The quadratic cycle cover problem

de Meijer, Frank; Sotirov, Renata

Published in:
Journal of Combinatorial Optimization

DOI:
10.1007/s10878-020-00547-7

Publication date:
2020

Document Version
Publisher's PDF, also known as Version of record

Journal of Combinatorial Optimization (2020) 39:1096–1128
https://doi.org/10.1007/s10878-020-00547-7

The quadratic cycle cover problem: special cases and efficient bounds

Frank de Meijer[1] · Renata Sotirov[2]

Published online: 25 February 2020

Abstract
The quadratic cycle cover problem is the problem of finding a set of node-disjoint cycles visiting all the nodes such that the total sum of interaction costs between consecutive arcs is minimized. In this paper we study the linearization problem for the quadratic cycle cover problem and related lower bounds. In particular, we derive various sufficient conditions for the quadratic cost matrix to be linearizable, and use these conditions to compute bounds. We also show how to use a sufficient condition for linearizability within an iterative bounding procedure. In each step, our algorithm computes the best equivalent representation of the quadratic cost matrix and its optimal linearizable matrix with respect to the given sufficient condition for linearizability. Further, we show that the classical Gilmore–Lawler type bound belongs to the family of linearization based bounds, and therefore apply the above mentioned iterative reformulation technique. We also prove that the linearization vectors resulting from this iterative approach satisfy the constant value property. The best among here introduced bounds outperform existing lower bounds when taking both quality and efficiency into account.

Keywords Quadratic cycle cover problem · Linearization problem · Equivalent representations · Gilmore–Lawler bound

1 Introduction

A disjoint cycle cover in a directed graph is a set of node-disjoint cycles such that every node belongs to exactly one cycle. The quadratic cycle cover problem (QCCP)

✉ Renata Sotirov
r.sotirov@uvt.nl

Frank de Meijer
f.j.j.demeijer@uvt.nl

[1] CentER, Department of Econometrics and OR, Tilburg University, Tilburg, The Netherlands

[2] Department of Econometrics and OR, Tilburg University, Tilburg, The Netherlands

is the problem of finding a disjoint cycle cover in a graph such that the total sum of interaction costs between consecutive arcs is minimized. Since we assume that all cycle covers in this paper are disjoint, we use the term cycle cover to denote this concept throughout this work. The QCCP is proven to be NP-hard (Fischer et al. 2009). The corresponding linear problem is called the cycle cover problem (CCP), in which one wants to find a minimum cycle cover with respect to linear arc costs. It is well known that the CCP is solvable in polynomial time.

In the literature several special cases with respect to the objective function of the QCCP are considered. In the angular metric cycle cover problem (Angle- CCP) the quadratic costs represent the change of the direction induced by two consecutive arcs. The goal of Angle- CCP is to find a cycle cover of the graph while minimizing the total angular costs. The Angle- CCP has applications in robotics (Aggarwal et al. 1999). In the same paper it is shown that Angle- CCP is NP-hard. Only recently Galbiati et al. (2014) introduced another special case of the QCCP: the minimum reload cost cycle cover problem (MinRC3). The MinRC3 problem asks for a minimum cycle cover in an arc-colored graph under the reload cost model. A reload cost is an interaction cost that is paid when two arcs of different colors are placed in succession on a cycle. The goal of the MinRC3 problem is to find a cycle cover such that the total reload cost is minimized. The problem is proven to be NP-hard in the strong sense (Galbiati et al. 2014). The notion of reload costs is introduced by Wirth and Steffan (2001), and it has many applications in various fields, e.g., in cargo, energy and telecommunication networks (Amaldi et al. 2011; Wirth and Steffan 2001). A detailed overview of the MinRC3 problem and its applications can be found in Büyükçolak et al. (2019). Several other combinatorial optimization problems including these reload costs have been investigated, see e.g., Amaldi et al. (2011), Galbiati (2008), Gamvros et al. (2012), Gourvès et al. (2010), Wirth and Steffan (2001).

The QCCP is closely related to the quadratic traveling salesman problem (QTSP) which is introduced in Jäger and Molitor (2008). The QTSP is the problem of finding a Hamiltonian cycle in a graph minimizing a quadratic cost function. It has applications in bioinformatics, robotics and telecommunication (Fischer et al. 2014). When we remove the subtour elimination constraints, the QTSP boils down to the QCCP. Therefore, the QCCP is often used to provide lower bounds for the QTSP (Fischer et al. 2014; Jäger and Molitor 2008; Staněk et al. 2019). For this reason, the quadratic cycle cover problem is an interesting optimization problem that has received more attention in the past few years.

Several papers have been written about solution methods for the QCCP or its related problems. Jäger and Molitor (2008) introduced the QCCP in order to use the QCCP bounds as lower bounds in a branch-and-bound algorithm for the QTSP. Staněk et al. (2019) use the QCCP combined with a rounding procedure to construct heuristics for the QTSP. Aggarwal et al. (1999) provide a $O(\log n)$-approximation algorithm for the Angle- CCP. Fischer (2013) studies the polyhedral properties of the QCCP by proving that some triangle inequalities are facet-defining. Galbiati et al. (2014) derive various integer programming formulations for the MinRC3 problem. They exploit one of those formulations together with a column generation approach to compute lower bounds for the original problem. Moreover, in Galbiati et al. (2014) a local search algorithm based on 2-exchange and 3-exchange neighbourhoods is constructed to compute upper bounds for the MinRC3 problem. Büyükçolak et al. (2019) study the MinRC3 problem on complete graphs with an equitable or nearly equitable 2-edge coloring. For these types of graphs (except some special cases) a polynomial time algorithm is derived that constructs a monochromatic cycle cover.

We focus here on the linearization problem of the QCCP and its applications. An instance of the quadratic cycle cover problem is called linearizable if there exists an instance of the linear cycle cover problem such that the associated costs for both problems are equal for all feasible cycle covers. The linearization problem of the quadratic cycle cover problem asks whether a given instance of the QCCP is linearizable. To the best of our knowledge, this is the first paper about the linearization problem of the QCCP.

In the past few years linearization problems have become an active field of research for many combinatorial optimization problems. Kabadi and Punnen study the linearization problem of the quadratic assignment problem (QAP) and provide polynomial time algorithms that solve it. In particular, Kabadi and Punnen (2011) (resp. Punnen and Kabadi (2013)) present an $O(n^4)$ (resp. $O(n^2)$) algorithm for the general (resp. Koopmans-Beckmann) QAP linearization problem, where n is the size of the problem. The linearization problem for the quadratic minimum spanning tree problem and the quadratic traveling salesman problem are studied by Ćustić and Punnen (2018) and Punnen et al. (2018), respectively. Hu and Sotirov (2019) develop a polynomial time algorithm that solves the linearization problem of the quadratic shortest path problem on directed graphs.

1.1 Main results and outline

In this paper, we first provide an elegant and compact proof that the quadratic cycle cover problem is strongly NP-hard and not approximable within any constant factor unless P = NP. Then, we consider the linearization problem of the QCCP and derive various sufficient conditions for an instance of the QCCP to be linearizable. In particular, we provide three different types of weak sum conditions on the data matrix for which the corresponding instance can be solved in polynomial time. Further, we present a general framework in which each sufficient condition of linearizability can be used to construct a lower bound on the optimal objective value. Each of these bounds can be computed by a solving a linear programming problem, as long as the set of linearizable matrices is a polyhedron. These types of bounds are called linearization based bounds (LBB), and were recently introduced by Hu and Sotirov (2019) for general binary quadratic problems. However, our LBBs exploit sufficient conditions of linearizability suited for the QCCP.

Furthermore, we show how to use a sufficient condition of linearizability within an iterative bounding procedure. In each iteration, we search for the best equivalent representation of the objective and its optimal linearizable matrix that satisfies a particular sufficient condition of linearizability. We refer to the resulting bound as the reformulation based bound (RBB). Our iterative bounding procedure can be seen as a generalization of similar iterative procedures, see e.g., Carraresi and Malucelli (1992), Rostami and Malucelli (2015) and Rostami et al. (2018).

Finally, we consider the classical Gilmore–Lawler (GL) type bound (Gilmore 1962; Lawler 1963). First, we show that the GL type bound for the QCCP can be obtained by solving a single linear programming problem instead of solving m (integer) subproblems, where m equals the number of arcs in the graph. Then, we prove that the GL type bound belongs to the family of linearization based bounds by providing the appropriate sufficient condition. We implement our iterative bounding procedure to compute the RBB using the GL type bound. In the literature, iterative approaches for various problems that are based on the GL type bounds use dual variables to obtain bounds, and do not search for equivalent reformulations that provide best bounds in each iteration. Clearly, our approach outperforms others in terms of strength of the bound. Another interesting result is that the linearization vectors resulting from this iterative procedure satisfy the constant value property – yet another important property for linearizability.

Our numerical results show that the introduced bounding approaches are efficient and provide strong bounds compared to several methods from the literature. In particular, our most prominent bound can be computed within 60 s for instances up to 15,000 arcs. Interestingly, the GL type bound that is known to be one of the computationally cheapest bounds for quadratic optimization problems cannot be computed on such large instances.

This paper is organized as follows. In Sect. 2, we formally introduce the QCCP and prove its NP-hardness. In Sect. 3, the linearization problem for the QCCP is introduced and several sufficient conditions for linearizability are derived. The general framework for the computation of the linearization based bounds is discussed in Sect. 4. These bounds are used in Sect. 5 to construct an iterative bounding procedure for each sufficient condition. In Sect. 6, we consider the classical GL type bound and prove that it belongs to the family of linearization based bounds. We also show how the iterative procedure for this linearization based bound boils down to the computation of the strongest GL type bound in each step. In Sect. 7, we briefly discuss several other bounds from the literature. Numerical results are given in Sect. 8.

1.2 Notation

A directed graph $G = (N, A)$ is given by a node set N and an arc set $A \subseteq N \times N$. For all nodes $i \in N$ we denote by $\delta^+(i)$ the set of arcs that are leaving i. Similarly, $\delta^-(i)$ denotes the set of arcs that are entering i. For all arcs $e \in A$ we let e^+ and e^- denote the starting and ending node of e, respectively. To avoid confusion, the letters e, f and g are only used to denote arcs in this work.

For any square matrix M, we introduce the operator $\text{diag} : \mathbb{R}^{n\times n} \to \mathbb{R}^n$ that maps a matrix to a vector consisting of its diagonal elements. Moreover, we denote by $\text{Diag} : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ its adjoint operator. That is, for any $v \in \mathbb{R}^n$ the matrix $\text{Diag}(v)$ equals a diagonal matrix with the entries of v on its main diagonal.

2 The quadratic cycle cover problem

In this section, we formally introduce the quadratic cycle cover problem.

An instance $\mathcal{I}$ of the QCCP is specified by the pair $\mathcal{I} = (G, Q)$, where $G = (N, A)$ is a directed graph with n vertices and m arcs and $Q = (Q_{ef}) \in \mathbb{R}^{m\times m}$ is a quadratic cost matrix. The entries in Q are such that $Q_{ef} = 0$ if f is not a successor of e. In other words, the quadratic cost of two arcs e and f can be nonzero only if the starting node of f equals the ending node of e. In case we also consider linear arc costs, i.e. we have a cost function $p : A \to \mathbb{R}$, we can put these arc costs on the diagonal of the quadratic cost matrix. Therefore, we assume that the cost structure of an instance of the QCCP is fully determined by its quadratic cost matrix.

Now let $x \in \{0, 1\}^m$ be a vector with $x_e = 1$ if arc e belongs to a cycle cover, and 0 otherwise. Then the QCCP can be formulated as

$$\begin{aligned} OPT(Q) := \quad \min \quad & x^T Q x \\ \text{s.t.} \quad & x \in X, \end{aligned} \tag{1}$$

where X denotes the set consisting of all disjoint cycle covers in G, i.e.

$$X := \left\{ x \in \{0, 1\}^m \mid \sum_{e\in\delta^+(i)} x_e = \sum_{e\in\delta^-(i)} x_e = 1\, , \ \forall i \in N \right\}. \tag{2}$$

The above set equals the set of directed 2-factors in G. For the existence of such a directed 2-factor in a directed graph, see e.g., Chiba and Yamashita (2018).

The quadratic cycle cover problem is NP-hard (Fischer et al. 2009). Also, the related problems ANGLE- CCP and the MINRC3 problem are shown to be NP-hard (Aggarwal et al. 1999) and strongly NP-hard (Galbiati et al. 2014), respectively. We now provide an alternative reduction that establishes strong NP-hardness which is based on a reduction from the quadratic assignment problem. We consider the Koopmans-Beckmann form of the QAP introduced in Koopmans and Beckmann (1957). Let F and P be a set of n facilities and n locations, respectively, $w : F \times F \to \mathbb{R}$ a weight function and $d : P \times P \to \mathbb{R}$ a distance function. Without loss of generality, we assume that $d_{ii} = w_{ii} = 0$ for all $i \in \{1, \ldots, n\}$. Then, we search for a bijection $\pi : F \to P$ such that $\sum_{i=1}^{n}\sum_{j=1}^{n} d_{\pi(i)\pi(j)} w_{ij}$ is minimized. The QAP is NP-hard in the strong sense and not approximable within any constant factor (Sahni and Gonzalez 1976).

Theorem 1 *The QCCP is NP-hard in the strong sense and cannot be approximated within a constant factor unless P = NP.*

Proof Let an instance $\mathcal{I}$ of the QAP be given, i.e., we consider sets $F = \{1, \ldots, n\}$ and $P = \{1', \ldots, n'\}$ with $|P| = |F| = n$, functions $w : F \times F \to \mathbb{R}$ and $d : P \times P \to \mathbb{R}$ and a positive integer K. We create an instance $\mathcal{I}'$ of the QCCP.

For the reduction we create a directed graph $G = (N, A)$ that consists of cells. A cell belongs to a single facility and consists of n nodes, each of them corresponding to an assignment to one of the n locations. These nodes are specified by the pairs $(i\ j')$

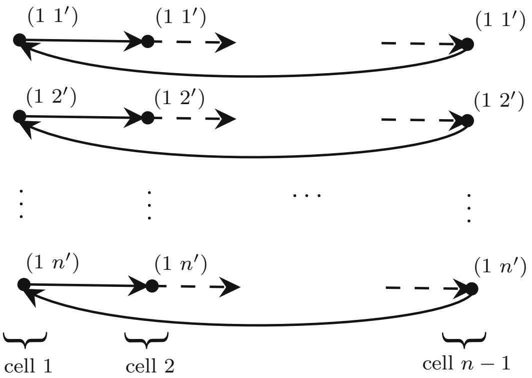

Fig. 1 Group consisting of $n-1$ cells corresponding to facility 1

where $i \in F$ and $j' \in P$. For each facility $i \in F$, we define a set of $n-1$ identical cells, which we call a group. The nodes corresponding to the same assignment within a group are placed on a directed cycle, where the arcs are oriented from cell i to cell $i+1$ for $i = 1, \ldots, n-2$, and from cell $n-1$ to cell 1. In this way, we obtain n cycles per group, which we call inner cycles. We set the interaction cost between each of the successive arcs within a group to zero for all groups. In Figure 1 the group corresponding to facility 1 is given. In a similar fashion we construct groups corresponding to the remaining facilities.

We now specify the connections between the groups. Each of the $n-1$ cells (in each group) is connected to precisely one cell from the $n-1$ other groups such that each cell is connected to a different group. Hence, each cell in G is connected to exactly one other cell in G. Since we have n groups and each group consists of $n-1$ cells, this results in $\binom{n}{2}$ connections. Connecting the cells of two groups is done by introducing a connection node and a relink node. Starting from the first group, we draw an arc from every node of one of its cells to the connection node. Successively, we draw an arc from the connection node to all the nodes of one of the cells of the second group. The same is done for the relink node, now in the reverse direction. Figure 2 depicts an overview of the connection between the last cell of group i and the first cell of group j. We denote the cycles between the groups by outer cycles. In Figure 2 solid arcs are used for the outer cycles, while the inner cycles are drawn using dashed arcs. A similar connection via connection and relink nodes exists for all other pairs of groups.

Observe that any arc in G either belongs to an inner cycle or to several outer cycles. The quadratic cost of a pair of successive arcs (e, f) where e belongs to an inner cycle and f to an outer cycle or vice versa, is set to ∞. It remains to specify the interaction cost between successive arcs on an outer cycle. We only specify the quadratic cost between the arcs entering and leaving the connection node, other costs are set to zero.

Let i and j be two distinct groups associated with facility i and j, respectively. Let a node in group i be given by $(i\ k')$ with $k' \in P$. Similarly, a node in group j is given by $(j\ l')$ with $l' \in P$. Let $e_{ik'}$ denote the arc between $(i\ k')$ and the connection node and let $f_{jl'}$ denote the arc between the connection node and $(j\ l')$. Then the quadratic cost between $e_{ik'}$ and $f_{jl'}$ is defined as follows:

$$Q_{e_{ik'},f_{jl'}} := \begin{cases} d_{k'l'}w_{ij} + d_{l'k'}w_{ji} & \text{if } k' \neq l' \\ \infty & \text{otherwise.} \end{cases}$$

We repeat this construction for any two connected cells. Figure 3 gives a simplified overview of G for $n = 4$. The circles in the center denote the connections between the cells, where the connection and relink nodes are drawn using the symbols '•' and '$*$', respectively. The graph G has $n^2(n-1)+2\binom{n}{2} = \mathcal{O}(n^3)$ nodes and $n^2(n-1)+4\binom{n}{2} = O(n^3)$ arcs.

It remains to show that there exists a cycle cover in G with cost at most K if and only if there exists a feasible assignment in $\mathcal{I}$ with cost at most K.

First, we verify that a cycle cover with finite cost in G corresponds to a feasible assignment of facilities and locations. Note that the connection and relink nodes must be covered by an outer cycle, since any other cycle would induce a cost of ∞. Besides the connection and relink nodes, this cycle contains two nodes that each correspond to an assignment of a different facility. Moreover, these facilities must be assigned to different locations, otherwise this implies an infinite cost. The nodes in a cell that are not covered by an outer cycle must be covered by an inner cycle. Consequently, nodes on these inner cycles cannot belong to an outer cycle. Therefore, for each group exactly one location is 'chosen' to be on an outer cycle, i.e., each facility is assigned to some location. Moreover, no two facilities are assigned to the same location, since this would imply a cost of ∞ at the connection node connecting these groups. We conclude that a cycle cover with finite cost corresponds to a feasible assignment and vice versa.

Observe that the objective value of a feasible assignment in the QAP instance equals the total cost of the corresponding cycle cover in the QCCP instance. Namely, the latter cost equals the sum of quadratic costs incurred at the $\binom{n}{2}$ connection nodes. If facility i (resp. j) where $i \neq j$ is assigned to location k' (resp. l') where $k' \neq l'$, then this cost equals $d_{k'l'}w_{ij} + d_{l'k'}w_{ji}$. Taking the sum over all connections, the total cost of the cycle cover equals $\sum_{i=1}^{n}\sum_{j=1}^{n} d_{\pi(i)\pi(j)}w_{ij}$ where $\pi : F \to P$ is the bijection corresponding to the assignment.

We conclude that there exists an assignment for the QAP instance with cost at most K if and only if there exists a feasible cycle cover in the corresponding QCCP instance of cost at most K. Since the QAP is strongly NP-hard and the numbers defined in the reduction are polynomially bounded (infinite costs can be replaced by an appropriate value which is polynomially bounded in the largest number and the size of $\mathcal{I}$), we conclude that the QCCP is strongly NP-hard.

Moreover, as the QAP cannot be approximated within any constant factor (Sahni and Gonzalez 1976) and the reduction above is clearly gap preserving, the result follows. □

3 The QCCP linearization problem

In this section, we formally introduce the linearization problem for the QCCP and derive various sufficient conditions for an instance of the QCCP to be linearizable.

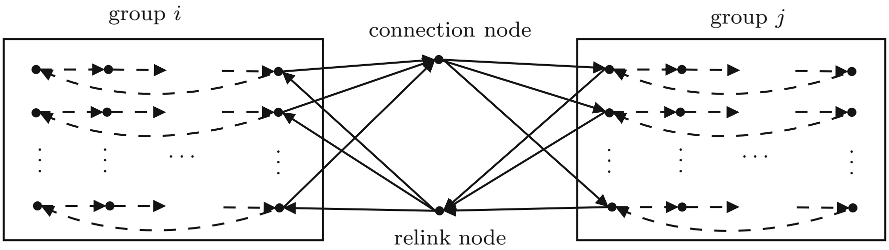

Fig. 2 Connection between two cells of group i and j

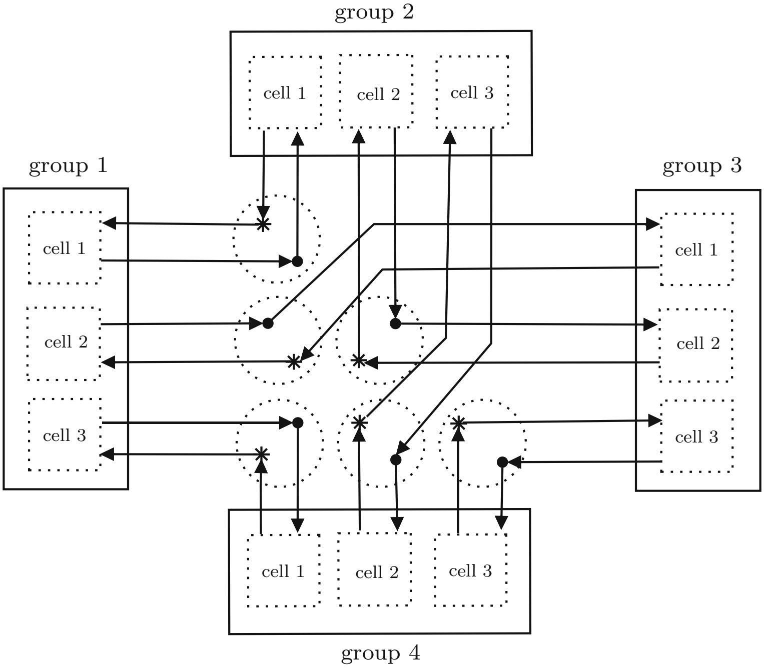

Fig. 3 Simplified overview of G for $n = 4$

Several of these conditions are used later on to construct lower bounds for the optimal value of the QCCP.

Let us consider the (linear) cycle cover problem. Given a cost function $p : A \to \mathbb{R}$, the CCP is the problem of finding a cycle cover of minimum cost. It can be written as follows:

$$\min_{x \in \{0,1\}^m} \left\{ p^T x \mid x \in X \right\}, \tag{3}$$

where X is given in (2). Since the constraint set of X is totally unimodular, it follows that the CCP is solvable in polynomial time. We call an instance $\mathcal{I} = (G, Q)$ of the QCCP linearizable if there exists a cost vector $p \in \mathbb{R}^m$ such that $x^T Qx = p^T x$ for all cycle covers $x \in X$. If such a vector p exists, we call p a linearization vector of Q for the QCCP.

The QCCP linearization problem can be stated as follows: Given an instance $\mathcal{I} = (G, Q)$ of the QCCP, verify whether it is linearizable and, if yes, compute a linearization vector p of Q.

In the remaining part of this section we provide sufficient conditions for the quadratic cost matrix Q to be linearizable. The first type of sufficient conditions for linearizability are related to the constant value property (CVP) for cost vectors or cost matrices. The definition associated with the CCP is stated below.

Definition 1 A cost matrix p satisfies the constant value property if $p^T x = p^T \bar{x}$ for all cycle covers $x, \bar{x} \in X$.

A similar definition holds for the quadratic version of the problem.

Definition 2 A cost matrix Q satisfies the constant value property if $x^T Qx = \bar{x}^T Q\bar{x}$ for all cycle covers $x, \bar{x} \in X$.

When Q satisfies the constant value property then Q is linearizable, as stated by the following proposition.

Proposition 1 *Assume that Q satisfies the constant value property, i.e. $x^T Qx = \xi$ where $\xi \in \mathbb{R}$ for all $x \in X$, then Q is linearizable with cost vector p defined as $p_e = \xi/n$ for all $e \in A$.*

Proof For all $x \in X$ we have $x^T Qx = \xi = n\frac{\xi}{n} = x^T p$ since x has exactly n nonzero elements. □

A more restricted version of the CVP is obtained when the interaction cost of a single arc with its successor or predecessor is constant for all cycle covers $x \in X$. We refer to these properties as the row and column constant value property, respectively. These definitions are based on similar definitions by Punnen et al. (2018) for the QTSP.

Definition 3 A cost matrix Q satisfies the row CVP if there exists some $b \in \mathbb{R}^m$ such that for all arcs $e \in A$ we have $Q_{ef} = Q_{eg} = b_e$ for all $f, g \in \delta^+(e^-)$ and $Q_{ef} = 0$ otherwise. A cost matrix Q satisfies the column CVP if there exists some $c \in \mathbb{R}^m$ such that for all arcs $e \in A$ we have $Q_{fe} = Q_{ge} = c_e$ for all $f, g \in \delta^-(e^+)$ and $Q_{fe} = 0$ otherwise.

It is not hard to verify that an instance of the QCCP is linearizable if the cost matrix Q satisfies the row or column CVP.

Proposition 2 *If Q satisfies the row CVP or the column CVP, then Q is linearizable.*

Proof We prove the case when Q satisfies the row CVP. Assume that $b \in \mathbb{R}^m$ is such that for all arcs $e \in A$, $Q_{ef} = Q_{eg} = b_e$ for all $f, g \in \delta^+(e^-)$ and $Q_{ef} = 0$ otherwise. Since $Q_{ef} = 0$ when e and f are not successors, we know that $x^T Qx = \sum_{e\in A} \sum_{f\in\delta^+(e^-)} Q_{ef} x_e x_f$. We have

$$\sum_{e\in A} \sum_{f\in\delta^+(e^-)} Q_{ef} x_e x_f = \sum_{e\in A} x_e b_e \sum_{f\in\delta^+(e^-)} x_f = \sum_{e\in A} x_e b_e = x^T b.$$

The proof for the column CVP is similar. □

A matrix $Q \in \mathbb{R}^{m\times m}$ is called a sum matrix if there exist $b, c \in \mathbb{R}^m$ such that $Q_{ef} = b_e + c_f$ for all e, f. A weak sum matrix is a matrix for which this property holds except for the entries on the diagonal, i.e., $Q_{ef} = b_e + c_f$ for all $e \neq f$. The weak sum property is used as a condition for linearizability for several quadratic problems, see e.g., Hu and Sotirov (2018) and Punnen et al. (2018). Since in this work we only incur a cost when two arcs are successive, we use a different form of the weak sum condition in which we only put a restriction on successive arcs. We call this condition the incident weak sum property.

Definition 4 A matrix Q is called incident weak sum if there exist vectors $b, c \in \mathbb{R}^m$ such that $Q_{ef} = b_e + c_f$ for all $e \in A$, $f \in \delta^+(e^-)$ and $Q_{ef} = 0$ otherwise, i.e., $Q_{ef} = b_e + c_f$ for all pairs of arcs e, f such that f is a successor of e. If such vectors b and c exist, these vectors are called supporting vectors of Q.

If the quadratic cost matrix Q is an incident weak sum matrix, then Q is linearizable as stated in the following proposition.

Proposition 3 *Let Q be an incident weak sum matrix with supporting vectors $b, c \in \mathbb{R}^m$. Then Q is linearizable with cost vector $p = b + c$.*

Proof We show that for all $x \in X$ we have $x^T Qx = p^T x$ where $p = b + c$. Note that we have $x^T Qx = \sum_{e\in A} \sum_{f\in\delta^+(e^-)} Q_{ef} x_e x_f$, since $Q_{ef} = 0$ for all arcs that are not successors. Now,

$$\begin{aligned}
\sum_{e\in A} \sum_{f\in\delta^+(e^-)} Q_{ef} x_e x_f &= \sum_{e\in A} \sum_{f\in\delta^+(e^-)} (b_e + c_f) x_e x_f \\
&= \sum_{e\in A} b_e x_e \sum_{f\in\delta^+(e^-)} x_f + \sum_{f\in A} c_f x_f \sum_{e\in\delta^-(f^+)} x_e \\
&= \sum_{e\in A} b_e x_e + \sum_{f\in A} c_f x_f = \sum_{e\in A} p_e x_e.
\end{aligned}$$

Here we use that $\sum_{f\in\delta^+(e^-)} x_f = \sum_{e\in\delta^-(f^+)} x_e = 1$, since x is a cycle cover. □

From Proposition 3 it follows that the incident weak sum property is a sufficient condition for Q to be linearizable. By including linear arc costs, this result remains valid, since we only increase the entries on the diagonal of Q.

Moreover, note that when Q satisfies the row or column CVP, then Q is an incident weak sum matrix. Next, we provide a special type of instance for which the cost matrix is not by definition linearizable, but for which we can still obtain its optimal value by solving a linear cycle cover problem.

Definition 5 A matrix $Q \in \mathbb{R}^{m\times m}$ is called a symmetric product matrix if $Q = aa^T$ for some vector $a \in \mathbb{R}^m$.

Equivalently, we can say that Q is a symmetric product matrix if it is a symmetric positive semidefinite matrix of rank one. Instances with such a quadratic cost matrix can be solved in polynomial time, as stated in the following proposition.

Proposition 4 *If the quadratic cost matrix Q is a symmetric product matrix, i.e. $Q = aa^T$ for some $a \in \mathbb{R}^m$, then the optimal cycle cover can be computed in polynomial time.*

Proof Let Q be such that $Q = aa^T$ for some $a \in \mathbb{R}^m$. Then,

$$x^T Qx = x^T aa^T x = (a^T x)^T (a^T x) = (a^T x)^2.$$

Minimizing $x^T Qx$ over all $x \in X$ is then equivalent to minimizing $a^T x$ over all cycle covers $x \in X$. □

Until now we considered instances for which Q is of the desired type, i.e., $Q_{ef} = 0$ when f is not a successor of e. Below we derive two sufficient conditions for linearizability of a matrix Q which can have nonzero interaction cost between non-consecutive arcs. Although these cost matrices do not meet the assumptions of the QCCP, we can still use them to derive strong bounds for the objective value of the original problem. This is addressed in Sect. 4.

Punnen et al. (2018) introduce a generalized version of the weak sum property for the QTSP. Their approach can be applied to the QCCP. However, since Punnen et al. (2018) prove the condition to hold for complete graphs, we provide a proof for general digraphs.

First, we define some new terminology. Instead of writing Q_{ef} for $e, f \in A$ we can also write $Q_{ij,kl}$ with $(i, j), (k, l) \in A$. Let N_i^+ denote the set of nodes j for which there exists an arc $(i, j) \in A$, i.e., $N_i^+ := \{j \in N \mid (i, j) \in A\}$. Similarly, let N_i^- be the set of nodes j for which an arc $(j, i) \in A$ exists, i.e., $N_i^- := \{j \in N \mid (j, i) \in A\}$. Now we can introduce the notion of a generalized weak sum matrix and prove that it is linearizable.

Definition 6 Q is called a generalized weak sum matrix if there exist $B, C \in \mathbb{R}^{m\times n}$ and $D, T \in \mathbb{R}^{n\times m}$ such that $Q_{ij,kl} = b_{ij,k} + c_{ij,l} + d_{i,kl} + t_{j,kl}$ for all i, j, k, l with $(i, j), (k, l) \in A$. If such B, C, D and T exist, these matrices are called supporting matrices of Q.

Now we can prove the following proposition.

Proposition 5 *If Q is a generalized weak sum matrix with supporting matrices $B, C \in \mathbb{R}^{m\times n}$ and $D, T \in \mathbb{R}^{n\times m}$, then Q is linearizable with cost vector p given by $p_{ij} = \sum_{k=1}^n b_{ij,k} + \sum_{k=1}^n c_{ij,k} + \sum_{k=1}^n d_{k,ij} + \sum_{k=1}^n t_{k,ij}$.*

Proof Let $\bar{b}_{ij} := \sum_{k=1}^n b_{ij,k}$, $\bar{c}_{ij} := \sum_{k=1}^n c_{ij,k}$, $\bar{d}_{ij} := \sum_{k=1}^n d_{k,ij}$, $\bar{t}_{ij} := \sum_{k=1}^n t_{k,ij}$ and $p_{ij} = \bar{b}_{ij} + \bar{c}_{ij} + \bar{d}_{ij} + \bar{t}_{ij}$ for all $(i, j) \in A$. Then, for $x \in X$,

$$
\begin{aligned}
x^T Qx &= \sum_{i\in N}\sum_{j\in N_i^+}\sum_{k\in N}\sum_{l\in N_k^+} Q_{ij,kl}x_{ij}x_{kl}\\
&= \sum_{i\in N}\sum_{j\in N_i^+}\sum_{k\in N}\sum_{l\in N_k^+} b_{ij,k}x_{ij}x_{kl} + \sum_{i\in N}\sum_{j\in N_i^+}\sum_{l\in N}\sum_{k\in N_l^-} c_{ij,l}x_{ij}x_{kl}\\
&\quad + \sum_{k\in N}\sum_{l\in N_k^+}\sum_{i\in N}\sum_{j\in N_i^+} d_{i,kl}x_{ij}x_{kl} + \sum_{k\in N}\sum_{l\in N_k^+}\sum_{j\in N}\sum_{i\in N_j^-} t_{j,kl}x_{ij}x_{kl}\\
&= \sum_{i\in N}\sum_{j\in N_i^+} x_{ij}\sum_{k\in N} b_{ij,k}\sum_{l\in N_k^+} x_{kl} + \sum_{i\in N}\sum_{j\in N_i^+} x_{ij}\sum_{l\in N} c_{ij,l}\sum_{k\in N_l^-} x_{kl}\\
&\quad + \sum_{k\in N}\sum_{l\in N_k^+} x_{kl}\sum_{i\in N} d_{i,kl}\sum_{j\in N_i^+} x_{ij} + \sum_{k\in N}\sum_{l\in N_k^+} x_{kl}\sum_{j\in N} t_{j,kl}\sum_{i\in N_j^-} x_{ij}\\
&= \sum_{i\in N}\sum_{j\in N_i^+} (\bar{b}_{ij}+\bar{c}_{ij}+\bar{d}_{ij}+\bar{t}_{ij})x_{ij} = \sum_{i\in N}\sum_{j\in N_i^+} p_{ij}x_{ij}\,,
\end{aligned}
$$

where we use the fact that $\sum_{l\in N_k^+} x_{kl} = \sum_{k\in N_l^-} x_{kl} = \sum_{j\in N_i^+} x_{ij} = \sum_{i\in N_j^-} x_{ij} = 1$ since x is a cycle cover. □

Note that an incident weak sum matrix can be seen as a special case of a generalized weak sum matrix. That is, for all $(i,j)\in A$ we set $b_{ij,j} = b_{ij}$ and $b_{ij,k} = 0$ for all $k \neq j$ and for all $(k,l)\in A$ we set $t_{k,kl} = t_{kl}$ and $t_{j,kl} = 0$ for all $j \neq k$. Moreover, let C and D be the zero matrix. Then we have $Q_{ij,jl} = b_{ij,j}+c_{ij,l}+d_{i,jl}+t_{j,jl} = b_{ij}+t_{jl}$ for all $(i,j),(j,l)\in A$ and $Q_{ij,kl} = 0$ otherwise.

When Q is a generalized weak sum matrix, we need $4mn$ parameters to describe Q. This number can be reduced by considering a more restricted version of a generalized weak sum matrix.

Definition 7 Q is called a restricted generalized weak sum matrix if there exist $C\in\mathbb{R}^{m\times n}$, $D\in\mathbb{R}^{n\times m}$ and $b,t\in\mathbb{R}^m$ such that $Q_{ij,jl} = b_{ij}+c_{ij,l}+d_{i,jl}+t_{jl}$ for all i,j,l with $(i,j),(j,l)\in A$ and $Q_{ij,kl} = c_{ij,l}+d_{i,kl}$ otherwise. If such C, D and b, t exist, these are called supporting matrices and vectors, respectively.

We can show that restricted generalized weak sum matrices are linearizable.

Proposition 6 *If Q is a restricted generalized weak sum matrix with supporting matrices C, D and supporting vectors b, t, then Q is linearizable with vector p given by $p_{ij} = b_{ij} + \sum_{k=1}^n c_{ij,k} + \sum_{k=1}^n d_{k,ij} + t_{ij}$.*

Proof Define $B\in\mathbb{R}^{m\times n}$ and $T\in\mathbb{R}^{n\times m}$ as follows:

$$
B_{ij,k} = \begin{cases} b_{ij} & \text{if } k = j\\ 0 & \text{otherwise}\end{cases} \qquad \forall (i,j)\in A,\ \forall k\in N
$$

$$
T_{k,ij} = \begin{cases} t_{ij} & \text{if } k = i\\ 0 & \text{otherwise}\end{cases} \qquad \forall (i,j)\in A,\ \forall k\in N
$$

Now the matrices B, C, D and T are such that they satisfy the conditions of Proposition 5. This implies that Q is linearizable with vector p' given by $p'_{ij} = \sum_{k=1}^n b_{ij,k} + \sum_{k=1}^n c_{ij,k} + \sum_{k=1}^n d_{k,ij} + \sum_{k=1}^n t_{k,ij}$. Since $\sum_{k=1}^n b_{ij,k} = b_{ij}$ and $\sum_{k=1}^n t_{k,ij} = t_{ij}$ it follows that Q is linearizable with vector $p := b_{ij} + \sum_{k=1}^n b_{ij,k} + \sum_{k=1}^n d_{k,ij} + t_{ij}$. □

4 Linearization based bounds for the QCCP

In this section we show how the sufficient conditions for linearizability can be used to derive bounds for the optimal value of the QCCP. The construction of these bounds is provided in Sect. 4.1. Section 4.2 shows some preliminary numerical results of these bounding procedures.

4.1 Construction of linearization based bounds

When an instance of the QCCP is linearizable, we can solve the problem in polynomial time by solving the corresponding linear cycle cover problem. When a quadratic cost matrix Q is not linearizable, we can still use the sufficient conditions for linearizability to find lower bounds for the optimal value of the problem. This approach is introduced by Hu and Sotirov (2019) for general binary quadratic problems. We here use tailor made sufficient conditions for the QCCP, which leads to efficient lower bounds as we show later in the numerical results.

Before we proceed, let us recall the linear cycle cover problem. We introduce the matrix $U \in \mathbb{R}^{n \times m}$ with $U_{ie} = 1$ if node i is the starting node of arc e and 0 otherwise. Similarly, we define $V \in \mathbb{R}^{n \times m}$ with $V_{ie} = 1$ if node i is the ending node of arc e and 0 otherwise. Since the matrix $[U^T, V^T]^T$ is totally unimodular, the optimal value of the CCP using cost vector p equals

$$OPT(p) := \min_{x \in \mathbb{R}^m} \{p^T x \mid \begin{bmatrix} U \\ V \end{bmatrix} x = \mathbb{1}_{2n}, \ x \geq 0\} \tag{4}$$

$$= \max_{y \in \mathbb{R}^{2n}} \{\mathbb{1}_{2n}^T y \mid [U^T, V^T] y \leq p\}, \tag{5}$$

where $\mathbb{1}_{2n} \in \mathbb{R}^{2n}$ equals the vector of ones. Note that (3) and (4) are equivalent optimization problems. The corresponding dual problem is given in (5).

When Q is linearizable with linearization vector p, we can find the optimal value for the QCCP by computing $OPT(p)$ using (4) or (5). If Q is not linearizable, we can search for a linearizable matrix $\hat{Q}$ that is as close as possible to Q. To guarantee that $\hat{Q}$ is indeed linearizable, it should satisfy one of the sufficient conditions for linearizability derived in Sect. 3. We define the sets $S_i(Q)$, for $i = 1, 2, 3$, consisting of cost matrices $\hat{Q}$ such that $\hat{Q}$ is linearizable w.r.t. a sufficient condition for linearizability and $Q - \hat{Q}$ is elementwise nonnegative. We have

$$S_1(Q) := \{\hat{Q} \in \mathbb{R}^{m\times m} \mid \hat{Q} \text{ is an incident weak sum matrix and } Q - \hat{Q} \geq 0\},$$
$$S_2(Q) := \{\hat{Q} \in \mathbb{R}^{m\times m} \mid \hat{Q} \text{ is a restricted generalized weak sum matrix and } Q - \hat{Q} \geq 0\},$$
$$S_3(Q) := \{\hat{Q} \in \mathbb{R}^{m\times m} \mid \hat{Q} \text{ is a generalized weak sum matrix and } Q - \hat{Q} \geq 0\}.$$

Remark 1 We do not consider the sets of cost matrices Q satisfying the row or column CVP, since these are special types of incident weak sum matrices. These type of matrices are contained in S_1.

$S_i(Q)$ can be seen as the set of all the linearizable cost matrices of a specific type that are suitable for obtaining lower bounds for the optimal value of the problem. For this purpose, we define for $i = 1, 2, 3$ the set $\tau_i(Q)$ of cost vectors $\hat{p} \in \mathbb{R}^m$ as

$$\tau_i(Q) := \{\hat{p} \in \mathbb{R}^m \mid x^T\hat{p} = x^T\hat{Q}x \text{ for all } x \in X\,,\ \ \hat{Q} \in S_i(Q)\}.$$

It is clear that for all i and all $\hat{p} \in \tau_i(Q)$ we have

$$OPT(Q) = \min_{x\in X}\{x^TQx\} \geq \min_{x\in X}\{x^T\hat{Q}x\} = \min_{x\in X}\{x^T\hat{p}\} = OPT(\hat{p}).$$

So, indeed, $OPT(\hat{p})$ is a lower bound for the optimal objective value of the QCCP for all i and $\hat{p} \in \tau_i(Q)$. By maximizing over all cost vectors in $\tau_i(Q)$, we obtain the strongest linearization based bound with respect to the set $S_i(Q)$, which we denote by v^i_{LBB}, see also Hu and Sotirov (2019):

$$v^i_{LBB} := \max_{\hat{p}\in\tau_i(Q)}\{OPT(\hat{p})\} = \max_{\substack{y\in\mathbb{R}^{2n}\\ \hat{p}\in\mathbb{R}^m}}\{\mathbb{1}^T_{2n}y \mid [U^T, V^T]y \leq \hat{p}\,,\ \ \hat{p} \in \tau_i(Q)\}. \tag{6}$$

The corresponding bounding approaches are denoted by $LBB1$, $LBB2$ and $LBB3$, respectively.

Remark 2 Recall that the matrices in $S_2(Q)$ and $S_3(Q)$ can have nonzero interaction cost for non-consecutive arcs, so they do not satisfy the assumptions on the quadratic cost matrix of the QCCP. Nevertheless, they can still be used to derive lower bounds for the original problem. In other words, we search for the general quadratic cost matrix that is linearizable and as close as possible to Q that gives us the best lower bound.

As long as the set $\tau_i(Q)$ is a polyhedron, the corresponding bound v^i_{LBB} can be calculated by solving the linear programming problem (6). The sets $\tau_i(Q)$ for $i = 1, 2, 3$ are indeed nonempty polyhedra, since they can be described by a finite number of linear equalities and inequalities. These polyhedral descriptions are provided in Table 1.

By construction, we have $\tau_1(Q) \subseteq \tau_2(Q) \subseteq \tau_3(Q)$ for all quadratic cost matrices Q. Consequently, we can establish the following result about the quality of the corresponding linearization based bounds.

Table 1 Polyhedral descriptions of the sets $\tau_1(Q)$, $\tau_2(Q)$ and $\tau_3(Q)$

Set	Type of cost matrix	(In)equalities which describe the set	
$\tau_1(Q)$	Incident weak sum matrix	$b_e + c_f \leq Q_{ef}$ $\hat{p}_e = b_e + c_e$ $b, c \in \mathbb{R}^m$	$\forall e \in A,\ f \in \delta^+(e^-)$ $\forall e \in A$
$\tau_2(Q)$	Restricted generalized weak sum matrix	$b_{ij} + c_{ij,l} + d_{i,jl} + t_{jl} \leq Q_{ij,jl}$ $c_{ij,l} + d_{i,kl} \leq Q_{ij,kl}$ $\hat{p}_{ij} = b_{ij} + \sum_{k=1}^n c_{ij,k} + \sum_{k=1}^n d_{k,ij} + t_{ij}$ $b, t \in \mathbb{R}^m, C \in \mathbb{R}^{m \times n}, D \in \mathbb{R}^{n \times m}$	$\forall (i,j), (j,l) \in A$ $\forall (i,j), (k,l) \in A,\ j \neq k$ $\forall (i,j) \in A$
$\tau_3(Q)$	Generalized weak sum matrix	$b_{ij,k} + c_{ij,l} + d_{i,kl} + t_{j,kl} \leq Q_{ij,kl}$ $\hat{p}_{ij} = \sum_{k=1}^n b_{ij,k} + \sum_{k=1}^n c_{ij,k}$ $+ \sum_{k=1}^n d_{k,ij} + \sum_{k=1}^n t_{k,ij}$ $B, C \in \mathbb{R}^{m \times n}, D, T \in \mathbb{R}^{n \times m}$	$\forall (i,j), (k,l) \in A$ $\forall (i,j) \in A$

Theorem 2 *For all instances of the* QCCP, *we have* $v_{LBB}^1 \leq v_{LBB}^2 \leq v_{LBB}^3$.

Proof The proof follows immediately from the construction of the linearization based bounds and the definitions of the incident weak sum matrix, the restricted generalized weak sum matrix and the generalized weak sum matrix. □

Hu and Sotirov (2019) argue that the linearization based bounds can be improved by extending the sets $\tau_i(Q)$ using a skew symmetric matrix M. That is, since each skew symmetric matrix is linearizable, a matrix $\hat{Q}$ is linearizable if and only if $\hat{Q} + M$ is linearizable for all M with $M + M^T = 0$. Using this, the set $\tau_1(Q)$ can be extended to:

$$\tau_1^{skew}(Q) := \left\{ \hat{p} \in \mathbb{R}^m \;\middle|\; \begin{array}{r} b_e + c_f + M_{ef} \leq Q_{ef},\ \forall e \in A,\ f \in \delta^+(e^-) \\ \hat{p}_e = b_e + c_e,\ \forall e \in A \\ M_{ef} = 0, \quad \forall e \in A,\ f \notin \delta^+(e^-) \\ b, c \in \mathbb{R}^m,\ M \in \mathbb{R}^{m \times m},\ M + M^T = 0 \end{array} \right\} \tag{7}$$

Note that in $\tau_1^{skew}(Q)$ we only include skew symmetric matrices whose support corresponds to the pairs of successive arcs in G, since adding dense skew symmetric matrices would increase computational complexity. Since $\tau_1(Q) \subseteq \tau_1^{skew}(Q)$, it follows that we can obtain a stronger bound by maximizing over $\tau_1^{skew}(Q)$, see Sect. 8. The same extension can be applied to any set $\tau_i(Q)$.

4.2 Preliminary results

In order to check the quality of the bounds derived above, we perform a preliminary numerical study. We create instances according to the $G(n, p)$ Erdős–Rényi model (Erdős and Rényi 1959). Here n equals the number of nodes and p equals the probability that an arc is included. We create instances for various values of n and p. The interaction cost between any two successive arcs is drawn uniformly at random as an

Table 2 Bounds and computation times in seconds of linearization based bounds on Erdős–Rényi instances

p	n	m	OPT	$LBB1$		$LBB2$		$LBB3$	
				Bound	Time	Bound	Time	Bound	Time
0.1	20	44	923	923	0.047	923	0.264	923	0.051
	25	76	1039	971	0.005	971	0.477	999	1.227
	30	100	1082	1066	0.013	1066	0.899	1082	0.787
0.3	15	61	485	478	0.010	478	0.635	485	0.714
	20	118	438	377	0.031	384	1.265	390	77.21
	25	172	382	291	0.050	295	2.531	295	869.0
0.5	15	116	226	215	0.034	215	1.407	215	90.39
	20	177	255	189	0.059	190	12.05	190	2105
	25	306	n.a.	172	0.516	173	353.6	n.a.	3600
0.7	15	149	173	127	0.017	128	0.986	128	580.5
	20	263	n.a.	116	0.094	116	7.778	n.a.	3600
	25	396	n.a.	129	0.194	129	18.26	n.a.	3600

integer from $\{1, \ldots, 100\}$. In Table 2 we present the bounds v^1_{LBB}, v^2_{LBB} and v^3_{LBB} and the computational times required for computing them. Experiments are performed using a pc with an Intel(R) Core(TM) i5-6500 CPU, 3.20 GHz and 8 GB memory using CPLEX 12.6 as the solver. The maximum computation time is set to 3600 s and we put 'n.a.' in the table when this maximum is reached before a solution is obtained.

By construction, the optimal solution has always an integer objective value. Therefore, we round up all bounds to the nearest integer. The results of Table 2 show that the linearization based bounds $LBB1$, $LBB2$ and $LBB3$ do not differ significantly, especially for the larger instances. At the same time, the computation times differ significantly. It turns out that $LBB1$ is most efficient. Therefore, this bound can be preferred when taking both quality and efficiency into account.

5 Reformulated LBB approach

In this section we discuss how a reformulation of the quadratic cost matrix can be used to obtain a non-decreasing sequence of lower bounds that are based on the linearization based bound. It is important to note that one can construct such a bounding procedure using any sufficient condition for linearizability, not only the ones discussed in Sect. 4.

Suppose we are given a sufficient condition for linearizability. Let $S(Q)$ and $\tau(Q)$ be as in Sect. 4, but now for a general sufficient condition. That is, $S(Q)$ is the set consisting of all linearizable cost matrices $\hat{Q}$ of this type with $\hat{Q} \leq Q$ and $\tau(Q)$ consists of the corresponding linearization vectors. Moreover, we assume that the set $\tau(Q)$ is a polyhedral set. Let Q_0 be the initial quadratic cost matrix. If $\hat{Q}_0 \in S(Q_0)$, we know there exists some $p_1 \in \tau(Q_0)$ such that $x^T \hat{Q}_0 x = x^T p_1$ for all $x \in X$. This leads to the following reformulation of the objective function

$$x^T Q_0 x = x^T \hat{Q}_0 x + x^T (Q_0 - \hat{Q}_0) x = x^T p_1 + x^T (Q_0 - \hat{Q}_0) x \tag{8}$$

for all cycle covers $x \in X$. By letting p_0 be the $m \times 1$ zero vector and $Q_1 := Q_0 - \hat{Q}_0$, this relation can be written as $x^T p_0 + x^T Q_0 x = x^T p_1 + x^T Q_1 x$ for all $x \in X$. The vector p_1 is taken to be the largest linearization of the matrix Q_0 (see (6)) with Q_1 being the residual quadratic part. Recall that the bound v_{LBB} is calculated by only taking the linear part of this new objective function into account. By construction we have $x^T p_0 \leq x^T p_1$ for all $x \in X$ and $0 \leq Q_1 \leq Q_0$.

Now we can proceed in a similar way by considering the linearization problem of the residual cost matrix Q_1. In other words, we search for a linearizable matrix in $S(Q_1)$ and its corresponding linearization vector p_2. If we let Q_2 be the new residual matrix, then the objective function can be reformulated as $x^T p_0 + x^T Q_0 x = x^T (p_1 + p_2) + x^T Q_2 x$. Since $x^T p_1 \leq x^T (p_1 + p_2)$, a stronger bound can be obtained. This procedure can be repeated to obtain a sequence of bounds. However, it is in general not possible to find a vector p_2 for which this bound has strictly improved. That is, since $p_1 + p_2 \in \tau(Q_0)$ this would imply that p_1 is not the optimal solution to (6). Thus, applying this procedure iteratively, the resulting sequence of bounds remains constant after the first iteration. To overcome this issue, we need to reformulate the residual cost matrix in each step.

In the literature, various iterative bounding procedures have been proposed, see e.g., Burkard et al. (2009), Punnen et al. (2019), Rostami and Malucelli (2015) and Rostami et al. (2018). In this paper we introduce a new approach that is different in two ways. First, the existing bounding procedures are mainly based on the classical Gilmore–Lawler type bound. Our approach is based on general sufficient conditions for linearizability and we can show that the Gilmore–Lawler type bounding procedure is a special case of this approach, see Sect. 6. Second, the existing bounding procedures mostly use a fixed reformulation of the cost matrix in each iteration. However, using a fixed reformulation is in general not the best one can do. Here, we search for the reformulation of the cost matrix that results in the strongest bound in the next iteration. For this purpose, we define the notion of an equivalent representation of a matrix, see e.g., Punnen et al. (2019).

Definition 8 Let (G, Q) be an instance of the QCCP. Then (G, R) is an equivalent representation of (G, Q) if $x^T Q x = x^T R x$ for all $x \in X$.

If there is no confusion about the graph G under consideration, we say that R is an equivalent representation of Q. It is easy to verify that a matrix R is an equivalent representation of Q if for all $e, f \in A$ we have $R_{ef} + R_{fe} = Q_{ef} + Q_{fe}$. Here, we focus on a specific type of equivalent representation, which we call an η-equivalent representation of Q.

Definition 9 Given $\eta \in [0, 1]$, an equivalent representation $Q^\eta := \eta Q + (1 - \eta) Q^T$ of Q is called an η-equivalent representation.

In other words, an η-equivalent representation is obtained by taking a convex combination of Q and Q^T. Moreover, it follows that if R and Q are equivalent representations, then a linearization of R is also a linearization of Q and vice versa.

Instead of considering the linearization problem of the residual matrix Q_1, we can consider the linearization problem of Q_1^η for some $\eta \in [0, 1]$. Since Q_1^η has a different structure than Q_1, it is in general possible to find a linearizable matrix $\hat{Q}_1 \in S(Q_1^\eta)$ and a corresponding linearization vector that result in a strictly stronger bound.

As already mentioned above, many approaches in the literature are based on taking a fixed value for η, e.g. $\eta = \frac{1}{2}$ which corresponds to the case of symmetrizing. This does not give the best bound in general. Instead, we search for $\eta \in [0, 1]$ that results in the strongest bound in each iteration. Suppose we are in step k of the algorithm in which we consider the linearization problem of the residual matrix Q_{k-1}. Then the optimal equivalent representation of Q_{k-1} and its corresponding vector $p_k \in \tau(Q_{k-1}^\eta)$ can be computed simultaneously by solving the following problem:

$$r_k := \max_{\substack{y \in \mathbb{R}^{2n} \\ p_k \in \mathbb{R}^m \\ \eta \in [0,1]}} \{\mathbb{1}_{2n}^T y \mid [U^T, V^T]y \leq p_k,\ p_k \in \tau(Q_{k-1}^\eta)\}, \tag{9}$$

which equals the additional amount of quadratic cost that can be linearized in iteration k. Note that if the set $\tau(Q_{k-1})$ is a polyhedron, then $\tau(Q_{k-1}^\eta)$ is also a polyhedron and the corresponding problem (9) can be solved in polynomial time. For the sufficient conditions mentioned in Sect. 4 this is indeed the case.

Finally, we provide a new bounding procedure that is based on iteratively finding the best equivalent representation of the residual cost matrix and its optimal linearizable matrix. Starting with $Q_0 = Q$, the goal is to find the best linearizable matrix $\hat{Q}_{k-1}$ of an equivalent representation of the residual matrix Q_{k-1} and its corresponding linearization vector p_k. In each iteration we compute r_k by (9) and let $d_k = d_{k-1} + p_k$ which equals the total linearization vector of Q. The final bound is given by the sum of all r_k's, which we call the reformulation based bound. The procedure is given in Algorithm 1.

Algorithm 1 *LBB* Reformulation Algorithm

```
1: $Q_0 = Q, d_0 = \mathbf{0}, k = 1, r_0 = \infty$
2: while $r_{k-1} > 0$ do
3:     Compute $r_k$, $p_k$ and $\eta$ using (9).
4:     Construct the linearizable matrix $\hat{Q}_{k-1}$ using the optimal solution of (9).    ▷ See Remark 3
5:     $Q_k \leftarrow \eta Q_{k-1} + (1-\eta)Q_{k-1}^T - \hat{Q}_{k-1}$
6:     $d_k \leftarrow d_{k-1} + p_k$
7:     $k \leftarrow k + 1$
8: end while
9: $v_{RBB} = \sum_{i=1}^{k-1} r_i$
10: return $d_k, v_{RBB}$
```

Remark 3 Note that steps 3 and 4 of Algorithm 1 depend on the specific sufficient condition for linearizability. For instance, for the incident weak sum condition we construct in step 4 the linearizable matrix $\hat{Q}_{k-1}$ in the following way $(\hat{Q}_{k-1})_{ef} := b_e + c_f$ for all $e \in A$, $f \in \delta^+(e^-)$ and $(\hat{Q}_{k-1})_{ef} := 0$ otherwise, where $b, c \in \mathbb{R}^m$ are obtained from (9).

Hu and Sotirov (2019) show that in the case that the linearizable matrix $\hat{Q}$ is of the form $\hat{Q} = [U^T, V^T]Y + \mathrm{Diag}(z)$ for some $Y \in \mathbb{R}^{2n\times m}$ and $z \in \mathbb{R}^m$, the bound v_{RBB} is dominated by the solution of the first level RLT relaxation introduced by Adams and Sherali (1986, 1990). Here RLT stands for reformulation linearization technique. In Hu and Sotirov (2019) it is moreover shown that the first level RLT bound, denoted by v_{RLT1}, can be obtained by searching for the optimal linearizable matrix $\hat{Q}$ of the form $\hat{Q} = [U^T, V^T]Y + M + \mathrm{Diag}(z)$ where M is a skew symmetric matrix.

Our preliminary numerical results show that the above algorithm does not improve significantly the $LBB1$ bound. However, in the next section we show that our approach outperforms known iterative approaches related to the Gilmore–Lawler bounds.

6 The Gilmore–Lawler type bound

In this section we consider the classical Gilmore–Lawler type bound. The GL procedure is a well-known approach to construct lower bounds for quadratic 0-1 optimization problems, see e.g., Gilmore (1962), Lawler (1963), Rostami and Malucelli (2015) and Rostami et al. (2018). We provide a compact formulation of the GL type bound that can be used to compute the bound by a single LP-problem, instead of solving m subproblems. Moreover, we show that this bound in fact belongs to the family of linearization based bounds. Therefore, based on the results of Sect. 5, we provide a bounding procedure that computes the best GL type bound in each step of the algorithm. We conclude this section by testing this new bounding procedure on some preliminary test instances.

6.1 The classical GL type bound

In the objective function of the QCCP, see (1), we have the quadratic term $x_e x_f$ for each two arcs $e, f \in A$ placed in succession on a cycle. To get rid of this quadratic term, for each given arc $e \in A$, potentially in the solution, we consider the cycle cover containing e with minimum interaction cost with e. We denote this minimum contribution of arc e to a solution by z_e. In particular, for all $e \in A$ we have

$$z_e := \min\{Q_{e,:}x \mid x \in X, x_e = 1\}, \qquad (P_e)$$

where $Q_{e,:}$ denotes the eth row of the cost matrix Q. The feasible set of (P_e) equals the set of all node-disjoint cycle covers containing arc e. If this set is empty, then we set z_e equal to 0 since arc e cannot contribute to a cycle cover.

Let $z \in \mathbb{R}^m$ be the vector consisting of the elements z_e for all $e \in A$. Then the classical GL type bound is obtained by solving the following CCP:

$$v_{GL} := \min\{z^T x \mid x \in X\}. \qquad (GL)$$

Note that the constraint matrices of (P_e) and (GL) are totally unimodular. For this reason, we can drop the integrality constraints and solve (GL) and (P_e) for all $e \in A$ as linear programming problems.

Besides computing the GL type bound by solving (GL) and (P_e) for all $e \in A$, we can also obtain its value by solving an integer linear programming (ILP) problem. The problem (GL_{ILP}) is defined as follows:

$$
\begin{aligned}
(GL_{ILP}) \quad & \min \sum_{e\in A}\sum_{f\in A} Q_{ef}\, y_{ef} \\
& \text{s.t.} \sum_{f\in\delta^+(i)} y_{ef} = \sum_{f\in\delta^-(i)} y_{ef} = x_e \qquad \forall i \in N, \forall e \in A && (10) \\
& y_{ee} = x_e \qquad \forall e \in A && (11) \\
& y_{ef} \in \{0,1\},\ x \in X \qquad \forall e, f \in A && (12)
\end{aligned}
$$

It follows from the constraints that if $x_e = 1$, then $y_{e,:} := [y_{e1}, \dots, y_{em}]$ is the characteristic vector of the cheapest cycle cover containing arc e and if $x_e = 0$, then $y_{e,:}$ equals the zero vector.

Let (CGL_{ILP}) be the continuous relaxation of (GL_{ILP}). In this continuous relaxation we can omit the upper bounds on x_e and y_{ef} for all $e, f \in A$, since it is never optimal to set the value of these variables larger than one. We can compute the GL type bound by solving (CGL_{ILP}) as stated by the following theorem. This theorem is based on a similar result for the QMST, see Rostami and Malucelli (2015).

Theorem 3 *The optimal value of* (CGL_{ILP}) *equals* v_{GL}.

Proof Let $\lambda_{e,i}$ and $\alpha_{e,i}$ be the dual variables corresponding to constraints (10), i.e. $\lambda_{e,i}$ corresponds to $\sum_{f\in\delta^+(i)} y_{ef} = x_e$ and $\alpha_{e,i}$ corresponds to $\sum_{f\in\delta^-(i)} y_{ef} = x_e$. Similarly, let μ_i and γ_i be the dual variables corresponding to the first and second equalities of constraints (2), and θ_e the dual variable corresponding to constraints (11). The dual problem of (CGL_{ILP}) is as follows:

$$
\begin{aligned}
(DCGL_{ILP}) \quad & \max \sum_{i\in N} \mu_i + \sum_{i\in N} \gamma_i \\
& \text{s.t.} \quad \lambda_{e,f^+} + \alpha_{e,f^-} \le Q_{ef} \qquad \forall e, f \in A,\ f \ne e && (13) \\
& \lambda_{e,e^+} + \alpha_{e,e^-} + \theta_e \le Q_{ee} \qquad \forall e \in A && (14) \\
& -\sum_{i\in N} \lambda_{e,i} - \sum_{i\in N} \alpha_{e,i} + \gamma_{e^-} + \mu_{e^+} - \theta_e \le 0 \qquad \forall e \in A. && (15)
\end{aligned}
$$

Constraint (15) can be rewritten as $\gamma_{e^-} + \mu_{e^+} \le \sum_{i\in N} \lambda_{e,i} + \sum_{i\in N} \alpha_{e,i} + \theta_e$ for all $e \in A$. In order to maximize the objective function of $(DCGL_{ILP})$, we maximize the right hand side of this inequality subject to constraints (13)–(14). This gives for each $e \in A$ the following subproblem:

$$z'_e := \max \left\{ \sum_{i \in N} \lambda_{e,i} + \sum_{i \in N} \alpha_{e,i} + \theta_e \mid (13), (14) \right\}. \qquad (DCP_e)$$

For each fixed $e \in A$ the subproblem given above equals the dual of the continuous relaxation of (P_e). By strong duality we know $z'_e = z_e$ for all $e \in A$. Substitution of this term into constraint (15) gives a rewritten formulation for $(DCGL_{ILP})$:

$$\max \left\{ \sum_{i \in N} \mu_i + \sum_{i \in N} \gamma_i \mid \mu_{e^+} + \gamma_{e^-} \leq z_e \quad \forall e \in A \right\}.$$

This problem equals the dual of the continuous relaxation of (GL). Because of strong duality, it follows that the optimal objective value of (CGL_{ILP}) equals v_{GL}. □

We can show that the Gilmore–Lawler type bound for the QCCP in fact belongs to the family of linearization based bounds introduced in Sect. 4. That is, we can obtain v_{GL} by searching for a linearizable quadratic cost matrix $\hat{Q}$ of a specific type that is as close as possible to Q. The required linearizability condition on $\hat{Q}$ is given below, and it differs from the sufficient conditions presented in Sect. 3.

Proposition 7 *If there exists $B, C \in \mathbb{R}^{m \times n}$ and $t \in \mathbb{R}^m$ such that $Q_{ef} = B_{e,f^+} + C_{e,f^-}$ for $e \neq f$ and $Q_{ee} = B_{e,e^+} + C_{e,e^-} + t_e$ for all $e \in A$, then Q is linearizable with vector p given by $p_e = t_e + \sum_{i=1}^n B_{e,i} + \sum_{i=1}^n C_{e,i}$.*

Proof Let $\tilde{Q}$ be defined as $\tilde{Q}_{ef} = B_{e,f^+} + C_{e,f^-}$ for all $e, f \in A$. Then $\tilde{Q}$ can be seen as a generalized weak sum matrix where D and T are equal to the zero matrix, see Definition 6. According to Proposition 5, $\tilde{Q}$ is linearizable with vector $\tilde{p} = \sum_{i=1}^n B_{e,i} + \sum_{i=1}^n C_{e,i}$. Since $Q = \tilde{Q} + \text{Diag}(t)$, it follows that Q is linearizable with vector p given by $p_e = t_e + \sum_{i=1}^n B_{e,i} + \sum_{i=1}^n C_{e,i}$. □

Similar to the notation used in Sect. 4, let $S_{GL}(Q)$ denote the set of all linearizable cost matrices $\hat{Q} \in \mathbb{R}^{m \times m}$ that satisfy the conditions of Proposition 7. Moreover, let $\tau_{GL}(Q)$ be the following polyhedron:

$$\tau_{GL}(Q) := \{\hat{p} \in \mathbb{R}^m \mid x^T \hat{p} = x^T \hat{Q} x \text{ for all } x \in X, \ \hat{Q} \in S_{GL}(Q)\}, \qquad (16)$$

and

$$v_{LBB}^{GL} := \max_{\substack{y \in \mathbb{R}^{2n} \\ \hat{p} \in \mathbb{R}^m}} \{\mathbb{1}_{2n}^T y \mid [U^T, V^T] y \leq \hat{p}, \ \hat{p} \in \tau_{GL}(Q)\}. \qquad (17)$$

Now we prove the main result of this section which states that the classical Gilmore–Lawler type bound can be seen as a special case of linearization based bound.

Theorem 4 *We have $v_{LBB}^{GL} = v_{GL}$.*

Proof By using the polyhedral description of $S_{GL}(Q)$ following from Proposition 7, the optimization problem in (17) can be written as follows:

$$
\begin{aligned}
v_{LBB}^{GL} = \max \quad & \sum_{i=1}^{2n} y_i && (18)\\
\text{s.t.} \quad & [U^T, V^T]y \leq \hat{p} && (19)\\
& B_{e,f^+} + C_{e,f^-} \leq Q_{ef} \qquad \forall e, f \in A, f \neq e && (20)\\
& B_{e,e^+} + C_{e,e^-} + t_e \leq Q_{ee} \qquad \forall e \in A && (21)\\
& \hat{p}_e = t_e + \sum_{i=1}^{n} B_{e,i} + \sum_{i=1}^{n} C_{e,i} \qquad \forall e \in A && (22)\\
& y \in \mathbb{R}^{2n}, \hat{p}, t \in \mathbb{R}^m, B, C \in \mathbb{R}^{m \times n}. && (23)
\end{aligned}
$$

We show that this optimization problem is equivalent to $(DCGL_{ILP})$, the dual problem of the continuous relaxation of (GL_{ILP}). Take $B_{e,i} = \lambda_{e,i}$ and $C_{e,i} = \alpha_{e,i}$ for all $e \in A$ and $i \in N$, where λ and α denote the dual vectors belonging to constraints (10). Similarly, let $t = \theta$ where θ equals the dual vector to constraints (11). Finally, let $y = [\mu^T, \gamma^T]^T$, where μ and γ are the dual variables belonging to constraints (2). By substitution of these variables and combining constraints (19) and (22), we obtain the problem $(DCGL_{ILP})$, i.e., the dual of (CGL_{ILP}). Thus, we have $v_{LBB}^{GL} = v_{GL}$. □

Theorem 4 shows that the GL type bound belongs to the family of linearization based bounds. This is also shown by Hu and Sotirov (2019) and Rostami et al. (2018), however our proof is very different as we exploit the fact that v_{GL} can be obtained by solving an LP problem, i.e., (CGL_{ILP}). Additionally, we show here that the computation of the GL type bound is equivalent to the search for the optimal linearizable cost matrix $\hat{Q}$ satisfying the properties of Proposition 7.

6.2 The best Gilmore–Lawler type bound

Section 6.1 shows that the calculation of the classical GL type bound fits in the general framework discussed in Sect. 4. In this section we apply the reformulation procedure of Sect. 5 to the GL type bound. We also show that our approach outperforms several iterative approaches from the literature.

In order to apply Algorithm 1 to the sufficient condition for linearizability of Proposition 7, we need to define how to calculate r_k for each iteration k, see (9). We rewrite the set $\tau_{GL}(Q)$, see (16), as follows:

$$\tau_{GL}(Q) = \{\hat{p} \in \mathbb{R}^m \mid t \in \mathbb{R}^m, B, C \in \mathbb{R}^{m \times n}, (20), (21), (22)\},$$

which is clearly a polyhedron. Then for all $k \geq 1$ we calculate the additional amount of quadratic cost that is linearized by solving:

$$r_k := \max_{\substack{y\in\mathbb{R}^{2n} \\ p_k\in\mathbb{R}^m \\ \eta\in[0,1]}} \{\mathbb{1}_{2n}^T y \mid [U^T, V^T]y = p_k,\, p_k \in \tau_{GL}(Q_{k-1}^{\eta})\}. \tag{24}$$

Observe that as opposed to the constraints in (9), we have replaced the constraint $[U^T, V^T]y \leq p_k$ by an equality constraint. This does not change the value of r_k. To verify this, suppose we solve (24) using the inequality constraint $[U^T, V^T]y \leq p_k$ and let $\hat{y}$, $\hat{p}_k$ and $\hat{t}$ be the corresponding optimal solutions. Let $e \in A$ be such that the inequality constraint is satisfied with strict inequality. Then without changing $\hat{y}$, we can reduce $\hat{t}_e$ (and thus $\hat{p}_e$) such that we get equality for $e \in A$. Although it changes the linearization vector $\hat{p}$, the resulting bound remains equal. To verify this, notice that only the left hand side of constraint (21) is decreased, so the solution is still feasible and the optimal value r_k remains unchanged. From this, it follows that one may replace $[U^T, V^T]y \leq p_k$ by an equality constraint and solve r_k as in (24).

Algorithm 1 using (24) to compute r_k, p_k and η gives a new bounding procedure for the QCCP. We call the resulting bound the reformulated GL type bound (RGL) and denote its value by v_{RGL}. By construction, it iteratively computes the best Gilmore–Lawler type bound among all equivalent representations of the quadratic cost matrix.

The algorithm proposed in this section satisfies another interesting property, namely the vectors d_k satisfy the constant value property for all $k \geq 0$. This is an important property for linearizability because the set of linearizable cost matrices for combinatorial optimization problems with interaction costs can be characterized by the constant value property, under certain conditions, see Lendl et al. (2019).

Theorem 5 *All d_k where $k \geq 0$ computed during the RGL approach, satisfy the constant value property, i.e., we have $x^T d_k = \bar{x}^T d_k$ for all feasible cycle covers $x, \bar{x} \in X$.*

Proof We apply a proof by induction on k. Note that the vector d_0 equals the $m \times 1$ vector of zeros which trivially satisfies the constant value property.

Now assume that the induction hypothesis is true for iteration $k-1$, i.e., $x^T d_{k-1} = \bar{x}^T d_{k-1}$ for all feasible cycle covers $x, \bar{x} \in X$. In iteration k we solve (24). Let $\hat{y} \in \mathbb{R}^{2n}$ and $\hat{p} \in \mathbb{R}^m$ be the optimal variables for this problem and split $\hat{y} = [\mu^T, \lambda^T]^T$ with $\mu, \lambda \in \mathbb{R}^n$. It follows that $[U^T, V^T]\hat{y} = U^T\mu + V^T\lambda = \hat{p}$. Now let $x \in X$ be any feasible cycle cover in G. Then we can sum up the rows of this system of equalities for all arcs $e \in A$ in the cycle cover implied by x:

$$\sum_{\{e\in A: x_e=1\}} (\mu_{e^+} + \lambda_{e^-}) = \sum_{\{e\in A: x_e=1\}} \hat{p}_e \quad \Leftrightarrow \quad \sum_{i\in N} \mu_i + \sum_{i\in N} \lambda_i = x^T \hat{p}$$

where we use the fact that each vertex is visited exactly once on a cycle cover. So the quantity $x^T\hat{p}$ is equal for all $x \in X$. As a result, $\hat{p}$ satisfies the constant value property.

The vector d_k is constructed as $d_{k-1} + p_k$ with $p_k = \hat{p}$. Since d_{k-1} and $\hat{p}$ satisfy the constant value property, it follows that d_k satisfies the constant value property. □

Remark 4 Since the GL type bound can be computed both as a linearization based bound and by solving (CGL_{ILP}) (see Theorem 4), the iterative approach derived in

Table 3 Bounds and computation times in seconds of GL type bounds on Erdős–Rényi instances

p	n	m	OPT	GL		RGL^{sym}		RGL	
				Bound	Time	Bound	Time	Bound	Time
0.1	20	44	923	923	0.017	923	0.015	923	0.088
	25	76	1039	681	0.039	864	4.652	1018	95.52
	30	100	1082	781	0.053	899	2.412	1082	15.18
0.3	15	61	485	347	0.061	368	1.481	485	7.140
	20	118	438	223	0.068	263	3.482	418	3600
	25	172	382	176	0.136	190	5.105	276	3600
0.5	15	116	226	102	0.336	110	2.602	222	1835
	20	177	255	93	0.118	103	5.365	169	3600
	25	306	n.a.	66	0.296	75	13.80	105	3600
0.7	15	149	173	63	0.080	67	3.530	117	1236
	20	263	n.a.	52	0.181	54	9.624	63	269.3
	25	396	n.a.	56	0.365	62	18.77	79	1085

this section can also be defined in terms of (CGL_{ILP}). In that case, we iteratively compute v_{GL} and reformulate the quadratic cost matrix using the dual variables of (CGL_{ILP}). The details of this equivalent approach can be found in Meijer (2018).

Since the linearizable matrix $\hat{Q}$ of Proposition 7 can be written in the form $\hat{Q} = [U^T, V^T]Y + \text{Diag}(z)$ for some $Y \in \mathbb{R}^{2n\times m}$ and $z \in \mathbb{R}^m$, it follows from Hu and Sotirov (2019) that we have $v_{RGL} \leq v_{RLT1}$.

6.3 Preliminary results

For the instances considered in the preliminary results of Sect. 4, we now test our Gilmore–Lawler type bounds. First, we compute the classical GL type bound, after symmetrizing the quadratic cost matrix Q. This bound is denoted by GL. Moreover, we consider the iterative GL type bounding approach where we symmetrize the quadratic cost matrix in each iteration. That is, we apply Algorithm 1 using (24) where instead of optimizing over η, we set $\eta = \frac{1}{2}$. We denote this bound by RGL^{sym}. Finally, we report the bound RGL which is introduced in Sect. 6.2. The maximum computation time is set at 3600 s. The results are given in Table 3.

From Table 3 it follows that the iterative approaches significantly improve the classical GL type bound. Among these iterative approaches, RGL provides much stronger bounds than RGL^{sym}. We conclude that this new approach of calculating the best GL type bound in each step provides better bounds than when setting $\eta = \frac{1}{2}$ in the reformulation. However, it turns out that this improvement in the quality comes at the cost of efficiency. Clearly, we can stop our algorithm after a pre-specified number of iterations and/or time.

7 Other bounds for the QCCP

In this section we present several known bounding approaches from the literature that can be applied to the QCCP. In the next section, we compare those bounds with the bounds introduced in this paper. We consider a column generation approach and a mixed integer linear programming (MILP) based bound.

Galbiati et al. (2014) construct a column generation approach for the MinRC3. This approach can be extended to the QCCP. To the best of our knowledge, this is the only implemented lower bounding approach for the MinRC3 in the literature.

Let $\mathcal{C}$ be the set of all directed cycles in G. Moreover, let $\overline{\mathcal{C}} \subseteq \mathcal{C}$ be a subset of cycles such that it contains at least one cycle cover. Let w_c be the cost of a cycle $c \in \mathcal{C}$. Then the restricted master problem (RMP) is given by:

$$\begin{aligned} (RMP)\ \min_{y} \quad & \sum_{c \in \overline{\mathcal{C}}} w_c y_c \\ \text{s.t.} \quad & \sum_{c \in \overline{\mathcal{C}} : i \in c} y_c = 1 && \forall i \in N && (25) \\ & y_c \geq 0 && \forall c \in \overline{\mathcal{C}}. && (26) \end{aligned}$$

Let $\pi \in \mathbb{R}^n$ be the vector of dual variables corresponding to constraint (25). Then the subproblem (SP) searches for the cycle in $\mathcal{C}$ with the smallest (negative) reduced costs with respect to π, i.e.

$$(SP) \qquad \min_{x,z} \{ x^T Q x - z^T \pi \mid \sum_{e \in \delta^+(i)} x_e = \sum_{e \in \delta^-(i)} x_e = z_i \quad \forall i \in N, \\ \sum_{e \in A} x_e \geq 2,\ x \in \{0,1\}^m,\ z \in \{0,1\}^n \},$$

where $z_i = 1$ if vertex i is on the cycle and 0 otherwise. As stated in Galbiati et al. (2014), the problem (SP) is strongly NP-hard itself. The quadratic objective function can be linearized by standard linearization techniques. A lower bound on the optimal value of the QCCP can be obtained by iteratively solving the master problem and its corresponding subproblem. If a cycle with negative reduced cost is found, we add it to the set $\overline{\mathcal{C}}$. This procedure is repeated until no more cycle with negative reduced cost is found or after some predefined stopping criteria. The obtained bound is denoted by v_{CG}.

Based on a procedure by Glover (1975) and Adams et al. (2004), we present the QCCP as an MILP problem. Let us first fix an equivalent representation of (G, Q). Let z_e be computed as in (P_e) for all $e \in A$, see Sect. 6.1. Moreover, we define for all $e \in A$

$$q_e^{\max} := \max\{ Q_{e,:} x \ :\ x \in X, x_e = 0 \}.$$

Note that $q_e^{\max}$ can be obtained by solving a linear programming problem. Then the QCCP can be formulated as an MILP:

$$
\begin{aligned}
(MILP)\ \min_{x,y}\ & \sum_{e \in A} y_e && && (27)\\
\text{s.t.}\ \ & y_e \geq z_e x_e && \forall e \in A &&\\
& y_e \geq Q_{e,:}x - q_e^{\max}(1 - x_e) && \forall e \in A && (28)\\
& x \in X,\ y \in \mathbb{R}^m. &&&&
\end{aligned}
$$

If we relax the binary constraint on x, then we obtain a lower bound for the QCCP. We call this bound the MILP based bound and we denote its value by v_{MILP}. The next result shows that v_{MILP} is at least as large as the Gilmore–Lawler type bound.

Theorem 6 *The MILP based bound dominates the Gilmore–Lawler type bound, i.e., $v_{GL} \leq v_{MILP}$.*

Proof Let $\beta, \delta \in \mathbb{R}_+^m$ denote the dual variables of (27) and (28), respectively. Moreover, let $\mu, \gamma \in \mathbb{R}^n$ denote the dual variables of the cycle cover constraints $\sum_{e \in \delta^+(i)} x_e = 1$ and $\sum_{e \in \delta^-(i)} x_e = 1$ for all $i \in N$, respectively. Then the dual of the MILP based bound equals

$$
\begin{aligned}
(DMILP)\ v_{MILP} := \max_{\beta,\delta,\mu,\gamma}\ & \textstyle\sum_{i \in N} \mu_i + \sum_{i \in N} \gamma_i - \sum_{e \in A} \delta_e q_e^{\max} && \\
\text{s.t.}\ \ & \beta_e + \delta_e = 1 && \forall e \in A\\
& \mu_{e^+} + \gamma_{e^-} \leq z_e \beta_e + \delta^T Q_{:,e} + \delta_e q_e^{\max} && \forall e \in A\\
& \beta_e, \delta_e \geq 0 && \forall e \in A,
\end{aligned}
$$

where $Q_{:,e}$ equals the eth column of Q. Now set $\delta_e = 0$ for all $e \in A$. Then, $\beta_e = 1$ for all $e \in A$ due to the first set of constraints above. Then, $(DMILP)$ reduces to

$$
\begin{aligned}
\max_{\mu,\gamma}\ & \sum_{i \in N} \mu_i + \sum_{i \in N} \gamma_i \\
\text{s.t.}\ \ & \mu_{e^+} + \gamma_{e^-} \leq z_e \qquad \forall e \in A.
\end{aligned}
$$

This problem equals the dual of the continuous relaxation of (GL). Hence, it follows that $v_{GL} \leq v_{MILP}$. □

Note the MILP based bound and the Gilmore–Lawler type bound are comparable if the same equivalent reformulation of (G, Q) is used in their computations.

8 Numerical results

In this section we test our bounding approaches on a set of test instances and compare them with several approaches from the literature. We take into account the linearization based bound $LBB1$ from Sect. 4.1, the classical GL type bound GL from Sect. 6.1, the reformulated GL type bound RGL discussed in Sect. 6.2, the column generation approach CG and the MILP based bound $MILP$ from Sect. 7, and the first level RLT bound $RLT1$, see Adams and Sherali (1986, 1990). The GL bound and the MILP based bound are computed after symmetrizing Q. Note that we do not take into account $LBB2$ and $LBB3$, since our preliminary experiments from Sect. 4.2 show that $LBB1$ is preferred when taking both quality and efficiency into account.

All lower bounds are implemented in Matlab on a pc with an Intel(R) Core(TM) i5-6500 CPU, 3.20 GHz and 8 GB memory using CPLEX 12.6 as the solver.
We consider the following types of instances:

- *Erdős-Rényi instances* These instances are created via the $G(n, p)$ Erdős–Rényi model (Erdős and Rényi 1959). The number of nodes is fixed to n and each arc is included with probability p independent of the other arcs. The quadratic cost between any pair of successive arcs is chosen discrete uniformly at random out of $\{0, \dots, 100\}$.
- *Manhattan instances* The Manhattan instances are introduced in Comellas et al. (2008) and resemble the street pattern of modern cities like Manhattan. Given a finite set of positive integers $(n_1, n_2, \dots, n_k)$, the graph consists of a $n_1 \times n_2 \times \cdots \times n_k$ directed grid. Each node in the interior is adjacent to its $2k$ neighbours. The nodes on the boundary are also incident to the corresponding nodes on the opposite boundary. For each dimension k, the arcs belonging to the same layer of the grid point in the same direction. However, the arcs of two consecutive layers point in the opposite direction. This results in a graph containing a large number of cycles. The quadratic cost between any pair of successive arcs is chosen discrete uniformly at random out of $\{0, \dots, 10\}$.
- *Angle-distance instances* The Angle-distance instances are originally constructed for the QTSP in Fischer (2013). The number of nodes n and the graph density p are given. The (x, y)-coordinates of each node is chosen discrete uniformly at random out of $\{0, \dots, 500\}^2$. Exactly $\lceil pn(n-1) \rceil$ arcs are chosen uniformly at random from the total set of arcs. For each arc $e \in A$, let d_e denote the Euclidean distance between the endpoints of e. Moreover, for each two successive arcs e and f, let α_{ef} denote the turning angle (in radians). Given some constant $\rho \in \mathbb{R}_+$, the quadratic cost of two successive arcs e and f is calculated as:

$$Q_{ef} = \left\lceil 0.1 \left(\rho \cdot \alpha_{ef} + \frac{d_e + d_f}{2} \right) \right\rceil .$$

Similar as in Fischer (2013), we take $\rho = 40$.

For Erdős–Rényi and Angle-distance instances, preliminary experiments show that instances up to approximately 300 arcs can be solved to optimality within one hour. For the Manhattan instances the limit is around 2000 arcs, due to the small density of these types of graphs.

In total we consider two sets of experiments: experiments on small instances and experiments on large instances. Since the optimum, $RLT1$ and CG cannot be calculated for large instances, we only test these approaches on the smaller instances. Moreover, we include the bounds introduced in this paper, namely $LBB1$ and RGL. The value and computation times (in seconds) on small Erdős–Rényi instances can be

Table 4 Bounds and computation times in seconds of $RLT1$, CG, $LBB1$ and RGL on small Erdős–Rényi instances

p	n	m	OPT		$RLT1$		CG		$LBB1$		RGL	
			Value	Time	Value	Time	Value	Time	Value	Time	Value	Time
0.3	20	119	319	10.28	301	4.825	289	102.3	260	0.020	285	3600
	25	177	386	19.04	331	20.09	331	928.9	305	0.040	280	3600
	30	280	n.a.	3600	284	70.62	n.a.	3600	274	0.134	185	3600
0.5	20	195	236	211.0	181	17.00	n.a.	3600	175	0.121	129	3600
	25	327	n.a.	3600	141	82.52	n.a.	3600	136	0.233	89	3600
	30	442	n.a.	3600	168	385.0	n.a.	3600	162	0.322	99	3600

found in Table 4. This table contains 6 instances for $n = 20, 25, 30$ and $p = 0.3, 0.5$. The results on Manhattan and Angle-distance instances are reported in Tables 5 and 6 , respectively. For the Angle-distance instances we take the same values for n and p as for the Erdős-Rényi instances, while for the Manhattan instances we consider several two- and three-dimensional instances. The maximum computation time is set to 3600 s. When after this time no bound is computed, we report 'n.a.' in the tables. Since the optimal value is always integer, we round up all bounds.

For the smaller instances, we see that $RLT1$ performs best in quality. When it can be computed, it is often close to the optimal value and it dominates the other bounds. $LBB1$ is often very close to $RLT1$, but can be computed much more efficiently. Namely, for the Erdős–Rényi and the Angle-distance instances the computation time of $LBB1$ for all small instances is below 0.4 s, whereas $RLT1$ cannot be computed within one hour for some of these instances. The column generation approach provides strong bounds, but in most cases it is not able to compute a bound in a time span of one hour. The reformulated GL type bound performs well on the Manhattan and Angle-distance instances, see Tables 5 and 6 . Although its total computation time is large, the advantage of this approach is that it provides a bound in a short time and then iteratively improves the value. This makes it possible to stop the procedure at any given time and still obtain a bound. The bounds computed by RGL are in almost all cases dominated by $LBB1$.

When taking both efficiency and quality into account, we conclude that the linearization based bound $LBB1$ outperforms the other approaches. Based on Tables 4, 5 and 6 , the value of $LBB1$ is at least 75% of the optimal value for the Erdős-Rényi instances. For the Angle-distance and Manhattan instances, this percentage equals 98% and 96%, respectively.

For the larger instances, we only compute the bounds that can be computed efficiently. That is, we do not consider the iterative approaches, but only the bounds GL, $MILP$ and $LBB1$. We also investigate the effect of a reformulation by adding an optimal incident skew symmetric matrix to the cost matrix, see Sect. 4.1. We apply this reformulation to $LBB1$, which implies that we optimize over the set $\tau_1^{skew}(Q)$, see (7), instead of $\tau_1(Q)$. The resulting bound is denoted by $LBB1^{skew}$. For the Manhattan instances this bound is omitted, since preliminary experiments showed that this

Table 5 Bounds and computation times in seconds of $RLT1$, CG, $LBB1$ and RGL on small Manhattan instances

Instance	n	m	OPT		$RLT1$		CG		$LBB1$		RGL	
			Value	Time	Value	Time	Value	Time	Value	Time	Value	Time
(5, 5)	25	50	103	0.483	103	1.534	103	1.484	103	0.006	103	5.756
(10, 10)	100	200	418	2.335	418	1.974	418	1645	418	0.022	371	16.06
(4, 4, 4)	64	192	199	6.312	193	9.415	196	691.4	193	0.081	175	3600
(6, 6, 6)	216	648	700	23.67	683	1152	n.a.	3600	683	0.568	551	3600
(8, 8, 8)	512	1536	1566	394.1	n.a.	3600	n.a.	3600	1530	1.213	n.a.	3600
(10, 10, 10)	1000	3000	n.a.	3600	n.a.	3600	n.a.	3600	3113	3.754	n.a.	3600

Table 6 Bounds and computation times in seconds of $RLT1$, CG, $LBB1$ and RGL on small Angle-distance instances

p	n	m	OPT		$RLT1$		CG		$LBB1$		RGL	
			Value	Time	Value	Time	Value	Time	Value	Time	Value	Time
0.3	20	114	474	2.490	474	1.719	474	416.5	474	0.002	474	64.35
	25	180	553	323.9	553	4.119	n.a.	3600	552	0.004	553	1559
	30	261	512	2951.0	512	19.52	n.a.	3600	512	0.079	494	3600
0.5	20	190	276	177.8	276	6.732	n.a.	3600	276	0.053	274	1319
	25	300	342	2163.6	340	53.43	n.a.	3600	338	0.142	320	3600
	30	435	n.a.	3600	381	490.5	n.a.	3600	377	0.332	355	3600

reformulation does not improve the bounds for most Manhattan instances. This could be due to the sparsity of Manhattan instances. The bounds and computation times (in seconds) for the Erdős-Rényi, Manhattan and Angle-distance instances are reported in Tables 7, 8 and 9 , respectively. For the Erdős-Rényi and Angle-distance instances we take for n values between 30 and 100 nodes and consider $p = 0.3$ and $p = 0.5$. For the Manhattan instances we consider large two-dimensional instances and one large three-dimensional instance. The maximum computation time for these bounds is set to 1800 s. Again, we round up all bound values.

For the larger instances, we see that $LBB1$ in all cases dominates GL and $MILP$ in both quality and efficiency. The difference in quality is most present for the Erdős-Rényi instances, see Table 7. For the Manhattan instances, we see that GL and $MILP$ can be calculated efficiently for instances up to 3000 arcs. However, $LBB1$ remains efficient even for larger instances. In particular, bounds for Manhattan instances up to 15,000 arcs can be computed within 60 s.

Moreover, we conclude from Tables 7 and 9 that the addition of an incidence skew symmetric matrix to the set $\tau_1(Q)$ only improves the bounds for some of the instances. In general, it turns out that the Erdős-Rényi instances can successfully be improved by this method, whereas for the Angle-distance instances only in a few cases there is an improvement. Although the computation times of $LBB1^{skew}$ are larger than those of $LBB1$, bounds can still be computed in a reasonable time span.

Table 7 Bounds and computation times in seconds of GL, $MILP$, $LBB1$ and $LBB1^{skew}$ on large Erdős–Rényi instances

p	n	m	GL		$MILP$		$LBB1$		$LBB1^{skew}$	
			Value	Time	Value	Time	Value	Time	Value	Time
0.3	30	284	111	0.272	122	0.435	230	0.083	232	0.766
	40	468	117	0.645	131	1.006	265	0.179	278	1.711
	50	754	121	1.598	130	2.410	267	0.404	274	4.184
	60	1062	103	4.068	118	5.788	272	0.726	272	8.048
	70	1481	114	8.910	123	12.94	255	1.660	258	15.38
	80	1842	113	14.26	122	20.82	263	2.740	267	24.67
	90	2385	114	23.25	122	37.61	259	5.296	261	41.74
	100	2962	119	36.03	126	63.49	269	13.32	270	69.90
0.5	30	434	73	0.557	79	0.783	161	0.182	163	9.218
	40	793	69	1.607	74	2.364	166	0.554	169	10.38
	50	1197	72	4.323	77	6.682	165	1.185	167	15.74

Table 8 Bounds and computation times in seconds of GL, $MILP$ and $LBB1$ on large Manhattan instances

Instance	n	m	GL		$MILP$		$LBB1$	
			Value	Time	Value	Time	Value	Time
(20, 20)	400	800	1237	5.31	1472	7.491	1537	0.100
(30, 30)	900	1800	2813	56.24	3343	86.46	3517	0.410
(40, 40)	1600	3200	5101	346.8	6028	553.5	6302	1.388
(50, 50)	2500	5000	7983	1225.3	9424	1897.8	9828	2.838
(17, 17, 17)	4913	14739	n.a.	1800	n.a.	n.a.	15398	54.79

Table 9 Bounds and computation times in seconds of GL, $MILP$, $LBB1$ and $LBB1^{skew}$ on Angle-distance instances

p	n	m	GL		$MILP$		$LBB1$		$LBB1^{skew}$	
			Value	Time	Value	Time	Value	Time	Value	Time
0.3	30	261	456	0.238	467	0.410	525	0.054	525	6.957
	40	468	507	0.693	516	0.984	567	0.463	567	7.795
	50	735	622	1.567	631	2.223	709	0.317	709	9.982
	60	1062	609	3.684	618	5.401	684	0.694	684	13.97
	70	1449	656	7.436	666	12.37	746	1.331	747	21.63
	80	1896	749	13.03	756	23.77	867	2.613	867	31.96
	90	2403	815	20.33	826	39.99	933	4.838	933	48.81
	100	2970	810	30.35	823	66.04	951	12.50	952	78.62
0.5	30	435	339	0.516	343	0.876	373	0.168	373	16.59
	40	780	411	1.456	418	2.386	464	0.474	464	16.21
	50	1225	466	4.159	473	7.550	534	1.177	535	22.34

9 Conclusion

In this paper we consider the linearization problem for the QCCP and its applications. We provide several sufficient conditions for linearizability, and show how these conditions can be used to obtain strong lower bounds for the QCCP. The linearization based bound $LBB1$, resulting from the incident weak sum property, is the most efficient LBB in terms of complexity and quality, see Table 2. We show here that the GL type bound for the QCCP also belongs to the family of linearization based bounds, see Theorem 4, by providing the appropriate sufficient condition, see Proposition 7.

The first level RLT bounds and/or the GL type bounds are the only linearization based bounds for quadratic binary optimization problems that are implemented for various binary quadratic optimization problems up to date. This paper shows that besides these two well-known bounds, the linearization based bounds introduced here are worth considering.

Here, we also present how each sufficient condition can be used in an iterative bounding procedure. In particular, we introduce a new reformulation technique in which we search for the best equivalent representation of the residual cost matrix and its optimal linearizable matrix, see Algorithm 1. We show how the resulting iterative procedure computes the best GL type bound in each iteration. Our approach outperforms known iterative bounding procedures that use the GL type bounds, see Table 3. Moreover, we prove that the resulting linearization vectors in each step satisfy the constant value property, see Theorem 5.

Finally, our numerical results show that our approach outperforms several other bounds from the literature if we take into account both quality and efficiency. Although the linearization based bounds $LBB1$ are dominated by the well known first level RLT bounds, they can be computed extremely fast. For the Manhattan instances, $LBB1$ bounds for instances up to 15,000 arcs can be computed within 60 s. However, other approaches fail to provide bounds for instances of this large sizes.

We expect that similar bounding procedures can be successfully applied to other binary quadratic optimization problems, such as the quadratic assignment problem, the quadratic minimum spanning tree, and the quadratic traveling salesman problem. However, this is a subject of our future research.

Acknowledgements We would like to thank two anonymous reviewers for their insightful comments and suggestions to improve an earlier version of this work.

References

Adams WP, Forrester RJ, Glover FW (2004) Comparisons and enhancement strategies for linearizing mixed 0–1 quadratic programs. Discrete Optim 1(2):99–120

Adams WP, Sherali HD (1986) A tight linearization and an algorithm for zero–one quadratic programming problems. Manag Sci 32(10):1274–1290

Adams WP, Sherali HD (1990) Linearization strategies for a class of zero–one mixed integer programming problems. Oper Res 38(2):217–226

Aggarwal A, Coppersmith D, Khanna S, Motwani R, Schieber B (1999) The angular-metric traveling salesman problem. SIAM J Comput 29:697–711

Amaldi E, Galbiati G, Maffioli F (2011) On minimum reload cost paths, tours and flows. Networks 57:254–260

Burkard R, Dell'Amico M, Martello S (2009) Assignment problems. SIAM, Philadelphia

Büyükçolak Y, Gözüpek D, Özkan S (2019) On minimum reload cost paths, tours and flows. Networks 74(3):274–286

Carraresi P, Malucelli F (1992) A new lower bound for the quadratic assignment problem. Oper Res 40:22–27

Chiba S, Yamashita T (2018) On directed 2-factors in digraphs and 2-factors containing perfect matchings in bipartite graphs. SIAM J Discrete Math 32(1):394–409

Comellas FCD, Fiol MA (2008) Multidimensional manhattan street networks. SIAM J Discrete Math 22(4):1428–1447

Ćustić A, Punnen AP (2018) A characterization of linearizable instances of the quadratic minimum spanning tree problem. J Comb Optim 35(2):436–453

de Meijer F (2018) Bounds on the minimum reload cost cycle cover problem. Master's thesis, Tilburg University

Erdős P, Rényi A (1959) On random graphs. Publ Math 6(2):197–290

Fischer A (2013) A polyhedral study of quadratic traveling salesman problems. PhD thesis, Chemnitz University of Technology

Fischer A, Fischer F, Jäger G, Keilwagen J, Molitor P, Grosse I (2014) Exact algorithms and heuristics for the quadratic traveling salesman problem with an application in bioinformatics. Discrete Appl Math 166:87–114

Fischer F, Jäger G, Lau A, Molitor P (2009) Complexity and algorithms for the traveling salesman problem and the assignment problem of second order. Lect Notes Comput Sci 5165:211–224

Galbiati G (2008) The complexity of a minimum reload cost diameter problem. Discrete Appl Math 156:3494–3497

Galbiati G, Gualandi S, Maffioli F (2014) On minimum reload cost cycle cover. Discrete Appl Math 164:112–120

Gamvros I, Gouveia L, Raghavan S (2012) Reload cost trees and network design. Networks 59:365–379

Gilmore P (1962) Optimal and suboptimal algorithms for the quadratic assignment problem. J Soc Ind Appl Math 10(2):305–313

Glover F (1975) Improved linear integer programming formulations of nonlinear integer problems. Manag Sci 22(4):455–460

Gourvès L, Lyra A, Martinhon C, Monnot J (2010) The minimum reload s-t path, trail and walk problems. Discrete Appl Math 158(13):1404–1417

Hu H, Sotirov R (2018) Special cases of the quadratic shortest path problem. J Comb Optim 35(3):754–777

Hu H, Sotirov R (2019) The linearization problem of binary quadratic problems and its applications. arXiv:1802.02426

Jäger G, Molitor P (2008) Algorithms and experimental study for the traveling salesman of second order. Lect Notes Comput Sci 5165:211–224

Kabadi SN, Punnen AP (2011) An $\mathcal{O}(n^4)$ algorithm for the QAP linearization problem. Math Oper Res 36:754–761

Koopmans TC, Beckmann MJ (1957) Assignment problems and the location of economic activities. Econometrica 25:53–76

Lawler EL (1963) The quadratic assignment problem. Manag Sci 9(4):586–599

Lendl S, Ćustić, Punnen AP (2019) Combinatorial optimization problems with interaction costs: complexity and solvable cases. Discrete Optim 33:101–117

Punnen AP, Kabadi SN (2013) A linear time algorithm for the Koopmans Beckmann QAP linearization and related problems. Discrete Optim 10(3):200–209

Punnen AP, Pandey P, Friesen M (2019) Representations of quadratic combinatorial optimization problems: a case study using quadratic set covering and quadratic knapsack problems. Comput Oper Res 112:104769

Punnen AP, Walter M, Woods BD (2018) A characterization of linearizable instances of the quadratic traveling salesman problem. arXiv: 1708.07217v3

Rostami B, Chassein A, Hopf M, Frey D, Buchheim C, Malucelli F, Goerigk M (2018) The quadratic shortest path problem: complexity, approximability and solution methods. Eur J Oper Res 268(2):473–485

Rostami B, Malucelli F (2015) Lower bounds for the quadratic minimum spanning tree problem based on reduced cost computation. Comput Oper Res 64:178–188

Sahni S, Gonzalez T (1976) P-complete approximation problems. J ACM 23:555–565

Staněk R, Greistorfer P, Ladner K, Pferschy U (2019) Geometric and lp-based heuristics for angular travelling salesman problems in the plane. Comp Oper Res 108:97–111

Wirth H, Steffan J (2001) Reload cost problems: minimum diameter spanning tree. Discrete Appl Math 113:73–85