

University of Parma Research Repository

A computational study on QP problems with general linear constraints

This is the peer reviewd version of the followng article:

Original

A computational study on QP problems with general linear constraints / Liuzzi, G.; Locatelli, M.; Piccialli, V.. - In: OPTIMIZATION LETTERS. - ISSN 1862-4472. - (2022). [10.1007/s11590-021-01846-6]

Availability: This version is available at: 11381/2917428 since: 2022-02-24T19:44:18Z

Publisher: Springer Science and Business Media Deutschland GmbH

Published DOI:10.1007/s11590-021-01846-6

Terms of use:

Anyone can freely access the full text of works made available as "Open Access". Works made available

Publisher copyright

note finali coverpage

(Article begins on next page)

Dear Author,

Here are the proofs of your article.

- You can submit your corrections **online**, via **e-mail** or by **fax**.
- For **online** submission please insert your corrections in the online correction form. Always indicate the line number to which the correction refers.
- You can also insert your corrections in the proof PDF and email the annotated PDF.
- For fax submission, please ensure that your corrections are clearly legible. Use a fine black pen and write the correction in the margin, not too close to the edge of the page.
- Remember to note the journal title, article number, and your name when sending your response via e-mail or fax.
- Check the metadata sheet to make sure that the header information, especially author names and the corresponding affiliations are correctly shown.
- Check the questions that may have arisen during copy editing and insert your answers/ corrections.
- Check that the text is complete and that all figures, tables and their legends are included. Also check the accuracy of special characters, equations, and electronic supplementary material if applicable. If necessary refer to the *Edited manuscript*.
- The publication of inaccurate data such as dosages and units can have serious consequences. Please take particular care that all such details are correct.
- Please **do not** make changes that involve only matters of style. We have generally introduced forms that follow the journal's style. Substantial changes in content, e.g., new results, corrected values, title and authorship are not allowed without the approval of the responsible editor. In such a case, please contact the Editorial Office and return his/her consent together with the proof.
- If we do not receive your corrections within 48 hours, we will send you a reminder.
- Your article will be published **Online First** approximately one week after receipt of your corrected proofs. This is the **official first publication** citable with the DOI. **Further changes are, therefore, not possible.**
- The printed version will follow in a forthcoming issue.

Please note

After online publication, subscribers (personal/institutional) to this journal will have access to the complete article via the DOI using the URL: http://dx.doi.org/[DOI].

If you would like to know when your article has been published online, take advantage of our free alert service. For registration and further information go to: <u>http://www.link.springer.com</u>.

Due to the electronic nature of the procedure, the manuscript and the original figures will only be returned to you on special request. When you return your corrections, please inform us if you would like to have these documents returned.

ArticleTitle A computational study on QP problems with general linear constraints Article Sub-Title Article CopyRight Springer-Verlag GmbH Germany, part of Springer Nature (This will be the copyright line in the final PDF) Journal Name Optimization Letters Corresponding Author Locatelli FamilyName Particle Given Name M. Suffix Division DIA Università degli Studi di Parma Organization Address Parma, Italy Phone Fax Email marco.locatelli@unipr.it URL ORCID https://orcid.org/0000-0001-7138-8653 Author FamilyName Liuzzi Particle Given Name G. Suffix Division Organization DIAG - "Sapienza" University of Rome Address Rome, Italy Phone Fax Email giampaolo.liuzzi@cnr.iasi.it URL ORCID Author Piccialli FamilyName Particle Given Name V. Suffix Division DICII - University of Rome Tor Vergata Organization Address Rome, Italy Phone Fax Email veronica.piccialli@uniroma2.it URL ORCID Schedule Received 31 May 2021 Revised 15 Dec 2021 Accepted In this paper we consider Quadratic Programming (QP) problems with general linear constraints. We show, through a computational Abstract investigation, that a careful selection of a suitable reformulation of such problems, together with the related relaxation, and an intensive application of bound tightening are simple but very effective ingredients in order to make a standard branch and bound approach very competitive and in some cases able to outperform even well known commercial solvers. Keywords (separated by '-Quadratic programming - Branch and bound - Linear and convex relaxations - Bound tightening ') Footnote Information

Metadata of the article that will be visualized in OnlineFirst

ORIGINAL PAPER



G. Liuzzi¹ · M. Locatelli² ⊡ · V. Piccialli³

Received: 31 May 2021 / Accepted: 15 December 2021 © The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract

- ² In this paper we consider Quadratic Programming (QP) problems with general linear
- ³ constraints. We show, through a computational investigation, that a careful selection
- 4 of a suitable reformulation of such problems, together with the related relaxation, and
- ⁵ an intensive application of bound tightening are simple but very effective ingredients
- 6 in order to make a standard branch and bound approach very competitive and in some
- ⁷ cases able to outperform even well known commercial solvers.
- ⁸ Keywords Quadratic programming · Branch and bound · Linear and convex
- 9 relaxations · Bound tightening

10 1 Introduction

- In this paper, we consider Quadratic Programming (QP) problems, where the objec-
- tive function is (non-convex) quadratic, and the feasible region is a polytope. More
 precisely, let

14

$$X = \left\{ \mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} \le \mathbf{b}, \ \mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq}, \ \mathbf{0}_n \le \mathbf{x} \le \mathbf{e}_n \right\},\$$

be a polytope, where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{A}_{eq} \in \mathbb{R}^{m_{eq} \times n}$, $\mathbf{b}_{eq} \in \mathbb{R}^{m_{eq}}$, while $\mathbf{0}_n$ and

 \mathbf{e}_n are the *n*-dimensional vectors with all components equal to 0 and 1, respectively.

 M. Locatelli marco.locatelli@unipr.it
 G. Liuzzi

giampaolo.liuzzi@cnr.iasi.it

V. Piccialli veronica.piccialli@uniroma2.it

- ¹ DIAG "Sapienza" University of Rome, Rome, Italy
- ² DIA, Università degli Studi di Parma, Parma, Italy
- ³ DICII University of Rome Tor Vergata, Rome, Italy

Deringer

2

Check fo

(1)

In case m = 0 ($m_{eq} = 0$), the constraints $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ ($\mathbf{A}_{eq}\mathbf{x} = \mathbf{b}$) are not present. Note that imposing $\mathbf{0}_n \leq \mathbf{x} \leq \mathbf{e}_n$ is without loss of generality, since we can always impose such constraints, possibly after a translation and a re-scaling of the variables. Then, the problem we consider in this paper is the following:

$$\min_{\mathbf{x}\in X}\frac{1}{2}\mathbf{x}^{\top}\mathbf{Q}\mathbf{x}+\mathbf{c}^{\top}\mathbf{x},$$

where $\mathbf{Q} \in \mathbb{R}^{n \times n}$ is symmetric and, usually, not positive semidefinite, while $\mathbf{c} \in \mathbb{R}^{n}$. Two relevant special cases are:

- $m = m_{eq} = 0$: minimization of a quadratic function over the unit box, denoted as BoxQP in what follows;

²⁶ $-m = 0, m_{eq} = 1, A_{eq} = \mathbf{e}_n^{\top}$, and $b_{eq} = 1$: minimization of a quadratic function ²⁷ over the unit simplex, denoted as StQP (Standard QP) in what follows.

Problem (1) turns out to be difficult. NP-hardness results have been proved also for 28 the BoxQP subclass (see, e.g., [16]), and for the StQP subclass (see the reformulation 29 of tha max clique problem as a StQP in [14]). Due to the difficulty of the problem, 30 Branch-and-Bound (B&B) approaches are usually recommended to tackle it. Many 31 recent works (e.g., [1,2,5,9,11,12,15,19]) have discussed B&B approaches for problem 32 (1) and its sub-classes. Such works differ under many respects like, e.g., the relaxations 33 employed to compute lower bounds, the branching strategies, and so on. We will briefly 34 review these aspects in the following sections. It is also worthwhile to remark that both 35 CPLEX and GUROBI, the best performing commercial solvers in the field of linear 36 and integer programming, have recently added the opportunity of solving problems 37 within the class (1). 38

In this paper we do not bring theoretical advances about QP problems, rather we 39 are interested in showing, through computational experiments, that when the structure 40 of the problem is weakened, say, when we move from highly structured problems 41 like BoxQP and StQP to QP problems over more general feasible polytopes, some 42 approaches become competitive. In particular, we would like to show that approaches 43 based on the choice of a suitable reformulation of a QP problem, with the related 44 relaxation, and on an intensive application of domain reduction strategies, turn out to 45 be very efficient. We believe that such computational observation is relevant and could 46 be taken into account in order to enhance the performance of other solvers. 47

The paper is structured as follows. In Sect. 2 we will present different reformulations 48 of problem (1) as well as the related relaxations. In Sect. 3 we briefly describe different 49 branching strategies, based on the optimal solutions of the relaxations. In Sect. 4 50 we discuss domain reduction techniques, which, as we will see, are able to strongly 51 enhance the performance of some B&B approaches. In Sect. 5 we discuss merits 52 and limitations of different reformulations and of the related relaxations. Finally, in 53 Sect. 6 we present and discuss some computational experiments over benchmark 54 instances. 55

Deringer

21

56 2 Problem reformulations and relaxations

Besides the original formulation (1), QP problems can be reformulated in alternative 57 ways, which also lead to different relaxations. Most of these reformulations and the 58 related relaxations are reviewed in [15]. Here we only report the two reformulations 59 (besides the original one) and the related relaxations which will be employed in this 60 paper, while some others are only briefly mentioned. The first, simple, reformulation 61 is what we call the *bilinear* reformulation. Interestingly, this is not reported in [15], but 62 we describe it here since, according to our experiments, in some cases it turns out to be 63 the one leading to the best results. Through the introduction of n additional variables 64 and the same number of equality constraints, the objective function is transformed 65 into a simple separable bilinear function: 66

67

72

$$\min_{\mathbf{x}\in X, \mathbf{y}} \frac{\frac{1}{2}\mathbf{x}^{\top}\mathbf{y} + \mathbf{c}^{\top}\mathbf{x}}{\mathbf{y} = \mathbf{Q}\mathbf{x}}.$$
(2)

⁶⁸ Next, let $\mathbf{Q} = \mathbf{U}\mathbf{D}\mathbf{U}^{\top}$ be the spectral decomposition of the symmetric matrix \mathbf{Q} . ⁶⁹ Then, after denoting by \mathbf{u}_i the eigen-vectors of matrix \mathbf{Q} (columns of matrix \mathbf{U}), with ⁷⁰ the related eigenvalues d_i , i = 1, ..., n, we call *spectral* reformulation of (1) the ⁷¹ following problem:

$$\min_{\mathbf{x}\in X, \mathbf{z}} \frac{1}{2} \sum_{i: d_i \ge 0} d_i \left[\mathbf{u}_i^\top \mathbf{x} \right]^2 + \frac{1}{2} \sum_{i: d_i < 0} d_i z_i^2 + \mathbf{c}^\top \mathbf{x}
z_i = \mathbf{u}_i^\top \mathbf{x} \qquad i: d_i < 0.$$
(3)

⁷³ Note that the dimension of vector \mathbf{z} is equal to the number of negative eigenvalues of \mathbf{Q} .

We also mention two further reformulations, namely: (i) the *KKT* (Karush-Kuhn-Tucker) reformulation, first employed, to the authors' knowledge, in [10], based on the observation that, due to the linearity of the constraints, all local optima of problem (1) are KKT points. Thus, after including also dual variables (the Lagrange multipliers), problem (1) can be reformulated through the addition of constraints imposing the KKT conditions;

(ii) the *MILP* reformulation, where the dual variables are added, the stationarity
 conditions of the KKT system are exploited to linearize the objective function (see
 [7]), and, finally, the nonlinear complementarity conditions are linearized after the
 addition of binary variables.

Relaxations In order to simplify the notation, we will use different symbols to 85 denote the feasible regions of different reformulations. In particular, we will denote 86 by: X_1 the feasible region of the original formulation (1), i.e., $X_1 \equiv X$; X_2 the feasible 87 region of reformulation (2); X_3 the feasible region of reformulation (3). Note that in 88 the description of each feasible region we need to consider all variables involved in 89 the reformulation. So, for instance, we have that $X_2 \subset \mathbb{R}^{2n}$ since in reformulation 90 (2) we need to include variables $\mathbf{y} \in \mathbb{R}^n$, besides the original variables $\mathbf{x} \in \mathbb{R}^n$. In 91 fact, when discussing relaxations, we will present them not (only) over the original 92 feasible regions but over subsets of these regions. More precisely, at some node of the 93

🖉 Springer

B&B tree, for some $i \in \{1, ..., 3\}$, we will consider an additional (polyhedral) set 94 \mathcal{X}_i , and we will define the relaxation over the subset $X_i \cap \mathcal{X}_i$ of the feasible region. 95 Usually, \mathcal{X}_i is a set obtained as a result of different branching operations. We will also 96 assume that in a given reformulation, say, the one denoted by index $i \in \{1, \ldots, 3\}$, 97 for all variables appearing in such reformulation, here generically denoted as ζ , lower 98 bounds ℓ_{ξ} and upper bounds u_{ξ} over $X_i \cap \mathcal{X}_i$ are available or, at least, can be easily 99 computed, e.g., by solving linear programs. Given this premise, now we will describe 100 in detail the relaxations. 101

A straightforward relaxation of the original formulation (1) is obtained by employ-102 ing McCormick under- and overestimators (see [13]): 103

 $O V = T_{T}$

 $1 \sum n$

$$\min_{\mathbf{x}\in X_{1}\cap\mathcal{X}_{1}, \mathbf{X}} \frac{\frac{1}{2}\sum_{i,j=1}^{n} Q_{ij}X_{ij} + \mathbf{c}^{\top}\mathbf{x}}{X_{ij} \ge \max\left\{\ell_{x_{i}}x_{j} + \ell_{x_{j}}x_{i} - \ell_{x_{i}}\ell_{x_{j}}, u_{x_{i}}x_{j} + u_{x_{j}}x_{i} - u_{x_{i}}u_{x_{j}}\right\}} \quad i, j : Q_{ij} > 0 \quad (4)$$

$$X_{ij} \le \min\left\{\ell_{x_{i}}x_{j} + u_{x_{j}}x_{i} - \ell_{x_{i}}u_{x_{j}}, u_{x_{i}}x_{j} + \ell_{x_{j}}x_{i} - u_{x_{i}}\ell_{x_{j}}\right\}} \quad i, j : Q_{ij} < 0,$$

where: the first set of constraints defines the convex envelope of the bilinear terms $x_i x_j$ 105 over the rectangle $[\ell_{x_i}, u_{x_i}] \times [\ell_{x_i}, u_{x_i}]$ for all *i*, *j* such that $Q_{ij} > 0$; the second set of 106 constraints defines the concave envelope of the bilinear terms $x_i x_j$ over the rectangle 107 $[\ell_{x_i}, u_{x_i}] \times [\ell_{x_i}, u_{x_i}]$ for all *i*, *j* such that $Q_{ij} < 0$. The relaxed problem is an LP with 108 (up to) n^2 additional variables, namely the entries X_{ij} , i, j = 1, ..., n, of the variable 109 matrix **X**, and (up to) $2n^2$ additional linear constraints. In fact, due to symmetries, the 110 number of variables and constraints can be (approximately) halved. Moreover, such 111 number is obviously strictly related to the sparsity of matrix \mathbf{Q} : the sparser matrix \mathbf{Q} is, 112 the lower the number of additional variables and constraints. The above relaxation is 113 also called McCormick relaxation. In the recent and interesting paper [1] it is observed 114 that such relaxation is weak but can be considerably strengthened with the addition 115 of valid linear inequalities. In particular, the authors consider Chvátal-Gomory cuts 116 for the so called Boolean Quadric Polytope, and prove that the only non-dominated 117 Chvátal-Gomory cuts are the odd-cycle inequalities. 118

McCormick underestimators can also be employed to define a relaxation of refor-119 mulation (2): 120

$$\min_{\mathbf{x},\mathbf{y}\in\mathcal{X}_{2}\cap\mathcal{X}_{2}, \mathbf{g}} \frac{1}{2} \mathbf{e}_{n}^{\top} \mathbf{g} + \mathbf{c}^{\top} \mathbf{x}$$

$$g_{i} \geq \max\left\{\ell_{x_{i}} y_{i} + \ell_{y_{i}} x_{i} - \ell_{x_{i}} \ell_{y_{i}}, u_{x_{i}} y_{i} + u_{y_{i}} x_{i} - u_{x_{i}} u_{y_{i}}\right\}, \quad i = 1, \dots, n.$$
(5)

Here, the right-hand side of the additional constraints defines the convex envelope of 122 the bilinear term $x_i y_i$ over the rectangle $[\ell_{x_i}, u_{x_i}] \times [\ell_{y_i}, u_{y_i}]$. The relaxed problem is 123 an LP with *n* additional variables g_i , i = 1, ..., n, and 2n additional constraints. 124

In reformulation (3) the objective function is separated into the sum of a convex 125 and a concave part. Then, a relaxation can be obtained by underestimating the concave 126 part: 127

🖄 Springer

121

128

$$\min_{\mathbf{x}, \mathbf{z} \in X_3 \cap \mathcal{X}_3, \mathbf{f}} \frac{\frac{1}{2} \sum_{i : d_i \ge 0} d_i \left[\mathbf{u}_i^{\top} \mathbf{x} \right]^2 + \frac{1}{2} \sum_{i : d_i < 0} d_i f_i + \mathbf{c}^{\top} \mathbf{x}}{f_i \le (\ell_{z_i} + u_{z_i}) z_i - \ell_{z_i} u_{z_i}} \qquad i : d_i < 0,$$
(6)

where each additional constraint defines the concave envelope of z_i^2 over the interval $[\ell_{z_i}, u_{z_i}]$. The relaxed problem is a Convex Programming (CP) problem, with a number of additional variables and constraints equal to the number of negative eigenvalues of matrix **Q**.

Concerning the KKT reformulation, in [5] it is first observed that it can be further
 reformulated as a completely positive problem and then the cone of completely positive
 matrices is relaxed into the tractable convex cone of doubly nonnegative matrices,
 thus leading to an SDP bound. The authors further observe that the relaxation can be
 strengthened by the addition of RLT constraints.

Finally, concerning the MILP reformulation, any valid relaxation for MILP prob-138 lems can be employed in this case. In fact, once the problem is formulated as a MILP, 139 there is no need to develop new solution methods: any of the existing methods, imple-140 mented in the best known commercial solvers, like CPLEX and GUROBI, can be 141 employed to solve them. Nevertheless, one can improve the performance of these 142 solvers by improving the input model. For instance, in the MILP reformulation com-143 plementarity conditions are translated into big-M constraints. Thus, strengthening the 144 upper bound values used in these constraints may have a relevant impact on the com-145 puting times. We refer to [9,19] for the discussion of MILP reformulations and their 146 application to QP problems. 147

148 3 Branching

The branching operation employed in a B&B algorithm is strictly related to the for mulation of the problem and the related relaxation. For QP problems we can classify
 branching into two broad categories:

152	Spatial branching:	the subset of the feasible region associated with a node of the
153		B&B tree is subdivided into two subsets, obtained by: i) selecting
154		a variable, say x_i ; (ii) selecting a reference value for that variable,
155		say x_i^* ; (iii) defining the first subset by adding constraint $x_i \leq x_i^*$,
156		and the second subset by adding constraint $x_i \ge x_i^*$. In this case
157		the two subsets are not disjoint but share a common face, where
158		the selected variable is equal to the reference value. By spatial
159		branching it is usually only possible to guarantee that in a finite
160		number of iterations the globally optimal solution is reached
161		within a given precision $\varepsilon > 0$;
162	KKT branching:	this is strictly associated to the KKT reformulation, where a node
163		of the B&B tree is split into two child nodes, by first selecting
164		one of the complementarity conditions and then in each child
165		node imposing that one of the two (linear) factors appearing in
166		a complementarity condition is equal to 0. In case of the MILP
167		reformulation this is obtained by fixing one binary variable to 0 in

168 169 170 one child node, and to 1 in the other child node. KKT branching allows to terminate in a finite number of iterations without the need of imposing a positive precision.

While we mentioned, for the sake of completeness, the KKT branching, in this paper 171 we will adopt spatial branching. This is the natural branching approach in case lower 172 bounds are computed via the linear relaxations (4) and (5), or via the convex relaxation 173 (6), as we will do throughout the paper. Concerning the selected variable, if the linear 174 relaxation (4) is employed, then it is one of the original variables $x_i, i \in \{1, \dots, n\}$, 175 while if the convex relaxation (6) is employed, it is one of the z_i variables, for all *i* such 176 that $d_i < 0$. In case the linear relaxation (5) is employed, then we can either select one 177 of the original variables x_i , or one of the variables y_i , $i \in \{1, \ldots, n\}$. It is worthwhile to 178 remark that finiteness of the B&B algorithm within a positive precision is guaranteed 179 even if branching is only performed with respect to variables x_i or only with respect 180 to variables y_i . This property derives from the convex envelope of a bilinear term over 181 a rectangle, defined by McCormick underestimators, converging to the bilinear term 182 itself even when the length of only one of the edges of the rectangle converges to 0. 183 In fact, according to our experiments, branching on y_i variables is more efficient than 184 branching on x_i variables, possibly because each variable y_i is a linear function of 185 multiple original variables and a limitation on such variable has an impact on all the 186 original variables on which it depends. 187

The branching variable is selected to be the one with the largest error. More precisely, once we solve a relaxation, we consider the difference between an underestimated function and its underestimator computed at the optimal solution of the relaxation, and select the variable with the largest error. Therefore, for relaxation (4), let ($\mathbf{X}^{\star}, \mathbf{x}^{\star}$) be the optimal solution of the relaxation. Then, we select the variable with index

 $k \in \arg\max_{i=1,\dots,n} \sum_{j=1}^{n} \mathcal{Q}_{ij}(x_j^{\star} x_i^{\star} - X_{ij}^{\star}).$ $\tag{7}$

For relaxation (5), let $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{g}^*)$ be the optimal solution of the relaxation. Then, we select the variable with index

197

200

194

 $k \in \arg \max_{i=1,\dots,n} x_i^* y_i^* - g_i^* \tag{8}$

(in this case the variable can be either x_k or y_k). For relaxation (6), let $(\mathbf{x}^*, \mathbf{z}^*, \mathbf{f}^*)$ be the optimal solution of the relaxation. Then, we select the variable with index

$$k \in \arg \max_{i : d_i < 0} f_i^{\star} - \left(z_i^{\star} \right)^2.$$
⁽⁹⁾

In all cases, the reference value, i.e., the value with respect to which we perform the branching operation, is the value of the selected variable at the optimal solution of the relaxation.

Deringer

SPI Journal: 11590 Article No.: 1846 TYPESET DISK LE CP Disp.: 2021/12/24 Pages: 15 Layout: Small-Ex

²⁰⁴ 4 Domain reduction/Bound tightening

As previously seen, in spatial branching the branching variable and the related branch-205 ing value are selected in such a way to reduce as much as possible the underestimating 206 error. Indeed, since underestimating functions are based on lower and upper bounds 207 of variables, by changing one of such bounds through the branching operation, we are 208 able to improve the quality of the underestimation. However, a much larger improve-209 ment at a given node of the B&B tree can be attained by so called *domain reduction* 210 or bound tightening procedures (see, e.g., [3,4,8,17,18]). These reduce the range of 211 the variables and, consequently, improve the quality of the underestimator. A rather 212 expensive but, as we will see, also quite effective domain reduction procedure is based 213 on the minimization and maximization of a single variable over the feasible region 214 of the B&B node with an additional constraint imposing that the linear or convex 215 underestimating function is not larger than the current global upper bound (GUB in 216 what follows). For instance, let us consider relaxation (6). We notice that the under-217 estimating function depends on variables $z_i, i \in \{1, ..., n\}$: $d_i < 0$. Let z_k be one 218 of such variables. Then, we can improve the lower and upper bound of this variable 219 by solving the following two convex programs: 220

$$\min / \max_{\mathbf{x}, \mathbf{z} \in X_3 \cap \mathcal{X}_3, \mathbf{f}} z_k$$

$$f_i \leq (\ell_{z_i} + u_{z_i}) z_i - \ell_{z_i} u_{z_i}$$

$$i : d_i < 0,$$

$$\frac{1}{2} \sum_{i : d_i \geq 0} d_i \left[\mathbf{u}_i^\top \mathbf{x} \right]^2 + \frac{1}{2} \sum_{i : d_i < 0} d_i f_i + \mathbf{c}^\top \mathbf{x} \leq GUB.$$

$$(10)$$

221

Note that, once new bounds for the variable are computed, these allow to strengthen 222 the last constraint and, thus, a further reduction is possible. In practice, one proceeds 223 as follows: (i) first, select a subset of variables (again, variables for which the under-224 estimating error is largest are selected); (ii) then, problems (10) are solved for each 225 one of these variables; (iii) next, a new lower bound is computed by solving relax-226 ation (6) with the updated bounds; (iv) finally, if the new lower bound significantly 227 improves the previous one, then it is worthwhile to try to further reduce the variable 228 ranges and, thus, the whole procedure is repeated. Of course, the same procedure can 229 be applied when relaxations (4) and (5) are employed. The overall procedure is quite 230 expensive, since at each B&B node many LP or CP problems need to be solved. But, 231 as we will see in Sect. 6, when applied to general, poorly structured QP problems, it 232 is also extremely effective, allowing for a very large reduction of the number of B&B 233 nodes to be explored. In order to reduce the computational burden of bound tightening, 234 in this paper we adopted the strategy of tightening bounds only for the variables for 235 which the error values (7)-(9) are positive. We remark that some recent papers, like, 236 e.g., [8], explore further filters to choose variables on which to apply bound tightening 237 procedures. 238

239 5 Merits and limits of different approaches

²⁴⁰ In the previous sections we briefly revised different approaches for the solution of ²⁴¹ QP problems. Not surprisingly, none of them strictly dominates the others. Special

🖉 Springer

structured QP problems, like BoxQP and StQP problems, have a strong combinatorial 242 component. For instance, in BoxQP it turns out (see [10]) that $Q_{ii} \leq 0 \Rightarrow x_i \in \{0, 1\}$. 243 Then, some of the variables can be immediately considered as binary ones. In StQP 244 problems, it holds that $Q_{ii} + Q_{ij} - 2Q_{ij} \le 0 \Rightarrow x_i x_j = 0$. Thus, if some variable x_i 245 is imposed to be positive at some node of the B&B tree, we can fix to 0 the value of 246 variables x_i , for each *i* such that the condition holds. All these combinatorial aspects 247 should be exploited for the efficient solution of these special QP problems, and that 248 also makes the use of spatial branching not advisable for them, even when powered 249 with bound tightening procedures. The best results for StQP problems are reported by 250 the QUADPROGIP approach discussed in [19] and by the approach presented in [9], 251 both based on a MILP reformulation, and by the approach presented in [11], based 252 on a suitable relaxation of the original reformulation, related to the computation of 253 the convex envelope of some quadratic functions over the unit simplex, and with a 254 branching rule strictly related to the KKT conditions of the StQP problem. For what 255 concerns BoxQP problems, very good results are reported by the approach discussed 256 in [1], based on the addition of Chvátal-Gomory cuts. In the same paper the remarkable 257 performance of CPLEX emerges and, moreover, it is observed that the B&B approach 258 based on the SDP bound proposed by [5], called QUADPROGBB, becomes extremely 259 competitive when the density of matrix Q increases. In [15] further QP problems with 260 a special structure are discussed and the authors propose approaches, embedded into 261 the BARON solver, which allow this solver to outperform CPLEX and GUROBI over 262 these problems. 263

But while spatial branching powered by bound tightening procedures does not 264 appear a valid alternative for problems with a special structure, it comes into play 265 again as soon as we weaken the structure. In the recent paper [12] we discussed QP 266 problems arising from an application in game theory, which, at a first glance, appear 267 as a mild modification of StQP problems. Indeed, in such problems the feasible region 268 is the unit simplex, while the objective function is the sum of a quadratic function and 269 a convex piecewise linear function. The problem can be converted into the form (1) 270 by replacing the convex piecewise linear function with a single variable, and adding 271 constraints imposing that this variable is not lower than any of the linear pieces. 272 In spite of many attempts with all the previously mentioned approaches and with 273 different commercial solvers such as CPLEX, GUROBI, BARON, it turned out that 274 the best approach is, by far, an approach based on the bilinear reformulation with 275 an intensive application of a bound tightening procedure. This suggested to us that 276 such an approach could be very competitive not only for the QP problems arising 277 from the game theory application discussed in [12], but also for all QP problems with 278 general linear constraints (though not for special structured problems such as BoxQP 279 and StQP). The aim of this paper is to bring a computational evidence of this fact 280 through experiments on benchmark instances. However, the approach proposed in 281 [12], while performing pretty well in some cases, also displays bad performance on 282 some of the benchmark instances. What we realized is that the bad performance is 283 related to the choice of the reformulation. More precisely, through our experiments, 284 we observed the following. Reformulation (2) and the related relaxation (5) is quite 285 competitive with and in many cases outperforms the original formulation (1) with the 286 related relaxation (4). Note that this was not easy to foresee. However, both relaxations 287

D Springer

are, in some cases, strongly outperformed by the spectral reformulation and the related 288 relaxation (6). Thus, the question is now how to choose a proper formulation. This 289 topic has been addressed also in [15], where, however, the bilinear reformulation was 290 not considered. In particular, in that paper it is suggested to test bounds based on 29 different reformulations at the initial nodes of the B&B tree, and then choose the 292 reformulation leading to the best bounds. Here we adopt a much simpler rule, based 293 on the observation that the convex bound (6) is expected to be more effective when the 294 number of nonnegative eigenvalues is large. Thus, our simple rule will be that of first 295 computing the eigenvalues of **Q**, and then using the spectral reformulation when the 296 number of negative ones is lower than a given fraction of n, or adopting the bilinear 297 reformulation otherwise. In particular, we employed the spectral decomposition only 298 when the number of negative eigenvalues is lower than 0.4n. We made this choice, 299 which favors the adoption of the bilinear reformulation with respect to the spectral one, 300 because the larger cost of solving CP problems with respect to LP problems suggests 301 to employ the spectral decomposition only when the dimension of the concave part in 302 the spectral decomposition (equivalent to the number of negative eigenvalues) is not 303 too large. In the experiments we observed that decreasing the threshold fraction to, 304 e.g., 0.3n does not worsen the performance, while increasing it may lead to poorer 305 performance over some instances. Note that while in this work we employed a simple 306 and nonadaptive rule, exploration of further adaptive rules, as done in [15], is indeed 307 an interesting topic. 308

309 6 Computational experiments

6.1 Setup of the experiments

In the literature there are many sets of benchmark instances for QP problems (see, 311 in particular, [6]). However, for what concerns OP problems with general linear con-312 straints the main ones, to the authors' knowledge, are CUTEr, Globallib and 313 RandQP. We tested all of them, but the first two classes appeared less challenging and 314 we do not report the results over them. Class RandQP includes 16 instances for each 315 dimension n = 20, 30, 40, 50. The instances can be downloaded, e.g., at https://github. 316 com/xiawei918/quadprogIP/blob/master/QuadProgBB instances.zip. We solved all 317 RandQP instances by the approaches proposed in this work and by the best perform-318 ing solvers for QP problems available in the literature. All tests have been performed 319 on an Intel[®] Core[™] i7-10750H CPU @ 2.60GHz with 16GB RAM and running Win-320 dows 10 Pro. The source code for the proposed approaches can be found at https:// 321 github.com/gliuzzi/QPL. 322

6.2 Discussion of the results

The CPU times required by all the tested approaches are given in Table 1, where, however, due to space limitation, we do not report the results for n = 20 (but all these instances are relatively simple and solved by most of the approaches within few

🖄 Springer

seconds). A time limit of 1800s has been imposed. All algorithms were stopped as 327 soon as a relative precision $\varepsilon = 10^{-5}$ was reached or, alternatively, the time limit 328 was reached (a - in the table means that the solver reached the time limit over the329 instance). Full results with additional information, such as the gap at termination 330 when the time limit is reached, can be found at https://github.com/gliuzzi/QPL/tree/ 331 main/results. At first, we discuss the comparison between the approaches presented in 332 this paper. Following [12], we call these approaches Branch-and-Tightening (B&T in 333 what follows), in order to put in evidence the relevance of bound tightening techniques 334 within them. We compared the following approaches: B&T (XY), based on the original 335 formulation (1) and the related relaxation (4); B&T (Bil), based on the reformulation 336 (2) and the related relaxation (5); B&T(Conv), based on the reformulation (3) and 337 the related relaxation (6); B&T (Mix), which chooses between reformulation (2) and 338 reformulation (3) according to the number of negative eigenvalues of \mathbf{Q} , namely, the 339 latter reformulation is chosen if the number of negative eigenvalues is lower than 340 0.4n. All these approaches have been implemented in Julia language (version 1.5.2) 341 by solving LP subproblems with Gurobi (version 9.1.1) and convex subproblems with 342 CPLEX (version 12.10). 343

The first, not obvious, observation is that B&T (XY) is outperformed by B&T (Bil) 344 and B&T (Conv), which is also the reason why the latter two approaches are mixed 345 in B&T (Mix). More precisely, we notice there are few instances where the com-346 puting times of B&T(XY) are better both than those of B&T(Bil) and than those 347 of B&T (Conv). But: i) these are mostly instances with dimension n = 20 (five, 348 overall), while there is only one instance at dimension n = 30, and none with dimen-349 sion n = 40, 50; ii) even when B&T (XY) is the best approach, its computing times 350 do not differ much from those of B&T (Bil). Instead, there are many cases where 351 B&T (Bil) is the best approach and strongly outperforms B&T (XY), in particular at 352 dimension n = 50. 353

The second relevant observation is that the mixed strategy B&T(Mix) is the best one. Indeed, the proposed rule to select the proper reformulation selects the best approach between B&T(Bil) and B&T(Conv) in 61 out of 64 cases, while in the remaining three cases the performance of the selected approach is close to that of the best approach.

The third observation does not emerge from the reported results but still is quite 359 relevant: all these approaches become quite inefficient without the application of a 360 bound tightening procedure. Indeed, the experiments we performed without bound 36 tightening (not reported here) show that even at dimension n = 20 the computing 362 times considerably increase and some instances are not solved within the time limit. 363 We also tested a version with a less intensive application of bound tightening. Namely, 364 rather than repeating bound tightening over all variables until there is a significant 365 reduction of the lower bound, we just performed a single round of bound tightening 366 over all variables. By this approach the number of nodes of the B&B tree increases, 367 but the computational cost per node decreases and the two effects tend to compensate 368 each other. Indeed, in terms of overall computing times we did not observe significant 369 differences between the intensive and less intensive version of bound tightening. For 370 what concerns the number of nodes, we remark that this is very small for the mixed 371 strategy B&T (Mix) with intensive bound tightening: 1861 nodes are visited for the 372

instance qp50_25_3_3, while in three other instances more than 100 nodes (at most 135) are visited, and in all the remaining ones less than 50 nodes are visited. In summary, selecting a proper reformulation and an intensive bound tightening are the keys for the good performance of B&T(Mix). The next step will be to show that B&T(Mix) is competitive with the current best solvers for QP problems with general linear constraints.

In [19] many experiments are reported with different class of QP problems and 379 different solvers. According to these experiments, solver QUADPROGBB, which 380 displays very interesting performance over BoxQP problems, usually has poor perfor-38 mance on RandQP instances. Another solver, QUADPROGIP, appears to be very good 382 at some instances, in particular at dimension n = 50, but, on the other hand, the same 383 solver is unable to solve some instances within a time limit of 10,000 s. According to the 384 results reported in that paper, the most robust solver over such instances is CPLEX. For 385 this reason, we compare the performance of B&T (Mix) with CPLEX itself (version 386 12.10), with GUROBI (version 9.1.1), which was not included in the computational 387 study of [19], with BARON (version 21.1.7), both in view of the enhancements of this 388 solver described in [15] and because of the fact that BARON relies on bound tighten-389 ing as the approaches discussed in this paper, and, finally, with OUADPROGIP. Note 390 that all these solvers have been run with their default settings. According to the results 391 reported in Table 1, we notice that: 392

- B&T (Mix) is better than BARON and QUADPROGIP at all dimensions,
 although, as also reported in [19], QUADPROGIP performs well on some large
 instances;
- with respect to CPLEX, B&T (Mix) is slightly worse at dimensions n = 20, 30(overall, it has better computing times in 12 out of 32 instances), but becomes better at dimensions n = 40, 50 (overall, it has better computing times in 21 out of 32 instances);
- with respect to GUROBI, B&T (Mix) is clearly worse at dimensions n = 20, 30(overall, it has better computing times only in 4 out of 32 instances), but becomes competitive at dimensions n = 40, 50 (overall, it has better computing times in 17 out of 32 instances).
- Figure 1 allows to make the most relevant observation. In this figure we report com-404 puting times (in seconds) over the x-axis and the fraction of problems solved along 405 the y-axis. It can be seen that the curve corresponding to B&T (Mix) is initially 406 below those of CPLEX and GUROBI, but then it gets above them. More precisely, 407 B&T (Mix) is able to solve all but one instance within 30s (and the remaining one 408 in approximately $330 \, \text{s}$), while all other approaches are unable to solve at least two 409 instances within the time limit. Thus, B&T (Mix) is not always the best performing 410 approach, but it appears to scale better than the other approaches with respect to the 411 dimension, and to be the most robust approach. We remark that at https://github.com/ 412 gliuzzi/QPL it is possible to download four additional figures, with the same informa-413 tion reported in Figure 1, but with the instances separated according to the four tested 414 dimensions n = 20, 30, 40, 50. 415

🖄 Springer

Table 1 CPU times (in seconds) over the set of RandQP instances with dimension n = 30, 40, 50, forsolvers B&T(Bil), B&T(XY), B&T(CONV), B&T(Mix), QuadprogIP, BARON, Gurobi, and CPLEX

Instance	Bil	XY	Conv	Mix	QuadprogIP	BARON	Gurobi	CPLEX
qp30_15_1_1	-	_	0.05	0.05	2.28	0.02	0.02	0.24
qp30_15_1_2	1.63	1.38	453.5	1.63	172.36	1.33	0.07	0.59
qp30_15_1_3	9.79	2.02	1.24	1.24	5.31	2.92	0.2	0.58
qp30_15_1_4	8.16	2.66	0.82	0.82	2.42	0.36	0.12	0.42
qp30_15_2_1	3.44	2.6	1.55	1.55	2.13	1.06	0.09	0.68
qp30_15_2_2	2.42	4.58	792.52	2.42	60.45	23.14	0.72	1.45
qp30_15_2_3	1.17	2.19	7.75	1.17	4.41	4.98	0.39	0.59
qp30_15_2_4	6.31	4.96	3.52	3.52	3.21	24.17	0.19	0.54
qp30_15_3_1	3.2	13.33	578.31	3.2	7.4	33.05	0.81	1.39
qp30_15_3_2	1.42	1.67	4.61	1.42	2.3	2.62	0.23	0.82
qp30_15_3_3	_	-	0.78	0.78	6.79		76.79	0.23
qp30_15_3_4	0.39	1.23	3.15	0.39	2.38	1.41	0.27	0.57
qp30_15_4_1	256.54	109.68	0.84	0.84	2.26	24.88	1.91	0.49
qp30_15_4_2	1.03	3.1	1335.0	1.03	19.57	10.27	0.77	1.28
qp30_15_4_3	426.81	80.8	2.77	2.77	2.62	203.75	3.89	3.56
qp30_15_4_4	0.53	1.86	2.05	0.53	2.28	3.34	0.34	1.17
qp40_20_1_1	1.13	2.46	-	1.13	167.5	5.67	0.33	0.77
qp40_20_1_2	14.02	18.33	6.3	14.02	10.41	-	0.2	0.54
qp40_20_1_3	1.36	1.86	19.64	1.36	8.13	-	0.29	0.61
qp40_20_1_4	17.2	12.69	4.12	4.12	3.92	62.19	0.9	1.11
qp40_20_2_1	627.02	83.76	1.96	1.96	3.6	0.83	0.19	0.39
qp40_20_2_2	-	-	7.26	7.26	946.33	-	-	-
qp40_20_2_3	6.78	29.02	-	6.78	7	230.11	10.86	11.16
qp40_20_2_4	-	-	0.08	0.08	200.37	0.03	0.01	0.16
qp40_20_3_1	1.08	4.48	111.02	1.08	6.15	13.88	2.44	1.83
qp40_20_3_2	-	-	2.81	2.81	106.13	-	299.94	649.84
qp40_20_3_3	0.97	3.3	10.54	0.97	3.78	114.58	4.53	23.02
qp40_20_3_4	1.59	10.69	32.42	1.59	6.36	65.59	6.92	6.66
qp40_20_4_1	-	-	6.67	6.67	1537.64	-	-	-
qp40_20_4_2	2.05	32.24	-	2.05	92.92	101.92	8.55	4.64
qp40_20_4_3	30.99	727.32	238.72	30.99	68.42	-	155.49	39.78
qp40_20_4_4	4.39	108.18	98.29	4.39	14.35	1314.08	74.36	12.01
qp50_25_1_1	1.76	7.24	-	1.76	-	12.98	0.2	1.18
qp50_25_1_2	1.84	4.99	5.19	5.19	15.74	278.41	2.71	1.54
qp50_25_1_3	11.2	64.27	7	11.2	-	1775.62	21.35	28.19
qp50_25_1_4	5.55	9.08	13.39	13.39	6.81	-	1.83	2.6
qp50_25_2_1	-	1022.15	0.06	0.06	11.03	0.08	0.01	0.22
qp50_25_2_2	2.7	22.03	-	2.7	1382.56	51.81	2.65	2.22

Instance	Bil	XY	Conv	Mix	QuadprogIP	BARON	Gurobi	CPLEX
qp50_25_2_3	1.67	14.04	708.68	1.67	436.25	358.06	2.02	2.46
qp50_25_2_4	1.36	4.3	25.19	1.36	6.8	_	4.9	1.14
qp50_25_3_1	11.34	58.82	_	11.34	_	498.78	7	5.46
qp50_25_3_2	0.7	1.59	2.49	0.7	6.77	22.11	0.7	2.53
qp50_25_3_3	332.57	1461.83	1349.27	332.57	457.53	_	247.29	984.61
qp50_25_3_4	335.29	782.67	6.98	6.98	7.69	-	53.57	458.4
qp50_25_4_1	_	_	11.93	11.93	12.82	-	113.93	364.25
qp50_25_4_2	4.25	47.39	9.43	9.43	7.1	1109.61	22.59	14.74
qp50_25_4_3	15.91	502.35	_	15.91	_		208.99	33.05
qp50_25_4_4	5.69	22.57	50.19	5.69	8.2	183.53	2.85	20.41



A - means that the solver reached the time limit (1800 s) over the instance



Fig. 1 Fraction of problems solved (y-axis) versus computing time (x-axis) for the different tested solvers

D Springer

416 7 Conclusions

In this work we have shown through some computational experiments that OP prob-417 lems with general linear constraints can be efficiently solved by a standard branch 418 and bound approach powered by: i) a careful selection of a suitable reformulation of 419 the QP problem and of its relaxation; ii) an intensive application of bound tightening. 420 Our computational experiences show that the proposed approach is competitive and 421 is sometimes able to outperform the best known solvers for QP problems. As a pos-422 sible topic for future research we would like to see whether the performance can be 423 further enhanced, e.g., by adaptive rules which may select different reformulations in 424 different nodes of the branch and bound tree (currently the reformulation is fixed in 425 advance), or by filtering techniques, such as those in [8], which are able to reduce the 426 computational burden of the bound tightening procedure, which is currently the main 427 cost of the proposed approaches. 428

429 References

- Bonami, P., Günlük, O., Linderoth, J.: Globally solving nonconvex quadratic programming problems with box constraints via integer programming methods. Math. Program. Comput. 10, 333–382 (2018)
- Burer, S., Vandenbussche, D.: Globally solving box-constrained non-convex quadratic programs with semidefinite-based finite branch-and-bound. Comput. Optim. Appl. 43(2), 181–195 (2009)
- 434
 Caprara, A., Locatelli, M.: Global optimization problems and domain reduction strategies. Math.
 435 Program. 125(1), 123–137 (2010)
- 4. Caprara, A., Locatelli, M., Monaci, M.: Theoretical and computational results about optimality-based domain reductions. Comput. Optim. Appl. 64, 513–533 (2016)
- 5. Chen, J., Burer, S.: Globally solving nonconvex quadratic programming problems via completely
 positive programming. Math. Program. Comput. 4(1), 33–52 (2012)
- 6. Furini, F., Traversi, E., Belotti, P., Frangioni, A., Gleixner, A., Gould, N., Liberti, L., Lodi, A., Misener, R., Mittelmann, H., Sahinidis, N.V., Vigerske, S., Wiegele, A.: QPLIB: a library of quadratic programming instances. Math. Program. Comput. 11(2), 237–265 (2019)
- Giannessi, F., Tomasin, E.: Nonconvex quadratic programs, linear complementarity problems, and
 integer linear programs, Lecture Notes in Computer Science, vol. 3, Springer, Berlin, pp. 437–449
 (1973)
- 8. Gleixner, A., Berthold, T., Müller, B., Weltge, S.: Three enhancements for optimization-based bound tightening. J. Global Optim. 67, 731–757 (2017)
- Gondzio, J., Yildrim, E.A.: Global solutions of nonconvex Standard Quadratic Programs via Mixed
 Integer Linear Programming reformulations, available at https://arxiv.org/pdf/1810.02307.pdf
- Hansen, P., Jaumard, B., Ruiz, M., Xiong, J.: Global minimization of indefinite quadratic functions
 subject to box constraints. Nav. Res. Logist. 40(3), 373–392 (1993)
- Liuzzi, G., Locatelli, M., Piccialli, V.: A new branch-and-bound algorithm for standard quadratic
 programming problems. Optim. Methods Softw. 34, 79–97 (2019)
- Liuzzi, G., Locatelli, M., Piccialli, V., Rass, S.: Computing mixed strategies equilibria in presence of
 switching costs by the solution of nonconvex QP problems, Computational OPtimization and Appli cations, to appear, available at (2021) http://www.optimization-online.org/DB_FILE/2020/03/7656.
 pdf
- 458
 13. McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: part I-convex underestimating problems. Math. Program. 10, 147–175 (1976)
- 460 14. Motzkin, T.S., Straus, E.G.: Maxima for graphs and a new proof of a theorem of Turán. Canadian J.
 461 Math. 17(4), 533–540 (1965)
- 462 15. Nohra, C.J., Raghunathan, A.U., Sahinidis, N.V.: Spectral relaxations and branching strategies for
 global optimization of mixed-integer quadratic programs. SIAM J. Optim. 31, 142–171 (2021)

- 464 16. Pardalos, P.M., Vavasis, S.A.: Quadratic programming with one negative eigenvalue is NP-hard. J.
 Global Optim. 1(1), 15–22 (1991)
- 466 17. Sahinidis, N.V.: BARON: a general purpose global optimization software package. J. Global Optim.
 467 8(2), 201–205 (1996)
- 468 18. Tawarmalani, M., Sahinidis, N.V.: Global optimization of mixed-integer nonlinear programs: a theo-469 retical and computational study. Math. Program. **99**(3), 563–591 (2004)
- 470
 19. Xia, W., Vera, J.C., Zuluaga, L.F.: Globally solving nonconvex quadratic programs via linear integer
 471
 471 programming techniques. INFORMS J. Comput. 32(1), 40–56 (2020)

472 Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps
 473 and institutional affiliations.

🖄 Springer

Author Query Form

Please ensure you fill out your response to the queries raised below and return this form along with your corrections

Dear Author

During the process of typesetting your article, the following queries have arisen. Please check your typeset proof carefully against the queries listed below and mark the necessary changes either directly on the proof/online grid or in the 'Author's response' area provided below

Query	Details required	Author's response
1.	Please confirm if the corresponding author is correctly identified. Amend if necessary.	000
2.	Please confirm if the inserted city name is correct. Amend if necessary.	