Theoretical Computer Science 704 (2017) 87–91

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Note

Upgrading min–max spanning tree problem under various cost functions

Ali Reza Sepasian [a,*], Ehsan Monabbati [b]

[a] Department of Mathematics, Fasa University, Fasa, Iran
[b] Department of Mathematics, Alzahra University, Tehran, Iran

ARTICLE INFO

Article history:
Received 11 February 2017
Received in revised form 26 July 2017
Accepted 7 August 2017
Available online 31 August 2017
Communicated by D.-Z. Du

Keywords:
Location problems
Upgrading problems
Min–max spanning tree

ABSTRACT

This paper addresses upgrading min–max spanning tree problem (MMST). Given a graph $G(V, E)$, the aim of this problem is to modify edge weights under certain limits and given budget so that the MMST with respect to perturbed graph improves as much as possible. We present a complexity result for general non-decreasing cost functions. In special case, it is shown that the problem under linear and sum-type Hamming cost function can be solved in $O(|E|^2)$ and $O(|E| \log |E| \log |V|)$ time, respectively.

1. Introduction

Usually the instances of classical network optimization problems are static and realistic, but, applications often admit some improvements of parameters under specific circumstances. This leads to the area of network "upgrading" problems. In these special network modification problems, one may invest a budget in order to change the parameters (weights of edges or vertices) of the given network within certain limits such that the optimal objective value with respect to the modified parameters is minimized while the topological structure of the graph remains unchanged [1,2]. In this paper we consider the upgrading min–max spanning tree (MMST) problem where a budget for reducing the weights of edges is assigned and the edge weights can be modified within given intervals. In this kind of problem, the network is modified before finding the MMST on the network.

Upgrading approach have already been applied to several classical optimization problems. For instance, Fulkerson and Harding [3] and Hambrusch and Hung-Yi Tu [4] investigated upgrading the shortest and longest path problems. Upgrading the network flow problem is considered by Phillips [5]. Gassner dealt with up- and downgrading 1-center and the 1-median problems in [6] and [7,8], respectively. Sepasian and Rahbarnia [9] proposed a linear time algorithm for solving upgrading 1-median on paths. Some authors also investigated upgrading minimum spanning tree problem. Dragmeister et al. [10], Frederickson and Solis-Oba [11] and Krumke et al. [12–14], developed up- and downgrading Steiner and minimum spanning tree problems. Another version of upgrading minimum spanning tree problem is investigated by Krumke et al. [15], [16] and Alvarez et al. [17] in which the weight of each edge is a function of the weights of two vertices connected to it. The aim is modifying the vertex weights in order to upgrading the minimum spanning tree.

* Corresponding author.
E-mail addresses: Sepasian@fasau.ac.ir (A.R. Sepasian), e.monabbati@alzahra.ac.ir (E. Monabbati).

http://dx.doi.org/10.1016/j.tcs.2017.08.006
0304-3975/© 2017 Published by Elsevier B.V.

Given a connected graph $G = (V, E)$, with $|V| = m$ and $|E| = n$, a nonnegative weight w_e is assigned to each $e \in E$. Let $\mathcal{T}$ denote the collection of all spanning trees of G. The weight of a spanning tree $T \in \mathcal{T}$ is defined as $w(T) = \max\{w_e | \; e \in T\}$. The MMST is a tree $T^* \in \mathcal{T}$ with smallest weight, i.e. $w(T^*) = \min\{w(T) | \; T \in \mathcal{T}\}$. The MMST problem has applications in communication network (under the name min–max broadcasting tree) [18–22], and molecular biology [23].

Camerini [24] proposed a linear time algorithm to solve the problem. After that, different versions of the problem was studied. Ishii and Nishida [25] investigated the stochastic version of the problem in which edge weights are random variables. Berman et al. [26] dealt with the constrained version where the sum of edge weights of the tree should not exceed a given upper bound. Afterward, their algorithm was improved by Punnen and Nair [27]. Recently, Anderson and Ras [28] studied the MMST with additional condition that every vertex degree is bounded above in MMST.

In this paper, we investigate upgrading MMST problem. The upgrading MMST problem is aimed at finding new values for edge weights and a new MMST associated with these new parameters, so that this tree is the best over all allowed parameters. Put simply, the aim is to find a MMST of the network when edge weights change.

Some kind of network improvement approaches was applied to MMST by several authors. Liu and Yao [29] introduced the inverse MMST in which the edge weights are modified within given budget so that a candidate spanning tree becomes the MMST. They considered the problem under the sum-type Hamming distance. Also, the constrained inverse MMST under the weighted Hamming distance was studied by Liu and Wang [30].

In the rest of this section the problem is introduced and formulated formally. In section 2 the problem is investigated under non-decreasing and linear cost functions, and a polynomial time algorithm is proposed. In section 3 the problem is considered under sum-type Hamming cost function.

Let us model the problem formally. Assume that the weight of each edge $e \in E$, can be decreased with a given cost, and the modified weight is $\tilde{w}_e$, that is $\tilde{w}_e \le w_e$. Each modified edge weight is restricted by a nonnegative lower bound $\underline{w}_e$ i.e. $\tilde{w}_e \ge \underline{w}_e$. The weight of a tree T with respect to the new vector of edge weights, $\tilde{w}$, is denoted by $\tilde{w}(T)$. Assume that the cost of reducing the weight of edge $e \in E$ is given by non-decreasing function $f_e(x)$. Moreover, let B be a positive budget which limits the cost of edge weight reductions. Using the above notations, the upgrading MMST problem is formally formulated as:

$$\begin{aligned} \min_{T \in \mathcal{T}} \quad & \tilde{w}(T) \\ \text{s.t.} \quad & \sum_{e \in E} f_e(w_e - \tilde{w}_e) \le B \\ & \underline{w}_e \le \tilde{w}_e \le w_e, \qquad \forall e \in E \end{aligned} \tag{1}$$

2. Upgrading MMST problem under non-decreasing cost functions

In this section we assume that cost function $f_e(x)$ for each edge $e \in E$ is non-decreasing. Let ν^* be the optimal value of (1). It is easy to see that ν^* is bounded above by the objective value of current MMST, $\bar{\nu}$, and is bounded below by $\min_{e \in E} \underline{w}$. A tighter lower bound for ν^* is the weight of MMST, $\underline{T}$, corresponding to weight vector $\underline{w}$. Denote this lower bound by $\underline{\nu}$. Observe that $\underline{\nu}$ and $\bar{\nu}$ can be found in linear time by [24]. Thus, ν^* belongs to interval $\mathcal{V} = [\underline{\nu}, \bar{\nu}]$. From now on we assume that $\sum_{e \in \underline{T}} f_e(w_e - \underline{w}_e) > B$, since otherwise, $\underline{T}$ is the upgraded MMST with weight $\nu = \underline{\nu}$ and nothing is left to solve. By this assumption, the whole budget will be used in order to get the optimal solution.

Before we consider how to solve the problem, we concentrate on a problem closely related to problem (1). Indeed, given $\nu \in \mathcal{V}$, the problem is to make a MMST with weight ν at minimum cost so that new weights satisfy bound restrictions.

For $\nu \in \mathcal{V}$, consider the new graph $G_\nu = (V, E)$ where the underlying graph is G and the weight of each $e \in E$, $\lambda_e(\nu)$, is defined as

$$\lambda_e(\nu) := \begin{cases} 0 & \text{if } \nu \ge w_e \\ f_e(w_e - \nu) & \text{if } \underline{w}_e \le \nu < w_e \\ B + 1 & \text{if } \nu < \underline{w}_e \end{cases} \tag{2}$$

For a spanning tree T of G_ν, define $D_T(\nu) = \sum_{e \in T} \lambda_e(\nu)$. If T is a minimum sum spanning tree of G_ν, then $D(\nu)$ denotes the sum of the edge weights of T. The following lemma is straightforward.

Lemma 1. *The perturbed G has a MMST with weight ν if G_ν has a minimum sum spanning tree, T, with total sum weight $D(\nu) \le B$. Furthermore, T is a MMST of G with weight ν.*

According to Lemma 1, one can find a minimum sum spanning tree T of G_ν and change the weight of each edge $e \in T$ with $\underline{w}_e \le \nu < w_e$ to ν. Therefore, T is the improved MMST with $w(T) = \nu$ and the total cost is equal to $D(\nu)$.

We have the following fact.

Lemma 2. *$D(\nu)$ is a non-increasing function of ν.*

Proof. If $\nu_1 < \nu_2$, then $\lambda_e(\nu_1) \geq \lambda_e(\nu_2)$ for each $e \in E$, since f_e is non-decreasing. This implies that $D(\nu)$ is non-increasing. $\square$

Let $\mathcal{R} = \{w_e | \ e \in E\} \cap \mathcal{V}$ and $\mathcal{S} = \{\underline{w}_e | \ e \in E\} \cap \mathcal{V}$. Define $\mathcal{P} = \mathcal{R} \cup \mathcal{S} \cup \{\underline{\nu}, \bar{\nu}\}$. Assume that the elements of $\mathcal{P}$ are sorted increasingly, i.e., $\underline{\nu} = p_1 < p_2 < \cdots < p_r = \bar{\nu}$. We find an interval $[p_k, p_{k+1}]$ with $D(p_k) > B$ and $D(p_{k+1}) \leq B$ by applying binary search algorithm. By Lemma 2, the optimal solution of (1), ν^*, belongs to $(p_k, p_{k+1}]$. If $D(p_{k+1}) = B$, then p_{k+1} is the optimal value. Otherwise, ν^* is in (p_k, p_{k+1}) with $D(\nu^*) = B$. Since this interval does not contain any element of $\mathcal{P}$, each edge weight $\lambda_e(\nu)$ is not piecewise. More precisely, for each e in optimal MMST, $\lambda_e(\nu)$ is identical to either $f_e(w_e - \nu)$ or 0 on (p_k, p_{k+1}).

When ν changes in (p_k, p_{k+1}), a sequence of minimum spanning trees are constructed. In [31,32], the problem of finding all such intermediate minimum spanning trees is solved when the edge weight functions are all linear. In our problem, edge weight functions are decreasing, however, with a similar method in linear case one can find all minimum spanning trees when ν varies in (p_k, p_{k+1}). Here, we discuss the main idea of the solution method. Suppose T is a minimum spanning tree corresponding to the given $\nu_0 \in (p_k, p_{k+1})$. Let $e \notin T$. When ν is increased from its current value, ν_0, the weight of each edge is decreased, since $\lambda_g = f_g(w_e - \nu)$ and for all edge g, f_g is non-decreasing by assumption. Now, e enters T only if its weight becomes less than at least one of edges, say $h \in T$, in the cycle induced by adding e to T. That is, h replaced by e only if $f_e(w_e - \nu) = f_h(w_h - \nu)$. Therefore, the new spanning tree could be found by inspecting between all intersection points of weight functions.

Our main goal is to find $\nu \in (p_k, p_{k+1})$ with $D(\nu) = B$. Thus, it is not necessary to compute all intermediate spanning trees. Indeed, we can choose all intersection points belonged to (p_k, p_{k+1}), sort them increasingly, and use binary search algorithm in order to find two points q_t and q_{t+1} with $D(q_t) > B$ and $D(q_{t+1}) \leq B$. Note that the minimum spanning tree does not change on interval $(q_t, q_{t+1}]$; this unique minimum spanning tree is the solution of the problem. It is left to calculate the minimum weight ν by solving the equation $D(\nu) = \sum_{e \in T} \lambda_e(\nu) = B$. Here we assume that this equation could be solved by calling function `solve()`. We also assume that finding the number of all intersection points of cost functions, s, needs $O(g(s))$ time. We have the following result.

Theorem 1. *The upgrading MMST problem under non-decreasing cost functions can be solved in* $O\left(g(s) + n \log m (\log s + \log n)\right)$ *time plus* $O(1)$ *call of function* `solve()`*.*

Proof. Computing $\bar{\nu}$ and $\underline{\nu}$ needs linear time. The elements of $\mathcal{P}$ can be sorted in $O(n \log n)$ time. Then, binary search algorithm is used for determining interval (p_k, p_{k+1}). Each iteration of binary search algorithm needs solving a MMST which can be solved by well-known Kruskal's algorithm in $O(n \log m)$. Thus, finding interval (p_k, p_{k+1}) needs $O(n \log m \log n)$. Similarly, finding interval $[q_t, q_{t+1}]$ needs $O(g(s) + n \log m \log s)$, where $O(g(s))$ is included for finding all intersections points of cost functions. Thus, the problem is solvable in $O\left(g(s) + n \log m(\log n + \log s)\right)$ time, plus $O(1)$ call of function `solve()`. $\square$

Corollary 1. *If the cost functions are linear, then upgrading MMST can be solved in* $O(n^2)$ *time.*

Proof. Since all cost functions are linear, then, $s = O(n^2)$ and each intersection point can be found in $O(1)$ time. In addition, equation $D(\nu) = B$ is linear and function `solve()` runs in $O(1)$. According to Theorem 1 and the fact that $n = O(m^2)$, we conclude that the problem can be solved in

$$\begin{aligned} &O\left(g(s) + n \log m(\log n + \log s)\right) + O(1) \text{ call of solve() function } = \\ &\quad O(n^2 + n \log m \log n^2 + n \log m \log n) + O(1) = O(n^2). \qquad \square \end{aligned}$$

Remark 1. There is another improvement network problem named downgrading MMST problem. The aim of this problem is increasing the edge weights under the given budget so that the weight of MMST problem with respect to new edge weights is maximized. We can solve this problem by a similar method mentioned in this paper with minor changes. We ignore details because of avoiding of duplication.

3. Upgrading MMST problem under sum-type Hamming cost function

In this section we assume that for each $e \in E$, the cost function $f_e(x_e)$ is of the form $c_e H(w_e, \tilde{w}_e)$ where c_e is positive scalar and $H(x, y)$ is the Hamming distance between x and y, i.e.,

$$H(x, y) := \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{cases} \tag{3}$$

Let $\underline{T}$ be the MMST with weight $\underline{\nu}$ with respect to weight vector $\underline{w}$. We assume that $\sum_{e \in \underline{T}} c_e H(w_e, \tilde{w}_e) > B$. Since, otherwise, $\underline{T}$ is upgraded MMST and nothing is left to solve. So, whole budget will be used. Therefore $\underline{\nu}$ is a lower bound for the optimal value of problem (1), ν^*, which can be found in linear time.

On the other hand, suppose that T' is the MMST with respect to the initial weight vector w. It is easy to see that the weight of T' is an upper bound for optimal solution of (1). However, a tighter upper bound can be found by improving the weight of T' as much as possible. Let $\bar{\nu}$ be the weight of the improved T' in the perturbed G. We can find $\bar{\nu}$ with a greedy approach as follows. First, notice that when the Hamming distance is used, we pay the whole cost to modify an edge weight without attention to the magnitude. So the best choice for new weights are the lower bounds. Let $\underline{w}_{max} = max\{\underline{w}_e | e \in T'\}$. It is clear that $\bar{\nu} \geq \underline{w}_{max}$. Thus, $\bar{\nu}$ belongs to $P = \{w_e | e \in T', w_e > \underline{w}_{max}\} \cup \{\underline{w}_{max}\}$. We first sort the elements of P in non-increasing order, i.e. $P = \{p_1, \ldots, p_k\}$ with $p_1 \geq p_2 \geq \ldots \geq p_k = \underline{w}_{max}$. For simplicity, we denote by c_k the constant in Hamming function cost corresponding to p_k. Now, if $c_1 \leq B$, then we reduce p_1 to its lower bound and modify the budget. We continue this way until c_i exceeds the remaining budget for some i. Therefore obtaining $\bar{\nu}$ needs $O\left(|E(T')| \log\left(|E(T')|\right)\right) = O\,(n \log n)$ time.

Again, we consider a problem closely related to problem (1). The aim is to answer the question of finding the MMST with weight $\nu \in \mathcal{V} = [\underline{\nu}, \bar{\nu}]$ under bound restrictions and budget constraint. Consider the graph G_ν introduced in section 2 where edge weights are defined as:

$$\lambda_e(\nu) := \begin{cases} 0 & \text{if } \nu \geq w_e \\ c_e & \text{if } \underline{w}_e \leq \nu < w_e \\ B+1 & \text{if } \nu < \underline{w}_e \end{cases} \tag{4}$$

and $D(\nu)$ is defined as before.

We have the following lemma.

Lemma 3. *The perturbed G has a MMST with weight ν if G_ν has a minimum spanning tree T of weight $D(\nu) \leq B$ and T is a MMST in G with weight ν.*

Similar to Lemma 2, $D(\nu)$ is a non-increasing function. We recall that again the edge weights does not change unless it reaches its lower bound. Therefore the new weights belong to the following set:

$$\mathcal{Q} = \left(\{w_e | e \in E\} \cup \{\underline{w}_e | e \in E\} \cup \{\underline{\nu}, \bar{\nu}\}\right) \cap [\underline{\nu}, \bar{\nu}].$$

Assume that the elements of $\mathcal{Q}$ are sorted in increasing order, i.e. $\mathcal{Q} = \{q_1, q_2, \ldots, q_s\}$ with $\underline{\nu} = q_1 < \ldots < q_s = \bar{\nu}$.

Now, q_k is the optimum value of the problem where k is the smallest index k with $D(q_k) \leq B$. It can be found by binary search. Therefore, we have the following result.

Theorem 2. *The upgrading MMST problem under sum-type Hamming cost function can be solved in $O\,(n \log m \log n)$ time.*

Proof. We need to solve a MMST to find $\underline{\nu}$; it needs linear time. Finding $\bar{\nu}$ needs sorting the elements of P, which runs in $O\,(n \log n)$ time. Sorting the elements of $\mathcal{Q}$ needs $O\,(n \log n)$ time. The last binary search runs in $O\,(n \log m \log n)$, since it calls Kruskal's algorithm at most $\log n$ times. Thus, problem can be solved in $O\,(n \log n) + O\,(n \log n) + O\,(n \log m \log n) = O\,(n \log m \log n)$. $\square$

4. Conclusion

In this paper the upgrading MMST problem has been investigated. It is proved that this problem could be transformed into parametric minimum sum spanning tree problem. It is shown that the upgrading MMST under linear and sum-type Hamming cost functions are solvable in polynomial time. Further research on the upgrading other versions of MMST problems (stochastic or constrained) on networks under various objective functions, seems to be promising.

References

[1] D. Paik, S. Sahni, Network upgrading problems, Networks 26 (1) (1995) 45–58.
[2] I. Demgensky, H.-C. Wirth, Cost flow improvement by upgrading costs and capacities, in: Operations Research Proceedings, Springer, 2001, pp. 42–48.
[3] D.R. Fulkerson, G.C. Harding, Maximizing the minimum source-sink path subject to a budget constraint, Math. Program. 13 (1) (1977) 116–118.
[4] S.E. Hambrusch, H.-Y. Tu, Edge weight reduction problems in directed acyclic graphs, J. Algorithms 24 (1) (1997) 66–93.
[5] C.A. Phillips, The network inhibition problem, in: Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, ACM, 1993, pp. 776–785.
[6] E. Gassner, Up- and downgrading the 1-center in a network, European J. Oper. Res. 198 (2) (2009) 370–377.
[7] E. Gassner, Up- and Downgrading the 1-Median in a Network, Tech. Rep., Graz University of Technology, 2007.
[8] E. Gassner, A game-theoretic approach for downgrading the 1-median in the plane with Manhattan metric, Ann. Oper. Res. 172 (1) (2009) 393.
[9] A.R. Sepasian, F. Rahbarnia, Upgrading p-median problem on a path, J. Math. Model. Alg. Oper. Res. 14 (2) (2015) 145–157.
[10] K.U. Drangmeister, S.O. Krurnke, M.V. Marathe, H. Noltemeier, S.S. Ravi, Modifying edges of a network to obtain short subgraphs, Theoret. Comput. Sci. 203 (1) (1998) 91–121.
[11] G.N. Frederickson, R. Solis-Oba, Increasing the weight of minimum spanning trees, J. Algorithms 33 (2) (1999) 244–266.
[12] S.O. Krumke, M.V. Marathe, H. Noltemeier, R. Ravi, S.S. Ravi, Approximation algorithms for certain network improvement problems, J. Comb. Optim. 2 (3) (1998) 257–288.

[13] S. Krumke, H. Noltemeier, M. Marathe, S. Ravi, R. Ravi, R. Sundaram, On Optimal Strategies for Upgrading Networks, Tech. Rep., Los Alamos National Lab., NM, United States, 1996.
[14] S.O. Krumke, H. Noltemeier, M.V. Marathe, S. Ravi, K.U. Drangmeister, Modifying networks to obtain low cost trees, in: International Workshop on Graph-Theoretic Concepts in Computer Science, Springer, 1996, pp. 293–307.
[15] S. Krumke, M. Marathe, H. Noltemeier, R. Ravi, S. Ravi, R. Sundaram, H. Wirth, Improving spanning trees by upgrading nodes, in: Automata, Languages and Programming, 1997, pp. 281–291.
[16] S.O. Krumke, H. Noltemeier, H.-C. Wirth, M.V. Marathe, R. Ravi, S. Ravi, R. Sundaram, Improving spanning trees by upgrading nodes, Theoret. Comput. Sci. 221 (1) (1999) 139–155.
[17] E. Alvarez-Miranda, M. Sinnl, Lagrangian and branch-and-cut approaches for upgrading spanning tree problems, Comput. Oper. Res. 83 (2017) 13–27.
[18] S. Guo, V. Leung, A distributed algorithm for min–max tree and max–min cut problems in communication networks, IEEE/ACM Trans. Netw. (TON) 18 (4) (2010) 1067–1076.
[19] D. Li, H. Du, X. Jia, Energy-efficient broadcast/multicast routing with min–max transmission power in wireless ad hoc networks, in: Advances in Wireless Ad Hoc and Sensor Networks, 2008, p. 301.
[20] H. Haanpää, A. Schumacher, P. Orponen, Distributed algorithms for lifetime maximization in sensor networks via min–max spanning subgraphs, Wirel. Netw. 16 (3) (2010) 875–887.
[21] I. Papadimitriou, L. Georgiadis, Energy-aware broadcast trees in wireless networks, Mob. Netw. Appl. 9 (6) (2004) 567–581.
[22] P. Li, S. Guo, J. Hu, R. Sarker, Lifetime optimization for reliable broadcast and multicast in wireless ad hoc networks, Wirel. Commun. Mob. Comput. 14 (2) (2014) 221–231.
[23] B. Nguyen, A. Spillner, B.C. Emerson, V. Moulton, Distinguishing between hot-spots and melting-pots of genetic diversity using haplotype connectivity, Algorithms Mol. Biol. 5 (1) (2010) 19.
[24] P.M. Camerini, The min–max spanning tree problem and some extensions, Inform. Process. Lett. 7 (1) (1978) 10–14.
[25] N. Ishii, T. Nishida, Stochastic bottleneck spanning tree problem, Networks 13 (3) (1983) 443–449.
[26] O. Berman, D. Einav, G. Handler, The constrained bottleneck problem in networks, Oper. Res. 38 (1) (1990) 178–181.
[27] A.P. Punnen, K. Nair, An improved algorithm for the constrained bottleneck spanning tree problem, INFORMS J. Comput. 8 (1) (1996) 41–44.
[28] P.J. Anderson, C.J. Ras, Minimum bottleneck spanning trees with degree bounds, Networks 68 (4) (2016) 302–314.
[29] L. Liu, E. Yao, Inverse min–max spanning tree problem under the weighted sum-type Hamming distance, Theoret. Comput. Sci. 396 (3) (2008) 28–34.
[30] L. Liu, Q. Wang, Constrained inverse min–max spanning tree problems under the weighted Hamming distance, J. Global Optim. 43 (1) (2009) 83–95.
[31] P.K. Agarwal, D. Eppstein, L.J. Guibas, M.R. Henzinger, Parametric and kinetic minimum spanning trees, in: Proceedings of the 39th Annual Symposium on Foundations of Computer Science, 1998, IEEE, 1998, pp. 596–605.
[32] D. Fernández-Baca, G. Slutzki, D. Eppstein, Using sparsification for parametric minimum spanning tree problems, Nordic J. Comput. 3 (4) (1996) 352–366.