Hybrid LP/SDP Bounding Procedure

Fabio Furini[1] and Emiliano Traversi[2]

[1] LIPN, Université Paris 13, 93430 Villetaneuse, France
fabio.furini@lipn.univ-paris13.fr,
[2] Fakultät für Mathematik, TU Dortmund, 44227 Dortmund, Germany
emiliano.traversi@math.tu-dortmund.de,

Abstract. The principal idea of this paper is to exploit Semidefinite Programming (SDP) relaxation within the framework provided by Mixed Integer Nonlinear Programming (MINLP) solvers when tackling Binary Quadratic Problems (BQP). SDP relaxation is well-known to provide strong bounds for BQP in practice. However, the method is not typically implemented in many state-of-the-art MINLP solvers based on Linear Programming (LP) relaxation. This paper demonstrates that this idea could be computationally useful. The Quadratic Stable Set Problem (QSSP) is adopted as the case study. The tests indicate that the Hybrid LP/SDP Bounding Procedure allows a noticeable reduction of computing time and a cut of almost one order of magnitude for the branching nodes. Furthermore, we propose and test a new linearization technique which outperforms the standard one for many classes of QSSP instances.

Keywords: Binary Quadratic Problem, Semidefinite Relaxation, Continuous Relaxation, Branch and Cut, Quadratic Stable Set Problem.

1 Introduction

There are two main classical approaches present in the literature for solving Binary Quadratic Problems (BQP). The first one is directly using a Mixed Integer Nonlinear Programming (MINLP) solver to tackle the mathematical formulation (possibly linearizing the quadratic terms). The second approach uses Branch and Bound techniques which rely on the Semidefinite Programming (SDP) relaxation. The advantage of using MINLP solvers is that they have been strongly developed for decades. To mention just a few examples, we can cite some commercial software like CPLEX [9], Gurobi [12], BARON [3] and GloMIQO [11]; as well as non commercial, for instance SCIP [17] and Bonmin [6]. They rely on sophisticated Branch and Cut (BC) algorithms based on a smart implicit enumeration of the branching tree. The bounding procedure typically makes use of Linear Programming (LP) Relaxation in order to prune the branching nodes. This relaxation can be efficiently computed but often, to be effective, it must be strengthened by adding families of valid inequalities. The second approach relies instead on the SDP relaxation. This relaxation is typically stronger than LP relaxation but generally it is computationally heavy. To the best of our knowledge, there are not many generic SDP-based solvers, BiqCrunch [4] to name one. On

the other hand, there are many problem-oriented SDP algorithms, and we refer the interested reader for instance to [5], [8] or [15]. Always in this context, we mention the work in [2], where the author mixes SDP and Linearization techniques proving that stronger bounds can be achieved.
The principal contribution of this paper is to combine the strengths of both approaches, i.e. to exploit the SDP relaxation when enhancing the pruning strategies of the MINLP solvers. Another important contribution of this paper is the proposal and the study of a new linearization technique. This new linearization provides the same LP relaxation as the standard one, however in some specific cases it better preserves the combinatorial structure of the original problem. Furthermore, as our results will underline, it can be computationally more effective. The presentation of the theoretical part of the paper is done for the generic case of the BQP. After each technique is presented, we apply it to the specific case of the Quadratic Stable Set Problem (QSSP). The QSSP will thus be adopted as the case study of this paper. This quadratic counterpart of the Linear Stable Set Problem has not received much attention in the literature (we refer the interested reader for the linear case to [1], [10] and [16]). Furthermore, to the best of our knowledge, the only papers that address the QSSP are [13] and [14]. In these works it appears as the sub-problem of a Column Generation algorithm and little computational analysis is presented. For this reason and because the QSSP is a good compromise between simplicity and generality, we have used it for our computational experiments. To summarize, the goal of this paper boils down to addressing the following research questions:

- Is it worth exploiting the strength of bounds provided by SDP relaxation within a well tuned LP-based BC framework?
- Which is the optimal trade-off between the time saved by pruning nodes and the time needed for computing the SDP relaxation?
- With respect to the QSSP, does the new linearization technique outperform the standard one?

2 Quadratic Formulation

The Quadratic Formulation (QF) of the BQP, with n variables and p constraints, is defined as follows:

$$\text{(QF)} \quad \min \sum_{i=1}^{n}\sum_{j=1}^{n} Q_{ij}x_ix_j + \sum_{i=1}^{n} L_ix_i$$
$$x \in K$$
$$x \in \{0,1\}^n,$$

with $Q \in \mathbb{R}^{n\times n}$, $L \in \mathbb{R}^n$, $K = \{x \in \mathbb{R}^n : Ax \geq b\}$, $A \in \mathbb{R}^{p\times n}$ and $b \in \mathbb{R}^p$. Q is a generic symmetric matrix, not restricted to being convex.

In the case of QSSP on a given undirected graph $G = (V, E)$, with V the set of vertices and E the set of edges, we have: $K = \{x \in \mathbb{R}^n : x_i + x_j \leq 1, \forall \{i, j\} \in E\}$. Due to these edge constraints, the only relevant quadratic costs are the ones relative to the couple of nodes that are not connected by an edge.

In order to solve the QF, many generic MINLP solvers are available. In addition, other solvers, explicitly defined for BQP can be used, such as BiqCrunch and GloMIQO. BiqCrunch is a semidefinite-based solver for binary quadratic problems. GloMIQO instead is a solver for mixed-integer quadratically constrained quadratic programs.

3 Linear Reformulations

Another option for modeling BQP is to linearize the quadratic terms using the RLT inequalities [19] and obtain the following Linear Reformulation (LF):

$$
\begin{aligned}
\text{(LF)} \quad \min \ & \sum_{i=1}^{n} \sum_{j=i}^{n} Q_{ij} y_{ij} + \sum_{i=1}^{n} L_i x_i \\
& y_{ij} \leq x_i && i, j = 1, \ldots, n \\
& y_{ij} \leq x_j && i, j = 1, \ldots, n \\
& y_{ij} \geq x_i + x_j - 1 && i, j = 1, \ldots, n \\
& y_{ij} \geq 0 && i, j = 1, \ldots, n \\
& x \in K \\
& x \in \{0, 1\}^n \ .
\end{aligned}
$$

This linearization increases the size of the problem adding at most n^2 non-negative variables and $3n^2$ constraints.
We now introduce the following linearization, based on the new *extended*-RLT inequalities and called Extended Linear Reformulation (ELF) in the following:

$$
\begin{aligned}
\text{(ELF)} \quad \min \ & \sum_{i=1}^{n} \sum_{j=1}^{n} Q_{ij} + \sum_{i=1}^{n} L_i x_i - \sum_{i=1}^{n} \sum_{j=1}^{n} Q_{ij} (z_{ij}^i + z_{ij}^j) \\
& z_{ij}^i + z_{ij}^j \leq 1 && i, j = 1, \ldots, n \\
& x_i + z_{ij}^i \leq 1 && i, j = 1, \ldots, n \\
& x_j + z_{ij}^j \leq 1 && i, j = 1, \ldots, n \\
& x_i + z_{ij}^i + z_{ij}^j \geq 1 && i, j = 1, \ldots, n && (1) \\
& x_j + z_{ij}^i + z_{ij}^j \geq 1 && i, j = 1, \ldots, n && (2) \\
& z_{ij}^i \ , \ z_{ij}^j \geq 0 && i, j = 1, \ldots, n \\
& x \in K \\
& x \in \{0, 1\}^n
\end{aligned}
$$

The following Observation shows the equivalence between the two linearizations:

Observation 1 *For every feasible solution of the LP relaxation of LF there exists one feasible solution of the LP relaxation of ELF of the same value and vice versa.*

Proof. Let $(\bar{x}, \bar{y})$ be a feasible solution to the LP relaxation of LF. We consider the following solution $(\tilde{x}, \tilde{z}^i, \tilde{z}^j)$ of the LP relaxation of ELF: $\tilde{x} = \bar{x}$, $\tilde{z}^i = 1 - \bar{x}_i$, $\tilde{z}^j = \bar{x}_i - \bar{y}$. It is easy to check that it is feasible and it has the same value. Now let $(\tilde{x}, \tilde{z}^i, \tilde{z}^j)$ be a feasible solution to the LP relaxation of ELF. We consider the following solution $(\bar{x}, \bar{y})$ of the LP relaxation of LF: $\bar{x} = \tilde{x}$, $\bar{y} = 1 - \tilde{z}_i - \tilde{z}_j$. Again, it is easy to check that it is feasible and it has the same value.□

Observation 1 implies that every valid inequality for LF is also valid for ELF. This linearization is inspired by the work presented in [13], where the authors proved that in the case of QSSP with non-negative Q_{ij}, ELF without inequalities (1)-(2) is a valid linearization. Their proof can be viewed as a special case of the following Observation:

Observation 2 *Eliminating from ELF the inequalities* (1)-(2)*, corresponding to a couple of indices i, j with non-negative Q_{ij}, does not change the set of optimal solutions of the LP relaxation of ELF.*

Proof. Let $(\tilde{x}, \tilde{z}^i, \tilde{z}^j)$ be a solution to LP relaxation of ELF satisfying all inequalities except some of the (1)-(2). Let $x_i + z^i_{ij} + z^j_{ij} \geq 1$ be one of the violated inequalities, this means that $\tilde{x}_i + \tilde{z}^i_{ij} + \tilde{z}^j_{ij} < 1$, hence one value $\hat{z}^i > \tilde{z}^i$ exists such that $\tilde{x}_i + \hat{z}^i_{ij} + \tilde{z}^j_{ij} = 1$. The point $(\tilde{x}, \hat{z}^i, \tilde{z}^j)$ is still feasible and with better objective function (because Q_{ij} is non positive). This means that $(\tilde{x}, \tilde{z}^i, \tilde{z}^j)$ is not an optimal solution. Hence the inequalities (1)-(2) corresponding to the couple of indices i, j with non-negative Q_{ij} are redundant □.

In [13] the authors showed that, with only non negative Q_{ij}, the QSSP reduces to a Linear Stable Set Problem on an suitable extended graph. In case of general Q, the stable set structure of the problem is modified by the additional constraints (1)-(2)*, in other words, the convex hull of the feasible solution of ELF can be viewed as the convex hull of the stable set polytope intersected with additional constraints. Observation 1 implies that the new linearization does not provide better bounds than the ones provided by the standard RLT inequalities at the price of more constraints and variables. On the other hand, if it is applied to QSSP the structure of the extended formulation is more easily handled by a MILP solver, as will be illustrated in the computational section.*

In order to tackle the LF and ELF, many generic MILP solvers are available, such as CPLEX, Gurobi or SCIP.

4 SDP Reformulation

BQP is equivalent to the following reformulation:

$$
\begin{aligned}
\min\ & \langle \tilde{Q}, Y \rangle \\
& Y = \begin{pmatrix} 1 \\ \bar{x} \end{pmatrix} \begin{pmatrix} 1 \\ \bar{x} \end{pmatrix}^\top && (3) \\
& \bar{x} \in \bar{K} \\
& \bar{x} \in \{-1, 1\}^n && (4)
\end{aligned}
$$

obtained after applying the linear transformation $\bar{x}_i = 2x_i - 1$, imposing

$$
\tilde{Q} = \begin{pmatrix} \frac{1}{2}\sum_{j=1}^n L_i + \frac{1}{4}\sum_{i=1}^n \sum_{j=1}^n Q_{ij} & \frac{1}{4}(L + \frac{1}{2}\sum_{j=1}^n Q_j)^\top \\ \frac{1}{4}(L + \frac{1}{2}\sum_{j=1}^n Q_j) & \frac{1}{4}Q \end{pmatrix}
$$

with Q_j the j-th column of Q and modifying K in $\bar{K}$ accordingly.

For the QSSP we have: $\bar{K} = \{x \in R^n : x_i + x_j \leq 0, \forall \{i,j\} \in E\}$.

Constraints (3) and (4) together can be rewritten as $Y \succeq 0$, $diag(Y) = e$ and $rank(Y) = 1$. Leading to the following equivalent formulation:

$$
\begin{aligned}
\min\ & \langle \tilde{Q}, Y \rangle \\
& Y \succeq 0 \\
& rank(Y) = 1 && (5) \\
& diag(Y) = e \\
& Y_0 \in \bar{K} && (6)
\end{aligned}
$$

with e being the all-ones vector and Y_0 being the first row of Y without the first element (note that $Y_0 = x$). By relaxing the rank constraints (5) we obtain an SDP relaxation which provides valid bounds for all the BQP formulations. Note that this relaxation can be weakened by eliminating constraints (6) or strengthened by adding valid inequalities for the LF, for the convex hull of $\bar{K}$ and for the convex hull of their intersection. Furthermore the SDP relaxation is computationally sensitive to the variable fixing during the BC tree explorations and it becomes easier to compute thanks to the reduction of the problem dimension. Finally, in order to tackle the SDP relaxation, there are different solvers available in the literature, for instance [7] and [18].

5 Hybrid Bounding Procedure

The Hybrid Bounding Procedure is the idea of mixing two different relaxation, i.e. the LP relaxation and the SDP relaxation. As previously underlined there are many different possible ways of deriving an SDP relaxation for a BQP. In the following we will define with Ψ a generic version of the SDP relaxation. Furthermore we introduce another function called Ω which controls the fact of performing function Ψ in a specific node or not. This is due to the fact that potentially the computation of Ψ can be time consuming and hence it has to be used ideally only if it is worth it, in other words, if it prunes the node. These two functions are available in each node of the branch and bound tree:

- Function Ψ is the SDP relaxation that takes as input Q, L and a partial fixing of the variables and returns a bound (LB^{SDP}) on the original objective function.
- Function Ω is an oracle that takes as input all the information about the current node and returns a binary variable indicating if Ψ has to be used or not.

The ideal node processing is represented in Algorithm 1, where UB is the incumbent best feasible solution. In addition to the effectiveness of the oracle Ω

input: best incumbent solution of value UB and current variable fixing;
solve the LP relaxation and get the bound LB^{LP};
if ($LB^{LP} \geq UB$) **then** fathom current node.
OK $\leftarrow \Omega$.
if (OK = FALSE) **then** continue branching.
solve Ψ and get the bound LB^{SDP}.
if ($LB^{SDP} \geq UB$) **then** fathom current node.
else continue branching.

Algorithm 1: Processing at each decision node.

a second subtle aspect has to be taken into account: even if the node is pruned the cost of pruning (in terms of computational time) via Ψ has to be lower than the cost of pruning via exploration of the underlying subtree. In Section 6.3 we measure the effectiveness of different options of Ψ. Finally several strategies for Ω are discussed in order to improve the overall efficiency.

6 Computational Experiments

We performed an extensive computational evaluation of the mathematical formulations of Sections 2 and 3, in order to define the best one for the QSSP. Then we tested different possible options and strategies for the Hybrid LP/SDP bounding procedure described in Section 5. All algorithms were coded in C, and run on a PC with a Intel(R) Xeon(R) E5-2670 at 2.6 GHz and 64 GB RAM memory, under Linux. The optimization software used in our test was Cplex 12.4 single thread. The SDP solver used is CSDP (described in [7]) and it was inserted in the Cplex framework using the `callback` functions of the `Callable Library` of Cplex.

6.1 Testbed description

We randomly generated a set of QSSP instances, the goal was to have a testbed with statistical relevance (several instances with the same features) and to have the complete control of the characteristics of the different classes of instance proposed. The instance generator produces random graphs according to the desired number of vertices n and density μ (which imply a number of edges equal to $\lfloor \mu \times \frac{n(n-1)}{200} \rfloor$). The linear and quadratic costs take a uniformly random integer value in the interval $[1, 100]$. A third parameter ν represents the percentage of costs that are changed in sign from positive to negative. We generated 45 instances by considering all combinations of:

- number of vertices: $n \in \{40, 50, 60, 70, 80\}$;
- density of edges: $\mu \in \{25\%, 50\%, 75\%\}$;
- percentage of negative costs: $\nu \in \{25\%, 50\%, 75\%\}$.

In addition we created 10 instances with the same characteristics using different random seeds, thus obtaining in total 450 QSSP instances. The whole set of instances is available upon request to the authors. In the next sections we discuss the computational outcome of the experiments.

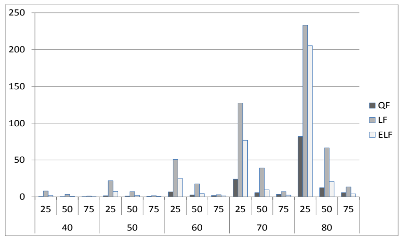
6.2 Identifying the best mathematical formulations

The goal of this section is to computationally evaluate the different MIP formulations, i.e. QF, LF and ELF. In order to do that, we used Cplex with default parameter settings. Figure (a) and (b) show the computing time required for the optimization, dividing the instances by item number and then by density or percentage of negative cost. Figure (c) and figure (d) respectively show the node numbers instead, and the vertical axes are in logarithmic scale. As far as the computing time is concerned, on average the best mathematical formulation is the QF. When comparing the linearized formulation we see that the ELF clearly outperforms the LF. Furthermore, the ELF for some specific subsets of instances outperforms the QF. These are the cases where the density is high. As far as the number of nodes is concerned, here we see the tendency for the QF to explore far more nodes than LF and ELF. LF and ELF on average explore the same amount of nodes, but the ELF is faster at processing each node. These experiments allow us to conclude that LF is dominated by ELF and hence it can be excluded from following tests. QF is better than ELF on average, so we will mainly focus on this.

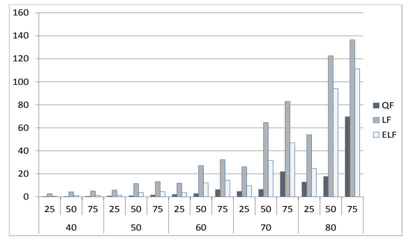
6.3 Different options and strategies for Ψ and Ω

We propose the following two options for Ψ.

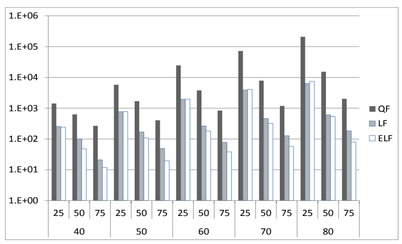
- `Constrained`: corresponds to the SDP relaxation described in Section 4.
- `Unconstrained`: is the same as `Constrained` but without edge constraints (6).



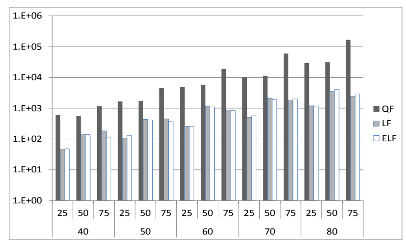
(a) *BC-Time* n,μ



(b) *BC-Time* n,ν



(c) *BC-Nodes* n,μ



(d) *BC-Nodes* n,ν

Fig. 1. Comparison between different Mathematical Formulations (QF- LF- ELF)

In the context of pursuing efficiency, different levels of tolerance (i.e. relative difference between primal and dual values) has been tried. Thanks to preliminary tests, imposing a tolerance of 1% turned out to be the best compromise.
Ideally the SDP relaxation should be performed in the cases in which it helps in pruning. To cope with these problems, we tried seven different strategies for the Oracle Ω. The strategies used are:

- 1 `Always`. The SDP relaxation Ψ is triggered at each node of the BC tree.
- 2 `OnOne`. Ψ is triggered every time we branch on one.
- 3 `UnderAverage`. We trigger Ψ if the current integrality gap is lower than the average integrality gap of the nodes explored so far.
- 4 `OverAverage`. It triggers Ψ if the current integrality gap is bigger than the average.
- 5 `SmallGap`. We trigger Ψ only when the integrality gap is within $[0\%, 5\%]$.
- 6 `MediumGap`. We trigger Ψ only when the integrality gap is within $[5\%, 30\%]$.
- 7 `Random`. Every time we have a random 50% probability of activating Ψ.

Let UB be the best solution so far and LB the current bound provided by the LP relaxation, the corresponding integrality gap is $100 \times \frac{UB-LB}{UB}$. Strategies `OverAverage` and `MediumGap` tend to prune the branch and bound tree at the first levels and they are effective in the case in which the SDP relaxation is much stronger than the LP relaxation, and/or they tend not to prune in the same points of the branching tree. Strategies `UnderAverage` and `SmallGap` tend to prune the nodes at the final levels and are effective in the cases in which the LP relaxation and the SDP relaxation are strong in the same nodes. Finally, strategy `Random` serves as a term of comparison for the previous strategies. In addition to the mentioned strategies we included a bound on the maximum size of the reduced dimension RD. More precisely we fixed $RD = 40, 50, 60$, i.e. we trigger Ψ if and only if the residual dimension after some branching is less than a given value.

6.4 Identifying the best Ψ and Ω

Let BF and ELF be the basic formulations and BF(Ψ,Ω) and ELF(Ψ,Ω) be the same formulations using a given couple of SDP relaxation and Oracle. Four indicators are collected for assessing the impact of a strategy:

$$\delta = \frac{nod_{BF(\Psi,\Omega)}}{nod_{BF}}, \quad \tau = \frac{t_{BF(\Psi,\Omega)}}{t_{BF}}, \quad \pi = \frac{\hat{t}_{BF(\Psi,\Omega)}}{t_{BF}}, \quad \beta = \frac{t_{SDP}}{t_{total}}$$

with t being the total computation time and nod the number of nodes. $\hat{t}$ represents the "ideal" computation time, in other words the total time minus the time used for non-pruning SDP relaxation Ψ. We will further analyze the results concerning BF and BF(Ψ,Ω), and in the Appendix we show the results concerning ELF and ELF(Ψ,Ω). The behavior of LF(Ψ,Ω) is not investigated because it is dominated by the other two options. All the indicators are reported in Tables 3 and 4, the first one concerns the `Constrained` option for Ψ and the second one the `Unconstrained` option. The structure of both tables is identical: vertically each table is split into three main columns, one for each reduced dimension RD; at the second level, each column contains the results concerning the 7 different strategies. Horizontally it is divided into four strips (one for each of the four indicators) and each strip is subdivided according to the instance dimensions n. As explained in Section 6.1, each entry in the table is an average value over 90 instances of similar features. For each strip the average values are provided. Each entry of τ and π is in bold if the ratio is less than one (i.e. if the addition of (Ψ,Ω) improved the performances).

Concerning the node reduction, strategy `Always` is the one giving the best insight about the maximum reduction achievable, in the `Constrained` option we have values of δ in the range of around 0.2 to 0.3, this means that the number of nodes decreases by one third or one forth. If we observe the same entries for the `Unconstrained` mode we see slightly worse ratios, but in the same order of magnitude, especially for bigger instances. This behavior changes strongly if we consider the time ratio τ, in this case moving from `Constrained` to `Unconstrained` greatly improves the performances, allowing to reach values of τ less than one for $RD = 50, 60$. This improvement is even more noticeable as the instance dimension increases, indicating a promising trend for even bigger instances.
Among the six remaining strategies, the best seems to be `UnderAverage`, because

of the average τ when $RD = 60$. Another advantage of `UnderAverage` is its adaptability, avoiding the need to specify additional parameters like in `SmallGap` or `MediumGap`. The rest of the strategies (except from `SmallGap`) seem to perform fairly well, the failure of `SmallGap` is probably due to the fact that when the optimality gap is so small it is better to let the BC finish the exploration instead of using the expensive procedure Ψ. The ratio π for the strategy `Always` gives what in statistics is called "Gold standard", in other words it gives an idea about what the results could be in presence of an "ideal" Ω, i.e. an Oracle that triggers the bounding procedure if and only if it will prune the node. The average value of $\pi = 0.75$ corresponding to $RD = 60$ is an interesting result: it means that there is still room for decreasing the running time by 25% just with the Oracle strategy, hence even if `UnderAverage` performs well, there is still room for improvement.
In addition to the Oracle, another aspect to take into account when evaluating possible improvements in our approach is the percentage of time used by the SDP solver with respect to the global solution time. The indicator β measures exactly this. For the `Constrained` option and strategy `Always`, slightly more than 80% of the time is spent in solving an SDP. This gives a clear idea about why `Constrained` is too heavy a bounding procedure. For `Unconstrained` this value goes down to around 45%. Having one value $\beta = 0.45$ gives an intuition about how strongly the solution time is connected to the quality of the state-of-the-art SDP solvers. In addition, a further reduction of computational time could be achieved by the fully integration of the SDP into the Cplex BC framework. Concerning RD, the best performances are obtained in correspondence of size equal to 60, this confirms the idea that it is worth using the SDP relaxation in the early level of the BC tree. In Appendix A the results concerning ELF are shown, the main problem is that the inner bounding procedure of the MILP solver is already pruning many nodes, making the use of Ψ less relevant. In order to avoid that it would be interesting to "soften" the bounding procedure of the MILP solver (by for example deactivating a consistent part of the cut separator). In the full version of the paper we will include also these types of tests.

7 Conclusions

In this paper we have explored the use of SDP-based bounding procedures within the BC framework provided by the MINLP solvers. In order to do that, we performed an extensive computational analysis on the QSSP that allows us to conclude that the Hybrid LP/SDP Bounding Procedure allows a noticeable reduction of computing time and a cut of almost one order of magnitude for the branching nodes. The SDP bounds help in pruning but are heavy to compute, and thus in this optic we proposed different strategies in order to make the Hybrid bounding procedure more efficient.

Furthermore we proposed and evaluated a different linearization technique that turned out to be computationally more efficient when compared to the clas-

dim	*40*							*50*							*60*						
str	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*1*	*2*	*3*	*4*	*5*	*6*	*7*
δ																					
40	0.13	0.31	0.14	0.54	0.90	0.44	0.27														
50	0.16	0.37	0.17	0.52	0.89	0.44	0.29	0.11	0.27	0.12	0.39	0.89	0.43	0.23							
60	0.34	0.58	0.34	0.74	0.91	0.58	0.47	0.11	0.30	0.12	0.39	0.91	0.45	0.23	0.08	0.23	0.09	0.31	0.90	0.44	0.19
70	0.56	0.74	0.56	0.88	0.93	0.72	0.66	0.24	0.48	0.24	0.61	0.90	0.51	0.36	0.09	0.25	0.10	0.32	0.90	0.44	0.19
80	0.69	0.82	0.70	0.92	0.97	0.83	0.76	0.39	0.62	0.40	0.75	0.91	0.64	0.51	0.16	0.38	0.17	0.48	0.90	0.50	0.27
	0.38	0.56	0.38	0.72	0.92	0.60	0.49	0.21	0.42	0.22	0.54	0.90	0.51	0.33	0.11	0.29	0.12	0.37	0.90	0.46	0.22
τ																					
40	5.64	1.81	5.30	2.60	1.42	1.57	3.51														
50	2.95	1.55	2.25	2.16	1.41	1.41	2.19	6.76	1.73	3.74	5.35	1.41	1.42	4.19							
60	1.85	1.35	1.55	1.44	1.34	1.30	1.52	3.56	1.39	1.98	3.06	1.38	1.29	2.45	7.58	1.46	2.97	6.75	1.38	1.29	4.58
70	1.42	1.16	1.25	1.21	1.20	1.13	1.26	2.32	1.38	1.50	2.03	1.38	1.27	1.77	4.88	1.34	1.80	4.62	1.40	1.25	2.96
80	1.19	1.05	1.09	1.15	1.10	1.05	1.13	1.75	1.19	1.26	1.54	1.25	1.15	1.43	3.18	1.23	1.34	2.97	1.32	1.18	2.08
	2.61	1.39	2.29	1.71	1.29	1.29	1.92	3.60	1.42	2.12	3.00	1.35	1.28	2.46	5.21	1.34	2.03	4.78	1.37	1.24	3.21
π																					
40	1.21	1.43	1.23	1.08	1.41	1.47	1.29														
50	1.22	1.45	1.23	1.06	1.40	1.37	1.27	1.11	1.40	1.13	1.01	1.40	1.38	1.18							
60	1.26	1.34	1.28	1.02	1.34	1.29	1.23	**1.00**	1.27	1.02	**0.96**	1.38	1.27	1.10	**0.94**	1.20	**0.96**	**0.92**	1.38	1.27	1.05
70	1.17	1.15	1.17	1.01	1.20	1.13	1.14	1.16	1.35	1.18	**0.97**	1.38	1.25	1.18	**0.92**	1.21	**0.93**	**0.91**	1.40	1.23	1.02
80	1.05	1.05	1.06	1.02	1.10	1.05	1.05	1.14	1.18	1.15	**0.97**	1.25	1.15	1.12	**1.00**	1.19	**1.00**	**0.91**	1.32	1.17	1.01
	1.18	1.29	1.20	1.04	1.29	1.26	1.20	1.10	1.30	1.12	0.98	1.35	1.26	1.14	0.95	1.20	0.96	0.91	1.37	1.22	1.03
β																					
40	0.91	0.61	0.90	0.71	0.34	0.52	0.83														
50	0.81	0.51	0.70	0.61	0.31	0.46	0.72	0.93	0.56	0.85	0.84	0.32	0.47	0.86							
60	0.65	0.43	0.55	0.35	0.28	0.39	0.53	0.86	0.50	0.68	0.75	0.29	0.44	0.77	0.94	0.54	0.83	0.89	0.30	0.45	0.88
70	0.45	0.30	0.37	0.18	0.20	0.25	0.33	0.75	0.46	0.56	0.55	0.28	0.41	0.63	0.90	0.50	0.66	0.85	0.28	0.43	0.82
80	0.28	0.17	0.20	0.11	0.11	0.13	0.20	0.62	0.38	0.43	0.35	0.23	0.32	0.48	0.82	0.48	0.57	0.69	0.26	0.41	0.72
	0.62	0.41	0.55	0.39	0.25	0.35	0.52	0.79	0.47	0.63	0.62	0.28	0.41	0.69	0.89	0.51	0.68	0.81	0.28	0.43	0.81

Table 1. QF + `Constrained` SDP (Ψ)

RD	*40*							*50*							*60*						
str	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*1*	*2*	*3*	*4*	*5*	*6*	*7*
δ																					
40	0.27	0.47	0.28	0.75	0.91	0.53	0.45														
50	0.25	0.48	0.26	0.71	0.89	0.50	0.42	0.21	0.41	0.22	0.63	0.89	0.49	0.39							
60	0.37	0.62	0.38	0.82	0.91	0.60	0.52	0.19	0.40	0.20	0.60	0.91	0.48	0.35	0.16	0.36	0.18	0.55	0.91	0.47	0.33
70	0.57	0.75	0.58	0.91	0.93	0.72	0.68	0.28	0.52	0.29	0.72	0.90	0.53	0.43	0.15	0.36	0.17	0.51	0.90	0.46	0.31
80	0.70	0.82	0.71	0.93	0.97	0.83	0.78	0.41	0.63	0.42	0.82	0.91	0.64	0.54	0.20	0.44	0.21	0.59	0.91	0.51	0.34
	0.43	0.63	0.44	0.82	0.92	0.64	0.57	0.27	0.49	0.28	0.69	0.90	0.53	0.43	0.17	0.39	0.19	0.55	0.90	0.48	0.33
τ																					
40	1.81	1.34	1.66	1.35	1.24	1.40	1.48														
50	1.35	1.19	1.25	1.15	1.19	1.26	1.23	1.51	1.20	1.32	1.29	1.19	1.27	1.29							
60	1.08	1.10	1.07	1.03	1.14	1.07	1.06	1.15	1.04	1.05	1.11	1.16	1.07	1.09	1.22	1.03	1.06	1.17	1.16	1.07	1.12
70	1.02	1.03	1.02	1.01	1.08	1.00	1.02	**0.99**	1.03	**0.97**	1.02	1.13	1.01	1.01	**0.96**	**0.95**	**0.94**	1.03	1.14	**0.99**	**1.00**
80	**1.00**	**1.00**	**1.00**	1.02	1.05	1.01	1.00	**0.95**	**0.98**	**0.95**	**1.00**	1.10	**0.97**	**0.97**	**0.86**	**0.93**	**0.82**	**0.94**	1.12	**0.92**	**0.88**
	1.25	1.13	1.20	1.11	1.14	1.15	1.16	1.15	1.06	1.07	1.10	1.14	1.08	1.09	1.01	0.97	0.94	1.05	1.14	0.99	1.00
π																					
40	**0.98**	1.08	**0.99**	**0.92**	1.20	1.14	1.00														
50	**0.93**	1.08	**0.95**	**0.92**	1.18	1.06	**0.98**	**0.87**	1.01	**0.91**	**0.87**	1.17	1.06	**0.93**							
60	**0.96**	1.07	**0.97**	**0.96**	1.13	1.01	**0.98**	**0.80**	**0.94**	**0.81**	**0.85**	1.16	**0.97**	**0.88**	**0.76**	**0.89**	**0.78**	**0.81**	1.15	**0.97**	**0.85**
70	**0.99**	1.03	1.00	**0.99**	1.08	**0.99**	1.00	**0.86**	1.00	**0.87**	**0.92**	1.13	**0.96**	**0.93**	**0.71**	**0.87**	**0.73**	**0.81**	1.14	**0.92**	**0.82**
80	**0.99**	**1.00**	**0.99**	1.01	1.05	1.01	**1.00**	**0.91**	**0.97**	**0.92**	**0.96**	1.10	**0.97**	**0.94**	**0.75**	**0.90**	**0.76**	**0.85**	1.12	**0.90**	**0.82**
	0.97	1.05	0.98	0.96	1.13	1.04	0.99	0.86	0.98	0.88	0.90	1.14	0.99	0.92	0.74	0.89	0.76	0.82	1.14	0.93	0.83
β																					
40	0.65	0.53	0.63	0.37	0.31	0.46	0.55														
50	0.51	0.45	0.48	0.27	0.27	0.41	0.44	0.58	0.48	0.52	0.38	0.28	0.42	0.49							
60	0.40	0.40	0.39	0.10	0.24	0.33	0.34	0.49	0.45	0.46	0.30	0.26	0.40	0.43	0.54	0.46	0.47	0.37	0.26	0.40	0.46
70	0.26	0.26	0.25	0.03	0.16	0.19	0.19	0.40	0.41	0.40	0.15	0.23	0.35	0.35	0.44	0.45	0.45	0.30	0.24	0.39	0.41
80	0.14	0.15	0.14	0.02	0.08	0.10	0.10	0.31	0.33	0.31	0.07	0.19	0.26	0.25	0.38	0.41	0.35	0.18	0.22	0.35	0.32
	0.39	0.36	0.38	0.16	0.21	0.30	0.33	0.44	0.42	0.42	0.23	0.24	0.36	0.38	0.45	0.44	0.42	0.28	0.24	0.38	0.40

Table 2. QF + `Unconstrained` SDP (Ψ)

sical one for several classes of QSSP instances. Finally, the SDP relaxation can be used to enhance any of the essential ingredients of a generic purpose MINLP solver, which are: bounding techniques, primal heuristics and branching strategies. Specifically the information provided by the solution of the SDP relaxation at each branching node can be used either to derive alternative branching strategies, to compute different heuristic solutions or to strengthen LP relaxation for pruning the node. In this paper we have focused on this last aspect, deriving effective Hybrid Bounding Procedures. We leave the other two aspects for further development.

References

1. G. Agnarsson, M. Halldrsson, and E. Losievskaja. Sdp-based algorithms for maximum independent set problems on hypergraphs. In *Automata, Languages and Programming*, volume 5555 of *LNCS*, pages 12–23. 2009.
2. K. M. Anstreicher. Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. *J. Global Optimization*, 43(2-3):471–484, 2009.
3. BARON, 2012. http://archimedes.cheme.cmu.edu/?q=baron.
4. BiqCrunch, 2012. http://www-lipn.univ-paris13.fr/BiqCrunch/.
5. Biqmac, 2012. http://biqmac.uni-klu.ac.at/.
6. Bonmin, 2012. https://projects.coin-or.org/Bonmin.
7. B. Borchers. CSDP, A C library for semidefinite programming. *Optimization Methods and Software*, 11:613–623, 1999.
8. S. Burer and D. Vandenbussche. Globally solving box-constrained nonconvex quadratic programs with semidefinite-based finite branch-and-bound. *Computational Optimization and Applications*, 43:181–195, 2009.
9. Cplex, 2012. http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/.
10. M. Giandomenico, A. N. Letchford, F. Rossi, and S. Smriglio. A new approach to the stable set problem based on ellipsoids. In *IPCO*, volume 6655 of *LNCS*, pages 223–234. Springer Berlin Heidelberg, 2011.
11. GloMIQO, 2012. http://helios.princeton.edu/GloMIQO/index.html.
12. Gurobi, 2012. http://www.gurobi.com/.
13. B. Jaumard, O. Marcotte, and C. Meyer. *Estimation of the Quality of Cellular Networks Using Column Generation Techniques*. Cahiers du GÉRAD. Groupe d'études et de recherche en analyse des décisions, 1998.
14. B. Jaumard, O. Marcotte, C. Meyer, and T. Vovor. Comparison of column generation models for channel assignment in cellular networks. *Discrete Applied Mathematics*, 112(13):217 – 240, 2001.
15. N. Krislock, J. Malick, and F. Roupin. Improved semidefinite bounding procedure for solving max-cut problems to optimality. *To appear in Mathematical Programming A*, 2012.
16. F. Mahdavi Pajouh, B. Balasundaram, and O. Prokopyev. On characterization of maximal independent sets via quadratic optimization. *Journal of Heuristics*, pages 1–16, 2011.
17. SCIP, 2012. http://scip.zib.de/.
18. SeDuMi, 2012. http://sedumi.ie.lehigh.edu/.
19. H. D. Sherali and W. P. Adams. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*. Springer, 1998.

A Appendix

dim	*40*							*50*							*60*						
str	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*1*	*2*	*3*	*4*	*5*	*6*	*7*
δ																					
40	0.83	0.86	0.84	0.99	0.96	0.88	0.92														
50	0.87	0.89	0.87	0.99	0.96	0.90	0.92	0.80	0.82	0.80	0.99	0.96	0.86	0.89							
60	0.95	0.95	0.95	1.00	0.99	0.97	0.97	0.82	0.85	0.82	1.00	0.96	0.87	0.89	0.78	0.80	0.78	0.99	0.96	0.85	0.88
70	0.98	0.99	0.98	1.00	1.00	0.99	1.00	0.91	0.91	0.91	0.99	0.97	0.94	0.95	0.79	0.81	0.79	0.98	0.95	0.87	0.89
80	0.99	0.99	0.99	0.99	1.00	0.99	0.99	0.95	0.95	0.95	0.99	0.99	0.97	0.97	0.87	0.89	0.87	0.98	0.97	0.91	0.91
	0.92	0.94	0.92	0.99	0.98	0.95	0.96	0.87	0.88	0.87	0.99	0.97	0.91	0.92	0.81	0.84	0.81	0.99	0.96	0.88	0.89
τ																					
40	2.55	1.19	2.46	1.38	1.07	1.51	1.69														
50	1.78	1.07	1.76	1.13	1.11	1.44	1.42	3.99	1.15	3.83	1.67	1.11	1.81	2.45							
60	1.30	1.03	1.31	1.04	1.06	1.19	1.19	2.91	1.09	2.74	1.20	1.11	1.78	1.99	6.37	1.15	5.78	2.26	1.12	1.94	3.85
70	1.12	1.02	1.12	1.03	1.03	1.10	1.07	1.85	1.04	1.80	1.07	1.11	1.40	1.44	4.76	1.08	4.36	1.42	1.14	2.35	2.87
80	1.07	1.02	1.07	1.02	1.03	1.05	1.04	1.41	1.02	1.39	1.04	1.06	1.23	1.21	3.07	1.05	2.83	1.12	1.10	1.67	1.99
	1.56	1.07	1.54	1.12	1.06	1.26	1.28	2.54	1.08	2.44	1.24	1.10	1.56	1.77	4.73	1.09	4.32	1.60	1.12	1.99	2.90
π																					
40	1.06	1.07	1.07	1.02	1.03	1.03	1.02														
50	1.03	1.03	1.05	1.02	1.04	1.03	1.02	1.03	1.03	1.05	1.01	1.03	1.02	**1.00**							
60	1.02	1.02	1.03	1.02	1.03	1.02	1.02	1.02	1.03	1.03	1.01	1.03	1.02	1.01	1.01	1.02	1.02	1.01	1.03	1.01	1.01
70	1.01	1.02	1.02	1.02	1.02	1.02	1.02	1.01	1.02	1.02	1.02	1.02	1.01	1.01	**1.00**	1.01	1.00	1.01	1.02	1.01	1.02
80	1.02	1.02	1.02	1.02	1.02	1.02	1.02	1.01	1.01	1.01	1.02	1.02	1.02	1.01	**1.00**	1.01	**0.99**	1.01	1.02	1.00	**0.99**
	1.03	1.03	1.04	1.02	1.03	1.03	1.02	1.02	1.02	1.03	1.01	1.02	1.02	1.01	1.00	1.01	1.01	1.01	1.02	1.01	1.01
β																					
40	0.51	0.16	0.49	0.22	0.06	0.28	0.35														
50	0.38	0.08	0.37	0.05	0.07	0.25	0.26	0.61	0.16	0.56	0.32	0.08	0.32	0.46							
60	0.20	0.03	0.19	0.02	0.04	0.12	0.12	0.53	0.10	0.48	0.12	0.07	0.30	0.39	0.66	0.16	0.60	0.42	0.08	0.33	0.53
70	0.08	0.01	0.08	0.00	0.01	0.06	0.05	0.36	0.04	0.33	0.04	0.06	0.21	0.23	0.62	0.12	0.54	0.21	0.09	0.35	0.49
80	0.04	0.00	0.04	0.00	0.01	0.03	0.02	0.21	0.01	0.20	0.02	0.03	0.13	0.13	0.51	0.06	0.45	0.08	0.07	0.27	0.36
	0.24	0.06	0.23	0.06	0.04	0.15	0.16	0.43	0.08	0.39	0.12	0.06	0.24	0.30	0.60	0.11	0.53	0.23	0.08	0.32	0.46

Table 3. ELF + `Constrained` SDP (Ψ)

dim	*40*							*50*							*60*						
str	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*1*	*2*	*3*	*4*	*5*	*6*	*7*
δ																					
40	0.93	0.94	0.93	0.99	0.99	0.95	0.96														
50	0.94	0.94	0.94	1.00	0.98	0.96	0.97	0.91	0.92	0.92	1.00	0.98	0.95	0.96							
60	0.97	0.97	0.97	1.00	0.99	0.99	0.99	0.92	0.92	0.92	1.00	0.98	0.94	0.96	0.90	0.90	0.90	1.00	0.98	0.93	0.95
70	0.98	0.98	0.98	1.00	1.01	0.99	0.99	0.96	0.96	0.96	1.00	1.01	0.98	0.99	0.91	0.91	0.91	1.00	1.00	0.95	0.96
80	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.97	0.98	0.97	0.99	1.00	0.98	0.99	0.94	0.94	0.94	0.99	0.98	0.96	0.97
	0.96	0.96	0.96	1.00	0.99	0.98	0.98	0.94	0.94	0.94	1.00	0.99	0.96	0.98	0.92	0.92	0.92	1.00	0.99	0.95	0.96
τ																					
40	1.15	1.05	1.13	1.04	1.04	1.08	1.09														
50	1.07	1.03	1.07	1.03	1.03	1.05	1.05	1.12	1.02	1.10	1.03	1.01	1.05	1.06							
60	1.03	1.02	1.04	1.03	1.02	1.03	1.03	1.08	1.03	1.07	1.03	1.02	1.04	1.05	1.12	1.03	1.09	1.04	1.01	1.03	1.06
70	1.02	1.02	1.02	1.02	1.02	1.03	1.03	1.04	1.02	1.04	1.02	1.02	1.03	1.03	1.08	1.02	1.07	1.03	1.03	1.05	1.05
80	1.02	1.02	1.03	1.02	1.02	1.02	1.02	1.02	1.01	1.02	1.02	1.02	1.02	1.02	1.04	1.01	1.03	1.02	1.02	1.03	1.03
	1.06	1.03	1.06	1.03	1.03	1.04	1.04	1.06	1.02	1.06	1.02	1.02	1.03	1.04	1.08	1.02	1.06	1.03	1.02	1.04	1.05
π																					
40	**0.99**	**0.99**	**0.98**	1.01	1.03	1.02	1.01														
50	**0.99**	**1.00**	1.00	1.02	1.02	1.01	1.01	**0.96**	**0.97**	**0.97**	**1.00**	**0.99**	**0.99**	**0.99**							
60	1.00	1.01	1.02	1.03	1.02	1.02	1.02	**0.99**	**1.00**	**1.00**	1.02	1.01	**1.00**	1.00	**0.98**	**0.98**	**0.98**	1.01	1.00	**0.98**	**0.99**
70	1.01	1.02	1.02	1.02	1.02	1.02	1.02	1.00	1.01	1.00	1.01	1.02	1.01	1.02	**0.99**	**0.99**	**1.00**	1.02	1.02	1.01	1.01
80	1.02	1.02	1.02	1.02	1.02	1.02	1.02	1.00	1.01	1.01	1.02	1.02	1.02	1.01	**1.00**	**1.00**	**1.00**	1.02	1.01	1.01	1.01
	1.00	1.01	1.01	1.02	1.02	1.02	1.02	0.99	0.99	1.00	1.01	1.01	1.00	1.01	0.99	0.99	0.99	1.01	1.01	1.00	1.00
β																					
40	0.15	0.08	0.13	0.03	0.02	0.07	0.08														
50	0.08	0.04	0.07	0.01	0.01	0.04	0.04	0.15	0.07	0.12	0.03	0.02	0.06	0.08							
60	0.02	0.01	0.02	0.00	0.00	0.01	0.01	0.09	0.04	0.08	0.01	0.01	0.05	0.05	0.14	0.06	0.11	0.03	0.01	0.05	0.07
70	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.04	0.02	0.04	0.00	0.01	0.02	0.02	0.09	0.04	0.08	0.02	0.01	0.04	0.05
80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.00	0.01	0.01	0.04	0.02	0.04	0.01	0.01	0.02	0.02
	0.05	0.03	0.05	0.01	0.01	0.03	0.03	0.07	0.03	0.06	0.01	0.01	0.03	0.04	0.09	0.04	0.08	0.02	0.01	0.04	0.05

Table 4. ELF + `Unconstrained` SDP (Ψ)