Available online at www.sciencedirect.com

SCIENCE DIRECT

Discrete Applied Mathematics 145 (2004) 52–71

DISCRETE APPLIED MATHEMATICS

www.elsevier.com/locate/dam

Block linear majorants in quadratic 0–1 optimization

Endre Boros[a,*], Isabella Lari[b], Bruno Simeone[b]

[a] RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854-8003, USA

[b] Dipartimento di Statistica, Probabilità e Statistiche Applicate, Universitá di Roma "La Sapienza", Piazzale Aldo Moro 5, Rome 00185 Italy

Received 15 December 2000; received in revised form 21 November 2002; accepted 19 September 2003

Abstract

A usual technique to generate upper bounds on the optimum of a quadratic 0–1 maximization problem is to consider a linear majorant (LM) of the quadratic objective function f and then solve the corresponding linear relaxation. Several papers have considered LMs obtained by termwise bounding, but the possibility of bounding *groups* of terms simultaneously does not appear to have been given much attention so far. In the present paper a broad and flexible computational framework is developed for implementing such a strategy. Here is a brief overview of our approach: in the first place, a suitable collection of "elementary" quadratic functions of few variables (typically, 3 or 4) is generated. All the coefficients of any such function (*block*) are either 1 or -1, and agree in sign with the corresponding coefficients of the given quadratic function. Next, for each block, a tightest LM (i.e., one having the same value as the block in as many points as possible), or a closest LM (i.e., one minimizing the sum of slacks) is computed. This can be accomplished through the solution of a small mixed-integer program, or a small linear program, respectively. Finally, the objective function is written as a weighted sum of blocks, with non-negative weights. Replacing in this expression each block by the corresponding LM yields an LM of f. We shall choose the weights in this process so that the maximum value of the resulting linear function is as small as possible. This amounts to a large-scale (but still polynomial-size) linear program, which may be solved exactly or, for larger instances, approximately by truncated column generation. The results of a set of 480 numerical tests with up to 200 variables are presented: the upper bounds on the quadratic optimum obtained by the above procedure are (provably) never worse, and often turn out to be substantially sharper, than those resulting from termwise bounding. Moreover, our bounds turn out to be close to the optimum in many (although not all) instances of some well-known benchmarks.

Keywords: Quadratic 0–1 optimization; Linear majorants; Roof duality; Column generation

1. Introduction

The present paper deals with the problem of finding tight upper bounds of the optimum value z^* of the *quadratic 0–1 maximization problem*

$$\max_{x \in \mathbf{B}^n} x^T Q x, \tag{1}$$

where Q is an $n \times n$ upper triangular real matrix with null diagonal entries, and $\mathbf{B} = \{0, 1\}$. Such problems are known to be NP-hard [14], in general. Thus, it makes sense to try to find good bounds of z^* at a reasonable computational cost.

* Corrresponding author.

E-mail addresses: boros@rutcor.rutgers.edu (E. Boros), isabella.lari@uniroma1.it (I. Lari), bruno.simeone@uniroma1.it (B. Simeone).

0166-218X/$ - see front matter
doi:10.1016/j.dam.2003.09.007

A usual method to generate upper bounds of z^* is to consider a *linear majorant* (abbr. LM), that is, a linear function $c_0 + \boldsymbol{cx}$ such that $c_0 + \boldsymbol{cx} \geqslant \boldsymbol{x}^{\mathrm{T}}Q\boldsymbol{x}$ for all $\boldsymbol{x} \in \mathbf{B^n}$. Then the optimal value of the *linear relaxation* of (1),

$$c_0 + \max_{x \in \mathbf{B^n}} \boldsymbol{cx} \tag{2}$$

is an upper bound of z^*. The simplest way to obtain an LM is to majorize each individual quadratic term $q_{ij}x_ix_j$ $(i < j)$ by a linear function of the form $a_{ij}x_i + b_{ij}x_j + c_{ij}$, where the parameters a_{ij}, b_{ij}, c_{ij} must be chosen so that

$$c_{ij} \geqslant 0, \quad a_{ij} + c_{ij} \geqslant 0, \quad b_{ij} + c_{ij} \geqslant 0, \quad a_{ij} + b_{ij} + c_{ij} \geqslant q_{ij} \tag{3}$$

and then add up all such linear terms. The LMs obtained in this way are called *paved upper planes* in [19]. An important subclass are the *roofs*, obtained through the majorization of each term $q_{ij}x_ix_j$ by either $\lambda_{ij}x_i + (q_{ij} - \lambda_{ij})x_j$ or $\lambda_{ij}(1 - x_i - x_j)$, where $0 \leqslant \lambda_{ij} \leqslant |q_{ij}|$, depending on whether q_{ij} is positive or negative. Among the paved upper planes, the roofs have the property that they minimize the sum of the slacks in inequalities (3). The *roof-dual* problem consists in finding a *best* roof, i.e., one that makes the optimal value of the corresponding linear relaxation as small as possible. The theory of roof-duality was set forth in [15], and further carried on in [1,7–10,16–20]. Several characterizations of the roof-dual optimum were given: in particular, it was shown in [19], that it coincides with the smallest possible value of (2) when $c_0 + \boldsymbol{cx}$ is a paved upper plane.

However, the possibility of bounding *groups* of terms, rather than individual ones, by an LM does not appear to have been given much attention so far. Actually, this idea is not entirely new: in [5] LMs of groups of terms involving at most k variables, for a fixed k, are introduced and some theoretical properties of the resulting upper bounds as k increases are investigated. The special case $k = 3$ is analyzed more closely in a subsequent paper [6]. However, as far as we know, computational issues have not been addressed in the literature. In the present paper a general and flexible algorithmic framework is developed for generating tight LMs of groups of terms, and for exploiting them in order to produce upper bounds on the quadratic optimum that are as sharp as possible. Our experiments clearly indicate that these bounds are often substantially sharper than the roof-dual bound.

We demonstrate the potential of the idea of bounding groups of terms by the following simple example. Consider the quadratic function of three binary variables

$$x_1x_2 + x_1x_3 - x_2x_3. \tag{4}$$

It is easy to check that x_1 is an LM, but not a roof, of (4). As a matter of fact, x_1 turns out to be an extremely tight majorant of (4), since its values coincide with those of (4) in 6, out of 8, points of $\mathbf{B}^3$. For the sake of comparison, the only best roof in this case is the constant 1, which coincides with (4) only in 3 points.

In the present paper, pursuing the strategy of bounding groups of terms, we take the following approach:

(1) A suitable collection $\boldsymbol{F}$ of "elementary" quadratic functions (*blocks*) of at most p variables (typically, p is equal to 3 or 4) is generated. We shall consider blocks with coefficients ± 1 according to the signs of the corresponding coefficients of the given function $f(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}Q\boldsymbol{x}$.

(2) For each block in $\boldsymbol{F}$ a tightest LM (where "tightest" means "having the same value as the block in as many points as possible") or a closest one (where "closest" means "minimizing the sum of slacks") is generated. This can be accomplished through the solution of a small mixed-integer program, or a small linear program, respectively. Actually, one needs to solve such a program only once and for all for each "template" block from a small-size catalogue independent of f.

(3) Then, the objective function $f(\boldsymbol{x})$ is written as a weighted sum (with non-negative weights) of these blocks. Clearly, if in this expression each block is replaced by the corresponding tightest, or closest LM, an LM of $f(\boldsymbol{x})$ of the form $c_0 + \boldsymbol{cx}$ is obtained. We shall then choose the weights so that the maximum value (2) of the resulting linear function is as small as possible. This optimization problem amounts to a large-scale (but still with size polynomial in n for fixed p) linear program (abbr. LP), which can be solved exactly or approximately by a variety of techniques, e.g., by straight column generation or some heuristic variant of it.

The paper is structured as follows. In Section 2, an appropriate notational framework is established, and basic definitions are given. Section 3 deals with properties that "good" collections of blocks are required to have. Section 4 shows how to compute both tightest and closest LMs; the section includes a full catalogue of tightest LMs for all possible "template" blocks with up to 4 variables. Section 5 deals with the problem of choosing optimal weights, as indicated above. A flexible column generation algorithmic scheme for the ensuing LP is described. The section ends with a brief analysis of the worst-case complexity of the different stages of our procedure, which is shown to run in overall polynomial time under the assumption that the maximum number of variables in a block of $\boldsymbol{F}$ is fixed. Finally, in Section 6 we report on the results of our numerical experimentation on 480 test problems with up to 200 variables. The paper is completed by two appendixes: in Appendix A some properties of tightest and closest LMs are discussed, and several conjectures about them are presented; in Appendix B some results about the maximum clique problem are given.

2. Basic notation and definitions

Let $f(\boldsymbol{x})=\boldsymbol{x}^{\mathrm{T}}Q\boldsymbol{x}$ be the given objective function to be maximized over $\mathbf{B^n}$. One can associate with f a signed undirected graph $G \equiv G_f = (V,E)$ as follows: the vertex-set V is the standard set $\{1,\ldots,n\}$ and the edge-set E is partitioned into two sets E^+ and E^-, where

$$E^+ = \{(i,j): 1 \leqslant i < j \leqslant n, \quad q_{ij} > 0\}, \quad E^- = \{(i,j): 1 \leqslant i < j \leqslant n, \quad q_{ij} < 0\}.$$

The unordered pair $\{i,j\}$ is simply denoted by ij and identified with the ordered pair (i,j), $i<j$.

We denote by $m=|E|$ the number of edges of G.

The *support* of G is the ordinary (unsigned) undirected graph obtained from G when the partition $E=E^+\cup E^-$ is disregarded.

Set

$$\sigma_{ij} = \begin{cases} -1 & \text{if } ij \in E^-, \\ +1 & \text{if } ij \in E^+. \end{cases}$$

A *signed isomorphism* is any isomorphism between two signed graphs that preserves the edge signs σ_{ij}. A *block* is a connected signed subgraph $B=(V(B),E(B))$ of G, with $E(B)=E^+(B)\cup E^-(B)$. If $x\in\mathbf{B^n}$ and B is any block, we denote by $\boldsymbol{x}_B$ the vector $(x_{i_1},\ldots,x_{i_r})$, where $i_1,\ldots,i_r$ are the elements of $V(B)$, taken in increasing order. For a block B, the associated *block function* (often also called a block for short) is the quadratic pseudo-boolean function

$$f_B(\boldsymbol{x}_B) = \sum_{ij\in E(B)} \sigma_{ij}x_ix_j. \tag{5}$$

Next, we define the convenient notions of "template" and "block-template". For any given positive integer p, consider a collection $\boldsymbol{C}$ of undirected connected graphs, all with vertex-set $\{1,\ldots,p\}$, such that every connected graph with p vertices is isomorphic to one and only one graph in $\boldsymbol{C}$. For each graph H in $\boldsymbol{C}$, consider the set of all signed graphs whose support is H, and partition this set into equivalence classes of sign-isomorphic graphs. Choose in each equivalence class a unique representative: any such representative will be called a *template* of *order p*.

Every connected signed graph is sign-isomorphic to a unique template. Hence with any given block B of the signed graph G one can associate a unique sign-isomorphic template $B^{\mathrm{t}}=(V(B^{\mathrm{t}}),\, E(B^{\mathrm{t}}))$. Any such B^{t} is called a *block-template* (relative to G). Notice that in general B^{t} is a block of K_n, the complete graph on n vertices, but not a block of G. A similar definition applies to block functions. For example, a common template of the three blocks $x_1x_5+x_1x_4-x_4x_5$, $x_3x_4+x_3x_6-x_4x_6$, and $x_1x_2+x_1x_3-x_2x_3$ is $-x_1x_2+x_1x_3+x_2x_3$, say.

Now let $g(\boldsymbol{x})=c_0+\boldsymbol{c}\boldsymbol{x}$ be an LM of $f(\boldsymbol{x})$. The *contact* of f and g is the set

$$Cont(f,g) = \{\boldsymbol{x}\in\mathbf{B^n} : g(\boldsymbol{x}) = f(\boldsymbol{x})\}. \tag{6}$$

The *excess* of g w.r.t. f is defined by

$$exc(f,g) = \sum_{x\in\mathbf{B^n}} (g(\boldsymbol{x}) - f(\boldsymbol{x})). \tag{7}$$

Let $\boldsymbol{M}$ be a family of LMs of f. The LM $g^*\in\boldsymbol{M}$ is said to be

- *tightest* (in $\boldsymbol{M}$) if $|Cont(f,g^*)| = \max\{|Cont(f,g)| : g\in\boldsymbol{M}\}$;
- *closest* (in $\boldsymbol{M}$) if $exc(f,g^*) = \min\{exc(f,g) : g\in\boldsymbol{M}\}$;
- *best* (in $\boldsymbol{M}$) if

$$\max_{x\in\mathbf{B^n}} g^*(\boldsymbol{x}) = \min_{g\in\mathrm{M}} \max_{x\in\mathbf{B^n}} g(\boldsymbol{x}). \tag{8}$$

We assume that all the above optima exist. This is certainly true, for example, if $\boldsymbol{M}$—regarded as a subset of $\mathbf{R}^{n+1}$—is closed and bounded.

3. Covering and exhaustive families of blocks

In the present section the question of choosing "good" collections $\boldsymbol{F}$ of blocks is addressed. Surely, in order to carry out the approach outlined in the introduction, the chosen family $\boldsymbol{F}$ must have the property that the objective function f

can be expressed as a weighted sum of the block functions f_B, $B \in \boldsymbol{F}$:

$$f(\boldsymbol{x}) = \sum_{B\in\boldsymbol{F}} w_B f_B(\boldsymbol{x}_B), \qquad \boldsymbol{x} \in \mathbf{B^n}, \tag{9}$$

where $w_B \geqslant 0$ for $B \in \boldsymbol{F}$. When this property holds the family $\boldsymbol{F}$ is said to be *exhaustive* for f. One obvious example of exhaustive family is the collection of all the edges of G, taken with their signs.

For each $ij \in E$, let $\boldsymbol{F}_{ij} = \{B \in \boldsymbol{F} : ij \in E(B)\}$. In view of (5), $\boldsymbol{F}$ is exhaustive for f if and only if the system of $\binom{n}{2}$ linear equations in the unknowns w_B

$$\sum_{B\in\boldsymbol{F}_{ij}} w_B = |q_{ij}|, \qquad ij \in E \tag{10}$$

has a non-negative solution.

Property 3.1. The family $\boldsymbol{F}$ is exhaustive for f if and only if there is no edge-weighting λ such that

$$\sum_{ij\in E(B)} \lambda_{ij} \geqslant 0, \qquad B \in \boldsymbol{F},$$

$$\sum_{ij\in E} \lambda_{ij}|q_{ij}| < 0. \tag{11}$$

Proof. Following Farkas's Lemma, system (10) has a non-negative solution if and only if there is no edge-weighting λ satisfying (11).

We shall say that $\boldsymbol{F}$ is a *covering family* for G if $\bigcup_{B\in\boldsymbol{F}} E^+(B) = E^+$ and $\bigcup_{B\in\boldsymbol{F}} E^-(B) = E^-$, i.e., if every edge of G belongs to some block of $\boldsymbol{F}$. Obviously, a necessary condition for $\boldsymbol{F}$ to be an exhaustive family for f is that $\boldsymbol{F}$ is a covering family for G_f.

Given a small positive integer p, one can use the following "skimming" procedure to obtain an exhaustive family of blocks with at most p vertices in each of them.

Skimming Algorithm

Step 1: Build the graph $G \equiv G_f$ associated with the input quadratic function $f(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}Q\boldsymbol{x}$;
  Let $H = G$;
  Let $R = Q$; {the entries of the current R are denoted r_{ij}}
  Let $F = \emptyset$;
Step 2: Find a block B of H, such that $|V(B)| \leqslant p$;
Step 3: Compute $\delta = \min\{|r_{ij}| : ij \in E(B)\}$;
Step 4: For each $ij \in E(B)$:
  Replace r_{ij} by $r_{ij} - \delta$;
  If the updated r_{ij} is zero, then delete edge ij from H;
Step 5: Add B to F;
  Set $w_B = \delta$;
Step 6: If $E(H) = \emptyset$ then stop: output F and w_B, for all $B \in F$;
  otherwise go to Step 2;

end

Notice that each block added to $\boldsymbol{F}$ must be different from all the subsequent ones, since it contains at least one edge that is missing from all of them.

The above algorithm runs in $O(mn^p)$ time for any fixed p. As a matter of fact, the number of iterations (additions of one block) is at most m, since at each iteration at least one edge is deleted from the residual graph H. Moreover, identifying each block B in Step 2 takes time polynomial in n for a fixed p.

4. Tightest and closest LMs of blocks

In this section, we discuss some techniques for generating tightest and closest LMs. Without loss of generality, we may assume that the block B under consideration is a template (see Section 2). A full catalogue of the TLMs of all template block functions with up to 4 variables will be given below. Such functions have been obtained by direct enumeration, taking into account the symmetries of the underlying graph.

Since B is a template, we can write $\boldsymbol{x}_B = \boldsymbol{x} = (x_1, \ldots, x_p)$, where $p = |V(B)|$. Accordingly, the block function (5) will be denoted simply by $h(\boldsymbol{x})$ and a TLM of it by $g(\boldsymbol{x}) = t_0 + \boldsymbol{tx}$.

Let $N = 2^p$, and let $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N$ be the complete list of the points of $\mathbf{B^p}$.

A TLM $t_0 + \boldsymbol{tx}$ of the block h can be generated through the solution of the following mixed-integer linear program:

$$\begin{aligned} &\max y_1 + \cdots + y_N, \\ &h(\boldsymbol{x}^k) \leqslant t_0 + \boldsymbol{t}\boldsymbol{x}^k \leqslant h(\boldsymbol{x}^k) y_k + M(1 - y_k), \quad k = 1, \ldots, N, \\ &(t_0, \boldsymbol{t}) \in \mathbf{R^{p+1}}, y \in \mathbf{B^N}, \end{aligned} \tag{12}$$

where M is a sufficiently large constant. In view of Corollary 4.3 and of Lemma 2.1 in Papadimitriou and Steiglitz [22], one can choose $M = (p+1)(p+1)!\binom{p}{2}$, since $|h(\boldsymbol{x})|$ is bounded from above by $\binom{p}{2}$. The binary variable y_k is equal to 1 iff $\boldsymbol{x}^k$ belongs to the contact of h and $t_0 + \boldsymbol{tx}$.

Notice that, since p is a small integer (typically, $p \leqslant 4$), the size of the mixed-integer program (15) is small: for example, when $p = 4$ there are 32 constraints, 16 binary variables and 5 continuous ones. Notice also that in practice we need to solve (15) only once and for all for each template. A full catalogue of TLMs for all the 60 templates with 3 or 4 vertices is given in Table 1,

where the following symbols are used:

$\mathbf{K_n}$	complete graph with n vertices,
$\mathbf{K_{q,r}}$	complete bipartite graph with q and r vertices,
$\mathbf{P_n}$	path with n vertices,
$\mathbf{D_n}$	$\mathbf{K_n} \setminus e$, where e is an arbitrary edge, and
$\mathbf{F_4}$	$\mathbf{K_{1,3}} \cup e$, where e is any non-existing edge.

In the simplest case, $p = 2$, the TLMs of the block $x_i x_j$ are x_i and x_j, those of the block -$x_i x_j$ are the constant 0 and $x_i + x_j - 1$. In both cases the contact consists of 3 points out of 4, and the TLMs are roofs with parameters $\lambda_{ij} = 0$ or 1.

It turns out that, at least for $p \leqslant 4$, TLMs of blocks are related to roofs in a very simple way. We need two preliminary definitions. Given an LM $g(\boldsymbol{x}) = c_0 + \boldsymbol{cx}$ of the (template) block h, let $\Delta = \min\{g(\boldsymbol{x}) - h(\boldsymbol{x}) : \boldsymbol{x} \in \mathbf{B^p}\}$. The *dried LM* $\tilde{g}(\boldsymbol{x}) = (c_0 - \Delta) + \boldsymbol{cx}$ is again an LM of h. If $c_0 - \lfloor c_o \rfloor \leqslant \Delta$, then the *truncated LM* $\hat{g}(\boldsymbol{x}) = \lfloor c_0 \rfloor + \boldsymbol{cx}$ is also an LM of h.

As a matter of fact, it turns out that all TLMs of template block functions with $p \leqslant 4$ variables are dried roofs. Actually, in 33 cases out of 60 they happen to be roofs; in 24 cases out of 60 they are truncated roofs; in the remaining 3 cases they are dried roofs with $\Delta = 1$. However, for $p = 5$ there are blocks whose TLMs are not dried roofs. Note: the above roofs do not have to be necessarily best ones.

A TLM of a block B is *basic* if it cannot be expressed as a conic combination (i.e., a linear combination with non-negative coefficients) of TLMs of sub-blocks of B. As indicated in Table 1, the only basic TLMs with at most 4 variables are:

- all the roofs,
- all the TLMs of K_3 and $K_{2,2}$ blocks that are truncated roofs,
- the TLM of the K_4 block: $-(x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4)$.

Finally, a closest LM (CLM) $u_0 + \boldsymbol{ux}$ of a template h can be obtained through the solution of the following LP:

$$\begin{aligned} &\min s_1 + \cdots + s_N, \\ &u_0 + \boldsymbol{u}\boldsymbol{x}^k - s_k = h(\boldsymbol{x}^k), \quad k = 1, \ldots, N, \\ &s_k \geqslant 0, \quad k = 1, \ldots, N, \\ &(u_0, \boldsymbol{u}) \in \mathbf{R^{p+1}}, \ \boldsymbol{s} \in \mathbf{R^N}. \end{aligned} \tag{13}$$

Again, since p is typically a small integer, in practice the size of this LP is small.

5. Finding optimal weights

Given an exhaustive family $\boldsymbol{F}$ for f, let us denote by $\boldsymbol{w}$ a non-negative $|\boldsymbol{F}|$-vector whose components w_B satisfy Eq. (10), and let W, denote the set of all such vectors $\boldsymbol{w}$. For each $B \in \boldsymbol{F}$ let $t_0\ (B) + \boldsymbol{t}(B)\boldsymbol{x}_B$ be a “good” LM (e.g., a TLM

Table 1

Templates with 3 variables

P$_3$ blocks		Tightest LMs						
x_1x_2	x_2x_3	const	x_1	x_2	x_3	C	E	
−1	−1	2	−1	−2	−1	5	4	Roof
−1	1	1	−1	−1	1	5	4	Roof
1	1	0	0	2	0	5	4	Roof

K$_3$ blocks			Tightest LMs						
x_1x_2	x_1x_3	x_2x_3	const	x_1	x_2	x_3	C	E	
−1	−1	−1	1	−1	−1	−1	6	2	Trunc. roof[a]
−1	1	1	0	0	0	1	6	2	Trunc. roof[a]
−1	−1	1	2	−2	−1	0	4	6	Roof
1	1	1	0	0	1	2	4	6	Roof

Templates with 4 variables

P$_4$ blocks			Tightest LMs							
x_1x_2	x_2x_3	x_3x_4	const	x_1	x_2	x_3	x_4	C	E	
−1	−1	−1	3	−1	−2	−2	−1	8	12	Roof
−1	−1	1	2	−1	−2	−1	1	8	12	Roof
−1	1	−1	1	−1	−1	1	0	8	12	Roof
−1	1	1	1	−1	−1	2	0	8	12	Roof
1	−1	1	1	1	−1	−1	1	8	12	Roof
1	1	1	0	1	0	2	0	8	12	Roof

K$_{1,3}$ blocks			Tightest LMs							
x_1x_2	x_1x_3	x_1x_4	const	x_1	x_2	x_3	x_4	C	E	
−1	−1	−1	0	0	0	0	0	9	12	Roof
−1	−1	1	0	1	0	0	0	9	12	Roof
−1	1	1	0	2	0	0	0	9	12	Roof
1	1	1	0	3	0	0	0	9	12	Roof

F$_4$ blocks				Tightest LMs							
x_1x_2	x_1x_3	x_1x_4	x_2x_3	const	x_1	x_2	x_3	x_4	C	E	
−1	−1	−1	−1	1	−1	−1	−1	0	9	8	Trunc. roof
−1	−1	1	−1	1	−1	−1	−1	1	9	8	Trunc. roof
−1	1	−1	1	0	0	0	1	0	9	8	Trunc. roof
−1	1	1	1	0	0	0	1	1	9	8	Trunc. roof
1	1	−1	−1	1	0	0	0	−1	9	8	Trunc. roof
1	1	1	−1	0	1	0	0	1	9	8	Trunc. roof
−1	−1	−1	1	0	0	0	1	0	7	16	Roof
−1	−1	1	1	0	1	1	0	0	7	16	Roof
−1	1	−1	−1	0	1	0	0	0	7	16	Roof
−1	1	1	−1	1	2	−1	−1	0	7	16	Roof
1	1	−1	1	0	2	1	0	0	7	16	Roof
1	1	1	1	0	3	0	1	0	7	16	Roof

Table 1. (continued)

$K_{2,2}$ blocks				Tightest LMs							
x_1x_2	x_1x_4	x_2x_3	x_3x_4	const	x_1	x_2	x_3	x_4	C	E	
−1	1	−1	−1	1	0	−1	−1	0	8	8	Trunc. roof[a]
−1	1	1	1	0	0	0	1	1	8	8	Trunc. roof[a]
−1	−1	−1	−1	4	−2	−2	−2	−2	7	16	Roof
−1	1	−1	1	2	−1	−2	−1	2	7	16	Roof
−1	1	1	−1	1	−1	−1	1	1	7	16	Roof
1	1	1	1	0	0	2	0	2	7	16	Roof

D_4 blocks					Tightest LMs							
x_1x_2	x_1x_3	x_1x_4	x_2x_3	x_3x_4	const	x_1	x_2	x_3	x_4	C	E	
−1	−1	1	−1	−1	2	0	−1	−2	−1	8	12	Trunc. roof
−1	1	1	−1	−1	2	0	−2	−1	0	8	12	Trunc. roof
−1	−1	1	1	1	1	−1	−1	1	1	8	12	Trunc. roof
−1	1	1	1	1	0	0	0	1	2	8	12	Trunc. roof
−1	−1	−1	−1	−1	3	−2	−2	−2	−1	7	12	Trunc. roof
−1	−1	1	−1	1	2	−1	−2	−1	1	7	12	Trunc. roof
−1	1	1	1	−1	0	0	0	1	1	7	12	Trunc. roof
−1	1	−1	1	1	1	−1	−1	2	0	7	12	Trunc. roof
1	−1	1	1	1	0	0	1	0	2	7	12	Trunc. roof
−1	1	−1	−1	−1	4	−2	−2	−1	−2	6	20	Roof
−1	1	1	−1	1	2	−1	−2	0	2	6	20	Roof
−1	−1	1	1	−1	2	0	1	−2	−1	6	20	Roof
−1	−1	−1	1	1	3	−3	−1	1	−1	6	20	Roof
1	1	1	1	1	0	2	0	3	0	6	20	Roof

K_4 blocks						Tightest LMs							
x_1x_2	x_1x_3	x_1x_4	x_2x_3	x_2x_4	x_3x_4	const	x_1	x_2	x_3	x_4	C	E	
−1	−1	−1	−1	−1	−1	1	−1	−1	−1	−1	10	8	Dried roof
−1	−1	1	−1	1	1	0	0	0	0	1	10	8	Dried roof
−1	1	1	1	1	−1	0	1	1	0	0	10	8	Dried roof
−1	−1	−1	−1	1	−1	3	−2	−2	−2	0	7	16	Trunc. roof
−1	−1	1	−1	1	−1	3	−1	−1	−3	0	7	16	Trunc. roof
−1	1	1	−1	1	−1	2	1	−1	−2	0	7	16	Trunc. roof
−1	1	1	−1	1	1	0	2	0	0	1	7	16	Trunc. roof
−1	1	1	1	1	1	0	0	0	1	3	7	16	Trunc. roof
−1	1	−1	−1	1	−1	4	−2	−2	−1	−1	5	24	Roof
−1	1	1	−1	−1	1	3	−1	−3	0	1	5	24	Roof
1	1	1	1	1	1	0	1	0	2	3	5	24	Roof

[a]The TLM is not a conic combination of TLMs of sub-blocks. C is the number of contact points, **E** is the excess of the TLM.

or a CLM) of the block function $f_B(\boldsymbol{x}_B)$. Then, for any $\boldsymbol{w}\in W$, $g(\boldsymbol{w};\boldsymbol{x})=\sum_{B\in\boldsymbol{F}} w_B(t_0(B)+\boldsymbol{t}(B)\boldsymbol{x}_B)$ is an LM of f.

Our aim is to choose $\boldsymbol{w}^*\in W$ such that $g(\boldsymbol{w}^*;\ \boldsymbol{x})$ is best among all LMs $g(\boldsymbol{w};\ \boldsymbol{x})$, $\boldsymbol{w}\in W$, i.e., the optimal value $L(\boldsymbol{w})$ of the corresponding linear relaxation (2) is as small as possible. Let $\boldsymbol{F}_i=\{B\in\boldsymbol{F}: i\in V(B)\}$, $i=1,\dots,n$, and as customary let $z^+=\max\{z,0\}$, $z\in\mathbf{R}$. Then we have

$$L(\boldsymbol{w})=\sum_{B\in\boldsymbol{F}} w_Bt_0(B)+\max_{x\in\mathbf{B^n}}\sum_{B\in\boldsymbol{F}} w_B\sum_{i\in V(B)} t_i(B)x_i=\sum_{B\in\boldsymbol{F}} w_Bt_0(B)+\sum_{i=1}^{n}\left(\sum_{B\in\boldsymbol{F}_i} w_Bt_i(B)\right)^+. \tag{14}$$

Therefore, we can compute an optimal weight vector $\boldsymbol{w}^*$ by solving the LP

$$\min \sum_{B\in \boldsymbol{F}} w_B t_0(B) + \sum_{i=1}^{n} u_i,$$

$$\text{(L)} \quad u_i \geqslant \sum_{B\in \boldsymbol{F}_i} w_B t_i(B), \quad i=1,\ldots,n, \tag{15}$$

$$\sum_{B\in \boldsymbol{F}_{ij}} w_B = |q_{ij}|, \quad ij \in E, \tag{16}$$

$$\boldsymbol{w} \geqslant \boldsymbol{0}, \quad \boldsymbol{u} \geqslant \boldsymbol{0}.$$

First of all, let us consider two special cases of (L).

1. Let us choose $\boldsymbol{F}$ as the family of all edges with their appropriate sign. For each block $x_i x_j$ we take as LM either the TLM x_i or the TLM x_j; for each block -$x_i x_j$ we take as LM the TLM given by the constant 0. Then the optimal value of (L) is $\sum_{ij\in E^+} q_{ij}$, a rather weak bound.
2. Let us choose again $\boldsymbol{F}$ as above. Let

$$g^*(\boldsymbol{x}) = \sum_{ij\in E^+} (\lambda^*_{ij} x_i + (1-\lambda^*_{ij})x_j) + \sum_{ij\in E^-} \lambda^*_{ij}(1 - x_i - x_j)$$

be a best roof, let $\bar{\lambda}_{ij} = \lambda^*_{ij}/|q_{ij}|$, $ij \in E$, and let us take

$\bar{\lambda}_{ij} x_i + (1-\bar{\lambda}_{ij})x_j$ as the LM of the block $x_i x_j$,

$\bar{\lambda}_{ij}(1 - x_i - x_j)$ as the LM of the block $-x_i x_j$.

In this case the optimal value of (L) coincides with the roof-dual optimum.

Coming back to the general version of (L), the total number of constraints is $n+m$, a manageable number, while the number of variables is $n + |\boldsymbol{F}|$, which may be quite large. Therefore, it is quite natural to think of a column generation approach to its solution.

We shall assume that the given family $\boldsymbol{F}$ always contains all the blocks consisting of single signed edges, and that the LMs corresponding to these blocks are defined as in 2. These choices ensure that:

(i) $\boldsymbol{F}$ is exhaustive;
(ii) The optimal value of (L) is at least as good as the roof-dual optimum.

Given a feasible basis for (L), let μ_i and π_{ij} be the associated multipliers of constraints (15) and (16), respectively. Then, the reduced cost of the variable $w_B, B \in \boldsymbol{F}$, can be written as

$$\bar{c}(B) = t_0(B) + \sum_{i\in V(B)} t_i(B)\mu_i - \sum_{ij\in E(B)} \pi_{ij}. \tag{17}$$

Let B^{t} be the template of B, and let us denote by $B(i)$ the image in $V(B)$ of $i \in V(B^{\mathrm{t}})$ under the sign-isomorphism between B^{t} and B. Then we can re-write (17) as

$$\bar{c}(B) = t_0(B^{\mathrm{t}}) + \sum_{i=1}^{|V(B)|} t_i(B^{\mathrm{t}})\mu_{B(i)} - \sum_{ij\in E(B)} \pi_{ij}. \tag{18}$$

In view of (18), only the LMs of block-templates need to be stored.

A formal description of the algorithm follows.

Column generation algorithm

Step 1: Let $k=0$;
Let $\boldsymbol{F}_0$ be the family of blocks defined in 1;
For each block B in $\boldsymbol{F}_0$, get an LM $t_0(B) + \boldsymbol{t}(B)\boldsymbol{x}_B$ from $g^*(\boldsymbol{x})$ as explained in 2;
Solve (L) with $\boldsymbol{F}$ replaced by $\boldsymbol{F}_0$;
{This is equivalent to solving the roof-dual of (1)}

Step 2: For each $B \in \boldsymbol{F} \setminus \boldsymbol{F}_k$, compute the reduced cost of the variable w_B according to (18):

$$\bar{c}(B) = t_0(B^{\mathrm{t}}) + \sum_{i=1}^{|V(B)|} t_i(B^{\mathrm{t}})\mu^k_{B(i)} - \sum_{ij \in E(B)} \pi^k_{ij};$$

compute $\bar{c}(B^*) = \min\{\bar{c}(B) : B \in \boldsymbol{F} \setminus \boldsymbol{F}_k\}$;

Step 3: If $\bar{c}(B^*) \geqslant 0$, then **stop**:

$$w^*_B = \begin{cases} w^k_B, & B \in \boldsymbol{F}_k, \\ 0, & \text{else} \end{cases}$$

Step 4: is an optimal weight vector, otherwise go to
Let $\boldsymbol{F}'$ be a set of blocks such that $\bar{c}(B) < 0$ for each $B \in \boldsymbol{F}'$. Set $\boldsymbol{F}_{k+1} = \boldsymbol{F}_k \cup \{\boldsymbol{F}'\}$; increase k by 1;

Step 5: At the kth iteration, let $\boldsymbol{F}_k \subseteq \boldsymbol{F}$ be the current family of blocks;
Solve (L) with $\boldsymbol{F}$ replaced by $\boldsymbol{F}_k$;
Let $(\boldsymbol{u}^k, \boldsymbol{w}^k)$ be an optimal solution; let μ^k_i and π^k_{ij} be the optimal multipliers associated with constraints (17) and (18), respectively.
Go to Step 2.

We end up this section with a brief discussion of the worst-case complexity of our procedure. As above, n is the number of variables in the quadratic 0–1 maximization problem, and p is the maximum number of variables in the blocks of the chosen family $\boldsymbol{F}$. For our procedure to be viable, p must be a small number (typically, $p \leqslant 4$). In our complexity analysis, we shall assume that p is a *fixed* positive integer.

It is fairly obvious that the number of template functions of order p depends only on p, and since their coefficients are 0, 1, or −1 such number is no more than $P = 3^{\binom{p}{2}}$. For each such template function, a TLM can be obtained through the solution of (12), an integer program involving 2^p binary variables, $p+1$ continuous variables, and 2^{p+1} constraints. Such a program can obviously be solved in $poly(2^p)$ time (see also [20]); thus the total time for precomputing templates and their TLMs depends only on p, and hence can be regarded as a constant for any fixed value of p (although this "constant" grows rapidly with p.)

We emphasize the fact that a catalogue of all templates of order at most p and their TLMs (only templates with basic TLMs need to be considered) is computed off-line once and for all, since it does not depend on the input function f, and is assumed to be available at the beginning of the execution of our procedure on any particular input f.

In the linear program (L) the number of variables (blocks) is no more than

$$\binom{n}{p} P = \mathrm{O}(n^p),$$

(where the constant factor on the r.h.s. is a function of p, but not of n), the number of constraints is $\mathrm{O}(n^2)$, and the size of the coefficients $t_i(B)$ depends only on p. Hence the input size of (L) is polynomial in n and $\log q_{\max}$, where $q_{\max} = \max_{ij \in E} |q_{ij}|$. Therefore, (L) can be solved in time polynomial in the size of f.

In conclusion, the overall procedure can be implemented so as to run in polynomial time, for any fixed p.

6. Computational experiments

The sharpness of the upper bounds given by the solution of problem (L) and the performance of the column generation procedure have been evaluated on a set of 480 test problems with a number of variables between 50 and 200. Each experiment is defined as follows:

$$\text{Test} = (n, density,\ npositive),$$

where $n \in \{50, 100, 150, 200\}$, *density* $\in \{5, 10, 25, 50\}$ and *npositive* $\in \{20, 50, 80\}$. Here, n is the number of variables, *density* is the density of G, that is, $density = p$ means that in the test problems the number of edges of G is p% of $\binom{n}{2}$ and *npositive* is the density of positive terms, that is, $npositive = s$ means that the number of positive terms q_{ij} is s% of $|E|$. For each fixed Test, 10 test problems were randomly generated in which the graph G is connected and each pair ij, for $i, j \in V$, has the same probability to belong to E. The absolute values of the terms $|qij|$ are integers uniformly distributed in the interval [1, 10], for each $ij \in E$.

We do not present the results with *npositive* = 80, since the optimal value of (L) was almost always equal to the roof-dual optimum (and, whenever the optimal value was known, very close to it).

In our experiments, we never took CLMs into consideration since, as mentioned in Appendix A, every TLM is also a CLM at least for all templates of order $p \leqslant 4$.

We have implemented the column generation procedure using the CPLEX 7.0 callable library and the Delphi environment; the tests have been performed on a Pentium III, 1.1 GHz with 512 MB of RAM and Windows 98 SE operating system.

The optimal value of (L) depends on the family of blocks $\boldsymbol{F}$. Initially, the family $\boldsymbol{F}$ was formed by all the blocks of order at most 4 having a basic TLM. When the density is large the number of the blocks of the above type is too large (even though it is polynomial). In order to obtain a family of blocks $\boldsymbol{F}$ containing a reasonable number of blocks, we defined the following procedure.

Procedure A.

(0) let $d_{ij} =$ (degree of vertex i)*(degree of vertex j) for each $ij \in E$ and let α be a positive real number;

(1) let $\boldsymbol{F} = \boldsymbol{F}_0$, where $\boldsymbol{F}_0$ is the family of signed edges of G (see 1., Section 5);
(2) repeat
  (2.1) generate a set V' of 3 or 4 vertices of V;
  (2.2) if (V' induces in G a signed subgraph $B \notin \boldsymbol{F}$ whose TLM is basic) and (each edge ij of B is contained in less than αd_{ij} blocks of $\boldsymbol{F}$) and ($|\boldsymbol{F}| < 10\,000$) then

    add B to $\boldsymbol{F}$

  until there is no block satisfying condition (2.2).

Here many aspects of the selection procedure are not fully specified; for instance, there seems to be some relation between the order in which different block types are generated and the quality of the results: we obtain better results if, when *npositive* = 20, we generate first the K_4's, then the K_3's and finally the C_4's; and when *npositive* = 50, we generate first the K_3's, then the K_4's and finally the C_4's.

For a given test problem, let *RD* and *Opt* be the roof-dual optimum and the optimum of (1), respectively, and let L^* be the upper bound on *Opt* obtained through the solution of (L); also, let

$$E_{L*} = \frac{L^* - Opt}{Opt},$$

$$E_{RD} = \frac{RD - Opt}{Opt},$$

$$E_{RD,L*} = \frac{RD - L*}{RD}.$$

For each experiment we computed the average values, over the 10 test problems, of the following performance indicators:

- $E_{L^*}, E_{RD}, E_{RD,L^*}$;
- the cardinality of $\boldsymbol{F} \setminus \boldsymbol{F_0}$;
- the number of columns at the end of the column generation procedure;
- the running times.

In a first set of experiments we solved problem (L) directly, without column generation. In order to obtain reasonable running times also for the problems with 200 variables and $density = 50$, we set $\alpha = 0.0001$. With this value of α each edge can be contained in at most one block of $\boldsymbol{F} \setminus \boldsymbol{F_0}$. In Table 2 the experimental results are shown. The improvement with respect to the roof-dual value increases as *density* increases and, for fixed *density* and *npositive*, as n increases. In fact, while for $density = 5$ or 10, the improvement with respect to the roof-dual value is 0 or close to 0, for problems with $density = 50$ and $n = 200$ it becomes significant, mainly if *npositive* = 50. On the other hand, when n and *density* are very large, the running times are relatively large. For the problems with 50 variables we were able to find the optimal value of (1) by means of a branch and bound procedure reported in [4]. For the problems with lower density both the roof-dual optimum and the bound obtained by (L) are equal or very close to the optimal value, whereas for the problems with $n = 50$, $density = 50$ and *npositive* = 20 the bound is almost twice the optimal value. This result is due to the choice of the value α.

Table 2

n	density	npositive	Running times (s)	$\|F \setminus F_0\|$	E_{L^*}	E_{RD}	E_{RD,L^*}
50	5	20	0.02	2.1	0.0000	0.0000	0.0000
50	10	20	0.049	14.8	0.0221	0.0427	0.0182
50	25	20	0.266	71	0.3343	0.8672	0.2864
50	50	20	1.051	172.3	0.9395	2.3984	0.4277
100	5	20	0.103	19.5	—	—	0.0136
100	10	20	0.407	85.3	—	—	0.1906
100	25	20	3.53	274.3	—	—	0.4085
100	50	20	13.666	517.2	—	—	0.5050
150	5	20	0.367	66.5	—	—	0.0998
150	10	20	3.422	225.4	—	—	0.2917
150	25	20	22.972	625.3	—	—	0.4656
150	50	20	69.253	1180.6	—	—	0.5403
200	5	20	1.419	142.9	—	—	0.1672
200	10	20	12.325	434.6	—	—	0.3506
200	25	20	84.557	1116.3	—	—	0.4976
200	50	20	374.52	2915	—	—	0.5906
50	5	50	0.052	1.3	0.0000	0.0000	0.0000
50	10	50	0.078	13.5	0.0000	0.0000	0.0000
50	25	50	0.302	68.4	0.0480	0.2785	0.1788
50	50	50	0.894	170.1	0.3355	0.8306	0.2700
100	5	50	0.137	21.1	—	—	0.0018
100	10	50	0.957	85.9	—	—	0.1250
100	25	50	3.217	326.4	—	—	0.2566
100	50	50	10.426	737.3	—	—	0.2980
150	5	50	0.501	65.5	—	—	0.0357
150	10	50	2.844	229.2	—	—	0.1831
150	25	50	25.396	778.6	—	—	0.2761
150	50	50	97.443	1711.2	—	—	0.3104
200	5	50	2.008	141.7	—	—	0.1009
200	10	50	7.911	443.4	—	—	0.2098
200	25	50	80.745	1431.9	—	—	0.2847
200	50	50	403.373	4401.5	—	—	0.3455

Table 3

n	density	npositive	Running times (s)	$\|F \setminus F_0\|$	E_{L^*}	E_{RD}	E_{RD,L^*}
50	5	20	0	2.2	0.0000	0.0000	0.0000
50	10	20	0.093	46.6	0.0014	0.0427	0.0363
50	25	20	1.984	1340.1	0.0207	0.8672	0.4511
50	50	20	23.652	10000	0.1165	2.3984	0.6674
50	5	50	0.101	1.3	0.0000	0.0000	0.0000
50	10	50	0.465	44.2	0.0000	0.0000	0.0000
50	25	50	9.486	1475.2	0.0000	0.2785	0.2149
50	50	50	80.857	10000	0.0002	0.8306	0.4510

In order to find the sharpness of the bound with a very large number of blocks, we performed a second set of experiments only for the problems with $n=50$. In these experiments we generated the blocks randomly as in procedure A, but we did not consider the bound on the number of blocks containing each edge. The results are shown in Table 3. If *npositive* = 50, the bound given by the optimal value of (L) is always equal or almost equal to *Opt*, and, if *npositive* = 20, the relative error is always less than 0.12. On the other hand, the running times increased up to 100 times.

Table 4

npositive = 50			1 iteration			2 iterations			4 iterations		
n	density	$\|F \setminus F_0\|$	R.T.	NCol	E_{RD,L^*}	R.T.	NCol	E_{RD,L^*}	R.T.	NCol	E_{RD,L^*}
200	5	143.4	1.26	137.6	0.1045	1.28	137.6	0.1045	1.33	137.6	0.1045
200	10	443.6	6.00	443.5	0.2098	6.06	443.6	0.2099	6.13	443.6	0.2099
200	25	4483.2	50.14	1166.4	0.2369	95.35	1862.2	0.3493	205.34	2738	0.4315
200	50	10000	10.04	1	0.0001	81.44	2068.4	0.1126	201.07	2388	0.2228

npositive = 50			8 iterations			12 iterations			16 iterations		
n	density	$\|F \setminus F_0\|$	R.T.	NCol	E_{RD,L^*}	R.T.	NCol	E_{RD,L^*}	R.T.	NCol	E_{RD,L^*}
200	5	143.4	1.41	137.6	0.1045	1.50	137.6	0.1045	1.59	137.6	0.10451
200	10	443.6	6.33	443.6	0.2099	6.52	443.6	0.2099	6.69	443.6	0.20985
200	25	4483.2	349.9	3096	0.4511	427.74	3202.2	0.454	479.47	3260.6	0.45499
200	50	10000	431.91	4323.2	0.3553	643.57	5188	0.4009	844.45	5693.1	0.41913

R.T. is the running times in seconds. NCol. is the number of columns corresponding to a block of $|F \setminus F_0|$ in the LP.

In a third set of experiments we tested a column generation procedure in which $\boldsymbol{F}' = \{B^*\}$, that is, only one block with minimum reduced cost is added to $\boldsymbol{F}_k$ at each iteration. The convergence turned out to be very slow. To improve on the convergence, we took $\boldsymbol{F}'$ as a maximal set of blocks $B \in \boldsymbol{F} \setminus \boldsymbol{F}_k$ such that each edge of G belongs to at most one block of $\boldsymbol{F}'$, and

$$\bar{c}(B) \leqslant \delta\bar{c}(B^*) + (1 - \delta)\bar{c}(B^{\max}),$$

where $B^{\max}$ is a block of $\boldsymbol{F} \setminus \boldsymbol{F}_k$ with maximum negative reduced cost and $0 < \delta < 1$. In the experiments we let $\delta = 0.5$. By setting $\alpha = 0.0001$ as in the previous experiments, all the blocks were added to $\boldsymbol{F}$ in the first two iterations. Then we set $\alpha = 0.001$ and, in order to limit the running times, the column generation algorithm was truncated after the 16th iteration and a time-bound of 1 min was imposed on the solution time of each iteration. In Table 4, the results of these experiments for the problems with $n = 200$ and $density = 50$ are shown. The bounds obtained at the 16th iteration are better than those shown in Table 2, but the running times for the larger problems are higher (for the problems with $n = 200$ and $density = 50$ the running times at the 12th iteration are greater than 10 min). Moreover, the bounds obtained at the 12th and at the 16th iteration are always very close (the difference between the indicators E_{RD,L^*} is less than 0.02). However, when truncated column generation is used with $\alpha = 0.001$, the bounds one obtains by solving (L) to optimality with $\alpha = 0.0001$ are reached, in the problems with lower density, after one or two iterations; and, in the problems with higher density, before the 8th iteration, with running times comparable to those of Table 2.

Tables 5 and 6 contain the results of the truncated column generation algorithm on the test problems with $n = 50$. As before, in these experiments we did not pre-set any bound on the number of blocks containing each edge, but we set instead a bound of 10 000 on the total number of blocks. By comparing these tables with Table 3, we observe that after 16 iterations the upper bounds are very close to the bounds obtained by the solution of problem (L) at optimality and the average running times, when *density* is 50%, decrease by 68% for *npositive* = 20, and by 70% for *npositive* = 50. Even the bounds obtained after 12 or 8 iterations are very good.

A further set of experiments has been performed on a set of DIMACS benchmarks for the maximum clique problem. Let $G = (V, E)$ be a graph, and let $G' = (V, E')$ be its complement. We considered the following formulation of the maximum clique problem in G as an unconstrained quadratic 0–1 maximization model:

$$Opt = \max_{x \in \mathbf{B^n}} \left(\sum_{i \in V} x_i - \sum_{ij \notin E'} x_i x_j \right), \tag{19}$$

where Opt is the maximum cardinality of a clique of G. For each of the selected instances, we solved directly problem (L) for formulation (19). Table 7 contains the results of these experiments. In the following, RD and L^* are the optimal values of the roof-dual and of (L), respectively, for formulation (19). Obviously, when these values are fractional one is allowed to round them down, but we chose not to, in order to report on the actual values of these two optima. We also

Table 5

npositive = 20			1 iteration			2 iterations			4 iterations		
n	density	$\|F \setminus F_0\|$	R.T.	NCol	E_{RD,L^*}	R.T.	NCol	E_{RD,L^*}	R.T.	NCol	E_{RD,L^*}
50	5	2.2	0	0	0	0	0	0	0.006	0	0
50	10	46.6	0.005	2.7	0.0281	0.016	3.4	0.035	0.032	3.8	0.0363
50	25	1340.1	0.039	4.9	0.0409	0.089	18	0.1127	0.231	62.4	0.3038
50	50	10000	0.319	43	0.175	0.671	72.1	0.2961	1.341	106.3	0.4088

npositive = 20			8 iterations			12 iterations			16 iterations		
n	density	$\|F \setminus F_0\|$	R.T.	NCol	E_{RD,L^*}	R.T.	NCol	E_{RD,L^*}	R.T.	NCol	E_{RD,L^*}
50	5	2.2	0.018	0	0	0.023	0	0	0.054	0	0
50	10	46.6	0.054	3.8	0.0363	0.06	3.8	0.0363	0.077	3.8	0.03629
50	25	1340.1	0.527	107.3	0.4279	0.708	125.7	0.4461	0.835	134.3	0.44948
50	50	10000	3.048	179.8	0.5317	5.383	263	0.6207	7.596	329.1	0.65852

R.T. is the running times in seconds. NCol. is the number of columns corresponding to a block of $|F \setminus F_0|$ in the LP.

Table 6

npositive = 50			1 iteration			2 iterations			4 iterations		
n	density	$\|F \setminus F_0\|$	R.T.	NCol	E_{RD,L^*}	R.T.	NCol	E_{RD,L^*}	R.T.	NCol	E_{RD,L^*}
50	5	1.3	0	0	0	0	0	0	0	0	0
50	10	44.2	0	0	0	0.006	0	0	0.006	0	0
50	25	1475.5	0.142	42.1	0.1129	0.296	83.2	0.187	0.417	107.7	0.2134
50	50	10000	0.264	13.4	0.0258	0.533	22.1	0.042	1.021	30.8	0.0539
npositive = 50			8 iterations			12 iterations			16 iterations		
n	density	$\|F \setminus F_0\|$	R.T.	NCol	E_{RD,L^*}	R.T.	NCol	E_{RD,L^*}	R.T.	NCol	E_{RD,L^*}
50	5	1.3	0.04	0	0	0.051	0	0	0.057	0	0
50	10	44.2	0.006	0	0	0.034	0	0	0.061	0	0
50	25	1475.5	0.511	112.1	0.2149	0.573	112.1	0.2149	0.643	112.1	0.21493
50	50	10000	3.652	267.1	0.3513	7.404	396.9	0.4299	10.292	458.8	0.44493

R.T. is the running times in seconds. NCol is the number of columns corresponding to a block of $|F \setminus F_0|$ in the LP.

computed the following upper bounds on the cardinality of the maximum clique:

$$Opt \leqslant [3 + \sqrt{(9 - 8(n - m))}]/2$$

(only for connected graphs) where m is the number of edges of G [2];

$$Opt \leqslant \rho(A_G) + 1,$$

where $\rho(A_G)$ is the maximum eigenvalue of the adjacency matrix A_G of G [25];

$$Opt \leqslant N_{-1} + 1,$$

where N_{-1} is the number of eigenvalues of A_G that do not exceed -1 [2];

$$Opt \leqslant (n + N_0')/2,$$

where N_0' is the number of zero eigenvalues of the adjacency matrix $A_{G'}$ of G' [12].

In the following U^* will denote the minimum of these four bounds and $E_{U^*} = (U^* - Opt)/Opt$.

Table 7

Problem name	n	m'	Opt	U^*	E_{U^*}	($\alpha = 10^{-7}$)					Running times from
						R.T.	*RD*	L^*	E_{RD}	E_{L^*}	DIMACS (s)
san200_0.9_1	200	17910	70	98	0.40	2.36	100	72.4	0.43	0.03	75.69
san200_0.9_2	200	17910	60	98	0.63	3.46	100	67.6	0.67	0.13	57
san200_0.9_3	200	17910	44	96	1.18	4.28	100	61.5	1.27	0.40	48.35
c-fat200-1	200	1534	12	17	0.42	248.42	100	50	7.33	3.17	0.03
c-fat200-2	200	3235	24	33	0.38	118.03	100	50	3.17	1.08	0.04
c-fat200-5	200	8473	58	85	0.47	90.57	100	66.7	0.72	0.15	20.35
MANN_a9	45	918	16	20	0.25	0.06	22.5	18	0.41	0.13	3.96
hamming 6-2	64	1824	32	42	0.31	0.05	32	32	0.00	0.00	0.30
hamming 6-4	64	704	4	22	4.50	0.6	32	16	7.00	3.00	0.04
johnson 8-2-4	28	210	4	8	1.00	0.06	14	7	2.50	0.75	0.03
johnson 8-4-4	70	1855	14	28	1.00	0.33	35	18	1.50	0.29	0.13
johnson 16-2-4	120	5460	8	16	1.00	2.09	60	30	6.50	2.75	18.60

m' is the number of edges in the complementary graph G'.

Table 8

Problem name	n	m'	Opt	Formulation (19) ($\alpha = 10^{-7}$)					Formulation (20) ($\alpha = 10^{-5}$)				
				R.T.	RD	L^*	E_{RD}	E_{L^*}	R.T.	RD	L^*	E_{RD}	E_{L^*}
c-fat 200-1	200	1534	12	248.42	100	50	7.33	3.17	355.36	39.67	27.30	2.31	1.28
c-fat 200-2	200	3235	24	118.03	100	50	3.17	1.08	610.99	57.38	40.08	1.39	0.67
c-fat 200-5	200	8473	58	90.57	100	66.7	0.72	0.15	667.79	92.55	68.30	0.60	0.18
hamming6-4	64	704	4	0.6	32	16	7.00	3.00	5.21	27.04	15.68	5.76	2.92

m' is the number of edges in the complementary graph G'.

The quality of the observed bounds L^* strongly depends on the ratio k/n. As a matter of fact, we can see in Table 7 that RD is always equal to $n/2$ and that L^* is always at least $n/4$. In particular, if the cardinality of the maximum clique is $n/4$ or less, then L^* is almost always equal to $n/4$. We will prove in Appendix B (Property B.1) that, when blocks with at most four variables are used, RD and L^* cannot be less than these values. In the cases where *Opt* is at least $n/4$, the observed relative error of the bound L^* is at most 0.15. Morevover the minimum of the other four bounds is better than L^* only in three cases in which $Opt < n/4$.

It is perhaps worth mentioning that sometimes one may get smaller values of both RD and L^* simply by changing the objective function in formulation (1) of the maximum clique problem. Along with (19), we have considered the formulation

$$\max_{x \in \mathbf{B}^n} \left(\sum_{ij \in E} x_i x_j - d \sum_{ij \notin E} x_i x_j \right), \tag{20}$$

where d is the maximum vertex-degree of G. The correctness of formulation (20) follows from Property B.2, proved in Appendix B. As shown there, the optimal value of (20) is equal to the number of edges in a clique of maximum size. Then, if s is a bound on the number of edges in a maximum clique, the positive solution y of the quadratic equation $\binom{y}{2} = s$ is a bound on the number of vertices in a maximum clique. In the Table 8, the roof-dual optimum RD and the optimal value L^* of (L) relative to two formulations (19) and (20) are compared for some of the DIMACS instances. Clearly, formulation (20) yields better bounds.

Unfortunately, this is not true for the other DIMACS benchmarks and for other graphs we have tried. Furthermore, the running times are much higher with respect to the other formulation: a simple explanation of this fact is that here *density* is always equal to 100%. So, formulation (20) has not been pursued any longer.

Table 9

n	density (%)	Opt	U^*	E_{U^*}	Formulation (19) ($\alpha = 10^{-9}$) R.T.	RD	L^*	E_{RD}	E_{L^*}
∗ 200	90	80	98.75	0.23	1.81	100	80	0.25	0.00
∗ 200	90	70	99.25	0.42	2.69	100	70	0.43	0.00
∗ 200	90	65	99.25	0.53	5.13	100	65.11	0.54	0.00
200	90	60	98.5	0.64	5.82	100	61.84	0.67	0.03
200	90	55	97.5	0.77	6.12	100	59.61	0.82	0.08
∗ 200	90	44	96.75	1.20	5.66	100	57.83	1.27	0.31
∗ 200	90	35	95.5	1.73	5.59	100	58.60	1.86	0.67
∗ 200	90	25	91.75	2.67	5.35	100	56.45	3.00	1.26
200	70	65	98.25	0.51	9.75	100	65	0.54	0.00
200	70	60	98.5	0.64	23.86	100	60	0.67	0.00
200	70	55	98.25	0.79	49.25	100	55	0.82	0.00
∗ 200	70	45	97.75	1.17	78.55	100	50	1.22	0.11
∗ 200	70	30	97.25	2.24	73.63	100	50	2.33	0.67
∗ 200	70	18	96	4.33	64.62	100	50	4.56	1.78
200	50	55	96.75	0.76	60.53	100	55	0.82	0.00
200	50	50	97.25	0.95	122.88	100	50	1.00	0.00
200	50	45	96.25	1.14	119.49	100	50	1.22	0.11
∗ 200	50	30	96.5	2.22	115.32	100	50	2.33	0.67
∗ 200	50	20	95.5	3.78	130.10	100	50	4.00	1.50
∗ 200	50	11	94.75	7.61	117.90	100	50	8.09	3.55

In order to perform other experiments on the maximum clique problem, we adapted a random graph generator proposed by Sanchis [23]. This generator produces a random graph G'' with a specified number of vertices and edges, and known minimum vertex cover size, so that the complement of G'' has known maximum clique size. In the graph G'', let n be the number of vertices, m'' the number of edges and k the required clique size; initially, the generator partitions the set of n vertices into k cliques; then chooses all but one of the vertices in each clique to be in the cover. The complement G' of G'' has maximum clique size equal to k. Then the generator adds edges at random, but in such a way that each added edge is adjacent to at least one vertex in the chosen cover. Sanchis points out that if the sizes of the k cliques are "as equal as possible", and the same is true for vertex degrees, the resulting graph instances turn out to be harder with respect to the computation of a maximum clique. In order to obtain graphs with this property, we start from k cliques of equal, or almost equal, sizes and iteratively add edges with probability inversely proportional to the product of the degrees of their endpoints. We tested the sharpness of the bounds RD and L^* on the Sanchis graphs generated as described above. Each test is defined by the number of vertices n, the density of G' and the size Opt of the maximum clique. For each test, 4 instances were randomly generated: the variance of the results over the 4 instances turned out to be very low. In Table 9 we show, for each test, the average values of the indicators. We show only the results obtained with formulation (19). In Table 9, the starred tests have been already considered in [24].

Notice that in all the above tests the parameters n, *density*, and Opt satisfy the inequality

$$density > (Opt/n)^2.$$

In the above-mentioned paper, Sanchis and Jagota essentially prove (Theorem 4.2) that the maximum clique problem remains NP-hard when restricted to those graph instances whose parameters n, *density*, and Opt satisfy the above inequality.

The results of Table 9 confirm the previous observations. In particular, if the maximum clique size is greater than $n/4$, L^* is almost always equal to Opt and is always less than the other four bounds described above.

Finally we considered the weighted max-2-sat problem, which can be easily transformed in a 0–1 quadratic maximization problem. In Table 10, we compare the optimal values of 16 problems presented in [3] with the upper bounds obtained by solving directly problem (L). In the experiments we fixed $\alpha = 1$. The upper bound is equal to the optimal value in 8 cases and, in the other cases, the relative error is at most 0.013. However, for the problems with lower density, the improvement with respect to the roof dual is small. Whereas in all the previous experiments the number of K_3 blocks generated by Procedure A was much greater than the number of $K_{2,2}$ blocks, in these latter experiments $K_{2,2}$ blocks form the largest group.

Table 10

n	No. of clauses	density (%)	Opt	R.T.	RD	L^*	E_{RD}	E_{L^*}
50	100	7.76	538	0.06	550.5	545	0.0232	0.0130
50	150	11.18	766	0.16	792	766	0.0339	0.0000
50	200	15.18	1034	0.39	1085	1034	0.0493	0.0000
50	250	18.94	1265	1.04	1348	1265	0.0656	0.0000
50	300	21.06	1502	1.37	1598	1502	0.0639	0.0000
50	350	24.57	1725	5.44	1896	1729	0.0991	0.0023
50	400	27.51	1993	5.27	2146	1993	0.0768	0.0000
50	450	30.53	2262	10.55	2451	2262	0.0836	0.0000
50	500	33.88	2502	35.21	2748	2502	0.0983	0.0000
100	200	3.94	1096	0.11	1097	1096	0.0009	0.0000
100	300	5.92	1567	0.44	1630	1581	0.0402	0.0089
100	400	7.90	2085	1.21	2202	2108	0.0561	0.0110
100	500	9.62	2579	3.68	2801	2606	0.0861	0.0105
100	600	11.47	3103	9.72	3342	3119	0.0770	0.0052
150	300	2.65	1610	0.22	1634	1620	0.0149	0.0062
150	450	3.92	2440	0.50	2505	2468	0.0266	0.0115

In conclusion, the present paper contributes a rather general and versatile computational machinery for the generation of tight LMs of groups of terms, and for their use in order to find "best" upper bounds—in a well-defined sense—on the optimum of an unconstrained quadratic 0–1 maximization problem. The above experimental results provide clear evidence that the proposed technique substantially improves on the bounds given by roof-duality, especially for larger and denser instances. The column generation procedure is highly flexible, allowing one to choose the total number of blocks, the number of columns added to the master problem at each iteration, and the maximum number of iterations. Straight column generation, in which one column at the time is added to the master problem, does not appear to be competitive, in terms of running times, with the Cplex primal simplex code. On the other hand, if column generation is truncated after a few iterations, the running times are much faster and the quality of the bound is not significantly affected. Our experiments on graphs with known maximum clique size k show that our bounds are not only considerably sharper than the roof-dual one, but also close to the optimum, when k lies in the interval $[n/4, n/2]$; however, they are weaker for instances with small ratio k/n. Moreover, in the experiments on the weighted max-2-sat problems proposed by Borchers, our bounds are very close to the optimum and, when the number of clauses is sufficiently large, they improve on the roof-dual values.

Certainly, further research is needed in order to select the block collection $\boldsymbol{F}$ in a more adaptive, instance-dependent, way; to design enhanced block search routines; to speed up the column generation algorithm through a suitable stabilization procedure [13], perhaps in the simpler form of constraint perturbation; to embed our upper bounding technique in a branch & bound exact code for quadratic 0–1 maximization; to investigate possible connections between our bounds for the maximum clique problem and other bounds reported in the literature (see, e.g., [11,21]; and, finally, to answer the many theoretical questions that were left open.

Acknowledgements

Part of this work was done during two visits of the first Author to the Department of Statistics, La Sapienza University, that were made possible by the support of the Italian Ministry of University and by the National Research Council of Italy; and during a visit of the third Author to RUTCOR and DIMACS (LSDO Special Year), Rutgers University, supported by these two centers.

The first Author also gratefully acknowledges the partial support by the Office of Naval Research (Grant N00014-92-J-1375) and by the National Science Foundation (Grant IIS-0118635).

We thank also the referees for their insightful comments, and Dr. Roberto Giaccio for his valuable assistance in our computational experimentation.

Appendix A. Tightest and closest LMs: properties and conjectures

In this appendix, some properties of tightest linear majorants and closest linear majorants will be presented, together with some conjectures. The validity of each conjecture has been established for all blocks with up to 4 variables.

We keep using the notation and definitions from Section 2. As in Section 4, let h be a given (template) block function of p variables, and set $N=2^p$. Let us introduce the (unbounded) polyhedron

$$P=\{(t_0,t)\in \mathbf{R}^{p+1}: t_0+\boldsymbol{t}\boldsymbol{x}^k \geqslant h(\boldsymbol{x}^k), \quad k=1,\dots,N\}.$$

For any $(t_0,t)\in P$, let $\mathrm{A}(t_0,t)=\{k: t_0+\boldsymbol{t}\boldsymbol{x}^k=h(\boldsymbol{x}^k)\}$.

Property A.1. If $g\in P$ is a non-negative combination of $g_i\in P$ for $i=1,2,\dots$, then $\mathrm{A}(g)\subseteq \mathrm{A}(g_i)$ holds for all $i=1,2,\dots$.

Property A.2. If $g,g'\in P$ are distinct vertices of P, then $\mathrm{A}(g)\neq \mathrm{A}(g')$.

Proof. Since the polyhedron P is clearly up-monotone and thus full-dimensional, the system of linear equations corresponding to $\mathrm{A}(g)$ is of full rank, and hence g is its unique solution. Similarly, g' is the unique solution of the full rank system of linear equations corresponding to $\mathrm{A}(g')$. Thus, $\mathrm{A}(g)\neq \mathrm{A}(g')$ follows by $g\neq g'$.

Corollary A.3. *If $g=t_0+\boldsymbol{t}\boldsymbol{x}$ is a TLM of h, then g is an extreme point of P, and therefore $|Cont(h,g)|\geqslant p+1$.*

Conjecture 1. *If g is a TLM of h and $|Cont(h,g)|=p+1$, then g is a roof.*

Property A.4. If g is a roof of h such that $Cont(h,g)\neq\emptyset$, then

(i) there is a roof g' whose coefficients are integers and $Cont(h,g')=Cont(h,g)$;
(ii) there are no two points $\boldsymbol{x},\boldsymbol{y}\in Cont(h,g)$ such that:

$$\text{either } x_i=1, x_j=0 \quad\text{and}\quad y_i=0, y_j=1 \text{ for some } ij\in E^+,$$

$$\text{or } x_i=0, x_j=0 \quad\text{and}\quad y_i=1, y_j=1 \text{ for some } ij\in E^-.$$

Proof. Since the roof

$$g(x)=\sum_{ij\in E^+}(\lambda_{ij}x_i+(1-\lambda_{ij})x_j)+\sum_{ij\in E^-}\lambda_{ij}(1-x_i-x_j),$$

where $0\leqslant\lambda_{ij}\leqslant 1$ for all $ij\in E$, is obtained by termwise bounding, a point $\boldsymbol{x}$ belongs to $Cont(h,g)$ iff

$$x_ix_j=\lambda_{ij}x_i+(1-\lambda_{ij})x_j, \quad ij\in E^+,$$

$$-x_ix_j=\lambda_{ij}(1-x_i-x_j), \quad ij\in E^-. \tag{21}$$

Hence if $ij\in E^+$

$$x_i=1,x_j=0\Rightarrow\lambda_{ij}=0,$$

$$x_i=0,x_j=1\Rightarrow\lambda_{ij}=1$$

and if $ij\in E^-$

$$x_i=0,x_j=0\Rightarrow\lambda_{ij}=0,$$

$$x_i=1,x_j=1\Rightarrow\lambda_{ij}=1.$$

From these implications (ii) easily follows.

In all remaining cases equalities (21) hold for arbitrary $0\leqslant\lambda_{ij}\leqslant 1$, thus the contact does not change if one always chooses, say, $\lambda_{ij}=1$. This proves (i).

Corollary A.5. *If some TLM of h is a roof, among the TLMs there is always a roof with integral coefficients.*

Conjecture 2. *There is always a TLM of h with integral coefficients.*

Conjecture 3. *For any block h, one can obtain every TLM of h by taking a suitable convex combination of TLMs of sub-blocks of h, and then drying the resulting LM.*

Conjecture 4. *Every TLM of a block is a CLM.*

Appendix B. Some results about the maximum clique problem

In this appendix we give the proofs of two results about the maximum clique problem, which were stated in Section 7. We keep here the notation of that section.

Property B.1. If in the optimal weighting problem (L) of Section 5 the collection $\boldsymbol{F}$ does not contain blocks with more than four variables, then $L^* \geqslant n/4$.

Proof. Since all the quadratic terms in (19) are negative and all the blocks in $\boldsymbol{F}$ have at most four variables, the only basic TLMs of the templates that can be used in (L) are the following:

$$\lambda(1 - x_1 - x_2),$$

$$(1 - x_1 - x_2 - x_3),$$

$$(1 - x_1 - x_2 - x_3 - x_4).$$

Hence the optimal value L^* of (L) is given by

$$\begin{aligned} L^* = \min \sum_{B\in \boldsymbol{F}} w_B + \sum_{i\in V} u_i \\ \text{s.t.} \quad u_i \geqslant 1 - \sum_{B:B\in \boldsymbol{F}, i\in B} w_B, \quad i \in V, \\ \sum_{B:ij\in E(B)} w_B = 1, \quad ij \in E', \\ \boldsymbol{u}, \boldsymbol{w} \geqslant \boldsymbol{0}, \end{aligned} \tag{22}$$

where, for the K_2 blocks, w_B is the product of the weight of the block and the parameter λ. The equality constraints of (22), together with the non-negativity of $\boldsymbol{w}$, imply the inequalities

$$\sum_{B\,:\,B\in \boldsymbol{F},\ i\in B} w_B \geqslant 1, \quad i \in V. \tag{23}$$

It follows that

$$\begin{aligned} L^* = \min \sum_{B\in \boldsymbol{F}} w_B \\ \text{s.t.} \quad \sum_{B:ij\in E(B)} w_B = 1, \quad ij \in E', \\ \boldsymbol{w} \geqslant \boldsymbol{0}. \end{aligned} \tag{24}$$

Hence L^* is bounded below by the optimal value L' of the relaxation of (24) obtained when the equality constraints are replaced by inequalities (23). Adding up all such inequalities one gets

$$\sum_{B\in \boldsymbol{F}} |B| w_B \geqslant n$$

and finally, since all blocks have cardinality at most 4, one obtains $L^* \geqslant L' \geqslant n/4$.

The above bound is attained when G' is partitionable into vertex-disjoint K_4's, all belonging to $\boldsymbol{F}$; for this to happen, G must admit a coloration whose colors have cardinality 4. Similarly, one can show that when only blocks with at most three variables are used then $L^* \geqslant n/3$, and that $RD = n/2$.

Property B.2. In any graph $G = (V, E)$ with n vertices, maximum vertex-degree d and largest clique size k, one has

$$\binom{k}{2} = \max_{x \in \mathbf{B^n}} \left(\sum_{ij \in E} x_i x_j - d \sum_{ij \notin E} x_i x_j \right). \tag{25}$$

Proof. Let f be the objective function in (25), $\boldsymbol{x}$ an arbitrary point in $\mathbf{B^n}$, S the subset of V whose characteristic vector is $\boldsymbol{x}$, and K a largest subset of S inducing a complete subgraph. For any two $A, B \subseteq V$, let $e(A, B)$ be the number of edges in G having one endpoint in A and the other one in B. Similarly for $e'(A, B)$ w.r.t. the complement G'. If $K = S$, then one has $f(\boldsymbol{x}) = e(K, K) \leqslant \binom{k}{2}$ the bound being attained when K is a largest clique in G.

On the other hand, when $K \subset S$, one has

$$f(\boldsymbol{x}) = e(K, K) + e(S - K, K) - de'(S - K, K) + e(S - K, S - K) - de'(S - K, S - K). \tag{26}$$

But

$$e(K, K) \leqslant \binom{k}{2};$$

$e'(S - K, K) \geqslant |S - K|$, since each vertex of $S - K$ is non-adjacent to some vertex of K;

$e(S - K, K) + e(S - K, S - K) \leqslant \sum_{i \in S-K}(d(i) - 1) < d|S - K|$, where $d(i)$ is the degree of vertex i in G, for the same reason.

From the above inequalities and (26), one gets $f(\boldsymbol{x}) < \binom{k}{2}$. Thus (25) is proved.

References

[1] W.P. Adams, P.M. Dearing, On the equivalence between roof duality and Lagrangian duality for unconstrained 0–1 quadratic programming problems, Technical Report No. URI-061, Clemson University, 1988.

[2] A.T. Amin, S.L. Hakimi, Upper bounds on the order of a clique of a graph, SIAM J. Appl. Math. 22 (1972) 569–573.

[3] B. Borchers, J. Furman, A two-phase exact algorithm for MAX-SAT and weighted MAX-SAT problems, J. Combin. Optim. 2 (1999) 299–306.

[4] E. Boros, Presentation at the Workshop on "Partitioning Data Sets, with Applications to Psychology, Vision, and Target Tracking, DIMACS, Rutgers University, April 19–21, 1993.

[5] E. Boros, Y. Crama, P.L. Hammer, Upper bounds for quadratic 0–1 maximization, Oper. Res. Lett. 9 (1990) 73–79.

[6] E. Boros, Y. Crama, P.L. Hammer, Chvátal cuts and odd cycle inequalities in quadratic 0–1 optimization, SIAM J. Discrete Math. 5 (1992) 163–177.

[7] E. Boros, P.L. Hammer, A max-flow approach to improved roof-duality in quadratic 0–1 minimization, Technical Report RRR 15-89, RUTCOR, Rutgers University, 1989.

[8] E. Boros, P.L. Hammer, Cut-polytopes, boolean quadric polytopes and nonnegative quadratic pseudo-boolean functions, Math. Oper. Res. 18 (1993) 245–253.

[9] E. Boros, P.L. Hammer, X. Sun, Network flows and minimization of quadratic pseudo-Boolean functions, RUTCOR Research Report RRR 17-91, RUTCOR, Rutgers University, 1991.

[10] J.-M. Bourjolly, P.L. Hammer, W.R. Pulleyblank, B. Simeone, Boolean-combinatorial bounding of maximum 2-satisfiability, in: O. Balci, R. Sharda, S.A. Zenios (Eds.), Computer Science and Operations Research: New Developments in their Interfaces, Pergamon Press, London, 1992, pp. 23–42.

[11] J. Bourjolly, G. Laporte, H. Mercure, A combinatorial column generation algorithm for the maximum clique and stable set problem, Technical Report G-94-33, Les Cahiers du Gerad, GERAD, Montréal, 1994.

[12] M. Budinich, P. Budinich, A Clifford algebra formulation of the maximum clique problem of a graph, preprint in preparation (will be available from http://www.ts.infn.it/~mbh/PubNN.html).

[13] O. Du Merle, D. Villeneuve, J. Desrosiers, P. Hansen, Stabilized column generation, Discrete Math. 194 (1999) 229–237.

[14] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-completeness, Freeman, San Francisco, 1979.

[15] P.L. Hammer, P. Hansen, B. Simeone, Upper planes of quadratic 0–1 functions and stability in graphs, in: O. Mangasarian, R.R. Meyer, S.M. Robinson (Eds.), Nonlinear Programming, Vol. 4, Academic Press, New York, 1981, pp. 395–414.
[16] P.L. Hammer, P. Hansen, B. Simeone, Roof-duality, complementation, and persistency in quadratic 0–1 optimization, Math. Programming 28 (1984) 121–155.
[17] P.L. Hammer, B. Kalantari, A bound on the roof-duality gap, in: B. Simeone (Ed.), Combinatorial Optimization, Lecture Notes in Mathematics, Vol. 1403, Springer, Heidelberg, 1989.
[18] P.L. Hammer, B. Simeone, Quadratic functions of binary variables, in: B. Simeone (Ed.), Combinatorial Optimization, Lecture Notes in Mathematics, Vol. 1403, Springer, Heidelberg, 1989, pp. 1–56.
[19] P. Hansen, S.H. Lu, B. Simeone, On the equivalence of paved duality and standard linearization in nonlinear 0–1 optimization, Discrete Appl. Math. 29 (1990) 187–193.
[20] S.H. Lu, A.C. Williams, Roof-duality for nonlinear 0–1 programming, Technical Report RRR 2-85, RUTCOR, Rutgers University, 1985.
[21] C. Mannino, A. Sassano, An exact algorithm for the maximum stable set problem, J. Comput. Optimization Appl. 3 (1994) 243–258.
[22] C.H. Papadimitriou, K. Steiglitz, Combinatorial Optimization, Prentice-Hall, Englewood Cliffs, NJ, 1982.
[23] L.A. Sanchis, Test case construction for the vertex cover problem, in: N. Dean, G.E. Shannon (Eds.), Computational Support for Discrete Mathematics, DIMACS Workshop, March 1992, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 15, American Mathematical Society, Providence, RI, 1994.
[24] L.A. Sanchis, A. Jagota, Some experimental and theoretical results on test case generators for the maximum clique problem, INFORMS J. Comput. 8 (2) (1996) 103–117.
[25] H.S. Wilf, The eigenvalues of a graph and its chromatic number, J. London Math. Soc. 42 (1967) 330–332.