The Quadratic Minimum Spanning Tree Problem: A Lower Bounding Procedure and an Efficient Search Algorithm

Temel Öncan * Abraham P. Punnen ⋆

* *Department of Industrial Engineering, Galatasaray University*
Ortaköy İstanbul 34357, TÜRKİYE

⋆ *Department of Mathematics, Simon Fraser University*
13450 102nd AV, Surrey, British Columbia,V3T 5X3, CANADA

Abstract

In this work we address the Quadratic Minimum Spanning Tree Problem (QMSTP) known to be NP-hard. Given a complete graph, the QMSTP consists of determining a minimum spanning tree (MST) considering interaction costs between pairs of edges to be modelled. A Lagrangian relaxation procedure is devised and an efficient local search algorithm with tabu thresholding is developed. Computational experiments are reported on randomly generated test instances. The local search heuristic yields at least as well as the results obtained with a previous Genetic Algorithm and the Lagrangian relaxation procedure gives tightest lower bound values for all instances.

Keywords: *Quadratic Minimum Spanning Tree Problem, Lagrangian Relaxation, Local search*

1 Introduction

Consider an undirected complete graph $G = (V, E)$ where $V = \{v_1, \ldots, v_n\}$ is the vertex set and $E = \{e_1, \ldots, e_m\}$ is the edge set. The Quadratic Minimum Spanning Tree Problem (QMSTP) consists of determining a minimum spanning tree (MST) considering interaction costs between pairs of edges to be modelled. The QMSTP has applications in transportation, telecommunication, irrigation and energy distribution. The problem was first introduced by Assad and Xu [1] who have shown that it is NP-hard. In their seminal paper, a variant of the QMSTP, namely adjacent only QMSTP (AQMSTP), is also discussed. In the AQMSTP, only the costs of the adjacent edge pairs are considered. Interaction costs of all nonadjacent edge pairs are assumed to be zero. Assad and Xu [1] have devised a lower bounding algorithm and two heuristic algorithms for this problem. Then, they incorporated these algorithms into a branch-and-bound scheme for obtaining exact solutions. The authors have reported that they have solved QMSTP instances up to 15 vertices to optimality. A genetic algorithm was devised by Zhou and Gen [7] and tested on randomly generated test instances.

The remainder of this paper is organized as follows. Section 2 introduces the QMSTP. Section 3 presents lower bounding procedures for the problem. This is followed in Section 4 by a local search algorithm for the QMSTP. Section 5 is where we present our computational experiments and conclusions

2 The Quadratic Minimum Spanning Tree Problem

In this section we present a formulation of the QMSTP proposed by Assad and Xu [1]. Using the decision variables $x_p = 1$ if and only if edge p is in the solution and $y_{pq} = 1$ if and only if both edges p and q are in the solution Assad and Xu [1] proposed the following formulation:

$$\min \sum_{p}^{m}\sum_{q}^{m} a_{pq}y_{pq} + \sum_{p=1}^{m} b_p x_p \tag{1}$$

$$\sum_{q=1}^{m} y_{pq} = (n-1)x_p \qquad p = 1,\ldots,m \tag{2}$$

$$\sum_{p=1}^{m} y_{pq} = (n-1)x_q \qquad q = 1,\ldots,m \tag{3}$$

$$y_{pp} = x_p \qquad p = 1,\ldots,n \tag{4}$$

$$0 \leq y_{pq} \leq 1 \qquad \forall p,q \tag{5}$$

$$\mathbf{x} \in \mathcal{T} \tag{6}$$

where $\mathbf{x}$ denotes solution vector consisting of x_p values. Here, b_p indicates the cost of selecting edge p and a_{pq} stands for the interaction cost associated with each pair of selected edges p and q. Note that if the interaction cost of edges p and q is a then it is assumed that $a_{pq} = a_{qp} = a/2$ holds for $p, q = 1,\ldots,m$ and $p \neq q$.

One can observe that, in the objective function (1) the quadratic terms have been eliminated at the cost of the additional decision variables y_{pq}. In this formulation constraints (2) denote that whenever edge p is selected then, the sum on q of y_{pq} variables must be exactly equal to $(n-1)$. Constraints (3) indicate that when edge q is selected then the sum on p of y_{pq} variables must be exactly equal to $(n-1)$. Observe that constraints (2)-(5) assure the linearization of the objective function. The QMSTP is known to be solvable in polynomial time if $\alpha_{pq} = u_p v_q$ where $u_1, u_2 \ldots, u_m$ and $v_1, v_2, \ldots, v_m$ are given constants and $u_i \geq 0, v_i \geq 0$ for all i. If we drop the restriction that $u_i \geq 0, v_i \geq 0$ then QMSTP is NP-hard even if $\alpha_{pq} = u_p v_q$ [6].

3 Lower Bounding Procedures

To compute lower bound values for the QMSTP we propose a Lagrangian relaxation scheme of an extended formulation of the QMSTP. Before introducing our Lagrangian scheme we will present the Gilmore and Lawler lower bound and the levelling procedure devised by Assad and Xu [1].

3.1 Gilmore and Lawler Lower Bound for the QMSTP

One of the earliest lower bounding schemes for the Quadratic Assignment Problem (QAP) is the one proposed by Gilmore [3] and Lawler [5]. Their approach can also be adopted to the QMSTP. For that purpose, we first solve m MSTPs by setting $x_q = 1$ for each edge q. In other words, we solve the following m problems, for each edge $p = 1,\ldots,m$.

$$e_p = \min\left\{ \sum_{q=1}^{m} a_{pq}x_q \,|\mathbf{x} \in \mathcal{T}; x_p = 1; x_q \in \{0,1\}\,\forall q \right\} \tag{7}$$

Hence, we obtain Gilmore and Lawler lower bound (GLB) for the QMSTP. For the QAP case, Gilmore and Lawler bounds are easy to compute but they are weaker than other state-of-the-art lower bounding schemes.

3.2 The Assad and Xu's Levelling Procedure

By using the transformation

$$b_p(\gamma) = b_p - (n-2)\gamma_p \qquad p = 1, \ldots, m$$
$$a_{pq}(\gamma) = a_{pq}(\gamma) + \gamma_q \qquad p, q = 1, \ldots, m$$

Assad and Xu [1] have proposed the following lower bound for the QMSTP:

$$AX(\gamma) = \min \left\{ \sum_{p=1}^{m} f_p(\gamma) x_p \,|\mathbf{x} \in \mathcal{T} \right\} \tag{8}$$

where

$$f_p(\gamma) = \min \left\{ b_p(\gamma) + \sum_{\substack{q=1 \\ q \neq p}}^{m} a_{pq}(\gamma) x_q \,|x_p = 1 \text{ and } \mathbf{x} \in \mathcal{T} \right\} \qquad p = 1, \ldots, m \tag{9}$$

and γ is a m dimensional parameter vector. Observe that the lower bound $AX(\gamma)$ proposed by Assad and Xu [1] depends on the parameters γ. To determine the values of the γ parameters yielding the largest lower bound $AX^* = \max_{\gamma} \{AX(\gamma)\}$, the authors proposed a levelling procedure. For $\gamma^1 = (0, \ldots, 0)$, the lower bound value $AX(\gamma^1)$ obtained by the levelling procedure is exactly the same as the lower bound yielded by the Gilmore and Lawler approach. Hence we can say that, the levelling procedure starts with the Gilmore and Lawler bound and iteratively improves it until the stopping criterion is satisfied.

3.3 A Lagrangian Relaxation Scheme

In this section we propose a Lagrangian relaxation scheme based on a new formulation of the QMSTP. In this formulation we define the following decision variables: s_{ij} equal to 1 if and only if a link from vertex i to vertex j for $i < j$ is established and w_{ijkl} equal to 1 if and only if links from vertex i to vertex j for $i < j$ and from vertex k to vertex l for $k < l$ are established. The parameter c_{ijkl} denotes the interaction cost if links from vertex i to vertex j and from vertex k to vertex l are established. The parameter b_{ij} stands for the cost of connecting vertex i to vertex j. Note that if the interaction cost of the links from vertex i to vertex j and from vertex k to vertex l is c then $c_{ijkl} = c_{jikl} = c/2$ and $c_{ijlk} = c_{ijkl} = c/2$ hold for $i, j, k, l = 1, \ldots, n$; $i < j$ and $k < l$. It is assumed that $c_{ijij} = 0$ for $i, j = 1, \ldots, n$ and $i < j$. In the following formulation we have implicitly assumed that $w_{ijkl} = s_{ij}s_{kl}$ holds and hence the objective function is now linearized.

$$\text{NEW:} \qquad \min \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \sum_{k=1}^{n} \sum_{\substack{l=1 \\ l \neq k}}^{n} c_{ijkl} w_{ijkl} + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} b_{ij} s_{ij} \tag{10}$$

$$\text{subject to} \quad \sum_{\substack{i=1 \\ i \neq j}}^{n} w_{ijkl} \geq s_{kl} \qquad j, k, l = 1, \ldots, n; k < l \tag{11}$$

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} w_{ijkl} \geq s_{kl} \qquad i, k, l = 1, \ldots, n; k < l \tag{12}$$

$$\sum_{\substack{k=1 \\ k \neq l}}^{n} w_{ijkl} \geq s_{ij} \qquad i, j, l = 1, \ldots, n; i < j \tag{13}$$

$$\sum_{\substack{l=1 \\ l \neq k}}^{n} w_{ijkl} \geq s_{ij} \qquad i, j, k = 1, \ldots, n; i < j \tag{14}$$

$$\sum_{k=1}^{n}\sum_{\substack{l=1\\l\neq k}}^{n} w_{ijkl} \leq (n-1)s_{ij} \qquad i,j=1,\ldots,n; i<j \tag{15}$$

$$\sum_{i=1}^{n}\sum_{\substack{j=1\\j\neq i}}^{n} w_{ijkl} \leq (n-1)s_{kl} \qquad k,l=1,\ldots,n; k<l \tag{16}$$

$$w_{ijij} = s_{ij} \qquad i,j=1,\ldots,n; i<j \tag{17}$$

$$\mathbf{s} \in \mathcal{S} \tag{18}$$

$$s_{ij} \in \{0,1\} \qquad i,j=1,\ldots,n; i<j \tag{19}$$

$$0 \leq w_{ijkl} \leq 1 \qquad i,j,k,l=1,\ldots,n; i\neq j; k\neq l \tag{20}$$

where $\mathbf{s}$ denotes solution vector consisting of s_{ij} values and $\mathcal{S}$ stands for the set of spanning trees.

At this stage we will show the validity of this formulation by the following result. The proof is based on the observation that the above formulation is a disaggregated version of the formulation (1)-(6) proposed by Assad and Xu [1].

Proposition 1 *The formulation (10)-(20) is a disaggregation of the formulation (1)-(6).*

Proof. To show this, first sum up constraints (11) over indices j and using the identities $s_{kl} = x_q$ and $w_{ijkl} = y_{pq}$, where edge q is $\{k,l\}$ and edge p is $\{i,j\}$, we can obtain the constraints (2). By the same identities, together with (11), we can use (16) to get (3). Similarly it is possible to derive constraints (4) which completes the proof. ∎

Now consider a Lagrangian relaxation of this formulation with multipliers α_{jkl} for constraints (11) and multipliers β_{ikl} for constraints (12). The Lagrangian function $L(\alpha,\beta)$ is thus defined by

$$\min \sum_{i=1}^{n}\sum_{\substack{j=1\\j\neq i}}^{n}\sum_{k=1}^{n}\sum_{\substack{l=1\\l\neq k}}^{n} \overline{c}_{ijkl} w_{ijkl} + \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \overline{b}_{ij} s_{ij} \tag{21}$$

$$\text{subject to} \quad (13)-(20) \tag{22}$$

where

$$\overline{c}_{ijkl} = c_{ijkl} - \alpha_{jkl} - \beta_{ikl} \qquad i,j,k,l=1,\ldots,n; i\neq j; k\neq l$$

$$\overline{b}_{ij} = b_{ij} + \sum_{k=1}^{n}\alpha_{kij} + \sum_{l=1}^{n}\beta_{lij} \qquad i,j=1,\ldots,n; i<j$$

For the QAP, which has a similar structure to the QMSTP, Frieze and Yadegar [2] have employed a decomposition approach within a Lagrangian relaxation scheme. Inspired with this idea, we adopt their approach as follows. We sequentially fix one edge and solve the MSTP such that the selected edge is in the solution. That is to say we solve the following m problems for each edge.

$$\overline{d}_{ij} = \min \sum_{k=1}^{n-1}\sum_{l=k+1}^{n} \overline{c}_{ijkl} t_{kl} \tag{23}$$

$$\text{subject to} \quad \mathbf{t} \in \mathcal{S} \tag{24}$$

$$t_{kl} \in \{0,1\} \qquad k,l=1,\ldots,n; k<l; k\neq i; l\neq j; \tag{25}$$

$$t_{ij} = 1 \tag{26}$$

where $\mathbf{t}$ denotes solution vector consisting of t_{kl} values.

Then for each edge we obtain the objective function values $\overline{d}_{ij}$. These values, which are computed for each edge, can be interpreted as the cost of being in the solution. Then using the costs $\overline{d}_{ij}$ we solve the following MSTP:

$$L(\alpha,\beta) = \min \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(\overline{d}_{ij} + \overline{b}_{ij}\right) s_{ij} \tag{27}$$

$$\text{subject to} \quad \mathbf{s} \in \mathcal{S} \tag{28}$$

$$s_{ij} \in \{0,1\} \qquad i,j = 1,\ldots,n; i<j \tag{29}$$

The solution of the above problem yields Gilmore and Lawler lower bound to the QMSTP. The Lagrangian relaxation subproblem will be solved by inspection as proposed by Frieze and Yadegar for the QAP case. Whenever $s^*_{ij} = 0$ let $w^*_{ijkl} = 0$. If $s^*_{ij} = 1$ then let $w^*_{ijij} = 1$ and let w^*_{ijkl} be the value of t_{kl} in the solution to (23)-(26). Now we will prove the following result.

Proposition 2 *(s^*, w^*) satisfy (13)-(20)*

Proof. First of all note that constraints (17)–(20) are satisfied by definition. Whenever $s_{ij} = 1$ in the solution of (27)-(29), then $t_{ij} = 1$ in the solution of (23)-(26), $(n-2)$ other edges, forming a spanning, are also selected. Then the setting of w_{ijkl}, variables as defined above, satisfy the constraints (15) since there are at most $(n-1)$ edges in a spanning tree obtained in the solution of (23)-(26). On the other hand, when $s_{ij} = 0$, all w_{ijkl} are set to zero and then again constraints (15) are satisfied. Similarly, we can show that constraints (16) are also satisfied.

In order to show that constraints (13) are satisfied, first consider the case $s_{ij} = 1$. Recall that w_{ijkl} are set according to the values of t_{kl} variables in the solution of (23)-(26). Hence there is at least one edge connected to the vertex l as a result of the spanning tree, which contains edge (i,j), obtained by (23)-(26). Therefore, constraints (13) are satisfied for the case $s_{ij} = 1$. On the other hand, for the case $s_{ij} = 0$ these constraints are satisfied by definition. Similarly, one can show that constraints (14) are also satisfied. This completes the proof. ■

4 A Local Search Algorithm for the QMSTP

In this section we propose a neighborhood and a local search algorithm for the QMSTP. The neighborhood is straightforward and considered by numerous authors in the spanning tree context (e.g. network simplex method with linear objective function.) Before defining the neighborhood scheme we will give the following definitions. A basic edge is an edge in the current feasible solution (spanning) tree $t \in \mathcal{T}$. All other edges, which are not in the current tree, are defined as nonbasic edges. The $N^k_{(t)}$ neighborhood is as follows. Randomly select k basic edges of the current tree $t \in \mathcal{T}$. Replace each basic edge $i \in t$ with a nonbasic edge $j \notin t$ such that a spanning tree is maintained at the end of this operation. As for instance, given a complete graph, the 1-edge neighborhood $N^1_{(t)}$ considers all possible $(n-2)$ spanning trees as the neighborhood of the current solution tree $t \in \mathcal{T}$. The choice of the neighborhood is a critical issue which affects both the efficiency and the accuracy of the search algorithm. For the sake of efficiency we confined ourselves only to employ $N^1_{(t)}$ neighborhood which seemed to yield good solutions in reasonable CPU time. The design of our local search algorithm was inspired on the tabu thresholding idea proposed by Glover [4]:

Algorithm: **Random Local Search With Tabu Thresholding**
begin
Input: k : degree of freedom,
$iter(k)$: iteration limit per degree of freedom
$count\_limit$: iteration limit of the local search
Output: The best solution found and its value
$T \longleftarrow$A random spanning tree (a feasible solution);

```
Compute f(T) (objective function value of T)
solution ⟵ T; value ⟵ f(T); count = 0;
repeat
    while T is not a local minimum do
        f(T_pq) = min{f(T_rs) : T_rs ∈ N(T)};
        T ⟵ T_pq;
    end
    /* A local minimum reached.  update the best solution, if necessary
    if value > f(T) then solution ⟵ T; value ⟵ f(T);
    /* Random move starts */
    repeat
        f(T_pq) := min{f(T_rs) : T_rs ∈ N(T)};
        if value > f(T_pq) then
            T ⟵ T_pq;
            value ⟵ f(T_pq);
            i := iter(k);
            /* Random move step terminated by aspiration criterion
        else
            j ⟵ A random integer in [1, k];
            T_uv ⟵ j^th best element of {f(T_rs) : T_rs ∈ N(T)};
            T ⟵ T_uv;  /* Random move */
            i := i + 1;
        end
    until i = iter(k);
    count := count + 1;
    /* Random move phase is over.  New local search begins.  */
    until count ≥ count_limit;
    Output solution value and STOP the algorithm
end
```

5 Computational Experiments and Conclusions

There are no standard test problems available for the QMSTP. Thus, we have randomly generated our test instances with size ranging from $n = 6$ to $n = 50$. In random test instance generation we have followed the method proposed by Assad and Xu [1]. All random test instances consist of complete graphs with n vertices and $m = n(n-1)/2$ edges. The algorithms are coded in C++ and tested on a Pentium IV PC with a 3GHz CPU and 2 GB RAM.

In Table 1, we present the average computational results for the lower bounding schemes. The second and third columns are for the average lower bound values and average CPU times respectively which we obtain with the Gilmore and Lawler lower bounding approach. The fourth and fifth columns present the average lower bound values and average CPU times respectively that are obtained by Assad and Xu's levelling procedure. The sixth and seventh columns stand for the average lower bound values and average CPU times respectively, which are observed with the Lagrangian relaxation approach. In the last two columns we present the percent improvements of the Assad and Xu levelling procedure and the Lagrangian relaxation approach over the Gilmore and Lawler lower bounding scheme, respectively. The formula used are $(z_{AX} - z_{GL})/z_{GL}$ and $(z_{LR} - z_{GL})/z_{GL}$, where z_{GL}, z_{AX}, and z_{GL} stand for the lower bound values obtained with the Gilmore and Lawler lower bounding approach, the Assad and Xu's levelling procedure and the Lagrangian relaxation approach, respectively. The overall average percent improvement of the Assad and Xu's levelling procedure over the Gilmore and Lawler lower bounding approach is % 150.5 while the overall average percent improvement of the Lagrangian relaxation approach over the Gilmore and Lawler lower bounding approach is % 212.4. From these results we can conclude that at the expense

of the increase in CPU time requirement, the Lagrangian relaxation approach yield better lower bound values than the ones obtained by the Assad and Xu levelling procedure.

In Table 2, computational experiments with the local search algorithm on random test instances are presented. The second and third columns are for the average upper bound values and average CPU times respectively that we have obtained with the first heuristic (H1) by Assad and Xu [1]. The fourth and fifth columns present the average upper bound values and average CPU times respectively which are obtained with the second heuristic (H2) by Assad and Xu [1]. The sixth and seventh columns are for the average upper bound values and average CPU times respectively, obtained with the local search heuristic. The eighth column denotes the average improvement obtained with the local search algorithm over the best solution obtained with the H1 and H2. To compute the improvement obtained with the local search algorithm over the best solution obtained with the H1 and H2 we have employed following formulae $(\min\{z_{H_1}, z_{H_2}\} - z_{LS}) / \min\{z_{H_1}, z_{H_2}\}$ where z_{H_1}, z_{H_2},and z_{LS} stand for the upper bound values obtained with H1, H2 and the local search heuristic, respectively. The ninth and tenth columns respectively stand for the maximum improvements and minimum improvements obtained with the local search algorithm over the best of the solution obtained with the H1 and H2.

Table 1: Summary of Average Computational Experiments with Lower Bounding Schemes on Random Test Instances

	Gilmore-Lawler		Assad-Xu		Lagrangian Rel.		AX vs. GL	LR vs. GL
	LB	CPU(s)	LB	CPU(s)	LB	CPU(s)	% dev	% dev
Average	273.6	0.0	726.0	0.1	913.8	232.8	150.5	212.4

Table 2: Summary of Average Computational Experiments with the Local Search Algorithm on Random Test Instances

	Assad-Xu		Assad-Xu				Aver	Max	Min
	H1	CPU(s)	H2	CPU(s)	LS	CPU(s)	Imp	Imp	Imp
Average	3152.8	0.0	2594.5	0.1	2510.4	303.8	4.4	9.6	0.8

In their early work, Zhou and Gen [7] had reported the best solution obtained with the heuristics devised by Assad and Xu, can be improved by average 9.6% and at most 12.83 % with their GA approach. These results are collected based on only 16 instances with number of nodes (n) varying from 6 to 50. Unfortunately, we were not able to reach these instance neither obtain their GA codes in order to perform a comparative analysis of our local search algorithm with their GA. Considering the best solutions obtained with the local search heuristic, we observe that the best solutions obtained with the H1 and H2, are improved by average 9.6 % and at most 11.7 % with our local search heuristic on the random test instances with the same size as the ones by Zhou and Gen [7]. From this point of view, we can assert that the local search algorithm is at least as well as the GA proposed by Zhou and Gen [7] with a much simpler algorithmic structure than the GA.

Acknowledgments

The first author acknowledges the support by Galatasaray University Research Foundation Grant 09.402.005.

References

[1] A. Assad and W. Xu, *The Quadatic Minimum Spanning Tree Problem.* Naval Research Logistics, **39**, (1991), pp. 399–417.

[2] A.M. Frieze and J. Yadegar, *On The Quadratic Assignment Problem.* Discrete Applied Mathematics, **5**, (1982), pp. 89–98.

[3] P.C. Gilmore, *Optimal and Suboptimal Algorithms for the Quadratic Assignment Problem*, SIAM Journal of Applied Mathematics, **10**, (1962), pp. 305–313.

[4] F. Glover, *Tabu Thresholding: Improved Search by Nonmonotonic Trajectories*, ORSA Journal on Computing, **7**, (1995), pp. 426–442.

[5] E.L. Lawler, *The Quadratic Assignment Problem*, Management Science, **19**, (1963), pp. 586–590.

[6] A.P. Punnen, *Combinatorial Optimization with Multiplicative Objective Function*, International Journal of Operations and Quantitative Management, **3**, (2001), pp. 222–227.

[7] G. Zhou and M. Gen, *An Effective Genetic Algorithm Approach to the Quadratic Minimum Spanning Tree Problem*, Computers and Operations Research, **25**, (1998), pp. 229–237.