Combining Discrete Ellipsoid-Based Search and Branch-and-Cut for Binary Quadratic Programming Problems

Wen-Yang Ku and J. Christopher Beck

Department of Mechanical & Industrial Engineering
University of Toronto, Toronto, Ontario M5S 3G8, Canada
{wku,jcb}@mie.utoronto.ca

Abstract. We propose a hybrid algorithm that combines discrete ellipsoid-based search (DEBS) and a branch-and-cut (B&C) MIP solver to solve binary quadratic programming (BQP) problems, an important class of optimization problems with a number of practical applications. We perform experiments on benchmark instances for the BQP problem and compare the performance of two B&C based solvers, the DEBS method that is commonly used in the communications community, and the new hybrid algorithm. Our experimental results demonstrate that the new hybrid algorithm outperforms both the well-known MIP solvers and the DEBS approach. Further comparison against two state-of-the-art special-purpose algorithms in the literature demonstrates that the hybrid approach is competitive: achieving the same or better performance on six of seven benchmark sets against one algorithm and performing competitively against the semi-definite programming (SDP) based algorithm for moderate size problems and some dense problems, while under-performing on larger problems.

1 Introduction

Binary quadratic programming (BQP) problems arise in many combinatorial optimization problems such as task allocation [1], quadratic assignment [2], and max-cut problems [3]. A variety of exact methods exist for solving BQP problems including the linearization method [4], discrete ellipsoid-based search (DEBS) [5,6], and mixed integer programming (MIP) [7,8].

Semi-definite programming (SDP) based branch-and-bound approaches [9,3] are often regarded as the state-of-the-art approach for solving BQPs. In this approach, the semi-definite relaxation bound of the objective function is used to prune nodes during the branch-and-bound process. Krislock et al. [3] showed that their SDP algorithm dominates existing approaches for the BQP problems in the Biq-Mac library [10].

Recently, Li et al. [11] proposed a specialized branch-and-bound approach and demonstrated its strong performance on benchmark instances from a number of sources. Li et al.'s techniques are derived from the geometric structure of the

BQP problem based on perturbation analysis. The results of the analysis are implemented in the form of problem-specific lower bounding techniques and inference rules to fix values. Variable and value ordering heuristics, as well as a primal heuristic to find high quality feasible solutions, are also proposed to accelerate the convergence of the search. Empirical results demonstrate that the algorithm is one of the state-of-the-arts for finding exact solutions to these benchmark BQP problems.

The common generic approach in Operations Research for exactly solving BQPs is the use of a commercial MIP solver such as CPLEX or Gurobi, solvers that have been extended over the past few years to be able to reason about quadratic constraints [12]. A modern MIP solver is able to outperform several other exact approaches [13] and MIP is commonly used as a standard comparison for evaluating heuristic methods [1].

In this paper we develop a new hybrid algorithm that combines discrete ellipsoid-based search (DEBS) and a branch-and-cut (B&C) MIP solver. DEBS is a specialized search used in the communications literature (e.g., see [14]) to solve integer least squares problems based on the clever enumeration of integer points within the hyper-ellipsoid defining the feasible space. As BQPs can be reformulated as integer least squares problems, we perform this transformation and, for the first time, evaluate DEBS on BQP. We then hybridize DEBS with a MIP solver (SCIP [15]), incorporating aspects of DEBS into presolving, into global-constraint-based inference, and as a primal heuristic.

Our experimental results demonstrate that DEBS performs much better than CPLEX on average, though approximately at the same level as default SCIP (without the hybrid extensions) on seven standard benchmark sets. Interestingly, DEBS is both significantly better and significantly worse than SCIP on different problem sets. The hybrid approach, implemented in SCIP, outperforms CPLEX, SCIP, and DEBS on all seven problem sets, though on some sets the improvement is marginal. Li et al.'s approach dominates the B&C solvers and DEBS overall while being comparable to our new hybrid approach: on one problem set Li et al.'s approach is superior, on another the hybrid performs approximately an order-of-magnitude better, and on the remaining five sets, the performance of the two algorithms is essentially equivalent. When compared to the SDP approach, the hybrid algorithm performs as efficiently for moderate size problems but lags behind the SDP approach for large problems. However, on some dense problems, the hybrid algorithm greatly outperforms the SDP approach. Overall, the SDP approach still appears to be the strongest for solving BQPs.

The contribution of this paper is two-fold. First, it is the first study that applies the DEBS approach from communications to BQPs. The somewhat different approach of DEBS compared to standard OR approaches provides new insight into this well-studied problem and may inspire future innovation. Second, we make use of this inspiration to propose a novel, competitive hybrid algorithm that combines DEBS and a B&C MIP solver. A particular advantage of this hybrid approach is that, unlike the SDP solver or Li et al.'s B&B, the hybrid algorithm can be applied to a broader class of problems, such as the unconstrained integer quadratic programming (IQP) problem and IQP problems with general integer bounds on the variables [16].

The rest of the paper is organized as follows. In Section 2 we define the BQP problem. Section 3 presents the necessary background, including a literature review for the DEBS method, B&C MIP solvers for the BQP problem, and previous results. Section 4 describes the hybrid algorithm. Sections 5 and 6 provide computational results and discussions. We conclude in Section 7.

2 Problem Definition

In this section we first define the BQP problem and its equivalent binary integer least squares (ILS) problem that can be solved with the DEBS method. Then we present the transformation between these two problems.

The BQP problem is defined as follows:

$$\min_{\boldsymbol{x} \in \{0,1\}} \frac{1}{2} \boldsymbol{x}^T \boldsymbol{H} \boldsymbol{x} + \boldsymbol{f}^T \boldsymbol{x}, \tag{1}$$

where $\boldsymbol{H} \in \mathbb{R}^{n \times n}$ is a symmetric semi-definite positive matrix and $\boldsymbol{f} \in \mathbb{R}^n$ is a vector.

Current MIP solvers can solve problems with quadratic objectives when the objective is semi-definite posiitve [12]. When $\boldsymbol{H}$ is not symmetric, the symmetric form can be obtained by setting $\boldsymbol{H} = (\boldsymbol{H} + \boldsymbol{H}^T)/2$. Another common issue for real world applications is the non-convexity property of the matrix $\boldsymbol{H}$. Convexifying $\boldsymbol{H}$ is possible in the BQP case since, for every variable, the relationship $x_i = x_i^2$ holds. Therefore, we can perturb the $\boldsymbol{H}$ matrix and make it semi-definite positive by using a vector $\boldsymbol{u}$ until $(\boldsymbol{H} + \mathrm{diag}(\boldsymbol{u}))$ is semi-definite positive. The convex equivalent BQP problem can be obtained as follows:

$$\min_{\boldsymbol{x} \in \{0,1\}} \frac{1}{2} \boldsymbol{x}^T (\boldsymbol{H} + \mathrm{diag}(\boldsymbol{u})) \boldsymbol{x} + \big(\boldsymbol{f} - \frac{1}{2} \boldsymbol{u}\big)^T \boldsymbol{x}. \tag{2}$$

A computationally inexpensive way to find such a $\boldsymbol{u}$ vector consists of computing the eigenvectors of $\boldsymbol{H}$ [13].

To solve the BQP problem with the DEBS method, we transform the BQP problem into its equivalent binary ILS problem through the relationship $\boldsymbol{H} = \boldsymbol{A}^T \boldsymbol{A}$ and $\boldsymbol{f} = -\boldsymbol{y}^T \boldsymbol{A}$. Thus, the binary ILS problem can be defined as follows

$$\min_{\boldsymbol{x} \in \{0,1\}} \|\boldsymbol{y} - \boldsymbol{A} \boldsymbol{x}\|_2^2 \,. \tag{3}$$

3 Background

In this section we present the discrete ellipsoid-based search technique in detail and provide other the necessary background.

3.1 Discrete Ellipsoid-Based Search

Discrete Ellipsoid-Based Search (DEBS)[1] was originally proposed to solve the integer least squares (ILS) problems that arise in some communication applications [14]. For example, to achieve a very high precision in a global navigation satellite system, an ILS problem has to be solved to estimate the double-difference carrier phase ambiguities obtained from pairs of satellite and receivers [14]. Loosely, this problem is to find the integer number of wavelengths between the satellite and the receiver in order to precisely estimate their range. The DEBS method is the most common method for solving the ILS problem to optimality in the communication literature [5].

The DEBS method consists of two phases: reduction and search. The reduction is a preprocessing step that transforms the given ILS problem into one for which the search process is more efficient [5]. Search consists of the enumeration of the integer points in a bounded region [17].

Geometrically, the optimal solution is found by searching discretely inside the ellipsoid defined by the objective function of the binary ILS problem (3). Suppose the optimal solution $\boldsymbol{z}$ to the binary ILS problem (3) satisfies the following bound

$$\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{z}\|_2^2 < \beta, \tag{4}$$

where β is a constant. This is a hyper-ellipsoid with center $\boldsymbol{A}^{-1}\boldsymbol{y}$. The DEBS method systematically searches for the optimal integer solution inside the ellipsoid.

Reduction Reduction can be regarded as a preprocessing technique that transforms $\boldsymbol{A}$ to an upper triangular matrix $\boldsymbol{R}$ with certain properties such that the search is more efficient. It has been shown that the order of the diagonal entries of $\boldsymbol{R}$ can greatly affect the performance of the DEBS method [5]. It is desirable to have the diagonal entries satisfying the following relationship:

$$|r_{11}| \leq |r_{22}| \leq \ldots \leq |r_{kk}| \leq \ldots \leq |r_{n-1,n-1}| \leq |r_{nn}|.$$

In order to keep the solution of the binary ILS problem (3) unchanged, a transformation applied to $\boldsymbol{A}$ from the left-hand side must be an orthogonal matrix. To keep the integer nature of the unknown integer vector and the bounds on the variables unchanged, a transformation to $\boldsymbol{A}$ from the right-hand side has to be a permutation matrix. Therefore, the reduction can be described as the process of transforming $\boldsymbol{A}$ to $\boldsymbol{R}$, by finding an orthogonal matrix $\boldsymbol{Q} \in \mathbb{R}^{m \times m}$ and a permutation matrix $\boldsymbol{P} \in \mathbb{Z}^{n \times n}$ such that

$$\boldsymbol{Q}^T\boldsymbol{A}\boldsymbol{P} = \begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{0}\end{bmatrix}, \quad \boldsymbol{Q} = \left[\boldsymbol{Q}_1, \boldsymbol{Q}_2\right],$$

[1] There does not seem to be a standard name for this approach in the communications literature and so we have adopted this term.

where $\boldsymbol{Q} \in \mathbb{R}^{m\times m}$ is orthogonal, $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ is upper triangular, and $\boldsymbol{P} \in \mathbb{Z}^{n\times n}$ is a permutation matrix. With the QR factorization, we have

$$\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 = \left\|\boldsymbol{Q}_1^T\boldsymbol{y} - \boldsymbol{R}\boldsymbol{P}^T\boldsymbol{x}\right\|_2^2 + \left\|\boldsymbol{Q}_2^T\boldsymbol{y}\right\|_2^2,$$

where

$$\bar{\boldsymbol{y}} = \boldsymbol{Q}_1^T\boldsymbol{y}, \quad \boldsymbol{z} = \boldsymbol{P}^T\boldsymbol{x}, \quad \bar{\boldsymbol{l}} = \boldsymbol{P}^T\boldsymbol{l}, \quad \bar{\boldsymbol{u}} = \boldsymbol{P}^T\boldsymbol{u}.$$

Since the original lower and upper bounds on the variables are all 0 and 1 for the binary ILS problem, the new bounds after applying the permutation matrix $\boldsymbol{P}$ remain the same. Therefore, the original binary ILS problem (3) is transformed to the new reduced binary ILS problem

$$\min_{\boldsymbol{z}\in\{0,1\}} \|\bar{\boldsymbol{y}} - \boldsymbol{R}\boldsymbol{z}\|_2^2\,, \tag{5}$$

where

$$\boldsymbol{z} = \boldsymbol{P}^T\boldsymbol{x}.$$

After the optimal solution $\boldsymbol{z}^*$ to the new binary ILS problem (5) is found, the optimal solution, $\boldsymbol{x}^*$, to the original binary ILS problem (3) can be recovered with the following relationship:

$$\boldsymbol{x}^* = \boldsymbol{P}^T\boldsymbol{z}^*.$$

An effective reduction algorithm for ILS problems with bounds on the variables was proposed by Chang & Han [5], in which different reduction strategies are empirically compared. In our implementation of the DEBS method, we adopt Chang & Han's reduction algorithm for the binary ILS problem.

Search After the reduction phase, we have the equivalent reduced problem (5). To illustrate the search strategy, we first explain the search with the unconstrained ILS problem, i.e., $\boldsymbol{z} \in \mathbb{Z}^n$ instead of $\boldsymbol{z} \in \{0,1\}$ then specialize the search for the binary ILS problem.

Suppose the optimal solution $\boldsymbol{z}$ satisfies the following bound

$$\|\bar{\boldsymbol{y}} - \boldsymbol{R}\boldsymbol{z}\|_2^2 < \beta,$$

or equivalently

$$\sum_{k=1}^{n}(\bar{y_k} - \sum_{j=k}^{n} r_{kj}z_j)^2 < \beta,$$

where β is a constant which can be obtained by substituting any feasible integer solution to equation (5). This is a hyper-ellipsoid with center $\boldsymbol{R}^{-1}\bar{\boldsymbol{y}}$. The goal is to search this ellipsoid to find the optimal integer solution.

Among several search strategies, the Schnorr & Euchner strategy is usually considered the most efficient [17]. Let $\boldsymbol{z}_i^n = [z_i, z_{i+1}, \ldots, z_n]^T$. Define the so far unknown (apart from c_n) and usually non-integer variables

$$c_n = \bar{y}_n/r_{nn}, \quad c_k = c_k(z_{k+1}^n) = (\bar{y}_k - \sum_{j=k+1}^{n} r_{kj}z_j)/r_{kk}, \quad k = n-1, \ldots, 1.$$

Notice that c_k is fixed when z_{k+1}^n is fixed. The above equation can be rewritten as

$$\sum_{k=1}^{n} r_{kk}^2 (z_k - c_k)^2 < \beta.$$

This inequality is equivalent to the following n inequalities:

$$\begin{aligned}
\text{level } n: &\quad (z_n - c_n)^2 < \frac{1}{r_{nn}^2}\beta, \\
\text{level } n-1: &\quad (z_{n-1} - c_{n-1})^2 < \frac{1}{r_{n-1,n-1}^2}[\beta - r_{nn}^2(z_n - c_n)^2], \\
&\quad \vdots \\
\text{level } k: &\quad (z_k - c_k)^2 < \frac{1}{r_{kk}^2}[\beta - \sum_{i=k+1}^{n} r_{ii}^2 (z_i - c_i)^2], \\
&\quad \vdots \\
\text{level } 1: &\quad (z_1 - c_1)^2 < \frac{1}{r_{11}^2}[\beta - \sum_{i=2}^{n} r_{ii}^2 (z_i - c_i)^2].
\end{aligned}$$

During the search process, z_k is determined at level k, where $z_n, z_{n-1}, \ldots, z_{k+1}$ have already been determined, but $z_{k-1}, z_{k-2}, \ldots, z_1$ are still unassigned (note that c_k depends on $z_{k+1}, z_{k+2}, \ldots, z_n$). Therefore, c_k is only changed when we move to level k from level $k+1$ but remains unchanged when we move to level k from level $k-1$. When we move from level $k+1$ to level k, we first compute c_k and then choose $z_k = \lfloor c_k \rceil$, the nearest integer to c_k. When we move from level $k-1$ to level k, we choose z_k to be the next nearest integer to c_k. At each level k, z_k takes on values in the order

$$\lfloor c_k \rceil, \lfloor c_k - 1 \rceil, \lfloor c_k + 1 \rceil, \lfloor c_k - 2 \rceil, \ldots, \quad \text{if } c_k \leq \lfloor c_k \rceil,$$

or

$$\lfloor c_k \rceil, \lfloor c_k + 1 \rceil, \lfloor c_k - 1 \rceil, \lfloor c_k + 2 \rceil, \ldots, \quad \text{if } c_k > \lfloor c_k \rceil.$$

In the Schnorr & Euchner strategy, the initial bound β can be set to ∞. When we first find the integer point, the k-th entry of the first integer point is $\lfloor c_k \rceil$ for $k = n, \ldots, 1$. We can use this integer point to update β, reducing the hyper-ellipsoid in what is, from an OR perspective, a bounding operation.

Alternatively, we can obtain an initial bound by rounding the solution of the continuous least squares (LS) problem. Let $\boldsymbol{z}_{LS}$ be the continuous LS solution, i.e., $\boldsymbol{z}_{LS} = \boldsymbol{R}^{-1}\bar{\boldsymbol{y}}$. Then we round each entry of $\boldsymbol{z}_{LS}$ to its nearest integer to get the integer vector $\lfloor \boldsymbol{z}_{LS} \rceil$. As the problem is unconstrained, $\lfloor \boldsymbol{z}_{LS} \rceil$ is feasible and therefore the optimal solution must be as least as good. So we can set $\beta = \|\bar{\boldsymbol{y}} - \boldsymbol{R}\lfloor \boldsymbol{z}_{LS} \rceil\|_2^2$.

The search algorithm for the binary ILS problem is modified to take into account the variable bounds, i.e., $\boldsymbol{z} \in \{0, 1\}$. Chang & Han [5] proposed an efficient algorithm for ILS problems with bounds on the variables based on the same search framework, which is considered the state-of-the-art search algorithm. We use this algorithm in our implementation for the binary ILS problem.

3.2 B&C MIP solvers

BQP problems can be modeled as a MIP, a well-known approach in OR that has been widely used for combinatorial optimization problems since the 1950s. It is also one of the most common approaches for solving discrete optimization problems. MIP can be further categorized into mixed integer linear programming (MILP) and mixed integer nonlinear programming (MINLP).

BQP is a special type of the more general integer quadratic programming problem (IQP), both of which belong to the sub-class of MINLP that is generally solved with the following four approaches based on MIP-related techniques: branch-and-bound and branch-and-cut (B&C) [18], outer approximation [19], extended cutting-planes [20], and generalized Benders decomposition [21]. Of the above methods, the outer-approximation and the B&C methods have received the most attention and both commercial and non-commercial solvers are available based on these two approaches. An empirical study of these two methods on binary quadratic programming problems concluded that the two approaches performs significantly differently on various classes of problems and it is not clear which is generally superior [7]. Though note that this conclusion was made in 1997.

The state-of-the-art MIP solvers are typically hybrid algorithms that combine many of the above elements. Many solvers such as SCIP [15] and FilMINT [22] incorporate the outer-approximation into the B&C framework and are efficient for solving convex MINLP problems. Commercial software such as CPLEX, GUROBI and BARON also provide functionality to solve convex mixed IQP problems and so can all be applied to solve the BQP problems. We refer interested readers to Bussieck & Vigerske [12] for a recent review of MINLP solvers.

3.3 Previous results

Our preliminary results in solving general IQP problems with variable bounds demonstrated that the DEBS method performs especially well on problems where the residual $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{z}^*\|_2^2$ is small, where $\boldsymbol{z}^*$ is the optimal integer solution [16]. However, a MIP solver was more efficient than DEBS for problems with small variable domains (i.e., narrow variable bounds) and larger residual. These results provide motivation to combine the techniques and design a DEBS/B&C hybrid algorithm for the BQP problem: the BQP problem has the smallest possible variable domains and the residual is generally large. A detailed analysis of the residuals of the problem instances used here is given below in Section 6.

4 A DEBS/B&C hybrid algorithm

In this section we describe the three techniques that we integrated into the B&C-based MIP solver SCIP: preconditioning, axis-aligned circumscribed box constraints, and a primal heuristic using DEBS. First, preconditioning applies the same reduction techniques from the DEBS method to the BQP problem to transform it into a new BQP problem with the hope of reducing the size of the search tree explored by the B&C algorithm. Second, the axis-aligned circumscribed box represents globally valid bounds of the variables derived from the geometry of the objective function of the BQP problem. The box is computed and used to update the bounds whenever a new incumbent solution is found. Finally, we also apply DEBS as a primal heuristic during the search to help find high-quality solutions.

4.1 Preconditioning

We first describe the procedure of preconditioning for the BQP problem and then explain the logic behind it. Preconditioning applies the reduction techniques from the DEBS method to the BQP problem to transform the original BQP problem into a new BQP problem, defined as follows:

$$\min_{\boldsymbol{x}\in\{0,1\}} \frac{1}{2}\boldsymbol{x}^T\bar{\boldsymbol{H}}\boldsymbol{x} + \bar{\boldsymbol{f}}^T\boldsymbol{x}, \tag{6}$$

where $\bar{\boldsymbol{H}} = \boldsymbol{R}^T\boldsymbol{R}$ and $\bar{\boldsymbol{f}} = -\bar{\boldsymbol{y}}^T\boldsymbol{R}$. Note that $\boldsymbol{R}$ and $\bar{\boldsymbol{y}}$ are from the reduced binary ILS problem (5).

The reason for performing preconditioning can be explained with the idea of *distance to integrality*. Let $\boldsymbol{x}^{\mathrm{R}}$ denote the optimal solution to the continuous relaxation of the ILS problem and $\boldsymbol{x}^{\mathrm{I}}$ the solution to the ILS problem. We define the distance to integrality as

$$d(\boldsymbol{x}^{\mathrm{R}}, \boldsymbol{x}^{\mathrm{I}}) = \left\|\boldsymbol{x}^{\mathrm{R}} - \boldsymbol{x}^{\mathrm{I}}\right\|_2 .$$

Thus, the distance to integrality is the Euclidean distance between the real and integer optimal solutions to the ILS problem. It has been shown that the above distance is a useful indicator of the size of the search tree explored in a B&C based MIP solvers [23]. The intuition is that if the root node $\boldsymbol{x}^{\mathrm{R}}$ is closer to the leaf node $\boldsymbol{x}^{\mathrm{I}}$, fewer branching decisions are required and hence the search tree will be smaller. Previous results on communication applications showed that the experimental correlation between the number of nodes and the distance to integrality in logarithmic scales is around 0.7 [23]. Preconditioning aims at decreasing this distance.

In the following we describe the relationship between preconditioning and the distance between the real and integer optimal solutions to the ILS problem.

Using $\boldsymbol{A}^\dagger$, the Moore-Penrose generalized inverse of $\boldsymbol{A}$, we can write $\boldsymbol{x}^{\mathrm{R}} = \boldsymbol{A}^\dagger\boldsymbol{y}$ and $\boldsymbol{A}^\dagger\boldsymbol{A} = \boldsymbol{I}$. Therefore

$$d(\boldsymbol{x}^{\mathrm{R}}, \boldsymbol{x}^{\mathrm{I}}) = \|\boldsymbol{x}^{\mathrm{R}} - \boldsymbol{x}^{\mathrm{I}}\|_2 = \|\boldsymbol{A}^\dagger\boldsymbol{y} - \boldsymbol{x}^{\mathrm{I}}\|_2 = \|\boldsymbol{A}^\dagger(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^{\mathrm{I}})\|_2,$$

leading to

$$d(\boldsymbol{x}^{\mathrm{R}}, \boldsymbol{x}^{\mathrm{I}}) \leq \|\boldsymbol{A}^{\dagger}\|_2 \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^{\mathrm{I}}\|_2. \tag{7}$$

Therefore, it is possible to decrease the distance to integrality by transforming the problem using a unimodular matrix $\boldsymbol{U}$ that minimizes $\|(\boldsymbol{A}\boldsymbol{U})^{\dagger}\|_2$. We refer the process of using the unimodular transformation to reduce the conditioning factor to as *preconditioning*.

Preconditioning is effective for the general unconstrained IQP problems in communication applications [23]. In this paper we apply the preconditioning technique to the BQP problem. In our implementation, we perform the preconditioning in the presolving stage of the B&C algorithm to the BQP problem (3) and transform the BQP problem to its reduced formulation (5), then we perform the B&C search on the reduced problem.

4.2 Axis-aligned Circumscribed Box Constraints

Based on the geometry of the objective function of the BQP problem, the axis-aligned circumscribed box can be computed and used to fix the values of the (binary) variables. Chang & Golub [6] proposed an efficient way to compute the smallest hyper-rectangle whose edges are parallel to the axes of the coordinate system and that includes the hyper-ellipsoid defined by Equation (4). Let $\mathcal{B}$ be the smallest hyper-rectangle including the reduced hyper-ellipsoid $\|\bar{\boldsymbol{y}} - \boldsymbol{R}\boldsymbol{z}\|_2^2 \leq \beta$, the lower bound $\boldsymbol{l}$ and the upper bound $\boldsymbol{u}$ can be computed as follows:

$$u_k = \left\lfloor \sqrt{\beta} \left\| \boldsymbol{R}^{-T} \boldsymbol{e}_k \right\|_2 + \boldsymbol{e}_k^T \boldsymbol{R}^{-1} \bar{\boldsymbol{y}} \right\rfloor,$$

$$l_k = \left\lceil -\sqrt{\beta} \left\| \boldsymbol{R}^{-T} \boldsymbol{e}_k \right\|_2 + \boldsymbol{e}_k^T \boldsymbol{R}^{-1} \bar{\boldsymbol{y}} \right\rceil.$$

In our implementation, the axis-aligned circumscribed box is computed every time a new incumbent is found. Since this box is globally feasible, the global bounds of the variables are updated if the box is tighter than the current global bounds of the variables. As our decision variables are binary, any bound tightening corresponds to fixing a value.

4.3 DEBS as a Primal Heuristic

Primal heuristics are important components in state-of-the-art MIP solvers [15] used to find feasible solutions. Good quality feasible solutions are beneficial in a number of ways [8]. First, the feasible solution proves that the problem is feasible, and the solution might already be good enough. Second, the feasible solution can be used to derive valid bound for pruning the search tree. Third, the feasible solution can be used for dual fixing or reduction to strengthen the problem formulation.

When applying the DEBS as a primal heuristic, we first employ a resource bound (e.g., a short time limit) for running the DEBS method at the root node after MIP presolving. If the DEBS method finds an optimal solution, SCIP returns it and the algorithm terminates. If the optimal solution is not found within the time limit, the best incumbent found by the DEBS method is given to SCIP as the starting integer solution. In addition, the DEBS method is executed during the search process. In our implementation, we follow SCIP's primal heuristics parameters: frequency and offset. The offset is the depth where the primal heuristic is executed first and frequency + offset defines the subsequent depths. We chose the values of 10 for frequency and 0 for offset meaning that the DEBS method is only used as a primal heuristic at the root and at nodes of depth 10. If the solution found by the DEBS is better than the current incumbent, the current incumbent is updated, and we re-compute the axis-aligned circumscribed box (Section 4.2) to further tighten the bounds of the variables.

During the B&C process, variables are fixed either by branching decisions or other techniques. When such fixing takes place, we reduce the size of the binary ILS problem. Let $\boldsymbol{U}$ be the set of indices of variables that are not yet fixed at a given node in the search tree and let $\boldsymbol{A}_i$ be the i-th column vector of $\boldsymbol{A}$ of the original binary ILS problem in Equation (3). The updated binary ILS can be defined as follows:

$$\min_{\tilde{\boldsymbol{x}}\in\{0,1\}} \left\|\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}\right\|_2^2,$$

where

$$\tilde{\boldsymbol{A}} = [\boldsymbol{A}_i], \quad i \in \boldsymbol{U},$$
$$\tilde{\boldsymbol{y}} = \boldsymbol{y} - \sum_{i\in\boldsymbol{U}} \boldsymbol{A}_i x_i, \quad i \in \boldsymbol{U},$$

$\tilde{\boldsymbol{A}} \in \mathbb{R}^{n\times|\boldsymbol{U}|}$ and $\tilde{\boldsymbol{y}} \in \mathbb{R}^n$. The resulting problem is $(n-|U|)$ dimensional smaller than the original problem.

5 Computational results

In this section we compare the performance of the B&C-based MIP solvers CPLEX and SCIP, the DEBS method, and the new hybrid algorithm implemented in SCIP. We also provide the best current results from Li et al. [11] and Krislock et al. [3]. For the problems we did not find results in the SDP literature (Carter and William type problems), we use the online SDP solver BiqCrunch [24] for comparison. We present results for CPLEX and SCIP in order to compare DEBS and our hybrid both to a state-of-the-art commercial MIP solver and to isolate the impact that our DEBS-based hybridizations have on default SCIP performance.

5.1 Experimental setup

All experiments were performed on a Intel Core i7 3.40 GHz machine (in 64 bit mode) with 8GB memory running Red Hat Enterprise 6.2 with one thread. We perform experiments with CPLEX Optimization Studio v12.5 and SCIP v3.0.2. We use MATLAB 7.7.0 for generating the problem instances. The DEBS approach is written in C and the new hybrid algorithm is implemented in SCIP. The CPU time limit for each run on each problem instance is 3600 seconds.

We use a subset of the benchmark instances presented by Li et al. [11], excluding the max-cut problems as they require additional transformations and cannot be solved with MIP or DEBS directly. We perform experiments on medium size BQP problems on seven problem sets: Carter type problems [25], William type problems [26], bqp50 and bqp100 problems, and gkaia, gkaib, and gkaid problems [10]. We generate ten problem instances for each of the Carter and William type problems. In order to ensure convexity, we perform the same transformation as Li et al. We compute the smallest eigenvalue for the $\boldsymbol{H}$ matrix of each problem and let it be λ_{min}. Then we apply the perturbation vector $\boldsymbol{u} = (-\lambda_{min} + 0.001)\boldsymbol{e}$ to the BQP problem (2) when λ_{min} is negative. Note that the transformed problem has the same optimal solution as the original problem. For the DEBS method, we use Cholesky decomposition on matrix $\boldsymbol{H}$ in the BQP problem to obtain matrix $\boldsymbol{A}$ in the binary ILS problem, and we obtain $\boldsymbol{y}$ from the equation $\boldsymbol{f} = -\boldsymbol{y}^T\boldsymbol{A}$, which gives us $\boldsymbol{y} = -(\boldsymbol{f}\boldsymbol{A}^{-1})^T$.

5.2 Results

The results for all seven problem sets are presented in Table 1. For the bqp50, bqp100, gkaia, gkaib, and gkaid problems, we report the CPU time for each instance. For the Carter and Williams type problems, we report the arithmetic mean CPU time "arith", and the shifted geometric mean CPU time "geo" on the ten instances for each problem size.[2]

The comparison of the identical MIP models in CPLEX and SCIP shows, somewhat surprisingly, that CPLEX performs substantially better than SCIP on only two of the seven problem sets (Williams and Carter). SCIP is clearly superior for bqp50, bqp100, gkaia, and gkaib sets while solving one problem more for gkaid. We attribute this strong performance of SCIP to the quadratic constraint handler described in Berthold et al. [8].

DEBS is noticeably more efficient than B&C (both CPLEX and SCIP) on the William and Carter type problems and superior to CPLEX on the bqp50, gkaia, and gkaib problems. While CPLEX does not out-perform DEBS for any problem instances, SCIP has the edge over DEBS on bqp50, bqp100, and gkaia. For the final problem set, gkaid, SCIP is able to solve one more problem than DEBS. Comparing the DEBS method and CPLEX, the DEBS method is noticeably more efficient than CPLEX for five problem sets: William type, Carter type, bqp50, gkaia and gkaib.

With respect to the first contribution of this paper, therefore, we conclude that the DEBS approach from the communications literature is a competitive

[2] The shifted geometric mean time is computed as follows: $\prod(t_i + s)^{1/n} - s$, where t_i is the actual CPU time, n is the number of instances, and s is chosen as 10. Using geometric mean can decrease the influence of the outliers of data [15].

Table 1: A comparison of four approaches plus Li et al.'s results and Krislock et al.'s SDP results for the seven problem sets. Bold numbers indicate the best approach for a given problem. The symbol '-' means that no problem instances were solved to optimality within 3600 seconds. For the Carter and William type problems, n is the size of the problem, and p and d are problem generation parameters. The superscripts indicate the number of instances for which no optimal solution was found.

Carter type problems

n	p	CPLEX		SCIP		DEBS		Hybrid		Li	SDP
		arith	geo	arith	geo	arith	geo	arith	geo	arith	arith
40	0.2	0.21	0.21	29.55	21.06	**0**	0	0.02	0.02	0.32	0.32
40	0.3	0.38	0.38	34.01	26.35	**0.01**	0.01	0.03	0.03	0.26	0.62
50	0.2	2.38	2.26	896.53	655.17	**0.15**	0.15	0.24	0.24	1.5	0.66
50	0.3	2.85	2.69	891.71	767.32	**0.80**	0.80	1.10	0.85	1.2	1.15
80	0.4	1170.01^{2}	349.34^{2}	-	-	1020.20^{2}	249.31^{2}	98.37	67.40	89.9	**21.39**
80	0.5	659.99	369.26	-	-	225.33	60.03	48.09	31.02	**2.5**	16.01
100	0.5	3222.90^{8}	2955.28^{8}	-	-	2766.31^{7}	2226.48^{7}	2249.53^{5}	1584.12^{5}	**10.8**	57.97

William type problems

n	d	CPLEX		SCIP		DEBS		Hybrid		Li	SDP
		arith	geo	arith	geo	arith	geo	arith	geo	arith	arith
40	0.5	0.25	0.25	8.29	7.59	**0.01**	0.01	**0.01**	0.01	0.44	0.33
40	0.7	0.22	0.21	14.16	12.16	**0.01**	0.01	**0.01**	0.01	0.47	0.34
50	0.2	1.75	1.70	12.00	10.51	**0.35**	0.32	0.44	0.40	6.6	0.52
50	0.4	1.74	1.68	148.64	115.64	**0.09**	0.09	0.14	0.14	24	0.75
50	1	2.78	2.61	1463.02^{2}	948.47^{2}	**0.13**	0.09	0.19	0.18	2.5	0.85
60	0.1	21.83	15.29	4.61	4.17	1.57	1.51	1.93	1.85	**0.72**	**0.72**
60	0.2	22.81	11.20	387.15	163.29	2.10	1.84	2.65	2.26	5.3	**1.33**
60	0.4	14.18	11.49	2568.63^{4}	1833.75^{4}	31.21	10.48	6.24	5.17	18.9	**2.32**
80	0.1	2014.32^{5}	1047.60^{5}	1688.28^{3}	850.38^{3}	157.52	80.99	178.82	89.52	7	**3.01**
80	0.2	1023.81^{1}	530.40^{1}	-	-	188.92	97.89	140.83	82.79	123.1	**2.19**

bqp50 problems

Instance	CPLEX	SCIP	DEBS	Hybrid	Li	SDP
bqp50-1	11.66	0.07	0.65	**0.06**	<0.2	0.25
bqp50-2	1.84	0.06	1.2	**0.03**	<0.2	0.24
bqp50-3	0.60	0.04	**0**	0.03	<0.2	0.17
bqp50-4	2.84	0.05	0.30	**0.04**	<0.2	0.22
bqp50-5	3.72	0.03	0.13	**0.02**	<0.2	0.21
bqp50-6	0.46	**0.01**	0.15	0.02	<0.2	0.21
bqp50-7	1.70	0.05	**0.02**	0.05	<0.2	0.26
bqp50-8	0.93	0.06	**0.03**	0.06	<0.2	0.19
bqp50-9	5.32	0.09	0.35	**0.07**	<0.2	0.25
bqp50-10	14.30	0.07	0.25	**0.06**	<0.2	0.24

bqp100 problems

Instance	CPLEX	SCIP	DEBS	Hybrid	Li	SDP
bqp100-1	-	55.50	-	46.13	1648	**1.72**
bqp100-2	-	6.74	-	7.12	63.4	**2.06**
bqp100-3	-	4.99	-	9.10	66.3	**1.6**
bqp100-4	-	8.83	-	8.96	165.7	**1.42**
bqp100-5	-	10.28	-	9.17	1230	**1.69**
bqp100-6	-	566.92	-	55.54	175.2	**11.12**
bqp100-7	-	42.52	-	13.37	427	**1.8**
bqp100-8	-	4.63	-	5.96	25.6	**1.63**
bqp100-9	-	3.50	1282.42	5.90	21.9	**1.46**
bqp100-10	-	4.43	-	4.45	57.2	**1.67**

gkaia problems

Instance	n	CPLEX	SCIP	DEBS	Hybrid	Li	SDP
gka1a	50	11.32	0.03	4.55	**0.02**	<1	0.33
gka2a	60	1.60	**0.02**	0.84	**0.02**	<1	0.42
gka3a	70	2817.3	0.93	1335.20	**0.79**	<1	0.93
gka4a	80	88.86	1.12	348.61	**0.86**	<1	1.07
gka5a	50	3.22	0.94	**0.11**	**0.11**	<1	0.29
gka6a	30	0.11	0.37	**0**	0.02	<1	0.06
gka7a	30	0.05	0.41	**0**	0.02	<1	0.07
gka8a	100	-	**0.08**	634.88	**0.08**	<1	2.11

gkaib problems

Instance	n	CPLEX	DEBS	SCIP	Hybrid	Li	SDP
gka1b	20	0.08	**0**	0.29	0.01	<1	0.1
gka2b	30	0.1	**0**	0.36	0.01	<1	2.18
gka3b	40	0.58	**0**	0.6	0.01	<1	9.44
gka4b	50	1.7	**0.01**	1.15	0.03	<1	18.42
gka5b	60	5.86	**0.04**	1.44	0.07	<1	39.2
gka6b	70	21.75	**0.15**	1.95	0.26	<1	86.56
gka7b	80	59.88	**0.34**	2.52	0.53	<1	212.73
gka8b	90	160.79	**0.98**	3.67	1.43	<1	789.1
gka9b	100	502.79	**2.66**	6.04	3.67	<1	1608.04
gka10b	125	-	20.7	9.51	12.06	<**1**	5104.21

gkaid problems

Instance	n	CPLEX	DEBS	SCIP	Hybrid	Li	SDP
gka1d	100	-	-	5.90	5.84	24	**2.1**
gka2d	100	-	-	-	-	5671	**2.65**
gka3d	100	-	-	-	-	1713	**2.79**
gka4d	100	-	-	-	-	3835	**4.23**
gka5d	100	-	-	-	-	5466	**55.72**
gka6d	100	-	-	-	-	1534	**3.03**
gka7d	100	-	-	-	-	4273	**46.04**
gka8d	100	-	-	-	-	683	**2.94**
gka9d	100	-	-	-	-	2481	**34.08**
gka10d	100	-	-	-	-	1878	**30.97**

approach to solving BQPs that are of interest to the OR community. In fact, DEBS is superior to a state-of-the-art commercial solver, CPLEX.

Turning to the hybridization of B&C and DEBS, we see that the new hybrid approach out-performs CPLEX on all problem sets while achieving clearly better performance than SCIP on the Williams and Carter type problems and marginally better performance on the other five problem sets: there are a number of problem instances in the latter five sets where the hybrid makes meaningful improvements on the default SCIP performance and none where it is substantially inferior. Similarly, the hybrid improves on the pure DEBS results on all problem sets though the improvement is marginal for bqp50, gkaib and gkaid.

While our experimental environment is different from that of Li et al., the new hybrid algorithm appears competitive on the Carter type, William type, bqp50, gkaia, and gkaib problems. The hybrid algorithm performs better on the bqp100 problems and worse on the gkaid problems. For the bqp100 problems, the hybrid algorithm outperforms Li et al.'s algorithm by about an order of magnitude.

Compared to the SDP approach, the hybrid algorithm performs basically the same or slightly better for moderate size problems, but significantly less efficiently for larger problems. However for some dense problem instances (gkaib), the hybrid algorithm greatly outperforms the SDP approach. The hybrid algorithm was able to solve all of them quickly, while the SDP approach is in general two orders of magnitudes slower. Overall, the SDP approach still appears to be most efficient for the BQP problems.

6 Discussion

Our preliminary work showed that the DEBS method performs especially well on ILS problems with bounds on the variables when the residual, $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{z}^*\|_2^2$, is small ($\boldsymbol{z}^*$ is the optimal integer solution). However, a MIP solver performs much better than DEBS with small variable domains and large residual [16]. To explain the role of the residual, we define the *noise*[3] in the data with the following linear model: $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{v}$, where $\boldsymbol{v} \in \mathbb{R}^n$ denotes the noise vector. A large noise vector results a large residual. The residual directly determines the search space of the DEBS method, since the minimum ellipsoid is defined by the optimal integer solution $\boldsymbol{z}^*$ and its associated residual (Eqn. (4)). In the extreme case, when $\boldsymbol{v} = \boldsymbol{0}$, there is no noise. When such a problem (3) is solved to optimality, $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{z}^*\|_2^2$ is equal to zero and so the ellipsoid reduces to a single point. As $\boldsymbol{v}$ increases, the DEBS method needs to search a larger space.

In communication applications, a noise vector $\boldsymbol{v}$ of $\sigma * \texttt{randn(m,1)}$, $\sigma \geq 5$ is generally considered as a very large noise in the data [23]. Note that `randn(m,1)` is a MATLAB function which returns an m-dimensional vector containing pseudorandom values drawn from the standard normal distribution $N(0, 1)$. We computed the noise vectors in the seven problem sets by substituting the optimal

[3] Noise has a recognized physical meaning in the communications literature [5].

solution or the best solution found within the time limit into the linear model and found that $\sigma > 5$ for all seven problem sets. Therefore these problems can be regarded as problems with very large noise and the poor performance of the DEBS method on the larger problems is consistent with both our previous results and the need to search a larger ellipsoid.

However, it is surprising that the DEBS method performs so well compared to the MIP solvers, especially CPLEX, given the large noise and tightest possible variable bounds in the problem instances. This contradicts our previous results on general IQPs that CPLEX performs better than the DEBS method for problems with large noise and small variable domains [16]. One interesting direction for future work is to further investigate the reason that the DEBS method performs well on the BQP problem and extract more information to enhance the hybrid approach.

Our results show that combining the DEBS method with a B&C based MIP solver indeed results in better performance when large noise is present in the problem instances. The additional techniques from MIP such as relaxation, cutting planes, and inference appear to complement the presolving, axis-aligned box constraints, and primal heuristic adopted from the DEBS approach.

Analysis of CPLEX solving behaviour shows that the reason that CPLEX is unable to prove optimality for bigger problems (e.g., bqp100 and gkaid) is mainly weakness in the dual bound. Solutions with good quality are found early in the search but the dual bound only improves slowly. For the DEBS method, optimal or near optimal solutions are typically found, though not as quickly as with CPLEX. However, similarly to CPLEX, DEBS is not able to prove optimality since the search space is too large and it does not have an effective dual bounding mechanism. SCIP, however, performs well on the gka1d, bqp50, and bqp100 problems because it is able to make rapid improvements in the dual bound. The lack of such an ability, conversely, appears to explain the very poor performance of SCIP on the Carter and Williams problems. The performance of the hybrid suggests that the poor performance of SCIP on Carter and Williams may be due to not being able to find a high-quality feasible solution. One area for future work is to investigate the performance differences between problem sets, in particular, to understand the structure that SCIP appears to be taking advantage of in the problems in which it performs well. The comparison of the hybrid algorithm and the SDP approach shows that the hybrid algorithm seems to outperform the SDP approach on dense problems (e.g., gkaib). Further investigation is required to identify the reasons.

7 Conclusions

In this paper we conducted the first empirical study comparing the performance of a B&C based MIP solver and the DEBS method for the BQP problem. We also proposed a new hybrid algorithm that combines techniques from MIP solving and DEBS, implementing DEBS-based presolving, axis-aligned box constraints, and a primal heuristic in a B&C MIP solver. Though we only examine binary quadratic problems here, the resulting hybrid algorithm can be used to solve problems with general integer bounds on the variables.

We compared the performance of the B&C based MIP solvers CPLEX and SCIP, the DEBS method, and the new hybrid algorithm for the BQP problems on seven sets of benchmark instances. Results show that the DEBS performs much better than CPLEX and about even with SCIP, though different problem sets show markedly different relative performance. The DEBS algorithm from the communications literature, therefore, is at least competitive with state-of-the-art B&C MIP approaches to binary quadratic problems.

The hybridization of DEBS and B&C out-performs both the B&C and DEBS based approaches across all problem sets. The improvement is substantial for some problem sets, though only marginal in others. Compared to the best special-purpose algorithms, Li et al.'s branch-and-bound algorithm [11] and an SDP approach [3], the new hybrid algorithm is competitive though overall not as strong as the SDP algorithm. We, therefore, conclude that the hybridization of DEBS with B&C is a strong contender on BQP problems, though the SDP approach remains the state of the art.

In the future, we are interested in exploring other possibilities of using geometric information to enhance MIP solving. We expect to extend the hybrid algorithm to solve generic MIP problems (i.e., problems with linear constraints) while preserving the ability to solve the BQP problems efficiently. We are investigating applications in communications where such constrained quadratic problems may be a natural model.

References

1. Lewis, M., Alidaee, B., Kochenberger, G.: Using xqx to model and solve the uncapacitated task allocation problem. Operations research letters **33**(2) (2005) 176–182
2. Finke, G., Burkard, R., Rendl, F.: Quadratic assignment problems. Surveys in combinatorial optimization (2011) 61
3. Krislock, N., Malick, J., Roupin, F.: Improved semidefinite bounding procedure for solving max-cut problems to optimality. Mathematical Programming (2012) 1–26
4. Watters, L.J.: Reduction of integer polynomial programming problems to zero-one linear programming problems. Operations Research **15**(6) (1967) 1171–1174
5. Chang, X.W., Han, Q.: Solving box-constrained integer least squares problems. Wireless Communications, IEEE Transactions on **7**(1) (2008) 277–287
6. Chang, X.W., Golub, G.H.: Solving ellipsoid-constrained integer least squares problems. SIAM Journal on Matrix Analysis and Applications **31**(3) (2009) 1071–1089
7. Borchers, B., Mitchell, J.E.: A computational comparison of branch and bound and outer approximation algorithms for 0–1 mixed integer nonlinear programs. Computers & Operations Research **24**(8) (1997) 699–701
8. Berthold, T., Heinz, S., Vigerske, S.: Extending a CIP framework to solve MIQCPs. In: Mixed-Integer Nonlinear Programming. Volume 154 of The IMA Volumes in Mathematics and its Applications. Springer (2012) 427–445

9. Rendl, F., Rinaldi, G., Wiegele, A.: Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. Mathematical Programming **121**(2) (2010) 307–335
10. Wiegele, A.: Biq mac library–a collection of max-cut and quadratic 0–1 programming instances of medium size. Preprint (2007)
11. Li, D., Sun, X., Liu, C.: An exact solution method for unconstrained quadratic 0–1 programming: a geometric approach. Journal of Global Optimization **52**(4) (2012) 797–829
12. Bussieck, M.R., Vigerske, S.: MINLP solver software. Wiley Encyclopedia of Operations Research and Management Science, Wiley, Chichester (2010)
13. Billionnet, A., Elloumi, S.: Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem. Mathematical Programming **109**(1) (2007) 55–68
14. Teunissen, P.J., Kleusberg, A., Teunissen, P.: GPS for Geodesy. Volume 2. Springer Berlin (1998)
15. Achterberg, T.: Constraint Integer Programming. PhD thesis, Technische Universität Berlin (2007)
16. Ku, W.Y., Beck, J.C.: Combining discrete ellipsoid-based search and branch-and-cut for integer least squares problems. Technical Report MIE-OR-TR2013-07, University of Toronto, Toronto (2013)
17. Schnorr, C.P., Euchner, M.: Lattice basis reduction: Improved practical algorithms and solving subset sum problems. Mathematical programming **66**(1) (1994) 181–199
18. Gupta, O.K., Ravindran, A.: Branch and bound experiments in convex nonlinear integer programming. Management Science **31**(12) (1985) 1533–1546
19. Duran, M.A., Grossmann, I.E.: An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Mathematical programming **36**(3) (1986) 307–339
20. Westerlund, T., Pettersson, F.: An extended cutting plane method for solving convex minlp problems. Computers & Chemical Engineering **19** (1995) 131–136
21. Geoffrion, A.M.: Generalized benders decomposition. Journal of optimization theory and applications **10**(4) (1972) 237–260
22. Abhishek, K., Leyffer, S., Linderoth, J.: Filmint: An outer approximation-based solver for convex mixed-integer nonlinear programs. INFORMS Journal on computing **22**(4) (2010) 555–567
23. Ku, W.Y., Anjos, M.F., Chang, X.W.: Lattice preconditioning for the real relaxation branch-and-bound approach for integer least squares problems. Technical report, Group for Research in Decision Analysis, Montreal (2013)
24. Krislock, N., Malick, J., Roupin, F.: BiqCrunch online solver (2012). http://lipn.univ-paris13.fr/BiqCrunch/solver (Retrieved: 12/22/2013)
25. Carter, M.W.: The indefinite zero-one quadratic problem. Discrete Applied Mathematics **7**(1) (1984) 23–44
26. Williams, A.: Quadratic 0-1 Programming Using the Roof Dual: With Computational Results. RUTCOR, Hill Center, Rutgers University (1985)