Noname manuscript No.
(will be inserted by the editor)

Extended Linear Formulation for Binary Quadratic Problems

Fabio Furini · Emiliano Traversi

Received: date / Accepted: date

Abstract We propose and test a new linearisation technique for the Binary Quadratic Problems (BQPs). We computationally prove that the new formulation, called Extended Linear Formulation, can be effective for different classes of problems in practice. Our tests are based on two sets of classical BQPs from the literature, i.e., the Unconstrained BQP and the Maximum Cut of edge-weighted graphs. Finally we discuss the relations between the Linear Programming relaxations of the different linearisation techniques presented and we discuss the elimination of constraint redundancy which is effective at speeding up the computational convergence.

Keywords Binary Quadratic Problems · Linearization Techniques · Max Cut Problem.

1 Introduction

A widely used technique for solving the Binary Quadratic Problems (BQPs) is by formulating them as Mixed Integer Linear Programs (MILPs). The big advantage of this method is the possibility of using solvers available for tackling generic MILPs which have been strongly developed for decades and are constantly improving (see for example Lodi [18]). On the other hand, it is worth mentioning that BQPs can be tackled also using solvers specifically conceived

Fabio Furini
LAMSADE, Université Paris-Dauphine, Place du Maréchal de Lattre de Tassigny,
75775 Paris, France
E-mail: fabio.furini@dauphine.fr

Emiliano Traversi
LIPN, Université Paris 13, 99 Avenue Jean-Baptiste Clément,
93430 Villetaneuse, France
E-mail: emiliano.traversi@lipn.univ-paris13.fr

for non-linear problems. To mention just a few examples, we can cite some commercial softwares like BARON [3] and CPLEX [8]; as well as non commercial, for instance SCIP [20] and Bonmin [5]. All the techniques, implemented in these solvers and used to tackle BQPs directly, without any linearisation of the quadratic terms, are out of the scope of this paper.

Literature Review. Several linearisation techniques have been proposed in the literature for reformulating BQPs as equivalent MILPs. The seminal works of this stream of research are Fortet [9] and Glover and Woolsey [12]. These papers propose linearisation techniques based on additional binary variables. Then, since the difficulty of BQPs typically depends more strongly on the number of integer variables than on the number of continuous variables, other linearisation techniques have been proposed, based only on additional continuous variables and linear constraints. In this stream of research we cite the following works: Glover and Woolsey [13], Glover [11], Chaovalitwongse et al. [6] and Sherali and Smith [22]. All these formulations will be presented in detail in Section 2 and compared in the computational Section 3. It is worth mentioning that other linearisation techniques have been also proposed but they will not be treated in this paper. Among these techniques, we quote the following ones. A tighter reformulation in terms LP-relaxation was proposed by Adams and Sherali [2], this approach was subsequently generalized to design the reformulation–linearization technique (RLT) in Sherali and Adams [21]. Recently other interesting linearisation techniques have been proposed in Hansen and Meyer [15] and in Gueye and Michelon [14].

Paper Contribution. In this work we focus on analysing an alternative linear formulation for BQPs. The linearisation proposed is called Extended Linear Formulation (ELF) and is valid for a generic BQP. This new formulation has been inspired by the ideas proposed in Jaumard et al. [16], where a reformulation specifically conceived for the Quadratic Stable Set Problem (QSSP) is proposed. The ELF is characterized by the same Linear Programming Relaxation (LP-relaxation) value of the standard linearisation technique (see Section 2.1) but it presents a slightly higher number of variables and constraints (same order of magnitude). Extended formulations (see for example Conforti et al. [7]) have been introduced to derive tighter formulations of hard combinatorial optimization problems, i.e. formulations that can provide stronger dual bounds (see for example Bertsimas and Weismantel [4]). Typically these formulations are characterized by a higher number of constraints or variables. Similarly, in this paper we use extended formulations but this time in order to obtain a new linearisation technique. A second contribution of this paper is a discussion of the Linear Programming relaxations of the different linearisation techniques presented. This analysis has brought to some new interesting interconnections between the classical linearisation techniques and to the identification of subsets of constraints which are redundant for the formulations. Finally the removal of this redundancy allows a significant speed up in terms of computing time necessary to solve the Linear Programming relaxations of each formulation respectively.

2 Linearisation techniques for Binary Quadratic Problems

A generic Binary Quadratic Problem (BQP), with n variables and p constraints, can be formulated using the following Quadratic Formulation (QF).

$$\begin{aligned}\text{QF}: \quad & \min \sum_{i=1}^{n}\sum_{j=1}^{n} Q_{ij}x_i x_j + \sum_{i=1}^{n} L_i x_i \\ & x \in K \\ & x \in \{0,1\}^n,\end{aligned}$$

where $Q \in \mathbb{R}^{n\times n}$, $L \in \mathbb{R}^n$, $K = \{x \in \mathbb{R}^n : Ax \geq b\}$, $A \in \mathbb{R}^{p\times n}$ and $b \in \mathbb{R}^p$. Q is a generic symmetric matrix, not restricted to being convex. In the following we describe several alternatives for linearising BQPs present in the literature and then we introduce a new extended linear formulation.

2.1 Glover-Woolsey Linear Formulation

The standard method for linearising the quadratic terms is the one introduced by Glover and Woolsey and described in [13]. This linear formulation, called $\text{GW}_{[13]}$, reads as follows:

$$\begin{aligned}\text{GW}_{[13]}: \quad & \min \sum_{i=1}^{n}\sum_{j=i}^{n} Q_{ij}y_{ij} + \sum_{i=1}^{n} L_i x_i && \\ & y_{ij} \leq x_i && i,j = 1,\ldots,n, \quad i<j \qquad (1) \\ & y_{ij} \leq x_j && i,j = 1,\ldots,n, \quad i<j \qquad (2) \\ & y_{ij} \geq x_i + x_j - 1 && i,j = 1,\ldots,n, \quad i<j \qquad (3) \\ & y_{ij} \geq 0 && i,j = 1,\ldots,n, \quad i<j \qquad (4) \\ & x \in K && \\ & x \in \{0,1\}^n\ . &&\end{aligned}$$

The new variables y_{ij} take the place of the products between the original variables x_i and x_j in the objective function.We recall that $\text{GW}_{[13]}$ increases the size of the problem by adding $n(n-1)/2$ variables and $4n(n-1)/2$ constraints.

The number of constraints can be reduced by eliminating redundant constraints. A constraint is redundant for a formulation if its removal does not change the set of optimal solution of its LP-relaxation. Since we are interested in a feasible region K described by only linear constraints (i.e. not involving directly the variables y_{ij}), we can take into account the signs of the entries of the quadratic cost matrix Q and eliminate some of them. This can be resumed in the following observation:

Observation 1 *Inequalities* (1) *and* (2)*, corresponding to non-positive entries of* Q*, and inequalities* (3) *and* (4)*, corresponding to non-negative entries of* Q*, are redundant for* $\mathrm{GW}_{[13]}$.

Observation 1 is due to the fact that the value of each variable y_{ij} is uncorrelated from the values of the other y variables and depends only on the bounds defined by (1)-(4). Moreover, the optimization sense of $\mathrm{GW}_{[13]}$ pushes the y variables to their bounds and hence inequalities (1) and (2), corresponding to non-positive entries of Q, and inequalities (3) and (4), corresponding to non-negative entries of Q, will never be active.

$\mathrm{GW}_{[13]}$ can be strengthened by applying the so-called Reformulation Linearisation Technique (RLT) presented in Sherali and Adams [21]. The RLT is a procedure divided into two steps: the reformulation step creates additional nonlinear constraints by multiplying the constraints in K by product factors of the binary variables x and their complements $(1-x)$, and subsequently enforces the identity $x^2 = x$. The linearisation step then substitutes a continuous variable for each distinct product of variables. A full characterization of the convex hull is available at the n-th level. Hence, $\mathrm{GW}_{[13]}$ can be viewed as first level RLT applied only on the bound constraints $0 \leq x \leq 1$.

2.2 Glover Linear Formulation

In this section we introduce the linearisation described in Glover [11]. This linear formulation, called $\mathrm{G}_{[11]}$, reads as follows.

$$\begin{aligned}
\mathrm{G}_{[11]}: \quad \min \ & \sum_{i=1}^{n} w_i + \sum_{i=1}^{n} L_i x_i \\
& w_i \leq Q_i^+ x_i && i = 1, \ldots, n && (5)\\
& w_i \geq Q_i^- x_i && i = 1, \ldots, n && (6)\\
& w_i \leq \sum_{j=1}^{n} Q_{ij} x_j - Q_i^-(1 - x_i) && i = 1, \ldots, n && (7)\\
& w_i \geq \sum_{j=1}^{n} Q_{ij} x_j - Q_i^+(1 - x_i) && i = 1, \ldots, n && (8)\\
& x \in K \\
& x \in \{0, 1\}^n \ ,
\end{aligned}$$

where Q_i^- and Q_i^+ are suitable large constants. The main intuition behind this linearisation is the introduction of the additional set of variables w_i representing the quantity $\sum_{j=1}^{n} Q_{ij} x_j$ if $x_i = 1$, or taking a value of 0 otherwise. Formulation $\mathrm{G}_{[11]}$ has the big advantage of using less variables and constraints than $\mathrm{GW}_{[13]}$. On the other hand, it requires the introduction of the so-called "big-M" constraints, leading to a weaker LP-relaxation.

An important point concerns the computation of Q_i^- and Q_i^+. In Glover [11], the author suggest to impose:

$$Q_i^- = \sum_{j=1}^{n} \min\{0, Q_{ij}\}, \quad Q_i^+ = \sum_{j=1}^{n} \max\{0, Q_{ij}\}\ . \tag{9}$$

Subsequently, in several works (see for example Adams et al. [1] and Wang et al. [23]) this approach has been improved by taking into account the feasible region K and hence computing the values for Q_i^- and Q_i^+ with more accuracy. This can be done as follows:

$$Q_i^- = \min_{x \in K} \sum_j Q_{ij} x_j, \quad Q_i^+ = \max_{x \in K} \sum_j Q_{ij} x_j. \tag{10}$$

It is easy to check that equations (9) and equations (10) coincide when $K = \emptyset$. From now on, when we refer to $\mathrm{G}_{[11]}$ we imply the version with Q_i^- and Q_i^+ defined in (10). Regardless of the method used for computing Q_i^- and Q_i^+, $\mathrm{G}_{[11]}$ increases the size of the problem by adding only n variables and $4n$ constraints. However, since we are minimizing and the coefficients of the w variables are positive, it is possible to reduce the number of constraints:

Observation 2 *Inequalities* (5) *and* (7) *are redundant for* $\mathrm{G}_{[11]}$.

The motivations for Observation 2 are analogous to the ones used for Observation 1.

2.3 Sherali-Smith Linear Formulation

The third method linearises the quadratic terms using the techniques described in Sherali and Smith [22]. This linear formulation, called $\mathrm{SS}_{[22]}$, reads as follows.

$$\begin{aligned}
\mathrm{SS}_{[22]}:\quad \min\ & \sum_{i=1}^{n} s_i + \sum_{i=1}^{n} (L_i + Q_i^-) x_i && \\
& y_i = \sum_{j=1}^{n} Q_{ij} x_j - s_i - Q_i^- && i = 1, \ldots, n \quad (11)\\
& y_i \le (Q_i^+ - Q_i^-)(1 - x_i) && i = 1, \ldots, n \quad (12)\\
& s_i \le (Q_i^+ - Q_i^-) x_i && i = 1, \ldots, n \quad (13)\\
& y_i \ge 0 && i = 1, \ldots, n \quad (14)\\
& s_i \ge 0 && i = 1, \ldots, n \quad (15)\\
& x \in K && \\
& x \in \{0,1\}^n\ , &&
\end{aligned}$$

where Q_i^+ and Q_i^- are defined as in (10). The idea behind $\mathrm{SS}_{[22]}$ is similar to the one behind $\mathrm{GW}_{[13]}$, i.e., the introduction of n additional variables representing this time the quantity $\sum_{j=1}^n Q_{ij} - Q_i^-$ using big-Ms constraints. $\mathrm{SS}_{[22]}$ increases the size of the problem by adding $2n$ variables, $4n$ inequalities and n equations.

Formulation $\mathrm{SS}_{[22]}$ is a strengthening of a precedent formulation proposed by Chaovalitwongse et al. [6] called $\mathrm{CPP}_{[6]}$ which uses instead the following definition of Q_i^+ and Q_i^-:

$$Q_i^+ = \max_i \sum_{j=1}^n |Q_{ij}|, \quad Q_i^- = -\max_i \sum_{j=1}^n |Q_{ij}|.$$

$\mathrm{CPP}_{[6]}$ is weaker than $\mathrm{SS}_{[22]}$. This is due to the fact of using worse big-M values (see Sherali and Smith [22]).

Like for $\mathrm{G}_{[11]}$, also for $\mathrm{SS}_{[22]}$ (and analogously for $\mathrm{CPP}_{[6]}$) some inequalities can be eliminated:

Observation 3 *Inequalities* (13) *and* (14)*, are redundant for* $\mathrm{SS}_{[22]}$.

The motivations for Observation 3 are analogous to the ones given for Observation 2. The strong connection between $\mathrm{G}_{[11]}$ and $\mathrm{SS}_{[22]}$ is finally confirmed by the following Observation:

Observation 4 *The LP-relaxation of* $\mathrm{G}_{[11]}$ *and* $\mathrm{SS}_{[22]}$ *are identical.*

Proof As fist step we apply to $\mathrm{SS}_{[22]}$ the variables substitution $s_i = w_i - Q_i^- x_i$ and subsequently equations (11) can be used to substitute variables y into the remaining constraints and into the objective function, thus obtaining exactly $\mathrm{G}_{[11]}$. □

2.4 Extended Linear Formulation

We now introduce a new linearisation, called Extended Linear Formulation (ELF).

$$\begin{aligned}
\mathrm{ELF}: \quad \min \ & \sum_{i=1}^n \sum_{j=i}^n Q_{ij} + \sum_{i=1}^n L_i x_i - \sum_{i=1}^n \sum_{j=i}^n Q_{ij}(z_{ij}^i + z_{ij}^j) \\
& z_{ij}^i + z_{ij}^j \le 1 \qquad i,j = 1,\dots,n, \quad i<j \qquad (16)\\
& x_i + z_{ij}^i \le 1 \qquad i,j = 1,\dots,n, \quad i<j \qquad (17)\\
& x_j + z_{ij}^j \le 1 \qquad i,j = 1,\dots,n, \quad i<j \qquad (18)\\
& x_i + z_{ij}^i + z_{ij}^j \ge 1 \quad i,j = 1,\dots,n, \quad i<j \qquad (19)\\
& x_j + z_{ij}^i + z_{ij}^j \ge 1 \quad i,j = 1,\dots,n, \quad i<j \qquad (20)\\
& x \in K \\
& x \in \{0,1\}^n \ .
\end{aligned}$$

This linearisation increases the size of the problem by adding $2n(n-1)$ variables and $5n(n-1)/2$ constraints. A pair of new variables z^i_{ij} and z^j_{ij} is used instead of the product of variables x_i and x_j. These new variables modify the objective function and appear in the new set of constraints. The total sum of the quadratic costs is paid in the objective function ($\sum_{i=1}^n \sum_{j=1}^n Q_{ij}$) and then the correct value is reconstructed with the use of the z variables ($\sum_{i=1}^n \sum_{j=1}^n Q_{ij}(z^i_{ij} + z^j_{ij})$). The values of the z variables are set according to the values of the x variables thanks to constraints (16)-(20), i.e. if $x_i = x_j = 1$ then $z^i_{ij} = z^j_{ij} = 0$, on the other hand if one or both variables x_i and x_j have a value of 0 then one of the variables z^i_{ij} and z^j_{ij} is forced to have a value of 1. In the final part of this section we introduce two properties of the ELF. The first Observation shows that ELF is an extended formulation of $\text{GW}_{[13]}$:

Observation 5 *The LP-relaxation of* $\text{GW}_{[13]}$ *and* ELF *are identical.*

Proof Let $(\bar{x}, \bar{y})$ be a feasible solution to the LP-relaxation of $GW_{[13]}$. We consider the following solution $(\tilde{x}, \tilde{z}^i, \tilde{z}^j)$ of the LP-relaxation of ELF: $\tilde{x} = \bar{x}$, $\tilde{z}^i = 1 - \bar{x}_i$, $\tilde{z}^j = \bar{x}_i - \bar{y}$. It is easy to check that it is feasible and it has the same value. Now let $(\tilde{x}, \tilde{z}^i, \tilde{z}^j)$ be a feasible solution to the LP-relaxation of ELF. We consider the following solution $(\bar{x}, \bar{y})$ of the LP-relaxation of LF: $\bar{x} = \tilde{x}$, $\bar{y} = 1 - \tilde{z}_i - \tilde{z}_j$. Again, it is easy to check that it is feasible and it has the same value. This imply that for every feasible solution of the LP-relaxation of $\text{GW}_{[13]}$ there exists one feasible solution of the LP-relaxation of ELF of the same value and vice versa, proving the statement (i.e. they are identical). □

Observation 5 implies that every valid inequality for $\text{GW}_{[13]}$ is also valid for ELF. A second Observation allows the reduction of the number of constraints also for ELF:

Observation 6 *Inequalities* (19) *and* (20)*, corresponding to non-positive entries of* Q*, and inequalities* (16)*,* (17) *and* (18)*, corresponding to non-negative entries of* Q *are redundant for ELF.*

Proof Let $(\tilde{x}, \tilde{z}^i, \tilde{z}^j)$ be a solution to LP-relaxation of ELF satisfying all inequalities except some of the (19)-(20) associated to an entry $Q_{ij} \leq 0$. Let $x_i + z^i_{ij} + z^j_{ij} \geq 1$ be one of the violated inequalities, this means that $\tilde{x}_i + \tilde{z}^i_{ij} + \tilde{z}^j_{ij} < 1$, hence one value $\hat{z}^i > \tilde{z}^i$ exists such that $\tilde{x}_i + \hat{z}^i_{ij} + \tilde{z}^j_{ij} = 1$. The point $(\tilde{x}, \hat{z}^i, \tilde{z}^j)$ is still feasible and has a better objective function (because Q_{ij} is non-positive). This means that $(\tilde{x}, \tilde{z}^i, \tilde{z}^j)$ is not an optimal solution. Similarly, let $(\tilde{x}, \tilde{z}^i, \tilde{z}^j)$ be a solution to LP-relaxation of ELF satisfying all inequalities except some of the (17)-(18), associated to an entry $Q_{ij} \geq 0$. Let $x_i + z^i_{ij} \leq 1$ be one of the violated inequalities, this means that $\tilde{x}_i + \tilde{z}^i_{ij} > 1$, hence one value $\hat{z}^i > \tilde{z}^i$ exists such that $\tilde{x}_i + \hat{z}^i_{ij} = 1$. The point $(\tilde{x}, \hat{z}^i, \tilde{z}^j)$ is still feasible and has a better objective function (because Q_{ij} is non-negative). This means that $(\tilde{x}, \tilde{z}^i, \tilde{z}^j)$ is not an optimal solution. Similar considerations can be done for inequalities (16). Hence, eliminating from ELF the inequalities (19) and (20), corresponding to a couple of indices i and j with non-negative Q_{ij} or

Formulations	Variables	Constraints			
		Original		Reduced	
		#Ineq.	#Eq.	#Ineq.	#Eq.
GW[13]	$n(n-1)$	$4n(n-1)$	0	$2n(n-1)$	0
G[11]	n	$4n$	0	$2n$	0
CPP[6], SS[22]	$2n$	$4n$	n	$2n$	n
ELF	$2n(n-1)$	$5n(n-1)$	0	$\approx \frac{5}{2}n(n-1)$	0

Table 1 Resume of the additional variables and constraints used in each formulation.

the inequalities (16), (17) and (18), corresponding to a couple of indices i and j with non-positive Q_{ij}, does not change the set of optimal solutions of the LP-relaxation of ELF (i.e., they are redundant). □

In Jaumard et al. [16], the authors proved that for the case of the QSSP with non-negative Q_{ij}, ELF without inequalities (19)-(20) is a valid linearisation. In other words the QSSP reduces to a Stable Set Problem on a suitable extended graph. For further details on the QSSP we also refer the interested reader to Furini and Traversi [10]. In case of general Q, the stable set structure of the problem is modified by the additional constraints (19)-(20), in other words, the convex hull of the feasible solution of ELF can be viewed as the convex hull of the stable set polytope intersected with the additional constraints (19)-(20).

2.5 Formulation Summary

In Table 1, we report the number of additional variables and constraints needed by each formulation respectively. The columns concerning the constraints are subdivided in two parts. Columns – Original – report the number of inequalities and equations needed by each formulation while columns – Reduced – report the number of non redundant inequalities and equations needed by each formulation (see Observations 1, 2, 3 and 6). In the following, this reduction of the formulation dimensions will be referred to as `Constraint-Redundancy` elimination. As far as GW[13], G[11], CPP[6] and SS[22] are concerned, regardless of the sign of the entries of Q, the `Constraint - Redundancy` elimination allows us to reduce by one half the number of additional constraints. For ELF the situation is slightly different because the number of non redundant constraints depends on the sign of Q and it is hence included in the interval $[2n(n-1), 3n(n-1)]$, for this reason we reported the average value of $\frac{5}{2}n(n-1)$.

In Table 2, we summarize the relations between the different LP-relaxations of the formulations studied. The relation "A ⇔ B" stands for "A and B have the same LP-relaxation" and "A ⇒ B" stands for "A has a stronger LP-

$$\begin{tikzcd}
\text{LP of GW}_{[13]} \arrow[r, Rightarrow, "\text{(see [11])}"] \arrow[d, Leftrightarrow, "(\textit{Obs.}\ 5)"] & \text{LP of G}_{[11]} \arrow[d, Leftrightarrow, "(\textit{Obs.}\ 4)"] & \\
\text{LP of ELF} & \text{LP of SS}_{[22]} \arrow[r, Rightarrow, "\text{(see [22])}"] & \text{LP of CPP}_{[6]}
\end{tikzcd}$$

Table 2 Resume of the relations between the strength of the LP-relaxations.

relaxation than B". For each relation, we report the reference where it is proved.

3 Computational experiments

In this section, we assess the computational performances of the linearisation techniques discussed in this paper, i.e., GW$_{[13]}$, G$_{[11]}$, CPP$_{[6]}$ and SS$_{[22]}$ and the new ELF. We first describe the *test problems* and then we report the *tables' discussion.*

Test problems. Our primary aim is to investigate the strength of the different formulations without the influence of additional constraints, and hence we focus on unconstrained problems ($K = \emptyset$). We adopt the Biq Mac library (see Wiegele [24]) as a case study. It is a collection of instances widely used in the literature, see for example Rendl et al. [19] or more recently Krislock et al. [17]. This library is composed by two families of problems: the first one is the *Unconstrained BQP* (UBQP):

$$\min\{\sum_{i=1}^{n}\sum_{j=i}^{n} Q_{ij}x_i x_j : x \in \{0,1\}^n\} \ ,$$

where Q is a symmetric matrix of order n. The second family is the *Max Cut* (MC) problem. The MC is to determine a maximum weighted bipartition of a graph G of n vertices (see for example Rendl et al. [19]) and it can be formulated as follows:

$$\max\{\sum_{i=1}^{n}\sum_{j=i}^{n} \tilde{Q}_{ij}x_i x_j : x \in \{-1,1\}^n\} \ ,$$

where $\tilde{Q}$ is the Laplacian matrix of the graph G. It is well known that the UBQP and the MC are equivalent (see Krislock et al. [17]), i.e., the MC can be transformed into an UBQP by considering a change of variable (note that in our experiments we used the UBQP version of the MC problem). The BicMac library is divided into five classes: `beasley`, `gka`, and `be` are UBQP instances, while `rudy`, `ising` are MC instances. The classes altogether form a test bed of 343 instances. Some are randomly generated instances and others come from a statistical physics application. In each class, the instances differ in terms of size n and density of the matrices Q and L (for further details on the instance features, we refer the reader to Wiegele [24]).

Tables' discussion. We run the tests on a PC with an Intel(R) Core2 Duo CPU E6550 at 2.33GHz and 2 GB RAM memory, under Linux Ubuntu 12, 64-bit. In the remaining part of this section, we discuss two sets of experiments aiming respectively at comparing the LP-relaxation of the different formulations and the computational behaviour for solving the test problems to proven optimality. In both tables, i.e., Table 3 and Table 5, we report the results concerning the whole Biq Mac Library. These tables are divided into 5 horizontal blocks, one for each class of instances (`be`, `beasly`, `gka`, `ising` and `rudy`), in each line we group together the results relative to instances of the same size (same values of n). In this way we create subclasses of instances and we report their average arithmetic values. The first two columns of both tables report the size of the subclasses of instances (value of n) and their cardinality (i.e., the number of instances of a specific subclass). For each run, we set a time limit of 600 seconds using `CPLEX` 12.4 [8] with default parameter settings. In case the time limit is reached for all the instances of a specific subclass and formulation, we report tl.

In Table 3 we report the results concerning the solution of the LP of the different formulations. The discussion of the results will be done for couples of formulations having the same value of LP-relaxation, i.e. $\text{G}_{[11]}$ – $\text{SS}_{[22]}$ (see Observation 4) and $\text{GW}_{[13]}$ – ELF (see Observation 5). The table is vertically divided into three parts. In the first part, called – LP times – , we report the average time in seconds necessary to solve the LP-relaxation. In the second part, called – LP iterations – , we report the average number of simplex iterations necessary to solve the LP-relaxation. As far as the comparison between $\text{G}_{[11]}$ and $\text{SS}_{[22]}$ is concerned, it is interesting to note that $\text{SS}_{[22]}$ tends to outperforms $\text{GW}_{[13]}$ for both the computational time and the simplex iterations. This better behaviour can be seen clearly comparing subclasses of instances with large values of n where $\text{SS}_{[22]}$ utilises roughly 50% of the simplex iterations less than $\text{GW}_{[13]}$. This behaviour can be possibly explained by the simplified version of the linearisation constraints. As far as the comparison between $\text{GW}_{[13]}$ and ELF is concerned, $\text{GW}_{[13]}$ slightly outperforms ELF for both the computational time and the simplex iterations. This behaviour can be explained considering the lower number of variables and constraints of the $\text{GW}_{[13]}$ formulation. There are no relevant differences to be reported between the behaviour of the formulations considering different classes of instances. In the third part of the table, called – LP comparisons –, we report the relative percentage gap between the LP-relaxation of $\text{G}_{[11]}$ and $\text{SS}_{[22]}$ with respect to LP-relaxation of $\text{GW}_{[13]}$ and ELF. It is worth stressing that, for the Mac instances, all the formulations have the same LP-relaxation values. For the Biq instances instead, $\text{GW}_{[13]}$ and ELF are characterized by stronger LP-relaxation values (this difference is less marked for the `be` class of instances). Finally we report the relative percentage gap of the formulation $\text{CPP}_{[6]}$ with respect to LP-relaxation of formulations $\text{GW}_{[13]}$ and ELF. In all the classes of instances $\text{CPP}_{[6]}$ is characterized by weaker LP-relaxation values, this is due to the fact of having larger big-M values. Since on our test bed this formulation is

			LP times				LP iterations				LP comparisons	
	n	#	$\mathrm{GW}_{[13]}$	$\mathrm{G}_{[11]}$	$\mathrm{SS}_{[22]}$	ELF	$\mathrm{GW}_{[13]}$	$\mathrm{G}_{[11]}$	$\mathrm{SS}_{[22]}$	ELF	$\mathrm{G}_{[11]}$, $\mathrm{SS}_{[22]}$	$\mathrm{CPP}_{[6]}$
be	*100*	*10*	0.1	0.0	0.0	0.7	4292	213	100	9084	0.00	56.56
	120	*20*	0.1	0.0	0.0	0.5	3451	278	122	7311	0.25	62.29
	150	*20*	0.5	0.0	0.0	1.0	4900	350	151	11220	0.09	61.26
	200	*20*	1.4	0.0	0.0	3.4	8669	456	201	19718	0.01	60.24
	250	*10*	0.1	0.1	0.1	0.2	2798	632	292	6070	1.79	69.93
beasly	*50*	*10*	0.0	0.0	0.0	0.0	121	99	63	194	10.10	72.87
	100	*10*	0.0	0.0	0.0	0.0	676	261	137	1055	11.37	75.63
	250	*10*	0.1	0.1	0.1	0.2	2822	688	279	6111	1.24	69.24
	500	*10*	1.0	0.5	0.4	2.3	10543	1293	504	23065	0.06	67.01
gka	*20*	*1*	0.0	0.0	0.0	0.0	28	21	21	408	3.77	95.22
	30	*3*	0.0	0.0	0.0	0.0	172	54	35	561	4.92	75.57
	40	*2*	0.0	0.0	0.0	0.0	338	67	41	1369	3.45	79.28
	50	*4*	0.0	0.0	0.0	0.0	312	99	59	1114	7.36	77.22
	60	*3*	0.0	0.0	0.0	0.0	329	104	71	1721	5.98	77.46
	70	*3*	0.0	0.0	0.0	0.0	390	125	82	2247	7.87	79.98
	80	*3*	0.0	0.0	0.0	0.0	466	154	98	2755	10.69	80.75
	90	*2*	0.0	0.0	0.0	0.1	341	151	109	4355	10.30	86.03
	100	*13*	0.0	0.0	0.0	0.2	1925	220	109	4815	3.10	67.21
	125	*1*	0.0	0.0	0.0	0.7	182	117	121	14876	7.36	99.36
	200	*5*	0.4	0.0	0.0	0.8	4898	482	210	10985	1.16	65.90
	500	*5*	71.0	0.7	0.8	153.8	50312	1199	500	114622	0.00	58.76
ising	*100*	*9*	0.1	0.0	0.0	0.2	3437	217	101	6405	0.00	79.78
	125	*3*	0.0	0.0	0.0	0.0	510	256	127	907	0.00	72.94
	150	*6*	0.2	0.0	0.0	1.1	10253	333	150	18812	0.00	76.13
	200	*6*	0.4	0.1	0.1	2.2	15686	447	200	28246	0.00	79.74
	216	*3*	0.0	0.0	0.0	0.0	777	454	218	1548	0.00	72.53
	225	*3*	0.0	0.0	0.0	0.0	633	469	235	1129	0.00	75.23
	250	*6*	0.6	0.1	0.1	3.4	20947	564	250	37984	0.00	78.97
	300	*6*	0.9	0.2	0.1	4.6	26308	680	300	48105	0.00	78.95
	343	*3*	0.0	0.1	0.0	0.1	1233	712	346	2401	0.00	74.87
	400	*3*	0.0	0.0	0.0	0.1	1117	852	416	2033	0.00	76.24
rudy	*60*	*10*	0.0	0.0	0.0	0.0	619	63	63	2345	0.00	40.09
	80	*30*	0.0	0.0	0.0	0.1	1372	144	81	3480	0.00	54.24
	100	*90*	0.0	0.0	0.0	0.2	1917	161	103	5526	0.00	54.26

Table 3 Computational comparison of LP-relaxation of the different formulations for Biq Mac instances

computationally dominated by $\mathrm{SS}_{[22]}$, it will be dropped from the comparison tests presented in the following tables.

In Table 4 we report the results concerning the `Constraint-Redundancy` elimination possibilities for the different formulations. The table is divided in two parts, the first one concerns the LP-relaxation computing time and the second one concerns the number of simplex iterations. Each entry of the table reports the ratio between the average values obtained without and with the `Constraint-Redundancy` elimination. As far as the computing times are concerned, surprisingly, the `Constraint-Redundancy` elimination has a great impact on the $\mathrm{GW}_{[13]}$. For this formulation and for some subclasses of instances, a reduction of up to 3 orders of magnitude can be achieved. A similar behaviour, but less marked, can be seen for ELF. The other two formulations,

			Constraint-Redundancy elimination							
			LP time ratios				LP iteration ratios			
	n	#	GW$_{[13]}$	G$_{[11]}$	SS$_{[22]}$	ELF	GW$_{[13]}$	G$_{[11]}$	SS$_{[22]}$	ELF
ising	*100*	*9*	2.8	1.8	1.1	3.3	1.4	1.0	1.0	1.6
	125	*3*	40.3	1.0	1.0	94.8	13.1	1.0	1.0	15.2
	150	*6*	18.1	1.6	1.1	2.2	1.1	1.0	1.0	1.2
	200	*6*	30.4	1.7	1.1	2.7	1.2	1.0	1.0	1.4
	216	*3*	1013.5	1.2	1.2	111.2	24.0	1.0	1.0	25.3
	225	*3*	473.5	1.0	1.7	130.1	37.5	1.0	1.0	40.8
	250	*6*	54.9	1.7	1.1	2.9	1.4	1.0	1.0	1.6
	300	*6*	77.8	1.6	1.0	3.5	1.8	1.0	1.0	1.7
	343	*3*	3186.0	1.2	1.2	216.6	40.0	1.0	1.0	41.8
	400	*3*	1141.5	1.4	1.0	284.1	70.1	1.0	1.0	75.1
rudy	*60*	*10*	2.6	1.0	1.0	3.7	1.9	1.0	1.0	1.7
	80	*30*	3.1	1.0	1.0	2.9	1.7	1.0	1.0	1.7
	100	*90*	4.4	1.3	1.0	3.6	1.9	1.0	1.0	1.6

Table 4 Ratios between the average time and the average number of simplex iterations before and after applying the `Constraint-Redundancy` policies for Mac instances

i.e., G$_{[11]}$ and SS$_{[22]}$, are instead less conditioned. As far as the number of simplex iterations is concerned, the `Constraint-Redundancy` elimination has again a large impact on GW$_{[13]}$ and ELF (with a reduction of almost two orders of magnitude). G$_{[11]}$ and SS$_{[22]}$, instead, maintain the same performance. We only report the data for the Mac instances since the behaviour for the Biq instances is similar. Summarizing, if we only consider the formulations without the `Constraint - Redundancy` elimination, surprisingly the ELF outperforms GW$_{[13]}$.

In Table 5 we report the results concerning the computational behaviour for solving the test problems to proven optimality. The table is vertically divided into three parts. In the first part, called – MILP times –, we report the average time in seconds necessary to solve the instances. If some of them are not solved within the time-limit of 600 seconds, we report the number of instances solved to optimality (we report tl and 0 in case all the instances of a specific subclass and formulation reach the time limit). The winning formulation is reported in bold text, i.e. less computing time or larger number of instances solved respectively. In the second part, called – MILP branching nodes –, we report the total number branching nodes. The formulations with the better computational behaviour for the Bic Mac instances are GW$_{[13]}$ and ELF. These formulations clearly outperform GW$_{[13]}$ and SS$_{[22]}$ concerning both the average computing time and the total number of instances solved. For the Biq instances, this is due to the fact of having stronger LP-relaxation values which allows a better computational convergence. For the Mac instances, where the LP-relaxation values coincides, the different behaviour can be explained by the better performance of GW$_{[13]}$ and ELF during the branching scheme or by the efficacy of the generic cuts of `CPLEX`. In the third part, called – MILP exit gaps –, we report the exit gap between the upper and the lower bounds computed by `CPLEX` in case the time limit is reached (0.0 is reported

			MILP times								MILP branching nodes				exit gaps			
	n	#	GW[13]		G[11]		SS[22]		ELF		GW[13]	G[11]	SS[22]	ELF	GW[13]	G[11]	SS[22]	ELF
be	*100*	*10*	tl	**0**	tl	**0**	tl	**0**	tl	**0**	0	615209	207440	37981	**44.7**	52.8	51.7	71.7
	120	*20*	tl	**0**	tl	**0**	tl	**0**	tl	**0**	1418	553856	90534	87416	**33.2**	45.2	43.5	57.8
	150	*20*	tl	**0**	tl	**0**	tl	**0**	tl	**0**	411	350572	59904	59286	56.9	**51.8**	53.2	66.0
	200	*20*	tl	**0**	tl	**0**	tl	**0**	tl	**0**	0	180912	55496	23142	74.5	**58.9**	64.3	74.2
	250	*10*	tl	**0**	tl	**0**	tl	**0**	tl	**0**	1647	299116	27184	90711	**29.0**	34.4	32.4	36.7
beasley	*50*	*10*	**0.0**	**10**	0.1	**10**	0.0	**10**	0.0	**10**	0	116	1	0	**0.0**	**0.0**	0.0	**0.0**
	100	*10*	1.7	**10**	556.2	2	15.9	**10**	**0.9**	**10**	39	1390824	4158	239	0.0	3.8	0.0	**0.0**
	250	*10*	tl	**0**	tl	**0**	tl	**0**	tl	**0**	1759	314432	27306	90936	**31.5**	35.9	34.2	38.8
	500	*10*	tl	**0**	tl	**0**	tl	**0**	tl	**0**	0	32398	60496	13602	97.7	**57.2**	59.5	63.5
gka	*20*	*1*	0.1	**1**	**0.0**	**1**	0.1	**1**	**0.0**	**1**	0	30	43	1	**0.0**	**0.0**	**0.0**	**0.0**
	30	*3*	0.2	**3**	0.2	**3**	0.2	**3**	**0.2**	**3**	2	560	108	21	**0.0**	**0.0**	**0.0**	**0.0**
	40	*2*	2.2	**2**	4.8	**2**	3.9	**2**	**1.3**	**2**	24	14194	4179	391	**0.0**	**0.0**	0.0	**0.0**
	50	*4*	**2.6**	**4**	78.8	**4**	37.7	**4**	3.0	**4**	20	236857	29454	1275	**0.0**	**0.0**	0.0	**0.0**
	60	*3*	2.9	**3**	146.8	**3**	14.9	**3**	**2.6**	**3**	37	344576	6488	1079	**0.0**	**0.0**	0.0	**0.0**
	70	*3*	3.8	**3**	205.0	2	13.5	**3**	**2.3**	**3**	59	582659	3565	543	**0.0**	1.2	0.0	**0.0**
	80	*3*	3.3	**3**	202.2	2	6.5	**3**	**1.9**	**3**	78	557691	1340	215	**0.0**	1.7	0.0	**0.0**
	90	*2*	7.5	**2**	24.5	**2**	**3.1**	**2**	3.4	**2**	158	57511	578	148	**0.0**	**0.0**	**0.0**	**0.0**
	100	*13*	416.8	**4**	465.4	3	416.0	**4**	**415.8**	**4**	2751	647969	100539	78583	**16.6**	28.4	27.1	36.0
	125	*1*	42.3	**1**	**9.8**	**1**	14.0	**1**	20.9	**1**	1596	2310	2356	695	**0.0**	**0.0**	**0.0**	**0.0**
	200	*5*	tl	**0**	tl	**0**	tl	**0**	tl	**0**	826	281697	38273	51362	53.1	**44.5**	48.2	54.9
	500	*5*	tl	**0**	tl	**0**	tl	**0**	tl	**0**	0	15542	35971	3193	97.2	**77.1**	78.4	89.3
ising	*100*	*9*	**241.6**	6	tl	0	tl	0	293.6	**7**	7605	2137539	2108598	22611	1.3	9.0	8.3	**0.8**
	125	*3*	36.9	**3**	tl	0	tl	0	**34.5**	**3**	16340	2280101	2272325	41888	0.0	21.3	18.3	**0.0**
	150	*6*	tl	**0**	tl	**0**	tl	**0**	tl	**0**	13656	682793	636598	19492	7.8	13.8	13.2	**7.5**
	200	*6*	tl	**0**	tl	**0**	tl	**0**	tl	**0**	14019	426038	377323	14007	**9.8**	15.6	14.4	11.0
	216	*3*	tl	**0**	tl	**0**	tl	**0**	tl	**0**	583415	1291068	1163878	521179	**10.1**	28.9	23.9	12.2
	225	*3*	**12.0**	**3**	tl	0	tl	0	25.2	**3**	17780	1610402	1547126	33277	**0.0**	13.6	12.5	0.0
	250	*6*	tl	**0**	tl	**0**	tl	**0**	tl	**0**	13544	273856	233316	12825	**13.3**	17.6	16.5	13.4
	300	*6*	tl	**0**	tl	**0**	tl	**0**	tl	**0**	10986	186106	152200	11669	**13.8**	17.9	16.5	14.3
	343	*3*	tl	**0**	tl	**0**	tl	**0**	tl	**0**	150820	814477	633599	393406	**19.6**	34.5	26.3	19.9
	400	*3*	tl	**0**	tl	**0**	tl	**0**	tl	**0**	765067	942371	864108	595209	**5.1**	16.6	15.6	6.5
rudy	*60*	*10*	tl	**0**	tl	**0**	tl	**0**	tl	**0**	11811	1079532	501550	358487	**9.9**	18.0	18.7	21.7
	80	*30*	**416.4**	**10**	tl	0	tl	0	422.3	**10**	2563	914335	276030	128149	**30.0**	40.0	36.9	37.3
	100	*90*	**577.5**	**9**	tl	0	tl	0	tl	3	27060	604112	183828	239895	**31.5**	38.6	36.6	42.1

Table 5 Computational comparison of the different formulations for solving the Biq Mac instances to proven optimality

otherwise). The winning formulation is reported in bold text, i.e., lower exit gaps.

We conclude this section stressing the fact that the ELF can be considered as a valid alternative to the standard $GW_{[13]}$, even if in some cases the $GW_{[13]}$ outperforms it. The principal cause is the computational difficulty of the LP-relaxation of ELF, which is slightly bigger and thus a bit slower than the LP-relaxation of $GW_{[13]}$. It is somehow surprising that both the classical $GW_{[13]}$ and the new ELF outperform the other formulations $G_{[11]}$ and $SS_{[22]}$ for Bic Mac instances, in particular as far as the number of instances solved to proven optimality within the time-limit are concerned.

4 Conclusion

In this paper we compared several linearisation techniques for Binary Quadratic Problems present in the literature, called here $GW_{[13]}$, $G_{[11]}$ and $SS_{[22]}$ with a new formulation called ELF. We showed that $G_{[11]}$ and $SS_{[22]}$ are identical and that $GW_{[13]}$ and ELF are identical, i.e., they have the same LP-relaxation values. Among the formulation studied, $GW_{[13]}$ and ELF provide the better performances in practice. Even if both formulations present the same LP-relaxation, their behaviour in practice can be significantly different, as the test on the BicMac instances have demonstrated. Finally, it is also interesting to notice that $SS_{[22]}$ outperforms $G_{[11]}$ in practice, and this is again due to having an LP-relaxation that is easier to solve.

Acknowledgments. We sincerely thank an anonymous referee for thoughtful and motivating feedback, which led to a more meaningful experimental setup and Charlotte Mitchell, for her linguistic assistance.

References

1. Adams, W.P., Forrester, R.J., Glover, F.W.: Comparisons and enhancement strategies for linearizing mixed 0-1 quadratic programs. Discret. Optim. **1**(2), 99–120 (2004)
2. Adams, W.P., Sherali, H.D.: A tight linearization and an algorithm for zero-one quadratic programming problems. Management Science **32**(10), 1274–1290 (1986)
3. BARON: (2012). URL http://archimedes.cheme.cmu.edu/?q=baron
4. Bertsimas, D., Weismantel, R.: Optimization Over Integers. Dynamic Ideas (2005)
5. Bonmin: (2012). URL https://projects.coin-or.org/Bonmin
6. Chaovalitwongse, W., Pardalos, P.M., Prokopyev, O.A.: A new linearization technique for multi-quadratic 01 programming problems. Operations Research Letters **32**(6), 517 – 522 (2004)
7. Conforti, M., Cornuéjols, G., Zambelli, G.: Extended formulations in combinatorial optimization. Annals of Operations Research **204**(1), 97–143 (2013)
8. Cplex: (2012). URL http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/
9. Fortet, R.: Lalgebre de boole et ses applications en recherche operationnelle. Trabajos de Estadistica **11**(2), 111–118 (1960)
10. Furini, F., Traversi, E.: Hybrid SDP Bounding Procedure. Lecture Notes in Computer Science **7933**, 248–259 (2013)
11. Glover, F.: Improved linear integer programming formulations of nonlinear integer programs. Management Science **22**(4), 455–460 (1975)
12. Glover, F., Woolsey, E.: Further reduction of zero-one polynomial programming problems to zero-one linear programming problems. Operations Research **21**(1), 156–161 (1973)
13. Glover, F., Woolsey, E.: Converting the 0-1 polynomial programming problem to a 0-1 linear program. Operations Research **22**(1), 180–182 (1974)
14. Gueye, S., Michelon, P.: A linearization framework for unconstrained quadratic (0-1) problems. Discrete Applied Mathematics **157**(6), 1255 – 1266 (2009)
15. Hansen, P., Meyer, C.: Improved compact linearizations for the unconstrained quadratic 01 minimization problem. Discrete Applied Mathematics **157**(6), 1267 – 1290 (2009)
16. Jaumard, B., Marcotte, O., Meyer, C.: Estimation of the Quality of Cellular Networks Using Column Generation Techniques. Cahiers du GÉRAD. Groupe d'études et de recherche en analyse des décisions (1998)
17. Krislock, N., Malick, J., Roupin, F.: Improved semidefinite bounding procedure for solving max-cut problems to optimality. Mathematical Programming pp. 1–26 (2012)

18. Lodi, A.: Mixed integer programming computation. In: M. Jünger, T.M. Liebling, D. Naddef, G.L. Nemhauser, W.R. Pulleyblank, G. Reinelt, G. Rinaldi, L.A. Wolsey (eds.) 50 Years of Integer Programming 1958-2008, pp. 619–645. Springer Berlin Heidelberg (2010)
19. Rendl, F., Rinaldi, G., Wiegele, A.: Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. Mathematical Programming **121**(2), 307–335 (2010)
20. SCIP: (2012). URL http://scip.zib.de/
21. Sherali, H.D., Adams, W.P.: A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems. Springer (1998)
22. Sherali, H.D., Smith, J.C.: An improved linearization strategy for zero-one quadratic programming problems. Optimization Letters **1**(1), 33–47 (2007)
23. Wang, H., Kochenberger, G., Glover, F.: A computational study on the quadratic knapsack problem with multiple constraints. Computers & Operations Research **39**(1), 3 – 11 (2012)
24. Wiegele, A.: Biq mac library, a collection of max-cut and quadratic 01 programming instances of medium size. Technical report, Alpen-Adria-Universitt Klagenfurt, Austria (2007)