Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Operations Research Letters 35 (2007) 17 – 22
Operations Research Letters
www.elsevier.com/locate/orl

The robust minimum spanning tree problem: Compact and convex uncertainty

Martha Salazar-Neumann

Service de Mathématiques de la Gestion, Université Libre de Bruxelles, Boulevard du Triomphe - CP 210/01, B-1050 Bruxelles, Belgium

Received 27 May 2003; accepted 29 December 2005
Available online 9 February 2006

Abstract

We consider the robust minimum spanning tree problem where edges costs are on a compact and convex subset of $\mathbb{R}^n$. We give the location of the robust deviation scenarios for a tree and characterizations of strictly strong edges and non-weak edges leading to recognition algorithms.

Keywords: Minimax regret optimization; Robust spanning tree; Uncertainty modeling

1. Introduction

This paper considers the robust minimum spanning tree problem with uncertainty in input data. In the approach given in [4], uncertainty is introduced by defining edge costs as interval numbers where each edge cost can take any value in its interval. The set of scenarios S is all potentially realizable input data over the values of edges of the graph. In this case the set of input data is a Cartesian product of intervals.

In this paper we consider a problem more general than this, the robust minimum spanning tree problem when the set D of input data is a compact and convex subset of $\mathbb{R}^n$. The approach used here to deal with this robust combinatorial problem is the worst case analysis, that consists on looking for a solution that performs “reasonably well” for all scenarios. “Reasonably well” depends on which robustness concept is used. In this work we will use the criteria given by Kouvelis and Yu [3].

In [3] authors prove that the absolute robust and the deviation robust minimum spanning tree problems are NP-hard for bounded number of scenarios and strongly NP-hard for unbounded number of scenarios.

The complexity of absolute robust tree problem with interval data is polynomial [4] and the robust deviation tree problem with interval data is NP-complete [1].

In this work we give the location of the worst and best deviation scenarios for a tree. In [4] authors give a mixed integer programming formulation for the robust deviation spanning tree problem with interval data and it is shown that characterizations of strong edges and non-weak edges can be very useful for pre-processing this mixed integer programming.

E-mail address: msalazar@ulb.ac.be.

0167-6377/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.orl.2005.12.007

In this work we give characterizations of strictly strong and non-weak edges and we show how to find such edges when the set of data D is a compact and convex subset of $\mathbb{R}^n$. Such characterizations are based on the topology of the graph combined with the structure of the data set.

These results has been developed with the goal to find new algorithms that reduce the time to compute a robust spanning tree when edges costs are on a compact and convex subset of $\mathbb{R}^n$.

2. Notation and definitions

Let G be a connected graph, we denote by $V(G)$ and $E(G)$ the set of vertices and edges of G, respectively. We suppose that $|V(G)| = m$, $|E(G)| = n$. Let S be the set of scenarios for the values of the edges of G and let D be the set of our input data. We will assume that D is a compact and convex subset of $\mathbb{R}^n$.

We use the notation $c^s = (c_1^s, \ldots, c_n^s) \in D$ to denote the data instance that corresponds to scenario s, where c_i^s is the cost of edge i in that scenario.

We denote by $T(G)$ the set of spanning trees of G. For a $T \in T(G)$ we denote by $E(T)$ the set of edges of T. To evaluate $T \in T(G)$ under the scenario s we use a function $f : T(G) \times D \to \mathbb{R} : (T, c^s) \to f(T, c^s)$. In this paper we assume that this function is the sum of the values of the edges of T under the scenario s.

$$f(T, c^s) = \sum_{i \in E(T)} c_i^s.$$

An optimal single scenario tree T_s^* for the scenario s satisfies:

$$f(T_s^*, c^s) = \min_{T \in T(G)} f(T, c^s).$$

We denote $T_s^*(G)$ the set of optimal single scenario trees of G for the scenario s. For a scenario $s \in S$ we denote by $s(k,t)$ the scenario for which $(c_1^{s(k,t)}, \ldots, c_k^{s(k,t)}, \ldots, c_n^{s(k,t)}) = (c_1^s, \ldots, c_k^s + t, \ldots c_n^s)$. The following definitions can be found in the book [3] .

Absolute robustness: The absolute robust tree T_a is such that

$$\max_{s \in S} f(T_a, c^s) = \min_{T \in T(G)} \max_{s \in S} f(T, c^s).$$

Robust deviation: The robust deviation tree T_d is such that

$$\max_{s \in S} (f(T_d, c^s) - f(T_s^*, c^s)) = \min_{T \in T(G)} \max_{s \in S} (f(T, c^s) - f(T_s^*, c^s)).$$

An absolute worst scenario for a tree T is a scenario in which $f(T, c^s)$, the cost of this spanning tree is maximum. A worst deviation scenario for a tree T is a scenario in which $f(T, c^s) - f(T_s^*, c^s)$, the difference between the cost of the spanning tree T and the cost of a minimum spanning tree in this scenario is maximum. The following definitions can be found in [4].

Definition 2.1. A spanning tree is a weak tree if it is a minimum spanning tree for some scenario.

Definition 2.2. An edge is a weak edge if it lies on some weak tree.

Definition 2.3. An edge is a strong edge if it lies on a minimum spanning tree for all scenario.

Now we will introduce the following definition:

Definition 2.4. An edge is a strictly strong edge if it lies on all minimum spanning tree for all scenario.

About the location of the absolute worst scenario of a fixed tree T we have the following remark:

Remark 1. If T is a tree of G and D is a compact and convex subset of $\mathbb{R}^n$, then for each tree the absolute best and worst scenarios are on the border of D and the worst scenario is a solution of the problem,

$$\max_{s \in S} \sum_{i \in E(T)} c_i^s$$

then if D is a polyhedron, we can find the worst scenario for a tree in polynomial time.

Now we define the functions

$$g : T(G) \times D \to \mathbb{R} : (T, c^s) \to g(T, c^s),$$

$$g(T, c^s) = \max_{T' \in T(G)} (f(T, c^s) - f(T', c^s)) = f(T, c^s) - f(T_s^*, c^s).$$

Let $T \in T(G)$, we will call the value $\max_{s\in S} f(T,c^s) - f(T_s^*,c^s)$ the maximum regret for the tree T. If $\tilde{T} \in T(G)$ is such that

$$\max_{s\in S} f(\tilde{T},c^s) - f(T_s^*,c^s) = \min_{T\in T(G)} \max_{s\in S} f(T,c^s) - f(T_s^*,c^s)$$

then we call $\tilde{T}$ a robust deviation tree.

We observe that for a fixed tree $T \in T(G)$ the function g becomes the function

$$g_T : D \to \mathbb{R} : c^s \to g_T(c^s),$$

$$g_T(c^s) = f(T,c^s) - f(T_s^*,c^s).$$

3. The worst deviation scenarios for a tree

In order to give the location of the worst and best deviation scenarios of a tree (Theorem 3.3), we will first study the properties of the functions $c^s \to f(T_s^*,c^s)$ and g_T.

Theorem 3.1. *If T_s^* denote an optimal single scenario tree of G for the scenario s then the functions $c^s \to f(T_s^*,c^s)$ and g_T are continuous and piece-wise linear over D.*

Proof. The function $c^s \to f(T_s^*,c^s)$ is such that;

$$f(T_s^*,c^s) = \min_{T\in T(G)} f(T,c^s)$$

as the number of spanning trees of G is finite, this function is the minimum of a finite number of linear functions and then is continuous. For a fixed T the function $c^s \to f(T,c^s)$ is linear then g_T is the difference of a linear function and a piece-wise linear function. So g_T is continuous and piece-wise linear. □

Now we will study the differentiability of $c^s \to f(T_s^*,c^s)$ and g_T but first the following lemma.

Lemma 3.1. *Let $t>0$, and let $c^s \in D$ such that $c^{s(k,-t)} \in D$. If $T_0 \in T_s^*(G)$ and $k \in E(T_0)$, then $T_0 \in T^*_{s(k,-t)}(G)$. Moreover $k \in E(T)$ for all $T \in T^*_{s(k,-t)}(G)$.*

Proof. We have $f(T_0,c^{s(k,-t)}) = f(T_0,c^s) - t \leqslant f(T,c^s) - t \leqslant f(T,c^{s_1})$ for all $T \in T(G)$. This implies that $T_0 \in T^*_{s(k,-t)}(G)$. If there exists $T_1 \in T^*_{s(k,-t)}(G)$ such that $k \notin E(T_1)$, then

$$f(T_1,c^{s(k,-t)}) = f(T_1,c^s) \geqslant f(T_0,c^s) = f(T_0,c^{s(k,-t)}) + t.$$

Therefore $T_1 \notin T^*_{s(k,-t)}(G)$ and we have a contradiction. □

Now we define the subset Ω of D for which all vicinity $B(c^s,\delta)$ of c^s contains points $c^{s_1}, c^{s_2} \in B(c^s,\delta)$ such that $T^*_{s_1} \in T^*_{s_1}(G)$ but $T^*_{s_1}$ is not optimal on the scenario s_2 and $T^*_{s_2} \in T^*_{s_2}(G)$ but $T^*_{s_2}$ not optimal on the scenario s_1.

Definition 3.1. Let $\Omega = \bigcup_{k=1}^n \Omega(k)$ where

$$\Omega(k) = \{c^s \in D : \exists T_1, T_2 \in T_s^*(G) \text{ such that } \forall t>0 \text{ if } c^{s(k,t)} \in D,\ T_1 \in T^*_{s(k,t)}(G) \text{ but } T_2 \notin T^*_{s(k,t)}(G) \text{ and if } c^{s(k,-t)} \in D,\ T_2 \in T^*_{s(k,-t)}(G) \text{ but } T_1 \notin T^*_{s(k,-t)}(G)\}.$$

Remark 2. By Lemma 3.1 $k \notin E(T_1)$ and $k \in E(T_2)$.

Remark 3. For each $k \in E(G)$ we can also define $\Omega(k)$ as follows:

$$\Omega(k) = \{c^s \in D : \exists\ T_1, T_2 \in T_s^*(G) \text{ and } j \in E(G): k \notin E(T_1),\ k \in E(T_2),\ j \in E(T_1),\ j \notin E(T_2) \text{ and } c_k^s = c_j^s\}.$$

Remark 4. If $\Omega = \emptyset$ then for all $c^s \in D$ there exists a neighborhood of c^s $B(c^s,t)$ such that for all $c^{s_0} \in B(c^s,t) \cap D$, $T_s^*(G) = T^*_{s_o}(G)$ and this implies that there exists a minimum spanning tree for all realizations of edges costs (a permanent tree for S [4]).

Theorem 3.2. *For all $T \in T(G)$ the functions $c^s \to f(T_s^*,c^s)$ and g_T are differentiable over $D\backslash\Omega$. Moreover for all $c^s \in \Omega$ both functions are not differentiable on c^s.*

Proof. If $c^s \in D\backslash\Omega$ there exists a neighborhood $B(c^s,t)$ of c^s such that for all $c^{s_o} \in B(c^s,t)\cap D$, $T_s^*(G) = T^*_{s_o}(G)$, then $c^s \to f(T_s^*,c^s)$ and g_T are differentiable over $B(c^s,t)\cap D$ and then over $D\backslash\Omega$.

In order to prove that for all $c^s \in \Omega$ $c^s \to f(T_s^*,c^s)$ is not differentiable on c^s we will prove that $\partial f(T_s^*,(c_1^s,\ldots,c_k^s,\ldots,c_n^s))/\partial c_k$ do not exist.

By definition

$$\frac{\partial f(T_s^*,(c_1^s,\ldots,c_k^s,\ldots,c_n^s))}{\partial c_k}=\lim_{h\to 0}\frac{f(T_{s(k,h)}^*,(c_1^s,\ldots,c_k^s+h,\ldots,c_n^s))-f(T_s^*,(c_1^s,\ldots,c_k^s,\ldots,c_n^s))}{h}.$$

Let $k\in I$ such that $\Omega(k)\neq\emptyset$ and let and T_2 as in Remark 3 then $T_2\in T_s^*(G)$ and $k\in E(T_2)$, if $h<0$ by Lemma 3.1 $T_2\in T_{s(k,h)}^*(G)$ and

$$=\lim_{h\to 0^-}\frac{f(T_2,(c_1^s,\ldots,c_k^s+h,\ldots,c_n^s))-f(T_2,(c_1^s,\ldots,c_k^s,\ldots,c_n^s))}{h}=\lim_{h\to 0^-}\frac{h}{h}=1$$

if $h>0$ as $T_1\in T_s^*(G)$ and $k\notin E(T_1)$ then

$$\lim_{h\to 0^+}\frac{f(T_{s(k,h)}^*,(c_1^s,\ldots,c_k^s+h,\ldots,c_n^s))-f(T_s^*,(c_1^s,\ldots,c_k^s,\ldots,c_n^s))}{h}$$
$$=\lim_{h\to 0^+}\frac{f(T_1,(c_1^s,\ldots,c_k^s+h,\ldots,c_n^s))-f(T_1,(c_1^s,\ldots,c_k^s,\ldots,c_n^s))}{h}=0$$

then $c^s\to f(T_s^*,c^s)$ and g_T are not differentiable over Ω. □

Lemma 3.2. *If D is convex then g_T is convex and $c^s\to f(T_s^*,c^s)$ is a concave function.*

Proof. Let $c^{s_1},c^{s_2}\in D$, as D is convex, for all $\lambda\in[0,1]$ if $c^{s_0}=\lambda c^{s_1}+(1-\lambda)c^{s_2}$ then $c^{s_0}\in D$ and then we have

$$\begin{aligned}g_T(c^{s_0})&=f(T,c^{s_0})-f(T_{s_0}^*,c^{s_0})\\&=\sum_{i\in E(T)}c_i^{s_0}-\sum_{i\in E(T_{s_0}^*)}c_i^{s_0}\\&=\sum_{i\in E(T)}(\lambda c_i^{s_1}+(1-\lambda)c_i^{s_2})\\&\quad-\sum_{i\in E(T_{s_0}^*)}(\lambda c_i^{s_1}+(1-\lambda)c_i^{s_2})\\&=\lambda\left(\sum_{i\in E(T)}c_i^{s_1}-\sum_{i\in E(T_{s_0}^*)}c_i^{s_1}\right)\\&\quad+(1-\lambda)\left(\sum_{i\in E(T)}c_i^{s_2}-\sum_{i\in E(T_{s_0}^*)}c_i^{s_2}\right)\\&=\lambda(f(T,c^{s_1})-f(T_{s_0}^*,c^{s_1}))\\&\quad+(1-\lambda)(f(T,c^{s_2})-f(T_{s_0}^*,c^{s_2}))\\&\leqslant\lambda(f(T,c^{s_1})-f(T_{s_1}^*,c^{s_1}))\\&\quad+(1-\lambda)(f(T,c^{s_2})-f(T_{s_2}^*,c^{s_2}))\\&=\lambda g_T(c^{s_1})+(1-\lambda)g_T(c^{s_2})\end{aligned}$$

then g_T is convex.

$$\begin{aligned}f(T_{s_o}^*,c^{s_0})&=\sum_{i\in E(T_{s_0}^*)}c_i^{s_0}\\&=\sum_{i\in E(T_{s_0}^*)}(\lambda c_i^{s_1}+(1-\lambda)c_i^{s_2})\\&=\sum_{i\in E(T_{s_0}^*)}\lambda c_i^{s_1}+\sum_{i\in E(T_{s_0}^*)}(1-\lambda)c_i^{s_2}\\&\geqslant\lambda f(T_{s_1}^*,c^{s_1})+(1-\lambda)f(T_{s_2}^*,c^{s_2})\end{aligned}$$

then $c^s\to f(T_s^*,c^s)$ is concave. □

The next theorem gives the location of the deviation worst and best scenarios for a tree when the set of scenarios is a compact and convex subset of $\mathbb{R}^n$.

Theorem 3.3. *Let T a tree of G and D a compact and convex subset of $\mathbb{R}^n$, then the data instances that corresponds to the deviation worst and best scenarios for T are on the boundary ∂D of D and on $\partial D\cup\Omega$, respectively.*

Proof. Easy, because g_T is a piece-wise linear and convex function defined in a compact and convex set. □

4. Strictly strong and non-weak edges

In this section we give first a partial classification of the edges of G, and then we give characterizations of the edges that are always on a minimum spanning tree (strictly strong edges) and the edges that are never on a minimum spanning tree (non-weak edge) of all realization of input data. Finally, we present algorithms to find the strictly strong edges and the non-weak edges. We will first give some definitions.

Definition 4.1. Let

$$B(k)=\{(c_1^s,\ldots,c_k^s,\ldots,c_n^s)\in D:\exists j\in E(G),\ j\neq k \text{ such that } c_k^s=c_j^s\},$$

$$A^-(k)=\{i\in E(G):\forall s\in S,\ c_i^s<c_k^s\},$$

$$C(k)=\{i\in E(G): i\neq k \text{ and } \exists s\in S \text{ such that } c_i^s=c_k^s\}.$$

Definition 4.2 (*Bondy and Murty [2]*). Let E' a nonempty subset of $E(G)$. The subgraph of G whose vertex set is the set of ends of edges in E' and whose edge set is E' is called the subgraph of G induced by E' and is denoted by $G[E']$.

Remark 5. For all $s,s'\in S$, $|E(T_s^*)\cap A^-(k)|=|E(T_{s'}^*)\cap A^-(k)|$ because for all $s\in S$, $G[E(T_s^*)\cap A^-(k)]$ is a spanning acyclic subgraph of $G[A^-(k)]$ and for all $s\in S$, $G[E(T_s^*)\cap A^-(k)]$ and $G[A^-(k)]$ are the same number of connected components.

Remark 6. If $\Omega(k)=\emptyset$ then k is strictly strong or k is non-weak, if $\Omega(k)\neq\emptyset$ then by definition k is a weak edge.

Theorem 4.1. *If $A^-(k)=\emptyset$ then k is weak.*

If $B(k)=\emptyset$ and $A^-(k)=\emptyset$ then k is strictly strong.

If $B(k)=\emptyset$ and there exists $s\in S$ such that $k\in E(T_s^*)$ then k is strictly strong.

If $B(k)=\emptyset$ and there exists $s\in S$ such that $k\notin E(T_s^*)$ then k is non-weak.

Proof. If $A^-(k)=\emptyset$ then there exists $s\in S$ such that when Kruskal's algorithm sorts the edges in nondecreasing order of their costs, the edge k is in the first place, then edge k is in a minimum spanning tree of G for the scenario $s\in S$ and then k is a weak edge. If $B(k)=\emptyset$ and $A^-(k)=\emptyset$, for all $s\in S$ the edge k is in the first place, then k is in all minimum spanning tree of G for all scenario $s\in S$, so k is a strictly strong edge.

If $B(k)=\emptyset$ and there exists $s\in S$ such that $k\in E(T_s^*)$ then adding edge k to $E(T_s^*)\cap A^-(k)$ at the point it was encountered would not have introduced a cycle, then k do not have two extremities in the same connected component of the subgraph of G induced by the edges in $E(T_s^*)\cap A^-(k)$. By Remark 5 for all $s'\in S$, k do not have two extremities in the same connected component of the subgraph of G induced by the edges in $E(T_{s'}^*)\cap A^-(k)$. As $B(k)=\emptyset$, for all $s'\in S$ Kruskal's algorithm add the edge k to $E(T_{s'}^*)\cap A^-(k)$, then k is a strictly strong edge.

If $B(k)=\emptyset$ and there exists $s\in S$ such that $k\notin E(T_s^*)$ then adding edge k to $E(T_s^*)\cap A^-(k)$ at the point it was encountered would have introduced a cycle, then k join two vertices in the same connected component of the subgraph of G induced by the edges of $E(T_s^*)\cap A^-(k)$. Then by Remark 5, for all $s'\in S$, adding k to $E(T_{s'}^*)\cap A^-(k)$ would have introduced a cycle, then k is non-weak. □

Theorem 4.2. *The edge k is strictly strong if and only if k is a cut edge of the subgraph of G induced by the edges in $C(k)\cup A^-(k)\cup\{k\}$.*

Proof. If k is a cut edge of $G[C(k)\cup A^-(k)\cup\{k\}]$, then for all $s\in S$, Kruskal's algorithm must select edge k to form a spanning tree of G and then for all $s\in S$ and for all $T_s^*\in T_s^*(G)$, $k\in T_s^*$, then k is a strictly strong edge. Now if k is not a cut edge of $G[C(k)\cup A^-(k)\cup\{k\}]$ then k is contained in a cycle of $G[C(k)\cup A^-(k)\cup\{k\}]$. By definition of $A^-(k)$ and $C(k)$ there exists a scenario $s'\in S$ such that $c_j^{s'}\leqslant c_k^{s'}$ for all $j\in C(k)\cup A^-(k)$. So there exists a minimum spanning tree $T_{s'}^*$ of G which does not contain the edge k and then k is not a strictly strong edge. □

Theorem 4.3. *The edge k is non-weak if and only if k is not a cut edge of the subgraph of G induced by the edges in $A^-(k)\cup\{k\}$.*

Proof. If k is non-weak for all $s\in S$ and for all $T_s^*\in T_s^*(G)$, $k\notin E(T_s^*)$. This implies that for all $s\in S$ adding edge k to $E(T_s^*)\cap A^-(k)$ at the point it was encountered would have introduced a cycle, so k is not a cut edge of $G[A^-(k)\cup\{k\}]$. Now if k is not a cut edge of $G[A^-(k)\cup\{k\}]$, k is contained in a cycle of $G[A^-(k)\cup\{k\}]$. By definition of $A^-(k)$ we know that for all $s\in S$ all minimum spanning acyclic subgraph of $G[A^-(k)\cup\{k\}]$ do not contain the edge k, so for all $s\in S$, $k\notin E(T_s^*)\cap A^-(k)$ and then k is non-weak. □

The last two theorems allow us to derive algorithms to find all the strictly strong and non-weak edges

Algorithm to find the strictly strong edges

1. For each $k \in E(G)$ obtain the set $A^-(k) = \{i \in E(G) : \forall s \in S,\ c_i^s < c_k^s\}$.
2. Obtain the set $C(k) = \{i \in E(G) : \exists s \in S$ such that $c_i^s = c_k^s\}$.
3. Construct the subgraph of G induced by the edges in $C(k) \cup A^-(k) \cup \{k\}$.
4. Verify if k is a cut edge of the subgraph $G[C(k) \cup A^-(k) \cup \{k\}]$.
5. If the answer is yes, k is a strictly strong edge, otherwise k is not strictly strong.

Algorithm to find the non-weak edges

1. For each $k \in E(G)$ obtain the set $A^-(k) = \{i \in E(G) : \forall s \in S,\ c_i^s < c_k^s\}$.
2. Construct the subgraph of G induced by the edges in $A^-(k) \cup \{k\}$.
3. Verify if k is not a cut edge of the subgraph $G[A^-(k) \cup \{k\}]$.
4. If the answer is yes, k is a non-weak edge, otherwise k is not non-weak.

When D is a bounded polyhedron $P=\{c^s \in \mathbb{R}^n : Ac^s \geqslant b, c^s \geqslant 0\}$, where A is an $m \times n$ matrix and b is a vector in $\mathbb{R}^m$, testing whether $i \in A^-(k)$ can be done by solving the linear program; $\max\{c_i^s - c_k^s : c^s \in D\}$. If the value of the solution of the program is negative $i \in A^-(k)$. Testing whether $i \in C(k)$ can be done by solving two linear programs; $\max\{c_i^s - c_k^s : c^s \in D\}$ and $\min\{c_i^s - c_k^s : c^s \in D\}$. If the value of the solution of the first program is positive or zero and the value of the solution of the second program is negative or zero, $i \in C(k)$. In this case the above algorithms are polynomial.

We observe that if D is a bounded polyhedron, the value c_i^s of each edge i is uncertain but is related with the value of the other edges of G.

References

[1] I. Aron, P. Hentenryck, On the complexity of the robust spanning tree problem, Oper. Res. Lett. 32 (1) (2004) 36–40.

[2] J.A. Bondy, U.S.R. Murty, Graph Theory with Applications, American Elsevier, New York, Macmillan, London.

[3] P. Kouvelis, G. Yu, Robust Discrete Optimization and its Applications, Academic Publisher, Boston, 1997.

[4] H. Yaman, O.E. Karasan, M.C. Pinar, The robust spanning tree problem with interval data, Oper. Res. Lett. 29 (2001) 31–40.