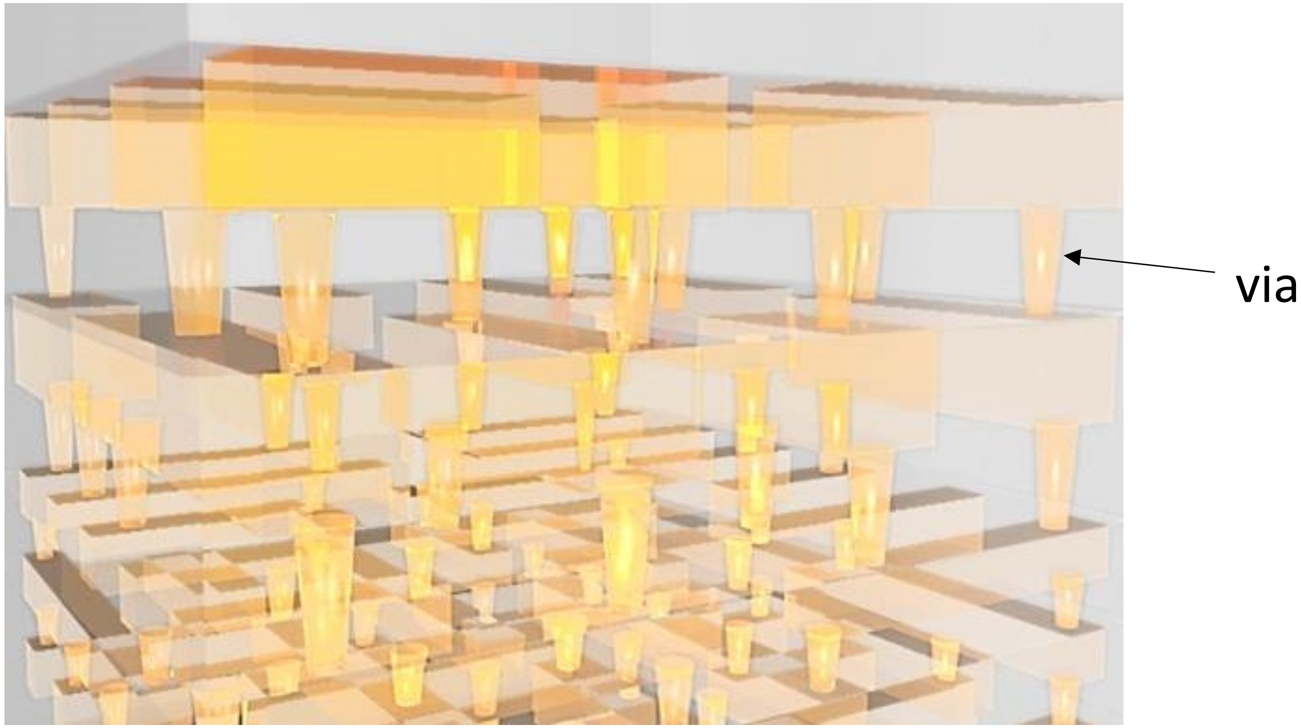
Redundant Via Insertion

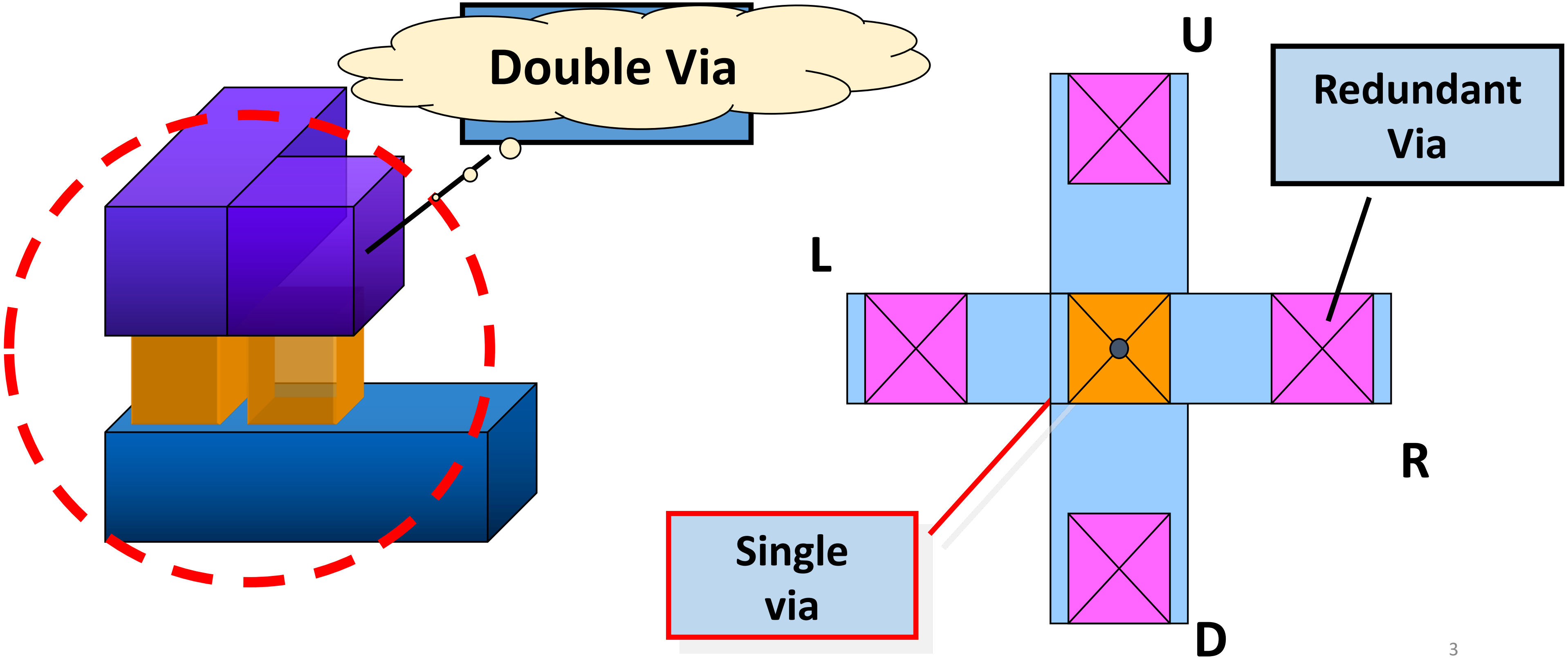
Introduction

- Vias connect wire segments on different metal layers
- A chip may have billions of vias
- One via defect can render an entire IC useless



Redundant Via

- Enable a single via failure to be tolerated
- Improve the chip yield and reliability



Post-Routing Redundant Via Insertion for Yield/Reliability Improvement

K.Y. Lee and T.C. Wang

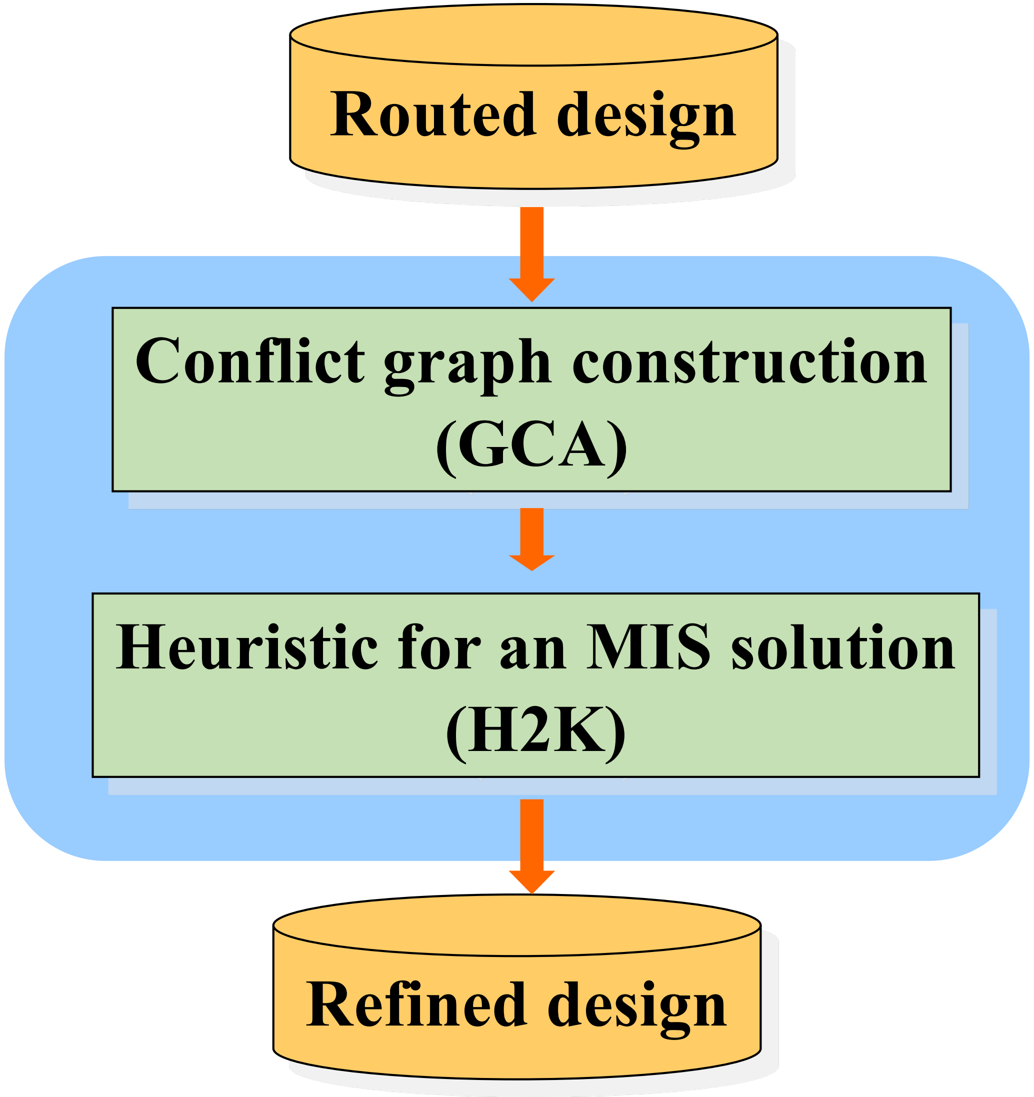
Optimal Post-Routing Redundant Via Insertion

K.Y. Lee, C.K. Koh, T.C. Wang, K.Y. Chao

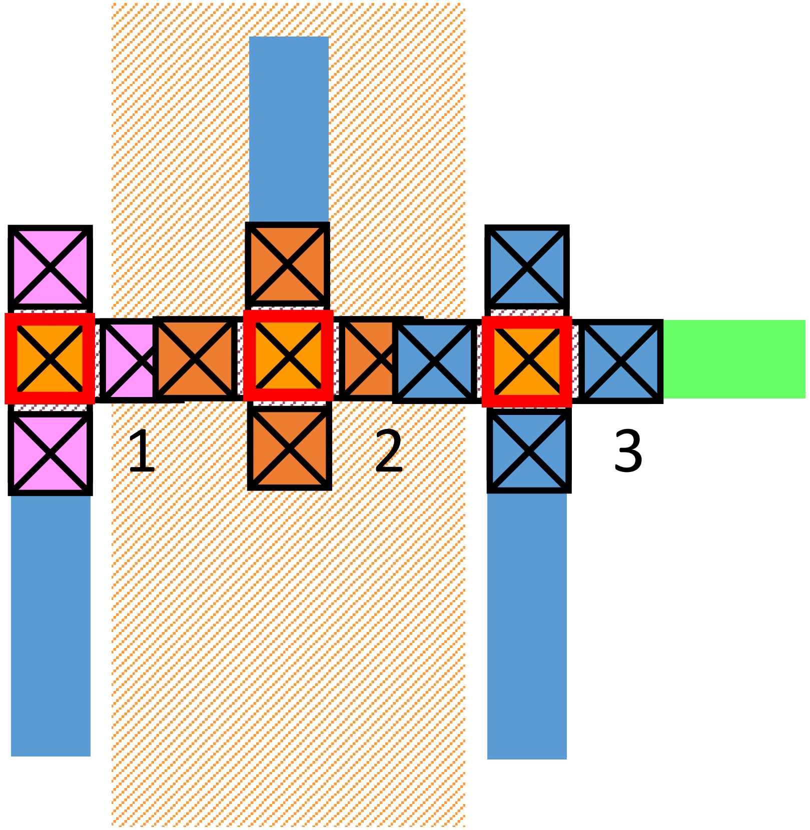
Post-Routing Double Via Insertion (DVI)

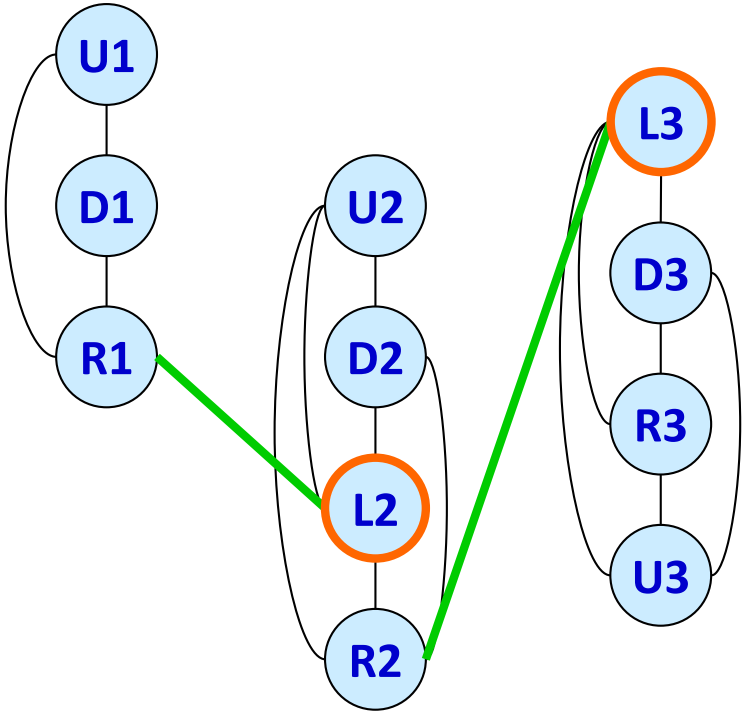
- **Input**
  - A routed design and a set of via-related design rules
- **Goal**
  - To replace as many single vias with double vias as possible
- **Constraints**
  - Do not re-route any net
  - Each single via either remains unchanged or is replaced by a double via
  - After replacement, no design rule is violated

Maximum Independent Set (MIS)-based approach to DVI [Lee+ ASPDAC06]


Routed design
Conflict graph construction (GCA)
Heuristic for an MIS solution (H2K)
Refined design

Conflict Graph Construction





Heuristic for solving the MIS Problem – H2K

- H2K solves the MIS problem on a conflict graph in an iterative manner
- In each iteration, a subgraph of size *k* is extracted from the conflict graph, a maximal independent set solution to the subgraph is sought and added to the final solution, and the conflict graph is updated

H2K (cont'd)

- For the conflict graph *G*(*V,E*), we construct a priority queue *Q* of *V* by using the **feasible number** and **degree** of a vertex as the first and second keys.
  - We give a vertex a **higher** priority if it has **smaller** feasible number and degree.
- Feasible no. of a vertex = # other feasible double vias originating from the same single via

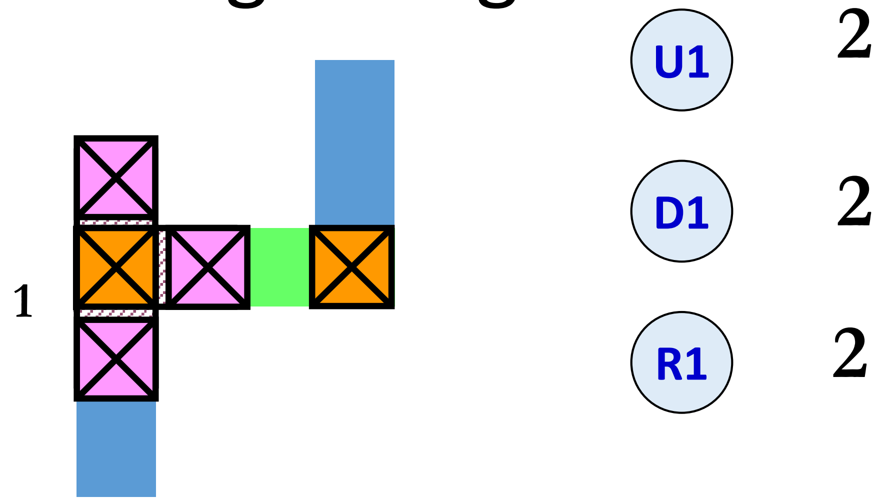
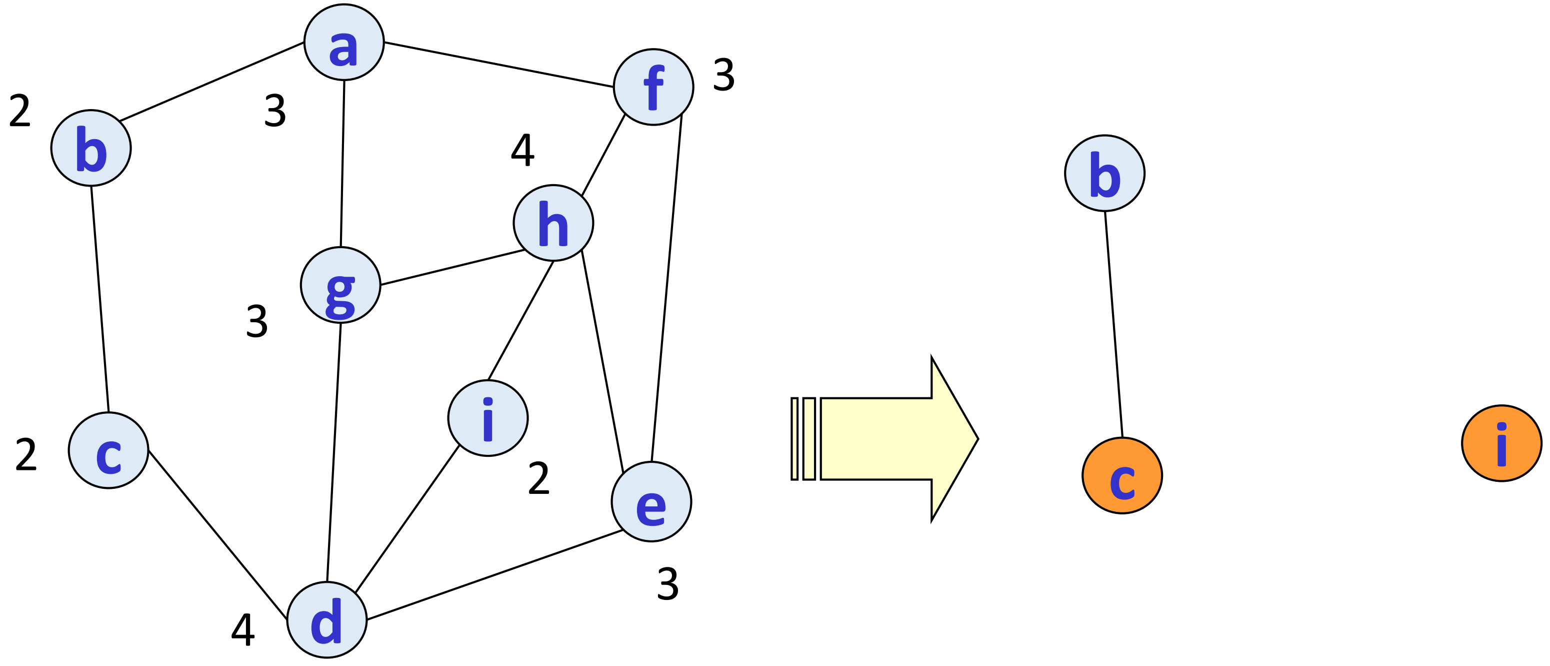


Illustration of H2K

Assume k = 3



high priority → low

b	c	i	a	e	f	g	d	h

Priority queue

c	i

MIS

Illustration of H2K (cont'd)

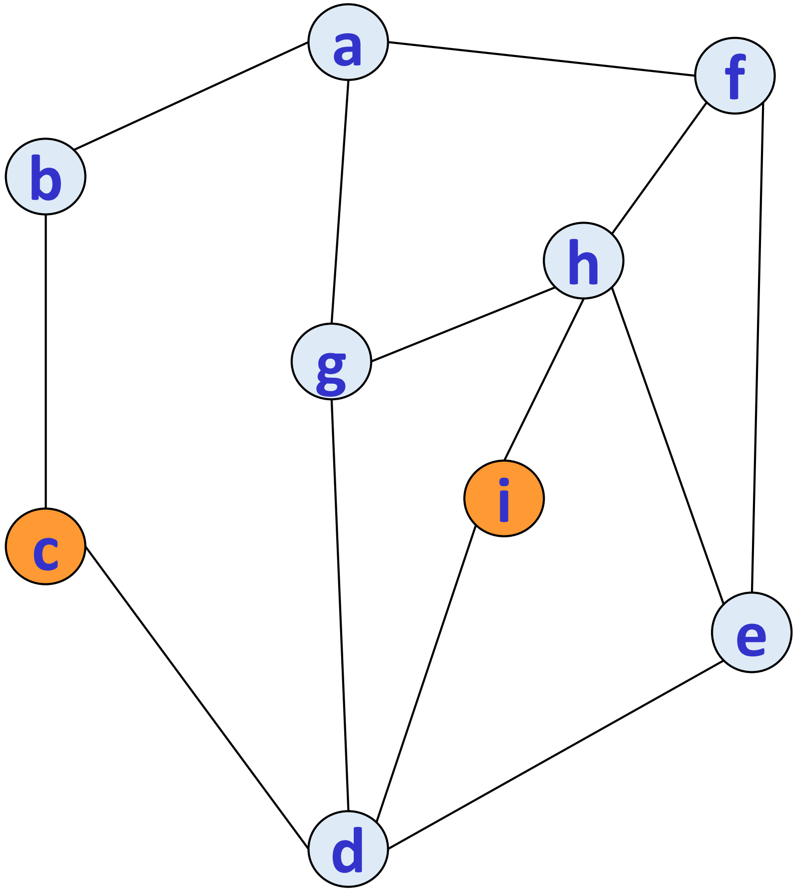
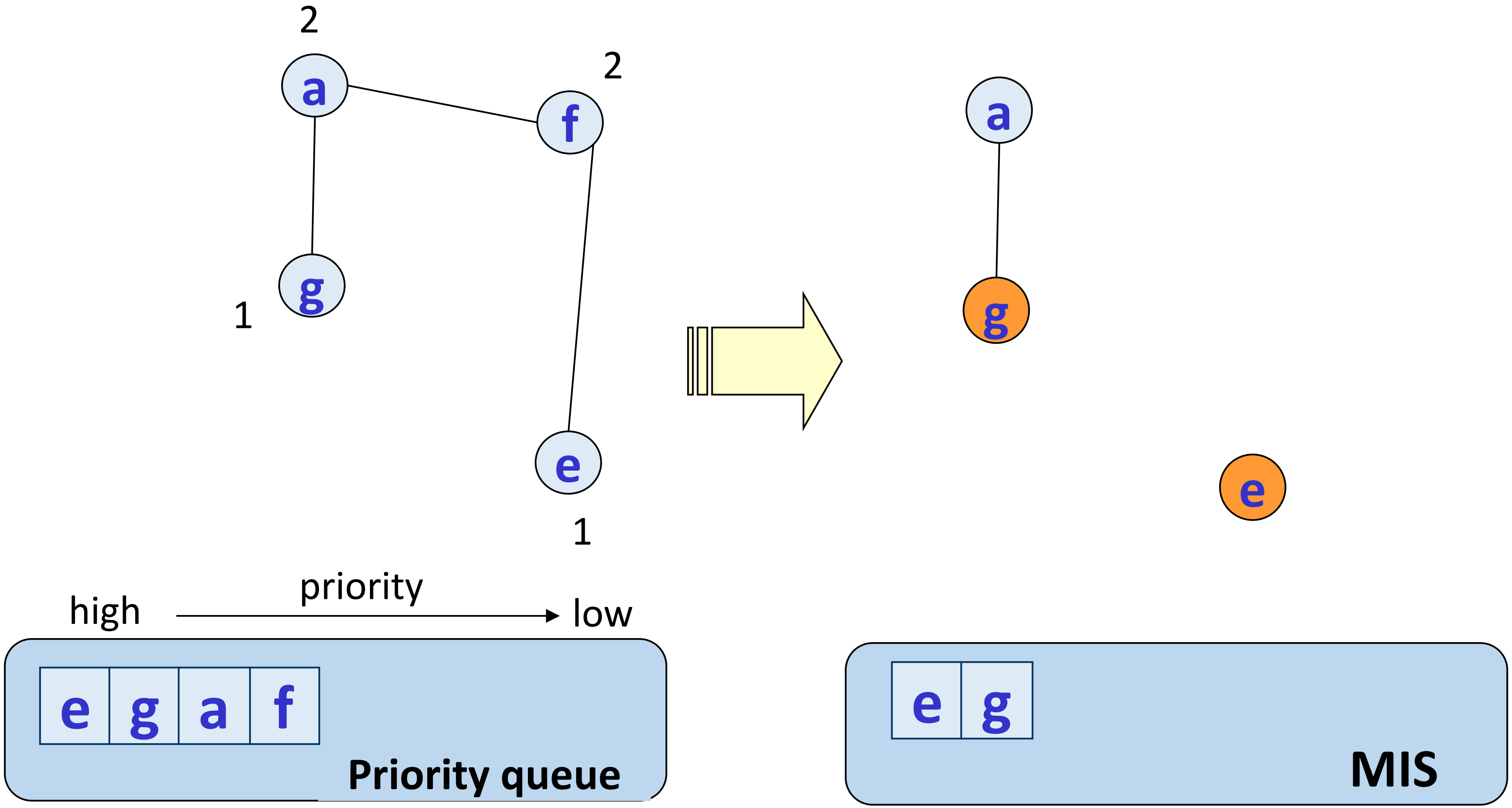


Illustration of H2K (cont'd)



high priority low

e g a f

Priority queue

e g

MIS

Illustration of H2K (cont'd)

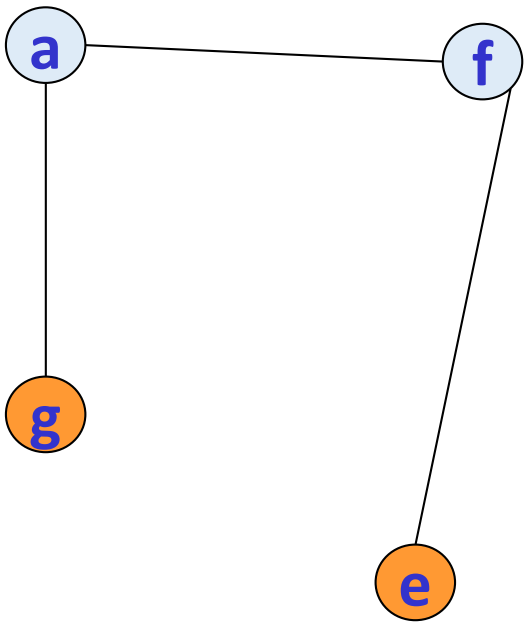
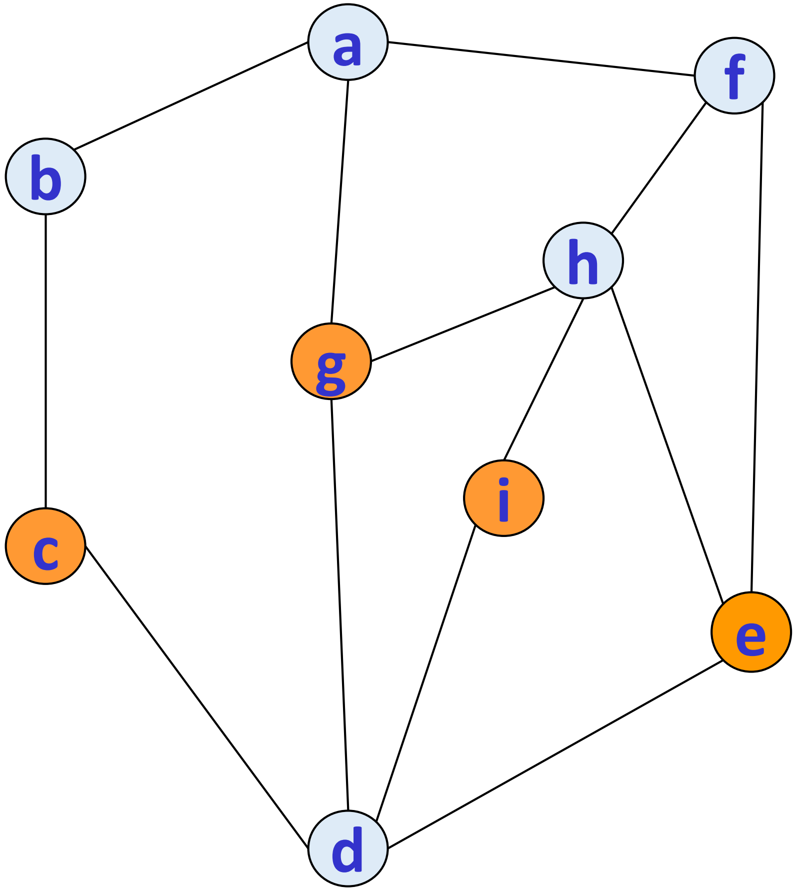


Illustration of H2K (cont’d)



0-1 ILP approach to DVI [Lee+ ISPD08]

Maximize $$\sum_{1 \le i \le 10} R_i$$

Subject to

$$R_2 + R_5 \le 1 \quad R_5 + R_6 \le 1 \quad R_1 + R_2 \le 1$$

$$R_2 + R_9 \le 1 \quad R_6 + R_7 \le 1 \quad R_1 + R_3 \le 1$$

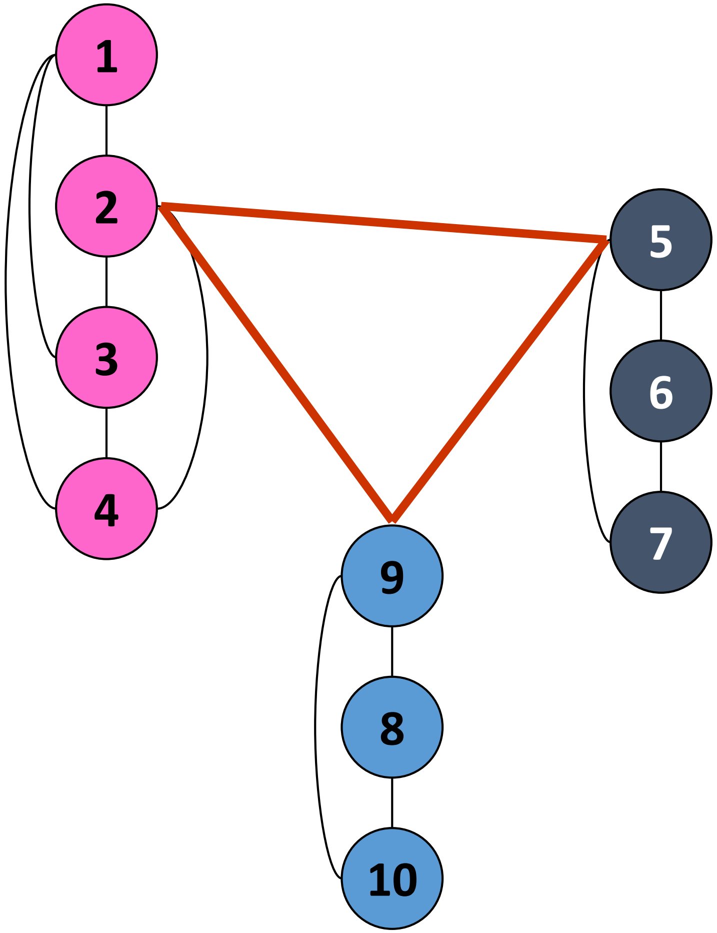
$$R_5 + R_9 \le 1 \quad R_5 + R_7 \le 1 \quad R_1 + R_4 \le 1$$

$$R_8 + R_9 \le 1 \quad R_2 + R_3 \le 1$$

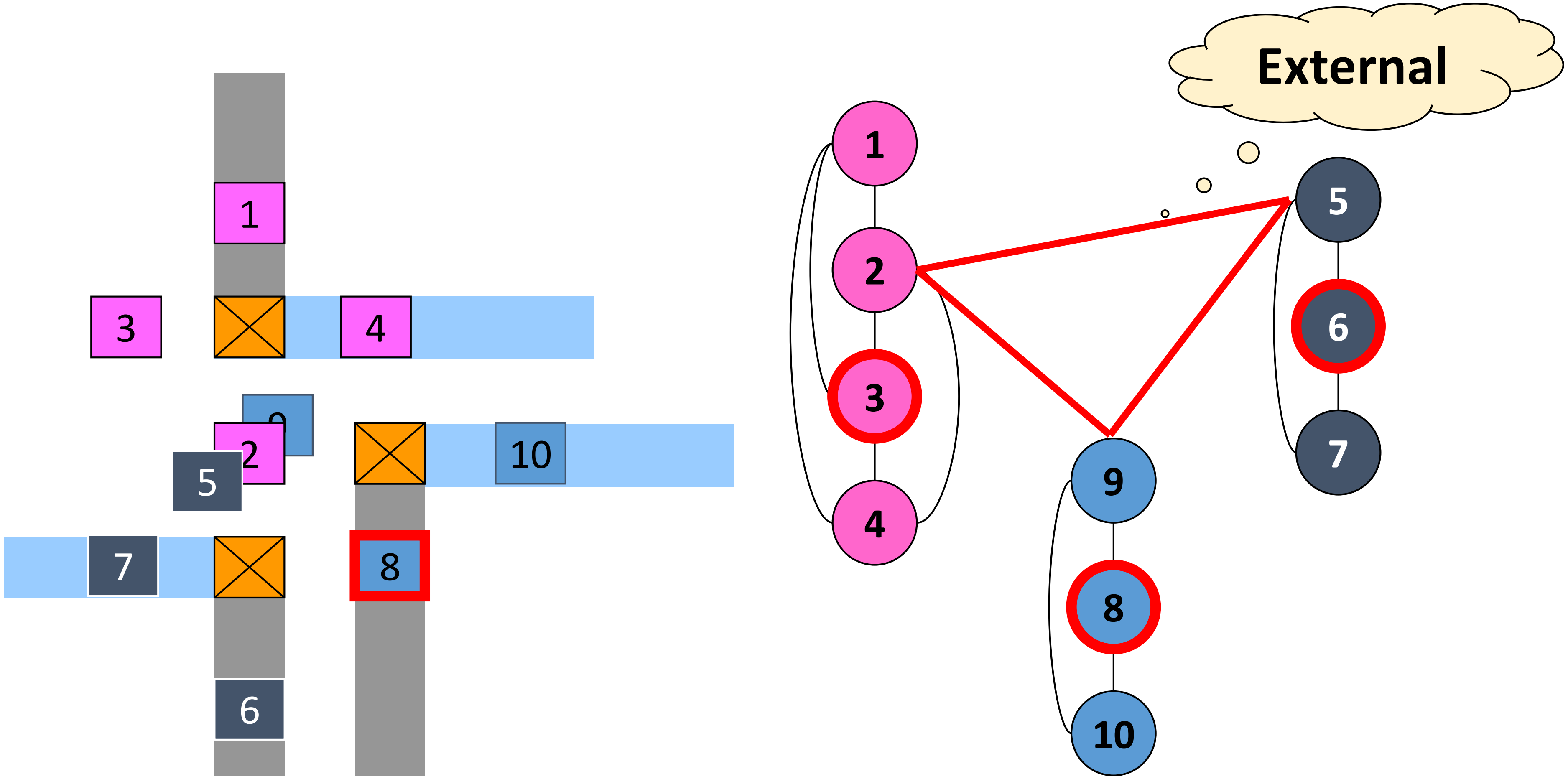
$$R_9 + R_{10} \le 1 \quad R_2 + R_4 \le 1$$

$$R_8 + R_{10} \le 1 \quad R_3 + R_4 \le 1$$

$$R_i \in \{0,1\}$$



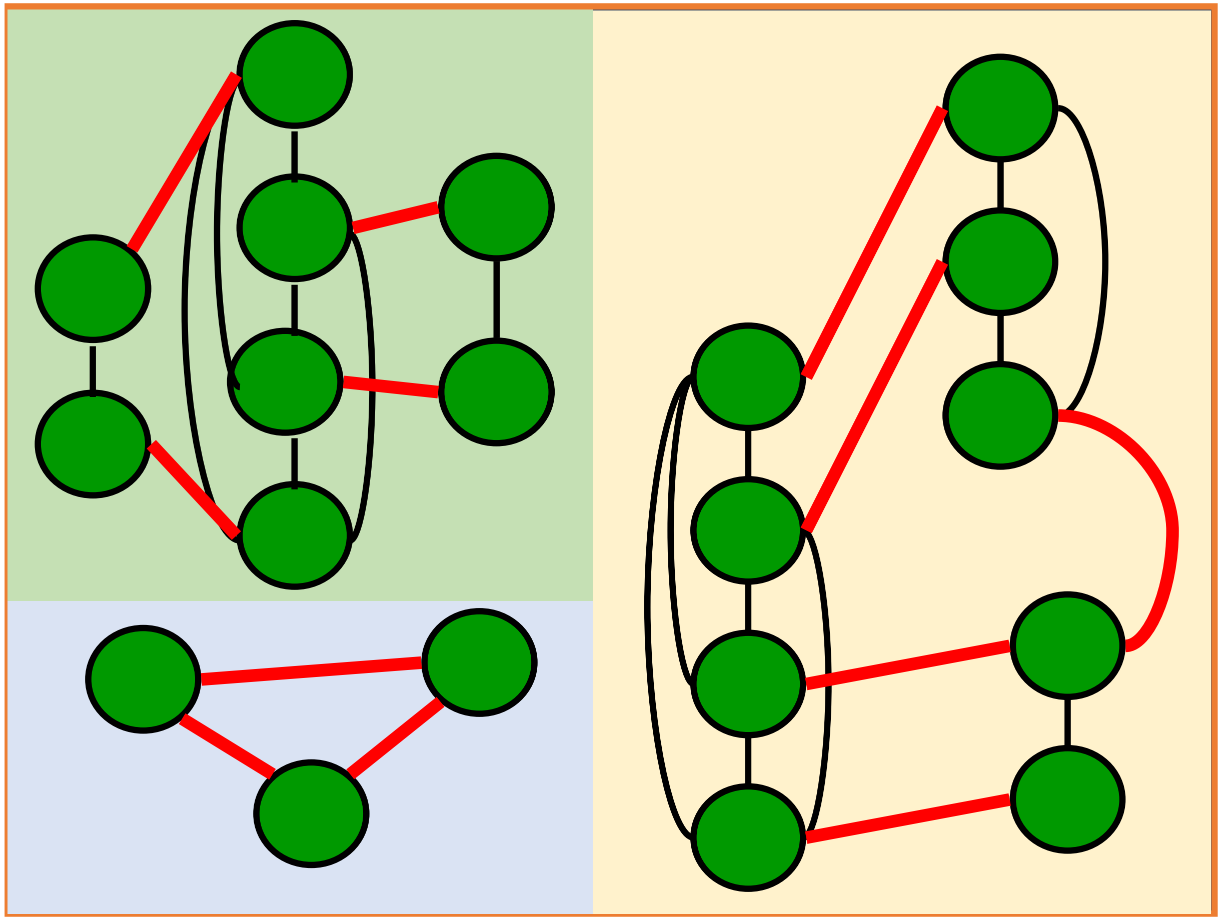
Speed-up – Pre-selection



- We can efficiently pre-select a subset of vertices
- The size of given conflict graph can be reduced
- May even solve the whole DVI problem directly

Speed-up – Connected Components

- Divide into smaller 0-1 ILP problems
  - Compute connected components (by depth-first-search algorithm)



Speed-up – Reduction in Constraints

$R_2 + R_5 \le 1$

$R_2 + R_9 \le 1$

$R_5 + R_9 \le 1$

$R_5 + R_6 \le 1$

$R_6 + R_7 \le 1$

$R_5 + R_7 \le 1$

$R_8 + R_9 \le 1$

$R_9 + R_{10} \le 1$

$R_8 + R_{10} \le 1$

$R_1 + R_2 \le 1$

$R_1 + R_3 \le 1$

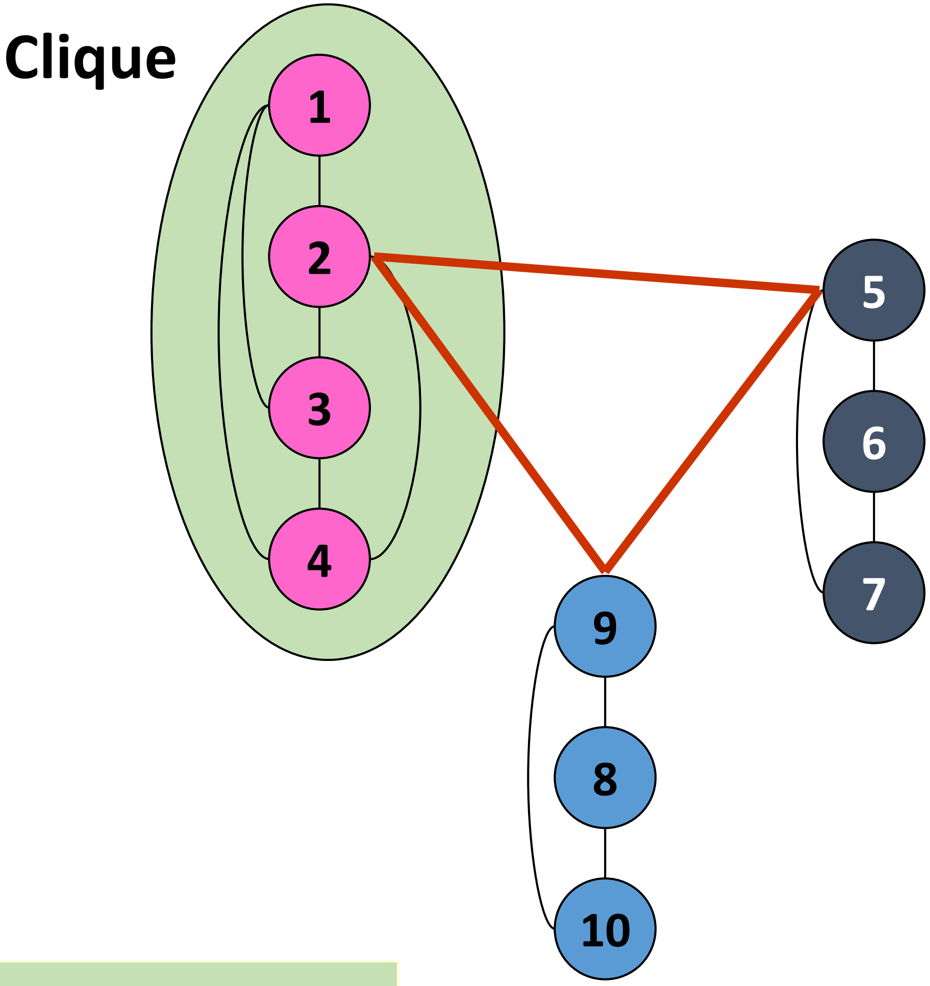
$R_1 + R_4 \le 1$

$R_2 + R_3 \le 1$

$R_2 + R_4 \le 1$

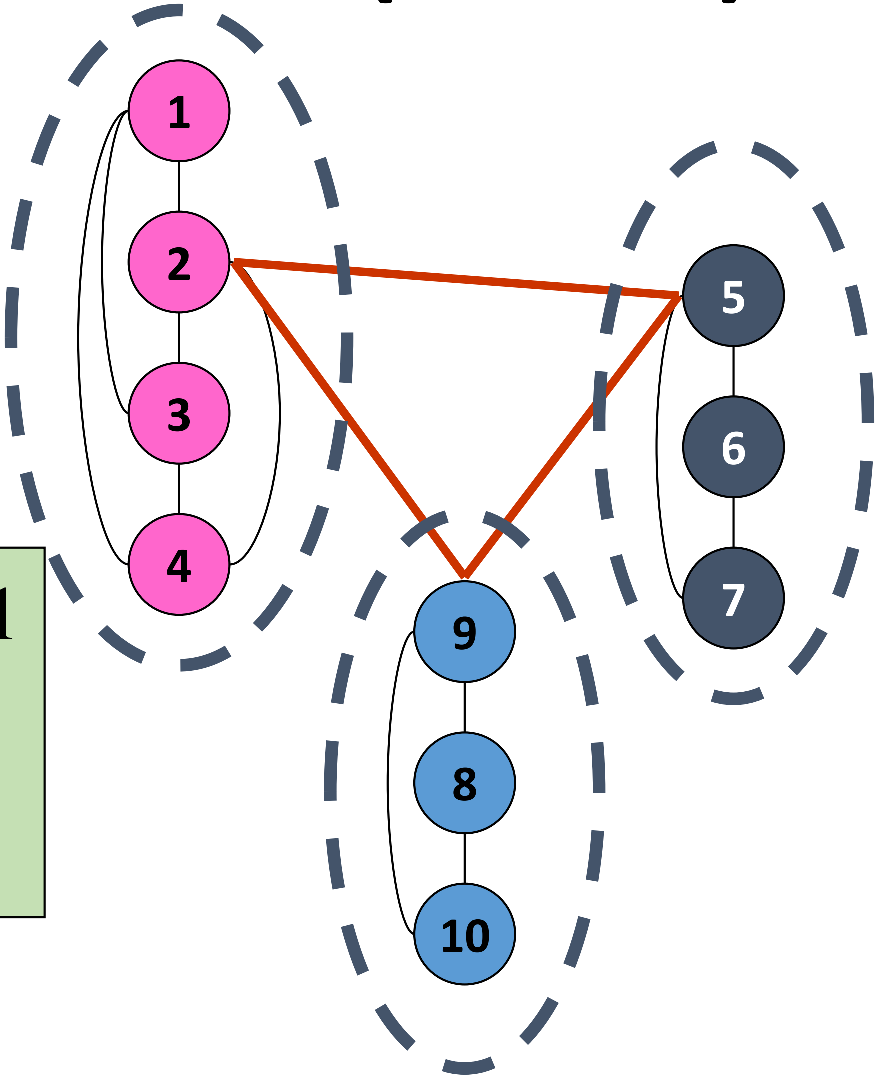
$R_3 + R_4 \le 1$

$R_1 + R_2 + R_3 + R_4 \le 1$



$R_i \in \{0,1\}$

Speed-up – Reduction in constraints (cont'd)



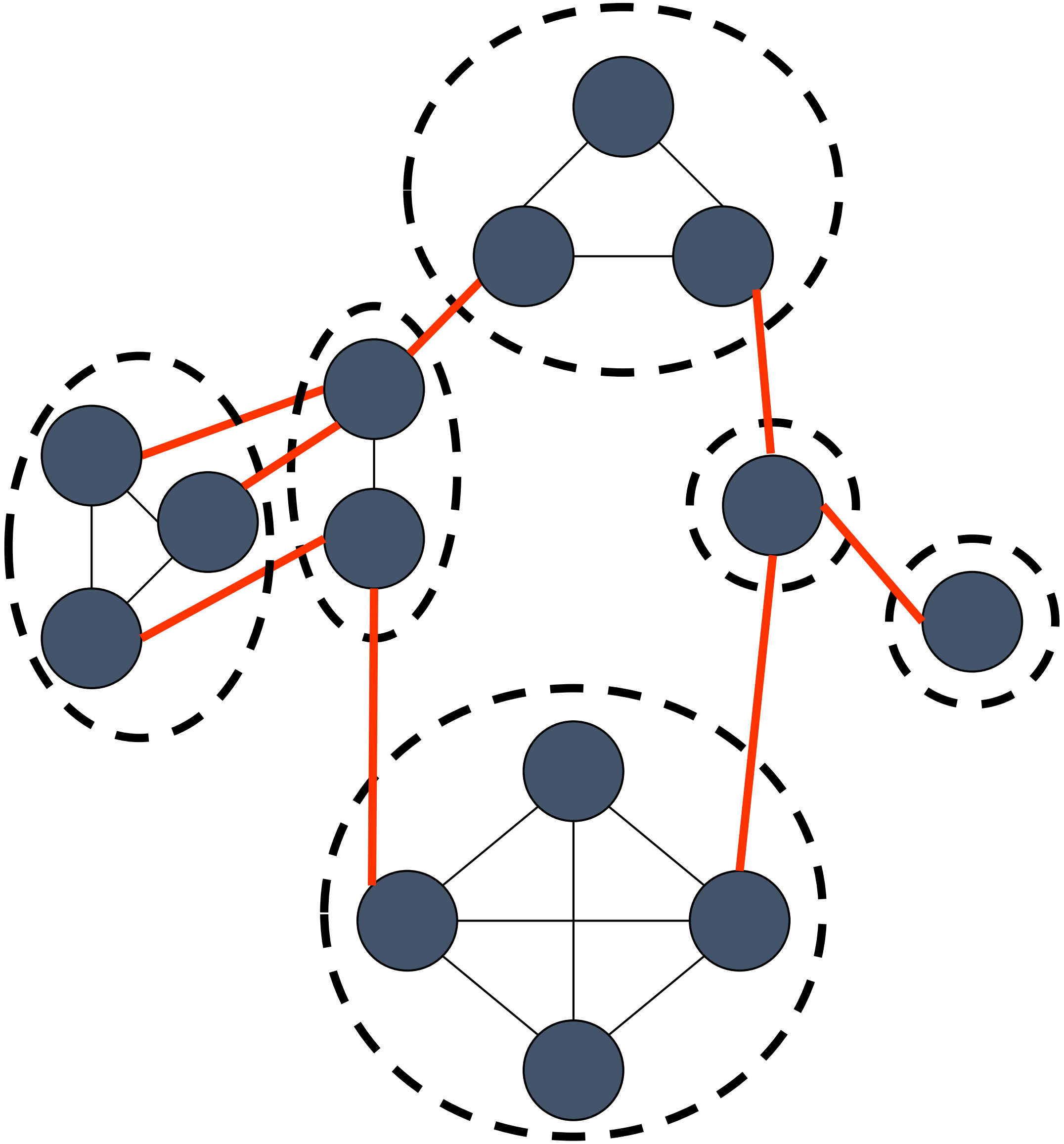
Maximize $$\sum_{1 \le i \le 10} R_i$$

Subject to

$$R_2 + R_5 \le 1 \quad \boxed{R_1 + R_2 + R_3 + R_4 \le 1}$$
$$R_2 + R_9 \le 1 \quad \boxed{R_5 + R_6 + R_7 \le 1}$$
$$R_5 + R_9 \le 1 \quad \boxed{R_8 + R_9 + R_{10} \le 1}$$
$$R_i \in \{0,1\}$$

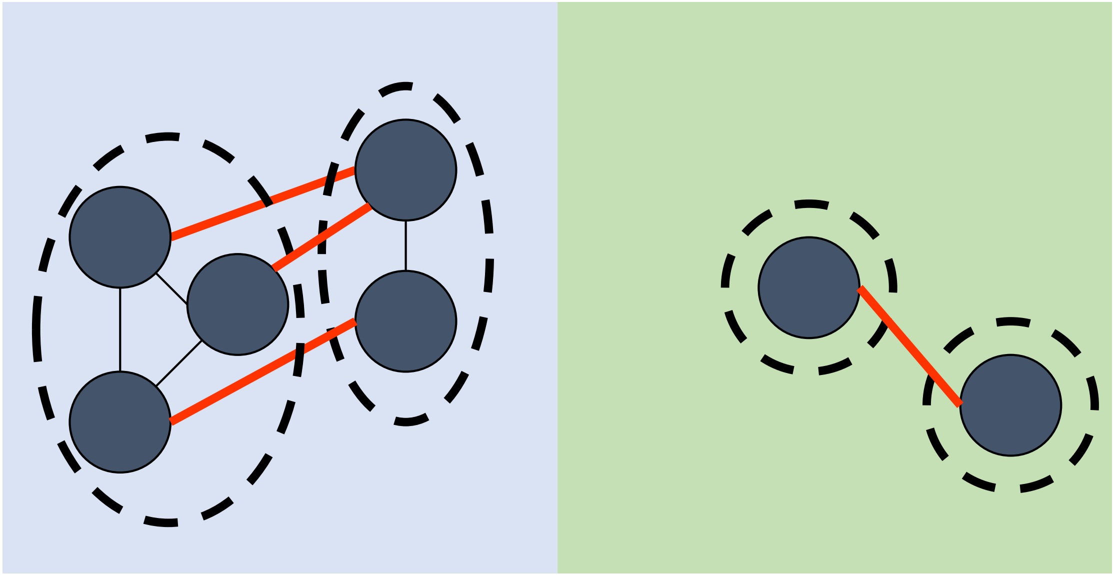
- # of inequalities: 15 → 6

Overall approach



1. Pre-selection

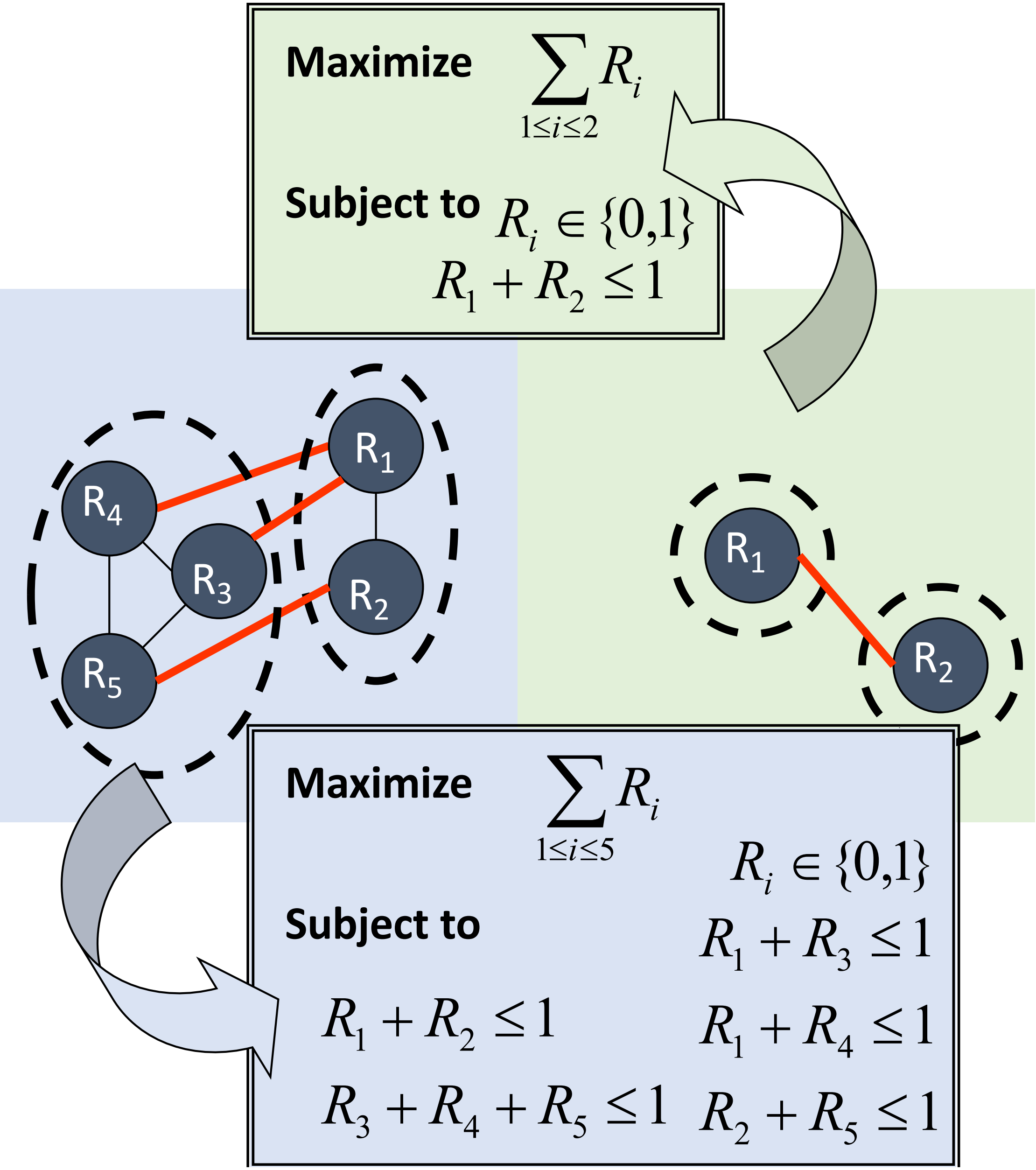
Overall approach



1. Pre-selection

2. Connected Components

Overall Approach



Maximize $\sum_{1 \le i \le 2} R_i$

Subject to $R_i \in \{0,1\}$

$R_1 + R_2 \le 1$

Maximize $\sum_{1 \le i \le 5} R_i$

Subject to

$R_i \in \{0,1\}$

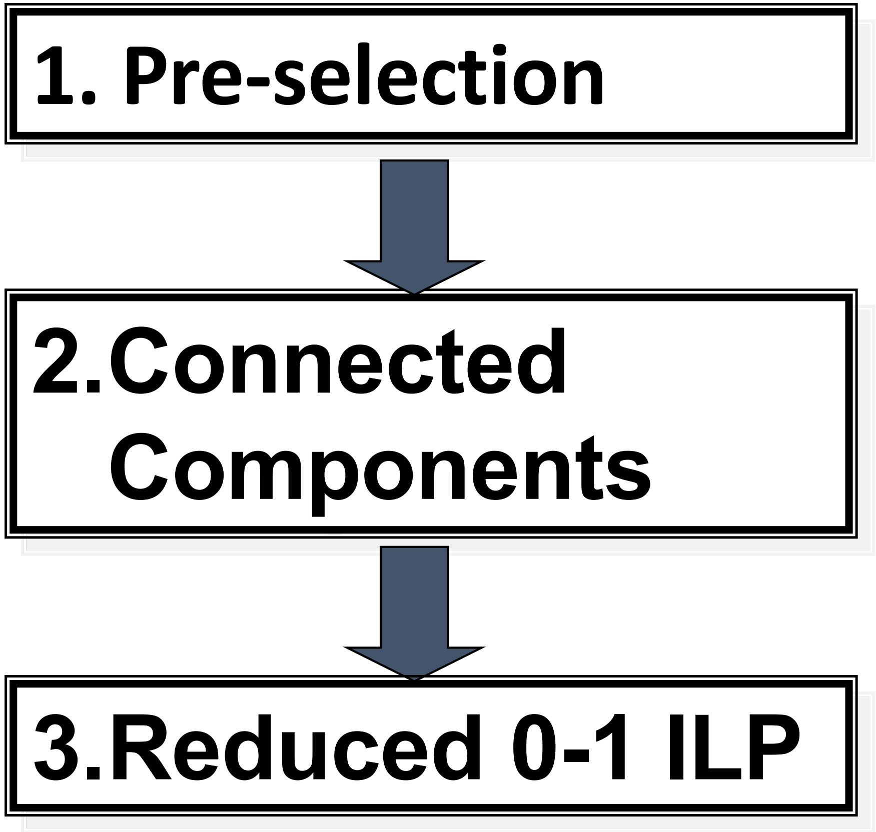
$R_1 + R_3 \le 1$

$R_1 + R_2 \le 1$

$R_1 + R_4 \le 1$

$R_3 + R_4 + R_5 \le 1$

$R_2 + R_5 \le 1$

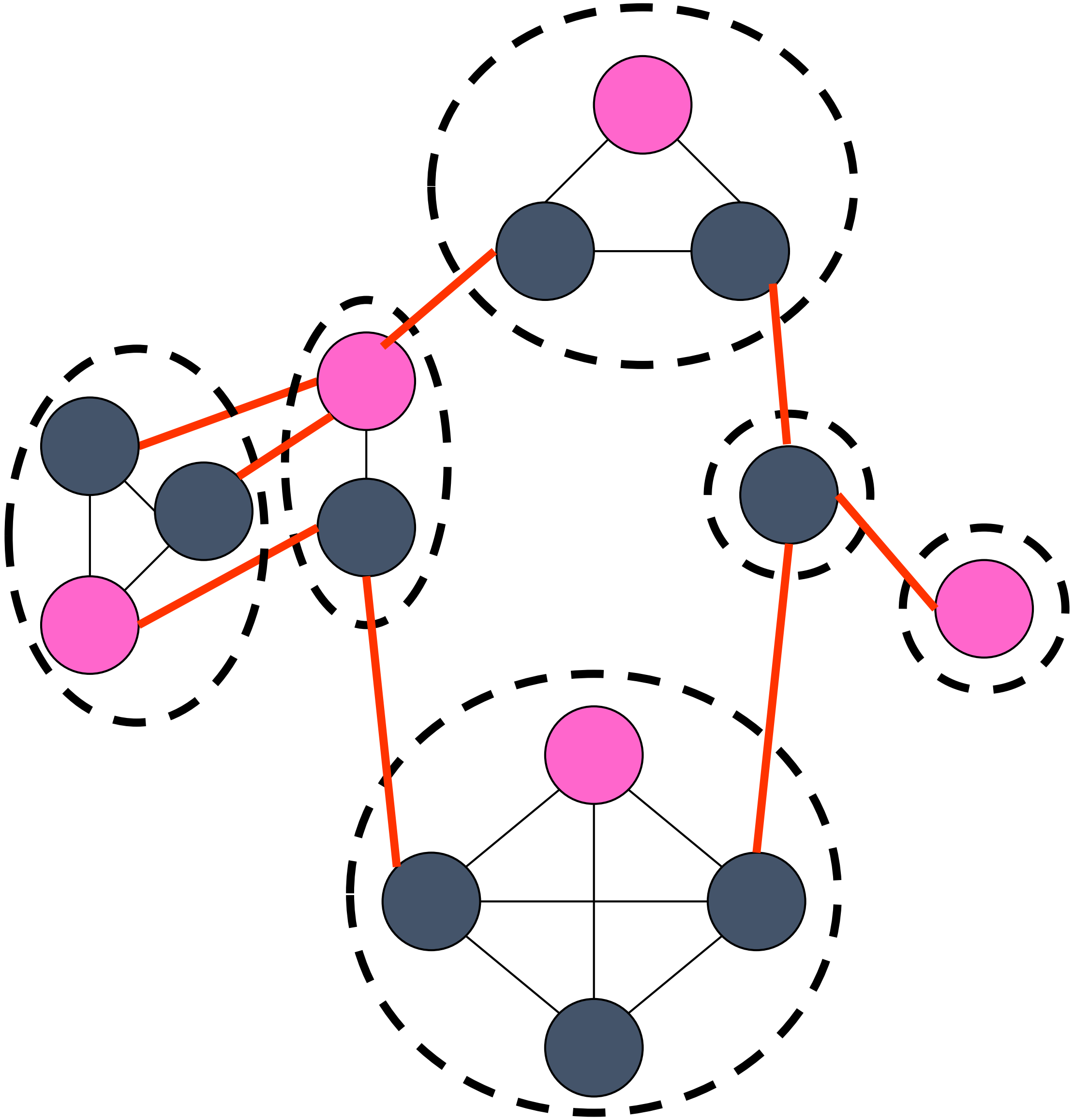


1. Pre-selection

2.Connected Components

3.Reduced 0-1 ILP

Overall Approach (cont'd)



Statistics on Test Circuits

Circuit	Size(μm)	#Nets	#I/Os	#Vias	#Layers
C1	350.000 *350.000	4309	20	24594	5
C2	419.433 *413.28	5252	211	41157	5
C3	799.124 *776.16	18157	85	127059	5
C4	691.272 *680.400	17692	415	151912	5
C5	1383.482 *1375.92	44720	99	357386	5

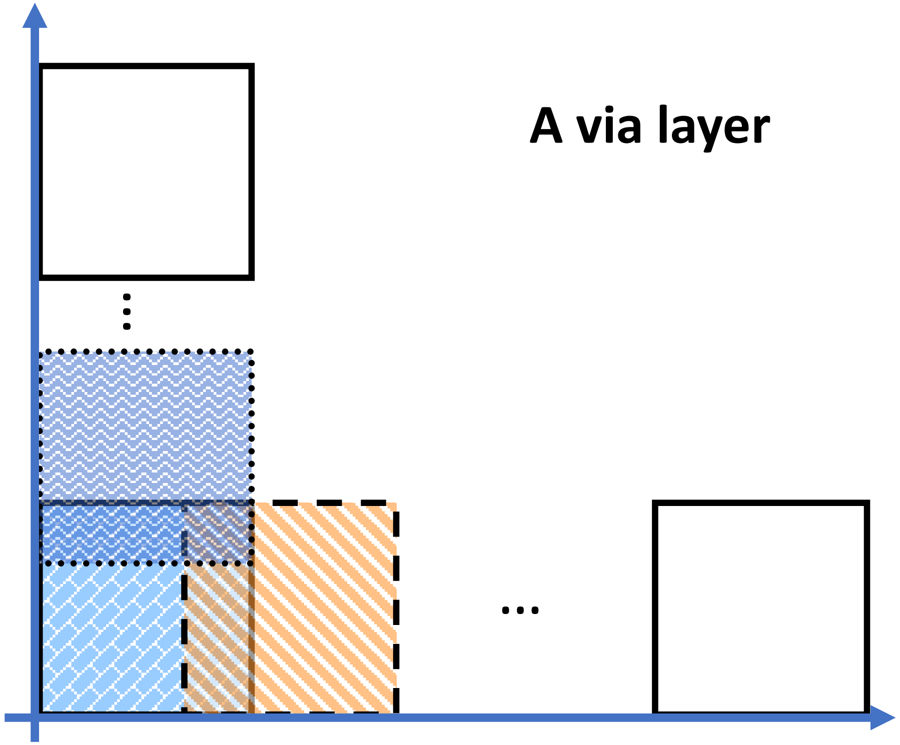
DVI Results

Circuit	Tool	MIS		0-1 ILP	
	#DVI	#DVI	T(s)	#DVI	T(s)
C1	14402	17461	5	17461	3
C2	25918	28507	11	28507	3
C3	80827	91461	86	91461	4
C4	91574	101765	86	101766	5
C5	225142	254428	104	254429	3
Normalized			15.7		1

Speed-up results

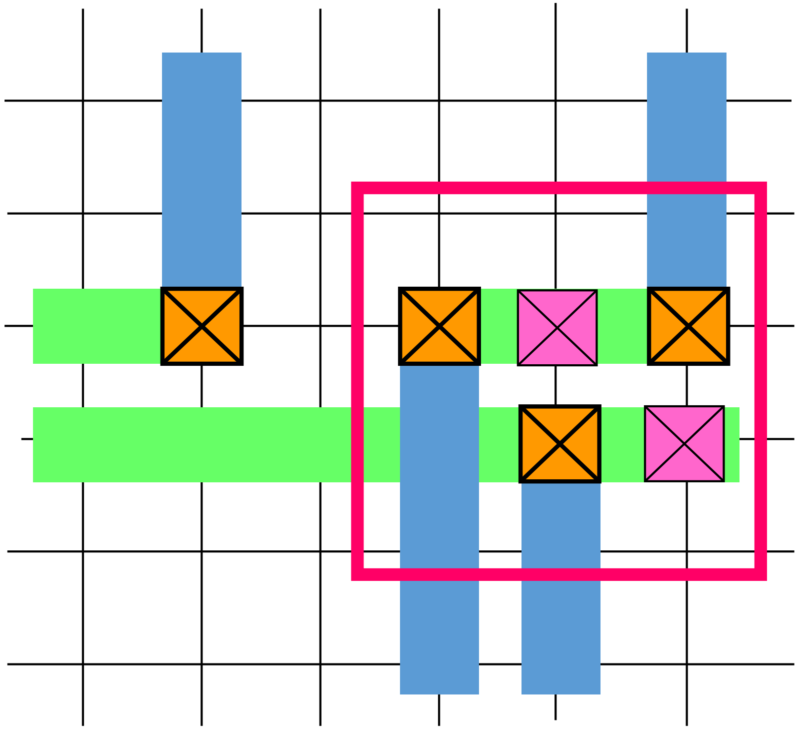
Circuit	Runtime T(s)		
	W/O	**W**	**Speed-up**
C1	**5**	**3**	**1.6X**
C2	**8**	**3**	**2.6X**
C3	**92**	**4**	**23X**
C4	**93**	**5**	**18.6X**
C5	**843**	**3**	**281X**

Additional Consideration: Via Density



A via layer

...



- Minimum via density rule
- Maximum via density rule

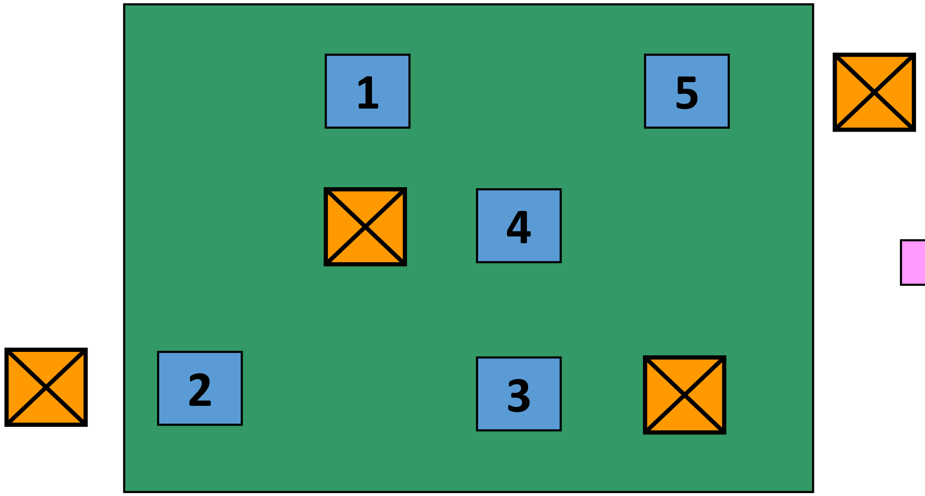
Double Via Insertion with Via Density Consideration (DVI w/ VD)

- **Input**
  - A routed design (satisfying via density rules) and a set of via-related design rules
- **Goal**
  - To replace as many single vias with double vias as possible
- **Constraints**
  - Do not re-route any net
  - Each single via either remains unchanged or is replaced by a double via
  - After replacement, no design rule is violated (including the maximum via density rule)

0-1 ILP approach to DVI w/ VD

- Add constraint for each region which may exceed maximum via density after DVI
- E.g.

max density = 5

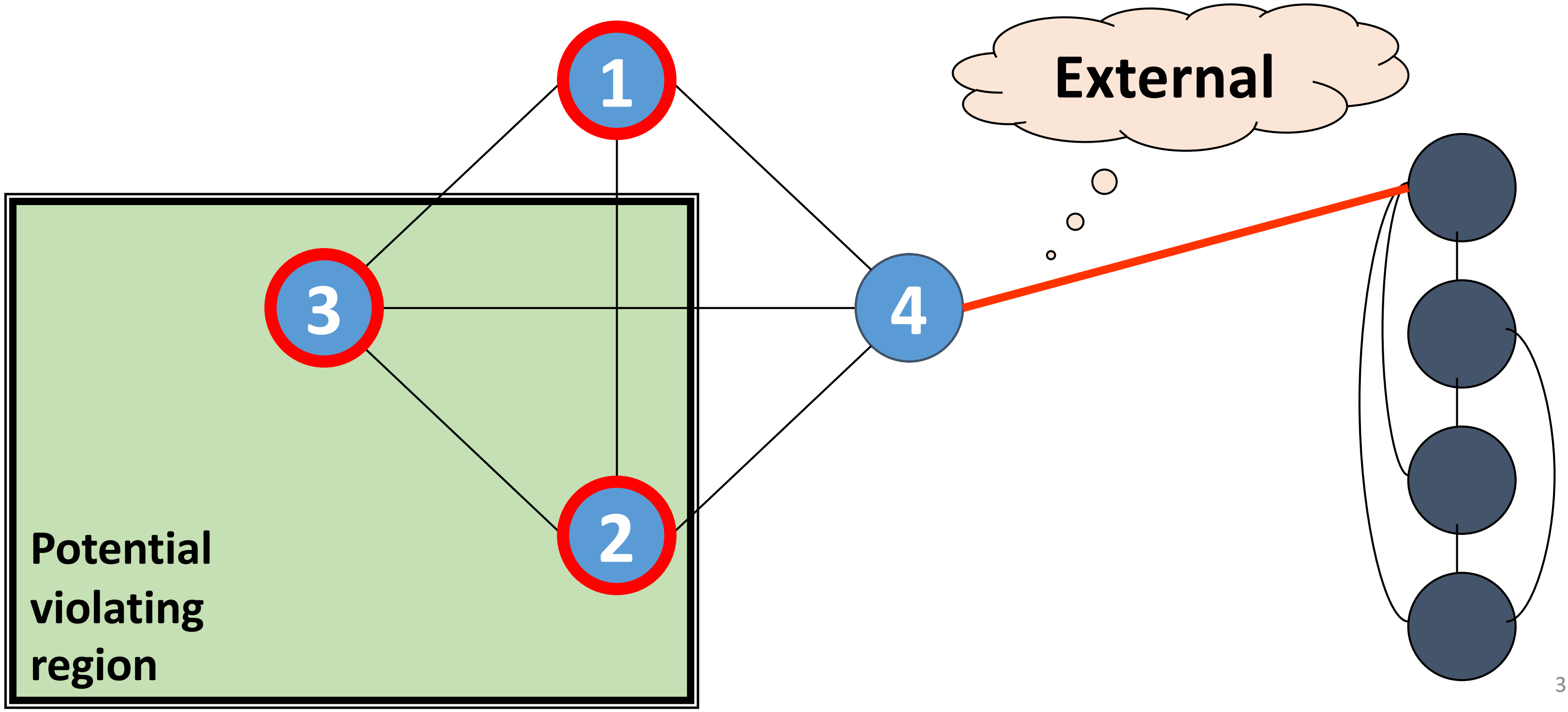


At most 3 redundant vias can be inserted

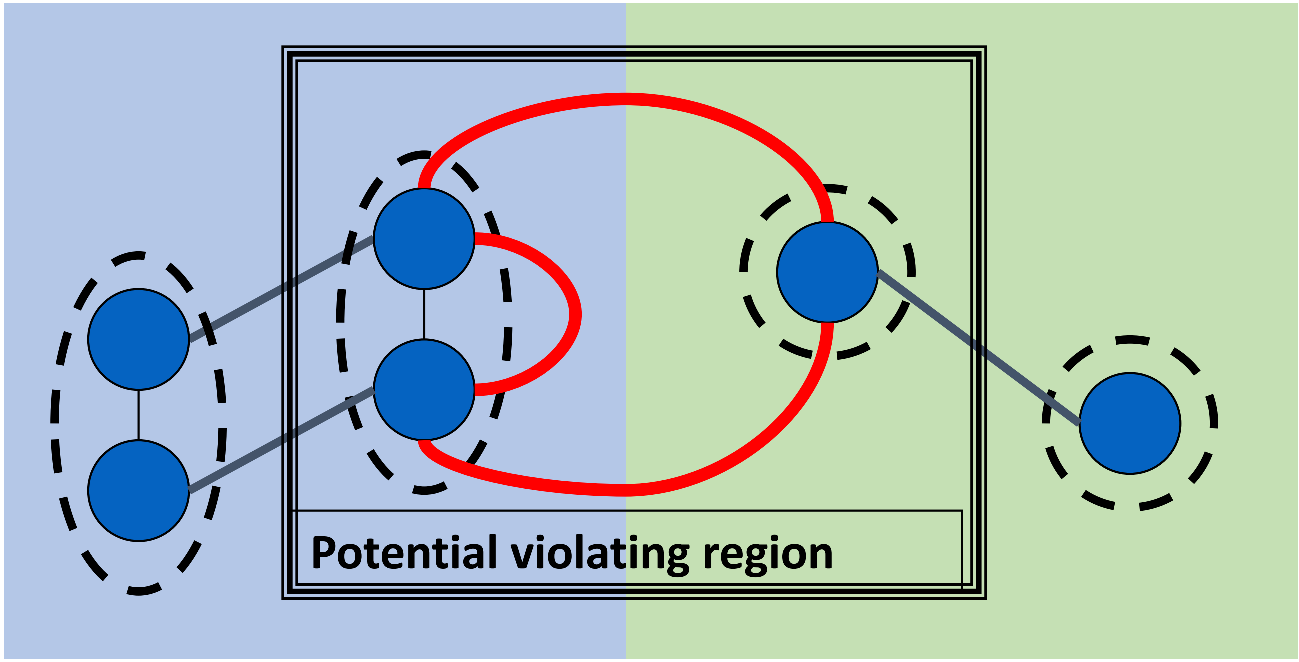
$$\boxed{R_1 + R_2 + R_3 + R_4 + R_5 \le 3}$$

Modifications – Pre-selection

- Vertex candidate for pre-selection
  - No external edges
  - Not involved in any potential violating region



Modifications – Connected Components



Avoid splitting a potential violating region into two ILPs

DVI w/ VD results

Circuit	Two-Stage		0-1 ILP	
	#DVI	T(s)	#DVI	T(s)
C1	17074	6	17249	3
C2	27881	12	28064	4
C3	89960	88	90482	5
C4	91134	100	93240	48
C5	252056	106	252768	3
Normalized		12.0		1

References

- [Lee+ ASPDAC06] Post-Routing Redundant Via Insertion for Yield/Reliability Improvement
- [Lee+ ISPD08] Optimal Post-Routing Redundant Via Insertion