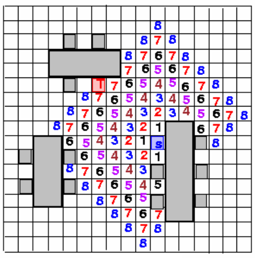
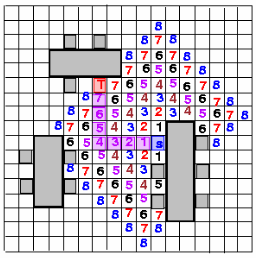
Unit 6: Maze (Area) and Global Routing

- Course contents
  - Routing basics
  - Maze (area) routing
  - Global routing
- Readings
  - Chapters 9.1, 9.2, 9.5

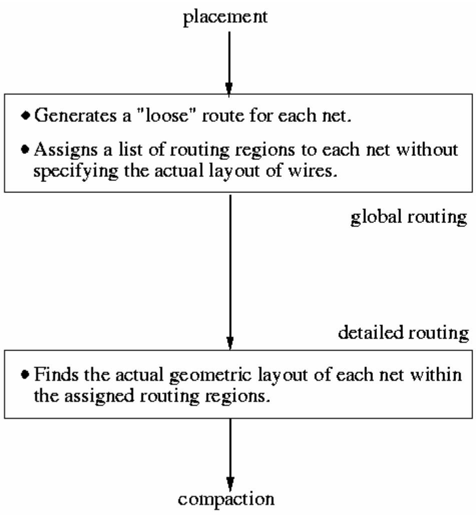


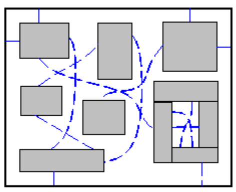
Filling



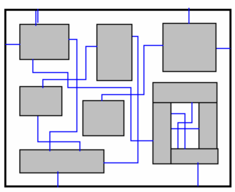
Retrace

Routing





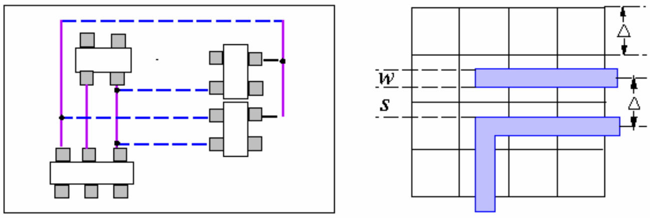
Global routing



Detailed routing

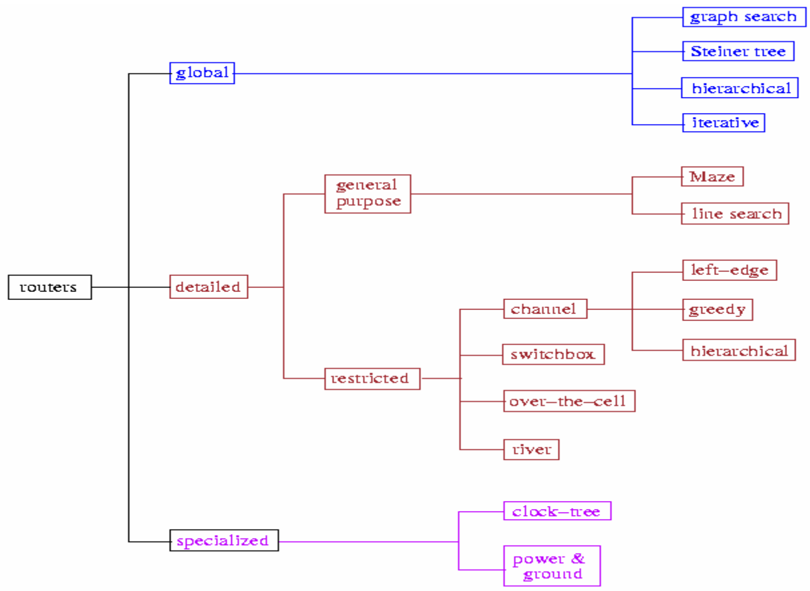
Routing Constraints

- 100% routing completion + area minimization, under a set of constraints:
  - Placement constraint: usually based on fixed placement
  - Number of routing layers
  - Geometrical constraints: must satisfy design rules
  - Timing constraints (performance-driven routing): must satisfy delay constraints
  - Crosstalk?



Two–layer routing *Geometrical constraint*

Classification of Routing

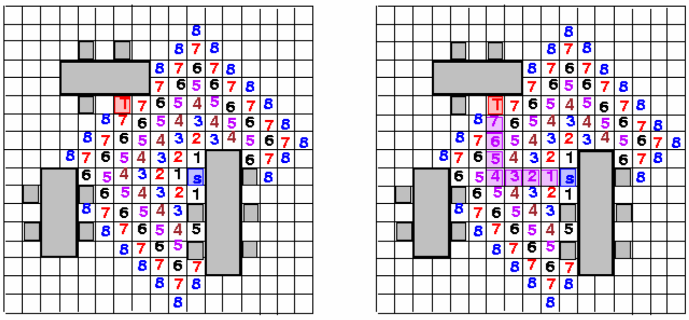


Maze Router: Lee Algorithm

- 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".

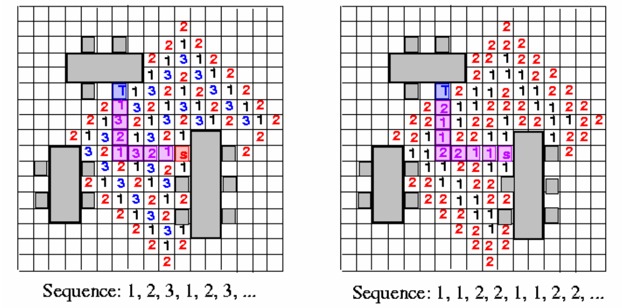


Filling Retrace

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

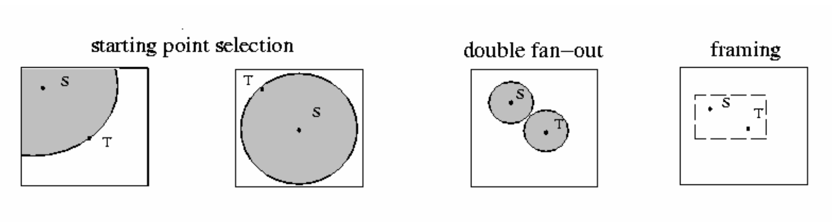
Reducing Memory Requirement

- 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, …; states: 1, 2, 3, *empty*, *blocked* (3 bits required)
- Way 2: coding sequence 1, 1, 2, 2, 1, 1, 2, 2, …; states: 1, 2, *empty*, *blocked* (need only 2 bits)



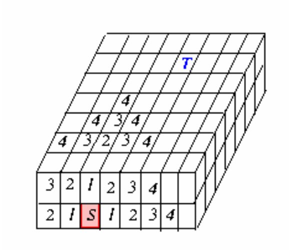
Reducing Running Time

- 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.

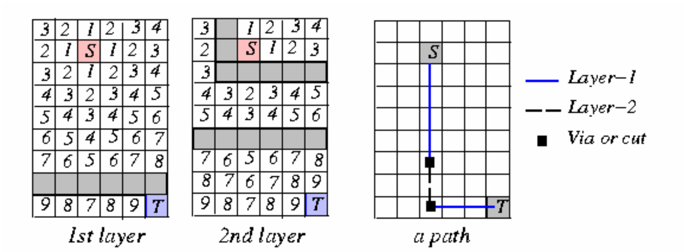


Multi-Layer Routing

- 3-D grid:

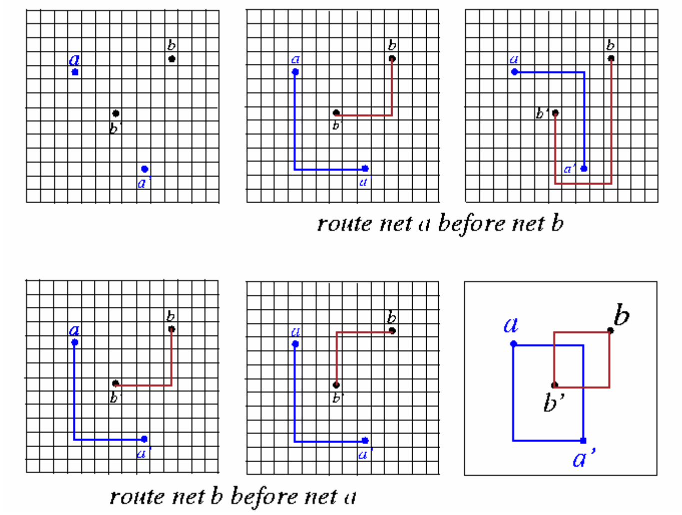


- Two planar arrays:
  - Neglect the weight for inter-layer connection through via.
  - Pins are accessible from both layers.



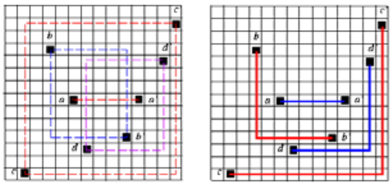
Net Ordering

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



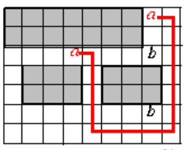
Net Ordering (cont'd)

- 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.

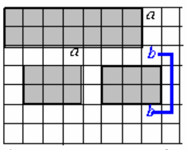


routing ordering: a (0) –> b (1) –> d (2) –> c (6)

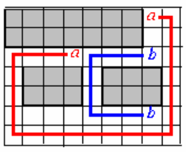
- A mutually intervening case:



a prevents routing of b



b prevents routing of a



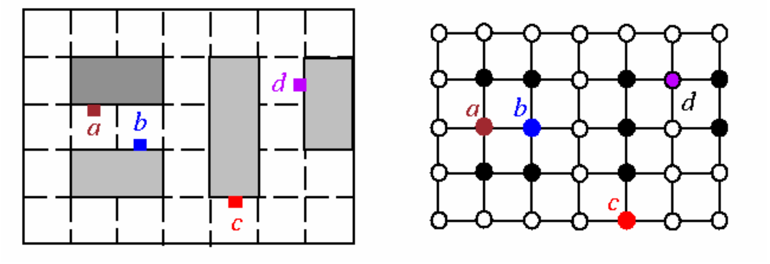
a feasible routing

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.

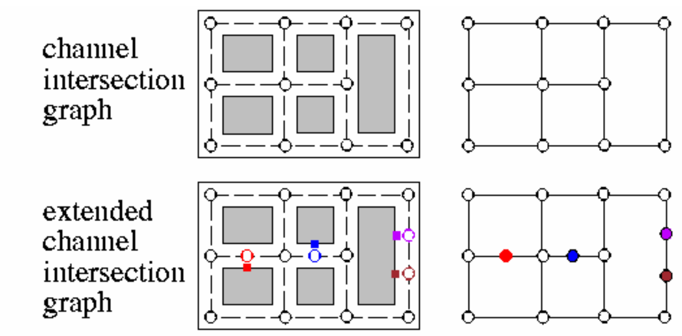
Graph Models for Global Routing: Grid Graph

- Each cell is represented by a vertex.
- Two vertices are joined by an edge if the corresponding cells are adjacent to each other.
- The occupied cells are represented as filled circles, whereas the others are as clear circles.



Graph Model: Channel Intersection Graph

- Channels are represented as edges.
- Channel intersections are represented as vertices.
- Edge weight represents channel capacity.
- Extended channel intersection graph: terminals are also represented as vertices.

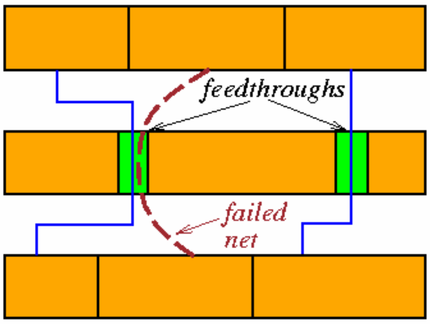


Global-Routing Problem

- Given a netlist N={N_1, N_2, …, N_n }, a routing graph $G = (V, E)$, find a Steiner tree T_i for each net N_i, $1 \le i \le n$, such that $U(e_j) \le c(e_j)$, $\forall\, e_j \in E$ and $\sum_{i=1}^{n} L(T_i)$ is minimized, where
  - $c(e_j)$: capacity of edge e_j;
  - x_{ij}=1 if e_j is in T_i; x_{ij} = 0 otherwise;
  - $U(e_j) = \sum_{i=1}^{n} x_{ij}$: # of wires that pass through the channel corresponding to edge e_j;
  - $L(T_i)$: total wirelength of Steiner tree T_i.
- For high-performance, the maximum wirelength ($\max_{i=1}^{n} L(T_i)$) is minimized (or the longest path between two points in T_i is minimized).

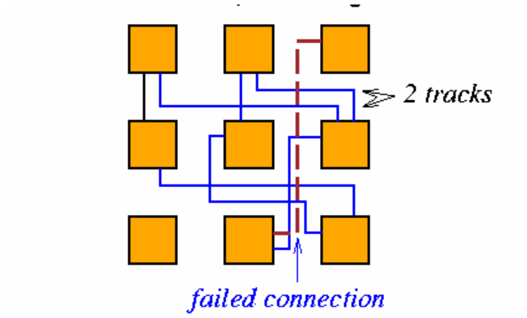
Global Routing in Standard Cell

- Objective
  - Minimize total channel height.
  - Assignment of **feedthrough**: Placement? Global routing?
- For high performance,
  - Minimize the maximum wire length.
  - Minimize the maximum path length.



Global Routing in Gate Array

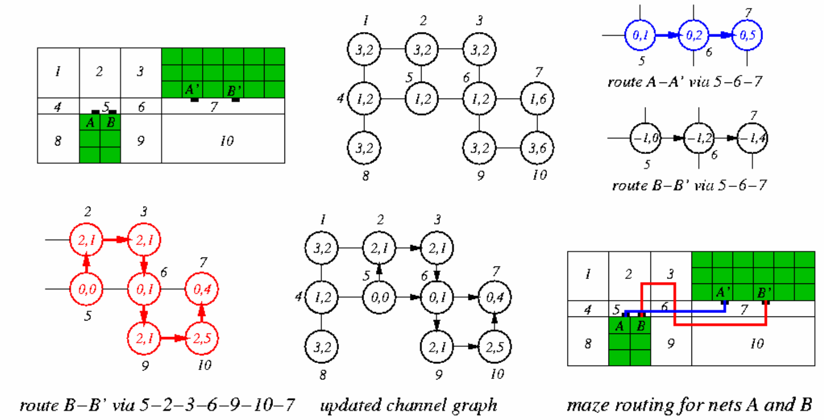
- Objective
  - **Guarantee 100% routability.**
- For high performance,
  - Minimize the maximum wire length.
  - Minimize the maximum path length.



Each channel has a capacity of 2 tracks.

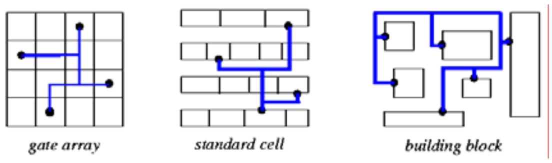
Global-Routing: Maze Routing

- Routing channels may be modelled by a weighted undirected graph called **channel connectivity graph**.
- Node ↔ channel; edge ↔ two adjacent channels; capacity: (*width*, *length*)

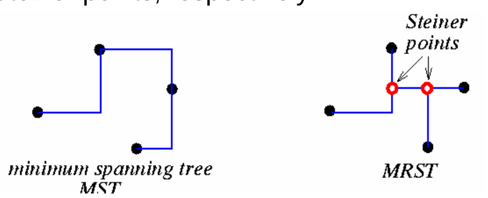


The Routing-Tree Problem

- **Problem:** Given a set of pins of a net, interconnect the pins by a "routing tree."

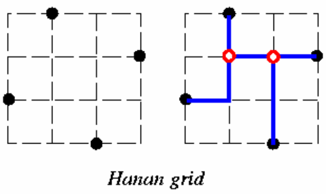


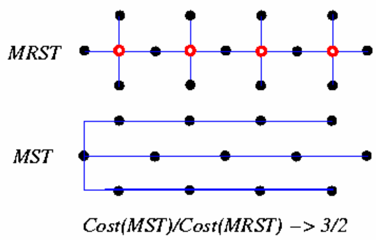
- **Minimum Rectilinear Steiner Tree (MRST) Problem:** Given *n* points in the plane, find a minimum-length tree of rectilinear edges which connects the points.
- *MRST*(*P*) = *MST*(*P* ∪ *S*), where *P* and *S* are the sets of original points and Steiner points, respectively.



Theoretic Results for the MRST Problem

- **Hanan's Thm:** There exists an MRST with all Steiner points (set *S*) chosen from the intersection points of horizontal and vertical lines drawn points of *P*.
  - Hanan, "On Steiner's problem with rectilinear distance," *SIAM J. Applied Math.*, 1966.
- **Hwang's Theorem:** For any point set *P*, $\frac{Cost(MST(P))}{Cost(MRST(P))} \le \frac{3}{2}$.
  - Hwang, "On Steiner minimal tree with rectilinear distance," *SIAM J. Applied Math.*, 1976.
- Other existing approximation algorithm: Performance bound 61/48 by Foessmeier *et al*.





Iterated 1-Steiner Heuristic for MRST

- Kahng & Robins, "A new class of Steiner tree heuristics with good performance: the iterated 1-Steiner approach," *ICCAD-90*..

```
Algorithm: Iterated_1-Steiner(P)
P: set P of n points.
begin
S ← ∅;
    /* H(P ∪ S): set of Hanan points */
    /* ΔMST(A, B) = Cost(MST(A)) - Cost(MST(A ∪ B)) */
while (Cand ← {x ∈ H(P ∪ S)| Δ MST(P ∪ S, {x}) > 0 } ≠ ∅ ) do
    Find x ∈ C and which maximizes  Δ MST(P ∪ S), {x});
    S ← S ∪ {x};
    Remove points in S which have degree ≤ 2 in MST(P ∪ S);
Output MST(P ∪ S);
end
```