Short Trees in Polygons

Pawel Winter, Martin Zachariasen, Jens Nielsen
Dept. of Computer Science, University of Copenhagen
Universitetsparken 1, DK-2100 Copenhagen Ø, Denmark
e-mail: {pawel,martinz,jensn}@diku.dk

Abstract

We consider the problem of determining a short Euclidean tree spanning a number of terminals in a simple polygon. First of all, linear time (in the number of vertices of the polygon) exact algorithms for this problem with three and four terminals are given. Next, these algorithms are used in a fast polynomial heuristic based on the concatenation of trees for appropriately selected subsets with up to four terminals. Computational results indicate that the solutions obtained are close to optimal solutions.

1 Introduction

We consider the following variant of the *Euclidean Steiner tree problem* (ESTP):

- **Given:** A simple polygon P with k vertices and a set Z of n terminals in P.
- **Find:** *Euclidean Steiner minimal tree* (ESMT) spanning the terminals and being completely in P.

This problem is a generalization of the ESTP *without* obstacles. It is more realistic than the obstacle-free version, and therefore will hopefully have more real-life applications in network design (Fig. 1). Furthermore, the techniques described in this paper can be used to solve the rectilinear Steiner tree problem with obstacles which has many important applications in VLSI-design.

ESMTs in the plane and with no obstructing polygon tend to consist of unions of ESMTs with very few terminals, each of degree 1. It is unusual to encounter (in randomly generated problem instances) ESMTs with 6 or more terminals [9]. Consequently, concatenation of small ESMTs (spanning subsets of up to 4 terminals) proved to yield good quality solutions for the obstacle-free case [6, 1]. Similar approach seems to be applicable when the terminals are inside a simple polygon without or with polygonal holes [13].

The problem of determining reasonable subsets of 2, 3 and 4 terminals inside a simple polygon (such that they are likely to appear in a small ESMT of the overall ESMT) is far from trivial. One approach is to use the geometric dual of the geodesic Voronoi diagram for all terminals inside P. Papadopoulou and Lee [3] gave an $O(m \log m)$ algorithm for this problem, where $m = k + n$. A small subset of terminals is then considered as a reasonable cluster if the subgraph of the dual induced by these terminals is connected. Alternatively, the Euclidean minimum spanning tree (EMST) for Z inside P can be used to select subsets. More specifically, subsets of terminals inducing connected subgraphs of the EMST are selected. Note that subsets of size 2 are identified by the edges of the EMST (edges represent geodesic paths between terminals in P).

The ESMT for 3 terminals in a simple polygon can be determined in $O(k)$ time and space [11]. In [12], we gave an $O(k \log k)$ time and $O(k)$ space algorithm for the determination of ESMTs for four terminals inside a simple polygon. In this paper we give a new algorithm for the four terminals problem requiring $O(k)$ time and space. We also give an overall description of the heuristic, provide some computational results, and compare them to exact solutions.

Once the ESMTs for subsets with up to 4 terminals have been determined, their concatenation can be carried out in several ways. The simplest is to place ESMTs on a priority queue ordered by

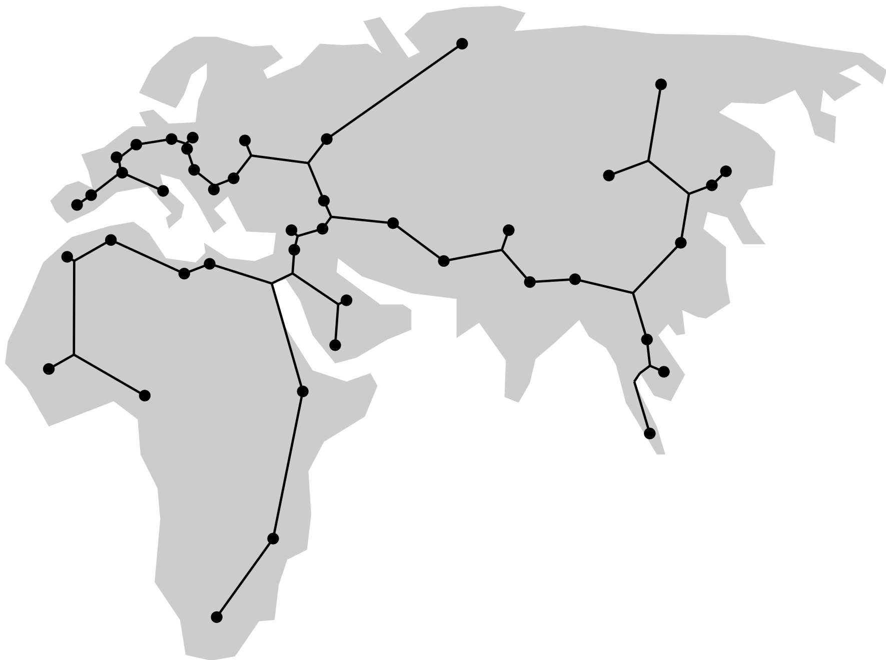


Figure 1: ESMT for selected places in Europe, Asia and Africa

increasing ratio between their lengths and the lengths of the corresponding EMSTs. Alternatively, the concatenation problem can be formulated as the NP-hard problem of finding a minimum spanning tree of an appropriately defined hypergraph. This problem can be cast as an integer programming problem. A branch-and-cut method suggested by Warme [8] can solve problem instances with several thousands of ESMTs in a reasonable amount of time.

The paper is organized as follows. Basic definitions are given in Section 2; however, the reader is referred to [2] for basic definitions and properties of ESMTs. The problems of determining ESMTs of three and four terminals in arbitrary polygons are reduced in Section 3 to the ESTP for up to four *semi-terminals* in smaller polygons of a very particular shape. The semi-terminals of the reduced problems need not to be identical with the original terminals. The linear time algorithm for the ESMT for three semi-terminals in the reduced polygon is given in Section 4. The linear time algorithm for ESMT with four semi-terminals is described in Sections 5 and 6. Our heuristic is described in Section 7. Computational results are given in Section 8. Conclusions and suggestions for further research are collected in Section 9.

2 Basic Definitions

A polygon P is defined as a closed polygonal chain. It is *simple* if it is not self-intersecting and its interior $i(P)$ is not empty and connected. A point p is said to be in P if $p \in i(P) \cup P$. A vertex v on P is *convex* if its interior angle is less than 180^o. Otherwise, it is *reflex*. A reflex vertex is said to be *wide* if its interior angle is at least 240^o (as will be explained below, three edges of an ESMT can meet on the boundary only if the angle is 240^o or more). Clockwise successor and predecessor vertices of a vertex v are denoted by v^+ and v^-, respectively. In order to simplify some proofs, it is assumed that v^-v and vv^+ are not colinear for any $v \in P$.

A simple polygon is called a *c-kite* iff precisely c of its vertices are convex. Boundaries of a c-kite P between two consecutive convex vertices are referred to as *sides* of P. A polygon P is *weakly-simple* if it is not self-intersecting. In particular, a weakly-simple polygon can have empty or disconnected interior.

The shortest path between two points u and v in a polygon P will be denoted by $P(u,v)$. $P(u,v)$ is a unique polygonal chain and its interior vertices are reflex vertices of P.

A line L is said to be an *interior tangent* of a c-kite P at a *touch vertex* $v \in P$ iff one of the following cases occurs.

- v is a reflex vertex, and the edges v^-v and vv^+ are on the same side of L.
- v is a convex vertex, and the edges v^-v and vv^+ are on the opposite sides of L.
- v^-v overlaps with L.

An interior tangent L with a touch-point v is *oriented* in such a way that the edge vv^+ is on its left. Two interior tangents of a c-kite P are distinct if they have different slopes or different touch vertices. We use the notation $L_i||L_j$ if interior tangents L_i and L_j are parallel. Similar notation is used for edges.

Lemma 1 *Every c-kite P, $c \geq 3$, has exactly $c-2$ interior tangents for any fixed slope.*

Proof. Every triangulation of a simple polygon has $k-2$ triangles. Each triangle contributes to the total sum of interior angles by π. The sum of interior angles of any simple polygon with k vertices is therefore $(k-2)\pi$. Let α_i, $1 \leq i \leq c$, denote interior angles of convex vertices. Let $\gamma_j = \pi + \beta_j$, $1 \leq j \leq k-c$, denote interior angles of reflex vertices. Then

$$\sum_{i=1}^{c} \alpha_i + \sum_{j=1}^{k-c} \beta_j = (k-2)\pi - (k-c)\pi = (c-2)\pi$$

Angles α_i and β_j denote maximal rotation of interior tangents at convex and reflex vertices, respectively. Furthermore, the slope interval at a particular vertex does not overlap but has a common boundary with the slope interval of next vertex on the polygon (Fig. 2). ■

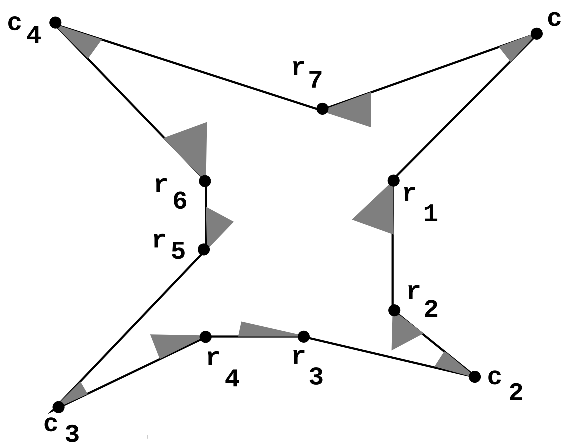
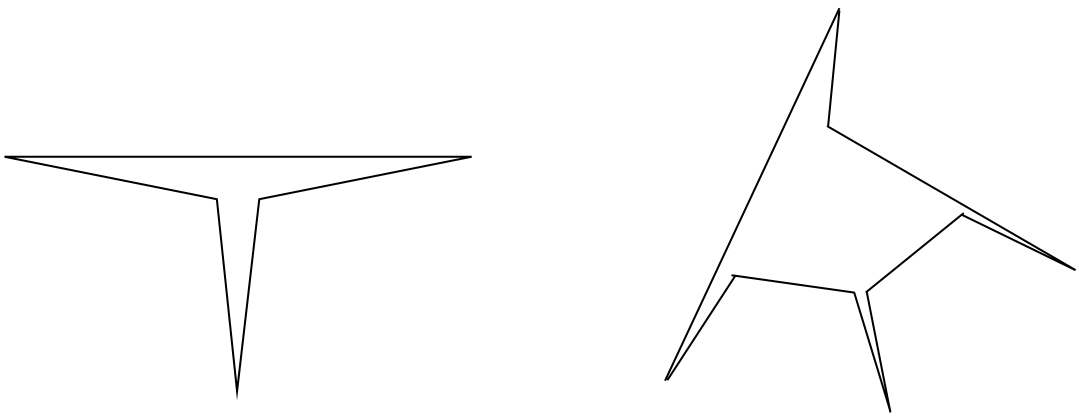


Figure 2: Interior angles of a 4-kite

Lemma 2 *A c-kite has at most $3c-7$ wide reflex vertices.*

Proof. A wide reflex vertex is a touch vertex of interior tangents whose slopes span over at least 60^o. The upper bound follows immediately from Lemma 1 where the sum over all rotation angles is shown to be $(c-2)\pi$. A 3-kite with 2 wide reflex vertices is shown in Fig. 3a. A 4-kite with 5 wide reflex vertices is shown in Fig. 3b. ■



(a) (b)

Figure 3: 3- and 4-kites with maximum number of wide reflex vertices

Consider a reflex vertex v of a c-kite P. Let q_v^- and q_v^+ denote the convex end-vertices of the side containing v. Let sv denote an edge in P overlapping with an interior tangent of v. Only one of the vertices v^- and v^+ is visible from s. Let q_v^s denote the convex vertex that can be reached from v by moving counterclockwise on P if v^- is invisible from s, and by moving clockwise if v^+ is invisible from s. If v is convex, let $q_v^s = v$.

An ESMT inside a simple polygon cannot have vertices of degree greater than three. Vertices of degree 3 are called *Steiner points* if they are located in the interior of P. The edges incident to Steiner points make 120^o with each other. They are called *degenerate Steiner points* if they are located on the boundary of P. Degenerate Steiner points can only occur at wide reflex vertices of P.

3 Polygon Reductions

Consider the unique polygon P' inside P containing the terminals Z such that its perimeter is as short as possible. Provan [5] proved that there always exists an ESMT for Z in P completely in P'. Toussaint [7] gave an $O(n(\log n + \log k) + k)$ algorithm to determine P'. The complexity of this algorithm reduces to $O(k)$ if n is fixed. P' is sometimes referred to as the *geodesic convex hull* for its polygon P and its terminals Z. It is denoted by $GCH(P, Z)$.

3.1 Three Terminals

Consider the set T of three terminals t_1, t_2, t_3 inside the simple polygon P. We show that the problem can be reduced to the ESTP in a 3-kite for its convex vertices (Fig. 4a).

Let $P' = GCH(P, Z)$. All terminals are on the boundary of P'. If $i(P') = \emptyset$, then the ESMT for T in P' is trivially given. We assume therefore in the following that $i(P') \neq \emptyset$.

Consider the two shortest paths from a terminal t_u, $u = 1, 2, 3$, to the remaining two terminals. Let q_u denote the last common vertex on these two paths. Note that q_u is well-defined; there is at least one common vertex, namely t_u. Consider the polygon P'' obtained from P' by cutting off $P'(q_u, t_u) \cup P'(t_u, q_u)$, $u = 1, 2, 3$. P'' can be obtained from P' in $O(k)$ time and space by a straightforward traversal of P' (using a stack). Note that $i(P'') \neq \emptyset$ and that p'' is a 3-kite. Let $Q = \{q_1, q_2, q_3\}$. Once the ESMT for *semi-terminals* in Q is determined, the ESMT for T is obtained by adding the paths $P'(t_u, q_u)$, $u = 1, 2, 3$.

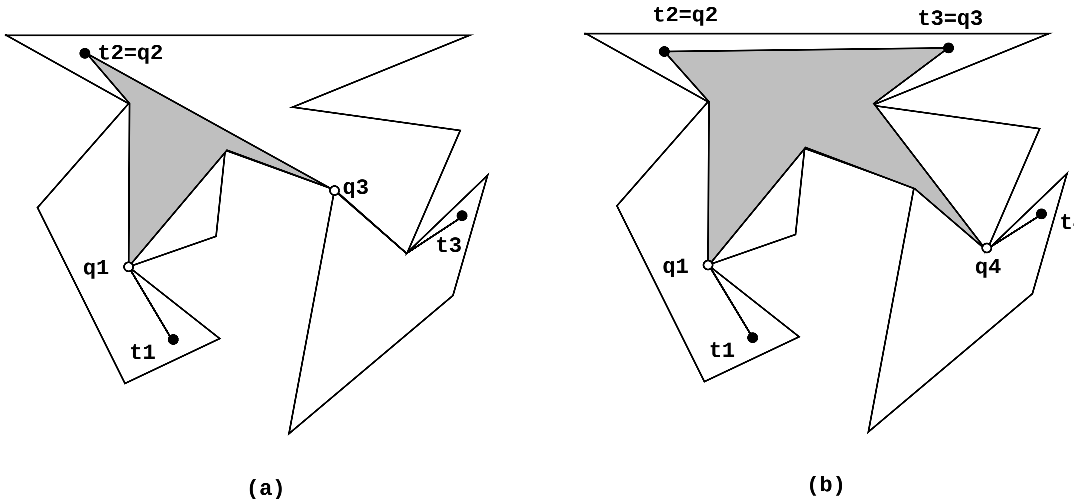


Figure 4: Problem instances with 3 and 4 terminals

3.2 Four Terminals

In this subsection we consider the set T of four terminals t_1, t_2, t_3, t_4 inside the simple polygon P. We show that the problem can be reduced to the ESTP in a c-kite, $c = 3, 4$, for its convex vertices (Fig. 4b).

Let $P' = GCH(P, Z)$. If $i(P') = \emptyset$, then the ESMT for T in P' is trivially given. We assume therefore in the following that $i(P') \neq \emptyset$. If $i(P')$ contains one of the terminals, then P'' is determined as described in Section 3.1. In the following, we assume therefore that all terminals are on the boundary of P'.

If $i(P')$ is not connected, then the problem breaks down into two smaller subproblems, each with three vertices as terminals. Such subproblems can be solved as described in Section 4 in $O(k)$ time and space. The connectivity check can also be done in $O(k)$ time. In the following we assume therefore that $i(P')$ is connected. Note however that P' can be weakly-simple.

Consider the shortest paths from a terminal t_u, $u = 1, 2, 3, 4$, to the remaining three terminals. Let q_u denote their last common vertex. Consider the polygon P'' obtained from P' by cutting off $P'(q_u, t_u) \cup P'(t_u, q_u)$, $u = 1, 2, 3, 4$. Let $Q = \{q_1, q_2, q_3, q_4\}$. Once the ESMT for *semi-terminals* in Q is determined, the ESMT for the terminals in T is obtained by adding the paths $P'(t_u, q_u)$, $u = 1, 2, 3, 4$. P'' is either a 3-kite or a 4-kite. If P'' is a 3-kite, then its fourth semi-terminal is a terminal in P''.

4 ESMTs for Three Semi-Terminals

The ESMT for $Q = \{q_1, q_2, q_3\}$ in P'' is the shortest of the following trees spanning Q (classified by the number of Steiner points).

- No Steiner points. Take the EMST for Q consisting of two shortest paths in P'' spanning Q. This can be done in $O(1)$ time if the shortest paths are given.
- One degenerate Steiner point. There are at most 2 wide reflex vertices in a 3-kite. For each of these, consider the EMST of Q and the wide reflex vertex. Retain the shortest of these 2 trees.
- One Steiner point. This case is covered in the remaining part of this section.

There is an obvious $O(k^3)$ time and $O(k)$ space algorithm for finding the unique ESMT for Q. Consider all $O(k^3)$ subsets of 3 vertices one by one until a Steiner tree with its edges overlapping with interior tangents is obtained. We give an $O(k)$ time and space algorithm which exploits circular rotations of three interior tangents.

- **Initialization:** Let L_1 denote the interior tangent overlapping with an arbitrary edge $v_1^- v_1$ of P''. Traverse the vertices of P'' clockwise, beginning at v_1, until reaching a vertex v_2 with an interior tangent L_2 making 120^o with L_1. Continue until reaching a vertex v_3 with an interior tangent L_3 making 240^o with L_1.

- **Iteration:** Let α denote the minimum angle so that the counterclockwise rotation of at least one of the three interior tangents by α causes it to overlap with an edge of P''. Determine a Steiner tree with one Steiner point s such that the three edges sv_1, sv_2 and sv_3 make 120^o with each other (or decide that it does not exist). If sv_1, sv_2, sv_3 overlap with interior tangents at respectively v_1, v_2, v_3, at most angle α from respectively L_1, L_2, L_3, then connect touch-points to semi-terminals $q^s_{v_1}$, $q^s_{v_2}$, $q^s_{v_3}$. If all three semi-terminals of Q are thereby spanned, save the tree, provided that its length is less than the length of the best solution found so far.

- **Sweep:** Rotate the interior tangents (counterclockwise) around their touch vertices by α. This rotation causes L_u and $v_u v_u^+$ to overlap for some $u = 1, 2, 3$. Replace v_u by v_u^+.

- **Termination:** Stop if the interior tangents have been rotated by at least 120^o; otherwise perform next **Iteration**.

5 ESMT for Four Semi-Terminals in a 4-Kite

When determining the ESMT for $Q = \{q_1, q_2, q_3, q_4\}$ in P'', assuming that $i(P'') \neq \emptyset$ and $i(P'')$ is connected, we need to distinguish between two cases depending on whether P'' is a 3-kite or a 4-kite.

If P'' is a 4-kite, then the ESMT for Q in P'' is the shortest of the following trees spanning Q (classified by the number of Steiner points):

- No Steiner points. Take the EMST for Q with shortest paths in P'' as edges. This can be done in $O(1)$ time if the shortest paths are given.

- One Steiner point. Given the ESMT for $Q_u = Q \setminus \{q_u\}$, $u = 1, 2, 3, 4$, a tree spanning Q is obtained by connecting q_u to the closest semi-terminal in Q_u. Retain the shortest of these four trees.

- One degenerate Steiner point v. There are at most five wide reflex vertices in a 4-kite which can act as v. Consider the EMST of $Q \cup \{v\}$. Retain the shortest of these five trees.

- One Steiner point and one degenerate Steiner point v. Connect v to the semi-terminals q_v^+ and q_v^-. Determine the ESMT for $Q \cup \{v\} \setminus \{q_v^+, q_v^-\}$. Retain the shortest of these five trees.

- Two degenerate Steiner points. Determine the EMSTs of Q and every pair of two wide reflex vertices. Retain the shortest of these ten trees.

- Two Steiner points. The case when Steiner points are visible to each other in $i(P'')$ is covered in Subsection 5.1. The case when they are invisible to each other is covered in Subsection 5.2.

5.1 Visible Steiner Points

In this subsection we discuss the problem of determining the shortest tree spanning four semi-terminals of a 4-kite P'' with two Steiner points s_{23} and s_{41} *visible* to each other (Fig. 5).

There is an obvious $O(k^4)$ time and $O(k)$ space algorithm. In the preliminary version of this paper [12], we gave an $O(k \log k)$ time and $O(k)$ space algorithm based on circular rotations of four interior tangents. Here we give an $O(k)$ time and space algorithm which also exploits circular rotations. However, six interior tangents are used. The additional two tangents make it possible to avoid an explicit intersection test between the edge connecting two Steiner points and P''. In fact, the algorithm becomes much simpler than its four-tangent predecessor.

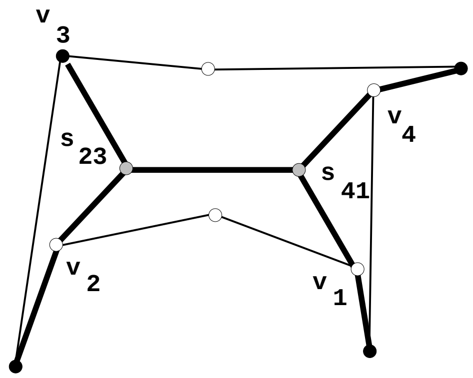


Figure 5: A tree with 2 visible Steiner points s_{23} and s_{41}

- **Initialization:** Let L_1 denote the interior tangent overlapping with an arbitrary edge $v_1^- v_1$ of P''. Traverse the vertices of P'' clockwise, beginning at v_1, until reaching a vertex v_{12} admitting an interior tangent L_{12} making 60^o with L_1, and a vertex v_2 with an interior tangent L_2 making 120^o with L_1. Continue until reaching vertices v_3, v_{34} and v_4 with interior tangents L_3, L_{34} and L_4 distinct but parallel with L_1, L_{12} and L_2, respectively.

- **Iteration:** Let α denote the minimum angle so that the counterclockwise rotation of at least one interior tangent by α causes it to overlap with an edge of P''. Determine a Steiner tree for v_1, v_2, v_3, v_4, with v_2, v_3 and v_4, v_1 having common Steiner points s_{23} and s_{41}, respectively (or decide that it does not exist). If $s_{41}v_1, s_{23}v_2, s_{23}v_3, s_{41}v_4$ overlap with interior tangents at respectively v_1, v_2, v_3, v_4, at most α from respectively L_1, L_2, L_3, L_4, then connect touch vertices to semi-terminals $q_{v_1}^{s_{41}}$, $q_{v_2}^{s_{23}}$, $q_{v_3}^{s_{23}}$, $q_{v_4}^{s_{41}}$. Check if all four semi-terminals of Q are thereby spanned, and if the edge $s_{23}s_{41}$ is to the right of interior tangents for v_{12} and v_{34}. These interior tangents must be at most α away from L_{12} and L_{34}. The tree is saved if its length is less than the length of the best solution found so far.

- **Sweep:** Rotate the interior tangents (counterclockwise) around their touch vertices by α. Suppose that this rotation causes L_u and $v_u v_u^+$ to overlap for some $u = 1, 2, 3, 4, 12, 34$. Replace v_u by v_u^+.

- **Termination:** Stop if the interior tangents have been rotated by at least 180^o; otherwise perform next **Iteration**.

Lemma 3 *Shortest tree spanning four semi-terminals of the 4-kite P'' with two Steiner points visible to each other can be determined in $O(k)$ time and space.*

Proof. The determination of touch vertices $v_1, v_{12}, v_2, v_3, v_{34}, v_4$ requires one scan of the vertices of P''. Semi-terminals q_v^-, q_v^+ and their distances from v for all $v \in P''$ can be determined using two scans of P''. Hence, the preprocessing and initialization can be done in $O(k)$ time and space.

The existence of the Steiner tree with v_2 and v_3 adjacent to s_{23}, and v_4 and v_1 adjacent to s_{41} can be verified in $O(1)$ time. If the Steiner tree exists, the locations of s_{41} and s_{23} are determined in $O(1)$ time.

The edges $v_1s_{41}, v_2s_{23}, v_3s_{23}, v_4s_{41}$ must overlap with interior tangents of P'' (with the same touch vertices) at most α away from L_1, L_2, L_3, L_4, respectively. The edge $s_{23}s_{41}$ must be to the right of two interior tangents of v_{12} and v_{34}. They must have the same touch vertices and be at most α away from L_{12} and L_{34}. The semi-terminals of Q must be covered by $\{q_{v_1}^{s_{41}}, q_{v_2}^{s_{23}}, q_{v_3}^{s_{23}}, q_{v_4}^{s_{41}}\}$. These facts can be verified in $O(1)$ time. Furthermore, they ensure that the Steiner tree is completely in P''. In order to verify this, consider first the edge v_1s_{41}. It cannot intersect the two sides of P'' with $q_{v_1}^{s_{41}}$ as their common end-vertex. These two sides are concave, turn away from each other and v_1s_{41} overlaps with an interior tangent at v_1 (which is a vertex of at least one of these two sides). The same can be shown for the other three edges $v_2s_{23}, v_3s_{23}, v_4s_{41}$ of the Steiner tree. This in particular implies that the side of P'' joining $q_{v_2}^{s_{23}}$ with $q_{v_3}^{s_{23}}$ cannot be intersected by $v_1 s_{41}$. Finally, the side of P'' joining $q_{v_3}^{s_{23}}$ and $q_{v_4}^{s_{41}}$ cannot be intersected by $v_1 s_{41}$. If it did, the same side would have to be intersected by $v_4 s_{41}$.

It remains to show that $s_{41} s_{23}$ is in P''. It cannot intersect the side connecting $q_{v_4}^{s_{41}}$ with $q_{v_1}^{s_{41}}$ as this would force $v_1 s_{41}$ or $v_4 s_{41}$ to intersect P''. Similarly, it cannot intersect the side connecting $q_{v_2}^{s_{23}}$ with $q_{v_3}^{s_{23}}$. Assume that it intersects the side connecting $q_{v_1}^{s_{41}}$ with $q_{v_2}^{s_{23}}$. Since $s_{41} s_{23}$ is between 2 parallel tangents, the intersection with P'' implies yet another parallel interior tangent. This contradicts the assumption that P'' is a 4-kite.

During each iteration, one touch vertex is replaced. Hence, $O(k)$ time and space is used in total. ■

5.2 Invisible Steiner Points

In this subsection, we discuss the problem of determining the ESMT for Q under the assumption that it has two Steiner points s_{23} and s_{41} *invisible* to each other (Fig. 6).

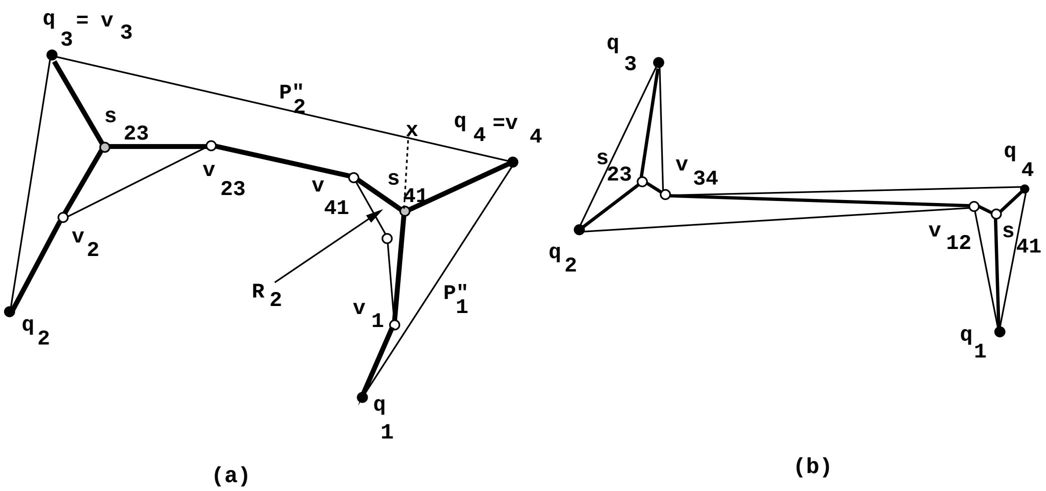


Figure 6: A tree with 2 invisible Steiner points s_{23} and s_{41}

Assume first that the polygonal chain connecting s_{23} and s_{41} in the ESMT for Q touches $P''(q_1, q_2)$ in at least one reflex vertex, and does not touch $P''(q_3, q_4)$, as shown in Fig. 6a. Let v_{41} and v_{23} denote the first and the last vertex of P'' on the path from s_{41} to s_{23} in the ESMT for Q.

Consider the half-line from v_1 through s_{41}. Its first intersection with P'' is denoted by x. The line-segment $v_1 x$ divides P'' such that q_4 is separated from both q_2 and q_3. Let P_1'' denote the part containing q_1, and let P_2'' denote the other part of P''. The ESMT for $Q_4 = Q \setminus \{q_4\}$ cannot go through the interior of P_1''. Furthermore, it cannot have a Steiner point in the region R_2 bounded by v_{41}, s_{41}, v_1 and $P''(v_1, v_{41})$. Hence the leg of the ESMT for Q_4 from q_1 to its Steiner point touches the segment $v_{41} s_{41}$. Behind this segment, the ESMT for Q_4 must overlap with the ESMT for Q. If not, the latter would not be optimal. Furthermore, the optimality of the ESMT for Q_4 implies that the segment $v_{41} s_{41}$ is touched at v_{41}.

It follows that in order to determine ESMT for Q with the polygonal chain connecting s_{23} and s_{41} touching $P''(q_1, q_2)$, one needs to determine the ESMT for Q_4 and the ESMT for $Q_3 = Q \setminus \{q_3\}$.

If the polygonal chain connecting s_{23} and s_{41} touches $P''(q_3, q_4)$ in at least one reflex vertex, and does not touch $P''(q_1, q_2)$, analogous arguments apply.

Assume next that the polygonal chain connecting s_{23} and s_{41} touches both $P''(q_3, q_4)$ and $P''(q_1, q_2)$ in at least one reflex vertex as shown in Fig 6b. Let L_{12} and L_{34} be interior tangents as defined in Subsection 5.1. In particular, their touch vertices are v_{12} and v_{34}, respectively. If v_{12} and v_{34} are on opposite sides of P'' and the line overlapping with $v_{12} v_{34}$ is an interior tangent at both v_{12} and v_{34} at most α away from both L_{12} and L_{34}, determine an ESMT for $\{q_1, q_4, v_{12}\}$ and an ESMT for $\{q_2, q_3, v_{34}\}$.

Join these two ESMTs by the edge $v_{12}v_{34}$. Rotate the interior tangents L_{12} and L_{34} by α and repeat. Stop after 180^o rotation.

$O(k)$ time is needed to determine a pair of ESMTs for 3 vertices. L_{12} and L_{34} can overlap at most twice while their touch vertices are reflex vertices of $P''(q_3, q_4)$ and $P''(q_1, q_2)$, respectively. Consequently, during the 180^o rotation, the need for the determination of ESMTs for 3 vertices can occur at most 4 times.

Lemma 4 *Shortest tree spanning four vertices of a 4-kite with its two Steiner points connected by a chain touching the 4-kite can be determined in $O(k)$ time and space.*

6 ESMT for Four Semi-Terminals in a 3-Kite

If P'' is a 3-kite with one of the semi-terminals in its interior, the following lemma excludes the most complicated case of Section 5 with two non-degenerate Steiner points. The other cases are as described in Section 5 (with fewer number of trees generated since the number of wide vertices is at most 2).

Lemma 5 *If P'' is a 3-kite with connected and non-empty $i(P'')$, then the ESMT for Q has at most one non-degenerate Steiner point.*

Proof. Suppose that the ESMT for Q has two non-degenerate Steiner points *visible* to each other (Fig. 7a). Edges incident with Steiner points make 120^o. Therefore $v_3s_{23}||v_1s_{41}$ as well as $v_2s_{23}||v_4s_{41}$. At least one pair of these parallel edges touches P''. These two edges overlap with distinct interior tangents, contradicting the assumption that P'' is a 3-kite.

Suppose next that the ESMT for Q has 2 Steiner points *invisible* to each other (Fig. 7b). Three of the vertices adjacent to Steiner points must be on the boundary of P''. The corresponding edges must overlap with interior tangents at these vertices. The ESMT for Q partitions the interior of P'' into four regions. Vertices of three of them (bounding shaded regions in Fig. 7b) admit interior tangents with slopes differing by 180^o in total (60^o each). Furthermore, vertices on the path from s_{23} to s_{41} admit additional interior tangents, contradicting again the assumption that P'' is a 3-kite. ■

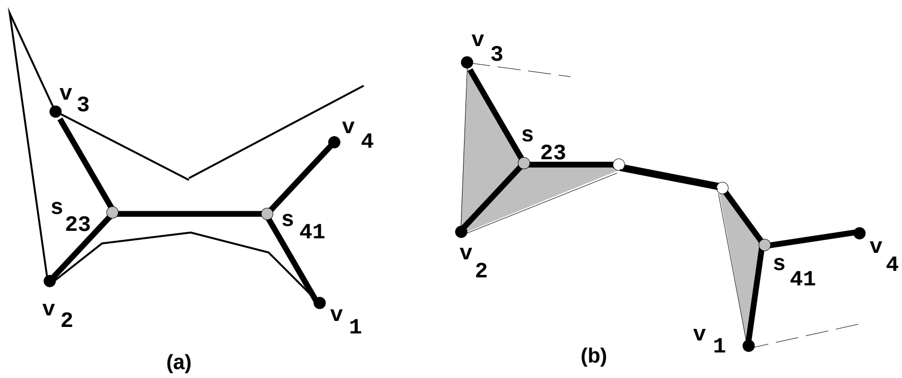


Figure 7: ESMTs with 2 Steiner points s_{23} and s_{41} in a 3-kite

7 Heuristic

The heuristic proposed in this paper consists of three major steps. In the first step appropriate terminal subsets with two, three and four elements are determined. Then ESMTs for these subsets are constructed by considering a constant number of possible topologies for each subset. At most $O(k)$ time and space is needed for each topology. Finally, concatenation of ESMTs for subsets is carried out to obtain a solution to the overall problem.

There are several ways of selecting subsets with two, three and four terminals. In Section 1, we suggested to use subsets that induce connected subgraphs of the EMSTs for all terminals (shortest paths between terminals are regarded as edges). Another option, generating more subsets, are geometric duals of the geodesic Voronoi diagrams for the terminals. A reasonable compromise between EMSTs (easy to implement, generating rather limited number of subsets) and dual of Voronoi diagrams (complicated to implement, generating perhaps too many subsets) would be relative neighbor graphs (mentioned in the next section) or Gabriel graphs. The issue of the best subset generator remains an open problem which should be addressed in the future. According to our limited experience, using EMSTs generates on average $3n$ subsets of size three and four. When using relative neighbourhood graphs, this increases to $5n$.

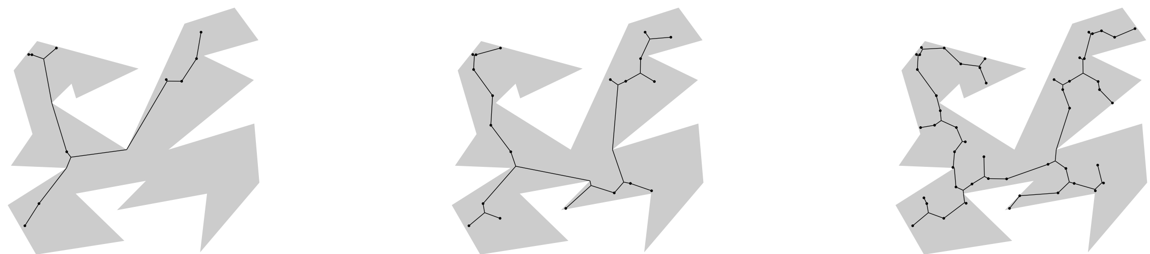
Construction of ESMTs for subsets with three and four terminals was the main issue of the preceeding sections. Given the shortest paths between all terminals (needed anyway to determine small subsets), we argued that ESMTs can be determined in $O(k)$ time and space using simultanous rotational sweep of several interior tangents.

Concatenation strategies will be briefly discussed in the next section. Also here further research is needed to uncover the most advantegous strategy. We experimented with three approaches: greedy (where ESMTs are sorted by the non-decreasing ratio of the length of the ESMT and EMST) and added to the overall solution provided that feasibility is maintained (no cycles are generated), greedy with a subsequent polynomial improvement phase, and exponential exact concatenation using branch-and-cut.

8 Computational Results

The heuristic was experimentally evaluated on an HP9000/C200 workstation using the programming language C++ and class library LEDA (version 3.7.1) [4]. In order to evaluate the quality of the trees produced, optimal solutions were computed using the exact algorithm of Zachariasen and Winter [13].

The first series of problem instances was generated using a hand-drawn polygon P_{26} with $k = 26$ vertices. For each $n = 10, 20, ..., 100, 150, 200, 250, ..., 500, 600, ..., 900, 1000$, we randomly generated ten sets of terminals (uniformly in the interior of P_{26}).



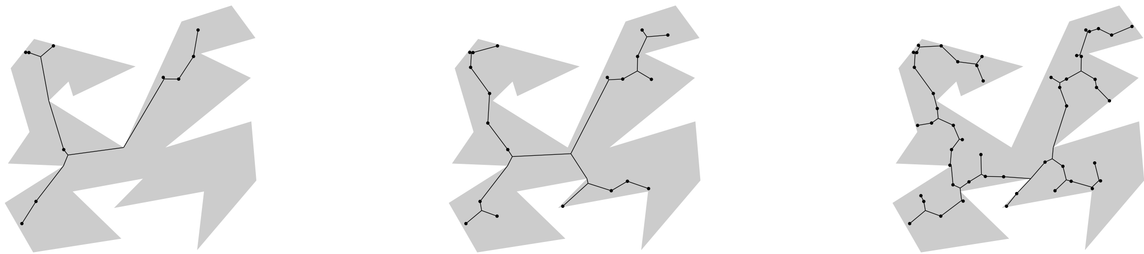
10 terminals | 20 terminals | 50 terminals

Figure 8: Heuristic solutions (using exact concatenation)

In the left part of Table 1, we present reductions in percent over the EMSTs. CPU-times are shown in the right part of Table 1.

- The ratios in the `Fast` column are obtained by the fast, straightforward $O(s \log s)$ concatenation, where s is the number of generated small ESMTs. ESMTs are ordered by non-decreasing ratio between their length and the length of the EMST spanning the same set of terminals. ESMTs are added to the final solution in greedy fashion provided that no cycle is created.

- The ratios in the `Slow` column are obtained by the greedy concatenation followed by a polynomial improvement phase. This $O(s^2)$ approach was also used successfully in connection with the heuristic for the ESTP in the plane (with no bounding polygon). The reader is referred to [14] for the description of this approach.



10 terminals | 20 terminals | 50 terminals

Figure 9: Exact solutions

- The ratios obtained by the exact concatenation based on the branch-and-cut algorithm are shown in the `Exact` column. The details concerning the exact concatenation can be found in [9] where it was successfully applied to find exact solutions to very large instances of the Euclidean and rectilinear Steiner tree problems.

- The ratios in the `RNG` column are obtained by using relative neighbourhood graphs instead of EMSTs when determining small subsets of terminals. Furthermore, exact concatenation based on the branch-and-cut approach is used.

- The ratios in the `4-ESMT` column are obtained by the exact algorithm where the generation of ESMTs is cut-off for more than four points. Note that ESMTs of terminals *and* polygonal vertices are generated.

- The ratios between optimal solutions and EMSTs are shown in the `ESMT` column. Only problem instances with up to 100 terminals were solved to optimality. The exact algorithm to solve the Euclidean Steiner tree problem inside a polygon (with or without holes) is described in [13].

- The `H-CPU` column shows CPU-times for the heuristic using the exact concatenation. Computational times for the heuristics using less elaborate concatenation are not much smaller and therefore are not shown. It should be noted that the CPU-times for large instances are dominated by the computation of the EMST for all terminals. The reason is that we use a straightforward algorithm which constructs the visibility graph of terminals and polygon vertices. For $n = 1000$ approximately 80% of the CPU-time is spent computing the EMST. By using a more elaborate algorithm for computing the EMST (e.g. based on the geodesic Voronoi diagram) this part of the algorithm would not have dominated the running time for large instances.

- The `R-CPU` column shows CPU-times for the heuristic using relative neighborhood graphs and exact concatenation. The number of subsets for which small ESMTs are generated increases significantly. For example, for 600 terminals, the number subsets of size 3 increased on average (over 10 problem instances) from 735 to 1154. The number of subsets of size 4 increased on average from 983 to 2091. The CPU-time for the generation of these small ESMTs went up from 249.90 to 357.55 on average. However, the real time-consuming part of the heuristic when using relative neighborhood graphs, is the exact concatenation. It went up from 27.31 (all but 1 exact concatenation took less than 2 seconds; the difficult one took 258.98) to 579.83 on average. If respectively slow or fast concatenation were used, CPU-times for the concatenation would drop to 12.51 and 0.04 respectively. The ratio drop would then be from 3.25 down to 3.18 and 2.92 respectively.

- The `4-ESMT` column shows CPU-times for the cut-off algorithm for FSTs spanning at most four points (terminals or polygonal vertices).

- The `E-CPU` column shows computational times needed to solve the same problem instances to optimality. As it can be seen, there are considerable CPU-savings available by using the heuristic.

The quality of the solutions obtained by the heuristic is on average not very far from the optimal solution. Some additional improvements possibilities are discussed in the concluding section.

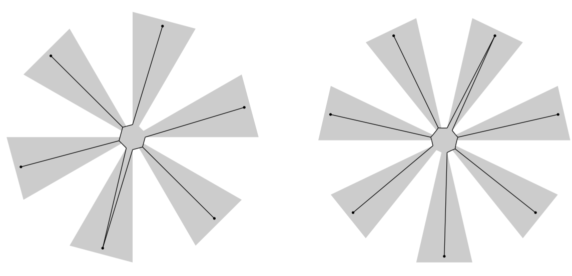
Table 1: Fixed polygon - Experimental results

n	Fast	Slow	Exact	RNG	4-ESMT	ESMT	H-CPU	R-CPU	4-CPU	E-CPU
10	5.68	5.79	5.81	5.81	5.83	5.83	0.50	0.42	4.68	27.95
20	3.92	4.04	4.06	4.06	4.34	4.35	0.91	0.97	17.74	163.95
30	3.87	3.95	3.95	3.95	4.17	4.17	1.35	1.56	28.28	271.44
40	3.55	3.74	3.76	3.81	4.14	4.15	2.03	2.42	44.96	667.27
50	3.28	3.40	3.41	3.48	3.72	3.73	2.49	3.07	62.81	810.19
60	3.04	3.19	3.21	3.31	3.50	3.51	3.14	4.14	84.50	1215.83
70	2.98	3.08	3.10	3.17	3.35	3.37	3.92	5.20	107.43	1440.90
80	2.78	2.89	2.93	3.03	3.16	3.17	4.61	6.42	132.75	1893.80
90	2.81	2.91	2.92	2.99	3.10	3.11	5.31	7.53	157.02	2359.92
100	2.76	2.86	2.88	3.02	3.20	3.21	6.15	9.35	171.81	2781.75
150	2.75	2.82	2.83	3.00			11.97	19.19		
200	2.72	2.79	2.81	2.96			20.71	41.99		
250	2.74	2.82	2.84	3.00			32.44	60.13		
300	2.74	2.85	2.87	3.07			48.21	98.05		
350	2.83	2.94	2.96	3.13			69.43	157.64		
400	2.93	3.04	3.06	3.19			93.14	546.37		
450	2.91	3.01	3.03	3.16			121.95	518.59		
500	2.94	3.04	3.05	3.29			158.30	342.85		
600	2.92	3.02	3.04	3.25			245.40	925.78		
700	2.84	2.93	2.95				357.42			
800	2.84	2.94	2.96				513.67			
900	2.87	2.96	2.98				701.26			
1000	2.90	2.99	3.01				925.06			

Comparing the columns `Exact` and `RNG` it is clear that better subset generation methods can improve solution quality substantially. For the obstacle-free problem Zachariasen and Winter [14] showed that the so-called Gabriel graph (which contains both a minimum spanning tree and the relative neighbourhood graph) produced the best results.

Furthermore, the `Fast` and `Slow` columns show that improved concatenation methods also play an important role in the performance of the heuristic. The results obtained by the more time consuming method are very close to the results obtained by exact concatenation — but further improvements can be obtained by using a local search algorithm [15].

In the second series of tests, the vertices of the polygon were restricted to be on two concentric circles, such that they alternated in a regular fashion between the two circles. The radius of the inner circle is 10 times less than the radius of the outer circle. Consequently, the polygon looks like a fan with a parametrized number of fans. Exactly one terminal was placed in the tip of each fan, see Fig. 10.

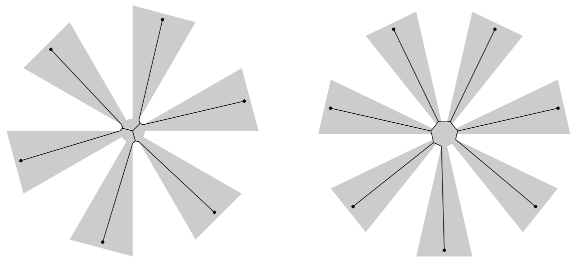


Fan with 6 wings

Fan with 7 wings

Figure 10: Fans - heuristic solutions (using exact concatenation)

Our results, shown in Table 2, clearly indicate that the heuristic solutions are not close to the optimal solutions. Furthermore, it does not really help to use relative neighborhood graphs rather than EMSTs when selecting small subsets of terminals. However, these instances are particularly difficult for the exact algorithm, since a huge number of FSTs has to be generated. On the other hand, ESMTs for fans with 7 or more wings will consist of FSTs spanning at most 3 terminals and/or polygonal points.



Fan with 6 wings   Fan with 7 wings

Figure 11: Fans - exact solutions

Consequently, the cut-off algorithm will perform extremely well in this case. In fact, entries in the `ESMT` column of Table 2 were obtained by the cut-off algorithm generating FSTs with at most 3 terminals and/or vertices (for fans with more than 6 wings).

ESMTs for fans are FSTs spanning all terminals. Consequently, they are not generated by the heuristic. One way out of this problem would be to modify the heuristic, so it generates ESMTs with small number of terminals *and/or* vertices. This in turn would complicate the concatenation of generated ESMTs unless using the exact concatenation based on the branch-and-cut algorithm.

Table 2: Fans - Experimental results

n	Fast	Slow	Exact	RNG	4-ESMT	ESMT	H-CPU	R-CPU	4-CPU	E-CPU
6	27.51	27.51	27.51	27.51	36.67	36.83	0.27	0.35	51.17	649.36
7	30.96	30.96	30.96	31.71	38.71	38.71	2.46	4.21	58.96	492.70
8	26.79	26.79	26.79	26.79	40.19	40.19	3.49	5.79	114.98	2159.19
9	29.52	29.52	29.52	29.95	41.33	41.33	4.70	6.94	234.43	17825.72
10	26.39	26.39	31.67	31.67	42.23	42.23	6.15	8.44	410.14	
11	28.92	28.92	28.92	28.92	42.96	42.96	7.82	10.66	1086.80	
12	26.14	26.14	30.49	30.49	43.56	43.56	10.00	12.63		
13	28.24	28.24	32.05	32.25	44.07	44.07	12.01	15.40		
14	29.67	29.67	29.67	29.67	44.39	44.39	13.58	17.28		
15	31.22	31.22	31.22	31.22	44.88	44.88	15.98	19.87		
16	25.83	25.83	32.29	32.29	44.95	44.95	18.74	22.76		

9 Conclusions

We presented $O(k)$ time and space algorithms for the ESMT for three and four terminals inside a simple polygon with k vertices. We also indicated how geodesic Voronoi diagrams and EMSTs can be used to determine reasonable subsets of terminals. Using $O(s \log s)$ time, where s is the number of selected subsets, ESMTs can be arranged in non-decreasing order of the ratio between the lengths of their ESMTs and EMSTs. Note that when using EMSTs, s would be of order $O(n)$ in the obstacle-free case; degrees of vertices in EMSTs are bounded by a constant. This is not necessarily the case inside a simple polygon for which there exist instances where s is of order $\Theta(n^3)$. When concatenated in greedy fashion (avoiding cycles), a reasonable solution to the Euclidean Steiner tree problem for *any* number of terminals inside a polygon is obtained. The time needed to determine ESMTs and EMSTs for the selected s subsets is $O(sk)$. Therefore, the overall running time complexity of the algorithm is $O(sk \log s + (n+k)\log(n+k))$, where the second term is the worst-case time complexity of the geodesic Voronoi diagram algorithm.

There is a number of interesting issues that remain open. Can ESMT's for 5, 6 or any fixed number of terminals be determined in $O(k)$ time and space? The determination of ESMTs for small subsets of terminals in presence of several (convex) obstacles is also of interest. In this context, Steiner visibility graphs introduced in [10] could prove useful. Finally, we mention the problem of preprocessing a simple polygon so that three and/or four terminals queries for ESMTs can be answered efficiently.

References

[1] J.E. Beasley, A heuristic for the Euclidean and rectilinear Steiner tree problem, *EJOR* **58** (1992) 284-292.

[2] F.K. Hwang, D.S. Richards and P. Winter, *The Steiner Tree Problem*, North-Holland (1992).

[3] E. Papadopoulou and D.T. Lee, A new approach for the geodesic Voronoi diagram of points in a simple polygon and other restricted polygonal domains, *Algorithmica* **20** (1998) 319-352.

[4] K. Mehlhorn and S. Näher, LEDA - A Platform for Combinatorial and Geometric Computing, Max Planck Institute for Computer Science, `http://www.mpi-sb.mpg.de/LEDA/leda.html` (1996).

[5] J.S. Provan, An approximation scheme for finding Steiner trees with obstacles, *SIAM J. Comput.* **17** (1988) 920-934.

[6] J. MacGregor Smith, D.T. Lee and J.S. Liebman, An $O(n \log n)$ heuristic for the Steiner minimal tree problem on the Euclidean metric, *Networks* **11** (1981) 23-29.

[7] G.T. Toussaint, Computing geodesic properties inside a simple polygon, Revue D'Intelligence Artificielle **3** (1989) 9-42.

[8] D.M. Warme, Spanning Trees in Hypergraphs with Application to Steiner Trees, Ph.D. Thesis, Univ. of Virginia (1998).

[9] D.M. Warme, P. Winter and M. Zachariasen, Exact algorithms for plane Steiner tree problems: A computational study. in D.-Z. Du, J.M. Smith and J.H. Rubinstein (Eds.) *Advances in Steiner Trees*, Kluwer (2000) 81-116.

[10] P. Winter, Euclidean Steiner minimal trees with obstacles and Steiner visibility graphs, *Discrete Applied Mathematics* **47** (1993) 187-206.

[11] P. Winter, Euclidean Steiner minimum trees for 3 terminals in a simple polygon, *Proc. of the Seventh Canadian Conference on Computational Geometry*, Univ. Laval, Quebec, Canada (1995) 247-255.

[12] P. Winter, Steiner minimum trees in simple polygons, DIMACS Tech. Report 95-43 (1995).

[13] M. Zachariasen and P. Winter, Obstacle-avoiding Euclidean Steiner trees in the plane: an exact algorithm, in M.T. Goodrich and C.C. McGeoch (Eds.), *Algorithm Engineering and Experimentation* **LNCS 1619** (1999) 282-295.

[14] M. Zachariasen and P. Winter, Concatenation-based greedy heuristics for the Euclidean Steiner tree problem, *Algorithmica* **25** (1999) 418-437.

[15] M. Zachariasen, Local Search for the Steiner Tree Problem in the Euclidean Plane, *European Journal of Operational Research* **119** (1999) 282-300.