# Several versions of the devour digest tidy-up heuristic for unconstrained binary quadratic problems — Source link

Saïd Hanafi, Ahmed-Riadh Rebai, Michel Vasquez

**Institutions:** University of Valenciennes and Hainaut-Cambresis, Texas A&M University at Qatar, Mines ParisTech

**Published on:** 01 Aug 2013 - Journal of Heuristics (Springer US)

**Topics:** Quadratic unconstrained binary optimization, Quadratic function, Heuristic, Local search (optimization) and Quadratic equation

# Several versions of the devour digest tidy-up heuristic for unconstrained binary quadratic problems

Said Hanafi, Ahmed-Riadh Rebai, Michel Vasquez

## To cite this version:

Said Hanafi, Ahmed-Riadh Rebai, Michel Vasquez. Several versions of the devour digest tidy-up heuristic for unconstrained binary quadratic problems. Journal of Heuristics, Springer Verlag, 2013, 19 (4), pp.645-677. 10.1007/s10732-011-9169-z . hal-00814716

# Several versions of Devour Digest Tidy-up Heuristic for Large-Scale Unconstrained Binary Quadratic Problem

**Saïd Hanafi[1], Ahmed-Riadh Rebai[2], Michel Vasquez[3]**

1. LAMIH, Université de Valenciennes, F 59313 Valenciennes Cedex9, France
Said.Hanafi@univ-valenciennes.fr

2. Electrical & Computer Engineering Program ; Texas A&M University at Qatar; Doha, Qatar
Ahmed.Rebai@qatar.tamu.edu

3. LGI2P, Ecole des Mines d'Alès, Parc scientifique Georges Besse, 30035 Nîmes, France
Michel.Vasquez@mines-ales.fr

Abstract :
The unconstrained binary quadratic optimization problem (UQP) is known to be NP-hard and due to its computational challenge and application potential, it becomes more and more considered and involved by the recent research studies, including both exact and heuristic solution approaches. Our work is in line with that of Glover, Alidaee, Rego and Kochenberger (2002) who proposed one pass heuristics as alternatives to the well-known Devour Digest Tidy-up procedure (DDT) of Boros, Hammer and Sun (1989). The "devour" step sets a term of the current representation to 0 or 1, and the "tidy-up" step substitutes the logical consequences derived from the "digest" step into the current quadratic function. We propose several versions of the DDT constructive heuristic based on the alternative representation of the quadratic function. We also present an efficient implementation of local search using r-flip moves that simultaneously change the values of r binary variables. Computational experiments performed on large scale instances show the efficiency of our implementation in terms of quality and CPU time.

Keywords : Unconstrained Quadratic Programming, Constructive Heuristic, Local Search, r-Flip Move

## 1. Introduction

The unconstrained binary quadratic minimization problem (UQP) is defined as an optimization problem that aims to find the minimum value of a given quadratic function $q(x)$ over $x$ in $\{0, 1\}^n$. Each quadratic function $q(x)$ is a real-value function defined in $\{0, 1\}^n$ which has at most a degree of 2 for its unique polynomial expression. The UQP problem is an NP-hard combinatorial optimization problem introduced by Hammer and Rudeanu (1968). The UQP can be formulated as follows:

$$\text{(UQP)} \quad \min \{ q(x) = xAx + bx + q_0 \colon x \in \{0,1\}^n \}$$

where $A$ is an $n$ by $n$ matrix, $b$ is an $n$-vector, $q_0$ is a real constant and $x$ is an $n$-vector of binary variables. The $q(x)$ value is the quadratic objective value of binary vector $x$.

The UQP problem has been investigated in numerous papers see e.g. Hammer and Rudeanu (1968), Hansen, Jaumard and Mathon (1993), Boros and Hammer (2002), and the references mentioned therein. The UQP's formulation is suitable to represent a wide range of important problems, including those from social psychology, Harary (1953); financial analysis, Laughunn (1970), McBride and Yormak (1980); computer aided design, Krarup and Pruzan (1978); traffic management, Gallo and Simeone (1998), Witsgall (1975); machine scheduling, Alidaee et al. (1994); cellular radio channel allocation, Chardaire and Sutter (1994); molecular conformation, Phillips and Rosen (1994). Moreover, many combinatorial optimization problems pertaining to graphs such as determining maximum cliques, maximum cuts, maximum vertex packing, minimum coverings, maximum independent sets, and maximum independent weighted sets are known to be capable of being formulated by the UQP problem as documented in papers of Pardalos and Rodgers (1990), and Pardalos and Xue (1994). Other additional applications and formulations can be found in Kochenberger et al. (1998). For example, the *maximum independent set* problem is equivalent to the following UQP:

$$(UQP)\ min\{\ x(A - I)x :\ x \in \{0,1\}^n\}$$

where $A$ is the adjacency matrix of a given undirected graph $G=(V, E)$ and $I$ is the identity matrix with appropriate dimension. Similarly, the maximum clique problem is given as follows:

$$(UQP) \qquad min\{\ x(\overline{A} - I)x :\ x \in \{0,1\}^n\}$$

where $\overline{A}$ is the adjacent matrix of $\overline{G} = (V, \overline{E})$ the complement graph of $G$. The m*aximum cut problem* can be addressed as the following UQP:

$$(UQP) \qquad min\{\ xAx - xAe :\ x \in \{0,1\}^n\}$$

where $A_{ij}$ *denotes* the weight of the edge $(i, j) \in E$.

Another application of the UQP takes place in condensed matter physics. *The Ising Spin Glasses* problem consists to found a configuration of the spins with minimum energy can be written as follows:

$$(UQP)\ min\{4xAx - 2(A + A^T)x + eAe :\ x \in \{0,1\}^n\}$$

where $A^T$ denotes the transpose matrix of $A$ and $A_{ij}$ denotes the interaction between site $i$ and $j$.

Other problems such that Maximum Vertex Packing, Minimum Covering, Maximum independent weighted sets, can be also formulated as an UQP problem.

$$(UQP) \qquad min\{\ x(\overline{A} - I)x :\ x \in \{0,1\}^n\}$$

Due to its computational challenge and application capability, the UQP becomes more and more considered and involved by the recent research studies, including both exact and heuristic solution approaches. Several exact methods have been developed and tested for UQP in the literature. Actual solving exact algorithms include those that attack the problem with some kind of branch and bound method or use linear programming techniques and some cutting plane generation methods. However, the exact methods degrade rapidly with problem size, and have meaningful application to general UQP problems with no more than 100 variables. Several heuristic algorithms, based on different ideas, were proposed recently in the literature to find acceptable solutions for such large problems. The heuristic ideas applied to UQP include Tabu Search (Glover et al. (1998); Beasley (1998); Palubeckis (2004-2006)), Scatter Search (Amini et al. (1999)), Simulated Annealing (Alkhamis et al. (1998); Beasley (1998), Katayama and Narihisa (2001)), Evolutionary Algorithms (Lodi et al. (1999); Merz and Freisleben (1999); Katayama et al. (2000); Borgulya (2005)), and Memetic Algorithms (Merz and Katayama (2004)). Recent studies addressing the UQP are those by Williams (1985), Pardalos and Rodgers (1992), Boros, Hammer and Sun (1989), Chardaire and Sutter (1994), Glover, Kochenberger and Alidaee (1998), Glover, Kochenberger, Alidaee, and Amini (1999), Alkhamis, Hasan and Ahmed (1998), Beasley (1998), Lodi, Allemand and Liebling (1997), Amini, Alidaee and Kochenberger (1999), Glover, Amini, Kochenberger and Alidaee (1999), Katayama, Tani and Narihisa (2000), and Merz and Freisleben (1999).

A large number of solution procedures have been reported in the literature.

This paper is focusing on the study of constructive heuristics for large-scale zero-one UQP problems. Such approaches can serve as advanced starting points input for more improving methods based on one or more solutions such that local search procedure or population methods . The rest of the paper is organized as follows. In section 2, we present the one-pass heuristics designed for this study. To provide a basis for comparison as well as a rationale for our heuristics, we begin with a discussion of DDT, the best known and most promising one-pass heuristic in the current literature. Then, in section 3, we present our computational experience. Our computational work is divided into two parts. The first part provides a relative comparison of the various one-pass methods by extensive testing on new test problems ranging from 1000 to 9000 variables. The second part shows how the best of the methods perform on standard test problems where "best known" solutions are available. Finally, section 4 presents conclusions and comments for future work.

## 1. Generalized Constructive Heuristics

By observing that $x_i x_j = x_j x_i$, when $x_i$ and $x_j$ are 0-1 variables, the UQP model can be stated in the following form:

$$(UQP) \qquad \min\left\{ q(x) = q_0 + xQx = q_0 + \sum_{i=1}^{n}\sum_{j=i}^{n} q_{ij}x_i x_j \; : x \in \{0,1\}^n \right\}$$

where $Q$ is an $n$ x $n$ lower triangular matrix defined from $A$ and $b$ by the preprocessing rules :

$q_{ij} = A_{ij} + A_{ji}$ for $i < j$
$q_{ii} = A_{ii} + b_i$ for $i = j$
$q_{ij} = 0$ for $i > j$.

Since for each binary variable we have $x_j^2 = x_j$ the quadratic function $q(x)$ also can be written as follows :

$$q(x) = q_0 + \sum_{i=1}^{n} q_{ii}x_i + \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} q_{ij}x_i x_j \qquad (1)$$

Defining the following sets:

$T^{+} = \{(i, j) : 1 \le i < j \le n \text{ and } q_{ij} > 0\}$,
$T^{-} = \{(i, j) : 1 \le i < j \le n \text{ and } q_{ij} < 0\}$,
$T^{=+} = \{(i, i) : 1 \le i \le n \text{ and } q_{ii} > 0\}$, and
$T^{=-} = \{(i, i) : 1 \le i \le n \text{ and } q_{ii} < 0\}$;

with $T = T^{-} \cup T^{+} \cup T^{=+} \cup T^{=-}$. Let $N = \{1, \ldots, n\}$.

sign(a) = 1 if a > 0 and sign(a) = -1 if a < 0.

In this paper we present several constructive and improvement heuristics for the UQP. The improved heuristics start from an initial solution and seek a better solution by iteratively moving from the current solution to the next one, according to the adjacency relationships defined by a given neighborhood structure.

based on the representation of the input quadratic function

Local Search and associated metaheuristic have been the focus of widespread scientific investigation during the last decade.
We accompany to the followed versions of DDT heuristic many running-up examples, to make the ideas more clear.

## 1. Standard Devour Digest Tidy-up Heuristic

We start by the standard version of the well-known *Devour Digest Tidy-up* (DDT) heuristic of Boros, Hammer, and Sun (1989) applied to an input quadratic function $q$ of UQP. In the "devour" step, a term with the largest absolute value of coefficient is set to 0 or 1. If this largest coefficient is negative the term is set to 1 which implies that the variables involved in this term are set to 1. Moreover if the term of the largest coefficient involves only one variable, then this variable is set to 0 if the sign of the largest coefficient is negative or set to 0 otherwise. In the "digest" step all the logical consequences of the "devour" step are derived. In the "tidy-up" step the logical consequences derived are substituted into the current quadratic function. The pseudo code of the standard DDT procedure is described as follows:

```
Standard DDT Procedure ($q$)
Initialization : Compute $T^+ = \{(i, j) : 1 \le i < j \le n$ and $q_{ij} > 0\}$,
        $T^- = \{(i, j) : 1 \le i < j \le n$ and $q_{ij} < 0\}$,  $T^{=+} = \{(i, i) : 1 \le i \le n$ and $q_{ii} > 0\}$, and
        $T^{=-} = \{(i, i) : 1 \le i \le n$ and $q_{ii} < 0\}$. Set $T = T^- \cup T^+ \cup T^{=+} \cup T^{=-}$.  Set the system set $S = \emptyset$.
Devour: Find a term $q_{i^*j^*}$ from $q$ with the largest $|q_{ij}|$, i.e. set
                $(i^*, j^*) = \text{argmax}\{ |q_{ij}| : (i, j) \in T\}$
        If $(i^*, j^*) \in T^-$ (i.e. $q_{i^*j^*} < 0$) then set $x_{i^*} = 1$ and $x_{j^*} = 1$ else if $i^* = j^*$ then set
        $x_{i^*} = (1 - \text{sign}(q_{i^*j^*}))/2$. Otherwise, add $x_{i^*}x_{j^*} = 1$ to $S$.
Digest: Draw all the logical conclusions $C$ from the boolean equations in $S$.
Tidy-up: Substitute the consequence $C$ into $q$ and update $q$, $T^-$, $T^+$ and $T$.
        If $T \neq \emptyset$ then return to Devour.
Output: Solve the Boolean equations in $S$, and output $x$.
```

To illustrate, we consider the following example from Boros et al. (2008).

**Example 1:** We seek to minimize the quadratic function given by

$$q(x) = 13 - 5x_1 + 9x_2 + x_3 + 12x_4 + 7x_5 - 12x_1x_2 + 8x_1x_4 + 4x_2x_3 - 10x_2x_4 - 6x_3x_4 - 8x_4x_5$$

A step-by step procedure of standard DDT applied to such quadratic function $q(x)$ is as follows:

*Iteration* 1 : We have $(i^*, j^*) = (4, 4)$ with $q_{i^*j^*} = 12$ therefore we set $x_4 = 0$ and update the function $q$ so that

$$q(x) = 13 - 5x_1 + 9x_2 + x_3 + 7x_5 - 12x_1x_2 + 4x_2x_3.$$

*Iteration* 2 : Now we have $(i^*, j^*) = (1, 2)$ with $q_{i^*j^*} = -12$, so set $x_1 = 1$ and $x_2 = 1$. Then propagate these equalities into the function $q$ to obtain

$$q(x) = 5 + 5x_3 + 7x_5.$$

*Iteration* 3 : The maximum of $|q_{ij}|$ is reached by $(i^*, j^*) = (5, 5)$ with $q_{i^*j^*} = 7$. Set $x_5 = 0$ and propagate this assignment into the function $q$ so that

$$q(x) = 5 + 5x_3.$$

*Iteration* 4 : We have $(i^*, j^*) = (3, 3)$ with $q_{i^*j^*} = 5$ therefore set $x_3 = 0$ and update the function $q$ so that

$$q(x) = 5.$$

Now the set $T = \emptyset$, then we stop and we return the solution $x = (1,1,0,0,0)$ with $q(x) = 5$.

In the next section, we describe the DDT versions applied to a posiform representation of the quadratic function $q$.

## 2. Devour Digest Tidy-up Heuristic with Posiform

The original Devour, Digest and Tidy-up method is fitted and applied to a posiform representation of $q$. For a binary variable $x_i$ let $\overline{x_i} = 1 - x_i$ be the complement of $x_i$. Given a quadratic function $q$, it is possible to write this function using several ways in the following form

$$q(x) = p_0 + \sum_{j=1}^{n}(p_i^+ x_i + p_i^- \overline{x_i}) + \sum_{i=1}^{n}\sum_{j=i+1}^{n}(p_{ij}^{++} x_i x_j + p_{ij}^{+-} x_i \overline{x_j} + p_{ij}^{-+} \overline{x_i} x_j + p_{ij}^{--} \overline{x_i}\,\overline{x_j}) \qquad (2)$$

with $p_i^+, p_i^-, p_{ij}^{++}, p_{ij}^{+-}, p_{ij}^{-+} \geq 0$ for all $i$ and $j$. This representation is called a *posiform* expression of the UQP problem. The concept of the posiform representation will be significant for the methods subsequently described. A posiform expression can be obtained by an appropriate choice of variables to complement in the given negative terms (i.e. $q_{ij}x_ix_j$ where $q_{ij} < 0$), in order to have positive value of all coefficient of $q$. There are many ways to transform a quadratic function to an equivalent posiform by substituting complements of some variables in $q$. More precisely, a posiform expression can also be generated by choosing value of variables $y_{ij} \in [0, 1]$ for $i, j = 1, \ldots, n$ and apply successively the following rules:

$$x_i x_j = y_{ij}(1-\overline{x_i})x_j + (1-y_{ij})x_i(1-\overline{x_j}) \qquad \text{for } (i, j) \in T^- \qquad (3\text{-a})$$

$$x_i x_j = y_{ij}x_i x_j + (1-y_{ij})(1-\overline{x_i})(1-\overline{x_j}) \qquad \text{for } (i, j) \in T^+ \qquad (3\text{-b})$$

$$x_i = 1-\overline{x_i} \qquad \text{for } (i, i) \in T^{=-} \qquad (3\text{-c})$$

**Remark 1:** Note that when the variables $y_{ij}$ are binaries in equation (3-a) this can be interpreted by

$$y_{ij} = \begin{cases} 1 & \textit{if} \quad x_i x_j \quad \text{is replaced by} \quad x_i(1-\overline{x_j}) \\ 0 & \textit{if} \quad x_i x_j \quad \text{is replaced by} \quad (1-\overline{x_i})x_j \end{cases}$$

**Remark 2:** A simple procedure to obtain a posiform consists to replace each quadratic product $x_ix_j$ associated with a negative coefficient by $-x_j + x_ix_j$ (i.e. $y_{ij} = 1$); then in a second step to replace each $x_i$ with a negative coefficient by $1-\overline{x_i}$ (i.e. rule 3-c).

**Remark 3:** Complementing only the smaller index variable in each complementation step corresponds to setting

$$y_{ij} = \begin{cases} 1 & \textit{if} \quad i < j \\ 0 & \textit{otherwise} \end{cases}$$

Complementing only the larger index variable in each complementation step is equivalent to set

$$y_{ij} = \begin{cases} 1 & \textit{if} \quad i > j \\ 0 & \textit{otherwise} \end{cases}$$

Other posiform representation can be constructed where in each complementation step, complementing randomly a variable or complementing alternatively the smaller and the larger index variable.

Since all coefficients of linear and quadratic term in the posiform expression are positives, the constant $p_0$ in (2) defines a lower bound on $q$ (i.e. $p_0 \leq \min\{q(x) : x \in \{0,1\}^n\}$). The biggest value of the constant $p_0$ with which the posiform representation satisfies (2) is called the *roof-dual bound*. Hammer, Hansen, and Simeone (1984) have introduced the "roof dual" of a UQP and they have proved that the roof-dual bound is equal to the optimal value of the LP-relaxation of the following Mixed Integer Linear Problem (MIP):

MIP minimise $\sum_{i=1}^{n} \sum_{j=i}^{n} q_{ij} y_{ij}$ (4-a)

subject to

$y_{ij} \leq \square x_i$ $\quad i, j \in N$ (4-b)

$y_{ij} \leq \square x_j$ $\quad i, j \in N$ (4-c)

$y_{ij} \geq \square\ x_i + x_j - 1$ $\quad i, j \in N$ (4-b)

$x_i \in \{0,1\}$ $\quad i \in N$ (4-e)

where binary variables $y_{ij}$ represent the quadratic terms $x_i x_j$. It can be easily seen that the MILP is equivalent to the quadratic problem UQP. Constraints (4-b) and (4-c) ensure that $y_{ij}$ must be zero if either of $x_i$ or $x_j$ are zero. Constraint (4-d) ensures that $y_{ij}$ is one if both $x_i$ and $x_j$ are one. Constraints (4-e) are the integrality constraints. This linearization was proposed independently by several authors Fortet (1959), Balas (1964), Zangwill (1965), Watters (1967), Glover and Wolsey (1974), Adams, Forrester and Glover (2004), Gueye and Michelon (2005), Fortet (1959-1960), Glover (1975-1984), Goldman (1983), and Plateau (2006). Recently Hansen and Meyer (2009) propose and compare three new compact linearizations for the UQP, two of them are achieving the same lower bound than the "standard linearization". The first linearization requires $n$ additional constraints with respect to Glover's one, where $n$ is the size of the quadratic 0–1 problem, while the two others require the same number of constraints. All three linearization require the same number of additional variables than Glover's linearization.

The variable $x_i$ and its complement $\overline{x_i}$ are called literals, we introduce a new variable $x_0$ assigned to one ($x_0 = 1$), and we denote by $L = \{ x_i, \overline{x_i} : i = 0, 1, \ldots, n\}$ the set of literals. To make simpler the notation, we assume in the rest of this paper that the quadratic function $q$ is given as

$$q(x) = q_0 + \sum_{(u,v)\in T} q_{uv} uv \qquad (5)$$

whereas $T = \{(u, v) : \text{for all } u, v \in L \text{ with } q_{uv} \neq 0\}$.

In the "devour" step of posiform DDT procedure, the term with the largest coefficient is set to 0. In case of this largest term involves only one literal, we fix this literal to 0. Using this procedure all the logical consequences are derived in the "devour" step. Then, these logical consequences are substituted into the current posiform in the "tidy-up" step. The posiform DDT procedure is described below.

**Posiform DDT Procedure** ($q$)

Input : UQP $q$ in a posiform**.**

Initialization : Compute $T = \{(u, v) : p_{uv} > 0, u,v \in L\}$ and set the system $S = \varnothing$.

Devour: Find the largest coefficient $p_{u^*v^*}$ in the current posiform $q$, set

$$(u^*, v^*) = \operatorname{argmax}\{ p_{uv} : (u, v) \in T\}$$

If $u^* = v^*$ then set $u^* = 0$ otherwise, add the pseudo cut $u^*v^* = 0$ to $S$.

Digest: Draw all the logical conclusions $C$ from the boolean equations in $S$.

Tidy-up: Substitute the consequence $C$ into $q$ and update $T$. If $T \neq \varnothing$ then return to Devour.
Output: Solve the Boolean equations in $S$, and output $x$.

**Example 2:** Consider again the quadratic function of Example 1. By applying the rules (3-a), (4-b) and (4-c) we obtain an equivalent posiform representation of $q$ will be:

$$q(x) = -10 + 17\overline{x}_1 + \overline{x}_2 + 5\overline{x}_3 + 4x_4 + 7x_5 + 12x_1\overline{x}_2 + 8x_1x_4 + 4x_2x_3 + 10x_2\overline{x}_4 + 6x_3\overline{x}_4 + 8x_4\overline{x}_5$$

A step by step procedure of DDT applied to such posiform is as follows:
*Iteration* 1 : We have ($u$*, $v$*) = $(\overline{x}_1, \overline{x}_1)$ with $q_{u*v*}$ = 17 therefore set $\overline{x}_1 = 0$ (hence $x_1$ = 1) and update the function q so that

$$q(x) = -10 + 13\overline{x}_2 + 5\overline{x}_3 + 12x_4 + 7x_5 + 4x_2x_3 + 10x_2\overline{x}_4 + 6x_3\overline{x}_4 + 8x_4\overline{x}_5$$

*Iteration* 2 : Now we have ($u$*, $v$*) = $(\overline{x}_2, \overline{x}_2)$ with $q_{u*v*}$ = -13, so set $\overline{x}_2 = 0$ (hence $x_2$ = 1). Then propagate this equality into the function $q$ to obtain

$$q(x) = 4 + \overline{x}_3 + 2x_4 + 7x_5 + 6x_3\overline{x}_4 + 8x_4\overline{x}_5$$

*Iteration* 3 : The maximum of $p_{uv}$ in the current posiform is reached by ($u$*, $v$*) = $(x_4, \overline{x}_5)$ with $p_{u*v*}$ = 8. Set $S = \{ x_4\overline{x}_5 = 0 \}$ and the current posiform become

$$q(x) = 4 + \overline{x}_3 + 2x_4 + 7x_5 + 8x_4\overline{x}_5$$

*Iteration* 4 : We have ($u$*, $v$*) = $(x_5, x_5)$ with $q_{u*v*}$ = 7 therefore set $x_5$ = 0 and draw all the logical conclusions from the Boolean equation $x_4\overline{x}_5 = 0$ in $S$ to obtain $x_4$ = 0. Substitute the consequence $x_4 = x_5 = 0$ into $q$ so that
$q(x) = 5 + 5x_3$
*Iteration* 5 : Finally, we have ($u$*, $v$*) = $(x_3, x_3)$ with $q_{u*v*}$ = 5 therefore set $x_3$ = 0 and update the function $q$ so that
$q(x) = 5$.
Now the set $T = \varnothing$. Stop and return $x$ = (1,1,0,0,0) with $q(x)$ = 5.

## 1. Devour Digest Tidy-up Heuristic with Negaform

A negaform expression can be obtained by an appropriate choice of variables to complement in the given positive terms (i.e. $q_{ij}\, x_i x_j$ where $q_{ij} > 0$), in order to have all coefficient of $q$ negative. A quadratic function $q$ can be transformed to an equivalent *negaform* by substituting complements of some variables in $q$. Specifically, a negaform can also be generated by choosing value of variables $y_{ij} \in [0, 1]$ for $i, j$ = 1, …, $n$ and apply successively the following rules:

$$x_i x_j = y_{ij}(1-\overline{x_i})x_j + (1-y_{ij})x_i(1-\overline{x_j}) \qquad \text{for } (i, j) \in T^{+} \qquad \text{(6-a)}$$

$$x_i x_j = y_{ij}x_i x_j + (1-y_{ij})(1-\overline{x_i})(1-\overline{x_j}) \qquad \text{for } (i, j) \in T^{-} \qquad \text{(6-b)}$$

$$x_i = 1-\overline{x_i} \qquad \text{for } (i, i) \in T^{=+} \qquad \text{(6-c)}$$

The similary remarks of the rules 3-x can be cited for 6-x.

**Negaform DDTprocedure** ($q$)
Input : UQP $q$ in a negaform.
Initialization : Compute $T = \{(u, v) : p_{uv} < 0, u,v \in L\}$.
Devour: Find the smallest coefficient $p_{u*v*}$ in the current posiform $q$, set

$$(u^*, v^*) = \text{argmin}\{ p_{uv} : (u, v) \in T\}$$

Digest and Tidy-up: Set $u^* = 1$ and $v^* = 1$. Substitute the assignment $u^* = v^* = 1$ into $q$ and update $T$. If $T \neq \varnothing$ then return to Devour.
Output: Return the solution $x$.

**Example 3:** We seek to minimize the quadratic function given by
$q(x) = 13 - 5x_1 + 9x_2 + x_3 + 12x_4 + 7x_5 - 12x_1x_2 + 8x_1x_4 + 4x_2x_3 - 10x_2x_4 - 6x_3x_4 - 8x_4x_5$
Consequently, by completing the small index the equivalent negaform representation of $q$ is:
$q(x) = 54 - 5x_1 - 9\bar{x}_2 - 5\bar{x}_3 - 20\bar{x}_4 - 7\bar{x}_5 - 12x_1x_2 - 8\bar{x}_1x_4 - 4\bar{x}_2x_3 - 10x_2x_4 - 6x_3x_4 - 8x_4x_5$
A step by step procedure of DDT applied to such negaform is as follows:
*Iteration* 1 : We have $(u^*, v^*) = (\bar{x}_4, \bar{x}_4)$ with $q_{u^*v^*} = -20$ therefore set $\bar{x}_4 = 1$ (hence $x_4 = 0$) and update the function $q$ so that
$q(x) = 34 - 5x_1 - 9\bar{x}_2 - 5\bar{x}_3 - 7\bar{x}_5 - 12x_1x_2 - 4\bar{x}_2x_3$
*Iteration* 2 : Now we have $(u^*, v^*) = (x_1, x_2)$ with $q_{u^*v^*} = -12$, therefore set $x_1 = 1$ and $x_2 = 1$. Then propagate these equalities into the function $q$ to obtain
$q(x) = 17 - 5\bar{x}_3 - 7\bar{x}_5$
*Iteration* 3 : The minimum of $q_{uv}$ in the current negaform is reached by $(u^*, v^*) = (\bar{x}_5, \bar{x}_5)$ with $p_{u^*v^*} = -7$. Therefore set $\bar{x}_5 = 1$ (hence $x_5 = 0$) and after propagation the current negaform become
$q(x) = 10 - 5\bar{x}_3$
*Iteration* 4 : Finally, we have $(u^*, v^*) = (\bar{x}_3, \bar{x}_3)$ with $q_{u^*v^*} = -5$ therefore set $\bar{x}_3 = 1$ (hence $x_3 = 0$) and update the function $q$ so that
$q(x) = 5$.
Now the set $T = \varnothing$. Stop and return $x = (1,1,0,0,0)$ with $q(x) = 5$.

## 1. Devour Digest Tidy-up Heuristic with Bi-form

A particular posiform of a quadratic function is the bi-form representation which was introduced by Boros, Hammer and Sun (1989) (see also Boros, Hammer, Sun and Tavares (2008)). Let $x_i$ and $x_j$ be two binary variables, the expression $x_{ij}^+ = x_i\bar{x}_j + \bar{x}_i x_j$ is called a positive bi-term and $x_{ij}^- = x_i x_j + \bar{x}_i\bar{x}_j$ is called a negative bi-term. It is easy to figure out that the bi-terms express naturally the equality or non-equality of the variables involved:

$$x_{ij}^+ = x_i\bar{x}_j + \bar{x}_i x_j = 0 \Leftrightarrow x_i = x_j$$

$$x_{ij}^- = x_i x_j + \bar{x}_i\bar{x}_j = 0 \Leftrightarrow x_i = \bar{x}_j$$

In the DDT heuristic, the term with the largest coefficient is set to 0 and the logical consequences derived are substituted into the current biform. A *bi-form* $q$ is a quadratic function containing only bi-terms with positive coefficient, i.e. $q(x) = b_0 + \sum_{i \neq j} (b_{ij}^- x_{ij}^- + b_{ij}^+ x_{ij}^+)$ with $b_{ij}^-, b_{ij}^+ > 0$. Thus, any quadratic function $q$ has a unique bi-form representation which can be obtained by applying successively the transformations (7-a) and (7-b) of its positive and negative quadratic terms ($1 \leq i < j \leq n$), and then the transformations (7-c) and (7-d) of its positive and negative linear terms ($i = 1, \ldots, n$), with $x_0 = 1$.

$$x_i x_j = \frac{1}{2}(x_i + x_j) - \frac{1}{2}(x_i \overline{x_j} + \overline{x_i} x_j) \qquad \text{for } (i, j) \in T^- \qquad \text{(7-a)}$$

$$x_i x_j = \frac{1}{2}(x_i x_j + \overline{x_i}\,\overline{x_j}) + \frac{1}{2}(x_i + x_j) - \frac{1}{2} \qquad \text{for } (i, j) \in T^+ \qquad \text{(7-b)}$$

$$x_i = 1 - (x_i \overline{x_0} + \overline{x_i} x_0) \qquad \text{for } (i, i) \in T^{=-} \qquad \text{(7-c)}$$

$$x_i = x_i x_0 + \overline{x_i}\,\overline{x_0} \qquad \text{for } (i, i) \in T^{=+} \qquad \text{(7-d)}$$

The given transformations (7-x) for linear and quadratic terms can be readdressed as follows :

$$x_i x_j = 1 - \frac{1}{2}(x_{ij}^- + x_{i0}^- + x_{0j}^-) \qquad \text{for } (i, j) \in T^- \qquad \text{(8-a)}$$

$$x_i x_j = \frac{1}{2}(x_{ij}^+ + x_{i0}^+ + x_{0j}^+ - 1) \qquad \text{for } (i, j) \in T^+ \qquad \text{(8-b)}$$

$$x_i = 1 - x_{i0}^- \qquad \text{for } (i, i) \in T^{=-} \qquad \text{(8-c)}$$

$$x_i = x_{i0}^+ \qquad \text{for } (i, i) \in T^{=+} \qquad \text{(8-d)}$$

By applying the rules (8-x) to the quadratic function $q(x) = xQx$, we obtain the following Biform representation:

$$q(x) = b_0 + \sum_{i=1}^{n}(b_{i0}^+ x_{i0}^+ - b_{i0}^- x_{0i}^-) + \sum_{i=1}^{n}\sum_{j=i+1}^{n}(b_{ij}^+ x_{ij}^+ - b_{ij}^- x_{ij}^-) \qquad \text{(9)}$$

where

$$b_0 = \sum_{i=1}^{n}(q_i^- - \sum_{j=i+1}^{n}(q_{ij}^- - \frac{q_{ij}^+}{2}))$$

$$b_{i0}^+ = \frac{1}{2}(q_i^+ + \sum_{i=1}^{n} q_{ij}^+) \qquad b_{ij}^+ = \frac{q_i^+}{2}$$

$$b_{i0}^- = \frac{1}{2}(q_i^- + \sum_{i=1}^{n} q_{ij}^-) \qquad b_{ij}^- = \frac{q_i^-}{2}$$

**Remark 4 :** $\min\{b_{ij}^+, b_{ij}^-\} = b_{ij}^+ b_{ij}^- = 0$

Given a quadratic function $q$ in the bi-form representation, $q(x) = b_0 + \sum_{0 \le i < j \le n}(b_{ij}^- x_{ij}^- + b_{ij}^+ x_{ij}^+)$, we set the associated graph $G_q$, whose vertices correspond to the indices $\{0, 1, \ldots, n\}$ of the variables, and whose edges correspond to those pairs $(i, j)$ representing a bi-term in $q$ involving the variables $x_i$ and $x_j$. An edge $(i, j)$ is called positive (negative) if the associated bi-term is positive $x_{ij}^+$ (negative $x_{ij}^-$) weighted by the positive coefficient $b_{ij}^+$ ($b_{ij}^-$) in $q$. In other words, $G_q$ is a weighted signed graph associated to the bi-form of $q$, defined by $G_q = (V, w, s)$ where $V = \{0, 1, \ldots, n\}$, weighted by $w$ and signed by $s$ such that :

$w_{ij} = w_{ji} = b_{ij}^+$ and $s_{ij} = s_{ji} = +1$ if the positive bi-term $x_{ij}^+$ is involved in $q$,

$w_{ij} = w_{ji} = b_{ij}^-$ and $s_{ij} = s_{ji} = -1$ if the negative bi-term $x_{ij}^-$ is involved in $q$,

$w_{ij} = w_{ji} = 0$ and $s_{ij} = s_{ji} = +1$ otherwise,

**Example 4:** Consider the quadratic function

$q(x) = -3x_1 + 12x_2 - x_3 + 3x_4 + 14x_5 - 10x_1x_2 + 12x_1x_3 - 6x_1x_5 - 14x_2x_3 + 4x_3x_4 - 10x_4x_5$.
The unique bi-form of $q$ is

$$q(x) = -13 + 5x^+_{01} + 6x^-_{05} + 5x^+_{12} + 6x^-_{13} + 3x^+_{15} + 7x^+_{23} + 2x^-_{34} + 5x^+_{45}$$

The weighted signed graph $G_q$ associated to the Bi-form $q$ is presented in the Figure 1. The positive edges are represented by solid line and the negative edges by dotted lines where $s_{05} = s_{13} = s_{34} = -1$ and $s_{ij} = 1$ for the other edges $(i, j)$.

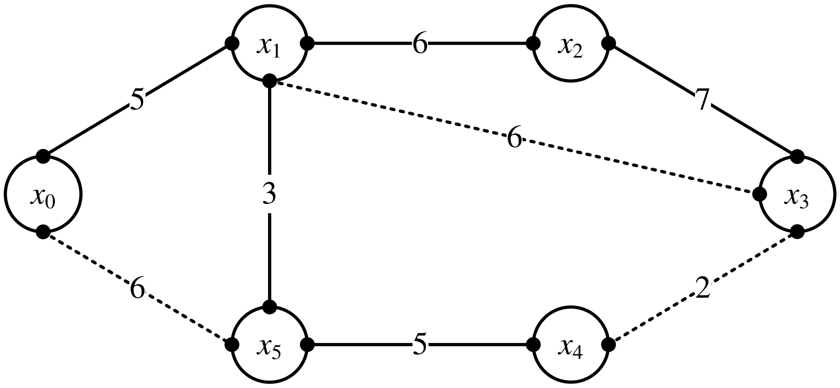


**Figure 1 : Weighted Signed Graph associated to the Bi-form**

Within the weighted signed graph $G_q$ associated to a given bi-form $q$, Boros, Hammer, and Sun (1989), show that substituting $x_i = x_j$ (or $x_i = \overline{x}_j$) into $q$ is equivalent to contracting the positive (or negative) edge $(i, j)$ of $G_q$. The contraction of the edge $(i, j)$ of $G_q$ yields a new weighted signed graph $G' = (V', w', s')$ defined by :

$$V' = V - \{j\}$$

$$w'_{lk} = w'_{kl} = \begin{cases} w_{lk} & \text{if } k, l \in V' - \{i\} \\ \left| s_{il} w_{il} + s_{ij} s_{jl} w_{jl} \right| & \text{if } k = i \text{ and } l \in V' - \{i\} \end{cases}$$

$$s'_{lk} = s'_{kl} = \begin{cases} s_{lk} & \text{if } k, l \in V' - \{i\} \\ +1 & \text{if } k = i, l \in V' - \{i\}, \text{and} s_{il} w_{il} + s_{ij} s_{jl} w_{jl} \geq 0 \\ -1 & \text{if } k = i, l \in V' - \{i\}, \text{and} s_{il} w_{il} + s_{ij} s_{jl} w_{jl} \leq 0 \end{cases}$$

The contraction operation generates the following constant

$$w'_{ij} = \sum_{k \in V' - \{i\} : s_{ik} s_{ij} s_{jk} = -1} \min(w_{ik}, w_{jk})$$

**DDT with Bi-form Procedure ($q$)**

Input : UQP $q$ in the bi-form representation.

Initialization : Compute the weighted signed graph $G_q = (V, w, s)$ associated to $q$. Let $b_0$ be the constant of $q$. Set the system $S = \emptyset$.

Devour: Find the largest weight $w_{i^*j^*}$ in the current graph $G_q$, i.e. $w_{i^*j^*} = \max\{ w_{ij} : i, j \in V\}$. If $s_{i^*j^*} = +1$ add $x^+_{i^*j^*} = 0$ to $S$ else add $x^-_{i^*j^*} = 0$ to $S$.

Digest-Tidy-up: Set $b_0 = b_0 + \sum_{j \in V - \{i^*\} : s_{\mu iJ} s_{\mu i \mu j} s_{j \mu J} = -1} \min(w_{i^*j}, w_{j^*j})$. Drop $j$ from vertex set $V = V - \{j\}$ and contract the edge $(i^*, j^*)$ by updating the weight and sign of the current graph as follows :

```
        For $j \in V - \{i^*\}$ do{
                $w_{i^*j} = w_{ji^*} = \left| s_{i^*j} w_{i^*j} + s_{i^*j^*} s_{j^*j} w_{j^*j} \right|$
                $s_{i^*j} = s_{ji^*} = \begin{cases} +1 & \text{if } s_{i^*j} w_{i^*j} + s_{i^*j^*} s_{j^*j} w_{j^*j} \geq 0 \\ -1 & \text{if } s_{i^*j} w_{i^*j} + s_{i^*j^*} s_{j^*j} w_{j^*j} \leq 0 \end{cases}$
        }
        If still there are edges with nonzero weight in $G_q$, then return to Devour.
Output: Solve the Boolean equations in $S$, and output $x$ and $b_0$.
```

To illustrate a step by step procedure of DDT with bi-form we applied it on the quadratic function of Example 1.

**Example 5:** The weighted signed graph $G_q$ associated to each current Bi-form $q$ is presented in the Figure 2.

*Iteration* 0: The equivalent biform of the quadratic function of Example 1 is
$q(x) = -13 + 7x^+_{01} + 4x^-_{04} + 3x^-_{05} + 6x^+_{12} + 4x^-_{14} + 2x^-_{23} + 5x^+_{24} + 3x^+_{34} + 4x^+_{45}$

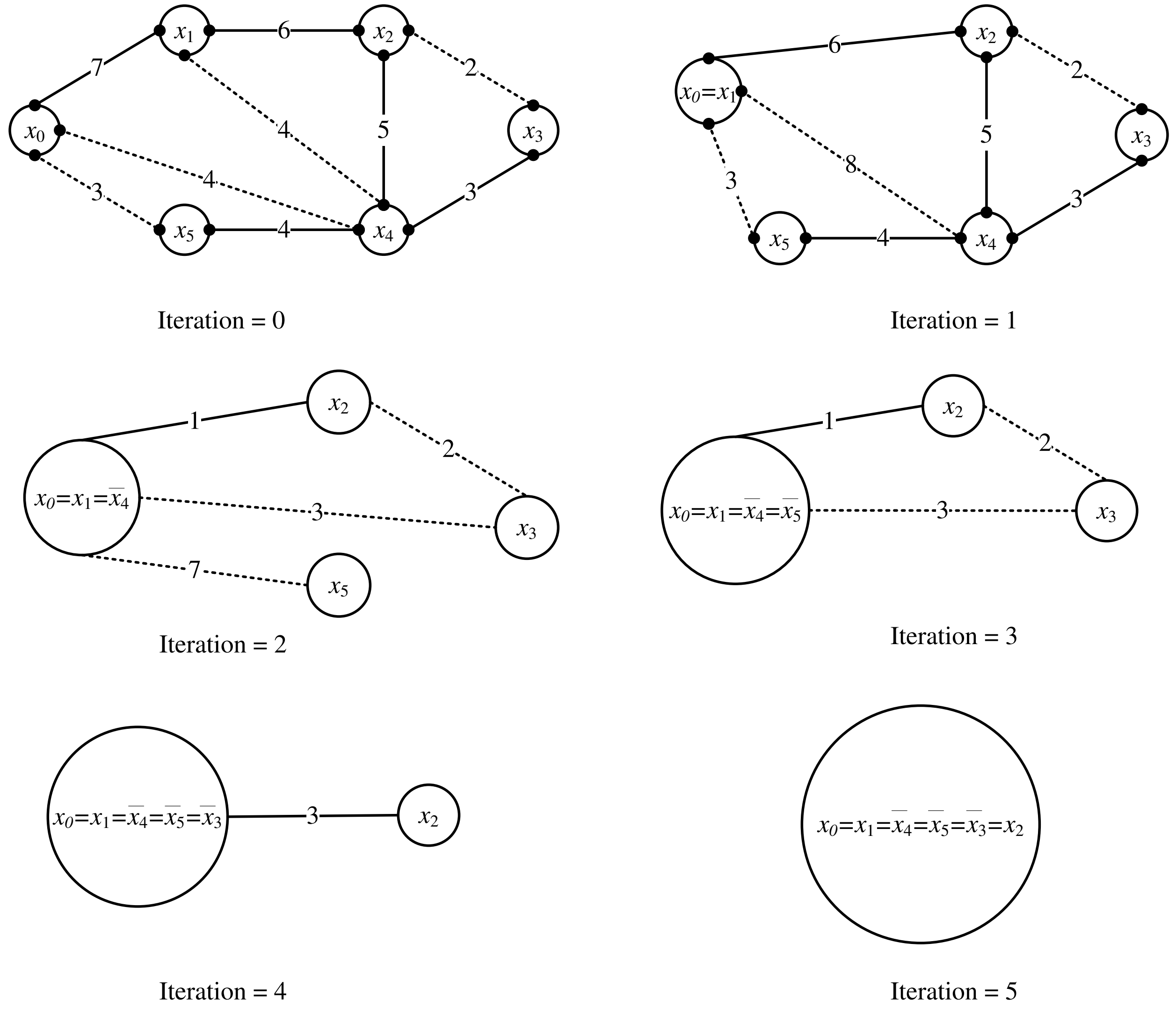
*Iteration* 1: First the largest weight $w_{i^*j^*} = 7$ in the current graph corresponds to the edge $(i^*, j^*) = (0, 1)$ is found. The contraction operation generates the constant $w'_{i^*j^*} = 0$, $q_0 = -13$ and we have $x_1 = x_0 = 1$. The current bi-form becomes
$q(x) = -13 + 6x^+_{02} + 8x^-_{04} + 3x^-_{05} + 2x^-_{23} + 5x^+_{24} + 3x^+_{34} + 4x^+_{45}$.

*Iteration* 2: Next the largest weight $w_{i^*j^*} = 8$ in the current graph corresponds to the edge $(i^*, j^*) = (0, 4)$ is found. The contraction operation generates the constant $w'_{i^*j^*} = 5$ and we have $\bar{x}_4 = x_0 = 1$ with $q_0 = -8$. So $x_4 = 0$ and the current bi-form becomes
$q(x) = -8 + x^+_{02} + 3x^-_{03} + 7x^-_{05} + 2x^-_{23}$.

*Iteration* 3: Then, we have $(i^*, j^*) = (0, 5)$ with $w_{i^*j^*} = 7$and $s_{i^*j^*} = -1$ therefore we get $\bar{x}_5 = x_0 = 1$ and $w'_{i^*j^*} = 0$ with $q_0 = -8$. So $x_5 = 0$ and the current bi-form becames
$q(x) = -8 + x^+_{02} + 3x^-_{03} + 2x^-_{23}$.

*Iteration* 4: We have $(i^*, j^*) = (0, 3)$ with $w_{i^*j^*} = 3$ and $s_{i^*j^*} = -1$ therefore we get $\bar{x}_3 = x_0 = 1$ and $w'_{i^*j^*} = 0$ with $q_0 = -8$. So $x_3 = 0$ and the current bi-form becomes
$q(x) = -8 + 3x^+_{02}$.

*Iteration* 5: At the end, we have $(i^*, j^*) = (0, 2)$ with $w_{i^*j^*} = 3$ and $s_{i^*j^*} = 1$ therefore we get $x_2 = x_0 = 1$ and $w'_{i^*j^*} = 0$ with $q_0 = -8$. So $x_2 = 1$ and the current bi-form becames $q(x) = -8$. Now the graph is empty so the algorithm terminates and return $x = (1, 1, 0, 0, 0)$ with $q(x) = -8$.



Iteration = 0

Iteration = 1

Iteration = 2

Iteration = 3

Iteration = 4

Iteration = 5

**Figure 2 : Weighted Signed Graphs corresponding to each step of DDT with Bi-form**

### 3. Devour Digest Tidy-up Heuristic with equality

The bi-form representation of a quadratic function can also be obtained by the following way. This new representation, called reprentation with equality, is obtained from the input quadratic function *q* given in the form of (1) by applying the following rule:

$$x_i x_j = \frac{x_i + x_j - (x_i - x_j)^2}{2} \tag{10}$$

By replacing $x_i x_j$ using the equation (9) and regrouping the terms, the expression of the quadratic function q(x) can be written as:

$$q(x) = c_0 + \sum_{i=1}^{n} c_{ii} x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} (x_i - x_j)^2 \tag{11-a}$$

where

$$c_0 = q_0 \tag{11-b}$$

$$c_{ii} = \frac{1}{2}\left(\sum_{j=1}^{n} q_{ij} - q_{ii}\right) \tag{11-c}$$

$$c_{ij} = -\frac{q_{ij}}{2} \tag{11-d}$$

Similarly to the standard DDT described in section ?, in the “devour” step of the DDT with equality, a term with the largest absolute value of coefficient is set to 0 or 1. If this largest coefficient is negative the term is set to 1 which implies that the variables involved in this term are set to 1. Precisely, at each iteration of this version of DDT, we find the largest absolute value of coefficient $c_{i^*j^*}$ in the current function $q$, i.e.

$$(i^*, j^*) = \operatorname{argmax}\{|c_{ij}| : (i, j) \in T\}$$

where $T = \{(i, j) : c_{ij} \neq 0\}$. The term associated to the largest coefficient is set to 0 if the sign of the largest coefficient is positive otherwise it is set to 1. Then the logical consequences of this assignment are derived and substituted into the current quadratic function. More specifically, we consider two cases. In case where $i^* = j^*$, then we set $x_{i^*} = \text{sign}(q_{i^*i^*})$. Then we substitute this assignment in to the current function $q$. By substituting this assignment and observing that

$$(x_{i^*} - x_j)^2 = (x_j - x_{i^*})^2 = (1-2x_{i^*})x_j + x_{i^*} \tag{12-a}$$

$$(x_{i^*} - x_{j^*})^2 = 0 \Leftrightarrow x_{i^*} = x_{j^*} \tag{12-b}$$

$$(x_{i^*} - x_{j^*})^2 = 1 \Leftrightarrow x_{i^*} = \overline{x}_{j^*} \tag{12-c}$$

The current quadratic function $q$ becomes

$$q(x) = c'_0 + \sum_{i=1}^{n} c'_{ii}\, x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c'_{ij}\, (x_i - x_j)^2 \tag{13-a}$$

where

$$c'_0 = c_0 + (Q^{i^*} e)x_{i^*} \tag{13-b}$$

$$c'_{ii} = c_{ii} + (1 - 2x_{i^*})c_{i^*i} \qquad \text{for } i \neq i^* \tag{13-c}$$

$$c_{i^*i} = c_{ii^*} = 0 \qquad \text{for all } i \tag{13-d}$$

In case where $i^* \neq j^*$, then we set $(x_{i^*} - x_{j^*})^2 = \text{sign}(q_{i^*j^*})$. Then we substitute this assignment in to the current function $q$. By substituting this assignment and observing that

$$(x_{j^*} - x_j)^2 = (1-2(x_{i^*} - x_{j^*})^2)(x_{i^*} - x_j)^2 + (x_{i^*} - x_{j^*})^2$$

The current quadratic function $q$ becomes

$$q(x) = c'_0 + \sum_{i=1}^{n} c'_{ii}\, x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c'_{ij}\, (x_i - x_j)^2 \tag{14-a}$$

where

$$c'_0 = c_0 + (Q^{j^*} e)(x_{i^*} - x_{j^*})^2 \tag{14-b}$$

$$c'_{i^*i} = c_{i^*i} + (1 - 2(x_{i^*} - x_{j^*})^2)c_{i^*i} \qquad \text{for } i \neq i^* \tag{14-c}$$

$$c_{j^*i} = c_{ij^*} = 0 \qquad \text{for all } i \tag{14-d}$$

Algorithme à décrire et à implémenter

## 4. Devour Digest Tidy-up Heuristic with One-Pass

Glover, Alidaee, Rego and Kochenberger (2002) suggest several alternative heuristics based on different ways of generating posiform representations. These variants evaluations schemes are slightly different than those used in the DDT procedure proposed by Boros. Glover et al. propose eight different ways of evaluating the contributions of the variables in the one-at-a-time assignments. The original DDT heuristic often results in setting several variables to 0 or 1 simultaneously after the assignment of a value to one variable. These assignments are triggered by giving a value to a literal that appeared in preceding pairs in the sequence that have been kept in the current system.
Different ways of evaluating the contributions of the variables in the one-at-a-time assignments lead to alternative ways of implementing the one-pass idea.

```
DDT with One-Pass Procedure($q$)
Input : UQP $q$ in a posiform.
Initialization : Compute $T = \{(u, v) : p_{uv} > 0, u,v \in L\}$.
Devour: Find the largest coefficient $p_{u*v*}$ in the current posiform $q$, set
                    $(u^*, v^*) = \text{argmax}\{ p_{uv} : (u, v) \in T\}$
Digest: If $u^* = v^*$ then set $u^* = 0$. Otherwise, Let
              $(u', v') = \text{argmax}\{q_{uv} : (u, v) \in T^{\neq} \text{ and } (\bar{u},\bar{u}) \text{ or } (\bar{v},\bar{v}) \in T^{=}\}$.
    If $(\bar{u},\bar{u}) \in T^{=}$ then set $v' = 0$; else if $(\bar{v},\bar{v}) \in T^{=}$ then set $u' = 0$;
    else $u^* = 0$.
Tidy-up: Substitute the assignment into $q$ and update $T$. If $T \neq \varnothing$ then return to Devour.
Output: the solution $x$.
```

## Local Search

Local search (LS) heuristic tries to improve an initial solution by searching in a neighborhood of the current solution for a better one until no further improvement can be made. LS progressively improves the initial solution by applying a series of local modifications (called also moves).

**Boros, Hammer *and* Tavares (2007)** present a family of LS heuristics for UQP and they analyze the effects of various parameters on the efficiency of these methods.
MERZ AND FREISLEBEN propose **greedy and LS heuristics for UQP**
In this section, we propose an efficient implementation of local search algorithms using r-flip moves. These heuristics are well suited for the incorporation into meta-heuristics such as Tabu search and evolutionary algorithms. The proposed heuristics—especially the *r-flipp* local search—offer a great potential for the incorporation in more sophisticated meta-heuristics.

- **Let $x \in \{0,1\}^n$**

$$N_k(x) = \left\{ y \in \{0,1\}^n : \sum_{j=1}^{n} |x_j - y_j| \le k \right\}$$

- **Stop = False**
- **While Not Stop{**
    - **Choose $x' \in N_k(x)$ such that $q(x') < q(x)$**

```
      — If x' exists then Set x = x'
      — Else Stop = True
      }
```

```
// x[0] = q(x) = xQ1x
int RL_V12_firstimprove(UPQ *Q, int *x)
{
for(h=1;h<=n;h++){
      Qx[h] = Q[1][h]*x[1];
      for(int j=2;j<=n;j++)   Qx[h] += Q[j][h]*x[j];
}
Qx = Q*x;
iter = delta = 0;
Best = q(x);
eval:
Best += delta;
for (h=1; h<= n;h++){
      delta = Q[h][h] + 2(1 - 2x[h])*Qx[h];
      if (delta < 0){
            for(j=1;j<=n;j++) Qx[j] += (1 - 2x[h])*Q[h][j];
            x[h] = 1 - x[h];
            goto eval;
      }
}
for (h=1; h < n; h++){
      for (k=h+1; k <= n; k++){
            delta = Q[h][h] + (2(1 - 2x[h])*Qx[h]) + Q[k][k] + 2((1 -
2x[k])*Qx[k] + (1 - 2x[h])* (1 - 2x[k])*Q[h][k]);
            if (delta < 0){
                  for(j=1; j<=n; j++){
                        Qx[j] += (1-2x[h])*Q[h][j] + (1-2x[k])*Q[k][j];
                  }
                  x[h] = 1 - x[h];
                  x[k] = 1 - x[k];
                  goto eval;
            }
      }
}
x[0] = Best;
return iter;
}
```

Recently Glover and Hao (2009b) propose a method for efficiently evaluating 2-flip moves in search methods for UQP. They extend thier method for efficiently evaluating 1-flip moves described in Glover and Hao (2009b).

There exist several ways to establish combined neighborhoods from 1-flip and 2-flip moves see for example VNS, TS.

Lü, Glover, and Hao (2009) propose a Tabu Search (Glover, Laguna, 1997, Glover, Hanafi, 2002) and Iterated Local Search (Lourenco, Martin, Stützle, (2003) that combines neighborhoods with 1-flip and 2-flip moves.

2-flip moves that simultaneously change the values of two 0-1 variables in search methods for UQP.
1-flip moves that change the value of a single 0-1 variable for the UQP problem.

Starting from the 1-flip move, let x and x′ represent two binary solutions where x′ is obtained from x by flipping the value of a single variable from $x_i$ to $x_i' = 1 - x_i$, so we have
$x' = x + (1 - 2x_i)e^i$.
Define $q(x) = q_o + xQx$. Then the objective function change produced by flipping $x_i$, given by
$\Delta_i = q(x') - q(x)$.
By developing and observing that $e^iQx = Q^ix = (Qx)_i$ and $(1 - 2x_i)^2 = 1$, $\Delta_i$ can be expressed as

$$\begin{aligned}
\Delta_i &= (x + (1 - 2x_i)e^i)Q(x + (1 - 2x_i)e^i) - xQx \\
\Delta_i &= (1 - 2x_i)e^iQx + (1 - 2x_i)e^iQx + (1 - 2x_i)e^iQ(1 - 2x_i)e^i) \\
\Delta_i &= 2(1 - 2x_i)Q^ix + (1 - 2x_i)^2 q_{ii} \\
\Delta_i &= 2(1 - 2x_i)(Qx)_i + q_{ii}
\end{aligned}$$

where the notation $Q^i$ refers to column i of matrix Q.
In the case of a 2-flip neighborhood, we are interested in the move from solution x to the neighbood solution x′ that results by flipping 2 variables, $x_i' = 1 - x_i$ and $x_j' = 1 - x_j$. We have
$x' = x + (1 - 2x_i)e^i + (1 - 2x_j)e^j$.
We will refer to the objective function change by $\Delta_{ij} = q(x) - q(x')$. By developing and observing also that $q_{ij} = q_{ji} = e^iQe^j = e^jQe^i$, $\Delta_{ij}$ can be expressed as

$$\begin{aligned}
\Delta_{ij} &= (x + (1 - 2x_i)e^i + (1 - 2x_j)e^j)Q(x + (1 - 2x_i)e^i + (1 - 2x_j)e^j) - xQx \\
\Delta_{ij} &= xQx + (1 - 2x_i)e^iQx + (1 - 2x_j)e^jQx + \\
&\quad xQ(1 - 2x_i)e^i + (1 - 2x_i)e^iQ(1 - 2x_i)e^i + (1 - 2x_j)e^jQ(1 - 2x_i)e^i + \\
&\quad xQ(1 - 2x_j)e^j + (1 - 2x_i)e^iQ(1 - 2x_j)e^j + (1 - 2x_j)e^jQ(1 - 2x_j)e^j) - xQx \\
\Delta_{ij} &= (1 - 2x_i)(Qx)_i + (1 - 2x_j)(Qx)_j + (1 - 2x_i)(Qx)_i + q_{ii} + (1 - 2x_j)(1 - 2x_i)q_{ji} + \\
&\quad (1 - 2x_j)(Qx)_j + (1 - 2x_i)(1 - 2x_j)q_{ij} + q_{jj} \\
\Delta_{ij} &= 2(1 - 2x_i)(Qx)_i + 2(1 - 2x_j)(Qx)_j + 2(1 - 2x_i)(1 - 2x_j)q_{ij} + q_{ii} + q_{jj}
\end{aligned}$$

Equivalently,

$$\Delta_{ij} = \Delta_i + \Delta_j + 2(1 - 2x_i)(1 - 2x_j)q_{ij}$$

F. Glover and M. Laguna. Tabu Search. Kluwer Academic, Boston, 1997.
H. R. Lourenco, O. Martin, T. Stäutzle, Iterated local search. Handbook of Meta-heuristics, Springer-Verlag, Berlin Heidelberg, 2003.
Lü, Z., F., Glover, and J.K., Hao (2009) "Neighborhood Combination for Unconstrained Binary Quadratic Programming," Proceedings of the 8th Metaheuristic International Conference (*MIC 2009*), Hamburg, Germany, 13-16 July, *2009*.
F. Glover and J.K. Hao. Efficient evaluations for solving large 0-1 unconstrained quadratic optimization problems. International Journal of Metaheuristics (2009) (To appear).

F. Glover and J.K. Hao. Fast 2-fip move evaluations for binary unconstrained quadratic optimization problems. Working paper, LERIA, Université d'Angers (2009).

## 1. Computational results

In this Section, we show computational results of the aforementioned versions of DDT heuristics. The variants of DDT heuristics were implemented in C++, compiled using the Microsoft Windows 32-bit C/C++ Optimizing Compiler (version 12) for 80×86, and linked with the Microsoft Incremental Linker (version 6). The computer used for testing has a Xeon (TM) CPU 3.06 GHz, 3.5 GB of RAM and has installed the Windows XP Professional (version 2002) operating system.

*Benchmark datasets*

Several authors have reported in the past computational results about UQP. The benchmark test problems used in this paper and in numerous other studies of UQP (e.g., Pardalos and Rodgers (1990), Amini et al. (1999), Beasley (1998), Glover et al. (2002), Glover et al. (1998a), Merz and Freisleben (1999) and Merz and Freisleben (2002)) were mostly taken from the Internet. Only the largest problem instances available there were considered.

Experiments are carried out on the set of the 53 largest instances with 1000 up to 6000 variables. We validate our heuristics on three sets of available instances of UQP. The first set is composed by 15 instances with n = 1000, the problems in this group *GK and KG* were taken from the Hearin Center for Enterprise Science website,2. The second set is composed by 20 instances with n in {1000, 2500} taken from the *OR-Library* website1. The last set is composed by 18 instances varying in size from 3000 to 6000 variables generated by PALUBECKIS and can be found in the wibeset3.

The basic parameters and references of these problems are shown in Table 1. These problems vary considerably in size, density, and in the characteristics of their Q matrices. In Table 1, $n$ denotes the number of variables in a problem, let $d$ be the "density" of the problem, i.e., the ratio of the number of nonzero coefficients of quadratic terms in the function and n(n-1)/2. Let $l-$ and $l+$ be respectively the minimum and the maximum of the coefficients of the linear terms in (1), and let $q-$ and $q+$ be equal to *one half* of the minimum, respectively of the maximum, of the coefficients of the quadratic terms in (1). For example, all the instances, named OR-n-1, . . . , OR-n-10 have density 10% and all nonzero coefficients of the objective function are drawn uniformly at random from the interval [−100, 100].

| Problem Type | Source | *n* | # pbs | D | Main Diagonal | Off Diagonal |
|---|---|---|---|---|---|---|
| G1 | Glover et al. [12] (1) | 1000 | 10 | 10 to 100 | [-75,+75] | [-50,+50] |
| G2 | Kochenberger et al. (1) | 1000 | 5 | 10 to 100 | [-99,0] | [0,+50] |
| OR | OR-Library (2) | 1000 and 2500 | 20 | 10 | [-100,+100] | [-100,+100] |
| GP | PALUBECKIS (3) | 3000 to 6000 | 18 | 50 to 100 | [-100,+100] | [-100,+100] |

**Table 1.** Characteristic of Benchmark test problems for UQP

1 Beasley, J.E. (11/11/2003). OR-Library: Unconstrained binary quadratic programming (Beasley, 1990 and Badics, 1996). *http://mscmga.ms.ic.ac.uk/jeb/orlib/bqpinfo.html.*
2 Hearin Center for Enterprise Science. (11/11/2003). Benchmarks for unconstrained binary quadratic problems. *http://hces.bus.olemiss.edu/tools.html.*
3. http://www.soften.ktu.lt/~gintaras/ubqop_its.html.

Let us denote by *q0** the *best known value* of the quadratic function *q*. The list of all best values for the test problems used in this study from the pseudo–Boolean optimization website: *http://rutcor.rutgers.edu/~pbo*, and recent papers on the UQP cited in this paper.

*F. Glover, G.A. Kochenberger, B. Alidaee, and M. Amini, (1998), Tabu search with critical event memory: An enhanced application for binary quadratic programs. In: Meta--heuristics-Advances and trends in local search paradigms for optimization, pp. 83-109.*

*F. Glover, G.A. Kochenberger, B. Alidaee, and M. Amini, (1998), Tabu search with critical event memory: An enhanced application for binary quadratic programs. In: Meta--heuristics-Advances and trends in local search paradigms for optimization, pp. 83-109.*

*P. Merz and B. Freisleben, (1999), "Genetic algorithms for binary quadratic programming", in Proceedings of the 1999 international Genetic and Evolutionary Computation Conference (GECCO'99), Morgan Kauffmann, 417-424.*

*J. E. Beasley, (1998), "Heuristic algorithms for the unconstrained binary quadratic programming problem", Technical Report, Management School, Imperial College, London, UK.*

*G. Palubeckis, (11-24-2003), http://www.soften.ktu.lt/~gintaras/*

P. Pardalos and G. P. Rodgers, (1990), "Computational aspects of a branch and bound algorithm for quadratic 0-1 programming", *Computing* **45** 131-144.

DDT0 : DDT Standard
DDT1 : DDT with Signed Graph
DDT2 : DDT with posiform
DDT4 : One pass A2
DDT5 : DDT with Negaform

| Pb | q* | DDT0 | DDT0* | DDT1 | DDT1* | DDT2 | DDT2* | DDT4 | DDT4* | DDT5 | DDT5* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| G2a | 4929 | 87,69 | 91,01 | 76,51 | 85,35 | 87,69 | 91,01 | 40,09 | 83,82 | 39,15 | 83,01 |
| G2b | 2050 | 82,56 | 83,18 | 71,25 | 84,78 | 82,56 | 83,18 | 39,77 | 74,88 | 42,22 | 82,33 |
| G2c | 1241 | 78,30 | 78,30 | 66,04 | 87,74 | 78,30 | 78,30 | 41,59 | 75,63 | 33,25 | 83,33 |
| G2d | 843 | 83,26 | 83,26 | 65,01 | 80,37 | 83,26 | 83,26 | 30,37 | 82,10 | 31,64 | 80,60 |
| G2e | 452 | 88,05 | 93,58 | 59,73 | 91,15 | 88,05 | 93,58 | 31,86 | 84,73 | 58,19 | 75,22 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OR1000_1 | 371438 | 62,62 | 82,79 | 83,71 | 84,02 | 70,78 | 83,79 | 70,72 | 83,66 | 4,32 | 83,78 |
| OR1000_2 | 354932 | 61,64 | 90,31 | 89,88 | 90,19 | 78,10 | 89,85 | 78,10 | 89,85 | 5,50 | 89,62 |
| OR1000_3 | 371226 | 73,42 | 89,04 | 88,86 | 89,56 | 79,70 | 89,69 | 79,70 | 89,69 | 4,08 | 89,15 |
| OR1000_4 | 370560 | 63,37 | 84,64 | 84,45 | 85,57 | 71,06 | 85,78 | 71,05 | 85,35 | 4,12 | 84,86 |
| OR1000_5 | 352736 | 71,69 | 88,00 | 87,33 | 87,96 | 73,93 | 87,36 | 73,90 | 87,74 | 4,28 | 87,71 |
| OR1000_6 | 359452 | 72,28 | 93,38 | 93,34 | 93,79 | 82,45 | 93,78 | 82,45 | 93,37 | 4,97 | 93,01 |
| OR1000_7 | 370999 | 68,66 | 86,28 | 86,45 | 87,37 | 73,00 | 87,28 | 73,01 | 87,13 | 4,08 | 86,35 |
| OR1000_8 | 351836 | 70,23 | 88,61 | 88,46 | 89,16 | 73,48 | 89,43 | 73,38 | 88,50 | 3,90 | 89,13 |
| OR1000_9 | 348732 | 66,91 | 91,08 | 91,62 | 92,59 | 78,11 | 92,10 | 78,11 | 92,20 | 5,38 | 92,17 |
| OR1000_10 | 351415 | 63,60 | 91,77 | 92,18 | 92,69 | 78,21 | 92,26 | 78,13 | 91,96 | 4,08 | 91,56 |
| OR2500_1 | 1515944 | 68,50 | 81,27 | 80,94 | 81,69 | 58,28 | 81,53 | 58,27 | 81,63 | 1,19 | 81,76 |
| OR2500_2 | 1471392 | 64,86 | 81,30 | 80,49 | 81,15 | 59,29 | 81,08 | 59,28 | 81,04 | 0,98 | 81,08 |
| OR2500_3 | 1414192 | 73,60 | 90,68 | 90,51 | 91,49 | 68,60 | 91,05 | 68,60 | 90,91 | 1,28 | 90,77 |
| OR2500_4 | 1507701 | 63,19 | 81,87 | 81,91 | 82,43 | 58,79 | 82,16 | 58,78 | 82,32 | 1,41 | 81,59 |
| OR2500_5 | 1491816 | 65,81 | 81,40 | 80,87 | 81,43 | 56,90 | 81,45 | 56,89 | 81,35 | 1,44 | 80,82 |
| OR2500_6 | 1469162 | 65,06 | 83,34 | 82,68 | 83,36 | 60,01 | 83,06 | 60,01 | 83,06 | 0,88 | 82,85 |
| OR2500_7 | 1479040 | 67,66 | 82,88 | 82,93 | 83,87 | 59,05 | 83,23 | 59,05 | 83,36 | 1,29 | 83,11 |
| OR2500_8 | 1484199 | 68,36 | 84,64 | 84,27 | 84,99 | 60,22 | 84,70 | 60,19 | 84,76 | 1,10 | 84,31 |
| OR2500_9 | 1482413 | 67,26 | 82,75 | 82,85 | 83,46 | 60,21 | 82,99 | 60,20 | 83,23 | 1,46 | 82,89 |
| OR2500_10 | 1483355 | 65,52 | 83,02 | 83,12 | 83,81 | 58,49 | 83,53 | 58,47 | 83,06 | 0,96 | 83,29 |
| GP_3000_01 | 3931583 | 80,76 | 100,02 | 99,66 | 100,26 | 53,13 | 99,96 | 53,13 | 100,05 | 0,13 | 99,85 |
| GP_3000_02 | 5193073 | 80,22 | 97,94 | 97,58 | 98,07 | 45,72 | 97,74 | 45,72 | 97,74 | 0,05 | 98,05 |
| GP_3000_03 | 5111533 | 82,43 | 98,70 | 98,53 | 99,24 | 47,98 | 99,04 | 47,98 | 99,04 | 0,03 | 98,93 |
| GP_3000_04 | 5761822 | 79,79 | 96,71 | 96,25 | 96,97 | 41,81 | 96,68 | 41,81 | 96,68 | 0,06 | 96,89 |
| GP_3000_05 | 5675625 | 82,75 | 100,67 | 100,31 | 101,18 | 47,40 | 100,70 | 47,40 | 100,70 | 0,02 | 100,68 |
| GP_4000_01 | 6181830 | 82,07 | 99,24 | 98,93 | 99,68 | 48,36 | 99,48 | 48,36 | 99,48 | 0,12 | 99,36 |
| GP_4000_02 | 7801355 | 82,92 | 100,87 | 100,51 | 101,26 | 46,49 | 101,20 | 46,49 | 101,20 | 0,05 | 101,01 |
| GP_4000_03 | 7741685 | 82,28 | 99,86 | 99,72 | 100,33 | 43,32 | 100,28 | 43,32 | 100,28 | 0,02 | 99,93 |
| GP_4000_04 | 8711822 | 82,84 | 100,58 | 100,15 | 101,08 | 44,44 | 100,78 | 44,44 | 100,62 | 0,02 | 100,55 |
| GP_4000_05 | 8908979 | 77,32 | 94,36 | 94,11 | 94,84 | 34,63 | 94,51 | 34,63 | 94,51 | 0,03 | 94,34 |
| GP_5000_01 | 8559680 | 79,24 | 98,02 | 97,55 | 98,46 | 45,20 | 98,04 | 45,20 | 98,04 | 0,07 | 98,10 |
| GP_5000_02 | 10836019 | 79,49 | 97,43 | 97,09 | 97,67 | 37,18 | 97,43 | 37,18 | 97,58 | 0,03 | 97,28 |
| GP_5000_03 | 10489137 | 87,11 | 106,02 | 105,79 | 106,38 | 48,22 | 106,24 | 48,22 | 106,18 | 0,04 | 106,34 |
| GP_5000_04 | 12252318 | 80,11 | 99,49 | 99,20 | 99,85 | 39,51 | 99,65 | 39,51 | 99,65 | 0,02 | 99,62 |
| GP_5000_05 | 12731803 | 73,99 | 93,59 | 92,92 | 93,71 | 31,03 | 93,65 | 31,03 | 93,65 | 0,02 | 93,41 |
| GP_6000_01 | 11384976 | 78,62 | 98,70 | 98,34 | 99,13 | 40,76 | 98,84 | 40,76 | 98,84 | 0,04 | 98,88 |
| GP_6000_02 | 14333855 | 85,48 | 102,26 | 101,68 | 102,34 | 41,64 | 101,89 | 41,64 | 101,89 | 0,02 | 102,11 |
| GP_6000_03 | 16132915 | 80,23 | 98,85 | 98,48 | 99,22 | 36,84 | 99,02 | 36,84 | 99,02 | 0,01 | 98,81 |

**Table 2 :** Solution quality of variants DDT heuristics for the UQP.

| Pb | CPU0 | CPU0* | CPU1 | CPU1* | CPU2 | CPU2* | CPU4 | CPU4* | CPU5 | CPU5* |
|---|---|---|---|---|---|---|---|---|---|---|
| G2a | 0,28 | 0,02 | 1,06 | 0,03 | 0,30 | 0,02 | 0,09 | 0,03 | 0,08 | 0,02 |
| G2b | 1,56 | 0,02 | 1,06 | 0,02 | 1,59 | 0,02 | 0,23 | 0,03 | 0,23 | 0,03 |
| G2c | 5,44 | 0,02 | 1,05 | 0,03 | 5,31 | 0,02 | 0,44 | 0,03 | 0,44 | 0,03 |
| G2d | 9,25 | 0,02 | 1,05 | 0,02 | 9,13 | 0,03 | 0,66 | 0,02 | 0,66 | 0,03 |
| G2e | 57,52 | 0,03 | 1,05 | 0,02 | 55,52 | 0,03 | 1,00 | 0,03 | 1,00 | 0,03 |
| OR1000_1 | 0,08 | 0,03 | 1,06 | 0,02 | 0,08 | 0,02 | 0,08 | 0,03 | 0,08 | 0,03 |
| OR1000_2 | 0,08 | 0,03 | 1,06 | 0,02 | 0,08 | 0,02 | 0,08 | 0,02 | 0,08 | 0,05 |
| OR1000_3 | 0,06 | 0,05 | 1,05 | 0,02 | 0,08 | 0,02 | 0,08 | 0,02 | 0,08 | 0,03 |
| OR1000_4 | 0,08 | 0,03 | 1,05 | 0,02 | 0,08 | 0,03 | 0,09 | 0,02 | 0,08 | 0,02 |
| OR1000_5 | 0,06 | 0,05 | 1,05 | 0,03 | 0,09 | 0,02 | 0,08 | 0,03 | 0,08 | 0,05 |
| OR1000_6 | 0,08 | 0,03 | 1,06 | 0,02 | 0,08 | 0,03 | 0,09 | 0,02 | 0,08 | 0,03 |
| OR1000_7 | 0,08 | 0,02 | 1,05 | 0,02 | 0,08 | 0,03 | 0,08 | 0,02 | 0,08 | 0,03 |
| OR1000_8 | 0,06 | 0,03 | 1,05 | 0,02 | 0,08 | 0,05 | 0,09 | 0,02 | 0,08 | 0,03 |
| OR1000_9 | 0,08 | 0,03 | 1,06 | 0,03 | 0,08 | 0,02 | 0,08 | 0,03 | 0,08 | 0,03 |
| OR1000_10 | 0,08 | 0,03 | 1,06 | 0,03 | 0,08 | 0,03 | 0,08 | 0,02 | 0,08 | 0,02 |
| OR2500_1 | 1,30 | 0,16 | 14,45 | 0,13 | 1,47 | 0,16 | 1,47 | 0,22 | 1,47 | 0,27 |
| OR2500_2 | 1,31 | 0,44 | 14,23 | 0,17 | 1,47 | 0,14 | 1,45 | 0,17 | 1,45 | 0,20 |
| OR2500_3 | 1,33 | 0,23 | 14,20 | 0,27 | 1,47 | 0,22 | 1,45 | 0,22 | 1,45 | 0,19 |
| OR2500_4 | 1,23 | 0,16 | 14,19 | 0,22 | 1,45 | 0,19 | 1,45 | 0,20 | 1,47 | 0,28 |
| OR2500_5 | 1,34 | 0,23 | 14,39 | 0,17 | 1,45 | 0,31 | 1,45 | 0,27 | 1,45 | 0,25 |
| OR2500_6 | 1,23 | 0,20 | 14,19 | 0,19 | 1,45 | 0,17 | 1,45 | 0,16 | 1,45 | 0,23 |
| OR2500_7 | 1,27 | 0,14 | 14,22 | 0,16 | 1,47 | 0,22 | 1,45 | 0,16 | 1,45 | 0,33 |
| OR2500_8 | 1,25 | 0,16 | 14,22 | 0,24 | 1,45 | 0,14 | 1,45 | 0,11 | 1,45 | 0,28 |
| OR2500_9 | 1,27 | 0,16 | 14,22 | 0,19 | 1,47 | 0,13 | 1,45 | 0,19 | 1,45 | 0,20 |
| OR2500_10 | 1,25 | 0,20 | 14,44 | 0,24 | 1,45 | 0,25 | 1,45 | 0,16 | 1,45 | 0,45 |
| GP_3000_01 | 13,17 | 0,28 | 24,09 | 0,20 | 14,69 | 0,25 | 14,55 | 0,31 | 14,67 | 0,44 |
| GP_3000_02 | 21,23 | 0,28 | 24,05 | 0,20 | 23,50 | 0,27 | 23,61 | 0,27 | 23,78 | 0,47 |
| GP_3000_03 | 21,33 | 0,25 | 24,19 | 0,22 | 23,52 | 0,28 | 23,38 | 0,28 | 23,49 | 0,44 |
| GP_3000_04 | 26,52 | 0,27 | 24,11 | 0,27 | 29,63 | 0,36 | 29,52 | 0,36 | 29,66 | 0,47 |
| GP_3000_05 | 26,84 | 0,27 | 24,31 | 0,23 | 29,36 | 0,30 | 29,25 | 0,30 | 29,47 | 0,47 |
| GP_4000_01 | 31,39 | 0,49 | 56,02 | 0,38 | 35,03 | 0,48 | 34,88 | 0,49 | 34,95 | 0,84 |
| GP_4000_02 | 50,47 | 0,56 | 56,27 | 0,41 | 55,88 | 0,56 | 55,63 | 0,56 | 55,84 | 0,89 |
| GP_4000_03 | 50,66 | 0,52 | 55,95 | 0,41 | 55,89 | 0,61 | 55,69 | 0,66 | 55,88 | 0,91 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| GP_4000_04 | 63,20 | 0,50 | 55,97 | 0,42 | 69,58 | 0,58 | 69,58 | 0,56 | 70,02 | 0,95 |
| GP_4000_05 | 63,02 | 0,64 | 55,74 | 0,45 | 69,74 | 0,61 | 69,48 | 0,61 | 70,13 | 0,92 |
| GP_5000_01 | 61,55 | 0,77 | 109,89 | 0,75 | 68,00 | 0,88 | 67,89 | 0,88 | 68,25 | 1,50 |
| GP_5000_02 | 98,69 | 0,80 | 110,05 | 0,88 | 108,97 | 0,91 | 108,98 | 1,25 | 109,52 | 1,66 |
| GP_5000_03 | 98,50 | 0,81 | 109,97 | 0,64 | 109,11 | 0,91 | 108,80 | 0,97 | 109,53 | 1,75 |
| GP_5000_04 | 123,36 | 0,86 | 110,05 | 0,67 | 136,08 | 1,22 | 136,05 | 1,22 | 136,78 | 1,58 |
| GP_5000_05 | 123,08 | 0,92 | 109,84 | 0,83 | 136,09 | 1,06 | 136,13 | 1,06 | 136,95 | 1,66 |
| GP_6000_01 | 106,56 | 1,27 | 184,63 | 0,88 | 117,42 | 1,53 | 117,28 | 1,53 | 117,92 | 2,50 |
| GP_6000_02 | 170,58 | 1,48 | 184,72 | 1,22 | 188,34 | 1,41 | 187,92 | 1,41 | 188,39 | 2,50 |
| GP_6000_03 | 212,81 | 1,56 | 184,53 | 1,30 | 235,38 | 1,83 | 235,59 | 1,83 | 236,03 | 2,47 |

**Table 3 :** Running times in seconds of variants DDT heuristics for the UQP.

## Conclusions

- If more than one literal has the same max value then ties can be broken using different strategies. This gives other versions of DTT.
- The various solutions generated can also serve as starting points for more advanced methods.
- Efficiently evaluating moves that complement values of 0–1 variables in local search can exploited.
- Those versions of DTT proposed here can be exploited in scatter search by employing adaptive memory ideas.

## Acknowledgement

## References

**One-pass Heuristics for Large-scale Unconstrained Binary Quadratic Problems , Glover, Alidaee, Rego and Kochenberger, 2002**

- Adams, W.P., R.J. Forrester, F.W. Glover, Comparisons and enhancement strategies for linearizing mixed 0–1 quadratic programs, Discrete Optimization 1 (2) (2004) 99–120.
- Alidaee, B, G. Kochenberger, and A. Ahmadian, “0-1 Quadratic Programming Approach for the Optimal Solution of Two Scheduling Problems,” International Journal of Systems Science, 25, 401-408, 1994.
- Alkhamis, T. M., M. Hasan, and M. A. Ahmed, “Simulated Annealing for the Unconstrained Binary Quadratic Pseudo-Boolean Function,” European Journal of Operational Research, 108, (1998), 641-652.

- Amini,, M., B. Alidaee, G. Kochenberger, “A Scatter Search Approach to Unconstrained Quadratic Binary Programs,” To appear in New Methods in Optimization, Corne, Dorigo, and Glover, eds., McGraw-Hill, 1999.
- areas (Springer-Verlag, Berlin, Heidelberg, New York, 1968).
- Voss, S. Martello, I. Osman, and C. Roucairol. Kluwer Academic Publisher, Boston, 1999.
- Balas E., Extension de l’algorithme additif à la programmation en nombres entiers et à la programmation non linéaire, Comptes rendus de l’Académie des Sciences (Paris), 1964.
- Beasley, J. E.,”Heuristic Algorithms for the Unconstrained Binary Quadratic Programming Problem, Working Paper, Imperial College, 1999.
- Boros E., P. L. Hammer, R. Sun, G. Tavares (2008). A max-flow approach to improved lower bounds for quadratic unconstrained binary optimization (QUBO). Discrete Optimization 5, pp 501–529.
- Boros E., P.L. Hammer, A max-flow approach to improved roof-duality in quadratic 0-1 minimization, Research Report 15, Rutgers Center for Operations Research, Rutgers University, New Brunswick, NJ, 1989.
- Boros E., P.L. Hammer, Pseudo-boolean optimization, Discrete Applied Mathematics 123 (2002) 155–225.
- Boros, E, P. Hammer, and X, Sun, “The DDT Method for Quadratic 0-1 Minimization,” RUTCOR Research Center, RRR 39-89, 1989.
- Chartaire, P, and A. Sutter, “A Decomposition Method for Quadratic Zero-One Programming,” Management Science, 41:4, 704-712, 1994.
- Eranda C¸ ela, Bettina Klinz, Christophe Meyer, Polynomially Solvable Cases of the Constant Rank Unconstrained Quadratic 0-1 Programming Problem. Les Cahiers du GERAD, G–2006–36, 2006 GERAD
- Fortet R., Applications de l’algèbre de Boole en recherche opérationelle, Rev. Française Informat, Recherche Opérationnelle 4 (1960) 17–26.
- Fortet, R, (1959). “ L’algèbre de Boole et ses Applications en Recherche Operationnelle, Cahiere du Centre d’Etudes de Recheche Operationnelle, 1:4, 5-36.
- Gallo, G, P. Hammer, and B. Simeone, “Quadratic Knapsack Problems,” Mathematical Programming, 12, 132-149, 1980.
- Glover F., An improved MIP formulation for products of discrete and continuous variables, Journal of Information and Optimization Sciences 5 (1) (1984) 469–471.
- Glover F., E. Woolsey, Converting the 0–1 polynomial programming problem to a 0–1 linear program, Operations Research 22 (1974) 180–182.
- Glover F., Improved linear integer programming formulations of nonlinear integer problems, Management Science 22 (1975) 455–460.
- Glover, F, and M. Laguna, “Tabu Search,” Kluwer Academic Publishers, 1997.
- Glover, F, D. Klingman, and Phillips, N, “Network Models in Optimization and Their Applications in Practice,” John-Wiley, New York, 1992.
- Glover, F., G. Kochenberger, B. Alidaee, and M. Amini, “ Unconstrained Quadratic Binary Program approach to Quadratic Knapsack Problems,” Working paper, Hearin Center for Enterprise Science, University of Mississippi, 1999.
- Glover, F., G. Kochenberger, B. Alidaee, and M.M. Amini, “Tabu with Search Critical Event Memory: An Enhanced Application for Binary Quadratic Programs,” In: Meta- Heuristics: Advances and Trends in Local Search Paradigms for Optimization, (Eds.) S.
- Glover, F., G. Kochenberger., and B. Alidaee, “Adaptive Memory Tabu Search for Binary Quadratic Programs,” Management Science, 44:3, 336-345, 1998.

- Glover, F., M. Amini, G. Kochenberger and B. Alidaee, “A New Evolutionary Metaheuristic for the Unconstrained Binary Quadratic Programming: A Case Study of the Scatter Search,” School of Business, University of Colorado, Boulder, September 1999.
- Goldman A.J., Linearization in 0–1 variables: a clarification, Operations Research 31 (5) (1983) 946–947.
- Gueye S., P. Michelon, “Miniaturized” linearizations for quadratic 0–1 problems, Annals of Operations Research 140 (2005)
- Hammer P.L., P. Hansen, B. Simeone, Roof duality, complementation and persistency in quadratic 0–1 optimization, Mathematical Programming, 28 (1984) 121–155.
- Hammer, P, “Some Network Flow problems Solved with Pseudo-Boolean programming,” Operations Research, 13, 388-399, 1965.
- Hammer, P, E. Boros, and X. Sun, “On Quadratic Unconstrained Binary Optimization,” INFORMS National Meeting, Seattle, October, 1998.
- Hammer, P., and S. Rudeanu, Boolean Methods in Operations Research, Springer- Verlag, New York, 1968.
- Hammer, P.L. and S. Rudeanu. Boolean methods in operations research and related areas (Springer-Verlag, Berlin, Heidelberg, New York, 1968).
- Hammer, P.L. and S. Rudeanu. Boolean methods in operations research and related
- Hansen and Meyer, 2009. Improved compact linearizations for the unconstrained quadratic 0–1 minimization problem,. Discrete Applied Mathematics 157 (2009) 1267–1290
- Hansen, P, B. Jaumard., and V. Mathon, “Constrained Nonlinear 0-1 Programming,” INFORMS Journal on Computing, 5:2, 97-119, 1993.
- Harary, F, “On the Notion of balanced of a signed Graph,” Michigan Mathematical Journal, 2, 143-146, 1953/54.
- Katayama, K., M. Tani, and H. Narihisa, “Solving Large Binary Quadratic Programming Problems by an Effective Genetic Local Search Algorithm,” In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO’00). Morgan Kaufmann, 2000. (to appear).
- Kochenberger, G. , B. Alidaee, and M. Amini, “Applications of the Unconstrained Quadratic Binary Program,” University of Colorado Working Paper, 1998.
- Krarup, J, and A. Pruzan, “Computer Aided Layout Design,” Mathematical Programming Study, 9, 75-94, 1978.
- Lakshminarayan, S., R. Lakshmanan, R. Papineau, and R. Rochette, “Order Preserving Allocation of Jobs to Two Non-identical Parallel machines: A solvable Case of the Maximum Cut problems,” INFOR, 17, 230-241, 1979.
- Laughunn, D.J, “Quadratic Binary programming,” Operations Research,14, 454-461, 1970.
- Lodi, A., K. Allemand, and T. M. Liebling, “An Evolutionary Heuristic for Quadratic 0-1 Programming,” Technical Report OR-97-12, D.E.I.S., University of Bologna, 1997.
- McBride, R.D., and J.S. Yormack, “An Implicit Enumeration Algorithm for Quadratic Integer Programming,” Management Science, 26, 282-296, 1980.
- Merz, P. and B. Freisleben, “Genetic Algorithms for Binary Quadratic Programming,” Proceedings of the 1999 International Genetic and Evolutionary Computation Conferecne (GECCO’99), pp. 417-424, Morgan Kaufmann, 1999.
- Pardalos, P, and G.P. Rodgers, “A Branch and Bound Algorithm for Maximum Clique problem,” Computer & OR, 19, 363-375, 1992.
- Pardalos, P, and G.P. Rodgers, “Computational Aspects of a Branch and Bound Algorithm for Quadratic Zero-one Programming,” Computing, 45, 131-144, 1990.

- Pardalos, P, and J. Xue, “The Maximum Clique Problem,” The Journal of Global Optimization, 4, 301-328, 1994.
- Pardalos, P, and J. Xue, “The Maximum Clique Problem,” The Journal of Global Optimization, 4, 301-328, 1994.
- Pardalos, P, and S. Jha, “Complexity of Uniqueness and Local Search in Quadratic 0-1 Programming,” OR Letts, 11, 119-123, 1992.
- Phillips, A.T., and J.B. Rosen, “A Quadratic Assignment Formulation of the Molecular Conformation Problem,” Journal of Global Optimization, 4, 229-241, 1994.
- Plateau M.-C., Reformulations quadratiques convexes pour la programmation quadratique en variables 0-1, Ph.D. Thesis, Conservatoire National des Arts et Métiers / CEDRIC, November 2006. (in French, available online at http://www.cedric.cnam.fr/PUBLIS/RC1115.pdf).
- W.I. Zangwill, Media selection by decision programming, Journal of Advertising Research 5 (3) (1965) 30–36.
- Watters L.J., Reduction of integer polynomial programming problems to zero-one linear programming problems, Operations Research 15 (1967) 1171–1174.
- Willaims, A. C., “Quadratic 0-1 Programming Using the Roof Duality with Computational Results,” Rutcor Research Report 8-85, Rutgers University, New Brunswick, NJ., 1985.
- Witsgall, C., “Mathematical Methods of site Selection for Electronic System (EMS),” NBS Internal Report, 1975.