Optimization Letters manuscript No.
(will be inserted by the editor)

A Lagrangian Decomposition Approach to Computing Feasible Solutions for Quadratic Binary Programs

Wei-An Chen · Zhen Zhu · Nan Kong

Received: date / Accepted: date

Abstract In this paper, we develop a Lagrangian decomposition based heuristic method for general quadratic binary programs (QBPs) with linear constraints. We extend the idea of Lagrangian decomposition by Chardaire and Sutter (1995) and Billionnet and Soutif (2004) in which the quadratic objective is converted to a bilinear function by introducing auxiliary variables to duplicate the original complicating variables in the problem. Instead of using linear constraints to assure the equity between the two types of decision variables, we introduce generalized quadratic constraints and relax them with Lagrangian multipliers. Instead of computing an upper bound for a maximization problem, we focus on lower bounding with Lagrangian decomposition based heuristic. We take advantage of the decomposability presented in the Lagrangian subproblems to speed up the heuristic and identify one feasible solution at each iteration of the subgradient optimization procedure. With numerical studies on several classes of representative QBPs, we investigate the sensitivity of lower-bounding performance on parameters of the additional quadratic constraints. We also demonstrate the potentially improved quality of preprocessing in comparison with the use of a QBP solver.

Keywords Quadratic binary programming · Lagrangian decomposition · Preprocess · Quadratic assignment problem

Wei-An Chen
School of Chemical Engineering, Purdue University, West Lafayette, Indiana, USA
E-mail: chen1702@purdue.edu

Zhen Zhu
School of Industrial Engineering, Purdue University, West Lafayette, Indiana, USA
E-mail: zzhu@purdue.edu

Nan Kong
Weldon School of Biomedical Engineering, Purdue University, West Lafayette, Indiana, USA
Tel.: +1 765-496-2467
Fax: +1 765-496-1459
E-mail: nkong@purdue.edu

1 Introduction

One of the most well-known and studied classes of nonlinear integer optimization problems is the maximization of a quadratic 0–1 function subject to a set of linear 0–1 constraints:

$$(\text{P0}): \quad \max_x \left\{ \sum_{i=1}^{n} c_i x_i + \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \frac{1}{2} c_{ij} x_i x_j \right\} \tag{1}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} a_{ki} x_i \leq b_k, \ \text{for } k = 1, \ldots, m; \tag{2}$$

$$x_i \in \{0, 1\}, \ \text{for } i = 1, \ldots, n, \tag{3}$$

where $c_i, c_{ij} \in \mathbb{R}$ for $1 \leq i \neq j \leq n$ and $b_k, a_{ki} \in \mathbb{R}$ for $k = 1, \ldots, m$ and $i = 1, \ldots, n$. Problem (P0) is typically referred to as a constrained *quadratic binary problem* (QBP) [11]. Since for each binary variable $x_i^2 = x_i$, then $c_{ii} x_i^2$ can be rewritten as $c_i x_i$ with $c_i = c_{ii}$. Without loss of generality, we also assume that $c_{ij} = c_{ji}$ for any pair (i, j) with $i \neq j$.

Many important problems in engineering, physics, chemistry, biology, and medicine, as well as several other application domains, can be formulated as QBPs. For example, QBPs have been studied in scheduling [3], solid-state physics [4, 5], protein design [28], computational biology [18], and epileptic seizure prediction [26]. In addition, many graph-theoretic problems can be naturally formulated with QBPs, including well-studied maximum clique and maximum independent set problems [37]. However, only a small number of classes of QBPs are known to be polynomially solvable [35, 39]. In general, QBPs are NP-hard combinatorial optimization problems [11, 36].

For solving general QBPs, we have witnessed the development of various heuristics and exact solution methods. A large set of exact solution methods focused on efficient linearization techniques, which shares the concept of replacing the nonlinear terms with auxiliary variables and including an additional set of linear constraints accordingly [1, 2, 9, 20, 31, 32]. Another large set of exact solution methods involved use of semidefinite programming (SDP) relaxation, which is shown to be equivalent to the Lagrangian dual of the primal problem in general cases [43]. Combined with cutting plane technique or branch-and-bound framework, SDP relaxation has been exceedingly beneficial to solving unconstrained QBPs [7, 25, 40], even for nonconvex QBPs [13, 44].

Additionally, innumerable metaheuristic approaches are designed for solving QBPs, improving solutions, or speeding up algorithms. Many of them were problem-specific and mainly focused on unconstrained QBPs [12, 21, 22, 23, 29, 30, 33]. In recent years, several metaheuristics were also proposed for quadratic assignment problems (QAPs) [16, 38, 41, 42].

Among the articles we have reviewed, few methods aimed at developing lower-bounding schemes rather than proposing tighter upper bounds with respect to maximization programs, and fewer pursued efficient computation of initial feasible solutions for generic problems. Some of the noteworthy articles are briefly reviewed in the following. First, Ivnescu [27] linked the problem of maximal flows in a network to the pseudo-Boolean problem by the minimum-cut theory [17]. Chaillou et al [14] worked on a Lagrangian relaxation of quadratic knapsack binary problems (QKBPs) and characterized the Lagrangian function following [27]. With the characterization, they developed an algorithm that quickly finds a feasible solution to QKBPs through iterative variable fixation. Billionnet and Calmels [6] further incorporated the algorithm with a fill-up and exchange procedure proposed by Gallo et al [19] to improve the solution. In addition, Gallo et al [19] reported theoretical and computational results of an upper plane technique for QKBPs, which relaxes the quadratic objective function with a linear function that has higher values over the entire feasible region. By solving this relaxed problem, an upper bound and a feasible solution are obtained. It is worth noting that the application of these methods may be restricted to QKBPs due to non-negative coefficients and single constraint. However, the concept of constructing a more amiable upper plane function to relax the original problem leads us to develop a more generic scheme for computing initial feasible solutions.

Our scheme was inspired by the idea of making copies of decision variables and introducing additional constraints to ensure the identity between each original decision variable and its copies. The most notable method is the one developed by Chardaire and Sutter [15] for unconstrained QBPs. Subsequently, Billionnet et al [10] and Billionnet and Soutif [8] extended their work to QBKPs. Consider a partition of decision variables $X = \{x_1, \ldots, x_n\}$ into p clusters ($p \leq n$) of variables, i.e., $X = X_1 \oplus \ldots \oplus X_p$, and define $Y_s = X \setminus X_s$. Let I_s and J_s be the index sets of variables in X_s and Y_s, respectively. The key idea is to introduce an auxiliary decision variable (or say copy variable) y_j^s that corresponds to each decision variable x_j and cluster s, and replace each term $c_{ij}x_ix_j$ in objective (1) by $c_{ij}x_iy_j^s$. Then (P0) can be equivalently reformulated as:

$$\text{(P1)}: \max_{x,y} \left\{ \sum_{s=1}^{p} \left(\sum_{i \in I_s}^{n} c_i x_i + \sum_{i \in I_s} \sum_{j \in I_s \setminus \{i\}} \frac{1}{2} c_{ij} x_i x_j + \sum_{i \in I_s} \sum_{j \in J_s} \frac{1}{2} c_{ij} x_i y_j^s \right) \right\} \tag{4}$$

$$\text{s.t.} \quad \sum_{i \in I_s} a_{ki} x_i + \sum_{j \in J_s} a_{\ell j} y_j^s \leq b_k, \ k = 1, \ldots, m, \ s = 1, \ldots, p; \tag{5}$$

$$y_j^s = x_j, \ j \in J_s, \ s = 1, \ldots, p; \tag{6}$$

$$x_i, y_j^s \in \{0, 1\}, \ i \in I_s, \ j \in J_s, \ s = 1, \ldots, p. \tag{7}$$

Note that an upper bound on (P0) can be obtained via Lagrangian relaxation of (P1) by dualizing constraints (6), though that x and y variables are not completely decomposable in computation of the Lagrangian dual.

In our scheme, we propose an alternative Lagrangian dual problem, where we still introduce copy variable y_j^i for each pair (i, j), $i \neq j$, to replace

$c_{ij}x_ix_j$ terms in the objective function with $c_{ij}x_iy_j^i$, but generate parameterized quadratic constraints instead of linear constraints (6). The equivalence between our proposed formulation and (P0) can be established by certain specification of the parameters in those quadratic constraints. More importantly, in contrast to the approach described above, the computation of our Lagrangian dual problem can be decomposed into solving n unconstrained linear binary programs that only involves decision variables y and one constrained linear binary program involving only decision variables x. By solving the subproblems, we can efficiently obtain a set of promising feasible solutions, which is expected to shorten the searching process in a branch-and-bound procedure.

2 Main Results

2.1 An Alternative Reformulation of (P0)

Consider the following general quadratic binary problem:

$$(\text{P2}): \max_{x,y} \sum_{i=1}^{n} \left(c_i x_i + \sum_{j=1, j\neq i}^{n} \frac{1}{2} c_{ij} x_i y_j^i \right) \tag{8}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} a_{ki} x_i \leq b_k, \quad k = 1, \ldots, m \tag{9}$$

$$\alpha_{ij}^{\ell} x_i y_j^i + \beta_{ij}^{\ell} x_j y_i^j + \theta_{ij}^{\ell} x_i + \gamma_{ij}^{\ell} x_j \geq \epsilon_{ij}^{\ell}, \quad i,j = 1, \ldots, n, i < j, \ \ell = 1, \ldots, r_{ij} \tag{10}$$

$$x_i, y_i^j \in \{0,1\}, \quad i,j = 1, \ldots, n, i \neq j \tag{11}$$

In (P2), auxiliary binary variables y_j^i are introduced to pair with x_i for every j ($\neq i$); hence, for each x_i, there are $n-1$ auxiliary variables y_i^j introduced. In addition, no additional linear constraints is proposed to link x and y but r sets of parameterized quadratic constraints for each pair (i,j) with $i < j$. Regarding the number of constraints in (10), r_{ij} can be any value of our choice under certain condition, which is stated more clearly in Section 2.3. Note that (8) is identical to (1) and equivalent to (4).

Knowing the equivalence between (P0) and (P1), in the following we show that the equivalence between (P1) and (P2) can be established as long as certain relationships between the parameters $(\alpha, \beta, \theta, \gamma, \epsilon)$ are satisfied. For convenience of the exposition, we use shorthand notation (x,y) to denote $(x_1, \ldots, x_n, y_2^1, \ldots, y_n^1, \ldots, y_1^n, \ldots, y_{n-1}^n)$ in the discussion below. Let

$$\mathcal{A} = \{(x,y) \in \{0,1\}^n \times \{0,1\}^{n(n-1)} \mid (5) - (7)\}$$

be the feasible solution region of (P1) and

$$\mathcal{B}(\alpha, \beta, \theta, \gamma, \epsilon) = \{(x,y) \in \{0,1\}^n \times \{0,1\}^{n(n-1)} \mid (9) - (11)\}$$

be the feasible solution region of (P2) parameterized on $\alpha := (\alpha_{ij}), \beta := (\beta_{ij}), \theta := (\theta_{ij}), \gamma := (\gamma_{ij})$, and $\epsilon := (\epsilon_{ij})$. For notational simplicity, we use $\mathcal{B}$ instead of $\mathcal{B}(\alpha, \beta, \theta, \gamma, \epsilon)$ when referring to a parameterized feasible solution region of (P2). The parameters are always specified when making such a reference. We also divide the set of all possible combinations of the decision variables $(x_i, x_j, y_i^j, y_j^i) \in \{0,1\}^4$ into the following subsets:

$$C_1 = \{(1,1,1,1),\ (1,0,1,0),\ (0,1,0,1),\ (0,0,0,0)\},$$
$$C_2 = \{(1,0,0,0),\ (0,1,0,0),\ (0,0,0,1),\ (0,0,1,0),\ (0,0,1,1)\},$$
$$C_3 = \{0,1\}^4 \backslash (C_1 \cup C_2).$$

To ensure the equivalence between (P1) and (P2), we essentially need to show that $\mathcal{A} \subseteq \mathcal{B}$ and no feasible solution to (P2) is from C_3.

Proposition 1 *If for all $\ell = 1, \ldots, r_{ij}$ and for all (i,j), where $i, j = 1, \ldots, n$ and $i < j$, the following conditions hold, then $\mathcal{A} \subseteq \mathcal{B}$.*

$$\alpha_{ij}^{\ell} + \beta_{ij}^{\ell} + \theta_{ij}^{\ell} + \gamma_{ij}^{\ell} \geq \epsilon_{ij}^{\ell} \tag{12}$$
$$\theta_{ij}^{\ell} \geq \epsilon_{ij}^{\ell} \tag{13}$$
$$\gamma_{ij}^{\ell} \geq \epsilon_{ij}^{\ell} \tag{14}$$
$$\epsilon_{ij}^{\ell} \leq 0 \tag{15}$$

Proposition 1 implies that with parameter specification as in (12) – (15), we ensure that any combination in C_1 must satisfy all corresponding constraints (10) for every index pair.

Proposition 2 *Consider any $(x, y) \in \mathcal{B}$. If $(x_i, x_j, y_i^j, y_j^i) \in C_1 \cup C_2$ for each (i,j), $i, j = 1, \ldots, n$ and $i < j$, then there exists $(x, \hat{y}) \in \mathcal{A}$ such that (x, y) in (P2) and $(x, \hat{y})$ in (P1) yield the same objective function value.*

Proposition 2 implies that although $\mathcal{A} \subseteq \mathcal{B}$ allows more feasible solutions to (P2) than (P1), such feasible solution region expansion does not affect the optimality equivalence between (P1) and (P2) as long as for all (i,j), $1 \leq i < j \leq n$, $(x_i, x_j, y_i^j, y_j^i) \in C_1 \cup C_2$. Next, we show certain expansion should not be allowed as it may affect the optimality equivalence between (P1) and (P2). Thus, we should prohibit such expansion by specifying parameters in constraint (10). We provide a sufficient condition to ensure this as fellows.

Proposition 3 *Suppose for some (s,t), $s, t = 1, \ldots, n$ and $s < t$, there exists an index set $L_{st} := (l_1, l_2, l_3, l_4, l_5) \in \{1, \ldots, r_{st}\}^5$ such that the following conditions hold:*

$$\alpha_{st}^{\ell_1} + \theta_{st}^{\ell_1} + \gamma_{st}^{\ell_1} < \epsilon_{st}^{\ell_1}; \tag{16}$$
$$\beta_{st}^{\ell_2} + \theta_{st}^{\ell_2} + \gamma_{st}^{\ell_2} < \epsilon_{st}^{\ell_2}; \tag{17}$$
$$\theta_{st}^{\ell_3} + \gamma_{st}^{\ell_3} < \epsilon_{st}^{\ell_3}; \tag{18}$$
$$\alpha_{st}^{\ell_4} + \theta_{st}^{\ell_4} < \epsilon_{st}^{\ell_4}; \tag{19}$$
$$\beta_{st}^{\ell_5} + \gamma_{st}^{\ell_5} < \epsilon_{st}^{\ell_5}. \tag{20}$$

For any (x, y), if $(x_s, x_t, y_s^t, y_t^s) \in C_3$, then $(x, y) \notin \mathcal{B}$.

Note that it is allowed in Proposition 3 that $l_{r_1} = l_{r_2}$ for $1 \le r_1 \ne r_2 \le 5$ and that the choice of L_{st} may not be unique for a given index pair (s,t). Proposition 3 implies that $\mathcal{B}$ should not contain any (x,y) from C_3, since such a solution may destroy the optimality equivalence between (P1) and (P2). With the three propositions presented earlier, we readily state one sufficient condition to ensure (P2) is a valid reformulation of (P0).

Theorem 1 *Formulations (P1) and (P2) are equivalent in the sense that they yield the same optimal objective function value and the same optimal solution with respect to x, if there exists $(\alpha,\beta,\theta,\gamma,\epsilon)$ such that for any index pair (i,j) with $i<j$, the following conditions hold:*

1. *for all $\ell = 1,\ldots,r_{ij}$,*

$$\alpha_{ij}^{\ell} + \beta_{ij}^{\ell} + \theta_{ij}^{\ell} + \gamma_{ij}^{\ell} \ge \epsilon_{ij}^{\ell}, \tag{21}$$

$$\theta_{ij}^{\ell} \ge \epsilon_{ij}^{\ell}; \tag{22}$$

$$\gamma_{ij}^{\ell} \ge \epsilon_{ij}^{\ell}; \tag{23}$$

$$\epsilon_{ij}^{\ell} \le 0. \tag{24}$$

2. *there exists an index set L_{ij}, i.e., $(\ell_1^{ij},\ell_2^{ij},\ell_3^{ij},\ell_4^{ij},\ell_5^{ij})$ such that*

$$\alpha_{ij}^{\ell_1^{ij}} + \theta_{ij}^{\ell_1^{ij}} + \gamma_{ij}^{\ell_1^{ij}} < \epsilon_{ij}^{\ell_1^{ij}}; \tag{25}$$

$$\beta_{ij}^{\ell_2^{ij}} + \theta_{ij}^{\ell_2^{ij}} + \gamma_{ij}^{\ell_2^{ij}} < \epsilon_{ij}^{\ell_2^{ij}}; \tag{26}$$

$$\theta_{ij}^{\ell_3^{ij}} + \gamma_{ij}^{\ell_3^{ij}} < \epsilon_{ij}^{\ell_3^{ij}}; \tag{27}$$

$$\alpha_{ij}^{\ell_4^{ij}} + \theta_{ij}^{\ell_4^{ij}} < \epsilon_{ij}^{\ell_4^{ij}}; \tag{28}$$

$$\beta_{ij}^{\ell_5^{ij}} + \gamma_{ij}^{\ell_5^{ij}} < \epsilon_{ij}^{\ell_5^{ij}}. \tag{29}$$

Theorem 1 establishes the fact that appropriate parameter specification for constraints (10) can yield a valid reformulation of (P0). Then with introduction of the copying variables, one may find it is viable to deal with (P2) instead of (P0). We focus on a Lagrangian decomposition based approach to obtain a promising feasible solution efficiently, and we hope the proposed lower-bounding scheme can contribute to exact solution development for general QBPs.

2.2 Lagrangian Decomposition of (P2)

Assume that the parameters in (10) are already specified to let (P2) be equivalent to (P0). For each pair $1 \le i < j \le n$ and each $\ell = 1,\ldots,r_{ij}$, we associate a Lagrangian multiplier λ_{ij}^{ℓ} with the corresponding constraint (10). Then the Lagrangian dual problem of (P2) is given by

$$\min_{\lambda \ge 0} \max_{x,y} \{ L(\lambda,x,y) \mid (9),(11) \}, \tag{30}$$

where $L(\lambda,x,y) = f(x,y) + \sum_{i=1}^{n}\sum_{j>i}\sum_{\ell=1}^{r_{ij}} \left(\lambda_{ij}^{\ell}(\alpha_{ij}^{\ell}x_iy_j^i + \beta_{ij}^{\ell}x_jy_i^j + \theta_{ij}^{\ell}x_i + \gamma_{ij}^{\ell}x_j - \epsilon_{ij}^{\ell})\right) =$

$$\sum_{i=1}^{n}\left(c_i + \sum_{j>i}\sum_{\ell=1}^{r_{ij}}\lambda_{ij}^{\ell}\theta_{ij}^{\ell} + \sum_{j<i}\sum_{\ell=1}^{r_{ji}}\lambda_{ji}^{\ell}\gamma_{ji}^{\ell} + \sum_{j>i}(\frac{c_{ij}}{2} + \sum_{\ell=1}^{r_{ij}}\lambda_{ij}^{\ell}\alpha_{ij}^{\ell})y_j^i + \sum_{j<i}(\frac{c_{ij}}{2} + \sum_{\ell=1}^{r_{ji}}\lambda_{ji}^{\ell}\beta_{ji}^{\ell})y_j^i\right)x_i - \sum_{i=1}^{n}\sum_{j>i}\sum_{\ell=1}^{r_{ij}}\lambda_{ij}^{\ell}\epsilon_{ij}^{\ell}. \quad (31)$$

Define dual function $L(\lambda)$ as:

$$\begin{aligned} L(\lambda) &= \max_{x,y} L(\lambda,x,y) = \max_{x,y}\left\{\sum_{i=1}^{n}\left(g_i(y^i)x_i\right) \;\middle|\; (9),(11)\right\} - \sum_{i=1}^{n}\sum_{j>i}\sum_{\ell=1}^{r_{ij}}\lambda_{ij}^{\ell}\epsilon_{ij}^{\ell} \\ &= \max_{x}\left\{\sum_{i=1}^{n}\left(\max_{y^i} g_i(y^i)\right)x_i \;\middle|\; (9),(11)\right\} - \sum_{i=1}^{n}\sum_{j>i}\sum_{\ell=1}^{r_{ij}}\lambda_{ij}^{\ell}\epsilon_{ij}^{\ell}, \end{aligned} \quad (32)$$

where

$$\begin{aligned} g_i(y^i) = c_i + \sum_{j>i}\sum_{\ell=1}^{r_{ij}}\lambda_{ij}^{\ell}\theta_{ij}^{\ell} + \sum_{j<i}\sum_{\ell=1}^{r_{ji}}\lambda_{ji}^{\ell}\gamma_{ji}^{\ell} + \sum_{j>i}(\frac{1}{2}c_{ij} + \sum_{\ell=1}^{r_{ij}}\lambda_{ij}^{\ell}\alpha_{ij}^{\ell})y_j^i \\ + \sum_{j<i}(\frac{1}{2}c_{ij} + \sum_{\ell=1}^{r_{ji}}\lambda_{ji}^{\ell}\beta_{ji}^{\ell})y_j^i, \end{aligned} \quad (33)$$

for $i = 1, \ldots, n$.

Lemma 1 *Consider a function $f(x,y) = g(y)h(x)$ with $x \in X$ and $y \in Y$. If $h(x) \geq 0$ for all $x \in X$, then $\max_{x\in X, y\in Y} f(x,y) = \max_{x\in X,y\in Y} g(y)h(x) = \max_{x\in X}\{(\max_{y\in Y} g(y))h(x)\}$.*

Using Lemma 1 we can decompose computing $L(\lambda)$ to n unconstrained linear binary problems and one linear binary program. For each $i = 1, \ldots, n$, let

$$g_i^* = \max_{y^i} g_i(y^i), \quad (34)$$

which is an unconstrained linear binary program only involving y^i. Then

$$L(\lambda) = \max_{x}\left\{\sum_{i=1}^{n} g_i^* x_i - \sum_{i=1}^{n}\sum_{j>i}\sum_{\ell=1}^{r_{ij}}\lambda_{ij}^{\ell}\epsilon_{ij}^{\ell} \;\middle|\; (9),(11)\right\}, \quad (35)$$

which is a linear binary program only involving x. We refer to (35) as the *master subproblem after decomposition (MSAD)*.

By applying Lemma 1, we can compute $L(\lambda)$ by solving n unconstrained linear binary problems, each of which corresponds to y^i, $i = 1, \ldots, n$, and one linear binary program, which corresponds to x. It is easy to solve each unconstrained integer program (IP) $\max_{y^i} g_i(y^i)$ with $n-1$ binary variables.

Thus, generally speaking, the complexity of solving $L(\lambda)$ is determined by the complexity of solving MSAD.

Note that MSAD preserves the feasible solution region of (P0). Hence, we can directly apply heuristics or exact solution methods that are shown efficient to the linear binary programming counterpart of (P2). More importantly, for any given $\lambda \geq 0$, solving MSAD yields a feasible solution to (P0). Thus, applying an iterative algorithm (e.g., subgradient algorithm) to solving MSAD may provide a number of promising feasible solutions to (P0).

2.3 Parameter Specification in Quadratic Constraints (10)

To ensure the equivalence of (P0) and (P2), specifying the parameters in constraints (10) is necessary. Theorem 1 provides a sufficient condition for such specification that as long as (21)-(29) are satisfied. For convenience, it is valid to use any constant number of quadratic constraints for each index pair, i.e., $r_{ij} = r$ for all $1 \leq i < j \leq n$. For this paper, we implemented the following set of parameters to conduct initial experiments.

$$\alpha_{ij}^1 = 1,\ \beta_{ij}^1 = 0,\ \theta_{ij}^1 = -1,\ \gamma_{ij}^1 = -1,\ \epsilon_{ij}^1 = -1; \tag{36}$$
$$\alpha_{ij}^2 = 0,\ \beta_{ij}^2 = 1,\ \theta_{ij}^2 = -1,\ \gamma_{ij}^2 = -1,\ \epsilon_{ij}^2 = -1; \tag{37}$$
$$\alpha_{ij}^3 = -1,\ \beta_{ij}^3 = 0,\ \theta_{ij}^3 = 0,\ \gamma_{ij}^3 = 1,\ \epsilon_{ij}^3 = 0; \tag{38}$$
$$\alpha_{ij}^4 = 0,\ \beta_{ij}^4 = -1,\ \theta_{ij}^4 = 1,\ \gamma_{ij}^4 = 0,\ \epsilon_{ij}^4 = 0; \tag{39}$$
$$\alpha_{ij}^5 = 1,\ \beta_{ij}^5 = -1,\ \theta_{ij}^5 = 0,\ \gamma_{ij}^5 = 0,\ \epsilon_{ij}^5 = 0; \tag{40}$$
$$\alpha_{ij}^6 = -1,\ \beta_{ij}^6 = 1,\ \theta_{ij}^6 = 0,\ \gamma_{ij}^6 = 0,\ \epsilon_{ij}^6 = 0. \tag{41}$$

It can be checked that (36)–(41) satisfy the conditions in Theorem 1. To be specific, for all (i, j) with $1 \leq i < j \leq n$, we have (i) (36)–(41) all satisfy conditions (21)–(24); and (ii) (37) and (41) satisfy (25), (36) and (40) satisfy (26), (36) and (37) satisfy (27), (38) and (41) satisfy (28), and (39) and (40) satisfy (29). Therefore, one can select ℓ_i, $i = 1, \ldots, 5$, accordingly. For example, by setting $(\ell_1, \ell_2, \ell_3, \ell_4, \ell_5) = (2, 1, 2, 3, 4)$, the conditions in Theorem 1 are satisfied, suggesting that the first four sets of parameters is sufficient enough to ensure the equivalence between (P0) and (P2).

With the parameters specified in (36)–(41), we can simplify the quadratic constraints (10) as:

$$x_i y_j^i \geq x_i + x_j - 1, \tag{42}$$
$$x_j y_i^j \geq x_i + x_j - 1, \tag{43}$$
$$x_i \geq x_j y_i^j, \tag{44}$$
$$x_j \geq x_i y_j^i, \tag{45}$$
$$x_i y_j^i = x_j y_i^j, \tag{46}$$

for each (i,j), $1 \leq i < j \leq n$. Note that (40) and (41) yield $x_i y_j^i \geq x_j y_i^j$ and $x_i y_j^i \leq x_j y_i^j$ respectively, and thus form equality (46). By replacing $z_{ij} = x_i y_j^i = x_j y_i^j$ for each pair (i,j), $1 \leq i < j \leq n$, it is clear and interesting to see the similarity between constraints (42)–(46) and the constraints in the standard linearization formulation of (P0).

Remark 1 It is impossible to cover all the rules specified in Theorem 1 with only one set of quadratic constraints in (P2) formulation, i.e., $r_{ij} = 1$ in (10), because (21), (28), and (29) can not be satisfied simultaneously. Therefore, the minimum number of quadratic constraints for each index pair (i,j) is 2, i.e., $r_{ij} \geq 2$ for all $1 \leq i < j \leq n$.

Remark 2 There are infinite number of valid parameter specifications. For example, if we use two quadratic constraints for each index pair (i,j) with $1 \leq i < j \leq n$, i.e., $r_{ij} = 2$, and set $\alpha_{ij}^1 = \beta_{ij}^2 = 1$, $\alpha_{ij}^2 = \beta_{ij}^1 = -1 - \epsilon$, and $\theta_{ij}^1 = \theta_{ij}^2 = \gamma_{ij}^1 = \gamma_{ij}^2 = \epsilon_{ij}^1 = \epsilon_{ij}^2 = \epsilon$, then for any value of ϵ where $-1 < \epsilon < 0$, there exist a parameter specification that satisfies Theorem 1.

3 Computational Experiments

We examined the performance of our lower-bounding scheme through solving four classes of QBPs with different constraints: unconstrained QBP, dense k-subgraph problem (DkSP), quadratic semi-assignment problem (QSAP), and quadratic assignment problem (QAP). The value of k in DkSP is set to be a half of the instance size, i.e., $k = n/2$. With parameters specified as in (36) – (41), we iteratively solved the decomposed subproblems, (34) and (35) with starting Lagrangian multiplier values of zeros, and updated the multipliers using a subgradient method [24] until the relaxed problem (30) was solved to optimality. Every solution obtained from (35) in each iteration was feasible to the original problem and therefore recorded. Our lower-bounding scheme ended with identifying the feasible solution that has the largest objective value. For comparison, the same instances were solved directly with the QBP solver of Gurobi 6.5.0.

In our experiment, test instances were randomly generated with the method described in [34]. Objective function coefficient matrices have diagonal elements sampled from uniform distribution with interval $[0, 75]$ and off-diagonal elements drawn uniformly from $[-50, 50]$. All the generated objective coefficient matrices are of full density. We considered three problem sizes for each class of QBPs and 10 test instances for each problem size. All computational work was implemented in Python 2.7.8 on a Linux 64bit machine with 66GB RAM and 32 CPU cores, where each job only uses one core with clock rate of 2.3 GHz and has a computing time limit of 5 hours.

Table 1 reports the comparative results of embedding our lower-bounding scheme in the Gurobi solver as a preporcessing scheme and without our scheme, in terms of optimality gap and computational time. The experiments without

Table 1 Comparative results on the QBP instances in terms of optimality and CPU time.

		Without Our Scheme				With Our Scheme				
QBP Class	Inst. Size	Opt. Gap (%) Max	Avg	Min	Time (s)	Opt. Gap (%) Max	Avg	Min	Time (s) L.D.	Total
Uncons -trained	$n = 30$		0		0.31		0		1.39	1.64
	$n = 50$		0		273		0		4.7	25.5
	$n = 100$	56.6	45.7	34.3	5 hr.	45.8	35.3	24.0	21	5 hr.
DkSP (k=n/2)	$n = 30$		0		0.51		0		1.42	1.75
	$n = 50$		0		129		0		4.5	87.1
	$n = 100$	51.2	43.6	36.4	5 hr.	51.0	43.0	35.9	20	5 hr.
QSAP	$n = 5$		0		0.24		0		0.99	1.24
	$n = 10$		0		15176		0		22	13976
	$n = 15$	474	425	386	5 hr.	453	423	407	118	5 hr.
QAP	$n = 5$		0		0.24		0		11.7	11.9
	$n = 10$		0		473		0		35	434
	$n = 15$	498	462	422	5 hr.	473	454	415	136	5 hr.

our scheme let the Gurobi solver solve the instances along with its default preprocessing functions of presolve and lower-bounding heuristic, whereas the experiments with our scheme used our preprocessing scheme first and inputted the identified feasible solution as the initial lower bound for the Gurobi's mixed-integer linear programming (MILP) solver. Among the four classes, only QAP employed the MILP solver to the one constrained linear binary subproblem (35) iteratively during the preprocessing. For each QBP class and size, we randomly generated 10 instances, and the columns time in Table 1 show the average CPU time of solving the 10 instances. The abbreviations L.D. means the time spent on the Lagrangian decomposition to acquire the feasible solution.

Suggested by the results in Table 1, when the instance size is small ($n = 30$ for unconstrained QBP and DkSP and $n = 5$ for QSAP and QAP), inputting the feasible solution we picked shortens the solving time taken by the Gurobi solver. However, solving Lagrangian dual problem takes too long and hence offsets the benefit it brings. As the instance size increases, the advantage of providing a good starting lower bound arises. For all but one of the instances with middle size ($n = 50$ for unconstrained QBP and DkSP and $n = 10$ for QSAP and QAP), using our lower-bounding scheme helps reaching the same optimal solutions in shorter amounts of time. The benefit of using our lower-bounding scheme remains when it comes to the large size instances ($n = 100$) of unconstrained QBP and DkSP, which is reflected on the optimality gaps. However, when the complexity of the problem keeps increasing, the disadvantage of our crude approach of incorporating the Lagrangian decomposition approach appears. Among large-size instances ($n = 15$) of QSAP and QAP, only six and seven out of ten instances, respectively, have smaller optimality gaps compared to the experiments without the use of our scheme. Nevertheless, an important point made through our preliminary experiment is that in every instance we have tested, the feasible solution obtained by our scheme is superior to the initial feasible solution came from Gurobi's default heuristics for its own lower bounding, even though this advantage may not be effective when solving large-size instances of QSAP and QAP.

Furthermore, we examined the effect of different parameter specifications on the computational results. Based on the six valid specifications we tested, we noted that there is no specification that guarantees the best lower bound. Different specifications yield better or worse lower bounds when applied to solving the instances. In terms of the experimental results, the CPU time and optimality gap are not sensitive to parameter specifications. The only suggestion we can confidently make is that it is not beneficial to add many quadratic constraints (i.e., the number of r_{ij} in (10) is greater than 6).

4 Concluding Remarks

In this paper, we describe a generic approach to acquiring a good feasible solution of general QBPs and how the parameterized quadratic constraints help attaining computational benefits through decomposing the original problem into binary linear subprograms but not changing the underlying structure. The feasible solution provides a promising initial lower bound for the followed branch-and-bound procedure, which likely contributes to the computational improvement, as shown in our experiments.

One limitation of this work is that the way we examined the performance of the lower bounds is preliminary and not elaborate. Simply inputting the feasible solution can result in poor compatibility with the Groubi solvers and thus worsened computational efficiency. Another limitation is that we only performed computational experiments on a small set of QBP classes. It is interesting to conduct comprehensive study on other common classes of QBPs and more general QBPs. In addition, it is interesting to derive more comprehensive Lagrangian decomposition schemes by considering the cases where each cluster contains more than one variable, i.e., $p < n$ and $|I_s| > 1$ for some cluster index s (see (P1)). It is also interesting to investigate the effect of combining the proposed quadratic constraints with other constraints for linearization (e.g., (6)), and study how to integrate our lower-bounding scheme with existing exact methods. For instance, expanding the use of our lower-bounding scheme to every node of branch-and-bound procedure instead of only at the root could have better performance. We leave the aforementioned topics and the potential refinement of our lower-bounding scheme to future research.

Acknowledgements This research was partially supported by Air Force Office of Scientific Research Grant FA9550-08-1-0268. We also thank Dr. Oleg A. Prokopyev's suggestions and comments on the research project.

References

1. Adams WP, Forrester RJ (2005) A simple recipe for concise mixed 0-1 linearizations. Operations Research Letters 33(1):55–61

2. Adams WP, Forrester RJ (2007) Linear forms of nonlinear expressions: New insights on old ideas. Operations Research Letters 35(4):510–518
3. Alidaee B, Kochenberger G, Ahmadian A (1994) 0-1 quadratic programming approach for optimum solutions of two scheduling problems. International Journal of Systems Science 25(2):401–408
4. Barahona F (1982) On the computational complexity of Ising spin glass models. Journal of Physics A: Mathematical and General 15:3241
5. Barahona F, Grötschel M, Jünger M, Reinelt G (1988) An application of combinatorial optimization to statistical physics and circuit layout design. Operations Research 36(3):493–513
6. Billionnet A, Calmels F (1996) Linear programming for the 0-1 quadratic knapsack problem. European Journal of Operational Research 92(2):310–325
7. Billionnet A, Elloumi S (2007) Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem. Mathematical Programming 109(1):55–68
8. Billionnet A, Soutif É (2004) An exact method based on Lagrangian decomposition for the 0-1 quadratic knapsack problem. European Journal of Operational Research 157(3):565–575
9. Billionnet A, Soutif E (2004) Using a mixed integer programming tool for solving the 0-1 quadratic knapsack problem. Informs Journal on Computing 16(2):188–197
10. Billionnet A, Faye A, Soutif É (1999) A new upper bound for the 0-1 quadratic knapsack problem. European Journal of Operational Research 112(3):664–672
11. Boros E, Hammer P (2002) Pseudo-boolean optimization. Discrete Applied Mathematics 123(1-3):155–225
12. Boros E, Hammer PL, Tavares G (2007) Local search heuristics for quadratic unconstrained binary optimization (qubo). Journal of Heuristics 13(2):99–132
13. Buchheim C, Wiegele A (2013) Semidefinite relaxations for non-convex quadratic mixed-integer programming. Mathematical Programming 141(1-2):435–452
14. Chaillou P, Hansen P, Mahieu Y (1989) Best network flow bounds for the quadratic knapsack problem, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 225–235
15. Chardaire P, Sutter A (1995) A decomposition method for quadratic zero-one programming. Management Science 41(4):704–712
16. Duman E, Uysal M, Alkaya AF (2011) Migrating birds optimization: A new meta-heuristic approach and its application to the quadratic assignment problem. Applications of Evolutionary Computation, Pt I 6624:254–263
17. Ford DR, Fulkerson DR (2010) Flows in Networks. Princeton University Press, Princeton, NJ, USA
18. Forrester R, Greenberg H (2008) Quadratic binary programming models in computational biology. Algorithmic Operations Research 3(2):110–129

19. Gallo G, Hammer PL, Simeone B (1980) Quadratic knapsack-problems. Mathematical Programming Study 12(Apr):132–149
20. Glover F (1975) Improved linear integer programming formulations of nonlinear integer problems. Management Science 22(4):455–460
21. Glover F, Kochenberger GA, Alidaee B (1998) Adaptive memory tabu search for binary quadratic programs. Management Science 44(3):336–345
22. Glover F, Alidaee B, Rego C, Kochenberger G (2002) One-pass heuristics for large-scale unconstrained binary quadratic problems. European Journal of Operational Research 137(2):272–287
23. Hanafi S, Rebai AR, Vasquez M (2013) Several versions of the devour digest tidy-up heuristic for unconstrained binary quadratic problems. Journal of Heuristics 19(4):645–677
24. Held M, Wolfe P, Crowder H (1974) Validation of subgradient optimization. Mathematical programming 6(1):62–88
25. Helmberg C, Rendl F (1998) Solving quadratic (0,1)-problems by semidefinite programs and cutting planes. Mathematical Programming 82(3):291–315
26. Iasemidis L, Pardalos P, Sackellares J, Shiau D (2001) Quadratic binary programming and dynamical system approach to determine the predictability of epileptic seizures. Journal of Combinatorial Optimization 5(1):9–26
27. Ivnescu PL (1965) Some network flow problems solved with pseudo-boolean programming. Operations Research 13(3):388–399
28. Klepeis J, Floudas C, Morikis D, Tsokos C, Lambriss J (2004) Design of peptide analogues with improved activity using a novel de novo protein design approach. Industrial and Engineering Chemistry Research 43(14):3817–3826
29. Lodi A, Allemand K, Liebling TM (1999) An evolutionary heuristic for quadratic 0-1 programming. European Journal of Operational Research 119(3):662–670
30. Lu ZP, Glover F, Hao JK (2010) A hybrid metaheuristic approach to solving the ubqp problem. European Journal of Operational Research 207(3):1254–1262
31. Nyberg A, Westerlund T (2012) A new exact discrete linear reformulation of the quadratic assignment problem. European Journal of Operational Research 220(2):314–319
32. Oral M, Kettani O (1992) A linearization procedure for quadratic and cubic mixed-integer problems. Operations Research 40(S1):S109–S116
33. Palubeckis G (2004) Multistart tabu search strategies for the unconstrained binary quadratic optimization problem. Annals of Operations Research 131(1-4):259–282
34. Pardalos P (1991) Construction of test problems in quadratic bivalent programming. ACM Transactions on Mathematical Software (TOMS) 17(1):74–87
35. Pardalos P, Jha S (1991) Graph separation techniques for quadratic zero-one programming. Computers & Mathematics with Applications 21(6-

7):107–113
36. Pardalos P, Jha S (1992) Complexity of uniqueness and local search in quadratic 0-1 programming. Operations Research Letters 11(2):119–123
37. Pardalos P, Xue J (1994) The maximum clique problem. Journal of Global Optimization 4(3):301–328
38. Paul G (2011) An efficient implementation of the robust tabu search heuristic for sparse quadratic assignment problems. European Journal of Operational Research 209(3):215–218
39. Picard J, Ratliff H (1975) Minimum cuts and related problems. Networks 5(4):357–370
40. Rendl F, Rinaldi G, Wiegele A (2010) Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. Mathematical Programming 121(2):307–335
41. Saremi HQ, Abedin B, Kermani AM (2008) Website structure improvement: Quadratic assignment problem approach and ant colony metaheuristic technique. Applied Mathematics and Computation 195(1):285–298
42. Sun JY, Zhang QF, Yao X (2014) Meta-heuristic combining prior online and offline information for the quadratic assignment problem. Ieee Transactions on Cybernetics 44(3):429–444
43. Xia Y, Xing WX (2015) Parametric lagrangian dual for the binary quadratic programming problem. Journal of Global Optimization 61(2):221–233
44. Xu Z, Hong MY, Luo ZQ (2014) Semidefinite approximation for mixed binary quadratically constrained quadratic programs. Siam Journal on Optimization 24(3):1265–1293