SAPIENZA
UNIVERSITÀ DI ROMA

Gradient-based methods with subspace acceleration for quadratic programming problems and applications

PhD in Automatica, Bioengineering and Operations Research

Curriculum in Operations Research – XXXI Course

Candidate

Marco Viola
ID number 1691842

Thesis Advisors

Prof. Gerardo Toraldo
Prof. Massimo Roma

Co-Advisor

Prof. Daniela di Serafino

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

October 31, 2018

Thesis defended on February 26, 2019
in front of a Board of Examiners composed by:

Prof. Giovanni Ulivi (chairman)

Prof. Giancarlo Bigi

Prof. Giovanni Sparacino

Gradient-based methods with subspace acceleration for quadratic programming problems and applications

Ph.D. thesis. Sapienza – University of Rome

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

Version: February 10, 2019

Author's email: marco.viola@uniroma1.it

Dedicated to
my parents Francesco and Lucrezia

Acknowledgments

Firstly, I would like to express my sincere gratitude to Prof. Gerardo Toraldo for his continuous support, his patience, his kindness and generosity. My journey into the world of nonlinear optimization started thanks to him almost five years ago, at the end of my first year as a Master's student, and it is mainly thanks to his guidance that I got to this point. A special thanks goes also to Prof. Daniela di Serafino without whom this thesis, and most of the work I carried out during my PhD career, would not have been possible. I'm really glad that, after having her as teacher as well as thesis advisor for my Bachelor's degree, I have had the chance to work with her. Gerardo and Daniela taught me lots during these years and pushed me to improve myself day after day. Words cannot express how lucky I feel to have them as my mentors.

I would like to thank Prof. Zdeněk Dostál from the Technical University of Ostrava, who hosted me for a few weeks in Ostrava, and Dr. Daniel Robinson from the Johns Hopkins University, who hosted me for 5 months in Baltimore, giving me the chance to see beautiful cities, to know different cultures, and to make new friends that I will never forget.

My sincere thanks also goes to Prof. Stefano Lucidi and Prof. Massimo Roma, for their support during my experience as a PhD student in Rome at the Sapienza University.

I would like to thank my father Francesco, my mother Lucrezia, my brother Raffaele, my sister Cristina, and my twin sister Caterina for their endless encouragement and belief in me, even if I always struggle when it comes to explain what my research is about. Thanks to my little nephew Giulio, for the joy he brought into our lives.

Last but not the least, I would like to thank my amazing girlfriend Nunzia. It is always sweet, but also kind of funny, to tell people that she, a girl from Naples, and I, a guy from Caserta (40 mins by car far from Naples), met each other in Edinburgh during a week-long course in our first year as PhD students respectively in Canterbury and in Rome. It's a small world after all. Thank you, Nunzia, for coming into my life, and thank you for all the love and support you've given me.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Quadratic Programming problems

We are concerned with the development of efficient *first-order* methods for the solution of Quadratic Programming problems (QPs), i.e. problems of the form

$$\begin{aligned} \min \quad & f(\mathbf{x}) := \frac{1}{2}\mathbf{x}^T H\,\mathbf{x} - \mathbf{c}^T\mathbf{x}, \\ \text{s.t.} \quad & A\,\mathbf{x} = \mathbf{b}, \\ & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \end{aligned} \tag{1.1}$$

where $H \in \mathbb{R}^{n\times n}$ is symmetric, $\mathbf{c} \in \mathbb{R}^n$, $A = (\mathbf{a}_1, \ldots, \mathbf{a}_m)^T \in \mathbb{R}^{m\times n}$, such that $m < n$ and $\text{rank}(A) = m$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{l} \in \{\mathbb{R} \cup \{-\infty\}\}^n$, $\mathbf{u} \in \{\mathbb{R} \cup \{+\infty\}\}^n$, and $l_i < u_i$ for all i. If, for all i, $l_i = 0$ and $u_i = +\infty$, the problem is said to be in *standard form*. It is worth noting that, by a change of variables and by introducing slack variables, every QP problem can be reduced to form (1.1).

Since the feasible set of (1.1)

$$\Omega := \{\mathbf{x} \in \mathbb{R}^n \,:\, A\,\mathbf{x} = \mathbf{b} \,\wedge\, \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}$$

is convex, if H is positive definite the problem is strictly convex and the unique local minimizer $\mathbf{x}^*$ is also a global minimizer and corresponds to the unique stationary point. However, we do not assume that the problem is strictly convex. In this case the problem can admit multiple local minimizers and f can also be unbounded from below. If H is positive semi-definite the problem is still convex but multiple local minimizers may exist, all corresponding to the same global minimum. In this case, if the problem is bounded, a local optimization method is still able to find the global minimum of (1.1). Finally, if H has at least one negative eigenvalue the problem is non-convex, different local minimizers may exist, and finding the global minimum becomes an NP-complete problem; actually, even verifying the local optimality of a given point is an NP-hard problem [112].

Many problems in science can be formulated as QPs. A well-known application is, e.g., portfolio optimization, which, according to the kind of model one wants to solve, can lead to problems with bound constraints and either one [114] or two [30, 71] linear constraints. Problems with a single linear constraint arise also in the dual formulation of support vector machine training [122], multicommodity

network flow and logistics [95], statistics estimates from a target distribution [2] and image reconstruction [92] (in this case the linear constraint expresses the so-called flux-conservation property [9]). A variety of applications lead to problems with bound constraints only, e.g., sparse signal reconstruction [68], contact and friction problems in rigid body mechanics [103], elastic-plastic torsion problems [26], journal bearing lubrication, flows through a porous medium [101]. As regards general QPs, they arise, e.g., in asset and liability management problems [81] and in various optimization problems on graphs [88], such as the quadratic assignment problem, the edge separator problem, the maximum clique problem [115], and the maximum independent set problem. Other applications and references can be found, e.g., in [114, 72]. In Chapter 5 we will see how QPs arise from a dual formulation of frictionless 3D contact mechanics problems and 2D contact mechanics problems with friction [57, 63]. QPs arise also as computational kernel of methods for general nonlinear optimization problems, e.g., in Sequential Quadratic Programming (SQP) [19, 111], in augmented Lagrangian methods [32, 3, 36, 5], or in branch-and-bound methods for mixed-integer quadratic programming problems, exploiting continuous relaxation and duality [25].

Due to the wide application range of QPs, designing efficient methods for their solution is still of great interest. In particular, first-order methods for QPs are preferable in the solution of large-scale and huge-scale problems thanks to their low iteration cost, low memory storage, and easiness of implementation. Furthermore, they can also be extended to the minimization of more general nonlinear functions.

We are interested in *gradient projection* methods, which are the generalization to the case of constrained optimization of the gradient descent methods for the solution of unconstrained optimization problems. As suggested by their name, at each step, gradient projection methods combine a descent along the gradient with a projection onto the feasible set to preserve feasibility. In particular, we focus our analysis on methods exploiting *subspace acceleration*, i.e. methods whose convergence is accelerated thanks to the introduction of steps in which the restriction of the problem onto a linear subspace is (approximately) solved. The methods under analysis belong to the wider class of *active-set* methods, i.e. methods which aim at identifying the constraints which are satisfied with equality (called *active*) at the solution. Under standard conditions on the regularity of the solution, gradient projection methods are indeed able to identify the active constraints in a finite number of iterations, provided that some sufficient decrease conditions are satisfied. Whereas in classical active-set methods the active set changes slowly, usually by a single index at each iteration, gradient projection methods are able to add/remove multiple constraints to/from the active set at each step, ensuring usually a faster identification.

Gradient projection methods are strictly related to the possibility of projecting a point onto the feasible set Ω. This makes them unpractical in the case of general linear constraints, for the computation of the projection is almost as expensive as the solution of the whole problem. There are, however, some exceptions. First of all, two particular cases of problem (1.1) for which gradient projection methods have been extensively studied and applied, thanks to the low cost of the projections onto the feasible set, are:

- the class of problems subject to bound constraints only, which we associate to

the case $m = 0$, and usually indicated as BQPs (or BCQPs);

- the class of problems subject to a single linear constraint and bound constraints, corresponding to the case $m = 1$, and usually indicated as SLBQPs (Single Linearly and Bound constrained Quadratic Programs).

It is clear that in the case of bound constraints the projection can be computed in $\mathcal{O}(n)$ operations; in Chapter 2 we will see that the projection onto the feasible set of SLBQPs can be computed cheaply, and that there actually exist algorithms with a theoretical complexity of $\mathcal{O}(n)$ operations. Recently, in [91], an efficient projection algorithm has been proposed for the case of problems of the form (1.1) in which the matrix A is a sparse matrix. This has opened the possibility to extend to this case theoretical results and numerical methods developed for BQPs and SLBQPs.

1.2 Contributions of this thesis

In this work we propose an active-set framework, called *Proportionality-based Subspace Accelerated framework for Quadratic Programming* (PSAQP), for the solution of problems of the form (1.1) based on gradient projection.

In Chapter 3, starting from a componentwise reformulation of the first-order optimality conditions for problem (1.1), we provide a definition of *binding set* at a point $\mathbf{x}$, generalizing the one used for BQPs, and we obtain a way of computing Lagrange multiplier estimates which, under standard regularity assumptions on the solution, are proved to converge to the optimal multipliers. This allows us to define suitable generalizations of the *free gradient* $\boldsymbol{\varphi}$ and the *chopped gradient* $\boldsymbol{\beta}$ at a point $\mathbf{x}$, introduced in [74, 75, 53] for the case of BQP problems. We prove that the defined quantities satisfy the following properties:

- vector $\boldsymbol{\varphi}(\mathbf{x})$ provides a measure of optimality within the space defined by the active constraints at $\mathbf{x}$;
- vector $\boldsymbol{\beta}(\mathbf{x})$ provides a measure of bindingness of the active set at $\mathbf{x}$;
- the two vectors are orthogonal and their sum is the projected gradient at $\mathbf{x}$, thus a point $\mathbf{x}$ is optimal if and only if both $\boldsymbol{\varphi}(\mathbf{x}) = \boldsymbol{\beta}(\mathbf{x}) = \mathbf{0}$.

This enables us to extend to problem (1.1) the concept of *proportional iterate*, henceforth also refereed to as *proportionality*, introduced in [14, 53] for BQPs. The PSAQP framework is based on the two-phase framework introduced by Calamai and Moré [27] and uses the gradient projection to identify the active set at the solution. Like in the GPCG method developed by Moré and Toraldo (for strictly convex BQPs)[109], in PSAQP we alternate gradient projection steps (*identification phase*) with minimization steps onto the reduced subspace defined by the current active set (*minimization phase*). The availability of the concept of proportional iterates for general QPs translates into the possibility of switching between the two phases by comparing a measure of optimality within the reduced space with a measure of the quality of the current active set. Furthermore, we are able to prove finite convergence of any method fitting into the proposed framework for strictly

convex quadratic problems even in case of degeneracy at the solution, provided that a method with finite termination properties is used in the minimization phase.

In Chapter 4 (based on [51]) we introduce a two-phase gradient projection method, called *Proportionality-based 2-phase Gradient Projection* (P2GP), for the solution of both SLBQP and BQP problems, which can be considered a specialization of the PSAQP framework. Besides targeting problems more general than strictly convex BQPs, the new method differs from GPCG because it exploits the concept of proportional iterate to decide when to terminate optimization in the reduced space. This change makes a significant difference in the effectiveness of the algorithm as our numerical experiments show. Moreover, according to the properties of PSAQP, the application of proportionality allows us to state *finite convergence for strictly convex problems* also for dual-degenerate solutions, whereas GPCG exhibits finite termination only in the case of non-degenerate solutions. The implementation of P2GP can also deal with *non-convex problems*. In this case, if the objective function is bounded, the algorithm converges to a stationary point as a result of a suitable application of the gradient projection method in the identification phase; otherwise, it is able to detect unboundedness and stop the computation. By exploiting a Householder transformation, we are able to reformulate the equality constrained subproblem of the minimization phase into an unconstrained quadratic problem whose conditioning is not worse that the one of the original problem. This reformulation allows one to exploit, for the solution of the subproblem, not only the conjugate gradient method, which guarantees finite termination, but also other methods for unconstrained QPs. In particular, we propose the use of spectral gradient methods, such as SDC [45] and SDA [47], which have proved to be efficient also in the solution of some ill-posed problems [46, 35]. In our opinion, P2GP can provide a way to exploit these methods and their regularizing properties when solving linear ill-posed problems with bounds and a single linear constraint. Furthermore, in Section 4.3.1, we introduce a novel procedure for the creation of SLBQPs with different sizes, spectral properties and levels of degeneracy, which can be used as a benchmark to test optimization algorithms for this class of problems.

Motivated by the good numerical performance of P2GP in the solution of BQPs, in Chapter 5 (based on [66]) we test the algorithm against the MPRGP algorithm developed by Dostál and Schöberl [54, 65] as a solver for the bound constrained subproblems arising in the solution of problems of the form (1.1) with an augmented Lagrangian algorithm called SMALBE [56]. First, we perform some tests on elliptic model problems representing the equilibrium of a 2D membrane. Then, we show how the discretization of contact mechanics problems leads to problem of the form (1.1) and compare the two algorithms in the solution of some 2D contact problems and 3D frictionless contact problems. The results show that P2GP is competitive with MPRGP and, thanks to the identification properties of the gradient projection, it can outperform MPRGP when the number of active constraints at the solution is high.

1.3 Notations and first definitions

Scalars are denoted by lightface Roman fonts (both Latin and Greek scripts), e.g., $a, \alpha \in \mathbb{R}$, vectors by boldface Roman fonts (both Latin and Greek scripts), e.g., $\mathbf{v}, \boldsymbol{\lambda} \in \mathbb{R}^n$, and matrices by italicized lightface capital fonts, e.g., $M \in \mathbb{R}^{m \times n}$. The vectors of the standard basis of $\mathbb{R}^n$ are denoted by $\mathbf{e}_1, \ldots, \mathbf{e}_n$, and the identity matrix of size n is denoted by I_n, where the subscript can be omitted if the size of the matrix is clear from the context.

For any finite set $\mathcal{S}$, $|\mathcal{S}|$ denotes its cardinality.

Given $\mathbf{v} \in \mathbb{R}^n$ and $\mathcal{C} \subseteq \{1, \ldots, n\}$, we set

$$\mathbf{v}_{\mathcal{C}} := (v_i)_{i \in \mathcal{C}} \in \mathbb{R}^{|\mathcal{C}|},$$

where v_i is the ith entry of $\mathbf{v}$. In a similar manner, given a matrix $M \in \mathbb{R}^{m \times n}$ and the subsets of indices $\mathcal{S} \subseteq \{1, \ldots, m\}$ and $\mathcal{C} \subseteq \{1, \ldots, n\}$, we set

$$M_{\mathcal{S}\mathcal{C}} := (m_{ij})_{i \in \mathcal{S}, j \in \mathcal{C}} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{C}|},$$

where m_{ij} the (i, j)th entry of M. We make use of the symbol "$\star$" when the given subset coincides with the whole set of indices, e.g., $M_{\star\mathcal{C}}$ and $M_{\mathcal{S}\star}$. We use the notation $M^T_{\mathcal{S}\mathcal{C}}$ in place of $[M_{\mathcal{S}\mathcal{C}}]^T$. For any vector $\mathbf{v}$, $\{\mathbf{v}\}^\perp$ is the space orthogonal to $\mathbf{v}$, and, for any matrix M, $\{M\}^\perp$ is the space orthogonal to the rows of M, i.e. the null space of M, indicated as $\mathcal{N}(M)$. For any symmetric matrix M, we use $\kappa(M)$, $\zeta_{min}(M)$ and $\zeta_{max}(M)$ to indicate the condition number, and the minimum and maximum eigenvalue of M, respectively. Norms $\|\cdot\|$ are ℓ_2, unless otherwise stated; moreover, we use $\langle \cdot, \cdot \rangle$ to indicate the inner product of $\mathbb{R}^n$.

We use superscripts to denote the elements of a sequence, e.g. $\left\{\mathbf{x}^k\right\}$.

Given a point $\mathbf{x} \in \Omega$, and supposing to associate a unique index to each of the $m + 2n$ constraints, the *active set at* $\mathbf{x}$ is usually defined as the set of indices corresponding to the constraints which are active at $\mathbf{x}$. This means that the active set includes the m indices corresponding to the linear equality constraints and the indices of the bound constraints which are active at $\mathbf{x}$. Since we are interested in feasible algorithms, the linear equality constraints will always be satisfied, therefore, our definition of active set will only involve the bound constraints. Furthermore, since we assumed $l_i < u_i$ for all i, the constraints $x_i \geq l_i$ and $x_i \leq u_i$ cannot be active at the same time; we can therefore define the active set as a subset of $\{1, \ldots, n\}$ containing the indices corresponding to the variables which are either on their lower or on their upper bound.

Definition 1.3.1. *We define the following index sets:*

$$\begin{aligned} &\mathcal{A}_l(\mathbf{x}) := \{i \,:\, x_i = l_i\}, \qquad &&\mathcal{A}_u(\mathbf{x}) := \{i \,:\, x_i = u_i\}, \\ &\mathcal{A}(\mathbf{x}) := \mathcal{A}_l(\mathbf{x}) \cup \mathcal{A}_u(\mathbf{x}), \qquad &&\mathcal{F}(\mathbf{x}) := \{1, \ldots, n\} \setminus \mathcal{A}(\mathbf{x}). \end{aligned}$$

$\mathcal{A}(\mathbf{x})$ *and* $\mathcal{F}(\mathbf{x})$ *are called respectively the* active set *and the* free set *at* $\mathbf{x}$.

Given $\mathbf{x}, \mathbf{y} \in \Omega$, by writing $\mathcal{A}(\mathbf{x}) \subseteq \mathcal{A}(\mathbf{y})$ we mean that

$$\mathcal{A}_l(\mathbf{x}) \subseteq \mathcal{A}_l(\mathbf{y}), \quad \mathcal{A}_u(\mathbf{x}) \subseteq \mathcal{A}_u(\mathbf{y})$$

both hold.

Definition 1.3.2. *For any* $\mathbf{x} \in \Omega$*, we define the following sets:*

$$\Omega(\mathbf{x}) := \{\mathbf{v} \in \mathbb{R}^n : A\,\mathbf{v} = \mathbf{b} \ \wedge \ v_i = x_i \ \forall\, i \in \mathcal{A}(\mathbf{x})\}, \tag{1.2}$$

$$\Omega_0(\mathbf{x}) := \{\mathbf{v} \in \mathbb{R}^n : A\,\mathbf{v} = \mathbf{0} \ \wedge \ v_i = 0 \ \forall\, i \in \mathcal{A}(\mathbf{x})\}. \tag{1.3}$$

We note that $\Omega(\mathbf{x})$ *corresponds to the* affine closure of the face *determined by the active set at* $\mathbf{x}$ *and* $\Omega_0(\mathbf{x})$ *is its support.*

To ease the notations, given $\mathbf{x}$, $\mathbf{x}^k$, $\overline{\mathbf{x}}$, and $\mathbf{x}^*$, we use

$$\begin{gathered} f^k \equiv f(\mathbf{x}^k), \quad \overline{f} \equiv f(\overline{\mathbf{x}}), \quad f^* \equiv f(\mathbf{x}^*), \\ \mathbf{g} \equiv \mathbf{g}(\mathbf{x}) \equiv \nabla f(\mathbf{x}), \quad \mathbf{g}^k \equiv \nabla f(\mathbf{x}^k), \quad \overline{\mathbf{g}} \equiv \nabla f(\overline{\mathbf{x}}), \quad \mathbf{g}^* \equiv \nabla f(\mathbf{x}^*), \\ \mathcal{A} \equiv \mathcal{A}(\mathbf{x}), \quad \mathcal{A}^k \equiv \mathcal{A}(\mathbf{x}^k), \quad \overline{\mathcal{A}} \equiv \mathcal{A}(\overline{\mathbf{x}}), \quad \mathcal{A}^* \equiv \mathcal{A}(\mathbf{x}^*), \\ \mathcal{F}^k \equiv \mathcal{F}(\mathbf{x}^k), \quad \mathcal{F} \equiv \mathcal{F}(\mathbf{x}), \quad \overline{\mathcal{F}} \equiv \mathcal{F}(\overline{\mathbf{x}}), \quad \mathcal{F}^* \equiv \mathcal{F}(\mathbf{x}^*). \end{gathered}$$

Finally, we recall the definition of orthogonal projection onto a non-empty closed convex set.

Definition 1.3.3. *Given a non-empty closed convex set* $\Theta \subset \mathbb{R}^n$ *and a point* $\mathbf{y} \in \mathbb{R}^n$*, the* orthogonal projection *of* $\mathbf{y}$ *onto* Θ *is defined as*

$$P_\Theta(\mathbf{y}) = \operatorname*{argmin}_{\mathbf{z} \in \Theta} \|\mathbf{z} - \mathbf{y}\|.$$

Chapter 2

Background and state of the art

Here we recall definitions, results and methods for the solutions of QPs which will be useful for the development of the following chapters of this work. We start by discussing spectral gradient methods for the solution of unconstrained QPs. Then, we recall optimality conditions for problems of the form (1.1), and present the gradient projection methods. We describe some two-phase gradient projection methods for the solution of BQPs and present the concept of free gradient, chopped gradient and proportional iterate for the case of BQPs. Finally, we describe methods for the projection of a point onto a polyhedron.

2.1 Gradient methods for unconstrained QPs

Consider the unconstrained quadratic programming problem

$$\min f(x) := \frac{1}{2}\mathbf{x}^T H\mathbf{x} - \mathbf{c}^T\mathbf{x}, \tag{2.1}$$

where $\mathbf{c} \in \mathbb{R}^n$ and $H \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix with eigenvalues $\zeta_1 > \zeta_2 \geq \ldots, \zeta_{n-1} > \zeta_n > 0$. The *gradient descent method* for the solution of (2.1) builds up a sequence of points $\{\mathbf{x}^k\}$ where, at each step,

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha^k \mathbf{g}^k,$$

with α^k a suitable step length. The most common choice for α^k is the *Cauchy step length* [29] which gives rise to the well-known *steepest descent* (SD) method. At each step, the step length α^k_{SD} is chosen as the unconstrained minimizer of the 1-dimensional problem

$$\alpha^k_{\mathrm{SD}} = \operatorname*{argmin}_{\alpha} f(\mathbf{x}^k - \alpha\mathbf{g}^k)$$

which leads to

$$\alpha^k_{\mathrm{SD}} = \frac{(\mathbf{g}^k)^T\mathbf{g}^k}{(\mathbf{g}^k)^T H\mathbf{g}^k}.$$

It can be proved (see [1]) that the sequence $\{\mathbf{x}^k\}$ generated by the SD method converges Q-linearly to the solution $\mathbf{x}^*$ to (2.1), with rate of convergence

$$\rho = \frac{\zeta_1 - \zeta_n}{\zeta_1 + \zeta_n} = \frac{\kappa(H) - 1}{\kappa(H) + 1}.$$

By following [47] and [50], we will now show how it is possible to improve the behavior of the standard method by step-length selection strategies which exploit informations on the spectrum of the Hessian. The derived methods are usually referred to as *spectral gradient methods*. Let $\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n$ be an orthonormal basis of eigenvectors of H, with $\mathbf{d}_i$ associated with the eigenvalue ζ_i. At the starting point $\mathbf{x}^0$ we have that

$$\mathbf{g}^0 = \sum_{i=1}^{n} \mu_i^0 \mathbf{d}_i,$$

where, w.l.o.g., we can assume μ_1^0 and μ_n^0 to be different from zero. We will refer to the scalars μ_i^k as the *eigencomponents* of the gradient at the step k. Since

$$\mathbf{g}^{k+1} = \mathbf{g}^k - \alpha^k H \mathbf{g}^k = \prod_{j=0}^{k} (I - \alpha^j H)\mathbf{g}^0, \tag{2.2}$$

we have

$$\mathbf{g}^{k+1} = \sum_{i=1}^{n} \mu_i^{k+1} \mathbf{d}_i, \quad \text{where } \mu_i^{k+1} = \mu_i^0 \prod_{j=0}^{k} (1 - \alpha^j \zeta_i) = \mu_i^k (1 - \alpha^k \zeta_i). \tag{2.3}$$

This relation suggests that, if at the $k-$th iteration $\mu_i^k = 0$ for some i, then for $h > k$ it will be $\mu_i^h = 0$; moreover, $\mu_i^{k+1} = 0$ if $\alpha^k = \frac{1}{\zeta_i}$. Furthermore, the SD method has finite termination if and only if at some iteration the gradient is an eigenvector of H. Relation (2.3) provides also other useful information on the effect of each step on the eigencomponents of the gradient. Indeed, if the step length satisfies $\alpha^k \approx \frac{1}{\zeta_i}$, for some i, then

$$\begin{cases} |\mu_i^{k+1}| \ll |\mu_i^k|, \\ |\mu_j^{k+1}| < |\mu_j^k|, & \text{if } j > i, \\ |\mu_j^{k+1}| > |\mu_j^k|, & \text{if } j < i \text{ and } \zeta_j > 2\zeta_i. \end{cases}$$

This suggests that small values of α^k can reduce the eigencomponents associated with large eigenvalues while increasing those associated with small eigenvalues. The vice-versa happens when α^k is large.

As shown in [110], the SD method tends to reduce the gradient eigencomponents corresponding to the largest and smallest eigenvalues more slowly than the other components. This eventually leads to a scenario in which the eigencomponents associated with $\mathbf{d}_2, \ldots, \mathbf{d}_{n-1}$ become negligible and the method turns into a minimization in to the 2-dimensional space spanned by $\mathbf{d}_1$ and $\mathbf{d}_n$. In particular, we have that

$$\lim_{k\to\infty} \frac{\mathbf{g}^{2k}}{\|\mathbf{g}^{2k}\|} = \mathbf{p}_1, \qquad \lim_{k\to\infty} \frac{\mathbf{g}^{2k+1}}{\|\mathbf{g}^{2k+1}\|} = \mathbf{p}_2,$$

with $\mathbf{p}_1, \mathbf{p}_2 \in \text{span}\{\mathbf{d}_1, \mathbf{d}_n\}$, i.e. the method assumes the well-known zigzagging behavior which generally yields to slow convergence.

2.1.1 The Barzilai-Borwein methods

The first work to analyze the possibility of introducing second-order information in the step-length selection was the seminal paper by Barzilai and Borwein [6]. By

setting $\mathbf{s}^{k-1} = \mathbf{x}^k - \mathbf{x}^{k-1}$ and $\mathbf{y}^{k-1} = \mathbf{g}^k - \mathbf{g}^{k-1}$, the authors introduced two step lengths satisfying a secant condition similar to the one used in the quasi-Newton methods, namely

$$\alpha_{BB1}^k = \operatorname*{argmin}_{\alpha} \left\| \alpha^{-1}\mathbf{s}^{k-1} - \mathbf{y}^{k-1} \right\|$$

and

$$\alpha_{BB2}^k = \operatorname*{argmin}_{\alpha} \left\| \mathbf{s}^{k-1} - \alpha\mathbf{y}^{k-1} \right\|.$$

These conditions lead respectively to the step lengths

$$\alpha_{BB1}^k = \frac{\|\mathbf{s}^{k-1}\|^2}{(\mathbf{s}^{k-1})^T\mathbf{y}^{k-1}} = \frac{(\mathbf{g}^{k-1})^T\mathbf{g}^{k-1}}{(\mathbf{g}^{k-1})^T H\mathbf{g}^{k-1}} \equiv \alpha_{\mathrm{SD}}^{k-1}, \tag{2.4}$$

$$\alpha_{BB2}^k = \frac{(\mathbf{s}^{k-1})^T\mathbf{y}^{k-1}}{\|\mathbf{y}^{k-1}\|^2} = \frac{(\mathbf{g}^{k-1})^T H\mathbf{g}^{k-1}}{(\mathbf{g}^{k-1})^T H^2\mathbf{g}^{k-1}}, \tag{2.5}$$

satisfying the relation

$$\frac{1}{\zeta_1} \leq \alpha_{BB2}^k \leq \alpha_{BB1}^k \leq \frac{1}{\zeta_n}.$$

We will refer to α_{BB1}^k as the *BB1 step length*, to α_{BB2}^k as the *BB2 step length* and to both of them as the *BB step lengths*. The two gradient methods derived from the application of α_{BB1}^k and α_{BB2}^k will be referred to respectively as the *BB1 method* and the *BB2 method*, and, together, as the *BB methods*. Even if it has been proved [40] that the BB methods have only an R-linear convergence rate, it is known that they are much faster than the standard SD method. This is possibly due to the fact that in BB methods the quantity $\frac{1}{\alpha^k}$ swipes the whole spectrum of H [69], thus avoiding the method to cycle in the "final" two-dimensional space.

2.1.2 Generalizations of BB methods

One powerful feature of BB step lengths is that they depend only on the difference between successive gradients ($\mathbf{y}^{k-1}$) and the difference between successive iterates ($\mathbf{s}^{k-1}$), thus they are well defined for a generic nonlinear smooth function. This led Raydan to analyze in [118] their extension to general nonlinear unconstrained minimization problems. Since it can be shown that both the BB1 and the BB2 step lengths may generate non-monotone iterates, Raydan included in his methods the well-known non-monotone line search from Grippo, Lampariello and Lucidi [84], usually indicated as GLL. Given an integer M, the GLL line search requires, at each step, that the function value at $\mathbf{x}^{k+1}$ satisfies

$$f(\mathbf{x}^{k+1}) \leq f_r + \mu(\mathbf{g}^k)^T(\mathbf{x}^{k+1} - \mathbf{x}^k),$$

where $\mu \in (0, 1)$ and

$$f_r = \max_{0 \leq j \leq M} f(\mathbf{x}^{k-j}).$$

This strategy, which for $M = 0$ corresponds to the standard Armijo line search, allows the objective function to increase at some iterations and still guarantees global convergence.

Various methods based on the alternation of the BB1 and the BB2 step lengths, or their modifications, have been proposed in literature, as in [85, 37, 39]. Among these, the one based on the $\text{ABB}_{\min}$ rule proposed in [73] proved to be very efficient in practice. The $\text{ABB}_{\min}$ step length is defined at each step as

$$\alpha^k_{\text{ABB}_{\min}} = \begin{cases} \min\left\{\alpha^j_{\text{BB2}} \,:\, j = \max\{1, k-s\}, \ldots, k\right\}, & \text{if } \dfrac{\alpha^k_{\text{BB2}}}{\alpha^k_{\text{BB1}}} < \tau, \\ \alpha^k_{\text{BB1}}, & \text{otherwise,} \end{cases} \tag{2.6}$$

where s is a non-negative integer and $\tau \in (0, 1)$. The switch between the two steps is based on the value

$$\frac{\alpha^k_{\text{BB2}}}{\alpha^k_{\text{BB1}}} = cos^2(\theta^{k-1}),$$

where θ^{k-1} is the angle between $\mathbf{g}^{k-1}$ and $H\,\mathbf{g}^{k-1}$, and allows the algorithm to select α^k_{BB1} when $\mathbf{g}^{k-1}$ is a sufficiently good approximation of an eigenvector of H (see [73, 50] for further details). A modification of (2.6) can be found in [21], where the authors proposed a strategy in which the fix scalar τ is replaced by an adaptive scalar τ_k which, starting from a given τ_0, is updated at each step by the rule

$$\tau_{k+1} = \begin{cases} 0.9 \cdot \tau_k, & \text{if } \dfrac{\alpha^k_{\text{BB2}}}{\alpha^k_{\text{BB1}}} < \tau_k, \\ 1.1 \cdot \tau_k, & \text{otherwise.} \end{cases}$$

Inspired by the fact that

$$\alpha^k_{BB1} = \alpha^{k-1}_{\text{SD}}, \tag{2.7}$$

Raydan and Svaiter proposed in [119] a gradient method which computes, at each step k, the Cauchy step length and then uses it twice. In detail, the method, called *Cauchy-Barzilai-Borwein* (CBB), at each iteration computes the vector $\mathbf{y}^k = \mathbf{x}^k - \alpha^k_{\text{SD}}\mathbf{g}^k$ and then sets $\mathbf{x}^{k+1} = \mathbf{y}^k - \alpha^k_{\text{SD}}\nabla f(\mathbf{y}^k)$. It can be shown that this is equivalent to set

$$\mathbf{x}^{k+1} = \mathbf{x}^k - 2\,\alpha^k_{\text{SD}}\mathbf{g}^k + \left(\alpha^k_{\text{SD}}\right)^2 H\,\mathbf{g}^k.$$

The authors proved that the sequence $\{\mathbf{x}^k\}$ generated by CBB converges to the solution $\mathbf{x}^*$ of (2.1); moreover, recalling that the norm induced by the symmetric positive definite matrix H^{-1} is defined as

$$\|\mathbf{v}\|_{H^{-1}} = \sqrt{\mathbf{v}^T H^{-1}\mathbf{v}},$$

the authors proved that the sequence

$$\left\{\|\mathbf{x}^k - \mathbf{x}^*\|_{H^{-1}}\right\}$$

converges Q-linearly to 0.

Relation (2.7), which allows one to refer to the BB1 method as a *gradient method with retard*, was also at the basis of the work by Friedlander et al. [76]. The authors investigated a generalization of the BB1 method in which at each step the steplength α^k is taken as the SD step at a previous iterate $\nu_k \in \{k,\, k-1,\, \ldots,\, \max\{0, k-s\}\}$, where s is a given positive integer. Observe that the method corresponds to the

classical SD when $s = 0$ and to the BB1 method when $s = 1$. Consider the sequence generated by the scheme

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha^k \mathbf{g}^k,$$

with $\alpha^k = \alpha_{\mathrm{SD}}^{\nu_k}$, and let $\mathbf{s}^{k-1} = \mathbf{x}^k - \mathbf{x}^{k-1}$. The authors proved that, if the sequence $\left\{\frac{\mathbf{s}^{k-1}}{\|\mathbf{s}^{k-1}\|}\right\}$ converges to a normalized vector $\mathbf{s} \in \mathbb{R}^n$, then

$$\lim_{k \to +\infty} \frac{1}{\alpha^k} = \sigma \equiv \mathbf{s}^T H \, \mathbf{s},$$

$\mathbf{s}$ is an eigenvector of H with eigenvalue σ, and the sequence $\{\mathbf{x}^k\}$ converges Q-superlinearly to the solution $\mathbf{x}^*$ of (2.1).

2.1.3 The RSD, SDC and SDA methods

Inspired by the work from Barzilai and Borwein [6], in 2002 Raydan and Svaiter [119] analyzed the use of over and under relaxation of the Cauchy step length to achieve better performances. Observe that, given $\mathbf{x}^k$ and $\mathbf{g}^k$, the function $f_k(\alpha) = f(\mathbf{x}^k - \alpha \mathbf{g}^k)$, which is a quadratic convex function in α whose unique minimum is obtained in α_{SD}^k, is such that

$$f_k(\alpha) \leq f(\mathbf{x}^k), \quad \text{for } \alpha \in \left[0,\, 2\,\alpha_{\mathrm{SD}}^k\right].$$

The authors proposed the *Relaxed Steepest Descent* (RSD) method which, at each step, uses $\alpha^k = \varsigma_k \alpha_{\mathrm{SD}}^k$, where ς_k is randomly chosen in $[0,\, 2]$. It has been proved that RSD converges to the optimal solution of (2.1) if the sequence $\{\varsigma_k\}$ admits an accumulation point $\overline{\varsigma} \in (0,\, 2)$. Even if the Cauchy step length is the best possible choice when the search direction is an eigenvector of the Hessian matrix, numerical experiments show that in practice the RSD method largely outperforms the SD method; however, it is still not able to outperform the BB methods. This led the authors to develop, in the same work, the CBB method presented in Section 2.1.2.

Other very efficient gradient methods for the solution of problem (2.1) are the SDC and the SDA method proposed respectively in [45] and [47].

The SDC method, whose name comes from *Steepest Descent with Constant (Yuan) step lengths*, is based on the following step length selection rule:

$$\alpha_{\mathrm{SDC}}^k = \begin{cases} \alpha_{\mathrm{SD}}^k & \text{if } \mathrm{mod}\,(k, m_s + m_c) < m_s, \\ \alpha_Y^t & \text{otherwise, with } t = \max\{i \leq k \,:\, \mathrm{mod}\,(i, m_s + m_c) = m_s\}, \end{cases} \tag{2.8}$$

where

$$\alpha_{\mathrm{Y}}^t = 2 \left(\sqrt{\left(\frac{1}{\alpha_{\mathrm{SD}}^{t-1}} - \frac{1}{\alpha_{\mathrm{SD}}^t}\right)^2 + 4 \frac{\|\mathbf{g}^t\|^2}{\left(\alpha_{\mathrm{SD}}^{t-1} \|\mathbf{g}^{t-1}\|\right)^2}} + \frac{1}{\alpha_{\mathrm{SD}}^{t-1}} + \frac{1}{\alpha_{\mathrm{SD}}^t} \right)^{-1} \tag{2.9}$$

is the Yuan step length [125]. In other words, the method performs m_s consecutive exact line searches and then, using the last two Cauchy step lengths α_{SD}^t and $\alpha_{\mathrm{SD}}^{t-1}$, computes the Yuan step length (2.9) and uses it for m_c consecutive iterations. The interest for SDC is motivated by its spectral properties, which dramatically speed

up the convergence [45, 50], while showing certain regularization properties useful to deal with linear ill-posed problems [46]. It has been proved in [45] that

$$\lim_{t\to\infty} \alpha_{\mathrm{Y}}^t = \frac{1}{\zeta_1}.$$

Therefore, when the SD starts minimizing in the 2-dimensional space spanned by $\mathbf{d}_1$ and $\mathbf{d}_n$, by (2.3), the use of the Yuan step length can lead to the solution of the unconstrained problem in only two steps. The alternation between the SD steps and the Yuan step aims at driving the minimization in the 2-dimensional space and, at the same time, to approximate the inverse of ζ_1. Observe that, if at some point $\alpha_{\mathrm{Y}}^t \approx \frac{1}{\zeta_1}$, the Yuan steps drives to zero the first eigencomponent. After that, the Yuan step lengths tend to approximate $\frac{1}{\zeta_2}$. This means that, ideally, SDC eliminates, one after another, the larger eigencomponents of $\mathbf{g}^k$ (starting from μ_1^k up to μ_n^k), until all the eigencomponents are zero and the stationary point is reached.

Similar regularization properties hold for the *Steepest Descent with Alignment* (SDA) method, in which the Yuan step length is replaced by the step length $\widetilde{\alpha}^t$ defined as

$$\widetilde{\alpha}^t = \left(\frac{1}{\alpha_{\mathrm{SD}}^{t-1}} + \frac{1}{\alpha_{\mathrm{SD}}^{t}} \right)^{-1},$$

which satisfies the property

$$\lim_{t\to\infty} \widetilde{\alpha}^t = \frac{1}{\zeta_1 + \zeta_n}.$$

In this case the alternation of the SD steps and the constant step aims at reaching the phase of 2-dimensional minimization and, at the same time, to align the gradient with $\mathbf{d}_n$.

2.1.4 The Limited Memory Steepest Descent method

Here, by following [51], we briefly describe the *Limited Memory Steepest Descent* (LMSD) introduced by Fletcher in [70]. The idea of the LMSD method is to divide the iterations into groups of size s, referred to as *sweeps*; at each step Ritz values of the Hessian [80], obtained by exploiting the gradients of the previous sweep, are used as step lengths for the current sweep.

In detail, consider, at the iteration $k \geq s$, the matrices $G \in \mathbb{R}^{n\times s}$ and $J \in \mathbb{R}^{(s+1)\times s}$ defined respectively as

$$G = \left(\mathbf{g}^{k-s}, \mathbf{g}^{k-s+1}, \ldots, \mathbf{g}^{k-1} \right),$$

and

$$J = \begin{pmatrix} \left(\alpha_{LMSD}^{k-s}\right)^{-1} & & \\ -\left(\alpha_{LMSD}^{k-s}\right)^{-1} & \ddots & \\ & \ddots & \left(\alpha_{LMSD}^{k-1}\right)^{-1} \\ & & -\left(\alpha_{LMSD}^{k-1}\right)^{-1} \end{pmatrix},$$

where α_{LMSD}^j is the step length associated with the direction $\mathbf{g}^j$. To ease the description, we will only consider the case in which G is full rank. Since, for each j,

we have that

$$\mathbf{g}^j = \mathbf{g}^{j-1} - \alpha_{LMSD}^{j-1} H\, \mathbf{g}^{j-1},$$

which is equivalent to

$$H\, \mathbf{g}^{j-1} = (\alpha_{LMSD}^{j-1})^{-1} \left(\mathbf{g}^{j-1} - \mathbf{g}^j\right),$$

we can write

$$H\, G = \left(G,\, \mathbf{g}^k\right)\, J.$$

Observe that, by applying s iterations of the Lanczos process to matrix H, starting from the vector $\mathbf{q}^1 = \frac{\mathbf{g}^{k-s}}{\|\mathbf{g}^{k-s}\|}$, we obtain the tridiagonal matrix $T = Q^T H\, Q \in \mathbb{R}^{s\times s}$, where $Q = (\mathbf{q}^1, \ldots, \mathbf{q}^s)$ has orthonormal columns spanning the vector subspace

$$\mathcal{S} = \text{span}\left\{\mathbf{g}^{k-s},\, H\mathbf{g}^{k-s},\, H^2\mathbf{g}^{k-s},\, \ldots,\, H^{s-1}\mathbf{g}^{k-s}\right\}.$$

Since the columns of G are vectors in $\mathcal{S}$, we can write $G = Q\, R$, where $R \in \mathbb{R}^{s\times s}$ is upper triangular and non-singular. This leads to

$$T = Q^T H\, G\, R^{-1} = \left(R,\, Q^T\mathbf{g}^k\right) J\, R^{-1}.$$

The LMSD method uses the s eigenvalues θ_i of T, which are known as Ritz values and provide s estimates of the eigenvalues of H, to determine the step length for the next s steps starting from k. In particular, the step lengths have the form

$$\alpha_{LMSD}^{k-1+i} = \frac{1}{\theta_i}, \quad i = 1, \ldots, s.$$

Observe that the derived method, which generates a non-monotone sequence converging to the solution of (2.1), for $s = 1$ corresponds to the BB1 method.

2.2 Stationarity conditions for QPs

We start by introducing the first-order Karush-Kuhn-Tucker (KKT) conditions.

Definition 2.2.1. *A point* $\mathbf{x}$ *is a* first-order KKT point *for problem* (1.1) *if there exist Lagrange multipliers vectors* $\boldsymbol{\theta} \in \mathbb{R}^m$ *and* $\boldsymbol{\lambda} \in \mathbb{R}^n$ *such that the triple* $(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\lambda})$ *satisfies*

$$A\,\mathbf{x} = \mathbf{b}, \qquad \mathbf{l} \le \mathbf{x} \le \mathbf{u}, \tag{2.10}$$

$$\mathbf{g} = \sum_{i=1}^{n} \lambda_i \mathbf{e}_i + \sum_{j=1}^{m} \theta_j \mathbf{a}_j = \sum_{i=1}^{n} \lambda_i \mathbf{e}_i + A^T\boldsymbol{\theta}, \tag{2.11}$$

$$\lambda_i \ge 0 \;\; \textit{if} \;\; i \in \mathcal{A}_l, \quad \lambda_i \le 0 \;\; \textit{if} \;\; i \in \mathcal{A}_u, \tag{2.12}$$

$$\lambda_i = 0 \;\; \textit{if} \;\; i \in \mathcal{F}, \tag{2.13}$$

Vector $\mathbf{x}$ is usually referred to as the *vector of primal variables*, whereas $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$ are referred to as the *vectors of dual variables*. Conditions (2.10) and (2.12) represent respectively the *primal feasibility* and the *dual feasibility*; (2.11) is referred to as the *stationarity condition*; finally, (2.13) is referred to as the *complementarity condition*. Since we are dealing with linearly constrained problems, stationary for problem (1.1) can be expressed in terms of the KKT conditions.

Definition 2.2.2. *A point* $\mathbf{x}^*$ *is a* stationary point *for problem* (1.1) *if and only if there exist Lagrange multipliers vectors* $\boldsymbol{\theta}^* \in \mathbb{R}^m$ *and* $\boldsymbol{\lambda}^* \in \mathbb{R}^n$ *such that the triple* $(\mathbf{x}^*, \boldsymbol{\theta}^*, \boldsymbol{\lambda}^*)$ *satisfies the KKT conditions* (2.10)-(2.13).

The algorithms considered here for the solution of QPs all aim at finding a stationary point $\mathbf{x}$. We recall that in the case of strictly convex problems ($H \succ 0$) problem (1.1) admits a unique stationary point, which coincides with the global minimizer of f over Ω.

Another way to express stationarity for problem (1.1) is by using the projected gradient. Given a point $\mathbf{x} \in \Omega$, the projected gradient of f at $\mathbf{x}$ is defined as the vector

$$\nabla_\Omega f(\mathbf{x}) := \operatorname{argmin} \{\|\mathbf{v} + \nabla f(\mathbf{x})\| \text{ s.t. } \mathbf{v} \in T_\Omega(\mathbf{x})\}, \tag{2.14}$$

where the *tangent cone* to Ω at $\mathbf{x}$ takes the form

$$T_\Omega(\mathbf{x}) = \{\mathbf{v} \in \mathbb{R}^n \,:\, A\,\mathbf{v} = 0 \;\wedge\; v_i \geq 0 \;\forall\, i \in \mathcal{A}_l(\mathbf{x}) \;\wedge\; v_i \leq 0 \;\forall\, i \in \mathcal{A}_u(\mathbf{x})\}.$$

We observe that $\mathbf{x}^* \in \Omega$ is a stationary point for (1.1) if and only if $\nabla_\Omega f(\mathbf{x}^*) = \mathbf{0}$, which is equivalent to

$$-\nabla f(\mathbf{x}^*) \in T_\Omega(\mathbf{x})^\circ,$$

where

$$T_\Omega(\mathbf{x})^\circ = \left\{\mathbf{w} \in \mathbb{R}^n \,:\, \mathbf{w}^T\mathbf{v} \leq 0 \;\forall\, \mathbf{v} \in T_\Omega(\mathbf{x})\right\}$$

is the polar of the tangent cone at $\mathbf{x}$, i.e. the *normal cone* to Ω at $\mathbf{x}$. By Farkas' Lemma, it can be shown that the normal cone has the form

$$\begin{aligned} T_\Omega(\mathbf{x})^\circ = \Big\{\mathbf{w} \in \mathbb{R}^n \;:\; -\mathbf{w} = \sum_{i \in \mathcal{A}(\mathbf{x})} \lambda_i \mathbf{e}_i + A^T\boldsymbol{\nu} \quad \wedge \\ \lambda_i \geq 0 \;\forall\, i \in \mathcal{A}_l(\mathbf{x}) \quad \wedge \quad \lambda_i \leq 0 \;\forall\, i \in \mathcal{A}_u(\mathbf{x}) \Big\}. \end{aligned} \tag{2.15}$$

2.3 Gradient Projection methods

Consider the constrained optimization problem

$$\begin{aligned} \min \quad & f(\mathbf{x}), \\ \text{s.t.} \quad & \mathbf{x} \in \Theta, \end{aligned} \tag{2.16}$$

where f is a continuously differentiable function and $\Theta \subset \mathbb{R}^n$ is a non-empty closed convex set.

The *gradient projection* (GP) method, introduced independently in the 60s by Goldstein [79] and by Levitin and Polyak [100], can be seen as a natural extension of the gradient descent method for unconstrained minimization. It is based on the simple iteration

$$\mathbf{x}^{k+1} = P_\Theta(\mathbf{x}^k - \alpha^k \mathbf{g}^k), \quad k = 0, 1, \ldots, \tag{2.17}$$

where α^k is a suitably chosen step length.

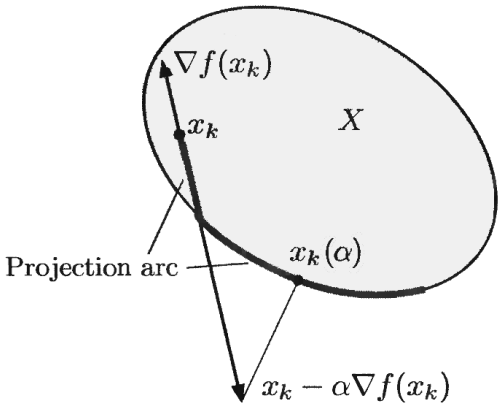

Figure 2.1. Illustration of the projection arc [12].

Definition 2.3.1. *Let* $\mathbf{x} \in \Theta$*, we define the* projection arc *as the set of points of the form*

$$\mathbf{x}(\alpha) = P_\Theta(\mathbf{x} - \alpha \mathbf{g}), \quad \alpha > 0.$$

The projection arc starts at $\mathbf{x}$ *and defines a curve continuously parametrized by* $\alpha \in \mathbb{R}^+$ *(see Figure 2.1).*

It can be proved that $\mathbf{x}^*$ is a stationary point for (2.16) if and only if

$$\mathbf{x}^* = P_\Theta(\mathbf{x}^* - \alpha \mathbf{g}^*), \quad \text{for some } \alpha > 0.$$

Observe that this condition guarantees that the gradient projection method will make no progress if $\mathbf{x}^k$ is stationary point.

Under the assumption that the gradient of f is Lipschitz continuous, i.e.,

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|, \quad \forall \mathbf{x}, \mathbf{y} \in \Theta,$$

where $L > 0$, Goldstein, Levitin and Polyak proved various convergence results in the case where α^k satisfies for all k the condition

$$0 < \epsilon \leq \alpha^k \leq \frac{2(1-\epsilon)}{L}.$$

Some years later McCormick [105] proved the convergence of a gradient projection method which did not require the Lipschitz continuity of the gradient of f. In particular he proposed, at each step, to take the step length α^k as

$$\alpha^k = \underset{\alpha \geq 0}{\operatorname{argmin}} \left\{ \psi_k(\alpha) := f\left(P_\Theta(\mathbf{x}^k - \alpha \mathbf{g}^k)\right) \right\}, \tag{2.18}$$

which, however, is not practical to compute for general constrained problems.

The first practical gradient projection method was proposed in 1976 by Bertsekas [10] for the case of a general non-negativity constrained problems, i.e. with f twice continuously differentiable and $\Theta = \{\mathbf{z} \in \mathbb{R}^n : \mathbf{z} \geq \mathbf{0}\}$. The method proposed by Bertsekas consists in choosing α^k by means of an *Armijo rule along the projection arc*. In detail, by setting

$$\mathbf{x}^k(\alpha) = P_\Theta(\mathbf{x}^k - \alpha \mathbf{g}^k),$$

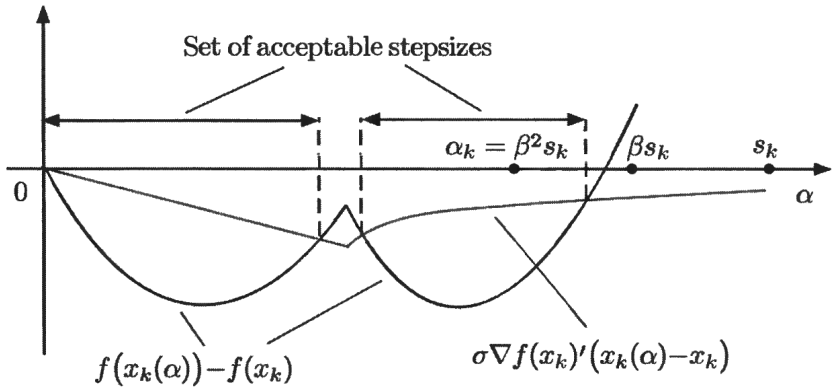

Figure 2.2. Illustration of the successive point tested by the Armijo rule along the projection arc [12].

Bertsekas proposed to take $\alpha^k = \beta^{s_k}\alpha_0$, where $0 < \beta < 1$, $\alpha_0 > 0$, and s_k is the minimum integer such that α^k satisfies the condition

$$f\left(\mathbf{x}^k(\alpha^k)\right) \leq f(\mathbf{x}^k) + \mu(\mathbf{g}^k)^T\left(\mathbf{x}^k(\alpha^k) - \mathbf{x}^k\right), \tag{2.19}$$

with $0 < \mu < 1$. The author proved that any limit point $\mathbf{x}^*$ of the sequence $\{\mathbf{x}^k\}$ generated by this algorithm is stationary. Moreover, he proved that if a stationary point $\mathbf{x}^*$ is non-degenerate and the matrix $\nabla^2 f(\mathbf{x}^*)$ is strictly positive definite, there exists $\delta > 0$ such that, if for some $\overline{k}$ it holds $\|\mathbf{x}^* - \mathbf{x}^{\overline{k}}\| < \delta$, then the sequence converges to $\mathbf{x}^*$ and the active-set at $\mathbf{x}^*$ is identified in a finite number of steps, i.e. there exists $\hat{k} > \overline{k}$ such that $\mathcal{A}^k = \mathcal{A}^*$, $\forall k > \hat{k}$.

One drawback of the selection rule proposed above is that it requires a projection onto the feasible region for each step reduction. This can lead to numerical inefficiency, especially in the case of more complicated constraints than the non-negativity constraints considered in the original proposal. In these cases an alternative to the Armijo rule along the projection arc is available. Given the positive scalar α_0, one can compute the feasible direction

$$\mathbf{p}^k = P_\Theta(\mathbf{x}^k - \alpha_0\mathbf{g}^k) - \mathbf{x}^k$$

and then perform a line search along $\mathbf{p}^k$, choosing $\mathbf{x}^{k+1}$ as

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \beta^{s_k}\mathbf{p}^k,$$

where $0 < \beta < 1$ and s_k is the minimum integer such that

$$f\left(\mathbf{x}^k + \beta^{s_k}\mathbf{p}^k\right) \leq f(\mathbf{x}^k) - \mu\beta^{s_k}(\mathbf{g}^k)^T\mathbf{p}^k,$$

with $0 < \mu < 1$. Performing this *line search along the feasible direction* $\mathbf{p}^k$ is less expensive, since it requires only one projection (the one performed to compute $\mathbf{p}^k$); the convexity of Θ ensures, indeed, that all the points of the form $\mathbf{x}^k + \gamma\mathbf{p}^k$, with $\gamma \in (0, 1)$ are feasible. While the line search along the projection arc usually returns points which lie on the boundary of Θ, the line search along the feasible direction

$\mathbf{p}^k$ is likely to return points which lie in its interior, which translates in a slower identification of the active constraints at the solution [13, 12].

The convergence results obtained by Bertsekas for his line-search based gradient projection method were extended by Dunn [67] for the general case of a convex set Θ and a differentiable function f.

In their seminal paper of 1987 [27], Calamai and Moré investigated the results on the convergence of gradient projection algorithms obtained by Bertsekas [10, 11] and Dunn [67], focusing on the case of linearly constrained nonlinear problems of the form

$$\begin{array}{ll} \min & f(\mathbf{x}), \\ \text{s.t.} & C\,\mathbf{x} \geq \mathbf{d}, \end{array} \tag{2.20}$$

where f is continuously differentiable, $C = (\mathbf{c}_1, \ldots, \mathbf{c}_m)^T \in \mathbb{R}^{m \times n}$, and $\mathbf{d} \in \mathbb{R}^m$. Observe that, in this case, the active set at a point $\mathbf{x}$ is defined as the set

$$\mathcal{A}(\mathbf{x}) = \left\{ i \ : \ \mathbf{c}_i^T \mathbf{x} = d_i \right\}.$$

Consider a non-degenerate stationary point $\mathbf{x}^*$ such that the set of the normals to the active constraints, i.e. the set $\{\mathbf{c}_i \ : \ i \in \mathcal{A}^*\}$, is linearly independent. The authors showed that, given any sequence $\{\mathbf{x}^k\}$ converging to $\mathbf{x}^*$, if $\left\{ \|\nabla_\Omega f(\mathbf{x}^k)\| \right\}$ converges to zero then there exists $\overline{k}$ such that $\mathcal{A}^k = A^*$ for each $k \geq \overline{k}$.

Given the current iterate $\mathbf{x}^k$, they proposed a gradient projection algorithm generating the successive iterate as

$$\mathbf{x}^{k+1} = P_\Omega(\mathbf{x}^k - \alpha^k \mathbf{g}^k),$$

where α^k satisfies the following sufficient decrease condition: given $\gamma_1, \gamma_2, \gamma_3 > 0$ and $\mu_1, \mu_2 \in (0, 1)$,

$$f^{k+1} \leq f^k + \mu_1 \, (\mathbf{g}^k)^T (\mathbf{x}^{k+1} - \mathbf{x}^k), \tag{2.21}$$

where

$$\begin{array}{c} \alpha^k \leq \gamma_1, \\ \alpha^k \geq \gamma_2 \ \text{ or } \ \alpha^k \geq \gamma_3 \, \overline{\alpha}^k > 0, \end{array} \tag{2.22}$$

with $\overline{\alpha}^k$ such that

$$f(\mathbf{x}^k(\overline{\alpha}^k)) > f^k + \mu_2 \, (\mathbf{g}^k)^T (\mathbf{x}^k(\overline{\alpha}^k) - \mathbf{x}^k), \tag{2.23}$$

where $\mathbf{x}^k(\overline{\alpha}^k) := P_\Omega(\mathbf{x}^k - \overline{\alpha}^k \mathbf{g}^k)$. It can be showed that the limit points of a bounded sequence $\{\mathbf{x}^k\}$ generated by such an algorithm are stationary; moreover, the sequence satisfies the condition

$$\lim_{k \to \infty} \|\nabla_\Omega f(\mathbf{x}^k)\| = 0. \tag{2.24}$$

The authors also proved that similar results hold for a more general family of algorithms (see [27, Algorithm 5.3]) which exploit the gradient projection only for an infinite subset $K \subset \mathbb{N}$ of the iterates while just requiring that all the other iterates satisfy the condition

$$f^{k+1} \leq f^k \quad \text{ and } \quad \mathcal{A}^k \subseteq \mathcal{A}^{k+1}.$$

2.3.1 Step-length selection

The convergence results proved by Calamai and Moré for the gradient projection method with Armijo line search is independent from the choice of the initial guess α_0. However, in Section 2.1, we have seen how, starting from the seminal paper of Barzilai and Borwein [6], efficient step length selection strategies exploiting second-order informations have been devised for gradient methods for unconstrained optimization. Some of these strategies have been successfully applied to the constrained case, giving rise to efficient methods with low computational cost.

Starting from the methods proposed by Raydan in [118], Birgin, Martínez and Raydan [16, 15, 17] developed for the case of constrained optimization the so-called *Spectral Projected Gradient* (SPG) methods. The two proposed methods, named SPG1 and SPG2, are based on the BB1 step length and exploit respectively the line search along the projection arc and the one along the feasible direction. These methods have been efficiently applied in a decomposition framework for the solution of support vector machine training problems by Serafini, Zanghirati and Zanni [120, 127] and in the solution of image segmentation problems [4]; moreover, they have been further analyzed by Dai and Fletcher [37, 38], who managed to build 2-dimensional BQPs in which projected BB methods without line search may cycle.

2.3.2 The Scaled Gradient Projection method

The *Scaled Gradient Projection* (SGP) method [21] is a variant of the classical gradient projection method, based on a the scaling of the descent direction by a positive definite matrix. In detail, at each step k, a search direction is computed as

$$\mathbf{p}^k = P_{\Theta,D_k}\left(\mathbf{x}^k - \alpha^k D_k^{-1}\mathbf{g}^k\right) - \mathbf{x}^k,$$

where D_k is a symmetric positive definite matrix whose eigenvalue lie in the interval $\left[\mu_k, \frac{1}{\mu_k}\right]$, with $\mu_k \geq 1$, and the projection operator $P_{\Theta,D}(\mathbf{z})$ is defined as

$$P_{\Theta,D}(\mathbf{z}) = \operatorname*{argmin}_{\mathbf{v}\in\Theta} \|\mathbf{v}-\mathbf{z}\|_D = \operatorname*{argmin}_{\mathbf{v}\in\Theta} \frac{1}{2}\mathbf{v}^T D\,\mathbf{v} - \mathbf{v}^T D\,\mathbf{z}.$$

It can be proved that, given a scalar $\alpha > 0$, a symmetric positive definite matrix D, and a point $\mathbf{x}^* \in \Theta$, then

$$\mathbf{p}^* = P_{\Theta,D}\left(\mathbf{x}^* - \alpha D^{-1}\mathbf{g}^*\right) - \mathbf{x}^*$$

is a feasible descent direction and $\mathbf{x}^*$ is stationary if and only if $\mathbf{p}^* = \mathbf{0}$. Once $\mathbf{p}^k$ is computed, a line search along it is performed, and $\mathbf{x}^{k+1}$ is taken as the point

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \beta^{s_k}\mathbf{p}^k,$$

where $0 < \beta < 1$ and s_k is the minimum integer such that

$$f\left(\mathbf{x}^k + \beta^{s_k}\mathbf{p}^k\right) \leq f(\mathbf{x}^k) - \gamma\beta^{s_k}(\mathbf{g}^k)^T\mathbf{p}^k,$$

with $0 < \gamma < 1$. Thanks to the availability of cheap projection algorithms (see Section 2.5.1) these methods have been efficiently applied in the solution of inverse

problems in imaging science [7, 117, 102] subject either to bound constraints or to bound constraints and a single linear constraint. It has been proved in [20] that if the matrix D_k reduces asymptotically to the identity matrix, i.e. $\{\mu_k\}$ satisfies

$$\mu_k^2 = 1 + \varsigma_k, \quad \text{with } \varsigma_k \geq 0 \text{ and } \sum_{k=0}^{\infty} \varsigma_k < \infty,$$

then the algorithm converges to a solution $\mathbf{x}^*$ to (2.16); moreover, if the gradient of f is Lipschitz continuous,

$$f(\mathbf{x}^{k+1}) - f(\mathbf{x}^*) = \mathcal{O}(\frac{1}{k}).$$

As for the standard gradient projection algorithm, the practical performances of SGP methods are affected by the choice of the step length α^k and the scaling matrix D_k. In regards to the scaling, as suggested in [126], aiming at improving the convergence rate of the algorithm without increasing its computational cost, one could take at each step D_k as a diagonal matrix approximating the inverse of the Hessian matrix $\nabla^2 f(\mathbf{x}^k)$. By following [99], in [102] the authors propose, in the case of problems subject to non-negativity constraints, to consider the following splitting of the gradient

$$\nabla f(\mathbf{x}) = V(\mathbf{x}) - U(\mathbf{x}),$$

with $V(\mathbf{x}) > \mathbf{0}$ and $U(\mathbf{x}) \geq \mathbf{0}$, and to take D_k^{-1} as the matrix

$$D_k^{-1} = \text{diag}(d_1^k, \ldots, d_n^k),$$

with

$$d_i^k = \max\left\{\min\left\{\frac{x_i^k}{V_i(\mathbf{x}^k)}, \mu_k\right\}, \frac{1}{\mu_k}\right\}.$$

In regards to the step-length selection, suitable generalization of the BB step lengths have been defined in [21], for the case of SGP methods, as

$$\alpha_{BB1S}^k = \frac{(\mathbf{s}^{k-1})^T D_k D_k \mathbf{s}^{k-1}}{(\mathbf{s}^{k-1})^T D_k \mathbf{y}^{k-1}}, \quad \text{and} \quad \alpha_{BB2S}^k = \frac{(\mathbf{s}^{k-1})^T D_k^{-1} \mathbf{y}^{k-1}}{(\mathbf{y}^{k-1})^T D_k^{-1} D_k^{-1} \mathbf{y}^{k-1}}.$$

This allows one to extend in this case the generalizations of the BB step lengths cited in Section 2.1.1.

2.4 Two-phase gradient projection methods for BQPs

Here we focus on two-phase gradient projection methods for the solution of BQP problems, i.e. problems of the form

$$\begin{array}{ll} \min & f(\mathbf{x}) := \frac{1}{2}\mathbf{x}^T H\, \mathbf{x} - \mathbf{c}^T \mathbf{x}, \\ \text{s.t.} & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}. \end{array} \tag{2.25}$$

This, apart from their pervasive presence in many application fields, is also due to the intrinsic ease in dealing with them both from the theoretical and practical point of view. Indeed, consider the first order KKT conditions for problem (2.25), i.e.

$$\mathbf{g}^* = \sum_{i=1}^{n} \lambda_i^* \mathbf{e}_i, \quad \lambda_i^* \geq 0 \text{ if } i \in \mathcal{A}_l^*, \quad \lambda_i^* \leq 0 \text{ if } i \in \mathcal{A}_u^*, \quad \lambda_i^* = 0 \text{ if } i \in \mathcal{F}^*,$$

they can be equivalently written as

$$g_i^* \geq 0 \text{ if } i \in \mathcal{A}_l^*, \quad g_i^* \leq 0 \text{ if } i \in \mathcal{A}_u^*, \quad g_i^* = 0 \text{ if } i \in \mathcal{F}^*, \tag{2.26}$$

Projections onto the feasible set of problem (2.25), i.e. the set

$$\Omega := \{\mathbf{x} \in \mathbb{R}^n \,:\, \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\},$$

are very cheap to compute. Indeed, given a point $\mathbf{x}$, the projection of $\mathbf{x}$ onto Ω can be computed by the operator P_Ω defined componentwise as

$$[P_\Omega(\mathbf{x})]_i := \begin{cases} x_i, & \text{if } \; l_i < x_i < u_i, \\ l_i, & \text{if } \; x_i \leq l_i, \\ u_i, & \text{if } \; x_i \geq u_i. \end{cases}$$

In the particular case of problem (2.25) we have that the tangent cone at a point $\mathbf{x} \in \Omega$ has the form

$$T_\Omega(\mathbf{x}) = \{\mathbf{v} \in \mathbb{R}^n \,:\, v_i \geq 0 \;\forall\, i \in \mathcal{A}_l(\mathbf{x}) \;\wedge\; v_i \leq 0 \;\forall\, i \in \mathcal{A}_u(\mathbf{x})\},$$

therefore the projected gradient of f at $\mathbf{x}$, defined in (2.14), can be written componentwise as

$$[\nabla_\Omega f(\mathbf{x})]_i := \begin{cases} -g_i, & \text{if } \; i \in \mathcal{F}, \\ \max\{-g_i, 0\}, & \text{if } \; i \in \mathcal{A}_l, \\ \min\{-g_i, 0\}, & \text{if } \; i \in \mathcal{A}_u. \end{cases} \tag{2.27}$$

Given an active-set estimate $\overline{\mathcal{A}} = \overline{\mathcal{A}}_l \cup \overline{\mathcal{A}}_u$ (and the respective estimate of the free set $\overline{\mathcal{F}}$), active-set methods for BQP are based on the solution of the subproblem

$$\begin{array}{lll} \min & f(\mathbf{x}), & \\ \text{s.t.} & x_i = l_i, & i \in \overline{\mathcal{A}}_l, \\ & x_i = u_i, & i \in \overline{\mathcal{A}}_u, \\ & l_i \leq x_i \leq u_i, & i \in \overline{\mathcal{F}}. \end{array} \tag{2.28}$$

Since each feasible point $\mathbf{v}$ for (2.28) satisfies $\mathbf{v}_{\overline{\mathcal{A}}_l} = \mathbf{l}_{\overline{\mathcal{A}}_l}$ and $\mathbf{v}_{\overline{\mathcal{A}}_u} = \mathbf{u}_{\overline{\mathcal{A}}_u}$, the problem can be solved by a minimization over the subspace of free variables. One can indeed focus on the unconstrained problem

$$\min_{\mathbf{v}_{\overline{\mathcal{F}}} \in \mathbb{R}^{|\overline{\mathcal{F}}|}} \quad \frac{1}{2}\mathbf{v}_{\overline{\mathcal{F}}}^T H_{\overline{\mathcal{F}}\,\overline{\mathcal{F}}}\, \mathbf{v}_{\overline{\mathcal{F}}} - \mathbf{c}_{\overline{\mathcal{F}}}^T \mathbf{v}_{\overline{\mathcal{F}}}, \tag{2.29}$$

and either stop as soon as one of the bound constraints

$$\mathbf{l}_{\overline{\mathcal{F}}} \leq \mathbf{v}_{\overline{\mathcal{F}}} \leq \mathbf{u}_{\overline{\mathcal{F}}}$$

is violated or find an approximate solution to the unconstrained subproblem and then project the resulting point back onto the feasible set of (2.25).

2.4.1 The GPCG method by Moré and Toraldo

Starting from the results of Calamai and Moré [27], Moré and Toraldo investigated the possibility of solving problems of the form (2.25) by means of active-set methods based on the identification properties of the gradient projection methods and their capability of adding/removing multiple variables to/from the active-set in a single iteration. In [108] they developed and analyzed an algorithm based on the alternation of two phases: the *identification phase* and the *minimization phase*. The *identification phase* consists in successive gradient projection iterations, for which the authors set an upper bound s, which eventually stopped if $\mathcal{A}(\mathbf{x}^j) = \mathcal{A}(\mathbf{x}^{j-1})$ for a certain index $j \in \{k+1, \ldots, k+s\}$. Supposing to start from a point $\mathbf{x}^k$, consider the function

$$\phi_k(\alpha) = f\left(P_\Omega(\mathbf{x}^k - \alpha \mathbf{g}^k)\right).$$

Recalling the definition of binding set at a point $\mathbf{x}^k$ (3.13), which in the case of BQPs is the set

$$\mathcal{B}^k = \mathcal{B}(\mathbf{x}^k) = \left\{ i \ : \ (i \in \mathcal{A}_l \ \wedge \ g_i^k \geq 0) \vee (i \in \mathcal{A}_u \ \wedge \ g_i^k \leq 0) \right\},$$

we can introduce the reduced gradient $\mathbf{r}^k = \mathbf{g}^k_{\mathcal{B}^k}$ and the reduced Hessian $A_k = H_{\mathcal{B}^k \mathcal{B}^k}$. It can be shown that the function $\phi_k(\alpha)$ is piecewise quadratic in α and its breakpoints, i.e. the points at which the function switches from one quadratic to the other, are related to the indices i such that $i \notin \mathcal{B}^k$, $g_i^k \neq 0$, and either l_i or u_i are finite. In particular supposing that the function has breakpoints

$$0 = \eta_0 < \eta_1 < \ldots < \eta_p < \eta_{p+1} = +\infty,$$

each breakpoint η_j $(1 \leq j \leq p)$ has the form

$$\eta_j = \begin{cases} \dfrac{x_i - u_i}{g_i}, & \text{if } g_i < 0 \text{ and } u_i < +\infty, \\ \dfrac{x_i - l_i}{g_i}, & \text{if } g_i > 0 \text{ and } l_i > -\infty, \end{cases}$$

for some $i \in \{1, \ldots, n\} \setminus \mathcal{B}^k$.

In the interval $[0, \eta_1]$, $\phi_k(\alpha)$ coincides with the function $f_k(-\alpha \mathbf{r}^k)$, where f_k is the restriction of f to the subspace in which the variables in $\mathcal{B}^k$ are fixed. Figure 2.3 illustrates the sufficient decrease condition for a function with 4 breakpoints. Note that α^k satisfies the sufficient decrease condition (2.21) if $\phi_k(\alpha^k) \leq \psi_k(\alpha^k)$ where $\phi_k(\alpha^k)$ is the piecewise linear function

$$\psi_k(\alpha) = f(\mathbf{x}^k) + \mu_1 (\mathbf{g}^k)^T \left(P_\Omega(\mathbf{x}^k - \alpha \mathbf{g}^k) - \mathbf{x}^k \right).$$

The authors relate the choice of α_0^k to the behavior of ϕ_k in $[0, \eta_1]$. In particular, if in that interval ϕ_k is strictly convex, the authors suggest to take α_0^k as the minimizer of the quadratic function representing ϕ_k in $[0, \eta_1]$. Since

$$\phi_k'(0) = -\|\mathbf{r}^k\|^2, \quad \text{and} \quad \phi_k''(0) = (\mathbf{r}^k)^T A_k \, \mathbf{r}^k,$$

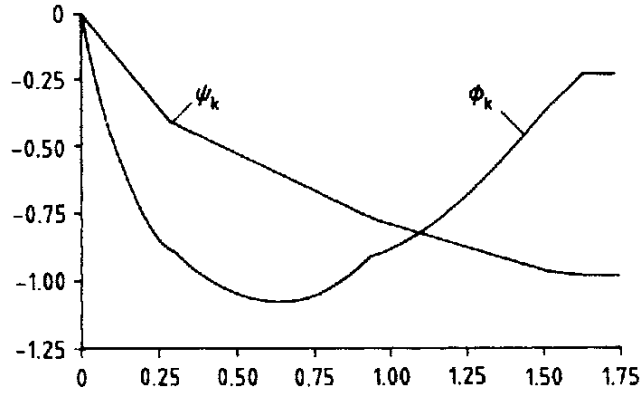

Figure 2.3. Illustration of the sufficient decrease condition for α_k [108].

the starting value for α^k is

$$\alpha_0^k = \frac{\|\mathbf{r}^k\|^2}{(\mathbf{r}^k)^T A_k\, \mathbf{r}^k}$$

whenever $(\mathbf{r}^k)^T A_k\, \mathbf{r}^k > 0$. Given the current estimate for α^k, i.e. α_j^k, the authors proposed to replace the original rule $\alpha_{j+1}^k = \beta \alpha_j^k$ with a more efficient safeguarded quadratic interpolation scheme. In detail they proposed to take

$$\alpha_{j+1}^k = \max\left\{\eta_1, \mathrm{mid}\left\{\frac{1}{100}\alpha_j^k, \overline{\alpha}_j^k, \frac{1}{2}\alpha_j^k\right\}\right\},$$

where $\overline{\alpha}_j^k$ is the minimizer of the quadratic interpolating $\phi_k(0)$, $\phi_k'(0)$ and $\phi_k(\alpha_j^k)$. In the cases in which $(\mathbf{r}^k)^T A_k\, \mathbf{r}^k \leq 0$, since the quadratic function representing ϕ_k in $[0,\, \eta_1]$ is strictly decreasing and unbounded below, the author decided to consider as starting value for α^k the largest finite breakpoint for ϕ_k, i.e. η_p.

The *minimization phase* consists in the computation of the solution $\mathbf{w}^k$ to problem

$$\min \quad f_k(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T H_{\overline{\mathcal{F}}\,\overline{\mathcal{F}}}\mathbf{w} + (\mathbf{g}_{\overline{\mathcal{F}}}^k)^T \mathbf{w}, \tag{2.30}$$

relying on the Cholesky factorization of $H_{\overline{\mathcal{F}}\,\overline{\mathcal{F}}}$. By defining the vector $\mathbf{d}^k \in \mathbb{R}^n$ such that $\mathbf{d}_{\mathcal{B}^k}^k = \mathbf{0}$ and $\mathbf{d}_{\mathcal{C}^k}^k = \mathbf{w}^k$ (with $\mathcal{C}^k = \{1, \ldots, n\} \setminus \mathcal{B}^k$), the authors proposed to compute the point $\mathbf{x}^{k+1}$ as $\mathbf{x}^{k+1} = \mathbf{x}^k + \gamma^k \mathbf{d}^k$, where

$$\gamma^k = \max\left\{\gamma \;:\; \mathbf{l} \leq \mathbf{x}^k + \gamma \mathbf{d}^k \leq \mathbf{u}\right\}.$$

It can be proved that, whenever the quadratic function f is bounded below on the set Ω, this algorithm is able to find a stationary point in a finite number of iterations.

The main weakness of the method proposed in [108] is that the exact solution of (2.30) can be "uselessly expensive" if the active set at the solution is far from being identified; on the other hand, due to the slow convergence of the gradient projection method, it is unpractical to wait until the identification of a suitable active-set. Inspired by the works by Dembo and Tulowitzki [48] and Wright [123], in their seminal 1991 paper [109], Moré and Toraldo proposed the well known GPCG (Gradient Projection Conjugate Gradient) algorithm. The idea of GPCG is to stop

the gradient projection iteration if it fails in making a reasonable progress and then proceed with the approximate solution of (2.30) by means of the conjugate gradient.

In detail, starting from $\mathbf{y}^0 = \mathbf{x}^k$, the GP identification phase is stopped if either of the two conditions

$$\mathcal{A}(\mathbf{y}^j) = \mathcal{A}(\mathbf{y}^{j+1}), \tag{2.31}$$

$$f(\mathbf{y}^j) - f(\mathbf{y}^{j+1}) \leq \eta \max_{1 \leq l < j} (f(\mathbf{y}^l) - f(\mathbf{y}^{l+1})), \tag{2.32}$$

is satisfied, with $\eta > 0$ a given constant. A criterion similar to (2.32) is used to stop the CG method in the solution of (2.30); the CG is indeed stopped if it generates a point $\mathbf{w}^j$ such that

$$f_k(\mathbf{w}^j) - f_k(\mathbf{w}^{j+1}) \leq \xi \max_{1 \leq l < j} \left\{ f_k(\mathbf{w}^l) - f_k(\mathbf{w}^{l+1}) \right\}, \tag{2.33}$$

with $\xi > 0$ a given constant. The search direction $\mathbf{d}^k$ derived from the solution of (2.30) is then used to compute the next iterate $\mathbf{x}^{k+1}$ as

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

where α^k is selected by means of the same projected line search with the safeguarded quadratic interpolation described previously for the GP phase in [108]. If the iterate $\mathbf{x}^{k+1}$ generated in this way appears to be in the face which contains the solution, the minimization phase continues. The decision, in particular, is based on the observation that if $\mathbf{x}^{k+1}$ is on the face that contains the solution, then $\mathcal{B}(\mathbf{x}^{k+1}) = \mathcal{A}(\mathbf{x}^{k+1})$, i.e. the active set is also binding. However, the condition $\mathcal{B}(\mathbf{x}^{k+1}) = \mathcal{A}(\mathbf{x}^{k+1})$ does not guarantee that $\mathbf{x}^{k+1}$ is in the face which contains the solution. If the current iterate does not lie on the face containing the solution, then the finite termination properties of the conjugate gradient method guarantee that a point $\mathbf{x}$ such that $\mathcal{B}(\mathbf{x}) \neq \mathcal{A}(\mathbf{x})$ will eventually be generated. The authors, which focused on the case of strictly convex problems, proved that in case of a non-degenarate staionary point, GPCG is able to find the solution in a finite number of steps thanks to the finite identification properties of the GP phase and the finite termination properties of the CG phase.

A critical issue about GPCG stands in the approximate minimization of (2.30). The required precision should, indeed, depend on how much that space is worth to be explored. In GPCG, instead, a purely heuristic criterion for the stopping of the minimization phase is used, based on a maximum number of iterates and a check on the bindingness of the active constraints. To overcome the numerical inefficiencies associated with using a heuristic approach for the switch between the identification phase and the minimization phase we will see in Chapter 4 how, by exploiting some ad-hoc optimality measures, the performances of the algorithm can be drastically improved. These measures are inspired to the works presented in the following sections in which the authors aimed at developing algorithms with finite termination properties in the case of a degenerate stationary point.

2.4.2 Gradient methods by Friedlander and Martínez

The active-set method for bound constrained problems proposed by Friedlander and Martínez in [74, 75] includes a procedure for deciding to leave a face of the

polyhedron which guarantees that return on the same face is not possible. Recalling the KKT conditions (2.26), the authors introduced two new vectors, which we indicate as $\boldsymbol{\varphi}(\mathbf{x})$ and $\boldsymbol{\beta}(\mathbf{x})$, defined componentwise as

$$\varphi_i(\mathbf{x}) := \begin{cases} g_i & \text{if } i \in \mathcal{F}(\mathbf{x}), \\ 0 & \text{if } i \in \mathcal{A}_l(\mathbf{x}), \\ 0 & \text{if } i \in \mathcal{A}_u(\mathbf{x}), \end{cases} \qquad \beta_i(\mathbf{x}) := \begin{cases} 0 & \text{if } i \in \mathcal{F}(\mathbf{x}), \\ \min\{0, g_i\} & \text{if } i \in \mathcal{A}_l(\mathbf{x}), \\ \max\{0, g_i\} & \text{if } i \in \mathcal{A}_u(\mathbf{x}). \end{cases} \tag{2.34}$$

It is straightforward to check that a feasible point $\mathbf{x}^*$ is stationary for problem (2.25), i.e. it satisfies the KKT conditions (2.26), if and only if

$$\boldsymbol{\beta}(\mathbf{x}^*) = \boldsymbol{\varphi}(\mathbf{x}^*) = \mathbf{0};$$

moreover, recalling the definition of projected gradient (2.27) for problem (2.25), we have that for every feasible point $\mathbf{x}$

$$\boldsymbol{\beta}(\mathbf{x}) + \boldsymbol{\varphi}(\mathbf{x}) = -\nabla_\Omega f(\mathbf{x}).$$

The idea behind the proposed method can be summarized as follows (the authors' proposal is based on the problem of maximizing a concave function, we will reformulate the content for the minimization of a convex one). Recalling the definition already given for problem (1.1), given a feasible point $\mathbf{x}^k$, we can observe that vector $\boldsymbol{\varphi}(\mathbf{x}^k)$ is the projection of the gradient onto $\Omega_0(\mathbf{x}^k)$, i.e. it can be used as a measure of optimality within the current face. Moreover, considering the set

$$B^k_{\delta,\Omega} = B_{\delta^k}(\mathbf{x}^k) \cap \Omega(\mathbf{x}^k),$$

by the first order Taylor expansion, we can estimate the difference between the value of f at $\mathbf{x}^k$ and its optimal value inside $B^k_{\delta,\Omega}$ with the quantity

$$\Delta^k_{\mathcal{F}} = \delta^k \|\boldsymbol{\varphi}(\mathbf{x}^k)\|.$$

Focusing on $\boldsymbol{\beta}(\mathbf{x}^k)$ (which the author called *chopped gradient*), it's easy to see that it is orthogonal to $\Omega_0(\mathbf{x}^k)$ and that the direction $\mathbf{d}^k = -\boldsymbol{\beta}(\mathbf{x}^k)/\|\boldsymbol{\beta}(\mathbf{x}^k)\|$ points towards the interior of the feasible set, i.e. it is possible to leave the current face (orthogonally) by a movement along $\mathbf{d}^k$. Doing this, the objective function can be decreased by an amount of at least

$$\Delta^k_{\mathcal{A}} = \widetilde{\alpha} \left\|\boldsymbol{\beta}(\mathbf{x}^k)\right\|^2 - \frac{\widetilde{\alpha}^2 L}{2} \left\|\boldsymbol{\beta}(\mathbf{x}^k)\right\|^2,$$

where $L \geq \|H\|$ is a Lipschitz constant for ∇f and

$$\widetilde{\alpha} = \min\left\{\frac{1}{L}, \frac{\gamma^k_{\mathcal{A}}}{\|\boldsymbol{\beta}(\mathbf{x}^k)\|}\right\},$$

with $\gamma^k_{\mathcal{A}} = \min\{u_i - l_i \ : \ i \in \mathcal{A}^k\}$, is the maximum feasible step length along the normal direction to the current face. The authors proposed to compare at each step the quantities $\Delta^k_{\mathcal{F}}$ and $\Delta^k_{\mathcal{A}}$:

- if $\Delta^k_{\mathcal{A}} < \Delta^k_{\mathcal{F}}$ then the current face is considered worth to be further explored; a new point $\mathbf{x}^{k+1} \in \Omega(\mathbf{x}^k)$, such that $f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k)$ is then obtained by means of one conjugate gradient step (eventually shrunk to avoid infeasibility);
- if $\Delta^k_{\mathcal{A}} \geq \Delta^k_{\mathcal{F}}$ then the face is abandoned and the next iterate can be taken as $\mathbf{x}^{k+1} = \mathbf{x}^k + \widetilde{\alpha}\mathbf{d}^k$, with the guarantee that the algorithm (being monotone) will never return inside the ball $B^k_{\delta,\Omega}(\mathbf{x}^k)$.

If $\delta^k \geq \overline{\delta} > 0$, $\forall k \in \mathbb{N}$, it can be proved that such an algorithm is able to find the solution to (2.25), in the case $H \succeq 0$, in a finite number of steps also in the case of a non-degenerate stationary point. In the follow-up paper [77], Friedlander, Martínez and Raydan developed an algorithm based on the one in [75] in which the conjugate gradient steps are replaced with a block of gradient descent steps with BB step lengths. The work was further extended in the 1997 paper by Bielschowsky et al. [14], in which the authors considered for the first time the case of non-convex problems and proposed to use different algorithm for the minimization over the current face, based on information regarding the size of the subproblem and the spectral properties of the reduced Hessian matrix. In detail for faces of small dimension, they proposed the use of Cholesky factorizations and, as the dimension increases, the use of sparse Cholesky factorization, conjugate gradient method and, finally, gradient methods with BB step lengths.

2.4.3 The MPRGP algorithm by Dostál

The works by Friedlander and Martínez inspired Dostál [53] to introduce the concept of proportional iterate. Recalling the definitions of $\boldsymbol{\varphi}(\mathbf{x})$ and $\boldsymbol{\beta}(\mathbf{x})$ given in (2.34) (which the author called respectively *free gradient* and *unbalanced contact gradient*), an iterate $\mathbf{x}^k$ is called proportional if, for a suitable constant $\Gamma > 0$,

$$\|\boldsymbol{\beta}(\mathbf{x}^k)\|_\infty \leq \Gamma \|\boldsymbol{\varphi}(\mathbf{x}^k)\|. \tag{2.35}$$

It is interesting to note that a similar check was introduced independently in the same year in [14]. Inequality (2.35) implies that the violation of the KKT conditions at the active variables does not excessively exceed the part of the gradient corresponding to the free variables. The author proved that, given a feasible point $\mathbf{x}$ and $\Gamma \geq \kappa(H)^{1/2}$, then if (2.35) holds, the minimizer of f over the face containing $\mathbf{x}$, which we will indicated as $\overline{\mathbf{x}}$, has to satisfy

$$\boldsymbol{\beta}(\overline{\mathbf{x}}) \neq \mathbf{0},$$

i.e. it is not the optimal solution of (2.25). The proposed algorithm, in the same vein as in [75], at each step checks condition (2.35). If it holds then the face is considered worth to be explored and a CG step is taken to find a new feasible point on the current face, eventually replaced by a maximum feasible step along the CG direction in case of infeasibility; otherwise, an optimal step along $-\boldsymbol{\beta}(\mathbf{x}^k)$, called *proportioning step*, is taken to leave the current face.

An enhancement for this scheme was introduced in [54] and it is known as *Modified Proportioning with Gradient Projection* (MPGP). The improvement lies in the behavior of the algorithm during the minimization on the face. If the CG step is infeasible, instead of considering a maximum feasible step, a gradient projection

step with constant step length $\overline{\alpha} \in \left(0,\ 2\|H\|^{-1}\right)$ is taken. This ensures an R-linear convergence rate in terms of bounds on the spectrum of the Hessian matrix and allows the algorithm to mantain finite termination properties in the case of a non-degenerate stationary point. The scheme was further improved in the work by Dostál and Schöberl [65] with the introduction of the *Modified Proportioning with Reduced Gradient Projection* (MPRGP) for problems with lower bound constraints only. The modification is based on the observation that, since in this case

$$\mathbf{x} - \sigma\boldsymbol{\beta} \in \Omega, \quad \forall \sigma > 0,$$

the GP step with fixed step length $\overline{\alpha}$ can be rewritten as

$$P_\Omega(\mathbf{x} - \overline{\alpha}\,\nabla f(\mathbf{x})) = \mathbf{x} - \overline{\alpha}\,(\widetilde{\boldsymbol{\varphi}}(\mathbf{x}) + \boldsymbol{\beta}(\mathbf{x}))\,,$$

where the vector $\widetilde{\boldsymbol{\varphi}}(\mathbf{x})$ is called the *reduced free gradient* and is defined componentwise as

$$\widetilde{\varphi}_i(\mathbf{x}) := \begin{cases} \min\left\{\frac{x_i - l_i}{\overline{\alpha}},\ g_i\right\} & \text{if } \ i \in \mathcal{F}(\mathbf{x}), \\ 0 & \text{if } \ i \in \mathcal{A}(\mathbf{x}). \end{cases}$$

The authors proposed to replace the GP step with the so called *expansion step* (since it is aimed at expanding the current active-set), defined as

$$\mathbf{x}^{k+1} = P_\Omega\left(\mathbf{x}^k - \overline{\alpha}\,\boldsymbol{\varphi}(\mathbf{x}^k)\right) = \mathbf{x}^k - \overline{\alpha}\,\widetilde{\boldsymbol{\varphi}}(\mathbf{x}^k),$$

and to replace (2.35) by introducing the concept of *strictly proportional iterate*, defined as an iterate satisfying

$$\left\|\boldsymbol{\beta}(\mathbf{x}^k)\right\|^2 \leq \Gamma^2\,\widetilde{\boldsymbol{\varphi}}(\mathbf{x}^k)^T\boldsymbol{\varphi}(\mathbf{x}^k), \tag{2.36}$$

with $\Gamma > 0$ a given constant. The scheme obtained is proved to have the R-linear rate of convergence of the MPGP and recovers the finite termination property also in the case of degenerate stationary points. The actual implementation of MPRGP, which is outlined in Algorithm 5.2, includes a maximum feasible step along the CG direction before the expansion step.

The work by Dostál and Schöberl was further extended by Mohy-ud-Din and Robinson [107] to the case of general bound constrained problems with possibly non-convex objective functions. Starting from the definition of $\boldsymbol{\varphi}$ and $\boldsymbol{\beta}$ in (2.34), given a scalar $\alpha > 0$, the authors defined the *reduced free gradient* $\widetilde{\boldsymbol{\varphi}}_\alpha(\mathbf{x})$ and the *reduced chopped gradient* $\widetilde{\boldsymbol{\beta}}_\alpha(\mathbf{x})$ componentwise as

$$\begin{aligned} \left[\widetilde{\varphi}_\alpha\right]_i(\mathbf{x}) &:= \begin{cases} \min\left\{\frac{x_i - l_i}{\alpha},\ \varphi_i(\mathbf{x})\right\} & \text{if } \varphi_i(\mathbf{x}) \geq 0, \\ \max\left\{\frac{x_i - u_i}{\alpha},\ \varphi_i(\mathbf{x})\right\} & \text{if } \varphi_i(\mathbf{x}) < 0, \end{cases} \\ \left[\widetilde{\beta}_\alpha\right]_i(\mathbf{x}) &:= \begin{cases} \min\left\{\frac{x_i - l_i}{\alpha},\ \beta_i(\mathbf{x})\right\} & \text{if } \beta_i(\mathbf{x}) \geq 0, \\ \max\left\{\frac{x_i - u_i}{\alpha},\ \beta_i(\mathbf{x})\right\} & \text{if } \beta_i(\mathbf{x}) < 0. \end{cases} \end{aligned} \tag{2.37}$$

The two vectors have similar roles to the original free and chopped gradient, indeed it is straightforward to show that, for each $\alpha > 0$,

$$P_\Omega\left(\mathbf{x} - \alpha\,\nabla f(\mathbf{x})\right) = \mathbf{x} - \alpha\left(\widetilde{\boldsymbol{\varphi}}_\alpha(\mathbf{x}) + \widetilde{\boldsymbol{\beta}}_\alpha(\mathbf{x})\right),$$

and the vector

$$\boldsymbol{\nu}(\mathbf{x}) = \mathbf{x} - P_\Omega\left(\mathbf{x} - \nabla f(\mathbf{x})\right) = \widetilde{\boldsymbol{\varphi}}_1(\mathbf{x}) + \widetilde{\boldsymbol{\beta}}_1(\mathbf{x})$$

is an appropriate measure of optimality for problem (2.25) [34]. One advantage in using $\widetilde{\boldsymbol{\varphi}}_\alpha(\mathbf{x})$, $\widetilde{\boldsymbol{\beta}}_\alpha(\mathbf{x})$ and $\boldsymbol{\nu}(\mathbf{x})$ in place of the original quantities is that they are continuous w.r.t. $\mathbf{x}$ whereas the projected gradient $\nabla_\Omega f(\mathbf{x})$ is only lower semicontinuous.

The definition of the reduced components of the gradient led to the definition of a new switching criterion to replace (2.35) and (2.36); the authors proposed to base the switch on the condition

$$\widetilde{\boldsymbol{\beta}}_{\overline{\alpha}}(\mathbf{x}^k)^T\boldsymbol{\beta}(\mathbf{x}^k) \leq \Gamma^2\,\widetilde{\boldsymbol{\varphi}}_{\overline{\alpha}}(\mathbf{x}^k)^T\boldsymbol{\varphi}(\mathbf{x}^k), \tag{2.38}$$

with $\overline{\alpha} \in \left(0,\ 2\|H\|^{-1}\right)$. The algorithm proposed in [107] is very similar to MPRGP, with the addition of suitable checks and steps to deal with the non-convexity of the objective function. Moreover, the author proposed to introduce further checks to guarantee that the algorithm stops at a stationary point satisfying some second-order optimality conditions. Under reasonable assumption, the convergence analysis shows that the proposed algorithm either terminates in a finite number of iterations to a second-order stationary point, or it generates a sequence of iterates along which the objective function converges to negative infinity.

It is worth mentioning that similar ideas, to the ones outlined in these sections, have been also used in [89] for general nonlinear problems subject to bound constraints. The authors proposed to alternate a gradient projection method, based on the Cyclic-Barzilay-Borwein [39] step length and on the GLL [84] non-monotone line search, and an unconstrained subspace minimization step based on the CG_DESCENT algorithm proposed in [86]. The switch between the two phases is based on a comparison between the optimality w.r.t. the full problem and the optimality w.r.t. the subspace defined by the active constraints. A theoretical extension of this framework to the case of nonlinear problems subject to polyhedral constraints has been proposed in [90].

2.5 Projection onto polyhedra

In the previous sections we introduced the gradient projection methods observing that for the class of BQP problems the projection operator can be computed very easily in $\mathcal{O}(n)$ operations. Nevertheless, the nice convergence properties of the method, analyzed in [27], hold for general optimization problems subject to linear constraints. The computational bottleneck in their application is the cost of the projection onto the polyhedron described by the constraint of the problem, which in the case of a general linearly constrained problem can be very expensive and cost as much as solving the whole problem.

2.5.1 The case of a single linear equality and bound constraints

One particular class of polyhedra which received much attention is the class of polyhedra of the form

$$\Omega = \left\{\mathbf{x} \in \mathbb{R}^n \ :\ \mathbf{a}^T\mathbf{x} = b \ \wedge\ \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\right\}, \tag{2.39}$$

where $\mathbf{a} \in \mathbb{R}^n$, $\mathbf{l} \in \{\mathbb{R} \cup \{-\infty\}\}^n$, $\mathbf{u} \in \{\mathbb{R} \cup \{+\infty\}\}^n$, and $\mathbf{l} \leq \mathbf{u}$. In the case of $\mathbf{a} = \mathbf{1}$ this set is known in literature as the *double-sided simplex*; with a little abuse of notations we will refer to the general case in the same way. A lot of work has been done on the analysis and development of efficient algorithms for the computation of the projection onto Ω, especially in the case in which it corresponds to the probability simplex, i.e.

$$\Omega = \left\{ \mathbf{x} \in \mathbb{R}^n \ : \ \sum_i x_i = b \ \wedge \ \mathbf{x} \geq \mathbf{0} \right\}.$$

This is also due to the fact that the projection problem can be seen as an instance of a more general class of QP problems of the form

$$\begin{aligned} \min \quad & \frac{1}{2}\mathbf{x}^T D\, \mathbf{x} - \mathbf{c}^T\mathbf{x}, \\ \text{s.t.} \quad & \mathbf{a}^T\mathbf{x} = b, \\ & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \end{aligned} \tag{2.40}$$

where D is a diagonal positive definite matrix with $\text{diag}(D) = [d_1, d_2, \ldots, d_n]$, which are usually referred to in literature as *quadratic knapsack problems* and arise in many application areas, such as resource allocation, hierarchical production planning, transportation problems, multicommodity network flows, etc. (see e.g., [96, 97, 98] and references therein). It is easy to check that the problem of projecting a point $\mathbf{z}$ onto Ω corresponds to the case in which $D = I_n$ and $\mathbf{c} = \mathbf{z}$.

Recall that a stationary point $\mathbf{x}^*$ for problem (2.40) has to satisfy the first-order KKT conditions

$$\begin{aligned} & D\mathbf{x} - \mathbf{c} = \lambda\mathbf{a}, \\ & \mathbf{a}^T\mathbf{x} = b, \\ & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \end{aligned}$$

where $\lambda \in \mathbb{R}$ is the Lagrange multiplier associated with the single linear equality constraint.

Here we will briefly describe solution strategies for problem (2.40) showing how it is possible to develop algorithms with the optimal computational cost $\mathcal{O}(n)$ [24, 28, 113]. The main idea behind the algorithms developed for the solution of this problem is to exploit a dual reformulation which, as we will see, allows one to turn the optimization problem into the solution of a piecewise linear equation. We will see, however, that from a practical point of view, algorithm with a larger theoretical cost can perform better on large-scale problems, such as the one based on variable fixing techniques [98], which has a theoretical $\mathcal{O}(n^2)$ complexity, the secant method proposed by Dai and Fletcher [38], and the semismooth Newton method proposed by Cominetti et al. in [31].

2.5.1.1 Dual reformulation

Observe that the quadratic knapsack problem, as any convex quadratic problem, agrees to strong duality. Thus we can recover the solution to the original problem by solving a dual problem. Following [27], since the box constraints are easy to handle,

we can consider the Lagrangian function associated with the linear constraint and introduce the following dual problem for (2.40)

$$\max_{\lambda\in\mathbb{R}} \inf_{\mathbf{l}\le\mathbf{x}\le\mathbf{u}} \left\{\frac{1}{2}\mathbf{x}^T D\mathbf{x} - \mathbf{c}^T\mathbf{x} + \lambda(b - \mathbf{a}^T\mathbf{x})\right\}. \tag{2.41}$$

Given a fixed λ, the solution of the inner infimum problem can be written as

$$\mathbf{x}(\lambda) = \operatorname{mid}\left\{\mathbf{l}, D^{-1}(\lambda\mathbf{a} + \mathbf{c}), \mathbf{u}\right\}, \tag{2.42}$$

or equivalently componentwise as

$$x_i(\lambda) = \max\left\{l_i,\, \min\left\{\frac{\lambda\, a_i + c_i}{d_i},\, u_i\right\}\right\}, \quad i = 1, \ldots, n.$$

Thus the KKT conditions for (2.40) can be rewritten as

$$\phi(\lambda) := \mathbf{a}^T\mathbf{x}(\lambda) - b = 0, \tag{2.43}$$

which is equivalent to primal-dual optimality, so that (2.40) reduces to the finding of the unique solution λ^* to the equation

$$\phi(\lambda) := \sum_{i=1}^{n} a_i \operatorname{mid}\left\{l_i, \frac{\lambda\, a_i + c_i}{d_i}, u_i\right\} = \sum_{i=1}^{n} \phi_i(\lambda) = b. \tag{2.44}$$

Each component $\phi_i : \mathbb{R} \to \mathbb{R}$ of the function ϕ is a piece-wise nondecreasing linear function with two breakpoints (left side of Figure 2.4), namely $\eta_\ell^i = \left\{\frac{d_i\, l_i - c_i}{a_i}\right\}$ and $\eta_u^i = \left\{\frac{d_i\, u_i - c_i}{a_i}\right\}$, where we define as breakpoint each point of discontinuity of the first derivative of a given function. Indeed, for λ in $(-\infty,\, \eta_\ell^i)$, ϕ_i assumes the constant value $a_i\, l_i$; in $[\eta_\ell^i,\, \eta_u^i]$ it is a monotone increasing linear function with slope $\frac{a_i^2}{d_i}$; finally, in $(\eta_u^i,\, +\infty)$ it assumes the constant value $a_i\, u_i$.

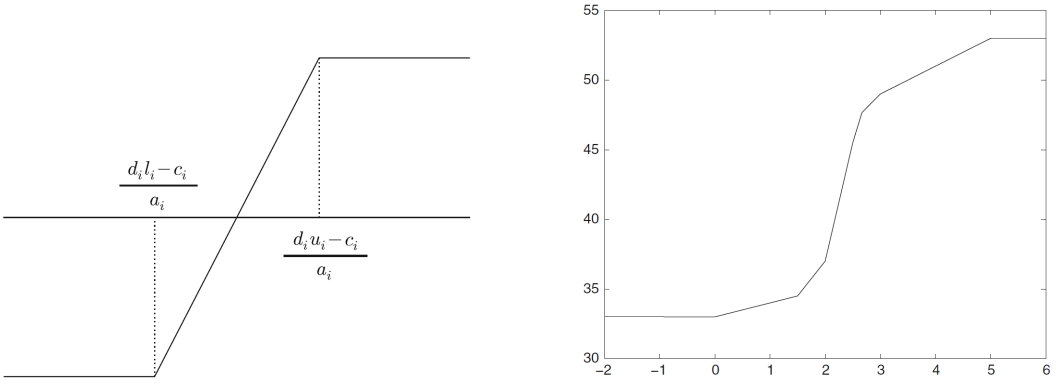

Figure 2.4. The i-th component of $\phi(\lambda)$ and its two breakpoints (*left*) and an example of $\phi(\lambda)$ with six breakpoints (*right*) [31].

The resulting function $\phi(\lambda)$ is a nondecreasing piece-wise linear function, whose set of breakpoints

$$\mathcal{E} = \bigcup_{i=1}^{n} \left\{\eta_\ell^i, \eta_u^i\right\}$$

contains at most $2n$ distinct elements, and which assumes constant values in $(-\infty, \eta_{min})$ and in $(\eta_{max}, +\infty)$, with $\eta_{min} = \min\{\mathcal{E}\}$ and $\eta_{max} = \max\{\mathcal{E}\}$. An example of such a function with 6 breakpoints is given in the right side of Figure 2.4.

It follows that the problem (2.44) has a solution if and only if

$$\inf\{\phi(\lambda) \ : \ \lambda \in \mathbb{R}\} \equiv \phi_{min} \leq b \leq \phi_{max} \equiv \sup\{\phi(\lambda) \ : \ \lambda \in \mathbb{R}^n\}$$

where $\phi_{min} = \phi(\eta_{min})$ and $\phi_{max} = \phi(\eta_{max})$.

The two following theorems, regarding the relation between a solution $\mathbf{x}^*$ to (2.40) and a solution λ^* to (2.44), have been proved in [28].

Theorem 2.5.1. *If $\mathbf{x}^*$ solves problem (2.40) then $x_i^* = x_i(\lambda^*)$ where*

$$x_i(\lambda) = \max\left\{l_i, \min\left\{\frac{\lambda\, a_i + c_i}{d_i}, u_i\right\}\right\}, \quad i = 1, \ldots, n, \tag{2.45}$$

and λ^* solves problem (2.44).

Theorem 2.5.2. *If λ^* solves the problem*

$$\min\{|\phi(\lambda) - b| \ : \ \lambda \in \mathbb{R}\},$$

then $\mathbf{x}(\lambda^*)$ solves (2.40) if $\phi(\lambda^*) = b$. Otherwise, $x(\lambda^*)$ solves the problem

$$\min\{|\mathbf{a}^T\mathbf{x} - b| \ : \ \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}.$$

2.5.1.2 Optimal $\mathcal{O}(n)$ algorithms

The structure of $\phi(\lambda)$ suggest to find the solution by looking at the set of breakpoints. Indeed, supposing that one has found the breakpoints

$$\eta_\ell^* = \max\{\eta \in \mathcal{E} \ : \ \phi(\eta) \leq b\} \quad \text{and} \quad \eta_u^* = \min\{\eta \in \mathcal{E} \ : \ \phi(\eta) \geq b\},$$

since no other breakpoint lies between them, the solution λ^* to (2.44) can be found by linear interpolation in the interval $[\eta_\ell^*, \eta_u^*]$ as the only value such that $\phi(\lambda^*) = b$.

The first proposed algorithm [93, 94] for the solution of (2.44) are based on a pre-ordering of the set $\mathcal{E}$ and then on bisection in order to find η_ℓ^* and η_u^*. This class of algorithms has a computational cost of order $\mathcal{O}(n \log n)$, i.e. the cost of the ordering dominates the cost of all the other operations.

The idea behind the optimal algorithms of order $\mathcal{O}(n)$ (e.g., [24, 28, 113, 97, 96]) is basically that to skip the expensive pre-ordering and performing the bisection on $\mathcal{E}$ by progressively splitting it computing medians of its subsets; this results in cheaper algorithm since the cost of the computation of the median of a set $\mathcal{S}$ is indeed $\mathcal{O}(|\mathcal{S}|)$. The bisection process either terminates if, for some λ_m, $\phi(\lambda_m) = b$ or returns two consecutive breakpoints λ_p and λ_m such that $\lambda^* \in [\lambda_p, \lambda_m]$. The solution can be computed by linear interpolation starting from λ_p and λ_m as

$$\lambda^* = \lambda_m - (\phi(\lambda_m) - b)\,\frac{\lambda_m - \lambda_p}{\phi(\lambda_m) - \phi(\lambda_p)}.$$

2.5.1.3 Variable fixing algorithms

Even if $\mathcal{O}(n)$ is the optimal complexity for the solution of quadratic knapsack problems, from the practical point of view optimal algorithms can be outperformed by algorithm with an higher computational cost. A first example is given by variable fixing algorithms derived from the work of Luss and Gupta [104] and further analyzed by Michelot [106] and Shor [121] (in the case of the simplex) and by Kiwiel [98]. This class of algorithms has a worst-case performance of $\mathcal{O}(n^2)$, however they have been showed to be competitive in practice with $\mathcal{O}(n)$ algorithms. The main idea behind this class of algorithms is to solve the quadratic knapsack problem by directly tackling its primal formulation (2.40). At each step the set $\{1, \dots, n\}$ is partitioned in three sets, i.e. L^k, U^k, and F^k, representing respectively the variables assumed to be on the lower bound, the variables assumed to be on the upper bound and the variables assumed to be free. Starting from a given partition, the algorithm finds the solution $\widetilde{\mathbf{x}}^k$ to the equality constrained subproblem

$$\begin{aligned} \min \quad & \frac{1}{2}\mathbf{x}^T D\,\mathbf{x} - \mathbf{c}^T\mathbf{x}, \\ \text{s.t.} \quad & \mathbf{a}^T\mathbf{x} = b, \\ & x_i = l_i, \quad i \in L^k, \\ & x_i = u_i, \quad i \in U^k, \end{aligned}$$

which can be obtained in closed form, and updates the partition of the indices according to some measures of feasibility for $\widetilde{\mathbf{x}}^k$. In what follows we will assume that $\mathbf{a} > \mathbf{0}$ without loss of generality, since all the other cases can be reduced to this by a change of variables. In [97], Kiwiel proposed to compute the quantities

$$\nabla_k = \sum_{i \in F_l^k} a_i \left(l_i - \widetilde{x}_i^k\right), \quad \text{where } F_l^k = \left\{ i \in F^k \ : \ \widetilde{x}_i^k \leq l_i \right\},$$

$$\Delta_k = \sum_{i \in F_u^k} a_i \left(\widetilde{x}_i^k - u_i\right), \quad \text{where } F_u^k = \left\{ i \in F^k \ : \ \widetilde{x}_i^k \geq u_i \right\}.$$

If $\nabla_k = \Delta_k$, it can be shown that the point $\mathbf{x}^k$ obtained by setting $x_i^k = l_i$ for $i \in L^k \cup F_l^k$, $x_i^k = u_i$ for $i \in U^k \cup F_u^k$ and $x_i^k = \widetilde{x}_i^k$ otherwise, is not only feasible, but also optimal for (2.40). If the previous relation does not hold different update strategies are used for the sets L^k, U^k, and F^k: if $\nabla_k > \Delta_k$ then

$$L^{k+1} = L^k \cup F_l^k, \quad F^{k+1} = F^k \setminus F_l^k, \ \text{ and } \ U^{k+1} = U^k,$$

otherwise, if $\nabla_k < \Delta_k$,

$$L^{k+1} = L^k, \quad F^{k+1} = F^k \setminus F_u^k, \ \text{ and } \ U^{k+1} = U^k \cup F_u^k.$$

2.5.1.4 The Dai-Fletcher secant-based algorithm

An algorithm which have proved to be very efficient in the solution of problems of the form (2.40) is the secant-based method introduced by Dai and Fletcher in [41]. The algorithm proposed for the solution of (2.44) consists into two phases: a *bracketing phase* in which an interval $[\lambda_l, \lambda_u]$ containing the solution to (2.44) is

Algorithm 2.1 Bracketing Phase of the Dai-Fletcher algorithm

Let $\lambda \in \mathbb{R}$, $\Delta\lambda > 0$;
Compute $\mathbf{x} = \mathbf{x}(\lambda)$ by (2.42); $\psi = \mathbf{a}^T\mathbf{x} - b$;
if $\psi < 0$ **then**
    $\lambda_l = \lambda$; $\psi_l = \psi$; $\lambda = \lambda + \Delta\lambda$;
    Compute $\mathbf{x}$ by (2.42); $\psi = \mathbf{a}^T\mathbf{x} - b$;
    while $\psi < 0$ **do**
        $\lambda_l = \lambda$; $\psi_l = \psi$; $s = \max\left\{\frac{\psi_l}{\psi} - 1,\, 0.1\right\}$;
        $\Delta\lambda = \Delta\lambda + \frac{\Delta\lambda}{s}$; $\lambda = \lambda + \Delta\lambda$;
        Compute $\mathbf{x}$ by (2.42); $\psi = \mathbf{a}^T\mathbf{x} - b$;
    end while
    $\lambda_u = \lambda$; $\psi_u = \psi$;
else
    $\lambda_u = \lambda$; $\psi_u = \psi$; $\lambda = \lambda - \Delta\lambda$;
    Compute $\mathbf{x}$ by (2.42); $\psi = \mathbf{a}^T\mathbf{x} - b$;
    while $\psi > 0$ **do**
        $\lambda_u = \lambda$; $\psi_u = \psi$; $s = \max\left\{\frac{\psi_u}{\psi} - 1, 0.1\right\}$;
        $\Delta\lambda = \Delta\lambda + \frac{\Delta\lambda}{s}$; $\lambda = \lambda - \Delta\lambda$;
        Compute $\mathbf{x}$ by (2.42); $\psi = \mathbf{a}^T\mathbf{x} - b$;
    end while
    $\lambda_l = \lambda$; $\psi_l = \psi$;
end if

found, and a *secant phase* in which this interval is progressively reduced until the algorithm finds the two consecutive breakpoints between which the optimal value of λ lies. Since the algorithm will be used in the MATLAB implementation of the P2GP method introduced in Chapter 4, we provide here (see Algorithm 2.1 and Algorithm 2.2) the pseudocodes for the two phases of the algorithm.

In the bracketing phase, outlined in Algorithm 2.1, the user is asked to provide an initial estimate for λ and the length $\Delta\lambda$ of the interval with extreme λ containing the solution. To ease the description of the algorithm we have defined the function $\psi(\lambda) = \phi(\lambda) - b$; clearly solving (2.44) is equivalent to finding a root for $\psi(\lambda)$. If $\psi(\lambda) < 0$ (respectively $\psi(\lambda) > 0$) the search for the interval takes place in the positive (respectively the negative) λ direction. Considering w.l.o.g. the case $\psi(\lambda) < 0$, the algorithm at each step starts with λ_l equal to the current estimate of λ, it checks the value of $\psi = \psi(\lambda + \Delta\lambda)$, if it is larger than 0, then it sets $\lambda_u = \lambda + \Delta\lambda$ and terminates, otherwise it updates the estimate for $\Delta\lambda$ depending on $\psi_l = \psi(\lambda_l)$ and ψ and then sets $\lambda = \lambda + \Delta\lambda$. If the problem is feasible, then the bracketing phase is guaranteed to terminate with a bracket $[\lambda_l, \lambda_u]$ containing a solution of the equation $\psi(\lambda) = 0$.

The algorithm terminates with the bracketing phase if for some λ the value of $\psi(\lambda)$ is sufficiently close to zero, otherwise it proceeds with the secant phase, reported in Algorithm 2.2.

Algorithm 2.2 Secant Phase of the Dai-Fletcher algorithm

Let $s = 1 - \dfrac{\psi_l}{\psi_u}$; $\Delta\lambda = \dfrac{\Delta\lambda}{s}$; $\lambda = \lambda_u - \Delta\lambda$;
Compute $\mathbf{x}$ by (2.42); $\psi = \mathbf{a}^T\mathbf{x} - b$;;
while not converged **do**
  if $\psi > 0$ **then**
    if $s \leq 2$ **then**
      $\lambda_u = \lambda$; $\psi_u = \psi$; $s = 1 - \dfrac{\psi_l}{\psi_u}$;
      $\Delta\lambda = \dfrac{\lambda_u - \lambda_l}{s}$; $\lambda = \lambda_u - \Delta\lambda$;
    else
      $s = \max\left\{\dfrac{\psi_u}{\psi} - 1,\, 0.1\right\}$; $\Delta\lambda = \dfrac{\lambda_u - \lambda}{s}$;
      $\lambda_{new} = \max\{\lambda - \Delta\lambda,\, 0.75\lambda_l + 0.25\lambda\}$;
      $\lambda_u = \lambda$; $\psi_u = \psi$; $\lambda = \lambda_{new}$; $s = \dfrac{\lambda_u - \lambda_l}{\lambda_u - \lambda}$;
    end if
  else
    if $s \geq 2$ **then**
      $\lambda_l = \lambda$; $\psi_l = \psi$; $s = 1 - \dfrac{\psi_l}{\psi_u}$;
      $\Delta\lambda = \dfrac{\lambda_u - \lambda_l}{s}$; $\lambda = \lambda_u - \Delta\lambda$;
    else
      $s = \max\left\{\dfrac{\psi_l}{\psi} - 1,\, 0.1\right\}$; $\Delta\lambda = \dfrac{\lambda - \lambda_l}{s}$;
      $\lambda_{new} = \min\{\lambda + \Delta\lambda,\, 0.75\lambda_u + 0.25\lambda\}$;
      $\lambda_l = \lambda$; $\psi_l = \psi$; $\lambda = \lambda_{new}$; $s = \dfrac{\lambda_u - \lambda_l}{\lambda_u - \lambda}$;
    end if
  end if
  Compute $\mathbf{x}$ by (2.42); $\psi = \mathbf{a}^T\mathbf{x} - b$;
end while

At each step of the secant phase the algorithm starts with an interval $[\lambda_l, \lambda_u]$, with $\psi(\lambda_l) < 0$ and $\psi(\lambda_u) > 0$, and with an estimate of the solution λ. If, w.l.o.g., $\psi(\lambda) > 0$ then the algorithm checks whether λ lies in the left half of the interval $[\lambda_l, \lambda_u]$ or in the right one. In the former case the new estimate for λ is computed by a secant step based on λ_l and λ; in the latter the algorithm compares a secant step based on λ and λ_u and a step to the point $\frac{3}{4}\lambda_l + \frac{1}{4}\lambda$, choosing whichever is smaller to generate the new estimate of λ. This choice ensures that the interval length is reduced at each step at least by a factor of 25%. In both cases the new bracket is originated by fixing λ_l and taking λ_u equal to the previous estimate of λ. The secant phase terminates if preset tolerances on either $\psi(\lambda)$ or $\Delta\lambda$ are met.

2.5.1.5 The Newton's method by Cominetti et al.

Starting from the secant-based algorithm of Dai and Fletcher, Cominetti et al. proposed in [31] a new algorithm for the solution of the dual problem (2.44) in which they replaced the secant method with Newton's method, which doesn't need a bracketing phase.

To ease the comprehension of the functioning of the algorithm we will follow the example of the authors and consider first the special case of problem (2.40) in which each variable only bounded from below, i.e. $u_i = +\infty$ for all i. In this case (2.42) reduces to

$$\mathbf{x}(\lambda) = \max(\mathbf{l}, D^{-1}(\mathbf{a}\lambda + \mathbf{c})) \tag{2.46}$$

and the functions ϕ_i have the form

$$\phi_i(\lambda) = \begin{cases} \max\left\{a_i l_i,\ \frac{a_i^2\lambda + a_i c_i}{d_i}\right\}, & \text{if } a_i > 0, \\ \min\left\{a_i l_i,\ \frac{a_i^2\lambda + a_i c_i}{d_i}\right\}, & \text{if } a_i < 0. \end{cases} \tag{2.47}$$

In the first case ($a_i > 0$) the breakpoint is said to be *positive*, because it increases the derivative of ϕ, while in the other case the breakpoint is said to be a *negative*. To compute the derivative $\phi'(\lambda)$ of the objective function in (2.44), it is sufficient to sum up the slopes $\frac{a_i^2}{d_i}$ corresponding to positive breakpoints to the left of λ and the slopes of negative breakpoints to the right of λ. In the case in which λ is itself a breakpoint, the function is clearly non differentiable, however the right and left derivatives are still well defined and can be computed by the formulas

$$\phi'_+(\lambda) = \sum_{\substack{a_i > 0 \\ \eta_\ell^i \le \lambda}} \frac{a_i^2}{d_i} + \sum_{\substack{a_i < 0 \\ \eta_\ell^i > \lambda}} \frac{a_i^2}{d_i}, \tag{2.48}$$

$$\phi'_-(\lambda) = \sum_{\substack{a_i < 0 \\ \eta_\ell^i > \lambda}} \frac{a_i^2}{d_i} + \sum_{\substack{a_i > 0 \\ \eta_\ell^i \le \lambda}} \frac{a_i^2}{d_i}. \tag{2.49}$$

The method for the solution of the problem with single bounds (see [31, Algorithm 1]) uses Newton's steps based either on $\phi'_+(\lambda)$ (in the case $\phi(\lambda) < b$ and $\phi'_+(\lambda) > 0$) or on $\phi'_-(\lambda)$ (in the case $\phi(\lambda) > b$ and $\phi'_-(\lambda) > 0$). The authors proved that it converges globally without the need of a globalization strategy; moreover, since ϕ has at most n breakpoints, the Newton's iterates may generate at most $n+1$ distinct points. The authors proved that Newton's algorithm for single bounds stops in at most $n+1$ iterations with an overall complexity of $\mathcal{O}(n^2)$ arithmetic operations.

Consider now the general case of problem (2.40), and assume again, w.l.o.g., that $\mathbf{a} > \mathbf{0}$. In this case, as observed in Section 2.5.1.1, each term $\phi_i(\lambda)$ is constant up to its first breakpoint η_ℓ^i, where it becomes an increasing linear function with slope a_i^2/d_i up to the next breakpoint η_u^i where it becomes constant again. As already observed, the function ϕ in this case is a non-decrasing piecewise linear function, and Newton's method may cycle as shown in Figure 2.5, corresponding to the case

$$n = 3,\ D = I,\ \mathbf{a} = \left(\sqrt{2},\ 1,\ 1\right)^T,\ \mathbf{l} = \left(-\frac{1}{\sqrt{2}},\ 0,\ -\infty\right)^T,\ \text{and } \mathbf{u} = \left(\frac{1}{\sqrt{2}},\ +\infty,\ 0\right)^T.$$

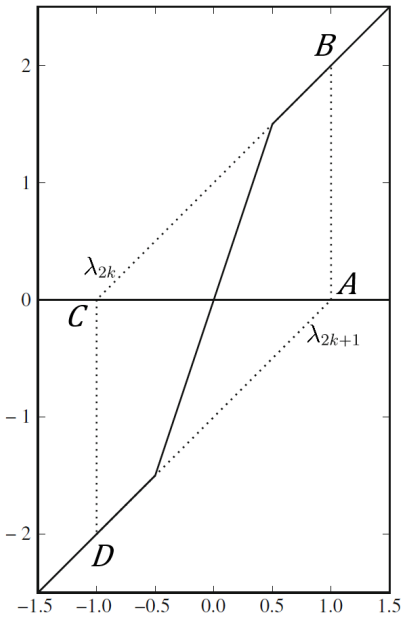

Figure 2.5. An example in which Newton's method cycles [31].

Starting from the point $A \equiv (\lambda_{2k+1}, 0)$, to which corresponds the point $B \equiv (\lambda_{2k+1}, \phi(\lambda_{2k+1}))$ on the function graph, the Newton's step returns the point $C \equiv (\lambda_{2k}, 0)$, to which corresponds the point $D \equiv (\lambda_{2k}, \phi(\lambda_{2k}))$ on the function graph; the Newton's step in C returns A, thus closing the cycle.

To ensure the convergence of the algorithm a globalization strategy is needed. The authors proposed to keep track at each step of α_k, defined as the largest iterate such that $\phi(\alpha_k) < b$ computed by the algorithm up to the k-th iteration, and of β_k, defined as the smallest iterate such that $\phi(\beta_k) > b$ computed up to the k-th iteration. If the next Newton's iterate falls outside the interval (α_k, β_k), a cycle can occur and a secant step is performed. The proposed algorithm can be considered as a variation of the secant method of Dai and Fletcher [38], in which the initial bracketing phase is replaced by Newton's iterations and Newton's method is used in place of the secant method whenever possible. At the end of each step of the algorithm a variable fixing strategy, inspired to the one described in Section 2.5.1.3, is used for helping to reduce the problem size as the method progresses. The authors proved that the algorithm performs at most $2n + 1$ Newton's steps and at most $2n$ secant steps, resulting in at most $4n + 1$ iterations with an overall complexity of $\mathcal{O}(n^2)$ arithmetic operations.

In [31] an interesting comparison between the algorithm described in this section was reported. The results show how in practice, on large-scale problems, the Newton's method converges is faster than the algorithms based on median finding, variable fixing, and secant techniques, even if it has an higher theoretical complexity.

2.5.2 The case of sparse linear constraints

As already mentioned, computing the projection of a point onto the feasible set of a general problem of the form (1.1) can be very expensive, for this reason the use

gradient projection methods has been mainly restricted to the cases of problems subject to bound constraints or bound constraint and a single linear constraint. However, recently, an efficient algorithm, called PPROJ, has been proposed by Hager and Zhang in [91] for the projection onto polyhedra defined by sparse linear constraints.

The authors focused their analysis on problems of the form

$$\begin{array}{ll} \min & \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|^2, \\ \text{s.t.} & C\,\mathbf{x}=\mathbf{b}, \\ & \mathbf{l}\le\mathbf{b}\le\mathbf{u}, \\ & \mathbf{x}\ge\mathbf{0}, \end{array} \tag{2.50}$$

where $\mathbf{y}\in\mathbb{R}^n$ is the point to project, $C=(\mathbf{c}_1,\dots,\mathbf{c}_n)\in\mathbb{R}^{m\times n}$ has nonzero columns, and $\mathbf{l},\mathbf{u}\in\mathbb{R}^m$, with $l_i\le u_i$ for all i.

Starting from the Lagrangian function associated with the linear equality constraints in (2.50), which has the form

$$\mathcal{L}(\mathbf{x},\mathbf{b},\boldsymbol{\lambda})=\frac{1}{2}\|\mathbf{y}-\mathbf{x}\|^2-(A\,\mathbf{x}-\mathbf{b})^T\boldsymbol{\lambda},$$

a solution $\mathbf{x}^*$ to (2.50) can be found by solving the dual problem

$$\max_{\boldsymbol{\lambda}\in\mathbb{R}^m} \quad L(\boldsymbol{\lambda}), \tag{2.51}$$

where the dual function $L(\boldsymbol{\lambda})$ is defined as

$$L(\boldsymbol{\lambda})=\min_{\mathbf{x}\in\mathbb{R}^n,\,\mathbf{b}\in\mathbb{R}^m}\left\{\mathcal{L}(\mathbf{x},\mathbf{b},\boldsymbol{\lambda})\ :\ \mathbf{l}\le\mathbf{b}\le\mathbf{u},\quad \mathbf{x}\ge\mathbf{0}\right\}. \tag{2.52}$$

By observing that the values of $\mathbf{x}(\boldsymbol{\lambda})$ and $\mathbf{b}(\boldsymbol{\lambda})$ for which the minimum in (2.52) is obtained are given respectively by

$$x_i(\boldsymbol{\lambda})=\max\{y_i+\mathbf{c}_i^T\boldsymbol{\lambda},\,0\},\qquad i=1,\dots,n,$$

and

$$b_j(\boldsymbol{\lambda})=\begin{cases} l_j, & \text{if } \lambda_j>0,\\ [l_j,\,u_j], & \text{if } \lambda_j=0,\\ u_j, & \text{if } \lambda_j<0,\end{cases}\qquad j=1,\dots,m,$$

it can be shown that the dual function $L(\boldsymbol{\lambda})$ is the sum of a differentiable piecewise quadratic function and a piecewise linear function (hence, it is overall piecewise quadratic).

The authors proposed to solve (2.51) by a dual active set strategy (DASS) which combines the SpaRSA algorithm by Wright, Nowak and Figueiredo [124] and the Dual Active Set Strategy (DASA) [87, 43]. SpaRSA, which has been show to have a Q-linear convergence rate to a solution $\boldsymbol{\lambda}^*$ of (2.51), is used to approximately identify the set

$$\mathcal{Z}(\boldsymbol{\lambda}^*)=\{i\ :\ x_i(\boldsymbol{\lambda}^*)=0\}$$

of the primal variables x_i which are zero at the solution and the sets

$$\mathcal{E}_+(\boldsymbol{\lambda}^*)=\{j\ :\ b_i(\boldsymbol{\lambda}^*)=l_i\}\quad\text{and}\quad\mathcal{E}_-(\boldsymbol{\lambda}^*)=\{j\ :\ b_i(\boldsymbol{\lambda}^*)=u_i\},$$

corresponding to the sets of inequalities that are treated as at their lower and upper bounds, respectively. By means of a suited switching criterion, the computation passes from SpaRSA to DASA to accelerate the convergence. The switching criterion is meant to guarantee that asymptotically only DASA is used.

At each step, starting from the current estimate $\boldsymbol{\lambda}^k$ and the corresponding sets $\mathcal{Z}^k = \mathcal{Z}(\boldsymbol{\lambda}^k)$ and $\mathcal{E}^k_\pm = \mathcal{E}_\pm(\boldsymbol{\lambda}^k)$, DASA finds a maximizer $\boldsymbol{\mu}$ for the local dual function

$$L_k(\boldsymbol{\lambda}) = \inf_{\mathbf{x},\,\mathbf{b}} \left\{ \mathcal{L}(\mathbf{x}, \mathbf{b}, \boldsymbol{\lambda}) \ : \ x_i = 0,\, \forall i \in \mathcal{Z}^k \ \wedge \ b_j = l_j,\, \forall j \in \mathcal{E}^k_+ \ \wedge \ b_j = u_j,\, \forall j \in \mathcal{E}^k_- \right\}$$

and uses it to compute $\boldsymbol{\lambda}^{k+1}$ by means of a line search. Observe that, by setting $\mathcal{R} = \mathcal{E}^k_+ \cup \mathcal{E}^k_-$, $\mathcal{C} = \{1, \ldots, n\} \setminus \mathcal{Z}^k$, and $\mu_j = 0$ for $i \notin \mathcal{R}$, the maximizer of L_k can be computed by solving the linear system

$$A_{\mathcal{R}\mathcal{C}} A^T_{\mathcal{R}\mathcal{C}} \boldsymbol{\mu}_{\mathcal{R}} = \mathbf{b}_{\mathcal{R}} - A_{\mathcal{R}\mathcal{C}} \mathbf{y}_{\mathcal{C}},$$

where $\mathbf{b}$ is the vector given in the definition of L_k. In the implementation of DASS provided by the authors this problem is solved by a preconditioned conjugate gradient method. DASA stops returning the solution to (2.51), from which the solution $\mathbf{x}^*$ to (2.50) can be derived by setting $\mathbf{x}^* = \mathbf{x}(\boldsymbol{\lambda}^*)$.

Chapter 3

A subspace accelerated gradient projection framework for QPs

Here we propose an active-set framework for the solution of problem (1.1) based on gradient projection. The proposed framework is based on the two-phase framework introduced by Calamai and Moré [27] and, as the GPCG method developed by Moré and Toraldo [109] (for strictly convex BQPs), uses the gradient projection to identify the active set at the solution, alternating it with unconstrained minimization steps onto the reduced subspace defined by the current active set. By a reformulation of the Karush–Kuhn–Tucker conditions we are able to define suitable generalizations of the free and the chopped gradient defined for BQPs and to introduce, for problem (1.1), the idea of proportional iterate introduced in [14, 53]. This translates into the possibility of switching between the identification and the minimization phase by comparing a measure of optimality within the reduced space with a measure of the "quality" of the current active set. We prove finite convergence of any method fitting into the proposed framework for strictly convex quadratic problems even in case of degeneracy at the solution, provided that a method with finite termination properties is used in the solution of the equality constrained subproblems.

3.1 Reformulating stationarity results for QPs

We recall that a stationary point for problem (1.1) can be characterized by the Karush–Kuhn–Tucker conditions.

Definition 3.1.1. *A feasible point* $\mathbf{x}^*$ *is a stationary point for problem* (1.1) *if and only if there exist Lagrange multipliers vectors* $\boldsymbol{\theta}^* \in \mathbb{R}^m$ *and* $\boldsymbol{\lambda}^* \in \mathbb{R}^n$ *such that*

$$\mathbf{g}^* = \sum_{i=1}^{n} \lambda_i^* \mathbf{e}_i + \sum_{j=1}^{m} \theta_j^* \mathbf{a}_j = \sum_{i=1}^{n} \lambda_i^* \mathbf{e}_i + A^T \boldsymbol{\theta}^*, \tag{3.1}$$

$$\lambda_i^* \geq 0 \;\; \textit{if} \;\; i \in \mathcal{A}_l^*, \quad \lambda_i^* \leq 0 \;\; \textit{if} \;\; i \in \mathcal{A}_u^*, \quad \lambda_i^* = 0 \;\; \textit{if} \;\; i \in \mathcal{F}^*, \tag{3.2}$$

The stationary point $\mathbf{x}^*$ *is said to be* non-degenerate *w.r.t. the bound constraints if the inequalities in* (3.2) *are strict, i.e.* $\lambda_i^* > 0$ *if* $i \in \mathcal{A}_l^*$ *and* $\lambda_i^* < 0$ *if* $i \in \mathcal{A}_u^*$. *Otherwise the point is said to be* degenerate.

Since we are interested in building an estimate for the Lagrange multipliers, we make the following assumption which guarantees their uniqueness.

Assumption 3.1.2 (Linear Independence Constraint Qualification - LICQ)**.** *Let* $\mathbf{x}^*$ *be a stationary point of* (1.1)*, then the active constraint normals*

$$\{\mathbf{a}_j \ : \ j = 1, \dots, m\} \cup \{\mathbf{e}_i \ : \ i \in \mathcal{A}^*\}$$

are linearly independent.

Since $\boldsymbol{\lambda}_{\mathcal{F}^*} = 0$, the KKT conditions (3.1) and (3.2) can be equivalently written as

$$\mathbf{g}^*_{\mathcal{F}^*} - \left[A^T\boldsymbol{\theta}^*\right]_{\mathcal{F}^*} = 0, \tag{3.3}$$

$$\lambda^*_i = g^*_i - \left[A^T\boldsymbol{\theta}^*\right]_i \geq 0, \ \text{ if } \ i \in \mathcal{A}^*_l, \tag{3.4}$$

$$\lambda^*_i = g^*_i - \left[A^T\boldsymbol{\theta}^*\right]_i \leq 0, \ \text{ if } \ i \in \mathcal{A}^*_u. \tag{3.5}$$

Since Assumption 3.1.2 holds, it has to be $|\mathcal{A}^*| \leq n - m$, or equivalently, $|\mathcal{F}^*| \geq m$. Moreover, the matrix $A_{\star\mathcal{F}^*}$ has full row-rank, hence

$$M = A_{\star\mathcal{F}^*} A^T_{\star\mathcal{F}^*} \in \mathbb{R}^{m\times m}$$

has full rank, i.e. is invertible.

Condition (3.3), which can be rewritten as

$$\mathbf{g}^*_{\mathcal{F}^*} = \left[A^T\boldsymbol{\theta}^*\right]_{\mathcal{F}^*} = A^T_{\star\mathcal{F}^*}\,\boldsymbol{\theta}^*,$$

by premultiplying by $A_{\star\mathcal{F}^*}$, leads to

$$\boldsymbol{\theta}^* = M^{-1}\, A_{\star\mathcal{F}^*}\, \mathbf{g}^*_{\mathcal{F}^*}. \tag{3.6}$$

It's worth noting that, if $|\mathcal{F}^*| = m$, (3.6) becomes

$$\boldsymbol{\theta}^* = A^{-T}_{\star\mathcal{F}^*}\, \mathbf{g}^*_{\mathcal{F}^*}.$$

From (3.6), (3.4) and (3.5) we can compute $\boldsymbol{\lambda}^*$.

Consider now any point $\mathbf{x} \in \Omega$ and suppose, for now, that $|\mathcal{F}(\mathbf{x})| > 0$ and $r = \text{rank}(A_{\star\mathcal{F}}) \neq 0$. Although $\text{rank}(A) = m$, removing columns from it to obtain $A_{\star\mathcal{F}}$ can lead to $r < m$, even in the case in which $|\mathcal{F}| \geq m$. Let $\mathfrak{J}(\mathbf{x})$ be the subset of the power set of $\{1, \dots, m\}$ containing all the subsets $\mathcal{J} = \{j_1, \dots, j_r\}$ of indices corresponding to a maximal set of independent rows of $A_{\star\mathcal{F}}$. For any $\mathcal{J} \in \mathfrak{J}(\mathbf{x})$, by setting $\widetilde{A} \equiv \widetilde{A}(\mathcal{J}) = A_{\mathcal{J}\mathcal{F}}$, and following a similar procedure as in (3.6), we define the vector

$$\boldsymbol{\xi}(\mathbf{x}; \mathcal{J}) := (\widetilde{A}\,\widetilde{A}^T)^{-1}\,\widetilde{A}\,\mathbf{g}_{\mathcal{F}} \in \mathbb{R}^r. \tag{3.7}$$

We can therefore define the vector $\boldsymbol{\theta}(\mathbf{x}; \mathcal{J}) \in \mathbb{R}^m$ as

$$\boldsymbol{\theta}_{\mathcal{J}}(\mathbf{x}; \mathcal{J}) = \boldsymbol{\xi}(\mathbf{x}; \mathcal{J}), \qquad \theta_j(\mathbf{x}; \mathcal{J}) = 0 \ \ \forall j \notin \mathcal{J}. \tag{3.8}$$

When $r = m$ we have that $\mathfrak{J}(\mathbf{x}) = \{\{1, \ldots, m\}\}$. Since there is no more dependence on the choice of $\mathcal{J}$, $\boldsymbol{\theta}(\mathbf{x}) = \boldsymbol{\xi}(\mathbf{x})$ is uniquely defined.

In the cases in which either $\mathcal{F}(\mathbf{x}) = \emptyset$ or $\text{rank}(A_{\star\mathcal{F}}) = 0$ (corresponding to the case in which the linear equality constraints depend on the normal of the active bound constraints at $\mathbf{x}$), we set

$$\boldsymbol{\theta}(\mathbf{x}) = \mathbf{0}. \tag{3.9}$$

Starting from (3.7)-(3.9) we can introduce the vector

$$\mathbf{h}(\mathbf{x}; \mathcal{J}) := \mathbf{g}(\mathbf{x}) - A^T \boldsymbol{\theta}(\mathbf{x}; \mathcal{J}); \tag{3.10}$$

from (3.3)-(3.5), a sufficient and necessary condition for $\mathbf{x}$ being a stationary point is that, for some $\mathcal{J} \in \mathfrak{J}(\mathbf{x})$,

$$h_i(\mathbf{x}; \mathcal{J}) = 0, \text{ if } i \in \mathcal{F}, \quad h_i(\mathbf{x}; \mathcal{J}) \geq 0, \text{ if } i \in \mathcal{A}_l, \quad h_i(\mathbf{x}; \mathcal{J}) \leq 0, \text{ if } i \in \mathcal{A}_u. \tag{3.11}$$

Remark 3.1.3. *We note that, because of the definition of $\boldsymbol{\theta}(\mathbf{x}; \mathcal{J})$ in* (3.7)-(3.8)*,*

$$\mathbf{h}_{\mathcal{F}}(\mathbf{x}; \mathcal{J}) = P_{\{A_{\star\mathcal{F}}\}^{\perp}} (\mathbf{g}_{\mathcal{F}}), \tag{3.12}$$

where $\mathcal{F} = \mathcal{F}(\mathbf{x})$ and $P_{\{A_{\mathcal{F}}\}^{\perp}} \in \mathbb{R}^{|\mathcal{F}| \times |\mathcal{F}|}$ is the orthogonal projection onto the subspace of $\mathbb{R}^{|\mathcal{F}|}$ orthogonal to the rows of $A_{\star\mathcal{F}}$ (i.e. the nullspace of $A_{\star\mathcal{F}}$). Even if, when $A_{\star\mathcal{F}}$ is rank deficient, the definition of $\boldsymbol{\theta}(\mathbf{x}; \mathcal{J})$ and $\mathbf{h}(\mathbf{x}; \mathcal{J})$ depends on the particular choice of the subset $\mathcal{J} \in \mathfrak{J}(\mathbf{x})$, vector $\mathbf{h}_{\mathcal{F}}(\mathbf{x})$ is uniquely defined.

Based on (3.11) we give the following definition of binding set.

Definition 3.1.4. *Let $\mathbf{x} \in \Omega$. Given $\mathcal{J} \in \mathfrak{J}(\mathbf{x})$, the* binding set (associated with $\mathcal{J}$) *at $\mathbf{x}$ is defined as*

$$\mathcal{B}(\mathbf{x}; \mathcal{J}) = \{i \ : \ (i \in \mathcal{A}_l \ \wedge \ h_i(\mathbf{x}; \mathcal{J}) \geq 0) \vee (i \in \mathcal{A}_u \ \wedge \ h_i(\mathbf{x}; \mathcal{J}) \leq 0)\}. \tag{3.13}$$

When $\text{rank}(A_{\star\mathcal{F}}) = m$, the binding set is unique and will be denoted as $\mathcal{B}(\mathbf{x})$.

Observe that in the case of bound constrained problems, $\mathbf{h}(\mathbf{x}) = \mathbf{g}(\mathbf{x})$ and (3.13) corresponds to the standard definition of binding set.

It is also possible to provide an estimate of the Lagrange multipliers based on (3.8)-(3.10).

Theorem 3.1.5. *Assume that $\left\{\mathbf{x}^k\right\}$ is a sequence in Ω which converges to a non-degenerate stationary point $\mathbf{x}^*$, and such that $\mathcal{A}(\mathbf{x}^k) = \mathcal{A}(\mathbf{x}^*)$ for all $k > \overline{k}$. Consider for each $k > \overline{k}$ the vector $\boldsymbol{\theta}^k = \boldsymbol{\theta}(\mathbf{x}^k)$ and the vector $\mathbf{h}^k = \mathbf{h}(\mathbf{x}^k)$. We have*

$$\begin{aligned} \lim_{k > \overline{k},\, k \to \infty} \theta_i^k &= \theta_i^*, \\ \lim_{k > \overline{k},\, k \to \infty} \lambda_i^k &= \lambda_i^*, \ \ i \in \mathcal{A}(\mathbf{x}^*), \end{aligned}$$

where $\boldsymbol{\lambda}^k = \boldsymbol{\lambda}(\mathbf{x}^k) \in \mathbb{R}^n$ is defined as

$$\lambda_i^k = \begin{cases} h_i^k, & \text{if } i \in \mathcal{A}(\mathbf{x}^k), \\ 0, & \text{if } i \in \mathcal{F}(\mathbf{x}^k). \end{cases}$$

Proof. For all $k > \overline{k}$, $\mathcal{F}^k = \mathcal{F}^*$, the matrix $A_{\star\mathcal{F}}$ is full rank and $\boldsymbol{\theta}(\mathbf{x}^k)$ and $\mathbf{h}(\mathbf{x}^k)$ are uniquely defined. Moreover, the matrix $\widetilde{A}$ in (3.7) coincides with $A_{\star\mathcal{F}^*}$ in (3.6). The thesis follows then from the continuity of $\nabla f(\mathbf{x})$. □

3.1.1 Least-Squares multipliers estimates

The Lagrange multipliers estimate previously introduced is indeed the Least-Squares (LS) multipliers estimate, defined e.g.,in [34, Section 12.4.1] and [78, Theorem 2.3]. For problem (1.1), given a point $\mathbf{x} \in \Omega$ the LS estimate is defined as

$$\begin{aligned} \underset{\boldsymbol{\theta},\boldsymbol{\lambda}}{\operatorname{argmin}} \quad & \left\| \mathbf{g} - A^T\boldsymbol{\theta} - \sum_{i=1}^{n} \lambda_i \mathbf{e}_i \right\|_2 \\ \text{s.t.} \quad & \lambda_i \geq 0 \ \text{ if } \ i \in \mathcal{A}_l, \\ & \lambda_i \leq 0 \ \text{ if } \ i \in \mathcal{A}_u, \\ & \lambda_i = 0 \ \text{ if } \ i \in \mathcal{F}. \end{aligned} \tag{3.14}$$

By defining

$$\mathbf{y} = \left(\boldsymbol{\theta}^T,\ \boldsymbol{\lambda}_{\mathcal{A}_l}^T,\ \boldsymbol{\lambda}_{\mathcal{A}_u}^T\right)^T, \quad \text{and} \quad B = \left(A^T,\ I_{\star\mathcal{A}_l},\ I_{\star\mathcal{A}_u}\right),$$

the nonzero components of the solution to (3.14) can be found by minimizing w.r.t. $\mathbf{y}$ the function

$$h(\mathbf{y}) = \mathbf{y}^T B^T B\mathbf{y} - 2\,\mathbf{g}^T B\mathbf{y},$$

whose unconstrained minimizer can be found by solving the system

$$B^T B\mathbf{y} - B^T\mathbf{g} = 0. \tag{3.15}$$

By making explicit the form of the system matrix $B^T B$, (3.15) can be written as

$$\begin{pmatrix} A\,A^T & A_{\star\mathcal{A}_l} & A_{\star\mathcal{A}_u} \\ A_{\star\mathcal{A}_l}^T & I & 0 \\ A_{\star\mathcal{A}_u}^T & 0 & I \end{pmatrix} \begin{pmatrix} \boldsymbol{\theta} \\ \boldsymbol{\lambda}_{\mathcal{A}_l} \\ \boldsymbol{\lambda}_{\mathcal{A}_u} \end{pmatrix} - \begin{pmatrix} A\,\mathbf{g} \\ \mathbf{g}_{\mathcal{A}_l} \\ \mathbf{g}_{\mathcal{A}_u} \end{pmatrix} = \mathbf{0} \tag{3.16}$$

or equivalently as the linear system

$$\begin{cases} A\,A^T\boldsymbol{\theta} + A_{\star\mathcal{A}_l}\boldsymbol{\lambda}_{\mathcal{A}_l} + A_{\star\mathcal{A}_u}\boldsymbol{\lambda}_{\mathcal{A}_u} & -A\,\mathbf{g} & = & \mathbf{0} \\ A_{\star\mathcal{A}_l}^T\,\boldsymbol{\theta} + \boldsymbol{\lambda}_{\mathcal{A}_l} & -\mathbf{g}_{\mathcal{A}_l} & = & \mathbf{0} \\ A_{\star\mathcal{A}_u}^T\,\boldsymbol{\theta} + \boldsymbol{\lambda}_{\mathcal{A}_u} & -\mathbf{g}_{\mathcal{A}_u} & = & \mathbf{0} \end{cases}. \tag{3.17}$$

By premultiplying the second equation $A_{\star\mathcal{A}_l}$ and the third one by $A_{\star\mathcal{A}_u}$, and substituting in the first equation, we obtain

$$\begin{aligned} \mathbf{0} \quad &= AA^T\boldsymbol{\theta} - A_{\star\mathcal{A}_l}A_{\star\mathcal{A}_l}^T\,\boldsymbol{\theta} - A_{\star\mathcal{A}_u}A_{\star\mathcal{A}_u}^T\,\boldsymbol{\theta} + \\ &\quad - A\,\mathbf{g} + A_{\star\mathcal{A}_l}\mathbf{g}_{\mathcal{A}_l} + A_{\star\mathcal{A}_u}\mathbf{g}_{\mathcal{A}_u} = \\ &= A_{\star\mathcal{F}}\left[A^T\boldsymbol{\theta}\right]_{\mathcal{F}} - A_{\star\mathcal{F}}\mathbf{g}_{\mathcal{F}}, \end{aligned} \tag{3.18}$$

where we exploited the fact that, for an index set $\mathcal{S} \subseteq \{1,\ldots,n\}$,

$$A_{\star\mathcal{S}}^T\,\boldsymbol{\theta} = \left[A^T\boldsymbol{\theta}\right]_{\mathcal{S}}$$

and that $\mathcal{F} = \{1,\ldots,n\} \setminus (\mathcal{A}_l \cup \mathcal{A}_u)$. The obtained relation coincides with the relation (3.3), found for the Lagrange multipliers related to the solution of (1.1), given the nullspace of $A_{\star\mathcal{F}}$ is trivial, i.e. $\mathcal{N}(A_{\star\mathcal{F}}) = \{\mathbf{0}\}$.

3.2 The projected gradient

We recall that in Chapter 2 we introduced the projected gradient as a measure of stationarity.

It could be argued that the projected gradient is inappropriate to measure closeness to a stationary point, since it is only lower semicontinuous (see [27, Lemma 3.3]). However, Calamai and Moré in [27] showed that the limit points of a sequence $\{\mathbf{x}^k\}$ generated by a gradient projection algorithm, with bounded step lengths satisfying suitable sufficient decrease conditions, are stationary and

$$\lim_{k\to\infty} \|\nabla_\Omega f(\mathbf{x}^k)\| = 0. \tag{3.19}$$

Similar results hold for a more general algorithmic framework (see [27, Algorithm 5.3]), in which gradient projection steps are alternated with general descent steps. Another important issue is that, for any sequence $\{\mathbf{x}^k\}$ converging to a non-degenerate stationary point $\mathbf{x}^*$, if (3.19) holds then $\mathcal{A}^k = \mathcal{A}^*$ for all k sufficiently large. However, for problem (1.1), condition (3.19) has an important meaning in terms of active constraints identification even in case of degeneracy, provided Assumption 3.1.2 holds.

The following proposition summarizes the convergence properties for a sequence $\{\mathbf{x}^k\}$ satisfying (3.19), both in terms of stationarity and active set identification.

Theorem 3.2.1. *Assume that $\left\{\mathbf{x}^k\right\}$ is a sequence in Ω that converges to a point $\mathbf{x}^*$ and $\lim_{k\to\infty} \|\nabla_\Omega f(\mathbf{x}^k)\| = 0$. Then*

(i) *$\mathbf{x}^*$ is a stationary point for problem (1.1);*

(ii) *if Assumption 3.1.2 holds, then $\mathcal{A}^*_N \subseteq \mathcal{A}^k$ for all k sufficiently large, where $\mathcal{A}^*_N = \{i \in \mathcal{A}^* \,:\, \lambda^*_i \neq 0\}$ and λ_i is the Lagrange multiplier associated with the i-th bound constraint.*

Proof. Item (i) trivially follows from the lower semicontinuity of $\|\nabla_\Omega f(\mathbf{x})\|$.

Item (ii) extends [27, Theorem 4.1] and [51, Theorem 2.3] to degenerate stationary points that satisfy Assumption 3.1.2. We first note that, since $\left\{\mathbf{x}^k\right\}$ converges to $\mathbf{x}^*$, we have $\mathcal{F}^* \subseteq \mathcal{F}^k$ and hence $\mathcal{A}^k \subseteq \mathcal{A}^*$ for all k sufficiently large. The proof is by contradiction. Assume that there is an index $\bar{\imath}$ and an infinite set $K \subseteq \mathbb{N}$ such that $\bar{\imath} \in \mathcal{A}^*_N \setminus \mathcal{A}^k$ for all $k \in K$. Without loss of generality, we assume $x^*_{\bar{\imath}} = u_{\bar{\imath}}$ and thus $\lambda^*_{\bar{\imath}} < 0$. Let P_Φ be the orthogonal projection onto

$$\Phi = \{\mathbf{v} \in \mathbb{R}^n \,:\, A\,\mathbf{v} = 0 \;\wedge\; (v_i = 0,\ \forall\, i \in \mathcal{A}^* \setminus \{\bar{\imath}\})\}\,.$$

Assumption 3.1.2 implies that $P_\Phi(\mathbf{e}_{\bar{\imath}}) \neq 0$. Since $\bar{\imath} \notin \mathcal{A}(\mathbf{x}^k)$, it is $P_\Phi(\mathbf{e}_{\bar{\imath}}) \in T_\Omega(\mathbf{x}^k)$. Then, by [27, Lemma 3.1],

$$(\mathbf{g}^k)^T P_\Phi(\mathbf{e}_{\bar{\imath}}) \geq -\left\|\nabla_\Omega f(\mathbf{x}^k)\right\| \, \|P_\Phi(\mathbf{e}_{\bar{\imath}})\|\,,$$

and since $\left\{\mathbf{x}^k\right\}$ converges to $\mathbf{x}^*$ and $\left\{\left\|\nabla_\Omega f(\mathbf{x}^k)\right\|\right\}$ converges to 0, we have

$$(\mathbf{g}^*)^T P_\Phi(\mathbf{e}_{\bar{\imath}}) \geq 0.$$

On the other hand, by (3.1) and the definition of P_Φ we get

$$(\mathbf{g}^*)^T P_\Phi(\mathbf{e}_{\bar{i}}) = \sum_{i \in \mathcal{A}^*} \lambda_i^* \mathbf{e}_i^T P_\Phi(\mathbf{e}_{\bar{i}}) + (\boldsymbol{\theta}^*)^T A\, P_\Phi(\mathbf{e}_{\bar{i}}) = \lambda_{\bar{i}}^* \mathbf{e}_{\bar{i}}^T P_\Phi(\mathbf{e}_{\bar{i}}) < 0,$$

where the last inequality derives from $\lambda_{\bar{i}}^* < 0$ and $(\mathbf{e}_{\bar{i}})^T P_\Phi(\mathbf{e}_{\bar{i}}) = \|P_\Phi(\mathbf{e}_{\bar{i}})\|^2 > 0$. The contradiction proves that the set K is finite, and hence $\bar{i} \in \mathcal{A}^k$ for all k sufficiently large. □

By Theorem 3.2.1, if an algorithm is able to drive the projected gradient toward zero, then it is able to identify the active variables that are non-degenerate at the solution in a finite number of iterations. Because of this interesting property of GP algorithms, in this thesis we will deal with the development of efficient active-set algorithms for problems of the form (1.1). We will now provide the generalization of the free gradient and the chopped gradient to the case of problem (1.1) and prove some useful properties.

3.3 The free and chopped gradients

We start by defining the free gradient $\boldsymbol{\varphi}(\mathbf{x})$ at $\mathbf{x} \in \Omega$ for problem (1.1).

Definition 3.3.1. *For any $\mathbf{x} \in \Omega$, the free gradient $\boldsymbol{\varphi}(\mathbf{x}) \in \mathbb{R}^n$ is defined as*

$$\boldsymbol{\varphi}_{\mathcal{F}}(\mathbf{x}) = \mathbf{h}_{\mathcal{F}}(\mathbf{x}), \qquad \boldsymbol{\varphi}_{\mathcal{A}}(\mathbf{x}) = \mathbf{0}. \tag{3.20}$$

Remark 3.1.3 ensures that $\boldsymbol{\varphi}(\mathbf{x})$ is uniquely defined also in the case in which $\operatorname{rank}(A_{\star\mathcal{F}}) < m$.

The following theorems state some properties of $\boldsymbol{\varphi}(\mathbf{x})$, including its relationship with the projected gradient.

Lemma 3.3.2. *Let $\mathbf{x} \in \Omega$. Then $\boldsymbol{\varphi}(\mathbf{x}) = 0$ if and only if $\mathbf{x}$ is a stationary point for*

$$\begin{aligned} \min \quad & f(\mathbf{u}), \\ s.t. \quad & \mathbf{u} \in \Omega(\mathbf{x}). \end{aligned} \tag{3.21}$$

Proof. Since $\boldsymbol{\varphi}(\mathbf{x})$ is invariant with respect to $\mathcal{J} \in \mathfrak{J}(\mathbf{x})$, w.l.o.g. we can choose any $\hat{\mathcal{J}} \in \mathfrak{J}(\mathbf{x})$. Because of (3.10), $\boldsymbol{\varphi}(\mathbf{x}) = 0$ if and only if

$$g_i - \left[A^T \boldsymbol{\theta}(\mathbf{x}; \hat{\mathcal{J}})\right]_i = 0, \quad \forall\, i \in \mathcal{F}(\mathbf{x}). \tag{3.22}$$

On the other hand, $\mathbf{x}$ is a stationary point for problem (3.21) if and only if

$$\mathbf{g} = \sum_{i \in \mathcal{A}(\mathbf{x})} \nu_i \mathbf{e}_i + A^T \boldsymbol{\mu},$$

with $\nu_i \in \mathbb{R}$ and $\boldsymbol{\mu} \in \mathbb{R}^m$, which implies

$$g_i = \left[A^T \boldsymbol{\mu}\right]_i, \quad \forall\, i \in \mathcal{F}(\mathbf{x}). \tag{3.23}$$

By (3.22) and (3.23), and writing $\mathcal{F} = \mathcal{F}(\mathbf{x})$, we obtain that

$$A_{\star\mathcal{F}}^T \left(\boldsymbol{\theta}(\mathbf{x}; \hat{\mathcal{J}}) - \boldsymbol{\mu}\right) = \mathbf{0}.$$

Observe that if matrix $A_{\star\mathcal{F}}$ is rank deficient, from its definition, the nonzero components of $\boldsymbol{\theta}(\mathbf{x})$ correspond to the indices in $\hat{\mathcal{J}}$. W.l.o.g., since the linear combination in (3.23) is not unique, the vector $\boldsymbol{\mu}$ can be taken such that it has zero components for the indices in $\{1,\ldots,m\}\setminus\hat{\mathcal{J}}$. This allows us to prove that $\boldsymbol{\theta}(\mathbf{x};\hat{\mathcal{J}})=\boldsymbol{\mu}$, therefore the thesis. □

Remark 3.3.3. *Theorem 3.3.2 shows that $\boldsymbol{\varphi}(\mathbf{x})$ can be considered as a measure of optimality within the reduced space determined by the active variables at $\mathbf{x}$.*

Lemma 3.3.4. *For any $\mathbf{x}\in\Omega$, $\boldsymbol{\varphi}(\mathbf{x})$ is the orthogonal projection of $-\nabla_\Omega f(\mathbf{x})$ onto $\Omega_0(\mathbf{x})$, where $\Omega_0(\mathbf{x})$ is given in* (1.3)*. Furthermore,*

$$\|\boldsymbol{\varphi}(\mathbf{x})\|^2=-(\nabla_\Omega f(\mathbf{x}))^T\boldsymbol{\varphi}(\mathbf{x}). \tag{3.24}$$

Proof. By the definition of projected gradient (see (2.14)),

$$A\left(\nabla_\Omega f(\mathbf{x})\right)=0, \tag{3.25}$$
$$\nabla_\Omega f(\mathbf{x})=-\mathbf{g}+A^T\boldsymbol{\nu}+\boldsymbol{\mu} \tag{3.26}$$

for some $\boldsymbol{\nu}\in\mathbb{R}^m$ and $\boldsymbol{\mu}\in\mathbb{R}^n$, with

$$\mu_i=0 \text{ if } i\in\mathcal{F}(\mathbf{x}),\quad \mu_i\geq 0 \text{ if } i\in\mathcal{A}_l(\mathbf{x}),\quad \mu_i\leq 0 \text{ if } i\in\mathcal{A}_u(\mathbf{x}).$$

W.l.o.g. consider any $\mathcal{J}\in\mathfrak{J}(\mathbf{x})$ and set $\boldsymbol{\theta}=\boldsymbol{\theta}(\mathbf{x};\mathcal{J})$ and $\mathbf{h}=\mathbf{h}(\mathbf{x};\mathcal{J})$ as defined in (3.7)-(3.10). Let

$$\boldsymbol{\sigma}=\boldsymbol{\nu}-\boldsymbol{\theta},\quad \tau_i=\mu_i-h_i \text{ if } i\in\mathcal{A}(\mathbf{x}),\quad \tau_i=0 \text{ if } i\in\mathcal{F}(\mathbf{x}).$$

Then (3.26) can be written as

$$\begin{aligned} h_i &= -(\nabla_\Omega f)_i(\mathbf{x})+\left[A^T\boldsymbol{\sigma}\right]_i+\tau_i \quad \text{if } i\in\mathcal{F}(\mathbf{x}),\\ 0 &= -(\nabla_\Omega f)_i(\mathbf{x})+\left[A^T\boldsymbol{\sigma}\right]_i+\tau_i \quad \text{if } i\in\mathcal{A}(\mathbf{x}),\end{aligned}$$

or, equivalently,

$$\boldsymbol{\varphi}(\mathbf{x})=-\nabla_\Omega f(\mathbf{x})+A^T\boldsymbol{\sigma}+\boldsymbol{\tau}, \tag{3.27}$$

with $\tau_i=0$ if $i\in\mathcal{F}(\mathbf{x})$. This, with (3.25) and $\varphi_i=0$ for $i\in\mathcal{A}(\mathbf{x})$, proves that

$$\boldsymbol{\varphi}(\mathbf{x})=\operatorname{argmin}\left\{\|\mathbf{v}+\nabla_\Omega f(\mathbf{x})\| \text{ s.t. } \mathbf{v}\in\Omega_0(\mathbf{x})\right\},$$

which is the first part of the thesis. Equation (3.24) follows from (3.27) and the definition of $\boldsymbol{\varphi}(\mathbf{x})$. □

Lemma 3.3.5. *Let $\mathbf{x}\in\Omega$ and let $\operatorname{rank}(A_{\star\mathcal{F}})=m$. Then $\mathcal{B}(\mathbf{x})$, is uniquely defined and we have $\mathcal{A}(\mathbf{x})=\mathcal{B}(\mathbf{x})$, if and only if*

$$\varphi(\mathbf{x})=-\nabla_\Omega f(\mathbf{x}). \tag{3.28}$$

Proof. We recall that in this case the vectors $\boldsymbol{\theta}(\mathbf{x})$ and $\mathbf{h}(\mathbf{x})$ are unique.

Assume that $\mathcal{A}(\mathbf{x}) = \mathcal{B}(\mathbf{x})$. Condition (3.28) can be written as

$$-\boldsymbol{\varphi}(\mathbf{x}) = \operatorname{argmin}\{\|\mathbf{v}+\mathbf{g}\| \text{ s.t. } \mathbf{v} \in T_\Omega(\mathbf{x})\}. \tag{3.29}$$

Since, by Lemma 3.3.4, $-\boldsymbol{\varphi}(\mathbf{x}) \in \Omega_0(\mathbf{x}) \subset T_\Omega(\mathbf{x})$, we need only to prove that

$$-\boldsymbol{\varphi}(\mathbf{x}) = -\mathbf{g} + A^T\boldsymbol{\nu} + \boldsymbol{\mu},$$

for some $\boldsymbol{\nu} \in \mathbb{R}^m$ and $\boldsymbol{\mu} \in \mathbb{R}^n$, with $\mu_i = 0$ if $i \in \mathcal{F}$, $\mu_i \geq 0$ if $i \in \mathcal{A}_l(\mathbf{x})$, $\mu_i \leq 0$ if $i \in \mathcal{A}_u(\mathbf{x})$. Since $\mathcal{A}(\mathbf{x}) = \mathcal{B}(\mathbf{x})$, recalling the definition of binding set, the previous equality holds trivially by setting $\boldsymbol{\nu} = \boldsymbol{\theta}(\mathbf{x})$, $\mu_i = h_i(\mathbf{x})$ for $i \in \mathcal{A}(\mathbf{x})$, and $\mu_i = 0$ otherwise.

Suppose, now, that (3.28) holds. From the definition of $\boldsymbol{\varphi}$ and (3.26), it follows that (3.28) can be written as

$$\varphi_i(\mathbf{x}) = g_i - \left[A^T\boldsymbol{\theta}(\mathbf{x})\right]_i = g_i - \left[A^T\boldsymbol{\nu}\right]_i \quad \forall\, i \in \mathcal{F}(\mathbf{x}), \tag{3.30}$$

$$0 = g_i - \left[A^T\boldsymbol{\nu}\right]_i - \mu_i \quad \forall\, i \in \mathcal{A}(\mathbf{x}), \tag{3.31}$$

with $\mu_i \geq 0$ if $i \in \mathcal{A}_l(\mathbf{x})$ and $\mu_i \leq 0$ if $i \in \mathcal{A}_u(\mathbf{x})$. From (3.30) we get

$$A^T_{\star\mathcal{F}}\left(\boldsymbol{\theta}(\mathbf{x}) - \boldsymbol{\nu}\right) = \mathbf{0}$$

which, since $\operatorname{rank}(A_{\star\mathcal{F}}) = m$, implies $\boldsymbol{\theta}(\mathbf{x}) = \boldsymbol{\nu}$, and then, from (3.31) and the definition of $h(\mathbf{x})$,

$$h_i(\mathbf{x}) \geq 0 \text{ if } i \in \mathcal{A}_l(\mathbf{x}), \quad h_i(\mathbf{x}) \leq 0 \text{ if } i \in \mathcal{A}_u(\mathbf{x});$$

thus $\mathcal{A}(\mathbf{x}) = \mathcal{B}(\mathbf{x})$. □

We observe that the hypotheses $\operatorname{rank}(A_{\star\mathcal{F}}) = m$ is not too restrictive. Since Assumption 3.1.2 holds, it always satisfied if, for example, $\mathcal{F}^* \subseteq \mathcal{F}$, which is true in a neighborhood of the solution.

If $\operatorname{rank}(A_{\star\mathcal{F}}) < m$, the binding set is no more unique and a weaker result can be proved.

Lemma 3.3.6. *Let $\mathbf{x} \in \Omega$. If there exists $\mathcal{J} \in \mathfrak{J}(\mathbf{x})$ such that $\mathcal{A}(\mathbf{x}) = \mathcal{B}(\mathbf{x};\mathcal{J})$, then*

$$\varphi(\mathbf{x}) = -\nabla_\Omega f(\mathbf{x}). \tag{3.32}$$

Proof. The proof is the same of the necessary condition in Lemma 3.3.5, with $\boldsymbol{\theta}(\mathbf{x};\mathcal{J})$ and $\mathbf{h}(\mathbf{x};\mathcal{J})$ in place of $\boldsymbol{\theta}(\mathbf{x})$ and $\mathbf{h}(\mathbf{x})$. □

Inspired by the two previous lemmas, we give the following definition.

Definition 3.3.7. *For any $\mathbf{x} \in \Omega$, the chopped gradient $\boldsymbol{\beta}(\mathbf{x})$ is defined as*

$$\boldsymbol{\beta}(\mathbf{x}) := -\nabla_\Omega f(\mathbf{x}) - \boldsymbol{\varphi}(\mathbf{x}). \tag{3.33}$$

Remark 3.3.8. *Because of Lemma 3.3.6, $\boldsymbol{\beta}(\mathbf{x}) \neq 0$ implies that for all the choices of $\mathcal{J}$ in $\mathfrak{J}(\mathbf{x})$, $\mathcal{A}(\mathbf{x}) \neq \mathcal{B}(\mathbf{x};\mathcal{J})$. Moreover, in the hypotheses of Lemma 3.3.5, we have that $\boldsymbol{\beta}(\mathbf{x}) = 0$ if and only if $\mathcal{A}(\mathbf{x}) = \mathcal{B}(\mathbf{x})$. Thus, $\boldsymbol{\beta}(\mathbf{x})$ can be regarded as a "measure of bindingness" of the active variables at $\mathbf{x}$.*

Some properties of $\boldsymbol{\beta}(\mathbf{x})$ are given next.

Lemma 3.3.9. *For any $\mathbf{x} \in \Omega$, $\boldsymbol{\beta}(\mathbf{x})$ has the following properties:*

$$\boldsymbol{\beta}(\mathbf{x}) \perp \boldsymbol{\varphi}(\mathbf{x}), \quad \boldsymbol{\beta}(\mathbf{x}) \in \{A\}^{\perp}, \tag{3.34}$$

$$-\boldsymbol{\beta}(\mathbf{x}) \in T_{\Omega}(\mathbf{x}). \tag{3.35}$$

Proof. Since

$$\boldsymbol{\beta}(\mathbf{x})^T\boldsymbol{\varphi}(\mathbf{x}) = (-\nabla_{\Omega} f(\mathbf{x}) - \boldsymbol{\varphi}(\mathbf{x}))^T\boldsymbol{\varphi}(\mathbf{x}) = (-\nabla_{\Omega} f(\mathbf{x}))^T\boldsymbol{\varphi}(\mathbf{x}) - \boldsymbol{\varphi}(\mathbf{x})^T\boldsymbol{\varphi}(\mathbf{x}),$$

the first orthogonality condition in (3.34) follows from (3.24). The second one follows from

$$A\,\boldsymbol{\beta}(\mathbf{x}) = A\,(-\nabla_{\Omega} f(\mathbf{x})) - A\,\boldsymbol{\varphi}(\mathbf{x}),$$

by observing that $\nabla_{\Omega} f(\mathbf{x})$ and $\boldsymbol{\varphi}(\mathbf{x})$ are in $\{A\}^{\perp}$. Finally, (3.35) trivially follows from the fact that $\boldsymbol{\varphi}(\mathbf{x}) \in \Omega_0(\mathbf{x})$ (see Lemma 3.3.4), and the definition of $\nabla_{\Omega} f(\mathbf{x})$. □

Theorem 3.3.10. *For any $\mathbf{x} \in \Omega$, $\|\boldsymbol{\beta}(\mathbf{x})\|^2 = \mathbf{g}^T\boldsymbol{\beta}(\mathbf{x})$.*

Proof. By [27, Lemma 3.1], we have $-\mathbf{g}^T\nabla_{\Omega} f(\mathbf{x}) = \|\nabla_{\Omega} f(\mathbf{x})\|^2$, which can be written as

$$\mathbf{g}^T\,(\boldsymbol{\varphi}(\mathbf{x}) + \boldsymbol{\beta}(\mathbf{x})) = \|\boldsymbol{\varphi}(\mathbf{x})\|^2 + \|\boldsymbol{\beta}(\mathbf{x})\|^2 \tag{3.36}$$

by exploiting (3.33) and (3.34). We note that the scalar product $\mathbf{g}^T\boldsymbol{\varphi}(\mathbf{x})$ involves only the entries corresponding to $\mathcal{F}(\mathbf{x})$. W.l.o.g. we can fix a $\mathcal{J} \in \mathfrak{J}(\mathbf{x})$. Since

$$\boldsymbol{\varphi}_{\mathcal{F}} = \mathbf{g}_{\mathcal{F}} - A_{\star\mathcal{F}}^T\boldsymbol{\theta},$$

where $\boldsymbol{\theta} \equiv \boldsymbol{\theta}(\mathbf{x};\mathcal{J})$ is given in (3.8)-(3.9), we get

$$\begin{aligned} \mathbf{g}^T\boldsymbol{\varphi} &= \|\mathbf{g}_{\mathcal{F}}\|^2 - \boldsymbol{\theta}^T A_{\star\mathcal{F}}\mathbf{g}_{\mathcal{F}}, \\ \|\boldsymbol{\varphi}\|^2 &= \|\mathbf{g}_{\mathcal{F}}\|^2 - 2\,\boldsymbol{\theta}^T A_{\star\mathcal{F}}\mathbf{g}_{\mathcal{F}} + \left\|A_{\star\mathcal{F}}^T\boldsymbol{\theta}\right\|^2, \end{aligned}$$

where for ease of notation we omitted the dependence from $\mathbf{x}$. By subtracting the two equations we get

$$\mathbf{g}^T\boldsymbol{\varphi} - \|\boldsymbol{\varphi}\|^2 = \boldsymbol{\theta}^T A_{\star\mathcal{F}}\mathbf{g}_{\mathcal{F}} - \boldsymbol{\theta}^T A_{\star\mathcal{F}}A_{\star\mathcal{F}}^T\boldsymbol{\theta}.$$

By recalling the definition of $\widetilde{A} \equiv \widetilde{A}(\mathcal{J})$ and $\boldsymbol{\xi} \equiv \boldsymbol{\xi}(\mathbf{x};\mathcal{J})$ in (3.7) and noting that $A_{\star\mathcal{F}}^T\boldsymbol{\theta} = \widetilde{A}^T\boldsymbol{\xi}$, the previous equation yields

$$\mathbf{g}^T\boldsymbol{\varphi} - \|\boldsymbol{\varphi}\|^2 = \boldsymbol{\xi}^T\widetilde{A}\,\mathbf{g}_{\mathcal{F}} - \boldsymbol{\xi}^T\widetilde{A}\widetilde{A}^T(\widetilde{A}\,\widetilde{A}^T)^{-1}\,\widetilde{A}\,\mathbf{g}_{\mathcal{F}} = 0;$$

then the thesis follows from (3.36). □

3.3.1 Proportional iterates for QPs

So far we managed to decompose the projected gradient $\nabla_\Omega f(\mathbf{x})$ into two parts: $-\boldsymbol{\varphi}(\mathbf{x})$, which provides a measure of stationarity within the reduced space determined by the active variables at $\mathbf{x}$, and $-\boldsymbol{\beta}(\mathbf{x})$, which gives a measures of bindingness of the active variables at $\mathbf{x}$. With this decomposition we can extend to problem (1.1) the definition (2.35) of proportional iterates introduced for the BQP case, as those $\mathbf{x}^k$ for which it holds

$$\|\boldsymbol{\beta}(\mathbf{x}^k)\|_\infty \leq \Gamma \|\boldsymbol{\varphi}(\mathbf{x}^k)\|, \tag{3.37}$$

with $\Gamma > 0$ a given constant. In the strictly convex case, disproportionality of $\mathbf{x}^k$ again guarantees that the solution of (1.1) does not belong to the face identified by the active variables at $\mathbf{x}^k$. This result is a consequence of the next theorem, which generalizes Theorem 3.2 in [53] and Theorem 3.8 in [51], and is the main result of this chapter.

Theorem 3.3.11. *Let H be the Hessian matrix in* (1.1) *and let $\mathcal{H} = V^T H V$ be positive definite, where $V \in \mathbb{R}^{n\times(n-m)}$ has orthonormal columns spanning $\{A\}^\perp$. Let $\mathbf{x} \in \Omega$ be such that $\|\boldsymbol{\beta}(\mathbf{x})\|_\infty > \kappa(\mathcal{H})^{1/2}\, \|\boldsymbol{\varphi}(\mathbf{x})\|_2$, and let $\overline{\mathbf{x}}$ be the solution of*

$$\begin{array}{ll} \min & f(\mathbf{u}), \\ \text{s.t.} & \mathbf{u} \in \Omega(\mathbf{x}), \end{array} \tag{3.38}$$

where $\Omega(\mathbf{x})$ is defined in (1.2). *If $\overline{\mathbf{x}} \in \Omega$, then $\boldsymbol{\beta}(\overline{\mathbf{x}}) \neq 0$.*

To prove Theorem 3.3.11, we need the lemma given next.

Lemma 3.3.12. *Let us consider the minimization problem*

$$\begin{array}{ll} \min & w(\mathbf{z}) := \frac{1}{2}\mathbf{z}^T K\, \mathbf{z} - \mathbf{p}^T\mathbf{z}, \\ \text{s.t.} & R\,\mathbf{z} = \mathbf{q}, \end{array} \tag{3.39}$$

where $K \in \mathbb{R}^{t\times t}$, $\mathbf{p} \in \mathbb{R}^t$, $R \in \mathbb{R}^{s\times t}$ with $t \geq s$ and $\operatorname{rank}(R) = s$, $\mathbf{q} \in \mathbb{R}^s$. Let $\Theta = \{\mathbf{z} \in \mathbb{R}^t : R\,\mathbf{z} = \mathbf{q}\}$ and $\Theta_0 = \{\mathbf{z} \in \mathbb{R}^t : R\,\mathbf{z} = \mathbf{0}\}$. Let P_{Θ_0} be the orthogonal projection onto Θ_0, and $U \in \mathbb{R}^{t\times(t-s)}$ a matrix with orthonormal columns spanning Θ_0. Finally, let $U^T K\, U$ be positive definite, and $\overline{\mathbf{z}}$ the solution of (3.39). *Then*

$$\mathbf{z} - \overline{\mathbf{z}} = B\, P_{\Theta_0} \nabla w(\mathbf{z}), \quad \forall \mathbf{z} \in \Theta, \tag{3.40}$$

where $B = U(U^T K U)^{-1} U^T$. Furthermore,

$$w(\mathbf{z}) - w(\overline{\mathbf{z}}) \leq \frac{1}{2}\, \|B\| \|P_{\Theta_0} \nabla w(\mathbf{z})\|^2. \tag{3.41}$$

Proof. Let $\mathbf{z} \in \Theta$; since $\mathbf{q} = R\,\mathbf{z}$ and $\operatorname{range}(U)$ is the space orthogonal to the rows of R, we have

$$\mathbf{z} = R^T \left(R\,R^T\right)^{-1} R\,\mathbf{z} + U\,\mathbf{y} = R^T \left(R\,R^T\right)^{-1} \mathbf{q} + U\,\mathbf{y} = \mathbf{r} + U\,\mathbf{y},$$

for some $\mathbf{y} \in \mathbb{R}^{t-s}$. Thus, (3.39) can be reduced to

$$\min\ \widetilde{w}(\mathbf{y}) := \frac{1}{2}\mathbf{y}^T U^T K\, U\mathbf{y} - (\mathbf{p}^T - \mathbf{r}^T K) U\mathbf{y}.$$

By writing $\overline{\mathbf{z}}$, the minimizer of (3.39), as $\overline{\mathbf{z}} = \mathbf{r} + U\,\overline{\mathbf{y}}$, we have

$$\mathbf{z} - \overline{\mathbf{z}} = U(\mathbf{y} - \overline{\mathbf{y}}) \tag{3.42}$$

and, by observing that $\nabla\widetilde{w}(\overline{\mathbf{y}}) = 0$, we obtain

$$\nabla\widetilde{w}(\mathbf{y}) = \nabla\widetilde{w}(\mathbf{y}) - \nabla\widetilde{w}(\overline{\mathbf{y}}) = U^T K U(\mathbf{y} - \overline{\mathbf{y}}) = U^T(\nabla w(\mathbf{z}) - \nabla w(\overline{\mathbf{z}})). \tag{3.43}$$

Since $\nabla w(\overline{\mathbf{z}}) = R\,\boldsymbol{\gamma}$ for some $\boldsymbol{\gamma} \in \mathbb{R}^t$, we get $UU^T\nabla w(\overline{\mathbf{z}}) = P_{\Theta_0}\nabla w(\overline{\mathbf{z}}) = \mathbf{0}$ and hence

$$U\nabla\widetilde{w}(\mathbf{y}) = UU^T(\nabla w(\mathbf{z}) - \nabla w(\overline{\mathbf{z}})) = P_{\Theta_0}\nabla w(\mathbf{z}). \tag{3.44}$$

From (3.42), (3.43) and (3.44) it follows that

$$\mathbf{z} - \overline{\mathbf{z}} = U(\mathbf{y} - \overline{\mathbf{y}}) = U(U^T K U)^{-1}U^T U\nabla\widetilde{w}(\mathbf{y}) = B\,P_{\Theta_0}\nabla w(\mathbf{z}),$$

which is (3.40).

Let $\phi(\mathbf{z}) = P_{\Theta_0}\nabla w(\mathbf{z})$. By applying (3.40), we get

$$w(\mathbf{z}) - w(\overline{\mathbf{z}}) = \frac{1}{2}(\mathbf{z} - \overline{\mathbf{z}})^T K(\mathbf{z} - \overline{\mathbf{z}}) = \frac{1}{2}\phi(\mathbf{z})^T B^T K B\,\phi(\mathbf{z}).$$

By observing that $B^T K B = U(U^T K U)^{-1}U^T K U(U^T K U)^{-1}U^T = B$, we have

$$w(\mathbf{z}) - w(\overline{\mathbf{z}}) = \frac{1}{2}\phi(\mathbf{z})^T B\,\phi(\mathbf{z}) \le \frac{1}{2}\|B\|\|\phi(\mathbf{z})\|^2,$$

which completes the proof. □

Now we are ready to prove Theorem 3.3.11.

Proof of Theorem 3.3.11. Let $\mathbf{y} = \mathbf{x} - \|\mathcal{H}\|^{-1}\boldsymbol{\beta}(\mathbf{x})$. By Lemma 3.3.10 and observing that $\|\cdot\| \ge \|\cdot\|_\infty$ and $\boldsymbol{\beta}(\mathbf{x}) = VV^T\boldsymbol{\beta}(\mathbf{x})$, because $\boldsymbol{\beta}(\mathbf{x}) \in \{A\}^\perp$, we get

$$\begin{aligned} f(\mathbf{y}) - f(\mathbf{x}) &= \frac{1}{2}\|\mathcal{H}\|^{-2}\boldsymbol{\beta}(\mathbf{x})^T H\boldsymbol{\beta}(\mathbf{x}) - \|\mathcal{H}\|^{-1}\mathbf{g}^T\boldsymbol{\beta}(\mathbf{x}) \\ &= \frac{1}{2}\|\mathcal{H}\|^{-2}\boldsymbol{\beta}(\mathbf{x})^T V\,\mathcal{H}\,V^T\boldsymbol{\beta}(\mathbf{x}) - \|\mathcal{H}\|^{-1}\|\boldsymbol{\beta}(\mathbf{x})\|^2 \\ &\le \frac{1}{2}\|\mathcal{H}\|^{-1}\|V^T\boldsymbol{\beta}(\mathbf{x})\|^2 - \|\mathcal{H}\|^{-1}\|\boldsymbol{\beta}(\mathbf{x})\|^2 = -\frac{1}{2}\|\mathcal{H}\|^{-1}\|\boldsymbol{\beta}(\mathbf{x})\|^2 \\ &< -\frac{1}{2}\|\mathcal{H}\|^{-1}\kappa(\mathcal{H})\|\boldsymbol{\varphi}(\mathbf{x})\|^2 = -\frac{1}{2}\|\mathcal{H}^{-1}\|\|\boldsymbol{\varphi}(\mathbf{x})\|^2. \end{aligned} \tag{3.45}$$

The point $\overline{\mathbf{x}}$ satisfies the KKT conditions of problem (3.38),

$$\begin{gathered} \overline{\mathbf{g}} = \sum_{i\in\mathcal{A}(\mathbf{x})}\eta_i\mathbf{e}_i + A^T\boldsymbol{\gamma}, \\ A\,\overline{\mathbf{x}} = \mathbf{b}, \quad \overline{x}_i = x_i \;\; \forall\, i \in \mathcal{A}(\mathbf{x}), \end{gathered} \tag{3.46}$$

where η_i and $\boldsymbol{\gamma}$ are the Lagrange multipliers, and hence

$$\overline{\mathbf{g}}^T(\mathbf{x} - \overline{\mathbf{x}}) = \left(\sum_{i\in\mathcal{A}}\eta_i\mathbf{e}_i + A^T\boldsymbol{\gamma}\right)^T(\mathbf{x} - \overline{\mathbf{x}}) = 0, \tag{3.47}$$

$$\overline{\mathbf{g}}_{\mathcal{F}} = \left[A^T\boldsymbol{\gamma}\right]_{\mathcal{F}}, \tag{3.48}$$

where $\mathcal{A} = \mathcal{A}(\mathbf{x})$ and $\mathcal{F} = \mathcal{F}(\mathbf{x})$. It follows that

$$f(\mathbf{x}) - f(\overline{\mathbf{x}}) = \frac{1}{2}(\mathbf{x}-\overline{\mathbf{x}})^T H(\mathbf{x}-\overline{\mathbf{x}}) + \overline{\mathbf{g}}^T(\mathbf{x}-\overline{\mathbf{x}}) = \frac{1}{2}(\mathbf{x}-\overline{\mathbf{x}})_{\mathcal{F}}^T H_{\mathcal{F}\mathcal{F}}(\mathbf{x}-\overline{\mathbf{x}})_{\mathcal{F}}. \quad (3.49)$$

Recall that a feasible point $\mathbf{v}$ for problem (3.38) satisfies $\mathbf{v}_{\mathcal{A}} = \mathbf{x}_{\mathcal{A}}$ and $A\,\mathbf{v} = \mathbf{b}$. In particular the second condition can be rewritten as

$$A_{\star\mathcal{F}}\mathbf{v}_{\mathcal{F}} = \mathbf{b} - A_{\star\mathcal{A}}\mathbf{x}_{\mathcal{A}}. \quad (3.50)$$

If $\mathcal{F}^* \subseteq \mathcal{F}$, Assumption 3.1.2 ensures that the matrix $A_{\star\mathcal{F}}$ has full rank. Otherwise it is always possible to choose, as in (3.7), the matrix $\widetilde{A} = [A]_{\mathcal{J}\mathcal{F}}$ with full row-rank and, by defining the vector $\widetilde{\mathbf{b}} = \mathbf{b}_{\mathcal{J}} - [A]_{\mathcal{J}\mathcal{A}}\mathbf{x}_{\mathcal{A}}$, replace (3.50) with the equivalent full-rank system $\widetilde{A}\,\mathbf{v}_{\mathcal{F}} = \widetilde{\mathbf{b}}$.

By applying Lemma 3.3.12 with $\mathbf{z} = \mathbf{x}_{\mathcal{F}}$, $K = H_{\mathcal{F}\mathcal{F}}$, $\mathbf{p} = \mathbf{c}_{\mathcal{F}} - H_{\mathcal{F}\mathcal{A}}\,\mathbf{x}_{\mathcal{A}}$, $R = \widetilde{A}$, $\mathbf{q} = \widetilde{\mathbf{b}}$, $\Theta_0 = \{A_{\star\mathcal{F}}\}^{\perp} \equiv \{\widetilde{A}\}^{\perp}$, and $w(\mathbf{z})$ defined as in (3.39), we obtain

$$w(\mathbf{x}_{\mathcal{F}}) - w(\overline{\mathbf{x}}_{\mathcal{F}}) \leq \frac{1}{2}\|B\|\,\|P_{\Theta_0}\nabla w(\mathbf{x}_{\mathcal{F}})\|^2, \quad (3.51)$$

where $B = W(W^T H_{\mathcal{F}\mathcal{F}} W)^{-1}W^T$ and $W \in \mathbb{R}^{|\mathcal{F}|\times(|\mathcal{F}|-|\mathcal{J}|)}$ has orthonormal columns spanning $\{A_{\star\mathcal{F}}\}^{\perp}$. By (3.12) and (3.20), we have

$$P_{\Theta_0}\nabla w(\mathbf{x}_{\mathcal{F}}) = P_{\{A_{\star\mathcal{F}}\}^{\perp}}(\mathbf{g}_{\mathcal{F}}) = \boldsymbol{\varphi}_{\mathcal{F}}(\mathbf{x}),$$

therefore, from (3.49) and (3.51), we get

$$f(\mathbf{x}) - f(\overline{\mathbf{x}}) \leq \frac{1}{2}\|B\|\,\|\boldsymbol{\varphi}_{\mathcal{F}}(\mathbf{x})\|^2. \quad (3.52)$$

We note that

$$\|B\| \leq \|(W^T H_{\mathcal{F}\mathcal{F}} W)^{-1}\| = \zeta_{max}\left((W^T H_{\mathcal{F}\mathcal{F}} W)^{-1}\right) = \frac{1}{\zeta_{min}(W^T H_{\mathcal{F}\mathcal{F}} W)}; \quad (3.53)$$

furthermore,

$$\begin{aligned}
\zeta_{min}(W^T H_{\mathcal{F}\mathcal{F}} W) &= \min_{\substack{\mathbf{s}\in\mathbb{R}^{|\mathcal{F}|-|\mathcal{J}|}\\ \mathbf{s}\neq 0}} \frac{\mathbf{s}^T W^T H_{\mathcal{F}\mathcal{F}} W\mathbf{s}}{\mathbf{s}^T\mathbf{s}} = \min_{\substack{\mathbf{w}\in\mathbb{R}^{|\mathcal{F}|},\ \mathbf{w}\neq 0\\ \mathbf{w}\in\{A_{\star\mathcal{F}}\}^{\perp}}} \frac{\mathbf{w}^T H_{\mathcal{F}\mathcal{F}}\mathbf{w}}{\mathbf{w}^T\mathbf{w}} \\
&= \min_{\substack{\mathbf{v}\in\mathbb{R}^n,\ \mathbf{v}\neq 0\\ \mathbf{v}_{\mathcal{F}}\in\{A_{\star\mathcal{F}}\}^{\perp},\ \mathbf{v}_{\mathcal{A}}=0}} \frac{\mathbf{v}^T H\mathbf{v}}{\mathbf{v}^T\mathbf{v}} \geq \min_{\substack{\mathbf{v}\in\mathbb{R}^n,\ \mathbf{v}\neq 0\\ \mathbf{v}\in A^{\perp}}} \frac{\mathbf{v}^T H\mathbf{v}}{\mathbf{v}^T\mathbf{v}} \qquad (3.54)\\
&= \min_{\substack{\mathbf{u}\in\mathbb{R}^{n-m}\\ \mathbf{u}\neq 0}} \frac{\mathbf{u}^T V^T H V\mathbf{u}}{\mathbf{u}^T V^T V\mathbf{u}} = \zeta_{min}(\mathcal{H}).
\end{aligned}$$

The last inequality, together with (3.52) and (3.53), yields

$$f(\mathbf{x}) - f(\overline{\mathbf{x}}) \leq \frac{1}{2}\frac{1}{\zeta_{min}(\mathcal{H})}\|\boldsymbol{\varphi}_{\mathcal{F}}(\mathbf{x})\|^2 = \frac{1}{2}\|\mathcal{H}^{-1}\|\,\|\boldsymbol{\varphi}(\mathbf{x})\|^2. \quad (3.55)$$

Then, by (3.45) and (3.55), we get

$$f(\mathbf{y}) - f(\overline{\mathbf{x}}) = f(\mathbf{y}) - f(\mathbf{x}) + f(\mathbf{x}) - f(\overline{\mathbf{x}}) < 0. \tag{3.56}$$

By using (3.56) we get

$$0 > f(\mathbf{y}) - f(\overline{\mathbf{x}}) = \overline{\mathbf{g}}^T (\mathbf{y} - \overline{\mathbf{x}}) + \frac{1}{2}(\mathbf{y} - \overline{\mathbf{x}})^T H (\mathbf{y} - \overline{\mathbf{x}}) > \overline{\mathbf{g}}^T (\mathbf{y} - \overline{\mathbf{x}}).$$

Because of the definition of $\mathbf{y}$ and (3.47), we have

$$\overline{\mathbf{g}}^T (\mathbf{y} - \overline{\mathbf{x}}) = \overline{\mathbf{g}}^T (\mathbf{y} - \mathbf{x}) + \overline{\mathbf{g}}^T (\mathbf{x} - \overline{\mathbf{x}}) = \overline{\mathbf{g}}^T (\mathbf{y} - \mathbf{x}) = -\|\mathcal{H}^{-1}\| \, \overline{\mathbf{g}}^T \boldsymbol{\beta}(\mathbf{x}),$$

and thus

$$\overline{\mathbf{g}}^T \boldsymbol{\beta}(\mathbf{x}) > 0. \tag{3.57}$$

For the remainder of the proof we assume that $\overline{\mathbf{x}} \in \Omega$ and we set $\overline{\mathcal{F}} \equiv \mathcal{F}(\overline{\mathbf{x}})$. From (3.48) and $\overline{\mathcal{F}} \subseteq \mathcal{F}$ it follows that $\overline{\mathbf{g}}_{\overline{\mathcal{F}}} = [A^T \boldsymbol{\gamma}]_{\overline{\mathcal{F}}}$, moreover, by Lemma 3.3.2 we have that

$$\boldsymbol{\varphi}(\overline{\mathbf{x}}) = \mathbf{0}. \tag{3.58}$$

By contradiction, suppose that $\boldsymbol{\beta}(\overline{\mathbf{x}}) = 0$. Since $\overline{\mathbf{x}} \in \Omega$, from (3.58) it follows that $\overline{\mathbf{x}}$ is the optimal solution of problem (1.1), and thus $-\overline{\mathbf{g}} \in T_\Omega(\overline{\mathbf{x}})^\circ$. We consider two cases.

(a) $\mathcal{A}(\mathbf{x}) = \mathcal{A}(\overline{\mathbf{x}})$. In this case $T_\Omega(\overline{\mathbf{x}})^\circ = T_\Omega(\mathbf{x})^\circ$, and, since $-\boldsymbol{\beta}(\mathbf{x}) \in T_\Omega(\mathbf{x})$ and $-\overline{\mathbf{g}} \in T_\Omega(\mathbf{x})^\circ$, it is $-\overline{\mathbf{g}}^T (-\boldsymbol{\beta}(\mathbf{x})) = \overline{\mathbf{g}}^T \boldsymbol{\beta}(\mathbf{x}) \leq 0$. This contradicts (3.57).

(b) $\mathcal{A}(\mathbf{x}) \subsetneq \mathcal{A}(\overline{\mathbf{x}})$. In this case the optimality of $\overline{\mathbf{x}}$ for problem (1.1) yield

$$\overline{\mathbf{g}} = \sum_{i \in \mathcal{A}(\overline{\mathbf{x}})} \lambda_i \mathbf{e}_i + A^T \boldsymbol{\nu}, \quad \lambda_i \geq 0 \ \text{ if } \ i \in \mathcal{A}_l(\overline{\mathbf{x}}), \quad \lambda_i \leq 0 \ \text{ if } \ i \in \mathcal{A}_u(\overline{\mathbf{x}}). \tag{3.59}$$

Since $\mathcal{F}(\overline{\mathbf{x}}) \subsetneq \mathcal{F}(\mathbf{x})$, by comparing (3.46) and (3.59) we find that

$$\overline{\mathbf{g}}_{\overline{\mathcal{F}}} = [A^T \boldsymbol{\gamma}]_{\overline{\mathcal{F}}} = [A^T \boldsymbol{\nu}]_{\overline{\mathcal{F}}},$$

and hence

$$A^T_{\star\overline{\mathcal{F}}}(\boldsymbol{\gamma} - \boldsymbol{\nu}) = \mathbf{0}.$$

Since Assumption 3.1.2 holds, matrix $A_{\star\overline{\mathcal{F}}}$ is such that $\operatorname{rank}(A_{\star\overline{\mathcal{F}}}) = m$, therefore $\mathcal{N}(A^T_{\star\overline{\mathcal{F}}}) = \{\mathbf{0}\}$ and $\boldsymbol{\gamma} = \boldsymbol{\nu}$. This implies that $\eta_i = \lambda_i$ for $i \in \mathcal{A}(\mathbf{x})$, whereas $\lambda_i = 0$ for $i \in \mathcal{A}(\overline{\mathbf{x}}) \setminus \mathcal{A}(\mathbf{x})$. Hence we can write

$$\overline{\mathbf{g}} = \sum_{i \in \mathcal{A}(\mathbf{x})} \lambda_i \mathbf{e}_i + A^T \boldsymbol{\nu}, \quad \lambda_i \geq 0 \ \text{ if } \ i \in \mathcal{A}_l(\mathbf{x}), \quad \lambda_i \leq 0 \ \text{ if } \ i \in \mathcal{A}_u(\mathbf{x}),$$

i.e. by (2.15), $-\overline{\mathbf{g}} \in T_\Omega(\mathbf{x})^\circ$, which leads to a contradiction as in case (a).

□

3.4 A subspace accelerated gradient projection framework for QPs

Let's briefly recall the gradient projection method introduced by Calamai and Moré in [27]. Given the current iterate $\mathbf{x}^k$, the next one is obtained as

$$\mathbf{x}^{k+1} = P_\Omega(\mathbf{x}^k - \alpha^k \mathbf{g}^k),$$

where P_Ω is the orthogonal projection onto Ω, and α^k satisfies the following sufficient decrease condition: given $\gamma_1, \gamma_2, \gamma_3 > 0$ and $\mu_1, \mu_2 \in (0,1)$,

$$f^{k+1} \leq f^k + \mu_1 (\mathbf{g}^k)^T (\mathbf{x}^{k+1} - \mathbf{x}^k), \tag{3.60}$$

where

$$\begin{aligned} \alpha^k &\leq \gamma_1, \\ \alpha^k \geq \gamma_2 \quad &\text{or} \quad \alpha^k \geq \gamma_3 \, \overline{\alpha}^k > 0, \end{aligned} \tag{3.61}$$

with $\overline{\alpha}^k$ such that

$$f(\mathbf{x}^k(\overline{\alpha}^k)) > f^k + \mu_2 (\mathbf{g}^k)^T (\mathbf{x}^k(\overline{\alpha}^k) - \mathbf{x}^k), \tag{3.62}$$

where $\mathbf{x}^k(\overline{\alpha}^k) = P_\Omega(\mathbf{x}^k - \overline{\alpha}^k \nabla f(\mathbf{x}^k))$.

In [27, Algorithm 5.3] a very general algorithmic framework is presented, where the previous GP steps are used in selected iterations, alternated with simple decrease steps aimed to speedup the convergence of the overall algorithm.

Algorithm 5.3 in Calamai-Moré [27]

Let $x_0 \in \Omega$ be given. For $k \geqslant 0$ choose x^{k+1} by either (a) or (b):

(a) $\mathbf{x}^{k+1} = P_\Omega(\mathbf{x}^k - \alpha^k \mathbf{g}^k)$ where α^k is such that the (3.60)-(3.62) are satisfied.

(b) Choose $\mathbf{x}^{k+1} \in \Omega$ such that

$$f(\mathbf{x}^{k+1}) \leqslant f(\mathbf{x}^k) \text{ and } \mathcal{A}(\mathbf{x}^k) \subseteq \mathcal{A}(\mathbf{x}^{k+1}).$$

The role of GP steps is to identify promising active sets, i.e. active variables that are likely to be active at the solution too. Once a suitable active set has been fixed at a certain iterate $\mathbf{x}^k$, a reduced problem is defined on the complementary set of free variables

$$\begin{aligned} \min \quad & f(\mathbf{x}), \\ \text{s.t.} \quad & \mathbf{x} \in \Omega(\mathbf{x}^k). \end{aligned} \tag{3.63}$$

Starting from this idea we propose a general framework for the solution of QP problems in the form (1.1) which we will call *Proportionality-based Subspace Accelerated framework for Quadratic Programming* (PSAQP). The framework is outlined in Algorithm 3.1. For the sake of brevity, $\boldsymbol{\varphi}(\mathbf{x}^k)$ and $\boldsymbol{\beta}(\mathbf{x}^k)$ are denoted by $\boldsymbol{\varphi}^k$ and $\boldsymbol{\beta}^k$, respectively. The idea is to alternate identification phases, where GP steps are performed that satisfy (3.60)-(3.62), and minimization phases, where an approximate solution to (3.63) is searched, with $\mathbf{x}^k$ inherited from the last identification phase. Unless a point satisfying

$$\|\boldsymbol{\varphi}^k + \boldsymbol{\beta}^k\| \leq tol \tag{3.64}$$

is found, the identification phase proceeds either until a promising active set $\mathcal{A}^{k+1}$ is identified (i.e. an active set that remains fixed in two consecutive iterations) or no reasonable progress is made in reducing the objective function, i.e.,

$$f^k - f^{k+1} \leq \eta \max_{m \leq l < k} (f^l - f^{l+1}), \tag{3.65}$$

where η is a suitable constant and m is the first iteration of the current identification phase. This choice follows that in [109]. In the minimization phase, an approximate solution to the reduced problem obtained by fixing the variables with indices in the current active set is searched for. The proportionality criterion (3.37) is used to decide when the minimization phase has to be terminated. Like the identification, the minimization phase is abandoned if a suitable approximation to a stationary point is computed.

Algorithm 3.1 PSAQP (Proportionality-based Subspace Accelerated framework for Quadratic Programming)

$x_0 \in \Omega$; $tol \geq 0$; $\eta \in (0,1)$; $\Gamma > 0$; $k = 0$;
$conv = (\left\| \boldsymbol{\varphi}^k + \boldsymbol{\beta}^k \right\| \leq tol)$; $phase1 = .true.$; $phase2 = .true.$
while $(\neg conv)$ **do** ▷ Main loop
  $m = k$;
  while $(phase1)$ **do** ▷ Identification Phase
    $\mathbf{x}^{k+1} = P_{\Omega}(\mathbf{x}^k - \alpha^k \mathbf{g}^k)$ with α^k such that (3.60)-(3.62) hold;
    $conv = (\left\| \boldsymbol{\varphi}^{k+1} + \boldsymbol{\beta}^{k+1} \right\| \leq tol)$;
    $phase1 = (\mathcal{A}^{k+1} \neq \mathcal{A}^k) \wedge (f^k - f^{k+1} > \eta \max_{m \leq l < k} (f^l - f^{l+1})) \wedge (\neg conv)$;
    $k = k + 1$;
  end while
  if $(conv)$ **then**
    $phase2 = .false.$;
  end if
  while $(phase2)$ **do** ▷ Minimization Phase
    Compute an approx. solution $\mathbf{d}^k$ to $\min\{f(\mathbf{x}^k + \mathbf{d})$ s.t. $A\,\mathbf{d} = 0$, $d_i = 0$ if $i \in \mathcal{A}^k\}$;
    $\mathbf{x}^{k+1} = P_{\Omega^k}(\mathbf{x}^k + \alpha^k \mathbf{d}^k)$ with α^k such that $f^{k+1} < f^k$ and $\Omega^k = \Omega \cap \Omega(\mathbf{x}^k)$
    $conv = (\left\| \boldsymbol{\varphi}^{k+1} + \boldsymbol{\beta}^{k+1} \right\| \leq tol)$;
    $phase2 = (\|\boldsymbol{\beta}^{k+1}\|_\infty \leq \Gamma \, \|\boldsymbol{\varphi}^{k+1}\|_2) \wedge (\neg conv)$;
    $k = k + 1$;
  end while
  $phase1 = .true.$; $phase2 = .true.$;
end while
return $\mathbf{x}^k$

We note that, thanks to the projection onto Ω^k, the minimization phase can add variables to the active set, but cannot remove them, and thus PSAQP fits into the general framework of Algorithm 5.3 in [27]. Thus, we may exploit general convergence results available for that algorithm. To this end, we introduce the following definition.

Definition 3.4.1. *Let* $\left\{\mathbf{x}^k\right\}$ *be a sequence generated by the PSAQP method applied*

to problem 1.1. The set

$$K_{GP} = \left\{k \in \mathbb{N} \ : \ \mathbf{x}^{k+1} \text{ is generated by step 6 of Algorithm 3.1}\right\}$$

is called set of GP iterations.

The following convergence result holds, which follows from [27, Theorem 5.2].

Theorem 3.4.2. *Let $\left\{\mathbf{x}^k\right\}$ be a sequence generated by applying PSAQP to problem* (1.1)*. Assume that the set of GP iterations, K_{GP}, is infinite. If some subsequence $\left\{\mathbf{x}^k\right\}_{k\in K}$, with $K \subseteq K_{GP}$, is bounded, then*

$$\lim_{k\in K,\, k\to\infty} \left\|\nabla_\Omega f(\mathbf{x}^{k+1})\right\| = 0. \tag{3.66}$$

Moreover, any limit point of $\left\{\mathbf{x}^k\right\}_{k\in K_{GP}}$ is a stationary point for problem (1.1)*.*

The identification property of the GP steps is inherited by the whole sequence generated by PSAQP, as shown by the following Lemma.

Lemma 3.4.3. *Let us assume that problem* (1.1) *is strictly convex with $\mathbf{x}^*$ optimal solution. If $\left\{\mathbf{x}^k\right\}$ is a sequence in Ω generated by PSAQP applied to* (1.1)*, then for all k sufficiently large*

$$\mathcal{A}_N^* \subseteq \mathcal{A}^k \subseteq \mathcal{A}^*$$

where $\mathcal{A}_N^*$ is defined in Theorem 3.2.1.

Proof. Since $f(\mathbf{x})$ is bounded from below and the sequence $\left\{f^k\right\}$ is decreasing, the sequence $\left\{\mathbf{x}^k\right\}$ is bounded, and, because of Theorem 3.4.2, there is a subsequence $\left\{\mathbf{x}^k\right\}_{k\in K^*}$, with $K^* \subseteq K_{GP}$, which converges to $\mathbf{x}^*$. Now we show that the whole sequence $\left\{\mathbf{x}^k\right\}$ converges to $\mathbf{x}^*$. For any $k \in \mathbb{N}$ we have

$$f^k - f^* \leq f(\mathbf{x}^{k^+}) - f^*, \tag{3.67}$$

where $k^+ = \min\{s \in K^* \ : \ s \geq k\}$. Moreover, for the stationarity of x^* we have $(\mathbf{g}^*)^T(\mathbf{x}^k - \mathbf{x}^*) \geq 0$, and then

$$\begin{aligned} f^k - f^* &= (\mathbf{g}^*)^T(\mathbf{x}^k - \mathbf{x}^*) + \frac{1}{2}(\mathbf{x}^k - \mathbf{x}^*)^T H(\mathbf{x}^k - \mathbf{x}^*) \\ &\geq \frac{1}{2}(\mathbf{x}^k - \mathbf{x}^*)^T V\mathcal{H}V^T(\mathbf{x}^k - \mathbf{x}^*) \geq \zeta_{min}(\mathcal{H})\,\|\mathbf{x}^k - \mathbf{x}^*\|^2, \end{aligned} \tag{3.68}$$

where $\mathcal{H}$ and V are defined in Theorem 3.3.11 and the equality

$$\mathbf{x}^k - \mathbf{x}^* = VV^T(\mathbf{x}^k - \mathbf{x}^*)$$

has been exploited. From (3.67) and (3.68) it follows that $\left\{\mathbf{x}^k\right\}$ converges to $\mathbf{x}^*$. Then, for k sufficiently large, $\mathcal{F}^* \subseteq \mathcal{F}^k$ and hence $\mathcal{A}^k \subseteq \mathcal{A}^*$. Furthermore, by Theorem 3.2.1, the convergence of $\left\{\mathbf{x}^k\right\}_{k\in K_{GP}}$ to $\mathbf{x}^*$, together with (3.66), yields $\mathcal{A}_N^* \subseteq \mathcal{A}(\mathbf{x}^k)$ for all $k \in K_{GP}$ sufficiently large. Since minimization steps do not remove variables from the active set, we have $\mathcal{A}_N^* \subseteq \mathcal{A}(\mathbf{x}^k)$ for all k sufficiently large. □

We note that in case of non-degeneracy ($\mathcal{A}_N^* = \mathcal{A}^*$) the active set eventually settles down, i.e. the identification property holds. This implies that the the solution of (1.1) reduces to the solution of an unconstrained problem in a finite number of iterations, which is the key ingredient to prove finite convergence of methods that fit into the framework of Algorithm 5.3 in [27], such as the proposed PSAQP. In case of degeneracy we can just say that the non-degenerate active constraints at the solution will be identified in a finite number of steps. However, in the strictly convex case, finite convergence can be achieved in this case too, provided a suitable value of Γ is taken, as stated by the following theorem which extends Theorem 4.4 in [51].

Theorem 3.4.4. *Let us assume that problem* (1.1) *is strictly convex and* $\mathbf{x}^*$ *is its optimal solution. Let* $\left\{\mathbf{x}^k\right\}$ *be a sequence in* Ω *generated by PSAQP applied to* (1.1)*, in which the minimization phase is performed by any algorithm that is exact for strictly convex quadratic programming. If one of the following conditions holds:*

(i) $\mathbf{x}^*$ *is non-degenerate,*

(ii) $\mathbf{x}^*$ *is degenerate and* $\Gamma \geq \kappa(\mathcal{H})^{1/2}$*, where* $\mathcal{H}$ *is defined in Theorem 3.3.11,*

then $\mathbf{x}^k = \mathbf{x}^*$ *for* k *sufficiently large.*

Proof. (i) By Lemma 3.4.3, in case of non-degeneracy $\mathcal{A}^k = \mathcal{A}^*$ for k sufficiently large, and the thesis trivially holds.

(ii) Thanks to Lemma 3.4.3, we have that P2GP is able to identify the active non-degenerate variables and the free variables at the solution for k sufficiently large. This means that there exists $\overline{k}$ such that for $k \geq \overline{k}$ the solution $\mathbf{x}^*$ of (1.1) is also solution of

$$\begin{array}{ll} \min & f(\mathbf{x}), \\ \text{s.t.} & \mathbf{x} \in \Omega(\mathbf{x}^k). \end{array} \tag{3.69}$$

Now assume that $\Gamma \geq \kappa(\mathcal{H})^{1/2}$ and suppose by contradiction that there exists $\widehat{k} \geq \overline{k}$ such that

$$\left\|\boldsymbol{\beta}(\mathbf{x}^{\widehat{k}})\right\|_\infty > \Gamma \left\|\boldsymbol{\varphi}(\mathbf{x}^{\widehat{k}})\right\|_2.$$

Then, by Theorem 3.3.11 it is $\boldsymbol{\beta}(\widehat{\mathbf{x}}) \neq 0$, where $\widehat{\mathbf{x}}$ is the solution of (3.69) with $k = \widehat{k}$. Since $\widehat{\mathbf{x}} = \mathbf{x}^*$, this contradicts the optimality of $\mathbf{x}^*$. Therefore, $\mathbf{x}^k$ is a proportional iterate for $k \geq \widehat{k}$ and PSAQP will use the algorithm of the minimization phase to determine the next iterate. Two cases are possible:

(a) $\mathbf{x}^{k+1} = \mathbf{x}^*$, therefore the thesis holds;

(b) $\mathbf{x}^{k+1} \neq \mathbf{x}^*$ is proportional and such that $\mathcal{A}(\mathbf{x}^k) \subsetneq \mathcal{A}(\mathbf{x}^{k+1})$, therefore $\mathbf{x}^{k+2}$ will be computed using again the algorithm of the minimization phase. Since the active sets are nested, either PSAQP is able to find $\mathcal{A}^*$ in a finite number of iterations or at a certain iteration it falls in case (a), and hence the thesis is proved.

□

3.4.1 Implementation issues

We have introduced an active-set framework for the solution of QPs of the form (1.1), and proved its finite convergence in the solution of possibly degenerate strictly convex problems. In all the other cases, provided the set K_{GP} is not finite and the objective function is bounded, Theorem 3.4.2 ensures that the algorithm is still able to converge to a stationary point of problem (1.1).

There are different aspects to care about in the implementation of PSAQP. First of all we need to compute the projections onto Ω and those onto $T_\Omega(\mathbf{x})$ needed to evaluate the projected gradient and its components. The efficiency of this operation is crucial for the efficiency of the overall algorithm, even if the number of projections can be lowered, e.g., by using, both in the identification and in the minimization phase, a line search along the feasible direction instead of along the projection arc (see Section 2.3). As we already mentioned in Section 2.5.2, in the case of sparse constraints an efficient algorithm has been recently proposed [91], which may be exploited in an implementation of PSAQP.

Another issue is the solution of the unconstrained subproblems (3.38), which can be written as

$$\begin{array}{rl} \min & \frac{1}{2}\mathbf{x}^T H\,\mathbf{x} - \mathbf{c}^T\mathbf{x}, \\ \text{s.t.} & A\,\mathbf{x} = \mathbf{b}, \\ & x_i = l_i, \quad i \in \mathcal{A}_l, \\ & x_i = u_i, \quad i \in \mathcal{A}_u. \end{array} \tag{3.70}$$

Observe that a solution to (3.70) can be found by solving the problem

$$\begin{array}{rl} \min\limits_{\mathbf{v}\in\mathbb{R}^{|\mathcal{F}|}} & \frac{1}{2}\mathbf{v}^T G\,\mathbf{v} - \mathbf{q}^T\mathbf{v}, \\ \text{s.t.} & B\,\mathbf{v} = \mathbf{d}, \end{array} \tag{3.71}$$

where

$$G = H_{\mathcal{F}\mathcal{F}}, \quad \mathbf{q} = \mathbf{c}_{\mathcal{F}} - H_{\mathcal{F}\mathcal{A}_l}\mathbf{l}_{\mathcal{A}_l} - H_{\mathcal{F}\mathcal{A}_u}\mathbf{u}_{\mathcal{A}_u}, \quad B = A_{\star\mathcal{F}}, \quad \text{and} \quad \mathbf{d} = \mathbf{b} - A_{\star\mathcal{A}_l}\mathbf{l}_{\mathcal{A}_l} - A_{\star\mathcal{A}_u}\mathbf{u}_{\mathcal{A}_u}.$$

A stationary point for problem (3.71) is a solution of the system

$$\left\{ \begin{array}{lcl} -G\mathbf{v} + \mathbf{q} + B^T\boldsymbol{\theta} & = & \mathbf{0}, \\ B\,\mathbf{v} & = & \mathbf{d}, \end{array} \right.$$

i.e. a solution of the *saddle point system*

$$\begin{pmatrix} -G & B^T \\ B & 0 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{v} \\ \boldsymbol{\theta} \end{pmatrix} = \begin{pmatrix} -\mathbf{q} \\ \mathbf{d} \end{pmatrix}, \tag{3.72}$$

which can be found, e.g., by preconditioned conjugate gradient methods [83, 8, 42].

Finally, since far from the solution the matrix $A_{\star\mathcal{F}}$ is likely to be row-rank deficient, especially in the case of a large number of constraints, we will need to address the problem of choosing the set $\mathcal{J}$ of independent rows of $A_{\star\mathcal{F}}$ for the computation of $\boldsymbol{\xi}$, as defined in (3.7).

A specialization of PSAQP, called P2GP, is available for the case of BQPs and SLBQPs and will be described in the following chapter. The implementation of a method for general QPs, fitting in the PSAQP framework, is currently under study.

Chapter 4

The Proportionality-based 2-phase Gradient Projection method

Here we are concerned with the solution of SLBQPs, i.e. QPs of the form

$$\begin{aligned} \min \quad & f(\mathbf{x}) := \frac{1}{2}\mathbf{x}^T H\,\mathbf{x} - \mathbf{c}^T\mathbf{x}, \\ \text{s.t.} \quad & \mathbf{a}^T\mathbf{x} = b, \\ & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \end{aligned} \tag{4.1}$$

where $H \in \mathbb{R}^{n\times n}$ is symmetric, $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{a} \in \mathbb{R}^n$, $b \in \mathbb{R}$, $\mathbf{l} \in \{\mathbb{R} \cup \{-\infty\}\}^n$, $\mathbf{u} \in \{\mathbb{R} \cup \{+\infty\}\}^n$, $\mathbf{l} \leq \mathbf{u}$. We propose a two-phase GP method, called *Proportionality-based 2-phase Gradient Projection* (P2GP). P2GP, which is able to deal both with SLBQPs and BQPs, can be considered as a specialization of the PSAQP framework proposed in Chapter 3 and as a generalization of the GPCG method by Moré and Toraldo [109] to a wider class of problems. Besides targeting problems more general than strictly convex BQPs, the new method differs from GPCG because it follows a different approach in deciding when to terminate optimization in the reduced space. Whereas GPCG uses a heuristic based on the bindingness of the active variables, P2GP relies on the comparison between a measure of optimality within the reduced space and a measure of bindingness of the variables that are on the bounds. This approach exploits the concept of *proportional iterate*, discussed in the previous chapters. To this end, we specialize the definition of free gradient, chopped gradient, and proportional iterates to the case of problem (4.1). As in GPCG, and unlike other algorithms for BQPs sharing a common ground (e.g.,[53, 64, 65, 107]), the task of adjusting the active set is left only to the GP steps; thus, for strictly convex BQPs our algorithm differs from GPCG in the criterion used to stop minimization of the reduced problem. This change makes a significant difference in the effectiveness of the algorithm as our numerical experiments show. In addition, the application of the proportionality concept allows to state finite convergence for strictly convex problems also for dual-degenerate solutions. In the case of non-convex problems and convex problems with semidefinite Hessian, if the objective function is bounded, the algorithm converges to a stationary point as a result of suitable application of the

GP method in the identification phase. Finally, if the problem is unbounded, the method is able to detect the unboundedness and interrupt the computation.

4.1 Stationarity results for SLBQPs

A point $\mathbf{x}^* \in \Omega$ is a stationary point for problem (4.1) if and only if there exist Lagrange multipliers $\rho^*, \lambda_i^* \in \mathbb{R}$, with $i \in \mathcal{A}^*$, such that

$$\mathbf{g}^* = \sum_{i \in \mathcal{A}^*} \lambda_i^* \mathbf{e}_i + \rho^* \mathbf{a}, \quad \lambda_i^* \geq 0 \ \text{ if } \ i \in \mathcal{A}_l^*, \quad \lambda_i^* \leq 0 \ \text{ if } \ i \in \mathcal{A}_u^*, \tag{4.2}$$

or, equivalently,

$$\mathbf{g}_{\mathcal{F}^*}^* - \rho^* \mathbf{a}_{\mathcal{F}^*} = \mathbf{0}, \tag{4.3}$$

$$\lambda_i^* = g_i^* - \rho^* q_i \geq 0 \ \text{ if } \ i \in \mathcal{A}_l^*, \tag{4.4}$$

$$\lambda_i^* = g_i^* - \rho^* q_i \leq 0 \ \text{ if } \ i \in \mathcal{A}_u^*. \tag{4.5}$$

If $\mathbf{a}_{\mathcal{F}^*} \neq \mathbf{0}$, by taking the scalar product of (4.3) with $\mathbf{a}_{\mathcal{F}^*}$, we obtain

$$\rho^* = \frac{\mathbf{a}_{\mathcal{F}^*}^T \mathbf{g}_{\mathcal{F}^*}^*}{\mathbf{a}_{\mathcal{F}^*}^T \mathbf{a}_{\mathcal{F}^*}}$$

(with a little abuse of notation we include $\mathcal{F}^* = \emptyset$ in the case $\mathbf{a}_{\mathcal{F}^*} = \mathbf{0}$). Then, by defining for all $\mathbf{x} \in \Omega$

$$\rho(\mathbf{x}) := \begin{cases} 0, & \text{if } \ \mathbf{a}_{\mathcal{F}} = \mathbf{0}, \\ \dfrac{\mathbf{a}_{\mathcal{F}}^T \mathbf{g}_{\mathcal{F}}}{\mathbf{a}_{\mathcal{F}}^T \mathbf{a}_{\mathcal{F}}}, & \text{otherwise}, \end{cases} \tag{4.6}$$

where $\mathcal{F} = \mathcal{F}(\mathbf{x})$, and

$$\mathbf{h}(\mathbf{x}) := \nabla f(\mathbf{x}) - \rho(\mathbf{x})\mathbf{a}, \tag{4.7}$$

conditions (4.3)-(4.5) can be expressed as

$$h_i^* = 0 \ \text{ if } i \in \mathcal{F}^*, \quad h_i^* \geq 0 \ \text{ if } i \in \mathcal{A}_l^*, \quad h_i^* \leq 0 \ \text{ if } i \in \mathcal{A}_u^*. \tag{4.8}$$

As for the case of general QPs, we can introduce the definition of binding set based on $\mathbf{h}(\mathbf{x})$.

Definition 4.1.1. *Let* $\mathbf{x} \in \Omega$. *The* binding set *at* $\mathbf{x}$ *is defined as*

$$\mathcal{B}(\mathbf{x}) := \{i \,:\, (i \in \mathcal{A}_l(\mathbf{x}) \,\wedge\, h_i(\mathbf{x}) \geq 0) \ \vee \ (i \in \mathcal{A}_u(\mathbf{x}) \,\wedge\, h_i(\mathbf{x}) \leq 0)\}. \tag{4.9}$$

We can also specialize the estimate of the Lagrange multipliers provided in Theorem 3.1.5.

Theorem 4.1.2. *Assume that* $\left\{\mathbf{x}^k\right\}$ *is a sequence in* Ω *that converges to a non-degenerate stationary point* $\mathbf{x}^*$, *and* $\mathcal{A}(\mathbf{x}^k) = \mathcal{A}(\mathbf{x}^*)$ *for all* k *sufficiently large. Then*

$$\lim_{k \to \infty} \rho(\mathbf{x}^k) = \rho^* \quad \textit{and} \quad \lim_{k \to \infty} \lambda_i(\mathbf{x}^k) = \lambda_i^* \ \ \forall i \in \mathcal{A}^*, \tag{4.10}$$

where $\lambda_i(\mathbf{x})$ *is defined as follows:*

$$\lambda_i(\mathbf{x}) := \begin{cases} \max\{0, h_i(\mathbf{x})\} & \text{if } i \in \mathcal{A}_l(\mathbf{x}), \\ \min\{0, h_i(\mathbf{x})\} & \text{if } i \in \mathcal{A}_u(\mathbf{x}), \\ 0 & \text{if } i \in \mathcal{F}(\mathbf{x}). \end{cases}$$

As in the previous chapter, we assume that the constraints satisfy the LICQ.

Assumption 4.1.3 (Linear Independence Constraint Qualification - LICQ)**.** *Let* $\mathbf{x}^*$ *be any stationary point of* (4.1)*. The active constraint normals* $\{\mathbf{a}\} \cup \{\mathbf{e}_i : i \in \mathcal{A}^*\}$ *are linearly independent.*

This assumption is always satisfied, for instance, when Ω is the standard simplex; furthermore, it guarantees $\mathbf{a}_{\mathcal{F}^*} \neq \mathbf{0}$.

4.1.1 Proportional iterates for SLBQPs

To measure the violation of the KKT conditions (4.3)-(4.5) and to balance optimality between free and active variables, we provide a specialization of the free and chopped gradient already introduced in Chapter 3 for the general case of problem (1.1).

Definition 4.1.4. *For any* $\mathbf{x} \in \Omega$*, the* free gradient $\boldsymbol{\varphi}(\mathbf{x})$ *is defined as*

$$\boldsymbol{\varphi}_{\mathcal{F}}(\mathbf{x}) = \mathbf{h}_{\mathcal{F}}(\mathbf{x}), \qquad \boldsymbol{\varphi}_{\mathcal{A}}(\mathbf{x}) = \mathbf{0},$$

where $\mathbf{h}(\mathbf{x})$ *is given in* (4.7)*. The* chopped gradient $\boldsymbol{\beta}(\mathbf{x})$ *is defined as*

$$\boldsymbol{\beta}(\mathbf{x}) := -\nabla_\Omega f(\mathbf{x}) - \boldsymbol{\varphi}(\mathbf{x}). \tag{4.11}$$

We note that

$$\boldsymbol{\varphi}_{\mathcal{F}}(\mathbf{x}) = P_{\{\mathbf{a}_{\mathcal{F}}\}^\perp} \left(\nabla f_{\mathcal{F}}(\mathbf{x}) \right), \tag{4.12}$$

where $\mathcal{F} = \mathcal{F}(\mathbf{x})$ and $P_{\{\mathbf{a}_{\mathcal{F}}\}^\perp} \in \mathbb{R}^{|\mathcal{F}| \times |\mathcal{F}|}$ is the orthogonal projection onto the subspace of $\mathbb{R}^{|\mathcal{F}|}$ orthogonal to $\mathbf{a}_{\mathcal{F}}$,

$$P_{\{\mathbf{a}_{\mathcal{F}}\}^\perp} = I - \frac{\mathbf{a}_{\mathcal{F}} \, \mathbf{a}_{\mathcal{F}}^T}{\mathbf{a}_{\mathcal{F}}^T \, \mathbf{a}_{\mathcal{F}}}.$$

The two vectors satisfy all the properties proved in Chapter 3. In summary, we split the projected gradient $\nabla_\Omega f(\mathbf{x})$ into two parts:

- $-\boldsymbol{\varphi}(\mathbf{x})$, which lies on the affine closure of the face determined by the active variables at $\mathbf{x}$ and gives measure of optimality within the reduced space;
- $-\boldsymbol{\beta}(\mathbf{x})$, is orthogonal to the same face, points towards the interior of the feasible set Ω, and gives a measure of optimality over the complementary space.

An example of the splitting is reported in Figure 4.1 for the case of an SLBQP problem characterized by $n = 3$, $\mathbf{a} = \mathbf{1}$, $b = 1$, $\mathbf{l} = \mathbf{0}$ and $\mathbf{u} = 0.75 \cdot \mathbf{1}$.

It is worth noting that the computation of these quantities is not too expensive, indeed the projected gradient can be computed with one of the projection algorithms

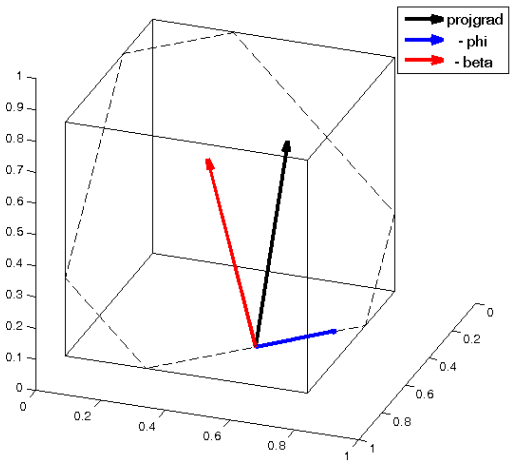

Figure 4.1. Visualization of the splitting of the projected gradient $\nabla_\Omega f(\mathbf{x})$ (here indicated as "projgrad") in the two orthogonal components $-\boldsymbol{\varphi}(\mathbf{x})$ and $-\boldsymbol{\beta}(\mathbf{x})$ in the case of a 3-dimensional SLBQP problem.

described in Section 2.5.1, $\boldsymbol{\varphi}$ can be computed by $\mathcal{O}(n)$ operations, and $\boldsymbol{\beta}$ simply as the difference between the two.

With this decomposition we can apply to problem (4.1) the definition of proportional iterates introduced in the previous chapters as an iterate satisfying

$$\|\boldsymbol{\beta}(\mathbf{x}^k)\|_\infty \leq \Gamma \|\boldsymbol{\varphi}(\mathbf{x}^k)\|. \tag{4.13}$$

In the strictly convex case, disproportionality of $\mathbf{x}^k$ again guarantees that the solution of (4.1) does not belong to the face identified by the active variables at $\mathbf{x}^k$. This result is a consequence of the next theorem, which is a specialization of Theorem 3.3.11.

Theorem 4.1.5. *Let H be the Hessian matrix in (4.1) and let $H_q = V^T H V$ be positive definite, where $V \in \mathbb{R}^{n\times(n-1)}$ has orthonormal columns spanning $\{\mathbf{a}\}^\perp$. Let $\mathbf{x} \in \Omega$ be such that $\|\boldsymbol{\beta}(\mathbf{x})\|_\infty > \kappa(H_q)^{1/2} \|\boldsymbol{\varphi}(\mathbf{x})\|_2$, and let $\overline{\mathbf{x}}$ be the solution of*

$$\begin{aligned} \min \quad & f(\mathbf{u}), \\ \text{s.t.} \quad & \mathbf{u} \in \Omega(\mathbf{x}), \end{aligned} \tag{4.14}$$

where $\Omega(\mathbf{x})$ is defined in (1.2). If $\overline{\mathbf{x}} \in \Omega$, then $\boldsymbol{\beta}(\overline{\mathbf{x}}) \neq 0$.

4.2 The Proportionality-based 2-phase Gradient Projection method

We now introduce the Proportionality-based 2-phase Gradient Projection (P2GP) method for problem (4.1). The method does not assume that (4.1) is strictly convex. However, if (4.1) is not strictly convex, the method only computes an approximation of a stationary point or finds that the problem is unbounded below. If strict convexity holds, P2GP provides an approximation to the optimal solution. The method is outlined in Algorithm 4.1 and explained in detail in the next sections. For the sake of brevity, $\boldsymbol{\varphi}(\mathbf{x}^k)$ and $\boldsymbol{\beta}(\mathbf{x}^k)$ are denoted by $\boldsymbol{\varphi}^k$ and $\boldsymbol{\beta}^k$, respectively. Like PSAQP,

proposed in Chapter 3, it alternates identification phases, consisting in GP steps satisfying the sufficient decrease conditions (3.60)-(3.62), and minimization phases, where an approximate solution to (3.63) is searched, with $\mathbf{x}^k$ inherited from the last identification phase. Unless a point satisfying

$$\|\boldsymbol{\varphi}^k + \boldsymbol{\beta}^k\| \leq tol \tag{4.15}$$

is found, or the problem is discovered to be unbounded below, the identification phase proceeds either until a promising active set $\mathcal{A}^{k+1}$ is identified (i.e. an active set that remains fixed in two consecutive iterations) or no reasonable progress is made in reducing the objective function, i.e.,

$$f^k - f^{k+1} \leq \eta \max_{m \leq l < k} (f^l - f^{l+1}), \tag{4.16}$$

where η is a suitable constant and m is the first iteration of the current identification phase. This choice follows that in [109], described in Chapter 2. In the minimization phase, an approximate solution to the reduced problem obtained by fixing the variables with indices in the current active set is searched for. The proportionality criterion (4.13) is used to decide when the minimization phase has to be terminated; this is a significant difference from the GPCG method, which exploits a condition based on the bindingness of the active variables. Note that the accuracy required in the solution of the reduced problem (3.63) affects the efficiency of the method and a loose stopping criterion must be used, since the control of the minimization phase is actually left to the proportionality criterion (more details are given in Section 4.2.2). Like the identification, the minimization phase is abandoned if a suitable approximation to a stationary point is computed or unboundedness is discovered. Nonpositive curvature directions are exploited as explained in Sections 4.2.1 and 4.2.2.

We note that the minimization phase can add variables to the active set, but cannot remove them, thus P2GP fits into the general framework of two phase algorithm proposed by Calamai and Moré [27, Algorithm 5.3]. The convergence results proved in Chapter 3 are still valid for the case of P2GP applied to problem (4.1). To ease the description of the algorithm we recall the main results: the first one allows us to state the general convergence of the algorithm, the second one states the finite convergence property in the case of strictly convex problems, also in the case of degenerate stationary points.

Theorem 4.2.1. *Let $\left\{\mathbf{x}^k\right\}$ be a sequence generated by applying the P2GP method to problem* (4.1)*. Assume that the set of GP iterations, K_{GP}, is infinite. If some subsequence $\left\{\mathbf{x}^k\right\}_{k \in K}$, with $K \subseteq K_{GP}$, is bounded, then*

$$\lim_{k \in K,\, k \to \infty} \left\| \nabla_\Omega f(\mathbf{x}^{k+1}) \right\| = 0. \tag{4.17}$$

Moreover, any limit point of $\left\{\mathbf{x}^k\right\}_{k \in K_{GP}}$ is a stationary point for problem (4.1)*.*

Theorem 4.2.2. *Let us assume that problem* (4.1) *is strictly convex and $\mathbf{x}^*$ is its optimal solution. Let $\left\{\mathbf{x}^k\right\}$ be a sequence in Ω generated by the P2GP method applied to* (4.1)*, in which the minimization phase is performed by any algorithm that is exact for strictly convex quadratic programming. If one of the following conditions holds:*

Algorithm 4.1 P2GP (Proportionality-based 2-phase Gradient Projection)

$x_0 \in \Omega$; $tol \geq 0$; $\eta \in (0,1)$; $\Gamma > 0$; $k = 0$;
$conv = (\left\| \boldsymbol{\varphi}^k + \boldsymbol{\beta}^k \right\| \leq tol)$; $unbnd = .false.$; $phase1 = .true.$; $phase2 = .true.$
while $(\neg\, conv \wedge \neg\, unbnd)$ **do** ▷ Main loop
  $m = k$;
  while $(phase1)$ **do** ▷ Identification Phase
    if $((\nabla_\Omega f^k)^T H\, (\nabla_\Omega f^k) \leq 0 \ \wedge\ \max\left\{\alpha > 0 \ \text{ s.t. } \mathbf{x}^k + \alpha \nabla_\Omega f^k \in \Omega\right\} = +\infty)$ **then**
      $unbnd = .true.$;
    else
      $\mathbf{x}^{k+1} = P_\Omega(\mathbf{x}^k - \alpha^k \mathbf{g}^k)$ with α^k such that (3.60)-(3.62) hold;
    end if
    if $(\neg\, unbnd)$ **then**
      $conv = (\left\| \boldsymbol{\varphi}^{k+1} + \boldsymbol{\beta}^{k+1} \right\| \leq tol)$;
      $phase1 = (\mathcal{A}^{k+1} \neq \mathcal{A}^k) \wedge (f^k - f^{k+1} > \eta \max\limits_{m \leq l < k} (f^l - f^{l+1})) \wedge (\neg\, conv)$;
      $k = k + 1$;
    end if
  end while
  if $(conv \vee unbnd)$ **then**
    $phase2 = .false.$;
  end if
  while $(phase2)$ **do** ▷ Minimization Phase
    Compute an approximate solution $\mathbf{d}^k$ (see end of Section 4.2.2) to problem

$$\min\left\{f(\mathbf{x}^k + \mathbf{d}) \text{ s.t. } \mathbf{a}^T \mathbf{d} = 0,\, d_i = 0 \text{ if } i \in \mathcal{A}^k\right\};$$

    if $((\mathbf{d}^k)^T H\, \mathbf{d}^k \leq 0)$ **then**
      Compute $\alpha^k = \max\left\{\alpha > 0 \ \text{ s.t. } \mathbf{x}^k + \alpha \mathbf{d}^k \in \Omega\right\}$;
      if $(\alpha = +\infty)$ **then**
        $unbnd = .true.$;
      else
        $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{d}^k$;
        $conv = (\left\| \boldsymbol{\varphi}^{k+1} + \boldsymbol{\beta}^{k+1} \right\| \leq tol)$;
        $k = k + 1$;
      end if
      $phase2 = .false.$;
    else
      $\mathbf{x}^{k+1} = P_{\Omega^k}(\mathbf{x}^k + \alpha^k \mathbf{d}^k)$ with α^k such that $f^{k+1} < f^k$ and $\Omega^k = \Omega \cap \Omega(\mathbf{x}^k)$
      $conv = (\left\| \boldsymbol{\varphi}^{k+1} + \boldsymbol{\beta}^{k+1} \right\| \leq tol)$;
      $phase2 = (\|\boldsymbol{\beta}^{k+1}\|_\infty \leq \Gamma\, \|\boldsymbol{\varphi}^{k+1}\|_2) \wedge (\neg\, conv)$;
      $k = k + 1$;
    end if
  end while
  $phase1 = .true.$; $phase2 = .true.$;
end while
if $(conv)$ **then**
  return $\mathbf{x}^k$
else
  return "problem (4.1) is unbounded";
end if

(i) $\mathbf{x}^*$ *is non-degenerate,*

(ii) $\mathbf{x}^*$ *is degenerate and* $\Gamma \geq \kappa(H_q)^{1/2}$*, where* H_q *is defined in Theorem 3.3.11,*

then $\mathbf{x}^k = \mathbf{x}^*$ *for* k *sufficiently large.*

We will now provide further details on the algorithm.

4.2.1 Identification phase

In the identification phase (Steps 4-16 of Algorithm 4.1), every projected gradient step needs the computation of a steplength α^k satisfying the sufficient decrease condition (3.60)-(3.62). As described in Chapter 2 for the GPCG algorithm, this steplength can be obtained by generating a sequence $\{\alpha_i^k\}$ of positive trial values such that

$$\alpha_0^k \in [\gamma_2, \gamma_1] \tag{4.18}$$

$$\alpha_i^k \in [\gamma_4 \alpha_{i-1}^k, \gamma_5 \alpha_{i-1}^k], \quad i > 0, \tag{4.19}$$

where γ_1 and γ_2 are given in (3.61) and $\gamma_4 < \gamma_5 < 1$, and by setting α^k to the first trial value that satisfies (3.60). Note that in practice γ_2 is a very small value and γ_1 is a very large one; therefore, we assume for simplicity that (4.19) holds for all the choices of α_0^k described next.

Motivated by the results reported in [37] for BQPs, we compute α_0^k by using a BB-like rule. Following recent studies on steplength selection in gradient methods [49, 50], we set α_0^k equal to the $\text{ABB}_{\min}$ steplength proposed in [73] and described in Section 2.1.1.

If $\alpha_0^k > 0$, we build the trial step lengths by using the quadratic interpolation strategy with the safeguard (4.19) described in Section 2.4.1. If $\alpha_0^k \leq 0$, we check if $(\nabla_\Omega f^k)^T H (\nabla_\Omega f^k) \leq 0$, which implies that the problem

$$\begin{array}{ll} \min & f(\mathbf{x}_k + \mathbf{v}), \\ \text{s.t.} & \mathbf{a}^T \mathbf{v} = 0, \\ & v_i = 0, \quad i \in \mathcal{B}^k \end{array}$$

is unbounded below along the direction $\nabla_\Omega f^k$. In this case we compute the breakpoints along $\nabla_\Omega f^k$ as proposed in [108]. For any $\mathbf{x} \in \Omega$ and any direction $\mathbf{p} \in T_\Omega(\mathbf{x})$, the breakpoints ω_i, with $i \in \{j : p_j \neq 0\}$, are given by the following formulas:

$$\text{if } p_i < 0, \text{ then } \omega_i = +\infty \text{ if } l_i = -\infty, \text{ and } \omega_i = \frac{l_i - x_i}{p_i} \text{ otherwise;}$$

$$\text{if } p_i > 0, \text{ then } \omega_i = +\infty \text{ if } u_i = +\infty, \text{ and } \omega_i = \frac{u_i - x_i}{p_i} \text{ otherwise.}$$

If the minimum breakpoint $\omega_{\min}$, which satisfies

$$\omega_{\min} = \max \left\{ \alpha > 0 \text{ s.t. } \mathbf{x}^k - \alpha \nabla_\Omega f^k \in \Omega \right\},$$

is infinite, then problem (4.1) is unbounded. Otherwise, we set $\alpha_0^k = \overline{\omega}$, where $\overline{\omega}$ is the maximum finite breakpoint. If α_0^k does not satisfy the sufficient decrease

condition, we reduce it by backtracking until this condition holds. Finally, if $\alpha_0^k \leq 0$ and $(\nabla_\Omega f^k)^T H (\nabla_\Omega f^k) > 0$, we set

$$\alpha_0^k = -\frac{(\nabla_\Omega f^k)^T \mathbf{g}^k}{(\nabla_\Omega f^k)^T H (\nabla_\Omega f^k)},$$

and proceed by safeguarded quadratic interpolation (see [108] and Section 2.4.1 for further details). In order to simplify the description, in Algorithm 4.1 we have omitted the selection of α_0^k.

Unless an approximation $\mathbf{x}^{k+1}$ to a point satisfying (4.15) is found, or unboundedness of the problem is discovered, the identification phase is left when either the active set does not change in two consecutive iterations or the GP method is not making sufficient progress in reducing the objective function (as described at the beginning of Section 4.2), where the progress is measured as in Step 16. This choice follows that in [109].

4.2.2 Minimization phase

The minimization phase (Steps 20-38 of Algorithm 4.1) requires the approximate solution of

$$\begin{array}{ll} \min & f(\mathbf{x}^k + \mathbf{d}), \\ \text{s.t.} & \mathbf{a}^T\mathbf{d} = 0, \quad d_i = 0 \ \text{ if } i \in \mathcal{A}(\mathbf{x}^k), \end{array} \tag{4.20}$$

which is equivalent to

$$\begin{array}{ll} \min & g(\mathbf{y}) := \dfrac{1}{2}\,\mathbf{y}^T H_{\mathcal{F}\mathcal{F}}\,\mathbf{y} + (\mathbf{g}_{\mathcal{F}}^k)^T\mathbf{y}, \\ \text{s.t.} & \mathbf{a}_{\mathcal{F}}^T\,\mathbf{y} = 0, \quad \mathbf{y} \in \mathbb{R}^s, \end{array} \tag{4.21}$$

where $\mathcal{F} = \mathcal{F}^k$ and $s = |\mathcal{F}|$.

Problem (4.21) can be formulated as an unconstrained quadratic minimization problem by using a Householder transformation

$$P = I - \mathbf{w}\mathbf{w}^T \in \mathbb{R}^{s\times s}, \quad \|\mathbf{w}\| = \sqrt{2}, \quad P\mathbf{a}_{\mathcal{F}} = \sigma\mathbf{e}_1,$$

where $\sigma = \pm\|\mathbf{a}_{\mathcal{F}}\|$ (see, e.g.,[18]). Letting $\mathbf{y} = P\,\mathbf{z}$, $M = PH_{\mathcal{F}\mathcal{F}}P$ and $\mathbf{r} = P\,\mathbf{g}_{\mathcal{F}}^k$, problem (4.21) becomes

$$\begin{array}{ll} \min & p(\mathbf{z}) := \dfrac{1}{2}\,\mathbf{z}^T M\,\mathbf{z} + \mathbf{r}^T\mathbf{z}, \\ \text{s.t.} & z_1 = 0, \end{array}$$

which simplifies to

$$\min_{\widetilde{\mathbf{z}}\in\mathbb{R}^{s-1}} \widetilde{p}(\widetilde{\mathbf{z}}) := \frac{1}{2}\,\widetilde{\mathbf{z}}^T\widetilde{M}\,\widetilde{\mathbf{z}} + \widetilde{\mathbf{r}}^T\widetilde{\mathbf{z}}, \tag{4.22}$$

where

$$M = \begin{pmatrix} m_{11} & \widetilde{\mathbf{m}}^T \\ \widetilde{\mathbf{m}} & \widetilde{M} \end{pmatrix}, \quad \mathbf{r} = \begin{pmatrix} r_1 \\ \widetilde{\mathbf{r}} \end{pmatrix}, \quad \mathbf{z} = \begin{pmatrix} z_1 \\ \widetilde{\mathbf{z}} \end{pmatrix}.$$

We note that $\mathbf{a}_{\mathcal{F}} = \sigma P\mathbf{e}_1$, i.e. $\mathbf{a}_{\mathcal{F}}$ is a multiple of the first column of P, hence the remaining columns of P span $\{\mathbf{a}_{\mathcal{F}}\}^\perp$. Furthermore, a simple computation

shows that $\widetilde{M} = \widetilde{P}^T H_{\mathcal{F}\mathcal{F}} \widetilde{P}$, where $\widetilde{P}$ is the matrix obtained by deleting the first column of P. By reasoning as in the proof of Theorem 3.3.11 (see (3.54)), we find that $\zeta_{min}(\widetilde{M}) \geq \zeta_{min}(H_q)$ and $\zeta_{max}(\widetilde{M}) \leq \zeta_{max}(H_q)$, where $H_q = V^T H V$ and $V \in \mathbb{R}^{n\times(n-1)}$ is any matrix with orthogonal columns spanning $\{\mathbf{a}\}^\perp$. Therefore, if H_q is positive definite, then

$$\kappa(\widetilde{M}) \leq \kappa(H_q).$$

For any other $Z \in \mathbb{R}^{n\times(n-1)}$ with orthogonal columns spanning $\{\mathbf{a}\}^\perp$, we can write $V^T = DZ^T$ with $D \in \mathbb{R}^{(n-1)\times(n-1)}$ orthogonal; therefore, $V^T H V$ and $Z^T H Z$ are similar and $\kappa(H_q)$ does not depend on the choice of the orthonormal basis of $\{\mathbf{a}\}^\perp$. Furthermore, if H is positive definite, by the Cauchy's interlace theorem [116, Theorem 10.1.1] it is $\kappa(H_q) \leq \kappa(H)$.

The finite convergence result for strictly convex problems (Theorem 4.2.2) relies on the exact solution of (4.22). In infinite precision, this can be achieved by means of the CG algorithm, as in the GPCG method. Of course, in presence of roundoff errors, finite convergence is generally neither obtained nor required.

We can solve (4.22) by efficient gradient methods too. In this work, we investigate the use of the SDC gradient method [45] (see Section 2.1.3) as a solver for the minimization phase in the strictly convex case. In our opinion the P2GP framework can provide a way to exploit the regularizing properties exhibited by SDC (and other spectral gradient methods) in the solution of linear ill-posed problems with bounds and problems with bounds and a single linear constraint. Of course, the CG solver is still the reference choice in general, especially because it is able to deal with non-convexity through directions of negative curvature (as done, e.g.,in [107]), whereas handling negative curvatures with spectral gradient methods may be a non-trivial task (see, e.g.,[36] and references therein).

Once a descent direction $\mathbf{d}^k$ is obtained by approximately solving (4.20), a full step along this direction is performed starting from $\mathbf{x}^k$, and $\mathbf{x}^{k+1}$ is set equal to the resulting point if this is feasible. Otherwise $\mathbf{x}^{k+1} = P_{\Omega^k}(\mathbf{x}^k + \alpha^k \mathbf{d}^k)$ where α^k satisfying the sufficient decrease conditions is computed by using the safeguarded quadratic interpolation described in [108] (see Section 2.4.1). It is worth to underline that the projected line search involves the projection onto $\Omega^k = \Omega \cap \Omega(\mathbf{x}^k)$, i.e. the face o the polyhedron containing $\mathbf{x}^k$. This has a twofold importance: from the theoretical point of view it ensures that $\mathcal{A}(\mathbf{x}^k) \subseteq \mathcal{A}(\mathbf{x}^{k+1})$, while, from the practical point of view, it allows one to avoid "strange behaviors" of the algorithm around the solution if one chooses to project over Ω instead of Ω^k.

To show this issue we considered a strictly convex 3-dimensional toy problem of the form

$$\begin{aligned} \min \quad & \frac{1}{2}\mathbf{x}^T H\, \mathbf{x} - \mathbf{c}^T \mathbf{x}, \\ \text{s.t.} \quad & \sum_{i=1}^{3} x_i = 1, \\ & 0 \leq x_i \leq 0.75, \quad \forall\, i \in \{1,2,3\}, \end{aligned}$$

built up such that the solution lies in $\mathbf{x}^* = [0.75,\, 0.25,\, 0]^T$. Supposing to start from point $\mathbf{x} = [0.5,\, 0.5,\, 0]^T$ and applying only minimization phase steps, the algorithm using the projection over Ω oscillates around $\mathbf{x}^*$ passing continuously from one of the

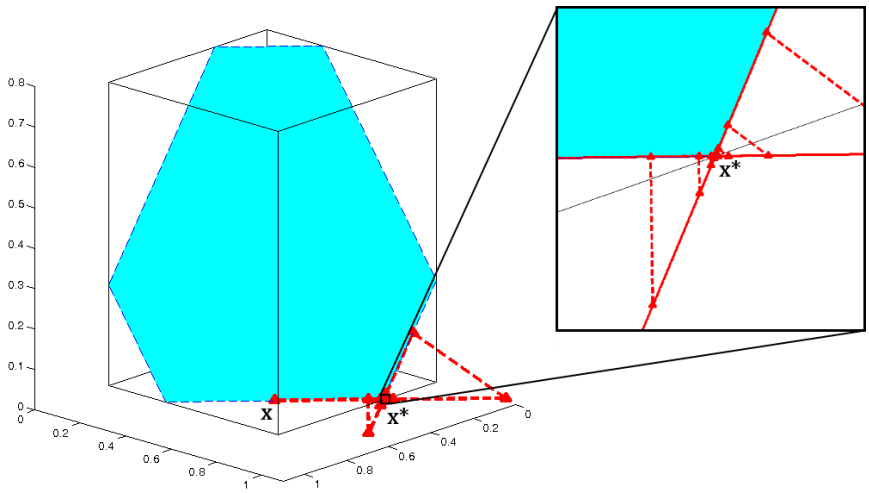

Figure 4.2. Visualization of the oscillating behavior of the algorithm in the case in which the projection onto Ω^k is replaced by the projection onto Ω in the line search following the minimization phase.

two edges containing $\mathbf{x}^*$ to the other, as can be seen in Figure 4.2. This oscillating behavior is clearly avoided if the proper projection onto Ω^k is considered, as shown in Figure 4.3. It is interesting to note that the difference in the use of the two projections is only noticeable in the case of SLBQPs and general QPs; in the case of BQP, indeed, the orthogonality between the faces meeting in the same vertex makes the projection operators $P_\Omega(\mathbf{y})$ and $P_{\Omega^k}(\mathbf{y})$ equivalent for all the points $\mathbf{y} \in \Omega(\mathbf{x}^k)$.

If the problem is not strictly convex, we choose the CG method for the minimization phase. If CG finds a direction $\mathbf{d}^k$ such that $(\mathbf{d}^k)^T H\,\mathbf{d}^k \leq 0$ we set $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{d}^k$, where α^k is the largest feasible steplength, i.e. the minimum breakpoint along $\mathbf{d}^k$, unless the objective function results to be unbounded along $\mathbf{d}^k$.

As already observed, the stopping criterion in the solution of problem (4.22) must not be too stringent, since the decision of continuing the minimization on the reduced space is left to the proportionality criterion. In order to stop the solver for problem (4.22), we check the progress in the reduction of the objective function as in the identification phase, i.e. we terminate the iterations if

$$\widetilde{p}(\widetilde{\mathbf{z}}^j) - \widetilde{p}(\widetilde{\mathbf{z}}^{j+1}) \leq \xi \max_{1 \leq l < j} \left\{ \widetilde{p}(\widetilde{\mathbf{z}}^l) - \widetilde{p}(\widetilde{\mathbf{z}}^{l+1}) \right\}, \tag{4.23}$$

where $\xi \in (0, 1)$ is not too small (the value used in the numerical experiments is given in Section 4.3). This choice follows [109]. If the active set has not changed and the current iterate is proportional, the minimization phase does not restart from scratch, but the minimization method continues its iterations as it had not been stopped.

4.2.3 Projections

Even if more efficient algorithms are available, since the size of the considered problems is in the order of tens of thousands, we decided to perform the projections by using the algorithm proposed by Dai and Fletcher in [38] (see Section 2.5.1.4 for

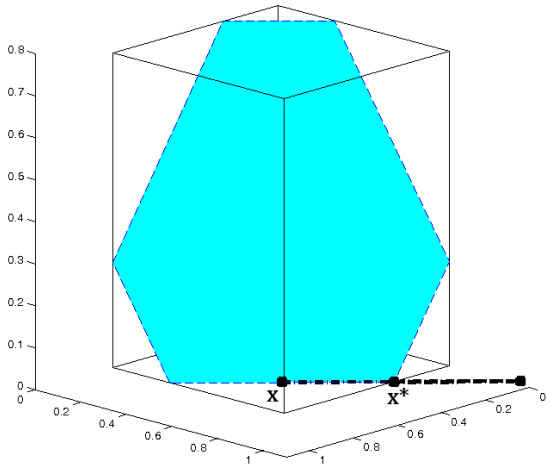

Figure 4.3. Behavior of the algorithm in the case in which the projection onto Ω^k is considered in the line search following the minimization phase.

further details) for which a MATLAB implementation is available and which has successfully used in gradient projection method for image-processing and machine learning problems [120, 127, 21]. Observe that, apart from the standard projections onto Ω (Step 9 of Algorithm 4.1), P2GP requires also projections onto $\Omega^k = \Omega \cap \Omega(\mathbf{x}^k)$ (Step 33 of Algorithm 4.1) and onto $T_\Omega(\mathbf{x}^k)$ (for the computation of $\nabla_\Omega f(\mathbf{x})$ and $\boldsymbol{\beta}(\mathbf{x}^k)$). To perform these projections we can still make use of the same algorithm. For the computation of $P_{\Omega^k}(\mathbf{y})$, it is sufficient to observe that the lower and upper bound vectors for Ω^k, $\bar{\mathbf{l}}$ and $\bar{\mathbf{u}}$, have components

$$\begin{aligned} \bar{l}_i^k &= \bar{u}_i^k = x_i^k, && \text{if } i \in \mathcal{A}(\mathbf{x}^k), \\ \bar{l}_i^k &= l_i,\ \bar{u}_i^k = u_i, && \text{if } i \in \mathcal{F}(\mathbf{x}^k). \end{aligned}$$

For the computation of the projected gradient, i.e. for the projection onto $T_\Omega(\mathbf{x}^k)$, we can use the same algorithm, imposing the linear constraint $\mathbf{a}^T\mathbf{y} = 0$ and observing that the bounds $\widetilde{\mathbf{l}}$ and $\widetilde{\mathbf{u}}$, have components

$$\begin{aligned} \widetilde{l}_i &= -\infty, \quad \widetilde{u}_i = +\infty, && \text{if } i \in \mathcal{F}(\mathbf{x}^k), \\ \widetilde{l}_i &= 0, \quad\ \widetilde{u}_i = +\infty, && \text{if } i \in \mathcal{A}_l(\mathbf{x}^k), \\ \widetilde{l}_i &= -\infty, \quad \widetilde{u}_i = 0, && \text{if } i \in \mathcal{A}_u(\mathbf{x}^k). \end{aligned}$$

4.3 Numerical experiments

In order to analyze the behavior of P2GP using both CG and SDC in the minimization phase, we performed numerical experiments on several problems, either generated with the aim of building test cases with varying characteristics (see Section 4.3.1) or coming from SVM training (see Section 4.3.2).

On the first set of problems, referred to as random problems because of the way they are built, we compared both versions of P2GP with the following methods:

- GPCG-like, a modification of P2GP where the termination of the minimization phase (performed by CG) is not driven by the proportionality criterion, but by the bindingness of the active variables, like in the GPCG method;

- PABB_{min}, a Projected Alternate BB method executing the line search as in P2GP and computing the first trial steplength with the ABB_{min} rule described in Section 4.2.1;

The first method was selected to evaluate the effect of the proportionality-based criterion in the minimization phase, the second one because of its effectiveness among general GP methods. P2GP, GPCG-like, and PABB_{min} were implemented in Matlab.

To further assess the behavior of P2GP, we also compared it, on the random and SVM problems, with the GP method implemented in BLG, a C code available from `http://users.clas.ufl.edu/hager/papers/Software/`. BLG solves nonlinear optimization problems with bounds and a single linear constraint, and can be considered as a benchmark for software based on gradient methods. Its details are described in [89, 82].

The following setting of the parameters was considered for P2GP: $\eta = 0.1$ in (4.16) and $\xi = 0.5$ in (4.23); $\mu_1 = 10^{-4}$ in (3.60); $\gamma_1 = 10^{12}$, $\gamma_2 = 10^{-12}$, $\gamma_3 = 10^{-2}$, and $\gamma_4 = 0.5$ in (4.18)-4.19; $q = 3$ and $\tau = 0.2$ in (2.6). Furthermore, when SDC was used in the minimization phase, $\overline{k} = 6$ and $l = 4$ were chosen in (2.8). A maximum number of 50 consecutive GP and CG (or SDC) iterations was also considered. The previous choices were also used for the GPCG-like method, except for the parameter ξ, which was set to 0.25. The parameters of PABB_{min} in common with P2GP were given the same values too, except τ, which was computed by the adaptive procedure described in [21], with 0.5 as starting value. Details on the stopping conditions used by the methods are given in Sections 4.3.3 and 4.3.4, where the results obtained on the test problems are discussed.

About the proportionality condition (4.13), a choice made according to Theorem 4.1.5 requires a knowledge which is usually unknown about the spectrum of H. A conservative approach would suggest to adopt a large value for Γ. However, such a choice is likely to be unsatisfactory in practice; in fact, a large Γ would foster high accuracy in the minimization phase, even at the initial steps of the algorithm, when the active constraints at the solution are far from being identified. Thus, we used the following adaptive strategy for updating Γ after line 37 of Algorithm 4.1:

if $\|\boldsymbol{\beta}^k\|_\infty > \Gamma \, \|\boldsymbol{\varphi}^k\|_2$ **then**
  $\Gamma = \max\{1.1 \cdot \Gamma,\ 1\}$;
else if $\mathcal{A}^k \neq \mathcal{A}^{k-1}$ **then**
  $\Gamma = \max\{0.9 \cdot \Gamma,\ 1\}$;
end if

Based on our numerical experience, we set the starting value of Γ equal to 1.

BLG was run using the gradient projection search direction (it also provides the Frank-Wolfe and affine-scaling directions). However, the code could switch to the Frank-Wolfe direction, according to inner automatic criteria. Note that BLG uses a cyclic BB step length $\overline{\alpha}^k$ as trial steplength, together with an adaptive non-monotone

line search along the feasible direction $P_\Omega(\mathbf{x}^k - \overline{\alpha}^k \nabla f^k) - \mathbf{x}^k$ (see [89] for the details). Of course, the BLG features exploiting the form of a quadratic objective function were used. The stopping criteria applied with the random problems and the SVM ones are specified in Sections 4.3.3 and 4.3.4, respectively. Further details on the use of BLG are given there.

All the experiments were carried out using a 64-bit Intel Core i7-6500, with maximum clock frequency of 3.10 GHz, 8 GB of RAM, and 4 MB of cache memory. BLG (v. 1.4) and SVMsubspace (v. 1.0) were compiled by using gcc 5.4.0. P2GP, GPCG-like, and PABB_{min} were run under MATLAB 7.14 (R2012a). The elapsed times reported for the Matlab codes were measured by using the `tic` and `toc` commands.

The MATLAB code implementing P2GP used in the experiments is available from `https://github.com/diserafi/P2GP`. It includes the test problem generator described in Section 4.3.1.

4.3.1 Random test problems

The implementations of all methods were run on random SLBQPs built by modifying the procedure for generating BQPs proposed in [108]. The new procedure first computes a point $\mathbf{x}^*$ and then builds a problem of type (4.1) having $\mathbf{x}^*$ as stationary point. Obviously, if the problem is strictly convex, $\mathbf{x}^*$ is its solution. The following parameters are used to define the problem:

- `n`, number of variables (i.e. n);
- `ncond`, $\log_{10} \kappa(H)$;
- `zeroeig` $\in [0, 1)$, fraction of zero eigenvalues of H;
- `negeig` $\in [0, 1)$, fraction of negative eigenvalues of H;
- `naxsol` $\in [0, 1)$, fraction of active variables at $\mathbf{x}^*$;
- `degvar` $\in [0, 1)$, fraction of active variables at $\mathbf{x}^*$ that are degenerate;
- `ndeg` $\in \{0, 1, 2, \ldots\}$, amount of near-degeneracy;
- `linear`, 1 for SLBQPs, and 0 for BQPs;
- `nax0` $\in [0, 1)$, fraction of active variables at the starting point.

The components of $\mathbf{x}^*$ are computed as random numbers from the uniform distribution in $(-1, 1)$. All random numbers considered next are from uniform distributions too. The Hessian matrix H is defined as

$$H = G\, D\, G^T, \tag{4.24}$$

where D is a diagonal matrix and $G = (I - 2\,\mathbf{p}_3\mathbf{p}_3^T)(I - 2\,\mathbf{p}_2\mathbf{p}_2^T)(I - 2\,\mathbf{p}_1\mathbf{p}_1^T)$, with $\mathbf{p}_j$ unit vectors. For $j = 1, 2, 3$, the components of $\mathbf{p}_j$ are obtained by generating

$\overline{\mathbf{p}}_j = (\overline{p}_{ji})_{i=1,\ldots,n}$, where the values $\overline{p}_{ji}$ are random numbers in $(-1,1)$, and setting $\mathbf{p}_j = \overline{\mathbf{p}}_j / \|\overline{\mathbf{p}}_j\|$. The diagonal entries of D are defined as follows:

$$d_{ii} = \begin{cases} 0 & \text{for approximately } \texttt{zeroeig} * \texttt{n} \text{ values of } i, \\ -10^{\frac{i-1}{n-1}(\texttt{ncond})} & \text{for approximately } \texttt{negeig} * \texttt{n} \text{ values of } i, \\ 10^{\frac{i-1}{n-1}(\texttt{ncond})} & \text{for the remaing values of } i. \end{cases}$$

We note that `zeroeig` and `negeig` are not the actual fraction of zero and negative eigenvalues. The actual fraction of zero eigenvalues is determined by generating a random number $\xi_i \in (0,1)$ for each i, and by setting $d_{ii} = 0$ if $\xi_i \leq \texttt{zeroeig}$; the same strategy is used to determine the actual number of negative eigenvalues. We also observe that $\kappa(H) = 10^{\texttt{ncond}}$, if H has no zero eigenvalues.

In order to define the active variables at $\mathbf{x}^*$, n random numbers $\chi_i \in (0,1)$ are computed, and the index i is put in $\mathcal{A}^*$ if $\chi_i \leq \texttt{naxsol}$; then $\mathcal{A}^*$ is partitioned into the sets $\mathcal{A}^*_N$ and $\mathcal{A}^* \setminus \mathcal{A}^*_N$, with $|\mathcal{A}^* \setminus \mathcal{A}^*_N|$ approximately equal to $\lfloor \texttt{degvar} * \texttt{naxsol} * \texttt{n} \rfloor$. More precisely, an index i is put in $\mathcal{A}^* \setminus \mathcal{A}^*_N$ if $\psi_i \leq \texttt{degvar}$, where ψ_i is a random number in $(0,1)$, and is put in $\mathcal{A}^*_N$ otherwise. The vector $\boldsymbol{\lambda}^*$ of Lagrange multipliers associated with the box constraints at $\mathbf{x}^*$ is initially set as

$$\lambda^*_i = \begin{cases} 10^{-\mu_i \, \texttt{ndeg}} & \text{if } i \in \mathcal{A}^*_N, \\ 0 & \text{otherwise,} \end{cases}$$

where μ_i is a random number in $(0,1)$. Note that the larger `ndeg`, the closer to 0 is the value of λ^*_i, for $i \in \mathcal{A}^*_N$ (in this sense `ndeg` indicates the amount of near-degeneracy). The set $\mathcal{A}^*$ is split into $\mathcal{A}^*_l$ and $\mathcal{A}^*_u$ as follows: for each $i \in \mathcal{A}^*$, a random number $\nu_i \in (0,1)$ is generated; i is put in $\mathcal{A}^*_l$ if $\nu_i < 0.5$, and in $\mathcal{A}^*_u$ otherwise. Then, if $i \in \mathcal{A}^*_u$, the corresponding Lagrange multiplier is modified by setting $\lambda^*_i = -\lambda^*_i$. The lower and upper bounds $\mathbf{l}$ and $\mathbf{u}$ are defined as follows:

$$\begin{array}{l} l_i = -1 \text{ and } u_i = 1 \quad \text{if } i \notin \mathcal{A}^*, \\ l_i = x^*_i \text{ and } u_i = 1 \quad \text{if } i \in \mathcal{A}^*_l, \\ l_i = -1 \text{ and } u_i = x^*_i \text{ if } i \in \mathcal{A}^*_u. \end{array}$$

If $\texttt{linear} = 0$, the linear constraint is neglected. If $\texttt{linear} = 1$, the vector $\mathbf{a}$ in (4.1) is computed by randomly generating its components in $(-1,1)$, the scalar b is set to $\mathbf{a}^T \mathbf{x}^*$, and the vector $\mathbf{c}$ is defined so that the KKT conditions at the solution are satisfied:

$$\mathbf{c} = \begin{cases} H\,\mathbf{x}^* - \boldsymbol{\lambda}^* & \text{if } \texttt{linear} = 0, \\ H\,\mathbf{x}^* - \boldsymbol{\lambda}^* - \rho^*\,\mathbf{a} & \text{if } \texttt{linear} = 1, \end{cases}$$

where ρ^* is a random number in $(-1,1) \setminus \{0\}$ representing the Lagrange multiplier associated with the linear constraint.

By reasoning as with $\mathbf{x}^*$, approximately $\texttt{nax0} * \texttt{n}$ components of the starting point $\mathbf{x}^0$ are set as $x^0_i = l_i$ or $x^0_i = u_i$. The remaining components are defined as $x^0_i = (l_i + u_i)/2$. Note that $\mathbf{x}^0$ may not be feasible; in any case, it will be projected onto Ω by the optimization methods considered here.

Finally, we note that although $\mathbf{x}^*$ is a stationary point of the problem generated by the procedure described so far, there is no guarantee that P2GP converges to $\mathbf{x}^*$ if the problem is not strictly convex.

The following sets of test problems, with size $\texttt{n} = 20000$, were generated:

- 27 strictly convex SLBQPs with non-degenerate solutions, obtained by setting `ncond` $= 4, 5, 6$, `zeroeig` $= 0$, `negeig` $= 0$, `naxsol` $= 0.1, 0.5, 0.9$, `degvar` $= 0$, `ndeg` $= 0, 1, 3$, and `linear` $= 1$;

- 18 strictly convex SLBQPs with degenerate solutions, obtained by setting `ncond` $= 4, 5, 6$, `zeroeig` $= 0$, `negeig` $= 0$, `naxsol` $= 0.1, 0.5, 0.9$, `degvar` $= 0.2, 0.5$, `ndeg` $= 1$, and `linear` $= 1$;

- 27 convex (but not stricltly convex) SLBQPs, obtained by setting `ncond` $= 4, 5, 6$, `zeroeig` $= 0.1, 0.2, 0.5$, `negeig` $= 0$, `naxsol` $= 0.1, 0.5, 0.9$, `degvar` $= 0$, `ndeg` $= 1$, and `linear` $= 1$;

- 27 non-convex SLBQPs, obtained by setting `ncond` $= 4, 5, 6$, `zeroeig` $= 0$, `negeig` $= 0.1, 0.2, 0.5$, `naxsol` $= 0.1, 0.5, 0.9$, `degvar` $= 0$, `ndeg` $= 1$, and `linear` $= 1$;

Since BQPs are special cases of SLBQPs, four sets of BQPs were also generated, by setting `linear` $= 0$ and choosing all remaining parameters as specified above. All the methods were applied to each problem with four starting points, corresponding to `nax0` $= 0, 0.1, 0.5, 0.9$.

4.3.2 SVM test problems

SLBQP test problems corresponding to the dual formulation of two-class C-SVM classification problems were also used (see, e.g., [122]). Ten problems from the LIBSVM data set, available from `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`, were considered, whose details (size of the problem, features and nonzeros in the data) are given in Table 4.1. A linear kernel was used, leading to problems with positive semidefinite Hessian matrices. The penalty parameter C was set to 10. For most of the problems, the number of nonzeros is much smaller than the product between size and features, showing that the data are relatively sparse. It is worth noting that SVMsubspace has been designed to take advantage of this issue, while the sparsity of the data has not been fully exploited when running P2GP.

problem	size	features	nonzeros	density
a6a	11220	122	155608	11.37%
a7a	16100	122	223304	11.37%
a8a	22696	123	314815	11.28%
a9a	32561	123	451592	11.28%
ijcnn1	49990	22	649870	59.09%
phishing	11055	68	331650	44.12%
real-sim	72309	20958	3709083	0.24%
w6a	17188	300	200470	3.89%
w7a	24692	300	288148	3.89%
w8a	49749	300	579586	3.88%

Table 4.1. Details of the SVM test set.

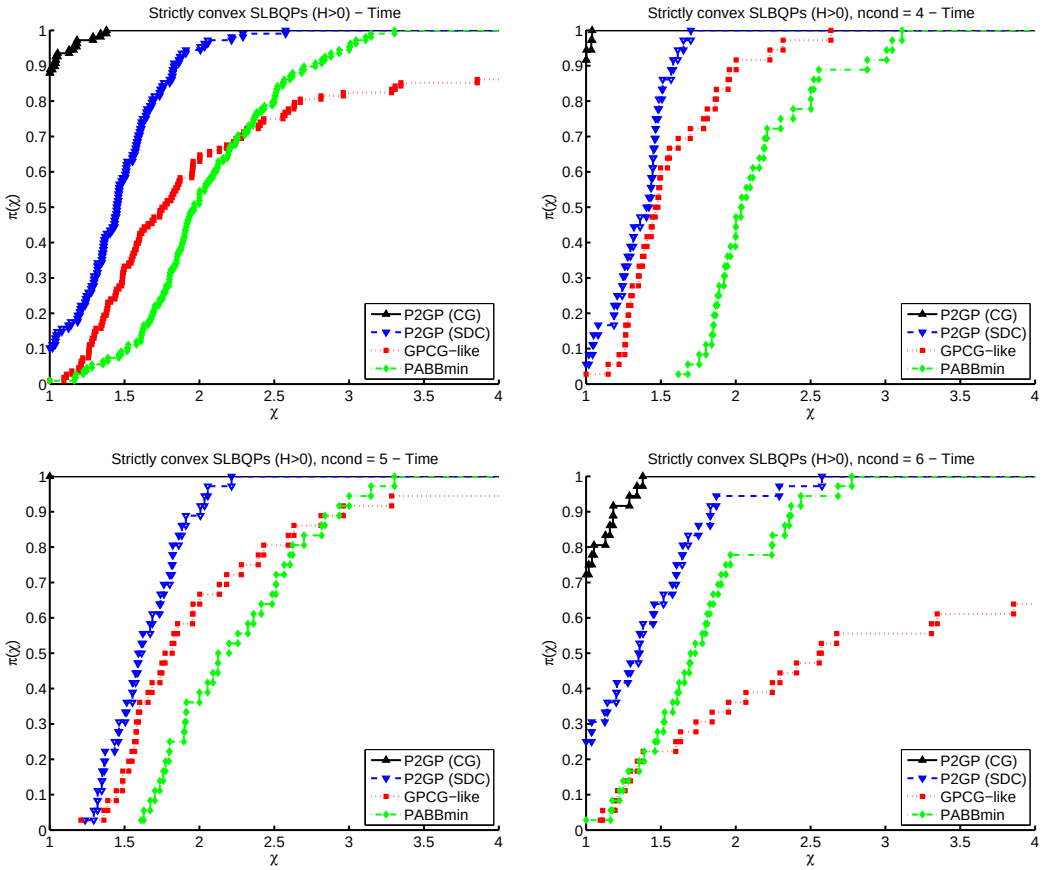

Figure 4.4. Performance profiles of P2GP (with CG and SDC), PABB_{min}, and GPCG-like on strictly-convex SLBQPs with non-degenerate solutions: execution times for all the problems (*top left*), for $\kappa(H) = 10^4$ (*top right*), for $\kappa(H) = 10^5$ (*bottom left*), and for $\kappa(H) = 10^6$ (*bottom right*).

4.3.3 Results on random problems

We first discuss the results obtained by running the implementations of the P2GP, PABB_{min} and GPCG-like methods on the problems described in Section 4.3.1. In the stopping condition (4.15), $tol = 10^{-6}\|\boldsymbol{\varphi}^0 + \boldsymbol{\beta}^0\|$ was used; furthermore, at most 30000 matrix-vector products and 30000 projections were allowed, declaring failures if these limits were achieved without satisfying condition (4.15). The methods were compared by using the performance profiles proposed by Dolan and Moré [52]. We note that the performance profiles in this section may show a number of failures larger than the actual one, because the range on the horizontal axis has been limited to enhance readability. However, all the failures will be explicitly reported in the text.

Figure 4.4 shows the performance profiles, $\pi(\chi)$, of the three methods on the set of strictly convex SLBQPs with non-degenerate solutions, using the execution time as performance metric. The profiles corresponding to all the problems and to those with $\kappa(H) = 10^4$, $\kappa(H) = 10^5$, and $\kappa(H) = 10^6$ are reported. We see that the version of P2GP using CG in the minimization phase has by far the best

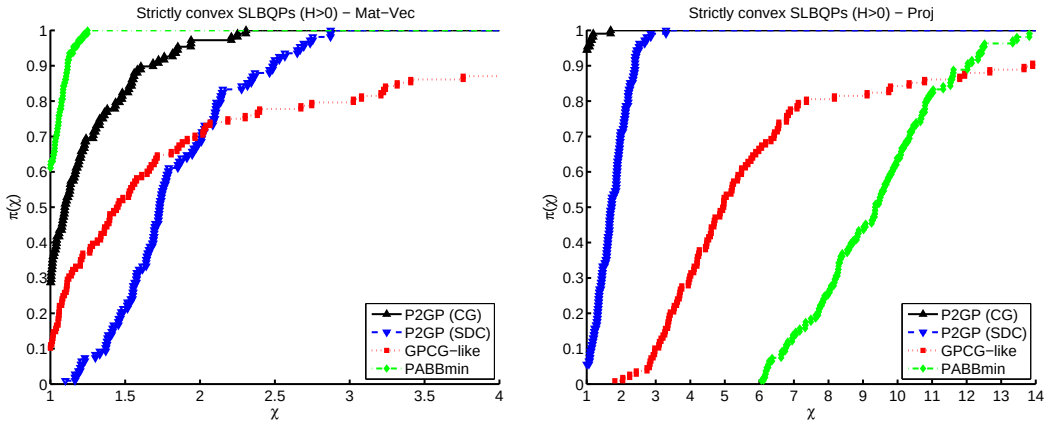

Figure 4.5. Performance profiles of P2GP (with CG and SDC), PABB_{min}, and GPCG-like on strictly convex SLBQPs with non-degenerate solutions: number of matrix-vector products (*left*) and projections (*right*).

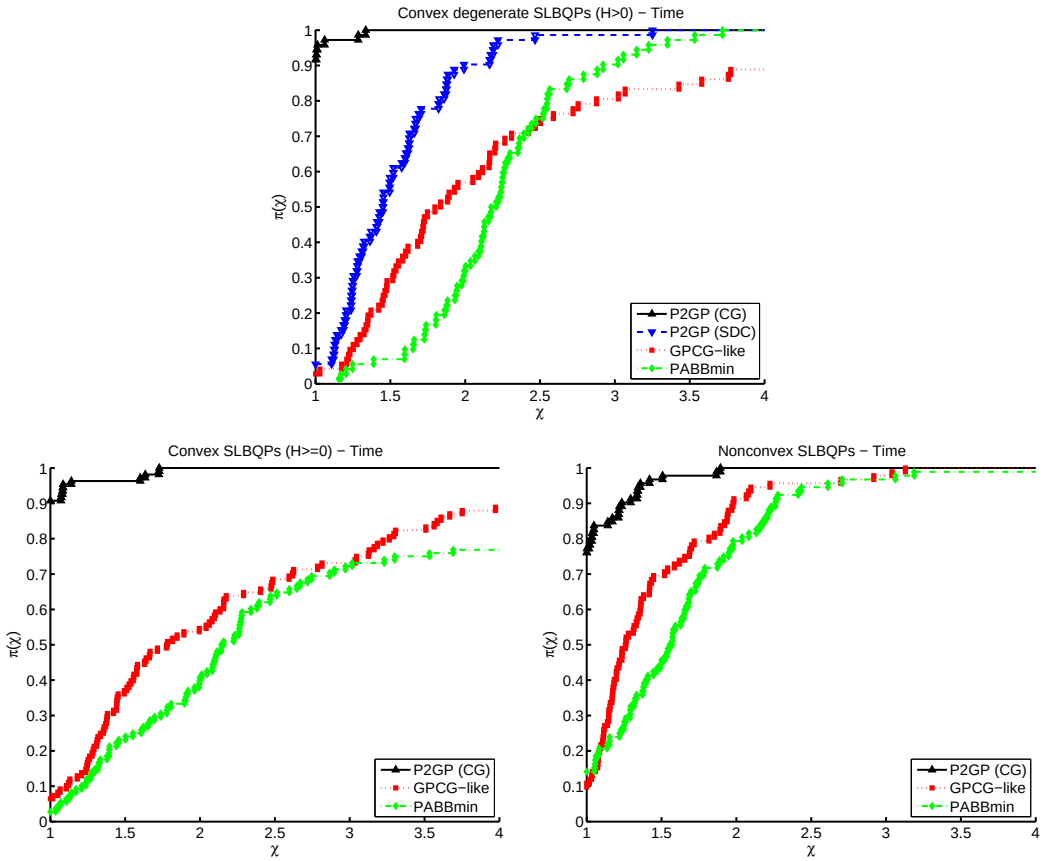

Figure 4.6. Performance profiles (execution times) of P2GP (with CG and SDC), PABB_{min}, and GPCG-like on strictly convex SLBQPs with degenerate solutions (*top*), convex SLBQPs (*bottom left*), non-convex SLBQPs (*bottom right*).

performance. P2GP with SDC is faster than the PABB_{min} and GPCG-like methods too. GPCG-like appears very sensitive to the condition number of the Hessian

matrix: its performance deteriorates as $\kappa(H)$ increases and the method becomes less effective than PABB_{min} when $\kappa(H) = 10^6$. This shows that the criterion used to terminate the minimization phase is more effective than the criterion based on the bindingness of the active variables, especially as $\kappa(H)$ increases. We also report that the GPCG-like method has 6 failures over 36 runs for the problems with $\kappa(H) = 10^6$.

For the previous problems, the performance profiles concerning the number of matrix-vector products and the number of projections are also shown, in Figure 4.5. We see that PABB_{min} performs the smallest number of matrix-vector products, followed by P2GP with GC, and then by GPCG-like and P2GP with SDC. On the other hand, the number of projections computed by P2GP with CG and with SDC is much smaller than for the other methods; as expected, the maximum number of projections is computed by PABB_{min}. This shows that the performance of the methods cannot be measured only in terms of matrix-vector products; the cost of the projections must also be considered, especially when the structure of the Hessian makes the computational cost of the matrix-vector products lower than $O(n^2)$. The good behavior of P2GP results from the balance between matrix-vector products and projections.

The performance profiles concerning the execution times on the strictly convex SLBQPs with degenerate solutions, on the convex (but not strictly convex) SLBQPs, and on the non-convex ones are reported in Figure 4.6. Of course, the version of P2GP using the SDC solver was not applied to the last two sets of problems. In the case of non-convex problems, only 85% of the runs were considered, corresponding to the cases where the values of the objective function at the solutions computed by the different methods differ by less than 1%. P2GP with CG is generally the best method, followed by GPCG-like and then by PABB_{min}. Furthermore, on strictly convex problems with degenerate solutions, P2GP with SDC performs better than GPCG-like and PABB_{min}. GPCG-like is less robust than the other methods, since it has 4 failures on the degenerate stricltly convex problems and 8 failures on the convex ones. This confirms the effectiveness of the proportionality-based criterion.

For completeness, we also run the experiments on the strictly convex problems with non-degenerate solutions by replacing the line-search strategy in PABB_{min} with a monotone line search along the feasible direction [13, Section 2.3.1], which requires only one projection per GP iteration. We note that this line search does not guarantee in general that the sequence generated by the GP method identifies in a finite number of steps the variables that are active at the solution (see, e.g.,[44]). Nevertheless, we made experiments with the line search along the feasible direction, to see if it may lead to any time gain in practice. The results obtained, not reported here for the sake of space, show that the two line searches lead to comparable times when the number of active variables at the solution is small, i.e. `naxsol` $= 0.1$. On the other hand, the execution time with the original line search is slightly smaller when the number of active variables at the solution is larger.

Finally, the performance profiles concerning the execution times taken by the P2GP, PABB_{min} and GPCG-like methods on the strictly convex BQPs with non-degenerate and degenerate solutions, on the convex (but not strictly convex) BQPs, and on the non-convex ones are shown in Figure 4.7. Only 97% of the runs on the non-convex problems are selected, using the same criterion applied to non-convex SLBQPs. P2GP with CG is again the most efficient method. The behavior of

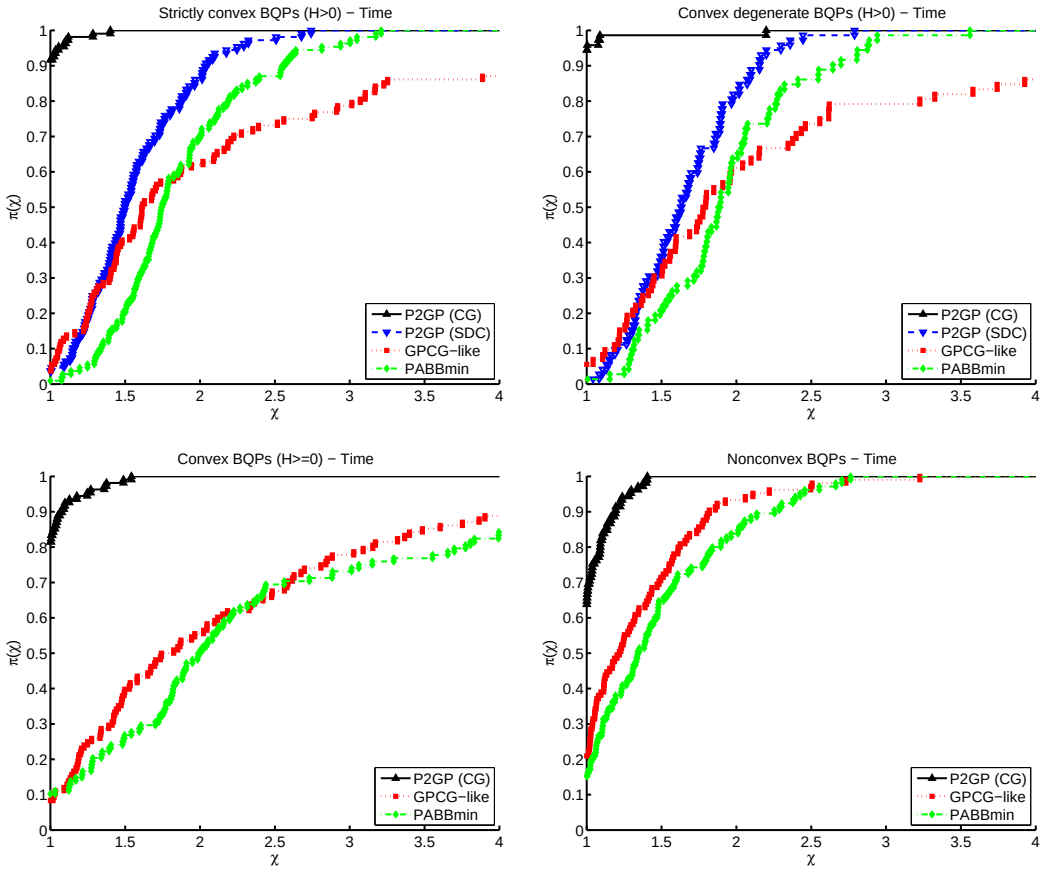

Figure 4.7. Performance profiles (execution times) of P2GP (with CG and SDC), PABB_{min}, and GPCG-like on strictly convex BQPs with non-degenerate solutions (*top left*), strictly convex BQPs with degenerate solutions (*top right*), convex BQPs (*bottom left*), non-convex BQPs (*bottom right*).

the methods is similar to that shown on SLBQPs. However, P2GP with SDC and PABB_{min} have closer behaviors, according to the smaller time required by projections onto boxes, which leads to a reduction of the execution time of PABB_{min}. GPCG-like has again some failures: 6 on the strictly convex problems with non-degenerate solutions, 5 on the ones with degenerate solutions, and 9 on the convex (but not strictly convex) problems.

Now we compare P2GP (using CG) with BLG on the random problems. BLG was run in its full-space mode (default mode), because the form of the Hessian (4.24) does not allow BLG to take advantage of the subspace mode. The stopping condition (4.15) was implemented in BLG, and the code was run with the same tolerance and the same maximum numbers of matrix-vector products and projections used for P2GP. Default values were used for the remaining parameters of BLG. Of course, a comparison of the two codes in terms of execution time would be misleading, since BLG is written in C, while P2GP has been implemented in Matlab. Therefore, we consider the matrix-vector products. We do not show a comparison in terms of projections too, because BLG does a projection at each iteration, and this generally results in many more projections than P2GP. Performance profiles are

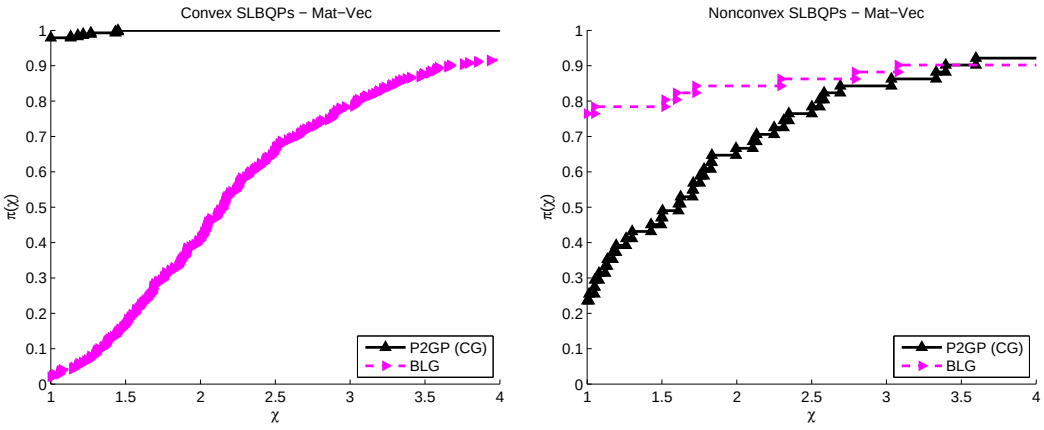

Figure 4.8. Performance profiles of P2GP (with CG) and BLG on convex (*left*) and non-convex (*right*) SLBQPs: number of matrix-vector products.

provided in Figure 4.8. The results concerning all the types of convex problems are shown together, since their profiles are similar. On these problems P2GP appears more efficient than BLG; we also verified that the objective function values at the solutions computed by the two codes agree on at least six significant digits and are smaller for P2GP for 70% of the test cases. Furthermore, in four cases BLG does not satisfy condition (4.15) within the maximum number of matrix-vector products and projections. The situation is different for the non-convex problems, where the number of matrix-vector products performed by BLG is smaller. In this case, we verified that BLG also used Frank-Wolfe directions, which were never chosen for the convex problems. This not only reduced the number of matrix-vector products, but often led to smaller objective function values. The values of the objective function at the solutions computed by the two methods differ by less than 1% for only 47% of the test cases, which are the ones considered in the performance profiles on the right of Figure 4.8. On the other hand, in three cases BLG performs the maximum number of matrix-vector products without achieving the required accuracy.

4.3.4 Results on SVM problems

In order to read the SVM problems, available in the LIBSVM format, BLG was run through the SVMsubspace code, available from `http://users.clas.ufl.edu/hager/papers/Software/`. Since we were interested in comparing P2GP with the GP implementation provided by BLG, SVMsubspace was modified to have the SVM subspace equal to the entire space, i.e. to apply BLG to the full SVM problem. For completeness we also run SVMsubspace in its subspace mode (see [82]), to see what the performance gain is with this feature. In the following, we refer to the former implementation as BLGfull, and to the latter as SVMsubspace.

Following [82], BLGfull and SVMsubspace were used with their original stopping condition, with tolerance 10^{-3}. P2GP was terminated when the infinity norm of the projected gradient was smaller then the same tolerance. With these stopping criteria, the two codes returned objective function values agreeing on about six significant digits, with smaller function values generally obtained by P2GP. At most 70000

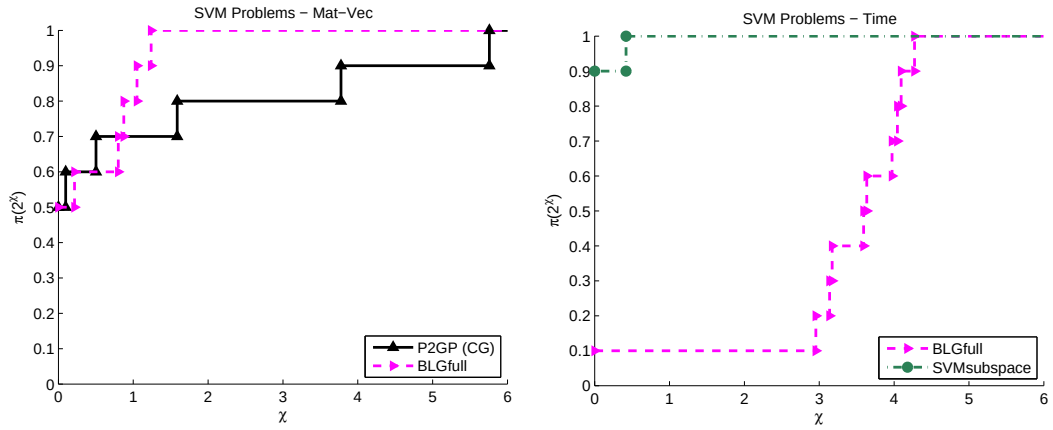

Figure 4.9. Performance profiles on SVM test problems: number of matrix-vector of P2GP (with CG) and BLG (*left*), and execution times of BLGfull and SVMsubspace (*right*).

matrix-vector products and 70000 projections were allowed, but they were never reached.

In Figure 4.9, left, the performance profiles (in logarithmic scale) concerning the matrix-vector products of P2GP (with CG) and BLGfull are shown. A comparison in terms of projections and execution times is not carried out for the same reasons explained for the random problems. BLGfull appears superior than P2GP; on the other hand, we verified that the number of projections performed by BLG is by far greater than that of P2GP for eight out of ten problems. However, it must be noted that SVMsubspace is much faster than BLGfull, as shown by the performance profiles concerning their execution times (see Figure 4.9, right). This confirms the great advantage of performing reduced-size matrix-vector products in solving the subspace problems for this class of test cases.

Chapter 5

Application to the solution of contact mechanics problems

In the previous chapter we introduced P2GP, a novel method for the solution of SLBQPs and BQPs. Even if the original idea was mainly to deal with SLBQPs, the numerical results shown in Section 4.3 suggest that the new method outperforms standard gradient projection schemes and the well known GPCG algorithm [109] in the solution of BQPs as well. To further asses the performance of P2GP in the solution of BQPs, in this chapter we compare it with MPRGP by Dostál [65] (described in Section 2.4.3), in solving the bound constrained subproblems arising in an Augmented Lagrangian method for problems of form (1.1) modeling contact mechanics applications. We start by introducing the Augmented Lagrangian framework called SMALBE (Semi-Monotonic Augmented Lagrangian for Bound and Equality constraints) [55] and by comparing the performance of P2GP and MPRGP in the solution of elliptic model problems. Then, we show how the discretization of contact mechanics problems by the TFETI (Total Finite Element Tearing and Interconnecting) domain decomposition method [60] leads to problems of the form (1.1). Finally, we compare the performance of P2GP and MPRGP in the solution of one 2D contact problem with friction and two 3D frictionless contact problems, showing the competitiveness of the proposed method.

5.1 Augmented Lagrangian methods

In the previous chapters we have explored the possibility of solving problems of the form (1.1) by gradient projection methods. In the final part of Chapter (3) we observed how these approaches are of practical interest in the case of sparse constraints, thanks to the availability of efficient projection methods.

In the case of dense constraints the use of *Augmented Lagrangian methods* can typically lead to better performances. The idea behind this class of methods, which we will briefly describe following [32, 34], is to ease the solution of the problem by getting rid of the dense linear constraints forcing implicitly their satisfaction by introducing a so called *penalty function*.

Starting from the Lagrangian function related to the equality constraints in (1.1),

i.e. the function

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{2}\mathbf{x}^T H\,\mathbf{x} - \mathbf{c}^T\mathbf{x} - (A\,\mathbf{x} - \mathbf{b})^T\boldsymbol{\theta},$$

we define the *Augmented Lagrangian function* as

$$\mathcal{L}_A(\mathbf{x}, \boldsymbol{\theta}, \rho) = \frac{1}{2}\mathbf{x}^T H\,\mathbf{x} - \mathbf{c}^T\mathbf{x} - (A\,\mathbf{x} - \mathbf{b})^T\boldsymbol{\theta} + \frac{\rho}{2}\|A\,\mathbf{x} - \mathbf{b}\|^2. \tag{5.1}$$

Given a fixed vector $\widehat{\boldsymbol{\theta}}$ and a fixed scalar $\hat{\rho}$, consider the problem

$$\begin{array}{ll} \min & \mathcal{L}_A(\mathbf{x}, \widehat{\boldsymbol{\theta}}, \hat{\rho}), \\ \text{s.t.} & \mathbf{l} \le \mathbf{x} \le \mathbf{u}. \end{array} \tag{5.2}$$

The solution $\overline{\mathbf{x}}$ to this problem satisfies the first order optimality conditions

$$\overline{\mathbf{g}} - A^T\widehat{\boldsymbol{\theta}} + \hat{\rho}A^T(A\,\overline{\mathbf{x}} - \mathbf{b}) = \sum_{i=1}^{n} \overline{\lambda}_i \mathbf{e}_i,$$

$$\overline{\lambda}_i \ge 0 \;\text{ if }\; i \in \mathcal{A}_l(\overline{\mathbf{x}}), \quad \overline{\lambda}_i \le 0 \;\text{ if }\; i \in \mathcal{A}_u(\overline{\mathbf{x}}), \quad \overline{\lambda}_i = 0 \;\text{ if }\; i \in \mathcal{F}(\overline{\mathbf{x}}),$$

where $\overline{\boldsymbol{\lambda}} = (\overline{\lambda}_i)_{i=1,\dots,n}$ is the vector of Lagrange multipliers associated with the bound constraints. By comparing the first condition to (3.1), one can clearly observe that $\overline{\mathbf{x}}$ concides with the solution $\mathbf{x}^*$ to (1.1) if

$$\boldsymbol{\theta}^* = \widehat{\boldsymbol{\theta}} - \hat{\rho}(A\,\overline{\mathbf{x}} - \mathbf{b}), \tag{5.3}$$

or, equivalently, if

$$A\,\overline{\mathbf{x}} - \mathbf{b} = \frac{1}{\hat{\rho}}(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*). \tag{5.4}$$

An Augmented Lagrangian method for problem (1.1), starting from a triple $(\mathbf{x}^0, \boldsymbol{\theta}^0, \rho^0)$, builds up a sequence of triplets $\left\{(\mathbf{x}^k, \boldsymbol{\theta}^k, \rho^k)\right\}$ in the following way. At each step k at first the estimate $\mathbf{x}^{k+1}$ is computed by solving the problem

$$\begin{array}{ll} \min & \mathcal{L}_A(\mathbf{x}, \boldsymbol{\theta}^k, \rho^k), \\ \text{s.t.} & \mathbf{l} \le \mathbf{x} \le \mathbf{u}, \end{array} \tag{5.5}$$

then the Lagrange multipliers estimate $\boldsymbol{\theta}^{k+1}$ and the penalty parameter ρ^{k+1} are updated accordingly.

Starting from (5.3), a possible way to update the Lagrange multipliers vector $\boldsymbol{\theta}^k$ is to set

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k - \rho^k(A\,\mathbf{x}^{k+1} - \mathbf{b}), \tag{5.6}$$

which is usually referred to as the *first-order Lagrange multiplier estimate*. It is interesting to note that, since

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}_A(\mathbf{x}, \boldsymbol{\theta}, \rho) = -(A\,\mathbf{x} - \mathbf{b}),$$

the update (5.6) can be seen as a gradient ascent step with step length ρ^k aimed at driving $(\mathbf{x}^k, \boldsymbol{\theta}^k)$ towards the saddle point of $\mathcal{L}_A$. Even if more sophisticated estimates could lead to better convergence rate for the method, the first order estimate is a

very common choice thanks to its low computational cost and its good practical performance.

As regards the update of ρ^k, the idea is to increase ρ^k, if needed, to foster the feasibility of $\mathbf{x}^{k+1}$. Observe that $\mathcal{L}_A(\mathbf{x}, \boldsymbol{\theta}^k, \rho^k)$ is a quadratic function, whose Hessian matrix is

$$\nabla_x^2 \mathcal{L}_A(\mathbf{x}, \boldsymbol{\theta}^k, \rho^k) = H + \rho^k A^T A,$$

which tends to be ill-conditioned as ρ^k increases. However, equation (5.4) suggests that to make $\mathbf{x}^k$ approach $\mathbf{x}^*$ there is no need to push ρ^k towards infinity if one can improve the quality of the Lagrange multipliers estimate $\boldsymbol{\theta}^k$.

To ease the description of the Augmented Lagrangian framework introduced in the next section, here we introduce the projected gradient of the Augmented Lagrangian function, w.r.t. the bound constraints, as

$$[\nabla_B \mathcal{L}_A(\mathbf{x}, \boldsymbol{\theta}, \rho)]_i = \begin{cases} -\nabla_{x_i} \mathcal{L}_A(\mathbf{x}, \boldsymbol{\theta}, \rho) & \text{if } i \in \mathcal{F}(\mathbf{x}), \\ \max\{0, -\nabla_{x_i} \mathcal{L}_A(\mathbf{x}, \boldsymbol{\theta}, \rho)\} & \text{if } i \in \mathcal{A}_l(\mathbf{x}), \\ \min\{0, -\nabla_{x_i} \mathcal{L}_A(\mathbf{x}, \boldsymbol{\theta}, \rho)\} & \text{if } i \in \mathcal{A}_u(\mathbf{x}). \end{cases}$$

The projected gradient is a natural choice as optimality measure for the bound constrained subproblem.

5.1.1 The SMALBE and SMALBE-M frameworks

The *Semi-Monotonic Augmented Lagrangian for Bound and Equality constraints* (SMALBE) [56] is an Augmented Lagrangian framework based on the original method proposed by Conn, Gould and Toint [32, 33]. The main difference between the two is the use of an adaptive precision control for the BQP subproblem [59] based on the feasibility of the current iterate, i.e.

$$\left\| \nabla_B \mathcal{L}_A(\mathbf{x}^{k+1}, \boldsymbol{\theta}^k, \rho^k) \right\| \leq \min \left\{ M \| A\,\mathbf{x}^{k+1} - \mathbf{b} \|, \, \eta \right\}, \tag{5.7}$$

with $M > 0$ and $\eta > 0$ given constants, in place of the original requirement of

$$\left\| \nabla_B \mathcal{L}_A(\mathbf{x}^{k+1}, \boldsymbol{\theta}^k, \rho^k) \right\| \leq \omega^k,$$

where $\{\omega^k\}$ is a priori defined and converges to zero. Another difference between SMALBE and the method in [32] lies in the adaptive strategy for the update of the penalty parameter ρ introduced in [55]. At each step of SMALBE, starting from the triple $(\mathbf{x}^k, \boldsymbol{\theta}^k, \rho^k)$, an approximate solution $\mathbf{x}^{k+1}$ to the BQP subproblem (5.5) is computed, using any convergent algorithm, such that it satisfies the optimality condition (5.7). Given $\mathbf{x}^{k+1}$, the Lagrangian parameters are updated using the first order estimate

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k - \rho^k \left(A\,\mathbf{x}^{k+1} - \mathbf{b} \right),$$

and a check on the sufficient decrease of the Augmented Lagrangian function is performed to decide whether increase the penalty parameter or not.

To avoid the ill-conditioning due to the increase of the penalty parameter, a modification of the SMALBE, called SMALBE-M, was proposed in [58]. In SMALBE-M the penalty parameter remains fixed and the constant M in (5.7) is updated instead. The SMALBE-M framework is depicted in Algorithm 5.1.

The following result can be proved [63, Theorem 9.2].

Algorithm 5.1 SMALBE-M (Semi-Monotonic Augmented Lagrangian for Bound and Equality constraints with modification of M)

$tol \geq 0$; $\eta > 0$; $\vartheta \in (0,1)$; $\rho > 0$; $M_0 = M_1 \in \mathbb{R}$; $\boldsymbol{\theta}^0 \in \mathbb{R}^m$; $\mathbf{x}^0 \in \mathbb{R}^n$; $k = 0$;
while $(\|\nabla_B \mathcal{L}_A(\mathbf{x}^k, \boldsymbol{\theta}^k, \rho)\| \leq tol \ \wedge \ \|A\,\mathbf{x}^k - \mathbf{b}\| \leq tol)$ **do** ▷ Main loop
  Find an approximate solution $\mathbf{x}^{k+1}$ to (5.5), such that ▷ BQP subproblem

$$\left\|\nabla_B \mathcal{L}_A(\mathbf{x}^{k+1}, \boldsymbol{\theta}^k, \rho)\right\| \leq \min\left\{M_{k+1}\|A\,\mathbf{x}^{k+1} - \mathbf{b}\|, \eta\right\};$$

  $\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k - \rho\left(A\,\mathbf{x}^{k+1} - \mathbf{b}\right)$;
  if $(M_{k+1} = M_k) \wedge \left(\mathcal{L}_A(\mathbf{x}^{k+1}, \boldsymbol{\theta}^{k+1}, \rho) < \mathcal{L}_A(\mathbf{x}^k, \boldsymbol{\theta}^k, \rho) + \frac{\rho}{2}\|A\,\mathbf{x}^{k+1} - \mathbf{b}\|\right)$ **then**
    $M_{k+2} = \vartheta M_{k+1}$; ▷ Tighten precision control
  else
    $M_{k+2} = M_{k+1}$;
  end if
  $k = k + 1$;
end while

Theorem 5.1.1. *Let H be symmetric and positive definite. Let $\{\mathbf{x}^k\}$, $\{\boldsymbol{\theta}^k\}$ and $\{M_k\}$ be generated by the SMALBE-M framework for the solution of problem* (1.1)*, with $\eta > 0$, $0 < \vartheta < 1$, $M_0 > 0$, $\rho > 0$, and $\boldsymbol{\theta}^0 \in \mathbb{R}^m$. Then the following statements hold.*

(i) The sequence $\{\mathbf{x}^k\}$ converges to the solution $\mathbf{x}^*$ to (1.1)*.*

(ii) If Assumption 3.1.2 holds, then the sequence $\{\boldsymbol{\theta}^k\}$ converges to a uniquely determined vector $\boldsymbol{\theta}^*$ of Lagrange multipliers for the equality constraints in (1.1)*.*

Moreover, it can be proved that the number of outer iterations needed by SMALBE-M to satisfy a given tolerance has a uniform bound which is independent on the conditioning of the constraint matrix A. In the case of strictly convex problems, under Assumption 3.1.2, it can be proved [58] that the framework is able to identify free and binding variables in a finite number of iterations and that, after the identification, it possesses an R-linear rate of convergence, provided that the algorithm used for the solution of (5.5) has R-linear rate of convergence.

5.2 Preliminary tests on elliptic model problems

We consider the problem of finding the solution $v : \mathbb{R}^2 \to \mathbb{R}$ to the following elliptic differential problem in $\Sigma = [0, 2] \times [0, 2]$

$$\begin{cases} \Delta v \equiv \dfrac{\partial^2 v}{\partial x^2} + \dfrac{\partial^2 v}{\partial y^2} = f(x,y), & (x,y) \in \Sigma \\ v(x,y) = 0, & (x,y) \in \partial\Sigma, \\ l(x,y) \leq v(x,y) \leq u(x,y), & (x,y) \in \Sigma, \end{cases}$$

representing the equilibrium of a membrane, fixed at the boundary of the square Σ, and pushed against the obstacles l (below) and u (above) by the force $f : \mathbb{R}^2 \to \mathbb{R}$.

By means of a finite differences discretization scheme, we can find a discretized version $\mathbf{v}$ of v by solving the QP problem

$$\begin{array}{ll} \min & \frac{1}{2}\mathbf{v}^T H\mathbf{v} - \mathbf{f}^T\mathbf{v} \\ \text{s.t.} & C\mathbf{v} = \mathbf{0}, \\ & \mathbf{l} \leq \mathbf{v} \leq \mathbf{u}, \end{array}$$

where H is the Laplace matrix, $\mathbf{f}$, $\mathbf{l}$ and $\mathbf{u}$ are the discretized version of respectively f, l and u, and the constraints $C\mathbf{v} = \mathbf{0}$ impose the condition $v_i = 0$ for all the i corresponding to points onto $\partial\Sigma$. We selected two different instances of the problem and tested the SMALBE-M framework equipped with P2GP [51] (Algorithm 4.1), and MPRGP [65] Algorithm 5.2.

Algorithm 5.2 MPRGP (Modified Proportioning with Reduced Gradient Projection)

$\mathbf{x}^0 \in \Omega$; $tol > 0$; $\overline{\alpha} \in (0, 2\|H\|^{-1})$; $\Gamma > 0$;
$k = 0$; $\mathbf{g} = \nabla f(\mathbf{x}^0) = H\,\mathbf{x}^0 - \mathbf{c}$; $\mathbf{p} = \boldsymbol{\varphi}(\mathbf{x}^0)$;
while $(\|\nabla_\Omega f(\mathbf{x}^k)\| > tol)$ **do** ▷ Main loop
  if $\left(\|\boldsymbol{\beta}(\mathbf{x}^k)\|^2 \leq \Gamma^2\, \widetilde{\boldsymbol{\varphi}}(\mathbf{x}^k)^T \boldsymbol{\varphi}(\mathbf{x}^k)\right)$ **then**
    $\alpha_{cg} = \dfrac{\mathbf{g}^T\mathbf{p}}{\mathbf{p}^T H\,\mathbf{p}}$, $\mathbf{y} = \mathbf{x}^k - \alpha_{cg}\mathbf{p}$;
    $\alpha_f = \max\{\alpha |\ \mathbf{x}^k - \alpha\mathbf{p} \in \Omega\}$;
    if $(\alpha_{cg} \leq \alpha_f)$ **then**
      $\mathbf{x}^{k+1} = \mathbf{y}$; $\mathbf{g} = \mathbf{g} - \alpha_{cg} H\,\mathbf{g}$; ▷ CG step
      $\gamma = \dfrac{\boldsymbol{\varphi}(\mathbf{y})^T H\,\mathbf{p}}{\mathbf{p}^T H\,\mathbf{p}}$; $\mathbf{p} = \boldsymbol{\varphi}(\mathbf{y}) - \gamma\mathbf{p}$;
    else
      $\mathbf{x}^{(k+\frac{1}{2})} = \mathbf{x}^k - \alpha_f\mathbf{p}$; $\mathbf{g} = \mathbf{g} - \alpha_f H\,\mathbf{p}$; ▷ Feasible halfstep
      $\mathbf{x}^{k+1} = \mathbf{x}^{(k+\frac{1}{2})} - \overline{\alpha}\,\widetilde{\boldsymbol{\varphi}}(\mathbf{x}^{(k+\frac{1}{2})})$; ▷ Expansion step
      $\mathbf{g} = H\,\mathbf{x}^{k+1} - \mathbf{b}$; $\mathbf{p} = \boldsymbol{\varphi}(\mathbf{x}^{k+1})$;
    end if
  else
    $\mathbf{d} = \boldsymbol{\beta}(\mathbf{x}^k)$; $\alpha_{cg} = \dfrac{\mathbf{g}^T\mathbf{d}}{\mathbf{d}^T H\,\mathbf{d}}$; ▷ Proportioning step
    $\alpha_{fcg} = \min\{\max\{\alpha |\ \mathbf{x}^k - \alpha\mathbf{d} \in \Omega\}, \alpha_{cg}\}$;
    $\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_{fcg}\mathbf{d}$; $\mathbf{g} = \mathbf{g} - \alpha_{fcg} H\,\mathbf{d}$; $\mathbf{p} = \boldsymbol{\varphi}(\mathbf{x}^{k+1})$;
  end if
  $k = k + 1$;
end while

For all the testes the discretization was based on a 640×640 2D grid, the tolerance was set as $1e-4\,\|\mathbf{f}\|$, $\rho = 1$, $M_0 = 1$, and $\vartheta = 0.5$.

Test 1

The first test is characterized by the following definitions

$$\begin{cases} f(x,y) = 5\pi^2 \sin(\pi\,x) \sin\left(\dfrac{\pi}{2} y\right), \\ l(x,y) = 0.1\, \sin\left(16\pi\,x - \dfrac{\pi}{6}\right) \sin\left(16\pi\,y - \dfrac{\pi}{6}\right) - 0.2, \\ u(x,y) = 0.1\, \cos\left(16\pi\,x - \dfrac{\pi}{6}\right) \cos\left(16\pi\,y - \dfrac{\pi}{6}\right) + 0.2. \end{cases}$$

The 2D solution to the problem is shown in Figure 5.1 together with its section at $y = 1$. The performances of the two algorithms in the solution of the problem

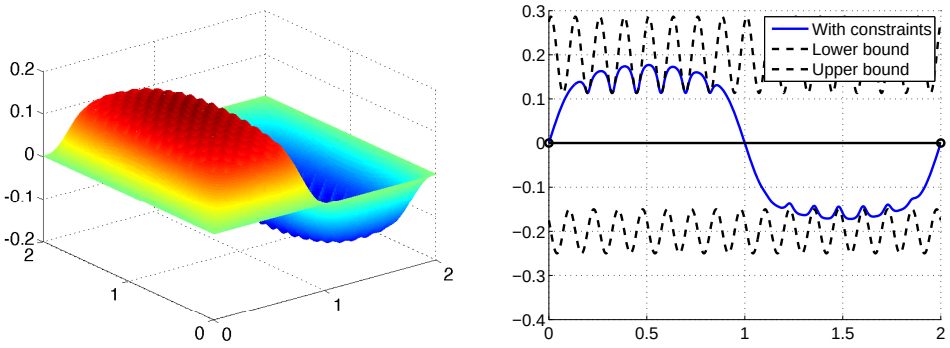

Figure 5.1. First 2-dimensional membrane equilibrium test (*left*). Section of the solution and the lower and upper bounds at $y = 1$ (*right*).

are reported iteration-wise in Table 5.1 and Table 5.2. Both algorithms were able

Table 5.1. Progress of SMALBE-M/MPRGP in the solution of first membrane test.

k	$\mathcal{L}_A(\mathbf{v}^k, \boldsymbol{\theta}^k, \rho)$	$\|\nabla_B \mathcal{L}_A(\mathbf{v}^k, \boldsymbol{\theta}^k, \rho)\|$	$\#H\mathbf{v}$	M_k	$\|\mathcal{F}(\mathbf{v}^k)\|/\|\mathcal{A}(\mathbf{v}^k)\|$
0	0.0	1.93 e-2	52	1.00	410881/0
1	2.688479 e+1	7.44	183	1.00	391164/19717
2	−1.081119	2.82 e-1	200	1.00	387807/23074
3	−1.150512	1.45 e-1	194	5.00 e-1	392628/18253
4	−1.163713	4.18 e-3	159	5.00 e-1	392955/17926
5	−1.163988	4.12 e-2	132	2.50 e-1	393172/17709
6	−1.165628	1.46 e-4	130	2.50 e-1	393363/17518
7	−1.165633	1.13 e-4	127	1.25 e-1	393356/17525
8	−1.165635	3.20 e-5	121	1.25 e-1	393358/17523
9	−1.165635	3.82 e-6	100	6.25 e-2	393359/17522
10	−1.165635	1.91 e-6	44	6.25 e-2	393359/17522

Table 5.2. Progress of SMALBE-M/P2GP in the solution of first membrane test.

k	$\mathcal{L}_A(\mathbf{v}^k, \boldsymbol{\theta}^k, \rho)$	$\|\nabla_B \mathcal{L}_A(\mathbf{v}^k, \boldsymbol{\theta}^k, \rho)\|$	$\#H\mathbf{v}$	M_k	$\|\mathcal{F}(\mathbf{v}^k)\|/\|\mathcal{A}(\mathbf{v}^k)\|$
0	0.0	1.93 e-2	52	1.00	410881/0
1	−1.164921	2.57 e-3	203	1.00	392957/17924
2	−1.165634	3.73 e-4	211	1.00	393357/17524
3	−1.165635	1.45 e-6	246	5.00 e-1	393359/17522

to reach the solution, however the SMALBE-M equipped with MPRGP required 11 outer iterations against the 4 required by SMALBE-M/P2GP, with an overall amount of 1442 Hessian-vector products against the 712 of the combination with P2GP.

Test 2

The second test is characterized by the following definitions

$$\begin{cases} f(x,y) = -5\pi^2 \sin\left(\frac{\pi}{2}x\right) \sin\left(\frac{\pi}{2}y\right), \\ l(x,y) = 0.01 \sin\left(\frac{\pi}{2}x\right) \sin(\pi\, y) - 0.1, & x \in [0,1) \\ l(x,y) = 0.1 \sin\left(16\pi\, x - \frac{\pi}{6}\right) \sin\left(16\pi\, y - \frac{\pi}{6}\right) - 0.2, & x \in [1,2] \\ u(x,y) = 0.2. \end{cases}$$

The 2D solution to the problem is shown in Figure 5.1 together with its section at $y = 1$. The performances of the two algorithms in the solution of the problem are

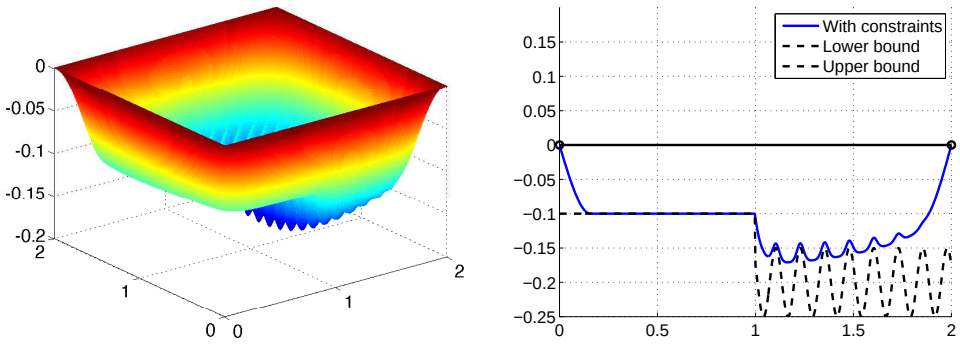

Figure 5.2. Second 2-dimensional membrane equilibrium test (*left*). Section of the solution and the lower and upper bounds at $y = 1$ (*right*).

reported iteration-wise in Table 5.3 and Table 5.4.

Again, both algorithms were able to reach the solution and the choice of P2GP led to a lower computational cost. Indeed the SMALBE-M equipped with MPRGP required 15 outer iterations and 2395 matrix-vector products against the respectively 5 and 811 required by SMALBE-M/P2GP.

In the solution of these two model problems P2GP clearly outperformed MPRGP as a solver for the BQP subproblems in SMALBE-M, probably thanks to its ability to grab or release multiple constraints during the identification phase. This conjecture is enforced by the second problem which was built to have a large number of variables at their lower bound at the solution (~ 146000 against the ~ 17500 of the first test). In the following sections we will see how the two algorithms perform in the solution of more complicated contact mechanic problems.

5.3 Discretization of contact mechanics problems

The aim of this section is to give an idea on how the discretization of contact mechanic problems, in the case of 2D problems (possibly with friction) and in the case of 3D problems, leads to the solution of problems of the form (1.1). Clearly a detailed formulation of contact problems is out of the scope of this work, therefore we refer the reader, e.g., to [63, Chapter 11] on which the content of this section

Table 5.3. Progress of SMALBE-M/MPRGP in the solution of second membrane test.

k	$\mathcal{L}_A(\mathbf{v}^k, \boldsymbol{\theta}^k, \rho)$	$\|\nabla_B \mathcal{L}_A(\mathbf{v}^k, \boldsymbol{\theta}^k, \rho)\|$	$\#H\mathbf{v}$	M_k	$\|\mathcal{F}(\mathbf{v}^k)\|/\|\mathcal{A}(\mathbf{v}^k)\|$
0	0.0	1.93 e-2	52	1.00	410881/0
1	−9.682160 e-1	1.85 e-1	192	1.00	232720/178161
2	−1.057361	5.91 e-3	183	1.00	249705/161176
3	−1.073283	2.57 e-2	170	5.00 e-1	260410/150471
4	−1.068916	1.04 e-1	196	5.00 e-1	262160/148721
5	−1.074556	1.01 e-3	197	5.00 e-1	262690/148191
6	−1.074600	1.46 e-2	198	2.50 e-1	263193/147688
7	−1.074744	9.95 e-3	198	2.50 e-1	263631/147250
8	−1.074804	3.36 e-4	191	1.25 e-1	263873/147008
9	−1.074807	3.55 e-5	180	1.25 e-1	263900/146981
10	−1.074808	4.22 e-5	161	6.25 e-2	263922/146959
11	−1.074809	3.06 e-5	155	6.25 e-2	263970/146911
12	−1.074809	2.46 e-5	149	3.13 e-2	263965/146916
13	−1.074809	1.40 e-5	141	3.13 e-2	263982/146899
14	−1.074809	1.84 e-6	32	1.56 e-2	264000/146881

Table 5.4. Progress of SMALBE-M/P2GP in the solution of second membrane test.

k	$\mathcal{L}_A(\mathbf{v}^k, \boldsymbol{\theta}^k, \rho)$	$\|\nabla_B \mathcal{L}_A(\mathbf{v}^k, \boldsymbol{\theta}^k, \rho)\|$	$\#H\mathbf{v}$	M_k	$\|\mathcal{F}(\mathbf{v}^k)\|/\|\mathcal{A}(\mathbf{v}^k)\|$
0	0.0	1.93 e-2	52	1.00	410881/0
1	−1.069534	3.45 e-3	204	1.00	273216/137665
2	−1.074795	3.25 e-4	204	1.00	264030/146851
3	−1.074809	2.02 e-5	204	5.00 e-1	263987/146894
4	−1.074809	1.42 e-6	147	5.00 e-1	263996/146885

is based. To ease the description we will focus on the frictionless Hertz problem, named after Heinrich Hertz who in 1882 solved the contact problem of two elastic bodies with curved surfaces.

5.3.1 The frictionless Hertz problem

Consider two bodies Ω^1 and Ω^2 of different materials, disposed as in Figure 5.3, whose surface will be indicated respectively as Γ^1 and Γ^2. Each body has a part of the boundary, namely Γ^1_C and Γ^2_C which can enter in contact with the other body (to ease the description we will here consider that the every point of Γ^1_C and Γ^2_C can get into contact with a point of the opposite surface). A pressure P is applied on the upper surface of the second body, i.e. on the part of its boundary indicated as Γ^2_F. The lower surface of the first body is fixed on the ground, which will translate into a Dirichlet boundary condition on $\Gamma^1_U \subset \Gamma^1$. After the deformation due to P, each point $\mathbf{x}^i \in \Omega^i \cup \Gamma^i$ is transformed into the point

$$\mathbf{y}^i(\mathbf{x}^i) = \mathbf{x}^i + \mathbf{u}^i(\mathbf{x}^i),$$

where $\mathbf{u}^i = \mathbf{u}^i(\mathbf{x}^i)$ is the displacement vector defining the deformation of Ω^i. A non-penetration condition is enforced, i.e. given a point $\mathbf{x}^i \in \Gamma^i_C$ it has to be

$\mathbf{y}^i(\mathbf{x}^i) \notin \Omega^j\,(j \neq i)$. It can be shown that the non-penetration condition can be restated in a more computation-friendly version by considering a bijective continuous map

$$\chi^{1,2} : \Gamma_C^1 \to \Gamma_C^2$$

and imposing that

$$\left(\mathbf{u}^1(\mathbf{x}) - \mathbf{u}^2 \circ \chi^{1,2}(\mathbf{x})\right)^T \mathbf{n}^1(\mathbf{x}) \leq \left(\chi^{1,2}(\mathbf{x}) - \mathbf{x}\right)^T \mathbf{n}^1(\mathbf{x}), \quad \mathbf{x} \in \Gamma_C^1, \tag{5.8}$$

where $\mathbf{n}^1(\mathbf{x})$ is an approximation of the outer unit normal to Γ^1 after the deformation. Condition (5.8) is known as the *(strong) linearized non-penetration condition.*

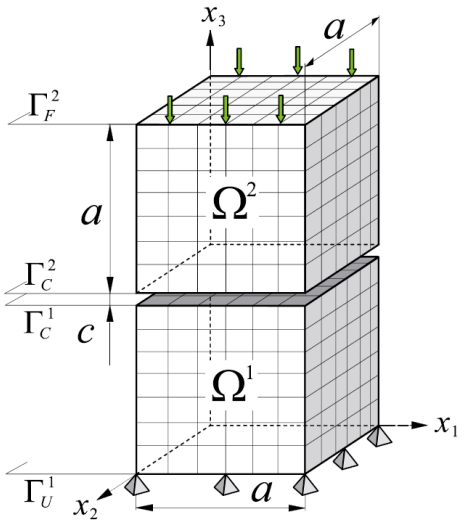

Figure 5.3. The 3D Hertz problem setting [61].

For simplicity, we assume that the bodies are made of an isotropic linear elastic material so that the constitutive equation for the *Cauchy stress tensor* σ is given in terms of the *fourth-order Hooke elasticity tensor* C, defined as

$$C_{ijkl} = \frac{E}{1+\nu}\left(\frac{\nu}{1-2\nu}\delta_{ij}\delta_{kl} + \delta_{ik}\delta_{jl}\right), \quad i,j,k,l = 1,2,3,$$

where E is the *Young's modulus*, ν is the *Poisson ratio* and δ_{rs} indicates the Kronecker delta and the *Cauchy's small strain tensor* $\varepsilon(\mathbf{v})$, defined componentwise as

$$\varepsilon_{ij}(\mathbf{v}) = \frac{1}{2}\left(\frac{\partial v_j}{\partial x_i} + \frac{\partial v_i}{\partial x_j}\right), \quad i,j = 1,2,3.$$

The stress tensor $\sigma(\mathbf{v})$ is thus defined componentwise as

$$\sigma_{ij}(\mathbf{v}) = (C\,\varepsilon(\mathbf{v})) = \sum_{k,l=1}^{3} C_{ijkl}\varepsilon_{kl}(\mathbf{v}), \quad i,j = 1,2,3.$$

Given the volume forces $\mathbf{f}^i : \Omega^i \to \mathbb{R}^3$, the zero boundary displacements $\mathbf{u}_\Gamma^1 : \Gamma_U^1 \to \{\mathbf{0}\}$ and the boundary traction $\mathbf{f}_\Gamma^i : \Gamma_F^2 \to \mathbb{R}^3$, the *linearized elastic equilibrium condition*

and the Dirichlet and Neumann boundary conditions for the displacement $\mathbf{u}$ can be written as

$$\begin{array}{rcll} -\operatorname{div}\sigma(\mathbf{u}) & = & \mathbf{f} & \text{in } \Omega, \\ \mathbf{u}^1 & = & \mathbf{0} & \text{on } \Gamma_U^1, \\ \sigma(\mathbf{u}^2)\mathbf{n}^2 & = & \mathbf{f}_\Gamma^2 & \text{on } \Gamma_F^2, \end{array} \tag{5.9}$$

where $\mathbf{n}^i$ is the outer unit normal to Γ^i. Having assumed that the contact is frictionless, the *surface traction* $\boldsymbol{\tau} = -\sigma(\mathbf{u}^1)\mathbf{n}^1$ on the contact interface Γ_C^1 has a null tangential component, i.e. $\boldsymbol{\tau} = \boldsymbol{\tau}_N = \left(\boldsymbol{\tau}^T\mathbf{n}^1\right)\mathbf{n}^1$, and we can write the linearized conditions of equilibrium as

$$\begin{gathered} \boldsymbol{\tau}^T\mathbf{n}^1 \geq 0, \\ \left(\boldsymbol{\tau}^T\mathbf{n}^1\right)\left(\left(\mathbf{u}^1(\mathbf{x}) - \mathbf{u}^2 \circ \chi^{1,2}(\mathbf{x})\right)^T \mathbf{n}^1(\mathbf{x}) - \mathbf{g}(\mathbf{x})\right) = 0, \quad \mathbf{x} \in \Gamma_C^1, \end{gathered} \tag{5.10}$$

where $\mathbf{g}(\mathbf{x}) = \left(\chi^{1,2}(\mathbf{x}) - \mathbf{x}\right)^T \mathbf{n}^1(\mathbf{x})$, with the second condition known as the *complementarity condition*. Since, according to Newton's third law, the normal traction acting on the two contacting surfaces is equal and opposite, we have that

$$-\sigma\left(\mathbf{u}^2 \circ \chi^{1,2}(\mathbf{x})\right)\mathbf{n}^1 = -\lambda, \quad \mathbf{x} \in \Gamma_C^1. \tag{5.11}$$

The system of equation constituted by (5.9), (5.10) and (5.11) constitutes the *classical formulation of two-body frictionless contact problems*.

5.3.2 The TFETI domain decomposition method

Real-life contact mechanics problems usually involve a larger number of domains with a much more complicated structure, this clearly makes the solution of the discretized problem very expensive. One way to deal with large problems is to tear the domains in smaller parts, allowing to separate the problem into the solution of a set of smaller problems with the cost of introducing additional constraints to "glue" the solution of the subproblems into the solution of the original one.

Focusing on the case of the Hertz problem represented in Figure 5.3, we can decompose each body (together with its boundaries) into subdomains of size H, as shown in Figure 5.4, assign each subdomain a unique number $i \in \{1, \ldots, s\}$. Similarly to what has been done for the case of contact surfaces, we will denote with Γ_G^{ij} the part of the boundary of Ω^i which is "glued" to the boundary of Ω^j; clearly $\Gamma_U^{ij} = \Gamma_U^{ji}$, for all (i,j) and $\Gamma_U^{ij} = \emptyset$ if the subdomains Ω^i and Ω^j are not adjacent components of the same original domain. We therefore introduce the *gluing conditions*

$$\begin{gathered} \mathbf{u}^i(\mathbf{x}) = \mathbf{u}^j(\mathbf{x}), \quad \mathbf{x} \in \Gamma_U^{ij}, \\ \sigma\left(\mathbf{u}^i\right)\mathbf{n}^i = -\sigma\left(\mathbf{u}^j\right)\mathbf{n}^j. \end{gathered} \tag{5.12}$$

After introducing regular grids with discretization parameter h in the subdomains Ω^i, so that they match across the interfaces between the subdomains, by indexing contiguously the nodes and entries of corresponding vectors in the subdomains, and using a Lagrangian finite element discretization, the solution of the Hertz problem

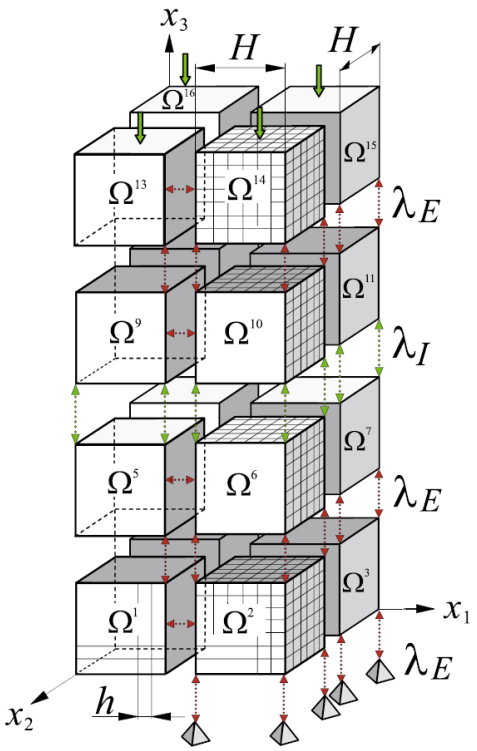

Figure 5.4. The 3D Hertz problem tearing and interconnecting [61].

turns into the solution of the quadratic programming problem

$$\begin{aligned} \min \quad & \frac{1}{2}\mathbf{u}^T K\mathbf{u} - \mathbf{u}^T\mathbf{f} \\ \text{s.t.} \quad & B_I\mathbf{u} \le \mathbf{c}_I, \\ & B_E\mathbf{u} = \mathbf{c}_E, \end{aligned} \tag{5.13}$$

where

$$K = \operatorname{diag}(K_1, \ldots, K_s) \in \mathbb{R}^{n\times n}$$

is a symmetric positive semidefinite block-diagonal matrix, where each diagonal block corresponds to one of the s subdomains. The matrices $B_I \in \mathbb{R}^{m_I\times n}$ and $B_E \in \mathbb{R}^{m_E\times n}$ are both full-rank matrices and describe, together with $\mathbf{c}_I \in \mathbb{R}^{m_I}$ and $\mathbf{c}_E \in \mathbb{R}^{m_E}$, respectively the linearized non-penetration conditions and the zero displacements on the part of the boundary with imposed Dirichlet's condition and the gluing conditions between the faces of the subdomains. It can be assumed, w.l.o.g., that the rows of the matrix $B = \left(B_E^T, B_I^T\right)^T$ are orthonormal; this can be achieved provided that each node is involved in at most one inequality. Finally $\mathbf{f} \in \mathbb{R}^n$ represents the force acting on each node of the discretization, arising either from the volume forces or from some other imposed traction.

5.3.3 Dual formulation

The Lagrangian function associated with (5.13) is

$$\mathcal{L}(\mathbf{u}, \boldsymbol{\theta}_I, \boldsymbol{\theta}_E) = \frac{1}{2}\mathbf{u}^T K\mathbf{u} - \mathbf{f}^T\mathbf{u} + \boldsymbol{\theta}_I^T(B_I\mathbf{u} - \mathbf{c}_I) + \boldsymbol{\theta}_E^T(B_E\mathbf{u} - \mathbf{c}_E),$$

where we denoted with $\boldsymbol{\theta}_I$ and $\boldsymbol{\theta}_E$ the Lagrange multipliers associated respectively with the inequality and the equality constraints.

By introducing the *Moore-Penrose inverse* of K, i.e. the matrix K^+ such that $KK^+K = K$, by indicating as R the orthonormal matrix which spans the kernel of K, and by defining

$$\boldsymbol{\theta} = [\boldsymbol{\theta}_I^T, \boldsymbol{\theta}_E^T]^T, \quad \mathbf{c} = [\mathbf{c}_I^T, \mathbf{c}_E^T]^T,$$

we have that, given a fixed $\overline{\boldsymbol{\theta}}$, the minimizer $\mathbf{u}$ of $\mathcal{L}(\mathbf{u}, \overline{\boldsymbol{\theta}}) \equiv \mathcal{L}(\mathbf{u}, \boldsymbol{\theta}_I, \boldsymbol{\theta}_E)$ satisfies

$$K\,\mathbf{u} - \mathbf{f} + B^T\boldsymbol{\theta} = \mathbf{0},$$

which can be verified if and only if

$$\mathbf{f} - B^T\boldsymbol{\theta} \in \mathrm{Im}(K), \quad \Leftrightarrow \quad R^T\left(\mathbf{f} - B^T\boldsymbol{\theta}\right) = \mathbf{0}.$$

Moreover, it can be shown that there exists a vector $\boldsymbol{\theta}$ such that

$$\mathbf{u} = K^+\left(\mathbf{f} - B^T\boldsymbol{\theta}\right) + R\,\boldsymbol{\theta}.$$

Therefore, the solution to (5.13) can be found by solving the problem

$$\begin{array}{ll} \min & \Theta(\boldsymbol{\theta}) = \dfrac{1}{2}\boldsymbol{\theta}^T F\boldsymbol{\theta} - \mathbf{r}^T\boldsymbol{\theta} \\ \text{s.t.} & \boldsymbol{\theta}_I \geq \mathbf{0}, \\ & G\,\boldsymbol{\theta} = \mathbf{h}, \end{array} \tag{5.14}$$

where $F = BK^+B^T$, $\mathbf{r} = (BK^+\mathbf{f} - \mathbf{c})$, $G = R^TB^T$ and $\mathbf{h} = R^T\mathbf{f}$.

5.3.4 Formulation in presence of friction

We have seen how the discretization of frictionless problems leads to problems of the form (1.1), however in the solution of real-life problems there is usually the need to take friction into account. Since well-known Coulomb law of friction leads to non-convex formulation, a different friction model due to Henri Tresca is often considered. This model, which however violates the laws of physics, assumes that the normal pressure is a priori known on the contact interface. Despite its theoretical flaws, it is often taken into account, thanks to the fact that it leads to convex well-posed problems and its solution can be used in the fixed-point iterations for the solution of problems with Coulomb's friction [63, Chapter 12].

From the point of view of the classical formulation, taking into account Tresca friction introduces constraints involving the tangential component of the surface traction $\boldsymbol{\tau}_T = \boldsymbol{\tau} - \boldsymbol{\tau}_N$, which will no more be null, and the tangential displacement $\mathbf{u}_T = \mathbf{u} - \mathbf{u}_N$. It can be shown (see, e.g., [62]) that this induces the presence in the dual formulation of constraints of the form

$$\|[\boldsymbol{\theta}_F]_i\| \leq \Psi_i$$

where $\Psi_i > 0$ and each dual variable $[\boldsymbol{\theta}_F]_i$ is a scalar in the case of 2D problems and a vector in $\mathbb{R}^2$ in the case of 3D problems. In the former case this leads to imposing lower and upper bounds on the dual variables, thus leading again to a problem of

the form (1.1). In the 3D case, instead, it leads to separable circular constraints, making the dual problem a Quadratically Constrained Quadratic Programming problem (QCQP). The peculiar structure of the quadratic constraints makes it cheap to compute a projection onto the feasible set; therefore, this class of problems can be solved by means of specific generalization of the algorithms considered in this chapter [23, 22]. Since the solution of QCQPs is out of the scope of this work, we consider only frictionless 3D problems in our numerical experiments.

5.4 Numerical experiments

We considered three test problems, one stationary 2D contact problem with friction and two 3D frictionless contact problems. As discussed in the previous section, in all the cases the problem is reformulated as the solution of a problem of form (1.1), which can be solved by means of the SMALBE-M framework introduced in the previous sections. We performed the test for this section (derived from [66]), using MATLAB implementation of SMALBE-M, MPRGP and P2GP.

5.4.1 Results on the 2D beam with material insets problem

Our first benchmark is the 2D beam problem depicted in left part of Figure 5.5, where inside the "soft" ($E = 4.4\,\mathrm{e}{+5}$, $\nu = 0.34$) rectangular beam there are 8 stiff ($E = 1.6\,\mathrm{e}{+7}$, $\nu = 0.32$) circular insets. The whole set of bodies is subject to a force applied to the right side of the soft beam as shown in the figure. The discretization of the problem leads to a problem with 2222 variables, 1024 of which are subject either to lower bounds or to both lower and upper bounds, and 60 linear equality constraints. The difficulty of this problem lies in the need for the iterative solver to distribute the global information across several nonlinear interfaces.

We solved six different instances of the problem, characterized by different choices for the boundary forces (F_x, F_y). The results obtained by solving the problem with the SMALBE-M framework equipped either with P2GP or with MPRGP are summarized in Table 5.5. In particular, the number of variables which are on a bound ($|\mathcal{A}|$), number of outer iterations and Hessian multiplications are shown.

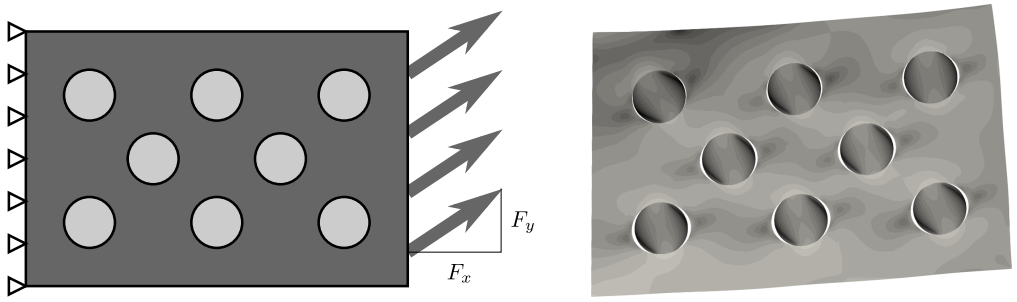

Figure 5.5. 2D beam with insets setting (*left*) and Huber-von Mises-Hencky stress (*right*).

			MPRGP			
F_x	F_y	$\|\mathcal{A}\|$	$\|\nabla_B \mathcal{L}_A\|$	$\|G\boldsymbol{\theta} - \mathbf{h}\|$	`#out_it`	$\#F\mathbf{v}$
100	0	890	2.7 e-14	5.0 e-7	19	1667
75	15	879	5.3 e-11	9.9 e-7	16	877
75	-15	878	5.9 e-13	8.0 e-7	19	1494
-100	0	618	4.1 e-13	5.2 e-8	14	1321
-75	15	667	1.1 e-12	1.6 e-7	14	1380
-75	-15	666	2.5 e-12	3.9 e-7	14	1332

			P2GP			
F_x	F_y	$\|\mathcal{A}\|$	$\|\nabla_B \mathcal{L}_A\|$	$\|G\boldsymbol{\theta} - \mathbf{h}\|$	`#out_it`	$\#F\mathbf{v}$
100	0	890	1.5 e-9	7.0 e-7	15	1298
75	15	879	7.7 e-9	1.3 e-6	18	1701
75	-15	878	4.0 e-9	8.3 e-7	16	1427
-100	0	618	3.8 e-9	9.9 e-7	14	1432
-75	15	667	5.8 e-9	1.1 e-7	16	1625
-75	-15	666	1.6 e-9	6.7 e-7	14	1396

Table 5.5. Test results for SMALBE-M equipped with MPGRP and P2GP on the 6 benchmarks of the 2D beam with insets.

5.4.2 Results on the frictionless Hertz 3D problem

The second benchmark is a 3D two-body contact problem depicted in Figure 5.6 (left). The stiff upper body ($E = 1.6\,\text{e+6}$, $\nu = 0.32$, $\rho = 5.08\,\text{e-9}$) is pressed toward the softer lower one ($E = 4.4\,\text{e+5}$, $\nu = 0.34$, $\rho = 1.04\,\text{e-9}$). The upper body has been divided in $3 \times 3 \times 2$ subdomains, while the lower one has been divided in $3 \times 3 \times 3$ subdomains; each subdomain has been discretized with $10 \times 10 \times 10$ nodes. The straight edges of the two bodies have length $L = 10$. For our test we fixed the radius of the lower body at $r_1 = -50$ which translates into a concave surface and we chose two different radii for the upper body, namely $r_2 = 30$ and $r_2 = 45$. The first problem is characterized by 34854 variables, 900 of them are subject to lower bounds, while the second one is characterized by 34914 variables, 960 of them subject to lower bounds; both problems are subject to 270 linear equality constraints. The problem is not easy as on the solution comprises many dual degenerate components on the boundary of the active contact interface. The performance of both algorithms is summarized in Table 5.6.

5.4.3 Results on the friction 3D ball bearing problem

As last benchmark we chose an example of a real-life application, i.e the 3D multi-body contact problem describing the interaction between the various components of a ball bearing; in particular only a segment of the ball bearing is considered (see the right side of Figure 5.7). The problem has been solved by means of the same variant of the FETI method used for the previous case. The discretization leads to a QP problem characterized by 19976 variables and 120 linear equality constraints.

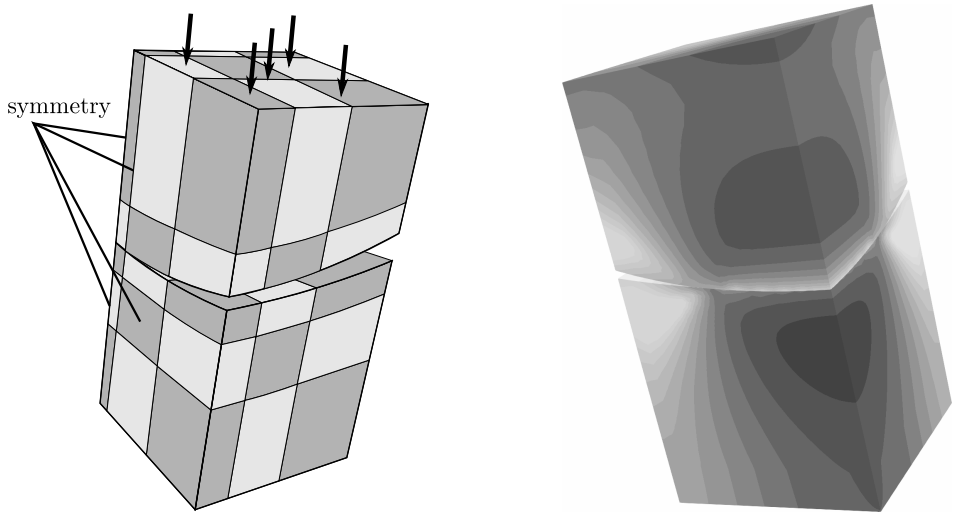

Figure 5.6. Frictionless Hertz 3D setting (*left*) and Huber-von Mises-Hencky stress (*right*).

			MPRGP			
r_2	P	$\lvert\mathcal{A}\rvert$	$\lVert\nabla_B\mathcal{L}_A\rVert$	$\lVert G\boldsymbol{\theta}-\mathbf{h}\rVert$	`#out_it`	$\#F\mathbf{v}$
30	1	856/900	4.6 e-7	3.6 e-6	61	1687
	10	822/900	8.2 e-7	3.6 e-6	28	1562
	100	729/900	3.5 e-6	3.3 e-6	16	1544
	1000	556/900	1.3 e-6	3.6 e-6	12	2447
45	1	887/960	1.5 e-7	5.1 e-7	27	1617
	10	821/960	4.6 e-7	6.0 e-7	18	1713
	100	659/960	1.6 e-7	5.4 e-7	13	2778
	1000	324/960	9.4 e-7	8.5 e-7	11	2496

			P2GP			
r_2	P	$\lvert\mathcal{A}\rvert$	$\lVert\nabla_B\mathcal{L}_A\rVert$	$\lVert G\boldsymbol{\theta}-\mathbf{h}\rVert$	`#out_it`	$\#F\mathbf{v}$
30	1	856/900	4.1 e-7	3.4 e-6	78	1954
	10	822/900	9.4 e-7	3.7 e-6	25	2103
	100	729/900	3.3 e-6	3.2 e-6	17	2176
	1000	556/900	1.4 e-6	2.2 e-6	14	3461
45	1	887/960	1.5 e-7	5.2 e-7	34	1900
	10	821/960	5.1 e-7	5.8 e-7	20	2306
	100	659/960	3.3 e-7	6.1 e-7	13	3301
	1000	324/960	1.1 e-6	9.7 e-7	13	3622

Table 5.6. Test results for SMALBE-M equipped with MPGRP and P2GP on the 8 instances of the frictionless 3D Hertz problem.

The active set at the optimal solution consists of 1199 variables among the 1212 subject to a lower bound. In this case the framework equipped with P2GP as inner solver took 51 iterations for the solution of the problem, with a total amount of 849 Hessian multiplications, thus outperforming the framework equipped with MPRGP which took 60 iterations and a total amount of 1188 Hessian products.

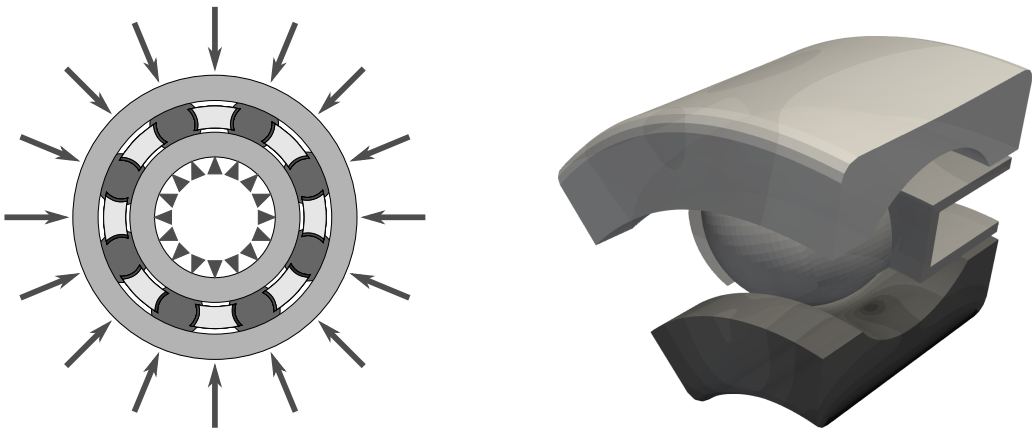

Figure 5.7. Ball bearing setting (*left*) and displacement stress (*right*).

The result of the performed tests show that in some cases P2GP appears to be competitive with MPRGP, which is a standard choice for contact problems, and is sometimes able to outperform it. These tests confirmed in some sense our conjecture that the "aggressive" strategy of P2GP for the expansion of the active set performs better in problems where the percentage of active constraints is higher (see for example the case of the ball bearing and the case of the 2D beam). On the other hand, the results in Table 5.6 indicate that MPRGP is more efficient in treating problems with many dual degenerate components of the solution. It is likely that a suitable combination of both algorithms can result in still faster solver.

Conclusions and future work

In this work we dealt with subspace accelerated gradient projection methods for the solution of quadratic programming problems.

We proposed a reformulation of the stationarity conditions for problems of the form (1.1), which allowed us to introduce a novel active-set framework, called *Proportionality-based Subspace Accelerated framework for Quadratic Programming* (PSAQP), for the solution of QPs. PSAQP alternates two phases: an identification phased, based on gradient projection steps, and a minimization phase, based on an unconstrained minimization onto the reduced subspace defined by the current active set. We introduced a criterion to switch between the two phases based on a comparison between a measure of optimality within the reduced space (called free gradient) and a measure of the quality of the current active set (called chopped gradient). From the theoretical point of view, a nice consequence of using this criterion is that finite convergence for strictly convex problems can be proved even in case of degeneracy at the solution.

A novel method for the solution of BQPs and SLBQPs, called *Proportionality-based 2-phase Gradient Projection* (P2GP), is proposed as a specialization of PSAQP. P2GP may be seen as a generalization of the GPCG method by Moré and Toraldo [109] to a wider class of problems. The most distinguishing feature of P2GP with respect to GPCG stands in the criterion used to stop the minimization phase. This is a critical issue, since requiring high accuracy in this phase can be a useless and time-consuming task when the face where a solution lies is far from being identified. Other important novelties are the ability to deal with SLBQPs, the introduction of BB-like step lengths in the identification phase and the possible use of spectral gradient methods in the minimization phase. The numerical tests reported in Section 4.3 show a strong improvement of the computational performance when the proportionality criterion is used to control the termination of the minimization phase. In particular, the comparison of P2GP with a modification of GPCG (using the same BB-like step lengths as P2GP in the identification phase) shows the clear superiority of P2GP and its smaller sensitivity to the Hessian condition number. Thus, proportionality allows one to handle the minimization phase in a more clever way than switching criteria based on heuristics as the one involving the bindingness of the active set used in GPCG. The numerical results also show that P2GP requires many fewer projections than efficient GP methods like PABB_{min} and the one implemented in BLG [89]. This leads to a significant time saving, especially when the Hessian matrix is sparse or has a structure that allows the computation of the matrix-vector product with a computational cost smaller than $\mathcal{O}(n^2)$, where n is the size of the problem. We also introduced a novel procedure for the creation of SLBQPs and BQPs with

different sizes, spectral properties and levels of degeneracy, which can be used as a benchmark to test optimization algorithms for these classes of problems. The MATLAB code implementing P2GP and the test problem generator is available from `https://github.com/diserafi/P2GP`.

The encouraging numerical results obtained for P2GP in the solution of both SLBQPs and BQPs led us to test it as a computational kernel in Augmented Lagrangian methods for the solution of some problems arising in contact mechanics. The results, reported in Chapter 5 show that P2GP is competitive with MPRGP [65], which is tailored for the solution of contact mechanics problems, and it is sometimes able to outperform it. The test performed on randomly generated problems have proved P2GP to be efficient when a good approximation to the solution is required. This could suggest that P2GP could not be the ideal choice in the context of Augmented Lagrangian methods, where a rough approximation to the subproblems' solutions, especially in the initial steps, can still lead to a fast convergent scheme in practice. However, the results on the elliptic model problems and the test performed on the 3D ball bearing problem allow us to conjecture that, thanks to the use of gradient projection steps, which are able to add/remove multiple constraints at a time to/from the active set, P2GP could perform better in problems where the percentage of active constraints at the solution is higher. From the physical point of view, since active constraints in the dual formulation are related to inactive contact points in the primal formulation, these scenarios are related to the cases in which only small regions of the contact surfaces of the bodies are effectively in contact at the solution.

An interesting feature of PSAQP is that it provides a general framework, allowing different step length rules in the GP steps, and different methods in the minimization phase. The encouraging theoretical and computational results suggest that this framework deserves to be further investigated. An implementation for general QPs is currently under study. As noted in Section 3.4.1, a crucial aspect for an efficient implementation are the algorithm for the projection onto the polyhedron and its tangent cone and the algorithm used to compute the estimate of the Lagrange multipliers needed in the computation of the free and the chopped gradients. The latter could also be useful for the introduction of the proposed Lagrange multiplier estimate in Augmented Lagrangian frameworks for the solution of QPs. This, together with possible extensions to more general optimization problems, will be the object of future works.

Bibliography

[1] Akaike, H. On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method. *Annals of the Institute of Statistical Mathematics*, **11** (1959), 1.

[2] Amaral, S., Allaire, D. L., and Willcox, K. Optimal L_2-norm empirical importance weights for the change of probability measure. *Statistics and Computing*, **27** (2017), 625.

[3] Andreani, R., Birgin, E. G., Martínez, J. M., and Schuverdt, M. L. On augmented Lagrangian methods with general lower-level constraints. *SIAM Journal on Optimization*, **18** (2007), 1286.

[4] Antonelli, L., De Simone, V., and di Serafino, D. On the application of the spectral projected gradient method in image segmentation. *Journal of Mathematical Imaging and Vision*, **54** (2016), 106.

[5] Arreckx, S., Lambe, A., Martins, J. R. R. A., and Orban, D. A matrix-free augmented lagrangian algorithm with application to large-scale structural design optimization. *Optimization and Engineering*, **17** (2016), 359.

[6] Barzilai, J. and Borwein, J. M. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, **8** (1988), 141.

[7] Benvenuto, F., Zanella, R., Zanni, L., and Bertero, M. Nonnegative least-squares image deblurring: improved gradient projection approaches. *Inverse Problems*, **26** (2010), 025004 (18 pp.).

[8] Benzi, M., Golub, G. H., and Liesen, J. Numerical solution of saddle point problems. *Acta Numerica*, **14** (2005), 1.

[9] Bertero, M., Lanteri, H., and Zanni, L. Iterative image reconstruction: a point of view. In *Mathematical Methods in Biomedical Imaging and Intensity-Modulated Radiation Therapy (IMRT)* (edited by M. Jiang, Y. Censor, and A. Louis), vol. 7 of *CRM*, pp. 37–63. Edizioni della Normale, Pisa (2008).

[10] Bertsekas, D. On the Goldstein-Levitin-Polyak gradient projection method. *IEEE Transactions on Automatic Control*, **21** (1976), 174.

[11] Bertsekas, D. Projected Newton methods for optimization problems with simple constraints. *SIAM Journal on Control and Optimization*, **20** (1982), 221.

[12] Bertsekas, D. *Convex Optimization Algorithms*. Athena Scientific (2015). ISBN 9781886529281.

[13] Bertsekas, D. P. *Nonlinear Programming*. Athena Scientific, Belmont, MA, USA (1999).

[14] Bielschowsky, R. H., Friedlander, A., Gomes, F. A. M., and Martínez, J. M. An adaptive algorithm for bound constrained quadratic minimization. *Investigacion Operativa*, **7** (1997), 67.

[15] Birgin, E. G. and Martínez, J. M. Large-scale active-set box-constrained optimization method with spectral projected gradients. *Computational Optimization and Applications*, **23** (2002), 101.

[16] Birgin, E. G., Martínez, J. M., and Raydan, M. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, **10** (2000), 1196.

[17] Birgin, E. G., Martínez, J. M., and Raydan, M. Spectral projected gradient methods: Review and perspectives. *Journal of Statistical Software, Articles*, **60** (2014), 1.

[18] Björck, Å. *Numerical methods for least squares problems*. SIAM, Philadelphia, PA, USA (1996).

[19] Boggs, P. T. and Tolle, J. W. Sequential quadratic programming for large-scale nonlinear optimization. *Journal of Computational and Applied Mathematics*, **124** (2000), 123 . Numerical Analysis 2000. Vol. IV: Optimization and Nonlinear Equations.

[20] Bonettini, S. and Prato, M. New convergence results for the scaled gradient projection method. *Inverse Problems*, **31** (2015), 095008.

[21] Bonettini, S., Zanella, R., and Zanni, L. A scaled gradient projection method for constrained image deblurring. *Inverse Problems*, **25** (2009), 015002.

[22] Bouchala, J., Dostál, Z., Kozubek, T., Pospíšil, L., and Vodstrčil, P. On the solution of convex QPQC problems with elliptic and other separable constraints with strong curvature. *Applied Mathematics and Computation*, **247** (2014), 848 .

[23] Bouchala, J., Dostál, Z., and Vodstrčil, P. Separable spherical constraints and the decrease of a quadratic function in the gradient projection step. *Journal of Optimization Theory and Applications*, **157** (2013), 132.

[24] Brucker, P. An O(n) algorithm for quadratic knapsack problems. *Operations Research Letters*, **3** (1984), 163 .

[25] Buchheim, C., Santis, M., Lucidi, S., Rinaldi, F., and Trieu, L. A feasible active set method with reoptimization for convex quadratic mixed-integer programming. *SIAM Journal on Optimization*, **26** (2016), 1695.

[26] Caffarelli, L. A. and Friedman, A. The free boundary for elastic-plastic torsion problems. *Transactions of the American Mathematical Society*, **252** (1979), 65.

[27] Calamai, P. H. and Moré, J. J. Projected gradient methods for linearly constrained problems. *Mathematical Programming*, **39** (1987), 93.

[28] Calamai, P. H. and Moré, J. J. Quasi-Newton updates with bounds. *SIAM Journal on Numerical Analysis*, **24** (1987), 1434.

[29] Cauchy, A.-L. Méthode générale pour la résolution des systèmes d'équations simultanées. *Compte Rendu des Séances de L'Académie des Sciences XXV*, **Série A** (1847), 536.

[30] Cesarone, F., Scozzari, A., and Tardella, F. A new method for mean-variance portfolio optimization with cardinality constraints. *Annals of Operations Research*, **205** (2013), 213.

[31] Cominetti, R., Mascarenhas, W. F., and Silva, P. J. S. A Newton's method for the continuous quadratic knapsack problem. *Mathematical Programming Computation*, **6** (2014), 151.

[32] Conn, A. R., Gould, N. I., and Toint, P. A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Numerical Analysis*, **28** (1991), 545.

[33] Conn, A. R., Gould, N. I. M., and Toint, P. L. *Lancelot: A Fortran Package for Large-Scale Nonlinear Optimization (Release A)*. No. 17 in Springer Series in Computational Mathematics (1992). ISBN 3-540-55470-X (Berlin, Heidelberg), 0-387-55470-X (New York).

[34] Conn, A. R., Gould, N. I. M., and Toint, P. L. *Trust-region methods*. MPS-SIAM Series on Optimization, Philadelphia, PA (2000).

[35] Cornelio, A., Porta, F., Prato, M., and Zanni, L. On the filtering effect of iterative regularization algorithms for discrete inverse problems. *Inverse Problems*, **29** (2013), 125013.

[36] Curtis, F. E. and Guo, W. Handling nonpositive curvature in a limited memory steepest descent method. *IMA Journal of Numerical Analysis*, **36** (2016), 717.

[37] Dai, Y.-H. and Fletcher, R. Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming. *Numerische Mathematik*, **100** (2005), 21.

[38] Dai, Y.-H. and Fletcher, R. New algorithms for singly linearly constrained quadratic programs subject to lower and upper bounds. *Mathematical Programming, Series A*, **106** (2006), 403.

[39] Dai, Y.-H., Hager, W. W., Schittkowski, K., and Zhang, H. The cyclic Barzilai-Borwein method for unconstrained optimization. *IMA Journal of Numerical Analysis*, **26** (2006), 604.

[40] Dai, Y.-H. and Liao, L.-Z. R-linear convergence of the Barzilai and Borwein gradient method. *IMA Journal of Numerical Analysis*, **22** (2002), 1.

[41] Dai, Y.-H. and Yang, X. A new gradient method with an optimal stepsize property. *Computational Optimization and Applications*, **33** (2006), 73.

[42] D'Apuzzo, M., De Simone, V., and di Serafino, D. On mutual impact of numerical linear algebra and large-scale optimization with focus on interior point methods. *Computational Optimization and Applications*, **45** (2010), 283.

[43] Davis, T. A. and Hager, W. W. A sparse proximal implementation of the LP dual active set algorithm. *Mathematical Programming*, **112** (2008), 275.

[44] De Angelis, P. L. and Toraldo, G. On the identification property of a projected gradient method. *SIAM Journal on Numerical Analysis*, **30** (1993), 1483.

[45] De Asmundis, R., di Serafino, D., Hager, W. W., Toraldo, G., and Zhang, H. An efficient gradient method using the Yuan steplength. *Computational Optimization and Applications*, **59** (2014), 541.

[46] De Asmundis, R., di Serafino, D., and Landi, G. On the regularizing behavior of the SDA and SDC gradient methods in the solution of linear ill-posed problems. *Journal of Computational and Applied Mathematics*, **302** (2016), 81 .

[47] De Asmundis, R., di Serafino, D., Riccio, F., and Toraldo, G. On spectral properties of steepest descent methods. *IMA Journal of Numerical Analysis*, **33** (2013), 1416.

[48] Dembo, R. S. and Tulowitzki, U. *On the minimization of quadratic functions subject to box constraints.* Yale University, Department of Computer Science (1984).

[49] di Serafino, D., Ruggiero, V., Toraldo, G., and Zanni, L. A note on spectral properties of some gradient methods. In *Numerical Computations: Theory and Algorithms (NUMTA-2016)*, vol. 1776 of *AIP Conference Proceedings*, p. 040003 (2016).

[50] di Serafino, D., Ruggiero, V., Toraldo, G., and Zanni, L. On the steplength selection in gradient methods for unconstrained optimization. *Applied Mathematics and Computation*, **318** (2018), 176.

[51] di Serafino, D., Toraldo, G., Viola, M., and Barlow, J. A two-phase gradient method for quadratic programming problems with a single linear constraint and bounds on the variables. *SIAM Journal on Optimization*, **28** (2018), 2809.

[52] Dolan, E. D. and Moré, J. J. Benchmarking optimization software with performance profiles. *Mathematical Programming, Series B*, **91** (2002), 201.

[53] Dostál, Z. Box constrained quadratic programming with proportioning and projections. *SIAM Journal on Optimization*, **7** (1997), 871.

[54] Dostál, Z. A proportioning based algorithm with rate of convergence for bound constrained quadratic programming. *Numerical Algorithms*, **34** (2003), 293.

[55] Dostál, Z. Inexact semimonotonic augmented Lagrangians with optimal feasibility convergence for convex bound and equality constrained quadratic programming. *SIAM Journal on Numerical Analysis*, **43** (2005), 96.

[56] Dostál, Z. An optimal algorithm for bound and equality constrained quadratic programming problems with bounded spectrum. *Computing*, **78** (2006), 311.

[57] Dostál, Z. *Optimal quadratic programming algorithms: with applications to variational inequalities*, vol. 23. Springer Science & Business Media (2009).

[58] Dostál, Z., Brzobohatý, T., Horák, D., Kozubek, T., and Vodstrčil, P. On R-linear convergence of semi-monotonic inexact augmented Lagrangians for bound and equality constrained quadratic programming problems with application. *Computers & Mathematics with Applications*, **67** (2014), 515 .

[59] Dostál, Z., Friedlander, A., and Santos, S. Augmented Lagrangians with adaptive precision control for quadratic programming with simple bounds and equality constraints. *SIAM Journal on Optimization*, **13** (2003), 1120. ,

[60] Dostál, Z., Horák, D., and Kučera, R. Total feti—an easier implementable variant of the feti method for numerical solution of elliptic PDE. *Communications in Numerical Methods in Engineering*, **22** (2006), 1155. ,

[61] Dostál, Z., Kozubek, T., Brzobohatý, T., Markopoulos, A., and Vlach, O. Scalable tfeti with optional preconditioning by conjugate projector for transient frictionless contact problems of elasticity. *Computer Methods in Applied Mechanics and Engineering*, **247-248** (2012), 37 .

[62] Dostál, Z., Kozubek, T., Horyl, P., Brzobohatý, T., and Markopoulos, A. A scalable tfeti algorithm for two-dimensional multibody contact problems with friction. *Journal of Computational and Applied Mathematics*, **235** (2010), 403 . Special Issue on Advanced Computational Algorithms.

[63] Dostál, Z., Kozubek, T., Sadowská, M., and Vondrák, V. *Scalable Algorithms for Contact Problems*, vol. 36. Springer (2017).

[64] Dostál, Z. and Pospíšil, L. Minimizing quadratic functions with semidefinite Hessian subject to bound constraints. *Computers and Mathematics with Applications*, **70** (2015), 2014.

[65] Dostál, Z. and Schöberl, J. Minimizing quadratic functions subject to bound constraints with the rate of convergence and finite termination. *Computational Optimization and Applications*, **30** (2005), 23.

[66] Dostál, Z., Toraldo, G., Viola, M., and Vlach, O. Proportionality-based gradient methods with applications in contact mechanics. In *High Performance Computing in Science and Engineering* (edited by T. Kozubek, M. Čermák, P. Tichý, R. Blaheta, J. Šístek, D. Lukáš, and J. Jaroš), pp. 47–58. Springer International Publishing, Cham (2018). ISBN 978-3-319-97136-0.

[67] Dunn, J. Global and asymptotic convergence rate estimates for a class of projected gradient processes. *SIAM Journal on Control and Optimization*, **19** (1981), 368.

[68] Figueiredo, M., Nowak, R., and Wright, S. Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems. *IEEE Journal of Selected Topics in Signal Processing*, **1** (2007), 586.

[69] Fletcher, R. On the Barzilai–Borwein method. In *Optimization and Control with Applications* (edited by L. Qi, K. Teo, X. Yang, P. M. Pardalos, and D. Hearn), vol. 96 of *Applied Optimization*, pp. 235–256. Springer (2005).

[70] Fletcher, R. A limited memory steepest descent method. *Mathematical Programming, Series A*, **135** (2012), 413.

[71] Fletcher, R. Augmented Lagrangians, box constrained QP and extensions. *IMA Journal of Numerical Analysis*, **37** (2017), 1635. ,

[72] Floudas, C. A. and Visweswaran, V. *Quadratic Optimization*, pp. 217–269. Springer US, Boston, MA (1995). ISBN 978-1-4615-2025-2.

[73] Frassoldati, G., Zanni, L., and Zanghirati, G. New adaptive step-size selections in gradient methods. *Journal of Industrial and Management Optimization*, **4** (2008), 299.

[74] Friedlander, A. and Martínez, J. M. On the numerical solution of bound constrained optimization problems. *RAIRO-Operations Research*, **23** (1989), 319.

[75] Friedlander, A. and Martínez, J. M. On the maximization of a concave quadratic function with box constraints. *SIAM Journal on Optimization*, **4** (1994), 177.

[76] Friedlander, A., Martínez, J. M., Molina, B., and Raydan, M. Gradient method with retards and generalizations. *SIAM Journal on Numerical Analysis*, **36** (1999), 275.

[77] Friedlander, A., Martínez, J. M., and Raydan, M. A new method for large-scale box constrained convex quadratic minimization problems. *Optimization Methods and Software*, **5** (1995), 57.

[78] Gill, P. E. and Murray, W. *Numerical methods for constrained optimization*. Academic Press London (1974). ISBN 9780122835506.

[79] Goldstein, A. A. Convex programming in hilbert space. *Bulletin of the American Mathematical Society*, **70** (1964), 709.

[80] Golub, G. H. and Van Loan, C. F. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA (1996). ISBN 0-8018-5414-8.

[81] Gondzio, J. and Grothey, A. Parallel interior-point solver for structured quadratic programs: Application to financial planning problems. *Annals of Operations Research*, **152** (2007), 319.

[82] Gonzalez-Lima, M. D., Hager, W. W., and Zhang, H. An affine-scaling interior-point method for continuous knapsack constraints with application to support vector machines. *SIAM Journal on Scientific Computing*, **21** (2011), 361.

[83] Gould, N. I. M., Hribar, M. E., and Nocedal, J. On the solution of equality constrained quadratic programming problems arising in optimization. *SIAM Journal on Optimization*, **23** (2001), 1376.

[84] Grippo, L., Lampariello, F., and Lucidi, S. A nonmonotone line search technique for Newton's method. *SIAM Journal on Numerical Analysis*, **23** (1986), 707.

[85] Grippo, L. and Sciandrone, M. Nonmonotone globalization techniques for the Barzilai-Borwein gradient method. *Computational Optimization and Applications*, **23** (2002), 143.

[86] Hager, W. and Zhang, H. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on Optimization*, **16** (2005), 170.

[87] Hager, W. W. The dual active set algorithm and its application to linear programming. *Computational Optimization and Applications*, **21** (2002), 263.

[88] Hager, W. W. and Hungerford, J. T. Continuous quadratic programming formulations of optimization problems on graphs. *European Journal of Operational Research*, **240** (2015), 328 .

[89] Hager, W. W. and Zhang, H. A new active set algorithm for box constrained optimization. *SIAM Journal on Optimization*, **17** (2006), 526.

[90] Hager, W. W. and Zhang, H. An active set algorithm for nonlinear optimization with polyhedral constraints. *Science China Mathematics*, **59** (2016), 1525.

[91] Hager, W. W. and Zhang, H. Projection onto a polyhedron that exploits sparsity. *SIAM Journal on Optimization*, **26** (2016), 1773.

[92] Hansen, P. C. *Rank-Deficient and Discrete Ill-Posed Problems. Numerical Aspects of Linear Inversion.* SIAM, Philadelphia, PA, USA (1998).

[93] Held, M., Wolfe, P., and Crowder, H. P. Validation of subgradient optimization. *Mathematical Programming*, **6** (1974), 62.

[94] Helgason, R., Kennington, J., and Lall, H. A polynomially bounded algorithm for a singly constrained quadratic program. *Mathematical Programming*, **18** (1980), 338.

[95] Kamesam, P. and Meyer, R. *Multipoint methods for separable nonlinear networks.* Springer (1984).

[96] Kiwiel, K. C. On linear-time algorithms for the continuous quadratic knapsack problem. *Journal of Optimization Theory and Applications*, **134** (2007), 549.

[97] Kiwiel, K. C. Breakpoint searching algorithms for the continuous quadratic knapsack problem. *Mathematical Programming*, **112** (2008), 473.

[98] Kiwiel, K. C. Variable fixing algorithms for the continuous quadratic knapsack problem. *Journal of Optimization Theory and Applications*, **136** (2008), 445.

[99] Lantéri, H., Roche, M., and Aime, C. Penalized maximum likelihood image restoration with positivity constraints: multiplicative algorithms. *Inverse Problems*, **18** (2002), 1397.

[100] Levitin, E. S. and Polyak, B. T. Constrained minimization methods. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, **6** (1966), 787.

[101] Lin, Y. and Cryer, C. An alternating direction implicit algorithm for the solution of linear complementarity problems arising from free boundary problems. *Applied Mathematics and Optimization*, **13** (1985), 1.

[102] Loli Piccolomini, E., Coli, V. L., Morotti, E., and Zanni, L. Reconstruction of 3D X-ray CT images from reduced sampling by a scaled gradient projection algorithm. *Computational Optimization and Applications*, **71** (2018), 171.

[103] Lötstedt, P. Numerical simulation of time-dependent contact and friction problems in rigid body mechanics. *SIAM Journal on Scientific & Statistical Computing*, **5** (1984), 370.

[104] Luss, H. and Gupta, S. K. Technical note—allocation of effort resources among competing activities. *Operations Research*, **23** (1975), 360.

[105] McCormick, G. P. Anti-zig-zagging by bending. *Management Science*, (1969), 315.

[106] Michelot, C. A finite algorithm for finding the projection of a point onto the canonical simplex of $\mathbb{R}^n$. *Journal of Optimization Theory and Applications*, **50** (1986), 195.

[107] Mohy-ud-Din, H. and Robinson, D. P. A solver for nonconvex bound-constrained quadratic optimization. *SIAM Journal on Optimization*, **25** (2015), 2385.

[108] Moré, J. and Toraldo, G. Algorithms for bound constrained quadratic programming problems. *Numerische Mathematik*, **55** (1989), 377.

[109] Moré, J. J. and Toraldo, G. On the solution of large quadratic programming problems with bound constraints. *SIAM Journal on Optimization*, **1** (1991), 93.

[110] Nocedal, J., Sartenaer, A., and Zhu, C. On the behavior of the gradient norm in the steepest descent method. *Computational Optimization and Applications*, **22** (2002), 5.

[111] Nocedal, J. and Wright, S. *Numerical Optimization*. Springer-Verlag, New York (2006). Second edition.

[112] Pardalos, P. and Schnitger, G. Checking local optimality in constrained quadratic programming is NP-hard. *Operations Research Letters*, **7** (1988), 33 .

[113] Pardalos, P. M. and Kovoor, N. An algorithm for a singly constrained class of quadratic programs subject to upper and lower bounds. *Mathematical Programming*, **46** (1990), 321.

[114] Pardalos, P. M. and Rosen, J. B. *Constrained global optimization: algorithms and applications*. Springer-Verlag, New York, NY, USA (1987).

[115] Pardalos, P. M. and Xue, J. The maximum clique problem. *Journal of Global Optimization*, **4** (1994), 301.

[116] Parlett, B. N. *The Symmetric Eigenvalue Problem*. SIAM, Philadelphia, PA, USA (1998). ,

[117] Porta, F., Prato, M., and Zanni, L. A new steplength selection for scaled gradient methods with application to image deblurring. *Journal of Scientific Computing*, **65** (2015), 895.

[118] Raydan, M. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM Journal on Optimization*, **7** (1997), 26.

[119] Raydan, M. and Svaiter, B. F. Relaxed Steepest Descent and Cauchy-Barzilai-Borwein Method. *Computational Optimization and Applications*, **21** (2002), 155.

[120] Serafini, T., Zanghirati, G., and Zanni, L. Gradient projection methods for quadratic programs and applications in training support vector machines. *Optimization Methods and Software*, **20** (2005), 353.

[121] Shor, N. Z. *Minimization methods for non-differentiable functions*, vol. 3. Springer Science & Business Media (2012).

[122] Vapnik, V. N. and Kotz, S. *Estimation of dependences based on empirical data*, vol. 40. Springer-Verlag, New York, NY, USA (1982).

[123] Wright, S. J. Implementing proximal point methods for linear programming. *Journal of Optimization Theory and Applications*, **65** (1990), 531.

[124] Wright, S. J., Nowak, R. D., and Figueiredo, M. A. T. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, **57** (2009), 2479.

[125] Yuan, Y. A new stepsize for the steepest descent method. *Journal of Computational Mathematics*, **24** (2006), 149.

[126] Zanella, R., Boccacci, P., Zanni, L., and Bertero, M. Efficient gradient projection methods for edge-preserving removal of Poisson noise. *Inverse Problems*, **25** (2009), 045010.

[127] Zanni, L. An improved gradient projection-based decomposition technique for support vector machines. *Computational Management Science*, **3** (2006), 131.