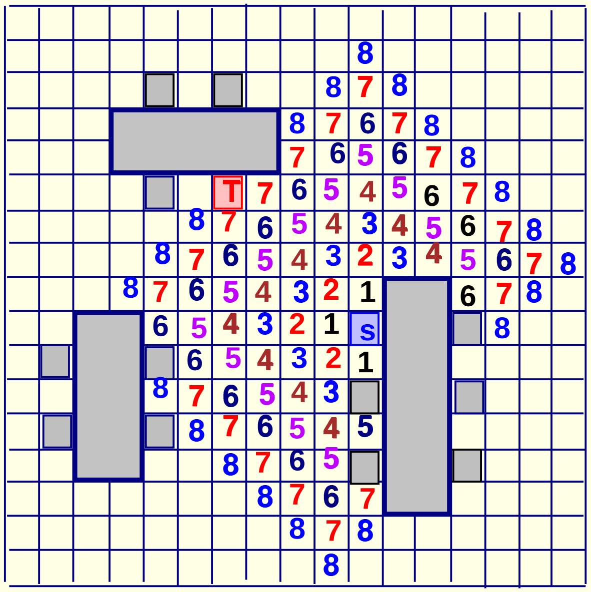
Maze Router: Lee Algorithm

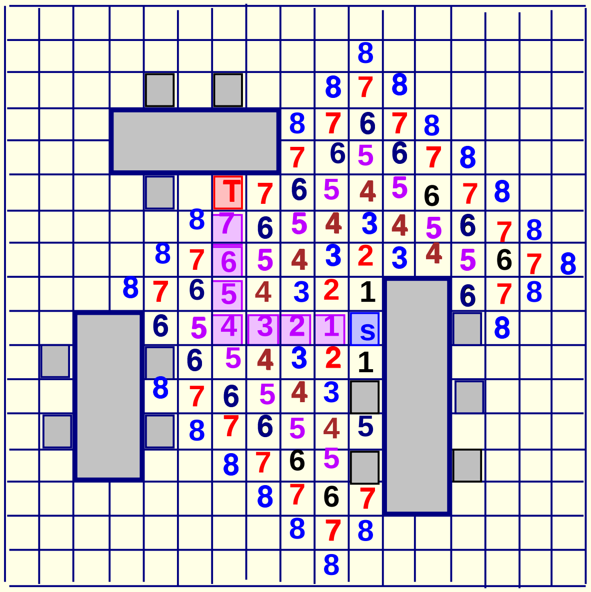
- Lee, "An algorithm for path connection and its application," IRE Trans. Electronic Computer, EC-10, 1961.
- Discussion mainly on single-layer routing
- **Strengths**
  - Guarantee to find connection between 2 terminals if it exists.
  - Guarantee minimum path.
- **Weaknesses**
  - Requires large memory for dense layout
  - Slow
- Applications: global routing, detailed routing

Lee Algorithm

- Find a path from S to T by "wave propagation".



Filing

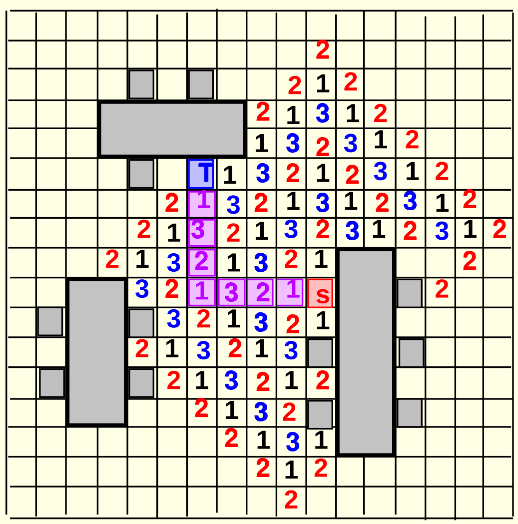


Retrace

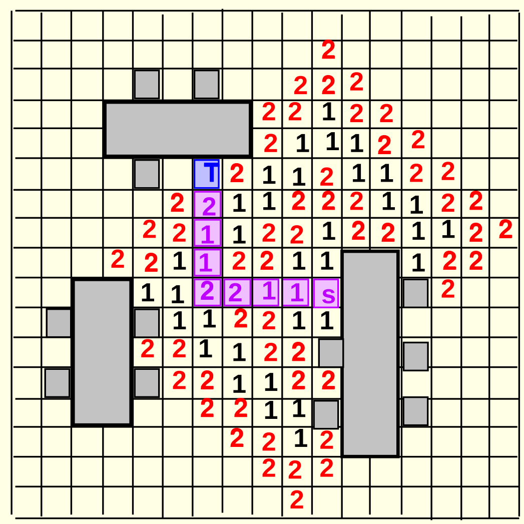
- Time & space complexity for an $M \times N$ grid: $O(MN)$ (**huge!**)

Reducing Memory Requirement

- Akers's Observations (1967)
  - Adjacent labels for k are either $k-1$ or $k+1$.
  - Want a labeling scheme such that each label has its preceding label different from its succeeding label.
- Way 1: coding sequence $1, 2, 3, 1, 2, 3, \ldots$; states: 1, 2, 3, empty, blocked (3 bits required)
- Way 2: coding sequence $1, 1, 2, 2, 1, 1, 2, 2, \ldots$; states: 1, 2, empty, blocked (need only 2 bits)



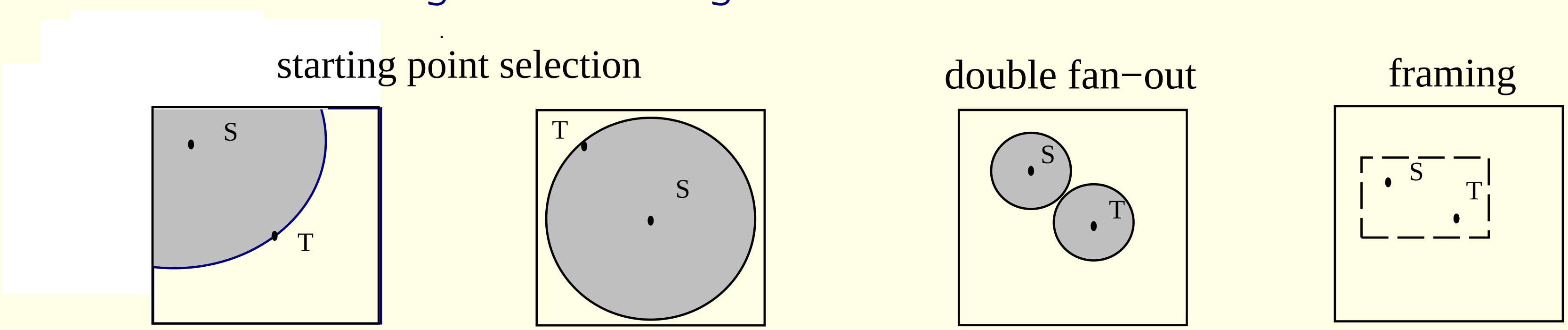
Sequence: 1, 2, 3, 1, 2, 3, ...



Sequence: 1, 1, 2, 2, 1, 1, 2, 2, ...

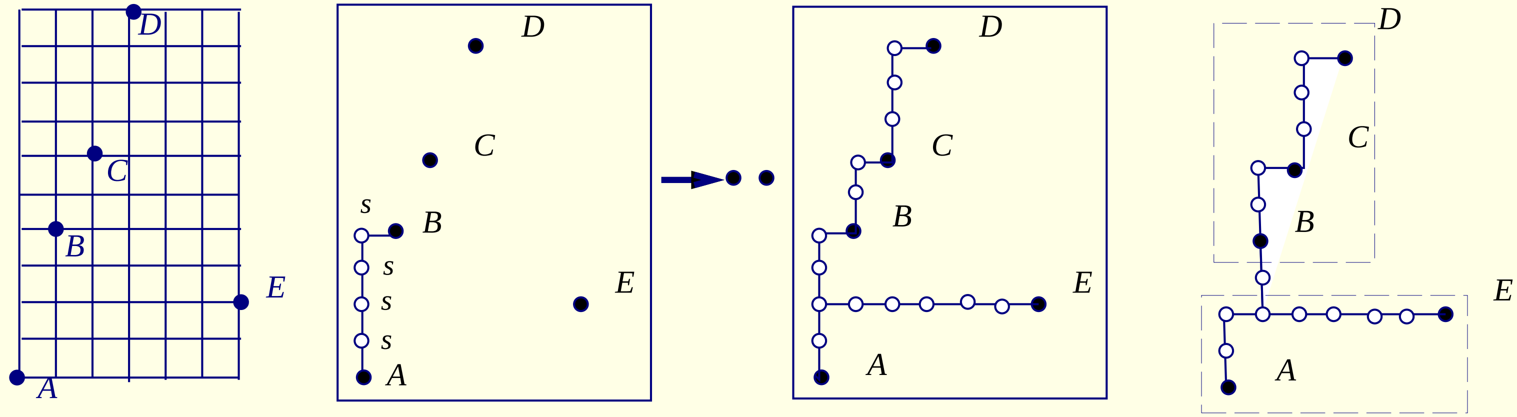
Reducing Running Time

- Starting point selection: Choose the point farthest from the center of the grid as the starting point.
- Double fan-out: Propagate waves from both the source and the target cells.
- Framing: Search inside a rectangle area 10–20% larger than the bounding box containing the source and target.
  - Need to enlarge the rectangle and redo if the search fails.



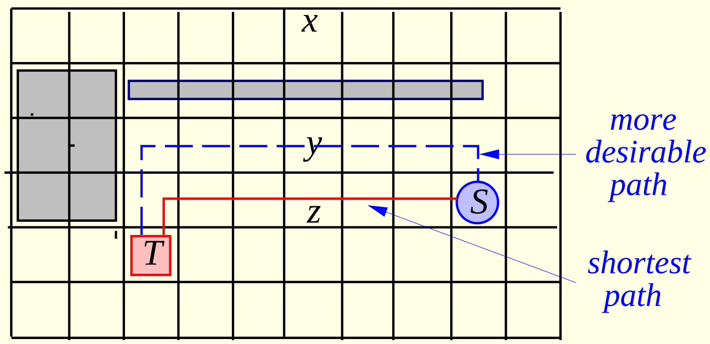
Connecting Multi-Terminal Nets

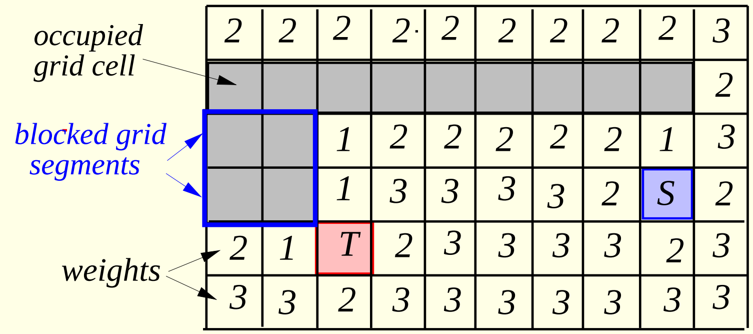
- Step 1: Propagate wave from the source s to the closet target.
- Step 2: Mark ALL cells on the path as s.
- Step 3: Propagate wave from ALL s cells to the other cells.
- Step 4: Continue until all cells are reached.
- Step 5: Apply heuristics to further reduce the tree cost.



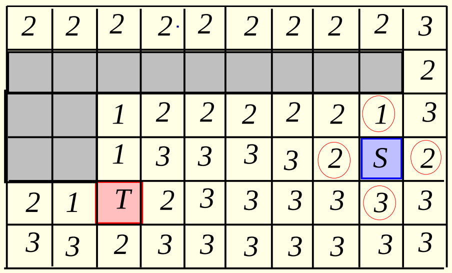
Routing on a Weighted Grid

- Motivation: finding more desirable paths
- $weight(grid\ cell) = \#\ of\ unblocked\ grid\ cell\ segments - 1$

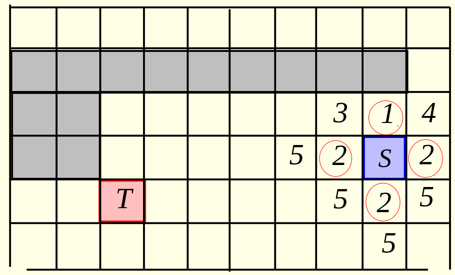




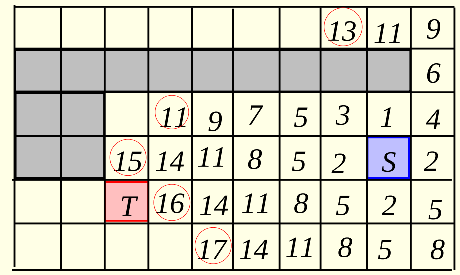
A Routing Example on a Weighted Grid

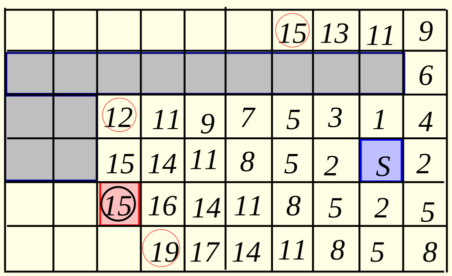


initialize cell weights

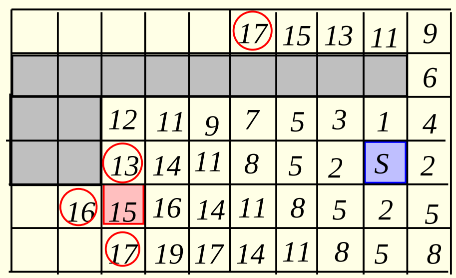


wave propagation

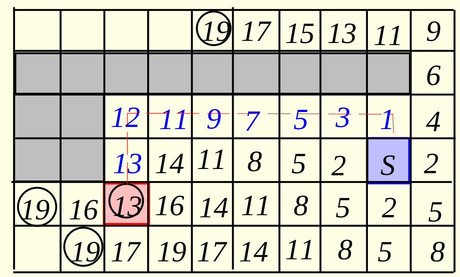




first wave reaches the target



finding other paths



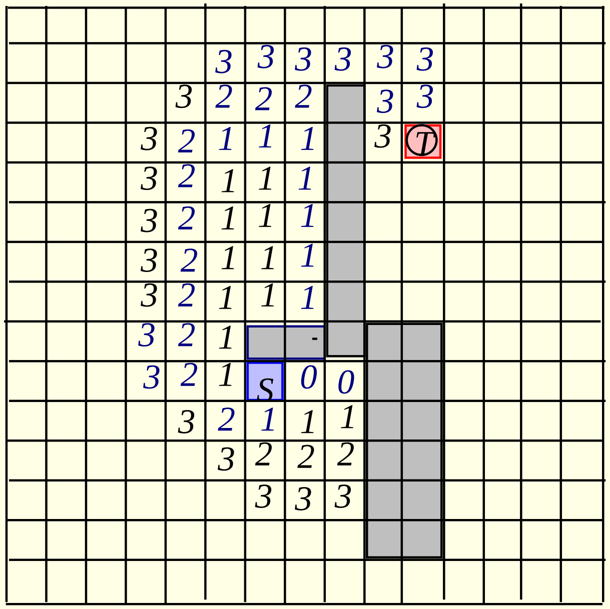
min−cost path found

Hadlock's Algorithm

- Hadlock, "A shortest path algorithm for grid graphs," Networks, 1977.
- Uses detour number (instead of labeling wavefront in Lee's router)
  - Detour number, $d(P)$: # of grid cells directed **away from** its target on path P.
  - $MD(S,T)$: the Manhattan distance between S and T.
  - Path length of P, $l(P)$: $l(P) = MD(S,T) + 2d(P)$.
  - $MD(S,T)$ fixed! $\Rightarrow$ Minimize $d(P)$ to find the shortest path.
  - For any cell labeled i, label its adjacent unblocked cells **away from** T $i+1$; label i otherwise.
- Time and space complexities: $O(MN)$, but substantially reduces the # of searched cells.
- Finds the shortest path between S and T.

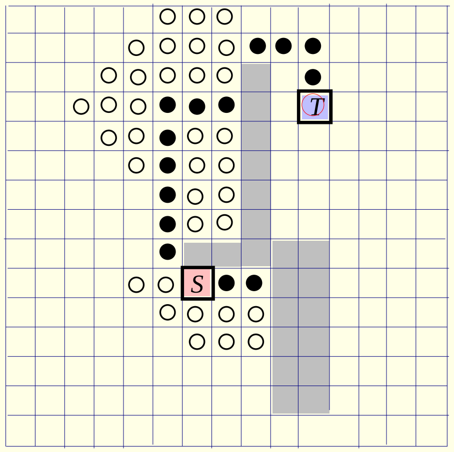
Hadlock's Algorithm (cont'd)

- $d(P)$: # of grid cells directed **away from** its target on path P.
- $MD(S,T)$: the Manhattan distance between S and T.
- Path length of P, $l(P)$: $l(P) = MD(S,T) + 2d(P)$.
- $MD(S,T)$ fixed! $\Rightarrow$ Minimize $d(P)$ to find the shortest path.
- For any cell labeled i, label its adjacent unblocked cells **away from** T $i+1$; label i otherwise.

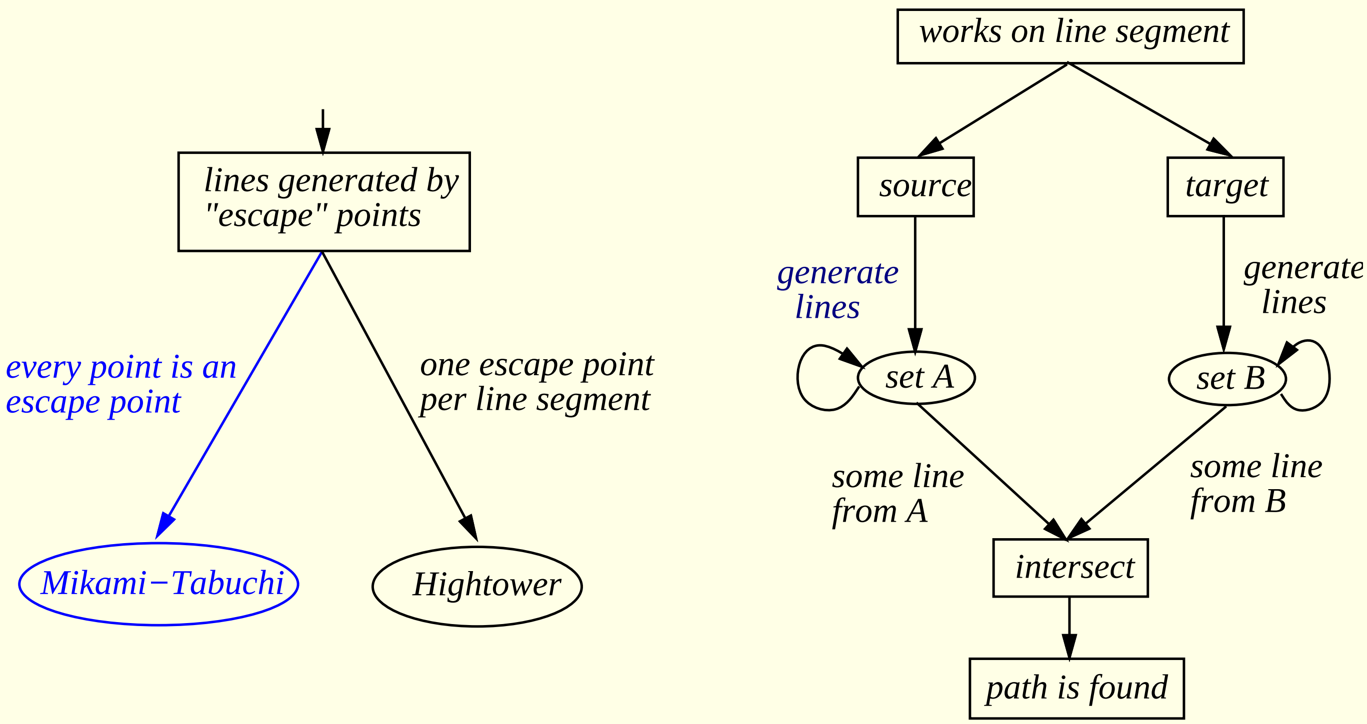


Soukup's Algorithm

- Soukup, "Fast maze router," DAC-78.
- Combined breadth-first and depth-first search.
  - Depth-first (**line**) search is first directed toward target T until an obstacle or T is reached.
  - Breadth-first (Lee-type) search is used to "bubble" around an obstacle if an obstacle is reached.
- Time and space complexities: $O(MN)$, but 10–50 times faster than Lee's algorithm.
- Find **a** path between S and T, but may not be the shortest!



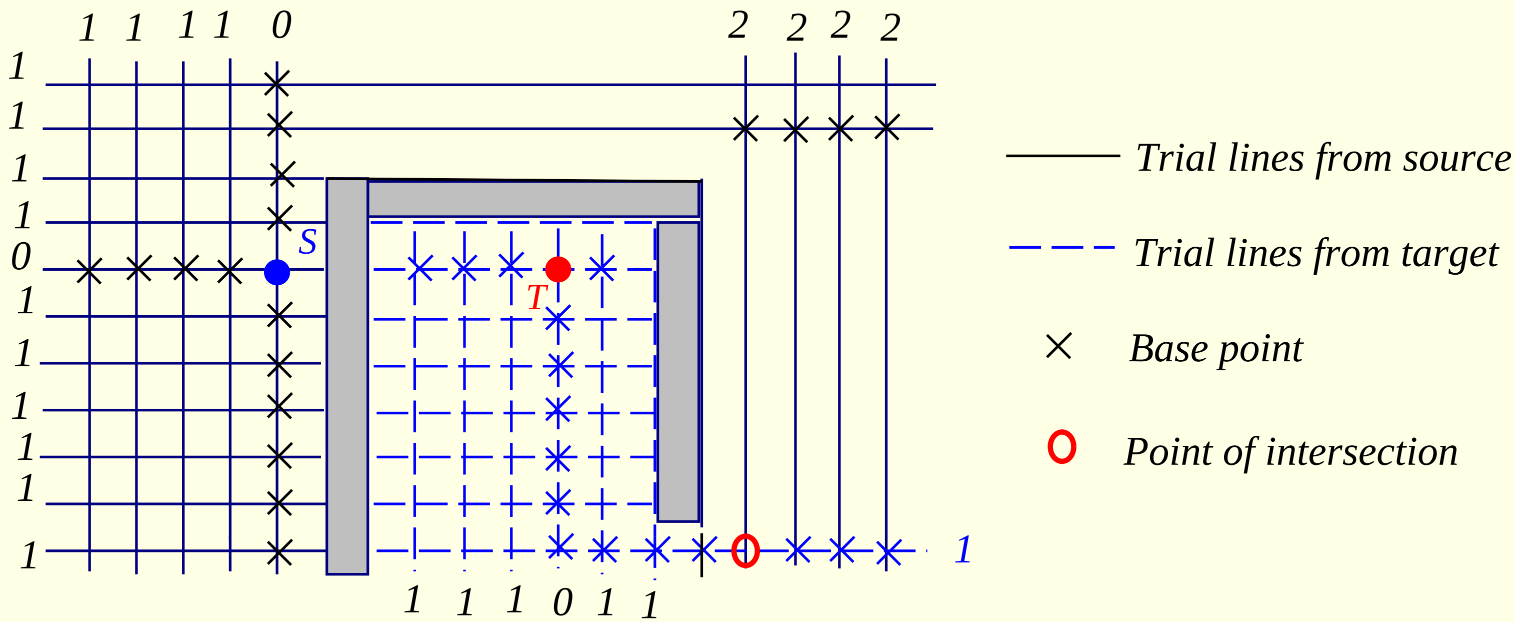
Features of Line-Search Algorithms



- Time and space complexities: $O(L)$, where L is the # of line segments generated.

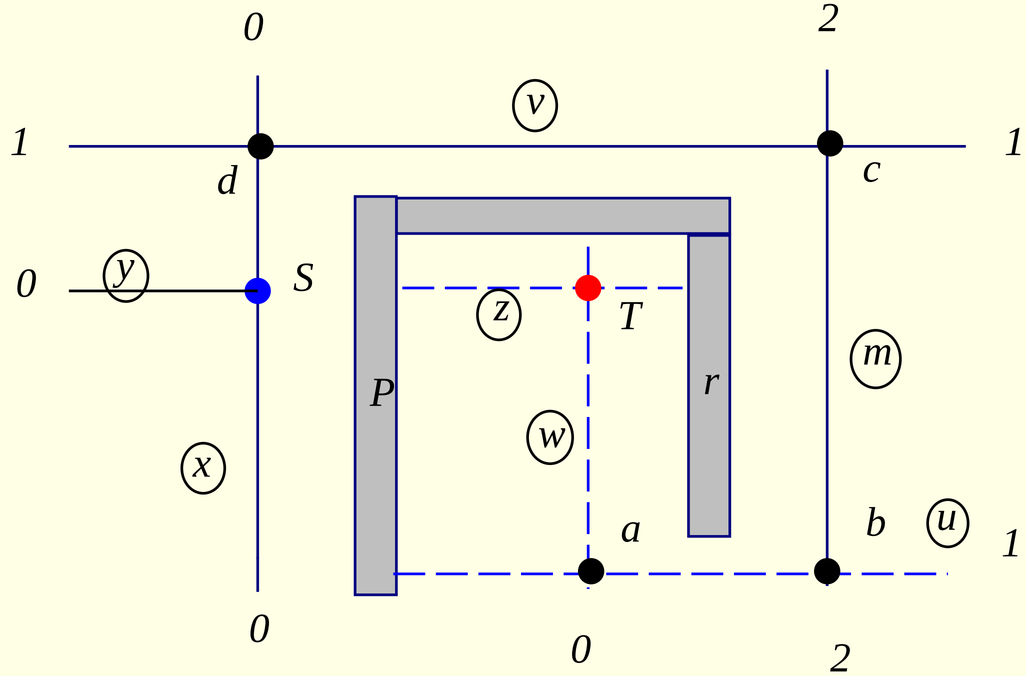
Mikami-Tabuchi's Algorithm

- Mikami & Tabuchi, "A computer program for optimal routing of printed circuit connectors," IFIP, H47, 1968.
- Every grid point is an escape point.



Hightower's Algorithm

- Hightower, "A solution to line-routing problem on the continuous plane," DAC-69.
- A single escape point on each line segment.
- If a line parallels to the blocked cells, the escape point is placed just past the endpoint of the segment.



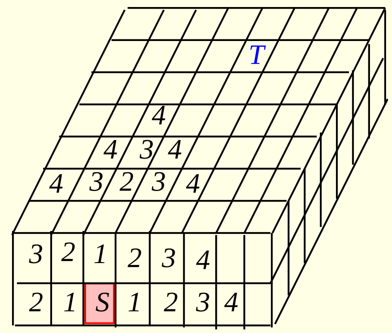
Comparison of Algorithms

	Maze routing			Line search	
	Lee	Soukup	Hadlock	Mikami	Hightower
Time	$O(MN)$	$O(MN)$	$O(MN)$	$O(L)$	$O(L)$
Space	$O(MN)$	$O(MN)$	$O(MN)$	$O(L)$	$O(L)$
Finds path if one exists?	yes	yes	yes	yes	no
Is the path shortest?	yes	no	yes	no	no
Works on grids or lines?	grid	grid	grid	line	line

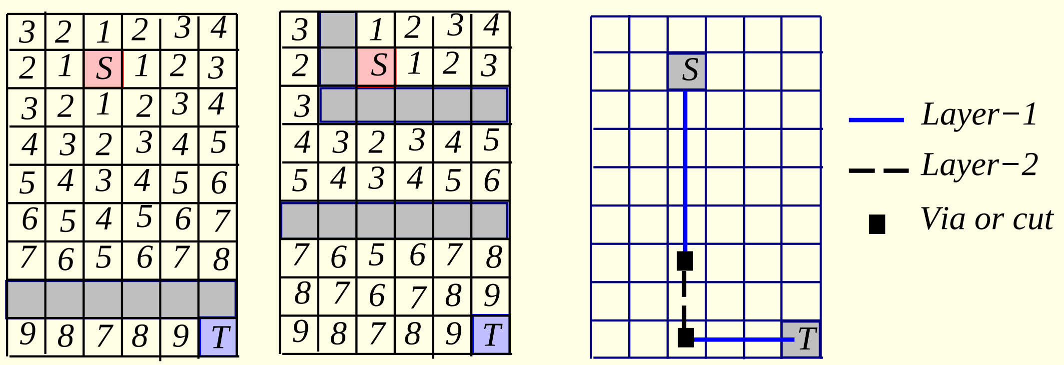
- Soukup, Mikami, and Hightower all adopt some sort of line-search operations $\Rightarrow$ cannot guarantee shortest paths.

Multi-layer Routing

- 3-D grid:



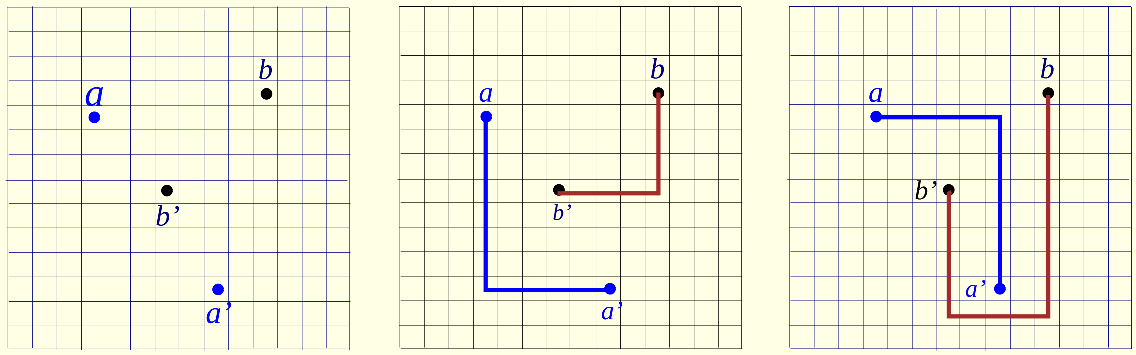
- Two planner arrays:
  - Neglect the weight for inter-layer connection through via.
  - Pins are accessible from both layers.



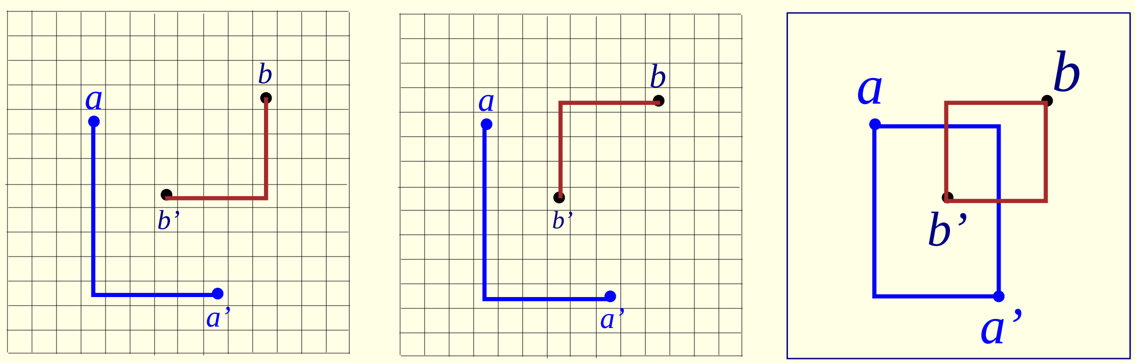
1st layer *2nd layer* *a path*

Net Ordering

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



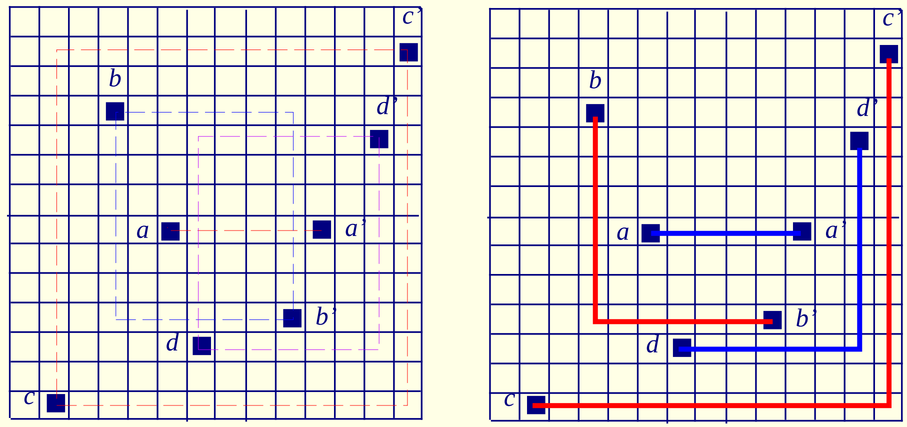
route net a before net b



route net b before net a

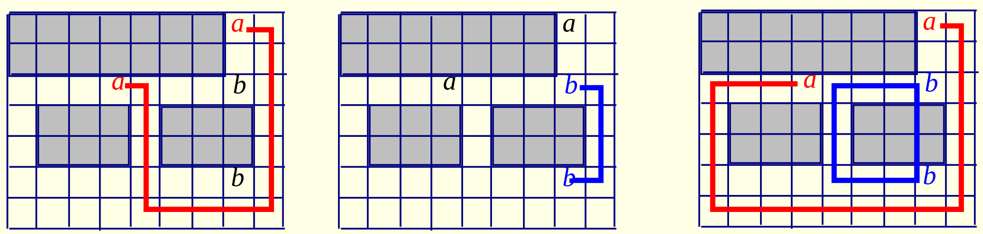
Net Ordering (cont'd)

- Order the nets in the ascending order of the # of pins within their bounding boxes.
- Order the nets in the ascending (or descending??) order of their lengths.
- Order the nets based on their timing criticality.



routing ordering: a (0) −> b (1) −> d (2) −> c (6)

- A mutually intervening case:



a prevents routing of b *b prevents routing of a* *a feasible routing*

Rip-Up and Re-routing

- Rip-up and re-routing is required if a global or detailed router fails in routing all nets.
- Approaches: the manual approach? the automatic procedure?
- Two steps in rip-up and re-routing
  1. Identify bottleneck regions, rip off some already routed nets.
  2. Route the blocked connections, and re-route the ripped-up connections.
- Repeat the above steps until all connections are routed or a time limit is exceeded.