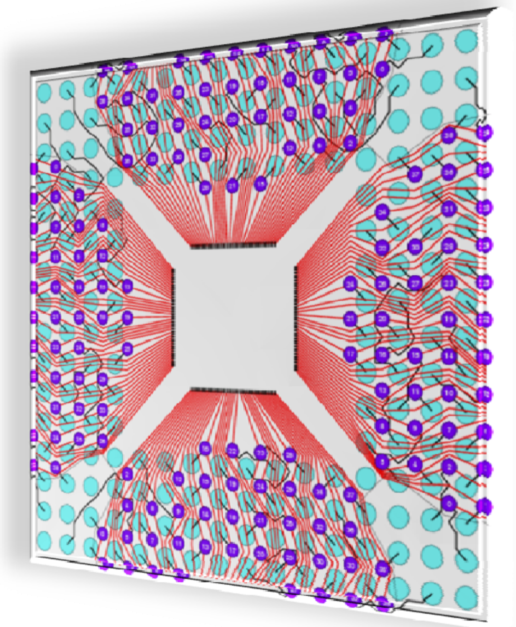


Routability Driven Modification Method of Monotonic Via Assignment for 2-layer Ball Grid Array Packages

Yoichi TOMIOKA , Atsushi TAKAHASHI

Department of Communications and Integrated Systems, Tokyo Institute of Technology, Japan

BACKGROUND

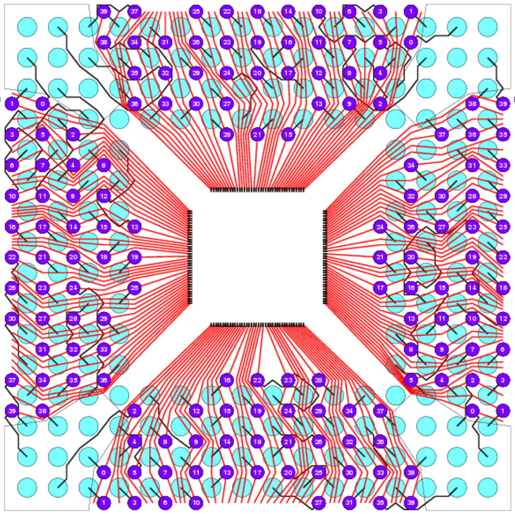
BGA Packages

- A number of connections
- High-density
- Consuming much time in manual routing → Automation

Routing Problem for BGA packages

Difficulty
- A number of nets in narrow area
- Tight Requirements

Characteristics
- Simple structure
- Near-monotonic netlist



Routing Method for 2-layer BGA
- Via placement
- Global routing design

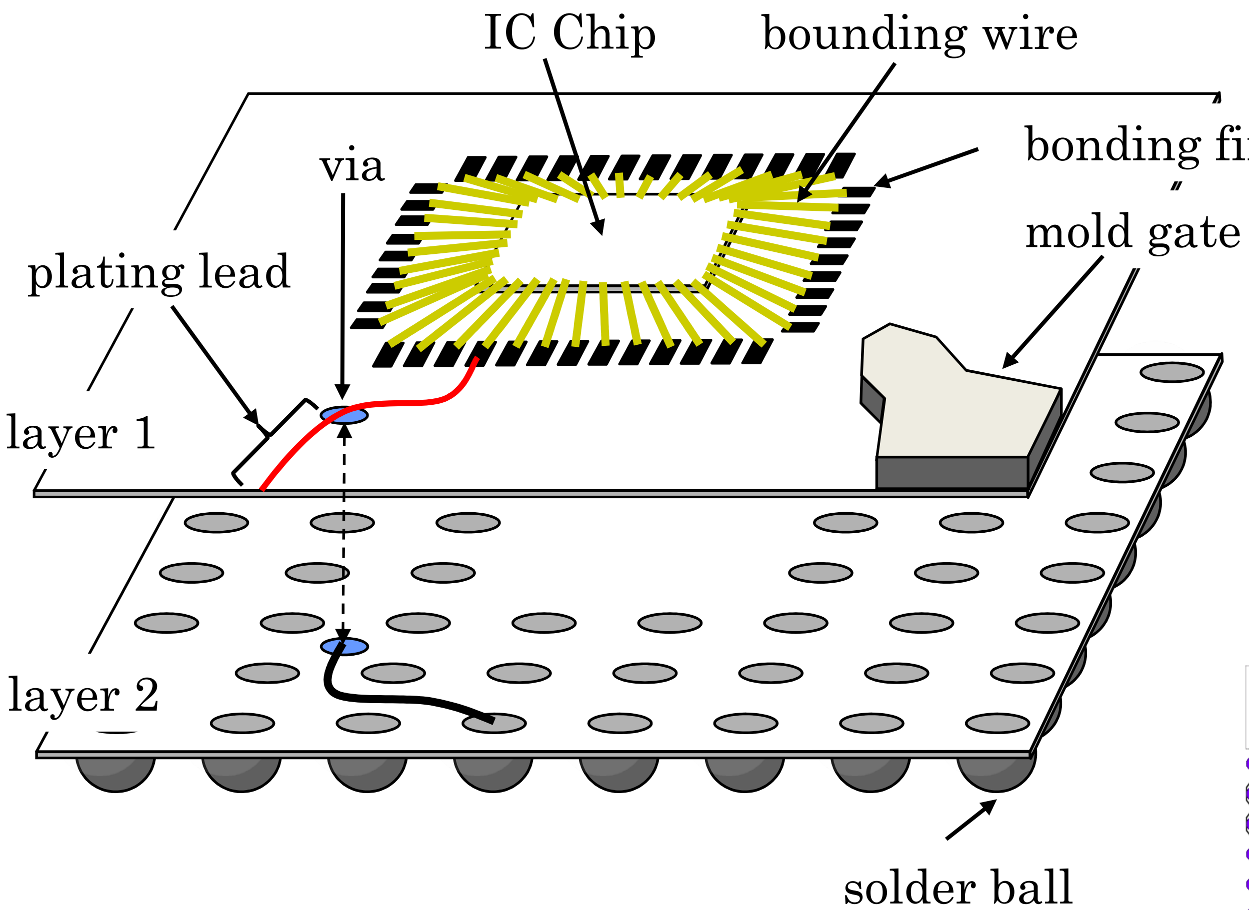
Previous works

- Single-layer
  - Monotonic pin assignment [ICCAD95, Yu *et al.*] [97, Shibata *et al.*]
  - Free-assignment flip chip routing [ICCAD05, Fang *et al.*]
  - Pre-assignment flip chip routing [DAC07, Fang *et al.*]
- Multi-layer (Give no via placements)
  - Layer assignment and routing for PGA [TCAD98, Tsai *et al.*]
  - Layer assignment and routing for BGA [ICCAD99, Chen *et al.*]
- **Two-layer**
  - Iterative via placement modification [TCAD06, Kubo *et al.*]
    Generate via placement and global routes on layer 1
    No global routes on layer 2

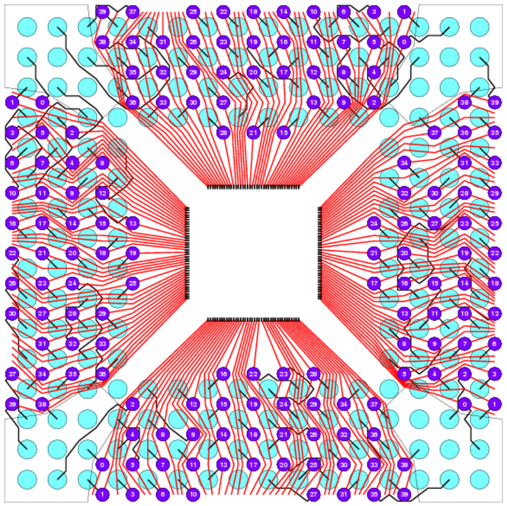
Our method proposed here (enhancement of Kubo's method)

- Improve computational complexity
- Generate global routes on layer 2

A Model of 2-layer BGA Package



Routing Design

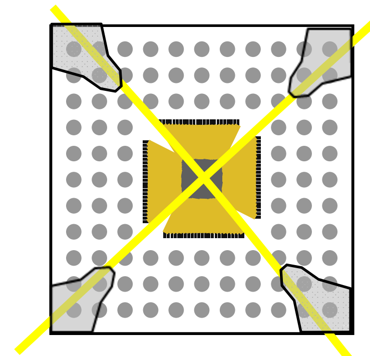


Connection requirements are given

- Wire connects a finger and a ball
- Plating leads are used for electrical plating

VIA GRID ARRAY

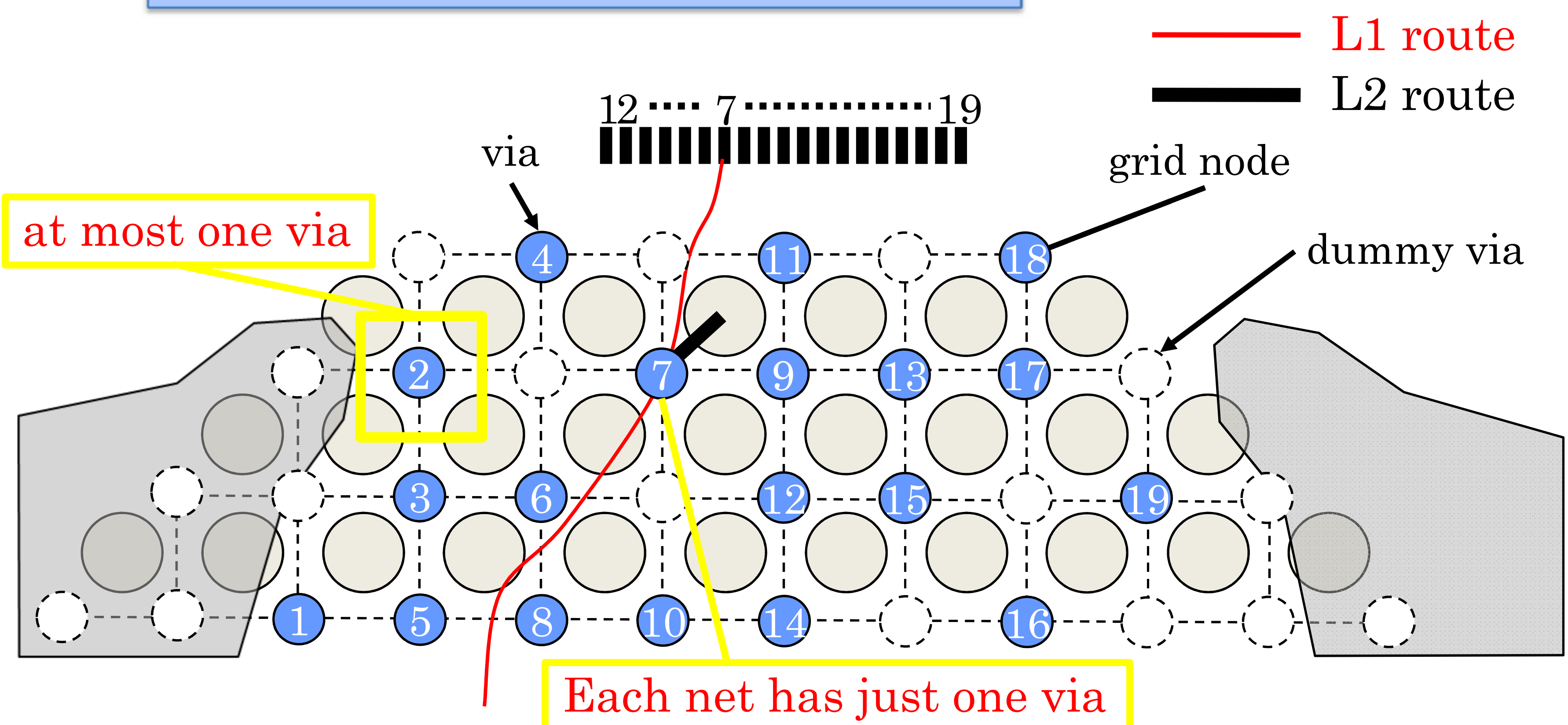
- Divide a package into four sectors



Via placement

⇕

Via assignment Φ : vias → grid nodes

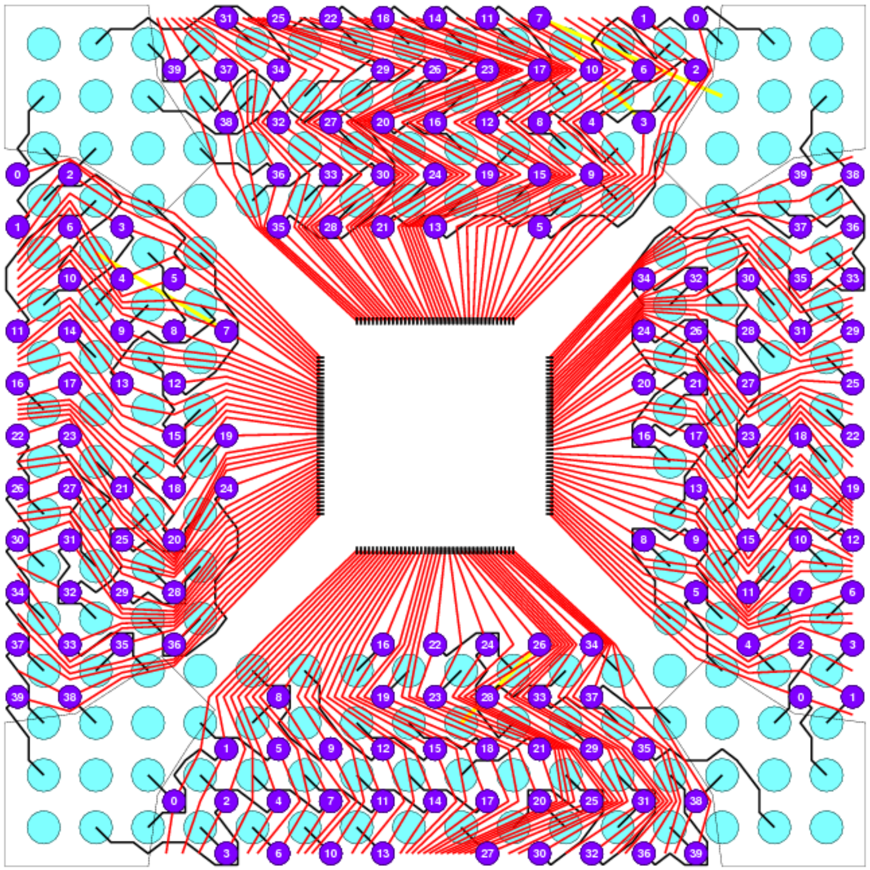


Via Assignment

Global routing depends on via assignment

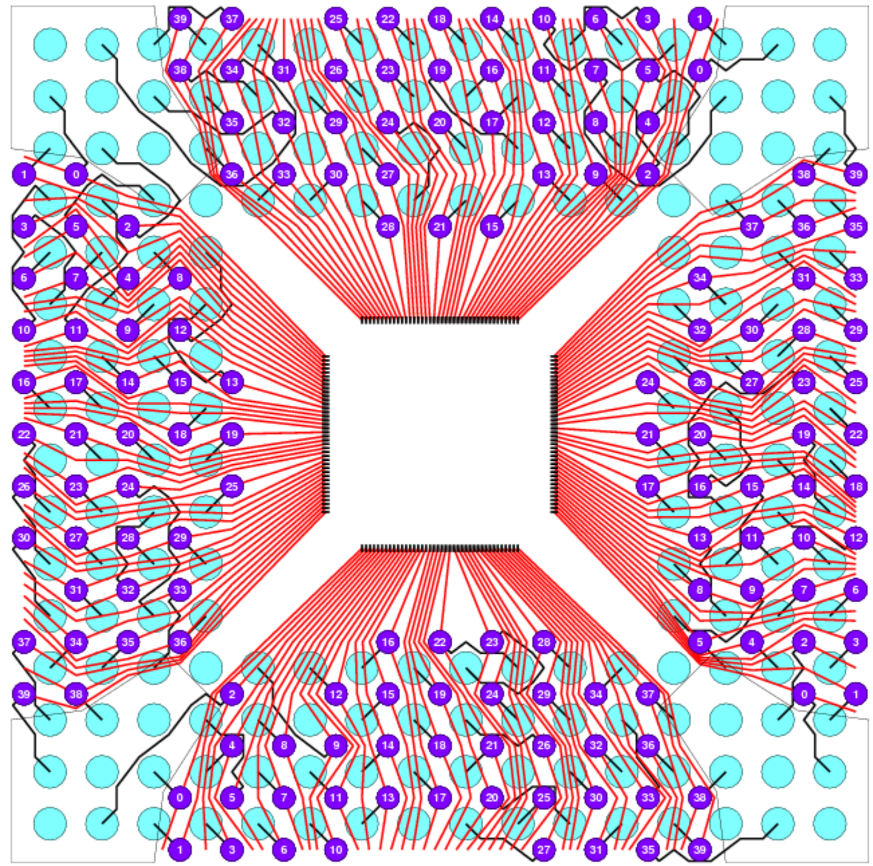
Bad via assignment

Low routability



Good via assignment

High routability



Propose improved via assignment method

Monotonic Via Assignment

Classification of Routes [ICCAD95, Yu *et al.*]

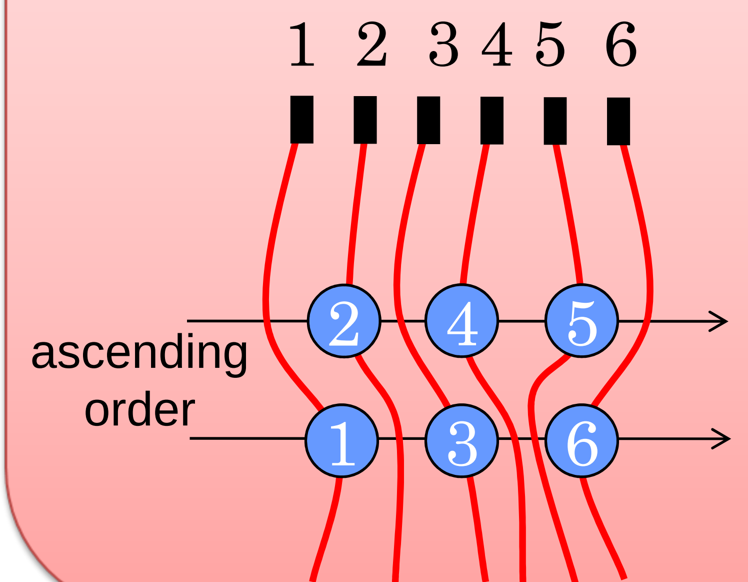
monotonic : if any horizontal line intersects at most once

non-monotonic : otherwise

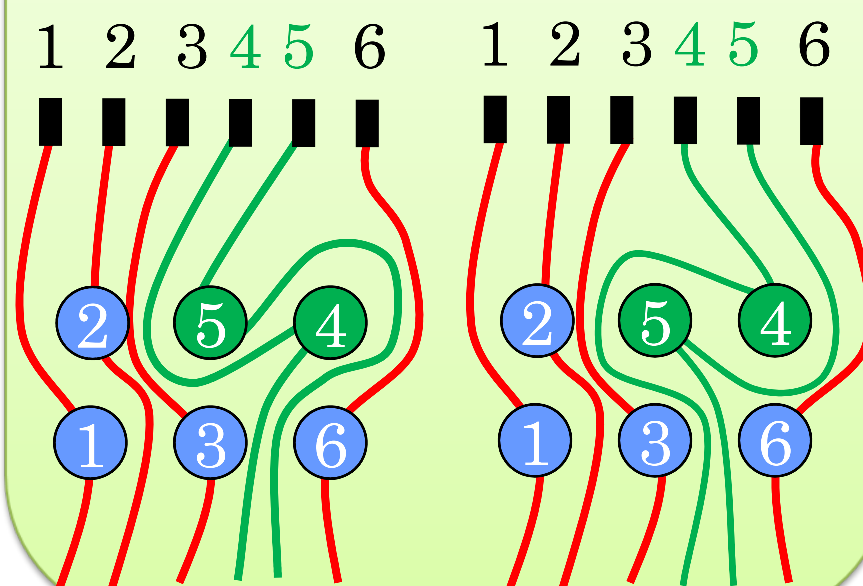
monotonic via assignment

All routes on layer 1 can be monotonic

Monotonic routing pattern is unique



non-monotonic via assignment



Objectives

Generate **monotonic via assignment** and global routing

Layer 1

Monotonic routing patterns

NG OK

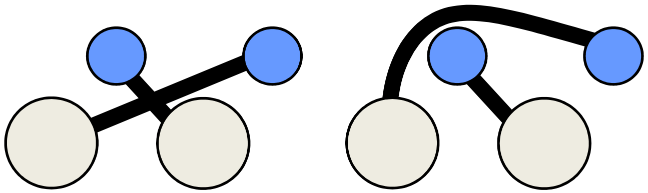
(maximum wire congestion) $\leq$ (allowable congestion)

Layer 2

Routing pattern generated on the routing graph

Connect all vias and balls without intersections

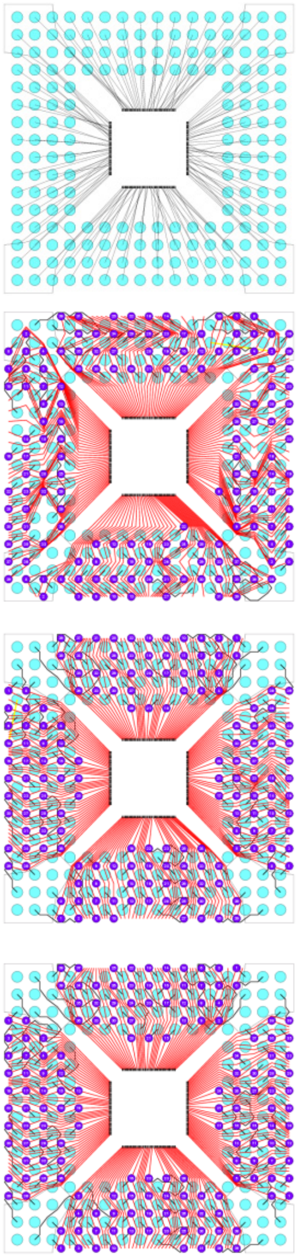
NG OK



Outline of Our Method

[TCAD06, Kubo *et al.*]

Our contribution



Netlist

Initial via assignment

Place vias near their balls under monotonic condition

Iterative via modification

Phase 1

Improve layer 1, Keep layer 2

Improve computation complexity

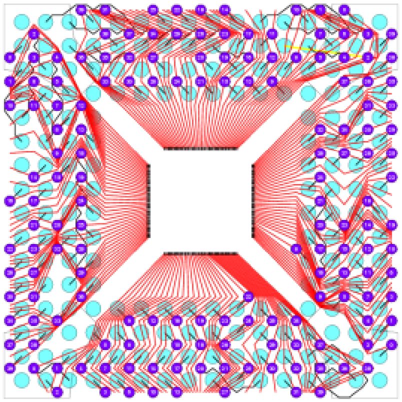
Phase 2

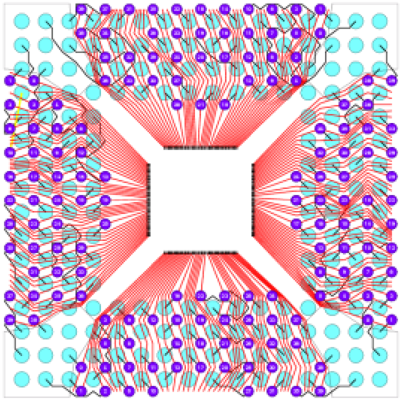
Improve layer 2, Keep layer 1

Introduce a new modification

Via assignment
Global Routing

Iterative modification of Phase 1





Objectives:

Layer 1 :

Improve { total wire length, max wire congestion }

Layer 2 : (no global routing)

Keep via-ball distance small

By modification: EXC, ROT, MSEQ

Routes change

- Layer 1: drastic
- Layer 2: small

In each iteration, apply max gain modification

- Max gain computation $O(N^2) \longrightarrow O(N)$
- Handling mold gates

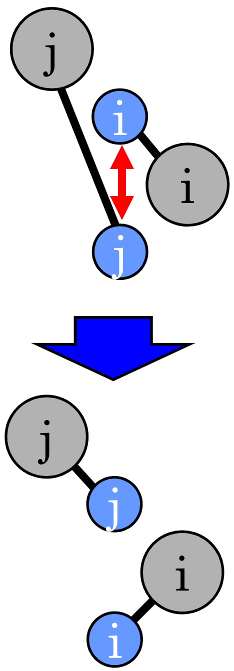
N : #(grid nodes)

MODIFICATIONS

[IEICE05, Kubo *et al.*]

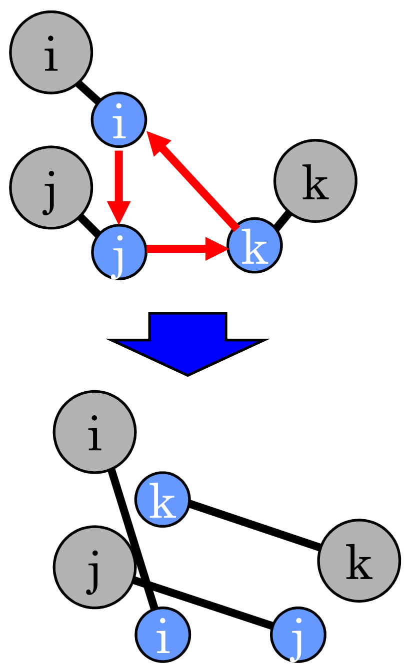
Exchange (EXC)

Vertically adjacent vias



Rotation (ROT)

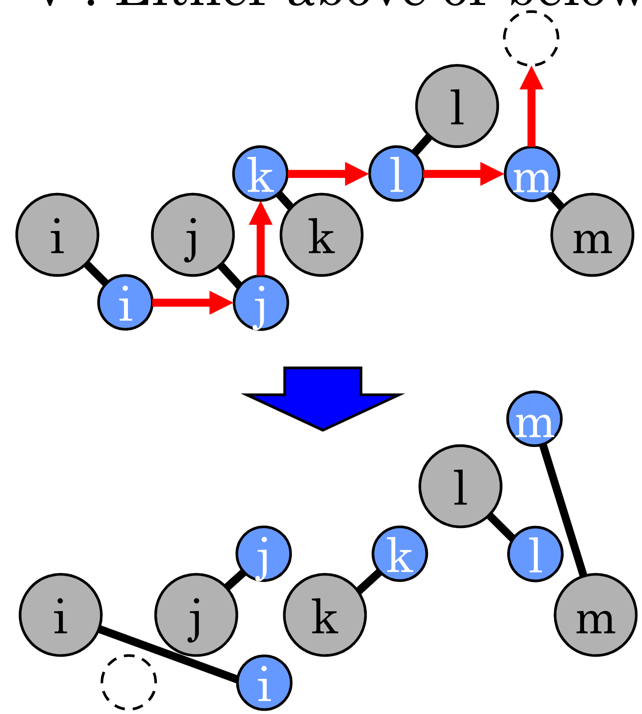
Three vias on a unit square



Monotonic Sequence (MSEQ)

H : Either left or right

V : Either above or below



	EXC	ROT	MSEQ
#patterns	$O(N)$	$O(N)$	exponential
max. gain	$O(N)$	$O(N)$	$O(N^2) \rightarrow O(N)$

N : #(grid nodes)

Cost Function of Phase 1

$$COST_1(\Phi) = \sum \begin{cases} \text{Wire congestion balance on L1} & F \\ \text{Wire length on L1} & C \\ \text{Wire length on L2} & D \\ \text{Penalty for mold gates} & OBS \end{cases}$$

$= \sum$ (local costs of vias)

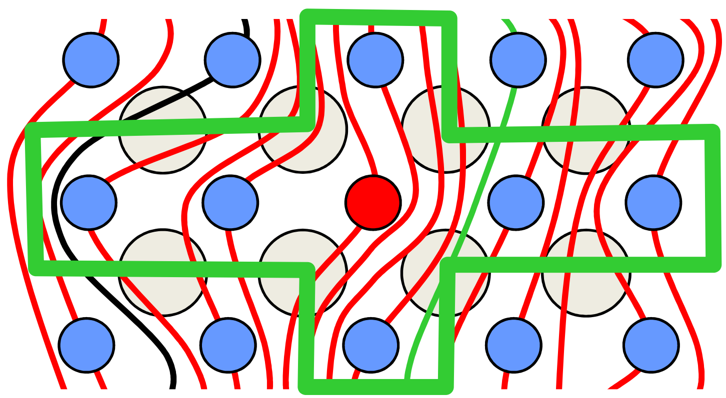
↑ nets in the region

Gain of each modification

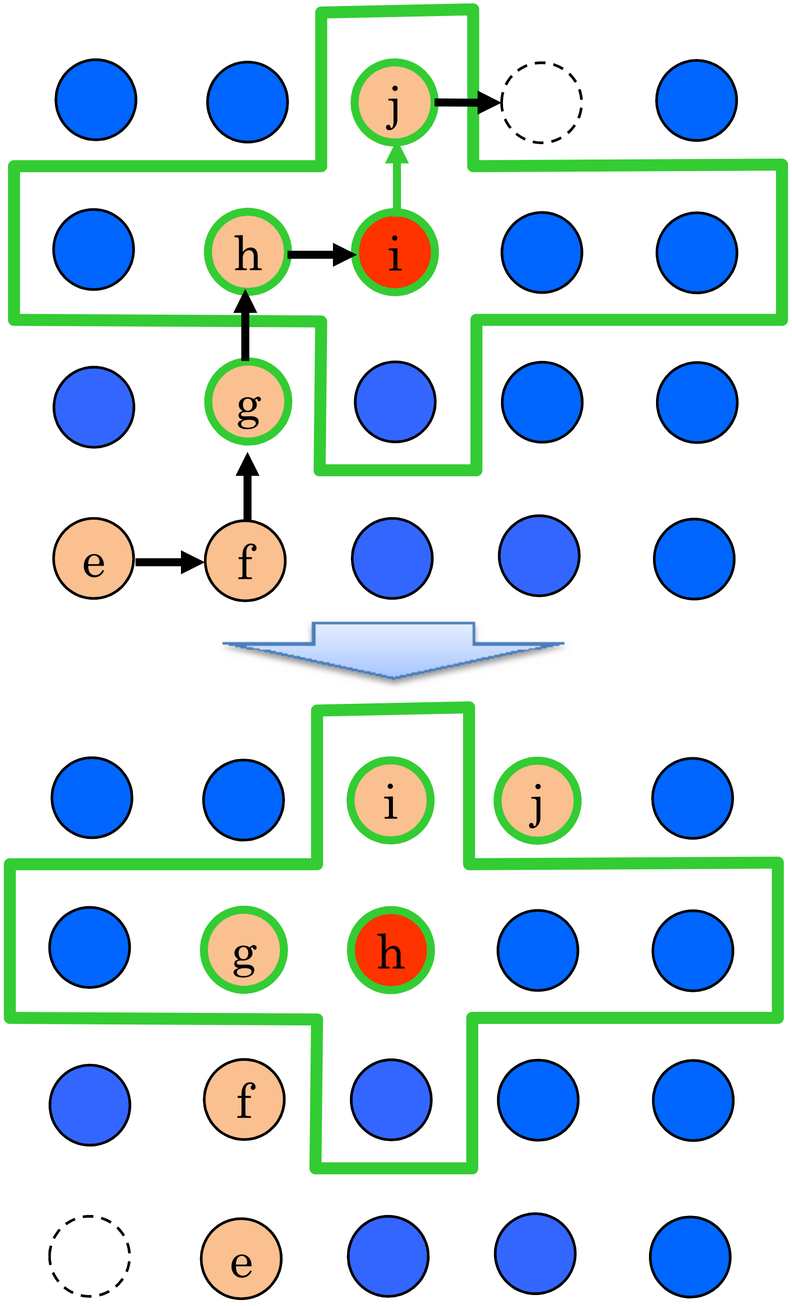
= Cost before mod. − Cost after mod.

$= \sum$ (local gains of grid nodes)

↑ nets in the region



MSEQ AND LOCAL GAIN

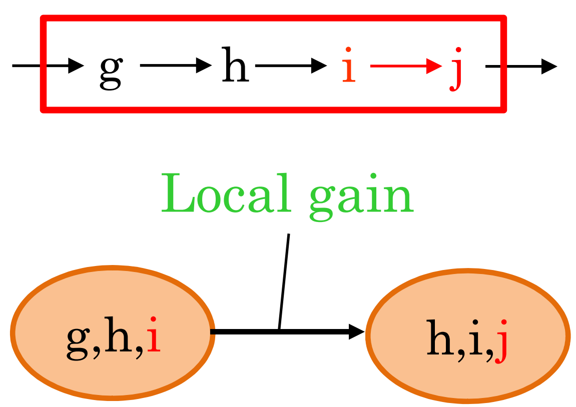


Focus on above-right MSEQs

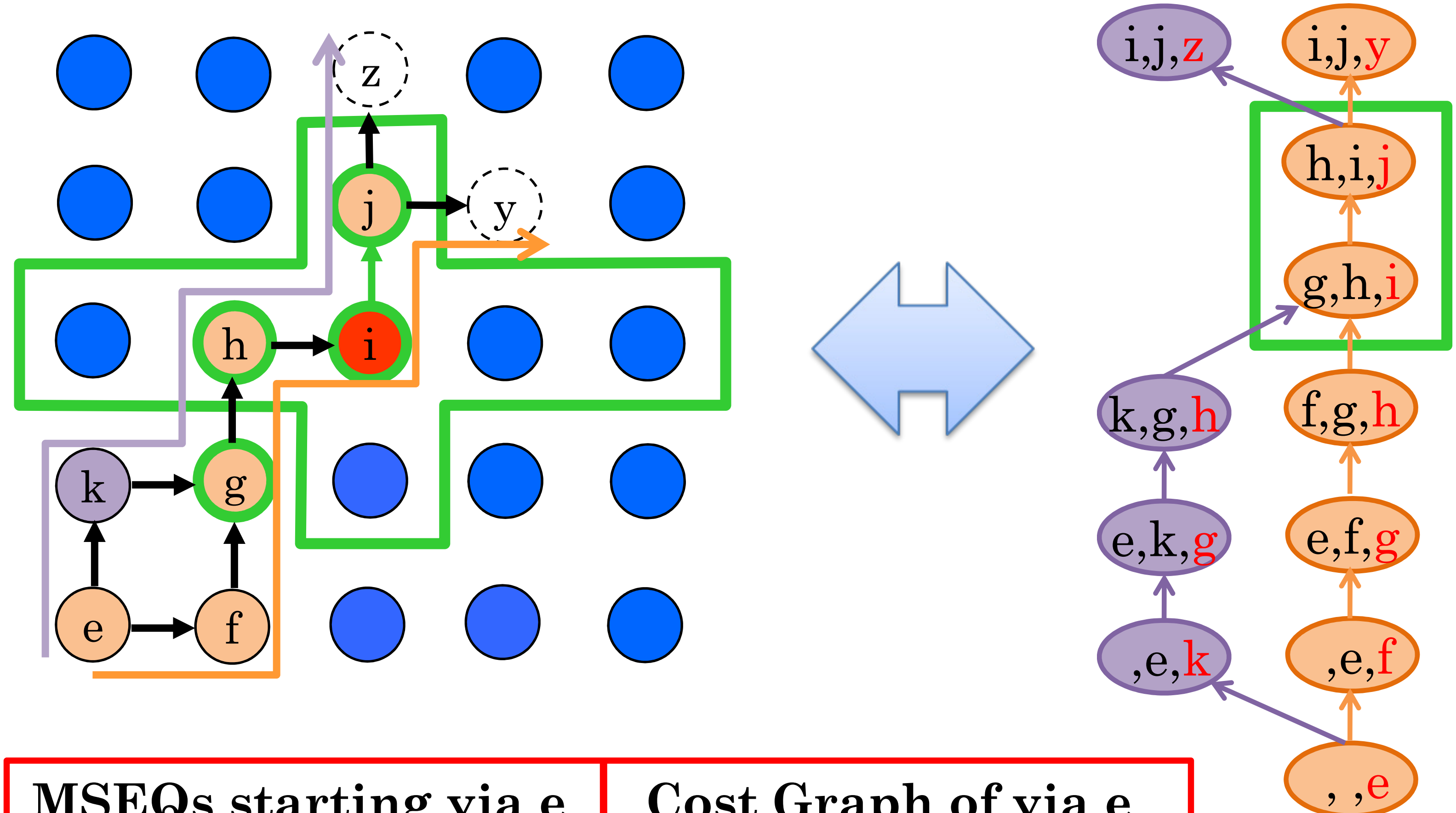
Four vias (g,h,i,j) in MSEQ

Nets in the region after applying the MSEQ are specified

Local gain in the region



COST GRAPH OF EACH VIA (KUBO'S COST GRAPH)



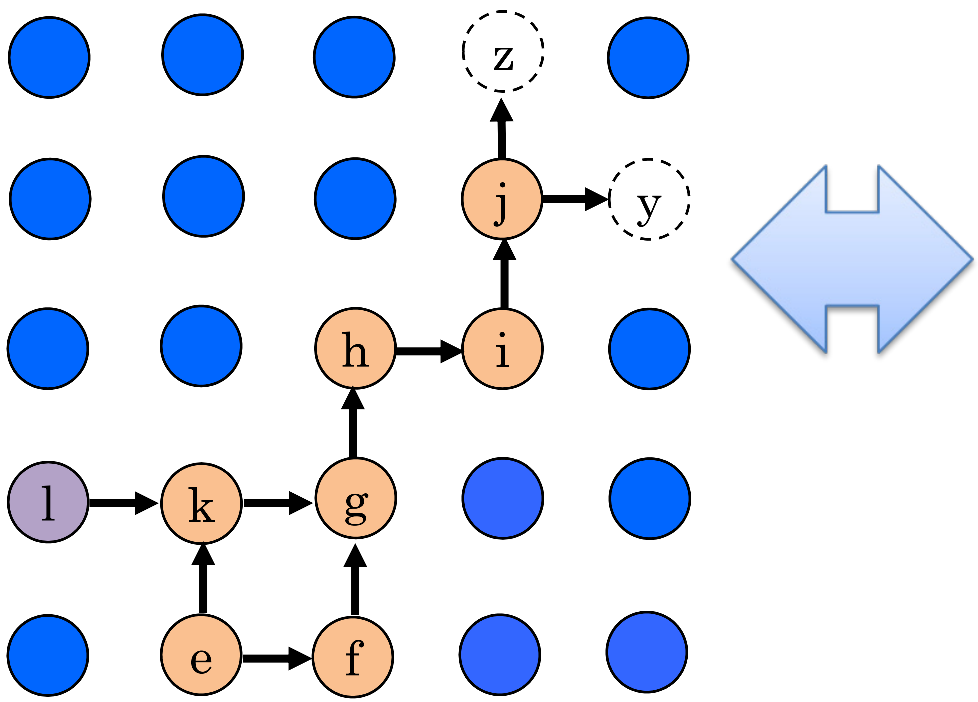
MSEQs starting via e	Cost Graph of via e
#MSEQs is exponential	|V|, |E| : $O(N)$
Max. gain MSEQ $O(N)$	Longest path $O(\|V\|+\|E\|)$

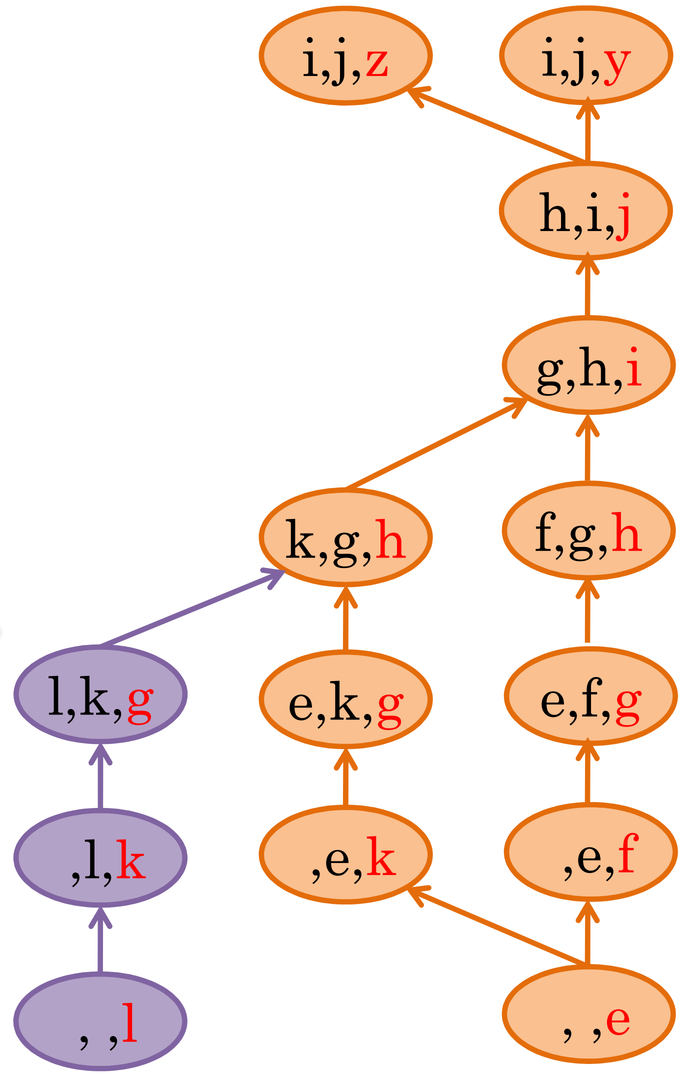
N : #(grid nodes)

Max. gain MSEQ : $O(N^2)$

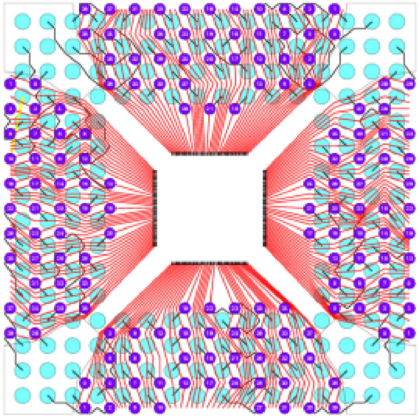
Our Cost Graph

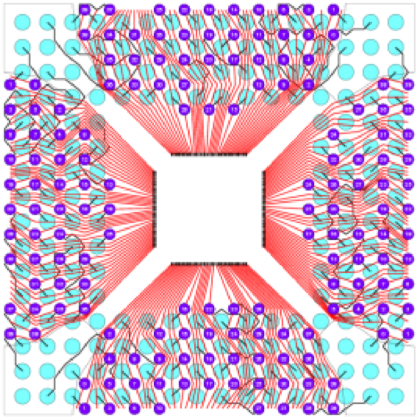
- Combine cost graphs of different vias
- |V| and |E| are $O(N)$
- Max. gain is obtained in $O(N)$





ITERATIVE MODIFICATION OF PHASE 2





Objectives:

Layer 2 : (global routing)

Improve completion ratio of nets

Layer 1:

Keep { total wire length / max wire congestion } small

By modifications : CEXC

Routes change

Layer 1 : small

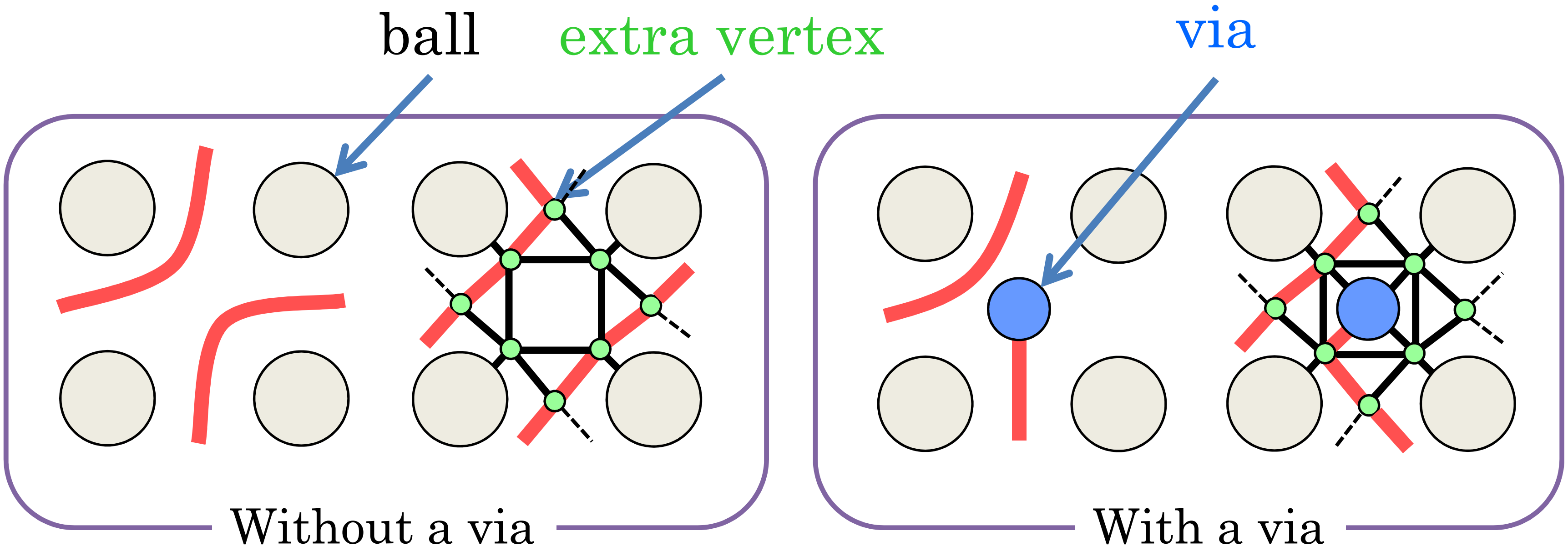
Layer 2 : drastic

In each iteration, apply max gain modification

- Generate global routing on layer 2
- Introduce a new modification

ROUTING GRAPH

- The routing graph for Via Assignment Φ



  - Correspond to the routing resource on layer 2

- Generate routes by the rip-up and reroute technique

Cost Function of Phase 2

$$COST_2(\Phi) = \sum \begin{cases} \text{Wire congestion violations on L1} & \Delta \\ \text{Wire length on L2} & L \\ \text{Unconnected nets on L2} & U \end{cases}$$

Phase 1

- Wire congestion balance on L1
- Wire length on L1
- Wire length on L2

Manhattan distance

- Penalty for mold gates

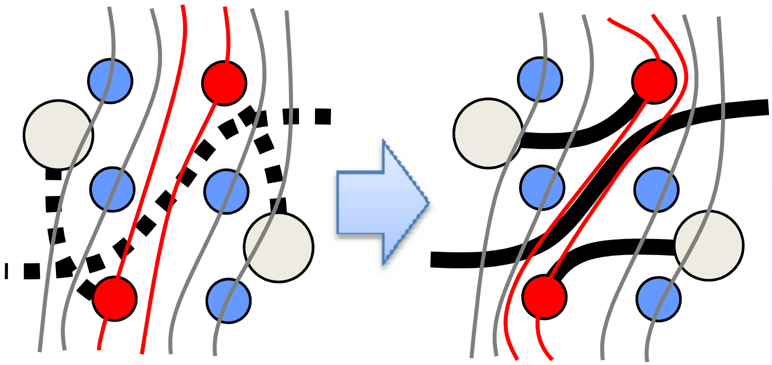
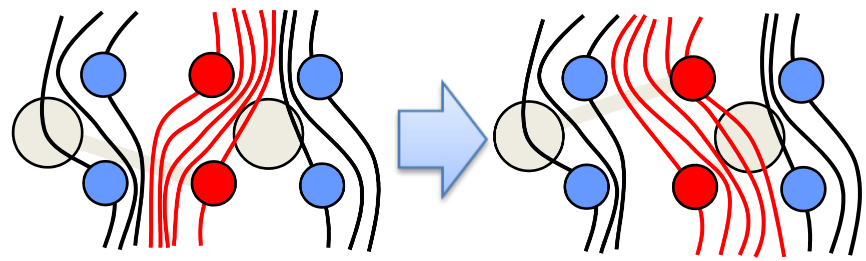
Phase 2

- Wire congestion violations on L1
- Wire length on L2
- Unconnected nets on L2

Generate routes on routing graph

CEXC (Coast Exchange)

- Exchange two vias

	CEXC	EXC
		
	Difference of labels is small	Vertically adjacent
L1	little	drastic
L2	drastic	little

EXPERIMENTS

- Implement our proposed method
  - C++ language
  - Intel Pentium 4, 3.4GHz
  - 1GB memory

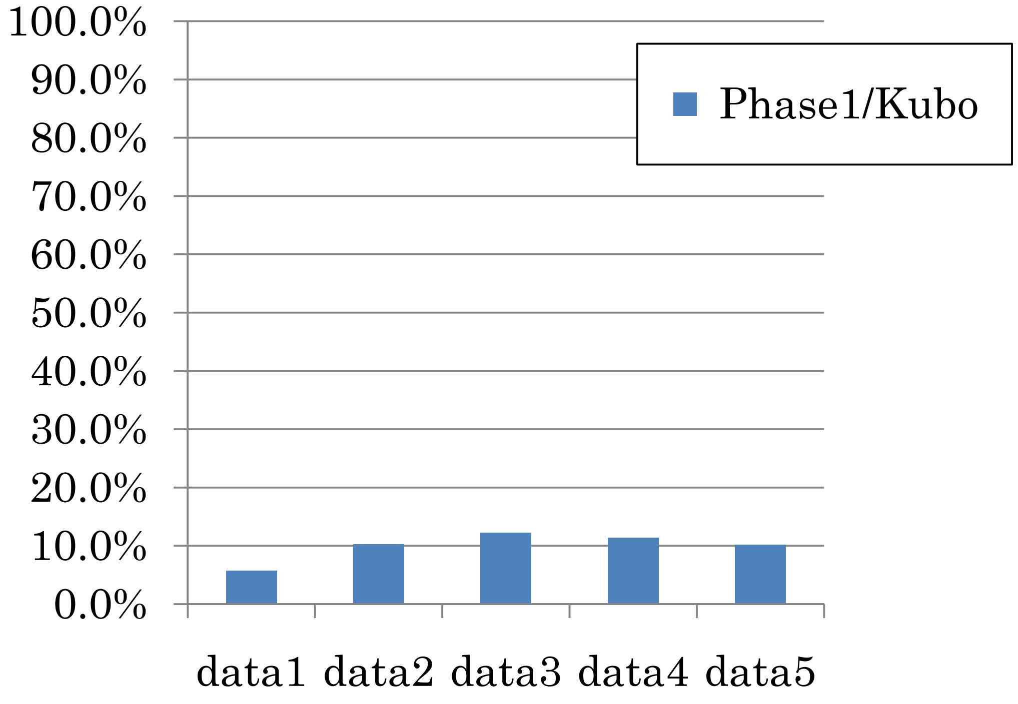
Inputs similar to practical cases are artificially generated

- Compare the execution time (Kubo's cost graph vs. Ours)
- Effect of CEXCs

Results of Phase 1

10 times faster than Kubo's method

	#net	Kubo [sec]	Phase1 [sec]
data1	316	44.96	2.58
data2	192	8.37	0.86
data3	160	10.58	1.29
data4	160	9.57	1.09
data5	160	8.33	0.85



Results of Phase 2

Improve completion ratio drastically

Connect all nets in four data

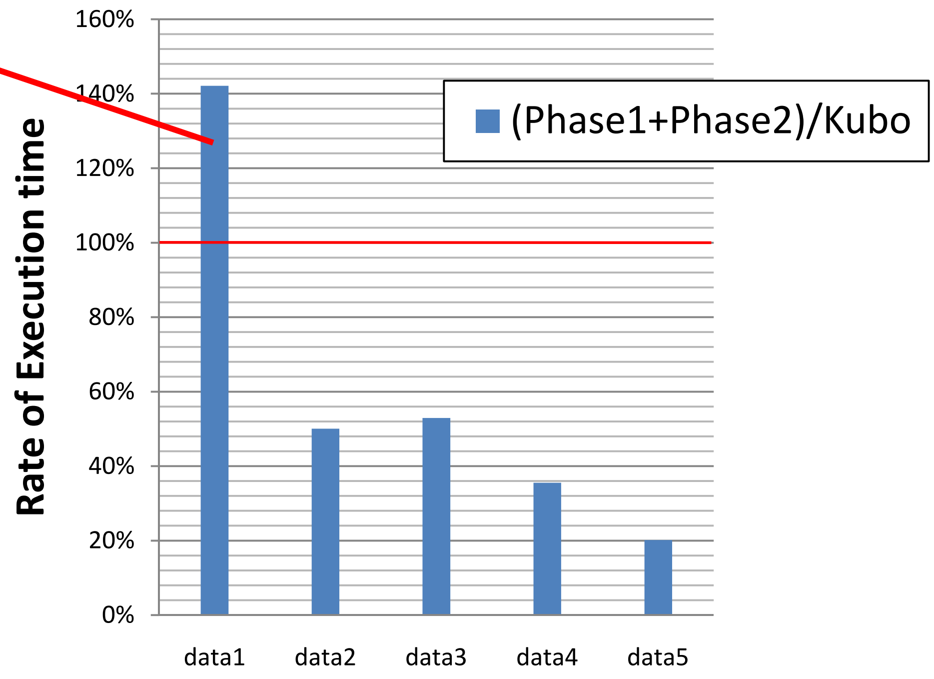
	#net	U			Δ			Phase 2 [sec]
data1	316	11	→	5	2.8	→	2.8	61.3
data2	192	3	→	0	47.6	→	40.0	3.3
data3	160	5	→	0	7.3	→	5.3	4.3
data4	160	3	→	0	11.5	→	11.5	2.3
data5	160	0	→	0	7.1	→	2.8	0.8

$\Delta = 1$

≈

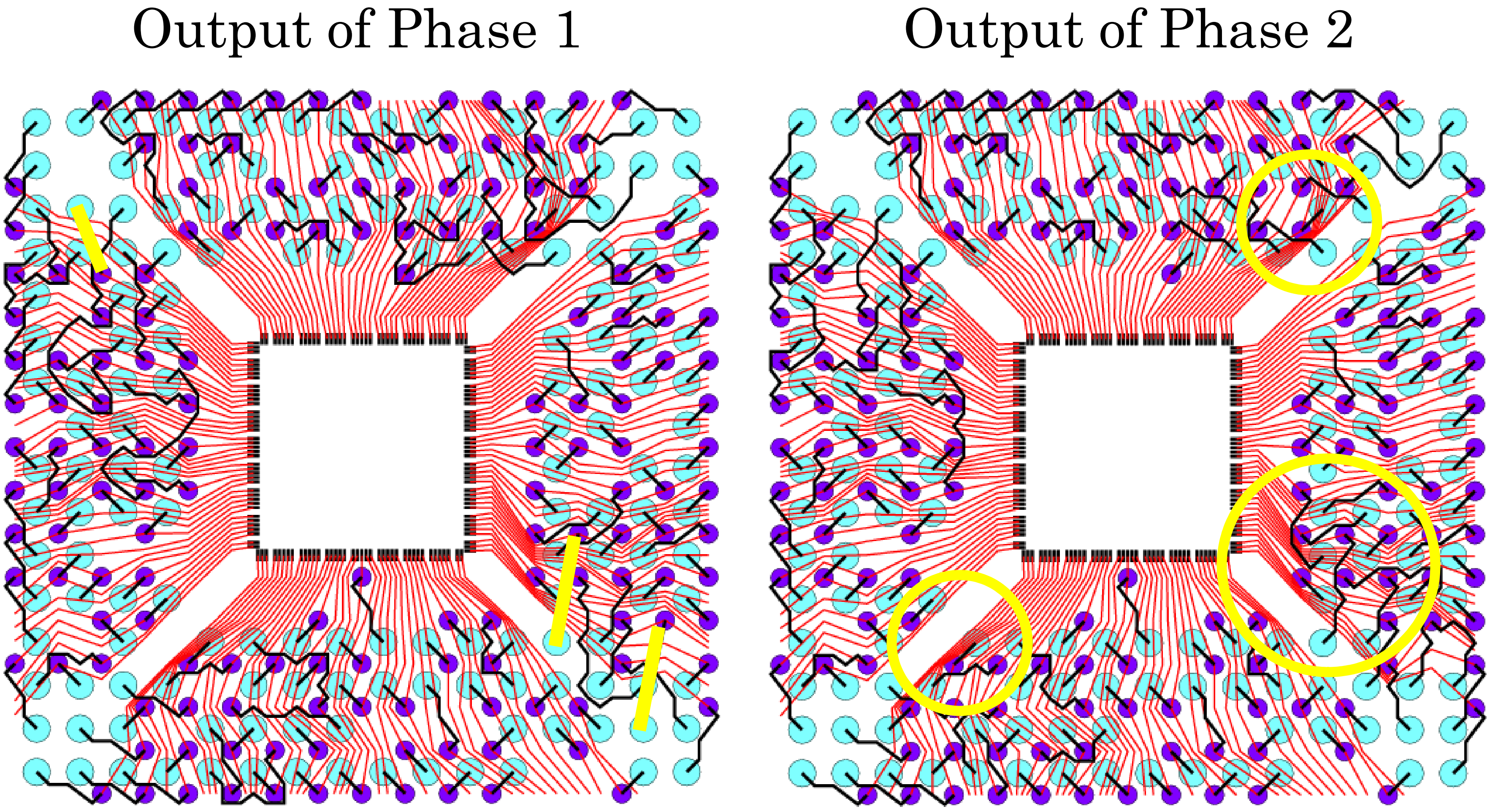
Exceed one wire between adjacent grid nodes

Try to connect many unconnected nets



Output Examples (Data4)

Completion ratio is improved without changing routes on layer 1 drastically



$\Delta = 11.5$

Conclusions

- Propose via assignment method based on Kubo's method
  - In phase 1, 10 times faster than Kubo's method
  - Improve the routability drastically and speedily

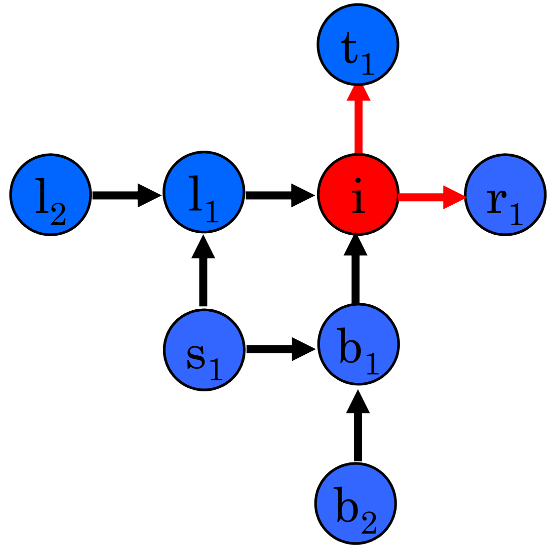
Future Work

- Initial via assignment
- Realization of plating leads on both layers
- Multi-layer via assignment and global routing

Quick package routing system
by using properties of BGA package

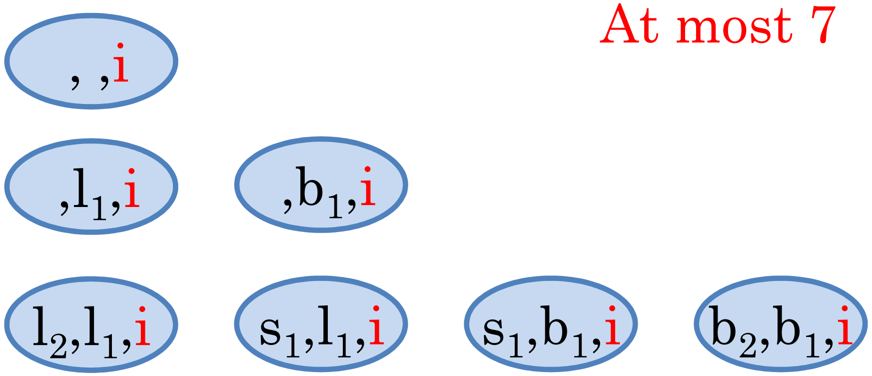
Thanks for your attentions !

#Vertices and #Edges in Our Cost Graph



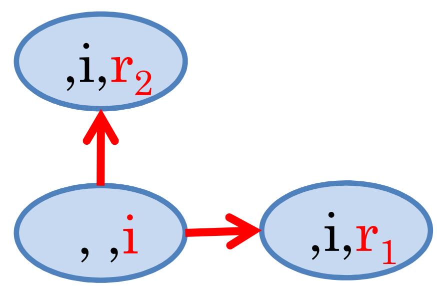
#(vertices whose last element is i)

At most 7



#(outgoing edges of a vertex)

At most 2

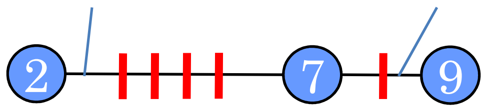


#vertices = $O(N)$ → #edges = $O(N)$

Cost Function of Phase 1

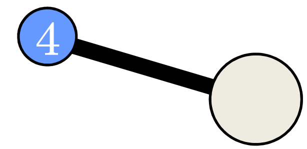
Wire congestion balance on L1 F

left congestion=2 right congestion=1



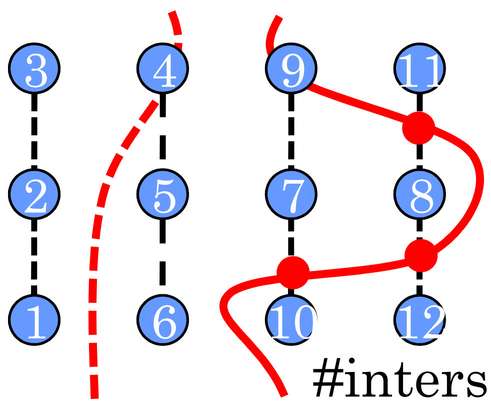
Congestion balance = | left – right |=1

Wire length on L2 D



Manhattan distance

Wire length on L1 C



#intersections = 3

#intersections = 0

Penalty for mold gates OBS

#(vias on mold gates)

$$COST_1(\Phi) = \alpha_1 C + \beta_1 D + \gamma_1 F + \delta_1 OBS$$

In our experiments, $\alpha_1 = \beta_1 = \gamma_1 = 1$ $\quad \alpha_1, \beta_1, \gamma_1 << \delta_1$

Cost Function of Phase 2

Wire congestion violations on L1 Δ

(violation) = MAX{(wire congestion) – (allowable congestion), 0}

Wire length on L2 L

Total wire length on the routing graph

Unconnected nets on L2 U

#(unconnected nets on the routing graph)

$$COST_2(\Phi) = \alpha_2\Delta + \beta_2 L + \gamma_2 U$$

In our experiments, $\alpha_2 = 1, \beta_2 = \frac{1}{4}$ $\alpha_2, \beta_2 << \gamma_2$