14884	-1.204e+00	4	156	0.4	-1.204e+00	92	806	0.5	-1.204e+00	20	241	0.7
TORSIOND	1024	-1.202e+00	3	49	0.0	-1.202e+00	19	72	0.0	-1.202e+00	8	266	0.1
TORSIOND	5476	-1.204e+00	4	123	0.2	-1.204e+00	51	203	0.1	-1.204e+00	11	1399	1.9
TORSIOND	10000	-1.204e+00	4	142	0.3	-1.204e+00	71	322	0.3	-1.204e+00	16	2136	7.0
TORSIOND	14884	-1.204e+00	7	339	1.3	-1.204e+00	103	468	0.5	-1.204e+00	10	9134	41.2
TORSIONE	1024	-2.846e+00	3	15	0.0	-2.846e+00	6	20	0.0	-2.846e+00	4	7	0.0
TORSIONE	5476	-2.850e+00	3	35	0.0	-2.850e+00	18	102	0.1	-2.850e+00	7	33	0.0
TORSIONE	10000	-2.851e+00	3	56	0.1	-2.851e+00	27	170	0.1	-2.851e+00	9	57	0.1
TORSIONE	14884	-2.851e+00	3	68	0.2	-2.851e+00	33	215	0.2	-2.851e+00	10	72	0.2
TORSIONF	1024	-2.846e+00	4	23	0.0	-2.846e+00	11	41	0.0	-2.846e+00	6	190	0.0
TORSIONF	5476	-2.850e+00	3	42	0.0	-2.850e+00	26	124	0.1	-2.850e+00	8	1960	2.2
TORSIONF	10000	-2.851e+00	3	66	0.1	-2.851e+00	33	184	0.2	-2.851e+00	9	4416	8.9
TORSIONF	14884	-2.851e+00	3	106	0.3	-2.851e+00	42	206	0.2	-2.851e+00	11	5112	18.3

Table B.5. Comparison between ASA-BCP, ALGENCANand LANCELOT B on the 62 problems considered in the performance profiles.

Problem	n	**ASA-BCP**				**ALGENCAN**				**LANCELOT B**			
		obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)
BIGGSB1	1000	1.500e-02	198	2272	0.1	1.500e-02	4034	$> 10^5$	2.5	1.500e-02	502	500	0.1
BIGGSB1	5000	1.595e-02	199	2442	0.6	1.500e-02	26150	$> 10^6$	283.3	1.500e-02	2502	2500	2.1
BIGGSB1	10000	1.595e-02	199	2442	1.3	1.500e-02	57308	$> 10^7$	2274.7	1.500e-02	5002	5000	7.4
BIGGSB1	20000	1.595e-02	199	2442	2.5	1.500e-02	$> 10^5$	$> 10^7$	18287.8	1.500e-02	10002	10000	32.6
BQPGAUSS	2003	-3.626e-01	9	9040	1.8	-3.626e-01	189	20144	1.3	-3.626e-01	12	4903	1.1
CHARDIS0	2000	6.384e-22	2	1	0.1	1.618e-21	2	2	0.3	7.275e-22	2	0	7.2
CHARDIS0	4000	4.648e-20	2	1	0.5	1.668e-20	2	2	1.1	5.438e-20	2	0	76.7
CHARDIS0	10000	1.246e-19	2	1	2.8	1.793e-19	2	2	6.6	1.799e-18	2	0	2226.3
CHENHARK	10000	-2.000e+00	108	1935	0.9	-2.000e+00	35	6092	1.3	-2.000e+00	615	2294	2.0
CHENHARK	50000	-2.000e+00	137	2643	4.5	-2.000e+00	35	10094	7.2	-2.000e+00	589	3953	7.9
CVXBQP1	100000	2.250e+08	15	12	0.2	2.250e+08	18	5	0.1	2.250e+08	2	0	16.0
JNLBRNG1	5625	-1.805e-01	4	278	0.4	-1.805e-01	92	1788	0.3	-1.805e-01	15	635	1.0
JNLBRNG1	10000	-1.806e-01	5	391	1.1	-1.806e-01	121	3239	0.9	-1.806e-01	20	1135	2.9
JNLBRNG1	15625	-1.806e-01	5	429	1.6	-1.806e-01	163	4889	2.0	-1.806e-01	25	1810	7.1
JNLBRNG2	10000	-4.149e+00	6	527	1.3	-4.149e+00	59	3376	0.8	-4.149e+00	12	537	1.3
JNLBRNG2	15625	-4.150e+00	7	659	2.3	-4.150e+00	79	5154	1.9	-4.150e+00	15	912	3.4
JNLBRNGA	10000	-2.711e-01	6	418	1.1	-2.711e-01	106	1199	0.5	-2.711e-01	19	824	2.2
JNLBRNGA	15625	-2.685e-01	5	446	1.8	-2.685e-01	129	1508	0.8	-2.685e-01	22	1327	5.2
JNLBRNGB	5625	-6.330e+00	6	1463	1.7	-6.330e+00	26	2744	0.4	-6.330e+00	10	106	0.2
JNLBRNGB	10000	-6.301e+00	7	1962	3.7	-6.301e+00	51	3418	0.9	-6.301e+00	10	156	0.5
JNLBRNGB	15625	-6.281e+00	8	2361	7.1	-6.281e+00	56	5074	1.9	-6.281e+00	11	329	1.3
MINSURFO	5931	2.485e+00	105	647	1.8	2.485e+00	10	948	0.2	2.485e+00	9	119	0.3
NCVXBQP1	100000	-1.985e+12	15	11	0.2	-1.985e+12	11	1	0.1	-1.985e+12	2	6	16.1
NCVXBQP2	100000	-1.334e+12	41	100	0.7	-1.334e+12	87	142	1.2	-1.334e+12	4	4013	18.5
NOBNDTOR	10000	-4.438e-01	5	318	0.9	-4.438e-01	173	2630	0.9	-4.438e-01	31	596	2.0
NOBNDTOR	14884	-4.405e-01	6	396	1.7	-4.405e-01	224	3763	1.9	-4.405e-01	37	790	4.0
NOBNDTOR	40000	-4.347e-01	9	659	7.6	-4.347e-01	408	8922	11.7	-4.347e-01	61	1791	23.9
OBSTCLAE	5625	1.863e+00	9	221	0.4	1.863e+00	50	788	0.2	1.863e+00	6	2488	4.7

continued on next page

Table B.5 – continued from previous page

Problem	n	**ASA-BCP**				**ALGENCAN**				**LANCELOT B**			
		obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)
OBSTCLAE	10000	1.886e+00	23	438	1.6	1.886e+00	85	945	0.6	1.886e+00	7	4646	15.1
OBSTCLAE	15625	1.901e+00	16	406	1.9	1.901e+00	105	2244	1.0	1.901e+00	5	7409	37.5
OBSTCLAL	15625	1.901e+00	4	211	0.7	1.901e+00	112	1862	0.9	1.901e+00	25	480	1.6
OBSTCLBL	5625	7.231e+00	4	116	0.2	7.231e+00	51	323	0.2	7.231e+00	13	953	1.5
OBSTCLBL	10000	7.272e+00	4	155	0.5	7.272e+00	94	435	0.5	7.272e+00	16	1797	4.9
OBSTCLBL	15625	7.296e+00	4	201	0.9	7.296e+00	110	516	1.1	7.296e+00	19	2761	11.8
OBSTCLBM	10000	7.272e+00	4	119	0.4	7.272e+00	32	285	0.2	7.272e+00	6	794	2.6
OBSTCLBM	15625	7.296e+00	4	164	0.8	7.296e+00	34	310	0.3	7.296e+00	6	1377	7.1
OBSTCLBU	10000	7.272e+00	4	163	0.5	7.272e+00	60	478	0.3	7.272e+00	17	425	1.2
OBSTCLBU	15625	7.296e+00	5	231	1.1	7.296e+00	81	687	0.7	7.296e+00	20	787	3.3
ODNAMUR	11130	9.237e+03	722	40213	24.5	9.237e+03	368	$> 10^5$	3037.8	9.237e+03	12	27886	23.6
QRTQUAD	5000	-2.649e+11	716	3078	1.1	-2.649e+11	531	2822	0.5	-2.649e+11	3568	11356	34.0
SCOND1LS	1002	2.967e-04	3657	$> 10^5$	57.3	8.938e-06	1045	$> 10^5$	12.3	1.158e-10	645	618	0.3
TORSION1	10000	-4.273e-01	4	265	0.6	-4.273e-01	173	1791	0.7	-4.273e-01	32	549	1.5
TORSION1	14884	-4.257e-01	5	338	1.2	-4.257e-01	217	2498	1.3	-4.257e-01	38	793	3.1
TORSION2	5476	-4.303e-01	4	151	0.3	-4.303e-01	87	562	0.2	-4.303e-01	10	1640	3.1
TORSION2	10000	-4.273e-01	4	264	0.8	-4.273e-01	117	817	0.4	-4.273e-01	10	3138	11.0
TORSION2	14884	-4.257e-01	4	260	1.1	-4.257e-01	143	983	0.7	-4.257e-01	10	4326	22.6
TORSION4	5476	-1.217e+00	3	102	0.1	-1.217e+00	50	201	0.1	-1.217e+00	10	1387	2.3
TORSION4	10000	-1.214e+00	4	131	0.3	-1.214e+00	72	319	0.2	-1.214e+00	12	4572	14.0
TORSION4	14884	-1.212e+00	5	221	0.8	-1.212e+00	102	464	0.4	-1.212e+00	16	5646	27.5
TORSION6	5476	-2.863e+00	3	47	0.0	-2.863e+00	26	76	0.1	-2.863e+00	8	2002	2.2
TORSION6	10000	-2.860e+00	3	65	0.1	-2.860e+00	37	110	0.1	-2.860e+00	9	4400	8.5
TORSION6	14884	-2.859e+00	4	116	0.3	-2.859e+00	43	111	0.2	-2.859e+00	11	4933	16.8
TORSIONA	10000	-4.184e-01	5	290	0.8	-4.184e-01	173	1802	0.7	-4.184e-01	32	558	1.6
TORSIONB	5476	-4.183e-01	4	164	0.3	-4.183e-01	73	488	0.1	-4.183e-01	8	1677	3.3
TORSIONB	10000	-4.184e-01	5	261	0.8	-4.184e-01	101	714	0.4	-4.184e-01	10	2593	9.5
TORSIONB	14884	-4.184e-01	5	284	1.2	-4.184e-01	127	909	0.7	-4.184e-01	10	4447	24.0
TORSIOND	5476	-1.204e+00	4	123	0.2	-1.204e+00	51	203	0.1	-1.204e+00	11	1399	1.9

continued on next page

Table B.5 – continued from previous page

Problem	n	ASA-BCP				ALGENCAN				LANCELOT B			
		obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)
TORSIOND	10000	-1.204e+00	4	142	0.3	-1.204e+00	71	322	0.3	-1.204e+00	16	2136	7.0
TORSIOND	14884	-1.204e+00	7	339	1.3	-1.204e+00	103	468	0.5	-1.204e+00	10	9134	41.2
TORSIONF	5476	-2.850e+00	3	42	0.0	-2.850e+00	26	124	0.1	-2.850e+00	8	1960	2.2
TORSIONF	10000	-2.851e+00	3	66	0.1	-2.851e+00	33	184	0.2	-2.851e+00	9	4416	8.9
TORSIONF	14884	-2.851e+00	3	106	0.3	-2.851e+00	42	206	0.2	-2.851e+00	11	5112	18.3

Table B.6. Comparison between `ASA-BCP`, `ALGENCAN`and `LANCELOT B`: problems solved in more than 1 second by at least one algorithm and such that all the algorithms find different stationary points.

Problem	n	ASA-BCP				ALGENCAN				LANCELOT B			
		obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)	obj	f-eval	cg-it	time (s)
NCVXBQP3	100000	-6.505e+11	287	1024	6.6	-6.556e+11	69	185	1.1	-6.557e+11	6	3441	20.3
POWELLBC	1000	3.103e+05	483	4169	59.3	3.103e+05	361	2971	17.7	3.198e+05	635	3296	79.7
SCOND1LS	5002	1.278e-02	1874	$> 10^5$	133.3	2.391e-04	1060	$> 10^6$	318.7	3.418e-05	740	716	1.8

Bibliography

[1] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt. Second-order negative-curvature methods for box-constrained and general constrained optimization. *Computational Optimization and Applications*, 45(2):209–236, 2010.

[2] J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988.

[3] D. P. Bertsekas. Projected Newton methods for optimization problems with simple constraints. *SIAM Journal on control and Optimization*, 20(2):221–246, 1982.

[4] D. P. Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 1999.

[5] E. G. Birgin and J. M. Gentil. Evaluating bound-constrained minimization software. *Computational Optimization and Applications*, 53(2):347–373, 2012.

[6] E. G. Birgin and J. M. Martínez. Large-scale active-set box-constrained optimization method with spectral projected gradients. *Computational Optimization and Applications*, 23(1):101–125, 2002.

[7] P. Bradley and O. Mangasarian. Feature Selection via Concave Minimization and Support Vector Machines. In *ICML*, volume 98, pages 82–90, 1998.

[8] C. Buchheim, M. De Santis, S. Lucidi, F. Rinaldi, and L. Trieu. A Feasible Active Set Method with Reoptimization for Convex Quadratic Mixed-Integer Programming. *SIAM Journal on Optimization*, 26(3):1695–1714, 2016.

[9] W. Cheng and D. Li. An active set modified Polak–Ribiere–Polyak method for large-scale nonlinear bound constrained optimization. *Journal of Optimization Theory and Applications*, 155(3):1084–1094, 2012.

[10] E. Chi and K. Lange. Splitting methods for convex clustering. *Journal of Computational and Graphical Statistics*, 2014.

[11] A. R. Conn, N. I. Gould, and P. L. Toint. Global convergence of a class of trust region algorithms for optimization with simple bounds. *SIAM journal on numerical analysis*, 25(2):433–460, 1988.

[12] A. Cristofari. Data Filtering for Cluster Analysis by ℓ_0-Norm Regularization. *arXiv preprint arXiv:1607.08756*, 2016.

[13] A. Cristofari, M. De Santis, S. Lucidi, and F. Rinaldi. A Two-Stage Active-Set Algorithm for Bound-Constrained Optimization. *Journal of Optimization Theory and Applications*, pages 1–33, 2016.

[14] A. Cristofari, M. De Santis, S. Lucidi, and F. Rinaldi. New Active-Set Frank-Wolfe Variants for Minimization over the Simplex and the L1-Ball. Technical report, Department of Computer, Control and Management Engineering, Sapienza University of Rome, 2016.

[15] E. De Klerk. The complexity of optimizing over a simplex, hypercube or sphere: a short survey. *Central European Journal of Operations Research*, 16(2):111–125, 2008.

[16] M. De Santis, G. Di Pillo, and S. Lucidi. An active set feasible method for large-scale minimization problems with bound constraints. *Computational Optimization and Applications*, 53(2):395–423, 2012.

[17] M. De Santis, S. Lucidi, and F. Rinaldi. A Fast Active Set Block Coordinate Descent Algorithm for ℓ_1-regularized least squares. *SIAM Journal on Optimization*, 26(1):781–809, 2016.

[18] R. S. Dembo and T. Steihaug. Truncated-Newton algorithms for large-scale unconstrained optimization. *Mathematical Programming*, 26(2):190–212, 1983.

[19] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[20] J. Dennis, M. Heinkenschloss, and L. N. Vicente. Trust-region interior-point SQP algorithms for a class of nonlinear programming problems. *SIAM Journal on Control and Optimization*, 36(5):1750–1794, 1998.

[21] I. Dhillon, Y. Guan, and B. Kulis. Kernel k-means, Spectral Clustering and Normalized Cuts. In *Proceedings of the 10th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 551–556, 2004.

[22] G. Di Pillo and L. Grippo. A class of continuously differentiable exact penalty function algorithms for nonlinear programming problems. In *System Modelling and Optimization*, pages 246–256. Springer, 1984.

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

[24] F. Facchinei and S. Lucidi. Quadratically and superlinearly convergent algorithms for the solution of inequality constrained minimization problems. *Journal of Optimization Theory and Applications*, 85(2):265–289, 1995.

[25] F. Facchinei, J. Júdice, and J. Soares. An active set Newton algorithm for large-scale nonlinear programs with box constraints. *SIAM Journal on Optimization*, 8(1):158–186, 1998.

[26] F. Facchinei, S. Lucidi, and L. Palagi. A truncated Newton algorithm for large scale box constrained optimization. *SIAM Journal on Optimization*, 12(4): 1100–1125, 2002.

[27] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456): 1348–1360, 2001.

[28] G. Fasano and S. Lucidi. A nonmonotone truncated Newton-Krylov method exploiting negative curvature directions, for large scale unconstrained optimization. *Optimization Letters*, 3(4):521–535, 2009.

[29] K. Fountoulakis and J. Gondzio. A second-order method for strongly convex l1-regularization problems. *Mathematical Programming A*, 156:189–219, 2014.

[30] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

[31] G. Gan, C. Ma, and J. Wu. *Data clustering: theory, algorithms, and applications*, volume 20. Siam, 2007.

[32] M. R. Garey and S. Johnson, David. Computers and intractability: a guide to the theory of NP-completeness. *WH Free. Co., San Fr*, 1979.

[33] M. Girolami. Mercer kernel-based clustering in feature space. *Neural Networks, IEEE Transactions on Neural Networks*, 13(3):780–784, 2002.

[34] N. I. Gould, S. Lucidi, M. Roma, and P. L. Toint. Exploiting negative curvature directions in linesearch methods for unconstrained optimization. *Optimization Methods and Software*, 14(1-2):75–98, 2000.

[35] N. I. Gould, D. Orban, and P. L. Toint. GALAHAD, a library of thread-safe Fortran 90 packages for large-scale nonlinear optimization. *ACM Transactions on Mathematical Software (TOMS)*, 29(4):353–372, 2003.

[36] N. I. Gould, D. Orban, and P. L. Toint. CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization. *Computational Optimization and Applications*, 60(3):545–557, 2015.

[37] L. Grippo and S. Lucidi. A differentiable exact penalty function for bound constrained quadratic programming problems. *Optimization*, 22(4):557–578, 1991.

[38] L. Grippo and M. Sciandrone. *Metodi di ottimizzazione non vincolata*. Springer Science & Business Media, 2011.

[39] L. Grippo, F. Lampariello, and S. Lucidi. A truncated Newton method with nonmonotone line search for unconstrained optimization. *Journal of Optimization Theory and Applications*, 60(3):401–419, 1989.

[40] L. Grippo, F. Lampariello, and S. Lucidi. A class of nonmonotone stabilization methods in unconstrained optimization. *Numerische Mathematik*, 59(1):779–805, 1991.

[41] J. Guélat and P. Marcotte. Some comments on Wolfe's 'away step'. *Mathematical Programming*, 35(1):110–119, 1986.

[42] W. W. Hager and H. Zhang. A survey of nonlinear conjugate gradient methods. *Pacific journal of Optimization*, 2(1):35–58, 2006.

[43] M. Heinkenschloss, M. Ulbrich, and S. Ulbrich. Superlinear and quadratic convergence of affine-scaling interior-point Newton methods for problems with simple bounds without strict complementarity assumption. *Mathematical Programming*, 86(3):615–635, 1999.

[44] M. R. Hestenes and E. Stiefel. *Methods of conjugate gradients for solving linear systems*, volume 49. NBS, 1952.

[45] T. Hocking, A. Joulin, F. Bach, and J.-P. Vert. Clusterpath an algorithm for clustering using convex fusion penalties. In *28th international conference on machine learning*, 2011.

[46] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2 (1):193–218, 1985.

[47] C. Kanzow and A. Klug. On affine-scaling interior-point Newton methods for nonlinear minimization with bound constraints. *Computational Optimization and Applications*, 35(2):177–197, 2006.

[48] S. Lacoste-Julien and M. Jaggi. On the global linear convergence of Frank-Wolfe optimization variants. In *NIPS 2015 - Advances in Neural Information Processing Systems*, 2015.

[49] M. Lichman. UCI Machine Learning Repository, 2013. URL `http://archive.ics.uci.edu/ml`.

[50] C.-J. Lin. On the convergence of the decomposition method for support vector machines. *IEEE Transactions on Neural Networks*, 12(6):1288–1298, 2001.

[51] C.-J. Lin and J. J. Moré. Newton's method for large bound-constrained optimization problems. *SIAM Journal on Optimization*, 9(4):1100–1127, 1999.

[52] F. Lindsten, H. Ohlsson, and L. Ljung. Clustering using sum-of-norms regularization: With application to particle filter output computation. In *Statistical Signal Processing Workshop (SSP), 2011 IEEE*, pages 201–204, 2011.

[53] S. Lucidi, F. Rochetich, and M. Roma. Curvilinear stabilization techniques for truncated Newton methods in large scale unconstrained optimization. *SIAM Journal on Optimization*, 8(4):916–939, 1998.

[54] O. Mangasarian. Machine learning via polyhedral concave minimization. In *Applied Mathematics and Parallel Computing*, pages 175–188. 1996.

[55] O. Mangasarian, R. Setiono, and W. Wolberg. Pattern recognition via linear programming: Theory and application to medical diagnosis. *Large-scale numerical optimization*, pages 22–31, 1990.

[56] Y. Marchetti and Q. Zhou. Solution path clustering with adaptive concave penalty. *Electronic Journal of Statistics*, 8(1):1569–1603, 2014.

[57] G. McLachlan and D. Peel. *Finite mixture models*. John Wiley & Sons, 2004.

[58] B. Mitchell, V. F. Dem'yanov, and V. Malozemov. Finding the point of a polyhedron closest to the origin. *SIAM Journal on Control*, 12(1):19–26, 1974.

[59] S. G. Nash. A survey of truncated-Newton methods. *Journal of Computational and Applied Mathematics*, 124(1):45–59, 2000.

[60] S. G. Nash and A. Sofer. Assessing a search direction within a truncated-Newton method. *Operations Research Letters*, 9(4):219–221, 1990.

[61] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[62] J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*, volume 30. Siam, 1970.

[63] W. Pan, X. Shen, and B. Liu. Cluster Analysis: Unsupervised Learning via Supervised Learning with a Non-convex Penalty. *Journal of Machine Learning Research*, 14(1):1865–1889, 2013.

[64] K. Pelckmans, J. De Brabanter, J. Suykens, and B. De Moor. Convex clustering shrinkage. In *PASCAL Workshop on Statistics and Optimization of Clustering Workshop*, 2005.

[65] J. C. Platt. Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods*, pages 185–208, 1999.

[66] E. Polak. *Optimization: algorithms and consistent approximations*, volume 124. Springer Science & Business Media, 2012.

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

[68] R. Rockafellar and R. Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

[69] B. Schölkopf and A. J. Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[70] B. Schölkopf, A. Smola, and K. Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10:1299–1319, 1998.

[71] A. Schwartz and E. Polak. Family of projected descent methods for optimization problems with simple bounds. *Journal of Optimization Theory and Applications*, 92(1):1–31, 1997.

[72] X. Shen, W. Pan, and Y. Zhu. Likelihood-based selection and sharp parameter estimation. *Journal of the American Statistical Association*, 107(497):223–232, 2012.

[73] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[74] Z. Wen, W. Yin, D. Goldfarb, and Y. Zhang. A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation. *SIAM Journal on Scientific Computing*, 32(4):1832–1857, 2010.

[75] P. Wolfe. Convergence theory in nonlinear programming. *Integer and nonlinear programming*, pages 1–36, 1970.

[76] H. Zhang and W. W. Hager. A nonmonotone line search technique and its application to unconstrained optimization. *SIAM Journal on Optimization*, 14 (4):1043–1056, 2004.