

Computing the Degree-4 Shortest Network under a Given Topology*

Xu Yinfeng,¹ Ye Jichang,¹ and Binhai Zhu²

¹School of Management, Xi'an Jiaotong University,
Xi'an, People's Republic of China
yfxu@xjtu.edu.cn

²Department of Computer Science,
City University of Hong Kong, Hong Kong
bhz@cs.cityu.edu.hk
and
Laurentian University,
Ontario, Canada

Abstract. Let F be a set of n (n being an even number) fixed points and let M be a set of $n/2 - 1$ moving points, whose locations are to be determined, in the plane. A *topology* is a set of edges connecting these points in the set $V = F \cup M$. Let E be a full 4-degree Steiner topology; i.e., the topology forms a tree which contains fixed points of degree 1 and moving points of degree 4, and let $H(E)$ be a set of topologies which include E and its degeneracies. We define a *canonical* tree over V as one whose topology belongs to $H(E)$ and in which the sum of two adjacent angles around any node is not less than π . In this paper we prove that if a canonical tree exists, then it is the shortest (degree-4) network under a given topology E . We present an $O(n^2)$ time algorithm for finding a degree-4 shortest network whose topology belongs to $H(E)$.

1. Introduction

Designing optimal networks is a very important problem in many engineering fields, e.g., in developing and building oil fields. Usually the petroleum pumped from oil wells needs to be stored or processed in concentrators and later they are transported elsewhere. Suppose that there are n wells in an oil field, each is connected to a

* This research was supported by NSF of China, No. 19731001, a research grant from City University of Hong Kong, and a research grant from Laurentian University.

concentrator through a pipeline whose cost varies directly with its length and each concentrator has a *capacity* k , i.e., the number of pipelines connecting to a concentrator cannot exceed k . The problem is how to choose locations for m concentrators so as to obtain a connected network minimizing the total cost. To satisfy all these constraints we must build a minimal number of concentrators. Note that the concentrators can be built anywhere, including at the sites of the oil wells; furthermore, the concentrators can be built on the same sites. By simple calculation, n, m, k satisfy the relation $m = \lceil (n - 2)/(k - 2) \rceil$ where $\lceil x \rceil$ is the maximum integer less than x .

In practice, usually wells must connect to concentrators in some specified way. In general, given a set F of n fixed points, a set M of $\lceil (n - 2)/(k - 2) \rceil$ moving points in the Euclidean plane, and a set H_1 of topologies for $V = F \cup M$, each of which is a set of edges interconnecting (points in) V and satisfying that every fixed point is of degree 1 and every moving point is of degree no more than k , we consider the problem of how to choose locations for M such that the sum of edge lengths of the network, whose topology belongs to H_1 , is minimized. Throughout this problem, zero-length connections are allowed (which implies the collapsing of sites).

The general problem when M, H_1 are not given in advance and $k = 3$ is the famous Steiner problem in the Euclidean plane. Melzak [8] first established many basic properties of a shortest interconnecting network and gave a finite solution to the Steiner problem. Gilbert and Pollak [5] gave a thorough treatment of the Steiner problem and coined the name “Steiner minimal tree” (SMT) for the shortest interconnecting networks. The Steiner problem is an intrinsically difficult one, having been shown to be NP-complete [4]. Polynomial time approximation algorithms were studied by many researchers [2], [3], [7], [9], [12]. In addition to those approximation algorithms, many exact algorithms have also been developed [1], [10]. The difficulty of computing the SMT is due to the large number of possible connections, or topologies. That might be the reason why people study the Steiner problem when M, H_1 are given (i.e., under a given topology) in addition to the practical constraints we described in the previous paragraphs.

A full (degree-3) Steiner topology for a set of fixed and moving (Steiner) points is one in which every fixed point is of degree 1 and every Steiner point is of degree 3. Given a full degree-3 Steiner topology (with the Steiner or moving points), one may minimize the total edge length by arranging the moving points to their optimal locations. The corresponding network is called the (degree-3) shortest network under the given Steiner topology. Hwang and Weng [6] proposed a luminary algorithm to compute the shortest network under a given full degree-3 Steiner topology. The luminary algorithm has a worst time complexity of $O(n^2)$. Xue and Du [11] proved that this algorithm can be improved to have average time complexity $O(n \log n)$.

In this paper we follow the footsteps of Hwang and Weng to study the shortest network problem under a given full Steiner topology by considering the case for $k = 4$. Following the aforementioned relation on n, m, k we assume that n is an even number. We propose an $O(n^2)$ time algorithm, also called the luminary algorithm, to find a degree-4 shortest network under a given full Steiner topology, if one exists. In Section 2 we investigate the problem by proving some basic properties and in Section 3 we present the luminary algorithm.

2. The Degree-4 Shortest Network Problem

Let F be a set of n (n being an even number) fixed points and let M be a set of $n/2 - 1$ moving points in the plane. (For the convenience of description we sometimes call the points in F *terminal* points and the points in M *Steiner* points.) A topology for $V = F \cup M$ is a set of edges connecting points in V . A topology is called a degree-4 full Steiner topology (D4FST) if every fixed point is of degree 1 and every Steiner point is of degree 4. A topology E is a degeneracy of another topology E' if E can be obtained from E' by contracting edges, i.e., shrinking an edge and collapsing its two endpoints. Let D denote the set of all D4FSTs for V and their degeneracies. Let t be a tree, and let $T(t)$ and $l(t)$ be the topology and length of t , respectively. The problem can be formulated as $\min_{T(t) \in D} l(t)$ and we refer to this problem as the *degree-4 shortest network problem*.

A *canonical* tree t_c for a given point set $V = F \cup M$ is a tree network interconnecting V such that its topology belongs to D and the sum of two adjacent angles around every point of V is not less than π (which is referred to as the *angle condition*). For two given points a and b , $[a, b]$ denotes the line segment between a and b and $|ab|$ denotes the length of $[a, b]$.

We first consider the following base problem: for a set S of four points a, b, c, d in the Euclidean plane, find a point v such that $|av| + |bv| + |cv| + |dv|$ is minimal. We call v the optimal moving point for S . With triangle inequality, the following lemma is easy to prove.

Lemma 1. *If the convex hull of S , $CH(S)$, is a convex quadrilateral, then the intersection of two diagonals of $CH(S)$ is the optimal moving point. If $CH(S)$ is a triangle, then the optimal moving point is the point of S which is not a vertex of $CH(S)$. If $CH(S)$ is a line segment, then either one of the two points in S which are not the endpoints of $CH(S)$ is the optimal moving point for S .*

If $T(t^*) \in D$ and $l(t^*) = \min_{T(t) \in D} l(t)$, then we refer to t^* as a degree-4 shortest network (over D). We first prove the following fundamental theorem.

Theorem 1. *A degree-4 shortest network t^* whose topology belongs to D must be a canonical tree.*

Proof. Assume that Theorem 1 is false, then there must exist a point v of t^* which violates the angle condition.

Suppose that a_1, a_2 , and a_3 are the points of t^* which connect to v such that $\angle a_1va_2 + \angle a_2va_3 < \pi$. If v is a Steiner point with degree 4, then let a_4 be the fourth point connecting to v in t^* . Take a point s on $[a_2, v]$ such that $|a_1s| + |a_3s| < |a_1v| + |a_3v|$ (Fig. 1). By the triangle inequality, $|a_4s| < |a_4v| + |vs|$. Therefore $|a_1s| + |a_2s| + |a_3s| + |a_4s| < |a_1v| + |a_2v| + |a_3v| + |a_4v|$. By deleting the edges connecting to v as well as the point v and adding the point s and the edges $[a_1, s], [a_2, s], [a_3, s]$, and $[a_4, s]$, we obtain a new tree t whose length $l(t)$ is obviously less than $l(t^*)$ and $T(t)$ is identical to $T(t^*)$. This contradicts the minimality of $l(t^*)$. In the same way, we can obtain the proof when v is a fixed point. □

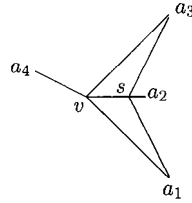


Fig. 1. Illustration for the proof of Theorem 1.

Theorem 1 shows that a degree-4 shortest network under a given topology is a canonical tree, this enlightens us to look for a canonical tree. Let E be a D4FST, let $H(E)$ be a set of topologies which includes E and its degeneracies, and let $H^*(E) \subseteq H(E)$ denote a subset of D4FSTs. The following theorem shows that if a canonical tree exists, then it is the degree-4 shortest network under a given topology.

Theorem 2. *Let E be a D4FST. If there exists a canonical tree whose topology is in $H(E)$, then the tree must be a shortest network for E .*

Proof. We prove this theorem by induction on even number n , which is the number of fixed points in E . For $n = 4$, the theorem is trivially true following Lemma 1.

Assume that the theorem holds for $n - 2$ where n is an even number. Let t_1 be a canonical tree whose topology $T(t_1)$ is in $H^*(E)$ and let the size of t_1 be n . Suppose that there exists a shorter tree t_2 whose topology is in $H(E)$. Let v_1, v_2 , and v_3 be the three fixed points adjacent to the same Steiner point s in E ; furthermore, let s' be the Steiner point adjacent to s . Let p_1 and p'_1 (p_2 and p'_2) denote the locations of s and s' in t_1 (t_2), respectively.

First, consider the case $p_1 \neq v_1, v_2, v_3$. Let $t_i^*, i = 1, 2$, be a tree obtained from t_i by deleting the edges adjacent to p_i as well as the points v_1, v_3, p_i and adding the edge $[p'_i, v_2]$. Note that t_i^* is a canonical tree for the set $V^* = V - \{v_1, v_3, s\}$ with $n - 2$ fixed points and following the induction hypothesis t_i^* is the corresponding degree-4 shortest network for V^* . Clearly we have $l(t_i^*) = l(t_1) - |v_1 v_3|$. In the remainder of this paragraph we show three subcases which all lead to a contradiction to the above induction hypothesis. These subcases are illustrated in Fig. 2. If p_2 does not collapse into v_1, v_2 , or v_3 , then $l(t_2^*) = l(t_2) - |v_1 v_3|$ as shown in Fig. 2(a). If p_2 collapses into

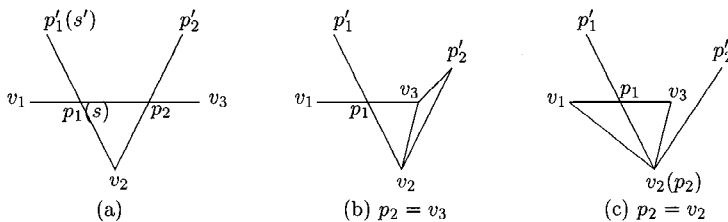


Fig. 2. Illustration for the proof of Theorem 2, case 1.

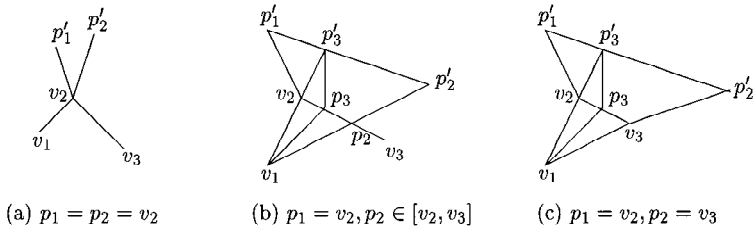


Fig. 3. Illustration for the proof of Theorem 2, case 2.

either v_1 or v_3 , without loss of generality, assume that p_2 collapses into v_3 as shown in Fig. 2(b), then by triangle inequality $l(t_2^*) = l(t_2) - |v_1 v_3| - |v_2 v_3| - |v_3 p_2'| + |v_2 p_2'| \leq l(t_2) - |v_1 v_3|$. If p_2 collapses into v_2 as shown in Fig. 2(c), then again by triangle inequality $l(t_2^*) = l(t_2) - |v_1 v_2| - |v_2 v_3| \leq l(t_2) - |v_1 v_3|$. Combining these three subcases we have $l(t_1^*) = l(t_1) - |v_1 v_3| > l(t_2) - |v_1 v_3| \geq l(t_2^*)$, i.e., $l(t_1^*) > l(t_2^*)$. On the other hand, let E^* be a D4FST for $V^* = V - \{v_1, v_3, s\}$ obtained from E by deleting the edges adjacent to s and the points v_1, v_3 , and s , and adding the edge $[v_2, s']$, we have $T(t_1^*) \in H^*(E^*)$, $T(t_2^*) \in H(E^*)$. This contradicts the induction hypothesis that $T(t_1^*)$ is a D4FST and t_1^* is the corresponding degree-4 shortest network.

Secondly, we consider the case when p_1 collapses into one of v_1, v_2 , and v_3 . Without loss of generality, we consider the case $p_1 = v_2$. (When $p_1 = v_1$ or $p_1 = v_3$, we can obtain similar proofs.) Similar to the previous paragraph we want to show that a shorter tree t_2 does not exist. Again we have three subcases as illustrated in Fig. 3. If $p_2 = v_2$ (Fig. 3(a)) then let t_i^* , $i = 1, 2$, be the tree obtained from t_i by deleting the edges $[v_1, v_2], [v_2, v_3]$ and the points v_1, v_3 . Note that t_1^* is a canonical tree for the set $V^* = V - \{v_1, v_3\}$ with $n - 2$ fixed points and following the induction hypothesis t_1^* is the corresponding degree-4 shortest network for V^* . However, in this subcase, $l(t_1^*) = l(t_1) - |v_1 v_2| - |v_1 v_3| > l(t_2) - |v_1 v_2| - |v_1 v_3| = l(t_2^*)$. As $T(t_1^*) \in H^*(E^*)$, $T(t_2^*) \in H(E^*)$ this contradicts the induction hypothesis that $l(t_1^*) \leq l(t_2^*)$.

We now consider two other subcases for case 2 and it turns out that we need some arguments which are different from those we have used. Without loss of generality, we assume that p_2 is either located inside the edge $[v_2, v_3]$ (Fig. 3(b)) or collapses into v_3 (Fig. 3(c)). When p_2 is located on the edge $[v_2, v_3]$, let the extended line of $[v_1, v_2]$ intersect $[p_1', p_2']$ at p_3' . Define $\alpha = |p_1' p_3'| / |p_1' p_2'|$ and let p_3 be a point on $[p_1, p_2]$ such that $|p_1 p_3| = \alpha |p_1 p_2|$. Assume that V_1 and V_2 denote the vertex sets of t_1 and t_2 , including the points collapsed into terminal points, we construct a Steiner tree t_3 with the vertex set $V_3 = \alpha V_1 + (1 - \alpha)V_2 = \{\alpha u_1 + (1 - \alpha)u_2 | u_1 \in V_1 \text{ and } u_2 \in V_2 \text{ correspond to the same vertex in } E\}$ (see Fig. 3). Then we have $T(t_3) \in H(E)$. Since the tree length is a convex function, $l(t_2) < l(t_3) < l(t_1)$ holds. Let t_3' be the tree obtained from t_3 by substituting $[v_2, p_3']$, $[v_1, v_2]$, and $[v_2, v_3]$ for the edges adjacent to p_3 (i.e., $[v_1, p_3], [v_2, p_3], [v_3, p_3]$, and $[p_3, p_3']$), then $T(t_3') \in H(E)$. Following triangle inequality $|v_2 p_3'| + |v_1 v_2| + |v_2 v_3| < |v_1 p_3| + |v_2 p_3| + |v_3 p_3| + |p_3 p_3'|$, hence $l(t_3') < l(t_3) < l(t_1)$. Consequently, if we delete v_1, v_2 , and p_3 from t_1 and t_3' we have a tree $t_3' - \{v_1, v_2\}$ shorter than $t_1 - \{v_1, v_2\}$; moreover, both with $n - 2$ fixed points. This contradicts the induction hypothesis that $t_1 - \{v_1, v_2\}$ is the shortest network for $n - 2$

fixed points under the given topology. (Notice that $t_1 - \{v_1, v_2\}$ is a canonical tree as $p_1 = v_2$.) We can use similar arguments to prove the subcase for $p_2 = v_3$ (Fig. 3(c)). We leave this as an exercise for the reader. \square

Theorems 1 and 2 basically show that a degree-4 shortest network under a given topology exists if and only if a canonical tree under the same topology exists. To obtain an efficient algorithm for computing a degree-4 shortest network (under a given topology) we still need to show some extra property of a canonical tree (under a given topology). The following theorem shows that if a degree-4 canonical tree under a D4FST exists, then it must be unique. This is important in helping us developing an efficient algorithm.

Theorem 3. *If E is a D4FST, then there exists at most one canonical tree whose topology belongs to $H(E)$.*

Proof. Again we prove this theorem by induction on even number n , which is the number of fixed points in E . For $n = 4$, the theorem is trivially true by Lemma 1.

Suppose to the contrary that there exist two canonical trees t_1 and t_2 with size n whose topologies $T(t_1)$ and $T(t_2)$ belong to $H^*(E)$. Let v_1, v_2 , and v_3 be three fixed points adjacent to the same Steiner point s in E , and let s' be the Steiner point adjacent to s . Denote $s(s')$ by $p_i(p'_i)$ in t_i for $i = 1, 2$ (see Fig. 2). If p_1 and p_2 are not on the line segment $[v_1, v_2]$ ($[v_2, v_3]$ or $[v_1, v_3]$), then by the angle condition t_i 's cannot be canonical. Without loss of generality, we assume that both p_1 and p_2 are on $[v_1, v_3]$.

If $p_1 = p_2$ or both p_1 and p_2 are inside $[v_1, v_3]$, then let t_i^* , $i = 1, 2$, denote the tree obtained from t_i by deleting the edges $[v_1, p_i]$ and $[v_3, p_i]$ and the point p_i . Let E^* be the D4FST from E by deleting the edges adjacent to s and adding the edge $[p'_1, v_2]$, then $T(t_1^*) \in H^*(E^*)$, $T(t_2^*) \in H^*(E^*)$. By the induction hypothesis, $t_1^* = t_2^*$ as they have the same number of fixed points, $n - 2$. It follows that $t_1 = t_2$, which contradicts the assumption that $t_1 \neq t_2$.

If $p_2 = v_3$ and either p_1 is inside $[v_1, v_3]$ (see Fig. 4(a)) or $p_1 = v_1$ (see Fig. 4(b)), then we define V_1 (V_2) as the vertex set of t_1 (t_2), including the collapsed points. Let $t_3(\beta)$ be a tree on $V_3 = \beta V_1 + (1 - \beta)V_2$, where $0 < \beta < 1$, then $T(t_3(\beta)) \in H(E)$. By Theorem 2, it follows that $l(t_1) = l(t_2) = l(t_3(\beta))$. This implies that $l(t_3(\beta))$ is also a shortest network. Let $p_3 = \beta p_1 + (1 - \beta)p_2$ and $p'_3 = \beta p'_1 + (1 - \beta)p'_2$. If there exists some β , $0 < \beta < 1$, such that v_2, p_3 , and p'_3 are not collinear, then we

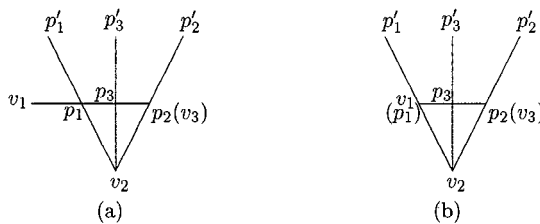


Fig. 4. Illustration for the proof of Theorem 3.

can easily shorten $t_3(\beta)$ by moving p_3 or p'_3 , this contradicts the minimality of $l(t_3(\beta))$. Therefore, the three points v_2, p_3 , and p'_3 are collinear for any $0 < \beta < 1$. Consequently v_2, p_1 , and p'_1 are collinear, so are v_2, v_3 , and p'_2 . Let t_1^*, t_2^* , and E^* be defined as above. Following the induction hypothesis, $t_1^* = t_2^*$, which implies that if we delete $[v_1, p_i]$, $[v_3, p_i]$, and the point p_i from t_i , then p'_1 coincides with p'_2 . Combining this with the collinearity of $v_2, p_1(v_1), p'_1$ and $v_2, p_2(v_3), p'_2$, we have $t_1 = t_2$ and this contradicts the assumption. \square

3. A Luminary Algorithm for Constructing a Canonical Tree

3.1. Some Concepts and Definitions

In the previous section we prove some important theorems for the degree-4 shortest network problem. These theorems let us focus on the construction of a canonical tree (under a given topology). The construction of a canonical tree in $H(E)$ can be viewed as orienting the edges of E with a given set of V , since the intersections of those edges determine locations of the moving points. It is convenient to consider an edge as directed and call a possible orientation a ray. A ray r has a *base* $b(r)$ which is the starting point of that (directed) edge and an angle $\theta(r)$ which measures the angle from the horizontal line through $b(r)$ to r in the counterclockwise direction. The other half-line $e(r)$ of r , which also includes $b(r)$, is called the *extension* of r . Two rays r_1 and r_2 are said to *e-intersect* if $e(r_1)$ intersects $e(r_2)$. An entity which radiates rays and is associated with an edge of E is called a *luminary*. Two luminaries are said be *adjacent* at s if their associated edges are both incident to s in E . A pair of rays, one from each luminary, which run into each other from opposite directions is called an *opposing* pair.

A *bundle* B is either a single ray or a set of rays satisfying the following conditions:

- (1) B consists of a continuous set of rays with a *span* $\theta(B) \leq \pi$. B includes either none or one of its two boundary rays.
- (2) All rays of B e-intersect at the same point $t(B)$, which is called the *tip* of B .
- (3) The set $b(B) = \{b(r) | r \in B\}$, which is called the *baseline* of B , is a straight line segment or a single point.

Two bundles are said to be adjacent if their angles neither overlap nor leave a gap and their baselines form a straight line segment. A *multibundle* $M(B_1, \dots, B_m)$ is either a bundle ($m = 1$) or a sequence of $m \geq 2$ adjacent bundles such that $\sum_{i=1}^m \theta(B_i)$, which is called the *angle* of M , does not exceed π and any two rays of M e-intersect each other (Fig. 5).

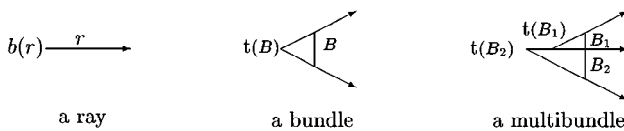


Fig. 5. A ray, a bundle, and a multibundle.

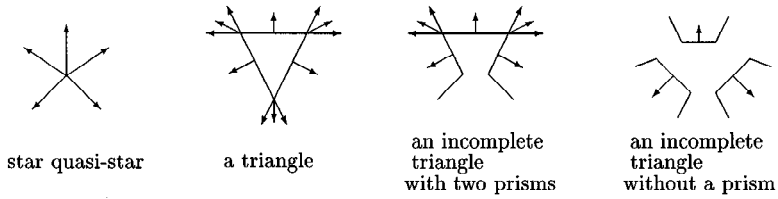


Fig. 6. Examples for star, triangle, and incomplete triangle.

An edge is a *terminal* edge if it is incident to a terminal point. Using the angle condition of a canonical tree, which implies that the sum of two adjacent angles around any point in the tree is at least π , we can show that the degree of the (moving) vertices in a canonical tree does not exceed 4. Hence an edge can shrink without introducing a vertex whose degree is more than 4 if and only if it is a terminal edge. Suppose that a luminary L has rays in all directions and its baseline $b(L) = \{b(r) | r \in L\}$ is a point, then L is called a *star* if the edge associated with L is a terminal edge, otherwise L is called a *quasi-star*. A luminary L is called a *triangle* (an *incomplete triangle*) if $b(L)$ is a triangle (L consists of some multibundles) (see Fig. 6).

3.2. The Luminary Algorithm

Now we describe the algorithm for constructing a canonical tree. The luminary algorithm constructs a canonical tree under a given D4FST. If a canonical tree does not exist for the given topology, then the algorithm returns empty. The luminary algorithm consists of a merging stage and an orienting stage. At each step of the merging stage, three luminaries L_1 , L_2 , and L_3 adjacent at a Steiner point s are merged into a new luminary L_4 associated with the fourth edge of s . The details of the process of generating L_4 are as follows.

First we find the opposing pair r_{ij} and r_{ji} for L_i and L_j , where $r_{ij} \in L_i, r_{ji} \in L_j, i < j, i, j = 1, 2, 3$. If r_{ij} and r_{ji} exist, and there exists a ray r in L_k ($k \neq i, j, 1 \leq k \leq 3$) such that r intersects the line segment $[b(r_{ij}), b(r_{ji})]$ at p and does not intersect $b(L_i)$ and $b(L_j)$, then r can be modified into the ray r' in L_4 by substituting the point p for the base of r , i.e., $b(r') = p$. This merge is called a *Steiner merge* (Fig. 7). If all of three opposing pairs exist and one of the three luminaries is a star, assuming that is L_1 , and r_{12} intersects with r_{13} and their directions are opposite, then L_4 is a quasi-star with the baseline $b(L_1)$. We call r_{21} and r_{31} the axes of the quasi-star and the corresponding merge a *base merge*. If one of L_1, L_2 , and L_3 is a star, assuming that is L_1 , and two opposing pairs r_{12}, r_{21} and r_{13}, r_{31} exist, then the angular area generated by r_{21} and r_{31} is a subset of L_4 . We call this angular area a *prism* and r_{21} and r_{31} the *prism handles*. This merge is called a *star merge*. Both base merge and star merge can be thought of as special cases for Steiner merge.

At the beginning of the merging stage, there are n stars associated with the n terminal edges, each merging step reduces the number of luminaries by two. The merging stage

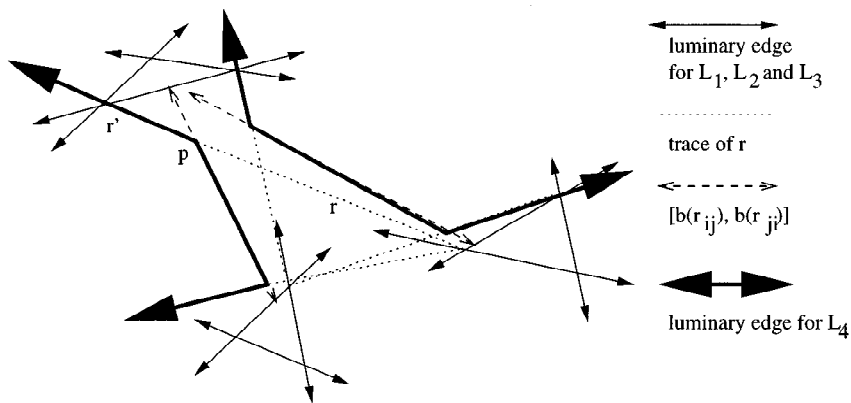


Fig. 7. A Steiner merge.

ends after $n/2 - 1$ steps when only two luminaries are left. For convenience, we require that neither of the remaining two luminaries be a star. This can be easily achieved, since at the $(n/2 - 1)$ th step, at least one of the four luminaries is not a star and we can merge the other three luminaries to satisfy the condition.

When only two luminaries are left, we enter the orienting stage. (This stage is very much the same as in [6].) Note that there are $\frac{3}{2}n - 2$ edges in E but each merging step retires three of them. Therefore at the end of the merging stage there is only one edge unretired which must be associated with both luminaries left, from different ends of the edge. We can orient the edge by finding the opposing pair, the segment between the two bases of the opposing pair of rays, and then identifying and orienting the edge, if the opposing pair exists. Otherwise, we claim that there does not exist a canonical tree in $H(E)$. Starting from this edge, we sequentially track back the rays which generated the previously oriented rays, so that these retraced rays also become oriented. Eventually, a set of $\frac{3}{2}n - 2$ edges are oriented which constitute the canonical tree. We have the following theorem.

Theorem 4. *The luminary algorithm outputs a canonical tree if and only if one exists in $H(E)$.*

Proof. For the three types of merging, it is clear that the merging stage preserves the given topology in $H(E)$. At the end of the merging stage we have two luminaries L and L' associated with the same edge e . By the assumption in the previous paragraphs, the last two luminaries cannot be stars. Therefore e is not a terminal edge. If e shrinks or an opposing pair cannot be found at the beginning of the orienting stage, then there is no canonical tree in $H(E)$.

As we have commented before, once an opposing pair is found, the remaining orienting stage is automatic and must yield a canonical tree. This implies that each opposing pair is unique, since the canonical tree in $H(E)$ is unique. □

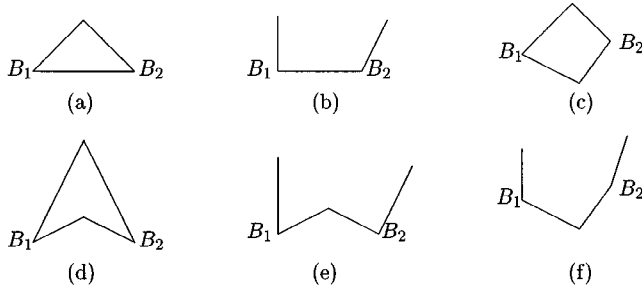


Fig. 8. Taking forms of B_1 and B_2 .

4. An Analysis of the Luminary Algorithm

In this section we present a detailed analysis of the luminary algorithm. Let B_1 and B_2 be two bundles. If the boundary rays of B_1 and B_2 form the shapes as shown in Fig. 8, then we refer to B_1 and B_2 as a taking form and call (a), (b), (c), (f) in Fig. 8 convex, and (d), (e) in Fig. 8 concave. Otherwise, we say that B_1 and B_2 is a nontaking form.

The following lemma is trivial and the proof is omitted.

Lemma 2. *There exists an opposing pair between two bundles B_1 and B_2 if and only if they are of a taking form and the shape of the taking form is convex.*

Secondly, we discuss the time complexity for finding an opposing pair between two multibundles.

Lemma 3. *Determining an opposing pair between two multibundles $M(B_1, \dots, B_p)$ and $M(B'_1, \dots, B'_q)$ or showing the nonexistence of an opposing pair can be done in $O(p + q)$ time.*

Proof. Without loss of generality, assume that the two multibundles $M(B_1, \dots, B_p)$ and $M(B'_1, \dots, B'_q)$ are shown as in Fig. 9(a). If B_1 and B'_1 is of a taking form and the shape of this taking form is convex, then by Lemma 2 we can obtain an opposing pair in $O(1)$ time by connecting $t(B_1)$ and $t(B'_1)$. If B_1 and B'_1 is of a taking form and the shape of this taking form is concave or B_1 and B'_1 is of a nontaking form, then no opposing pair between B_1 and B'_1 exists (Fig. 9(b)). Therefore, in the latter case an opposing pair

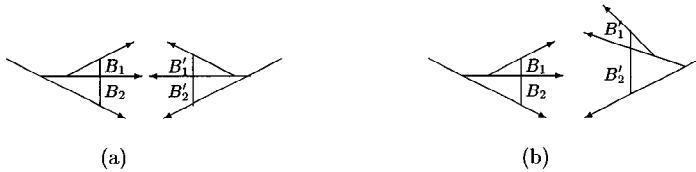


Fig. 9. Illustration for the proof of Lemma 3.

between $M(B_1, \dots, B_p)$ and $M(B'_1, \dots, B'_q)$ exists if and only if the opposing pair is between $M(B_2, \dots, B_p)$ and $M(B'_2, \dots, B'_q)$. We can repeat this procedure recursively. Therefore, we can obtain an opposing pair or show its nonexistence in at most $O(p + q)$ steps. \square

In Fig. 9(b) B_1 and B'_1 is of a nontaking form (as the common boundary of B_1 and B_2 does not intersect the boundary of B'_1); therefore, to find the opposing pair between $M(B_1, \dots, B_p)$ and $M(B'_1, \dots, B'_q)$ or to show its nonexistence we only need to search for the opposing pair or to show its nonexistence between $M(B_2, \dots, B_p)$ and $M(B'_2, \dots, B'_q)$.

Thirdly, we discuss the time complexity of merging three luminaries L_1, L_2 , and L_3 into a luminary L_4 with a number of bundles. Suppose that the number of bundles in $L_i, i = 1, 2, 3$, is l_i .

The merge of L_1, L_2 , and L_3 is nothing but a base merge, a star merge, or a Steiner merge. No matter which merge is performed, we need to find an opposing pair. By Lemma 3, determining the opposing pairs takes $O(l_1 + l_2 + l_3)$ time. Assume we have found the opposing pairs, we now discuss the three types of merges for the given opposing pairs. For a base merge, we know that the merge takes constant time and generates a quasi-star which consists of two bundles. A star merge yields a prism which can be determined by its two supporting handles, hence this merge can also be completed in constant time. For a Steiner merge, we first merge the given opposing pairs with a bundle, a star, or a quasi-star. By the definition of the Steiner merge, each of these takes constant time and generates at most one bundle. As in L_1, L_2 , and L_3 we have l_1, l_2 , and l_3 bundles, respectively, merging L_1, L_2 , and L_3 with the opposing given pairs takes at most $O(l_1 + l_2 + l_3)$ time and the resulting L_4 contains at most $l_1 + l_2 + l_3$ bundles.

Finally, we summarize the complexity of our algorithm. The algorithm starts with n stars, hence $2n$ bundles. We know that the total number of bundles does not increase at each merging step, hence L contains at most $2n$ bundles, all the time. Thus we conclude that each merging step takes at most $O(n)$ time. Since there are $n/2 - 1$ such steps, the merging stage takes a total of $O(n^2)$ time.

At the first orienting step we seek an opposing pair of rays and it takes at most $O(n)$ time to find such a pair or claim its nonexistence following the above discussion. In the remaining orienting stage we need to identify the three rays which generate a given ray at most $n/2 - 1$ times. Now all we need to show is that each orienting step takes constant time. If the base of a ray $r, b(r)$, is on a fixed point, then the bundle B containing the ray r is either a prism or a quasi-star. (The case when the bundle is a star will not occur because the ray r may not be generated by other rays in this case.) The two supporting handles of the prism or the two axes of the quasi-star are two of the three rays which generate r , and the edge with which the third ray is associated degenerates into a point. If $b(r)$ is on a Steiner point, then the baseline of $B, b(B)$, is associated with two fixed points u and v , and $[u, b(r)]$ and $[v, b(r)]$ are two of the three rays generating r . Suppose that the third ray is in B_3 , then $[t(B_3), b(r)]$ is the third ray. Since the tip and baseline of each bundle will be stored during the construction of the bundles each canonical tree edge can be oriented in constant time. Therefore, the orienting stage takes at most $O(n)$ time.

Therefore, we have the following theorem:

Theorem 5. *The luminary algorithm for constructing a canonical tree under a given full Steiner topology takes $O(n^2)$ time.*

An interesting question is whether we can apply an even more detailed analysis to show the average time complexity of the luminary algorithm, like in [11].

References

1. E. J. Cockayne and D. E. Hewgill, Exact computation of Steiner minimal tree in the Euclidean plane, *Inform. Process. Lett.*, 22 (1986), 151–156.
2. D. Z. Du and F. K. Hwang, A proof of Gilbert–Pollak conjecture on the Steiner ratio, *Algorithmica*, 7 (1992), 121–136.
3. D. Z. Du and Y. J. Zhang, On better heuristic for Steiner minimum trees, *Math. Programming*, 57 (1992), 193–202.
4. M. R. Garey, R. L. Graham, and D. S. Johnson, The complexity of computing Steiner minimal trees, *SIAM J. Appl. Math.*, 32 (1977), 835–859.
5. E. N. Gilbert and H. D. Pollak, Steiner minimal tree, *SIAM J. Appl. Math.*, 16 (1968), 1–29.
6. F. K. Hwang and J. F. Weng, The shortest network under a given topology, *J. Algorithms*, 13 (1992), 468–488.
7. M. Karpinski and A. Zelikovsky, New approximation algorithms for the Steiner tree problems, *J. Combin. Optim.*, 1 (1997), 47–65.
8. Z. A. Melzak, On the problem of Steiner, *Canad. Math. Bull.*, 4 (1961), 143–148.
9. D. Trietsch and F. K. Hwang, An improved algorithm for Steiner trees, *SIAM J. Appl. Math.*, 50 (1990), 244–263.
10. P. Winter, An algorithm for the Steiner problem in the Euclidean plane, *Networks*, 15 (1985), 323–345.
11. G. L. Xue and D. Z. Du, $O(n \log n)$ -average-time algorithm for shortest network under a given topology, In: *Computing and Combinatorics*, LNCS 1090, pp. 11–20, Springer-Verlag, Berlin, 1996.
12. A. Zelikovsky, An $\frac{11}{6}$ -approximation algorithm for the network Steiner problem, *Algorithmica*, 9 (1993), 463–470.

Received November 16, 1998.