FastPlace 2.0: An Efficient Analytical Placer for Mixed-Mode Designs

Natarajan Viswanathan
Min Pan
Chris Chu

Iowa State University

ASP-DAC 2006

Work supported by SRC under Task ID: 1206.001

Mixed-Mode Placement

- Design style consisting of a combination of macro blocks and standard cells

- Varied sizes of placeable components complicates the placement step

macro block

standard cells

- Traditionally, divided into two stages: floorplanning or block/module placement and cell placement.

- Designs today can have thousands of macro blocks along with millions of standard cells.

- Therefore, need efficient techniques to simultaneously handle this combination of placeable objects.

Previous Work: FastPlace 1.0

- Natarajan Viswanathan and Chris C.-N. Chu.
  FastPlace: Efficient Analytical Placement using Cell Shifting, Iterative Local Refinement and a Hybrid Net Model.
  In *Proc. International Symposium on Physical Design*, pages 26-33, 2004

- Natarajan Viswanathan and Chris C.-N. Chu.
  FastPlace: Efficient Analytical Placement using Cell Shifting, Iterative Local Refinement and a Hybrid Net Model.
  IEEE Transactions Computer-Aided Design, 24(5): 722-733, 2005

  - Standard cell placement
  - Wirelength minimization
  - Flat placement

  - 13x, 102x and 20x faster than Capo, Dragon and Gordian-Domino
  - Comparable in wirelength

Extend the Standard-Cell placement technique to handle
Mixed-Mode designs

Overview of FastPlace 2.0

Techniques in FastPlace 1.0

Stage 1: Global Placement

1. Hybrid Net Model
2. Cell Shifting for mixed-mode designs
3. Iterative Local Refinement

Stage 2: Legalization

1. Legalize and fix movable macros
2. Legalize standard cells

Stage 3: Detailed Placement

1. Global Swap
2. Vertical Swap
3. Local Re-ordering
4. Single-segment Clustering

Global Placement in FastPlace

- Framework:

  repeat

  Solve the convex quadratic program ❶

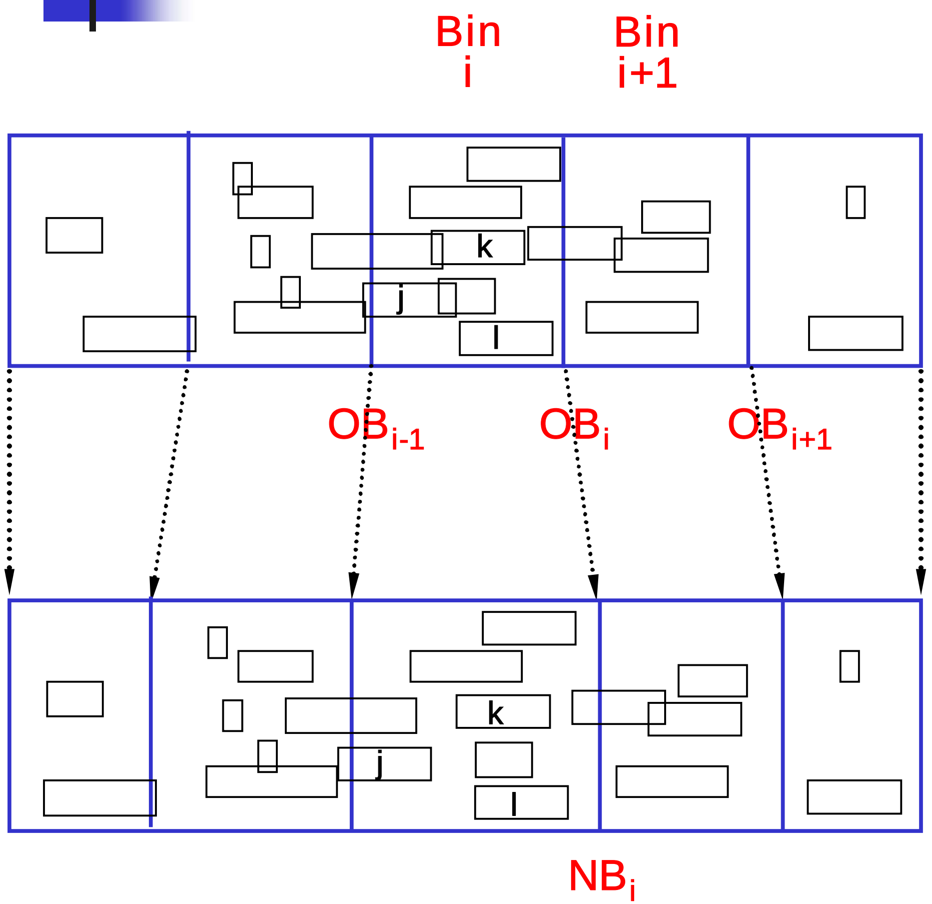
  Reduce wirelength by iterative heuristic ❷

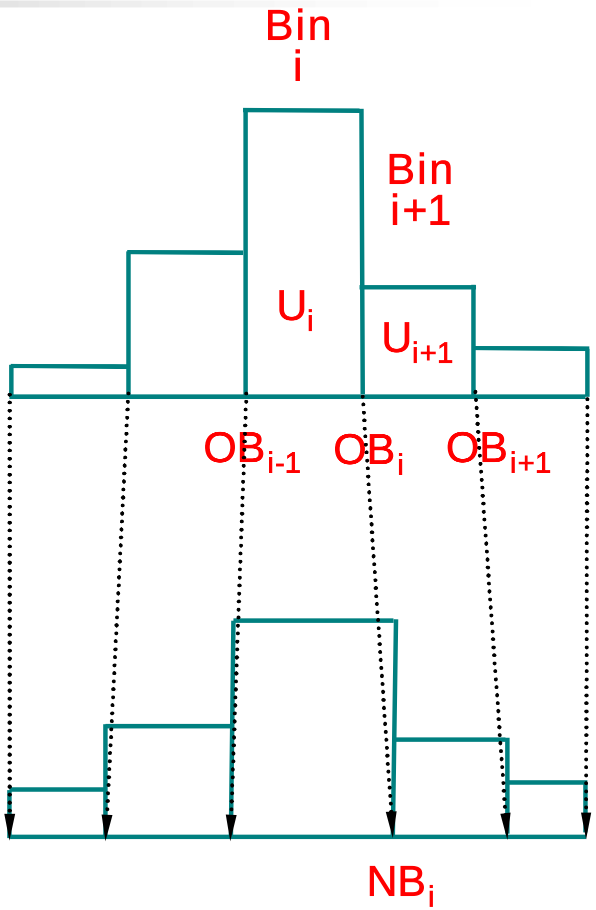
  Spread the cells ❸

  until the cells are evenly distributed ❹

- Special features of FastPlace:
  - Hybrid Net Model
    - Speed up solving of convex QP ❶
  - Cell Shifting
    - Easy-to-compute technique ❸
    - Enable fast convergence ❹
  - Iterative Local Refinement
    - Minimize wirelength based on linear objective ❷

Cell Shifting for Standard Cells

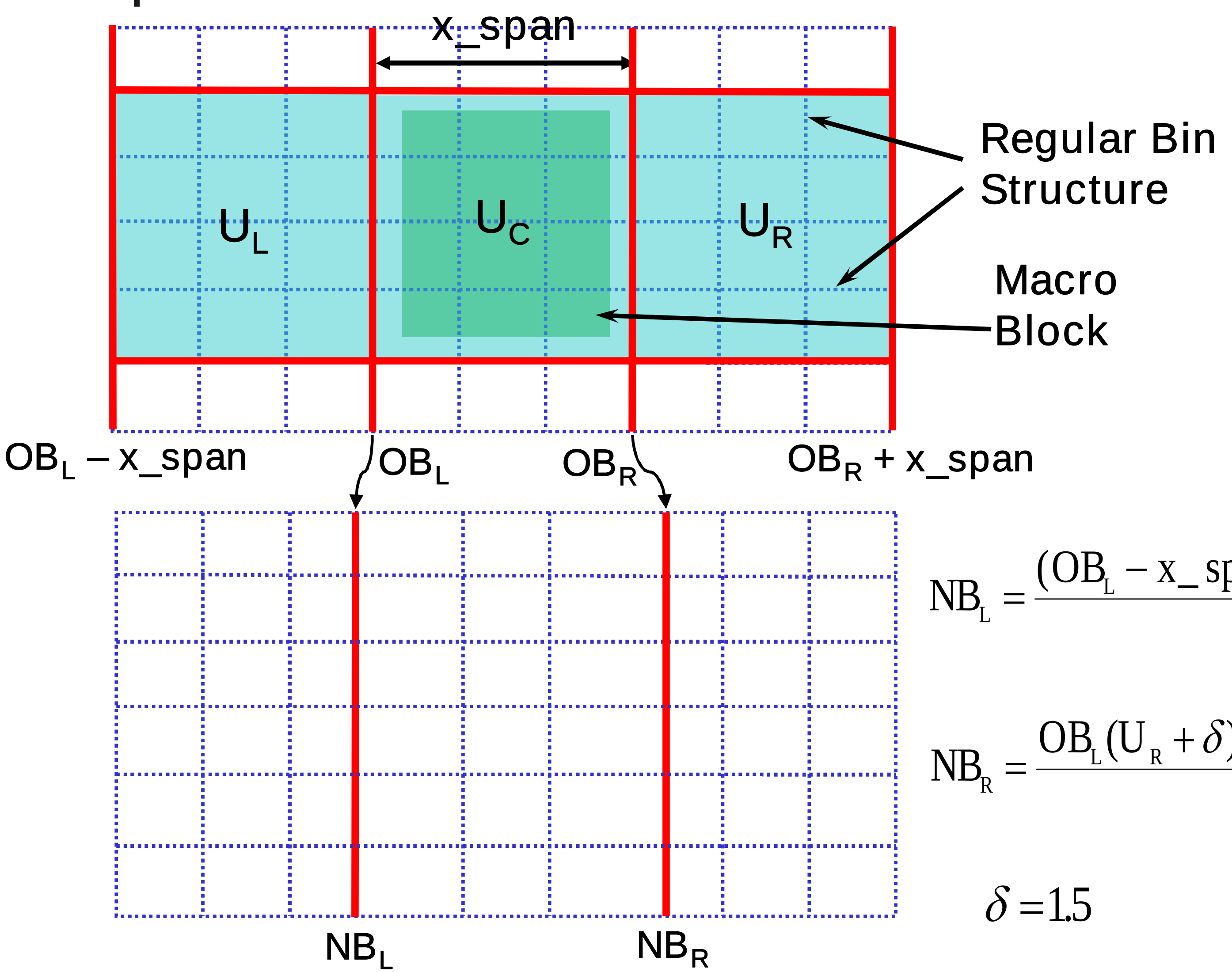




$$NB_i = \frac{OB_{i-1}(U_{i+1} + \delta) + OB_{i+1}(U_i + \delta)}{(U_{i+1} + \delta) + (U_i + \delta)}$$

where δ = 1.5 to prevent cross-over of bin boundaries in the unequal bin structure

Cell Shifting for macros



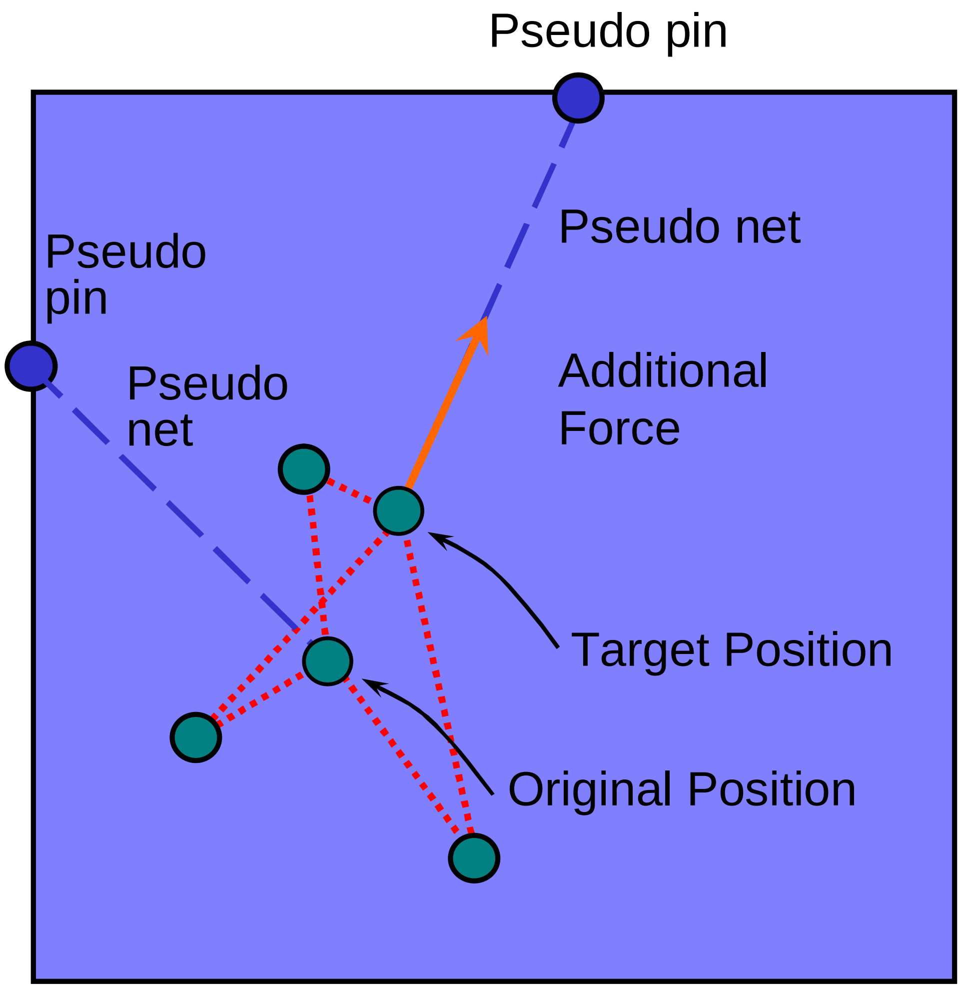
$$NB_L = \frac{(OB_L - x\_span)(U_C + \delta) + OB_R(U_L + \delta)}{U_L + U_C + 2\delta}$$

$$NB_R = \frac{OB_L(U_R + \delta) + (OB_R + x\_span)(U_C + \delta)}{U_R + U_C + 2\delta}$$

$$\delta = 1.5$$

Pseudo pin and Pseudo net

- Need to add forces to prevent cells from collapsing back
- Done by adding pseudo pins and pseudo nets
- Only diagonal and linear terms of the quadratic system need to be updated
- Takes a single pass of *O(n)* time to regenerate quadratic system matrix



Outline

Stage 1: Global Placement

1. Hybrid Net Model
2. Cell Shifting for mixed-mode designs
3. Iterative Local Refinement

Stage 2: Legalization

1. Legalize and fix movable macros
2. Legalize standard cells

Stage 3: Detailed Placement

1. Global Swap
2. Vertical Swap
3. Local Re-ordering
4. Single-segment Clustering

Macro Block Legalization

- Formulated as a fixed-outline floorplanning problem to resolve overlaps with minimum perturbation
- Sequence pair (SP) to represent a floorplan
- Use low temperature simulated annealing to generate a good sequence pair
- Solve the problem independently for x and y

Minimum Perturbation Floorplan Realization Problem

Given: n macros with target coordinates (x_i^*, y_i^*) for $i = 1,\ldots,n$ and a sequence pair (p,q)

Determine: Legalized Coordinates (x_i, y_i) s.t. $\sum_{i=1}^{n} |x_i - x_i^*| + |y_i - y_i^*|$ is minimized.

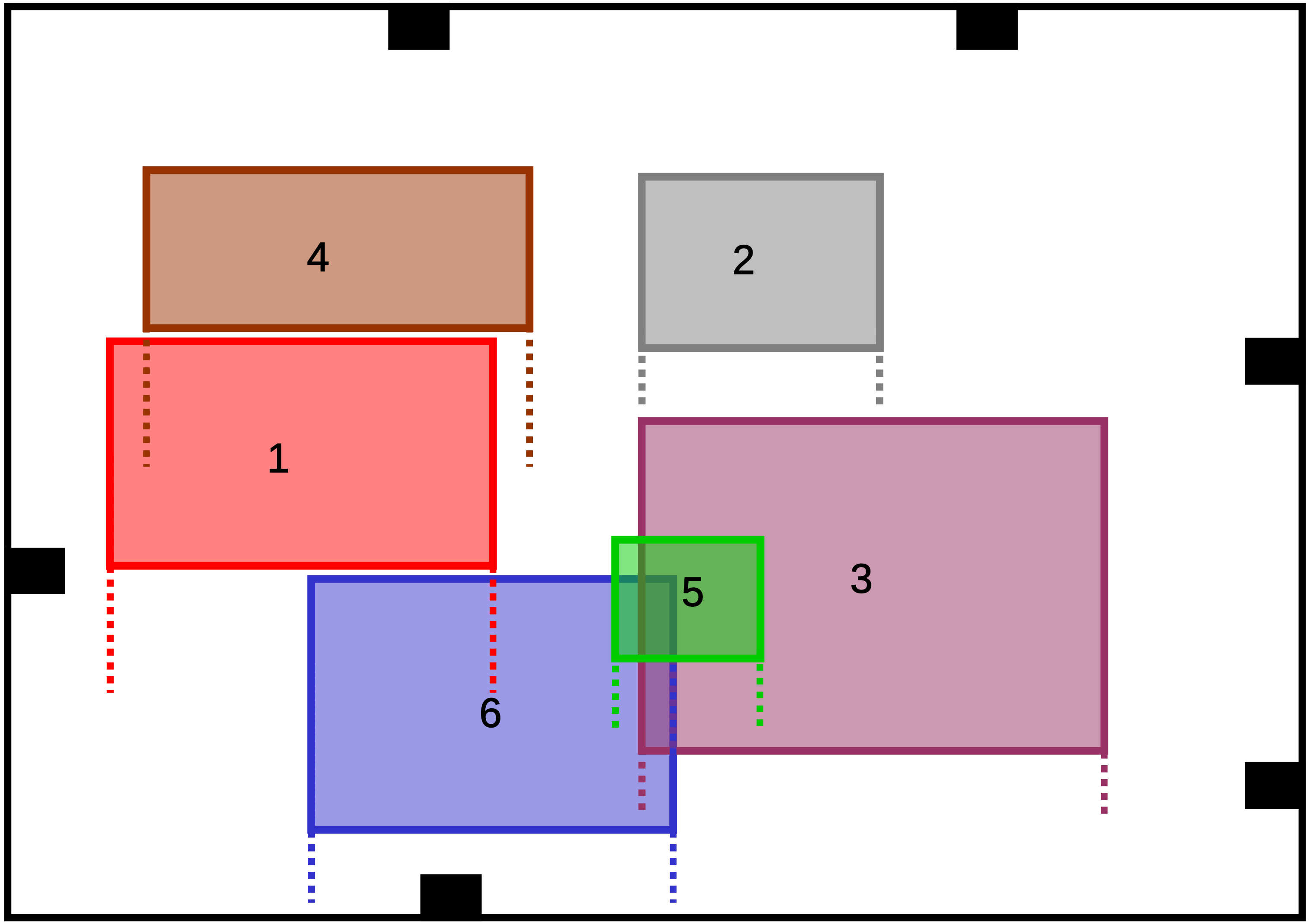
Iterative Clustering Algorithm

- Solves the Minimum Perturbation Floorplan Realization Problem

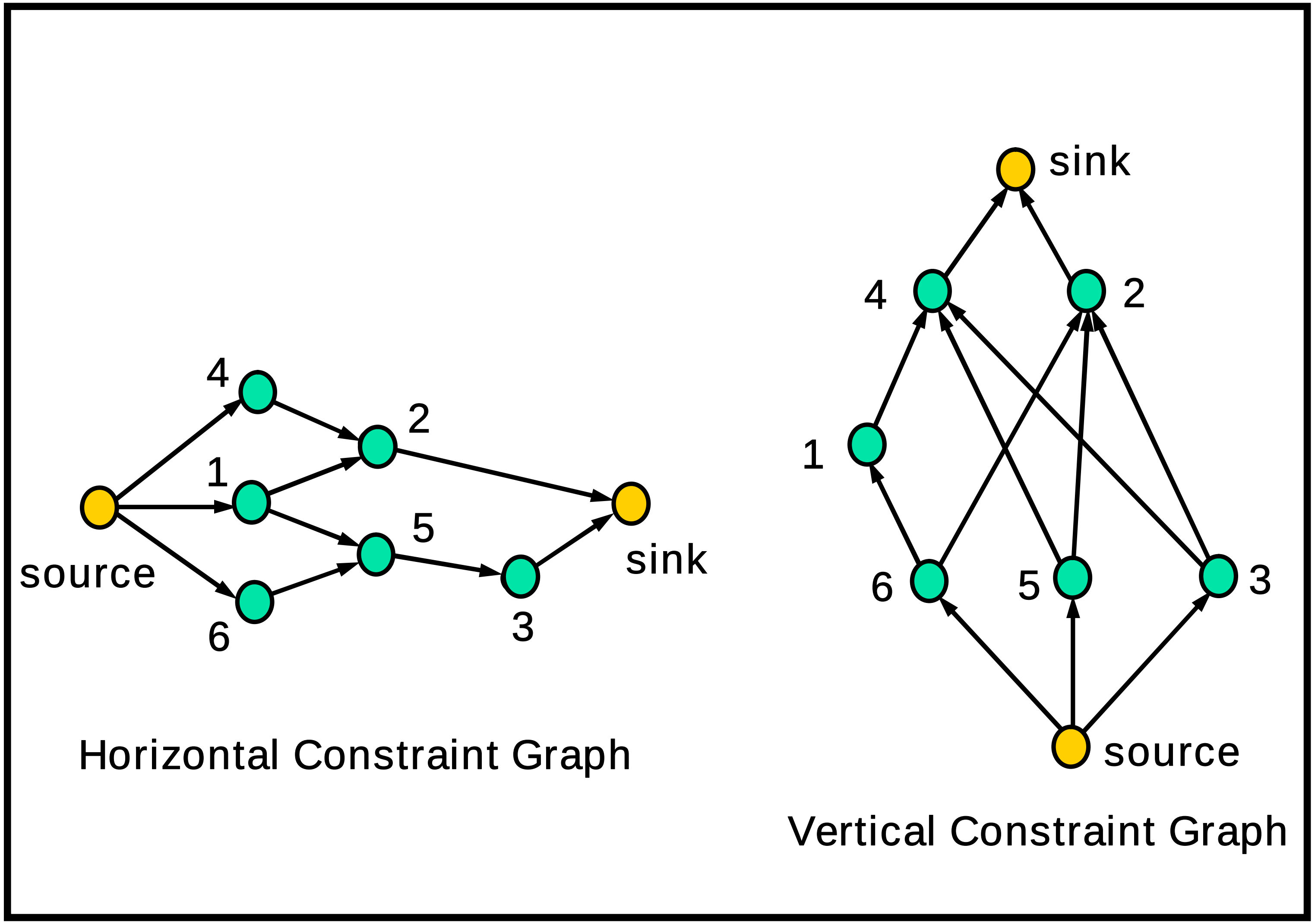
Main Idea:

- If we know which macros to cluster in the optimal solution, then the position of the cluster is easy to find.
- To find correct clustering:
  - Add macros one by one from left to right.
  - Form cluster if overlap occurs.
  - Shift clusters to the left towards their optimal positions
- Optimal for a given sequence pair
- Runtime Complexity: $O(n^2)$. Very efficient in practice.

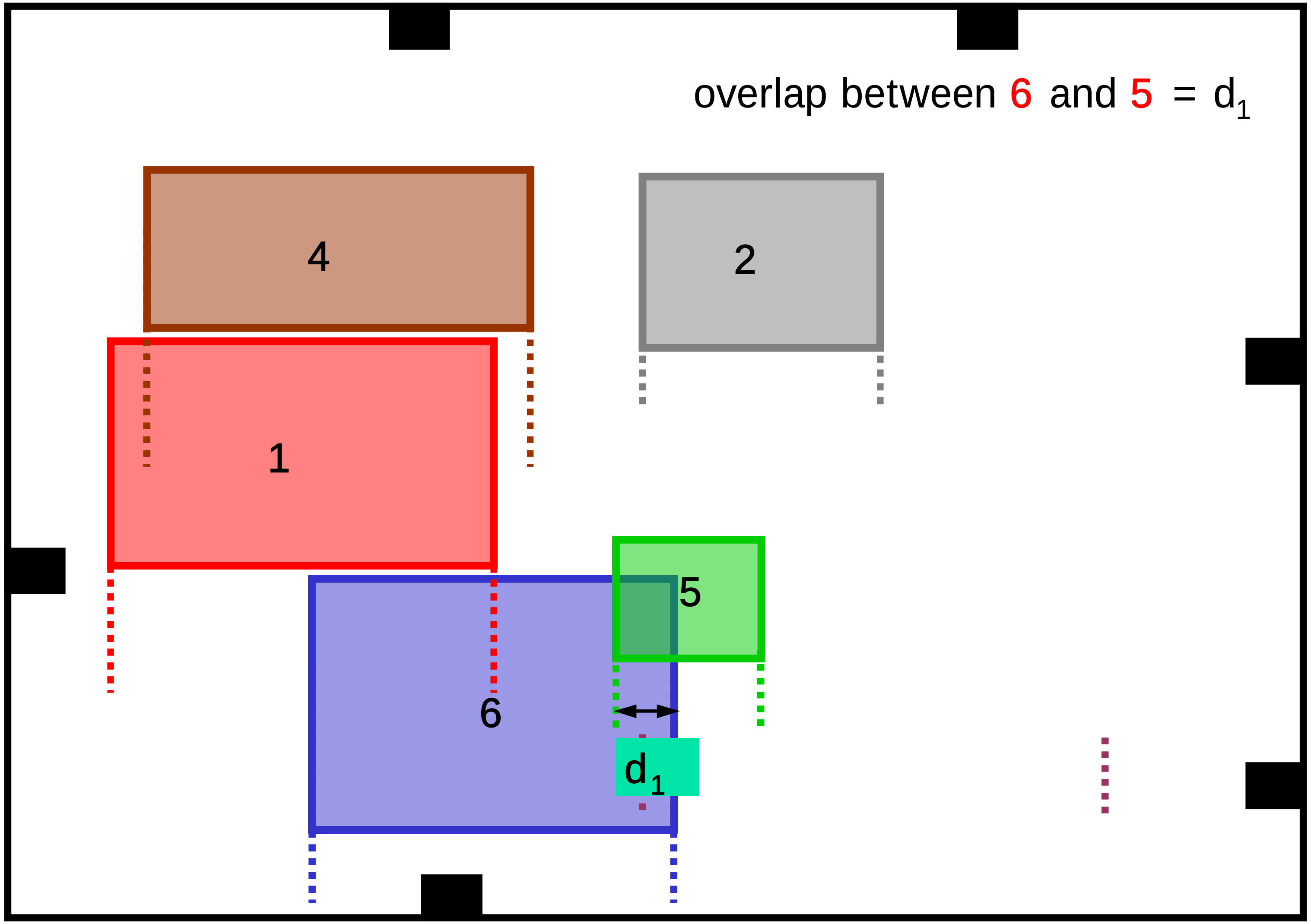
(1) Initial Placement of macros with horizontal overlap



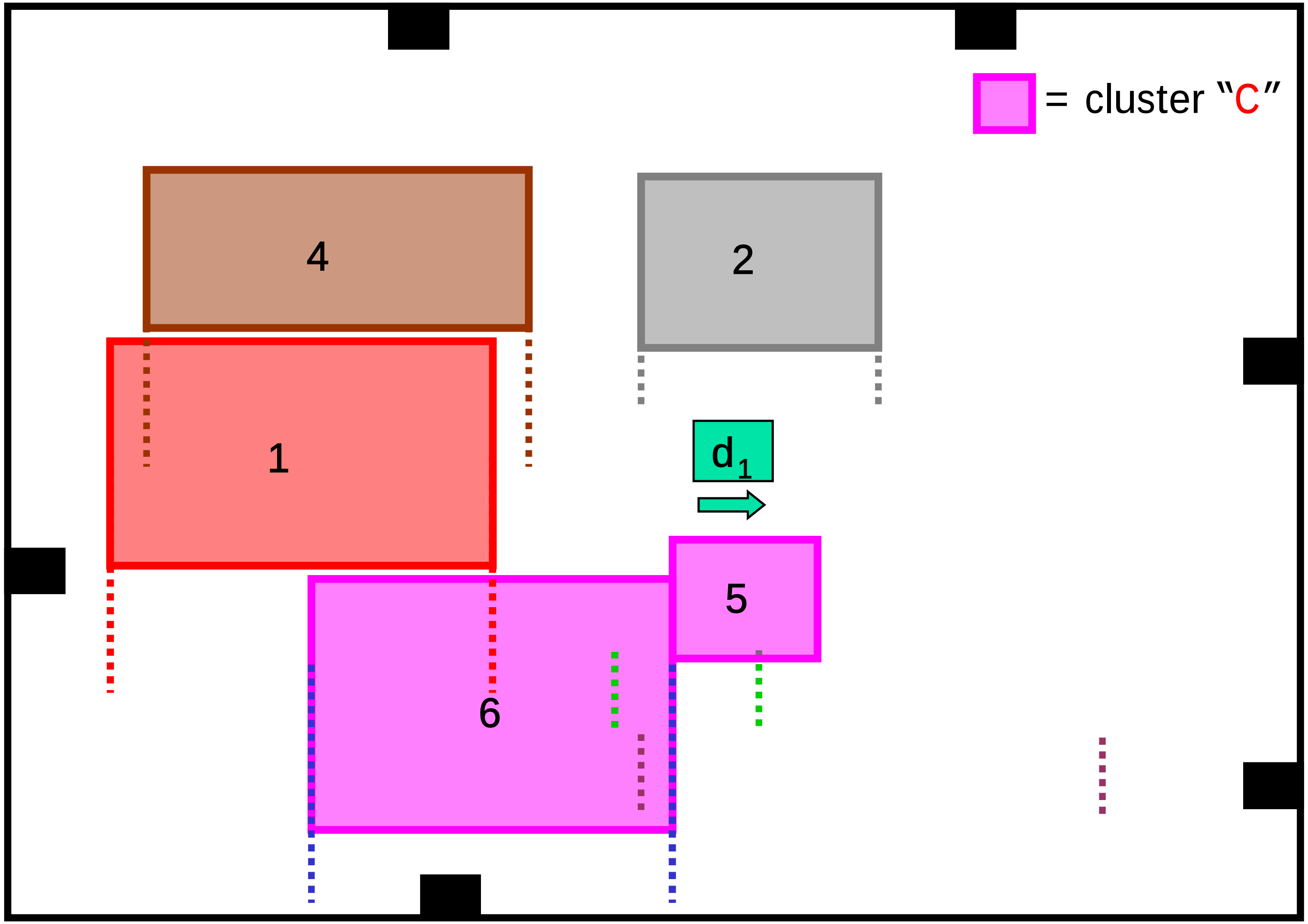
(2) Best Sequence Pair (p, q) = (4 1 2 6 5 3,   6 1 5 3 4 2)



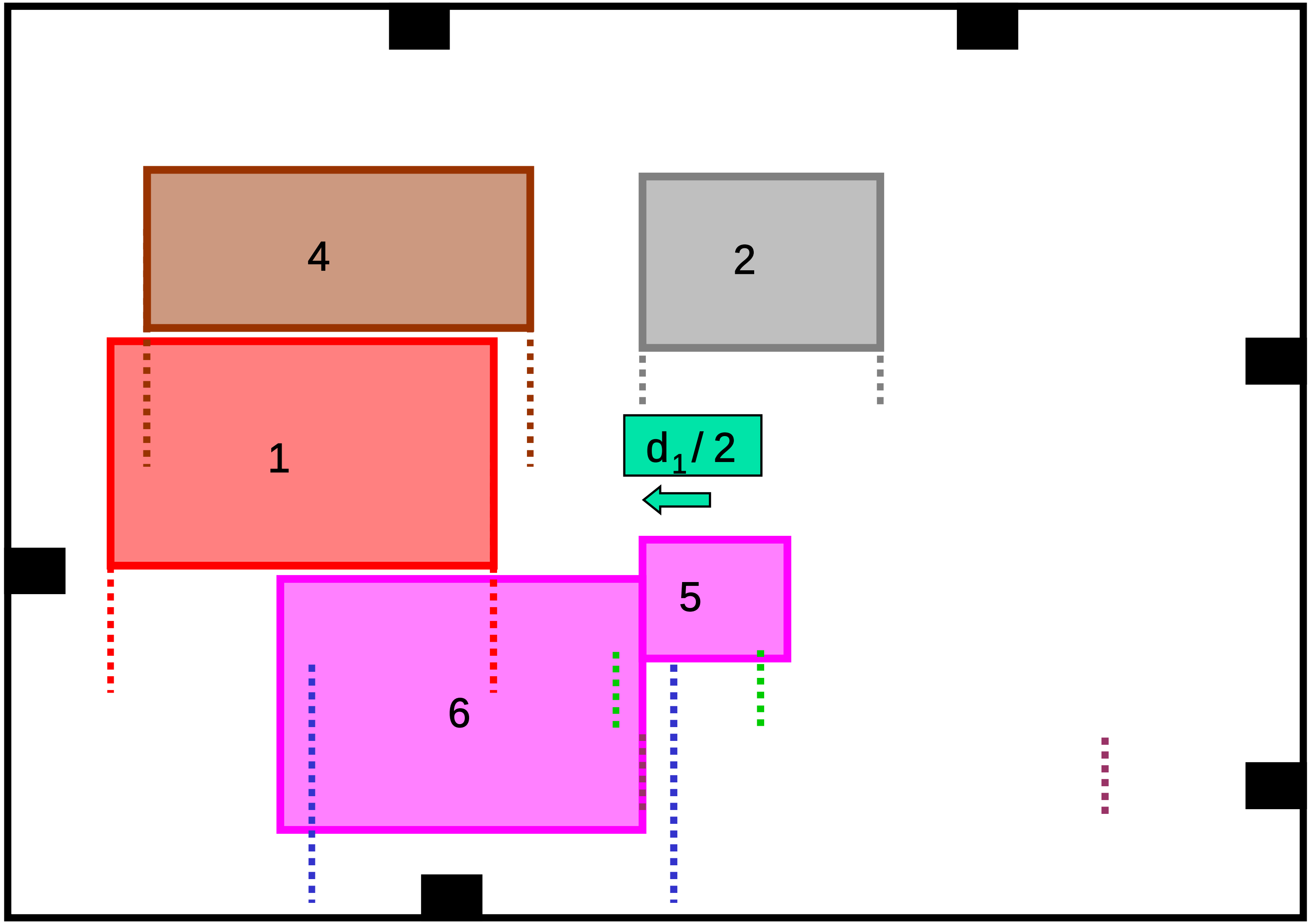
(3) Go through horizontal sequence and add macros from left to right; macros added: 4, 1, 2, 6, 5



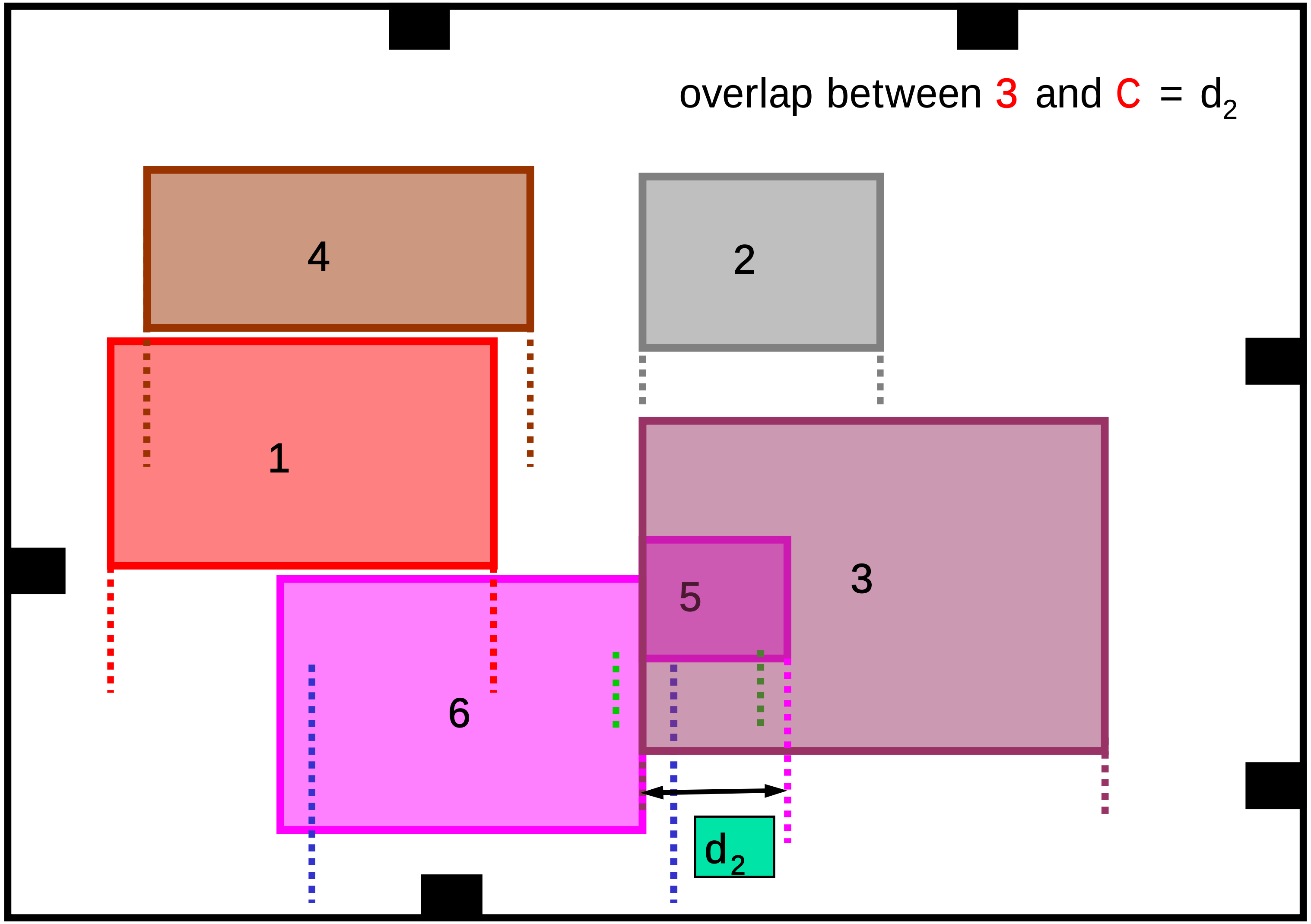
(4) Shift macro 5 from target position because of overlap and form cluster



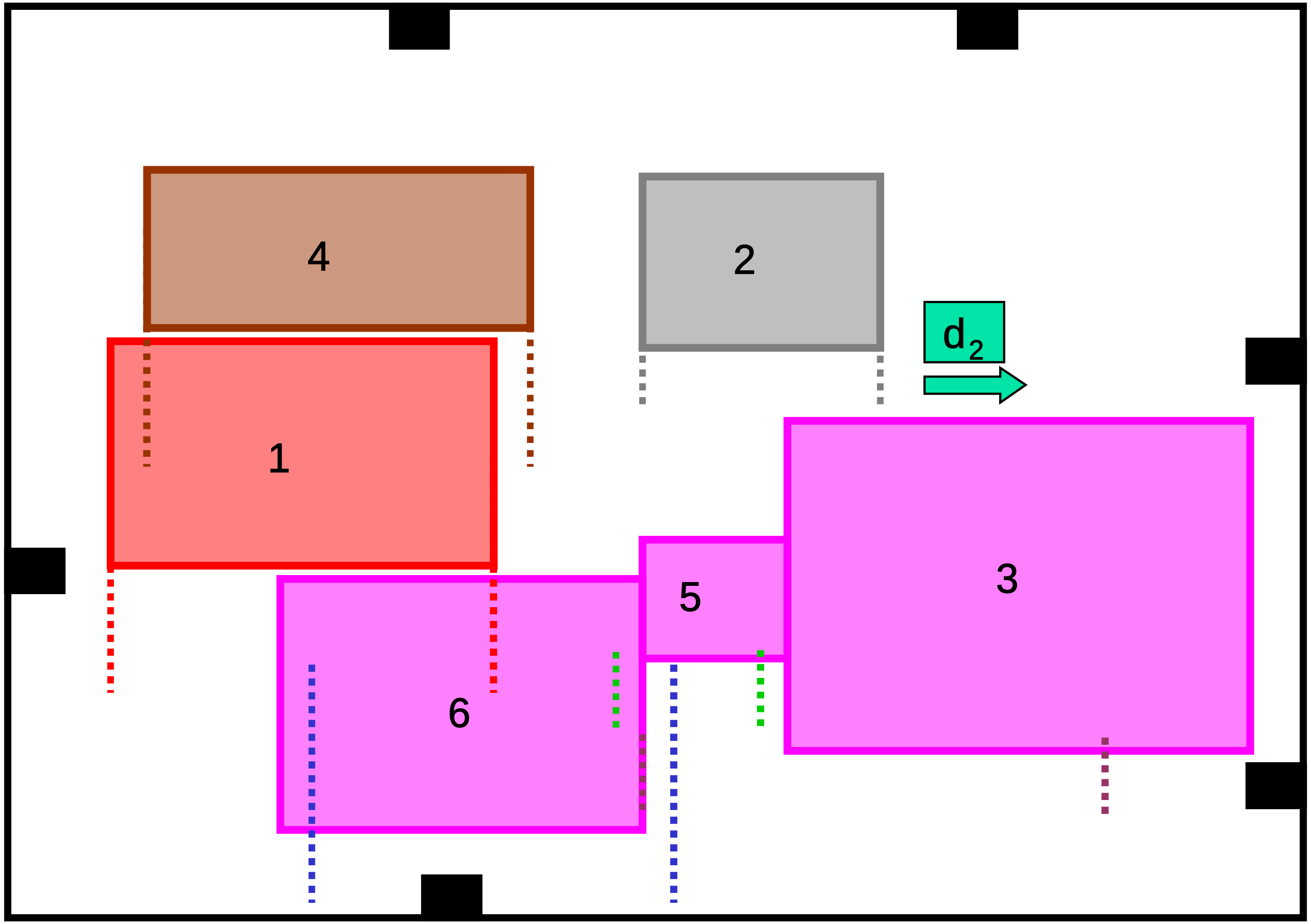
(5) Shift cluster C to the left towards its optimal position



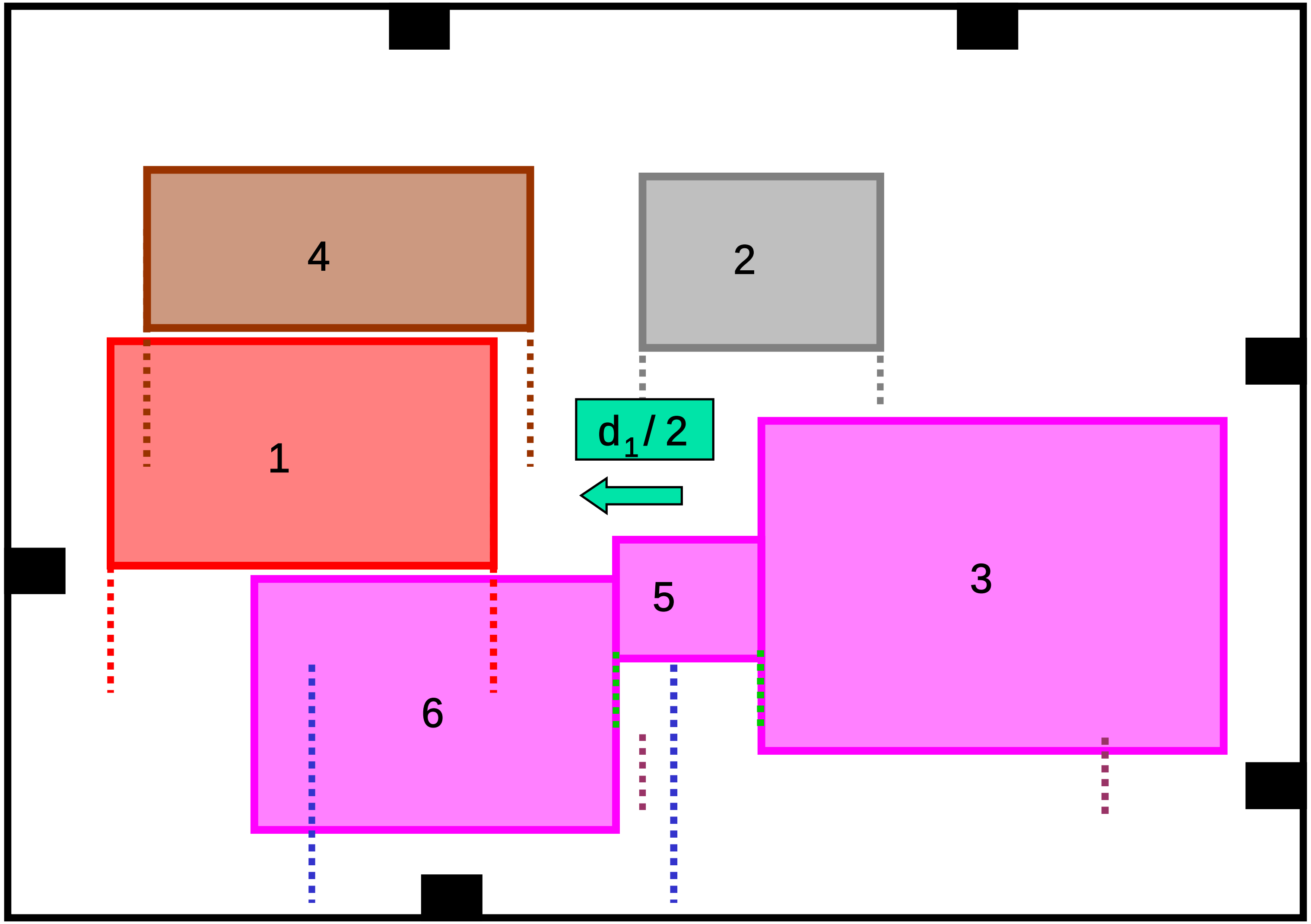
(6) Add next macro in the sequence (macro 3)



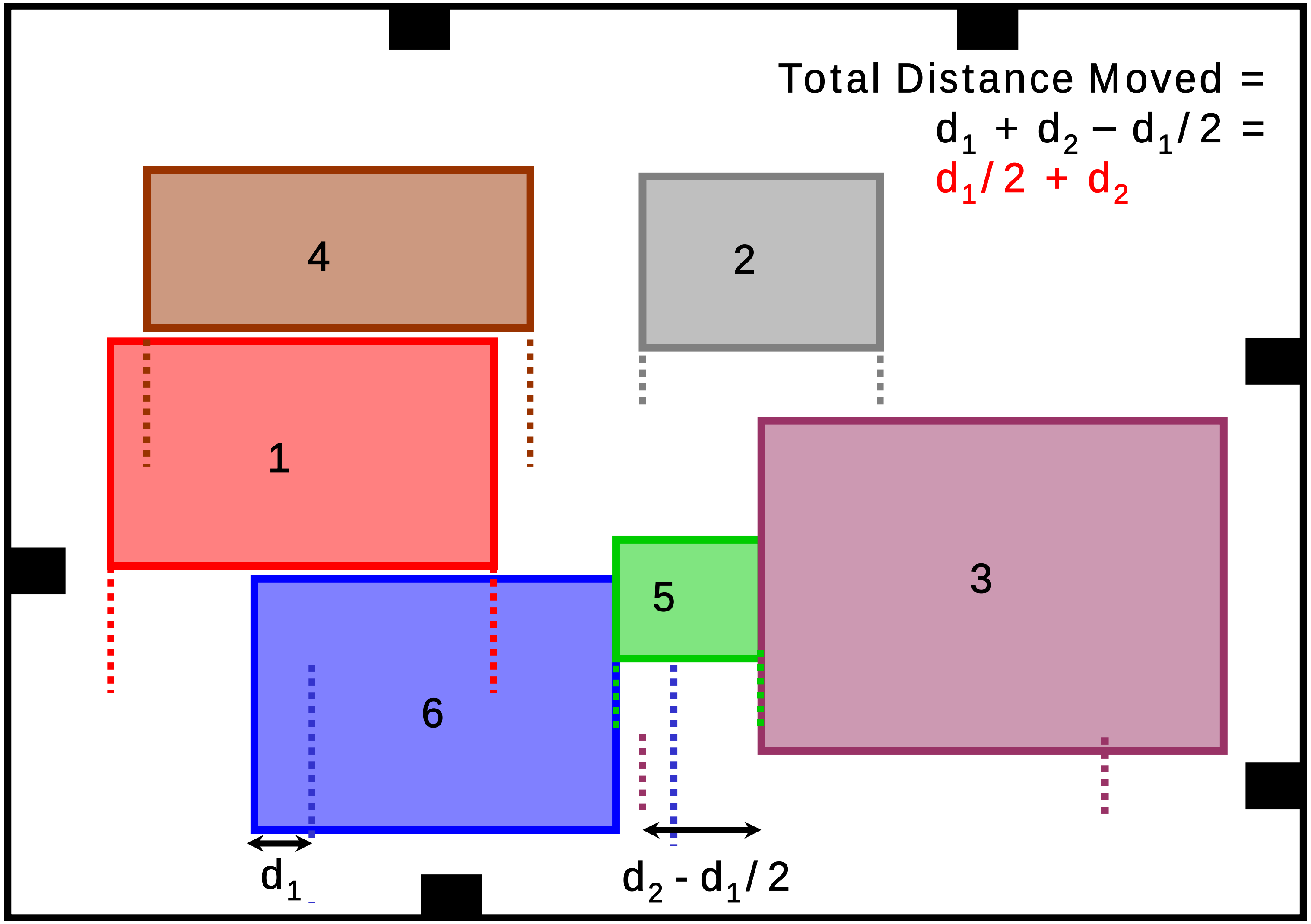
(7) Shift macro 3 from target position because of overlap and form cluster



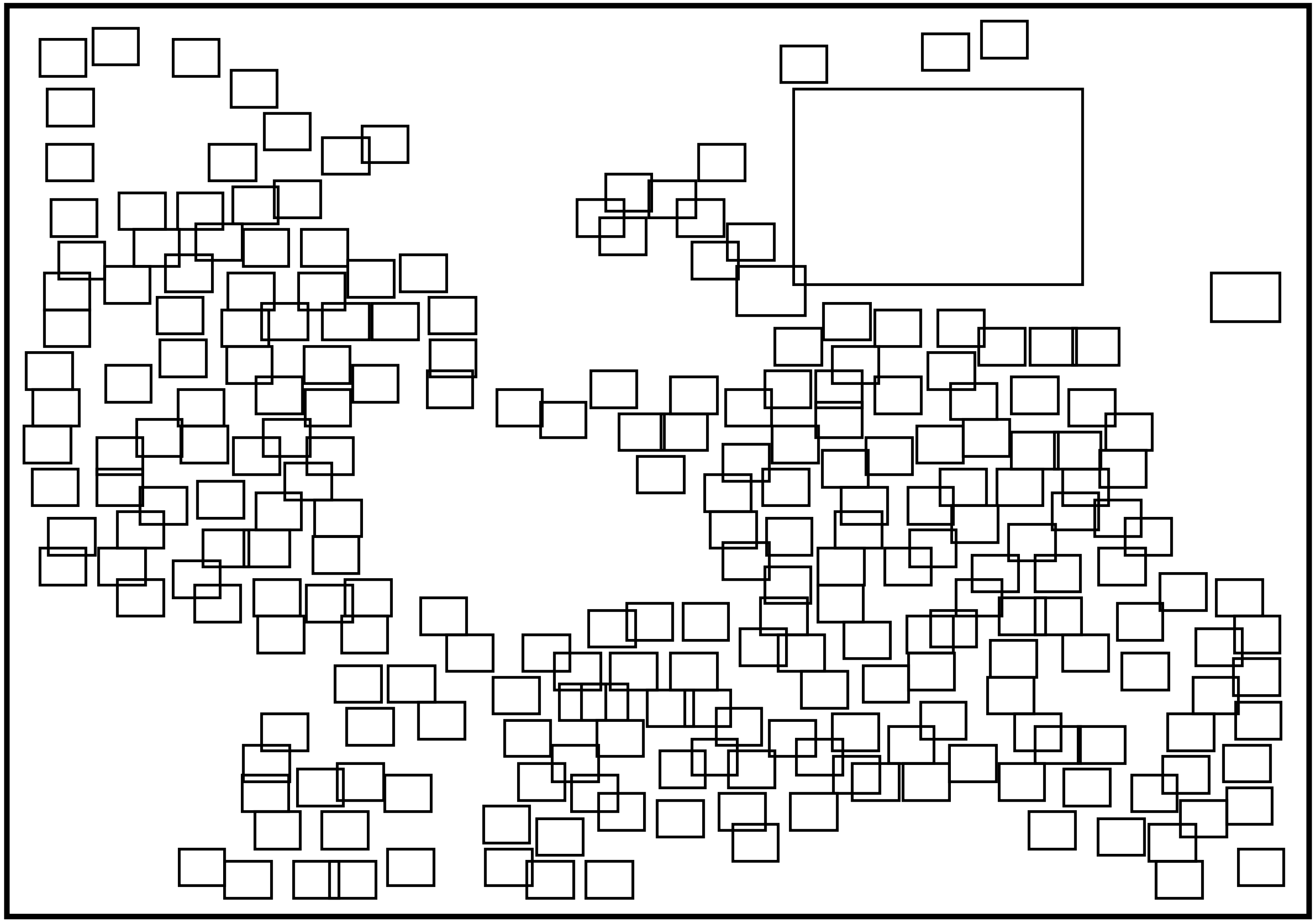
(8) Shift cluster C to the left towards its optimal position



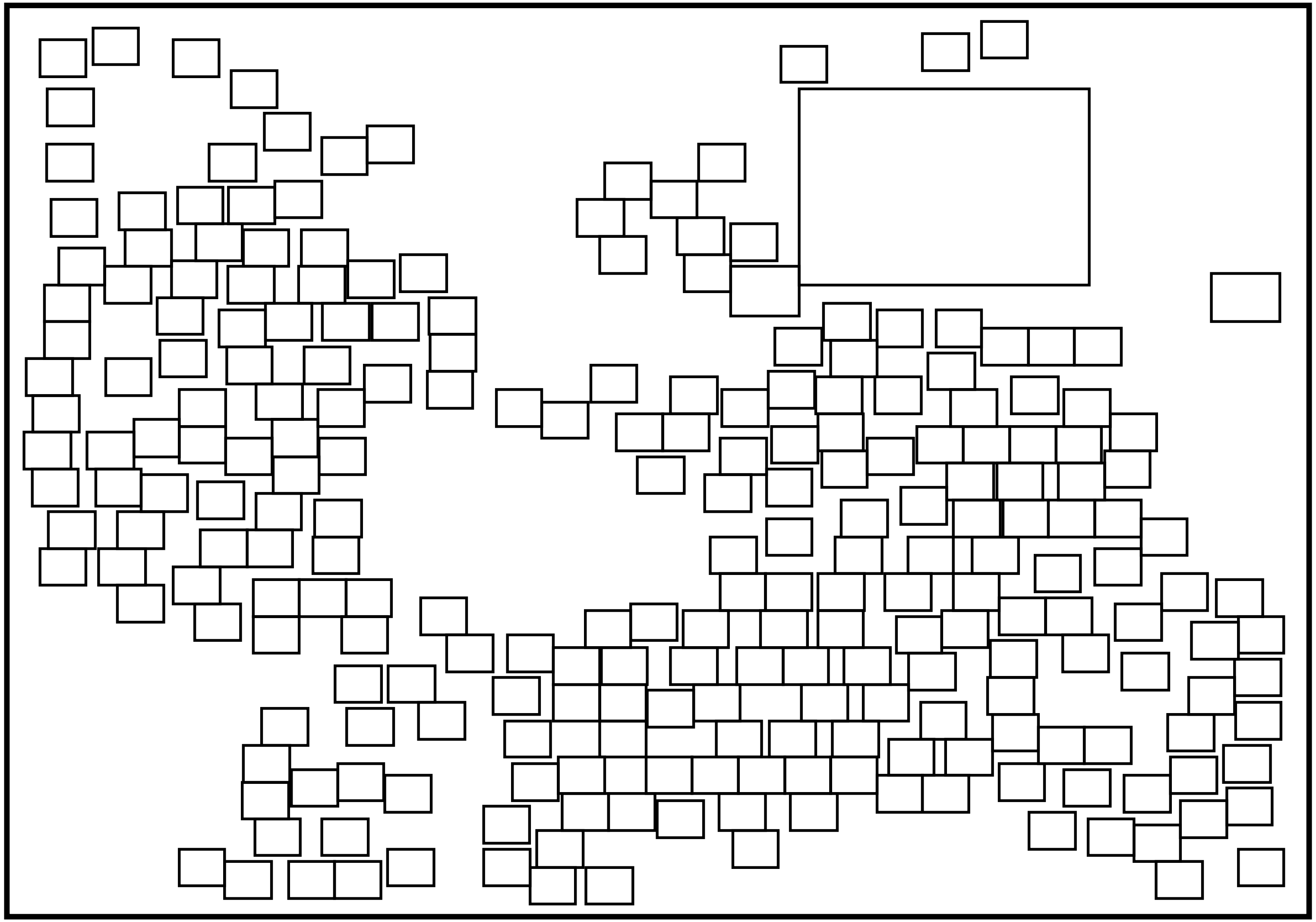
(9) Final Position of macros with no overlap



Macros Before Legalization (ibm01)



Macros After Legalization (ibm01)



Legalize Standard Cells

- Divide rows in the placement region not occupied by macros into segments.
- To Satisfy Segment Capacities
  - Iteratively go through all segments.
  - For each cell in a segment, consider moving it to 8 nearest segments
  - Compute a score for each movement based on
    - Half-perimeter wirelength (HPWL) reduction
    - Cell density at the source and destination segments
  - Move to the segment with highest positive score

    (Do not move if no positive score)
- Legalize cells within segments

Outline

Stage 1: Global Placement

1. Hybrid Net Model
2. Cell Shifting for mixed-mode designs
3. Iterative Local Refinement

Stage 2: Legalization

1. Legalize and fix movable macros
2. Legalize standard cells

Stage 3: Detailed Placement

1. Global Swap
2. Vertical Swap
3. Local Re-ordering
4. Single-segment Clustering

Detailed Placement: Approach

Perform *Single-Segment Clustering*

Repeat

Perform *Global Swap*

Perform *Vertical Swap*

Perform *Local Re-ordering*

Until no significant improvement

Repeat

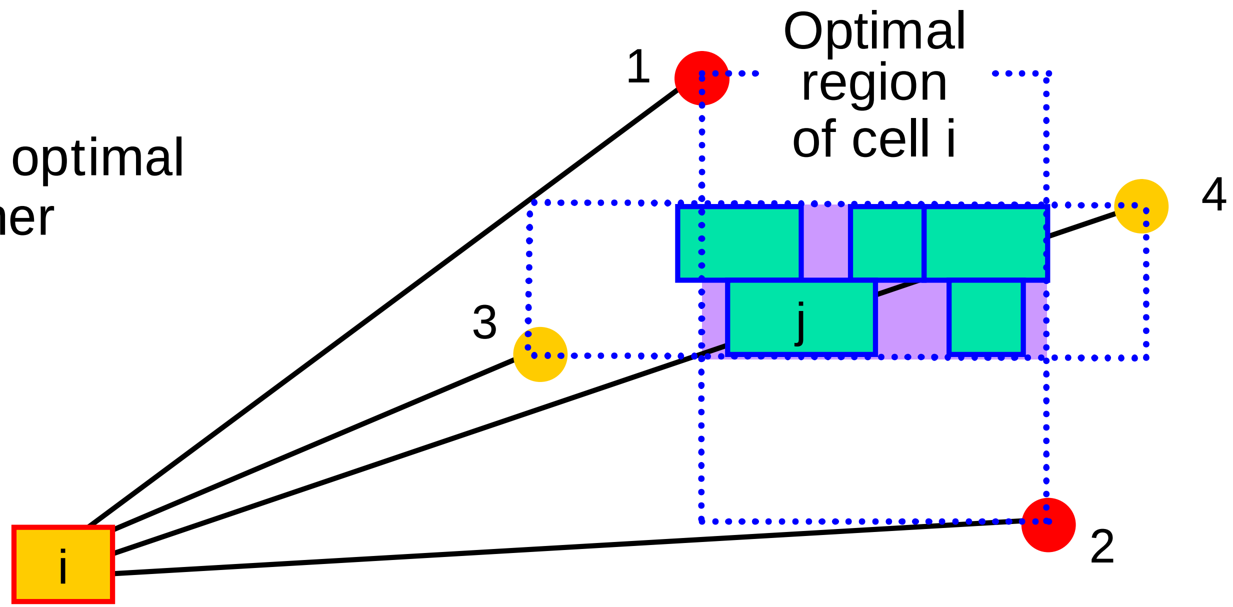
Perform *Single-Segment Clustering*

Until no significant improvement

Main DP Technique: Global Swap

- Idea:

  move a cell to its optimal position while other cells are fixed



- Major steps:
  - For each standard cell i, get its optimal region
  - For every candidate cell j in the optimal region of cell i

    - compute the benefit to swap i with j
  - For every candidate space s in the optimal region of cell i

    - compute the benefit to swap i with s
  - Pick the cell or space with best benefit to perform the swap

Other DP Techniques

- Vertical Swap:
  - Move a cell vertically toward its optimal region to reduce the wirelength.
  - Interleave with global swap to accelerate convergence.
- Local Re-ordering
  - For 3 consecutive cells in a segment, get the best order in terms of wirelength
- Fixed-Order Single Segment Clustering
  - Cluster cells within segments to further reduce wirelength

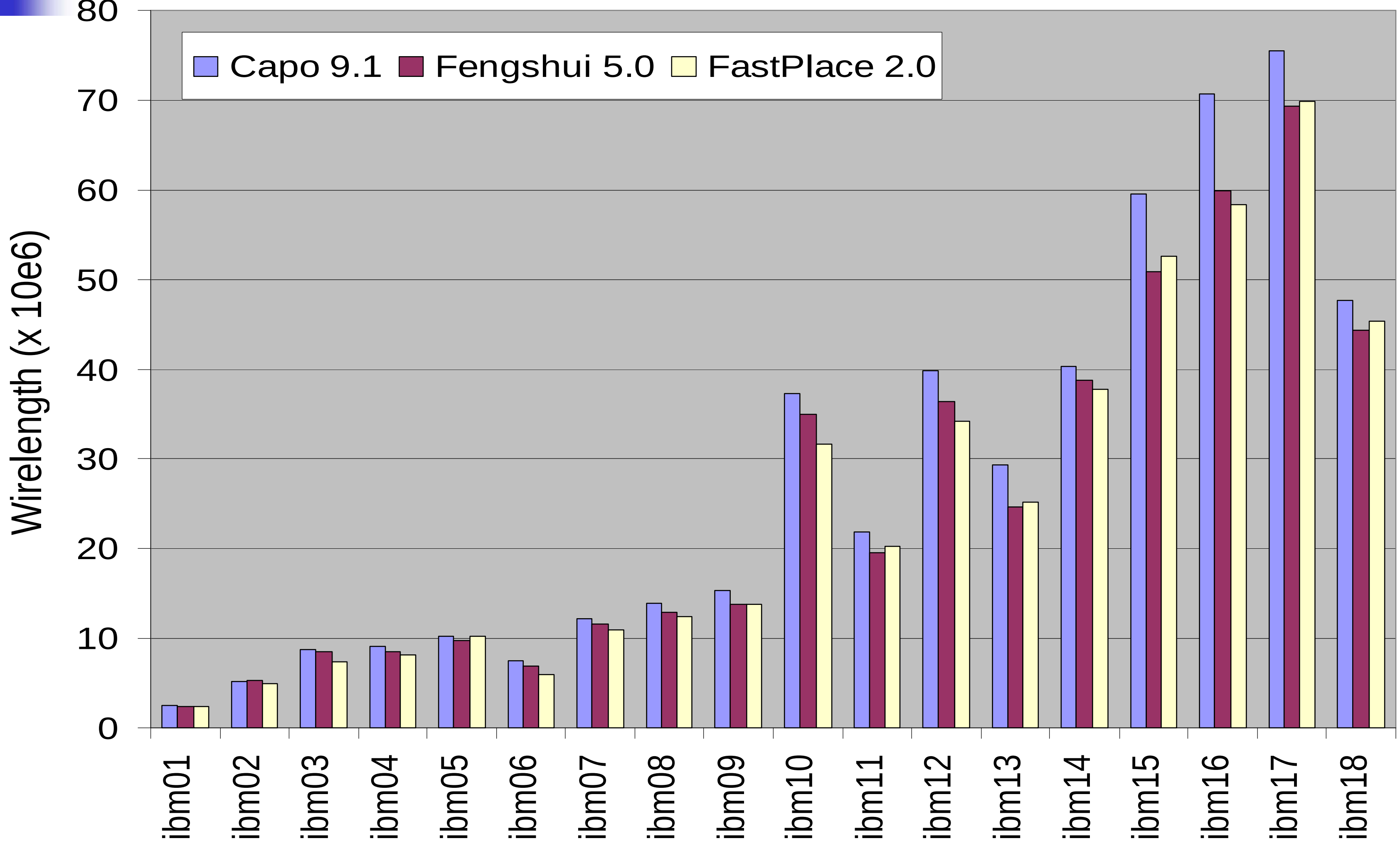
Experimental Setup

- ISPD02 IBM-MS Mixed-size placement benchmarks
- 12k – 210k movable nodes
- up to 786 macro blocks
- 20% whitespace

- FastPlace 2.0 implemented in C
- Compared with:
  - Capo 9.1
  - Fengshui 5.0

- All experiments are on an Intel Xeon, 3.06GHz CPU

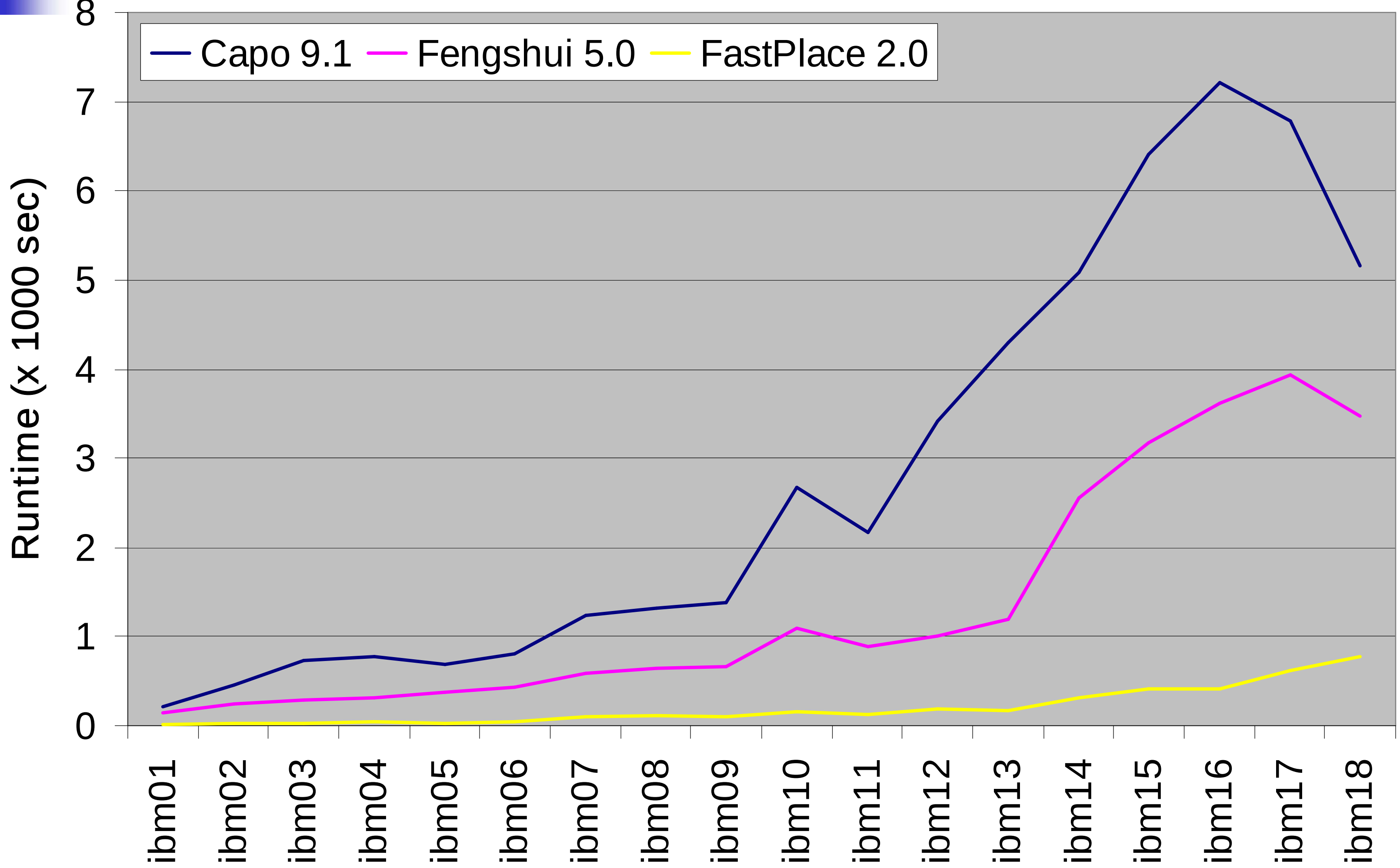
Half-Perimeter Wirelength



Average Wirelength Ratio

Capo / FastPlace : **1.12** Fengshui / FastPlace : **1.03**

Runtime



Average Runtime Ratio

Capo / FastPlace : **16.84** Fengshui / FastPlace : **7.84**

Conclusions and Future Work

- FastPlace 2.0: An Efficient analytical placer for Mixed-Mode designs
  - Cell Shifting for macro blocks
  - Iterative Clustering Algorithm for macro block legalization
  - Efficient standard-cell legalization in the presence of macros
  - Efficient and effective detailed placement algorithm
- 12% better in wirelength and 16.8X faster than Capo 9.1
- 3% better in wirelength and 7.8X faster than Fengshui 5.0
- Need to handle rotation and mirroring for macro blocks

Thank You

Questions ??

Iterative Clustering Algorithm

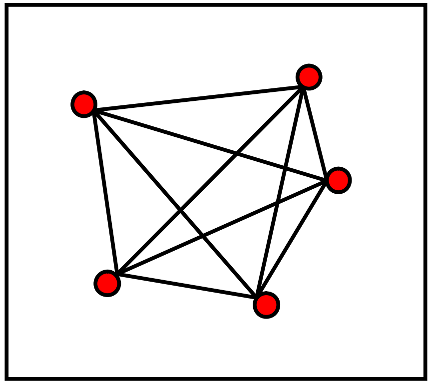
1. Find the immediate left and right neighbors of all macros
2. **for** *i = 1* **to** *n*
3. Place macro p_i in its target position
4. Let *C* be a new cluster consisting of p_i
5. **while** *C* overlaps with other clusters **do**
6. Merge *C* with the closest cluster on its left
7. Let *C* be the new cluster formed
8. Shift *C* to its optimal position
9. **if** macro *m* in *C* is at its target position **do**
10. Detach *m* from *C* if necessary and goto step 8
11. **endwhile**
12. **endfor**

Analytical Placement Formulation

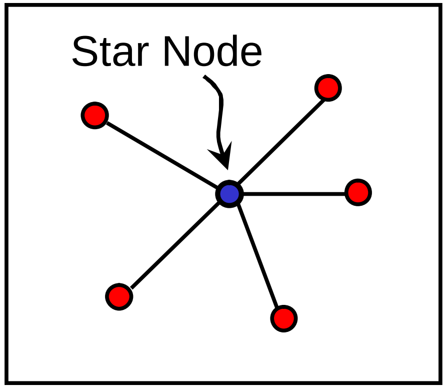
- Let (x_i, y_i) = Coordinates of the center of cell i
  w_{ij} = Weight of the net between cell i and cell j
  x, y = Solution vectors
- Cost of the net between cell i and cell j
  $$= \frac{1}{2} w_{ij} \left((x_i - x_j)^2 + (y_i - y_j)^2 \right)$$
- Total cost $= \frac{1}{2} x^T Q x + d_x^T x + \frac{1}{2} y^T Q y + d_y^T y + \text{const}$
- Analytical Placement Framework:
  repeat
  Solve the convex quadratic program
  Spread the cells
  until the cells are evenly distributed

Hybrid Net Model

- Need to replace multi-pin nets by 2-pin nets in the convex QP formulation
- Use Incomplete Cholesky Conjugate Gradient (ICCG) solver
- Runtime is proportional to # of non-zero entries in Q
- Each non-zero entry in Q corresponds to one 2-pin net
- Traditionally, placers model each multi-pin net by a clique
- High-degree nets will generate a lot of 2-pin nets
- Slow down convex QP algorithms significantly



Clique Model



Star Model

# pins	Net Model
2	Clique
3	Clique
4	Star
5	Star
6	Star
…	…