Better Online Steiner Trees on Outerplanar Graphs

Akira Matsubayashi[1,a)]

Abstract: This report addresses the classical online Steiner tree problem on edge-weighted graphs. It is known that a greedy (nearest neighbor) online algorithm is $O(\log n)$-competitive on arbitrary graphs with n vertices. It is also known that no deterministic algorithm is $o(\log n)$-competitive even on series-parallel graphs. The greedy algorithm is trivially 1- and 2-competitive on trees and rings, respectively, but $\Omega(\log n)$-competitive even on outerplanar graphs. The author proposed a non-greedy algorithm and proved that the algorithm is 8-competitive on outerplanar graphs. In this report, we improve the analysis and prove that this algorithm is 7.464-competitive on outerplanar graphs. We also present a lower bound of 4 for arbitrary deterministic online Steiner tree algorithms on outerplanar graphs.

Keywords: Steiner tree, outerplanar graph, online algorithm, competitive analysis

1. Introduction

This report addresses the classical online Steiner tree problem on edge-weighted graphs. We are given a graph $G = (V_G, E_G)$ with non-negative edge-weights $w : E_G \to \mathbb{R}^+$ and a subset R of vertices of G. The (offline) Steiner tree problem is to find a *Steiner tree*, i.e., a subtree $T = (V_T, E_T)$ of G that contains all the vertices in R and minimizes its cost $c(T) = \sum_{e \in E_T} w(e)$. In the online version of this problem, vertices $r_1, \ldots, r_{|R|} \in R$ are revealed one by one, and for each $i \geq 1$, we must construct a tree containing r_i by growing the previously constructed tree for $r_1, \ldots, r_{i-1}$ (null tree for $i = 1$) without information of $r_{i+1}, \ldots, r_{|R|}$.

It is known that a greedy (nearest neighbor) online algorithm is $O(\log n)$-competitive on arbitrary graphs with n vertices [6]. It is also known that no deterministic algorithm is $o(\log n)$-competitive even on series-parallel graphs [6]. The greedy algorithm is trivially 1- and 2-competitive on trees and rings, respectively, but $\Omega(\log n)$-competitive even on outerplanar graphs. No other nontrivial class of graphs that admits constant competitive deterministic Steiner tree algorithms had been known, until the author recently presented a non-greedy algorithm that is 8-competitive on outerplanar graphs [7]. As for randomized algorithms, a probabilistic embedding of outerplanar graphs into tree metrics with distortion 8, presented by Gupta, Newman, Rabinovich, and Sinclair [5], implies an 8-competitive online Steiner tree algorithm against oblivious offline adversaries. Various generalizations of the online Steiner tree problem are also studied, such as generalized STP [2], vertex-weighted STP [8], and asymmetric STP [1].

In this report, we improve the analysis of the algorithm proposed in [7] and prove that this algorithm is 7.464-competitive on outerplanar graphs. This algorithm connects a requested vertex and the previously constructed tree using a path that is constant times longer than a shortest path between the requested vertex and the tree. An interesting application of the online steiner tree problem is the file allocation problem, in which we maintain a dynamic allocations of multiple copies of data file on a network with servicing online read/write requests. Bartal, Fiat, and Rabani [3] propose a file allocation algorithm based on any online Steiner algorithm. With this result, our result implies a $7.464(2 + \sqrt{3})(\approx 27.86)$-competitive randomized file allocation algorithm against adaptive online adversaries.

2. Preliminaries

Graphs considered here are undirected and have non-negative edge-weights, $w(e) \geq 0$ for any edge e. For a graph G, we denote its vertex set and edge set by V_G and E_G, respectively. We use the notation of w also for graphs, i.e., $w(G) := \sum_{e \in E_G} w(e)$. For a subset R of vertices of G, a *Steiner tree of G for R* is a subtree T of G such that $R \subseteq V_T$. T is said to be *minimum* if T has the minimum cost $w(T)$ overall Steiner trees of G for R.

Suppose that G is a planar graph. The *weak dual* of G is a graph H such that V_H is the set of bounded faces of G, and E_H is the set of two bounded faces F and F' that have a common edge. G is *outerplanar* if it can be drawn on the plane so that all the vertices belong to the unbounded face, or equivalently, if H is a forest [4]. We say an edge of G to be *outer* if the edge is contained in the unbounded face, *inner* otherwise.

In the rest of the report, we assume that G is biconnected, because finding a minimum Steiner tree of G can easily be reduced to finding minimum Steiner trees of biconnected components of G. This assumption implies that H is a tree. Let $d_G(u, v)$ be the distance (i.e., the length of a shortest path) of vertices u and v in G. We use the notation of d_G also for the distance between a graph and a vertex, i.e., $d_G(G', v) := \min\{d_G(u, v) \mid u \in V_{G'}\}$ for a subgraph G' of G and $v \in V_G$.

[1] Division of Electrical Engineering and Computer Science, Kanazawa University Kakuma-machi, Kanazawa, 920–1192 Japan
[a)] mbayashi@t.kanazawa-u.ac.jp

3. Algorithm and Analysis

3.1 Algorithm α-Detour

Suppose that we are given an outerplanar graph G with edge-weights $w : E_G \to \mathbb{R}^+$, and a sequence $r_1, r_2, \dots, r_{|R|} \in R \subseteq V_G$. Our algorithm, denoted by α-Detour ($\alpha > 1$), constructs trees $T_1, T_2, \dots, T_{|R|}$ as follows:

For the first vertex r_1, we define T_1 as the tree consisting of the single vertex r_1. We suppose that the weak dual H of G is a tree rooted by a face containing r_1. We introduce a forest F with $V_F = E_G$ as follows: If C is the root of H, then all the edges of C are the roots of the connected components of F. Moreover, if C is a face of G, and C' is a child of C in H, then all the edges of $E_{C'} \setminus E_C$ are the children of the unique edge $e \in E_C \cap E_{C'}$ in F. For any inner edge e of G, let F_e be the sub-forest of F induced by the descendants of e in F, and G^F_e be the subgraph of G induced by V_{F_e}, i.e., by the descendants of e in F_e. (Note that neither F_e nor G^F_e contains e.)

For the ith vertex r_i with $i \geq 2$, α-Detour performs the following steps:

α-Detour

(1) If $r_i \in V_{T_{i-1}}$, then return $T_i := T_{i-1}$.
(2) Otherwise, find a shortest path $P = (p_1, p_2, \cdots, p_{|P|})$ between a vertex p_1 in T_{i-1} and $p_{|P|} = r_i$. If there are two or more such shortest paths, then choose one consisting of edges as close to roots in F as possible.
(3) Let $T_i := T_{i-1}$.
(4) For $j = 1$ to $|P| - 1$, if $p_{j+1} \notin V_{T_i}$, then call Detour-edge(α, p_j, p_{j+1}) defined below.
(5) Return T_i.

Detour-edge(x, u, v) is a procedure to modify T_i by adding a maximal path between T_i and v of length at most $x \cdot w(uv)$, where $x \geq 1$, and uv is an edge such that $u \in V_{T_i}$, $v \notin V_{T_i}$, and $w(uv) \leq d_G(T_i, v)$. The procedure is formally defined as follows:

Detour-edge(x, u, v)

(1) If uv is outer, then add uv to T_i, and return.
(2) If uv is inner, then find a shortest path $Q = (q_1, \dots, q_{|Q|})$ from a vertex q_1 in T_i to $q_{|Q|} = v$ in G^F_{uv}. If there are two or more such shortest paths, then choose one consisting of edges as close to uv in F_{uv} as possible.
(3) If $w(Q)/w(uv) > x$, then add uv to T_i.
(4) Otherwise, call Detour-edge($x \cdot w(uv)/w(Q), q_j, q_{j+1}$) for $j = 1$ to $|Q| - 1$.
(5) Return.

3.2 Correctness

Since α-Detour and Detour-edge only add edges to T_{i-1}, T_i contains T_{i-1} as a subgraph. Therefore, it suffices to show that α-Detour connects r_i to T_i.

Lemma 1 *Detour-edge(x, u, v) adds a path of length at most $x \cdot w(uv)$ that connects a vertex of T_i and v.*

Proof We prove this lemma by induction on the *depth of uv*, i.e., the distance in F from uv to the root. If uv is outer, then the procedure chooses uv as a path connecting u and v. Therefore, this path has length $w(uv) \leq x \cdot w(uv)$.

Assume that uv is inner, and that the lemma holds for any depth larger than that of uv. If $w(Q)/w(uv) > x$ in Step 3, then the procedure chooses uv as a path connecting u and v, and therefore, the lemma holds. Otherwise, by induction hypothesis, Detour-edge($x \cdot w(uv)/w(Q), q_1, q_2$) adds a path of length at most $x \cdot w(uv)w(q_1q_2)/w(Q)$ that connects a vertex in T_i and q_2 in $G^F_{q_1q_2}$. We note that because q_1q_2 is a descendant of uv, every path connecting a vertex in T_i and v must pass through q_2 at this point. This means that $q_3 \notin V_{T_i}$ and $w(q_2q_3) = d_G(T_i, q_3)$. Therefore, by induction hypothesis again, Detour-edge($x \cdot w(uv)/w(Q), q_2, q_3$) adds a path of length at most $x \cdot w(uv)w(q_2q_3)/w(Q)$ that connects a vertex in T_i and q_3 in $G^F_{q_2q_3}$. Repeating this process for all $j < |Q|$, we conclude that Detour-edge(x, u, v) adds a path of length at most $\sum_j (x \cdot w(uv)w(q_jq_{j+1})/w(Q)) = x \cdot w(uv)$ that connects a vertex in T_i and v.

□

Since α-Detour calls Detour-edge(α, p_j, p_{j+1}) unless p_{j+1} has already been contained in T_i, by Lemma 1, we have the following lemma:

Lemma 2 *For $i \geq 2$, α-Detour connects r_i to T_i with a path of length at most $\alpha \cdot d_G(T_{i-1}, r_i)$.*

3.3 Competitiveness

To analyze competitiveness of α-Detour, we modify F as the Steiner tree grows. Then, we partition a planar drawing of G according to the modified forest.

3.3.1 Modifying Forest

Every time Detour-edge(α, p_j, p_{j+1}) is called in Step 4 of α-Detour, we mark p_jp_{j+1} "greedy". Before processing the Detour-edge(α, p_j, p_{j+1}), if p_jp_{j+1} is an ancestor of one or more maximal subtrees of F rooted by "greedy" edges e, then we remove (e, e') from E_F, where e' is a parent of e. This yields new connected components rooted by "greedy" edges.

Let F^* denote the modified forest. For any inner edge e in G, just as defined for F, F^*_e is the sub-forest of F^* induced by the descendants of e in F^*, and $G^{F^*}_e$ is the subgraph of G induced by $V_{F^*_e}$, i.e., by the descendants of e in F^*_e.

For every edge uv such that Detour-edge(x, u, v) is called, let Q_{uv} be the path Q constructed in Step 2 for uv. We note that Detour-edge(x, u, v) is processed only in G^F_{uv}. Moreover, for any edge $u'v'$ in F_{uv} that is an ancestor of an edge of Q_{uv}, Detour-edge($\cdot, u', v'$) will never be called later. This is because, by the definition of Q_{uv} in Step 2 of Detour-edge, we can find a path along Q_{uv} shorter than the edge $u'v'$ from the already constructed Steiner tree to u' or v'. This implies the following lemma:

Lemma 3 *For any edge uv such that Detour-edge(x, u, v) is called, uv and edges of Q_{uv} are contained in the same connected component of F^*.*

3.3.2 Partition of Planar Drawing

We regard edges and paths as line segments of the preserved length on an outerplanar drawing of G. We partition the drawing by subdividing edges in bottom-up fashion. We define that X is the set of inner edges e such that $G_e^{F^*}$ does not contain an outer edge in G. Such e and any of its descendants in F are in different connected components of F^*, or both of them are in X. The following is the procedure to subdivide edges:

Subdivision

(1) We do not subdivide any outer edge. We consider the subdivision of an outer edge to be itself.

(2) For an inner edge e, suppose that all its children $c_1, \ldots, c_k$ in F^* (or, all roots of connected components of F_e^*) but not in X have already been subdivided. Such children induces a path in G. For otherwise, there would be two children in F^*, and a child in F but not in F^*, which is between the former two children in G. This implies that at least one of the two children in F^* should have been in X. We define S_e as the path in G obtained by concatenating k elements, ith of which is e_i if e_i is outer or $w(e_i) \leq w(S_{c_i})$, and S_{c_i} otherwise.

(3) We subdivide e into ℓ consecutive line segments of lengths $w(e)w(s_1)/w(S_e), \ldots, w(e)w(s_\ell)/w(S_e)$, where $s_1, \ldots, s_\ell$ are the consecutive line segments into which S_e has been subdivided.

This procedure naturally induces a partition of the outerplanar drawing of G, in such a way that in Step 3, s_i and the line segment of e of length $w(e)w(s_i)/w(S_e)$ are in the same partition. We denote by $e[s_i]$ the line segment on e in this partition. Generally, we consider S_e to be projected onto e and denote $e[\bigcup_{i \in I} s_i] := \bigcup_{i \in I} e[s_i]$ for a subset I of $\{1, \ldots, \ell\}$, implying $e[S_e] = e$. For an edge e' is in F_e^*, by the definition of S_e in Step 2, either e' or $S_{e'}$ can be a part of S_e. Therefore, if e' is an ancestor of an edge in S_e, then it follows that $S_{e'}$ is a subpath of S_e. For such e', we define $e[e'] := e[S_{e'}]$. For the case that e' is a descendant of an edge in S_e, we further extend this notion in such a way that if $e[e'']$ and $e''[e']$ are already defined for some edge e'', then $e[e'] := e[e''[e']]$. With these definitions, we have defined $e[e']$ for any edges e and e' in F_e^*.

A path is said to *cover* an edge if the edge has its ancestor in the path. We can observe that S_e is:

Condition 1

(1) a path covering any outer edge in F_e^*;

(2) such a shortest path in $G_e^{F^*}$ passing through edges as close to e in F^* as possible.

Lemma 4 *For any edge uv such that Detour-edge(x, u, v) is called, it follows that $Q_{uv} = S_{uv}$.*

Proof By Lemma 3, it suffices to prove that Q_{uv} is a shortest path satisfying Condition 1(1). Let O be the set of outer edges that are descendants of uv when Q_{uv} is constructed. If O equals the set of outer edges in F_{uv}^*, then the lemma clearly holds. Assume that some edges are removed from O at later point. I.e., an ancestor ur of the removed edges is newly marked "greedy", where u is a vertex of the current Steiner tree T, and r is a new request. By Lemma 3, ur is neither contained in Q_{uv} nor an ancestor of an edge of Q_{uv}. Any path contains neither u nor r has unchanged length not shorter than Q_{uv} by its minimality. Consider a path containing u and r. Since ur is "greedy", $d_G(T \setminus ur, r) \geq w(ur)$. Therefore, to cover the remaining outer edges, we need a cost at least $w(ur)$. This means that such a path has the length same as the path containing ur, which is not shorter than Q_{uv} by its minimality. Therefore, we cannot obtain a shorter path covering outer edges. □

By a similar proof, we also have the following lemma:

Lemma 5 *For any edge uv with $w(uv) = d_G(u, v)$, it follows that $w(uv) \leq w(S_{uv})$.*

Lemma 6 *Suppose that uv is a "greedy" edge in P_i for some i, and that $\bar{P}_i$ is the path connecting a vertex of T_i and v that is constructed by Detour-edge(α, u, v) in Step 4 of α-Detour. If e is an edge in $F_{e'}^*$ for some edge e' in $\bar{P}_i$, then $w(e) > \alpha \cdot w(uv[e])$.*

Proof We prove the lemma by induction on the number of recursive depths for Detour-edge(α, u, v) to output e'.

Assume first that $uv = e'$, i.e., uv is added to T_i in Step 3 of Detour-edge(α, u, v). If e is in Q_{uv}, then since $w(Q_{uv})/w(uv) > \alpha$, it follows that

$$\begin{aligned} w(e) &= w(e)w(Q_{uv})/w(S_{uv}) \quad \text{[by Lemma 4]} \\ &> w(e) \cdot \alpha \cdot w(uv)/w(S_{uv}) \\ &= \alpha \cdot w(uv[e]). \quad \text{[by the definition of } uv[\cdot]] \end{aligned}$$

Otherwise, since an edge in F_{uv}^* that is an ancestor of an edge of Q_{uv} cannot be "greedy", e is a descendant of an edge e'' of Q_{uv}. Any path containing e and covering any outer edge in $F_{e''}^*$ is not shorter than $S_{e''}$, which is not shorter than e'' by Lemma 5. This means that $w(e) \geq w(e''[e])$. Combining with $w(e'') > \alpha \cdot w(uv[e''])$, we have $w(e) \geq w(e''[e]) > \alpha \cdot w(uv[e''[e]]) = \alpha \cdot w(uv[e])$.

Assume next that e' is output through two or more recursive calls of Detour-edge, and that the lemma holds for a smaller number of recursive calls. By this assumption, Detour-edge(x, u', v') is recursively called with $x = \alpha \cdot w(uv)/w(Q_{uv})$ for some edge $u'v'$ in Q_{uv}. Regarding this Detour-edge as being called in x-Detour, we have

$$\begin{aligned} w(e) &> x \cdot w(u'v'[e]) \quad \text{[by induction hypothesis]} \\ &= (\alpha \cdot w(uv)/w(Q_{uv})) \cdot w(u'v'[e]) \\ &= \alpha \cdot w(uv)w(u'v'[e])/w(S_{uv}) \quad \text{[by Lemma 4]} \\ &= \alpha \cdot w(uv[u'v'[e]]) \quad \text{[by the definition of } uv[\cdot]] \\ &= \alpha \cdot w(uv[e]). \end{aligned}$$

Thus, we have the lemma. □

3.3.3 Comparison to Minimum Steiner Tree

Suppose that Z is any Steiner tree for R. If an inner edge uv, shared by a face C and its child C' of G, is contained in Z, then we decompose G into two graphs G' and G'' induced by C and its ancestor faces in H, and by C' and its descendant faces in H, respectively. Decomposing G by all inner edges contained in Z, we obtain a set $\mathcal{B}$ of biconnected outerplanar subgraphs of G, each of which contains edges of Z only in its unbounded face. Unless $B \in \mathcal{B}$ contains the root of H, there is an edge e_B in B that is an ancestor in F of all the other edges of B. We note that uv is contained in Z. For convenience, if B contains the root of H, then we suppose $e_B := r_1 r_1$ and regard e_B to have weight 0. Let Z_B be the path induced by $E_B \cap E_Z$.

Lemma 7 *Suppose that for any edge z in $Z_B \setminus e_B$, $e_z \in E_B$ is the outermost "greedy" edge such that $F^*_{e_z}$ contains z. Then, it follows that*

$$\sum_e w(e[z]) < \frac{\alpha}{\alpha - 1} w(e_z[z]),$$

where the summation is overall "greedy" edges e such that F^*_e contains z.

Proof Since z is contained in $F^*_{e_z}$, any path S containing z and covering any outer edge in $F^*_{e_z}$ not shorter than S_{e_z}, which is not shorter than e_z by Lemma 5. This implies that $w(z) \geq w(e_z[z])$.

By Lemma 6, for any edges e and an descendant e' of e to be summed, it follows that $w(e') > \alpha \cdot w(e[e'])$, implying that $\alpha^{-1} w(e'[z]) > w(e[e'[z]]) = w(e[z])$. Therefore, we have $\sum_e w(e[z]) < \sum_{i \geq 1} \alpha^{-1(i-1)} w(e_t[z]) < \frac{\alpha}{\alpha-1} w(e_z[z])$. □

Lemma 8 *Suppose that O is the set of edges contained in the unbounded face of B but not in in Z_B. Then, it follows that*

$$\sum_{o \in O,\ e} w(e[o]) \leq w(Z_B),$$

where the summation is overall "greedy" edges e such that F^*_e contains o.

Proof Let D be the partitioned region of the outerplanar drawing that contains edges O. Because no vertex in R resides inside D, if a "greedy" edge e such that F^*_e contains an edge of O first enters D, then the edge must get out of D along a path consisting of "greedy" edges and reach a vertex in Z_B. We associate e with the path on Z_B connecting the end-vertices of the "greedy" path, which is not longer than the associated path. These two paths form a cycle.

A subsequent "greedy" edge e' such that $F^*_{e'}$ contains an edge of O cannot join two vertices of the cycle, for otherwise, "greedy" edges of the cycle are removed from $F^*_{e'}$, resulting only edges of Z_B in $F^*_{e'}$. Therefore, there exists a "greedy" path containing e' and satisfying either of the following conditions: If the "greedy" path connects two vertices in Z_B and not in the cycle, then we associate e' with the path on Z_B connecting these vertices. If the "greedy" path connects a vertex u in Z_B and not in the cycle, and a vertex in the cycle, then we associate e' with the path on Z_B connecting u and the cycle. In either case, the associated path with e' is edge-disjoint with the cycle and not shorter than the "greedy" path containing e'. Repeating this argument, we have the lemma. □

Lemma 9 *It follows that $w(T_{|R|}) < \alpha(3 + 1/(\alpha - 1))w(Z)$.*

Proof We can upper bound $w(T_{|R|})$ by summing up $w(e[z])$ for z and e satisfying the conditions of Lemma 7 and $w(e[o])$ for o and e satisfying the conditions of Lemma 8, for all $B \in \mathcal{B}$. Noting that edge e_B of B incurs not $w(e[z])$ in Lemma 7, it follows that

$$\begin{aligned}
w(T_{|R|}) &\leq \alpha \sum_{B \in \mathcal{B}} \left[\sum_{z \neq e_B,\ e} w(e[z]) + \sum_{o \in O,\ e} w(e[o]) \right] \quad \text{[by Lm 2]} \\
&< \alpha \sum_{B \in \mathcal{B}} \left[\sum_{z \neq e_B} \frac{\alpha}{\alpha - 1} w(e_z[z]) + w(Z_B) \right] \quad \text{[by Lms 7 \& 8]} \\
&= \alpha \sum_{B \in \mathcal{B}} \left[\frac{\alpha}{\alpha - 1} w(Z_B \setminus e_B) + w(Z_B \setminus e_B) + w(e_B) \right] \\
&= \alpha \sum_{B \in \mathcal{B}} \left[\left(2 + \frac{1}{\alpha - 1}\right) w(Z_B \setminus e_B) + w(e_B) \right] \\
&= \alpha \left(3 + \frac{1}{\alpha - 1}\right) w(Z).
\end{aligned}$$

□

Setting $\alpha = 1 + 1/\sqrt{3} \approx 1.577$, we have the following theorem:

Theorem 10 *Algorithm* 1.577*-Detour is* 7.464*-competitive.*

4. Lower Bound

In this section, we prove a lower bound of 4 for any deterministic Steiner tree algorithm on outerplanar graphs.

4.1 Definition of Graph

Let m be a positive integer and ϵ be a positive real number. Let G_0 be a path of weight 1. The unique edge of G_0 is said to be of level 0. For $i \geq 1$, let G_i be the graph obtained from G_{i-1} by adding m^i edges of weight $(1 + \epsilon)^i / \prod_{j=1}^{i} m^j$ to each edge of level $i - 1$ in such a way that the added m edges form a path connecting the end-vertices of the edge of level $i - 1$. All the added edges are said to be of level i. We suppose $G := G_i$ with sufficiently large i. We define F as the rooted tree with $V_F = E_G$ such that for an edge e of level $i - 1$, m^i edges added to e are children in F of e. We note that such children has the total weight of $(1 + \epsilon)w(e)$.

4.2 Adversary

We use a sequence K_i for $i \geq 0$ defined as follows: Let $K_0 := 1$ and K_1 be less than but sufficiently close to 3. For $i \geq 1$, we define

$$K_{i+1} := \begin{cases} (K_0 + K_1)(K_i - K_{i-1}) & \text{if } K_i < (K_0 + K_1)(K_i - K_{i-1}), \\ K_i & \text{if } K_i \geq (K_0 + K_1)(K_i - K_{i-1}). \end{cases}$$

Our adversary Adv generates a request sequence against a deterministic Steiner tree algorithm Alg on G. In the initial phase, called the 0th phase, Adv defines $Z_0 := G_0$ and requests vertices of Z_0. Let T_0 be the Steiner tree computed by Alg for these requests, and P_0 be the path in T_0 connecting the requests. For the ith phase with $i \geq 1$, Adv defines the path Z_i consisting of children in F of edges of P_{i-1}, and requests vertices of Z_i that have not been requested. Let T_i be the Steiner tree computed by Alg for all the requested vertices thus far. For an edge e in P_{i-1}, vertices incident to a child of e must be contained in the subgraph S of T_i induced by the descendants of e. If S is connected, then there is a path Q_e in S connecting the end-vertices of e. Otherwise, since T_i is connected, there is a unique child m_e such that $S \cup m_e$ has a path Q_e connecting the end-vertices of e. Let P_i be the path obtained by concatenating Q_e for all edges e in P_{i-1}.

We can inductively observe that P_i and Z_i are Steiner trees for the requests up to the ith phase. If $w(P_i) > \gamma_i w(P_{i-1})$, then Adv quits generating requests, where $\gamma_i := K_i/K_{i-1} \geq 1$. Otherwise, Alg performs the next phase.

4.3 Analysis

The following lemma is used to guarantee that Adv quits in finite phases.

Lemma 11 *There exists $\ell \geq 1$ such that $K_{\ell+1} = K_\ell$.*

Proof Let $(a_i)_{i\geq}$ be a sequence with the recurrence $a_{i+1} = b(a_i - a_{i-1})$ with $0 < b < 4$. If the recurrence is equivalent to $a_{i+1} - Aa_i = B(a_i - Aa_{i-1})$, i.e., $a_{i+1} = (A + B)a_i - ABa_{i-1}$, then $A + B = AB = b$. Hence, A and B are solutions of $x^2 - bx + b = 0$, i.e., $(b \pm \sqrt{b^2 - 4b})/2$. These solutions are conjugate complex numbers since $0 < b < 4$. This means that $a_i = \frac{B^i - A^i}{B-A}(a_1 - Aa_0) + A^i$ obtained from the recurrence oscillates. Therefore, there exists $\ell \geq 1$ such that $a_\ell \geq a_{\ell+1} = b(a_\ell - a_{\ell-1})$, implying $K_{\ell+1} = K_\ell$. □

Lemma 11 implies $\gamma_{\ell+1} = K_{\ell+1}/K_\ell = 1$, while

$$w(P_i) \geq w(Z_i) = (1 + \epsilon)w(P_{i-1}) \qquad (1)$$

by the definitions of P_i and Z_i. Therefore, Adv performs at most $\ell + 1$ phases.

The following lemma is used to estimate the ratio of the cost of Alg to the cost of Adv.

Lemma 12 $\sum_{i=0}^{j} K_i/K_{j-1} \geq K_0 + K_1$ *for any* $j \geq 1$.

Proof We prove the lemma by induction on j. The lemma is immediate for $j = 1$ since $K_0 = 1$. For $j \geq 1$, it follows that

$$\begin{aligned}\frac{\sum_{i=0}^{j+1} K_i}{K_j} &\geq \frac{(K_0 + K_1)K_{j-1} + K_{j+1}}{K_j} \quad \text{[by induction hypothesis]}\\ &\geq \frac{(K_0 + K_1)K_{j-1} + (K_0 + K_1)(K_j - K_{j-1})}{K_j}\\ &= K_0 + K_1.\end{aligned}$$

□

Lemma 13 *If* Adv *quits at the qth phase, then $w(T_q)/w(Z_q)$ tends to 4 as $m \to \infty$, $\epsilon \to 0$, and $K_1 \to 3$.*

Proof By definition, P_i consists of descendants of edges in P_{i-1}. This means that P_i and P_{i-1} are edge-disjoint. Therefore, it follows that $w(T_j) \geq \sum_{i=0}^{q} w(P_i) - \delta$, where δ is the sum of $w(m_e)$ overall edges e in $P_0, \ldots, P_{q-1}$ having m_e. We can upper bound δ by summing weight of one of all edges, i.e.,

$$\delta \leq \sum_{i\geq 1} \prod_{j=1}^{i-1} m^j \cdot \frac{(1+\epsilon)^i}{\prod_{j=1}^{i} m^j} = \sum_{i\geq 1} \left(\frac{1+\epsilon}{m}\right)^i < \frac{\frac{1+\epsilon}{m}}{1 - \frac{1+\epsilon}{m}} \to 0$$

as $m \to \infty$.

Since Adv quits at the qth phase, it follows that $w(P_i) \leq \gamma_i w(P_{i-1})$ for $1 \leq i < q$ and $w(P_q) > \gamma_q w(P_{q-1})$. Therefore, it follows that

$$\begin{aligned}\frac{w(T_q)}{w(Z_q)} &\to \frac{\sum_{i=0}^{q} w(P_i)}{w(Z_q)} \quad [m \to \infty]\\ &= \frac{\sum_{i=0}^{q-1} w(P_i) + w(P_q)}{(1+\epsilon)w(P_{q-1})} \quad \text{[by (1)]}\\ &> \frac{\sum_{i=0}^{q-1} \prod_{j=i}^{q-2} \gamma_{j+1}^{-1} w(P_{q-1})}{(1+\epsilon)w(P_{q-1})} + \frac{\gamma_{q-1}}{1+\epsilon}\\ &= \frac{1}{1+\epsilon}\left(\frac{\sum_{i=0}^{q-1} K_i}{K_{q-1}} + \frac{K_q}{K_{q-1}}\right) \quad \text{[by the definition of } \gamma_i]\\ &\geq \frac{K_0 + K_1}{1+\epsilon} \quad \text{[by Lemma 12]}\\ &\to 4. \quad [\epsilon \to 0, K_1 \to 3, K_0 = 1]\end{aligned}$$

□

Thus, we have the following theorem.

Theorem 14 *If a deterministic online Steiner tree algorithm is ρ-competitive on outerplanar graphs, then $\rho \geq 4$.*

References

[1] Angelopoulos, S.: On the Competitiveness of the Online Asymmetric and Euclidean Steiner Tree Problems, *WAOA 2009*, pp. 1–12 (2010).
[2] Averbuch, B., Azar, Y. and Bartal, Y.: On-Line Generalized Steiner Problem, *Theoret. Comput. Sci.*, Vol. 324, pp. 313–324 (2004).
[3] Bartal, Y., Fiat, A. and Rabani, Y.: Competitive Algorithms for Distributed Data Management, *J. Comput. Sys. Sci.*, Vol. 51, No. 3, pp. 341–358 (1995).
[4] Fleischner, H. J., Geller, D. P. and Harary, F.: Outerplanar Graphs and Weak Duals, *J. Indian Math. Soc.*, Vol. 38, pp. 215–219 (1974).
[5] Gupta, A., Newman, I., Rabinovich, Y. and Sinclair, A.: Cuts, Trees, and ℓ_1-Embedding of Graphs, *Combinatorica*, Vol. 24, No. 2, pp. 233–269 (2004).
[6] Imase, M. and Waxman, B. M.: Dynamic Steiner Tree Problem, *SIAM J. Discrete Math.*, Vol. 4, No. 3, pp. 369–384 (1991).
[7] Matsubayashi, A.: Online Steiner Trees on Outerplanar Graphs, *IPSJ SIG Technical Report*, Vol. 2014-AL-150, No. 8 (2014).
[8] Naor, J. S., Panigrahi, D. and Singh, M.: Online Node-Weighted Steiner Tree and Related Problems, *Proc. 52nd Annual Symposium on Foundations of Computer Science*, pp. 210–219 (2011).