Simultaneous Template Optimization and Mask Assignment for DSA with Multiple Patterning

Jian Kuang, Junjie Ye, Evangeline F.Y. Young

Department of Computer Science and Engineering

The Chinese University of Hong Kong
香 港 中 文 大 學

Outline

- Introduction to DSA & Multiple Patterning
- Problem Formulation
- DSA with Double Patterning
- DSA with Triple Patterning
- Results & Conclusions

DSA

- Directed Self-Assembly
- contact/via manufacturing
- promising Next Generation Lithography for 10nm

DSA

- *guiding templates* are needed to form the vias (contacts) correctly
- single-hole template v.s. multiple-hole template
- costs of templates: # of holes, irregular shape, distance

Multiple patterning

- divide the features into multiple masks
- coloring problem

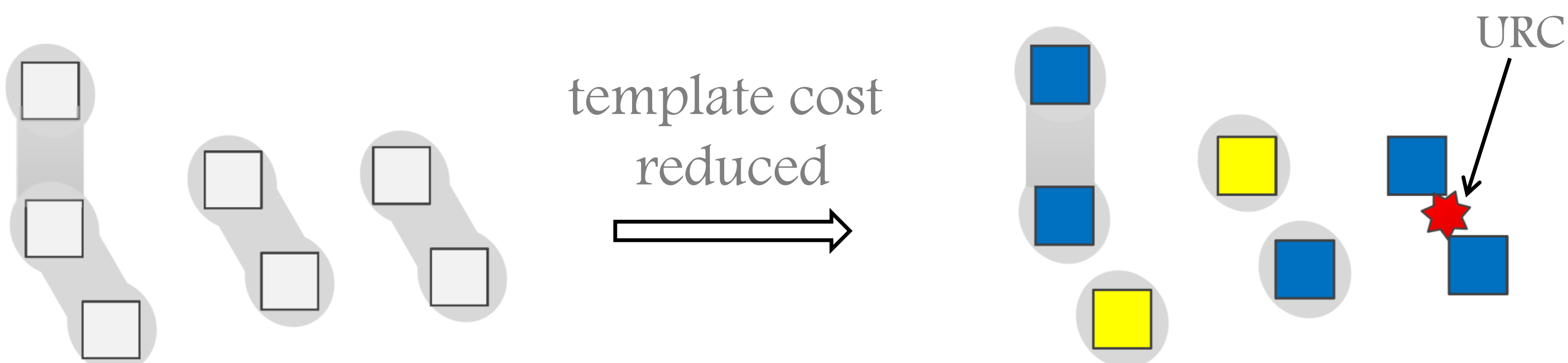
DSA + multiple patterning

- Step 1：multiple masks to print templates
- Step 2： templates to guide the DSA process

DSA + multiple patterning

- for two vias within a *threshold-distance*
  - on different masks
    - or
  - on the same mask & grouped (in the same template)

- otherwise, an UnResolved Conflict (URC) occurs



DSA + multiple patterning

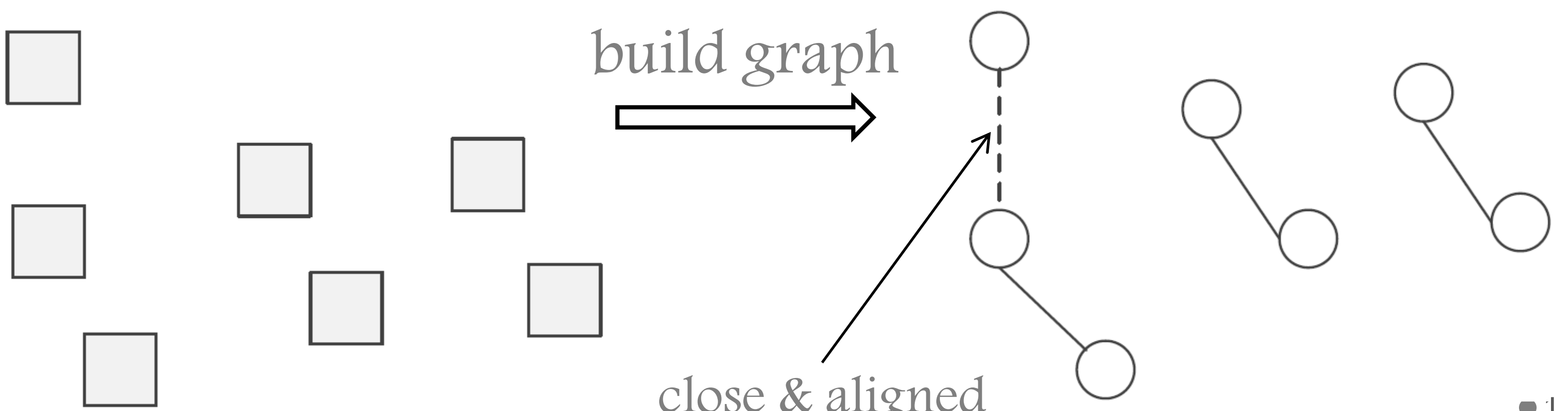
- DSA with double patterning (2 masks)
- DSA with triple patterning (3 masks)
- simultaneously perform grouping (into the same template) and mask assignment

Outline

- ❑ Introduction to DSA & Multiple Patterning
- ❑ Problem Formulation
- ❑ DSA with Double Patterning
- ❑ DSA with Triple Patterning
- ❑ Results & Conclusions

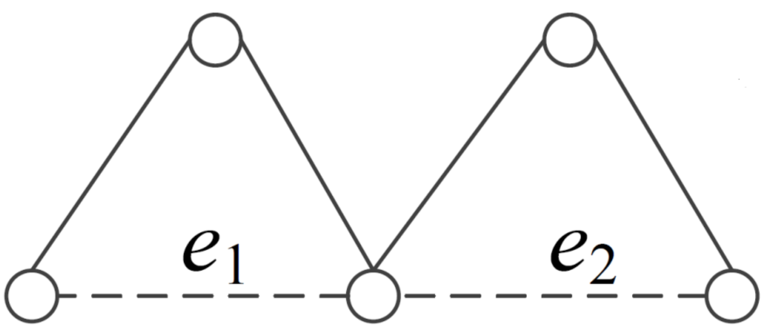
Conflict graph $G(V, E_g \cup E_n)$

- nodes: vias (*V*)
- edges: inter-distance < *threshold*
  - dash edges: **grouping** edges (E_g)
    - two vias can be grouped (into the same template)
    - only horizontally/vertically aligned vias can be grouped
  - solid edges : **non-grouping** edges (E_n)



Incompatible grouping

- a set of grouping edges are incompatible if they cannot be used for grouping at the same time
- when *max_template_size* = 2 (single-hole or 2-hole templates only), any adjacent grouping edges are incompatible

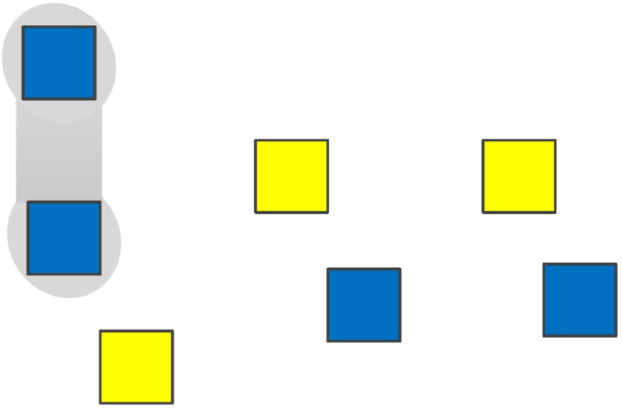


- e_1 and e_2 are incompatible

Problem definition

- Given a layout of contacts/vias and *N* masks, assign the features to different masks and group some of the features

- such that
  - no incompatible groupings happen
  - minimize # of URCs (unresolved conflicts)
  - minimize total cost of the used DSA templates



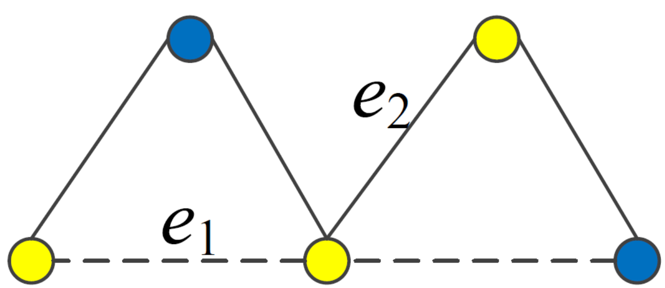
Outline

- Introduction to DSA & Multiple Patterning
- Problem Formulation
- DSA with Double Patterning
- DSA with Triple Patterning
- Results & Conclusions

Simplified graph problem

- $G(V, E_g \cup E_n)$
- set edge cost ← $cost(e) = \begin{cases} Template\_cost(e), if\, e \in \text{grouping} \\ URCcost, if\ e \notin \text{grouping} \end{cases}$
- set incompatible constraints

- N-color (2 or 3-color) graph and minimize cost of invalid edges (edges whose end-nodes have the same color)

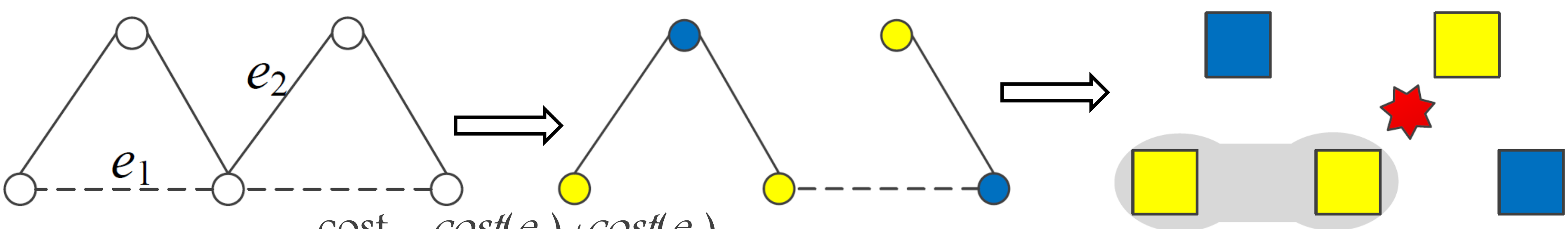


cost = $cost(e_1)+cost(e_2)$

Simplified graph problem

- *N*-color (2 or 3-color) graph and minimize cost of invalid edges

- **Simplified Graph Problem:**
  delete min-cost edges to make remaining graph *N*-colorable



cost = $cost(e_1) + cost(e_2)$

- If a grouping edge (a,b) is removed: a and b are grouped
- If a non-grouping edge (a,b) is removed: a and b has a URC

Dividing conflict graph

- identify *bridges, independent components*… to divide conflict graph into subgraphs

- in highly dense layout, more than 99.9% of the subgraphs are planar

- first solve planar subgraphs
- non-planar graphs can be modified to planar

- first assume *max_template_size* = 2

Exact method for Simplified Graph Problem

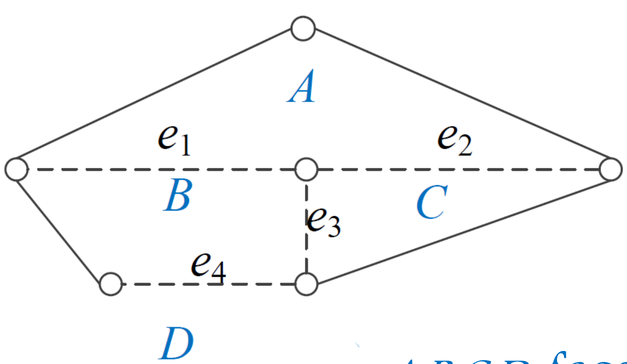
2-colorable

- Constrained Edge Deletion Bipartite Problem

a graph is 2-colorable iff no odd cycle

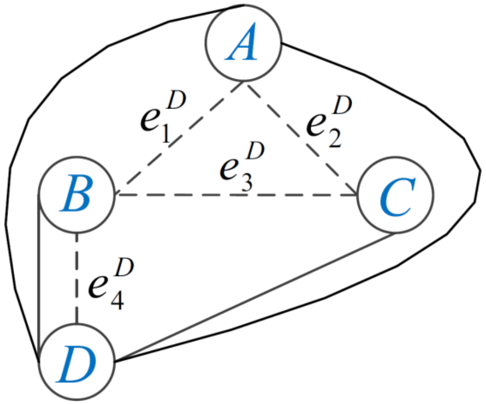
- Constrained Odd Vertex Pairing (OVP) on dual graph

OVP: remove edge to make graph w/o odd-degree vertex



A B C D: faces

dual graph



Exact method for Simplified Graph Problem

- Constrained OVP

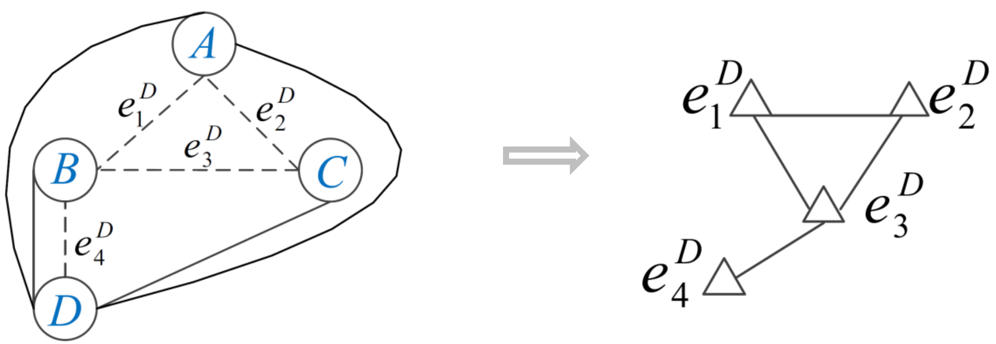
⇩

- a set of subproblems

each subproblem is OVP w/o constraints

Finding all subproblems

- edge constraint graph $\hat{G}(\hat{V}, \hat{E})$



- all subproblems $\Longrightarrow$ all maximal independent sets of $\hat{G}$

Maximal Independent Set: adding any element will make it not independent

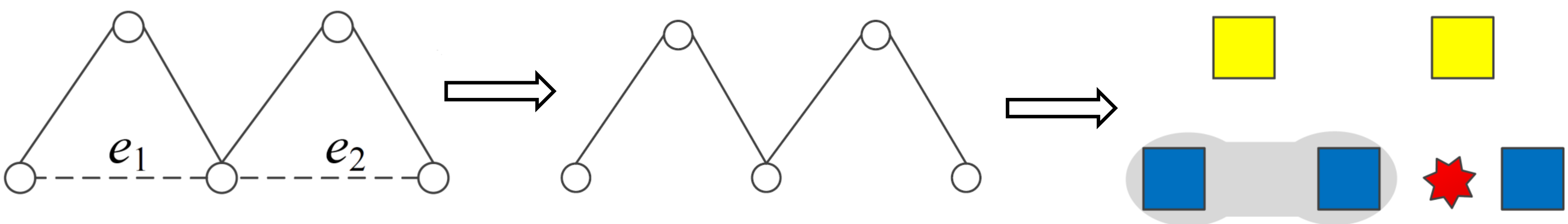
Algorithm

- for each maximal independent set I
  - modify weights of grouping edges $\notin I$ as ∞
  - call OVP solver
  - edges $\notin I$ will not be removed
  - edges $\in I$ may or may not be removed
- it is guaranteed
  - removed grouping edges are always compatible
  - the minimum cost result returned by OVP solver is optimal

Accurate formulation on graph

- Simplified: If a grouping edge (*a,b*) is removed:
  a and *b* are grouped

- Accurate: If a grouping edge (*a,b*) is removed:
  a and *b* are grouped
  or *a* and *b* has a URC (2 vias that can be grouped can also have a URC)

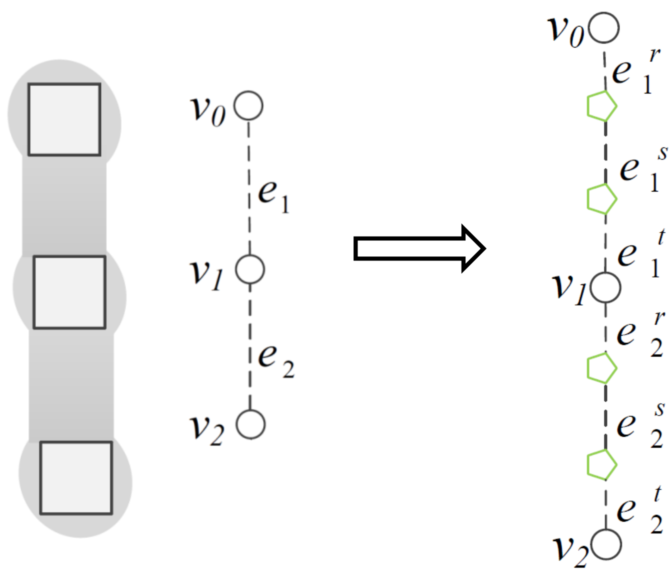


- reduce from accurate to simplified by adding dummy nodes/edges

Extension to larger templates

- *max_template_size* > 2

- Problem: how to calculate cost of larger templates accurately?
- Solution: add dummy nodes and edges to graph



Outline

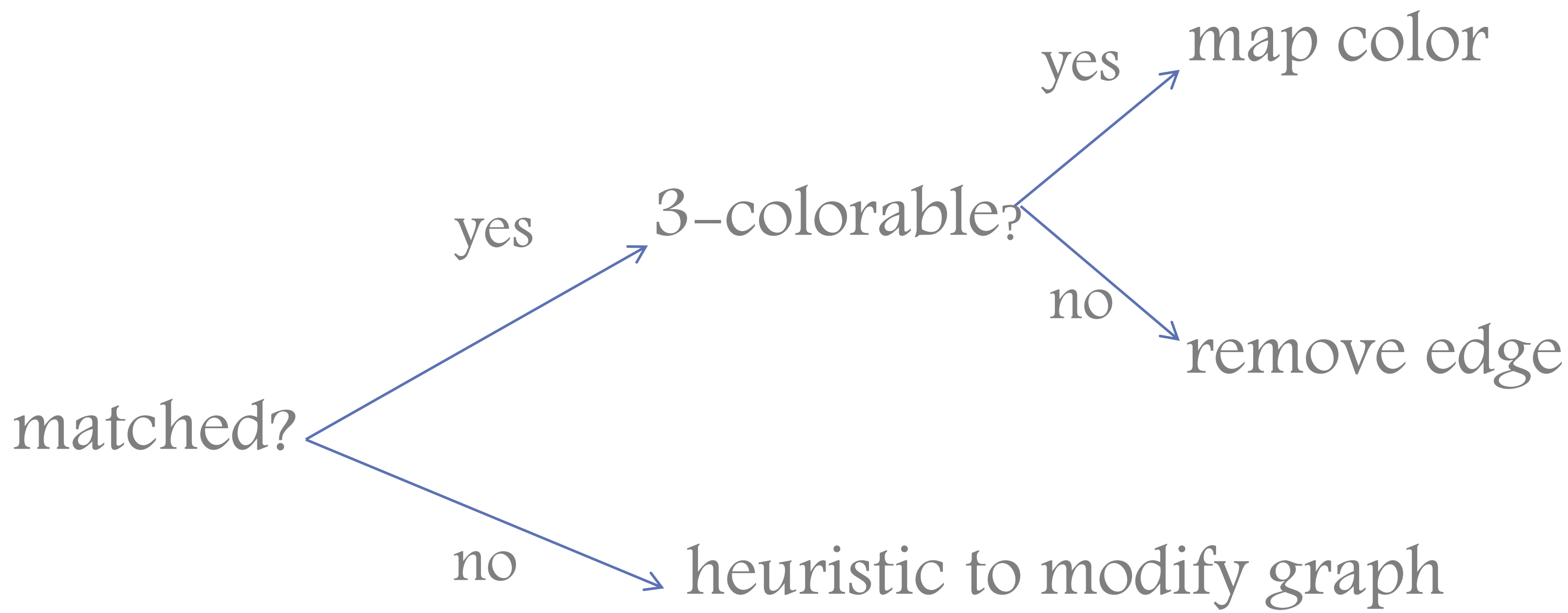
- ❑ Introduction to DSA & Multiple Patterning
- ❑ Problem Formulation
- ❑ DSA with Double Patterning
- ❑ DSA with Triple Patterning
- ❑ Results & Conclusions

LUT based method

- Problem: remove edge to make graph 3-colorable

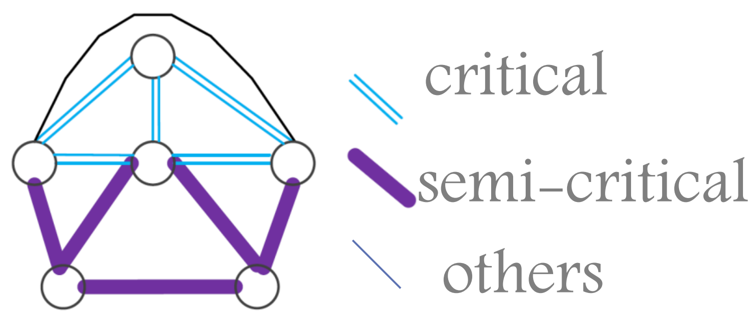
- divide the conflict graph to **subgraphs**
- build LUT of **small graphs** ⟺ match

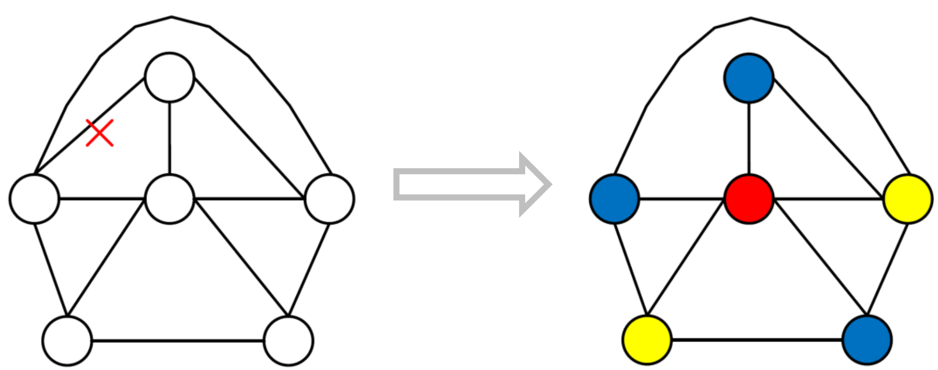
Match with LUT



Critical /Semi-critical edge

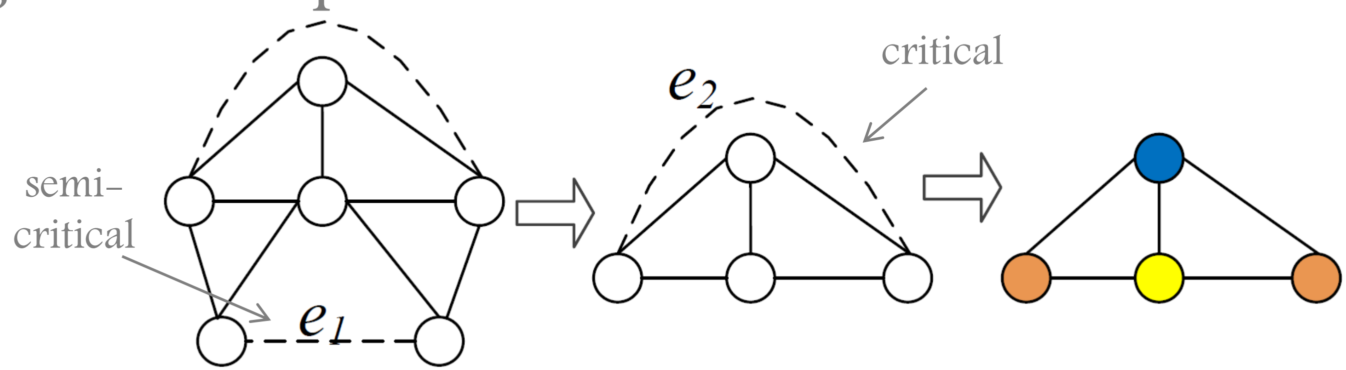
- for each non-3-colorable graph in LUT
  - critical edge: its removal make a graph 3-colorable
  - semi-critical edge: after removing it, there will be removable (degree < 3) nodes





Algorithm for removing edges

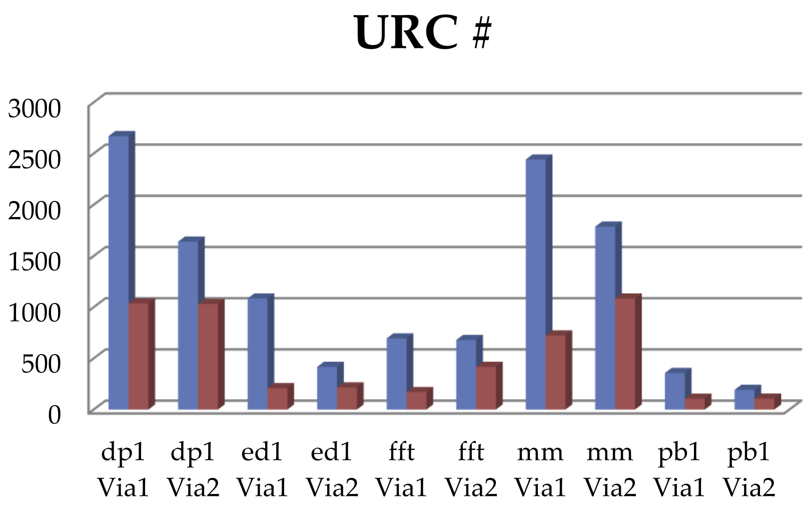
- *e*=min-cost **critical edge**
  - *e* is grouping edge:
    - return removing *e* (optimal)
  - *e* is non-grouping edge **or** *e*=null:
    - try to remove each **semi-critical** edge *se*
    - simplify graph and recursively call the algorithm
    - get a set of possible solutions and find min-cost one

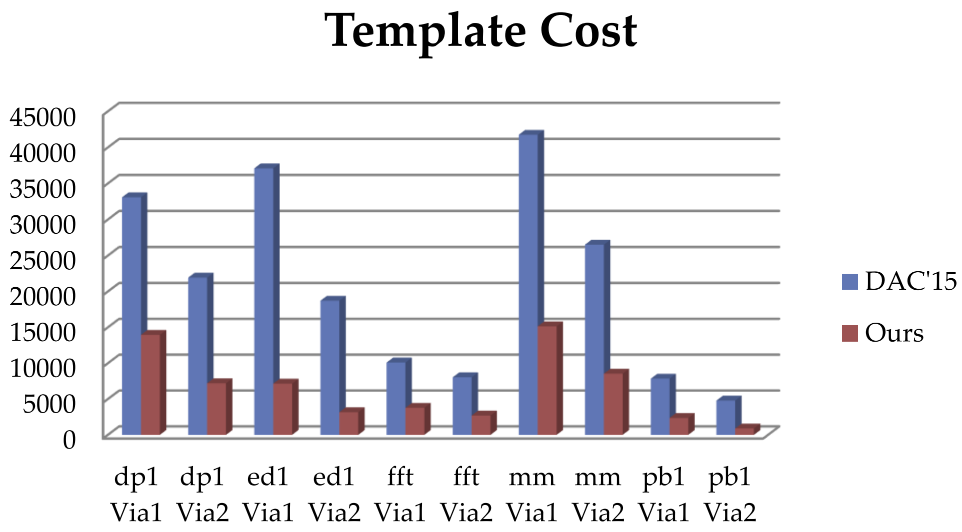


Outline

- Introduction to DSA & Multiple Patterning
- Problem Formulation
- DSA with Double Patterning
- DSA with Triple Patterning
- Results & Conclusions

Results for DSA + Double Patterning

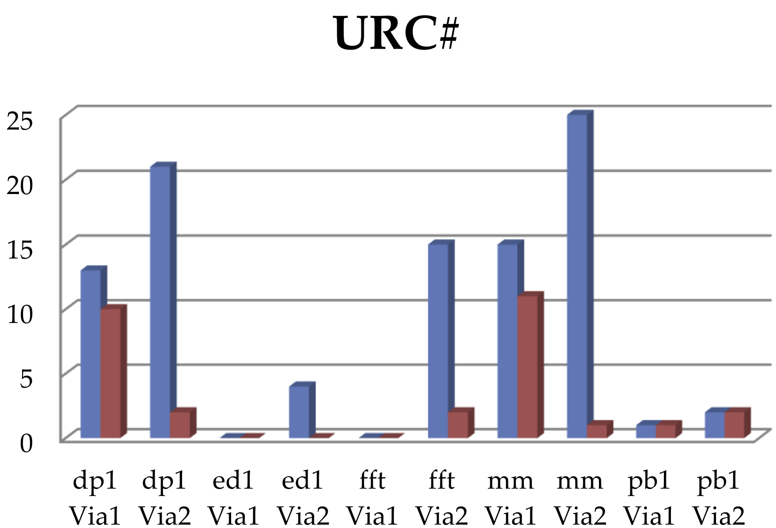


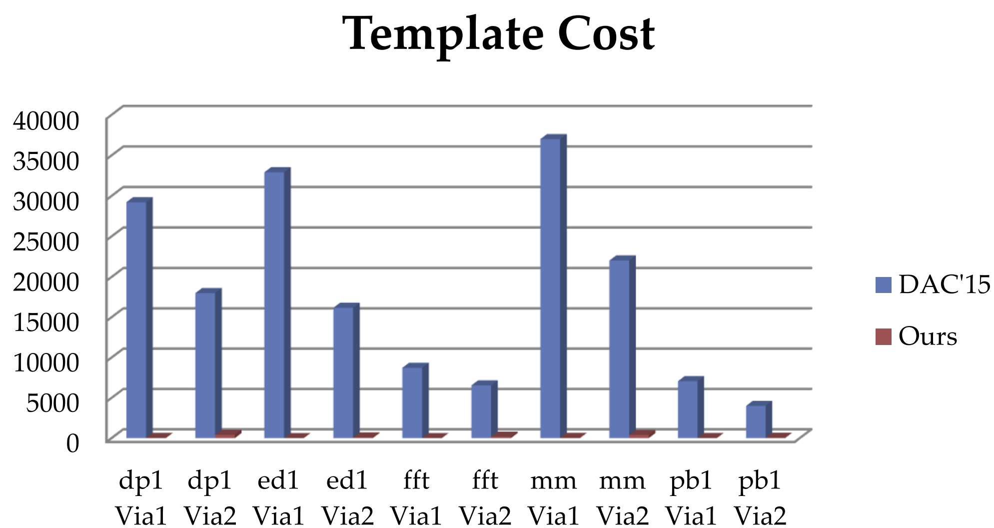


- target at **10nm** Via1 and Via2
- significantly reduce URC# and Template Cost
- resolve 400,000 vias in **2.8 seconds**
- verified by ILP: our result is **optimal for planar graphs**

Y. Badr, et al., "Mask assignment and synthesis of DSA-MP hybrid lithography for sub-7nm contacts/vias," in *Proc. DAC*, June 2015.

Results for DSA + Triple Patterning





- reduction on Template Cost is even larger
- resolve 400,000 vias within 3.6 seconds

Y. Badr, et al., "Mask assignment and synthesis of DSA-MP hybrid lithography for sub-7nm contacts/vias," in *Proc. DAC*, June 2015.

Conclusions

- DSA + Double/Triple Patterning
- Co-optimize template cost and mask assignment
- Reduce manufacturing cost remarkably

Thanks!