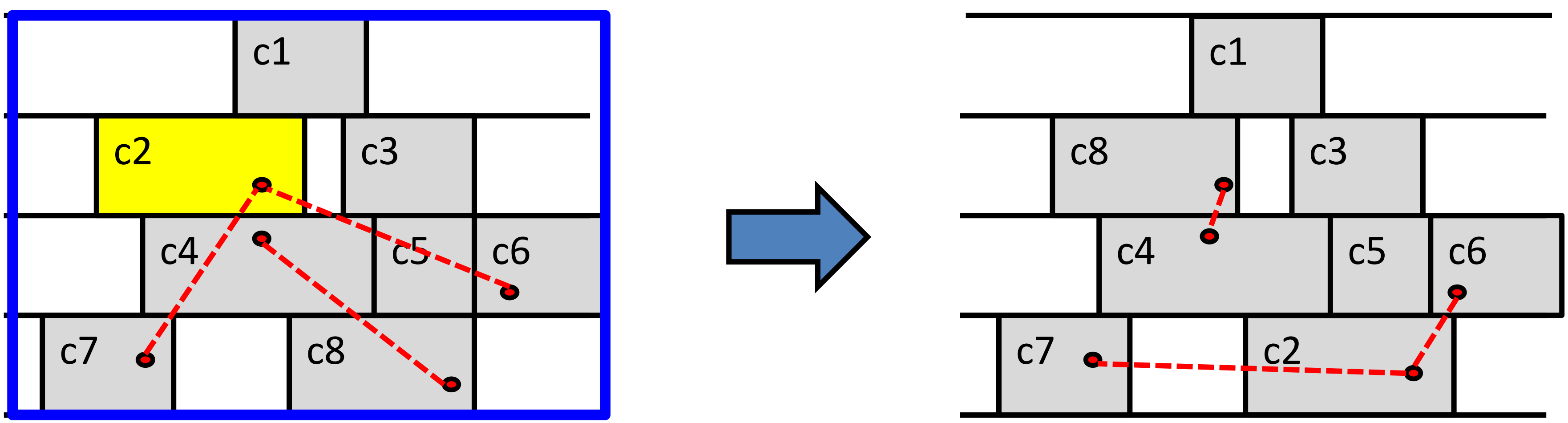
Detailed Placement

- Given a legalized placement solution, detailed placement further improves the wirelength (or other objectives) by locally rearranging cells while maintaining legality.
- Room for wirelength improvement?
  - A global placement algorithm typically uses an inaccurate wirelength model
  - A global placement algorithm might place a cell in a subregion without paying attention to the location of the cell within the subregion
  - During legalization, wirelength is likely to be worsened

Detailed Placement Techniques (1/5)

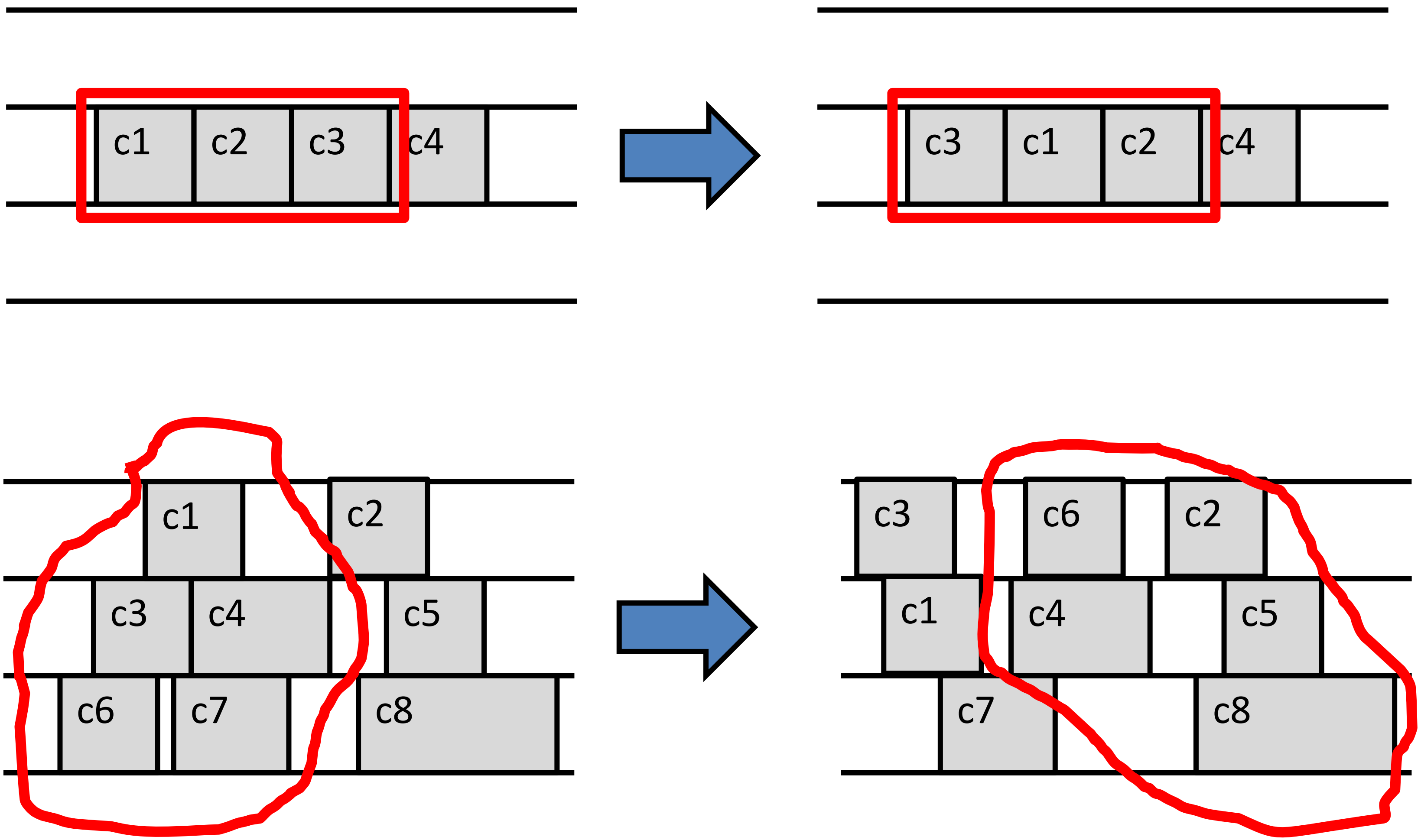
- Moving cells to minimize total HPWL without cell overlapping

Cell Swapping



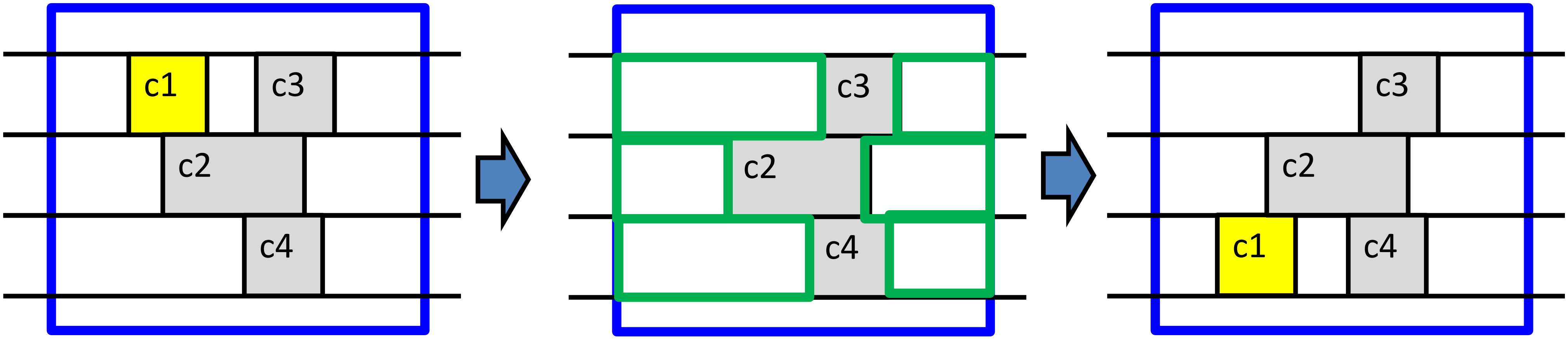
Detailed Placement Techniques (2/5)

Sliding Windows (followed by Branch and Bound or Network Flow)

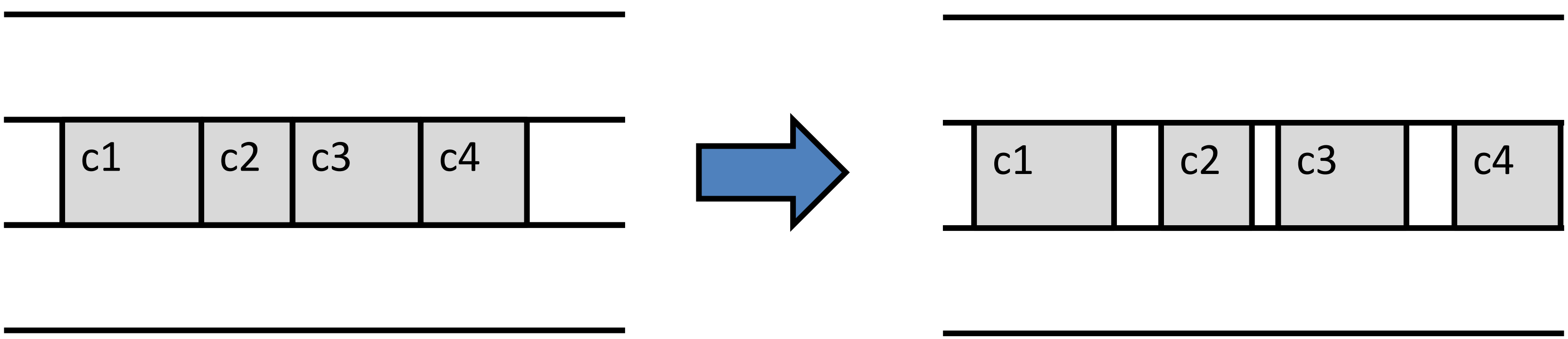


Detailed Placement Techniques (3/5)

Moving Cells to Empty Spots

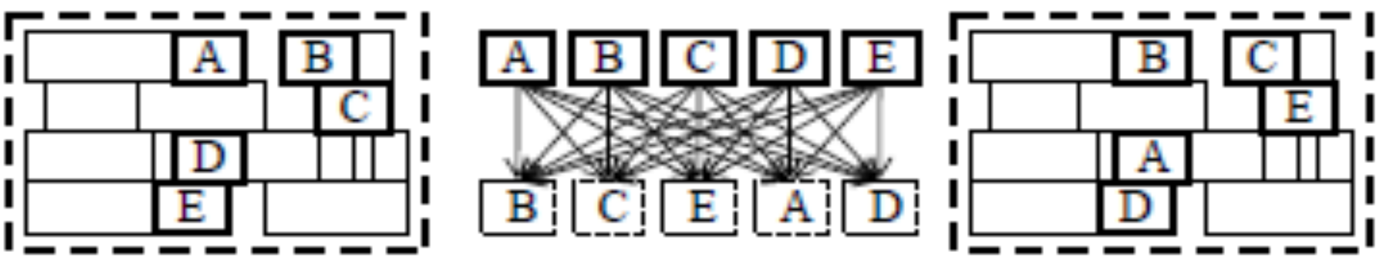


Dynamic-Programming-based Single-Row Placement



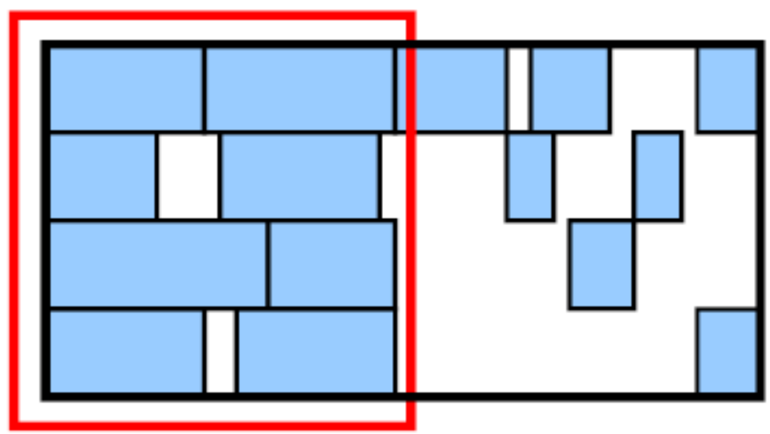
Detailed Placement Techniques (4/5)

- Cell matching
  - Select a window
  - Select **independent** cells (i.e., no common net between any pair of cells) from the window
  - Create a bipartite matching problem (edge weight = wirelength)
  - Find the minimum weighted matching to optimize the wirelengt
  - Update cell positions



Detailed Placement Techniques (5/5)

- Cell sliding: density optimization
- Steps
  - Select an overflow window
  - Calculate the amount of area to shift
  - Move cells left/right to reduce the density

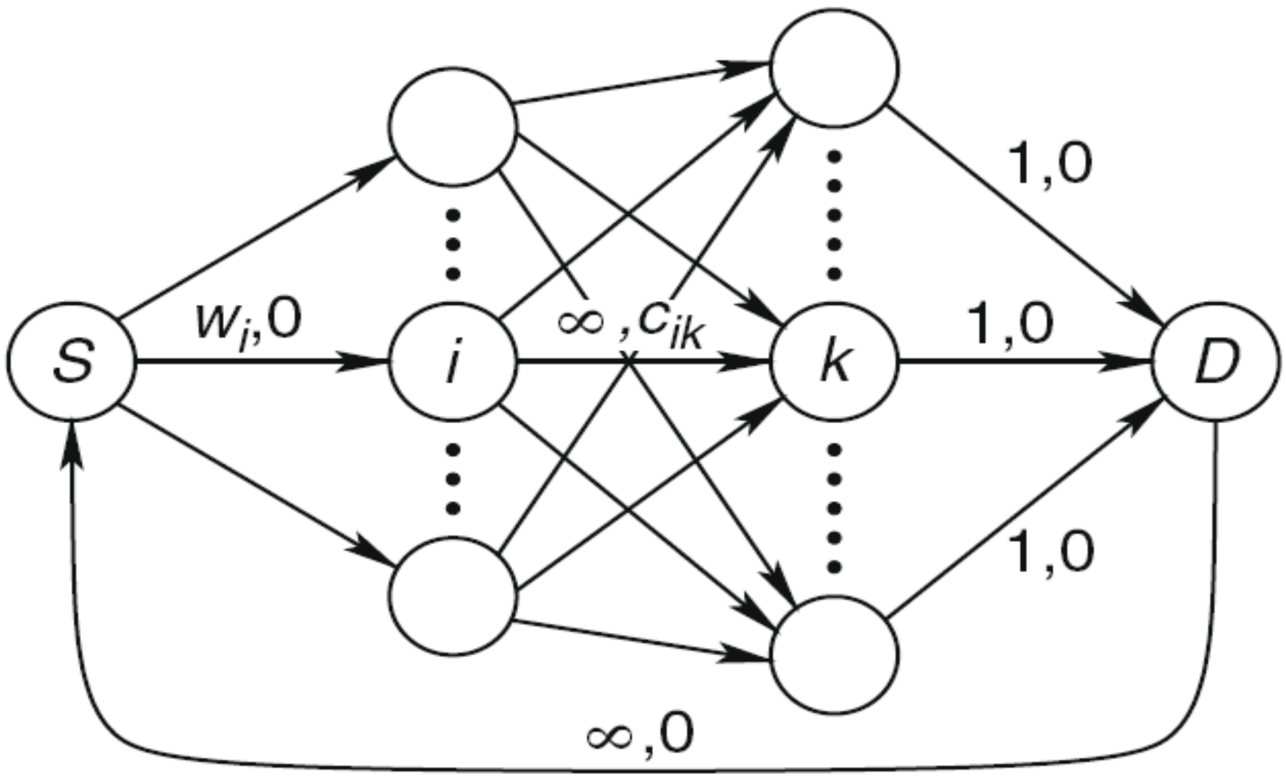


Detailed Placement: Domino [IEEE TCAD-94]

- Use a sliding window approach to iteratively refine a small region
- The problem of cell assignment is formulated as a transportation problem
- Each cell is divided into unit-width subcells
- The problem of transporting the subcells to the placement slots that minimizes WL is transformed into a **minimum cost maximum flow** problem

Min Cost Max Flow Problem

- c_{ik} models the total HPWL of nets connected to cell i if a subcell of i is assigned to location k
  – Estimated by analyzing different cases



Cost Calculation

- e: a net connected to cell i
- e_i: the set of cells connected to net e inside the region.
- Three cases
  - $|e_i| = 1$: The only cell inside the region that net e connects to is cell i. The HPWL of net e can be calculated exactly.
  - $1 < |e_i| < |e|$: Net e connects cells other than cell i both inside and outside the region. The unknown locations of cells in $e_i - \{i\}$ are estimated by their coordinates in the current placement. Then the HPWL of net e can be calculated.
  - $|e_i| = |e|$: Net e connects only to cells inside the region. The locations of cells in $e_i - \{i\}$ are estimated by their coordinates in the current placement. Besides, a virtual cell is introduced at the center of gravity of the cells in e with respect to the current placement. The HPWL of all cells in e together with the virtual cell is calculated and used.

Cell Arrangement

- After solving the network flow problem, subcells are assigned to locations.
- For all subcells of each cell, as they are associated with the same transportation cost and are pulled towards the cheapest location, they tend to lie side by side.
- Each cell is placed in the row holding most of its subcells, and the x-coordinate of the cell is determined by the center of gravity of its subcells.
- The cells in each row of the region are packed according to their x-coordinates to prevent overlap.

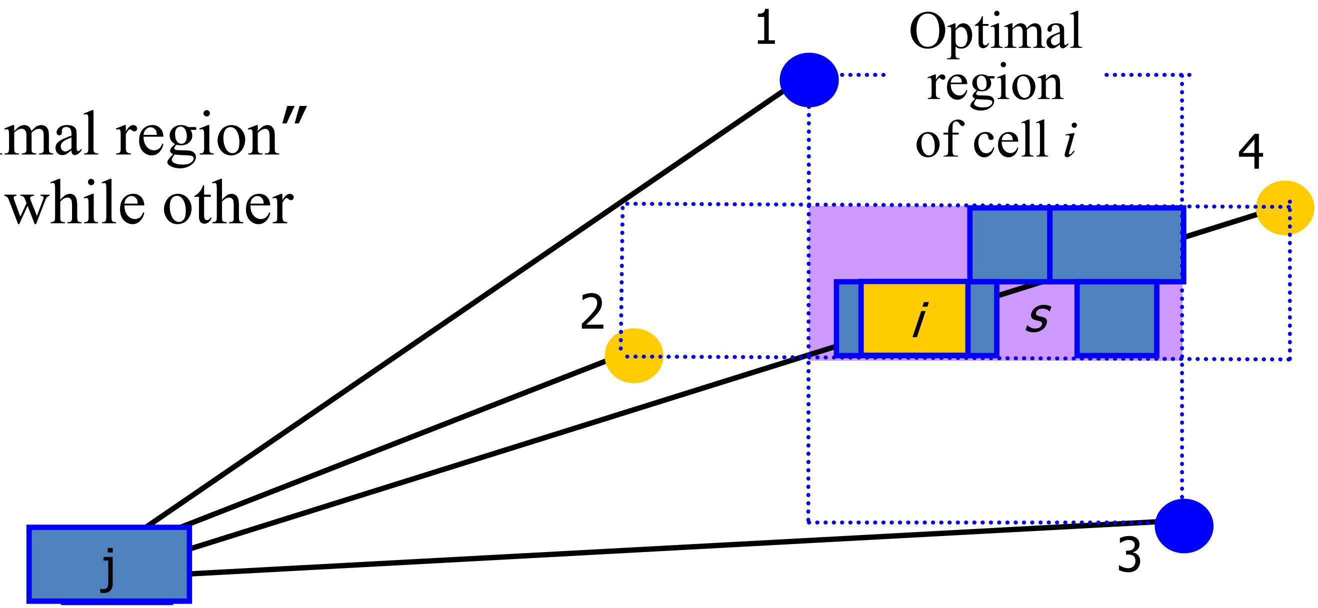
Detailed Placement: FastDP [ICCAD-05]

- A very fast and high-quality greedy heuristic

1. Perform single-segment clustering
2. Repeat
3. Perform global swap
4. Perform vertical swap
5. Perform local reordering
6. Until no significant improvement in wirelength
7. Repeat
8. Perform single-segment clustering
9. Until no significant improvement in wirelength

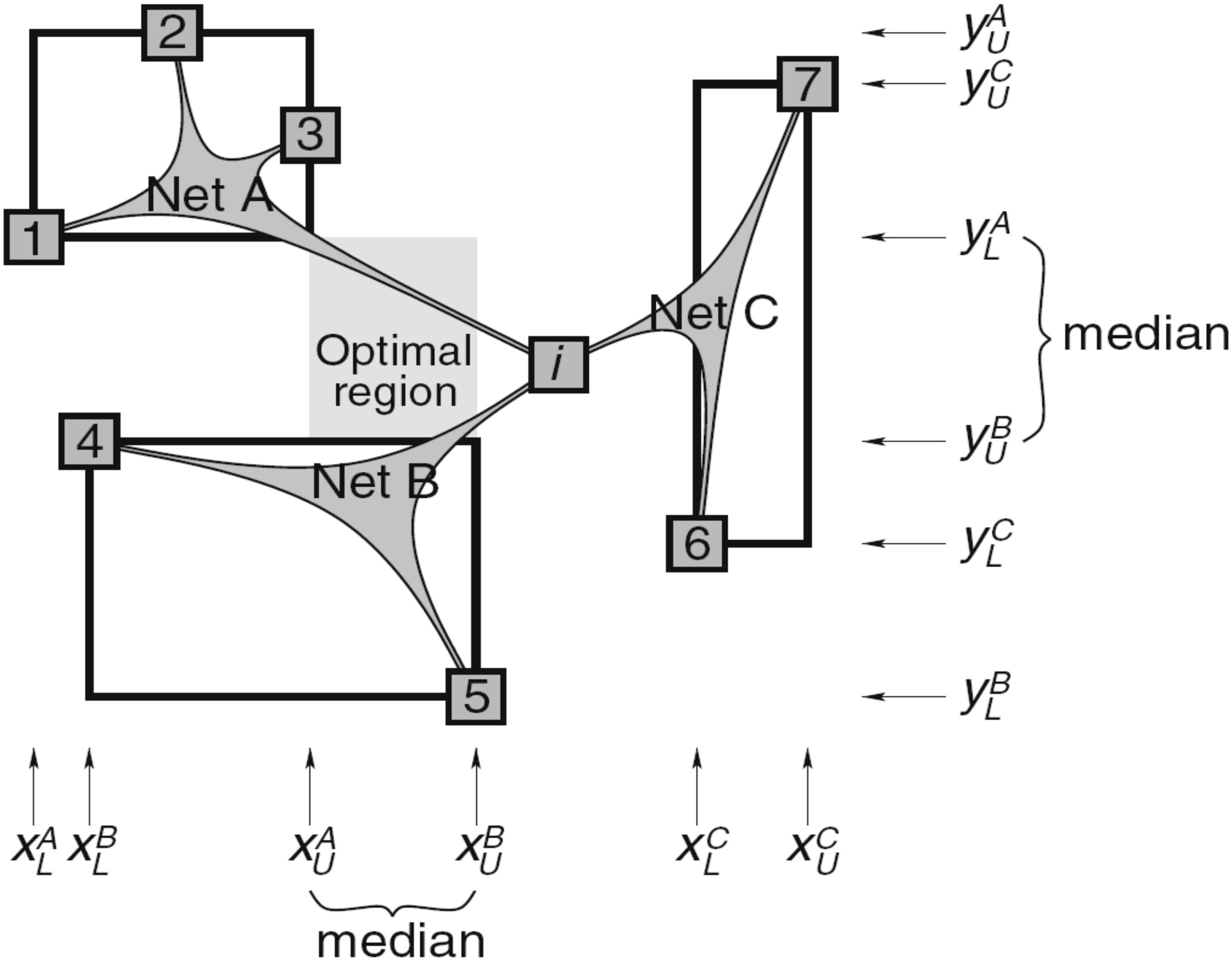
Global Swap

- **Main idea:**
  Move a cell to its "optimal region" (HPWL is minimized) while other cells are fixed



- **Major steps:**
  – For each standard cell *i*, find its optimal region
  – For every candidate cell *j* in the optimal region of cell *i, c*ompute the benefit to swap *i* with *j*
  – For every candidate space *s* in the optimal region of cell *i, c*ompute the benefit to swap *i* with *s*
  – Pick the cell or space with best positive benefit to perform swap

Optimal Region of a Cell [IEEE CAS-81]



Other Techniques in FastDP

- **Vertical swap:**
  - Move a cell one row up or down toward its optimal region to reduce the wirelength.
- **Local reordering:**
  - To fix local errors by reordering 3 adjacent cells
- **Single segment clustering:**
  - Optimally shift cells in a segment (i.e., a maximal unbroken section of a row) to minimize HPWL in linear time

Survey Papers on Placement

- Y.-W. Chang, Z.-W. Jiang, and T.-C. Chen, "Essential Issues in Analytical Placement Algorithms", IPSJ Trans. on Systems LSI Design Methodology, 2009.

- I. Markov, J. Hu, and M.-C. Kim, "Progress and Challenges in VLSI Placement Research", Proceedings of the IEEE, 2015.

Summary (1/10)

- Strengths (+) and weaknesses (-) for three types of placers [IPSJ 2009]

Simulated Annealing Placement
(+) Easier to consider multiple objectives simultaneously (+) Good quality for small designs (–) Harder to handle modules of very different sizes (–) Slower and less scalable for large circuits
Min-Cut Placement
(+) More efficient and scalable, even for large circuits (+) Good at mixed-size circuit legalization (–) Harder to handle multiple objectives simultaneously (–) Harder for whitespace management, especially for designs with low utilization rates
Analytical Placement
(+) More efficient and scalable, even for large circuits (+) Better quality for large-scale designs (+) Good at whitespace management, regardless of utilization rates (+) Easier to handle multiple objectives simultaneously (–) Harder to legalize large macros (–) Harder to optimize macro orientations

Summary (2/10)

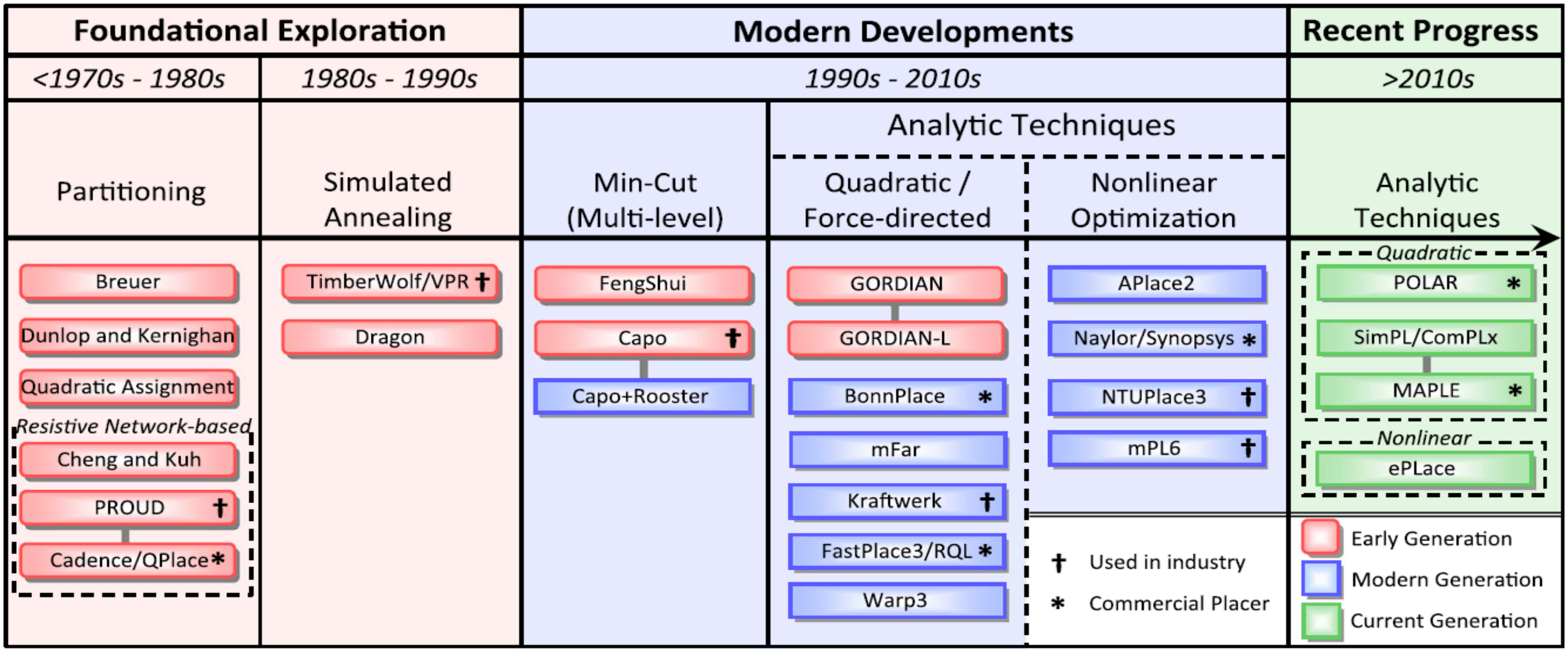
- Comparisons among academic analytical placers

Placer	Wirelength Model	Overlap Reduction	Integration	Optimization
APlace	LSE	Bin Density	Penalty Method	Nonlinear
BonnPlace	Quadratic	Partitioning	Region Constraint	Quadratic
DPlace	Quadratic	Diffusion	Fixed Point	Quadratic
ePlace	WA	Density (electrostatics)	Penalty Method	Nonlinear (Nesterov)
FastPlace	Quadratic	Cell Shifting	Fixed Point	Quadratic
FDP	Quadratic	Density	Fixed Point	Quadratic
Gordian	Quadratic	Partitioning	Region Constraint	Quadratic
hATP	Quadratic	Partitioning	Region Constraint	Quadratic
Kraftwerk2	Bound2Bound	Bin Density	Fixed Point	Quadratic
mFAR	Quadratic	Density	Fixed Point	Quadratic
mPL6	LSE	Bin Density	Penalty Method	Nonlinear
NTUplace3/4	**WA/LSE**	**Bin Density**	**Penalty Method**	**Nonlinear** (gradient)
Ripple	Bound2Bound	Partitioning	Penalty Method	Quadratic
RQL	Quadratic	Cell Shifting	Fixed Point	Quadratic
SimPL	Bound2Bound	Partitioning	Penalty Method	Quadratic
Vassu	LSE	Assignment	Fixed Point	Nonlinear

Source: “Modern Circuit Placement” by Y.-W. Chang at 2015 VLSI Design/CAD Symp.

Summary (3/10)

- Historical development of global placement algorithms and implementations in academia [IEEE 2015]



Summary (4/10)

- Legalization and detailed placement [IEEE 2015]

STAGE	TECHNIQUE
LEGALIZATION	greedy moves to free locations [39], [73], [74], [180], [197]
	ripple cell movement [87]
	diffusion PDE [161]
	dynamic programming [4], [96], [180]
	computational geometry [132]
	network flow [12], [15], [44], [58], [59]
	linear programming [55]
	top-down opt. & clustering [74], [116]
DETAILED PLACEMENT	branch-and-bound [18], [24]
	network flow [58]
	simulated annealing [172]
	mixed ILP [25], [120]
	single-row optimization [14], [98]
	cell-to-slot matching [39]
	cell swapping [55], [149]
	clustering [87], [149]
	dynamic programming [85]
	global-placer integration [166]

Summary (5/10)

- Mixed-size placement [IEEE 2015]

FLOW	TECHNIQUE
SIMULTANEOUS	macro shredding [17], [106], [164]
	macro or cell shifting [39], [107], [197]
	iterative re-legalization [27], [56]
	top-down legalization [27], [55], [164], [167]
	force-directed / non-convex optimization [27], [39], [62], [76], [99], [106], [107], [195], [197]
	floorplacement [45], [164], [167]
SEQUENTIAL	periphery macro packing [38]
	macro shredding [2]
	separate floorplanning & placement steps [2, Flow 1], [35], [38], [216]
	simulated annealing [125], [135]
POST-PROCESS	floorplan repair [139]
	linear programming [17], [55], [181]
	force-directed optimization [2, Flow 2]

Summary (6/10)

- Congestion estimation for placement [IEEE 2015]

APPROACH	TECHNIQUE
STATIC	net bounding box [22], [91]
	Steiner trees [165]
	pin density [13], [223]
	counting nets in regions [203]
PROBABILISTIC	L-shaped routes [71], [72], [78], [179]
	smoothened wire density [189]
	pattern routing [209]
CONSTRUCTIVE	using A*-search [210]
	using a global router: • FastRoute [214] in IPR [49] and POLAR [123] • BFG-R [83] in SimPLR [104], CoPR [84] • NCTUgr [128] in Ripple 2.0 [72]

Summary (7/10)

- Routability-driven placement [IEEE 2015]

PLACEMENT PHASE	TECHNIQUE
GLOBAL PLACEMENT	relocating movable objects: • moving nets [71], [91] • modifying forces [51], [137], [179] • incorporating congestion in objective function [78], [189] • adjusting target density [104]
	cell bloating [13], [71], [72], [75], [84], [104], [127]
	macro porosity [78], [91]
	pin density control [78]
	expanding/shrinking placement regions [154]
INTERMEDIATE	local placement refinement [49]
LEGALIZATION AND DETAILED PLACEMENT	linear placement in small windows [90], [165]
	congestion embedded in objective function [222]
	cell swapping [49], [71], [104]
	cell shifting [57], [78]
POST PLACEMENT	whitespace injection or reallocation [119], [165], [217]
	simulated annealing [40], [79], [200]
	linear programming [122]
	network flows [201], [202]
	shifting modules by expanding gcells [222]
	cell bloating [169]
	cost-based cell relocation [129]

Summary (8/10)

- Timing-driven placement [IEEE 2015]

TECHNIQUE	IMPLEMENTATION
STATIC NET WEIGHTS	slack [20], [28], [61], [111]
	sensitivity [66], [163], [213]
DYNAMIC NET WEIGHTS	incremental timing analysis [20], [160]
	based on previous iterations [62], [160]
NET CONSTRAINTS	in global placement [64], [67], [159], [185], [188]
	in detailed placement [45], [68], [86], [162]
PATH-BASED	partitioning [89]
	simulated annealing [183]
	Lagrangian relaxation [69], [182]
	differential timing analysis [48]
	local movement transforms [33], [136]
	force-directed [138]
COMPOUND	net weights & constraints [130]

Summary (9/10)

- Power-driven placement

POWER TYPE	TECHNIQUE
STATIC	multiple supply voltages [82], [113], [124], [157], [218]
	logic and physical adjacency [158]
DYNAMIC	weights on signal nets [42], [143], [171], [190]
	register clustering [32], [42], [92], [176]
	explicit register relocation [43], [133], [153]
	clock-tree co-synthesis [50], [115], [153], [176], [204]

Summary (10/10)

- Placement-aware physical synthesis [IEEE 2015]

TECHNIQUE	IMPLEMENTATION
LOGIC TRANSFORMATIONS	fanin restructuring [208], [212]
	cloning and decomposition [60]
	simulation-driven restructuring based on don't-cares [155]
INTERCONNECT BUFFERING	fanout restructuring [60]
	delay-optimal [152], [174]
	Steiner tree construction [5]
GATE SIZING	discrete [81], [147], [151]
	continuous [184], [198]
COMPOUND TRANSFORMATIONS	area management [118]
	retiming-based physical synth. [150]
	design flows [151], [153], [186], [198]