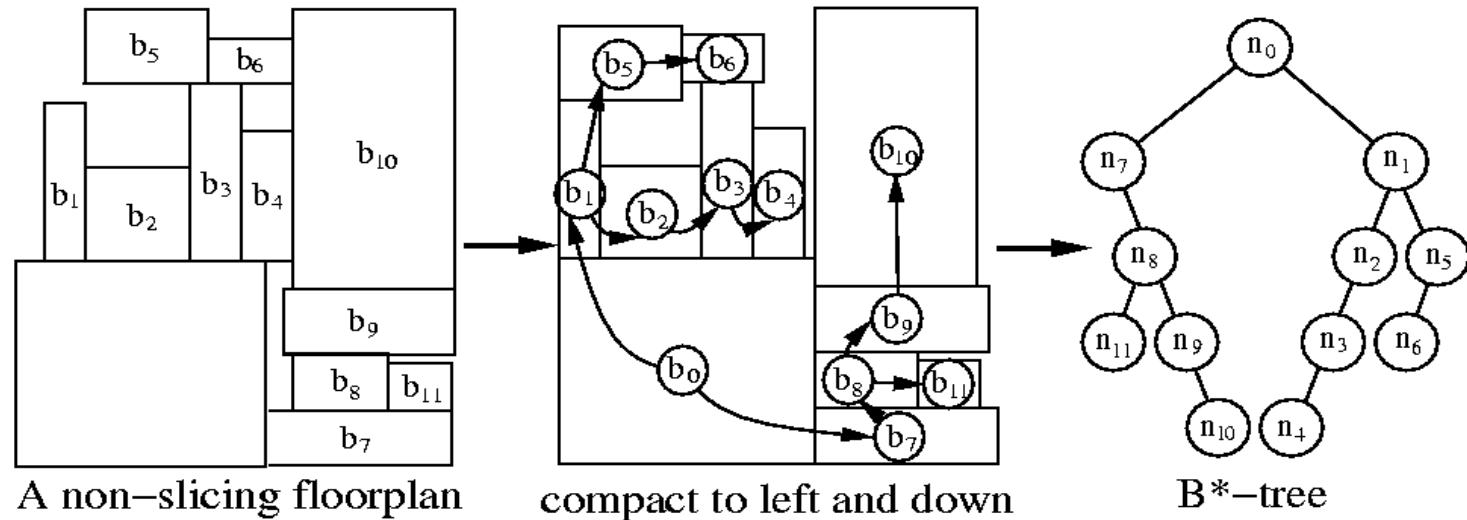


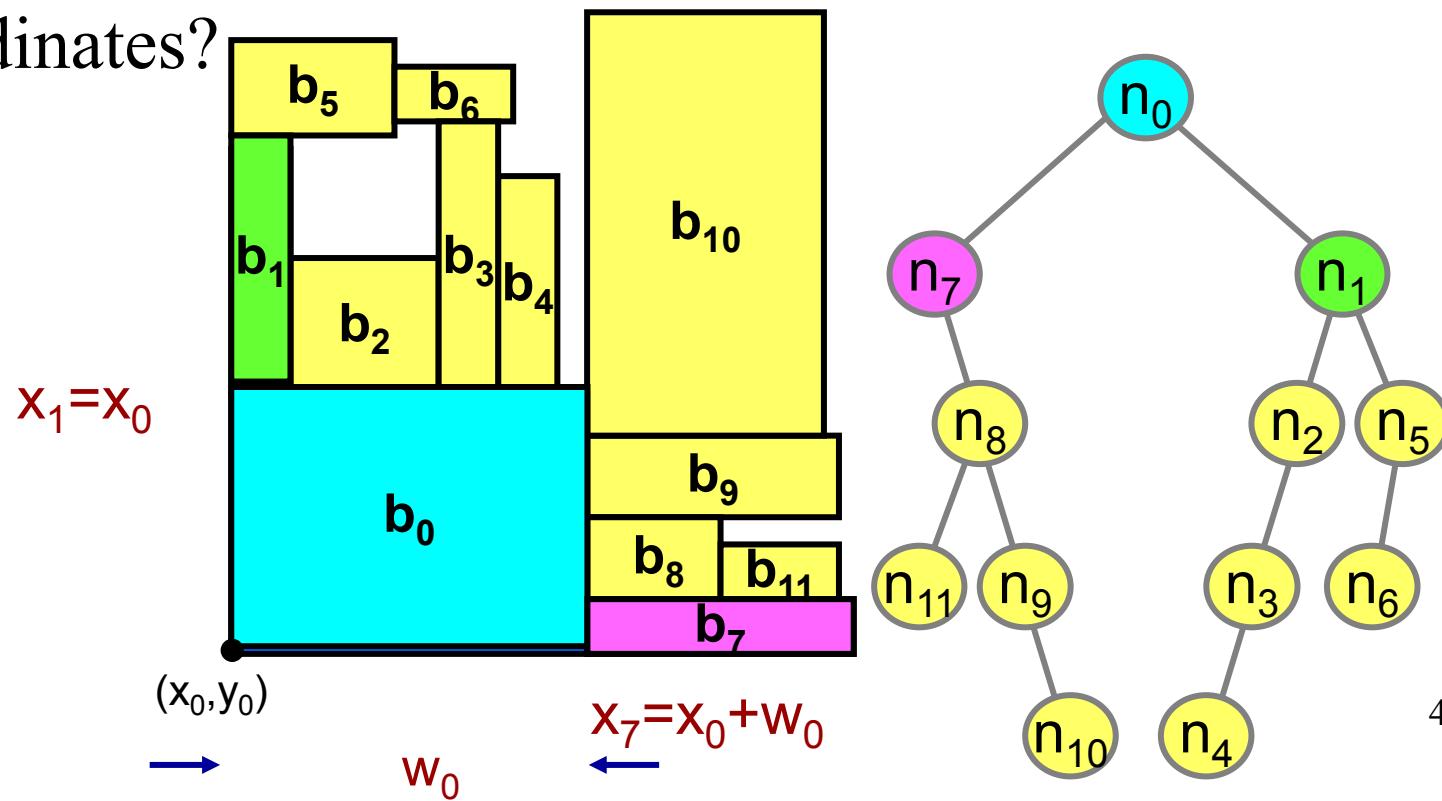
B*-Tree

- Chang, Chang, Wu, and Wu, “B*-tree: a new representation for non-slicing floorplans,” DAC’00.
- Ideas:
 - From an admissible placement to a B*-tree:
Left child: the lowest module on the right.
Right child: the module above, with the same left-side coordinate.



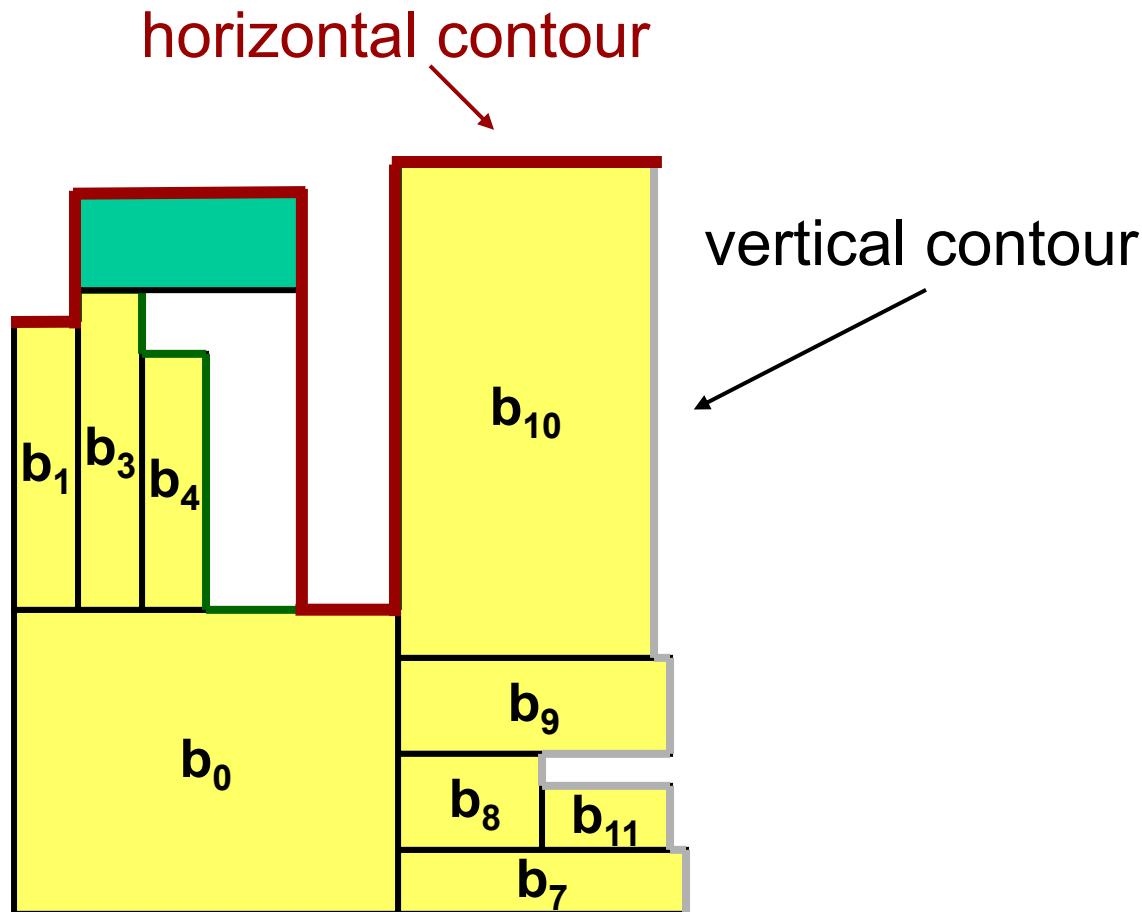
B*-tree Packing

- x-coordinates can be determined by the tree structure.
 - Left child: the lowest, adjacent block on the right ($x_j = x_i + w_i$).
 - Right child: the first block above, with the same x-coordinate ($x_j = x_i$).
- y-coordinates?



Computing y-coordinates

- Reduce the complexity of computing a y-coordinate to amortized O(1) time.

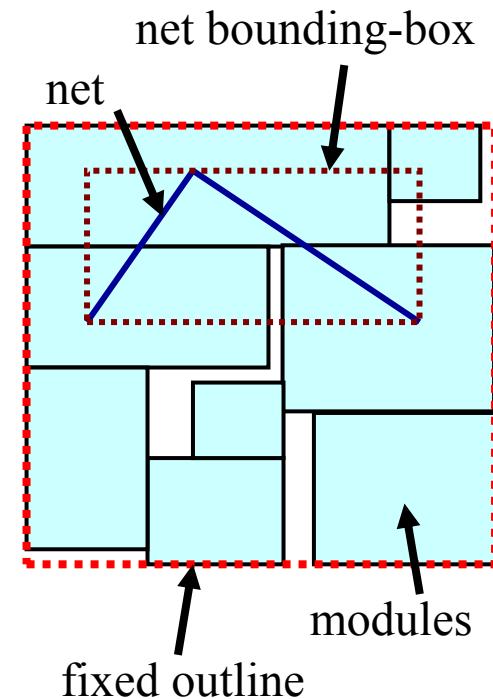


Perturbations & Solutions

- Perturbing B*-trees in simulated annealing
 - Op1: Rotate a module.
 - Op2: Flip a module.
 - Op3: Move a module to another place.
 - Op4: Swap two modules

Fixed-Outline Floorplanning

- Chen and Chang, “Modern floorplanning based on fast simulated annealing,” ISPD’05 & TCAD’06
- Input: modules, netlist, fixed outline
- Output: module positions, orientations
- Objectives
 - Minimize the half-perimeter wirelength (HPWL)
 - All modules are within the fixed die (fixed-outline constraint) and no overlaps occur between modules



Fixed-Outline Constraint

- Given the total area A of modules, the percentage Γ of dead space, and the desired aspect ratio R^* , the outline is defined by

$$H^* = \sqrt{(1+\Gamma)AR^*} \quad W^* = \sqrt{(1+\Gamma)A/R^*}$$

$$- R^* = H^*/W^*, H^*W^* = (1+\Gamma)A$$

- Cost for floorplan F

$$\Phi(F) = \alpha A + \beta L + (1 - \alpha - \beta)(R^* - R)^2$$

- A : floorplan area
- L : wirelength
- R^* : fixed-outline aspect ratio
- R : floorplan aspect ratio

Simulated Annealing Schedules

- Classical simulated annealing (SA)

- Non-zero probability for up-hill move:

$$p = e^{-\Delta C/T}$$

- Initial temperature: $T = |\Delta_{\text{avg}}| / \ln p$, p is the initial acceptance rate (typically, close to 1.0), Δ_{avg} is the average cost of the up-hill moves
 - Classical temperature updating function: λ is set to a fixed value (e.g., 0.85)

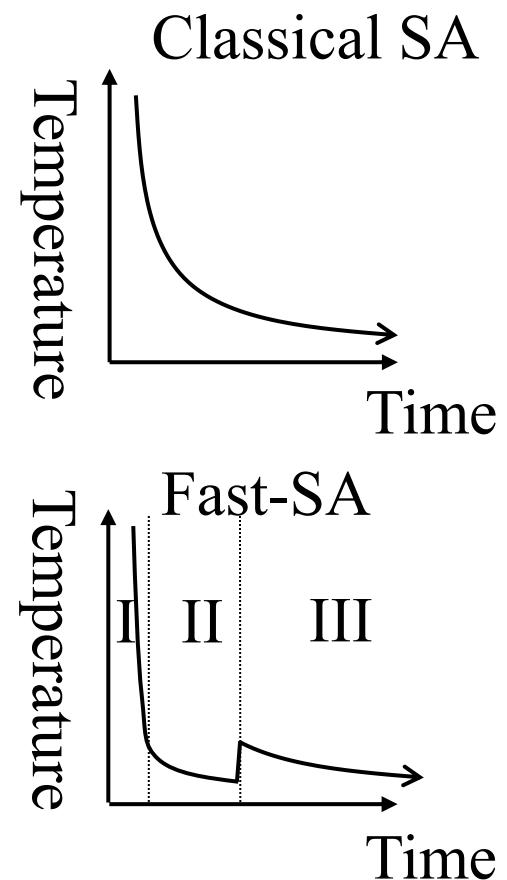
$$T_{\text{new}} = \lambda T_{\text{old}}, \quad 0 < \lambda < 1$$

- TimberWolf annealing schedule (Sechen and Sangiovanni-Vincentelli, DAC'86)

- Increase λ gradually from its lowest value (0.8) to its highest value (approximately 0.95) and then gradually decreases λ back to its lowest value.

Fast Simulated Annealing

- Fast Simulated Annealing (Fast-SA) consists of 3 stages
 - High-temperature random search (temperature $T \rightarrow$ a very large value)
 - Pseudo-greedy local search ($T \rightarrow 0$)
 - Hill-climbing search (increase T to simulate regular SA)



Fast Simulated Annealing (cont'd)

- Temperature update (T_1 : initial temperature)

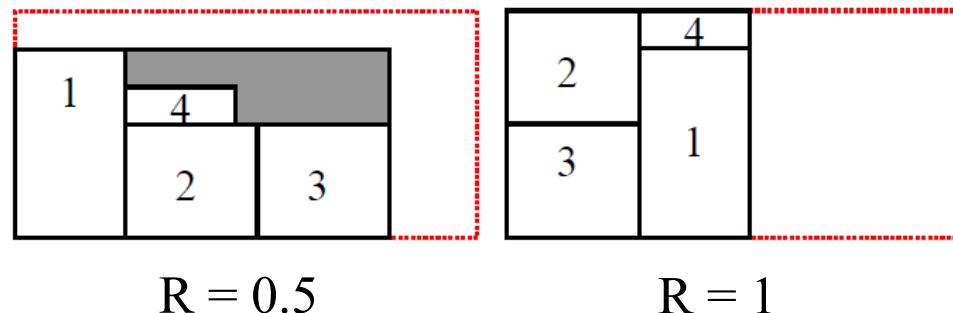
$$T_r = \begin{cases} \frac{\Delta_{avg}}{\ln P} & r = 1 \\ \frac{T_1 \langle \Delta_{cost} \rangle}{rc} & 2 \leq r \leq k \\ \frac{T_1 \langle \Delta_{cost} \rangle}{r} & r > k \end{cases}$$

Δ_{avg} Average uphill cost
 P Initial acceptance rate
 $\langle \Delta_{cost} \rangle$ Average cost change for current temperature
 r Number of iterations
 c, k User-specified parameters
(e.g., $c = 100, k = 7$)

- If $\langle \Delta_{cost} \rangle$ is larger, temperature decreases slowly.
- If $\langle \Delta_{cost} \rangle$ is smaller, temperature decreases quickly.

Adaptive Fast-SA

- The aspect ratio of the best floorplan area in the fixed outline is not the same as that of the outline.
- Decrease the weight of aspect ratio penalty ($1 - \alpha - \beta$) to concentrate more on the floorplan wirelength/area optimization (i.e., increase α and β).
 - Adopt an adaptive method to control the weights in the cost function based on n most recent floorplans.
 - The more feasible floorplans, the less aspect ratio penalty.

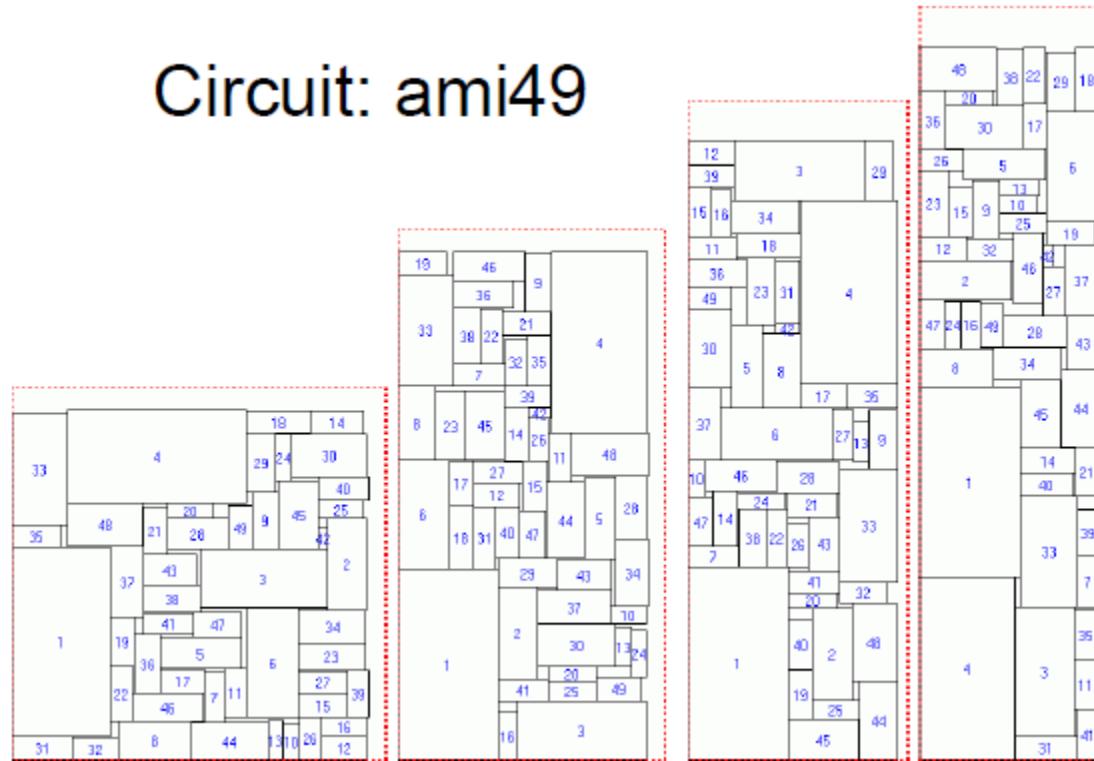


fixed-outline aspect
ratio $R^* = 0.5$

B*-tree Fixed-Outline Floorplanning Results

- B*-tree representation
- Fixed-outline floorplans with 10% dead space and aspect ratios 1, 2, 3, and 4.

Circuit: ami49



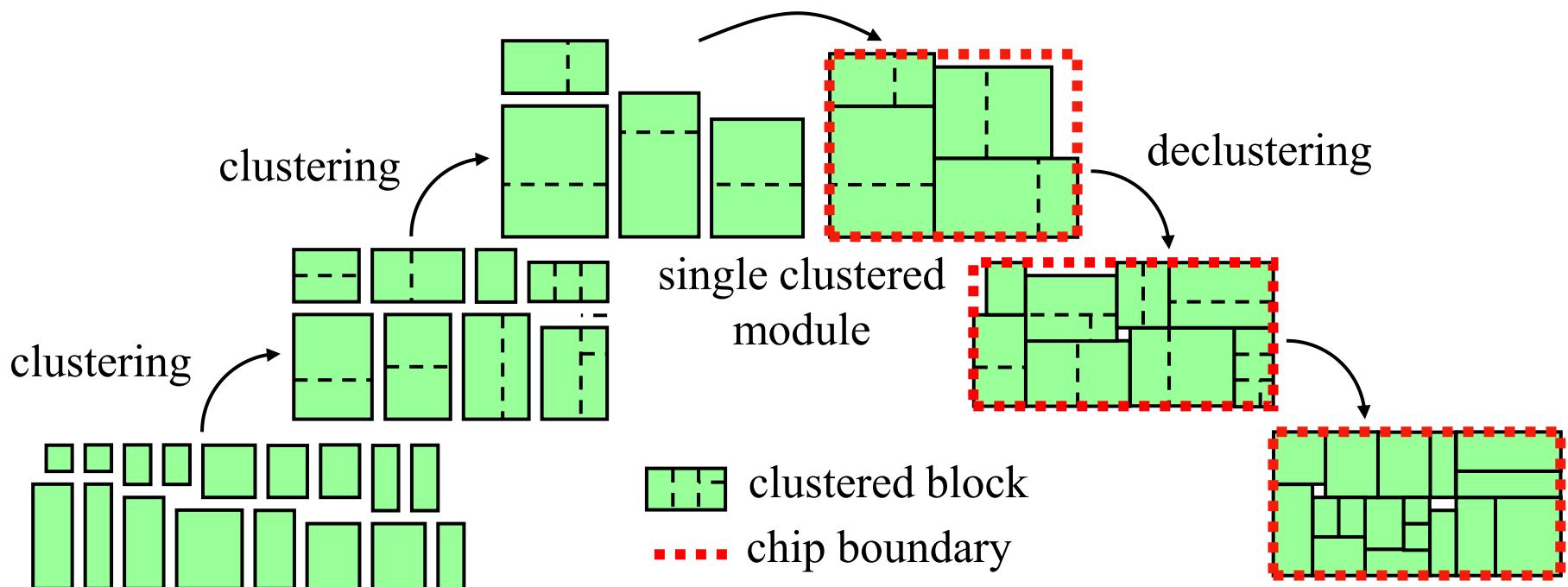
Floorplanning for Large-Scale Circuits

- Lee, Hsu, Chang, Yang, “Multilevel floorplanning/placement for large-scale modules using B*-trees,” DAC’03 & TCAD’07
- Clustering (bottom-up coarsening) + declustering (top-down uncoarsening
 - Clustering
 - Iteratively groups a set of modules based on area utilization and module connectivity
 - Constructs a B*-tree to keep the geometric relations for the newly clustered modules
 - Declustering
 - Iteratively ungroups a set of the previously clustered modules (i.e., perform tree expansion)
 - Refines the solution using simulated annealing

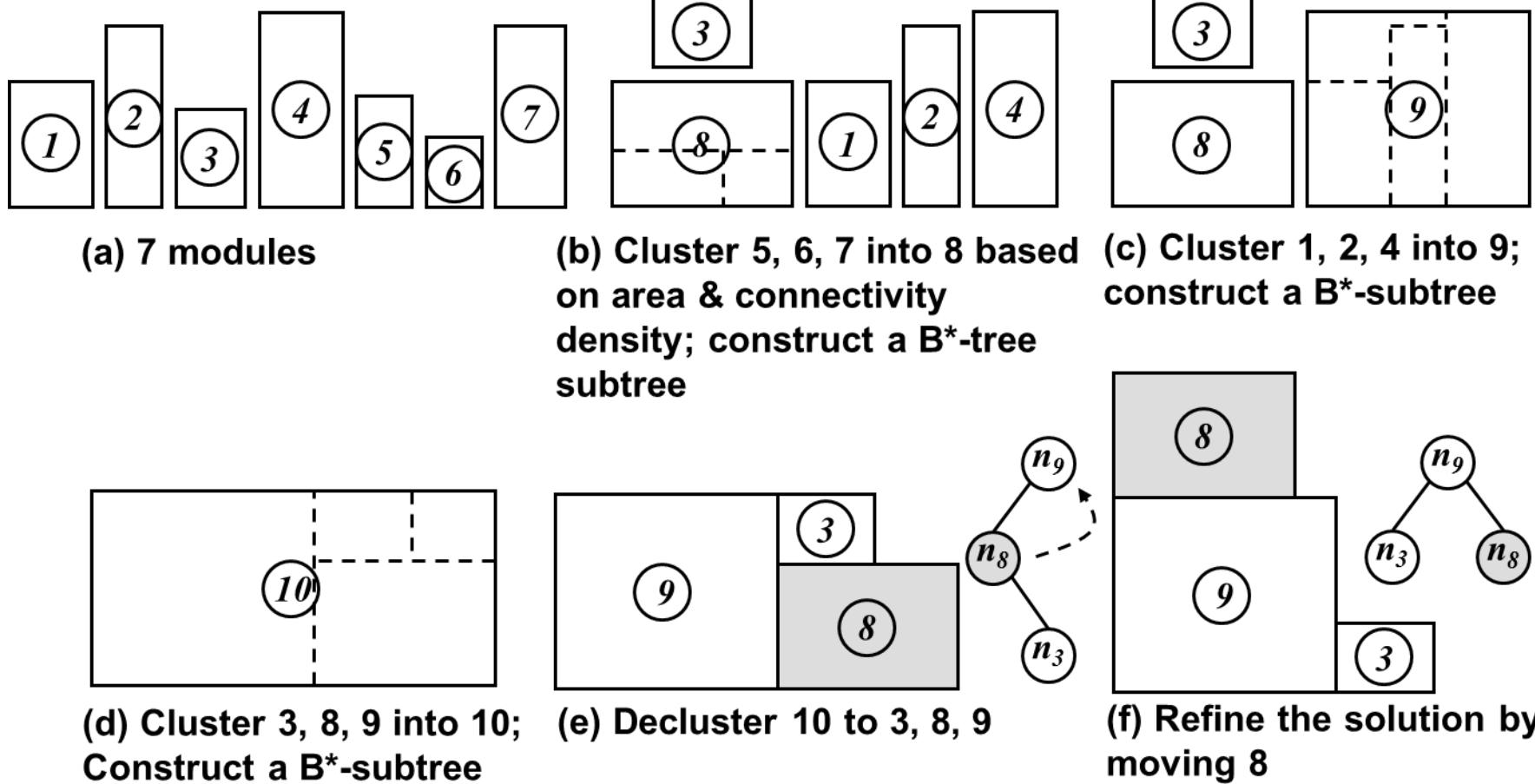
Λ -Shaped Multilevel Floorplanning

Cluster the modules based on area and local connectivity and create clustered modules for the next level.

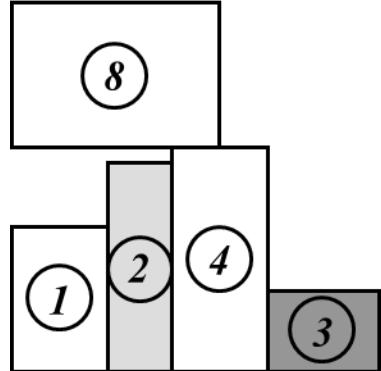
Recursively decluster the clusters and use simulated annealing to refine the floorplan.



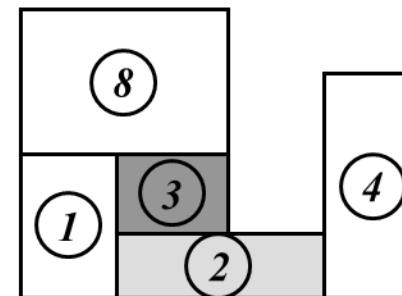
MB*-tree: Multilevel B*-tree



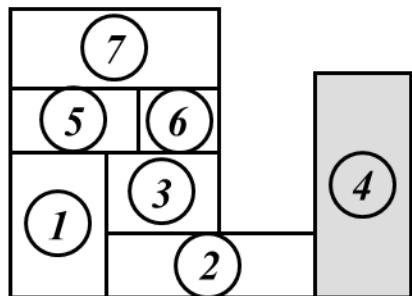
MB*-tree: Multilevel B*-tree (cont'd)



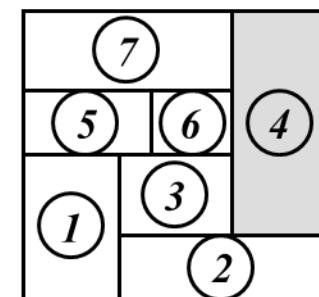
(g) Decluster 9 to 1, 2, 4



(h) Refine the solution by moving 2, 3



(i) Decluster 8 to 5, 6, 7



(j) Refine the solution by moving 4