Applications: global routing, detailed routing

- Slow
- Requires large memory for dense routes

Weaknesses
- Guaranteed minimum path
- Guaranteed to find connections between 2 terminals if exists

Strengths
- Discussion mainly on single-layer routing


Maze Router: Lei Algorithm

Classification of Routing

Routing Constraints

100% routing completion + area minimization, under a set of constraints
Step 1: Propagate wave from the source to the closest target.

Step 2: Mark all cells on the path as visited.

Step 3: Propagate wave from ALL visited cells to the other cells.

Step 4: Continue until all cells are reached.

Step 5: Apply heuristics to further reduce the tree cost.

- Need to ensure the rectangle and row of the source falls.
- Corner cells.
- Double fan-out: Propagate waves from both the source and the target cells.
- Starting point selection: Choose the point farthest from the center.

Reducing Running Time

Reducing Memory Requirement
Hedrick's Algorithm (cont.)

1. For any cell blocked, label its adjacent unblocked cells away from T + 1. Label
2. Find the shortest path between S and T.

Hedrick’s Algorithm

A Routing Example on a Weighted Grid
Hightower's Algorithm

Mikami-Tabuchi Algorithm

Features of Line-Search Algorithms

Soukup's Algorithm
route net b before net a

Net Ordering

In the example, we should route net b before net a.
Net ordering already affects routing solutions.

Comparison of Algorithms

Operations cannot guarantee shortest paths.
Sokolov, Mikhail, and Highetowar all adopt some sort of heuristics.

MULTILAYER ROUTING
Graph Model: Channel Intersection Graph

Extended channel intersection graph: terminals are also roots.

- Edges with weights represent channel capacity.
- Channel intersections are represented as vertices.
- Channels are represented as edges.

Routing Ordering: a (0) → b (1) → d (2) → c (6)

Net Ordering (cont.)
Each channel has a capacity of 2 tracks.

- Minimize the maximum path length.
- Minimize the maximum wire length.
- For high performance, guarantee 100% routability.

Global Routing

Global Routing in Gate Array

Global Routing in Standard Cell

Global Routing in Different Design Styles

Global Routing Problem
Global Routing by Integer Programming

**Key:**
- Routing: Feasible for all problem sizes
- Time: Total time can be reduced
- Capacity: The capacity of each channel may be exceeded
- Constraint: For each net, only one path will be selected
- Note: There are no possible two nets at the same time (independent)

### Classification of Global Routing Algorithms

- **Concurrent Approach:** All nets are considered at the same time (complexity)
- **Sequential Approach:** Routing properties to nets’ orders one by one at a time based

### Global Routing in FPGAs

Each channel has a capacity of 2 tracks.
Hierarchical Global Routing

Theoretical results for the MRST problem:

The minimal Steiner tree for any point set $P$, denoted by $\text{MST}(P)$, is the graph $G = (V, E)$ that spans all points of $P$ and has the smallest possible sum of edge lengths.

Minimum Steiner tree problem: Given $n$ points, construct the minimum Steiner tree connecting these points.

Seminal points: Consider the set $S$ of original points and the set $\text{MST}(S)$. The problem is to find a minimum-length tree connecting the points in $S$ that contains the points in $\text{MST}(S)$.

The routing-free problem: Given a set of points, determine if there exists a tour that connects all points without using a network.

An Integer Programming Example

Let $\mathbf{A}$ be the adjacency matrix of the network, and $\mathbf{b}$ be the vector of total demands at each node. The problem is to find a feasible flow $\mathbf{x}$ that satisfies the capacity constraints and minimizes the total cost:

$$\min \sum_{ij} c_{ij} x_{ij}$$

subject to

$$\sum_{j \neq i} x_{ij} - \sum_{j \neq i} x_{ji} = b_i$$

for all nodes $i$.

Let $x_{ij}$ be the flow from node $i$ to node $j$, and $c_{ij}$ the cost of sending one unit of flow from $i$ to $j$. The problem is to find $\mathbf{x}$ that minimizes the total cost while satisfying the flow conservation constraint.
Two possible L-shaped layouts per edge

Two L-shaped MST of the given MST

About 6% smaller than \( \text{Cost}(\text{MST}) \).

3. Maximizing overlaps by dynamic programming.
2. Each edge is straight or L-shaped.
1. Construct an MST from an MST.

Problem: "IEEE TCDP, 1990." New algorithms for the rectilinear Steiner Problem with the MST Problem

A Simple Performance Bound

1. Construct \( T \) based on the traversal.
2. Perform a depth-first traversal on the reduced tree \( T \).

1. Select any point in \( T \) as a root.
2. Select a spanning tree \( T \) on a point set \( S \).

Given any MST \( T \) on point set \( S \) with Steiner point set \( S' \), can-

Easy to show the bound.

Algorithm: Rectified Steiner (RST)

MST (minimum spanning tree) \( \leftrightarrow \) minimum-cost SPT (shortest

Not clear how to find a spanning tree with a good trade-off.

Question: How to find a spanning tree with a good trade-off?

Cost\( (T) = 1 + 2 + 5 + 1 + 2 = 11 \)

Radius\( (T) = \max \{ 1 + 2, 1 + 5 + 1, 1 + 5 + 2 \} = 8 \)

The Iterated L-Steiner Heuristic for the MST Problem
\[ \text{Algorithm: Bounded-Radius\-Bounded-Cost\-Spanning\-Tree}(G) \]

1. **Begin**

2. **Check** if \( G \) is a Minimum Spanning Tree (MST) of \( G \).

3. If yes, \( \text{Cost}(G) = \infty \).

4. Compute MST of \( G \).

5. \( \text{Cost}(G) \) may be \( \leq \text{Cost}(\text{MST}) \).

6. **End**
Detailed routing

switchbox routing

Channel & Switchbox Routing