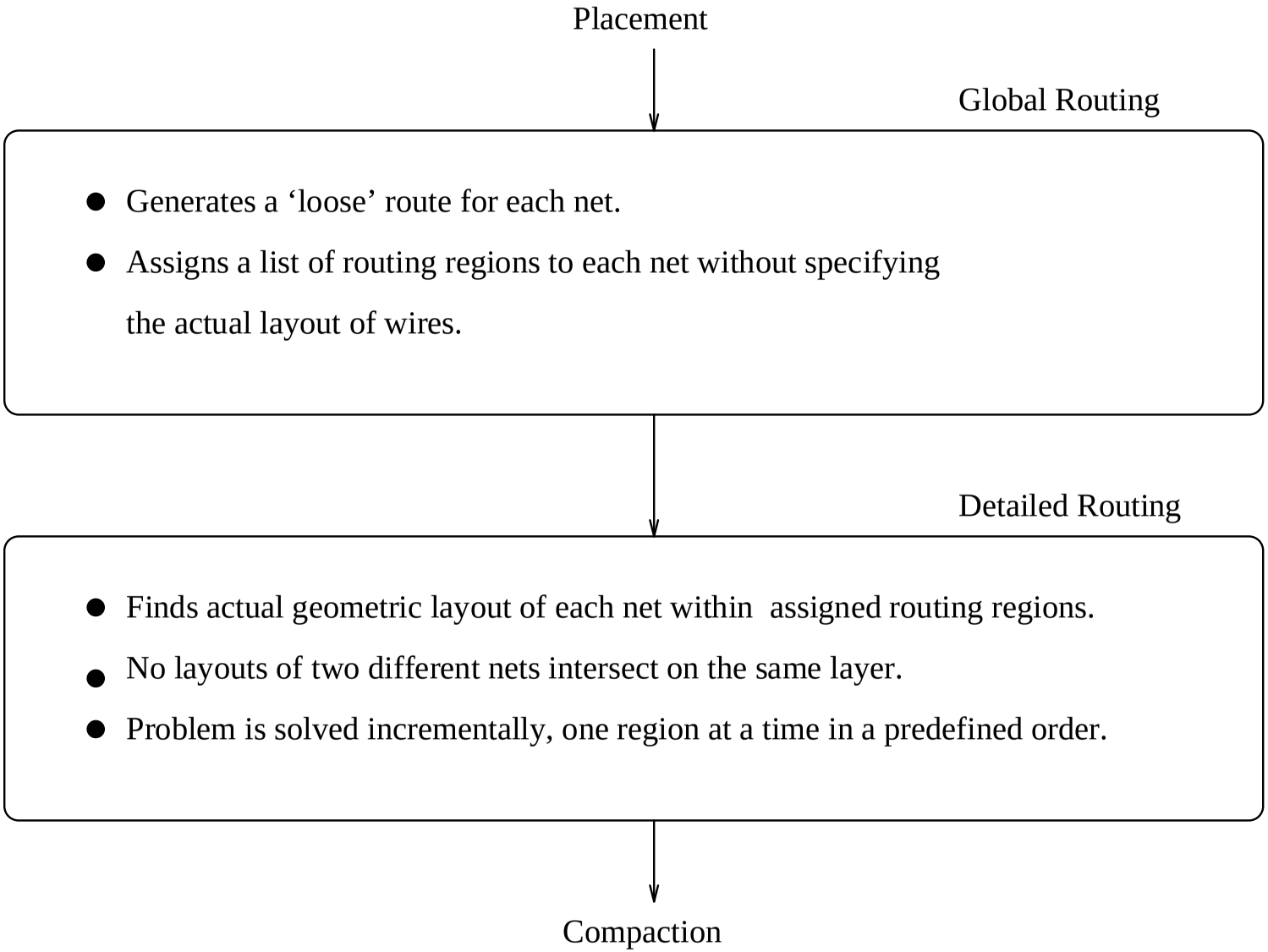
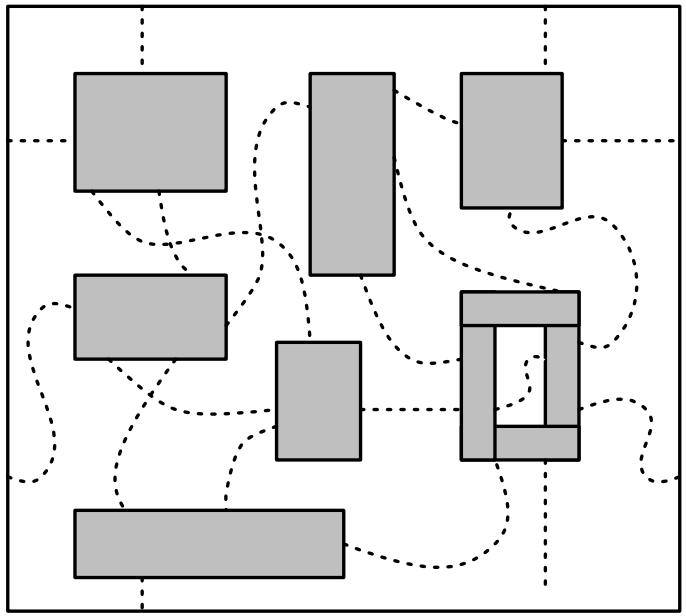
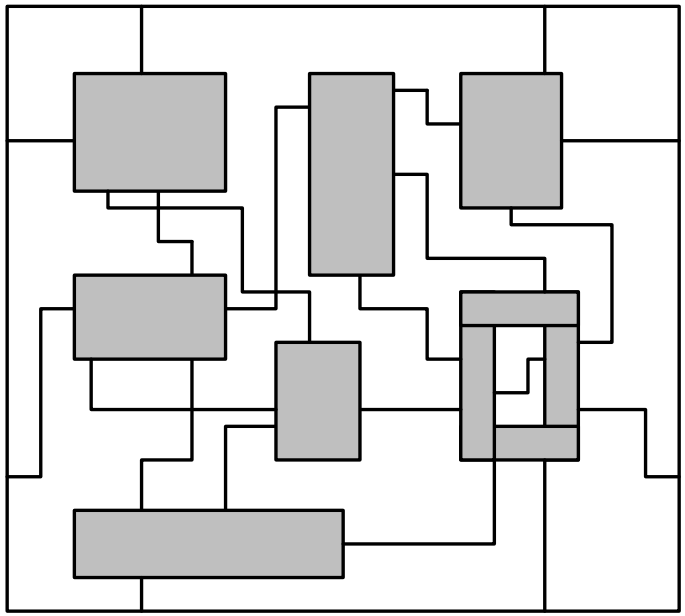
Detailed Routing



A Routing Example

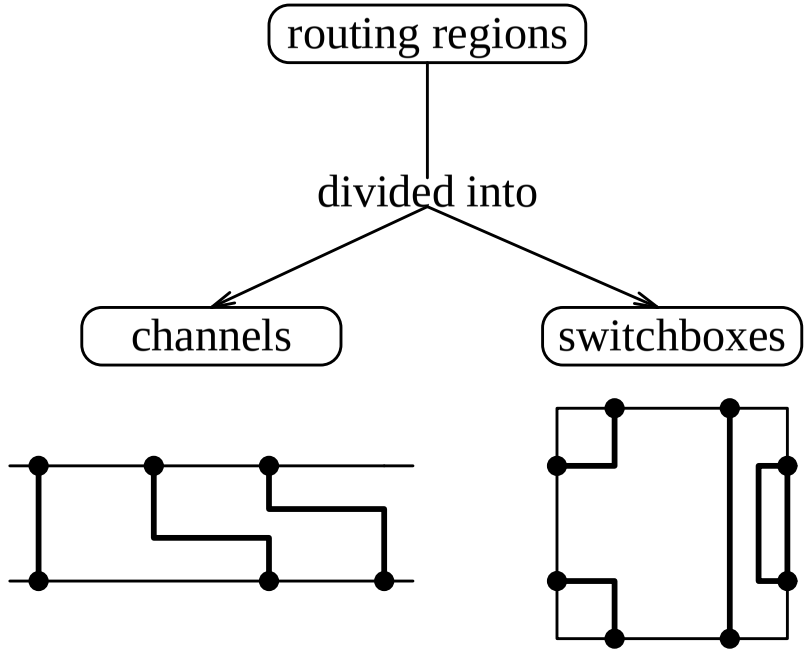


Global Routing
(a)

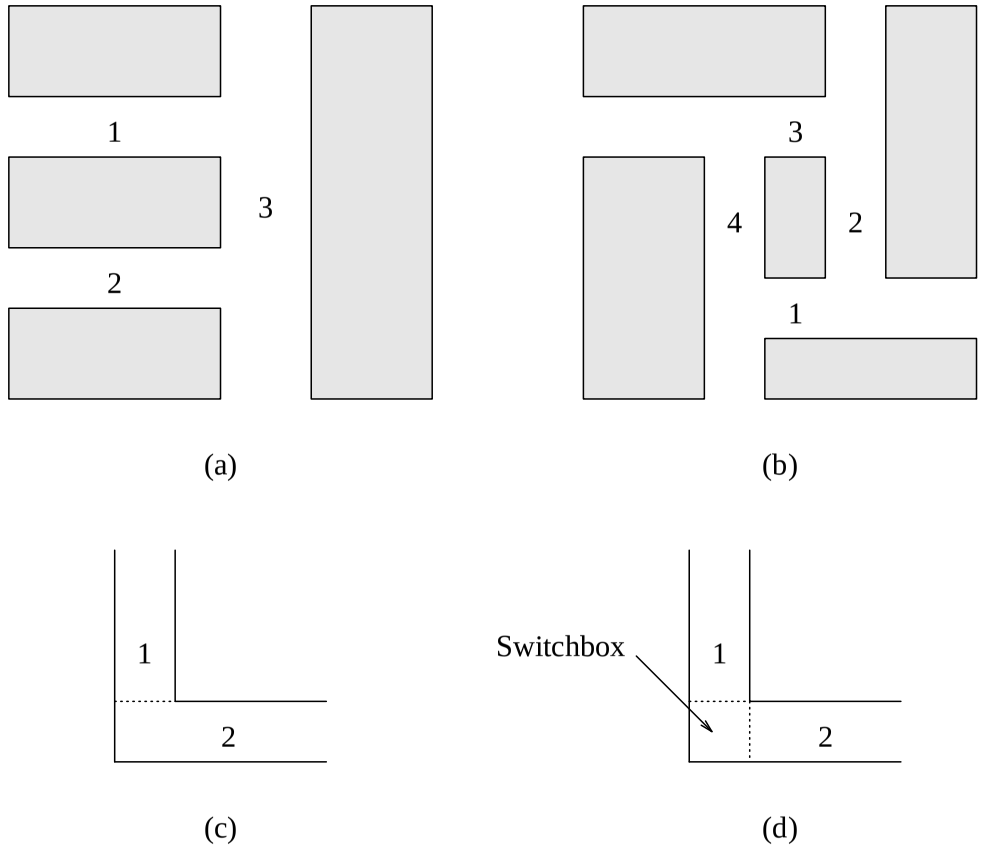


Detailed Routing
(b)

Channels and Switchboxes



Order of Routing Regions and L-channels



(a) No conflicts in case of routing in the order of 1, 2 and 3.

(b) No ordering is possible to avoid conflicts.

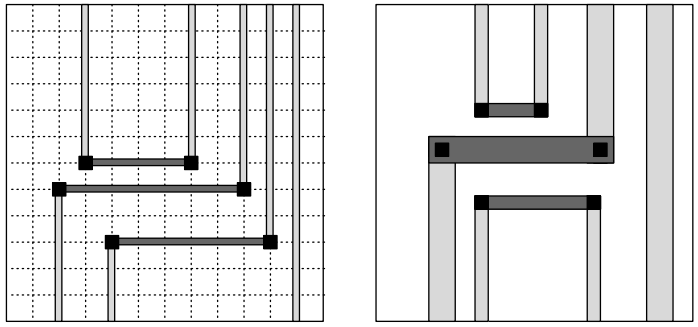
(c) The situation of (b) can be resolved by using L-channels.

(d) L-channles can be decomposed into channals and a switchbox.

Routing Considerations

- Number of terminals (i.e. two terminal vs. multiterminal nets)
- Net width (i.e. power and ground vs. signal nets)
- Via restrictions (i.e. stacked vs. conventional vias)
- Boundary type (i.e. regular vs. irregular)
- Number of layers (i.e. two vs. three layer model)
- Net types (i.e. critical vs. non-critical nets)

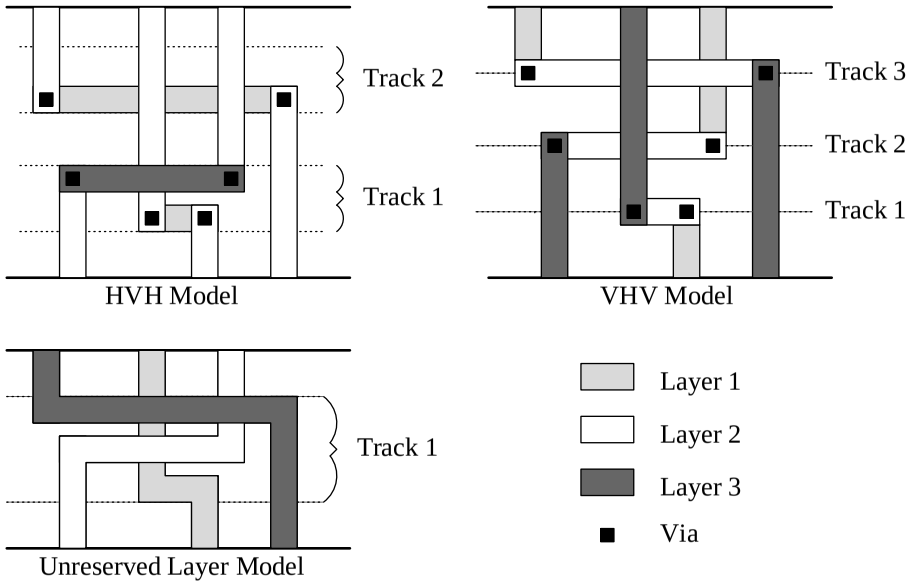
Routing Models



(a) Grid-based (b) Gridless

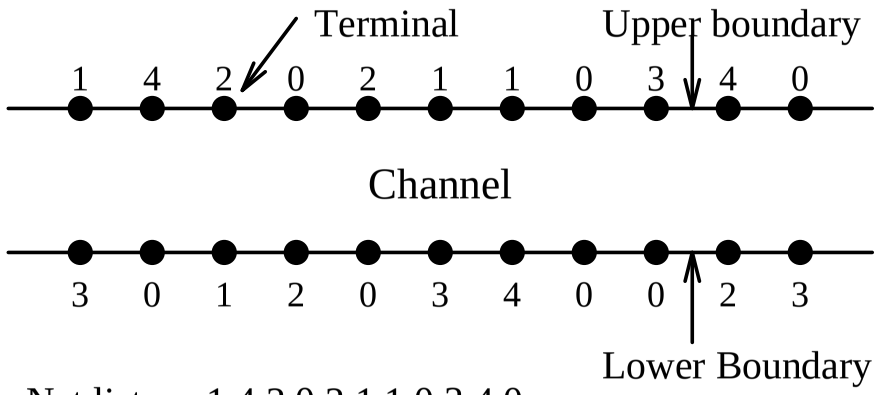
- **Grid-based model**
  1. A grid is super-imposed on the routing region.
  2. Wires follow paths along the grid lines.
- **Gridless model**
  1. Any model that does not follow this 'gridded' approach.

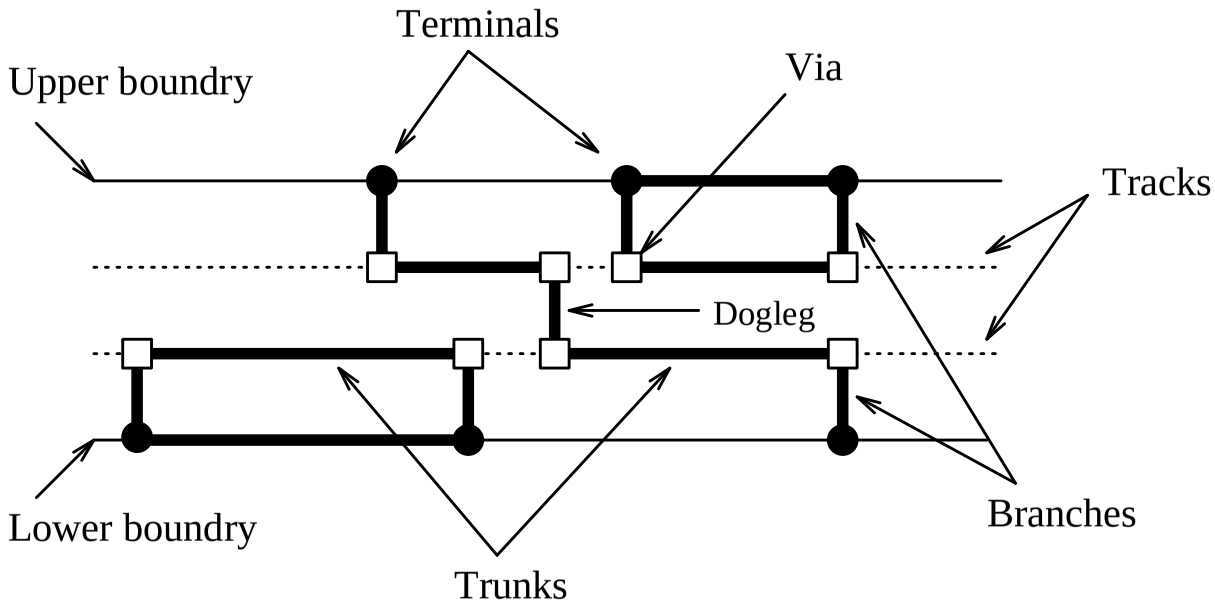
Models for Multi-layer Routing



- **Unreserved Layer Model:** Any net segment is allowed to be placed in any layer.
- **Reserved Layer Model:** Certain type of segments are restricted to particular layer(s).
  1. Two-Layer
     (a) HV (Horizontal-Vertical)
     (b) VH
  2. Three-Layer
     (a) VHV
     (b) HVH

Terminology for Channel Routing Problems



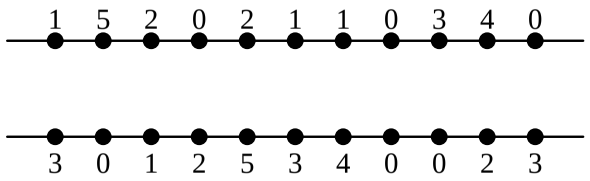


Channel Routing Problem Formulation

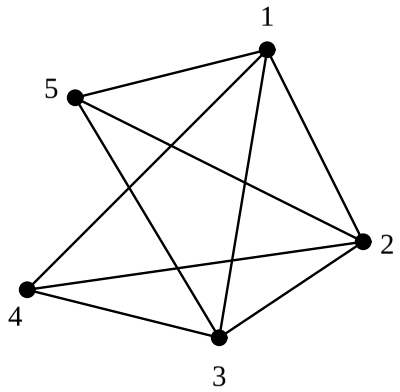
- Assignments of horizontal segments of nets to tracks.
- Assignments of vertical segments to connect
  1. horizontal segments of the same net in different tracks, and
  2. the terminals of the net to horizontal segments of the net.
- Horizontal and vertical constraints must not be violated.
  1. Horizontal constraints between two nets: The horizontal span of two nets overlaps each other.
  2. Vertical constraints between two nets: There exists a column such that the terminal on top of the column belongs to one net and the terminal on bottom of the column belongs to the other net.
- Channel height is minimized.

Horizontal Constraint Graph (HCG)

- HCG $G = (V, E)$ is undirected graph where $V = \{v_i | v_i$ represents a net $n_i\}$ and $E = \{(v_i, v_j) |$ a horizontal constrain exists between n_i and $n_j\}$.



(a)

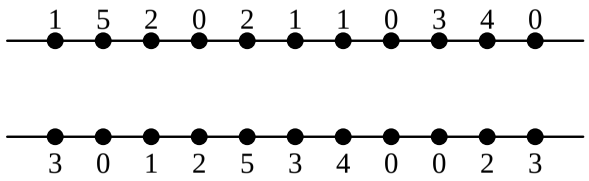


(b)

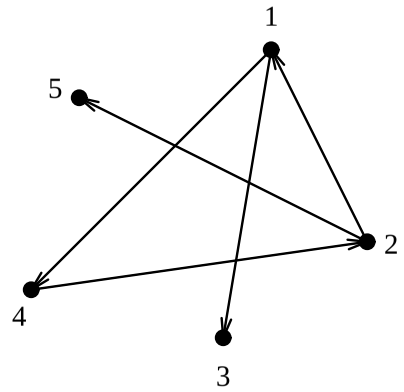
A routing problem and its HCG

Vertical constraint Graph (VGC)

- VCG $G = (V, E)$ is directed graph where $V = \{v_i | v_i$ represents a net $n_i\}$ and $E = \{(v_i, v_j) |$ a vertical constrain exists between n_i and $n_j\}$.



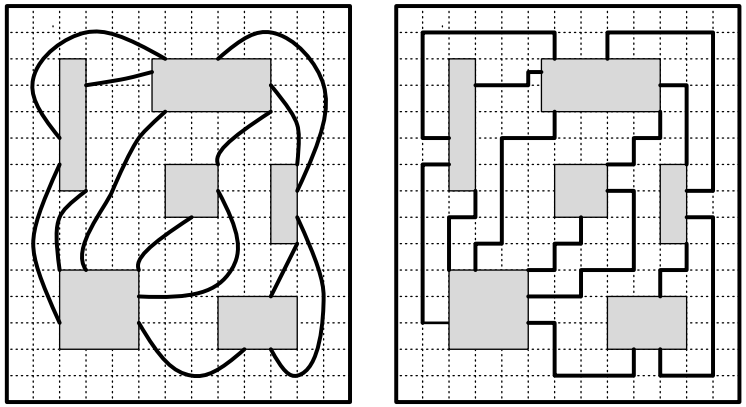
(a)



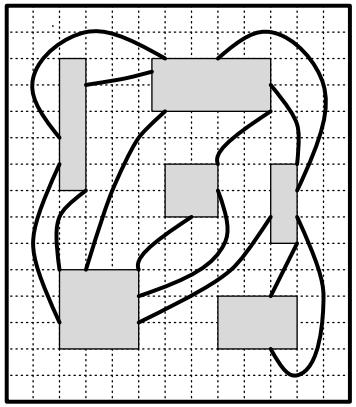
(b)

A routing problem and its VCG

Single-Layer Routing Problems



A routable single-layer routing problem



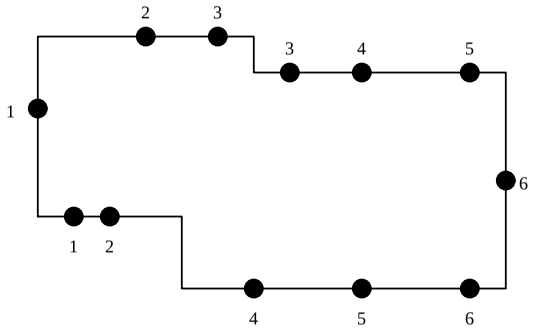
An unroutable single-layer routing problem

Single-Layer Routing Algorithms

1. General River Routing Algorithm
2. Single Row Routing Algorithm
   (a) Basic Single Row Routing Algorithm
   (b) Algorithm for Street Congestion Minimization
   (c) Algorithm for Minimizing Doglegs

General River Routing Problem

- A special case of single-layer routing problem.
- All terminals lie on the boundary of the region.
- All nets are two-terminal nets.
- There are no blocks in the region.
- Nets must be planar.



Example

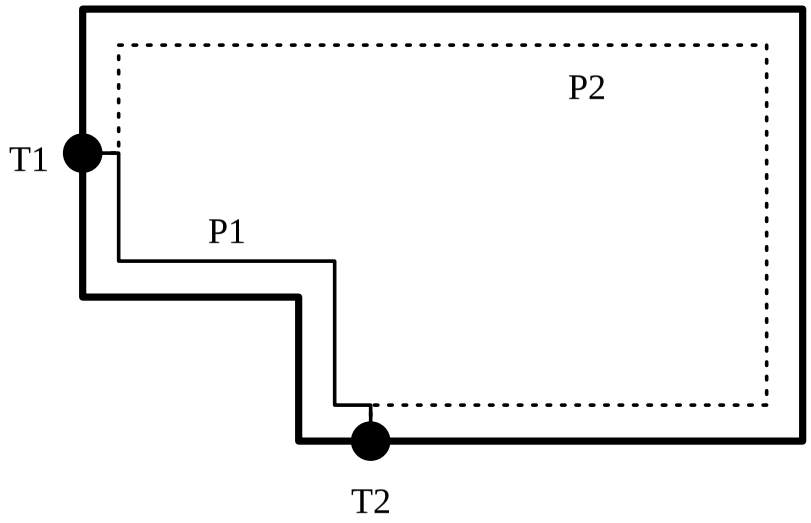
General River Routing Algorithm

- Routing in arbitrary shaped rectilinear routing regions
- Gridless-based
- Nets are routed one-by-one.
- Each net is routed as close to the boundary as possible in counterclock direction.
- Guarantees to find a solution if one exists.
- The algorithm consists of following phases:
  1. Starting Terminal Assignment
  2. Net Ordering
  3. Path Searching
  4. Corner Minimization

Hsu *DAC* 1983

Starting Terminal Assignment

- Each net is routed in the counter-clock direction.
- A starting terminal is assigned to each net such that the length of its layout along boundary is shorter.



T1 is starting terminal since P1 is shorter than P2.

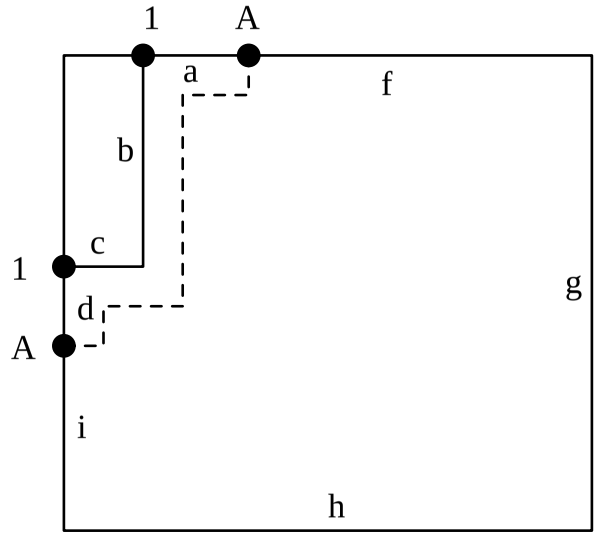
Net Ordering

- Nets are routed in an order such that all nets whose layouts are 'contained' in the layout of the net to be routed have been routed.

```
Algorithm NET-ORDERING
begin
    for i = 1 to 2n do
        if END-TERMINAL(T_i) then MATCHED(T_i) = 0;
        else MARKED(T_i) = 0;
    stack = φ; i = 1;
    T = any terminal in the circular list;
    while i ≤ n do
        if START-TERMINAL(T) and MARKED(T) = 0
        then PUSH(T, stack);
            MARKED(T) = 1;
        else if END-TERMINAL(T) and MATCHED(T) = 0
        then T1 = POP(stack);
            if T = T1 then
                MATCHED(T) = 1; ASSIGN-NUMBER(i, NET(T)); i = i + 1;
            else exit;
        T = next terminal in the circular list;
end.
```

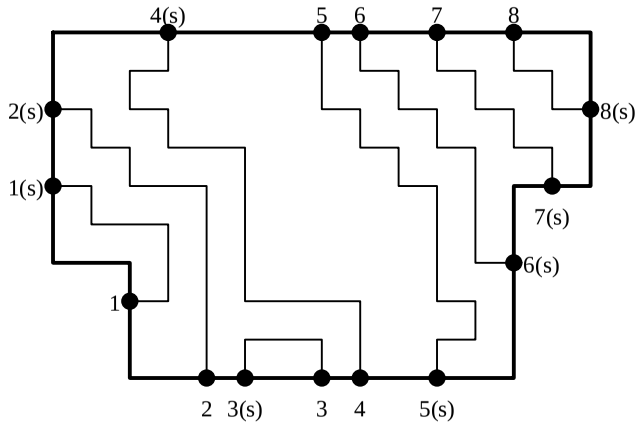
Path Searching

- Based on the net order, each net is routed as close to the pseudo-boundary as possible.
- Pseudo-boundary is formed by the boundary and the paths of nets that have been routed.

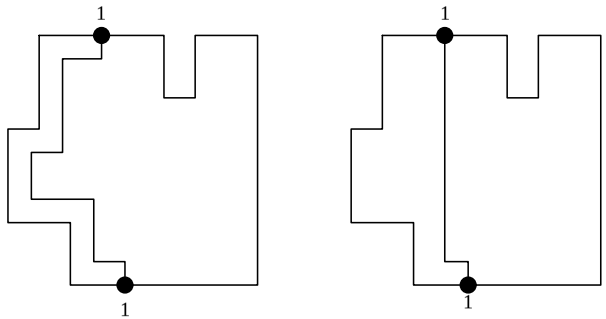


Corner Minimization

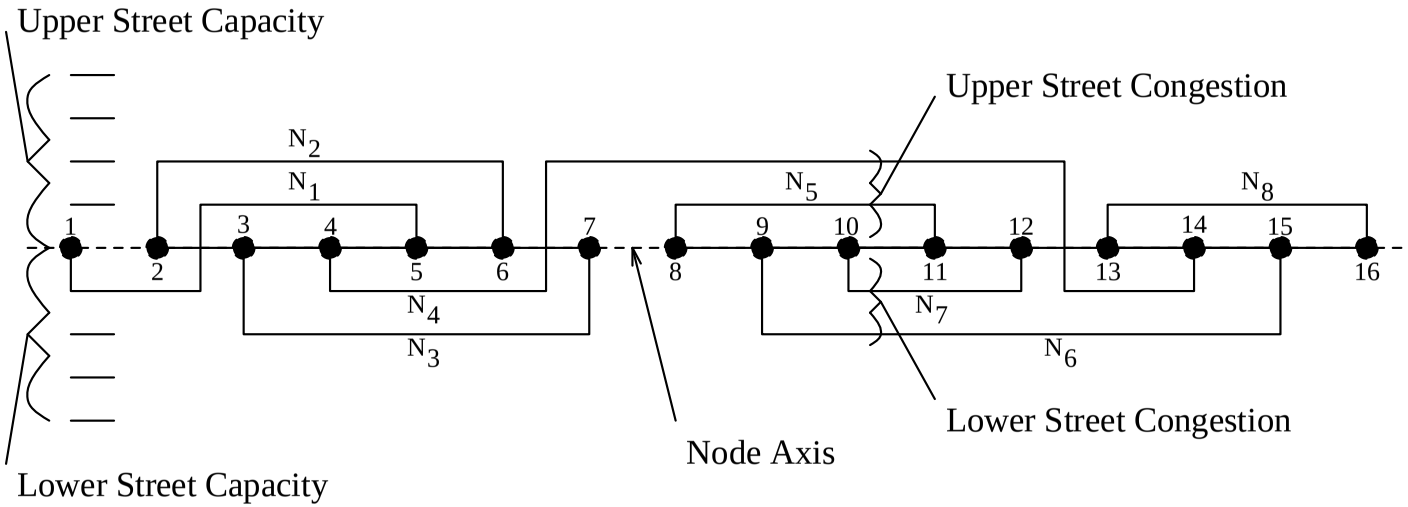
- After routing, corners are a lot more denser than the center.



- Length of nets can be minimized by flipping corners toward the inside of the routing region.

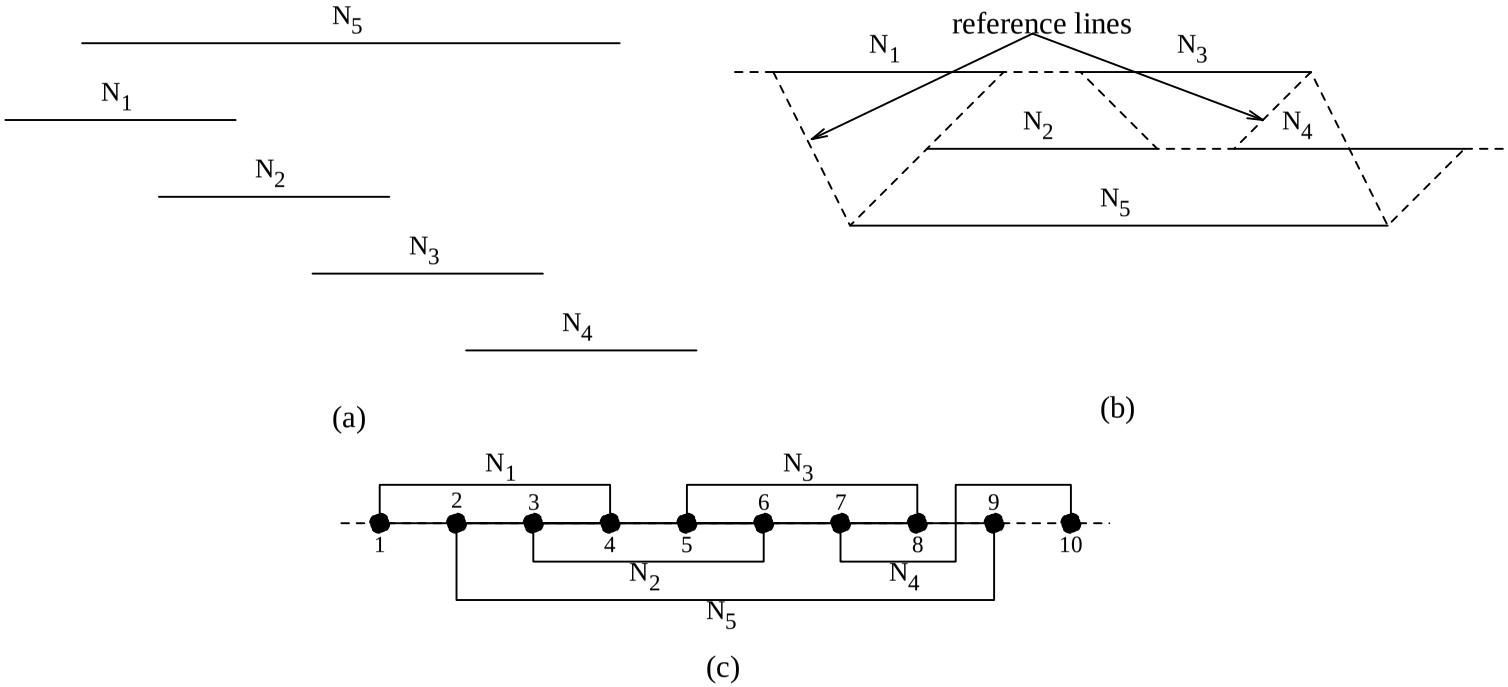


Terminology for Single Row Routing Problems (SRRP)

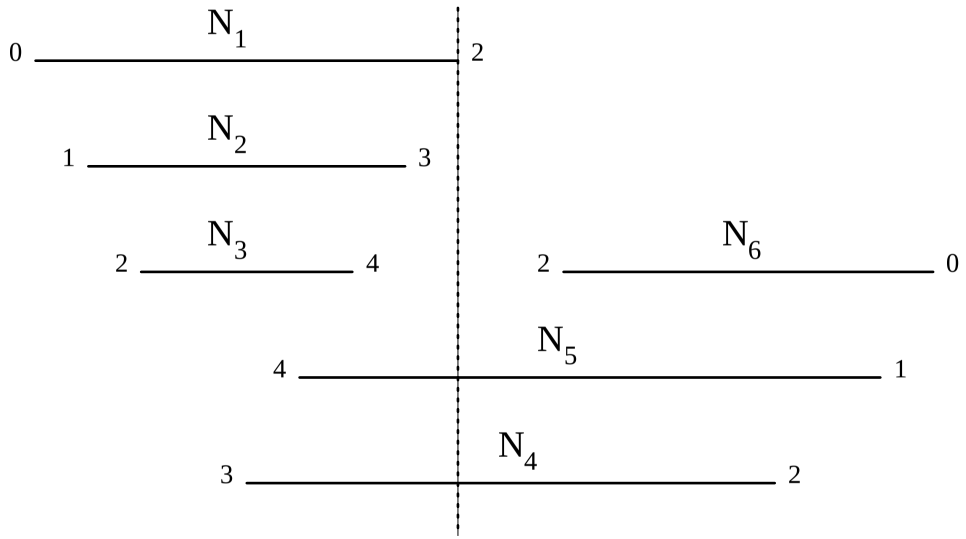


Basic Single Row Routing Algorithm

- Nets are represented by intervals.
- Nets or segments of nets above reference lines are mapped onto paths in the upper street while those below reference lines are mapped onto paths in the lower stree.



Street Congestion Minimization



- Let q_{max} and q_{min} be the maximum and minimum over the cut numbers of all nets respectively.
- In the example, $q_{max} = 4$ and $q_{min} = 2$.
- The problem of finding the layout with minimum congestion is NP-Hard.
- Let Q_0 be the maximum over the upper and lower street congestions, then $Q_0 \geq \max\{q_{min}, \lceil \frac{q_max}{2} \rceil\}$.

Algorithm for Street Congestion Minimization

- Cliques are found in the interval graph.
- If the clique intersection of two neighboring (pseudo-) cliques is high,

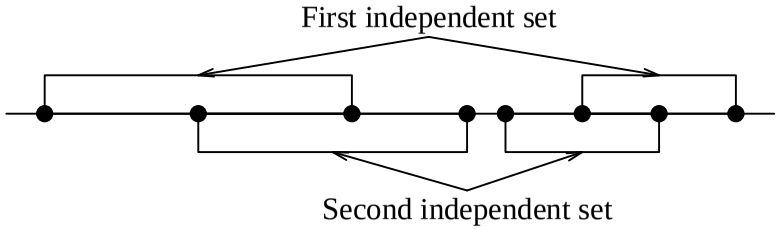
  a pseudo-clique is formed by combining them.
- Maximum pseudo-clique is routed first. Other pseudo-cliques are then routed.
- Time complexity is $O(N^2 \log n)$ where n is the number of intervals.

Algorithm SRRP-ROUTE ( )
begin
    FIND-CLIQUES (L, C)
    COMBINE-CLIQUES (L, C, D)
        (* D contains super-cliques $SC_j, j = 1, \ldots, r$ *)
    MAX-PSEUDO-CLIQUE (D, SC_k)
    SOLVE (SC_k, M)
    for $j = 1$ to $k - 1$
        INSERT (SC_j)
    for $j = k + 1$ to r
        INSERT (SC_j)
end.

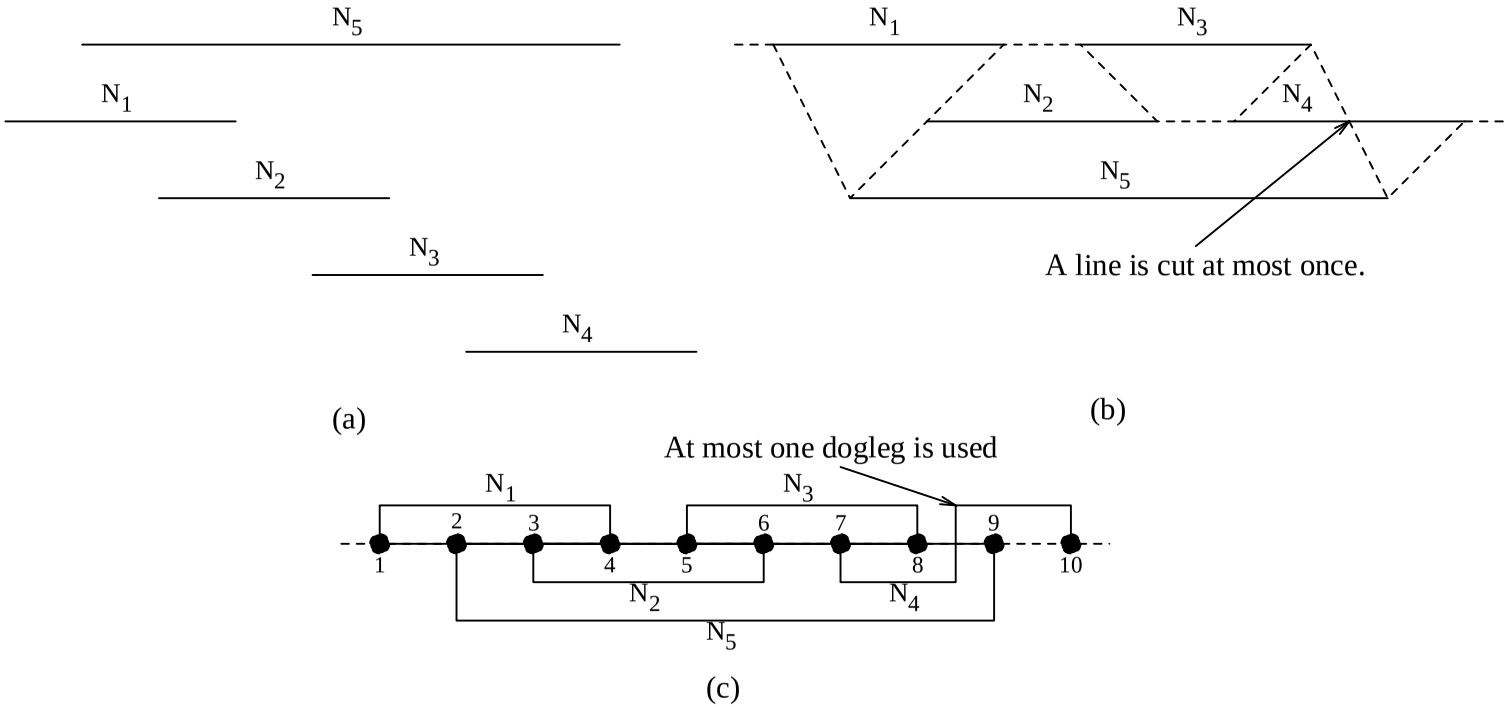
Sherwani, Deogun, and Roy *Journal of Circuits, Systems and Computers* June 1993

Dogleg Minimization

- An SRRP can be routed without doglegs. <==> Overlap graph is bipartite.

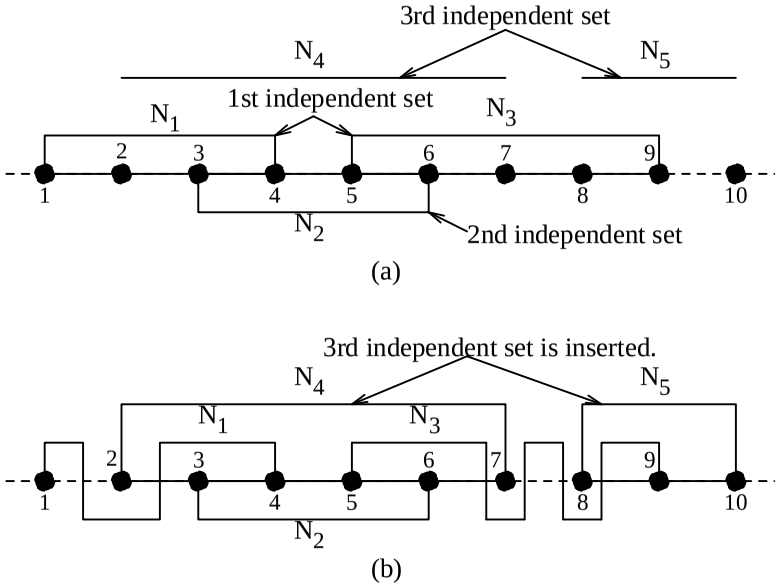


- An SRRP can be routed with at most 1 dogleg. <==> Containment graph is null.



Features of Algorithm for Minimizing Doglegs

- The interval graph is decomposed into k independent sets.
- First two independent sets are routed using upper and lower streets.
- Remaining independent sets are inserted using maximum $O(k)$ doglegs per net.
- Time complexity is $O(n \log n)$ where n is the number of nets.



Wu, Sarrafzadeh, and Sherwani *Transactions of Circuits and Systems* May 1992

Algorithm for Minimizing Doglegs

Algorithm K-DOGLEG-I()
begin
Phase 1:
(* Use Left_edge algorithm to decompose $\mathcal{N}$ into k independent net lists *)
(* lists ($\mathcal{N}_1$, $\mathcal{N}_2$,..., $\mathcal{N}_k$) of $\mathcal{N}$. *)
$\mathcal{N}_i$=LEDGE($\mathcal{N}$); $(i = 1, .., k)$.
for ($N_i \in \mathcal{N}_1$ $i = 1, .., m_1$) **do** (* Assign $\mathcal{N}_1$ to the upper street. *)
$T_1^U = T_1^U \cup N_i$;
(* Assign N_2 to the lower street. *)
for ($N_i \in \mathcal{N}_2$ $i = 1, .., m_2$) **do**
$T_1^B = T_1^B \cup N_i$;
Phase 2:
(* Insert the remaining independent net lists.*)
$t = U$; $u = 1$; $l = 1$;
for (G_I^i $i = 3, .., k$) **do**
for ($N_j \in \mathcal{N}_i (j = 1, .., m_i)$) **do**
$k = \min\{q | 1 \leq q \leq p, N_j \in T_q^t\}$; (* Find the smallest track which contains N_j. *)
if (N_j contained by previously routed net at T_k^t)
then
INSERT(N_j, T_k^t); (* Insert N_j under T_k^t. *)
else
$T_p^t = T_p^t \cup N_j$; (* Assign the new net to the outer track. *)
if ($t = U$) **then** (* Switch street. *) $t = B$; $l = l + 1$; $p = l$;
else $t = U$; $u = u + 1$; $p = u$;
end.

Two-Layer Channel Routing Algorithms

- Left-Edge Algorithms (LEA)
  1. Basic Left-Edge Algorithm
  2. Dogleg Router
  3. Symbolic Channel Router: YACR2
- Constraint-Graph Based Routing Algorithms
  1. Net Merge Channel Router
  2. Glitter: A Gridless Channel Router
- Greedy Channel Router
- Hierarchical Channel Router

Features of Basic Left-Edge Algorithm

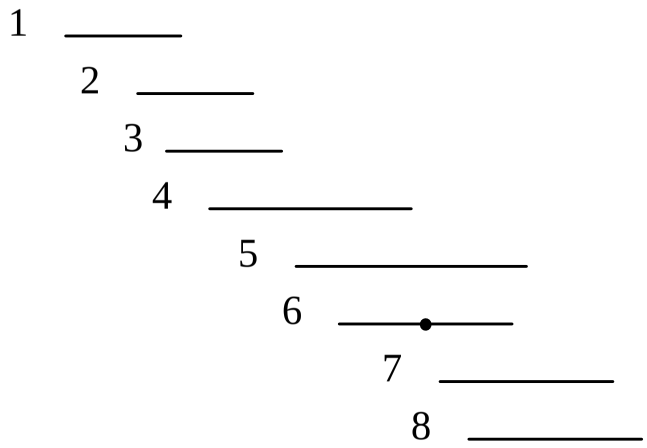
- Only two-terminal nets in the problem.
- No Vertical constraint in the problem.
- HV layer model is used.
- Doglegs are not allowed.
- Nets are sorted according to the x-coordinate of the leftmost terminal of the net.
- Nets are routed one-by-one according to the order.
- For a net, tracks are scanned from top to bottom, and the first track that can accommodate the net is assigned to the net.
- It produces a routing solution with minimum number of tracks.

Hashimoto and Stevens *Proceedings of the 8th Design Automation Workshop* 1971

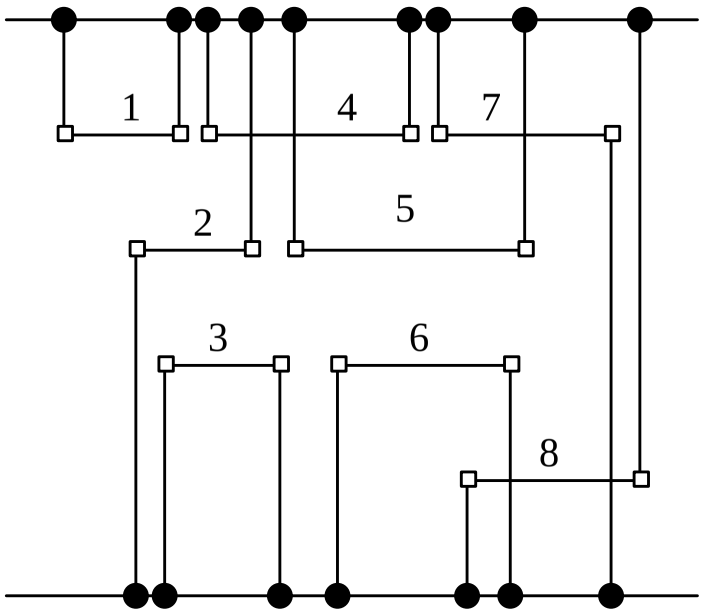
Basic Left-Edge Algorithm

Algorithm LEA $(\mathcal{N}, \mathcal{I})$
begin
    FORM-INTERVAL$(\mathcal{N}, \mathcal{I})$;
    FORM-HCG($\mathcal{I}$,HCG);
    $d =$ DENSITY(HCG);
    let $T = \{T_1 T_2, \ldots, T_d\}$ denote the set of routing
        tracks from top to bottom;
    SORT-INTERVAL($\mathcal{I}$);
    for $i = 1$ to n **do**
        for $j = 1$ to d **do**
            if DOES-NOT-OVERLAP(I_i, T_j) **then**
                assign interval I_i to T_j;
    for $i = 1$ to n **do**
        (* connect the vertical segments of net N_i to its *)
        (* horizontal segment *)
        VERTICAL-SEGMENT(left(I_i), left(N_i));
        VERTICAL-SEGMENT(right(I_i), right(N_i));
end.

Example of LEA



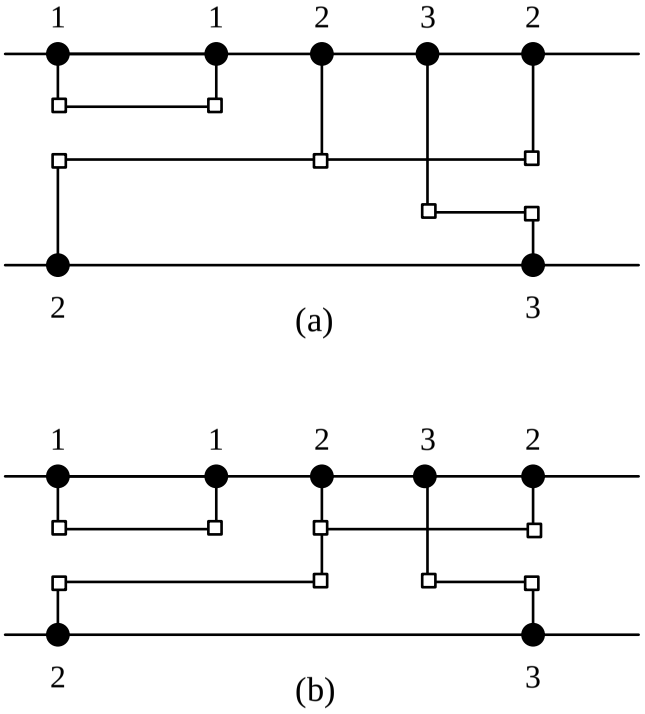
(a)



(b)

Dogleg Router

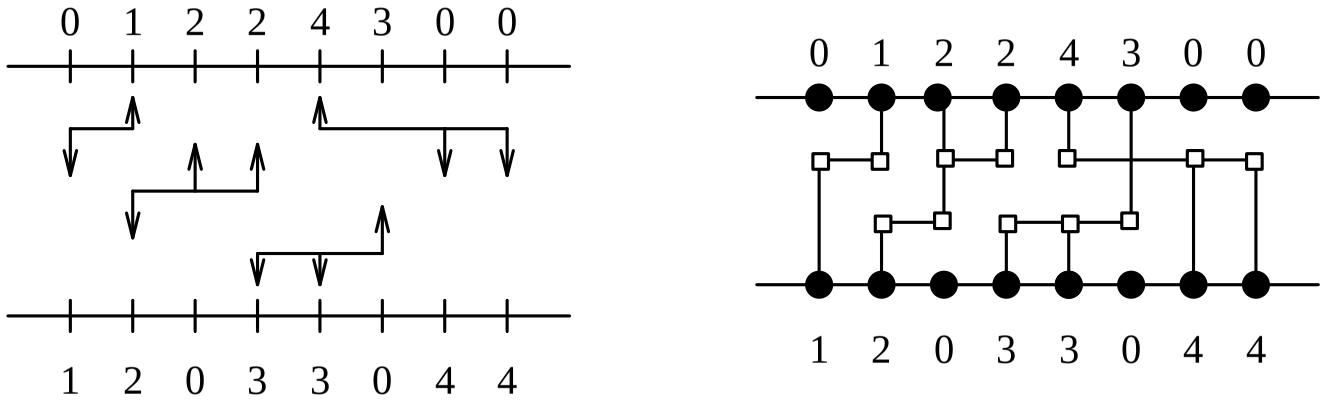
- Drawback of LEA: the entire net is on a single track.
- Doglegs are used to place parts of a net on different tracks, thereby minimizing channel height.



Using a dogleg to reduce channel height

Dogleg Router

- Each Multi-terminal net is broken into a set of two-terminal nets.
- Two parameters are used to control routing:
  1. range: Determine the number of consecutive two-terminal subnets of the same net that can be placed on the same track.
  2. routing sequence: Specifies the starting position and the direction of routing along the channel.
- Modified LEA is applied to each subnet.



Example of Dogleg Router

Vertical Constraint Violations

- Drawback of LEA: Vertical constraints are not allowed.

  Vertical constrants may be violated if LEA is used.

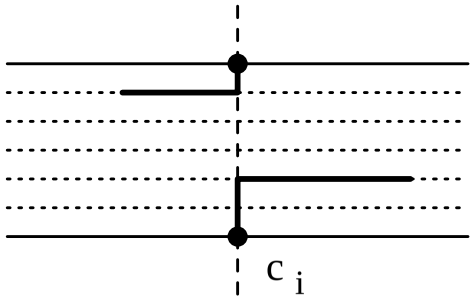
- Vertical constraint violations are localized problem

  and may be resolved by

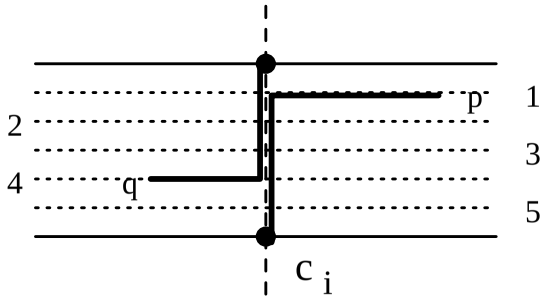
  1. local rip-up and reroute
  2. localized maze routing

Features of YACR2

- YACR2 uses two steps:
  1. Horizontal segments are assigned to tracks such that $vof(c_i)$ is minimized for each column c_i.
  2. For each column c_i:
     (a) If $vof(c_i) = 0$, appropriate vertical segments are placed in c_i;
     (b) If $vof(c_i) > 0$: localized maze routing techniques (Maze1, Maze2) are used to resolve the vertical constraint violations in c_i.



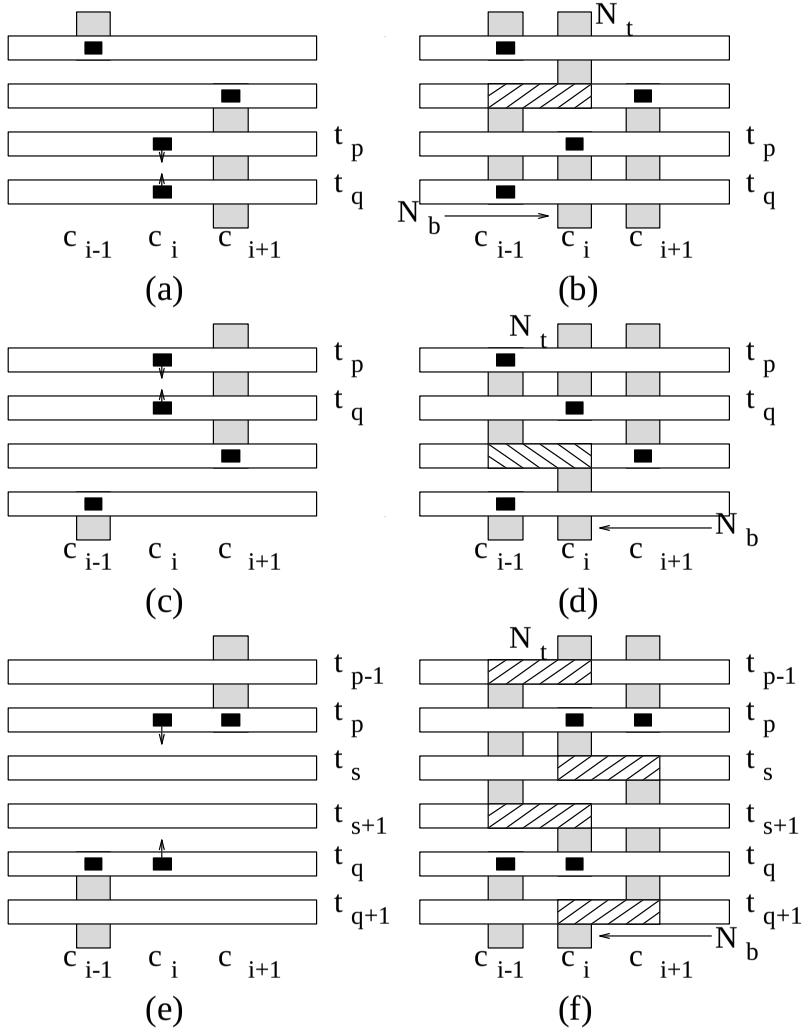
No vertical constraint violation
$vof(c_i) = 0$



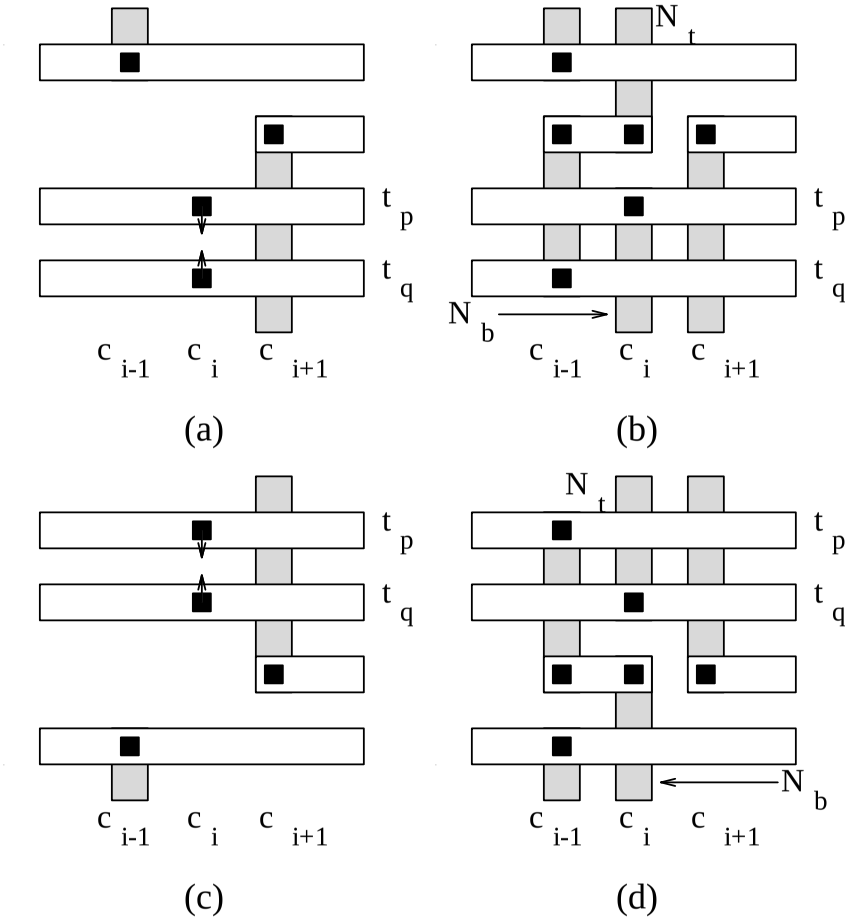
Vertical constraint violation
$vof(c_i) = q\text{-}p\text{+}1 = 4\text{-}1\text{+}1 = 4$

Reed, Sangiovanni-Vincentelli and Santamauro *TCAD* 1985

Maze1 Routing of YACR2



Maze2 Routing of YACR2



Net Merge Channel Router(YK Algorithm)

- YK algorithm considers both HCG and VCG.
- Nets are assigned to minimize the effect of vertical constraint chains in VCG.
- Does not allow doglegs and cannot handle vertical constraint cycles.
- Algorithm consists of two major steps:
  1. Zone representation of Horizontal segements
  2. Merging of Nets
- These two steps are carried out so at minimize vertical constraints and track assignment.

T. Yoshimura and E. S. Kuh, IEEE-TCAD, 1982.

Steps of YK Algorithm

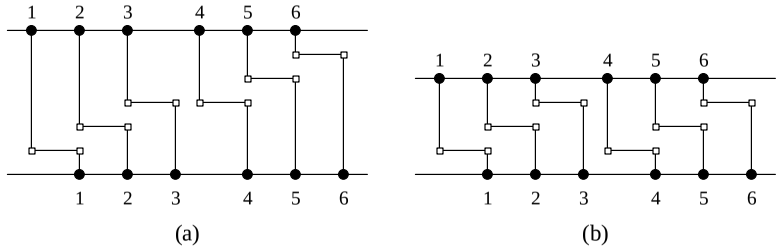
1. Zone representation of Horizontal Segments:
   - Zones are maximal clique in the interval graph of horizontal net segements.
   - $S(i)$ = set of nets whose horizontal segments intersect column i.
   - Zone numbers are assigned to the columns at which $S(i)$ is maximum.
2. Merging of Nets:

   Two nets, N_i and N_j will be merged if
   - There is no edge between v_i and v_j in HCG.
   - No directed path exists between v_i and v_j in VCG.

   Same idea has been extended by Chen and Liu for three layers.

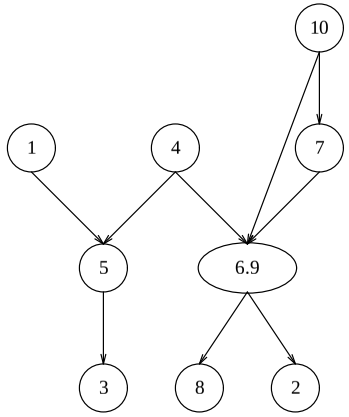
Y. Chen and M. Liu, IEEE-TCAD, 1984.

Net Merge Channel Router

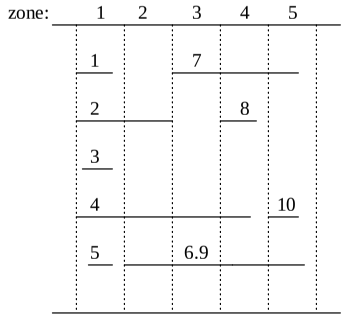


Effect of net merging on channel height

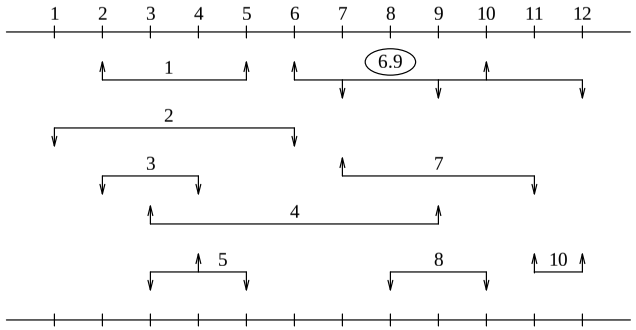
Example of Merging of Nets



(a)



(b)



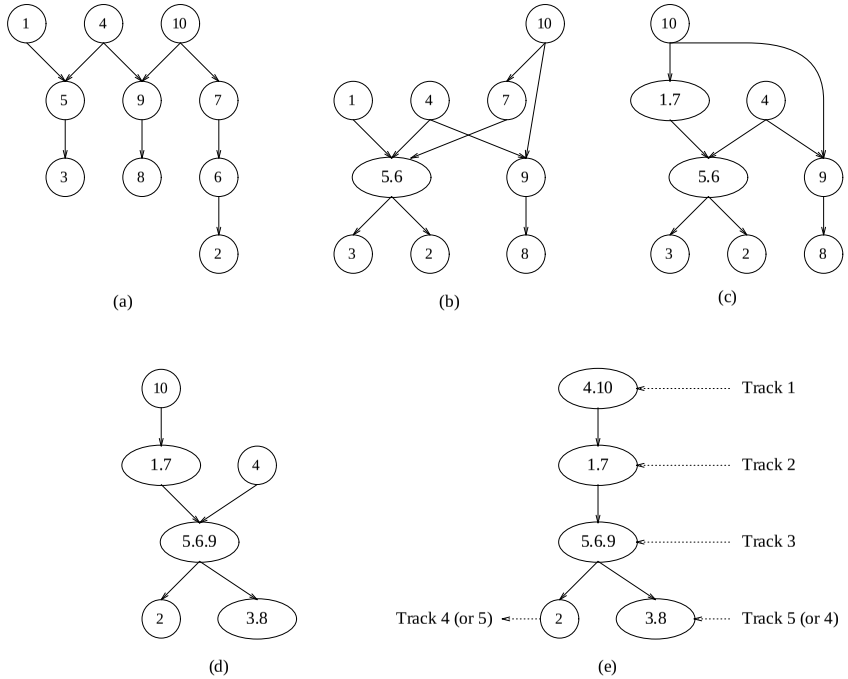
(c)

Algorithm for Merging Nets

Algorithm NET-MERGE
begin
$L = \phi$;
for $z = (z_1$ to $z_{t-1})$ **do**
$L = L + \{z_i - (z_i \cap z_{i+1})\}$;
$R = \{z_{i+1} - (z_i \cap z_{i+1})\}$;
L' =MERGE(L, R);
$L = L - L'$;
end.

Function MERGE merges two list L and R so as to minimize the increase in the longest path length in VCG.

Illustration of Algorithm NetMerge



Glitter: A Gridless Channel Router

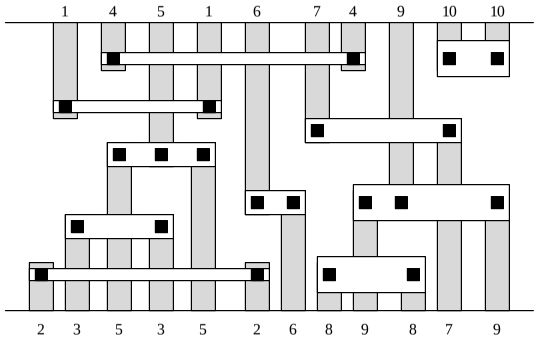
- First gridless, variable width channel router.
- Supports multiple layer technology and design rules.
- Terminals can be located at arbitrary positions
- No columns are track used for routing
- Nets are allowed to have different widths to satisfy design needs.
- It is a reserved-layer model routing algorithm.

H. H. Chen and E. Kuh, IEEE-TCAD, 1986.

Features of Glitter

- Construct a weighted constraint graph using HCG & VCG.
- The weight of the edge between two nodes is the minimum vertical distance required between the two corresponding nets.
- The idea is to assign directions to each undirected edge such that:
  1. No cylces are generated
  2. Total weight of maximum weighted directed path is minimized.
- It can be extended to accommodate irregularly-shaped channels.

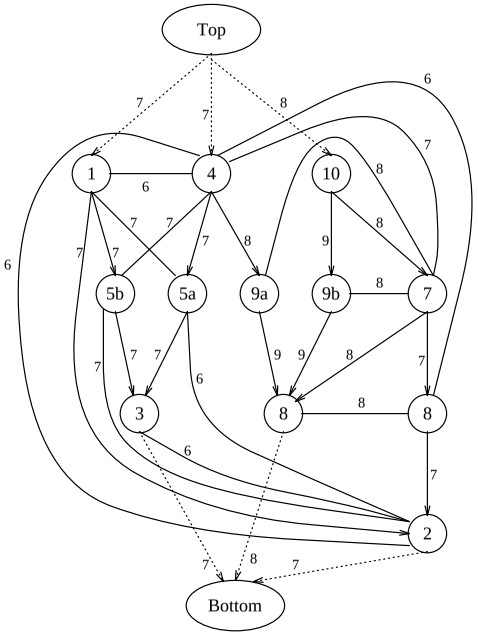
Glitter: A Gridless Channel Router



(a)

net	1	2	3	4	5	6	7	8	9	10
width	2	2	4	2	4	4	4	6	6	6

(b)



(c)

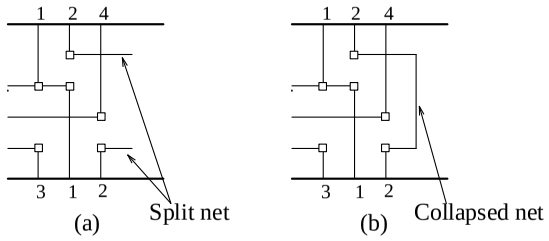
Greedy Channel Router

- Starting from the leftmost column, place all net segments column by column.
- Connect any terminal to the trunk segment of corresponding net.
- Collapse any split nets using a vertical segment.
- Try to reduce the range or distance between two tracks of same net.
- Try to move the nets closer to the boundary which contains the next terminal of that net.
- Add additional tracks if no tracks are available

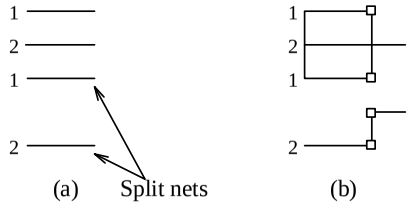
R. Rivest and C. Fiduccia, 19th DAC, 1982.

Examples of Greedy Channel Router

(a) A split net (b) The collapsed split net



Reducing the distance between split nets

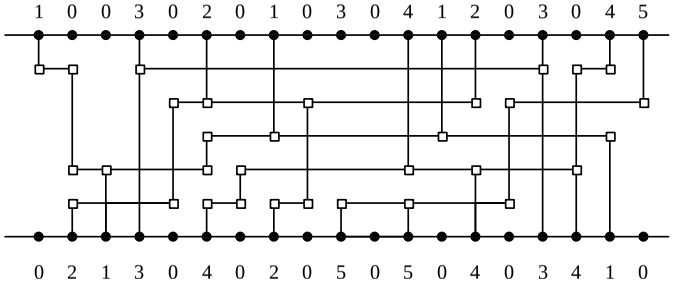


Greedy Channel Router

Algorithm GREEDY-CHANNEL-ROUTER ($\mathcal{N}$)
begin
    d = DENSITY($\mathcal{N}$);
(* calculate the lower bound of channel density *)
    insert d tracks to channel;
    for $i = 1$ to m **do**
        $T1$ = GET-EMPTY-TRACK;
        if $T1 = 0$ **then**
            ADD-TRACK($T1$);
            ADD-TRACK($T2$);
        else
            $T2$ = GET-EMPTY-TRACK;
            if $T2 = 0$ **then**
                ADD-TRACK($T2$);
        CONNECT($T_i, T1$);
        CONNECT($B_i, T2$);

Greedy Channel Router

join split nets as much as possible;
bring split nets closer by jogging;
bring nets closer to either top or bottom boundary;
while split nets exists **do**
increase number of column by 1;
join split nets as much as possible;
end.



Channel routed using a greedy router

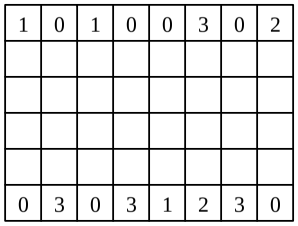
Hierarchical Channel Router

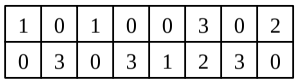
- Uses a divide and conquer approach.
- A routing problem in $m \times n$ grid is reduced to $2 \times n$ grid.
- Each column in these subgrids is considered as a supercell.
- Capacity of each vertical boundary is the sum of corresponding boundary capacities.
- Nets are routed one at a time in the $2 \times n$ grid.
- Each $2 \times n$, grid is partitioned into a $2 \times n$ grid.
- Terminal positions for the new $2 \times n$ are defined by the routing in previous hierarchy.

M. Burstein and R. Pelavin, 20th DAC, 1983.

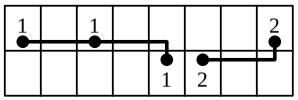
Example of Hierarchical channel router

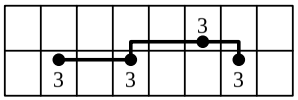
Reducing $(m \times n)$ grid to $(2 \times n)$ grid

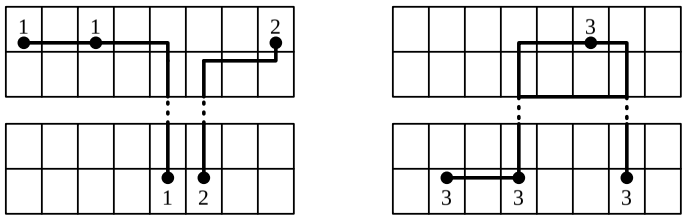




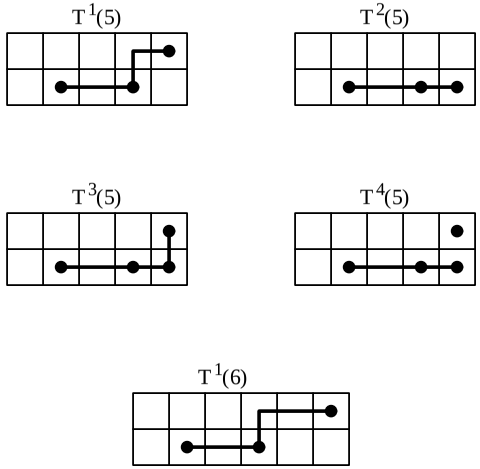
First and second levels of hierarchy







Recursively Computing $T^1(6)$ from $T^i(5), i = 1, 2, 3, 4$



Comparison of Two-Layer Channel Routers

Features	Algorithm						
	LEA	Dogleg	Y-K	Greedy	YACR2	Hierar-chical	Glitter
Model	grid-based	grid-based	grid-based	grid-based	grid-based	grid-based	grid-less
Dogleg	not allowed	allowed	allowed	allowed	allowed	allowed	not allowed
Layer assign-ment	reserved	reserved	reserved	reserved	reserved*	reserved	reserved
vertical constra-ints	not allowed	allowed	allowed	allowed	allowed	allowed	allowed
cyclic constra-ints	not allowed	not allowed	not allowed	allowed	allowed	not allowed	allowed

Comparison of Two-Layer Channel Routers

ROUTER	Tracks	Vias	Wire length
LEA	31	290	6526
Dogleg router	21	346	5331
Y-K router	20	403	5381
Greedy router	20	329	5078
Hierarchical router	19	336	5023
YACR2	19	287	5020

Three-Layer Channel Routing Algorithms

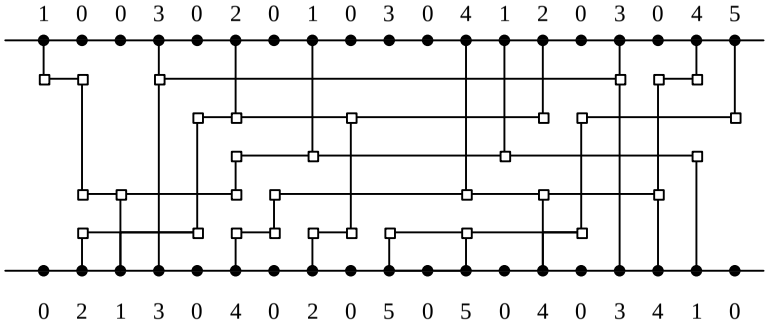
- Extended Net Merge Channel Router
- HVH Routing from HV Solution
- Hybrid HVH-VHV Router

HVH Routing from HV Solution

- Similar to YK algorithm.
- Composite nets in YK algorithm are combined to form supercomposite nets.
- Objective is to reduce the number of supercomposite nets.
- Two composite nets in a supercomposite net can be assigned to different layers on the same track.
- A track ordering graph is used.

J. Cong, D. F. Wong and C. L. Liu, ICCAD, 1987.

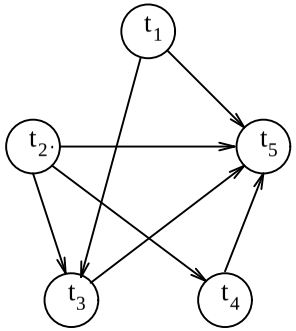
An example of HVH Routing from HV Solution



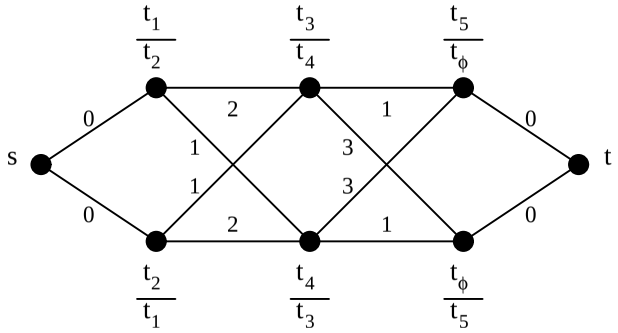
(a)

Time	P_1	P_2
1	t_1	t_2
2	t_3	t_4
3	t_5	

(c)

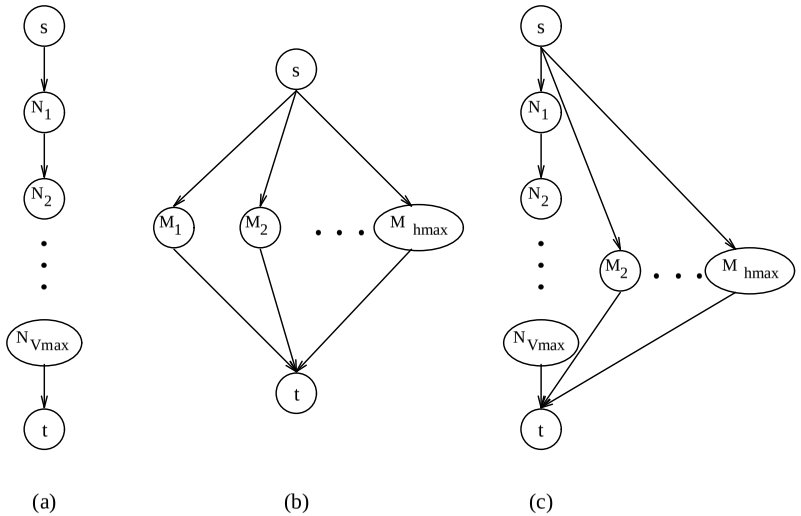


(b)

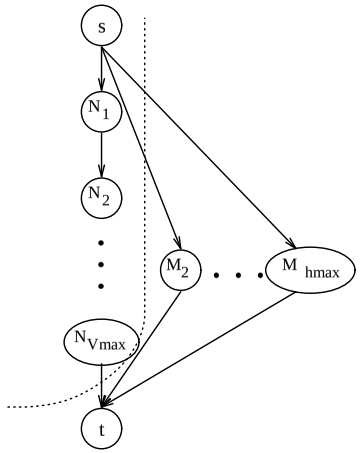


(d)

Limitations of HVH or VHV Router



Partitioning for hybrid routing

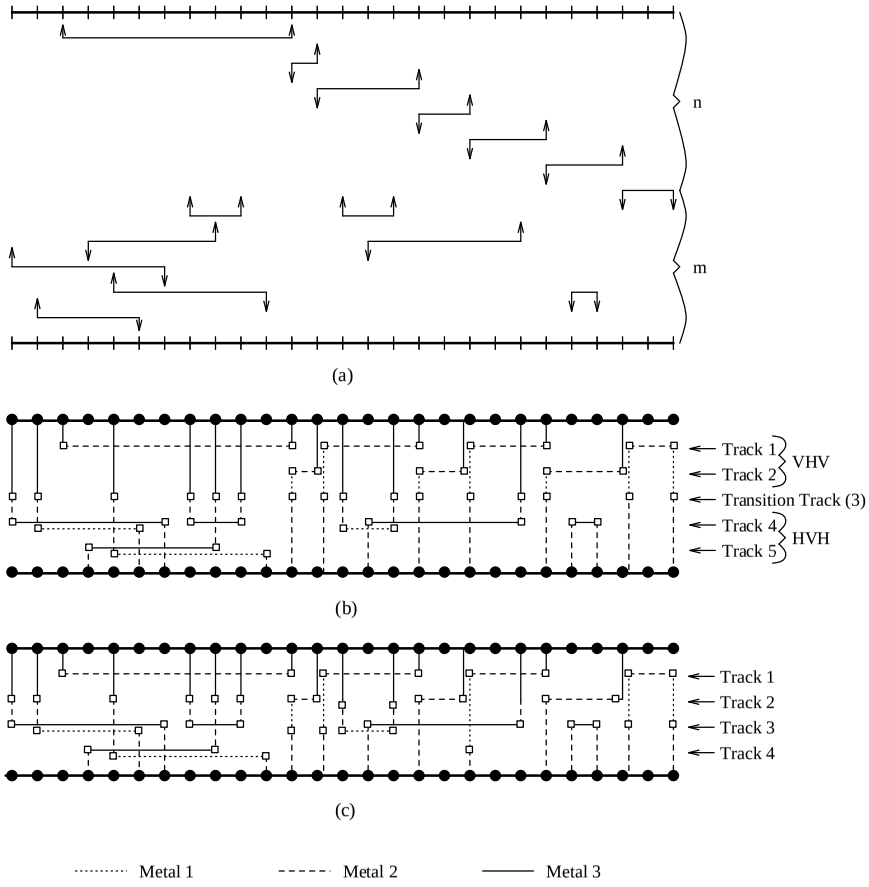


Hybrid HVH-VHV router

- Uses both HVH and VHV routing schemes.
- Pure HVH and VHV are special cases of Hybrid router.
- It partitions the channel into two portions - not necessarily of the same size.
- One portion is used for HVH and the other for VHV.
- One track is required for interconnection between the two portions.

V. Pitchumani and Q. Zhang, ICCAD, 1987.

Example of Hybrid Routing



Switchbox Routing Algorithms

- Greedy Router
- Rip-up and Re-route Based Router: MIGHTY
- Computational Geometry Based Router: BEAVER

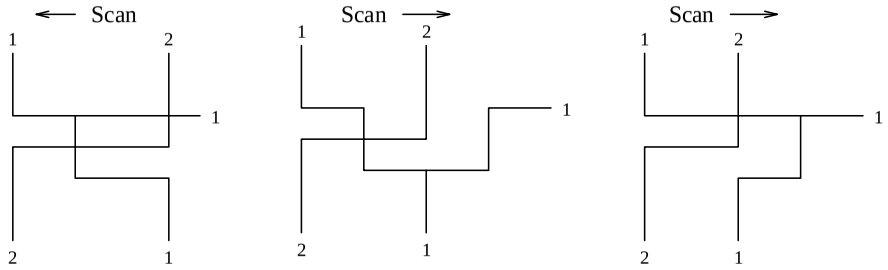
Greedy Switchbox Router

- Terminals on left boundary are brought to first column as horizontal tracks.
- Nets are jogged to target rows in addition to jogging to next top or bottom terminal.
- Nets are jogged based on the following priority scheme:
  1. A net whose right side of the target row and vertical track between the net and target row is empty.
  2. A net whose right side of the target row is empty and priority is based on how close a net can be jogged to target row.
  3. A net that can be brought closer to its target row.

W. K. Luk, Integration, The VLSI Journal, 1985.

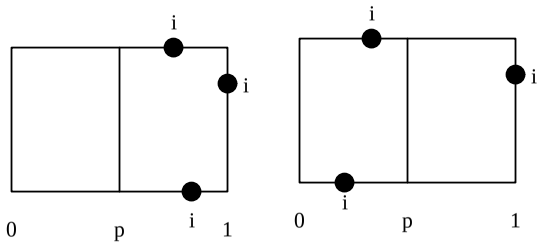
Routing Schemes

Routing schemes: (a) scheme 1,(b) scheme 2,(c) scheme 3



(a) (b) (c)

Routing scheme 4 (a) JOG_r for net N_i, (b) $JOG_{t/b}$ for net N_i



(a) (b)

Algorithm GREEDY-SB-ROUTER

```
Algorithm GREEDY-SB-ROUTER
begin
    Determine the scan direction;
    Bring left terminals into column 1;
    for i = 1 to M do
        if no empty track exists then
            increase number of tracks;
        bring T(i) and B(i) to empty tracks;
        join split nets as much as possible;
        for each net with no right terminals do
            bring split nets closer by jogging;
        for each net with a right terminal do
            use scheme 4;
        if close to right edge then
            fanout to all target rows;
    while split net exist do
        join split nets as much as possible;
end.
```

Algorithm MIGHTY

Algorithm MIGHTY
begin
1. Extend all pins on the boundaries of the region inside
   by one unit;

$L \rightarrow \phi$; (* Initialize list *)

2. (* path finder *)

for each net **do**
   MAZE-ROUTE(net, L);

3. sort L in increasing value of costs;
4. **while** $L \neq \phi$ **do**
   Get next path p from L;
   if no grid cell in p is occupied **then**
      Implement p; goto step 5;
      else invoke the path-finder to find a new feasible
         minimum path connecting two unconnected
            subnets of the net;
         Let δ be the increase in cost for the new path p';

Algorithm MIGHTY

if $\delta < MAXINCREASE$ **then**
    Implement p'; goto step 5;
    (* weak modification *)
    Push implemented nets around to
    obtain a 'good' connection for the given net;
    if weak modification fails **then**
        (* strong modification *)
        Remove an existing connection and
        try to obtain 'good' connection;
5. Remove p from L;

end.

Computational Geometry based router

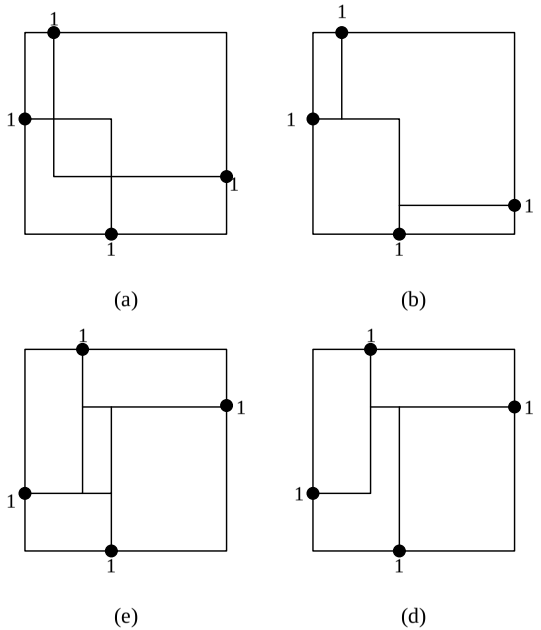
- Based on a delayed layering scheme with computational geometry techniques.
- Main objectives are via and wire length minimization.
- Its an unreserved layer model routing algorithm.
- Uses a priority queue to determine the order in which nets are interconnected.
- Uses 3 methods to find interconnections for nets:
  1. Corner router
  2. Line sweep router
  3. Thread router

J. P. Cohoon and P. L. Heck, IEEE-TCAD, 1988.

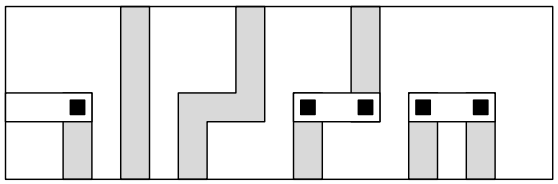
Algorithm BEAVER

```
Algorithm BEAVER
begin
    Initialize control information;
    Initialize corner-priority-queue;
    corner route;
    if there are nets to be routed then
        Initialize line-sweep-priority-queue;
        Line sweep route;
        if there are nets to be routed then
            Relax control constraints;
            Reinitialize line-sweep-priority-queue;
            Line sweep route;
            if there are nets to be routed then
                Initialize thread-priority-queue;
                Thread route;
    Perform layer assignment;
end.
```

Example of Overlap Cycles and Their Solutions



Prototype Linesweep Connections



Summary

1. The detailed routing problem is solved by routing the channels and switchboxes.
2. Routing results may differ based on the selection of routing models. A routing model can be grid-based or based on the layer assignments of different net segments.
3. The objectives for routing a channel is to minimize channel density, the length of routing nets, and the number of vias.
4. The main objective of (channel) routing is to minimize the total routing area.
5. The objective of switchbox routing is to determine the routability.