Constraint Satisfaction Problems Solved by Semidefinite Relaxations

MOHAMED ETTAOUIL and CHAKIR LOQMAN
UFR: Scientific calculation and Computing, Engineering sciences
Department of Mathematics and Computer science
Faculty of Science and Technology of Fez
Box 2202, University Sidi Mohammed Ben Abdellah Fez
MOROCCO
mohamedettaouil@yahoo.fr chakirfst@yahoo.fr

Abstract: We consider the constraint satisfaction problem (CSP), where the values must be assigned to variables which are subject to a set of constraints. This problem is naturally formulated as 0-1 quadratic knapsack problem subject to quadratic constraint. In this paper, we present a branch-and-bound algorithm for 0-1 quadratic programming, which is based on solving semidefinite relaxations. At each node of the enumeration tree, a lower bound is given naturally by the value of (SDP) problem and an upper bound is computed by satisfying the quadratic constraint. We show that this method is able to determine whether a (CSP) has a solution or not. Then we give some hints on how to reduce as much as possible the initial size of the (CSP). Some numerical examples assess the effectiveness of the theoretical results shown in this paper, and the advantage of the new modelization.

Key-Words: Constraint satisfaction problem, 0-1 Quadratic knapsack problem, SDP relaxation, Branch-and-bound, Filtering algorithms.

1 Introduction

Constraint Satisfaction Problems (CSP) has been recognized as efficient models for solving many combinatorial and complexes problems. For example, problems from timetabling, scheduling, resource allocation, planning, Airspace sectorization [9], etc...

This problem was introduced in 1974 by Montanari [22]. A (CSP) is stated as a triple (X, D, C), where:

- $= \{X_1, X_2, ..., X_N$ is a set of n variables.
- $D = \{D(X_1), ..., D(X_N)$ is the domain of each variable $_i \in X$
- $C = \{c_1, ..., c_m$ is a set of constraints.

The constraint satisfaction problems (CSP) are usually represented as graphs, where nodes correspond to variables and edges to constraints. As arc consistency is one of the basic properties of (CSP) [12], it guarantees that any value of the domain of a variable can be found in, at least, a support of any constraint. Many algorithms have been proposed to establish arc consistency such as AC_3 [21], [26], AC_7 [2]. Another method to solve this problem is to modelize it as 0-1 quadratic knapsack problem subject to quadratic constraint, and to use the lagrangian dual to solve the latter model [7], [8].

In this paper, our objective is to present another branch-and-bound algorithm for 0-1 quadratic programming (QK), based on solving semidefinite relaxation. At each node of the enumeration tree, a lower bound is given naturally by the value of (SDP) problem. An upper bound is computed by satisfying the quadratic constraint, using an exact algorithm for solving unconstrained quadratic 0-1 programming [4]. We show that this method is able to determine whether a (CSP) has a solution or not.

In section 3, we present a modelization of binary (CSP) as 0-1 quadratic knapsack problem (QK). In section 4, we introduce strong (SDP) relaxations for 0-1 quadratic knapsack problem (QK) [19]. In section 5 and 6, we present some theoretical results and algorithms for computing upper and lower bounds. The latter will be able to determine whether a (CSP) has a solution or not. Section 7 is devoted to give some hints on how to reduce as much as possible the initial size of the (CSP). Section 8 is a computation experiment.

2 Preliminaries

In this section, we collect several basic results about Positive Semidefinite matrices and semidefinite programming. Further results will be mentioned as needed. Most of the results on matrices quoted in this paper can be found in standard matrix theory books, such as [16], [18].

Let S_n denotes the space of symmetric $n \times n$ matrices, and S_n^+ the set of positive semidefinite matrices. We will use the notation $X \geq 0$ to express that X is positive semidefinite.

Given $A, B \in S_n$, we consider the (Frobenius) inner product $A \bullet B$ defined by:

$$A \bullet B = Tr(AB) = \sum_{i,j=1}^{n} A_{ij} B_{ij}$$

The quadratic form $x^T Ax$ can thus also be written as $A \bullet (xx^T)$. Semidefinite programs are linear programs over the cone of positive semidefinite matrices. They can be expressed in many equivalent forms, e.g.

$$(SDP)\quad \begin{cases} Max \quad \mathrm{A}_0 \bullet X \\ Subject\ to \\ \qquad A_i \bullet X = c_i \qquad i = 1, \ldots, m \\ \qquad X \geq 0 \end{cases}$$

$$(DSDP)\quad \begin{cases} Min \quad c^T y \\ Subject\ to \\ \qquad F(x) = \sum_{i=1}^{m} A_i y_i - A_0 \geq 0 \\ \qquad y \in IR^m \end{cases}$$

Where $c \in IR^m$, and $A_i \in S_n \quad \forall i \in \{0, \ldots\ldots, m\}$.

(DSDP) is the dual program of (SDP).

Lemma 2.1 (Rank-One Constraints) [14]

The set of $(X, x) \in S_n \times IR^n$ satisfying $X \geq xx^T$ is closed and convex. Actually

$$X - xx^T \geq 0 \ \ [resp. \succ 0] \Leftrightarrow \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \geq 0 \ \ [resp. \succ 0]$$

These preliminaries results can also be used to solve the 0-1 quadratic problem with quadratic constraint using semidefinite relaxations.

3 Modelization of the binary (CSP)

The constraint satisfaction problem (CSP) is stated as a triple (X, D, C) where:

- $X = \{X_1, X_2, \ldots\ldots, X_N\}$ is a set of n variables.
- $D = \{D(X_1), \ldots\ldots, D(X_N)\}$ is the domain of each variable $X_i \in X$
- $C = \{c_1, \ldots\ldots, c_m\}$ is a set of constraints.

In this modelization, we focus on the binary (CSP), i.e, the (CSP) with constraints of arity less than or equal to 2, and each constraint C_{ij} between the variables X_i and X_j is defined by its relation R_{ij}.

In the following, we want to present a new formulation of the binary (CSP).

For each variable X_i of the (CSP), we introduce d_i ($d_i = |D(X_i)|$) binary variables $x_{ik} \quad k = 1, \ldots, d_i$ such that:

$$x_{ik} = \begin{cases} 1 \ \ if \quad \mathrm{X}_i = v_k \quad v_k \in D(X_i) \\ 0 \qquad \text{Otherwise} \end{cases}$$

Since each variable X_i must be assigned to exactly one of the N domains, the following set of equations has to be satisfied:

$$\sum_{k=1}^{d_i} x_{ik} = 1 \quad \text{for all} \quad i = 1, \ldots\ldots, N \qquad (1)$$

The modelization process of the binary constraint satisfaction problem (CSP) to a 0-1 quadratic program subject to quadratic constraint leads to a complex formula with a lot of indices on variables, which can be seen very hard to understand. We prefer to describe this process by an example.

Example 3.1

We are giving a binary constraint satisfaction problem (CSP), with $X = \{X_1, X_2, X_3\}$ and $D = \{D(X_1), D(X_2), D(X_3)\}$ where $D(X_1) = \{v_1, v_2\}$, $D(X_2) = \{v_1, v_2, v_3\}$ and $D(X_3) = \{v_1, v_2\}$. Each constraint C_{ij} between the variables X_i and X_j is defined by its relation R_{ij} (R_{ij} is a subset of the cartesian product $D_i \times D_j$, specifying the compatible values between x_i and x_j).

$R_{12} = \{(v_1, v_2), (v_2, v_1), (v_2, v_3)\}$
$R_{13} = \{(v_1, v_1), (v_2, v_2)\}$
$R_{23} = \{(v_1, v_2), (v_2, v_1), (v_2, v_2), (v_3, v_2)\}$

If C_{ij} is a constraint between two variables X_i and X_j defined by its relation R_{ij} of the binary (CSP). For each couple (v_r, v_s) such that $\neg R_{ij}(v_r, v_s)$ we generate a constraint:

$$x_{ir} x_{js} = 0 \qquad (2)$$

These constraints can be aggregated in a single constraint:

$$f^{'}(x) = x_{11}x_{21} + x_{11}x_{23} + x_{12}x_{22} + x_{11}x_{32} + x_{12}x_{31} + x_{21}x_{31} + x_{23}x_{31} = 0 \quad (3)$$

The constraint (1) implies that:

$$\begin{cases} x_{11} = 1 - x_{12} \\ x_{21} = 1 - x_{22} - x_{23} \\ x_{31} = 1 - x_{32} \end{cases}$$

By substitution in the equation (3), we obtain:

$$f^{'}(x) = 2x_{12}x_{22} - 2x_{12}x_{32} + x_{22}x_{32} - 2x_{22} + 2 = 0$$

The constraints $x_{12} \le 1$, $x_{22} + x_{23} \le 1$ and $x_{32} \le 1$ can be rewritten as follows:

$$g(x) = x_{12}x_{32} + x_{22}x_{23} \le 1 \quad \text{and} \quad x_{22}x_{23} = 0$$

Thus, the (CSP) problem is equivalent to the following system:

$$\begin{cases} f(x) = f^{'}(x) + x_{22}x_{23} = 0 \\ \qquad g(x) \le 1 \\ \qquad x \in \{0,1\}^4 \end{cases} \qquad (4)$$

The (CSP) has a solution if and only if the system has a one.

Finally, we consider the following 0-1 quadratic program with quadratic constraint (QK):

$$(QK) \begin{cases} Min \quad f(x) \\ Subject\ to \\ \qquad g(x) \le 1 \\ \qquad x \in \{0,1\}^4 \end{cases}$$

Theorem 3.2

We consider the 0-1 Quadratic program (QK). The value V (QK) is equal to 0 if and only if the binary Constraint Satisfaction Problem (CSP) has a solution. With V(QK) is optimal value of the problem (QK).

Proof

By construction of the system, the value of the problem (QK) is nonnegative. The value V (QK) is equal to 0 if and only if there exists a solution of the system (4). Then, the CSP has a solution if and only if the value V (QK) is equal to 0. The theorem 3.2 is valid for any binary (CSP) and the equivalent system can be obtained easily following the above modelization.

Without loss of generality, as we observed above, we will consider in this paper the following problem:

$$(QK) \quad \begin{cases} Min \quad q_0 + \sum_{i=1}^{n} q_i x_i + \sum_{i,j=1\ i \prec j}^{n} q_{ij} x_i x_j \\ Subject\ to \\ \qquad \sum_{i=1}^{n} a_i x_i + \sum_{i,j=1\ i \prec j}^{n} a_{ij} x_i x_j \le b \\ \qquad x_i \in \{0,1\} \quad \text{for all} \quad i = 1,...,n \end{cases}$$

Where q_0 , q_i , q_{ij} , a_i and a_{ij} are integers and n the number of binary variables of (QK) problem
$(n \le Nd$ where $d = \max\{d_i,\ i = 1,..., N\})$ and
$0 \prec b \prec \sum_{i=1}^{n} a_i + \sum_{i,j=1\ ,i \prec J}^{n} a_{ij}$, $a_{ij} \ge 0 \ \ (1 \le i, j \le n)$

Based on this formulation we will develop a branch-and-bound method for 0-1 quadratic programming (QK), which is based on solving semidefinite relaxations.

4 Semidefinite Relaxation of (QK)

The SDP relaxations can be used to solve the generalized following 0-1 quadratic Knapsack problem with quadratic constraint (QK):

$$(QK)\quad \begin{cases} Min \quad q_0 + \sum_{i=1}^{n} q_i x_i + \sum_{i,j=1\, i\prec j}^{n} q_{ij} x_i x_j \\ Subject\ to \\ \quad \sum_{i=1}^{n} a_i x_i + \sum_{i,j=1\, i\prec j}^{n} a_{ij} x_i x_j \le b \\ \quad x_i \in \{0,1\} \text{ for all } i = 1,\dots,n \end{cases}$$

The above problem can be written as the following form:

$$(QK)\quad \begin{cases} Min \quad x^T Q x + q^T x + q_0 \\ Subject\ to \\ \quad x^T A x + a^T x \le b \\ \quad x \in \{0,1\}^n \end{cases}$$

Where Q and A are $n\times n$ real matrix, and $q \in IR^n$, $a \in IR^n$, $b \in IR$. Without loss of generality, we can suppose that Q and A are symmetric. If this is not the case, Q can be converted to symmetric form $(Q+Q^T)/2$.

The following constraint $x_i \in \{0,1\}$ can be written in the following form:

$$x_i \in \{0,1\},\ i=1,\dots,n \Leftrightarrow x_i^2 - x_i = 0,\ i=1,\dots,n$$
$$\Leftrightarrow diag(xx^T) - x = 0$$

Setting $X = xx^T$ can therefore be written as

$$diag(X) - x = 0$$
$$X = xx^T$$

We formulate this problem (QK) using an additional variable $X = xx^T$:

$$(QK)\quad \begin{cases} Min \quad x^T Q x + q^T x + q_0 \\ Subject\ to \\ \quad x^T A x + a^T x \le b \\ \quad X = xx^T \text{ and } diag(X) - x = 0 \end{cases}$$

A natural method to obtain a semidefinite relaxation of (QK) is to relax the last constraint $X = xx^T$ to $X \ge xx^T$, which is now convex with respect to (x, X) (lemma 2.1).

$$X \ge xx^T \quad \Leftrightarrow \quad \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \ge 0$$

Then, we obtain:

$$(SDP_{\{0,1\}})\quad \begin{cases} Min \quad Q\bullet X + q^T x + q_0 \\ Subject\ to \\ \quad A\bullet X + a^T x \le b \\ \quad \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \ge 0 \text{ and } d(X) = x \\ \quad (X, x) \in S_n \times IR^n \end{cases}$$

This relaxation $(SDP_{\{0,1\}})$ is equivalent to another relaxation $(SDP_{\{-1,1\}})$ [23]:

$$(SDP_{\{-1,1\}})\quad \begin{cases} Min \quad W^T \begin{pmatrix} q_0 & \frac{1}{2}q^T \\ \frac{1}{2}q & Q \end{pmatrix} W \bullet Y \\ Subject\ to \\ \quad W^T \begin{pmatrix} 0 & \frac{1}{2}a^T \\ \frac{1}{2}a & A \end{pmatrix} W \bullet Y \le b \\ \quad Y \ge 0 \quad diag(Y) = e_{n+1} \quad Y \in S_{n+1} \end{cases}$$

With $W = \begin{pmatrix} 1 & 0 \\ \frac{1}{2}e & \frac{1}{2}I_n \end{pmatrix}$ and $e = (1,\dots\dots,1)^T$

Proposition 4.1 [20]
The dual problem of (QK) is equivalent to (SDP) problem with $(\lambda,u)\in IR^{+}\times IR^{n}$ and $r\in IR$:

$$(DL_{(QK)})\quad\begin{cases} Max \quad r \\ Subject\ to \\ \begin{pmatrix} q_0(\lambda)-r & \frac{1}{2}q(\lambda,u)^T \\ \frac{1}{2}q(\lambda,u) & Q(\lambda,u)\end{pmatrix}\geq 0 \end{cases}$$

Where $Q(\lambda,u)=Q+\lambda A+D(u)$,
$q(\lambda,u)=q+\lambda a-u$, $q_0(\lambda)=q_0-\lambda b$
Here $D(u)$ denotes the diagonal matrix constructed from the vector u.

Note that the dual problem of the problem $(DL_{(QK)})$ (i.e., the bidual of problem (QK)) is given by [19]:

$$(SDP_{\{0,1\}})\quad\begin{cases} Min \quad x^TQx+q^Tx+q_0 \\ Subject\ to \\ x^TAx+a^Tx\leq b \\ \begin{pmatrix}1 & x^T\\ x & X\end{pmatrix}\geq 0 \text{ and } diag(X)=x \\ (X,x)\in S_n\times IR^n \end{cases}$$

5 computation of a lower bound

In this section, we present a method to compute a lower bound, using Semidefinite Relaxations. In addition to some of the classical results [10][11], we also present a few either very recent or less well known results. In particular, we describe the relationship between the (CSP) and SDP relaxations for 0-1 Quadratic Knapsack problem (QK). Therefore, we show that the notion of SDP with some complementary assumptions can detect whether a (CSP) has a solution or not.

Proposition 5.1
Let $V(DL_{(QK)})$ be an optimal value of the problem $(DL_{(QK)})$ and $V(SDP_{\{0,1\}})$ an optimal value of the problem $(SDP_{\{0,1\}})$.
Then
$$V(QK)\geq V(SDP_{\{0,1\}})\geq V(DL_{(QK)})$$

Proof

- The last inequality is weak duality between $(DL_{(QK)})$ and its dual $(SDP_{\{0,1\}})$
  Then $V(SDP_{\{0,1\}})\geq V(DL_{(QK)})$
- For the first inequality, take x feasible in (QK). Then $(x, X=xx^T)$ is feasible in $(SDP_{\{0,1\}})$ and has the same objective value. Thus, the feasible domain in $(SDP_{\{0,1\}})$ is larger than in (QK).
  Then $V(QK)\geq V(SDP_{\{0,1\}})$
  Finally $V(QK)\geq V(SDP_{\{0,1\}})\geq V(DL_{(QK)})$

Theorem 5.2
Let $V(DL_{(QK)})$ be an optimal value of the problem $(DL_{(QK)})$ and $V(SDP_{\{0,1\}})$ an optimal value of the problem $(SDP_{\{0,1\}})$.
If $V(DL_{(QK)})\succ 0$ or $V(SDP_{\{0,1\}})\succ 0$ then the (CSP) problem has no solution.

Proof
Just apply proposition 5.1
We have $V(QK)\geq V(SDP_{\{0,1\}})\geq V(DL_{(QK)})$
If $V(DL_{(QK)})\succ 0$ or $V(SDP_{\{0,1\}})\succ 0$
Then $V(DL_{(QK)})\succ 0$
The theorem3.1 implies that the (CSP) has no solution.

Theorem 5.3
Let $(\overline{X},\overline{x})$ be an optimal solution of the $(SDP_{\{0,1\}})$.
If

1. $\overline{X}=\overline{x}\overline{x}^T$
2. $V_{SDP_{\{0,1\}}}(\overline{X},\overline{x})=0$

Then the (CSP) problem has a solution.

Proof

If $(\overline{X},\overline{x})$ is a solution of the problem $(SDP_{\{0,1\}})$ and $\overline{X}=\overline{x}\,\overline{x}^T$. Then $\overline{x}$ is a feasible solution in (QK) problem.

Take any y feasible in (QK) then (y, yy^T) is feasible in $(SDP_{\{0,1\}})$.

Because $(\overline{X},\overline{x})$ is a solution of the $(SDP_{\{0,1\}})$, we can write:

$$Q\bullet \overline{x}\,\overline{x}^T+q^T\overline{x}+q_0=Q\bullet \overline{X}+q^T\overline{x}+q_0\le Q\bullet yy^T+q^Ty+q_0$$

Then $\overline{x}$ is a solution of the problem (QK) and $V_{(QK)}(\overline{x})=V_{(SDP_{\{0,1\}})}(\overline{X},\overline{x})=0$.

The theorem 3.2 implies that (CSP) has a solution.

Algorithm 1

This algorithm computes a lower bound for the 0-1 Quadratic Knapsack problem (QK):

begin

Solve the semidefinite program $(SDP_{\{0,1\}})$.

Let $(\overline{X},\overline{x})$ be an optimal solution of the $(SDP_{\{0,1\}})$.

If $V_{SDP_{\{0,1\}}}(\overline{X},\overline{x})\succ 0$

Then the (CSP) problem has no solution

Else

If ($V_{SDP_{\{0,1\}}}(\overline{X},\overline{x})=0$ and $\overline{X}=\overline{x}\,\overline{x}^T$)

Then $\overline{x}$ is an optimal solution of (QK)

Else

$V_{SDP_{\{0,1\}}}(\overline{X},\overline{x})$ is a lower bound of (QK)

End if

End if

End

Example 5.4

To illustrate how these results can be understood, let us consider the instance of example 3.1:

$$(QK)\begin{cases} Min \quad f(x)=2x_{12}x_{22}-2x_{12}x_{32}+x_{22}x_{32}+x_{22}x_{23}-2x_{22}+2 \\ Subject\ to \\ \qquad g(x)=x_{12}x_{32}+x_{22}x_{23}\ \le 1 \\ \qquad x\in\{0,1\}^4 \end{cases}$$

Its SDP relaxation is:

$$\begin{cases} Min \quad \begin{pmatrix} 2 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ -1 & 1 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0.5 & 0 & 0 \\ 0 & -1 & 0.5 & 0 & 0 \end{pmatrix}\bullet Y \\ Subject\ to \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 \end{pmatrix}\bullet Y\le 1\ , \quad \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}\bullet Y=1 \\ X_{ii}=x_i \qquad i=1,\dots,4 \\ Y\ge 0 \end{cases}$$

Where:

$$Y=\begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix}\ ,\ x=\left[x_{12},x_{22},x_{23},x_{32}\right]^T \text{ and } X\in S_4$$

The lower bound of (QK) is the optimal value $V(SDP_{\{0,1\}})=-0.61\le 0$. Recall that, in this example, the optimal value $V(QK)=0$.

6 computation of an upper bound

In this section, our main objective is to compute an upper bound. One of the methods that are used to compute this latter is to satisfy the quadratic constraint, i.e. solving the unconstrained quadratic 0-1 programming:

$$(P)\ \begin{cases} Min \quad g(x)=x^TAx+a^Tx \\ Subject\ to \\ \qquad x\in\{0,1\}^n \end{cases}$$

Various approaches have been used to solve the unconstrained quadratic 0-1 programming (P). Two recent overviews of these approaches are presented in [13] and [15]. One of the possible techniques introduced by Billionnet and Elloumi [4] is to convexify the objective function and then use a Mixed-Integer Quadratic Programming (MIQP) solver for solving the problem (P). Their algorithm works in detail as follows:

For any vector $u \in IR^n$, let us define the perturbed function $g_u(x)$ in the following way

$$g_u(x) = x^T(A - D(u))x + (a+u)^T x$$

Here $D(u)$ denotes the diagonal matrix constructed from the vector u.
It is easy to see, that an equivalent problem to (P) is

$$(P_u)\quad \begin{cases} Min \quad g_u(x) \\ Subject\ to \\ \quad x \in \{0,1\}^n \end{cases}$$

Relaxing the integrality constraint in problem (P_u) gives the lower bound $\beta(u)$ on (P):

$$\begin{cases} \beta(u) = Min \quad g_u(x) \\ Subject\ to \\ \quad x \in [0,1]^n \end{cases}$$

If the vector u is chosen, such that $A - D(u) \geq 0$, $\beta(u)$ is obtained by solving a convex quadratic problem, which can be done efficiently.
Now, if u^* is the maximize of $\beta(u)$, the optimal lower bound β^* will be obtained, i.e.

$$\begin{cases} \beta^* = \beta(u^*) = \quad Max \quad \beta(u) \\ \quad Subject\ to \\ \quad u \in IR^n \\ \quad A - D(u) \geq 0 \end{cases}$$

Proposition 6.1[4]
The bound $\beta(u^*)$ is equal to the value of the semidefinite program:

$$(SDP)\quad \begin{cases} Max \quad r \\ Subject\ to \\ \begin{pmatrix} -r & \frac{1}{2}(a+u)^T \\ \frac{1}{2}(a+u) & A - D(u) \end{pmatrix} \geq 0 \end{cases}$$

And is also equal to the optimal value of its dual

$$(DSDP)\quad \begin{cases} Min \quad A \bullet X + a^T x \\ Subject\ to \\ \quad X_{ii} = x_i \\ \quad \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \geq 0 \\ \quad x \in IR^n \quad X \in IR^{n \times n} \end{cases}$$

Moreover, if (r^*, u^*) is an optimal solution of (SDP), then $A - D(u) \geq 0$ holds and $\beta(u^*) = \beta^*$.

Proposition 6.2
Let y be an optimal solution of (P). If $g(y) \succ b$ then the set of feasible solutions is empty.

Proof
Let y be an optimal solution of problem (P).
We have

$$g(y) = Min \ \left\{ g(x) / x \in \{0,1\}^n \right\} \leq g(x) \quad \forall x \in \{0,1\}^n$$

If $g(y) \succ b$
Then $g(x) \succ b \quad \forall x \in \{0,1\}^n$
Finally, the set of feasible solutions is empty.

Proposition 6.3
If $\beta(u^*) \succ b$ then the (CSP) has no solution.

Proof
$\beta(u^*)$ is the optimal lower bound for (P) then $V(P) \geq \beta(u^*)$
If $\beta(u^*) \succ b$ then $V(P) \succ b$
The proposition 6.2 implies that the (CSP) has no solution.

Algorithm 2

The present algorithm computes an upper bound for the 0-1 Quadratic Knapsack problem (QK):

Begin

Solve the semidefinite program (SDP).

Let (r^*, u^*) be an optimal solution of (SDP)

If $\beta(u^*) \succ b$

Then the (CSP) problem has no solution

Else

Solve the Quadratic problem (P_{u^*}) whose Continuous relaxation is convex.

End if

Let y be the optimal solution of problem (P)

If $g(y) \succ b$

Then (CSP) has no solution

Else

y is feasible for the problem (QK)

End if

End

Example 6.4

To illustrate how these results can be understood, let us consider the instance of the quadratic constraint exists in example 3.1:

$$\begin{cases} Min \quad g(x) = x_{12}x_{32} + x_{22}x_{23} \\ Subject\ to \\ \qquad x \in \{0,1\}^4 \end{cases}$$

Its (SDP) relaxation is:

$$\begin{cases} Min \quad \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 \end{pmatrix} \bullet \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \\ Subject\ to \\ \qquad X_{ii} = x_i \quad i = 1,...,4 \\ \qquad \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \geq 0 \ \ and \ \ (X, x) \in S_4 \times IR^4 \end{cases}$$

Where $x = [x_{12}, x_{22}, x_{23}, x_{32}]$

Proposition 6.1 gives practical method for computing the best u^* and the lower bound $\beta(u^*)$, based on the resolution of a semidefinite program [5]. The optimal vector can be obtained by solving the (SDP) is $u^* = [-0.5, -0.5, -0.5, -0.5]$ and the corresponding solution is:

$$\begin{pmatrix} 1 & 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.0625 & 0.0625 & -0.125 \\ 0.25 & 0.0625 & 0.25 & -0.125 & 0.0625 \\ 0.25 & 0.0625 & -0.125 & 0.25 & 0.0625 \\ 0.25 & -0.125 & 0.0625 & 0.0625 & 0.25 \end{pmatrix}$$

The lower bound is $\beta(u^*) = -0.25$. We use a Mixed-Integer Quadratic Programming (MIQP) solver for solving the problem (P_{u^*}).

Then, we obtain an exact solution for the unconstrained quadratic 0-1 programming:

$$[x_{12}, x_{22}, x_{23}, x_{32}] = [1, 0, 1, 0]$$

The value of the upper bound is equal to zero $V(P) = 0$

7 Reduction of the size of the problem (CSP)

The main idea of this filter is to try to reduce as much as possible the initial size of the (CSP)[1]. Arc consistency algorithms are widely used to prune the search space of (CSP). Many arc consistency algorithms have been proposed. On the one side, there are heavyweight arc consistency algorithms such as AC-2001[3], AC-3.1[26] and AC-7[2] that use additional data structures to avoid repeating their support checks. All these algorithms have optimal worst case time complexity of $\varphi(ed^2)$, where e is the number of constraints and d is the maximum domain size of the variables $(d = \max\{d_i,\ i = 1,..., N\})$. On the other side there are lightweight arc consistency algorithms such as AC-3[21], AC-3_d [24], and AC-3_p [25] which do not use additional data structures. These algorithms repeat their support checks and have non-optimal bound of $\varphi(ed^3)$ for their worst case time complexity. However, despite the fact that these algorithms do not have an optimal worst case time complexity, experimental evaluation of these algorithms has demonstrated that they are efficient on average [25].

8 Computational experiments

In order to assess the effectiveness of the theoretical results shown in this paper, and the advantage of the new modelization, the preliminary numerical experiments were performed on randomly generated test (CSP) problems (with a relatively small number of variables, containing 5 to 20 variables). The computing of a solution consists of 2 phases. During the first phase we apply the algorithm AC-3. Then, some binary variables will be assigned to 0 or 1. The values are reported on the 0-1 quadratic knapsack problem (QK). In the second phase, the obtained optimization problem was also solved with semidefinite relaxations, convex quadratic programming for the unconstrained 0-1 quadratic problems and branch-and-bound procedure. We choose to solve semidefinite programs using CSDP [5] software, applying the Interior-Point Method.

We consider randomly generated instances of the 0-1 quadratic knapsack problem; the absolute values of the coefficients q_i and q_{ij} of the objective function are integers distributed between 0 and 20, and the coefficients a_i and a_{ij} of the quadratic constraint are integers distributed between 0 and 20, while b is an integer between 1 and $\sum_{i=1}^{n} a_i + \sum_{i,j=1\ i\prec j}^{n} a_{ij}$.

Our SDP relaxation is indeed very efficient for solving the 0-1 quadratic knapsack problem up to n=60 (number of binary variables). It is important to note that the objective of these simulations is to assess the effectiveness of the theoretical results and not to compare our results with those obtained using other exact methods. The preliminary results that have been obtained suggest that the proposed algorithm is promising as an efficient method for solving the (CSP) problem.

To illustrate how these results can be understood, let us consider the instance of example 3.1:

We use a branch-and-bound algorithm for 0-1 quadratic programming, in which a lower bound is $V(SDP_{\{0,1\}}) = -0.61$ and an upper bound is $V(P) = 0$

We obtain an exact solution for the 0-1 quadratic knapsack problem (QK):

$$x = [x_{12}, x_{22}, x_{23}, x_{32}] = [1, 0, 0, 1]$$

By substitution in the system:

$$\begin{cases} x_{11} = 1 - x_{12} \\ x_{21} = 1 - x_{22} - x_{23} \\ x_{31} = 1 - x_{32} \end{cases}$$

We obtain $x_{11} = 0$, $x_{21} = 1$ and $x_{31} = 0$

Then $[x_{11}, x_{12}, x_{21}, x_{22}, x_{23}, x_{31}, x_{32}] = [0,1,1,0,0,0,1]$

We apply

$$x_{ik} = \begin{cases} 1 \ \ if \quad \mathrm{X}_i = v_k \quad v_k \in D(X_i) \\ 0 \qquad \text{Otherwise} \end{cases}$$

Then we have a solution of problem (CSP)

$$X_1 = v_2 \ , \ X_2 = v_1 \text{ and } X_3 = v_2$$

Experiments results are in progress for searching adequate values of parameters such as threshold value of the Branch-and-Bound algorithm, the parameters of genetic algorithm (GA) [17].

Summary and conclusions

In this paper, we discuss the use of various SDP relaxations to find a solution of (CSP) problem. This problem has been presented as a 0-1 quadratic knapsack problem subject to quadratic constraint. To solve this problem, we propose a branch-and-bound method. At each node of the enumeration tree, a lower bound is given naturally by the value of (SDP) problem and an upper bound is computed by satisfying the quadratic constraint. Then a solution or failure may be detected prematurely. Some numerical examples assess the effectiveness of the theoretical results shown in this paper, and the advantage of the new modelization.

Several directions can be investigated to try to improve this method: use a more efficient metaheuristics, for example genetic algorithm (GA) [17], determines the best frequency to simplify the problem when some variables are fixed.

To make this approach more efficient, it can be combined with other metaheuristics or it can be computationally optimized by introducing analytical improvements, such as replacing the quadratic constraint in the 0-1 quadratic knapsack problem (QK) by linear constraints. Moreover, the approach introduced in this paper is also intrinsically easy to parallelize.

This (CSP) problem has been presented as an optimization problem with quadratic constraint, in this way; new combinatorial optimization problem

can be solved via the neural networks approaches [6].
Finally, others studies are in progress to applied this approach to many problems such as Airspace sectorization, Aircraft conflict [9], etc…

ACKNOWLEDGEMENTS
The authors would like to express their sincere thanks for the referee for his/her helpful suggestions.

References:

[1] M. Azlinach, Y. Marina, M.A. Iaza, M. Sofianita, R.A. Shuzilina, Constraint Satisfaction Problem Using Modified Branch and Bound, *WSEAS Transactions on Computers*, 2008, 1(7), pp. 1-7.

[2] C. Bassière, E.C. Freuder, and J. Regin, Using Constraint metaknowledge to reduce arc consistency computation, *Artificial Intelligence* , 1999, 107, pp. 125 -148.

[3] C. Bessière and J-C . Régin, Refining the basic constraint propagation algorithm, *In proceedings of the Seventeeth International Joint conference on Artificial Intelligence (IJCAI'2001)*, 2001, pp. 309-315.

[4] A. Billionnet and S. Elloumi, Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem, *Math. Programming*, 2007, 109, pp. 55-68.

[5] S. Borchers, CSDP software, *https: //projects.coin-or.ord/Csdp/*, 2006.

[6] O. Diego, D. Carlos, A. Bernardino , M. Minia, A framework for the definition and generation of artificial neural networks, *Proceedings of the 6th WSEAS International Conference on Applied Computer Science*, 2006, pp. 280-284.

[7] M. Ettaouil, A 0-1 quadratic knapsack problem for modelizing and solving the constraint satisfaction problem, *In Lecture Notes in Artificial Intelligence N°1323,Springer Verlag, Berlin*, 1997, pp. 61-72.

[8] M. Ettaouil, *contribution à l'étude des problèmes de satisfaction de contraintes et à la programmation quadratiques en nombres entiers, allocation statique de tâche dans les systèmes distribués.* thèse d'état, Université Sidi Mohammed ben Abdellah, F.S.T. de Fès, 1999.

[9] M. Ettaouil and C. Loqman, sectorisation de l'espace aérien et réseaux de contraintes, *premier congres international de la société marocaine de mathématiques appliquées*, 2008, pp. 150-152.

[10] A. Faye and F. Roupin, Partial Lagrangian relaxation for general quadratic programming, *4OR*, 2007, 5, pp. 75-88.

[11] M. Fernanda, P. Costa, Edite M.G.P. Fernandes, Implementation of an Interior Point Multidimensional Filter Line Search Method for Constrained Optimization, *Proceedings of the 5th WSEAS Int. Conf. on System Science and Simulation in Engineering*, 2006, pp. 391-396.

[12] S. Frantisek, Relation between UML2 Activity Diagrams and CSP algebra, *WSEAS Transactions on Computers*, 2005, 10(4), pp. 1223-1233.

[13] P. Hansen, B. Jaumard and C.Meyer, *A simple enumerative algorithm for unconstrainted 0-1 quadratic programming*, Technical Report G-2000-59, Les Cahiers du GERAD, 2000.

[14] C. Helmberg, Semidefinite programming, *European Journal of Opertional Research*, 2002, 37, pp. 461-482.

[15] C. Helmberg and F. Rendl, Solving quadratic (0-1) problem by semidefinite programs and cutting planes, *Math. Programming*, 1998, 82, pp. 291-315.

[16] R. Horn and C. Johnson, *Matrix Analysis*, Combridge University Press, 1985, pp. 231-279.

[17] B.J. Huang, J. Zhuang, D .H. Yu, A Novel and Accelerated Genetic Algorithm, *7th WSEAS Int. Conf. on APPLIED COMPUTER & APPLIED COMPUTATIONAL SCIENCE (ACACOS '08)*, 2008, pp. 245-253.

[18] P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, Academic Press, Orlando, FL, 1985.

[19] C. Lemaréchal and F. Oustry, *Semidefinite relaxations and lagrangian duality with application to combinatorial optimization*, Rapport de recherche N° 3710 disponible sur http://www.inria.fr, 1999.

[20] C. Lemaréchal and F. Oustry, *Semidefinite relaxations in combinatorial optimization from a lagrangian point of view*, In Advances in Convex Analysis and Global Optimization. N. Hadjisavvas and P.M. Pardalos, Kluwer, 2001.

[21] A.K. Marckworth, consistency in networks of relations, *Artificial Intelligence*, 1977, 8, pp. 118-126.

[22] U. Montanari, Networks of constraints: Fundamental proprieties and applications to picture processing, *Information sciences*, Vol. 7, N.2, 1974, pp. 95-132.

[23] F. Roupin, *Algorithmes Combinatoires et Relaxations par Programmation Linéaire et Semidéfinie. Application à la Résolution de Problèmes Quadratiques et d'Optimisation dans les Graphes*, Thèse CEDRIC, Habilitation à Diriger des Recherches en Informatique de l'Université Paris Nord, 2006.

[24] M.R.C. Van Dongen, *AC-3_d an efficient arc-consistency algorithm with a low space-complexity*, In P. Van Hentenryck, editor, Proceedings of the Eighth International Conference on Principles and Practice of Constraint Programming, volume 2470 of Lecture notes in Computer Science, Springer, 2002, pp. 755-760.

[25] M.R.C. Van Dongen, Saving support-checks does not always save time, *Artificial Intelligence Review*, 2004, Accepted for publication.

[26] Y. Zhang and R.H.C. Yap, Making AC-3 an optimal algorithm, in proceedings of the Seventeeth International Joint conference on Artificial Intelligence (IJCAI'2001), 2001, pp. 316-321.