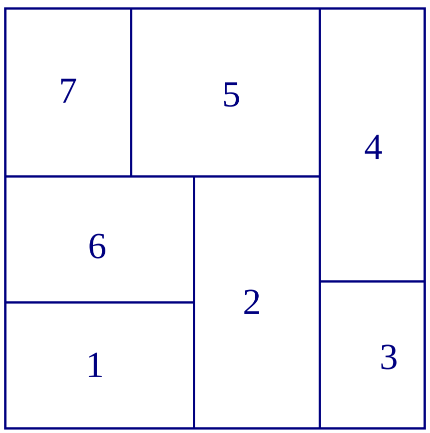
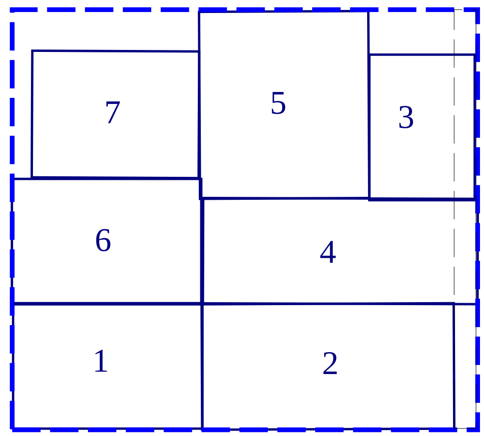
Floorplanning

- Inputs to the floorplanning problem:
  - A set of blocks, fixed or flexible.
  - Pin locations of fixed blocks.
  - A netlist.
- Objectives: **Minimize area**, **reduce wirelength** for (critical) nets, **maximize routability**, determine shapes of flexible blocks

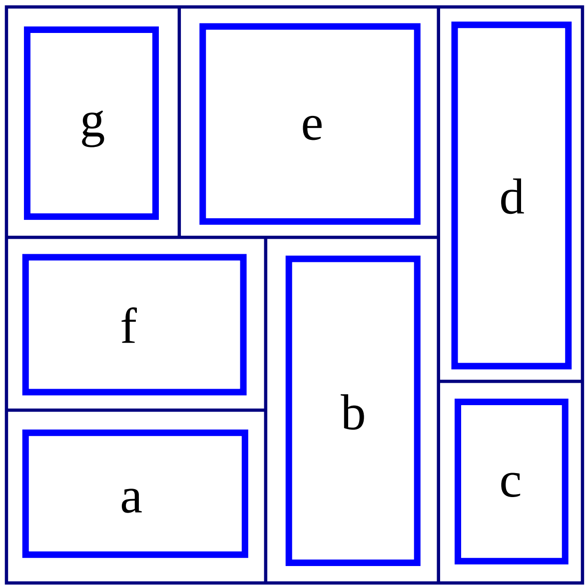


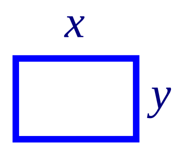
An optimal floorplan,
in terms of area

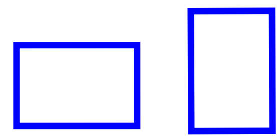


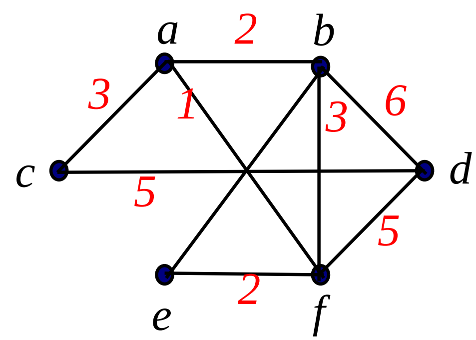
A non−optimal floorplan

Floorplan Design



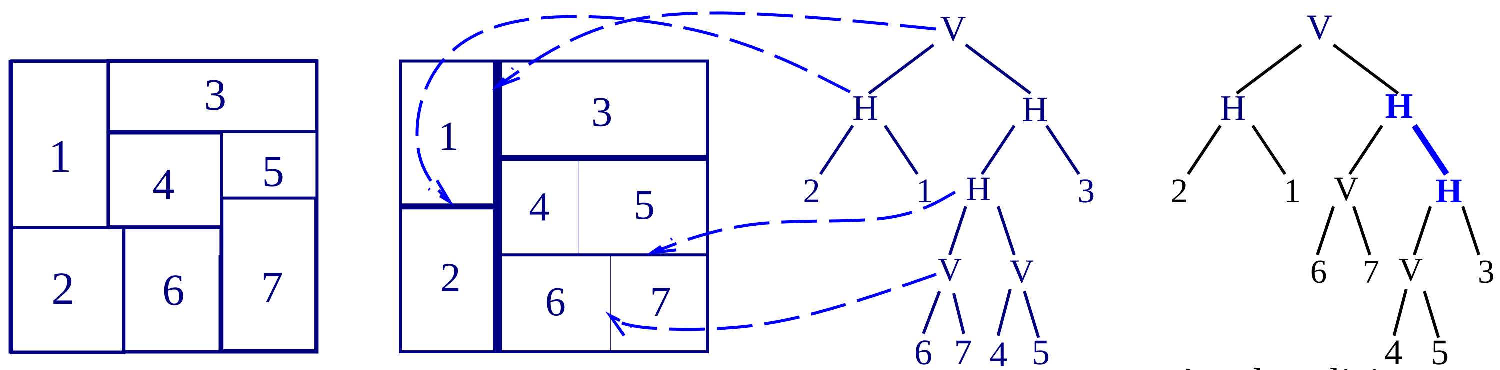
- *Modules:* 

- *Area: A=xy*
- *Aspect ratio: r <= y/x <= s*
- *Rotation:* 
- *Module connectivity*



Floorplanning: Terminology

- **Rectangular dissection:** Subdivision of a given rectangle by a finite # of horizontal and vertical line segments into a finite # of non-overlapping rectangles.
- **Slicing structure:** a rectangular dissection that can be obtained by repetitively subdividing rectangles horizontally or vertically.
- **Slicing tree:** A binary tree, where each internal node represents a vertical cut line or horizontal cut line, and each leaf a basic rectangle.
- **Skewed slicing tree:** One in which no node and its **right** child are the same.



Non−slicing floorplan | Slicing floorplan | A slicing tree (skewed) | Another slicing tree (non−skewed)

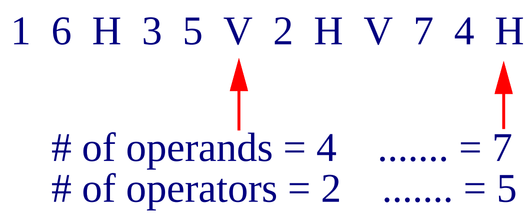
Floorplan Design by Simulated Annealing

- Related work
  - Wong & Liu, "A new algorithm for floorplan design," DAC'86.
    - Consider slicing floorplans.
  - Wong & Liu, "Floorplan design for rectangular and L-shaped modules," ICCAD'87.
    - Also consider L-shaped modules.
  - Wong, Leong, Liu, Simulated Annealing for VLSI Design, pp. 31–71, Kluwer academic Publishers, 1988.
- Ingredients: solution space, neighborhood structure, cost function, annealing schedule?

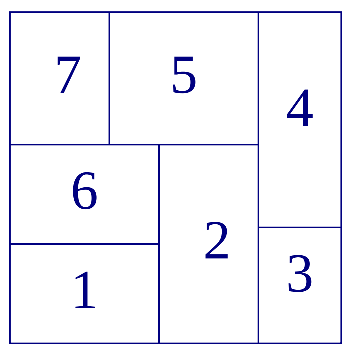
Solution Representation

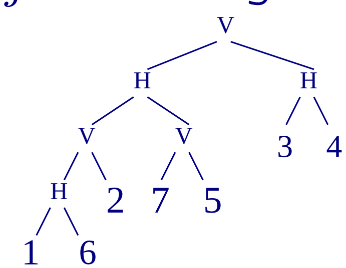
- An expression $E = e_1 e_2 \ldots e_{2n-1}$, where $e_i \in \{1, 2, \ldots, n, H, V\}, 1 \leq i \leq 2n-1$, is a **Polish expression** of length $2n-1$ iff

  1. every operand j, $1 \leq j \leq n$, appears exactly once in E;
  2. (**the balloting property**) for every subexpression $E_i = e_1 \ldots e_i, 1 \leq i \leq 2n-1$, #operands > #operators.



- Polish expression $\longleftrightarrow$ Postorder traversal.
- ijH: rectangle i on bottom of j; ijV: rectangle i on the left of j.



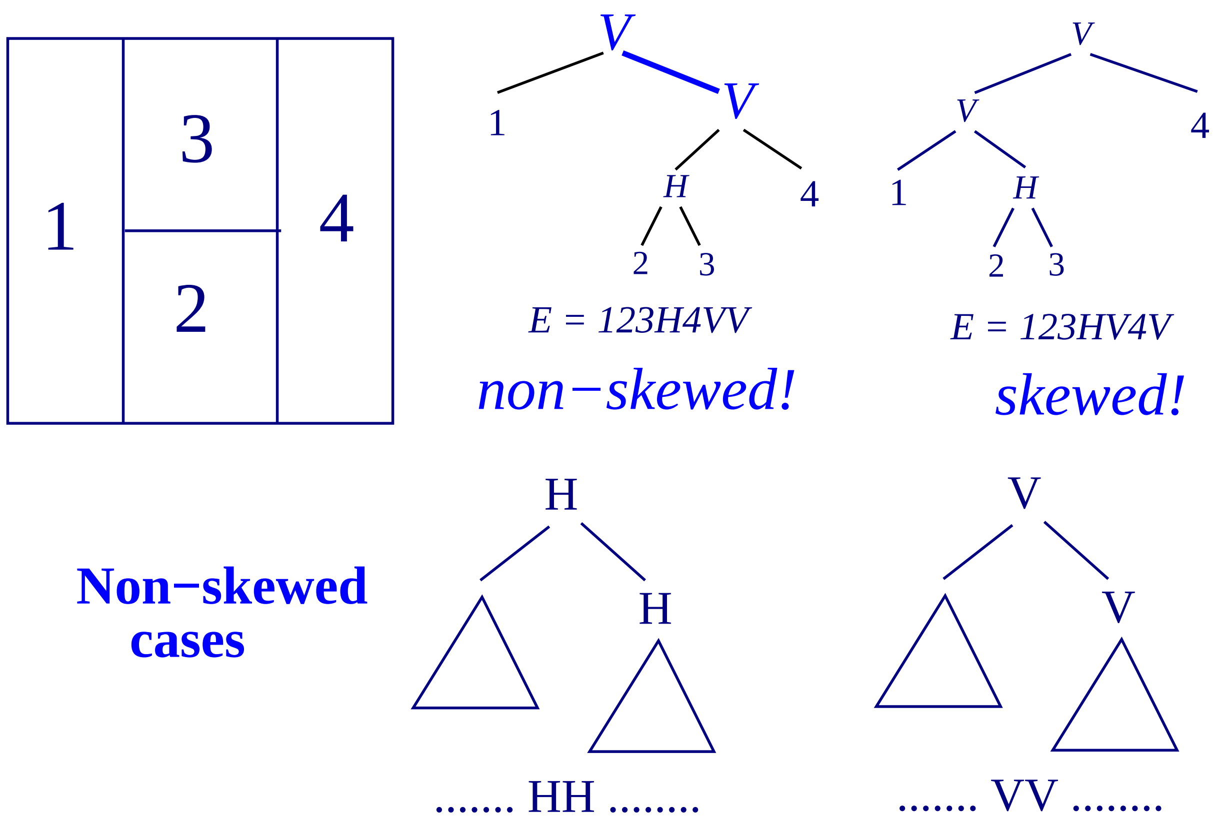


E = 16H2V75VH34HV

*E = 16+2*75*+34+**

Postorder traversal of a tree!

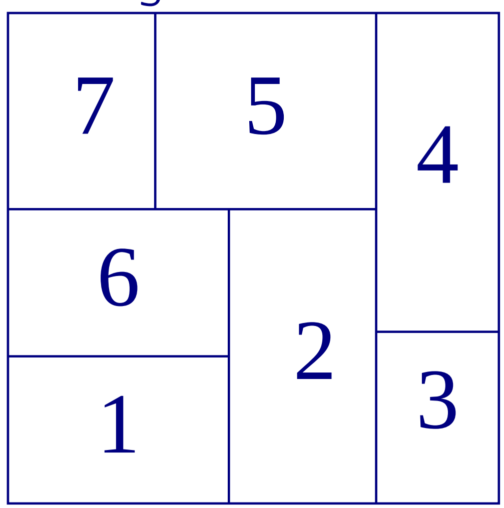
Solution Representation (cont'd)

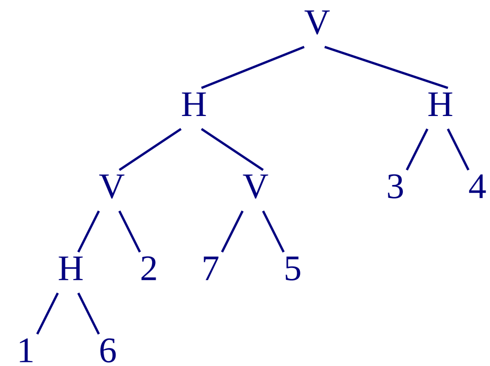


- **Question:** How to eliminate ambiguous representation?

Normalized Polish Expression

- A Polish expression $E = e_1 e_2 \ldots e_{2n-1}$ is called **normalized** iff E has no consecutive operators of the same type (H or V).
- Given a **normalized** Polish expression, we can construct a **unique** rectangular slicing structure.



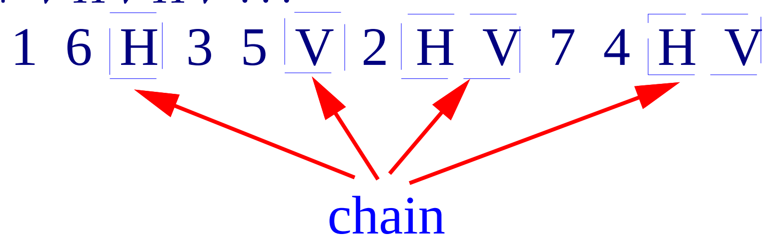


E = 16H2V75VH34HV

A normalized Polish expression

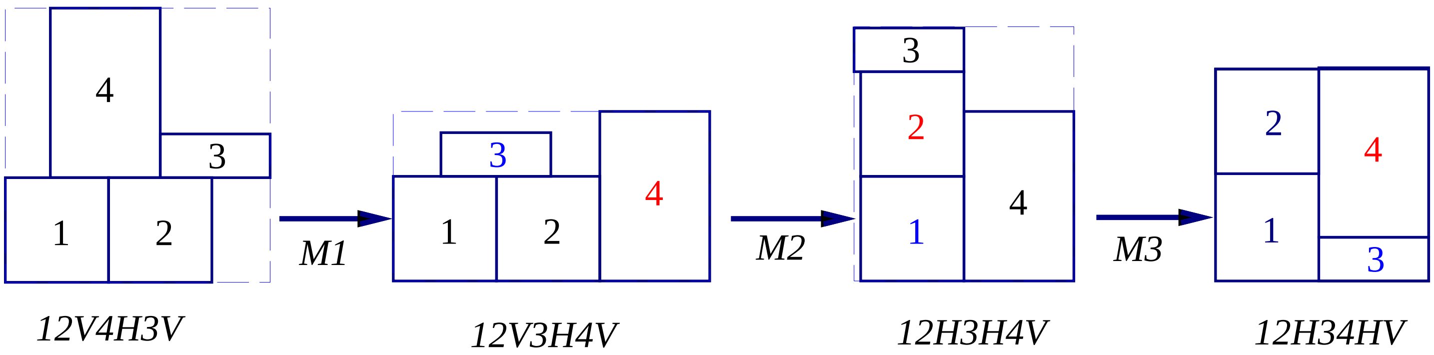
Neighborhood Structure

- **Chain:** $HVHVH\ldots$ or $VHVHV\ldots$



- **Adjacent:** 1 and 6 are adjacent operands; 2 and 7 are adjacent operands; 5 and V are adjacent operand and operator.
- 3 types of moves:
  - $M1$ **(Operand Swap)**: Swap two adjacent operands.
  - $M2$ **(Chain Invert)**: Complement some chain ($\overline{V} = H, \overline{H} = V$).
  - $M3$ **(Operator/Operand Swap)**: Swap two adjacent operand and operator.

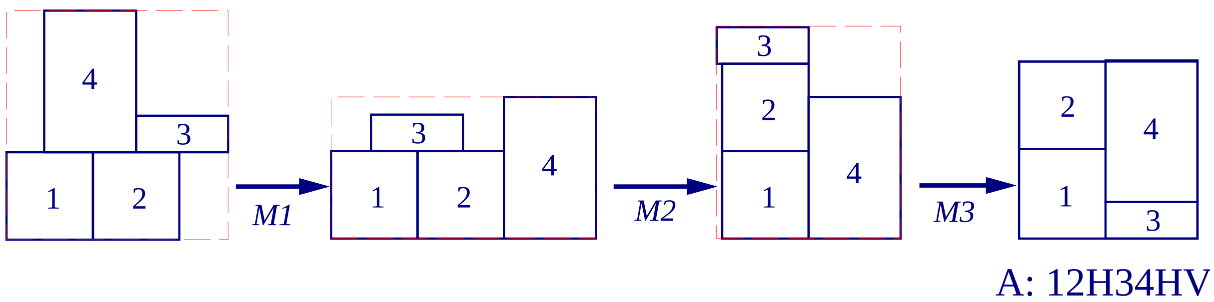
Effects of Perturbation



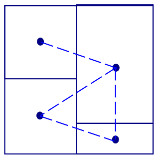
- **Question:** The balloting property holds during the moves?
  - $M1$ and $M2$ moves are OK.
  - **Check the $M3$ moves!** Reject "illegal" $M3$ moves.
- **Check $M3$ moves:** Assume that the M_3 move swaps the operand e_i with the operator e_{i+1}, $1 \leq i \leq k-1$. Then, the swap will not violate the balloting property iff $2N_{i+1} < i$.
  - N_k: # of operators in the Polish expression $E = e_1 e_2 \ldots e_k, 1 \leq k \leq 2n-1$.

Cost Function

- $\Phi = A + \lambda W$.
  - A: area of the smallest rectangle
  - W: overall wiring length
  - λ: user-specified parameter

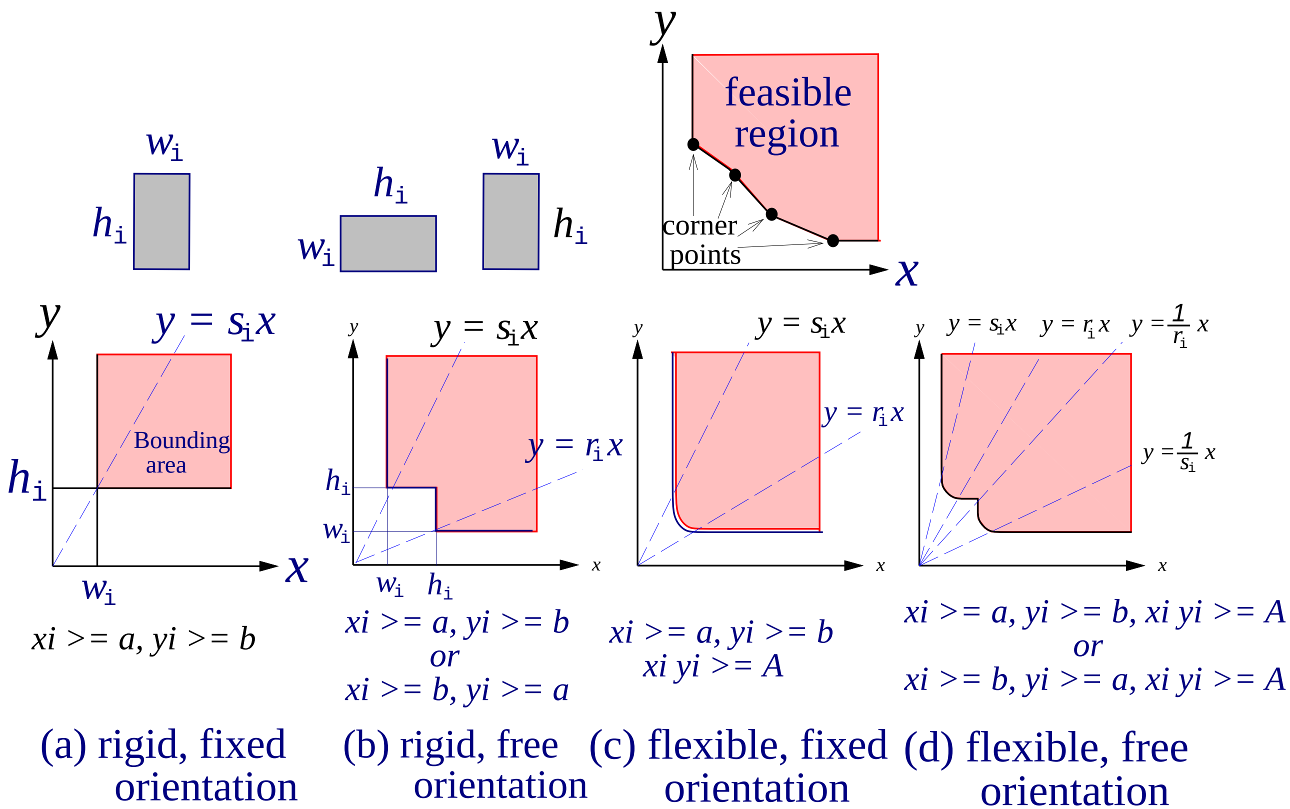


- $W = \sum_{ij} c_{ij} d_{ij}$.
  - c_{ij}: # of connections between blocks i and j.
  - d_{ij}: center-to-center distance between basic rectangles i and j.

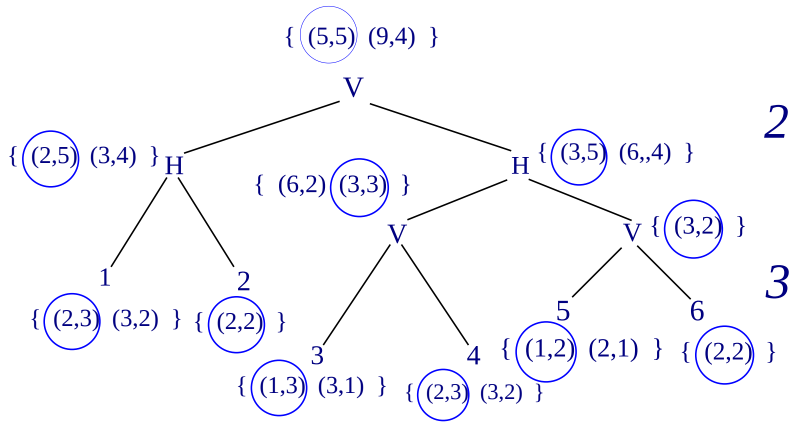


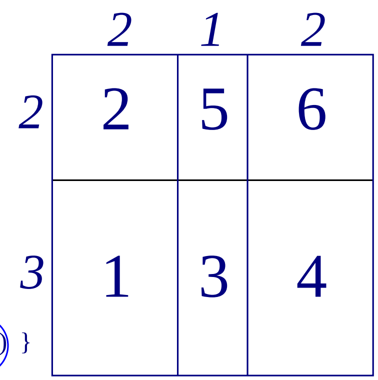
Cost Evaluation: Shape Curves

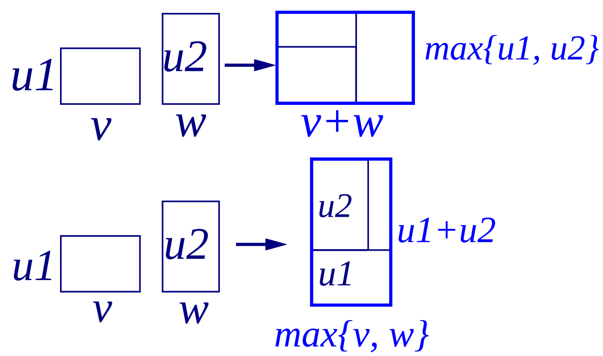
- Shape curves correspond to different kinds of constraints where the shaded areas are feasible regions.

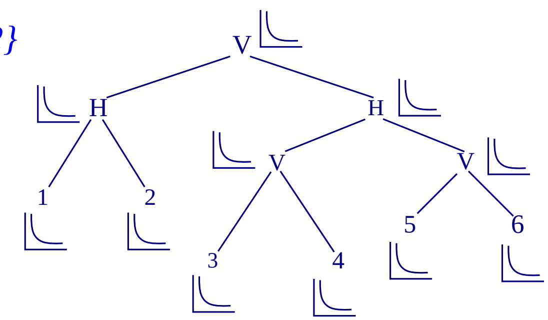


Area Computation





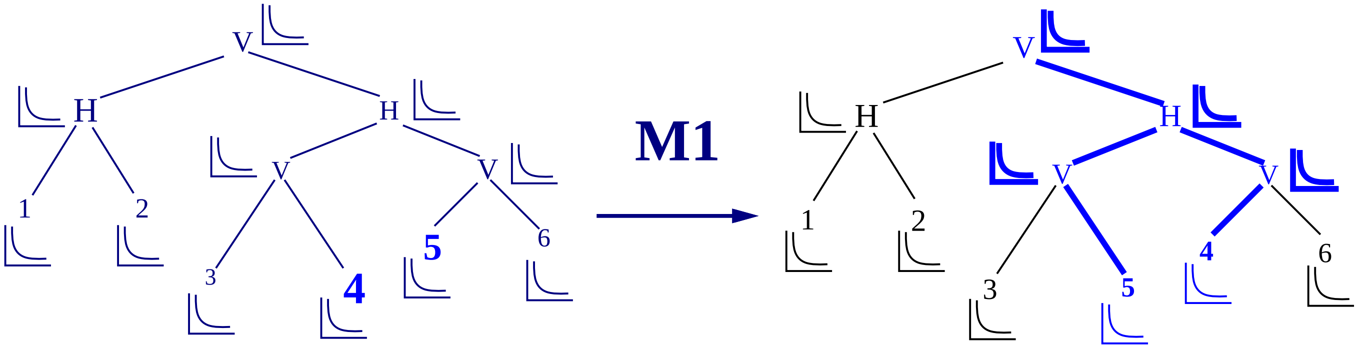




- Wiring cost?

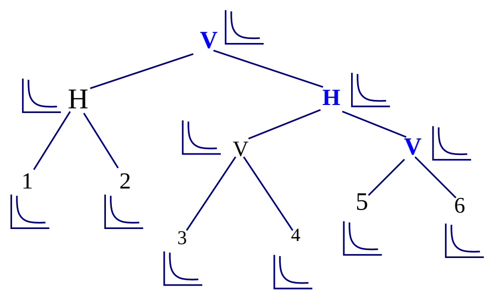
Incremental Computation of Cost Function

- Each move leads to only a minor modification of the Polish expression.
- At most **two paths** of the slicing tree need to be updated for each move.

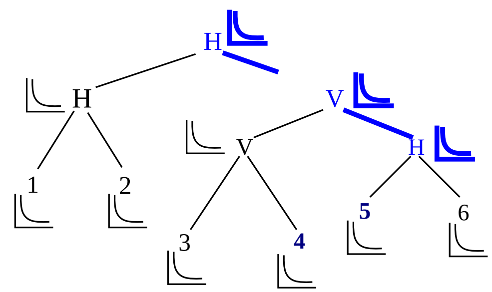


E = 12H34V56VHV **E = 12H35V46VHV**

Incremental Computation of Cost Function (cont'd)

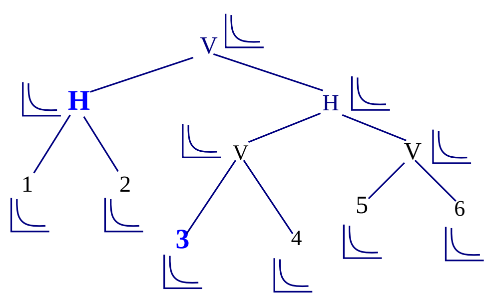


M2

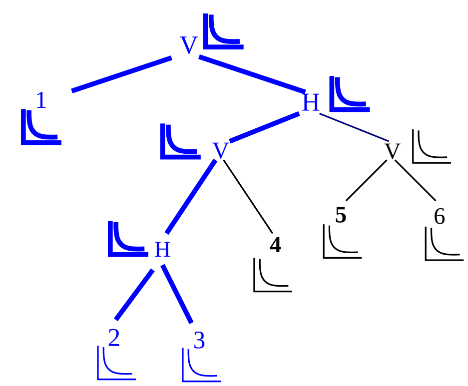


E = 12H34V56VHV

E = 12H34V56HVH



M3

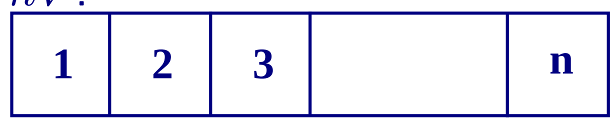


E = 12H34V56VHV

E = 123H4V56VHV

Annealing Schedule

- Initial solution: $12V3V \ldots nV$.



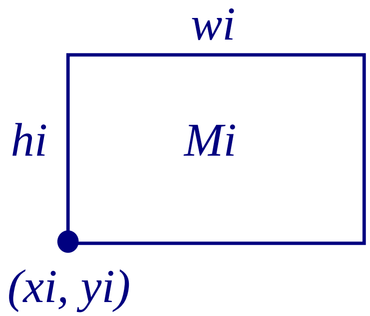
- $T_i = r^i T_0, i = 1, 2, 3, \ldots;\ r = 0.85$.
- At each temperature, try kn moves (k = 5–10).
- Terminate the annealing process if
  - # of accepted moves $<$ 5%,
  - temperature is low enough, or
  - run out of time.

Algorithm: Simulated_Annealing_Floorplanning(P, ϵ, r, k)

begin
$E \leftarrow 12V3V4V \ldots nV$; `/* initial solution */`
$Best \leftarrow E$; $T_0 \leftarrow \frac{\Delta_{avg}}{ln(P)}$; $M \leftarrow MT \leftarrow uphill \leftarrow 0$; $N = kn$;
repeat
  $MT \leftarrow uphill \leftarrow reject \leftarrow 0$;
  repeat
    `SelectMove(`M`);`
    `Case` M `of`
    M_1: `Select two adjacent operands` e_i `and` e_j; $NE \leftarrow Swap(E, e_i, e_j)$;
    M_2: `Select a nonzero length chain` C; $NE \leftarrow Complement(E, C)$;
    M_3: $done \leftarrow FALSE$;
      while not $(done)$ **do**
        `Select two adjacent operand` e_i `and operator` e_{i+1};
        if $(e_{i-1} \neq e_{i+1})$ **and** $(2N_{i+1} < i)$ **then** $done \leftarrow TRUE$;
      $NE \leftarrow Swap(E, e_i, e_{i+1})$;
    $MT \leftarrow MT + 1$; $\Delta cost \leftarrow cost(NE) - cost(E)$;
    if $(\Delta cost \leq 0)$ **or** $(Random < e^{\frac{-\Delta cost}{T}})$
    then
      if $(\Delta cost > 0)$ **then** $uphill \leftarrow uphill + 1$;
      $E \leftarrow NE$;
      if $cost(E) < cost(best)$ **then** $best \leftarrow E$;
    else $reject \leftarrow reject + 1$;
  until $(uphill > N)$ **or** $(MT > 2N)$;
  $T = rT$; `/* reduce temperature */`
  until $(\frac{reject}{MT} > 0.95)$ **or** $(T < \epsilon)$ **or** $OutOfTime$;
  end

Floorplanning by Mathematical Programming

- Sutanthavibul, Shragowitz, and Rosen, "An analytical approach to floorplan design and optimization," 27th DAC, 1990.
- Notation:
  - w_i, h_i: width and height of module M_i.
  - (x_i, y_i): coordinate of the lower left corner of module M_i.
  - $a_i \leq w_i/h_i \leq b_i$: aspect ratio w_i/h_i of module M_i. (Note: We defined aspect ratio as h_i/w_i before.)
- Goal: Find a mixed **integer linear programming (ILP)** formulation for the floorplan design.
  - **Linear** constraints? Objective function?



*Area = hi * wi*

Aspect ratio = wi / hi

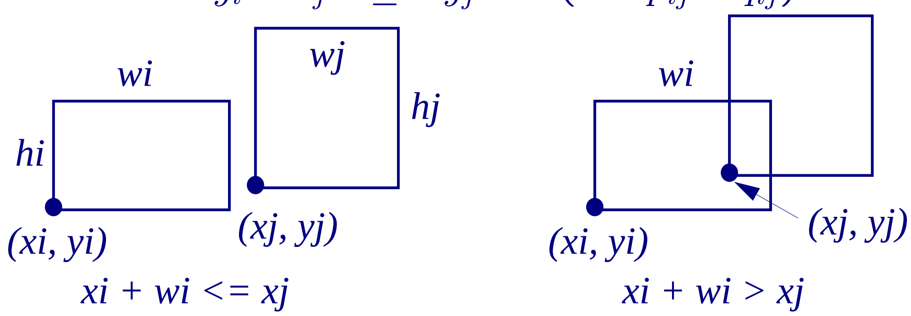
Nonoverlap Constraints

- Two modules M_i and M_j are nonoverlap, if at least one of the following linear constraints is satisfied (cases encoded by p_{ij} and q_{ij}):

		p_{ij}	q_{ij}
M_i to the left of M_j:	$x_i + w_i \leq x_j$	0	0
M_i below M_j:	$y_i + h_i \leq y_j$	0	1
M_i to the right of M_j:	$x_i - w_j \geq x_j$	1	0
M_i above M_j:	$y_i - h_j \geq y_j$	1	1

- Let W, H be upper bounds on the floorplan width and height, respectively.
- Introduce two 0, 1 variables p_{ij} and q_{ij} to denote that one of the above inequalities is enforced; e.g., $p_{ij} = 0, q_{ij} = 1 \Rightarrow y_i + h_i \leq y_j$ is satisfied.

$$\begin{aligned}
x_i + w_i &\leq x_j + W(p_{ij} + q_{ij}) \\
y_i + h_i &\leq y_j + H(1 + p_{ij} - q_{ij}) \\
x_i - w_j &\geq x_j - W(1 - p_{ij} + q_{ij}) \\
y_i - h_j &\geq y_j - H(2 - p_{ij} - q_{ij})
\end{aligned}$$



Cost Function & Constraints

- Minimize $Area = xy$, **nonlinear!** (x, y: width and height of the resulting floorplan)
- How to fix?
  - Fix the width W and minimize the height y!
- Four types of constraints:
  1. no two modules overlap ($\forall i, j : 1 \leq i < j \leq n$);
  2. each module is enclosed within a rectangle of width W and height H ($x_i + w_i \leq W, y_i + h_i \leq H, 1 \leq i \leq n$);
  3. $x_i \geq 0$, $y_i \geq 0, 1 \leq i \leq n$;
  4. $p_{ij}, q_{ij} \in \{0, 1\}$.
- w_i, h_i are known.

Mixed ILP for Floorplanning

Mixed ILP for the floorplanning problem with rigid, fixed modules.

$$\min \quad y$$

subject to

$$
\begin{aligned}
x_i + w_i &\leq W, & 1 \leq i \leq n & \quad (1)\\
y_i + h_i &\leq y, & 1 \leq i \leq n & \quad (2)\\
x_i + w_i &\leq x_j + W(p_{ij} + q_{ij}), & 1 \leq i < j \leq n & \quad (3)\\
y_i + h_i &\leq y_j + H(1 + p_{ij} - q_{ij}), & 1 \leq i < j \leq n & \quad (4)\\
x_i - w_j &\geq x_j - W(1 - p_{ij} + q_{ij}), & 1 \leq i < j \leq n & \quad (5)\\
y_i - h_j &\geq y_j - H(2 - p_{ij} - q_{ij}), & 1 \leq i < j \leq n & \quad (6)\\
x_i, y_i &\geq 0, & 1 \leq i \leq n & \quad (7)\\
p_{ij}, q_{ij} &\in \{0, 1\}, & 1 \leq i < j \leq n & \quad (8)
\end{aligned}
$$

- Size of the mixed ILP: for n modules,
  - # continuous variables: $O(n)$; # integer variables: $O(n^2)$; # linear constraints: $O(n^2)$.
  - Unacceptably huge program for a large n! (How to cope with it?)
- Popular LP software: LINDO, lp_solve, etc.

Mixed ILP for Floorplanning (cont'd)

Mixed ILP for the floorplanning problem: rigid, freely oriented modules.

$$
\begin{aligned}
&\text{min} \quad y \\
&\textit{subject to} \\
x_i + r_i h_i + (1 - r_i) w_i &\leq W, && 1 \leq i \leq n && (9) \\
y_i + r_i w_i + (1 - r_i) h_i &\leq y, && 1 \leq i \leq n && (10) \\
x_i + r_i h_i + (1 - r_i) w_i &\leq x_j + M(p_{ij} + q_{ij}), && 1 \leq i < j \leq n && (11) \\
y_i + r_i w_i - (1 - r_i) h_i &\leq y_j + M(1 + p_{ij} - q_{ij}), && 1 \leq i < j \leq n && (12) \\
x_i - r_j h_j + (1 - r_j) w_j &\geq x_j - M(1 - p_{ij} + q_{ij}), && 1 \leq i < j \leq n && (13) \\
y_i - r_j w_j - (1 - r_j) h_j &\geq y_j - M(2 - p_{ij} - q_{ij}), && 1 \leq i < j \leq n && (14) \\
x_i, y_i &\geq 0, && 1 \leq i \leq n && (15) \\
p_{ij}, q_{ij} &\in \{0, 1\}, && 1 \leq i < j \leq n && (16)
\end{aligned}
$$

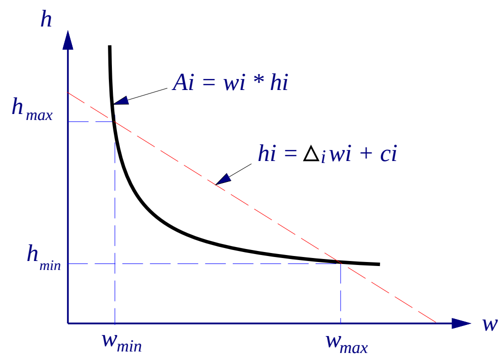
- For each module i with free orientation, associate a 0-1 variable r_i:
  - $r_i = 0$: 0° rotation for module i.
  - $r_i = 1$: 90° rotation for module i.
- $M = \max\{W, H\}$.

Flexible Modules

- Assumptions: w_i, h_i are unknown; area lower bound: A_i.
- Module size constraints: $w_i h_i \geq A_i$; $a_i \leq \frac{w_i}{h_i} \leq b_i$.
- Hence, $w_{min} = \sqrt{A_i a_i}$, $w_{max} = \sqrt{A_i b_i}$, $h_{min} = \sqrt{\frac{A_i}{b_i}}$, $h_{max} = \sqrt{\frac{A_i}{a_i}}$.
- $w_i h_i \geq A_i$ nonlinear! How to fix?
  - Can apply a first-order approximation of the equation: a line passing through (w_{min}, h_{max}) and (w_{max}, h_{min}).

$$
\begin{aligned}
h_i &= \Delta_i w_i + c_i && /*\ \ y = mx + c\ \ */ \\
\Delta_i &= \frac{h_{max} - h_{min}}{w_{min} - w_{max}} && /*\ \ slope\ \ */ \\
c_i &= h_{max} - \Delta_i w_{min} && /*\ \ c = y_0 - mx_0\ \ */
\end{aligned}
$$

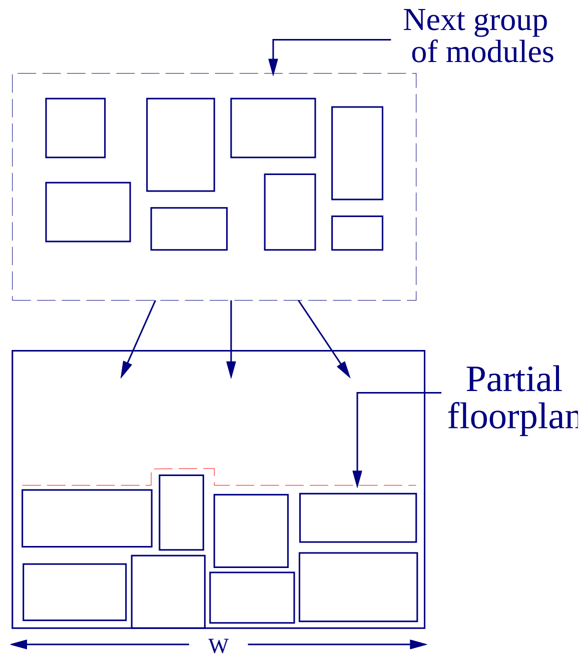
  - Substitute $\Delta_i w_i + c_i$ for h_i to form linear constraints (x_i, y_i, w_i are unknown; Δ_i, Δ_j, c_i, c_j can be computed as above).



h
Ai = wi * hi
h max
hi = △i wi + ci
h min
w
w min
w max

Reducing the Size of the Mixed ILP

- Time complexity of a mixed ILP: exponential!
- Recall the large size of the mixed ILP: # variables, # constraints: $O(n^2)$.
  - How to fix it?
- Key: Solve a partial problem at each step (successive augmentation)
- Questions:
  - How to select next subgroup of modules? $\Rightarrow$ linear ordering based on connectivity.
  - How to minimize the # of required variables?



Next group
of modules
Partial
floorplan
W

Reducing the Size of the Mixed ILP (cont'd)

- Size of each successive mixed ILP depends on (1) # of modules in the next group; (2) "size" of the partially constructed floorplan.
- Keys to deal with (2)
  - Minimize the problem size of the partial floorplan.
  - Replace the already placed modules by a set of covering rectangles.
  - # rectangles is usually much smaller than # placed modules.