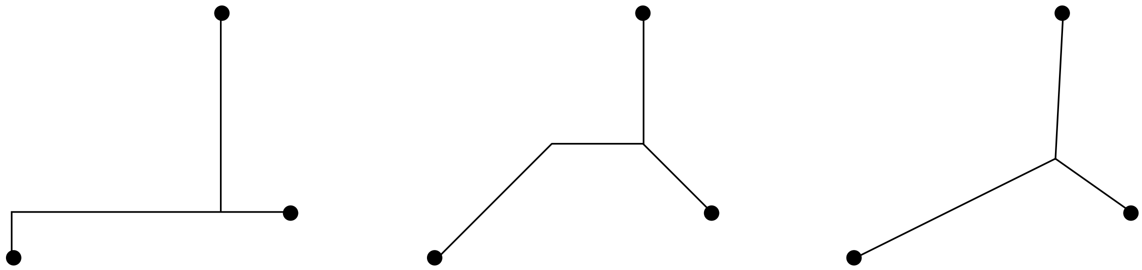
Algorithms for Steiner Trees in Uniform Orientation Metrics



Martin Zachariasen
University of Copenhagen

Outline

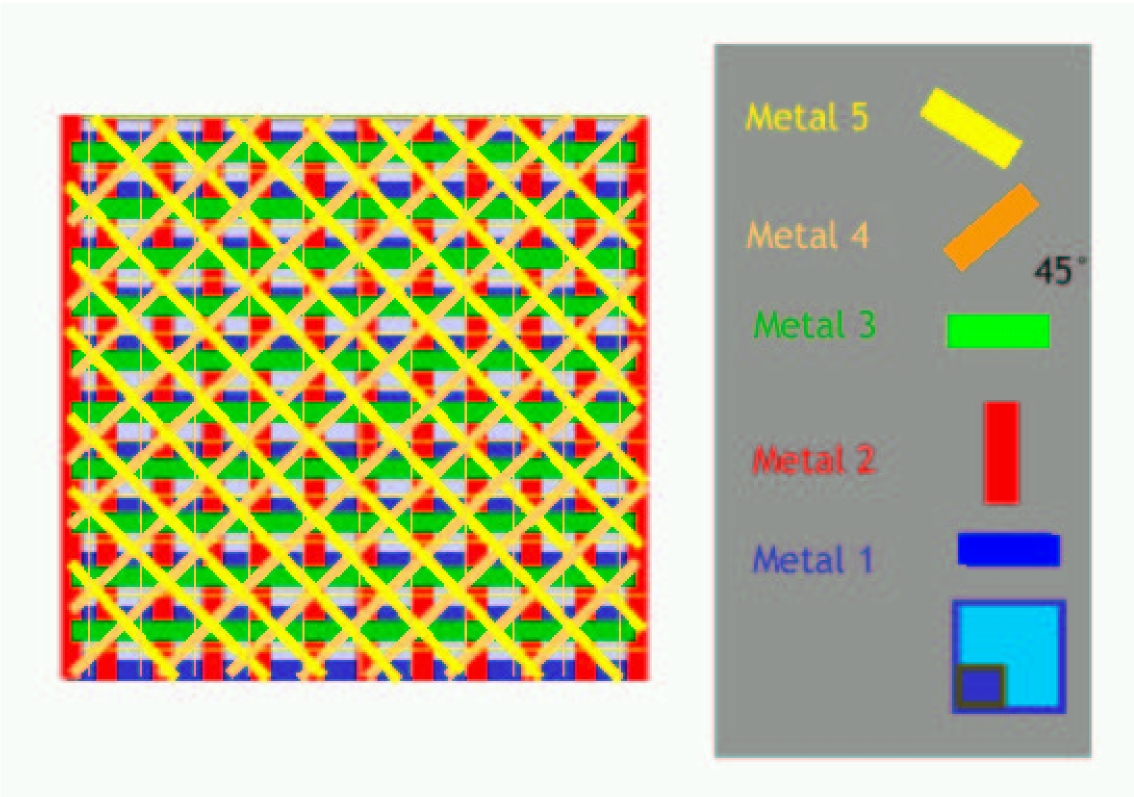
- Motivation and Background
- Fixed and Uniform Orientation Metrics
- Steiner Trees in Uniform Orientation Metrics
- Edge Directions
- Length Preserving Shifts
- Steiner Trees for a Given Topology
- Exact Algorithm for General Problem
- Concluding Remarks

Motivation and Background

Motivation: Routing in VLSI design

- Manhattan architecture:
  Horizontal and vertical wires only

- X architecture:
  Horizontal, vertical and diagonal wires



X Architecture

Introduced in June 2001 by a consortium of chip manufacturers and chip software companies
$\longrightarrow$ `www.xinitiative.org`

Promises 20% reductions in wire length as a result of the use of diagonal routing — however, this requires placement and routing algorithms that take full advantage of the X architecture.

> "On every meaningful measure of layout quality, the X architecture is superior to the Manhattan architecture, which is why we expect that five years from today, virtually all high-end chips will use X."
> [Teig, 2002]

Fixed Orientation Metrics

Introduced by [Widmayer, Wu & Wong, 1987].

Given a set A of at least two distinct orientations in the plane.

Orientation = angle with the x-axis of corresponding straight line.

A-oriented line segment/line: The orientation of the segment/line is in A.

A-distance: Euclidean length of a shortest (zig-zag) path in which each line segment is A-oriented.

- $|pq|$: Euclidean distance between p and q
- $|pq|_A$: A-distance between p and q

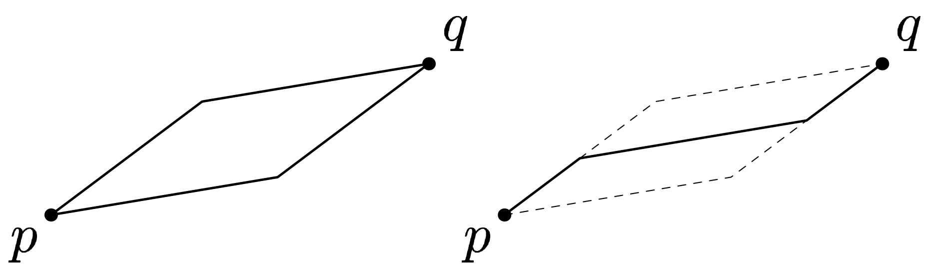
Fixed Orientation Metrics: Basic Results

---

1. The A-distance induces a metric for any given set A.

2. For any two points p and q does there exist a point r such that

$$|pq|_A = |pr| + |rq|$$



3. For any A does there exist a constant c_A such that

$$|pq|_A \leq c_A|pq|$$

Uniform Orientation Metrics (λ-Metrics)

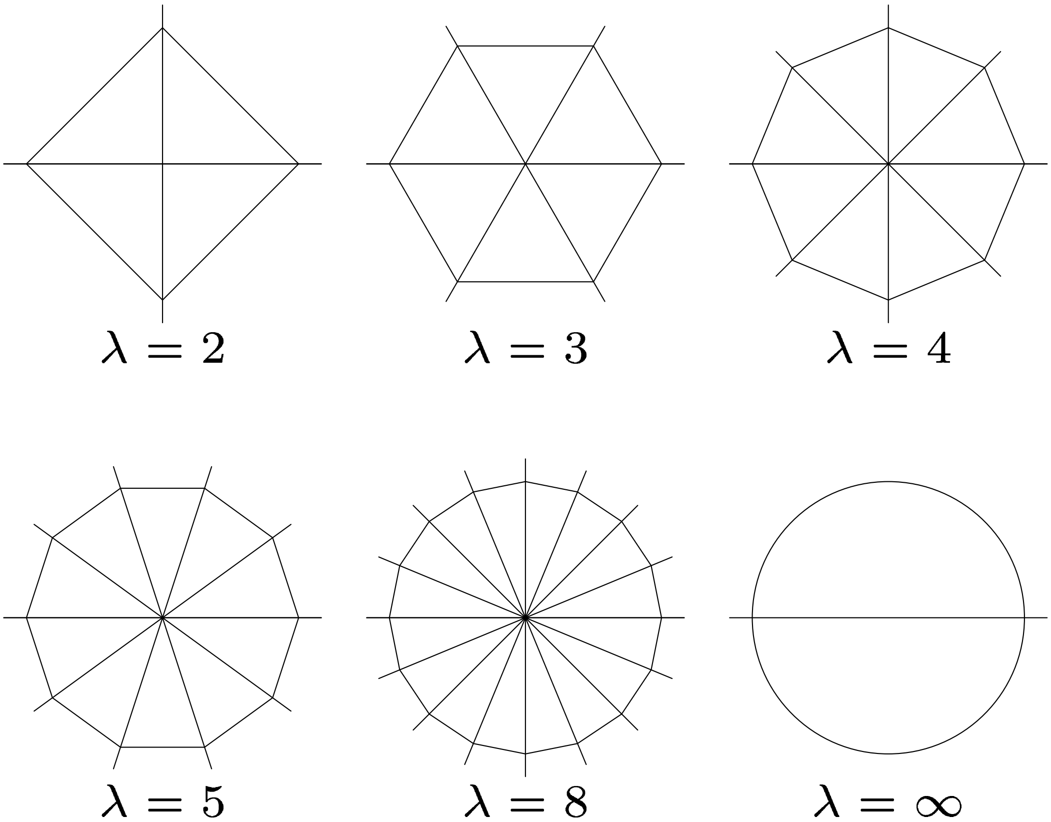
For $\lambda \geq 2$ consider fixed orientation metric given by the orientations

$$A_\lambda = \{\omega, 2\omega, \ldots, \lambda\omega\}$$

where $\omega = \pi/\lambda$.

λ-oriented line segment/line: The orientation of the segment/line is in A_λ.

λ-distance = A_λ-distance = Euclidean length of a shortest (zig-zag) path in which each line segment is λ-oriented.



Steiner Trees in Uniform Orientation Metrics (λ-SMTs)

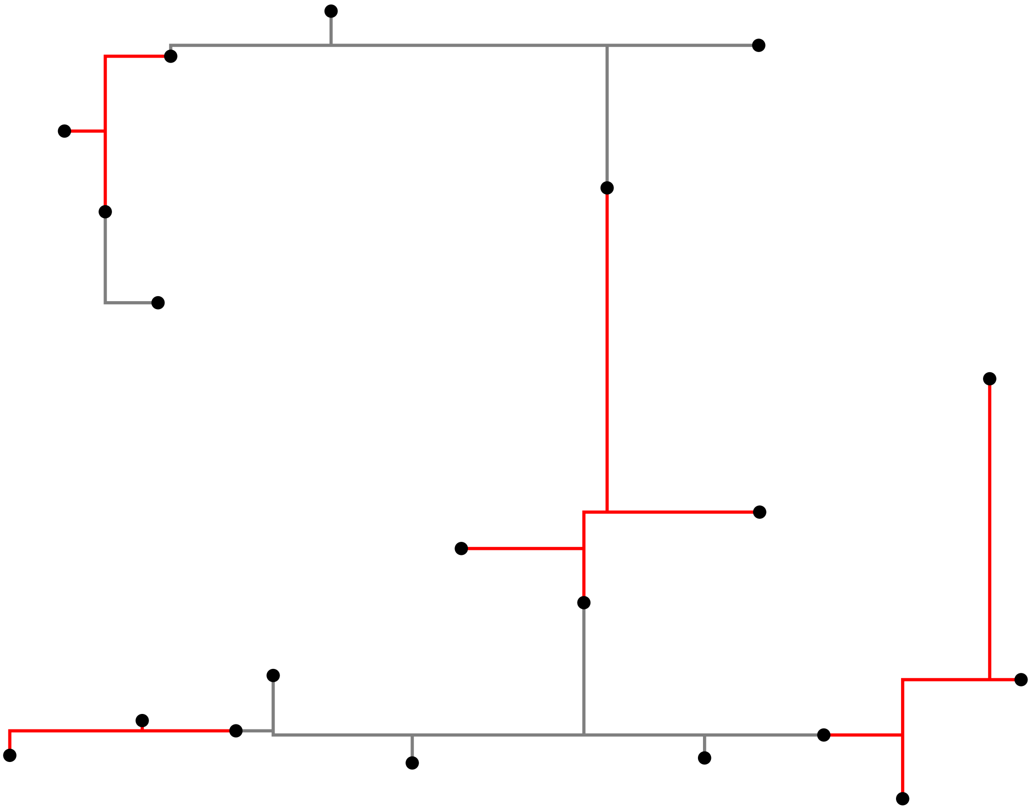
Definition: Given $\lambda \geq 2$ and a set N of terminals in the plane, find a shortest interconnection of the terminals under the λ-metric.

Complexity: NP-hard since the rectilinear Steiner tree problem ($\lambda = 2$) is a special case [Garey & Johnson, 1977].

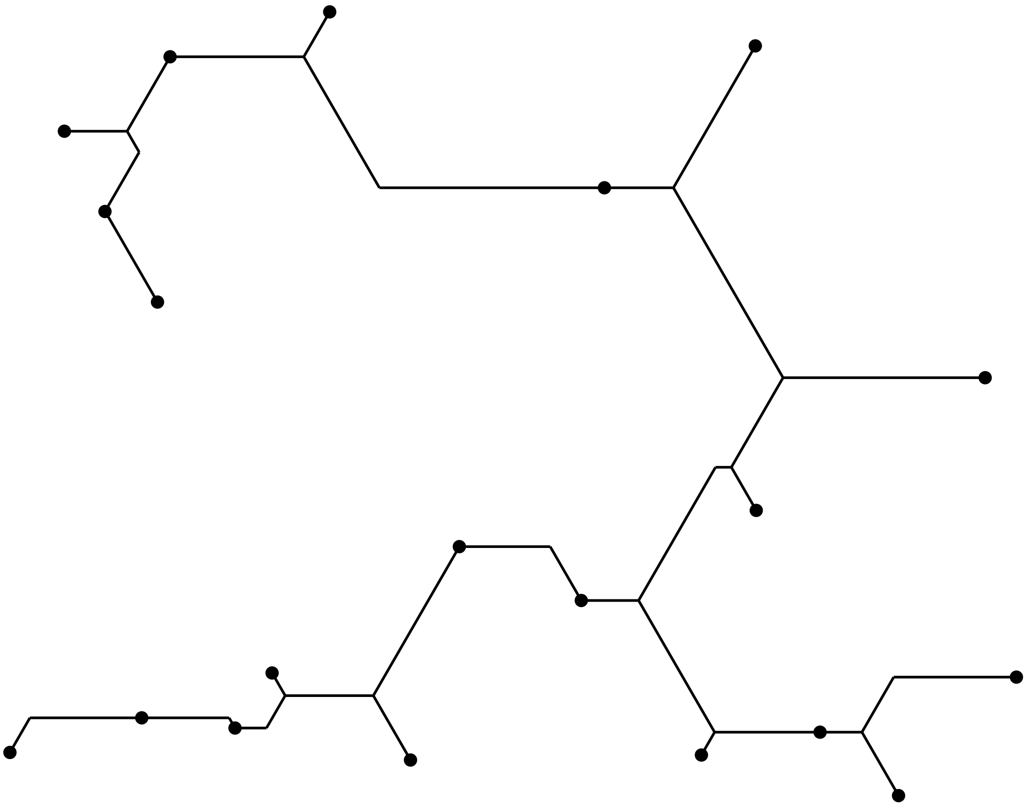
Approximation: Has a polynomial-time approximation scheme [Arora, 1996]; works since the λ-distance is within a constant factor of the Euclidean distance.

Applications: VLSI design, printed circuit board layout etc.

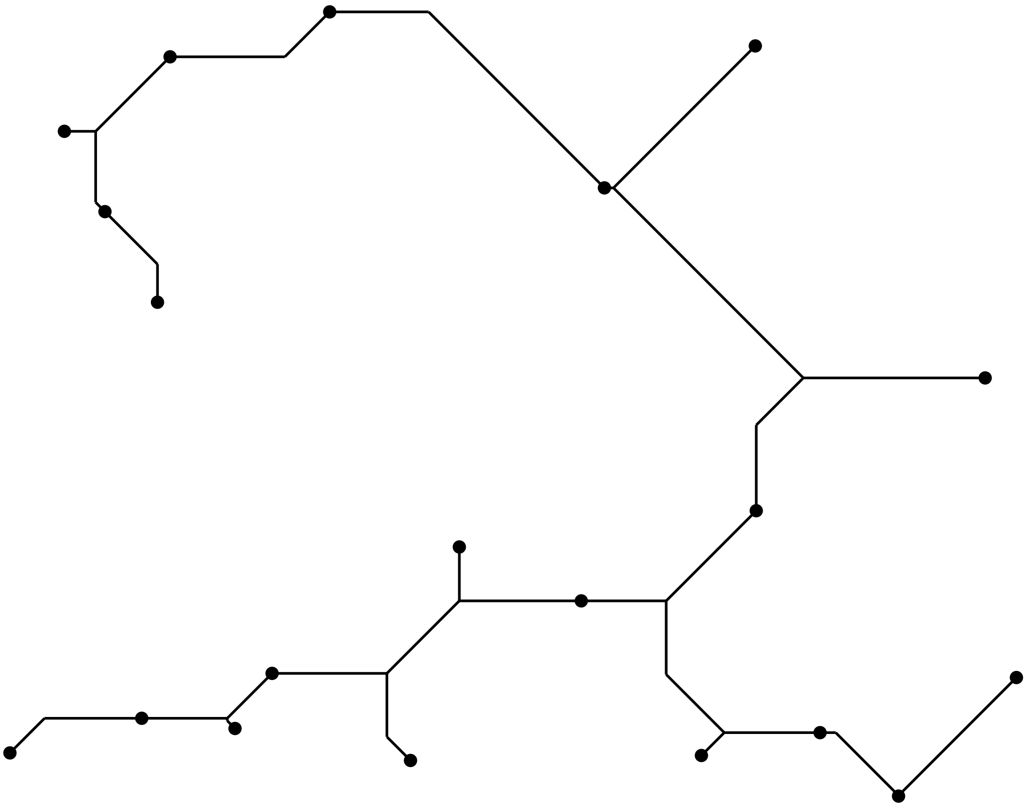
Rectilinear Steiner Tree ($\lambda = 2$)



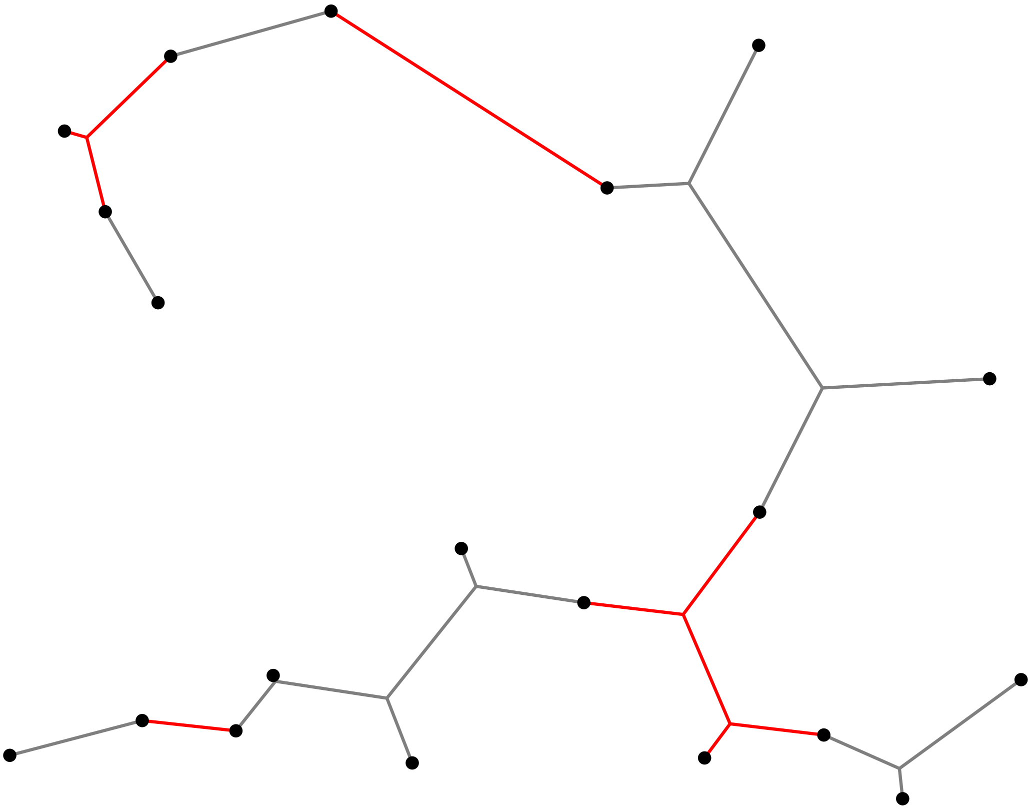
Hexagonal Steiner Tree ($\lambda = 3$)



Octilinear Steiner Tree ($\lambda = 4$)



Euclidean Steiner Tree ($\lambda = \infty$)



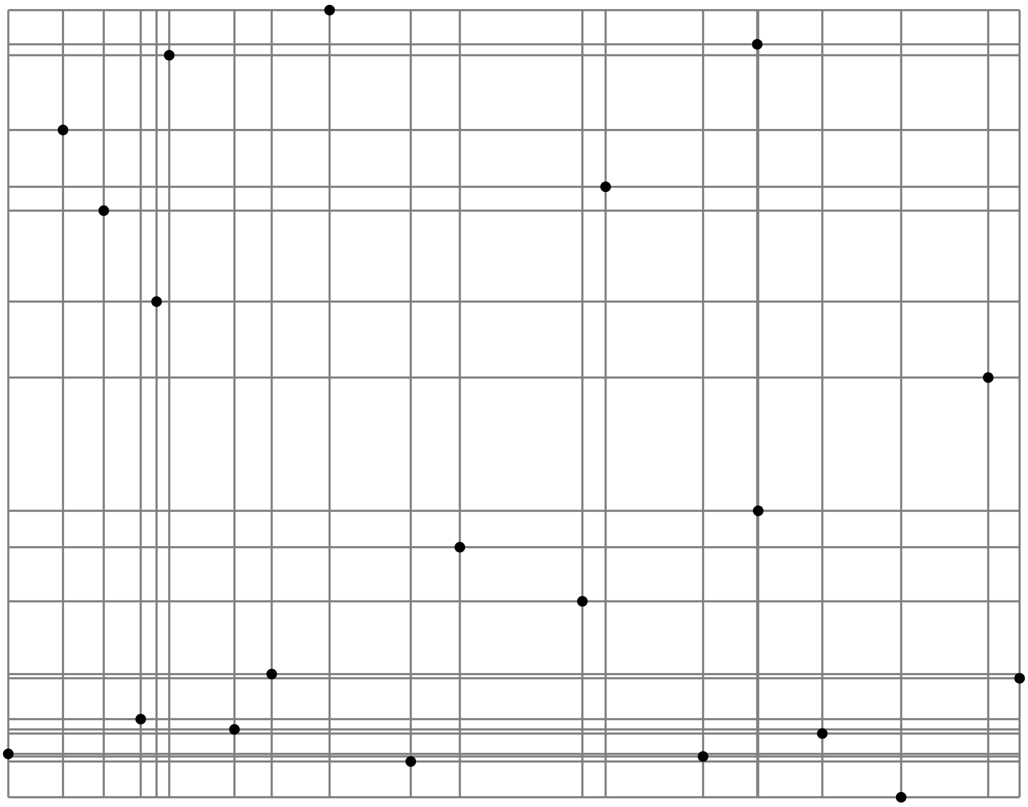
Full Steiner Trees and Fulsome λ-SMTs

Full Steiner tree (FST): Steiner tree in which all terminals are leaves (and all Steiner points are interior nodes). A λ-SMT is a union of FSTs.

Fulsome λ-SMT: A λ-SMT for which the number of FSTs is maximized. In particular, no FST can be split into two or more FSTs.

Canonical λ-SMT: Any characterization of λ-SMTs which reduces the set of optimal solutions; for $\lambda = 2$ one canonical form is the Hwang FST topology.

The Hanan Grid



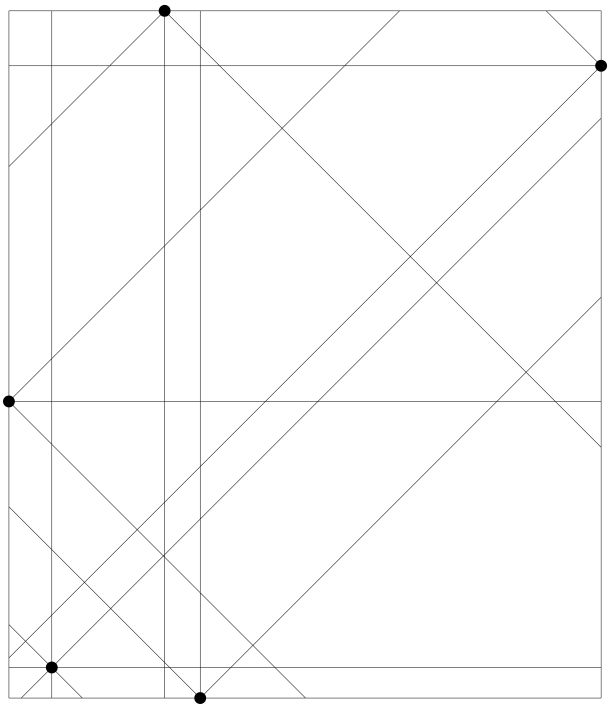
Theorem [Hanan, 1966]
There exists a rectilinear SMT for which all Steiner points are vertices in the grid graph for N.

Multi-level Grids

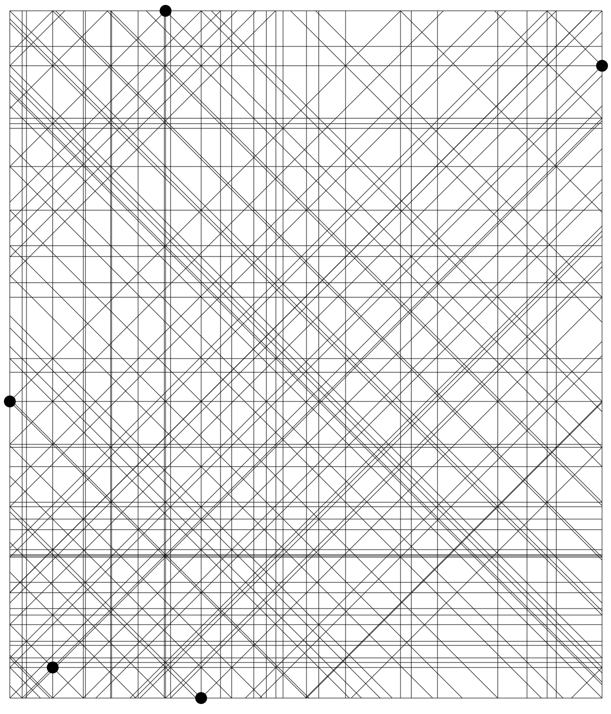
For a given set A of orientations recursively define

$GG_0(N) = N$

$i \geq 1$: $GG_i(N)$ = Intersections of all A-oriented lines through all points in $GG_{i-1}(N)$.



$GG_1(N)$



$GG_2(N)$

Note that for $\lambda = 2$ we have that $GG_1(N)$ is identical to the intersections of the Hanan grid for N.

Basic Structural Properties

Theorem [Brazil, Thomas & Weng, 2000]
The minimum angle at a Steiner point is $\lceil 2\lambda/3 - 1 \rceil \omega$ while the maximum angle is $\lfloor 2\lambda/3 + 1 \rfloor \omega$.
(Note that $(2\lambda/3)\omega = 2\pi/3 = 120^\circ$).

Corollary
The maximum degree of any vertex in a λ-SMT is 3, except when $\lambda = 2, 3, 4$ or 6.

Theorem [Brazil, Thomas & Weng, 2000]
For every set N of terminals there exists a λ-SMT for N such that each of its FSTs has at most one bent edge.

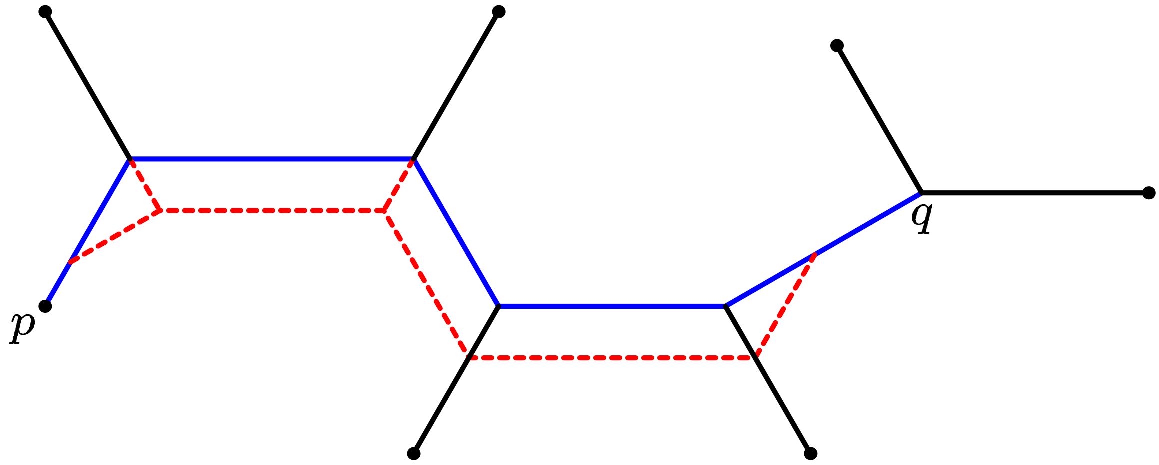
Theorem [Brazil, Thomas & Weng, 2000]
For each set N of n terminals, there exists a λ-SMT T for N such that all Steiner points in T are grid points in $GG_{n-2}(N)$.

Shifts in λ-Geometry

Given two nodes p and q in a tree T, a shift is a perturbation of the internal Steiner points on the path P from p to q in T, such that

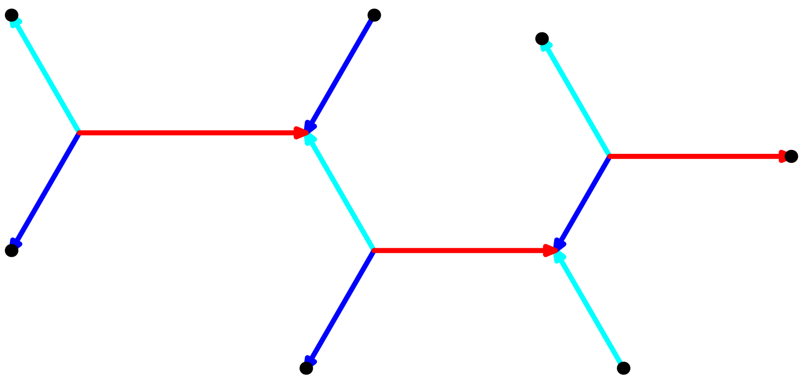
(1) each Steiner point moves along the incident edge which is not on P

(2) each internal edge is keeps its orientation



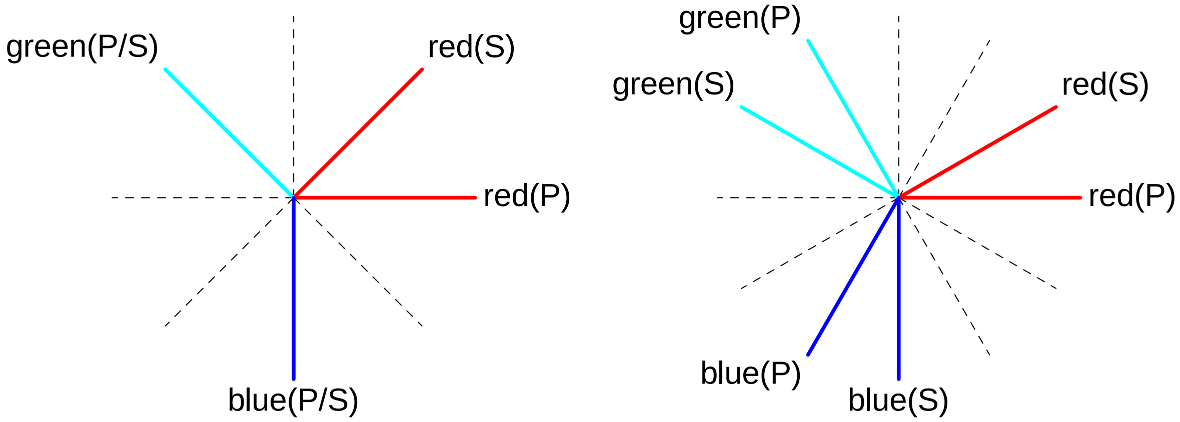
Edge Coloring and Primary/Secondary Labeling

Direct and color edges of a full λ-SMT:



Theorem

The edges in a full λ-SMT can have at most 4 different directions for $\lambda \neq 3m$ and at most 6 different directions for $\lambda = 3m$ (called direction sets).



$\lambda = 4$ $\lambda = 6$

Identify (exclusively) primary and secondary edges.

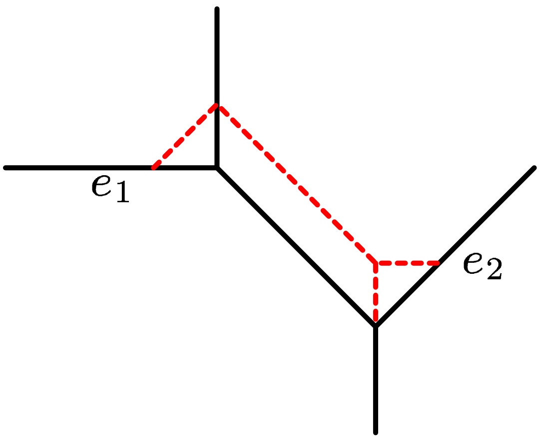
Zero-Shifts

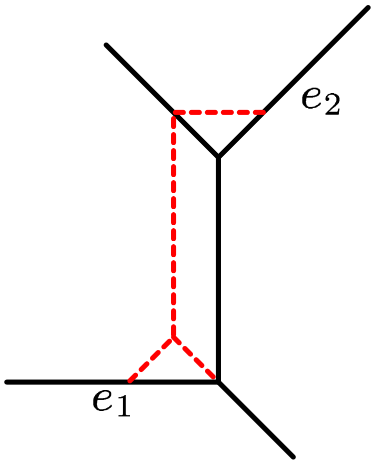
A zero-shift in a full λ-SMT T is a shift that does not increase the length of T.

Theorem
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 zero-shift on the path P between e_1 and e_2.

Proof
By induction on the number of internal edges on the path P.

Basis (one internal edge):





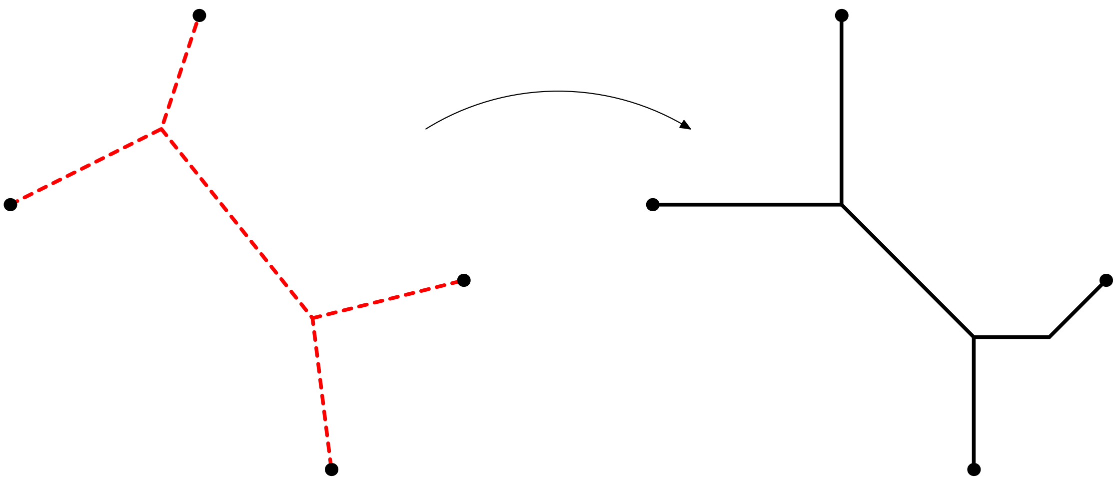
Inductive step: Decompose P into two subpaths and move primary/secondary material to intermediate half edge or exclusively primary/secondary edge.

Steiner Trees for a Given Topology

Topology: Graph structure for the interconnections of terminals and/or Steiner points.

Full Steiner topology: A topology for which all terminals have degree 1 (are leaves) and all Steiner points have degree 3.

> Given a λ-metric and a full Steiner topology $\mathcal{T}$ for a set N of terminals, find a λ-SMT for $\mathcal{T}$ (or locate the Steiner points in $\mathcal{T}$ such that the corresponding tree is a local minimum).



Topology $\mathcal{T}$ $\qquad$ λ-SMT for $\mathcal{T}$

Merging Neighbouring Nodes

Lemma
Assume for a given topology $\mathcal{T}$ and direction set that

- the locations of two neighbours u and v of a Steiner point s are known
- edges (s, u) and (s, v) should be straight and have been labeled primary or secondary

Then, if s exists then its location is unique and can be computed in constant time.

Corollary
If a direction set is given and all edges except one in a topology $\mathcal{T}$ with n terminals have been labeled primary or secondary, then in $O(n)$ time we can either construct a full λ-SMT for $\mathcal{T}$ with the given labeling, or show that no such tree exists.

Proof
Root $\mathcal{T}$ at the unlabeled edge (=bent edge) and iteratively merge nodes bottom-up.

Quadratic Time Algorithm

Arbitrarily assign the numbers 1 to $2n-3$ to the edges of $\mathcal{T}$. Consider some λ-SMT for $\mathcal{T}$. Move primary material to low-numbered edges using zero-shifts.

Now there exists a number k (with $1 \leq k \leq 2n-3$) such that

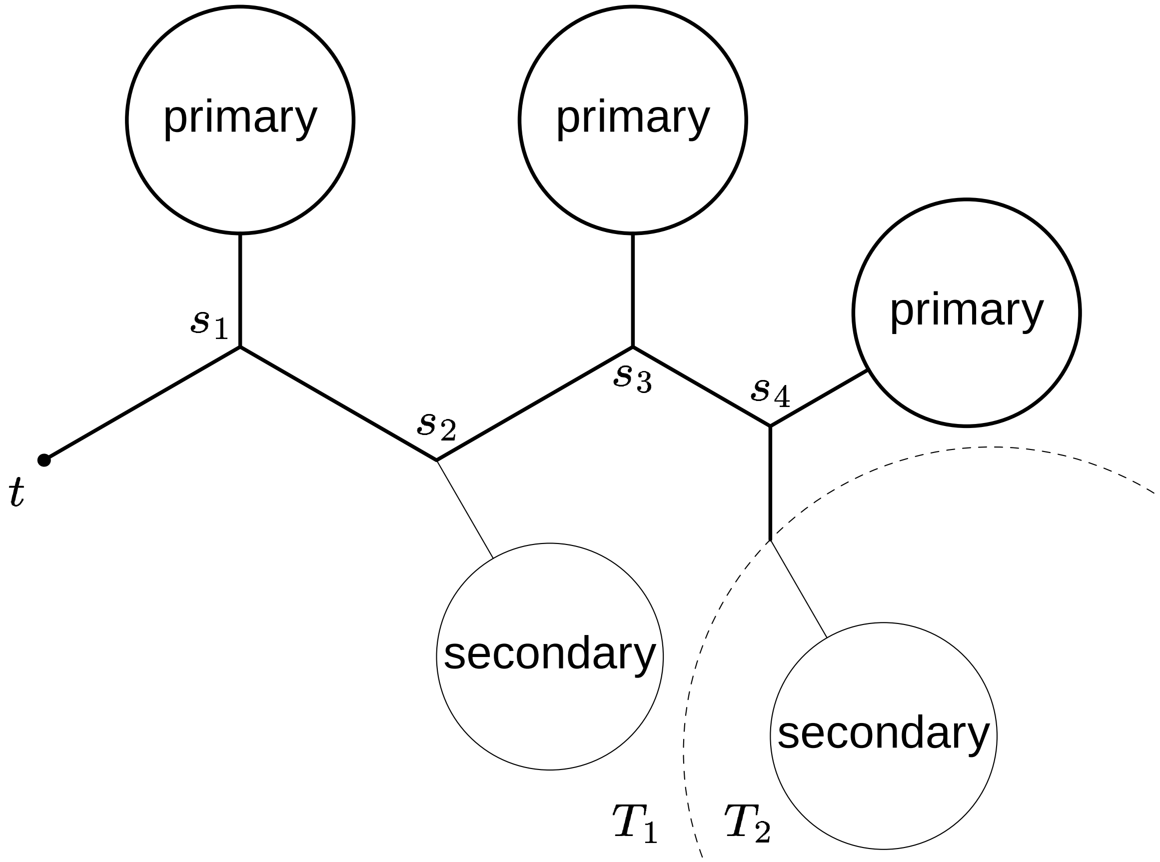
- all edges numbered $1, \ldots, k-1$ are primary
- all edges numbered $k+1, \ldots, 2n-3$ are secondary

Gives a straightforward algorithm for computing a λ-SMT for $\mathcal{T}$: For each direction set and each choice of $k \in \{1, \ldots, 2n-3\}$ use the $O(n)$ algorithm to compute a tree having the corresponding labeling.

Running time: $O(\lambda n^2)$

Canonical Forms

Any numbering of the edges gives a canonical form, e.g. choosing a root t and visiting the edges in a depth-first order gives a powerful canonical form:



Linear Time Algorithm

Assume that a direction set is given.

Initially we label all edges as being secondary, and define the locations of all Steiner points as undefined.

Root the topology $\mathcal{T}$ at some terminal t; let L$[v]$ and R$[v]$ denote the children of node v.

Perform two depth-first order traversals of $\mathcal{T}$ using the following algorithm:

TRAVERSE(u, v)
    if (2. traversal) **then** TRYBENTEDGE(u, v)
    if (v is a Steiner point) **then**
        TRAVERSE(v, L$[v]$)
        TRAVERSE(v, R$[v]$)
        $\Phi[v] =$ MERGERAYS(L$[v]$, R$[v]$)

Operation MERGERAYS(L$[v]$, R$[v]$) attempts to compute the location of the parent based on the location of its children.

Linear Time Algorithm: Bent Edge Construction

TRYBENTEDGE(u, v)
PS[u] = *primary*
if ($u = r$) **then**
    $\Phi[u]$ = ray with source u having the same colour as $\Phi[v]$
else
    $x =$ P[u] and y = third neighbour of u
    $\Phi[u]$ = MERGERAYS(x,y)
if (v is a terminal) **then**
    $\Phi[v]$ = ray with source v having the same colour as $\Phi[u]$
if ($\Phi[u]$ and $\Phi[v]$ have same colour and intersect at c) **then**
    c is corner point in constructed tree
PS[v] = *primary*

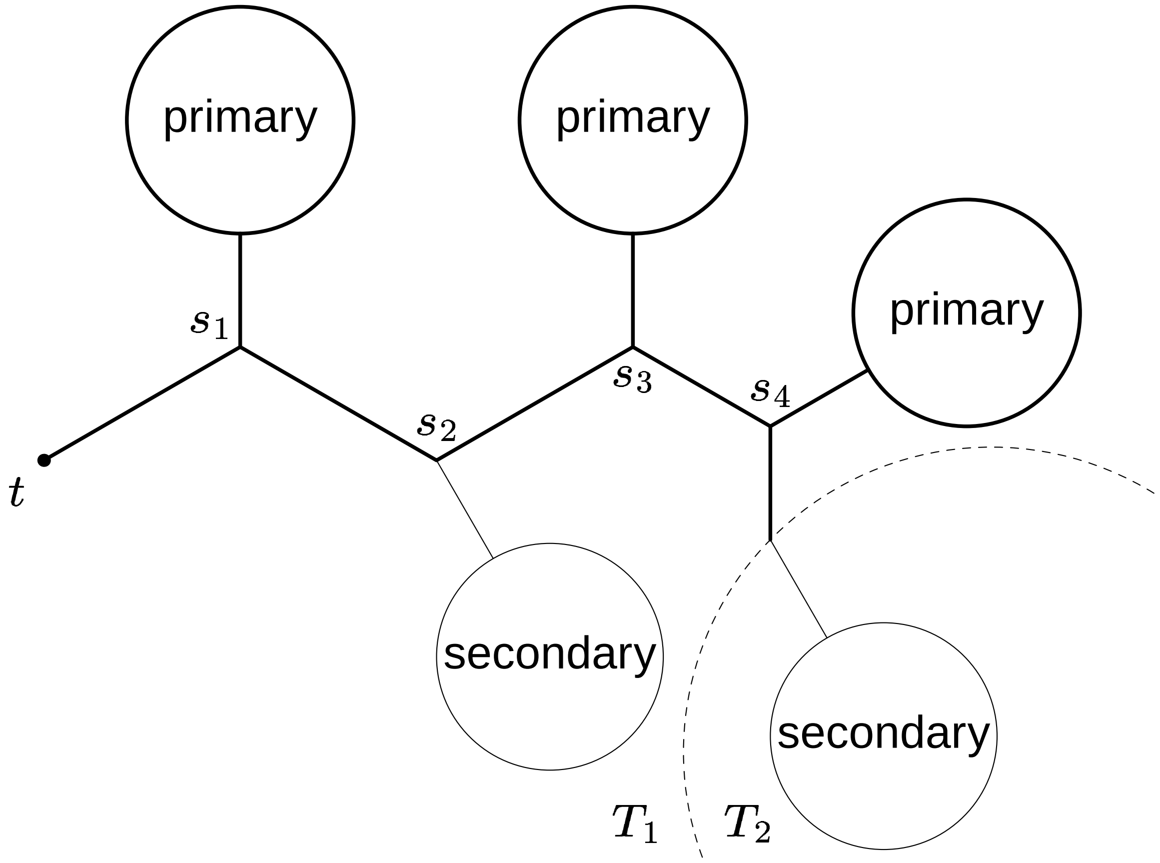
PS[u]: Labeling of edge from node u towards the (coming) bent edge

$\Phi[u]$: Ray located at u with colour (=direction) of the edge from u towards the (coming) bent edge

Linear Time Algorithm: Correctness

Consider path P from the root t to the bent edge:

- all edges on P are primary
- all subtrees connected to P are either primary or secondary
- secondary subtrees (from 1. traversal) are merged with primary subtrees (from 2. traversal)



FST Based Exact Algorithm

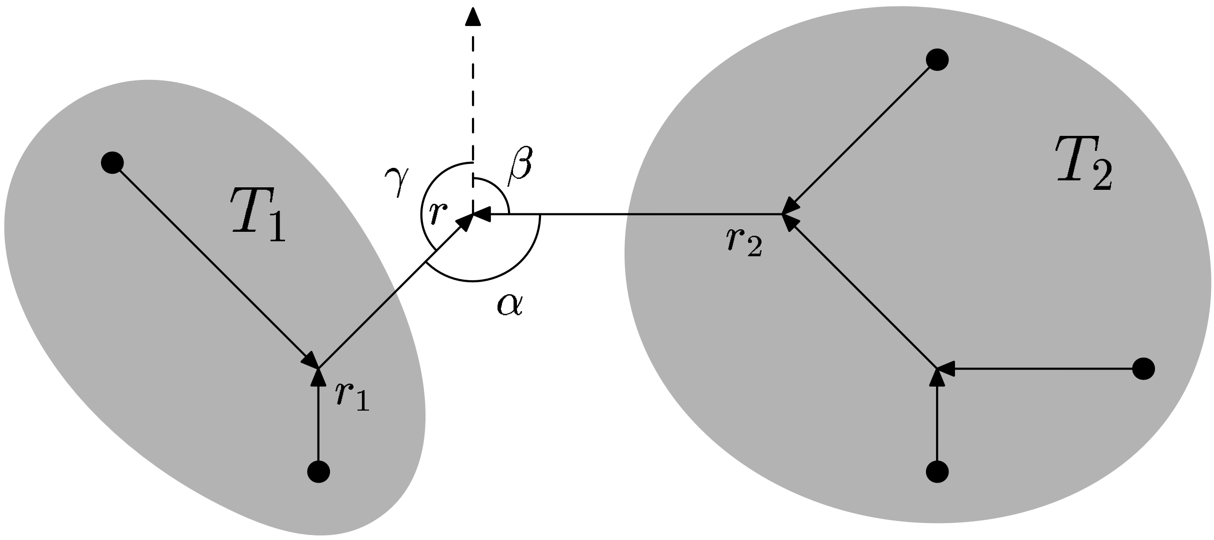
- Phase 1: FST Generation
  - Generate a set $\mathcal{F} = \{F_1, F_2, ..., F_m\}$ of FSTs such that there exists a λ-SMT identified as a subset $\mathcal{F}^* \subseteq \mathcal{F}$.

- Phase 2: FST Concatenation
  - Find a subset $\mathcal{F}^* \subseteq \mathcal{F}$ such that the FSTs in $\mathcal{F}^*$ interconnect all terminals and the total length is as small as possible. Not metric dependent.
  - Equivalent to solving a minimum spanning tree problem in a hypergraph.
  - Can be formulated as an integer program which is solved by linear program relaxation and branch-and-cut. [Warme, 1997].

Generation of FSTs

Root FSTs at the corner point of the bent edge.

Half-FST: Subtree of rooted FST with straight edges only and one “dangling” extension ray.



Observations:

1. Every FST is a combination of two FSTs.

2. Every half FST is a combination of two half FSTs.

Basic approach: Generate half FSTs and prune away non-optimal and non-canonical (half) FSTs.

Pruning Techniques

- Meeting angles
- Bottleneck Steiner distances (BSDs)
- Lune property
- Upper bounds
- Canonical forms

Experimental Results: Pruning

Pruning tests	$n = 20$ #	$n = 20$ %	$n = 50$ #	$n = 50$ %
Number of combinations	349866	100.00	2968266	100.00
Mixed/clean test	247977	70.88	2140068	72.10
Disjoint combinations	173996	49.73	1873572	63.12
Simple angle test	87125	24.90	937147	31.57
Ray intersection	16318	4.66	172694	5.82
BSD test	2681	0.77	6505	0.22
Lune test	1629	0.47	4477	0.15
Upper bounds	622	0.18	1872	0.06
#Generated FSTs	47	0.01	120	0.00
#FSTs in SMT	10	0.00	32	0.00

Experimental Results: Running Times

Results for OR-library instances. VLSI instances showed similar behaviour.

Size	$\lambda = 2$	$\lambda = 3$	$\lambda = 4$	$\lambda = 8$	$\lambda = \infty$
10	0.005	0.028	0.042	0.055	0.034
20	0.021	0.061	0.156	0.269	0.208
30	0.044	0.154	0.364	0.724	0.564
40	0.069	0.343	0.887	1.633	1.056
50	0.157	0.355	1.147	2.246	1.335
60	0.147	0.634	2.016	3.883	1.729
70	0.154	0.891	2.529	5.407	2.447
80	0.237	1.491	3.474	7.941	3.071
90	0.293	1.180	4.319	10.879	3.323
100	0.366	2.195	6.895	19.515	4.785
250	2.841	18.087	80.063	197.227	15.339
500	13.159	102.193	386.071	905.087	40.764
1000	202.483	2386.574	1618.026	3784.987	108.399

All running times in seconds and averages over 15 instances.

930 MHz Pentium III (Linux) machine with 1 Gb of memory.

GeoSteiner was used for $\lambda = 2$ and $\lambda = \infty$.

Concluding Remarks

- Presented a linear-time (optimal) algorithm for constructing a λ-SMT for a given Steiner topology.
- The concept of zero-shifts appears to have a potential for improving global routing algorithms in VLSI design.
- Proposed canonical form very powerful and has resulted in a significant speed-up of an exact algorithm for the problem.
- Exact algorithm part of GeoSteiner package:
  $\longrightarrow$ `www.diku.dk/geosteiner`