Journal of Theoretical and Applied Information Technology
10th March 2013. Vol. 49 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

CONVEX QUADRATIC REFORMULATIONS FOR SOLVING DAYS-OFF SCHEDULING PROBLEM

[1]CHAKIR LOQMAN, [2]MOHAMED ETTAOUIL, [3]YOUSSEF HAMI, [4]KHALID HADDOUCH

[1]Department of Computer Engineering, High School of technology,
Moulay Ismaïl University, B. P. 3103, 50000, Toulal, Meknes, MOROCCO.
[2, 3,4] UFR: Scientific Computing and Computer sciences, Engineering sciences, Modeling and Scientific Computing Laboratory, Faculty of Science and Technology. University Sidi Mohamed Ben Abdellah
Box 2202, Fez, MOROCCO.

E-mail:[1]chakirfst@yahoo.fr, [2]mohamedettaouil@yahoo.fr,[3]y.hami@hotmail.com,
[4]haddouchk@yahoo.fr

ABSTRACT

Workforce scheduling is one of the most important and practical problems in the service of industry and continuous manufacturing settings. Solving this problem requires determining how many workers must be assigned to each of the planning periods of work time for an organization. In this paper, our main objective is to solve the days-off scheduling problem with day task constraints. This problem is naturally formulated as 0-1 quadratic programming subject to linear constraints. To solve the latter problem, we formulate it as an equivalent 0-1 quadratic programming with a convex objective function using two convexification techniques, the first one is based on the smallest eigenvalue and the second uses the semidefinite relaxation. Some numerical examples and computational experiments assess the effectiveness of the theoretical results shown in this paper.

Keywords: *Workforce Scheduling, Days-Off Scheduling, Convex Quadratic Reformulation, Semidefinite Programming, Quadratic Programming.*

1. INTRODUCTION

Workforce scheduling is both an important and a common problem to all organizations, especially for organizations that operate seven days a week or 24 hours a day. The problem is to determine how many workers must be assigned to each of the planning periods of work time for an organization.

In the literature, the workforce scheduling problems are traditionally classified into three categories [1]-[2]: Shift scheduling determines each employee's work and break hours per day. Days-off scheduling determines each employee's work-days and off days per week or multiple-week work cycle. Tour scheduling combines the shift and days-off scheduling problems by determining each employee's daily work hours and weekly workdays.

In this paper, we consider the days-off scheduling problem which tries to give priority to the worker and the company. It allows them to organize the spare time while maximizing the number of consecutive off days and minimizing the costs of transport. This problem includes the following constraints:

1. For any day, the number of workers is determined
2. Each worker has a fixed number of days off per week
3. Allow to specify the classes of necessary workers for the tasks during a given period.

In this paper, our main objective is to present two methods to solve the days-off scheduling problem through presenting a model in the form of a 0-1 quadratic program. The latter problem consists in minimizing a quadratic function subject to linear constraints (QP). A 0-1 quadratic program is often reformulated before searching for its optimal solution because the objective function is not convex. Therefore, many approaches have been proposed to solve (QP) through 0-1 linear reformulations [3] or 0-1 convex quadratic reformulations [4]-[5]. The most well known approaches are the 0-1 linear reformulations, however; other methods have been proposed, for example, enumerative methods based on different types of relaxations such as, Lagrangian relaxation,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

semidefinite relaxation or convex quadratic relaxation [6]- [7]- [8]- [9] -[10] -[11]- [12]- [13].

This paper is organized as follows: In section 2, we introduce a description of days-off scheduling problem and we provide a formulation of this problem as 0-1 quadratic programming (QP). In section 3, we apply two methods to solve (QP), the first one is based on the smallest eigenvalue method and the second uses the semidefinite programming. Then, we turn our attention to present some theoretical results that are able to convexify the problem (QP). Section 4 is devoted to giving some hints on how to generate the satisfiable instances of the days-off scheduling problem in a random way and presenting some experimental results.

2. DESCRIBING AND MODELING THE DAYS-OFF SCHEDULING PROBLEM

The days-off scheduling problem can be formulated in several ways. The mathematical model discussed in this section is concerned with finding the solution that maximizes the number of consecutive off days for employees. We will first introduce the notation that we use throughout the paper, and we give detailed formulation of this problem as 0-1 quadratic programming. This formulation is simple and general for any companies.

2.1 DESCRIPTION OF PROBLEM

The days-off scheduling problem determines each employee’s work days and off days per week or multiple-week work cycle. This paper considers a specific type of days-off scheduling in which the off days are assigned to the workers in order to have a maximum of consecutive off days. This problem is defined by [14]:

- p is the number of workers.
- T_i is the number of days-off for worker i in the week, with $i \in \{1,.., p\}$.
- n_k is the number of workers needed on day k, with $k \in \{1,..,7\}$.

Maximizing the number of consecutive off days for employees in these conditions does not depict real life. In this paper, we introduce a new class of constraints in order to satisfy daily labor demands such that a specific task should be assigned to a class of workers who have skills for it. These constraints are defined by a set $C = \{c_1, c_2, ..., c_7\}$, where c_k is the class of q_k workers needed to realize a specific task for day k, with $q_k \in \{1,..., n_k\}$ and $k \in \{1,..,7\}$.

2.2 PROBLEM FORMULATION

In this section, we present a model of days-off scheduling problem in terms of 0-1 quadratic program with linear constraints.

For each worker $i \in \{1,.., p\}$, we introduce 7 binary variables x_{ik} for $k \in \{1,..,7\}$, such that:

$$x_{ik} = \begin{cases} 1 & \textit{If day } k \textit{ is an off day for worker } i \\ 0 & \textit{Otherwise} \end{cases}$$

This matrix is converted to a n-vector, where $n = 7p$:

$$x = \left(x_{11}, x_{12}, ..., x_{17}, x_{21}, x_{22}, ..., x_{27}, ..., x_{p1}, x_{p2}, ..., x_{p7}\right)^T$$

The main objective is to maximize the number of consecutive off days in the week. Then, we can define the objective function $g(x)$ in the following way:

$$g(x) = \sum_{i=1}^{p} \left(\sum_{k=1}^{6} x_{ik} x_{ik+1} + x_{i7} x_{i1} \right) = x^T Q' x$$

Where $Q' \in IR^{n \times n}$ and $Q_0 \in IR^{7 \times 7}$ are the symmetric matrices:

$$Q' = \begin{pmatrix} Q_0 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & Q_0 \end{pmatrix}, \quad Q_0 = \begin{pmatrix} 0 & 0.5 & 0 & 0 & 0 & 0 & 0.5 \\ 0.5 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0 & 0 & 0 & 0.5 & 0 \end{pmatrix}$$

The linear constraints associated with this problem are defined by:

• For each day k, the number of workers who take this day off is $p - n_k$. These constraints are defined by:

$$\sum_{i=1}^{p} x_{ik} = p - n_k, \; k \in \{1,..,7\} \quad \Leftrightarrow \quad A_0 x = b'$$

The matrix $A_0 \in IR^{7 \times n}$ and the vector $b' \in IR^7$ of the linear constraints are:

$$A_0=\begin{pmatrix}1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & \cdots & \cdots & 1 & 0 & \cdots & 0\\ 0 & \ddots & \ddots & \vdots & 0 & \ddots & \ddots & \vdots & \cdots & \cdots & 0 & \ddots & \ddots & \vdots\\ \vdots & \ddots & \ddots & 0 & \vdots & \ddots & \ddots & 0 & \cdots & \cdots & \vdots & \ddots & \ddots & 0\\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 1 & \cdots & \cdots & 0 & \cdots & 0 & 1\end{pmatrix}, b'=\begin{pmatrix}p-n_1\\ p-n_2\\ \vdots\\ p-n_7\end{pmatrix}$$

• Each worker has T_i off-days in the week:

$$\sum_{k=1}^{7} x_{ik} = T_i,\ i\in\{1,.., p\} \quad\Leftrightarrow\quad A_1x = T$$

The matrix $A_1 \in IR^{p\times n}$ and the vector $T\in IR^p$ of the linear constraints are:

$$A_1=\begin{pmatrix}1 & \dots & 1 & \dots & 1 & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0 & \dots & 0\\ \vdots & & \vdots & & \vdots & \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots\\ 0 & \dots & 0 & \dots & 0 & 1 & \dots & 1 & \dots & 1 & \dots & 0 & \dots & 0 & \dots & 0\\ \vdots & & \vdots & & \vdots & \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots\\ 0 & \dots & 0 & \dots & 0 & 0 & \dots & 0 & \dots & 0 & \dots & 1 & \dots & 1 & \dots & 1\end{pmatrix}, T=\begin{pmatrix}T_1\\ T_2\\ \vdots\\ \vdots\\ T_p\end{pmatrix}$$

Maximizing the number of consecutive off days for workers in these conditions does not depict real life because some tasks for each day of the week can have unique workers requirement. In this way, we solve days-off scheduling problem with the constraints defined by day task. Every day task should be assigned to a class of workers who have skills for it. In order to group some workers into active consecutive days on the week:

• The constraints defined by day task are a set $C=\{c_1,c_2,...,c_7\}$, where c_k is the class of $q_k\in\{1,...,n_k\}$ workers needed to realize a specific task for day $k\in\{1,..,7\}$. These classes are determined by the company and represented by the matrix $A_2\in\{0,1\}^{7\times n}$ such that:

$$a_{ik}=\begin{cases}1 & \text{The presence of the worker } i \text{ is required in day } k\\ 0 & \text{Otherwise}\end{cases}$$

Then, the set linear constraints will satisfy: $A_2x=0$

Finally, we obtain the following 0-1 quadratic program with linear constraints $(Q'P)$:

$$(Q'P)\begin{cases}Max & g(x) = x^TQ'x\\ Subject\ \ to & \\ & A_0x=b'\\ & A_1x=T\\ & A_2x=0\\ & x\in\{0,1\}^n\end{cases}$$

Without loss of generality, the model $(Q'P)$ can be written as the following form:

$$(QP)\begin{cases}Min & f(x) = x^TQx\\ Subject\ \ to & \\ & Ax=b\\ & x\in\{0,1\}^n\end{cases}$$

With $Q=-Q'\in IR^{n\times n}$, $A=\begin{pmatrix}A_0\\ A_1\\ A_2\end{pmatrix}\in IR^{m\times n}$ and

$b=\begin{pmatrix}b'\\ T\\ 0\end{pmatrix}\in IR_+^m$, where $m=14+p$ and $n=7p$.

3. RESOLUTION OF THE 0-1 QUADRATIC PROGRAM (QP)

Consider the following linearly-constrained 0-1 quadratic program:

$$(QP)\begin{cases}Min & f(x) = x^TQx\\ Subject\ \ to & \\ & Ax=b\\ & x\in\{0,1\}^n\end{cases}$$

Therefore, Q is $n\times n$ matrix with the general term is denoted by q_{ikjl}, A is $m\times n$ matrix and $b\in IR_+^m$. Without loss of generality, we can suppose that Q is symmetric and also that diagonal terms of Q are equal to 0. If this matrix is not symmetric, it can be converted to the symmetric form $\dfrac{Q+Q^T}{2}$ and the linear terms $q_{ikik}x_{ik}$ can be substituted for the diagonal terms $q_{ikik}x_{ik}^2$, because $x_{ik}^2=x_{ik}$ for $x_{ik}\in\{0,1\}$.

Several approaches are proposed to solve (QP). For example, the linear reformulations 0-1 [3], quadratic convex reformulations 0-1, the methods enumerative based on the various relaxations as the lagrangian relaxation [15] and the semidefinite relaxation [16]-[17]-[18]. In this section, we present two methods to solve the quadratic program (QP) using the quadratic convex reformulation [19]-[20]. These methods are based on repeating a quadratic optimization problem (QP) with an objective

Journal of Theoretical and Applied Information Technology
10th March 2013. Vol. 49 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

function which is not convex in a quadratic optimization problem (QP_u) with the objective function is convex. The obtained problem is solved by the general methods of resolution of the quadratic programs in mixed variables, with a convex objective function and linear constraints.

3.1 Convex Quadratic Reformulation Based On Smallest Eigenvalue Method

In this section, we present a method to solve the quadratic program (QP) using the quadratic convex reformulation based on the smallest eigenvalue method [20].

For $u \in IR$, let us define the perturbed function $f_u(x)$ in the following way:

$$f_u(x) = f(x) + u \times \sum_{i=1}^{p}\sum_{k=1}^{7}(x_{ik}^2 - x_{ik}) = x^T Q_u x - u\sum_{i=1}^{p}\sum_{k=1}^{7} x_{ik}$$

With $Q_u = Q + u \times I_n$, where I_n is the identity matrix of size n.

We are going to determine $u \in IR$ such as:

- $f_u(x)$ is convex (Q_u is positive semidefinite matrix).
- The optimal value of the continuous relaxation (QP_u) is maximal.

$$(QP_u)\begin{cases} Min \quad f_u(x) = x^T Q_u x - u\sum_{i=1}^{p}\sum_{k=1}^{7} x_{ik} \\ Subjectto \\ \quad Ax = b \\ \quad x_{ik} \in \{0,1\}, i \in \{1,..,p\}, k \in \{1,..,7\} \end{cases}$$

Hammer and Rubin [20] propose reformulating non-convex 0-1 (QP) into convex quadratic program (QP_u), with $u = -\lambda_{min}(Q)$. Where $\lambda_{min}(Q)$ is the smallest eigenvalue of Q.

It is well known that a convex reformulation using the smallest eigenvalue method is computationally expensive, because the complexity of the problem for finding eigenvalues of the matrix increases rapidly with increasing its size. But in this case, the following theorem determines the smallest eigenvalue of the matrix defining the objective function of the model (QP) independent of its size.

Theorem 3.1

Let Q be a $n \times n$ matrix, and let Q_0 be a 7×7 matrix defined by:

$$Q = \begin{pmatrix} -Q_0 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -Q_0 \end{pmatrix}, \quad Q_0 = \begin{pmatrix} 0 & 0.5 & 0 & 0 & 0 & 0 & 0.5 \\ 0.5 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0 & 0 & 0 & 0.5 & 0 \end{pmatrix}$$

Then

1. The matrices Q and $-Q_0$ has the same eigenvalues: $Sp(Q) = Sp(-Q_0)$

   Where $Sp(D)$ is the spectrum of a matrix D defined by: $Sp(D) = \{\lambda \in IR / \exists x \neq 0 : Dx = \lambda x\}$

2. $\lambda_{min}(Q) = -\lambda_{max}(Q_0) = -1$.

Proof.

1) If $\lambda \in Sp(-Q_0)$ then $\exists x \in IR_*^7$ such that $-Q_0 x = \lambda x$.

We consider $y \in IR_*^{7p}$ such that $y = (x, 0, ..., 0)^T$. Thus, $Q \times y = \lambda \times y$ then $\lambda \in Sp(Q)$. This implies that $Sp(-Q_0) \subset Sp(Q)$.

If $\lambda \in Sp(Q)$ then $\exists x = (x_1, x_2, ..., x_p)^T \in IR_*^{7p}$ such that $Qx = \lambda x$. Thus, $Qx = (-Q_0 x_1, -Q_0 x_2, ..., -Q_0 x_p)^T$ and

$$\lambda x = \lambda(x_1, x_2, ..., x_p)^T = (\lambda x_1, \lambda x_2, ..., \lambda x_p)^T$$

So $-Q_0 x_i = \lambda x_i, \forall i \in \{1,2,...,p\}$. As the vector $x \in IR_*^{7p}$ then $\exists i \in \{1,2,..., p\}$ such as $x_i \in IR_*^7$ and $-Q_0 x_i = \lambda x_i$ then $\lambda \in Sp(-Q_0)$. This implies that $Sp(Q) \subset Sp(-Q_0)$.

Finally, we conclude that $Sp(Q) = Sp(-Q_0)$.

2) We have $Sp(-Q_0) = -Sp(Q_0)$ then

$$Min(Sp(Q)) = Min(Sp(-Q_0)) = -Max(Sp(Q_0))$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

We conclude that $\lambda_{min}(Q) = -\lambda_{max}(Q_0)$ and the maximum eigenvalue of the matrix $Q_0 \in IR^{7\times7}$ is: $\lambda_{max}(Q_0) = 1$.

Remark 3.1

Using the smallest eigenvalue method to convexify the model of days-off scheduling problem (QP) has many advantages:

- This reformulation consists of perturbing the Q matrix by adding 1 in its diagonal in order to obtain a positive semidefinite matrix.
- We reformulate (QP) into an equivalent program, with a convex objective function independent of computation time (see theorem 3.1).
- The days-off scheduling problem become a convex quadratic program independent of the number of workers.

Consequently, the transformed problem can be solved by a standard solver that uses a Branch and Bound algorithm. It is well known that the behavior of the associated Branch and Bound algorithm is very dependent on the bound at the root of the search tree. In order to maximize the lower bound obtained by solving the continuous relaxation of the (QP_u) problem, we are going to reformulate the (QP) problem using the semidefinite programming.

3.2 Convex Quadratic Reformulation Based On Semidefinite Programming

In this section, we present a method to solve (QP) using the quadratic convex reformulation based on the semidefinite programming [19]. This approach is divided into two steps: the first step involves convexifing the objective function using semidefinite programming. The second step concerns solving (QP_{u^*}) using a Mixed-Integer Quadratic Programming solver.

Let $Y = \{x : Ax = b, x \in \{0,1\}^n\}$ be the set of feasible solutions of problem (QP) and $\overline{Y} = \{x : Ax = b, x \in [0,1]^n\}$ the continuous set of feasible solutions.

For any $u \in IR^n$ with $n = 7p$,

$$u = (u_{11}, u_{12}, \ldots, u_{17}, u_{21}, u_{22}, \ldots, u_{27}, \ldots, u_{p1}, u_{p2}, \ldots, u_{p7})^T$$

Let us define the perturbed function $f_u(x)$ in the following way:

$$f_u(x) = f(x) + \sum_{i=1}^{p}\sum_{k=1}^{7} u_{ik}(x_{ik}^2 - x_{ik}) = x^T Q_u x - \sum_{i=1}^{p}\sum_{k=1}^{7} u_{ik} x_{ik}$$

And that $f_u(x) = f(x)$ for all $x \in \{0,1\}^n$

$$f_u(x) = x^T Q_u x + c_u^T x$$

Where $Q_u = Q + Diag(u)$, $c_u = -u$ and $Diag(u)$ is a square n-matrix with the elements of u on the main diagonal.

It is obvious to notice that an equivalent problem to (QP) is (QP_u):

$$(QP_u) \quad \begin{cases} Min & f_u(x) \\ Subject\ to & \\ & x \in Y \end{cases}$$

We are going to determine $u \in IR^n$ such as $f_u(x)$ is convex and the value of the continuous relaxation of (QP_u) is maximal.

The semidefinite positive relaxations can be used to solve the following generalized 0-1 quadratic problem with linear constraint (QP_u). They are linear programs over the cone of positive semidefinite matrices. Most of the results quoted in this section can be found in standard matrix theory books, as [21]-[22].

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

$$\begin{aligned} & x_{ik} \in \{0,1\},\ i \in \{1,..,p\},\ k \in \{1,..,7\} \\ \Leftrightarrow\ & x_{ik}^2 - x_{ik} = 0,\ i \in \{1,..,p\},\ k \in \{1,..,7\} \\ \Leftrightarrow\ & diag(xx^T) - x = 0 \end{aligned}$$

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

$$Diag(X) - x = 0,\ X = xx^T$$

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

Journal of Theoretical and Applied Information Technology
10th March 2013. Vol. 49 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$(QP_u)\quad \begin{cases} Min & f_u(x) \\ Subject\ to & \\ & Ax = b \\ & X = xx^T, \quad Diag(X) = x \end{cases}$$

We will use the notation $X \geq 0$ to express that X is positive semidefinite matrix. An obvious method to obtain a semidefinite relaxation of (QP_u) is to relax the last constraint $X = xx^T$ to $X \geq xx^T$, which is now convex with respect that the set of $(X,x) \in S_n \times \mathrm{P}^n$ (S_n is the space of symmetric $n \times n$ matrices) satisfying $X \geq xx^T$ is closed and convex [10]. Actually:

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

Then, we obtain the following theorem:

Theorem 3.2 [19]

The optimal values u^*_{ik} of u_{ik}, $i \in \{1,..,p\}$ and $k \in \{1,..,7\}$ are given by the optimal values of the dual variables associated with constraints (1) in the problem (SDP):

$$(SDP)\quad \begin{cases} Min & \sum_{i=1}^{p}\sum_{k=1}^{7}\sum_{j=1}^{p}\sum_{l=1}^{7} q_{ikjl} X_{ikjl} \\ subject\ \ to & \\ & X_{ikik} = x_{ik}, i \in \{1,..,p\}, k \in \{1,..,7\} \quad (1) \\ & Ax = b \\ & \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \geq 0 \\ & x \in IR^n \quad , \quad X \in S_n \end{cases}$$

Where q_{ikjl} is the general term of matrix Q, which defines the initial objective function.

Remark 3.2

For each instance of the days-off scheduling problem, we reformulate it into a convex quadratic program using the semidefinite programming (SDP). The computation time required to obtain this convexification depends on the number of workers, but it maximizes the lower bound obtained by solving the continuous relaxation of this instance. The improved bounds obtained by reformulation can be expected to lead to a reduction in the number of branch-and-bound nodes.

The resolution process of days-off scheduling problem using the semidefinite programming compared by the resolution using the smallest eigenvalue method leads to a complex resolution with a lot of indices on variables. The latter can be seen very hard to understand. We prefer to describe these processes by an example.

3.3 Example

To explain the main steps of or methods, we consider the days-off scheduling problem defined by:

- $p = 3$ is the number of workers.
- T_i is the number of days-off for worker i in the week, with $i \in \{1,2,3\}$ such as $T_1 = T_3 = 3$ and $T_2 = 2$.
- n_k is the number of workers needed on day k, with $k \in \{1,..,7\}$ such as $n_1 = n_2 = n_5 = n_6 = 2$, $n_3 = n_4 = 1$ and $n_7 = 3$.
- $C = \{c_1, c_2, c_5, c_6, c_7\}$ is the set of classes of workers required to perform a task on the day j, with $c_1 = \{worker\ 3\}, c_2 = \{worker\ 2\}$, $c_5 = \{worker\ 1\}$, $c_6 = \{worker\ 2\}$ and $c_7 = \{worker\ 1,\ worker\ 2,\ worker\ 3\}$.

This problem is modeled in the form of a quadratic program with 0-1 variables by:

$$(QP)\begin{cases} Min & f(x) = x^T Qx \\ Subject\ \ to & \\ & Ax = b \\ & x \in \{0,1\}^{21} \end{cases}$$

Where A is a 17×21 matrix and $b \in IR^{17}_+$. Then, an equivalent problem to (QP) is (QP_u):

$$(QP_u)\quad \underset{x \in Y}{Min}\, x^T Q_u x - u \sum_{i=1}^{p}\sum_{k=1}^{7} x_{ik}$$

Where $Y = \{x \in \{0,1\}^{21}, Ax = b\}$.

The quadratic convex reformulation using the smallest eigenvalue method solve (QP_u) with $u = -\lambda_{min}(Q) = 1$.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The quadratic convex reformulation using the semidefinite programming solve (QP_u), where $u \in IR^{21}$ is given by the optimal values of the dual variables associated with constraints (1) in the semidefinite program (SDP) (see theorem 3.2).

The following table sums up the results obtained by these reformulations:

Optimal Value	*Smallest eigenvalue method*			
	CPU	*Bound*	*GAP*	*nodes*
5	0.07	9.93	98.71	11

Optimal Value	*SDP method*			
	CPU	*Bound*	*GAP*	*nodes*
5	0.07	7.89	57.82	10

Then, we obtain the exact solution of quadratic program (QP) with these techniques.

$$x = (1,1,1,0,0,\ 0,0,0,0,1,\ 1,0,0,0,0,\ 0,0,1,1,1,\ 0)^T$$

By applying the following coding, for $k \in \{1,..,7\}$, *and* $i \in \{1,2,3\}$:

$$x_{ik} = \begin{cases} 1 & \textit{If day k is an off day for worker i} \\ 0 & \textit{Otherwise} \end{cases}$$

We obtain the solution of days-off scheduling problem.

- The consecutive off days of worker 1: Monday, Tuesday, Wednesday.
- The consecutive off days of worker 2: Wednesday, Thursday.
- The consecutive off days of worker 3: Thursday, Friday, Saturday.

4. COMPUTATIONAL RESULTS

In order to test the performance of these convexification algorithms for solving days-off scheduling problem, one needs a benchmark. An easy way to build such a benchmark consists in randomly generating days-off scheduling problem instances.

4.1 Generation Of Satisfiable Instances

The goal of this subsection is to describe a generator of instances that have a specified degree of interchangeability.

To realize this program, we use the following assumptions:

- Each day has the same number of workers.
- All classes have the same number of workers.

Each class of random days-off scheduling problem instances is defined by:

- p denotes the number of workers.
- r denotes the number of workers needed on each day: $r = n_k \quad \forall k \in \{1,..,7\}$.
- q denotes the number of workers of the class for each weekday: $q = q_k \quad \forall k \in \{1,..,7\}$.

To generate random days off scheduling problem instance, we need to generate random numbers of days-off in the week of each worker. We generate these numbers by the following steps:

1. Let $m = 7p - 7r$ be an integer.
2. Calculate the remainder u and the quotient v of the integer division of m by p.
3. Choose the number of days-off for worker i in the week, with $i \in \{1,.., p\}$ such that:

$$T_i = \begin{cases} v+1 & \textit{If} \quad i \in \{1,.., u\} \\ v & \textit{If} \quad i \in \{u+1,.., p\} \end{cases}$$

Finally, we present a new method to generate random hard satisfiable instances for the days-off scheduling problem. These instances have computational properties more similar to real-world (industrial) instances.

4.2 Numerical Results

For evaluating and showing the practical interest of these reformulations, we have effectuated the series of experimentations to solve the days-off scheduling problem. These experiments are effectuated in personal computer with processor Intel Core i3 2, 53 GHz, and 3 Go of RAM. We use Cplex $v12.1$ [23] to solve the convex quadratic programming problem (QP) and we choose to solve (SDP) using $CSDP$ [24].

Journal of Theoretical and Applied Information Technology
10th March 2013. Vol. 49 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table1: Comparison Of Smallest Eigenvalue Method And SDP Method

Instances			*Opt*	*Smallest eigenvalue method*				*SDP method*				
p	*r*	*q*		*CPU*	*Bound*	*GAP*	*nodes*	CPU_S	CPU_C	*Bound*	*GAP*	*nodes*
6	3	1	15	0.33	27.05	80.32	6478	0.06	0.27	24.81	65.41	5766
7	4	2	14	0.23	25.71	83.64	3954	0.03	0.16	22.82	63.01	1847
8	4	2	18	0.45	30.75	69.86	9501	0.03	0.42	29.24	62.43	8526
8	5	3	11	0.22	23.13	110.25	4109	0.07	0.14	19.60	78.17	2106
9	5	3	16	0.31	28.65	79.05	5916	0.09	0.20	24.85	55.31	3635
9	6	3	12	2.39	27.73	131.10	59401	0.07	1.00	23.14	92.82	20270
12	7	4	20	110.67	39.32	96.59	2739334	0.12	83.37	36.07	80.33	2028010
12	8	5	14	33.95	32.81	134.39	876316	0.20	2.14	24.47	74.82	48211
12	9	5	9	121.06	29.76	230.72	2944369	0.15	13.82	21.22	135.78	330119
12	9	6	8	2.50	25.72	221.49	63584	0.15	1.12	20.19	152.38	25719
14	9	5	18	428.55	40.04	122.47	9259950	0.14	67.50	32.75	81.96	1515657
14	10	6	12	217.73	33.36	177.97	5034857	0.23	11.51	24.79	106.55	265771
14	10	7	14	192.22	35.77	155.47	4512576	0.15	39.58	29.89	113.51	909904
14	11	7	6	403.64	29.64	394.01	8164777	0.15	72.27	20.87	247.91	1602462
14	11	8	6	49.83	28.15	369.16	1213658	0.15	11.18	19.04	217.40	256839
16	10	7	23	2855.47	46.15	100.66	46442044	0.26	67.67	39.03	69.71	1345552
16	11	8	16	162.88	38.79	142.43	3791387	0.20	19.27	30.16	88.52	434666
16	12	8	12	368.71	35.56	196.31	7565007	0.18	7.49	25.21	110.11	160488
18	14	10	7	270.43	30.95	342.20	5885938	0.24	68.73	22.12	216.06	1465095
20	15	12	12	985.57	38.05	217.11	17787035	0.37	3.56	27.21	126.76	79965

Legend of the Table 1

- *Opt*: Value of the optimal or best known solution.
- *CPU* : Total CPU time required by CPLEX 12.1 to solve $(QP_{\lambda_{min}})$.
- CPU_S : Total CPU time required by *CSDP* .
- CPU_C : Total CPU time required by CPLEX 12.1 to solve (QP_{u^*}).
- *Bound* : Optimal value of the continuous relaxation of $(QP_{\lambda_{min}})$ or (QP_{u^*}).
- $GAP = |\frac{Bound - opt}{opt}| * 100$.
- *nodes* : Number of nodes in the search tree.

Table 1 summarizes the results of the executions of these approaches on these randomly generated instances. The column CPU_S denotes the *CPU* time spent by *CSDP* program to solve the associated semidefinite relaxation to obtain the optimal parameters for the convex reformulation. This *CPU* time depends on the number of workers, but the smallest eigenvalue method is a very quick transformation: We add one to each term of the Hessian diagonal. Generally, *SDP* is better than smallest eigenvalue: It is consumes less time. Note that the GAP obtained by *SDP* method is better than smallest eigenvalue method.

5. CONCLUSION

In this paper, we discuss the use of various convex reformulations to find a solution for days-off scheduling problem with day task constraints. This problem has been presented as 0-1 quadratic programming subject to linear constraints. To solve this problem, we have used the quadratic convex reformulation based on two techniques, the first one is based on the smallest eigenvalue method and the second uses the semidefinite programming. Some numerical examples which assess the effectiveness of the theoretical results as well as the advantages of this model are shown in this paper. Several directions can be investigated to try to improve this

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

method, such as using the constraint satisfaction problem [18] and neural networks approaches [25].

REFRENCES:

[1] H. K. Alfares, "Optimum workforce scheduling under the (14, 21) days-off timetable", *Journal of Applied Mathematics and Decision sciences*, 6 (3), 2002, pp. 191-199.

[2] M. Elshafei and H. K. Alfares, "A dynamic programming algorithm for days-off scheduling with sequence dependent labor costs", *Springer Science-Business Media*, LLC 2007.

[3] H. D. Sherali and WP. Adams, "A Reformulation-Linearization Technique for solving Discrete and Continuous No convex Problems", Kluwer Academic Publishers, Norwell, 1999, MA.

[4] M. W. Carter, "The indefinite zero-one quadratic problem", *Discrete Applied Mathematics*, 7, 1984, pp. 23-44.

[5] M. C. Plateau, A. Billionnet, and S. Elloumi, "Eigen value methods for linearly constrained quadratic 0-1 problems with application to the densest k-sub graph problem", In 6ème congrès ROADEF, Tours, 14-16 février, Presses Universitaires Francois Rabelais, 2005, pp. 55-66.

[6] K. M. Anstreicher and N. W. Brixius, "A new bound for the quadratic assignment problem based on convex quadratic programming", *Mathematical Programming*, 89, 2001, pp. 341-357.

[7] P. Chardaire and A. Sutter, "A decomposition method for quadratic zero-one programming", *Management Science*, 41(4), 1995, pp. 704-712.

[8] F. Glover and E. Woolsey, "Further reduction of 0-1 polynomial programming problems to 0-1 linear programming problems", *Operations Research*, 21, 1973, pp. 156-161.

[9] M. X. Goemans, "Semi definite programming in combinatorial optimization", *Mathematical Programming*, 79, 1997, pp. 143-161.

[10] C. Helmberg. "Semidefinite programming", *European Journal of Operational Research*, 37, 2002, pp. 461-482.

[11] C. Helmberg and F. Rendl, "Solving quadratic (0, 1)-problems by semi definite programs and cutting planes", *Math. Programming*, 8(3), 1998, pp. 291-315.

[12] P. Michelon and N. Maculan, "Lagrangian decomposition for integer non linear programming with linear constraints", *Mathematical Programming*, 52(2), 1991, pp. 303-314.

[13] N. Z. Shor. "Quadratic optimization problems", Soviet Journal of Computer Systems Sciences, 25, 1987, pp. 1-11.

[14] A. Billionnet, "Integer programming to schedule a hierarchical workforce with variable demands", *European Journal of Operational Research*, 114, 1999, pp. 105-114.

[15] P. Michelon and N. Maculan, "Lagrangian decomposition for integer non linear programming with linear constraints", *Mathematical Programming*, 52(2), 1991, pp. 303-314.

[16] W. P. Adams, R. Forrester and F. Glover, "Comparisons and enhancement strategies for linearizing mixed 0-1 quadratic programs", *Discrete Optimization*, 1(2), 2004, pp. 99-120.

[17] M. Ettaouil and C. Loqman, "Constraint Satisfaction Problems Solved by Semi definite Relaxations", *Journal of Wseas Transactions on Computers*, 7(7), 2008, pp. 951-961.

[18] M. Ettaouil and C. Loqman, "A New Optimization Model for Solving the Constraint Satisfaction Problem", *Journal of Advanced Research in Computer Science*, 1(1), 2009, pp. 13-31.

[19] A. Billionnet, E. Soutif, and M. C. Plateau, "Convex quadratic programming for exact solution of 0-1 quadratic programs", *Technical Report* RC856, CEDRIC, 2005.

[20] P. L. Hammer and A. A. Rubin, "Some remarks on quadratic programming with 0-1 variables", RAIRO, 3, 1970, pp. 67-79.

[21] R. Horn and C. Johnson, "Matrix Analysis", Cambridge University Press, 1985, pp. 231-279.

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

[23] "IBM ILOG CPLEX: High-performance mathematical programming engine." http://www.01.ibm.com/software/integration/optimization/cplex/, 2010.

[24] B. Borchers, "CSDP, a C library for semidefinite programming". *Optim. Methods Softw*, 11/12(1-4), 1999, pp. 613-623.

[25] M. Ettaouil, C. Loqman, Y. Hami, K. Haddouch, "Task Assignment Problem Solved by Continuous Hopfield Network", *International Journal of Computer Science*, 9(2), 2012, pp. 206-212