Canonical Forms and Algorithms for Steiner Trees in Uniform Orientation Metrics*

M. Brazil † D.A. Thomas† J.F. Weng† M. Zachariasen‡

March 4, 2004

Abstract

We present some fundamental structural properties for minimum length networks (known as Steiner minimum trees) interconnecting a given set of points in an environment in which edge segments are restricted to λ uniformly oriented directions. We show that the edge segments of any full component of such a tree contain a total of at most 4 directions if λ is not a multiple of 3, or 6 directions if λ is a multiple of 3. This result allows us to develop useful canonical forms for these full components.

The structural properties of these Steiner minimum trees are then used to resolve an important open problem in the area: does there exist a polynomial-time algorithm for constructing a Steiner minimum tree, if the topology of the tree is known? We obtain a simple linear time algorithm for constructing a Steiner minimum tree for any given set of points and a given Steiner topology.

*Partially supported by grants from the Australia Research Council and from the Danish Natural Science Research Council.

†ARC Special Research Centre for Ultra-Broadband Information Networks, Department of Electrical and Electronic Engineering, The University of Melbourne, Victoria 3010, Australia

‡Department of Computer Science, University of Copenhagen, DK-2100 Copenhagen Ø, Denmark

1 Introduction

Interconnects in VLSI design have traditionally used Manhattan (or rectilinear) routing, in which two perpendicular wiring directions are allowed. Due to technological advances, there is now an increasing interest in the use of *diagonal* interconnects (see, e.g., `www.Xinitiative.org`). Both traditional Manhattan routing and the addition of diagonal routing are examples of so-called λ-geometry, in which a fixed set of $\lambda \geq 2$ uniformly oriented directions are allowed.

In VLSI routing, one of the principal objectives is to minimize the total length of an interconnection, that is, to compute a λ-geometry *Steiner minimum tree* (or λ-SMT). This is in general an NP-hard problem. However, the problem becomes significantly simpler if the topology of the tree is known. For the Euclidean case ($\lambda = \infty$), the well-known Hwang-Melzak algorithm [5, 6] solves this problem in linear time. For the rectilinear case ($\lambda = 2$), a theorem of Hwang [4] again solves this problem in linear time, by showing that there exists a Steiner minimum tree in which the full components take on restricted canonical forms.

In [2] an open question was asked as to whether, for each fixed $\lambda \geq 2$, there exists a polynomial-time algorithm for finding a λ-SMT for any given set of n terminals and given full Steiner topology (in which all terminals have degree 1 and all Steiner points degree 3). Such an algorithm would immediately generalise to any λ-SMT, by decomposing it into its full components. The existence of such an algorithm was recently proved by Xue and Thulasiraman in [16], by showing that the problem can be transformed to a linear programming problem whose size is bounded by a polynomial in n and λ. It should be noted, however, that the complexity of this algorithm will generally have very high degree, and, indeed, although the algorithm is polynomial in the total input size, it has not been shown to be polynomial in n and λ. In another recent result Nielsen et al. [12] obtained a linear time algorithm for constructing a λ-SMT for a given full Steiner topology when λ is a multiple of 3. However, they also showed that their methods cannot be extended to other values of λ.

In this paper we fully answer the above question by presenting a linear time algorithm for constructing a λ-SMT for any given set of terminals and given Steiner topology. In developing this algorithm, we also establish canonical forms for the full components of λ-SMTs which give a useful insight into the geometry of these trees, and have already resulted in a substantial speed-up in an exact algorithm for constructing λ-SMTs [11].

The organisation of this paper is as follows. In Section 2, we fix our notation, give definitions and state some known results which will be used later. In

Section 3, we investigate properties of edge directions, showing that the edge segments of any full component of a λ-SMT contain a total of at most 4 directions if λ is not a multiple of 3, or 6 directions if λ is a multiple of 3. In Section 4, we define a 0-shift, a mechanism which allows us to change the directions of two edges in a full λ-SMT while leaving the directions of all other edges fixed. Using this concept, we then establish our canonical forms in Section 5. Then, in Section 6, we first give a simple quadratic time algorithm for constructing a λ-SMT for a given full Steiner topology; by careful refinement of the algorithm we can reduce the running time to linear time.

2 Background and Preliminaries

Let $\lambda \geq 2$ be a given positive integer. Given λ orientations $i\omega$ $(i = 1, 2, ..., \lambda)$ in the Euclidean plane, where $\omega = \pi/\lambda$ is a unit angle, we represent these orientations by the angles with the x-axis of corresponding straight lines. A line or line segment with one of these orientations is said to be in a *legal direction*. Objects composed of line segments in legal directions are said to belong to a *λ-geometry*.

A *Minkowski plane* (or *normed plane*) is a two-dimensional real normed space $(\mathbb{R}^2, \|\cdot\|)$ with *unit ball* $\{\mathbf{x} : \|\mathbf{x}\| \leq 1\}$. (For more background on Minkowski planes, see [14].) The λ-geometry is an example of a Minkowski plane where the unit ball is a regular 2λ-gon. Hence, the distance function in λ-geometry is a norm.

In this paper we study properties of minimum length networks in λ-geometry interconnecting a given set of points, also referred to as *terminals*. We only consider the problem for $4 \leq \lambda < \infty$ as strong results for the cases where $\lambda = 2$, 3 or ∞ are already known [1, 4, 5, 8].

Since any minimum length network clearly is a tree, we will only discuss networks in λ-geometry that are trees. We define a λ-tree to be a tree network in λ-geometry interconnecting a given set of terminals. A λ-tree can contain nodes of degree 3 or more that are not terminals. These nodes are referred to as *Steiner points*. A λ-tree is said to be *full* if all terminals have degree 1. Clearly any λ-tree can be decomposed into a union of full subtrees meeting only at terminals. This decomposition allows us to limit our attention to full λ-trees in this paper.

We next define a *Steiner λ-tree* to be a λ-tree that cannot be shortened by the perturbation of a single Steiner point. For the purposes of this paper we will assume that all Steiner points in a full Steiner λ-tree have degree 3. Degree 4 Steiner points can occur in some λ-SMTs, but in such cases there always exists a

λ-SMT where every degree 4 Steiner point is adjacent to four terminals. (Degree 4 Steiner points only occur in the cases where $\lambda = 4$ or 6. The proof of the above statement for $\lambda = 4$ is given in [1]; the proof for $\lambda = 6$ is similar.)

Edges in a Steiner λ-tree are assumed to be geodesics (in λ-geometry) between their endpoints. We refer to such an edge as a *straight edge* if it is a single straight line segment (in a legal direction), or else as a *bent edge* if it consists of two or more straight line components. It has been shown in [15] that bent edges are composed of line segments in exactly two legal directions differing by an angle of ω. Furthermore, although there are infinitely many ways of embedding a bent edge pq in the Euclidean plane, there are only two embeddings composed of exactly two straight line segments, as shown in Figure 1(a). Each of these contains two edges of the parallelogram pr_1qr_2 where the interior angles at r_1 and r_2 are $\pi - \omega$. The straight line components in such an embedding are referred to as *half-edges* and the points r_1 and r_2 as *corner points*.

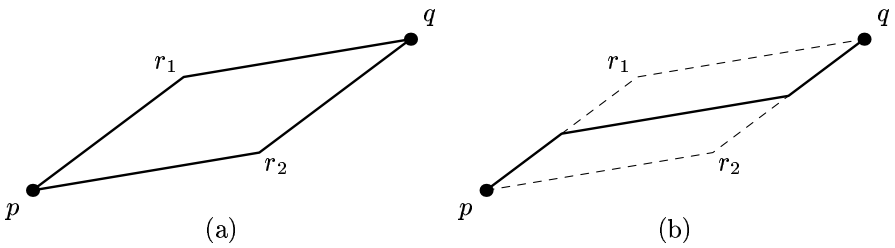

Figure 1: Embeddings of a bent edge in the Euclidean plane.

A bent edge can also be embedded so that the two line segments incident with the endpoints are parallel, as shown in Figure 1(b). This embedding will prove useful later in this paper.

An important property of a Steiner λ-tree is that there are strong restrictions on the possible angles at Steiner points, as indicated in the following result [2, 9, 13].

Proposition 2.1 *For a given λ, let T be a Steiner λ-tree. Then, the only possible angles at any Steiner point of T are:*
(1) $(2m-1)\omega$, $(2m)\omega$ $(= 2\pi/3)$ *or* $(2m+1)\omega$ *if* $\lambda = 3m$*;*
(2) $(2m)\omega$ *or* $(2m+1)\omega$ *if* $\lambda = 3m+1$*; and*
(3) $(2m+1)\omega$ *or* $(2m+2)\omega$ *if* $\lambda = 3m+2$.

Note that the angles in Proposition 2.1 are precisely those that differ by no more than ω from $2\pi/3$.

A Steiner λ-tree T is said to be *locally minimal* if there is no perturbation of its Steiner points which reduces the length of T. Unlike the Euclidean case, this is a strictly stronger condition than simply saying that T is a Steiner λ-tree, as it includes the simultaneous perturbation of a number of Steiner points at once (see [2] for examples proving that this is a stronger condition). We will usually refer to a locally minimal Steiner λ-tree with a given topology $\mathcal{T}$ as a *λ-SMT for $\mathcal{T}$*. However, if we refer to a λ-tree T as a λ-SMT without reference to its topology then we mean that T is a globally minimum Steiner λ-tree for its terminal set.

Let P be a path between two nodes in a full Steiner λ-tree T. It is useful to define the *angle sequence* of P, denoted $A(P)$, to be the sequence of angles encountered on the right at Steiner points while traversing P. For example, suppose $P = p \ldots s_1 s_2 s_3 \ldots q$, where s_1, s_2, s_3 are Steiner points. Let $s_2 r$ be the edge incident to s_2 not lying on P. Then if $s_2 r$ is on the left of P, s_2 contributes a single angle, $\angle s_1 s_2 s_3$, to $A(P)$. If $s_2 r$ is on the right of P, s_2 contributes the two angles $\angle s_1 s_2 r$ and $\angle r s_2 s_3$ to $A(P)$. If P contains bent edges, then for each bent edge we can choose any fixed embedding which minimises the length between endpoints when computing $A(P)$. However, if a bent edge is an interior edge of P it is convenient to choose an embedding of the type shown in Figure 1(b); this will be our convention throughout this paper.

We will make extensive use of the following theorem.

Theorem 2.2 *[3] For a given λ, let T be a full λ-Steiner tree. Then T is locally minimal if and only if for every subpath P of T the following condition is satisfied:*

$$\left| \sum_{\alpha \in A(P)} (\alpha - \frac{2\pi}{3}) \right| \leq \omega. \tag{1}$$

Theorem 2.2 has important implications for the possible structures of a λ-SMT. It should be noted that the theorem is a generalisation of the conditions on angles at a Steiner point. In particular, if we restrict ourselves to subpaths which contain no internal edges, we immediately obtain Proposition 2.1 as a corollary.

3 Edge Directions

In this section we look at some corollaries of Theorem 2.2, and in particular some important basic structural properties of a full λ-SMT that are implied by the theorem.

3.1 Colouring and Orienting Edges

Let T be a full λ-SMT. We begin by describing a method of colouring and orienting the edges of T.

For the colouring of edges, we define an equivalence relation on the edges of T. Two edges are said to be equivalent if the angle sequence of the path P connecting them in T has cardinality 3ℓ for some natural number ℓ. Clearly this relation is transitive and reflexive. To see that it is symmetric, note that if P has k internal vertices then $\text{card}(A(P)) + \text{card}(A(-P)) = 3k$, hence $\text{card}(A(-P)) = 3(k - \ell)$. Thus, the equivalence relation is well defined. If T contains at least one Steiner point, then there are exactly three equivalence classes of edges. We will colour each edge of T red, green or blue depending on which equivalence class it belongs to.

An orientation of the edges of T can be defined as follows. Let r be a fixed vertex of T. Assign a parity (either "odd" or "even") to every vertex of T based on whether the path in T from r to that vertex contains an odd or even number of edges. Since T is a tree, this assignment of parity is well defined, and each edge of T is incident with one odd vertex and one even vertex of T. We can now think of each edge of T as being a directed edge pointing in the direction of the incident odd vertex, as illustrated in Figure 2.

A half-edge is assumed to inherit the colour and orientation of the edge to which it belongs.

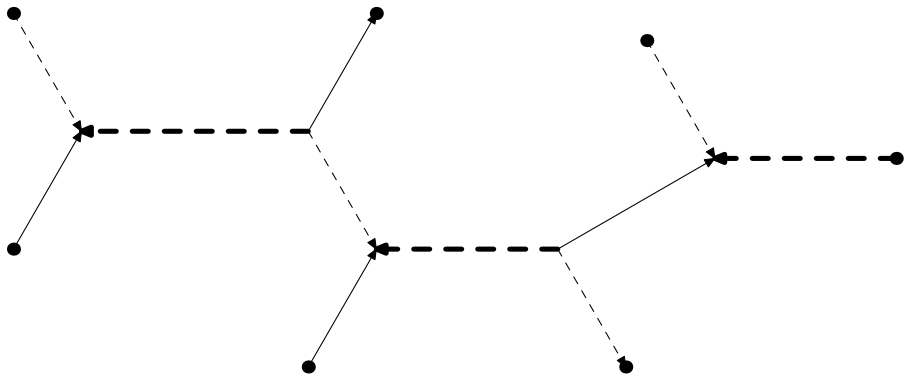

Figure 2: A coloured and oriented Steiner tree. The different types of dashed edges represent the three different colours, and the arrows indicate the orientation of each edge.

3.2 Feasible Edge Directions

We now show that there are strong restrictions on the possible directions of edges in a full λ-SMT.

Lemma 3.1 *Given any two edges or half-edges in a full λ-SMT T, coloured and oriented as above, the smaller angle γ between them is either 0 or ω if they are the same colour, or else satisfies $|\gamma - 2\pi/3| \leq \omega$.*

Proof. This is a straightforward corollary of Theorem 2.2. Let P be the path between the two edges (including the edges themselves). We first observe that the change in direction between the first and last edge of P equals the sum of the angles in $A(P)$. This observation is obvious if P contains no internal edges, and is easily proved by induction (on the number of edges in P) for longer paths. By the definition of $A(P)$, this observation is still true if some internal edges of P are bent edges.

Now, if the two edges are in the same equivalence class, then $\text{card}(A(P))$ is a multiple of 3, and hence $\sum_{\alpha \in A(P)}(\alpha)$ and $\sum_{\alpha \in A(P)}(\alpha - 2\pi/3)$ differ by a multiple of 2π. Thus the result follows by Theorem 2.2. Similarly, if the edges are in different equivalence classes $|\gamma - 2\pi/3| \leq \omega$. ∎

We will refer to the possible directions for oriented straight edges and half edges in a full λ-SMT T as *feasible directions*.

Now suppose, in a given full λ-SMT T, edges or half-edges of some colour, say red, occur in two different directions. Without loss of generality (by rotating if necessary) we can assume the directions of the red edges are 0 and ω.

If $\lambda \neq 3m$ it immediately follows from Lemma 3.1 that the direction of each blue and green edge is fixed. If $\lambda = 3m$ then there are exactly two possible directions for each green and blue edge. These observations are summarised in the following corollary.

Corollary 3.2 *Let T be a full λ-SMT. If $\lambda \neq 3m$ then edges in T use at most 4 different directions. If $\lambda = 3m$ then edges in T use at most 6 different directions.*

Furthermore, up to rotation by multiples of ω, the set of feasible directions in T is as listed in the table in Figure 3.

A set of feasible directions for a given λ will be referred to as a *direction set*. The direction sets for $\lambda = 6$, 7 and 8 are illustrated in Figure 4.

It will be useful to introduce two categories for edges in feasible directions, namely, primary and secondary.

λ	feasible directions
$3m$	$0,\ \omega,\ 2m\omega,\ (2m+1)\omega,\ 4m\omega,\ (4m+1)\omega$
$3m+1$	$0,\ \omega,\ (2m+1)\omega,\ (4m+2)\omega$
$3m+2$	$0,\ \omega,\ (2m+2)\omega,\ (4m+3)\omega$

Figure 3: Feasible directions (up to rotation by a multiple of ω) for edges in a full λ-SMT.

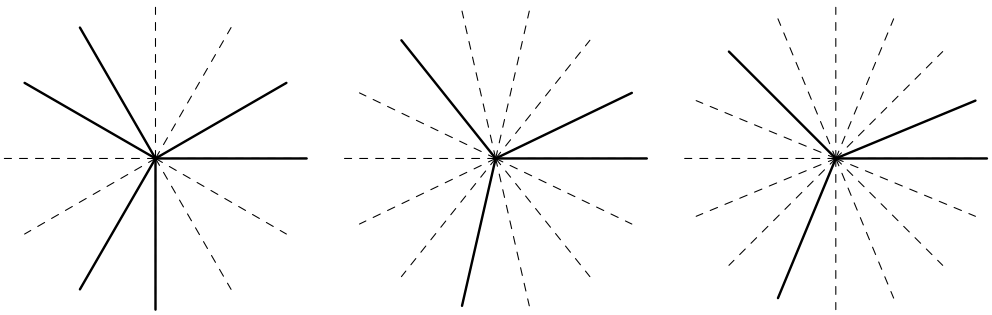

Figure 4: Feasible directions for edges in a full λ-SMT, illustrated for $m = 2$.

Let e be a straight edge or half-edge in a full λ-SMT T, oriented in direction $j\omega$. Then e is said to be *primary* if $(j-1)\omega$ is not a feasible direction for T. Similarly, e is said to be *secondary* if $(j+1)\omega$ is not a feasible direction for T. If $\lambda \neq 3m$ then it is possible for an edge to be both primary and secondary. We say that e is *exclusively primary* (or *exclusively secondary*) if it is primary, but not secondary (or, respectively, secondary, but not primary). For example, for the sets of feasible directions in the table above, if $\lambda = 3m$ then an edge with direction 0, $2m\omega$ or $4m\omega$ is exclusively primary, whereas an edge with one of the remaining three directions is exclusively secondary. On the other hand, if $\lambda = 3m+1$ then a primary edge can have direction 0, $(2m+1)\omega$ or $(4m+2)\omega$, but only an edge of direction 0 is exclusively primary. Note that every component of a bent edge must be either exclusively primary or exclusively secondary.

3.3 Convexity

Let $|T|$ be the total edge length of a λ-tree T whose topology corresponds (possibly in a degenerate way) to a fixed full Steiner topology. Since the distance function in λ-geometry is a norm, it follows that $|T|$, treated as a function of the positions of the Steiner points of T, is a convex function. Note, however, that this convexity is not strict (since the boundary of the unit ball contains non-trivial line segments [14]).

The above convexity property immediately implies the following theorem.

Theorem 3.3 *For a given set of terminals, a locally minimal Steiner λ-tree T is minimum with respect to its Steiner topology. In particular, if T has the same topology as a λ-SMT, then T is itself a λ-SMT.*

Theorem 3.3 cannot reduce the maximum running time of the algorithms in this paper (in Sections 5 and 6) but it can help reduce the average running time, since it means that an algorithm for constructing T can stop as soon as it constructs a locally minimal Steiner tree with the correct topology.

We also briefly note the following result, which says that for a given full Steiner topology the choice of feasible directions for a λ-SMT is unique. The result is straightforward to prove; so we simply give a sketch of the proof below.

Theorem 3.4 *If $\lambda \neq 6$, then the edges of all full λ-SMTs for a given set of terminals and given full Steiner topology use a unique set of feasible directions.*

Proof. (Sketch) Here we give an outline of the proof for the case where λ is not a multiple of 3. If λ is a multiple of 3 then a similar argument applies, except for $\lambda = 6$, where two sets of feasible directions are possible in some cases.

Let T and T' be full λ-SMTs for a given set of terminals and a given full Steiner topology $\mathcal{T}$. Assume, contrary to the statement of the theorem, that T and T' have different sets of feasible directions. Let $\{s_i\}$ be the Steiner points of T and $\{s'_i\}$ the corresponding Steiner points of T'. Now suppose, over a given time interval, we move each Steiner point s_i at constant velocity from s_i to s'_i along the line segment between the two points. This gives us a continuum of λ-trees between T and T', each of which must be a λ-SMT for $\mathcal{T}$, by the convexity of the length function. There exists a tree T^* in this continuum representing a point at which the set of feasible directions changes. It follows that T^* either uses at least 5 directions, contradicting Corollary 3.2, or exactly 3 directions. But in the latter case, all perturbations of Steiner points in T^* are strictly length-increasing, by [3], again giving a contradiction. ∎

4 Length Preserving Shifts

We begin this section by defining the key concept of a 'shift' in λ-geometry. The idea of a shift was first introduced by [7], and the following definition is from [2, 10].

We define a *shift* of a straight edge pq to be a move of p to a new point $p' \neq p$ and a simultaneous move of q to a new point $q' \neq q$ such that $p'q' \parallel pq$. Similarly, a *shift* of a bent edge pq is defined to be a move of p to p' and a simultaneous move of q to q' such that $p'q'$ is either a bent edge with half-edges in the same directions as those in pq or a straight edge whose direction is the same as that of one of the half-edges of pq. The concept of a shift can be generalised to a path P in a full Steiner λ-tree T as follows. A *shift* of $P = ps_1s_2 \ldots q$ is a perturbation of T that moves the internal Steiner points s_i of P to s_i' (and fixes all other nodes of T) such that the following conditions are satisfied (see also Figure 5):

(1) each internal Steiner point s_i of P moves along the line through s_i containing the straight edge or half-edge of T incident to s_i not lying on P; and

(2) the shift of P induces a shift on each internal edge of P.

It is important to note that the effect of a shift is that it does not change the direction of any straight edge of T except possibly the first and last edges of P.

An example of a shift is given in Figure 5.

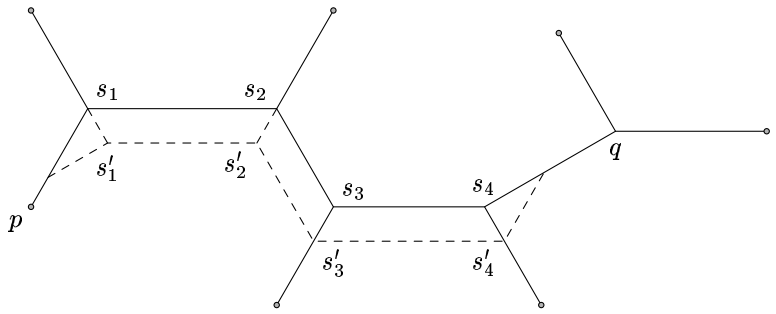

Figure 5: A shift of a path P in T.

Given a subpath of a full locally minimal Steiner λ-tree T, we define a shift on that subpath to be a 0-*shift* if the shift does not increase the length of T.

Theorem 4.1 *Given an exclusively primary edge or half-edge e_1 and an exclusively secondary edge or half-edge e_2 in a full λ-SMT, there exists a 0-shift on the path between e_1 and e_2.*

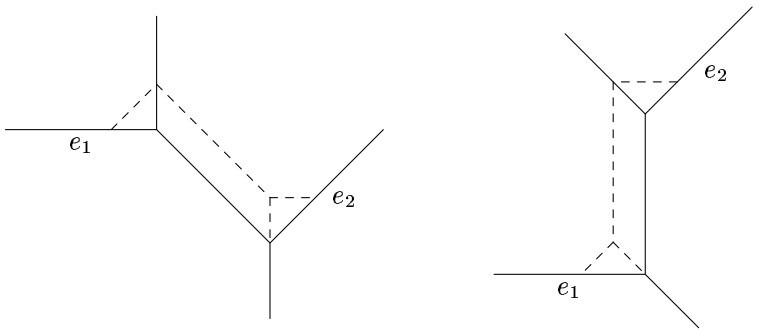

Figure 6: The base case for the inductive proof of Theorem 4.1. For a given set of feasible directions there are two possibilities for P, as shown. In each case it is clear, by inspection, that the indicated shift is a 0-shift.

Proof. The proof is by induction on the number of internal edges of the subpath P between e_1 and e_2. If P contains at most one internal edge then the theorem can be confirmed either by direct computation or using the results of [3]. This is illustrated for $\lambda = 4$ in Figure 6.

Now suppose P contains at least two internal edges. This means that P contains a Steiner point s which is not incident with either e_1 or e_2. Some edge or half edge e incident with s is either exclusively primary or exclusively secondary (by Corollary 3.2). Suppose e is exclusively secondary. Then, by the inductive assumption, there exists a 0-shift on the path between e_1 and e. Applying a small 0-shift effectively transfers an arbitrarily small exclusively primary component to e (see Figure 5). More specifically, it reduces the length of the exclusively primary edge or half-edge e_1, and adds an exclusively primary component to e. Again applying the inductive assumption, there exists a 0-shift on the path between the primary component of e and the edge e_2. Because the primary component of e can be assumed to be arbitrarily small, it follows that this second shift can transfer the entire primary component of e to e_2. Hence, together these two shifts give a 0-shift on the path between e_1 and e_2. ∎

The following two corollaries follow immediately from Theorem 4.1.

Corollary 4.2 *Given any two edges in a full λ-SMT with angle ω between them, there exists a 0-shift on the path between the two edges.*

A stronger corollary holds if $\lambda = 3m$.

Corollary 4.3 *Let $\lambda = 3m$, $m \geq 1$. Given any two edges in a full λ-SMT such that the angle between them is not a multiple of $2\pi/3$, there exists a 0-shift on the path between the two edges.*

We can use the concept of a 0-shift to resolve the question of precisely when degree 4 Steiner points can occur if $\lambda = 4$ or 6. Some properties of degree 4 Steiner points were established in [1] and [2], but the following theorem is a stronger result. Note that a full λ-SMT is said to be *fulsome* if every λ-SMT on the same set of terminals is full. Also, a full Steiner tree is said to be a *cross* if it contains no bent edges and exactly one Steiner point, and that Steiner point has degree 4 and all incident angles equal to $\pi/4$.

Theorem 4.4 *Let $\lambda = 4$ or 6. Suppose T is a a full and fulsome λ-SMT containing a Steiner point of degree 4. Then T is a cross.*

Proof. Consider first the case where $\lambda = 4$. Let s be the Steiner point of degree 4, and, in clockwise order, let v_1, v_2, v_3, v_4 be the four vertices adjacent to s. Suppose, contrary to the statement of the theorem, that at least one of these vertices, say v_1, is a Steiner point. We observe some basic properties of T (the first two of which are from [2]):

- Each edge sv_i is a straight edge.
- If v_i is a Steiner point, then the angle between $v_i s$ and each of the other edges incident with v_i is $3\pi/4$.
- Vertices v_2 and v_4 must be terminals. This follows from the fact that the shift in Figure 7(a) reduces the length of T.
- Let T_i ($i = 1, \ldots, 4$) be the subtrees resulting from decomposing T into four subtrees meeting at s, where each T_i is the subtree containing v_i. Then, for each i, the edges in T_i use at most three directions. To see why this is the case, note that if any T_i contained four directions, then, by Theorem 4.1, there would exist a 0-shift between two edges in T_i. This 0-shift could create a bent edge, contradicting Lemma 4.6 of [1].

Let v_5 and v_6 be the two vertices other than s adjacent to v_1. To obtain a contradiction to v_1 being a Steiner point, consider the shift shown in Figure 7(b), which moves v_1 to v_1' and splits s into two degree 3 points s_1' and s_2'. It is clear, by

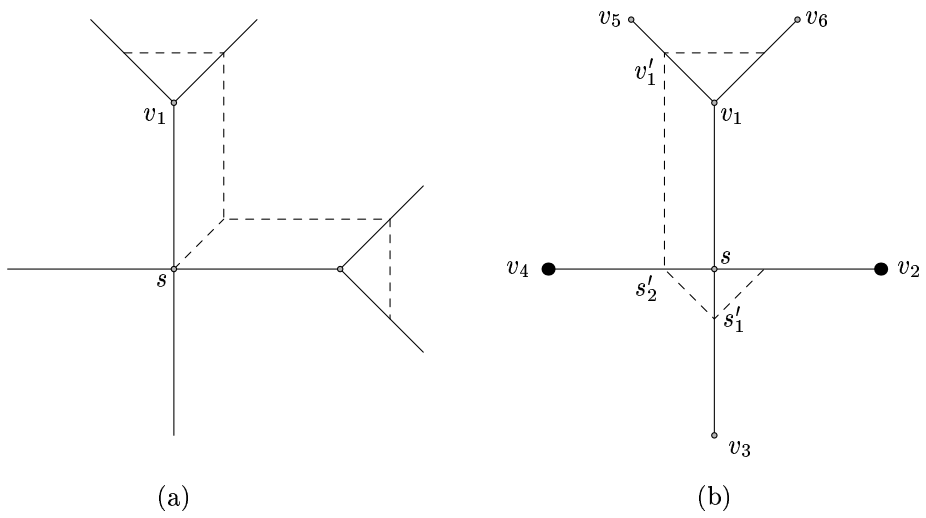

Figure 7: Proof of Theorem 4.4. The shift in (a) reduces the length of the tree, while the shift in (b) is a 0-shift.

inspection, that this is a 0-shift. We can continue to perform this shift without changing the length of T until one (or more) of the following occurs:

(1) s_2' coincides with v_4;

(2) s_1' coincides with v_3;

(3) v_1' coincides with v_5;

(4) $v_1'v_6$ or $s_1'v_2$ is a straight edge.

If (1) occurs we have a contradiction to T being fulsome. Similarly, if (2) occurs and v_3 is a terminal, or (3) occurs and v_5 is a terminal, T cannot be fulsome. If (3) occurs and v_5 is a Steiner point, then, since T_1 contains only three edge directions, one of the angles at v_5 is now $\pi/4$, contradicting minimality. If (4) occurs, then by sliding $v_1's_2'$ across to the right we obtain, by symmetry, the same contradiction as for cases (1) or (3). Finally, if (2) occurs and v_3 is a Steiner point then (assuming (4) does not occur) T now contains a degree 4 Steiner point and a bent edge, again contradicting Lemma 4.6 of [1]. This completes the proof for $\lambda = 4$.

If $\lambda = 6$ a similar argument applies. Crucially, we can once again decompose T into four subtrees meeting at s such that each subtree uses at most three directions (either the three primary directions or the three secondary directions). By applying the 0-shift in Figure 8, we can again obtain a contradiction to T being either fulsome or minimal. ∎

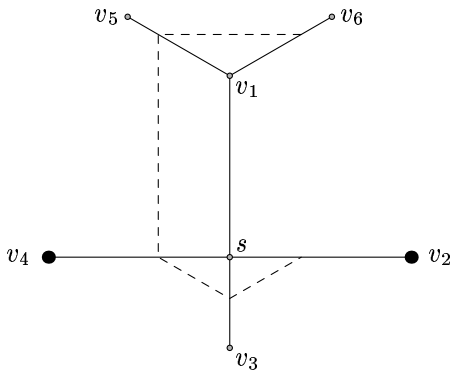

Figure 8: Proof of Theorem 4.4. A 0-shift at a degree 4 Steiner point for $\lambda = 6$.

5 Canonical Forms for full λ-SMTs

Our aim in this section is to develop a useful class of canonical forms for a λ-SMT with a given set of terminals N and a given full Steiner topology $\mathcal{T}$. The canonical forms are obtained from Theorem 4.1 and the geometric properties developed in the previous section. They will allow us to find efficient algorithms for constructing a λ-SMT for a given full Steiner topology in Section 6. We will also discuss some of the important structural properties that stem from an understanding of canonical forms.

We now define the concept of a canonical tree. Recall that a full λ-SMT is said to be *fulsome* if every λ-SMT on the same set of terminals is full. Suppose we are given a full and fulsome λ-SMT T for N with topology $\mathcal{T}$. Given an ordering of the edges of $\mathcal{T}$, T is said to be *canonical* with respect to that ordering if T contains at most one bent edge and all primary edges and half-edges of T come before secondary edges and half-edges of T under the ordering. In particular, this means that there exists an edge e that is either primary or bent, such that all edges that come before e under the ordering are primary, and all edges greater than e are secondary. The edge e will be referred to as the *transition edge*.

The next theorem is the main result for this section.

Theorem 5.1 *Let a set of n terminals N and a full Steiner topology $\mathcal{T}$ for that set of terminals be given. Suppose there exists a full and fulsome λ-SMT for N with topology $\mathcal{T}$. Then for any ordering of the edges of $\mathcal{T}$ there exists a λ-SMT for N (with topology $\mathcal{T}$) which is canonical with respect to that ordering.*

Proof. We give a constructive proof. Let T be a full λ-SMT for N with topology

$\mathcal{T}$. Since T is full, $\mathcal{T}$ contains $2n-3$ edges. Assign the natural numbers 1 to $2n-3$ to the edges of $\mathcal{T}$ to reflect the given ordering of edges. We can use the edge numbers of $\mathcal{T}$ to also number the corresponding edges of T, or any λ-SMT for N with topology $\mathcal{T}$.

We next show that we can iteratively apply Theorem 4.1 to transform T to another full λ-SMT for N with the same topology, which we call T', and which is canonical with respect to the given ordering. Suppose, on the contrary, there is an exclusively secondary edge or half-edge e_1 whose label is strictly less than that of an exclusively primary edge or half-edge e_2. Then, by Theorem 4.1, there exists a 0-shift on the path between these two edges. We can continue to perform this shift without changing the length of T until one of the following occurs:

(1) Two Steiner points coincide;

(2) A Steiner point coincides with a terminal;

(3) e_1 has no exclusively secondary component; or

(4) e_2 has no exclusively primary component.

Now note that if (1) occurs, we necessarily have a contradiction to the minimality of T, unless $\lambda = 4$ or 6 in which case, by Theorem 4.4, we have a contradiction to T being either fulsome or minimal (or T is a cross). If (2) occurs we have a contradiction to the fact that T is fulsome. So eventually either (3) or (4) must occur, which reduces the number of primary/secondary edge pairs where the primary edge has a higher number than the secondary edge. Hence, by iterative application of 0-shifts, we can construct a tree T', with the same length as T, such that no primary edge of T' has a higher number than any secondary edge. ∎

An important property of the canonical form is that the chosen numbering scheme cannot rely on the geometric embedding of an underlying canonical λ-SMT, but only on the topology $\mathcal{T}$. (However, if some λ-SMT for $\mathcal{T}$ is known, the edges may be numbered according to the geometric embedding of this λ-SMT.) The transition edge can occur anywhere within T, depending on the positions of the terminals. The canonical form will depend on the ordering of the edges of T, i.e., on how the natural numbers (1 to $2n-3$) are assigned to edges. Two examples of ordering that result in nice canonical forms are the following:

- *Breadth-first ordering*. In this ordering, $\mathcal{T}$ is rooted at a terminal, and the tree is then traversed in a breadth-first fashion. An edge is assigned the next available natural number when it is traversed. Hence, no secondary edge will be closer to the root than any primary edge.

- *Depth-first ordering*. In this ordering, $\mathcal{T}$ is rooted at a terminal, and the

tree is then traversed in a depth-first fashion. An edge is assigned the next available natural number when it is traversed.

A full λ-SMT T constructed using either breadth-first or depth-first orderings has the following important property: if we divide T into two subtrees by deleting the transition edge, then one of these subtrees contains only secondary edges (namely the subtree not containing the root). Furthermore, the path from the root to the transition edge contains primary edges only. Finally, in the subtree containing the root, any path from a terminal to the transition edge may change the category of its edges from secondary to primary (at most once), but never from primary to secondary. Some stronger structural properties of the canonical forms based on the depth-first ordering will be discussed in Section 6.3.

The significance of these observations is that, given an appropriate canonical form, a strong restriction on the structure of full λ-SMTs can be assumed. This property was recently incorporated into the first exact algorithm for the Steiner tree problem in uniform orientation metrics [11]. In this algorithm the possible full Steiner trees (or FSTs) of a λ-SMT are built up by constructing "half FSTs" with straight edges, which are either merged together at a Steiner point to form a larger half-FST, or at a corner point to form an FST. The numbering scheme — and thus the canonical form — that was chosen in the exact algorithm was the following:

1. Every FST was rooted at its lowest index terminal (as given by the input to the algorithm) and the edges were numbered according to a *depth-first* traversal from this root.

2. The children of a node were visited in the order given by their geometric location such that the leftmost child was visited before the rightmost child.

Note that in this case a numbering scheme that depended on the geometric embedding was possible since subtrees with known embedding were combined into larger subtrees; this again made it possible to assume that the transition edge occurred "outside" half-FSTs (see [11] for details). This canonical form had a significant impact on the running time of the exact algorithm. For $\lambda = 3m$ a speed-up of several orders of magnitude was obtained.

As a final application of the canonical forms developed in this section, we now give a stronger version of Corollary 3.2, that follows immediately from choosing a clever ordering of the edges in the topology.

Theorem 5.2 *Let T be a full and fulsome λ-SMT for terminal set N with topology $\mathcal{T}$. Then there exists a λ-SMT T' for terminal set N with topology $\mathcal{T}$ that uses at most 4 different directions.*

Proof. Order the edges of T by their *colour*; for example, the red edges may come first in the ordering followed by the green edges and then the blue edges. Consider a canonical λ-SMT T' that comes from this ordering (by Theorem 5.1 such a tree exists). Then it follows from the canonical form that the edges of any given colour, not the same as the colour of the transition edge, are either all primary or all secondary. Hence T' uses at most 4 directions. ∎

6 Quadratic and Linear Time Algorithms

In this section we present two algorithms for constructing a λ-SMT for a given set of n terminals N and a given topology $\mathcal{T}$. Both algorithms depend strongly on a so-called *merging* operation that can be performed in constant time (Section 6.1). The first simple algorithm that we present runs in $O(\lambda n^2)$ time (Section 6.2), while the second runs in $O(\lambda n)$ time – which is optimal for fixed λ (Section 6.3).

6.1 Constant-Time Merging Operation

The following lemma establishes conditions under which we can uniquely construct a Steiner point adjacent to two given points. The operation of constructing such a Steiner point and effectively replacing the two given points by this Steiner point, will be referred to, throughout the remainder of this paper, as a *merging* operation.

Lemma 6.1 *Let T be a full λ-SMT with topology $\mathcal{T}$, and let s be a Steiner point in $\mathcal{T}$. Assume that the locations in T of two of the neighbours of s, u and v, are known; furthermore, assume that each edge (u, s) and (v, s) is straight and has been labeled primary or secondary. Finally, assume that we know the direction set of edges. If Steiner point s exists in T, then its location is unique; also, the colours of the edges (u, s) and (v, s) — and hence also the colour of the third edge incident with s — are unique. The existence and the location of s can be determined in constant time.*

Proof. For the given direction set, we first make the following observations, which follow immediately from Corollary 3.2, and our assumption that $\lambda > 3$:

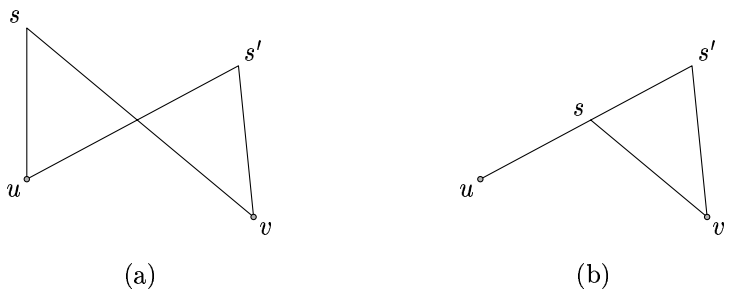

Figure 9: Proof of Lemma 6.1.

- The smaller angle between any two directed straight edges of different colours is at least $\pi/2$, and equality can only occur if one of the edges is exclusively primary and the other exclusively secondary;
- Furthermore, the smaller angle between any two directed straight edges is strictly less than π.

Now suppose, contrary to the statement of the lemma, that it is possible to construct two distinct Steiner points s and s' adjacent to both u and v. We consider a couple of different cases. First, assume that the interiors of the edges (u, s), (v, s), (u, s') and (v, s') are all disjoint. Since (u, s) and (u, s') are either both primary or both secondary it follows from the previous observation that the sum of interior angles in the quadrilateral $usvs'$ is strictly greater than 2π, which gives a contradiction. Similarly, if the interiors of two of the edges, say (u, s) and (v, s'), intersect at a single point (as in Figure 9(a)), then the sum of the angles at u, s', s and v is again greater than 2π, but must be less than the sum of interior angles in the quadrilateral $uss'v$, giving another contradiction.

The only remaining possibility is that s and s' are both collinear with exactly one of the points u and v, say u (as in Figure 9(b)). Since (v, s) and (v, s') are either both primary or both secondary, it follows that the two edges have different colours, while the edge (u, s') must have the third colour. Hence the sum of the angles between (v, s) and (v, s') and between (s', s) and (s', v) must be strictly greater than π, giving a contradiction.

We conclude that at most one Steiner point can be constructed.

Finally, note that this construction can be done in constant time, since the direction set is known. ∎

6.2 Quadratic Time Algorithm

We are given a set of n terminals N and a full Steiner topology $\mathcal{T}$ for the set of terminals. We would like to construct a full and fulsome λ-SMT T for N with topology $\mathcal{T}$ (or prove that no such λ-SMT exists). In this section we assume that the topology $\mathcal{T}$ is given by any sparse graph representation.

First we choose an ordering of the $2n-3$ edges in $\mathcal{T}$. This ordering is completely arbitrary, and it is clearly possible to find an ordering in $O(n)$ time given any sparse graph representation of $\mathcal{T}$. The algorithm presented in this section constructs a λ-SMT having the canonical form given by this ordering.

Let us first assume that we are know the direction set *and* the identity of the transition edge for some canonical λ-SMT T under the given ordering of the edges; assume that the transition edge has number k under the chosen ordering, where $1 \leq k \leq 2n-3$. Then, in $O(n)$ time, we can construct T as follows.

Label all edges numbered less than k as primary edges and label all edges numbered greater than k as secondary edges. We root $\mathcal{T}$ at the transition edge and iteratively apply the merging operation to leaf nodes sharing a parent until the locations of the endpoints of the transition edge have been constructed. Since the primary/secondary labelling of every edge except the transition edge is known, every merging operation has a unique solution and can be performed in constant time by Lemma 6.1.

Since there are up to 2λ different directions sets and $2n-3$ different choices for the transition edge, iterating over all possible combinations and using the $O(n)$ algorithm just described will clearly construct a full and fulsome λ-SMT for N and $\mathcal{T}$ (if any such λ-SMT exists). The result is summarized in the following theorem:

Theorem 6.2 *Let a set of n terminals N and a full Steiner topology $\mathcal{T}$ for that set of terminals be given. Suppose there exists a full and fulsome λ-SMT for N with topology $\mathcal{T}$. Then such a tree can be found in $O(\lambda n^2)$ time.*

6.3 Linear Time Algorithm

In this section we develop a linear time algorithm for constructing a λ-SMT for a given full Steiner topology. More precisely, for a given direction set of edges, the algorithm runs in $O(n)$ time. By iterating over the (up to) 2λ different edge direction sets, a λ-SMT for the full topology (if it exists) will be constructed in $O(\lambda n)$ time.

The given full Steiner topology is assumed to be a rooted tree $\mathcal{T}$ with node set V, where $N \subseteq V$. The tree $\mathcal{T}$ is rooted at a terminal $r \in V$ and all other terminals are leaves in $\mathcal{T}$. All internal nodes (Steiner points) have degree 3. For each internal node $v \in V$, the children of v are denoted by $\mathrm{L}[v]$ and $\mathrm{R}[v]$; the child of the root node r is denoted by $\mathrm{L}[r]$. For each node $v \neq r$ the parent of v is denoted by $\mathrm{P}[v]$.

In the algorithm we will initially label all edges in $\mathcal{T}$ as being *secondary*. Instead of storing the primary/secondary labelling with the edges, it turns out to be convenient to store this information with the nodes. We let $\mathrm{PS}[v]$ indicate the primary/secondary labelling of the node v; it will later become clear how this node information is used to label the edges.

We will use *rays* to represent subtrees of the λ-SMT that is to be constructed. A ray is an oriented halfline with a source (point in the plane) and a direction. There is one ray $\Phi[v]$ for each node in $\mathcal{T}$ and initially all rays are *undefined*. The source of a node ray will correspond to the current geometric location of the node while the direction will correspond to one of the three directions in the direction set having the labeling given by $\mathrm{PS}[v]$. Informally, when a λ-SMT for $\mathcal{T}$ has been constructed, all node rays will point towards the transition edge (or bent edge) in the tree.

The linear time algorithm consists of two depth-first traversals of $\mathcal{T}$. Both traversals use an operation that merges two rays into one; this operation is described next.

Merging Rays

The basic constant time operation that we will use is denoted $\mathsf{MergeRays}(x, y)$. This operation assumes that x and y are neighbours to a common Steiner point v, and that the primary/secondary labels of the edges (x, v) and (y, v) are given by $\mathrm{PS}[x]$ and $\mathrm{PS}[y]$, respectively, and that both edges are to become straight edges. We distinguish between two cases:

Both x and y are terminals By the proof of Lemma 6.1 the location of Steiner point v is unique and it can be constructed in constant time (if it exists). If the Steiner point v exists, the result of $\mathsf{MergeRays}(x, y)$ is the ray having its source in v, and direction of the unique third edge out of v with primary/secondary label $\mathrm{PS}[v]$ — otherwise the result of $\mathsf{MergeRays}(x, y)$ is undefined.

Either x or y (or both) are Steiner points First assume that x is a terminal (the

case where y is a terminal is similar). If $\Phi[y]$ is undefined, the result of MergeRays(x, y) is undefined. Otherwise, again if Steiner point v exists, the result of MergeRays(x, y) is the ray having its source in v, and direction of the unique third edge out of v with primary/secondary label PS$[v]$ — otherwise the result of MergeRays(x, y) is undefined. If both x and y are Steiner points the resulting ray has its source at the intersection between $\Phi[x]$ and $\Phi[y]$; if $\Phi[x]$ or $\Phi[y]$ are undefined or they do not intersect, the result of MergeRays(x, y) is undefined.

First Traversal of $\mathcal{T}$

In the following we let T_v denote some constructed subtree of $\mathcal{T}$ rooted at v. A *primary* (or *secondary*) subtree T_v is defined to be a tree that contains primary (respectively, secondary) edges only. When constructing a primary (respectively, secondary) subtree T_v, $\Phi[v]$ has its source located at the constructed root of T_v and the direction of $\Phi[v]$ corresponds to that of the primary (respectively, secondary) edge from v to its parent, that is, it will be directed towards the root of $\mathcal{T}$.

In the first traversal of $\mathcal{T}$ we attempt to construct secondary subtrees for each node $v \neq r$; we assume that PS$[v]$ = *secondary* for all nodes $v \neq r$. This is done bottom-up by applying the recursive algorithm in Figure 10 to the edge $(r, \mathrm{L}[r])$. If a secondary subtree rooted at v cannot be constructed, $\Phi[v]$ is undefined when the traversal finishes.

```
Traverse(u, v)
1  if (second traversal) then TryBentEdge(u, v)
2  if (v is a Steiner point) then
3     Traverse(v, L[v])
4     Traverse(v, R[v])
5     Φ[v] = MergeRays(L[v], R[v])
```

Figure 10: Traversal of full Steiner topology $\mathcal{T}$.

Second Traversal of $\mathcal{T}$

In the second traversal of $\mathcal{T}$ (again by the algorithm shown in Figure 10), the secondary subtrees constructed in the first traversal are merged with primary subtrees. More specifically, we try to construct all possible trees having the canonical

form which comes from the edge ordering given by the traversal: there exists a bent edge (u, v) such that all edges visited before (u, v) are *primary* edges while all edges visited after (u, v) are *secondary* edges.

The TryBentEdge(u, v) procedure (Figure 11) processes each edge (u, v) in the order given by the traversal. In line 1 the label of node u is changed to primary (the node v is still labelled secondary at this point). If $u = r$ then we let $\Phi[u]$ have the same colour as $\Phi[v]$ (line 3), such that $\Phi[u]$ and $\Phi[v]$ (if defined) will correspond to two half edges of a bent edge. If the rays intersect (lines 9–10), a candidate tree has been constructed.

```
TryBentEdge(u, v)
 1  PS[u] = primary
 2  if (u = r) then
 3     Let Φ[u] be the ray with source at u having the same colour as Φ[v]
 4  else
 5     Let x = P[u] and y be the two other neighbours of Steiner point u
 6     Φ[u] = MergeRays(x,y)
 7  if (v is a terminal) then
 8     Let Φ[v] be the ray with source at v having the same colour as Φ[u]
 9  if (Φ[u] and Φ[v] have the same colour and intersect each other) then
10     Intersection point c is corner point in constructed tree
11  PS[v] = primary
```

Figure 11: Trying (u, v) as a bent edge.

If u is a Steiner point, its geometric location is first found by merging the rays of its neighbours other than v (lines 5–6). Then the intersection between $\Phi[u]$ and $\Phi[v]$ is computed as before (lines 9–10). A special case occurs when v is a terminal (lines 7–8): In this case we let $\Phi[v]$ have the same colour as $\Phi[u]$, similar to the $u = r$ case. Finally, the label of node v is changed to primary in line 11.

The length of the constructed tree can be computed in constant time if the length of each subtree is maintained during the execution of the algorithm. This is most easily achieved by letting the merge operation compute the sum of the lengths stored in the merged nodes and adding this to the distances from their root nodes to the new intersection point.

For a given direction set, the algorithm clearly runs in $O(n)$ time: Two traver-

sals of $\mathcal{T}$ are performed and the processing of each edge takes constant time. By iterating over all direction sets, a λ-SMT for $\mathcal{T}$ (if it exists) will be constructed in $O(\lambda n)$ time.

Correctness of Linear Time Algorithm

In the following we assume that a λ-SMT T^* for $\mathcal{T}$ exists and that it uses the current edge direction set. Furthermore, we assume that for the chosen canonical form (edge ordering), the bent edge is (u, v) with corner point c (which may coincide with u or v if T^* contains no bent edge). We will now prove that the algorithm does construct T^*.

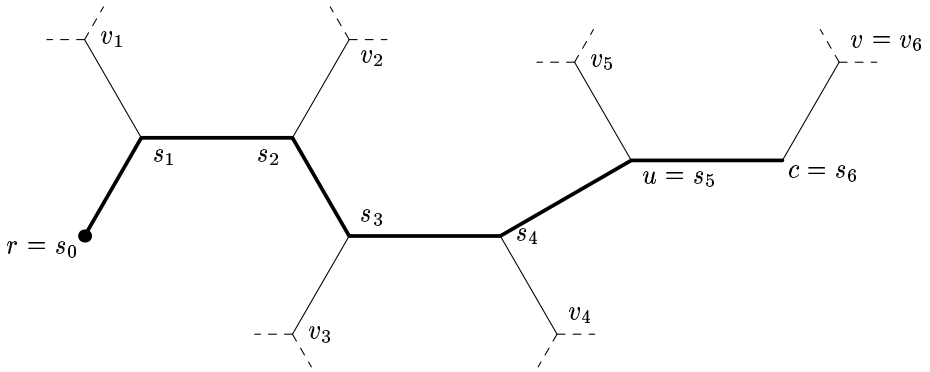

Figure 12: Canonical form used by linear time algorithm.

In the following discussion, we say that a ray *overlaps* with a line segment if the ray is contained in the line representing the extension of that line segment.

Consider the path P in T^* from the root r to the corner point c. Let $s_0 = r, s_1, \ldots, s_{k-1} = u, s_k = c$ be the nodes on this path. For each $i = 1, \ldots, k$, let v_i denote the node adjacent to s_i but not on P, and let T_i denote the largest subtree of T^* rooted at s_i and not containing any edges of P (Figure 12).

Lemma 6.3 *When the algorithm processes edge (u, v) in the second traversal, for each $i = 1, \ldots, k$, we have that*

i) edge (s_{i-1}, s_i) is a primary edge, and T_i is either a primary or secondary subtree,

ii) the location of the source of ray $\Phi[v_i]$ is identical to the location of v_i in T^*, and $\Phi[v_i]$ overlaps with the edge (v_i, s_i) in T^*,

iii) the location of the source of ray $\Phi[s_{i-1}]$ is identical to the location of s_{i-1} in T^*, and $\Phi[s_{i-1}]$ overlaps with the edge (s_{i-1}, s_i) in T^*.

Proof. Property i) follows from the depth-first ordering of the traversal, while property ii) follows from the definition of the merge ray operation and from the fact that the secondary (respectively, primary) subtrees are constructed bottom-up in the first (respectively, second) traversal of $\mathcal{T}$.

When the edge (s_{i-1}, s_i) is processed by the algorithm, all the subtrees $T_1, \ldots, T_{i-1}$ have been constructed and remain the same until edge (u, v) is processed. This follows from the depth-first order of the traversal.

The edges from the root to (u, v) are visited in the order they appear on the path. Assume that nodes $s_1, \ldots, s_{i-2}$ already have been correctly constructed when edge (s_{i-1}, s_i) is processed. (This is certainly true for $i = 2$.) Lines 5-6 in TryBentEdge(s_{i-1}, s_i) merge the rays out of s_{i-2} and v_{i-1}. Since both nodes have the correct location and rays overlapping with the edges adjacent to s_{i-1}, the resulting ray $\Phi[s_{i-1}]$ will be located at the correct position and overlaps with (s_{i-1}, s_i) in T^*. ∎

Theorem 6.4 *When the algorithm processes edge (u, v) in the second traversal, the λ-SMT T^* is correctly constructed.*

Proof. When (u, v) is processed it follows from Lemma 6.3 that $\Phi[u]$ becomes correctly constructed and overlaps with the half edge from u to the corner point of the bent edge. Furthermore, the subtree T_k is a secondary subtree and $\Phi[v]$ has been correctly constructed in the first traversal. The theorem follows since TryBentEdge(u, v) in lines 9-10 computes the intersection between the rays $\Phi[u]$ and $\Phi[v]$. ∎

Implementation

The linear time algorithm is fairly simple to implement. In our implementation a ray $\Phi[v]$ for a node v was represented by the x- and y-coordinates of the source and a number between 0 and $2\lambda - 1$, corresponding to the (up to) 2λ different legal directions extending from the source. All numerical computations can be performed in the ray merging operation. In particular, only the ray merge operation depends on the data types used for representing coordinates. The remaining code can be written more or less exactly as presented in Figures 10 and 11.

Our experiments with the algorithm show that it is very fast and that the constants involved are small. However, a simple exact algorithm that enumerates

all subsets and all full Steiner topologies can only solve problems with less than 10 terminals, in a reasonable amount of time. This is due to the large number of possible subsets and large number of different full Steiner topologies for each subset (e.g., for a subset with 8 terminals there are 10395 different full Steiner topologies).

7 Concluding Remarks

In this paper we presented a powerful canonical form for uniformly-oriented Steiner trees in the plane, and gave a linear time algorithm for constructing a λ-SMT for a given full Steiner topology. The canonical forms have already proved to be very useful in the implementation of an exact algorithm for the problem [11], resulting in substantial speed-up and making it feasible to solve large-scale problems to optimality. We also believe that the canonical forms and the linear time algorithm will turn out to be very useful in the design of fast heuristics for the problem.

Acknowledgments

The authors would like to thank Benny K. Nielsen and Pawel Winter for many fruitful discussions. Also we thank the anonymous referees for their comments and suggestions.

References

[1] M. Brazil. Steiner Minimum Trees in Uniform Orientation Metrics. In D.-Z. Du and X. Cheng, editor, *Steiner Trees in Industries*, pages 1–27. Kluwer Academic Publishers, 2001.

[2] M. Brazil, D. A. Thomas, and J. F. Weng. Minimum Networks in Uniform Orientation Metrics. *SIAM Journal on Computing*, 30:1579–1593, 2000.

[3] M. Brazil, D. A. Thomas, and J. F. Weng. Locally Minimal Uniformly-Oriented Steiner Trees. Technical report, University of Melbourne, 2003.

[4] F. K. Hwang. On Steiner Minimal Trees with Rectilinear Distance. *SIAM Journal on Applied Mathematics*, 30:104–114, 1976.

[5] F. K. Hwang. A Linear Time Algorithm for Full Steiner Trees. *Operations Research Letters*, 4(5):235–237, 1986.

[6] F. K. Hwang, D. S. Richards, and P. Winter. *The Steiner Tree Problem*. Annals of Discrete Mathematics 53. Elsevier Science Publishers, Netherlands, 1992.

[7] D. T. Lee and C. F. Shen. The Steiner Minimal Tree Problem in the λ-Geometry Plane. In *ISAAC'96, Lecture Notes in Computer Science 1178*, pages 247–255, 1996.

[8] Y. Y. Li, S. K. Cheung, K. S. Leung, and C. K. Wong. Steiner Tree Constructions in λ_3-Metric. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 45(5):563–574, 1998.

[9] Y. Y. Li, K. S. Leung, and C. K. Wong. Efficient Heuristics for Orientation Metric and Euclidean Steiner Tree Problems. *Journal of Combinatorial Optimization*, 4:79–98, 2000.

[10] G.-H. Lin and G. Xue. The Steiner Tree Problem in λ_4-Geometry Plane. In *ISAAC'98, Lecture Notes in Computer Science 1533*, pages 327–337, 1998.

[11] B. K. Nielsen, P. Winter, and M. Zachariasen. An Exact Algorithm for the Uniformly-Oriented Steiner Tree Problem. In *Proceedings of the 10th European Symposium on Algorithms, Lecture Notes in Computer Science*, volume 2461, pages 760–772. Springer, 2002.

[12] B. K. Nielsen, P. Winter, and M. Zachariasen. On the Location of Steiner Points in Uniformly-Oriented Steiner Trees. *Information Processing Letters*, 83:237–241, 2002.

[13] M. Sarrafzadeh and C. K. Wong. Hierarchical Steiner Tree Construction in Uniform Orientations. *IEEE Transactions on Computer-Aided Design*, 11:1095–1103, 1992.

[14] A. C. Thompson. *Minkowski Geometry: Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 1996.

[15] P. Widmayer, Y. F. Wu, and C. K. Wong. On Some Distance Problems in Fixed Orientations. *SIAM Journal on Computing*, 16(4):728–746, 1987.

[16] G. Xue and K. Thulasiraman. Computing the Shortest Network under a Fixed Topology. *IEEE Transactions on Computers*, 51:1117–1120, 2002.