Available online at www.sciencedirect.com

ScienceDirect

Discrete Optimization 5 (2008) 501–529

DISCRETE OPTIMIZATION

www.elsevier.com/locate/disopt

A max-flow approach to improved lower bounds for quadratic unconstrained binary optimization (QUBO)

Endre Boros[a,*], Peter L. Hammer[a,✠], Richard Sun[b], Gabriel Tavares[a]

[a] RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854, USA
[b] Lucent Technologies, Holmdel, NJ 07733, USA

Received 27 March 2006; accepted 3 February 2007
Available online 29 October 2007

Abstract

The "roof dual" of a QUBO (Quadratic Unconstrained Binary Optimization) problem has been introduced in [P.L. Hammer, P. Hansen, B. Simeone, Roof duality, complementation and persistency in quadratic 0–1 optimization, Mathematical Programming 28 (1984) 121–155]; it provides a bound to the optimum value, along with a polynomial test of the sharpness of this bound, and (due to a "persistency" result) it also determines the values of some of the variables at the optimum. In this paper we provide a graph-theoretic approach to provide bounds, which includes as a special case the roof dual bound, and show that these bounds can be computed in $O(n^3)$ time by using network flow techniques. We also obtain a decomposition theorem for quadratic pseudo-Boolean functions, improving the persistency result of [P.L. Hammer, P. Hansen, B. Simeone, Roof duality, complementation and persistency in quadratic 0–1 optimization, Mathematical Programming 28 (1984) 121–155]. Finally, we show that the proposed bounds (including roof duality) can be applied in an iterated way to obtain significantly better bounds. Computational experiments on problems up to thousands of variables are presented.

Keywords: Quadratic unconstrained binary optimization; Roof duality; Persistency; Network flows

1. Introduction

It is well-known that numerous problems in operations research (e.g. portfolio problems), in computer science (e.g. maximum satisfiability problems), in discrete mathematics (e.g. weighted stability number of graphs) and in physics (e.g. the Ising problem) can be formulated as unconstrained minimization problems of quadratic real-valued polynomials in 0–1 variables. The papers [12,20,26] list these and many other applications.

A problem of this type, appearing in discrete mathematics, is the "graph balancing" problem (see [28]). In this problem a weighted signed graph is given, i.e. a graph with "positive" and "negative" edges, and with positive real weights on all edges. The question is to find a set of edges of minimum total weight, the removal of which makes the graph "balanced", i.e. having no circuits which involve an odd number of negative edges. In [23] it has been noticed

* Corresponding author.
E-mail address: boros@rutcor.rutgers.edu (E. Boros).
✠ Our colleague and friend Peter L. Hammer passed away in a tragic car accident while we were completing the revised version of this paper.

1572-5286/$ - see front matter
doi:10.1016/j.disopt.2007.02.001

that by associating a 0–1 variable to every vertex, this problem can be reduced to a QUBO problem in these variables. Noticing that every unconstrained quadratic minimization problem in n variables can be transformed to a balancing problem of a graph on $n + 1$ vertices, in this paper we reformulate the "roof duality" approach of [24] in terms of graph balancing.

In [24] the "roof dual" of a QUBO problem has been introduced in order to provide (i) a bound to the optimum value, (ii) a polynomial test for checking the sharpness of this bound, and (iii) a "persistency" result, which allows the fixation of some of the variables at their optimal values. More precisely, three different linear programming-based approaches were shown to yield the same bound (*roof dual*), and transformed the QUBO problem into an equivalent one, in which the linear part necessarily vanishes in all optimal binary points (*strong persistency*). These linear programs are equivalent to the so-called *standard linearization* of the QUBO problem, and the corresponding polyhedron was called later the *Boolean quadric polytope* (see e.g., [26,27,41]). Among the subsequent studies related to roof duality we mention [1,13–15,25,38,37,44].

Roof duality was later generalized to a complete hierarchy of increasingly sharper bounds C_k, $k = 2, 3, \dots, n$, where C_2 is the roof dual value, C_3 is called the *cubic dual*, and C_n is the optimum value of the QUBO problem [8]. Each bound C_k can be obtained by solving a Linear Program (LP) involving $O(n^k)$ variables. In particular, C_3 was shown to be equal to the LP optimum over the first Chvátal closure of the Boolean quadric polytope [9], and it is also known to be the optimum value of a special class of QUBO problems [3]. An analogous LP formulation, known as the elementary closure of lift-and-project cuts (see [18]) for the Boolean quadric polytope, provides almost exactly the same bound, involving also $O(n^3)$ variables [7]. However, the use of these LP-based bounds, as reported in [7], becomes computationally too expensive already for QUBO problems with 200 variables.

In this paper we introduce a new bound, the *iterated roof dual*, which is stronger than the C_2 bound, but somewhat weaker than the C_3 bound, and which can be computed using a network flow formulation much more efficiently than the LP-based calculation of C_3. By recalling a combination of Boolean, linear programming and graph-theoretic techniques from [10,43], we establish first the equivalence of roof duality to a particular packing problem in an associated graph. We derive in this way an efficient algorithm to compute the roof dual via the solution of a max-flow problem in a network of $2n + 2$ vertices. This approach improves also the persistency results, allowing the efficient detection of the optimal values of the largest possible subset of variables that could be fixed using the principles identified in [24]. Finally, we propose a method consisting in the iterative application of the network-based computation of the roof dual for finding the *iterated roof dual* bound. We demonstrate with the help of an extensive computational study that this method is very efficient, and highly competitive with other bounding techniques.

In Sections 2 and 3 we introduce some of the basic notations, definitions and problems studied in this paper. In Section 3 the equivalence of roof duality and a particular packing problem is proved. We also show that other variants of this packing problem may provide better bounds than roof duality. In Section 4 a network model for these packing problems is presented, while in Section 5 the structure and persistency results are established. In Section 6 we introduce the iterated roof dual bound, and in Section 7 we present an extensive computational study, involving numerous benchmark problem sets, as well as randomly generated problems with up to thousands of variables. In our study we compare the quality of bounds and computing times of our approach to several other well-established bounding techniques available in the literature.

2. Notations and definitions

A *pseudo-Boolean function* (PBF) is a real-valued function defined on $\{0, 1\}^n$. Every PBF has a unique multilinear polynomial expression. A PBF is called *quadratic* if its unique polynomial expression has degree at most 2.

The *quadratic 0–1 minimization problem* is to find the minimum in $\{0, 1\}^n$ of a quadratic PBF.

Let $V = \{x_1, \dots, x_n\}$ denote the set of 0–1 *variables*. The *complement* of a 0–1 variable x is defined by $\overline{x} = 1 - x$. The elements of the set $L = \{x_1, \overline{x}_1, \dots, x_n, \overline{x}_n\}$ are called *literals*.

A polynomial expression with variables in L and having only positive coefficients (with the possible exception of the free term) will be called a *posiform*. Every PBF has posiform representations (which however, are not unique, e.g. $f = 3 + x - 2xy$ can be represented by the posiforms $2 + \overline{x} + 2x\overline{y}$ or $1 + x + 2\overline{y} + 2\overline{x}\,y$).

Given a PBF f, a variable x_i will be called *essential* if there exists a point X^* such that $f(x_1^*, \dots, x_n^*) \neq f(x_1^*, \dots, x_{i-1}^*, \overline{x}_i^*, x_{i+1}^*, \dots, x_n^*)$, (e.g. $xy + \overline{x} + y + x\overline{y}$ depends essentially on y, but not on x).

Definition 2.1. If x and y are variables, then the expression $x\overline{y} + \overline{x}y$ is called a *positive bi-term*, while $xy + \overline{x}\,\overline{y}$ is called a *negative bi-term*.

Bi-terms naturally express the equality or non-equality of the variables involved:

$$x\overline{y} + \overline{x}y = 0 \Longleftrightarrow x = y,$$
$$xy + \overline{x}\,\overline{y} = 0 \Longleftrightarrow x \neq y.$$

Definition 2.2. If E is a collection of bi-terms, such that no pair of variables is involved in more than one element of E, and $\alpha_e > 0$ for all $e \in E$, then the quadratic pseudo-Boolean expression $\phi = \sum_{e\in E} \alpha_e e$ is called a *bi-form*.

Bi-forms offer a natural representation of quadratic PBF's:

Remark 2.3. For any quadratic PBF f in variables $x_1, \ldots, x_n$ there is a unique constant c and a unique bi-form ϕ in the variables $x_0, x_1, \ldots, x_n$ such that

$$f(x_1, \ldots, x_n) = c + \phi(1, x_1, \ldots, x_n). \tag{2.1}$$

In this case ϕ is called the *bi-form of* f. The variable x_0 will be called the *root* of ϕ; we shall also say that ϕ is rooted at x_0.

Proof. The representation can be obtained by successively applying to the given polynomial expression of f the transformations

$$\begin{aligned} x_i x_j &= \frac{1}{2}(x_i x_j + \overline{x}_i \overline{x}_j) + \frac{1}{2}(x_i + x_j) - \frac{1}{2} \\ -x_i x_j &= \frac{1}{2}(x_i \overline{x}_j + \overline{x}_i x_j) - \frac{1}{2}(x_i + x_j) \end{aligned} \tag{2.2}$$

of its positive and negative quadratic terms ($1 \le i < j \le n$), and then the transformations

$$\begin{aligned} x_i &= (x_i x_0 + \overline{x}_i \overline{x}_0) \\ -x_i &= (x_i \overline{x}_0 + \overline{x}_i x_0) - 1 \end{aligned} \tag{2.3}$$

of its positive and negative linear terms ($i = 1, \ldots, n$). □

Example 1. Consider the quadratic PBF given by

$$f = -3x_1 + 12x_2 - x_3 + 3x_4 + 14x_5 - 10x_1x_2 + 12x_1x_3 - 6x_1x_5 - 14x_2x_3 + 4x_3x_4 - 10x_4x_5.$$

The unique bi-form of f is then

$$\begin{aligned} \phi = {} & 5(x_0\overline{x}_1 + \overline{x}_0 x_1) + 6(x_0 x_5 + \overline{x}_0\overline{x}_5) + 5(x_1\overline{x}_2 + \overline{x}_1 x_2) \\ & + 6(x_1x_3 + \overline{x}_1\overline{x}_3) + 3(x_1\overline{x}_5 + \overline{x}_1 x_5) + 7(x_2\overline{x}_3 + \overline{x}_2 x_3) + 2(x_3x_4 + \overline{x}_3\overline{x}_4) + 5(x_4\overline{x}_5 + \overline{x}_4 x_5), \end{aligned}$$

satisfying the equation $f(x_1, \ldots, x_5) = \phi(1, x_1, \ldots, x_5) - 13$.

Remark 2.4. If ϕ is a bi-form, then $\phi(x_0, x_1, \ldots, x_n) = \phi(\overline{x}_0, \overline{x}_1, \ldots, \overline{x}_n)$ for every binary vector $(x_0, x_1, \ldots, x_n) \in \{0, 1\}^{n+1}$.

Proof. Follows directly from the definitions, since the value of ϕ depends only on equalities and non-equalities of the variables, that is on relations which do not change when simultaneously all the variables are complemented. □

Remark 2.5. If ϕ is the unique bi-form of the quadratic PBF f, then we have

$$\min_{(x_1,\ldots,x_n)\in\{0,1\}^n} f(x_1, \ldots, x_n) = c + \min_{(x_0,x_1,\ldots,x_n)\in\{0,1\}^{n+1}} \phi(x_0, x_1, \ldots, x_n).$$

Proof. Follows readily by Remark 2.4. □

The above remarks imply that instead of x_0, any of the $n+1$ variables of the bi-form ϕ could be fixed at 1, without changing the set of values ϕ assumes.

Corollary 2.6. *If ϕ is the unique bi-form of the quadratic PBF f, as in (2.1), and $g(x_0, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = c + \phi(x_0, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n)$, i.e., if we obtain g from ϕ by fixing $x_i = 1$, then both f and g are quadratic PBFs and have the same minimum value.* □

Example 2. Returning to the quadratic PBF f given in Example 1 and its unique bi-form ϕ, we can see that the function

$$\begin{aligned} g(x_0, x_2, x_3, x_4, x_5) &= -13 + \phi(x_0, 1, x_2, x_3, x_4, x_5) \\ &= 21 - 11x_0 + 2x_2 + 11x_3 + 3x_4 - 4x_5 + 12x_0x_5 - 14x_2x_3 + 4x_3x_4 - 10x_4x_5 \end{aligned}$$

has the same minimum value as f.

Remark 2.3 shows that the minimization of a PBF is equivalent to the minimization of the corresponding unique bi-form. Since the above transformation can be performed in $O(n^2)$ time, we shall assume from now on that quadratic PBFs are given as bi-forms in the variables $x_0, x_1, \ldots, x_n$. We shall use in the sequel both f and ϕ to denote bi-forms.

Definition 2.7. Given a bi-form, $f = \sum_{e \in E} \alpha_e e$, we associate to it a graph G_f, whose vertices correspond to the indices $\{0, 1, \ldots, n\}$ of the variables, and whose edges correspond to those pairs (i, j) for which there is a bi-term in f involving the variables x_i and x_j. The edge $e = (i, j)$ will sometimes refer in the sequel to the edge (i, j) of G_f, and some other times to the Boolean expression $e(X) = (x_i \overline{x}_j + \overline{x}_i x_j)$ or $= (x_i x_j + \overline{x}_i \overline{x}_j)$ associated to it in f. An edge will be called *positive* (*negative*) if the associated bi-term is positive (negative); the weight of an edge e is the corresponding positive coefficient α_e in f. In other words, G_f is a weighted signed graph, associated to the bi-form of f.

Definition 2.8. If X is a 0–1 vector of $n+1$ components, then an edge $e \in E$ is called *conflicting* with X if $e(X) \neq 0$, otherwise we say it *agrees* with X.

Remark 2.9. For any 0–1 vector $X \in \{0, 1\}^{n+1}$,

$$f(X) = \sum_{e \text{ is conflicting with } X} \alpha_e.$$

We consider paths in G_f with a possible repetition of edges (in the literature such paths are called sometimes *walks*). The number of times an edge e is used by a path P will be called the multiplicity of e with respect to P, and will be denoted by $m_P(e)$. We call a path *closed*, if its first and last vertices coincide.

Definition 2.10. A path is called *negative* if the sum of the multiplicities of the negative edges in it is odd. A closed negative path without repetition of edges is called a *negative cycle*, while a closed negative path, with possibly repeated edges, is called a *noose*. A noose is called *rooted* if it passes through the root of f. To a rooted noose N (which we will consider as a subset N of the edges together with a multiplicity function m_N) we shall also associate the PBF $N = \sum_{e \in N} m_N(e)e$.

The following easy remarks (see e.g. [10,15]) will be useful later in this paper.

Remark 2.11. The equation $f(X) = 0$ is consistent if and only if there is no negative cycle in G_f. Moreover, the equation $f(X) = 0$ has a unique solution (assuming $x_0 = 1$) if and only if G_f is connected and does not contain negative cycles.

Remark 2.12. If e and e' are bi-terms involving the pairs of variables x, y and y, z, respectively, then

$$e + e' = e'' + c$$

for some cubic posiform c and a bi-term e'' involving x and z. Moreover the sign of e'' is the product of the signs of e and e'.

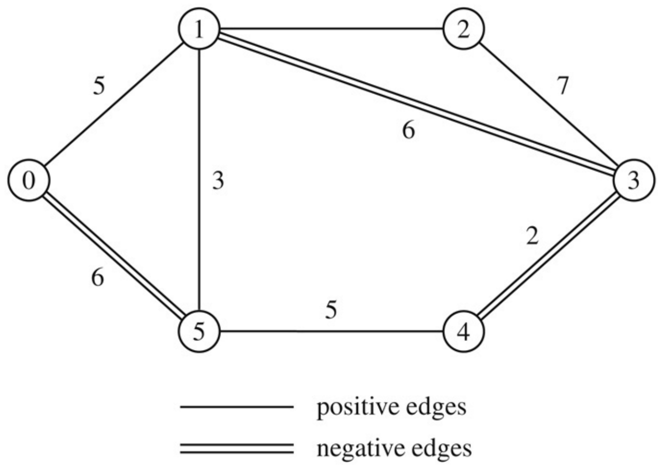

Fig. 1. The graph G_f of the bi-form given in Example 1.

Proof.

$$
\begin{aligned}
&(xy + \overline{x}\,\overline{y}) + (yz + \overline{y}\,\overline{z}) = (x\overline{z} + \overline{x}\,z) + 2(xyz + \overline{x}\,\overline{y}\,\overline{z}),\\
&(xy + \overline{x}\,\overline{y}) + (y\overline{z} + \overline{y}\,z) = (xz + \overline{x}\,\overline{z}) + 2(xy\overline{z} + \overline{x}\,\overline{y}\,z),\\
&(x\overline{y} + \overline{x}\,y) + (y\overline{z} + \overline{y}\,z) = (x\overline{z} + \overline{x}\,z) + 2(x\overline{y}\,z + \overline{x}\,y\overline{z}). \quad \square
\end{aligned}
\tag{2.4}
$$

Example 3. Consider the rooted noose $N = \{(0,1), (1,2), (2,3), (3,1), (1,0)\}$ in the graph of Fig. 1. Applying Remark 2.12, we have

$$
\begin{aligned}
N &= 2(x_0\overline{x}_1 + \overline{x}_0x_1) + (x_1\overline{x}_2 + \overline{x}_1x_2) + (x_2\overline{x}_3 + \overline{x}_2x_3) + (x_1x_3 + \overline{x}_1\overline{x}_3)\\
&= (x_0\overline{x}_1 + \overline{x}_0x_1) + (x_0\overline{x}_2 + \overline{x}_0x_2) + (x_2\overline{x}_3 + \overline{x}_2x_3) + (x_1x_3 + \overline{x}_1\overline{x}_3) + 2(x_0\overline{x}_1x_2 + \overline{x}_0x_1\overline{x}_2)\\
&= (x_0\overline{x}_1 + \overline{x}_0x_1) + (x_0\overline{x}_3 + \overline{x}_0x_3) + (x_1x_3 + \overline{x}_1\overline{x}_3) + 2(x_0\overline{x}_1x_2 + \overline{x}_0x_1\overline{x}_2) + 2(x_0\overline{x}_2x_3 + \overline{x}_0x_2\overline{x}_3)\\
&= (x_0\overline{x}_1 + \overline{x}_0x_1) + (x_0x_1 + \overline{x}_0\overline{x}_1) + 2(x_0\overline{x}_1x_2 + \overline{x}_0x_1\overline{x}_2)\\
&\quad + 2(x_0\overline{x}_2x_3 + \overline{x}_0x_2\overline{x}_3) + 2(x_0\overline{x}_1\overline{x}_3 + \overline{x}_0x_1x_3)\\
&= 1 + 2[(x_0\overline{x}_1x_2 + \overline{x}_0x_1\overline{x}_2) + (x_0\overline{x}_2x_3 + \overline{x}_0x_2\overline{x}_3) + (x_0\overline{x}_1\overline{x}_3 + \overline{x}_0x_1x_3)].
\end{aligned}
$$

Thus

$$N(1, x_1, x_2, x_3) = 1 + 2[\overline{x}_1x_2 + \overline{x}_2x_3 + \overline{x}_1\overline{x}_3].$$

More generally,

Remark 2.13. If N is a rooted noose in G_f, then

$$N(1, x_1, \ldots, x_n) = 1 + q(x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n),$$

where q is a quadratic posiform.

3. Roof duality and packing of nooses

Roof duality is a procedure proposed in [24] for associating to a given quadratic 0–1 maximization problem a relaxation of it, thus providing a bound on the optimum (called the *roof dual* of the function), along with a polynomial test to check the sharpness of this bound, and to determine the values of some of the variables in the optimum.

In this section we give a new interpretation of this method in terms of G_f and its rooted nooses. This interpretation will provide the basis for an efficient calculation of the roof dual, to be presented in Section 4.

Let f be a given bi-form, x_0 its root, and let $\mathcal{C}$, $\mathcal{N}$ and $\mathcal{N}_0$ denote the set of negative cycles, the set of nooses and the set of rooted nooses in G_f, respectively. If (P) is an optimization problem, let us denote its optimum value by $\omega(P)$.

With the notation of the previous section we shall associate to a bi-form f the following problems:

- a "cycle covering" problem

$$\begin{aligned} \text{minimize} \quad & v(Y) = \sum_{e \in E} \alpha_e y_e \\ \text{s.t.} \quad & \sum_{e \in C} y_e \geq 1 \quad \forall C \in \mathcal{C}, \\ & y_e \in \{0, 1\} \quad \forall e \in E, \end{aligned} \tag{\textbf{CC}}$$

- a "noose covering" problem

$$\begin{aligned} \text{minimize} \quad & v(Y) = \sum_{e \in E} \alpha_e y_e \\ \text{s.t.} \quad & \sum_{e \in N} m_N(e) y_e \geq 1 \quad \forall N \in \mathcal{N}, \\ & y_e \in \{0, 1\} \quad \forall e \in E, \end{aligned} \tag{\textbf{NC}}$$

- the continuous relaxation (**NC$^{\mathbf{c}}$**) of the noose covering problem, obtained from (**NC**) by replacing the conditions $y_e \in \{0, 1\}$ by $y_e \geq 0$ for all $e \in E$. (The conditions $y_e \leq 1$ are omitted since they hold automatically in the optimum.)

We shall also consider in the sequel the "noose packing" problem

$$\begin{aligned} \text{maximize} \quad & w(\xi) = \sum_{N \in \mathcal{N}} \xi_N \\ \text{s.t.} \quad & \sum_{N \ni e} m_N(e) \xi_N \leq \alpha_e \quad \forall e \in E, \\ & \xi_N \geq 0 \quad \forall N \in \mathcal{N} \end{aligned} \tag{\textbf{NP}}$$

and the "rooted noose packing" problem (**RNP**) which is obtained from (**NP**) by replacing $\mathcal{N}$ by $\mathcal{N}_0$.

As we shall see below, problems (**CC**) and (**NC**) are integer programming problems which are equivalent with the minimization of f, while (**NC$^{\mathbf{c}}$**), (**NP**) and (**RNP**) are weaker linear programming relaxations of the above integer programming problems, and the weakest one (**RNP**) turns out to be equivalent with roof duality.

Finally let us rephrase the "roof duality" approach of [24] for the minimization of bi-forms. For this, let us consider the bi-form of f written as

$$f = \sum_{(i,j) \in E^+} \alpha_{ij} (x_i \overline{x}_j + \overline{x}_i x_j) + \sum_{(i,j) \in E^-} \alpha_{ij} (x_i x_j + \overline{x}_i \overline{x}_j),$$

where E^+ and E^- denote the set of positive and negative edges in G_f, respectively. For each quadratic term of f, its L_1 optimal linear lower bounds are given by

$$\begin{aligned} & x_i x_j \geq \lambda_{ij} (x_i + x_j - 1), \quad \text{for any } 0 \leq \lambda_{ij} \leq 1, \\ & x_i \overline{x}_j \geq \lambda_{i\overline{j}} (x_i - x_j), \quad \text{for any } 0 \leq \lambda_{i\overline{j}} \leq 1, \\ & \overline{x}_i x_j \geq \lambda_{\overline{i}j} (x_j - x_i), \quad \text{for any } 0 \leq \lambda_{\overline{i}j} \leq 1, \\ & \overline{x}_i \overline{x}_j \geq \lambda_{\overline{i}\overline{j}} (1 - x_i - x_j), \quad \text{for any } 0 \leq \lambda_{\overline{i}\overline{j}} \leq 1, \end{aligned}$$

for $0 \leq i < j \leq n$. For a fixed parameter vector λ let

$$L_\lambda(X) = \sum_{(i,j) \in E^-} \alpha_{ij} (\lambda_{\overline{i}\overline{j}} - \lambda_{ij}) + \sum_{i=0}^{n} x_i \left[\sum_{(i,j) \in E^-} \alpha_{ij} (\lambda_{ij} - \lambda_{\overline{i}\overline{j}}) + \sum_{(i,j) \in E^+} \alpha_{ij} (\lambda_{i\overline{j}} - \lambda_{\overline{i}j}) \right].$$

It can be seen (as in [24]) that the roof dual $\rho(f)$ of f is given by

$$\rho(f) = \max_{\lambda} \min_{X} L_\lambda(X). \tag{3.1}$$

The main result of this section is

Theorem 3.1.

$$\begin{aligned}\min_{X\in\{0,1\}^{n+1}} f(X) &= \omega(\mathbf{CC})\\ &= \omega(\mathbf{NC})\\ &\geq \omega(\mathbf{NC^c})\\ &= \omega(\mathbf{NP})\\ &\geq \omega(\mathbf{RNP}) = \rho(f).\end{aligned}$$

The rest of this section will be devoted to the proof of this theorem; the main effort of the proof concerns the last equation.

Lemma 3.2. $\min_{X\in\{0,1\}^{n+1}} f(X) = \omega(\mathbf{CC})$.

Proof. For any binary vector X let Y^X be the vector whose components are defined by

$$y_e = \begin{cases}1 & \text{if } e \text{ is conflicting with } X,\\ 0 & \text{otherwise.}\end{cases}$$

Then, by Remark 2.9 we have $v(Y^X) = f(X)$, implying $\omega(\mathbf{CC}) \leq \min f(X)$.

Conversely, let Y be a feasible solution to (**CC**). Then the subgraph formed by the edges $\{e|y_e = 0\}$ does not contain a negative cycle. It follows from Remark 2.11 that there exists a vector X^Y such that $e(X^Y) = 0$ for edges e with $y_e = 0$. Thus, we have $v(Y) \geq f(X^Y)$, implying $\omega(\mathbf{CC}) \geq \min f(X)$. □

Lemma 3.3. $\omega(\mathbf{CC}) = \omega(\mathbf{NC})$.

Proof. For any $N \in \mathcal{N}$ there is a cycle $C \in \mathcal{C}$ with $C \subseteq N$, therefore if Y is a feasible solution to (**CC**), then

$$\sum_{e\in N} m_N(e) y_e \geq \sum_{e\in C} y_e \geq 1,$$

implying that the feasible solutions of (**CC**) are also feasible in (**NC**). The converse is immediately true, since $\mathcal{C} \subseteq \mathcal{N}$. □

Lemma 3.4. $\omega(\mathbf{NC}) \geq \omega(\mathbf{NC^c}) = \omega(\mathbf{NP}) \geq \omega(\mathbf{RNP})$.

Proof. Problem (**NC^c**) is the linear programming relaxation of the minimization problem (**NC**), and problem (**NP**) is the linear programming dual of problem (**NC^c**). Finally, problem (**RNP**) is the same maximization problem as (**NP**) with added restrictions requiring $\xi_N = 0$ for all $N \in \mathcal{N} \setminus \mathcal{N}_0$. □

In order to establish the remaining equality in Theorem 3.1 we shall first show

Lemma 3.5. $\omega(\mathbf{RNP}) \leq \rho(f)$.

Proof. It has been shown in [24] that $\rho(f)$ is the largest constant c satisfying the equation $f = c + g$ with some quadratic posiform g. For a feasible solution ξ of (**RNP**) let us define

$$\beta_e \stackrel{\text{def}}{=} \alpha_e - \sum_{N\ni e} m_N(e)\xi_N,$$

for $e \in E$. The feasibility of ξ implies that all $\beta_e \geq 0$, and therefore $f = \sum_{N\in\mathcal{N}_0} \xi_N N + \sum_{e\in E} \beta_e e$ is a decomposition of f into two bi-forms. It follows from Remark 2.13 that for each rooted noose $N \in \mathcal{N}_0$ there is a quadratic posiform q_N such that $N = 1 + q_N$. Hence we have

$$\begin{aligned}f &= \sum_{N\in\mathcal{N}_0} \xi_N(1 + q_N) + \sum_{e\in E} \beta_e e\\ &= w(\xi) + \sum_{N\in\mathcal{N}_0} \xi_N q_N + \sum_{e\in E} \beta_e e.\end{aligned}$$

Therefore f is the sum of $w(\xi)$ and of a posiform, implying $\omega(\mathbf{RNP}) \leq \rho(f)$. □

In order to obtain the converse inequality, we shall show that to any fixed parameter λ there corresponds a feasible solution $\xi = \xi(\lambda)$ of (**RNP**) satisfying the inequality $w(\xi) \geq \min_X L_\lambda(X)$.

Let $0 \leq \hat{\lambda} \leq 1$ be a fixed parameter vector, and let us introduce

$$\beta_{ij} = \begin{cases} \alpha_{ij}|\hat{\lambda}_{ij} - \hat{\lambda}_{\bar{i}\,\bar{j}}|, & \text{if } (i,j) \in E^-, \\ \alpha_{ij}|\hat{\lambda}_{i\bar{j}} - \hat{\lambda}_{\bar{i}\,j}|, & \text{if } (i,j) \in E^+; \end{cases} \tag{3.2}$$

furthermore, let us associate a linear function to each edge in G_f by putting

$$l_{ij} = \begin{cases} (x_i + x_j - 1), & \text{if } (i,j) \in E^- \text{ and } \hat{\lambda}_{ij} > \hat{\lambda}_{\bar{i}\,\bar{j}}, \\ (1 - x_i - x_j), & \text{if } (i,j) \in E^- \text{ and } \hat{\lambda}_{ij} < \hat{\lambda}_{\bar{i}\,\bar{j}}, \\ (x_i - x_j), & \text{if } (i,j) \in E^+ \text{ and } \hat{\lambda}_{i\bar{j}} > \hat{\lambda}_{\bar{i}\,j}, \\ (x_j - x_i), & \text{if } (i,j) \in E^+ \text{ and } \hat{\lambda}_{i\bar{j}} < \hat{\lambda}_{\bar{i}\,j}, \end{cases} \tag{3.3}$$

and let $\hat{E} = \{(i,j) | \beta_{ij} > 0\}$. Then,

Remark 3.6. $\hat{G}_f = (V, \hat{E})$ is a subgraph of G_f, $\beta_{ij} \leq \alpha_{ij}$ for any $(i,j) \in E(G_f)$, and

$$L_{\hat{\lambda}}(X) = \sum_{(i,j)\in\hat{E}} \beta_{ij} l_{ij}. \tag{3.4}$$

Definition 3.7. An ordered sequence $A = \{(i_1, j_1), (i_2, j_2), \ldots, (i_s, j_s)\}$ of edges of $\hat{G}_f$ is called an *alternating noose* if $j_k = i_{k+1(mod\, s)}$ and if x_{j_k} appears with different signs in the expressions $l_{i_k j_k}$ and $l_{i_{k+1} j_{k+1}}$, $k = 1, \ldots, s-1$. The vertex i_1 is called the *root* of A.

Lemma 3.8. *If A is an alternating noose, rooted at r, and if we denote*

$$L_A = \sum_{k=1}^{s} l_{i_k j_k} \tag{3.5}$$

then:

(i) $L_A = 0$, *if x_r appears with different signs in $l_{i_1 j_1}$ and $l_{i_s j_s}$;*
(ii) $L_A = 2x_r - 1$, *if x_r appears with positive signs in both $l_{i_1 j_1}$ and $l_{i_s j_s}$;*
(iii) $L_A = 1 - 2x_r$, *if x_r appears with negative signs in both $l_{i_1 j_1}$ and $l_{i_s j_s}$.*

Therefore, the value of L_A is 0 *or* $\pm$ 1 *in every binary point X.*

Remark 3.9. If A is an alternating noose in $\hat{G}_f$ and $L_A \neq 0$, then the edges of A with the corresponding multiplicities form a noose in G_f.

Proof. This follows by noticing that an expression l_{ij} given by (3.3) contains a ± 1 constant only if $(i,j) \in E^-$, and that the sum of these constants in (3.5) is also ± 1 by Lemma 3.8. Thus the ordered sequence of edges A in G_f passes through on an odd number of negative edges. □

Denoting by $\mathcal{A}$ the set of alternating nooses of $\hat{G}_f$ rooted at 0, and successively generating such nooses and "subtracting" them from $\hat{G}_f$ we shall eventually obtain a representation of $L_{\hat{\lambda}}$ of the form

$$L_{\hat{\lambda}} = \sum_{A\in\mathcal{A}} \eta_A L_A + \hat{L}, \tag{3.6}$$

where the η_A's are non-negative reals satisfying

$$\sum_{A\ni(i,j)} \eta_A \leq \beta_{ij}, \quad \forall (i,j), \tag{3.7}$$

and such that (3.6) is maximal, i.e. $\hat{L}$ does not contain an alternating noose $A \in \mathcal{A}$.

To the decomposition (3.6) of $L_{\hat{\lambda}}$ we shall associate a vector $\hat{\xi}$ by putting $\hat{\xi}_N = \eta_A$ if N is the rooted noose of G_f corresponding (by Remark 3.9) to A.

Lemma 3.10. *$\hat{\xi}$ is a feasible solution to* (**RNP**) *and*

$$w(\hat{\xi}) \geq \sum_{A \in \mathcal{A}} \eta_A L_A(X)$$

for any binary vector X.

Proof. The feasibility of $\hat{\xi}$ follows from (3.7) and Remark 3.6. Since $L_A(X) \in \{0, +1, -1\}$ by Lemma 3.8,

$$w(\hat{\xi}) = \sum \hat{\xi}_N = \sum \eta_A \geq \sum \eta_A L_A(X). \quad \square$$

Definition 3.11. Let us define for any pair of (not necessarily distinct) indices i, j a set of *elementary* functions by

$$\mathcal{E}_{ij} \stackrel{\text{def}}{=} \begin{cases} \{\pm(x_i - x_j), \pm(1 - x_i - x_j)\} & \text{if } i \neq j, \\ \{(1 - 2x_i),\ (2x_i - 1)\} & \text{if } i = j \end{cases}$$

for $0 \leq i, j \leq n$. Two elementary functions $l_{ij} \in \mathcal{E}_{ij}$ and $l_{jk} \in \mathcal{E}_{jk}$ will be called *conflicting* at j if the signs of x_j in the two functions are different.

Remark 3.12. If $l_{ij} \in \mathcal{E}_{ij}$ and $l_{jk} \in \mathcal{E}_{jk}$ are conflicting at j and $i \neq k$, then the function $l_{ik} \stackrel{\text{def}}{=} l_{ij} + l_{jk}$ belongs to $\mathcal{E}_{ik}$.

Definition 3.13. A positive combination of elementary functions will be called *conflict-free* if it contains no conflicting pairs of elementary functions.

Lemma 3.14. *Any non-negative combination $L = \sum_{l_{ij} \in \mathcal{E}_{ij}} \gamma_{ij} l_{ij}$ of elementary functions has a conflict-free representation.*

Proof. Starting with the given representation of L we check for the existence of a conflicting pair, say l_{ij} and l_{jk} with positive coefficients. If such a pair is found, and say $0 < \gamma_{ij} \leq \gamma_{jk}$, we modify the representation by introducing the elementary function $l_{ik} = l_{ij} + l_{jk}$ (see Remark 3.12), and applying the equation

$$\gamma_{ij} l_{ij} + \gamma_{jk} l_{jk} = \gamma_{ij} l_{ik} + (\gamma_{jk} - \gamma_{ij}) l_{jk}.$$

Repeating this argument for all the other pairs of elementary functions conflicting at j, we shall arrive in a finite number of steps to a representation of L without conflicts at j. Remarking that if there is no conflict at j then the above procedure does not introduce a new conflict at j, and repeating the previous steps for the other indices, we shall obtain in a finite number of steps a conflict-free representation of L. $\square$

Returning to the decomposition (3.6) we can state

Lemma 3.15.

$$\min_{x_i \in \{0,1\}, i=1,\ldots,n} \hat{L}(1, x_1, \ldots, x_n) \leq 0.$$

Proof. By the definition of $L_{\hat{\lambda}}$,

$$\hat{L} = \sum \hat{\gamma}_{ij} l_{ij}$$

is given as a positive combination of elementary functions not belonging to $\mathcal{E}_{00}$. Let us consider a conflict-free representation of $\hat{L}$,

$$\hat{L} = \sum \gamma_{ij} l_{ij}, \tag{3.8}$$

the existence of which is assured by Lemma 3.14. An element of $\mathcal{E}_{ii}$ with positive coefficient can be introduced by the above transformation only if there is an alternating noose $A = \{(i_1, j_1), \ldots, (i_t, j_t)\}$ rooted at i with all positive coefficients $\hat{\gamma}_{i_s,j_s} > 0$, $s = 1, \ldots, t$. Due to the "maximality" of (3.6) no element of $\mathcal{E}_{00}$ has positive coefficient in (3.8). We can also notice that fixing any variable x_i to the value 1 (0) if it appears with a negative (positive) sign in l_{ij}, we get $l_{ij} \leq 0$. For every $i = 1, \ldots, n$ let us put $x_i^* = 0$ if x_i appears only with positive coefficients, $x_i^* = 1$ if x_i appears only with negative coefficients in (3.8). We have now $l_{ij}(X^*) \leq 0$ for all the terms with $\gamma_{ij} > 0$, thus proving the lemma. □

Lemma 3.16.

$$w(\hat{\xi}) \geq \min_{X \in \{0.1\}^{n+1}} L_{\hat{\lambda}}(X).$$

Proof. The lemma follows from Lemmas 3.10 and 3.15. □

Finally, Lemmas 3.5 and 3.16 imply

Lemma 3.17. $\omega(\mathbf{RNP}) = \rho(f)$.

Theorem 3.1 follows now directly from Lemmas 3.2–3.4 and 3.17.

Let us remark that by the symmetries observed in Corollary 2.6, we could use any of the variables as roots, and consider analogous rooted noose packing problems: Denoting by $\mathcal{N}_i$ the set of nooses rooted at vertex x_i, for $i = 0, 1, \ldots, n$, we can consider the problems

$$\begin{aligned} &\text{maximize} \quad w(\xi) = \sum_{N \in \mathcal{N}} \xi_N \\ &\qquad\text{s.t.} \quad \sum_{N \ni e} m_N(e)\xi_N \leq \alpha_e \quad \forall e \in E, \\ &\qquad\qquad\; \xi_N \geq 0 \quad \forall N \in \mathcal{N}_i. \end{aligned} \tag{RNP(i)}$$

Clearly, problem (**RNP(0)**) is the same as (**RNP**). Furthermore, the optimum value of each of the problems (**RNP(i)**), $i = 0, 1, \ldots, n$, is a lower bound on the minimum of f. Thus, Lemma 3.17 implies readily the following corollary.

Corollary 3.18.

$$\min_{X \in \{0,1\}^{n+1}} f(X) \geq \max_{i=0,\ldots,n} \omega(\mathbf{RNP(i)}) \geq \omega(\mathbf{RNP}) = \rho(f). \quad \square$$

Example 4. For the bi-form ϕ in Example 1 the following is an optimal rooted noose packing for the root x_0:

$$\xi_N = \begin{cases} 3, & \text{if } N = [(0,1), (1,5), (5,0)], \\ 2, & \text{if } N = [(0,1), (1,3), (3,4), (4,5), (5,0)], \\ 0, & \text{otherwise}, \end{cases}$$

implying by Lemma 3.17 that $\rho(\phi) = 5$. However, switching the root to vertex x_1, we can obtain the following noose packing (rooted at x_1):

$$\xi_N = \begin{cases} 3, & \text{if } N = [(1,0), (0,5), (5,1)], \\ 5, & \text{if } N = [(1,2), (2,3), (3,1)], \\ 1, & \text{if } N = [(1,3), (3,4), (4,5), (5,0), (0,1)], \\ 0, & \text{otherwise}, \end{cases}$$

providing by Corollary 3.18 the value 9 as a lower bound for ϕ.

In concluding this section, let us make a few side remarks, leaving the easy proofs for the reader.

Let us note first that in the noose packing problem (**NP**) we could replace $\mathcal{N}$ by $\mathcal{C}$ without changing the optimum value. Furthermore, the resulting negative cycle packing problem can easily be shown to be equivalent with negative triangle packing, which we state here for completeness. For this, let us introduce a positive edge e for every pair of variables which are not connected by an edge in G_f, assume that $\alpha_e = 0$ for these newly introduced edges, and denote by $\tilde{E}$ this extended set of edges. Let us also denote by $\overline{e}$ the sign complement of edge e (i.e., if e is a negative edge between variables x_i and x_j, then $\overline{e}$ denotes a positive edge between x_i and x_j, etc.), and note that if $e \in \tilde{E}$, then we have $\overline{e} \notin \tilde{E}$. Let us denote finally by $\mathcal{T}$ the collection of all negative triangles (i.e., negative cycles consisting of three edges), and consider the problem

$$\begin{aligned}\text{maximize}\quad & w(\xi) = \sum_{T\in\mathcal{T}} \xi_T - \sum_{e\in\tilde{E}} \left(\sum_{T\ni\overline{e}} \xi_T \right) \\ \text{s.t.}\quad & 0 \le \sum_{T\ni e} \xi_T - \sum_{T\ni\overline{e}} \xi_T \le \alpha_e \quad \forall e \in \tilde{E}, \\ & \xi_T \ge 0 \quad \forall T \in \mathcal{T}.\end{aligned} \tag{TP}$$

We can thus conclude that ω(**NP**) $=$ ω(**TP**). Furthermore, comparing problem (**TP**) with the formulation of the so-called *cubic dual* bound C_3 introduced in [8], and in particular with the triangle inequalities-based formulations of it (see [3,9]), we can easily show the following claim.

Remark 3.19. ω(**NP**) $=$ ω(**TP**) $= C_3$. □

Though this bound is generally better than the roof dual bound, and can also be computed by linear programming in polynomial time, computationally it is still much more involved than the network flow-based computation of an optimal rooted noose packing, which we present in the next section. Due to this computational advantage, in this paper we focus on rooted noose packings, and their iterated versions.

4. Rooted noose packing via maximum flows

In this section we show that the optimum value and an optimal solution of a rooted noose packing problem can be computed by solving a maximum-flow problem in a network on $2n + 2$ vertices. Together with the result of the previous section this implies that the roof dual $\rho(f)$ of a quadratic pseudo-Boolean function f in n variables, as well as any of the possibly improved lower bounds ω(**RNP(i)**) can be computed in $O(n^3)$ time. We present our approach for the case of x_0 as root, though it can be applied directly for any other choice of a root.

Definition 4.1. If f is a bi-form rooted at x_0, then let $N_f = (W, A)$ be the network, whose $2n + 2$ *nodes* correspond to the literals of the set $W = \{x_0, \overline{x}_0, \dots, x_n, \overline{x}_n\}$, and whose *edges* are associated to the edges of G_f in the following way. If $e \in E$ is a positive edge between i and j, i.e. $e = x_i\overline{x}_j + \overline{x}_i x_j$, then there are two corresponding edges in A: an edge e' between x_i and x_j and another edge e'' between $\overline{x}_i$ and $\overline{x}_j$. If $e \in E$ is a negative edge between i and j, i.e. $e = x_i x_j + \overline{x}_i\overline{x}_j$, then there are two corresponding edges in A: an edge e' between x_i and $\overline{x}_j$ and another edge e'' between $\overline{x}_i$ and x_j. Let in both cases $c(e') = c(e'') = \frac{1}{2}\alpha_e$ be the *capacities* of these edges in N. As an example, the network N_f corresponding to the bi-form f of Example 1 is given in Fig. 2.

If P is a path from x_0 to $\overline{x}_0$ in N_f, going through the vertices $\{u_1, \dots, u_p\}$ (i.e. $u_1 = x_0$, $u_p = \overline{x}_0$), then the sequence $\{\overline{u}_p, \dots, \overline{u}_1\}$ describes another path $\overline{P}$ between x_0 and $\overline{x}_0$. The pair $P, \overline{P}$ will be called a *bi-path*.

The following lemmas can be seen easily.

Lemma 4.2. *There is a one-to-one correspondence between the rooted nooses in G_f and the bi-paths in N_f.*

Proof. A rooted noose provides a closed walk from x_0 to x_0 in G_f, in which we pass an odd number of times negative edges (some of them possibly twice). Thus, by the above definitions, the corresponding edges in N_f form a path P from x_0 to $\overline{x}_0$ and its twin $\overline{P}$, i.e., a bi-path. Conversely, a bi-path $P, \overline{P}$ in N_f corresponds to a closed walk W from x_0 to x_0 in G_f. Since along the path P (and $\overline{P}$) we must move an odd number of times from an un-complemented variable to a complemented one, in W we must pass through an odd number of negative edges, i.e., W is a rooted noose in G_f. □

It is well-known in the theory of network flows that a flow F from x_0 to $\overline{x}_0$ (in N_f) can always be decomposed into the sum of a finite number of elementary flows $F_1, \dots, F_t$, going through the paths $P_1, \dots, P_t$ from x_0 to $\overline{x}_0$. Thus, due to the symmetric nature of N_f, the following claim follows readily from the definitions.

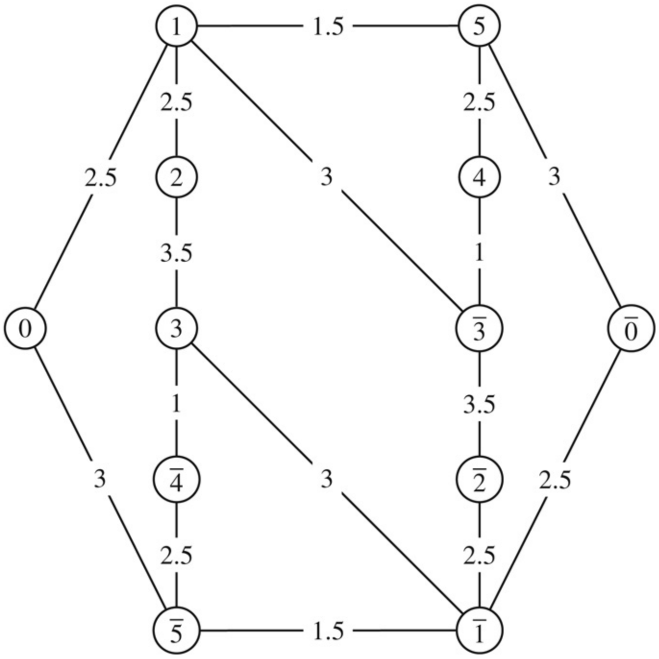

Fig. 2. The network N_f corresponding to the bi-form f given in Example 1.

Lemma 4.3. *Let F_i, $i = 1, \ldots, t$ be elementary flows from x_0 to $\overline{x}_0$ through the paths P_i, and having values f_i, respectively. Further, let $\overline{F}_i$ be the elementary flow through the path $\overline{P}_i$ having the value f_i for $i = 1, \ldots, t$. If $F = \sum F_i$ is a feasible flow in N_f, then $\overline{F} = \sum \overline{F}_i$ is also a feasible flow in N_f (having the same value as F).* $\square$

A flow F from x_0 to $\overline{x}_0$ in N_f with the property $F = \overline{F}$ is called a *bi-flow*.

Lemma 4.4. *To every feasible rooted noose packing $\xi = \sum_{N \in \mathcal{N}_0} \xi_n N$ there is a corresponding bi-flow of N_f with $\sum_{N \in \mathcal{N}_0} \xi_N$ as its flow value. Conversely, every feasible bi-flow in N_f corresponds in this way to a feasible solution of* (**RNP**) *(however, this correspondence may not be one-to-one, in general).*

Proof. Since a convex combination of feasible flows is again a feasible flow, Lemma 4.3 implies that from any feasible flow F of N_f we can obtain a feasible bi-flow with the same flow value, by considering simply $\frac{1}{2}F + \frac{1}{2}\overline{F}$. Therefore, Lemmas 4.2 and 4.3 imply readily the claim.

To illustrate that rooted noose packings of G_f and bi-flows of N_f are not necessarily in a one-to-one correspondence, let us consider the bi-form f defined by

$$\begin{aligned} f(x_0, x_1, x_2, x_3, x_4) = {} & 2(x_0x_1 + \overline{x}_0\overline{x}_1) + 2(x_0x_2 + \overline{x}_0\overline{x}_2) + 2(x_1x_2 + \overline{x}_1\overline{x}_2) \\ & + 2(x_1x_3 + \overline{x}_1\overline{x}_3) + 2(x_3x_4 + \overline{x}_3\overline{x}_4) + 2(x_4x_1 + \overline{x}_4\overline{x}_1) \end{aligned}$$

and its graph G_f. The nooses

$$\begin{aligned} N_1 &= 2(x_0x_1 + \overline{x}_0\overline{x}_1) + (x_1x_3 + \overline{x}_1\overline{x}_3) + (x_3x_4 + \overline{x}_3\overline{x}_4) + (x_4x_1 + \overline{x}_4\overline{x}_1) \\ N_2 &= 2(x_0x_2 + \overline{x}_0\overline{x}_2) + 2(x_1x_2 + \overline{x}_1\overline{x}_2) + (x_1x_3 + \overline{x}_1\overline{x}_3) \\ &\quad + (x_3x_4 + \overline{x}_3\overline{x}_4) + (x_4x_1 + \overline{x}_4\overline{x}_1) \end{aligned}$$

with weights $\xi_{N_1} = \xi_{N_2} = 1$ form a feasible rooted noose packing in G_f. In the corresponding bi-flow of N_f however the flows cancel out on some of the arcs (corresponding to a circulation), and the non-zero edges of the resulting bi-flow correspond to the rooted noose packing consisting of a single noose N_3 with weight $\xi_{N_3} = 2$, where

$$N_3 = (x_0x_1 + \overline{x}_0\overline{x}_1) + (x_0x_2 + \overline{x}_0\overline{x}_2) + (x_1x_2 + \overline{x}_1\overline{x}_2). \quad \square$$

Finally, Lemma 4.4 implies immediately the main statement of this section:

Theorem 4.5. *ω(**RNP**) is equal to the value of the maximum flow from x_0 to $\overline{x}_0$ in N_f.* $\square$

Corollary 4.6. *Problem (**RNP**) can be solved in $O(n^3)$ time.* $\square$

Let us add that whenever the given bi-form f has integral coefficients, the corresponding network N_f has a half-integral maximum bi-flow, as the simple argument in the proof of Lemma 4.4 shows. Consequently, for integral bi-forms we have half-integral optimal noose packings.

Let us also note that the undirected network N_f could be viewed for algorithmic purpose as the directed network obtained from N_f by replacing every undirected edge $e = (u, v)$ by two directed arcs $e' = (u, v)$ and $e'' = (v, u)$ between the same pair of literals, and assigning to both of them capacity $c(e') = c(e'') = c(e)$. Recall further from the theory of network flows that the set of nodes reachable from the source in the residual network corresponding to an arbitrary maximum flow is always the same set, not depending on the particular maximum flow. This justifies the following definition:

Definition 4.7. Let us define the *source side of f* as the unique set of literals reachable from x_0 in the residual network corresponding to a maximum bi-flow, and denote it by Q_f.

Note that by the symmetry of N_f, the set $\overline{Q}_f$ is exactly the set of nodes from which $\overline{x}_0$ can be reached in a residual network corresponding to a maximum bi-flow.

Let us further remark that the directed version of N_f was generalized in [43] (see also [12] for a detailed description), associating a directed network N_f to an arbitrary quadratic posiform of a quadratic PBF f, such that the maximum-flow value in N_f is the same as the roof dual value of f. In our current presentation we prefer the symmetric bi-form-based model, because that leads us to the cycle and noose packing problems, and allows for an iterated version of rooted noose packing, providing substantial improvements over roof duality (see Sections 6 and 7).

Let us finally add that computing the roof dual value by the above undirected network model provides sometimes even algorithmic advantages. For instance, [33] shows that in an undirected network of n nodes and m edges a maximum flow of value v can be computed in $O(nm^{2/3}v^{1/6})$ time. Thus, for bi-forms with ”small” integer coefficients the roof dual value could be obtained more efficiently by using the algorithm of [33] in the above undirected network model than by standard network flow algorithms in the directed network model of [12,43].

5. Structure and persistency

The following “persistency” property of roof duality has been proved in [24]: if $\rho_0 + \sum_{j=1}^{n} \rho_j x_j$ is an optimal roof ($L_\lambda(X)$ in our notations) of a quadratic PBF f, then for every index j for which $\rho_j > 0$ ($\rho_j < 0$) we must have $x_j^* = 0$ (respectively, $x_j^* = 1$) in every minimum point X^* of f. Of course, the set of variables, which can be fixed at their optimal value in this way depends on the chosen optimal roof, and it can indeed be different for different optimal roofs. It was remarked in [24] that there exists a unique maximal subset of such variables, obtainable from certain optimal roofs, called the *master* roofs.

In this section we derive a decomposition of a bi-form from an optimal solution of a corresponding rooted noose packing problem (for an arbitrary root), and show that the set Q_f of literals, when x_0 is the chosen root, defines the unique maximal set of variables corresponding to a master roof. In this way the network flow-based computation of an optimal rooted noose packing provides us automatically also with a master roof (see also [10,6,12,43]).

Similarly to the terminology of [24] we call a bi-form f *gap-free* if $\min f = \rho(f)$.

Definition 5.1. Let $f = \sum_{e \in E} \alpha_e e$ and $g = \sum_{e \in E'} \beta_e e$ be two bi-forms defined on the same set of variables. We shall say that $g \preceq f$ if $E' \subseteq E$ and $\beta_e \le \alpha_e$ for all $e \in E'$; we write $g \prec f$ if $g \preceq f$ and the strict inequality $\beta_e < \alpha_e$ holds for at least one edge $e \in E'$, or $\alpha_e > 0$ for at least one edge $e \in E \setminus E'$.

Using these notations, we formulate now the following “structure” theorem:

Structure Theorem. *Any bi-form f can be decomposed into three parts*

$$f = h_f + g_f + f'_f$$

satisfying the following properties:

(i) h_f *is gap-free,* $\rho(h_f) = \rho(f)$ *and* h_f *is '$\preceq$'-minimal with respect to these properties, i.e. if* h' *is gap-free,* $\rho(h') = \rho(f)$ *and* $h' \preceq h$*, then* $h' = h$.

(ii) g_f *and* f'_f *have no common variables,* $\min g_f = 0$*, and* $h_f + g_f$ *is also gap-free, with* $\rho(h_f + g_f) = \rho(f)$.

(iii) *If* $x_0 = 1$ *the equation* $g_f(X) = 0$ *has a unique solution (in its essential variables), and every minimum point of* f *satisfies this equation.*

(iv) *The bi-form* f'_f *is "arbitrary", i.e. if* F' *is an arbitrary bi-form in the same variables and in the same bi-terms as* f'_f*, and if by definition* $F = h_f + g_f + F'$*, then* F *has a decomposition* $h_F = h_f$*,* $g_F = g_f$ *and* $f'_F = F'$*, satisfying the above properties.*

If f *has* n *variables, then such a decomposition can be found in* $O(n^3)$ *time.*

First we shall associate a decomposition of f to every optimal solution of (**RNP**); we shall show further in the section that for certain optimal solutions this decomposition satisfies the Structure Theorem.

Definition 5.2. Given the bi-form f, and a feasible solution $\xi = (\xi_N | N \in \mathcal{N}_0)$ of (**RNP**) we shall define the ξ-*decomposition* of f

$$f = h_\xi + g_\xi + f'_\xi \tag{5.1}$$

as follows. First, let

$$h_\xi = \sum_{N \in \mathcal{N}_0} \xi_N N = \sum_{e \in E} \beta_e e, \tag{5.2}$$

where $\beta_e = \sum_{N \ni e} m_N(e)\xi_N$ for $e \in E$. The sum of the remaining terms of $f - h_\xi$ (which is a bi-form, since $0 \le \beta_e \le \alpha_e$ for $e \in E$ due to the feasibility of ξ) will be regrouped into two parts, g_ξ and f'_ξ, where g_ξ is the bi-form whose graph is the connected component of G_{f-h_ξ} containing 0, and f'_ξ is the sum of the remaining terms. Further, we shall denote by P_ξ the set of vertices of G_{g_ξ}.

Persistency Theorem. *Let* ξ *be an arbitrary optimal solution of* (**RNP**). *Then at any minimum point* X^* *of* f *we have* $g_\xi(X^*) = 0$.

Example 5. As we have noted in Example 4, for the bi-form f of Example 1 the following is an optimal solution

$$\xi_N = \begin{cases} 3, & \text{if } N = [(0,1),(1,5),(5,0)], \\ 2, & \text{if } N = [(0,1),(1,3),(3,4),(4,5),(5,0)], \\ 0, & \text{otherwise,} \end{cases}$$

showing that $\rho(f) = \rho(\phi) - 13 = 5 - 13 = -8$. Furthermore, we have

$$\begin{aligned} h_\xi &= 3[(x_0\overline{x}_1 + \overline{x}_0 x_1) + (x_1\overline{x}_5 + \overline{x}_1 x_5) + (x_0 x_5 + \overline{x}_0\overline{x}_5)] \\ &\quad + 2[(x_0\overline{x}_1 + \overline{x}_0 x_1) + (x_1 x_3 + \overline{x}_1\overline{x}_3) + (x_3 x_4 + \overline{x}_3\overline{x}_4) + (x_4\overline{x}_5 + \overline{x}_4 x_5) + (x_0 x_5 + \overline{x}_0\overline{x}_5)] \\ &= 5(x_0\overline{x}_1 + \overline{x}_0 x_1) + 5(x_0 x_5 + \overline{x}_0\overline{x}_5) + 2(x_1 x_3 + \overline{x}_1\overline{x}_3) \\ &\quad + 3(x_1\overline{x}_5 + \overline{x}_1 x_5) + 2(x_3 x_4 + \overline{x}_3\overline{x}_4) + 2(x_4\overline{x}_5 + \overline{x}_4 x_5). \end{aligned}$$

The graph of G_{f-h_ξ} is shown in Fig. 3. The connected component of this graph containing 0 is induced by $P_\xi = \{0, 4, 5\}$, and therefore,

$$\begin{aligned} g_\xi &= (x_0 x_5 + \overline{x}_0\overline{x}_5) + 3(x_4\overline{x}_5 + \overline{x}_4 x_5), \\ f'_\xi &= 5(x_1\overline{x}_2 + \overline{x}_1 x_2) + 4(x_1 x_3 + \overline{x}_1\overline{x}_3) + 7(x_2\overline{x}_3 + \overline{x}_2 x_3). \end{aligned}$$

The Persistency Theorem shows that at any minimum point of f we must have $g_\xi(X) = 0$, i.e. from $x_0 = 1$ it follows that $x_4 = x_5 = 0$.

Definition 5.3. We say that a bi-form $h = \sum_{e \in E} \beta_e e$ is *critical* if there is no other bi-form $h' \preceq h$ such that $\rho(h) = \rho(h')$.

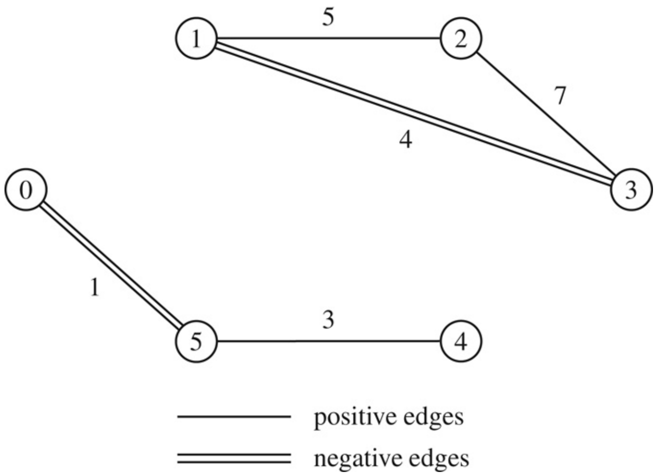

Fig. 3. The graph of G_{f-h} ($= G_g \cup G_{f'}$) given in Example 5.

Lemma 5.4. *If h is a critical bi-form, then*

(i) $h = \sum_{N \in \mathcal{N}_0} \xi_N N$, $\xi_N \geq 0$ *for the rooted nooses of G_h.*
(ii) *h is gap-free, i.e. there is a binary vector X' such that $h(X') = \rho(h)$.*
(iii) *For any binary vector X' with $h(X') = \rho(h)$ and for any rooted noose N with $\xi_N > 0$ there is exactly one edge of N conflicting with X'.*

Proof. (i) Let ξ be an optimal solution of the (**RNP**) problem corresponding to h. Since h is critical, Lemma 3.17 implies that $h = \sum \xi_N N$.

(ii) Let $\overline{h}$ be the "quadratic complement" of the PBF h (see [24]), i.e. the quadratic posiform for which $h = \rho(h) + \overline{h}$. We shall show that $\overline{h} = 0$ is consistent; obviously any solution X' of it satisfies $h(X') = \rho(h)$. The characterization of Boolean equations in [2] implies that if $\overline{h} = 0$ is inconsistent then we have a variable, say x, and some literals, say $u_1, \ldots, u_s$ and $v_1, \ldots, v_p$, such that the terms

$$xu_1, \overline{u}_1 u_2, \ldots, \overline{u}_s x$$

and

$$\overline{x} v_1, \overline{v}_1 v_2, \ldots, \overline{v}_p \overline{x}$$

are all present in $\overline{h}$. (These are the terms corresponding to a contradictory cycle in the implication graph; see [2].) Applying the transformations (2.2) we have

$$xu_1 + \sum_{i=1}^{s-1} \overline{u}_i u_{i+1} + \overline{u}_s x = G + (2x - 1)$$

$$\overline{x} v_1 + \sum_{j=1}^{p-1} \overline{v}_j v_{j+1} + \overline{v}_p \overline{x} = G' + (2\overline{x} - 1),$$

where G and G' are bi-forms. It can be seen from the identities (2.4) that the bi-form $G + G'$ contains bi-terms which are also present in h. Therefore, for some positive ϵ we have

$$h - \epsilon(G + G') = \rho(h) + [\overline{h} - \epsilon(xu_1 + \cdots + \overline{u}_s x + \overline{x} v_1 + \cdots + \overline{v}_p \overline{x})].$$

This equality is in contradiction with the fact that h is critical.

(iii) Applying Lemma 3.17 we have $\rho(h) = \sum \xi_N$. Since a rooted noose is a bi-form with integral coefficients, for any binary vector X and for any rooted noose N it follows that $N(X) \geq 1$. From this fact and from (i) above it follows that if we have $h(X') = \rho(h)$ then we also have

$$\rho(h) = h(X') = \sum \xi_N N(X') \geq \sum \xi_N = \rho(h),$$

implying that $N(X') = 1$ for every rooted noose N with $\xi_N > 0$; it follows that exactly one edge in N is in conflict with X'. $\square$

Let us observe next that the following claim follows readily by the definitions.

Remark 5.5. If ξ_1 and ξ_2 are optimal solutions of (**RNP**), such that $\xi_1 \preceq \xi_2$ holds, then we have $P_{\xi_1} \supseteq P_{\xi_2}$. $\square$

Definition 5.6. Let ξ be an optimal solution of (**RNP**). If the bi-form h_ξ is critical, then it will be called a *roof minor* of f. (Notice that a roof minor can be easily obtained, e.g. by solving the (**RNP**) problem with a max-flow algorithm based on shortest augmenting paths.)

Remark 5.7. If ξ_1 and ξ_2 are both optimal solutions of (**RNP**), then $\xi_3 = \frac{1}{2}(\xi_1 + \xi_2)$ is also an optimal solution of it, $h_{\xi_3} = \frac{1}{2}(h_{\xi_1} + h_{\xi_2})$, $g_{\xi_3} \geq \frac{1}{2}(g_{\xi_1} + g_{\xi_2})$, and $P_{\xi_3} \supseteq P_{\xi_1} \cup P_{\xi_2}$.

Proof. Immediate by the definitions. $\square$

The above remarks imply that there exists a *master* roof minor h_ξ, i.e. one defining a unique maximal set $P = P_\xi$ of persistent variables. We claim next that in fact we have this equality for any roof minor.

Remark 5.8. Let ξ be an optimal solution of (**RNP**) for which the bi-form h_ξ is critical. Then we have $P_\xi = P$.

Proof. Let F be a maximum flow in N_f to which ξ corresponds as in Lemma 4.4. Let us note that to any arc with a positive residual capacity corresponding to the maximum flow F, there is a corresponding term of g_ξ (with the possible exceptions of arcs entering x_0), and conversely, any non-zero term of g_ξ corresponds to a non-zero residual capacity in N_f. Thus, the variables involved in the set Q_f are exactly the essential variables of g_ξ, that is the set P_ξ. Since Q_f is unique, it does not depend on the chosen maximum flow, and hence we must have $P = P_\xi$. $\square$

Let f be a given bi-form, and let us fix an optimal solution ξ of the corresponding problem (**RNP**), such that h_ξ is a roof minor of f (i.e., for which $P = P_\xi$). We claim next that $h_\xi + g_\xi$ is a gap-free function.

Lemma 5.9. *If h_ξ is a roof minor of f, then there exists a binary vector X such that $h_\xi(X) + g_\xi(X) = \rho(h)$ (= $\rho(h + g) = \rho(f)$).*

Proof. By using a similar argument to that in (ii) of Lemma 5.4 we shall show that $\overline{h} + g_\xi = 0$ is consistent, where $\overline{h}$ denotes again the quadratic complement of h_ξ.

If $\overline{h} + g_\xi$ is always positive, then – similarly to the proof of (ii) of Lemma 5.4 – we can find a subset of the terms of $\overline{h} + g_\xi$, the sum of which is a bi-form, denoted by $G + G'$, such that $G + G' = 0$ is also inconsistent. If $G + G'$ has no common variables with g_ξ, then we get the same contradiction as in (ii) of Lemma 5.4. Otherwise the graph $G_{g_\xi} \cup G_{G+G'}$ contains a rooted noose $N \in \mathcal{N}_0$, in contradiction with the optimality of ξ. $\square$

From the previous results it is obvious that we can get a binary vector $\hat{X}$, satisfying $\rho(f) = \rho(h_\xi) = h_\xi(\hat{X})$ and $g_\xi(\hat{X}) = 0$ in $O(n^3)$ time. We are going to show below that at any minimum point X^* of f, the vectors X^* and $\hat{X}$ coincide in the components x_i for $i \in P$.

Definition 5.10. For any subset $S \subset V$, $0 \notin S$, we shall associate to $\hat{X}$ the binary vector X^S by putting

$$x_i^S = \begin{cases} \hat{x}_i & \text{if } i \notin S, \\ 1 - \hat{x}_i & \text{if } i \in S. \end{cases}$$

Lemma 5.11. *If $S \subset V$, $0 \notin S$ and $S \cap P \neq \emptyset$, then $f'_\xi(X^S) = f'_\xi(X^{S \setminus P})$, $g_\xi(X^S) > 0$, and $g_\xi(X^{S \setminus P}) = 0$.*

Proof. The first equation follows immediately from the fact that f'_ξ does not depend on variables x_i for $i \in P$. The second inequality follows from the facts that $g_\xi(\hat{X}) = 0$, $S \cap P \neq \emptyset$, and that for any vertex $i \in P$ there is an incident edge of G_{g_ξ} with positive weight. The third equality follows from the fact that g_ξ does not depend on any variables x_i for $i \notin P$. $\square$

The maximality of $P = P_\xi$ implies that there is a special relation between P and the rooted nooses N with $\xi_N > 0$.

Lemma 5.12. *Let $I = \{0, i_1, \ldots, i_k, i_{k+1}, \ldots, i_{t+1}, 0\}$ be the ordered sequence of vertices of a rooted noose N with $\xi_N > 0$. Let $e_k = (i_k, i_{k+1})$ be the edge (see (iii) of Lemma 5.4) of N conflicting with $\hat{X}$. If $I \not\subset P$ then let m denote the maximal index for which $i_m \in P$ but $i_{m+1} \notin P$. Then $i_j \in P$ for all $j \leq m$ and $m \leq k$.*

Proof. If $m > k$, then $G_{g_\xi} \cup \{e_k\}$ contains a rooted noose, say N'. The solution

$$\xi'_M = \begin{cases} \xi_M + \epsilon & \text{if } M = N', \\ \xi_M - \epsilon & \text{if } M = N, \\ \xi_M & \text{otherwise,} \end{cases}$$

is also optimal, and has strictly smaller weights in $h_{\xi'}$ on the edges of N connecting P and $V \setminus P$. Therefore $P_{\xi'} \supset P$, in contradiction with the maximality of P.

Similarly, if $m \leq k$ and there is a $j < m$ with $i_j \notin P$, then let N' be the rooted noose in the graph $G_{g_\xi} \cup \{(i_m, i_{m+1}), \ldots, (i_t, i_{t+1}), (i_{t+1}, 0)\}$. A similar modification of ξ as above would result in another optimal solution ξ'' with an $h_{\xi''}$ having strictly smaller weights on the edges $(i_j, i_{j+1}), \ldots, (i_{m-1}, i_m)$, implying the same contradiction as above. □

Lemma 5.13. *If $S \subset V$, $0 \notin S$ and $S \cap P \neq \emptyset$, then $N(X^S) \geq N(X^{S \setminus P})$ for all rooted nooses with $\xi_N > 0$.*

Proof. Follows immediately by Lemma 5.12. □

Lemma 5.14. *Let ξ be an optimal solution of the Problem (**RNP**) such that h_ξ is a roof minor of f. Further, let $\hat{X}$ be a binary vector for which $g_\xi(\hat{X}) = 0$. Then for any subset $S \subset V$, $0 \notin S$ and $S \cap P \neq \emptyset$,*

$$f(X^S) > f(X^{S \setminus P}).$$

Proof. Follows readily from Lemmas 5.11 and 5.13. □

An easy consequence of Lemma 5.14 is the following claim.

Corollary 5.15. *Let ξ be an optimal solution of (**RNP**) such that h_ξ is a roof minor of f. Then at any minimum point X^* of f we have $g_\xi(X^*) = 0$, implying that the optimal values of x_i^* for $i \in P$ are uniquely determined.* □

Proof of the Persistency Theorem. Let ξ be an arbitrary optimal solution of (**RNP**), and let ξ' be a roof minor. Then we have $P = P_{\xi'} \supseteq P_\xi$ according to Remarks 5.5 and 5.8. Let us then consider $\xi'' = \frac{\xi + \xi'}{2}$. Obviously $P_{\xi''} = P_{\xi'} = P$. Furthermore, by Remark 5.7 we have $g_{\xi''} \geq \frac{1}{2}(g_\xi + g_{\xi'})$. Hence, for the binary vector X with $g_{\xi''}(X) = 0$ (the existence of which follows from the optimality of ξ'' for (**RNP**)), $g_\xi(X) = g_{\xi'}(X) = 0$. By Corollary 5.15 and Remark 2.11, $x_i = x_i^*$, $i \in P$ for any minimum point X^* of f. Therefore $g_\xi(X^*) = 0$ follows, too. □

Proof of the Structure Theorem. The statement follows easily from the above results for the ξ-decomposition of f for which h_ξ is a roof minor of f, which can be constructed from a maximal flow in N_f, as in Lemma 4.4, obtainable in $O(n^3)$ time. □

6. Iterated roof duality

Given a bi-form f, let

$$f = h_\xi + g_\xi + f'_\xi$$

be the ξ-decomposition of f for an optimal ξ. The bi-form f'_ξ is obviously independent of the root 0 of f. Designating an arbitrary variable of f'_ξ as the root of it, we can further decompose f'_ξ. In general, starting with $f^{(0)} = f$ we shall write

$$f^{(k)} = h_\xi + g_\xi + f^{(k+1)}$$

for some optimal ξ of the (**RNP**) problem corresponding to $f^{(k)}$, and defining $f^{(k+1)} = f_\xi^{(k)'}$.

Clearly, there is a k ($\leq n$) for which $f^{(k)} = 0$. Let us call $\hat{\rho}(f) = \sum_{i=0}^{k-1} \rho(f^{(i)})$ the *iterated roof dual* of f.

Remark 6.1. $\min f \geq \hat{\rho}(f)$, and $\hat{\rho}(f)$ can be computed in $O(n^4)$ time.

Proof. The above approach constructs a feasible solution for the noose packing problem (**NP**), which by Theorem 3.1 provides a lower bound to the minimum of f. Furthermore, in every iteration we designate one of the original variables as the root, and this variable is not involved in subsequent iterations. Thus, there are at most n iterations in the above procedure, from which the $O(n^4)$ time complexity follows by Corollary 4.6. □

Example 6. Returning to the bi-form of Example 1, we can see from Example 5 that after one iteration we have

$$f^{(1)} = 5(x_1\overline{x}_2 + \overline{x}_1 x_2) + 4(x_1 x_3 + \overline{x}_1\overline{x}_3) + 7(x_2\overline{x}_3 + \overline{x}_2 x_3).$$

Here we can declare x_1 as the root variable, and then $N = [(1, 2), (2, 3), (1, 3)]$ is the only rooted noose, for which we have $\xi_N = 4$ in the optimal solution, yielding $\rho(f^{(1)}) = 4$,

$$g_\xi = (x_1\overline{x}_2 + \overline{x}_1 x_2) + 3(x_2\overline{x}_3 + \overline{x}_2 x_3),$$

and $f^{(2)} = 0$. Thus, we have $\hat{\rho}(\phi) = 5 + 4$, i.e., $\hat{\rho}(f) = 9 - 13 = -4$, which in this particular case is the true minimum of f.

Let us remark that the above approach constructs a feasible (and not necessarily optimal) solution for the noose packing problem (**NP**), by starting with the roof dual value, by Lemma 3.17, and increasing it iteratively. Thus, by Remark 3.19 we have

$$\omega(\mathbf{NP}) = C_3 \geq \hat{\rho}(f) \geq \rho(f) = \omega(\mathbf{RNP}).$$

Let us add finally that in the recent publication [36] a branch-and-bound implementation is reported in which another efficient heuristic solution for the cycle packing problem (equivalently, for (**NP**)) is used to provide the bounds.

7. Computational experiments

We have carried out extensive computational experimentation of the proposed techniques using several benchmark problems available in the literature. Let us note that the maximization and the minimization of quadratic pseudo-Boolean functions are equivalent problems, since minimizing f is equivalent with maximizing $-f$. Thus, the presented results can be applied directly to maximization problems, as well. Following the literature, we also present our computational results in the context of maximization.

In order to compare the quality and the computing times obtained with our implementations of the roof dual and the iterated roof dual algorithms, we used the *SemiDefinite Relaxation*, or SDR (see Goemans and Williamson [22]). There are many publicly available semidefinite solvers on the Internet. Each solver has strengths and weaknesses, which are very much dependent on the type and size of the problem to be solved. There is no solver which clearly dominates the others in all aspects, e.g. robustness, memory management, or solution speed (see [40]). We have used the following solvers that have been proposed to solve *SemiDefinite Programs*, or SDPs:

SDPA – SDPA is a software package for solving SDPs (see e.g. [19]). It is an implementation of a Mehrotra-type primal-dual predictor-corrector interior-point method. The Windows version 6.2.1 of SPDA was used in this study.

DSDP – DSDP is a software implementation of the dual interior-point method for SDP (see e.g. [5]). It provides primal and dual solutions, exploits low-rank structure and sparsity in the data, and has relatively low memory requirements for an interior-point method. The version 5.8 of DSDP was used in this study.

SBM – SBM is a software implementation of the spectral bundle method (see e.g. [30,31]), for minimizing the maximum eigenvalue of an affine matrix function (real and symmetric). The code is suited for large scale problems. It allows us to exploit structural properties of the matrices such as sparsity and low-rank structure. The version 1.1.3 of SBM was considered in this study.

We did not have a special preference for selecting any of the above methods. Our goal was to cover a variety of SDR solution methods, e.g. by using a robust method that handles small and medium sized problems (like SPDA), by using a method that handles larger problems with a sparse structure (like DSDP), and by adopting a method (like SBM) that is quicker than the others (although this speedup is achieved at the price of obtaining an approximate solution to SDR).

Table 1
Characteristics of computer systems used for testing the algorithms

Computer systems	I	II	III
CPU	Intel Pentium 4	Xeon	Intel Pentium 4
Clock speed	2.8 GHz	3.06 GHz	3.6 GHz
Hyper-Threading?	yes	no	yes
RAM	512 MB	3.5 GB	2 GB
Cache	L_2 512 KB	L_2 512 KB	L_2 1 MB
Operating system	Wind. XP[a]	Linux[b]	Wind. XP[a]
Algorithms tested	RDA, IRDA, SDPA	DSDP, SBM	XPRESS

[a] Microsoft Windows XP Professional version 5.1.2600 (service pack 2).
[b] Fedora Core Linux 2.6.9-1.667smp i686.

Finally, we should mention that for a restricted number of problems we have added computational experience also for the case when the C_3 bound was used (see end of Section 3). Other than by using linear programming, we are unaware of any other approach that could provide C_3 in polynomial time. Since solving these large LP problems is time and memory consuming, we computed this bound only for one family of problems.

The method used to solve the LPs is the Newton–Barrier algorithm that comes with Xpress-MP 2005B (release 16.10.02). The presolve and the crossover was turned off in all runs. We use XPRESS in the text to identify the results returned by this particular LP solver, with the options previously mentioned.

The Roof Dual and the Iterated Roof Dual Algorithms (respectively called RDA and IRDA hereafter) 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) using the single-threaded run-time library.

Three computer systems were used for testing. The decision to use these many computers and not one, is related to licensing requirements, operating system restrictions, and amount of physical memory available. Table 1 shows the main characteristics of each system, and also shows which algorithm(s) were tested in each one of them. The three platforms are comparable in terms of speed with a maximum speedup smaller than two, between the fastest (Computer III) and the slowest machine (Computer I).

7.1. MAX-CUT

In this study we included the analysis of two large groups of problems, one involving maximum-cut problems (or MAX-CUT in short) on graphs (known to be equivalent to quadratic unconstrained binary maximization problems [11]), and another using randomly generated quadratic binary optimization problems proposed by Glover et al. [21] and Beasley [4].

The families of graphs used in the MAX-CUT experiments are listed below, along with their original references, and their brief descriptions:

- $G_{n,d}$ – Random graphs proposed by Kim and Moon [35]. Each graph has n vertices (n being 500 or 1000), and an edge is placed between two vertices with probability p, independently of other edges. The probability p is chosen so that the expected vertex degree is $d = p\,(n-1)$.
- $U_{n,d}$ – Random geometric graphs proposed by Kim and Moon [35]. Each graph has n vertices (n being 500 or 1000) that lie in the unit square and whose coordinates are chosen uniformly from the unit interval. There is an edge between two vertices if their Euclidean distance is t, which results in an expected vertex degree of $d = n\pi t^2$.
- R_n – Sparse random graphs proposed by Homer and Peinado [32]. Each graph has an edge probability of $10/n$, and the number of vertices n varies from 1000 to 8000. These graphs belong to the random graph class C in Goemans and Williamson [22].
- *via* – Graphs provided by Homer and Peinado [32], derived from layer assignment problems in the design process for VLSI chips. Each edge has a coefficient associated to it, some of them being negative.
- $sg3dl_L$ – 3D-toroidal graphs proposed by Burer, Monteiro and Zhang [17], consisting of thirty cubic lattices having randomly generated ± 1 interaction magnitudes. Each graph has a side length L, has $n = L^3$ vertices and $3n$ edges. There are ten graphs for each value of the side length L, which are the values 5, 10 and 14.

torus – 3D-toroidal graphs originated from the Ising model of spin glasses in physics. They were taken from the DIMACS library of mixed semidefinite-quadratic-linear programs[1] (see also [40]). Two graphs have ± 1 interaction magnitudes, whereas the other two graphs have interactions determined by a Gaussian distribution.

The first experiments concerned the finding of upper bounds for MAX-CUT in the 16 graphs of Kim et al. [35]. The results are shown in Table 4(a) and (b) in the Appendix. The tables show that in almost all these 16 cases the upper bound given by the semidefinite relaxation was slightly better (on the average 7.1% lower) than that given by the iterated roof dual, and better than (on the average 27% lower) the roof dual bound. The difference in the quality of the bounds was amply compensated by the computing times needed to find them. Indeed, on the average, the time needed by DSDP (the most efficient of the three implementations of semidefinite relaxations) was of 18.5 s, while that needed by the iterated roof dual algorithm was of only 1.5 s, and that needed by the roof dual algorithm was of 0.01 s.

Turning now to the MAX-CUT problem for the graphs of Homer and Peinado [32] (see Table 5(a) and (b) in the Appendix) we notice that the comparative values of roof dual-based versus semidefinite-relaxation-based upper bounds differ substantially between the group R of random graphs, and the group of *via* graphs coming from VLSI design. For the group R, the upper bounds of SDR are 21.1% better than those of roof duality, and 8.1% better than those coming from iterated roof duality. The situation of the *via* graphs is quite different, since the three upper bounds are quite comparable within this group. More precisely, the upper bounds of SDR are only 1.7% better than those of roof duality, but the upper bounds of iterated roof duality are 0.5% better than those of SDR. As far as computing times go the average time needed by SBM (the most efficient of the three implementations of SDR for the group of R graphs) was of 477.4 s, while for RDA the average time is 0.02 s and for IRDA it is of 35.8 s. For the group of *via* graphs, the average time needed by DSDP (the most efficient of the three implementations of SDR for this group) was of 42 s, while for RDA the average time is 0.03 s and for IRDA it is of 0.13 s.

The next group of MAX-CUT problems concerns cubic lattice graphs (similar in structure to the graphs appearing in Ising problems) of Burer et al. [17] (see Table 6(a) and (b) in the Appendix). It can be seen that for these graphs, the upper bound given by the iterated roof dual is 1.4% better than that given by SDR, while the SDR bound is 31.5% better than that given by the roof dual bound. It is interesting to note that the computing time required by SBM (the most efficient of the three implementations of SDR for the cubic lattices) for finding the upper bound associated to an average graph in the family was of 42.7 s, while that of IRDA was of 2.7 s, and that of RDA was less than 0.01 s.

The last group of MAX-CUT problems examined are associated to torus graphs (having also a similar structure to that of the graphs appearing in Ising problems) proposed at the 7th DIMACS Implementation Challenge on Semidefinite Programming, which are frequently used as benchmarks in computational studies concerning semidefinite programming (e.g. [16,17,29,40]; see Table 7(a) and (b) in the Appendix). For these graphs, the upper bound given by the iterated roof dual bound is 1.4% better than that given by SDR, which in its turn is 26.9% better than that given by roof duality. The average computing time required by SBM (the most efficient of the three implementations of SDR for the torus graphs) is 115.7 s, while that required by IRDA is of 4.5 s, and that required by RDA is of about 0.01 s.

In summary (see Table 2(a) and (b)), the upper bounds for MAX-CUT examined in this study present the following characteristics:

- In the case of G, U and R graphs the best bounds are obtained by SDR;
- In the case of via, $sg3dl$ and $torus$ graphs the best bounds are obtained by IRDA;
- The shortest computing times are those of RDA, followed by those of IRDA. The average computing time per graph is of 92.3 s for SDP (the fastest implementation of the considered semidefinite programs), 6.0 s for IRDA, and about 0.014 s for RDA.

7.2. Randomly generated quadratic binary optimization problems

The second group of problems includes standard randomly generated families of problems, having a constant density (i.e., proportion of coefficients with value zero) and having all non-zero coefficients from a closed interval. The following two families were considered:

[1] The DIMACS library of mixed semidefinite-quadratic-linear programs: http://dimacs.rutgers.edu/Challenges/Seventh/Instances/.

Table 2
Bounding MAX-CUT

(a) Average relative gap (g) to the largest known cut (z)

Family	Number of problems	SDR gap ($g = 1 - \varsigma/z$) (%)	Roof dual gap ($g = 1 - \rho/z$) (%)	Iter. roof dual gap ($g = 1 - \widehat{\rho}/z$) (%)
G graphs	8	**5.68**	28.79	11.04
U graphs	8	**2.64**	56.28	13.24
R graphs	8	**7.31**	36.01	16.72
via graphs	10	0.53	2.32	**0.02**
sg3dl graphs	30	14.86	67.65	**13.27**
torus graphs	4	12.47	54.26	**10.88**

(b) Average computing times

Family	Best	Roof dual algorithms	
	SDP[b] (s)	RDA[a] (s)	IRDA[a] (s)
G graphs	23.2	<0.01	0.9
U graphs	13.8	<0.01	2.2
R graphs	477.4	<0.02	35.8
via graphs	42.0	<0.03	0.1
sg3dl graphs	42.7	<0.01	2.7
torus graphs	115.7	<0.01	4.5

[a] Computed on computer system I.
[b] Computed on computer system II.

D – A set of 10 QUBO problems proposed by Glover et al. [21] having 100 variables per problem, and densities varying from 10% to 100% in steps of 10%.

ORL – A set of 60 QUBO test problems ($n = 50, 100, 250, 500, 1000, 2500$; 10 problems for each value of n) proposed by Beasley [4]. These problems have been randomly generated with constant characteristics (10% density; all linear and quadratic coefficients being integers, uniformly drawn respectively from $[-100, +100]$ and $[-200, +200]$). The problems with 50 variables turned out to be all solved to optimality by applying iterated roof duality (and in most cases even by applying roof duality). Therefore, these problems have been eliminated from the study. The best known solutions (i.e. lower bounds on the maximum) for the remaining problems were collected from the following other studies [4,34,39,42].

We present in Table 8(a) in the Appendix the maximum values of some randomly generated quadratic functions with binary variables, along with the values of four upper bounds to the maximum (SDR, RDA, IRDA and C3), expressed as percentages of the values of the corresponding exact maxima. It can be seen that the best bounds for problems with densities of at most 40% were provided by the cubic dual (averaging 2.2% over the maximum), while for problems having densities of 50% or higher the best upper bounds were given by SDR (averaging 7.6% above the maximum). On the other hand, the best computing times (see Table 8(b) in the Appendix) were achieved by RDA (averaging less than 0.01 s) and IRDA (averaging less than 0.2 s). It follows that for problems which have low densities, the most efficient methods may be those based on roof duality. It is worth noting that numerous problem classes (e.g., minimum vertex covers of planar graphs or power-law graphs, MAX-CUT of Ising problems) belong to this category.

In Table 9 in the Appendix we present three upper bounds obtained by SDR, roof duality and iterated roof duality for the 10% dense quadratic unconstrained binary optimization problems of Beasley [4] having up to 2500 variables. It can be seen that for the "small" problems, i.e. those with 100 variables, the bounds given by IRDA are the best among the three upper bounds considered; the average gap between IRDA and the true maximum of the function is of 3.3% (see Table 3(a)). However, for problems having 250 or more variables the best bounds are those given by SDR; the average gap between SDR and the best known solution (representing a lower bound to the maximum) is of 8.9%.

Table 3
10% dense quadratic unconstrained binary optimization problems (Beasley [4])

(a) Average relative gap (g) to the best known lower bound (z)

Family	Variables (n)	SDR gap $\left(g = \frac{\varsigma - z}{z}\right)$ (%)	Roof dual gap $\left(g = \frac{\rho - z}{z}\right)$ (%)	Iter. roof dual gap $\left(g = \frac{\widehat{\rho} - z}{z}\right)$ (%)
ORL-100	100	6.3	15.3	**3.3**
ORL-250	250	**7.6**	78.1	18.5
ORL-500	500	**9.0**	150.6	41.6
ORL-1000	1000	**9.5**	248.8	73.0
ORL-2500	2500	**9.4**	430.4	129.1

(b) Average computing times

Family	Semidefinite programs			Roof dual algorithms	
	DSDP[b] (s)	SBM[b] (s)	SDPA[a] (s)	RDA[a] (s)	IRDA[a] (s)
ORL-100	0.21	2.55	1.01	<0.005	0.01
ORL-250	2.20	28.88	14.98	0.02	0.24
ORL-500	24.97	115.56	131.21	0.05	2.38
ORL-1000	269.29	673.49	1096.21	0.20	21.47
ORL-2500	8385.05	107 623.67[d]	n/a[c]	1.49	416.78

[a] Computed on computer system I.
[b] Computed on computer system II.
[c] Memory exceeded for all problems with 2 500 variables.
[d] Computing time found for the first problem only.

As in the previous cases the computing times of the different methods follow a clear pattern. The average time needed by DSDP (the fastest of the three SRD procedures; see Table 3(b)) is of 1736.3 s. For the same problems, the average time required by RDA is of 0.4 s, and by IRDA is of 88.2 s.

8. Conclusions

The above results demonstrate that the iterated roof dual bound can be computed very efficiently with the proposed IRDA implementation. The computing time of this bound is much faster than the computation of semidefinite bounds or the cubic dual. We can also see that the quality of the iterated roof dual bound is highly competitive with other approaches. In particular, for sparse problems, which are quite frequent in applications, these bounds are superior to all other methods we tested, and can be computed on average 20–50 times faster than those. We can also see that the cubic dual bound is the best on a larger range of mostly sparser problems, however its time complexity makes its application impractical for larger problems.

Let us also add that by using a branch-and-bound implementation, which uses the iterated roof dual as a bounding procedure, we found optimal solutions for several of the benchmark problems. For instance, in a very small computing time this implementation found the MAX-CUT value of problem G500.2.5, the optimal MAX-CUT values of all *via* graphs, the optimal values of all problems with 100 variables from the ORL, and the optimal solutions of the group D of problems up to 40% density.

We are currently working on designing an optimal family of techniques for various combinatorial optimization problems (e.g. graph stability, maximum clique) using roof duality-based algorithms coupled with branching techniques. In spite of the fact that the bounds given by semidefinite relaxation are of high quality, the time and memory requirements of the roof duality-based methods being substantially smaller, assure the practical applicability of this latter group of methods, and guarantee their high efficiency. Moreover, in view of the typical sparsity of real life quadratic unconstrained binary optimization problems, the use of roof duality-based methods is both effective and efficient (see [13]).

Acknowledgements

The authors are grateful to Franz Rendl and Bruno Simeone for their helpful remarks and suggestions.

Appendix

See Tables 4–9.

Table 4
MAX-CUT (Kim et al. [35])

(a) Upper bounds

Family	Problem name	Vert. (n)	Edges	MAX-CUT	Upper bounds to the maximum		
					Semidefinite relaxation	Roof dual (ρ)	Iter. roof dual ($\widehat{\rho}$)
$G_{n,d}$	G500.2.5	500	625	574	598.15	620.5	**590.50**
	G500.05		1223	≥1008	**1 070.06**	1 217.0	1 086.00
	G500.10		2355	≥1735	**1 847.97**	2 346.0	1 960.64
	G500.20		5120	≥3390	**3 566.74**	5 103.0	4 006.70
	G1000.2.5	1000	1272	≥1173	1 223.01	1 268.5	**1 212.91**
	G1000.05		2496	≥2053	**2 191.80**	2 490.5	2 232.66
	G1000.10		5064	≥3705	**3 954.67**	5 052.5	4 245.18
	G1000.20		10 107	≥6729	**7 105.60**	10 090.0	8 059.02
$U_{n,d}$	U500.05	500	1282	900	**922.42**	1 274.0	962.00
	U500.10		2355	≥1 546	**1 587.86**	2 345.0	1 716.09
	U500.20		4549	≥2 783	**2 864.27**	4 534.0	3 229.47
	U500.40		8793	≥5 181	**5 303.45**	8 765.0	6 164.78
	U1000.05	1000	2394	≥1 711	**1 752.76**	2 388.5	1 830.25
	U1000.10		4696	≥3 073	**3 158.95**	4 686.5	3 424.81
	U1000.20		9339	≥5 737	**5 890.78**	9 319.5	6 617.69
	U1000.40		18 015	≥10 560	**10 851.01**	17 986.0	12 593.03

(b) Computing times

Problem	Semidefinite programs			Roof dual algorithms	
	DSDP[b] (s)	SBM[b] (s)	SDPA[a] (s)	RDA[a] (s)	IRDA[a] (s)
G500.2.5	3.35	1279.20	85.84	<0.005	0.05
G500.05	3.72	70.67	105.98	<0.005	0.09
G500.10	4.80	4.28	104.16	<0.005	0.31
G500.20	8.54	4.64	106.08	<0.005	0.84
G1000.2.5	17.85	4056.78	694.05	<0.005	0.23
G1000.05	29.56	138.02	843.69	<0.005	0.53
G1000.10	47.80	11.37	871.16	<0.005	1.47
G1000.20	69.98	15.00	826.34	0.02	3.36
U500.05	3.71	2179.61	2179.61	<0.005	0.17
U500.10	3.15	52.15	107.58	<0.005	0.38
U500.20	3.74	6.32	111.56	<0.005	0.89
U500.40	4.41	3.85	106.80	<0.005	1.81
U1000.05	21.96	3596.08	794.62	<0.005	0.82
U1000.10	20.79	403.90	837.03	<0.005	1.88
U1000.20	22.87	22.24	871.67	<0.005	3.91
U1000.40	29.90	13.87	941.30	0.02	7.66

[a] Computed on computer system I.
[b] Computed on computer system II.

Table 5
MAX-CUT (Homer and Peinado [32])

(a) Upper bounds

Family	Problem name	Vertices (n)	Edges	MAX-CUT	Upper bounds to the maximum		
					Semidefinite relaxation	Roof dual (ρ)	Iter. roof dual ($\widehat{\rho}$)
R	*R*1000	1000	5 033	≥3 687	**3 934.5**	5 021.5	4 220.96
	*R*2000	2000	9 943	≥7 308	**7 820.0**	9 932.5	8 454.98
	*R*3000	3000	14 965	≥10 997	**11 790.6**	14 953.5	12 800.36
	*R*4000	4000	19 939	≥14 684	**15 729.2**	19 927.0	17 118.74
	*R*5000	5000	24 794	≥18 225	**19 587.1**	24 783.0	21 362.02
	*R*6000	6000	29 862	≥21 937	**23 602.7**	29 849.0	25 798.04
	*R*7000	7000	35 110	≥25 763	**27 730.6**	35 097.0	30 363.51
	*R*8000	8000	39 642	≥29 140	**31 382.1**	39 629.5	34 375.45
via	via.c1n	828	1 389	6 150	6 182.42	6 339.0	**6 150.00**
	via.c2n	980	1 712	7 098	7 117.75	7 473.0	**7 098.00**
	via.c3n	1327	2 393	6 898	6 943.72	7 282.0	**6 906.25**
	via.c4n	1366	2 539	10 098	10 110.59	10 437.0	**10 098.00**
	via.c5n	1202	2 129	7 956	8 003.15	8 427.0	**7 962.00**
	via.c1y	829	1 693	7 746	7 795.87	**7 746.0**	**7 746.00**
	via.c2y	981	2 039	8 226	8 276.36	**8 226.0**	**8 226.00**
	via.c3y	1328	2 757	9 502	9 572.56	**9 502.0**	**9 502.00**
	via.c4y	1367	2 848	12 516	12 556.58	**12 516.0**	**12 516.00**
	via.c5y	1203	2 452	10 248	10 327.99	**10 248.0**	**10 248.00**

(b) Computing times

Problem	Semidefinite programs			Roof dual algorithms	
	DSDP[b] (s)	SBM[b] (s)	SDPA[a] (s)	RDA[a] (s)	IRDA[a] (s)
*R*1000	43.15	13.09	815.81	0.02	1.47
*R*2000	389.74	55.28	8053.94	0.02	5.75
*R*3000	8 751.11	102.93	n/a[c]	0.02	13.00
*R*4000	2 492.46	244.18	n/a[c]	0.02	22.74
*R*5000	4 625.69	352.35	n/a[c]	0.02	35.59
*R*6000	9 055.98	472.75	n/a[c]	0.03	50.50
*R*7000	13 566.78	1725.05	n/a[c]	0.03	69.23
*R*8000	18 212.75	853.85	n/a[c]	0.05	87.92
via.c1n	14.23	92.94	506.45	0.02	0.09
via.c2n	20.02	133.56	825.53	0.02	0.39
via.c3n	47.97	376.64	2242.39	0.02	0.16
via.c4n	51.55	239.53	2431.20	0.03	0.19
via.c5n	34.69	227.62	1644.06	0.00	0.13
via.c1y	19.97	493.52	576.74	0.03	0.05
via.c2y	28.83	636.06	944.25	0.03	0.05
via.c3y	70.66	2502.97	2363.20	0.05	0.08
via.c4y	77.51	1098.55	2849.97	0.05	0.06
via.c5y	54.55	3716.97	1759.91	0.03	0.06

[a] Computed on computer system I.
[b] Computed on computer system II.
[c] Memory exceeded.

Table 6
MAX-CUT on cubic lattice graphs (Burer et al. [17])

(a) Upper bounds

Family	Problem name	Vert. (n)	Edges	MAX-CUT	Upper bounds to the maximum		
					Semidefinite relaxation	Roof dual (ρ)	Iter. roof dual ($\widehat{\rho}$)
sg3dl05	sg3dl051000	125	375	110	126.53	185	**126.38**
	sg3dl052000	125	375	112	128.20	185	**127.73**
	sg3dl053000	125	375	106	**123.98**	185	126.50
	sg3dl054000	125	375	114	128.18	185	**124.09**
	sg3dl055000	125	375	112	127.06	185	**129.13**
	sg3dl056000	125	375	110	126.88	185	**128.25**
	sg3dl057000	125	375	112	126.81	185	**126.00**
	sg3dl058000	125	375	108	125.48	185	**123.38**
	sg3dl059000	125	375	110	126.00	185	**127.25**
	sg3dl0510000	125	375	112	127.68	185	**124.31**
sg3dl10	sg3dl101000	1000	3000	$\geq$896	1025.91	1497	**1001.31**
	sg3dl102000	1000	3000	$\geq$900	1036.47	1497	**1008.46**
	sg3dl103000	1000	3000	$\geq$892	1021.92	1497	**1003.93**
	sg3dl104000	1000	3000	$\geq$898	1031.34	1497	**1011.13**
	sg3dl105000	1000	3000	$\geq$886	1021.29	1497	**1001.17**
	sg3dl106000	1000	3000	$\geq$888	1023.34	1497	**1001.66**
	sg3dl107000	1000	3000	$\geq$900	1030.06	1497	**1014.06**
	sg3dl108000	1000	3000	$\geq$882	1023.74	1497	**1006.17**
	sg3dl109000	1000	3000	$\geq$902	1029.24	1497	**1010.40**
	sg3dl1010000	1000	3000	$\geq$894	1027.65	1497	**1005.20**
sg3dl14	sg3dl141000	2744	8232	$\geq$2446	2816.90	4113	**2773.56**
	sg3dl142000	2744	8232	$\geq$2458	2825.79	4113	**2762.61**
	sg3dl143000	2744	8232	$\geq$2442	2815.40	4113	**2762.61**
	sg3dl144000	2744	8232	$\geq$2450	2817.45	4113	**2764.10**
	sg3dl145000	2744	8232	$\geq$2446	2809.86	4113	**2772.49**
	sg3dl146000	2744	8232	$\geq$2450	2822.92	4113	**2765.19**
	sg3dl147000	2744	8232	$\geq$2444	2813.08	4113	**2757.21**
	sg3dl148000	2744	8232	$\geq$2446	2818.70	4113	**2771.18**
	sg3dl149000	2744	8232	$\geq$2424	2793.42	4113	**2744.38**
	sg3dl1410000	2744	8232	$\geq$2458	2826.35	4113	**2763.24**

(b) Average computing times

Family	Semidefinite programs			Roof dual algorithms	
	DSDP[b] (s)	SBM[b] (s)	SDPA[a] (s)	RDA[a] (s)	IRDA[a] (s)
sg3dl05	0.19	0.87	1.72	<0.005	0.01
sg3dl10	25.29	14.07	871.23	<0.005	0.91
sg3dl14	431.17	113.09	n/a[c]	0.01	7.16

[a] Computed on computer system I.
[b] Computed on computer system II.
[c] Memory exceeded.

Table 7
MAX-CUT for torus graphs (7th DIMACS Implementation Challenge)

(a) Upper bounds

Problem name	Vert. (n)	Edges	MAX-CUT	Upper bounds to the maximum		
				Semidefinite relaxation	Roof dual (ρ)	Iter. Roof dual ($\widehat{\rho}$)
toruspm3-8-50	512	1 536	≥458	527.81	765.0	**523.05**
toruspm3-15-50	3375	10 125	≥3 016	3 475.13	5 060.0	**3 414.49**
torusg3-8	512	1 536	41 684 814	45 735 854.8	58 921 474.5	**45 100 733.03**
torusg3-15	3375	10 125	≥285 790 637	313 457 107.3	402 667 673.0	**308 433472.25**

(b) Computing times

Problem	Semidefinite programs			Roof dual algorithms	
	DSDP[b] (s)	SBM[b] (s)	SDPA[a] (s)	RDA[a] (s)	IRDA[a] (s)
toruspm3-8-50	4.16	5.00	114.78	<0.005	0.22
toruspm3-15-50	763.58	226.89	n/a[c]	0.02	10.92
torusg3-8	8.03	6.35	186.98	<0.005	0.14
torusg3-15	1301.45	224.62	n/a[c]	0.02	6.70

[a] Computed on computer system I.
[b] Computed on computer system II.
[c] Memory exceeded.

Table 8
100 variable quadratic unconstrained binary optimization (Glover et al. [21])

(a) Upper bounds

Problem name	Density (d) (%)	Maximum	Upper bounds to the maximum			
			Semidefinite relaxation	Roof dual (ρ)	Iter. roof dual ($\widehat{\rho}$)	Cubic dual
1d	10	6 333	6 592.77	7 063.50	6 424.50	**6 333.00**
2d	20	6 579	7 234.24	12 297.00	7 791.00	**6 709.82**
3d	30	9 261	9 962.97	18 053.50	10 875.25	**9 374.79**
4d	40	10 727	11 592.46	25 156.50	13 425.50	**11 321.82**
5d	50	11 626	**12 632.10**	30 732.00	15 538.13	13 044.50
6d	60	14 207	**15 235.31**	37 334.50	18 041.50	15 664.33
7d	70	14 476	**15 671.97**	44 171.50	20 614.75	18 340.00
8d	80	16 352	**17 353.30**	50 239.50	22 723.50	20 625.67
9d	90	15 656	**17 010.86**	55 130.00	24 109.00	21 753.67
10d	100	19 102	**20 421.35**	63 830.50	28 370.50	25 951.67

(b) Computing times

Problem	Semidefinite programs			Roof dual algorithm		Cubic dual LP
	DSDP[b] (s)	SBM[b] (s)	SDPA[a] (s)	RDA[a] (s)	IRDA[a] (s)	XPRESS[c] (s)
1d	0.22	1.72	1.03	<0.005	<0.005	118
2d	0.26	2.68	1.06	<0.005	0.02	164
3d	0.23	4.91	1.08	<0.005	0.05	173
4d	0.22	3.16	1.11	<0.005	0.08	143
5d	0.22	4.95	1.17	0.02	0.11	122
6d	0.20	8.90	1.08	<0.005	0.14	74
7d	0.22	6.87	1.17	0.02	0.20	71
8d	0.23	11.19	1.19	0.02	0.25	72
9d	0.23	8.78	1.20	<0.005	0.27	68
10d	0.23	9.35	1.25	<0.005	0.33	69

[a] Computed on computer system I.
[b] Computed on computer system II.
[c] Computed on computer system III.

Table 9
Upper bounds of 10% dense quadratic unconstrained binary optimization problems (Beasley [4])

Family (n)	Problem number	Maximum	Upper bounds to the maximum		
			Semidefinite relaxation	Roof dual (ρ)	Iter. roof dual ($\widehat{\rho}$)
ORL-100	1	7 970	**8 721.11**	10 160.5	8 725.50
	2	11 036	11 704.18	12 285.5	**11 245.50**
	3	12 723	13 336.70	13 664.5	**12 864.00**
	4	10 368	10 927.93	12 099.0	**10 656.00**
(100)	5	9 083	9 736.93	10 617.0	**9 339.50**
	6	10 210	11 073.07	13 086.5	**11 042.00**
	7	10 125	10 906.86	12 016.5	**10 489.00**
	8	11 435	12 078.48	12 638.0	**11 542.06**
	9	11 455	11 926.97	12 235.0	**11 581.00**
	10	12 565	13 151.28	13 686.0	**12 749.00**
ORL-250	1	45 607	**48 732.37**	78 321.0	52 528.63
	2	44 810	**48 093.50**	78 258.5	52 728.61
	3	49 037	**51 745.40**	80 919.0	55 145.06
	4	41 274	**44 391.58**	75 411.0	49 577.34
(250)	5	47 961	**50 803.63**	79 972.5	54 165.75
	6	$\geq$41 014	**44 547.53**	78 452.5	50 704.70
	7	46 757	**49 709.76**	80 040.0	54 096.54
	8	$\geq$35 726	**40 005.60**	72 599.5	46 508.41
	9	48 916	**52 330.23**	81 838.5	56 244.03
	10	40 442	**44 026.14**	75 752.5	49 320.57
ORL-500	1	$\geq$116 586	**128 402.72**	308 706.5	171 327.46
	2	$\geq$128 339	**138 237.20**	309 825.5	175 209.92
	3	$\geq$130 812	**140 738.05**	317 653.5	181 089.10
	4	$\geq$130 097	**141 602.11**	315 733.0	180 490.21
(500)	5	$\geq$125 487	**136 578.72**	311 891.5	176 444.61
	6	$\geq$121 772	**132 960.20**	310 139.5	173 953.18
	7	$\geq$122 201	**134 273.56**	312 285.5	175 700.29
	8	$\geq$123 559	**135 438.79**	313 878.5	177 959.90
	9	$\geq$120 798	**132 615.73**	312 183.0	175 938.65
	10	$\geq$130 619	**141 076.28**	317 514.5	180 860.23
ORL-1000	1	$\geq$371 438	**403 684.0**	1256 488.0	627 769.04
	2	$\geq$354 932	**390 028.8**	1251 578.0	619 465.32
	3	$\geq$371 236	**404 445.7**	1263 836.0	628 844.76
	4	$\geq$370 675	**403 911.4**	1269 344.0	633 482.54
(1000)	5	$\geq$352 760	**388 304.0**	1260 413.5	622 233.15
	6	$\geq$359 629	**392 175.5**	1257 474.5	620 005.81
	7	$\geq$371 193	**405 621.7**	1259 282.5	628 620.56
	8	$\geq$351 994	**388 940.6**	1253 255.0	620 828.19
	9	$\geq$349 337	**385 204.7**	1254 976.0	617 834.57
	10	$\geq$351 415	**385 664.6**	1240 515.5	613 573.00
ORL-2500	1	$\geq$1515 944	**1652 473.3**	7886 424.0	3417 034.96
	2	$\geq$1471 392	**1614 710.7**	7843 106.0	3384 231.31
	3	$\geq$1414 192	**1558 172.2**	7810 572.5	3354 353.86
	4	$\geq$1507 701	**1642 588.4**	7860 349.5	3408 753.16
(2500)	5	$\geq$1491 816	**1626 210.5**	7858 834.0	3390 133.74
	6	$\geq$1469 162	**1608 890.8**	7827 394.0	3377 820.75
	7	$\geq$1479 040	**1619 037.2**	7852 577.0	3386 395.44
	8	$\geq$1484 199	**1616 263.5**	7831 767.5	3381 058.32
	9	$\geq$1482 413	**1622 399.3**	7868 242.0	3400 093.17
	10	$\geq$1483 355	**1625 693.3**	7840 749.5	3392 232.40

References

[1] W.P. Adams, P.M. Dearing, On the equivalence between roof duality and lagrangian duality for unconstrained 0–1 quadratic programming, Discrete Applied Mathematics 48 (1994) 1–20.

[2] B. Aspvall, M.F. Plass, R.E. Tarjan, A linear time algorithm for testing the truth of certain quantified boolean formulas, Information Processing Letters 8 (1979) 121–123.

[3] F. Barahona, The max-cut problem in graphs not contractible to k_5, Operations Research Letters 2 (1983) 107–111.

[4] J.E. Beasley, Heuristic algorithms for the unconstrained binary quadratic programming problem, Tech. Rep., Management School, Imperial College, London, UK, 1998.

[5] S.J. Benson, Y. Ye, X. Zhang, Solving large-scale sparse semidefinite programs for combinatorial optimization, SIAM Journal on Optimization 10 (2) (2000) 443–461.

[6] A. Billionnet, A. Sutter, Persistency in quadratic 0–1 optimization, Mathematical Programming, Series B 54 (1) (1992) 115–119.

[7] P. Bonami, M. Minoux, Exact max-2sat solution via lift-and-project closure, Operations Research Letters 34 (2006) 387–393.

[8] E. Boros, Y. Crama, P.L. Hammer, Upper-bounds for quadratic 0–1 maximization, Operations Research Letters 9 (1990) 73–79.

[9] E. Boros, Y. Crama, P.L. Hammer, Chvátal cuts and odd cycle inequalities in quadratic 0–1 optimization, SIAM Journal on Discrete Mathematics 5 (2) (1992) 163–177.

[10] E. Boros, P.L. Hammer, A max-flow approach to improved roof-duality in quadratic 0–1 minimization, Research Report RRR 15-1989, RUTCOR, Rutgers University, 1989.

[11] E. Boros, P.L. Hammer, The max-cut problem and quadratic 0–1 optimization, polyhedral aspects, relaxations and bounds, Annals of Operations Research 33 (1991) 151–180.

[12] E. Boros, P.L. Hammer, Pseudo-boolean optimization, Discrete Applied Mathematics 123 (2002) 155–225.

[13] E. Boros, P.L. Hammer, G. Tavares, Preprocessing of unconstrained quadratic binary optimization, Research Report RRR 10-2006, RUTCOR, Rutgers University, 2006.

[14] J.M. Bourjolly, Integral and fractional node-packings, and pseudo-boolean programs, Ph.D. Thesis, University of Waterloo, 1986.

[15] J.M. Bourjolly, P.L. Hammer, W.R. Pulleyblank, B. Simeone, Boolean-combinatorial bounding of maximum 2-satisfiability, in: Computer Science and Operations Research, Williamsburg, VA 1992, Pergamon, Oxford, 1999, pp. 23–42.

[16] S. Burer, R.D.C. Monteiro, A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization 95 (2) (2003) 329–357.

[17] S. Burer, R.D.C. Monteiro, Y. Zhang, Rank-two relaxation heuristics for max-cut and other binary quadratic programs, SIAM Journal on Optimization 12 (2) (2001) 503–521.

[18] G. Cornuéjols, Y. Li, Elementary closures for integer programs, Operations Research Letters 28 (2001) 1–8.

[19] K. Fujisawa, M. Fukuda, M. Kojima, K. Nakata, Numerical evaluation of the sdpa (semidefinite programming algorithm), in: H. Frenk, K. Roos, T. Terlaky, S. Zhang (Eds.), High Performance Optimization, Kluwer Academic Press, 2000, pp. 267–301.

[20] F. Glover, B. Alidaee, C. Rego, G. Kochenberger, One-pass heuristics for large-scale unconstrained binary quadratic problems, European Journal of Operational Research 137 (2) (2002) 272–287.

[21] F. Glover, G. Kochenberger, B. Alidaee, Adaptative memory tabu search for binary quadratic programs, Management Science 44 (3) (1998) 336–345.

[22] M.X. Goemans, D.P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, Journal of the ACM 42 (1995) 1115–1145.

[23] P.L. Hammer, Pseudo-boolean remarks on balanced graphs, International Series of Numerical Mathematics 36 (1977) 69–78.

[24] P.L. Hammer, P. Hansen, B. Simeone, Roof duality, complementation and persistency in quadratic 0–1 optimization, Mathematical Programming 28 (1984) 121–155.

[25] P.L. Hammer, B. Kalantari, A bound on the roof duality gap, Combinatorial Optimization, Lecture Notes in Mathematics 1403 (1989) 254–257.

[26] P. Hansen, Methods of nonlinear 0–1 programming, Annals of Discrete Mathematics 5 (1979) 53–70.

[27] P. Hansen, S.H. Lu, B. Simeone, On the equivalence of paved-duality and standard linearization in nonlinear optimization, Discrete Applied Mathematics 29 (1990) 187–193.

[28] F. Harary, On the notion of balance of a signed graph, The Michigan Mathematical Journal 2 (1953–54) 143–146.

[29] C. Helmberg, Numerical evaluation of sbmethod, Mathematical Programming 95 (2) (2003) 381–406.

[30] C. Helmberg, K.C. Kiwiel, A spectral bundle method with bounds, Mathematical Programming 93 (2002) 173–194.

[31] C. Helmberg, F. Rendl, A spectral bundle method for semidefinite programming, SIAM Journal on Optimization 10 (3) (2000) 673–696.

[32] S. Homer, M. Peinado, Design and performance of parallel and distributed approximation algorithms for maxcut, Journal of Parallel and Distributed Computing 46 (1997) 48–61.

[33] D.R. Karger, M.S. Levine, Finding maximum flows in undirected graphs seems easier than bipartite matching, in: Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, 1998, pp. 69–78.

[34] K. Katayama, H. Narihisa, Performance of simulated annealing-based heuristic for the unconstrained binary quadratic programming problem, European Journal of Operational Research 134 (2001) 103–119.

[35] S.H. Kim, Y.H. Kim, B.R. Moon, A hybrid genetic algorithm for the max cut problem, in: Genetic and Evolutionary Computation Conference, 2001, pp. 416–423.

[36] M. Koga, Yagiura, K. Nonobe, T. Imamichi, H. Nagamochi, T. Ibaraki, Efficient branch-and-bound algorithms for weighted max-2-sat, in: Proceedings of the Third International Asian Applied Computing Conference, AACC, Nepal, 2005.

[37] S.H. Lu, A.C. Williams, Roof duality and linear relaxation for quadratic and polynomial 0–1 optimization, Research Report RRR 8-1987, RUTCOR, Rutgers University, 1987.
[38] S.H. Lu, A.C. Williams, Roof duality for 0–1 nonlinear optimization, Mathematical Programming 37 (1987) 357–360.
[39] P. Merz, B. Freisleben, Genetic algorithms for binary quadratic programming, in: Proceedings of the 1999 International Genetic and Evolutionary Computation Conference, GECCO'99, 1999, pp. 417–424.
[40] H.D. Mittelmann, An independent benchmarking of sdp and socp solvers, Mathematical Programming.
[41] M. Padberg, The boolean quadric polytope: Some characteristics, facets and relatives, Mathematical Programming 45 (1989) 139–172.
[42] G. Palubeckis, Multistart tabu search strategies for the unconstrained binary quadratic optimization problem, Annals of Operations Research 131 (2004) 259–282.
[43] X. Sun, Combinatorial algorithms for boolean and pseudo-Boolean functions, Ph.D. Thesis, RUTCOR, Rutgers University, 1991.
[44] A.C. Williams, Quadratic 0–1 programming using the roof dual with computational results, Research Report RRR 8-1985, RUTCOR, Rutgers University, December 1985.