UTPlaceF: A Routability-Driven FPGA Placer with Physical and Congestion Aware Packing

Wuxi Li, Shounak Dhar, David Z. Pan
ECE Department, University of Texas at Austin

Outline

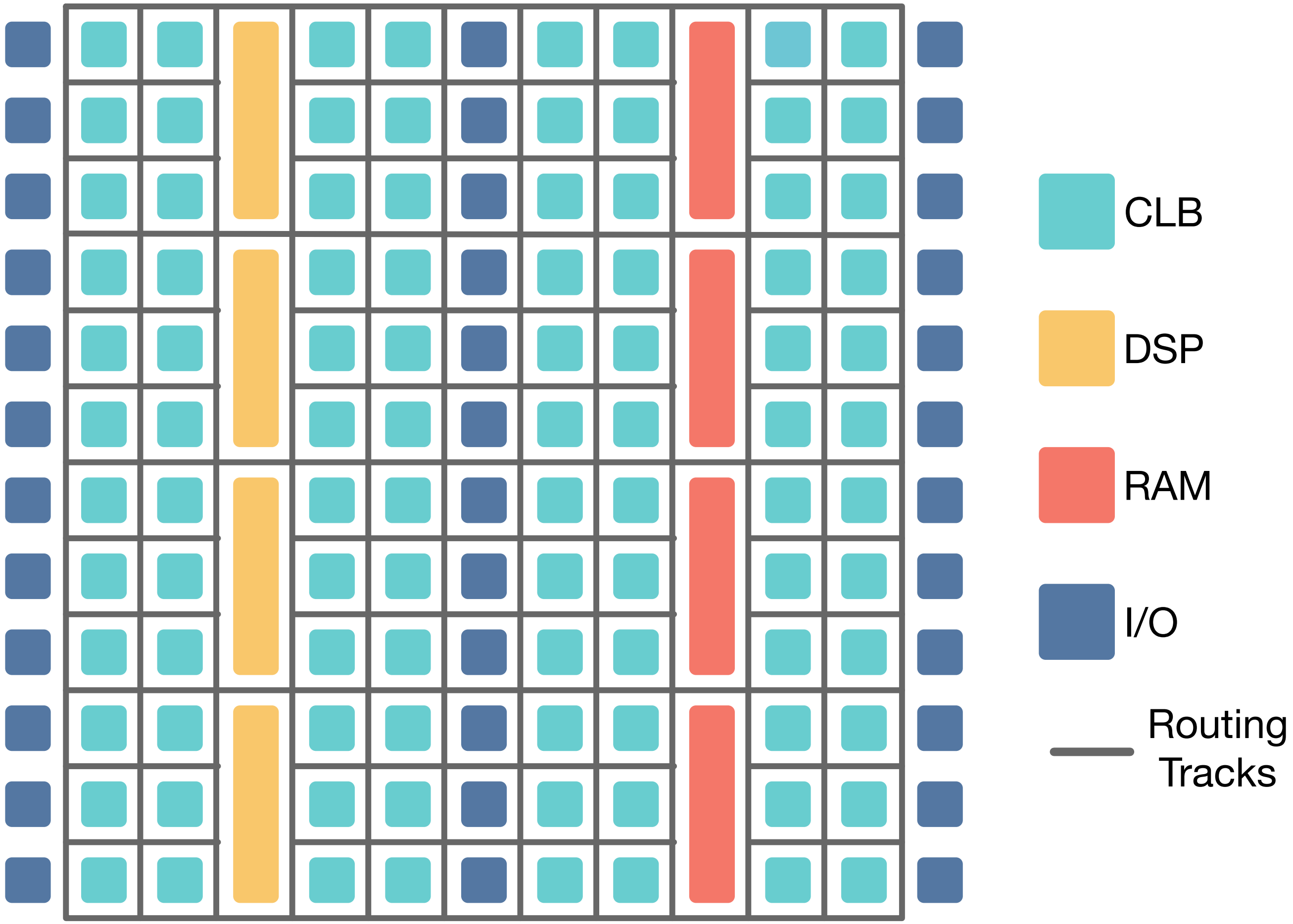
- Introduction
- Previous Work
- UTPlaceF Algorithms
- Experimental Results
- Conclusion

Introduction

- Field programmable gate array (FPGA)
  - Pre-manufactured integrated circuit
  - Reconfigurable
- FPGAs are becoming popular
  - Lower development costs
  - Re-programming in the field
  - Shorter time to market
  - Gate count has reached scale of millions
- More digital systems are moving towards FPGAs
  - ASIC emulators
  - Highly customizable SoCs
  - Hardware acceleration

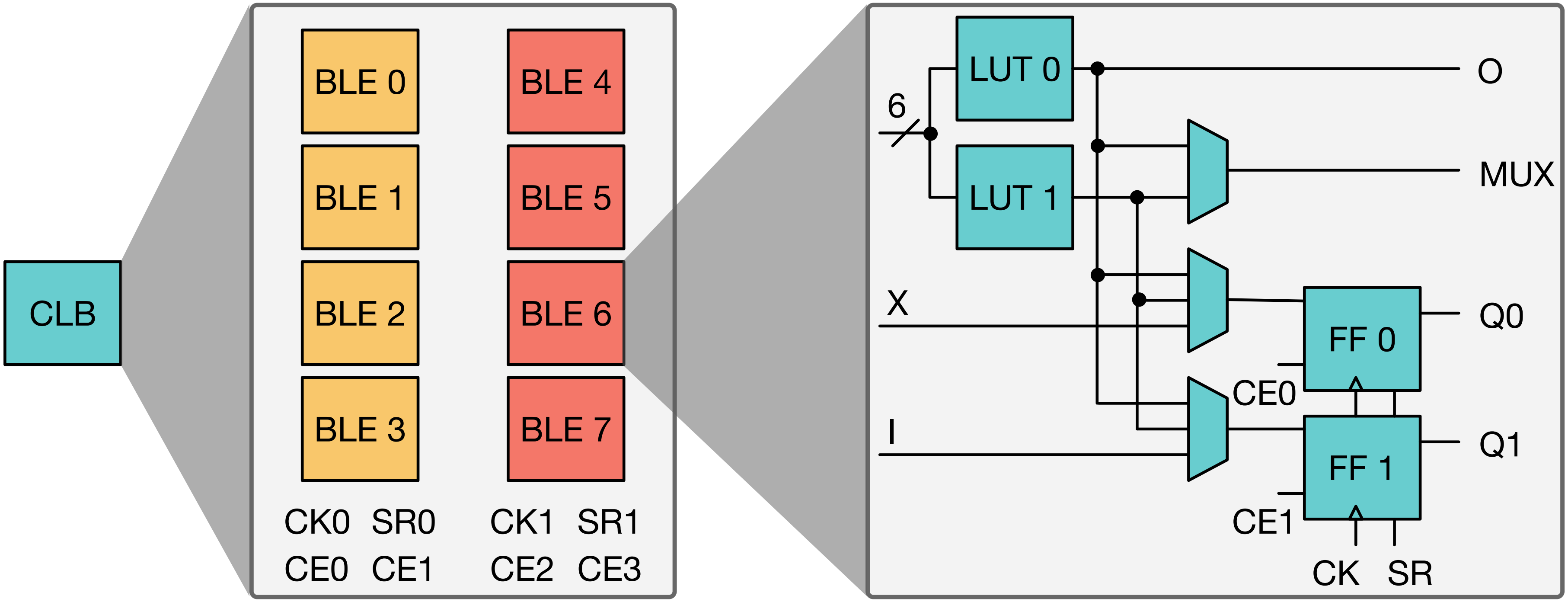
FPGA Architecture

- Configurable logic block (CLB)
  - Can be configured as some combinational/sequential logic
- DSP, RAM, I/O



CLB Architecture

- ISPD'16 contest FPGA architecture: Xilinx UltraScale
- Capacity constraint
- Input-pin constraint
- Control signal constraint

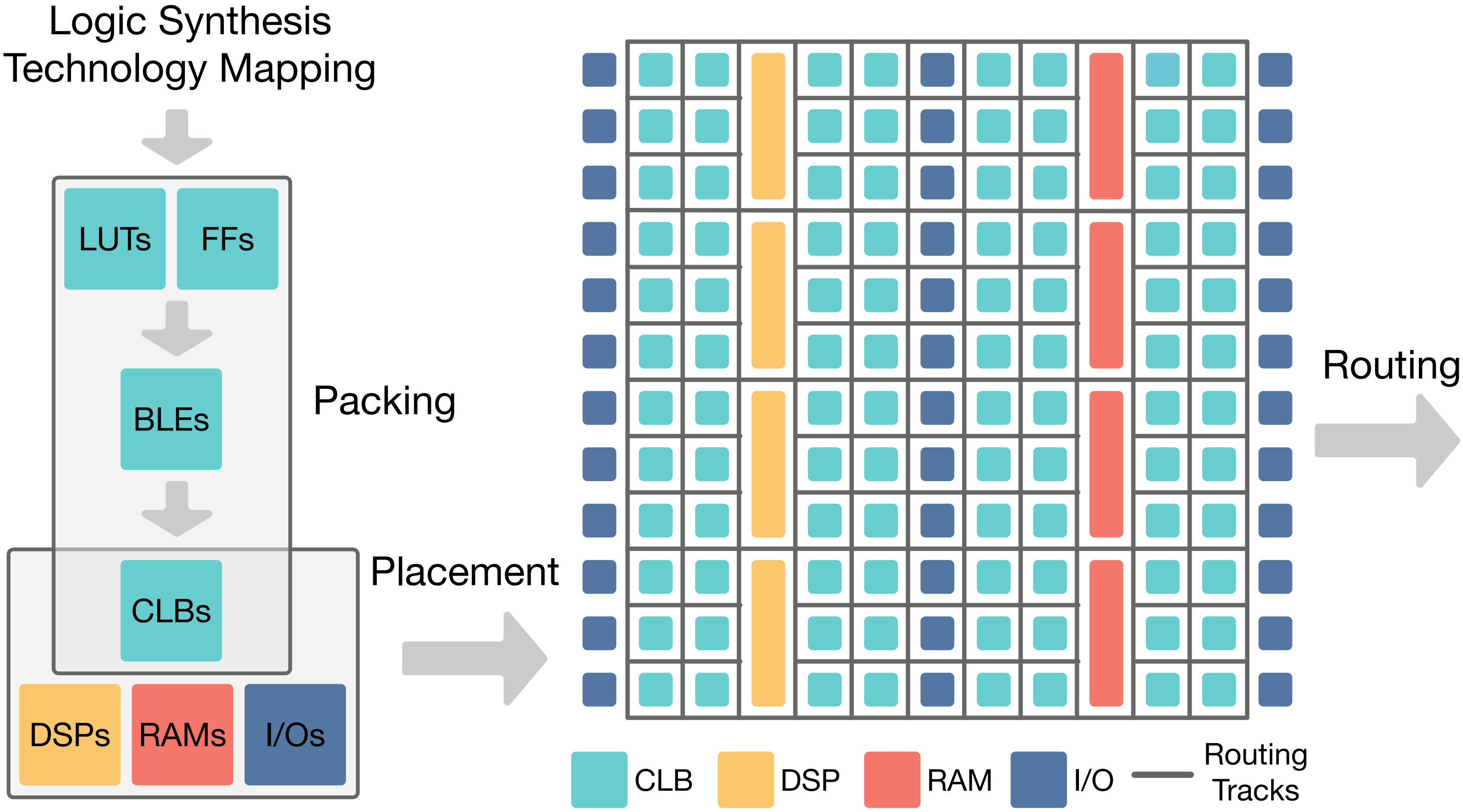


Configurable logic Block (CLB)

Basic logic element (BLE)

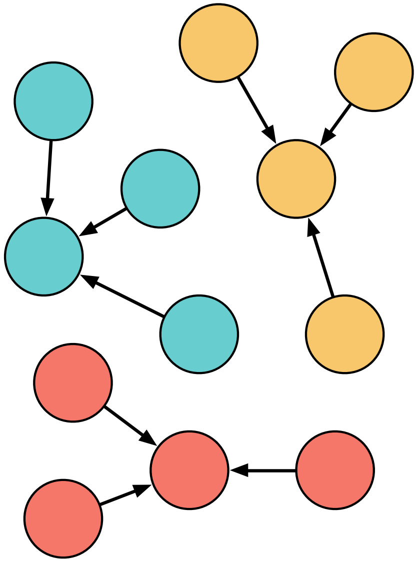
FPGA CAD Flow

- Packing: Cluster LUT/FF to CLBs
- Placement: Determine cell locations

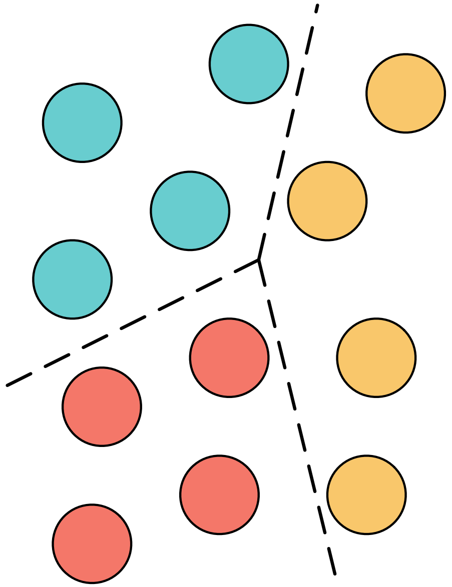


Previous Packing Algorithms

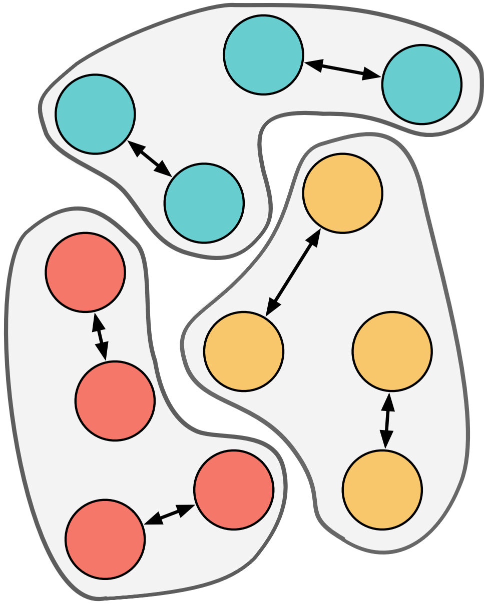
- Seed-based
  - [T-VPack, FPGA’99], [Rpack, ASPDAC’01], [iRAC, TODAES’02], [MO-Pack, DAC’11], [T-NDPack, IJRC’10]
- Partitioning-based
  - [Marrakchi+, ReConfig’05], [PPack, FPT’12]
- Cluster-merging-based
  - [HDPack, FPL’07]



Seed-based



Partition-based



Cluster-merging-based

Previous Placement Algorithms

- Similar to ASIC placement
- Simulated Annealing
  - [VPR, FPL’97]
- Min-cut-partitioning based
  - [Maidee+, DAC’03, TCAD’05]
- Analytical placement
  - [Xu+, FPL’05], [Gopalakrishnan+, DAC’06], [Gort+, FPL’12], [Lin+, DAC’13], [Chen+, ICCAD’14]

Major Contributions of This Work

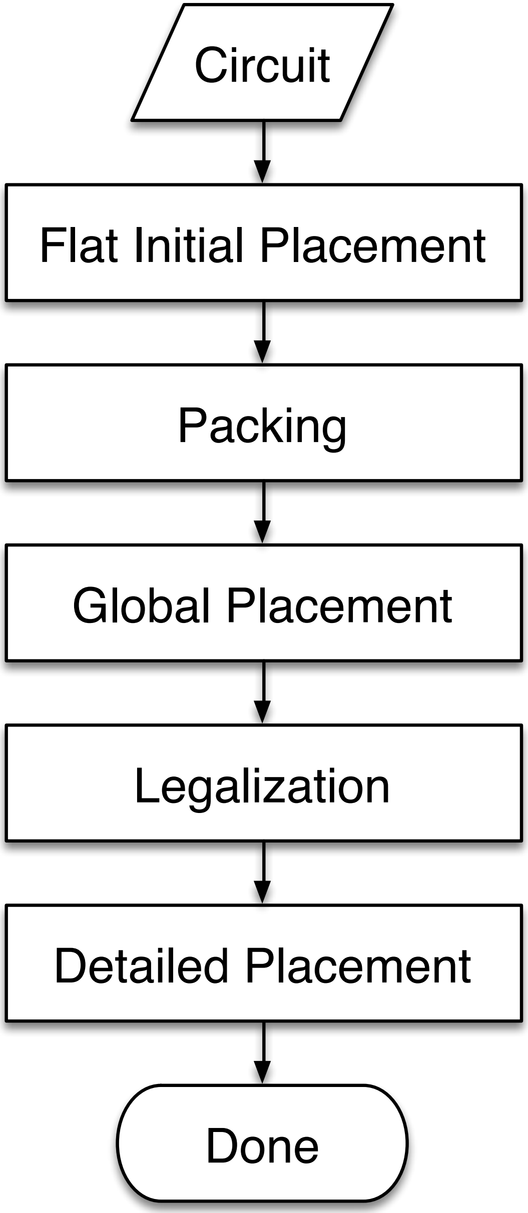
- A novel physical and congestion aware packing algorithm guided by a high-quality analytical flat initial placement
- Congestion aware detailed placement techniques to improve wirelength without routability degradation.
- Won the first-place prize of ISPD’16 contest.

Problem Formulation

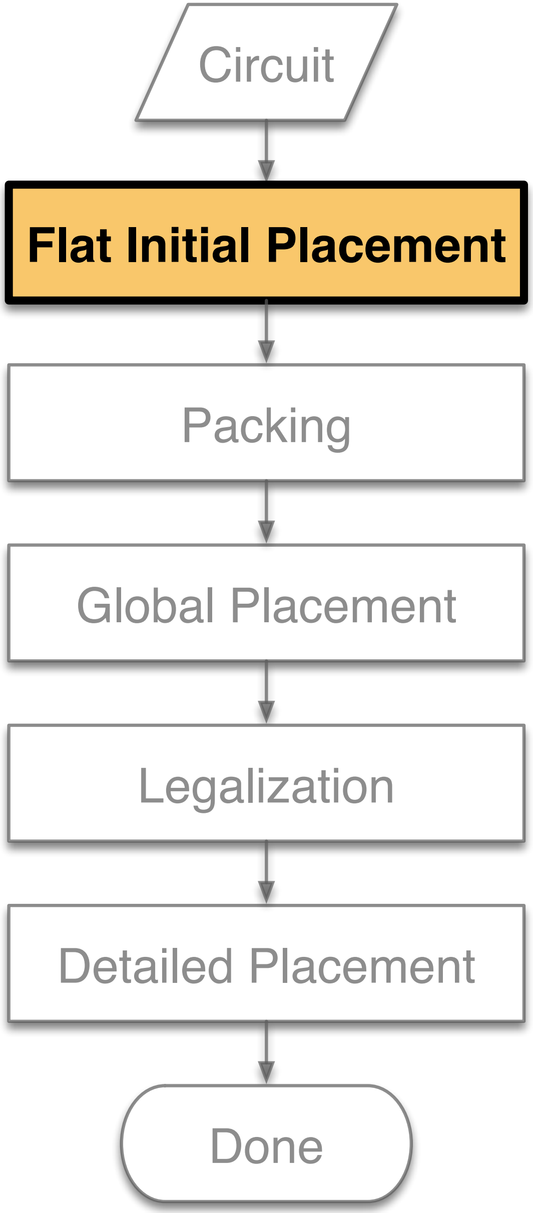
- Input
  - A netlist of LUTs, FFs, DSPs, RAMs, and I/Os
  - CLB architecture constraints
  - Placement regions

- Output
  - A placement with optimized wirelength and routability
  - All architecture constraints are satisfied

UTPlaceF Overall Flow



Flat Initial Placement (FIP)



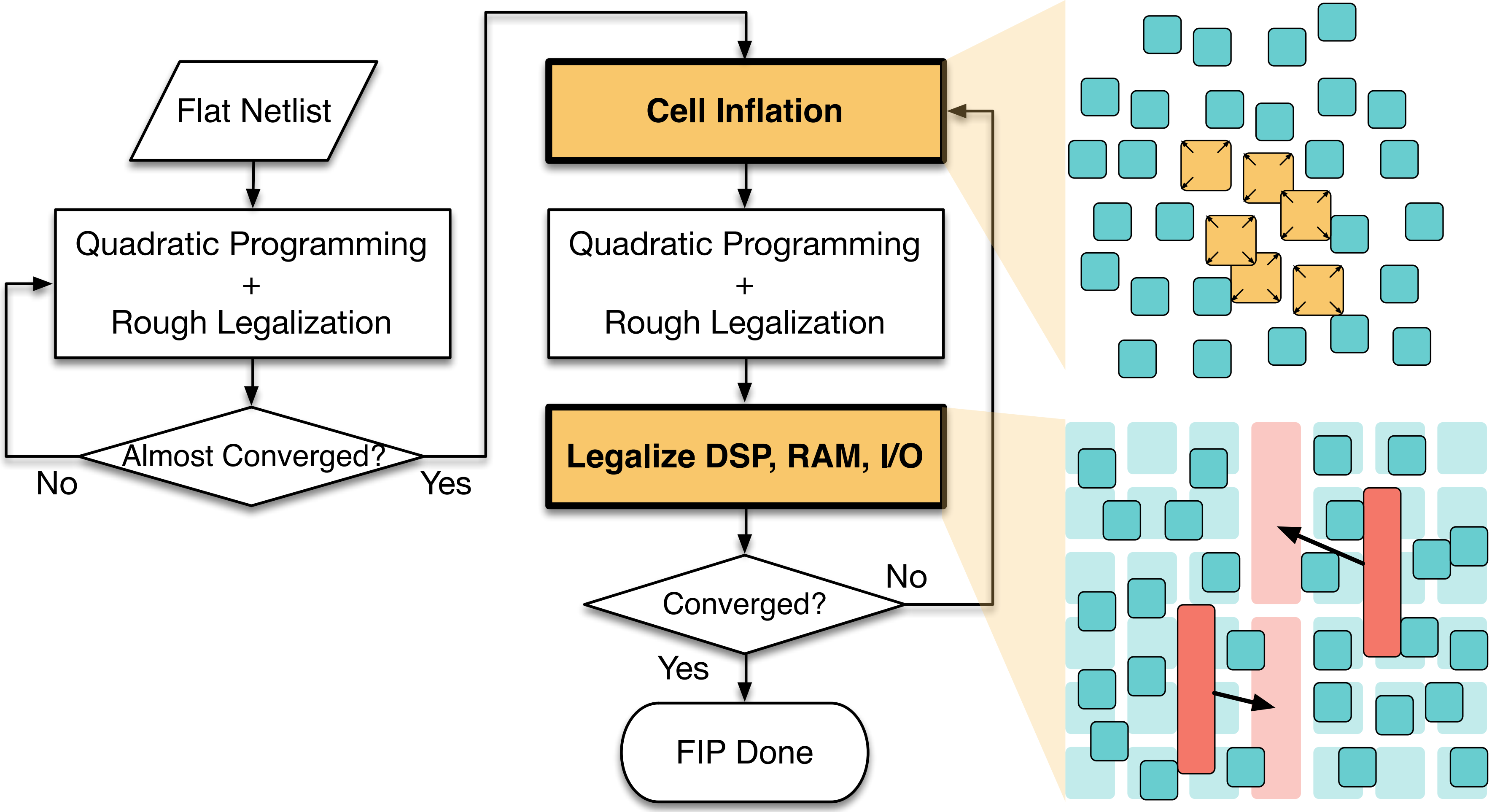
Flat Initial Placement (FIP)

- Routability-driven quadratic placement
  - Adopt the main framework of [POLAR 2.0, DAC'14]
- Objectives
  - Generate physical locations for all cells (LUT / FF / DSP / RAM / IO) in the flat netlist
  - Detect LUTs and FFs that are in routing congested regions
- Guide packing using physical and congestion information

FIP Flow

Wirelength-driven Phase

Routability-driven Phase



Flat Netlist

Quadratic Programming + Rough Legalization

Almost Converged?

No

Yes

Cell Inflation

Quadratic Programming + Rough Legalization

Legalize DSP, RAM, I/O

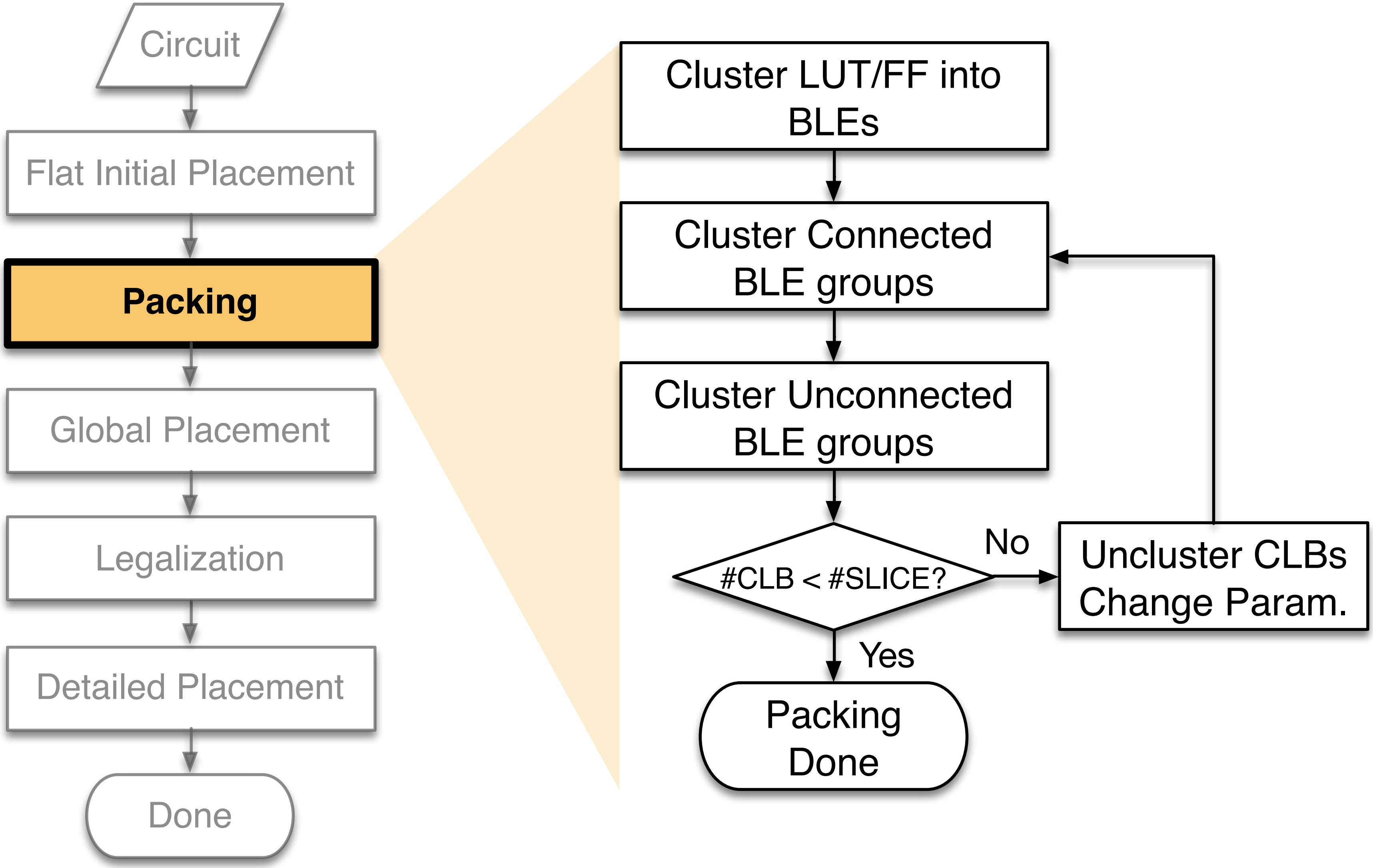
Converged?

No

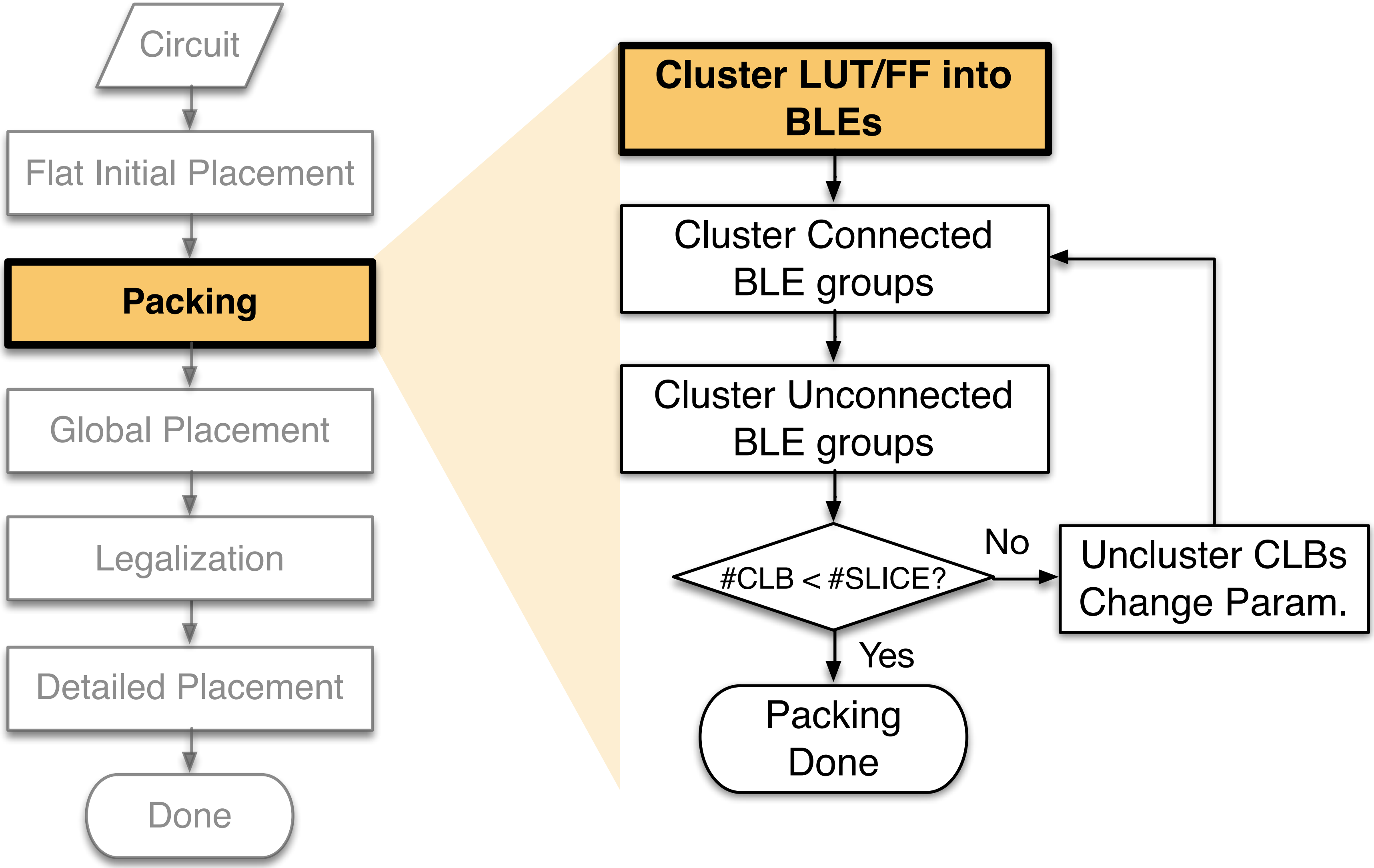
Yes

FIP Done

Packing Flow

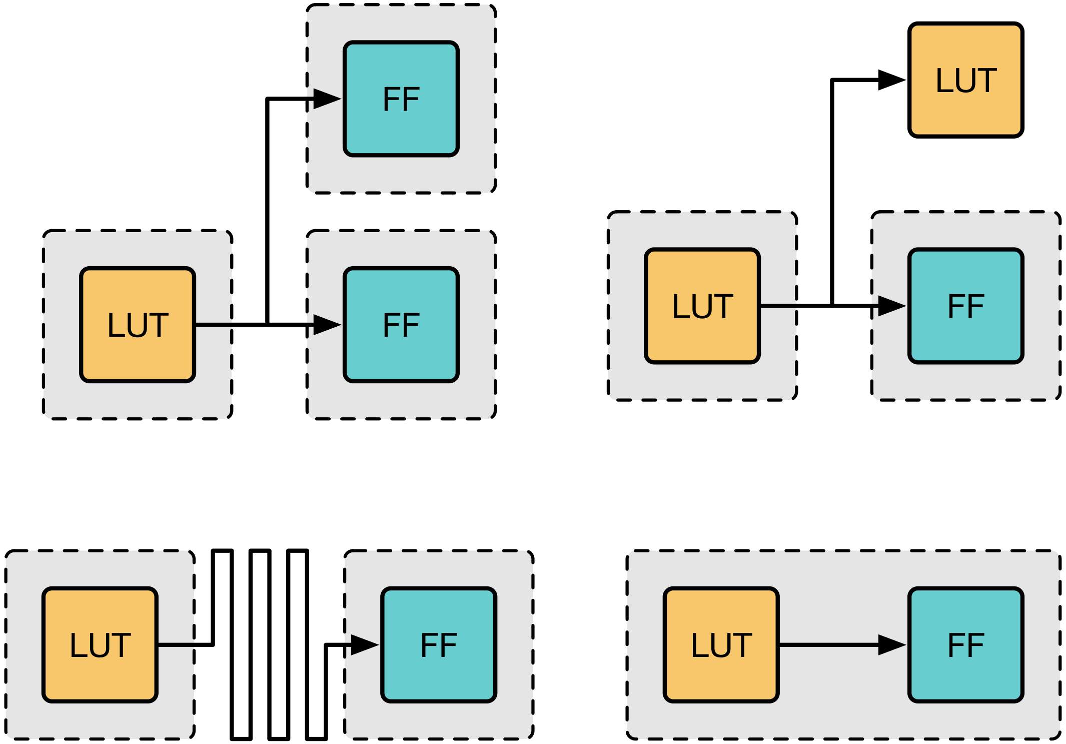


Cluster LUT/FF into BLEs



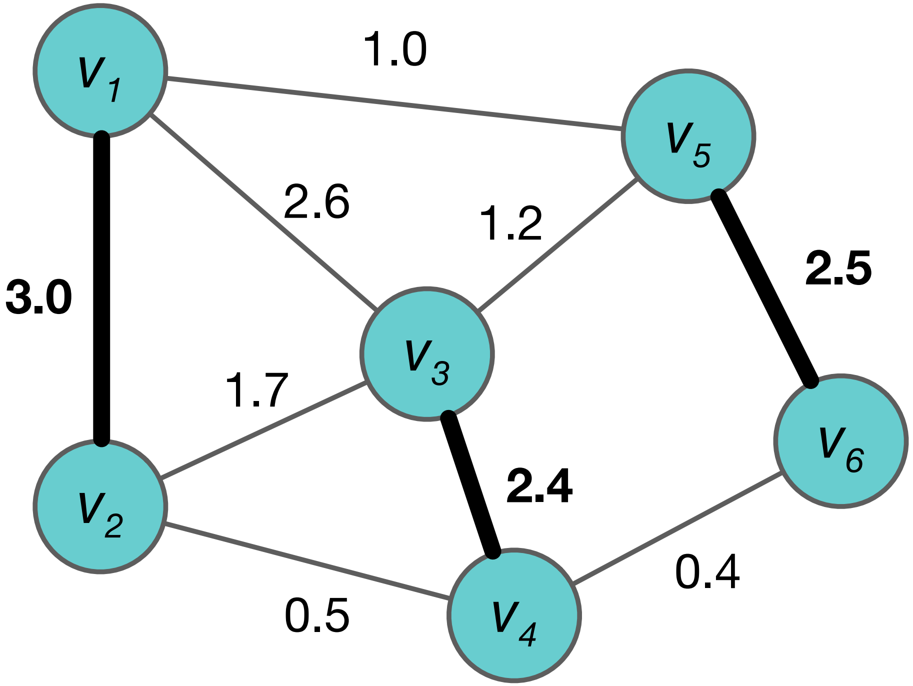
Cluster LUT/FF into BLEs: Pair LUT/FF

- Group a LUT and a FF together if the LUT only fanouts to the FF
- Reject long-distance pairing



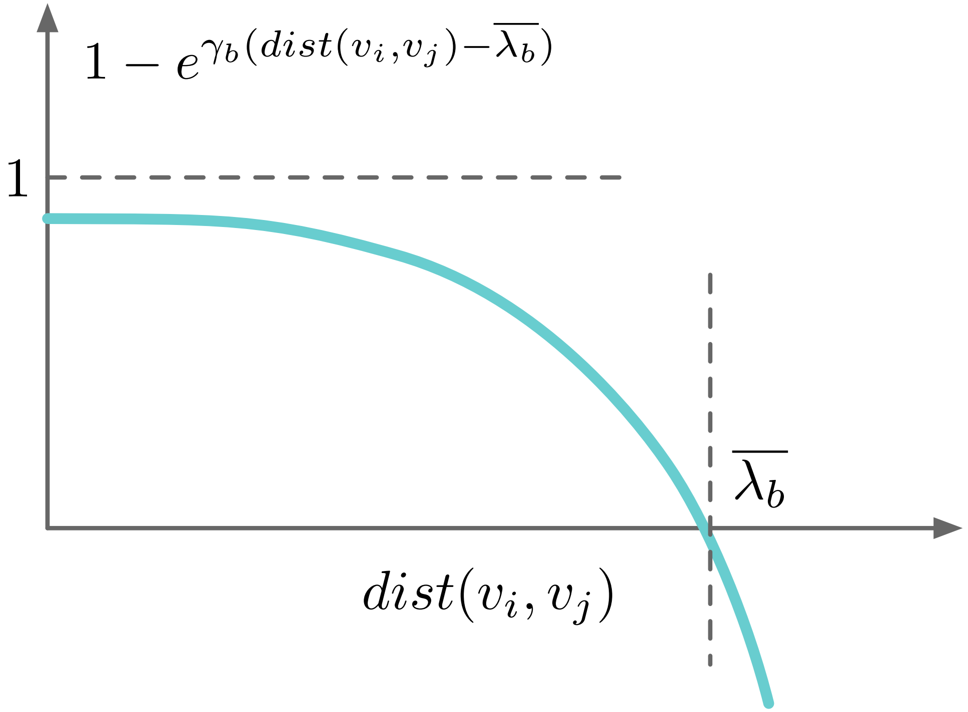
Cluster LUT/FF into BLEs: Max-Weighted Matching

- Construct an undirected weighted graph
  - Vertices: Paired/unpaired LUT/FF
  - Edge weight = $\phi_b(v_i, v_j)$
- Call max-weighted matching

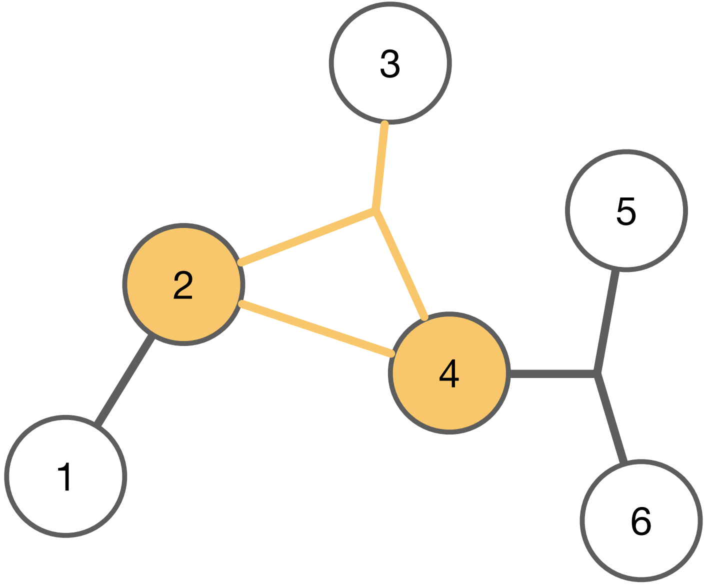


Cluster LUT/FF into BLEs: Attraction Function

$$\phi_b(v_i, v_j) = \underbrace{\left(1 - e^{\gamma_b(dist(v_i,v_j) - \overline{\lambda_b})}\right)}_{\text{Distance Term}} \underbrace{\sum_{p \in Net(v_i \cap v_j)} \frac{k_b}{deg(p) - 1}}_{\text{Connectivity Term}}$$

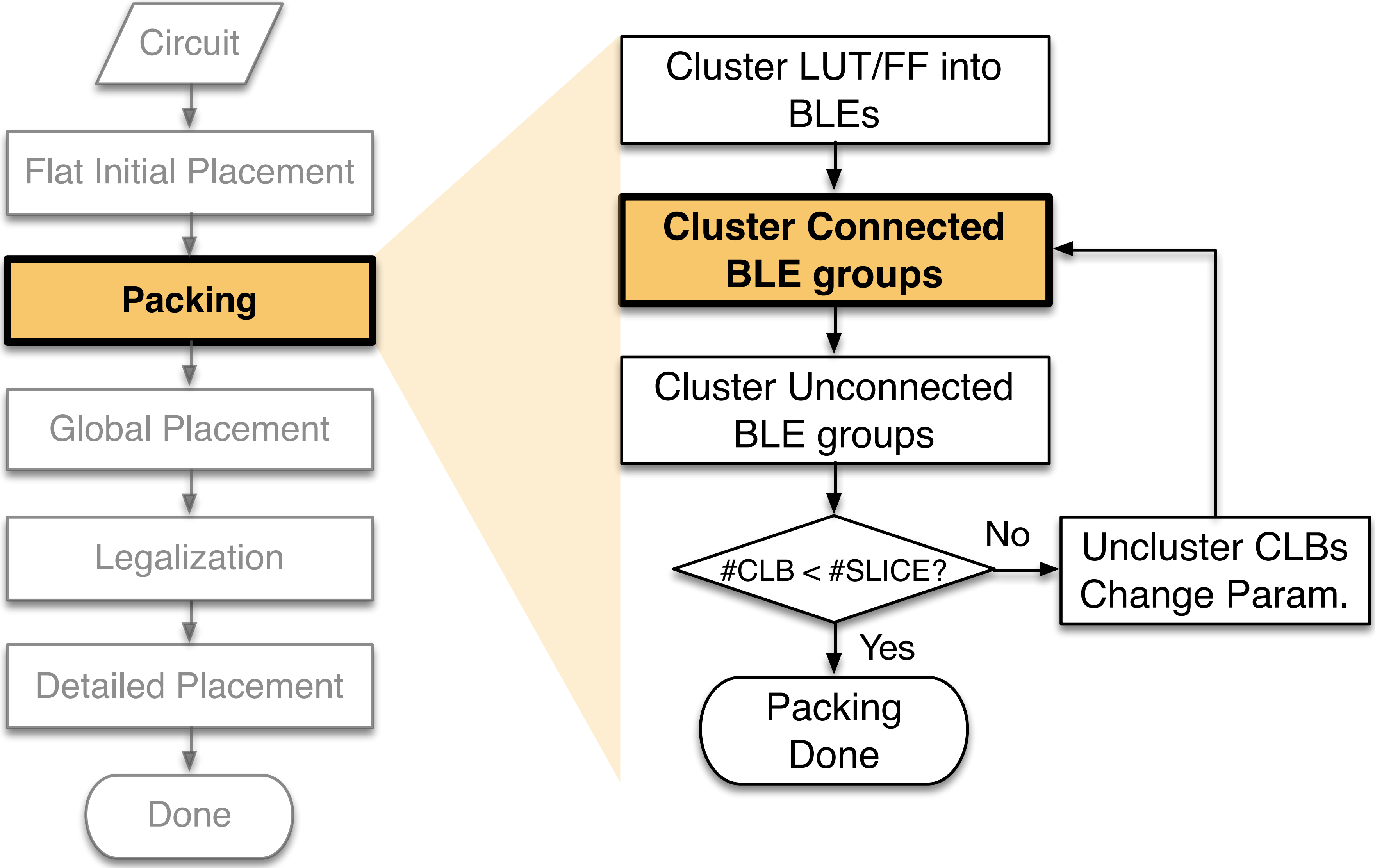


$$k_b = \begin{cases} 2, & \text{for 2-pin nets} \\ 1, & \text{otherwise} \end{cases}$$



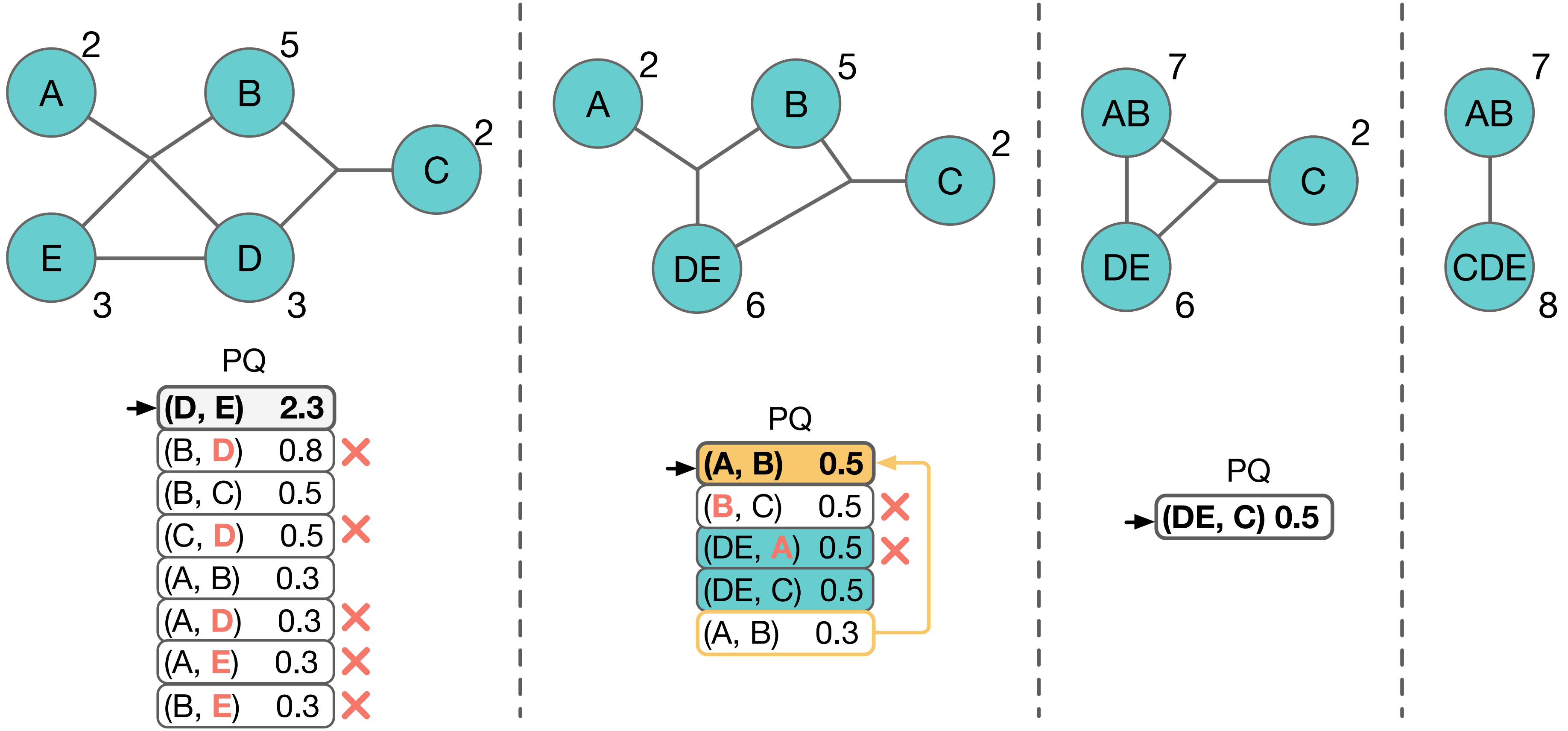
$$\sum_{p \in Net(v_i \cap v_j)} \frac{k_b}{deg(p) - 1} = \frac{2}{2-1} + \frac{1}{3-1} = 2.5$$

Cluster Connected BLEs



Cluster Connected BLEs

- Adopt BestChoice Clustering [Nam+, TCAD'06]
- Maintain a priority queue (PQ), and iteratively cluster the top
- Attraction function is similar to LUT/FF clustering



PQ

(D, E)	**2.3**	
(B, D)	0.8	✗
(B, C)	0.5	
(C, D)	0.5	✗
(A, B)	0.3	
(A, D)	0.3	✗
(A, E)	0.3	✗
(B, E)	0.3	✗

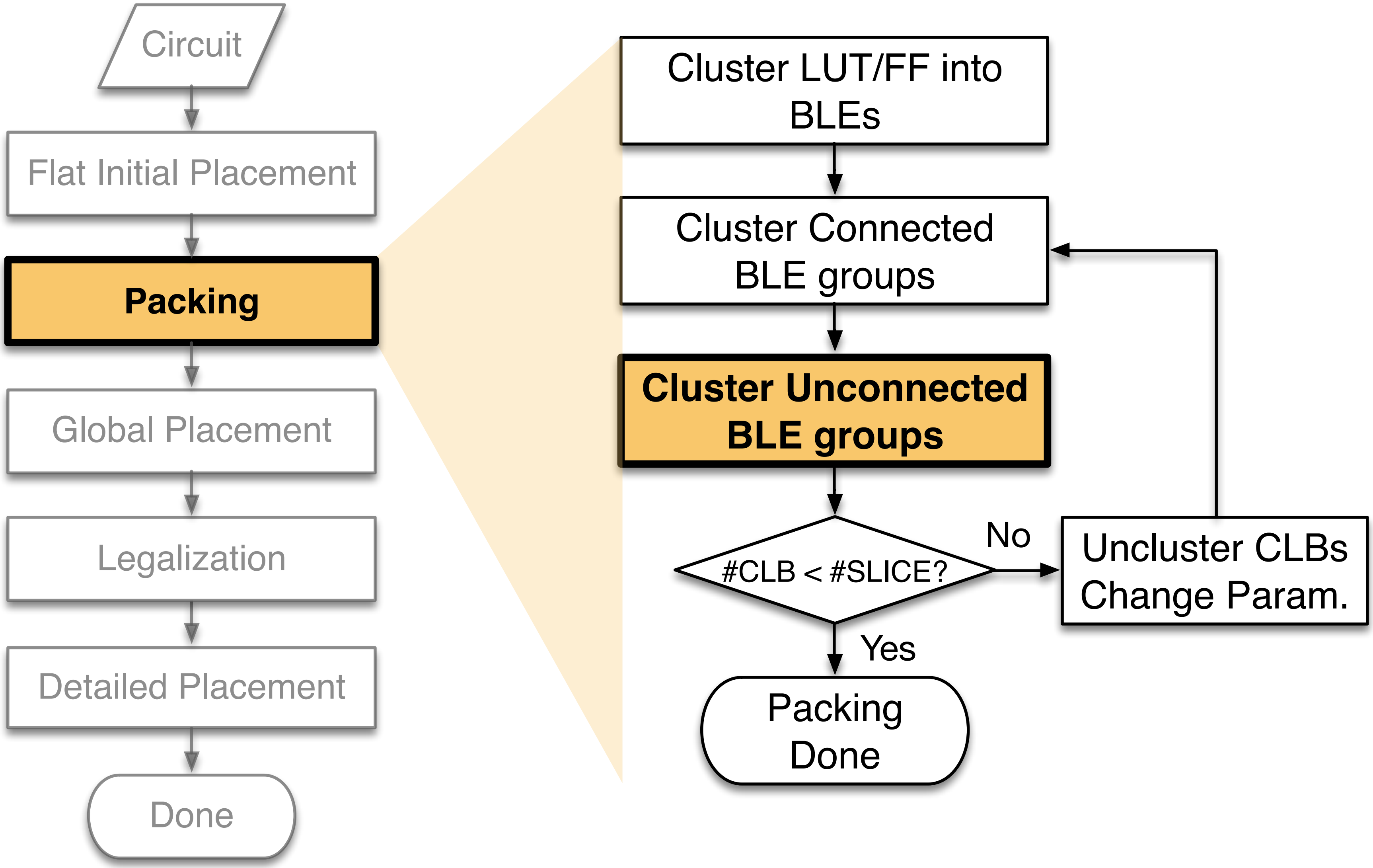
PQ

(A, B)	**0.5**	
(B, C)	0.5	✗
(DE, A)	0.5	✗
(DE, C)	0.5	
(A, B)	0.3	

PQ

(DE, C) 0.5

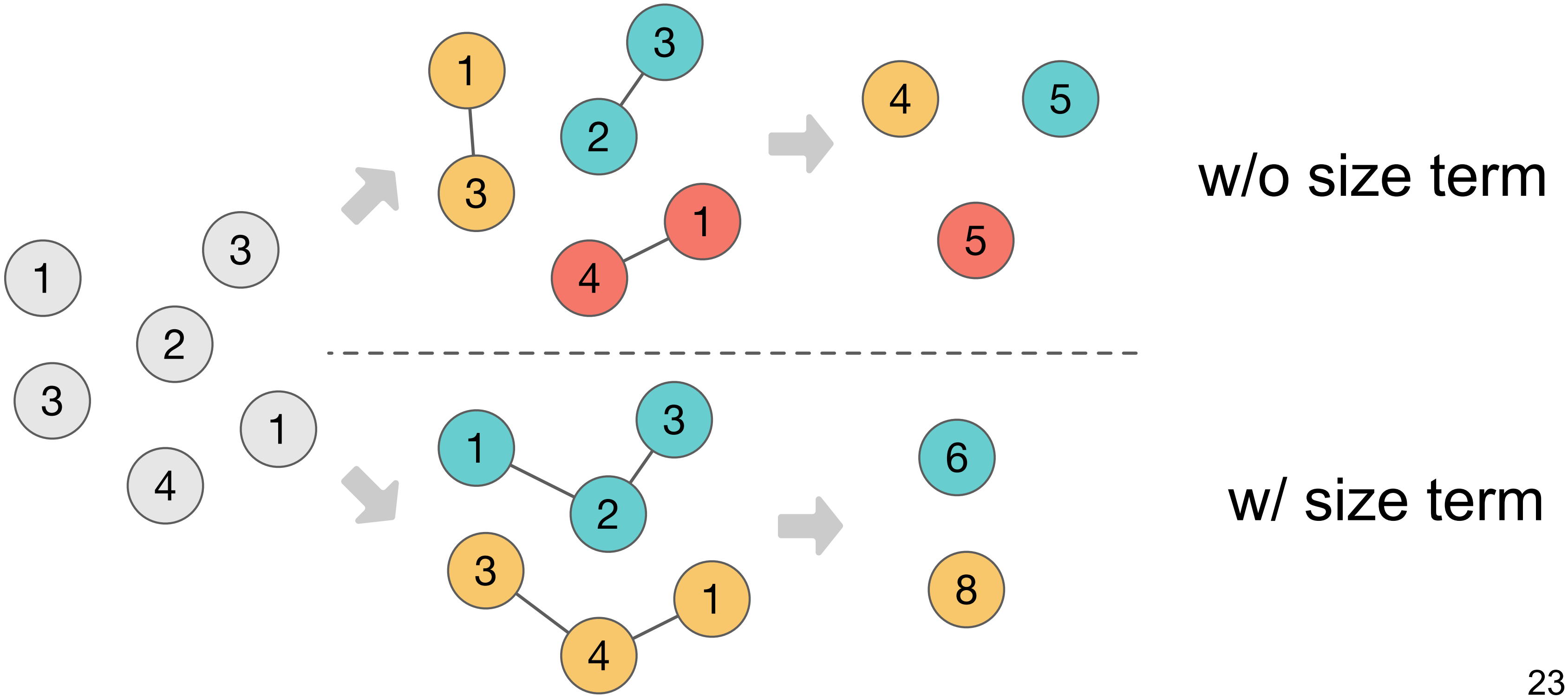
Cluster Unconnected BLEs



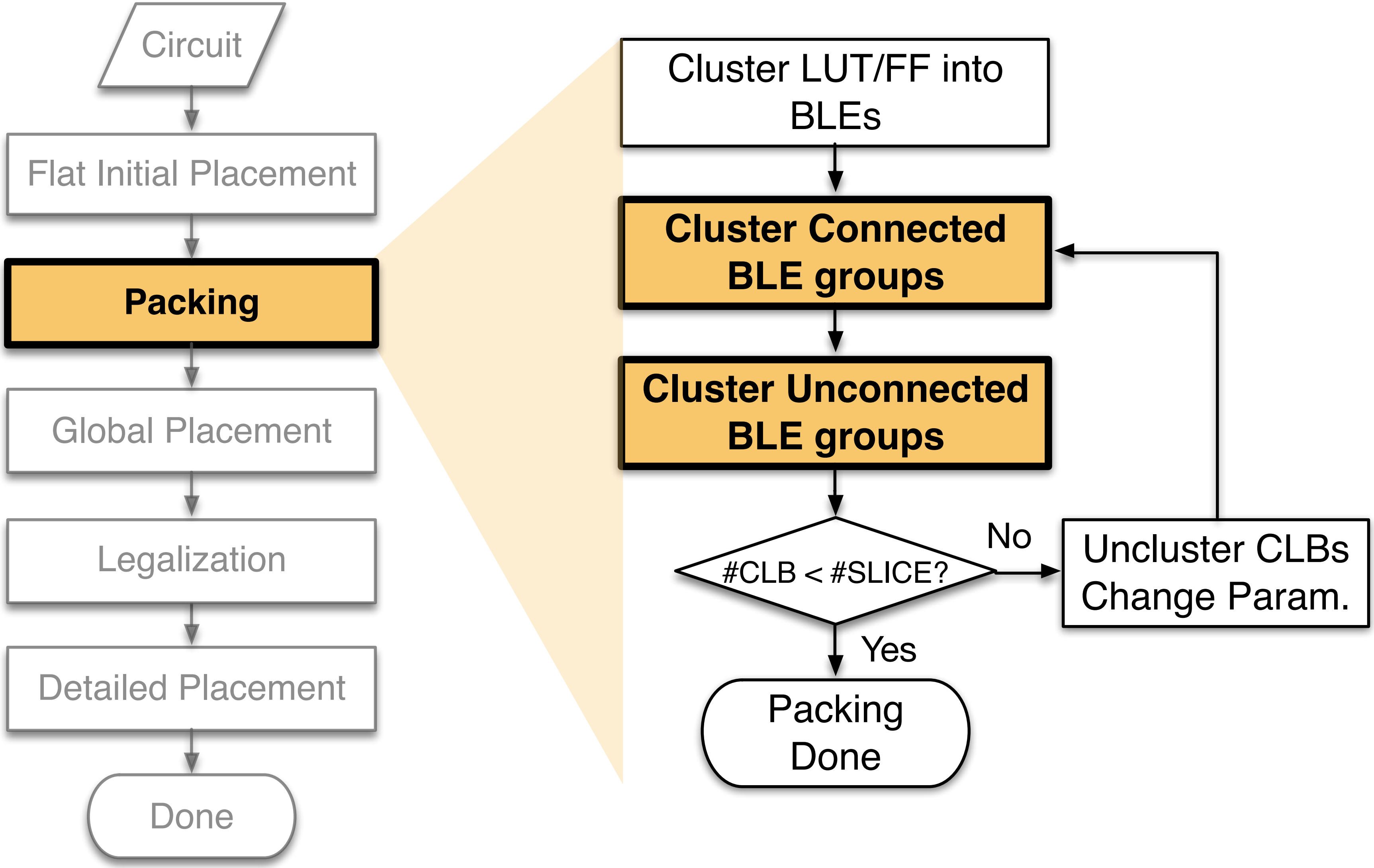
Cluster Unconnected BLEs

- Merge isolated BLE groups to further reduce #CLBs
- BestChoice clustering
- Give mergings that result in larger CLBs higher weight

$$\phi_{uc}(c_i, c_j) = \underset{\text{Distance Term}}{\left(1 - e^{\gamma_{uc}(dist(c_i,c_j) - \overline{\lambda_{uc}})}\right)} \cdot \underset{\text{Size Term}}{\left(N_{BLE}(c_i) + N_{BLE}(c_j)\right)}$$



Congestion Aware Depopulation



Circuit

Flat Initial Placement

Packing

Global Placement

Legalization

Detailed Placement

Done

Cluster LUT/FF into BLEs

Cluster Connected BLE groups

Cluster Unconnected BLE groups

#CLB < #SLICE?

No

Uncluster CLBs Change Param.

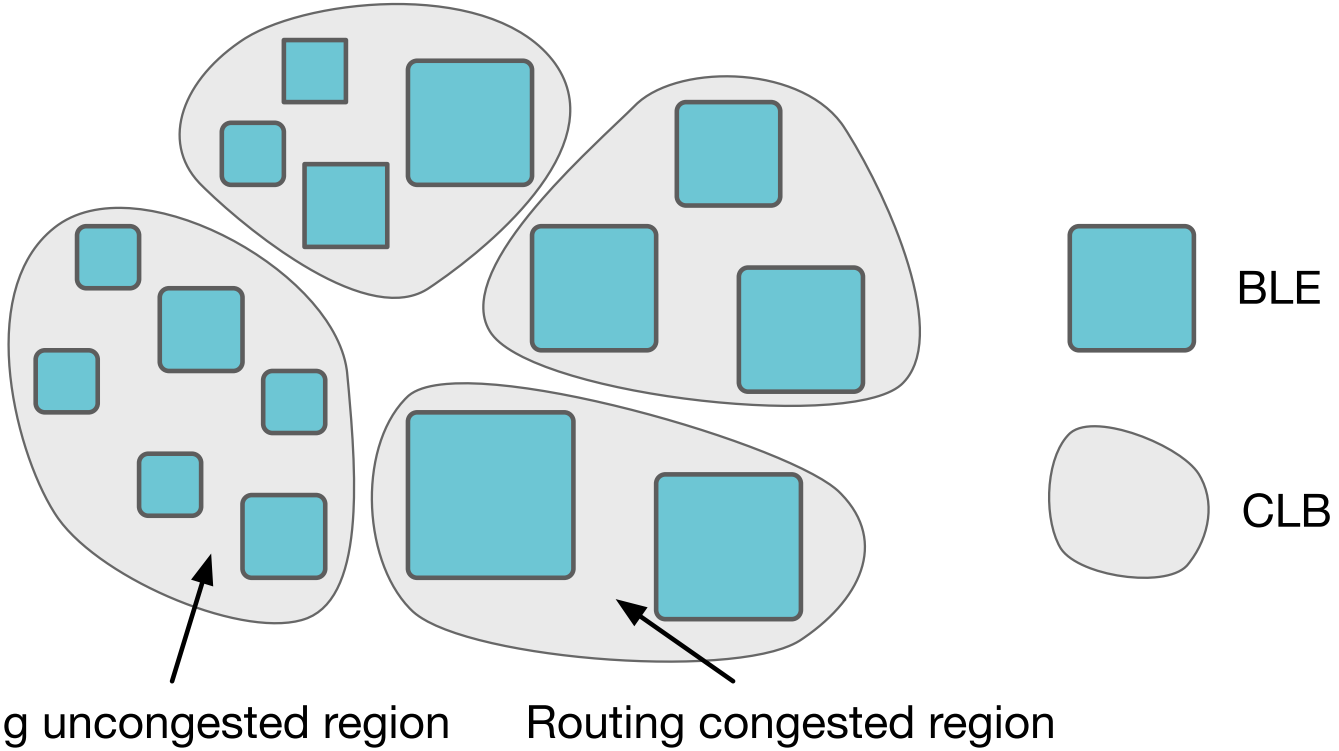
Yes

Packing Done

Congestion Aware Depopulation

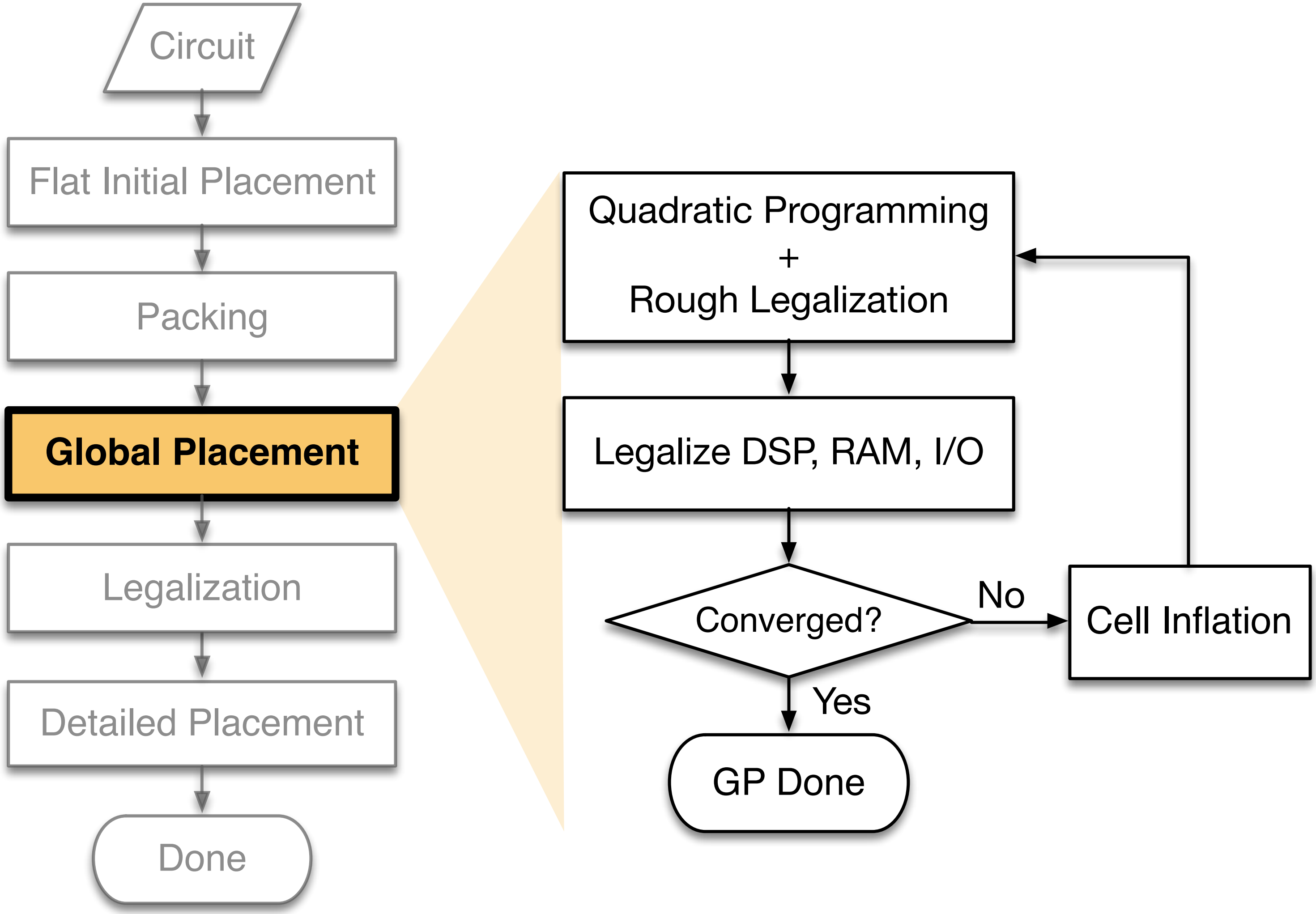
- Avoid over-packing in routing congested regions
- Cell area indicates routing congestion level
- Extra area constraint

$$A_i + A_j \leq \overline{A_c}$$

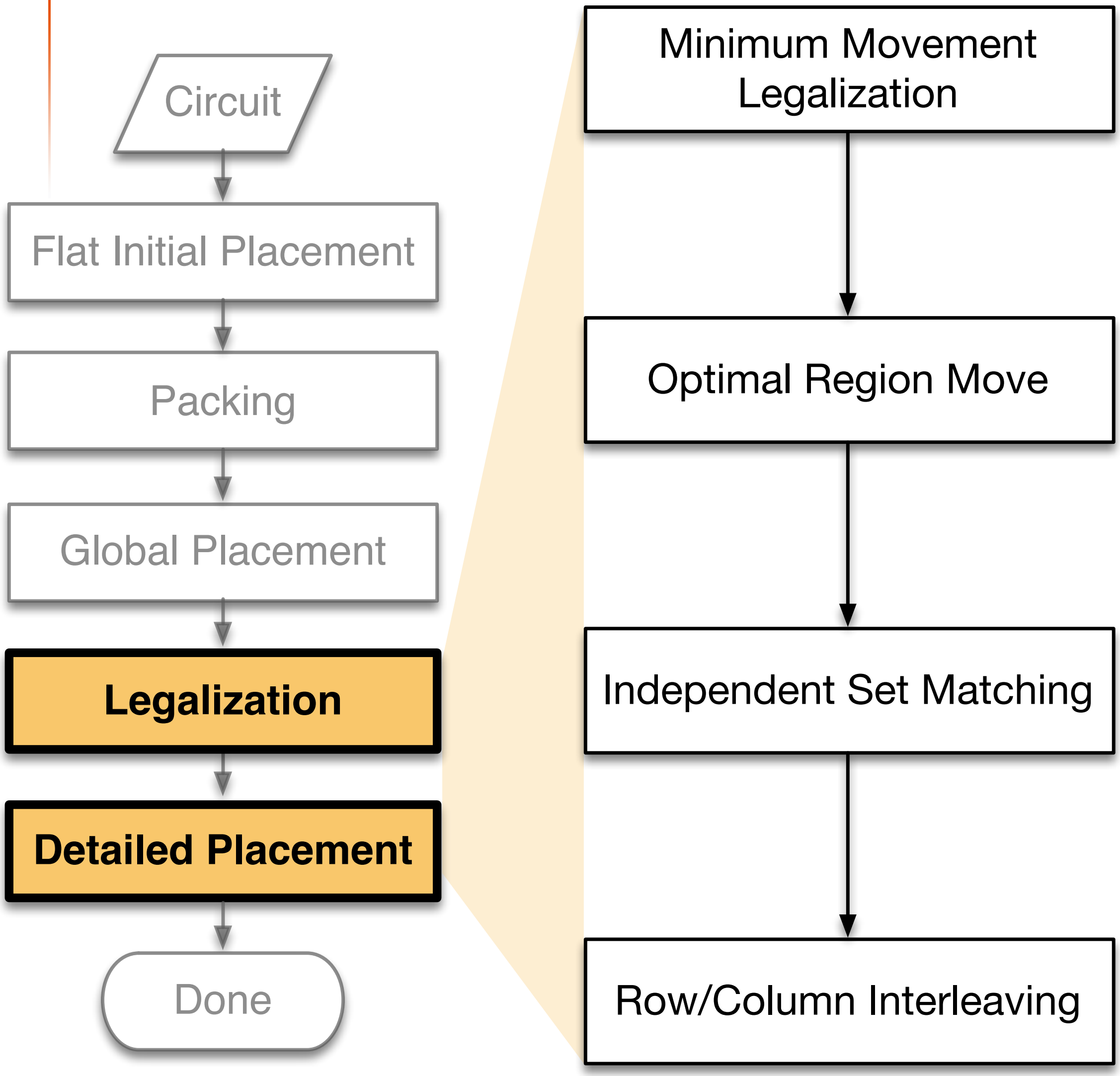


Global Placement

- Similar to FIP
- Place post-packing netlist (CLB, DSP, RAM, IO)



Legalization & Detailed Placement



Circuit

Flat Initial Placement

Packing

Global Placement

Legalization

Detailed Placement

Done

Minimum Movement Legalization

Optimal Region Move

Independent Set Matching

Row/Column Interleaving

Min-cost bipartite matching

[Pan+, ICCAD'05]

- Extend to chain move

[Chen+, TCAD'08]

- Introduced extra constraints to preserve routability

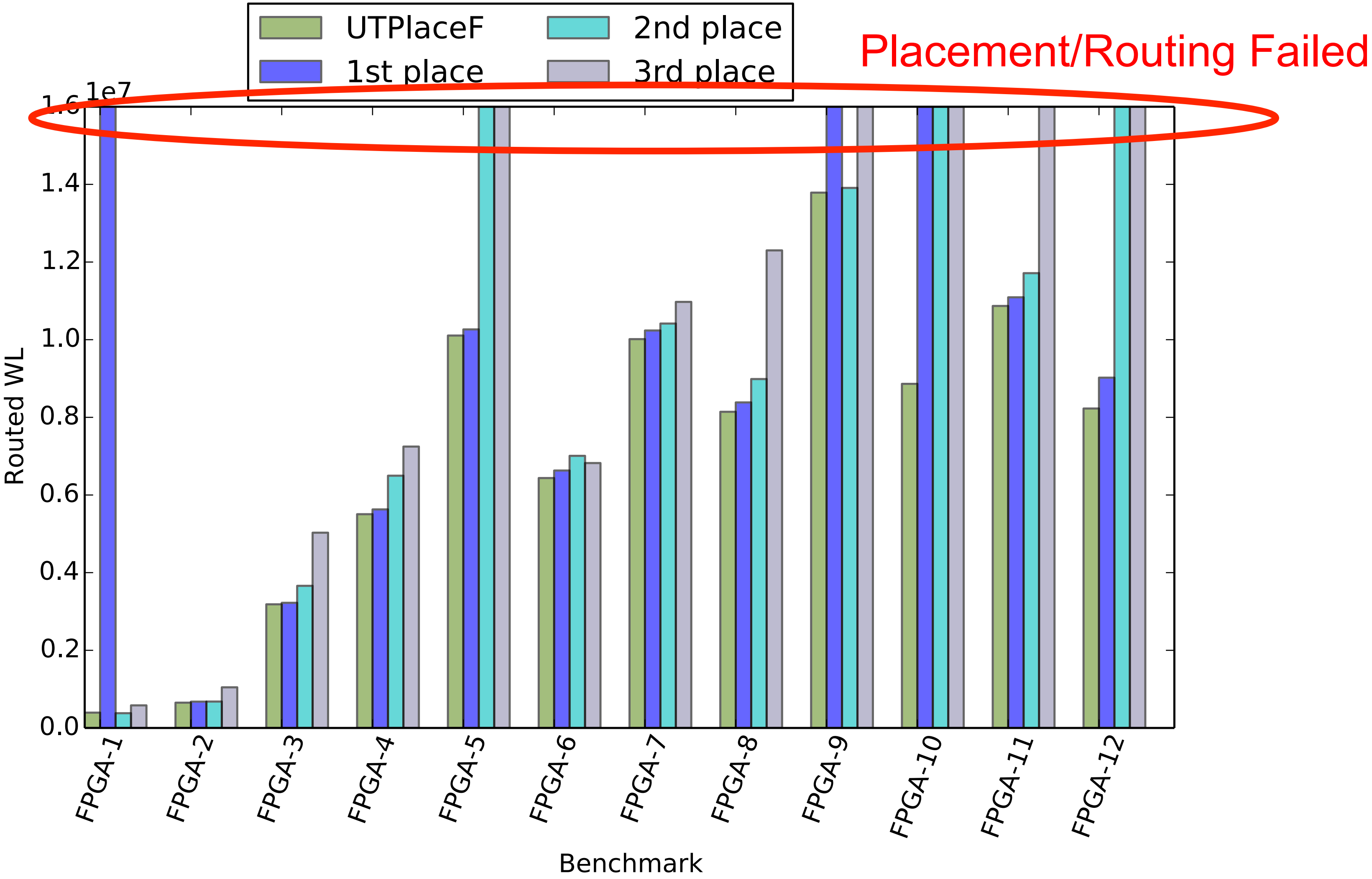
[Hur+, ICCAD'00]

- Consider spaces and
- Fixed cells/spaces in congested regions

Experimental Setup

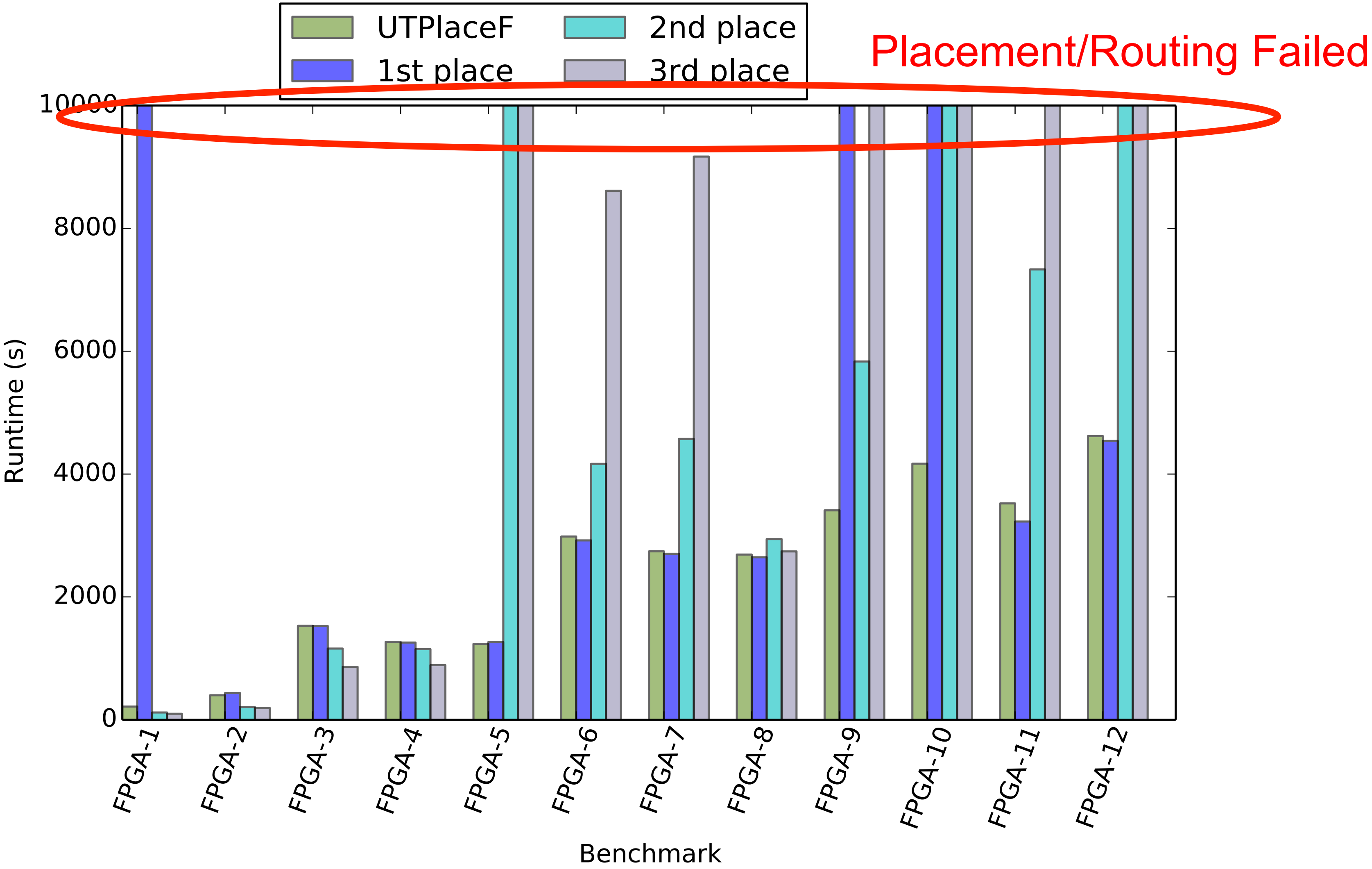
- Implemented in C++ with single thread
- 3.4GHz Linux server
- 32GB RAM
- ISPD'16 contest benchmark
- Routed wirelength reported by Xilinx Vivado v2015.4

Experimental Results – Routed Wirelength



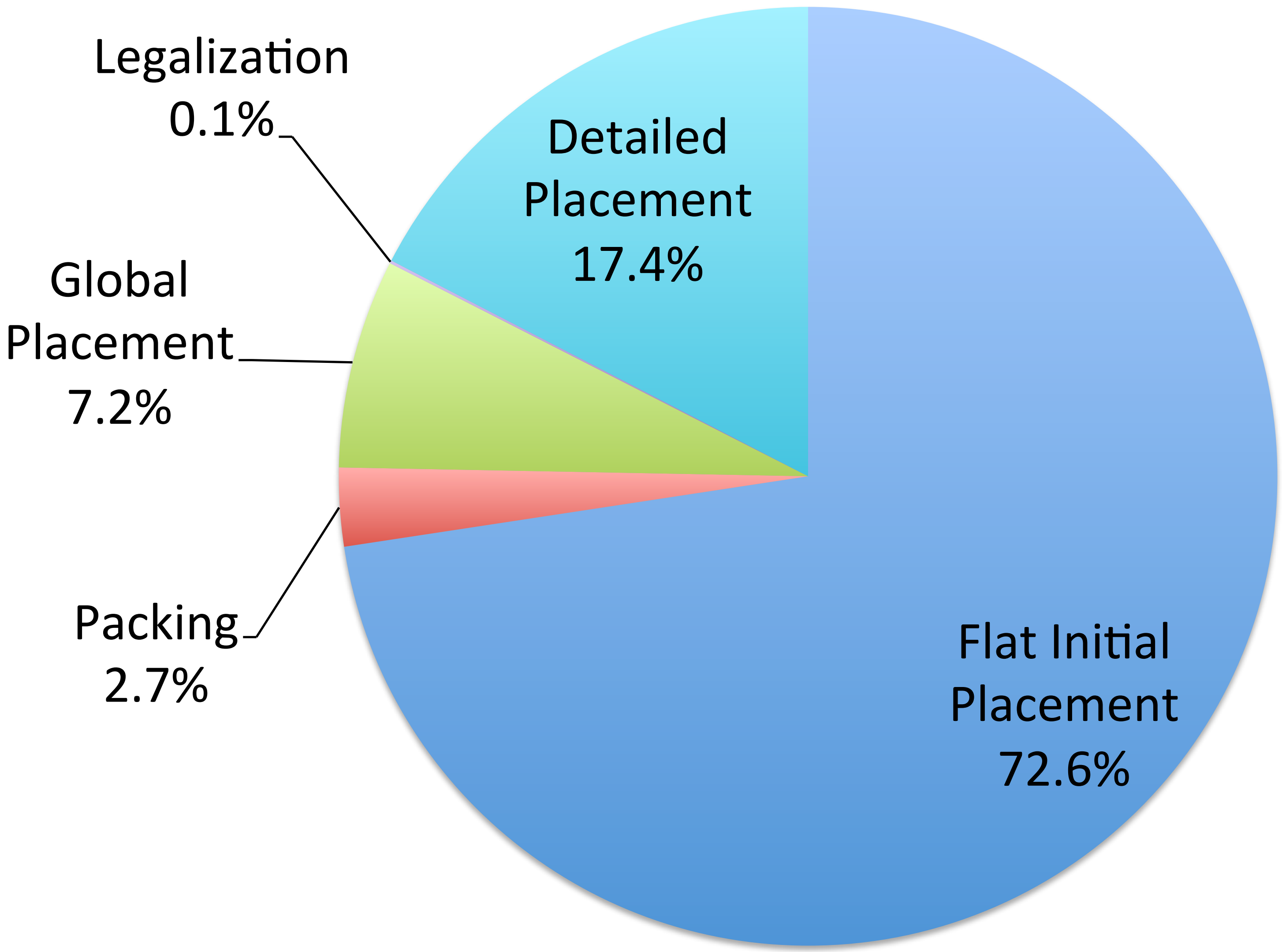
UTPlaceF outperforms by 3.3%, 7.7%, and 28.3% on routed wirelength compared with the top 3 contest winners

Experimental Results – Runtime



UTPlaceF has similar or faster runtime compared with the top 3 contest winners

UTPlaceF Runtime Breakdown



Conclusion

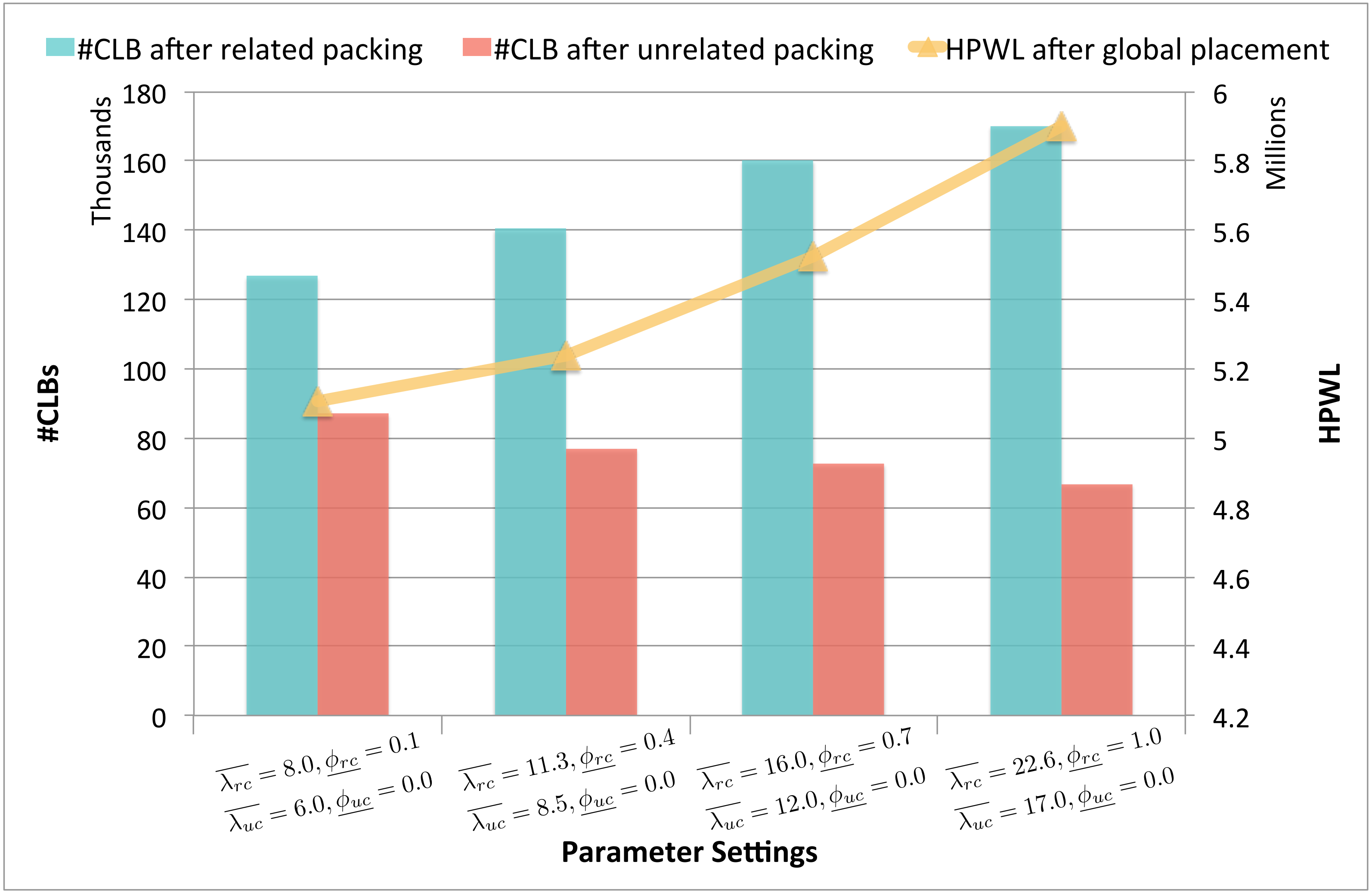
- Proposed a routability−driven FPGA packing and placement engine called UTPlaceF
  - A novel physical and congestion aware packing algorithm
  - Congestion aware detailed placement techniques
  - Better results cf. ISPD'16 contest winners
- Further work
  - Wirelength aware packing algorithms
  - Detailed packing optimization after initial packing

Thanks!

Backup

Wirelength and Packing Tightness Trade-off

- Benchmark: ISPD’16 Contest FPGA12

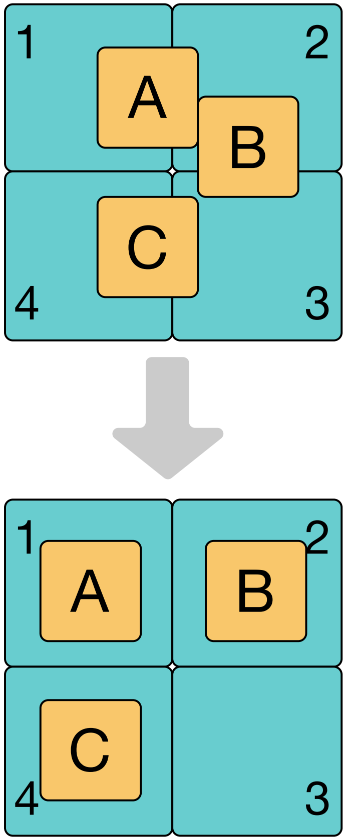


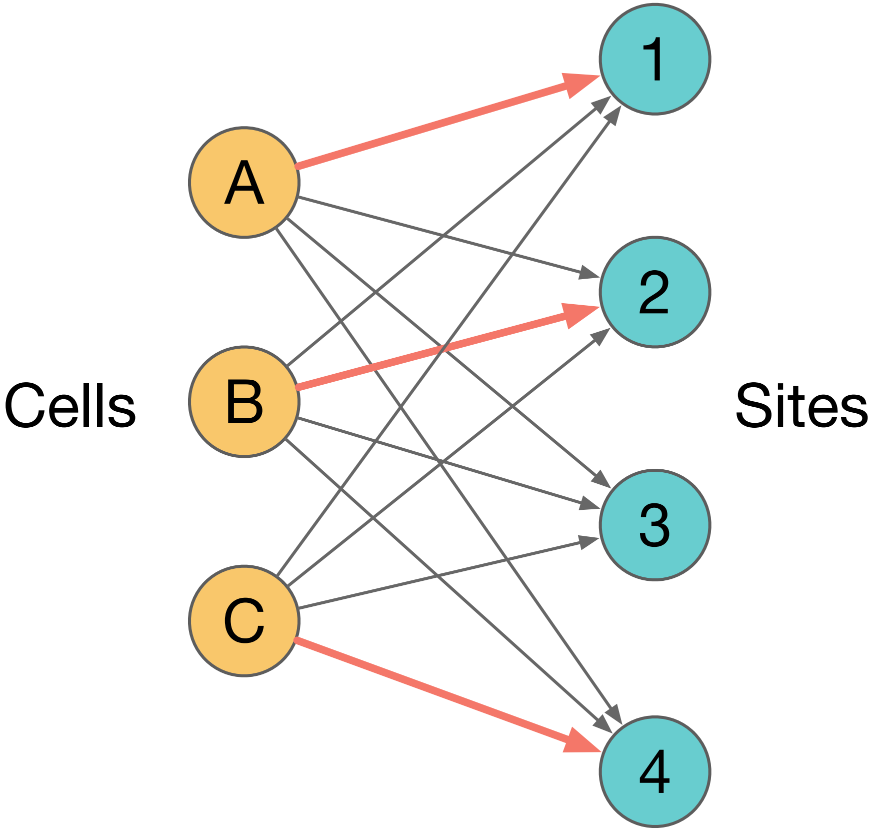
Looser packing
Better wirelength

Tighter packing
Worse wirelength

Minimum Movement Legalization

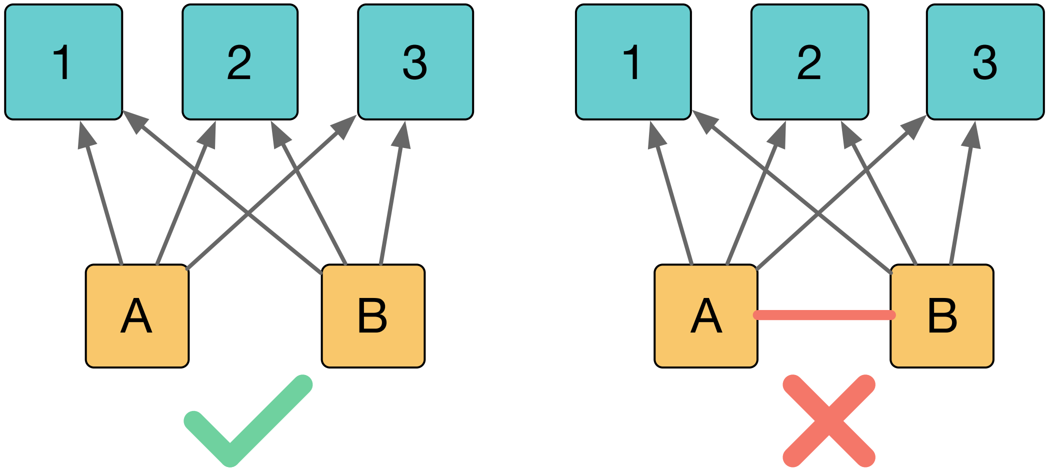
- Min-cost bipartite matching formulation
  - Cost = Manhattan distance between cells and sites
  - Minimize total cell movement





Detailed Placement: Independent Set Matching (ISM)

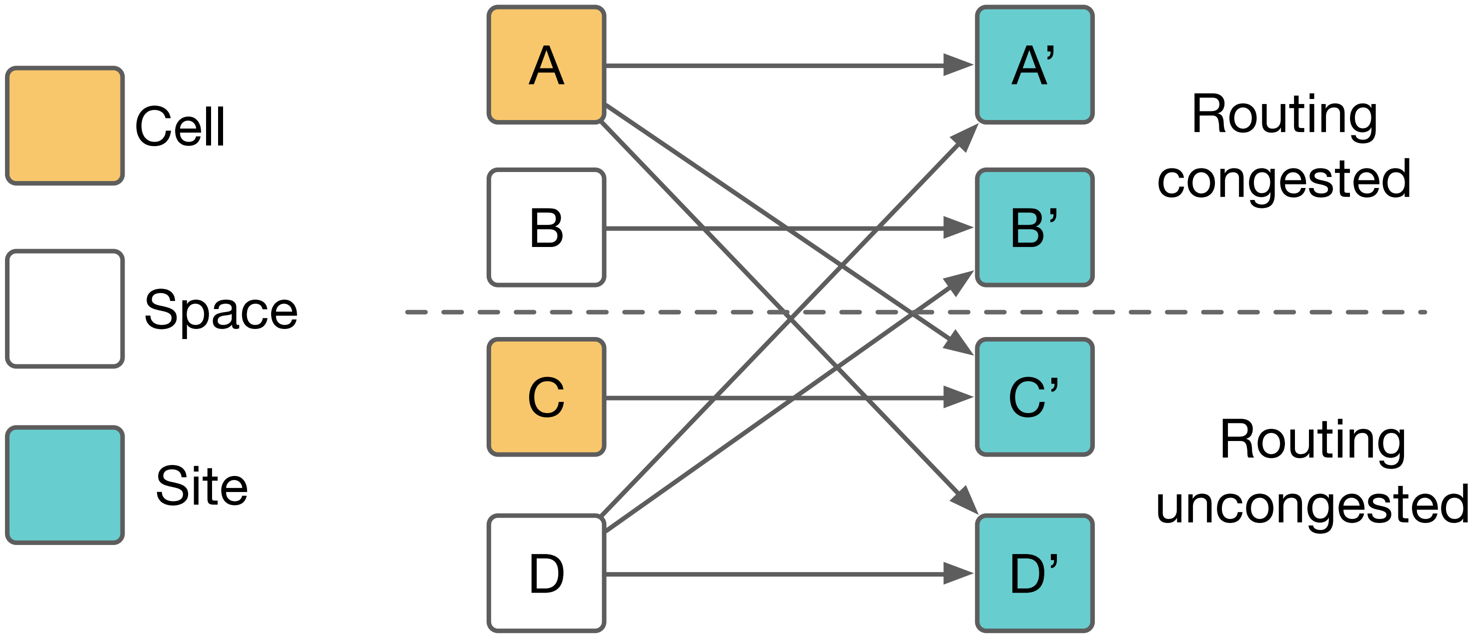
- Original ISM idea: NTUplace3 [Chen+, TCAD'08]
- Minimize HPWL using min-cost matching
  - cost = HPWL increase
- Min-cost matching cannot accurately model HPWL when moving a set of connected cells



- Only apply min-cost matching to a set of unconnected cells

Detailed Placement: Congestion Aware ISM

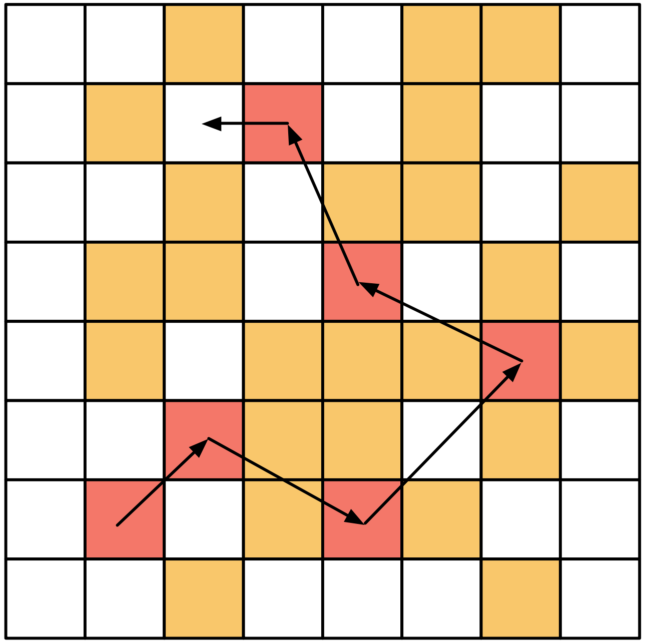
- Extra constraints for routability
  - Cell can be moved out of but not into routing congested regions
  - Spaces can be moved into but not out of routing congested regions
  - Moves within congested regions are not allowed

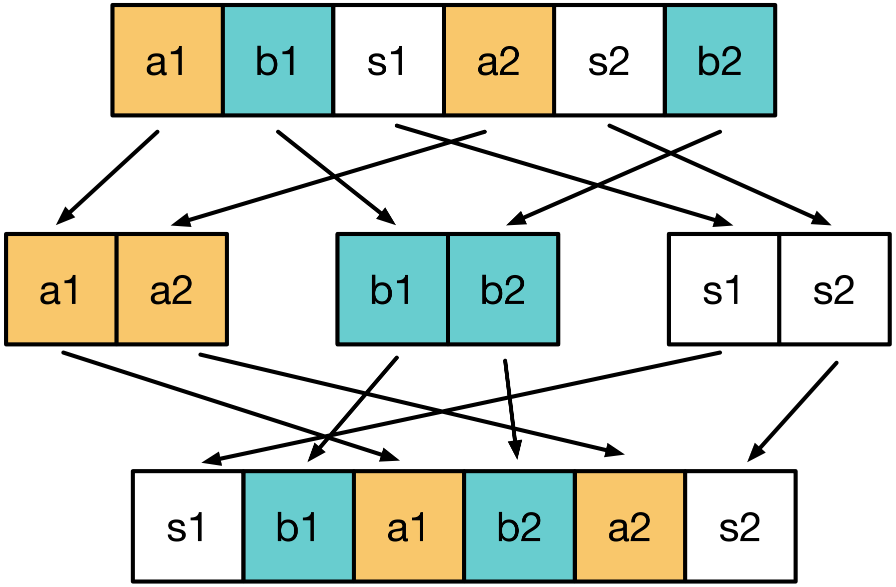


- Update routing congestion map after certain number of ISM iterations

Other Detailed Placement Techniques

- Global move [Pan+, ICCAD'05]
  - Extend to chain move to preserve cell density
- Cell interleaving [Hur+, ICCAD'00]




a1
b1
s1
a2
s2
b2
a1
a2
b1
b2
s1
s2
s1
b1
a1
b2
a2
s2