A computational study on QP problems with general linear constraints

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]

Publisher:
Springer Science and Business Media Deutschland GmbH

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

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: http://www.link.springer.com.

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.

Metadata of the article that will be visualized in OnlineFirst

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	FamilyName	**Locatelli**
	Particle	
	Given Name	**M.**
	Suffix	
	Division	DIA
	Organization	Università degli Studi di Parma
	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	FamilyName	**Piccialli**
	Particle	
	Given Name	**V.**
	Suffix	
	Division	
	Organization	DICII - University of Rome Tor Vergata
	Address	Rome, Italy
	Phone	
	Fax	
	Email	veronica.piccialli@uniroma2.it
	URL	
	ORCID	
Schedule	Received	31 May 2021
	Revised	
	Accepted	15 Dec 2021
Abstract	In this paper we consider Quadratic Programming (QP) problems with general linear constraints. We show, through a computational 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		

Optimization Letters
https://doi.org/10.1007/s11590-021-01846-6

ORIGINAL PAPER

A computational study on QP problems with general linear constraints

G. Liuzzi[1] · M. Locatelli[2] · V. Piccialli[3]

Received: 31 May 2021 / Accepted: 15 December 2021

Abstract

In this paper we consider Quadratic Programming (QP) problems with general linear constraints. We show, through a computational investigation, that a careful selection 1
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. 2

Keywords Quadratic programming · Branch and bound · Linear and convex relaxations · Bound tightening

1 Introduction

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

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

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

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

In case $m = 0$ ($m_{eq} = 0$), the constraints $\mathbf{Ax} \le \mathbf{b}$ ($\mathbf{A}_{eq}\mathbf{x} = \mathbf{b}$) are not present. Note that imposing $\mathbf{0}_n \le \mathbf{x} \le \mathbf{e}_n$ 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:

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

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;
- $m = 0,\ m_{eq} = 1$, $\mathbf{A}_{eq} = \mathbf{e}_n^\top$, and $b_{eq} = 1$: minimization of a quadratic function 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 the BoxQP subclass (see, e.g., [16]), and for the StQP subclass (see the reformulation of tha max clique problem as a StQP in [14]). Due to the difficulty of the problem, Branch-and-Bound (B&B) approaches are usually recommended to tackle it. Many recent works (e.g., [1,2,5,9,11,12,15,19]) have discussed B&B approaches for problem (1) and its sub-classes. Such works differ under many respects like, e.g., the relaxations employed to compute lower bounds, the branching strategies, and so on. We will briefly review these aspects in the following sections. It is also worthwhile to remark that both CPLEX and GUROBI, the best performing commercial solvers in the field of linear and integer programming, have recently added the opportunity of solving problems within the class (1).

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

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

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

2 Problem reformulations and relaxations

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

$$\begin{aligned} \min_{\mathbf{x}\in X,\ \mathbf{y}} \ & \tfrac{1}{2}\mathbf{x}^\top\mathbf{y} + \mathbf{c}^\top\mathbf{x} \\ & \mathbf{y} = \mathbf{Q}\mathbf{x}. \end{aligned} \tag{2}$$

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

$$\begin{aligned} \min_{\mathbf{x}\in X,\ \mathbf{z}} \ & \tfrac{1}{2}\textstyle\sum_{i\,:\,d_i\geq 0} d_i \left[\mathbf{u}_i^\top\mathbf{x}\right]^2 + \tfrac{1}{2}\textstyle\sum_{i\,:\,d_i<0} d_i z_i^2 + \mathbf{c}^\top\mathbf{x} \\ & z_i = \mathbf{u}_i^\top\mathbf{x} \qquad\qquad i\ :\ d_i < 0. \end{aligned} \tag{3}$$

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

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

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

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

$$
\begin{aligned}
\min_{\mathbf{x} \in X_1 \cap \mathcal{X}_1,\ \mathbf{X}} \quad & \tfrac{1}{2} \textstyle\sum_{i,j=1}^{n} Q_{ij} X_{ij} + \mathbf{c}^\top \mathbf{x} \\
& X_{ij} \geq \max\left\{\ell_{x_i} x_j + \ell_{x_j} x_i - \ell_{x_i}\ell_{x_j},\, u_{x_i} x_j + u_{x_j} x_i - u_{x_i} u_{x_j}\right\} \quad i, j \,:\, Q_{ij} > 0 \\
& X_{ij} \leq \min\left\{\ell_{x_i} x_j + u_{x_j} x_i - \ell_{x_i} u_{x_j},\, u_{x_i} x_j + \ell_{x_j} x_i - u_{x_i} \ell_{x_j}\right\} \quad i, j \,:\, Q_{ij} < 0,
\end{aligned}
\tag{4}
$$

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

McCormick underestimators can also be employed to define a relaxation of reformulation (2):

$$
\begin{aligned}
\min_{\mathbf{x},\mathbf{y} \in X_2 \cap \mathcal{X}_2,\ \mathbf{g}} \quad & \tfrac{1}{2} \mathbf{e}_n^\top \mathbf{g} + \mathbf{c}^\top \mathbf{x} \\
& g_i \geq \max\left\{\ell_{x_i} y_i + \ell_{y_i} x_i - \ell_{x_i}\ell_{y_i},\, u_{x_i} y_i + u_{y_i} x_i - u_{x_i} u_{y_i}\right\}, \quad i = 1, \ldots, n.
\end{aligned}
\tag{5}
$$

Here, the right-hand side of the additional constraints defines the convex envelope of 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 an LP with n additional variables g_i, $i = 1, \ldots, n$, and $2n$ additional constraints.

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

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

$$\begin{array}{lll} \min_{\mathbf{x},\mathbf{z}\in X_3\cap\mathcal{X}_3,\ \mathbf{f}} & \frac{1}{2}\sum_{i\,:\,d_i\geq 0} d_i\left[\mathbf{u}_i^\top\mathbf{x}\right]^2+\frac{1}{2}\sum_{i\,:\,d_i<0} d_i f_i+\mathbf{c}^\top\mathbf{x} & \\ & f_i\leq(\ell_{z_i}+u_{z_i})z_i-\ell_{z_i}u_{z_i} & i\,:\,d_i<0, \end{array} \tag{6}$$

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 $\mathbf{Q}$.

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 problems can be employed in this case. In fact, once the problem is formulated as a MILP, there is no need to develop new solution methods: any of the existing methods, implemented in the best known commercial solvers, like CPLEX and GUROBI, can be employed to solve them. Nevertheless, one can improve the performance of these solvers by improving the input model. For instance, in the MILP reformulation complementarity conditions are translated into big-M constraints. Thus, strengthening the upper bound values used in these constraints may have a relevant impact on the computing times. We refer to [9,19] for the discussion of MILP reformulations and their application to QP problems.

3 Branching

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

Spatial branching: the subset of the feasible region associated with a node of the B&B tree is subdivided into two subsets, obtained by: i) selecting a variable, say x_i; (ii) selecting a reference value for that variable, say x_i^*; (iii) defining the first subset by adding constraint $x_i \leq x_i^*$, and the second subset by adding constraint $x_i \geq x_i^*$. In this case the two subsets are not disjoint but share a common face, where the selected variable is equal to the reference value. By spatial branching it is usually only possible to guarantee that in a finite number of iterations the globally optimal solution is reached within a given precision $\varepsilon > 0$;

KKT branching: this is strictly associated to the KKT reformulation, where a node of the B&B tree is split into two child nodes, by first selecting one of the complementarity conditions and then in each child node imposing that one of the two (linear) factors appearing in a complementarity condition is equal to 0. In case of the MILP reformulation this is obtained by fixing one binary variable to 0 in

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

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

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

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

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

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

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

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

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.

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

4 Domain reduction/Bound tightening

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

$$\begin{aligned} \min / \max_{\mathbf{x},\mathbf{z} \in X_3 \cap \mathcal{X}_3,\, \mathbf{f}} \quad & z_k \\ & f_i \leq (\ell_{z_i} + u_{z_i}) z_i - \ell_{z_i} u_{z_i} \qquad i : d_i < 0, \\ & \tfrac{1}{2} \textstyle\sum_{i : d_i \geq 0} d_i \left[\mathbf{u}_i^\top \mathbf{x}\right]^2 + \tfrac{1}{2} \textstyle\sum_{i : d_i < 0} d_i f_i + \mathbf{c}^\top \mathbf{x} \leq GUB. \end{aligned} \tag{10}$$

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

5 Merits and limits of different approaches

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

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

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

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

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

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

6 Computational experiments

6.1 Setup of the experiments

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

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

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

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

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

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

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

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

- `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 computing times (in seconds) over the x-axis and the fraction of problems solved along the y-axis. It can be seen that the curve corresponding to `B&T(Mix)` is initially below those of CPLEX and GUROBI, but then it gets above them. More precisely, `B&T(Mix)` is able to solve all but one instance within 30 s (and the remaining one in approximately 330 s), while all other approaches are unable to solve at least two instances within the time limit. Thus, `B&T(Mix)` is not always the best performing approach, but it appears to scale better than the other approaches with respect to the dimension, and to be the most robust approach. We remark that at https://github.com/gliuzzi/QPL it is possible to download four additional figures, with the same information reported in Figure 1, but with the instances separated according to the four tested dimensions $n = 20, 30, 40, 50$.

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

Table 1 CPU times (in seconds) over the set of RandQP instances with dimension $n = 30, 40, 50$, for solvers 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**	–	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	–	**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**

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

Table 1 continued

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

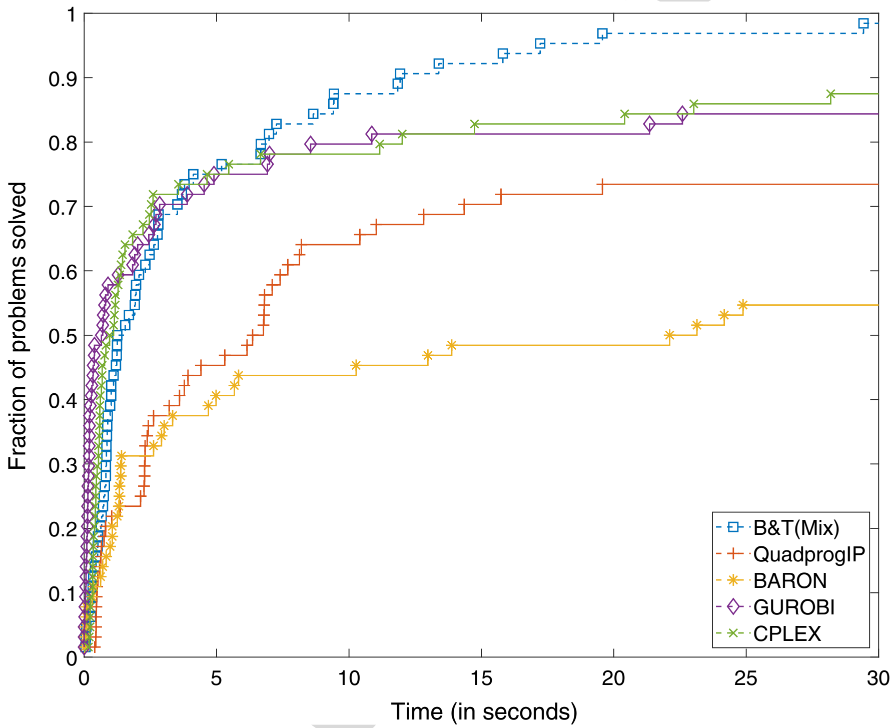

Fig. 1 Fraction of problems solved (y-axis) versus computing time (x-axis) for the different tested solvers

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

7 Conclusions

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

References

1. 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)
2. 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)
3. Caprara, A., Locatelli, M.: Global optimization problems and domain reduction strategies. Math. 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)
7. 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)
9. 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
10. 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)
11. Liuzzi, G., Locatelli, M., Piccialli, V.: A new branch-and-bound algorithm for standard quadratic programming problems. Optim. Methods Softw. **34**, 79–97 (2019)
12. 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 Applications, to appear, available at (2021) http://www.optimization-online.org/DB_FILE/2020/03/7656.pdf
13. McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: part I-convex underestimating problems. Math. Program. **10**, 147–175 (1976)
14. Motzkin, T.S., Straus, E.G.: Maxima for graphs and a new proof of a theorem of Turán. Canadian J. Math. **17**(4), 533–540 (1965)
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)

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

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

Journal: 11590 Article: 1846

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.	
2.	Please confirm if the inserted city name is correct. Amend if necessary.	