UNIVERSITÀ DEGLI STUDI DI STUDI DI MODENA E REGGIO EMILIA

Dottorato di Ricerca in Matematica in convenzione con l'Università degli Studi di Ferrara e l'Università degli Studi di Parma

Ciclo XXXII

SPECTRAL PROPERTIES OF GRADIENT-BASED METHODS FOR OPTIMIZATION PROBLEMS WITH SPECIAL CONSTRAINTS

Candidato: SERENA CRISCI

Relatore: PROF. VALERIA RUGGIERO

Coordinatore del Corso di Dottorato: PROF. CRISTIAN GIARDINÀ

If you find that you're spending almost all your time on theory,
start turning some attention to practical things;
it will improve your theories.
If you find that you're spending almost all your time on practice,
start turning some attention to theoretical things;
it will improve your practice.

— Donald E. Knuth

ACKNOWLEDGEMENTS

This thesis represents the final result of the work carried out during the doctoral program at the University of Ferrara. Looking back over the past three years, I can undoubtedly say that this experience has been a real milestone in my personal as well as professional growth. This would not have been possible without the support of many people who have put great trust in me and encouraged me in this exciting, but sometimes hard, journey.

First of all, I would like to express my sincere gratitude and deep admiration to Prof. Valeria Ruggiero, as my research advisor. Working closely with her gave me the chance to appreciate her valuable skills as a mathematician, her depth of thought and dedication to work. A true lesson I learned from her: never stop studying. I would like to thank Prof. Luca Zanni, talking with him has always been inspiring and extremely clarifying for me; his contribution to the research included in this thesis has been essential. I thank Dr. Federica Porta, with whom I started to collaborate last year, and Prof. Gaetano Zanghirati for his precious piece of advice and his way of being always supportive, conveying positivity. I also thank Prof. Gerardo Toraldo for taking care of me, even from a distance.

A special thank goes to all the kind persons I met during my short-term visit at the Dept. of Applied Mathematics of the Technical University of Ostrava. Indeed, I am really grateful to Prof. Zdeněk Dostál and his team, especially Dr. David Horák, Marek Pecha and Jakub Kružík. Some of the numerical experiments I made there are included in the thesis. I hope that our cooperation will continue in the future.

I would also like to thank all those who, for various reasons, have accompanied me in this doctoral route, the OASIS research group, people met at conferences and schools, the university staff. I especially remember the meaningful discussions and moments of real fun shared with my office fellows, namely Elisa Miragliotta, Ngyuen Thi Ngoc Giao, Giorgio Menegatti, Leonardo Spinosa, Francesco Galuppi, Sara Durighetto, Sara Bagossi, Elsa Corniani, Matteo Bergami, and other dear colleagues in Modena, i.e. Simone Rebegoldi, Carla Bertocchi, Giorgia Franchini. The greatest thank to my beloved family.

PUBLICATIONS

The thesis is based on the ideas appeared in the following contributions:

Accepted journal article:

S. Crisci, F. Porta, V. Ruggiero, L. Zanni. Spectral properties of Barzilai-Borwein rules in solving singly linearly constrained optimization problems subject to lower and upper bounds. *SIAM Journal of Optimization*, in press.

S. Crisci, V. Ruggiero, L. Zanni. Steplength selection in gradient projection methods for box-constrained quadratic programs. *Applied Mathematics and Computation*, 356, pp. 312 – 327, 2019.

Conference Proceeding:

S. Crisci, F. Porta, V. Ruggiero, L. Zanni. A Limited Memory Gradient Projection Method for box-constrained quadratic optimization problems. In: Sergeyev Y., Kvasov D. (eds) Numerical Computations: Theory and Algorithms. NUMTA 2019. Lecture Notes in Computer Science, vol 11973. Springer, Cham.

CONTENTS

INTRODUCTION

In the last years, first-order methods received renewed attention, due to the growing demand in solving large-scale optimization problems arising in many important real-world applications (e. g., imaging, signal processing, machine learning). Their popularity is essentially related to the computational simplicity combined with low cost per iteration and low memory requirements for solving such problems with sufficient (but not high) accuracy.

Gradient methods are first-order iterative algorithms devised for unconstrained optimization of differentiable functions and are based on the use of steps proportional to the negative gradients of the objective function, until a local minimum is approached; hence, they only require the evaluation of the objective function and the corresponding gradient at each iterate. When dealing with constrained problems, such methods are referred to as gradient projection methods; in this case, the additional cost of applying a projection operator to ensure the feasibility of each iterate, must be taken into account; however, in many real applications, the computational effort required to fulfil this task may not be excessively demanding, for example in presence of simple constraints (e. g., bound constraints).

Steplength selection is a fundamental task for improving the efficiency of gradient-based algorithms; it requires to balance the need of accelerating the convergence rate without increasing the overall computational cost. In the last decades, the role of the steplength selection strategies has been widely investigated; indeed, starting from the seminal paper of Barzilai and Borwein (1988), many efficient steplength selection rules have been designed, making the gradient approaches an effective tool for solving optimization problems that handle large-scale data and require real-time solutions. Most of these steplength rules have been conceived for the unconstrained framework, with the aim of exploiting some second-order information to achieve a fast annihilation of the gradient of the objective function in inexpensive way. Nevertheless, these rules have been successfully employed also within gradient projection methods for constrained optimization, without any prior analysis on the possible effects of the constraints on the steplength formulation, at least to our knowledge.

The aim of this thesis is to investigate how the presence of the constraints can affect the spectral properties of some well-known steplength selection rules in gradient projection method for solving constrained optimization problems. We focus, in particular, on two special cases, very frequent in the applications: box-constrained problems and constrained problems subject to a single linear equality con-

straint in addition to box-constraints.
The main contribution of the thesis consists in the spectral analysis and in the introduction of modified versions of the Barzilai-Borwein (BB) rules in presence of special feasible regions. Indeed, we remind that the convergence criteria do not require restrictive hypothesis on the sequence of the steplength parameters, by simply assuming that it is bounded below and above by positive constants; this flexibility in the choice of the steplength can be exploited to develop updating strategies aimed at optimizing the numerical behaviour. Hence, our novel proposals are specifically designed in order to take account of the information related to the first-order optimality conditions. In particular, we prove that the proposed modifications to the steplength selection rules are able to sweep the spectra of suitable submatrices of the Hessian matrix during the iterative procedure, improving the performance of the method. Our analysis is not limited to the BB-like rules but, in the case of box-constraints, it is extended also to the LMSD steplength selection strategy developed by R. Fletcher (2012). Finally, we describe the results of an extensive numerical experimentation, aimed at evaluating the practical effectiveness of the novel strategies on both quadratic and non-quadratic test problems, large-scale problems and real-world applications.

The thesis is organized as follows.

In Chapter 1 we remind the basic schemes of gradient methods for unconstrained optimization and gradient projection methods for constrained optimization, included the variable metric approach known as scaled gradient projection method. Then, the spectral properties of Barzilai-Borwein rules and the LMSD approach are discussed in detail in the context of unconstrained problems. Furthermore, without claiming to be exhaustive, we also provide an overview of some state-of-the-art steplength selection strategies.

The first part of Chapter 2 is dedicated to the spectral analysis of the BB rules in gradient projection methods for solving box-constrained quadratic programming problems. We prove that, when the active constraints start to stabilizing in the iterative procedure, the first Barzilai-Borwein rule (BB1) is able to produce a sequence of steplengths sweeping the spectrum of a proper submatrix of the Hessian matrix, which depends only on the inactive constraints at the current iterate. This property does not hold for the second BB rule (BB2); therefore, our proposal consists in forcing the BB2 rule to mimic the natural behaviour of BB1 by neglecting the quantities that depends on the active constraints. This simple modification helps to recover suitable spectral properties, improving the practical performance of the method, also when it is used within alternating strategies. The resulting rules are named, respectively, BoxBB2 and BoxABB$_{\text{min}}$. The second part of the chapter is devoted to analyse the spectral behaviour of

steplength selection strategies based on the use of Ritz values. This approach was originally proposed by Fletcher within the LMSD method, and successively employed, after slight modifications, within a gradient projection scheme for the box-constrained framework, by Porta et al. (2015). Since the spectral properties of both the mentioned approaches were not inspected, we propose a special analysis aimed at highlighting the relation between subsequent gradients in a sweep. Then we introduce an adaptive strategy, called Box-LMGP2, that allows to set the length of a sweep during the iterative procedure, based on the gradient components that provide feasible steps in a sweep.

In Chapter 3 we extend the analysis developed for the box-constrained case to the case of singly linearly equality constrained minimization problems subject to lower and upper bounds (SLB). In particular, starting from the quadratic framework, we show that the BB1 rule produces a sequence of steplengths whose reciprocals belong to the spectra of special submatrices of the Hessian matrix, corresponding to suitable approximations of the Hessian matrix restricted to the tangent space of the active constraints at the solution. As observed in the box-constrained case, this property does not hold for the BB2 rule and, therefore, we introduce suitable modifications, in order to foster a behaviour similar to that of the BB1 rule. Such modified rule is named EQ-BB2 and is exploited also in the alternating scheme here denominated EQ-ABB$_{\min}$. Furthermore, we prove that the spectral properties of BB1 and EQ-BB2 observed for quadratic programming still hold also in the general non-quadratic case. The chapter ends by reporting a further generalization of the BB rules to a variable metric scheme.

Numerical experiments aimed at evaluating the practical effectiveness of gradient projection scheme combined with the proposed rules are reported in Chapter 4. The set of experiments collected in the first section is devoted to comparing the behaviour of both standard and modified versions of the BB rules and the limited memory approaches on box-constrained quadratic programming problems of medium and large dimensions. The numerical results are coherent with the theoretical analysis developed in Chapter 2, confirming also the higher efficiency achieved by using the BB-based alternating scheme rather than the other rules. The general outcomes obtained in the first part of numerical tests, are confirmed by the two successive sets of experiments concerning, respectively, SLB quadratic programming problems and non-quadratic problems. Furthermore, each section of Chapter 4 contains a special discussion of the results gained on some real-world applications related to Support Vector Machines, journal bearing problem and imaging processing problems.

Finally, in Appendix A, some basic definitions and useful results on projection matrices, used throughout the thesis, are presented for the reader's convenience.

ACRONYMS

ABB Adaptive Barzilai-Borwein [90]

BB Barzilai-Borwein [5]

BQP Box-constrained Quadratic Programming

GA Golden Arcsine [73]

GBB Global Barzilai-Borwein [76]

GLL Grippo-Lampariello-Lucidi [52]

GMR Gradient Methods with Retards [48]

GP Gradient Projection

MPRGP Modified Proportioning with Reduced Gradient Projection [39]

PBB Projected Barzilai-Borwein [27]

PBBF Projected Barzilai-Borwein with fall-back [58]

QP Quadratic Programming

SD Steepest Descent

SDA Steepest Descent with Alignment [34]

SDC Steepest Descent with Constant steplength [35]

SGP Scaled Gradient Projection [15]

SLB Singly Linearly Bound

SLBQP Singly Linearly Bound Quadratic Programming

SPD Symmetric Positive Definite

SPS Symmetric Positive Semi-definite

SVM Support Vector Machine

$\mathcal{M}_{n,m}(\mathbb{R})$ denotes the set of all the n-by-m matrix over $\mathbb{R}$; $\mathcal{M}_{n,n}(\mathbb{R})$ is denoted by $\mathcal{M}_n(\mathbb{R})$.

$I_n \in \mathcal{M}_n(\mathbb{R})$ denotes the identity matrix of order n.

$\mathcal{O}_{r,s}$ denotes the r-by-s matrix with null entries.

If $x, y \in \mathbb{R}^n$, then $x^T y = \sum_{i=1}^{n} x_i y_i$ denotes the scalar product.

$\|\cdot\|$ denotes the Euclidean norm: $\|x\| = \|x\|_2 = \sqrt{x^T x}$.

Given two subsets of indices $\mathcal{I} \subseteq \{1, 2, \ldots, n\}$ and $\mathcal{J} \subseteq \{1, 2, \ldots, m\}$, for any matrix $A \in \mathcal{M}_{n,m}(\mathbb{R})$, the notation $A_{\mathcal{I},\mathcal{J}}$ indicates the submatrix defined by the intersections of the rows and columns with indices in $\mathcal{I}$ and $\mathcal{J}$. Similarly, for any vector $x \in \mathbb{R}^n$, $x_{\mathcal{I}}$ denotes the subvector whose entries have indices in $\mathcal{I}$.

$\|\cdot\|_D$ denotes the norm induced by a symmetric positive definite matrix $D \in \mathcal{M}_n(\mathbb{R})$: $\|x\|_D = \sqrt{x^T D x}$.

Given a symmetric matrix $A \in \mathcal{M}_n(\mathbb{R})$, for any non-zero vector $x \in \mathbb{R}^n$, $R_A(x)$ denotes the Rayleigh quotient of A at x.

1

SPECTRAL ANALYSIS IN GRADIENT-BASED METHODS: A REVIEW OF SOME STATE-OF-THE-ART STEPLENGTH SELECTION STRATEGIES

Gradient methods fall into the broader class of first-order methods, designed to address general optimization problems of type

$$\min_{x \in \Omega} f(x) \tag{1.1}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function over the nonempty subset $\Omega \subseteq \mathbb{R}^n$. The case $\Omega = \mathbb{R}^n$ is referred to as the *unconstrained case*, while if Ω is a subset of $\mathbb{R}^n$, the problem (1.1) is said to be *constrained*. In the latter case, the *feasible region* Ω is generally assumed closed and convex.

Gradient-based methods have been widely used for solving nonlinear programming problems, revealing themselves as the most convenient choice in many large-scale applications, due to their simplicity combined with low iteration cost and low memory requirements.

Starting from an arbitrary initial guess $x^{(0)}$, standard gradient methods for unconstrained minimization generate a sequence of vectors $x^{(1)}, x^{(2)}, \ldots$, in accordance with the general rule:

$$x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}, \qquad k = 0, 1, 2, \ldots, \tag{1.2}$$

where $\alpha_k > 0$ is the *steplength* parameter and, for $\nabla f(x^{(k)}) \neq 0$, the direction $d^{(k)}$ is chosen such that

$$\nabla f(x^{(k)})^T d^{(k)} < 0, \tag{1.3}$$

i.e., $d^{(k)}$ is a *descent direction*. Typically, the iterative process is not finitely convergent; it is terminated when $\|\nabla f(x^{(k)})\| \leqslant \epsilon_1$ or $\|x^{(k+1)} - x^{(k)}\| \leqslant \epsilon_2 \|x^{(k)}\|$, where ϵ_1 and ϵ_2 are small positive scalars. The descent direction (1.3) is usually defined as

$$d^{(k)} = -D_k \nabla f(x^{(k)}), \tag{1.4}$$

where D_k is a symmetric positive definite (SPD) matrix, known as *scaling matrix*. Different methods arise from different choices of the scaling matrix [9]; in this chapter, we will deal, at first, with the simplest choice $D_k = I_n$, $k = 0, 1, 2, \ldots$, which gives rise to the so-called *steepest descent* (SD) method.

Most of the popular gradient-based approaches have the desirable feature of being *iterative descent* algorithms; this means that they generate a sequence of iterates able to foster a decrease in the objective function at each iteration, that is:

$$f(x^{(k+1)}) < f(x^{(k)}), \qquad k = 0, 1, 2, \ldots, \quad . \tag{1.5}$$

Condition (1.5) is typically achieved by imposing further requirements on the choice of the steplength α_k. As a consequence, the effectiveness of gradient schemes strongly relies on appropriate selections of the steplength α_k, as well as the direction $d^{(k)}$.
A classical example of steplength selection is the minimization rule, where the steplength is computed by means of an *exact linesearch* along the steepest descent direction $d^{(k)} = -\nabla f(x^{(k)})$, namely

$$\alpha_k = \arg\min_{\alpha \geqslant 0} f(x^{(k)} + \alpha d^{(k)}). \tag{1.6}$$

The steplength defined by (1.6) is the classical Cauchy steplength [18]. In general, the minimizing steplength cannot be exactly computed in the practice and it turns out to be an inefficient strategy, due to the expensive computations needed for implementing one-dimensional minimization algorithms; furthermore, it often reveals poor practical behaviour and low convergence rate, as pointed out for the first time by Akaike [1], who highlighted the phenomenon of the typical zig-zag pattern exhibited by the Cauchy steepest descent method in the unconstrained quadratic case. As a consequence, most of the subsequent attempts in literature were successfully directed towards inexpensive ways to compute an acceptable steplength instead of the optimal one. The aim of these novel steplength selection rules is to accelerate the steepest descent method, without increasing the computational costs, and avoiding the iterates falling into a two dimensional subspace, which is the reason for the inefficiency of the Cauchy steplength (see e. g. [31, 53, 73, 77, 78]).

Among the wide variety of possibilities for choosing the steplength, we intend to give special attention to steplength selection strategies that exploit spectral properties of the Hessian matrix of the objective function by means of low cost approximations of second-order information, which do not require the explicit computation of the Hessian. These techniques, starting from the inspiring work of Barzilai and Borwein [5], gave new impulse to first-order approaches and contributed to the development of a class of methods known as *spectral gradient methods*, which are still considered a valuable tool in the context of nonlinear unconstrained optimization (see e. g. [25, 26, 30, 33–35, 45, 47, 48, 51, 72, 74, 77, 90]). Many of these steplength selections have been successfully exploited (without any modification to their definitions) also in gradient projection methods for solving constrained optimization problems arising from real-world applications (see e. g. [6, 10, 15, 27, 28, 61, 68, 71, 78]).

This chapter is organized as follows. In Section 1.1 we recall the spectral properties of some well-known steplength selection rules in the context of unconstrained minimization, whereas in Section 1.2 we report the basics of gradient projection methods and some other well-known algorithm for constrained optimization.

1.1 SPECTRAL PROPERTIES IN THE UNCONSTRAINED CASE

This section is devoted to describing some of the most popular steplength selection strategies based on spectral properties of the Hessian matrix. These techniques have been first designed for solving a special instance of the general problem (1.1), given by the following quadratic programming (QP) problem

$$\min_{x\in\mathbb{R}^n} f(x) = \frac{1}{2}x^T Ax - b^T x + c, \tag{1.7}$$

where $b \in \mathbb{R}^n$, $c \in \mathbb{R}$ and $A \in M_n(\mathbb{R})$ is a SPD matrix. The quadratic model (1.7) represents the natural setting for analysing how the updating steplength rules are related to the eigenvalues of the Hessian matrix and provides useful information that may enable to extend the approach to the case of more general non-quadratic optimization problems.
The theoretical motivation behind this approach for developing novel strategies is related to a relevant property possessed by all the gradient methods for the minimization of a strictly convex quadratic function: that is, the solution can be reached in a finite number of iterations by using as steplengths the sequence of the reciprocals of the exact eigenvalues of the Hessian matrix. Indeed, let consider the standard SD iteration

$$x^{(k+1)} = x^{(k)} - \alpha_k g^{(k)} \tag{1.8}$$

applied to the minimizing problem (1.7), where $g^{(i)} = g(x^{(i)})$ with $g(x) = \nabla f(x) = Ax - b$; then a recurrence formula for the gradient can be easily derived:

$$g^{(k+1)} = g^{(k)} - \alpha_k A g^{(k)} = \prod_{j=0}^{k} \left(I_n - \alpha_j A\right) g^{(0)}; \tag{1.9}$$

let suppose, without loss of generality [44], that the matrix A has distinct eigenvalues

$$0 < \lambda_1 < \lambda_2 < \cdots < \lambda_n, \tag{1.10}$$

and $g_i^{(0)} \neq 0$ for all $i = 1, \ldots, n$; by denoting with $\{v_1, v_2, \ldots, v_n\}$ a set of associated orthonormal eigenvectors, which form a basis of $\mathbb{R}^n$, the gradient can be expressed as

$$g^{(k+1)} = \sum_{i=1}^{n} \mu_i^{(k+1)} v_i, \tag{1.11}$$

where $\mu_i^{(k+1)}, i = 1, \dots, n$, are suitable scalars called *eigencomponents* of $g^{(k+1)}$; in particular, since (1.11) holds true for $g^{(0)}$, from (1.9) we have

$$g^{(k+1)} = \sum_{i=1}^{n} \mu_i^{(0)} \left(\prod_{j=0}^{k} (I_n - \alpha_j A) \right) v_i; \tag{1.12}$$

then, pairing the eigencomponents in equations (1.11)-(1.12) and recalling that $Av_i = \lambda_i v_i$, for $i = 1 \dots, n$, the i-th eigencompontent of the gradient recurs according to the following formula

$$\mu_i^{(k+1)} = \mu_i^{(0)} \prod_{j=0}^{k} (1 - \alpha_j \lambda_i) = \mu_i^{(k)} (1 - \alpha_k \lambda_i). \tag{1.13}$$

From (1.13) it follows:

- if $\mu_i^{(k)} = 0$ for some i, then $\mu_i^{(h)} = 0$ for $h \geqslant k$ and the component of the gradient along v_i will be zero at all subsequent iterations;
- if $\alpha_k = 1/\lambda_i$, then $\mu_i^{(k+1)} = 0$.

As a consequence, a finite termination property of the steepest descent method can be deduced: in particular, if the eigenvalues of A are known and the first n steplengths are defined by setting in any order

$$\alpha_{k_j} = \frac{1}{\lambda_j}, \quad j = 1, \dots, n,$$

then the method converges in, at most, n iterations. However, the eigenvalues of A are not usually available. Then, in order to take advantage of the previous results, a lot of efforts has been devoted in designing steplength selection strategies able to generate suitable approximations of the inverses of the eigenvalues of the Hessian. Gradient methods that share the idea of choosing the steplengths based on the spectral properties of the underlying Hessian, rather than the standard decrease in the objective function, are known in literature as *spectral gradient methods*; as first observed by Fletcher in [43], their effectiveness is strictly related to the ability in sweeping the spectrum of the inverse of the Hessian matrix (see also [35, 44, 72, 80, 87]).

1.1.1 *The Barzilai-Borwein rules and their generalizations*

The Barzilai-Borwein (BB) rules arise from the clever intuition of combining quasi-Newton properties with a SD method for the unconstrained minimization of a differentiable function $f(x)$. This technique allows to accelerate the annihilation of the gradient of the objective function by means of a suitable sequence of steplengths α_k able to capture second-order information in an inexpensive way.

The idea of introducing second-order information within a gradient scheme avoiding the explicit computation of the Hessian matrix is typical of quasi-Newton methods. These methods provide a less demanding alternative to standard Newton's method, ensuring a superlinear convergence rate [65]; in particular, they only require the knowledge of two successive iterates and the corresponding gradients for updating, at each step, an approximate Hessian matrix B_k that solves the *secant equation*:

$$B_k s^{(k-1)} = y^{(k-1)}, \tag{1.14}$$

where $s^{(k-1)} = x^{(k)} - x^{(k-1)}$ and $y^{(k-1)} = \nabla f(x^{(k)}) - \nabla f(x^{(k-1)})$. Indeed, condition (1.14) enables to yield curvature information at each step by mimicking the approximate relation that holds for the true Hessian and it is exact for quadratic functions:

$$\nabla^2 f(x^{(k)})(x^{(k)} - x^{(k-1)}) \approx \nabla f(x^{(k)}) - \nabla f(x^{(k-1)}).$$

The BB strategy for the steplength selection adopts this principle to incorporate second-order information within the search direction: in particular, the steplength α_k is determined by minimizing the error on a secant equation, in two alternative ways:

$$\alpha_k = \arg\min_{\alpha} \left\| \alpha^{-1} s^{(k-1)} - y^{(k-1)} \right\|, \tag{1.15}$$

or

$$\alpha_k = \arg\min_{\alpha} \left\| s^{(k-1)} - \alpha y^{(k-1)} \right\|. \tag{1.16}$$

In other words, the BB rules define a gradient method described by the iteration (1.8) where the steplength is obtained by forcing the matrix $(\alpha_k I_n)^{-1}$ to approximate the Hessian of the objective function. From (1.15) and (1.16), respectively, two updating rules for the steplength can be defined:

$$\alpha_k^{BB1} = \frac{\|s^{(k-1)}\|^2}{s^{(k-1)^T} y^{(k-1)}}, \tag{1.17}$$

and

$$\alpha_k^{BB2} = \frac{s^{(k-1)^T} y^{(k-1)}}{\|y^{(k-1)}\|^2}. \tag{1.18}$$

The Barzilai-Borwein rules involved within a gradient scheme define a spectral method. Indeed, in the strictly convex quadratic case (1.7) it can be readily seen that the steplengths (1.17)-(1.18) correspond to the reciprocals of particular Rayleigh quotients of the Hessian matrix,providing suitable approximations of its eigenvalues.

Proposition 1.1 *Let* $f\colon \mathbb{R}^n \to \mathbb{R}$ *be defined as in* (1.7) *with SPD matrix* A *and let* $\{x^{(k)}\}_{k\in\mathbb{N}}$ *be the sequence generated by a gradient method of the form* (1.8)*. Then the BB rules* (1.17)-(1.18) *can be rewritten, respectively, as follows:*

$$\alpha_k^{BB1} = \frac{g^{(k-1)^T} g^{(k-1)}}{g^{(k-1)^T} A g^{(k-1)}} = \left(R_A(g^{(k-1)})\right)^{-1}, \tag{1.19}$$

$$\alpha_k^{BB2} = \frac{g^{(k-1)^T} A g^{(k-1)}}{g^{(k-1)^T} A^2 g^{(k-1)}} = \left(R_A(A^{1/2} g^{(k-1)})\right)^{-1}, \tag{1.20}$$

where $g^{(k-1)} = \nabla f(x^{k-1})$*. Furthermore, if* λ_1 *and* λ_n *denote, respectively, the minimum and the maximum eigenvalues of* A*, then the following inequality holds*

$$\lambda_1 \leqslant \frac{1}{\alpha_k^{BB1}} \leqslant \frac{1}{\alpha_k^{BB2}} \leqslant \lambda_n. \tag{1.21}$$

Proof: from the gradient scheme (1.8) we can write

$$s^{(k-1)} = -\alpha_k g^{(k-1)}. \tag{1.22}$$

Similarly, the recurrence property (1.9) gives

$$y^{(k-1)} = -\alpha_k A g^{(k-1)}. \tag{1.23}$$

Replacing the relations (1.22)-(1.23) in (1.17), we have

$$\alpha_k^{BB1} = \frac{(-\alpha_k g^{(k-1)})^T(-\alpha_k g^{(k-1)})}{(-\alpha_k g^{(k-1)})^T(-\alpha_k A g^{k-1})} = \frac{g^{(k-1)^T} g^{(k-1)}}{g^{(k-1)^T} A g^{(k-1)}};$$

thus the identity (1.19) is proved. Applying a similar argument to α_k^{BB2} (1.18), we obtain (1.20). From the extremal properties of the Rayleigh quotients (see [55], Theorem 4.2.2), it follows that $\frac{1}{\alpha_k^{BB1}} = \frac{g^{(k-1)^T} A g^{(k-1)}}{g^{(k-1)^T} g^{(k-1)}} \geqslant \lambda_1$ and $\frac{1}{\alpha_k^{BB2}} = \frac{(g^{(k-1)^T} A^{1/2})A(A^{1/2} g^{(k-1)})}{(g^{(k-1)^T} A^{1/2})(A^{1/2} g^{(k-1)})} \leqslant \lambda_n$. Finally, from the Cauchy-Schwarz inequality it follows: $s^{(k-1)^T} y^{(k-1)} \leqslant \|s^{(k-1)}\| \|y^{k-1}\|$, which implies

$$\begin{aligned} \alpha_k^{BB2} &= \frac{s^{(k-1)^T} y^{(k-1)}}{\|y^{(k-1)}\|^2} \leqslant \frac{\|s^{(k-1)}\| \|y^{(k-1)}\|}{\|y^{(k-1)}\|^2} = \frac{\|s^{(k-1)}\|}{\|y^{(k-1)}\|} \leqslant \\ &\leqslant \frac{\|s^{(k-1)}\|^2}{\|s^{(k-1)}\| \|y^{(k-1)}\|} \leqslant \frac{\|s^{(k-1)}\|^2}{s^{(k-1)^T} y^{(k-1)}} = \alpha_k^{BB1}. \end{aligned}$$

□

We remark that, if $g^{(k-1)}$ is different from both the eigenvectors associated with the eigenvalues λ_1 and λ_n, then the steplengths α_k defined by the BB rules satisfy

$$\lambda_1 < \frac{1}{\alpha_k} < \lambda_n. \tag{1.24}$$

In [5] the authors established a first convergence result of the gradient scheme equipped with BB rules in the two-dimensional strictly convex quadratic case; in particular, in this special case, they proved that the gradient method (1.8) with the steplength defined by (1.18) converges R-superlinearly with R-order equal to $\sqrt{2}$. This surprising result, however, cannot be generalized to the n-dimensional case, where only R-linear convergence can rather be expected. In particular, in 1993, Raydan [75] proved the convergence of the BB method for any-dimensional strictly convex quadratic problem and this result was later refined by Dai and Liao [29], who established the following result.

Theorem 1.1 *[29, Thm. 2.5] Let $f(x)$ be a strictly convex quadratic function. Let $\{x^{(k)}\}_{k\in\mathbb{N}}$ be the sequence generated by the gradient method (1.8) with the steplength defined by (1.17) (or, equivalently, (1.18)). Then, either $g^{(k)} = 0$ for some finite k, or the sequence $\{\|g^{(k)}\|\}_{k\in\mathbb{N}}$ converges to zero R-linearly.*

In many applications, it is shown that BB rules are able to greatly speed up the method, exhibiting a practical behaviour of higher order; however, a theoretical explanation of this phenomenon is still missing.

For what concern the possibility of extending the use of the Barzilai-Borwein rules to general (continuously differentiable) non-quadratic minimization problems, linesearch techniques are necessary to ensure the convergence to a stationary point. At this regard, the first important contribution was that of Raydan [76] who proposed the use of the Grippo-Lampariello-Lucidi (GLL) nonmonotone linesearch strategy [52], summarized in Algorithm 1, to force a sufficient decrease in the objective function. The choice of a weaker nonmonotone technique rather than a standard monotone Armijo-type linesearch, is motivated by the need of preserving some local properties of the method related to the nonmonotone behaviour induced by the BB rules, which is behind the effectiveness of the method. In particular, the scheme proposed in [76], called Global Barzilai-Borwein (GBB) method, requires to compute the steplength $\alpha_k = \alpha_k^{BB1}$ provided that it belongs to a bounded interval $\left(\frac{1}{\epsilon}, \epsilon\right)$, $\epsilon > 0$, otherwise $\alpha_k = \delta > 0$; then, the gradient iteration is given by

$$x^{(k+1)} = x^{(k)} + \nu_k d^{(k)}, \tag{1.25}$$

where $d^{(k)} = -g^{(k)}$, $\nu_k = \gamma^s \alpha_k$, $\gamma \in (0,1)$ and s is the first nonnegative integer such that the following condition is satisfied:

$$f(x^{(k)} + \gamma^s \alpha_k d^{(k)}) \leqslant f_{ref} + \sigma \gamma^s \alpha_k {g^{(k)}}^T d^{(k)}, \tag{1.26}$$

where $\sigma \in (0,1)$ and the reference function value f_{ref} is defined as the maximum value of the objective function over the last M iterations

$$f_{ref} = \max_{0 \leqslant j \leqslant \min(k,M-1)} f(x^{(k-j)}).$$

Let observe that for $M = 1$ the standard monotone Armijo rule is recovered:

$$f(x^{(k)} + \alpha d^{(k)}) \leqslant f(x^{(k)}) + \sigma \alpha g^{(k)^T} d^{(k)}, \tag{1.27}$$

where the steplength is set equal to $\alpha = \gamma^s \alpha_k$ and s is the first nonnegative integer such that (1.27) is satisfied. The convergence properties of the GBB algorithm are reported in the following theorem.

Theorem 1.2 *[76, Thm. 2.1] Assume that $\Omega_0 = \{x : f(x) \leqslant f(x^{(0)})\}$ is a bounded set. Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable in some neighborhood N of Ω_0. Let $\{x^{(k)}\}_{k\in\mathbb{N}}$ be the sequence generated by the GBB algorithm. Then either $g^{(k)} = 0$ for some finite k, or the following properties hold:*

(i) $\lim_{k\to\infty} \|g^{(k)}\| = 0$;

(ii) no limit point of $\{x^{(k)}\}_{k\in\mathbb{N}}$ is a local maximum of f;

(iii) if the number of stationary points of f in Ω_0 is finite, then the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ is convergent.

We observe that other nonmonotone linesearch strategies are available (e. g., see [89]). For a more detailed analysis on the convergence properties of nonmonotone linesearch methods see also [24].

Algorithm 1 Nonmonotone linesearch algorithm (GLL)

1: **Initialize**: $x^{(k)}, g^{(k)} \in \mathbb{R}^n$, $\overline{\alpha} > 0$, $M \in \mathbb{N}$, $\sigma, \gamma \in (0,1)$, $f_{ref} = \max\{f(x^{(k-j)}, 0 \leqslant j \leqslant \min(k, M)\}$;
2: $\alpha \leftarrow \overline{\alpha}$;
3: **while** $f(x^{(k)} - \alpha g^{(k)}) > f_{ref} - \sigma \alpha g^{(k)^T} g^{(k)}$ **do**
4: $\quad \alpha \leftarrow \gamma \alpha$;
5: **end while**
6: $\alpha_k \leftarrow \alpha$;
7: **return** α_k

It is worth noting that in the non-quadratic case we need to assume the well-known *curvature condition* [65]

$$s^{(k-1)^T} y^{(k-1)} > 0. \tag{1.28}$$

Under this assumption, the inequality $\alpha_k^{BB2} \leqslant \alpha_k^{BB1}$ still holds and, in addition, from Taylor's theorem ([65, Thm. 11.1]) we have

$$y^{(k-1)} = \int_0^1 \nabla^2 f(x^{(k-1)} + t s^{(k-1)}) s^{(k-1)} dt,$$

and, then, we obtain

$$\frac{1}{\alpha_k^{BB1}} = \int_0^1 \frac{s^{(k-1)^T} \nabla^2 f(x^{(k-1)} + ts^{(k-1)}) s^{(k-1)} dt}{\|s^{k-1}\|^2},$$

that is, the inverse of the steplength defined by the BB1 rule can be interpreted as the Rayleigh quotient related to the average of the Hessian matrix along the segment $s^{(k-1)}$. A similar property holds for α_k^{BB2}.

Improving the BB rules: the alternating strategies

Despite their nice spectral properties, the BB rules may produce a nonmonotonic behaviour in the sequences $\{f(x^{(k)})\}$ and $\{\|g^{(k)}\|\}$. Indeed, the recurrence (1.13) implies, for any $i = 1, \ldots, n$:

$$\alpha_k \approx \frac{1}{\lambda_i} \quad \Rightarrow \quad \begin{cases} \left|\mu_i^{(k+1)}\right| \ll \left|\mu_i^{(k)}\right|, & \\ \left|\mu_j^{(k+1)}\right| < \left|\mu_j^{(k)}\right| & \text{if} \quad j < i, \\ \left|\mu_j^{(k+1)}\right| > \left|\mu_j^{(k)}\right| & \text{if} \quad j > i \;\; \lambda_j > 2\lambda_i. \end{cases} \tag{1.29}$$

From (1.29) we may observe that:

- when α_k is close to the reciprocal of an eigenvalue λ_i, then it forces a remarkable reduction of $|\mu_i^{(k+1)}|$ with respect to $|\mu_i^{(k)}|$; in particular, from (1.24), the sequence $|\mu_1^{(k)}|$ is monotonically decreasing;
- if α_k is close to $1/\lambda_1$, it can amplify the absolute values of the gradient's eigencomponents corresponding to the larger eigenvalues, in particular $|\mu_n^{(k+1)}|$ is increased by a factor close to the condition number of A;
- if α_k is close to $1/\lambda_n$, the effect on the reduction of small index eigencomponents is negligible, especially when the condition number λ_n/λ_1 is large.

As observed by Fletcher in [44], even tough this nonmonotonic behaviour might seem undesirable, it is somehow responsible for the effectiveness and fast convergence of the method, and it should not be completely prevented by using technique for limiting it, but may be properly exploited, particularly when the condition number is large. Indeed, when an iteration occurs on which small index components dominate the gradient $g^{(k-1)}$, this gives rise to a small steplength α_k (from (1.19)), which in turn causes a large increase in large index components of the gradient; the effect of this increase is to produce subsequent large steplengths, leading to gradients in which large index components are no longer dominant, and therefore, in successive

iterations the contribution of small index components in the gradient can be significantly reduced as well.

A way to take advantage of the described nonmonotonic behaviour is proposed in [90], by considering new strategies which adaptively select a small steplength or a large one at every iteration or in a more dynamical way. The heuristic idea behind this approach is the following: assuming that the eigenvalues of A can be simply classified as the large ones and the small ones, then a certain superlinear behavior is related to how frequently the gradient is approaching the eigenvectors associated with the large or small eigenvalues of A, during the iterative procedure.

A practical implementations of this idea results in the Adaptive Barzilai-Borwein (ABB) scheme:

$$\alpha_k^{ABB} = \begin{cases} \alpha_k^{BB2} & \text{if} \quad \frac{\alpha_k^{BB2}}{\alpha_k^{BB1}} < \tau, \\ \\ \alpha_k^{BB1} & \text{otherwise,} \end{cases} \tag{1.30}$$

where $\tau \in (0,1)$.
The adaptive criterion used to switch between the two steplengths is based on the value

$$\frac{\alpha_k^{BB2}}{\alpha_k^{BB1}} = \frac{\left(g^{(k-1)^T} A g^{(k-1)}\right)^2}{\|Ag^{(k-1)}\|^2 \|g^{(k-1)}\|^2} = \cos^2(\psi_{k-1}), \tag{1.31}$$

where ψ_{k-1} is the angle between $g^{(k-1)}$ and $Ag^{(k-1)}$. Hence, when $\frac{\alpha_k^{BB2}}{\alpha_k^{BB1}} \approx 1$ this means that $g^{(k-1)}$ is a sufficiently good approximation of an eigenvector of A. In other words, the method tends to force a sequence of small steplengths α_k^{BB2} with the aim of reducing the larger index eigencomponents of the gradient, until α_k^{BB1} becomes a suitable approximation of the inverse of some small eigenvalue.

A further improvement of this approach is represented by the ABB_{min} strategy proposed in [47], which consists in substituting α_k^{BB2} in (1.30) with a shorter steplength, as follows:

$$\alpha_k^{ABB_{min}} = \begin{cases} \min\{\alpha_j^{BB2} \mid j = \max\{1, k-m_\alpha\}, \ldots, k\} & \text{if } \frac{\alpha_k^{BB2}}{\alpha_k^{BB1}} < \tau, \\ \\ \alpha_k^{BB1} & \text{otherwise,} \end{cases} \tag{1.32}$$

where $m_\alpha > 0$ is a positive integer. This choice is motivated by the idea (supported by numerical experiments) that more efficient schemes could be obtained by improving the ability to approximate λ_1^{-1}, achieving a fast reduction of the first gradient component. Then, the

rule (1.32) is aimed at forcing steplengths that reduce large index components of the gradient, in such a way that a following BB1 step will likely depend on a gradient dominated by small components.
It is well-known that the alternating strategies generally outperform other BB-based gradient methods, in particular for ill-conditioned problems; furthermore, the ABB_{min} approach has been proved to improve the efficiency of the original ABB method [47, 80, 90].

For QP problems, the R-linear convergence of both (1.30) and (1.32) can be derived from the general result obtained for the class of methods known as Gradient Methods with Retards (GMR) [25, 48]. This approach, inspired by the BB methods, defines a family of gradient methods which exploit the cyclic use of a steplength in some consecutive iterations. Given a positive integer m and a set of real numbers $q_i \geqslant 1$, $i = 1, \ldots, m$, the GMR steplength is given by

$$\alpha_k^{GMR} = \frac{e^{(\beta_k)^T} A^{\rho_k - 1} e^{(\beta_k)}}{e^{(\beta_k)^T} A^{\rho_k} e^{(\beta_k)}}, \tag{1.33}$$

where $e^{\beta_k} = x^{(\beta_k)} - x^*$, $\beta_k \in \{\max\{0, k-m\}, \ldots, k-1, k\}$ and $\rho_k \in \{q_1, q_2, \ldots, q_m\}$, assuming that x^* is the unique solution of (1.7).
Clearly, the BB rules and their alternated versions are special instances of (1.33). Indeed, by observing that $Ae^{(\beta_k)} = A(x^{(\beta_k)} - x^*) = g^{(\beta_k)}$, for $m = 1$, $\beta_k = \max\{0, k-m\}$ and $\rho_k \equiv 3$ we can recover the BB1 rule; similarly for $\rho_k \equiv 4$ we obtain the BB2 rule. The ABB method can be regarded as a member of GMR family for $\beta_k = k-1$ and $\rho_k \in \{3, 4\}$, where the switching criterion beetween the two possible values for ρ_k is based on the ratio (1.31); in a similar way, the ABB_{min} strategy is obtained with suitable choices of β_k and ρ_k based on the value (1.31).

We report the more general convergence result, established in [25], for the gradient method (1.8) applied to any-dimensional QP problems of type (1.7), where the steplength satisfies the following property (see [25, Property(A) p.9]):

Let $0 < \lambda_1 \leqslant \cdots \leqslant \lambda_n$ be the eigenvalues of the matrix A. The steplength α_k has the Property (A) if there exist an integer m and positive constants $M_1 \geqslant \lambda_1$ and M_2 such that

(i) $\lambda_1 \leqslant \alpha_k^{-1} \leqslant M_1$;

(ii) for any integer $l \in [1, n-1]$ and real number $\epsilon > 0$, if

$$\sum_{i=1}^{l} (g_i^{(k-j)})^2 \leqslant \epsilon \text{ and } (g_{l+1}^{(k-j)})^2 \geqslant M_2 \epsilon \text{ hold for } j \in [0, \min\{k, m\} - 1]$$

then $\alpha_k^{-1} \geqslant \frac{2}{3}\lambda_{l+1}$.

Combining (i) and (ii) the following theorem can be proved as generalization of the convergence result by Dai and Liao reported in Theorem 1.1.

Theorem 1.3 *[25, Thm. 4.1] Let $f(x)$ be a strictly convex quadratic function. Let $\{x^{(k)}\}_{k\in\mathbb{N}}$ be the sequence generated by the gradient method (1.8) where the steplength α_k has the Property(A). Then either $g^{(k)} = 0$ for some finite k, or the sequence $\{\|g^{(k)}\|\}_{k\in\mathbb{N}}$ converges to zero R-linearly.*

Finally, the R-linear convergence of the GMR method can be derived as corollary of Theorem 1.3, since the steplength (1.33) satisfies Property (A) (see [25]).

We conclude this section by mentioning also the ABB_{min} variant in which a variable setting of the parameter τ is exploited, as suggested in [15]:

$$\alpha_k^{ABBmin} = \begin{cases} \min\{\alpha_j^{BB2} \,|\, j = \max\{1, k-m_\alpha\}, \ldots, k\} & \text{if } \frac{\alpha_k^{BB2}}{\alpha_k^{BB1}} < \tau_k, \\ \alpha_k^{BB1} & \text{otherwise,} \end{cases} \tag{1.34}$$

where

$$\tau_{k+1} = \begin{cases} \tau_k/\vartheta & \text{if } \frac{\alpha_k^{BB2}}{\alpha_k^{BB1}} < \tau_k, \\ \tau_k \cdot \vartheta & \text{otherwise,} \end{cases} \tag{1.35}$$

with $\vartheta > 1$. A typical value for ϑ is 1.1. It is worthwhile observing that this variable setting makes the efficiency of the steplength strategy less dependent on the value of τ provided by the user and, in several applications, it allowed remarkable performance improvements with respect to the standard ABB_{min} strategy (see for example [15, 61, 71, 88]).

1.1.2 *The Limited Memory Steepest Descent approach*

We now describe the steplength updating rule proposed by Fletcher in [45] within the Limited Memory Steepest Descent (LMSD) method. This approach was first tailored for the quadratic case and then adapted for general non-quadratic functions, introducing some form of monotonicity needed to drive the gradients to zero. The basic idea of the method is to divide up the sequence of steepest descent iterations into groups of m iterations, referred to as *sweeps*, where m is a small positive integer and, at each new sweep, to compute the steplengths as the reciprocals of the so-called Ritz values [49] of the Hessian matrix (arising from the Lanczos process [59]), assuming that m Ritz values of the Hessian are available from the previous sweep. The Ritz values belong to the spectrum of A and provide m approximations

of its eigenvalues [49]. Since the Lanczos process requires the Hessian matrix to be available, Fletcher suggested an alternative way of computing the Ritz values that exploits the gradients of the objective function related to m previous iterations. The main steps of this procedure are briefly outlined below.

At any iteration, let suppose that, in addition to $x^{(k)}$ and $g^{(k)}$, the most recent m back gradients are available

$$G = \left[g^{(k-m)} g^{(k-m+1)} \cdots g^{(k-1)} \right], \tag{1.36}$$

where $m \leqslant \overline{m}$ is limited to an upper bound $\overline{m}$ on the number of such vectors that can be stored. In general, it is assumed $\overline{m} \ll n$.
The $n \times m$ matrix G helps to rewrite the recurrence equations (1.9) in matrix form as

$$AG = [G,\ g^{(k)}]\, J, \tag{1.37}$$

where J is the $(m+1) \times m$ matrix

$$J = \begin{pmatrix} \frac{1}{\alpha_{k-m}} & & \\ -\frac{1}{\alpha_{k-m}} & \ddots & \\ & \ddots & \frac{1}{\alpha_{k-1}} \\ & & -\frac{1}{\alpha_{k-1}} \end{pmatrix},$$

and α_{k-i} is the steplength associated with the gradient $g^{(k-i)}$, for $i = 1, \ldots, m$. Equation (1.37) can be used to compute the tridiagonal matrix T resulting from the application of m iterations of the Lanczos process to the matrix A, without explicitly involving this latter one. To this purpose, let remind that the recurrence properties of the gradient descent method (1.8) for minimizing a quadratic objective function of the form (1.7), implies

$$x^{(k)} - x^{(k-m)} \in \operatorname{span}\{g^{(k-m)}, Ag^{(k-m)}, A^2 g^{(k-m)}, \ldots, A^{m-1} g^{(k-m)}\}, \tag{1.38}$$

that is, the vector $x^{(k)} - x^{(k-m)}$ belongs to the m-dimensional Krylov space spanned by the Krylov sequence started from $g^{(k-m)}$. In particular, it follows that

$$g^{(k)} - g^{(k-m)} \in \operatorname{span}\left\{ Ag^{(k-m)}, A^2 g^{(k-m)}, \ldots, A^m g^{(k-m)} \right\}. \tag{1.39}$$

The importance of this Krylov sequence is related to its ability in providing m distinct estimates of the eigenvalues of the matrix A: indeed, the Lanczos iterative process applied to A, with starting vector $q_1 = \dfrac{g(x^{(k-m)})}{\|g(x^{(k-m)})\|}$, allows to generate a matrix $Q = [q_1,\ q_2,\ \ldots,\ q_m]$

whose columns are an orthonormal basis for the Krylov space (1.38) and such that the matrix

$$T = Q^T A Q \tag{1.40}$$

is a $m \times m$ tridiagonal matrix. The eigenvalues of T are the so-called *Ritz values*. From (1.39), the columns of G are a set of generators of the m-dimensional Krylov subspace and, assuming that they are linearly independent, we may write $G = QR$, where R is an upper triangular and non singular matrix; hence, from (1.40) and (1.37) we have

$$T = R^{-T} G^T A G R^{-1} = R^{-T} G^T [G \; g^{(k)}] \, J R^{-1} = [R \; r] \, J R^{-1}, \tag{1.41}$$

where the vector r is the solution of the linear system $R^T r = G^T g^{(k)}$. Since the matrix R can be obtained from the Cholesky factorization of $G^T G$, then the computation of Q by means of the Lanczos process is not required. At the beginning of each sweep, assuming that m Ritz values θ_i, $i = 1, \ldots, m$ are available from a previous sweep, the steplengths for the next m gradient iterations are defined as the reciprocals of the Ritz values:

$$\alpha_{k-1+i} = \frac{1}{\theta_i}, \quad i = 1, \ldots, m. \tag{1.42}$$

Let observe that for $m = 1$ we recover as special case the BB1 rule. Within each sweep, m steepest descent steps are computed

$$x^{(k+i)} = x^{(k-1+i)} - \alpha_{k-1+i} \, g^{(k-1+i)}, \quad i = 1, \ldots, m,$$

starting from $x^{(k)}$ and using the steplengths (1.42). Then, the iterate $x^{(k+m)}$ is the starting vector for the next sweep and the gradients $g^{(k-1+i)}$, $i = 1, \ldots, m$, obtained on the sweep are used to calculate Ritz values for the next sweep. The convergence theorem proposed in [45], follows an argument similar to that used in [75] for the BB scheme. Furthermore, a proof of the R-linear convergence of the sequence $\{\|g^{(k)}\|\}_{k\in}$ to zero, under suitable assumptions, can be found in [23].

Similarly to the BB-based approaches, the sequences of gradient norms $\{\|g^{(k)}\|\}$ and function values $\{f^{(k)}\}$ exhibit a nonmonotonic behaviour. Indeed, if θ_k is the current Ritz value, from (1.13) we can write

$$\mu_i^{(k+1)} = \left(1 - \frac{\lambda_i}{\theta_k}\right) \mu_i^{(k)}, \quad i = 1, \ldots, n,$$

where $\lambda_1 < \theta_k < \lambda_n$ (for the properties of the Rayleigh quotients); then $\{|\mu_1^{(k)}|\}$ is monotonically decreasing, with slow rate when θ_k is close to λ_n; on the other hand, when θ_k is close to λ_1, the eigencomponent of the gradient corresponding to the smallest eigenvalue is

drastically reduced, while those corresponding to the larger eigenvalues may increase: for example the quantity

$$|\mu_n^{k+1}| = \left|1 - \frac{\lambda_n}{\theta_k}\right| |\mu_n^{(k)}|$$

increases by a factor close to the condition number of A.

Some issues concerning how to ensure monotonocity and how to adapt the method to non-quadratic case are treated in [45]. With regard to the former task, the author suggests to select the Ritz values in decreasing order of size, during a sweep; if a new iterate fails to improve on the value $f^{(k)}$ obtained at the beginning of a sweep, then the Cauchy steplength is computed and the sweep is terminated. A further check is implemented to guarantee the decreasing of the gradient norm within a sweep. According to the author, imposing monotonicity does not seem to reduce or particularly affect the effectiveness of the method, in the quadratic case.

Similar strategies to ensure monotonicity are needed in the non-quadratic case, to guarantee the convergence of the method; in addition, suitable solutions have to be found to handle the following issues, arising in the non-quadratic case:

- the matrix T is upper Hessenberg, but not usually tridiagonal;
- the matrix A is no longer available to compute the Cauchy steplength in the algorithm;
- various effects related to the existence of non-positive curvature.

Of course, there are different possibilities to address them. For what concern the matrix T, Fletcher suggests to construct a symmetric tridiagonal matrix $\tilde{T}$, by replacing the strict upper triangle of T by the transpose of the strict lower triangle, and use the eigenvalues of $\tilde{T}$ to compute Ritz-like values for the next sweep. To overcome the second drawback, one could implement alternative linesearch strategies (for example that suggested by Raydan [76]); however, in [45] a slightly more elaborate search aiming to satisfy Wolfe- Powell conditions is chosen (essentially that described in Section 2.6 of Fletcher [42]), since it ensures that the resulting $\tilde{T}$ has at least one positive eigenvalue.

Effects related to non-positive curvature can result also in the presence of some negative eigenvalues of matrix $\tilde{T}$. In this case, some possibilities are: to discard the oldest gradients and recompute a smaller matrix $\tilde{T}$, or either terminate the sweep when a negative Ritz value is found, or carry out a linesearch before terminating the sweep. The last option is implemented in [45].

Finally, Fletcher in his paper mentioned the possibility of implementing a preconditioned version of the sweep method and also a version involving the *harmonic Ritz values* [66], which are all weighted

harmonic means of the eigenvalues of A (while the standard Ritz values are weighted arithmetic means of A's eigenvalues); in this case the second BB rule is obtained as special instance for $m = 1$.

1.1.3 *Other efficient steplength selection rules*

We conclude this section by mentioning some other efficient steplength selection strategies based on spectral properties, which share R-linear convergence rate in the quadratic case.

Some of them have been proposed with the aim of generalising the BB methods, and are based either on the alternation of Cauchy steplength and BB steplengths, like in the Alternate Step gradient method (AS) [25], or on their cyclic use as in the Adaptive Cyclic Barzilai-Borwein (ACBB) [84] or the Cauchy Barzilai-Borwein method (CBB) [77]. All these rules are special implementations of the GMR scheme (1.33).

Recent proposals based on a different philosophy are the alternating strategies SDA and SDC suggested in [34, 35] for QP problems. They are based on the idea of alternating Cauchy steplengths $\alpha_k^{SD} = \dfrac{{g^{(k)}}^T g^{(k)}}{{g^{(k)}}^T A g^{(k)}}$ with constant steplengths $\tilde{\alpha}_k$, with the aim of escaping from the two-dimensional space where the steepest descent method with Cauchy steplength asymptotically reduces.
Given an integer $h \geqslant 2$, the SDA method (the acronym stands for Steepest Descent with Alignment) computes a sequence of Cauchy steplengths α_k^{SD} until a switch condition is satisfied, then it performs h consecutive iteration with the following constant steplength

$$\tilde{\alpha}_k \equiv \tilde{\alpha}_k^{SDA} = \left(\frac{1}{\alpha_{k-1}^{SD}} + \frac{1}{\alpha_k^{SD}} \right)^{-1} , \tag{1.43}$$

provided that it produces a decrease in the objective function, otherwise the steplength $2\alpha_k^{SD}$ is used. The switch condition is defined as

$$|\tilde{\alpha}_k - \tilde{\alpha}_{k-1}| < \epsilon \tag{1.44}$$

where $\epsilon > 0$ is a small positive scalar. The rationale behind this strategy is related to the following result.

Proposition 1.2 *[34, Prop. 3.2] Let x^* denote the solution of problem (1.7). Under the assumptions (1.10) and*

$${g^{(0)}}^T v_1 \neq 0, \quad {g^{(0)}}^T v_n \neq 0 ,$$

the sequence $\{x^{(k)}\}_{k \in \mathbb{N}}$ generated by the gradient method (1.8) with constant steplength

$$\alpha = \frac{1}{\lambda_1 + \lambda_n} \tag{1.45}$$

converges to x^*. *Moreover*

$$\lim_k \frac{\mu_h^{(k)}}{\mu_1^{(k)}} = \frac{\mu_h^{(0)}}{\mu_1^{(0)}} \lim_k \left(\frac{\lambda_1}{\lambda_n} + \frac{\lambda_n - \lambda_h}{\lambda_n} \right)^k = 0, \quad h = 2, \dots, n, \tag{1.46}$$

where $\mu_i^{(k)}$, $i = 1, \dots, n$, *are the eigencomponents of* $g^{(k)}$.

When the steplength (1.45) is adopted, relation (1.46) ensures that the sequence $\{\mu_h^{(k)}\}_{k \in \mathbb{N}}$, for $h > 1$, converges to zero faster than the sequence $\{\mu_1^k\}_{k \in \mathbb{N}}$; as a consequence, the search direction tends to align with the eigendirection v_1 associated with the minimum eigenvalue λ_1. For k sufficiently large, the value (1.43) is an approximation of the steplength (1.45) (see [34, Prop. 3.1]).

Given two integers $h \geqslant 2$ and $m_c \geqslant 1$, the SDC updating rule [35] computes h consecutive Cauchy steplengths and then switchs to a different steplength, which is applied in m_c consecutive iterations, in accordance with the following scheme

$$\alpha_k^{\text{SDC}} = \begin{cases} \alpha_k^{\text{SD}} & \text{if} \quad \bmod(k, h + m_c) < h, \\ \alpha_s^{\text{Y}} & \text{otherwise, with } s = \max\{i \leqslant k : \quad \bmod(i, h + m_c) = h\}, \end{cases} \tag{1.47}$$

where the steplength α_s^{Y} is computed by means of the Yuan formula [87]

$$\alpha_k^{\text{Y}} = 2 \left(\sqrt{\left(\frac{1}{\alpha_{k-1}^{\text{SD}}} + \frac{1}{\alpha_k^{\text{SD}}} \right)^2 + 4 \frac{\|g^{(k)}\|}{\alpha_{k-1}^{\text{SD}} \|g^{(k-1)}\|}} + \frac{1}{\alpha_{k-1}^{\text{SD}}} + \frac{1}{\alpha_k^{\text{SD}}} \right)^{-1}. \tag{1.48}$$

Let remind that the steplength (1.48) is defined as the value $\overline{\alpha}$ such that, for 2-dimensional convex quadratics problems, three iterations of the SD method equipped with the steplengths

$$\begin{aligned} \alpha_1 &= \alpha_1^{\text{SD}} \\ \alpha_2 &= \overline{\alpha} \\ \alpha_3 &= \alpha_3^{\text{SD}} \end{aligned}$$

are sufficient to find the minimizer of the objective function in exact arithmetic. In particular, Yuan steplength α_s^{Y} satisfies the inequalities

$$\left(\frac{1}{\alpha_{k-1}^{\text{SD}}} + \frac{1}{\alpha_k^{\text{SD}}} \right)^{-1} \leqslant \alpha_s^{\text{Y}} \leqslant \min\{\alpha_{k-1}^{\text{SD}}, \ \alpha_k^{\text{SD}}\}$$

and tends to approximate the reciprocal of the largest eigenvalue λ_n of the Hessian matrix (see [35, Thm. 3.1]). The R-linear convergence of SDA and SDC strategies can be derived from the general result

stated in Theorem 1.3, since the authors proved that both the rules satisfy the Property (A).

We now report the steplength rule proposed in [73], which we refer to as Golden Arcsine (GA). This method generates steplengths α_k such that the asymptotic distribution of the sequence $\{\frac{1}{\alpha_k}\}$ is close to the distribution with the arcsine density on the interval $[m, M]$:

$$p(t) = \frac{1}{\pi\sqrt{(t-m)(M-t)}}, \quad m \leqslant t \leqslant M$$

where $m = \lambda_1 + \epsilon$, $M = \lambda_n + \epsilon$, with small $\epsilon \geqslant 0$. Indeed, in this case, the asymptotic rate of convergence of the gradient method can approach that of the Conjugate Gradient (see [73, Prop. 5]). However, in practice, λ_1 and λ_n are unknown; then during the iterative procedure suitable estimators of the extremal eigenvalues are computed, based on the moments of the probability measures associated with the spectrum of A at each iteration. To sum up, the steplength updating rule is defined as

$$\alpha_k^{GA} = \frac{1}{\beta_k}, \qquad \beta_k = m_k + (M_k - m_k)z_k \tag{1.49}$$

where m_k and M_k are suitable approximation of the minimum and maximum eigenvalues of A, respectively, updated at each iteration and the sequence $\{z_k\}_{k\in\mathbb{N}}$ with the asymptotic arcsine density is given by

$$z_k = (1 + \cos(\pi u_k))^2, \quad k = 0, 1, \ldots,$$

where $u_{2k} = \min\{v_k, 1 - v_k\}$, $u_{2k+1} = \max\{v_k, 1 - v_k\}$, v_k is the fractional part of $(k+1)\phi$, with ϕ denoting the golden ratio, $\phi = \frac{\sqrt{5}+1}{2}$.

Another technique to build steplengths aimed at breaking the zig-zagging pattern of the Cauchy steplength is developed in [51], and is based on the use of the Chebyshev nodes, i. e., the roots of the Chebyshev polynomial of the first kind.

An interesting analysis of the spectral behaviour of the mentioned strategies within the gradient method can be found in [80].

1.2 CLASSICAL GRADIENT PROJECTION METHODS

Let now consider the constrained version of the differentiable optimization problem (1.1), where the feasible region Ω is a closed and convex subset of $\mathbb{R}^n$.

The basic iteration of the standard Gradient Projection (GP) method along the feasible direction [9, Chapter 2] is defined by

$$\begin{aligned} x^{(k+1)} &= x^{(k)} + \nu_k d^{(k)} = \\ &= x^{(k)} + \nu_k \left(\Pi_\Omega \left(x^{(k)} - \alpha_k \nabla f(x^{(k)}) \right) - x^{(k)} \right) \end{aligned} \tag{1.50}$$

where $\Pi_\Omega(\cdot)$ denotes the Euclidean projection onto the feasible region, $\alpha_k \in [\alpha_{min}, \alpha_{max}]$ is the steplength parameter and $\nu_k \in (0, 1]$ is a linesearch parameter ensuring a sufficient decrease of the objective function along the direction $d^{(k)}$, by means of an Armijo rule (1.27) or its non-monotone version (1.26). This guarantees the convergence of the method; at this regard, in [27] some box-constrained quadratic 2-dimensional counter-examples were constructed in order to show how a gradient projection scheme equipped with BB-based steplength selection rules may fail to converge without some kind of linesearch. However, this failure would appear to be very unlikely in practice.

We report in Algorithm 2 the main steps of a general GP method performing a linesearch along the feasible direction. Let observe that for $M = 1$ in the step 3 of the algorithm, the standard Armijo rule is obtained.

Algorithm 2 Gradient Projection (GP) method

Initialize: $x^{(0)} \in \Omega, \quad \delta, \sigma \in (0, 1),$
$M \in \mathbb{N}, \quad 0 < \alpha_{min} \leqslant \alpha_{max},$
$\alpha_0 \in [\alpha_{min}, \alpha_{max}];$

1: **for** $k = 0, 1, \ldots$ **do**
2: $\quad d^{(k)} \leftarrow \Pi_\Omega \left(x^{(k)} - \alpha_k g(x^{(k)}) \right) - x^{(k)};$
3: $\quad \nu_k \leftarrow 1; \qquad f_{ref} \leftarrow \max\{f(x^{(k-i)}),\ 0 \leqslant i \leqslant \min(k, M-1)\};$
4: $\quad$ **while** $f(x^{(k)} + \nu_k d^{(k)}) > f_{ref} + \sigma \nu_k g(x^{(k)})^T d^{(k)}$ **do**
5: $\quad\quad \nu_k \leftarrow \delta \nu_k;$
6: $\quad$ **end while**
7: $\quad x^{(k+1)} \leftarrow x^{(k)} + \nu_k d^{(k)};$
8: $\quad$ define the steplength $\alpha_{k+1} \in [\alpha_{min}, \alpha_{max}];$
9: **end for**

In the next theorems we recall the basic convergence results for the sequence $\{x^{(k)}\}_{k \in \mathbb{N}}$ generated by Algorithm 2. We remark that Theorem 1.4 refers to a GP method where a nonmonotone linsearch strategy is performed as well as the Armijo rule, while the stronger results

stated in Theorems 1.5-1.6 hold when a standard Armijo rule is considered.

Theorem 1.4 *[10, Thm. 2.2] Let $\alpha_{min}, \alpha_{max}$ be positive constants such that $0 < \alpha_{min} \leqslant \alpha_{max}$. Let $\{\alpha_k\}_{k\in\mathbb{N}} \subset [\alpha_{min}, \alpha_{max}]$ be a sequence of parameters. Then any accumulation point of the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ generated by Algorithm 2 is a stationary point for problem (1.1).*

Theorem 1.5 *[56, Thm. 1] Assume that the objective function of (1.1) is convex and the solution set is not empty. Then the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ generated by Algorithm 2 combined with the Armijo rule,converges to a solution of (1.1).*

Theorem 1.6 *[14, Thm. 3.2] Let the hypotheses of Theorem 1.5 hold and assume, in addition, that ∇f is globally Lipschitz on Ω (or ∇f is locally Lipschitz and f is level bounded on Ω). Let f^* be the optimal function value for problem (1.1), then*

$$f(x^{(k)}) - f^* = O\left(\frac{1}{k}\right).$$

The analysis developed in [85] ensures the R-linear convergence of a GP method combined with the BB rules and the GLL nonmonotone linesearch procedure for the minimization of general strongly convex functions. A variant of the GP method, developed to improve the convergence rate, is the scaled GP method (or SGP method), defined by the following iteration

$$\begin{aligned} x^{(k+1)} &= x^{(k)} + \nu_k d^{(k)} = \\ &= x^{(k)} + \nu_k \left(\Pi_{\Omega,D_k}\left(x^{(k)} - \alpha_k D_k^{-1}\nabla f(x^{(k)})\right) - x^{(k)}\right), \end{aligned} \tag{1.51}$$

where D_k, $k = 0, 1, \ldots$, is a SPD matrix whose eigenvalues lies in the bounded interval $\left[\frac{1}{\mu}, \mu\right]$, $\mu \geqslant 1$, and $\Pi_{\Omega,D_k}(\cdot)$ is the projection operator onto Ω with respect to the D_k-norm:

$$\Pi_{\Omega,D_k}(y) = \arg\min_{x\in\Omega} \|x - y\|^2_{D_k}. \tag{1.52}$$

In the practice, the sequence $\{D_k\}_{k\in\mathbb{N}}$ is chosen as a set of diagonal SPD matrices. For a convergence analysis of general SGP methods we refer to [11, 14, 15].

Many well-known gradient projection methods for constrained optimization exploit the same steplength selections designed for the unconstrained case in combination with some kind of linesearch strategy [2, 10, 12, 27, 41, 58, 61, 78, 86, 91].

Among them, one of the most popular is the Nonmonotone Spectral Projected Gradient method developed by Birgin et al. in [10], which consists in a scheme of the type described by Algorithm 2,

where the spectral steplength updating rule is provided by the BB1 rule (1.17). The authors considered also another classical GP scheme performing a nonmonotone linesearch on the feasible arc, instead of the linesearch on the feasible direction considered in Algorithm 2; however, they highlighted that these algorithms are especially effective when the projection step is easy to compute, for example when the region Ω is a n-dimensional box. A generalization of this approaches to the case where the constraints are not simple (and consequently the projection step is more difficult) is proposed in [11].

A different linesearch technique is suggested in [27] within a GP method involving BB-based steplength updating rules (called Projected Barzilai-Borwein or PBB), for box-constrained quadratic programs. In particular, based on the works [32, 82], the authors implemented an adaptive nonmonotone linesearch strategy with an Armijo-type acceptability test

$$f(x^{(k)} + \alpha d^{(k)}) \leqslant f_{ref} + \sigma\alpha {g^{(k)}}^T d^{(k)}$$

where $\sigma \in (0,1)$ and the reference function value f_{ref} is updated in accordance with the following scheme:

Initialize: $f_{ref} = +\infty$, $f_{best} = f_c = f(x^{(0)})$, $L \in \mathbb{N}$, $l = 0$;
1: **for** $k = 1, 2 \ldots$ **do**
2: **if** $f(x^{(k)}) < f_{best}$ **then**
3: $f_{best} \leftarrow f(x^{(k)})$, $f_c \leftarrow f(x^{(k)})$, $l \leftarrow 0$;
4: **else**
5: $f_c \leftarrow \max\{f_c, f(x^{(k)})\}$, $l \leftarrow l + 1$;
6: **if** $l = L$ **then**
7: $f_{ref} \leftarrow f_c$, $f_c \leftarrow f(x^{(k)})$, $l \leftarrow 0$;
8: **end if**
9: **end if**
10: **end for**

This choice seems to be motivated by practical considerations. Indeed, the authors claimed that this adaptive nonmonotone linesearch strategy is able to better preserve the properties of the spectral steplength rules rather than the GLL technique, especially when relatively small values of the parameter M are used in this latter one.

We now outline a variant of the PBB method, named projected Barzilai-Borwein with fall-back (PBBF), recently appeared in [58], for solving QP problems subject to separable convex constraints. This algorithm is based on the following theorem, which gives an estimate of the decrease of the convex quadratic objective function f defined in (1.7).

Theorem 1.7 *[39, Prop. 5.10] Let x^* denote the unique solution of (1.7), and let λ_{min} denote the smallest eigenvalue of A. If $\overline{\alpha} \in (0, 2\|A\|^{-1}]$ where $\|A\|$ denote the matrix norm induced by the Euclidean norm, then*

$$f(\Pi_\Omega(x - \overline{\alpha}g(x))) - f(x^*) \leqslant \eta(\overline{\alpha})\,(f(x) - f(x^*))\,, \tag{1.53}$$

where $\eta(\overline{\alpha}) = 1 - \hat{\alpha}\lambda_{min}$ and $\hat{\alpha} = \min\{\overline{\alpha}, 2\|A\|^{-1} - \overline{\alpha}\}$.

The estimate (1.53) is exploited within the PBBF method to achieve a sufficient decrease in the objective function, avoiding the use of a standard linesearch strategy. In particular, the algorithm generates the projected BB iterations until there is either the improvement of the objective function, or there are K consecutive iterations without improvement. In the first case, an additional GP iteration with a fixed steplength $\overline{\alpha} \in (0, 2/\lambda_{max}]$ is carried out to perform the acceptability test for achieving a sufficient decrease of the objective function at the next iteration; in the second case, the new iterate is defined by the fixed steplength GP iteration from the best of the last K iterations, again with $\overline{\alpha} \in (0, 2/\lambda_{max}]$. The R-linear convergence of the sequence $\{f(x^{(k)})\}$ generated by the PBBF scheme can be derived from Theorem 1.7 (see [39, Thm. 2]).

The mentioned methods and, in general, spectral gradient methods for constrained optimization share a lack of attention on the spectral properties of the steplength selection strategies related to the feasible region. In our opinion, this issue deserves an appropriate analysis, since most of these strategies have been designed for achieving a fast annihilation of the gradient of the objective function, which is not the main goal in constrained optimization. Therefore, a better understanding of the role of the steplength in gradient projection methods may be useful for improving first-order approaches in constrained optimization, at least for some special set of constraints.

Before concluding, we recall that QP problems subject to special closed and convex feasible set, such as box-constraints, can be addressed by means of other efficient techniques based on a different philosophy, which generally combine active set strategies with gradient projection method (see, e. g., [17, 38, 64, 79]). Among them, we will mention the Modified Proportioning with Reduced Gradient Projections (MPRGP) method [38, 39], which will be considered for some numerical experiments in Chapter 4. For this scheme, the R-linear rate of convergence is proved not only for the decrease of the cost function, but also for the norm of the projected gradient (see [39]).

2

A SIMPLE CASE: STEPLENGTH SELECTION IN GP METHODS FOR BOX-CONSTRAINED PROBLEMS

This chapter is devoted to investigate how some steplength selection rules designed in the context of unconstrained optimization can be adapted to gradient projection methods for solving box-constrained problems. Indeed, the observations made at the end of Chapter 1 make clear that the study of the role of the steplength in gradient projection methods has to be deepened, based on the idea that the presence of the feasible region might influence the spectral properties of the methods. To this purpose, we will proceed by first investigating the behaviour of the methods in the quadratic case, with the aim of extending the analysis to general non quadratic box-constrained optimization problems.
The chapter is organized as follows. In Section 2.1 we recall the first-order optimality conditions for a quadratic problem subject to box-constraints. In Section 2.2 the spectral properties of the BB rules are investigated, highlighting the relationship between the reciprocals of the steplengths generated by the rules and the spectrum of a special submatrix of the Hessian matrix. In Section 2.3 we propose a limited memory approach that combine the LMSD strategy with a gradient projection method, taking into account the presence of the box-constraints.

2.1 OPTIMALITY CONDITIONS

Throughout the chapter we will consider the following box-constrained quadratic programming (BQP) problem

$$\min_{x\in\Omega} f(x) \equiv \frac{1}{2}x^T Ax - b^T x + c, \tag{2.1}$$

where $A \in M_n(\mathbb{R})$ is an SPD matrix, $b \in \mathbb{R}^n$, $c \in \mathbb{R}$, and, given $\ell, u \in \mathbb{R}^n$ with $\ell \leqslant u$, the feasible region is defined by

$$\Omega = \{x \in \mathbb{R}^n : \ \ell \leqslant x \leqslant u\} . \tag{2.2}$$

We assume $\Omega \neq \varnothing$. Problem (2.1) has a unique solution x^*, satisfying the Karush-Khun-Tucker (KKT) first-order optimality conditions, which can be rewritten as

$$\begin{cases} g(x^*)_i = 0 & \text{for} \quad \ell_i < x_i^* < u_i, \\ g(x^*)_i \leqslant 0 & \text{for} \quad x_i^* = u_i, \\ g(x^*)_i \geqslant 0 & \text{for} \quad x_i^* = \ell_i. \end{cases} \tag{2.3}$$

We remind that $g(x)$ denotes the gradient of the objective function at x: $g(x) \equiv \nabla f(x) = Ax - b$. Hereafter, we define $\mathcal{N} = \{1, 2 \ldots, n\}$, $\mathcal{I}^* = \{i \in \mathcal{N}: \ell_i < x_i^* < u_i\}$ and $\mathcal{A}^* = \{i \in \mathcal{N}: x_i^* = \ell_i, x_i^* = u_i\}$. Apart from trivial cases, we assume $\varnothing \subset \mathcal{I}^* \subset \mathcal{N}$ and denote $m = \sharp\mathcal{I}^*$, $0 < m < n$.

We observe that the KKT conditions can be reformulated in terms of the *projected gradient*. Indeed, for any $x \in \Omega$, let denote the active set at x by $\mathcal{A}(x) = \mathcal{A}_\ell(x) \cup \mathcal{A}_u(x)$, where $\mathcal{A}_\ell(x) = \{i \in \mathcal{N}: x_i = \ell_i\}$ and $\mathcal{A}_u(x) = \{i \in \mathcal{N}: x_i = u_i\}$. The part of the gradient which violates the KKT conditions can be decomposed into the *free gradient* φ and the *chopped gradient* β, which are defined component-wise as

$$\begin{aligned} &\varphi_i(x) = g_i(x) \quad \text{for } i \in \mathcal{N}\backslash\mathcal{A}(x), \quad \varphi_i(x) = 0 \quad \text{for } i \in \mathcal{A}(x), \\ &\beta_i(x) = 0 \quad \text{for } i \in \mathcal{N}\backslash\mathcal{A}(x), \quad \beta_i(x) = \begin{cases} \max\{0, g_i(x)\} & \text{for } i \in \mathcal{A}_u(x), \\ \min\{0, g_i(x)\} & \text{for } i \in \mathcal{A}_\ell(x). \end{cases} \end{aligned} \tag{2.4}$$

The projected gradient is defined by

$$g^P(x) = \varphi(x) + \beta(x). \tag{2.5}$$

Clearly, the KKT conditions (2.3) at the solution x^* can be rewritten as follows:

$$g^P(x^*) = 0_{n,1}. \tag{2.6}$$

Condition (2.6) suggests a useful stopping criterion for practical implementations of GP methods. In particular, since (2.6) cannot be satisfied exactly in practice (especially for large-scale problems), it can be reasonable to require the weaker condition

$$\|g^P(x)\| \leqslant \epsilon, \tag{2.7}$$

for small $\epsilon > 0$ and $x \in \Omega$. The following result guarantees that any feasible vector x that satisfies (2.7) is near the solution.

Proposition 2.1 *[39, Lemma 5.1] Let x^* be the solution of (1.7) where A is a SPD matrix and let $g^P(x)$ denote the projected gradient at $x \in \Omega$. Then*

$$\|x - x^*\|_A^2 \leqslant 2(f(x) - f(x^*)) \leqslant \|g^P(x)\|_{A^{-1}} \leqslant \frac{\|g^P(x)\|}{\lambda_{min}}$$

where λ_{min} denotes the minimum eigenvalue of A.

2.2 SPECTRAL ANALYSIS OF THE BB RULES RELATED TO THE REDUCED HESSIAN MATRIX

Our analysis starts by considering the GP method described by Algorithm 2 for solving problem (2.1). In order to inspect the spectral

properties of the BB rules (1.17)-(1.18) we need to investigate how the gradient projection step influences their definitions. We will first study the BB1 rule and then, by proceeding in a similar way, we will develop the analysis for the second selection rule BB2.

As already observed, in the unconstrained case the effectiveness of the BB-based rules results in a fast annihilation of the gradient, which must be equal to zero at the solution. In the box-constrained case, in accordance with the KKT conditions (2.3), only the gradient's components corresponding to the inactive variables are zero at the solution. This simple observation suggests to implement a novel strategy able to foster a reduction of the components of the gradient related to the variables that remain inactive during the iterative procedure; at the same time, the components corresponding to the active variables of the current iterate are discarded; this procedure continues until the solution is reached and, consequently, the subset of the active box-constraints has stabilized.

In other words, to take advantage of the spectral properties induced by the BB rules, we have to properly select the second-order information to rely on. As a consequence, the effectiveness of the steplength selection in the constrained case depends on its ability in sweeping the spectrum of a particular submatrix of the Hessian rather than the whole Hessian matrix.

Surprisingly, this ability is inherently owned by the BB1 rule, as we will show in the next results, while the BB2 steplength does not seem to inherit such ability and needs appropriate modifications to preserve the spectral properties.

For the sake of clearness, we now introduce two subsets of indices that are indispensable to our analysis and will be extensively used throughout the chapter.

Let $\{x^{(k)}\}_{k\in\mathbb{N}}$ be the sequence generated by the GP algorithm. At the k-th iteration, the set of indices $\mathcal{N}$ may be partitioned into the following subsets:

$$\begin{aligned}\mathcal{J}_{k-1} &= \left\{ i \in \mathcal{N} \colon (x_i^{(k-1)} = \ell_i \wedge x_i^{(k)} = \ell_i) \vee (x_i^{(k-1)} = u_i \wedge x_i^{(k)} = u_i) \right\}, \\ \mathcal{I}_{k-1} &= \mathcal{N} \setminus \mathcal{J}_{k-1}.\end{aligned} \tag{2.8}$$

Therefore, during the iterative procedure two families of subsets of $\mathcal{N}$ arise, namely $(\mathcal{J}_{k-1})_{k\in\mathbb{N}}$ and $(\mathcal{I}_{k-1})_{k\in\mathbb{N}}$, which tend to approximate the corresponding sets $\mathcal{A}^*$ and $\mathcal{I}^*$ related to the solution, as k increases.

REMARK It worth noting that the subset $\mathcal{J}_{k-1}$ in (2.8) is equivalent to a special instance of the so-called *binding set* defined for a general

constrained minimization problem of the form (1.1), as follows (e. g., see [17] p.112 - Eq. (6.5)):

$$\mathcal{B}(x) = \{i \in \mathcal{N}\colon i \in \mathcal{A}(x), \xi_i(x) \geqslant 0)\}, \quad x \in \Omega, \tag{2.9}$$

where $\xi(x)$ is an estimate (computed by the algorithm) of the Lagrangian multiplier associated to the constraints at x. In particular, if Ω is the box-constrained set (2.2) and the Langrangian multipliers are chosen by the first-order estimates, then (2.9) corresponds to the following set

$$\mathcal{B}(x) = \{i \in \mathcal{N}\colon (x_i^{(k-1)} = \ell_i \wedge g_i^{(k-1)} \geqslant 0) \vee (x_i^{(k-1)} = u_i \wedge\ g_i^{(k-1)} \leqslant 0)\}, \tag{2.10}$$

which in turn is equivalent to $\mathcal{J}_{k-1}$.

Turning back to the BB1 rule, it can be readily observed that when some variables remain active along two subsequent iterations, they do not contribute to the computation of the steplength used to recover the next iterate. Indeed, if at the k-th iteration the subset $\mathcal{J}_{k-1}$ is not empty, then $s_j^{(k-1)} = 0$ for all $j \in \mathcal{J}_{k-1}$, and we can write

$$\alpha_k^{BB1} = \frac{\|s^{(k-1)}\|^2}{s^{(k-1)^T} y^{(k-1)}} = \frac{\|s_{\mathcal{J}_{k-1}}^{(k-1)}\|^2}{s_{\mathcal{J}_{k-1}}^{(k-1)^T} y_{\mathcal{J}_{k-1}}^{(k-1)}}. \tag{2.11}$$

Equation (2.11) provides a significant interpretation of the first BB rule in the box-constrained case; indeed, similarly to the unconstrained case, the updating rule α_k^{BB1} corresponds to the least-squares solution of a suitable secant equation

$$\alpha_k^{BB1} = \arg\min_{\alpha} \left\| \alpha^{-1} s_{\mathcal{J}_{k-1}}^{(k-1)} - y_{\mathcal{J}_{k-1}}^{(k-1)} \right\|^2. \tag{2.12}$$

In other words, α_k^{BB1} is defined by forcing the matrix $(\alpha_k I_n)^{-1}$ to approximate the submatrix $A_{\mathcal{J}_{k-1},\mathcal{J}_{k-1}}$ of A defined by the intersection of the rows and columns with indices in $\mathcal{J}_{k-1}$. We will refer to the matrix $A_{\mathcal{J}_{k-1},\mathcal{J}_{k-1}}$ as the *reduced Hessian matrix* at the $(k-1)$-th iteration.

Now, we are able to prove that the spectral properties induced by the BB1 rule are preserved with respect to the submatrix $A_{\mathcal{J}_{k-1},\mathcal{J}_{k-1}}$. In particular, the next theorem shows that the inverse of α_k^{BB1} belongs to the spectrum of the reduced Hessian matrix.

Theorem 2.1 *Let $\{x^{(k)}\}_{k\in\mathbb{N}}$ be the sequence generated by the GP method (1.50) for solving problem (2.1), where A is a SPD matrix. Let $\mathcal{J}_{k-1}$ be the subset of indices defined in (2.8); then we have*

$$\lambda_{min}(A_{\mathcal{J}_{k-1},\mathcal{J}_{k-1}}) \leqslant \frac{1}{\alpha_k^{BB1}} \leqslant \lambda_{max}(A_{\mathcal{J}_{k-1},\mathcal{J}_{k-1}}), \tag{2.13}$$

where $\lambda_{min}(A_{\mathcal{J}_{k-1},\mathcal{J}_{k-1}})$ and $\lambda_{max}(A_{\mathcal{J}_{k-1},\mathcal{J}_{k-1}})$ are the minimum and the maximum eigenvalues of $A_{\mathcal{J}_{k-1},\mathcal{J}_{k-1}}$, respectively.

Proof: We assume that, at the iteration $(k-1)$, the rows/columns of A and the entries of any vector are reordered so that $\mathcal{I}_{k-1}$ is related to the first indices and $\mathcal{J}_{k-1}$ contains the last indices; by dropping for simplicity the iteration counter $k-1$ from $\mathcal{I}_{k-1}$ and $\mathcal{J}_{k-1}$, we can write

$$x^{(k-1)} = \begin{pmatrix} x_{\mathcal{I}}^{(k-1)} \\ x_{\mathcal{J}}^{(k-1)} \end{pmatrix}, \qquad A = \begin{pmatrix} A_{\mathcal{I},\mathcal{I}} & A_{\mathcal{I},\mathcal{J}} \\ A_{\mathcal{I},\mathcal{J}}^T & A_{\mathcal{J},\mathcal{J}} \end{pmatrix}. \tag{2.14}$$

In view of the GP iteration, we have that the entries of the iterate $x^{(k)}$ are

$$x_i^{(k)} = \begin{cases} x_i^{(k-1)} + \nu_{k-1}(p_i^{(k-1)} - x_i^{(k-1)}) & \text{for } i \in \mathcal{I}, \\ x_i^{(k-1)} & \text{for } i \in \mathcal{J}, \end{cases} \tag{2.15}$$

where $p_i^{(k-1)} = \max(\ell_i, \min(x_i^{(k-1)} - \alpha_{k-1} g_i^{(k-1)}, u_i))$, $i \in \mathcal{I}$.
Let $p^{(k-1)}$ denote the vector of length $\sharp\mathcal{I}$ with entries $p_i^{(k-1)}$, $i \in \mathcal{I}$. Then, the vector $s^{(k-1)}$ can be partitioned as follows

$$s^{(k-1)} = \begin{pmatrix} s_{\mathcal{I}}^{(k-1)} \\ s_{\mathcal{J}}^{(k-1)} \end{pmatrix} = \begin{pmatrix} \nu_{k-1}(p^{(k-1)} - x_{\mathcal{I}}^{(k-1)}) \\ 0 \end{pmatrix}. \tag{2.16}$$

In view of (2.15), any entry $g_i^{(k)}, i = 1, \ldots, n$, of the gradient $g^{(k)}$ has the following expression:

$$\begin{aligned} g_i^{(k)} &= \sum_{j=1}^{n} a_{ij} x_j^{(k)} - b_i \\ &= \sum_{j \in \mathcal{I}} a_{ij}(x_j^{(k-1)} + \nu_{k-1}(p_j^{(k-1)} - x_j^{(k-1)})) + \sum_{j \in \mathcal{J}} a_{ij} x_j^{(k-1)} - b_i \\ &= g_i^{(k-1)} + \nu_{k-1} \sum_{j \in \mathcal{I}} a_{ij}(p_j^{(k-1)} - x_j^{(k-1)}). \end{aligned}$$

Consequently, from (2.16), we can write

$$y^{(k-1)} = \begin{pmatrix} y_{\mathcal{I}}^{(k-1)} \\ y_{\mathcal{J}}^{(k-1)} \end{pmatrix} = g^{(k)} - g^{(k-1)} = \begin{pmatrix} A_{\mathcal{I},\mathcal{I}} s_{\mathcal{I}}^{(k-1)} \\ A_{\mathcal{J},\mathcal{I}} s_{\mathcal{J}}^{(k-1)} \end{pmatrix}. \tag{2.17}$$

From definition (2.11), the value of α_k^{BB1} can be written as

$$\alpha_k^{BB1} = \frac{s_{\mathcal{I}}^{(k-1)^T} s_{\mathcal{I}}^{(k-1)}}{(s_{\mathcal{I}}^{(k-1)})^T A_{\mathcal{I},\mathcal{I}} s_{\mathcal{I}}^{(k-1)}}, \tag{2.18}$$

and, therefore, α_k^{BB1} is the reciprocal of a Rayleigh quotient of the submatrix $A_{\mathcal{I},\mathcal{I}}$. From the extremal properties of the Rayleigh quotients the thesis (2.13) follows. □

Theorem 2.1 confirms the natural ability of the BB1 rule in capturing the spectral properties of the reduced Hessian matrix, rather than the whole Hessian, consistently with the information deriving from the KKT conditions at the solution.

REMARK We observed that the presence of the projection operator to ensure the feasibility of each iterates in GP schemes does not allow to prove the validity of any recurrence formula for the gradient's components. This might prevent the desirable sweeping process exhibited by the BB-type methods in the unconstrained quadratic case. Nevertheless, based on the previous analysis, we may argue that the BB1 rule still plays a role in reducing the gradient components with indices belonging to $\mathcal{I}_{k-1}$. In particular, under the special assumption that $\mathcal{I}_k = \mathcal{I}_{k-1}$ and $p_i^{(k)} = (x_i^{(k)} - \alpha_k g_i^{(k)}), i \in \mathcal{I}_k$, we have

$$g_{\mathcal{I}_k}^{(k+1)} = g_{\mathcal{I}_k}^{(k)} + A_{\mathcal{I}_k,\mathcal{I}_k} s_{\mathcal{I}_k}^{(k)} = g_{\mathcal{I}_k}^{(k)} - \nu_k \alpha_k A_{\mathcal{I}_k,\mathcal{I}_k} g_{\mathcal{I}_k}^{(k)}.$$

Let $r = \sharp\mathcal{I}_k$, by denoting with $\gamma_1, \ldots, \gamma_r$ and $u_1, \ldots, u_r$ the eigenvalues and the associated orthonormal eigenvectors of $A_{\mathcal{I}_k,\mathcal{I}_k}$ respectively, we can write

$$g_{\mathcal{I}_k}^{(k+1)} = \sum_{i=1}^{r} \bar{\mu}_i^{(k+1)} u_i \quad \text{and} \quad g_{\mathcal{I}_k}^{(k)} = \sum_{i=1}^{r} \bar{\mu}_i^{(k)} u_i.$$

Hence, similarly to the unconstrained case, we obtain the following recurrence formula for the gradient's eigencomponents:

$$\bar{\mu}_i^{(k+1)} = \bar{\mu}_i^{(k)} (1 - \nu_k \alpha_k \gamma_i), \quad i \in \mathcal{I}_k.$$

As a consequence, in this special case, if the selection rule (2.18) provides a good approximation of $\dfrac{1}{\gamma_i}$, a useful reduction of $|\bar{\mu}_i^{(k+1)}|$ can be achieved.

Let now inspect how the subsets of indices $\mathcal{J}_{k-1}$ and $\mathcal{I}_{k-1}$ can affect, at each iteration, the definition of the BB2 rule. As in the proof of Theorem 2.1, let assume that at the iteration $(k-1)$ the rows/columns of A and the entries of any vector are reordered so that $\mathcal{I}_{k-1}$ is related to the first indices and $\mathcal{J}_{k-1}$ contains the last indices.
If $\mathcal{J}_{k-1} \neq \emptyset$, we have $s^{(k-1)^T} y^{(k-1)} = s_{\mathcal{I}_{k-1}}^{(k-1)^T} y_{\mathcal{I}_{k-1}}^{(k-1)}$, whereas the norm $\|y^{(k-1)}\|^2$ can be splitted into the sum $\|y_{\mathcal{I}_{k-1}}^{(k-1)}\|^2 + \|y_{\mathcal{J}_{k-1}}^{(k-1)}\|^2$. Therefore, the steplength (1.18) can be written as

$$\alpha_k^{BB2} = \frac{s_{\mathcal{I}_{k-1}}^{(k-1)^T} y_{\mathcal{I}_{k-1}}^{(k-1)}}{\|y_{\mathcal{I}_{k-1}}^{(k-1)}\|^2 + \|y_{\mathcal{J}_{k-1}}^{(k-1)}\|^2}, \tag{2.19}$$

and, in view of (2.17), we have

$$\alpha_k^{BB2} = \frac{s_{\mathcal{I}_{k-1}}^{(k-1)^T} A_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}} s_{\mathcal{I}_{k-1}}^{(k-1)}}{s_{\mathcal{I}_{k-1}}^{(k-1)^T} A_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}}^2 s_{\mathcal{I}_{k-1}}^{(k-1)} + s_{\mathcal{I}_{k-1}}^{(k-1)^T} A_{\mathcal{J}_{k-1},\mathcal{I}_{k-1}}^T A_{\mathcal{J}_{k-1},\mathcal{I}_{k-1}} s_{\mathcal{I}_{k-1}}^{(k-1)}}. \tag{2.20}$$

Both the expressions (2.19)-(2.20) emphasize that the steplength provided by the second Barzilai-Borwein rule does not correspond, in

general, to any Rayleigh quotients of $A_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}}$, and, consequently, $1/\alpha_k^{BB2}$ might be outside of the spectrum of the current reduced Hessian at $x^{(k-1)}$.
A simple way to correct (2.19) consists in redefining its value as follows

$$\alpha^{\text{BoxBB2}} = \frac{{s_{\mathcal{I}_{k-1}}^{(k-1)}}^T y_{\mathcal{I}_{k-1}}^{(k-1)}}{\|y_{\mathcal{I}_{k-1}}^{(k-1)}\|^2}. \tag{2.21}$$

For the modified BB2 rule (2.21), hereafter called BoxBB2, the following result holds.

Theorem 2.2 *Let $\{x^{(k)}\}_{k\in\mathbb{N}}$ be the sequence generated by the GP method (1.50) for solving problem (2.1), where A is a SPD matrix. Let $\mathcal{I}_{k-1}$ be the subset of indices defined in (2.8), then we have*

$$\lambda_{min}(A_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}}) \leqslant \frac{1}{\alpha_k^{BB1}} \leqslant \frac{1}{\alpha_k^{BoxBB2}} \leqslant \lambda_{max}(A_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}}), \tag{2.22}$$

where $\lambda_{min}(A_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}})$ and $\lambda_{max}(A_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}})$ are the minimum and the maximum eigenvalues of $A_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}}$, respectively.

Proof: from the definition (2.17) of $y_{\mathcal{I}_{k-1}}^{(k-1)}$ readily follows

$$\lambda_{min}(A_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}}) \leqslant \frac{1}{\alpha_k^{\text{BoxBB2}}} \leqslant \lambda_{max}(A_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}}).$$

Similar arguments to that used in Theorem 2.1 show the remaining inequalities. □

To provide a first insight about the different effect produced by the steplength updating rules BB1, BB2 and BoxBB2 on the GP method, we show the results obtained on a quadratic toy problem subject to lower bounds of size $n = 20$, with ten active constraints at the solution. The eigenvalues of the Hessian matrix are logarithmically distributed in the interval $[1, 500]$ and the spectral condition number is equal to 500.
For the GP implementation we refer to Algorithm 2, with the following parameters setting: $\alpha_{min} = 10^{-10}$, $\alpha_{max} = 10^{6}$, $\alpha_0 = \dfrac{{g^{(0)}}^T g^{(0)}}{{g^{(0)}}^T A g^{(0)}}$.
The feasible initial point $x^{(0)}$ is randomly generated with inactive entries. The stopping criterion is based on the projected gradient (2.5), which allows to check the possible violation of the KKT conditions; in particular, we set

$$\|g^P(x^{(k)})\| \leqslant tol\|g(x^{(0)})\|, \tag{2.23}$$

where $tol = 10^{-8}$.

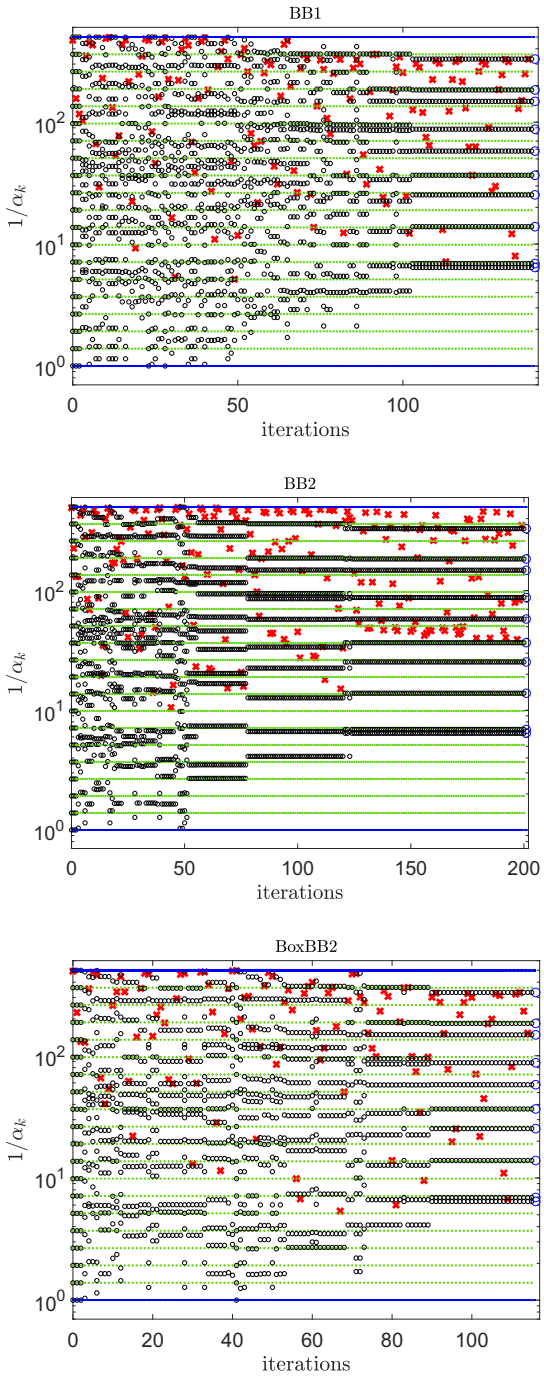

Figure 1: Distribution of $\left\{\frac{1}{\alpha_k}\right\}$ (red crosses) with respect to iterations for BB1 (top panel), BB2 (middle panel) and BoxBB2 (bottom panel) on a toy problem of size $n = 20$.

In Figure 1 the reciprocals of the the steplength α_k generated by each updating rule are reported (red crosses) with respect to the iterations, together with the eigenvalues of the Hessian matrix (green dotted lines) and the eigenvalues of the reduced Hessian matrix (black circles). In each panel of the figure, the blue lines correspond the maximum and the minimum eigenvalues of the whole Hessian matrix, whereas the blue circles at the right of each panel are used to plot the eigenvalues of the submatrix $A_{J*,J*}$.

We observe that the inverses of the steplengths produced by the BB2 method may sometimes fall outside the spectrum of the reduced Hessian matrix, showing also a curious pattern; on the other hand, the sequences $\left\{\frac{1}{\alpha_k}\right\}$ generated by BB1 and BoxBB2 are able to sweep the spectrum of the reduced Hessian matrix. In particular, the modification introduced within the BoxBB2 rule clearly acts as a correction factor for the BB2 rule, providing also a reduction of the iterations needed to satisfy the stopping criterion. This general trend will be confirmed by numerical experiments on randomly generated test problem of larger size (see Chapter 4)).

In view of Theorem 2.2, we propose to exploit the BoxBB2 rule within the alternating strategy (1.32), for an effective combination of short and long steps also in the framework of gradient methods for constrained optimization. Hence, we denote by BoxABB $_{\min}$ the modified ABB$_{\min}$ selection in which the BB2 rule is replaced by the BoxBB2:

$$\alpha_k^{\text{BoxABB}_{\min}} = \begin{cases} \min\left\{\alpha_j^{\text{BoxBB2}} : j = \max\{1, k-m_\alpha\}, \dots, k\right\} & \text{if } \dfrac{\alpha_k^{\text{BoxBB2}}}{\alpha_k^{\text{BB1}}} < \tau \\ \alpha_k^{\text{BB1}} & \text{otherwise} \end{cases} \tag{2.24}$$

where $\tau \in (0,1)$ and m_α is a nonnegative integer.
Furthermore, we will denote by BoxVABB $_{\min}$ the variant of BoxABB $_{\min}$ inspired by the rule (1.34) (1.35) with $\vartheta = 1.1$.
We compare the spectral behaviours of the updating strategies ABB $_{\min}$, BoxABB $_{\min}$ and BoxVABB $_{\min}$ within the GP method on the toy problem. The results obtained for $m_\alpha = 2$ are shown in Figure 2. We observe that the sequence $\{\frac{1}{\alpha_k}\}$ generated by standard ABB$_{\min}$ performs poorly, due to the bad behaviour of the original BB2 steplength. The ehnancement achieved by using the modified rule BoxBB2 within the alternating strategies is clear from the corresponding panels in Figure 2: in both cases, the steplength sequences belong to the spectra of the reduced Hessian matrices, and the numbers of iterations needed to satisfy the stopping criterion are reduced by half with respect to standard ABB$_{\min}$ rule, with a slight improvement for BoxVABB $_{\min}$. In Figure 3 are reported the performances of the alternating strategies for $m_\alpha = 4$; in this case, the standard ABB$_{\min}$ rule shows a behaviour similar to the previous one, while the modified rules are negatively affected by this choice, resulting in an higher number of iterations. A possible explanation is that using steplength values computed at

too far iterations, when the inactive set is not yet stabilized, may not be appropriate. This suggest to exploit small values for the memory parameter m_α, in order to preserve the property that the inverse of the steplength is in the spectrum of the current reduced Hessian.

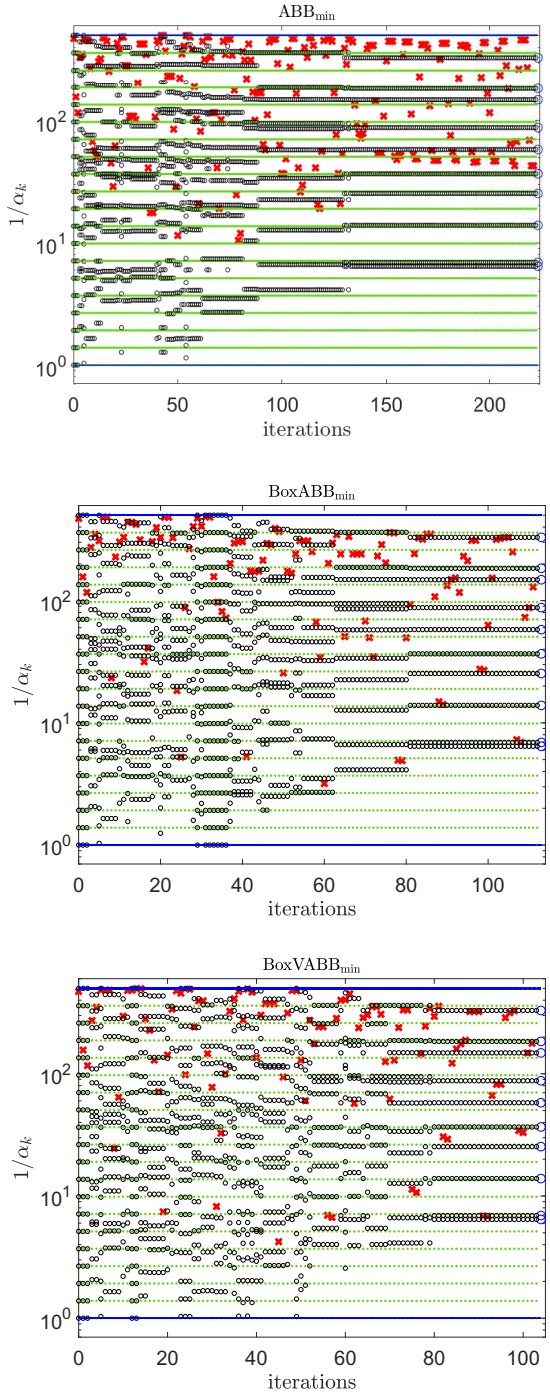

Figure 2: Distribution of $\frac{1}{\alpha_k}$ with respect to iterations for ABB$_{\min}$ (top panel), BoxABB$_{\min}$ (middle panel) and BoxVABB$_{\min}$ (bottom panel), with $m_\alpha = 2$, on a toy problem of size $n = 20$.

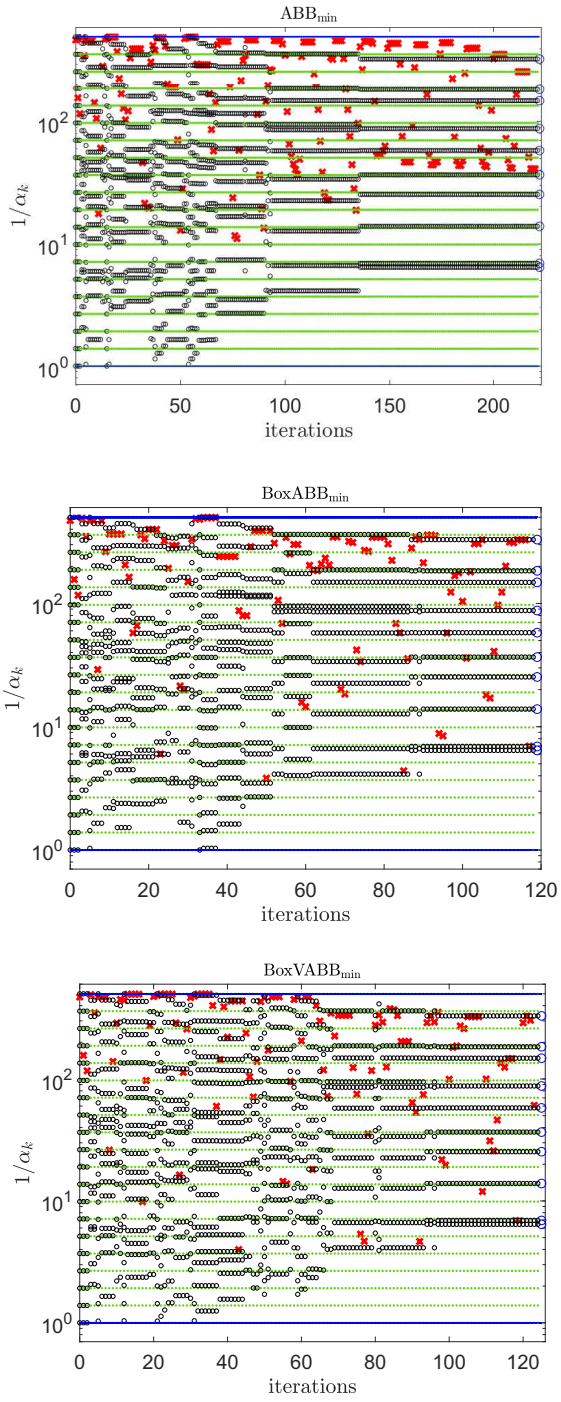

Figure 3: Distribution of $\left\{\frac{1}{\alpha_k}\right\}$ (red crosses) with respect to iterations for ABB$_{\min}$ (top panel), BoxABB$_{\min}$ (middle panel) and BoxVABB$_{\min}$ (bottom panel), with $m_\alpha = 4$, on a toy problem of size $n = 20$.

2.3 A LIMITED MEMORY APPROACH IN GP METHODS FOR BQP PROBLEMS

In this section we investigate the spectral properties of the gradient projection method combined with the steplength selection rule based on Ritz values, proposed in [45].

When we solve a problem of type (2.1) by means of the GP method (1.50), the procedure to build the matrix G defined in (1.36) using the whole gradients may not be convenient; indeed, in this case, the recurrence (1.9) does not hold; as a consequence, the matrix obtained by procedure (1.41) may not exhibit a tridiagonal structure, but it is an upper Hessenberg matrix.

The previous analysis on the BB-based rules suggests that suitable rules based on the information related to the inactive constraints can improve the performance of the method. Indeed, when the set of the inactive variables starts to get settled, a recurrence formula for the gradient's components can be recovered again. A first attempt at suggesting a possible way to employ the limited memory steplength selection rule in the framework of box-constrained optimization was done in [69]. In particular, the authors proposed to consider a generalized matrix T, called $\tilde{T}$, defined as

$$\tilde{T} = \tilde{R}^{-T}\tilde{G}^{T}\,[\tilde{G}\ \tilde{g}^{(k+m)}]\,\tilde{J}\,\tilde{R}^{-1},$$

where the vector $\tilde{g}^{(k)}$ is given by

$$\tilde{g}_i^{(k)} = \begin{cases} g_i^{(k)} & \text{if } \ell_i < x_i^{(k)} < u_i \\ 0 & \text{otherwise} \end{cases}, \tag{2.25}$$

$\tilde{G} = [\tilde{g}^{(k)}\ \tilde{g}^{(k+1)}\ \cdots\ \tilde{g}^{(k+m-1)}]$, $\tilde{R}$ is such that $\tilde{R}^{T}\tilde{R} = \tilde{G}^{T}\tilde{G}$ and $\tilde{J}$ is the $(m+1)$-by-m lower bidiagonal matrix

$$\tilde{J} = \begin{bmatrix} \frac{1}{\alpha_k \nu_k} & & \\ -\frac{1}{\alpha_k \nu_k} & \ddots & \\ & \ddots & \frac{1}{\alpha_{k+m-1}\nu_{k+m-1}} \\ & & -\frac{1}{\alpha_{k+m-1}\nu_{k+m-1}} \end{bmatrix}. \tag{2.26}$$

However, in [69] the spectral properties of the steplengths generated by the suggested approach were not inspected. Thus, we propose a special analysis, clarifying the relation between subsequent gradients in a sweep. The idea is to introduce an adaptive strategy for setting the length of a sweep during the iterative procedure, based on the gradient components which provide feasible steps in a sweep. This technique is consistent with the strategy we have adopted for the gradient projection method equipped with BB-based rules: we select only

some portions of the gradient to compute the steplengths, attempting to approximate the inverse of the eigenvalues of a proper submatrix of the Hessian matrix of the objective function, instead of considering the whole Hessian matrix.

At any iteration, let consider the following subsets of $\mathcal{N}$:

$$\begin{aligned} \mathcal{F}_k &= \{i \in \mathcal{N} : \ \ell_i \leqslant x_i^{(k)} - \alpha_k g_i^{(k)} \leqslant u_i\}, \\ \mathcal{B}_k &= \mathcal{N} \setminus \mathcal{F}_k \ . \end{aligned} \tag{2.27}$$

REMARK We observe that the subsets defined in (2.27) are different from the subsets $\mathcal{I}_k$ and $\mathcal{J}_k$ introduced in the previous section; in particular, we have

$$\mathcal{F}_k \subseteq \mathcal{I}_k \quad \text{and} \quad \mathcal{J}_k \subseteq \mathcal{B}_k, \ \ k = 0, 1, \dots \ .$$

Based on (2.27), the entries of the iterate $x^{(k+1)}$ generated by the GP method (1.50) are

$$x_i^{(k+1)} = \begin{cases} x_i^{(k)} + \nu_k \left(x_i^{(k)} - \alpha_k g_i^{(k)} - x_i^{(k)} \right) & i \in \mathcal{F}_k, \\ x_i^{(k)} + \nu_k \left(\gamma_i^{(k)} - x_i^{(k)} \right) & i \in \mathcal{B}_k, \end{cases} \tag{2.28}$$

where

$$\gamma_i^{(k)} = \begin{cases} \ell_i & \text{if } x_i^{(k)} - \alpha_k g_i^{(k)} < \ell_i, \\ u_i & \text{if } x_i^{(k)} - \alpha_k g_i^{(k)} > u_i. \end{cases}$$

As a consequence, for any $i = 1, \dots, n$, the new gradient components are given by

$$\begin{aligned} g_i^{(k+1)} &= \sum_{j=1}^{n} a_{ij} x_j^{(k+1)} - b_i = \\ &= \sum_{j \in \mathcal{F}_k} a_{ij} \left(x_j^{(k)} - \nu_k \alpha_k g_j^{(k)} \right) + \sum_{j \in \mathcal{B}_k} a_{ij} \left(x_j^{(k)} - \nu_k (x_j^{(k)} - \gamma_j^{(k)}) \right) - b_i = \\ &= g_i^{(k)} - \nu_k \alpha_k \sum_{j \in \mathcal{F}_k} a_{ij} g_j^{(k)} - \nu_k \alpha_k \sum_{j \in \mathcal{B}_k} a_{ij} \frac{x_j^{(k)} - \gamma_j^{(k)}}{\alpha_k}. \end{aligned}$$

From the previous equation we can write

$$A_{\mathcal{F}_k, \mathcal{F}_k} g_{\mathcal{F}_k}^{(k)} = \begin{bmatrix} g_{\mathcal{F}_k}^{(k)} & g_{\mathcal{F}_k}^{(k+1)} \end{bmatrix} \begin{bmatrix} \frac{1}{\alpha_k \nu_k} \\ -\frac{1}{\alpha_k \nu_k} \end{bmatrix} - A_{\mathcal{F}_k, \mathcal{N}} p^{(k)}, \tag{2.29}$$

where $p^{(k)}$ is a vector with n entries defined component-wise as

$$p_i^{(k)} = \begin{cases} 0 & i \in \mathcal{F}_k, \\ \frac{x_i^{(k)} - \gamma_i^{(k)}}{\alpha_k} & i \in \mathcal{B}_k. \end{cases} \tag{2.30}$$

At the next iteration, using the same argument employed to obtain (2.29), we get

$$A_{\mathcal{F}_{k+1},\mathcal{F}_{k+1}} g^{(k+1)}_{\mathcal{F}_{k+1}} = \begin{bmatrix} g^{(k+1)}_{\mathcal{F}_{k+1}} & g^{(k+2)}_{\mathcal{F}_{k+1}} \end{bmatrix} \begin{bmatrix} \frac{1}{\alpha_{k+1}\nu_{k+1}} \\ -\frac{1}{\alpha_{k+1}\nu_{k+1}} \end{bmatrix} - A_{\mathcal{F}_{k+1},\mathcal{N}} p^{(k+1)}, \tag{2.31}$$

with the obvious definitions for $\mathcal{F}_{k+1}$ and $\mathcal{B}_{k+1}$. Under the assumption $\mathcal{F}_k \cap \mathcal{F}_{k+1} \neq \emptyset$, we consider the following subsets of indices by taking into account all the possible cases that may occur at the $(k+1)$-th iteration:

$$\begin{aligned} \mathcal{F}_{(k,k+1)} &:= \mathcal{F}_k \cap \mathcal{F}_{k+1}, \\ \overline{\mathcal{F}}^{k}_{(k,k+1)} &:= \mathcal{F}_k \backslash (\mathcal{F}_k \cap \mathcal{F}_{k+1}), \\ \overline{\mathcal{F}}^{k+1}_{(k,k+1)} &:= \mathcal{F}_{k+1} \backslash (\mathcal{F}_k \cap \mathcal{F}_{k+1}). \end{aligned}$$

From (2.29) and (2.31), we may write

$$\begin{aligned} & A_{\mathcal{F}_{(k,k+1)},\mathcal{F}_{(k,k+1)}} \begin{bmatrix} g^{(k)}_{\mathcal{F}_{(k,k+1)}} & g^{(k+1)}_{\mathcal{F}_{(k,k+1)}} \end{bmatrix} = \\ & = \begin{bmatrix} g^{(k)}_{\mathcal{F}_{(k,k+1)}} & g^{(k+1)}_{\mathcal{F}_{(k,k+1)}} & g^{(k+2)}_{\mathcal{F}_{(k,k+1)}} \end{bmatrix} \begin{bmatrix} \frac{1}{\alpha_k \nu_k} & 0 \\ -\frac{1}{\alpha_k \nu_k} & \frac{1}{\alpha_{k+1}\nu_{k+1}} \\ 0 & -\frac{1}{\alpha_{k+1}\nu_{k+1}} \end{bmatrix} + \\ & - A_{\mathcal{F}_{(k,k+1)},\mathcal{N}} \begin{bmatrix} p^{(k)} & p^{(k+1)} \end{bmatrix} + \\ & - \begin{bmatrix} A_{\mathcal{F}_{(k,k+1)},\overline{\mathcal{F}}^{k}_{(k,k+1)}} g^{(k)}_{\overline{\mathcal{F}}^{k}_{(k,k+1)}} & A_{\mathcal{F}_{(k,k+1)},\overline{\mathcal{F}}^{k+1}_{(k,k+1)}} g^{(k+1)}_{\overline{\mathcal{F}}^{k+1}_{(k,k+1)}} \end{bmatrix}. \end{aligned}$$

The argument can be generalized to a sweep of length m starting from the iteration k. By defining $\mathcal{F}_{(k,k+m-1)} := \cap_{s=k}^{k+m-1} \mathcal{F}_s$, we have

$$\begin{aligned} & A_{\mathcal{F}_{(k,k+m-1)},\mathcal{F}_{(k,k+m-1)}} \begin{bmatrix} g^{(k)}_{\mathcal{F}_{(k,k+m-1)}} & \cdots & g^{(k+m-1)}_{\mathcal{F}_{(k,k+m-1)}} \end{bmatrix} = \\ & = \begin{bmatrix} g^{(k)}_{\mathcal{F}_{(k,k+m-1)}} & \cdots & g^{(k+m-1)}_{\mathcal{F}_{(k,k+m-1)}} & g^{(k+m)}_{\mathcal{F}_{(k,k+m-1)}} \end{bmatrix} \tilde{J} + \\ & - A_{\mathcal{F}_{(k,k+m-1)},\mathcal{N}} \begin{bmatrix} p^{(k)} & \cdots & p^{(k+m-1)} \end{bmatrix} + \\ & - \begin{bmatrix} A_{\mathcal{F}_{(k,k+m-1)},\overline{\mathcal{F}}^{k}_{(k,k+m-1)}} g^{(k)}_{\overline{\mathcal{F}}^{k}_{(k,k+m-1)}} & \cdots & A_{\mathcal{F}_{(k,k+m-1)},\overline{\mathcal{F}}^{k+m-1}_{(k,k+m-1)}} g^{(k+m-1)}_{\overline{\mathcal{F}}^{k+m-1}_{(k,k+m-1)}} \end{bmatrix}, \end{aligned}$$

where $\tilde{J}$ is the $(m+1)$-by-m lower bidiagonal matrix given in (2.26). If $\mathcal{F}_{k+j} \subseteq \mathcal{F}_{(k,k+m-1)}$, for $j = 0, \ldots, m-1$, the term

$$A_{\mathcal{F}_{(k,k+m-1)},\overline{\mathcal{F}}^{k+j}_{k,k+m-1}} g^{(k+j)}_{\mathcal{F}_{(k,k+m-1)},\overline{\mathcal{F}}^{k+j}_{k,k+m-1}}$$

does not contribute to the previous relation, which can be rewritten as

$$\begin{aligned} & A_{\mathcal{F}_{(k,k+m-1)},\mathcal{F}_{(k,k+m-1)}} \left[g^{(k)}_{\mathcal{F}_{(k,k+m-1)}} \quad \cdots \quad g^{(k+m-1)}_{\mathcal{F}_{(k,k+m-1)}} \right] = \\ & = \left[g^{(k)}_{\mathcal{F}_{(k,k+m-1)}} \quad \cdots \quad g^{(k+m-1)}_{\mathcal{F}_{(k,k+m-1)}} \quad g^{(k+m)}_{\mathcal{F}_{(k,k+m-1)}} \right] \tilde{J} + \\ & \quad - A_{\mathcal{F}_{(k,k+m-1)},\mathcal{N}} \left[p^{(k)} \quad \cdots \quad p^{(k+m-1)} \right]. \end{aligned} \tag{2.32}$$

In order to preserve the validity of (2.32) and correctly neglect the term $A_{\mathcal{F}_{(k,k+m-1)},\overline{\mathcal{F}}^{k+j}_{k,k+m-1}} g^{(k+j)}_{\mathcal{F}_{(k,k+m-1)},\overline{\mathcal{F}}^{k+j}_{k,k+m-1}}$, $j = 0, \ldots, m-1$, we propose to interrupt a sweep and to restart the collection of new restricted gradient vectors when the condition

$$\mathcal{F}_{k+j} \subseteq \mathcal{F}_{(k,k+m-1)}, \quad j = 0, \ldots, m-1, \tag{2.33}$$

is not satisfied. In this way we develop a technique which *adaptively* controls the length of the sweep, up to the given value m. Typically, at the beginning of the iterative process, this condition does not hold and the length of the first sweeps is equal to 1, at most; however, as the number of iterations increases, the components that are going to be projected onto the feasible set Ω tend to stabilize and, as a consequence, condition (2.33) starts to occur for a growing number of iterations.

Hereafter, let suppose that (2.33) holds. The equality (2.32) may be considered as a possible extension of the equation (1.37) that holds in the unconstrained framework. Hence, in presence of box-constraints, we suggest to avoid the storage of m back *whole* gradients vectors, by considering m back gradients restricted to the set of indices $\mathcal{F}_{(k,k+m-1)}$. Driven by these considerations, our implementation of the limited memory steplength rule for the constrained case is based on the following submatrix of the matrix G:

$$G_{(k,k+m-1)} = \left[g^{(k)}_{\mathcal{F}_{(k,k+m-1)}} \quad \cdots \quad g^{(k+m-1)}_{\mathcal{F}_{(k,k+m-1)}} \right]. \tag{2.34}$$

Given $m \geqslant 1$ and the $m \times m$ matrix $R_{(k,k+m-1)}$ such that

$$R^T_{(k,k+m-1)} R_{(k,k+m-1)} = G^T_{(k,k+m-1)} G_{(k,k+m-1)},$$

we propose to compute, at each new sweep, m steplengths as inverses of the eigenvalues of the symmetric matrix

$$\tilde{T}_{(k,k+m-1)} = R^{-T}_{(k,k+m-1)} G^T_{(k,k+m-1)} A_{\mathcal{F}_{(k,k+m-1)},\mathcal{F}_{(k,k+m-1)}} G_{(k,k+m-1)} R^{-1}_{(k,k+m-1)}, \tag{2.35}$$

with the aim of approximating the inverses of the eigenvalues of the matrix $A_{\mathcal{F}_{(k,k+m-1)},\mathcal{F}_{(k,k+m-1)}}$.

REMARK This idea mimics the approach proposed for the BB-based rules in the box-constrained case. Indeed, under the special assumptions $m = 1$, $\mathcal{F}_{k-1} = \mathcal{F}_k$ and $\nu_{k-1} = 1$, in view of $\gamma^{(k)}_{\mathcal{B}_k} = x^{(k)}_{\mathcal{B}_k}$, the recurrence (2.29) can be simplified as

$$g^{(k+1)}_{\mathcal{F}_k} = g^{(k)}_{\mathcal{F}_k} - \nu_k \alpha_k A_{\mathcal{F}_k,\mathcal{F}_k} g^{(k)}_{\mathcal{F}_k}.$$

Let denote with $\{\delta_1, \ldots, \delta_r\}$ and $\{w_1, \ldots, w_r\}$, respectively, the eigenvalues and the associated orthonormal eigenvectors of $A_{\mathcal{F}_k,\mathcal{F}_k}$ where $r = \sharp\mathcal{F}_k$; by writing $g^{(k+1)}_{\mathcal{F}_k} = \sum_{i=1}^{r} \bar{\mu}_i^{(k+1)} w_i$ and $g^{(k)}_{\mathcal{F}_k} = \sum_{i=1}^{r} \bar{\mu}_i^{(k)} w_i$, we obtain the following recurrence formula for the eigencomponents:

$$\bar{\mu}_i^{(k+1)} = \bar{\mu}_i^{(k)} (1 - \nu_k \alpha_k \delta_i).$$

This means that if the selection rule provides a good approximation of $\frac{1}{\delta_i}$, a useful reduction of $|\bar{\mu}_i^{(k+1)}|$ can be achieved.
We underline that, if $m = 1$, α_k is computed in order to estimate the inverse of an eigenvalue of $A_{\mathcal{F}_{k-1},\mathcal{F}_{k-1}}$; obviously, if $\mathcal{F}_{k-1} = \mathcal{F}_k$, α_k can also provide a good approximation of $\frac{1}{\delta_i}$ and thus reduce the corresponding component $|\bar{\mu}_i^{(k+1)}|$.

In view of (2.32), the matrix $\tilde{T}_{(k,k+m-1)}$ defined in (2.35) has the following form

$$\begin{aligned}
\tilde{T}_{(k,k+m-1)} &= R^{-T}_{(k,k+m-1)} G^{T}_{(k,k+m-1)} \left[G_{(k,k+m-1)} \; g^{(k+m)}_{\mathcal{F}_{(k,k+m-1)}} \right] \cdot \tilde{J} R^{-1}_{(k,k+m-1)} + \\
&\quad - R^{-T}_{(k,k+m-1)} G^{T}_{(k,k+m-1)} A_{\mathcal{F}_{(k,k+m-1)},\mathcal{N}} \cdot \left[p^{(k)} \quad \ldots \quad p^{(k+m-1)} \right] R^{-1}_{(k,k+m-1)} = \\
&= \left[R_{(k,k+m-1)}, \quad r_{(k,k+m-1)} \right] \tilde{J} R^{-1}_{(k,k+m-1)} + \\
&\quad + R^{-T}_{(k,k+m-1)} G^{T}_{(k,k+m-1)} A_{\mathcal{F}_{(k,k+m-1)},\mathcal{N}} \cdot \left[p^{(k)} \quad \ldots p^{(k+m-1)} \right] R^{-1}_{(k,k+m-1)},
\end{aligned} \tag{2.36}$$

where the vector $r_{(k,k+m-1)}$ is the solution of the system

$$R^{T}_{(k,k+m-1)} r_{(k,k+m-1)} = G^{T}_{(k,k+m-1)} g^{(k+m)}_{\mathcal{F}_{(k,k+m-1)}}.$$

From a practical point of view, we want to avoid to explicitly use the matrix $A_{\mathcal{F}_{(k,k+m-1)},\mathcal{N}}$; hence, we do not consider the exact relation (2.32), but its inexact version where the term

$$A_{\mathcal{F}_{(k,k+m-1)},\mathcal{N}} \left[p^{(k)} \quad \ldots \quad p^{(k+m-1)} \right]$$

is neglected. For this reason, we do not compute the eigenvalues of $\tilde{T}_{(k,k+m-1)}$ but the eigenvalues of the symmetric part of the matrix

$$Z_{(k,k+m-1)} = \left[R_{(k,k+m-1)}, r_{(k,k+m-1)} \right] \tilde{J} R^{-1}_{(k,k+m-1)}.$$

To explain the relation between the eigenvalues of $\tilde{T}_{(k,k+m-1)}$ and those of the symmetric part of $Z_{(k,k+m-1)}$, we start to clarify the

details of our approach in the easier case of $m = 1$, where $\tilde{T}_{(k,k+m-1)}$ reduces to a scalar.
In this case, at iteration $k+1$ only one gradient is available $G_k = g^{(k)}_{\mathcal{F}_k}$. We are interested in computing

$$\tilde{T}_k = R_k^{-T} G_k^T A_{\mathcal{F}_k,\mathcal{F}_k} G_k R_k^{-1},$$

where

$$G_k^T G_k = {g^{(k)}_{\mathcal{F}_k}}^T g^{(k)}_{\mathcal{F}_k} = \sqrt{(g^{(k)}_{\mathcal{F}_k})^T g^{(k)}_{\mathcal{F}_k}} \sqrt{(g^{(k)}_{\mathcal{F}_k})^T g^{(k)}_{\mathcal{F}_k}} = R_k^T R_k.$$

Then, using (2.29), the matrix $\tilde{T}_k$ is given by

$$R_k^{-T} G_k^T \left(\begin{bmatrix} g^{(k)}_{\mathcal{F}_k} & g^{(k+1)}_{\mathcal{F}_k} \end{bmatrix} \begin{bmatrix} \frac{1}{\alpha_k \nu_k} \\ -\frac{1}{\alpha_k \nu_k} \end{bmatrix} - A_{\mathcal{F}_k,\mathcal{N}} p^{(k)} \right) R_k^{-1} =$$

$$= \frac{{g^{(k)}_{\mathcal{F}_k}}^T}{\sqrt{{g^{(k)}_{\mathcal{F}_k}}^T g^{(k)}_{\mathcal{F}_k}}} \left(\frac{g^{(k)}_{\mathcal{F}_k} - g^{(k+1)}_{\mathcal{F}_k}}{\alpha_k \nu_k} - A_{\mathcal{F}_k,\mathcal{N}} p^{(k)} \right) \frac{1}{\sqrt{{g^{(k)}_{\mathcal{F}_k}}^T g^{(k)}_{\mathcal{F}_k}}} \quad (2.37)$$

$$= \frac{{g^{(k)}_{\mathcal{F}_k}}^T A_{\mathcal{F}_k,\mathcal{F}_k} g^{(k)}_{\mathcal{F}_k}}{{g^{(k)}_{\mathcal{F}_k}}^T g^{(k)}_{\mathcal{F}_k}}.$$

Hence, in the special case $m = 1$, if we consider the exact expression of $A_{\mathcal{F}_k,\mathcal{F}_k} G_k$ given by the right-hand side of (2.29), we obtain as unique eigenvalue of $\tilde{T}_k$ the value $\frac{{g^{(k)}_{\mathcal{F}_k}}^T A_{\mathcal{F}_k,\mathcal{F}_k} g^{(k)}_{\mathcal{F}_k}}{{g^{(k)}_{\mathcal{F}_k}}^T g^{(k)}_{\mathcal{F}_k}}$, which is the inverse of the Rayleigh quotient of the matrix $A_{\mathcal{F}_k,\mathcal{F}_k}$.
However, in practice we compute the scalar

$$Z_k = \quad = \frac{(g^{(k)}_{\mathcal{F}_k})^T A_{\mathcal{F}_k,\mathcal{F}_k} g^{(k)}_{\mathcal{F}_k}}{(g^{(k)}_{\mathcal{F}_k})^T g^{(k)}_{\mathcal{F}_k}} + \frac{(g^{(k)}_{\mathcal{F}_k})^T A_{\mathcal{F}_k,\mathcal{N}} p^{(k)}}{(g^{(k)}_{\mathcal{F}_k})^T g^{(k)}_{\mathcal{F}_k}} \quad (2.38)$$

that is a value in the spectrum of $A_{\mathcal{F}_k,\mathcal{F}_k}$ affected by en error, due to the presence of the second term at the right-hand side of equation (2.38). An estimation of this error, at iteration $k+1$, is given by

$$\rho_k \leqslant \frac{\|A_{\mathcal{F}_k,\mathcal{N}} p^{(k)}\|}{\|g^{(k)}_{\mathcal{F}_k}\|}.$$

From equations (2.28) and (2.30), the following results hold

$$\|p^{(k)}\| = \|p^{(k)}_{\mathcal{B}_k}\| = \frac{\|x^{(k+1)}_{\mathcal{B}_k} - x^{(k)}_{\mathcal{B}_k}\|}{\alpha_k \nu_k} \quad \text{and} \quad \|g^{(k)}_{\mathcal{F}_k}\| = \frac{\|x^{(k+1)}_{\mathcal{F}_k} - x^{(k)}_{\mathcal{F}_k}\|}{\alpha_k \nu_k}.$$

As a consequence,

$$\rho_k \leqslant \frac{\|A_{\mathcal{F}_k,\mathcal{B}_k} p^{(k)}_{\mathcal{B}_k}\|}{\|g^{(k)}_{\mathcal{F}_k}\|} \leqslant \frac{\|A_{\mathcal{F}_k,\mathcal{B}_k}\| \|p^{(k)}_{\mathcal{B}_k}\|}{\|g^{(k)}_{\mathcal{F}_k}\|} = \frac{\|A_{\mathcal{F}_k,\mathcal{B}_k}\| \|x^{(k+1)}_{\mathcal{B}_k} - x^{(k)}_{\mathcal{B}_k}\|}{\|x^{(k+1)}_{\mathcal{F}_k} - x^{(k)}_{\mathcal{F}_k}\|}.$$

From this bound on ρ_k, we can state that, when k is sufficiently large so as the components to project onto the feasible region are almost settled, the error ρ_k is negligible and the steplength α_{k+1} is approximating the inverse of an eigenvalue of $A_{\mathcal{F}_k,\mathcal{F}_k}$. In the more general case $m > 1$, we compute the eigenvalues of the symmetric part of $Z_{(k,k+m-1)}$ given by

$$\tilde{Z}_{(k,k+m-1)} = \frac{1}{2}\left(Z_{(k,k+m-1)} + Z^T_{(k,k+m-1)}\right). \tag{2.39}$$

Then, from equation (2.36) we have

$$\begin{aligned}\tilde{Z}_{(k,k+m-1)} &= \tilde{T}_{(k,k+m-1)} + \\ &+ \frac{1}{2}\left(R^{-T}_{(k,k+m-1)} G^T_{(k,k+m-1)} A_{\mathcal{F}_{(k,k+m-1)},\mathcal{N}} \left[p^{(k)} \cdots p^{(k+m-1)}\right] R^{-1}_{(k,k+m-1)}\right) + \\ &+ \frac{1}{2}\left(R^{-T}_{(k,k+m-1)} \left[p^{(k)} \cdots p^{(k+m-1)}\right]^T A^T_{\mathcal{F}_{(k,k+m-1)},\mathcal{N}} \cdot G_{(k,k+m-1)} R^{-1}_{(k,k+m-1)}\right).\end{aligned} \tag{2.40}$$

A result of perturbation matrix theory (see Corollary 6.3.4 [55]) ensures that

$$\left|\lambda_j\left(\tilde{Z}_{(k,k+m-1)}\right) - \lambda_j\left(\tilde{T}_{(k,k+m-1)}\right)\right| \leqslant \left\|\tilde{Z}_{(k,k+m-1)} - \tilde{T}_{(k,k+m-1)}\right\|, \tag{2.41}$$

where $\lambda_j(C)$ is the $j-$th eigenvalue of C. By denoting with $\|D\|_F$ the Frobenius norm of a matrix D, the right-hand side of (2.41) can be bounded from above as

$$\begin{aligned}&\left\|\tilde{Z}_{(k,k+m-1)} - \tilde{T}_{(k,k+m-1)}\right\| \leqslant \\ &\quad\leqslant \left\|R^{-T}_{(k,k+m-1)} G^T_{(k,k+m-1)} A_{\mathcal{F}_{(k,k+m-1)},\mathcal{N}} \left[p^{(k)} \quad \cdots \quad p^{(k+m-1)}\right] R^{-1}_{(k,k+m-1)}\right\| \leqslant \\ &\quad\quad\leqslant \left\|A_{\mathcal{F}_{(k,k+m-1)},\mathcal{N}} \left[p^{(k)} \quad \cdots \quad p^{(k+m-1)}\right]\right\| \|R^{-1}_{(k,k+m-1)}\| \leqslant \\ &\quad\quad\leqslant \left\|A_{\mathcal{F}_{(k,k+m-1)},\mathcal{N}}\right\| \left\|\left[p^{(k)} \quad \cdots \quad p^{(k+m-1)}\right]\right\| \|R^{-1}_{(k,k+m-1)}\| \leqslant \\ &\quad\quad\leqslant \left\|A_{\mathcal{F}_{(k,k+m-1)},\mathcal{N}}\right\| \left\|\left[p^{(k)} \quad \cdots \quad p^{(k+m-1)}\right]\right\|_F \|R^{-1}_{(k,k+m-1)}\| \leqslant \\ &\quad\quad\leqslant \left\|A_{\mathcal{F}_{(k,k+m-1)},\mathcal{N}}\right\| \sqrt{\sum_{i=0}^{m-1} \left\|p^{(k+i)}_{\mathcal{B}_{k+i}}\right\|^2} \|R^{-1}_{(k,k+m-1)}\|,\end{aligned}$$

where in the second inequality we used

$$\|R^{-T}_{(k,k+m-1)} G^T_{(k,k+m-1)}\| = \sqrt{\|R^{-T}_{(k,k+m-1)} G^T_{(k,k+m-1)} G_{(k,k+m-1)} R^{-1}_{(k,k+m-1)}\|} = 1.$$

We can conclude that, if m is relatively small, the matrix $\tilde{Z}_{(k,k+m-1)}$ approaches the matrix $\tilde{T}_{(k,k+m-1)}$ as the number of iterations k increases.

We now report a first result on the behaviour of the the described approach on the toy problem of size $n = 20$ considered in Section 2.2 In particular we compare the behaviour of the GP method combined with

- the original limited memory steplength selection rule that collects the whole back gradients, here named LMGP;
- the modified limited memory steplength selection rule suggested in [69], which considers the modified gradients given in (2.25), named Box-LMPG1;
- the modified LM steplength selection rule suggested in this section exploiting the matrix $G_{(k,k+m-1)}$ defined in (2.34), which we refer to Box-LMGP2.

The considered methods share a monotone Armijo linesearch (1.27) and the stopping criterion defined by (2.23), with $\mathtt{tol} = 10^{-8}$.

In Figure 4-5-6 we report the behaviour of $\left\{\frac{1}{\alpha_k}\right\}$ (red crosses) with respect to the eigenvalues of the Hessian matrix (green dotted lines) and the restricted Hessian submatrix (black circles) at each iteration k, for $m = 3, 5, 7$. We observe that the inverses of the steplengths produced by the LMGP method may sometimes fall outside the spectrum of the restricted Hessian or even the spectrum of the whole Hessian, while the other two approaches are able to restrain this effect. In particular, the sequence $\left\{\frac{1}{\alpha_k}\right\}$ generated by the Box-LMGP2 scheme, belongs to the spectra of the current restricted Hessian matrices, providing also a reduction of the iterations needed to satisfy the stopping criterion. Indeed, the effectiveness of the Box-LMGP2 procedure allows an earlier stabilization of the active set, with respect to the other two approaches, and seems to be less sensitive to the length of the sweep thanks to the adaptive strategy.

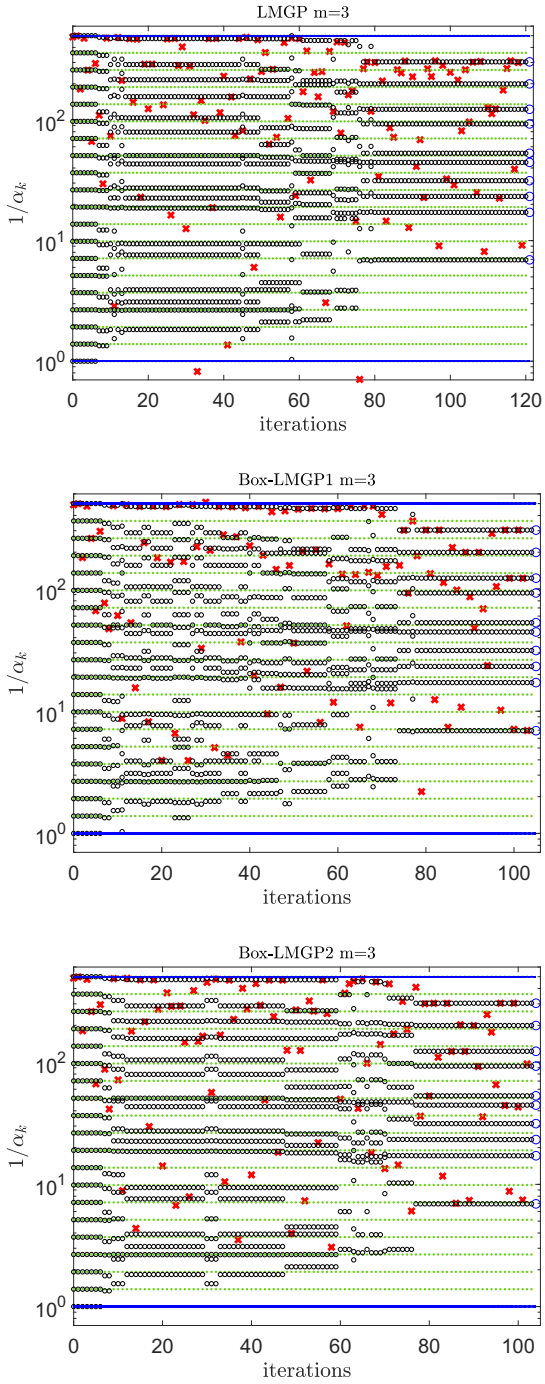

Figure 4: Distribution of $\left\{\frac{1}{\alpha_k}\right\}$ (red crosses) with respect to iterations for LMGP (top panel), Box-LMGP1 (middle panel) and Box-LMGP2 (bottom panel) on a toy problem of size $n = 20$, for $m = 3$.

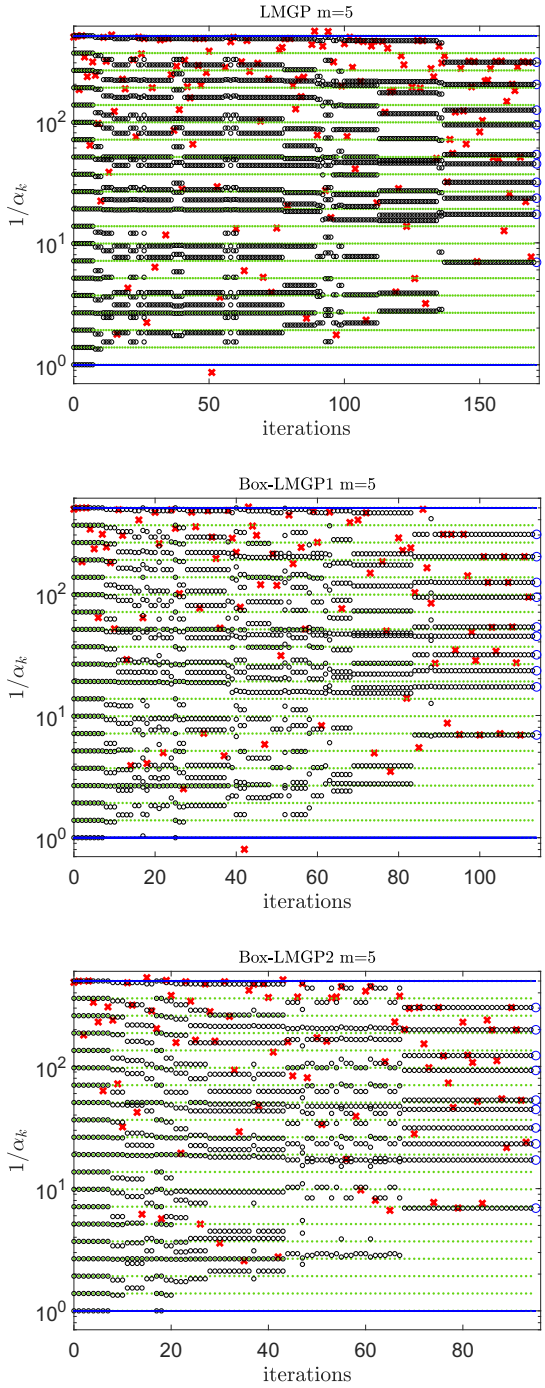

Figure 5: Distribution of $\left\{\frac{1}{\alpha_k}\right\}$ (red crosses) with respect to iterations for LMGP (top panel), Box-LMGP1 (middle panel) and Box-LMGP2 (bottom panel) on a toy problem of size $n = 20$, for $m = 5$.

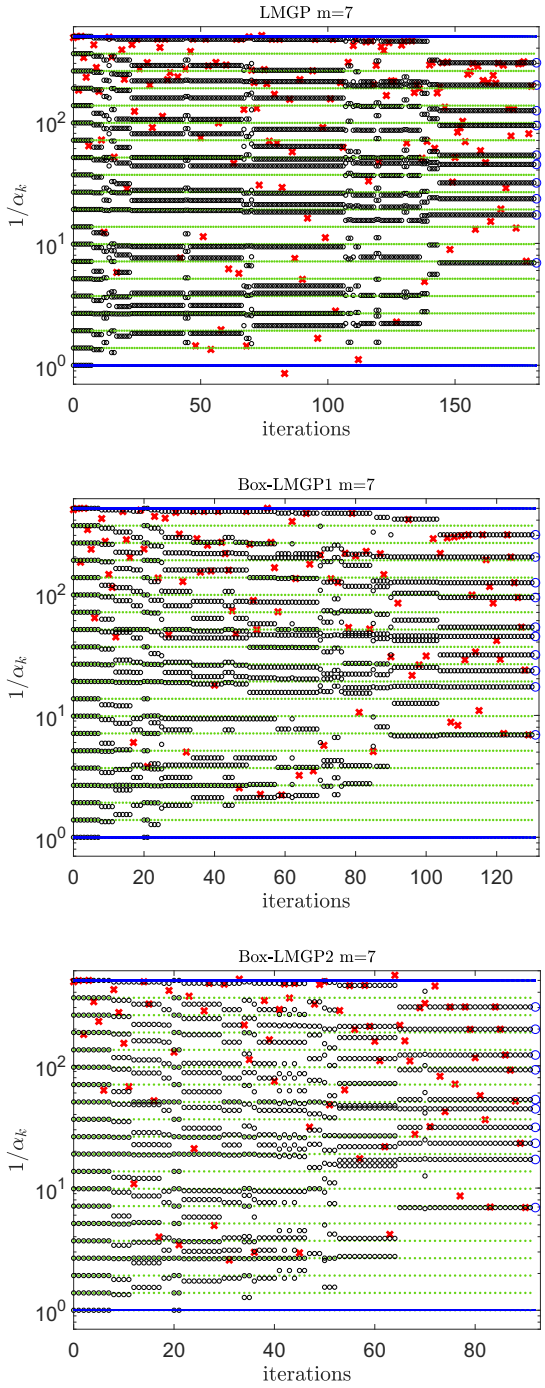

Figure 6: Distribution of $\left\{\frac{1}{\alpha_k}\right\}$ (red crosses) with respect to iterations for LMGP (top panel), Box-LMGP1 (middle panel) and Box-LMGP2 (bottom panel) on a toy problem of size $n = 20$, for $m = 7$.

3

STEPLENGTH SELECTION IN GP METHODS FOR SINGLY LINEARLY CONSTRAINED PROBLEMS SUBJECT TO LOWER AND UPPER BOUNDS

This chapter is dedicated to extend the spectral analysis of the GP method to a more general class of constrained optimization problems, which differs from the previous one due to the presence of an additional linear equality constraint:

$$\min_{x\in\Omega} f(x), \tag{3.1}$$

where $f(x)\colon \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable function,

$$\Omega = \{x \in \mathbb{R}^n \,:\, \ell \leqslant x \leqslant u, \quad v^T x = e\}, \tag{3.2}$$

and $\ell, u, v \in \mathbb{R}^n$, $e \in \mathbb{R}$. The feasible region Ω is assumed not empty. We refer to (3.1) as the general SLB problem. The study of this minimization model is quite relevant since it allows to formalize real-life applications in different areas, such as imaging, signal processing, machine learning and portfolio optimization (see for example [7, 8, 57, 67, 83]).

The chapter is organized as follows. In the first section we briefly recall the optimality conditions in the quadratic case; in Section 3.2 we introduce the main properties of the Hessian matrix restricted to the tangent space of the active constraints at the solution, which is involved in the spectral analysis of the BB rules of Section 3.3. Finally, the analysis is extended to the non-quadratic case and variable metric GP schemes.

3.1 OPTIMALITY CONDITIONS

We start our analysis from the simpler case of a SLB quadratic programming problem (SLBQP) of the form:

$$\min_{x\in\Omega} f(x) \equiv \frac{1}{2}x^T A x - b^T x + c, \tag{3.3}$$

where $A \in \mathcal{M}_n(\mathbb{R})$ is a SPD matrix, $b \in \mathbb{R}^n$, $c \in \mathbb{R}$, and the feasible region Ω is defined as in (3.2). Problem (3.3) admits a unique

solution x^*, satisfying the KKT optimality conditions, i. e. there exist $\psi^* \in \mathbb{R}$ and $\mu^*, \nu^* \in \mathbb{R}^n$ such that

$$\begin{cases} g(x^*) - \psi^* \nu - \mu^* + \nu^* = 0, \\ \nu^T x^* = e, \\ \mu^* \cdot (x^* - \ell) = 0, \qquad x^* - \ell \geqslant 0, \qquad \mu^* \geqslant 0, \\ \nu^* \cdot (u - x^*) = 0, \qquad u - x^* \geqslant 0, \qquad \nu^* \geqslant 0, \end{cases} \tag{3.4}$$

where the products between vectors have to be intended component–wise. We denote $\mathcal{N} = \{1, 2 \ldots, n\}$, $\mathcal{J}^* = \{i \in \mathcal{N} \colon x_i^* = \ell_i,\ x_i^* = u_i\}$, and $\mathcal{I}^* = \mathcal{N} \setminus \mathcal{J}^*$ with cardinality $m = \sharp \mathcal{I}^*$. Apart from trivial cases, we assume that $0 < m < n$, i. e. $\varnothing \subset \mathcal{I}^* \subset \mathcal{N}$, and that ν and the columns of the identity matrix of order n with indices in $\mathcal{J}^*$ are linearly independent. Consequently, the entries of ν corresponding to $\mathcal{I}^*$ are not all equal to zero. For the sake of simplicity, we assume that the rows and columns of A and the entries of any vector are permuted so that $\mathcal{I}^*$ is related to the first m indices and $\mathcal{J}^*$ contains the last $n - m$ indices.
As for the box-constrained case, we will show that the reciprocals of the steplengths defined by the BB rules are related to the spectrum of special matrices obtained by restricting the matrix A to the subspaces depending on the constraints that become active during the iterative process. To this end, we first introduce the definition of Hessian matrix restricted to the tangent space of the active constraints at the solution and then we suggest a possible way to approximate this matrix during the iterations of gradient projection schemes.

3.2 THE HESSIAN MATRIX RESTRICTED TO THE TANGENT SPACE OF THE ACTIVE CONSTRAINTS AT THE SOLUTION

Let S^* be the surface defined by the active constraints at the solution. The tangent space $\mathcal{T}^*(x^*)$ to the surface S^* at x^* is defined as

$$\begin{aligned} \mathcal{T}^*(x^*) &= \left\{ x \in \mathbb{R}^n \ : \ \begin{bmatrix} \mathcal{O}_{n-m,m} & I_{\mathcal{J}^*} \\ \nu^T & \end{bmatrix} x = \mathcal{O}_{n-m+1,1} \right\} \\ &= \operatorname{range}\left(\begin{bmatrix} I_{\mathcal{J}^*} \\ \mathcal{O}_{n-m,m} \end{bmatrix} \right) \cap \ker(\nu^T). \end{aligned} \tag{3.5}$$

Let $\mathcal{T}^* \equiv \mathcal{T}^*(x^*)$. Taking into account that the dimension of the surface of the active constraints is $(n - m + 1)$, we have that the dimension of $\mathcal{T}^*$ is $(m - 1)$. We can consider the matrix $A^* \in \mathcal{M}_{m-1}(\mathbb{R})$ given by

$$A^* = \tilde{U}^{*T} A \tilde{U}^* \tag{3.6}$$

where $\tilde{U}^*$ denotes an $n \times (m - 1)$ matrix whose columns are an orthonormal basis of $\mathcal{T}^*$. We call A^* the Hessian matrix restricted to

the tangent space of the active constraints at the solution, hereafter named *restricted Hessian matrix*. Let observe that this matrix is the obvious generalization to the SLB case of the reduced Hessian matrix arisen in the box-constrained framework.

In order to characterize the SPD matrix A^*, some useful results on projection matrices are collected in the following lemmas (see Appendix for the proofs).

Let $\mathcal{V}$ and $\mathcal{W}$ be two subspaces of $\mathbb{R}^n$ such that $\mathbb{R}^n = \mathcal{V} \oplus \mathcal{W}$.

Lemma 3.1 *A matrix $P \in \mathcal{M}_n(\mathbb{R})$ is a projection matrix onto $\mathcal{V} = \text{range}(P)$ along $\mathcal{W} = \ker(P)$ if and only if*

$$P^2 = P. \tag{3.7}$$

Let P be a projection matrix onto $\mathcal{V}$ along $\mathcal{W}$ and let $x = v + w$, $v \in \mathcal{V}$, $w \in \mathcal{W}$. The matrix $Q = I_n - P$ satisfies $Qx = (I_n - P)x = w$ and

$$Q^2 = (I_n - P)^2 = I_n - 2P + P^2 = I_n - P = Q,$$

then Q is a projection matrix onto $\mathcal{W}$ along $\mathcal{V}$. We also have

$$PQ = P(I_n - P) = P - P^2 = \mathcal{O}_n,$$

implying $\text{range}(Q) \subseteq \ker(P)$; from Lemma 3.1, $\text{range}(Q) = \ker(P)$. Similarly $QP = \mathcal{O}_n$, then $\text{range}(P) = \ker(Q)$.

Lemma 3.2 *Let $\mathcal{V}$ be subspace of $\mathbb{R}^n$ and let $\mathcal{V}^\perp$ denote the complement subspace of $\mathcal{V}$. A matrix $P \in \mathcal{M}_n(\mathbb{R})$ is an orthogonal projection matrix onto $\mathcal{V}$ along $\mathcal{V}^\perp$ if and only if*

$$P^2 = P \quad \textit{and} \quad P^T = P. \tag{3.8}$$

Furthermore, the eigenvalues of P are 0 or 1.

The following result descend from the previous lemmas and general results on spectral decomposition of symmetric matrices.

Lemma 3.3 *Let $u \in \mathbb{R}^m$ be a non-zero vector and let $\mathcal{V}$ be the subspace of $\mathbb{R}^m$ spanned by the vector u: $\mathcal{V} = \text{range}(u)$. Then, the matrix $V = \frac{uu^T}{u^Tu}$ is the orthogonal projection onto $\mathcal{V}$. Moreover, it holds that*

(a) the spectral decomposition of V is

$$V = W \begin{bmatrix} \mathcal{O}_{m-1,m-1} & \mathcal{O}_{m-1,1} \\ \mathcal{O}_{1,m-1} & 1 \end{bmatrix} W^T,$$

where $W = \begin{bmatrix} \tilde{W} & w \end{bmatrix}$ is an orthogonal matrix of order m such that $\tilde{W} \in \mathcal{M}_{m,m-1}(\mathbb{R})$, $w \in \mathbb{R}^m$; the subspace $\text{range}(V) = \text{range}(w)$ is the one-dimensional eigenspace associated to the eigenvalue 1, whereas $\ker(V) = \text{range}(\tilde{W})$ is the eigenspace of dimension $m - 1$ associated to the eigenvalue 0;

(b) Let $P = I_m - V$. *The spectral decomposition of* P *is*

$$P = W \begin{bmatrix} I_{m-1} & \mathcal{O}_{m-1,1} \\ \mathcal{O}_{1,m-1} & 0 \end{bmatrix} W^T;$$

the subspace $\text{range}(P) = \text{range}(\tilde{W})$ *is the eigenspace of dimension* $m-1$ *associated to the eigenvalue* 1 *and* $\ker(P) = \text{range}(w)$ *is the one-dimensional eigenspace associated to the eigenvalue* 0*;*

(c) $P = \tilde{W}\tilde{W}^T$ *and* $V = ww^T$, *with* $\tilde{W}^T\tilde{W} = I_{m-1}$ *and* $w^Tw = 1$, $w = \frac{u}{\|u\|}$; *furthermore* $\tilde{W}$ *is an orthonormal basis of* $\text{range}(P) = \mathcal{V}^\perp = \ker(u^T)$.

By using the notation $v^T = \begin{bmatrix} v_{\mathcal{J}*}^T & v_{\mathcal{G}*}^T \end{bmatrix}$ and applying Lemma 3.3 with $u = v_{\mathcal{J}*}$ it is possible to construct the matrix $\tilde{U}^*$ involved in (3.6). Indeed, if P^* denotes the orthogonal projection onto $\ker(v_{\mathcal{J}*}^T)$, from part (c) of Lemma 3.3, there exists a matrix $\tilde{W}^* \in \mathcal{M}_{m,m-1}(\mathbb{R})$ whose columns are an orthonormal basis of $\ker(v_{\mathcal{J}*}^T)$. Consequently, the matrix $\tilde{U}^* = \begin{bmatrix} \tilde{W}^* \\ \mathcal{O}_{n-m,m-1} \end{bmatrix}$ provides an orthonormal basis for $\mathcal{T}^*$, since any vector $x \in \mathcal{T}^*$ can be expressed as $x^T = \begin{bmatrix} x_{\mathcal{J}*}^T & x_{\mathcal{G}*}^T \end{bmatrix}$ with $x_{\mathcal{J}*} \in \ker(v_{\mathcal{J}*}^T)$ and $x_{\mathcal{G}*} = \mathcal{O}_{n-m,1}$.

The spectrum of A^* plays a crucial role in the analysis of the steplength rules, as we will see in the next section.

3.3 SPECTRAL ANALYSIS OF THE BB RULES RELATED TO APPROXIMATING RESTRICTED HESSIAN MATRICES

Since both the matrix A (and consequently A^*) and the solution x^* of problem (3.3) are not generally available, we need to provide a way to realize a sequence of suitable approximations of A^* during the iterations, in order to highlight the relationship between the spectra of these approximating matrices and the BB rules.

Let consider the set of indices introduced in (2.8), and assume that the rows/columns of A and the entries of any vector are reordered so that $\mathcal{I}_{k-1}$ is related to the first $m_k = \sharp\mathcal{I}_{k-1}$ indices and $\mathcal{J}_{k-1}$ contains the last $n - m_k$ indices.
Let P_{k-1} be the orthogonal projection onto $\ker(v_{\mathcal{J}_{k-1}}^T)$:

$$P_{k-1} = I_{\mathcal{I}_{k-1}} - \frac{1}{v_{\mathcal{J}_{k-1}}^T v_{\mathcal{I}_{k-1}}} v_{\mathcal{I}_{k-1}} v_{\mathcal{J}_{k-1}}^T. \tag{3.9}$$

From part (c) of Lemma 3.3, there exists a matrix $\tilde{W}_{k-1} \in \mathcal{M}_{m_k,m_k-1}(\mathbb{R})$ such that

$$P_{k-1} = \tilde{W}_{k-1}\tilde{W}_{k-1}^T \tag{3.10}$$

and the $n \times (m_k - 1)$ matrix

$$\tilde{U}_{k-1} = \begin{bmatrix} \tilde{W}_{k-1} \\ \mathcal{O}_{n-m_k, m_k-1} \end{bmatrix} \tag{3.11}$$

is an orthonormal basis for the subspace

$$\mathcal{T}_{k-1} = \left\{ x \in \mathbb{R}^n \ : \ \begin{bmatrix} \mathcal{O}_{n-m_k, m_k} & I_{\mathcal{J}_{k-1}} \\ & v^T \end{bmatrix} x = \mathcal{O}_{n-m_k+1,1} \right\}. \tag{3.12}$$

Therefore, the SPD matrix $\tilde{U}_{k-1}^T A \tilde{U}_{k-1}$ represents an approximation of the matrix (3.6) at the iteration k.
Recalling that $y^{(k-1)} = g^{(k)} - g^{(k-1)}$, we introduce the vector:

$$t^{(k-1)} = g^{(k)} - \psi_k v - (g^{(k-1)} - \psi_{k-1} v) = y^{(k-1)} - (\psi_k - \psi_{k-1}) v, \tag{3.13}$$

where the quantities ψ_{k-1} and ψ_k are defined as

$$\psi_{k-1} = \frac{v_{\mathcal{J}_{k-1}}^T g_{\mathcal{J}_{k-1}}^{(k-1)}}{v_{\mathcal{J}_{k-1}}^T v_{\mathcal{J}_{k-1}}}, \qquad \psi_k = \frac{v_{\mathcal{J}_{k-1}}^T g_{\mathcal{J}_{k-1}}^{(k)}}{v_{\mathcal{J}_{k-1}}^T v_{\mathcal{J}_{k-1}}}. \tag{3.14}$$

The values (3.14) correspond to approximations computed, respectively, at iterations $k-1$ and k of the Langrange multiplier ψ^* associated to the equality constraint (see KKT conditions (3.4)). We notice that quantities similar to $t^{(k-1)}$ were also considered in the framework of interior point methods [54].

The vector (3.13) can be partitioned as $t^{(k-1)} = \begin{bmatrix} t_{\mathcal{J}_{k-1}}^{(k-1)} \\ t_{\mathcal{J}_{k-1}}^{(k-1)} \end{bmatrix}$, where

$$t_{\mathcal{J}_{k-1}}^{(k-1)} = P_{k-1} y_{\mathcal{J}_{k-1}}^{(k-1)} = \tilde{W}_{k-1} \tilde{W}_{k-1}^T y_{\mathcal{J}_{k-1}}^{(k-1)}. \tag{3.15}$$

To better understand the role of $t_{\mathcal{J}_{k-1}}^{(k-1)}$ within the definitions of the BB steplength rules, we need to prove the following lemma.

Lemma 3.4 *Given $s^{(k-1)} = x^{(k)} - x^{(k-1)}$ and $t^{(k-1)}$ as in (3.13), it holds that*

(a) $s^{(k-1)^T} v = 0$;

(b) $s_{\mathcal{J}_{k-1}}^{(k-1)^T} v_{\mathcal{J}_{k-1}} = 0$;

(c) $s^{(k-1)^T} y^{(k-1)} = s^{(k-1)^T} t^{(k-1)} = s_{\mathcal{J}_{k-1}}^{(k-1)^T} t_{\mathcal{J}_{k-1}}^{(k-1)}$.

Proof: (a). $s^{(k-1)^T} v = 0$ since both $x^{(k-1)}$ and $x^{(k)}$ satisfy the equality constraint $v^T x = e$.
(b). We show that $s_{\mathcal{J}_{k-1}}^{(k-1)^T} v_{\mathcal{J}_{k-1}} = 0$. Indeed,

$$0 = s^{(k-1)^T} v = \sum_{i \in \mathcal{J}_{k-1}} s_i^{(k-1)^T} v_i + \sum_{i \in \mathcal{J}_{k-1}} s_i^{(k-1)^T} v_i = \sum_{i \in \mathcal{J}_{k-1}} s_i^{(k-1)^T} v_i,$$

where the last equality holds since $s^{(k-1)}_{\mathcal{J}_{k-1}} = \mathcal{O}_{n-m_k,1}$ from the definition of $\mathcal{J}_{k-1}$.
(c). From parts (a), we have that

$$s^{(k-1)^T} t^{(k-1)} = s^{(k-1)^T} \left(y^{(k-1)} - (\psi_k - \psi_{k-1})v \right) = s^{(k-1)^T} y^{(k-1)}, \tag{3.16}$$

and, since $s^{(k-1)}_{\mathcal{J}_{k-1}} = \mathcal{O}_{n-m_k,1}$, the last equality follows easily. □
Lemma 3.4 allows to state that the classical formulation (1.17) of the first BB rule provides a steplength depending only on the indices belonging to the subset $\mathcal{I}_{k-1}$:

$$\alpha_k^{\text{BB1}} = \frac{s^{(k-1)^T} s^{(k-1)}}{s^{(k-1)^T} y^{(k-1)}} = \frac{s^{(k-1)^T}_{\mathcal{J}_{k-1}} s^{(k-1)}_{\mathcal{J}_{k-1}}}{s^{(k-1)^T}_{\mathcal{J}_{k-1}} t^{(k-1)}_{\mathcal{J}_{k-1}}}. \tag{3.17}$$

As in the box-constrained case, the rule α_k^{BB1} computes the steplength by capturing, in a natural way, the information related to the current inactive constraints, discarding the effect of those constraints that remain active in the last two iterations. The original BB2 rule does not fulfill a similar property, due to the special form of its denominator. By analogy with the modified rule (2.21), we suggest the following rule

$$\alpha_k^{\text{EQ-BB2}} = \frac{s^{(k-1)^T} y^{(k-1)}}{t^{(k-1)^T}_{\mathcal{J}_{k-1}} t^{(k-1)}_{\mathcal{J}_{k-1}}}. \tag{3.18}$$

It is worth noting that the BB rules (2.11)-(2.21), obtained in the box-constrained framework, are special instances of (3.17) and (3.18), in the case where the SLBQP problem (3.3) reduces to (2.1) . Therefore, the following theorem extends the result of Theorem 2.2 to the case of SLBQP problems, showing that the reciprocals of α_k^{BB1} and $\alpha_k^{\text{EQ-BB2}}$ give spectral information related to the approximating restricted matrix $\tilde{U}_{k-1}^T A \tilde{U}_{k-1}$, at each iteration.

Theorem 3.1 *Let $\{x^{(k)}\}_{k\in\mathbb{N}}$ be the sequence generated by the GP method (1.50) for solving problem (3.3), where A is a SPD matrix. Let $\mathcal{I}_{k-1}$ be the subset of indices defined in (2.8) and then we have*

$$\lambda_{\min}(\tilde{U}_{k-1}^T A \tilde{U}_{k-1}) \leqslant \frac{1}{\alpha_k^{BB1}} \leqslant \frac{1}{\alpha_k^{EQ\text{-}BB2}} \leqslant \lambda_{\max}(\tilde{U}_{k-1}^T A \tilde{U}_{k-1}) \tag{3.19}$$

where $\lambda_{\min}(\tilde{U}_{k-1}^T A \tilde{U}_{k-1})$ and $\lambda_{\max}(\tilde{U}_{k-1}^T A \tilde{U}_{k-1})$ are the minimum and the maximum eigenvalues of $\tilde{U}_{k-1}^T A \tilde{U}_{k-1})$, respectively, and $\tilde{U}_{k-1}^T$ is defined as in (3.11).

Proof: In the following, we drop for simplicity the iteration counter $k-1$ from $\mathcal{I}_{k-1}$ and $\mathcal{J}_{k-1}$. In view of the gradient projection iteration (1.50), we have that the entries of the iterate $x^{(k)}$ are

$$x_i^{(k)} = \begin{cases} x_i^{(k-1)} + \nu_{k-1}(r_i^{(k-1)} - x_i^{(k-1)}) & \text{for } i \in \mathcal{I}, \\ x_i^{(k-1)} & \text{for } i \in \mathcal{J}, \end{cases} \tag{3.20}$$

where $r_i^{(k-1)} = (\Pi_\Omega(x^{(k-1)} - \alpha_{k-1} g^{(k-1)}))_i, i \in \mathcal{I}$. The vector $s^{(k-1)}$ can be partitioned as follows

$$s^{(k-1)} = \begin{pmatrix} s_{\mathcal{I}}^{(k-1)} \\ s_{\mathcal{J}}^{(k-1)} \end{pmatrix} = \begin{pmatrix} \nu_{k-1}(r^{(k-1)} - x_{\mathcal{I}}^{(k-1)}) \\ 0_{\mathcal{J},1} \end{pmatrix}. \tag{3.21}$$

Any entry $g_i^{(k)}, i = 1, \ldots, n$, of the gradient $g^{(k)}$ has the following expression:

$$\begin{aligned} g_i^{(k)} &= \sum_{j=1}^{n} a_{ij} x_j^{(k)} - b_i \\ &= \sum_{j \in \mathcal{I}} a_{ij}(x_j^{(k-1)} + \nu_{k-1}(r_j^{(k-1)} - x_j^{(k-1)})) + \sum_{j \in \mathcal{J}} a_{ij} x_j^{(k-1)} - b_i \\ &= g_i^{(k-1)} + \nu_{k-1} \sum_{j \in \mathcal{I}} a_{ij}(r_j^{(k-1)} - x_j^{(k-1)}). \end{aligned}$$

Consequently, from (3.21), we can write

$$y^{(k-1)} = \begin{pmatrix} y_{\mathcal{I}}^{(k-1)} \\ y_{\mathcal{J}}^{(k-1)} \end{pmatrix} = \begin{pmatrix} A_{\mathcal{I},\mathcal{I}} s_{\mathcal{I}}^{(k-1)} \\ A_{\mathcal{J},\mathcal{I}} s_{\mathcal{I}}^{(k-1)} \end{pmatrix}. \tag{3.22}$$

Furthermore, from $P_{k-1} = P_{k-1}^2$ it follows that

$$P_{k-1} s_{\mathcal{I}}^{(k-1)} = \left(I_{\mathcal{I}} - \frac{1}{v_{\mathcal{I}}^T v_{\mathcal{I}}} v_{\mathcal{I}} v_{\mathcal{I}}^T \right) s_{\mathcal{I}}^{(k-1)} = s_{\mathcal{I}}^{(k-1)}. \tag{3.23}$$

Hence, from (3.23) and (3.10) we observe that

$$s^{(k-1)^T} s^{(k-1)} = s_{\mathcal{I}}^{(k-1)^T} s_{\mathcal{I}}^{(k-1)} = s_{\mathcal{I}}^{(k-1)^T} P_{k-1} s_{\mathcal{I}}^{(k-1)} = \|\tilde{W}_{k-1}^T s_{\mathcal{I}}^{(k-1)}\|^2. \tag{3.24}$$

From (3.15), (3.22), (3.23), (3.10) and (3.11), we obtain

$$\begin{aligned} s^{(k-1)^T} y^{(k-1)} &= s_{\mathcal{I}}^{(k-1)^T} t_{\mathcal{I}}^{(k-1)} \\ &= s_{\mathcal{I}}^{(k-1)^T} P_{k-1} y_{\mathcal{I}}^{(k-1)} \\ &= s_{\mathcal{I}}^{(k-1)^T} P_{k-1} A_{\mathcal{I},\mathcal{I}} s_{\mathcal{I}}^{(k-1)} \\ &= s_{\mathcal{I}}^{(k-1)^T} P_{k-1} A_{\mathcal{I},\mathcal{I}} P_{k-1} s_{\mathcal{I}}^{(k-1)} \\ &= s_{\mathcal{I}}^{(k-1)^T} \tilde{W}_{k-1} \tilde{W}_{k-1}^T A_{\mathcal{I},\mathcal{I}} \tilde{W}_{k-1} \tilde{W}_{k-1}^T s_{\mathcal{I}}^{(k-1)} \\ &= s_{\mathcal{I}}^{(k-1)^T} \tilde{W}_{k-1} \tilde{U}_{k-1}^T A \tilde{U}_{k-1} \tilde{W}_{k-1}^T s_{\mathcal{I}}^{(k-1)}. \end{aligned} \tag{3.25}$$

From (3.24) and (3.25), we can conclude that $1/\alpha_k^{BB1}$ is the Rayleigh quotient of the matrix $\tilde{U}_{k-1}^T A \tilde{U}_{k-1}$ at the vector $\tilde{W}_{k-1}^T s_{\mathcal{I}}^{(k-1)}$; then, from the extremal properties of the Rayleigh quotient, we have

$$\lambda_{min}(\tilde{U}_{k-1}^T A \tilde{U}_{k-1}) \leqslant \frac{1}{\alpha_k^{BB1}} \leqslant \lambda_{max}(\tilde{U}_{k-1}^T A \tilde{U}_{k-1}).$$

Similar arguments to that of (3.25), enable to write

$$t_{\mathcal{J}}^{(k-1)} = \tilde{W}_{k-1}\tilde{U}_{k-1}^T A\tilde{U}_{k-1}\tilde{W}_{k-1}^T s_{\mathcal{J}}^{(k-1)}. \tag{3.26}$$

As a consequence, recalling that $\tilde{W}_{k-1}^T\tilde{W}_{k-1} = I_{m-1}$, we obtain

$$t_{\mathcal{J}}^{(k-1)^T} t_{\mathcal{J}}^{(k-1)} = s_{\mathcal{J}}^{(k-1)^T}\tilde{W}_{k-1}(\tilde{U}_{k-1}^T A\tilde{U}_{k-1})^2\tilde{W}_{k-1}^T s_{\mathcal{J}}^{(k-1)}. \tag{3.27}$$

Since $\tilde{U}_{k-1}^T A\tilde{U}_{k-1}$ is a SPD matrix, we can introduce the vector

$$z^{(k-1)} = (\tilde{U}_{k-1}^T A\tilde{U}_{k-1})^{1/2}\tilde{W}_{k-1}^T s_{\mathcal{J}}^{(k-1)}, \tag{3.28}$$

so that the scalar product in (3.27) can be written as

$$t_{\mathcal{J}}^{(k-1)^T} t_{\mathcal{J}}^{(k-1)} = z^{(k-1)^T}\tilde{U}_{k-1}^T A\tilde{U}_{k-1} z^{(k-1)}, \tag{3.29}$$

and

$$s^{(k-1)^T} y^{(k-1)} = z^{(k-1)^T} z^{(k-1)}.$$

Hence, $1/\alpha_k^{\text{EQ-BB2}}$ is the Rayleigh quotient of the matrix $\tilde{U}_{k-1}^T A\tilde{U}_{k-1}$ at the vector $z^{(k-1)}$ and the following inequality holds

$$\lambda_{min}(\tilde{U}_{k-1}^T A\tilde{U}_{k-1}) \leqslant \frac{1}{\alpha_k^{\text{EQ-BB2}}} \leqslant \lambda_{max}(\tilde{U}_{k-1}^T A\tilde{U}_{k-1}). \tag{3.30}$$

Finally, from the Cauchy-Schwarz inequality, it follows that

$$\begin{aligned}
\frac{1}{\alpha_k^{\text{BB1}}} &= \frac{s^{(k-1)^T} y^{(k-1)}}{s^{(k-1)^T} s^{(k-1)}} = \frac{s_{\mathcal{J}_{k-1}}^{(k-1)^T} t_{\mathcal{J}_{k-1}}^{(k-1)}}{s_{\mathcal{J}_{k-1}}^{(k-1)^T} s_{\mathcal{J}_{k-1}}^{(k-1)}} \\
&\leqslant \frac{\|s_{\mathcal{J}_{k-1}}^{(k-1)}\|\|t_{\mathcal{J}_{k-1}}^{(k-1)}\|}{\|s_{\mathcal{J}_{k-1}}^{(k-1)}\|^2} = \frac{\|t_{\mathcal{J}_{k-1}}^{(k-1)}\|\|t_{\mathcal{J}_{k-1}}^{(k-1)}\|}{\|s_{\mathcal{J}_{k-1}}^{(k-1)}\|\|t_{\mathcal{J}_{k-1}}^{(k-1)}\|} \\
&\leqslant \frac{t_{\mathcal{J}_{k-1}}^{(k-1)^T} t_{\mathcal{J}_{k-1}}^{(k-1)}}{s_{\mathcal{J}_{k-1}}^{(k-1)^T} t_{\mathcal{J}_{k-1}}^{(k-1)}} = \frac{t_{\mathcal{J}_{k-1}}^{(k-1)^T} t_{\mathcal{J}_{k-1}}^{(k-1)}}{s^{(k-1)^T} y^{(k-1)}} = \frac{1}{\alpha_k^{\text{EQ-BB2}}}.
\end{aligned} \tag{3.31}$$

□

REMARK The spectral properties described in Theorem 3.1 are useful to highlight the ability of the BB steplength updating rule to foster a reduction of the quantities $|g_i^{(k-1)} - \psi_{k-1}v_i|$, $i \in \mathcal{J}_{k-1}$, which is a remarkable skill since at the solution x^* we have $g_{\mathcal{J}*}(x^*) - \psi^* v_{\mathcal{J}*} = 0$, as prescribed by the KKT conditions. The next theorem is crucial to prove the mentioned ability.

Theorem 3.2 *Let assume that* $\mathcal{I}_{k-1} = \mathcal{I}_k$ *and*

$$\ell_i < \left(\Pi_\Omega(x^{(k-1)} - \alpha_{k-1} g^{(k-1)})\right)_i < u_i, \quad i \in \mathcal{I}_{k-1}.$$

The following equality holds:

$$P_k g^{(k)}_{\mathcal{I}_k} = \left(I_{m_k} - \alpha_{k-1} v_{k-1} P_{k-1} A_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}} P_{k-1}\right) P_{k-1} g^{(k-1)}_{\mathcal{I}_{k-1}}. \quad (3.32)$$

Proof: By using equations (3.21)-(3.23) and the assumption $\mathcal{I}_{k-1} = \mathcal{I}_k$, we can write

$$\begin{aligned} P_k g^{(k)}_{\mathcal{I}_k} &= P_{k-1}\left(g^{(k-1)}_{\mathcal{I}_{k-1}} + A_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}} s^{(k-1)}_{\mathcal{I}_{k-1}}\right) \\ &= P_{k-1} g^{(k-1)}_{\mathcal{I}_{k-1}} + v_{k-1} P_{k-1} A_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}} P_{k-1}\left(r^{(k-1)} - x^{(k-1)}_{\mathcal{I}_{k-1}}\right), \end{aligned} \quad (3.33)$$

where $r^{(k-1)} = \left(\Pi_\Omega(x^{(k-1)} - \alpha_{k-1} g^{(k-1)})\right)_{\mathcal{I}_{k-1}}$.
Since $x^{(k)}_{\mathcal{I}_{k-1}} = x^{(k-1)}_{\mathcal{I}_{k-1}}$ and $\ell_{\mathcal{I}_{k-1}} < r^{(k-1)} < u_{\mathcal{I}_{k-1}}$, the vector $\Pi_\Omega(x^{(k-1)} - \alpha_{k-1} g^{(k-1)})$ can be partitioned as follows

$$\Pi_\Omega(x^{(k-1)} - \alpha_{k-1} g^{(k-1)}) = \begin{pmatrix} r^{(k-1)} \\ x^{(k-1)}_{\mathcal{I}_{k-1}} \end{pmatrix}.$$

Let denote $\tilde{e} = e - v^T_{\mathcal{I}_{k-1}} x^{(k-1)}_{\mathcal{I}_{k-1}}$, the vector $r^{(k-1)}$ solves the problem

$$r^{(k-1)} = \underset{\left\{r \,:\, v^T_{\mathcal{I}_{k-1}} r = \tilde{e}\right\}}{\arg\min} \frac{1}{2}\|r - (x^{(k-1)} - \alpha_{k-1} g^{(k-1)})_{\mathcal{I}_{k-1}}\|^2. \quad (3.34)$$

From the KKT conditions related to problem (3.34), the vector $r^{(k-1)}$ has the following expression

$$\begin{aligned} r^{(k-1)} = {} & x^{(k-1)}_{\mathcal{I}_{k-1}} - \alpha_{k-1} g^{(k-1)}_{\mathcal{I}_{k-1}} + \\ & - \frac{v^T_{\mathcal{I}_{k-1}}(x^{(k-1)}_{\mathcal{I}_{k-1}} - \alpha_{k-1} g^{(k-1)}_{\mathcal{I}_{k-1}})}{v^T_{\mathcal{I}_{k-1}} v_{\mathcal{I}_{k-1}}} v_{\mathcal{I}_{k-1}} + \frac{\tilde{e} v_{\mathcal{I}_{k-1}}}{v^T_{\mathcal{I}_{k-1}} v_{\mathcal{I}_{k-1}}} = \\ = {} & x^{(k-1)}_{\mathcal{I}_{k-1}} - \alpha_{k-1} g^{(k-1)}_{\mathcal{I}_{k-1}} - \frac{v^T_{\mathcal{I}_{k-1}} x^{(k-1)}_{\mathcal{I}_{k-1}} v_{\mathcal{I}_{k-1}}}{v^T_{\mathcal{I}_{k-1}} v_{\mathcal{I}_{k-1}}} + \\ & + \frac{\alpha_{k-1} v^T_{\mathcal{I}_{k-1}} g^{(k-1)}_{\mathcal{I}_{k-1}} v_{\mathcal{I}_{k-1}}}{v^T_{\mathcal{I}_{k-1}} v_{\mathcal{I}_{k-1}}} + \frac{\tilde{e} v_{\mathcal{I}_{k-1}}}{v^T_{\mathcal{I}_{k-1}} v_{\mathcal{I}_{k-1}}} = \\ = {} & x^{(k-1)}_{\mathcal{I}_{k-1}} - \alpha_{k-1} P_{k-1} g^{(k-1)}_{\mathcal{I}_{k-1}}. \end{aligned} \quad (3.35)$$

From equations (3.33) and (3.35), the thesis is proved. □

Assuming $\mathcal{I}_k = \mathcal{I}_{k-1}$, as immediate corollary of Theorem 3.2 we have

$$\tilde{W}_k \tilde{W}^T_k g^{(k)}_{\mathcal{I}_k} = \tilde{W}_{k-1}\left(I_{m_{k-1}} - \alpha_{k-1} v_{k-1} \tilde{U}^T_{k-1} A \tilde{U}_{k-1}\right) \tilde{W}^T_{k-1} g^{(k-1)}_{\mathcal{I}_{k-1}}, \quad (3.36)$$

where (3.10) is used; in this case $\tilde{W}_{k-1} = \tilde{W}_k$ and, due to the linear independence of their columns, from (3.36) it follows that

$$\tilde{W}_k^T g_{\mathcal{I}_k}^{(k)} = \left(I_{m_k-1} - \alpha_{k-1}\nu_{k-1}\tilde{U}_{k-1}^T A\tilde{U}_{k-1}\right)\tilde{W}_{k-1}^T g_{\mathcal{I}_{k-1}}^{(k-1)}. \quad (3.37)$$

If we denote by $(\lambda_1,\ldots,\lambda_{m_k-1})$ and $(\xi_1,\ldots,\xi_{m_k-1})$ the eigenvalues and the associated orthonormal eigenvectors of $\tilde{U}_{k-1}^T A\tilde{U}_{k-1}$, we may write $\tilde{W}_k^T g_{\mathcal{I}_k}^{(k)} = \sum_{i=1}^{m_k-1}\gamma_i^{(k)}\xi_i$ and $\tilde{W}_{k-1}^T g_{\mathcal{I}_{k-1}}^{(k-1)} = \sum_{i=1}^{m_k-1}\gamma_i^{(k-1)}\xi_i$. For the eigencomponents $\gamma_i^{(k)}$ the following recurrence formula can be easily derived from (3.37):

$$\gamma_i^{(k)} = (1-\alpha_{k-1}\nu_{k-1}\lambda_i)\,\gamma_i^{(k-1)}, \qquad i = 1,\ldots,m_k-1. \quad (3.38)$$

Formula (3.38) highlights that, if α_{k-1} is an accurate approximation of the inverse of an eigenvalue of $\tilde{U}_{k-1}^T A\tilde{U}_{k-1}$, since $\nu_k \in (0,1]$, a reduction of $\left|\gamma_i^{(k)}\right|$ with respect to $\left|\gamma_i^{(k-1)}\right|$ is obtained.

Therefore, a reduction on the eigencomponents of $\tilde{W}_k^T g_{\mathcal{I}_k}^{(k)}$ results in a reduction on the quantities $\|g_{\mathcal{I}_k}^{(k)} - \psi_k\nu_{\mathcal{I}_k}\|$, as the set $\mathcal{I}_k$ tends to stabilize; indeed, when $\mathcal{I}_{k-1} = \mathcal{I}_k$, we may observe that

$$g_{\mathcal{I}_k}^{(k)} - \psi_k\nu_{\mathcal{I}_k} = g_{\mathcal{I}_k}^{(k)} - \frac{\nu_{\mathcal{I}_k}^T g_{\mathcal{I}_k}^{(k)}}{\nu_{\mathcal{I}_k}^T\nu_{\mathcal{I}_k}}\nu_{\mathcal{I}_k} = P_k g_{\mathcal{I}_k}^{(k)},$$

$$g_{\mathcal{I}_{k-1}}^{(k-1)} - \psi_{k-1}\nu_{\mathcal{I}_{k-1}} = g_{\mathcal{I}_{k-1}}^{(k-1)} - \frac{\nu_{\mathcal{I}_{k-1}}^T g_{\mathcal{I}_{k-1}}^{(k-1)}}{\nu_{\mathcal{I}_{k-1}}^T\nu_{\mathcal{I}_{k-1}}}\nu_{\mathcal{I}_{k-1}} = P_{k-1} g_{\mathcal{I}_{k-1}}^{(k-1)}.$$

By recalling that $P^* g_{\mathcal{I}^*}(x^*) = \tilde{W}^*\tilde{W}^{*T} g_{\mathcal{I}^*}(x^*) = 0$, we can conclude that the use of a steplength rule providing good approximations of the inverse of the eigenvalues of $\tilde{U}_{k-1}^T A\tilde{U}_{k-1}$ can be a fruitful strategy for accelerating gradient projection methods for problem (3.3).

We provide a first example of the spectral behaviour of the GP method combined with the steplength updating rules BB1, BB2 and EQ-BB2, on a SLBQP toy problem of size $n = 20$, with ten active constraints at the solution and such that the eigenvalues of the Hessian matrix are logarithmically distributed in the interval $[1,500]$; the spectral condition number is equal to 500.
For the GP implementation we refer to Algorithm 2, with the following parameters setting: $\alpha_{min} = 10^{-10}$, $\alpha_{max} = 10^6$, $\alpha_0 = \dfrac{g^{(0)T}g^{(0)}}{g^{(0)T}Ag^{(0)}}$.
We remind that the projection onto the feasible set Ω can be formulated as a root-finding problem and have been effectively computed by means of the secant-like algorithm developed in [28]. The feasible

initial point $x^{(0)}$ is randomly generated and then projected into the feasible region. The stopping criterion is defined as:

$$\|x^{(k)} - x^{(k-1)}\|_\infty \leqslant \mathtt{tol}, \tag{3.39}$$

with $\mathtt{tol} = 10^{-7}$. Figure 7 shows the distribution of the sequences $\left\{\frac{1}{\alpha_k}\right\}$ generated by each rule with respect to the eigenvalues of the matrices $\tilde{U}_{k-1}^T A \tilde{U}_{k-1}$, during the iterative procedure. In each panel, at the k-th iteration, the black circles denote the eigenvalues of the restricted Hessian matrix $\tilde{U}_{k-1}^T A \tilde{U}_{k-1}$, whereas the reciprocals of the steplengths α_k are plotted by red crosses; the blue lines correspond to maximum and minimum eigenvalues of the Hessian matrix A and the blue circles on the right denote the eigenvalues of the restricted Hessian matrix at the prefixed solution x^*. These plots confirm the ability of the reciprocals of BB1 and EQ-BB2 steplengths to sweep the spectrum of the restricted Hessian matrices, while the reciprocals of the steplengths produced by the original BB2 scheme are placed near the largest eigenvalues of the restricted Hessian matrices during the iterative procedure, and seems unable to sweep their whole spectra.

By analogy with the box-constrained case, the previous analysis suggests that the modified BB2 rule (3.18) can be exploited within the alternating strategy (1.32) and its adaptive version (1.34). The resulting updating rules will be denoted, respectively, by $\alpha_k^{\text{EQ-ABB}_{\min}}$ and $\alpha_k^{\text{EQ-VABB}_{\min}}$, summarized below

$$\alpha_k^{\text{EQ-VABB}_{\min}} = \begin{cases} \min\left\{\alpha_j^{\text{EQ-BB2}} : j = \max\{1, k - m_\alpha\}, \dots, k\right\} & \text{if } \dfrac{\alpha_k^{\text{EQ-BB2}}}{\alpha_k^{\text{BB1}}} < \tau_k \\ \alpha_k^{\text{BB1}} & \text{otherwise} \end{cases} \tag{3.40}$$

where $m_\alpha \leqslant 0$, τ_k is updated as in (1.35); for $\tau_k \equiv \tau_1$ the steplength $\alpha_k^{\text{EQ-ABB}_{\min}}$ is recovered.
We can guarantee that $1/\alpha_k^{\text{EQ-VABB}_{\min}}$ and $1/\alpha_k^{\text{EQ-ABB}_{\min}}$ belongs to the spectrum of $\tilde{U}_{k-1}^T A \tilde{U}_{k-1}$ at any iteration only when $m_\alpha = 0$. Indeed, if $m_\alpha > 0$, the inequalities (3.30) do not hold, in general, for $\alpha_j^{\text{EQ-BB2}}$ with $j = \max\{1, k - m_a\}, \dots, k - 1$. However, small values for m_α are acceptable since the final active set stabilizes at some point of the iterative process. We compare the spectral behaviours of the updating strategies $\text{ABB}_{\min}$, $\text{EQ-ABB}_{\min}$ and $\text{EQ-VABB}_{\min}$ within the GP method on the previous toy problem. The results obtained for $m_\alpha = 2$ are shown in Figure 8. We observe that the pattern shown by the sequence $\{\frac{1}{\alpha_k}\}$ generated by $\text{ABB}_{\min}$ is negatively influenced by the presence of the standard BB2 rule, resulting in a poor behaviour. The improvement achieved by using the modified rule EQ-BB2 within the alternating strategies is clear from the corresponding panels in Figure 8.

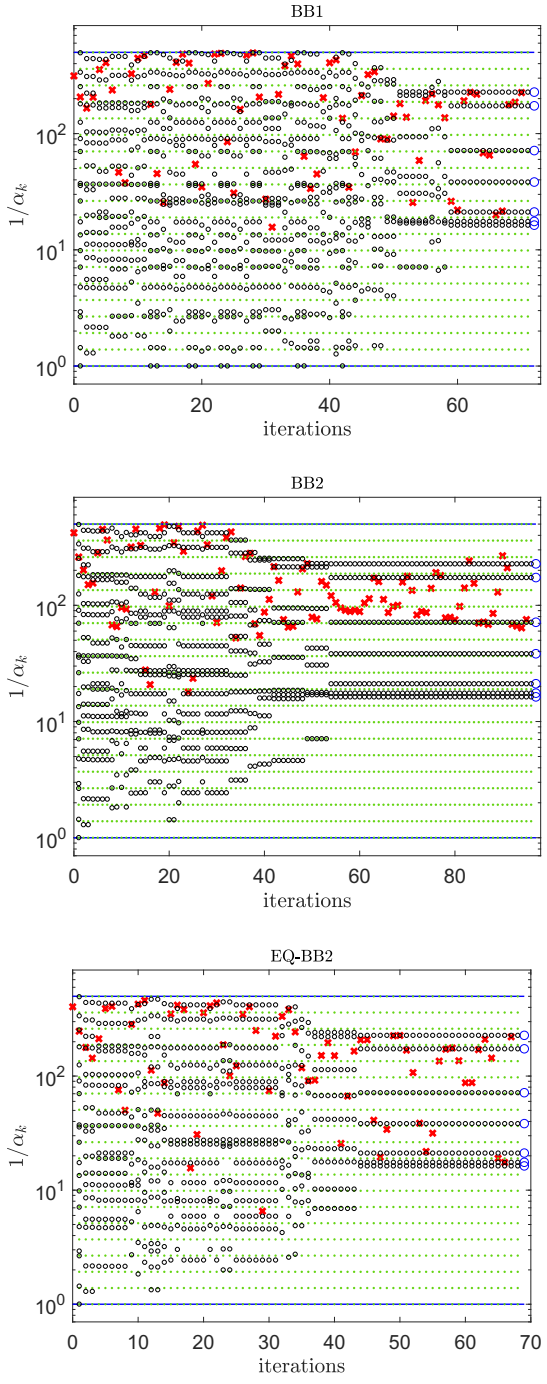

Figure 7: Distribution of $\left\{\frac{1}{\alpha_k}\right\}$ (red crosses) with respect to iterations for BB1 (top panel), BB2 (middle panel) and EQ-BB2 (bottom panel) on a toy problem of size $n = 20$.

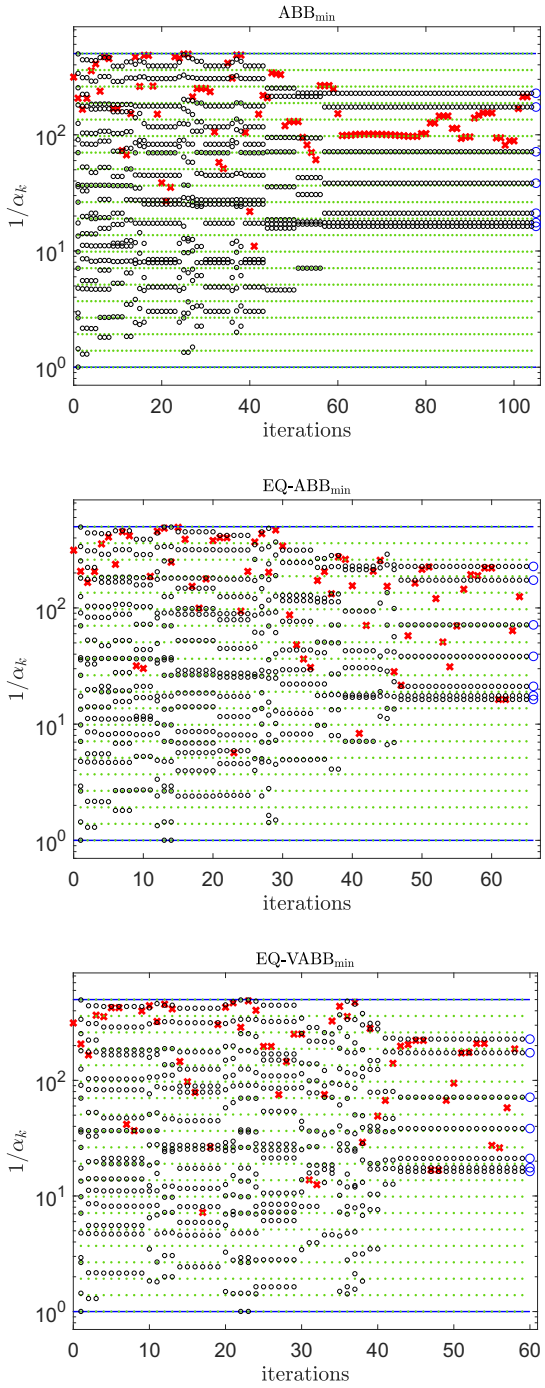

Figure 8: Distribution of $\left\{\frac{1}{\alpha_k}\right\}$ (red crosses) with respect to iterations for ABB$_{\min}$ (top panel), EQ-ABB$_{\min}$ (middle panel) and EQ-VABB$_{\min}$ (bottom panel) on a toy problem of size $n = 20$.

3.3.1 *The non-quadratic case*

In this section we derive the spectral properties of α_k^{BB1} and $\alpha_k^{\text{EQ-BB2}}$ when the GP method is applied for solving a more general constrained problem of type (3.1).
The multidimensional variant of Taylor's theorem [65, Theorem 11.1] allows to write the following equation:

$$y^{(k-1)} = \int_0^1 \nabla^2 f(x^{(k-1)} + \rho s^{(k-1)}) s^{(k-1)} d\rho. \tag{3.41}$$

From (3.15) and (3.41) and by recalling that $s^{(k-1)}_{\mathcal{J}_{k-1}} = \mathcal{O}_{n-m_k,1}$, it holds

$$\begin{aligned} t^{(k-1)}_{\mathcal{J}_{k-1}} &= [P_{k-1} \;\; \mathcal{O}_{m_k,n-m_k}]\, y^{(k-1)} = \\ &= \int_0^1 P_{k-1} \nabla^2 f(x^{(k-1)} + \rho s^{(k-1)})_{\mathcal{I}_{k-1},\mathcal{J}_{k-1}} s^{(k-1)}_{\mathcal{J}_{k-1}} d\rho. \end{aligned}$$

From the equality $s^{(k-1)}_{\mathcal{I}_{k-1}} = P_{k-1} s^{(k-1)}_{\mathcal{J}_{k-1}}$ and the properties of projection matrices, we have

$$\begin{aligned} {s^{(k-1)}}^T y^{(k-1)} &= {s^{(k-1)}_{\mathcal{J}_{k-1}}}^T t^{(k-1)}_{\mathcal{J}_{k-1}} \\ &= {s^{(k-1)}_{\mathcal{J}_{k-1}}}^T \int_0^1 \tilde{W}_{k-1}\tilde{W}_{k-1}^T \nabla^2 f(x^{(k-1)} + \rho s^{(k-1)})_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}} \\ &\qquad \tilde{W}_{k-1}\tilde{W}_{k-1}^T s^{(k-1)}_{\mathcal{J}_{k-1}} d\rho \\ &= {s^{(k-1)}_{\mathcal{J}_{k-1}}}^T \tilde{W}_{k-1} \int_0^1 \left(\tilde{U}_{k-1}^T \nabla^2 f(x^{(k-1)} + \rho s^{(k-1)}) \tilde{U}_{k-1}\right) \\ &\qquad \tilde{W}_{k-1}^T s^{(k-1)}_{\mathcal{J}_{k-1}} d\rho. \end{aligned} \tag{3.42}$$

Since ${s^{(k-1)}_{\mathcal{J}_{k-1}}}^T \tilde{W}_{k-1}\tilde{W}_{k-1}^T s^{(k-1)}_{\mathcal{J}_{k-1}} = {s^{(k-1)}}^T s^{(k-1)}$, from the previous equation we can conclude that $1/\alpha_k^{\text{EQ-BB1}}$ corresponds to a Rayleigh quotient related to the average matrix $\tilde{U}_{k-1}^T \nabla^2 f(x^{(k-1)} + \rho s^{(k-1)}) \tilde{U}_{k-1}$ along the vector $\tilde{W}_{k-1}^T s^{(k-1)}_{\mathcal{J}_{k-1}}$.

In order to give a similar interpretation for $\alpha_k^{\text{EQ-BB2}}$, we consider the linear function $\phi\colon \mathbb{R}^{m-1} \to \mathbb{R}^{m-1}$ defined as

$$\phi(x_{\tilde{W}}) = z_{\tilde{W}} = \tilde{W}_{k-1}^T P_{k-1} \nabla f(x_{\mathcal{I}_{k-1}}, x_{\mathcal{J}_{k-1}})_{\mathcal{I}_{k-1}},$$

where $x_{\tilde{W}} \in \mathbb{R}^{m-1}$, $x_{\mathcal{I}_{k-1}} = \tilde{W}_{k-1} x_{\tilde{W}}$ and $x_{\mathcal{J}_{k-1}}$ is defined with respect to the iterations $k-1$ and k. We have

$$\tilde{W}_{k-1}\phi(x_{\tilde{W}}) = \tilde{W}_{k-1} z_{\tilde{W}} = P_{k-1} \nabla f(\tilde{W}_{k-1} x_{\tilde{W}}, x_{\mathcal{J}_{k-1}})_{\mathcal{I}_{k-1}}.$$

Let assume that ∇f is continuously differentiable and locally invertible in the intersection of $\mathcal{T}_{k-1}$ with a neighborhood of $x^{(k-1)}$ including $x^{(k)}$; in this intersection we can define the inverse function ϕ^{-1} as

$$\phi^{-1}(z_{\tilde{W}}) = x_{\tilde{W}} \Leftrightarrow \phi(x_{\tilde{W}}) = z_{\tilde{W}},$$

or, equivalently,

$$\begin{aligned}\phi^{-1}(z_{\tilde{W}}) = x_{\tilde{W}} \quad &\Leftrightarrow \quad \tilde{W}_{k-1}^T P_{k-1} \nabla f(\tilde{W}_{k-1} x_{\tilde{W}}, x_{\mathcal{J}_{k-1}})_{\mathcal{I}_{k-1}} = \\ &= \quad \tilde{W}_{k-1}^T \nabla f(\tilde{W}_{k-1} x_{\tilde{W}}, x_{\mathcal{J}_{k-1}})_{\mathcal{I}_{k-1}} = z_{\tilde{W}}.\end{aligned}$$

The Jacobian matrix of ϕ^{-1} at $z_{\tilde{W}}$ is

$$(\tilde{W}_{k-1}^T \nabla^2 f(\tilde{W}_{k-1} x_{\tilde{W}}, x_{\mathcal{J}_{k-1}})_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}} \tilde{W}_{k-1})^{-1},$$

equal to the inverse of $\tilde{U}_{k-1}^T \nabla^2 f(x_{\mathcal{I}_{k-1}}, x_{\mathcal{J}_{k-1}}) \tilde{U}_{k-1}$.
Setting

$$\begin{aligned}\phi^{-1}(z_{\tilde{W}}^{(k-1)}) &= x_{\tilde{W}}^{(k-1)}, \quad \text{with } x_{\mathcal{I}_{k-1}}^{(k-1)} = \tilde{W}_{k-1} x_{\tilde{W}}^{(k-1)}, \\ \phi^{-1}(z_{\tilde{W}}^{(k)}) &= x_{\tilde{W}}^{(k)}, \quad \text{with } x_{\mathcal{I}_{k-1}}^{(k)} = \tilde{W}_{k-1} x_{\tilde{W}}^{(k)},\end{aligned}$$

we can write

$$x_{\tilde{W}}^{(k)} - x_{\tilde{W}}^{(k-1)} = \int_0^1 (\tilde{U}_{k-1}^T \nabla^2 f(x^{(k-1)} + \rho s^{(k-1)}) \tilde{U}_{k-1})^{-1} (z_{\tilde{W}}^{(k)} - z_{\tilde{W}}^{(k-1)}) d\rho.$$

Multiplying both the sides of the previous equality by ${y_{\mathcal{I}_{k-1}}^{(k-1)}}^T \tilde{W}_{k-1}$, we have

$$\begin{aligned}{y_{\mathcal{I}_{k-1}}^{(k-1)}}^T s_{\mathcal{I}_{k-1}}^{(k-1)} &= {y_{\mathcal{I}_{k-1}}^{(k-1)}}^T \tilde{W}_{k-1} \left(x_{\tilde{W}}^{(k)} - x_{\tilde{W}}^{(k-1)} \right) \\ &= {y_{\mathcal{I}_{k-1}}^{(k-1)}}^T \tilde{W}_{k-1} \int_0^1 (\tilde{U}_{k-1}^T \nabla^2 f(x^{(k-1)} + \rho s^{(k-1)}) \tilde{U}_{k-1})^{-1} \\ &\qquad \tilde{W}_{k-1}^T y_{\mathcal{I}_{k-1}}^{(k-1)} d\rho.\end{aligned}$$

Since ${t_{\mathcal{I}_{k-1}}^{(k-1)}}^T t_{\mathcal{I}_{k-1}}^{(k-1)} = {y_{\mathcal{I}_{k-1}}^{(k-1)}}^T \tilde{W}_{k-1} \tilde{W}_{k-1}^T y_{\mathcal{I}_{k-1}}^{(k-1)}$, we can conclude that $\alpha_k^{\text{EQ-BB2}}$ can be interpreted as a Rayleigh quotient relative to the average inverse of the matrix $\tilde{U}_{k-1}^T \nabla^2 f(x^{(k-1)} + \rho s^{(k-1)}) \tilde{U}_{k-1}$ along the vector $\tilde{W}_{k-1}^T y_{\mathcal{I}_{k-1}}^{(k-1)}$.

REMARK Inspired by the idea at the basis of the classical BB rules, we observe that the steplength selections (3.17) and (3.18) can be interpreted also as solutions of the following modified secant conditions:

$$\alpha_k^{\text{BB1}} = \arg\min_{\alpha \in \mathbb{R}} \|\alpha^{-1} s_{\mathcal{I}_{k-1}}^{(k-1)} - t_{\mathcal{I}_{k-1}}^{(k-1)}\|, \tag{3.43}$$

$$\alpha_k^{\text{EQ-BB2}} = \arg\min_{\alpha \in \mathbb{R}} \|s_{\mathcal{I}_{k-1}}^{(k-1)} - \alpha t_{\mathcal{I}_{k-1}}^{(k-1)}\|. \tag{3.44}$$

As final remark, we observe that this analysis for general non-quadratic case holds, with the obvious modifications, also for the BB rules defined for box-constrained problems, since they are special cases of the version here considered.

3.3.2 *Variable metric gradient projection method*

In this section we consider the more general variable metric scheme defined in (1.51) and suggest how to properly modify the BB updating rules for this case. From a theoretical point of view, the convergence of the variable metric gradient projection method is still ensured for any value of the steplength α_k belonging to a compact subset of $\mathbb{R}_+$ [14, Thm. 2.1]; since this condition is not restrictive, it allows some flexibility in the definitions of novel rules.

A natural way to extend the previous ideas consists in requiring the steplength α_k to satisfy generalized secant conditions written in terms of the norm induced by the matrix $(D_k)_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}}$:

$$\begin{aligned}\alpha_k^{\text{P-BB1}} &= \arg\min_{\alpha\in\mathbb{R}} \|\alpha^{-1} s^{(k-1)}_{\mathcal{I}_{k-1}} - (D_k^{-1})_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}} t^{(k-1)}_{\mathcal{I}_{k-1}}\|_{(D_k)_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}}}\\ \alpha_k^{\text{P-EQ-BB2}} &= \arg\min_{\alpha\in\mathbb{R}} \|s^{(k-1)}_{\mathcal{I}_{k-1}} - \alpha(D_k^{-1})_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}} t^{(k-1)}_{\mathcal{I}_{k-1}}\|_{(D_k)_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}}},\end{aligned} \tag{3.45}$$

providing the following updating rules

$$\alpha_k^{\text{P-BB1}} = \frac{{s^{(k-1)}_{\mathcal{I}_{k-1}}}^T (D_k)_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}} s^{(k-1)}_{\mathcal{I}_{k-1}}}{{s^{(k-1)}_{\mathcal{I}_{k-1}}}^T t^{(k-1)}_{\mathcal{I}_{k-1}}}, \tag{3.46}$$

$$\alpha_k^{\text{P-EQ-BB2}} = \frac{{s^{(k-1)}_{\mathcal{I}_{k-1}}}^T t^{(k-1)}_{\mathcal{I}_{k-1}}}{{t^{(k-1)}_{\mathcal{I}_{k-1}}}^T (D_k^{-1})_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}} t^{(k-1)}_{\mathcal{I}_{k-1}}}. \tag{3.47}$$

Again, we may observe that

$$\alpha_k^{\text{P-BB1}} = \frac{{s^{(k-1)}_{\mathcal{I}_{k-1}}}^T (D_k)_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}} s^{(k-1)}_{\mathcal{I}_{k-1}}}{{s^{(k-1)}_{\mathcal{I}_{k-1}}}^T t^{(k-1)}_{\mathcal{I}_{k-1}}} = \frac{{s^{(k-1)}}^T D_k s^{(k-1)}}{{s^{(k-1)}}^T y^{(k-1)}},$$

while the rule (3.47) can be viewed as the modified version of the following strategy

$$\alpha_k^{\text{P-BB2}} = \frac{{s^{(k-1)}}^T y^{(k-1)}}{{y^{(k-1)}}^T D_k^{-1} y^{(k-1)}}, \tag{3.48}$$

which takes into account the presence of the scaling matrix but does not consider the inactive constraints of the feasible region at each iteration. Furthermore, the steplengths (3.46)-(3.47) reduce, respectively to the standard BB1 and EQ-BB2 in case of non-scaled gradient methods, i. e. when the matrix D_k is equal to the identity matrix for all k. It is interesting to observe that, when ${s^{(k-1)}}^T y^{(k-1)} > 0$, from the inequality

$$\left({s^{(k-1)}_{\mathcal{I}_{k-1}}}^T t^{(k-1)}_{\mathcal{I}_{k-1}}\right)^2 \leqslant \left({s^{(k-1)}_{\mathcal{I}_{k-1}}}^T (D_k)_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}} s^{(k-1)}_{\mathcal{I}_{k-1}}\right)\left({t^{(k-1)}_{\mathcal{I}_{k-1}}}^T (D_k^{-1})_{\mathcal{I}_{k-1},\mathcal{I}_{k-1}} t^{(k-1)}_{\mathcal{I}_{k-1}}\right),$$

we easily obtain that $\alpha_k^{\text{P-EQ-BB2}} \leqslant \alpha_k^{\text{P-BB1}}$. This suggests that the rule (3.46) and (3.47) are suitable to be tested within a strategy generalizing the alternating scheme(3.49) to the variable metric case:

$$\alpha_k^{\text{P-EQ-VABB}_{\min}} = \begin{cases} \min\{\alpha_j^{\text{P-EQ-BB2}} \ : \ j = \max\{1, k - m_\alpha\}, \dots, k\} & \text{if } \frac{\alpha_k^{\text{P-EQ-BB2}}}{\alpha_k^{\text{P-BB1}}} < \tau_k, \\ \alpha_k^{\text{P-BB1}} & \text{otherwise,} \end{cases} \tag{3.49}$$

where m_α and τ_k are defined as in (1.34) and (1.35).

4

NUMERICAL EXPERIMENTS

This chapter is devoted to evaluating the practical effectiveness of the proposed steplength selection rules within GP methods for solving special constrained optimization problems. We report and discuss the results of different numerical investigations to evaluate the effectiveness of the techniques developed in Chapters 2 and 3. First, in order to provide practical evidences of the theoretical analysis, we analyse the spectral behaviour of the steplength rules on some selected medium-scale quadratic problems; afterwards, we evaluate the performance of the methods on a set of large-scale benchmark quadratic test problems; then, we test the ability of the methods to solve general non-quadratic minimization problems. Furthermore, we test our approaches on some optimization problems arising from real-world applications.

All the numerical experiments were carried out on a workstation equipped with an Intel Xeon QuadCore E5620 processor at 2,40 GHz and 18 Gb of RAM, by implementing the GP methods in the Matlab R2019a environment.

The chapter is organized as follows. Section 4.1 is devoted to the numerical tests on BQP problems, whereas in Section 4.2 SLBQP problems are considered. In Section 4.3 we report the results obtained on general non-quadratic box-constrained and SLB test problems.

4.1 NUMERICAL EXPERIMENTS ON BQP PROBLEMS

In this section we provide a comparison of the spectral behaviours of the original BB-rules and the limited memory approaches with those of the modified versions proposed in Chapters 2 on BQP test problems. To this aim, we start by studying how the steplengths sequences generated by each rule are distributed with respect to the eigenvalues of the reduced Hessian matrices obtained during the GP iterative process. Then, we test the efficiency of the methods on a dataset of large-scale BQP problems, reporting the results by means of performance profiles plots. We conclude the section with a comparison between the GP method combined with the BoxVABB$_{\text{min}}$ rule and the MPRGP method on some real-world applications.

Table 1: Spectral features of the BQP test problems subject to lower bounds

	$n = 1000$		$na = 400$	
	$\lambda_{min}(A)$	$\lambda_{max}(A)$	$\lambda_{min}(A_{\mathcal{J}^*,\mathcal{J}^*})$	$\lambda_{max}(A_{\mathcal{J}^*,\mathcal{J}^*})$
BQP1	1	1000	41.78	957.19
BQP2	3.10	997.70	64.57	819.53
BQP3	1	1000	3.08	753.26
BQP4	9.40	1013.95	10	1000

4.1.1 *Spectral inspection on BQP problems*

We generated four test problems of the form (2.1), with size $n = 10^3$, $u = +\infty$, and dense SPD Hessian matrices A, characterized by the following different distributions of eigenvalues:

BQP1: for $i = 1, \dots, n$,

$$\lambda_i = \frac{\underline{\lambda} + \overline{\lambda}}{2} + \frac{\underline{\lambda} - \overline{\lambda}}{2} \cos\left(\frac{\pi(i-1)}{n-1}\right),$$

where $\underline{\lambda} = 1$ and $\overline{\lambda} = 10^3$;

BQP2: for $i = 1, \dots, n$,

$$\lambda_i = \frac{(\underline{\lambda} b - \overline{\lambda} a)}{(b-a)} + \frac{(\underline{\lambda} - \overline{\lambda})}{(b-a)} \omega_i,$$

where $\underline{\lambda} = 1$, $\overline{\lambda} = 10^3$, $a = (1-c)^2$, $b = (1+c)^2$, $c = 1/2$ and the values ω_i are distributed according to the Marčenko-Pastur density $p_c(x) = \dfrac{\sqrt{(b-a)(x-a)}}{2\pi x c^2}$, $a < x < b$ [62];

BQP3: logarithmic distribution in $[1, 10^3]$ such that $\lambda_1 = 1$, $\lambda_n = 10^3$ and $\dfrac{\lambda_i}{\lambda_{i-1}}$ is constant, generated through the MATLAB function `logspace`;

BQP4: eigenvalues of the reduced Hessian matrix at the solution x^* with logarithmic distribution in $[10, 10^3]$, generated through the MATLAB function `logspace`.

For each problem, the optimal solution x^* is randomly chosen from a uniform distribution in $(0, 5)$ and it has prefixed number na of active variables; the vector b is set as $b = Ax^*$ and the feasible initial point $x^{(0)}$ is randomly generated in order to have inactive entries. Other spectral features of the problems are reported in Table 1. Problems BQP1 and BQP2 are suggested also in [50, 73].
The GP variants exploiting the different steplength rules are distinguished by means of the rule's name (BB1, BB2, BoxBB2, BoxABB $_{\min}$,

BoxVABB $_{\min}$), whereas the limited memory GP approaches are named as in Section 2.2, p.43. With regard to Algorithm 2, the following parameter setting is used: $\alpha_0 = \frac{g^{(0)T} g^{(0)}}{g^{(0)T} A g^{(0)}}$, $\alpha_{min} = 10^{-10}$, $\alpha_{max} = 10^6$, $\sigma = 10^{-4}$, $\delta = 0.5$.
For the BB-based rules a nonmonotone GLL linesearch procedure with $M = 9$ is considered, while for LMGP, Box-LMGP1, Box-LMGP2 the value $M = 1$ is used; furthermore, $\tau = \tau_1 = 0.5$ and $m_\alpha = 2$ are set in BoxABB $_{\min}$ and BoxVABB $_{\min}$, whereas in the limited memory approaches two different values for the parameter m are used, i. e. $m = 3, 5$. All the methods share the same stopping criterion defined as in (2.23), with $tol = 10^{-8}$.

Figures 9-12 show the behaviour of BB1, BB2, ABB$_{\min}$, BoxBB2, BoxABB $_{\min}$ and BoxVABB$_{\min}$ rules on the four test problems. In each panels of the figures, at the k-th iteration, the black dots denote 20 eigenvalues of the reduced Hessian matrix $A_{\mathcal{I}_{k-1}\mathcal{I}_{k-1}}$, with linearly spaced indices (included the maximum and the minimum eigenvalues), and the red crosses correspond to the reciprocals of the steplengths α_k. The blue lines show the maximum and the minimum eigenvalues of the Hessian matrix A and the blue circles on the right are used to plot 20 eigenvalues of the reduced Hessian matrix at the solution x^*, with linearly spaced indices (included the maximum and the minimum eigenvalues). Figures 13-16 report, for each problem, the history of relative error on the solution, $\|x^{(k)} - x^*\|/\|x^*\|$, the function error $f(x^{(k)}) - f(x^*)$, and the projected gradient norm, $\|g^P(x^{(k)})\|$. The plots in Figures 9-12 confirm the ability of BB1 and BoxBB2 rules to produce steplengths α_k whose reciprocals belong to the spectrum of the reduced Hessian matrix $A_{\mathcal{I}_{k-1}\mathcal{I}_{k-1}}$, at each iteration. Furthermore, we may observe that, at the last iterations, the eigenvalues of the reduced Hessian matrices tend to stabilize and to well approximate the eigenvalues of $A_{\mathcal{I}^*\mathcal{I}^*}$. This means that the two rules play a role in reducing the gradient components with indices belonging to $\mathcal{I}^*$, which are the gradient components that have to be zero at the solution, as prescribed by the KKT conditions. On the other hand, the reciprocals of the BB2 steplengths seem unable to effectively sweep the spectrum of the reduced Hessian matrices with dangerous effects on the performance. The benefits achieved with the BoxBB2 rule reflect also on the behaviour of the alternating strategies BoxABB $_{\min}$ and BoxVABB$_{\min}$, which confirm their ability to speed up the method within the same or (in some cases) slightly higher accuracy with respect to the single rules, as it is clear from the convergence plots in Figures 13-16.

The results obtained with the limited memory approaches are shown in Figures 17-21. The distribution of the sequences $\{\frac{1}{\alpha_k}\}$ generated by LMGP, Box-LMGP1 and Box-LMGP2 rules are reported in Figures 17-20, for each problem. From these plots we may observe that the Box-

LMGP1 and Box-LMGP2 strategies are able to better approximate the eigenvalues of the submatrices of A restricted to rows and columns corresponding to the inactive components of the current iterate rather than the LMGP algorithm; in particular, for the Box-LMGP1 method, the choice $m = 5$ seems preferable, while the adaptive strategy implemented by Box-LMGP2 approach seems to be less dependent on the choice of the parameter m, confirming its ability in adaptively controlling the length of the sweep. Indeed, such ability allows to consider shorter sweeps at the beginning of the iterative process when the sequence of the restricted Hessian matrices is not yet stabilized towards $A_{\mathcal{F}*,\mathcal{F}*}$ and, hence, the inverses of the steplengths generated by the strategy in a sweep could not yet provide suitable approximations of the eigenvalues of the restricted Hessian matrices involved in the next sweep; longer sweeps are instead promoted with the stabilization of the final active set. Panels in Figure 21 report the history of the relative error on the solution, $\|x^{(k)} - x^*\|/\|x^*\|$ and the function error $f(x^{(k)}) - f(x^*)$, for each problem. These plots reveal how both the modified approaches Box-LMGP1 and Box-LMGP2 are, in general, able to accelerate the GP method with respect to the LMGP strategy. We cannot observe a particular supremacy of one over the other.

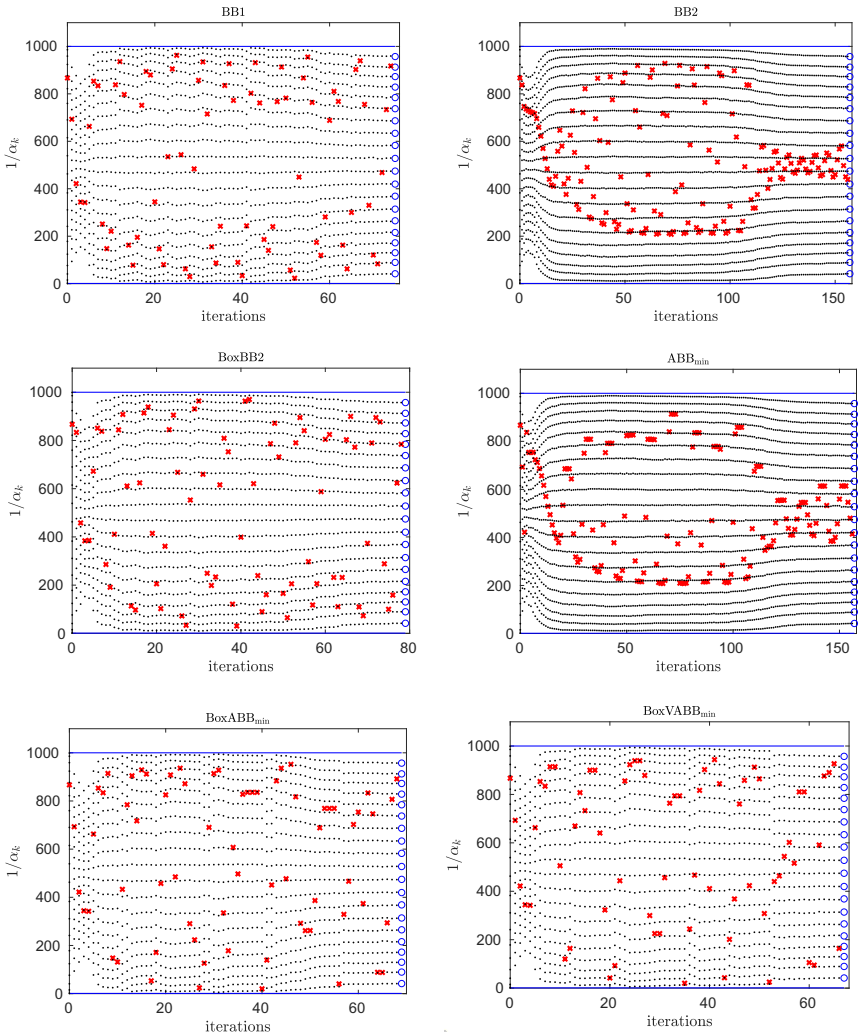

Figure 9: Behaviour of GP equipped with BB-based steplength rules on BQP1. Distribution of $\left\{\frac{1}{\alpha_k}\right\}$ with respect to the iterations for BB1, BB2 (top panels), BoxBB2, ABB$_{\min}$ (middle panels), BoxABB$_{\min}$ and BoxVABB$_{\min}$ (bottom panels).

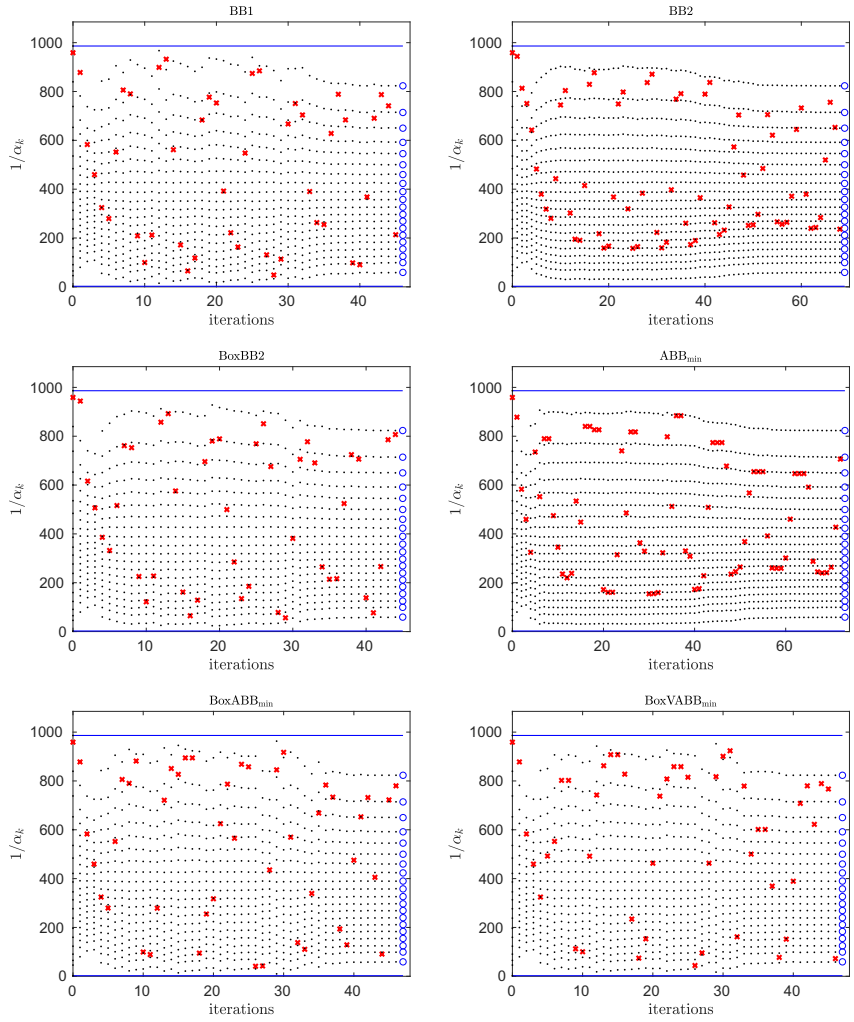

Figure 10: Behaviour of GP equipped with BB-based steplength rules on BQP2. Distribution of $\left\{\frac{1}{\alpha_k}\right\}$ with respect to the iterations for BB1, BB2 (top panels), BoxBB2, ABB$_{min}$ (middle panels), BoxABB$_{min}$ and BoxVABB$_{min}$ (bottom panels).

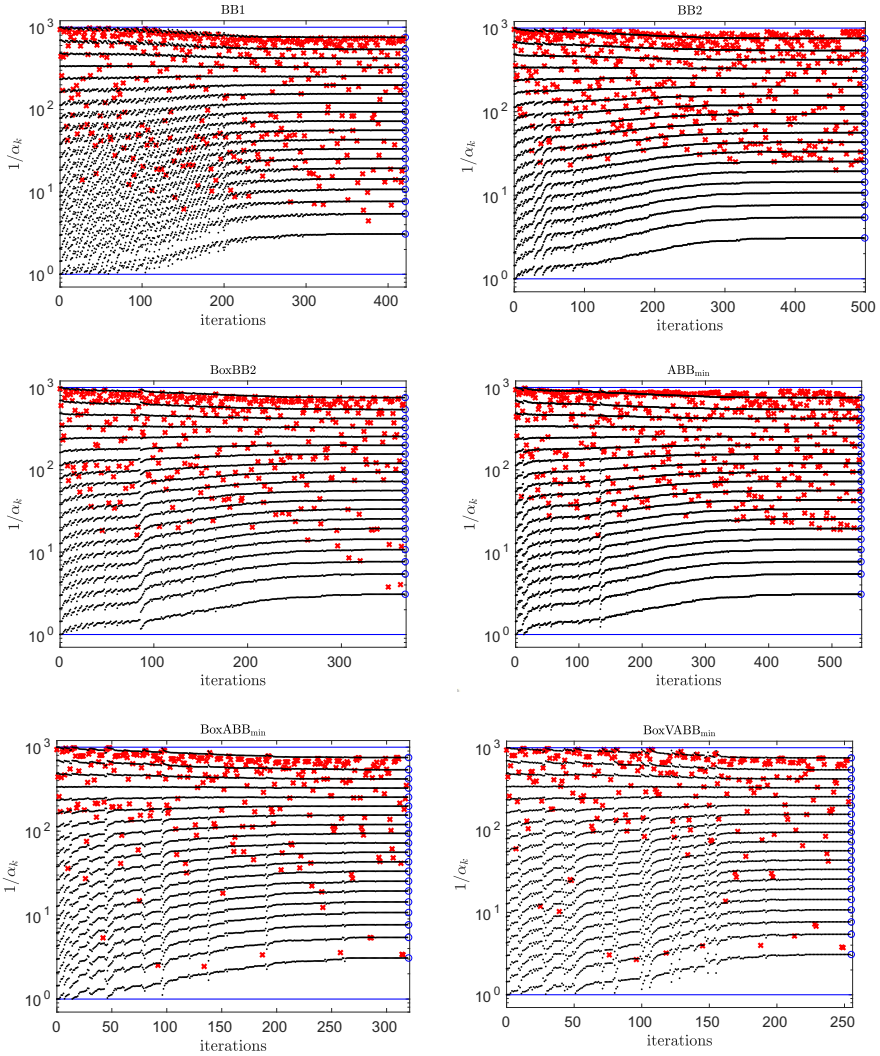

Figure 11: Behaviour of GP equipped with BB-based steplength rules on BQP3. Distribution of $\{\frac{1}{\alpha_k}\}$ with respect to the iterations for BB1, BB2 (top panels), BoxBB2, ABB$_{\min}$ (middle panels), BoxABB $_{\min}$ and BoxVABB$_{\min}$ (bottom panels).

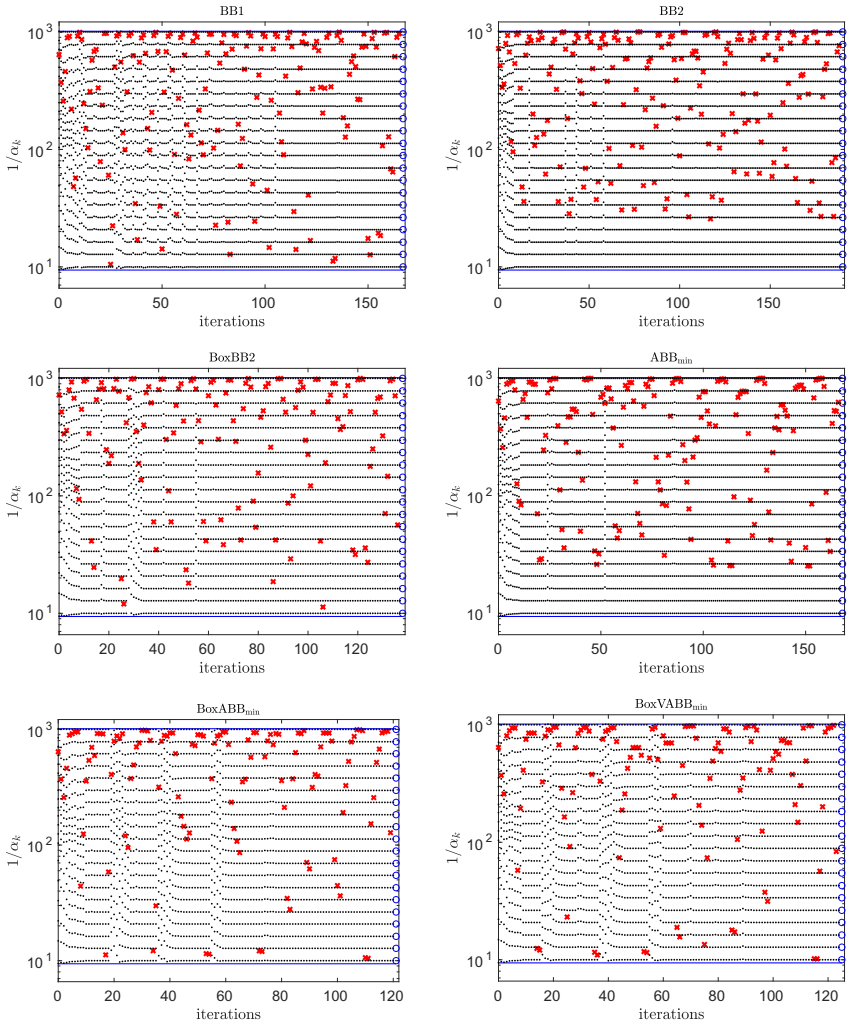

Figure 12: Behaviour of GP equipped with BB-based steplength rules on BQP4. Distribution of $\{\frac{1}{\alpha_k}\}$ with respect to the iterations for BB1, BB2 (top-left panels), BoxBB2, ABB$_{\text{min}}$ (top-right panels), BoxABB $_{\text{min}}$ and BoxVABB$_{\text{min}}$ (bottom panels).

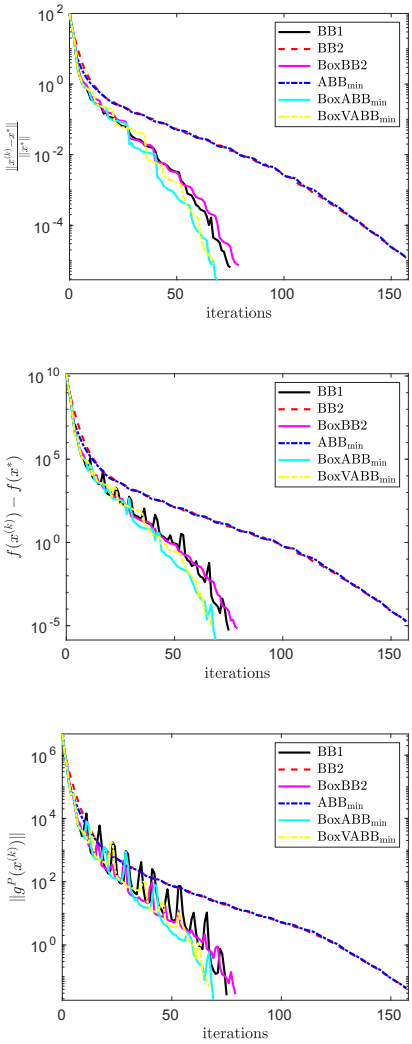

Figure 13: Test problem BQP1. History of relative error on the solution (top panel), function error (middle panel), projected gradient norm (bottom panel).

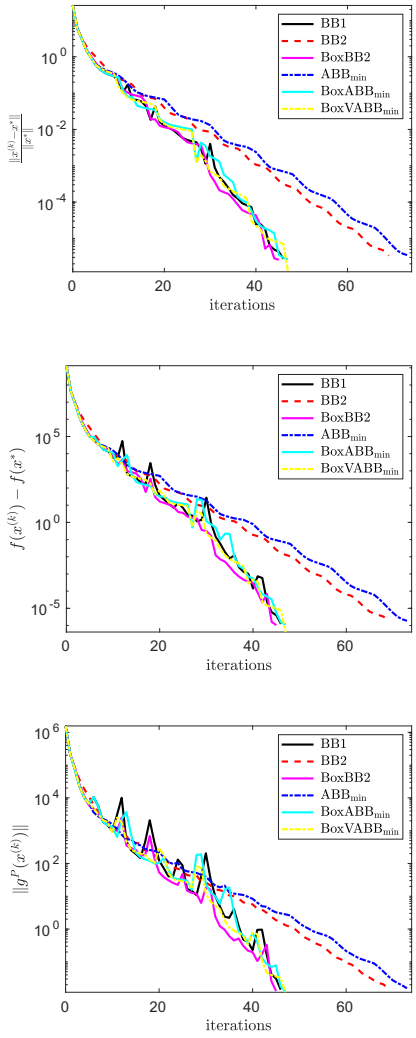

Figure 14: Test problem BQP2. History of relative error on the solution (top panel), function error (middle panel), projected gradient norm (bottom panel).

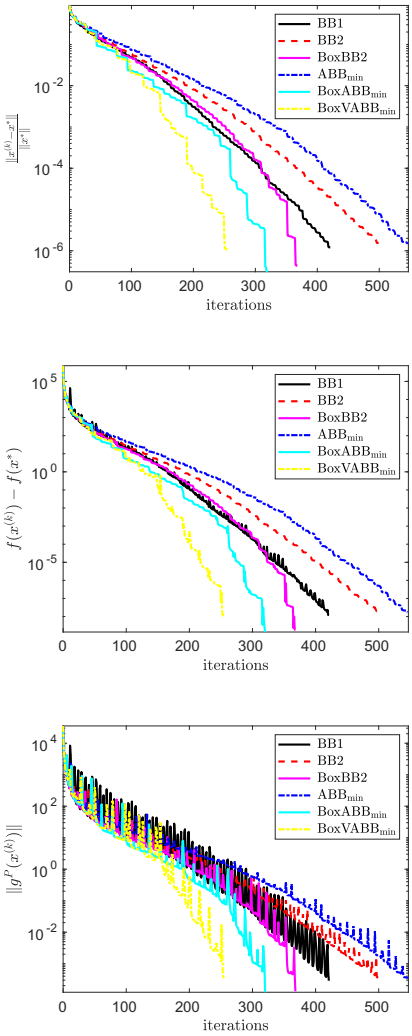

Figure 15: Test problem BQP3. History of relative error on the solution (top panel), function error (middle panel), projected gradient norm (bottom panel).

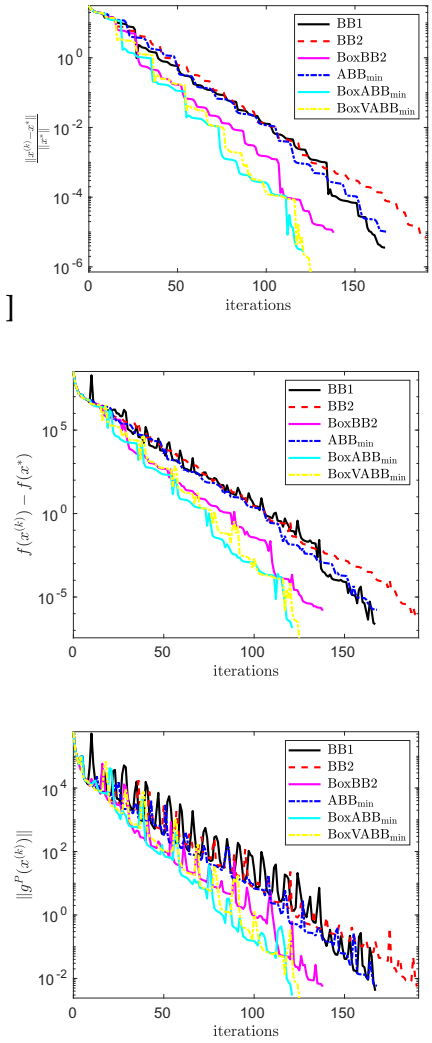

Figure 16: Test problem BQP4. History of relative error on the solution (top panel), function error (middle panel), projected gradient norm (bottom panel).

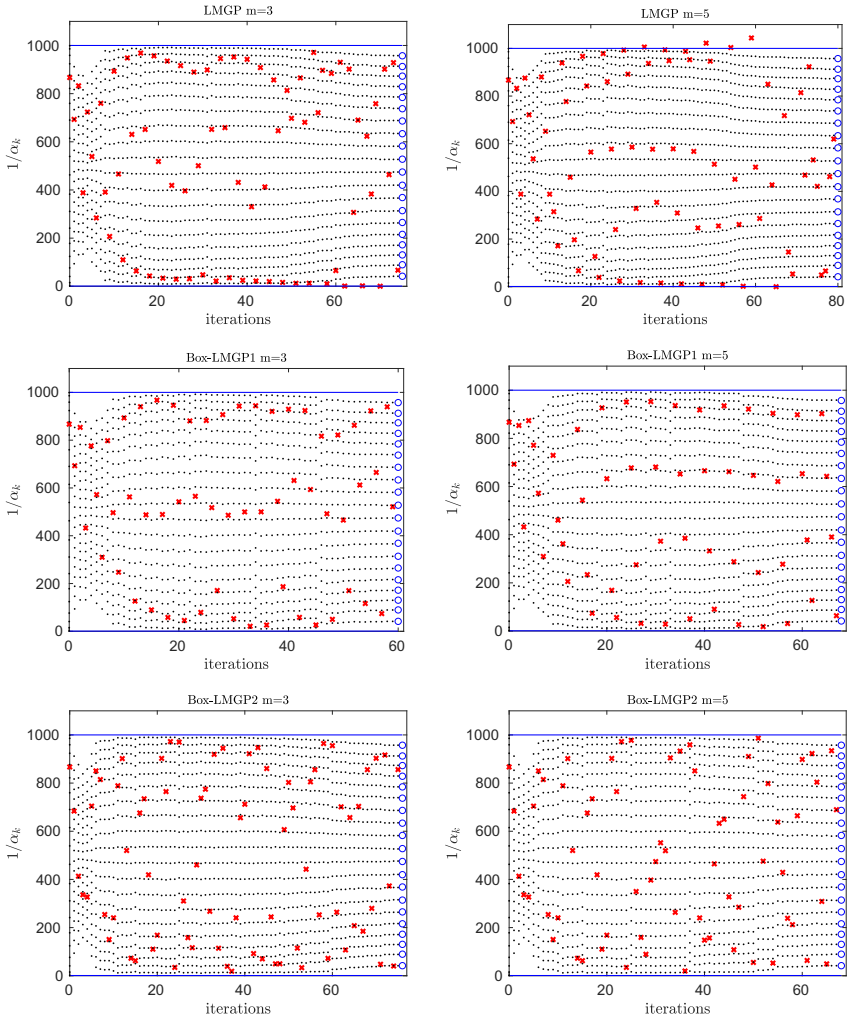

Figure 17: Behaviour of limited memory approaches on BQP1. Distribution of $\{\frac{1}{\alpha_k}\}$ with respect to the iterations for LMGP (top panels), Box-LMGP1(middle panels), Box-LMGP2 (bottom panels), for $m = 3$ (first column) and $m = 5$ (second column).

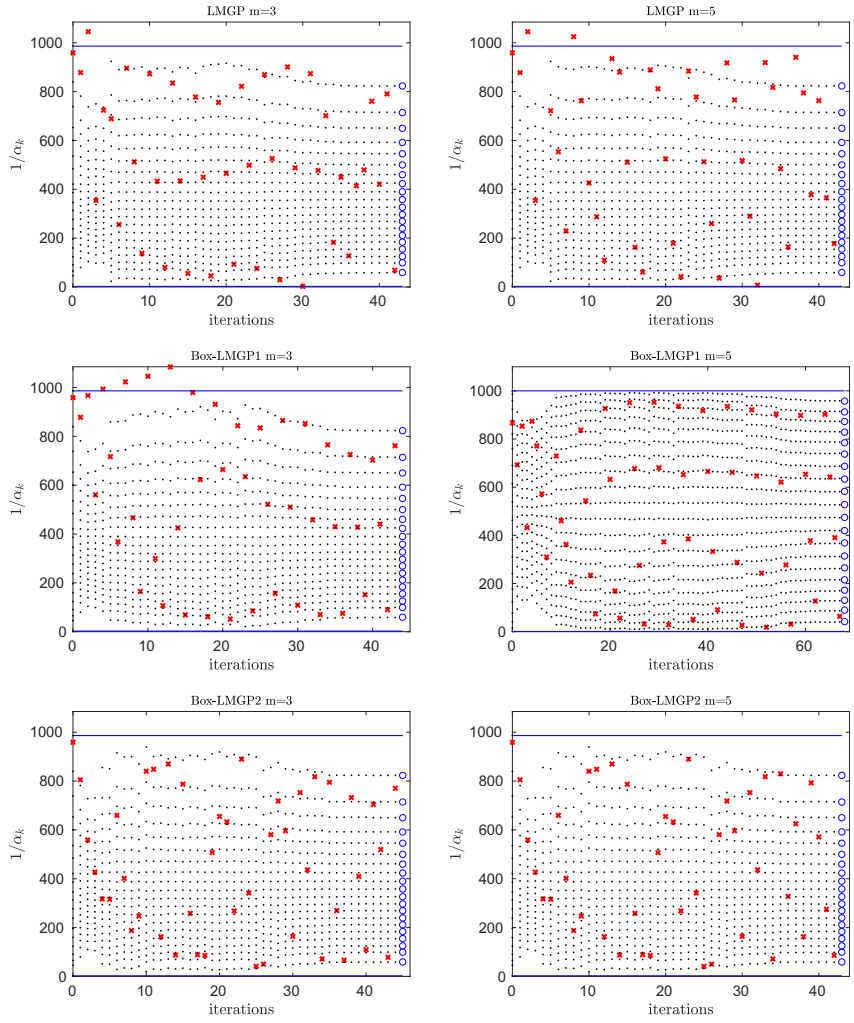

Figure 18: Behaviour of limited memory approaches on BQP2. Distribution of $\{\frac{1}{\alpha_k}\}$ with respect to the iterations for LMGP (top panels), Box-LMGP1(middle panels), Box-LMGP2 (bottom panels), for $m = 3$ (first column) and $m = 5$ (second column).

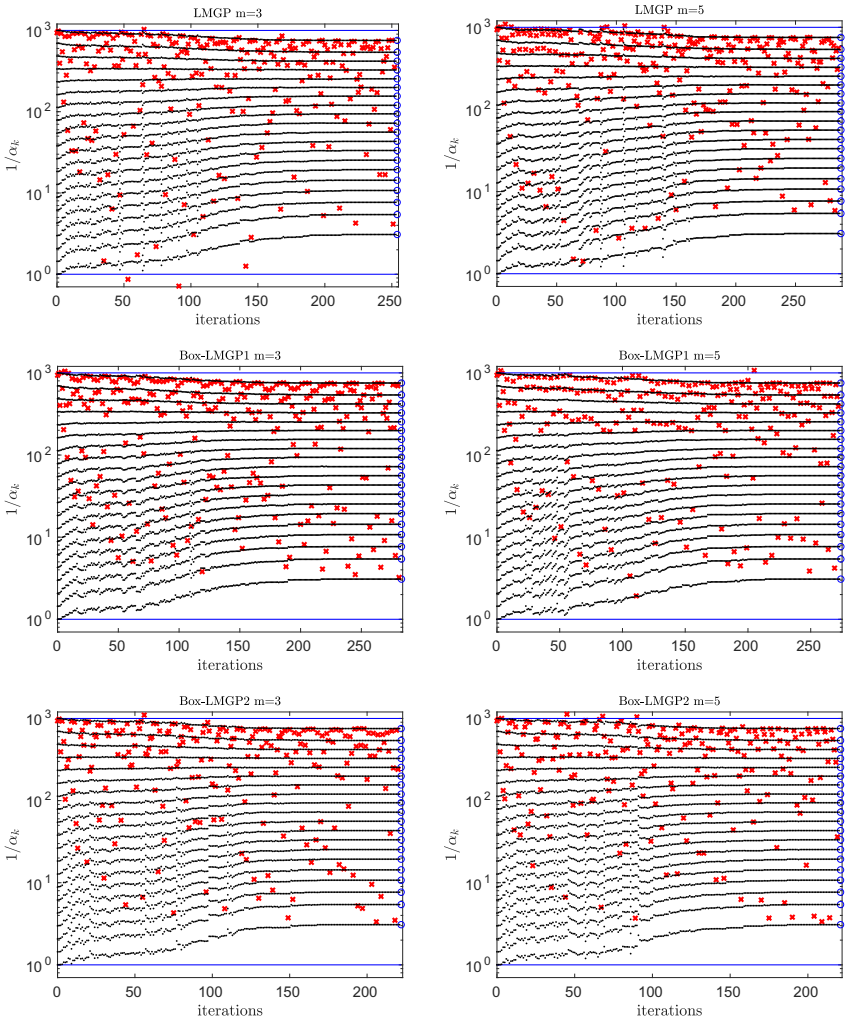

Figure 19: Behaviour of limited memory approaches on BQP3. Distribution of $\{\frac{1}{\alpha_k}\}$ with respect to the iterations for LMGP (top panels), Box-LMGP1(middle panels), Box-LMGP2 (bottom panels), for $m = 3$ (first column) and $m = 5$ (second column).

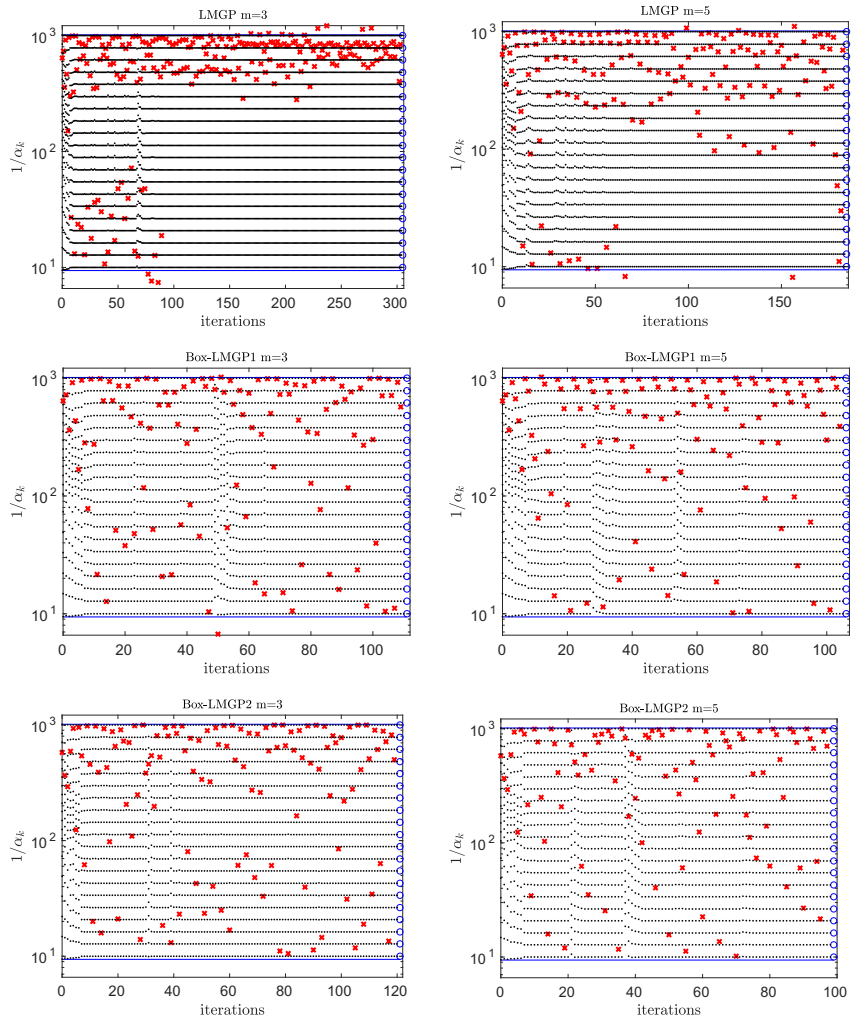

Figure 20: Behaviour of limited memory approaches on BQP4. Distribution of $\{\frac{1}{\alpha_k}\}$ with respect to the iterations for LMGP (top panels), Box-LMGP1(middle panels), Box-LMGP2 (bottom panels), for $m = 3$ (first column) and $m = 5$ (second column).

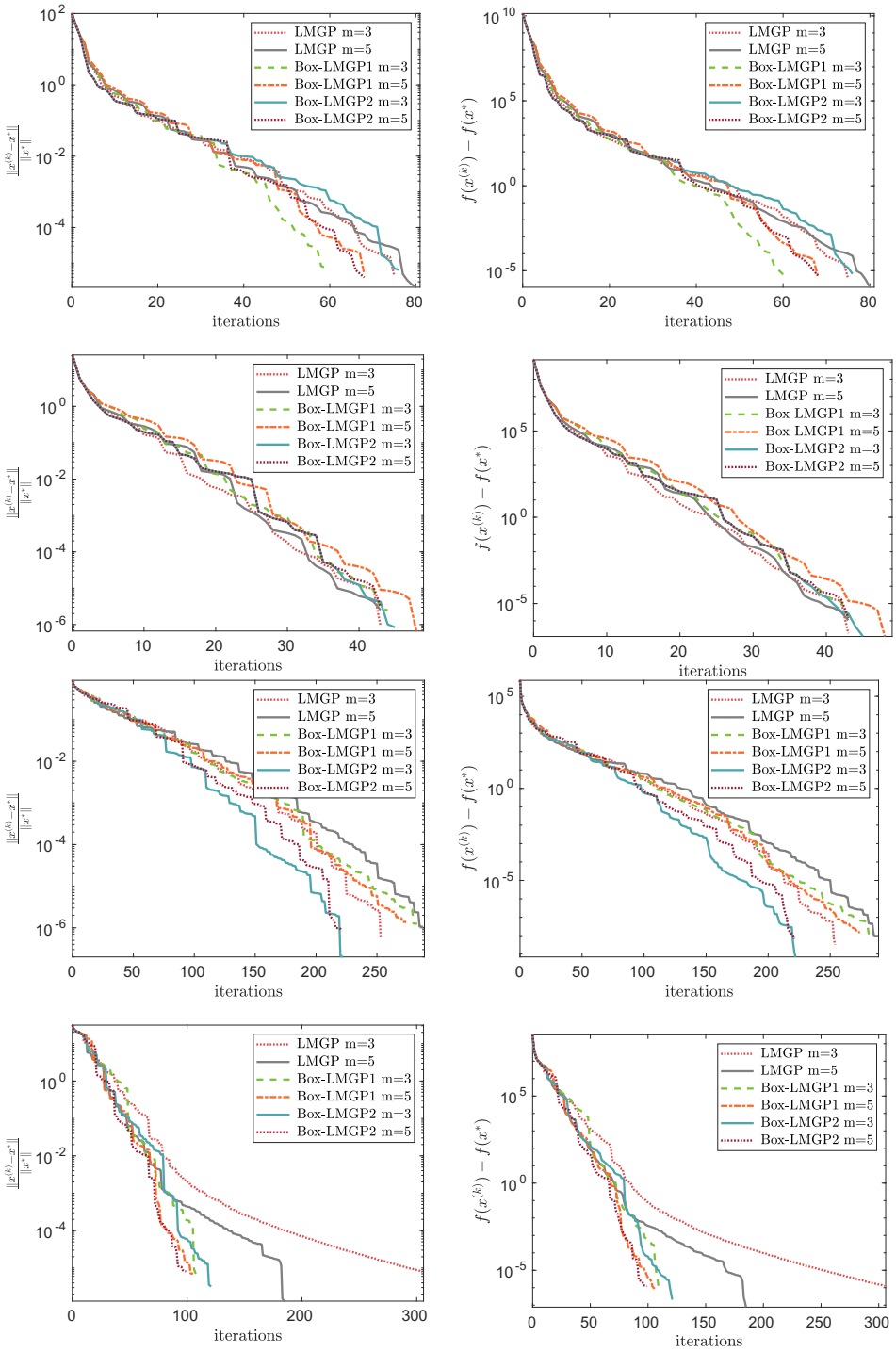

Figure 21: History of relative error on the solution (left panel) and function error (right panel) for test problems BQP1 (first row), BQP2 (second row), BQP3 (third row), BQP4 (fourth row).

4.1.2 *Numerical results on random large-scale BQP problems*

In order to verify the effectiveness of the methods on a more exhaustive set of large-scale problems, we used the software package downloadable at `http://www.dimat.unina2.it/diserafino/dds_sw.htm` for generating random BQP problems of the form (1.7). Following the procedure proposed in [63, 79], the software allows to generate test problems with prefixed features (size, number of active constraints at the solution, condition number, etc), which can be arbitrarly set by the user. For our experiments. we assigned different values to the following parameters:

- n, size of the problem;
- $\kappa(A)$, spectral condition number of the Hessian matrix A;
- $na \in [0, 1)$, fraction of active variables at the solution x^*;
- $ndeg \in \{0, 1, 2, \dots, \}$, amount of near-degeneracy.

Then we generated two sets of test problems:

DATASET 1: 108 strictly convex BQP problems with non-degenerate solutions, obtained setting $n = 15000,\ 20000,\ 25000$, $na = 0.1, 0.5, 0.9$, $\kappa(A) = 10^4, 10^5, 10^6$, $ndeg = 1, 4, 7, 10$.

DATASET 2: 108 strictly convex BQP problems with degenerate solutions, obtained setting $n = 15000,\ 20000,\ 25000$, $na = 0.1, 0.5, 0.9$, $\kappa(A) = 10^4, 10^5, 10^6$, $ndeg = 1, 4, 7, 10$. The fraction of active variables at the solution that are degenerate is equal to 0.2.

For each problem, the solution x^* is randomly chosen from a uniform distribution in $(-1, 1)$, the starting point $x^{(0)}$ is set equal to $\dfrac{\ell + u}{2}$ and the Hessian matrix is defined as $A = GDG^T$ where G is the product of three Householder matrices associated to randomly generated unit vectors and D is the diagonal matrix of the eigenvalues, which are log-spaced between 1 and $\kappa(A)$. Hence, the matrix A is not stored in memory but only the operator for computing the matrix-vector products is available. Furthermore, the levels of near-degeneracy are obtained by setting the positive Lagrangian multipliers associated to the active constraints equal to $10^{-\eta_i ndeg}$, $i = 1, \dots, na \cdot n$, where $\eta_i \in (0, 1)$ is a random number.
The parameter setting of the methods is the same of the previous section and the stopping criterion is defined by (2.23), with $tol = 10^{-7}$. Concerning the alternating rules $\text{BoxABB}_{\text{min}}$ and $\text{BoxVABB}_{\text{min}}$, we remark that the choice $m_\alpha = 2$ appears preferable. The behaviours of the BB-based steplength rules and limited memory strategies are evaluated by means of the performance profile plots proposed in [36]. In particular, to compare the methods, we consider as performance

measure of interest the execution time required by each solver to satisfy the stopping criterion. To this aim, for each problem p in a given dataset $\mathcal{P}$ of n_p problems, we need to compute the *performance ratio* $r_{p,s}$ defined as the computing time of the solver s divided by the best time of all the solvers; then, the performance profile of a solver s is given by

$$\rho_s(\theta) = \frac{\sum_{p\in\mathcal{P}} P(r_{p,s},\theta)}{n_p},$$

where $\theta \geqslant 1$ and

$$P(r_{p,s},\theta) = \begin{cases} 1 & \text{if} \quad r_{p,s} \leqslant \theta, \\ 0 & \text{otherwise.} \end{cases}$$

Hence, $\rho_s(\theta)$ is the probability for solver s that $r_{p,s} \leqslant \theta$. If the stopping criterion is not fulfilled by a solver within the maximum number of 40000 iterations, the corresponding performance ratio is set equal to a fixed value r_M larger than any ratio $r_{p,s}$.
For all the test problems described above, the performance profiles of the different GP versions are reported in Figures 22-29. Each profile $\rho(\theta)$ gives the fraction of problems that a solver is able to solve within a factor θ of the best time of all the solvers; in particular, the value $\rho(1)$ represents the fraction of problems for which the considered solver is the winner, whereas the performance profile corresponding to the largest value of θ gives the fraction of problems for which the considered solver is successful.

The BoxBB2 steplength rule is compared with the standard BB2 and BB1 rules in the top-left and middle-left panels of Figure 22, respectively, whereas a comparison between BoxBB2 and BoxABB$_{\min}$ is provided in the top-right panel; a summarizing analysis including also the BoxVABB$_{\min}$ is given in the bottom-right panel of Figure 22. Figure 22 shows that, on DATASET 1 , the BoxBB2 rule is generally preferable to the standard BB2 and BB1 selections, which are not successful on the total amount of test problems. Furthermore, the adaptive alternation of BB1 and BoxBB2 selections exploited in the BoxABB$_{\min}$ rule is still more convenient than the use of a single steplength rule. In particular, the version BoxVABB$_{\min}$ is able to further improve the performance provided by the BoxABB$_{\min}$ thanks to the updating formula (1.35) for the switching parameter τ.
The general trend shown in the bottom-right panel of Figure 22 is confirmed also by the plots in Figure 23; indeed, this last figure highlights that the results are independent from the different values of the parameter na. Finally, Figure 24 shows the performance profiles obtained on the subsets of BQP test problems with non-degenerate solution generated by setting $ndeg = 7$ (left panel) and $ndeg = 10$ (right panel), i. e. in the near-degeneracy case. The comparison of Figure 24 with the bottom-right panel of Figure 22 puts in evidence that

the level of degeneracy of the constraints weakly affects the performance of the different rules.

Performance profiles of the limited memory approaches are reported in Figures 25-27. Figure 25 refers to DATASET 1; in particular, in the top-left panel the performance profiles of LMGP ($m = 3$) and Box-LMGP1 ($m = 3$) are reported: we can observe that the LMGP and Box-LMGP1 methods for $m = 3$ are not successful on about 20% of the problems and, hence, their version with $m = 5$ are preferable; on the other hand, the adaptive strategy implemented within the Box-LMGP2 approach enables to solve all the problems in both cases, however the best performance is achieved for $m = 5$, as shown in the bottom-left panel of the figure. The best version of the different strategies are compared, in pairs, in the top-right and middle-right panels of Figure 25; finally, the summarizing plot in the bottom-right panel shows that the Box-LMGP1 ($m = 5$) approach is able to solve about 80% of the problems within the best computing time with respect to both LMGP and Box-LMGP2 ($m = 5$), while this last one is still preferable to the LMGP strategy. Figure 26 highlights the results obtained on the subsets of problems characterized by different values of the parameter na: in particular, when the number of active constraints at the solution is equal to $0.1 \cdot n$ (top-left panel) the performances of LMGP and Box-LMGP2 improve with respect to the other cases, where the general behaviour observed in Figure 25 is confirmed. The performance profiles obtained for $ndeg = 7$ and $ndeg = 10$ are reported in Figure 27: also in this case the level of degeneracy of the constraints seems to weakly affect the performance of the different rules.

In Figure 28 the runtime performance profile of the BB-based rules and limited memory approaches are compared on the set of test problems with non-degenerate solutions (DATASET 1): we may observe that the GP method equipped with $\text{BoxVABB}_{\text{min}}$ rule clearly outperforms the other solvers, however the Box-LMGP1 method proves to be competitive with $\text{BoxVABB}_{\text{min}}$, while BoxLMGP2 exhibits a runtime performance profile similar to that of BoxBB2 steplength rule.

The results obtained on the set of BQP problem with degenerate solutions (DATASET 2) are summarized in Figure 29. Comparing the panels in Figure 29 with the corresponding profiles obtained on DATASET 1, we can observe that the performances of the solvers are weakly influenced by the presence of degeneracy, confirming the overall behaviours exhibit in the non-degenerate case.

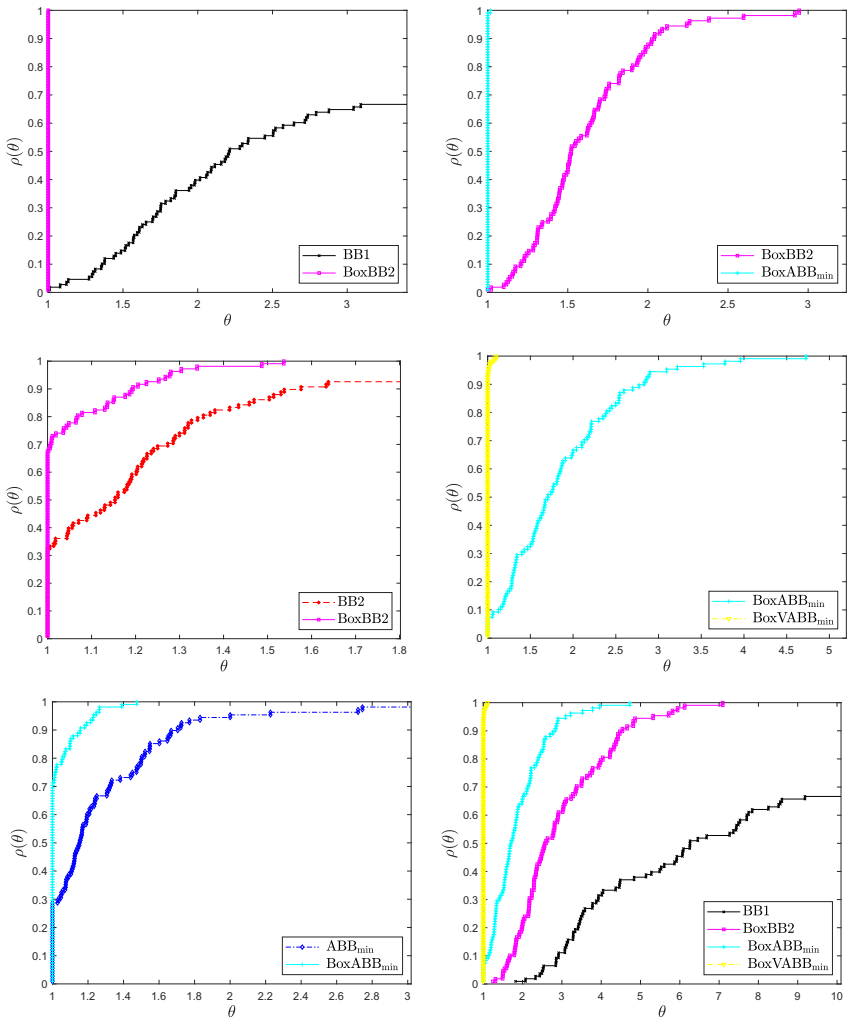

Figure 22: Runtime performance profiles of GP method equipped with BB-based rules on DATASET 1 of BQP test problems with non-degenerate solutions.

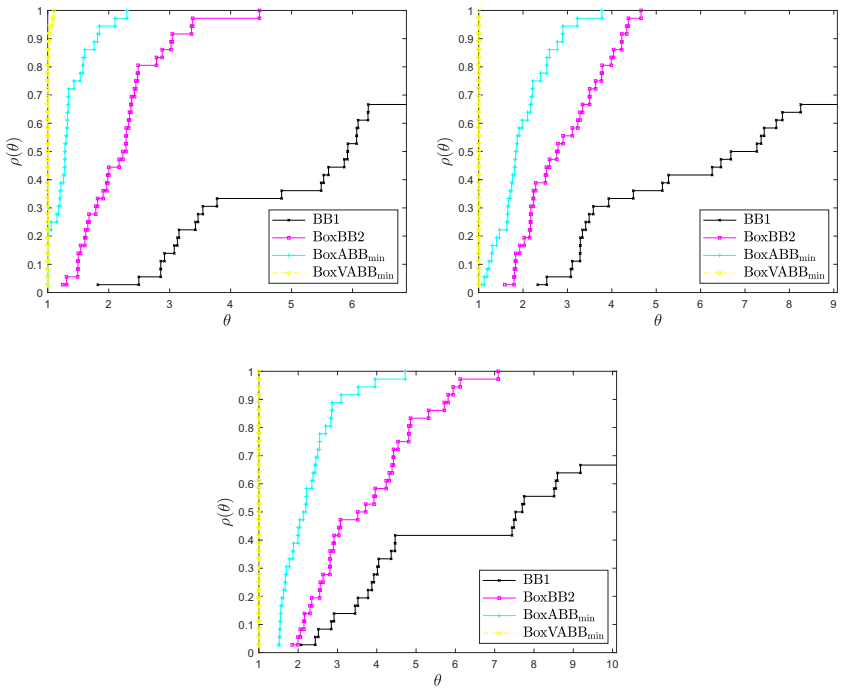

Figure 23: Runtime performance profiles of the GP method equipped with BB-based rules on DATASET 1 of BQP test problems with non-degenerate solutions. Fraction of active constraints at the solution: $0.1 \cdot n$ (top-left panel), $0.5 \cdot n$ (top-right panel), $0.9 \cdot n$ (bottom panel).

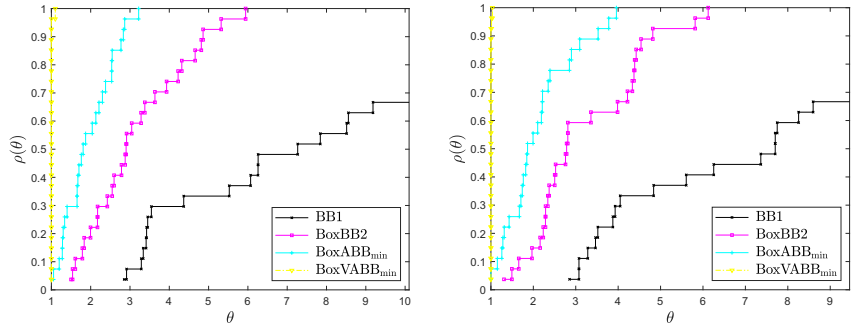

Figure 24: Runtime performance profiles of the GP method equipped with BB-based rules on the subsets of DATASET 1 corresponding to the parameters $ndeg = 7$ (left panel) and $ndeg = 10$ (right panel).

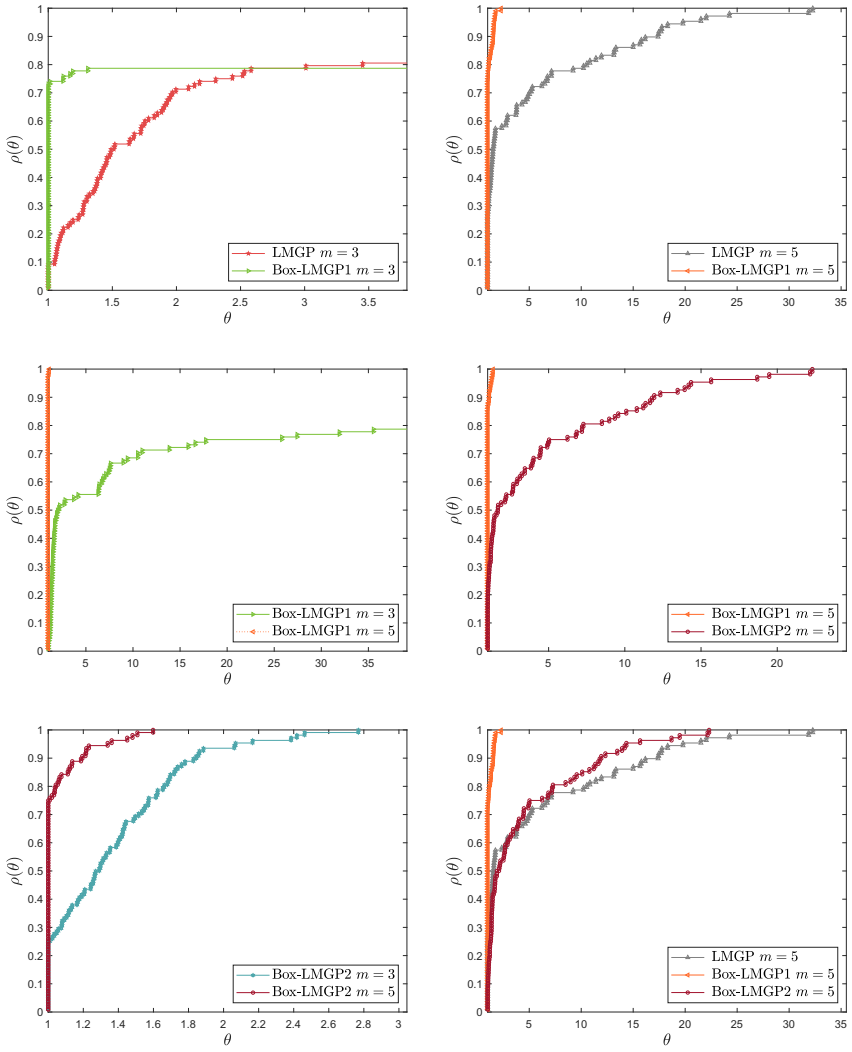

Figure 25: Runtime performance profiles of limited memory approaches on a DATASET 1 of BQP test problems with non-degenerate solutions.

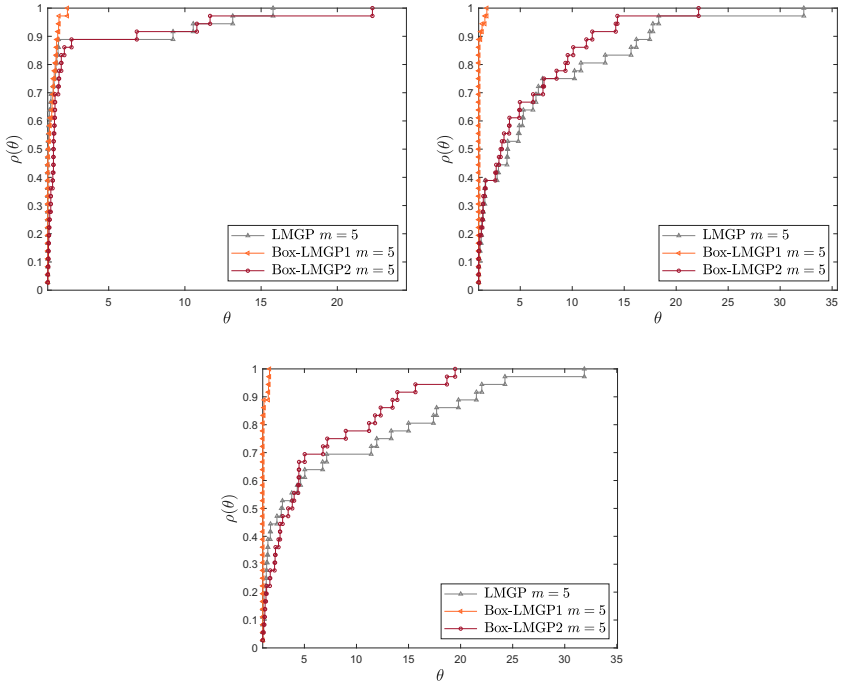

Figure 26: Runtime performance profiles of the GP method equipped with limited memory approaches on DATASET 1 of BQP test problems with non-degenerate solutions. Fraction of active constraints at the solution: $0.1 \cdot n$ (top-left panel), $0.5 \cdot n$ (top-right panel), $0.9 \cdot n$ (bottom panel).

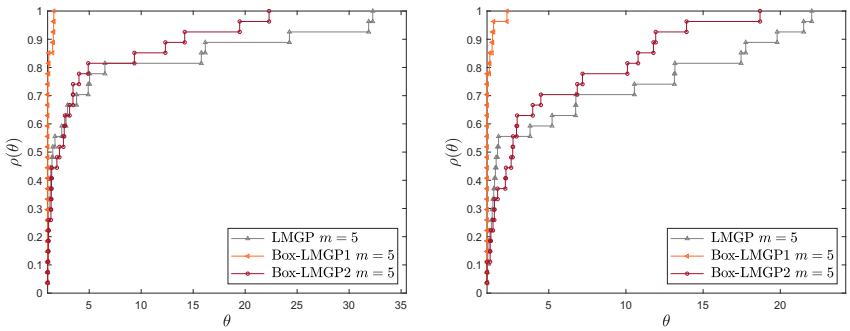

Figure 27: Runtime performance profiles of the limited memory approaches on the subsets of DATASET 1 corresponding to parameters $ndeg = 7$ (left panel) and $ndeg = 10$ (right panel).

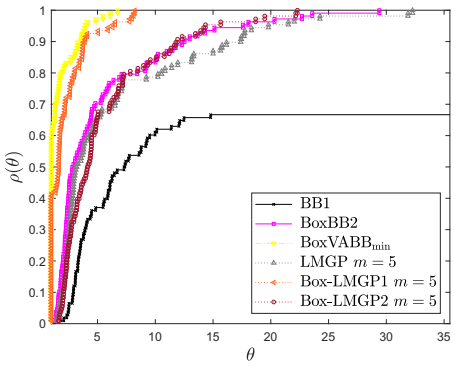

Figure 28: Runtime performance profiles of the GP method equipped with different steplength rules on the DATASET 1 of BQP test problems with non-degenerate solutions.

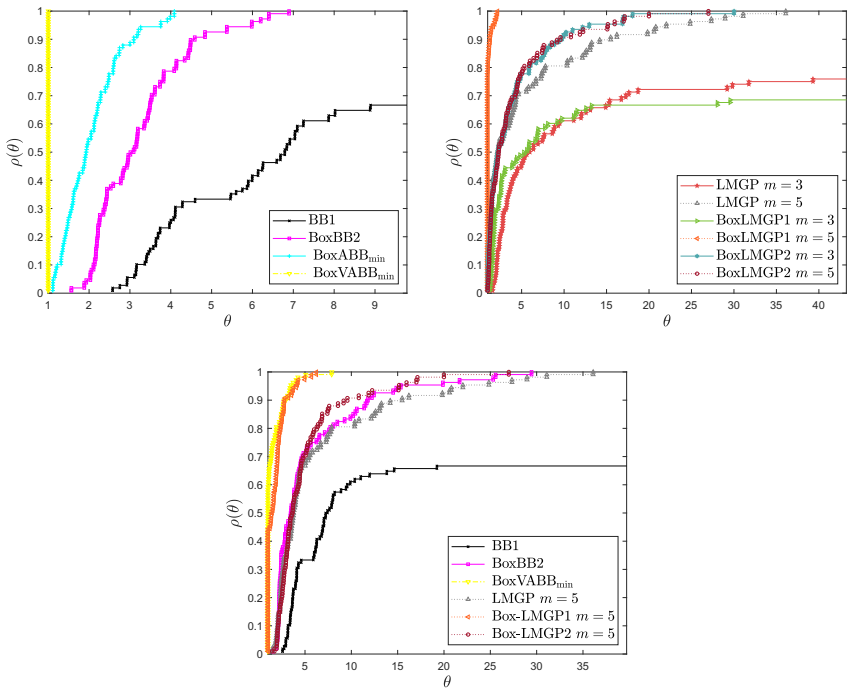

Figure 29: Runtime performance profiles of the GP method equipped with different steplength rules on the DATASET 2 of BQP test problems with degenerate solutions. Fraction of active variables at the solution that are degenerate equal to $0.2 \cdot n_a$.

4.1.3 *A comparison between GP-BoxVABB$_{min}$ method and MPRGP method on Support Vector Machines and contact problems*

Based on the soundness of the theoretical results shown in Chapter 2 and the numerical results obtained in the previous tests, we provide a comparison between the GP method combined with BoxVABB$_{\text{min}}$ rule and the MPRGP method [39], on some real applications related to SVMs and contact problems.
MPRGP scheme is presented in Algorithm 3.

Algorithm 3 Modified Proportioning with Reduced Gradient Projections (MPRGP) method

Initialize: A, $x^{(0)} \in \Omega$, b, $\Gamma > 0$, $\overline{\alpha} \in (0, 2||A||^{-1}]$
$g = Ax^{(0)} - b$, $p = \varphi(x^{(0)})$, $k = 0$;

while $||g^P(x^{(k)})||$ is not small **do**
  if $||\beta(x^{(k)})||^2 \leqslant \Gamma^2 g^r(x^{(k)})^T \varphi(x^{(k)})$ **then**
    $\alpha_f \leftarrow \max\{\alpha_{cg} : x^{(k)} - \alpha_{cg} p\}$;
    $\alpha_{cg} \leftarrow g^T p/(p^T A p)$;
    if $\alpha_{cg} < \alpha_f$ **then**
      $x^{(k+1)} \leftarrow x^{(k)} - \alpha_{cg} p$ CG step
      $g \leftarrow g - \alpha_{cg} A p$;
      $\beta \leftarrow \varphi(x^{(k+1)})^T A p / p^T A p$;
      $p \leftarrow \varphi(x^{(k+1)}) - \beta p$;
    else
      $x^{(k+\frac{1}{2})} \leftarrow x^{(k+1)} - \alpha_f p$ Expansion step
      $g \leftarrow g - \alpha_f p$;
      $x^{(k+1)} \leftarrow x^{(k+\frac{1}{2})} - \overline{\alpha} g^r(x^{(k+\frac{1}{2})})$;
      $g \leftarrow A x^{(k+1)} - b$;
      $p \leftarrow \varphi(x^{(k+1)})$;
    end if
  else
    $\alpha_{cg} \leftarrow g^T \beta(x^{(k)}) / (\beta(x^{(k)})^T A \beta(x^{(k)}))$ Proportioning step
    $x^{(k+1)} \leftarrow x^{(k)} - \alpha_{cg} \beta(x^{(k)})$;
    $g \leftarrow g - \alpha_{cg} A \beta(x^{(k)})$;
    $p \leftarrow \varphi(x^{(k+1)})$;
  end if
  $k \leftarrow k + 1$;
end while
return $x^{(k)}$

No-bias data classification

We start by reporting the results obtained on Support Vector Machines (SVM). We recall that Support Vector Machines belong to the

conventional machine learning techniques, and they are practically used for classification, i. e. the problem of identifying the category to which a new observation belongs. For a detailed discussion on SVMs framework we refer to [16, 22]).

In case of no-bias data binary classification, i. e. when the bias of the hyperplane from the origin is not considered in the classification model, the problem is formulated as a BQP problem with a symmetric positive semi-definite (SPS) Hessian matrix. We briefly outline the main features of the problem considered in our experiments.
Let $D = \{(z_i, y_i),\ \ i = 1, \dots, n,\ \ z_i \in \mathbb{R}^m,\ \ y_i \in \{-1, 1\}\}$ be a training set of labeled examples; the classification of new examples $z \in \mathbb{R}^m$ is performed by using a decision function $F : \mathbb{R}^m \longrightarrow \{-1, 1\}$ of the form

$$F(z) = \operatorname{sign}\left(\sum_{i=1}^{n} x_i^* y_i K(z, z_i)\right), \tag{4.1}$$

where $K\colon \mathbb{R}^m \times \mathbb{R}^m \longrightarrow \mathbb{R}$ denotes a kernel function and the vector $x^* = (x_1^*, \dots, x_n^*)^T$ is the solution of

$$\begin{aligned} \min f(x) &= \frac{1}{2} x^T A x - \sum_{i=1}^{n} x_i, \\ &\text{subject to } \ 0 \leqslant x \leqslant Ce. \end{aligned} \tag{4.2}$$

Here, $e = (1, 1 \dots, 1)^T \in \mathbb{R}^n$, $C > 0$ is a positive parameter arising from the penalization of the misclassification error, and the matrix A has entries $A_{ij} = y_i y_j K(z_i, z_j),\ \ i, j = 1, 2, \dots, n$. We refer to problem (4.2) as the ℓ_1-loss SVM formulation. This formulation can be modified so that the Hessian matrix becomes positive definite. Then, we consider also the problem

$$\begin{aligned} \min f(x) &= \frac{1}{2} x^T \left(A + C^{-1} I\right) x - \sum_{i=1}^{n} x_i, \\ &\text{subject to } \ x \geqslant 0, \end{aligned} \tag{4.3}$$

where the Hessian matrix is regularized by matrix $C^{-1}I$, obtaining a SPD matrix. Problem (4.3) is referred to as ℓ_2-loss SVM formulation.

For our experiments the classification datasets, named *mushrooms* and *phishing*, were downloaded from the LIBSVM dataset page at https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/binary.html. The *mushrooms* dataset is used for training a machine to determine whether a mushroom is edible or poisonous, whereas the *phishing* dataset is used for the detection of phishing websites. The main characteristics of the datsets are summarized in Table 2. Furthermore, a linear kernel is adopted, i. e. $K(z_i, z_j) = z_i^T z_j$.

Table 2: Characteristics of the classification datasets.

Dataset	Samples+	Samples-	features	training data	testing data
mushrooms	2830	2613	112	5443	2681
phishing	4073	3333	68	7406	3649

Table 3: Confusion matrix layout related to a binary classifier

		Actual class	
		Class A	Class B
Predicted class	Class A	True Positive (TP)	False Positive (FP)
	Class B	False Negative (FN)	True Negative (TN)

Tables 4-7 report the results obtained on these datasets with the GP-BoxVABB$_{\text{min}}$ method ($M = 9$, $m_\alpha = 2$, $\tau_1 = 0.5$, $\vartheta = 1.1$), the MPRGP method described in Algorithm 3 and its variant, named MPRGPp, where the expansion step is realized by a projected CG step method. The methods share the same stopping criterion defined as in (2.23) with $\mathtt{tol} = 0.1$. For each method, the tables report the number $\overline{n}$ of matrix-vector products performed by the algorithm, the computational time needed to satisfy the stopping criterion and the performance scores, namely accuracy (Acc.), precision (Prec.), sensitivity (Sens), F1-score. The symbol "$-$"denotes that the stopping criterion has not been satisfied within the maximum number of 10000 iterations. The performance scores are defined on the basis of the *confusion matrix*, reported in Table 3, which gives a first insight of the prediction quality of a classifier; indeed, when a classification system has been trained to distinguish between two classes, e. g. Class A and Class B, the confusion matrix summarizes the results of testing the algorithm, by counting the number of true positive (TP), false positive (FP), false negative (FN) and true negative (TN) samples.

Then, the perfomance scores are defined as follows:

- $\text{accuracy} = \dfrac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{FN}+\text{TN}} \cdot 100\%$,
- $\text{precision} = \dfrac{\text{TP}}{\text{TP}+\text{FP}} \cdot 100\%$,
- $\text{sensitivity} = \dfrac{\text{TP}}{\text{TP}+\text{FN}} \cdot 100\%$,
- $\text{F1-score} = 2\,\dfrac{\text{precision}\cdot\text{sensitivity}}{\text{precision}+\text{sensitivity}} \cdot 100\%$ i. e.F1 is the armonic mean of sensitivity and precision.

Tables 4 and 5 reports the results obtained on the *mushrooms* dataset for ℓ_1-loss and ℓ_2-loss SVM formulations, respectively. In particular, in case of ℓ_1-loss (Table 4) we can observe that the methods are not sensitive to the values assigned to parameter C, since the values of $\overline{n}$ and the performance scores are the same for each case; furthermore, the GP-BoxVABB$_{min}$ scheme is able to considerably reduce the number of matrix-vector multiplications and the computational time compared with the other two methods, while the best performance scores are obtained with MPRGPp; however, the performance scores related to GP-BoxVABB$_{min}$ and MPRGP are still satisfying. Also in the case of ℓ_2-loss formulation the GP-BoxVABB$_{min}$ method shows better result than those obtained with the MPRGP methods in terms of number of matrix-vector products and execution times, within the same quality, except for the cases $C = 10$ and $C = 50$, in which the MPRGPp method provides the best performance scores (see Table 5). On the other hand, we obtain performances of models corresponding to *phishing* dataset that are lower than in *mushrooms* dataset, as we can see from the results reported in Tables 6-7. Indeed, for each method, these results compared with those obtained on *mushrooms* show an increase in the number of matrix-vector products and computational times, and a decrease of the performance scores, which, however, are still greater than 90% in any case. We may observe that all the methods are, in general, less performing in solving the ℓ_1-loss formulation; the GP-BoxVABB$_{min}$ method is not able to satisfy the stopping criterion within the prefixed number of iterations in two cases (see Table 6); on the other hand, the results of each methods improve in the regularized version of the problem (see Table 7).

Table 4: Dataset *mushrooms*, ℓ_1-loss SVM formulation, $C = 1, 5, 10, 50, 100$. For each method the mean time over the five runs for the different value of C is reported.

	$\overline{n}$	time (s)	Acc.	Prec.	Sens.	F1
MPRGP	994	15.177	99.89%	100.00%	99.78%	99.78%
MPRGPp	264	4.0012	100.00%	100.00%	100.00%	100.00%
GP-BoxVABB$_{min}$	72	1.1119	99.89%	100.00%	99.78%	99.78%

Application to journal bearing problem

The journal bearing problem arises in the determination of the pressure distribution in a thin film of lubricant between two circular cylinders (see e. g. [4, 37, 64]). The infinite dimensional version of the problem is of the form

$$\min_{v \in K} q(v) \equiv \int_{\mathcal{D}} \left(\frac{1}{2} w_q(x) \|\nabla v(x)\|^2 - w_l(x) v(x) \right) dx$$

Table 5: Dataset *mushrooms*, ℓ_2-loss SVM formulation.

	$\overline{\pi}$	time (s)	Acc.	Prec.	Sens.	F1
			$C = 1$			
MPRGP	962	15.337	99.89%	100.00%	99.78%	99.78%
MPRGPp	286	4.2746	99.89%	100.00%	99.78%	99.78%
GP-BoxVABB$_{\text{min}}$	57	0.8347	99.89%	100.00%	99.78%	99.78%
			$C = 5$			
MPRGP	982	16.384	99.89%	100.00%	99.78%	99.78%
MPRGPp	301	4.4594	99.89%	100.00%	99.78%	99.78%
GP-BoxVABB$_{\text{min}}$	64	0.9472	99.89%	100.00%	99.78%	99.78%
			$C = 10$			
MPRGP	998	14.730	99.89%	100.00%	99.78%	99.78%
MPRGPp	307	4.5771	100.00%	100.00%	100.00%	100.00%
GP-BoxVABB$_{\text{min}}$	72	1.3334	99.89%	100.00%	99.78%	99.78%
			$C = 50$			
MPRGP	994	16.1077	99.89%	100.00%	99.78%	99.78%
MPRGPp	298	4.4650	100.00%	100.00%	100.00%	100.00%
GP-BoxVABB$_{\text{min}}$	63	0.9512	99.89%	100.00%	99.78%	99.78%
			$C = 100$			
MPRGP	994	15.3936	99.89%	100.00%	99.78%	99.78%
MPRGPp	294	4.3089	99.89%	100.00%	99.78%	99.78%
GP-BoxVABB$_{\text{min}}$	66	0.9876	99.89%	100.00%	99.78%	99.78%

Table 6: Dataset *phishing*, ℓ_1-loss SVM formulation.

	$\overline{n}$	time (s)	Acc.	Prec.	Sens.	F1
			$C = 1$			
MPRGP	3830	67.587	93.56%	94.91%	93.88%	93.88%
MPRGPp	175	2.8094	93.51%	95.20%	93.54%	93.54%
GP-BoxVABB$_{\text{min}}$	781	13.125	93.59%	94.77%	94.05%	94.05%
			$C = 5$			
MPRGP	4936	91.202	93.64%	94.72%	94.18%	94.18%
MPRGPp	341	5.5824	93.72%	95.01%	94.06%	94.06%
GP-BoxVABB$_{\text{min}}$	1321	22.446	93.70%	94.91%	94.10%	94.10%
			$C = 10$			
MPRGP	5356	96.723	93.89%	94.15%	95.10%	95.10%
MPRGPp	566	10.808	93.75%	93.81%	95.18%	95.18%
GP-BoxVABB$_{\text{min}}$	2639	63.909	93.67%	94.77%	94.18%	94.18%
			$C = 50$			
MPRGP	5548	136.55	93.64%	95.11%	93.84%	93.84%
MPRGPp	886	21.410	93.75%	94.24%	94.79%	94.79%
GP-BoxVABB$_{\text{min}}$	–	253.49	93.89%	94.63%	94.67%	94.67%
			$C = 100$			
MPRGP	5913	147.16	93.64%	95.15%	94.24%	93.80%
MPRGPp	1247	30.854	93.81%	94.43%	94.71%	94.71%
GP-BoxVABB$_{\text{min}}$	–	255.17	93.89%	94.77%	94.54%	94.54%

Table 7: Dataset *phishing*, ℓ_2-loss SVM formulation.

	$\overline{n}$	time (s)	Acc.	Prec.	Sens.	F1
			$C = 1$			
MPRGP	2503	55.36	93.92%	94.67%	94.67%	94.57%
MPRGPp	1143	2.6696	93.61%	95.20%	93.72%	93.72%
GP-BoxVABB$_{min}$	257	5.3164	93.92%	95.01%	94.38%	94.38%
			$C = 5$			
MPRGP	3154	68.804	94.05%	94.82%	94.77%	94.77%
MPRGPp	232	6.065	93.64%	94.53%	94.35%	94.35%
GP-BoxVABB$_{min}$	443	5.3164	93.67%	94.00%	94.87%	94.87%
			$C = 10$			
MPRGP	3231	67.600	94.08%	94.82%	94.82%	94.82%
MPRGPp	213	3.7882	93.64%	94.63%	94.26%	94.26%
GP-BoxVABB$_{min}$	1461	26.3447	93.67%	94.43%	94.48%	94.48%
			$C = 50$			
MPRGP	3537	77.885	93.67%	94.39%	94.52%	94.52%
MPRGPp	272	5.4236	93.67%	94.53%	94.39%	94.39%
GP-BoxVABB$_{min}$	2226	51.098	94.00%	94.67%	94.81%	94.81%
			$C = 100$			
MPRGP	3625	67.429	93.89%	94.82%	94.50%	94.50%
MPRGPp	666	10.596	93.42%	94.15%	94.33%	94.33%
GP-BoxVABB$_{min}$	3993	62.435	93.97%	94.72%	94.72%	94.72%

where $w_q(x_1, x_2) = (1 + \epsilon \cos x_1)^3$, $w_l(x_1, x_2) = \epsilon \sin x_1$ for some constant $\epsilon \in (0, 1)$, and $\mathcal{D} = (0, 2\pi) \times (0, 2d)$, for some constant $d > 0$. The convex set K is defined as $K = \{v \in H_0^1(\mathcal{D}) : v \geqslant 0 \text{ on } \mathcal{D}\}$, where $H_0^1(\mathcal{D})$ is the Hilbert space of the functions with compact support on $\mathcal{D}$ such that v and $\|\nabla v\|^2$ belong to $L^2(\mathcal{D})$. The finite differences discretizetion of this problem lead to a quadratic programming problem of the form

$$\min_{v \in \Omega} \frac{1}{2} v^T A v - b^T v$$

where $v \in \mathbb{R}^{n_x n_y}$, $\Omega = \{v \in \mathbb{R}^{n_x n_y} : v_{i,j} \geqslant 0\}$ and $v_{i,j}$ is the value of v at the corresponding grid point of the discretization of the rectangle $\mathcal{D}$. For our tests we consider the journal bearing problem with $\epsilon = 0.1$ and $d = 10$ and different discretizations: $n_x = n_y = 50$; $n_x = n_y = 100$; $n_x = 200, n_y = 50$; $n_x = 400, n_y = 25$.

Table 8 shows the results obtained with the GP-BoxVABB$_{\text{min}}$ method ($M = 9$, $m_\alpha = 2$, $\tau_1 = 0.5$, $\vartheta = 1.1$), the MPRGP method described in Algorithm 3 and its variant MPRGPp. The methods share the same stopping criterion defined as in (2.23) with $\texttt{tol} = 10^{-7}$. For each method, the table reports the number $\overline{n}$ of matrix-vector products performed, the absolute value of the objective function at computed solution $|f(x^{(k)})|$, the number of active constraints at the computed solution, $na(x^{(k)})$ and the computational time needed to satisfy the stopping criterion. We can observe that the GP-BoxVABB$_{\text{min}}$ approach performs a lower number of matrix-vector products and employes less computational time to satisfy the stopping criterion.

The solutions returned by each method are compared in Table 9, where $\bar{x}_1, \bar{x}_2, \bar{x}_3$ denote, respectively, the computed solutions of MPRGP, MPRGPp and GP-BoxVABB$_{\text{min}}$.

Table 8: Journal bearing test problem.

	$\overline{n}$	$\|f(x^k)\|$	$na(x^{(k)})$	Time(s)
		$n_x = 50$, $n_y = 50$		
MPRGP	279	$1.804880e-01$	824	0.16950
MPRGPp	294	$1.804880e-01$	824	0.14008
GP-BoxVABB$_{min}$	165	$1.804880e-01$	824	0.07245
		$n_x = 100$, $n_y = 100$		
MPRGP	589	$1.805744e-01$	3232	1.11416
MPRGPp	589	$1.805744e-01$	3232	1.00365
GP-BoxVABB$_{min}$	314	$1.805744e-01$	3232	0.23941
		$n_x = 200$, $n_y = 50$		
MPRGP	1055	$1.802781e-01$	3214	2.11197
MPRGPp	1077	$1.802781e-01$	3214	1.57805
GP-BoxVABB$_{min}$	656	$1.802780e-01$	3214	0.44538
		$n_x = 400$, $n_y = 25$		
MPRGP	2089	$1.793250e-01$	3195	3.94401
MPRGPp	2314	$1.793250e-01$	3195	3.63034
GP-BoxVABB$_{min}$	872	$1.792793e-01$	3195	0.52482

Table 9: Journal bearing test problem.

$\|\bar{x}_1 - \bar{x}_3\|_\infty$	$\|\bar{x}_2 - \bar{x}_3\|_\infty$
$n_x = 50$, $n_y = 50$	
$5.5185e-05$	$5.5184e-05$
$n_x = 100$, $n_y = 100$	
$1.0543e-04$	$1.0543e-04$
$n_x = 200$, $n_y = 50$	
$7.0144e-04$	$7.0144e-04$
$n_x = 400$, $n_y = 25$	
$1.1147-03$	$1.1147-03$

4.2 NUMERICAL EXPERIMENTS ON RANDOM SLBQP PROBLEMS

This section is devoted to evaluating the numerical behaviour of the BB-based rules introduced in Chapter 3 within GP methods for solving SLBQP problems (3.3). The aim of the section is to provide practical confirmations of the theoretical results proved in Chapter 3. We recall that, in this case, the projection onto the feasible set Ω is computed by means of the secant-like algorithm developed in [28].

4.2.1 *Spectral inspection on SLBQP problems*

We start by analysing the distribution of the steplength sequences generated by the BB strategies and their modified versions with respect to the eigenvalues of the Hessian matrix. We randomly generated three SLBQP test problems of the form (3.3), where the distributions of the eigenvalues of the Hessian matrix A are defined as follows:

SLBQP1: for $i = 1, \dots, n$,

$$\lambda_i = \frac{(\underline{\lambda} b - \overline{\lambda} a)}{(b-a)} + \frac{(\underline{\lambda} - \overline{\lambda})}{(b-a)} \omega_i,$$

where $\underline{\lambda} = 1$, $\overline{\lambda} = 10^4$, $a = (1-c)^2$, $b = (1+c)^2$, $c = 1/2$ and the values ω_i are distributed according to the Marčenko-Pastur density [62] $p_c(x) = \dfrac{\sqrt{(b-a)(x-a)}}{2\pi x c^2}$, $a < x < b$;

SLBQP2: logarithmic distribution in $[1, 10^3]$ such that $\lambda_1 = 1$, $\lambda_n = 10^3$ and $\dfrac{\lambda_i}{\lambda_{i-1}}$ is constant, generated through the MATLAB function `logspace`;

SLBQP3: eigenvalues of the restricted Hessian matrix with logarithmic distribution in $[10, 10^2]$, generated through the MATLAB function `logspace`.

For each problem, the optimal solution x^* and the vector v are randomly chosen from a uniform distribution in $(-1, 1)$; the number n_a of active constraints at the solution is equal to $0.4 \cdot n$ and the feasible initial point $x^{(0)}$ is randomly generated such that its entries are inactive. Table 10 summarizes the spectral features of the problems.
We evaluate the behaviour of the GP method equipped with different steplength selection rules: BB1, BB2, EQ-BB2, ABB_{min}, EQ-ABB_{min} and EQ-VABB_{min}. The parameter setting for Algorithm 2 is the same of the previous section. The stopping criterion is defined as in(3.39) $\text{tol} = 10^{-7}$.
Figures 30-32-34 show the distribution of the sequences $\left\{\frac{1}{\alpha_k}\right\}$ generated by each rule with respect to the eigenvalues of the matrices $\tilde{U}_{k-1}^T A \tilde{U}_{k-1}$, during the iterative procedure. In each panel, at the k-th iteration, we plotted 20 eigenvalues (black dots) of the restricted

Table 10: Some features of the SLBQP problems.

	$\lambda_{min}(A)$	$\lambda_{max}(A)$	$\lambda_{min}(A_{\mathcal{J}*,\mathcal{J}*})$	$\lambda_{max}(A_{\mathcal{J}*,\mathcal{J}*})$
SLBQP1	19	9923	39.21	9865.62
SLBQP2	1	1000	3.34	722.20
SLBQP3	0.05	1396.3	10	100

Hessian matrix $\tilde{U}_{k-1}^T A \tilde{U}_{k-1}$ with linearly spaced indices (including the maximum and minimum eigenvalues), and the reciprocal of the steplength α_k (red cross); the blue lines correspond to maximum and minimum eigenvalues of the Hessian matrix A and the blue circles on the right denote 20 eigenvalues of the restricted Hessian matrix at the prefixed solution x^*, with linearly spaced indices (including the maximum and minimum eigenvalues). These plots confirm the ability of the reciprocals of BB1 and EQ-BB2 steplengths to sweep the spectrum of the restricted Hessian matrices, as predicted by Theorem 3.1; the reciprocals of the steplengths obtained by applying the original BB2 scheme can sometimes fall outside the spectrum of the restricted Hessian matrices, and, in general the rule provides unsatisfying results, which reflect also in the behaviour of the alternating scheme ABB_{min}. On the contrary, the use of the EQ-BB2 rule within alternating schemes helps to enhance the performances of the method both in terms of spectral behaviour and efficiency.

Figures 31-33-35 report the decrease of the relative error on the solution, $\frac{\|x^{(k)}-x^*\|}{\|x^*\|}$, and the function error, $|f(x^{(k)}) - f^*|$. We may observe that the decrease of both the errors is considerably accelerated by employing the EQ-BB2 strategy instead of BB2. Furthermore, the GP method equipped with the alternating rules EQ-ABB_{min} and EQ-$VABB_{min}$ preserves the ability to outperform the single steplength rule, as observed in the unconstrained case, thanks to the approach that takes into accounts the presence of the feasible region. In particular, the algorithm that adopts the variable scheme EQ-$VABB_{min}$ shows higher efficiency than the other schemes.

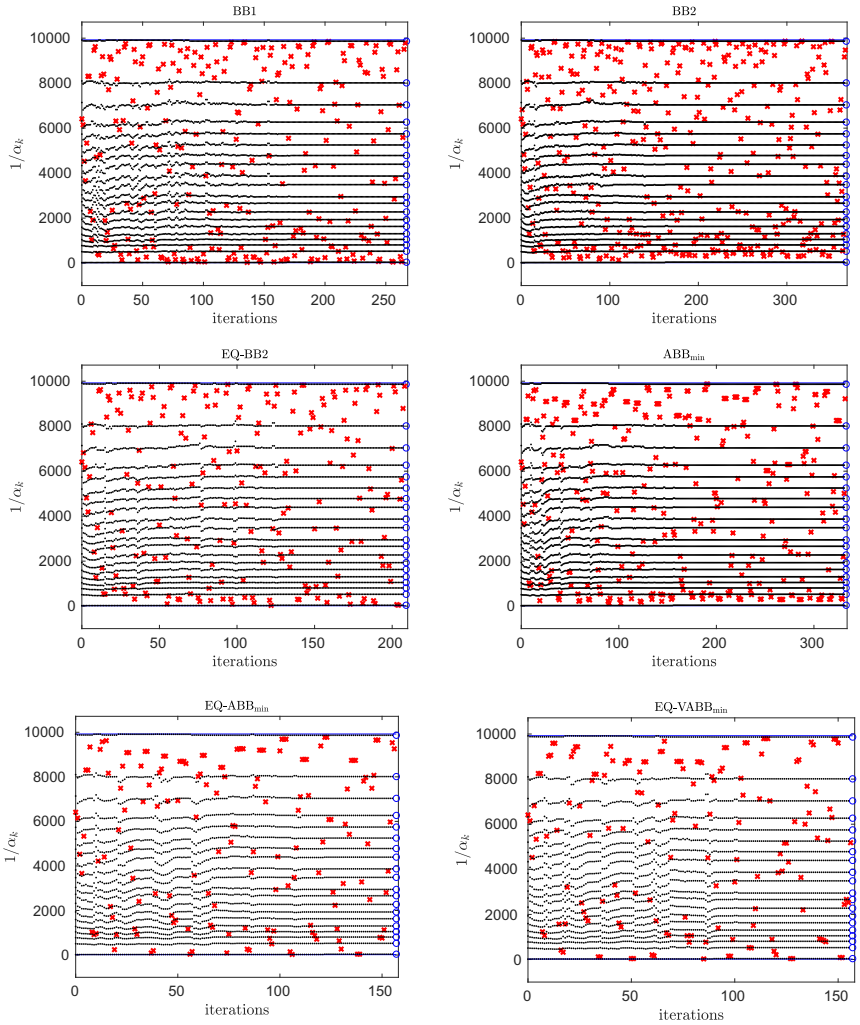

Figure 30: Behaviour of GP equipped with BB-based steplength rules on SLBQP1. Distribution of $\left\{\frac{1}{\alpha_k}\right\}$ with respect to the iterations for BB1, BB2 (top panels), EQ-BB2, ABB$_{min}$ (middle panels), EQ-ABB$_{min}$ and EQ-VABB$_{min}$ (bottom panels).

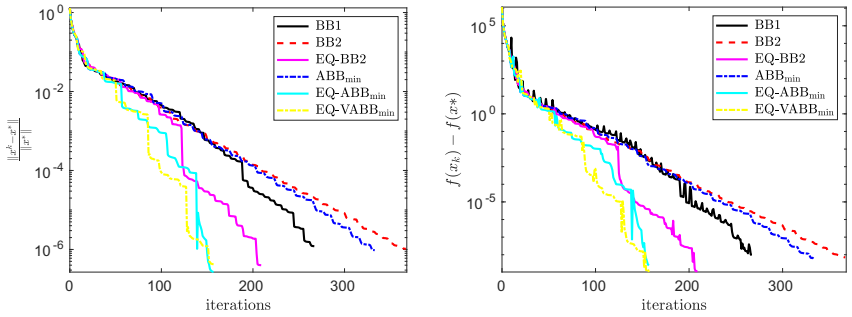

Figure 31: Test problem SLBQP1. History of relative error on the solution (top-left panel) and function error (top-right panel).

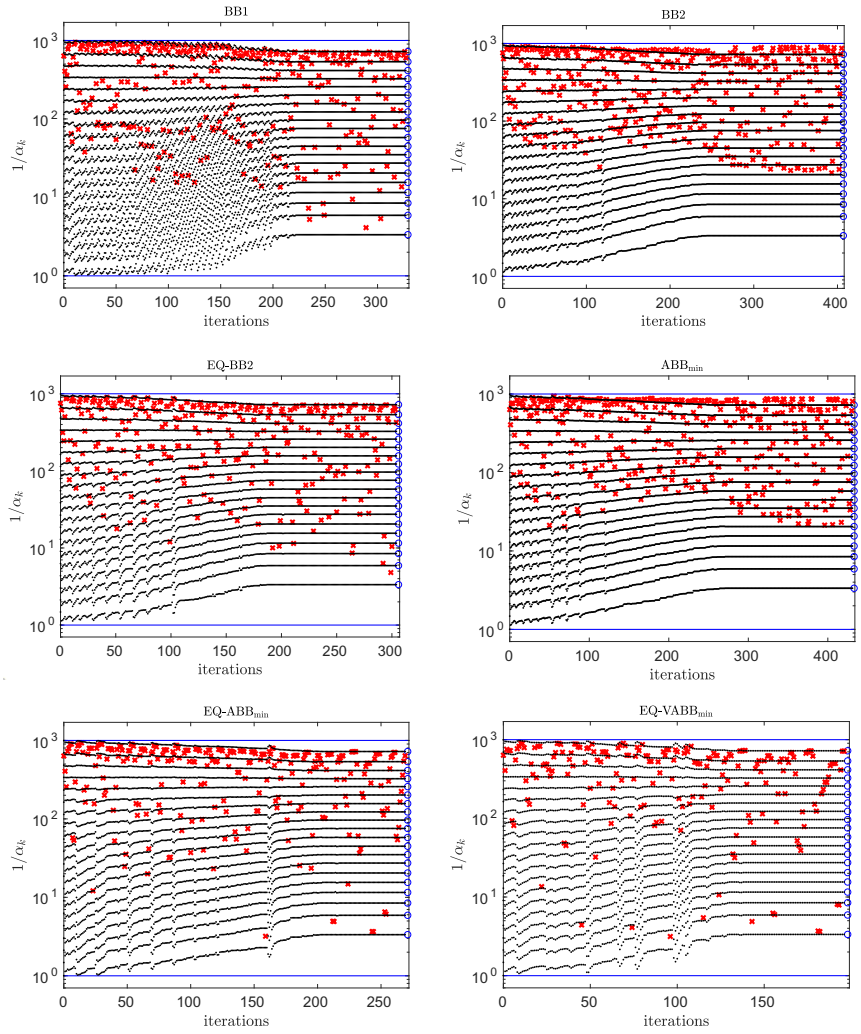

Figure 32: Behaviour of GP equipped with BB-based steplength rules on SLBQP2. Distribution of $\left\{\frac{1}{\alpha_k}\right\}$ with respect to the iterations for BB1, BB2 (top panels), EQ-BB2, ABB$_{min}$ (middle panels), EQ-ABB$_{min}$ and EQ-VABB$_{min}$ (bottom panels).

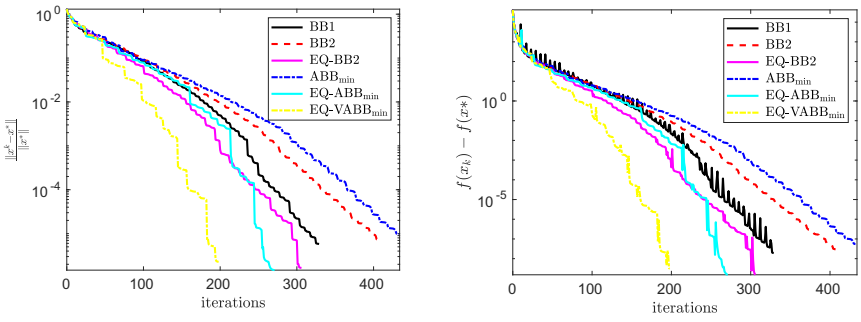

Figure 33: Test problem SLBQP2. History of relative error on the solution (top-left panel) and function error (top-right panel).

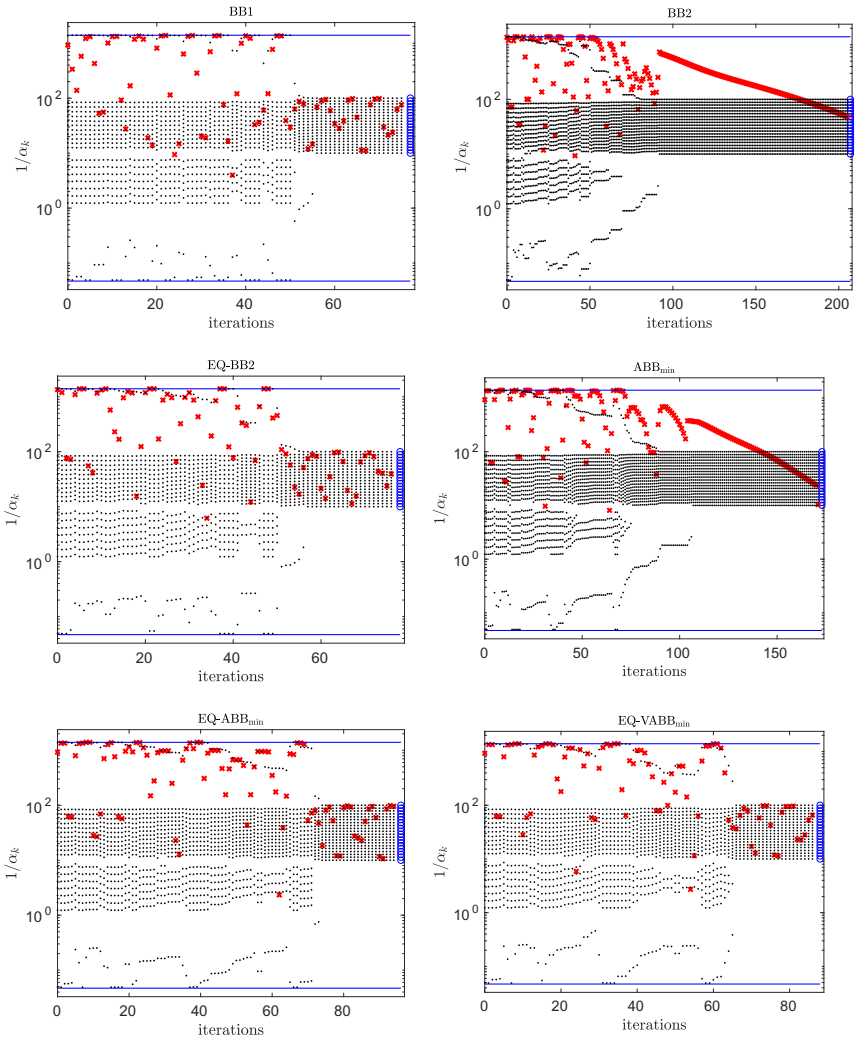

Figure 34: Behaviour of GP equipped with BB-based steplength rules on SLBQP3. Distribution of $\left\{\frac{1}{\alpha_k}\right\}$ with respect to the iterations for BB1, BB2 (top panels), EQ-BB2, ABB$_{min}$ (middle panels), EQ-ABB$_{min}$ and EQ-VABB$_{min}$ (bottom panels).

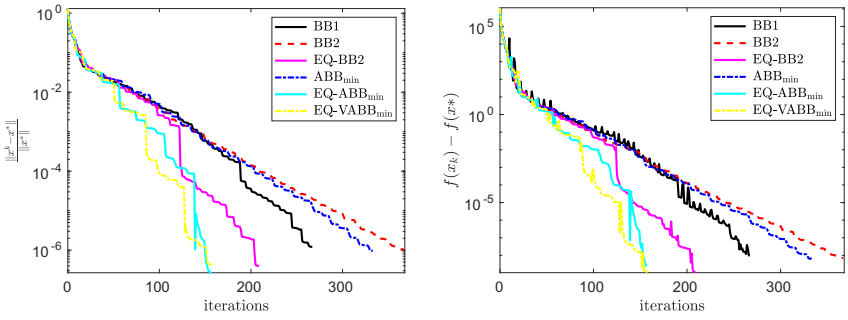

Figure 35: Test problem SLBQP3. History of relative error on the solution (top-left panel) and function error (top-right panel).

4.2.2 *Numerical results on random large-scale SLBQP problems*

We report the numerical results obtained on a set of large-scale SLBQP problems with non-degenerate solution, generated as in Section 4.1.2, with the same parameter setting used in DATASET 1. In particular, by means of performance profile plots, we compare the computational time required by the GP method combined with the following steplength updating rules: BB1, BB2, EQ-BB2, EQ-ABB$_{\min}$ and EQ-VABB$_{\min}$.

From Figure 36 we may observe that the EQ-BB2 outperforms the standard BB2 rule; on the other hand, only the alternating schemes are able to solve all the problems within the maximum number of 40000 iterations, while the single rule seems to suffer on this dataset, especially the BB1 rule. As shown in Figure 37, the results seem from the number of active variables at the solution, whereas the level of degeneracy weakly affect the general behaviour of the considered schemes (see Figure 38).

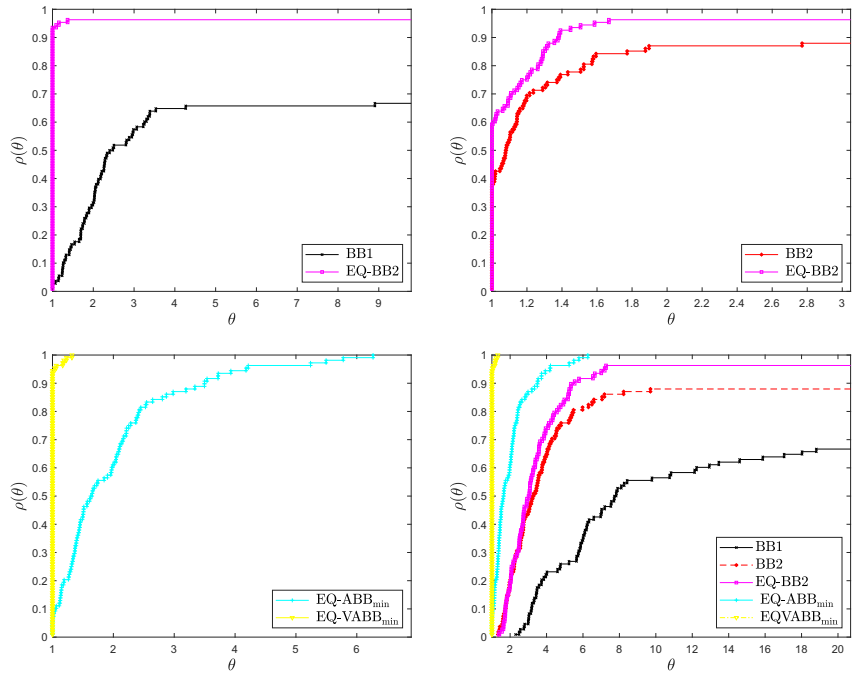

Figure 36: Runtime performance profiles of GP method equipped with BB-based rules on a set of SLBQP test problems with non-degenerate solutions.

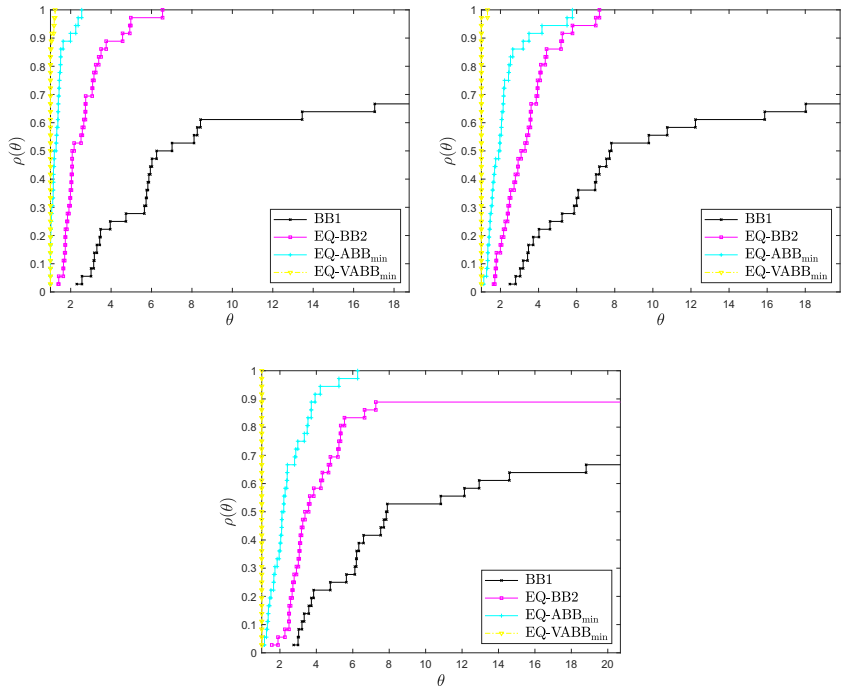

Figure 37: Runtime performance profiles of the GP method equipped with BB-based rules on a set of SLBQP test problems with non-degenerate solutions. Fraction of active constraints at the solution: $0.1 \cdot n$ (top-left panel), $0.5 \cdot n$ (top-right panel), $0.9 \cdot n$ (bottom panel).

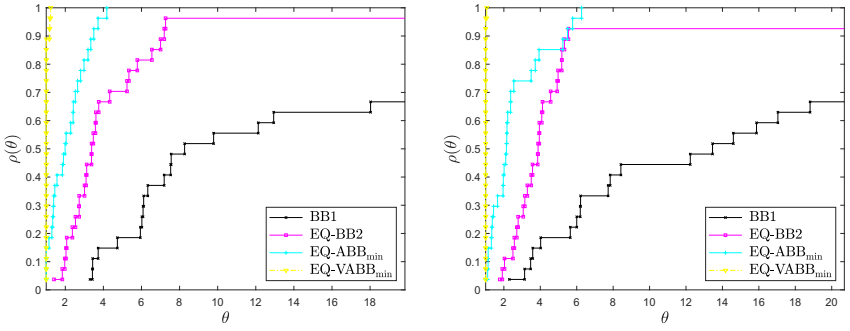

Figure 38: Runtime performance profiles of the GP method equipped with BB-based rules on the subsets of SLBQP test problems corresponding to the parameters $ndeg = 7$ (left panel) and $ndeg = 10$ (right panel).

4.2.3 *Application in Support Vector Machines*

For these experiments we consider a problem of binary data classification formulated as follows.
Let $D = \{(z_i, y_i),\ \ i = 1, \dots, n,\ \ z_i \in \mathbb{R}^m,\ \ y_i \in \{-1, 1\}\}$ be a training set of labeled examples, the SVM algorithm performs classification of new examples $z \in \mathbb{R}^m$ by using a decision function $F : \mathbb{R}^m \longrightarrow \{-1, 1\}$ of the form

$$F(z) = \operatorname{sign}\left(\sum_{i=1}^{n} x_i^* y_i K(z, z_i) + b^*\right), \tag{4.4}$$

where $K : \mathbb{R}^m \times \mathbb{R}^m \longrightarrow \mathbb{R}$ denotes a kernel function and the vector $x^* = (x_1^*, \dots, x_n^?)^T$ is the solution of

$$\begin{aligned} &\min f(x) = \frac{1}{2} x^T A x - \sum_{i=1}^{n} x_i \\ &\text{subject to } \ 0 \leqslant x \leqslant Ce, \quad \sum_{i=1}^{n} y_i x_i = 0, \end{aligned} \tag{4.5}$$

where $e = (1, 1, \dots, 1)^T \in \mathbb{R}^n$ and $C > 0$. Once the vector x^* is computed, the quantity b^* in (4.4) is easily derived. The Hessian matrix A has entries $A_{ij} = y_i y_j K(z_i, z_j),\ \ i, j = 1, 2, \dots, n$. For our test problems we consider a Gaussian kernel, namely $K(z_i, z_j) = e^{-\frac{\|z_i - z_j\|_2^2}{2\sigma^2}}$, with $\sigma \in \mathbb{R}$.
The dataset for our tests from the repository LIBSVM, available at https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/. In particular, the following test problems are considered:

MNIST1000 $n = 1000$, $C = 10$, $\sigma = 1800$, $\operatorname{rank}(A) = 1000$;

MNIST2000 $n = 2000$, $C = 10$, $\sigma = 1800$, $\operatorname{rank}(A) = 2000$;

ADU $n = 1000$, $C = 1$, $\sigma = \sqrt{10}$, $\operatorname{rank}(A) = 985$;

WEB $n = 1000$, $C = 5$, $\sigma = \sqrt{10}$, $\operatorname{rank}(A) = 736$.

We recall that MNIST is the well-known dataset of handwritten digits that is commonly used for training various image processing systems, ADU dataset is used for training a machine to determine whether a person has an annual income greater than 50.000 dollars, and WEB dataset is related to web spam detection.

We compare the behaviour of the GP method combined with the steplength rules BB1 , BB2, VABB_{min}, EQ-BB2, $\text{EQ-VABB}_{\}}$ on the described problems. For the alternating schemes the following parameter setting is used: $m_\alpha = 2, \tau_1 = 0.7$, and $\vartheta = 1.3$. The GP methods share a nonmonotone linsearch procedure ($M = 9$) and are stopped when either the relative distance between two successive iterations is

lower than 10^{-8} or 1000 iterations have been performed. The initial point for all the considered schemes is the null vector.
Table 11 shows the number of iterations and the computational time needed by each scheme to satisfy

$$\frac{|f(x^{(k)}) - f^*|}{|f^*|} \leqslant tol \tag{4.6}$$

where f^* is the minimum of the objective function values obtained by the different methods at the end of the iterative process. The symbol "−"denotes that condition (4.6) has not been satisfied within the maximum number of 1000 iterations.
In Figure 39, we can appreciate the decrease of the relative function error defined in (4.6), with respect to the computational time, for the four datasets. The results reported in Table 11 and Figure 39 confirm the effectiveness of the modified BB2 selection rule compared with those obtained with the original rule, also within the alternating scheme EQ-VABB$_{min}$. Indeed, the benefits gained by employing EQ-BB2 in place of BB2 are clear in terms of both number of iterations and computational time.

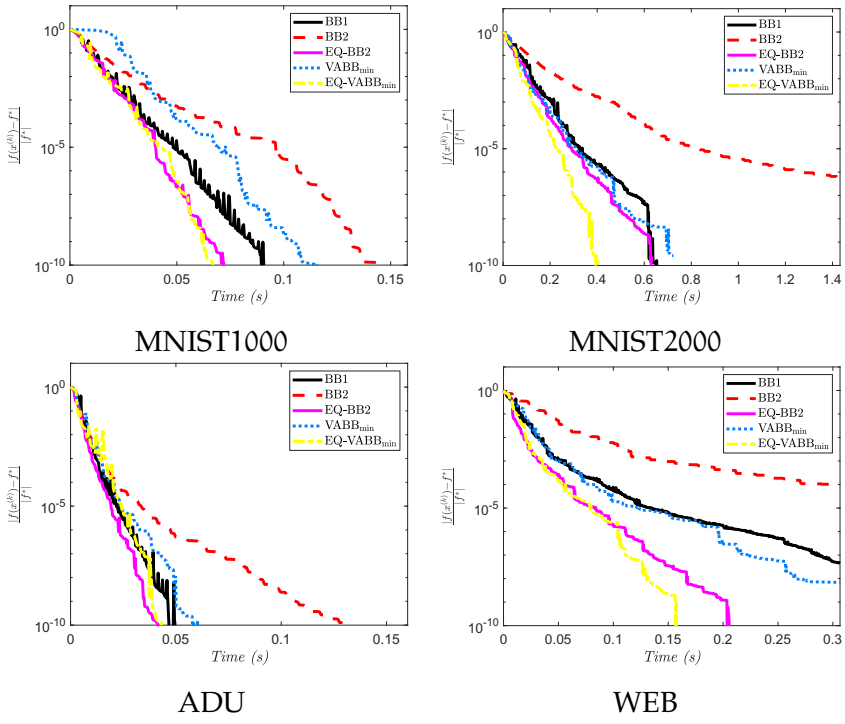

Figure 39: Plots of the relative difference (4.6) with respect to the computational time for the four SVM test problems

Table 11: SVM test problems. Number of iterations required by each algorithm satisfy (4.6) for different tolerances. The corresponding computational time (averaged over 20 runs) is also reported.

	tol = 10^{-2}		tol = 10^{-4}		tol = 10^{-6}		tol = 10^{-8}	
	It.	Time	It.	Time	It.	Time	It.	Time
	MNIST1000							
BB1	474	0.0194	91	0.0365	143	0.0570	185	0.0735
BB2	91	0.0298	219	0.0703	333	0.1068	400	0.1282
EQ-BB2	41	0.0169	88	0.0348	114	0.0447	154	0.0604
$VABB_{min}$	48	0.0363	92	0.0556	145	0.0789	175	0.0921
EQ-$VABB_{min}$	36	0.0155	81	0.0328	121	0.0483	142	0.0567
	MNIST2000							
BB1	77	0.1190	166	0.2571	286	0.4467	392	0.6167
BB2	175	0.2437	427	0.6060	890	1.2740	–	–
EQ-BB2	78	0.1101	158	0.2255	251	0.3588	376	0.5379
$VABB_{min}$	71	0.1068	162	0.2476	270	0.4144	373	0.5713
EQ-$VABB_{min}$	57	0.0825	130	0.1895	189	0.2754	252	0.3658
	ADU							
BB1	23	0.0080	45	0.0158	79	0.0277	113	0.0399
BB2	30	0.0091	73	0.0220	158	0.0475	299	0.0897
EQ-BB2	22	0.0070	47	0.0150	72	0.0228	104	0.0329
$VABB_{min}$	29	0.0089	67	0.0203	109	0.0328	164	0.0492
EQ-$VABB_{min}$	23	0.0077	60	0.0203	85	0.0286	111	0.0374
	WEB							
BB1	84	0.0315	243	0.0902	599	0.2214	952	0.3523
BB2	281	0.0867	977	0.2994	–	–	–	–
EQ-BB2	59	0.0187	175	0.0553	349	0.1111	523	0.1670
$VABB_{min}$	103	0.0325	260	0.0818	620	0.1958	847	0.2668
EQ-$VABB_{min}$	60	0.0205	163	0.0549	314	0.1043	393	0.1304

4.2.4 *Application in reconstruction of fiber orientation distribution in diffusion MRI*

We consider the problem of intra-voxel reconstruction of the fiber orientation distribution function (FOD) in each voxel of white matter of brain from diffusion MRI data. In [57] the authors clarify that the diffusion signal can be represented as the convolution of a response function with the FOD function and, as a consequence, the estimation of the intra-voxel structure can be shaped through a linear model of the form

$$b = \Phi x + \eta, \tag{4.7}$$

where $x \in \mathbb{R}^n$ represents the FOD function, $b \in \mathbb{R}^d$ is the vector of measurements, Φ is the linear measurement operator, and η is the acquisition noise. Since problem (4.7) is ill-posed, the solution is approximated by means of a reweighted ℓ_1-minimization process which involves, at each step, the solution of a convex problem of the form

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} f(x) \equiv \|\Phi x - b\|_2^2 \\ &\text{subject to } \; x \geqslant 0, \quad \|Wx\|_1 = K, \end{aligned} \tag{4.8}$$

where $W \in \mathbb{R}^{n\times n}$ is a diagonal matrix with positive entries and K is the estimated maximum number of fibres to be detected in the brain volume [3]. The weighted ℓ_1-norm constraint induces sparsity on the solution, and the weighting matrix W forces some anatomical properties of the fiber bundles in neighboring voxels. A complete overview about the properties of W can be found in [3].

It has been shown in [20] and [13] that the presence of a variable metric in first-order methods can significantly improve the performance in solving problem (4.8) with respect to their standard non-scaled versions. We report the results obtained by using different steplength selection strategy within the SGP scheme (1.51) We consider the P-BB1 and P-BB2 rule defined in (3.46) and (3.48) respectively, the modified version P-EQ-BB2 defined in (3.47), and the alternating strategies P-EQ-VABB$_{\min}$ (3.49) and P-VABB$_{\min}$, which can be obtained from (3.49) by using the P-BB2 rule in place of P-EQ-BB2. The values $m_\alpha = 2$, $\tau_1 = 0.5$ and $\vartheta = 3$ are used in P-EQ-VABB$_{\min}$ and P-VABB$_{\min}$. The sequence $\{D_k\}_{k\in\mathbb{N}}$ is selected by mimicking the split gradient-based scaling proposed in [6] for quadratic problems: the scaling matrix has the following form

$$D_k = \operatorname{diag}\left(\max\left(\frac{1}{\mu_k}, \min\left(\mu_k, \frac{x^{(k)}}{\Phi^T\Phi x^{(k)}}\right)\right)\right)^{-1}, \tag{4.9}$$

where $\mu_k = \sqrt{1 + \dfrac{10^{11}}{(k+1)^2}}$. The parameter μ_k forces the sequence $\{D_k\}_{k\in\mathbb{N}}$ to asymptotically approach the identity matrix [21, Lemma

2.3]. This condition ensures the convergence of the sequence of the iterates generated by the SGP scheme to a solution of the minimization problem, as proved in [14, Theorem 3.1].

For the numerical tests, we employed the Phantom dataset, available at https://github.com/basp-group/co-dmri, and described in [3]. In particular, the test problem has $d = 19200$, $n = 257280$, and $K = 3840$. Table 12 summarizes, for each scheme, the number of iterations and the computing time needed to guarantee

$$\frac{|f(x^{(k)}) - f^*|}{|f^*|} \leqslant \mathtt{tol}. \tag{4.10}$$

the distance (4.6) is below certain thresholds tol. Symbol "−"denotes that condition (4.10) has not been satisfied within the maximum number of 4000 iterations. Figure 40 shows the relative function error (4.10) between the objective function values provided by the different methods and the minimum computed value f^*.

By analysing the results in Table 12 and Figure 40, we can affirm that the modified version of the BB2 rule allows to largely improve the behaviour of the SGP algorithm in terms of number of iterations and computational time compared to the performance obtained with the standard BB2 strategy. Similar considerations hold also for the alternating schemes: the use of EQ-BB2 in place of BB2 makes P-EQ-VABB$_{\text{min}}$ more effective than P-VABB$_{\text{min}}$ in finding the solution of the optimization problem; furthermore, this test confirms that alternating strategies still outperform the result obtained using a single steplength rule.

Table 12: Number of iterations and computational time required by SGP equipped with different steplength rules satisfy condition 4.10 for different tolerance values.

	tol = 10^{-3}		tol = 10^{-5}		tol = 10^{-8}		tol = 10^{-10}	
	It.	Time	It.	Time	It.	Time	It.	Time
P-BB1	263	2.7	1025	10.6	3541	38.7	–	–
P-BB2	620	6.4	–	–	–	–	–	–
P-EQ-BB2	282	3.1	830	9.4	2552	29.9	–	–
P-VABB$_{\text{min}}$	283	2.9	3175	36.0	–	–	–	–
P-EQ-VABB$_{\text{min}}$	271	3.1	683	8.1	2169	26.5	3034	37.7

4.3 BEYOND THE QUADRATIC CASE

In this section we investigate the practical efficiency of the proposed steplength selection rules within GP methods for constrained minimization of general non-quadratic objective functions.

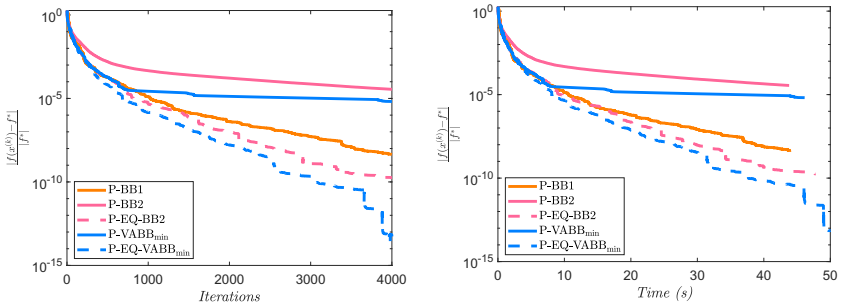

Figure 40: Test problem (4.8): plots of the relative function error (4.10) with respect to the number of iterations (first panel) and the computational time (second panel) achieved by the SGP methods.

4.3.1 *Numerical results on general box-constrained problems*

We start by analysing the effect of the different steplength selection rules in accelerating the GP method on general box-constrained test problems and we compare the results by means of performance profile plots introduced in Section 4.1.
The test problems are generated as follows. Starting from an unconstrained minimization problem of a twice continuously differentiable objective function $\Psi(x)$, for which a local minimum point x^* is known, we used the technique proposed in [40] to generate a box-constrained problem of the form

$$\min_{\ell \leqslant x \leqslant u} f(x) = \Psi(x) + \sum_{i \in \mathcal{A}^*_\ell} h_i(x_i) - \sum_{i \in \mathcal{A}^*_u} h_i(x_i) \tag{4.11}$$

where $h_i : \mathbb{R} \to \mathbb{R}$, $i \in \mathcal{A}^*$, are twice continuously differentiable non-decreasing functions. Due to the special definition of $f(x)$, the point x^* is a solution of (4.11).

For our tests, we selected some well-known non-quadratic functions $\Psi(x)$, described below.

TRIGONOMETRIC FUNCTION [46] :

$$\Psi(x) = \|b - (Av(x) + Bu(x))\|^2,$$

where $v(x) = (\sin(x_1), ..., \sin(x_n))^T$, $u(x) = (\cos(x_1), ..., \cos(x_n))^T$, and A and B are square matrices of order $n = 500$ with entries generated as random integers in $(-100, 100)$. Given a vector $x^* \in \mathbb{R}^n$ with entries randomly generated from a uniform distribution in $(-\pi, \pi)$, the vector b is defined such that $\Psi(x^*) = 0$. The starting vector is set as $x^{(0)} = \min(u, (\max(\ell, x^* + 0.3\, r)))$, where $r \in \mathbb{R}^n$ has random entries from a uniform distribution in $[-\pi, \pi]$.

CHAINED ROSENBROCK FUNCTION [81] :

$$\Psi(x) = \sum_{i=2}^{n} (4\varphi_i(x_{i-1} - x_i^2)^2 - (1 - x_i)^2),$$

where $n = 500$, the values φ_i, $i = 1, \ldots, 50$, are defined as in [81, Table 1] and $\varphi_{i+50j} = \varphi_i$, $i = 1, ..., 50$, $j = 1, \ldots, 9$. In this case, a solution of the unconstrained problem is $x^* = (1, 1, ..., 1)^T$. The starting vector is set as $x^{(0)} = \min(u, (\max(\ell, x^* + 0.8\, r)))$, where $r \in \mathbb{R}^n$ has random entries from a uniform distribution in $[-1, 1]$.

LAPLACE2 FUNCTION [44] :

$$\Psi(x) = \frac{1}{2} x^T A x - b^T x + \frac{1}{4} h^2 \sum_i x_i^4,$$

where A is a square matrix of order $n = N^3$, $N = 100$, arising from the discretization of a 3D Laplacian on the unit box by a standard seven-point finite difference formula, $h = \dfrac{1}{N+1}$ and b is chosen such that $x_i^* \equiv x(kh, rh, sh)$ with

$$x(kh, rh, sh) = h^3 krs(kh - 1)(rh - 1)(sh - 1)e^{-\frac{1}{2}((kh - d_1)^2 + (rh - d_2)^2 + (sh - d_3)^2)},$$

where the index i is associated with the mesh point (kh, rh, sh), $k, r, s = 1, \ldots, N$. Two different settings for the parameters d, d_1, d_2 and d_3 are considered:

a) $d = 20$, $d_1 = d_2 = d_3 = 0.5$,

b) $d = 50$, $d_1 = 0.4$, $d_2 = 0.7$, $d_3 = 0.5$.

In both cases, the starting vector is $x^{(0)} = \frac{\ell + u}{2}$.

For each function $\Psi(x)$, we built the corresponding constrained versions with the following choices for the functions $h_i(x)$, as suggested in [40]:

(1) $\beta_i \left(x_i - x_i^*\right)$,

(2) $\alpha_i \left(x_i - x_i^*\right)^3 + \beta_i \left(x_i - x_i^*\right)$,

(3) $\alpha_i \left(x_i - x_i^*\right)^{7/3} + \beta_i \left(x_i - x_i^*\right)$,

where α_i are random numbers in $(0.001, 0.011)$, $\beta_i = 10^{-\eta_i ndeg}$, η_i are random number in $(0, 1)$, and $ndeg = 1, 4, 10$; in particular, the values β_i correspond to the Lagrangian multipliers associated to the active constraints and, therefore, the values assigned to $ndeg$ allow to control the degeneracy of the problem at x^*. The vectors ℓ and u are set in order to have a number of active constraints at the solution equal to a prefixed value na; the same number of lower and upper active constraints is considered and different problems are generated by

setting $n_a \approx 0.1 \cdot n, 0.5 \cdot n, 0.9 \cdot n$. In this way, we built a total amount of 108 box-constrained non-quadratic test problems.

Figure 41-43 show the performance profiles obtained by solving the non-quadratic problems by the GP method with different steplength rules. All the parameters involved in the GP algorithm and in the steplength rules are set as in the quadratic case, except for the initial steplength $\alpha_0 = 1$. The maximum number of iteration is set equal to 4000. As regard the limited memory approaches we report the results obtained for $m = 5$, since they gave the best performances in the quadratic case. From Figure 41 we may observe that the BoxBB2 rule confirms its better behaviour compared with the profile of the standard BB2 selection. Furthermore, the steplength strategies based on adaptive alternation of BB1 and BoxBB2 still provide the best performance. Since the proposed selections are designed essentially for achieving an effective approximation of the spectrum of the reduced Hessian and, consequently, speed up the method, we may conclude that the spectral properties exhibited by the modified BB-based steplength rules play a crucial role for improving the gradient methods also in case of general box-constrained non-quadratic optimization problems.

Figure 42 report the results obtained with LMGP, BoxLMGP1, and BoxLMGP2 for $m = 5$: from these plots we can affirm that the general behaviour observed in the quadratic case is confirmed. Finally, Figure 43 shows a summarizing plot that compares the performance profile obtained with both BB-based rules and limited memory approaches: on the considered test problems, the GP method combined with the $\text{BoxVABB}_{\text{min}}$ scheme turns out to be the best solvers, followed by the BoxBB2 and the BoxLMGP1 strategies.

4.3.2 *Numerical results on general SLB problems*

In this section we report the result obtained in the solution of non-quadratic SLB test problems, generated with a similar approach of that described in [40].

Starting from an unconstrained minimization problem of a twice continuously differentiable objective function $\Psi(x)$ with known solution x^*, we built SLB constrained test problems of two possible formulations:

$$\min_{x \in \Omega} f(x) = \Psi(x) + v^T(x - x^*) + \sum_{i \in \mathcal{A}_\ell^*} h_i(x_i) - \sum_{i \in \mathcal{A}_u^*} h_i(x_i) \tag{4.12}$$

$$\Omega = \{x \in \mathbb{R}^n : \ell \leqslant x \leqslant u, \quad v^T x = e\},$$

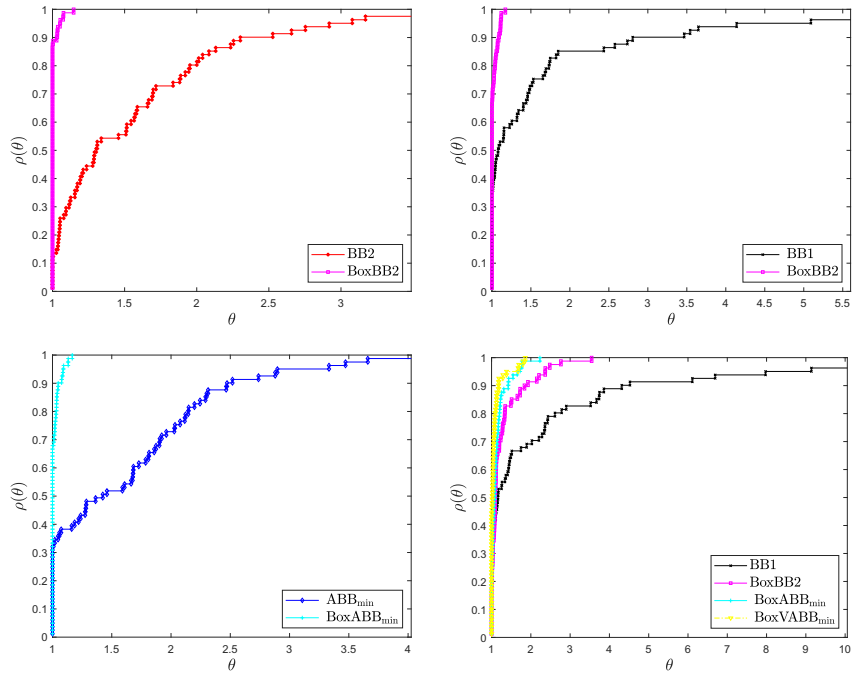

Figure 41: Runtime performance profiles obtained by the GP method equipped with different BB-based steplength rules on a set of box-constrained non-quadratic test problems.

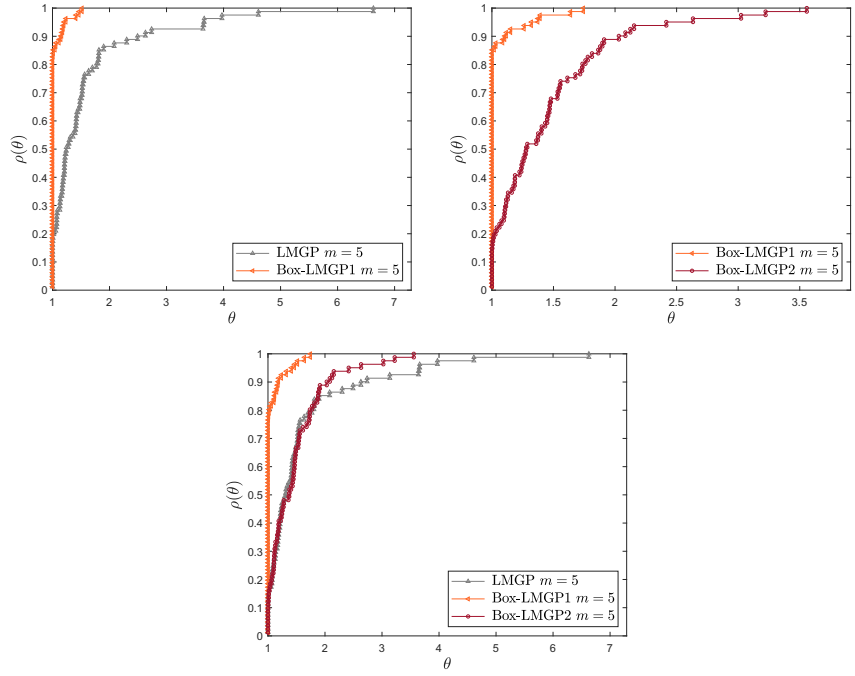

Figure 42: Runtime performance profiles obtained by different limited memory approaches on a set of box-constrained non-quadratic test problems.

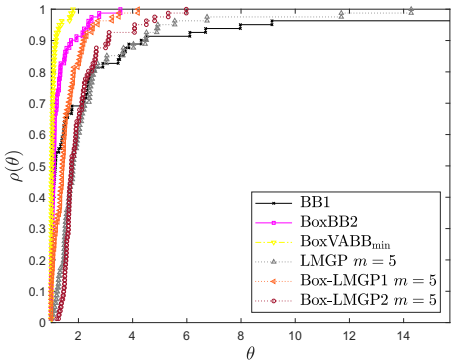

Figure 43: Runtime performance profiles obtained by by the GP method equipped with different steplength rules on a set of box-constrained non-quadratic test problems.

$$\min_{x\in\Omega} f(x) = \Psi(x)v^T x + \sum_{i\in\mathcal{A}^*_\ell} h_i(x_i) - \sum_{i\in\mathcal{A}^*_u} h_i(x_i)$$
$$\Omega = \{x \in \mathbb{R}^n : \ell \leqslant x \leqslant u, \quad v^T x = e,\ e > 0\}, \tag{4.13}$$

where $v \in \mathbb{R}^n$, $e \in \mathbb{R}$, and $h_i : \mathbb{R} \to \mathbb{R}$, $i \in \mathcal{A}^*$, are twice continuously differentiable non-decreasing functions. The constrained problems defined by (4.12) and (4.13) have the same solution x^* of the unconstrained problem of minimizing $\Psi(x)$; we notice that the scalar e in the second formulation (4.13) must be positive to ensure this property. For our experiments, we selected the same non-quadratic functions described in the previous subsection. To build the corresponding constrained versions of Trigonometric and Chained Rosenbrock test problems we used both the formulations (4.12) and (4.13), while for Laplace2 function we used the latter form only. The vector v is randomly generated from a uniform distribution in $(0, 1)$ and the following choices for the functions $h_i(x)$ are considered, as in the previous case:

(1) $\beta_i \left(x_i - x_i^*\right)$,

(2) $\alpha_i \left(x_i - x_i^*\right)^3 + \beta_i \left(x_i - x_i^*\right)$,

(3) $\alpha_i \left(x_i - x_i^*\right)^{7/3} + \beta_i \left(x_i - x_i^*\right)$,

where α_i are random numbers in $(0.001, 0.011)$ and $\beta_i = 10^{-\eta_i ndeg}$, with η_i random numbers in $(0, 1)$ and $ndeg = 1, 4, 10$. In order to retain first-order optimality conditions, the Lagrangian multiplier of the single linear equality constraint must be equal to 1 up to sign (for the case (4.12)) or to $\Psi(x^*)$ up to sign (for the case (4.13)), while the Lagrangian multipliers associated to the active constraints are easily assigned equal to the values β_i, and, therefore, the parameter $ndeg$

allows to control the degeneracy of the problem at x^*. The vectors ℓ and u are defined in order to have the number of active constraints at the solution equal to a prefixed value n_a; in particular, we set $n_a \approx 0.1 \cdot n, 0.5 \cdot n, 0.9 \cdot n$ and the same number of lower and upper active constraints at x^*. Finally, the resulting dataset is composed of 162 non-quadratic SLB test problems.

We evaluated the performance obtained by running the GP method equipped with the steplengths rules: BB1, BB2, ABB_{min}, VABB_{min}, EQ-BB2, EQ-ABB_{min}, EQ-VABB_{min}. The considered schemes share the following parameter setting: $\alpha_0 = 1$, $\tau = 0.7$ and $m_a = 2$ for defining $\alpha_k^{\text{ABB}_{\text{min}}}$ and $\alpha_k^{\text{EQ-ABB}_{\text{min}}}$ and $\tau_1 = 0.7$, $m_a = 2$ and $\zeta = 1.3$ for $\alpha_k^{\text{EQ-VABB}_{\text{min}}}$. The starting vectors are defined as follows:

- $x^{(0)} = \Pi_\Omega\,(x^* + 0.3\,r)$, where $r \in \mathbb{R}^n$ has random entries from a uniform distribution in $[-\pi, \pi]$, when $\Psi(x)$ is the Trigonometric function;
- $x^{(0)} = \Pi_\Omega(x^* + 0.8\,r)$, where $r \in \mathbb{R}^n$ has random entries from a uniform distribution in $[-1, 1]$, when $\Psi(x)$ is the Chained Rosenbrock function;
- $x^{(0)} = \Pi_\Omega\left(\frac{\ell+u}{2}\right)$, when $\Psi(x)$ is the Laplace2 function.

The stopping criterion adopted is defined as in (3.39). The performance profiles shown in Figure 44 confirm the results obtained in the quadratic framework. By capturing the information about the active set at each iteration, the EQ-BB2 steplength strategy allows to speed up the GP method with respect to the case the original BB2 rule is used. This enhancement is further emphasized by using the alternating strategies employing the modified rule EQ-BB2.

4.3.3 *Application in image deblurring with Poisson noise*

In order to evaluate the behaviour of the proposed steplength rules on a non-quadratic problem arising from real-applications, we consider the problem of recovering a blurred image corrupted by Poisson noise. In general, in a Bayesian framework, an approximation of the original object can be obtained by solving a constrained problem where the objective function is given by the sum of a discrepancy function (typically depending on the noise type affecting the data) and a regularization term adding *a priori* information; simple constraints, expressing physical requirements, can be considered. In the case of Poisson noise, the discrepancy function measuring the distance from the data b is the generalized Kullback-Leibler (KL) divergence, having the form

$$f_0(Ax + c; b) = \sum_{i=1}^{n} b_i \log \frac{b_i}{(Ax+c)_i} + (Ax + c)_i - b_i, \tag{4.14}$$

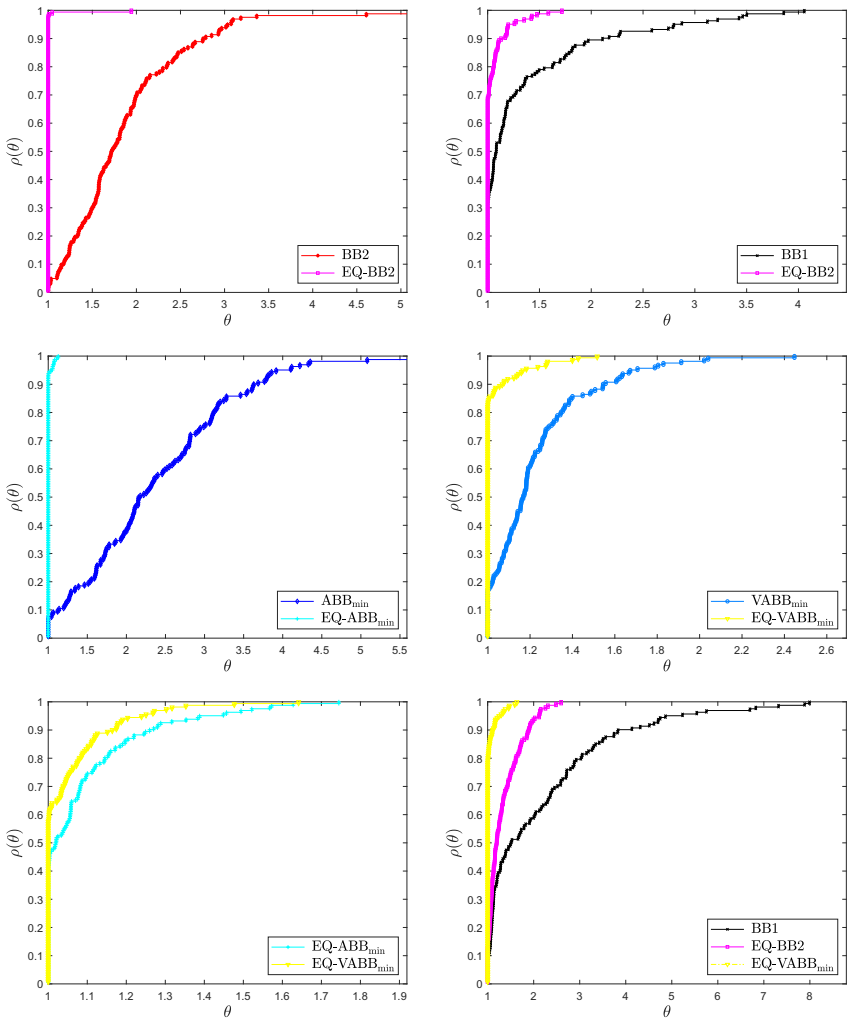

Figure 44: Runtime performance profiles obtained by the GP method equipped with different steplength rules on a set of non-quadratic SLB test problems.

where $A \in \mathcal{M}_n(\mathbb{R})$ is a linear operator modelling the distortion due to the image acquisition system and $c \in \mathbb{R}^n$ is a known positive background radiation constant. A typical assumption for the matrix A is that it has nonnegative elements and each row and column has at least one positive entry (see [8] for details about the image deblurring problem in presence of Poisson noise). A widely used edge-preserving regularizer is the discrete smooth Total Variation functional, also known as Hypersurface regularizer, which is defined, for an image of $n = N \times N$ pixels, as

$$f_1(x) = \sum_{k,\ell} \sqrt{(x_{k+1,\ell} - x_{k,\ell})^2 + (x_{k,\ell+1} - x_{k,\ell})^2 + \gamma^2} - \gamma, \quad (4.15)$$

where γ is a small positive constant and periodic boundary conditions are assumed. A *maximum a posteriori* estimate of the original image is a solution of the following nonlinear programming problem

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} f(x) \equiv f_0(Ax + c; b) + \rho f_1(x) \\ &\qquad \text{subject to } \; x \geqslant 0, \quad \sum_{i=1}^{n} x_i = K, \end{aligned} \quad (4.16)$$

where $K = \sum_{i=1}^{n} b_i - n \cdot c$ is the flux of the image and ρ is a positive parameter balancing the role of the regularization term and the discrepancy function; the inequality constraints and the single linear equality constraint express the non-negativity of the pixels and the conservation of the image flux, respectively.
We consider as test problem a 512×512 object representing a microtubulin network inside a cell [70]. In this case, the values of the original object $\bar{x}$ are in the range $[0, 686]$, whereas those of the blurred and noisy image b are in $[0, 446]$; the background was set equal to 1 and the relative distance between the original object and the blurred noisy data in Euclidean norm is 0.756; furthermore, $\rho = 4 \cdot 10^{-4}$ and $\gamma = 10^{-6} \cdot \max_i\{b_i\}$. A ground-truth solution x^*, i.e., an estimate of the real minimum point of the problem (4.16), is obtained by executing a huge number of iterations of the SGP method. Indeed, it is well known that the above problem can be efficiently addressed by the gradient projection method equipped with a variable metric (see for example [15, 19, 60]). Mimicking the split gradient-based scaling,the sequence of scaling matrices $\{D_k\}_{k \in \mathbb{N}}$ can be selected as follows

$$B_k = \operatorname{diag}\left(\max\left(\frac{1}{\mu_k}, \min\left(\mu_k, \frac{x^{(k)}}{A^T \mathbf{1} + \rho V(x^{(k)})}\right)\right)\right)^{-1}, \quad (4.17)$$

where $\mathbf{1}$ is a vector with all entries equal to 1, $V(x^{(k)})$ is the positive part of the splitting of $\nabla f_1(x) = V(x) - U(x)$ at $x^{(k)}$ (see [8, Cap. 5]) and $\mu_k = \sqrt{1 + \dfrac{10^{11}}{(k+1)^2}}$.

In Table 13 and Figure 45 we report the behaviour of the SGP method combined with the steplength rules P-BB1, P-BB2, P-EQ-BB2, P-VABB$_{\text{min}}$ and P-EQ-VABB$_{\text{min}}$. For the alternating rules, the following setting of parameters is considered: $m_a = 2$, $\tau_1 = 0.5$ and $\vartheta = 3$. Table 13 shows the number of iterations and the computational time, in seconds, required by the considered methods to satisfy

$$\frac{|f(x^{(k)}) - f^*|}{|f^*|} \leqslant \mathtt{tol}, \tag{4.18}$$

where $f^* \equiv f(x^*)$. We also report the relative reconstruction error (RRE) $\frac{\|x^{(\mathrm{It.})} - \overline{x}\|}{\|\overline{x}\|}$ at the iteration It. If one of the approaches is not able to reduce the relative error on the objective function below a prefixed tolerance within 1000 iterations, Table 13 displays the computational time spent and the RRE achieved after the 1000 iterations performed, and the corresponding results are indicated by a star. Figure 45 shows the relative error of the objective function and the relative minimization error $\frac{\|x^{(k)} - x^*\|}{\|x^*\|}$ with respect to the number of iterations and the computational time.

We observe that P-BB2 and P-EQ-BB2 rules show the same behaviour, which is very similar to that of P-BB1, as well; indeed, for this problem, the variable metric (in particular the term $x^{(k)}$ in (4.17)) hides the effects of the rules that take into account the constraints and, at the same time, the equality constraint plays a minor role, since the assumptions on the matrix A already induce the iterates to satisfy the flux constraint.

Nevertheless, when the alternating rule is adopted, the use of P-EQ-BB2 can still improve the performance of SGP, achieving in 33 s. (565 iterations) with P-EQ-VABB$_{\text{min}}$ the value of the objective function obtained with P-VABB$_{\text{min}}$ after 56 s. (1000 iterations).

Figure 46 shows the relative error of the objective function and the relative minimization error obtained in absence of a variable metric, with the same parameter setting used for SGP. The general effect of the scaling matrix is clear by comparing these results with those reported in Figure 45: in particular, we can observe that the scaled version of the methods are able to significantly reduce both the errors with respect to the standard GP schemes; furthermore, as already remarked, its influence overlaps with the effect of the modification in the P-EQ-BB2 rule.

Table 13: Image deblurring test problem: number of iterations and computational time (average over 10 runs) required to reduce the relative error on the objective function below a prefixed tolerance. The corresponding relative reconstruction error achieved is reported.

	$\mathtt{tol} = 5 \cdot 10^{-2}$			$\mathtt{tol} = 10^{-3}$			$\mathtt{tol} = 5 \cdot 10^{-4}$		
	It.	Time	RRE	It.	Time	RRE	It.	Time	RRE
P-BB1	64	3.8	0.568	933	48	0.447	1000*	52*	0.447*
P-BB2	28	2.1	0.531	1000*	53*	0.446*	1000*	54*	0.446*
P-EQ-BB2	27	2.0	0.533	1000*	56*	0.446*	1000*	56*	0.447*
P-VABB$_{min}$	26	1.9	0.537	471	25	0.438	1000*	55*	0.438*
P-EQ-VABB$_{min}$	37	3.0	0.531	267	15	0.438	565	33	0.438

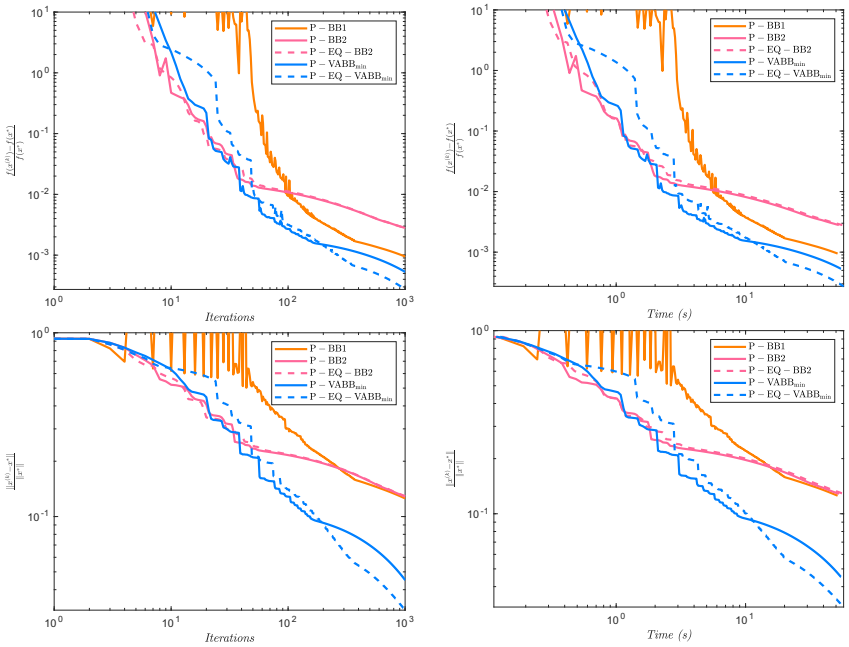

Figure 45: Image deblurring test problem: numerical results of 1000 iterations of the SGP method combined with different steplength rules; first row: relative error of the objective function with respect to iterations (left panel) and computational time (right panel); second row: relative minimization error $\frac{\|x^{(k)}-x^*\|}{\|x^*\|}$ with respect to iterations (left panel) and computational time (right panel).

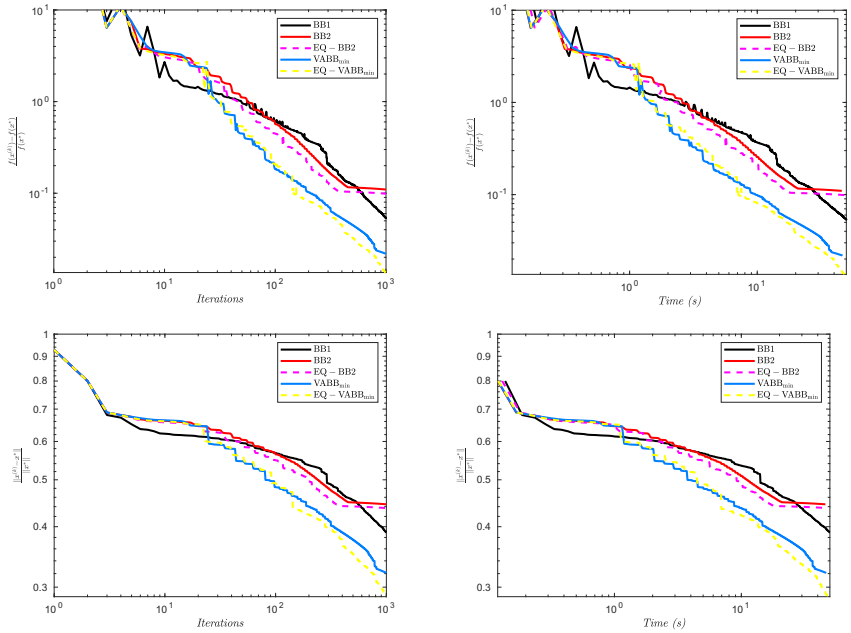

Figure 46: Image deblurring test problem: numerical results of 1000 iterations of the GP method combined with different steplength rules; first row: relative error of the objective function with respect to iterations (left panel) and computational time (right panel); second row: relative minimization error $\frac{\|x^{(k)}-x^*\|}{\|x^*\|}$ with respect to iterations (left panel) and computational time (right panel).

CONCLUSIONS

In this thesis we presented a spectral analysis of the Barzilai-Borwein and the LMSD steplength rules in gradient projection methods for solving special constrained QP problems. We proposed suitable modifications to the original strategies, in order to exploit the information prescribed by the KKT optimality conditions. Concerning the BB rules, we proved that BB1 rule is able to capture the spectral properties of special submatrices of the Hessian matrix, which depend only on the inactive variables at each iteration; in accordance with this analysis, we proposed novel versions of BB2 rule with the same features of BB1; these rules seem very useful in improving the effectiveness of the method and, consequently, can be efficiently exploited within state-of-the-art adaptive alternation steplength strategies.

With regard to the limited memory method, we introduced an ad hoc strategy that allows to adaptively set the length of the sweep during the iterative procedure. An extensive numerical experimentation confirmed the benefits of the proposed rules, also for solving constrained problems deriving from real-world applications.

The carried out analysis opens the way to further investigations. Indeed, optimization problems subject to more complex feasible regions, such as, e. g., an n-simplex or a polyhedron, may be addressed using techniques similar to those presented in this thesis.

Furthermore, another interesting topic for future works consists in the spectral analysis of the LMSD procedure when it is combined with a GP method exploiting a linesearch on the feasible arc. This last procedure is interesting, although generally more expensive, because it allows to identify the set of active constraints in a finite number of iterations.

These open issues could be considered beyond the class of gradient projection methods: the behaviour of the novel BB-rules may be analysed in the context of forward-backward splitting methods where the non-smooth part of the objective function contains the indicator function of the non-negative orthant or the n-simplex.

A further new challenge is how these techniques could be exploited in the context of stochastic gradient methods for the big data scenario. This is a widely investigated current research area. Indeed, the numerical resolution of the current machine learning problems requires to obtain in a limited time, not asymptotically, the maximum decrease of the objective function. The performance of current stochastic methods may therefore benefit from the development of tools similar to the BB rules used as steplengths or learning rates.

A

APPENDIX

This appendix is devoted to recall some basic concepts and definitions; furthermore, we report here the proofs of lemmas 3.1-3.2-3.3

A.1 SOME DEFINITIONS AND GENERAL RESULTS

Definition. For any symmetric matrix $Q \in \mathcal{M}_n(\mathbb{R})$, the *Rayleigh quotient* of Q at a non zero vector $x \in \mathbb{R}^n$ is defined as the value

$$R_Q(x) = \frac{x^T Q x}{\|x\|^2},$$

that minimize the quantity $\|(Q - \lambda I)x\|$ with respect to λ. Clearly, if x is an approximate eigenvector, then $R_Q(x)$ is a reasonable estimate of the corresponding eigenvalue [49].

Definition. (Rates of convergence) Let $\{x^{(k)}\}_{k\in\mathbb{N}} \subseteq \mathbb{R}^n$ be a sequence that converges to x^*. The convergence is said *Q-linear* if there is a constant $r \in (0,1)$ such that

$$\frac{\|x^{(k+1)} - x^*\|}{\|x^{(k)} - x^*\|} \leqslant r,$$

for all k sufficiently large. The convergence is said *Q-superlinear* if

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

In general, we say that the Q-order of convergence is p ($p > 1$) if there is a positive constant M such that

$$\frac{\|x^{(k+1)} - x^*\|}{\|x^{(k)} - x^*\|^p} \leqslant M,$$

for all k sufficiently large.
The convergence is R-linear if there is a sequence of nonnegative scalars $\{v_k\}_{k\in\mathbb{N}}$ such that

$$\|x^{(k)} - x^*\| \leqslant v_k$$

for all k and $\{v_k\}_{k\in\mathbb{N}}$ converges Q-linearly to zero.

A.2 GENERAL RESULTS ON PROJECTION MATRICES

Let $x \in \mathbb{R}^n = \mathcal{V} \oplus \mathcal{W}$. Then, x can be uniquely decomposed as

$$x = x_1 + x_2, \qquad x_1 \in \mathcal{V},\ x_2 \in \mathcal{W}.$$

The projection onto $\mathcal{V}$ (along $\mathcal{W}$) is the linear transformation that maps x into x_1; the matrix that represents this transformation is called projection matrix.

Let P denote the projection matrix onto $\mathcal{V}$ along $\mathcal{W}$.
Proof of Lemma 3.1.
(Necessary condition) For any $z \in \mathbb{R}^n = \mathcal{V} \oplus \mathcal{W}$, $Pz = x \in \mathcal{V}$. Then, we have

$$P(Pz) = Px = x = Pz \Rightarrow P^2 z = Pz \Rightarrow P^2 = P.$$

(Sufficient condition) Let now suppose that the matrix P represents a linear transformation such that $P^2 = P$. We first observe that

$$\ker P = \text{range}(I_n - P),$$

indeed, let $x \in \ker P$, $x = (I_n - P)x$ then $\ker P \subseteq \text{range}(I_n - P)$; on the other hand, for any $z \in \mathbb{R}^n$, $P(I_n - P)z = Pz - P^2 z = 0_{n,1}$ and $\text{range}(I_n - P) \subseteq \ker P$. Furthermore, we have

$$z = z - Pz + Pz = (I_n - P)z + Pz,$$

hence, any vector can be decomposed as the sum of $x_1 = Pz \in \text{range}\, P$ and $x_2 = (I_n - P)z \in \ker P$. Since $\text{range}\, P \cap \ker P = O_{n,1}$, it follows that P represents a projection onto $\mathcal{V} = \text{range}\, P$ along $\mathcal{W} = \ker P$.

Proof of Lemma 3.2.
(Necessary condition) From Lemma 3.1 it follows $P^2 = P$.
Let $x, y \in \mathbb{R}^n$, $x = x_1 + x_2$ and $y = y_1 + y_2$, where $x_1, y_1 \in \mathcal{V}$ and $x_2, y_2 \in \mathcal{V}^\perp$; let recall that $x_2^T P y = 0$ and $y_2^T P x = 0$, then

$$x^T P y = (Px + x_2)^T P y = (Px)^T P y = (Px)^T (y - y_2) = x^T P^T y,$$

which implies that $P^T = P$.
(Sufficient condition) Let $x = Pz \in \text{range}\, P$, then $Px = P^2 z = Pz = x$; let $y \in \text{range}(P)^\perp$; for any x, we have $(Px)^T y = x^T P^T y = x^T P y = 0$, which implies that $Py = 0$. Therefore, from Lemma 3.1 P is a projection matrix onto $\text{range}\, P$ along $\text{range}(P)^\perp$. Finally, let λ be an eigenvalue of P; then for some $x \neq 0$, $Px = \lambda x$. We have

$$\lambda x = Px = P^2 x = \lambda P x = \lambda^2 x,$$

which implies that $\lambda = 0$ or $\lambda = 1$.

BIBLIOGRAPHY

[1] H. Akaike. "On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method". In: *Ann. Inst. Stat. Math. Tokyo* 11 (1959), pp. 1–16.

[2] L. Antonelli, V. De Simone, and D. di Serafino. "On the Application of the Spectral Projected Gradient Method in Image Segmentation". In: *J. Math. Imaging Vis.* 54.1 (2015), pp. 106–116.

[3] A. Auria, A. Daducci, J.-P. Thiran, and Y. Wiaux. "Structured sparsity for spatially coherent fibre orientation estimation in diffusion MRI". In: *NeuroImage* 115 (2015), pp. 245–255.

[4] Brett M. Averick, Richard G. Carter, and Jorge J. Moré. *The Minpack-2 Test Problem Collection*. 1991.

[5] J. Barzilai and J. M. Borwein. "Two-point step size gradient methods". In: *IMA J. Numer. Anal.* 8 (1988), pp. 141–148.

[6] F. Benvenuto, R. Zanella, L. Zanni, and M. Bertero. "Nonnegative least-squares image deblurring: improved gradient projection approaches". In: *Inverse Probl.* 26 (2010), p. 025004.

[7] M. Bertero and P. Boccacci. *Introduction to Inverse Problems in Imaging*. Institute of Physics Pub., 1998.

[8] M. Bertero, P. Boccacci, and V. Ruggiero. *Inverse Imaging with Poisson Data*. 2053-2563. IOP Publishing, 2018. ISBN: 978-0-7503-1437-4.

[9] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, 1999.

[10] E. G. Birgin, J. M. Martínez, and M. Raydan. "Nonmonotone spectral projected gradient methods on convex sets". In: *SIAM J. Optim.* 10 (2000), pp. 1196–1211.

[11] E. G. Birgin, J. M. Martínez, and M. Raydan. "Inexact spectral projected gradient methods on convex sets". In: *IMA J. Numer. Anal.* 23 (2003), pp. 539–559.

[12] E. G. Birgin, J. M. Martínez, and M. Raydan. "Spectral Projected Gradient Methods: Review and Perspectives". In: *J. Stat. Soft.* 60.3 (2014), pp. 1–21.

[13] S. Bonettini, F. Porta, and V. Ruggiero. "A variable metric forward–backward method with extrapolation". In: *SIAM J. Sci. Comput.* 38.4 (2016), A2558–A2584.

[14] S. Bonettini and M. Prato. "New convergence results for the scaled gradient projection method". In: *Inverse Problems* 31.9 (2015).

[15] S. Bonettini, R. Zanella, and L. Zanni. "A scaled gradient projection method for constrained image deblurring". In: *Inverse Probl.* 25.1 (2009), p. 015002.

[16] C. J. C. Burges. "A tutorial on support vector machines for pattern recognition". In: *Data Min. Knowl. Discov.* 2.2 (1998), pp. 121–167.

[17] P. H. Calamai and J. J. Moré. "Projected gradient methods for linearly constrained problems". In: *Mathematical Programming* 39 (1987), pp. 93–116.

[18] A. Cauchy. "Méthodes générales pour la résolution des systèmes d'équations simultanées". In: *CR. Acad. Sci. Par.* 25 (1847), pp. 536–538.

[19] E. Chouzenoux, J.-C. Pesquet, and A. Repetti. "Variable metric forward–backward algorithm for minimizing the sum of a differentiable function and a convex function". In: *J. Optim. Theory Appl.* 162.1 (2014), pp. 107–132.

[20] V. L. Coli, V. Ruggiero, and L. Zanni. "Scaled first-order methods for a class of large-scale constrained least squares problems". In: *AIP Conference Proceedings 1776*. 2016, p. 040002.

[21] P. L. Combettes and B. C. Vũ. "Variable metric quasi-Féjer monotonicity". In: *Nonlinear Anal.* 78 (2013), pp. 17–31.

[22] C. Cortes and V. N. Vapnik. "Support vector network". In: *Mach. Learn.* 20 (1995), pp. 1–25.

[23] F. E. Curtis and W. Guo. *R-Linear Convergence of Limited Memory Steepest Descent*. Tech. rep. arXiv:1610.03831. 2016.

[24] Y. H. Dai. "On the Nonmonotone Line Search". In: *J. Optim. Theory Appl.* 112 (2002), pp. 315–330.

[25] Y. H. Dai. "Alternate step gradient method". In: *Optimization* 53 (4-5 2003), pp. 395–415.

[26] Y. H. Dai and R. Fletcher. "On the asymptotic behaviour of some new gradient methods". In: *Math. Programming* 103 (2005), pp. 541–559.

[27] Y. H. Dai and R. Fletcher. "Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming". In: *Numer. Math.* 100 (2005), pp. 21–47.

[28] Y. H. Dai and R. Fletcher. "New algorithms for singly linearly constrained quadratic programming problems subject to lower and upper bounds". In: *Math. Programming* 106 (3 2006), pp. 403–421.

[29] Y. H. Dai and L. Z. Liao. "R-linear convergence of the Barzilai and Borwein gradient method". In: *IMA J Numer Anal* 22.1 (2002), pp. 1–10.

[30] Y. H. Dai and Y. Yuan. "Alternate minimization gradient method". In: *IMA J. Numer. Anal.* 23 (3 2003), pp. 377–393.

[31] Y. H. Dai and Y. Yuan. "Analyses of Monotone gradient methods". In: *J. Ind. Manag. Optim.* 1 (2 2005), pp. 181–192.

[32] Y. H. Dai and H. Zhang. "Adaptive two-point stepsize gradient algorithm". In: *Numerical Algorithms* 27 (2001), pp. 377–385.

[33] Yu-Hong Dai, Yakui Huang, and Xin-Wei Liu. "A Family of Spectral Gradient Methods for Optimization". In: *Comput. Optim. Appl.* 74.1 (2019), pp. 43–65.

[34] R. De Asmundis, D. di Serafino, F. Riccio, and G. Toraldo. "On spectral properties of steepest descent methods". In: *IMA J. Numer. Anal.* 33 (4 2013), pp. 1416–1435.

[35] R. De Asmundis, D. di Serafino, W.W. Hager, G. Toraldo, and H. Zhang. "An efficient gradient method using the Yuan steplength". In: *Comput. Optim. Appl.* 59.3 (2014), pp. 541–563.

[36] E. D. Dolan and J. J. Moré. "Benchmarking Optimization Software with Performance Profiles". In: *Mathematical Programming* 91.2 (2002), pp. 201–213.

[37] Elizabeth D. Dolan and Jorge J. Moré. "Benchmarking optimization software with COPS." In: 2001.

[38] Zdenek Dostál. *Box constrained quadratic programming with proportioning and projections*. Vol. 7. 1997, 871–887.

[39] Zdenek Dostál. *Optimal quadratic programming algorithms: with applications to variational inequalities*. Vol. 23. Springer Science & Business Media, 2009.

[40] F. Facchinei, J. Judice, and J. Soares. "Generating box-constrained optimization problems". In: *ACM Trans. Math. Softw.* 23.3 (1997), pp. 443–447.

[41] M. A. Figueiredo, R. D. Nowak, and S. J. Wright. "Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems". In: *IEEE Journal of Selected Topics in Signal Processing* 1 (2007), pp. 586–597.

[42] R. Fletcher. *Practical Methods of Optimization*. 2nd edn. Wiley, Chichester, 1987.

[43] R. Fletcher. "Low storage methods for unconstrained optimization". In: *Lectures in Applied Mathematics* 26 (1990), pp. 165–179.

[44] R. Fletcher. "On the Barzilai-Borwein Method". In: *Optimization and Control with Applications*. Vol. 96. Applied Optimization. Springer, US, 2005, pp. 235–256.

[45] R. Fletcher. "A limited memory steepest descent method". In: *Math. Program., Ser. A* 135 (2012), pp. 413–436.

[46] R. Fletcher and M. J. D. Powell. "A rapidly convergent descent method for minimization". In: *Comput. J.* 6 (1963), pp. 163–168.

[47] G. Frassoldati, L. Zanni, and G. Zanghirati. "New adaptive stepsize selections in gradient methods". In: *J. Ind. Manag. Optim.* 4.2 (2008), pp. 299–312.

[48] A. Friedlander, J. M. Martínez, B. Molina, and M. Raydan. "Gradient Method with Retards and Generalizations". In: *SIAM J. Numer. Anal.* 36 (1 1999), pp. 275–289. ISSN: 0036-1429.

[49] G. H. Golub and C. F. Van Loan. *Matrix Computations, 3rd edn.* Applied Optimization. Baltimore and London: John Hopkins University Press, 1996.

[50] C. C. Gonzaga. "On the worst case performance of the steepest descent algorithm for quadratic functions". In: *Math. Program., Ser. A* 160 (2016), pp. 307–320.

[51] C.C. Gonzaga and R. M. Schneider. "On the steepest descent algorithm for quadratic functions". In: *Comput. Optim. Appl.* 63.2 (2016), pp. 523–542.

[52] L. Grippo, F. Lampariello, and S. Lucidi. "A nonmonotone line search technique for Newton's method". In: *SIAM J. Numer. Anal.* 23 (1986), pp. 707–716.

[53] L. Grippo and M. Sciandrone. "Nonmonotone globalization techniques for the Barzilai-Borwein gradient method". In: *Comput. Optim. Appl.* 23 (2002), pp. 143–169.

[54] William W. Hager and Hongchao Zhang. "An affine scaling method for optimization problems with polyhedral constraints". In: *Comput. Optim. Appl.* 59 (2014), pp. 163–183.

[55] R. H. Horn and C. R. Johnson. *Matrix Analysis*. 2nd. Vol. 33. New York, NY, USA: Cambridge University Press, 2012.

[56] A.N. Iusem. "On the convergence properties of the projected gradient method for convex optimization". In: *Computational and Applied Mathematics* 22.1 (2003), pp. 37–52.

[57] B. Jian and B. Vermuri. "A unified computational framework for deconvolution to reconstruct multiple fibers from diffusion weighted MRI". In: *IEEE Trans. Med. Imag.* 26 (2007), pp. 1464–1471.

[58] Z.Dostál L. Pospíšil. "The projected Barzilai-Borwein method with fall-back for strictly convex QCQP problems with separable constraints". In: *Math. Comp. Simul.* 145 (2018), pp. 79–89.

[59] C. Lanczos. "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators". In: *J. Res. Nat. Bur. Stand.* 45 (1950), pp. 255–282.

[60] H. Lantéri, M. Roche, and C. Aime. "Penalized maximum likelihood image restoration with positivity constraints: multiplicative algorithms". In: *Inverse Problems* 18 (2002), pp. 1397–1419.

[61] I. Loris, M. Bertero, C. De Mol, R. Zanella, and L. Zanni. "Accelerating Gradient Projection Methods for l1-Constrained Signal Recovery by Steplength Selection Rules". In: *Applied and Computational Harmonic Analysis* 27 (2009), pp. 247–154.

[62] V. A. Marčenko and L. A. Pastur. "Distribution of eigenvalues for some sets of random matrices". In: *Mathematics of the USSSR - Sbornik* 1.4 (1967), pp. 457–483.

[63] J. J. Morè and G. Toraldo. "Algorithms for bound constrained quadratic programming problems". In: *Numerische Mathematik* 55.4 (1989), pp. 377–400.

[64] Jorge J. Moré and Gerardo Toraldo. "On the Solution of Large Quadratic Programming Problems with Bound Constraints". In: *SIAM Journal on Optimization* 1 (1991), pp. 93–113.

[65] J. Nocedal and S. J. Wright, eds. *Numerical Optimization*. Springer–Verlag, New York, Inc., 1999.

[66] Parlett B.N. van derVorst H Paige C.C. "Approximate solutions and eigenvalue bounds from Krylov subspaces." In: *Numer. Linear Algebra Appl.* 2 (1995), pp. 115–133.

[67] P. M. Pardalos and Rosen J. B. *Constrained global optimization: algorithms and applications*. New York, NY, USA: Springer-Verlag, 1987.

[68] E. Loli Piccolomini, V. L. Coli, E. Morotti, and Luca Zanni. "Reconstruction of 3D X-ray CT images from reduced sampling by a scaled gradient projection algorithm". In: *Comp. Opt. and Appl.* 71.1 (2018), pp. 171–191.

[69] F. Porta, M. Prato, and L. Zanni. "A new steplength selection for scaled gradient methods with application to image deblurring". In: *J. Sci. Comp.* 65 (2015), pp. 895–919.

[70] F. Porta, R. Zanella, G. Zanghirati, and L. Zanni. "Limited-memory scaled gradient projection methods for real-time image deconvolution in microscopy". In: *Commun. Nonlinear Sci. Numer. Simul.* 21 (2015), pp. 112–127.

[71] M. Prato, R. Cavicchioli, L. Zanni, P. Boccacci, and M. Bertero. "Efficient deconvolution methods for astronomical imaging: algorithms and IDL-GPU codes". In: *Astron. Astrophys.* 539 (2012), A133–(11pp).

[72] L. Pronzato and A. Zhigljavsky. "Gradient algorithms for quadratic optimization with fast convergence rates". In: *Comput. Optim. Appl.* 50 (2011), pp. 597–617.

[73] L. Pronzato, A. Zhigljavsky, and E. Bukina. "An asymptotically optimal gradient algorithm for quadratic optimization with low computational cost". In: *Optimization Letters* 7.6 (2013), pp. 1047–1059.

[74] L. Pronzato, A. Zhigljavsky, and E. Bukina. "Estimation of Spectral Bounds in Gradient Algorithms". In: *Acta Appl. Math.* 127 (2013), pp. 117–136.

[75] M. Raydan. "On the Barzilai and Borwein choice of steplength for the gradient method". In: *IMA J. Numer. Anal.* 13 (1993), pp. 321–326.

[76] M. Raydan. "The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem". In: *SIAM J. Optim.* 7.1 (1997), pp. 26–33.

[77] M. Raydan and B. F. Svaiter. "Relaxed Steepest Descent and Cauchy-Barzilai-Borwein Method". In: *Comput. Optim. Appl.* 21 (2 2002), pp. 155–167.

[78] T. Serafini, G. Zanghirati, and L. Zanni. "Gradient Projection Methods for Large Quadratic Programs and Applications in Training Support Vector Machines". In: *Optim. Methods Softw.* 20 (2005), pp. 353–378.

[79] D. di Serafino, G. Toraldo, M. Viola, and J. L. Barlow. "A two-phase gradient method for quadratic programming problems with a single linear constraint and bounds on the variables". In: *SIAM J. Optim.* 28.4 (2018), 2809–2838.

[80] D. di Serafino, V. Ruggiero, G. Toraldo, and L. Zanni. "On the steplength selection in gradient methods for unconstrained optimization". In: *Appl. Math. Comput.* 318 (2018), pp. 176–195.

[81] Ph. L. Toint. "Some numerical results using a sparse matrix updating formula in unconstrained optimization". In: *Math. Comput.* 32 (1978), pp. 839–852.

[82] Ph. L. Toint. "A non-monotone trust region algorithm for nonlinear optimization subject to convex constraints". In: *Math. Prog.* 77 (1997), pp. 69–94.

[83] V. N. Vapnik and S. Kotz. *Estimation of dependences based on empirical data*. New York, NY, USA: Springer-Verlag, 1982.

[84] K. Schittkowski H. Zhang Y. H. Dai W. W. Hager. "The cyclic Barzilai-Borwein method for unconstrained optimization". In: *IMA J. Num. Anal.* 25.3 (2006), pp. 604–627.

[85] H. Liu Y. Huang. "On the rate of convergence of projected Barzilai-Borwein methods". In: *Optim. Methods Softw.* 30.4 (2015), pp. 880–892.

[86] G. Yu, L. Qi, and Y. H. Dai. "On Nonmonotone Chambolle Gradient Projection Algorithms for Total Variation Image Restoration". In: *J. Math. Imaging Vis.* 35 (2009), pp. 143–154.

[87] Y. Yuan. "A new stepsize for the steepest descent method". In: *J. Comp. Math.* 24 (2006), pp. 149–156.

[88] R. Zanella, G. Zanghirati, R. Cavicchioli, L. Zanni, P. Boccacci, M. Bertero, and G. Vicidomini. "Towards real-time image deconvolution: application to confocal and sted microscopy". In: *Sci. Rep.* 3 (2013), p. 2523.

[89] H. Zhang and W. Hager. "A nonmonotone line search technique and its application to unconstrained optimization". In: *SIAM J. Optim.* 14.4 (2004), pp. 1045–1056.

[90] B. Zhou, L. Gao, and Y. H. Dai. "Gradient Methods with Adaptive Step-Sizes". In: *Comput. Optim. Appl.* 35.1 (2006), pp. 69–86.

[91] M. Zhu, S. J. Wright, and T. F. Chan. "Duality-based algorithms for total-variation-regularized image restoration". In: *Comput. Optim. Appl.* 47.3 (2008), pp. 377–400.