Upper bounds for the binary quadratic knapsack problem

Marco Costa, Marcia Fampa and Daniela Cristina Lubke

Federal University of Rio de Janeiro
Systems Engineering and Computer Sciences Department, UFRJ-COPPE-PESC

SBPO 2015: XLVII Simpósio Brasileiro de Pesquisa Operacional
Porto de Galinhas, Pernambuco, 25-28 de agosto de 2015

COPPE UFRJ

COPPE UFRJ

Introduction

Quadratic Knapsack Problem

- $N = \{1, \ldots, n\}$ - set of items;
- w_j - weight of item j ;
- $P = \{p_{ij}\}$ - symmetric nonnegative integer matrix $n \times n$ with profit achieved if item j;
- x_j - binary variable which indicates whether or not item j is selected;
- c - knapsack capacity $c \in \mathbb{N}$

The QKP is defined as the problem of selecting a subset of items from N with maximum profit, whose overall weight does not exceed c.

Introduction

The binary quadratic knapsack problem (QKP) was introduced by Gallo et al.:

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

Semidefinite Programming

Notation

- X, Y - symmetric $n \times n$ real matrices;
- $\langle X, Y \rangle = \text{trace}(X^T Y) = \sum_{i=1}^n \sum_{j=1}^n x_{ij} y_{ij}$ - Inner product between X, Y;
- $X \succeq 0$ - X is positive semidefinite.
- $\text{diag}(X) = (x_{11}, \cdots, x_{nn})^T$ - vector in $\mathbb{R}^n$ of diagonal elements of X.

QKP_{lifted}

Replacing each quadratic term $x_i x_j$ with a new variable X_{ij} and defining the symmetric matrix $X = xx^T$ as the matrix with entry X_{ij}, the QKP is equivalent to:

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

QKP_{lifted}

Replacing each quadratic term $x_i x_j$ with a new variable X_{ij} and defining the symmetric matrix $X = xx^T$ as the matrix with entry X_{ij}, the QKP is equivalent to:

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

Relaxations for the QKP

Relaxations for the QKP have been obtained by relaxing the integrality constraints, $x_j \in \{0,1\}$, to $x_j \in [0,1]$.
Relaxing the constraint $X = xx^T$ in two possible ways:

A) 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

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

Relaxations for the QKP

B) By replacing the constraint $X = xx^T$ with linear inequalities known as RLT (Reformulation Linearization Technique) inequalities.

RLT inequalities replacing each nonlinear term $x_i x_j$ with a new variable X_{ij}.

Reformulation Linearization Technique - RLT

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

$$\begin{aligned} &X_{ij} \leq x_i, \\ &X_{ij} \leq x_j, \\ &x_i + x_j \leq 1 + X_{ij}, \\ &X_{ij} \geq 0. \end{aligned} \tag{2}$$

COPPE UFRJ

Reformulation Linearization Technique - RLT

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

$$\sum_{j \in N} w_j x_j \leq c$$

COPPE UFRJ

Reformulation Linearization Technique - RLT

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

$$\sum_{j \in N} w_j x_j \leq c \qquad \longleftarrow \times \quad x_i, \tag{3}$$

obtaining

Reformulation Linearization Technique - RLT

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

$$\sum_{j \in N} w_j x_j \leq c \qquad \longleftarrow \times \quad x_i, \tag{3}$$

obtaining

$$\sum_{j \in N} w_j \overbrace{X_{ij}}^{x_j x_i} \leq c x_i. \tag{4}$$

COPPE UFRJ

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

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

As a consequence, we have that

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

COPPE UFRJ

We can strengthen inequality

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

to

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

using Schur's complement:

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

Replacing $X_{ii} = x_i$ in capacity constraint:

$$\sum_{j \in N} w_j X_{ij} \leq c x_i. \tag{9}$$

obtaining

$$\sum_{j \in N} w_j X_{ij} - X_{ii} c \leq 0. \tag{10}$$

SDP problem

Helmberg, Rendl, and Weismantel propose a SDP relaxation for the QKP, given by

$$
\begin{array}{llll}
(HRW) & \text{maximize} & \langle P, X\rangle & \\
& \text{subject to} & \sum_{j\in N} w_j X_{ij} - X_{ii} c \leq 0, & i \in N, \\
& & X - \operatorname{diag}(X)\operatorname{diag}(X)^T \succeq 0, &
\end{array}
$$

New upper bounds

Construct an outer approximation of the feasible set of the lifted problem (QKPlifted) 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.

GOAL: derive good bounds as the SDP relaxation (HRW), but solving only LP problems.

Relaxation of the SDP constraint in (HRW)

$X - \mathrm{diag}(X)\mathrm{diag}(X)^T \succeq 0 \quad \text{to} \quad X = X^T$

and using RLT inequalities we obtaining $(\tilde{L}P)$ model:

$(\tilde{LP})$

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

Weaker Relaxation

Billionnet and Calmels propose a slightly weaker LP relaxation for the QKP, given by

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

A cutting plane algorithm - CPA

Our goal with the cutting plane algorithm is to generate bounds tighter than the solution of (BC) and cheaper to compute than the ones given by (HRW).

A cutting plane algorithm - CPA

Let us define the symmetric matrix $Y_{(n+1)\times(n+1)}$ as

$$Y := \begin{pmatrix} 1 & \text{diag}(X)^T \\ \text{diag}(X) & X \end{pmatrix}. \tag{10}$$

The procedure describe in cutting plane algorithm is based in this equivalences:

$$\begin{aligned} Y \succeq 0 \quad & \text{if and only if} \quad X - \text{diag}(X)\text{diag}(X)^T \succeq 0, \\ Y \succeq 0 \quad & \text{if and only if} \quad v^T Y v \geq 0, \text{ for all } v \in \mathbb{R}^{n+1}, \end{aligned}$$

We adds to the relaxation of the QKP, SDP cuts of the form $\bar{v}^T Y \bar{v} \geq 0$.

The spectral decomposition of Y is given by

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

where:

- λ_k - eigenvalues of Y
- v_k, for $k = 1, \ldots, n+1$ - 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 CPA iteratively separate SDP cuts, and add them to our initial formulation $(\tilde{LP})$ in order to tight the bound computed.

We could impose the cutting plane algorithm to stop only when the matrix $\tilde{Y}$ becomes positive semidefinite.

In this case the bound 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.

procedure CuttingPlaneAlgorithm(CPA)
  while *StoppingCriterion* **do**
    Let $\tilde{X}$ be an optimal solution of $(\tilde{LP})$;
    Let $\tilde{Y} := \begin{pmatrix} 1 & \mathrm{diag}(\tilde{X})^T \\ \mathrm{diag}(\tilde{X}) & \tilde{X} \end{pmatrix}$;
    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 (20);
      $k := k + 1$;
    end while
  end while
  return the optimal solution value of $(\tilde{LP})$.
end procedure

Numerical experiments

Our code was implemented in Matlab R2014a using the convex optimization toolbox CVX 2.1 and the solver MOSEK 7.1.

All runs were conducted on a 1.90GHz Intel(R) Core i7 CPU, 4GB, running under Linux Ubuntu, version 14.04.

Numerical experiments

The main focus of our numerical experiments is:

The analysis of the trade-off between the quality of the bound obtained by the cutting plane algorithm and the computational effort required.

Instances

The procedure to generate the instances was based on:

- Billionnet and Calmels, 1996;
- Caprara et al., 1999;
- Chaillou et al., 1989;
- Gallo et al., 1980;
- Michelon and Veilleux, 1996.

In our experiments, we used the same randomly generated instances that were used by Jesus Cunha.

Instances

Cunha also provided us the optimal solutions of the instances.

The instances are denoted in Table 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.

COPPE UFRJ

Instances

The capacity of the knapsack c is randomly selected in the interval $[50, \sum_{j=1}^{n} w_j]$;

The weight w_j is randomly selected in the interval $[1, 50]$, for each $j \in N$.

Goal of experiments

Compare the upper bounds for the QKP that are obtained with two relaxations (HRW) and (BC) with different versions of our cutting plane algorithm (CPA).

We compare the upper bounds obtained with the five following relaxations:

- **LP** - The LP relaxation (BC);
- **SDP** - The SDP relaxation (HRW);
- **CPA$_1$** - The LP relaxation obtained with our CPA, considering $K_{MAX} = 1$;
- **CPA$_5$** The LP relaxation obtained with our CPA, considering $K_{MAX} = 5$;
- **CPA$_{10}$** The LP relaxation obtained with our CPA, considering $K_{MAX} = 10$.

COPPE UFRJ

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 : Gaps obtained with different relaxations for the QKP

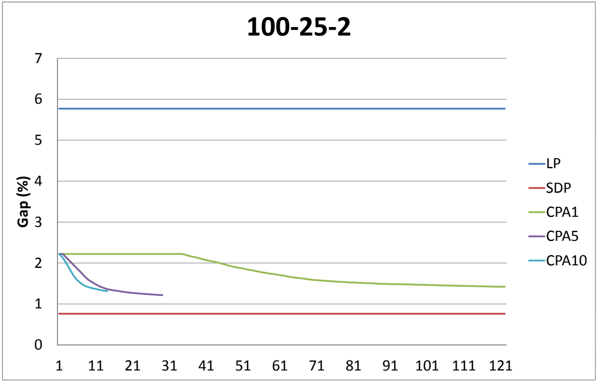

Figure : Bounds during the execution of the CPAs: 100-25-2

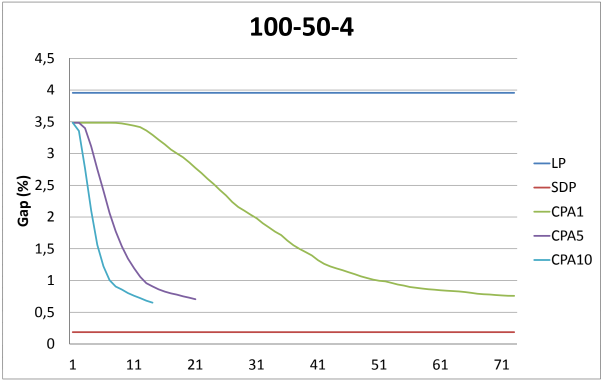

Figure : Bounds during the execution of the CPAs: 100-50-4

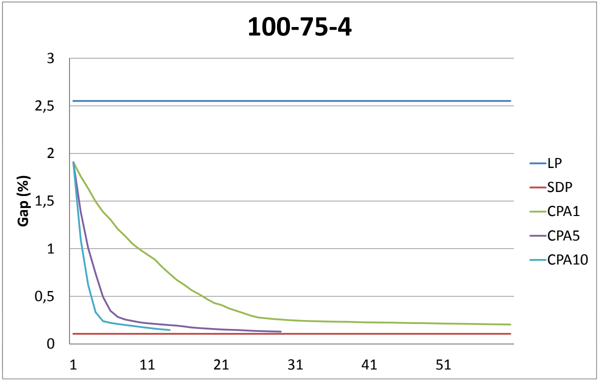

Figure : Bounds during the execution of the CPAs: 100-75-4

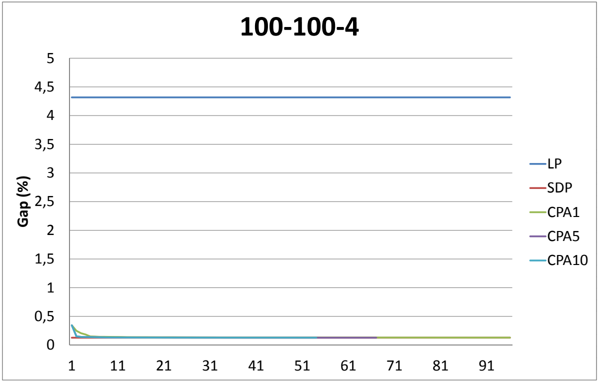

Figure : Bounds during the execution of the CPAs: 100-100-4

Preliminary Numerical experiments

- In these preliminary tests we allowed the CPAs to run for a longer time than SDP, in order to analyze their convergence behavio;
- CPAs find a better bound than LP for 13 out of 16 instances;
- CPAs obtain the same bound as SDP for 4 instances;

COPPE UFRJ

Conclusion

- 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 LP relaxation, and can be computed more efficiently than a stronger SDP relaxation, when the number of variables increases;

Thank you!

- **Marco Costa:** marcosil@cos.ufrj.br
- **Marcia Fampa:** fampa@cos.ufrj.br
- **Daniela Cristina Lubke:** danielalubke@cos.ufrj.br

COPPE UFRJ

Upper bounds for the binary quadratic knapsack problem

Marco Costa, Marcia Fampa and Daniela Cristina Lubke

Federal University of Rio de Janeiro
Systems Engineering and Computer Sciences Department, UFRJ-COPPE-PESC

SBPO 2015: XLVII Simpósio Brasileiro de Pesquisa Operacional
Porto de Galinhas, Pernambuco, 25-28 de agosto de 2015

COPPE
UFRJ