Combining Constraint Propagation and Discrete Ellipsoid-Based Search to Solve the Exact Quadratic Knapsack Problem

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 an extension to the discrete ellipsoid-based search (DEBS) to solve the exact quadratic knapsack problem (EQKP), an important class of optimization problem with a number of practical applications. For the first time, our extension enables DEBS to solve convex quadratically constrained problems with linear constraints. We show that adding linear constraint propagation to DEBS results in an algorithm that is able to outperform both the state-of-the-art MIP solver CPLEX and a semi-definite programming approach by about one order of magnitude on two variations of the EQKP.

1 Introduction

The exact quadratic knapsack problem (EQKP) [1] consists of selecting a subset of elements such that the sum of the distances between the chosen elements is maximized while also satisfying a knapsack constraint. The EQKP is an extension of the well studied maximum diversity problem [2], the quadratic knapsack problem [3], and the exact linear knapsack problem [4], which arise in a wide range of real world applications such as wind farm optimization [5,6] and task allocation [7]. The EQKP consists of one quadratic objective function and two linear constraints, i.e., one knapsack constraint and one cardinality constraint, where all variables are binary. The EQKP was first studied by Létocart [1] who proposed an efficient heuristic method for the problem.

For solving EQKPs exactly, the common generic approach in Operations Research is the use of a commercial MIP solver such as CPLEX or Gurobi. These solvers have been extended to reason about quadratic constraints [8] and they are able to outperform several other exact approaches [9]. The other generic approach is the semi-definite programming (SDP) based branch-and-bound algorithm [10], which is often regarded as the state-of-the-art approach for solving binary quadratic programming problems (BQP). EQKP is an extension of the BQP and it can be solved with the SDP approach. However, the SDP approach has not been evaluated on problems with the EQKP structure.

In this paper, we extend discrete ellipsoid-based search (DEBS) method to solve the EQKP. DEBS is a specialized search used in the communications literature (e.g., see [11]) to solve integer least squares problems (ILS) based on

the clever enumeration of integer points within the hyper-ellipsoid defining the feasible space. We have previously shown that DEBS can be applied to the BQP efficiently [12]. As EQKPs can be reformulated as ILS problems, we perform this transformation and extend DEBS for solving convex quadratically constrained problems with linear constraints. To our knowledge, this is the first time that DEBS has been applied to problems with linear constraints so our extension enables DEBS to solve a much broader class of problems, i.e., problems with a quadratic objective function and any number of linear constraints. As an initial test-bed, we use the EQKP to show that our extension achieves state-of-the-art for solving the EQKP and a variation, outperforming both CPLEX and the SDP based approach. Interestingly, the three exact approaches agree on problem difficulty: instances that take more time for the DEBS method to solve are also more challenging for the other two approaches.

2 The Exact Quadratic Knapsack Problem

The EQKP problem is defined as follows:

$$\begin{aligned} \max_{\boldsymbol{x}\in\{0,1\}} \quad & \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} h_{ij}x_i x_j, \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i = K, \\ & \sum_{i=1}^{n} a_i x_i \leq B, \end{aligned} \tag{1}$$

where $n \in \mathbb{Z}$, $K \in \mathbb{Z}$, $a_i \in \mathbb{R}^+, \forall i$, $B \in \mathbb{R}^+$, $h_{ij} \in \mathbb{R}, \forall i, j$.

For the simplicity of explanation, we reformulate the EQKP in its minimization form with matrix representation as follows:

$$\min_{\boldsymbol{x}\in\{0,1\}} \frac{1}{2}\boldsymbol{x}^T\boldsymbol{H}\boldsymbol{x} + \boldsymbol{f}^T\boldsymbol{x}, \quad \text{s.t. } \boldsymbol{c}_1^T\boldsymbol{x} = K,\ \boldsymbol{c}_2^T\boldsymbol{x} \leq B, \tag{2}$$

where $\boldsymbol{H} \in \mathbb{R}^{n\times n}$ is a symmetric semi-definite positive matrix, $\boldsymbol{f} \in \mathbb{R}^n$ is a vector equal to zeros, $\boldsymbol{c}_1 \in \mathbb{R}^n$ is a vector equal to ones, and $\boldsymbol{c}_2 \in \mathbb{R}^n$ is a vector equal to a_is. Note that $\boldsymbol{H}$ can always be made symmetric semi-definite positive to ensure convexity when all variables are binary [9].

To solve the EQKP with the DEBS method, we need to transform the objective function into its equivalent integer least squares (ILS) form through the relationship $\boldsymbol{H} = \boldsymbol{A}^T\boldsymbol{A}$ and $\boldsymbol{f} = -\boldsymbol{y}^T\boldsymbol{A}$. Thus, the equivalent ILS problem can be defined as follows:

$$\min_{\boldsymbol{x}\in\{0,1\}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2, \quad \text{s.t. } \boldsymbol{c}_1^T\boldsymbol{x} = K,\ \boldsymbol{c}_2^T\boldsymbol{x} \leq B. \tag{3}$$

3 The DEBS Method

The DEBS method has been developed to address three types of the ILS problems: unconstrained, box-constrained, and ellipsoid-constrained [13,14,15]. To the best of our knowledge, the DEBS method has not been extended to solve ILS problems with general linear constraints.

The DEBS method consists of two phases: reduction and search. The reduction is a preprocessing step that transforms $\boldsymbol{A}$ to an upper triangular matrix $\boldsymbol{R}$ with properties such that the search is more efficient [13]:

$$\min_{\boldsymbol{x}\in\mathbb{Z}^n} \|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2 \rightarrow \min_{\boldsymbol{z}\in\mathbb{Z}^n} \|\bar{\boldsymbol{y}}-\boldsymbol{R}\boldsymbol{z}\|_2^2 . \tag{4}$$

Geometrically, the optimal solution is found by searching discretely inside the ellipsoid defined by the objective function of the reduced ILS problem (right hand side of Equation (4)). Suppose the optimal solution $\boldsymbol{z}^*$ to the ILS problem satisfies the bound $\|\bar{\boldsymbol{y}}-\boldsymbol{R}\boldsymbol{z}^*\|_2^2 < \beta$, where β is a constant. This expression defines a hyper-ellipsoid with center $\boldsymbol{R}^{-1}\bar{\boldsymbol{y}}$.

Reduction. The reduction phase of DEBS transforms $\boldsymbol{A}$ into an upper triangular matrix $\boldsymbol{R}$ such that the diagonal entries are ordered in non-decreasing order:

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

It has been shown that the above order can greatly affect the efficiency of the DEBS search by reducing the branching factor at the top of the search tree [13].

Transforming $\boldsymbol{A}$ to $\boldsymbol{R}$ can be achieved by finding an orthogonal matrix $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ and a permutation matrix $\boldsymbol{P} \in \mathbb{Z}^{n\times n}$ such that $\boldsymbol{Q}^T\boldsymbol{A}\boldsymbol{P} = \begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{0}\end{bmatrix}$, $\boldsymbol{Q} = \begin{bmatrix}\boldsymbol{Q}_1, \boldsymbol{Q}_2\end{bmatrix}$. Therefore we have:

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

Let $\bar{\boldsymbol{y}} = \boldsymbol{Q}_1^T\boldsymbol{y}$, $\boldsymbol{z} = \boldsymbol{P}^T\boldsymbol{x}$, and $\bar{\boldsymbol{c}}_2 = \boldsymbol{c}_2\boldsymbol{P}$, the original EQKP (3) is then transformed to the new reduced EQKP:

$$\min_{\boldsymbol{z}\in\{0,1\}} \|\bar{\boldsymbol{y}}-\boldsymbol{R}\boldsymbol{z}\|_2^2, \quad \text{s.t. } \boldsymbol{c}_1^T\boldsymbol{x} = K,\ \bar{\boldsymbol{c}}_2^T\boldsymbol{x} \leq B. \tag{5}$$

Note that applying the permutation matrix $\boldsymbol{P}$ to the original problem changes the order of the variable bounds and the coefficients of the linear constraints. However, since the original lower and upper bounds on the variables are all zeros and ones, and the coefficients of the cardinality constraint are all ones for the EQKP, the new bounds and $\boldsymbol{c}_1$ remain unchanged after applying the permutation matrix $\boldsymbol{P}$. For problems with general variable bounds, i.e., $\boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}$, the reordering can be done with $\bar{\boldsymbol{l}} = \boldsymbol{P}^T\boldsymbol{l}$ and $\bar{\boldsymbol{u}} = \boldsymbol{P}^T\boldsymbol{u}$.

After the optimal solution $\boldsymbol{z}^*$ to the new EQKP problem (5) is found, the optimal solution, $\boldsymbol{x}^*$, to the original EQKP problem (3) can be recovered with the relationship $\boldsymbol{x}^* = \boldsymbol{P}\boldsymbol{z}^*$.

Search. The search is performed on the reduced problem (5). Among several search strategies in the literature, the Schnorr & Euchner strategy is usually considered the most efficient [16]. The search systematically enumerates all the integer points in the bounded hyper-ellipsoid with specific variable and value ordering heuristics. In addition, the hyper-ellipsoid is contracted during the enumeration process to further constrain the search space. The main ideas of the search are as follows.

Suppose the optimal solution $\boldsymbol{z}^*$ satisfies the 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). 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:

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

Note that c_k is a function of z_{k+1} to z_n, and it is fixed when z_{k+1} to z_n are fixed. The above equation can be rewritten as $\sum_{k=1}^{n} r_{kk}^2(z_k - c_k)^2 < \beta$, which defines the possible values that z_k can take on. This inequality is equivalent to the following n inequalities:

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

The search starts at level n, heuristically assigning $z_n = \lfloor c_n \rceil$, the nearest integer to c_n. Given the value of z_n, c_{n-1} can be calculated from the above equation as $c_{n-1} = (\bar{y}_{n-1} - r_{n-1,n}z_n)/r_{n-1,n-1}$. From this value, we can set $z_{n-1} = \lfloor c_{n-1} \rceil$ and search continues. 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. At some level $k-1$ in the search, it is likely that the inequality cannot be satisfied, requiring the search to backtrack to a previous decision. When we backtrack from level $k-1$ to level k, we choose z_k to be the next nearest integer to c_k, and so on.

In the Schnorr & Euchner strategy, the initial bound β can be set to ∞. Once the first integer point is found, we can use this integer point to update β, reducing the hyper-ellipsoid thus the search space.

In order to take into account the variable bounds, i.e., $\boldsymbol{l} \leq \boldsymbol{z} \leq \boldsymbol{u}$, the algorithm limits the enumeration of z_k at level k to be within $[l_k, u_k]$.

4 Extending the DEBS Method to Solve the EQKP

From a constraint programming perspective, DEBS does three things. First, the search strategy implies a fixed variable ordering heuristic $(z_n, z_{n-1}, \ldots, z_k, \ldots, z_1)$ that is determined after the reduction phase, where the variables are reordered with the permutation matrix as $\boldsymbol{z} = \boldsymbol{P}^T\boldsymbol{x}$. Second, through the calculation of c_k, DEBS provides a value ordering heuristic: for z_k, integer values closer to c_k are preferred. Third, the inequalities described above induce an interval domain for each z_k. DEBS is essentially the enumeration of these domains under the prescribed variable and value orderings. We use this insight to integrate linear constraints into DEBS in a straightforward way: linear constraints are made arc consistent to further reduce the domains of the z_k variables.

A linear constraint is defined as $\underline{b} \leq \boldsymbol{a}^T\boldsymbol{z} \leq \bar{b}$, where $\underline{b},\ \bar{b} \in \mathbb{R}^n \cup \{\pm\infty\}$, and $\boldsymbol{a} \in \mathbb{R}^n$ is the coefficients of the constraint. Given a linear constraint, let

$$\underline{\alpha}_k = \min\{\boldsymbol{a}^T\boldsymbol{z} - a_k z_k \mid \boldsymbol{l} \leq \boldsymbol{z} \leq \boldsymbol{u}\} \quad \text{and} \quad \bar{\alpha}_k = \max\{\boldsymbol{a}^T\boldsymbol{z} - a_k z_k \mid \boldsymbol{l} \leq \boldsymbol{z} \leq \boldsymbol{u}\}$$

be the minimal and maximal values that $\boldsymbol{a}^T\boldsymbol{z}$ can achieve over all variables except z_k with respect to the variable bounds. These values can be computed by substituting the z_ks with their lower or upper bounds, depending on the signs of the a_ks. The propagation rule for each integer variable z_k is then defined as follows:

$$\left\lceil \frac{\underline{b} - \bar{\alpha}_k}{a_k} \right\rceil \leq z_k \leq \left\lfloor \frac{\bar{b} - \underline{\alpha}_k}{a_k} \right\rfloor \quad \text{if } a_k > 0, \tag{6}$$

$$\left\lceil \frac{\bar{b} - \underline{\alpha}_k}{a_k} \right\rceil \leq z_k \leq \left\lfloor \frac{\underline{b} - \bar{\alpha}_k}{a_k} \right\rfloor \quad \text{if } a_k < 0. \tag{7}$$

Although the idea of propagating the linear constraint is straightforward and not novel, the implementation involves maintaining useful information efficiently as the search moves between levels. As a variable is fixed when the search moves down the levels, the values $\underline{\alpha}_k$ and $\bar{\alpha}_k$ actually consist of two parts: the sum of all the variables that are fixed already (z_{k+1} to z_n), and the sum of the maximum or minimum values that the unfixed variables (z_1 to z_{k-1}) can achieve, according to the variable bounds. Below we show how to update $\underline{\alpha}_k$ and $\bar{\alpha}_k$ using the cardinality constraint as an example.

For the cardinality constraint, we have $a_k = 1, \forall k$, and $\underline{b} = \bar{b} = K$. Therefore, using Equation (6), the lower bound L_k and upper bound U_k imposed by the cardinality constraint can be derived as follows:

$$L_k = K - \sum_{i=k+1}^{n} z_i - \sum_{i=1}^{k-1} u_i \quad \text{and} \quad U_k = K - \sum_{i=k+1}^{n} z_i - \sum_{i=1}^{k-1} l_i,$$

where $\sum_{i=1}^{k-1} l_i = 0$ and $\sum_{i=1}^{k-1} u_i = k-1$, since z_is are binary for the EQKP.

During the search, the sum of the fixed values $S = \sum_{i=k+1}^{n} z_i$ is only changed when moving up or down between two adjacent levels. Therefore, we can use a single variable to keep track this value, efficiently updating S rather than computing from scratch. When the search backtracks from level k to $k+1$, we set $S = S - z_k$ to reverse the assignment of z_k. Similarly, when the search moves down from level k to $k-1$, we set $S = S + z_k$ to take into account the current assignment of z_k.

Equations (6) and (7) are general and they can be applied to all types of linear constraints. In addition, some types of constraints, e.g., the cardinality constraint as shown above, can be specialized from the general linear constraint formulation and be propagated more efficiently. Similarly, each linear constraint keep tracks of its own sum of the current assigned values S. Therefore our extended DEBS can be used to solve problems with any number of linear constraints of any type. The variable domain at each level is therefore the intersection of all the bounds imposed by the linear constraints.

A naive extension of the DEBS method for solving problems with linear constraints can be achieved by using the linear constraints only as "checkers": the constraints simply return true or false when all the variables are instantiated. If all linear constraints agree on an instantiation, it is accepted as an integer solution and the search proceeds correspondingly. However from our preliminary experiments, this naive implementation is, unsurprisingly, orders of magnitude slower than the MIP and SDP approaches. Thus, we do not report the results but note that our simple propagation is critical for results presented below.

Solving Non-Binary Problems. Our extended DEBS method can solve non-binary problems without any modification. However, it requires that the quadratic objective function be inherently convex, as convexifying a non-binary problem is not possible. The reduction and the propagation on the linear constraints are not affected by the variable domains and therefore remain the same.

5 Experimental Results

Experimental Setup. The CPU time limit for each run on each problem instance is 3600 seconds. All experiments were performed on a Intel(R) Xeon(R) CPU E5-1650 v2 3.50GHz machine (in 64 bit mode) with 16GB memory running MAC OS X 10.9.2 with one thread. The DEBS approach is written in C. We use CPLEX Optimization Studio v12.6 and the SDP solver BiqCrunch downloaded from the website [17] for comparison. Both solvers are executed with their default settings. For the SDP solver, there are four specialized versions that deal with problem-specific structures. Three of them are consistent with the EQKP problem and the SDP results presented are the best of the three versions for each individual problem instance, representing the "virtual best" SDP solver. We report the

arithmetic mean CPU time "arith", and the shifted geometric mean CPU time "geo" to find and prove optimality for each problem set.[1]

{0,1} Problems

n	DEBS		DEBS+Red.		CPLEX		SDP	
	arith	geo	arith	geo	arith	geo	arith	geo
10	**0.00**	0.00	**0.00**	0.00	0.04	0.04	0.14	0.14
20	**0.01**	0.01	**0.01**	0.01	0.18	0.18	4.94	4.35
30	0.72	0.68	**0.69**	0.65	18.82	12.21	336.94	152.53
40	115.64	31.91	**109.63**	30.78	1274.05^{2}	401.5^{2}	2374.78^{4}	1238.91^{4}
50	**317.83**	49.87	329.31	50.91	1810.34^{6}	1006.41^{6}	3237.45^{7}	3119.11^{7}

{0,1,2} Problems

n	DEBS		DEBS+Red.		CPLEX	
	arith	geo	arith	geo	arith	geo
10	**0.00**	0.00	**0.00**	0.00	0.01	0.01
20	**0.02**	0.02	**0.02**	0.02	0.26	0.25
30	1.05	0.95	**0.99**	0.90	11.49	6.78
40	**276.28**	65.86	308.52	69.78	1203.59^{2}	281.98^{2}
50	$\mathbf{2354.50}^{6}$	1216.43^{6}	2406.48^{6}	1262.99^{6}	3149.80^{8}	2700.36^{8}

Table 1: A comparison of CPLEX, DEBS and the SDP approach for the EQKP ({0,1} Problems) and its problem variation ({0,1,2} Problems). Bold numbers indicate the best approach for a given problem set. The superscripts indicate the number of instances not solved to optimality within 3600 seconds.

Test sets. We use five sets of the benchmark instances with different sizes generated in the same way as Létocart et al. [1], with 10 instances in each set. For additional comparison, we relax the binary domains $x_j \in \{0,1\}$ to $x_j \in \{0,1,2\}$. This means that we have the option of selecting two "copies" of each element when maximizing the quadratic objective function but $h_{jj} = 0$ (see Formulation 1). To the best of our knowledge, this problem has not been studied in the literature.

In order to ensure convexity, we compute the smallest eigenvalue for the $\boldsymbol{H}$ matrix of each problem and let it be λ_{min}. If λ_{min} is negative, i.e., the problem is non-convex, we apply the perturbation vector $\boldsymbol{u} = (-\lambda_{min} + 0.001)\boldsymbol{e}$ such that the original objective function is transformed to: $\min_{\boldsymbol{x}\in\{0,1\}} \frac{1}{2}\boldsymbol{x}^T\bar{\boldsymbol{H}}\boldsymbol{x} + \bar{\boldsymbol{f}}^T\boldsymbol{x}$, where $\bar{\boldsymbol{H}} = \boldsymbol{H} + \mathrm{diag}(\boldsymbol{u})$ and $\bar{\boldsymbol{f}} = \left(\boldsymbol{f} - \frac{1}{2}\boldsymbol{u}\right)^T$. Note that this convex formulation has the same optimal solution as the original one. For the DEBS method, we use Cholesky decomposition on the perturbed matrix $\bar{\boldsymbol{H}}$ to obtain matrix $\boldsymbol{A}$ in the ILS formulation (3), and we obtain $\boldsymbol{y}$ from the relationship $\bar{\boldsymbol{f}} = -\boldsymbol{y}^T\boldsymbol{A}$, which

[1] 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 [18].

gives us $\boldsymbol{y} = -(\bar{\boldsymbol{f}}\boldsymbol{A}^{-1})^T$. If the problem is originally convex, we obtain $\boldsymbol{A}$ and $\boldsymbol{y}$ with $\boldsymbol{H}$ and $\boldsymbol{f}$.

Results and Discussion. From Table 1, it is clear that the DEBS method performs the best both for the EQKP and its relaxed problem. For the EQKP, DEBS is on average one to two orders of magnitude faster than CPLEX and it strictly dominates CPLEX on each problem instance in our experiments. The SDP approach performs the worst among the three approaches. This is surprising given the strong results for solving the BQPs, and the fact that these results are the best per instance results over the three BiqCrunch variations. Analysis of CPLEX and the SDP solving behaviour shows that the reason that both approaches are unable to prove optimality for larger problems is mainly the weakness in the dual bound. This apparently favours the DEBS method as it does not rely on dual bounding. In fact, DEBS completely lacks a dual bounding mechanism, depending on propagation to reduce the search space.

Interestingly, the running times for all three approaches increase significantly when K is increased. The reason is that during the search, when the sum of the already fixed variables at a node is equal to K, we know that the unfixed variables have to be set to zero to satisfy the cardinality constraint. Intuitively, if K is small, such solutions are early in the search tree, as fewer branching decisions are required to reach such nodes. For the same reason, we expect that the running time to be also decreased when K is further increased towards n.

From Table 1, it is observed that the relaxed domain makes the problem much more difficult to solve, but DEBS still dominates CPLEX. The SDP approach cannot be used to solve the relaxed problem without transforming the problem back to binary domain at the cost of introducing additional variables and constraints. Therefore, we expect its performance to be inefficient.

It is interesting to observe that the reduction does not seem to improve the performance for the EQKP as it does on the BQP problems without general linear constraints [12]. Reduction even worsens performance on the relaxed problems. This suggests that it might be of interest to develop new reduction algorithms that take into the account of the linear constraints.

6 Conclusion

We proposed an extension to the discrete ellipsoid-based search (DEBS) method to solve the exact quadratic knapsack problem (EQKP) and a variation. The core of our extension required the modification of the DEBS approach to incorporate linear constraints. We did this by adopting standard linear constraint propagation algorithms. Results showed that our algorithm outperforms both the state-of-the-art MIP and SDP approaches. For future work, we would like to further extend DEBS to solve general mixed integer convex quadratic programming problems and test its ability as a general solution approach. It is also interesting to investigate dual bounding mechanism for the DEBS search algorithm, as it is possible to compute a dual bound at each level to further limit the search interval.

References

1. Létocart, L., Plateau, M.C., Plateau, G.: An efficient hybrid heuristic method for the 0-1 exact k-item quadratic knapsack problem. Pesquisa Operacional **34**(1) (2014) 49–72
2. Martí, R., Gallego, M., Duarte, A.: A branch and bound algorithm for the maximum diversity problem. European Journal of Operational Research **200**(1) (2010) 36–44
3. Caprara, A., Pisinger, D., Toth, P.: Exact solution of the quadratic knapsack problem. INFORMS Journal on Computing **11**(2) (1999) 125–137
4. Caprara, A., Kellerer, H., Pferschy, U., Pisinger, D.: Approximation algorithms for knapsack problems with cardinality constraints. European Journal of Operational Research **123**(2) (2000) 333–345
5. Turner, S., Romero, D., Zhang, P., Amon, C., Chan, T.: A new mathematical programming approach to optimize wind farm layouts. Renewable Energy **63** (2014) 674–680
6. Zhang, P.Y., Romero, D.A., Beck, J.C., Amon, C.H.: Solving wind farm layout optimization with mixed integer programs and constraint programs. EURO Journal on Computational Optimization **2**(3) (2014) 195–219
7. 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
8. Bussieck, M.R., Vigerske, S.: MINLP solver software. Wiley Encyclopedia of Operations Research and Management Science, Wiley, Chichester (2010)
9. 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
10. Krislock, N., Malick, J., Roupin, F.: Improved semidefinite bounding procedure for solving max-cut problems to optimality. Mathematical Programming (2012) 1–26
11. Teunissen, P.J., Kleusberg, A., Teunissen, P.: GPS for Geodesy. Volume 2. Springer Berlin (1998)
12. Ku, W.Y., Beck, J.C.: Combining discrete ellipsoid-based search and branch-and-cut for binary quadratic programming problems. In: Integration of AI and OR Techniques in Constraint Programming. Springer (2014) 334–350
13. Chang, X.W., Han, Q.: Solving box-constrained integer least squares problems. Wireless Communications, IEEE Transactions on **7**(1) (2008) 277–287
14. 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
15. Ku, W.Y., Beck, J.C.: Combining discrete ellipsoid-based search and branch-and-cut for integer least squares problems. Submitted to IEEE Transactions on Wireless Communications (2014)
16. Schnorr, C.P., Euchner, M.: Lattice basis reduction: Improved practical algorithms and solving subset sum problems. Mathematical programming **66**(1) (1994) 181–199
17. Krislock, N., Malick, J., Roupin, F.: BiqCrunch online solver (2012). http://lipn.univ-paris13.fr/BiqCrunch/download (Retrieved: 11/06/2014)
18. Achterberg, T.: Constraint Integer Programming. PhD thesis, Technische Universität Berlin (2007)