Pin Accessibility-Driven Detailed Placement Refinement

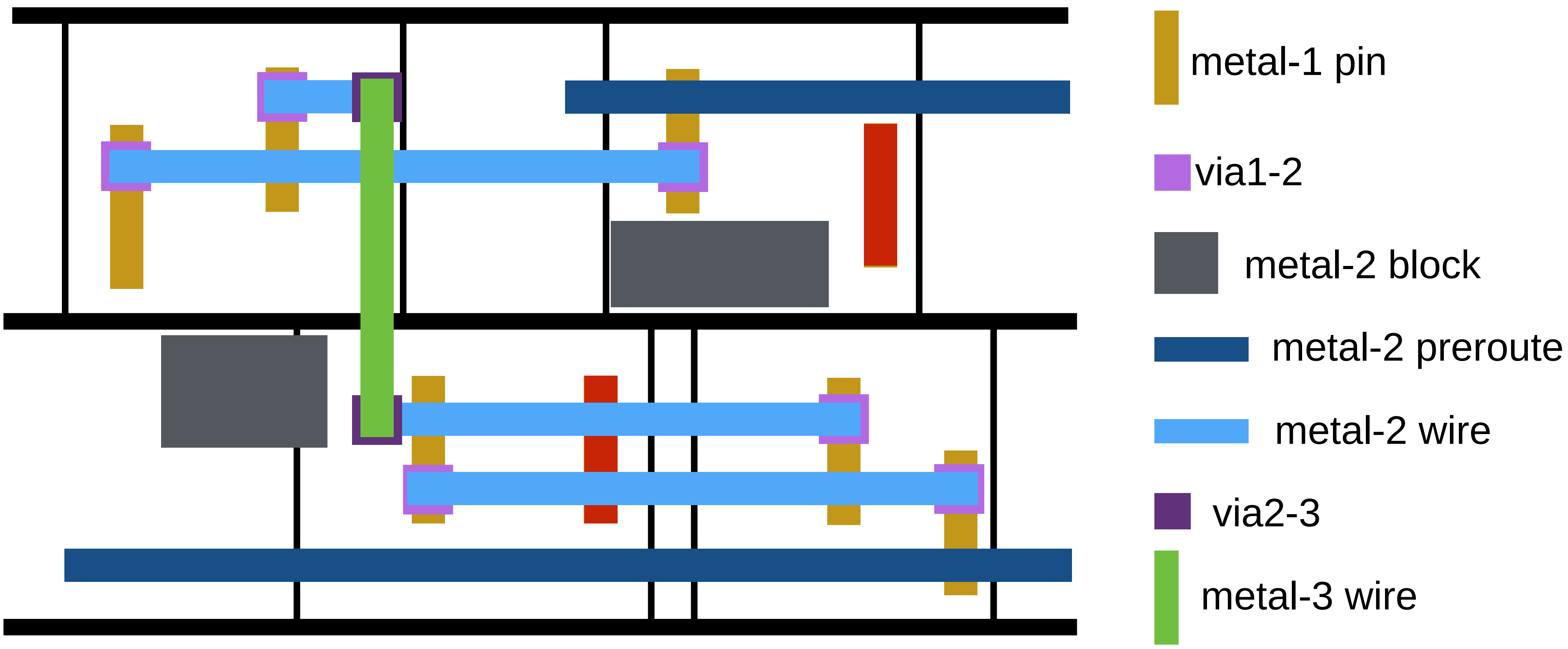
Yixiao Ding @ Cadence Design System
Chris Chu @ Iowa State University
Wai-Kei Mak @ National Tsing Hua University

Outline

- Introduction
  - Why pin access is a critical problem
  - Previous works and our motivation
- Overview of our solution
  - Pin accessibility-driven detailed placement (DP) refinement
  - Our contributions
- Problem formulation
- Background knowledge
  - Assumptions
  - Pin access region
  - Pin access penalty
- Proposed solution
- Experimental results

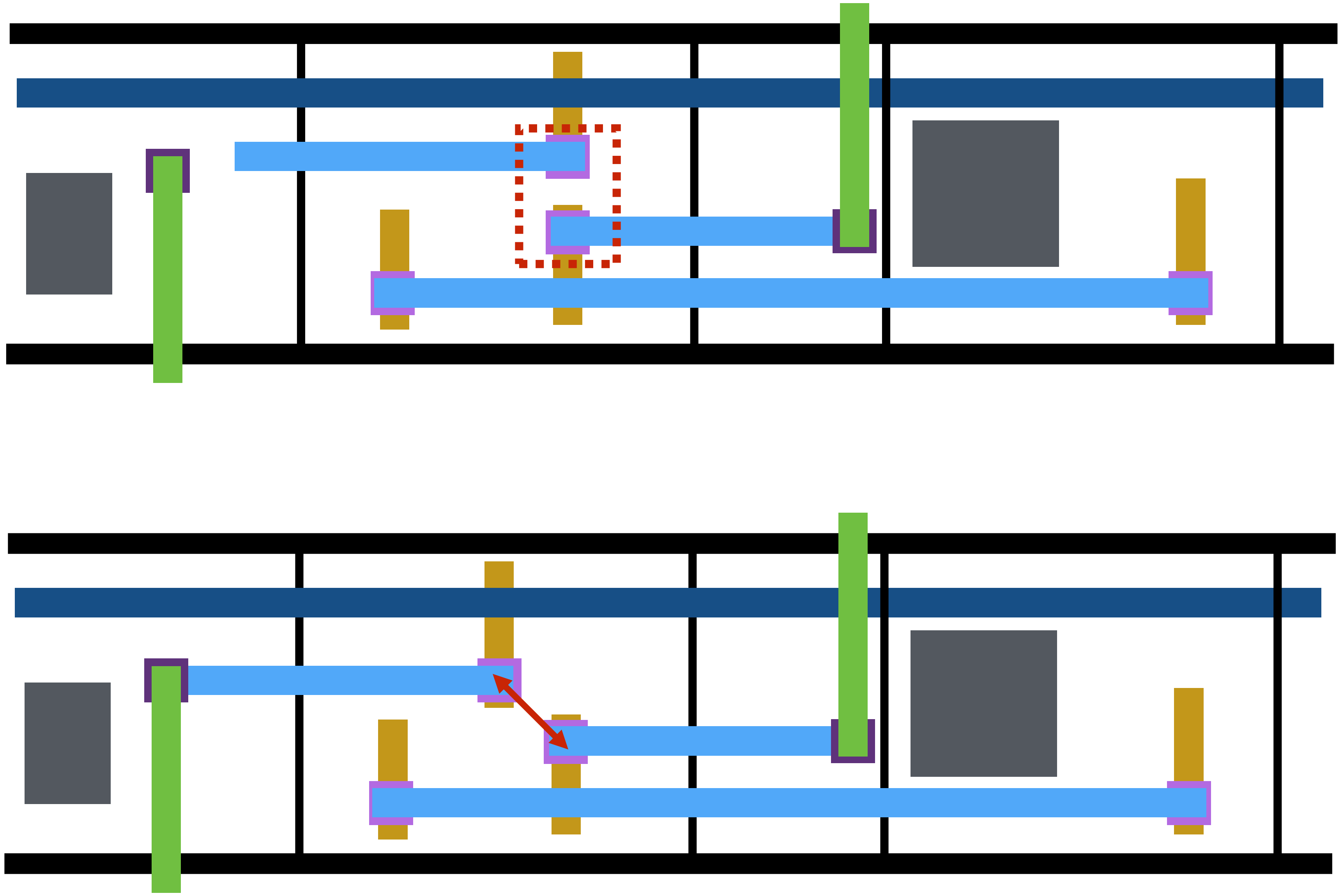
A critical pin access problem (1/2)

- Unidirectional routing trend in advanced nodes
  - Metal-1 pins can be easily blocked by straight metal-2 wires
  - Fierce routing resource competition on metal-2

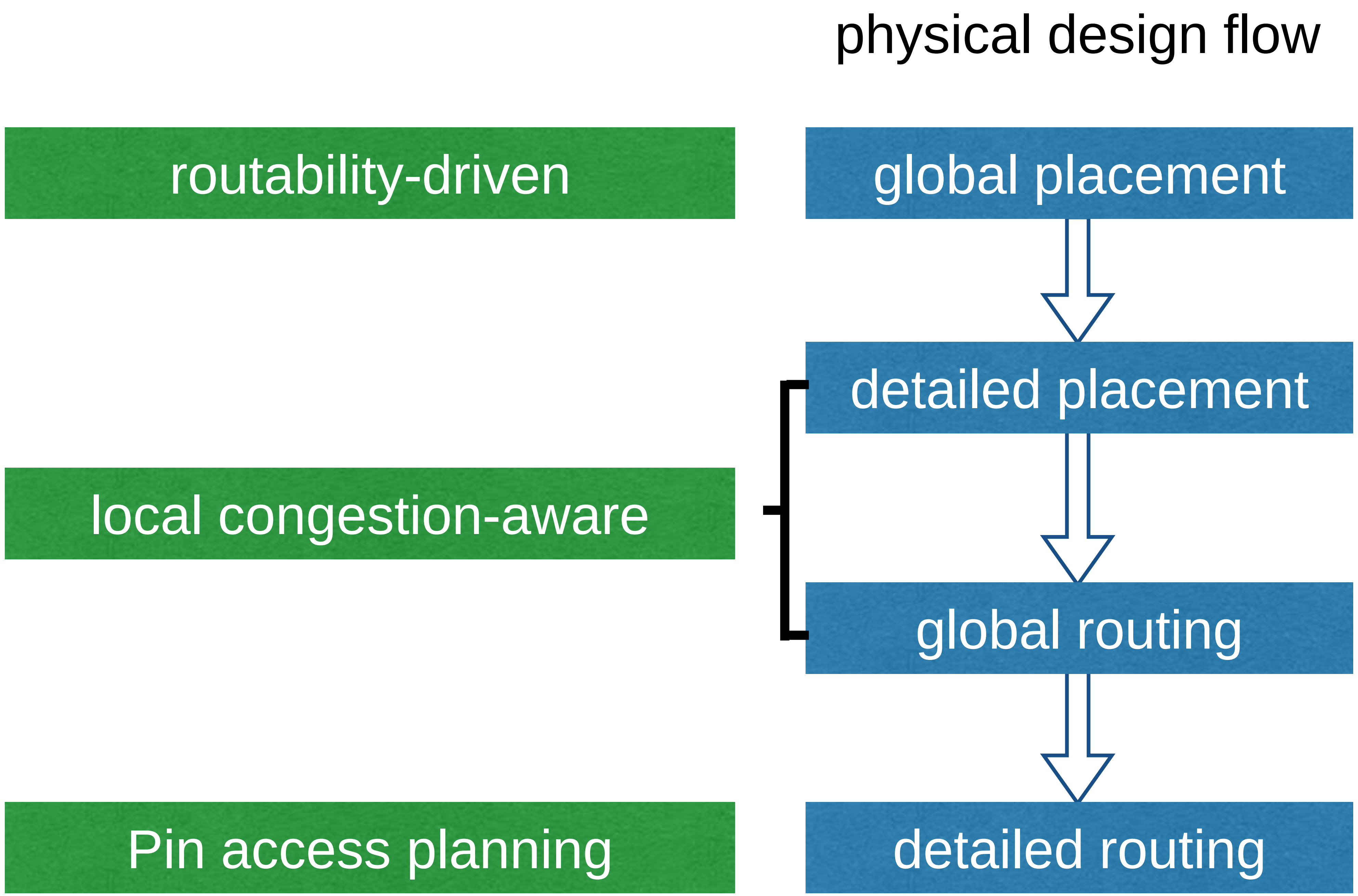


A critical pin access problem (2/2)

- More restricted routing design rules in advanced nodes
  - e.g., more space between vias
  - e.g., metal layer patterns are compliant to SADP design rules



Improve pin access in different design stages

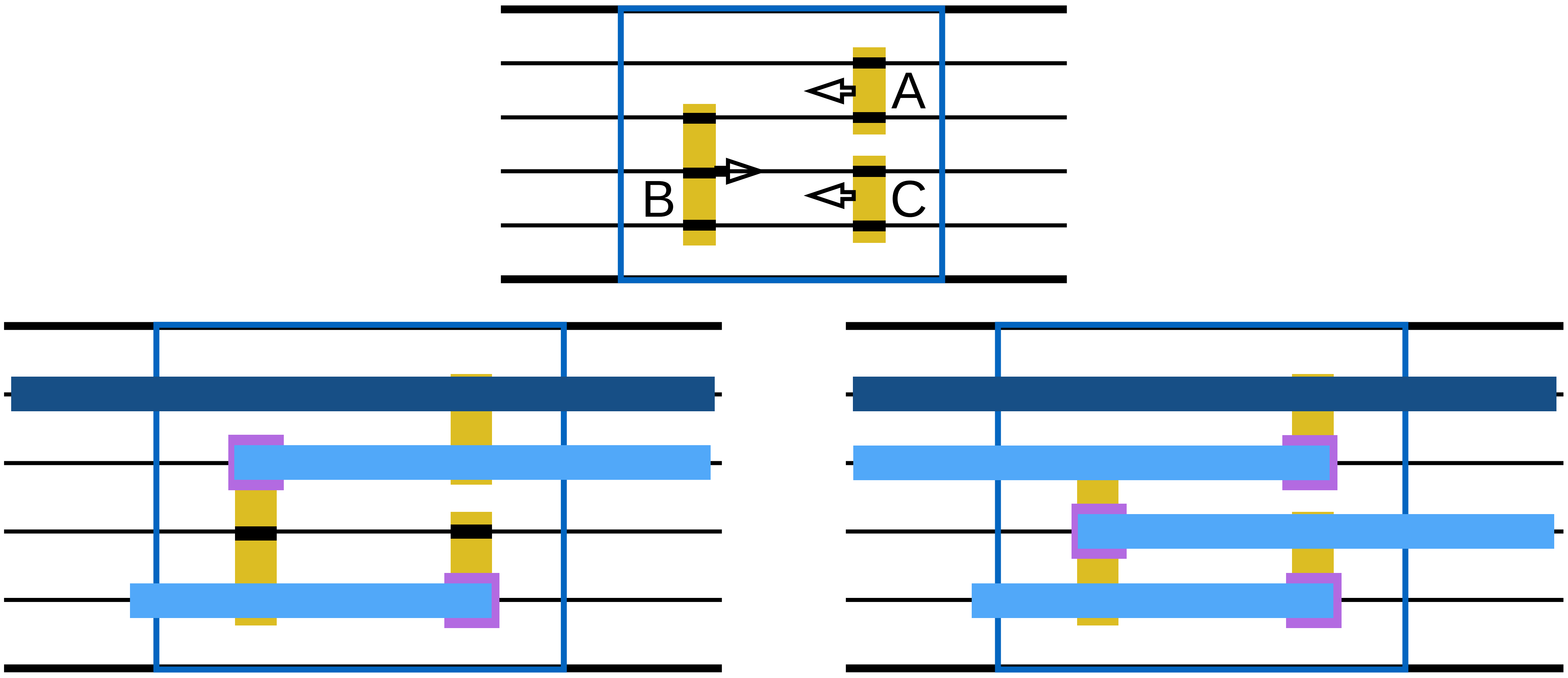


Selected previous works (1/3)

- Routability-driven global placement
  - T. Lin et al, “POLAR 2.0: An Efficient Routability-Driven Placer”, In Proc. of DAC’15
  - Cell spreading in congested region. too rough
- Local congestion-aware detailed placement
  - T. Taghavi et al, “New Placement Prediction and Mitigation Techniques for Local Routing Congestion”, In Proc. of ICCAD’10
  - Identify hard-to-route cell based on pin area and resolution. not exact
- Local congestion and pin access-aware global routing
  - C. Alpert et al, “Consideration of local routing and pin access during VLSI global routig”, US Patent’ 13
  - Consider pin count, relative location, and Steiner tree length. limited
- Pin access planning in detailed routing (DR)
  - Next two slides

Pin access planning in DR (1/2)

1. X. Xu et al, “PARR: Pin Access Planning and Regular Routing for Self-Aligned Double Patterning”, In Proc. of DAC’15
2. M. Ozdal et al, “Detailed-Routing Algorithms for Dense Pin Clusters in Integrated Circuits”, In TCAD’09

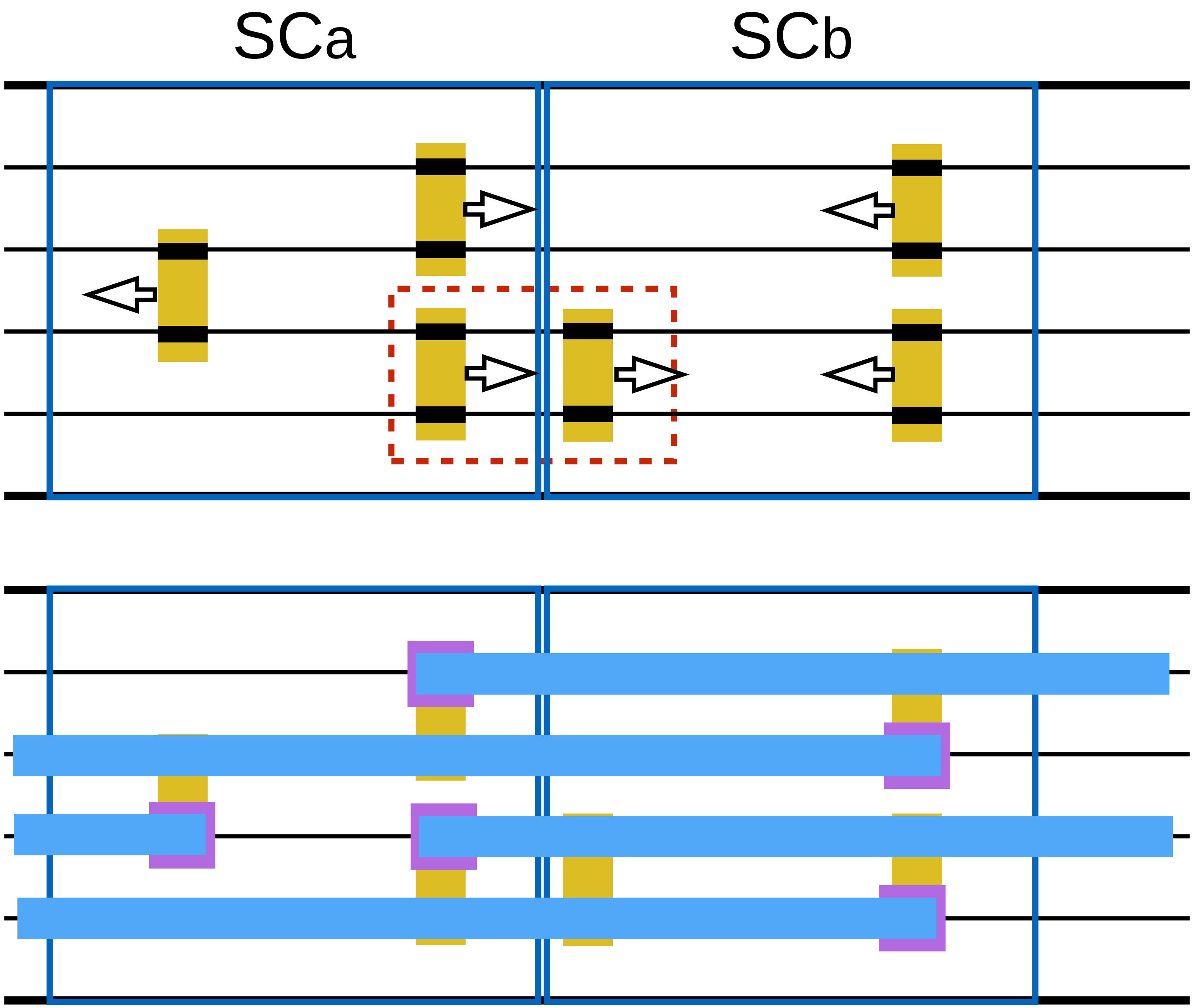


a bad planning

a wise planning

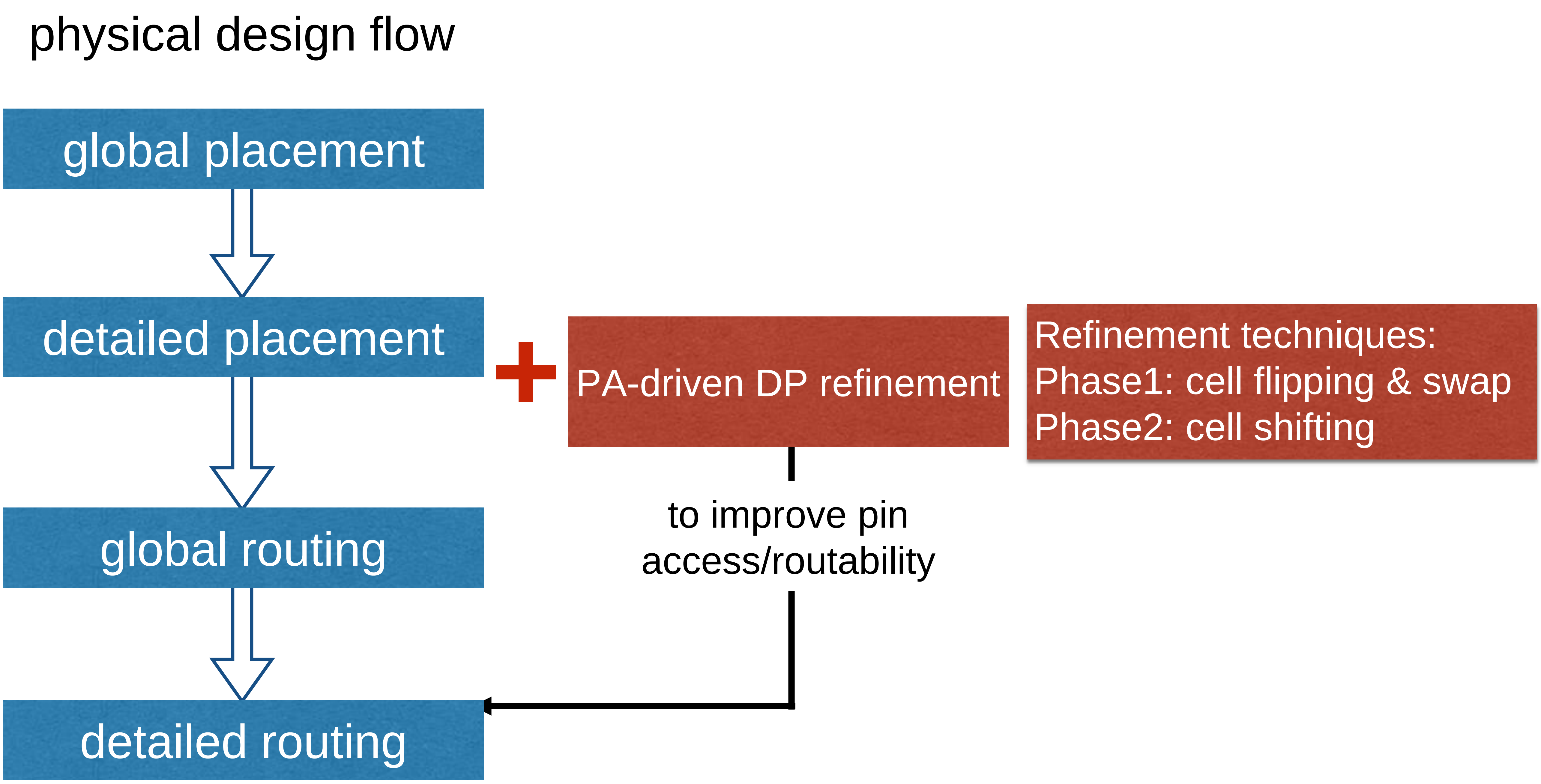
pin access planning in DR (2/2)

- It is not alway effective, especially in area with high pin density.



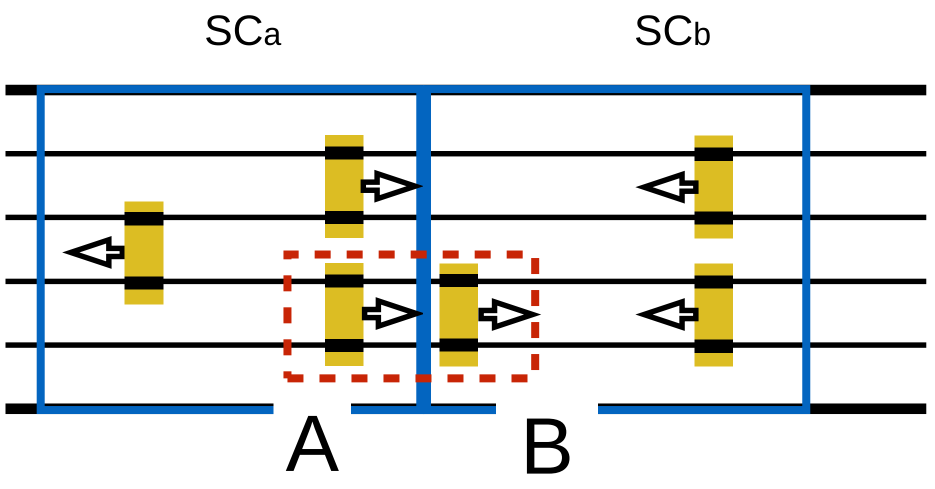
- Movtivation: we want to resolve pin access issue here!

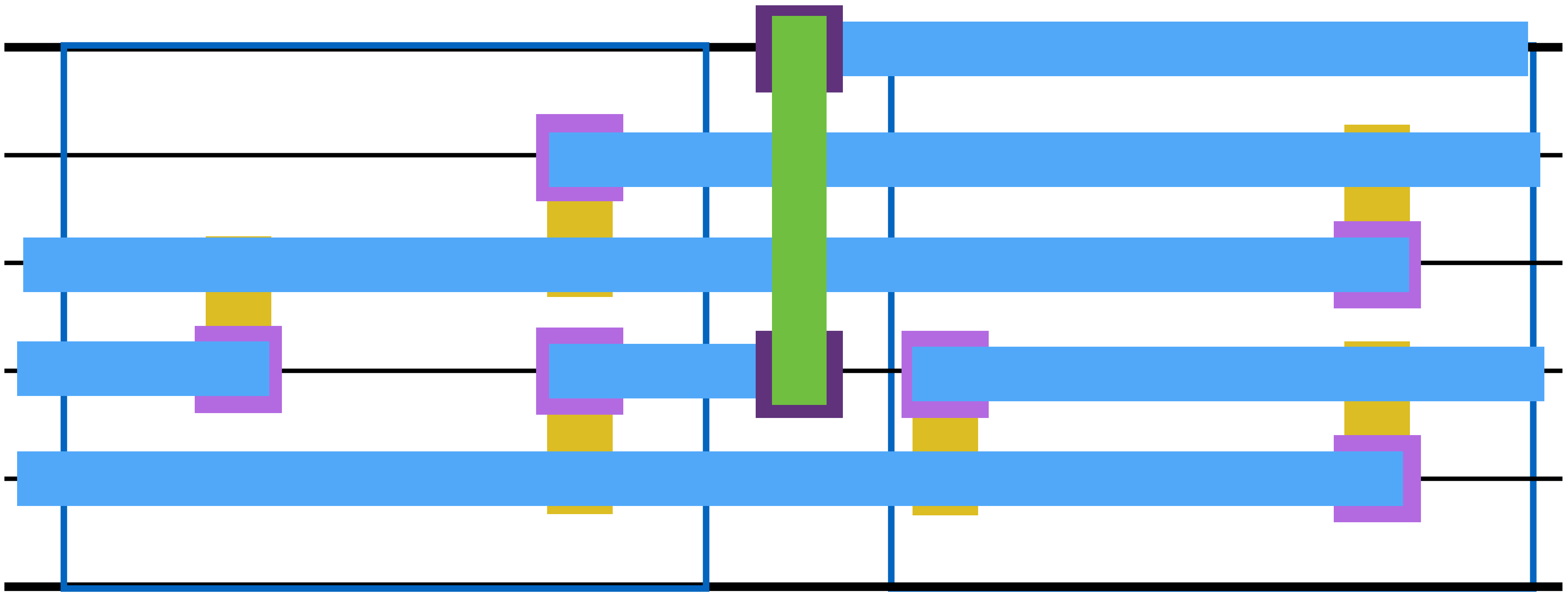
Our proposed solution



DP refinement techniques (1/2)

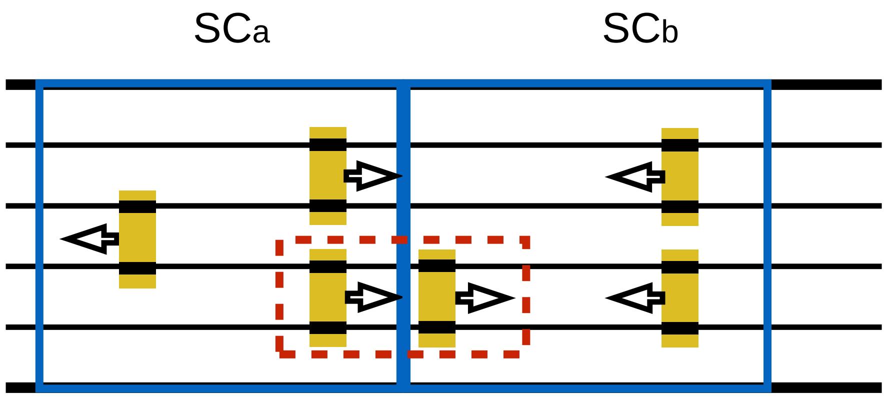
- Consider pin access in DP, when cell movement is allowed
- Cell shifting, adjacent cell swap, and cell flipping

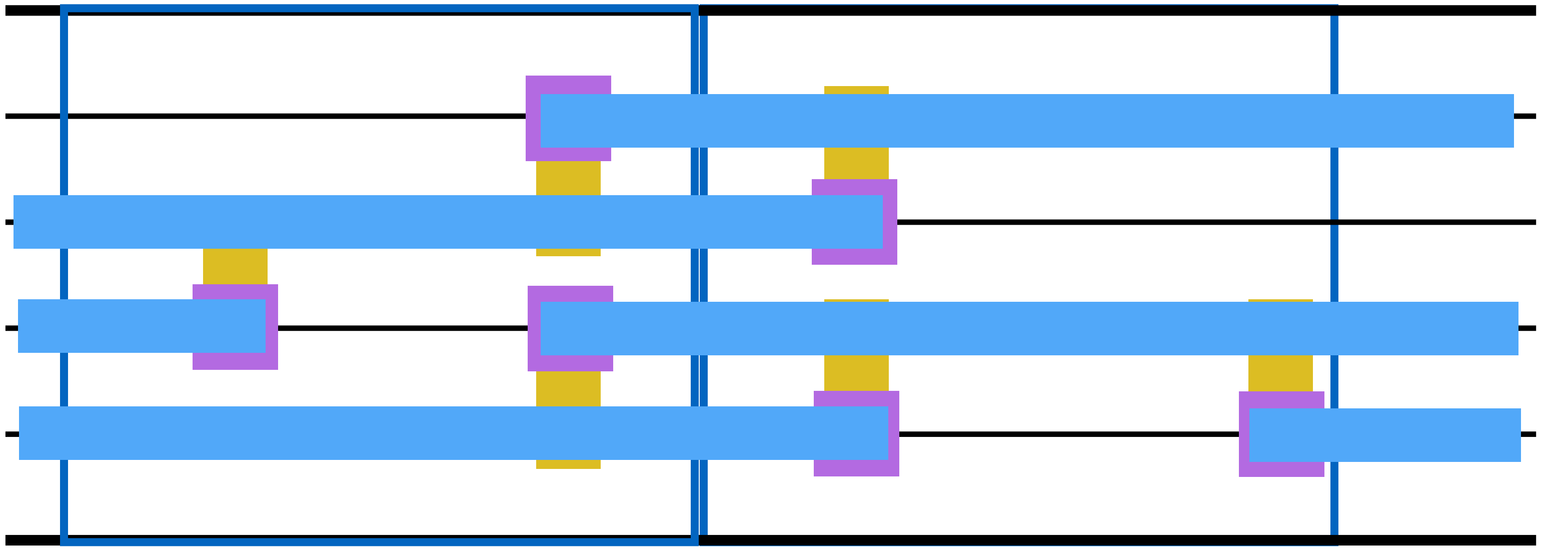




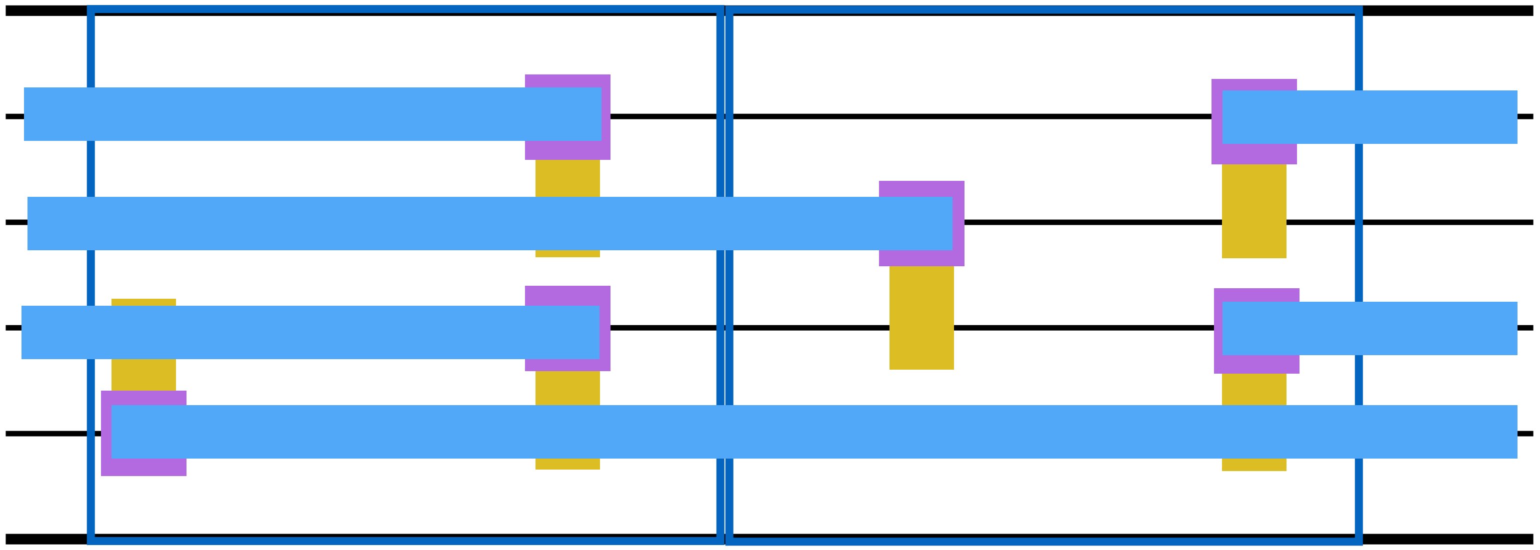
shift SCb to right

DP refinement techniques (2/2)





flip SCb



swap SCa and SCb

Our contributions

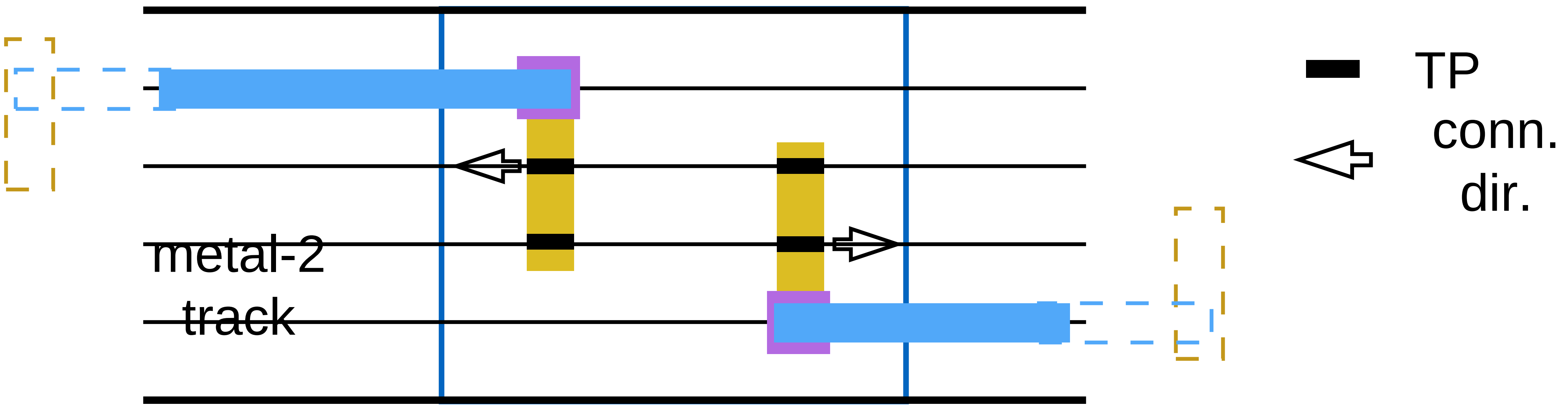
- It is the first work to directly consider pin access issue in detailed placement (DP) stage
- An accurate model is proposed to capture pin access scenario in detailed routing. A cost function is presented to guide DP refinement to improve pin access
- Our DP refinemnt techniques are limited to cell flipping, adjacent cell swap, and cell shifting. Our proposed solution is dynamic programming and linear programming-based.
  - Respect the given placement solution
  - Gurantee good solution quality with fast runtime
- Experimental results demonstrates the effectiveness of our proposed pin access-driven DP refinement

Problem formulation

- Given
  - A legalized placement
- We try to refine the placement by cell shifting, cell flipping, and adjacent cell swap.
- Objective
  - Pin accessiblity / routability is improved in DR stage
  - Placement perturbation should be minimized
  - The overheads of WL, via count in DR solution should be small or unchanged
- Constraint
  - Refined placement is legal

Assumptions

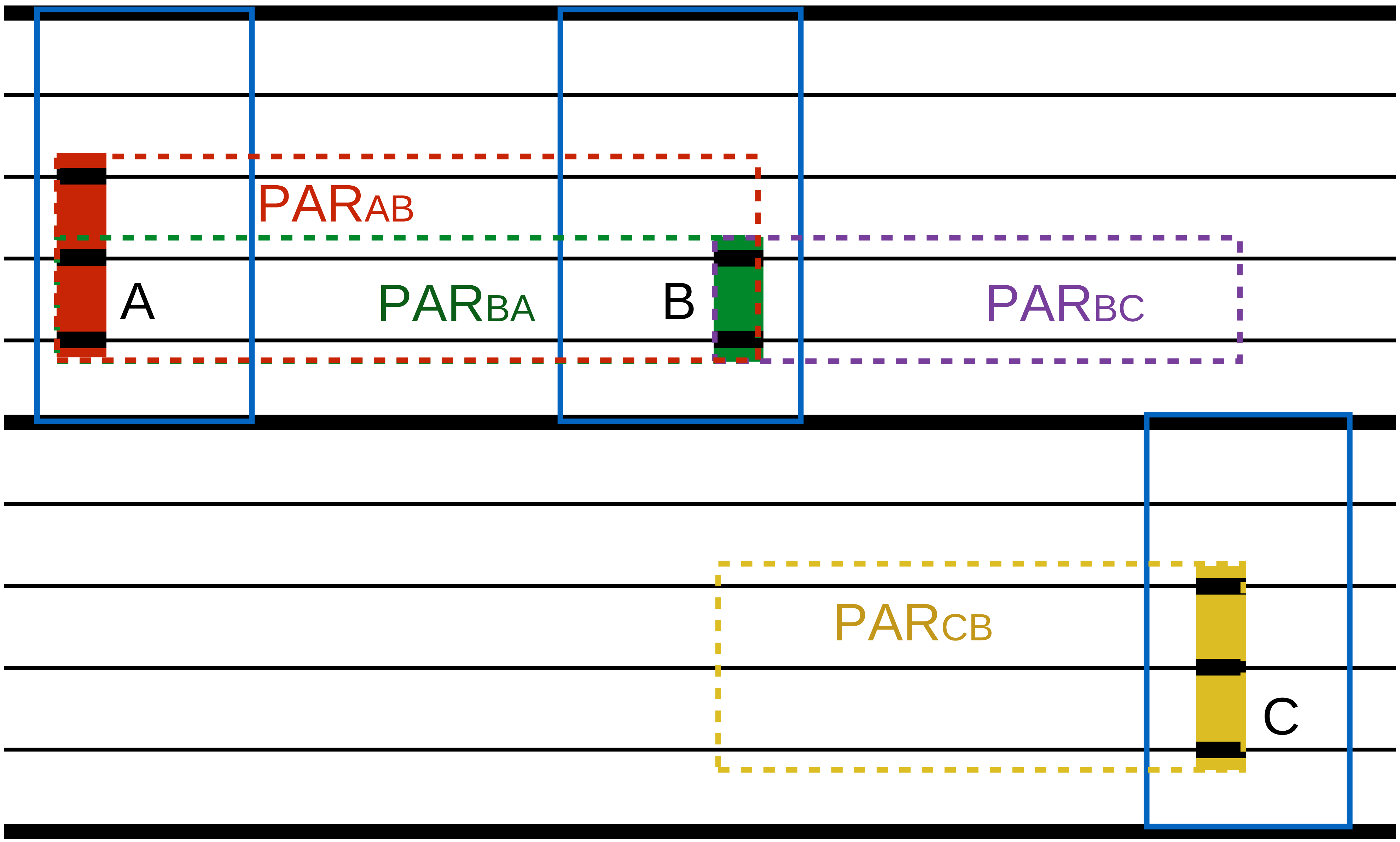
- Each metal layer has a preferred routing direction.
  - Metal1 is unroutable. metal2 horizontal, metal3 vertical…
- Standard cell (SC) ’s pins are rectlinear polygons on metal1. Each pin may span several metal-2 tracks.
  - The tapping point (TP) of a pin is defined as the overlap of metal-2 track and the pin shape



Pin access is to select a TP as a via location to connect metal-1 pin and metal-2 wire segment such that the metal-2 wire segment can be extended toward conn. dir. until the other connected pin.

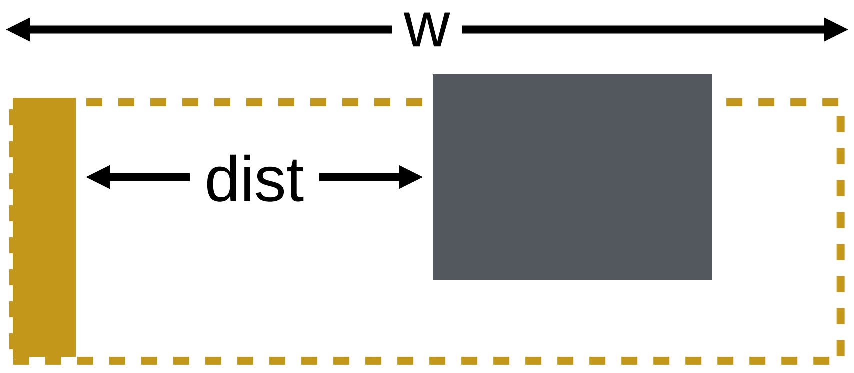
Pin access region (PAR)

- Given a 3-pin net {A, B, C}, a PAR for is defined for each connection of each pin
  - Same-row connection AB
  - Different-row connection BC

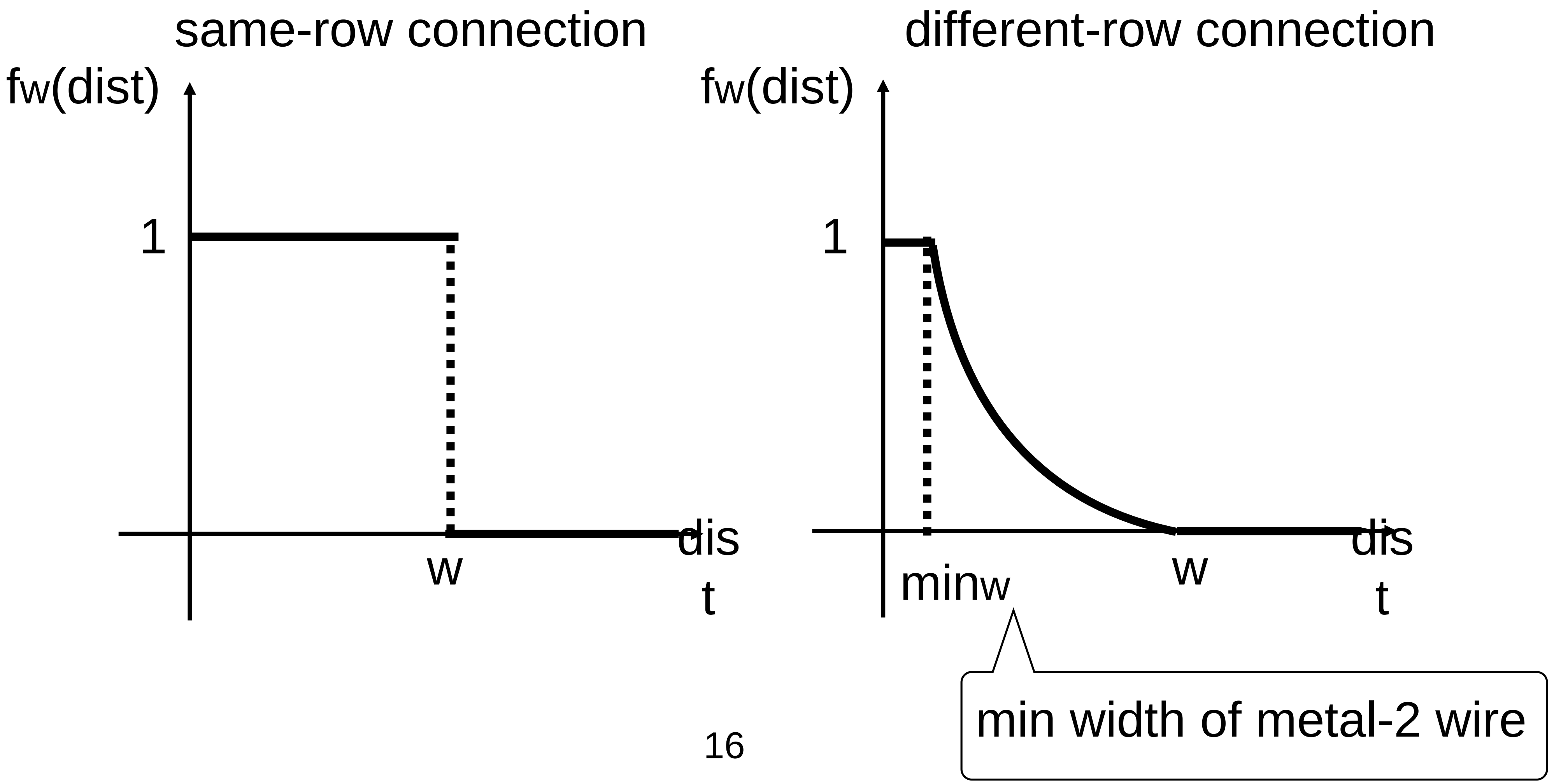


Pin access penalty (PAP) (1/3) - penalty function

- Objects (e.g., metal-2 block, wire segment) in PAR are penalized.



- Penalty function $f_w(dist)$



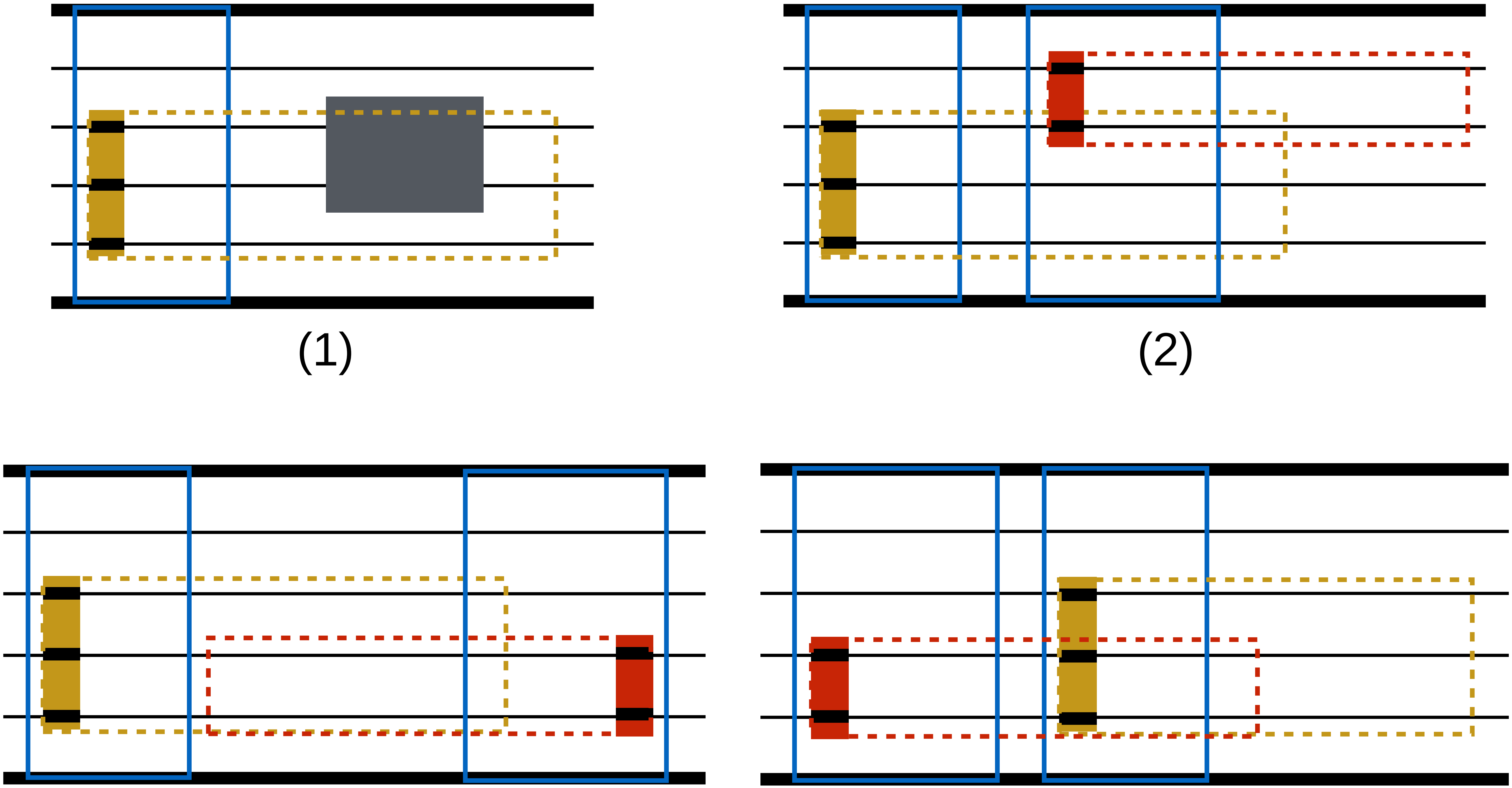
Pin access penalty (PAP) (2/3)

- For each connection, PAP reflects its pin accessibility
- Given a connection, a conflicting block (CB) is a block obstruct its PAR
- Given a connection, a coflicting connection (CC) is a connection with a PAR intersects with its PAR.
- PAP of a connection is computed by accumulating the penalty cost due to all the CBs and CCs.

$$PAP_{AA'} = \sum_{block \in CB} PAP_{AA'}^{block} + \sum_{conn \in CC} PAP_{AA'}^{conn}$$

- PAP with a CB or CC = probability x f_W(dist)
  - Four different scenarios, and equations are different in differernt scenarios

Pin access penalty (PAP) (3/3)
- four scenarios



(1) (2) (3) (4)

Quantify the pin accessibilty of a DP

- For each connection, pin access penalty (PAP)

$$PAP_{AA'} = \sum_{block \in CB} PAP_{AA'}^{block} + \sum_{conn \in CC} PAP_{AA'}^{conn}$$

- For each cell c, cell pin access penalty (CPAP)

$$CPAP_c = \sum_{A \in Pin_c} \sum_{AA' \in Conn_A} PAP_{AA'}$$

- For a DP, total cell pin access penalty (TCPAP)

$$TCPAP = \sum_{c \in All\_Cells} CPAP_c$$

- **TCPAP** reflects pin accessiblity of a DP, and should be **minimized**

Two-phase PA-driven DP refinement

legalized DP

↓

pin accessibility-driven placement refinement

↓

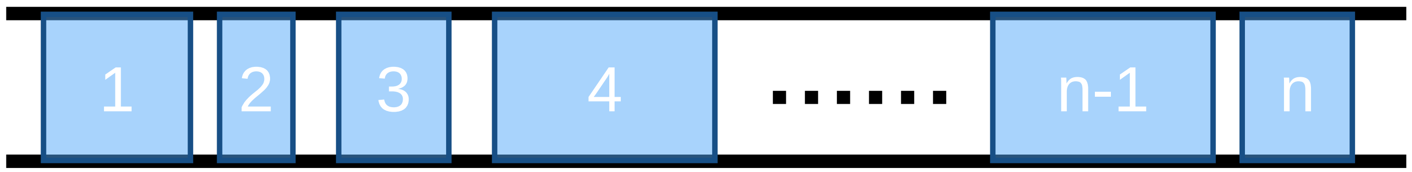
refined legalized DP

Phase 1: cell flipping & cell swap
- dynamic programming row by row

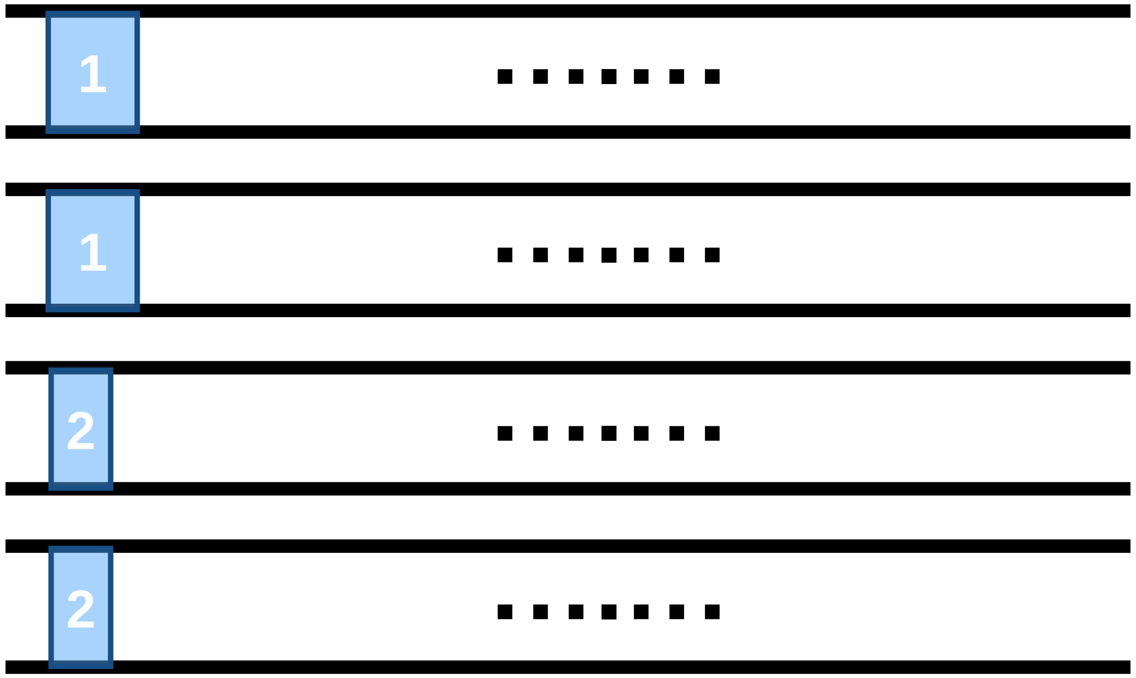
Phase 2: cell shifting
- linear programming

Refinement phase 1 (1/3)

- Given a row of placement, we try to minimize $\sum_{c \in C_{row}} CPAP_c$

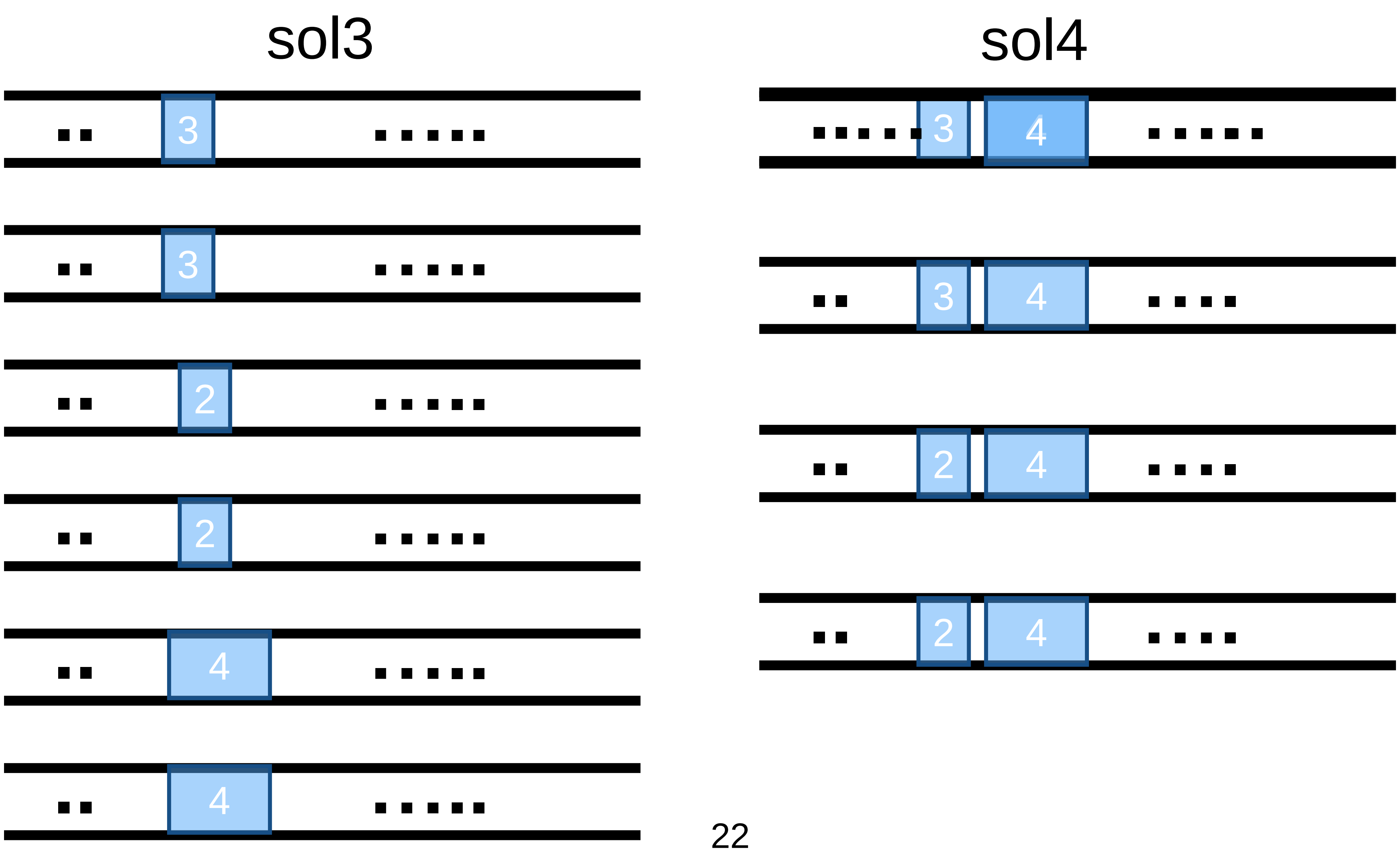


- Dynamic programming
  - Sol_k contains optimal refined prefix placememnt with k cells
  - Base cases sol_1



Refinement phase 1 (2/3)

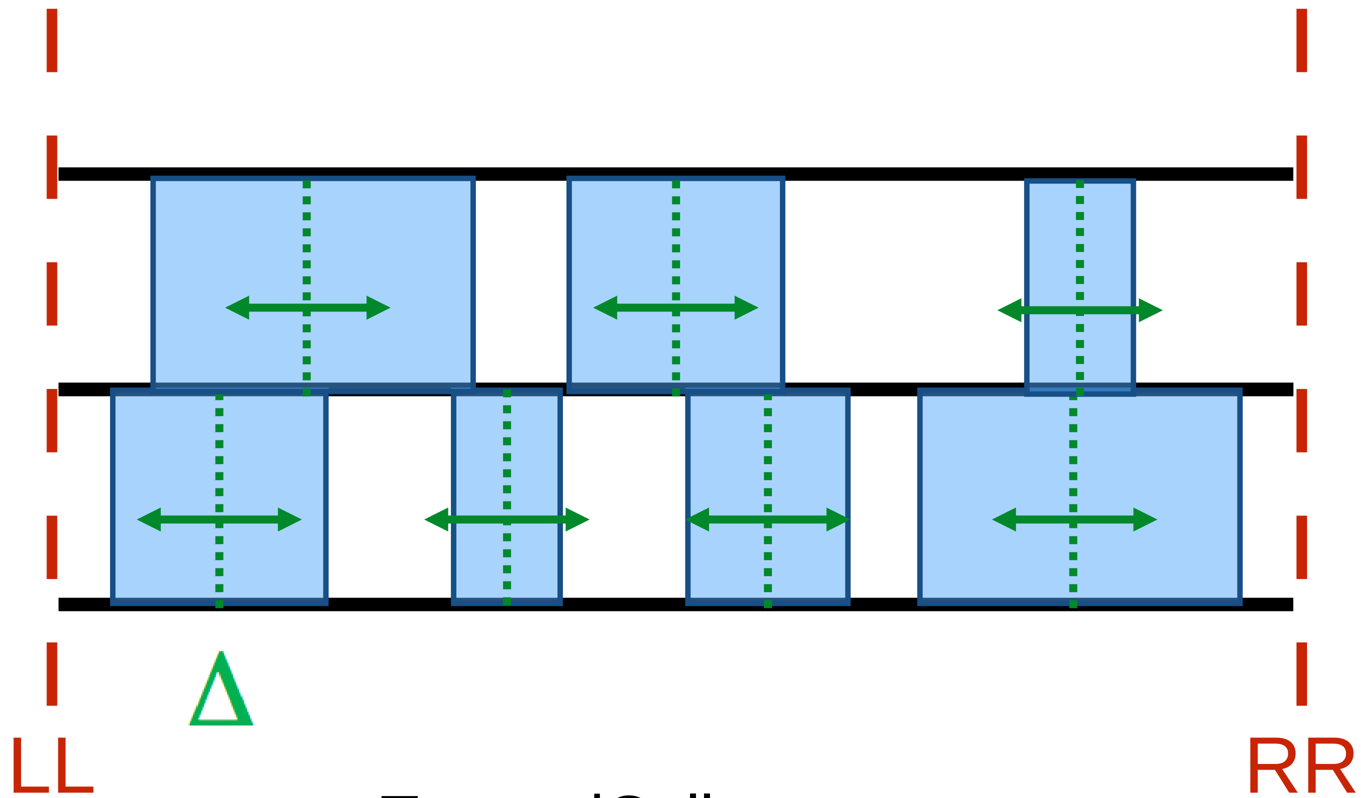
- Recursive formula
  - 6 kinds of solk should be kept in dynamic program to obtain soln, each kind is determined by the last cell placement in solk
  - Given a solk, solk+1 is obtained by finding min. total CPAP for cells in solk+1
  - An example to obtain sol4 from sol3



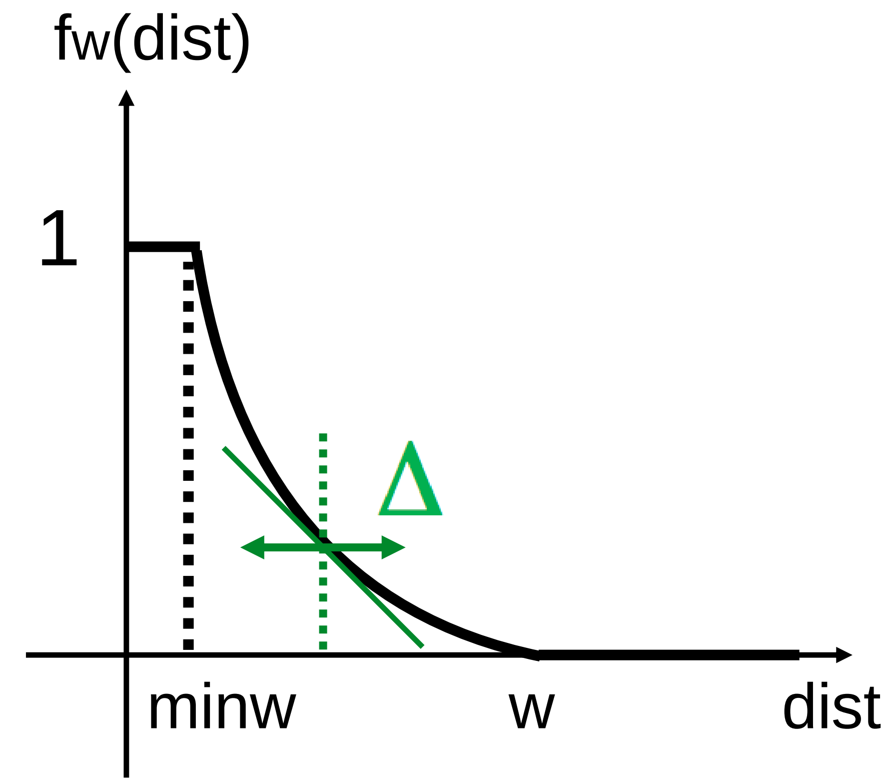
Refinement phase 1 (3/3)

- Opt is obtained by finding min. total CPAP among 4 kinds of soln

Refinement phase 2



Two stdCell rows



linear approximation

- Objective: minimize TCPAP
  - Linear approximation on penalty function
- Continuous variable denotes the x-location of each cell's lowerleft corner
- Δ controls the threshold of max cell shifting distance
- Linear constraints to ensure cells are not overlapped and out of LL & RR

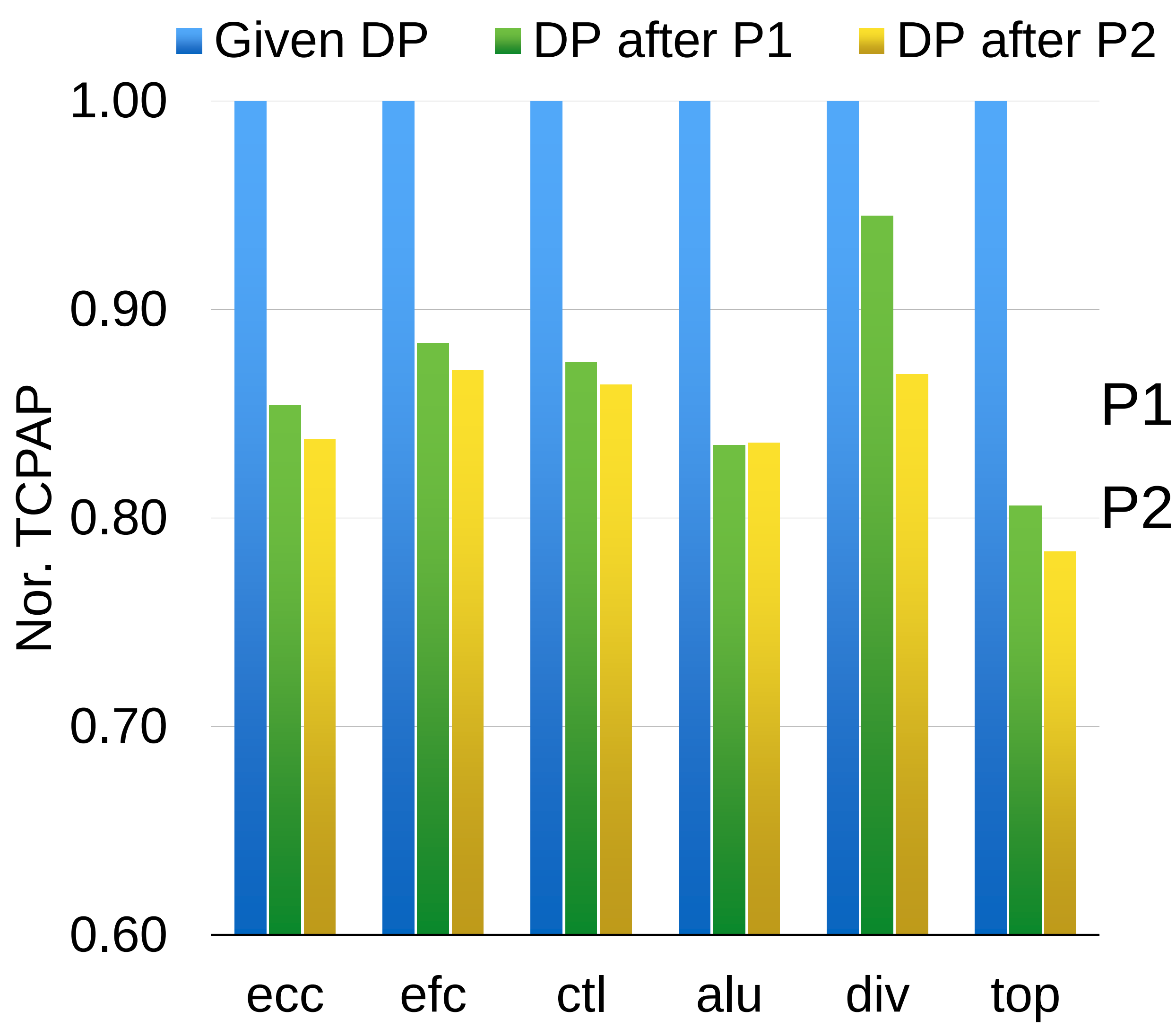
Experimental set-up

- PA-driven DP refinemnt is implemented by C++.
- Experiments run on 2.4 GHz Intel Core i5 and 8GB memory.
- Gurobi 6.0 is called to solve linear program in phase 2.
- Original benchmarks are from [1].
  - [1] X. Xu et al, “PARR: Pin Access Planning and Regular Routing for Self-Aligned Double Patterning”, In Proc. of DAC’15
- SADP-aware detailed router in [2] is called to route refined DP.
  - [2] Y. Ding, et al, “Self-aligned double pattering lithography-aware detailed routing with color pre-assignment”, TCAD’16
- Two sets of experimental results are demonstrated:
  - PA-DP refinement
  - Detailed routing on refined placement

PA-driven DP refinement (1/2)

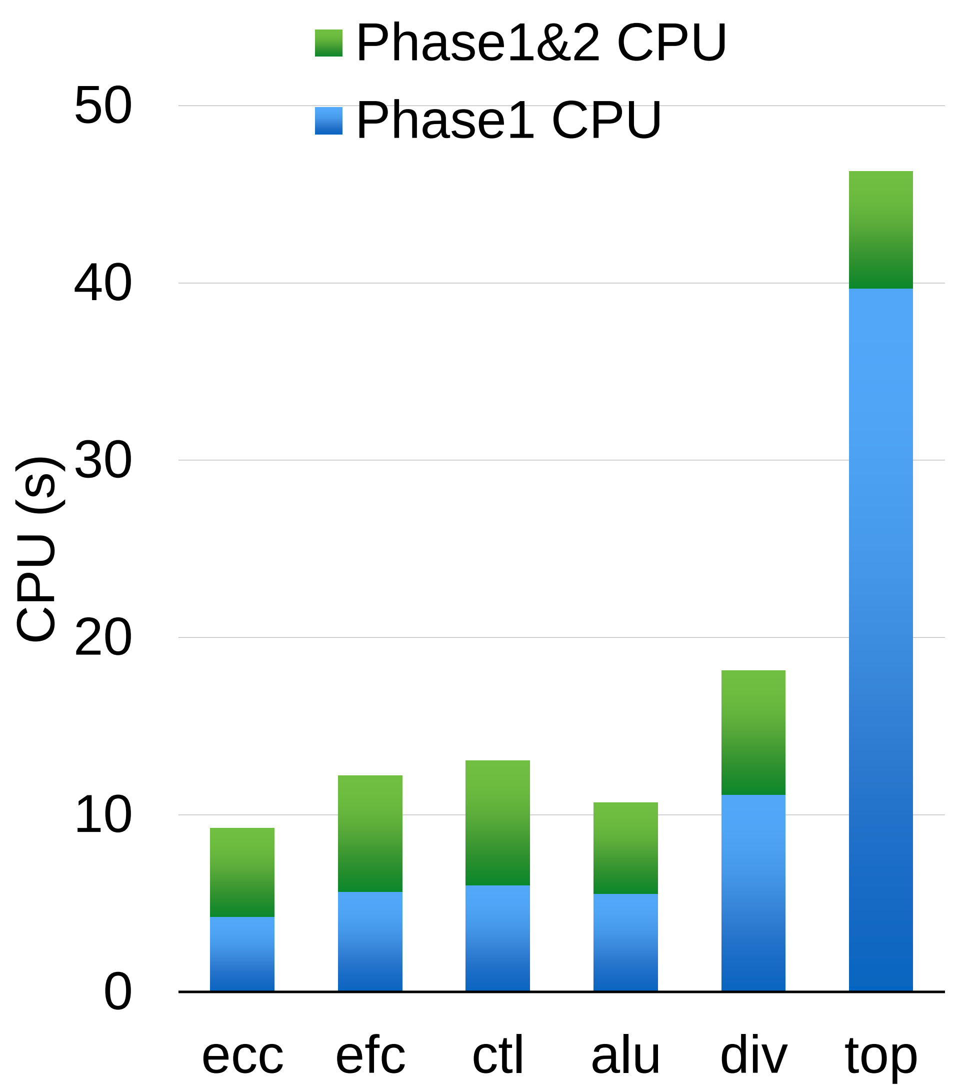
- Benchmark statistics

	ecc	efc	ctl	alu	div	top
#cells	1302	1197	1715	1802	3260	12576

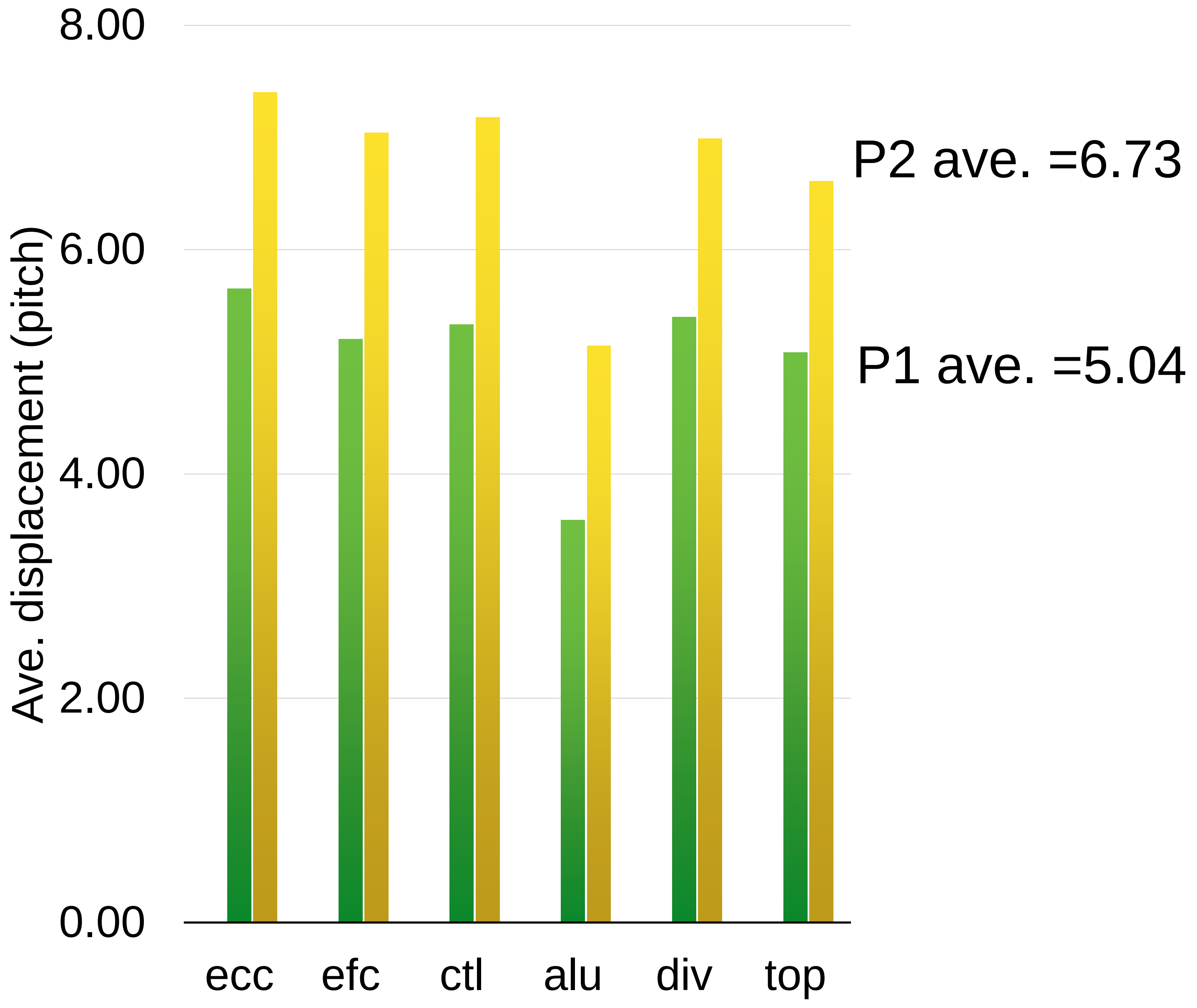


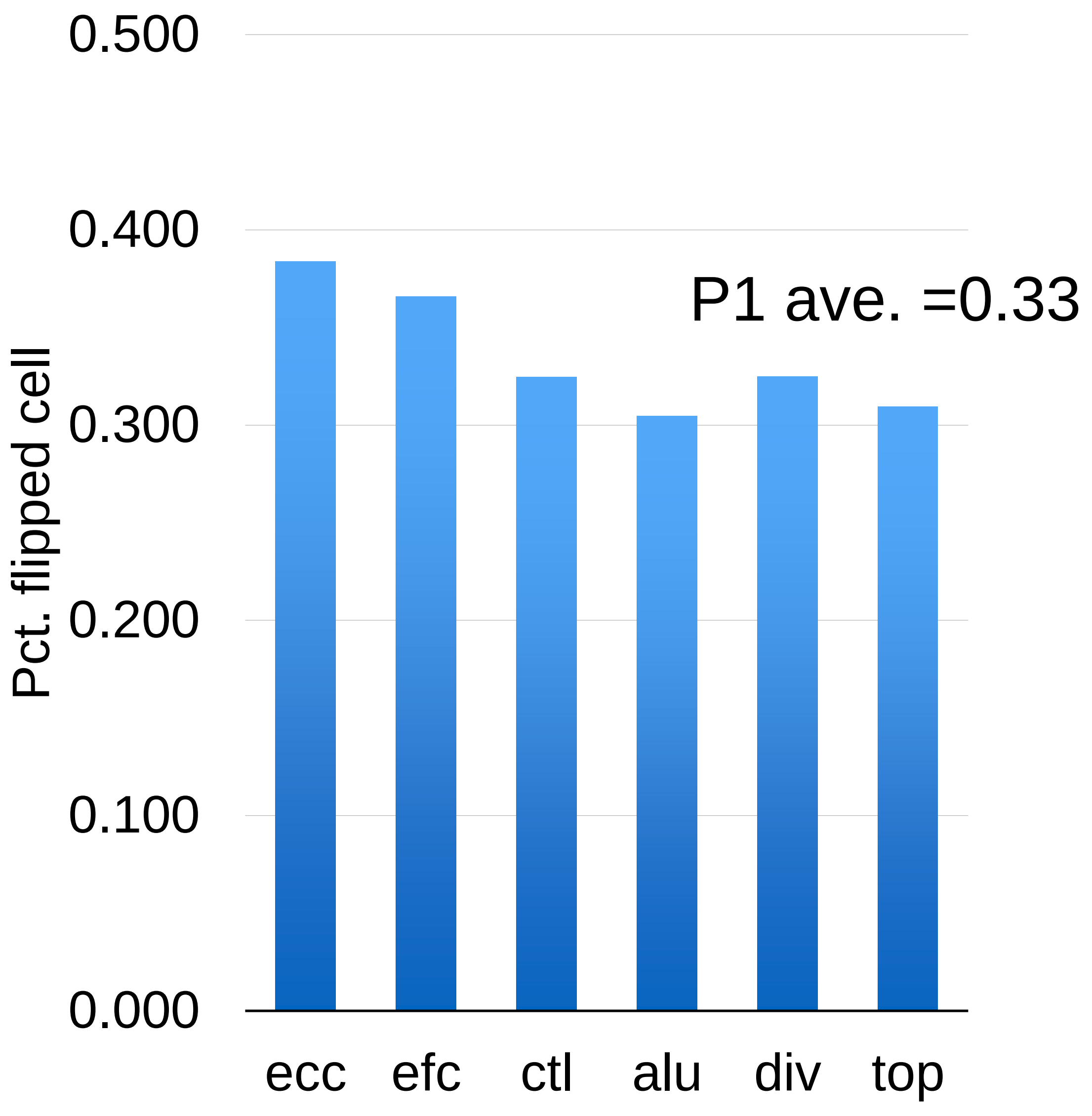
P1 ave. =0.85

P2 ave. =0.82

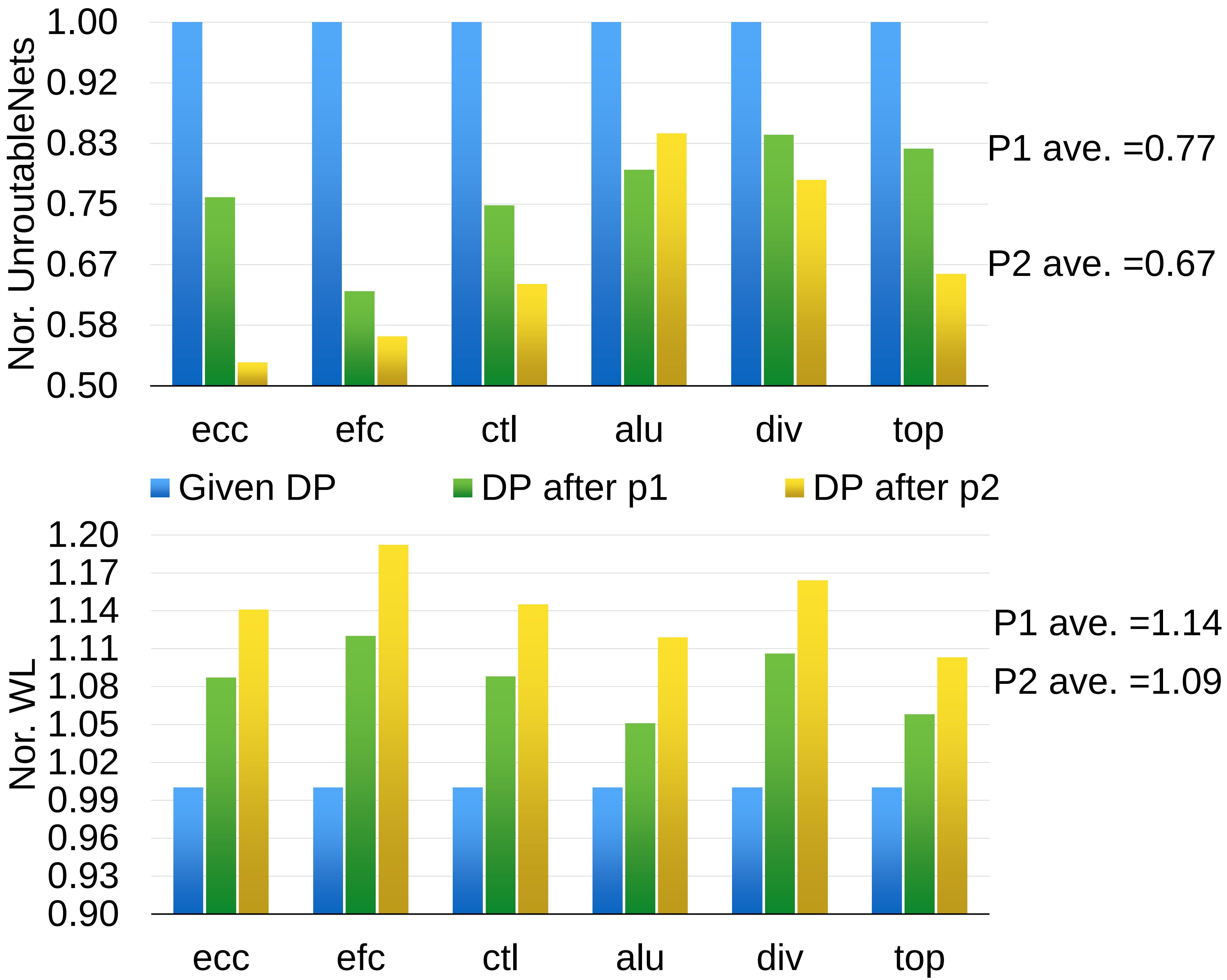


PA-driven DP refinement (2/2)

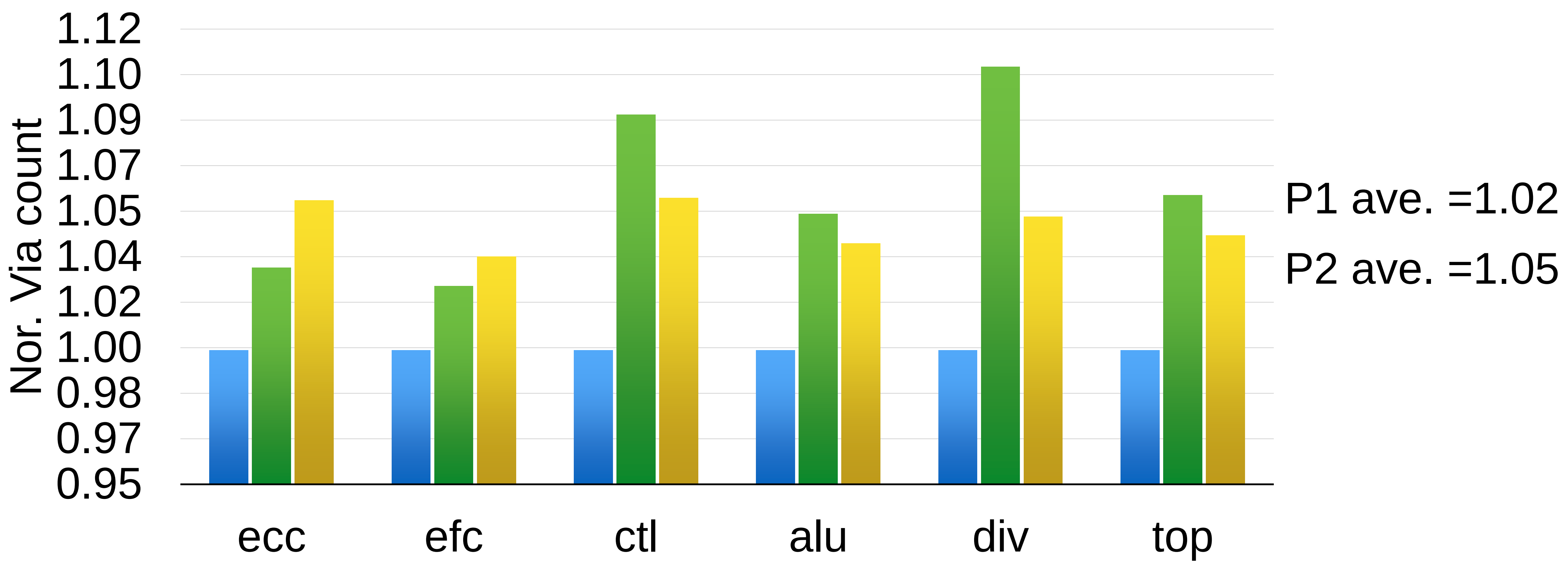


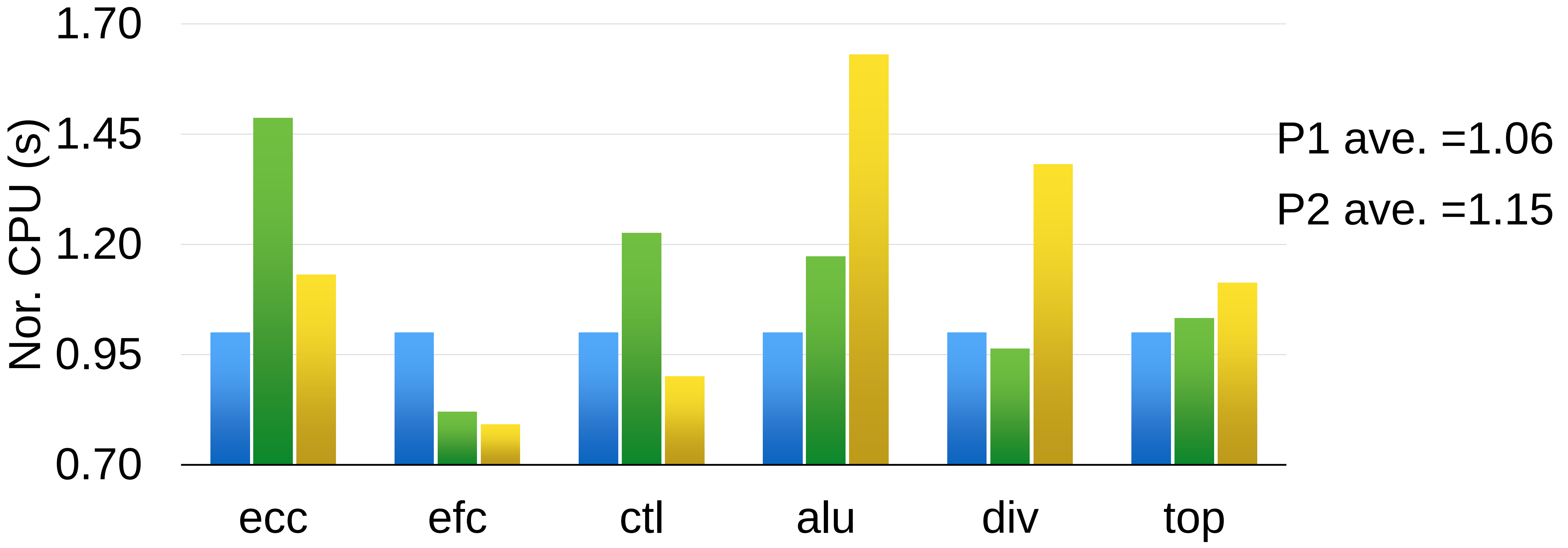


Detailed routing results (1/2)



Detailed routing results (2/2)





Thank you!

Q & A