4OR-Q J Oper Res
https://doi.org/10.1007/s10288-017-0364-0

RESEARCH PAPER

Compact linearization for binary quadratic problems subject to assignment constraints

Sven Mallach[1]

Received: 18 November 2016 / Revised: 13 November 2017

Abstract We introduce and prove new necessary and sufficient conditions to carry out a compact linearization approach for a general class of binary quadratic problems subject to assignment constraints that has been proposed by Liberti (4OR 5(3):231–245, 2007, https://doi.org/10.1007/s10288-006-0015-3). The new conditions resolve inconsistencies that can occur when the original method is used. We also present a mixed-integer linear program to compute a minimally sized linearization. When all the assignment constraints have non-overlapping variable support, this program is shown to have a totally unimodular constraint matrix. Finally, we give a polynomial-time combinatorial algorithm that is exact in this case and can be used as a heuristic otherwise.

Keywords Non-linear programming · Binary quadratic programming · Mixed-integer programming · Linearization

Mathematics Subject Classification 68R01 · 90C05 · 90C09 · 90C10 · 90C11 · 90C20 · 90C30

1 Introduction

In this paper, we are concerned with binary quadratic programs (BQPs) that comprise some assignment constraints over a set of $\{0, 1\}$-variables x_i, $i \in N$, where $N = \{1, \ldots, n\}$ and $n \in \mathbb{N}^+$. More precisely, let the assignment constraints be given as a collection K of index sets $A_k \subseteq N$, $k \in K$, such that exactly one of the variables

✉ Sven Mallach
mallach@informatik.uni-koeln.de

1 Institut für Informatik, Universität zu Köln, Albertus-Magnus-Platz, 50923 Cologne, Germany

Published online: 02 December 2017

x_i, $i \in A_k$, is required to attain the value 1. Bilinear terms $y_{ij} = x_i x_j$, $i, j \in N$, are permitted to occur in the objective function as well as in the set of constraints and are assumed to be collected in an ordered set $E \subset N \times N$. By commutativity, there is no loss of generality in requiring that $i \leq j$ for each $(i, j) \in E$. We further assume that for each $(i, j) \in E$ there exist indices k, l such that $i \in A_k$ and $j \in A_l$, i.e., for each variable being part of a bilinear term there is some assignment constraint involving it. With an arbitrary set of $m \geq 0$ linear constraints $Cx + Dy \geq b$ where $C \in \mathbb{R}^{m \times n}$ and $D \in \mathbb{R}^{m \times |E|}$, a general form of the mixed-integer programs considered can be stated as:

$$\begin{aligned}
\min\ & c^T x + d^T y \\
\text{s.t.}\ & \sum_{i \in A_k} x_i = 1 && \text{for all } k \in K && (1) \\
& Cx + Dy \geq b \\
& y_{ij} = x_i x_j && \text{for all } (i, j) \in E && (2) \\
& x_i \in \{0, 1\} && \text{for all } i \in
\end{aligned}$$

This general form covers for example the quadratic assignment problem which is known to be NP-hard (Sahni and Gonzalez 1976), as are BQPs with box constraints in general (Sahni 1974), with some exceptions discussed by Allemand et al. (2001). While there exist approaches to tackle BQPs directly, linearizations of quadratic and, more generally, polynomial programming problems, enable the application of well-studied mixed-integer linear programming techniques and have hence been an active field of research since the 1960s. In this paper, we concentrate on the question how to realize constraints (2) for this particular type of problem by means of additional variables and additional linear constraints.

The seminal idea to model binary conjunctions using additional (binary) variables is attributed to Fortet (1959, 1960) and addressed by Hammer and Rudeanu (1968). This method, that is also proposed in succeeding works by Balas (1964), Zangwill (1965) and Watters (1967), and further discussed by Glover and Woolsey (1973), requires two inequalities per linearization variable. Only shortly thereafter, Glover and Woolsey (1974) found that the same effect can be achieved using *continuous* linearization variables when replacing one of these inequalities with two different ones. The outcome is a method that is until today regarded as 'the standard linearization technique' and where, in the binary quadratic case, each product $x_i x_j$ is modeled using a variable $y_{ij} \in [0, 1]$ and three constraints:

$$y_{ij} \leq x_i \tag{3}$$

$$y_{ij} \leq x_j \tag{4}$$

$$y_{ij} \geq x_i + x_j - 1 \tag{5}$$

Succeeding developments include a linearization technique without any additional variables but using a family of (exponentially many) inequalities by Balas and Mazzola (1984). Adams and Sherali (1986) showed how the introduction and subsequent linearization of additional non-linear constraints can be used to obtain tighter linear programming (LP) relaxations for binary problems with (initially) linear constraints and a quadratic objective function. This approach was later generalized in Adams and Sherali (1999) to the so-called reformulation-linearization-technique (RLT). A single application of the RLT to the bounds constraints $0 \leq x_i \leq 1$ of a binary program leads exactly to the above 'standard linearization'.

Further linearization methods with more emphasis on problems where all non-linearities appear only in the objective function are by Glover (1975), Oral and Kettani (1992a, b), Chaovalitwongse et al. (2004), Sherali and Smith (2007), Furini and Traversi (2013), and, for general integer variables, by Billionnet et al. (2008). Specialized formulations for unconstrained binary quadratic programming problems have been given by Gueye and Michelon (2009), and Hansen and Meyer (2009).

For the particular BQP introduced at the beginning, Liberti (2007) developed a *compact* linearization approach that exploits the structure imposed by the assignment constraints very elegantly. It can be seen as a special application of the RLT. First, it determines for each set A_k, $k \in K$, another set B_k indexing original variables. These are then multiplied with the according assignment constraint related to A_k which yields new additional equations and induces a set of bilinear terms F. The sets B_k need to be chosen such that F covers the original set of products E. Finally, the products in F are replaced by linearization variables. As already noted by Liberti (2007), conceptually, this approach is in line with and a generalization of the one applied by Frieze and Yadegar (1983) to the quadratic assignment problem.

In this paper, we show that one of the conditions originally specified as being necessary for the sets B_k to hold in order to yield a correct compact linearization is in fact neither necessary nor sufficient. As a consequence, when applying the original method to determine the sets B_k, inconsistent value assignments to the set of created variables can result. We reveal two new necessary and sufficient conditions to obtain a consistent linearization and prove their correctness. As a positive side effect, these conditions can lead to smaller sets B_k and hence to a smaller number of linearization variables and constraints. In Liberti (2007), also an integer program to compute sets B_k leading to a minimum number of additional equations has been given. We present a similar mixed-integer linear program that establishes the new conditions and can be used to minimize the number of created additional variables and equations. Moreover, we show that the constraint matrix of this program is totally unimodular if all the sets A_k, $k \in K$, are pairwise disjoint. In addition, we provide an exact combinatorial and polynomial-time algorithm to compute optimal sets B_k in this case. With small modifications, the algorithm can also be used as a heuristic for the more general setting with overlapping sets A_k.

The article is organized as follows: In Sect. 2, we review the compact linearization approach as developed in Liberti (2007) and show that consistency of the linearization variables with their associated original variables is not implied by the conditions specified. The new conditions for a consistent linearization are characterized in Sect. 3 together with a correctness proof. Further, a revised mixed-integer linear program and a new combinatorial algorithm to compute compact linearizations are given. In Sect. 4, we repeat the computational experiments carried out in Liberti (2007) for the graph partitioning problem with the resulting new linearized problem formulation. We close this paper with a conclusion and final remarks in Sect. 5.

2 Compact linearization for binary quadratic problems

The compact linearization approach for binary quadratic problems with assignment constraints by Liberti (2007) is as follows: With each index set A_k, we associate a corresponding index set B_k such that for each $j \in B_k$ the assignment constraint (1) w.r.t. A_k is multiplied with x_j. We thus obtain the equations:

$$\sum_{i \in A_k} x_i x_j = x_j \quad \text{for all } j \in B_k, \text{ for all } k \in K \tag{6}$$

Each product $x_i x_j$ induced by any of the Eqs. (6) is then replaced by a continuous linearization variable y_{ij} (if $i \leq j$) or y_{ji} (otherwise). We denote the set of bilinear terms created this way with F and we may again assume without loss of generality that $i \leq j$ holds for each $(i, j) \in F$. More formally, Liberti (2007) defined the set F as

$$F = \left\{ \phi(i, j) \mid (i, j) \in \bigcup_{k \in K} A_k \times B_k \right\}$$

where $\phi(i, j) = (i, j)$ if $i \leq j$ and $\phi(i, j) = (j, i)$ otherwise. By rewriting Eqs. (6) using F, we obtain the linearization equations:

$$\sum_{i \in A_k, (i,j) \in F} y_{ij} + \sum_{i \in A_k, (j,i) \in F} y_{ji} = x_j \qquad \text{for all } j \in B_k, \text{ for all } k \in K \tag{7}$$

The choice of the sets B_k is crucial for the correctness and the size of the resulting linearized problem formulation. It directly determines the cardinality of the set F as well as the number of additional equations. Satisfying $F \supseteq E$, possibly (and in practice almost surely) involves creating additional linearization variables for some $i \in A_k$ and $j \in B_k$ where neither $(i, j) \in E$ nor $(j, i) \in E$. On the other hand, as is discussed by Liberti (2007) and called *constraint-side compactness*, the number of equations can be considerably smaller than $3|E|$ as it would be with the 'standard' approach. Still, the resulting formulation will be at least as strong in terms of the tightness of the LP relaxation. This property can, e.g., be shown by arguing that solutions obeying all the Eqs. (7) also satisfy the inequalities (3), (4), and (5) for all the variables introduced based on F. Since the added equations are all equations of the original problem multiplied by original variables, this also proves correctness of the linearization—no solutions feasible for the original problem can be excluded like this. Liberti followed exactly this strategy. In his proof however, a choice of the sets B_k satisfying simultaneously the conditions $E \subseteq F$ and $A_k \subseteq B_k$ for all $k \in K$ was expected to be sufficient in order to achieve a consistent linearization which is not the case in general.

To see this, let $k \in K$ be such that $A_k \subsetneq B_k$. We can assume without loss of generality that such a k exists since otherwise $A_k = B_k$ must hold for all $k \in K$. If this was the case, however, generating $(i, j) \in F$ for $(i, j) \in E$ by picking some index l where $i \in$

A_l and $j \in B_l$ (or vice versa) implies $i, j \in A_l$. Consequently, all bilinear terms could be resolved trivially as $y_{ij} = 0$ for all $(i, j) \in E, i \neq j$, and $y_{ii} = x_i$ for all $(i, i) \in E$. So let now $j \in B_k \setminus A_k$. In our linearization system, we obtain thus an equation:

$$\sum_{a \in A_k, (a,j) \in F} y_{aj} + \sum_{a \in A_k, (j,a) \in F} y_{ja} = x_j \tag{8}$$

Now fix an arbitrary $i = a$, $a \in A_k$, and assume, without loss of generality, that $i < j$ ($i = j$ is impossible since $j \notin A_k$). Hence $(i, j) \in F$ and Eq. (8) clearly establishes $y_{ij} \leq x_j$. The condition $A_k \subseteq B_k$ implies $i \in B_k$ and hence that there must be another equation for i:

$$\sum_{a \in A_k, (a,i) \in F} y_{ai} + \sum_{a \in A_k, (i,a) \in F} y_{ia} = x_i \tag{9}$$

However, since $j \notin A_k$, the variable y_{ij} does not appear on the left hand side of (9), meaning that $y_{ij} \leq x_i$ is not enforced—and it will not be unless there is another $l \in K$, $l \neq k$, such that $i \in B_l$ and $j \in A_l$ which is not assured by the conditions originally specified. The opposite case where, for some $(i, j) \in F$, $j \in A_k$ and $i \in B_k \setminus A_k$ but there is no $l \in K$, $l \neq k$, such that $j \in B_l$ and $i \in A_l$, leads to the converse problem that there are equations that enforce $y_{ij} \leq x_i$ but none that enforce $y_{ij} \leq x_j$. Moreover, the absence of any of the two required equations may also cause violations of the inequality $y_{ij} \geq x_i + x_j - 1$ as the value assignment $x_i = x_j = 1$ will then not be correctly imposed on the variable y_{ij}.

Based on these observations, one can easily construct a small example where there exist $(i, j) \in F$ for which inconsistent value assignments to x_i, x_j and y_{ij} result.

Example Consider the following BQP subject to assignment constraints.

$$\begin{aligned}
\min\ & y_{13} + y_{14} + y_{23} + y_{24} & & \\
\text{s.t.}\ & x_1 + x_2 & = & \ 1 \\
& x_3 + x_4 & = & \ 1 \\
& x_1 x_3 & = & \ y_{13} \\
& x_1 x_4 & = & \ y_{14} \\
& x_2 x_3 & = & \ y_{23} \\
& x_2 x_4 & = & \ y_{24} \\
& x_1, x_2, x_3, x_4 & \in & \ \{0, 1\} \\
& y_{13}, y_{14}, y_{23}, y_{24} & \geq & \ 0
\end{aligned}$$

We have $A_1 = \{1, 2\}$, $A_2 = \{3, 4\}$ and $E = \{(1, 3), (1, 4), (2, 3), (2, 4)\}$. Choosing an optimum solution of the originally proposed integer program to compute the sets B_k, we set $B_1 = \{1, 2, 3, 4\} \supset A_1$ and $B_2 = \{3, 4\} = A_2$, obtaining

$$F = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4), (3, 3), (3, 4), (4, 4)\} \supset E.$$

One can easily verify that the resulting program according to B_1 and B_2 is:

$$
\begin{array}{lll}
\min & y_{13} + y_{14} + y_{23} + y_{24} & \\
\text{s.t.} & x_1 + x_2 & = 1 \\
& x_3 + x_4 & = 1 \\
& y_{11} + y_{12} & = x_1 \\
& y_{12} + y_{22} & = x_2 \\
& y_{13} + y_{23} & = x_3 \\
& y_{14} + y_{24} & = x_4 \\
& y_{33} + y_{34} & = x_3 \\
& y_{34} + y_{44} & = x_4 \\
& y_{11}, y_{12}, y_{13}, y_{14}, y_{23}, y_{24}, y_{33}, y_{34}, y_{44} & \geq 0
\end{array}
$$

Due to the assignment constraints, one of x_1 and x_2 and one of x_3 and x_4 must be set to 1. So in any correct solution, exactly one of the y-variables in the objective function will be equal to 1. Choose e.g. $x_1 = x_3 = 1$. After substituting for the right hand sides accordingly, a choice remains in the third linearization equation to spread the '1' across y_{13} and y_{23} while only $y_{13} = 1$ and $y_{23} = 0$ would be consistent. As one can see as well, this problem occurs due to a missing equation that imposes $y_{23} \leq x_2$. Further equations are missing that impose $y_{13} \leq x_1$, $y_{14} \leq x_1$ and $y_{24} \leq x_2$, so analogous issues result when choosing the values of the x-variables differently. Moreover, this example problem can be consistently linearized by choosing $B_1 = \{3, 4\}$ and $B_2 = \{1, 2\}$, i.e., $A_k \nsubseteq B_k$ for both $k = 1, 2$. This shows that $A_k \subseteq B_k$ is neither a *necessary* nor a *sufficient* condition.

3 Revised compact linearization

The previous section revealed that the condition $A_k \subseteq B_k$ for all $k \in K$ is neither necessary nor sufficient to enforce that the inequalities $y_{ij} \leq x_i$ and $y_{ij} \leq x_j$ are satisfied for all $(i, j) \in F$. The discussion also indicated that the two following conditions are *necessary* to enforce this.

Condition 1 *For each $(i, j) \in F$, there is a $k \in K$ such that $i \in A_k$ and $j \in B_k$.*

Condition 2 *For each $(i, j) \in F$, there is an $l \in K$ such that $j \in A_l$ and $i \in B_l$.*

For these two conditions, clearly, $k = l$ is a valid choice. In this section, we will prove that these conditions are also *sufficient* in order to yield a correct linearization.

Theorem 3 *Let $(i, j) \in F$. If Conditions 1 and 2 are satisfied, then it holds that $y_{ij} \leq x_i$, $y_{ij} \leq x_j$ and $y_{ij} \geq x_i + x_j - 1$.*

Proof By Condition 1, there is a $k \in K$ such that $i \in A_k$, $j \in B_k$ and hence the equation

$$\sum_{a \in A_k, (a,j) \in F} y_{aj} + \sum_{a \in A_k, (j,a) \in F} y_{ja} = x_j \tag{$*$}$$

exists and has y_{ij} on its left hand side. This establishes $y_{ij} \leq x_j$.

Similarly, by Condition 2, there is an $l \in K$ such that $j \in A_l$, $i \in B_l$ and hence the equation

$$\sum_{a \in A_l, (a,i) \in F} y_{ai} + \sum_{a \in A_l, (i,a) \in F} y_{ia} = x_i \tag{$**$}$$

exists and has y_{ij} on its left hand side. This establishes $y_{ij} \leq x_i$.

As a consequence, $y_{ij} = 0$ whenever $x_i = 0$ or $x_j = 0$. In this case, the inequality $y_{ij} \geq x_i + x_j - 1$ is trivially satisfied. Now let $x_i = x_j = 1$. Then the right hand sides of both ($*$) and ($**$) are equal to 1. The variable y_{ij} (is the only one that) occurs on the left hand sides of both of these equations. If $y_{ij} = 1$, this is consistent and correct. So suppose that $y_{ij} < 1$ which implies that, in Eq. ($*$), there is some y_{aj} (or y_{ja}), $a \neq i$, with $y_{aj} > 0$ ($y_{ja} > 0$). Then, by the previous arguments and integrality of the x-variables, $x_a = 1$. This is however a contradiction to the assumption that $x_i = 1$ as both i and a are contained in A_k. □

A minimum number of additional equations, $\sum_{k \in K} |B_k|$, and variables, $|F|$, can be obtained using a mixed-integer program. The following one is a modification of the integer program by Liberti (2007) in order to implement Conditions 1 and 2, but even more importantly, to enforce them not only for $(i, j) \in E$ but for $(i, j) \in F$.

$$\min\ w_{eqn} \left(\sum_{k \in K} \sum_{1 \leq i \leq n} z_{ik} \right) + w_{var} \left(\sum_{1 \leq i \leq n} \sum_{i \leq j \leq n} f_{ij} \right)$$

$$\begin{aligned}
\text{s.t.}\quad & f_{ij} = 1 && \text{for all } (i, j) \in E && (10)\\
& f_{ij} \geq z_{jk} && \text{for all } k \in K, i \in A_k, j \in N, i \leq j && (11)\\
& f_{ji} \geq z_{jk} && \text{for all } k \in K, i \in A_k, j \in N, j < i && (12)\\
& \sum_{k: i \in A_k} z_{jk} \geq f_{ij} && \text{for all } 1 \leq i \leq j \leq n && (13)\\
& \sum_{k: j \in A_k} z_{ik} \geq f_{ij} && \text{for all } 1 \leq i \leq j \leq n && (14)\\
& f_{ij} \in [0, 1] && \text{for all } 1 \leq i \leq j \leq n &&\\
& z_{ik} \in \{0, 1\} && \text{for all } k \in K, 1 \leq i \leq n &&
\end{aligned}$$

The formulation involves binary variables z_{ik} to be equal to 1 if $i \in B_k$ and equal to zero otherwise. Further, to account for whether $(i, j) \in F$, there is a (continuous) variable f_{ij} for all $1 \leq i \leq j \leq n$ that will be equal to 1 in this case and 0 otherwise.

The constraints (10) fix those f_{ij} to 1 where the corresponding pair (i, j) is contained in E. Whenever some $j \in N$ is assigned to some set B_k, then we need the corresponding variables $(i, j) \in F$ or $(j, i) \in F$ for all $i \in A_k$ which is established by (11) and (12). Finally, if $(i, j) \in F$, then we require Conditions 1 and 2 to be satisfied, namely that there is a $k \in K$ such that $i \in A_k$ and $j \in B_k$ (13) and a (possibly different) $k \in K$ such that $j \in A_k$ and $i \in B_k$ (14). These constraints impose a certain minimum on the sum of cardinalities $\sum_{k \in K} |B_k|$ and hence on the number of equations. Different solutions achieving this minimum may lead to different cardinalities of $|F|$. In general, it is even possible that a minimum $|F|$ can only be achieved by adding more than a minimum number of equations. But, in practice, this is seldom the case as adding equations rather induces more variables. The weights w_{var} and w_{eqn} of the objective function allow to find the best compromise. A rational choice would be to set $w_{var} = 1$ and $w_{eqn} > \max_{k \in K} |A_k|$. This results in a solution with a minimum number of additional equations that, among these, also induces a minimal number of additional variables.

The program models a special two-stage covering problem whose complexity depends on how the sets A_k interrelate. In particular, if $A_k \cap A_l = \emptyset$ for all $k, l \in K,\ k \neq l$, then the decisions to be made in constraints (13) and (14) are unique and the problem can be solved as a linear program because the constraint matrix arising in this case is totally unimodular (TU). For a proof, we employ the following lemma [cf. Nemhauser and Wolsey (1988)]:

Lemma 1 *If the* $\{-1, 0, 1\}$*-matrix* H *has no more than two non-zero entries in each row and if* $\sum_j h_{ij} = 0$ *whenever row* i *has two non-zero coefficients, then* H *is TU.*

Theorem 4 *If* $A_k \cap A_l = \emptyset$*, for all* $k, l \in K$*,* $k \neq l$*, then the constraint matrix of the above mixed-integer program is TU.*

Proof We interpret the constraint set (11)–(14) as the rows of a matrix $H = [F\ Z]$ where F is the upper triangular matrix defined by $\{f_{ij} \mid 1 \leq i \leq j \leq n\}$ and $Z = (z_{ik})$ for $i \in N$ and $k \in K$. Constraints (11) and (12) cause exactly one 1-entry in F, and exactly one -1-entry in Z. Conversely, constraints (13) and (14) yield exactly one -1-entry in F, and – since $|\{k : i \in A_k\}| = 1$ for all $i \in N$ – exactly one 1-entry in Z. Hence, each row i of H has exactly two non-zero entries and $\sum_j h_{ij} = 0$. Finally, constraints (10) correspond to rows with a single non-zero entry or can equivalently be interpreted as removing columns from H causing some other rows to have less than two non-zero entries. Thus, by Lemma 1, H is TU. □

Under these conditions, an exact solution can also be obtained using a combinatorial algorithm (listed as Algorithm 1) where we assume that the unique index k such that $i \in A_k$ is given as $K(i)$. An initial set $F_1 = E$ then requires some variable indices i to be assigned to some unique sets B_k. This may lead to new variables yielding a set $F_2 \supseteq F_1$ which in turn possibly requires further unique extensions of the sets B_k and so on until a steady state is reached. The asymptotic running time of Algorithm 1 can be bounded by $O(n^3)$.

For the general setting with overlapping A_k-sets and hence $|\{k : i \in A_k\}| \geq 1$ for $i \in N$, the above proof of Theorem 4 fails and, indeed, one can construct small instances where the corresponding linear programs have non-integral optima. Nonetheless, the combinatorial algorithm can still be used when equipped with a subroutine to determine the indices k^* and l^*. One might consider the following heuristic idea: To ease notation, let $a(i,k) = 1$ if $i \in A_k$ and $a(i,k) = 0$ otherwise. Similarly, let $b(i,k) = 1$ if $i \in B_k$ and $b(i,k) = 0$ otherwise. Adding some i to some B_k for the first time potentially induces additional variables (u,i) or $(i,u) \in F$ for $u \in A_k$. More precisely, whether such a variable must be newly created depends on whether or not there already is some $l \in K$ where $u \in A_l$ and $i \in B_l$ or $i \in A_l$ and $u \in B_l$. Hence, the number of necessarily created variables when adding i to B_k is $c(i,k) = \sum_{u \in A_k} (\min_{l \in K} 1 - \max\{a(u,l)b(i,l), a(i,l)b(u,l)\})$. So for each $(i,j) \in F$, a locally best choice is to select $k^* = \operatorname{argmin}_{k: i \in A_k} c(j,k)$ and then $l^* = \operatorname{argmin}_{k: j \in A_k} c(i,k)$. This, however, neither incorporates the interdependences with any other choices nor the consequences of the possibly induced pairs (a,j) (or (j,a)) for $a \in A_k^*$ and (a,i) (or (i,a)) for $a \in A_l^*$.

4 Computational experiments

We repeat the computational experiments for the graph partitioning problem (GPP) as carried out in Liberti (2007) since the new conditions derived in this paper lead to a different linearization of the associated integer programming formulation.

Given an undirected graph $G = (V,H)$ and an integer $m \geq 2$, the GPP asks for a partition of V into m subsets such that the number of edges whose endpoints are assigned to different partitions is minimum. A corresponding BQP with variables x_{vk} equal to 1 if vertex $v \in V$ is assigned to subset $k \in \{1,\ldots,m\}$ and equal to 0 otherwise reads:

Algorithm 1 A combinatorial algorithm to construct F and the sets B_k for all $k \in K$.

```
function ConstructSets(Sets E, K and A_k for all k ∈ K)
    for all k ∈ K do
        B_k ← ∅
    F ← E
    F_new ← E
    while ∅ ≠ F_add ← Append(F, F_new, K, A_k, B_k) do
        F ← F ∪ F_add
        F_new ← F_add
    return F and B_k for all k ∈ K
function Append(Sets F, F_new, K, A_k and B_k for all k ∈ K)
    F_add ← ∅
    for all (i, j) ∈ F_new do
        k* ← K(i)
        l* ← K(j)
        if j ∉ B*_k then
            B*_k ← B*_k ∪ {j}
            for all a ∈ A*_k do
                if a ≤ j and (a, j) ∉ F then F_add ← F_add ∪ {(a, j)}
                else if a > j and (j, a) ∉ F then F_add ← F_add ∪ {(j, a)}
        if i ∉ B*_l then
            B*_l ← B*_l ∪ {i}
            for all a ∈ A*_l do
                if a ≤ i and (a, i) ∉ F then F_add ← F_add ∪ {(a, i)}
                else if a > i and (i, a) ∉ F then F_add ← F_add ∪ {(i, a)}
    return F_add
```

$$\begin{array}{llll} \min & \sum_{(v,w)\in H} \sum_{1\le k\neq l\le m} x_{vk}x_{wl} & & \\ \text{s.t.} & \sum_{1\le k\le m} x_{vk} & = 1 & \text{for all } v \in V \\ & \sum_{v\in V} x_{vk} & \ge 1 & \text{for all } 1 \le k \le m \\ & x_{vk} & \in \{0,1\} & \text{for all } v \in V,\ 1 \le k \le m \end{array}$$

For this problem, the compact linearization is unique since the assignment constraints have non-overlapping variable support. While the 'standard linearization' method would introduce exactly the $m(m-1)|H|$ variables $y_{vw}^{kl} = x_{vk}v_{wl}$ necessary to reformulate the objective function, the compact approach will induce also the products for $k = l$ and hence $m^2|H|$ variables. On the constraint side, we have $3m(m-1)|H|$ inequalities in the first and only $2m|H|$ equations in the latter case. The equations are of the form:

$$\begin{array}{ll} \sum_{1\le k\le m} y_{vw}^{kl} = x_{wl} & \text{for all } (v,w) \in H,\ 1 \le l \le m \\ \sum_{1\le l\le m} y_{vw}^{kl} = x_{vk} & \text{for all } (v,w) \in H,\ 1 \le k \le m \end{array}$$

As in Liberti (2007), we compare the compactly linearized program to the formulations 'B2' (for $m = 2$) and 'B' (for $m = 3, 4$) proposed by Boulle (2004), and additionally to the model obtained with the 'standard' linearization. Like in the original article, we also add the following maximum cardinality constraints to each of the problem formulations.

$$\sum_{v\in V} x_{vk} \le \left\lfloor \frac{|V|}{m} + \left(1 - \frac{1}{m}\right) \right\rfloor \qquad \text{for all } 1 \le k \le m$$

We implemented and solved all the models using version 7 of Gurobi.[1] In order to reduce side effects, we configured it to use only a single thread and disabled all internal heuristics. Each experiment was executed on a Debian Linux machine with an Intel Core i7-3770T processor running at 2.5 GHz and with 32 GB RAM. The results are listed in Table 1. Columns 'BB' show the total numbers of branch-and-bound nodes explored and columns 'MIP' display the used CPU and system time in seconds. In addition, columns 'RR' depict the time needed to solve the linear programming relaxation at the root node. We specified a time limit of one hour. In case of a timeout (marked with 't/o'), columns 'LB' and 'UB' reflect the dual bound (rounded to two decimal digits) and the objective value of the best feasible solution sustained when the timeout occurred, otherwise they show the optimum objective value. Sometimes, the optima for `Breg` instances differ from those in Liberti (2007) which is because

[1] http://www.gurobi.com/.

Table 1 Computational results for the graph partitioning problem with $m = 2, 3,$ and 4

Instance	|V|	|H|	'Compact'					'B2'					'Standard'				
			LB	UB	BB	MIP	RR	LB	UB	BB	MIP	RR	LB	UB	BB	MIP	RR
m = 2																	
Breg100.04	100	150	4	4	15	0.05	0.00	4	4	27	0.11	0.00	4	4	16	0.18	0.00
Breg100.08	100	150	8	8	317	0.21	0.00	8	8	1484	1.10	0.00	8	8	274	0.47	0.00
Breg100.20	100	150	16	16	15,923	11.75	0.00	16	16	13,798	10.91	0.00	16	16	6131	7.10	0.00
Breg500.12	500	750	12	12	1375	6.26	0.01	12	12	6288	32.94	0.02	12	12	3191	45.48	0.04
Breg500.16	500	750	16	16	2687	19.28	0.01	16	16	6058	52.96	0.02	16	16	1445	26.69	0.04
Breg500.20	500	750	20	20	7742	108.04	0.02	20	20	24,453	387.94	0.02	20	20	8557	124.69	0.04
Cat.0352	352	351	1	1	0	0.10	0.01	1	1	0	1.42	0.01	1	1	0	3.64	0.01
Cat.0702	702	701	1	1	0	0.19	0.05	1	1	0	1.19	0.02	1	1	0	1.48	0.05
Cat.1052	1052	1051	1	1	0	0.32	0.10	1	1	0	1.53	0.06	1	1	0	1.88	0.09
Grid.100	100	180	10	10	77	0.13	0.00	10	10	314	0.40	0.00	10	10	61	0.31	0.00
Grid.900	900	1740	30	30	17,252	1613.14	0.03	30	30	29,816	2547.05	0.06	24.22	30	20,884	t/o	0.13
RCat.134	134	133	1	1	0	0.02	0.00	1	1	0	0.09	0.00	1	1	0	0.12	0.00
RCat.554	554	553	1	1	0	0.10	0.02	1	1	25	1.76	0.01	1	1	0	0.60	0.03
WGrid.100	100	200	20	20	1080	1.08	0.00	20	20	1226	1.23	0.00	20	20	1479	2.96	0.00
G124.02	124	149	13	13	582	0.40	0.00	13	13	811	0.58	0.00	13	13	1272	1.51	0.01
G124.04	124	318	63	63	193,553	687.31	0.00	53.00	63	1,257,935	t/o	0.00	63	63	13,4129	1036.46	0.01
G124.08	124	620	101.47	184	320,277	t/o	0.01	97.69	185	359,437	t/o	0.00	88.08	183	147,182	t/o	0.01
G250.01	250	331	29	29	20,307	75.83	0.01	29	29	35,378	72.84	0.01	29	29	39,552	202.61	0.01
G500.005	500	625	42.55	50	676,416	t/o	0.02	36.23	51	788,510	t/o	0.01	34.14	51	490,188	t/o	0.03

Table 1 continued

Instance	|V|	|H|	'Compact'					'B'					'Standard'				
			LB	UB	BB	MIP	RR	LB	UB	BB	MIP	RR	LB	UB	BB	MIP	RR
m = 3																	
Breg100.04	100	150	17	17	40,580	146.08	0.01	17	17	9609	39.07	0.01	12.37	18	239,655	t/o	0.01
Breg100.08	100	150	13	13	2817	12.17	0.01	13	13	1507	9.13	0.01	13	13	16,343	279.56	0.01
Breg100.20	100	150	21	21	401,463	1280.07	0.01	21	21	43,253	175.72	0.01	18.64	22	23,6245	t/o	0.01
Breg500.12	500	750	16.18	70	44710	t/o	0.08	13.65	80	37,299	t/o	0.17	6.41	97	12,198	t/o	0.09
Breg500.16	500	750	16.17	70	39,442	t/o	0.09	13.73	81	40,530	t/o	0.17	5.67	101	11,171	t/o	0.11
Breg500.20	500	750	15.77	74	39,417	t/o	0.09	12.96	80	30,139	t/o	0.17	6.43	102	10,321	t/o	0.13
Cat.0352	352	351	5	5	218,473	2077.14	0.04	5	5	10,212	92.00	0.05	5	5	7370	465.39	0.04
Cat.0702	702	701	3.05	6	97,775	t/o	0.10	3.57	6	200,001	t/o	0.20	1.00	9	8054	t/o	0.07
Cat.1052	1052	1051	2	2	449	9.12	0.15	2	2	744	519.08	0.47	0.44	3	2359	t/o	0.14
Grid.100	100	180	18	18	2124	13.66	0.01	18	18	1640	14.52	0.01	18	18	8105	138.50	0.03
Grid.900	900	1740	18.61	73	12,401	t/o	0.21	15.63	87	9164	t/o	0.27	6.00	93	8393	t/o	0.21
RCat.134	134	133	2	2	15	0.24	0.01	2	2	123	0.80	0.01	2	2	52	1.00	0.01
RCat.554	554	553	2	2	36	1.54	0.13	2	2	220	13.40	0.09	2	2	72	47.04	0.05
WGrid.100	100	200	34	34	319,671	1948.04	0.01	34	34	157,082	1129.75	0.01	25.36	34	120,681	t/o	0.02
G124.02	124	149	18	18	5148	21.19	0.02	18	18	26,914	112.94	0.02	18	18	37,721	448.19	0.02
G124.04	124	318	45.73	93	236,267	t/o	0.02	49.25	92	110,943	t/o	0.02	37.33	95	134,661	t/o	0.04
G124.08	124	620	75.88	272	74,548	t/o	0.03	75.03	268	56,648	t/o	0.04	56.96	277	43,314	t/o	0.03
G250.01	250	331	32.94	41	223,223	t/o	0.03	31.18	43	104,792	t/o	0.05	19.62	45	53,279	t/o	0.03
G500.005	500	625	26.15	79	49,255	t/o	0.11	20.51	94	49,306	t/o	0.15	16.00	85	19,053	t/o	0.11

Table 1 continued

Instance	|V|	|H|	'Compact'					'B'					'Standard'				
			LB	UB	BB	MIP	RR	LB	UB	BB	MIP	RR	LB	UB	BB	MIP	RR
m = 4																	
Breg100.04	100	150	18	18	15,674	113.76	0.03	18	18	14,768	165.11	0.03	15.01	20	124,989	t/o	0.05
Breg100.08	100	150	18	18	36,371	281.38	0.01	18	18	75,587	629.06	0.02	14.21	20	125,999	t/o	0.02
Breg100.20	100	150	21.43	27	501,855	t/o	0.01	18.29	29	219,960	t/o	0.03	17.98	28	108,461	t/o	0.02
Breg500.12	500	750	12.6	90	17,227	t/o	0.24	6.92	140	11,126	t/o	0.52	4.72	152	13,445	t/o	0.21
Breg500.16	500	750	12.76	96	14,416	t/o	0.24	8.27	138	12,109	t/o	0.36	3.25	167	13,546	t/o	0.24
Breg500.20	500	750	12.00	116	13,875	t/o	0.22	7.37	135	10,255	t/o	0.43	4.10	165	15,517	t/o	0.31
Cat.0352	352	351	4.02	10	39,444	t/o	0.09	4.17	10	161,837	t/o	0.09	3.94	9	55,721	t/o	0.05
Cat.0702	702	701	3	3	94	63.74	0.20	3	3	732	123.60	0.36	0.43	9	10,270	t/o	0.33
Cat.1052	1052	1051	2.53	15	1417	t/o	0.35	3.11	12	18,681	t/o	0.91	0.00	20	4508	t/o	0.98
Grid.100	100	180	20	20	10,803	153.02	0.02	20	20	10,412	171.52	0.03	20	20	47,869	1362.92	0.04
Grid.900	900	1740	11.52	108	3153	t/o	0.68	8.56	136	4236	t/o	1.13	0.00	264	10,202	t/o	0.67
RCat.134	134	133	3	3	70	0.71	0.02	3	3	232	2.55	0.02	3	3	330	7.95	0.02
RCat.554	554	553	3	3	42	8.29	0.18	3	3	822	120.29	0.25	3	3	222	37.69	0.13
WGrid.100	100	200	30.09	42	175,218	t/o	0.02	24.29	47	102,872	t/o	0.03	20.79	45	65,024	t/o	0.03
G124.02	124	149	23	23	33,683	271.37	0.04	23	23	15,360	177.71	0.03	17.46	24	162,449	t/o	0.03
G124.04	124	318	44.96	113	74,296	t/o	0.04	33.42	122	59,910	t/o	0.04	27.47	115	64,289	t/o	0.07
G124.08	124	620	58.93	328	24,287	t/o	0.07	41.94	365	21,794	t/o	0.08	41.48	323	28,634	t/o	0.06
G250.01	250	331	28.77	54	77,865	t/o	0.07	22.79	58	39,457	t/o	0.11	16.06	57	44,902	t/o	0.08
G500.005	500	625	17.32	97	17,203	t/o	0.26	13.72	107	12,879	t/o	0.33	4.86	121	18,119	t/o	0.17

these instances are not exactly the same ones but only randomly generated obeying the same rules as specified in Bui et al. (1987).

The computational results basically confirm those obtained in Liberti (2007) in that the compactly linearized program is, in most of the cases, superior to the models 'B' and 'B2' in terms of total as well as root relaxation solution times, the number of branching nodes, and the sustained lower bounds if an instance could not be solved within the time limit. For $m = 3$, however, there are some instances where formulation 'B' performs better. One can also see that it is indeed the compact linearization that makes the originally quadratic program competitive or even superior to Boulle's, as the model obtained with the 'standard linearization' is clearly inferior to both others. We remark that for structural reasons there is always a zero objective solution to the root linear program of each of the three formulations. However, we observed that the root lower bounds could be successively improved by cutting planes much better in case of the compactly linearized model.

5 Conclusion

In this paper, we introduced new necessary and sufficient conditions to apply the compact linearization approach for binary quadratic problems subject to assignment constraints as proposed by Liberti (2007). These conditions were proven to lead to consistent value assignments for all induced linearization variables. In addition, we presented a mixed-integer program that can be used to compute linearizations with a minimum number of additional variables and constraints. We also showed that, in the case where all the assignment constraints have non-overlapping variable support, this problem can be solved as a linear program as the constraint matrix is totally unimodular. As an alternative, an exact polynomial-time combinatorial algorithm is proposed that can also be used in a heuristic fashion for the more general setting with overlapping variable sets in the assignment constraints. Computational experiments confirmed the findings of Liberti that the compact linearization approach can lead to formulations that are superior to those obtained with an ordinary linearization.

Acknowledgements I would like to most sincerely thank Leo Liberti and an anonymous referee for helpful comments on the presentation of the examples and results in this paper.

References

Adams WP, Sherali HD (1986) A tight linearization and an algorithm for zero-one quadratic programming problems. Manag Sci 32(10):1274–1290. https://doi.org/10.1287/mnsc.32.10.1274

Adams WP, Sherali HD (1999) A reformulation-linearization technique for solving discrete and continuous nonconvex problems, nonconvex optimization and its applications. Springer, Berlin. https://doi.org/10.1007/978-1-4757-4388-3

Allemand K, Fukuda K, Liebling TM, Steiner E (2001) A polynomial case of unconstrained zero-one quadratic optimization. Math Progr 91(1):49–52. https://doi.org/10.1007/s101070100233

Balas E (1964) Extension de l'algorithme additif à la programmation en nombres entiers et à la programmation non linéaire. Comptes rendus de l'Académie des Sciences Paris 258:3136–5139

Balas E, Mazzola JB (1984) Nonlinear 0–1 programming: I. Linearization techniques. Math Progr 30(1):1–21. https://doi.org/10.1007/BF02591796
Billionnet A, Elloumi S, Lambert A (2008) Linear reformulations of integer quadratic programs. Springer, Berlin, pp 43–51. https://doi.org/10.1007/978-3-540-87477-5_5
Boulle M (2004) Compact mathematical formulation for graph partitioning. Optim Eng 5(3):315–333. https://doi.org/10.1023/B:OPTE.0000038889.84284.c7
Bui TN, Leighton FT, Chaudhuri S, Sipser M (1987) Graph bisection algorithms with good average case behavior. Combinatorica 7(2):171–191. https://doi.org/10.1007/BF02579448
Chaovalitwongse W, Pardalos PM, Prokopyev OA (2004) A new linearization technique for multi-quadratic 0–1 programming problems. Oper Res Lett 32(6):517–522. https://doi.org/10.1016/j.orl.2004.03.005
Fortet R (1959) L'algèbre de boole et ses applications en recherche opérationnelle. Cahiers du Centre d'Etudes de Recherche Opérationnelle 1:5–36
Fortet R (1960) Applications de l'algèbre de boole en recherche opérationnelle. Revue de la Société Française de Recherche Opérationnelle 4:17–26
Frieze AM, Yadegar J (1983) On the quadratic assignment problem. Discrete Appl Math 5(1):89–98. https://doi.org/10.1016/0166-218X(83)90018-5
Furini F, Traversi E (2013) Extended linear formulation for binary quadratic problems. Optim Online
Glover F (1975) Improved linear integer programming formulations of nonlinear integer problems. Manag Sci 22(4):455–460. https://doi.org/10.1287/mnsc.22.4.455
Glover F, Woolsey E (1973) Further reduction of zero-one polynomial programming problems to zero-one linear programming problems. Oper Res 21(1):156–161
Glover F, Woolsey E (1974) Converting the 0–1 polynomial programming problem to a 0–1 linear program. Oper Res 22(1):180–182. https://doi.org/10.1287/opre.22.1.180
Gueye S, Michelon P (2009) A linearization framework for unconstrained quadratic (0–1) problems. Discrete Appl Math 157(6):1255–1266. https://doi.org/10.1016/j.dam.2008.01.028
Hammer PL, Rudeanu S (1968) Boolean methods in operations research and related areas. Ökonometrie und Unternehmensforschung/Econometrics and Operations Research. Ökonometrie und Unternehmensforschung/Econometrics and Operations Research, Springer, Berlin. https://doi.org/10.1007/978-3-642-85823-9
Hansen P, Meyer C (2009) Improved compact linearizations for the unconstrained quadratic 0–1 minimization problem. Discrete Appl Math 157(6):1267–1290. https://doi.org/10.1016/j.dam.2007.12.008
Liberti L (2007) Compact linearization for binary quadratic problems. 4OR 5(3):231–245. https://doi.org/10.1007/s10288-006-0015-3
Nemhauser GL, Wolsey LA (1988) Integer and combinatorial optimization. Wiley, New York, NY
Oral M, Kettani O (1992a) A linearization procedure for quadratic and cubic mixed-integer problems. Oper Res 40(1–supplement–1):S109–S116. https://doi.org/10.1287/opre.40.1.S109
Oral M, Kettani O (1992b) Reformulating nonlinear combinatorial optimization problems for higher computational efficiency. Eur J Oper Res 58(2):236–249. https://doi.org/10.1016/0377-2217(92)90210-Z
Sahni S (1974) Computationally related problems. SIAM J Comput 3(4):262–279. https://doi.org/10.1137/0203021
Sahni S, Gonzalez T (1976) P-complete approximation problems. JACM 23(3):555–565. https://doi.org/10.1145/321958.321975
Sherali HD, Smith JC (2007) An improved linearization strategy for zero-one quadratic programming problems. Optim Lett 1(1):33–47. https://doi.org/10.1007/s11590-006-0019-0
Watters LJ (1967) Reduction of integer polynomial programming problems to zero-one linear programming problems. Oper Res 15:1171–1174
Zangwill WI (1965) Media selection by decision programming. J Advert Res 5:23–37