アルゴリズム 48−9
(1995. 11. 17)

オンライン全域木の平均コンペティティブ比について

鮑 豊　梅 傲寒　五十嵐善英

群馬大学 工学部 情報工学科
〒376 桐生市天神町 1-5-1

本論文は、ユークリッド平面上の同じ分布点からなるオンライン全域木の平均コンペティティブ比について論じる。どのようなオンラインアルゴリズムでもそのオンライン全域木の平均コンペティティブ比が $\frac{1}{8}\ln n - \frac{1}{2}$ 以上になるような n 個の点の分布を示す。

Average Competitive Ratios of On-Line Spanning Trees

Feng Bao, Aohan Mei and Yoshihide Igarashi

Department of Computer Science
Gunma University, Kiryu, 376 Japan

We study the average competitive ratio of on-line spanning trees with the same distribution of points in the Euclidean plane. We show a distribution of n points such that the average competitive ratio of on-line spanning trees by any on-line algorithm cannot be less than $\frac{1}{8}\ln n - \frac{1}{2}$.

1 Introduction

The cost of a tree is defined to be the sum of its edge lengths. The minimum spanning tree of a set of points in the Euclidean plane is a spanning tree of the point set such that its cost is not greater than the cost of any other spanning tree of the same point set, where the length of each edge connecting two points is the Euclidean distance between the two points. The Euclidean distance between two points u and v is denoted by $|u, v|$. We assume that each point is supplied one by one. Each point is connected to one of the points supplied already when it is supplied and it is not on any edge. When the new point is on an edge that has been already added, the edge is divided into two edges at the new point. Each connection should not be changed once it has been chosen, but the case where a new point is supplied on an edge is an exception. For example, if a point v is supplied on an edge (v_1, v_2) that has been already chosen, then (v_1, v) and (v, v_2) are new edges and (v_1, v_2) is not an edge any more. We assume that we do not know what points will be supplied in advance. The on-line spanning tree problem is to construct a spanning tree in the Euclidean plane from a sequence of points under these assumptions so that its cost is as small as possible. A logarithm in basis 2 is denoted by log, and a logarithm in basis e is denoted by ln.

The competitive ratio of a spanning tree constructed by an on-line algorithm is the ratio of its cost to the cost of the minimum spanning tree of the same point set. It has been shown that the worst case competitive ratio of any on-line algorithm for the on-line spanning tree problem is at least $\frac{1}{2}\log n$ [2]. It has been also shown that a lower bound on the worst case competitive ratio for the on-line Steiner tree problem in the Euclidean plane is $\Omega(\frac{\log n}{\log\log n})$ [1].

The average competitive ratio of spanning trees with an identical set of n points constructed by an on-line algorithm is the average of the competitive ratios taken among $n!$ different orders of supplying the n points in the set. It has been unknown whether the average competitive ratio of the on-line spanning tree problem is also $\Omega(\log n)$. In this paper we show a distribution of n points such that the average competitive ratio by any on-line algorithm cannot be less than $\frac{1}{8}\ln n - \frac{1}{2}$. That is, a lower bound on the worst case of the average competitive ratios for the on-line spanning tree problem is $\Omega(\log n)$.

2 Average Competitive Ratios

Let S_n be the collection of sets with n points in the Euclidean plane. The cost of the minimum spanning tree of a point set S is denoted by $OPT(S)$. The set of sequences of the points in a set S is denoted by $\pi(S)$. The cost of a spanning tree constructed by an on-line algorithm A for input $(v_1, v_2, \cdots, v_n)$ supplied in this order is denoted by $A(v_1, v_2, \cdots, v_n)$. The average competitive ratio of an on-line algorithm A for a set S with n points is defined to be $\sum_{q\in\pi(S)} A(q)/(n!\ OPT(S))$. The worst case of average competitive ratios of an on-line algorithm A is defined as the following function $\tilde{R}(A, n)$:

$$\tilde{R}(A, n) = \sup_{S\in S_n}\{\frac{1}{n!}\frac{\sum_{q\in\pi(S)} A(q)}{OPT(S)}\}.$$

We first consider the following on-line greedy algorithm G: Whenever a point is supplied, it is connected to the nearest point among the points that have been already supplied. Assume that as shown in Figure 1, n points are supplied in the order $v_1, v_2, \cdots, v_n$ as an input to the on-line greedy algorithm G. The Euclidean distance between v_1 and v_2 is 1, and the length of an arc connecting v_1 and v_2 is less than $1+\epsilon$ for an arbitrary small ϵ, where all of $v_1, v_2, \cdots, v_n$ are on the arc as shown in Figure 1. The cost of the minimum spanning tree of the set of these points is between 1 and $1+\epsilon$. On the other hand, the cost of the spanning tree constructed by the on-line greedy algorithm G for the input sequence, $v_1, v_2, \cdots, v_n$ is given as follows:

$$|v_1, v_2| + |v_1, v_3| + |v_1, v_4| + |v_3, v_5| + |v_1, v_6| + |v_3, v_7| + \cdots$$

$$> 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \cdots$$

If $n = 2^k + 1$, the right hand side of the formula above is equal to $1 + \frac{1}{2}k$. For this example, any on-line algorithm cannot be better than the on-line greedy algorithm, from

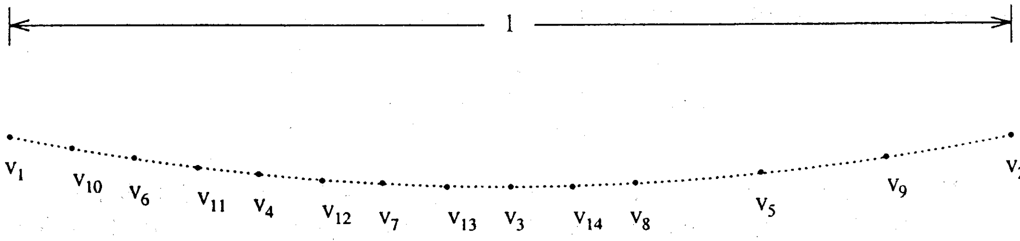


Figure 1: An input sequence of n points.

the inequality above a lower bound on the worst case of competitive ratios of any on-line algorithm A is $\frac{1}{2}(\log n)/(1+\epsilon)$ for each n. This simply obtained fact coincides with the corresponding result about the steiner tree problem given in [2].

We now consider average competitive ratios for the on-line spanning tree problem. Our first result is a lower bound on the worst case of average competitive ratios of the greedy on-line algorithm G.

Theorem 1 *For the on-line greedy algorithm G, $\tilde{R}(G,n) > \frac{1}{4}\ln n - 1$.*

Proof. Let S be a set of n points uniformly distributed on a very gentle arc (i.e., a small part of a circle with a very large radius). Then the length of any part of the arc is nearly equal to the Euclidean distance between its end points (see Figure 2). Then any part of the arc may be regarded as an almost straight line, but no three points on the arc are on any straight line.

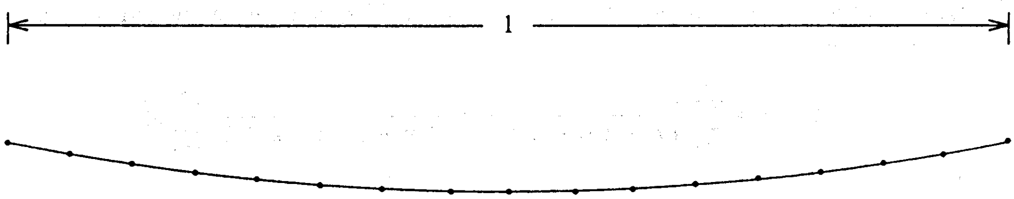


Figure 2: A set S of n points on an arc.

From the definition of $\tilde{R}$, we have

$$\tilde{R}(G,n) \geq \frac{1}{n!}\sum_{q\in\pi(S)} G(q).$$

Suppose that $k-1$ points have been already supplied. Let S_{k-1} be the set of these $k-1$ points. The kth point is now being supplied. It is equally probable to be one of the other $n-k+1$ points in S. We denote the distance between v in $S-S_{k-1}$ and its nearest point in S_{k-1} by $\rho(S_{k-1},v)$. The expected distance between the kth point and S_{k-1} is

$$\tilde{\rho}(S_{k-1}) = \left(\sum_{v\in S-S_{k-1}} \rho(S_{k-1},v)\right) / (n-k+1).$$

From the definition of $\tilde{R}(G,n)$, the following equality is immediate:

$$\tilde{R}(G,n) = \sum_{k=2}^{n} \left(\frac{\sum_{S_{k-1}\in S^{k-1}} \tilde{\rho}(S_{k-1})}{\binom{n}{k-1}} \right).$$

We first prove that for any S_{k-1}, $\tilde{\rho}(S_{k-1}) > (n-k+1)/(4k(n-1))$. Suppose that $S_{k-1} = \{v_1, v_2, \cdots, v_{k-1}\}$ and the sorted order of these points on the arc from left to right is $v_1, v_2, \cdots, v_n$.

Suppose that in $S - S_{k-1}$ there are s_1 points before v_1, s_k points after v_{k-1}, and s_i points between v_{i-1} and v_i for each i $(1 \leq i \leq k-1)$ (see Figure 3).

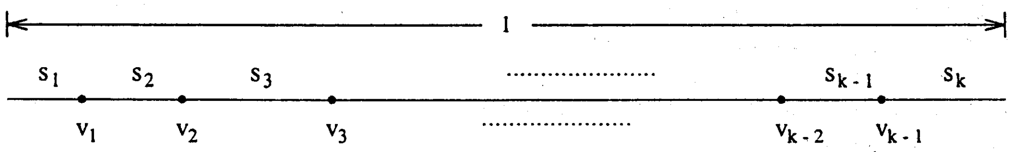


Figure 3: The point distribution in S.

Then apparently, $\sum_{i=1}^{k} s_i = n-k+1$. Since v_k is randomly chosen from $S - S_{k-1}$, the probability that v_k appears between v_{i-1} and v_i is $\frac{s_i}{n-k+1}$ for each i $(1 \leq i \leq k)$, where for $i=1$ and $i=k$ we mean that v_k appears before v_1 and after v_{k-1}, respectively. Suppose that v_k appears between v_{i-1} and v_i (again for $i=1$ or k it means before v_1 or after v_{k-1}, respectively). Then the expected length between v_k and its nearest point in S_{k-1} is larger than $s_i/(4(n-1))$ unless $s_i = 0$. If $s_i = 0$ the expected length is 0. Hence, we have

$$\tilde{\rho}(S_{k-1}) > \sum_{i-1}^{k} \frac{1}{4} \frac{s_i}{n-1} \frac{s_i}{n-k+1} = \frac{1}{4(n-1)(n-k+1)} \sum_{i-1}^{k} s_i^2.$$

Since

$$s_1^2 + s_2^2 + \cdots + s_k^2 \geq k\left(\frac{s_1+s_2+\cdots+s_k}{k}\right)^2 = \frac{(n-k+1)^2}{k},$$

we have

$$\tilde{\rho}(S_{k-1}) > (n-k+1)/(4k(n-1)).$$

Let S^{k-1} be the collection of sets with k points chosen from S. From the inequality shown above we have

$$\begin{aligned} \tilde{R}(G,n) &= \sum_{k=2}^{n} \left(\frac{\sum_{S_{k-1}\in S^{k-1}} \tilde{\rho}(S_{k-1})}{\binom{n}{k-1}} \right) \\ &> \frac{1}{4}\sum_{k=2}^{n} \frac{n-k+1}{(n-1)k} > \frac{1}{4}\sum_{k=2}^{n}\left(\frac{1}{k} - \frac{1}{n-1}\right) \\ &\simeq \frac{1}{4}(\ln n - 2). \end{aligned}$$

□

3 Comparison with Other On-Line Algorithms

In this section we show that for any input sequence of points, no on-line algorithm is more than twice as good as the on-line greedy algorithm.

Theorem 2 *For any input sequence of points, the competitive ratio of a spanning tree constructed by the on-line greedy algorithm is not greater than twice the competitive ratio of a spanning tree constructed by any on-line algorithm.*

Proof. Let A be an on-line algorithm for the on-line spanning tree problem. Let $(v_0, v_1, v_2, \cdots, v_n)$ be an input sequence of $n+1$ points in the Euclidean plane. Suppose that $a_1, a_2, \cdots, a_n$ are the edges constructed by A in this order, where the following modification is used. For each i $(2 \leq i \leq n)$, if v_i is on an edge already constructed by A, then a_i is an edge with the null length and the edge already constructed remains unchanged. Let $g_1, g_2, ..., g_n$ be the edges constructed by the on-line greedy algorithm G in this order, where we also apply the same modification as described for $a_1, a_2, \cdots, a_n$. Apparently the cost of any spanning tree by an on-line algorithm is equal to the cost of the corresponding modified one. We denote the length of an edge l by $|l|$.

We classify set $\{1, 2, ..., n\}$ into two classes, N and Z as follows. If $|a_i| \neq 0$ then $i \in N$, and otherwise $i \in Z$. From the definition of the on-line greedy algorithm, for each $i \in N$, $|a_i| \geq |g_i|$. Hence, we have

$$\sum_{i \in N} |a_i| \geq \sum_{i \in N} |g_i|.$$

If $i \in Z$ and v_i appears on an edge already constructed by A, say a_j, then $i > j$. Let $v_{i_1}, v_{i_2}, ..., v_{i_k}$ be all the points on a_j and $j < i_1 < \cdots < i_k$. Then $|a_j| > \sum_{j=1}^{k} |g_{i_j}|$, and we have

$$\sum_{i \in N} |a_i| \geq \sum_{i \in Z} |g_i|.$$

Hence, we have

$$2\sum_{i=1}^{n} |a_i| \geq \sum_{i=1}^{n} |g_i|.$$

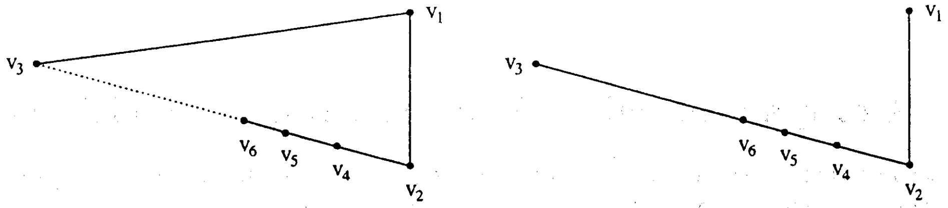
□

Consider an input sequence $v_1, v_2, \cdots, v_n$ as shown in Figure 4. If $|v_1, v_2|$ is much smaller than the distance between v_1 and v_3, then for this input sequence the competitive ratio of a spanning tree constructed by the on-line greedy algorithm trends to twice the competitive ration of a spanning tree constructed by an on-line algorithm such that for each i $(i = 1, 2, \cdots)$, v_{i+1} is connected with v_i. From this example we can say that the assertion of Theorem 2 is critical.

From Theorem 1 and Theorem 2 the following theorem is immediate.

Theorem 3 *For any on-line algorithm A for the on-line spanning tree problem,*

$$\tilde{R}(A, n) > \frac{1}{8} \ln n - \frac{1}{2}.$$



(a). Construction by the on-line greedy algorithm.

(b). Construction by an on-line algorithm.

Figure 4: A comparison by an example of an input sequence.

4 Conclusion

It has been known that the worst case competitive ratio of the on-line greedy algorithm is $\Theta(\log n)$ [2]. In this paper we showed that the worst case of average competitive ratios of spanning trees with identical point sets by the on-line greedy algorithm is also $\Theta(\log n)$. It is still open whether the worst case of average competitive ratios of steiner trees with identical point sets is $\Omega(\frac{\log n}{\log \log n})$.

References

[1] N. Alon and Y. Azar, "On-line steiner trees in the Euclidean plane." In *Proc. 8th ACM Symposium on Computational Geometry*, pp. 337-343, 1992.

[2] M. Imase and B. M. Waxman, "Dynamic steiner tree problem." *SIAM J. Discrete Math.* Vol. 4, No. 3, pp. 369-384, 1991.

[3] Y. T. Tsai and C. Y. Tang, "The competitiveness of random algorithms for on-line Steiner tree and on-line spanning tree problems." *Information Processing Letters*, 48(1993), pp. 177-182.