Spanning Tree

Problem Formulation:

Given a graph $G = (V, E)$, select a subset $V' \subseteq V$, such that V' has property $\mathcal{P}$.

Minimum Spanning Tree

Problem Formulation:

Given an edge-weighted graph $G = (V, E)$, select a subset of edges $E' \subseteq E$ such that E' induces a tree and the total cost of edges $\Sigma_{e_i \in E'} wt(e_i)$, is minimum over all such trees, where $wt(e_i)$ is the cost or weight of the edge e_i.

– Used in routing applications.

©Sherwani 92

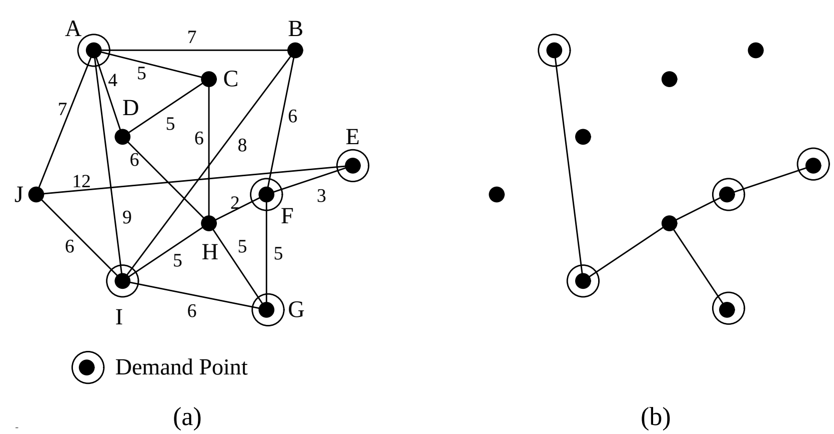
Steiner Trees

1. **Problem formulation:**

   Given an edge weighted graph $G = (V, E)$ and a subset $D \subseteq V$, select a subset $V' \subseteq V$, such that $D \subseteq V'$ and V' induces a tree of minimum cost over all such trees.

   The set D is referred to as the set of *demand points* and the set $V' - D$ is referred to as *Steiner points.*

- Used in the global routing of multi-terminal nets.

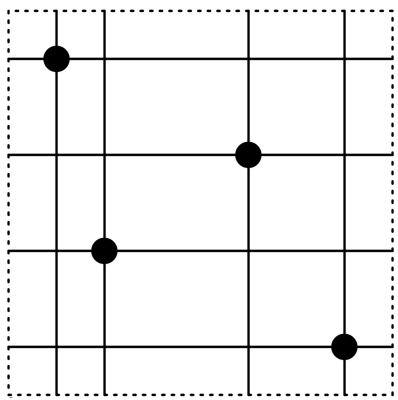


©Sherwani 92

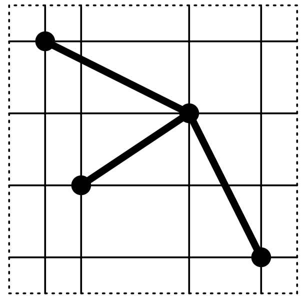
Underlying Grid Graph

The underlying grid graph is defined by the intersections of the horizontal and vertical lines drawn through the demand points.

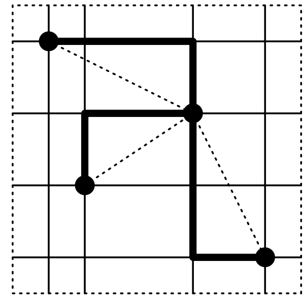
Hanan's Thm (69'):
There exists an optimal RST with all Steiner points (set S) chosen from the intersection points of horizontal and vertical lines drawn from points of D.



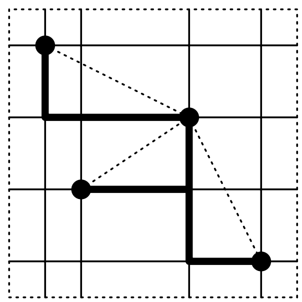
Different Steiner trees constructed from a MST



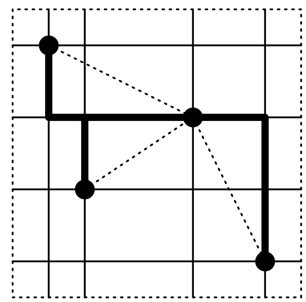
(a)



(b)

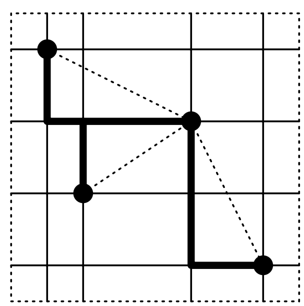


(c)



(d)

Hwang's Thm (76'): The ratio of the cost of a rectilinear MST to that of an optimal RST is no greater than 3/2.



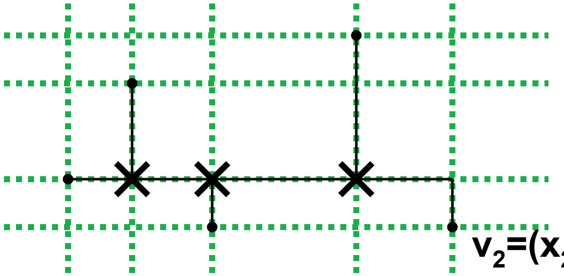
(e)

©Sherwani 92

Rectilinear Steiner Trees

Given a set of points on the plane

Determine a Steiner tree using only horizontal and vertical wires(lines)



Manhattan distance:

$cost(v_1,v_2) = |x_1-x_2|+|y_1-y_2|$

$v1=(x_1,y_1)$, $v2=(x_2,y_2)$

Steiner points

Draw a horizontal and a vertical line through each point.

Need to consider only grid points as steiner points

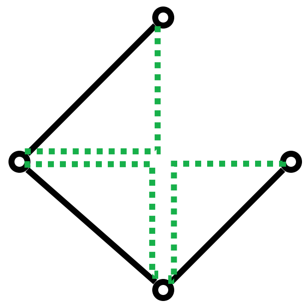
Prim-based algorithm:

Grow a connected subtree by iteratively adding the closest points

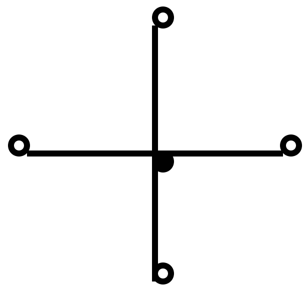
♋ It gives 3/2-approximation, i.e. $cost(T) \leq 3/2 cost(Topt)$

Steiner Tree Heuristics

- Observation: MST approximation can be easily improved

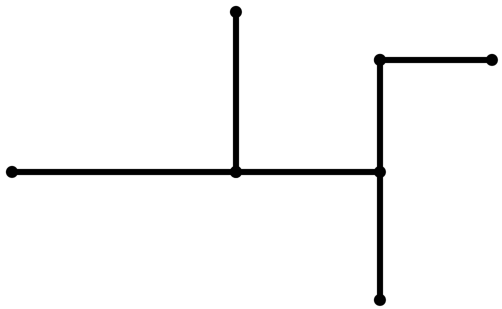


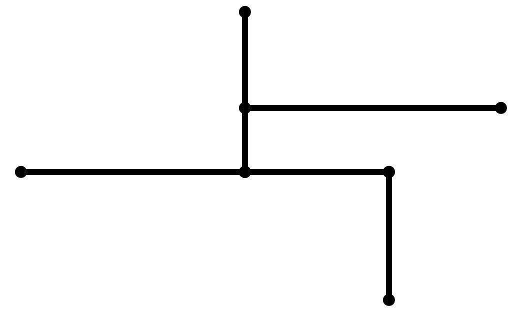
cost(T)=6



cost(T)=4

- Difficulty: where to add Steiner points??

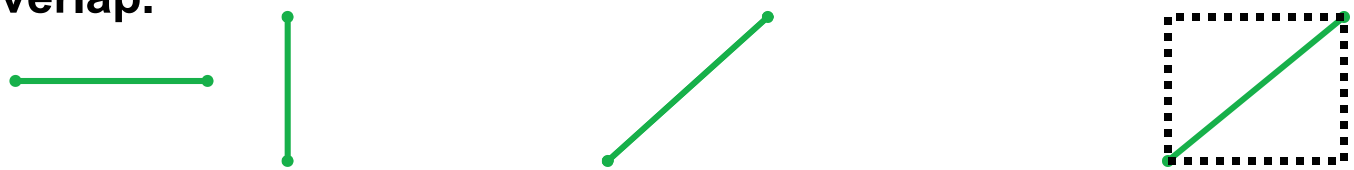




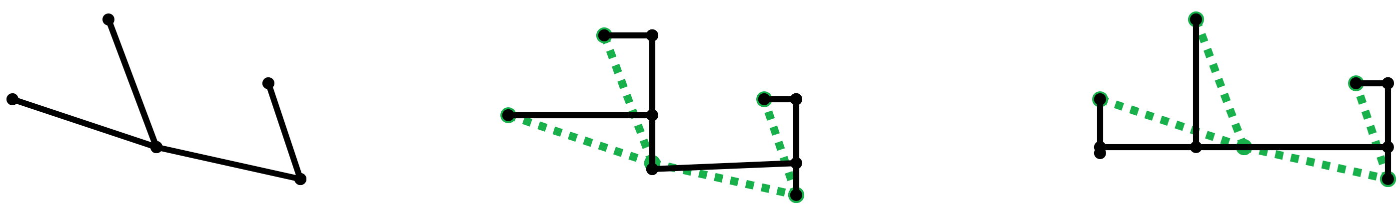
L-Shaped MST approach

Ho, Vijayan and Wong, " A new approach to the rectilinear steiner tree problem", DAC'89, pp161-166

Basic Idea: **Each on-degenerated edge in MST has two possible L-shaped layouts, choose one for each edge in MST to maximize overlap.**



degenerate edges **non-degenerate edge** **two L-shaped layouts**

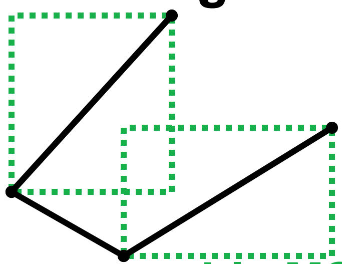


MST **one L-shaped mapping** **another L-shaped mapping**

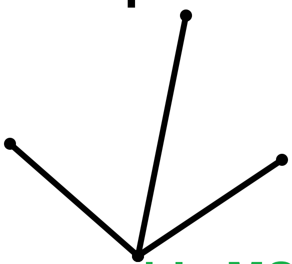
Problem: Compute the best L-shaped mapping

Key Ideas in L-RST Approach

Separable MST: bounding boxes of every two non-adjacent edges don't intersect or overlap



non-separable MST



separable MST

Theorem: Every point set has a separable MST

Theorem: Each node is adjacent to at most 8 edges (6 non-degenerate edges) in a rectilinear MST

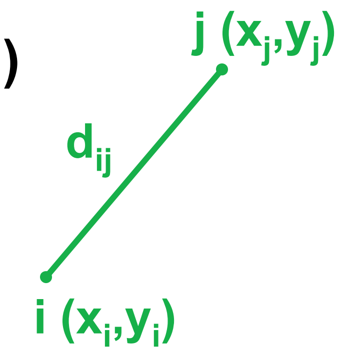
Theorem: We can compute an optimal L-shaped implementation of an MST in $O(2^d \cdot n)$ time. (Dynamic Programming Approach).

Note that $d \leq 6$

Compute a Separable MST

- Weight $\omega(i,j)$ of each edge is a 4-tuple

  $w(i,j) = (d_{ij}, -|y_i - y_j|, -\max(y_i,y_j), -\max(x_i,x_j))$



- Weights are compared under lexicographic ordering

- Use Prim's algorithm to compute a MST based on the weight function
  - we obtain a rectilinear MST since the 1st component of $w(i,j)$ is d_{ij}
  - This MST is separable since the next three components in $w(i,j)$ help break ties

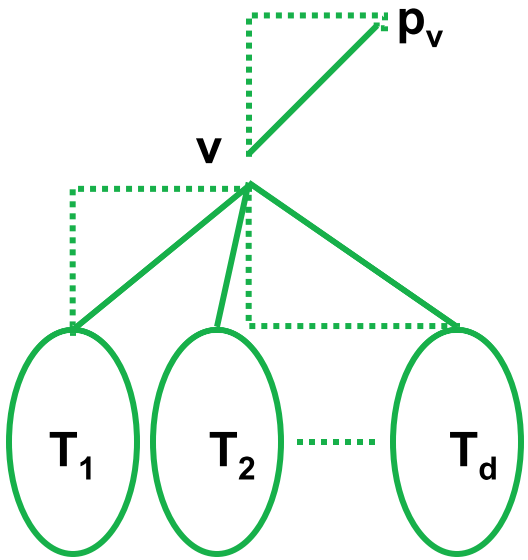
- Run time $O(n^2)$

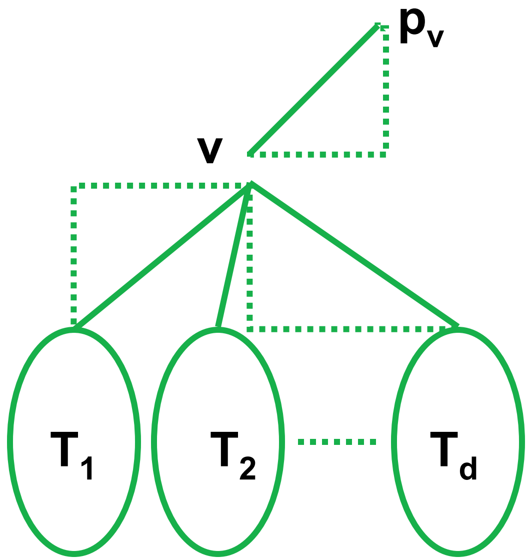
Compute an Optimal L-shaped Mapping

- Choose an arbitrary node as root
- Using dynamic programming
- Compute at each node v the following
  - $\Phi_l(v_i)$: min cost of T_i with (v,v_i) using lower L-shape
  - $\Phi_u(v_i)$: min cost of T_i with (v, v_i) using upper L-shape

Compute an Optimal L-shaped Mapping
(Cont'd)

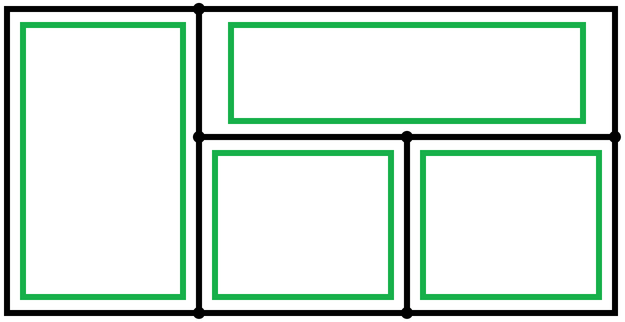
$\Phi_l(v)$ (or $\Phi_u(v)$) can be computed by examining 2^d combinations of $\Phi_l(v_i)$ and $\Phi_u(v_i)$ ($1 \le i \le d$) with (p_v,v) taking upper (or lower) L





Applying Spanning/Steiner Tree Algorithms

- General cell design: channel intersection graphs



- Standard cell/Gate array/Sea-of-gate design rectilinear steiner/spanning trees or grid graphs