Journal of the Operations Research
Society of Japan
Vol. 24, No. 1, March 1981

MINIMUM SPANNING TREE WITH NORMAL VARIATES AS WEIGHTS

Tetsuo Ichimori
Osaka University

Shogo Shiode
Osaka University

Hiroaki Ishii
Osaka University

Toshio Nishida
Osaka University

(Received May 10, 1980; Final September 3, 1980)

Abstract When weights of arcs in a graph are normal variates, we seek a spanning tree maximizing the probability that the sum of weights of arcs in the spanning tree is not greater than a given constant. An $O(e^2n)$ algorithm for it is given.

1. Introduction

The minimum spanning tree problem is one of the most important and fundamental combinatorial optimization problems. The best time complexity of algorithms for the problem is $O(e\log\log n)$ which is due to Yao[5] where e and n are the numbers of arcs and nodes respectively in a graph. However, in the problem, the weights of arcs in the graph are in general assumed to be deterministic rather than stochastic.

We assume in this paper that weights of arcs in a graph are independent random variables according to the normal distributions. We seek a spanning tree maximizing the probability that the sum of arc-weights in the tree (the weight of the tree) is not greater than a given constant value. In this context we refer to this spanning tree as a minimum spanning tree. We assume in addition that the given constant value is greater than the maximum value of the mean of sum of arc-weights among all spanning trees. If the assumption fails, since no spanning tree has a probability greater than 0.5 (this is shown below), we can not accept any spanning tree in an ordinary sense.

We first discuss this problem and then develop an algorithm for it which runs in time $O(e^2n)$.

2. Minimum Spanning Tree

Let G=(N,A) be a graph where N is the set of nodes and A is the set of undirected arcs connecting nodes. Arcs (denoted by $e(j)$ or only by j) have weights $c(j)$, and weights $c(j)$ are independent normal variates with means $m(j)$ and variances $v(j)$. Let T denote a spanning tree in G and let AST refer to the set of all spanning trees in G.

Given any tree T, we define the following:

$$C(T)=\sum_{j\in T} c(j),\quad M(T)=\sum_{j\in T} m(j),\quad S(T)=\Big(\sum_{j\in T} v(j)\Big)^{0.5} \text{ and } V(T)=\sum_{j\in T} v(j).$$

We assume a given constant f is such that $f\geq \min_{T\in \text{AST}} M(T)$. Then the (stochastic) minimum spanning tree problem is written;

$$\max_{T\in\text{AST}} Pr\{C(T)\leq f\}.$$

Noting that $C(T)$ is also distributed according to the normal distribution with mean $M(T)$ and variance $V(T)$, the problem of solving the preceding problem is equivalent to solving the following one;

$$\max_{T\in\text{AST}} Pr\{(C(T)-M(T))/S(T)\leq (f-M(T))/S(T)\}.$$

Hence our task is to maximize $(f-M(T))/S(T)$. Note that $\max(f-M(T))/S(T)>0$ is equivalent to $f>\min M(T)$. Hence if $f\leq\min M(T)$, then $\max Pr\{C(T)\leq f\}\leq 0.5$.

Let $T(t)$ denote a spanning tree minimizing $(M(T)+tS(T))$ for a parameter $t>0$, and let T^* and t^* denote the spanning tree and value of t satisfying $\min(M(T)+tS(T))=f$, i.e., $M(T^*)+t^*S(T^*)=f$. We note that from the results of ratio minimization problems (see[3]), we have $\max(f-M(T))/S(T)=t^*$. Therefore what we must do is to seek T^*.

3. Search for T*

For two spanning trees T and T', if $M(T)=M(T')$ and $S(T)=S(T')$, then we write $T\cong T'$. In other words we do not distinguish them even if the arcs in T and those in T' are different. Otherwise we write $T\ncong T'$.

Theorem 1. For any $x>0$ define $\overline{T}(x)$ to be a spanning tree minimizing $(M(T)+xV(T))$ over every T in AST. Then we have

$T^*\in\{\overline{T}(x)\,|\,x>0\}$.

Proof: Let $T\cong T(t)$ and $T'\ncong T(t)$. Then we have $M(T')+tS(T')\geq M(T)+tS(T)$ by the optimality of $T(t)$ or $M(T')-M(T)\geq t(S(T)-S(T'))$, and we have $M(T')\neq M(T)$ and/or $S(T')\neq S(T)$ by definition of different trees, $T'\ncong T$.

If we set $x=t/(2S(T))$, then it follows that

$$\begin{aligned}M(T')+xV(T')-(M(T)+xV(T)) &= M(T')+x(S(T'))^2-(M(T)+x(S(T))^2)\\ &= M(T')-M(T)-(t/(2S(T)))((S(T))^2-(S(T'))^2)\\ &\geq t(S(T)-S(T'))-(t/(2S(T)))((S(T))^2-(S(T'))^2)\\ &= (t/(2S(T)))(S(T)-S(T'))(2(S(T))-(S(T)+S(T')))\\ &= (t/(2S(T)))(S(T)-S(T'))^2 \geq 0.\end{aligned}$$

If $M(T')+tS(T')=M(T)+tS(T)$ and $S(T')=S(T)$, then we have $M(T')=M(T)$, which is a contradiction. Hence $M(T')+tS(T')=M(T)+tS(T)$ means $S(T')\neq S(T)$, and $S(T')=S(T)$ means $M(T')+tS(T')>M(T)+tS(T)$. Thus for $x=t/(2S(T))$ we have $M(T')+xV(T')>M(T)+xV(T)$, i.e., $T(t)\cong\overline{T}(x=t/(2S(T(t)))$. (Recall the definition of $T(t)$ and $\overline{T}(x)$.) This shows $T(t)\in\{\overline{T}(x)\,|\,x>0\}$. Since T^* is a spanning tree such that $M(T(t))+tS(T(t))=f$, we have $T^*\in\{\overline{T}(x)\,|\,x>0\}$. □

Let $t(x)=(f-M(\overline{T}(x)))/S(\overline{T}(x))$ for $\overline{T}(x)$, then $t^*=max\{t(x)\,|\,x>0\}$ by definition of t^* and Theorem 1. Although it is very difficult to find $T(t)$, $\overline{T}(x)$ is easily found, since

$$M(T)+xV(T)=\sum_{j\in T}(m(j)+xv(j))=\sum_{j\in T}w_x(j),$$

where $w_x(j)=m(j)+xv(j)$ for any arc $e(j)$, namely, since finding $\overline{T}(x)$ is equivalent to finding a usual minimum spanning tree. We refer to $w_x(j)$ as a (deterministic) weight of arc $e(j)$.

Define $R(i,j)=(m(i)-m(j))/(v(j)-v(i))$ for each pair of arcs such that $m(i)<m(j)$ and $v(i)>v(j)$. Rearrange different positive $R(i,j)$ as follows;

$$0=R(0)<R(1)<R(2)<\cdots<R(h)<R(h+1)=\infty,$$

where h is the number of such $R(i,j)$.

Theorem 2. For any x and x' such that $R(k)<x(x')<R(k+1)$, $k=0,1,\cdots,h$, we have $\overline{T}(x)\cong\overline{T}(x')$.

Proof: See [1]. □

Letting $x(k)=(R(k)+R(k+1))/2$, $k=0,1,\cdots,h-1$, and $x(h)=R(h)+1$, we have the following Theorem 3.

Theorem 3. $T^*\in\{\overline{T}(x)\,|\,x=x(k),\ k=0,1,\cdots,h\}$.

Proof: From Theorem 2 for any x such that $R(k)<x<R(k+1)$, $\overline{T}(x)\cong\overline{T}(x(k))$. This and Theorem 1 together show this theorem. □

Let $R(k)=R(i,j)$, then note that we have $w_x(i)<w_x(j)$ for $x<R(k)$ and $w_x(i)>w_x(j)$ for $x>R(k)$. That is, if $x=x(k-1)$, then $w_x(i)<w_x(j)$, and if $x=x(k)$, then $w_x(i)>w_x(j)$.

Theorem 4. If $e(i)\in\overline{T}(x(k-1))$, $e(j)\notin\overline{T}(x(k-1))$, and $\overline{T}(x(k-1))\cup e(j)$ has a cycle in which both $e(i)$ and $e(j)$ are included, then $\overline{T}(x(k))=\overline{T}(x(k-1))\cup e(j)-e(i)$. Otherwise $\overline{T}(x(k))=\overline{T}(x(k-1))$.

Proof: With respect to $x(k-1)$ and $x(k)$ the order among all weights is the same except $w_x(i)$ and $w_x(j)$. Then it is obvious that $\overline{T}(x(k))$ created as above is a minimum spanning tree since there exists no other spanning tree through an elementary tree transformation whose weight is less than that of $\overline{T}(x(k))$, (Deo [2], Theorem 3-16). □

Let two arcs $e(i)$ and $e(j)$ for $R(k)=R(i,j)$ be written as $i(k)$ and $j(k)$ respectively.

Algorithm

(1) Compute $x(k)$.

(2) (i) Generate $\overline{T}(x(k))$ from $\overline{T}(x(k-1))$.

(ii) Set $t(k)=(f-M(\overline{T}(x(k))))/S(\overline{T}(x(k)))$.

(3) Find $t^*=max\ t(k)$. Stop.

4. Time Complexity

The number h is bounded by the number of intersection points of e functions $w_x(j)=m(j)+xv(j)$, which means $h=O(e^2)$. Hence Step 1 is computed in time $O(e^2)$. Let us consider the time complexity of Step 2. Initially $\overline{T}(x(0))$ is obtained in time $O(e\log\log n)$ as mentioned before.

Label each arc as 1 if it belongs to a spanning tree and as 0 otherwise. Then the check whether $i(k)\in\overline{T}(x(k-1))$ is done in time $O(1)$ by looking up the label of $i(k)$. If $i(k)\in\overline{T}(x(k-1))$ and $j(k)\notin\overline{T}(x(k-1))$, then it is necessary to determine whether or not $i(k)$ and $j(k)$ are both included in a cycle of $\overline{T}(x(k-1))\cup j(k)$, which implies whether $\overline{T}(x(k-1))\cup j(k)-i(k)$ is a spanning tree or not. If $\overline{T}(x(k-1))\cup j(k)-i(k)$ is not a spanning tree, then it must consist of two components, which is detected in time $O(n)$ by the depth first search due to Tarjan [4]. Hence Step 2 (i) needs $O(e^2n)$ time. If we have the value of $M(\overline{T}(x(k-1)))$, then the value of $M(\overline{T}(x(k)))$ is given in time $O(1)$ since at most two arcs change. So is $S(\overline{T}(x(k)))$. Hence Step 2 (ii) is done in time $O(e^2)$. Lastly Step 3 needs $O(e^2)$. Therefore the overall time complexity is $O(e^2n)$.

References

[1] Chandrasekaran, R.: Minimal Ratio Spanning Trees. *Math. of Opns. Res.* Vol.4, (1979), 414-424.

[2] Deo, N.: *Graph Theory with Applications to Engineering and Computer Science*. Prentice-Hall, N.J., 1974.

[3] Lawler, E. L.: *Combinatorial Optimization; Networks and Matroids.*, Holt, Rinehart and Winston, New York, 1976.

[4] Tarjan, R. E.: Depth First Search and Linear Graph Algorithms. *SIAM J, Computing*, Vol.1, (1972), 146-160.

[5] Yao, A. C.: An O(|E|loglog|V|) Algorithm for Finding Minimum Spanning Trees. *Information Processing Lett.*, Vol.4, (1975), 21-23.

Tetsuo ICHIMORI: Department of
Applied Physics,
Faculty of Engineering,
Osaka University,
Suita, Osaka, 565,
Japan.