A Local Optimal Method on DSA Guiding Template Assignment with Redundant/Dummy Via Insertion

Xingquan Li[1], Bei Yu[2], Jianli Chen[1], Wenxing Zhu[1],

24th Asia and South Pacific Design Automation Conference

Tokyo Japan 2019

[1]Fuzhou University

[2]The Chinese University of Hong Kong

福州大学 FUZHOU UNIVERSITY

香港中文大學 The Chinese University of Hong Kong

Outline

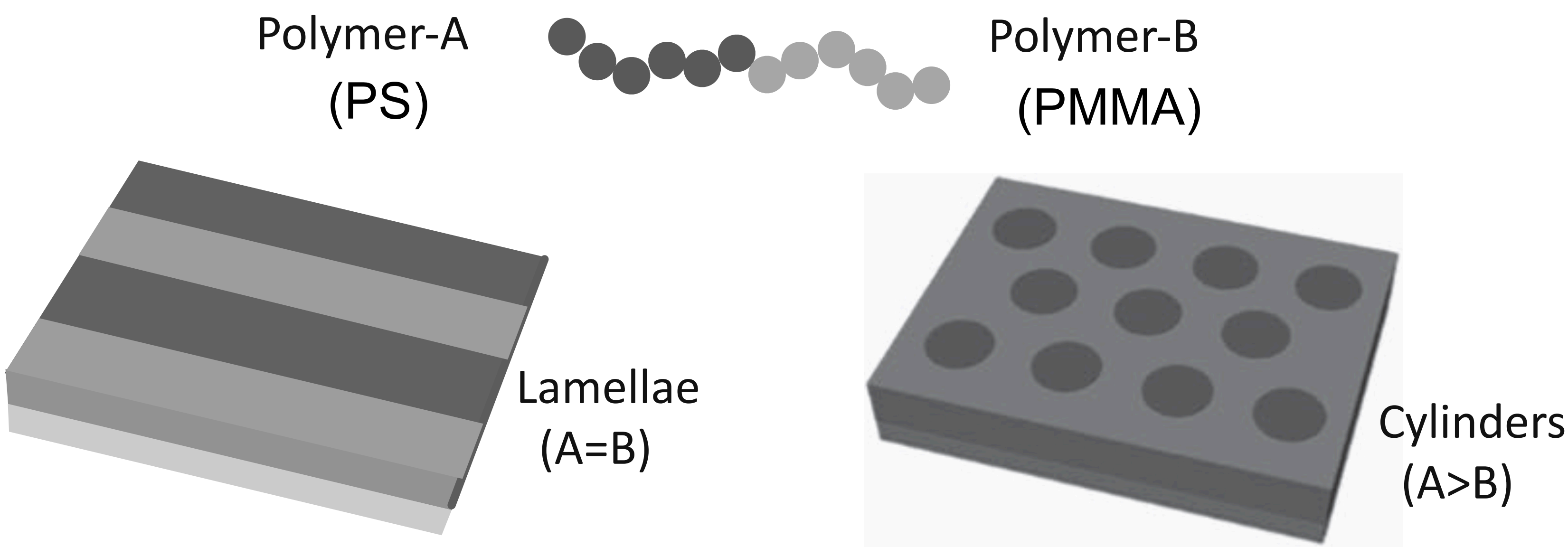
- Introductions
- Problem Formulation
- Algorithm
- Experimental Results
- Conclusions

Outline

- Introductions
- Problem Formulation
- Algorithm
- Experimental Results
- Conclusions

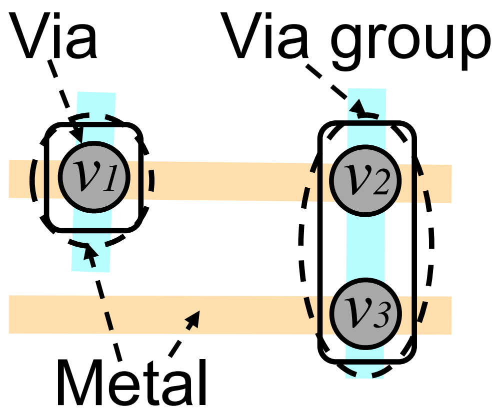
Block Copolymers Directed Self-Assembly (DSA)

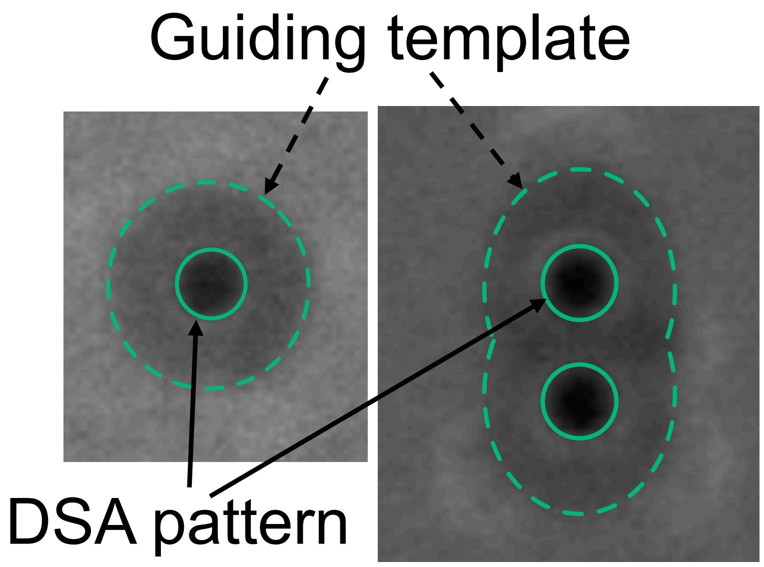
- Block copolymer (BCP)
  - A unique string of two types of polymer.
  - One type of polymer is hydrophilic and another is hydrophobic.
- Nanostructures
  - Cylinders, spheres, and lamellae.
  - The cylindrical nanostructure is suitable for patterning contacts and vias.



DSA Process

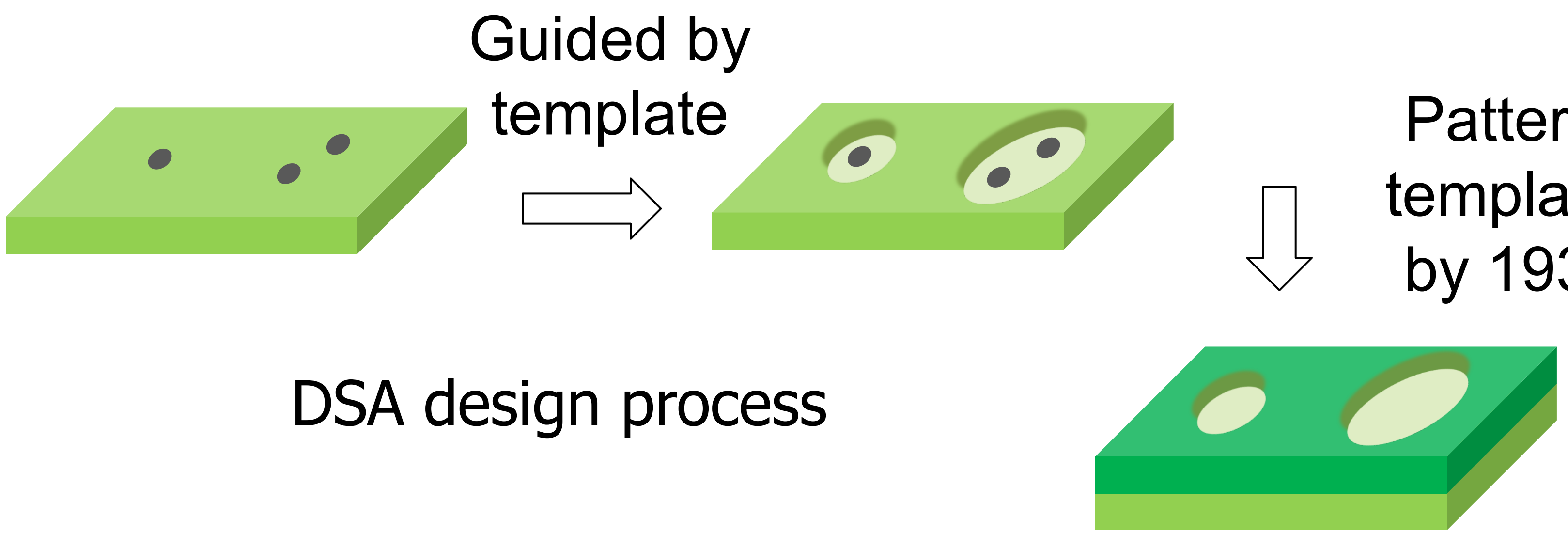
- Vias are not regularly placed in practical layout.
- A simple regular hole array generated by standard DSA is not suitable for IC fabrication.
- Topological template guided DSA process has been proposed to support patterning irregularly vias layout.
- Closed vias are grouped; And a guiding template is identified for each group.





DSA Process

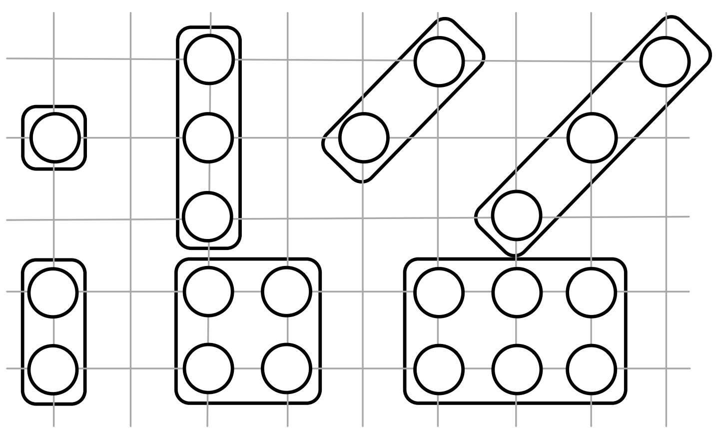
- Given a vias layout, we should assign the guiding templates for every via.
- Guiding templates are patterned on a wafer through optical lithography.
- Each guiding template is filled with BCPs.
- DSA can be controlled by thermal annealing process.

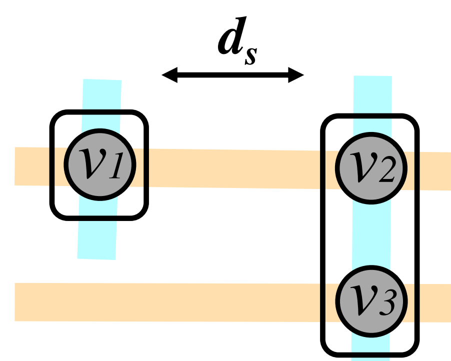


DSA design process

Guiding Templates

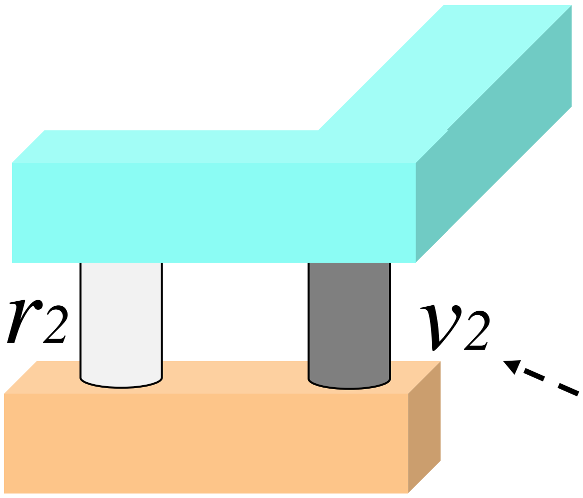
- Pre-defined DSA pattern set to improve robust.
- Within-group contact/via distance.
- Complex shapes are difficult to print.
- Unexpected holes and placement error of holes for some patterns.
- The distance of any two guiding templates should larger than minimum optical resolution spacing d_s.

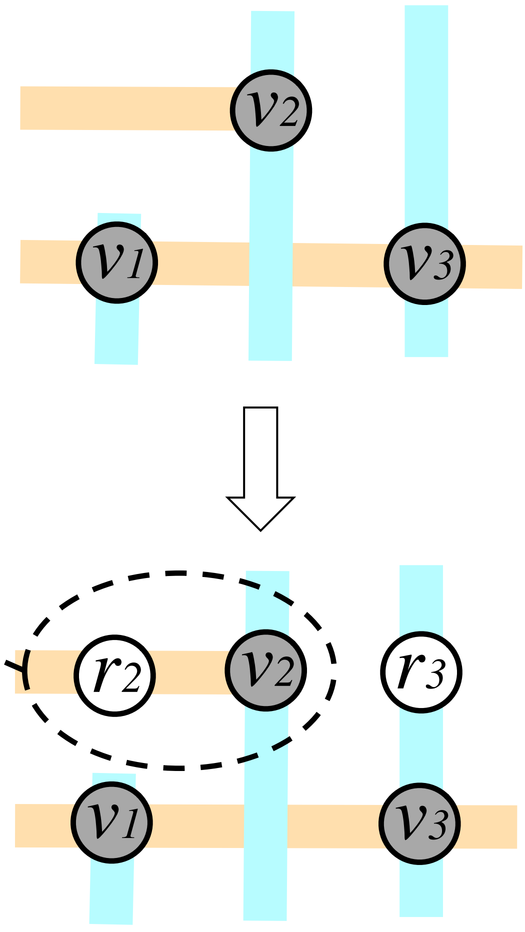


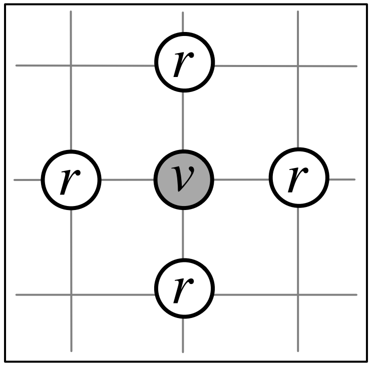


Redundant Via Insertion (RVI)

- Insert an extra via near a single via.
- Prevent via failure, improve circuit yield and reliability.

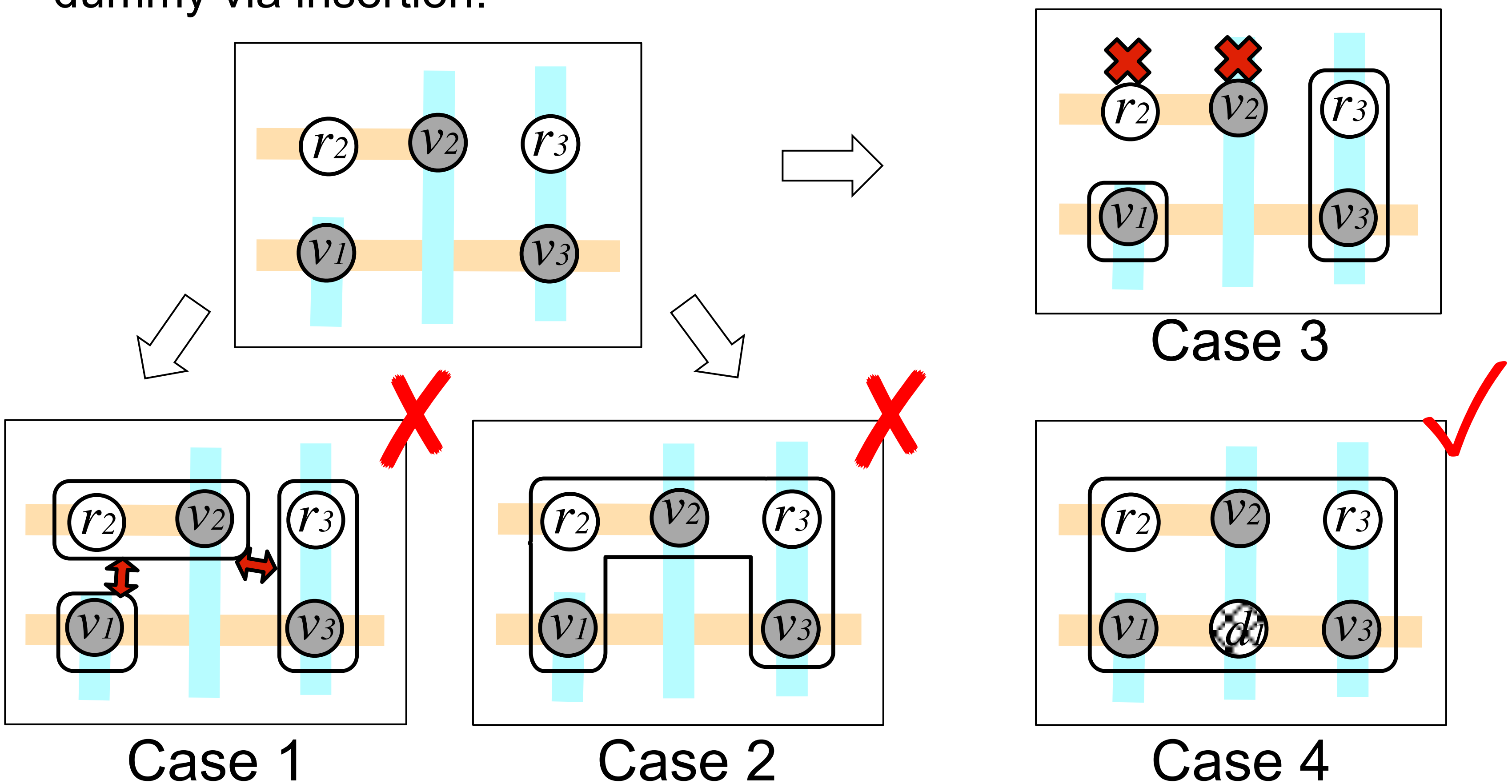






Dummy Via

- Due to the characteristic of DSA, vias in a group must match some specific patterns so that they can be assigned to the same guiding template.
- Increase the choices to form guiding templates with the help of dummy via insertion.

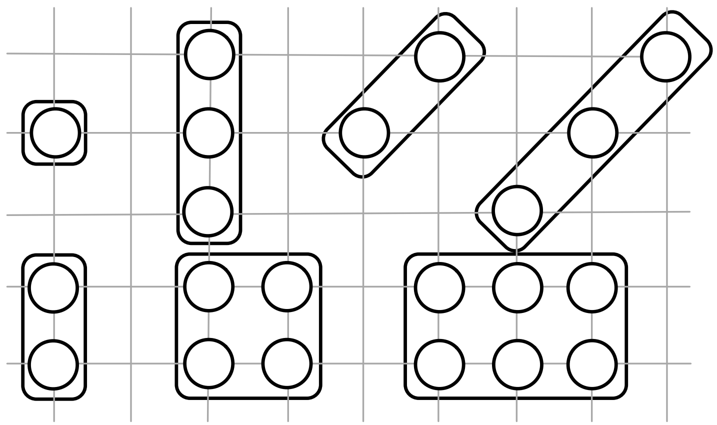


Outline

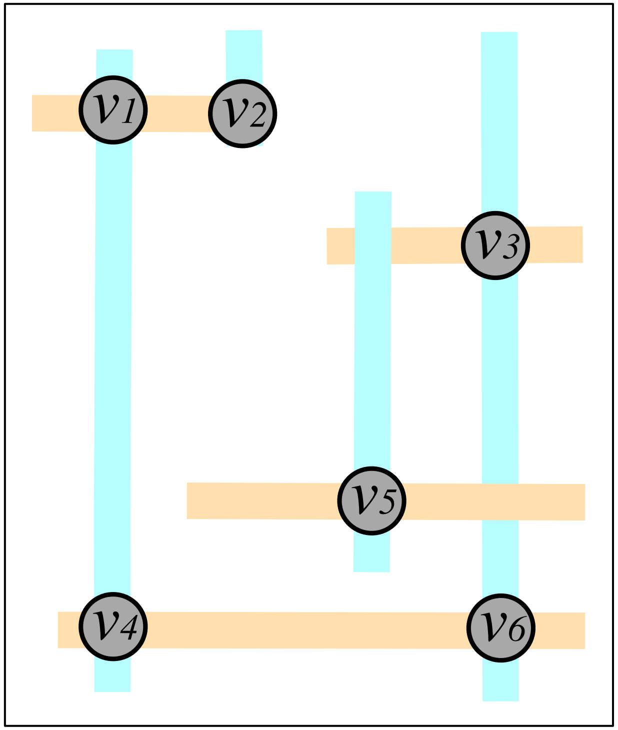
- Introductions
- Problem Formulation
- Algorithm
- Experimental Results
- Conclusions

DSA Guiding Template Assignment with Redundant/Dummy Via Insertion (DRDV)

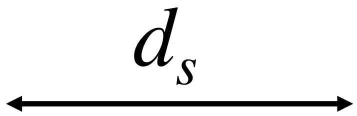
- Input
  - Post-routing layout
  - Usable DSA guiding templates
  - Optical resolution limit space



Usable DSA guiding templates



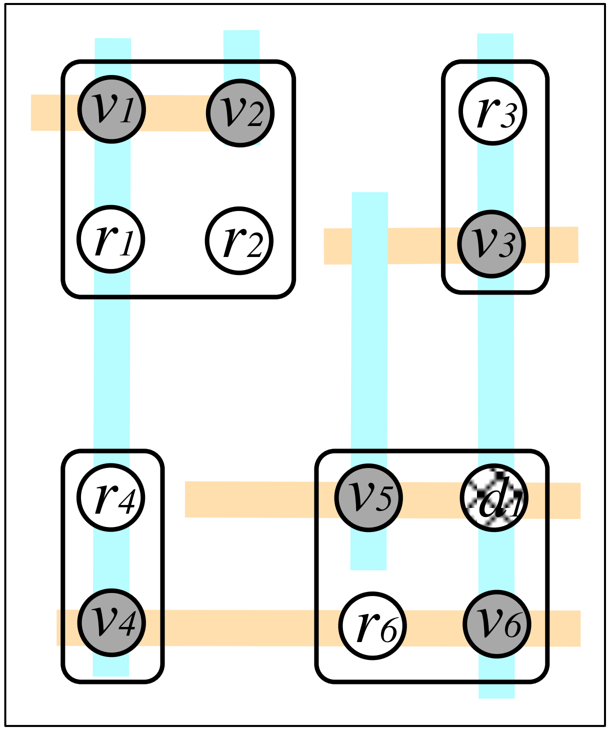
Post-routing layout



Optical resolution limit space

DSA Guiding Template Assignment with Redundant/Dummy Via Insertion (DRDV)

- Input
  - Post-routing layout
  - Usable DSA guiding templates
  - Optical resolution limit space
- Output
  - Redundant via insertion for every via
  - Guiding template assignment with suitable dummy vias for every via and redundant via



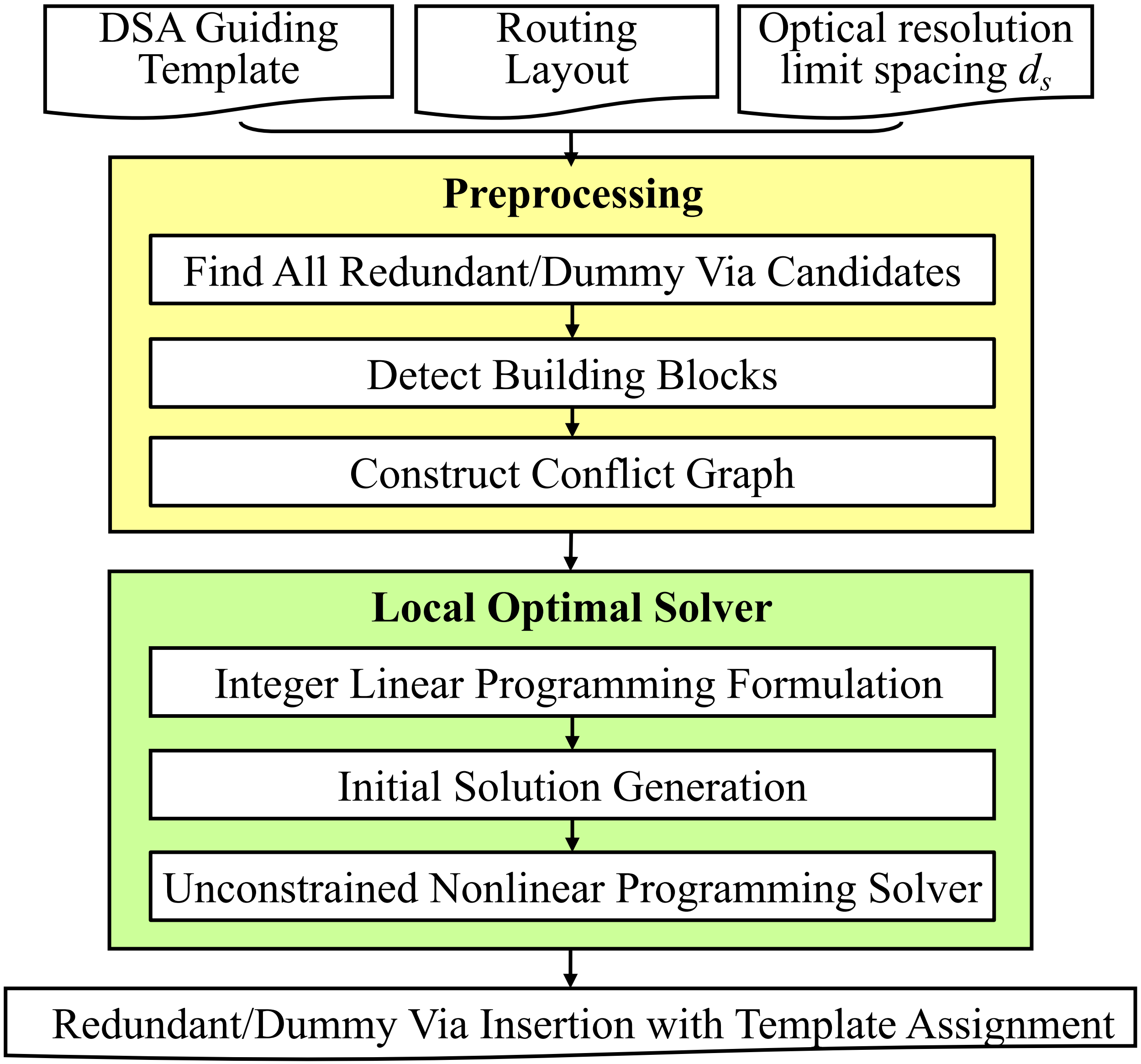
DSA Guiding Template Assignment with Redundant/Dummy Via Insertion (DRDV)

- Input
  - Post-routing layout
  - Usable DSA guiding templates
  - Optical resolution limit space
- Output
  - Redundant via insertion for every via
  - Guiding template assignment with suitable dummy vias for every via and redundant via
- Constraints
  - Inserted redundant vias should be legal
  - The spacing between neighboring guiding template should larger than the optical resolution limit space
- Objectives
  - Maximize the number (ratio) of inserted redundant vias
  - Maximize the number (ratio) of patterned vias by DSA

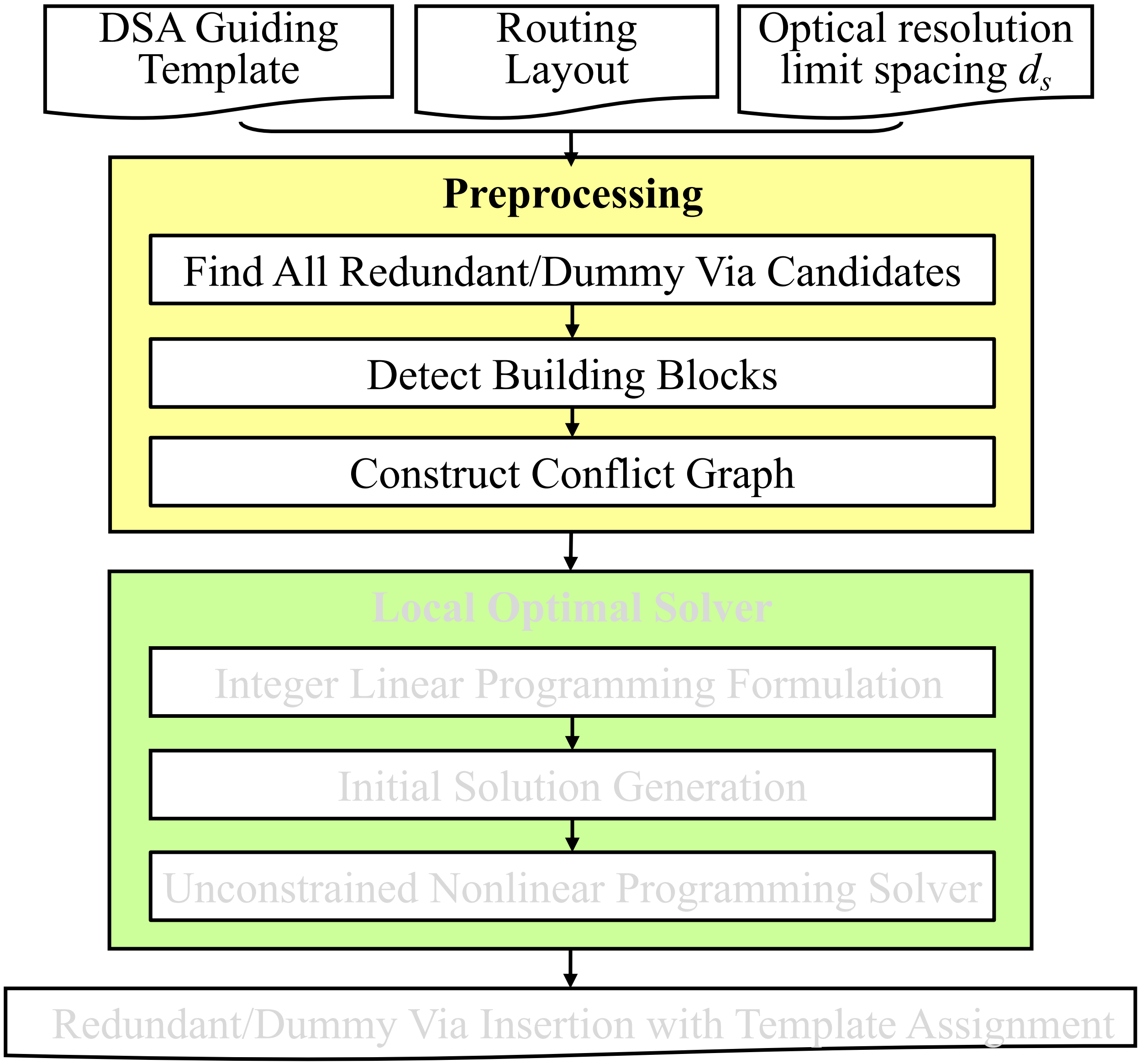
Outline

- Introductions
- Problem Formulation
- Algorithm
- Experimental Results
- Conclusions

Solution Flow

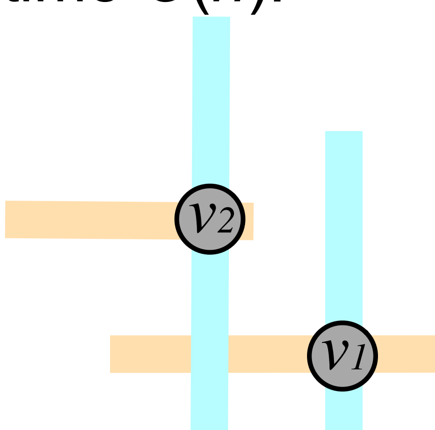


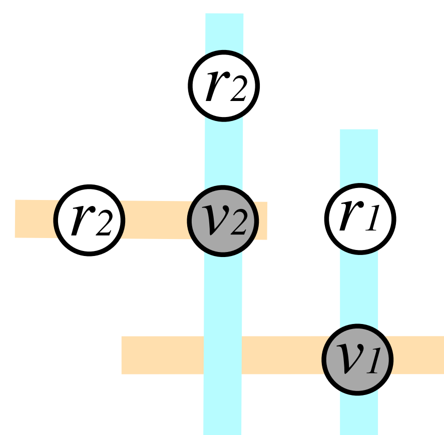
Preprocessing



Redundant/Dummy Via Candidates

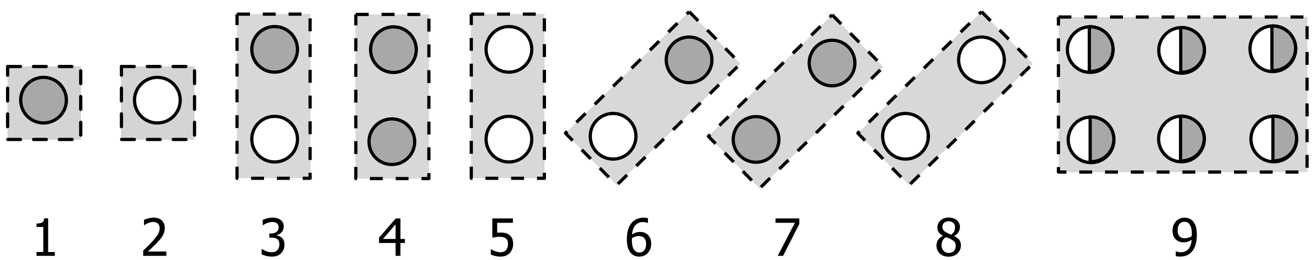
- Redundant via candidate
  - It should be inserted next to every via.
  - It should not overlap with any metal wire from other nets of wires.
- Dummy via candidate
  - It can make up a multi-hole (not less than three holes) guiding template with other vias or redundant vias.
  - It can improve the insertion rate or manufacture rate.
- Find all redundant/dummy via candidates for every via in time $O(n)$.



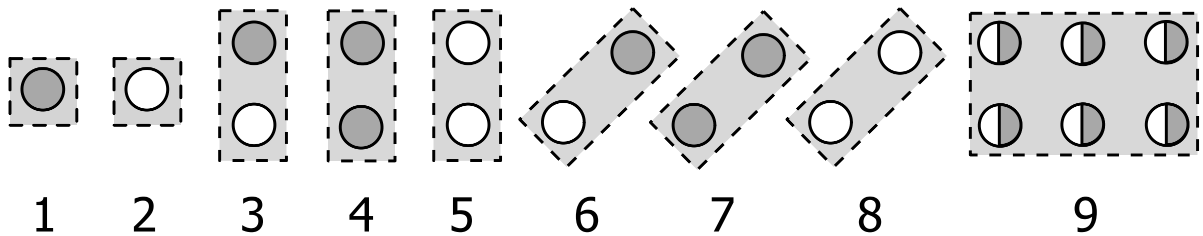


Building-Blocks

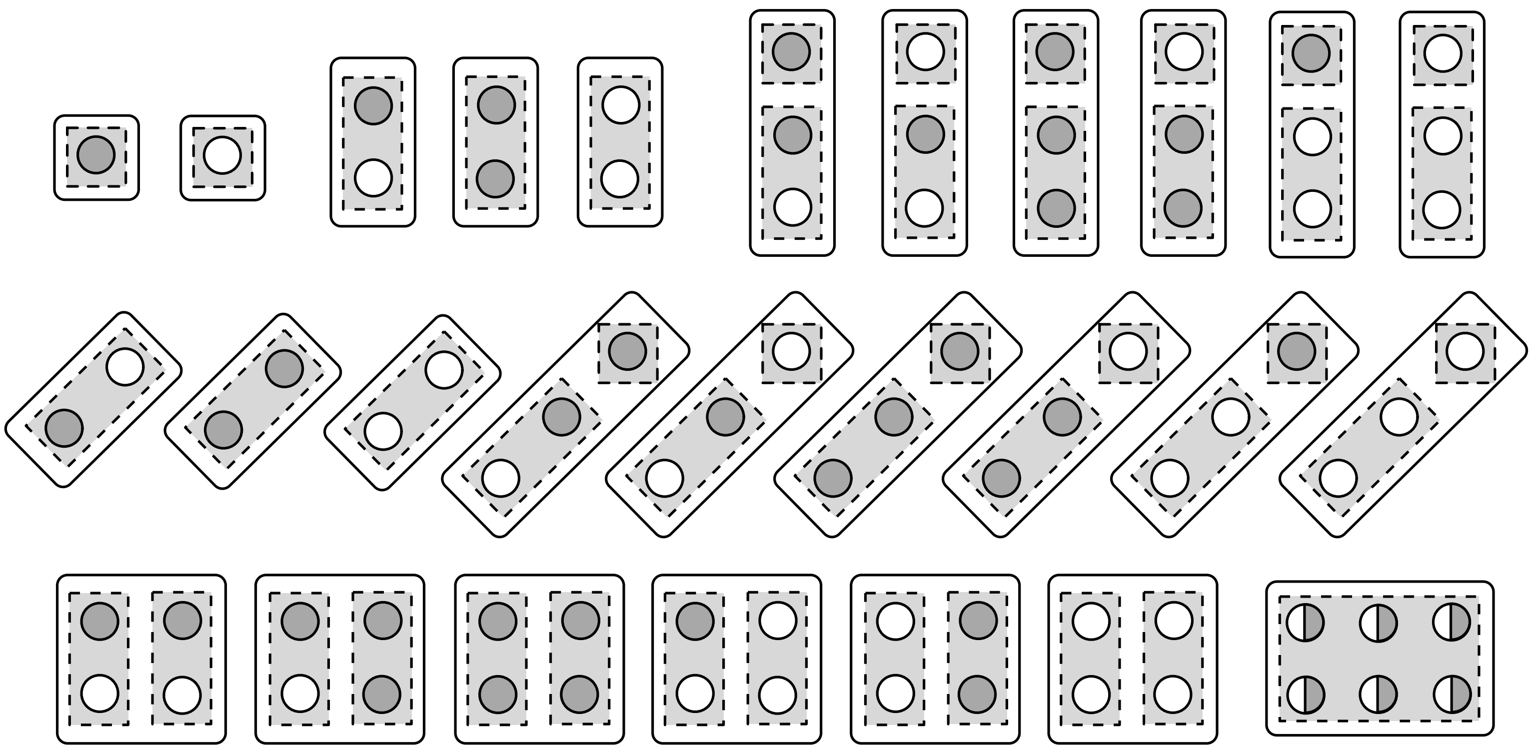
- building-block1: a original via
- building-block2: a redundant via
- building-block3: a original via and a redundant via
- building-block4: two original vias
- building-block5: two redundant vias
- building-block6: a original via and a redundant via (diagonal)
- building-block7: two original vias (diagonal)
- building-block8: two redundant vias (diagonal)
- building-block9: six original/redundant vias



Combinations of Building-Blocks

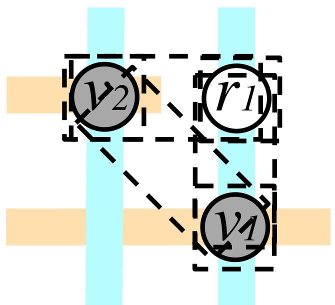


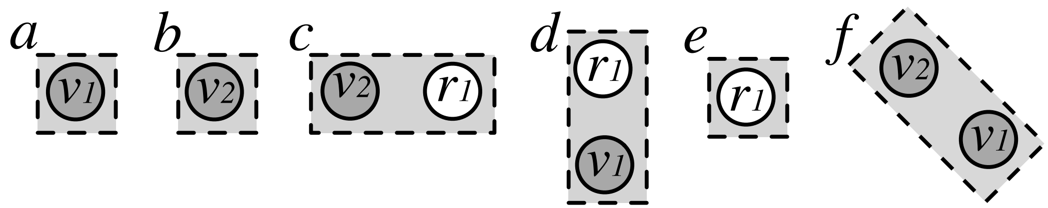
- Combinations of building-blocks to form guiding templates

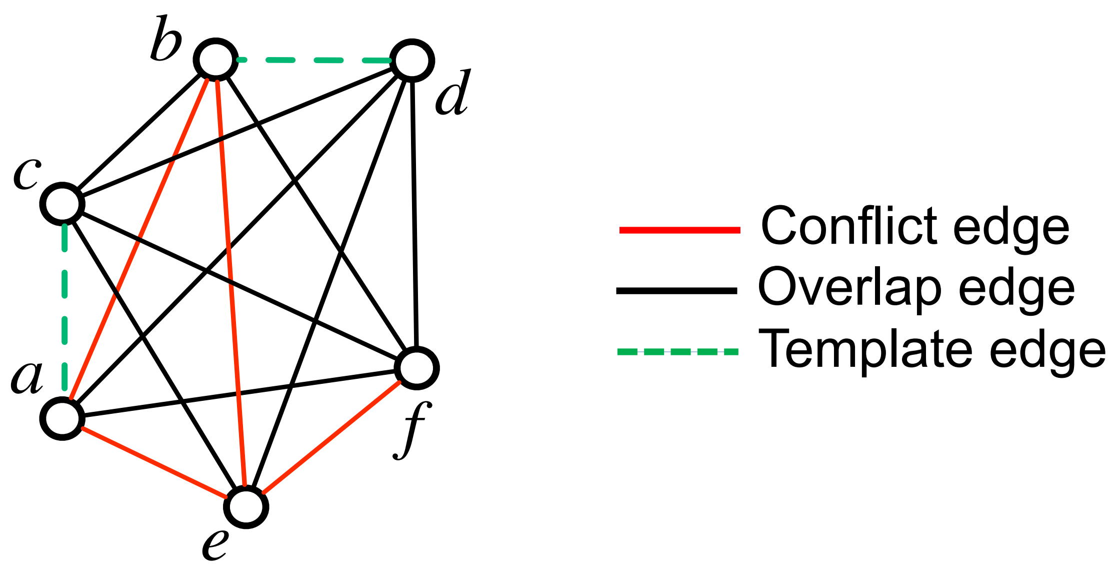


Building-Blocks Detection & Conflict Graph

- Conflict graph *CG (V, E)*
  - —vertex $v \in V$ denotes a *building-block*,
  - —$e_{ij} \in E$ is an edge and $E = (E_C - E_T) \cup E_O$. E_C , E_T and E_O are the sets of conflict edges, template edges and overlap edges.

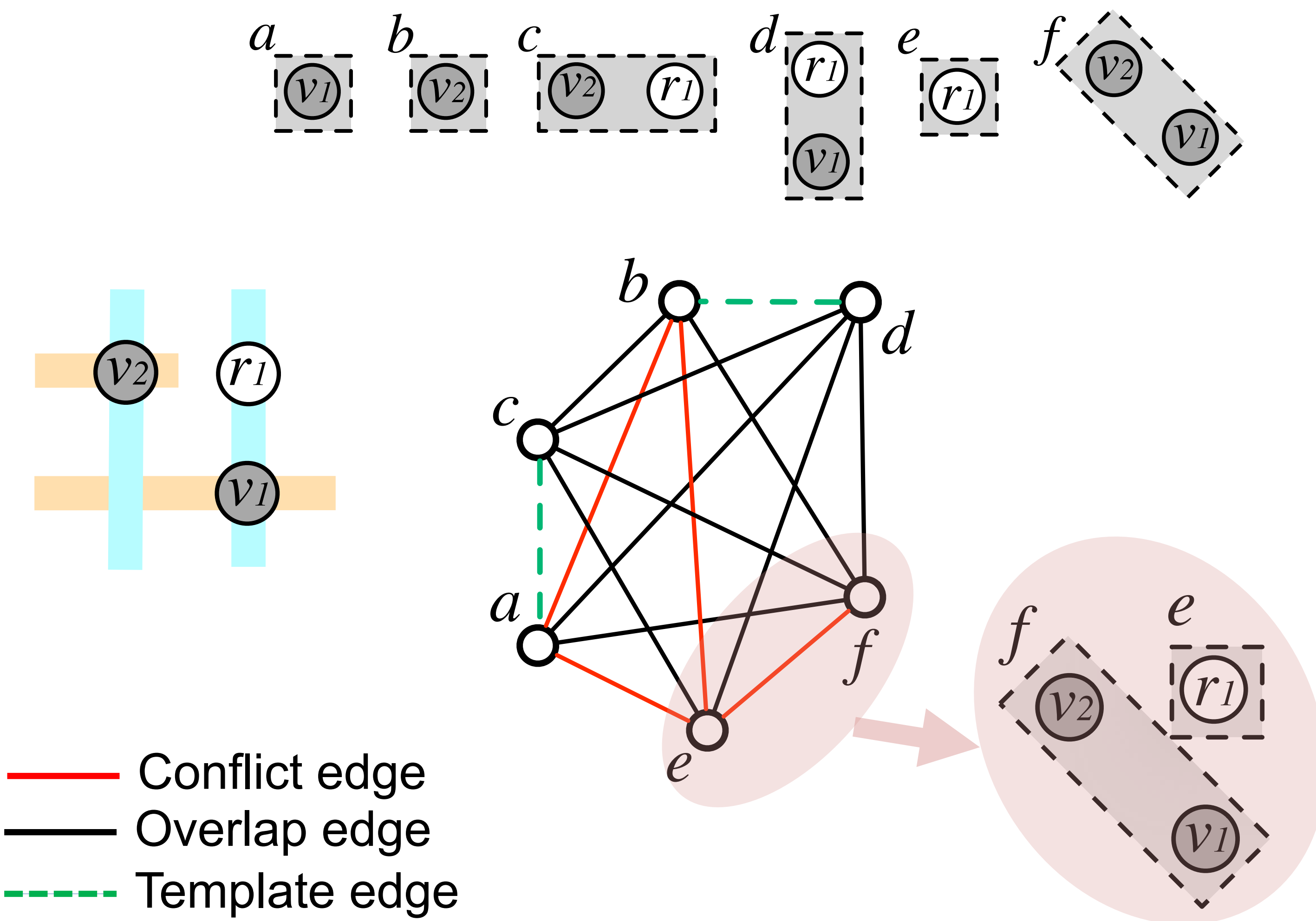






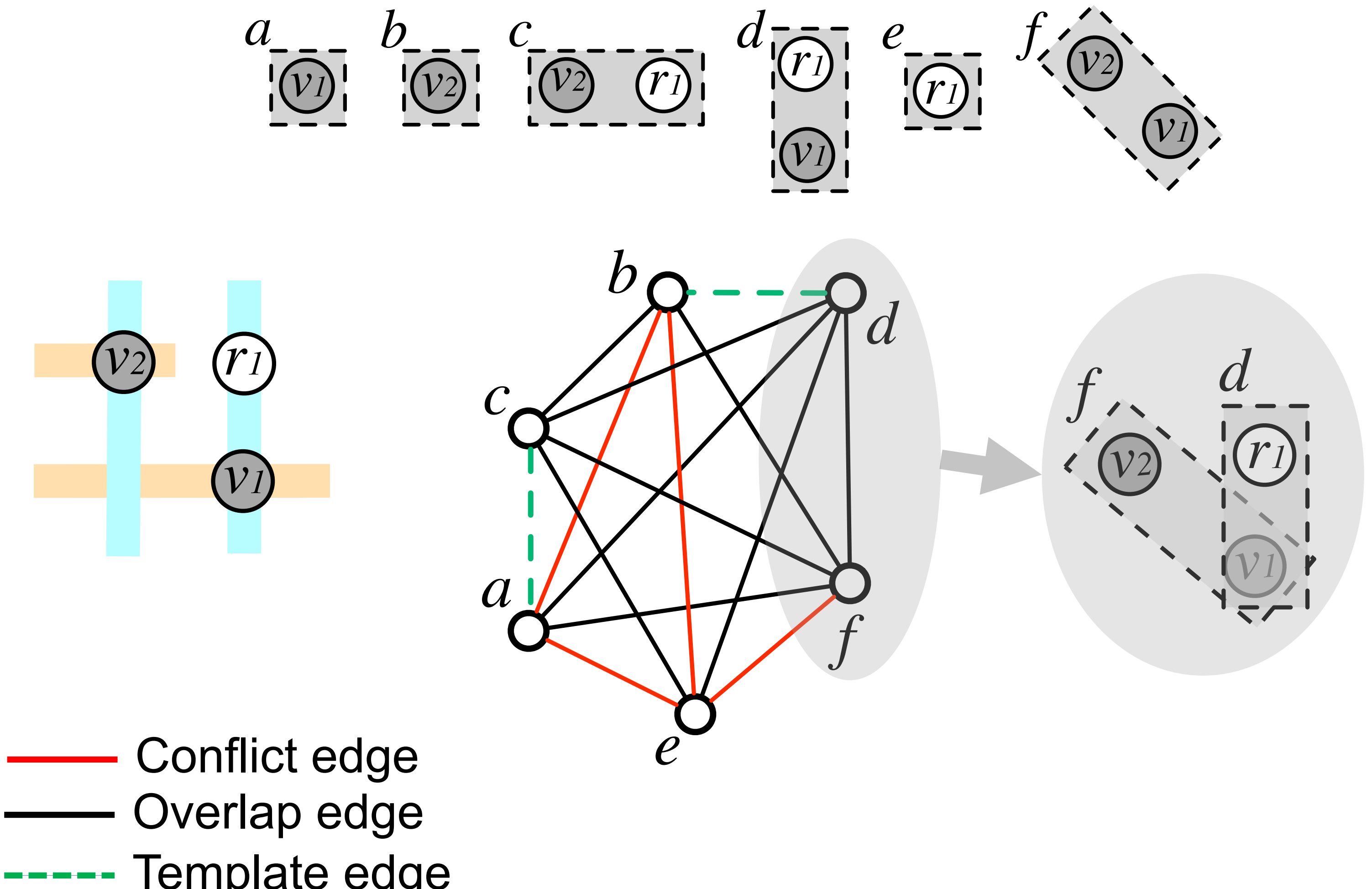
Conflict Edges

- The distance between two building-blocks are less than resolution limit space d_s.



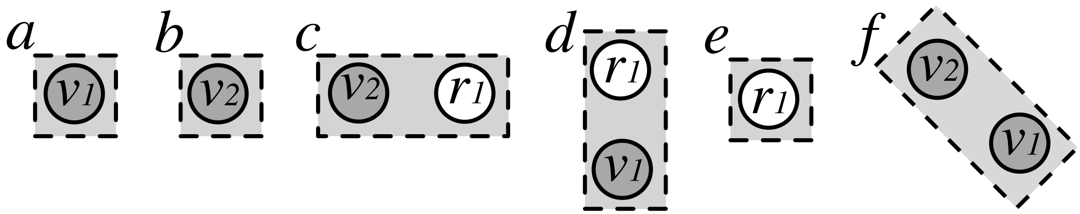
Overlap Edges

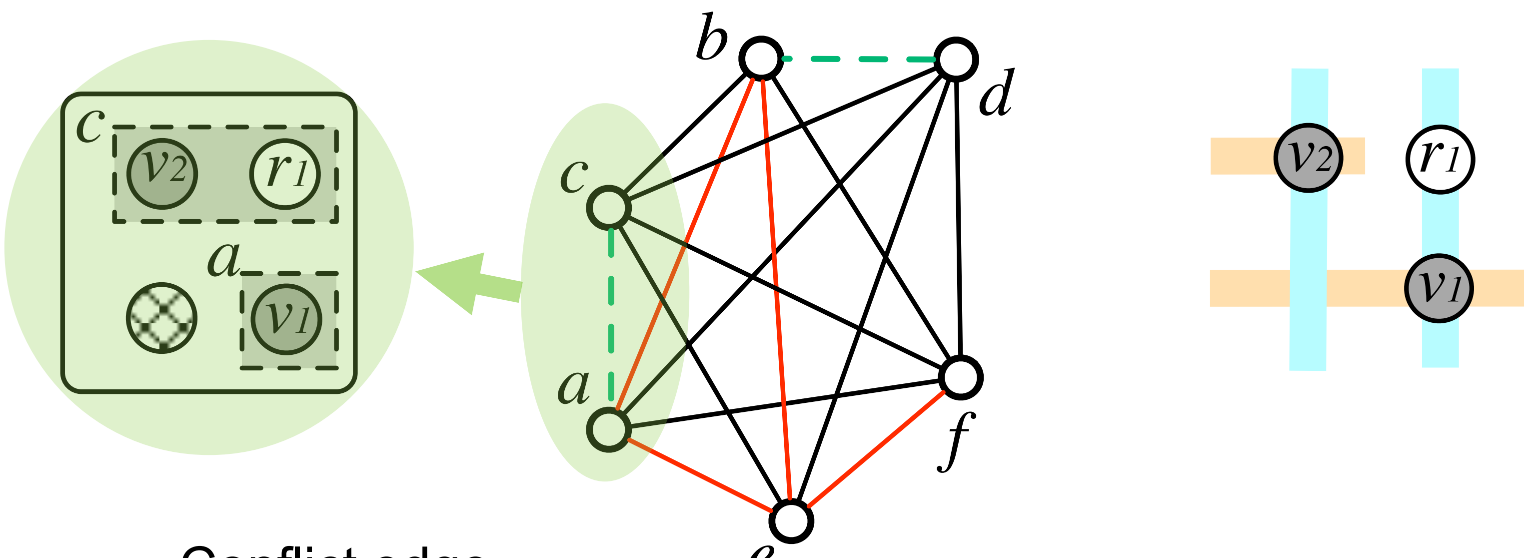
- Two building-blocks are overlapped.



Template Edges

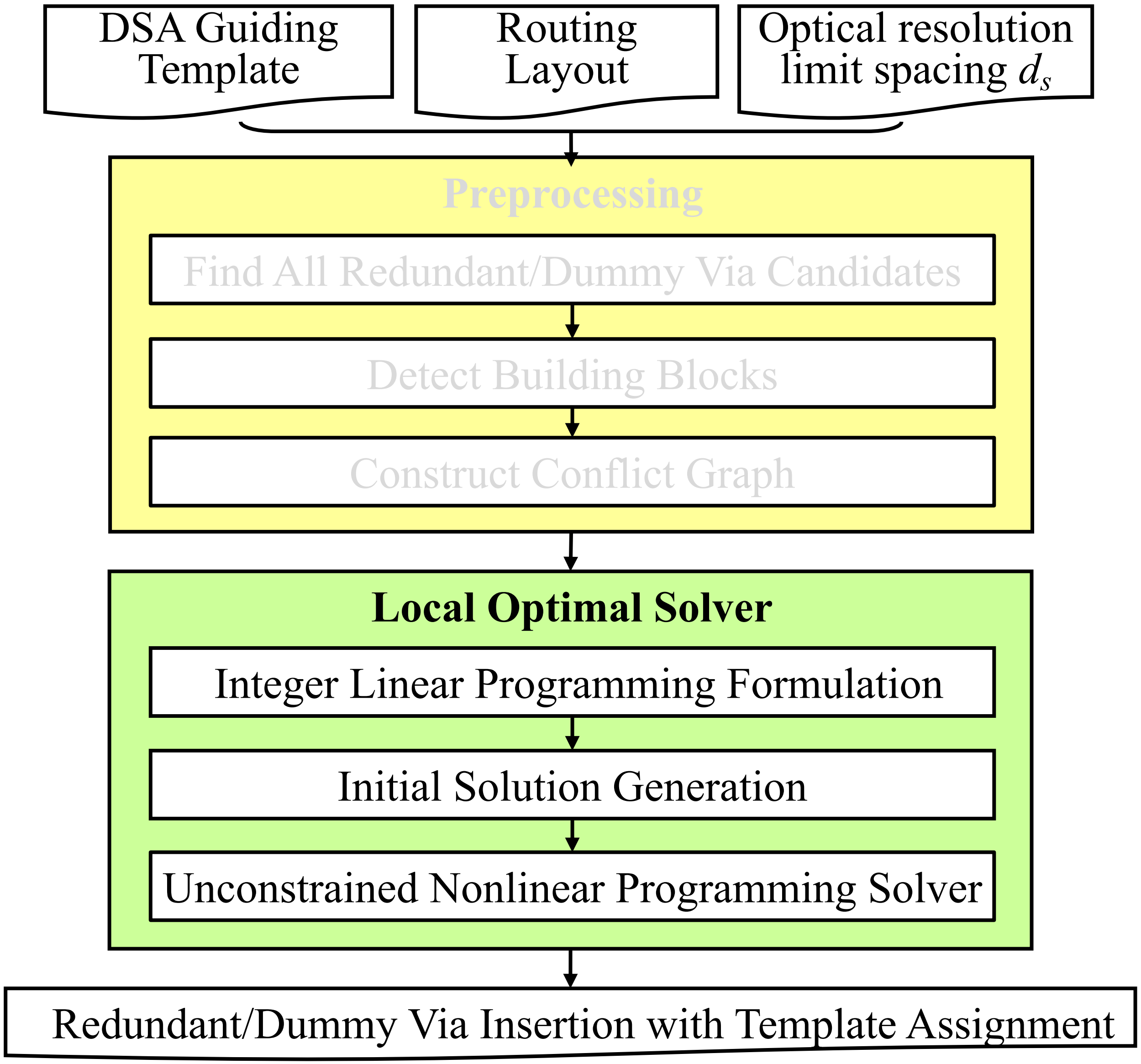
- If building-blocks *i* and *j* with $e_{ij} \in E_C$ can be assigned to a guiding template without any design error.



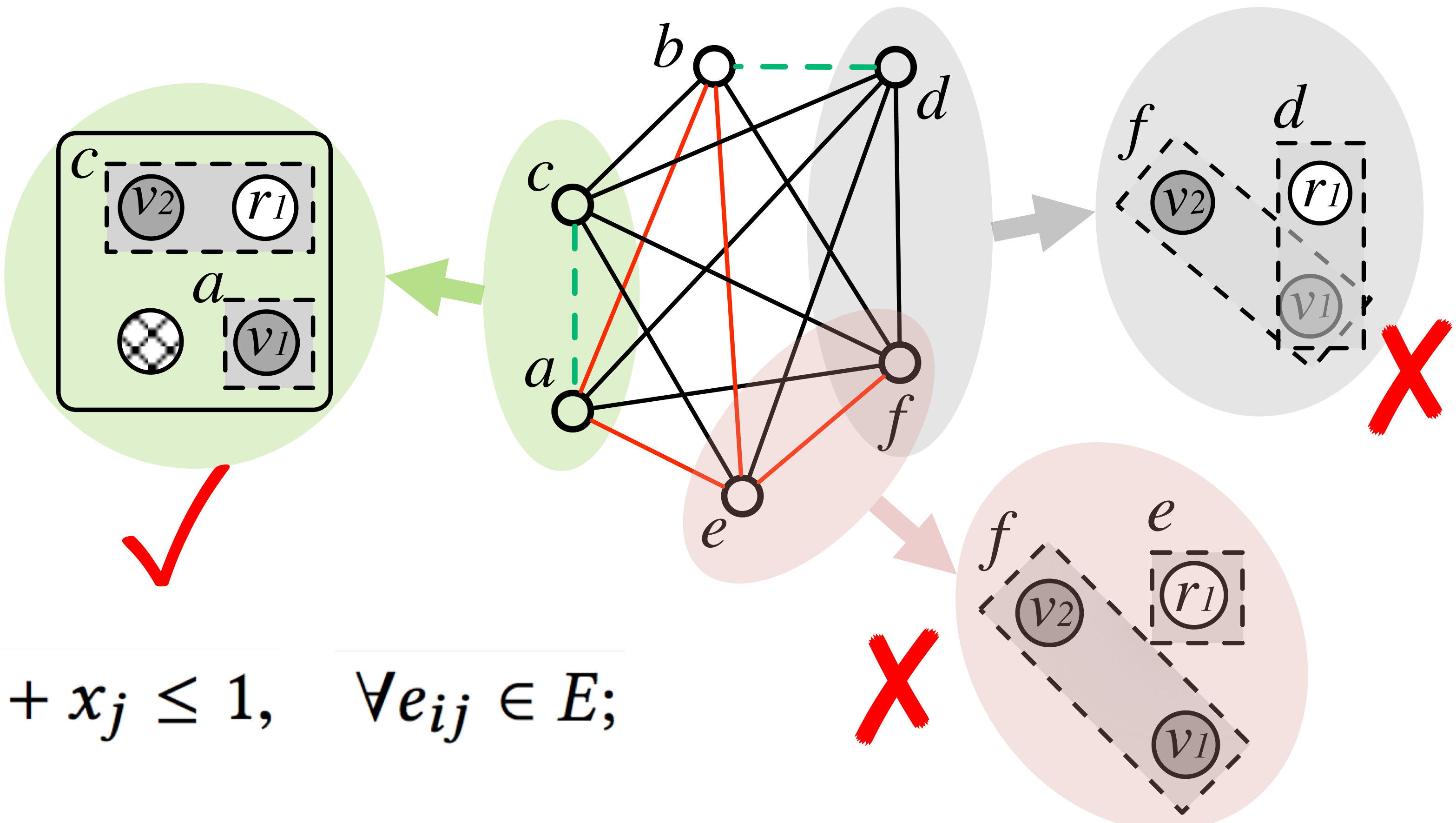


Conflict edge
Overlap edge
Template edge

Solution Flow



Constraints



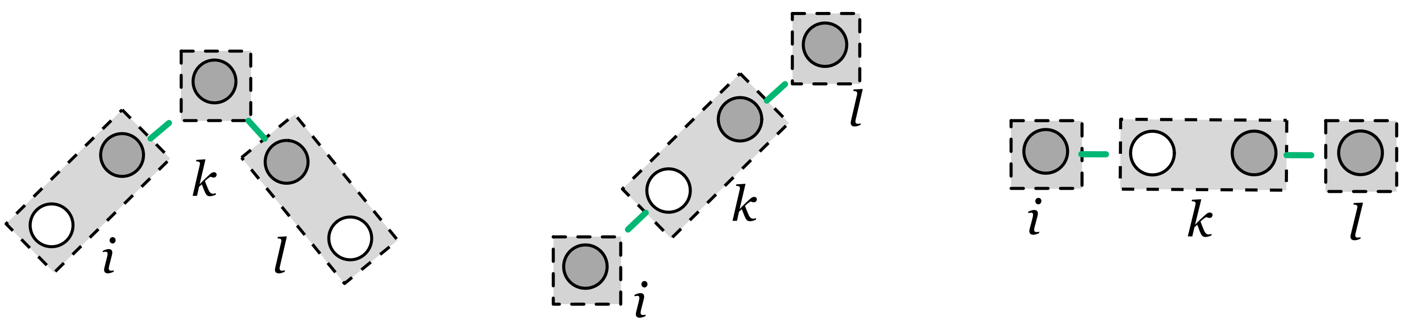
$$x_i + x_j \le 1, \quad \forall e_{ij} \in E;$$

$$E = (E_C - E_T) \cup E_O$$

Conflict edge Template edge Overlap edge

Conflict Structure Constraint

- Template constraint
  - If two building-blocks i and j are connected by a template edge, then they may be assigned to the same guiding template, but not necessarily.
  - If both of building-blocks i and l connect with k by template edges, then i, k, l may not be assigned to a same guiding template.
- Conflict structure (CS)
  - Three bblocks i, k and l, in which e_{ik} and e_{kl} are template edges and there does not exist any edge between *i* and l.



$$x_i + x_k + x_l \leq 2, \qquad \forall(i,k,l) \in CS;$$

Integer Linear Programming (ILP)

- Objectives:
  - Maximize the number of inserted redundant vias
  - Maximize the number of patterned vias by DSA
  - Let N_v and N_r are the numbers of included vias and redundant vias by building-block i, and

$$w_i = N_v + \beta \cdot N_r,$$

- ILP Formulation

$$\begin{aligned}
\max_{\mathbf{x}} \quad & \sum_{i \in V} w_i x_i & & \textbf{(1)} \\
\text{s.t.} \quad & x_i + x_j \leq 1, & & \forall e_{ij} \in E; \\
& x_i + x_k + x_l \leq 2, & & \forall (i,k,l) \in CS; \\
& x_i \in \{0,1\}, & & \forall i \in V.
\end{aligned}$$

Inequality Constraints

- **Claim 1.** The ILP is equivalent to the DRDV problem.

$$
\begin{aligned}
\max_{\mathbf{x}} \quad & \sum_{i \in V} w_i x_i && && \textbf{(1)}\\
\text{s.t.} \quad & x_i + x_j \le 1, && \forall e_{ij} \in E; \\
& x_i + x_k + x_l \le 2, && \forall (i,k,l) \in CS; \\
& x_i \in \{0,1\}, && \forall i \in V.
\end{aligned}
$$

- Transfer inequality constraints to equality constraints.

$$
\begin{aligned}
\max_{\mathbf{x}} \quad & \sum_{i \in V} w_i x_i && && \textbf{(2)}\\
\text{s.t.} \quad & x_i x_j = 0, && \forall e_{ij} \in E; \\
& x_i x_k x_l = 0, && \forall (i,k,l) \in CS; \\
& x_i \in \{0,1\}, && \forall i \in V,
\end{aligned}
$$

Equality Constraints

- Relax equality constraints to objective function.

$$
\begin{aligned}
\max_{\mathbf{x}} \quad & \sum_{i \in V} w_i x_i & & \quad (2)\\
\text{s.t.} \quad & x_i x_j = 0, & & \forall e_{ij} \in E;\\
& x_i x_k x_l = 0, & & \forall (i,k,l) \in CS;\\
& x_i \in \{0,1\}, & & \forall i \in V,
\end{aligned}
$$

$$
\begin{aligned}
\max_{\mathbf{x}} \quad & \sum_{i \in V} \{ w_i x_i \boxed{\prod_{\substack{j \in V \\ e_{ij} \in E}} (1 - x_j)} \boxed{\prod_{\substack{k,l \in V \\ (i,k,l) \in CS}} (1 - x_k x_l)} \} \quad (3)\\
\text{s.t.} \quad & x_i \in \{0,1\}, \ \forall i \in V.
\end{aligned}
$$

Adjacent Matrix & CS Tensor

- Handle adjacent matrix and CS tensor.

$$\max_{\mathbf{x}} \quad \sum_{i\in V}\{w_i x_i \prod_{\substack{j\in V \\ \boxed{e_{ij}\in E}}} (1-x_j) \prod_{\substack{k,l\in V \\ \boxed{(i,k,l)\in CS}}} (1-x_k x_l)\} \qquad \textbf{(3)}$$

$$\text{s.t.} \quad x_i \in \{0,1\},\ \forall i \in V.$$

$$\max_{\mathbf{x}} \quad \sum_{i\in V}\{w_i x_i \prod_{j\in V} (1-x_j)^{\boxed{B_{ij}}} \prod_{k,l\in V} (1-x_k x_l)^{\boxed{C_{ikl}}}\} \qquad \textbf{(4)}$$

$$\text{s.t.} \quad x_i \in \{0,1\},\ \forall i \in V.$$

$$B_{ij} = \begin{cases} 1, & e_{ij} \in E \\ 0, & e_{ij} \notin E \end{cases}$$

$$C_{ikl} = \begin{cases} 1, & (i,k,l) \in CS \\ 0, & (i,k,l) \notin CS \end{cases}$$

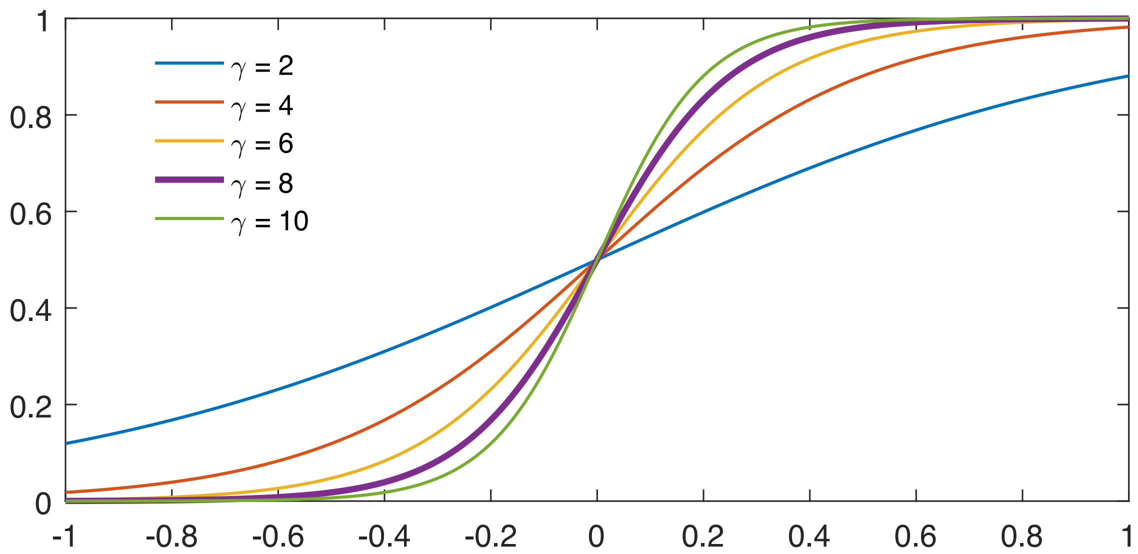
Unconstrained Nonlinear Programming (UNP)

$$\max_{\mathbf{x}} \quad \sum_{i \in V} \{ w_i x_i \prod_{j \in V} (1 - x_j)^{B_{ij}} \prod_{k,l \in V} (1 - x_k x_l)^{C_{ikl}} \} \tag{4}$$

$$\text{s.t.} \quad x_i \in \{0, 1\}, \ \forall i \in V.$$

$$\boxed{x_i \approx \sigma(y_i) = (1 + e^{-\gamma y_i})^{-1}}$$

$$\boxed{x_i = \begin{cases} 1, & y_i \geq 0 \\ 0, & y_i < 0 \end{cases}}$$



$$\max_{\mathbf{y}} \ f(\mathbf{y}) = \sum_{i \in V} \{ w_i \sigma(y_i) \prod_{j \in V} (1 - \sigma(y_j))^{B_{ij}} \prod_{k,l \in V} (1 - \sigma(y_k)\sigma(y_l))^{C_{ikl}} \} \tag{5}$$

UNP Solver

$$\max_{\mathbf{y}} \; f(\mathbf{y}) = \sum_{i \in V} \{ w_i \sigma(y_i) \prod_{j \in V} (1 - \sigma(y_j))^{B_{ij}} \prod_{k,l \in V} (1 - \sigma(y_k)\sigma(y_l))^{C_{ikl}} \} \quad \text{(5)}$$

Input: A connected component of $CG(V, E, W)$, convergence threshold $\delta = 10^{-4}$;

Output: Solution $\mathbf{x}^*$ of ILP (2);

Initialize $t \leftarrow 0$;

Generate $\mathbf{x}^{(0)}$;

If $x_i^{(0)} = 1$, let $y_i^{(0)} \leftarrow 1$; otherwise, let $y_i^{(0)} \leftarrow$ -1;

repeat

  $\forall i \in V$, compute $g_i^{(t)}$;

  Obtain $\nabla f(\mathbf{y}^{(t)})$;

  $\alpha \leftarrow \texttt{LineSearch}(\mathbf{y}^{(t)})$;

  $\mathbf{y}^{(t+1)} \leftarrow \mathbf{y}^{(t)} + \alpha \nabla f(\mathbf{y}^{(t)})$;

  $t \leftarrow t + 1$;

until $||\nabla f(\mathbf{y}^{(t)})|| < \delta$

Get x_i^* by rounding $\sigma(y_i^{(t)})$ to the nearest integer, $\forall i \in V$.

Greedy selection method

$$[\nabla f(\mathbf{y}^{(t)})]_i = \partial f(\mathbf{y}^{(t)}) / \partial y_i$$

Wolfe-Powell inexact line search method

$$\boxed{\mathcal{O}(\max\{|V| \cdot |E|, |V| \cdot ||\mathbf{C}||_0\})}$$

Local Optimal Convergence

$$g_i(\mathbf{y}) = \sigma(y_i) \prod_{j \in V} (1 - \sigma(y_j))^{B_{ij}} \prod_{k,l \in V} (1 - \sigma(y_k)\sigma(y_l))^{C_{ikl}}$$

$$[\nabla f(\mathbf{y}^{(t)})]_i = \partial f(\mathbf{y}^{(t)}) / \partial y_i$$

$$= \gamma w_i g_i^{(t)} \{(1 - \sigma(y_i^{(t)})) - \sum_j B_{ij} \sigma(y_j^{(t)}) - \sum_k \sum_l C_{ikl} \frac{\sigma(y_k^{(t)})(1 - \sigma(y_k^{(t)}))\sigma(y_l^{(t)})}{1 - \sigma(y_k^{(t)})\sigma(y_l^{(t)})}\}$$

- **Lemma 1.** Under above Equations, $\sum_i w_i \Delta g_i \geq 0$.
- **Theorem 1.** Under above Equations, f(y) does not decrease.
- **Corollary 1.** Strict inequality $\sum_i w_i \Delta g_i > 0$ cannot be achieved.
- **Theorem 2.** Our UNP solver converges to a local maximum.

Outline

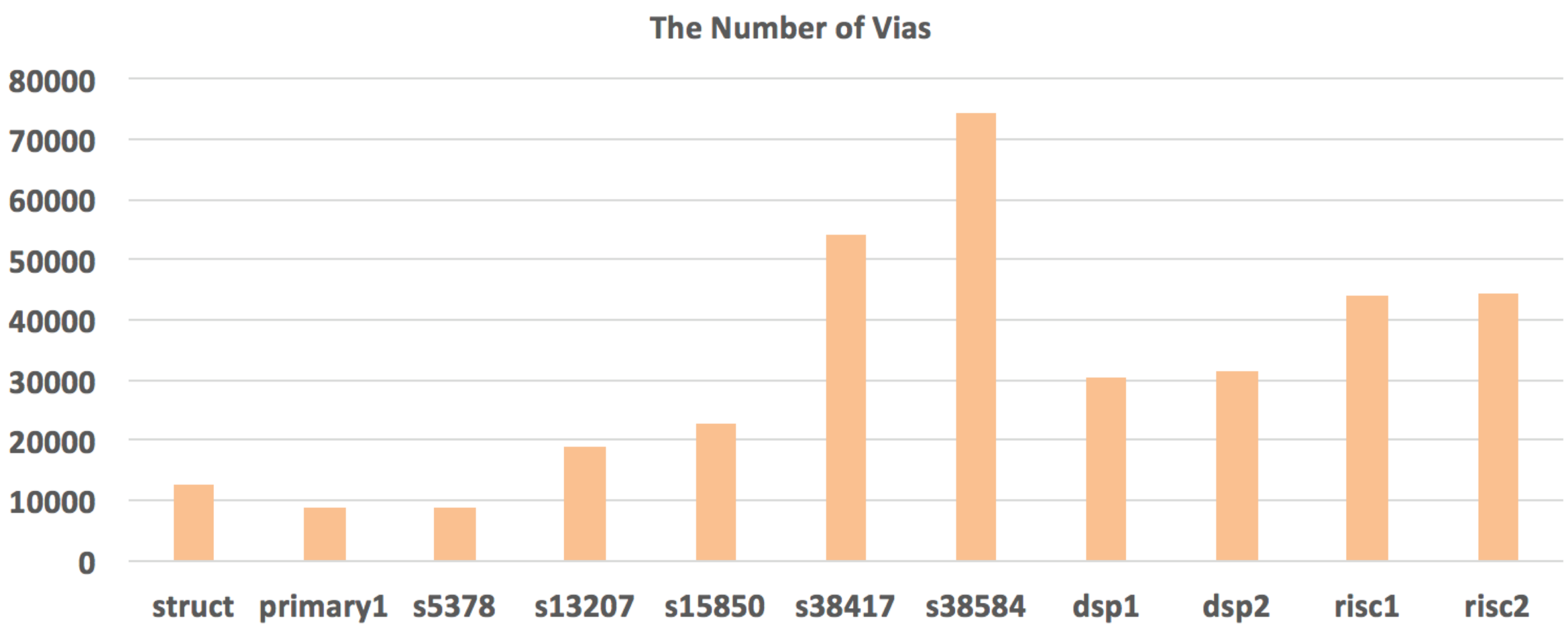
- Introductions
- Problem Formulation
- Algorithm
- Experimental Results
- Conclusions

Experimental Settings

- Platform
  - C++ programming language
  - Unix machine with Intel Core 2.70 GHz CPU and 8 GB memory
  - ILP solver: CPLEX
- Benchmarks
  - 11 circuits are provided by Prof. Fang, modified from MCNC benchmarks and an industry Faraday benchmarks
- Algorithms
  - TCAD’17: DSA+RVI, ILP+Speed-up
  - ASPDAC’17: DSA+RVI+DVI, ILP+Speed-up
  - TVLSI’18: DSA+RVI+DVI, Two Stage MWIS Solver
  - Ours: DSA+RVI+DVI, UNP Solver
- Indicators
  - Manufacture rate, insertion rate, runtime

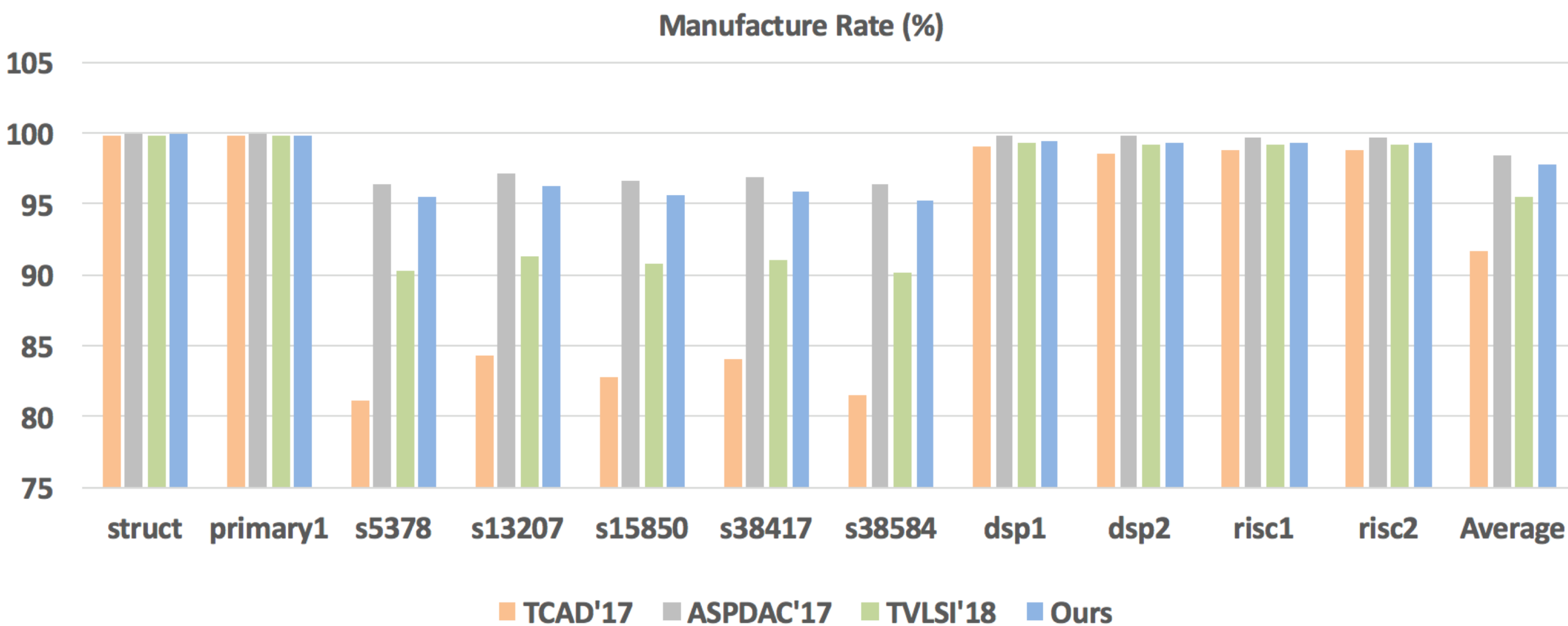
The Number of Vias

- The numbers of vias of benchmarks range from eight thousand to seventy thousand.



Comparison: Manufacture Rate

- Compared with "TCAD'17," "ASPDAC'17," and "TVLSI'18," our algorithm achieves **6%**, **0%**, and **3%** improvement on manufacture rate.



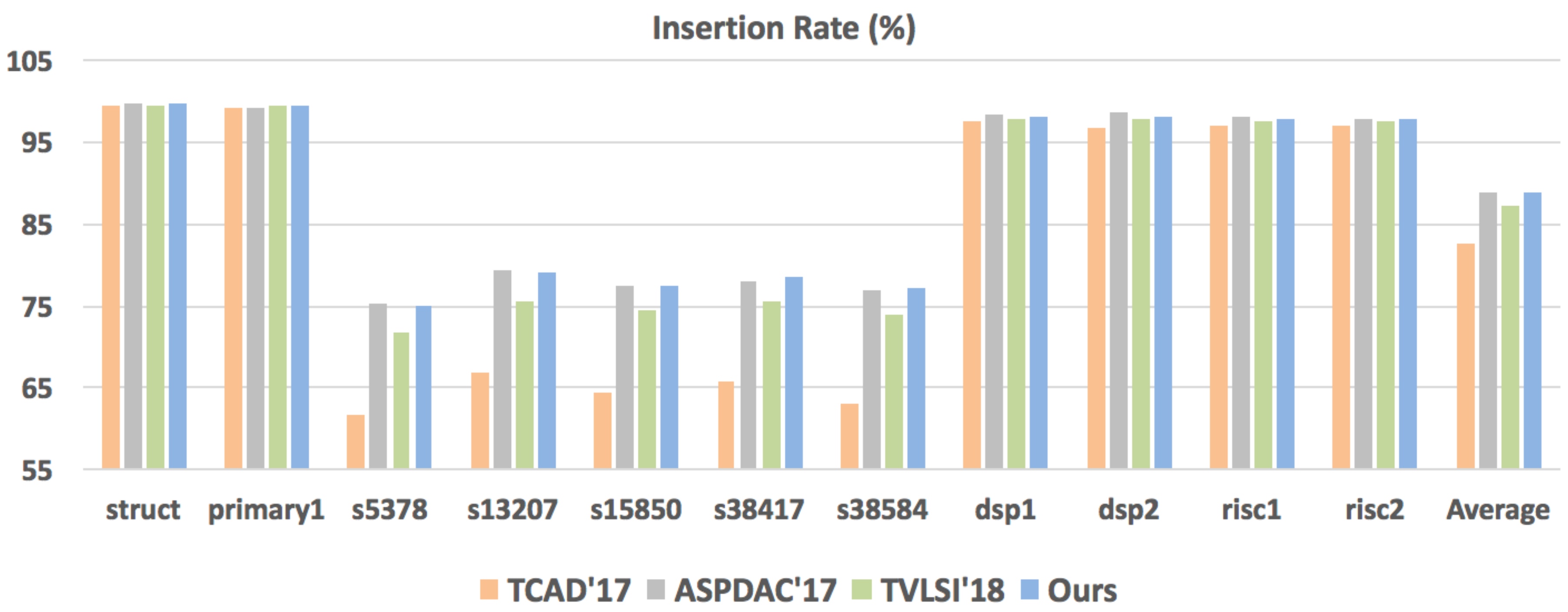
[TCAD'17] S.-Y. Fang, Y.-X. Hong, and Y.-Z. Lu, "Simultaneous guiding template optimization and redundant via insertion for directed self-assembly," *IEEE TCAD*, 2017.

[ASPDAC'17] C.-Y. Hung, P.-Y. Chou, and W.-K. Mak, "Optimizing DSA-MP decomposition and redundant via insertion with dummy vias", In *Proc. of ASPDAC*, 2017.

[TVLSI'18] X. Li, B. Yu, J. Ou, J. Chen, D. Z. Pan and W. Zhu, "Graph based redundant via insertion and guiding template assignment for DSA-MP", *IEEE TVLSI*, 2018.

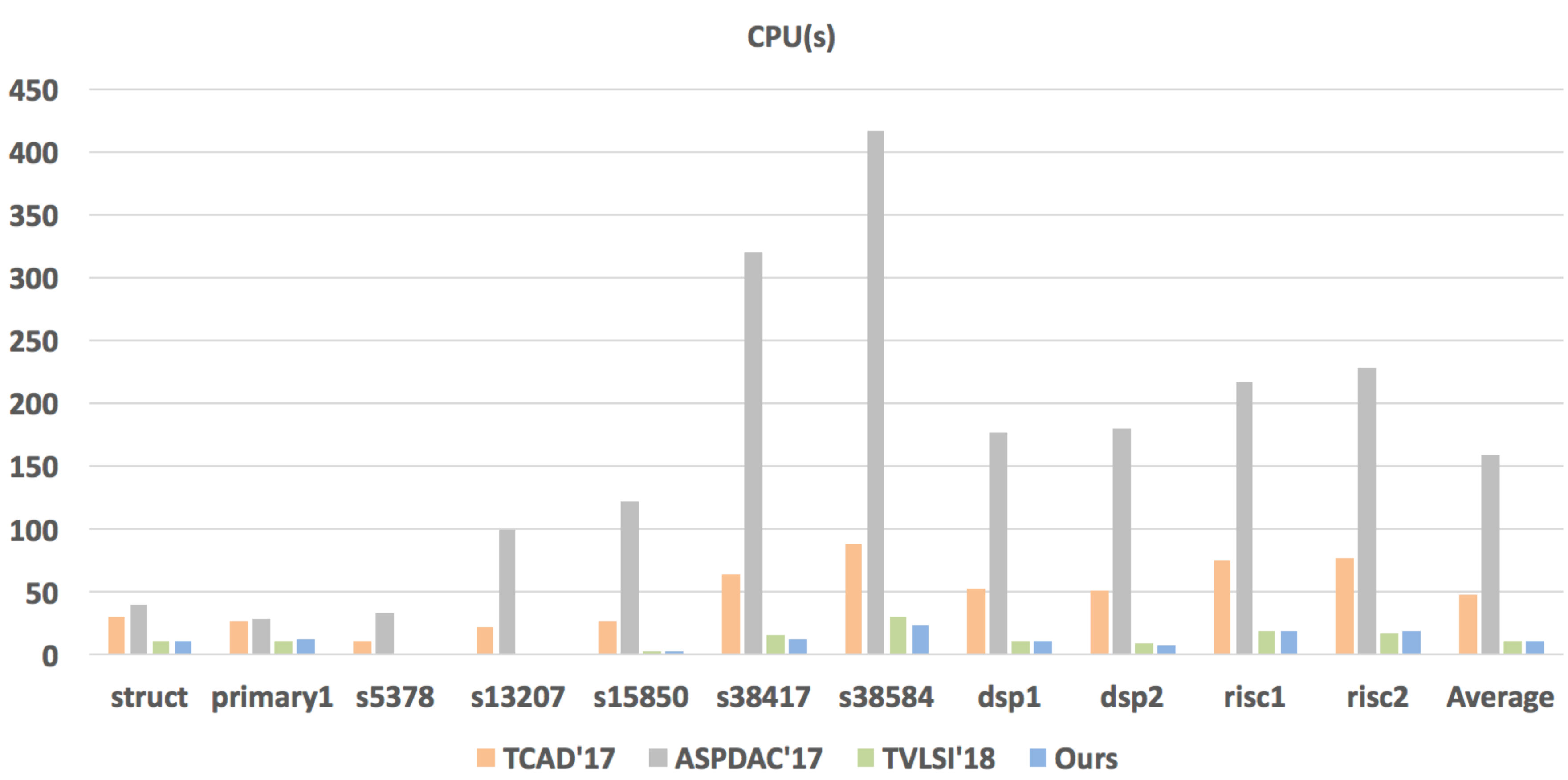
Comparison: Insertion Rate

- Compared with "TCAD'17," "ASPDAC'17," and "TVLSI'18," our algorithm achieves **7%**, **0%**, and **2%** improvement on insertion rate.



Comparison: Runtime

- Our algorithm is **3.99X and 13.32X** faster than "TCAD'17", "ASPDAC'17".



Outline

Conclusions

- We introduce a building-block based manner instead of guiding template candidate to express solution.
- We proposed a general ILP formulation and relaxed it to an UNP. Furthermore, we develop a first-order optimization method to solve the UNP, which is a local optimal algorithm.
- Experimental results verify our algorithm achieves comparable experimental results with a state-of-the-art work, and saves 92% runtime.

Handle Guiding Template Cost

- A building-block would be assigned to a guiding template.
- A guiding template composed of one or two building-blocks.
- Assign proper weights to building-block w_i ($\forall\, i \in V$) and corresponding template edge $\widetilde{w}_{ij}$ ($\forall\, e_{ij} \in E_T$).

$$
\begin{aligned}
\max_{\mathbf{x}} \quad & \sum_{i \in V} w_i x_i + \frac{\lambda}{2} \sum_{e_{ij} \in E_T} \widetilde{w}_{ij} x_i x_j && \text{(1')} \\
\text{s.t.} \quad & x_i + x_j \le 1, && \forall e_{ij} \in E; \\
& x_i + x_k + x_l \le 2, && \forall (i,k,l) \in CS; \\
& x_i \in \{0, 1\}, && \forall i \in V.
\end{aligned}
$$

Handle Guiding Template Cost

$$\max_{\mathbf{x}} \quad \sum_{i \in V} w_i x_i \{1 + \frac{\lambda}{2w_i} \sum_{\substack{j \in V \\ e_{ij} \in E_T}} \widetilde{w}_{ij} x_j\} \qquad \textbf{(2')}$$

$$\begin{aligned}
\text{s.t.} \quad & x_i x_j = 0, && \forall e_{ij} \in E; \\
& x_i x_k x_l = 0, && \forall (i,k,l) \in CS; \\
& x_i \in \{0,1\}, && \forall i \in V,
\end{aligned}$$

$$\max_{\mathbf{x}} \quad \sum_{i \in V} w_i x_i \{1 + \frac{\lambda}{2w_i} \sum_{\substack{j \in V \\ e_{ij} \in E_T}} \widetilde{w}_{ij} x_j\} \prod_{\substack{j \in V \\ e_{ij} \in E}} (1 - x_j) \prod_{\substack{k,l \in V \\ (i,k,l) \in CS}} (1 - x_k x_l)\} \qquad \textbf{(3')}$$

$$\text{s.t.} \quad x_i \in \{0,1\}, \ \forall i \in V.$$

Thank You!