Upper bounds for the binary quadratic knapsack problem

Marco Costa, Marcia Fampa and Daniela Cristina Lubke
COPPE - Universidade Federal do Rio de Janeiro
{marcosil,fampa,danielalubke}@cos.ufrj.br

ABSTRACT

We address the binary quadratic knapsack problem (QKP) of selecting from a set of items, a subset with maximum profit, and whose overall weight does not exceed a given capacity c. The objective function of the problem, which measures the profit of the selection, is a nonconvex quadratic function, and the QKP is naturally formulated as a quadratic binary problem. Several works have proposed relaxations for the QKP varying from linear programs to more sophisticated semidefinite programs. In this work we propose the application of a cutting plane algorithm that iteratively strengthen an initial linear programming relaxation of the problem with the goal of obtaining bounds of good quality, with no need of solving semidefinite programming problems. The valid inequalities added to the initial relaxation are well known and commonly referred to in the literature as RLT inequalities and SDP cuts. Computational results illustrate the trade-off between the quality of the bounds computed and the computational effort required by the cutting plane algorithm.

KEY WORDS. quadratic knapsack problem, upper bound, semidefinite programming; cutting plane algorithm.

Main area: PM - Mathematical Programming

1. Introduction

In this paper, we address the binary quadratic knapsack problem (QKP) where we are given: a knapsack capacity c, a set of items $N = \{1, \ldots, n\}$, where each item j has a positive integer weight w_j, and an $n \times n$ symmetric nonnegative integer matrix $P = \{p_{ij}\}$, where p_{jj} corresponds to the profit achieved if item j is selected from N and $2p_{ij}$ corresponds to the profit achieved if both items i and j are selected. The QKP is then defined as the problem of selecting a subset of items from N with maximum profit, whose overall weight does not exceed c. Defining a binary variable x_j which indicates whether or not item j is selected, the problem may be formulated as:

$$\begin{array}{lll} (QKP) & \text{maximize} & \sum_{i \in N} \sum_{j \in N} p_{ij} x_i x_j \\ & \text{subject to} & \sum_{j \in N} w_j x_j \leq c, \\ & & x_j \in \{0, 1\}, \qquad j \in N. \end{array}$$

The QKP was introduced by Gallo et al. (Gallo *et al.*, 1980) and was proved to be NP-Hard by reduction from the clique problem. The simplicity of the QKP formulation together with its difficulty have brought a lot of attention to the problem in the last decades. Several papers have proposed branch-and-bound algorithms for the QKP and the main difference between them is the method used to obtain upper bounds for the subproblems (Chaillou *et al.*, 1989; Billionnet and Calmels, 1996; Caprara *et al.*, 1999; Billionnet *et al.*, 1999; Helmberg *et al.*, 1996, 2000). The well known trade-off between the strength of the bounds and the computational effort required to obtain them is intensively discussed in (Pisinger, 2007), where semidefinite programming (SDP) relaxations proposed in (Helmberg *et al.*, 1996) and (Helmberg *et al.*, 2000) are presented as the strongest relaxations for the QKP.

Although SDP relaxations have been very effective in generating tight bounds for integer programming problems since the seminal works (Lovász, 1979; Lovász and Schrijver, 1991; Goemans and Williamson, 1995), it is well known that the required computational effort to solve the relaxations may be considerable, especially when the size of the relaxation becomes too big due to the inclusion of valid inequalities. To overcome this difficulty, linear programming (LP) outer approximations of the SDP relaxations have been investigated in several works. For example, in (Sherali and Fraticelli, 2002), the authors propose LP relaxations of SDP constraints with the aim of capturing most of the strength of SDP relaxations. A cutting plane algorithm is used to iteratively strengthen the initial LP relaxation with the addition of the so-called SDP cuts.

In this paper, we initially consider the strongest SDP relaxation of the QKP presented in (Pisinger, 2007) and eliminate from it the SDP constraint that ensures that the matrix variable is positive semidefinite. The linear relaxation thereby obtained is a weak relaxation. We strengthen it using the well known RLT (Reformulation Linearization Technique) inequalities and also the SDP cuts. We apply to the QKP, a cutting plane algorithm based on the work presented in (Sherali and Fraticelli, 2002), aiming at obtaining tight bounds for the problem by solving only LP relaxations. Computation results compare the bounds computed with the tight bounds obtained with the SDP relaxation of the problem. The trade-off between the quality of the bounds and the computational effort required by the cutting plane algorithm is also investigated.

Notation

Given two symmetric $n \times n$ real matrices X, Y, we define the inner product between X and Y as $\langle X, Y \rangle = \text{trace}(X^T Y) = \sum_{i,j=1}^{n} X_{ij} Y_{ij}$. We use $X \succeq 0$ to denote that the matrix X is positive semidefinite and diag(X) to denote the vector in $\Re^n$ of diagonal elements of X.

2. A strong SDP bound from the literature

Besides (QKP), an alternative lifted formulation for the QKP is obtained by replacing each quadratic term $x_i x_j$ with a new variable X_{ij}. Defining the symmetric matrix $X = xx^T$ as the matrix with entry X_{ij}, the QKP is equivalent to

$$
\begin{array}{lll}
(QKP_{lifted}) & \text{maximize} & \langle P, X\rangle \\
& \text{subject to} & \sum_{j\in N} w_j x_j \le c, \\
& & x_j \in \{0,1\}, \qquad j \in N. \\
& & X = xx^T.
\end{array}
$$

The difficulty in solving problem (QKP_{lifted}) comes from the nonconvex constraints $x_j \in \{0,1\}$ and $X = xx^T$. Relaxations for the QKP have been obtained by relaxing the integrality constraints to $x_j \in [0,1]$, and relaxing the constraint $X = xx^T$ in two possible ways:

- By replacing the constraint $X = xx^T$ with the convex inequality $X - xx^T \succeq 0$, or equivalently, using Schur's complement, with the linear SDP inequality

$$
\begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0. \tag{1}
$$

- By replacing the constraint $X = xx^T$ with linear inequalities known as RLT inequalities. These inequalities are obtained by the Reformulation Linearization Technique (RLT) (Sherali and Adams, 1998), using products of pairs of original constraints and bounds and replacing each nonlinear term $x_i x_j$ with a new variable X_{ij}, as follows:

  1. For every pair of variables x_i and x_j, $i, j \in \{1, \ldots, n\}$, we consider the bound constraints $0 \le x_i \le 1$ and $0 \le x_j \le 1$, obtaining

$$
\begin{array}{l}
X_{ij} \le x_i, \\
X_{ij} \le x_j, \\
x_i + x_j \le 1 + X_{ij}, \\
X_{ij} \ge 0.
\end{array} \tag{2}
$$

  2. For every variable x_i, $i \in \{1, \ldots, n\}$, we consider the bound constraint $x_i \ge 0$ and the capacity constraint

$$
\sum_{j\in N} w_j x_j \le c, \tag{3}
$$

  obtaining

$$
\sum_{j\in N} w_j X_{ij} \le c x_i. \tag{4}
$$

Finally, considering that all variables in the QKP are binary variables, we have that $X_{ii} := x_i x_i = x_i$, for all $i \in \{1, \ldots, n\}$ or, equivalently,

$$
\text{diag}(X) = x. \tag{5}
$$

As a consequence, we have that

$$
X_{ii} \le 1, \tag{6}
$$

and we can strengthen inequality (1) to

$$
\begin{pmatrix} 1 & \text{diag}(X)^T \\ \text{diag}(X) & X \end{pmatrix} \succeq 0, \tag{7}
$$

or, equivalently, to

$$X - \text{diag}(X)\text{diag}(X)^T \succeq 0. \tag{8}$$

In (Helmberg *et al*., 1996) and (Helmberg *et al*., 2000), Helmberg, Rendl, and Weismantel propose different SDP relaxations for the QKP based on the lifted formulation (QKP_{lifted}) and on the relaxations presented above. From the comparison numerical results presented in (Pisinger, 2007) between different bounds for the problem, we conclude that the strongest relaxation is the SDP problem formulated as

$$\begin{array}{lll} (HRW) & \text{maximize} & \langle P, X\rangle \\ & \text{subject to} & \sum\limits_{j\in N} w_j X_{ij} - X_{ii}c \le 0, \qquad i \in N, \\ & & X - \text{diag}(X)\text{diag}(X)^T \succeq 0, \end{array}$$

Problem (HRW) is derived from the lifted formulation (QKP_{lifted}), where the capacity constraint (3) is replaced by (4) and the nonconvex constraint $X = xx^T$ is relaxed to (8). Note that the bound constraints $0 \le x_i \le 1$ are also ensured by (8).

3. New upper bounds

In this section we investigate the application of a cutting plane algorithm to iteratively obtain tighter bounds for the QKP. At each iteration of the the algorithm a stronger LP relaxation is solved, obtained with the addition of SDP cuts. The idea is to iteratively construct an outer approximation of the feasible set of the lifted problem (QKP_{lifted}) by solving a sequence of LP problems. At each iteration of the procedure the cut added to the LP formulation eliminates the solution of the previous relaxation from the feasible set, turning the bound tighter. The goal is to derive as good bounds as the SDP relaxation (HRW), but solving only LP problems.

The procedure initiates taking into account the following model

$$\begin{array}{llll} (\tilde{LP}) & \text{maximize} & \langle P, X\rangle & \\ & \text{subject to} & \sum\limits_{j\in N} w_j X_{jj} \le c, & \\ & & \sum\limits_{j\in N} w_j X_{ij} - X_{ii}c \le 0, & i \in N, \\ & & X = X^T, & \\ & & X_{ij} \le X_{ii}, & i,j \in N, i < j, \\ & & X_{ij} \le X_{jj}, & i,j \in N, i < j, \\ & & X_{ij} \ge 0, & i,j \in N, i < j, \\ & & 0 \le X_{jj} \le 1, & j \in N, \end{array}$$

which corresponds to relaxation (HRW) weakened on one side by the relaxation of the SDP constraint $X - \text{diag}(X)\text{diag}(X)^T \succeq 0$ to $X = X^T$, and strengthened on the other side by the addition of the capacity constraint (3), the RLT inequalities (2) and the bound inequalities (6). We didn't include the RLT inequalities $X_{ii}+X_{jj} \le 1+X_{ij}$ in the model because they are not necessary, once we have $p_{ij} \ge 0$ in the objective function.

We note here that in (Billionnet and Calmels, 1996), Billionnet and Calmels propose a slightly weaker LP relaxation for the QKP, given by

$$
\begin{array}{lll}
(BC) & \text{maximize} & \sum_{i,j\in N, i<j} 2p_{ij}y_{ij} + \sum_{j\in N} p_{jj}x_j \\
& \text{subject to} & \sum_{j\in N} w_j x_j \leq c, \\
& & y_{ij} \leq x_i, \qquad\qquad\qquad i,j\in N, i<j, \\
& & y_{ij} \leq x_j, \qquad\qquad\qquad i,j\in N, i<j, \\
& & x_i + x_j \leq 1 + y_{ij}, \qquad i,j\in N, i<j, \\
& & y_{ij} \geq 0, \qquad\qquad\qquad i,j\in N, i<j, \\
& & 0 \leq x_j \leq 1, \qquad\qquad\quad j\in N.
\end{array}
$$

In (Pisinger, 2007), the authors present (BC) as one of the weakest, and also cheapest to solve, relaxations of the QKP. Our goal with the cutting plane algorithm proposed in this work, is to generate bounds tighter than the solution of (BC) and cheaper to compute than the ones given by (HRW).

3.1. A cutting plane algorithm

Let us define the $(n+1)\times(n+1)$ symmetric matrix Y as

$$Y := \begin{pmatrix} 1 & \mathrm{diag}(X)^T \\ \mathrm{diag}(X) & X \end{pmatrix}. \tag{9}$$

In the remainder of this subsection, we describe the cutting plane algorithm to strengthen the initial relaxation $(\tilde{LP})$. The procedure is based in the equivalences:

$Y \succeq 0$ if and only if $X - \mathrm{diag}(X)\mathrm{diag}(X)^T \succeq 0$,

$Y \succeq 0$ if and only if $v^T Y v \geq 0,$ for all $v \in \Re^{n+1}$,

and iteratively adds to the relaxation of the QKP, SDP cuts of the form $\bar{v}^T Y \bar{v} \geq 0$, where the vectors $\bar{v}$ are judiciously selected, as done in (Sherali and Fraticelli, 2002) for nonconvex programming problems.

For the $(n+1)\times(n+1)$ symmetric matrix Y, its spectral decomposition is given by

$$Y = \sum_{k=1}^{n+1} \lambda_k v_k v_k^T,$$

where λ_k and v_k, for $k = 1,\ldots,n+1$, are respectively, the eigenvalues and corresponding orthonormal eigenvectors of Y. If $Y \succeq 0$, then $\lambda_k \geq 0$ for all $k = 1,\ldots,n+1$, otherwise there is at least one $\bar{k}$ such that $\lambda_{\bar{k}} < 0$.

As

$$v_{\bar{k}}^T Y v_{\bar{k}} = \lambda_{\bar{k}},$$

the inequality

$$v_{\bar{k}}^T Y v_{\bar{k}} \geq 0,$$

which is satisfied by all positive semidefinite $(n+1)\times(n+1)$ matrices, is violated by Y.

The cutting plane algorithm presented in Figure 1 uses the ideas discussed above to iteratively separate SDP cuts, and add them to our initial formulation $(\tilde{LP})$ in order to tight the bound computed.

The stopping criterion $StoppingCriterion$, mentioned in Figure 1 could impose the cutting plane algorithm to stop only when the matrix $\tilde{Y}$ becomes positive semidefinite, or in other words, only when $\lambda_k \geq 0$, for all $k = 1,\ldots,n+1$. In this case the bound computed by the algorithm would not be worse than the bound given by (HRW). Nevertheless, the computational effort required to satisfy this criterion may be too big to compensate. The analysis of the trade-off between the quality of the bound obtained by the cutting plane algorithm and the computational effort required is the main focus of our numerical experiments described in the next section.

procedure CUTTINGPLANEALGORITHM(CPA)
  while $StoppingCriterion$ **do**
    Let $\tilde{X}$ be an optimal solution of $(\tilde{LP})$;
    Let $\tilde{Y} := \begin{pmatrix} 1 & \text{diag}(\tilde{X})^T \\ \text{diag}(\tilde{X}) & \tilde{X} \end{pmatrix}$;
    Let λ_k and v_k for $k = 1, \ldots, n+1$ be respectively, the eigenvalues and corresponding orthonormal eigenvectors of $\tilde{Y}$, such that $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{n+1}$;
    Let $k := 1$;
    while $\lambda_k < \lambda_{MAX}$ **and** $k \leq K_{MAX}$ **do**
      Add the constraint $v_k^T Y v_k \geq 0$ to $(\tilde{LP})$, where Y is defined in (9);
      $k := k + 1$;
    end while
  end while
  return the optimal solution value of $(\tilde{LP})$.
end procedure

Figure 1: Cutting plane algorithm

4. Preliminary Numerical experiments

Our code was implemented in Matlab R2014a using the convex optimization toolbox CVX 2.1 (Grant and Boyd, 2014) and the solver MOSEK 7.1 (Andersen and Andersen, 1999). All runs were conducted on a 1.90GHz Intel(R) Core i7 CPU, 4GB, running under Linux Ubuntu, version 14.04.

In our experiments, we used the same randomly generated instances that were used by Jesus Cunha in (Cunha, 2014). Cunha also provided us the optimal solutions of the instances. The instances are denoted in Table 1 presented below by $\mathrm{I}_{n,d,i}$, where

- n is the number of variables,
- d is the density of the profit matrix P, i.e., the percentage of positive profits $p_{ij}, i \leq j, i, j \in N$, which are randomly selected in the interval $[1, 100]$,
- i is the instance index.

The capacity of the knapsack c is randomly selected in the interval $[50, \sum_{j=1}^{n} w_j]$ and the weight w_j is randomly selected in the interval [1, 50], for each $j \in N$. The procedure used by Cunha to generate the instances was based on other previous works (Billionnet and Calmels, 1996; Caprara *et al.*, 1999; Chaillou *et al.*, 1989; Gallo *et al.*, 1980; Michelon and Veilleux, 1996).

The aim of our experiments is to compare the upper bounds for the QKP that are obtained with two relaxations from the literature, (HRW) and (BC), and with different versions of our cutting plane algorithm (CPA), where what differs in the versions is the maximum number of SDP cuts that are added to the relaxation at each iteration, denoted in Figure 1 by K_{MAX}. More specifically, we compare the upper bounds obtained with the five following relaxations:

- The SDP relaxation (HRW) strengthened by (3) and (2) (SDP).
- The LP relaxation (BC) (LP).
- The LP relaxation obtained with our CPA, considering $K_{MAX} = 1$ (CPA_1).
- The LP relaxation obtained with our CPA, considering $K_{MAX} = 5$ (CPA_5).
- The LP relaxation obtained with our CPA, considering $K_{MAX} = 10$ (CPA_{10}).

The stopping criterion, identified in Figure 1 as $StoppingCriterion$, was chosen on these preliminary numerical experiments with the goal of allowing a good analysis of the convergence of the CPA. We run the tests with a time limit of 360 seconds or until the matrix variable Y becomes positive semidefinite. In order to avoid a premature interruption of the runs due to lack of memory, we check at each 5 iterations, which SDP cuts are not active, and eliminate them from the model.

Table 1 presents the results of our experiments. In the first column we specify the instance considered. In the other columns we shows the relative gap between the upper bound (UB_i) obtained with the i-th relaxation, and the optimal solution value of the problem z^*, specifically given by

$$\text{Gap} = \frac{UB_i - z^*}{z^*} \times 100, \text{ for } i = 1, \ldots, 5.$$

Instance	LP	SDP	CPA1	CPA5	CPA10
$I_{100,25,1}$	0.25	0.16	0.25	0.25	0.25
$I_{100,50,1}$	1.11	0.04	0.37	0.16	0.17
$I_{100,75,1}$	6.01	0.49	0.50	0.49	0.49
$I_{100,100,1}$	3.46	0	0	0	0
$I_{100,25,2}$	5.77	0.76	1.42	1.21	1.31
$I_{100,50,2}$	2.82	0.43	0.48	0.48	0.47
$I_{100,75,2}$	1.67	0.20	0.23	0.22	0.23
$I_{100,100,2}$	2.51	0.46	0.46	0.46	0.46
$I_{100,25,4}$	1.05	0.12	1.05	0.55	0.57
$I_{100,50,4}$	3.96	0.19	0.76	0.70	0.65
$I_{100,75,4}$	2.55	0.10	0.20	0.13	0.14
$I_{100,100,4}$	4.32	0.13	0.13	0.13	0.13
$I_{200,25,1}$	0.16	-	0.16	0.16	0.16
$I_{200,50,1}$	0.16	-	0.16	0.16	0.16
$I_{200,75,1}$	16.83	-	0.51	0.48	0.48
$I_{200,100,1}$	0.06	-	0.03	0.03	0.03
Mean	3.29	0.26	0.42	0.35	0.36

Table 1: Gaps obtained with different relaxations for the QKP

We first note that the time limit of 360 seconds is longer than the time required to solve SDP, whenever SDP can be solved, i.e. whenever $n = 100$. The maximum time required to solve SDP when $n = 100$ is 210 seconds. Nevertheless, SDP cannot be solved for any instance with $n = 200$, due to lack of memory or numerical problems, which confirms the well known difficulty in solving strong SDP relaxations as the number of variables increases. The time required to solve LP for all instances is very small, up to 0.6 seconds, however, the bounds given by this relaxation sometimes are very weak. Our next goal on our research, is to define a stopping criterion for the CPAs, such that they can obtain in average a better bound than LP, but in a shorter time than SDP. In these preliminary tests, however, we allowed the CPAs to run for a longer time than SDP, in order to analyze their convergence behavior. We also note that for CPA_5 and CPA_{10}, several runs were interrupted for lack of memory. For all others the stopping criterion was the the time limit of 360 seconds. On no run of the CPAs, we obtained a positive semidefinite matrix Y.

From the results in Table 1, we see that CPA_1 finds a better bound than LP for 12 out of 16 instances, and the other two CPAs find a better bound than LP for 13 out of 16 instances. Also CPA_1 obtains the same bound as SDP for 3 instances, while the other two CPAs obtain the same bound as SDP for 4 instances. It's also worth to mention that for the 2 out of 4 instances for which SDP fails, the CPAs obtain better bounds than LP. The most impressive result was for instance $I_{200,75,1}$, as the gap obtained by LP is 16.83%, while the gaps obtained by the CPAs are very good, 0.51% and 0.48%. Finally, we point out that for the only instance for which SDP obtains the optimal solution value, i.e., gap equal to 0%, the three CPAs also obtain it. For this same instance ($I_{100,100,1}$), LP finds a gap of 3.46%. The average gaps presented in the last row of Table 1 show how close to the SDP bounds, the CPAs can get in 360 seconds, and with the amount of memory available.

Figure 2 shows the behavior of the three cutting plane algorithms during the iterations, for four instances. The horizontal and vertical axes on the graphics correspond, respectively, to the number of iterations and the relative gap. The two horizontal lines on the graphics indicate to the bounds given by the relaxations (LP) and (SDP).

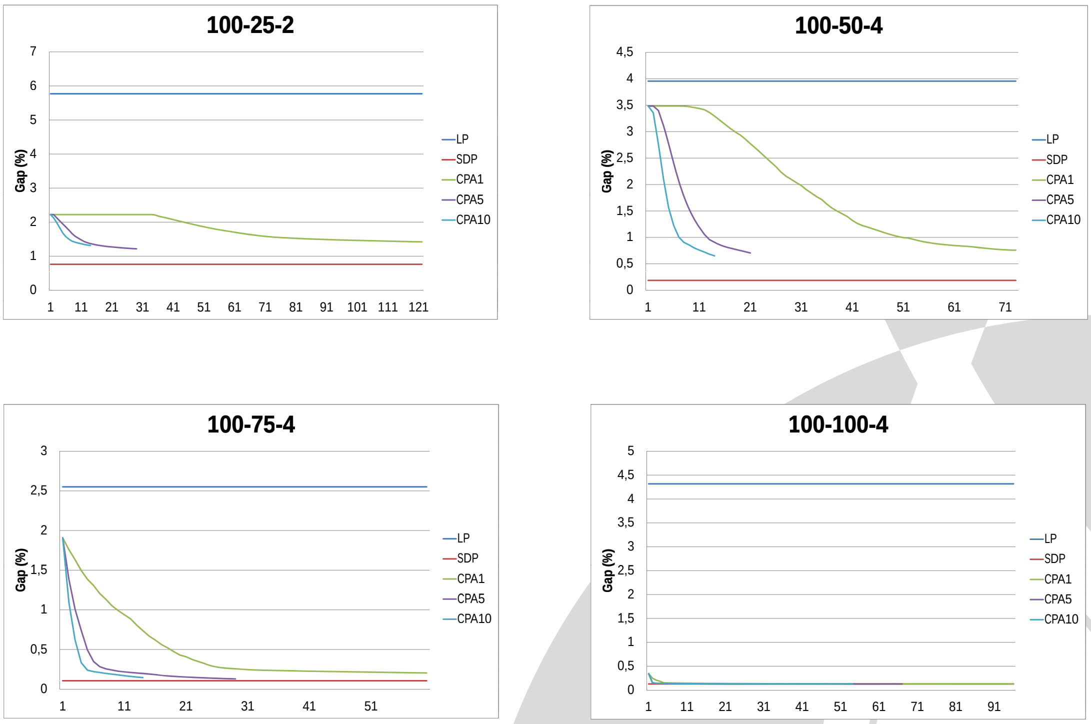

Figure 2: Bounds during the execution of the CPAs

In Figure 2 we see that as we increase the number of cuts added to the LP relaxations at each iteration of the CPA algorithm, better bounds are computed in less iterations. This result indicates that the cuts added are being really effective. It is clear that the convergence of CPA_1 is much slower than of the other two CPAs. Together with the average results from Table 1, this rules out the CPA_1 algorithm from our future research, leaving CPA_5 and CPA_{10} as better options. Furthermore, we see that it is common to get basically the same bounds with these two last CPAs in 360 seconds, so it is important now to have a better analysis of the time spent at each iteration of these algorithms to identify the best limit on the number of cuts to be added. Finally, Figure 2 shows that the bounds given by the CPAs are always between the ones given by LP and SDP. Furthermore, at each iteration, the CPAs' bounds get closer to the SDP bounds and become more distant from the LP bounds.

5. Conclusion

Several works have proposed different relaxations to the quadratic knapsack problem (QKP). The analysis of the trade-off between the quality of the bounds and the computational effort required to compute them has been done in the literature and is an important tool for the development of successful branch-and-bound algorithms for the QKP. In this work we propose the application of a cutting plane algorithm that iteratively solves stronger linear programming relaxations of the problem, adding valid inequalities well known in the literature as RLT inequalities and SDP cuts. Similar ideas have been proposed for more general nonconvex quadratic problems in the literature. Here, we specialize the ideas to better fit the QKP, and compare, through numerical experiments, different versions of the cutting plane algorithm. We conclude that the methodology proposed gives promising results for the QKP, the bounds computed for some instances are much better than the ones given by a simpler linear programming relaxation, and can be computed more efficiently than a stronger SDP relaxation, when the number of variables increases. We note that the computational time of the cutting plane algorithm is still large, when compared to other linear programming relaxations. However, in future work we plan to apply some well known techniques to reduce it, as for example, to consider the RLT inequalities to be added to the original relaxation, also with a separation algorithm, as it is done for the SDP cuts. Procedures to eliminate from the relaxation inactive RLT constraints at its optimal solution, at each iteration of the cutting plane algorithm, can also be investigated. Finally, the sparsification of the SDP cuts has been studied in (Qualizza *et al.*, 2011) and successfully applied to quadratically constrained quadratic programs, in order to reduce the computational time of a similar cutting plane algorithm. The authors mention in the paper that the usual high density of the SDP cuts, in general leads to a slow cutting plane algorithm, and propose a procedure that generate sparse SDP cuts. The same procedure can be applied to the QKP, and is part of our future research, to investigate it as well.

Acknowledgements

Marco Costa was supported by a Research Grant from CAPES. Daniela Cristina Lubke was supported by Research Grants from CAPES and from PSR - Power Systems Research, Rio de Janeiro, Brazil. The authors are also grateful to Jesus O. Cunha for providing us the test problems.

References

Andersen, E. and Andersen, K. The mosek interior point optimizer for linear programming: An implementation of the homogeneous algorithm. et al., H. F. (Ed.), *High Performance Optimization; Appl. Opt.*, volume 33. Kluwer, 1999.

Billionnet, A. and Calmels, F. (1996), Linear programming for the 0-1 quadratic knapsack problem. *European Journal of Operational Research*, v. 92, n. 2, p. 310–325.

Billionnet, A., Faye, A. and Soutif, É. (1999), A new upper bound for the 0-1 quadratic knapsack problem. *European Journal of Operational Research*, v. 112, n. 3, p. 664–672.

Caprara, A., Pisinger, D. and Toth, P. (1999), Exact solution of the quadratic knapsack problem. *INFORMS Journal on Computing*, v. 11, n. 2, p. 125–137.

Chaillou, P., Hansen, P. and Mahieu, Y. Best network flow bounds for the quadratic knapsack problem. Simeone, B. (Ed.), *Combinatorial Optimization*, volume 1403 of *Lecture Notes in Mathematics*, p. 225–235. Springer Berlin Heidelberg, 1989.

Cunha, J. O. *Algoritmos para o problema da mochila quadrática 0-1*. Tese de D.Sc., Universidade Federal do Rio de Janeiro - COPPE, Rio de Janeiro, RJ, Brasil, 2014.

Gallo, G., Hammer, P. and Simeone, B. Quadratic knapsack problems. Padberg, M. (Ed.), *Combinatorial Optimization*, volume 12 of *Mathematical Programming Studies*, p. 132–149. Springer Berlin Heidelberg, 1980.

Goemans, M. X. and Williamson, D. P. (1995), Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of ACM*, v. 42, n. 6, p. 1115–1145.

Grant, M. and Boyd, S. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, 2014.

Helmberg, C., Rendl, F. and Weismantel, R. Quadratic knapsack relaxations using cutting planes and semidefinite programming. Cunningham, W., McCormick, S. and Queyranne, M. (Eds.), *Integer Programming and Combinatorial Optimization*, volume 1084 of *Lecture Notes in Computer Science*, p. 175–189. Springer Berlin Heidelberg, 1996.

Helmberg, C., Rendl, F. and Weismantel, R. (2000), A semidefinite programming approach to the quadratic knapsack problem. *Journal of Combinatorial Optimization*, v. 4, n. 2, p. 197–215.

Lovász, L. (1979), On the shannon capacity of a graph. *IEEE Transactions on Information Theory*, v. 25, n. 1, p. 1–7.

Lovász, L. and Schrijver, A. (1991), Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, v. 1, p. 166–190.

Michelon, P. and Veilleux, L. (1996), Lagrangean methods for the 0-1 quadratic knapsack problem. *European Journal of Operational Research*, v. 92, n. 2, p. 326 – 341.

Pisinger, D. (2007), The quadratic knapsack problem-a survey. *Discrete Applied Mathematics*, v. 155, p. 623–648.

Qualizza, A., Belotti, P. and Margot, F. Linear programming relaxations of quadratically constrained quadratic programs. Lee, J. and Leyfer, S. (Eds.), *IMA Volumes Series*, volume 154. Springer, 2011.

Sherali, H. D. and Adams, W. P. *A reformulation-linearization technique for solving discrete and continuous nonconvex problems*. Kluwer, Dordrecht, 1998.

Sherali, H. D. and Fraticelli, B. M. P. (2002), Enhancing rlt relaxations via a new class of semidefinite cuts. *J. Global Optim.*, v. 22, p. 233–261.