Obstacle-Aware Length-Matching Bus Routing

Jin-Tai Yan† and Zhi-Wei Chen‡
† Department of Computer Science and Information Engineering
‡ College of Engineering

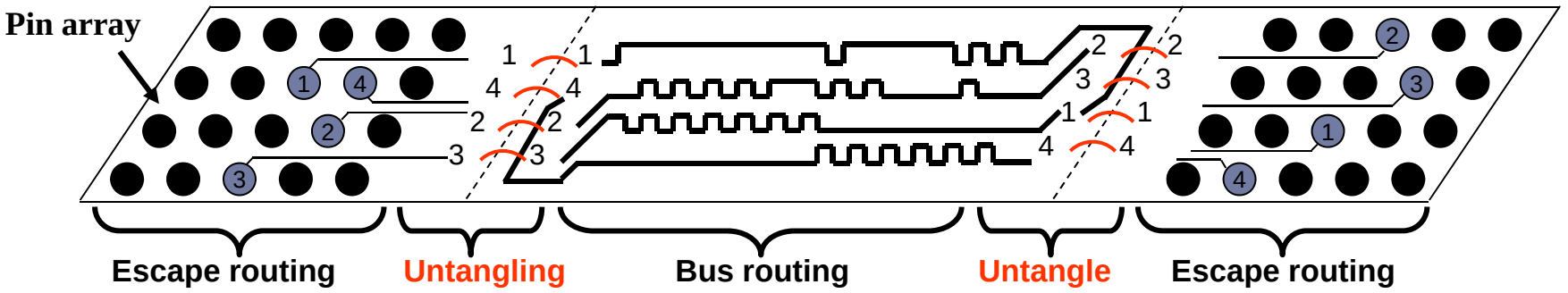
ISPD-2011

Outline

- Introduction
- Problem Formulation
- Obstacle-Aware Length-Matching Bus Routing
  - **Obstacle-Aware Region Partition in Routing Grids**
  - **Obstacle-Aware Shortest-Path Generation**
  - **Iterative Length-Matching Path Generation**
- Experimental Results
- Conclusions

Routing in PCB

- Three processes in PCB routing
  1. Escape Routing
  2. Untangling Processing
  3. Bus Routing



PCB Routing Challenges

- More than thousands of pins are involved in modern PCBs.
- Physical “obstacles” on PCB
  - Component mounting pads: DIP / SMD
  - Pre-route predictions: analog / TTL design reservation
- Various design constraints
  - Layer constraints
  - Length matching: each routed net must be bounded within specific wire length

Source from : http://www.pc01.cn/

Previous Works

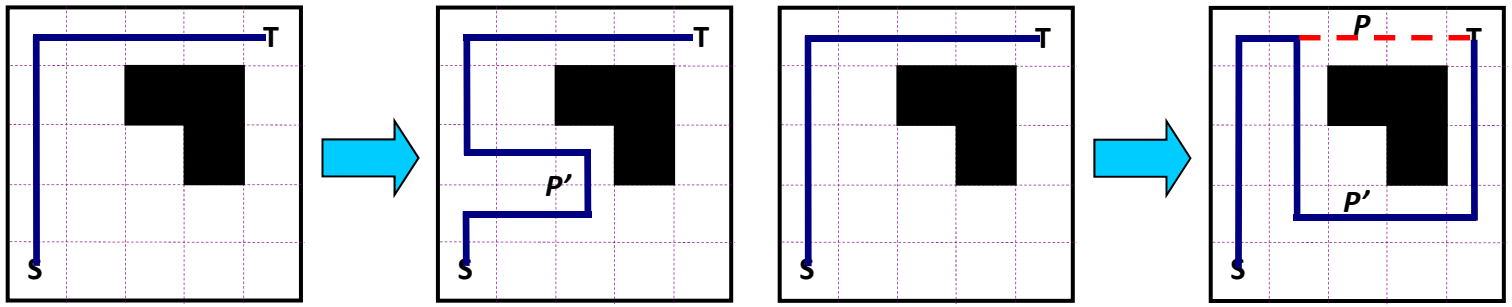
- **BGA-Route**
  - T. Yan and M. D. F. Wong, "BSG-Route: A length-matching router for general topology," *ICCAD,2008.*
  - **Limitations**: no obstacle constraint / remainder area in BSG cells
- **US Routing**
  - Y. Kohira et. al, "A fast longer path algorithm for routing grid with obstacles using biconnectivity based length upper bound," *ASPDAC,2009*
  - **Limitations**: high time complexity / optimal solution is not guaranteed
- **CAFE-router**
  - Y. Kohira et.al, "CAFE router: A Fast Connectivity Aware Multiple Nets Routing Algorithm for Routing Grid with Obstacles," *ASPDAC,2010*
  - **Limitations**: length-matching with high length error

Motivations

- Limitations in the previous works
  1. Without consideration of obstacles
  2. Length-matching with length error
- An efficient length-matching approach for bus routing
  1. Bus routing with considering obstacles
  2. More accurate result under the constraint of length-matching

Preliminary

- Routing area in a single layer can be represented by a set of routing grids
  - An arbitrary-shaped obstacle can be form by a set of routing grids (drawn by black)
- Grid-based routing path for a net
  - Two grids are specified as the locations of the start and target terminals, S and T
  - A path connects S and T by using horizontal or vertical segments and passes each available grid **at most once**
  - The ***length*** of a connecting path is defined as the number of passed grids in the routing path
- Flip operations for path detouring
  - **R-flip**: a partial path can be detoured by using adjacent routing grids and inserting a detouring path, P'
  - **C-flip**: a partial path, P, can be replaced by using another partial path, P', where the length of P' is larger than that of P



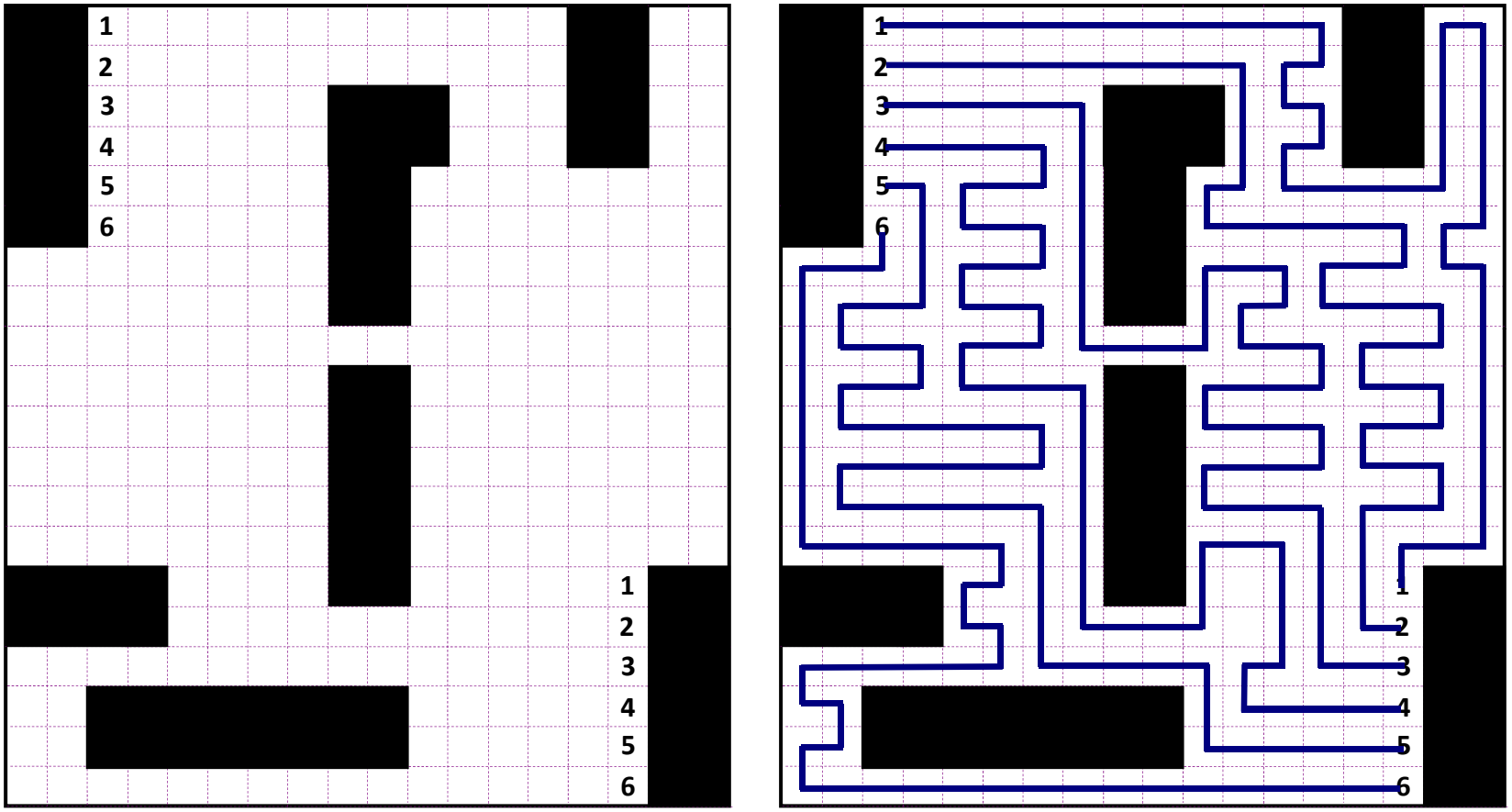
Problem Formulation

- **Input**
  - Two sequential sets of start and target terminals of ***r*** nets in a bus
    - Routing panel is divided into a ***mxn*** routing grids with ***s*** obstacle grids
    - Each net, i, has its connecting length of the shortest path, d_i
- **Constraints**
  - Length constraints, $l_1, l_2, \ldots, l_r$ are given to satisfy the timing constraints

$$\begin{cases} l_i < d_i \ \ or \ \sum_{i=1}^{r} l_i > mn - s & \Rightarrow \textbf{No Solution!!} \\ l_i \geq d_i \ \ or \ \sum_{i=1}^{r} l_i \leq mn - s & \Rightarrow \textbf{Solution available!!} \end{cases}$$

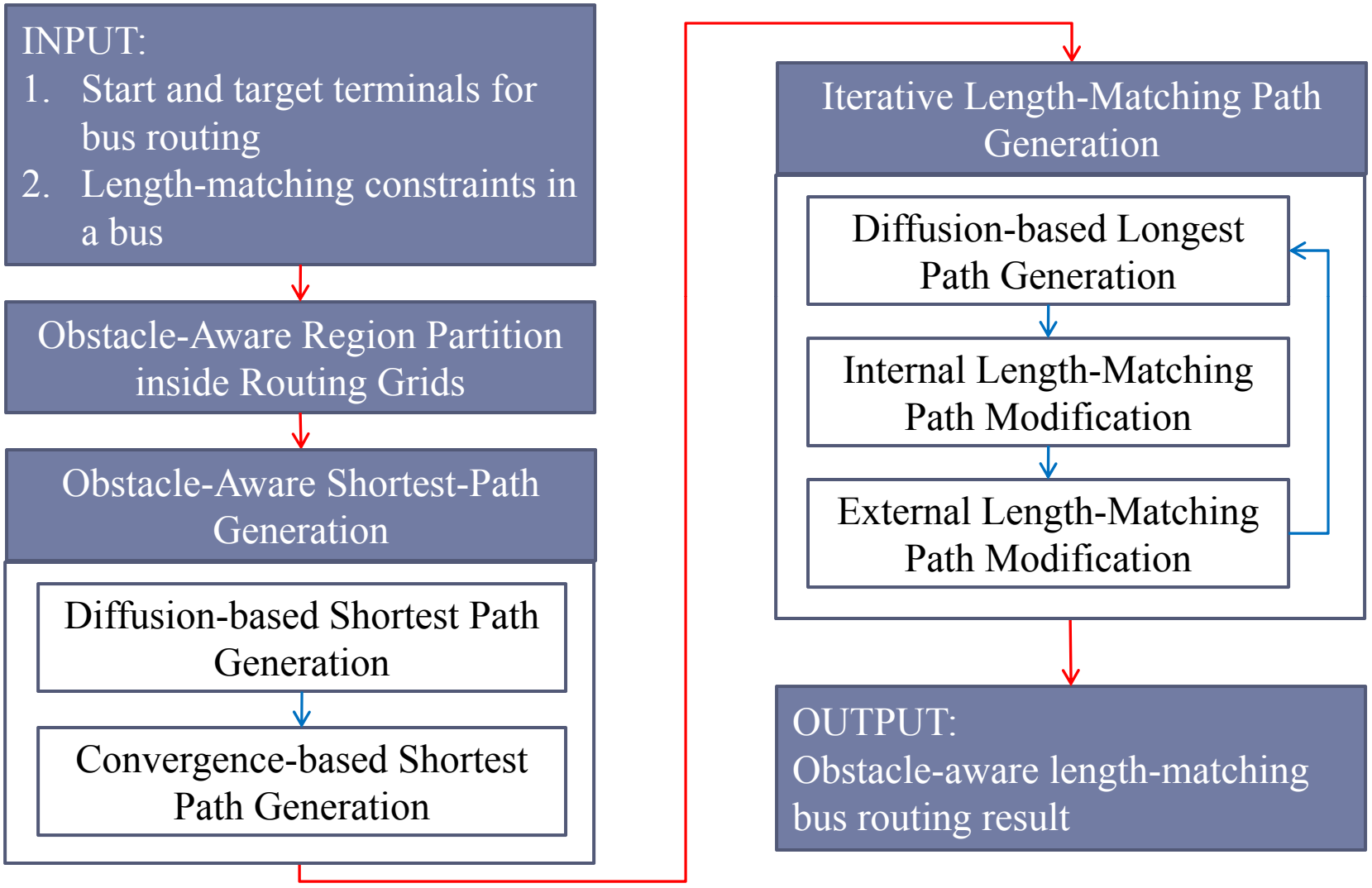
  - Non-crossing between any two pair of nets
- **Objective**
  - The length-matching paths of ***r*** nets are generated in the bus

Example



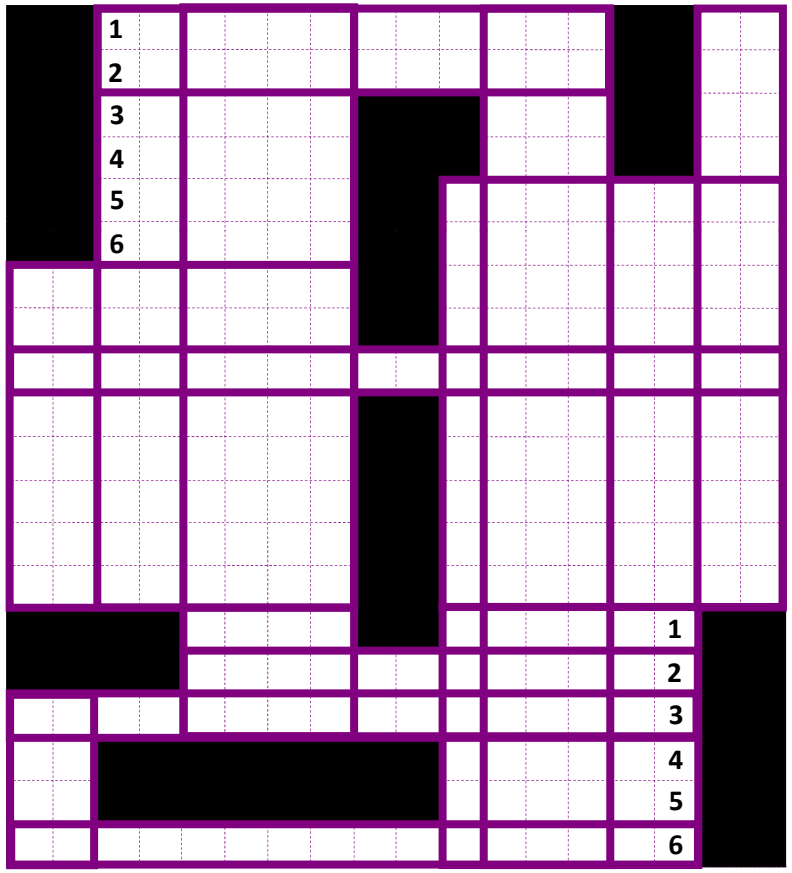
20x18 routing grids; **82** obstacle grids; length constraints = **46**
(d_1=28, d_2=28, d_3=28, d_4=28, d_5=,28 d_6=34**)**

Design Flow


INPUT:
1. Start and target terminals for bus routing
2. Length-matching constraints in a bus
Obstacle-Aware Region Partition inside Routing Grids
Obstacle-Aware Shortest-Path Generation
Diffusion-based Shortest Path Generation
Convergence-based Shortest Path Generation
Iterative Length-Matching Path Generation
Diffusion-based Longest Path Generation
Internal Length-Matching Path Modification
External Length-Matching Path Modification
OUTPUT:
Obstacle-aware length-matching bus routing result

Region Definition

- Outer Boundary : minimum covering rectangular boundaries
  - Including all the routing grids: available & obstacle grids
- **Extended lines** on hor. and vert. boundaries of obstacles
  - Independent region: hor. or vert. extended line
    1. Two obstacle boundaries
    2. One obstacle boundary and one outer boundary



Iterative Obstacle-Aware Region Partition

- Partition process

  Step 1: If all the routing grids are assigned into independent regions

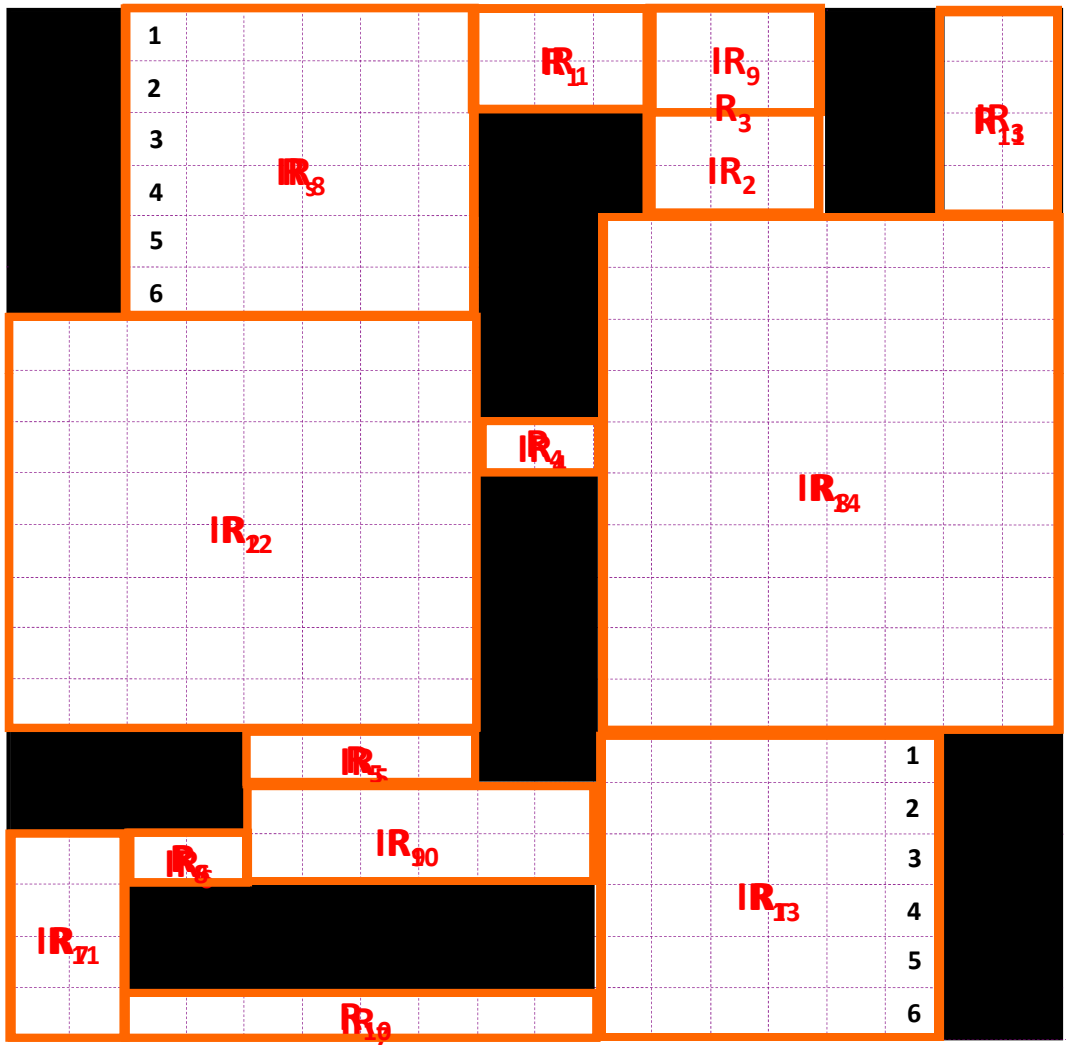
  ➔ process done

  Step 2: According to the extended lines of obstacles and outer boundary, independent region can be found.

  Step 3: Take the found independent regions as newly obstacles, go to Step 1

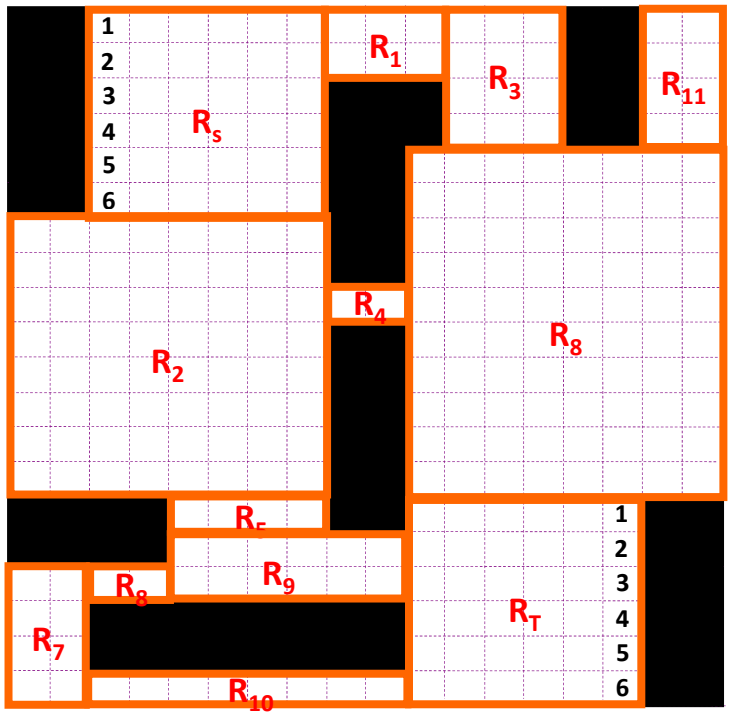
- Merging process
  - If any independent region is able to be horizontally or vertically merged with its adjacent independent region, merging with their adjacent regions
- Regions including start or target terminals are defined as start region and target region

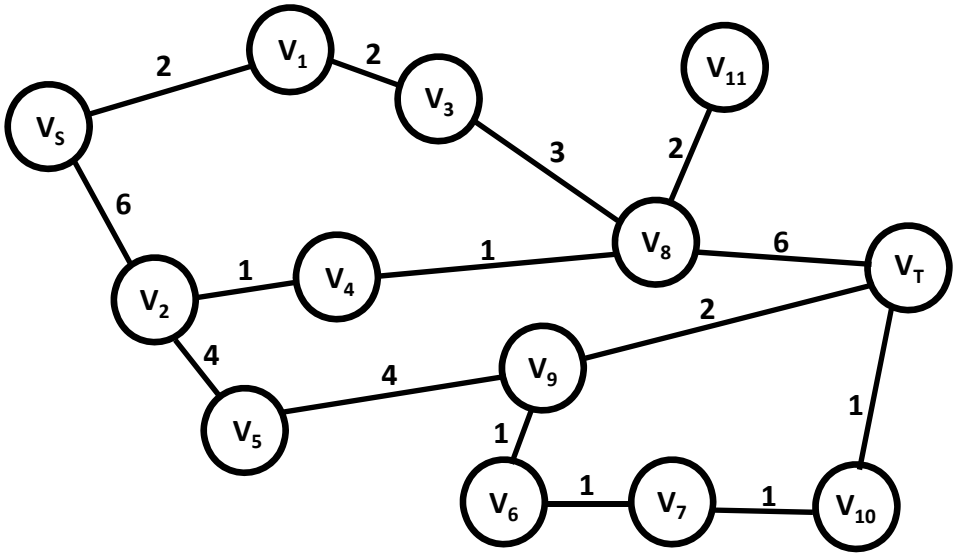
Partition & Merging Process



Adjacent Graph Construction

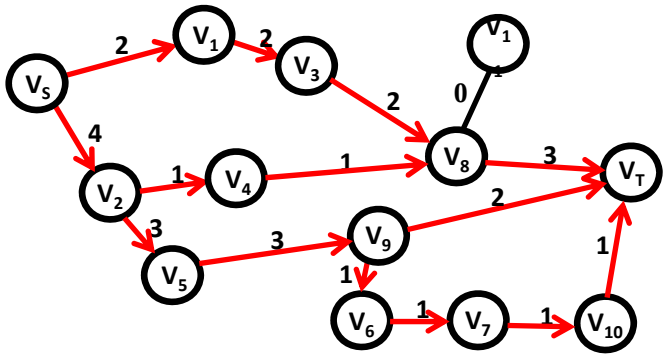
- Adjacent graph, G(V, E) : an undirected edge-weight graph
  - Any V_i, in V represents a partitioned region, R_i.
  - Any $E_{i,j}$, in E represents the adjacent relation between two partitioned regions, R_i and R_j, and the edge weight, $w_{i,j}$, represents the allowable net capacity between R_i and R_j.





Flow-Based Path Generation

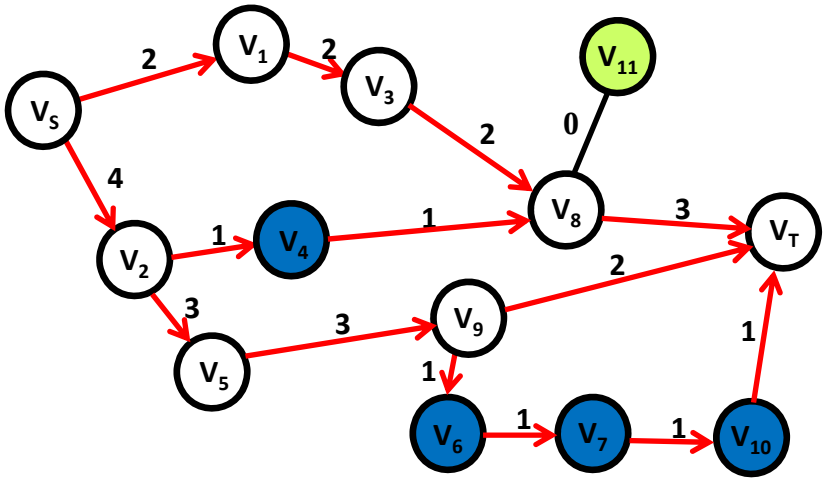
- According to the adjacent graph, with V_S and V_T, the number of *maximum flow(routed path), M,* can be obtained by performing maximum-flow algorithm
  - Un-routable condition
    - The flow, *M*, is smaller than the number of nets, *r*, in a bus
- Routed path assignment
  - M flows, form 1 to M and r nets, from 1 to r, and $M \geq r$
  - Balanced distribution expression



$$f_i = 1 + \left\lfloor \frac{(i-1)(M-1)}{r-1} \right\rfloor, \quad if\ 1 \leq i \leq r$$

Assigned Region Relations

- Three relations between net and its assigned routed region
  1. Private region
     - Only one net is assigned into this region ➔ All routing grids inside this region are only used by the route of the net
  2. Share region
     - Without any net is assigned into the region ➔ All routing grids inside this region can be used by all possible nets
  3. Public region
     - Some nets are assigned into this region ➔ All routing grids inside this region can be used by the routes of these assigned nets



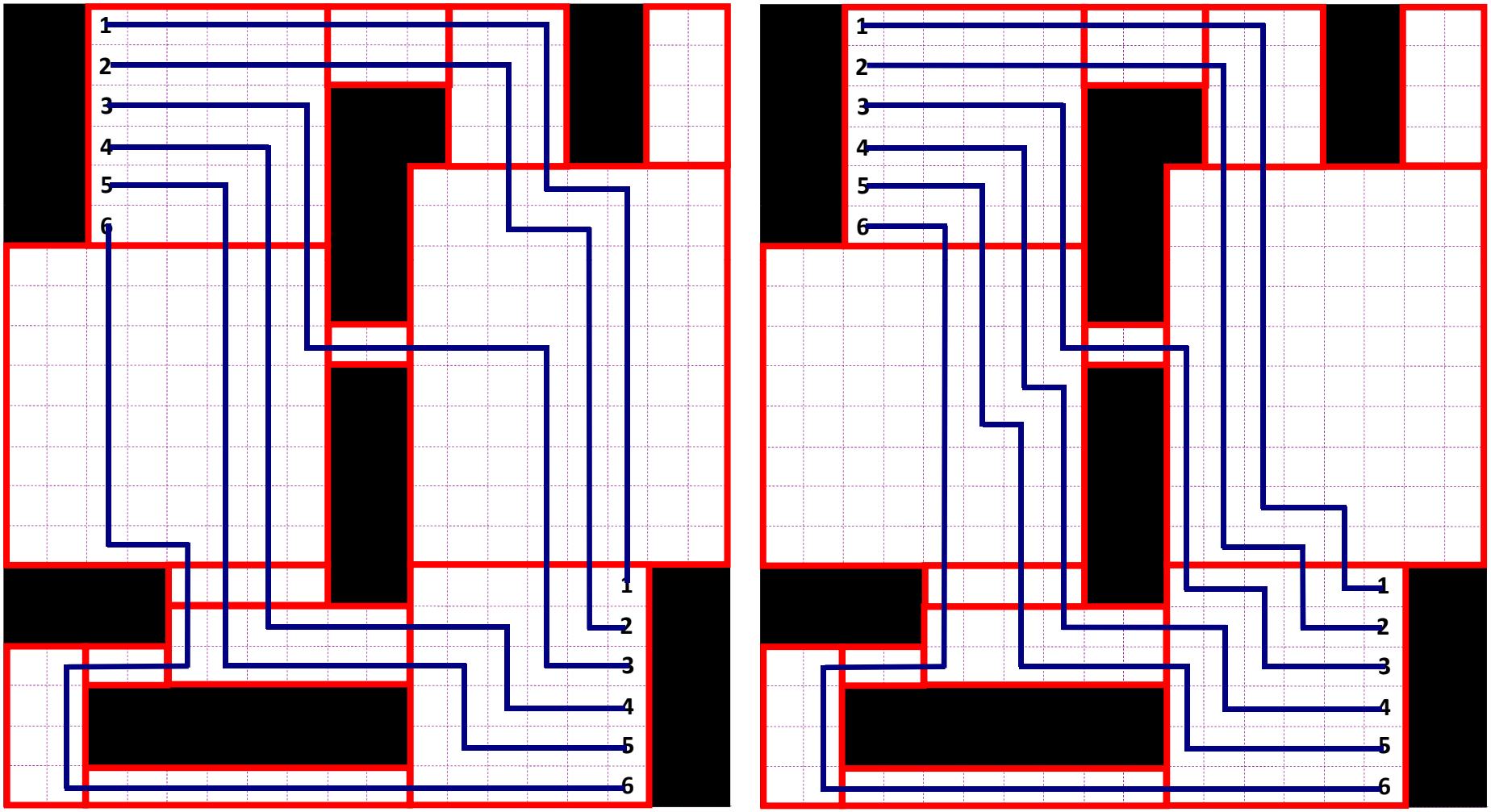
Assign region path :

- Net1, Net2 : $V_S \rightarrow V_1 \rightarrow V_3 \rightarrow V_8 \rightarrow V_T$
- Net3 : $V_S \rightarrow V_2 \rightarrow V_4 \rightarrow V_8 \rightarrow V_T$
- Net4, Net5 : $V_S \rightarrow V_2 \rightarrow V_5 \rightarrow V_9 \rightarrow V_T$
- Net6 : $V_S \rightarrow V_2 \rightarrow V_5 \rightarrow V_9 \rightarrow V_6 \rightarrow V_7 \rightarrow V_{10} \rightarrow V_T$

Initial Obstacle-Aware Path Generation

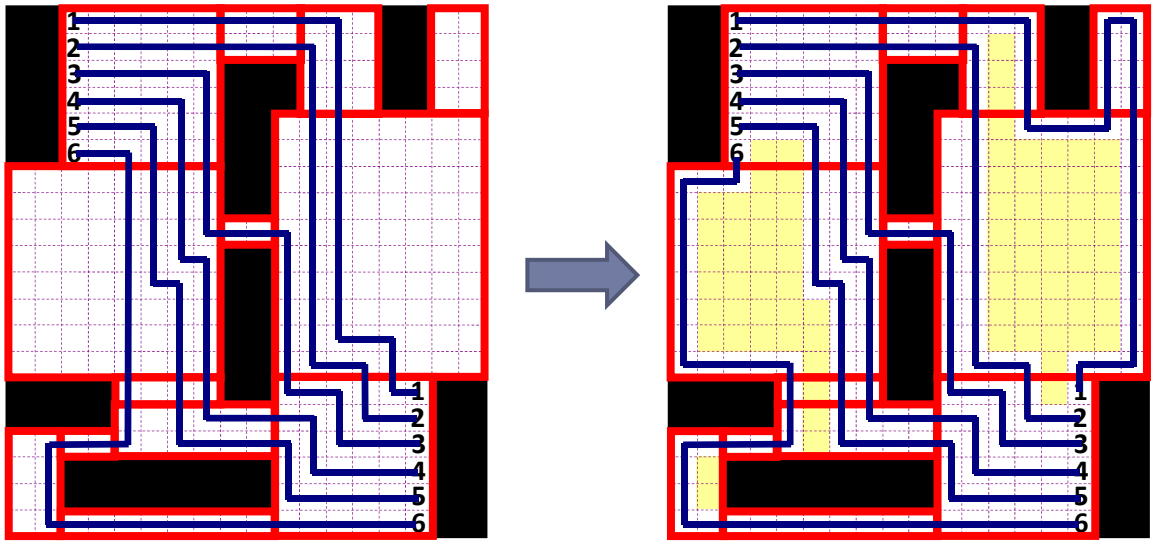
- Objective: determine the routing path inside the assigned region for all nets
- Two-phase shortest path generation
  1. Diffusion-based shortest path: sequentially generate the shortest path for all nets
     - First and last of un-routed nets are routed firstly
     - Route as the outer routing grids as possible
     - The routed nets are regarded as obstacles
  2. Convergence-based shortest path: integrate the available routing grids together and distribute in the outer routing area
     - Routing with reverse routing order in the first phase
     - Reassign the routing grids as convergent as possible

Example of Initial Path Generation



Iterative Length-Matching Path Generation

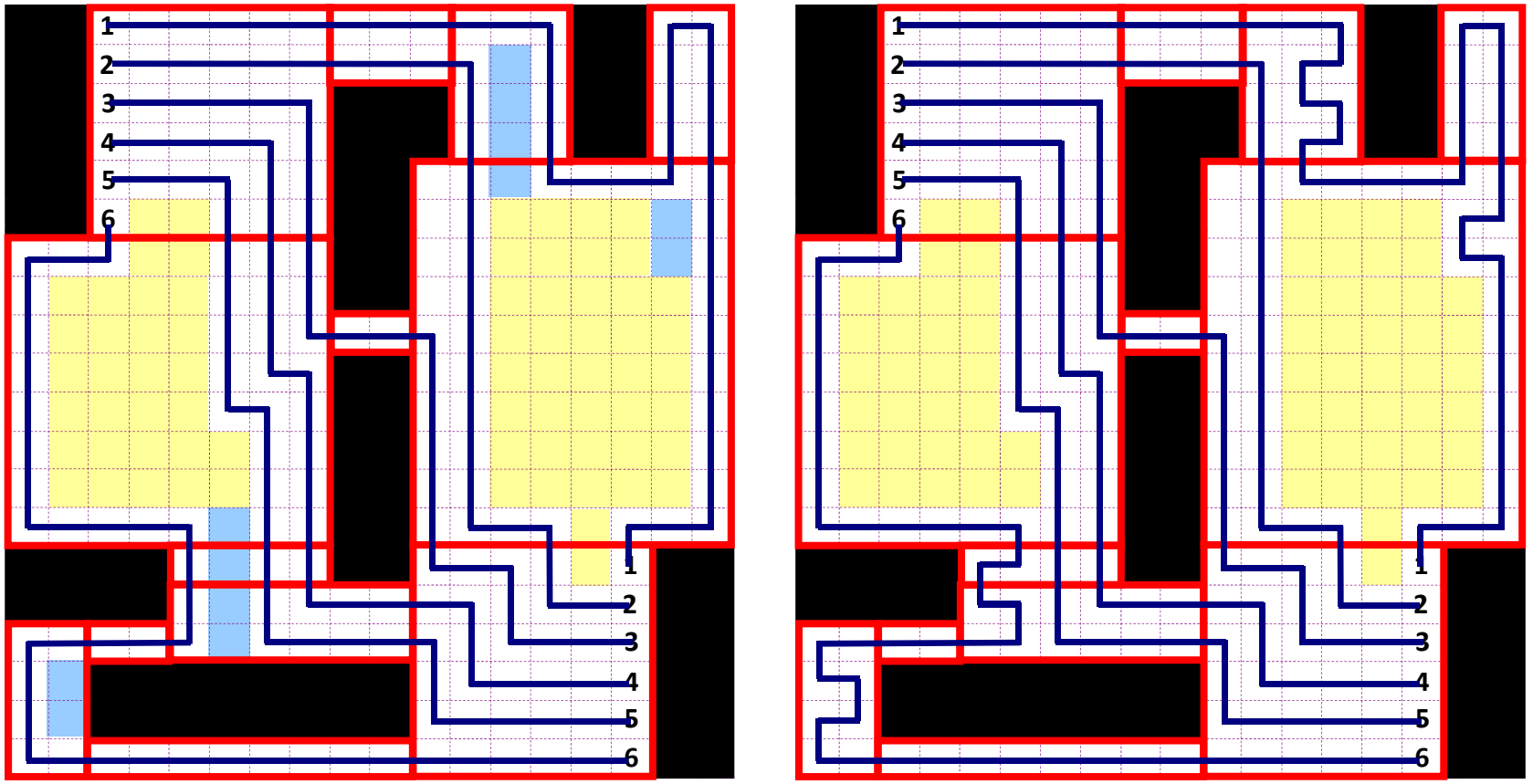
- Three sub-processes for length-matching path generation
  1. Diffusion-based longest path generation
  2. Internal length-matching path modification
  3. External length-matching path modification
- Diffusion-based longest path generation
  - The first and the last nets are firstly considered to generate their diffusion-based longest paths.
  - A path along the outer boundary and adjacent obstacles.
  - Go through the regions on **its region** path and available **share regions**.



Internal Length-Matching Path Modification

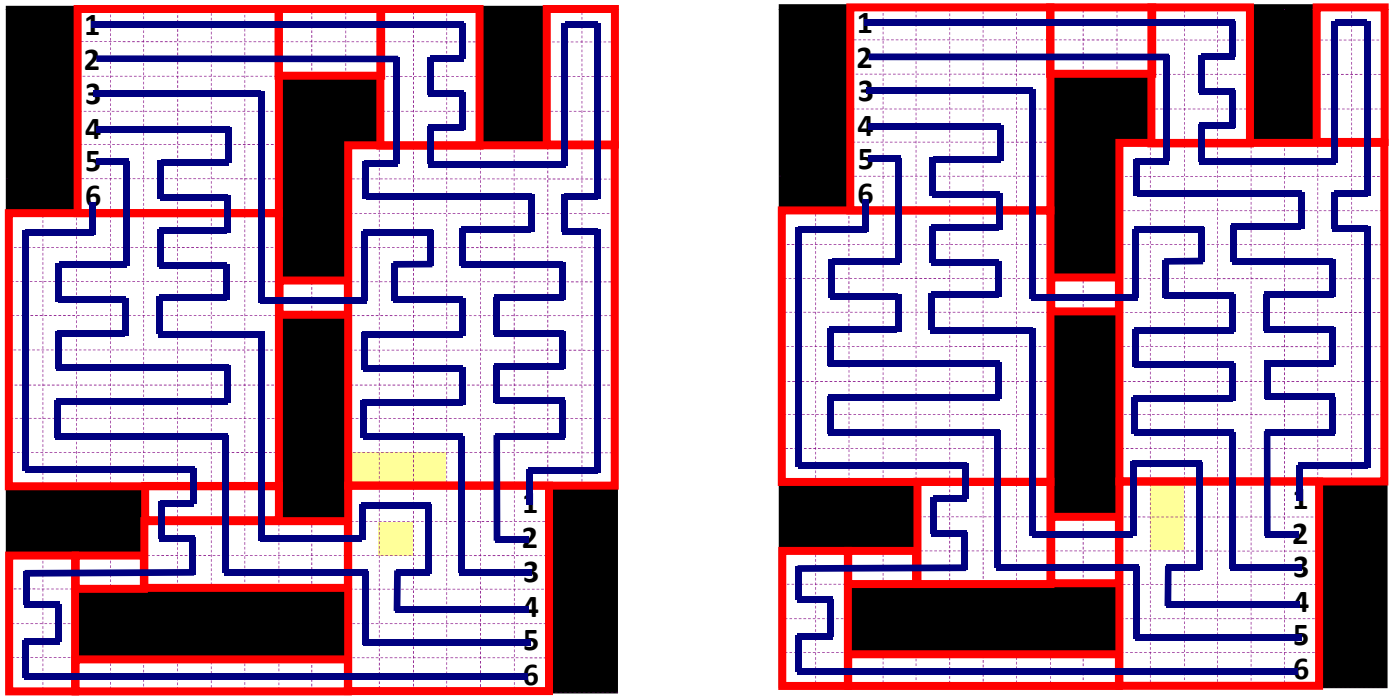
- Objective: modify the diffusion-based longest path result to satisfy the given length constraint
- Three possible conditions
  1. Equal to length constraint
     - Two assigned length-matching pats will be treated as obstacles
     - Process the next pair of un-routed nets
  2. Larger than length constraint
     - Shorten the partial or full paths inside the share regions
  3. Smaller than length constraint
     1. Detour the partial or full paths inside the regions
        - Private region > Share region > Public region
     2. Detour operations: R-flip / C-flip

Example of Internal Path Modification



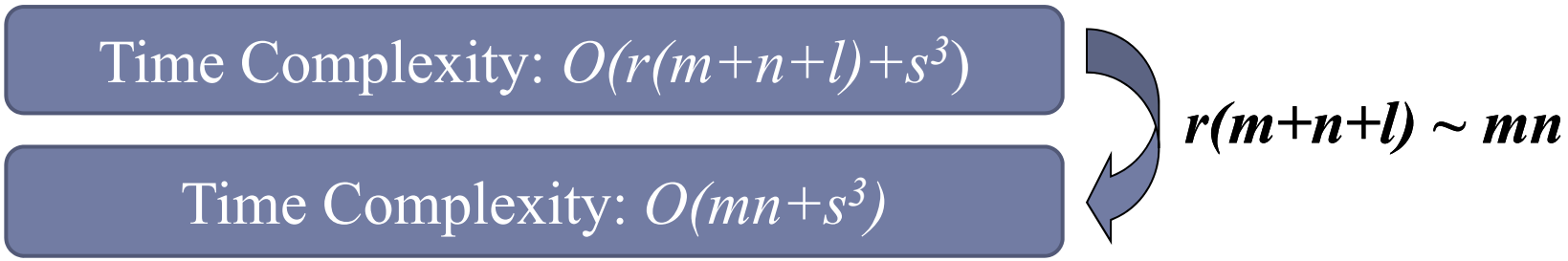
External Length-Matching Path Modification

- Objective: modify the detour path without satisfying the given length constraint after internal pat modification
  - *R-flip* and *C-flip* operations on the other public region to satisfy the given length constraint



Analysis of Time Complexity

- Obstacle-aware region partition inside routing grids
  - Partitioning obstacle-aware regions: $O(s^2)$
  - Adjacent graph construction: $O(s)$
- Obstacle-aware shortest-path generation
  - Finding the maximum flow in adjacent graph: $O(s^3)$
  - Feasible region path assignment for all the nets: $O(rs)$
  - Generating diffusion-based and convergence-based shortest path: $O(mn)$
- Iterative length-matching path generation
  - Generating two diffusion-based longest path in each iteration: $O(m+n)$
  - Internal and external length-matching path modification: $O(l)$
    - l: the given maximum length constrain



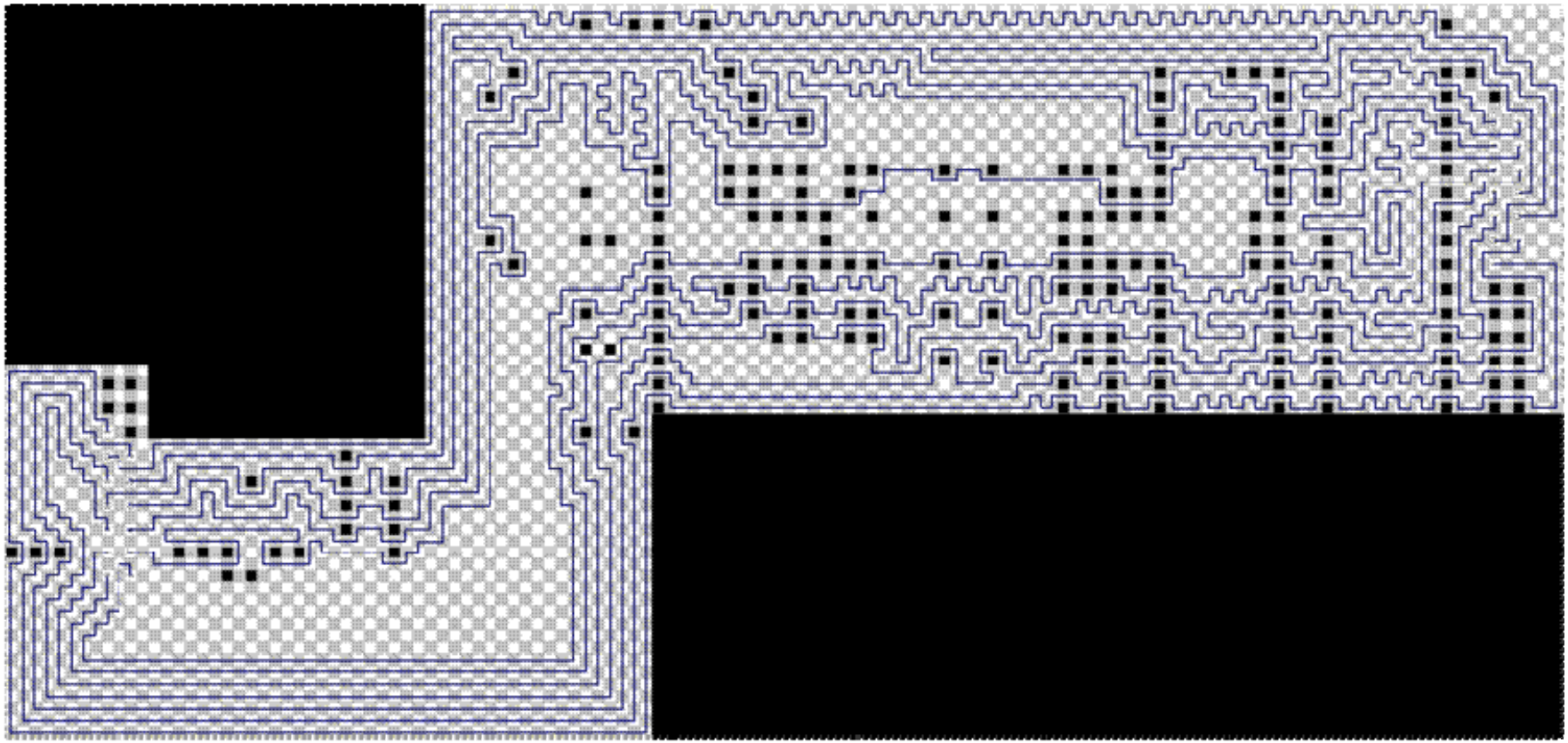
Experimental Results

- Experimental environment
  - PC: Intel Quad Core 2.66GHz with 2GB memory
  - Programming language : C++ language
- Two test examples are from the previous published paper -- CAFÉ router
- The information of two test examples and the results of comparison show as following :

Circuits	***Area***	***#Obstacle Grids***	***Number of Nets***	***Length Constraint***	
Data1	***28x28***	***13***	***4***	***100***	***150***
Data2	***61x130***	***3441***	***13***	***251***	***301***

Example	Length constraint	CAFÉ Router		Our Routing	
		Length Error	CPU Time(s)	Length Error	CPU Time(s)
Data1	100	0	0.35(100%)	0	0.08(22.9%)
	150	10	0.42(100%)	0	0.09(21.4%)
Data2	251	10	25.63(100%)	0	4.96(19.4%)
	301	6	26.85(100%)	0	5.28(19.7%)
Total		26	53.25(19.5%)	0	10.41(19.5%)

Data2 Results



Length constraint = 251

Conclusions

- Signal propagation delays on PCBs are requested to meet the timing constraints with very high accuracy
- Based on obstacle-aware region partition, obstacle-aware shortest path generation and two detouring operations, an efficient approach is proposed to generate the length-matching paths

Thanks for your attention!!

Q&A