# Multiple Patterning Layout Compliance with Minimizing Topology Disturbance and Polygon Displacement


Hua-Yu Chang and Iris Hui-Ru Jiang

SYNOPSYS®
Silicon to Software™

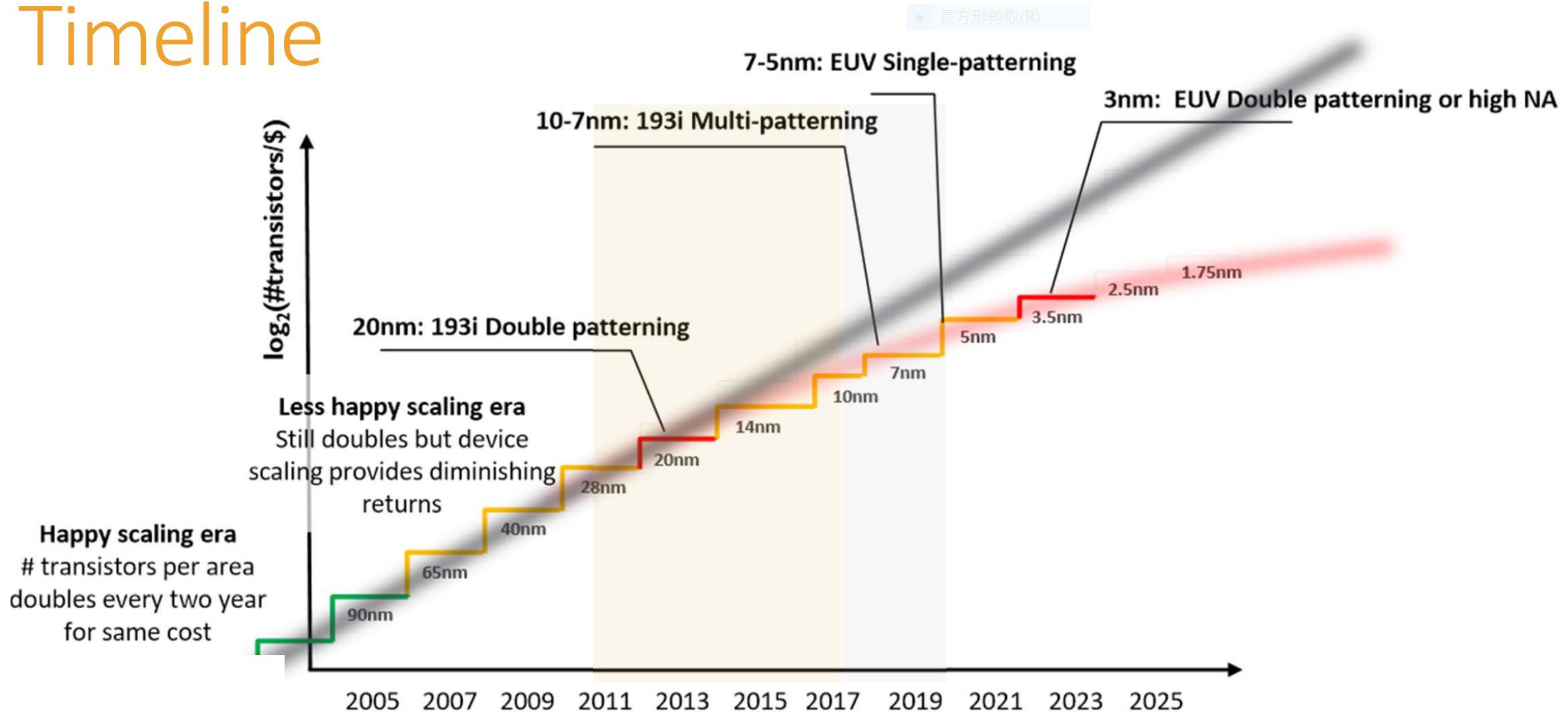
國立臺灣大學
National Taiwan University

# Outline

# Outline

# Will EUV Kill Multi-Patterning?

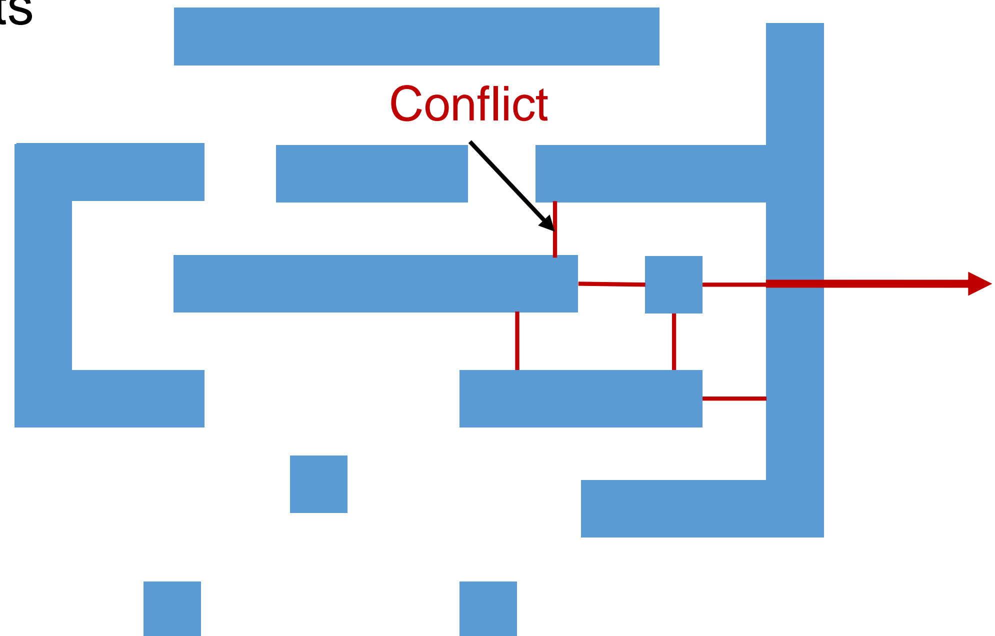
- Multiple patterning lithography (MPL) is still indispensable
  - Cost effectiveness and hybrid lithography capability




An Steegen (IMEC), iedm (2016)

# Multiple Patterning Lithography (MPL)

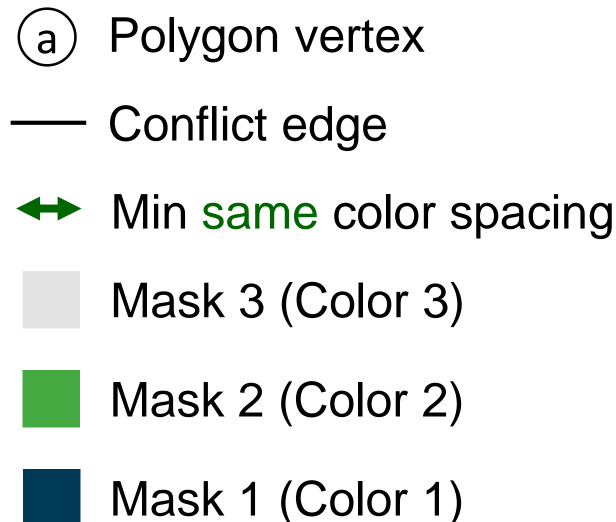
- Divides a layout into serval masks (colors)
- Manufactures the masks through a series of exposure and etching processes
- Relies on two major tasks
  - Layout decomposition
    - Reports coloring conflicts
  - Layout compliance
    - Modifies the layout to remove conflicts

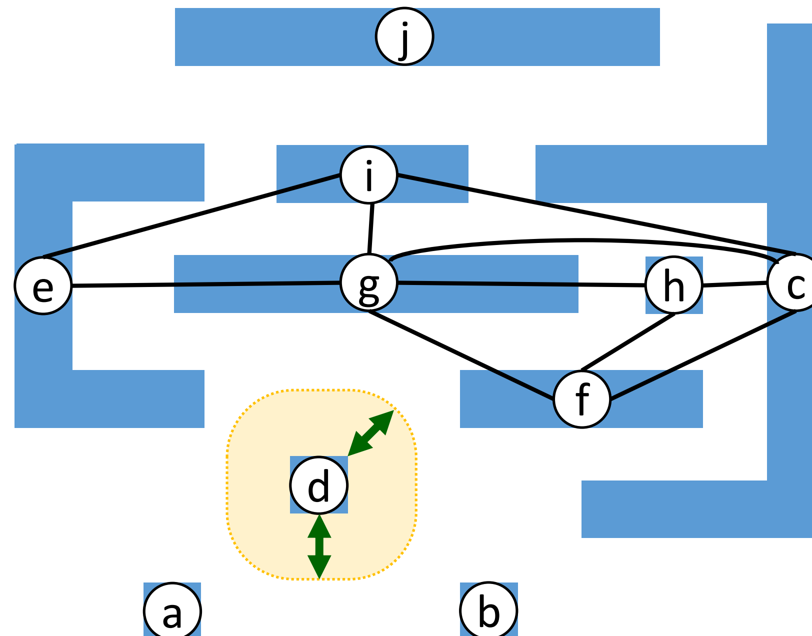



Mask 1 Mask 2 Mask 3

# MPL Layout Decomposition (MPLD)

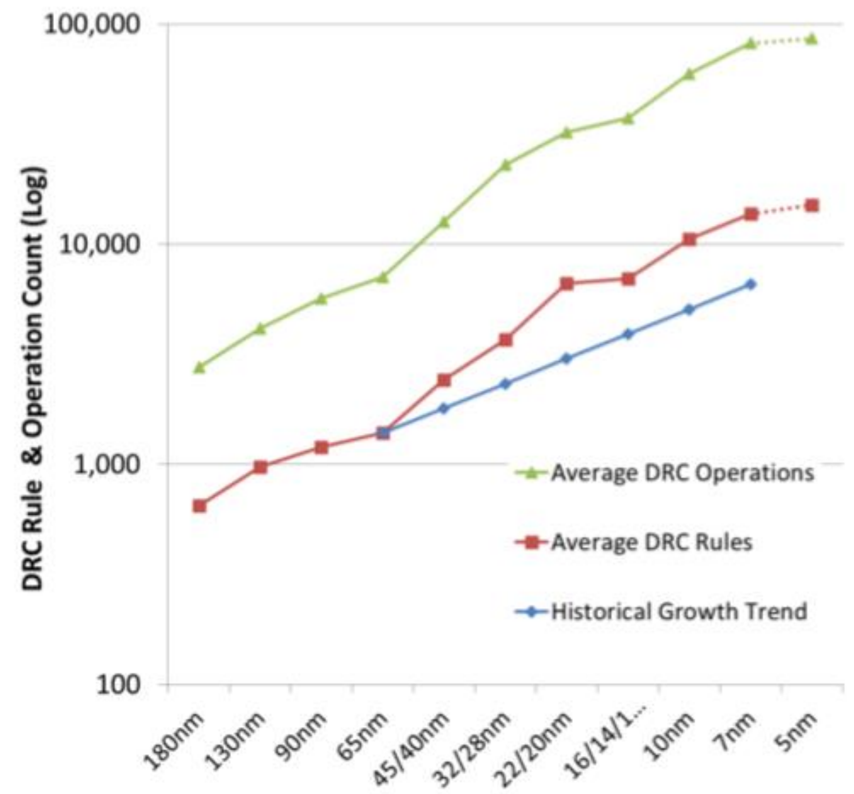
- Main focus for prior research endeavors
- Reduced to graph coloring on a conflict graph
  - Each mask corresponds to a color
  - Each vertex represents a polygon
  - Each edge indicates the same color spacing violation



# MPL Layout Compliance
## *Issues*

- A layout cannot be manufactured with unresolved conflicts
- Design rules explode in size and complexity
  - From 1,000 to 10,000+
  - Manual or semi-auto fixing is not applicable
- Little research in literature
  - Has not been automated well
  - Semi-automated approach
  - Fix only special patterns
    - e.g., $K_4$ for TPL

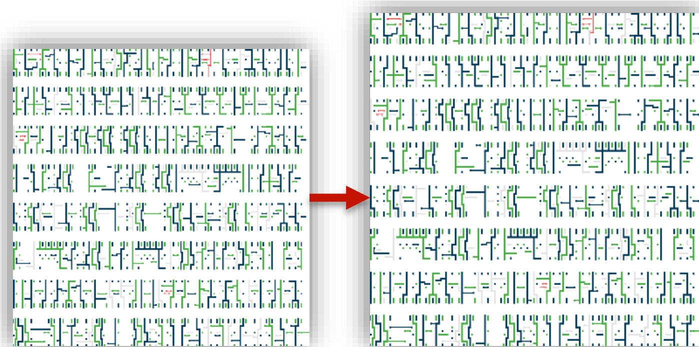



Ref: E. Sperling. 2018. Design rule complexity rising. (April 2018). Manufacturing & Process Technology, Semiconductor Engineering. Retrieved from https://semiengineering.com/design-rule-complexity-rising/

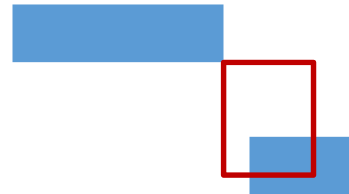
# MPL Layout Compliance

***Designer's Perspective***

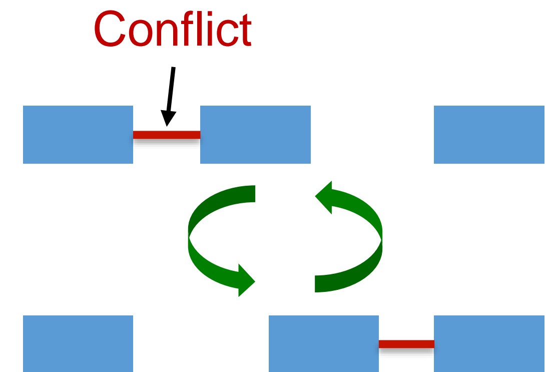
- Minimal polygon displacement
  - An input layout has been optimized for power, timing, and area
- Minimal topology disturbance
  - Modern design rules are strongly correlated to topology and polygon shapes
- Fast convergence
  - Not to create new conflicts/violations and not to alter polygon shapes



Enlarged design area



End-of-line keepout rule




Ping-Pong

# Our Contributions

- Propose the first fully automatic multiple patterning layout compliance approach
  - Is advantageous from a designer's perspective
- Devise a novel row slicing scheme
  - Facilitate extracting topology relations of polygons
- Model the problem as a polygon legalization problem
  - Solve the corresponding quadratic program efficiently
- Collect multiple edges from an arbitrary conflict pattern
  - Enhance the fixing flexibility
- Present a novel polygon displacement estimation technique
  - Select proper breaking edges to minimize layout change

# Outline

- Introduction

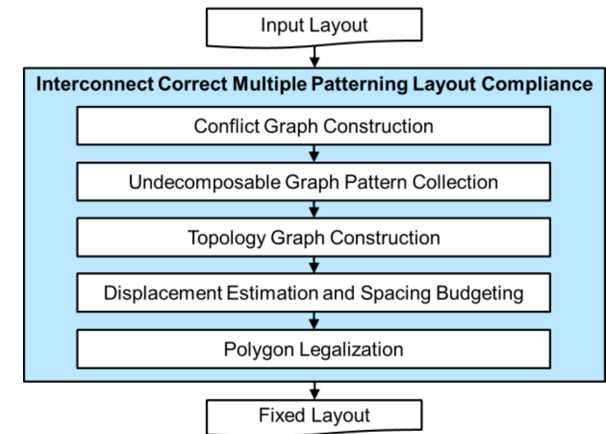
- **Problem Formulation**

- Our Approach

- Experimental Results

- Conclusion

# Problem Formulation




- Given
  - Design
    - A layout represented by a set of polygons
  - MPL design rules
    - The number of available masks
    - The minimum different color spacings
    - The minimum same color spacings of each polygon
- Do
  - Shift polygons
- Objective
  - Minimize the number of coloring conflicts without creating new conflicts
  - Minimize topology disturbance and polygon displacement

# Outline

- Introduction

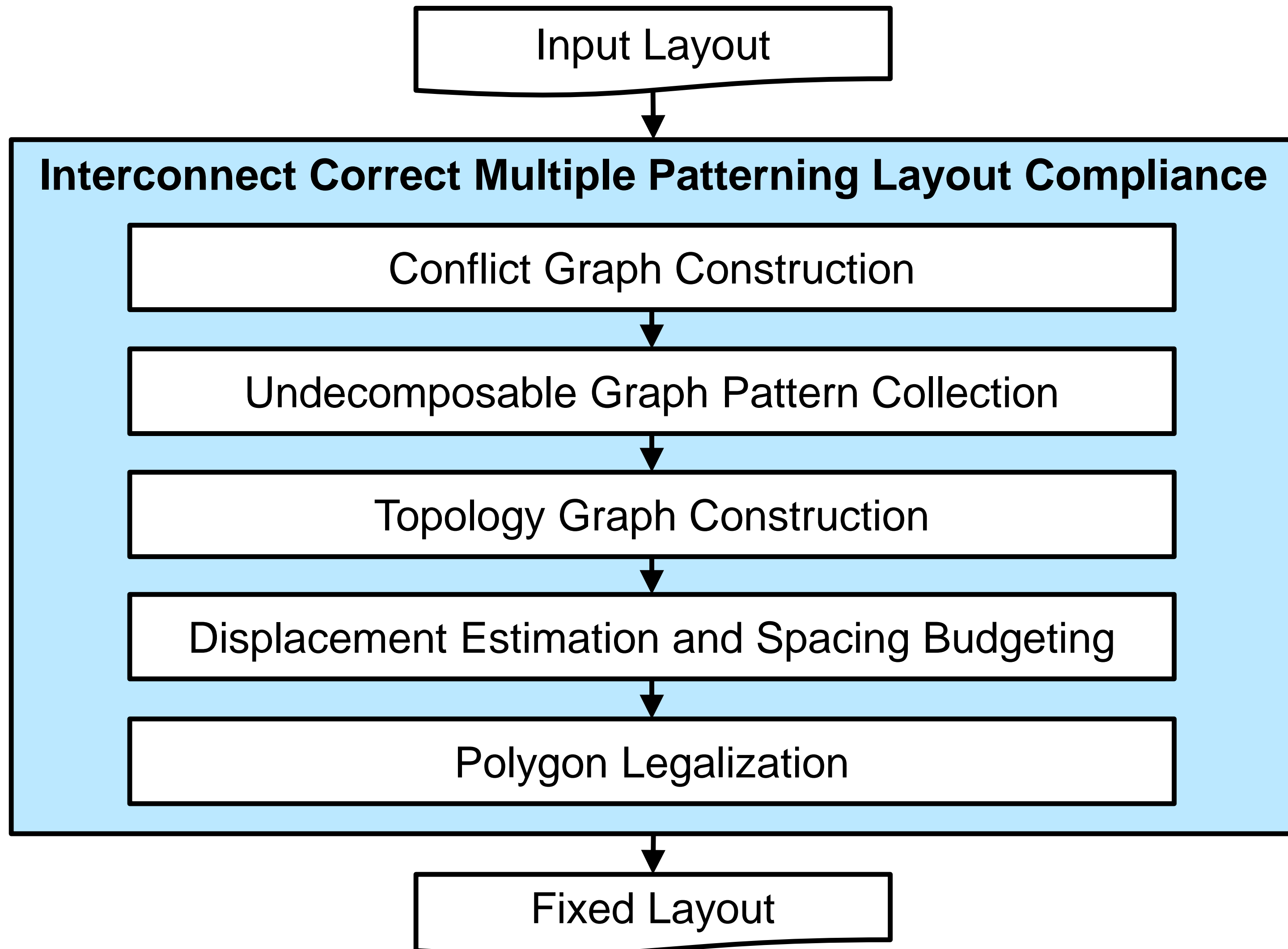
- Problem Formulation

- **Our Approach**

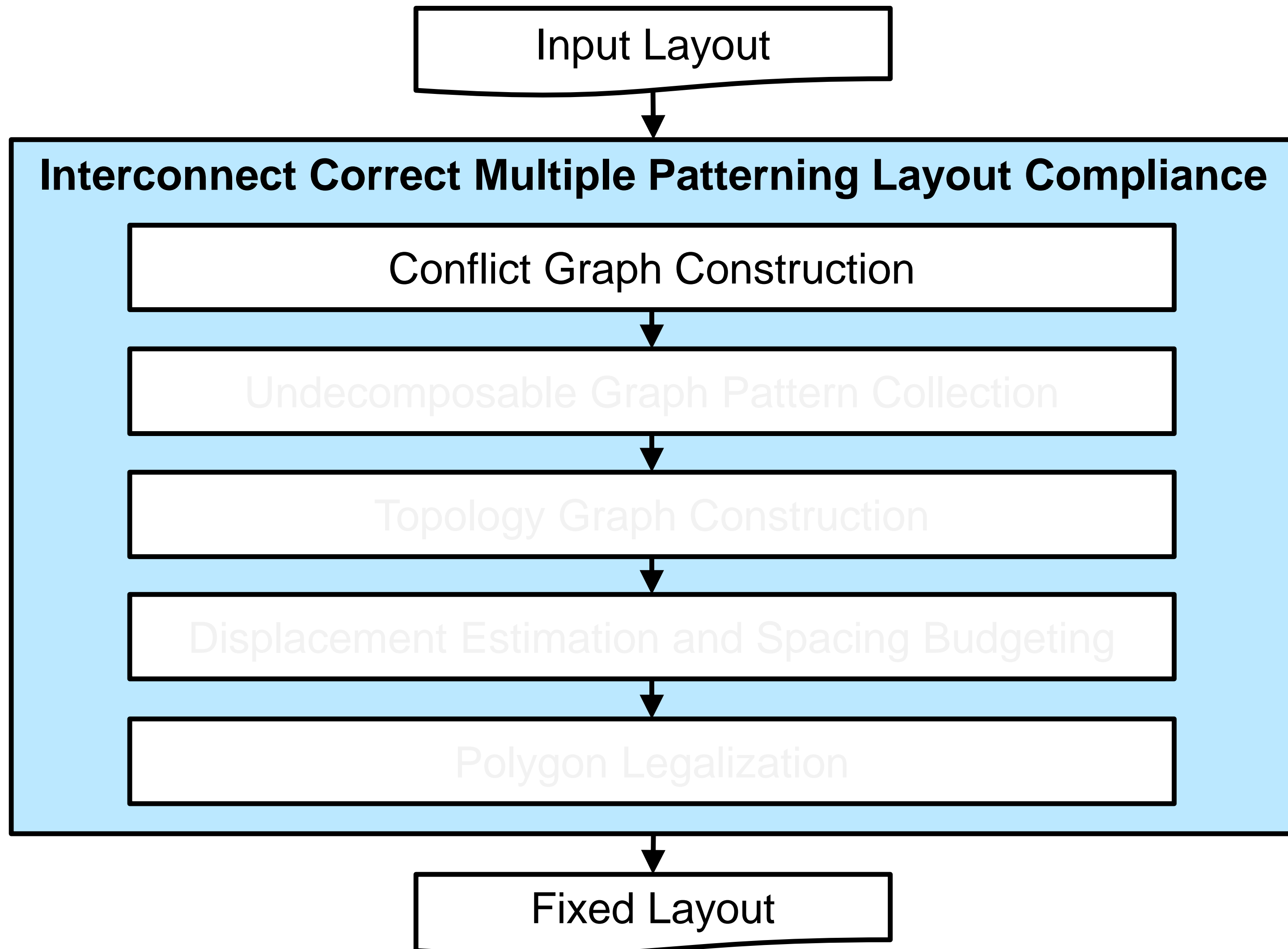
- Experimental Results

- Conclusion

# Overview

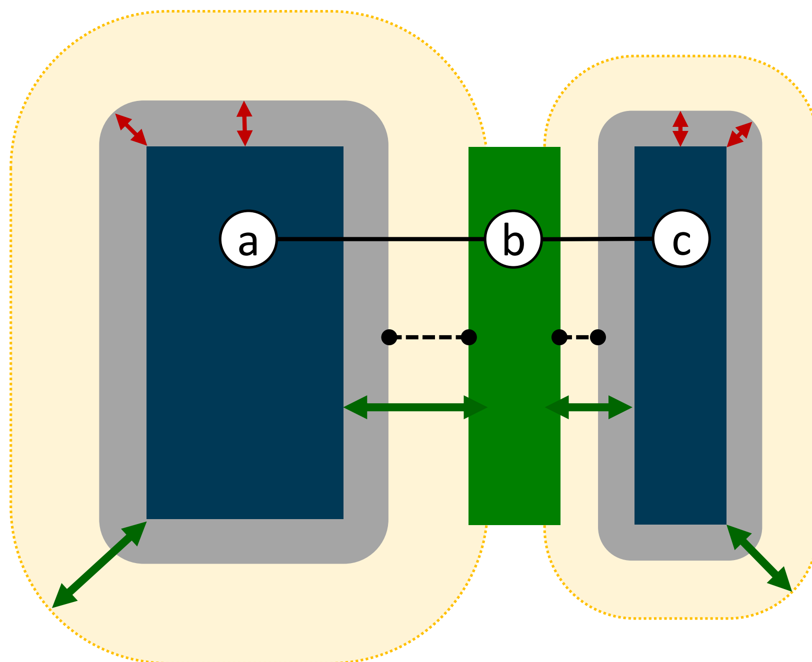


# Overview



# Conflict Graph Construction

- Assume an input layout is free of different color spacing violations
- Build conflict graph based on same color spacings
  - Add one edge for two polygons if there is a violation between them




a Polygon vertex

Conflict edge

Minimum different color spacing

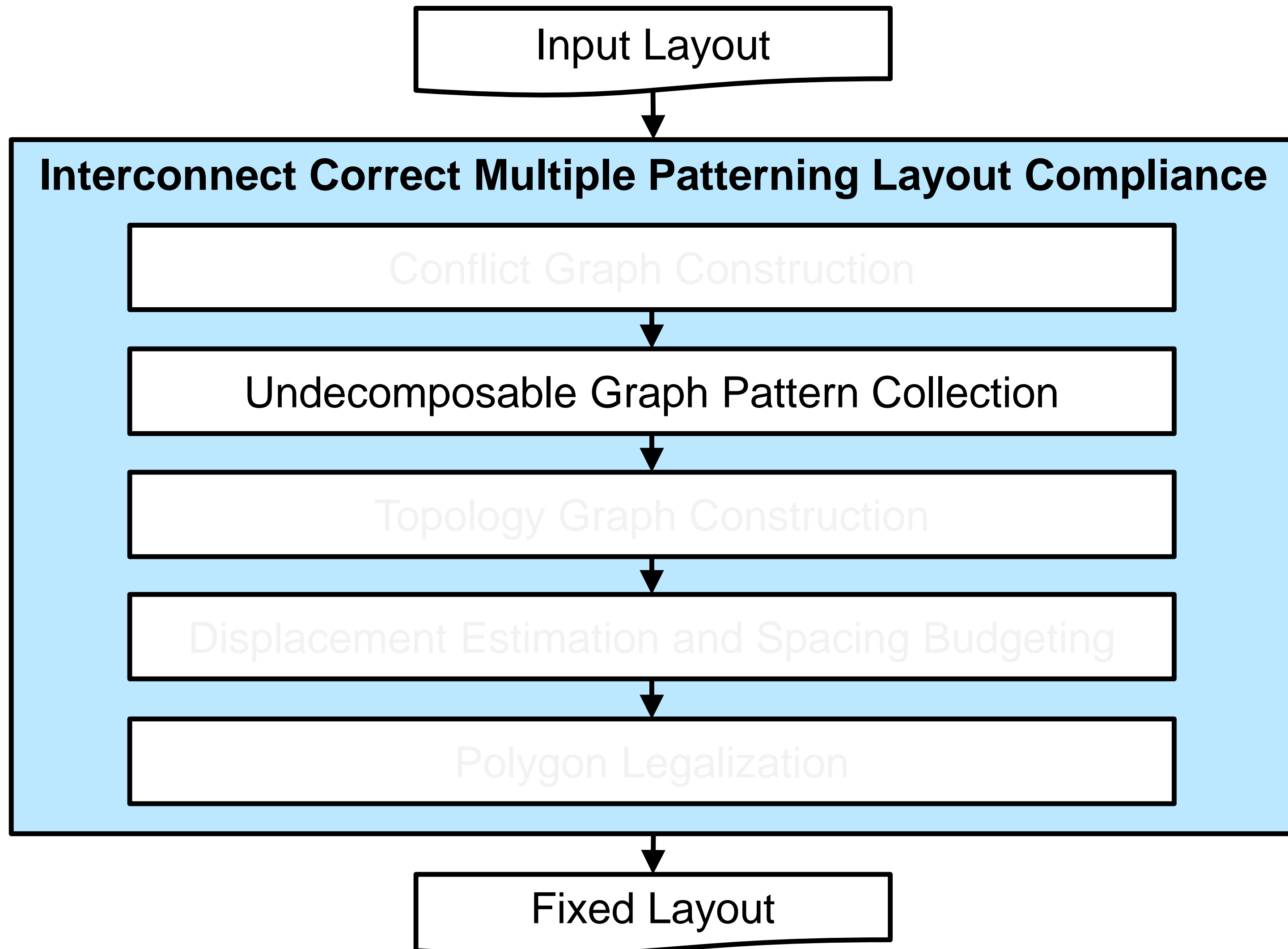
Minimum same color spacing

Slack

Spacing territory

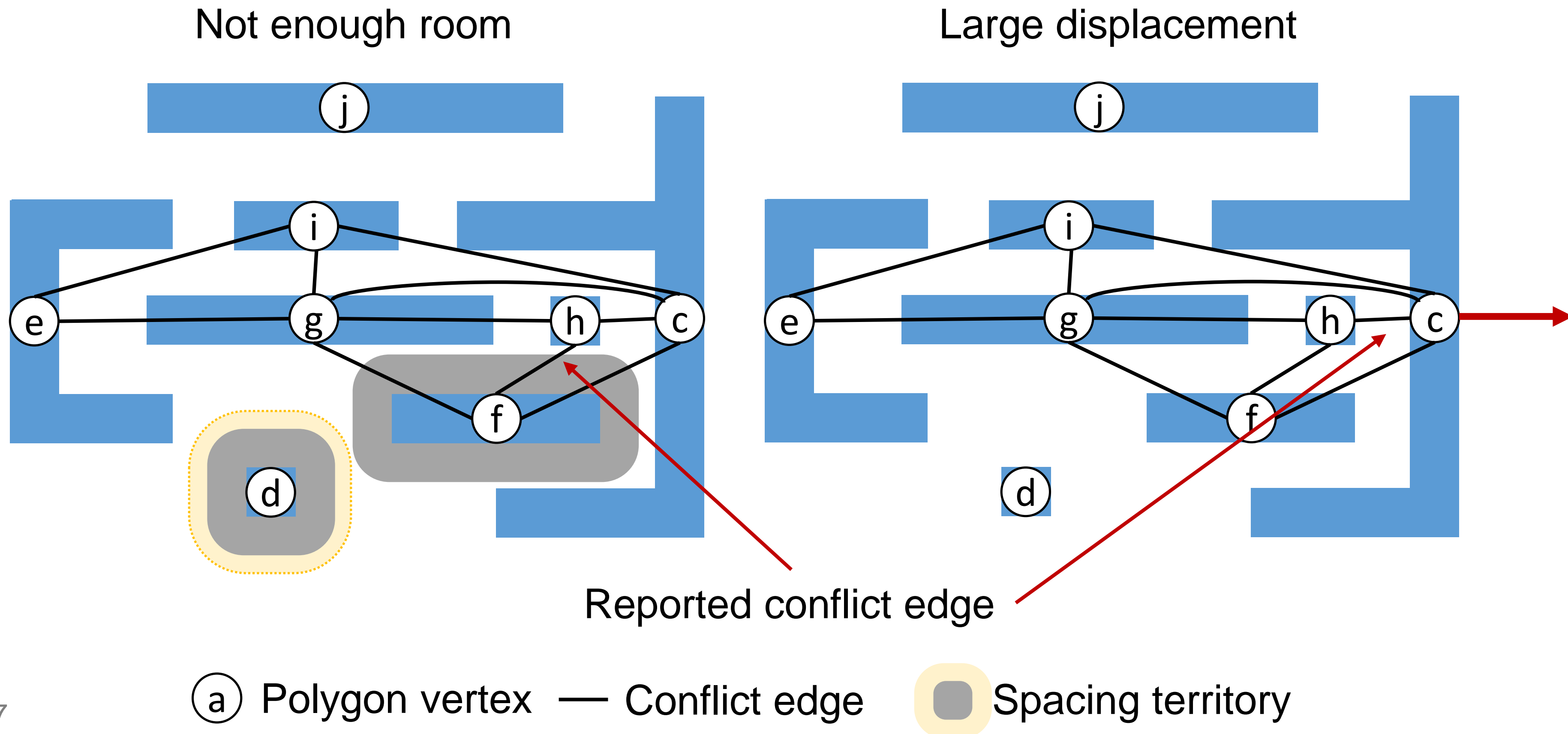
Mask 1 Mask 2

# Overview



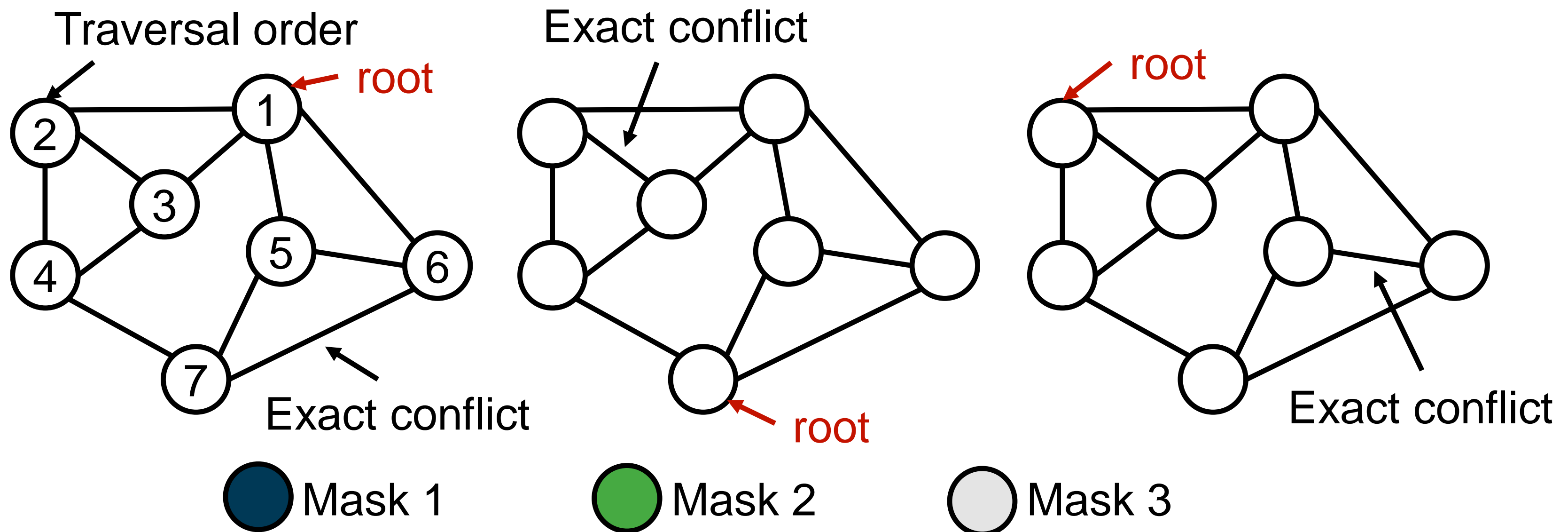
# Insufficient Conflict Edge Report

- A reported conflict edge may not be good for shifting
  - Conventional MPLD reports one conflict edge for one conflict graph pattern



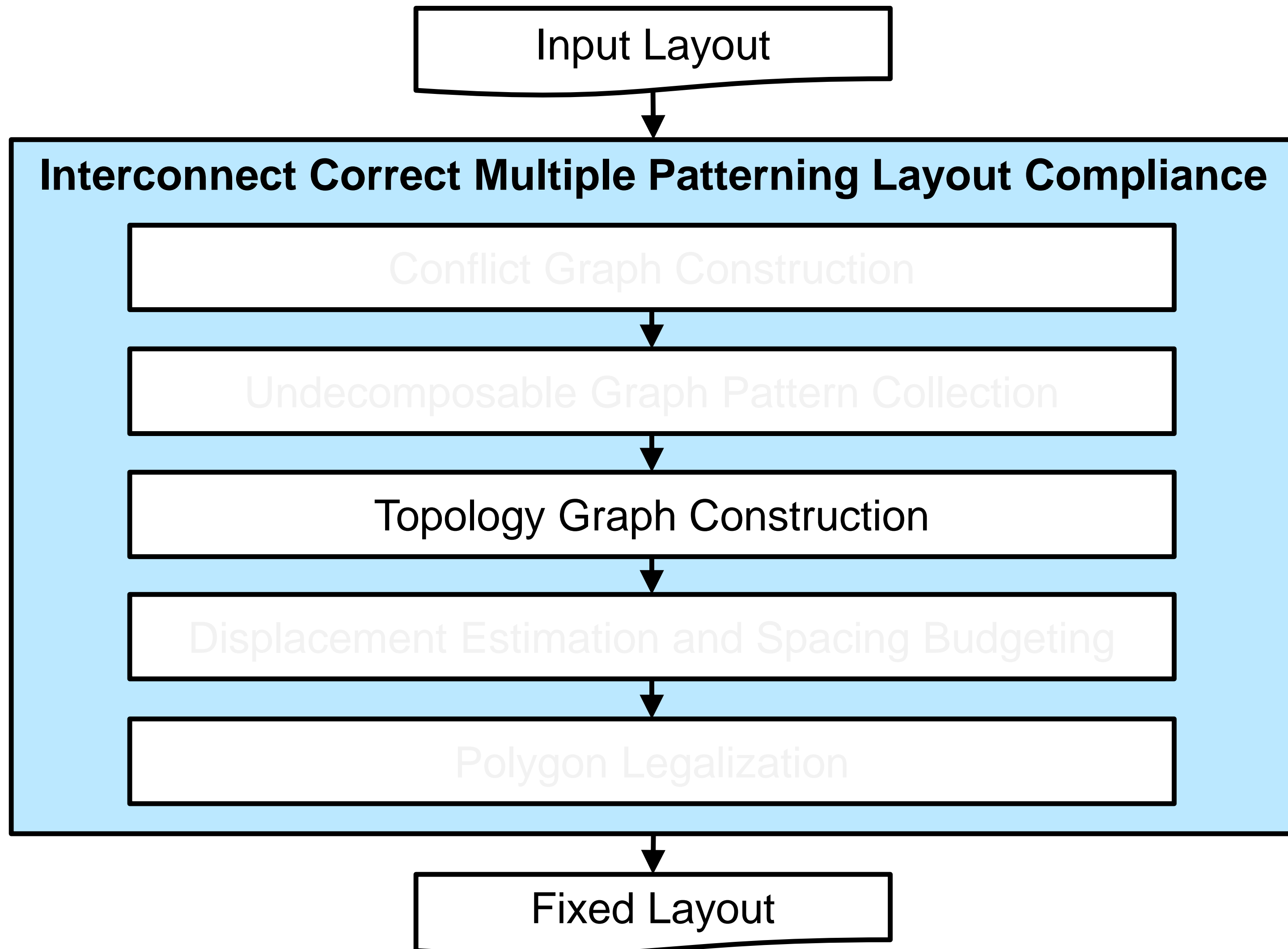
# Undecomposable Graph Pattern Collection

- Attempt to identify partial undecomposable graph patterns by extending the exact conflict reporting
  - Identifying native conflict graph patterns is an open problem
- Change the traversal orders of vertices of Algorithm X*
  - Provide fixing flexibility



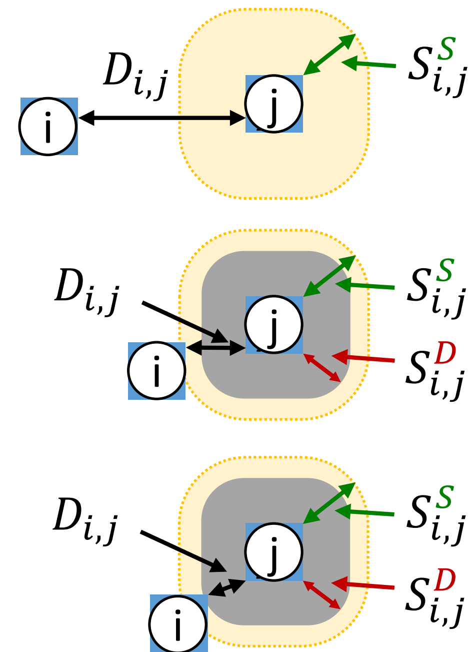

I. H.-R. Jiang, H.-Y. Chang. Multiple patterning layout decomposition considering complex coloring rules and density balancing. *IEEE TCAD,* 36, 12 (Dec. 2017), 2080-2092. Also see in *Proc. DAC '16*, Article 40, 6 pages.

# Overview

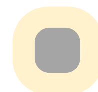


# Topology Extraction

- Record topology relations of polygons
  - Protect polygon shifting against new spacing violations
- Three coloring conditions of two polygons $i, j$
  - Case 1: $D_{i,j} \geq S_{i,j}^S$
    - No conflict edge between them
  - Case 2: $S_{i,j}^S > D_{i,j} \geq S_{i,j}^D$
    - A conflict edge between them
  - Case 3: $S_{i,j}^D > D_{i,j}$
    - A hard spacing violation between them




$S_{i,j}^S$ Minimum same color spacing



Spacing territory

$S_{i,j}^D$ Minimum different color spacing

$D_{i,j}$ Euclidean distance

# Principle of Topology Extraction

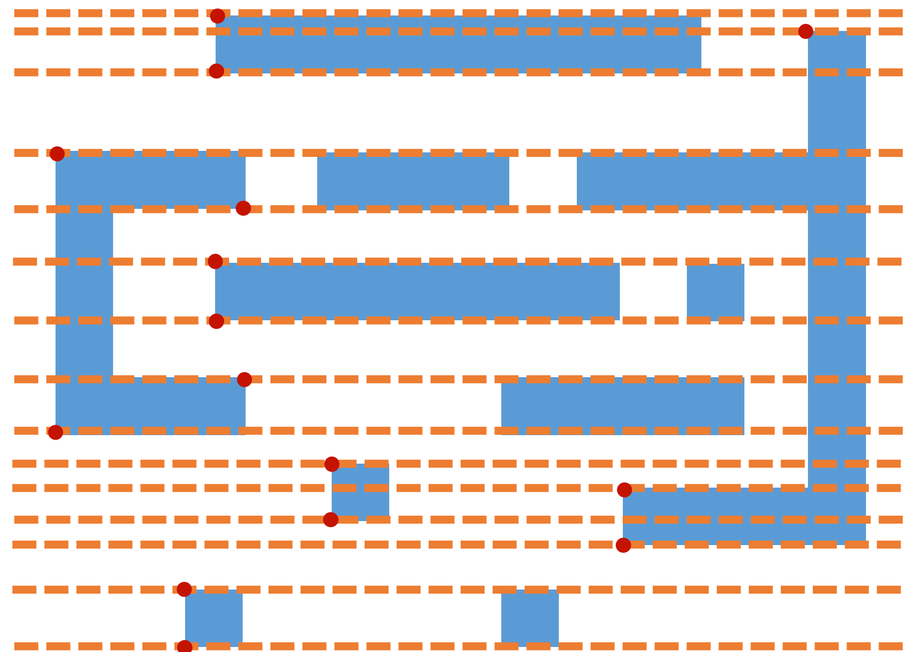
*"Two polygons have a topology relation if polygon shifting may alter and worsen the coloring condition between them."*

- *i.e., from case 1 to case 2 or from case 2 to case 3*

- *View a polygon shift as a horizontal and/or a vertical shift*
  - *Construct horizontal and vertical topology graph*
- *Exhaustively testing every two polygon edges/corners to extract all topology relations is time consuming*
  - *Layout slicing is helpful for capturing local topology*

# Conventional Topology Slicing

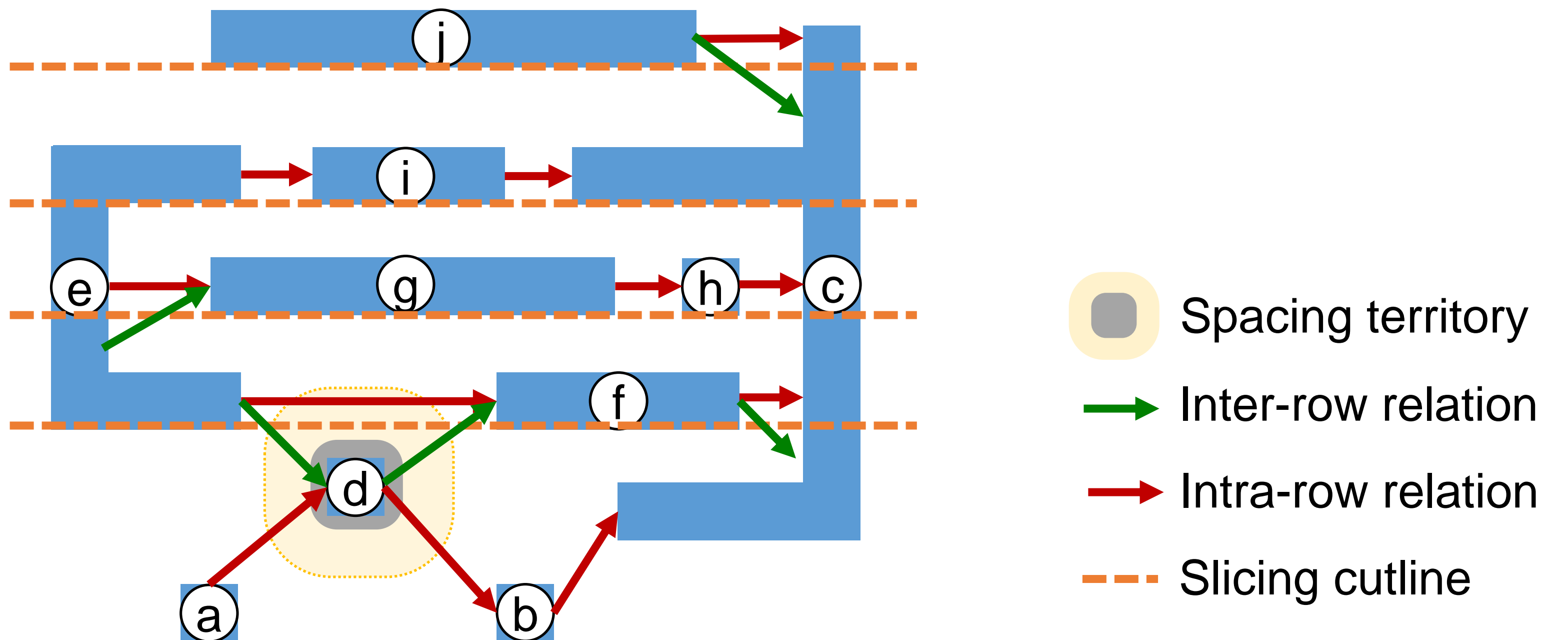
- Cut along polygon corners
  - May generate numerous narrow rows
  - May contain duplicated/partial topology information of other rows



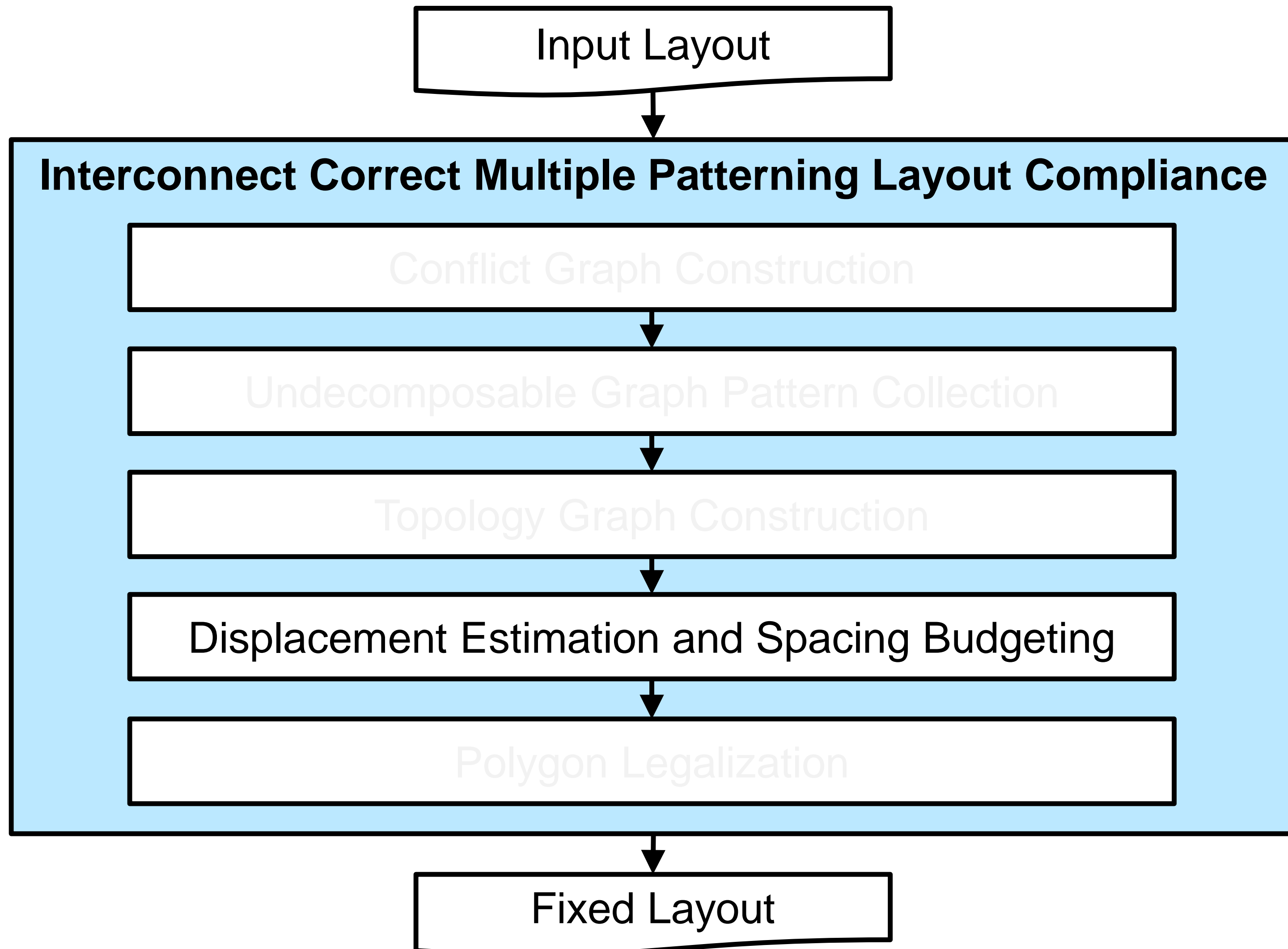
Slicing cutline • Corner

# Topology Graph Construction

- Our slicing
  - Sort all polygons in the non-decreasing lexicographic order of $(y, x)$
  - Cut a line if their projections overlap in $x$ axis, but not in $y$ axis
- Construct topology graph
  - Based on slicing and the principle of topology extraction
  - Preserve the strictest spacing requirement if more than one arc

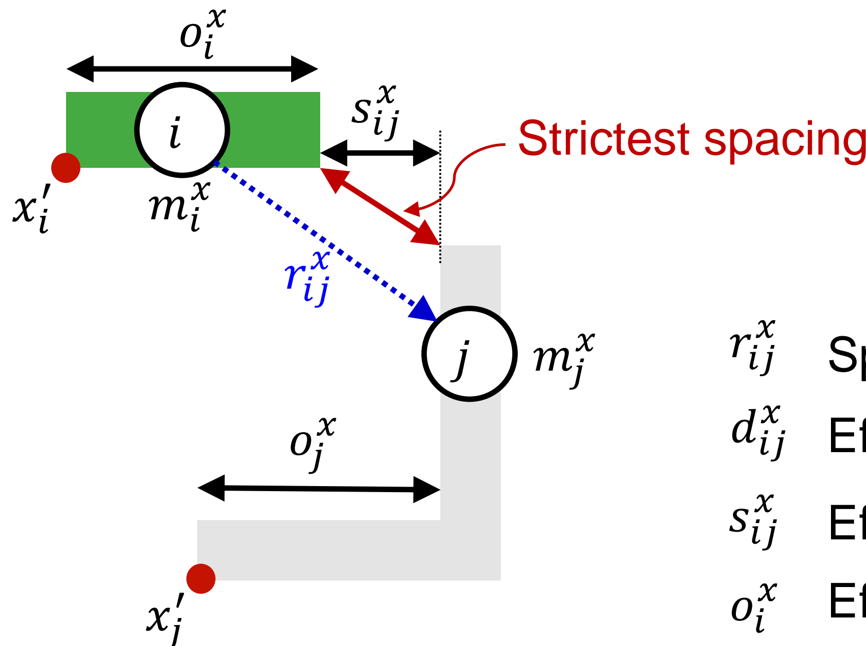


# Overview



# Displacement Estimation

- For fixing a conflict…
  - Breaking any edge within a conflict pattern can resolve this conflict
  - Break the edge with sufficient shifting room and least influence
  - Split to horizontal + vertical shift if cannot resolved by one direction
- Spacing slack calculation
  - E.g. horizontal (x) shifting




$$r_{ij}^x = (x_j' - x_i') - d_{ij}^x$$

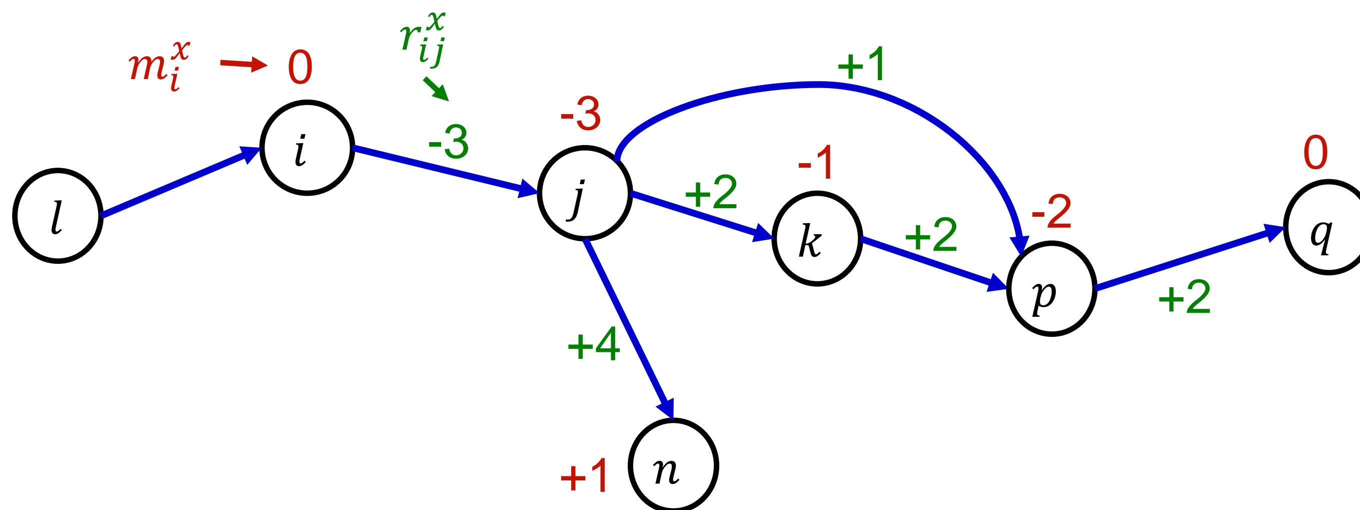
$$d_{ij}^x = o_i^x - o_j^x + s_{ij}^x$$

| | |
|---|---|
| $r_{ij}^x$ | Spacing slack in x |
| $d_{ij}^x$ | Effective distance requirement in x |
| $s_{ij}^x$ | Effective spacing constraint in x |
| $o_i^x$ | Effective spacing offset in x respect to $x_i'$ |

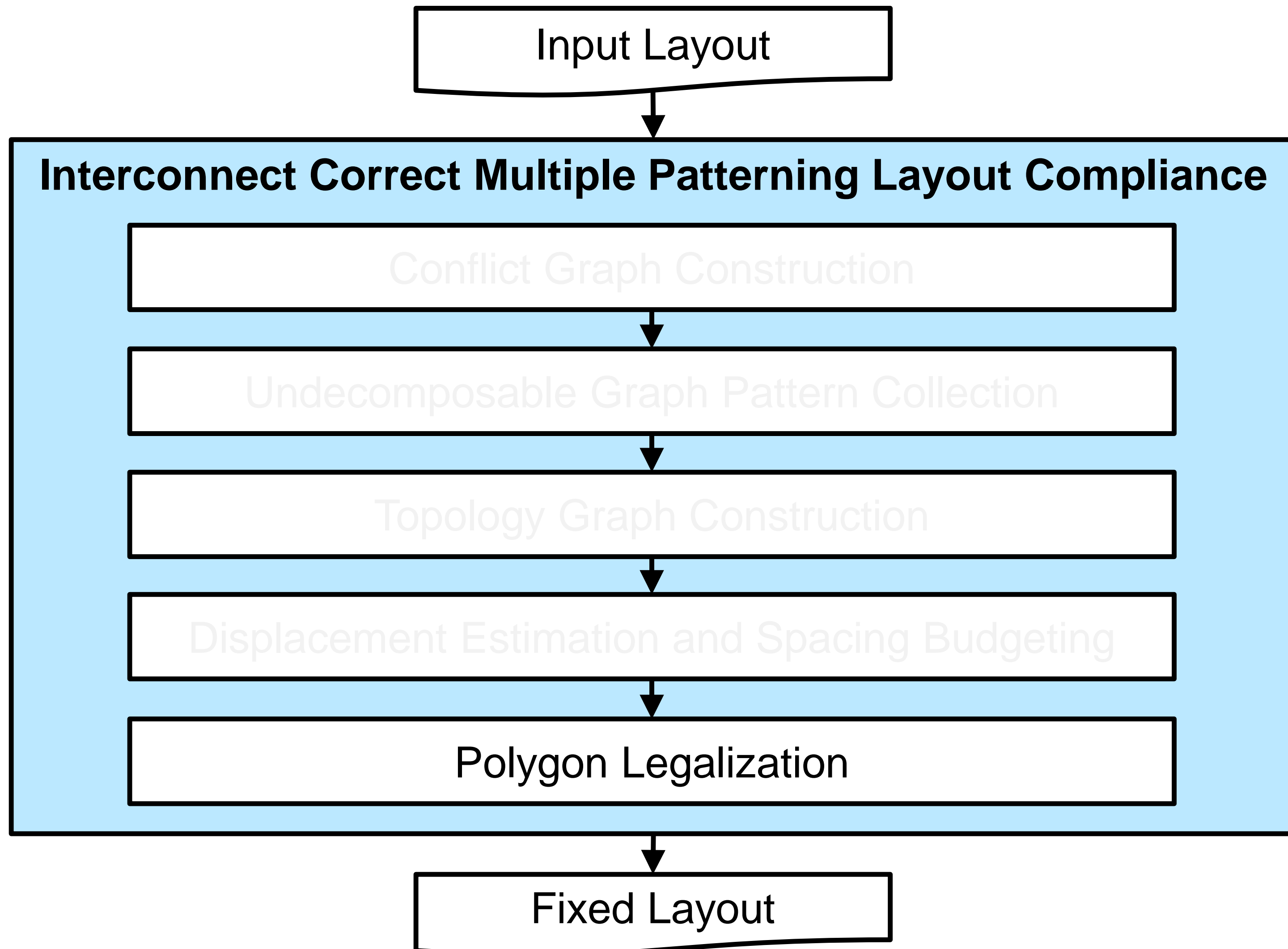
# Slack Absorption

***Spacing budgeting***

- To break $(i, j)$, modify $s_{ij}^x$ to meet same color spacing
- Compute slack $r_{ij}^x$ from arc $(i, j)$ and shift $j$ rightwards
  - Propagate slack in topological order
- Compute movements for influenced polygons
  - $m_j^x = m_i^x + r_{ij}^x,\ m_i^x = 0$
- Propagate until the movement is nonnegative
  - No feasible solution if negative cycle exist



# Overview



# Polygon Legalization

- Fix coloring conflicts by shifting polygons
  - Without creating new conflicts and with least layout change
- Select the edge with the best estimated displacement as the target breaking edge for each conflict
- Formulate the polygon legalization problem as a quadratic program

$$\min \frac{1}{2}\sum_{i=1}^{n}(x_i - x_i')^2$$

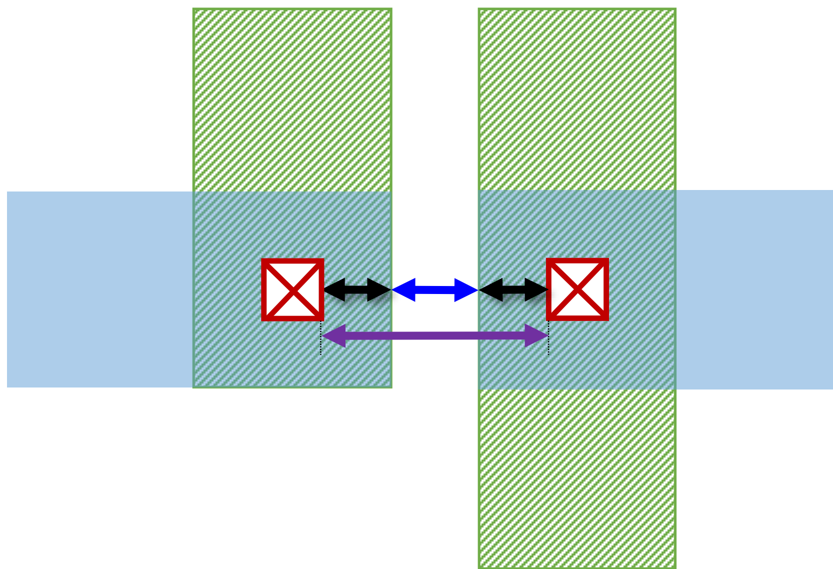
$$s.t.\ x_j - x_i \geq d_{ij}^{x}, if\ (i,j) \in E_T \cup E_B, x_i \geq 0$$

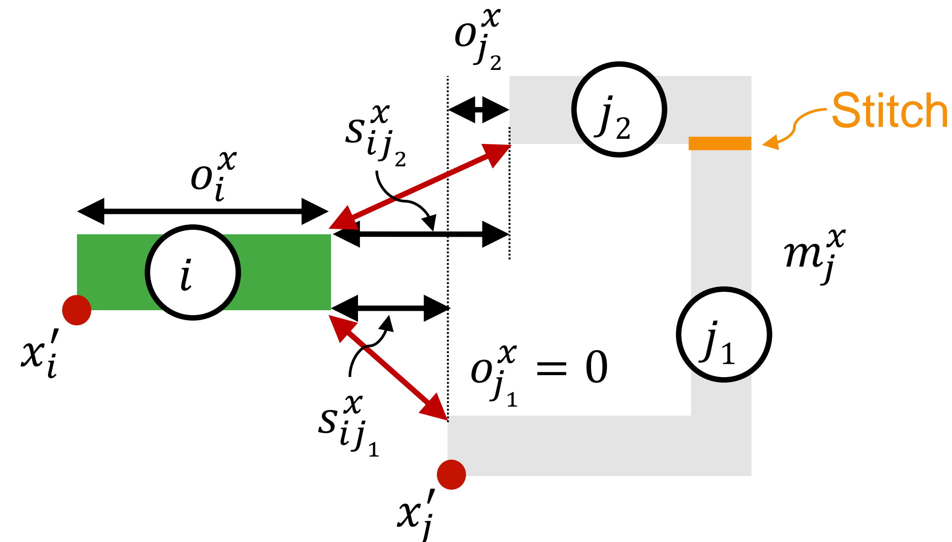
$$E_T: Topology\ relation\ edge$$

$$E_B: Target\ breaking\ edge$$

# Multilayer and Stitch Handling

- Multilayer can be handled easily by considering multiplayer related constraints
- Stitches can be viewed as additional slicing cutlines






Via12 M1 M2

Via12 Mx Enclosure (cross-layer)

Via Spacing (same-layer) Mx Spacing (same-layer)

# Outline

# Experimental Results

***Benchmark Statistics***

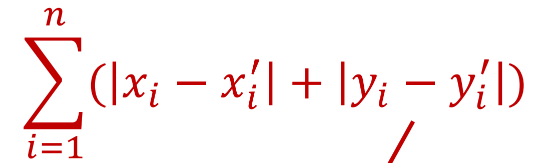
- Implemented in C++ and adopted GUROBI as solver
- Conducted on ISCAS-85&89 benchmark suite
  - Used by state-of-the-art MPLD works

**Benchmark Statistics**

| Circuit | $\lvert \mathbf{V_C} \rvert$ | $\lvert \mathbf{E_C} \rvert$ | $\lvert \mathbf{C} \rvert$ | Circuit | $\lvert \mathbf{V_C} \rvert$ | $\lvert \mathbf{E_C} \rvert$ | $\lvert \mathbf{C} \rvert$ |
|---|---|---|---|---|---|---|---|
| C432 | 1,109 | 1,222 | 4 | S1488 | 4,611 | 5,504 | 2 |
| C499 | 2,216 | 2,817 | 0 | S38417 | 67,696 | 79,527 | 73 |
| C880 | 2,411 | 2,686 | 7 | S35932 | 157,455 | 186,052 | 84 |
| C1355 | 3,262 | 3,326 | 3 | S38584 | 168,319 | 196,072 | 152 |
| C1908 | 5,125 | 5,598 | 1 | S15850 | 159,952 | 190,796 | 131 |
| C2670 | 7,933 | 9,336 | 6 | | | | |
| C3540 | 10,189 | 11,968 | 9 | | | | |
| C5315 | 14,603 | 16,881 | 9 | | | | |
| C6288 | 14,575 | 15,605 | 206 | | | | |
| C7552 | 21,253 | 24,372 | 22 | | | | |

$\lvert V_C \rvert$ #vertices (polygon) $\lvert E_C \rvert$ #conflict edges $\lvert C \rvert$ #coloring conflicts reported

# Experimental Results



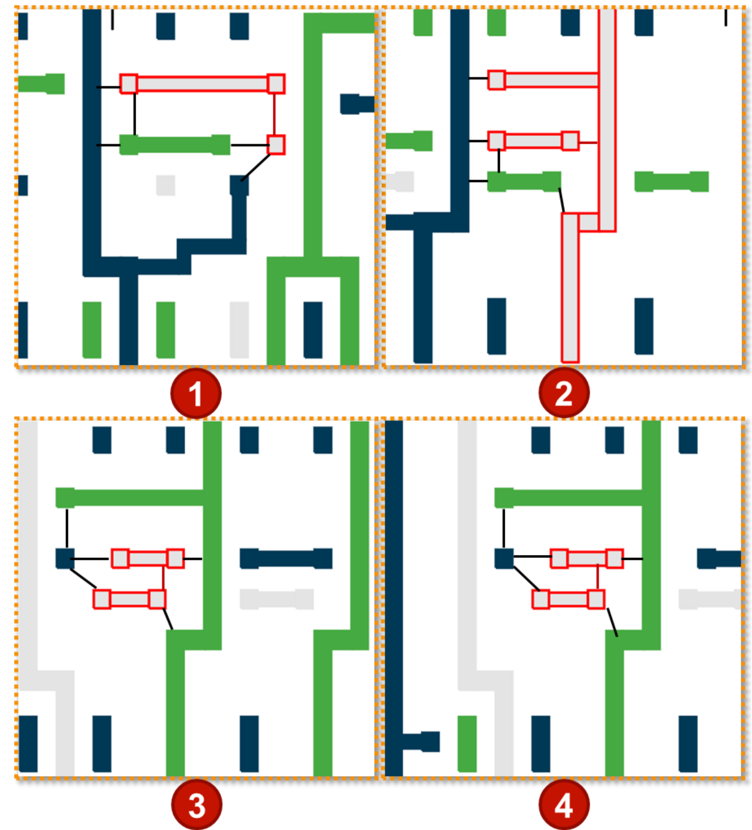

- Direct fixing: our approach without pattern collection

| | Direct Fixing | | | | | | | Ours | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Circuit** | $\lvert C\rvert$ | $\lvert E_x\rvert$ | $\lvert E_y\rvert$ | **#Cons** | **Disp** | **$CPU_{OT}$** | **$CPU_{QP}$** | $\lvert C\rvert$ | $\lvert E_x\rvert$ | $\lvert E_y\rvert$ | **#Cons** | **Disp** | **$CPU_{OT}$** | **$CPU_{QP}$** |
| C432 | 0 | 3 | 1 | 6,447 | 10,649 | 0.054 | 0.17 | 0 | 1 | 3 | 6,445 | 183 | 0.047 | 0.15 |
| C499 | 0 | 0 | 0 | 0 | 0 | 0.085 | 0.00 | 0 | 0 | 0 | 0 | 0 | 0.093 | 0.00 |
| C880 | 0 | 6 | 1 | 13,283 | 2,147 | 0.084 | 0.24 | 0 | 4 | 3 | 13,283 | 282 | 0.078 | 0.31 |
| C1355 | 0 | 3 | 0 | 17,906 | 453 | 0.138 | 0.17 | 0 | 2 | 1 | 17,906 | 288 | 0.125 | 0.27 |
| C1908 | 0 | 0 | 1 | 27,872 | 88 | 0.210 | 0.20 | 0 | 1 | 0 | 27,872 | 26 | 0.203 | 0.12 |
| C2670 | 0 | 6 | 0 | 43,618 | 5,183 | 0.322 | 0.39 | 0 | 4 | 3 | 43,618 | 202 | 0.329 | 0.95 |
| C3540 | 0 | 7 | 2 | 55,315 | 3,700 | 0.510 | 0.94 | 0 | 4 | 6 | 55,316 | 323 | 0.500 | 1.98 |
| C5315 | 0 | 7 | 2 | 83,954 | 5,637 | 0.635 | 2.94 | 0 | 3 | 6 | 83,953 | 275 | 0.640 | 3.27 |
| C6288 | 1 | 153 | 53 | 80,661 | 94,602 | 0.673 | 5.81 | 0 | 128 | 80 | 80,656 | 12,196 | 0.750 | 3.12 |
| C7552 | 0 | 15 | 7 | 119,658 | 21,267 | 0.904 | 5.81 | 0 | 16 | 7 | 119,657 | 5,666 | 0.907 | 4.19 |
| S1488 | 0 | 2 | 0 | 28,167 | 127 | 0.231 | 0.29 | 0 | 0 | 2 | 28,167 | 12 | 0.360 | 0.26 |
| S38417 | 1 | 41 | 32 | 371,102 | 19,507 | 3.654 | 14.16 | 0 | 30 | 47 | 371,104 | 521 | 3.757 | 16.16 |
| S35932 | 1 | 36 | 51 | 876,664 | 10,221 | 9.280 | 36.07 | 0 | 39 | 50 | 876,859 | 873 | 10.048 | 35.71 |
| S38584 | 0 | 81 | 71 | 920,196 | 20,989 | 9.460 | 35.06 | 0 | 73 | 79 | 920,195 | 1,968 | 11.063 | 35.38 |
| S15850 | 1 | 76 | 56 | 876,323 | 36,099 | 9.351 | 35.28 | 0 | 69 | 65 | 876,319 | 1,190 | 10.762 | 37.19 |
| **Ratio** | | | | | **16.11** | **1.01** | **0.96** | | | | | **1.00** | **1.00** | **1.00** |

Disp: displacement | $|E_{x/y}|$: # conflicts solved by x/y shifting | $CPU_{QP}$: QP solving time

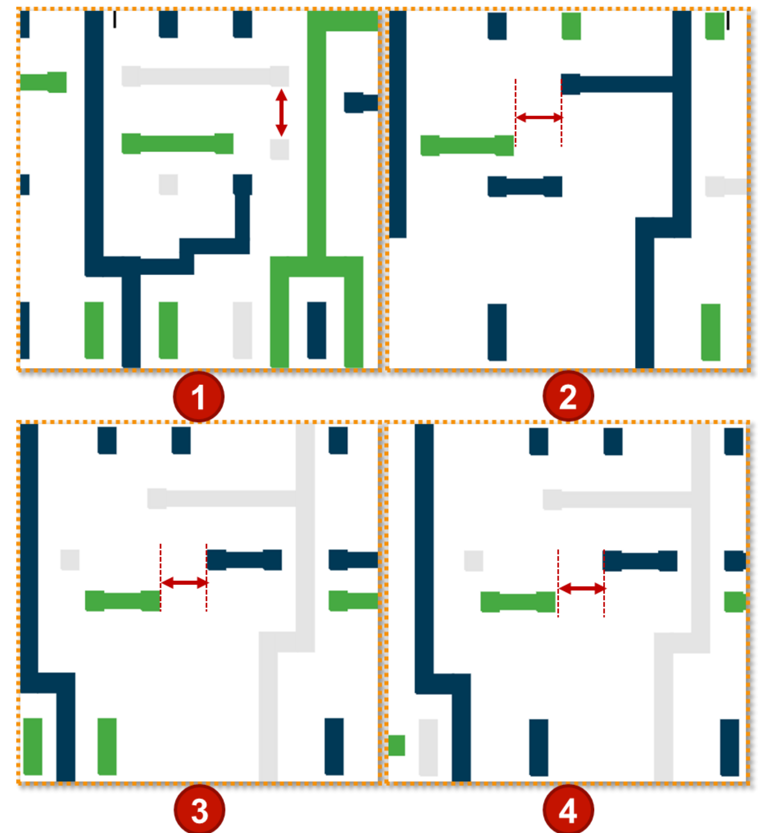
#Cons: #Constraints in the QP | $|C|$: # remaining conflict | $CPU_{OT}$: runtime of other steps

# C432 – 4 Conflicts



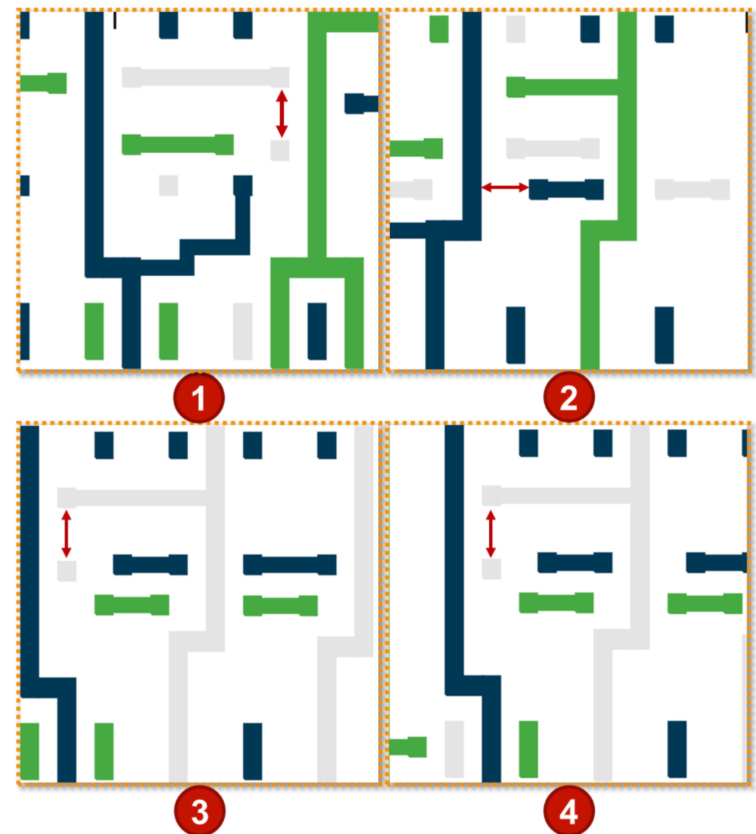
Influenced region

# C432 – Direct Fix

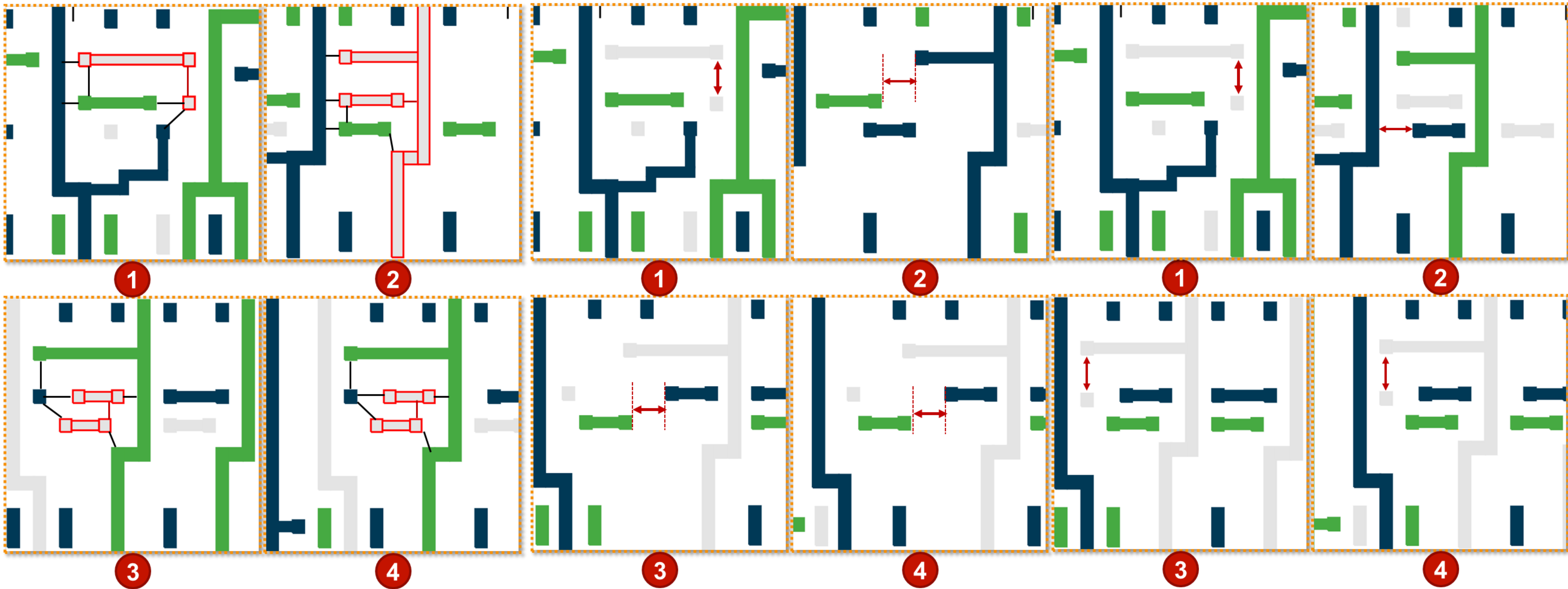


Influenced region

# C432 – Ours



# C432

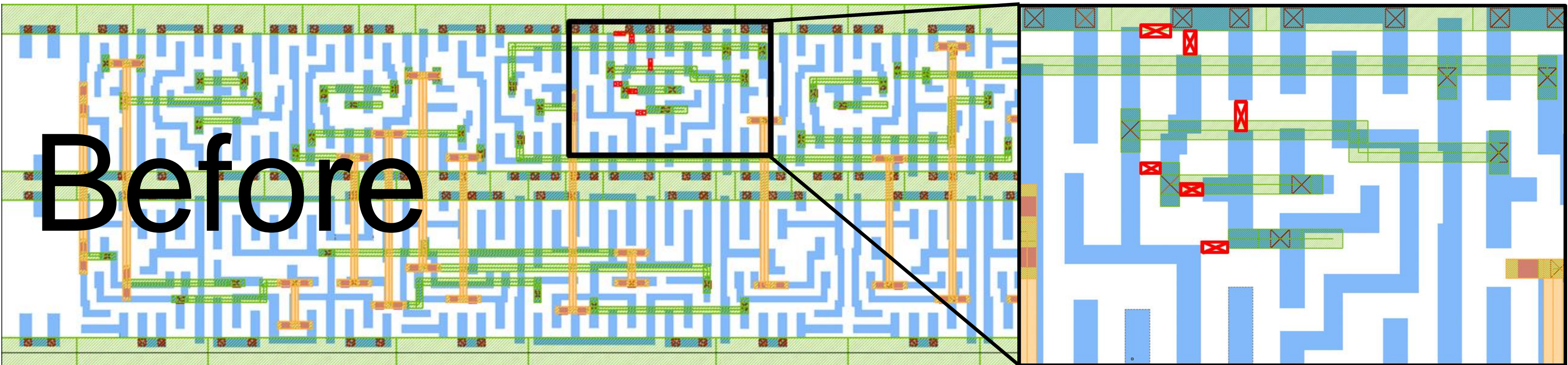



4 Conflicts

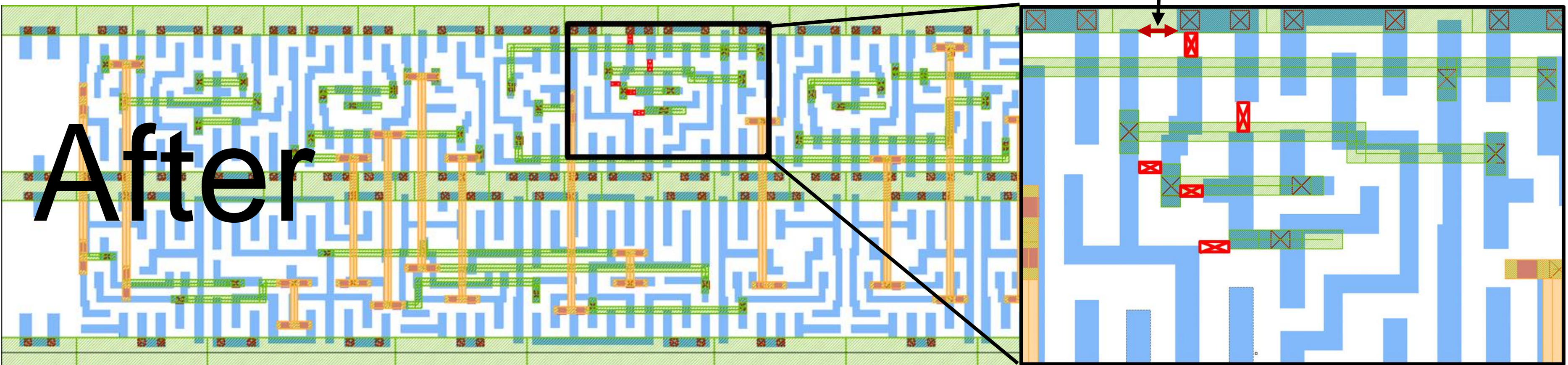
Direct Fixing

Ours

# Multilayer Result






M1 M2 M3 V12 V23 conflict

# Outline

# Conclusion

- Presented the first fully automatic multiple patterning layout compliance
  - Novel row slicing scheme
    - Facilitate topology extraction on arbitrary rectilinear shapes
  - Modeling the polygon legalization problem
    - Can be solved efficiently
  - Undecomposable pattern collection
    - No restricted to only special conflict patterns
  - Slack absorption
    - Estimate polygon displacement and identify the influenced region
- Experimental results show that our approach has superior efficiency and effectiveness

# Thank you !