A Convex Reformulation and an Outer Approximation for a Large Class of Binary Quadratic Programs

Published Version:
Rostami, B., Errico, F., Lodi, A. (2023). A Convex Reformulation and an Outer Approximation for a Large Class of Binary Quadratic Programs. OPERATIONS RESEARCH, 71(2), 471-486 [10.1287/opre.2021.2241].

Availability:
This version is available at: https://hdl.handle.net/11585/905161 since: 2024-03-01

Published:
DOI: http://doi.org/10.1287/opre.2021.2241

(Article begins on next page)

Submitted to *Operations Research*
manuscript OPRE-2020-09-613.R2

A convex reformulation and an outer approximation for a large class of binary quadratic programs

Borzou Rostami
Lazaridis School of Business and Economics, Wilfrid Laurier University
CIRRELT and Canada Excellence Research Chair in Data-Science for Real-time Decision-Making, Polytechnique Montréal, Canada
bo.rostami@gmail.com

Fausto Errico
Department of Civil Engineering, École de Technologie Supérieure de Montréal, GERAD and CIRRELT, Canada
fausto.errico@etsmtl.ca

Andrea Lodi
CERC, Polytechnique Montréal, Canada
andrea.lodi@polymtl.ca

In this paper, we propose a general modeling and solving framework for a large class of binary quadratic programs subject to variable partitioning constraints. Problems in this class have a wide range of applications, as many binary quadratic programs with linear constraints can be represented in this form. By exploiting the structure of the partitioning constraints, we propose mixed-integer nonlinear programming (MINLP) and mixed-integer linear programming (MILP) reformulations and show the relationship between the two models in terms of the relaxation strength. Our solution methodology relies on a convex reformulation of the proposed MINLP and a branch-and-cut algorithm based on outer approximation cuts, where the cuts are generated on the fly by efficiently solving separation subproblems. To evaluate the robustness and efficiency of our solution method, we perform extensive computational experiments on various quadratic combinatorial optimization problems. The results show that our approach outperforms the state-of-the-art solver applied to different MILP reformulations of the corresponding problems.

Key words: binary quadratic program, convex reformulation, outer approximation, variable partitioning constraint

1. Introduction

The objective of this paper is to study, under a unifying framework, a large class of binary quadratic programs, which we call "partitioning binary quadratic program", BQP_P. The BQP_P is a binary quadratic program with linear constraints (BQP) that arises in the modelling of real-life applications. The graph partitioning problem (Hager and Krylyuk 1999, Zha et al. 2001, Fan and Pardalos 2010), single allocation hub location problem (Campbell and O'Kelly 2012, Contreras and Fernández 2014, Rostami et al. 2018b), multi-object tracking in computer vision (Trinh et al. 2012, Dehghan and Shah 2018, Henschel et al. 2018), multi-processor scheduling with communication delays (Padberg and Rijal 2012), quadratic semi-assignment problem (Hansen and Lih 1992, Malucelli and Pretolani 1994, Schüle et al. 2009), and test assignment problem (Duives et al. 2013) are among the known problems that can be represented as BQP_P.

Let n and m be positive integers, $\mathbb{B} = \{0,1\}$, $\mathbb{R}$ denote the set of reals, $Q \in \mathbb{R}^{n\times n}$ be a symmetric matrix, $A \in \mathbb{R}^{m\times n}$, and $c \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$ be real vectors. Moreover, let $N = \{1,2,\dots,n\}$ and $I_1, I_2, \dots, I_K$ define a partition of N with index set $\mathcal{K} = \{1,2,\dots,K\}$ such that for each $k \in \mathcal{K}$, $I_k \subset N$ and for each pair $k, \ell \in \mathcal{K}$, $I_k \cap I_\ell = \emptyset$. We consider the following problem:

$$\begin{aligned} BQP_P: \quad \min \quad & c^T x + x^T Q x \\ \text{s.t.} \quad & x \in X \cap P \cap \mathbb{B}^n, \end{aligned}$$

with

$$X = \{0 \le x \le 1 : \; Ax = b\}, \tag{1}$$

and

$$P = \{x | \; \textstyle\sum_{i \in I_k} x_i = 1, \; k \in \mathcal{K}\}. \tag{2}$$

Without loss of generality, we assume that $q_{ii} = 0, \; i \in N$ because one can add the diagonal elements of Q to the linear cost c. The set of constrains P restricts the number of binary variables in each subset I_k, $k \in \mathcal{K}$, to be one. By exploiting the structure of the set P, we develop a new non-convex MINLP formulation of the BQP_P and show how to transform it into a convex one. We then apply an outer approximation scheme and develop a novel branch-and-cut algorithm. We finally perform computational experiences on three classes of problems. Results show the superiority of our approach with respect to the state-of-the-art models and algorithms on each of these problems.

It is worth mentioning that it is possible to solve the BQP_P directly using CPLEX, Gurobi, and the Semidefinite-Programming-based software BiqCrunch (Krislock et al. 2017). The overall performance of those methods in "out-of-the-box" mode is, however, disappointing. More precisely, the computing times and the number of problems solved within time limit are much worse than both our proposed algorithm and the best improved formulations in the literature.

1.1. Literature Review

A known solution approach for the BQP is to use an initial linearization to transform the problem into an equivalent MILP. However, dealing with linear programming (LP) based reformulations, two different issues must be considered: the increased size of the problem in terms of the number of variables and constraints as well as the tightness of the obtained lower bounds. The standard strategy to linearize the quadratic terms $x_i x_j$ for all $i, j \in N$ is to introduce new binary variables $y_{ij} = x_i x_j$ satisfying the following set of constraints:

$$y_{ij} \le x_i, \quad y_{ij} \le x_j, \quad \text{and} \quad y_{ij} \ge x_i + x_j - 1.$$

The new formulation requires $O(n^2)$ additional variables and constraints and it is well known in the literature (see Glover and Woolsey 1974, Hansen 1979). To reduce the size of the linearized model, Glover (1975) proposed a new strategy to linearize the quadratic terms $x_i x_j$ through the introduction of n unrestricted continuous variables and $4n$ linear inequalities. Adams and Forrester (2005), Adams et al. (2004), Chaovalitwongse et al. (2004), and Sherali and Smith (2007) provided different $O(n)$ linearization approaches. The reformulation-linearization technique (RLT) is an alternative and successful approach to linearize the BQP (see Adams and Sherali 1986, Sherali and Adams 2013). Applying RLT to the special cases of the BQP leads to tight linear relaxations (see, for instance, Adams and Johnson 1994, Hahn et al. 2012, Rostami and Malucelli 2014, 2015).

Another relevant track of research on the BQP is to study the polyhedral structure of the set of feasible solutions to strengthen the LP-based reformulation bounds. One way to construct such polyhedral relaxation is to generate some valid inequalities dynamically by using cutting-plane methods. Padberg (1989) proposed a polytope, called boolean quadric polytope, associated with a linearized integer programming formulation of the unconstrained binary quadratic program and introduced three families of valid and facet-defining inequalities for it. There are several papers devoted to studying the polyhedral structure of the special cases of the BQP (see, for instance, Jünger and Kaibel 2001, Saito et al. 2009, Helmberg et al. 2000, Fischer and Helmberg 2013, Fischer 2014).

Semidefinite programming (SDP) is another popular approach to generate strong relaxations of the BQP. The SDP can be viewed as an extension of linear programming where the nonnegativity constraints are replaced by positive semidefinite constraints on matrix variables. More precisely for any vector $x \in \mathbb{B}^n$ of decision variables, a new matrix $Y = xx^T$ is introduced, which transforms the quadratic function of x into a linear function of Y, and then a "rank one" non-convex constraint $Y = xx^T$ is imposed to the problem. Because of the non-convexity of the rank one constraint, a relaxation of this constraint is considered such that the resulting problem is an SDP. Applications of SDP for different types of BQP problems can be found in Poljak et al. (1995), Helmberg et al. (2000), Gao and Li (2013), Sotirov (2013), and Ferreira et al. (2018).

Quadratic reformulations are alternative approaches that transform a BQP into an equivalent one with either convex or non-convex objective function. The Quadratic convex reformulation

(QCR) is proposed by Billionnet et al. (2009) to convert the non-convex quadratic objective function to an equivalent convex one. In the QCR, the optimal multipliers for the new convex program are found by solving a SDP. Some applications of the QCR can be found in Duives et al. (2013) and Buchheim and Traversi (2018). The idea of quadratic non-convex reformulations is to perturb the objective function with some multipliers to obtain a tighter lower bound. The application of this approach on various special cases of BQP can be found in Carraresi and Malucelli (1992), Rostami and Malucelli (2015), Rostami et al. (2015), and Rostami et al. (2018a).

1.2. Our contributions

Our main scientific contributions are summarized as follows.

- We propose a unifying model for a large class of binary quadratic programs which include a variety of important problems in management science, computer science, transportation, and logistics.
- We exploit the structure of the proposed model to reformulate the BQP_P as a convex MINLP. This problem possesses a special structure which naturally lends itself to decomposition techniques. We show that one can find an alternative MILP reformulation of the BQP_P using a special application of the RLT. Moreover, we analyze the relationship between the two formulations in terms of relaxation strength.
- We apply an outer approximation approach to reformulate the proposed convex MINLP as a MILP. However, due to the size of the resulting MILP, it is not practical to solve it directly using the state-of-the-art solver. Instead, we develop a branch-and-cut algorithm, where the outer approximation cuts are generated on the fly by efficiently solving separation subproblems. Besides, we use some algorithmic features such as multiple outer approximation cuts and a stabilized cut generation scheme to speed up the basic implementation.
- We consider three classes of problems in the literature and show how to represent each as a BQP_P. To evaluate the robustness and efficiency of our solution methods, we perform extensive computational experiments on instances of the quadratic semi-assignment problem, single allocation hub location problem, and test assignment problem. Moreover, for each problem, we compare our results with those obtained by Gurobi using the RLT-based model and to the best-known MILP model in the literature. The results indicate a significant superiority of our solution method.

The remainder of the paper is organized as follows. In Section 2, we present the MINLP, the MILP reformulations and the relationship between their continuous relaxations. In Section 3, we describe the outer approximation based solution method and present some acceleration strategies that improve the convergence and efficiency of the algorithm. The results of extensive computational experiments performed on different problem types are presented in Section 4. Finally, our concluding remarks and possible future works are presented in Section 5.

2. Mixed-integer linear and nonlinear reformulations

In this section, we propose two alternative reformulations for the BQP_P. In Subsection 2.1, we present a MINLP reformulation, while in Subsection 2.2 we give a MILP reformulation based on an application of the RLT. In Subsection 2.3, we analyze the relationship between the two reformulations in terms of the quality of the lower bounds provided.

2.1. A mixed-integer nonlinear programming reformulation

Let us consider the BQP_P. By using (2), we first rewrite the objective function in the following extended form:

$$c^T x + x^T Q x = \sum_{i\in N} c_i x_i + \sum_{i,j\in N} q_{ij} x_i x_j = \sum_{k\in\mathcal{K}} \sum_{i\in I_k} c_i x_i + \sum_{\substack{k,\ell\in\mathcal{K}\\ k\neq\ell}} \sum_{i\in I_k} \sum_{j\in I_\ell} q_{ij} x_i x_j. \tag{3}$$

Notice that for each pairs $i, j \in I_k$, the quadratic expression $x_i x_j = 0$ due to (2). Therefore, the quadratic part of the objective function can be expressed in terms of partitions' interaction costs rather than the individuals variables quadratic costs. To this end, let us define new continuous variables $y^i_{k\ell}$ for any $k, \ell \in \mathcal{K}$, $k \neq \ell$, and any given $i \in I_k$, representing the interaction cost between $i \in I_k$ and the partition I_ℓ. Accordingly, the quadratic cost between each two partitions $k, \ell \in \mathcal{K}$, $k \neq \ell$ is computed as

$$\sum_{i\in I_k} y^i_{k\ell} x_i + \sum_{j\in I_\ell} y^j_{\ell k} x_j. \tag{4}$$

Given the fact that in any feasible solution we only select one variable x from each partition, there must exist $i_k \in I_k$ and $j_\ell \in I_\ell$ with $x_{i_k} = x_{j_\ell} = 1$. Therefore, the quadratic cost between each two partitions $k, \ell \in \mathcal{K}$, $k \neq \ell$ is reduced to the quadratic costs between $i_k \in I_k$ and $j_\ell \in I_\ell$, i.e.,

$$\sum_{i\in I_k} y^i_{k\ell} x_i + \sum_{j\in I_\ell} y^j_{\ell k} x_j = y^{i_k}_{k\ell} + y^{j_\ell}_{\ell k} \quad \text{if} \quad x_{i_k} = x_{j_\ell} = 1. \tag{5}$$

We then propose the following MINLP reformulation for the BQP_P:

$$\begin{aligned} \text{MINLP1:} \quad \min_{x} \quad & \sum_{k\in\mathcal{K}} \sum_{i\in I_k} c_i x_i + \phi(x) \\ \text{s.t.} \quad & x \in X \cap P \cap \mathbb{B}^n, \end{aligned} \tag{6}$$

where, for each $x \in X \cap P \cap \mathbb{B}^n$, $\phi(x)$ is the optimal value function of the parametric optimization problem

$$\phi(x) = \min \quad \sum_{\substack{k,\ell\in\mathcal{K}\\ k\neq\ell}} \Big(\sum_{i\in I_k} y^i_{k\ell} x_i + \sum_{j\in I_\ell} y^j_{\ell k} x_j \Big) \tag{7}$$

$$\text{s.t.} \quad y^i_{k\ell} + y^j_{\ell k} \geq q_{ij} \quad k, \ell \in \mathcal{K}, k \neq \ell,\ i \in I_k, j \in I_\ell. \tag{8}$$

Theorem 1. *Problem MINLP1 is a reformulation of BQP_P.*

Proof. We have to prove that, for any feasible solution of either problem there exists a feasible solution to the other problem with the same objective values.

Consider any solution $\hat{x} \in X \cap P \cap \mathbb{B}^n$. This would be feasible for both problems. We need to show that the objective values of MINLP1 and BQP_P are identical at $\hat{x}$. For each $k, \ell \in \mathcal{K},\ k \neq \ell$, there exist $i_k \in I_k$ and $j_\ell \in I_\ell$ such that $\hat{x}_{i_k} = \hat{x}_{j_\ell} = 1$. Therefore, the value of the objective function of BQP_P is given by

$$\sum_{k \in \mathcal{K}} c_{i_k} + \sum_{\substack{k,\ell \in \mathcal{K} \\ k \neq \ell}} q_{i_k j_\ell}. \tag{9}$$

Let $F = \{y |\ (8)\}$ represent the feasible space of the minimization problem $\phi(x)$ in (7) and (8). If F is not empty, the problem can be either unbounded or bounded for any arbitrary choice of $\hat{x}$. First, we show that F is not empty. For each $k, \ell \in \mathcal{K},\ k < \ell$, if we set

$$y^i_{k\ell} = q_{ij_\ell} \quad i \in I_k, \tag{10}$$

$$y^j_{\ell k} = \max_{i \in I_k} \{q_{ij} - y^i_{k\ell}\} \quad j \in I_\ell, \tag{11}$$

then $y^i_{k\ell} + y^j_{\ell k} \geq q_{ij}$, i.e., $F \neq \emptyset$. Now, we show that $\phi(\hat{x})$ is bounded.

$$\begin{aligned}
\phi(x) = & \min \left\{ \sum_{\substack{k,\ell \in \mathcal{K} \\ k \neq \ell}} \left(\sum_{i \in I_k} y^i_{k\ell} x_i + \sum_{j \in I_\ell} y^j_{\ell k} x_j \right) : (8) \right\} \\
= & \min \left\{ \sum_{\substack{k,\ell \in \mathcal{K} \\ k \neq \ell}} \left(y^{i_k}_{k\ell} x_{i_k} + y^{j_\ell}_{\ell k} x_{j_\ell} \right) : (8) \right\} = \min \left\{ \sum_{\substack{k,\ell \in \mathcal{K} \\ k \neq \ell}} \left(y^{i_k}_{k\ell} + y^{j_\ell}_{\ell k} \right) : (8) \right\},
\end{aligned} \tag{12}$$

where, the second and third equalities follows from the feasibility of $\hat{x}$.

We argue that when the minimum in (12) is attained, $\phi(\hat{x}) = \sum_{\substack{k,\ell \in \mathcal{K} \\ k \neq \ell}} q_{i_k j_\ell}$. Indeed, for $k, \ell \in \mathcal{K},\ k < \ell$ and $i = i_k,\ j = j_\ell$, using (10) and (11), we have $y^{i_k}_{k\ell} = q_{i_k j_\ell}$ and $y^{j_\ell}_{\ell k} = 0$. Therefore, the inequalities (8) are tight for $k, \ell \in \mathcal{K},\ k < \ell$ and $i = i_k,\ j = j_\ell$ and the obtained value in (12) is actually a minimum. Hence, using (12) we have $\phi(\hat{x}) = \sum_{\substack{k,\ell \in \mathcal{K} \\ k \neq \ell}} q_{i_k j_\ell}$ which is bounded. This also proves that MINLP1 and BQP_P have the same objective values in $\hat{x}$.

□

Next, we show how to transform the MINLP1 into a convex MINLP where one minimizes a convex objective over the intersection of a convex region and integrality requirements. In doing so, let us define a new convex function $\psi : X \cap P \to \mathbb{R}^+$ in continuous variables $x \in X \cap P$, described as follows:

$$\psi(x) = \max \sum_{\substack{k,\ell \in \mathcal{K} \\ k \neq \ell}} \left(\sum_{i \in I_k} y^i_{k\ell} x_i + \sum_{j \in I_\ell} y^j_{\ell k} x_j \right) \tag{13}$$

$$\text{s.t.} \quad y^i_{k\ell} + y^j_{\ell k} \leq q_{ij} \quad k, \ell \in \mathcal{K}, k \neq \ell,\ i \in I_k, j \in I_\ell. \tag{14}$$

The following theorem forms the base of our transformation.

Theorem 2. *For any integer solution $\hat{x} \in X \cap P \cap \mathbb{B}^n$, we have $\psi(\hat{x}) = \phi(\hat{x})$.*

Proof. Given the fact that in any feasible solution $\hat{x} \in X \cap P \cap \mathbb{B}^n$ (because of constraints (2)), we only select one variable x from each partition, there must exist one $i_k \in I_k$ with $\hat{x}_{i_k} = 1$ for each $k \in \mathcal{K}$. Therefore,

$$\sum_{\substack{k,\ell \in \mathcal{K} \\ k \neq \ell}} \Big(\sum_{i \in I_k} y^i_{k\ell} x_i + \sum_{j \in I_\ell} y^j_{\ell k} x_j \Big) = \sum_{\substack{k,\ell \in \mathcal{K} \\ k \neq \ell}} \Big(y^{i_k}_{k\ell} + y^{j_\ell}_{\ell k} \Big).$$

Hence, as shown in Theorem 1 we have

$$\phi(\hat{x}) = \min \Big\{ \sum_{\substack{k,\ell \in \mathcal{K} \\ k \neq \ell}} \Big(y^{i_k}_{k\ell} + y^{j_\ell}_{\ell k} \Big) : \; y^i_{k\ell} + y^j_{\ell k} \geq q_{ij} \quad k,\ell \in \mathcal{K}, k \neq \ell, \; i \in I_k, j \in I_\ell \Big\} = \sum_{\substack{k,\ell \in \mathcal{K} \\ k \neq \ell}} q_{i_k j_\ell}.$$

Analogous to (10) and (11), for each $k,\ell \in \mathcal{K}, \; k < \ell$, we define

$$\begin{aligned} y^i_{k\ell} &= q_{ij_\ell} \quad i \in I_k, \\ y^j_{\ell k} &= \min_{i \in I_k} \{ q_{ij} - y^i_{k\ell} \} \quad j \in I_\ell. \end{aligned}$$

Following the same reasoning as in Theorem 1, we have

$$\psi(\hat{x}) = \max \Big\{ \sum_{\substack{k,\ell \in \mathcal{K} \\ k \neq \ell}} \Big(y^{i_k}_{k\ell} + y^{j_\ell}_{\ell k} \Big) : \; y^i_{k\ell} + y^j_{\ell k} \leq q_{ij} \quad k,\ell \in \mathcal{K}, k \neq \ell, \; i \in I_k, j \in I_\ell \Big\} = \sum_{\substack{k,\ell \in \mathcal{K} \\ k \neq \ell}} q_{i_k j_\ell},$$

which completes the proof. □

As a consequence of Theorem 2, we can replace the inner minimization problem in MINLP1 by $\psi(x)$, $x \in X \cap P \cap \mathbb{B}^n$, to obtain the following convex MINLP:

$$\begin{aligned} \text{MINLP2:} \quad \min_{x} \quad & \sum_{k \in \mathcal{K}} \sum_{i \in I_k} c_i x_i + \psi(x) \\ \text{s.t.} \quad & x \in X \cap P \cap \mathbb{B}^n. \end{aligned}$$

Remark 1. Using the results of Theorem 2 and the fact that for $x \in X \cap P$, functions $\phi(x)$ and $\psi(x)$ are concave and convex functions, respectively, one could expect the linearity of these functions. However, the linearity does not hold as functions ϕ and ψ are equal only at binary points (i.e., $X \cap P \cap \mathbb{B}^n \subseteq \{0,1\}^n$). To prove this, let us consider a simple example of BQP_P where, $N =$

$\{1,2,3,4\}, \mathcal{K}=\{1,2\}, I_1=\{1,2\}$, $I_2=\{3,4\}$ and $x\in P$ with $(x_1,x_2,x_3,x_4)=(0.5,0.5,0.75,0.25)$. It is easy to see that $\phi(x)=2.75$, while $\psi(x)=2.5$. This property can also be explained in the context of *integrally convex* functions which fail the discrete separation theorem (see Favati 1990, Murota 1998, 2019, for more details).

2.2. A mixed-integer linear programming reformulation

In this section, we present a MILP formulation for the BQP_P based on a special application of the RLT. The RLT is known for generating tight linear programming relaxations, not only for constructing exact solution algorithms, but also for designing powerful heuristic procedures. However, there is a trade-off between the size of the resulting MILP and the tightness of the LP relaxation. The RLT essentially consists of two steps: (i) a reformulation step in which nonlinear valid inequalities are generated by combining constraints of the original problem, and (ii) a linearization step in which each product term is replaced by a single continuous variable.

In the following, we show that the partitioning constraints P defined in (2) are the only set of constraints needed to construct an RLT-based reformulation of BQP_P. To this end, let us, first, multiply each equation of P in (2) by each variable x_j with $j\in I_\ell$, and $\ell\in\mathcal{K}$ to form equations

$$\sum_{i\in I_k} x_i x_j = x_j \quad k,\ell\in\mathcal{K},\ k\neq\ell, j\in I_\ell, \tag{15}$$

and add them to the problem constraints. Then, for each $k,\ell\in\mathcal{K},\ k\neq\ell, i\in I_k, j\in I_\ell$, we replace the product x_ix_j throughout the objective function and constraints with a single nonnegative and continuous variable w_{ij}. Finally, impose $w_{ij}=w_{ji}$ to result in the following MILP formulation:

$$\begin{aligned}
RLT_P\colon\quad \min\quad & \sum_{k\in\mathcal{K}}\sum_{i\in I_k} c_i x_i + \sum_{\substack{k,\ell\in\mathcal{K}\\ k\neq\ell}}\sum_{i\in I_k}\sum_{j\in I_\ell} q_{ij} w_{ij} \\
\text{s.t.}\quad & x\in X\cap P\cap\mathbb{B}^n \\
& \sum_{i\in I_k} w_{ij} = x_j \quad k,\ell\in\mathcal{K},\ k\neq\ell, j\in I_\ell && (16)\\
& w_{ij}=w_{ji} \quad k,\ell\in\mathcal{K},\ k<\ell, i\in I_k, j\in I_\ell && (17)\\
& w_{ij}\geq 0 \quad k,\ell\in\mathcal{K},\ k\neq\ell, i\in I_k, j\in I_\ell. && (18)
\end{aligned}$$

Theorem 3. *RLT_P is a reformulation of BQP_P.*

Proof. Using constraints (17), one can reduce the number of variables and constraints to obtain a more compact reformulation. The proof follows Billionnet and Elloumi (2001), where the validity of this compact reformulation is shown for the quadratic semi-assignment problem.

□

2.3. A comparison between relaxations strength

We now turn our attention to compare the strength of the LP relaxation of RLT_P with the continuous relaxation of MINLP2. The following theorem formally shows the relationship between these two relaxations.

THEOREM 4. *Let $CRLT_P$ and CMINLP2 represent the continuous relaxations of problems RLT_P and MINLP2, respectively. Then, $CRLT_P$ and CMINLP2 are equivalent.*

Proof. The proof is done in two steps. First, we construct a Lagrangian dual corresponding to the dual problem of the $CRLT_P$. Then, we show the equivalence between the constructed Lagrangian dual problem and the CMINLP2.

Let us consider $CRLT_P$ and write its dual problem as follows:

$$
\begin{aligned}
DCRLT_P: \quad \max_{(\alpha,\pi,\lambda,\mu)} \quad & \sum_{k\in\mathcal{K}} \pi_k + \sum_{r=1}^{m} \alpha_r b_r \\
\text{s.t.} \quad & c_j - \sum_{r=1}^{m} \alpha_r\, a_{rj} + \sum_{\ell\in\mathcal{K}:\ell\neq k} \lambda^j_{k\ell} - \pi_k \geq 0 \quad k\in\mathcal{K}, j\in I_k && (19)\\
& q_{ij} - \lambda^i_{kl} - \mu^{ij}_{k\ell} \geq 0 \quad k,\ell\in\mathcal{K},\ k\neq\ell, i\in I_k, j\in I_\ell && (20)\\
& \mu^{ji}_{\ell k} = -\mu^{ij}_{k\ell} \quad k,\ell\in\mathcal{K}, k<\ell,\ i\in I_k, j\in I_\ell, && (21)
\end{aligned}
$$

where (α,π,λ,μ) are the dual variables corresponding to (1), (2), (16), and (17), respectively.

Note that we do not consider constraints $x\leq 1$ in the $CRLT_P$ as it results from partitioning constraints (2). Moreover, according to constraints (17), variables $\mu^{ij}_{k\ell}$ are defined for all $k,\ell\in\mathcal{K}$, $k<\ell$. However, for ease of exposition, we also define $\mu^{ij}_{k\ell}$ for $k>\ell$ and impose constraints (21).

We apply Lagrangian relaxation to constraints (19) using dual multipliers x to obtain the following Lagrangian dual problem:

$$
\min_{x\geq 0} \quad L(x) = \sum_{k\in\mathcal{K}}\sum_{j\in I_k} c_j x_j + g(x) + f(x) + h(x),
$$

where

$$
\begin{aligned}
LP_\alpha: \quad & g(x) = \max_{\alpha} \quad \sum_{r=1}^{m} \alpha_r (b_r - \sum_{k\in\mathcal{K}}\sum_{j\in I_k} a_{rj} x_j),\\
LP_\pi: \quad & f(x) = \max_{\pi} \quad \sum_{k\in\mathcal{K}} \pi_k (1 - \sum_{j\in I_k} x_j), && (22)\\
LP_{\mu\lambda}: \quad & h(x) = \max_{\lambda,\mu} \quad \sum_{\substack{k,\ell\in\mathcal{K}\\ k\neq\ell}} \sum_{i\in I_k} \lambda^i_{k\ell}\, x_i \\
& \text{s.t.} \quad q_{ij} - \lambda^i_{k\ell} - \mu^{ij}_{k\ell} \geq 0 \quad k,\ell\in\mathcal{K},\ k\neq\ell, i\in I_k, j\in I_\ell && (23)\\
& \qquad \mu^{ji}_{\ell k} = -\mu^{ij}_{k\ell} \quad k,\ell\in\mathcal{K}, k<\ell,\ i\in I_k, j\in I_\ell.
\end{aligned}
$$

The optimal objective value of the Lagrangian dual problem gives the optimal value of $CRLT_P$. Since the original problem is bounded, LP_α and LP_π have optimal objective values equal to zero. More precisely, for the latter subproblem, if $1-\sum_{j\in I_k} x_j \neq 0$, for each $k\in\mathcal{K}$, then the variables π can get very large positive or negative values, which in turn, means that the Lagrangian dual problem would be unbounded. Using the same argument $b_r - \sum_{k\in\mathcal{K}}\sum_{j\in I_k} a_{rj}x_j = 0$ for all $r=1,\ldots,m$. Therefore, x must belong to the set $X\cap P$. Accordingly, the Lagrangian dual is reduced to the following problem:

$$\begin{aligned}\min \quad & L(x) = \sum_{k\in\mathcal{K}}\sum_{i\in I_k} c_i x_i + h(x)\\ \text{s.t.} \quad & x\in X\cap P.\end{aligned}$$

To complete the proof, we need to show that $h(x)=\psi(x)$ for all $x\in X\cap P$. To this end, we prove that for any feasible solution of either inner maximization problems, there exists a feasible solution for the other problem with the same objective value. Let us first consider a feasible solution $\bar{y}$ of problem $\psi(x)$ in (13)-(14). We construct a feasible solution for $LP_{\mu\lambda}$ as follows:

$$\begin{aligned}&\bar{\lambda}^i_{k\ell} = 2\,\bar{y}^i_{k\ell} \qquad k,\ell\in\mathcal{K},\ k\neq\ell, i\in I_k\\ &\bar{\mu}^{ij}_{k\ell} = \begin{cases} 2\bar{y}^j_{\ell k} - q_{ij} & k,\ell\in\mathcal{K},\ k<\ell,\ i\in I_k, j\in I_\ell\\ -2\bar{y}^i_{k\ell} + q_{ij} & k,\ell\in\mathcal{K},\ k>\ell,\ i\in I_k, j\in I_\ell.\end{cases}\end{aligned}$$

It is clear that $\mu^{ji}_{\ell k} = -\mu^{ij}_{k\ell}, k,\ell\in\mathcal{K}, k<\ell,\ i\in I_k, j\in I_\ell$. We now show that $(\bar{\lambda},\bar{\mu})$ also satisfies constraints (23). For given $k,\ell\in\mathcal{K}, k\neq\ell,\ i\in I_k$ and $j\in I_\ell$, constraints (23) at $(\bar{\lambda},\bar{\mu})$ read as

$$\bar{\lambda}^i_{kl} + \bar{\mu}^{ij}_{kl} = \begin{cases} 2\bar{y}^i_{k\ell} + 2\bar{y}^j_{\ell k} - q_{ij} = 2(\bar{y}^i_{k\ell} + \bar{y}^j_{\ell k}) - q_{ij} \le q_{ij} & k<\ell,\\ 2\bar{y}^i_{k\ell} - 2\bar{y}^i_{k\ell} + q_{ij} = q_{ij} \le q_{ij} & k>\ell,\end{cases}$$

where the inequality in the first part follows from the feasibility of $\bar{y}$. Moreover,

$$h(x) = \sum_{\substack{k,\ell\in\mathcal{K}\\ k\neq\ell}}\sum_{i\in I_k}\bar{\lambda}^i_{k\ell}\,x_i = \sum_{\substack{k,\ell\in\mathcal{K}\\ k\neq\ell}}\sum_{i\in I_k} 2\,\bar{y}^i_{k\ell}\,x_i = \sum_{\substack{k,\ell\in\mathcal{K}\\ k\neq\ell}}\Big(\sum_{i\in I_k}\bar{y}^i_{k\ell}x_i + \sum_{i\in I_k}\bar{y}^i_{k\ell}x_i\Big) = \sum_{\substack{k,\ell\in\mathcal{K}\\ k\neq\ell}}\Big(\sum_{i\in I_k}\bar{y}^i_{k\ell}x_i + \sum_{j\in I_\ell}\bar{y}^j_{\ell k}x_j\Big) = \psi(x),$$

where the last equality follows from the fact that $\sum_{\substack{k,\ell\in\mathcal{K}\\ k\neq\ell}}\sum_{i\in I_k}\bar{y}^i_{k\ell}x_i = \sum_{\substack{k,\ell\in\mathcal{K}\\ k\neq\ell}}\sum_{j\in I_\ell}\bar{y}^j_{\ell k}x_j$. This is true because the external sum is over any pair k,ℓ, with k different from ℓ which allows us to exchange the indices' names k and ℓ, and also i and j.

Now, let us consider a feasible solution $(\bar{\lambda},\bar{\mu})$ of $LP_{\mu\lambda}$. We show that there exists a feasible solution $\bar{y}$ for problem $\psi(x)$ in (13)-(14) with $h(x)=\psi(x)$. If we define

$$\bar{y}^i_{k\ell} = 1/2\bar{\lambda}^i_{k\ell} \qquad \text{for all} \quad k,\ell\in\mathcal{K}, k\neq\ell, i\in I_k, \tag{24}$$

then, $\bar{y}$ satisfies constrains (14), i.e.,

$$\bar{y}^i_{k\ell} + \bar{y}^j_{\ell k} = 1/2(\bar{\lambda}^i_{k\ell} + \bar{\lambda}^j_{\ell k}) \leq 1/2(q_{ij} - \bar{\mu}^{ij}_{k\ell}) + 1/2(q_{ij} - \bar{\mu}^{ji}_{\ell k}) = q_{ij} \quad k,\ell \in \mathcal{K}, k \neq \ell,\ i \in I_k, j \in I_\ell,$$

where the last equality follows from the fact that $\bar{\mu}^{ij}_{\ell k} = -\bar{\mu}^{ij}_{k\ell}$ for all $k,\ell \in \mathcal{K}, k < \ell,\ i \in I_k, j \in I_\ell$. Moreover, we have

$$\begin{aligned}
\psi(x) &= \sum_{\substack{k,\ell \in \mathcal{K} \\ k \neq \ell}} \Big(\sum_{i \in I_k} \bar{y}^i_{k\ell} x_i + \sum_{j \in I_\ell} \bar{y}^j_{\ell k} x_j \Big) \\
&= 1/2 \sum_{\substack{k,\ell \in \mathcal{K} \\ k \neq \ell}} \Big(\sum_{i \in I_k} \bar{\lambda}^i_{k\ell} x_i + \sum_{j \in I_\ell} \bar{\lambda}^j_{\ell k} x_j \Big) = 1/2 \sum_{\substack{k,\ell \in \mathcal{K} \\ k \neq \ell}} \sum_{i \in I_k} \bar{\lambda}^i_{k\ell} x_i + 1/2 \sum_{\substack{k,\ell \in \mathcal{K} \\ k \neq \ell}} \sum_{j \in I_\ell} \bar{\lambda}^j_{\ell k} x_j \\
&= \sum_{\substack{k,\ell \in \mathcal{K} \\ k \neq \ell}} \sum_{i \in I_k} \bar{\lambda}^i_{k\ell} x_i = h(x).
\end{aligned}$$

□

Observe that one can develop an alternative RLT reformulation in which in addition to the partitioning constraints P defined in (2) the other constraints in (1) are also multiplied by the original variables x. Although this new reformulation would generally lead to a tighter LP bound, solving the linear relaxation requires longer computing times because of its substantially larger size. We refer the reader to Tables 2 and 3 in Section 4 where even without these additional constraints the LP relaxation of the current RLT_p (which is very tight) takes long computing times for small size instances and could not even be solved in the time limit of two hours for medium to large size instances.

3. Solution approach

In this section, we discuss our approach to solve the reformulation MINLP2. In Section 3.1, we develop an outer approximation algorithm to solve MINLP2. Next, we analyze two ways to improve the convergence and stability of the proposed algorithm in Sections 3.2 and 3.3, respectively.

3.1. An outer approximation algorithm

Let us consider problem MINLP2. By introducing an auxiliary variable η, we rewrite MINLP2 as follows:

$$\begin{aligned}
\text{MINLP3:} \quad \min \quad & \sum_{k \in \mathcal{K}} \sum_{i \in I_k} c_i x_i + \eta \\
\text{s.t.} \quad & \eta \geq \psi(x) \\
& x \in X \cap P \cap \mathbb{B}^n.
\end{aligned}$$

Note that constraints defining x are enough to ensure feasibility, the value $\psi(x)$ is bounded. Moreover, if (x,η) is an optimal solution of MINLP3, then x is optimal for BQP_P. Furthermore, since function $\psi : X \cap P \to \mathbb{R}^+$ is convex and the objective function of MINLP3 is linear, the optimal solution of the problem always lies on the boundary of the convex hull of the feasible set. This allows us to use cutting-plane techniques to solve the problem. More precisely, for any given (possibly non-integer) $\bar{x} \in X \cap P \subseteq [0,1]^n$, since $\psi(x)$ is convex, it can be underestimated by a supporting hyperplane in $\bar{x}$. Let $\bar{s} \in \partial\psi(\bar{x})$ be a subgradient of $\psi(x)$ at $\bar{x}$. Then, following the generalized Benders decomposition of Geoffrion (1972) and outer approximation of Duran and Grossmann (1986), we can linearize the convex function $\psi(x)$ around $\bar{x}$ in a set of points $\mathcal{P} = \{\bar{x}^1, \bar{x}^2, \ldots, \bar{x}^M\}$ to obtain the following master problem:

$$\begin{aligned} MP(\mathcal{P})\colon \quad \min \quad & \sum_{k\in\mathcal{K}}\sum_{i\in I_k} c_i x_i + \eta \\ \text{s.t.} \quad & \eta \geq \psi(\bar{x}) + \bar{s}(x-\bar{x}) \quad \bar{x} \in \mathcal{P} \qquad (25) \\ & x \in X \cap P \cap \mathbb{B}^n. \end{aligned}$$

If $MP(\mathcal{P})$ contains a suitable set of points, then it has the same optimal value as MINLP2 (see Duran and Grossmann 1986, Fletcher and Leyffer 1994, Bonami et al. 2008, for more details). Therefore, the main goal of outer approximation algorithms is to build this equivalent MILP. However, it is not practical to solve $MP(\mathcal{P})$ because one would have first to enumerate all feasible solutions $\bar{x} \in X \cap P \cap \mathbb{B}^n$. Instead, we solve $MP(\mathcal{P})$ as a MILP by a branch-and-cut algorithm, where (25) are generated on the fly as described below. Each term $\psi(x)$ is computed by solving a subproblem. More precisely, for a given solution $\bar{x}$ of $MP(\mathcal{P})$, we solve subproblem $\psi(x)$ with $x = \bar{x}$ to obtain an optimal solution $\bar{y}$. Since constraints (14) are independent of $\bar{x}$, the subgradient for each $i \in N$ is given by

$$\bar{s}_i = \sum_{\substack{\ell\in\mathcal{K}:\\ k\neq\ell}} (2\bar{y}^i_{k\ell}),$$

where $k \in \mathcal{K}$ is the index for which $i \in I_k$. The subgradient cut then can be written as

$$\eta \geq \psi(\bar{x}) + \sum_{i\in N} \bar{s}_i(x_i - \bar{x}_i). \qquad (26)$$

The subgradient cut (26) is added to the master problem as it is identified along the branch-and-cut tree. Starting with an empty set of subgradient cuts at the root node, the linear programming relaxation of $MP(\mathcal{P})$ is solved at each node of the search tree, and the subgradient cut (26) is added if it is violated. Otherwise, the algorithm proceeds by branching on binary variables with non-binary values.

3.2. A revised multicut reformulation and subgradient computation

Given that function $\psi(x)$ defined in (13)-(14) is a separable convex function, we can rewrite it as the sum of the compositions of convex functions $\psi_{kl}(x)$ for each $k,\ell \in \mathcal{K}, k \neq \ell$ where

$$\begin{aligned}\psi_{k\ell}(x) = \quad \max \quad & \sum_{i\in I_k} y^i_{k\ell}\, x_i + \sum_{j\in I_\ell} y^j_{\ell k} x_j \\ \text{s.t.} \quad & y^i_{k\ell} + y^j_{\ell k} \le q_{ij} \quad i\in I_k, j \in I_\ell.\end{aligned} \tag{27}$$

Let $\bar{y}$ be the optimal solution of the above problem and $\bar{s}_{k\ell} = \sum_{i\in I_k} \bar{s}_i$ be a subgradient of $\psi_{k\ell}(x)$ for each $k,\ell \in \mathcal{K}, k \neq \ell$. Then, for each $\bar{x} \in X \cap P \cap \mathbb{B}^n$, the subgradient cut (25) is replaced by

$$\eta_{k\ell} \ge \psi_{k\ell}(\bar{x}) + \bar{s}_{k\ell}(x - \bar{x}) \quad k,\ell \in \mathcal{K}, k \neq \ell. \tag{28}$$

Although the number of the new subgradient cuts are much larger than those in $MP(\mathcal{P})$, our computational experiments indicate that the overall computational time needed by the branch-and-cut algorithm to solve an instance of the problem is significantly shorter.

For each $k,\ell \in \mathcal{K}, k \neq \ell$ and for any $x \in X \cap P$, the subproblem (27) is a linear program and can be solved efficiently by the state-of-the-art solvers. However, we can exploit the structure of the subproblem (27) to obtain subgradient cuts (28) more efficiently than by using an LP solver. Following theorem formally gives the main results.

THEOREM 5. *Given $\bar{x} \in X \cap P \subseteq [0,1]^n$, a feasible solution of subproblem* (27) *for each $k,\ell \in \mathcal{K}, k < \ell$, can be obtained by setting*

$$\bar{y}^i_{k\ell} = \sum_{j\in I_\ell} q_{ij}\bar{x}_j \quad i \in I_k \tag{29}$$

$$\bar{y}^j_{\ell k} = \min_{i\in I_k} \{q_{ij} - \bar{y}^i_{k\ell}\} \quad j \in I_\ell. \tag{30}$$

Moreover, if $\bar{x}$ is integer, then (29) *and* (30) *provide an optimal solution of subproblem* (27).

Proof. According to (29) and (30), we can see that $\bar{y}^i_{k\ell} + \bar{y}^j_{\ell k} \le q_{ij}$ for all $i \in I_k, j \in I_\ell$. Therefore, $\bar{y}$ is a feasible solution of subproblem (27). To prove the optimality for integer $\bar{x}$, we consider the dual of problem (27) with variables u. We can show that $\bar{u}_{ij} = \bar{x}_i\bar{x}_j$ for all $i \in I_k, j \in I_\ell$, is a feasible solution for the dual problem. It is enough to show that the objective value of the primal problem at $\bar{y}$ is equal to the dual objective value at $\bar{u}$. i.e.,

$$\begin{aligned}&\sum_{i\in I_k} \bar{y}^i_{k\ell}\, \bar{x}_i + \sum_{j\in I_\ell} \bar{y}^j_{\ell k}\bar{x}_j \\ =&\sum_{i\in I_k}\sum_{j\in I_\ell} q_{ij}\bar{x}_i\bar{x}_j + \sum_{j\in I_\ell}(q_{a(j)j} - \sum_{r\in I_\ell} q_{a(j)r}\bar{x}_r)\bar{x}_j \\ =&\sum_{i\in I_k}\sum_{j\in I_\ell} q_{ij}\bar{x}_i\bar{x}_j + \sum_{j\in I_\ell} q_{a(j)j}\bar{x}_j - \sum_{j\in I_\ell}\sum_{r\in I_\ell} q_{a(j)r}\bar{x}_r\bar{x}_j\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i\in I_k}\sum_{j\in I_\ell} q_{ij}\bar{x}_i\bar{x}_j + \sum_{j\in I_\ell} q_{a(j)j}\bar{x}_j - \sum_{j\in I_\ell} q_{a(j)j}\bar{x}_j - \sum_{j\in I_\ell}\sum_{r\in I_\ell: r\neq j} q_{a(j)r}\bar{x}_r\bar{x}_j \\
&= \sum_{i\in I_k}\sum_{j\in I_\ell} q_{ij}\bar{x}_i\bar{x}_j \\
&= \sum_{i\in I_k}\sum_{j\in I_\ell} q_{ij}\bar{u}_{ij},
\end{aligned}
$$

where

$$a(j) = \arg\min_{i\in I_k} \left\{ q_{ij} - \bar{y}^i_{k\ell} \right\} \quad j\in I_\ell,$$

and $\sum_{j\in I_\ell}\sum_{r\in I_\ell: r\neq j} q_{a(j)r}\bar{x}_r\bar{x}_j = 0$ as we only select one variable from each partition because of (2). □

Note that for any given fractional solution $\bar{x}$, (29) and (30) only provide a feasible point to the subproblem (27) and, hence, the objective value of the subproblem at this point gives a lower bound on true objective value of subproblem. However, our computational experiments showed that this bound is very tight, which in turn leads to a very efficient strategy to separate the fractional solutions (see Table 1). Therefore, we also use (29) and (30) to compute subgradients and, hence, generating valid cuts at the nodes of the decision tree with fractional solutions $\bar{x}$.

3.3. A stabilized cutting plane

In the branch-and-cut algorithm proposed in Section 3.1, at each node of the search tree and at each cut loop iteration, we generate one or more cuts that are violated by the current solution $\bar{x}$, add them to the current relaxation, reoptimize it, and get a new optimal solution $\bar{x}$ to be cut at the next iteration. The efficiency of the algorithm depends mainly on the number of iterations required. To decrease the number of iteration, we generate stronger cuts via a stabilized approach. This approach uses an interior point of the feasible region of the master problem in contrast to Kelley's cutting plane method, where the solution provided by the master problem is an extreme point (see Kelley 1960). In the spirit of the works Ben-Ameur and Neto (2007) and Fischetti et al. (2016), here we use two points $\bar{x}$ and $\mathring{x}$ to generate a stabilized solution $\tilde{x} = \alpha\,\bar{x} + (1-\alpha)\mathring{x}$, where $\bar{x}$ is the current solution of the relaxed master problem, $\mathring{x}$ is a point that belongs to the relative interior of the convex hull $x \in X \cap P \cap \mathbb{B}^n$ and $0 < \alpha \leq 1$. The new point $\tilde{x}$ is used instead of $\bar{x}$ in the subproblem and the following subgradient cut is added to the current LP

$$\eta \geq \psi(\tilde{x}) + \tilde{s}(x - \tilde{x}), \tag{31}$$

where $\tilde{s}$ is the subgradient of $\psi(x)$ at $\tilde{x}$. In our implementation, $\mathring{x}$ is set to a relative interior of the convex hull $x \in X \cap P \cap \mathbb{B}^n$ at the beginning, it is updated using $\mathring{x}_{new} = 1/2(\bar{x} + \mathring{x}_{old})$ at each cut loop iteration.

4. Applications and computational study

As we mentioned in Section 1, many quadratic binary programming problems can be represented as BQP_P. In this section, we provide an extensive experimental study on a large class of test instances from three practical binary quadratic programming problems: quadratic semi-assignment problem (QSAP), single allocation hub location problem (SAHLP), and test assignment problem (TAP). Beyond the importance of these problems in the modeling of a large range of practical applications, the set of linear constraints defining X has a different structure in each problem, which in turn, provides a challenging ground for our solution methods.

For each problem, we compare our outer approximation based branch-and-cut algorithm (OABC) with both the RLT-based model and the most effective MILP formulation proposed in the literature. For the OABC algorithm, we used the closed-form solution presented in Theorem 5 to compute the subgradients and, hence, the optimality cuts as this decreases the computational times significantly (see Table 1). Moreover, after trying different settings, the best performance of the OABC was achieved when we separated all integer solutions in the tree and only fractional solutions at the root node. As mentioned in the introduction, we also tried to solve the BQP_P directly using CPLEX, Gurobi, and the SDP-based software BiqCrunch (Krislock et al. 2017). However, their overall performances were much worse. Hence, we do not report detailed results for them here.

We implemented our algorithm in C++ with the use of the Gurobi solver as a subroutine. The experiments were performed on a machine running Linux Intel Xeon(R) CPU E3-1270 (2 quad-core CPUs with 3.60 GHz) with 64 gigabytes of RAM. The time limit was set to two hours.

In the following, for each problem we provide a brief description, a short literature review, as well as details about the adopted benchmark instances and results.

4.1. Single Allocation p-Hub Median Problem

Hub location problems are strategic planning problems that have been studied for almost 30 years O'Kelly (1987), Alumur and Kara (2008), Campbell and O'Kelly (2012). The problem consists of organizing the mutual exchange of flows among a broad set of depots by choosing a set of hubs out of the set of possible locations and assigning each flow to a path from source to sink being processed at a small number of hubs in between. Hub nodes are used to sort, consolidate, and redistribute flows and their main purpose is to realize economies of scale: while the construction and operation of hubs and the resulting detours lead to extra costs, the bundling of flows decreases costs. The economies of scale are usually modeled as being proportional to the transport volume, defined by multiplication with a discount factor $\alpha \in [0,1]$. The resulting trade-off has to be optimized. Typical applications of hub-based networks arise in airline, postal, cargo, telecommunication, and public transportation services (see, for instance, Jaillet et al. 1996, Ernst and Krishnamoorthy 1996, Taylor et al. 1995, Klincewicz 1998, Nickel et al. 2001).

Consider a complete directed graph $G = (V, A)$, where V is a set of nodes (representing the origins, destinations, and possible hub locations), and A is the edge set. Let $w_{k\ell}$ be the amount of flow to be transported from node k to node ℓ. We denote by $O_k = \sum_{\ell \in V} w_{k\ell}$ and $D_k = \sum_{\ell \in V} w_{\ell k}$ the total outgoing flow from node k and the total incoming flow to node k, respectively. For each $i \in V$, let f_i represent the fixed set-up cost of a hub located at node i. The cost per unit of flow for each path $k - i - j - \ell$ from an origin node k to a destination node ℓ that passes hubs i and j respectively, is $\chi d_{ki} + \alpha d_{ij} + \delta d_{j\ell}$, where χ, α, and δ are the nonnegative collection, transfer, and distribution costs respectively, and $d_{k\ell}$ represents the distance between nodes k and ℓ. The Single Allocation p-Hub Median Problem (SApHMP) consists of selecting p nodes as hubs and assigning the remaining nodes to these hubs such that each non-hub node is assigned to exactly one hub node with the minimum overall cost.

O'Kelly (1987) proposed the first quadratic integer programming formulation for the SApHMP. Since then, many exact and heuristic algorithms have been proposed in the literature, dealing with locating both a fixed and a variable number of hubs (e.g., Campbell 1994, Ernst and Krishnamoorthy 1996, Skorin-Kapov et al. 1996, Ilić et al. 2010, Rostami et al. 2016).

To model the problem, we define binary variables x_{ik} indicating whether a source/sink $k \in V$ is allocated to a hub located at $i \in V$. In particular, the variables x_{ii} are used to indicate whether i becomes a hub. For ease of presentation, we set

$$
\begin{aligned}
c_{ik} &:= d_{ki} \, (\chi \, O_k + \delta \, D_k) \\
q_{ikj\ell} &= \alpha \, w_{k\ell} \, d_{ij}.
\end{aligned}
$$

The SApHMP can then be formulated as follows:

$$
\begin{aligned}
\min \quad & \sum_{i,k \in V} c_{ik} x_{ik} + \sum_{i,j} \sum_{k,\ell} q_{ikj\ell} \, x_{ik} x_{j\ell} \\
\text{s.t.} \quad & \textstyle\sum_{i \in V} x_{ik} = 1 \quad (k \in V) && (32) \\
& x_{ik} \leq x_{ii} \quad (i, k \in V) && (33) \\
& \textstyle\sum_{k \in V} x_{ii} = p && (34) \\
& x_{ik} \in \{0, 1\} \quad (i, k \in V),
\end{aligned}
$$

where the objective function measures the total transportation costs consisting of the collection and distribution costs of nonhub-hub and hub-nonhub connections, the hub-hub transfer costs. Constraints (32) force every node to be allocated to precisely one hub node. Constraints (33) state that k can only be allocated to node i if node i is chosen as a hub. Constraint (34) enforces the number of open hubs to be p.

Due to the quadratic nature of the problem, many attempts have been made in the literature to linearize the objective function. Skorin-Kapov et al. (1996) and Ernst and Krishnamoorthy (1996) proposed two main MILP formulations for the problem that are based on a path and a flow representation, respectively. The path-based formulation of Skorin-Kapov et al. (1996), which can also be obtained by applying the RLT to constraints (32), has $O(|V|^4)$ variables and $O(|V|^3)$ constraints and its LP relaxation provides tight lower bounds for some well-known test instances in the literature. However, due to a large number of variables and constraints, it is only able to solve instances with small to medium sizes. The flow-based formulation (F-MILP) use $O(|V|^3)$ and $O(|V|^2)$ additional variables and constraints, respectively, to linearize the original formulation. Among the existing formulations for the SApHMP, the F-MILP is the one that is often considered the most effective in the literature.

In order to use the outer approximation our approach described in Section 3, it is enough to define

$$N = V \times V, \quad K = V, \quad \text{and,}$$
$$I_k = \{(i,k) | i \in V\} \quad k \in V.$$

Because of the single allocation constraints (32), we have $I_k \cap I_\ell = \emptyset$ for all $k, \ell \in K$ with $k \neq \ell$. Therefore, the results of Theorem 1 and the developed solution method can be used to solve the problem.

4.1.1. Results In this section, we provide an extensive experimental evaluation of our approach based on some well-know benchmark instances. First, we show the advantage of the closed-form solution provided in Theorem 5. We then compare our algorithm computationally to Gurobi applied to both the RLT-based and the F-MILP formulations. For numerical tests, we used the well-known Australian Post (AP) set of instances which is the most commonly used in hub location literature. It consists of postal flow and Euclidean distances between 200 districts in an Australian city. The AP dataset was introduced by Ernst and Krishnamoorthy (1996) and it is available in the OR library (see, Beasley (1990).) We have selected small to medium size instances with $|V| = 25, 40, 50, 60, 70, 75$ and medium to large size instances with $|V| = 90, 100, 125,$ and 150 nodes. The transportation cost parameters are chosen as usual: $\alpha = 0.75$, $\chi = 3.0$, and $\delta = 2.0$.

Table 1 shows the performance of the OABC algorithm where subproblems (27) are solved using Theorem 5 or directly by the solver. We denote the latter version of the algorithm by OABC0.[1]

[1] Note that the LP used to solve subproblems (27) is warm started from the old dual basis by the dual simplex method to guarantee an efficient solution.

We compare the results in terms of the solution time (t) and the number of subgradient cuts. For each instance with $|V|$ nodes and p hubs, column under t/t_0 represents the ratio of the total time of the OABC to the total time of the OABC0, while column under cut/cut_0 shows the ratio of the total number of subgradient cuts in the OABC to number of subgradient cuts in the OABC0. We only report the results for those instances for which both versions of the OABC algorithm were able to solve them within the 2 hours time limit (see Tables 2 and 3 for more details on the performance of the OABC with the closed-form solution). As we can see from the column cut/cut_0, the average number of generated optimality cuts by the OABC is higher than those generated by OABC0. This indicates that solving the subproblems by the exact method generates stronger cuts. However, looking at t/t_0, we see that the OABC algorithm with the help of the closed form solution outperforms the OABC0 significantly. On average, the OABC takes 6% (in the best case) to 21% (in the worst case) of the total time of the OABC0 to obtain the optimal solutions. It is worth to mention that, while OABC0 is unable to solve many instances with $|V| \geq 100$ within the time limit of 2 hours, the OABC could solve all instances in less than one hour (see Tables 2 and 3). Although it is conceivable that subproblems (27) could be optimally solved efficiently by specialized solvers, due to the effective performances of OABC (where subproblems (27) are solved using Theorem 5), in the following we concentrate on this version of the algorithm.

Table 1 Performance of the OABC algorithm with and without the results of Theorem 5

	$p=2$		$p=3$		$p=4$		$p=5$	
Instance	t/t_0	cut/cut_0	t/t_0	cut/cut_0	t/t_0	cut/cut_0	t/t_0	cut/cut_0
25	0.03	0.73	0.04	1.04	0.04	0.85	0.03	0.85
40	0.05	1.09	0.09	1.77	0.09	1.43	0.14	1.71
50	0.04	0.82	0.06	1.00	0.12	1.65	0.12	0.48
60	0.06	1.32	0.07	1.28	0.19	0.78	0.19	2.14
70	0.05	0.93	0.09	1.60	0.14	1.80	0.25	0.63
75	0.05	0.92	0.07	1.29	0.12	1.46	0.26	1.57
90	0.05	0.97	0.08	0.58	0.11	1.19	0.45	0.97
100	0.11	1.19	0.11	1.33	0.16	2.07	-	-
125	0.07	1.18	0.06	2.74	-	-	-	-
150	0.10	0.38	-	-	-	-	-	-
Ave.	0.06	0.95	0.07	1.40	0.12	1.40	0.21	1.19

Tables 2 and 3 report the results for the SApHMP. In each table, the first three columns give, for each instance, the number of nodes ($|V|$), the number of hubs (p), and the optimal objective value (Opt.). The next columns present the results of Gurobi applied to RLT-based and the flow-based models, respectively, and our OABC algorithm. For each algorithm, we present the total number of nodes enumerated in the search tree (nodes), the total required time, in seconds, to solve the

Table 2 Comparing the RLT-based and the flow-based models with our OABC algorithm on small to medium size instances of the AP dataset for SApHMP proposed in Ernst and Krishnamoorthy (1996)

Instance			Gurobi + RLT				Gurobi + F-MILP			OABC algorithm			t(G/O)	
$\|V\|$	p	Opt.	$g_{lp}(\%)$	nodes	$g(\%)$	time(s)	nodes	$g(\%)$	time(s)	nodes	$g(\%)$	time(s)	RLT	F-MILP
25	2	175542	0.0	0	0.0	9.2	0	0.0	0.8	2	0.0	0.7	**13.1**	**1.1**
25	3	155256	0.0	0	0.0	9.2	1	0.0	4.2	7	0.0	0.8	**12.3**	**5.7**
25	4	139197	0.0	0	0.0	11.5	29	0.0	3.5	5	0.0	0.9	**12.2**	**3.7**
25	5	123574	0.0	0	0.0	7.5	5	0.0	3.7	7	0.0	0.9	**8.2**	**4.0**
40	2	177472	0.0	0	0.0	108	0	0.0	4.6	3	0.0	2.8	**39.3**	**1.7**
40	3	158831	0.0	0	0.0	103.4	4	0.0	16.2	8	0.0	7.8	**13.3**	**2.1**
40	4	143969	0.0	0	0.0	89.5	70	0.0	60.9	29	0.0	11.2	**8.0**	**5.4**
40	5	134265	0.0	0	0.0	93.9	36	0.0	50.0	26	0.0	13.8	**6.8**	**3.6**
50	2	178484	0.0	0	0.0	433.2	1	0.0	17.3	3	0.0	7.6	**56.7**	**2.3**
50	3	158570	0.0	0	0.0	366.1	0	0.0	23.6	4	0.0	8.1	**45.1**	**2.9**
50	4	143378	0.0	0	0.0	295.8	0	0.0	24.6	7	0.0	9	**33.1**	**2.7**
50	5	132367	0.0	0	0.0	277.9	3	0.0	30.3	41	0.0	8	**34.6**	**3.8**
60	2	179920	0.0	0	0.0	1454.2	0	0.0	47.6	3	0.0	15	**96.8**	**3.2**
60	3	160339	0.0	0	0.0	1232.2	4	0.0	101.3	16	0.0	36.4	**33.9**	**2.8**
60	4	144720	0.0	0	0.0	1129.4	17	0.0	148.2	12	0.0	23.4	**48.2**	**6.3**
60	5	132850	0.0	0	0.0	902.4	73	0.0	222.7	68	0.0	35.6	**25.3**	**6.3**
70	2	180093	0.0	0	0.0	3371.8	0	0.0	105.3	3	0.0	20.5	**164.8**	**5.2**
70	3	160933	0.0	0	0.0	3130.6	11	0.0	380.8	5	0.0	32.1	**97.7**	**11.9**
70	4	145620	0.0	0	0.0	2477.4	19	0.0	328.6	9	0.0	39.3	**63.1**	**8.4**
70	5	135835	0.0	0	0.0	2913.6	63	0.0	706.4	22	0.0	86.8	**33.6**	**8.1**
75	2	180119	0.0	0	0.0	5156.9	0	0.0	183.3	3	0.0	26.6	**193.9**	**6.9**
75	3	161057	0.0	0	0.0	4715.5	4	0.0	355.6	5	0.0	44.7	**105.5**	**8.0**
75	4	145734	0.0	0	0.0	4231.7	21	0.0	367.9	11	0.0	56.5	**74.9**	**6.5**
75	5	136011	0.0	0	0.0	4892.7	98	0.0	667.5	36	0.0	167.8	**29.2**	**4.0**

problem (time), and the final percentage optimality gap ($g(\%)$). For the RLT-based model, we also report the percentage LP relaxation gap ($g_{lp}(\%)$). In the last column, we report the corresponding ratios ($t(G/O)$) between the running times of Gurobi applied to the RLT-based model or the flow-based model and the OABC algorithm. A value greater than one means an improvement in terms of computing time. We used the time limit as running time for non-solved instances. Since the RLT-based model on medium to large size instances could not be solved in the time limit of two hours, in Table 3, we only report the results of Gurobi applied to the flow-based models and our OABC algorithm. Note that, on the medium to large size instances even the LP relaxation of the RLT-based model could not be solved within the time limit.

Tables 2 and 3 reveal several interesting facts. All the algorithms could solve the small to medium size instances within the time limit. Inspecting the column $g_{lp}(\%)$ of the RLT-based model, we can observe the tightness of the LP relaxation of the RLT-based model, i.e., $g_{lp} = 0$ for all the instances. Regarding the computational times, the flow-based model is solved much faster than the RLT-based model. However, by inspecting the last two columns of the table, we can see the

Table 3 Comparing the flow-based models with our OABC algorithm on medium to large size instances of the AP dataset for SApHMP proposed in Ernst and Krishnamoorthy (1996)

Instance			Gurobi + F-MILP				OABC algorithm			
$\lvert V\rvert$	p	Opt.	Ub	nodes	$g(\%)$	time(s)	nodes	$g(\%)$	time(s)	t(G/O)
90	2	179822	179822	0	0.0	629.3	3	0.0	76.6	**8.2**
90	3	160437	160437	0	0.0	799.3	12	0.0	143.5	**5.6**
90	4	145134	145134	25	0.0	862.5	21	0.0	215.6	**4.0**
90	5	135808	135808	152	0.4	7200.0	27	0.0	268.4	$>$ **26.8**
100	2	180224	180224	5	0.0	884.8	3	0.0	103.6	**8.5**
100	3	160847	160847	5	0.0	1381.7	13	0.0	283.9	**4.9**
100	4	145897	145897	27	0.0	1822.0	9	0.0	191.4	**9.5**
100	5	136929	136929	200	0.1	7200.0	27	0.0	386.7	$>$ **18.6**
125	2	180372	180372	0	0.0	4472.1	3	0.0	220.6	**20.3**
125	3	161117	161117	0	0.0	5317.2	10	0.0	460.8	**11.5**
125	4	146173	148355	1	1.7	7200.0	13	0.0	431.2	$>$ **16.7**
125	5	137176	141563	0	4.6	7200.0	27	0.0	933.1	$>$ **7.7**
150	2	180899	-	0	-	7200.0	4	0.0	487.5	$>$ **14.8**
150	3	161490	-	0	-	7200.0	11	0.0	1578.4	$>$ **4.6**
150	4	146521	-	0	-	7200.0	15	0.0	1191.8	$>$ **6.0**
150	5	137426	-	0	-	7200.0	41	0.0	3345.5	$>$ **2.2**

* Not that even the LP relaxation of the RLT-based model couldn't be solved within the time limit.

superiority of our OABC algorithm over the other two approaches. The OABC algorithm is on average 52 and 5 times faster than the RLT-based and the flow-based models, respectively. The performance of our OABC algorithm on the medium to large size instances is very promising. When both the flow-based and the OABC algorithm can solve an instance to optimality within the time limit, our OABC algorithm is about 9 times faster. Moreover, our algorithm can solve to optimality all instances whereas Gurobi applied to the flow-based model, solves only 8 instances out of 16. Note that for the largest instances, even the LP relaxation of the flow-based model could not be solved within the time limit.

4.2. Quadratic semi-assignment problem

The QSAP is a binary quadratic optimization problem that is NP-hard as shown in Sahni and Gonzalez (1976) and has many applications in modeling real-life problems. It has been applied to model clustering and partitioning problems (Hansen and Lih 1992), equipartition problems (Simeone 1986), schedule synchronization problems (Malucelli 1996), and some scheduling problems (see, for instance, Chrétienne 1989). More recently, the QSAP has been applied to model the data association problem in multiple object tracking (see for instance, Trinh et al. 2012, Dehghan and Shah 2018, Henschel et al. 2018). In multiple object tracking, racking-by-detection is a standard paradigm that splits the problem into two steps: object detection and data association. In a cluttered environment, where occlusions are common, the received measurements may not all arise from the targets of interest. Some of them may be from clutter or false alarm. As a result, there always exist ambiguities in the association between the previous known targets and measurements. The goal of the data association step is then to fill in the gaps between detections and filter out false positives.

Here, we present the QSAP in a general setting. We are given two sets $V = \{1,\dots,p\}$ and $M = \{1,\dots,m\}$ of p objects and m locations, respectively. Let c_{ik} represent the cost of assigning object $k \in V$ to location $i \in M$ and $q_{ijk\ell}$ denote the cost of assigning object k to location i and object l to location j, simultaneously. The quadratic semi-assignment problem seeks to assign each object to exactly one location with minimum overall cost. Here, we define the binary variable x_{ik} equals 1 if object $k \in V$ is assigned to location $i \in M$, and 0 otherwise to obtain the following binary quadratic formulation:

$$
\begin{aligned}
\text{QSAP:} \quad \min \quad & \sum_{k\in V}\sum_{i\in M} c_{ik}x_{ik} + \sum_{k,\ell\in V}\sum_{i,j\in M} q_{ijk\ell}\, x_{ik}x_{j\ell} \\
\text{s.t.} \quad & \textstyle\sum_{i\in M} x_{ik} = 1 \quad (k\in V) \\
& x_{ik} \in \{0,1\} \quad (i\in M, k\in V).
\end{aligned} \tag{35}
$$

To solve the QSAP, Malucelli and Pretolani (1994, 1995) propose lower bounds by decomposing it into reducible graphs within a Lagrangian dual framework. The RLT is a well-known approach used to solve the problem in the literature. Schüle et al. (2009) investigate different levels of RLT to obtain the convex hull of feasible solutions. Billionnet and Elloumi (2001) show that the best reduction of the QSAP using a quadratic pseudo-boolean function with nonnegative coefficients is the level-1 RLT. Saito (2006) computationally demonstrates that the level-1 RLT formulation gives integer optimal solutions on many instances derived from the AP dataset for hub location problem.

In order to use our approach described in Section 3, we define

$$
\begin{aligned}
& N = M \times V, \quad K = V \quad \text{and,} \\
& I_k = \{(i,k) | i \in M\} \quad k \in K.
\end{aligned}
$$

Because of constraints (35) it is easy to see that $I_k \cap I_\ell = \emptyset$ for all $k,\ell \in K$ with $k \neq \ell$. Therefore, Theorem 1 can be applied to reformulate the problem and hence our solution method can be used to solve it.

Table 4 Comparing the RLT-based model with our OABC algorithm on a set of randomly generated QSAP instances

Instance				Gurobi + RLT				OABC algorithm			
$\|V\|$	$\|M\|$	Class	Opt.	g_{lp}(%)	nodes	g(%)	time(s)	nodes	g(%)	time(s)	t(G/O)
35	15	C50	85032.4	2.1	27	0.0	33.8	27	0.0	6.9	**4.9**
35	15	C25	68581.2	2.5	23	0.0	49.3	69	0.0	18.3	**2.7**
35	15	C10	56101.4	3.1	17	0.0	53.6	20	0.0	12.7	**4.2**
35	15	C01	48225.7	3.7	7	0.0	28.6	36	0.0	13.3	**2.2**
53	22	C50	88213.3	4.5	11	0.0	69.1	34	0.0	22.9	**3.0**
53	22	C25	66455.8	3.8	55	0.0	119.2	32	0.0	22.5	**5.3**
53	22	C10	51874.3	4.8	31	0.0	156.4	29	0.0	23.4	**6.7**
53	22	C01	42645.2	5.5	7	0.0	143.9	17	0.0	21.8	**6.6**
70	30	C50	103917.0	6.8	0	0.0	79.1	16	0.0	31.6	**2.5**
70	30	C25	80506.7	5.6	0	0.0	162.6	34	0.0	88.3	**1.8**
70	30	C10	63440.6	12.9	16	0.0	569.4	50	0.0	141.7	**4.0**
70	30	C01	50218.7	18.1	18	0.0	1023.2	37	0.0	140.9	**7.3**
88	37	C50	107483.0	5.1	45	0.0	769.6	61	0.0	143.7	**5.4**
88	37	C25	80387.2	2.6	49	0.0	2550.1	48	0.0	276.9	**9.2**
88	37	C10	59100.0	7.1	42	0.0	3779.4	60	0.0	320.2	**11.8**
88	37	C01	43671.2	8.3	38	0.0	2468.9	41	0.0	433.6	**5.7**
105	45	C50	107260.0	2.4	0	0.0	2768.2	31	0.0	205.6	**13.5**
105	45	C25	75695.8	2.7	44	0.0	6858.7	97	0.0	730.0	**9.4**
105	45	C10	50548.7	2.8	17	0.0	6673.3	65	0.0	1077.2	**6.2**
105	45	C01	32950.1	3.8	38	2.9	7200.0	30	0.0	913.5	$>$ **7.9**

4.2.1. Results As we mentions earlier, the RLT-based model is considered the most effective model for the QSAP in the literature. Therefore, in this section, we compare the results of Gurobi applied to the RLT-based model and the OABC algorithm. To evaluate the performance of the algorithms, following Saito (2006), we first considered the AP dataset of hub location problems. However, it turned out that these are very easy instances for both the RLT model and the OABC algorithm; both methods could solve all instances with $|V| = 200$ in less than one minute though the OABC faster. Therefore, to find challenging instances, we randomly generated instances with different number of objects and locations. We considered complete graphs with size ranging from $n = 50$ to $n = 150$ nodes (generated randomly in a 100×100 square) and partitioned the node set of each graph into two subsets V and M with $|V| = 0.7n$ and $|M| = n - |V|$. For each two items $k, \ell \in V$ and each two locations $i, j \in M$, we set $q_{ijk\ell} = F_{k\ell} D_{ij}$, where F and D are the flow and distance matrices associated with each graph.The flow matrix, F, is generated uniformly at random from $\{l, \ldots, 100\}$, where $l \in \{1, 10, 25, 50\}$, while the distance matrix, D, is the Euclidean distance matrix. In the spirit of the works Malucelli and Pretolani (1994, 1995), we defined D_{jj} for $j \in M$, to be the 50 percent of the average graph distance to prevent assigning all the objects to the same location. The linear cost c_{ik} for each object $k \in V$ and location $i \in M$ is set to $\delta_{ik} D_{ik}$, where δ_{ik} is generated uniformly at random from $\{l, \ldots, 100\}$. We considered four different values for $l \in \{1, 10, 25, 50\}$ to vary the contribution of the linear costs, which in turn, resulted in four different classes of instances, i.e., $C01, C10, C25$, and $C100$.

Table 4 reports the results. The first four columns give, for each instance, the number of nodes ($|V|$), the number of locations ($|M|$), the instance's class name (Class), and the optimal objective value (Opt.) obtained by the OABC algorithm. Columns five to nine present the results of Gurobi applied to the RLT-based model, while columns ten to twelves give the results of our OABC algorithm. For each algorithm, we present the total number of nodes enumerated in the search tree (nodes), the total required time, in seconds, to solve the problem (time), and the final percentage optimality gap ($g(\%)$). For the RLT-based model, and for each instance, we also report the percentage LP relaxation gap ($g_{lp}(\%)$). In the last column, we report the corresponding ratios ($t(G/O)$) between the running times of Gurobi applied to the RLT-based model and the OABC algorithm where a value greater than one means an improvement in terms of computing time. We used the time limit as running time for non-solved instances.

As we can observe from the table, the OABC algorithm outperforms the solver significantly in terms of overall computing time; when both approaches can solve an instance to optimality within the time limit, the OABC algorithm is about six times faster. Moreover, the OABC algorithm can solve all instances to optimality, while for the largest instances of the class $C01$, Gurobi reaches the time limit of two hours with optimality gap of 2.9 %.

4.3. Test-assignment

Consider the problem of assigning the test variants of a written exam to the desks of a classroom in such a way that desks that are close-by receive different variants. This problem is a generalized version of the vertex coloring problem (see Malaguti and Toth (2010)) and is defined as follows. We are given an undirected graph $G = (V, A)$ with a set of nodes V and set of edges A with positive weights w associated with edges, and a set of available colors H. For each pair of colors $i, j \in H$, we have a positive weight f_{ij} that represents the similarity of the two colors. If node k receives color i and node ℓ receives color j, the vicinity of the edge-color assignment is $w_{k\ell} f_{ij}$. In general, the students will not completely fill the classroom, and there will be p empty desks, thus only $|V| - p$ nodes of G must be colored. By defining binary variables x_{ik} taking value 1 if node k gets color i and 0 otherwise, the problem is formulated as the following BQP:

$$\text{TAP:} \quad \min \sum_{(k,\ell) \in A} \sum_{i,j \in H} w_{k\ell} f_{ij} x_{ik} x_{j\ell}$$

$$\text{s.t.} \quad \sum_{i\in H} x_{ik} \leq 1 \quad (k \in V) \tag{36}$$

$$\sum_{k\in V}\sum_{i\in H} x_{ik} = |V| - p \tag{37}$$

$$x_{ik} \in \{0,1\} \quad (i \in V, k \in H),$$

where constraints (36) restrict each vertex to receive at most one color, and constraint (37) states that $|V|-p$ vertices must be colored (i.e., $|V|-p$ students must be seated). Following Duives et al. (2013), we can define a dummy color 0 given to the p uncolored nodes with $f_{0i}=0$, $i \in H$, and replace constraints (36) and (37) by the following constraints:

$$\sum_{i\in H} x_{ik} = 1 \quad (k \in V) \tag{38}$$

$$\sum_{i\in V} x_{i0} = p.$$

Duives et al. (2013) apply a general-purpose solver to some convex reformulations of the problem and develop a Tabu Search algorithm to find feasible solutions. To find a convex BQP, the authors use the smallest eigenvalue technique (see, for instance, Hammer and Rubin 1970) and the Quadratic Convex Reformulation (QCR) method of Billionnet et al. (2009).

To represent the TAP as the BQP_P, we define

$$N = H \times V, \quad K = V, \quad \text{and,}$$

$$I_k = \{(i,k) \mid i \in H\} \quad k \in V.$$

Because of constraints (38), the results of Theorem 1 and our solution method can be used to solve the problem.

4.3.1. Results To evaluate our proposed method, we used the test instances introduced in Duives et al. (2013). We selected instances derived from 2 real classrooms, with 20 and 49 desks, used for written exams in the Engineering Faculty of the University of Bologna. A graph $G=(V,A)$ is associated with each classroom where the node set V represents the desks, and A is the set of links between the desks. For every graph, in addition to the dummy color (0), three sets of available colors with sizes $2,3$ and 4 are considered. The number of empty desks (nodes that must receive color 0) is selected from the sets $\{0,5,10\}$ and $\{0,10,20\}$ for the classrooms with 20 and 49 desks, respectively.

Table 5 compares the OABC algorithm with the RLT-based model. The first four columns give, for each instance, the number of desks ($|V|$), the number of colors (nc), the number of empty desks (nuc), and the best-known solution value (BSV) from Duives et al. (2013). Columns five to nine present the results of Gurobi applied to the RLT-based model, while columns ten to thirteen give

Table 5 Comparing the RLT-based model with the OABC algorithm on real-life TAP instances presented in Duives et al. (2013)

Instance				Gurobi + RLT					OABC algorithm				
$\|V\|$	nc	nuc	BSV	$g_{lp}(\%)$	Ub	$g(\%)$	nodes	time(s)	Ub	$g(\%)$	nodes	time(s)	$t(G/O)$
20	2	0	20.90	21.5	20.90	0.0	184	0.1	20.90	0.0	117	0.1	**1.0**
20	2	5	7.95	20.7	7.95	0.0	782	1.8	7.95	0.0	2261	0.5	**3.6**
20	2	10	1.85	100.0	1.85	0.0	144	0.9	1.85	0.0	1126	0.1	**9.0**
20	3	0	15.15	30.7	15.15	0.0	1223	1.1	15.15	0.0	1046	0.1	**11.0**
20	3	5	5.58	25.6	5.58	0.0	2030	4.8	5.58	0.0	2182	0.9	**5.3**
20	3	10	1.22	100.0	1.22*	0.0	147	1.0	1.22*	0.0	406	0.1	**10.0**
20	4	0	11.95	31.3	11.95	0.0	2659	8.1	11.95	0.0	3095	3.0	**2.7**
20	4	5	3.98	20.8	3.98*	0.0	3762	8.5	3.98*	0.0	4474	2.5	**3.4**
20	4	10	0.93	100.0	0.93*	0.0	147	1.3	0.93*	0.0	1102	0.3	**4.3**
47	2	0	72.60	24.5	72.60	0.0	382219	335.4	72.60	0.0	67506	30.2	**11.1**
47	2	10	35.45	27.0	35.45	0.0	49738	1554.4	35.45	0.0	3638811	1294.2	**1.2**
47	2	20	12.65	54.7	12.65	0.0	63917	1493.1	12.65	0.0	94384	74.6	**20.0**
47	3	0	53.72	32.7	53.83	26.5	117888	7200.0	53.72*	0.0	3106567	1482.1	$>$ **4.9**
47	3	10	24.84	31.2	24.85	9.7	35973	7200.0	24.84	3.6	2247011	7200.0	**1.0**
47	3	20	8.52	60.2	8.55	10.7	40265	7200.0	8.52*	0.0	564665	532.9	$>$ **13.5**
47	4	0	43.88	37.5	43.98	20.4	36751	7200.0	43.93	13.9	900948	7200.0	**1.0**
47	4	10	19.05	32.1	19.63	25.8	29888	7200.0	19.37	17.4	1119366	7200.0	**1.0**
47	4	20	6.38	59.4	6.38	15.5	19707	7200.0	6.33*	0.0	1847924	1741.1	$>$ **4.1**

* Previously unsolved TAP instances in Duives et al. (2013) proven to be optimal solutions by our solution approaches.

the results of our OABC algorithm. For each algorithm, we present the best feasible solution value (Ub), the total number of nodes enumerated in the search tree (nodes), the total required time, in seconds, to solve the problem (time), and the final percentage optimality gap ($g(\%)$). For the RLT-based model, and for each instance, we also report the percentage LP relaxation gap ($g_{lp}(\%)$). In the last column, we report the corresponding ratios ($t(G/O)$) between the running times of Gurobi applied to the RLT-based model and the OABC algorithm where a value greater than one means an improvement in terms of computing time. We used the time limit as running time for non-solved instances.

As we can observe from the table, the OABC algorithm is able to solve 15 out of 18 instances, whereas Gurobi applied to the RLT-based model can solve 12 instances. When both approaches can solve an instance to optimality within the time limit, the OABC is about 7 times faster. Instances marked by asterisks are previously unsolved instances which we could solve to optimality within the two hours time limit. Note that, the OABC algorithm is the only approach that can solve all the six unsolved instances to optimality while, the Gurobi applied to the RLT-based model can only solve three instances with $|V| = 20$.

5. Conclusions

In this paper, we have studied a class of binary quadratic programming problems that arise in many real-life optimization problems. We have proposed a convex mixed-integer nonlinear program reformulation as well as a mixed-integer linear programming reformulation and analyzed their relaxation strength. Moreover, we have developed a branch-and-cut algorithm to solve the convex mixed-integer nonlinear reformulation, where at each node of the search tree, efficiently solvable subproblems are considered to generate some outer approximation cuts. To evaluate the robustness and efficiency of our solution method, we performed extensive computational experiments on different types of problems from the literature. In particular, we applied our solution approach on instances of quadratic semi-assignment problem, single allocation hub location problem, and test assignment problem and compare our results with the results obtained from commercial solvers applied to RLT-based models as well as to some well-known MILP formulations from the literature. The overall results indicate a significant superiority of our solution method.

Acknowledgments

The authors would like to thank Pierre Bonami for his comments on a preliminary version of the paper, which improved the quality of this work. We are also indebted to two anonymous referees for their useful criticism that helped us cleaning up the theory of the paper. The first author gratefully acknowledge funding provided by Canadian Natural Sciences and Engineering Research Council (NSERC) under Discovery Grant RGPIN-2020-05395. The second author acknowledge funding under RGPIN 2015-06289 and FRQNT NC-198928.

References

Adams, W. P. and Forrester, R. J. (2005). A simple recipe for concise mixed 0-1 linearizations. *Operations Research Letters*, 33(1):55 – 61.

Adams, W. P., Forrester, R. J., and Glover, F. W. (2004). Comparisons and enhancement strategies for linearizing mixed 0-1 quadratic programs. *Discrete Optimization*, 1(2):99–120.

Adams, W. P. and Johnson, T. A. (1994). Improved linear programming-based lower bounds for the quadratic assignment problem. *DIMACS series in discrete mathematics and theoretical computer science*, 16:43–77.

Adams, W. P. and Sherali, H. D. (1986). A tight linearization and an algorithm for zero-one quadratic programming problems. *Management Science*, 32(10):1274–1290.

Alumur, S. A. and Kara, B. Y. (2008). Network hub location problems: The state of the art. *European Journal of Operational Research*, 190(1):1–21.

Beasley, J. E. (1990). Or-library: Distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41(11):1069–1072.

Ben-Ameur, W. and Neto, J. (2007). Acceleration of cutting-plane and column generation algorithms: Applications to network design. *Networks*, 49(1):3–17.

Billionnet, A. and Elloumi, S. (2001). Best reduction of the quadratic semi-assignment problem. *Discrete Applied Mathematics*, 109(3):197–213.

Billionnet, A., Elloumi, S., and Plateau, M.-C. (2009). Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: The QCR method. *Discrete Applied Mathematics*, 157(6):1185–1197.

Bonami, P., Biegler, L. T., Conn, A. R., Cornuéjols, G., Grossmann, I. E., Laird, C. D., Lee, J., Lodi, A., Margot, F., Sawaya, N., et al. (2008). An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5(2):186–204.

Buchheim, C. and Traversi, E. (2018). Quadratic combinatorial optimization using separable underestimators. *INFORMS Journal on Computing*, 30(3):424–437.

Campbell, J. F. (1994). Integer programming formulations of discrete hub location problems. *European Journal of Operational Research*, 72(2):387–405.

Campbell, J. F. and O'Kelly, M. E. (2012). Twenty-five years of hub location research. *Transportation Science*, 46(2):153–169.

Carraresi, P. and Malucelli, F. (1992). A new lower bound for the quadratic assignment problem. *Operations Research*, 40(1-Supplement-1):S22–S27.

Chaovalitwongse, W., Pardalos, P. M., and Prokopyev, O. A. (2004). A new linearization technique for multi-quadratic 0–1 programming problems. *Operations Research Letters*, 32(6):517–522.

Chrétienne, P. (1989). A polynomial algorithm to optimally schedule tasks on a virtual distributed system under tree-like precedence constraints. *European Journal of Operational Research*, 43(2):225–230.

Contreras, I. and Fernández, E. (2014). Hub location as the minimization of a supermodular set function. *Operations Research*, 62(3):557–570.

Dehghan, A. and Shah, M. (2018). Binary quadratic programing for online tracking of hundreds of people in extremely crowded scenes. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):568–581.

Duives, J., Lodi, A., and Malaguti, E. (2013). Test-assignment: a quadratic coloring problem. *Journal of heuristics*, pages 1–16.

Duran, M. A. and Grossmann, I. E. (1986). An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36(3):307–339.

Ernst, A. T. and Krishnamoorthy, M. (1996). Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location science*, 4(3):139–154.

Fan, N. and Pardalos, P. M. (2010). Linear and quadratic programming approaches for the general graph partitioning problem. *Journal of Global Optimization*, 48(1):57–71.

Favati, P. (1990). Convexity in nonlinear integer programming. *Ricerca Operativa*, 53:3–44.

Ferreira, J. F. B., Khoo, Y., and Singer, A. (2018). Semidefinite programming approach for the quadratic assignment problem with a sparse graph. *Computational Optimization and Applications*, 69(3):677–712.

Fischer, A. (2014). An analysis of the asymmetric quadratic traveling salesman polytope. *SIAM Journal on Discrete Mathematics*, 28(1):240–276.

Fischer, A. and Helmberg, C. (2013). The symmetric quadratic traveling salesman problem. *Mathematical Programming*, 142(1-2):205–254.

Fischetti, M., Ljubić, I., and Sinnl, M. (2016). Redesigning benders decomposition for large-scale facility location. *Management Science*, 63(7):2146–2162.

Fletcher, R. and Leyffer, S. (1994). Solving mixed integer nonlinear programs by outer approximation. *Mathematical programming*, 66(1-3):327–349.

Gao, J. and Li, D. (2013). Optimal cardinality constrained portfolio selection. *Operations Research*, 61(3):745–761.

Geoffrion, A. M. (1972). Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260.

Glover, F. (1975). Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22(4):455–460.

Glover, F. and Woolsey, E. (1974). Technical note: Converting the 0-1 polynomial programming problem to a 0-1 linear program. *Operations Research*, 22(1):180–182.

Hager, W. W. and Krylyuk, Y. (1999). Graph partitioning and continuous quadratic programming. *SIAM Journal on Discrete Mathematics*, 12(4):500–523.

Hahn, P. M., Zhu, Y.-R., Guignard, M., Hightower, W. L., and Saltzman, M. J. (2012). A level-3 reformulation-linearization technique-based bound for the quadratic assignment problem. *INFORMS Journal on Computing*, 24(2):202–209.

Hammer, P. L. and Rubin, A. A. (1970). Some remarks on quadratic programming with 0-1 variables. *RAIRO-Operations Research-Recherche Opérationnelle*, 4(V3):67–79.

Hansen, P. (1979). Methods of nonlinear 0-1 programming. *Ann. Discrete Math. v5*, pages 53–70.

Hansen, P. and Lih, K.-W. (1992). Improved algorithms for partitioning problems in parallel, pipelined, and distributed computing. *IEEE Transactions on Computers*, 41(6):769–771.

Helmberg, C., Rendl, F., and Weismantel, R. (2000). A semidefinite programming approach to the Quadratic Knapsack Problem. *Journal of Combinatorial Optimization*, 4(2):197–215.

Henschel, R., Leal-Taixé, L., Cremers, D., and Rosenhahn, B. (2018). Fusion of head and full-body detectors for multi-object tracking. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Ilić, A., Urošević, D., Brimberg, J., and Mladenović, N. (2010). A general variable neighborhood search for solving the uncapacitated single allocation p-hub median problem. *European Journal of Operational Research*, 206(2):289–300.

Jaillet, P., Song, G., and Yu, G. (1996). Airline network design and hub location problems. *Location science*, 4(3):195–212.

Jünger, M. and Kaibel, V. (2001). Box-inequalities for quadratic assignment polytopes. *Mathematical Programming*, 91(1):175–197.

Kelley, J. E. (1960). The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712.

Klincewicz, J. G. (1998). Hub location in backbone/tributary network design: a review. *Location Science*, 6(1):307–335.

Krislock, N., Malick, J., and Roupin, F. (2017). BiqCrunch: a semidefinite branch-and-bound method for solving binary quadratic problems. *ACM Transactions on Mathematical Software (TOMS)*, 43(4):32.

Malaguti, E. and Toth, P. (2010). A survey on vertex coloring problems. *International transactions in operational research*, 17(1):1–34.

Malucelli, F. (1996). A polynomially solvable class of quadratic semi-assignment problems. *European Journal of Operational Research*, 91(3):619–622.

Malucelli, F. and Pretolani, D. (1994). Quadratic semi-assignment problems on structured graphs. *Ricerca Operativa*, 69:57–78.

Malucelli, F. and Pretolani, D. (1995). Lower bounds for the quadratic semi-assignment problem. *European Journal of Operational Research*, 83(2):365–375.

Murota, K. (1998). Discrete convex analysis. *Mathematical Programming*, 83(1-3):313–371.

Murota, K. (2019). A survey of fundamental operations on discrete convex functions of various kinds. *Optimization Methods and Software*, pages 1–47.

Nickel, S., Schöbel, A., and Sonneborn, T. (2001). Hub location problems in urban traffic networks. In *Mathematical methods on optimization in transportation systems*, pages 95–107. Springer.

O'Kelly, M. E. (1987). A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, 32(3):393–404.

Padberg, M. (1989). The boolean quadric polytope: Some characteristics, facets and relatives. *Mathematical Programming*, 45(1-3):139–172.

Padberg, M. W. and Rijal, M. P. (2012). *Location, scheduling, design and integer programming*, volume 3. Springer Science & Business Media.

Poljak, S., Rendl, F., and Wolkowicz, H. (1995). A recipe for semidefinite relaxation for (0, 1)-quadratic programming. *Journal of Global Optimization*, 7(1):51–73.

Rostami, B., Buchheim, C., Meier, J. F., and Clausen, U. (2016). Lower bounding procedures for the single allocation hub location problem. *Electronic Notes in Discrete Mathematics*, 52:69 – 76. INOC 2015 7th International Network Optimization Conference.

Rostami, B., Chassein, A., Hopf, M., Frey, D., Buchheim, C., Malucelli, F., and Goerigk, M. (2018a). The quadratic shortest path problem: complexity, approximability, and solution methods. *European Journal of Operational Research*.

Rostami, B., Kämmerling, N., Buchheim, C., and Clausen, U. (2018b). Reliable single allocation hub location problem under hub breakdowns. *Computers & Operations Research*, 96:15–29.

Rostami, B. and Malucelli, F. (2014). A revised reformulation-linearization technique for the quadratic assignment problem. *Discrete Optimization*, 14:97 – 103.

Rostami, B. and Malucelli, F. (2015). Lower bounds for the quadratic minimum spanning tree problem based on reduced cost computation. *Computers & Operations Research*, 64:178 – 188.

Rostami, B., Malucelli, F., Frey, D., and Buchheim, C. (2015). On the quadratic shortest path problem. In Bampis, E., editor, *Experimental Algorithms*, pages 379–390. Springer International Publishing, Cham.

Sahni, S. and Gonzalez, T. (1976). P-complete approximation problems. *Journal of the ACM (JACM)*, 23(3):555–565.

Saito, H. (2006). The symmetric quadratic semi-assignment polytope. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 89(5):1227–1232.

Saito, H., Fujie, T., Matsui, T., and Matuura, S. (2009). A study of the quadratic semi-assignment polytope. *Discrete Optimization*, 6(1):37 – 50.

Schüle, I., Ewe, H., and Küfer, K.-H. (2009). Finding tight RLT formulations for quadratic semi-assignment problems. In *Proceedings of 8th Cologne-Twente Workshop on Graphs and Combinatorial Optimization, CTW*, pages 109–112.

Sherali, H. D. and Adams, W. P. (2013). *A reformulation-linearization technique for solving discrete and continuous nonconvex problems*, volume 31. Springer Science & Business Media.

Sherali, H. D. and Smith, J. C. (2007). An improved linearization strategy for zero-one quadratic programming problems. *Optimization Letters*, 1(1):33–47.

Simeone, B. (1986). An asymptotically exact polynomial algorithm for equipartition problems. *Discrete Applied Mathematics*, 14(3):283–293.

Skorin-Kapov, D., Skorin-Kapov, J., and O'Kelly, M. (1996). Tight linear programming relaxations of uncapacitated p-hub median problems. *European Journal of Operational Research*, 94(3):582–593.

Sotirov, R. (2013). An efficient semidefinite programming relaxation for the graph partition problem. *INFORMS Journal on Computing*, 26(1):16–30.

Taylor, G. D., Harit, S., English, J. R., and Whicker, G. (1995). Hub and spoke networks in truckload trucking: Configuration, testing and operational concerns. *Logistics and Transportation Review*, 31(3):209.

Trinh, H., Fan, Q., Gabbur, P., and Pankanti, S. (2012). Hand tracking by binary quadratic programming and its application to retail activity recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1902–1909. IEEE.

Zha, H., He, X., Ding, C., Simon, H., and Gu, M. (2001). Bipartite graph partitioning and data clustering. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 25–32. ACM.