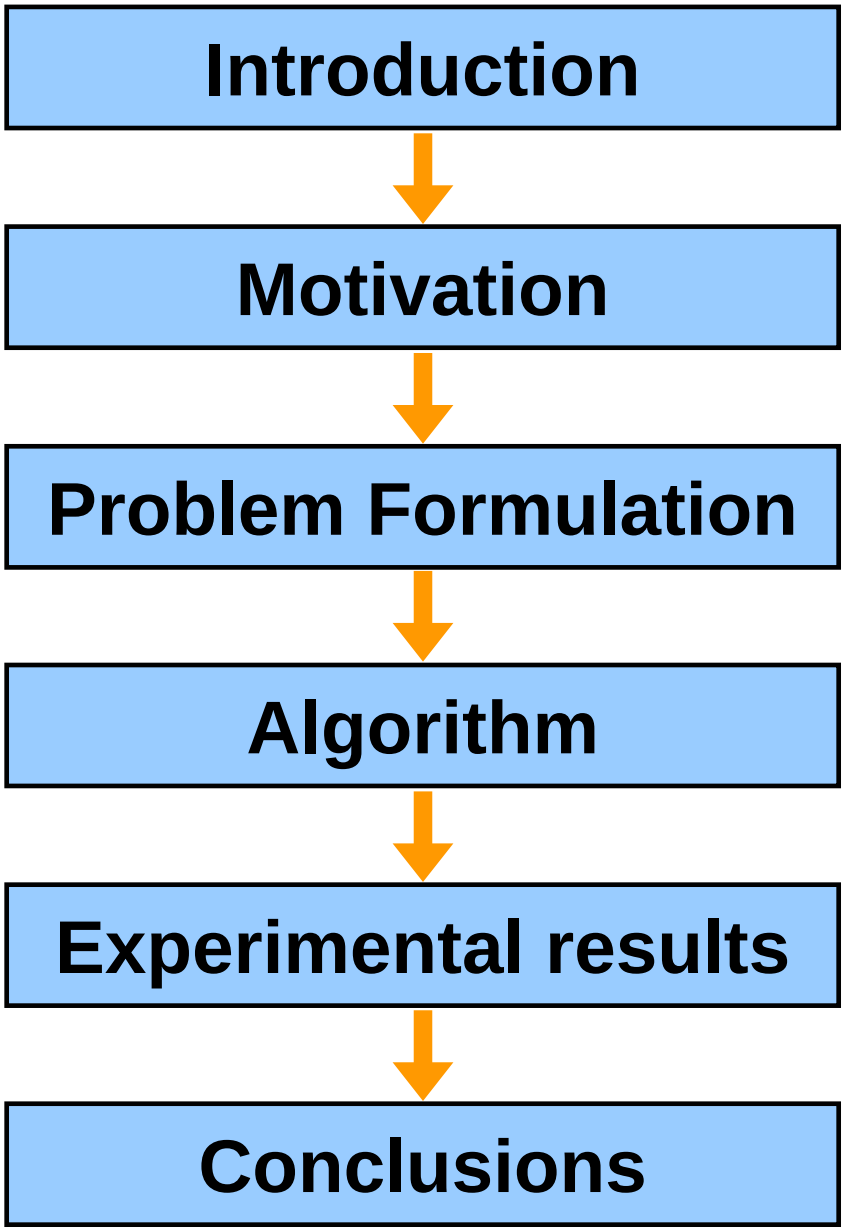
Robust Layer Assignment for Via Optimization in Multi-layer Global Routing

Tsung-Hsien Lee
Inst. of Information Science
Academia Sinica

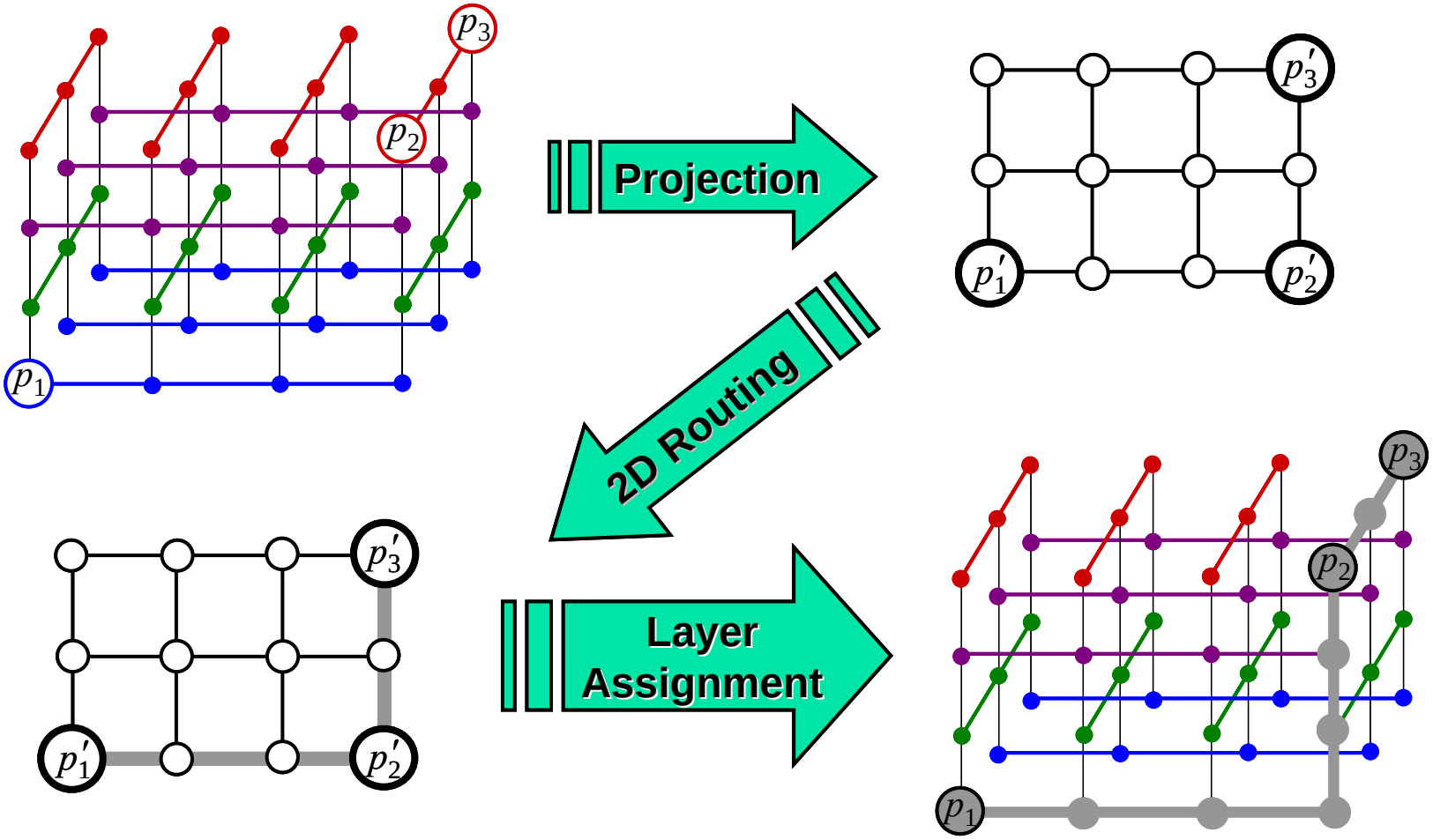
Ting-Chi Wang
Dept. of Computer Science
National Tsing Hua University

Outline


Introduction
Motivation
Problem Formulation
Algorithm
Experimental results
Conclusions

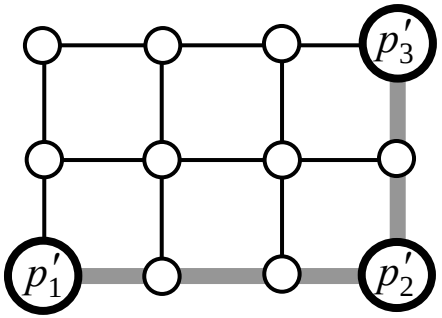
Introduction to Layer Assignment

- **Layer assignment** is a major step in multi-layer global routing



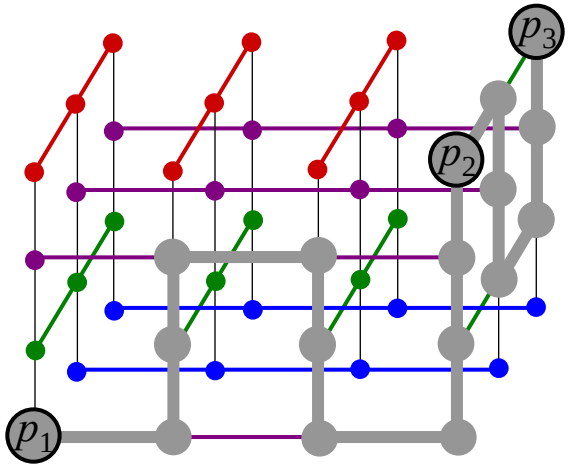
Introduction to Layer Assignment

- **Layer assignment** determines the final routing result
  - **A bad layer** assignment devastates all the previous efforts



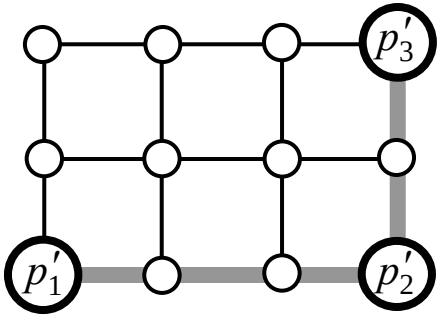
Additional wire overflow

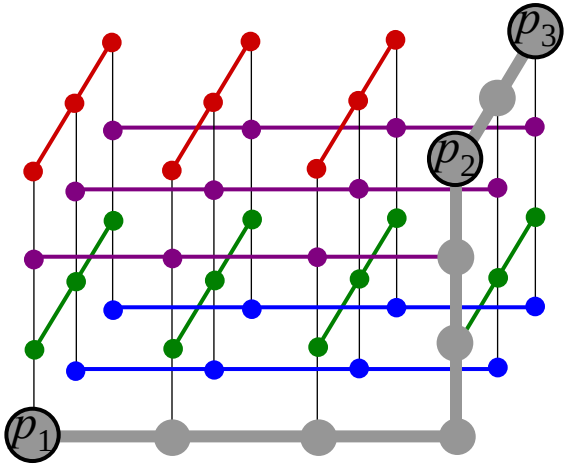
longer wirelength



Introduction to Layer Assignment

- **Layer assignment** determines the final routing result
  - **A good layer** assignment keeps all the previous efforts



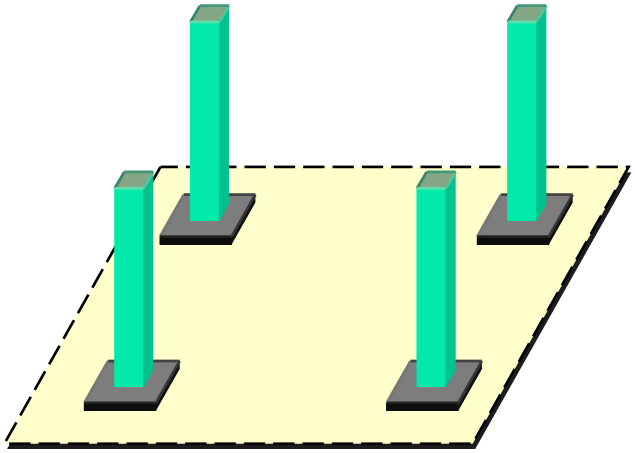


No additional wire overflow

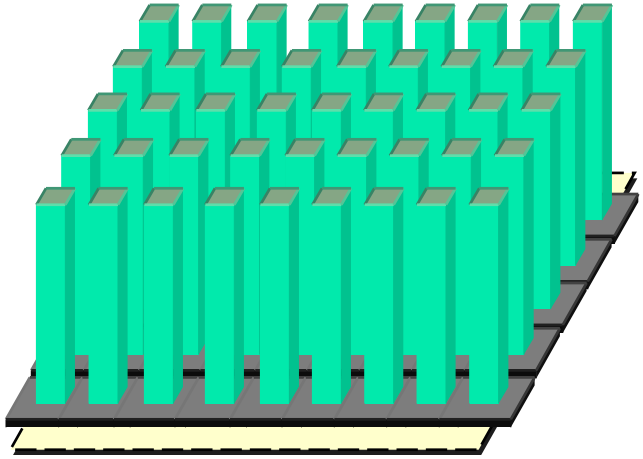
Minimal wirelength

Motivation

- ISPD’07 and ISPD’08 Global Routing Contest did not limit # of **vias** placed in a tile



Allowable in the contest

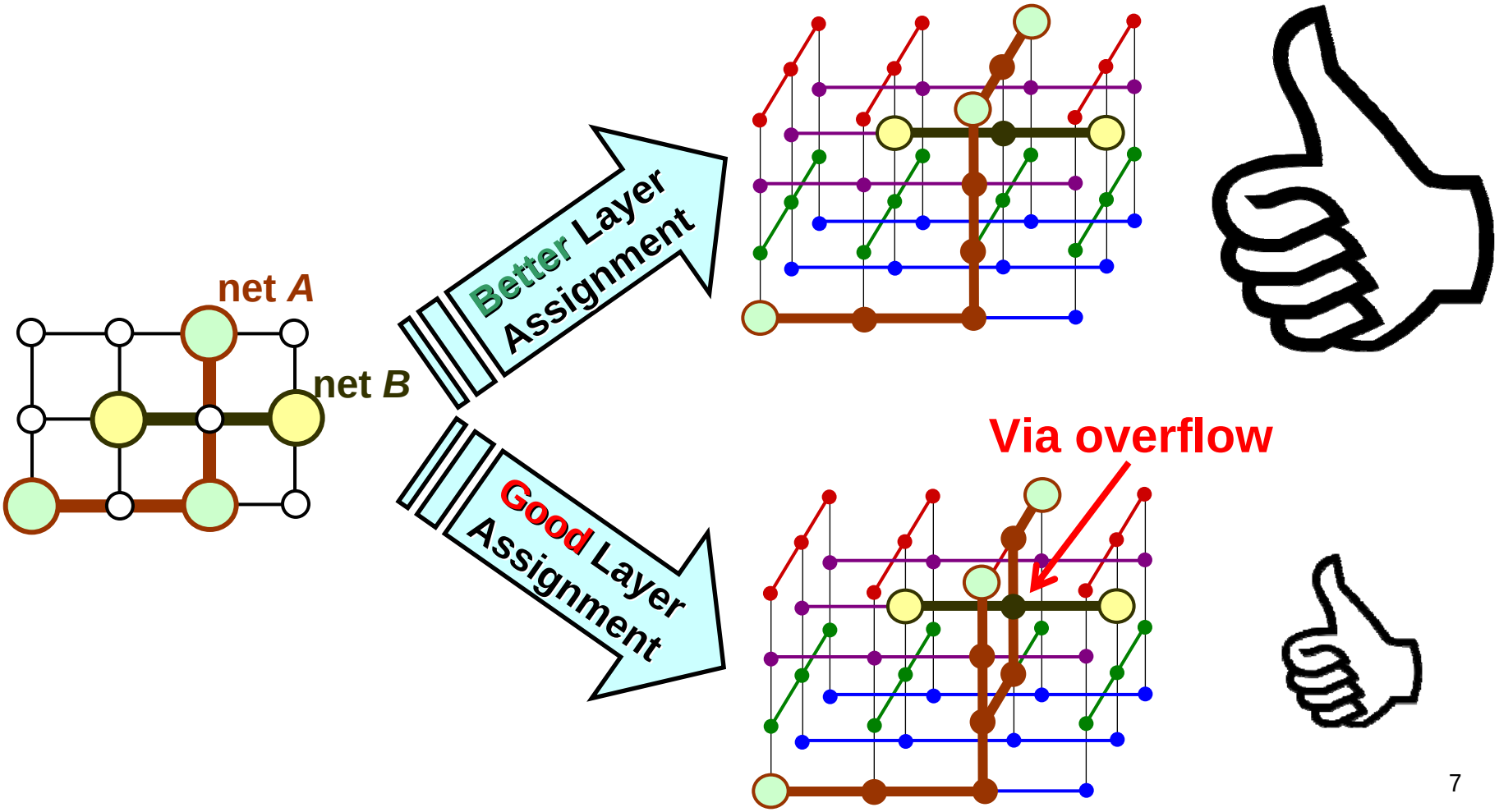


Still allowable in the contest

- Routing result without considering **via capacity** is not practical !

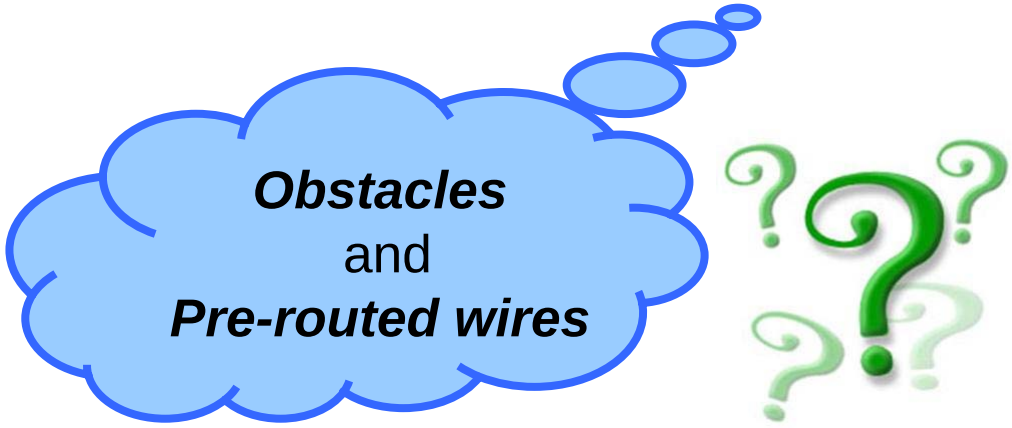
Motivation

- A better layer assignment should take the **via capacity** into account



Previous Work for Via Capacity

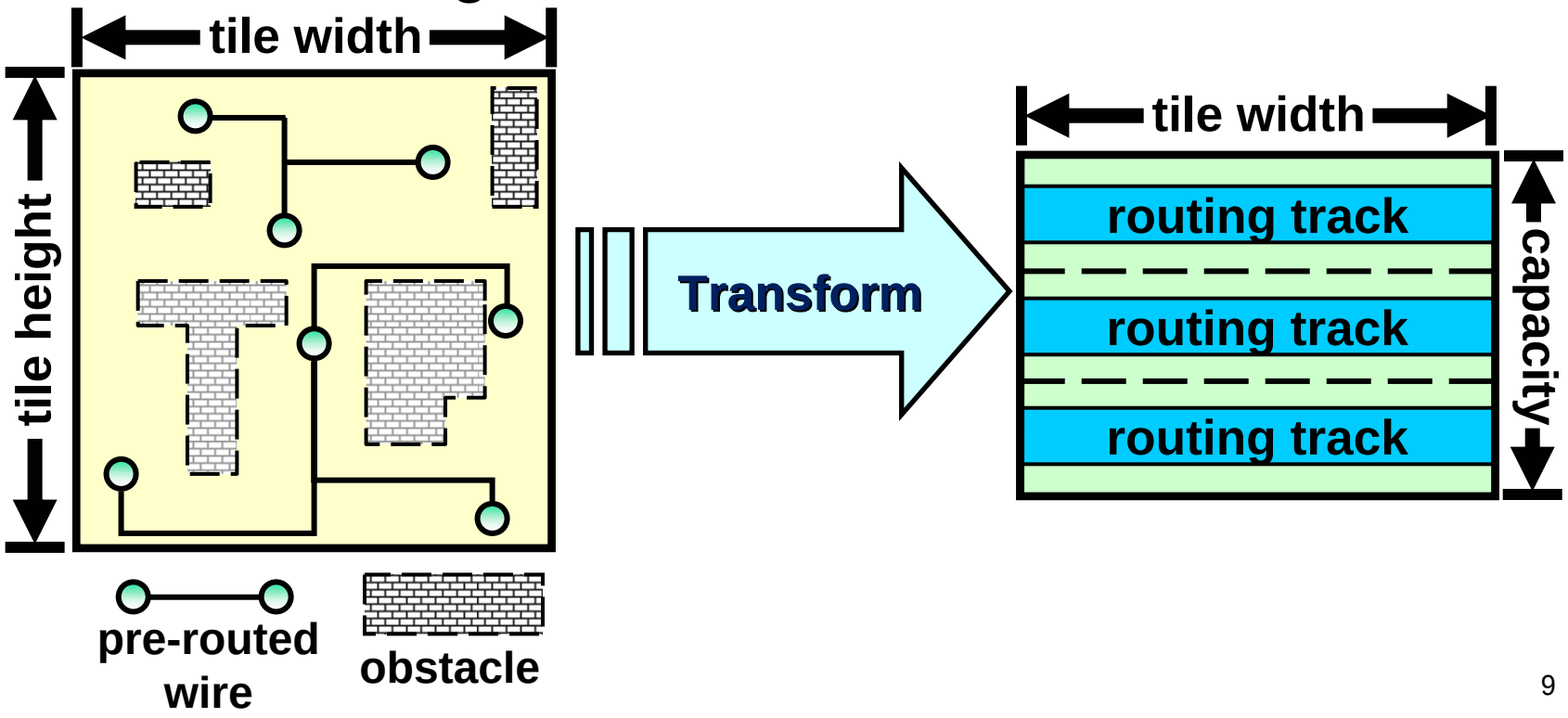
- **[Hsu et al., ICCAD'08]**
  "*Multi-layer Global Routing Considering Via and Wire Capacities*"
  - Considering via capacity for each tile
  - No detailed information of its layer assignment step
- Via capacity of a tile
  = *remaining_area / via_area*
  = (*tile_area* – ***preoccupied_area***) / *via_area*



Didn't specify in the benchmarks

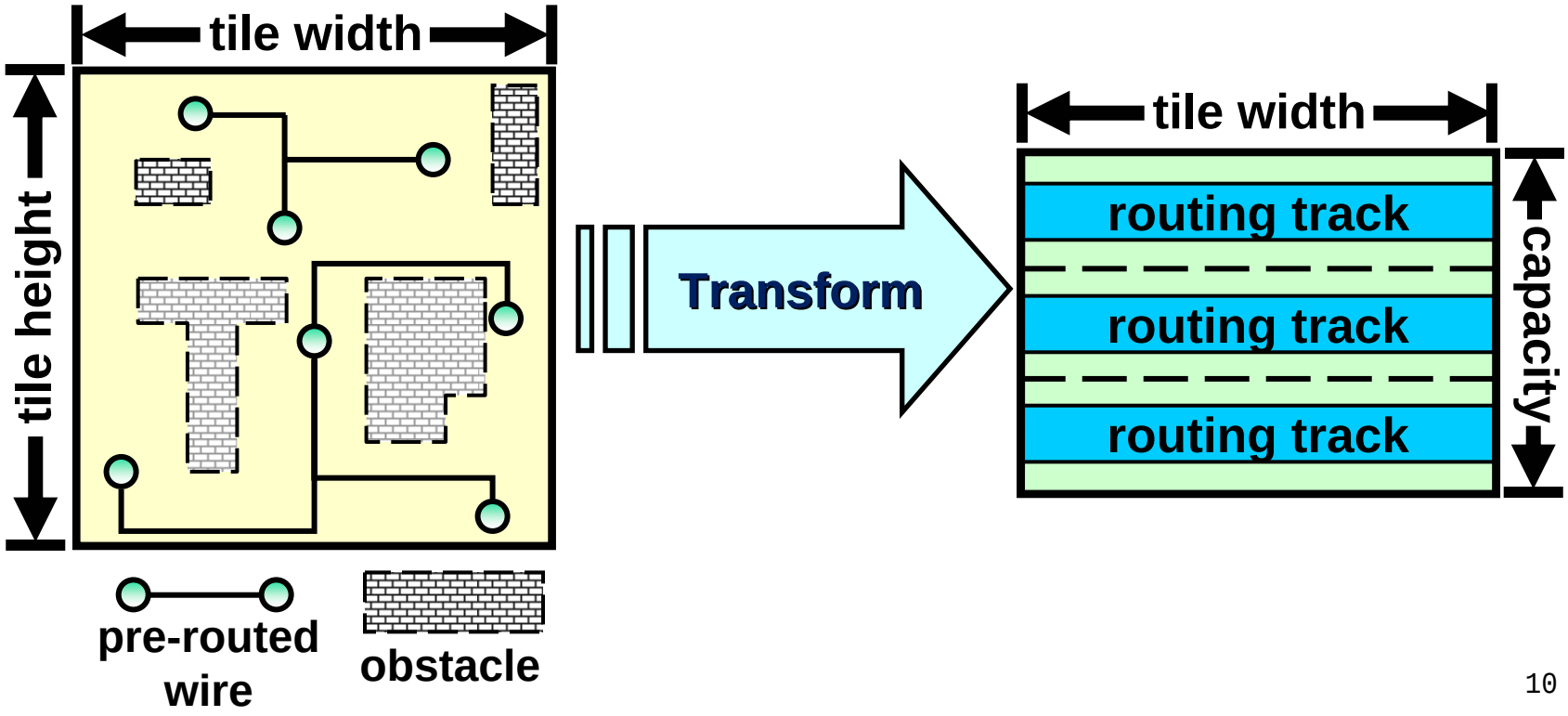
Problem Formulation

- "***Congestion is modeled by including capacity adjustments. In the global routing benchmarks, there may be obstacles, or pre-routed wires.***" — quoted from "details of file formats" of ISPD'08 Global Routing Contest rules



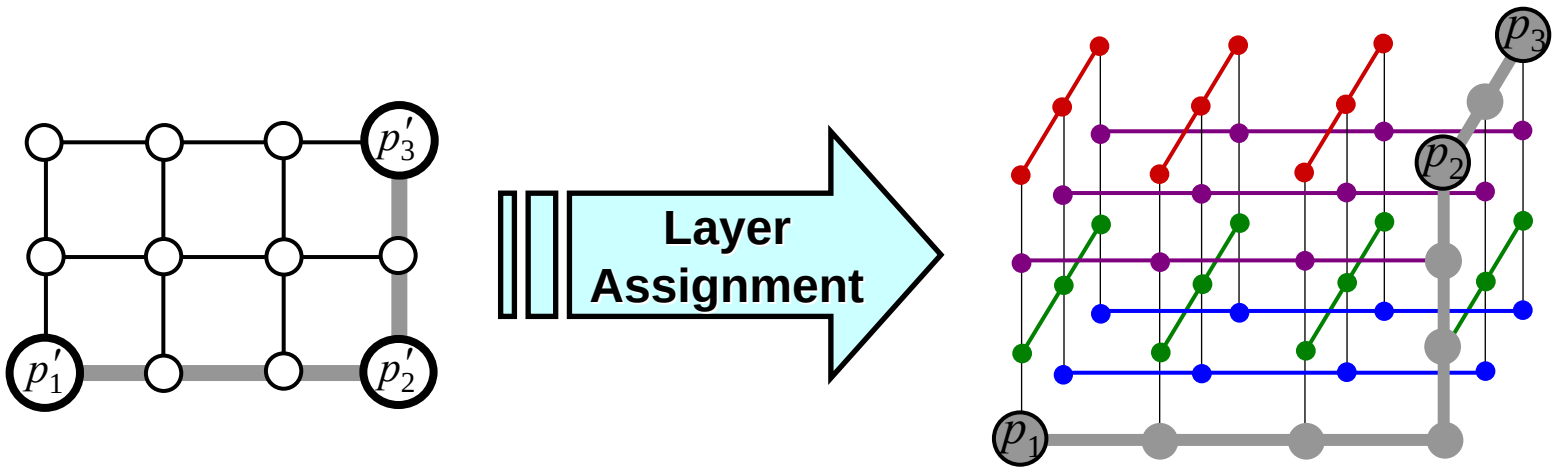
Problem Formulation

- Via capacity of a tile
  - = (*tile_area – preoccupied_area*) / *via_area*
  - = (***capacity*** × ***tile_width***) / *via_area*



Problem Formulation

- Given a 2D routing result, finds a 3D counterpart through layer assignment
  - Minimize ***via overflow***, and ***wirelength***
  - Keep the same ***wire overflow*** from 2D routing result



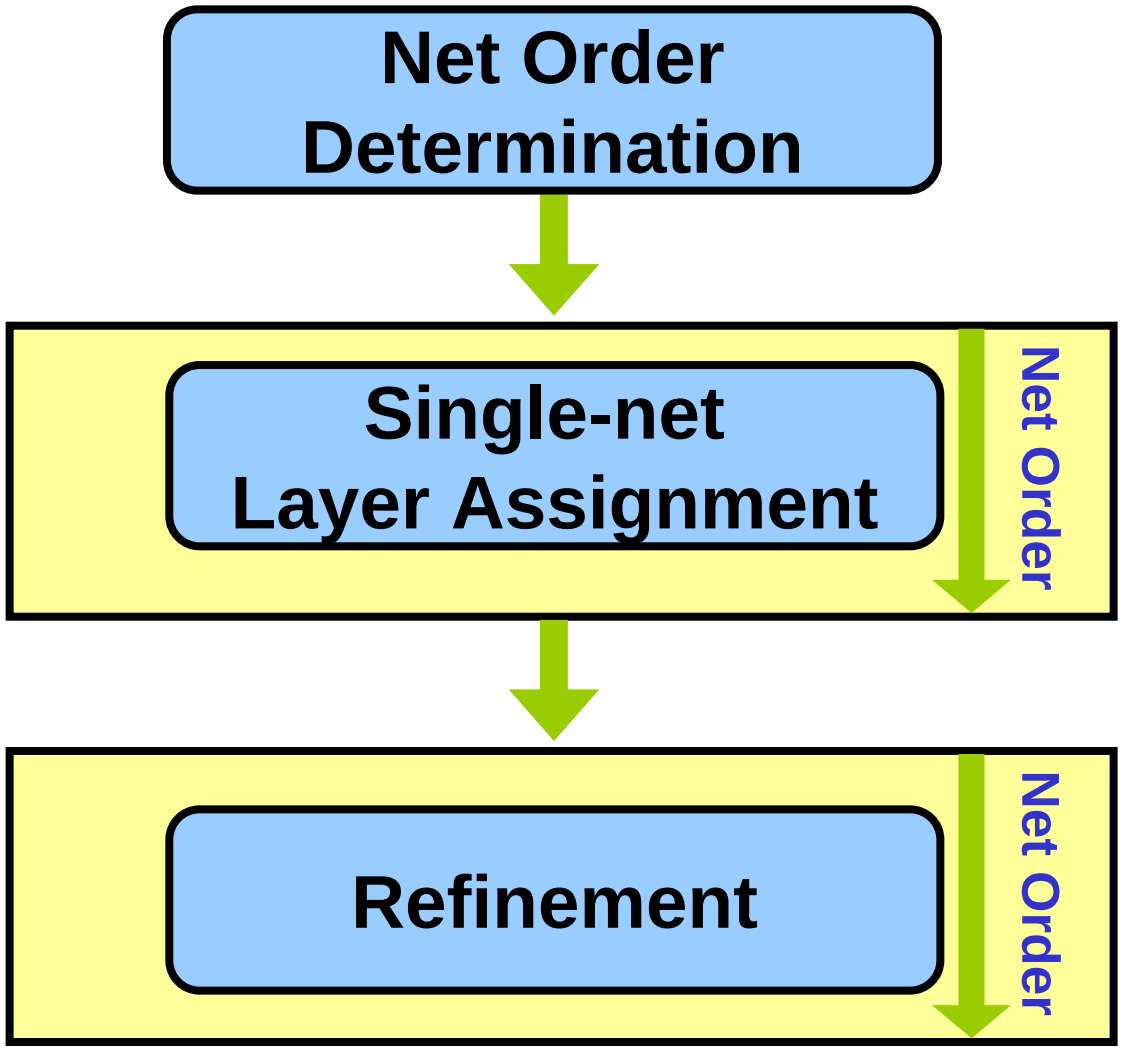
No additional wire overflow

Minimal via overflow

Minimal wirelength

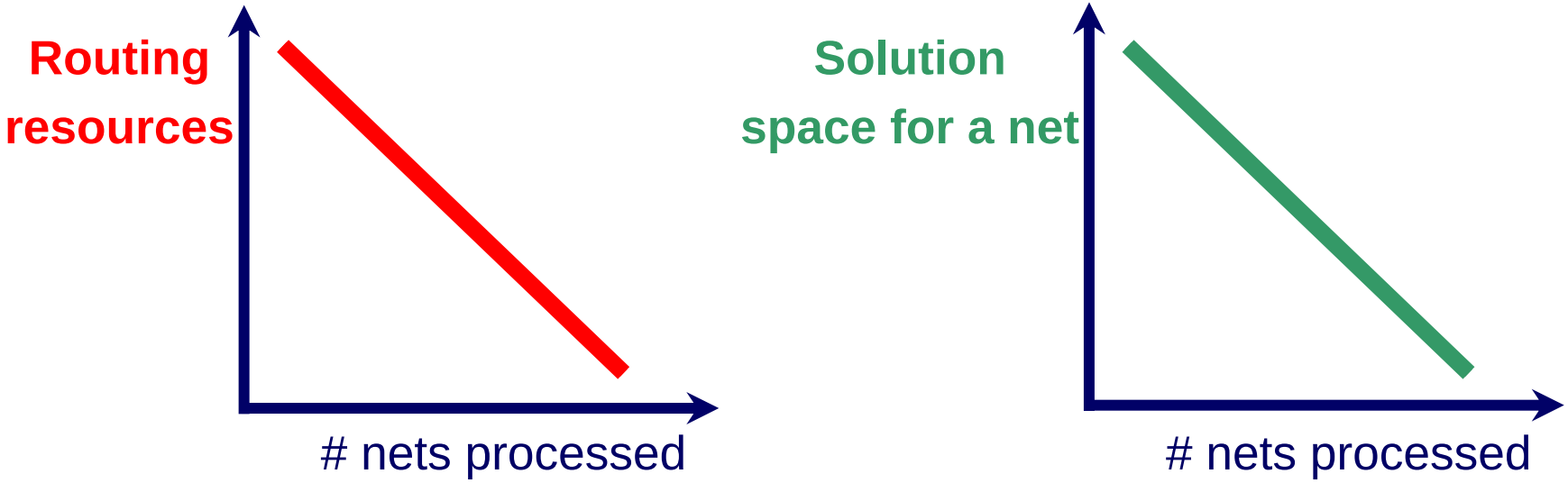
Algorithm

- Our algorithm contains 3 steps



Net Order Determination

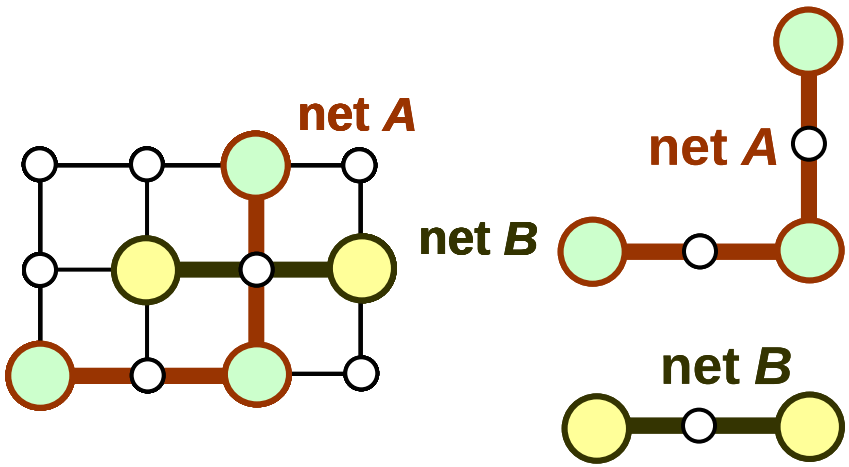
- Due to the **limited routing resources**, a net processed earlier has **larger solution space**



- **Net order** should maximize resource utilization

Net Order Determination

- For each 2D net *T*, we use 3 parameters to determine its order — ***Score*(*T*)**
  - ***Length*(*T*)** : # of **edges** in *T*
  - ***PinNum*(*T*)** : # of **pins** in *T*
  - ***Bends*(*T*)** : # of **bends** in *T*
- **Net Order** derived from sorting *Score*(*T*) for each net *T* **decreasingly**

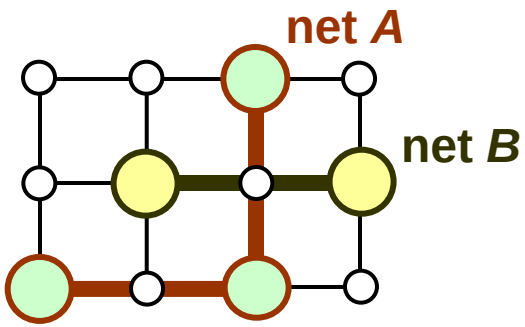


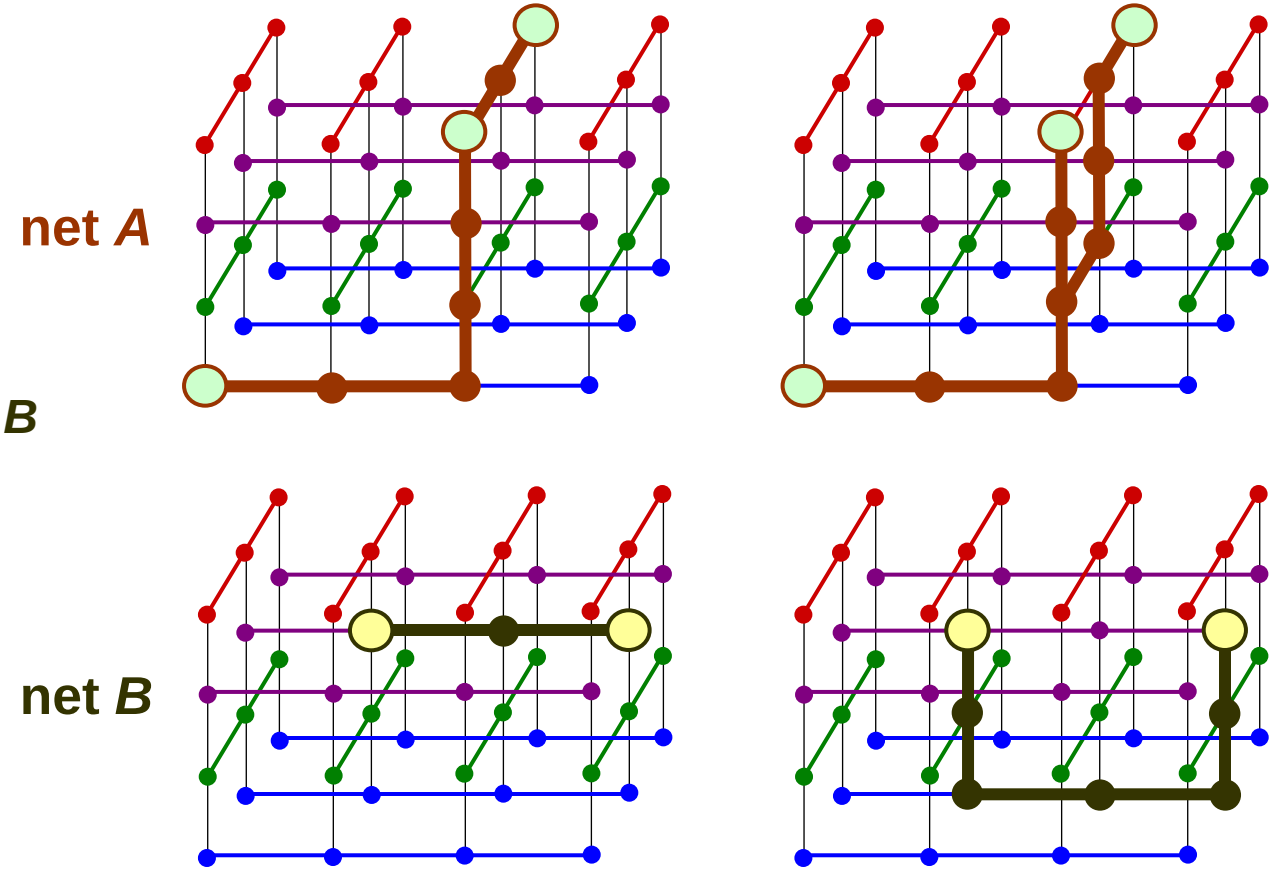
A **has 4 edges, 3 pins, and 1 bend**

B **has 2 edges, 2 pins, and 0 bend**

Net Order Determination — *Length*

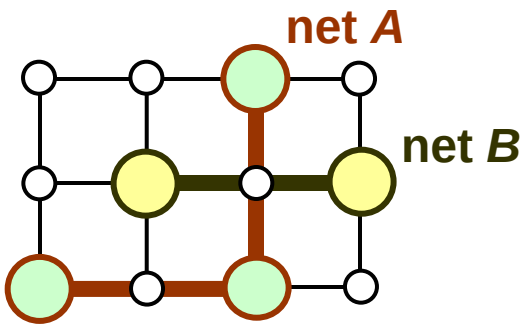
- A net with **longer length** will **occupy more routing resources**
  - ***Length*(*T*) ↑, *Score*(*T*) ↓**

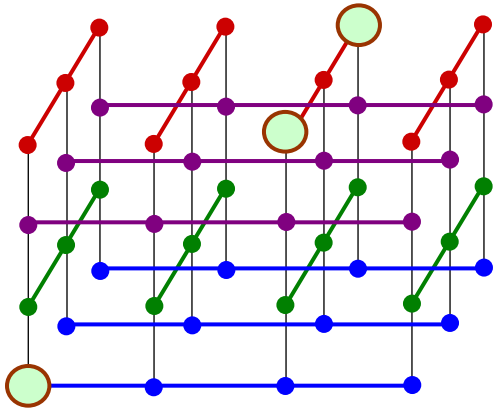




Net Order Determination — *PinNum*

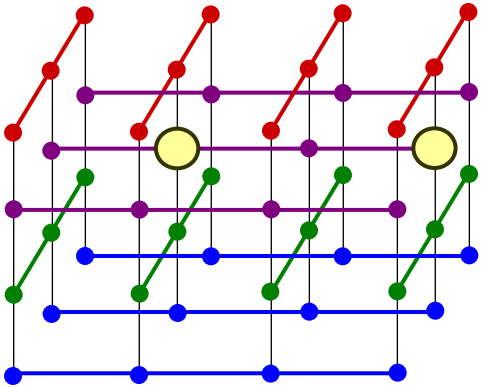
- The role of **pin** in global routing, just like the role of **checkpoints** in race
  - ***PinNum*(*T*) ↑, *Score*(*T*) ↑**





net *A*

3 pins = 3 checkpoints

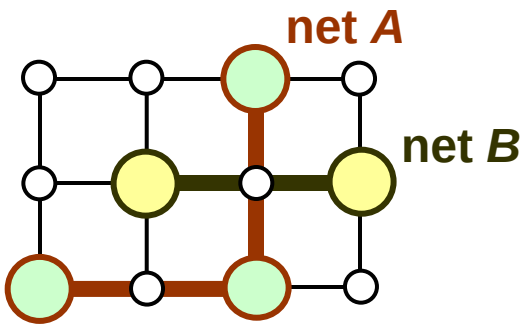


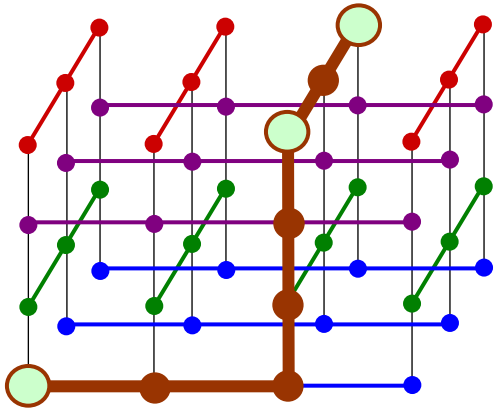
net *B*

2 pins = 2 checkpoints

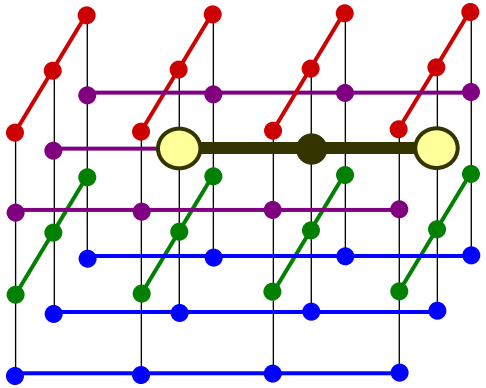
Net Order Determination — *Bends*

- **Changing routing direction needs vias**, so bend is similar to pin
  - ***Bends*(*T*) ↑, *Score*(*T*) ↑**





1 bend at least needs an via



0 bend may not need vias

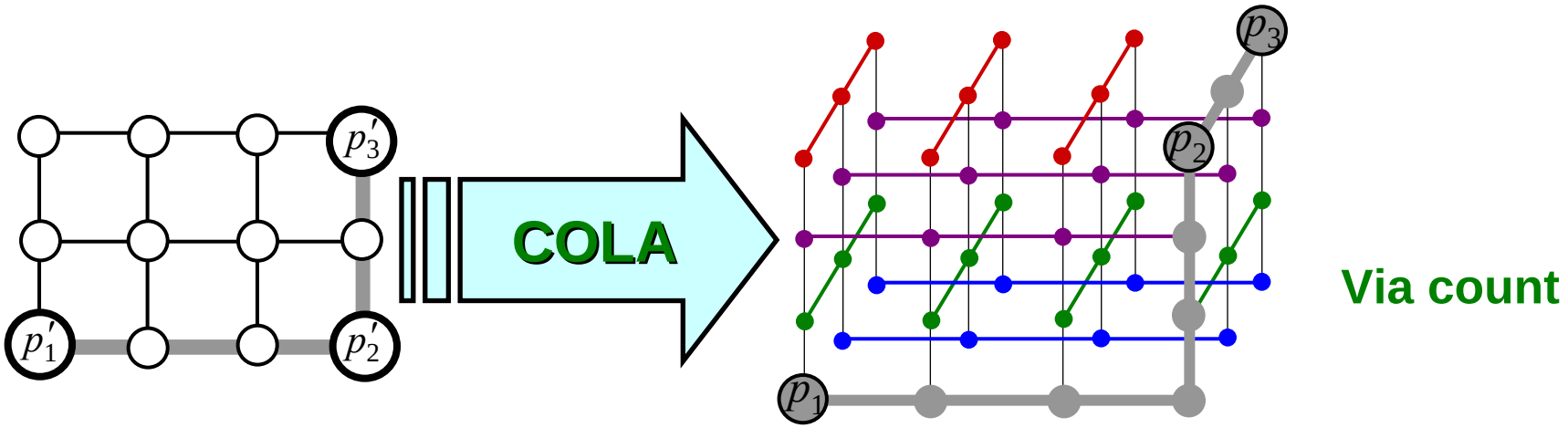
Net Order Determination

- *Score*(*T*)'s relationships with *Length*(*T*), *PinNum(T*), and *Bends*(*T*).
  - ***Length*(*T*) ↑ , *Score*(*T*) ↓**
  - ***PinNum*(*T*) ↑ , *Score*(*T*) ↑**
  - ***Bends*(*T*) ↑ , *Score*(*T*) ↑**
- ***Score*(*T*)**

  = ($\alpha \times$ *Bends*(*T*) + $\beta \times$ *PinNum*(*T*)) / *Length*(*T*)

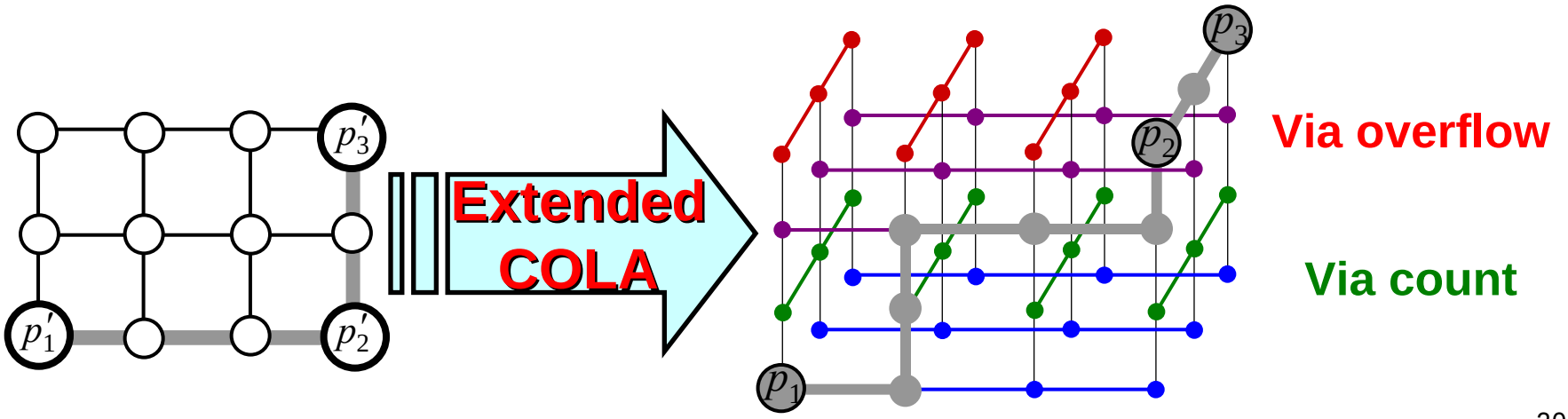
Single-net Layer Assignment

- **[Lee et al., TCAD'08]**
  "*Congestion-Constrained Layer Assignment for Via Minimization in Global Routing*"
  - COLA finds a layer assignment result with minimum via count for a 2D net



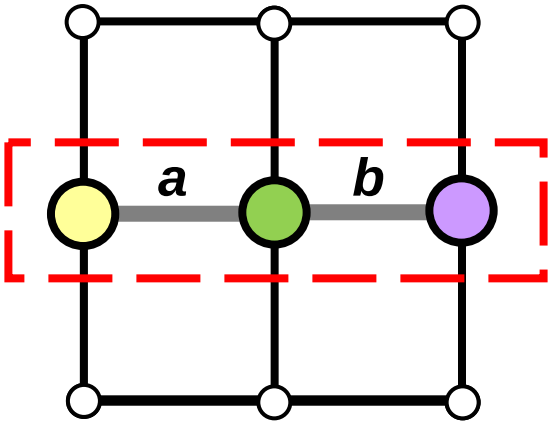
Single-net Layer Assignment

- **[Lee et al., TCAD'08]**
  "*Congestion-Constrained Layer Assignment for Via Minimization in Global Routing*"
  - COLA finds a layer assignment result with minimum via count for a 2D net
- Extends from COLA, our algorithm can deal with via overflow and via count

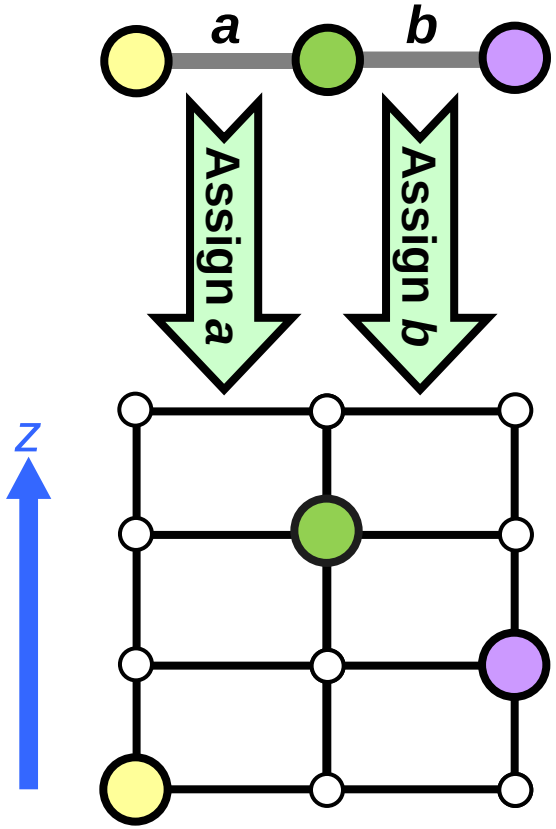


Single-net Layer Assignment

- DP-based layer assignment method
  - Minimize increase on ***via overflow***, and ***via count***
- For a net, assign one edge at a time



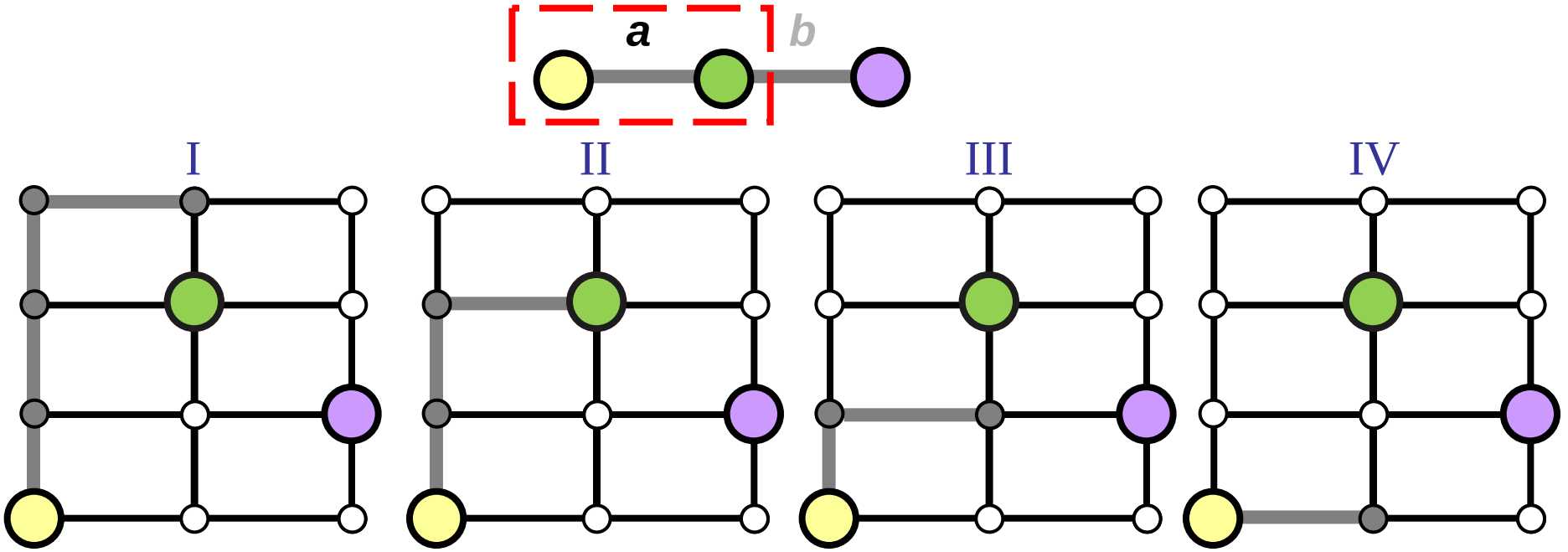
2D view



3D view

Single-net Layer Assignment

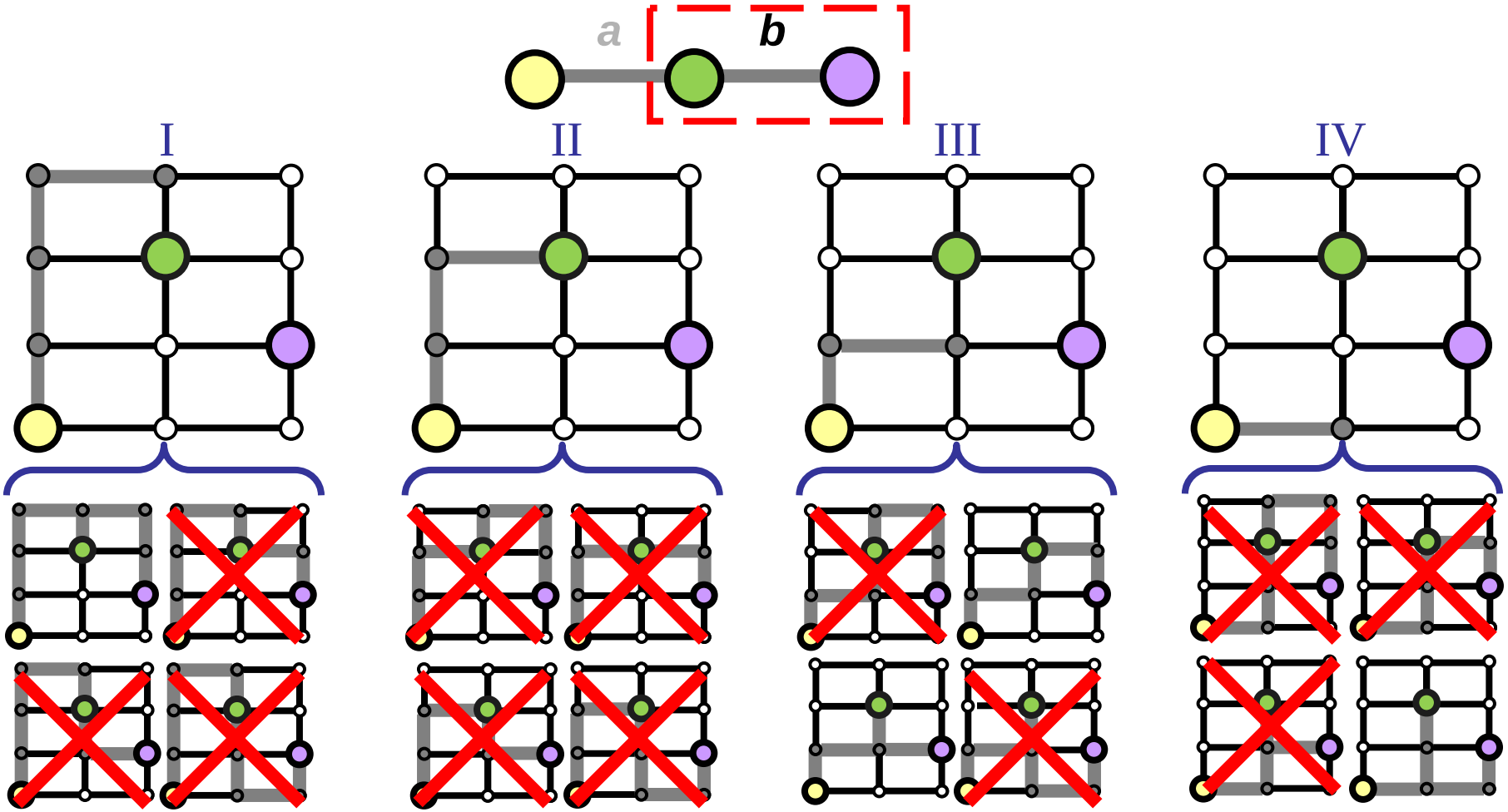
- Vias are placed after edges are assigned
  - Vias are determined by ***edges*** and ***pins***



- Since vias on different tiles are independent, the ***via overflow increase*** on each tile can be calculated independently

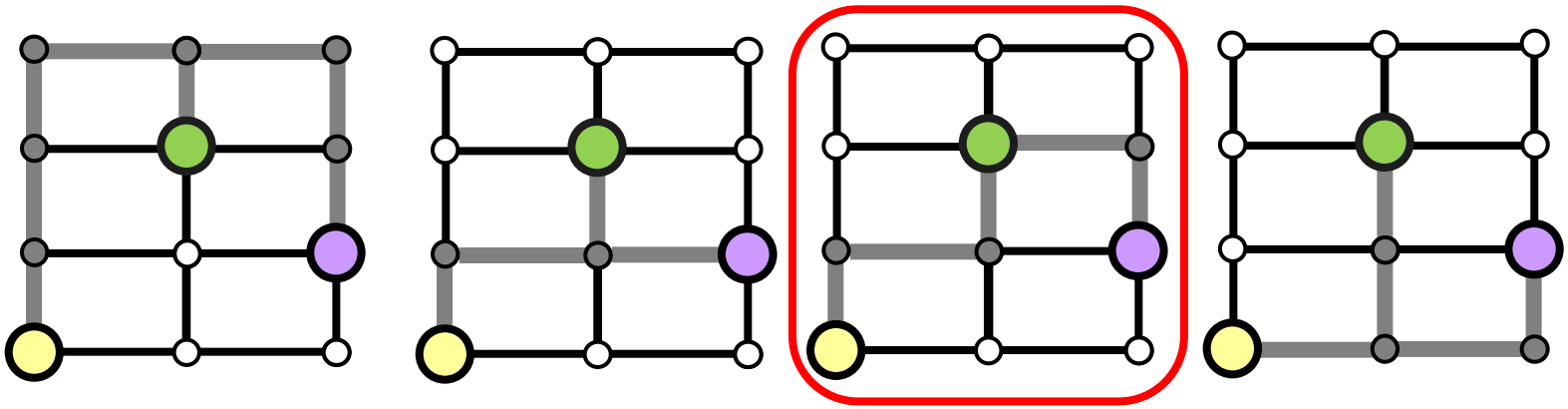
Single-net Layer Assignment

- Memorize the **minimum via overflow** one on each stage and propagate it to the next stage



Single-net Layer Assignment

- In the end, the optimal result is the one with **minimum total via overflow increase**

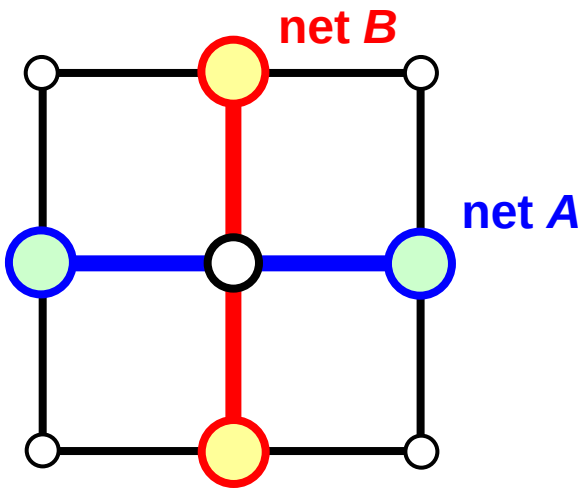


Optimal result

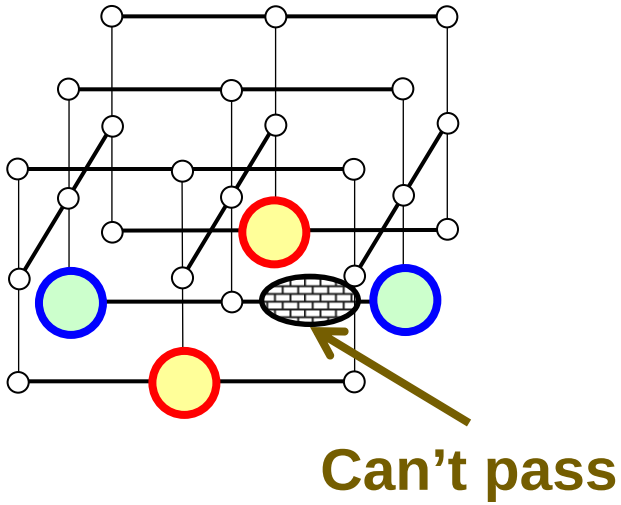
- If there is a tie on via overflow increase, choose the one with **minimum via count**

Refinement

- Refinement can improve the via overflow of the original layer assignment result
  - Rip-up and re-layer assignment for each net
- Enhance the net order



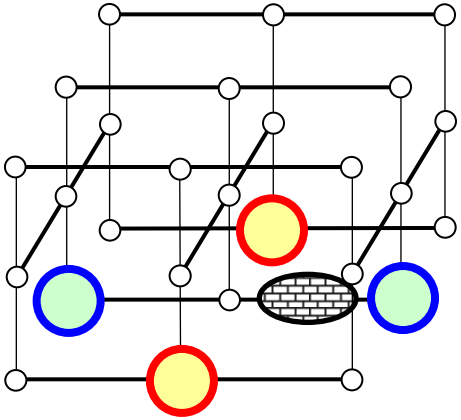
2D view

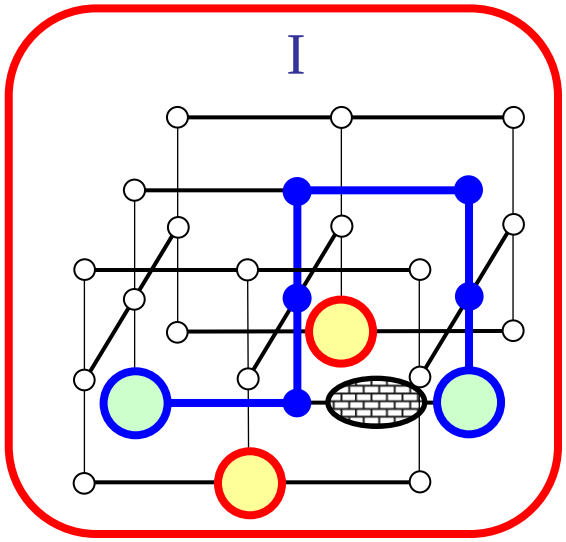


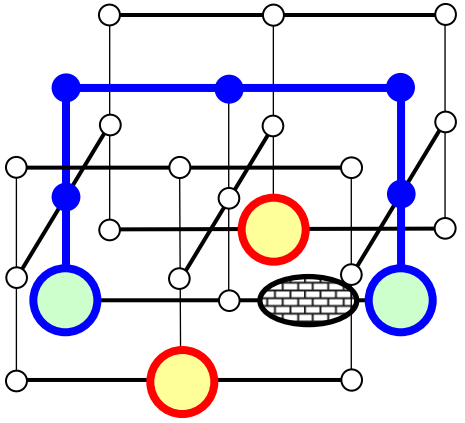
3D view

Refinement

- Net *A* assigned first, and net *B* assigned second
  - Net *A* has 2 choices with the same via overflow increase and via count

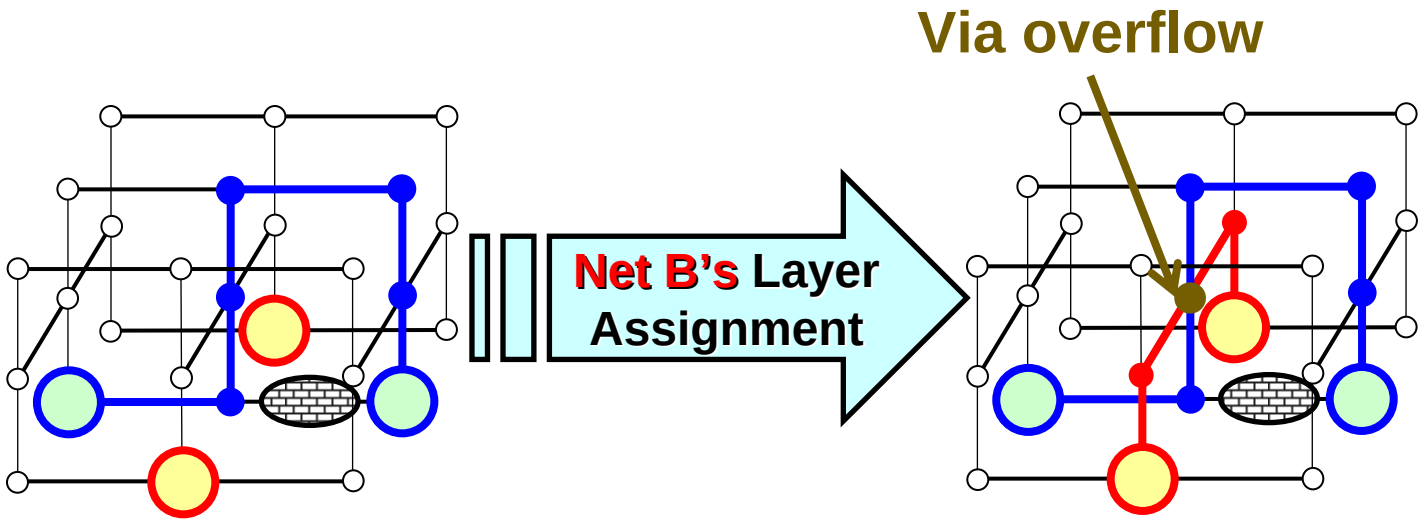






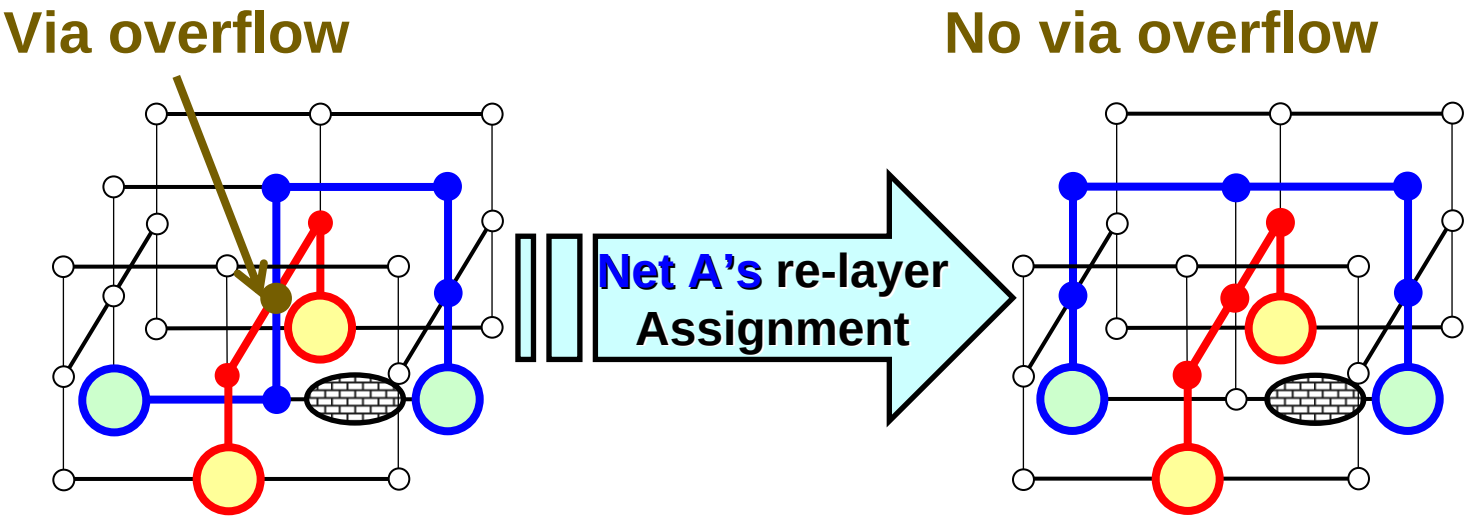
Refinement

- If net *A* chooses improperly, net *B* will generate via overflow inevitably
  - It is impossible for net *A* to know how to choose before net *B* is assigned

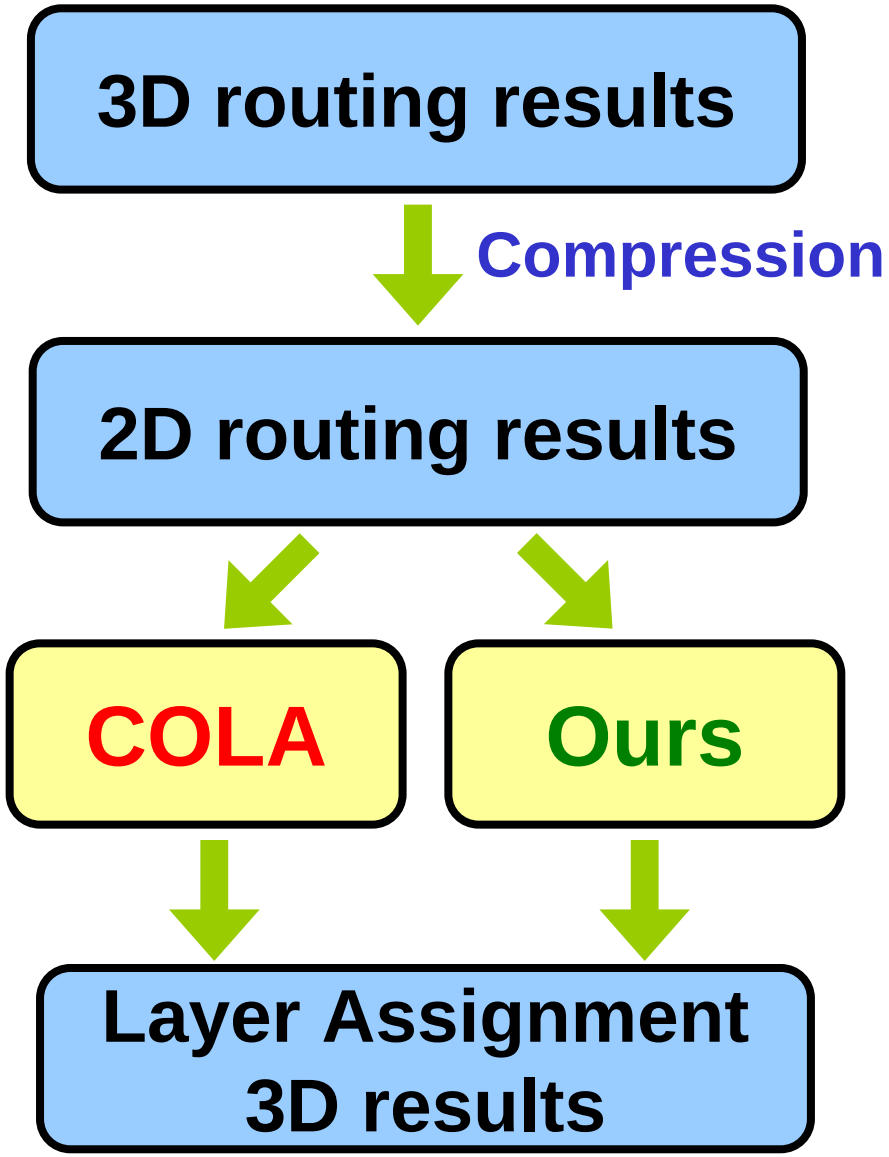


Refinement

- With refinement, the via overflow will be improved by the re-layer assignment of net *A*
  - The re-layer assignment will not generate worse result than pervious

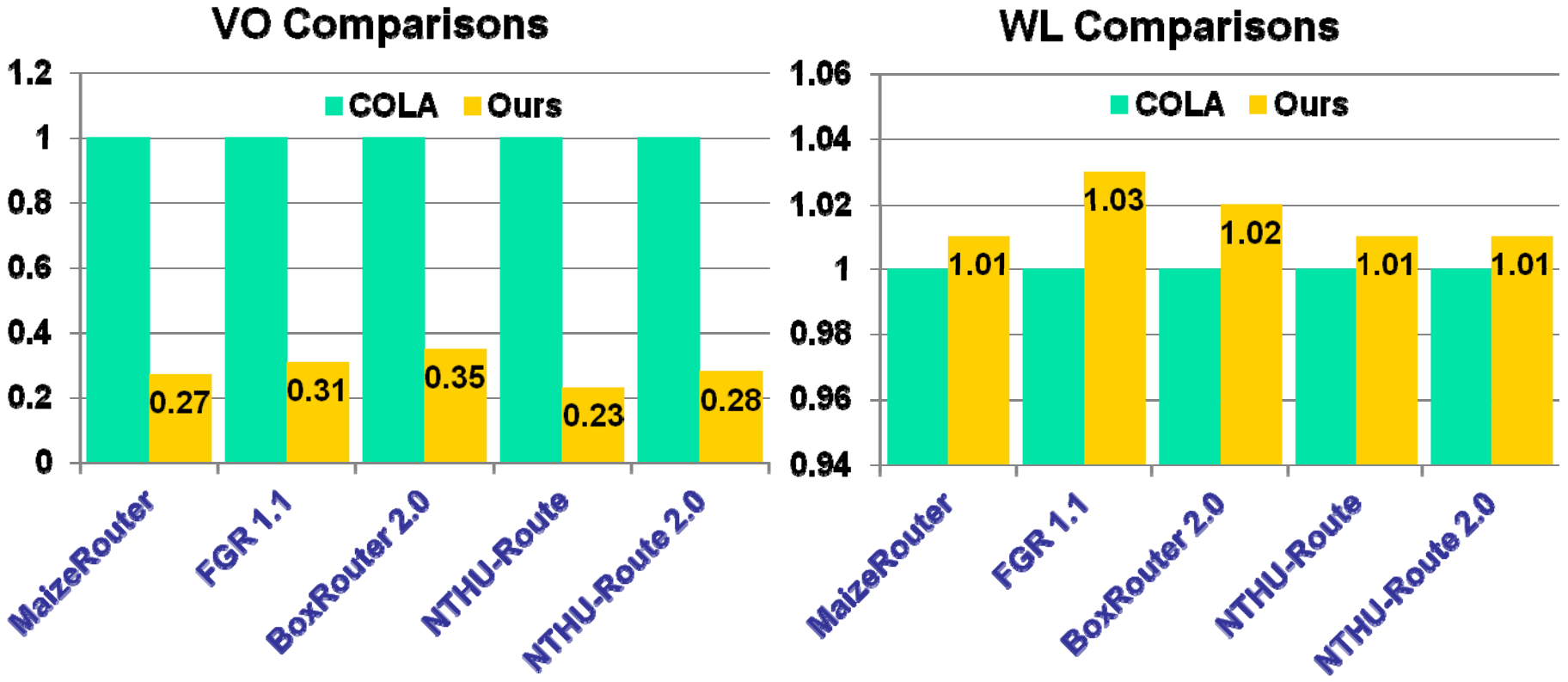


Experiment Flow



Experimental Results without Refinement

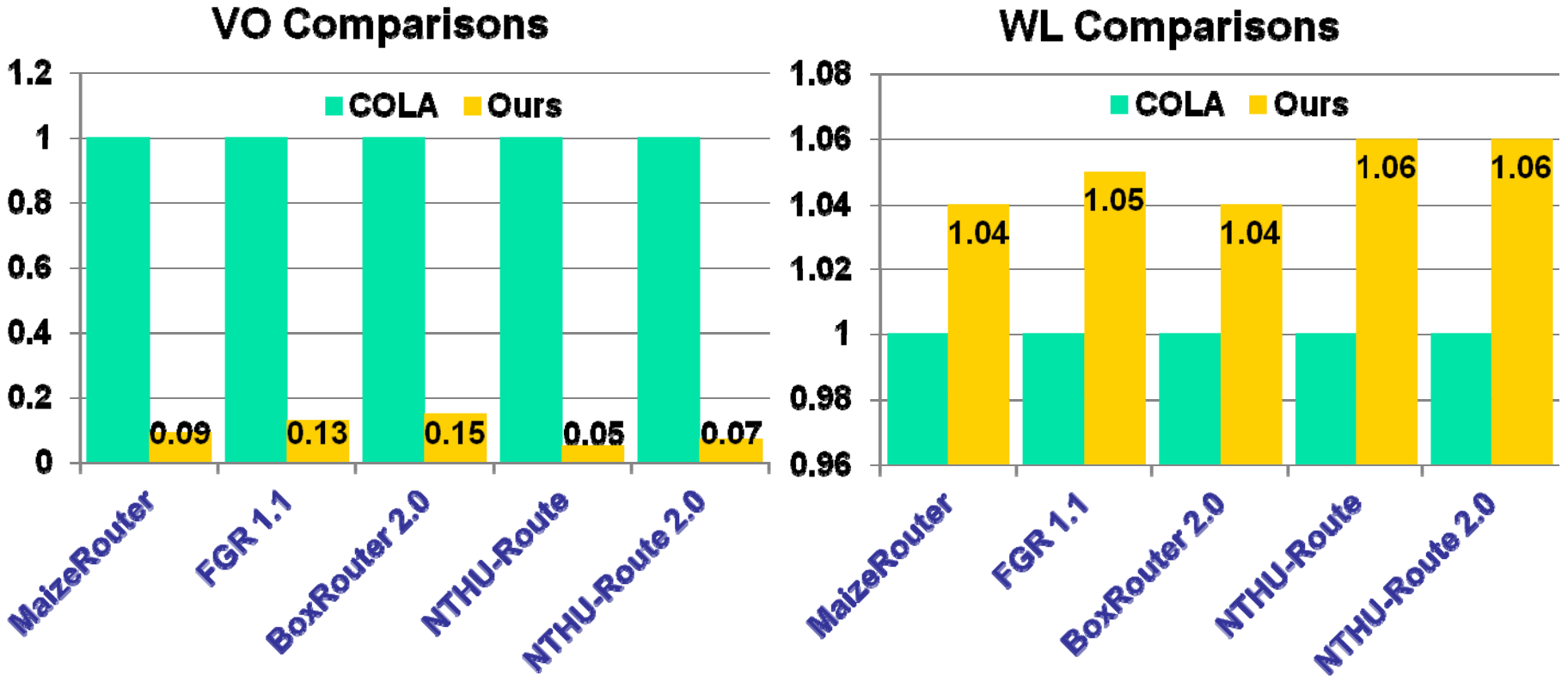
- **Without refinement**, our algorithm induced **23~35%** via overflow with **1~3%** WL increase compared with COLA [1]



[1] Lee et al., TCAD'08, "*Congestion-Constrained Layer Assignment for Via Minimization in Global Routing*"

Experimental Results with Refinement

- **With refinement**, our algorithm induced **5~15%** via overflow with **4~6%** WL increase compared with COLA [1]



[1] Lee et al., TCAD'08, "*Congestion-Constrained Layer Assignment for Via Minimization in Global Routing*"

Conclusions

- Develop a layer assignment algorithm considering **via overflow** and **via count**
- **Future work**: a more effective **net order** and **layer assignment method**

Q & A

Thank You!

and