Math. Prog. Comp. (2018) 10:333–382
https://doi.org/10.1007/s12532-018-0133-x

FULL LENGTH PAPER

Globally solving nonconvex quadratic programming problems with box constraints via integer programming methods

Pierre Bonami[1] · **Oktay Günlük**[2] · **Jeff Linderoth**[3]

Received: 15 June 2016 / Accepted: 2 December 2017 / Published online: 24 February 2018

Abstract We present effective linear programming based computational techniques for solving nonconvex quadratic programs with box constraints (BoxQP). We first observe that known cutting planes obtained from the Boolean Quadric Polytope (BQP) are computationally effective at reducing the optimality gap of BoxQP. We next show that the Chvátal–Gomory closure of the BQP is given by the odd-cycle inequalities even when the underlying graph is not complete. By using these cutting planes in a spatial branch-and-cut framework, together with a common integrality-based preprocessing technique and a particular convex quadratic relaxation, we develop a solver that can effectively solve a well-known family of test instances. Our linear programming based solver is competitive with SDP-based state of the art solvers on small instances and sparse instances. Most of our computational techniques have been implemented in the

The work of author Linderoth is supported in part by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under Contract Number AC02-06CH11357. IBM Research is also gratefully acknowledged for creating the vibrant research environment where most of this work was accomplished.

✉ Jeff Linderoth
linderoth@wisc.edu

Pierre Bonami
pierre.bonami@es.ibm.com

Oktay Günlük
gunluk@us.ibm.com

[1] IBM Spain, Madrid, Spain

[2] IBM Research, Yorktown Heights, NY, USA

[3] Department of Industrial and Systems Engineering, Wisconsin Institutes of Discovery, University of Wisconsin-Madison, Madison, WI, USA

recent version of CPLEX and have led to significant performance improvements on nonconvex quadratic programs with linear constraints.

Keywords Nonconvex quadratic programming · Global optimization · Boolean Quadric Polytope

Mathematics Subject Classification 90C20 · 90C26 · 90C57

1 Introduction and preliminaries

We consider the non-convex quadratic programming problem with box constraints

$$\text{BoxQP:} \qquad \begin{aligned} z_{\text{BOXQP}} := \min \quad & \frac{1}{2} x^T Q x + c^T x \\ \text{s.t.} \quad & u \geq x \geq \ell, \end{aligned} \tag{1}$$

where $Q \in \mathbb{R}^{n \times n}$ is a symmetric matrix. We assume that the bounds on x are finite, so we may without loss of generality assume that $\ell = \mathbf{0}$, $u = \mathbf{1}$. BoxQP is arguably the simplest non-convex mathematical optimization problem and yet instances with more than 100 variables seem to be computationally challenging for state of the art solvers.

As the objective function $q(x) := \frac{1}{2} x^T Q x + c^T x$ is not necessarily convex, a locally optimal solution to BoxQP may not be globally optimal. Consequently, solution methods for BoxQP typically combine convex relaxations of $q(x)$ with branch-and-bound to find globally optimal solutions.

A standard mechanism for obtaining a convex relaxation of BoxQP begins with the construction of an extended formulation. Let N denote the set $\{1, 2, \ldots, n\}$ and let

$$E := \big\{\{i, j\} \in N \times N \mid i \neq j \text{ and } Q_{ij} \neq 0\big\}$$

consist of the indices of the off-diagonal non-zero entries in Q. Consider the optimization problem

$$\text{EBoxQP:} \qquad \min \quad \sum_{\{i,j\} \in E} Q_{ij} X_{ij} + \frac{1}{2} \sum_{i \in N} Q_{ii} Y_i + \sum_{i \in N} c_i x_i \tag{2}$$

$$\text{s.t.} \quad Y_i = x_i^2 \quad \forall i \in N \tag{3}$$

$$X_{ij} = x_i x_j \quad \forall \{i, j\} \in E \tag{4}$$

$$1 \geq x_i \geq 0 \quad \forall i \in N. \tag{5}$$

Clearly, if (x^*, X^*, Y^*) is a feasible (optimal) solution to EBoxQP, then x^* is a feasible (optimal) solution to BoxQP, and vice versa. As the objective function (2) is linear, this optimization problem can also be solved by optimizing the objective function over the convex hull of feasible solutions of the set

$$\mathcal{Q} := \left\{(x, X, Y) \in \mathbb{R}^n \times \mathbb{R}^{|E|} \times \mathbb{R}^n \mid (3) - (5)\right\}.$$

It is not surprising, then, that previous authors have studied properties of $\mathbf{conv}(\mathcal{Q})$, specifically in the case where the set E consists of all pairs of distinct elements of N, see [4,15,22].

1.1 The McCormick relaxation of $\mathcal{Q}$

As the optimization problem BoxQP is NP-Hard [25], it should be difficult to obtain a complete linear description of the set $\mathbf{conv}(\mathcal{Q})$. Instead, the extended formulation EBoxQP is typically solved by a (spatial) branch-and-bound method that uses a convex outer-approximation of $\mathcal{Q}$. A commonly used relaxation is the so-called *McCormick relaxation* [27] which is obtained by replacing the nonlinear inequalities (3)–(4) with the *McCormick inequalities*, resulting in the set

$$\mathcal{M} := \Big\{(x, X, Y) \in \mathbb{R}^n \times \mathbb{R}^{|E|} \times \mathbb{R}^n \mid \begin{array}{l} x_i \geq Y_i \geq 2x_i - 1, \quad Y_i \geq 0 \quad \forall i \in N, \\ x_i \geq X_{ij}, \quad X_{ij} \geq 0 \quad \forall \{i, j\} \in E, \\ x_j \geq X_{ij} \geq x_i + x_j - 1 \quad \forall \{i, j\} \in E, \end{array} \Big\}.$$

Clearly $\mathcal{Q} \subseteq \mathcal{M}$, and consequently, the linear program

$$z_{\mathcal{M}} := \min \Big\{ \sum_{\{i,j\} \in E} Q_{ij} X_{ij} + \frac{1}{2} \sum_{i \in N} Q_{ii} Y_i + \sum_{i \in N} c_i x_i \mid (x, X, Y) \in \mathcal{M} \Big\}$$

gives a relaxation of BoxQP, $z_{\text{BOXQP}} \geq z_{\mathcal{M}}$.

1.2 SDP relaxations

A different extended formulation of BoxQP, first proposed by Shor [34], uses a symmetric matrix Z of variables defined as $Z = xx^T$:

$$\text{E2BoxQP:} \qquad \min \left\{ \frac{1}{2} Q \cdot Z + c^T x \mid Z = xx^T, 1 \geq x_i \geq 0 \ \forall i \in N \right\}, \quad (6)$$

where $Q \cdot Z$ denotes the standard inner product between the two matrices. Note that the only difference between E2BoxQP and EBoxQP is that EBoxQP has a new variable only if the corresponding entry of Q is non-zero, whereas E2BoxQP has a new variable regardless. Consequently, the *dense* extended formulation E2BoxQP has $n + n^2$ variables, whereas the *sparse* extended formulation EBoxQP has $n + |E|$ variables. Also note that the matrix equation $Z = xx^T$ implies that $Z_{ji} = Z_{ij}$ for all $n \geq i > j \geq 1$.

For a square matrix Y, let $Y \succeq 0$ denote that Y is a symmetric, positive semidefinite matrix. Then, the matrix equation $Z = xx^T$ can be relaxed to

$$Z - xx^T \succeq 0,$$

which in turn is equivalent to the matrix inequality

$$W(x, Z) := \begin{bmatrix} 1 & x^T \\ x & Z \end{bmatrix} \succeq 0.$$

Therefore the following (convex) set

$$\mathcal{S} = \{(x, Z) \in \mathbb{R}^n \times \mathbb{R}^{n\times n} \mid W(x, Z) \succeq 0, 1 \geq x_i \geq 0 \;\; \forall i \in N\}$$

contains $\mathcal{Q}$ and consequently the *semidefinite program*

$$z_{\mathcal{S}} := \{\min \frac{1}{2}\, Q \cdot Z + c^T x \mid W(x, Z) \succeq 0, 1 \geq x_i \geq 0 \;\; \forall i \in N\}$$

gives a relaxation of BoxQP, $z_{\text{BOXQP}} \geq z_{\mathcal{S}}$.

Stronger relaxations of BoxQP that use semidefinite programming have also been proposed. In particular, combining $\mathcal{S}$ with the McCormick inequalities added for each element of the matrix variable Z leads to very strong bounds [5]. Two other variants of this approach that approximately project the set $\mathcal{S}$ into the space of x variables using convex quadratic cutting planes [31] or linear cutting planes [21] have also been developed.

Some of the strongest relaxations can be obtained by leveraging Burer's reformulation of non-convex MIQPs as optimization problems over the cone of copositive matrices. The *doubly nonnegative* relaxation of the copositive reformulation relaxes the copositivity requirement to the intersection of the cones of positive semi-definite matrices and element-wise nonnegative matrices. The doubly nonnegative relaxation has the property that it implicitly includes all McCormick inequalities (see [4]), and is related to the Reformulation Linearization Technique proposed in [33].

As semidefinite programs can be solved using state of the art solvers such as Mosek [37], SeDuMi [36], CVXOPT [2], or SDPLR [16], these relaxations have been successfully used in practice to obtain tight bounds for z_{BOXQP}. Combining these SDP approaches with enumeration has led to the development of several algorithms to solve BoxQP to optimality [14,17]. In Sect. 5.1, we will review several bounds that can be used for SDP and compare them with the bounds obtained using our approach.

1.3 Branching schemes

Neither the McCormick relaxation nor the semidefinite programming relaxation are sufficient to obtain globally optimal solutions to BoxQP. The relaxation schemes must be embedded in a branch-and-bound algorithm. Two different branching schemes are typically used to solve BoxQP. The first scheme is *spatial branching*, which is often employed with the McCormick relaxation. The the solution space is partitioned by branching on a variable x_i, and the associated McCormick inequalities are updated with the new bounds to obtain better convex relaxations of $q(x)$. Spatial branching is the method used by CPLEX to solve BoxQP and is also employed to solve more general classes of nonconvex optimization problems [28,33,39].

The second branching scheme is *KKT branching*, which is based on the first-order KKT conditions satisfied by local optima of BoxQP. The convex relaxation solved in methods that employ KKT branching is formed using both the primal and dual space of BoxQP. Branching is performed to enforce the complementarity conditions between prial and dual variables [17,40]. KKT branching is only applied in the literature to non-convex QPs and can be seen as less general than spatial branching. However, KKT branching has a significant advantage over spatial branching as there are a finite number of branching disjunctions in KKT branching, so the algorithm is finite. In spatial branching, typically only finiteness to ϵ-optimal and feasible solutions can be ensured. The current state of the art solvers for BoxQP combine KKT branching with polyhedral and SDP relaxations [14], such as those derived from the copositive reformulation. Also see [1] for a different approach for solving BoxQP based on the difference of convex functions.

In Sect. 5.2, we will present a comparison using state of the art solvers that use both types of relaxations and branching techniques.

1.4 Our contribution

In the next section we describe the Boolean Quadric Polytope (BQP) and, following [15], discuss how it can be used to obtain relaxations and restrictions of BoxQP. We then computationally show that even though the McCormick relaxation gives a weak relaxation to BoxQP, augmenting it with linear valid inequalities can result in a very strong relaxation. The relaxation remains quite strong even if the only valid inequalities used to improve the formulation are the well-known Chvátal–Gomory cuts for BQP.

In Sect. 3 we study 0-$\frac{1}{2}$ Chvátal–Gomory cuts for the BQP, and establish that the well-known odd cycle inequalities dominate 0-$\frac{1}{2}$ Chvátal–Gomory cuts for BQP. We then show that all Chvátal–Gomory cuts for BQP are 0-$\frac{1}{2}$ Chvátal–Gomory cuts, therefore establishing that the odd cycle inequalities are the only non-dominated Chvátal–Gomory cuts for BQP. The computational insight we gain from this observation is that the odd cycle inequalities for the BQP can be quite strong and therefore can lead to strong relaxations of BoxQP. Furthermore, the technology already implemented in CPLEX for separating 0-$\frac{1}{2}$ Chvátal–Gomory cuts for BQP allows to easily obtain these strong relaxations.

In Sect. 4, we describe a spatial branch-and-cut algorithm for BQP that solves the problem to optimality. The relaxations that we solve in our algorithm have a separable, convex quadratic objective function and linear constraints. We also employ known computational techniques to help to solve the problem more efficiently such as introducing integrality requirements for some of the variables and strengthening the cutting planes using local bound information.

Finally in Sect. 5, we put all these ingredients together to develop a solver that is orders of magnitude better than the previous version of the CPLEX solver for BoxQP instances. These improvements, when implemented in CPLEX, make it one of the fastest solvers on this problem class. However, for large and dense instances in our test suite, solvers based on SDP bounds still dominate our solver. We also note that most techniques we derive are applicable for a more general class of problems where the variables may be subject to additional linear or quadratic constraints. We present computational experiments with CPLEX that show our technique has a more limited but still beneficial effect on these more general problems.

2 Box QP and the Boolean Quadric Polytope

In this section we consider the relationship between BoxQP and the Boolean Quadric Polytope,

$$\text{BQP}^+ := \textbf{conv}\left\{(x, X^+) \in \{0,1\}^{n+|E^+|} \mid X^+_{ij} = x_i x_j \quad \forall \{i,j\} \in E^+\right\},$$

where

$$E^+ := \left\{\{i,j\} \in N \times N \mid i \neq j\right\}$$

denotes the set of all pairs of distinct elements of N. The Boolean Quadric Polytope was introduced in [29] and its polyhedral structure has been extensively studied [10,11,15, 41]. The set BQP^+ is closely related to the Cut Polytope [8], as it can be obtained from the Cut Polytope via an affine transformation [35]. Consequently, valid inequalities for the Cut Polytope readily give valid inequalities for the Boolean Quadric Polytope. We will use the set BQP^+ as a basis for deriving lower and upper bounds for the optimal solution value z_{BOXQP}.

Consider the set of feasible solutions to the extended formulation E2BoxQP written in vector notation:

$$\mathcal{Q}^+ := \Big\{(x, X^+, Y) \in \mathbb{R}^n \times \mathbb{R}^{|E^+|} \times \mathbb{R}^n \mid X^+_{ij} = x_i x_j \ \forall \{i,j\} \in E^+; \\ Y_i = x_i^2,\ 1 \geq x_i \geq 0\ \forall i \in N\Big\}.$$

Clearly any point $(x, X, Y) \in \mathcal{Q}$ can be extended to a unique point $(x, X^+, Y) \in \mathcal{Q}^+$ by letting $X^+_{ij} = X_{ij}$ for $\{i,j\} \in E$ and letting $X^+_{ij} = x_i x_j$ for $\{i,j\} \in E^+ \setminus E$. Furthermore, the reverse is also true and therefore there is a one-to-one correspondence between the members of $\mathcal{Q}$ and $\mathcal{Q}^+$. Consequently, $\mathcal{Q}$ can be obtained from $\mathcal{Q}^+$ by projecting out the variables X^+_{ij} for all $\{i,j\} \in E^+ \setminus E$.

Now consider the set

$$\mathcal{B}^+ := \left\{(x, X^+) \in \mathbb{R}^n \times \mathbb{R}^{|E^+|} \mid X^+_{ij} = x_i x_j \quad \forall \{i,j\} \in E^+, 1 \geq x_i \geq 0 \quad \forall i \in N\right\}$$

obtained from $\mathcal{Q}^+$ by projecting out the Y_i variables for all $i \in N$. The set $\mathcal{B}^+$ is closely related to the Boolean Quadric Polytope as

$$\text{BQP}^+ = \textbf{conv}\left(\mathcal{B}^+ \cap (\{0,1\}^n \times \mathbb{R}^{|E^+|})\right).$$

Furthermore, Burer and Letchford [15] show an interesting and important relationship between the sets $\mathcal{B}^+$ and BQP^+. More precisely, they observe that the convex hull of $\mathcal{B}^+$ is precisely the Boolean Quadric Polytope. The theorem states that projecting out the Y variables from the convex hull of $\mathcal{Q}^+$, one obtains the Boolean Quadric Polytope, an object whose extreme points are all integer-valued.

Theorem 1 (Burer and Letchford [15])

$$\text{BQP}^+ = \mathbf{conv}(\mathcal{B}^+) = \mathbf{proj}_{(x,X^+)}(\mathbf{conv}(\mathcal{Q}^+)).$$

This observation can be easily extended to the case of the sparse BoxQP set $\mathcal{Q}$, its projection $\mathcal{B}$, and the sparse Boolean Quadric Polytope BQP defined by simply replacing E^+ in the definition of BQP^+ with E. The proof is based on the observation that any non integral point in the projection of $\mathcal{Q}$ can be written as a convex combination of two points both of which have at least one more integral component than the original point. Therefore, a point can be extreme (in the sense that it cannot be written as a convex combination of other points) for the projection only if it is integral. This implies that one precisely obtains BQP after projecting out the variables Y_i, that stand for x_i^2, for $i \in N$ from $\mathbf{conv}(\mathcal{Q})$.

Corollary 1 $\text{BQP} = \mathbf{conv}(\mathcal{B}) = \mathbf{proj}_{(x,X)}(\mathbf{conv}(\mathcal{Q}))$.

This observation leads to what we call the BQP relaxation and BQP restriction of BoxQP that we describe next.

2.1 The BQP relaxation of BoxQP

The preceding discussion suggests a relaxation of EBoxQP obtained by first relaxing its objective function (2)

$$\sum_{\{i,j\}\in E} Q_{ij}X_{ij} + \frac{1}{2}\sum_{i\in N} Q_{ii}Y_i + \sum_{i\in N} c_i x_i$$

to eliminate the terms with the Y variables and then relaxing the feasible region $\mathcal{Q}$ by replacing it with $\mathcal{B} \times \mathbb{R}^n$. Note that for any number $a \in [0, 1]$ we have $a \geq a^2 \geq 0$ and consequently, for all $(x, X, Y) \in \mathcal{Q}$ we have $x_i \geq Y_i \geq 0$ for all $i \in N$. Define the sets $N^+, N^- \subset N$ as $N^- := \{i \in N \mid Q_{ii} < 0\}$ and $N^+ := N \setminus N^-$. Clearly, for $(x, X, Y) \in \mathcal{Q}$ we have

$$\sum_{i\in N} Q_{ii}Y_i \geq \sum_{i\in N^-} Q_{ii}Y_i \geq \sum_{i\in N^-} Q_{ii}x_i.$$

Consequently, for $(x, X, Y) \in \mathcal{Q}$ we observe that

$$\sum_{\{i,j\}\in E} Q_{ij}X_{ij} + \frac{1}{2}\sum_{i\in N} Q_{ii}Y_i + \sum_{i\in N} c_i x_i \geq \sum_{\{i,j\}\in E} Q_{ij}X_{ij} + \sum_{i\in N} \bar{c}_i x_i$$

where $\bar{c}$ is obtained from c by increasing c_i by $\frac{1}{2}Q_{ii}$ for $i \in N^-$. Furthermore, as we have

$$\mathcal{Q} \subseteq \mathbf{conv}(\mathcal{Q}) \subseteq \mathbf{conv}(\mathcal{B}) \times \mathbb{R}^n = \text{BQP} \times \mathbb{R}^n,$$

by Corollary 1, we can write the following relaxation of BoxQP.

$$\begin{aligned} z^L_{\text{BQP}} = \min \quad & \sum_{\{i,j\}\in E} Q_{ij}X_{ij} + \sum_{i\in N} \bar{c}_i x_i \\ s.t. \quad & (x, X, Y) \in \text{BQP} \times \mathbb{R}^n. \end{aligned}$$

Using the well-known linear description of the Boolean Quadric Polytope, this is the same as

$$z^L_{\text{BQP}} = \min \quad \sum_{\{i,j\}\in E} Q_{ij}X_{ij} + \sum_{i\in N} \bar{c}_i x_i \tag{7}$$

$$s.t. \quad \min\{x_i, x_j\} \geq X_{ij} \geq \max\{0, x_i + x_j - 1\} \quad \forall \{i, j\} \in E \tag{8}$$

$$x_i \in \{0, 1\} \quad \forall i \in N \tag{9}$$

Solving this mixed integer linear program, therefore, provides a valid lower bound on the optimal solution value of the BoxQP:

$$z_{\text{BOXQP}} \geq z^L_{\text{BQP}}.$$

2.2 The BQP restriction of BoxQP

This time consider a restriction of BoxQP obtained by requiring the solutions to be integral. In other words, consider

$$\begin{aligned} \min \quad & \frac{1}{2} x^T Q x + c^T x \\ \text{s.t.} \quad & x_i \in \{0, 1\} \end{aligned}$$

and note that $x_i^2 = x_i$ when $x_i \in \{0, 1\}$. Consequently, an extended formulation of this problem is

$$\begin{aligned} \min \quad & \sum_{\{i,j\}\in E} Q_{ij}X_{ij} + \sum_{i\in N} (\frac{1}{2}Q_{ii} + c_i)x_i \\ \text{s.t.} \quad & (x, X) \in \text{BQP}. \end{aligned}$$

Using the linear description of BQP, this is the same as

$$
\begin{aligned}
z^U_{\text{BQP}} = \min \quad & \sum_{\{i,j\}\in E} Q_{ij} X_{ij} + \sum_{i\in N} (\frac{1}{2} Q_{ii} + c_i) x_i \\
s.t. \quad & \min\{x_i, x_j\} \geq X_{ij} \geq \max\{0, x_i + x_j - 1\} \quad \forall \{i, j\} \in E \\
& x_i \in \{0, 1\} \quad \forall i \in N.
\end{aligned}
$$

As this mixed integer linear program is a restriction of the BoxQP , it provides a valid upper bound on the optimal solution value:

$$z^U_{\text{BQP}} \geq z_{\text{BOXQP}}.$$

2.3 Computational experiments with BQP relaxation and restriction of BoxQP

To measure the quality of the bounds z^L_{BQP} and z^U_{BQP} empirically, we performed computational experiments on some well known problem instances. Our test set consists of all 99 instances studied in the papers [13,17,40]. For each instance, we solve the McCormick relaxation to obtain the lower bound $z_{\mathcal{M}}$, the BQP Relaxation to obtain the lower bound z^L_{BQP}, and finally the BQP Restriction to obtain the upper bound z^U_{BQP}. The optimal solution value to the Box QP problem, z_{BOXQP}, is taken from literature. For this experiment, Cplex v12.6.1 was used to solve all instances.

We divided the problem instances into four groups depending on the size of the instance. We call an instance *small* if $n \in \{20, 30, 40\}$, *medium* if $n \in \{50, 60, 70\}$, *large* if $n \in \{80, 90\}$, and *jumbo* if $n \in \{100, 125\}$. In addition, we further divided the instances within each group depending on how dense the Q matrix is. We call an instance *low density* if percentage of non-zeroes in the Q matrix is at most 40%, *high density* if the percentage is more than 60%, and *medium density* if it is between 40 and 60%. This categorization leads to 12 subgroups, each subgroup containing instances with similar sizes and densities.

In Table 1, we present statistics for each group of test instances. The first column gives the number of instances in the group. The second column gives the optimality gap of the McCormick relaxation, defined as,

$$\frac{z_{\text{BOXQP}} - z_{\mathcal{M}}}{z_{\mathcal{M}}} \times 100. \tag{10}$$

In the remaining three columns we report on the quality of the relaxations and restrictions by measuring how much of the remaining optimality gap they close. More precisely, we report

$$\frac{z - z_{\mathcal{M}}}{z_{\text{BOXQP}} - z_{\mathcal{M}}} \times 100 \tag{11}$$

where we respectively use z^L_{BQP} and z^U_{BQP} in place of z in the last two columns. As z^U_{BQP} is an upper bound, the statistics reported in the last column are all greater than 100.

Table 1 Boolean quadric relaxation bounds

Size	Density	#	MC gap	BQP relax. root	BQP relax.	BQP-restrict.
Small	Low	6	35.49	90.34	90.48	100.00
	Medium	9	59.93	90.12	90.24	100.08
	High	27	78.96	89.30	89.69	100.03
Medium	Low	12	47.37	94.88	94.88	100.03
	Medium	6	108.81	93.80	95.66	100.02
	High	3	163.47	91.55	96.74	100.02
Large	Low	6	68.65	95.60	96.92	100.06
	Medium	6	124.88	94.26	97.32	100.00
	High	6	180.85	89.10	96.22	100.00
Jumbo	Low	6	93.91	94.30	97.87	100.01
	Medium	6	170.78	89.89	94.36	100.07
	High	6	232.44	84.89	88.71	100.39

In the third column, we use the lower bound obtained by Cplex at the root node while solving the BQP Relaxation to obtain the lower bound z^L_{BQP}. All numbers in the table are obtained by taking the geometric mean across the instances in the group. We give the numeric value of the bounds for all instances in Table 8 in the appendix Sect. A.2.

To summarize the results of this computational experiment, we first note that the McCormick relaxation is very poor in that especially for larger and denser instances it significantly underestimates the optimal value. Note that a value of 100 in the second column indicates that $z_{\mathcal{M}} = \frac{1}{2}\, z_{\text{BOXQP}}$. Next, we note that the BQP restriction, reported in the last column, gives extremely good upper bounds for the problem, consistently within 0.1% of the optimal solution. The quality of the BQP restriction clearly depends on the instance at hand and one can construct examples where the restriction gives bad upper bounds. We note that in the test instances we consider, roughly half of the eigenvalues of the cost matrix are negative and the rest positive. Similarly, half the diagonal entries are positive and the rest negative. We did not observe a corelation between the number of negative eigenvalues or diagonal entries and the quality of the bound given by the BQP restriction.

In terms of the relaxations, the second to last column shows that the BQP relaxation provides a very good approximation of the optimal solution of the BoxQP, almost always within 10% of the optimal value. Finally, we note that the root bound of the BQP relaxation, obtained by adding cutting planes to the linear relaxation gives surprisingly good lower bounds as well. More than 90% of the gap closed by the BQP relaxation can be realized by adding cutting planes to the LP relaxation of the BQP relaxation. In solving all the BQP relaxations, more than 126,194 cutting planes were applied by CPLEX, and 121,222 ($\geq$ 96%) of the inequalities applied were 0-$\frac{1}{2}$ Chvátal–Gomory (CG) cuts. There were a small number of lift-and-project and Gomory Mixed Integer cuts applied as well. In the next section, we build upon this computational insight and precisely describe the relationship between 0-$\frac{1}{2}$ CG cuts obtained from the linear programming relaxation of BQP and a well-known family of facets of BQP.

3 The Chvátal–Gomory closure of BQP

In this section we study valid inequalities for the BQP and show that the Chvátal–Gomory Closure of BQP is given by the so-called odd cycle inequalities. This is an extension of the earlier work [10], which showed that in the case $E = E^+$, the well-known triangle inequalities give the Chvátal–Gomory Closure of BQP^+.

3.1 Chvátal–Gomory cuts

Consider the feasible set of solutions to a generic integer program $P^I = P^{LP} \cap \mathbb{Z}^n$ where

$$P^{LP} = \{x \in \mathbb{R}^n \mid Ax \geq b\}$$

and the matrix A has m rows. For any non-negative vector $\alpha \in \mathbb{R}^m_+$, the inequality $\alpha^T Ax \geq \alpha^T b$ is satisfied by all feasible solutions of P^{LP}. Furthermore, if $\alpha^T A \in \mathbb{Z}^n$ then the strengthened inequality

$$\alpha^T Ax \geq \lceil \alpha^T b \rceil \tag{12}$$

is also satisfied by all feasible solutions of P^I. This inequality is called a *Gomory fractional cut* [23], or, *Chvátal–Gomory cut* [19]. In the special case when $\alpha \in \{0, \frac{1}{2}\}^m$, Inequality (12) is called a 0-$\frac{1}{2}$*cut* [18]. If a Chvátal–Gomory (or, 0-$\frac{1}{2}$) cut can be expressed as a positive combination of a valid inequality for P^{LP} and another Chvátal–Gomory (or, 0-$\frac{1}{2}$) cut, then we say that the first cut is *dominated* by the second. A cut is called *non-dominated*, if there is no cut dominating it. As BQP is full-dimensional [29], it is not possible to have two different inequalities to dominate each other and consequently, adding all non-dominated Chvátal–Gomory (or, 0-$\frac{1}{2}$) cuts to the LP relaxation of BQP has the same effect as adding all possible Chvátal–Gomory (or, 0-$\frac{1}{2}$) cuts. We next describe the well-known odd cycle inequalities for BQP and relate them to 0-$\frac{1}{2}$ cuts.

3.2 Odd cycle inequalities for BQP

Consider the McCormick formulation of $\text{BQP} = \textbf{conv}(\text{BQP}^{LP} \cap \mathbb{Z}^n \times \mathbb{Z}^{|E|})$ where

$$\text{BQP}^{LP} = \left\{(x, X) \in \mathbb{R}^n \times \mathbb{R}^{|E|} \mid \min\{x_i, x_j\} \geq X_{ij} \geq \max\{0, x_i + x_j - 1\} \; \forall \{i, j\} \in E\right\}.$$

Notice that the McCormick inequalities in the description of BQP^{LP} imply that $x_j \geq x_i + x_j - 1$. Consequently, $1 \geq x_i$ for all $i \in N$, and all variables are bounded between zero and one. Also note that the following inequalities are implied by McCormick inequalities for any $\{i, j\} \in E$:

$$2X_{ij} - x_i - x_j \geq -1, \tag{A_{ij}}$$
$$-2X_{ij} + x_i + x_j \geq 0. \tag{B_{ij}}$$

Let $\{i, j\}, \{j, k\}, \{k, i\} \in E$ be given and consider adding up inequalities (A_{ij}), (A_{jk}), and (A_{ki}) associated with these indices and dividing the resulting inequality by 2:

$$X_{ij} + X_{jk} + X_{ki} - x_i - x_j - x_k \geq -\frac{3}{2}.$$

As all variables above are integral, taking the ceiling of the right hand side leads to the 0-$\frac{1}{2}$ cut

$$X_{ij} + X_{jk} + X_{ki} - x_i - x_j - x_k \geq -1. \tag{13}$$

Similarly, combining inequalities (A_{ij}), (B_{jk}), and (B_{ki}) with weights $\frac{1}{2}$ gives

$$X_{ij} - X_{jk} - X_{ki} + x_k \geq 0. \tag{14}$$

Inequalities (13) and (14) are called *triangle inequalities*, see Padberg [29].

Furthermore, combining an odd number of inequalities of type (A_{ij}) with inequalities of type (B_{ij}) yields more general valid inequalities. More precisely, given $E^A, E^B \subseteq E$ such that $|E^A|$ is odd and $E^A \cup E^B$ gives a simple cycle of the graph $G = (N, E)$, let $N^A \subseteq N$ denote the nodes that are incident to exactly two edges in E^A, and let $N^B \subseteq N$ denote the set of nodes that are incident to exactly two edges in E^B. Combining inequalities (A_{ij}) for $\{i, j\} \in E^A$ with inequalities (B_{ij}) for $\{i, j\} \in E^B$ and rounding up the right hand side leads to the valid inequality

$$\sum_{i \in N^B} x_i - \sum_{i \in N^A} x_i - \sum_{\{i,j\} \in E^B} X_{ij} + \sum_{\{i,j\} \in E^A} X_{ij} \geq \left\lceil -\frac{|E^A|}{2} \right\rceil \tag{15}$$

which is called an *odd-cycle* inequality for the BQP [29]. As pointed out by Simone [35] these inequalities are affine transformations of the cycle inequalities for the cut polytope studied earlier by Barahona and Mahjoub [8] who also give a separation algorithm.

It is known that inequality (15) is dominated by another odd cycle inequality unless $E^A \cup E^B$ gives a chordless simple cycle of the graph $G = (N, E)$. Consequently, the only non-dominated odd cycle inequalities for BQP$^+$ are the triangle inequalities as the underlying graph $G = (N, E^+)$ does not have any other chordless cycles. Furthermore, Boros et al. [10] have shown that for BQP$^+$ the triangle cuts dominate, and therefore are equally as strong as, Chvátal–Gomory cuts. In the next subsection, we show that for BQP the odd cycle inequalities dominate, and therefore are equally as strong as, Chvátal–Gomory cuts.

3.3 Chvátal–Gomory closure of BQP

Adding all Chvátal–Gomory cuts (12) to BQP^{LP} for all $\alpha \in \mathbb{R}^m_+$ gives the so-called Chvátal–Gomory closure of BQP^{LP}. Similarly, the 0-$\frac{1}{2}$ closure of BQP^{LP} is obtained by adding all possible 0-$\frac{1}{2}$ cuts to it. We start off with analyzing the 0-$\frac{1}{2}$ cuts for the BQP. Consider the inequality system defining BQP^{LP}

$$\begin{array}{lrl}
(\alpha_i^0) & x_i \geq 0 & \forall i \in N \\
(\alpha_i^1) & -x_i \geq -1 & \forall i \in N \\
(\alpha_{ij}^2) & x_i - X_{ij} \geq 0 & \multirow{2}{*}{$\forall \{i,j\} \in E$} \\
(\alpha_{ij}^3) & x_j - X_{ij} \geq 0 & \\
(\alpha_{ij}^4) & -x_i - x_j + X_{ij} \geq -1 & \forall \{i,j\} \in E \\
(\alpha_{ij}^5) & X_{ij} \geq 0 & \forall \{i,j\} \in E
\end{array}$$

and a vector of multipliers $\alpha \in \{0, \frac{1}{2}\}^{2|N|+4|E|}$. Even though they are implied, we have explicitly included the bound constraints on the x variables in this system as it makes our proof simpler. Also note that the bound constraints are simply obtained by adding the McCormick inequalities and therefore they do not change the Chvátal–Gomory or the 0-$\frac{1}{2}$ closure of BQP^{LP}. Also notice that we are slightly abusing notation here to differentiate between the coefficients associated with the inequalities $x_i - X_{ij} \geq 0$ and $x_j - X_{ij} \geq 0$ for a given $\{i, j\} \in E$. Here we treat $\{i, j\}$ as an ordered pair and associate the multiplier α_{ij}^2 with the smaller index and α_{ij}^3 with the larger one. Combining the constraints above with the corresponding multiplier leads to the following implied inequality for BQP^{LP}:

$$\sum_{\{i,j\}\in E} a_{ij} X_{ij} + \sum_{i \in N} d_i x_i \geq -f$$

where $f = \sum_{i\in N} \alpha_i^1 + \sum_{\{i,j\}\in E} \alpha_{ij}^4$, $a_{ij} = -\alpha_{ij}^2 - \alpha_{ij}^3 + \alpha_{ij}^4 + \alpha_{ij}^5$, and

$$d_i = \alpha_i^0 - \alpha_i^1 + \sum_{\substack{j:\{i,j\}\in E \\ i<j}} \alpha_{ij}^2 + \sum_{\substack{j:\{i,j\}\in E \\ i>j}} \alpha_{ij}^3 - \sum_{j:\{i,j\}\in E} \alpha_{ij}^4.$$

If the vectors a and d are integral, and the right hand side $-f$ is fractional (and therefore is equal to an integer plus $\frac{1}{2}$), then the inequality

$$\sum_{\{i,j\}\in E} a_{ij} X_{ij} + \sum_{i \in N} d_i x_i \geq \lceil -f \rceil \tag{16}$$

is a 0-$\frac{1}{2}$ cut. To argue that if a point $(\bar{x}, \bar{X}) \in \text{BQP}^{LP}$ violates a 0-$\frac{1}{2}$ cut then it also violates an odd cycle inequality, we next show some properties of the multipliers α that lead to non-dominated 0-$\frac{1}{2}$ cuts.

Lemma 1 *Let* (16) *be a non-dominated* 0-$\frac{1}{2}$ *cut, then it can be generated using multipliers that satisfy* $\alpha^2_{ij} = \alpha^3_{ij}$ *and* $\alpha^4_{ij} = \alpha^5_{ij}$ *for all* $\{i, j\} \in E$.

Proof Consider a point $(\bar{x}, \bar{X}) \in \text{BQP}^{LP}$ that violates a non-dominated 0-$\frac{1}{2}$ cut and let α be the coefficient vector associated with the cut. As the cut is not dominated, it is not implied by a combination of another 0-$\frac{1}{2}$ cut and some inequalities defining BQP^{LP}. For any $\{i, j\} \in E$, as a_{ij} is integral, we have $(-\alpha^2_{ij} - \alpha^3_{ij} + \alpha^4_{ij} + \alpha^5_{ij}) \in \{-1, 0, 1\}$.

If $a_{ij} = 1$, we clearly have $\alpha^2_{ij} = \alpha^3_{ij} = 0$ and $\alpha^4_{ij} = \alpha^5_{ij} = \frac{1}{2}$. On the other hand, if $a_{ij} = -1$, we have $\alpha^2_{ij} = \alpha^3_{ij} = \frac{1}{2}$ and $\alpha^4_{ij} = \alpha^5_{ij} = 0$. Therefore, in both cases, $\alpha^2_{ij} = \alpha^3_{ij}$ and $\alpha^4_{ij} = \alpha^5_{ij}$.

Finally, if $a_{ij} = 0$, we either have $\alpha^2_{ij} = \alpha^3_{ij} = \alpha^4_{ij} = \alpha^5_{ij} = \frac{1}{2}$, in which case the base inequality is a combination of the trivial inequality $0 \geq -1$ and another base inequality obtained by setting $\alpha^2_{ij} = \alpha^3_{ij} = \alpha^4_{ij} = \alpha^5_{ij} = 0$. Clearly the 0-$\frac{1}{2}$ cut on hand is dominated by the new 0-$\frac{1}{2}$ cut. When $a_{ij} = 0$, a second possibility is to have $\alpha^2_{ij} + \alpha^3_{ij} = \alpha^4_{ij} + \alpha^5_{ij} = \frac{1}{2}$. Without loss of generality, let $\alpha^2_{ij} = \frac{1}{2}$. In this case, if $\alpha^4_{ij} = \frac{1}{2}$, then the same base inequality can be obtained by setting $\alpha^2_{ij} = \alpha^4_{ij} = 0$ and increasing α^1_j by $\frac{1}{2}$ (note that if α^1_i was already $\frac{1}{2}$, then the cut is improved by setting α^1_{ij} to 0). On the other hand, if $\alpha^5_{ij} = \frac{1}{2}$, the same base inequality can be obtained by setting $\alpha^2_{ij} = \alpha^5_{ij} = 0$ and increasing α^0_i by $\frac{1}{2}$.

Combining these observations, we conclude that any non-dominated 0-$\frac{1}{2}$ cut can be generated using multipliers that satisfy $\alpha^2_{ij} = \alpha^3_{ij}$ and $\alpha^4_{ij} = \alpha^5_{ij}$ for all $\{i, j\} \in E$. □

Therefore, any non-dominated 0-$\frac{1}{2}$ cut for BQP is in fact a 0-$\frac{1}{2}$ cut for the following simple system of inequalities:

$$
\begin{array}{lrl}
(\alpha^0_i) & x_i \geq 0 & \forall i \in N \\
(\alpha^1_i) & -x_i \geq -1 & \forall i \in N \\
(\beta^1_{ij}) & x_i + x_j - 2X_{ij} \geq 0 & \forall \{i, j\} \in E \\
(\beta^2_{ij}) & -x_i - x_j + 2X_{ij} \geq -1 & \forall \{i, j\} \in E
\end{array}
$$

where $\beta^1_{ij} = \alpha^2_{ij} = \alpha^3_{ij}$ and $\beta^2_{ij} = \alpha^4_{ij} = \alpha^5_{ij}$ for $\{i, j\} \in E$. We next show that any non-dominated 0-$\frac{1}{2}$ cut for the above system of inequalities satisfies $\alpha^0_i = \alpha^1_i = 0$. for all $i \in N$.

Lemma 2 *Let* (16) *be a non-dominated* 0-$\frac{1}{2}$ *cut, then* $\alpha^0_i = 0$ *for all* $i \in N$.

Proof Assume that $\alpha^0_i = \frac{1}{2}$ for some $i \in N$. In this case, clearly $\alpha^1_i = 0$, otherwise both α^0_i and α^1_i can be set to zero to obtain a stronger inequality. In addition, as d_i is integral, at least one of β^1_{ij} or β^2_{ij} equals $\frac{1}{2}$ for some $j \in N$ such that $\{i, j\} \in E$.

First assume that $\beta_{ij}^1 = \frac{1}{2}$ for some $\{i, j\} \in E$ and $j \in N$. In this case, the contribution of the associated inequalities to the base inequality is:

$$x_i + \frac{1}{2}x_j - X_{ij} \geq 0.$$

In this case, setting $\alpha_i^0 = \beta_{ij}^1 = 0$ and increasing α_j^0 by $\frac{1}{2}$ reduces the left hand side of the base inequality by $x_i - X_{ij}$ while not changing the right hand side. As $x_i - X_{ij} \geq 0$ is a valid inequality for the BQP^{LP}, the new 0-$\frac{1}{2}$ cut dominates the original one.

Next assume that $\beta_{ij}^2 = \frac{1}{2}$ for some $\{i, j\} \in E$. In this case, the contribution of the associated inequalities to the base inequality is:

$$-\frac{1}{2}x_j + X_{ij} \geq -\frac{1}{2}.$$

However, setting $\alpha_i^0 = \beta_{ij}^2 = 0$ and increasing α_j^1 by $\frac{1}{2}$ reduces the left hand side of the base inequality by X_{ij}. As $X_{ij} \geq 0$ is a valid inequality for the BQP^{LP}, the new 0-$\frac{1}{2}$ cut again dominates the original one. □

Lemma 3 *Let* (16) *be a non-dominated* 0-$\frac{1}{2}$ *cut, then* $\alpha_i^1 = 0$ *for all* $i \in N$.

Proof Assume that $\alpha_i^1 = \frac{1}{2}$ for some $i \in N$. As d_i is integral, at least one of β_{ij}^1 or β_{ij}^2 equal $\frac{1}{2}$ for some $j \in N$ such that $\{i, j\} \in E$. First assume that $\beta_{ij}^1 = \frac{1}{2}$ for some $\{i, j\} \in E$ and $j \in N$. In this case, the contribution of the associated inequalities to the base inequality is:

$$\frac{1}{2}x_j - X_{ij} \geq -\frac{1}{2}.$$

In this case, setting $\alpha_i^1 = \beta_{ij}^1 = 0$ and increasing α_j^1 by $\frac{1}{2}$ reduces the left hand side of the base inequality by $x_j - X_{ij}$ while keeping the right hand side the same. As $x_j - X_{ij} \geq 0$ is a valid inequality for the BQP^{LP}, the new 0-$\frac{1}{2}$ cut dominates the original one.

Next assume that $\beta_{ij}^2 = \frac{1}{2}$ for some $\{i, j\} \in E$. In this case, the contribution of the associated inequalities to the base inequality is:

$$-x_i - \frac{1}{2}x_j + X_{ij} \geq -1.$$

Setting $\alpha_i^1 = \beta_{ij}^2 = 0$ and increasing α_j^0 by $\frac{1}{2}$ reduces the left hand side of the base inequality by $-x_i - x_j + X_{ij}$ and reduces the right hands side by -1. As $-x_i - x_j + X_{ij} \geq -1$ is a valid inequality for the BQP^{LP}, the new 0-$\frac{1}{2}$ cut again dominates the original one. □

Consequently, any 0-$\frac{1}{2}$ cut for BQP can in fact be derived as a 0-$\frac{1}{2}$ cut for the following system of inequalities:

$$(\beta_{ij}^1) \qquad x_i + x_j - 2X_{ij} \geq 0 \qquad \forall\{i, j\} \in E \qquad (17)$$

$$(\beta_{ij}^2) \qquad -x_i - x_j + 2X_{ij} \geq -1 \qquad \forall\{i, j\} \in E \qquad (18)$$

Furthermore, it is easy to see that coefficients β^1_{ij} and β^2_{ij} associated with $\{i, j\} \in E$ cannot be positive at the same time. We now associate a subgraph $G = (N, E')$ with the cut where $E' \subseteq E$ and $\{i, j\} \in E'$ only if one of β^1_{ij} or β^2_{ij} is positive.

Theorem 2 *All non-dominated* 0-$\frac{1}{2}$ *cuts for the* BQP *are odd cycle inequalities.*

Proof As the coefficients of x variables (denoted by d) in the base inequality are integral, each node of the graph $G = (N, E')$ has even degree and therefore E' can be decomposed into simple cycles $C_1, \ldots, C_q$. Assume that $q > 1$. In this case, the base inequality is a combination of q valid inequalities, each associated with a simple cycle. More precisely, for each cycle C_t, we associate the inequality

$$\sum_{\{i,j\}\in C_t} (\beta^2_{ij} - \beta^1_{ij})2X_{ij} + \sum_{\{i,j\}\in C_t} (\beta^1_{ij} - \beta^2_{ij})(x_i + x_j) \geq - \sum_{\{i,j\}\in C_t} \beta^2_{ij}. \tag{19}$$

Notice that for $\{i, j\} \in C_t$ exactly one of β^1_{ij} or β^2_{ij} is equal to $\frac{1}{2}$ and consequently the coefficient of X_{ij} is either 1 or -1. In addition, as each node in the cycle is incident on exactly two edges, the coefficients of the x variables in (19) are either -1 or 0 or 1. If the right hand side of (19) is also integral, then the 0-$\frac{1}{2}$ cut is a combination of an integral inequality implied by the formulation and another 0-$\frac{1}{2}$ cut obtained by setting the multipliers associated with the cycle C_t to zero. Therefore, the right hand side of (19) has to be fractional if the cut is not dominated. But in this case, adding up the inequalities associated with all cycles except C_t, one obtains an integral implied inequality and again the 0-$\frac{1}{2}$ cut becomes a combination of an integral inequality implied by the formulation and another 0-$\frac{1}{2}$ cut obtained by setting the multipliers associated with the other cycles to zero. Therefore, $q \not> 1$ and consequently, $E' = C_1$. Furthermore, as the right hand side of the base inequality needs to be fractional, E' contains an odd number of edges that have the β^2 coefficients associated with the edge non-zero. □

We next show that odd cycle inequalities give the Chvátal–Gomory closure for the BQP. The proof uses the same approach as [10] which is based on the proof of a related result in [29].

Theorem 3 *All non-dominated Chvátal–Gomory cuts for the* BQP *are* 0-$\frac{1}{2}$ *cuts.*

Proof Let $y = (x, X)$ and consider a non-dominated Chvátal–Gomory (CG) cut $ay \geq f$ for the BQP. By definition, a is an integral (row) vector and $f = \lceil z^* \rceil$ where $z^* = \min\{ay \mid Ay \geq b\}$ and the inequality system $Ay \geq b$ is composed of the McCormick inequalities defining BQP^{LP}. By linear programming theory, there exists a collection of inequalities $By \geq b'$ from among the ones that define BQP^{LP} such that B is a non-singular square matrix and $z^* = \min\{ay \mid By \geq b'\}$. Furthermore, the second problem has an optimal dual solution (row vector) $w^* \geq 0$ such that $w^* B = a$ and $w^* b' = z^*$. In other words, using $w^* = aB^{-1}$ as weights, one can combine the inequalities $By \geq b'$ to obtain $ay \geq f$ as a CG cut. We next argue that B^{-1} is $\frac{1}{2}$-integral and therefore w^* is also $\frac{1}{2}$-integral.

In [29], to prove that all vertices of BQP^{LP} are $\frac{1}{2}$-integral, Padberg shows that after elementary (unimodular) row and column operations, it is possible to transform any basis B of A to a block diagonal matrix with $\{-1, 0, 1\}$ components such that one of the blocks is an identity matrix and the remaining blocks have exactly two non-zero entries per row and column. In other words, $B = UDV$ where U and V are unimodular (and therefore both U^{-1} and V^{-1} are integral) and D is block diagonal. Using an earlier result from [30] on almost totally unimodular matrices, this implies that each diagonal block of D is either the identity matrix, or, has determinant ± 2. Consequently, D^{-1} and therefore $B^{-1} = V^{-1}D^{-1}U^{-1}$ is $\frac{1}{2}$-integral. This implies that $w^* = aB^{-1}$ is also $\frac{1}{2}$-integral. □

3.4 Connections to the cut polytope

We next present a brief discussion to relate the CG cuts for the BQP and BQP^{+}. As we discussed earlier, BQP^{+} can be obtained from the cut polytope (collection of incidence vectors of edges that induce a cut in the graph) via a linear bijection (sometimes called the *covariance mapping*) introduced by Simone [35]. In other words, there is a 1-to-1 correspondence between the integral solutions of the two sets. Furthermore, for the complete graph $G = (V, E^{+})$, it is known that solutions (and facets) to the continuous relaxation of the triangle formulation of the cut polytope and BQP^{LP} strengthened with triangle inequalities also have a 1-to-1 correspondence. In addition, it is also known that the CG closure of BQP^{LP} is obtained by adding triangle cuts to BQP^{LP+} [10]. Finally, Barahona [6] has shown that projecting out some of the edge variables from the triangle formulation of the cut polytope leads to an odd cycle formulation which in turn can be mapped to the BQP^{LP} strengthened with odd cycle cuts. Combining these known facts, Theorem 3 can be seen as a direct proof of the fact that the CG closure of the BQP defined on a sparse graph is equal to the projection of the CG closure of the BQP defined on the complete graph.

3.5 Computational experiments with odd cycle inequalities

Theorem 3 establishes the theoretical strength of odd cycle inequalities by showing that these inequalities give the CG closure of the BQP. To complement this observation, we made some experiments to measure their computational performance on instances obtained as relaxations of BoxQP instances in our test set. As described in Sect. 3.4, it is possible to optimize over the the 0-$\frac{1}{2}$-closure of BQP by optimizing over an extended formulation augmented with triangle inequalities. To compute the optimality gap closed by all odd cycle inequalities, we solve this large (but polynomially-sized) linear program rather than separating exponentially many odd cycle inequalities. We also computed the gap closed by 0-$\frac{1}{2}$ cuts heuristically separated by CPLEX at the root node. Even though it is NP-hard to separate 0-$\frac{1}{2}$ cuts in general [18], odd cycle inequalities can be separated in polynomial time [7,8]. However, instead of implementing a special purpose separation algorithm, we use heuristics that have proved

Table 2 Comparison of CG-closure to other Boolean quadric relaxation bounds

Size	Density	#	MC gap	Cplex 0-$\frac{1}{2}$ cuts	All 0-$\frac{1}{2}$ cuts	BQP relax.
Small	Low	6	35.49	90.34	90.48	90.48
	Medium	9	59.93	90.12	90.24	90.24
	High	27	78.96	89.30	89.45	89.69
Medium	Low	12	47.37	94.88	94.88	94.88
	Medium	6	108.81	93.80	94.52	95.66
	High	3	163.47	91.55	92.00	96.74
Large	Low	6	68.65	95.60	96.71	96.92
	Medium	6	124.88	94.26	95.64	97.32
	High	6	180.85	89.10	89.47	96.22
Jumbo	Low	6	93.91	94.30	95.84	97.87
	Medium	6	170.78	89.89	90.53	94.36
	High	6	232.44	84.89	84.95	88.71

to be quite successful in practice [26], and most MIP solvers have powerful heuristic separation algorithms for this class of inequalities. We note that we tried but did not see any benefit in using inequalities (17)–(18) in place of McCormick inequalities as the base system to generate 0-$\frac{1}{2}$ cuts.

Table 2 below summarizes our findings and detailed results for each instance can be seen in Table 8 in the Appendix. The first two columns of Table 2 as well as the last column are taken from Table 1. The third column gives the average gap closed by 0-$\frac{1}{2}$ cuts separated heuristically. The fourth column, on the other hand, gives the gap closed by all odd cycle inequalities (or, equivalently, all 0-$\frac{1}{2}$ cuts) at the root node. Comparing the third column to the last column , we verify that restricting Cplex to use only 0-$\frac{1}{2}$ cuts at the root node to strengthen the LP relaxation does not degrade the optimality gap closed. This is a complementary observation to [3], who demonstrates that the doubly non-negative relaxation, when augemented with the triangle inequalities, closes all of the gap on small and medium instances, save for one instance. Finally, comparing the third column to the fourth column, we conclude that heuristically separated 0-$\frac{1}{2}$ cuts approximate the effect of all 0-$\frac{1}{2}$ cuts quite well.

4 Algorithmic improvements

In this section we discuss three additional computational techniques that proved very useful in solving BoxQP. First we describe an improved convex relaxation of BoxQP that keeps some quadratic terms in the formulation, we then observe that certain variables may be restricted to take only binary values, and finally we show how to use bound information to strengthen valid inequalities obtained from the Boolean Quadric Polytope.

The prototype implementation, upon which most of our computational results are based, was done using CPLEX v.12.6.1. The version of CPLEX we employed was augmented with the ability to directly call the 0-$\frac{1}{2}$ cut generator available in CPLEX. Cutting planes were computed using this generator and added in the CPLEX cut callback function. Cuts were added iteratively in rounds until the relaxation solution was no longer able to be separated. All other features of the CPLEX default implementation, including sophisticated branching rules, preprocessing, and cut filtering and handling were employed. All experiments in Sect. 4 were performed on a computer with an Intel i5-4570 CPU processor clocked at 3.2 GHz and with 32 GB of main memory.

4.1 Strengthened convex relaxation

A stronger convex relaxation for EBoxQP can be obtained by adapting the McCormick relaxation $\mathcal{M}$ to not relax the convex inequalities $Y_i \geq x_i^2$, resulting in the convex set

$$\begin{aligned}\mathcal{M}^2 := \Big\{(x, X, Y) \in \mathbb{R}^n \times \mathbb{R}^{|E|} \times \mathbb{R}^n \mid\ & x_i \geq Y_i \geq x_i^2,\ Y_i \geq 0 \quad \forall i \in N,\\ & x_i \geq X_{ij},\ x_j \geq X_{ij} \geq x_i + x_j - 1\ \forall \{i, j\} \in E,\\ & X_{ij} \geq 0 \quad \forall \{i, j\} \in E,\ 1 \geq x_i \geq 0 \quad \forall i \in N\Big\}.\end{aligned}$$

This relaxation, first suggested by [32] can be seen as implementing some, but not all, of the strength of the semidefinite-programming based relaxation.

The value of the optimization problem

$$z_{\mathcal{M}^2} := \min\Big\{\sum_{\{i,j\}\in E} Q_{ij}X_{ij} + \sum_{i\in N}\frac{1}{2}\,Q_{ii}Y_i + c^Tx \mid (x, X, Y) \in \mathcal{M}^2\Big\}$$

provides a potentially stronger lower bound than does $z_{\mathcal{M}}$: $z_{\mathcal{M}} \leq z_{\mathcal{M}^2} \leq z_{\text{BOXQP}}$. The set $\mathcal{M}^2$ has n (convex) quadratic constraints in its definition, which may be a disadvantage for computation. However, by projecting out variables from $\mathcal{M}^2$, the computation of $z_{\mathcal{M}^2}$ can be transformed into an optimization problem with a convex quadratic objective and linear constraints [9]. Specifically, recalling that $N^+ := \{i \in N \mid Q_{ii} \geq 0\}$ and $N^- := \{i \in N \mid Q_{ii} < 0\}$, we can easily see that

$$\begin{aligned}z_{\mathcal{M}^2} = \min\quad & \sum_{(i,j)\in E} Q_{ij}X_{ij} + \sum_{i\in N^-}\frac{1}{2}\,Q_{ii}Y_i + \sum_{i\in N^+}\frac{1}{2}\,Q_{ii}x_i^2 + c^Tx\\ \text{s.t. } & x_i \geq Y_i \quad \forall i \in N^-,\\ & x_i \geq X_{ij},\ x_j \geq X_{ij} \geq x_i + x_j - 1 \quad \forall \{i, j\} \in E,\\ & X_{ij} \geq 0 \quad \forall \{i, j\} \in E, \quad 1 \geq x_i \geq 0 \quad \forall i \in N.\end{aligned}$$

This problem can be solved using an active-set method. We use the simplex method for QP available in CPLEX to solve the relaxations arising in the algorithm.

To understand the improvement in bound obtained by using the convex quadratic relaxation $\mathcal{M}^2$ versus the linear programming relaxation $\mathcal{M}$, we compared the percentage of the gap closed at the root by the different relaxations, and Table 3 shows the

Table 3 Bounds improvement with $\mathcal{M}^2$ relaxations

Size	Density	#	% Gap closed			
			$\mathcal{M}^2$	$\mathcal{M}$ + 0-$\frac{1}{2}$	$\mathcal{M}^2$ + 0-$\frac{1}{2}$	$\Delta(\mathcal{M}^2)$
Small	Low	6	4.68	90.34	99.29	8.95
	Medium	9	3.67	90.12	98.58	8.46
	High	27	3.55	89.30	98.64	9.34
Medium	Low	12	2.39	94.88	99.69	4.82
	Medium	6	1.72	93.80	96.83	3.03
	High	3	1.23	91.55	93.04	1.49
Large	Low	6	1.08	95.60	97.81	2.21
	Medium	6	1.11	94.26	95.99	1.73
	High	6	0.97	89.10	90.17	1.07
Jumbo	Low	6	0.96	94.30	95.80	1.50
	Medium	6	0.84	89.89	90.82	0.93
	High	6	0.66	84.89	85.64	0.75

results of this experiment. The table aggregates information about the percentage of the optimality gap closed [defined in Eq. (11)] by the improved McCormick relaxation $\mathcal{M}^2$, the McCormick relaxation with 0-$\frac{1}{2}$ cuts, and the relaxation $\mathcal{M}^2$ with 0-$\frac{1}{2}$ cuts added. The last column shows the difference between the previous two columns and highlights the contribution of the improved McCormick relaxation in the presence of 0-$\frac{1}{2}$ cuts. The Avg MC Gap is the geometric mean computed over all instances in the category, and the % gap closed values are the geometric means of all values. The full results of this experiment, listing the bounds for each instance is given in Table 9 in the appendix Sect. A.3

While the percentage optimality gap closed by using the improved McCormick relaxation $\mathcal{M}^2$ is relatively modest, especially for the largest instances in our test set, the effect is amplified in the presence of zerohalf cuts. We note that the computational overhead incurred by solving the convex quadratic programming relaxation instead of the linear programming relaxation is very modest. This is due to the fact that CPLEX employs a pivoting-based method for obtaining the bound during branch and bound, and consequently, child nodes can be warm-started using the parent solution. Thus, the computational effort to solve branch and bound nodes is comparable, regardless of the relaxation ($\mathcal{M}$ or $\mathcal{M}^2$) being used. Therefore, we conclude that the improvement due to using the convex quadratic programming formulation is clearly worth the additional computational effort.

4.2 Improving spatial branching using implicit integrality

As mentioned earlier, global optimization solvers combine convex relaxations of the problem at hand with enumeration to find globally optimal solutions. For BoxQP, the convex relaxations we use are the McCormick relaxations strengthened with cutting planes. If the relaxation solution $(\hat{x}, \hat{X}, \hat{Y})$ has $\hat{x}_i \hat{x}_j \neq \hat{X}_{ij}$, then the solution is not feasible to BoxQP and the infeasibility must be resolved via *spatial branching*—e.g. creating one child node with the additional constraint $x_i \leq \hat{x}_i$, a second child node with the additional constraint $x_i \geq \hat{x}_i$ and tightening the McCormick relaxations in both nodes according to these bounds. We next observe that if the diagonal matrix element Q_{kk} is not positive for some $k \in N$, then there exists an optimal solution x^* with x_k^* at one of its bounds.

Lemma 4 [24] *If* $Q_{kk} \leq 0$ *for some* $k \in N$*, then there exists an optimal solution* x^* *to BoxQP with* $x_k^* \in \{\ell_k, u_k\}$.

Proof Take any feasible solution $\hat{x}$ to BoxQP . Fixing $x_i = \hat{x}_i$, for all $i \in N \setminus \{k\}$ in the objective function to BoxQP , we obtain the one-dimensional function

$$q_{\hat{x}}(x_k) := \frac{1}{2}\left(Q_{kk}x_k^2 + \sum_{i \in N\setminus\{k\}} Q_{ik}\hat{x}_i x_k\right) + c_k x_k + \underbrace{\frac{1}{2}\left(\sum_{i \in N\setminus\{k\}}\sum_{j \in N\setminus\{k\}} Q_{ij}\hat{x}_i\hat{x}_j + \sum_{i \in N\setminus\{k\}} c_i\hat{x}_i\right)}_{\text{Constant}}.$$

If $Q_{kk} \leq 0$, then $q_{\hat{x}}(x_k)$ is a concave function with a minimum at one of the endpoints of its domain. Fixing variables x_i, $i \neq k$ does not affect the range of values that x_k can take, so $q_{\hat{x}}(x_k)$ must have a minimum at ℓ_k or u_k. So if $\hat{x}$ is such that $\ell_k \leq \hat{x}_k < u_k$, there exists a feasible solution x^* whose objective is no worse with $x_k^* \in \{\ell_k, u_k\}$. □

Lemma 4 can be used to strengthen branching divisions when branching on a variable x_k that has $Q_{kk} \leq 0$. Instead of creating two nodes with bounds $x_k \leq \hat{x}_k$, and $x_k \geq \hat{x}_k$, respectively, one can instead create two nodes with bounds $x_k = \ell_k$, and $x_k = u_k$ and therefore avoid branching on variable x_k again in this branch of the enumeration tree. Assuming the bounds are $0 \leq x_i \leq 1\ \forall i \in N$, this idea can simply be implemented by declaring all such variables as binary variables in a solver. This preprocessing technique was also proposed in [17].

Figures 1 and 2 show the results of an experiment designed to demonstrate the impact of employing these strengthened branching disjunctions. Each of the instances in our test suite was solved both with declaring $x_i \in \{0, 1\}\ \forall i \in N^-$ and without performing this preprocessing step. The instances were solved for at most 3600 CPU seconds using our instrumented version of CPLEX v12.6.1 with the improved $\mathcal{M}^2$ relaxation and 0-$\frac{1}{2}$ cuts. Figure 1 is a scatter plot comparing the number of nodes in the enumeration tree for all instances that were solved by both methods with $x_i \in [0, 1]\ \forall i \in N^-$ as the x component and $x_i \in \{0, 1\}\ \forall i \in N^-$ as the y-component. The plot is on a log − log scale and demonstrates that especially for instances with a larger enumeration tree, declaring variables to be integer-valued can have a significantly beneficial effect. Figure 2 is a similar scatter plot showing the remaining relative optimality gap for all instances that were not solved by both methods. The final relative optimality gap tolerance used in this experiment was 0.01%, and therefore Fig. 2 demonstrates that 9 instances were solved within 1 h when declaring $x_i \in \{0, 1\}$, $i \in N^-$ that were not solved otherwise. The detailed computational performance of the

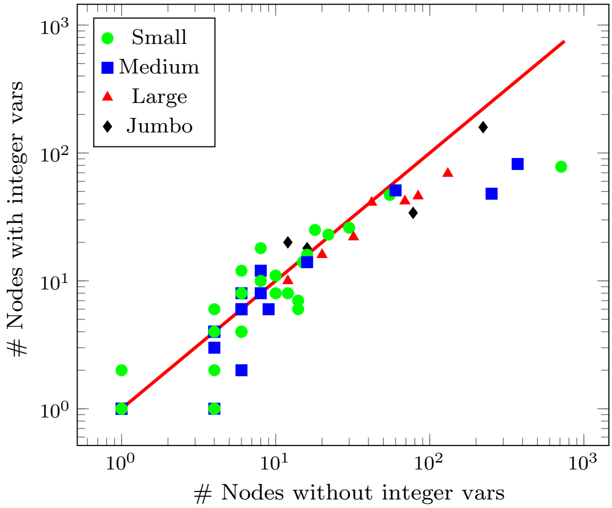

Fig. 1 Comparing # of nodes in search tree for solved instances, with and without declaring appropriate variables integer

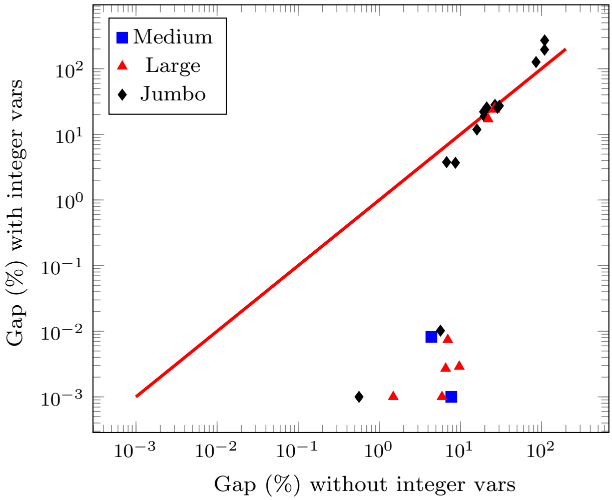

Fig. 2 Comparing final optimality gap for unsolved instances, with and without declaring appropriate variables integer

two methods for each instance in the test suite is given in Table 10 in the appendix. The conclusion from this experiment is that for box-constrained instances, this simple preprocessing technique is definitely worthwhile.

4.3 Cuts at branch and bound nodes

Theorem 1 establishes how inequalities from the Boolean Quadric Polytope BQP may be applied to the box-constrained quadratic program BoxQP. The BQP inequalities are valid as long as the variables have bounds $x_i \in [0, 1]\, \forall i \in N$. Once (spatial) branching is performed, the bounds on x_i may be arbitrary. Using the same proof as Theorem 1, it is simple to show that if the bounds on x are arbitrary, and we generalize the set $\mathcal{B}$ to have arbitrary bounds on the variables,

$$\mathcal{B}(\ell, u) := \left\{ (x, X) \in \mathbb{R}^n \times \mathbb{R}^{|E|} \mid X_{ij} = x_i x_j \;\; \forall \{i, j\} \in E, u_i \geq x_i \geq \ell_i \;\; \forall i \in N \right\},$$

then

$$\mathbf{conv}(\mathcal{B}(\ell, u)) = \left\{ (x, X) \in \mathbb{R}^n \times \mathbb{R}^{|E|} \mid X_{ij} = x_i x_j \;\; \forall \{i, j\} \in E, x_i \in \{\ell_i, u_i\} \;\; \forall i \in N \right\}.$$

In Appendix A.1, we discuss how to transform valid inequalities for BQP to be valid for the set $\mathbf{conv}(\mathcal{B}(\ell, u))$ so that they can be applied to box-constrained quadratic programs with arbitrary bounds. In our implementation, we keep a list of "canonical" BQP cuts found by the 0-$\frac{1}{2}$ cut generator and before calling the generator for a new point, we first check if any of the inequalities already in our list, when modified to be valid for the current bounds, is violated.

We performed an experiment to test the impact of locally strengthening cuts using updated variable bound information from branching versus an implementation where BQP cuts were added only at the root node of the search tree. Figure 3 summarizes the results of this experiment with a scatter plot of CPU times for the two methods on the instances solved by both within 1 h. The results of the experiment demonstrate that for some instances, there is a significant improvement obtained by doing local cut strengthening. For instances not solved by one of the two methods, the local cut strengthening did not seem to have a significantly positive or negative effect on the final gap obtained after one CPU hour. Local cut strengthening is implemented in the final version of our code. Full results of the experiment are given in Table 10 in the appendix.

5 Computational results

In this section, we first present a comparison of BQP-based bounds with bounds obtained by various SDP-based relaxations of BoxQP. A second experiment compares our 0-$\frac{1}{2}$-cut based solved to other state-of-the-art software for solving BoxQP. A final section demonstrates the very significant improvement that 0-$\frac{1}{2}$ cuts made to the CPLEX solver for BoxQP and for more general non-convex quadratic programs that also have linear constraints.

5.1 Comparison of BQP-based and SDP-based bounds for BoxQP

Some of the best known bounds for BoxQP in the literature are based on variants of the SDP relaxation $z_{\mathcal{S}}$ presented in Sect. 1.2. In particular, a detailed theoretical and computational study performed in [5] shows that the relaxation obtained by combining $\mathcal{S}$ with the McCormick relaxation leads to bounds that are significantly stronger than

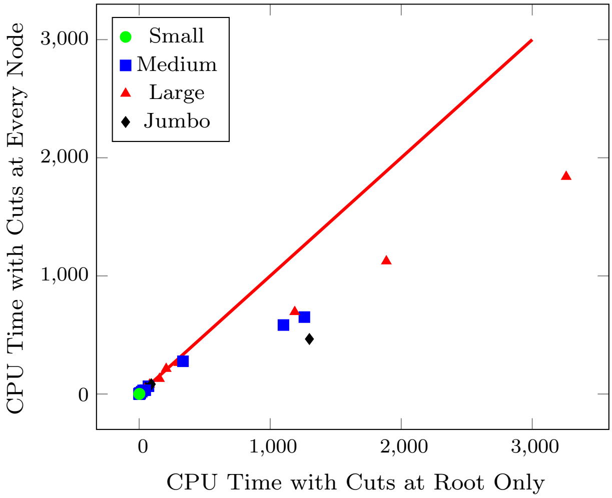

Fig. 3 Comparison of CPU time for solved instances, with and without local cut strengthening

either those obtained with $\mathcal{M}$ or $\mathcal{S}$. We denote this SDP relaxation by $\mathcal{S}+\mathcal{M}$. As stated in the introduction, this relaxation can be equivalently obtained through the doubly nonnegative relaxation of the copositive reformulation of BoxQP . We denote this alternative formulation of $\mathcal{S}+\mathcal{M}$ by $\mathcal{S}^{\geq 0}$. From a computational point of view, $\mathcal{S}^{\geq 0}$ presents the advantage that specific approximate algorithms have been developped to compute it [13].

In Table 4, we present a comparison of the three SDP-based bounds and the bound obtained with $\mathcal{M}^2 + 0\text{-}\frac{1}{2}$ (from Table 3). The two bounds for $\mathcal{S}, \mathcal{S}+\mathcal{M}$ were computed by using the solver SDPLR [12,16]. The results for $\mathcal{S}^{\geq 0}$ were obtained by taking the results at the root of the branch-and-bound of QuadProgBB [14]. Note that in spite of $\mathcal{S}+\mathcal{M}$ and $\mathcal{S}^{\geq 0}$ being equivalent, we still report both relaxations as $\mathcal{S}^{\geq 0}$ is only computed approximately in QuadProgBB

We report the geometric means of computing times for $\mathcal{S}$, $\mathcal{S}+\mathcal{M}$, $\mathcal{S}^{\geq 0}$, and $\mathcal{M}^2$ + $0\text{-}\frac{1}{2}$ in Fig. 4. We give the values of the bounds for methods $\mathcal{S}$, $\mathcal{S}+\mathcal{M}$, $\mathcal{S}^{\geq 0}$, and $\mathcal{M}^2 + 0\text{-}\frac{1}{2}$ and all instances in Table 11 of the appendix.

The results clearly confirm that $\mathcal{S}+\mathcal{M}$ is a very strong relaxation. It closes the most gap for all categories but two ($\mathcal{M}^2 + 0\text{-}\frac{1}{2}$ performs slightly better on the small and medium-size, low-density instances). The approximation of $\mathcal{S}+\mathcal{M}$ with $\mathcal{S}^{\geq 0}$ is also very good and bounds are very close for all models categories. On small instances and instances with low density, the gap closed by $\mathcal{M}^2 + 0\text{-}\frac{1}{2}$ is significantly better than the one closed by $\mathcal{S}$. However on large instances of high density and jumbo instances of medium and high density all the SDP-based bounds close more gap than $\mathcal{M}^2$ + $0\text{-}\frac{1}{2}$. Regarding computing time, the differences between $\mathcal{S}$, $\mathcal{S}^{\geq 0}$, $\mathcal{S}+\mathcal{M}$ and $\mathcal{M}^2$ + $0\text{-}\frac{1}{2}$ are quite significant. The relaxation $\mathcal{S}$ is up to 3 or 4 orders of magnitude faster than $\mathcal{S}^{\geq 0}$ and $\mathcal{M}^2 + 0\text{-}\frac{1}{2}$, and up to 5 orders of magnitude faster than $\mathcal{S}+\mathcal{M}$. The relaxation $\mathcal{S}+\mathcal{M}$ is significantly slower than all the others and, in particular, for sparse instances, it is orders of magnitude slower than $\mathcal{M}^2 + 0\text{-}\frac{1}{2}$. The remaining two

Table 4 SDP-based bounds and BQP-based bounds for BoxQP

Size	Density			% Gap closed			
		#	$\mathcal{M}^2$ Gap	$\mathcal{S}$	$\mathcal{S}^{\geq 0}$	$\mathcal{S}+\mathcal{M}$	$\mathcal{M}^2+0\text{-}\frac{1}{2}$
Small	Low	6	35.49	80.65	99.11	99.29	99.51
	Medium	9	59.93	89.79	99.4	99.46	99.29
	High	27	78.97	94.15	99.76	99.8	99.13
Medium	Low	12	47.37	85.85	99.33	99.55	99.90
	Medium	6	108.81	93.0	98.77	98.86	98.01
	High	3	163.47	95.68	99.24	99.31	93.52
Large	Low	6	68.65	88.61	98.2	98.65	98.28
	Medium	6	124.89	94.96	99.05	99.25	97.48
	High	6	180.85	96.34	99.14	99.29	90.60
Jumbo	Low	6	93.91	92.9	98.35	98.84	96.28
	Medium	6	170.78	95.25	98.6	98.82	91.42
	High	6	232.44	96.67	98.96	99.16	85.68

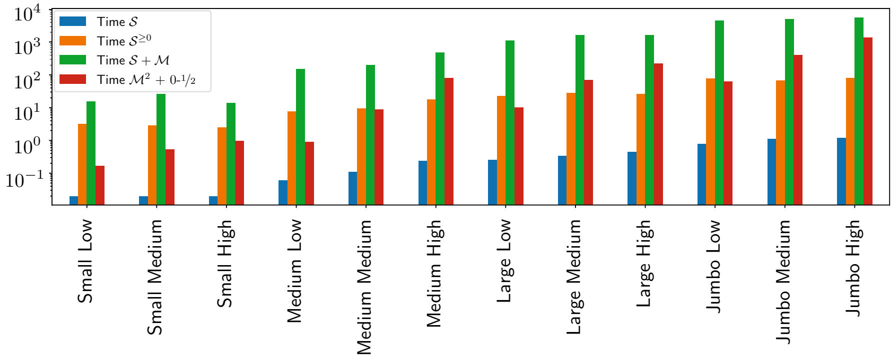

Fig. 4 Geometric means of running times for SDP based bounds and $\mathcal{M}^2+0\text{-}\frac{1}{2}$

methods $\mathcal{S}^{\geq 0}$ and $\mathcal{M}^2+0\text{-}\frac{1}{2}$ are relatively comparable, the former one being faster on large and dense instances and the latter one faster on the rest.

Finally, in Fig. 5, we report the number of cuts generated to obtain the bounds of $\mathcal{M}^2+0\text{-}\frac{1}{2}$. We report the total number of cuts generated as well as the final number of cuts that are necessary to obtain the bound (those that are active in the last relaxation). We first note that the number of cuts generated increases by orders of magnitude as the instances get larger and denser. This explains the longer computation times necessary to solve large and dense instances. Also notice that the final number of cuts in the relaxation is about 2 orders of magnitude smaller. Therefore, the resulting relaxation is suitable for a branch-and-cut procedure. We note that, we only rely on CPLEX internal methods for the management of cuts which seems to be very effective.

Overall, the results indicate that bounds obtained from the $\mathcal{M}^2+0\text{-}\frac{1}{2}$ relaxation are competitive for instances with low density both in terms of quality and time. For denser

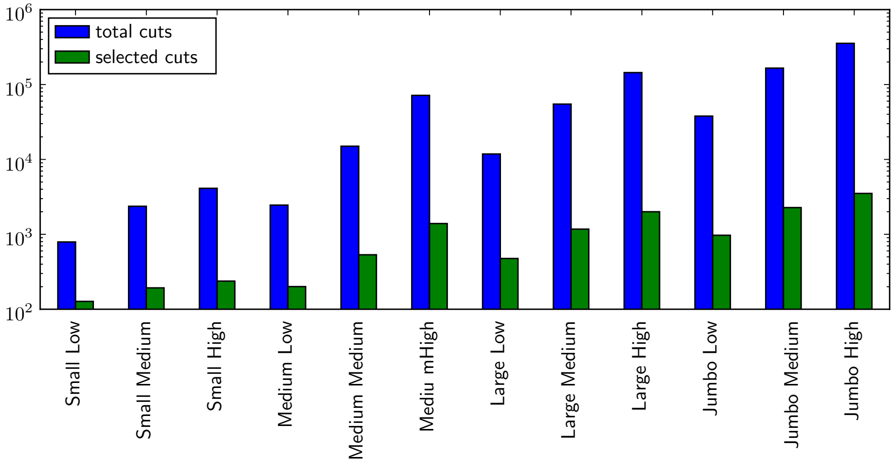

Fig. 5 Geometric means of number of cuts separated for $\mathcal{M}^2 + 0\text{-}\frac{1}{2}$

instances, SDP-based methods that are significantly cheaper to compute have better performance. We note that even though the SDP-based bounds seem to be significantly better for instances of high density, using the $\mathcal{M}^2 + 0\text{-}\frac{1}{2}$ relaxation has the advantage of being directly usable in a simplex-based branch-and-bound method.

5.2 Solver comparisons

In this section, we report results of an experiment designed to compare the performance of our solver, which we call BGL, against other state-of-the-art methods for solving non-convex quadratic programs with box constraints to global optimality. Each instance in the test suite was solved using our solver BGL, Baron [38], GloMIQO [28], and QuadProgBB [14] for a time limit of 2 CPU hours and to a final relative optimality gap tolerance of 0.01%. The computing environment for this experiment is the same one described in Sect. 4. Table 5 shows a summary of the results of this experiment aggregated by instance size and density. The exact CPU time and final optimality gap for each solver on each instance is given in Table 12 in the online supplement Sect. A.5. Figure 6 shows a performance profile [20] of the CPU time comparing the solvers on all instances, and Fig. 7 shows the same performance profile with all small and jumbo/high-density instances removed from the comparison.

The results of the experiment show that BGL, whose primary computational enhancement is to employ $0\text{-}\frac{1}{2}$ cuts for strengthening the natural relaxation is competitive with more mature state-of-the-art solvers for this problem class. It should be noted that for the large and dense problems in our test suite, the performance of the semidefinite-programming based solver QuadProgBB dominates the other solvers. For example, for jumbo/high-density instances, the average final optimality gap for QuadProgBB after two CPU hours is 0.13%, while the gap is orders of magnitude larger for the other three solvers.

Table 5 Summary comparison of solver performance on test suite

		BGL			Baron			GloMIQO			QuadProgBB		
		# Sol	Gap	Time	# Sol	Gap	Time	# Sol	Gap	Time	# Sol	Gap	Time
Small	Low	6	0.00	0.27	6	0.00	0.15	6	0.00	0.31	6	0.00	10.44
	Medium	9	0.00	0.99	9	0.00	0.35	9	0.00	2.29	9	0.00	12.84
	High	27	0.00	1.71	27	0.00	0.44	27	0.00	1.59	27	0.00	6.06
Medium	Low	12	0.00	1.48	12	0.00	0.69	12	0.00	1.45	12	0.00	33.90
	Medium	6	0.00	19.05	6	0.00	9.59	5	0.02	96.45	6	0.00	69.29
	High	3	0.00	454.16	3	0.00	2864.53	0	5.83	7200.00	3	0.00	321.05
Large	Low	6	0.00	17.76	6	0.00	19.33	6	0.00	149.69	6	0.00	273.09
	Medium	6	0.00	284.67	6	0.00	218.18	4	0.05	1057.22	6	0.00	540.35
	High	3	0.37	3351.73	3	0.38	5401.87	0	50.17	7200.00	6	0.00	1226.13
Jumbo	Low	6	0.00	318.67	5	0.02	642.90	4	0.05	2445.02	5	0.01	1391.94
	Medium	2	1.14	6086.15	1	2.68	6984.06	0	21.53	7200.00	3	0.06	4812.57
	High	0	38.65	7200.00	0	28.76	7200.00	0	151.60	7200.00	0	0.13	7200.00

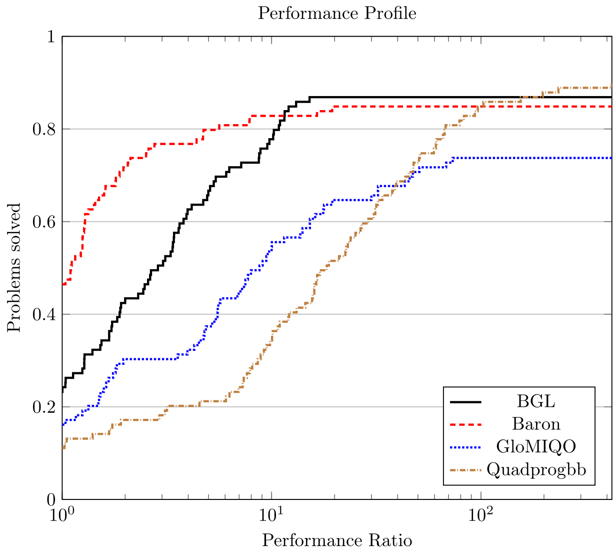

Fig. 6 CPU time performance profile comparing solvers on all instances

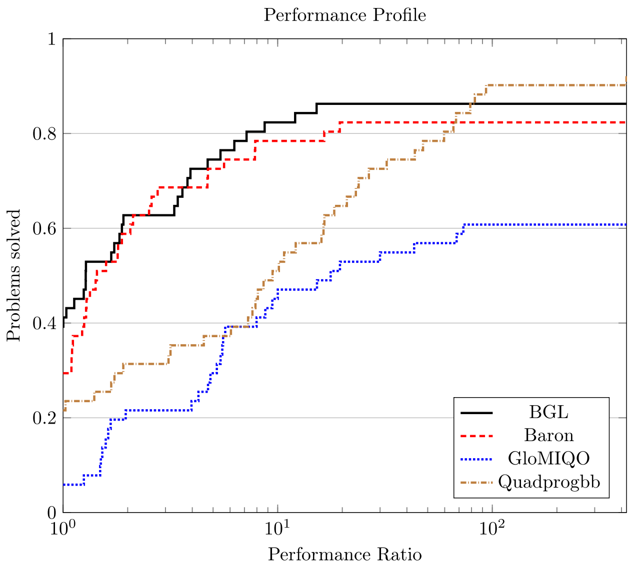

Fig. 7 CPU time performance profile comparing solvers on larger solved instances

5.3 Experiments with CPLEX implementation

The CPLEX solver implements a spatial branch-and-bound algorithm for solving nonconvex QP and MIQP problems to global optimality, see [39] for a general description of such algorithms. The $\mathcal{M}^2$ relaxation is formed and a branch-and-bound tree is developed using this relaxation by branching on variables involved in products and

Table 6 Comparison of CPLEX 12.6.3 with and without cuts on BoxQP

		Without BQP cuts			With BQP cuts		Ratios	
Category	#	# time out	Av. time	Av. nodes	Av. time	Av. nodes	Time	Nodes
All	79	35	255.77	253,301	5.38	23	40.24	7598.63
> 1 s	65	35	812.47	1,062,026	8.27	30	87.76	26,274.28
> 10 s	56	35	1847.49	2,079,462	11.45	37	148.42	43,925.40

strengthening the envelopes using the tighter bounds. For a more detailed description of the algorithm we refer the reader to [9].

Based on promising initial results from our work, recently cutting planes from the Boolean Quadric Polytope for solving non-convex QP were incorporated to CPLEX in version 12.6.2. The implementation follows closely the details described in Sect. 4 except that it does not exploit integrality as proposed in Sect. 4.2. 0-$\frac{1}{2}$ cuts are heuristically separated from the boolean quadratic polytope and added as local cuts at nodes of the branch-and-bound tree using the transformation presented in Sect. 4.3. The activation of the cut separator is controlled by the parameter `mip cuts bqp`. In the default setting cuts are separated at the root node and in the tree. An internal heuristic controls the level of effort put in the separation.

To assess the effect of the cutting planes within CPLEX, we ran the last version of CPLEX (version 12.6.3) with the cuts activated (default value of the parameter `mip cuts bqp`) and deactivated (by setting the parameter `mip cuts bqp` to the value `-1`). All experiments in this section are conducted on identical 12 core Intel Xeon CPU E5430 machines running at 2.66 GHz and equipped with 24 GB of memory. We separate the experiment in two parts. In the first one, we only consider the results of the Box-QP instances. In the second part, we consider more general QP instances with linear constraints.

The summarized results for Box-QP instances are in Table 6. All 99 Box-QP's from [13,17,40] are solved with CPLEX with and without cuts with a time limit of 10,000 s. Each row of the table shows aggregated results on a different subset of instances. The first row considers all instances that are solved by at least one of the two methods (20 instances not solved are removed, note that CPLEX with BQP cuts solves all the remaining 79 instances). The second row considers all instances solved in at least one second by the slowest of the two methods. The last row considers instances solved in at least 10 s by the slowest solver. For each row, we give the number of instances in that group, the number of time outs for CPLEX without BQP cuts, the average time (computed using the geometric mean), and the average number of nodes in the branch-and-bound tree (computed using the geometric mean). Finally we give the average ratios of time and number of nodes between the two methods.

As could be expected from the improvement of the root lower bound shown in the previous sections, the cuts produce a dramatic improvement for solving BoxQP. The reduction of time is roughly a factor of 40 on the entire test set and is a factor of about 150 for the 56 instances that took more than 10 s to solve. The node reduction is even larger, more than 3 orders of magnitude. Regarding the 20 instances that are

Table 7 Comparison of CPLEX 12.6.3 without and with cuts on instances that are not box QPs

		Without BQP cuts		With BQP cuts		Ratios	
Category	#	Av. time	Av. nodes	Av. time	Av. nodes	Time	Nodes
All	75	9.90	4008	8.84	2894	1.11	1.38
> 1 s	43	48.80	23,397	40.15	13,895	1.21	1.68
> 10 s	29	179.30	53,092	134.60	27,349	1.33	1.94

not solved by either method, CPLEX with cuts always terminates with a significantly better bound. For those 20 instances, the geometric mean of the final optimality gap is 148% for CPLEX without cuts and 11% for CPLEX with cuts.

In the second experiment, we run CPLEX on instances with a non-convex quadratic objective function and at least one linear constraint coming from the internal CPLEX development test set. We run all the instances with and without BQP cuts, and 329 instances out of the 403 were solved identically in both settings (i.e. no BQP cut was generated). We report results on the remaining 75 instances in Table 7. The grouping of instance and information reported is similar to Table 6 excepted that there were no time outs and that column is removed.

The results in Table 7 show an average improvement both in terms of time and number of nodes from the addition of BQP cuts. Although this improvement is far smaller than for BoxQP , it is still 33% in terms of time for instances that took more than 10 s and a reduction by a factor of almost 2 in terms of number of nodes.

6 Conclusions

In this paper, we demonstrate how cutting planes from the Boolean Quadric Polytope (BQP) may be effectively used to solve nonconvex quadratic programs. We show that the polytope obtained by adding all odd-cycle inequalities from BQP is equivalent to both the 0-$\frac{1}{2}$ closure and the Chvátal–Gomory closure of the McCormick relaxation. By employing CPLEX's heuristic 0-$\frac{1}{2}$ cut generator, strengthening the inequalities using local-bound information, in combination with integrality-based bound strengthening and a convex quadratic programming relaxation, we constructed a solver than can solve a well-known class of box-constrained quadratic programming instances as well as more mature software. Many of these ideas have been incoporated into recent version of the CPLEX software and have resulted in signifcant improvement on more general nonconvex quadratic instances.

A Appendix

A.1 Transforming valid inequalities

We next describe how to transform valid inequalities for BQP to be valid inequalities for the set $\mathbf{conv}(\mathcal{B}(\ell, u)) =: \text{BQP}(\ell, u)$ so that they can be applied to box-constrained quadratic programs with arbitrary bounds. We explain the transformations in the context of finding a valid inequality violated by a point $(\hat{x}, \hat{X})$ with variable bounds $\ell_i \leq x_i \leq u_i \; \forall i \in N$. The first step is to transform the variables via the equations

$$x'_i = \frac{x_i - \ell_i}{u_i - \ell_i} \qquad \forall i \in N \tag{20}$$

$$X'_{ij} = \frac{X_{ij} - \ell_i x_j - \ell_j x_i + \ell_i \ell_j}{(u_i - \ell_i)(u_j - \ell_j)} \qquad \forall \{i, j\} \in E. \tag{21}$$

The transformation (21) is obtained by scaling each variable in the product

$$x_i x_j \rightarrow \left(\frac{x_i - \ell_i}{u_i - \ell_i}\right)\left(\frac{x_j - \ell_j}{u_j - \ell_j}\right),$$

and replacing $x_i x_j$ with X_{ij}. The point $(x, X) \in \text{BQP}(\ell, u)$ if and only if the transformed point $(x', X') \in \text{BQP}$. The point (x', X') is given to CPLEX's standard 0-$\frac{1}{2}$ cut generator, which attempts to separate (x', X') for BQP. If successful, a "canonical," inequality of the form

$$\sum_{i \in N} \alpha_i x'_i + \sum_{\{i,j\} \in E} B_{ij} X'_{ij} \leq \gamma \tag{22}$$

is returned. We undo the transformation by substituting (20) and (21) into (22) to get the following equivalent inequality for $\text{BQP}(\ell, u)$:

$$\sum_{i \in N} \left(\frac{\alpha_i}{u_i - \ell_i}\right) x_i + \sum_{\{i,j\} \in E} \frac{B_{ij} X_{ij} - B_{ij} \ell_j x_i - B_{ij} \ell_i x_j}{(u_i - \ell_i)(u_j - \ell_j)} \leq \gamma + \sum_{i \in N} \frac{\alpha_i \ell_i}{(u_i - \ell_i)} - \sum_{\{i,j\} \in E} \frac{B_{ij} \ell_i \ell_j}{(u_i - \ell_i)(u_j - \ell_j)}. \tag{23}$$

To implement inequality (23), the coefficient of the x_i variable in (22) is multiplied by $(u_i - \ell_i)^{-1}$ and the quantity $\sum_{j:\{i,j\} \in E} B_{ij} \ell_j / (u_i - \ell_i)(u_j - \ell_j)$ is subtracted from the coefficient. For each X_{ij} variable, we scale the coefficient by coefficients of the X_{ij} variables by $(u_i - \ell_i)(u_j - \ell_j)$. We also add the quantity

$$\sum_{i \in N} \frac{\alpha_i \ell_i}{u_i - \ell_i} - \sum_{\{i,j\} \in E} \frac{B_{ij} \ell_i \ell_j}{(u_i - \ell_i)(u_j - \ell_j)}$$

to the right-hand-side of the returned inequality. When computing the coefficient of each variable, if the difference in bounds for the variable is smaller than a pre-specified tolerance, $(u_i - \ell_i < \epsilon)$, we take $u_i = \ell_i + 1$ when calculating the transformations (20) (21) and (23). The resulting inequality is clearly valid, since it is valid for a relaxation of the nodal subproblem being considered. Further, numerical difficulties arising from possibly large coefficient values in (23) are avoided.

When these transformations are applied to the defining inequalities of the Boolean Quadric Polytope, one obtains the well-known general version of the McCormick inequalities:

$$\begin{aligned}
X_{ij} - x_i - x_j \geq 1 \quad &\rightarrow \quad X_{ij} \geq u_j x_i + u_i x_j - u_i u_j \\
X_{ij} \geq 0 \quad &\rightarrow \quad X_{ij} \geq \ell_j x_i + \ell_i x_j - \ell_i \ell_j \\
X_{ij} \leq x_i \quad &\rightarrow \quad X_{ij} \leq \ell_j x_i + u_i x_j - u_i \ell_j \\
X_{ij} \leq x_j \quad &\rightarrow \quad X_{ij} \leq u_j x_i + \ell_i x_j - \ell_i u_j
\end{aligned}$$

A.2 BQP Bounds

Table 8 contains the raw numbers that were aggregated to make Tables 1 and 2. The value in the column z_{BoxQP} is the optimal solution value, z^U_{BQP} is the upper bound on z_{BoxQP} obtained by the BQPrestriction, and z^L_{BQP} is the lower bound on z_{BoxQP} obtained from the BQPrelaxation. The z^L_{BQP}(root) column contains the final root relaxation value from CPLEX when solving the integer program associated with z^L_{BQP}, and z_{Δ^+} is the bound at the root that could be obtained if exact separation of the odd cycle inequalities was done. The column $z_{\mathcal{M}}$ contains the lower bound obtained from optimizing over the McCormick relaxation $\mathcal{M}$.

A.3 Algorithmic improvements

Table 9 contains the raw numbers that were aggregated to make Table 3. The value in the column z_{BoxQP} is the optimal solution value, $z_{\mathcal{M}^2}$+ 0-$\frac{1}{2}$ is the bound obtained from the strengthened McCormick relaxation with CPLEX 0-$\frac{1}{2}$-cuts added, $z_{\mathcal{M}}$+0-$\frac{1}{2}$the bound from the standard McCormick relaxation with CPLEX 0-$\frac{1}{2}$-cuts added. The columns $z_{\mathcal{M}^2}$ and $z_{\mathcal{M}}$ show the lower bounds obtained from the strengthened and traditional McCormick relaxation, respectively.

Table 10 contains raw numbers on the computational performance of algorithmic techniques described in Sects. 4.2 and 4.3.

A.4 SDP relaxation bounds

This appendix contains computational time and relaxation bound information for various SDP-based approaches for solving BoxQP , as described in Sect. 5.1. For the results of the two methods SBL and Dong, the reader is invited to refer directly to the papers [21,31] (Table 11).

A.5 Solver comparison

This appendix contains the raw performance numbers comparing different solvers for BoxQP . The experiment is described in Sect. 5.2 (Table 12).

Table 8 Bounds for BQP relaxations

Name	z^U_{BQP}	z_{BoxQP}	z^L_{BQP}	$z_{\Delta+}$	z^L_{BQP}(root)	$z_{\mathcal{M}}$
spar020-100-1	− 706.50	− 706.50	− 776.00	− 776.00	− 776.00	− 1066.00
spar020-100-2	− 856.50	− 856.50	− 951.50	− 951.50	− 951.50	− 1289.00
spar020-100-3	− 772.00	− 772.00	− 877.00	− 877.00	− 877.00	− 1168.50
spar030-060-1	− 706.00	− 706.00	− 761.50	− 761.50	− 761.50	− 1454.75
spar030-060-2	− 1377.00	− 1377.17	− 1449.00	− 1449.00	− 1449.00	− 1699.50
spar030-060-3	− 1293.50	− 1293.50	− 1388.00	− 1388.00	− 1388.00	− 2047.00
spar030-070-1	− 654.00	− 654.00	− 716.00	− 716.00	− 738.24	− 1569.00
spar030-070-2	− 1313.00	− 1313.00	− 1461.00	− 1461.00	− 1461.00	− 1940.25
spar030-070-3	− 1656.50	− 1657.40	− 1784.50	− 1784.50	− 1784.50	− 2302.75
spar030-080-1	− 952.50	− 952.73	− 1017.00	− 1017.00	− 1030.85	− 2107.50
spar030-080-2	− 1597.00	− 1597.00	− 1627.50	− 1627.50	− 1627.50	− 2178.25
spar030-080-3	− 1808.50	− 1809.78	− 1870.00	− 1870.00	− 1870.00	− 2403.50
spar030-090-1	− 1296.50	− 1296.50	− 1371.00	− 1371.00	− 1371.00	− 2423.50
spar030-090-2	− 1464.00	− 1466.84	− 1607.00	− 1607.00	− 1607.00	− 2667.00
spar030-090-3	− 1494.00	− 1494.00	− 1585.00	− 1585.00	− 1585.00	− 2538.25
spar030-100-1	− 1226.00	− 1227.12	− 1391.50	− 1391.50	− 1391.50	− 2602.00
spar030-100-2	− 1260.50	− 1260.50	− 1350.00	− 1350.00	− 1350.00	− 2729.25
spar030-100-3	− 1510.00	− 1511.05	− 1640.00	− 1640.00	− 1640.00	− 2751.75
spar040-030-1	− 839.50	− 839.50	− 853.00	− 853.00	− 853.00	− 1088.00
spar040-030-2	− 1429.00	− 1429.00	− 1461.00	− 1461.00	− 1461.00	− 1635.00
spar040-030-3	− 1086.00	− 1086.00	− 1110.50	− 1110.50	− 1110.50	− 1303.25
spar040-040-1	− 837.00	− 837.00	− 879.50	− 879.50	− 883.57	− 1606.25
spar040-040-2	− 1428.00	− 1428.00	− 1500.00	− 1500.00	− 1500.00	− 1920.75
spar040-040-3	− 1173.50	− 1173.50	− 1208.50	− 1208.50	− 1211.45	− 2039.75
spar040-050-1	− 1154.50	− 1154.50	− 1209.50	− 1209.50	− 1209.50	− 2146.25
spar040-050-2	− 1430.00	− 1430.98	− 1493.50	− 1493.50	− 1493.50	− 2357.25
spar040-050-3	− 1649.50	− 1653.63	− 1672.50	− 1672.50	− 1672.50	− 2616.00
spar040-060-1	− 1321.50	− 1322.67	− 1434.00	− 1434.00	− 1451.23	− 2872.00
spar040-060-2	− 2004.00	− 2004.23	− 2106.00	− 2106.00	− 2106.00	− 2917.50
spar040-060-3	− 2454.50	− 2454.50	− 2566.50	− 2566.50	− 2566.50	− 3434.00
spar040-070-1	− 1605.00	− 1605.00	− 1757.00	− 1757.00	− 1757.00	− 3144.00
spar040-070-2	− 1867.50	− 1867.50	− 1940.50	− 1940.50	− 1940.50	− 3369.25
spar040-070-3	− 2436.50	− 2436.50	− 2527.00	− 2527.00	− 2527.00	− 3760.25
spar040-080-1	− 1838.50	− 1838.50	− 2000.00	− 2000.00	− 2000.00	− 3846.50
spar040-080-2	− 1952.50	− 1952.50	− 2078.00	− 2078.00	− 2078.00	− 3833.00
spar040-080-3	− 2545.50	− 2545.50	− 2666.50	− 2666.50	− 2666.63	− 4361.50
spar040-090-1	− 2135.50	− 2135.50	− 2253.00	− 2253.00	− 2253.00	− 4376.75
spar040-090-2	− 2113.00	− 2113.00	− 2278.50	− 2278.50	− 2278.50	− 4357.75
spar040-090-3	− 2535.00	− 2535.00	− 2664.50	− 2664.50	− 2664.50	− 4516.75

Table 8 continued

Name	z^U_{BQP}	z_{BoxQP}	z^L_{BQP}	$z_{\Delta+}$	z^L_{BQP}(root)	$z_{\mathcal{M}}$
spar040-100-1	– 2474.50	– 2476.38	– 2687.50	– 2687.50	– 2687.50	– 5009.75
spar040-100-2	– 2102.50	– 2102.50	– 2163.50	– 2170.83	– 2189.17	– 4902.75
spar040-100-3	– 1864.50	– 1866.07	– 2035.00	– 2234.50	– 2237.58	– 5075.75
spar050-030-1	– 1324.50	– 1324.50	– 1359.00	– 1359.00	– 1359.00	– 1858.25
spar050-030-2	– 1668.00	– 1668.00	– 1695.00	– 1695.00	– 1695.00	– 2334.00
spar050-030-3	– 1453.00	– 1453.61	– 1498.50	– 1498.50	– 1498.50	– 2107.25
spar050-040-1	– 1411.00	– 1411.00	– 1490.50	– 1490.50	– 1490.50	– 2632.00
spar050-040-2	– 1745.00	– 1745.76	– 1832.50	– 1832.50	– 1832.50	– 2923.25
spar050-040-3	– 2094.50	– 2094.50	– 2186.00	– 2186.00	– 2186.00	– 3273.50
spar050-050-1	– 1195.50	– 1198.41	– 1280.50	– 1418.41	– 1450.99	– 3536.00
spar050-050-2	– 1776.00	– 1776.00	– 1849.00	– 1849.00	– 1855.04	– 3500.50
spar050-050-3	– 2106.00	– 2106.10	– 2293.50	– 2293.50	– 2293.88	– 4119.75
spar060-020-1	– 1212.00	– 1212.00	– 1223.50	– 1223.50	– 1223.50	– 1757.25
spar060-020-2	– 1925.50	– 1925.50	– 1925.50	– 1925.50	– 1925.50	– 2238.25
spar060-020-3	– 1483.00	– 1483.00	– 1518.00	– 1518.00	– 1518.00	– 2098.75
spar070-025-1	– 2538.00	– 2538.91	– 2615.50	– 2615.50	– 2615.50	– 3832.75
spar070-025-2	– 1888.00	– 1888.00	– 1935.00	– 1935.00	– 1935.00	– 3248.00
spar070-025-3	– 2811.00	– 2812.28	– 2880.00	– 2880.00	– 2880.00	– 4167.25
spar070-050-1	– 3252.50	– 3252.50	– 3356.50	– 3391.61	– 3462.90	– 7210.75
spar070-050-2	– 3296.00	– 3296.00	– 3384.50	– 3384.50	– 3409.56	– 6620.00
spar070-050-3	– 4306.50	– 4306.50	– 4419.50	– 4419.50	– 4419.50	– 7522.00
spar070-075-1	– 4655.50	– 4655.50	– 4857.00	– 5055.33	– 5089.09	– 11,647.75
spar070-075-2	– 3861.00	– 3865.15	– 4106.50	– 4562.67	– 4583.49	– 10,884.75
spar070-075-3	– 4328.50	– 4329.40	– 4569.50	– 4905.17	– 4944.31	– 11,262.25
spar080-025-1	– 3157.00	– 3157.00	– 3195.00	– 3195.00	– 3195.00	– 4840.75
spar080-025-2	– 2310.00	– 2312.34	– 2397.00	– 2397.00	– 2432.78	– 4378.50
spar080-025-3	– 3090.50	– 3090.88	– 3156.00	– 3156.00	– 3156.00	– 5130.25
spar080-050-1	– 3447.50	– 3448.10	– 3669.00	– 4087.75	– 4154.85	– 9783.25
spar080-050-2	– 4449.00	– 4449.20	– 4561.50	– 4561.50	– 4595.53	– 9270.00
spar080-050-3	– 4886.00	– 4886.00	– 5079.00	– 5100.82	– 5216.82	– 10,029.75
spar080-075-1	– 5896.00	– 5896.00	– 6115.00	– 6664.33	– 6683.91	– 15,250.75
spar080-075-2	– 5341.00	– 5341.00	– 5542.00	– 5997.83	– 6058.35	– 14,246.50
spar080-075-3	– 5980.50	– 5980.50	– 6282.00	– 6673.58	– 6776.83	– 14,961.50
spar090-025-1	– 3372.50	– 3372.50	– 3428.50	– 3464.46	– 3526.12	– 6171.50
spar090-025-2	– 3495.00	– 3500.29	– 3641.50	– 3641.50	– 3668.68	– 6015.00
spar090-025-3	– 4299.00	– 4299.00	– 4329.00	– 4329.00	– 4367.98	– 6698.25
spar090-050-1	– 5152.00	– 5152.00	– 5287.50	– 5499.30	– 5611.51	– 12,584.00
spar090-050-2	– 5386.50	– 5386.50	– 5512.50	– 5512.50	– 5573.10	– 11,920.50
spar090-050-3	– 6151.00	– 6151.00	– 6324.50	– 6324.50	– 6439.11	– 12,514.00

Table 8 continued

Name	z_{BQP}^{U}	z_{BoxQP}	z_{BQP}^{L}	z_{Δ^+}	z_{BQP}^{L}(root)	$z_{\mathcal{M}}$
spar090-075-1	– 6267.00	– 6267.45	– 7044.08	– 7994.83	– 8016.93	– 19,054.25
spar090-075-2	– 5647.50	– 5647.50	– 6333.52	– 7402.50	– 7406.32	– 18,245.50
spar090-075-3	– 6450.00	– 6450.00	– 6841.88	– 7972.33	– 7974.10	– 18,929.50
spar100-025-1	– 4027.50	– 4027.50	– 4135.00	– 4180.88	– 4255.79	– 7660.75
spar100-025-2	– 3892.50	– 3892.56	– 3948.50	– 3948.50	– 4002.76	– 7338.50
spar100-025-3	– 4453.50	– 4453.50	– 4559.50	– 4559.50	– 4581.40	– 7942.25
spar100-050-1	– 5490.00	– 5490.00	– 6170.36	– 6383.92	– 6474.28	– 15,415.75
spar100-050-2	– 5866.00	– 5866.00	– 6108.00	– 6559.28	– 6665.38	– 14,920.50
spar100-050-3	– 6485.00	– 6485.00	– 6655.50	– 7084.22	– 7179.24	– 15,564.25
spar100-075-1	– 7383.50	– 7384.20	– 8634.77	– 9636.67	– 9645.89	– 23,387.50
spar100-050-2	– 6755.50	– 6755.50	– 7959.95	– 8913.33	– 8927.68	– 22,440.00
spar100-050-3	– 7513.50	– 7554.00	– 9095.90	– 9683.33	– 9692.79	– 23,243.50
spar125-025-1	– 5567.50	– 5572.00	– 5749.00	– 6154.57	– 6244.26	– 12,251.00
spar125-025-2	– 6156.00	– 6156.06	– 6288.00	– 6481.02	– 6596.36	– 12,732.00
spar125-025-3	– 6815.50	– 6815.50	– 6845.00	– 6946.94	– 7055.47	– 12,650.75
spar125-050-1	– 9261.00	– 9308.38	– 10,502.38	– 10,956.33	– 11,030.48	– 24,993.00
spar125-050-2	– 8391.00	– 8395.00	– 9138.73	– 10,380.00	– 10,388.46	– 24,810.50
spar125-050-3	– 8329.00	– 8343.91	– 9953.62	– 10,089.67	– 10,126.01	– 24,424.50
spar125-050-1	– 12,222.50	– 12,330.00	– 16,121.85	– 16,152.00	– 16,181.85	– 38,202.00
spar125-050-2	– 10,137.00	– 10,382.47	– 13,972.83	– 15,156.67	– 15,158.92	– 37,466.75
spar125-050-3	– 9435.50	– 9635.50	– 13,406.12	– 14,001.67	– 14,002.17	– 36,202.25

Table 9 Comparing bounds for $\mathcal{M}$ and $\mathcal{M}^2$ relaxations

Name	z_{BoxQP}	$z_{\mathcal{M}^2+0\text{-}\frac{1}{2}}$	$z_{\mathcal{M}+0\text{-}\frac{1}{2}}$	$z_{\mathcal{M}^2}$	$z_{\mathcal{M}}$
spar020-100-1	– 706.50	– 706.89	– 776.00	– 1038.38	– 1066.00
spar020-100-2	– 856.50	– 868.32	– 951.50	– 1258.38	– 1289.00
spar020-100-3	– 772.00	– 772.13	– 877.00	– 1142.00	– 1168.50
spar030-060-1	– 706.00	– 725.11	– 761.50	– 1430.00	– 1454.75
spar030-060-2	– 1377.17	– 1379.18	– 1449.00	– 1668.25	– 1699.50
spar030-060-3	– 1293.50	– 1315.19	– 1388.00	– 2006.50	– 2047.00
spar030-070-1	– 654.00	– 704.17	– 738.24	– 1547.25	– 1569.00
spar030-070-2	– 1313.00	– 1318.22	– 1461.00	– 1888.25	– 1940.25
spar030-070-3	– 1657.40	– 1677.21	– 1784.50	– 2251.12	– 2302.75
spar030-080-1	– 952.73	– 987.81	– 1030.85	– 2072.00	– 2107.50
spar030-080-2	– 1597.00	– 1597.00	– 1627.50	– 2158.12	– 2178.25
spar030-080-3	– 1809.78	– 1809.78	– 1870.00	– 2376.25	– 2403.50
spar030-090-1	– 1296.50	– 1298.70	– 1371.00	– 2385.12	– 2423.50
spar030-090-2	– 1466.84	– 1474.93	– 1607.00	– 2622.75	– 2667.00

Table 9 continued

Name	z_{BoxQP}	$z_{\mathcal{M}^2+0\text{-}\frac{1}{2}}$	$z_{\mathcal{M}+0\text{-}\frac{1}{2}}$	$z_{\mathcal{M}^2}$	$z_{\mathcal{M}}$
spar030-090-3	– 1494.00	– 1494.88	– 1585.00	– 2499.38	– 2538.25
spar030-100-1	– 1227.12	– 1242.51	– 1391.50	– 2541.50	– 2602.00
spar030-100-2	– 1260.50	– 1270.51	– 1350.00	– 2698.88	– 2729.25
spar030-100-3	– 1511.05	– 1526.60	– 1640.00	– 2703.75	– 2751.75
spar040-030-1	– 839.50	– 839.50	– 853.00	– 1067.00	– 1088.00
spar040-030-2	– 1429.00	– 1429.36	– 1461.00	– 1617.75	– 1635.00
spar040-030-3	– 1086.00	– 1086.00	– 1110.50	– 1297.12	– 1303.25
spar040-040-1	– 837.00	– 857.50	– 883.57	– 1575.50	– 1606.25
spar040-040-2	– 1428.00	– 1428.79	– 1500.00	– 1895.75	– 1920.75
spar040-040-3	– 1173.50	– 1184.06	– 1211.45	– 2017.25	– 2039.75
spar040-050-1	– 1154.50	– 1159.72	– 1209.50	– 2120.88	– 2146.25
spar040-050-2	– 1430.98	– 1439.17	– 1493.50	– 2334.88	– 2357.25
spar040-050-3	– 1653.63	– 1653.63	– 1672.50	– 2603.00	– 2616.00
spar040-060-1	– 1322.67	– 1392.62	– 1451.23	– 2817.88	– 2872.00
spar040-060-2	– 2004.23	– 2010.40	– 2106.00	– 2872.62	– 2917.50
spar040-060-3	– 2454.50	– 2454.65	– 2566.50	– 3386.12	– 3434.00
spar040-070-1	– 1605.00	– 1611.33	– 1757.00	– 3070.12	– 3144.00
spar040-070-2	– 1867.50	– 1871.20	– 1940.50	– 3323.00	– 3369.25
spar040-070-3	– 2436.50	– 2441.81	– 2527.00	– 3724.50	– 3760.25
spar040-080-1	– 1838.50	– 1844.24	– 2000.00	– 3788.62	– 3846.50
spar040-080-2	– 1952.50	– 1964.38	– 2078.00	– 3775.38	– 3833.00
spar040-080-3	– 2545.50	– 2556.93	– 2666.63	– 4311.12	– 4361.50
spar040-090-1	– 2135.50	– 2145.50	– 2253.00	– 4325.50	– 4376.75
spar040-090-2	– 2113.00	– 2148.73	– 2278.50	– 4304.38	– 4357.75
spar040-090-3	– 2535.00	– 2550.19	– 2664.50	– 4453.38	– 4516.75
spar040-100-1	– 2476.38	– 2489.87	– 2687.50	– 4932.12	– 5009.75
spar040-100-2	– 2102.50	– 2145.04	– 2189.17	– 4855.25	– 4902.75
spar040-100-3	– 1866.07	– 2166.61	– 2237.58	– 5017.25	– 5075.75
spar050-030-1	– 1324.50	– 1324.82	– 1359.00	– 1837.75	– 1858.25
spar050-030-2	– 1668.00	– 1669.28	– 1695.00	– 2324.62	– 2334.00
spar050-030-3	– 1453.61	– 1456.21	– 1498.50	– 2093.75	– 2107.25
spar050-040-1	– 1411.00	– 1415.59	– 1490.50	– 2580.62	– 2632.00
spar050-040-2	– 1745.76	– 1749.01	– 1832.50	– 2891.88	– 2923.25
spar050-040-3	– 2094.50	– 2096.04	– 2186.00	– 3236.00	– 3273.50
spar050-050-1	– 1198.41	– 1415.71	– 1450.99	– 3506.25	– 3536.00
spar050-050-2	– 1776.00	– 1806.44	– 1855.04	– 3467.12	– 3500.50
spar050-050-3	– 2106.10	– 2151.14	– 2293.88	– 4052.12	– 4119.75
spar060-020-1	– 1212.00	– 1212.00	– 1223.50	– 1745.50	– 1757.25
spar060-020-2	– 1925.50	– 1925.50	– 1925.50	– 2230.00	– 2238.25
spar060-020-3	– 1483.00	– 1483.42	– 1518.00	– 2081.00	– 2098.75

Table 9 continued

Name	z_{BoxQP}	$z_{\mathcal{M}^2+0\text{-}\frac{1}{2}}$	$z_{\mathcal{M}+0\text{-}\frac{1}{2}}$	$z_{\mathcal{M}^2}$	$z_{\mathcal{M}}$
spar070-025-1	− 2538.91	− 2553.18	− 2615.50	− 3788.88	− 3832.75
spar070-025-2	− 1888.00	− 1895.01	− 1935.00	− 3232.88	− 3248.00
spar070-025-3	− 2812.28	− 2819.92	− 2880.00	− 4148.38	− 4167.25
spar070-050-1	− 3252.50	− 3411.63	− 3462.90	− 7151.12	− 7210.75
spar070-050-2	− 3296.00	− 3341.97	− 3409.56	− 6573.88	− 6620.00
spar070-050-3	− 4306.50	− 4308.40	− 4419.50	− 7473.88	− 7522.00
spar070-075-1	− 4655.50	− 5010.60	− 5089.09	− 11,578.12	− 11,647.75
spar070-075-2	− 3865.15	− 4469.11	− 4583.49	− 10,793.38	− 10,884.75
spar070-075-3	− 4329.40	− 4826.89	− 4944.31	− 11,162.38	− 11,262.25
spar080-025-1	− 3157.00	− 3157.23	− 3195.00	− 4829.12	− 4840.75
spar080-025-2	− 2312.34	− 2383.57	− 2432.78	− 4351.00	− 4378.50
spar080-025-3	− 3090.88	− 3108.04	− 3156.00	− 5102.88	− 5130.25
spar080-050-1	− 3448.10	− 4042.38	− 4154.85	− 9696.62	− 9783.25
spar080-050-2	− 4449.20	− 4498.70	− 4595.53	− 9205.50	− 9270.00
spar080-050-3	− 4886.00	− 5044.25	− 5216.82	− 9967.25	− 10,029.75
spar080-075-1	− 5896.00	− 6579.91	− 6683.91	− 15,154.75	− 15,250.75
spar080-075-2	− 5341.00	− 5945.51	− 6058.35	− 14,146.62	− 14,246.50
spar080-075-3	− 5980.50	− 6652.30	− 6776.83	− 14,860.88	− 14,961.50
spar090-025-1	− 3372.50	− 3469.80	− 3526.12	− 6135.25	− 6171.50
spar090-025-2	− 3500.29	− 3588.62	− 3668.68	− 5978.38	− 6015.00
spar090-025-3	− 4299.00	− 4341.99	− 4367.98	− 6681.88	− 6698.25
spar090-050-1	− 5152.00	− 5547.69	− 5611.51	− 12,522.38	− 12,584.00
spar090-050-2	− 5386.50	− 5514.60	− 5573.10	− 11,851.38	− 11,920.50
spar090-050-3	− 6151.00	− 6345.88	− 6439.11	− 12,452.50	− 12,514.00
spar090-075-1	− 6267.45	− 7899.63	− 8016.93	− 18,944.50	− 19,054.25
spar090-075-2	− 5647.50	− 7290.15	− 7406.32	− 18,132.50	− 18,245.50
spar090-075-3	− 6450.00	− 7866.58	− 7974.10	− 18,823.50	− 18,929.50
spar100-025-1	− 4027.50	− 4177.38	− 4255.79	− 7611.38	− 7660.75
spar100-025-2	− 3892.56	− 3975.38	− 4002.76	− 7303.12	− 7338.50
spar100-025-3	− 4453.50	− 4504.17	− 4581.40	− 7894.75	− 7942.25
spar100-050-1	− 5490.00	− 6407.46	− 6474.28	− 15,341.75	− 15,415.75
spar100-050-2	− 5866.00	− 6573.51	− 6665.38	− 14,814.62	− 14,920.50
spar100-050-3	− 6485.00	− 7080.22	− 7179.24	− 15,480.12	− 15,564.25
spar100-075-1	− 7384.20	− 9504.28	− 9645.89	− 23,277.12	− 23,387.50
spar100-075-2	− 6755.50	− 8768.39	− 8927.68	− 22,307.00	− 22,440.00
spar100-075-3	− 7554.00	− 9553.26	− 9692.79	− 23,109.62	− 23,243.50
spar125-025-1	− 5572.00	− 6164.92	− 6244.26	− 12,184.75	− 12,251.00
spar125-025-2	− 6156.06	− 6462.89	− 6596.36	− 12,662.62	− 12,732.00
spar125-025-3	− 6815.50	− 7019.67	− 7055.47	− 12,627.50	− 12,650.75
spar125-050-1	− 9308.38	− 10,879.26	− 11,030.48	− 24,880.25	− 24,993.00

Table 9 continued

Name	z_{BoxQP}	$z_{\mathcal{M}^2+0\text{-}\frac{1}{2}}$	$z_{\mathcal{M}+0\text{-}\frac{1}{2}}$	$z_{\mathcal{M}^2}$	$z_{\mathcal{M}}$
spar125-050-2	– 8395.00	– 10,215.57	– 10,388.46	– 24,669.38	– 24,810.50
spar125-050-3	– 8343.91	– 9999.21	– 10,126.01	– 24,308.00	– 24,424.50
spar125-075-1	– 12,330.00	– 16,009.38	– 16,181.85	– 38,058.12	– 38,202.00
spar125-075-2	– 10,382.47	– 15,025.72	– 15,158.92	– 37,341.38	– 37,466.75
spar125-075-3	– 9635.50	– 13,857.12	– 14,002.17	– 36,033.00	– 36,202.25

Table 10 Computational performance of integer strengthening and local cut improvement

Name	With int. vars			Without int. vars			Without local cuts		
	# Nodes	CPU (s)	Gap (%)	Nodes	CPU (s)	Gap (%)	Nodes	CPU (s)	Gap (%)
spar020-100-1	2	0.10	0.00	1	0.09	0.01	2	0.10	0.00
spar020-100-2	7	0.16	0.00	14	0.17	0.00	7	0.15	0.00
spar020-100-3	1	0.13	0.00	1	0.10	0.00	1	0.13	0.00
spar030-060-1	10	0.82	0.00	8	0.60	0.00	63	1.04	0.00
spar030-060-2	2	0.10	0.00	4	0.09	0.00	2	0.10	0.00
spar030-060-3	16	0.74	0.00	16	0.55	0.00	64	0.76	0.00
spar030-070-1	23	2.34	0.00	22	1.87	0.00	34	2.06	0.00
spar030-070-2	4	0.39	0.00	6	0.33	0.00	4	0.39	0.00
spar030-070-3	47	0.53	0.01	55	0.48	0.00	81	0.51	0.00
spar030-080-1	6	1.91	0.00	14	1.77	0.00	15	1.89	0.00
spar030-080-2	1	0.26	0.00	1	0.22	0.00	1	0.25	0.00
spar030-080-3	1	0.23	0.00	1	0.21	0.00	1	0.22	0.00
spar030-090-1	1	1.04	0.00	1	0.73	0.01	1	1.02	0.00
spar030-090-2	18	1.60	0.00	8	1.01	0.00	10	1.53	0.00
spar030-090-3	1	0.65	0.00	4	0.62	0.00	1	0.64	0.00
spar030-100-1	8	2.74	0.00	6	1.65	0.00	10	2.67	0.00
spar030-100-2	4	2.87	0.00	4	1.98	0.00	4	2.85	0.00
spar030-100-3	14	1.79	0.01	15	1.55	0.01	19	1.77	0.01
spar040-030-1	1	0.08	0.00	1	0.07	0.00	1	0.07	0.00
spar040-030-2	4	0.09	0.00	4	0.10	0.00	4	0.09	0.00
spar040-030-3	1	0.09	0.00	1	0.08	0.00	1	0.09	0.00
spar040-040-1	11	1.64	0.00	10	1.08	0.00	18	1.61	0.00
spar040-040-2	1	0.32	0.00	4	0.23	0.00	1	0.31	0.00
spar040-040-3	4	1.67	0.00	4	1.24	0.00	4	1.65	0.00
spar040-050-1	10	2.63	0.00	8	1.62	0.00	14	2.55	0.00
spar040-050-2	8	1.40	0.00	6	0.77	0.00	8	1.35	0.00
spar040-050-3	1	1.52	0.00	1	0.98	0.00	1	1.50	0.00

Table 10 continued

Name	With int. vars			Without int. vars			Without local cuts		
	# Nodes	CPU (s)	Gap (%)	Nodes	CPU (s)	Gap (%)	Nodes	CPU (s)	Gap (%)
spar040-060-1	26	5.43	0.00	30	4.63	0.00	115	5.40	0.00
spar040-060-2	4	0.87	0.00	6	0.69	0.00	4	0.86	0.00
spar040-060-3	1	0.98	0.00	1	0.72	0.01	1	0.97	0.00
spar040-070-1	4	3.41	0.00	4	2.08	0.00	4	3.38	0.00
spar040-070-2	4	2.92	0.00	6	2.10	0.00	4	2.90	0.00
spar040-070-3	6	2.59	0.00	4	1.77	0.00	6	2.58	0.00
spar040-080-1	4	5.85	0.00	4	3.63	0.00	4	5.66	0.00
spar040-080-2	4	5.38	0.00	6	3.36	0.00	6	5.12	0.00
spar040-080-3	25	4.64	0.01	18	3.08	0.00	28	4.21	0.00
spar040-090-1	4	7.08	0.00	6	5.54	0.00	4	6.92	0.00
spar040-090-2	8	7.94	0.00	12	7.26	0.00	8	7.71	0.00
spar040-090-3	12	6.05	0.00	6	4.05	0.00	11	5.62	0.00
spar040-100-1	10	8.91	0.00	8	6.33	0.00	12	8.63	0.00
spar040-100-2	8	11.06	0.00	10	9.87	0.00	12	10.85	0.00
spar040-100-3	78	35.98	0.00	713	396.50	0.00	440	45.83	0.00
spar050-030-1	1	0.41	0.00	1	0.27	0.00	1	0.39	0.00
spar050-030-2	2	1.12	0.01	6	0.94	0.00	2	1.01	0.01
spar050-030-3	8	1.10	0.00	8	0.70	0.00	10	0.99	0.00
spar050-040-1	4	4.04	0.00	4	2.38	0.00	4	3.69	0.00
spar050-040-2	6	3.47	0.00	6	2.25	0.00	6	3.23	0.00
spar050-040-3	1	2.46	0.00	4	1.64	0.00	1	2.28	0.00
spar050-050-1	82	32.36	0.00	372	150.22	0.00	681	46.20	0.00
spar050-050-2	6	7.23	0.00	9	6.32	0.00	10	6.80	0.00
spar050-050-3	14	9.13	0.00	16	7.50	0.00	23	8.54	0.00
spar060-020-1	1	0.65	0.00	1	0.44	0.00	1	0.63	0.00
spar060-020-2	1	0.23	0.00	1	0.20	0.00	1	0.22	0.00
spar060-020-3	3	1.22	0.00	4	0.81	0.00	3	1.08	0.00
spar070-025-1	12	5.18	0.00	8	3.37	0.00	12	4.82	0.00
spar070-025-2	4	6.07	0.00	4	4.43	0.00	4	5.76	0.00
spar070-025-3	6	5.89	0.00	6	4.56	0.00	6	5.51	0.00
spar070-050-1	51	65.80	0.00	60	102.67	0.00	206	70.75	0.00
spar070-050-2	8	30.81	0.00	6	32.35	0.00	16	29.91	0.00
spar070-050-3	4	19.82	0.00	4	17.14	0.00	4	19.29	0.00
spar070-075-1	48	277.19	0.00	252	1122.20	0.00	198	334.16	0.00
spar070-075-2	267	651.12	0.00	770	3600.01	7.73	1854	1260.79	0.00
spar070-075-3	260	583.86	0.01	847	3600.00	4.39	1465	1101.09	0.00
spar080-025-1	1	8.11	0.00	1	5.97	0.00	1	7.83	0.00

Table 10 continued

Name	With int. vars			Without int. vars			Without local cuts		
	# Nodes	CPU (s)	Gap (%)	Nodes	CPU (s)	Gap (%)	Nodes	CPU (s)	Gap (%)
spar080-025-2	16	18.00	0.00	20	17.93	0.00	115	21.87	0.00
spar080-025-3	10	12.17	0.00	8	9.98	0.00	10	11.78	0.00
spar080-050-1	1289	1727.20	0.00	1036	3600.00	9.70	5670	3600.00	3.46
spar080-050-2	10	81.16	0.00	12	79.02	0.00	18	79.85	0.00
spar080-050-3	46	129.29	0.00	84	215.34	0.00	259	154.98	0.00
spar080-075-1	529	2043.05	0.00	355	3600.00	6.59	2739	3600.00	1.58
spar080-075-2	234	1124.30	0.00	430	3600.00	5.93	1257	1886.49	0.00
spar080-075-3	575	1839.54	0.01	442	3600.00	7.00	2607	3259.04	0.01
spar090-025-1	42	39.60	0.00	69	66.82	0.00	135	40.87	0.00
spar090-025-2	41	33.23	0.00	42	45.38	0.00	174	37.34	0.01
spar090-025-3	11	23.05	0.00	10	21.22	0.00	14	23.05	0.00
spar090-050-1	203	695.32	0.00	826	3600.00	1.49	1138	1187.22	0.00
spar090-050-2	22	214.09	0.00	32	232.73	0.00	30	206.16	0.00
spar090-050-3	69	264.47	0.00	131	494.24	0.00	259	301.71	0.01
spar090-075-1	375	3600.00	25.07	149	3600.00	26.48	1466	3600.00	27.24
spar090-075-2	380	3600.00	23.93	164	3600.00	26.72	1471	3600.02	24.81
spar090-075-3	258	3600.01	17.03	126	3600.00	22.06	867	3600.00	17.12
spar100-025-1	34	83.08	0.00	78	135.09	0.00	136	92.41	0.00
spar100-025-2	18	63.57	0.00	16	64.41	0.00	28	62.97	0.00
spar100-025-3	20	52.88	0.00	12	50.62	0.00	37	50.71	0.00
spar100-050-1	730	3600.00	11.85	283	3600.00	15.99	2000	3600.00	15.74
spar100-050-2	909	3600.01	3.69	436	3600.00	8.69	2557	3600.00	7.94
spar100-050-3	887	3147.78	0.01	451	3600.00	5.68	2469	3600.00	4.92
spar100-075-1	111	3600.00	27.20	53	3600.00	30.24	549	3600.00	31.23
spar100-075-2	130	3600.04	25.41	67	3600.00	28.74	638	3600.00	26.58
spar100-075-3	171	3600.00	28.17	79	3600.00	26.70	853	3600.00	24.30
spar125-025-1	1197	3600.00	3.77	651	3600.00	6.77	3145	3600.00	5.15
spar125-025-2	651	1545.90	0.00	912	3600.00	0.56	3882	3600.00	1.22
spar125-025-3	159	465.63	0.00	222	905.61	0.00	1750	1299.17	0.01
spar125-050-1	109	3600.00	22.09	55	3600.00	19.36	667	3600.00	19.36
spar125-050-2	145	3600.00	25.69	70	3600.06	21.13	764	3600.00	21.96
spar125-050-3	77	3600.00	18.87	49	3600.00	19.69	528	3600.01	18.58
spar125-075-1	1	3600.02	126.37	1	3600.00	85.72	1	3600.06	126.37
spar125-075-2	1	3600.01	195.12	1	3600.00	109.14	1	3600.00	197.69
spar125-075-3	1	3600.00	270.10	1	3600.01	109.43	1	3600.32	272.93

Table 11 Bound and solution time for the various SDP based relaxations

Name	Bound			Time		
	$\mathcal{S}$	$\mathcal{S}^{\geq 0}$	$\mathcal{S}+\mathcal{M}$	$\mathcal{S}$	$\mathcal{S}^{\geq 0}$	$\mathcal{S}+\mathcal{M}$
spar020-100-1	– 739.39	– 706.69	– 706.51	0.00	3.38	3.61
spar020-100-2	– 900.20	– 858.99	– 857.91	0.00	1.44	29.36
spar020-100-3	– 785.51	– 771.91	– 772.00	0.01	0.98	0.56
spar030-060-1	– 768.12	– 714.74	– 714.67	0.01	2.14	23.11
spar030-060-2	– 1426.94	– 1377.03	– 1377.17	0.01	2.01	2.79
spar030-060-3	– 1370.11	– 1298.54	– 1298.21	0.02	2.26	76.29
spar030-070-1	– 746.43	– 675.01	– 674.00	0.01	2.11	13.08
spar030-070-2	– 1375.06	– 1313.34	– 1313.00	0.01	2.02	3.74
spar030-070-3	– 1719.77	– 1658.78	– 1657.55	0.02	2.17	199.38
spar030-080-1	– 1050.76	– 965.30	– 965.24	0.02	2.09	19.27
spar030-080-2	– 1622.77	– 1596.80	– 1597.00	0.01	1.67	1.97
spar030-080-3	– 1836.79	– 1809.60	– 1809.78	0.02	2.00	1.60
spar030-090-1	– 1348.47	– 1296.32	– 1296.50	0.01	1.58	4.71
spar030-090-2	– 1527.86	– 1467.21	– 1466.84	0.02	1.99	3.62
spar030-090-3	– 1516.80	– 1494.00	– 1494.00	0.02	1.61	2.11
spar030-100-1	– 1285.74	– 1226.97	– 1227.12	0.02	2.00	4.87
spar030-100-2	– 1365.32	– 1262.43	– 1261.08	0.01	2.18	64.32
spar030-100-3	– 1611.11	– 1514.29	– 1513.08	0.02	2.18	185.92
spar040-030-1	– 876.60	– 839.67	– 839.50	0.01	2.99	7.18
spar040-030-2	– 1496.83	– 1429.76	– 1429.00	0.02	3.43	13.56
spar040-030-3	– 1156.52	– 1086.47	– 1086.00	0.01	3.25	11.00
spar040-040-1	– 956.08	– 863.44	– 863.09	0.02	3.28	39.73
spar040-040-2	– 1452.51	– 1427.83	– 1428.00	0.03	2.85	5.39
spar040-040-3	– 1269.83	– 1184.03	– 1180.84	0.02	3.41	66.52
spar040-050-1	– 1276.78	– 1161.64	– 1160.43	0.03	3.38	33.93
spar040-050-2	– 1517.50	– 1437.98	– 1436.00	0.03	3.31	217.53
spar040-050-3	– 1747.29	– 1654.24	– 1653.63	0.04	3.54	23.86
spar040-060-1	– 1481.95	– 1353.15	– 1352.92	0.03	3.45	132.09
spar040-060-2	– 2099.57	– 2004.98	– 2004.23	0.03	3.67	12.08
spar040-060-3	– 2508.65	– 2454.23	– 2454.50	0.04	2.84	4.96
spar040-070-1	– 1663.97	– 1605.19	– 1605.00	0.07	3.41	11.68
spar040-070-2	– 1931.33	– 1867.26	– 1867.50	0.05	3.13	8.67
spar040-070-3	– 2522.69	– 2437.87	– 2436.50	0.04	3.72	34.57
spar040-080-1	– 1936.15	– 1838.28	– 1838.50	0.05	2.90	12.70
spar040-080-2	– 2012.90	– 1952.25	– 1952.50	0.07	3.50	8.54
spar040-080-3	– 2638.33	– 2548.93	– 2545.73	0.03	3.82	217.32
spar040-090-1	– 2262.50	– 2135.87	– 2135.50	0.04	3.35	24.76
spar040-090-2	– 2268.85	– 2115.35	– 2113.65	0.03	3.80	277.56
spar040-090-3	– 2594.22	– 2535.00	– 2535.00	0.08	3.06	6.80

Table 11 continued

Name	Bound			Time		
	$\mathcal{S}$	$\mathcal{S}^{\geq 0}$	$\mathcal{S}+\mathcal{M}$	$\mathcal{S}$	$\mathcal{S}^{\geq 0}$	$\mathcal{S}+\mathcal{M}$
spar040-100-1	– 2557.23	– 2476.96	– 2476.38	0.04	3.68	13.59
spar040-100-2	– 2216.62	– 2107.72	– 2106.34	0.03	3.73	64.89
spar040-100-3	– 2037.17	– 1908.33	– 1908.17	0.06	3.60	48.90
spar050-030-1	– 1389.03	– 1324.30	– 1324.50	0.06	5.04	14.35
spar050-030-2	– 1755.67	– 1674.60	– 1671.31	0.05	5.41	378.34
spar050-030-3	– 1565.75	– 1455.79	– 1454.84	0.04	5.48	167.29
spar050-040-1	– 1483.01	– 1411.51	– 1411.00	0.06	5.73	31.75
spar050-040-2	– 1881.33	– 1751.24	– 1749.43	0.03	5.42	283.27
spar050-040-3	– 2176.90	– 2094.70	– 2094.50	0.05	5.39	25.39
spar050-050-1	– 1417.76	– 1305.81	– 1302.23	0.06	5.25	73.74
spar050-050-2	– 1942.51	– 1790.56	– 1789.56	0.06	5.84	360.85
spar050-050-3	– 2268.02	– 2122.48	– 2121.92	0.05	5.70	164.95
spar060-020-1	– 1297.41	– 1213.32	– 1212.00	0.05	8.36	55.44
spar060-020-2	– 2010.47	– 1926.53	– 1925.50	0.07	8.93	31.79
spar060-020-3	– 1604.59	– 1492.83	– 1491.05	0.06	8.95	241.24
spar070-025-1	– 2692.99	– 2549.85	– 2544.85	0.10	15.13	1920.24
spar070-025-2	– 2060.77	– 1911.10	– 1908.87	0.10	13.89	878.59
spar070-025-3	– 2996.88	– 2832.96	– 2826.29	0.09	13.56	821.30
spar070-050-1	– 3533.87	– 3282.56	– 3278.25	0.35	14.88	504.93
spar070-050-2	– 3478.22	– 3305.90	– 3300.30	0.19	15.68	603.37
spar070-050-3	– 4375.15	– 4306.00	– 4306.50	0.17	16.31	45.78
spar070-075-1	– 4892.26	– 4675.08	– 4670.21	0.14	17.63	757.74
spar070-075-2	– 4218.84	– 3951.39	– 3946.05	0.23	17.94	460.14
spar070-075-3	– 4644.28	– 4382.76	– 4378.32	0.44	17.37	339.10
spar080-025-1	– 3332.73	– 3165.29	– 3157.00	0.21	17.93	266.81
spar080-025-2	– 2557.24	– 2357.42	– 2344.45	0.17	16.33	470.57
spar080-025-3	– 3291.44	– 3111.87	– 3103.89	0.22	16.43	1013.67
spar080-050-1	– 3890.89	– 3596.05	– 3584.24	0.28	22.31	430.58
spar080-050-2	– 4636.46	– 4456.36	– 4451.08	0.30	21.60	2149.75
spar080-050-3	– 5166.54	– 4927.89	– 4920.67	0.26	22.31	1045.51
spar080-075-1	– 6201.88	– 5945.90	– 5923.92	0.27	22.46	694.05
spar080-075-2	– 5655.92	– 5408.53	– 5388.87	0.38	21.23	1364.52
spar080-075-3	– 6337.73	– 6054.59	– 6038.69	0.29	21.60	2648.52
spar090-025-1	– 3656.10	– 3442.26	– 3433.13	0.31	29.72	2590.54
spar090-025-2	– 3828.14	– 3559.72	– 3549.29	0.42	31.18	1903.97
spar090-025-3	– 4610.85	– 4353.00	– 4341.46	0.30	30.03	3024.68
spar090-050-1	– 5548.68	– 5242.96	– 5223.85	0.30	33.09	2287.87
spar090-050-2	– 5590.35	– 5397.40	– 5386.69	0.48	33.27	3345.94
spar090-050-3	– 6494.22	– 6212.89	– 6192.08	0.52	41.11	2842.32

Table 11 continued

Name	Bound			Time		
	$\mathcal{S}$	$\mathcal{S}^{\geq 0}$	$\mathcal{S}+\mathcal{M}$	$\mathcal{S}$	$\mathcal{S}^{\geq 0}$	$\mathcal{S}+\mathcal{M}$
spar090-075-1	– 6730.48	– 6419.10	– 6403.55	0.48	31.53	2553.88
spar090-075-2	– 6157.54	– 5791.01	– 5778.30	0.81	32.70	1595.51
spar090-075-3	– 6889.42	– 6542.45	– 6531.28	0.70	32.99	2100.68
spar100-025-1	– 4290.51	– 4084.61	– 4066.38	0.65	53.08	2091.85
spar100-025-2	– 4136.46	– 3938.88	– 3923.82	0.55	57.13	4366.51
spar100-025-3	– 4714.92	– 4495.41	– 4476.86	0.48	58.45	2631.61
spar100-050-1	– 6026.30	– 5688.73	– 5671.45	1.34	45.30	4800.97
spar100-050-2	– 6390.11	– 6010.12	– 5995.12	0.68	46.10	1671.70
spar100-050-3	– 6919.95	– 6562.03	– 6540.72	0.73	45.63	3478.39
spar100-075-1	– 7940.76	– 7537.74	– 7514.48	1.16	41.82	5578.85
spar100-075-2	– 7291.11	– 6916.78	– 6883.89	0.59	47.47	3942.80
spar100-075-3	– 8182.46	– 7700.01	– 7681.68	0.85	45.02	2504.42
spar125-025-1	– 6104.64	– 5753.43	– 5733.72	1.41	106.08	4709.49
spar125-025-2	– 6545.53	– 6262.82	– 6222.69	1.28	108.66	8499.27
spar125-025-3	– 7216.31	– 6898.89	– 6867.50	0.75	110.02	8801.00
spar125-050-1	– 10,044.93	– 9569.28	– 9517.79	1.32	111.10	8518.94
spar125-050-2	– 9018.75	– 8580.10	– 8541.97	0.87	95.41	8712.94
spar125-050-3	– 8987.96	– 8535.57	– 8498.15	2.34	95.40	8633.25
spar125-075-1	– 13,005.48	– 12,535.36	– 12,464.79	0.97	203.64	8706.82
spar125-075-2	– 11,285.96	– 10,771.68	– 10,714.10	1.59	96.21	8535.99
spar125-075-3	– 10,464.94	– 9927.91	– 9867.81	3.03	145.47	8538.35

Table 12 CPU time and final optimality gap of each solver on all instances

Name	BGL		Baron		GloMIQO		Quadprogbb	
	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)
spar020-100-1	0.10	0.00	0.05	0.00	0.04	0.00	6.20	0.00
spar020-100-2	0.15	0.00	0.06	0.00	0.97	0.00	2.86	0.00
spar020-100-3	0.12	0.00	0.05	0.00	0.04	0.00	1.56	0.00
spar030-060-1	0.79	0.00	0.23	0.00	10.50	0.00	10.40	0.00
spar030-060-2	0.09	0.00	0.05	0.00	0.09	0.00	3.05	0.00
spar030-060-3	0.67	0.00	0.22	0.00	7.09	0.00	11.25	0.00
spar030-070-1	2.27	0.00	1.32	0.00	23.50	0.00	9.72	0.00
spar030-070-2	0.37	0.00	0.07	0.00	0.54	0.00	4.29	0.00
spar030-070-3	0.49	0.00	0.96	0.00	4.44	0.00	13.18	0.00
spar030-080-1	1.82	0.00	1.18	0.00	8.38	0.00	10.84	0.00
spar030-080-2	0.24	0.00	0.07	0.00	0.07	0.00	2.75	0.00
spar030-080-3	0.20	0.00	0.06	0.00	0.08	0.00	3.04	0.00

Table 12 continued

Name	BGL		Baron		GloMIQO		Quadprogbb	
	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)
spar030-090-1	0.97	0.00	0.14	0.00	0.11	0.00	2.50	0.00
spar030-090-2	1.52	0.00	0.66	0.00	2.97	0.00	6.49	0.00
spar030-090-3	0.61	0.00	0.12	0.00	0.12	0.00	2.70	0.00
spar030-100-1	2.62	0.00	0.30	0.00	2.06	0.00	3.02	0.00
spar030-100-2	2.73	0.00	0.28	0.00	0.28	0.00	4.41	0.00
spar030-100-3	1.70	0.00	0.64	0.00	3.54	0.00	9.27	0.00
spar040-030-1	0.07	0.00	0.05	0.00	0.04	0.00	7.89	0.00
spar040-030-2	0.08	0.00	0.08	0.00	0.14	0.00	8.22	0.00
spar040-030-3	0.08	0.00	0.06	0.00	0.04	0.00	9.38	0.00
spar040-040-1	1.56	0.00	1.12	0.00	10.73	0.00	35.43	0.00
spar040-040-2	0.29	0.00	0.09	0.00	0.07	0.00	4.70	0.00
spar040-040-3	1.60	0.00	0.44	0.00	3.25	0.00	12.76	0.00
spar040-050-1	2.49	0.00	0.50	0.00	3.75	0.00	18.84	0.00
spar040-050-2	1.29	0.00	0.85	0.00	3.03	0.00	26.17	0.00
spar040-050-3	1.43	0.00	0.15	0.00	0.13	0.00	12.56	0.00
spar040-060-1	5.13	0.00	8.26	0.00	260.78	0.00	87.27	0.00
spar040-060-2	0.82	0.00	0.31	0.00	3.56	0.00	10.48	0.00
spar040-060-3	0.91	0.00	0.15	0.00	0.15	0.00	4.70	0.00
spar040-070-1	3.25	0.00	0.30	0.00	0.30	0.00	7.55	0.00
spar040-070-2	2.77	0.00	0.28	0.00	0.27	0.00	5.07	0.00
spar040-070-3	2.46	0.00	0.62	0.00	9.45	0.00	9.68	0.00
spar040-080-1	5.50	0.00	0.54	0.00	0.56	0.00	4.79	0.00
spar040-080-2	4.95	0.00	0.43	0.00	0.50	0.00	5.66	0.00
spar040-080-3	4.06	0.00	1.64	0.00	23.04	0.00	10.33	0.00
spar040-090-1	6.58	0.00	0.57	0.00	1.04	0.00	9.12	0.00
spar040-090-2	7.44	0.00	0.78	0.00	25.14	0.00	8.53	0.00
spar040-090-3	5.63	0.00	0.43	0.00	20.34	0.00	5.17	0.00
spar040-100-1	8.40	0.00	2.93	0.00	29.40	0.00	8.53	0.00
spar040-100-2	10.24	0.00	2.08	0.00	9.95	0.00	14.46	0.00
spar040-100-3	32.73	0.00	143.45	0.00	448.75	0.00	34.38	0.00
spar050-030-1	0.36	0.00	0.10	0.00	0.10	0.00	7.89	0.00
spar050-030-2	0.95	0.00	0.32	0.00	0.25	0.00	23.38	0.00
spar050-030-3	0.94	0.00	0.74	0.00	4.01	0.00	32.23	0.00
spar050-040-1	3.56	0.00	0.41	0.00	3.27	0.00	13.20	0.00
spar050-040-2	3.08	0.00	0.43	0.00	4.06	0.00	35.66	0.00
spar050-040-3	2.17	0.00	0.20	0.00	0.18	0.00	12.20	0.00
spar050-050-1	29.48	0.00	42.40	0.00	7200.00	4.41	279.03	0.00
spar050-050-2	6.47	0.00	3.53	0.00	5.73	0.00	30.34	0.00

Table 12 continued

Name	BGL		Baron		GloMIQO		Quadprogbb	
	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)
spar050-050-3	8.24	0.00	6.60	0.00	116.51	0.00	66.84	0.00
spar060-020-1	0.57	0.00	0.45	0.00	0.67	0.00	26.82	0.00
spar060-020-2	0.19	0.00	0.10	0.00	0.15	0.00	42.21	0.00
spar060-020-3	1.01	0.00	1.10	0.00	1.59	0.00	47.84	0.00
spar070-025-1	4.56	0.00	4.55	0.00	22.11	0.00	105.26	0.00
spar070-025-2	5.37	0.00	8.52	0.00	10.52	0.00	85.89	0.00
spar070-025-3	5.17	0.00	6.34	0.00	7.87	0.00	137.45	0.00
spar070-050-1	59.12	0.00	78.88	0.00	1772.29	0.00	102.61	0.00
spar070-050-2	28.20	0.00	8.24	0.00	46.94	0.00	66.63	0.00
spar070-050-3	18.22	0.00	1.20	0.00	2.00	0.00	28.61	0.00
spar070-075-1	268.17	0.00	899.72	0.00	7200.00	3.24	159.93	0.00
spar070-075-2	627.61	0.00	4931.82	0.00	7200.00	7.87	643.82	0.00
spar070-075-3	556.58	0.00	5297.14	0.00	7200.00	7.78	321.39	0.00
spar080-025-1	7.29	0.00	1.16	0.00	1.45	0.00	76.42	0.00
spar080-025-2	16.33	0.00	45.04	0.00	1197.34	0.00	198.13	0.00
spar080-025-3	11.12	0.00	12.25	0.00	57.89	0.00	233.66	0.00
spar080-050-1	1640.67	0.00	2947.25	0.00	7200.00	6.07	2293.21	0.00
spar080-050-2	76.32	0.00	16.18	0.00	76.63	0.00	127.48	0.00
spar080-050-3	122.16	0.00	108.38	0.00	949.01	0.00	335.93	0.00
spar080-075-1	1983.27	0.00	7134.76	0.00	7200.00	45.60	366.40	0.00
spar080-075-2	1073.00	0.00	2655.99	0.00	7200.00	14.97	561.45	0.00
spar080-075-3	1784.98	0.00	3512.88	0.00	7200.00	26.70	1396.49	0.00
spar090-025-1	36.49	0.00	77.31	0.00	1578.06	0.00	596.72	0.00
spar090-025-2	30.49	0.00	39.22	0.00	594.79	0.00	503.23	0.00
spar090-025-3	21.28	0.00	26.68	0.00	118.46	0.00	390.38	0.00
spar090-050-1	671.74	0.00	1212.47	0.00	7200.00	0.51	1125.99	0.00
spar090-050-2	204.36	0.00	61.98	0.00	342.07	0.00	280.56	0.00
spar090-050-3	253.37	0.00	277.53	0.00	1082.63	0.00	802.29	0.00
spar090-075-1	7200.00	15.82	7200.00	17.84	7200.00	101.07	3058.95	0.00
spar090-075-2	7200.00	16.56	7200.00	17.78	7200.00	96.33	3027.09	0.00
spar090-075-3	7200.00	11.09	7200.00	11.07	7200.00	89.81	1277.38	0.00
spar100-025-1	77.20	0.00	363.28	0.00	5266.25	0.00	588.78	0.00
spar100-025-2	59.30	0.00	121.99	0.00	904.11	0.00	431.56	0.00
spar100-025-3	48.81	0.00	126.16	0.00	193.59	0.00	522.82	0.00
spar100-050-1	7200.00	7.61	7200.00	5.43	7200.00	12.89	7149.61	0.00
spar100-050-2	6299.12	0.00	7200.00	3.11	7200.00	6.00	6082.07	0.00
spar100-050-3	3002.26	0.00	5997.69	0.00	7200.00	3.88	765.47	0.00
spar100-075-1	7200.00	20.69	7200.00	21.05	7200.00	121.92	7200.00	0.00

Table 12 continued

Name	BGL		Baron		GloMIQO		Quadprogbb	
	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)
spar100-075-2	7200.00	21.06	7200.00	23.52	7200.00	118.29	7200.00	0.18
spar100-075-3	7200.00	18.32	7200.00	20.78	7200.00	116.90	7200.00	0.03
spar125-025-1	7121.34	0.00	7200.00	5.19	7200.00	6.10	7200.00	0.55
spar125-025-2	1472.88	0.00	2768.70	0.00	7200.00	0.94	2810.68	0.00
spar125-025-3	446.70	0.00	633.46	0.00	4471.04	0.00	2705.34	0.00
spar125-050-1	7200.00	13.27	7200.00	11.03	7200.00	63.87	7200.00	0.82
spar125-050-2	7200.00	16.39	7200.00	15.83	7200.00	80.17	7200.00	0.48
spar125-050-3	7200.00	13.78	7200.00	12.72	7200.00	64.64	7200.00	0.35
spar125-075-1	7200.00	64.51	7200.00	29.04	7200.00	169.18	7200.00	0.06
spar125-075-2	7200.00	79.95	7200.00	44.16	7200.00	201.10	7200.00	1.72
spar125-075-3	7200.00	80.95	7200.00	42.84	7200.00	211.60	7200.00	0.88

A.6 CPLEX performance

This appendix contains the raw performance numbers for the computational experiment comparing the performance of CPLEX v12.6.3 with and without the BQP-based cutting planes described in Sect. 5.3 (Table 13).

Table 13 Results of complete optimization without and with cuts

Name	Without cuts		With cuts	
	Time	Nodes	Time	Nodes
spar020-100-1	0.1	133	0.1	1
spar020-100-2	0.2	513	0.2	9
spar020-100-3	0.1	317	0.1	3
spar030-060-1	9.9	51,890	0.6	9
spar030-060-2	0.1	61	0.1	3
spar030-060-3	1.9	9771	0.5	19
spar030-070-1	72.6	314,490	1.5	29
spar030-070-2	0.4	1575	0.2	7
spar030-070-3	0.4	1618	0.4	40
spar030-080-1	134.6	447,951	1.8	17
spar030-080-2	0.2	455	0.1	1
spar030-080-3	0.2	463	0.1	1
spar030-090-1	17.4	61,009	0.5	3
spar030-090-2	20.8	70,980	0.6	7
spar030-090-3	7.5	28,198	0.3	1
spar030-100-1	136.5	391,167	1.0	7
spar030-100-2	249.8	652,357	1.5	3
spar030-100-3	22.9	70,689	0.8	11

Table 13 continued

Name	Without cuts		With cuts	
	Time	Nodes	Time	Nodes
spar040-030-1	0.1	85	0.1	1
spar040-030-2	0.1	81	0.1	3
spar040-030-3	0.1	81	0.1	1
spar040-040-1	19.5	91,337	1.0	18
spar040-040-2	0.3	673	0.1	1
spar040-040-3	27.4	133,282	1.2	15
spar040-050-1	43.6	149,827	0.8	9
spar040-050-2	13.5	50,493	0.5	5
spar040-050-3	19.0	63,619	0.6	1
spar040-060-1	$>$ 3 h	12,918,559	3.6	55
spar040-060-2	3.8	13,431	0.4	5
spar040-060-3	2.6	7783	0.3	1
spar040-070-1	451.3	1,116,397	1.3	3
spar040-070-2	211.9	498,200	0.8	3
spar040-070-3	73.0	176,689	0.7	3
spar040-080-1	$>$ 3 h	10,676,606	2.1	3
spar040-080-2	1396.2	2,589,489	1.6	5
spar040-080-3	751.2	1,35,3795	2.2	17
spar040-090-1	$>$ 3 h	8,800,277	2.5	3
spar040-090-2	$>$ 3 h	9,385,959	4.5	11
spar040-090-3	1413.7	2,281,783	1.9	5
spar040-100-1	$>$ 3 h	8,803,200	3.2	7
spar040-100-2	$>$ 3 h	7,165,207	6.5	11
spar040-100-3	$>$ 3 h	6,428,449	104.8	1221
spar050-030-1	0.5	1605	0.1	1
spar050-030-2	4.3	18,851	1.0	7
spar050-030-3	3.2	13,809	0.6	7
spar050-040-1	729.4	2,156,993	1.7	5
spar050-040-2	334.7	909,959	1.4	5
spar050-040-3	129.5	403,241	0.8	3
spar050-050-2	$>$ 3 h	10,158,947	7.0	31
spar050-050-3	$>$ 3 h	9,057,565	4.9	23
spar060-020-1	2.5	12,085	0.2	1
spar060-020-2	0.2	171	0.1	1
spar060-020-3	3.7	16,547	0.6	3
spar070-025-1	$>$ 3 h	18,261,950	1.5	9
spar070-025-2	$>$ 3 h	11,162,373	3.0	3
spar070-025-3	$>$ 3 h	17,117,643	3.7	7
spar070-050-1	$>$ 3 h	5,313,653	70.8	111
spar070-050-2	$>$ 3 h	5,891,966	19.8	11
spar070-050-3	$>$ 3 h	5,991,645	5.5	3

Table 13 continued

Name	Without cuts		With cuts	
	Time	Nodes	Time	Nodes
spar070-075-1	> 3 h	3,443,957	466.2	397
spar070-075-2	> 3 h	3,569,921	> 3 h	20,215
spar070-075-3	> 3 h	3,422,028	5147.9	11,942
spar080-025-1	> 3 h	8,787,217	2.6	1
spar080-025-2	> 3 h	7,564,065	11.6	17
spar080-025-3	> 3 h	8,22,8971	6.9	9
spar080-050-1	> 3 h	3,947,164	> 3 h	25,322
spar080-050-2	> 3 h	4,460,132	44.4	9
spar080-050-3	> 3 h	4,283,999	112.3	209
spar080-075-1	> 3 h	2,615,954	> 3 h	7593
spar080-075-2	> 3 h	2,760,183	8774.1	9018
spar080-075-3	> 3 h	2,741,087	> 3 h	11,796
spar090-025-1	> 3 h	6,454,171	35.5	57
spar090-025-2	> 3 h	6,712,858	28.9	61
spar090-025-3	> 3 h	6,435,041	21.6	33
spar090-050-1	> 3 h	3,079,272	2019.0	5259
spar090-050-2	> 3 h	3,639,335	118.3	31
spar090-050-3	> 3 h	3,499,548	215.4	353
spar090-075-1	> 3 h	1,946,980	> 3 h	1612
spar090-075-2	> 3 h	2,022,430	> 3 h	1664
spar090-075-3	> 3 h	1,994,188	> 3 h	1389
spar100-025-1	> 3 h	5,344,623	98.4	183
spar100-025-2	> 3 h	5,567,581	46.5	21
spar100-025-3	> 3 h	5,172,256	31.8	25
spar100-050-1	> 3 h	2,386,711	> 3 h	4845
spar100-050-2	> 3 h	2,818,271	> 3 h	9537
spar100-050-3	> 3 h	2,677,484	> 3 h	9728
spar100-075-1	> 3 h	1,510,814	> 3 h	605
spar100-075-2	> 3 h	1,576,131	> 3 h	747
spar100-075-3	> 3 h	1,551,344	> 3 h	949
spar125-025-1	> 3 h	3,404,271	> 3 h	14,914
spar125-025-2	> 3 h	3,542,776	1584.6	3818
spar125-025-3	> 3 h	3,428,487	275.9	291
spar125-050-1	> 3 h	1,542,944	> 3 h	995
spar125-050-2	> 3 h	1,551,429	> 3 h	1040
spar125-050-3	> 3 h	1,552,933	> 3 h	813
spar125-075-1	> 3 h	763,199	> 3 h	78
spar125-075-2	> 3 h	716,066	> 3 h	140
spar125-075-3	> 3 h	810,317	> 3 h	109

References

1. An, L.T.H., Tao, P.D.: A branch and bound method via d.c. optimization algorithms and ellipsoidal technique for box constrained nonconvex quadratic problems. J. Glob. Optim. **13**, 171–206 (1998)
2. Andersen, M., Dahl, J., Vandenberghe, L.: CVXOPT user's guide, release 1.1.8 (2015)
3. Anstreicher, K.: On convex relaxations for quadratically constrained quadratic programming. Math. Program. **136**, 233–251 (2012)
4. Anstreicher, K., Burer, S.: Computable representations for convex hulls of low-dimensional quadratic forms. Math. Program. **124**, 33–43 (2010)
5. Anstreicher, K.M.: Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. J. Glob. Optim. **43**(2), 471–484 (2008)
6. Barahona, F.: On cuts and matchings in planar graphs. Math. Program. **60**, 53,58 (1993)
7. Barahona, F., Grötschel, M., Jünger, M., Reinelt, G.: Experiments in quadratic 01 programming. Math. Program. **44**, 127–137 (1989)
8. Barahona, F., Mahjoub, A.: On the cut polytope. Math. Program. **36**, 157–173 (1986)
9. Bliek, C., Bonami, P., Lodi, A.: Solving mixed-integer quadratic programming problems with IBM-CPLEX: a progress report. In: Proceedings of the Twenty-Sixth RAMP Symposium, pp. 171–180 (2014)
10. Boros, E., Crama, Y., Hammer, P.L.: Chvátal cuts and odd cycle inequalities in quadratic 0–1 optimization. SIAM J. Discrete Math. **5**(2), 163–177 (1992)
11. Boros, E., Hammer, P.L.: Cut-polytopes, Boolean quadric polytopes and nonnegative quadratic pseudo-Boolean functions. Math. Oper. Res. **18**(1), 245–253 (1993)
12. Burer, S., Monteiro, D.R.: A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. Math. Program. **95**(2), 329–357 (2003). https://doi.org/10.1007/s10107-002-0352-8
13. Burer, S.: Optimizing a polyhedral-semidefinite relaxation of completely positive programs. Math. Program. Comput. **2**(1), 119 (2010)
14. Burer, S., Chen, J.: Globally solving nonconvex quadratic programming problems via completely positive programming. Math. Program. Comput. **4**(1), 33–52 (2012)
15. Burer, S., Letchford, A.: On nonconvex quadratic programming with box constriants. SIAM J. Optim. **20**(2), 1073–1089 (2009)
16. Burer, S., Monteiro, R., Choi, C.: SDPLR 1.03-beta user's guide (short version) (2009). http://sburer.github.io/files/SDPLR-1.03-beta-usrguide.pdf
17. Burer, S., Vandenbussche, D.: Globally solving box-constrained nonconvex quadratic programs with semdefinite-based finite branch-and-bound. Comput. Optim. Appl. **43**, 181–195 (2009)
18. Caprara, A., Fischetti, M.: $\{0, \frac{1}{2}\}$ chvátal-gomory cuts. Math. Program. **74**, 221–235 (1996)
19. Chvátal, V.: Edmonds polytopes and weakly Hamiltonian graphs. Math. Program. **5**, 29–40 (1973)
20. Dolan, E., Moré, J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)
21. Dong, H.: Relaxing nonconvex quadratic functions by multiple adaptive diagonal perturbations. SIAM J Optim **26**(3), 1962–1985 (2014). https://doi.org/10.1137/140960657
22. Dong, H., Linderoth, J.: On valid inequalities for quadratic programming with continuous variables and binary indicators. In: IPCO 2013: The Sixteenth Conference on Integer Programming and Combinatorial Optimization, vol. 7801, pp. 169–180. Springer (2013)
23. Gomory, R.E.: Outline of an algorithm for integer solutions to linear programs. Bull. Am. Math. Mon. **64**, 275–278 (1958)
24. Hansen, P., Jaumard, B., Ruiz, M., Xiong, J.: Global minimization of indefinite quadratic functions subject to box constraints. Naval Res. Logist. **40**(3), 373–392 (1993)
25. Horst, H., Pardalos, P.M., Thoai, V.: Introduction to Global Optimization, 2nd edn. Kluwer, Dordrecht (2000)
26. Koster, A., Zymolka, A., Kutschka, M.: Algorithms to separate 0,1/2-Chvátal–Gomory cuts. Algorithmica **55**(2), 375–391 (2009)
27. McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: part I—convex underestimating problems. Math. Program. **10**, 147–175 (1976)
28. Misener, R., Smadbeck, J.B., Floudas, C.A.: Dynamically-generated cutting planes for mixed-integer quadratically-constrained quadratic programs and their incorporation into GloMIQO 2.0. Optim. Methods Softw. **30**, 215–249 (2015)

29. Padberg, M.: The boolean quadric polytope: some characteristics, facets, and relatives. Math. Program. **45**, 139–172 (1989)
30. Padberg, M.W.: Total unimodularity and the Euler-subgraph problem. Oper. Res. Lett. **7**(4), 173–179 (1988)
31. Saxena, A., Bonami, P., Lee, J.: Convex relaxations of non-convex mixed integer quadratically constrained programs: projected formulations. Math. Program. **130**, 359–413 (2011). Version with appendix available at http://www.optimization-online.org/DB_FILE/2008/11/2145.pdf
32. Sherali, H., Tuncbilek, C.: A new reformulation-convexification approach for solving nonconvex quadratic programming problems. J. Glob. Optim. **7**, 1–31 (1995)
33. Sherali HD, Alameddine AR (1990) An explicit characterization of the convex envelope of a bivariate function over special polytopes. Ann. Oper. Res. Comput. Methods Glob. Optim. **25**(1): 197–210
34. Shor, N.Z.: Quadratic optimization problems. Sov. J. Circuits Syst. Sci. **25**(6), 1–11 (1987)
35. Simone, C.D.: The cut polytope and the boolean quadric polytope. Discrete Math. **79**, 71–75 (1989)
36. Sturm, J.F.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optim. Methods Softw. **11–12**, 625–653 (1999)
37. The MOSEK command line tool. Version 7.1 (revision 51) (2016). http://docs.mosek.com/7.1/tools/index.html
38. Tawarmalani, M., Sahinidis, N.: A polyhedral branch-and-cut approach to global optimization. Math. Program. **103**, 225–249 (2005)
39. Tawarmalani, M., Sahinidis, N.V.: Global optimization of mixed integer nonlinear programs: a theoretical and computational study. Math. Program. **99**, 563–591 (2004)
40. Vandenbussche, D., Nemhauser, G.L.: A branch-and-cut algorithm for nonconvex quadratic programs with box constraints. Math. Program. **102**, 559–575 (2005)
41. Yajima, Y., Fujie, T.: A polyhedral approach for nonconvex quadratic programming problems with box constraints. J. Glob. Optim. **13**, 151–170 (1998)