Vietnam Journal of Mathematics
https://doi.org/10.1007/s10013-024-00694-y

ORIGINAL ARTICLE

The Quadratic Minimum Spanning Tree Problem: Lower Bounds via Extended Formulations

Renata Sotirov[1] · **Zoe Verchére**[2]

Received: 10 November 2023 / Accepted: 11 March 2024

Abstract
The quadratic minimum spanning tree problem (QMSTP) is the problem of finding a spanning tree of a graph such that the total interaction cost between pairs of edges in the tree is minimized. We first show that the bounding approaches for the QMSTP in the literature are closely related. Then, we exploit an extended formulation for the minimum spanning tree problem to derive a sequence of relaxations for the QMSTP with increasing complexity and quality. The resulting relaxations differ from the relaxations in the literature. Namely, our relaxations have a polynomial number of constraints and can be efficiently solved by a cutting plane algorithm. Moreover our bounds outperform most of the bounds from the literature.

Keywords Quadratic minimum spanning tree problem · Linearization problem · Extended formulation · Gilmore–Lawler bound

Mathematics Subject Classification (2010) 90C20 · 90C27 · 90C57

1 Introduction

The quadratic minimum spanning tree problem (QMSTP) is the problem of finding a spanning tree of a connected and undirected graph such that the sum of interaction costs over all pairs of edges in the tree is minimized. The QMSTP was introduced by Assad and Xu [1] who have proven that the problem is strongly $\mathcal{NP}$-hard. The QMSTP remains $\mathcal{NP}$-hard even when the cost matrix is of rank one [27]. The adjacent-only quadratic minimum spanning tree problem (AQMSTP), that is the QMSTP where the interaction costs of all non-adjacent edge pairs are assumed to be zero, is also introduced in [1]. That special version of the QMSTP is also strongly $\mathcal{NP}$-hard. The QMSTP has applications in fields such as telecommunication, irrigation, transportation, and energy and hydraulic networks, see e.g., [1, 4, 5].

The first author thank Prof. Tamás Terlaky for being a mentor of great impact and for his continued support.

✉ Renata Sotirov
r.sotirov@uvt.nl

Zoe Verchére
zoe.verchere@ensta-paris.fr

1 Department of Econometrics and OR, Tilburg University, Tilburg, The Netherlands

2 Doctorant á UMA, ENSTA Paris, Institut Polytechnique de Paris, 91120 Palaiseau, France

The QMSTP can be seen as a generalization of several well known optimization problems, including the quadratic assignment problem [1] and the satisfiability problem [6]. There are also numerous variants of the QMSTP problem such as the minimum spanning tree problem with conflict pairs, the quadratic bottleneck spanning tree problem, and the bottleneck spanning tree problem with conflict pairs. For description of those problems, see e.g., Ćustić et al. [8]. In the same paper the authors investigate the complexity of the QMSTP and its variants, and prove intractability results for the mentioned variants on fan-stars, fans, wheels, (k, n)-accordions, ladders and (k, n)-ladders. The authors also prove that the AQMSTP on (k, n)-ladders and the QMSTP where the cost matrix is a permuted doubly graded matrix are polynomially solvable.

Ćustić and Punnen [7] prove that the QMSTP on a complete graph is linearizable if and only if a symmetric cost matrix is a symmetric weak sum matrix. An instance of the QMSTP is called linearizable if there exists an instance of the minimum spanning tree problem (MSTP) such that the associated costs for both problems are equal for every feasible spanning tree. The authors of [7] also present linearizability results for the QMSTP on complete bipartite graphs, on the class of graphs in which every biconnected component is either a clique, a biclique or a cycle, as well as on biconnected graphs that contain a vertex with degree two. Note that linearizable instances for the QMSTP can be solved in polynomial time.

There is a substantial amount of research on lower bounding approaches and exact algorithms for the QMSTP. Assad and Xu [1] propose a lower bounding procedure that generates a sequence of lower bounds. In each iteration of the algorithm, they solve a minimum spanning tree problem. Öncan and Punnen [21] propose a Lagrangian relaxation scheme based on the linearized QMSTP formulation from [1] that is strengthened by adding two sets of valid inequalities. Cordone and Passeri [6] re-implement the Lagrangian approach from [1] with some minor modifications that result in improved CPU times but slightly weaker lower bounds. Lower bounding approaches in [1, 6, 21] yield Gilmore-Lawler (GL) type bounds for the QMSTP. The GL procedure is a well-known approach to construct lower bounds for quadratic binary optimization problems, see e.g., [12, 17]. Pereira et al. [24] derive lower bounds for the QMSTP using the reformulation linearization technique (RLT) and introduce bounding procedures based on Lagrangian relaxation. The RLT provides a hierarchy of relaxations that are ranging between the continuous and convex hull representation for linear $0-1$ mixed-integer programs [31, 32]. In [33], the RLT is generalized for solving discrete and continuous nonconvex problems. The RLT consists of the following two steps: a reformulation step in which additional nonlinear valid inequalities are generated, and a linearization step in which each product term is replaced by a continuous variable. The level of the hierarchy corresponds to the degree of polynomial terms produced during the reformulation stage. The authors from [24] report solving instances with up to 50 vertices to optimality. Rostami and Malucelli [29] consider the GL procedure, as well as the (incomplete) first and second level RLT bounds for the QMSTP. To compute those bounds, the authors solve Lagrangian relaxations. Relaxations presented in [24, 29] are not complete first and second level RLT relaxations. Namely, those relaxations do not contain a large class of RLT constraints. Guimarães et al. [13] consider semidefinite programming (SDP) lower bounds for the QMSTP and report the best exact solution approach for the problem up to date. The SDP bounds do not belong to the RLT type bounds. Exact approaches for the QMSTP are also considered in [1, 6, 24, 25].

Various heuristic approaches for the QMSTP are tested in the literature. For example, genetic algorithms for the QMSTP are implemented in [4, 11, 37], a tabu search in [6, 18, 23], a swarm intelligence approach in [36], and evolutionary algorithms in [34, 35]. Palubeckis [23] compare simulated annealing, hybrid genetic and iterated tabu search algorithms for the QMSTP. The results show that their tabu search algorithm outperforms the other two approaches in terms of both solution quality and computation time. A local search algorithm for the QMSTP that alternatively performs a neighbourhood search and a random move phase is introduced in [21].

1.1 Main Results and Outline

Most of the lower bounding approaches for the QMSTP are closely related as we show in the next section. Namely, those bounds are based on Lagrangian relaxations obtained from the RLT type of bounds and solved by specialized iterative methods. This connection was not made till now, to the best of our knowledge. Even semidefinite programming lower bounds for the QMSTP involve Lagrangian relaxations.

In this paper we use a different approach for solving the QMSTP. We exploit an extended formulation for the minimum spanning tree problem to compute lower bounds for the QMSTP. Our relaxations have a polynomial number of constraints and we solve them using a cutting plane algorithm.

To derive our relaxations we first consider a linearization based relaxation for the QMSTP. The linearization based relaxation finds the best possible linearizable matrix for the given QMSTP instance, by solving a linear programming (LP) problem. In particular, it searches for the best under-estimator of the quadratic objective function that is in the form of a weak sum matrix. Linearization based bounds are introduced by Hu and Sotirov [14] and further exploited by de Meijer and Sotirov [9]. Next, we consider the continuous relaxation of an exact linear formulation for the QMSTP from [1], and prove that its dual is equivalent to the linearization based relaxation. To derive both relaxations, we exploit the polynomial size extended formulation for the MSTP by Martin [20].

In order to improve the continuous relaxation of the QMSTP formulation from [1] we add facet defining inequalities of the Boolean Quadric Polytope (BQP) [22]. We show that the linearization based relaxation with a particular subset of the BQP inequalities is not dominated by the incomplete first level RLT relaxation from the literature [24, 29], or vice versa. However, after adding all BQP cuts to the linearization based relaxation the resulting bounds outperform even the incomplete second level RLT bounds from the literature on some instances. Since adding all BQP inequalities at once is computationally expensive, we implement a cutting plane approach that considers the most violated constraints.

This paper is organized as follows. In Section 2 we formally introduce the QMSTP and present the extended formulation for the MSTP from [19]. In Section 3 we provide an overview of lower bounds for the QMSTP. In particular, Section 3.1 presents the Gilmore-Lawler type bounds including Assad–Xu and Öncan–Punnen bounds, and Section 3.2 the RLT type bounds. New relaxations are presented in Section 4. Section 4.1 introduces a linearization based relaxation, and Section 4.2 provides several new relaxations of increasing complexity. Numerical results and concluding remarks are in Section 5 and Section 6, respectively.

Notation

Given a subset $S \subseteq V$ of vertices in a graph $G = (V, E)$, we denote the set of edges with both endpoints in S by $E(S) := \{e \in E \mid e = \{i, j\},\ i, j \in S\}$. Furthermore, we use $\delta(S) \subseteq E$ to denote the set of edges with exactly one endpoint in S.

We introduce the operator $\text{Diag} : \mathbb{R}^n \to \mathbb{R}^{n \times n}$ that maps a vector to a diagonal matrix whose diagonal elements correspond to the elements of the input vector. For two matrices

$X = (x_{ij}),\ Y = (y_{ij}) \in \mathbb{R}^{n\times m}$, $X \geq Y$ means $x_{ij} \geq y_{ij}$ for all i, j. We denote by $\mathbf{0}$ the all-zero vector of appropriate size.

2 The Quadratic Minimum Spanning Tree Problem

Let us formally introduce the quadratic minimum spanning tree problem. We are given a connected, undirected graph $G = (V, E)$ with $n = |V|$ vertices and $m = |E|$ edges, and a matrix $Q = (q_{ef}) \in \mathbb{R}^{m\times m}$ of interaction costs between edges of G. For $e = f$, q_{ee} represents the cost of edge e. We assume without loss of generality that $Q = Q^{\mathrm{T}}$. The QMSTP can be formulated as follows:

$$\min \left\{ \sum_{e\in E} \sum_{f\in E} q_{ef} x_e x_f \mid x \in \mathcal{T} \right\}, \tag{1}$$

where $\mathcal{T}$ denotes the set of all spanning trees, and each spanning tree is represented by an incidence vector x of length m, i.e.,

$$\mathcal{T} := \left\{ x \in \{0, 1\}^m \;\middle|\; \sum_{e\in E} x_e = n - 1,\ \sum_{e\in E(S)} x_e \leq |S| - 1,\ S \subset V,\ |S| \geq 2 \right\}.$$

We denote by $\mathcal{T}_{\mathcal{R}}$ the convex hull of the elements from $\mathcal{T}$.

If the matrix Q is a diagonal matrix i.e., $Q = \mathrm{Diag}(p)$ for some $p \in \mathbb{R}^m$, then the QMSTP reduces to the minimum spanning tree problem:

$$\min \left\{ \sum_{e\in E} p_e x_e \mid x \in \mathcal{T} \right\}. \tag{2}$$

It is a well known result by Edmonds [10] that the linear programming relaxation of (2) has an integer polyhedron. However, the spanning tree polytope has an exponential number of constraints. Nevertheless, the MSTP is solvable in polynomial time by e.g., algorithms developed by Prim [26] and Kruskal [16].

There exists a polynomial size extended formulation for the minimum spanning tree problem, due to Martin [20]. He derived the polynomial size formulation for the MSTP by exploiting a known result (see e.g., [19]) that one can test if a vector violates subtour elimination constraints by solving a max flow problem. Martin [20] uses the following valid separation linear program for subtour elimination constraints that contain vertex k:

$$(SP_k(x)) \qquad \begin{array}{lll} \max & \sum_{e\in E} x_e \alpha_e^k - \sum_{i=1, i\neq k}^{n} \theta_i^k & \\ \text{s.t.} & \alpha_e^k - \theta_i^k \leq 0 & \forall e \in \delta(i),\ i \in V, \\ & \theta_i^k \geq 0 & \forall i. \end{array} \tag{3}$$

The above max flow problem formulation is from [28]. Thus, for $\tilde{x} \in \mathbb{R}^m$, $\tilde{x} \geq 0$, the objective value of $SP_k(\tilde{x})$ equals zero if and only if $\sum_{e\in E(S)} \tilde{x}_e \leq |S| - 1$ where $k \in S$, see

[20]. By exploiting the dual problem of (3) for all vertices $k \in V$, Martin [20] derived the following extended formulation for the minimum spanning tree problem:

$$
\begin{aligned}
\min \quad & \sum_{e \in E} p_e x_e && && \text{(4a)}\\
\text{s.t.} \quad & \sum_{e \in E} x_e = n - 1, && && \text{(4b)}\\
& z_{kij} + z_{kji} = x_e, && k = 1, \ldots, n,\ e = \{i, j\} \in E, && \text{(4c)}\\
& \sum_{s>i} z_{kis} + \sum_{h<i} z_{kih} \le 1, && k = 1, \ldots, n,\ i \neq k, && \text{(4d)}\\
& \sum_{s>k} z_{kks} + \sum_{h<k} z_{kkh} \le 0, && k = 1, \ldots, n, && \text{(4e)}\\
& z_{kij} \ge 0, && k, i, j = 1, \ldots, n, && \text{(4f)}\\
& x_e \ge 0 && \forall e \in E. && \text{(4g)}
\end{aligned}
$$

There are $O(n^3)$ constraints and $O(n^3)$ variables in the above LP relaxation. Although it is impractical to use (4) to find the minimal cost spanning tree because there exist several efficient algorithms for solving the MSTP, we show that it is beneficial to use an extended formulation for solving the QMSTP. There are several equivalent formulations of the extended model (4), see e.g., [15].

Let us define

$$
\mathcal{T}_E := \left\{ x \in \mathbb{R}^m,\ z \in \mathbb{R}^{m \times m \times m} \mid \text{(4b)–(4g)} \right\}.
$$

Now, the QMSTP can be formulated as follows:

$$
\min \left\{ \sum_{e \in E} \sum_{f \in E} q_{ef} x_e x_f \mid (x, z) \in \mathcal{T}_E \right\}.
$$

For future reference, we present the dual of (4):

$$
\begin{aligned}
\max \quad & -(n-1)\epsilon - \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} \mu_{ij} && && \text{(5a)}\\
\text{s.t.} \quad & \sum_{k=1}^{n} \theta_{ke} - \epsilon \le p_e, && \forall e \in E, && \text{(5b)}\\
& \mu_{ki} + \sum_{e \in E(\{i,j\})} \theta_{ke} \ge 0, && k, i, j = 1, \ldots, n, && \text{(5c)}\\
& \mu_{ki} \ge 0, && k, i = 1, \ldots, n, && \text{(5d)}\\
& \epsilon \in \mathbb{R},\ \ \theta_{ke} \in \mathbb{R}, && k = 1, \ldots, n,\ e \in E. && \text{(5e)}
\end{aligned}
$$

3 An Overview of Lower Bounds for the QMSTP

In this section we present several known lower bounding approaches for the QMSTP. We classify bounds into the Gilmore–Lawler type bounds and the RLT type bounds. Then, we show how those two types of bounding approaches are closely related.

3.1 The Gilmore–Lawler Type Bounds

We consider here the classical Gilmore–Lawler type bound for the QMSTP. The GL procedure is a well-known approach to construct lower bounds for binary quadratic optimization problems. The bounding procedure was introduced by Gilmore [12] and Lawler [17] to compute a lower bound for the quadratic assignment problem. Nowadays, the GL bounding procedure is extended to many other optimization problems including the quadratic minimum spanning tree problem [29], the quadratic shortest path problem [30], and the quadratic cycle cover problem [9].

The GL procedure for the QMSTP is as follows. For each edge $e \in E$, solve the following optimization problem:

$$z_e := \min \left\{ \sum_{f \in E} q_{ef} x_f \mid x_e = 1,\ x \in \mathcal{T} \right\}. \tag{6}$$

The value z_e is the minimum contribution to the QMSTP objective with e being in the solution. Then, solve the following minimization problem:

$$GL(Q) := \min \left\{ \sum_{e \in E} z_e x_e \mid x \in \mathcal{T} \right\}. \tag{7}$$

The optimal solution of the above optimization problem is the GL type lower bound for the QMSTP. Note that in (6) and (7) one can equivalently solve the corresponding continuous relaxations, thus optimize over $\mathcal{T}_{\mathcal{R}}$ or $\mathcal{T}_E$, or use one of the efficient algorithms for solving the MSTP.

Rostami and Malucelli [29] prove that one can also compute the GL type bound by solving the following LP problem:

$$
\begin{aligned}
GL(Q) := \min \ & \sum_{e, f \in E} q_{ef} y_{ef} && \text{(8a)}\\
\text{s.t.} \ & \sum_{f \in E} y_{ef} = (n-1) x_e, \quad \forall e \in E, && \text{(8b)}\\
& \sum_{f \in E(S)} y_{ef} \le (|S|-1) x_e, \quad \forall e \in E,\ \forall S \subset V,\ S \neq \emptyset, && \text{(8c)}\\
& y_{ee} = x_e, \quad \forall e \in E, && \text{(8d)}\\
& y_{ef} \ge 0, \quad \forall e, f \in E, && \text{(8e)}\\
& x \in \mathcal{T}_{\mathcal{R}}. && \text{(8f)}
\end{aligned}
$$

Note that from (8c) it follows that $y_{ef} \le x_e$ for all $e, f \in E$. Clearly, it is more efficient to compute the GL bound by solving (6)–(7), than solving (8).

The Gilmore–Lawler type bound can be further improved within an iterative algorithm that exploits equivalent reformulations of the problem. The so obtained iterative bounding scheme is known as the generalized Gilmore–Lawler bounding scheme. The generalized GL bounding scheme was implemented for several optimization problems, see e.g., [3, 9, 14, 30]. Hu and Sotirov [14] show that bounds obtained by the GL bounding scheme are dominated by the first level RLT bound. The bounding procedure in the next section can be seen as a special case of the generalized Gilmore–Lawler bounding scheme.

3.1.1 Assad–Xu Leveling Procedure

Assad and Xu [1] propose a method that generates a monotonic sequence of lower bounds for the QMSTP, which results in an improved GL type lower bound. The approach is based on equivalent reformulations of the QMSTP.

Assad and Xu [1] introduce first the following transformation of costs:

$$\begin{aligned} q_{ef}(\gamma) &= q_{ef} + \gamma_f, \qquad && \forall e, f \in E,\ e \neq f, \\ q_{e,e}(\gamma) &= q_{e,e} - (n-2)\gamma_e, && \forall e \in E, \end{aligned} \tag{9}$$

where γ_e $(e \in E)$ is a given parameter, and then apply the GL procedure for a given γ. In particular, Assad and Xu [1] solve the following problem:

$$f_e(\gamma) := \min \left\{ \sum_{f \in E} q_{ef}(\gamma) x_f \mid x_e = 1,\ x \in \mathcal{T} \right\}, \tag{10}$$

for each edge $e \in E$. The value $f_e(\gamma)$ is the minimum contribution to the reformulated QMSTP objective with e being in the solution. Then $f_e(\gamma)$ is used in the following optimization problem:

$$AX(\gamma) := \min \left\{ \sum_{e \in E} f_e(\gamma) x_e \mid x \in \mathcal{T} \right\}. \tag{11}$$

The optimal solution of the above optimization problem is a GL type lower bound for the QMSTP that depends on a parameter γ. In the case that γ is all zero-vector, the corresponding lower bound $AX(\gamma)$ is equal to the GL type lower bound described in the previous section.

The function $AX(\gamma)$ is a piecewise linear, concave function. In order to find γ that provides the best possible bound of type (11), that is to solve $\max_\gamma AX(\gamma)$, the authors propose Algorithm 1.

Algorithm 1 Assad–Xu leveling procedure

1: $\epsilon > 0$, $\gamma^1 = \mathbf{0}$, $i \leftarrow 1$
2: **while** $(\max_e f_e(\gamma^i) - \min_e f_e(\gamma^i) > \epsilon)$ **do**
3: Compute transformed costs using (9).
4: Compute $f_e(\gamma^i)$ and $AX(\gamma^i)$ using (10) and (11), resp.
5: Update $\gamma_e^{i+1} \longleftarrow \gamma_e^i + \frac{f_e(\gamma^i)}{n-1}$ for $e \in E$.
6: $i \leftarrow i + 1$
7: **return** $AX(Q) \leftarrow AX(\gamma^{i-1})$.

Algorithm 1 resembles the generalized Gilmore–Lawler bounding scheme. Since the algorithm starts with $\gamma^1 = \mathbf{0}$, the first computed bound equals the classical GL type bound. Note that Algorithm 1, in Step 3 adds to the off diagonal elements of the cost matrix Q a weak sum matrix and then subtracts its linearization vector on the diagonal. For a relation between linearizable weak sum matrices and corresponding linearization vectors see Section 4.1. Thus Algorithm 1 generates a sequence of equivalent representations of the QMSTP while converging to the optimal γ^*.

3.1.2 Öncan–Punnen Bound

The following exact, linear formulation for the QMSTP is presented in [1]:

$$\min \sum_{e,f\in E} q_{ef} y_{ef} \tag{12a}$$

$$\text{s.t.} \quad \sum_{f\in E} y_{ef} = (n-1)x_e, \quad \forall e \in E, \tag{12b}$$

$$\sum_{e\in E} y_{ef} = (n-1)x_f, \quad \forall f \in E, \tag{12c}$$

$$y_{ee} = x_e, \quad \forall e \in E, \tag{12d}$$

$$0 \le y_{ef} \le 1, \quad \forall e, f \in E, \tag{12e}$$

$$x \in \mathcal{T}. \tag{12f}$$

Öncan and Punnen [21] propose a lower bound for the QMSTP that is derived from (12) and also includes the following valid inequalities:

$$\sum_{e\in\delta(i)} y_{ef} \ge x_f, \quad \forall i \in V, f \in E, \tag{13}$$

$$\sum_{f\in\delta(i)} y_{ef} \ge x_e, \quad \forall i \in V, e \in E, \tag{14}$$

where $\delta(i)$ denotes the set of all incident edges to vertex i. In particular, the authors from [21] propose to solve the Lagrangian relaxation of (12a)–(12f), (13)–(14) obtained by dualizing constraints (13). For a fixed Lagrange mulitiplier λ, the resulting Lagrangian relaxation is solved in a similar fashion as the GL type bound, where costs in objectives of problems (6) and (7) are:

$$q_{ef}(\lambda) := q_{ef} - \sum_{i\in V,\ f\in\delta(i)} \lambda_{i,f}, \quad \forall e, f \in E,\ e \ne f$$

and

$$q_{ee}(\lambda) := q_{ee} + \sum_{i\in V,\ e\in\delta(i)} \lambda_{i,e}, \quad \forall e \in E,$$

respectively. After the solution of the Lagrangian relaxation with a given λ is obtained, a subgradient algorithm is implemented to update the multiplier, and the process iterates. Numerical results show that Öncan–Punnen bounds provide, in general, stronger bounds than Assad–Xu bounds, see also [29]. In [21], the authors also consider replacing the condition $x \in \mathcal{T}$ by the multicommodity flow constraints from [19], but only for graphs up to 20 vertices.

3.2 The RLT Type Bounds

Pereira et al. [24] present a relaxation for the QMSTP that is derived by applying the first level reformulation linearization technique on (1). Nevertheless, their numerical results are obtained by solving the following *incomplete* first level RLT relaxation:

$$\begin{aligned}\widetilde{RTL_1}(Q) := \min & \sum_{e,f\in E} q_{ef} y_{ef} \\ \text{s.t.} \;\; & y_{ef} = y_{fe} \qquad \forall e, f \in E, \\ & \text{(8b)–(8f)}.\end{aligned} \tag{15}$$

We denote by $\widetilde{RLT_1}$ the relaxation (15). To complete the above model to the first level RLT formulation of the QMSTP, one needs to add the following constraints:

$$\sum_{f\in E(S)} (x_f - y_{ef}) \leq (|S|-1)(1-x_e) \qquad \forall e \in E, \; \forall S \subset V, \; |S| \geq 2. \tag{16}$$

It is written in [24] that preliminary computational experiments show that bounds obtained from the (complete) first level RLT relaxation (RLT_1) do not significantly differ from bounds obtained from the incomplete first level RLT relaxation. However, computational effort to solve the full RLT relaxation significantly increases due to the constraints (16). Therefore, Pereira et al. [24] compute only the incomplete first level RLT bound.

The (complete) first level RLT formulation for the QMSTP needs not to include constraints:

$$y_{ef} \leq x_e, \quad y_{fe} \leq x_e, \quad \forall e, f \in E, \tag{17}$$

$$y_{ef} \geq x_e + x_f - 1, \quad \forall e, f \in E, \tag{18}$$

since these are implied by the rest of the constraints. Namely, constraints (17)–(18) readily follow by considering S with two elements in (8c) and (16). However, constraints (18) are not implied by the constraints of the incomplete RLT relaxation (15). Nevertheless, Pereira et al. [24] do not impose those constraints to the incomplete model for the same reason that constraints (16) are not added; that is that constraints (18) do not significantly improve the value of the bound but are expensive to include.

To approximately solve (15), Pereira et al. [24] dualize constraints $y_{ef} = y_{fe}$ $(e, f \in E)$ for a given Lagrange multiplier and then apply the GL procedure. Then, they use a subgradient algorithm to derive a sequence of improved multipliers and compute the corresponding bounds. The so obtained bounds converge to the optimal value of the incomplete first level RLT lower bound. It is clear from the above discussion that the bound from [24] is related to the Gilmore–Lawler type bounds.

Pereira et al. [24] prove

$$P_{\widetilde{RLT_1}} \subseteq P_{OP} \subseteq P_{AX},$$

where P_{AX} denotes the convex hull of the QMSTP formulation (12), P_{OP} denotes the convex hull of (12b)–(12f) and (13)–(14), and $P_{\widetilde{RLT_1}}$ the convex hull of the incomplete first level RLT relaxation (15). We remark that in [24] the authors add to P_{AX} and P_{OP} constraints (8c) that are not in the original description of those polyhedrons, see [21]. Nevertheless, their result as well as the statement above are correct.

Remark 1 It is interesting to note that the continuous relaxation (8) and the incomplete first level RLT relaxation differ only in the symmetry constraints $y_{ef} = y_{fe}$ $(\forall e, f)$ that are included in (15). In other words, the dual of the relaxation $\widetilde{RLT_1}$ contains also dual variables, say δ_{ef} and δ_{fe} $(\forall e, f)$, that correspond to the symmetry constraints. Thus, for each e, f there is a constraint in the dual of the incomplete first level RLT relaxation that has a term $q_{ef} + (\delta_{ef} - \delta_{fe})$. Since $\delta_{ef} - \delta_{fe} = -(\delta_{fe} - \delta_{ef})$, this term corresponds to adding a skew

Table 1 RLT type bounds

n	d	UB	GL	$\widetilde{RLT1}$	RLT_1	$\widetilde{RLT_2}$
10	33	350	299	**350.0**	**350.0**	344.1
10	67	255	149	202.2	204.1	**226.1**
15	33	745	445	578.2	603.3	**637.8**
15	67	659	283	385.4	385.7	**488.9**
20	33	1379	690	888.0	891.7	**1056.7**

symmetric matrix to the cost matrix Q. This implies that the optimal solution of the dual provides the best skew symmetric matrix that is added to the quadratic cost to improve the GL type bound. Thus one can obtain the optimal solution for the incomplete first level RLT relaxation for the QMSTP in *one step* of the GL bounding scheme.

Furthermore, in [24] it is proposed a generalization of the RLT relaxation that is based on decomposing spanning trees into forests of a fixed size. The larger the size of the forest, the stronger the formulation. The resulting relaxations are solved by using a Lagrangian relaxation scheme. In [29] it is implemented a dual-ascent procedure to approximate the value of the (incomplete) level two RLT relaxation ($\widetilde{RLT_2}$).

Since we couldn't find results on the first level RLT relaxation in the literature, we computed them for several instances. In Table 1 we compare GL, $\widetilde{RLT_1}$, RLT_1 and $\widetilde{RLT_2}$ bounds for small CP1 instances. For the description of instances see Section 5. UB stands for upper bounds. Bounds for the (incomplete) first and second level RLT relaxation are from [29]. Table 1 shows that the GL type bounds are significantly weaker than the other bounds. The best lower bounds in that table are marked in bold. The results also show that difference between $\widetilde{RLT_1}$ and RLT_1 bounds is not always small. $\widetilde{RLT_2}$ bounds dominate all presented bounds in all but one case. This discrepancy should be due to the way that those bounds are computed, i.e., by implementing a dual ascent algorithm on the incomplete second level RLT relaxation. For more such examples see [29].

4 New Lower Bounds for the QMSTP

4.1 Linearization-based Lower Bounds

Ćustić and Punnen [7] prove that the QMSTP on a complete graph is linearizable if and only if its cost matrix is a symmetric weak sum matrix. In this section, we exploit weak sum matrices to derive lower bounds for the QMSTP. Hu and Sotirov [14] introduce the concept of linearization based bounds, and show that many of the known bounding approaches including the GL type bounds are also linearization based bounds. De Meijer and Sotirov [9] derive strong and efficient linearization based bounds for the quadratic cycle cover problem.

A matrix $Q \in \mathbb{R}^{m\times m}$ is called a sum matrix if there exist $a, b \in \mathbb{R}^m$ such that $q_{ef} = a_e + b_f$ for all $e, f \in \{1, \ldots, m\}$. A weak sum matrix is a matrix for which this property holds except for the entries on the diagonal, i.e., $q_{ef} = a_e + b_f$ for all $e \neq f$. However for a symmetric weak sum matrix we have that vectors a and b are equal, i.e., $q_{ef} = a_e + a_f$ for all $e \neq f$.

It is proven in [7, Theorem 5] that a symmetric cost matrix Q of the QMSTP on a complete graph is linearizable if and only if it is a symmetric weak sum matrix. In particular, the authors show that for a symmetric weak sum matrix of the form $q_{ef} = a_e + a_f$ and a tree T in a complete graph one has:

$$\begin{aligned}\sum_{e\in T}\sum_{f\in T} q_{ef} &= \sum_{e\in T}\sum_{\substack{f\in T\\ f\neq e}}(a_e + a_f) + \sum_{e\in T} q_{ee}\\ &= 2(n-2)\sum_{e\in T} a_e + \sum_{e\in T} q_{ee}\\ &= \sum_{e\in T} p_e,\end{aligned}$$

where $p \in \mathbb{R}^m$ with

$$p_e = 2(n-2)a_e + q_{ee} \tag{19}$$

is a linearization vector. Thus, solving the QMSTP in which the cost matrix is a symmetric weak sum matrix corresponds to solving a minimum spanning tree problem.

In [9, 14] it is proven that one can obtain a lower bound for the optimal value $OPT(Q)$ of a minimization quadratic optimization problem with the cost matrix Q from a linearization matrix $\hat{Q}$, which satisfies $Q \geq \hat{Q}$. In particular, we have:

$$OPT(Q) = \min_{x\in X}\{x^{\mathrm{T}}Qx\} \geq \min_{x\in X}\{x^{\mathrm{T}}\hat{Q}x\} = \min_{x\in X}\{x^{\mathrm{T}}\hat{p}\} = OPT(\hat{p}),$$

where $\hat{p}$ is a linearization vector of $\hat{Q}$, X is the feasible set of the optimization problem, and $OPT(\hat{p})$ is the optimal solution of the corresponding linear problem.

Thus, for a weak sum matrix $\hat{Q}$ such that $Q \geq \hat{Q}$ and the corresponding linearization vector $\hat{p}$, the optimal solution for the MSTP (2) with the cost vector $\hat{p}$ is a lower bound for the QMSTP (1). By maximizing the right hand side in the above inequality over all linearization vectors of the form (19), one obtains the strongest linearization based bound for the QMSTP. Therefore, the strongest linearization based lower bound for the QMST is the optimal solution of the following problem:

$$LBB(Q) := \max -(n-1)\epsilon - \sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n} \mu_{ij} \tag{20a}$$

$$\text{s.t.}\quad a_e + a_f \leq q_{ef} \qquad \forall e, f \in E,\ e \neq f, \tag{20b}$$

$$\sum_{k=1}^{n} \theta_{ke} - \epsilon \leq 2(n-2)a_e + q_{ee} \qquad \forall e \in E, \tag{20c}$$

$$\mu_{ki} + \sum_{e\in E(\{i,j\})} \theta_{ke} \geq 0, \qquad k, i, j = 1, \ldots, n, \tag{20d}$$

$$\mu_{ki} \geq 0, \qquad k, i = 1, \ldots, n, \tag{20e}$$

$$\epsilon \in \mathbb{R},\quad \theta_{ke} \in \mathbb{R},\quad a_e \in \mathbb{R}, \qquad k = 1, \ldots, n,\ e \in E. \tag{20f}$$

We denote the linear programming relaxation (20) by LBB. The LBB is derived from the dual of the extended formulation of the MSTP, see (5). Note that in computations one can exploit the fact that $Q = Q^{\mathrm{T}}$ to reduce the number of constraints in (20b). In the next section we relate relaxation (20) with a relaxation obtained from the linear formulation of the QMSTP.

4.2 Strong and Efficient Lower Bounds

The QMSTP formulation (12) is introduced in [1]. We exploit that QMSTP formulation to derive a sequence of linear relaxations for the QMSTP. First, by replacing $\mathcal{T}$ by $\mathcal{T}_E$, and adding the symmetry constraints $y_{ef} = y_{fe}$ $(\forall e, f)$, we obtain the following relaxation for the QMSTP:

$$
\begin{aligned}
VS_0(Q) := \min \sum_{e,f \in E} q_{ef} y_{ef} \quad & \quad (21a)\\
\text{s.t.} \sum_{f \in E} y_{ef} = (n-1)x_e \quad \forall e \in E, & \quad (21b)\\
y_{ee} = x_e \quad \forall e \in E, & \quad (21c)\\
y_{ef} = y_{fe} \quad \forall e, f \in E, & \quad (21d)\\
(x, z) \in \mathcal{T}_E, & \quad (21e)\\
0 \le y_{ef} \le 1 \quad \forall e, f \in E. & \quad (21f)
\end{aligned}
$$

We denote by VS_0 the above relaxation. In the above optimization problem, we omit constraints:

$$\sum_{e \in E} y_{ef} = (n-1)x_f \qquad \forall f \in E,$$

since they are implied by constraints (21b) and (21d). Next, we compare the relaxation (21) and the strongest linearization based relaxation (20). Let us first present the dual of the relaxation (21):

$$
\begin{aligned}
& \max -(n-1)\epsilon - \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \ne i}}^{n} \mu_{ij} && && (22a)\\
& \text{s.t.} \quad \alpha_e + \gamma_e \ge 0 && \forall e \in E, && (22b)\\
& -\alpha_e - \delta_{ef} + \delta_{fe} \le q_{ef} && \forall e, f \in E,\ e \ne f, && (22c)\\
& \sum_{k=1}^{n} \theta_{ke} + (n-1)\alpha_e + \gamma_e - \epsilon \le q_{ee} && \forall e \in E, && (22d)\\
& \mu_{ki} + \sum_{e \in S(\{i,j\})} \theta_{ke} \ge 0, && k, i, j = 1, \ldots, n, && (22e)\\
& \mu_{ki} \ge 0, && k, i = 1, \ldots, n, && (22f)\\
& \epsilon \in \mathbb{R},\ \ \theta_{ke}, \alpha_e, \gamma_e, \delta_e \in \mathbb{R}, && k = 1, \ldots, n,\ e \in E. && (22g)
\end{aligned}
$$

To derive the above dual problem we remove upper bounds on x_e and y_{ef} in the corresponding primal problem, since it is never optimal to set the values of these variables larger than one. The following result shows that the strongest linearization based bound equals the bound obtained by solving (21), or equivalently (22).

Theorem 4.1 *Let G be a complete graph. Then, optimization problems* (20) *and* (22) *are equivalent.*

Proof We show that for every feasible solution for (22), we can find a feasible solution for (20) with the same objective value, and conversely for every feasible solution for (20) we can find a feasible solution for (22) with the same objective value.

Let $(\alpha, \gamma, \delta, \epsilon, \mu, \theta)$ be a feasible solution for the optimization problem (22). Let us find a feasible solution $(\hat{a}, \hat{\epsilon}, \hat{\mu}, \hat{\theta})$ for (20). We define $\hat{\epsilon} := \epsilon$, $\hat{\mu} := \mu$ and $\hat{\theta} := \theta$. Thus, the objective values (22a) and (20a) are equal, and constraints (20d) (resp., (20e)) correspond to constraints (22e) (resp., (22f)).

Let us define $\hat{a}_e := -\frac{1}{2(n-2)}((n-1)\alpha_e + \gamma_e)$ for $e \in E$ and note that

$$\hat{a}_e = -\frac{(n-1)\alpha_e}{2(n-2)} - \frac{\gamma_e}{2(n-2)} \leq -\frac{(n-1)\alpha_e}{2(n-2)} + \frac{\alpha_e}{2(n-2)} = -\frac{\alpha_e}{2},$$

where $\alpha_e \geq -\gamma_e$ follows from (22b). Further from (22c) we have that

$$-\alpha_e - (\delta_{ef} - \delta_{fe}) \leq q_{ef},$$
$$-\alpha_f - (\delta_{fe} - \delta_{ef}) \leq q_{fe},$$

for every pair of edges (e, f) such that $e \neq f$. By adding these two inequalities, we obtain $-(\alpha_e + \alpha_f) \leq q_{ef} + q_{fe}$. However, the inequality $\hat{a}_e \leq -\frac{\alpha_e}{2}$ implies $\hat{a}_e + \hat{a}_f \leq -\frac{1}{2}(\alpha_e + \alpha_f) \leq \frac{1}{2}(q_{ef} + q_{fe}) = q_{ef}$ for every pair (e, f) such that $e \neq f$. Note that we assumed that $Q = Q^{\mathrm{T}}$. Thus constraints (20b) are also satisfied. To show that constraints (20c) are satisfied, we use (22d) from where it follows:

$$\sum_{k=1}^{n} \theta_{ke} - \epsilon \leq q_{ee} - (n-1)\alpha_e - \gamma_e = q_{ee} + 2(n-2)\hat{a}_e \qquad \forall e \in E.$$

Conversely, let $(\hat{a}, \hat{\epsilon}, \hat{\mu}, \hat{\theta})$ be a feasible solution for (20). Below, we construct a feasible solution $(\alpha, \gamma, \delta, \epsilon, \mu, \theta)$ for (22). We define $\epsilon := \hat{\epsilon}$, $\mu := \hat{\mu}$ by $\theta := \hat{\theta}$. Thus, we have that the objective values (22a) and (20a) are equal, and constraints (20d) (resp., (20e)) correspond to constraints (22e) (resp., (22f)). We define $\alpha_e := -2\hat{a}_e$ and $\gamma_e := 2\hat{a}_e$ for all $e \in E$, from where it follows $\alpha_e + \gamma_e = 0$. Thus, constraint (22b) is satisfied.

From the definitions of α and γ we have $-(n-1)\alpha_e - \gamma_e = 2(n-2)\hat{a}_e$, and from (20c) it follows

$$\sum_{k=1}^{n} \theta_{ke} - \epsilon \leq q_{ee} + 2(n-2)\hat{a}_e = q_{ee} - (n-1)\alpha_e - \gamma_e, \qquad e \in E.$$

Thus, for each $e \in E$ constraint (22d) is satisfied.

It remains to verify constraints (22c). For that purpose we consider constraints (20b) for a pair of edges (e, f) such that $e \neq f$. After multiplying the constraint with two, and using $q_{ef} = q_{fe}$ and $\alpha_e = -2\hat{a}_e$ we obtain

$$-\alpha_e - q_{ef} \leq q_{fe} + \alpha_f.$$

Thus, there exist δ_{ef} and δ_{fe} such that

$$-\alpha_e - q_{ef} \leq (\delta_{ef} - \delta_{fe}),$$
$$(\delta_{ef} - \delta_{fe}) \leq q_{fe} + \alpha_f.$$

This finishes the proof. □

Theorem 4.1 shows that the linearization based relaxation for the QMSTP is equivalent to the relaxation of an exact linear formulation for the QMSTP. Our preliminary numerical results show that the bound $VS_0(Q)$ is not dominated by $GL(Q)$, or vice versa.

Table 2 VS relaxations

name	constraints	complexity
VS_0	(21b)–(21f)	$\mathcal{O}(n^3+m^2)$
VS_1	(21b)–(21e), (23a)–(23b)	$\mathcal{O}(n^3+m^2)$
VS_2	(21b)–(21e), (23a)–(23d)	$\mathcal{O}(n^3+m^3)$

To strengthen the relaxation VS_0, see (21), one can add the following facet defining inequalities of the Boolean Quadric Polytope, see e.g., [22]:

$$0 \le y_{ef} \le x_e, \tag{23a}$$
$$x_e + x_f \le 1 + y_{ef}, \tag{23b}$$
$$y_{eg} + y_{fg} \le x_g + y_{ef}, \tag{23c}$$
$$x_e + x_f + x_g \le y_{ef} + y_{eg} + y_{fg} + 1, \tag{23d}$$

where $e, f, g \in E$, $e \neq f \neq g$. Table 2 introduces relaxations with increasing complexity that are obtained by adding subsets of the BQP inequalities to the linear program (21).

To solve the relaxations VS_1 and VS_2 we use a cutting plane scheme that iteratively adds the most violated inequalities, see section on numerical results for more details.

The following result relates $\widetilde{RLT}_1$ and VS_1 relaxations.

Proposition 4.2 *The relaxation VS_1 is not dominated by the incomplete first level RLT relaxation* (15)*, or vice versa.*

Proof We consider a feasible solution (x, Y) of (15) for an instance on complete graph with 6 vertices. Let $x := (0.75, 0.75, 0, 0, 0, 0, 0, 0, 0.75, 1, 0, 0.75, 1, 0, 0)^{\mathrm{T}}$ and define the 15×15 symmetric matrix Y whose diagonal elements correspond to the elements of vector x. Furthermore, elements on the positions $y_{ef} = y_{fe}$ where

e	1	2	1	2	9	1	2	10	1	2	9	12
f	9	9	10	10	10	12	12	12	13	13	13	13

equal 0.75, while $y_{10,13} = 1$, and all other elements are zero. By direct verification it follows that (x, Y) is feasible for (15). On the other hand, we have that $1 + y_{12} - y_{11} - y_{22} < 0$. Thus, (23b) are violated and (x, Y) is not feasible for (VS_1). The (incomplete) RLT bound for this particular instance is 386.5, while $VS_1(Q) = 372.4$

Conversely, in Section 5 we provide examples where the optimal values of VS_1 are strictly greater than the optimal values of $\widetilde{RLT}_1$. □

The results of the previous proposition are not surprising since constraints (23b) are not included in the relaxation $\widetilde{RLT}_1$. It is not difficult to show the following results.

Corollary 4.3 *The relaxation VS_1 is dominated by the first level RLT relaxation for the QMSTP.*

Corollary 4.4 *The relaxation VS_2 is not dominated by the first level RLT relaxation for the QMSTP, or vice versa.*

Note that Corollary 4.2 follows from the discussion in Section 3.2, and Corollary 4.4 from the fact that constraints (23d) are not present in the RLT_1 relaxation.

5 Numerical Results

In this section, we compare several lower bounds from the literature with the bounds introduced in Section 4. Numerical experiments are performed on an Intel(R) Core(TM) i7-9700 CPU, 3.00 GHz with 32 GB memory. We implement our bounds in the Julia Programming Language [2] and use CPLEX 12.7.1.

To solve VS_2 relaxation we implement a cutting plane algorithm that starts from VS_1 relaxation and iteratively adds the most violated $n \cdot m$ cuts. The algorithm stops if no more violated cuts or after two hours.

To compute upper bounds for the here introduced benchmark instances SV we implement the tabu search algorithm and variable neighbourhood search algorithm from [6]. In the mentioned paper the authors suggest restarting the tabu search algorithm for better performance of the algorithm. We notice that a small number of restarts (i.e., at most 5) and then running the neighbourhood search algorithm can be beneficial for large instances. The total number of iterations of the tabu search algorithm is 5000.

5.1 Test Instances

We test our bounds on the following benchmark sets.

The benchmark set CP is introduced by Cordone and Passeri [6] and consists of 108 instances. These instances consist of graphs with $n \in \{10, 15, \ldots, 50\}$ vertices and densities $d \in \{33\%, 67\%, 100\%\}$. There are four types of random instances denoted by CP1, CP2, CP3, CP4. In CP1 instances, the linear and the quadratic costs are uniformly distributed at random in $\{1, \ldots, 10\}$. In CP2 instances, the linear (resp., quadratic) costs are uniformly distributed at random in $\{1, \ldots, 10\}$ (resp., $\{1, \ldots, 100\}$). In CP3 instances, the linear (resp., quadratic) costs are uniformly distributed at random in $\{1, \ldots, 100\}$ (resp., $\{1, \ldots, 10\}$). In CP4 instances, the linear and the quadratic costs are uniformly distributed at random in $\{1, \ldots, 100\}$.

The benchmark set OP is introduced by Öncan and Punnen [21] and consists of 480 instances. These instances consist of complete graphs with $n \in \{6, 7, \ldots, 17, 18, 30, 30, 40, 50\}$ vertices and are divided into three types. In particular, in OPsym the linear (resp., quadratic) costs are uniformly distributed at random in $\{1, \ldots, 100\}$ (resp., $\{1, \ldots, 20\}$). In OPvsym the linear costs are uniformly distributed at random in $\{1, \ldots, 10000\}$, while the quadratic costs are obtained associating to the vertices $i \in V$ random values $w(i)$ uniformly distributed in $\{1, \ldots, 10\}$ and setting $q_{\{i,j\},\{k,l\}} = w(i)w(j)w(k)w(l)$. In OPesym the vertices are spread uniformly at random in a square of length side 100, the linear costs are the Euclidean distances between the end vertices of each edge, while the quadratic costs are the Euclidean distances between the midpoints of the edges.

We introduce the benchmark set SV[1]. Our benchmark consists of 24 instances. For given a size $n \in \{10, 12, 14, 16, 18, 20, 25, 30\}$, density $d \in \{33\%, 67\%, 100\%\}$, a maximum cost for the diagonal entries, and a maximum cost for the off-diagonal entries, we generate an instance in the following way. 10% of the rows are randomly chosen. These rows will have high costs with each other (between 90% and 100% of the maximum off-diagonal cost), and low costs with rest (between 20% and 40% of the maximum off-diagonal cost). The rows that are not selected have an interaction cost of between 50% and 70% of the maximum off-diagonal cost. Finally, the impact of diagonal entries is greatly minimized, between 0 and

[1] Interested reader can download SV instances from the following link:
https://drive.google.com/drive/folders/1bpF8AfAn2K5QGSoB6Yl76znw6PwdvItb

20% of the maximum diagonal cost. The cost matrices obtained this way are not symmetric, but they can be made so at the user's convenience.

5.2 Computational Results

We first present results for our benchmark set SV. Table 3 reads as follows. In the first two columns we list the number of vertices and density of a graph, respectively. In the third column we provide upper bounds computed as mentioned earlier. In the following three columns we list the incomplete first level RLT bound $\widetilde{RLT}_1$ see (15), VS_1 bound that is (21b)–(21e), (23a)–(23b), and VS_2 bound that is (21b)–(21e), (23a)–(23d), see also Table 2. In columns 7–9 we present gaps by using the formula $100 \cdot (UB - LB)/UB$, where LB stands for the value of the lower bound. Note that the gap we present in our numerical results differs from the gap used in other QMSTP papers, where the authors use $100 \cdot (UB - LB)/LB$. In the last two columns of Table 3 we presents time in seconds required to solve our relaxations. We do not report time for computing $\widetilde{RLT}_1$ as we implement (15) directly, while the other authors that compute RTL type bounds use a more efficient way to compute those bounds.

Table 3 SV instances: bounds and gaps

instance			lower bounds			Gap (%)			time (s)	
n	$d(\%)$	UB	$\widetilde{RLT}_1$	VS_1	VS_2	$\widetilde{RLT}_1$	VS_1	VS_2	VS_1	VS_2
10	33	4217	**4217**	**4217**	**4217**	**0**	**0**	**0**	< 0.05	< 0.05
10	67	3981	3752.6	3876.8	**3981**	5.8	2.6	**0**	< 0.05	0.3
10	100	3930	3499.1	3604.3	**3857**	11.0	8.2	**1.9**	0.1	1.9
12	33	6141	5859.4	6058.1	**6136**	4.6	1.3	**0.1**	0.0	0.2
12	67	6050	5759.6	5910.4	**6038**	4.8	2.3	**0.2**	0.1	3.1
12	100	6051	5726	5868.1	**5991.8**	5.4	3.0	**1.0**	0.3	5.0
14	33	8736	8495.2	8619.1	**8709.1**	2.8	1.3	**0.3**	0.1	0.4
14	67	8606	8096.8	8306.0	**8542.8**	5.9	3.5	**0.7**	0.2	4.8
14	100	8513	7556	7731.7	**8267.4**	11.2	9.2	**2.9**	2.4	61.1
16	33	11735	11231.5	11465	**11711.5**	4.3	2.3	**0.2**	0.1	1.0
16	67	11610	10559.8	10816.2	**11335.3**	9.0	6.8	**2.4**	1.7	50.4
16	100	11516	10089.6	10302.7	**11003.5**	12.4	10.5	**4.5**	8.4	241.1
18	33	15125	13995.8	14382.9	**14916.5**	7.5	4.9	**1.4**	0.3	3.6
18	67	15019	12976.9	13210.6	**14204.6**	13.6	12.0	**5.4**	3.2	200.1
18	100	14943	13217.6	13501.8	**14336.0**	11.5	9.6	**4.1**	2.6	89.6
20	33	19057	17777.2	18178.9	**18778.2**	6.7	4.6	**1.5**	0.3	7.1
20	67	18830	16325.5	16433.5	**17130.4**	13.3	12.7	**9.2**	4.8	163.2
20	100	18812	16026	16299.6	**17528.9**	14.8	13.4	**6.8**	4.1	185.15
25	33	30747	28436.4	29102.6	**30084.4**	7.5	5.3	**2.2**	4.8	179.1
25	67	30554	27061.5	27546.6	**29186.1**	11.4	9.8	**4.5**	2.8	377.9
25	100	30405	24455.6	24257.7	**26251.7**	19.7	20.3	**13.7**	10.4	1260.7
30	33	45184	40946.1	41995.1	**43889.3**	7.1	9.4	**2.7**	34.3	2268.6
30	67	44989	37676.7	38089.8	**41162.8**	16.25	15.3	**8.5**	13.5	1359.9
30	100	44847	35116.9	35037.5	**37489.8**	21.7	21.9	**16.4**	41.8	2700.0

The results in Table 3 show that VS_1 bounds are stronger than $\widetilde{RLT}_1$ bounds for all instances except for the SV instance with $n = 25$ and $d = 100$, and the SV instance with $n = 30$ and $d = 100$. In all tables the best bounds are marked in bold. The difference between gaps for $\widetilde{RLT}_1$ and VS_1 bounds for both instances is less than 1%. Moreover VS_2 relaxation provides a better bound than $\widetilde{RLT}_1$ for the SV instance with $n = 25$ and $d = 100$ within 2 minutes of the cutting plane algorithm. Similarly, VS_2 relaxation provides a better bound than $\widetilde{RLT}_1$ relaxation for the SV instance $n = 30$ and $d = 100$ after a few iterations of the cutting plane algorithm. Recall that the VS_1 relaxation is not dominated by the incomplete first level RLT relaxation, or vice versa in general, see Proposition 4.2. Therefore our benchmark set SV can be used to test the quality of QMSTP bounds that are not RTL type bounds. Table 3 shows that computational times for solving the relaxation VS_1 are very small for all instances. The computational effort for solving the relaxation VS_2 is small for instances with $n \leq 25$ and $d \in \{33\%, 67\%\}$. The results also show that it is computationally more challenging to solve VS_2 for dense instances. Note also that we can stop our cutting plane algorithm at any time and obtain a valid lower bound.

Tables 4, 5, 6, 7, 8 and 9 read similarly to Table 3. We do not report results for OPvsym instances since gaps for VS_1 bounds for those instances are less than or equal to 0.2%. Also, we do not present results for instances with $n > 35$ since the corresponding gaps are (too) large for all bounds in the literature, including ours. Clearly, it is a big challenge to obtain good bounds for the QMSTP when $n \geq 20$ for most of the instances and approaches.

Running times required to solve VS_1 relaxation for most of the test instances in Tables 4–9 are similar to times given in Table 3. That is, for instances CP1, CP2, CP3 and CP4 with $n \leq 20$ and $d \in \{33\%, 67\%, 100\%\}$ as well as instances with $n = 25$ and $d \in \{33\%, 67\%\}$ computation times required to solve VS_1 relaxation are a few seconds. Running times for

Table 4 CP1 instances: bounds and gaps

instance			lower bounds					Gap (%)				
n	$d(\%)$	UB	GL	$\widetilde{RLT}_1$	$\widetilde{RLT}_2$	VS_1	VS_2	GL	$\widetilde{RLT}_1$	$\widetilde{RLT}_2$	VS_1	VS_2
10	33	350	299	**350**	344,1	**350**	**350**	14.6	**0**	1.7	**0**	**0**
10	67	255	149	202.2	226.1	166.4	**248.8**	41.6	20.7	11.3	34.7	**2.4**
10	100	239	120	159.7	199.0	140.7	**201.3**	49.8	33.2	16.7	41.1	**15.8**
15	33	745	445	578.2	637.8	487	**709.2**	40.3	22.4	14.4	34.6	**4.8**
15	67	659	283	385.4	**488.9**	324.4	478.9	57.1	41.5	**25.8**	50.8	27.3
15	100	620	246	320.9	**442.9**	281.5	386.1	60.3	48.2	**28.6**	54.6	37.7
20	33	1379	690	888	1056.7	740.9	**1165.4**	50.0	35.6	23.4	46.3	**15.5**
20	67	1252	454	603.1	**843.1**	523.3	742.7	63.7	51.8	**32.7**	58.2	40.7
20	100	1174	398	506.9	**737.9**	436.7	587.7	66.1	56.8	**37.1**	62.8	49.9
25	33	2185	985	1285.1	1594.9	1045.3	**1615.2**	54.9	41.2	27.0	52.2	**26.1**
25	67	2023	660	834.9	**1239.6**	717	1003.7	67.4	58.7	**38.7**	64.6	50.4
25	100	1943	596	719.1	**1091.0**	645.6	823.4	69.3	63.0	**43.9**	66.8	57.6
30	33	3205	1260	1617.9	**2149.6**	1352.6	2062.12	60.7	49.5	**32.9**	57.8	35.7
30	67	2998	916	1118.2	**1681.4**	956.6	1309.6	69.4	62.7	**43.9**	68.1	56.3
30	100	2874	854	986.3	**1495.3**	906.9	1069.5	70.3	65.7	**48.0**	68.4	62.8
35	33	4474	1597	2014.4	**2812.1**	1690.5	2529.2	64.3	55.0	**37.1**	62.2	43.5
35	67	4147	1215	1437.3	**2208.2**	1262.7	1653.2	70.7	65.3	**46.8**	69.6	60.1
35	100	4000	1156	1303.8	**1953.1**	1222.1	1286.4	71.1	67.4	**51.2**	69.4	67.8

Table 5 CP2 instances: bounds and gaps

instance			lower bounds					Gap (%)				
n	$d(\%)$	UB	GL	$\widetilde{RLT_1}$	$\widetilde{RLT_2}$	VS_1	VS_2	GL	$\widetilde{RLT_1}$	$\widetilde{RLT_2}$	VS_1	VS_2
10	33	3122	2562	**3122**	3045.9	3114.6	**3122**	17.9	**0**	2.4	0.2	**0**
10	67	2042	809	1399.6	1710.2	1014.9	**1946.2**	60.4	31.5	16.2	50.3	**4.7**
10	100	1815	553	937	1384.4	733	**1388.4**	69.5	48.4	23.7	59.6	**23.5**
15	33	6539	3272	4684.1	5329.3	3690.9	**6133.7**	50.0	28.4	18.5	43.6	**6.2**
15	67	5573	1555	2589.7	**3760.5**	1908.1	3637	72.1	53.5	**32.5**	65.8	34.7
15	100	5184	1070	1829.9	**3236.0**	1406	2581.3	79.4	64.7	**37.6**	72.9	50.2
20	33	12425	4801	7035.7	8849.7	5390.8	**10106.7**	61.4	43.4	28.8	56.6	**18.7**
20	67	10893	2352	3816.7	**6573.9**	2907.5	5415.4	78.4	65.0	**39.6**	73.3	50.3
20	100	10215	1676	2748.2	**5442.2**	2060.2	3720.7	83.6	73.1	**46.7**	79.8	63.6
25	33	19976	6642	10068.5	13460.9	7361.8	**13722.1**	66.8	49.6	32.6	63.1	**31.3**
25	67	18251	3196	5054.3	**9666.8**	3795.7	7015.5	82.5	72.3	**47.0**	79.2	61.6
25	100	17411	2123	3469	**7921.3**	2623.2	4759.9	87.8	80.1	**54.5**	84.9	72.7
30	33	29731	7953	12046.6	**17942.7**	9120.5	17066.6	73.3	59.5	**39.6**	69.3	42.6
30	67	27581	3991	6382	**12840.3**	4684.5	8621.6	85.5	76.9	**53.4**	83.0	68.7
30	100	26146	2712	4332.4	**10624.1**	3271.8	5880	89.6	83.4	**59.4**	87.5	77.5
35	33	42305	9995	14684.1	**23568.2**	10771	20378.3	76.4	65.3	**44.3**	74.5	51.8
35	67	38490	4778	7409	**16274.8**	5517.8	10165.3	87.6	80.8	**57.7**	85.7	73.6
35	100	36723	3388	5241.7	**13687.3**	4016.8	6782.8	90.8	85.7	**62.7**	89.1	81.5

Table 6 CP3 instances: bounds and gaps

instance			lower bounds					Gap (%)				
n	$d(\%)$	UB	GL	$\widetilde{RLT_1}$	$\widetilde{RLT_2}$	VS_1	VS_2	GL	$\widetilde{RLT_1}$	$\widetilde{RLT_2}$	VS_1	VS_2
10	33	646	589	**646**	**646**	**646**	**646**	8.8	**0**	**0**	**0**	**0**
10	67	488	320	**488**	476.1	474.9	**488**	34.4	**0**	2.4	2.7	**0**
10	100	426	234	386.2	401.1	360.4	**426**	45.1	9.3	5.8	15.4	**0**
15	33	1236	845	1180.5	1176.0	1113.5	**1218**	31.6	4.5	4.9	9.9	**1.5**
15	67	966	508	848.1	886.2	771.7	**965.3**	47.4	12.2	8.3	20.1	**0.1**
15	100	975	421	780	849.3	724.9	**910**	56.8	20.0	12.9	25.7	**6.7**
20	33	1972	1131	1672.6	1759.1	1511.9	**1911.6**	42.6	15.2	10.8	23.3	**3.1**
20	67	1792	743	1307.1	1470.1	1190	**1541**	58.5	27.1	18.0	33.6	**14.0**
20	100	1544	532	1056.1	1220.6	949.7	**1257.5**	65.5	31.6	20.9	38.5	**18.6**
25	33	2976	1448	2289.2	2488.3	2071.7	**2730.3**	51.3	23.1	16.4	30.4	**8.3**
25	67	2546	888	1630	1917.2	1468.8	**1932.1**	65.1	36.0	24.7	42.3	**24.1**
25	100	2471	761	1409.6	**1740.1**	1284.1	1657.5	69.2	43.0	**29.6**	48.0	32.9
30	33	4070	1834	2856.1	3235.3	2574.2	**3383.1**	54.9	29.8	20.5	36.8	**16.9**
30	67	3649	1152	2053.5	**2516.6**	1854.9	2425.4	68.4	43.7	**31.0**	49.2	33.5
30	100	3483	986	1776.1	**2257.3**	1627.5	2048.2	71.7	49.0	**35.2**	53.3	41.2
35	33	5423	2060	3360	3946.9	3007.1	**4016**	62.0	38.0	27.2	44.5	**25.9**
35	67	4981	1430	2515.7	**3195.0**	2324.1	2938.9	71.3	49.5	**35.9**	53.3	41.0
35	100	4770	1288	2190.1	**2866.6**	1995.7	2504.1	73.0	54.1	**39.9**	58.2	47.5

Table 7 CP4 instances: bounds and gaps

instance			lower bounds					Gap (%)				
n	$d(\%)$	UB	GL	$\widetilde{RLT}_1$	$\widetilde{RLT}_2$	VS_1	VS_2	GL	$\widetilde{RLT}_1$	$\widetilde{RLT}_2$	VS_1	VS_2
10	33	3486	2891	3486	3424.4	**3486**	**3486**	17.1	**0**	1.8	**0**	**0**
10	67	2404	1158	1794	2076.0	1408.8	**2330.3**	51.8	25.4	13.6	41.4	**3.1**
10	100	2197	823	1321.1	1743.7	1120.2	**1794.7**	62.5	39.9	20.6	49.0	**18.3**
15	33	7245	3859	5354.8	6017.4	4371.2	**6819.9**	46.7	26.1	16.9	39.7	**5.9**
15	67	6188	2003	3214.5	**4339.4**	2542.9	4276.8	67.6	48.1	**29.9**	58.9	30.9
15	100	5879	1567	2490	**3865.2**	2056.7	3236.1	73.3	57.6	**34.3**	65.0	45.0
20	33	13288	5646	7914.2	9741.9	6256.3	**10966.6**	57.5	40.4	26.7	52.9	**17.5**
20	67	11893	2949	4693.4	**7442.4**	3791.7	6303.3	75.2	60.5	**37.4**	68.1	47.0
20	100	11101	2103	3574.1	**6219.0**	2887	4556.5	81.1	67.8	**44.0**	74.0	59.0
25	33	21176	7631	11186.5	14584.0	8507.1	**14859.1**	64.0	47.2	31.1	59.8	**29.8**
25	67	19207	3821	6095.5	**10652.8**	4828.6	8061.4	80.1	68.3	**44.5**	74.9	58.0
25	100	18370	2534	4490.3	**8874.4**	3649.3	5808.4	86.2	75.6	**51.7**	80.1	68.4
30	33	31077	9255	13401	**19278.5**	10451	18403.3	70.2	56.9	**38.0**	66.4	40.8
30	67	28777	4830	7566.9	**14017.0**	5935.5	9885.3	83.2	73.7	**51.3**	79.4	65.6
30	100	27314	3198	5555	**11803.8**	4495.3	7120.8	88.3	79.7	**56.8**	83.5	73.9
35	33	43629	11107	16170.9	**24988.0**	12271.2	21875	74.5	62.9	**42.7**	71.9	49.9
35	67	39660	5631	8884.4	**17970.1**	6998.4	11635.8	85.8	77.6	**54.7**	82.4	70.7
35	100	38049	3917	6707	**15039.1**	5477.4	8271	89.7	82.4	**60.5**	85.6	78.3

solving CP1, CP2, CP3 and CP4 instances with $n \in \{30, 35\}$ and $d \in \{33\%, 67\%\}$ do not exceed 76 seconds per instance. For the remaining CP instances, largest computational time required for solving VS_1 relaxation is for CP3 instance with $n = 35$ and $d = 100$, and that

Table 8 OPesym instances: bounds and gaps

instance			lower bounds					Gap (%)				
n	$d(\%)$	UB	GL	$\widetilde{RLT}_1$	$\widetilde{RLT}_2$	VS_1	VS_2	GL	$\widetilde{RLT}_1$	$\widetilde{RLT}_2$	VS_1	VS_2
6	100	541.2	471.8	**539**	538.5	468.2	475.1	12.8	**0.4**	0.5	13.5	12.2
7	100	783.7	675.8	**781.4**	774.4	651.1	664.3	13.8	**0.3**	1.2	16.9	15.2
8	100	1020.1	887	**1016**	1001.1	807.5	826.1	13.0	**0.4**	1.9	20.8	19.0
9	100	1356	1168.1	**1347.9**	1326.8	1172.1	1199.9	13.9	**0.6**	2.2	13.6	11.5
10	100	1427.1	1224.4	**1420**	1395.0	1156.1	1197.4	14.2	**0.5**	2.2	19.0	16.1
11	100	1545.1	1324.6	**1540.5**	1508.9	1265.2	1315.9	14.3	**0.3**	2.3	18.1	14.8
12	100	1901.6	1623.5	**1894**	1849.8	1568.7	1626.1	14.6	**0.4**	2.7	17.5	14.5
13	100	2175.3	1861.1	**2168.8**	2114.0	1803.1	1889.5	14.4	**0.3**	2.8	17.1	13.1
14	100	2527.9	2164	**2522.9**	2456.7	2028.1	2121.3	14.4	**0.2**	2.8	19.8	16.1
15	100	2588.8	2237.9	**2578.5**	2515.8	2256.2	2351.1	13.6	**0.4**	2.8	12.8	9.2
16	100	2980.1	2524.6	**2962.3**	2884.9	2461.6	2575.2	15.3	**0.6**	3.2	17.4	13.6
17	100	3372.2	2837.4	**3342.1**	3251.9	2550.7	2667	15.9	**0.9**	3.6	24.4	20.9
18	100	3569	2999.2	**3551.2**	3435.0	2844.9	2995.1	16.0	**0.5**	3.8	20.3	16.1
30	100	8056.7	6608.5	**7922.0**	7579.2	6299.5	6753.0	18.0	**1.7**	5.9	21.8	16.2

Table 9 OPsym instances: bounds and gaps

instance			lower bounds					Gap (%)				
n	$d(\%)$	UB	GL	$\widetilde{RLT_1}$	$\widetilde{RLT_2}$	VS_1	VS_2	GL	$\widetilde{RLT_1}$	$\widetilde{RLT_2}$	VS_1	VS_2
6	100	258.4	147.5	253.5	254.8	246.4	**257.4**	42.9	1.9	1.4	4.6	**0.4**
7	100	326.8	167	310.4	317.6	294.6	**325.9**	48.9	5.0	2.8	9.9	**0.3**
8	100	438.5	203.9	399.7	414.5	361.9	**434.4**	53.5	8.8	5.5	17.5	**0.9**
9	100	534.9	232.2	466.3	496.7	418.8	**531.8**	56.6	12.8	7.1	21.7	**0.6**
10	100	653.9	240.3	529.9	584.4	473.2	**638.8**	63.3	19.0	10.6	27.6	**2.3**
11	100	785.9	250.4	584.3	668.3	513.4	**738.7**	68.1	25.7	15.0	34.7	**6.0**
12	100	918.5	256.1	646.8	755.3	565.4	**828.4**	72.1	29.6	17.8	38.4	**9.8**
13	100	1067.1	282.4	706.7	848.9	614.4	**900.3**	73.5	33.8	20.4	42.4	**15.6**
14	100	1249.8	300	799.1	968.8	693.7	**1006.1**	76.0	36.1	22.5	44.5	**19.5**
15	100	1390.2	287.8	815.8	1032.1	694.6	**1045.9**	79.3	41.3	25.8	50.0	**24.8**
16	100	1629.3	299.2	886.9	**1270.9**	761.7	1129.8	81.6	45.6	**22.0**	53.2	30.7
17	100	1823.8	324	961.9	**1258.7**	826.1	1223.8	82.2	47.3	**31.0**	54.7	32.9
18	100	2981	344.4	1467.0	**1998.0**	880.8	1298.2	88.4	50.8	**33.0**	70.5	56.5
20	100	2572.7	339	1132.8	**1608.9**	967.3	1441.8	86.8	56.0	**37.5**	62.4	44.0
30	100	6015.9	401.3	1664.1	**2843.1**	1396.8	2123.3	93.3	72.3	**52.7**	76.8	64.7

is 4 minutes. VS_1 bounds for instances OPsym and OPesym with $n \leq 20$ are computed in a few seconds, and for $n = 30$ in less than 134 seconds.

Computational time used to compute VS_2 bound for each CP instance with $n = 30$, $d = 100$ and $n = 35$, $d \in \{67\%, 100\%\}$ is two hours. Note that we allow our algorithm to run at most two hours. Computational times required for computing VS_2 bounds for other instances in Tables 4–9 are comparable to running times in Table 3.

Results in Tables 4–7 present bounds for CP instances and show that the GL bounds are significantly weaker than the other listed bounds. The results also show that $\widetilde{RLT_1}$ bounds are stronger than VS_1 bounds for all instances, while VS_2 bounds dominate $\widetilde{RLT_1}$ bounds for all instances except the CP1 instance with $n = 35$ and $d = 100$. The results also show that VS_2 bounds are not dominated by $\widetilde{RLT_2}$ bounds or vice versa.

Numerical results in Table 8 provide bounds for OPesym instances and show that VS_1 and VS_2 bounds are weaker than $\widetilde{RLT_1}$ and $\widetilde{RLT_2}$ bounds for all OPesym instances. This is due to the fact that there are not many violated cuts of type (23a)–(23d) for those instances.

Table 9 presents results for OPsym instances and shows that VS_2 bounds are stronger than $\widetilde{RLT_2}$ bounds for instances with less than 16 vertices. Moreover, VS_2 bounds are stronger than $\widetilde{RLT_1}$ bounds for all presented instances except for $n = 18$. Note that results in Table 8 and Table 9 present an average over 10 instances of a given size.

Pereira et al. [24] derive strong bounds for the QMSTP by using the idea of partitioning spanning trees into forests of a given fixed size. The resulting model can be seen as a generalization of the RLT relaxation. To obtain bounds, the authors introduce a bounding procedure based on Lagrangian relaxation. In [24], Table 1 the authors provide bounds for CP instances and $n = 25$. Our lower bound VS_2 is better than their bound only for CP3 instance with $n = 25$ density 33%. Namely, Pereira et al. [24] report bound 2652.5 and we compute 2730.3. Although bounds that result from the generalized RLT relaxation are stronger than the bounds obtained from the incomplete first level RLT relaxation, the former were not used within a Branch and Bound algorithm in [24]. Instead, the authors use $\widetilde{RLT_1}$ relaxation to solve several instances to optimality. Note that our VS_2 bounds are stronger than $\widetilde{RLT_1}$ bounds for all CP instances except for the CP1 instance with $n = 35$ and $d = 100$, and all OPsym instances except for $n = 18$.

6 Conclusion

This paper introduces a hierarchy of lower bounds for the quadratic minimum spanning tree problem. Our bounds exploit an extended formulation for the MSTP from [20] and the linear, exact formulation for the QMSTP from [1]. We prove that our simplest relaxation VS_0 is equivalent to the linearization based relaxation derived in Section 4.1, see Theorem 4.1. To improve the relaxation VS_0 we add facet defining inequalities of the Boolean Quadric Polytope. The resulting relaxations VS_1 and VS_2 are presented in Table 2. Our relaxations have a polynomial number of constraints and can be solved by a cutting plane algorithm.

On the other hand, all relaxations in the literature have an exponential number of constraints and most of them belong to the RLT type of bounds, see Section 3. The fact that our relaxations differ in both mentioned aspects from the other relaxations, enables us to efficiently compute bounds that are not dominated by the RLT type of bounds.

Acknowledgements We would like to thank Frank de Meijer for an insightful discussions on linearization based bounds. We also thank Renate van der Knaap for programming heuristics algorithms and computing upper bounds.

References

1. Assad, A., Xu, W.: The quadratic minimum spanning tree problem. Nav. Res. Logist. **39**, 399–417 (1992)
2. Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B.: Julia: A fresh approach to numerical computing. SIAM Rev. **59**, 65–98 (2017)
3. Carraresi, P., Malucelli, F.: A new lower bound for the quadratic assignment problem. Oper. Res. **40**, 22–27 (1992)
4. Chiang, T.-C., Liu, C.-H., Huang, Y.-M.: A near-optimal multicast scheme for mobile ad hoc networks using a hybrid genetic algorithm. Expert Syst. Appl. **33**, 734–742 (2007)
5. Chou, W., Kershenbaum, A.: A unified algorithm for designing multidrop teleprocessing networks. IEEE Trans. Commun. **22**, 1762–1772 (1974)
6. Cordone, R., Passeri, G.: Solving the quadratic minimum spanning tree problem. Appl. Math. Comput. **218**, 11597–11612 (2012)
7. Ćustić, A., Punnen, A.P.: A characterization of linearizable instances of the quadratic minimum spanning tree problem. J. Comb. Optim. **35**, 436–453 (2018)
8. Ćustić, A., Zhang, R., Punnen, A.P.: The quadratic minimum spanning tree problem and its variations. Discrete Optim. **27**, 73–87 (2018)
9. de Meijer, F., Sotirov, R.: The quadratic cycle cover problem: special cases and efficient bounds. J. Comb. Optim. **39**, 1096–1128 (2020)

10. Edmonds, J.: Matroids and the greedy algorithm. Math. Program. **1**, 127–136 (1971)
11. Gao, J., Lu, M.: Fuzzy quadratic minimum spanning tree problem. Appl. Math. Comput. **164**, 773–788 (2005)
12. Gilmore, P.C.: Optimal and suboptimal algorithms for the quadratic assignment problem. J. Soc. Ind. Appl. Math. **10**, 305–313 (1962)
13. Guimarães, D.A., da Cunha, A.S., Pereira, D.L.: Semidefinite programming lower bounds and branch-and-bound algorithms for the quadratic minimum spanning tree problem. Eur. J. Oper. Res. **280**, 46–58 (2020)
14. Hu, H., Sotirov, R.: The linearization problem of a binary quadratic problem and its applications. Ann. Oper. Res. **307**, 229–249 (2021)
15. Kaibel, V.: Extended formulations in combinatorial optimization (2011). http://www.mathopt.org/Optima-Issues/optima85.pdf
16. Kruskal, J.B.: On the shortest spanning subtree of a graph and the traveling salesman problem. Proc. Amer. Math. Soc. **7**, 48–50 (1956)
17. Lawler, E.L.: The quadratic assignment problem. Manag. Sci. **9**, 586–599 (1963)
18. Lozano, M., Glover, F., García-Martínez, C., Rodríguez, F.J., Martí, R.: Tabu search with strategic oscillation for the quadratic minimum spanning tree. IIE Trans. **46**, 414–428 (2014)
19. Magnanti, T.L., Wolsey, L.A.: Optimal trees. In: Handbooks in Operations Research and Management Science, vol. 7, chap. 9, pp. 503–615. Elsevier (1995)
20. Martin, K.R.: Using separation algorithms to generate mixed integer model reformulations. Oper. Res. Lett. **10**, 119–128 (1991)
21. Öncan, T., Punnen, A.P.: The quadratic minimum spanning tree problem: A lower bounding procedure and an efficient search algorithm. Comput. Oper. Res. **37**, 1762–1773 (2010)
22. Padberg, M.W.: The boolean quadric polytope: Some characteristics, facets and relatives. Math. Program. **45**, 139–172 (1989)
23. Palubeckis, G., Rubliauskas, D., Targamadz, A.: Metaheuristic approaches for the quadratic minimum spanning tree problem. Inform. Tech. Control **29**, 257–268 (2010)
24. Pereira, D.L., Gendreau, M., da Cunha, A.S.: Lower bounds and exact algorithms for the quadratic minimum spanning tree problem. Comput. Oper. Res. **63**, 149–160 (2015)
25. Pereira, D.L., Gendreau, M., da Cunha, A.S.: Branch-and-cut and branch-and-cut-and- price algorithms for the adjacent only quadratic minimum spanning tree problem. Networks **65**, 367–379 (2015)
26. Prim, R.C.: Shortest connection networks and some generalizations. Bell Syst. Tech. J. **36**, 1389–1401 (1957)
27. Punnen, A.P.: Combinatorial optimization with multiplicative objective function. Int. J. Oper. Quant. Manag. **7**, 205–209 (2001)
28. Rhys, J.M.W.: A selection problem of shared fixed costs and network flows. Manag. Sci. **17**, 200–207 (1970)
29. Rostami, B., Malucelli, F.: Lower bounds for the quadratic minimum spanning tree problem based on reduced cost computation. Comput. Oper. Res. **64**, 178–188 (2015)
30. Rostami, B., Chassein, A., Hopf, M., Frey, D., Buchheim, C., Malucelli, F., Goerigk, M.: The quadratic shortest path problem: complexity, approximability and solution methods. Eur. J. Oper. Res. **268**, 473–485 (2018)
31. Sherali, H.D., Adams, W.P.: A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. SIAM J. Discrete Math. **3**, 411–430 (1990)
32. Sherali, H.D., Adams, W.P.: A hierarchy of relaxations and convex hull characterizations for mixed-integer zero–one programming problems. Discrete Appl. Math. **52**, 83–106 (1994)
33. Sherali, H.D., Adams, W.P.: A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems. Nonconvex Optimization and Its Applications, vol. 31. Springer, New York (2013)
34. Soak, S.-M., Corne, D., Ahn, B.-H.: A new evolutionary algorithm for spanning-tree based communication network design. IEICE Trans. Commun. **E88-B**, 4090–4093 (2005)
35. Soak, S.-M., Corne, D.W., Ahn, B.-H.: The edge-window-decoder representation for tree-based problems. IEEE Trans. Evol. Comput. **10**, 124–144 (2006)
36. Sundar, S., Singh, A.: A swarm intelligence approach to the quadratic minimum spanning tree problem. Inf. Sci. **180**, 3182–3191 (2010)
37. Zhout, G., Gen, M.: An effective genetic algorithm approach to the quadratic minimum spanning tree problem. Comput. Oper. Res. **25**, 229–237 (1998)