



On the approximability of robust spanning tree problems

Adam Kasperski^a, Paweł Zieliński^{b,*}

^a Institute of Industrial, Engineering and Management, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland

^b Institute of Mathematics and Computer Science, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland

ARTICLE INFO

Article history:

Received 22 June 2010

Accepted 3 October 2010

Communicated by: J. Díaz

Keywords:

Combinatorial optimization

Approximation

Robust optimization

Two-stage optimization

Computational complexity

ABSTRACT

In this paper the minimum spanning tree problem with uncertain edge costs is discussed. In order to model the uncertainty a discrete scenario set is specified and a robust framework is adopted to choose a solution. The min–max, min–max regret and 2-stage min–max versions of the problem are discussed. The complexity and approximability of all these problems are explored. It is proved that the min–max and min–max regret versions with nonnegative edge costs are hard to approximate within $O(\log^{1-\epsilon} n)$ for any $\epsilon > 0$ unless the problems in NP have quasi-polynomial time algorithms. Similarly, the 2-stage min–max problem cannot be approximated within $O(\log n)$ unless the problems in NP have quasi-polynomial time algorithms. In this paper randomized LP-based approximation algorithms with performance bound of $O(\log^2 n)$ for min–max and 2-stage min–max problems are also proposed.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

The usual assumption in combinatorial optimization is that all input parameters are precisely known. However, in real life this is rarely the case. There are two popular optimization settings of problems for hedging against uncertainty of parameters: *stochastic optimization setting* and *robust optimization setting*.

In the stochastic optimization, the uncertainty is modeled by specifying probability distributions of the parameters and the goal is to optimize the expected value of a solution built (see, e.g., [7,22]). One of the most popular models of the stochastic optimization is a *2-stage model* [7]. In the 2-stage approach the precise values of the parameters are specified in the first stage, while the values of these parameters in the second stage are uncertain and are specified by probability distributions. The goal is to choose a part of a solution in the first stage and complete it in the second stage so that the expected value of the obtained solution is optimized. Recently, there has been a growing interest in combinatorial optimization problems formulated in the 2-stage stochastic framework [9,10,12,16,21].

In the robust optimization setting [17] the uncertainty is modeled by specifying a set of all possible realizations of the parameters called *scenarios*. No probability distribution in the scenario set is given. In the *discrete scenario case*, which is considered in this paper, we define a scenario set by explicitly listing all scenarios. Then, in order to choose a solution, two optimization criteria, called the *min–max* and the *min–max regret*, can be adopted. Under the min–max criterion, we seek a solution that minimizes the largest cost over all scenarios. Under the min–max regret criterion we wish to find a solution that minimizes the largest deviation from optimum over all scenarios. A deeper discussion on both criteria can be found in [17]. The min–max (regret) versions of some basic combinatorial optimization problems with discrete structure of uncertainty have been extensively studied in the recent literature [2,3,14,19]. Furthermore, both robust criteria can be easily extended to the 2-stage framework. Such an extension has been recently done in [8,16].

* Corresponding author. Tel.: +48 71 320 30 67.

E-mail addresses: adam.kasperski@pwr.wroc.pl (A. Kasperski), pawel.zielinski@pwr.wroc.pl (P. Zieliński).

In this paper, we wish to investigate the min–max (regret) and min–max 2-stage versions of the classical *minimum spanning tree* problem. The classical deterministic problem is formally stated as follows. We are given a connected graph $G = (V, E)$ with edge costs c_e , $e \in E$. We seek a *spanning tree* of G of the minimal total cost. We use Φ to denote the set of all spanning trees of G . The classical deterministic minimum spanning tree is a well studied problem, for which several very efficient algorithms exist (see, e.g., [1]).

In the robust framework, the edge costs are uncertain and the set of scenarios Γ is defined by explicitly listing all possible edge cost vectors. So, $\Gamma = \{S_1, \dots, S_K\}$ is finite and contains exactly K scenarios, where a *scenario* is a cost realization $S = (c_e^S)_{e \in E}$. In this paper we consider the *unbounded case*, where the number of scenarios is a part of the input. We will denote by $C^*(S) = \min_{T \in \Phi} \sum_{e \in T} c_e^S$ the cost of a minimum spanning tree under a fixed scenario $S \in \Gamma$. In the MIN–MAX SPANNING TREE problem, we seek a spanning tree that minimizes the largest cost over all scenarios, that is

$$OPT_1 = \min_{T \in \Phi} \max_{S \in \Gamma} \sum_{e \in T} c_e^S. \quad (1)$$

In the MIN–MAX REGRET SPANNING TREE, we wish to find a spanning tree that minimizes the maximal regret:

$$OPT_2 = \min_{T \in \Phi} \max_{S \in \Gamma} \left\{ \sum_{e \in T} c_e^S - C^*(S) \right\}. \quad (2)$$

The formulation (1) is a single-stage decision one. We can extend this formulation to a 2-stage case as follows. We are given the first stage edge costs c_e , $e \in E$, and in the second stage there are K possible cost realizations (scenarios) listed in scenario set Γ . The 2-STAGE SPANNING TREE problem consists in determining a subset of edges E_1 in the first stage and a subset of edges E_2^S that augments it to form a spanning tree $T^S = E_1 \cup E_2^S \in \Phi$ under scenario S in the second stage for each scenario $S \in \Gamma$. The goal is minimize the maximum cost of the determined subsets of edges $E_1, E_2^{S_1}, \dots, E_2^{S_K}$:

$$OPT_3 = \min_{E_1, E_2^{S_1}, \dots, E_2^{S_K}} \max_{S \in \Gamma} \left\{ \sum_{e \in E_1} c_e + \sum_{e \in E_2^S} c_e^S : T^S = E_1 \cup E_2^S \in \Phi \right\}. \quad (3)$$

Let us now recall some known results on the problems under consideration. In the bounded case (when the number of scenarios is bounded by a constant), the MIN–MAX (REGRET) SPANNING TREE problem is NP-hard even if Γ contains only 2 scenarios [17] and admits an FPTAS [3], whose running time, however, grows exponentially with K . In the unbounded case, the MIN–MAX (REGRET) SPANNING TREE problem is strongly NP-hard [2,17] and not approximable within $(2-\epsilon)$, for any $\epsilon > 0$, unless $P = NP$ even for edge series-parallel graphs [14]. The MIN–MAX (REGRET) SPANNING TREE problem is approximable within K [3]. However, up to now the existence of an approximation algorithm with a constant performance ratio for the unbounded case has been an open question. To the best of the authors' knowledge the 2-stage version of the minimum spanning tree problem seems to exist only in the stochastic setting [9,10,12]. Recently, the robust 2-stage framework has been employed in [8,16] for some network design and matching problems.

Our results. In this paper we prove that the MIN–MAX SPANNING TREE and MIN–MAX REGRET SPANNING TREE problems are hard to approximate with a constant performance ratio (Theorem 3 and Corollary 1). Namely, they are not approximable within $O(\log^{1-\epsilon} n)$ for any $\epsilon > 0$, where n is the input size, unless $NP \subseteq DTIME(n^{\text{poly} \log n})$. We thus give a negative answer to the open question about the existence of approximation algorithms with a constant performance ratio for these problems. Moreover, if both positive and negative edge costs are allowed, then the MIN–MAX SPANNING TREE problem is not at all approximable unless $P = NP$ (Theorem 4). For the 2-STAGE SPANNING TREE problem, we show that it is not approximable within any constant, unless $P = NP$, and within $(1 - \epsilon) \ln n$ for any $\epsilon > 0$, unless $NP \subseteq DTIME(n^{\log \log n})$ (Theorem 6). The above negative results encourage us to find randomized approximation algorithms that yield a $O(\log^2 n)$ approximation ratio for MIN–MAX SPANNING TREE (Theorem 5) and 2-STAGE MIN–MAX SPANNING TREE (Theorem 7).

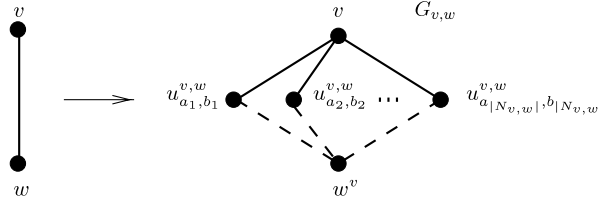
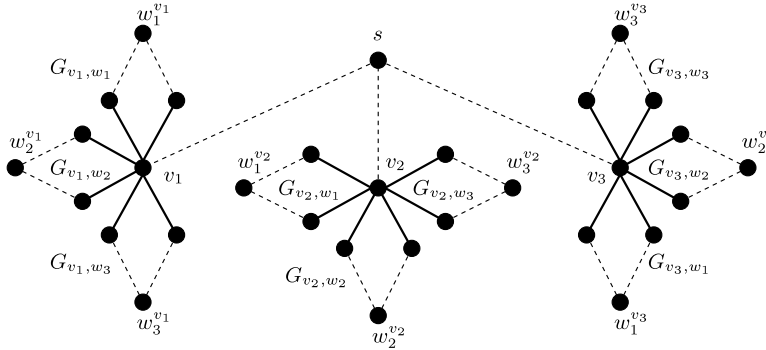
2. Min–max (regret) spanning tree

In this section, we study the MIN–MAX SPANNING TREE and MIN–MAX REGRET SPANNING TREE problems. We improve the results obtained in [2,14], by showing that both problems are hard to approximate within a ratio of $O(\log^{1-\epsilon} n)$ for any $\epsilon > 0$, unless the problems in NP have quasi-polynomial time algorithms. We then provide an LP-based randomized algorithm with approximation ratio of $O(\log^2 n)$ for MIN–MAX SPANNING TREE.

2.1. Hardness of approximation

We reduce a variant of the LABEL COVER problem (see e.g., [5,19]) to MIN–MAX SPANNING TREE.

LABEL COVER: *Input:* A regular bipartite graph $G = (V, W, E)$, $E \subseteq V \times W$; an integer N that defines the set of labels, which are integers in $\{1, \dots, N\}$; for every edge $(v, w) \in E$ a partial function $\sigma_{v,w} : \{1, \dots, N\} \rightarrow \{1, \dots, N\}$. A *labeling*

Fig. 1. Replacing edge $(v, w) \in E$ with component $G_{v,w}$.Fig. 2. A sample of graph G' , where graph G in \mathcal{L} is $K_{3,3}$.

of the instance $\mathcal{L} = (G, N, \{\sigma_{v,w}\}_{(v,w) \in E})$ is a function l assigning a nonempty set of labels to each vertex in $V \cup W$, namely $l : V \cup W \rightarrow 2^N$. A labeling satisfies an edge $(v, w) \in E$ if

$$\exists a \in l(v), \exists b \in l(w) : \sigma_{v,w}(a) = b.$$

A total labeling is a labeling that satisfies all edges. The value of a total labeling l is $\max_{x \in V \cup W} |l(x)|$.

Output: A total labeling of the minimum value. This value is denoted by $val(\mathcal{L})$.

We now recall the following theorem [5,19]:

Theorem 1. There exists a constant $\gamma > 0$ so that for any language $L \in NP$, any input \mathbf{w} and $N > 0$, one can construct an instance \mathcal{L} of LABEL COVER, with $|\mathbf{w}|^{O(\log N)}$ vertices and the label set of size N , so that:

$$\begin{aligned} \mathbf{w} \in L &\Rightarrow val(\mathcal{L}) = 1, \\ \mathbf{w} \notin L &\Rightarrow val(\mathcal{L}) \leq N^{-\gamma}. \end{aligned}$$

Furthermore, \mathcal{L} can be constructed in time polynomial in its size.

We now state and prove a theorem which is essential in showing the hardness results for the problems of interest.

Theorem 2. There exists a constant $\gamma > 0$ so that for any language $L \in NP$, any input \mathbf{w} , any $N > 0$ and any $g \leq N^\gamma$, one can construct an instance \mathcal{T} of MIN-MAX SPANNING TREE in time $O(|\mathbf{w}|^{O(g \log N)} N^{O(g)})$, so that:

$$\begin{aligned} \mathbf{w} \in L &\Rightarrow OPT_1(\mathcal{T}) \leq 1, \\ \mathbf{w} \notin L &\Rightarrow OPT_1(\mathcal{T}) \geq g. \end{aligned}$$

Proof. Let L be a language in NP and let $\mathcal{L} = (G = (V, W, E), N, \{\sigma_{v,w}\}_{(v,w) \in E})$ be the instance of LABEL COVER from Theorem 1 constructed for L . Let us introduce some additional notations:

- $\delta(x)$ is the set of edges of G incident to vertex $x \in V \cup W$,
- $N_{v,w} = \{(a, b) \in N \times N : \sigma_{v,w}(a) = b\}$.

We now transform \mathcal{L} to an instance \mathcal{T} of MIN-MAX SPANNING TREE. Let us fix $g \leq N^\gamma$, where γ is the constant from Theorem 1. We first construct graph G' in the following way. We replace every edge $(v, w) \in E$ with paths $(v, u_{a,b}^{v,w}, w^v)$ for all $(a, b) \in N_{v,w}$ (see Fig. 1). The edges of the form $(u_{a,b}^{v,w}, w^v)$ (the dashed edges) are called *dummy edges* and the edges of the form $(v, u_{a,b}^{v,w})$ (the solid edges) are called *label edges*. We say that label edge $(v, u_{a,b}^{v,w})$ assigns label a to v and label b to w . We will denote the obtained component by $G_{v,w}$ and we will use $E_{v,w}^l$ to denote the set of all label edges of $G_{v,w}$, obviously $|E_{v,w}^l| = |N_{v,w}|$. We finish the construction of G' by an adding additional vertex s and connecting all the components by additional dummy edges (s, v) for all $v \in V$. A sample graph G' , where G is $K_{3,3}$, is shown in Fig. 2.

We now form scenario set Γ . We first note that all dummy edges under all scenarios have costs equal to 0. We say that two label edges are *label-distinct* if they do not assign the same label to any vertex v or w . Namely, $(v, u_{a_i, b_i}^{v, w})$ and $(v', u_{a'_i, b'_i}^{v', w'})$ are label-distinct if $a_i = a'_i$ implies $v \neq v'$ and $b_i = b'_i$ implies $w \neq w'$. Consider vertex $v \in V$, for which there is the set of $p = |\delta(v)|$ components $\mathcal{G} = \{G_{v, w_1}, \dots, G_{v, w_p}\}$. For every subset $\mathcal{F} \subseteq \mathcal{G}$ of exactly g components, $\mathcal{F} = \{G_{v, w_1}, \dots, G_{v, w_g}\}$ and for every g -tuple of pairwise label-distinct edges $((v, u_{a_1, b_1}^{v, w_1}), \dots, (v, u_{a_g, b_g}^{v, w_g})) \in E_{v, w_1}^l \times \dots \times E_{v, w_g}^l$ we form a scenario under which all these edges have cost 1 and all the remaining edges have cost 0. We repeat this procedure for all vertices $v \in V$. Consider then vertex $w \in W$, for which there is the set of $q = |\delta(w)|$ components $\mathcal{G} = \{G_{v_1, w}, \dots, G_{v_q, w}\}$. For every subset $\mathcal{F} \subseteq \mathcal{G}$ of exactly g components, $\mathcal{F} = \{G_{v_1, w}, \dots, G_{v_g, w}\}$ and for every g -tuple of pairwise label-distinct edges $((v_1, u_{a_1, b_1}^{v_1, w}), \dots, (v_g, u_{a_g, b_g}^{v_g, w})) \in E_{v_1, w}^l \times \dots \times E_{v_g, w}^l$ we form a scenario under which all these edges have cost 1 and all the remaining edges have cost 0. We repeat this for all vertices $w \in W$. In order to ensure $\Gamma \neq \emptyset$, we add to Γ the scenario in which every edge has zero cost.

Assume that $\mathbf{w} \in L$ and thus $\text{val}(\mathcal{L}) = 1$. Thus, there exists a total labeling l satisfying all edges in G such that $\max_{x \in V \cup W} |l(x)| = 1$. Each edge $(v_i, w_i) \in E$ in G corresponds to exactly one component G_{v_i, w_i} in G' . Let (a_i, b_i) be the pair of labels satisfying the edge (v_i, w_i) in total labeling l , i.e. $a_i \in l(v_i)$ and $b_i \in l(w_i)$. We form a spanning tree T in G' by adding exactly one edge $(v_i, u_{a_i, b_i}^{v_i, w_i})$ from every component G_{v_i, w_i} and we complete the construction by adding a necessary number of dummy edges. Since the labeling l is such that $\max_{x \in V \cup W} |l(x)| = 1$, no pair of label-distinct edges have been chosen while constructing T , so $\sum_{e \in T} c_e^S \leq 1$ for all $S \in \Gamma$ and consequently $\max_{S \in \Gamma} \sum_{e \in T} c_e^S \leq 1$.

Assume that $\mathbf{w} \notin L$ and thus $\max_{x \in V \cup W} |l(x)| \geq N^\gamma \geq g$ for all total labellings l . Consider any spanning tree T in G' . Without loss of generality, we can assume that T contains exactly one label edge from every component $G_{v, w}$. The set of all label edges contained in T corresponds to a total labeling l of \mathcal{L} . Since $|l(x)| \geq g$, for some vertex $x \in V \cup W$, we have to use at least g distinct labels in the labeling l . Suppose that $x = v \in V$ and we use distinct labels a_1, \dots, a_g for v . Then, T contains pairwise label-distinct edges $(v, u_{a_i, b_i}^{v, w_i})$, $i = 1, \dots, g$, and $\sum_{e \in T} c_e^S = g$ under scenario S that correspond to this g -tuple of edges. The reasoning for $x = w \in W$ is the same. In consequence $\max_{S \in \Gamma} \sum_{e \in T} c_e^S = g$ and $\text{OPT}_1(\mathcal{T}) = g$.

Let us now examine the size of the resulting instance of the MIN-MAX SPANNING TREE problem. The size of the set of edges E' is at most $|V| + 2|E|N^2$, the size of the set of vertices V' is at most $1 + |V| + |E|N^2 + |W||V|$ and the number of scenarios is at most $1 + 2|E|^g N^g N^g$. Hence, and from $|E| = |\mathbf{w}|^{O(\log N)}$, we deduce that the size of the constructed instance (G', Γ) is $|\mathbf{w}|^{O(g \log N)} N^{O(g)}$, so it can be constructed in $O(|\mathbf{w}|^{O(g \log N)} N^{O(g)})$ time. \square

From Theorem 2, we obtain the following result:

Theorem 3. *The MIN-MAX SPANNING TREE problem with nonnegative edge costs under all scenarios is not approximable within $O(\log^{1-\epsilon} n)$ for any $\epsilon > 0$, where n is the input size, unless $\text{NP} \subseteq \text{DTIME}(n^{\text{poly log } n})$.*

Proof. Let γ be the constant from Theorem 2. For any $\beta > 0$ we fix $g = \log^\beta |\mathbf{w}|$ and $N = \log^{O(\beta)} |\mathbf{w}|$, so that inequality $g \leq N^\gamma$ is satisfied for the constant γ (see Theorem 2). The input size of the resulting instance (G', Γ) from Theorem 2 is $n = |\mathbf{w}|^{O(g \log N)} N^{O(g)} = |\mathbf{w}|^{O(\log^{\beta+\delta} |\mathbf{w}|)}$ for some constant $\delta > 0$, so it can be constructed in $O(|\mathbf{w}|^{\text{poly log } |\mathbf{w}|})$ time. Since $g = \log^\beta |\mathbf{w}|$ and $n = 2^{O(\log^{\beta+\delta+1} |\mathbf{w}|)}$, we get $g = O(\log^{\frac{\beta}{\beta+\delta+1}} n)$ and the gap is $O(\log^{1-\epsilon} n)$ for any $\epsilon > 0$. \square

Corollary 1. *The MIN-MAX REGRET SPANNING TREE problem is not approximable within $O(\log^{1-\epsilon} n)$ for any $\epsilon > 0$, where n is the input size, unless $\text{NP} \subseteq \text{DTIME}(n^{\text{poly log } n})$.*

Proof. The corollary follows easily if we assume that each component $G_{v, w}$ in the construction from Theorem 2 has at least 2 label edges or, equivalently, every edge in the instance of LABEL COVER has at least two pairs of labels. In this case, under every scenario $S \in \Gamma$, there is a spanning tree of 0 cost (recall that we never assign two 1's to the same component in S). Hence $\text{OPT}_1(\mathcal{T}) = \text{OPT}_2(\mathcal{T})$ and the proof is completed. If some edge in the instance of LABEL COVER has only one pair of labels, then this pair trivially forces an assignment of labels to two vertices, which (after checking consistency with other edges) can be removed from the instance before applying the construction from Theorem 2. \square

Up to this point we have assumed that the edge costs under all scenarios are nonnegative. The following theorem demonstrates that violation of this assumption makes the MIN-MAX SPANNING TREE problem not at all approximable:

Theorem 4. *If both positive and negative costs are allowed, then the MIN-MAX SPANNING TREE problem is not at all approximable unless $\text{P} = \text{NP}$ even for edge series-parallel graphs.*

Proof. We show a gap-introducing reduction from 3-SAT [13]. We will assume that in an instance of 3-SAT for every variable x_k , $k = 1, \dots, n$, both x_k and $\sim x_k$ appear in collection $C = \{C_1, \dots, C_m\}$ of clauses. Obviously, under such assumption 3-SAT remains strongly NP-complete. Given an instance of 3-SAT we construct an instance of MIN-MAX SPANNING TREE as follows. For each clause $C_i = (l_1^i \vee l_2^i \vee l_3^i)$ we create a graph G_i composed of 5 vertices: $s_i, v_1^i, v_2^i, v_3^i, t_i$ and 6 edges: the edges (s_i, v_1^i) , (s_i, v_2^i) , (s_i, v_3^i) correspond to literals in C_i , the edges (v_1^i, t_i) , (v_2^i, t_i) , (v_3^i, t_i) have costs equal to -1 under every scenario. In order to construct a connected graph $G = (V, E)$ with $|V| = 4m + 1$, $|E| = 6m$, we identify vertex t_i of G_i with vertex s_{i+1} of G_{i+1} for $i = 1, \dots, m-1$. Note that the resulting graph G is edge series-parallel. Finally, we form scenario set Γ as follows. For every pair of edges of G , (s_i, v_j^i) and (s_q, v_r^q) , that correspond to contradictory literals l_j^i and l_r^q , i.e. $l_j^i = \sim l_r^q$, we create

scenario S such that under this scenario the costs of the edges (s_i, v_j^i) and (s_q, v_r^q) are set to $4m - 1$ and the costs of all the remaining edges are set to -1 . It is easy to verify that each spanning tree T in the constructed instance has nonnegative maximal cost over all scenarios.

Suppose that 3-SAT is satisfiable. Then there exists a spanning tree T of G containing exactly $4m$ edges that do not correspond to contradictory literals. Thus, under every scenario S , the tree contains at most one edge with the cost $4m - 1$ and all the remaining $4m - 1$ edges have costs equal to -1 . In consequence we get $\sum_{e \in T} c_e^S = 0$ under every $S \in \Gamma$ and $OPT_1 = 0$. If 3-SAT is unsatisfiable, then every spanning trees T of G contains at least two edges which correspond to contradictory literals, and so $OPT_1 = \max_{S \in \Gamma} \sum_{e \in T} c_e^S \geq 4m$. Consequently MIN-MAX SPANNING TREE is not approximable, unless $P = NP$. Otherwise, any polynomial time approximation algorithm applied to the constructed instance could decide if an instance of 3-SAT is satisfiable. \square

2.2. Randomized algorithm for min-max spanning tree

If the edge costs are nonnegative under all scenarios, then the MIN-MAX SPANNING TREE problem is approximable within K , K is the number of scenarios, and this is the best approximation ratio known so far [3]. On the other hand the problem is not at all approximable if negative costs are allowed (Theorem 4). In this section, we assume that all costs are nonnegative and we give a polynomial time approximation algorithm for the problem which returns an $O(\log^2 n)$ -approximate spanning tree, where n is the number of vertices of G . The algorithm is based on a randomized rounding of a solution to an iterative linear program.

It is easy to check that binary solutions to the following program $LP_{\min \max}(C)$ are in one-to-one correspondence with solutions to MIN-MAX SPANNING TREE of edge costs in every scenario at most C :

$$LP_{\min \max}(C): \sum_{e \in E} c_e^S x_e \leq C \quad \forall S \in \Gamma, \quad (4)$$

$$\sum_{e \in E} x_e = n - 1, \quad (5)$$

$$\sum_{e \in \delta(W)} x_e \geq 1 \quad \forall W \subset V, \quad (6)$$

$$0 \leq x_e \leq 1 \quad \forall e \in E, \quad (7)$$

$$\text{if } c_e^S > C \text{ then } x_e = 0 \quad \forall e \in E \text{ and } \forall S \in \Gamma, \quad (8)$$

where $\delta(W)$ denotes the cut determined by vertex set W , i.e. $\delta(W) = \{(i, j) \in E : i \in W, j \in V \setminus W\}$. The core of $LP_{\min \max}(C)$ (constraints (5)–(7)) is the relaxation of the *cut-set formulation* for spanning tree [18]. The polynomial time solvability of $LP_{\min \max}(C)$ follows from an efficient polynomial time separation based on the min-cut problem (see [18]). Solving $LP_{\min \max}(C)$ consists in rejecting all edges $e \in E$ having $c_e^S > C$ under some scenario $S \in \Gamma$ and solving then the resulting linear programming problem. Using binary search in $[0, (n - 1)c_{\max}]$, where $c_{\max} = \max_{e \in E} \max_{S \in \Gamma} c_e^S$, one can find the minimal value of parameter C , for which there is a feasible solution to $LP_{\min \max}(C)$. Let \hat{C} be this minimal value and let $(\hat{x}_e)_{e \in E}$ be a feasible solution to $LP_{\min \max}(\hat{C})$. Clearly $\hat{C} \leq OPT_1$. Furthermore, if $\hat{x}_e > 0$, then $c_e^S \leq \hat{C}$ and thus $c_e^S \leq OPT_1$ for each scenario $S \in \Gamma$.

We now give an algorithm that randomly rounds a feasible solution of $LP_{\min \max}(\hat{C})$ to an $O(\log^2 n)$ -approximate min-max spanning tree (see Algorithm 1).

Algorithm 1: Randomized algorithm for MIN-MAX SPANNING TREE

Use binary search in $[0, (n - 1)c_{\max}]$ to find the minimal value of C such that there exists a feasible solution to $LP_{\min \max}(C)$, i.e., \hat{C} and $(\hat{x}_e)_{e \in E}$.

Initially \hat{F} contains only vertices of G , that is n components.

$r \leftarrow \left\lceil 2 \left(11 + \sqrt{21} \right) \ln n \right\rceil$

for $k \leftarrow 1$ **to** r **do**

 For all $e \in E$, add edge e independently with probability \hat{x}_e to \hat{F} .

if \hat{F} is connected **then**

exit for-loop

if \hat{F} is connected **then**

return a spanning tree of \hat{F}

Let us analyze Algorithm 1. Obviously the algorithm is polynomial. The following lemma shows that the total cost of edges included in each iteration under any scenario $S \in \Gamma$ is $O(\ln n)OPT_1$ with probability at least $1 - \frac{1}{n}$:

Lemma 1. Let \hat{E}_k be a set of edges added to \hat{F} at iteration k of Algorithm 1 and let $K \leq n^{\rho_2}$, $1 \leq f \leq n^{\rho_3}$, where $f, \rho_1, \rho_2, \rho_3$ are nonnegative constants such that $\rho_2 + \rho_3 \leq ((e - \frac{1}{2})^2 - 1)\rho_1 \approx 3.92 \cdot \rho_1$, $\rho_1 \geq 2$. Then

$$\max_{S \in \Gamma} \sum_{e \in \hat{E}_k} c_e^S \leq (\rho_1 \ln n + 1.5) \left(1 + 2\sqrt{1 + \frac{\ln K + \ln f}{\rho_1 \ln n}} \right) OPT_1 \quad (9)$$

holds with probability at least $1 - \frac{1}{fn^{\rho_1-1}}$.

Proof. See Appendix.

We now analyze the feasibility of an output solution \hat{F} . Let \hat{F}_k be the forest obtained from \hat{F}_{k-1} after the k -th iteration. Initially, $\hat{F}_0, \hat{F}_0 \subset G$, has no edges. Let C_k denote the number of connected components of \hat{F}_k . Obviously, $C_0 = n$. We say that an iteration k is “successful” if either $C_{k-1} = 1$ (\hat{F}_{k-1} is connected) or $C_k < 0.9C_{k-1}$; otherwise, it is “failure”. We now recall a result of Alon [4]. His proof is repeated in Appendix for completeness.

Lemma 2 (Alon [4]). For every k , the conditional probability that iteration k is “successful”, given any set of components in \hat{F}_{k-1} , is at least $1/2$.

From Lemma 2, it follows that the probability of the event that iteration k is “successful” is at least $1/2$. This is a lower bound on the probability of success of given any history. Note that, if forest \hat{F}_k is not connected ($C_k > 1$) then the number of “successful” iterations has been less than $-\log_{0.9} n < 10 \ln n$. Let X be a random variable denoting the number of “successful” iterations among r performed iterations of the algorithm. The probability $\Pr[X < 10 \ln n]$ can be upper bounded by $\Pr[Y < 10 \ln n]$, where $Y = \sum_{k=1}^r Y_k$ is the sum of r independent Bernoulli trials such that $\Pr[Y_k = 1] = 1/2$. This estimation can be done, since we have a lower bound on success of given any history. Clearly, $\mathbb{E}[Y] = r/2$. We apply the Chernoff bound (see for instance [20]) and determine the values of $\delta \in (0, 1]$ and r in order to fulfill the following inequality:

$$\Pr[X < 10 \ln n] \leq \Pr[Y < 10 \ln n] = \Pr[Y < (1 - \delta)\mathbb{E}[Y]] < e^{-\mathbb{E}[Y]\delta^2/2} = \frac{1}{n}. \quad (10)$$

It is easily seen that inequality (10) holds if the following system of equations

$$\begin{cases} (1 - \delta)r/2 = 10 \ln n, \\ r\delta^2/4 = \ln n \end{cases} \quad (11)$$

holds true. An easy computation for δ and r in (11), shows that $r = 2(11 + \sqrt{21}) \ln n$, $\delta = \sqrt{\frac{2}{11 + \sqrt{21}}}$. Hence, after r iterations, $r = \lceil 2(11 + \sqrt{21}) \ln n \rceil$, we obtain with probability at least $1 - 1/n$ a spanning tree. By the union bound and Lemma 1 (set $f = r$), with probability at least $1 - 1/n$ in every iteration, $k = 1, \dots, r$, the set of edges \hat{E}_k included at iteration k satisfies the bound (9). We conclude that after r iterations, we get with probability at least $1 - 2/n$ a spanning tree whose total cost in every scenario is $O(r \ln n) OPT_1$. We have, thus proved the following theorem:

Theorem 5. There is a polynomial time randomized algorithm for MIN-MAX SPANNING TREE that returns with probability at least $1 - \frac{2}{n}$ a solution whose total cost in every scenario is $O(\log^2 n) OPT_1$.

3. 2-stage spanning tree

In this section, we discuss the 2-STAGE SPANNING TREE problem in a robust optimization setting. We show that the problem is hard to approximate within a ratio of $O(\log n)$ unless the problems in NP have quasi-polynomial algorithms. Then, we give an LP-based randomized approximation algorithm with ratio of $O(\log^2 n)$.

3.1. Hardness of approximation

Theorem 6. The 2-STAGE SPANNING TREE problem is not approximable within any constant, unless $P = NP$, and within $(1 - \epsilon) \ln n$ for any $\epsilon > 0$, unless $NP \subseteq DTIME(n^{\log \log n})$.

Proof. We proceed with a cost preserving reduction from SET COVER to 2-STAGE SPANNING TREE. The reduction is similar to that in [12] for the 2-stage stochastic spanning tree. SET COVER is defined as follows (see, e.g., [5, 13]):

SET COVER: Input: A ground set $\mathcal{U} = \{1, \dots, n\}$ and a collection of its subsets U_1, \dots, U_m such that $\bigcup_{i=1}^m U_i = \mathcal{U}$.

A subcollection $I \subseteq \{1, \dots, m\}$ covers \mathcal{U} if $\bigcup_{i \in I} U_i = \mathcal{U}$, where $|I|$ is the size of the subcollection.

Output: A minimum sized subcollection that covers \mathcal{U} .

The SET COVER problem is not approximable within any constant, unless $P = NP$, and within $(1 - \epsilon) \log n$ for any $\epsilon > 0$, unless $NP \subseteq DTIME(n^{\log \log n})$, where n is the size of the ground set (see [6,11]). For a given instance $\mathcal{C} = (\mathcal{U}, U_1, \dots, U_m)$ of SET COVER, we construct an instance $\mathcal{T} = (G = (V, E), \Gamma)$ of 2-STAGE SPANNING TREE as follows. Graph $G = (V, E)$ is a complete graph with $m + n + 1$ vertices $V = \{u_1, \dots, u_m, 1, \dots, n, r\}$. Vertices u_1, \dots, u_m correspond to m subsets U_1, \dots, U_m , vertices $1, \dots, n$ correspond to n elements of set \mathcal{U} . The costs of the edges (r, u_i) , $i = 1, \dots, m$, in G in the first stage are set to 1 and the costs of all the remaining edges in G are set to $m + 1$. Now we form scenario set Γ in the second stage. Each scenario $S_j \in \Gamma$ corresponds to vertex j , $j = 1, \dots, n$. Let $T_j = \{j\} \cup \{u_i : j \in U_i\}$ and let $(T_j, V \setminus T_j)$ be the cut separating T_j from all other vertices of G . Each second stage scenario S_j is defined as: the costs of the edges from cut $(T_j, V \setminus T_j)$ are set to $m + 1$ and the costs of the remaining edges in G are set to 0.

We now prove that there is a subcollection of size at most $k \leq m$ that covers \mathcal{U} if and only if there exists a spanning tree in G of the maximum 2-stage cost at most $k \leq m$. Given a subcollection U_{i_1}, \dots, U_{i_k} of size k that covers \mathcal{U} . In the first stage, we include in E_1 the edges (r, u_{i_j}) , where vertices u_{i_j} correspond to subsets U_{i_j} , $j = 1, \dots, k$. The cost of E_1 is equal to k . In the second stage, we augment E_1 to form a spanning tree with edges of cost zero in each scenario S_j , $j = 1, \dots, n$. Hence, the maximum 2-stage cost of the obtained spanning tree equals k . Conversely, let T be a spanning tree in G with the maximum 2-stage cost at most k . Hence, this tree does not contain any edge with cost $m + 1$. Consequently, in the first stage the tree contains $k' \leq k$ edges of the form (r, u_{i_j}) , $j = 1, \dots, k'$, and in the second stage in each scenario it contains zero cost edges. The vertices u_{i_j} correspond to subsets U_{i_j} , $j = 1, \dots, k'$. It is easily seen that any element $i \in \mathcal{U}$ must be covered by at least one of subsets U_{i_j} , $j = 1, \dots, k'$. Otherwise the solution would contain an edge of cost $m + 1$. Thus, U_{i_j} , $j = 1, \dots, k'$, form a subcollection of the size at most k that covers \mathcal{U} .

The presented reduction is cost preserving. Hence, 2-STAGE SPANNING TREE has the same approximation bounds as SET COVER. \square

3.2. Randomized algorithm for 2-stage spanning tree

In this section we construct a randomized approximation algorithm for 2-STAGE SPANNING TREE, which is based on a similar idea as the corresponding algorithm for MIN-MAX SPANNING TREE (see Section 2.2). Consider the following program $LP_{2stage}(C)$, whose binary solutions correspond to the solutions of 2-STAGE SPANNING TREE:

$$\begin{aligned}
 LP_{2stage}(C): \quad & \sum_{e \in E} c_e x_e + \sum_{e \in E} c_e^S x_e^S \leq C \quad \forall S \in \Gamma \\
 & \sum_{e \in E} (x_e + x_e^S) = n - 1 \quad \forall S \in \Gamma \\
 & \sum_{e \in \delta(W)} (x_e + x_e^S) \geq 1 \quad \forall W \subset V, \forall S \in \Gamma \\
 & 0 \leq x_e, x_e^S \leq 1 \quad \forall e \in E, \forall S \in \Gamma \\
 & \text{if } c_e > C \text{ then } x_e = 0 \quad \forall e \in E \\
 & \text{if } c_e^S > C \text{ then } x_e^S = 0 \quad \forall e \in E, \forall S \in \Gamma.
 \end{aligned}$$

The algorithm (Algorithm 2) randomly rounds a feasible solution \hat{x}_e, \hat{x}_e^S , $S \in \Gamma$, $e \in E$, of $LP_{2stage}(\hat{C})$, where \hat{C} denotes the minimal value of C for which there is a feasible solution to $LP_{2stage}(C)$.

Algorithm 2: Randomized algorithm for 2-STAGE SPANNING TREE

```

 $c_{\max} \leftarrow \max_{e \in E} \{c_e, \max_{S \in \Gamma} c_e^S\}$ 
Use binary search in  $[0, (n - 1)c_{\max}]$  to find the minimal value of  $C$  such that there exists a feasible solution of
 $LP_{2stage}(C)$ , i.e.,  $\hat{x}_e, \hat{x}_e^S$ ,  $S \in \Gamma$ ,  $e \in E$ .
Initially  $\hat{F}^S$  contains only vertices of  $G$  for  $S \in \Gamma$ .
 $r \leftarrow \lceil (\sqrt{\ln n + \ln K} + \sqrt{21 \ln n + \ln K})^2 \rceil$ 
for  $k \leftarrow 1$  to  $r$  do
    In the first stage: For all  $e \in E$ , choose edge  $e$  independently with probability  $\hat{x}_e$  and add it to each  $\hat{F}^S$  for  $S \in \Gamma$ .
    In the second stage: for every  $S \in \Gamma$  and every  $e \in E$ , add edge  $e$  independently with probability  $\hat{x}_e^S$  to  $\hat{F}^S$ .
if all  $\hat{F}^S$ ,  $S \in \Gamma$ , are connected then
    return  $\{\hat{F}^S\}_{S \in \Gamma}$ 

```

An analysis of Algorithm 2 proceeds similarly to the one of Algorithm 1. The following lemma holds (the proof goes in similar manner to the proof of Lemma 1):

Lemma 3. Let \hat{E}_k and \hat{E}_k^S be the sets of edges in the first stage and in the second stage for every $S \in \Gamma$, respectively, added to \hat{F}^S at iteration k of Algorithm 2 and let $K \leq n^{\rho_2}$, $1 \leq f \leq n^{\rho_3}$, where $f, \rho_1, \rho_2, \rho_3$ are nonnegative constants such that $\rho_2 + \rho_3 \leq ((e - \frac{1}{2})^2 - 1)\rho_1 \approx 3.92 \cdot \rho_1$, $\rho_1 \geq 2$. Then

$$\sum_{e \in \hat{E}_k} c_e + \sum_{e \in \hat{E}_k^S} c_e^S \leq (\rho_1 \ln n + 1.5) \left(1 + 2\sqrt{1 + \frac{\ln K + \ln f}{\rho_1 \ln n}} \right) OPT_3 \quad \forall S \in \Gamma \quad (12)$$

holds with probability at least $1 - \frac{1}{fn^{\rho_1-1}}$.

Let \hat{F}_k^S be the forest for $S \in \Gamma$ after the k -th iteration of Algorithm 2, Let C_k^S denote the number of connected components of \hat{F}_k^S . Again, we say that an iteration k is “successful” if either $C_{k-1}^S = 1$ or $C_k^S < 0.9C_{k-1}^S$; otherwise it is “failure”. The probability of the event that iteration k is “successful” is at least $1/2$, which is due to Lemma 2.

Consider any scenario $S \in \Gamma$. If forest \hat{F}_k^S is not connected then the number of “successful” iterations is less than $-\log_{0.9} n < 10 \ln n$. We estimate $\Pr[X < 10 \ln n]$ by $\Pr[Y < 10 \ln n]$, where X is random variable denoting the number of “successful” iterations among r iterations and $Y = \sum_{k=1}^r Y_k$ is the sum of r independent Bernoulli trials such that $\Pr[Y_k = 1] = 1/2$, $\mathbf{E}[Y] = r/2$. We use the Chernoff bound and compute the values of $\delta \in (0, 1]$ and r satisfying the following inequality:

$$\Pr[X < 10 \ln n] \leq \Pr[Y < 10 \ln n] = \Pr[Y < (1 - \delta)\mathbf{E}[Y]] < e^{-\mathbf{E}[Y]\delta^2/2} = \frac{1}{nK}. \quad (13)$$

This gives $r = (\sqrt{\ln n + \ln K} + \sqrt{21 \ln n + \ln K})^2$ and $\delta = \frac{2\sqrt{\ln n + \ln K}}{\sqrt{\ln n + \ln K} + \sqrt{21 \ln n + \ln K}}$. Recall that K is the number of scenarios. By the union bound, the probability that a forest in at least one scenario S is not connected is less than $1/n$. Again, by the union bound and Lemma 3 (set $f = r$), with probability at least $1 - 1/n$ in every k iteration, $k = 1, \dots, r$, the sets of edges \hat{E}_k and \hat{E}_k^S for each $S \in \Gamma$, included at iteration k , satisfy the bound (12). Thus, after r iterations, $r = \lceil (\sqrt{\ln n + \ln K} + \sqrt{21 \ln n + \ln K})^2 \rceil$, with probability at least $1 - 2/n$, we obtain spanning trees of cost $O(r \ln n)OPT_3$ in every scenario. We get the following theorem:

Theorem 7. There is a polynomial time randomized algorithm for 2-STAGE MINIMUM SPANNING TREE that returns with probability at least $1 - \frac{2}{n}$ a spanning tree whose cost in every scenario is $O(\log^2 n)OPT_3$.

4. Conslusions

In this paper we have investigated three different robust models of the minimum spanning tree problem with discrete unbounded scenario set. All of them are strongly NP-hard and hard to approximate within a constant factor. There are still some open questions concerning these problems. The MIN-MAX SPANNING TREE problem with nonnegative costs is not approximable within $O(\log^{1-\epsilon} n)$ but it admits a randomized $O(\log^2 n)$ -approximation algorithm. So, the existence of an $O(\log n)$ -approximation algorithm (deterministic or randomized) for this problem is still an open question. A similar situation holds for the 2-STAGE SPANNING TREE with min-max criterion. For MIN-MAX REGRET SPANNING TREE we only have negative results. It is not easy to modify the randomized algorithms designed in this paper for the min-max regret criterion. So, the construction of better approximation algorithms for the min-max regret version of the minimum spanning tree is an interesting subject for further research.

Acknowledgement

This work was partially supported by Polish Committee for Scientific Research, grant N N206 492938.

Appendix. Some proofs

Proof of Lemma 1. In order to prove the bound (9), we will apply a technique used in [16,15]. Consider any scenario $S \in \Gamma$. Let us sort the costs in S in nonincreasing order $c_{e[1]}^S \geq c_{e[2]}^S \geq \dots \geq c_{e[m]}^S$, (m is the number of edges of G). We partition the ordered set of edges E into groups as follows. The first group $G^{(1)}$ consists of edges $e[1], \dots, e[j^{(1)}]$, where $j^{(1)}$ is the maximum such that $\hat{x}_{e[1]} + \dots + \hat{x}_{e[j^{(1)}]} \leq \rho_1 \ln n$. The subsequent groups $G^{(l)}$, $l = 2, \dots, t$, are defined in the same way, that is $G^{(l)}$ consists of edges $e[j^{(l-1)} + 1], \dots, e[j^{(l)}]$, where $j^{(l)}$ is the maximum such that $\hat{x}_{e[j^{(l-1)} + 1]} + \dots + \hat{x}_{e[j^{(l)}]} \leq \rho_1 \ln n$. The optimal value OPT_1 satisfies:

$$OPT_1 \geq \hat{C} \geq \sum_{i=1}^m c_{e[i]}^S \hat{x}_{e[i]} \geq \sum_{l=1}^t \left[\left(\min_{e \in G^{(l)}} c_e^S \right) \sum_{e \in G^{(l)}} \hat{x}_e \right] \geq (\rho_1 \ln n - 1) \sum_{l=1}^{t-1} \min_{e \in G^{(l)}} c_e^S. \quad (14)$$

Let X_e be a binary random variable with $\Pr[X_e = 1] = \hat{x}_e$. It holds that

$$\begin{aligned} \sum_{e \in \hat{E}_k} c_e^S &\leq \sum_{l=1}^t \sum_{e \in G^{(l)}} c_e^S X_e \leq \sum_{l=1}^t \sum_{e \in G^{(l)}} \left(\max_{e \in G^{(l)}} c_e^S \right) X_e \\ &\leq \left(\max_{e \in G^{(1)}} c_e^S \right) \sum_{e \in G^{(1)}} X_e + \sum_{l=2}^t \left[\left(\min_{e \in G^{(l-1)}} c_e^S \right) \sum_{e \in G^{(l)}} X_e \right]. \end{aligned} \quad (15)$$

Let us recall a Chernoff bound (see e.g., [20]). Suppose X_1, \dots, X_N are independent Poisson trials such that $\Pr[X_i = 1] = p_i$. Let $X = \sum_{i=1}^N X_i$. Then the inequality holds: $\Pr[X > \mathbf{E}[X](1 + \delta)] < e^{-\mathbf{E}[X]\delta^2/4}$ for any $\delta \leq 2e - 1$. We use this Chernoff bound to estimate $\sum_{e \in G^{(l)}} X_e$ in each group $G^{(l)}$. Consider a group $G^{(l)}$. It holds that $\mathbf{E}[\sum_{e \in G^{(l)}} X_e] = \sum_{e \in G^{(l)}} \hat{x}_e \leq \rho_1 \ln n$. Set $\delta = 2\sqrt{(\rho_1 \ln n + \ln K + \ln f)/(\rho_1 \ln n)}$. Since $K \leq n^{\rho_2}$, $1 \leq f \leq n^{\rho_3}$ and $\rho_2 + \rho_3 \leq ((e - 1/2)^2 - 1)\rho_1$, $\rho_1 \geq 2$, inequality $\delta \leq 2e - 1$ holds. Thus the Chernoff bound yields:

$$\Pr \left[\sum_{e \in G^{(l)}} X_e > \rho_1 \ln n (1 + \delta) \right] < e^{-(\rho_1 \ln n + \ln K + \ln f)} = 1/(fKn^{\rho_1}). \quad (16)$$

By the union bound, the probability that $\sum_{e \in G^{(l)}} X_e > \rho_1 \ln n (1 + \delta)$ holds for at least one group $G^{(l)}$ is less than $1/(fKn^{\rho_1-1})$ (because the number of groups is at most n). Now applying the bound $\sum_{e \in G^{(l)}} X_e \leq \rho_1 \ln n (1 + \delta)$ for every $l = 1, \dots, t$ to (15) and using the fact that $\max_{e \in G^{(1)}} c_e^S \leq OPT_1$ and inequality (14) we obtain:

$$\sum_{e \in \hat{E}_k} c_e^S \leq \rho_1 \ln n \left(1 + 2\sqrt{\frac{\rho_1 \ln n + \ln K + \ln f}{\rho_1 \ln n}} \right) \left(OPT_1 + \frac{OPT_1}{\rho_1 \ln n - 1} \right).$$

The fact that $\rho_1 \ln n / (\rho_1 \ln n - 1) \leq 3/2$, $n \geq 5$, and an easy computation show that: $\sum_{e \in \hat{E}_k} c_e^S \leq (\rho_1 \ln n + 1.5) \left(1 + 2\sqrt{1 + \frac{\ln K + \ln f}{\rho_1 \ln n}} \right) OPT_1$. The probability that the bound fails for a given scenario S is less than $1/(fKn^{\rho_1-1})$ so, by the union bound, the probability that it fails for at least one scenario $S \in \Gamma$ is less than $1/(fn^{\rho_1-1})$. \square

Proof of Lemma 2. If \hat{F}_{k-1} is connected then we are done. Otherwise, let us denote by $H = (V_H, E_H)$ the graph obtained from \hat{F}_{k-1} by contracting all its connected components to a single vertex. An edge e is not included in \hat{F}_k with probability $1 - \hat{x}_e$. Hence, the probability that any vertex v of H remains isolated is

$$\prod_{e \in \delta(v)} (1 - \hat{x}_e) \leq \exp \left(- \sum_{e \in \delta(v)} (1 - \hat{x}_e) \right) \leq 1/e,$$

where $\delta(v)$ denotes the set of edges incident to v . The last inequality follows from the fact that $\sum_{e \in \delta(v)} (1 - \hat{x}_e) \geq 1$. By linearity of expectation, the expected number of isolated vertices of H is $|V_H|/e$, and thus with the probability at least $1/2$ the number of isolated vertices is at most $2|V_H|/e$. Hence, the number of connected components of \hat{F}_k is at most

$$\frac{2|V_H|}{e} + \frac{1}{2} \left(|V_H| - \frac{2|V_H|}{e} \right) = \left(\frac{1}{2} + \frac{1}{e} \right) |V_H| < 0.9|V_H|.$$

Since $|V_H| = C_{k-1}$, the lemma follows. \square

References

- [1] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.
- [2] H. Aissi, C. Bazgan, D. Vanderpooten, Approximation of min-max and min-max regret versions of some combinatorial optimization problems, *European Journal of Operational Research* 179 (2007) 281–290.
- [3] H. Aissi, C. Bazgan, D. Vanderpooten, Min-max and min-max regret versions of combinatorial optimization problems: a survey, *European Journal of Operational Research* 197 (2009) 427–438.
- [4] N. Alon, A note on network reliability, in: D. Aldous, P. Diaconis, J. Spencer, J.M. Steele (Eds.), *Discrete Probability and Algorithms*, in: IMA Volumes in Mathematics and its Applications, vol. 72, Springer-Verlag, 1995, pp. 11–14.
- [5] S. Arora, C. Lund, *Hardness of approximations*, in: D. Hochbaum (Ed.), *Algorithms for NP-Hard Problems*, PWS, 1995.
- [6] M. Bellare, O. Goldreich, M. Sudan, Free bits, PCs and non-approximability – towards tight results, *SIAM Journal on Computing* 27 (1998) 804–915.
- [7] J.R. Birge, F. Louveaux, *Introduction to Stochastic Programming*, Springer-Verlag, 1997.
- [8] K. Dhamdhere, V. Goyal, R. Ravi, Pay today for a rainy day: improved approximation algorithms for demand-robust min-cut and shortest path problems, in: STACS 2006, in: *Lecture Notes in Computer Science*, vol. 3884, Springer-Verlag, 2006, pp. 206–217.
- [9] K. Dhamdhere, R. Ravi, M. Singh, On two-stage stochastic minimum spanning trees, in: M. Jünger, V. Kaibel (Eds.), *IPCO 2005*, in: *Lecture Notes in Computer Science*, vol. 3509, Springer-Verlag, 2005, pp. 321–334.
- [10] B. Escoffier, L. Gourves, J. Monnot, O. Spanjaard, Two-stage stochastic matching and spanning tree problems: polynomial instances and approximation, *European Journal of Operational Research* 205 (2010) 19–30.
- [11] U. Feige, A threshold of $\ln n$ for approximating set cover, *Journal of the ACM* 45 (1998) 634–652.
- [12] A.D. Flaxman, A.M. Frieze, M. Krivelevich, On the random 2-stage minimum spanning tree, *Random Structures and Algorithms* 28 (2006) 24–36.

- [13] M.R. Garey, D.S. Johnson, *Computers and Intractability. A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, 1979.
- [14] A. Kasperski, P. Zieliński, On the approximability of minmax (regret) network optimization problems, *Information Processing Letters* 109 (2009) 262–266.
- [15] A. Kasperski, P. Zieliński, A randomized algorithm for the min–max selecting items problem with uncertain weights, *Annals of Operations Research* 172 (2009) 221–230.
- [16] I. Katriel, C. Kenyon-Mathieu, E. Upfal, Commitment under uncertainty: two-stage matching problems, *Theoretical Computer Science* 408 (2008) 213–223.
- [17] P. Kouvelis, G. Yu, *Robust Discrete Optimization and its Applications*, Kluwer Academic Publishers, 1997.
- [18] T.L. Magnanti, L.A. Wolsey, Optimal trees, in: M.O. Ball, T.L. Magnanti, C.L. Monma, G.L. Nemhauser (Eds.), *Network Models*, in: *Handbook in Operations Research and Management Science*, vol. 7, North-Holland, Amsterdam, 1995, pp. 503–615.
- [19] M. Mastrolilli, N. Mutsanas, O. Svensson, Approximating single machine scheduling with scenarios, in: *APPROX-RANDOM 2008*, in: *Lecture Notes in Computer Science*, vol. 5171, Springer-Verlag, 2008, pp. 153–164.
- [20] R. Motwani, P. Raghavan, *Randomized Algorithms*, Cambridge University Press, 1995.
- [21] R. Ravi, A. Sinha, Hedging uncertainty: approximation algorithms for stochastic optimization problems, *Mathematical Programming* 108 (2006) 97–114.
- [22] M.H. van der Vlerk, *Stochastic programming bibliography*. <http://mally.eco.rug.nl/spbib.html>, 1996–2007.