Automated Layout and Phase Assignment for Dark Field PSM

P. Berman, Penn State

A. B. Kahng, UCLA

D. Vidhani, UCLA

H. Wang, UCLA

A. Zelikovsky, GSU

Supported by a grant from Cadence Design Systems, Inc.

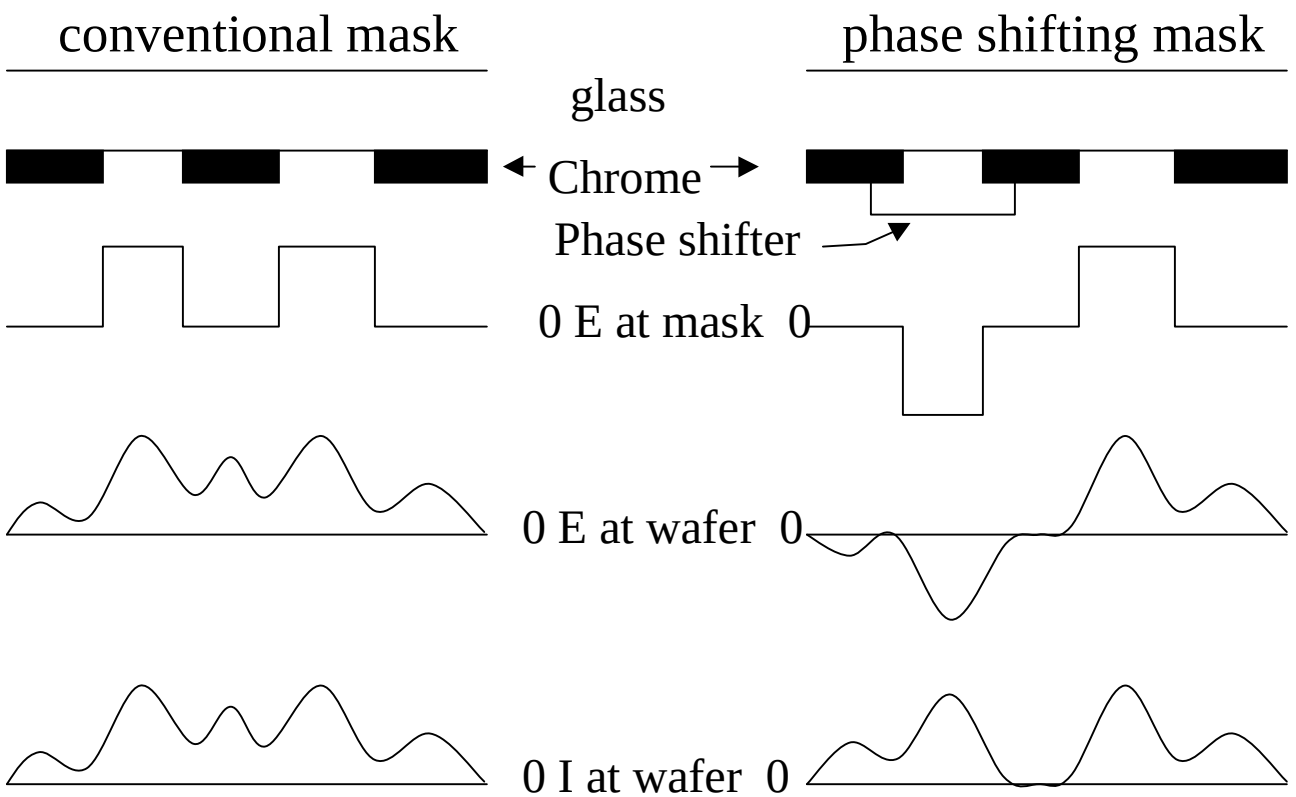
Outline

- Phase assignment for dark-field alternating PSM
- Minimum-perturbation PSM layout
  - previous work
  - reduction to the T-join problem
- Algorithms for T-join problems
- Implementation experience
- Conclusions

Outline

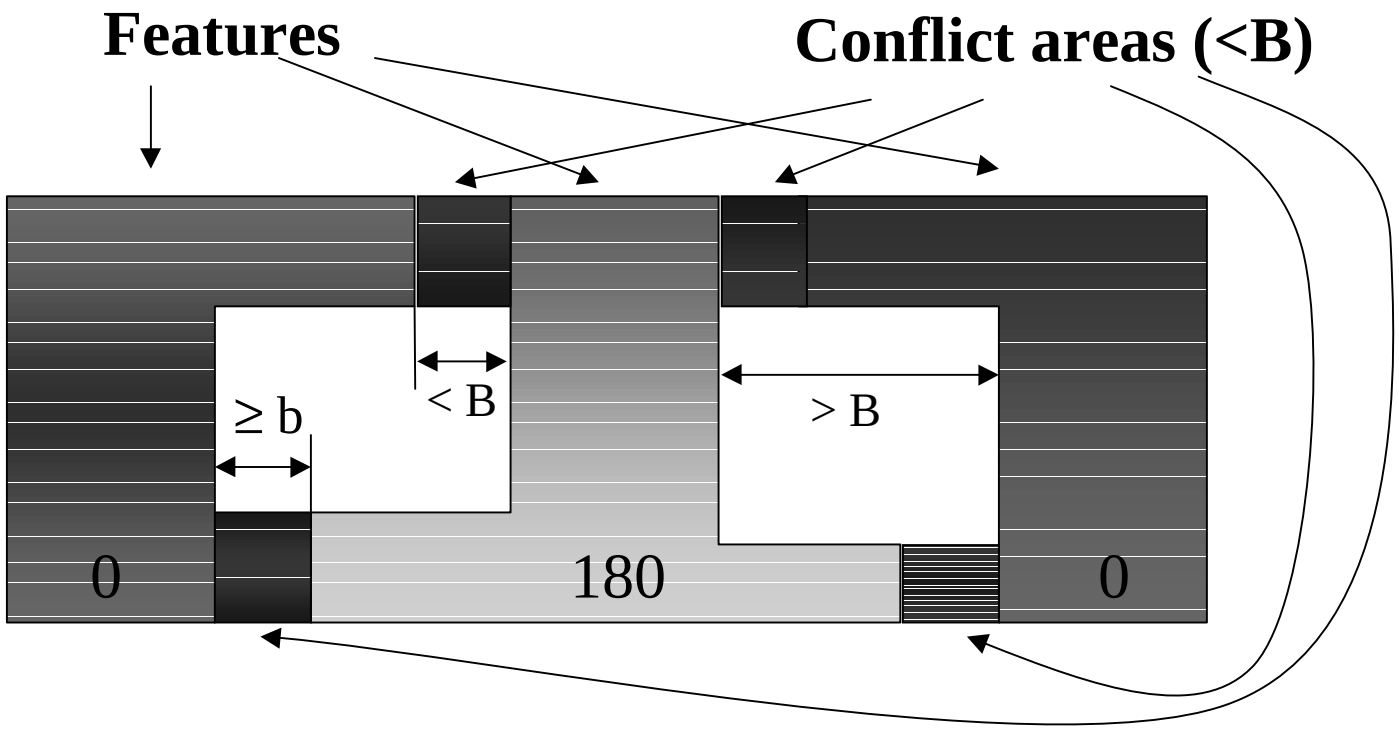
- Phase assignment for dark-field alternating PSM
- Minimum-perturbation PSM layout
  - previous work
  - reduction to the T-join problem
- Algorithms for T-join problems
- Implementation experience
- Conclusions

Alternating PSM



Phase Assignment Problem

Assign phases 0, 180 to all features s.t. pairs with separation < B have opposite phases



b ≡ minimum separation

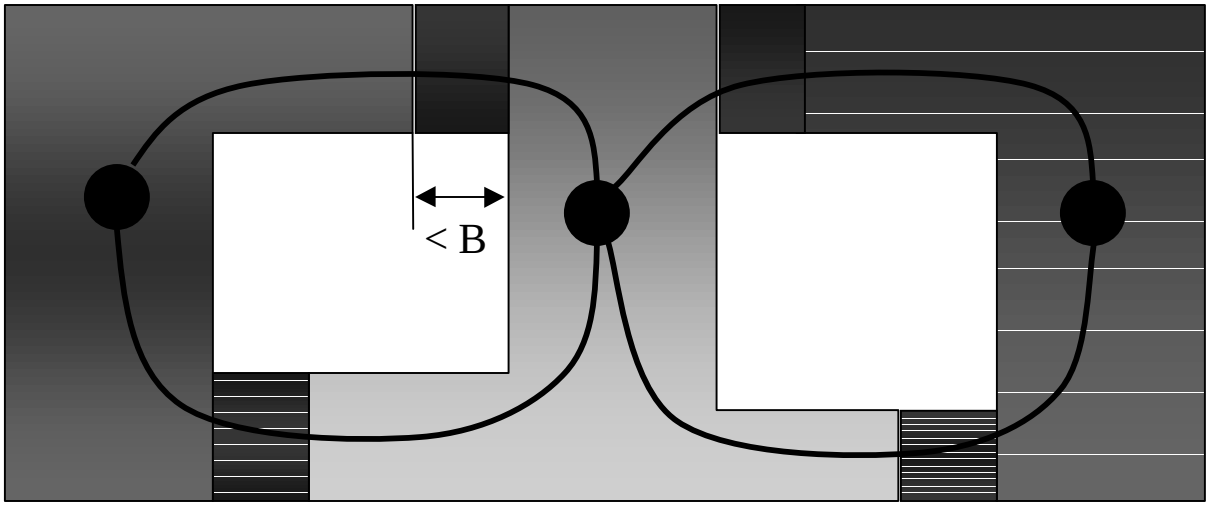
B ≡ minimum separation between same-phase features

Conflict Graph

Vertices: features

Edges: conflicts

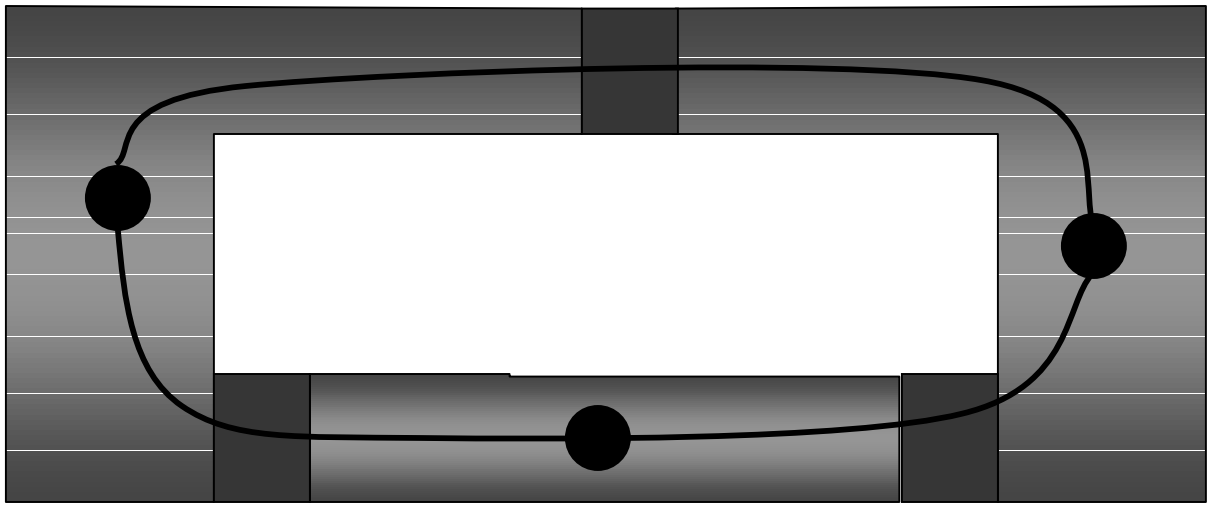
(feature pairs with separation < B)



Odd Cycles in Conflict Graph

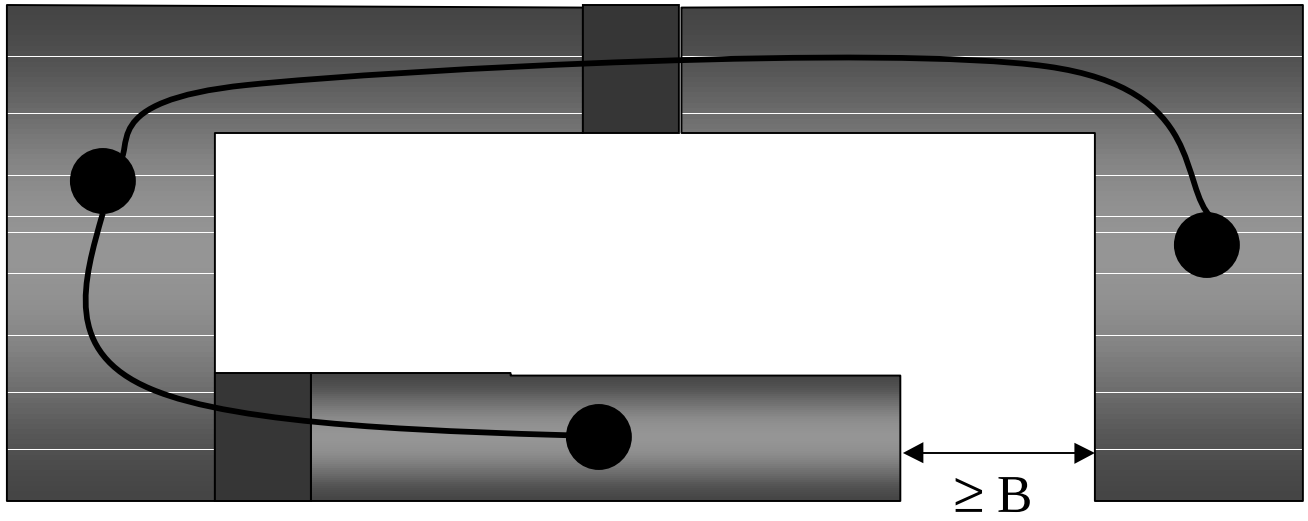
No **valid** phase assignment exists, because of odd cycle (triangle) in conflict graph

Valid assignment ≡ 2-colorable ≡ bipartite ≡ no odd cycles



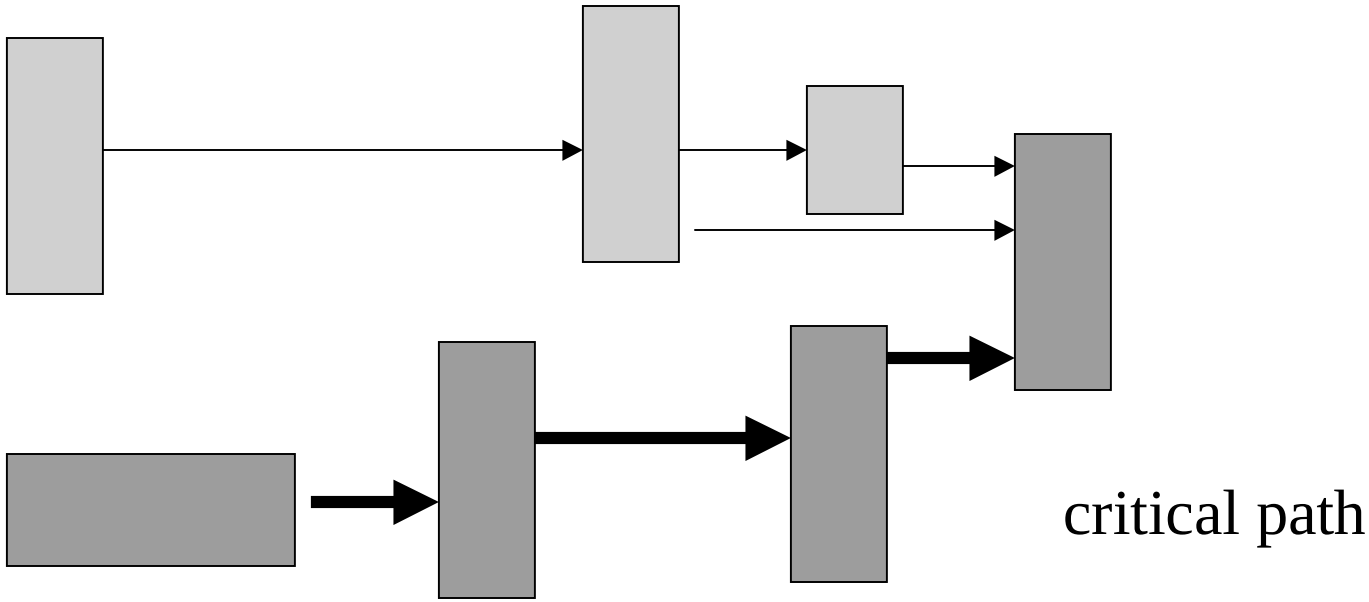
Breaking an Odd Cycle

- Perturb the layout so that a conflict edge is deleted
- Each edge deletion causes an extra min-spacing constraint



Conflict Edge Weight

- Which conflict edges are cheapest to break?
- Critical paths (e.g., in compactor) in x- and y-directions define layout area
- Conflict edges not on critical path: break for free



Outline

- Phase assignment for dark-field alternating PSM
- Minimum-perturbation PSM layout
  - previous work
  - reduction to the T-join problem
- Algorithms for T-join problems
- Implementation experience
- Conclusions

Minimum-Perturbation PSM Layout

- Input: layout in, e.g., .25um design rules
- Goal: adjust feature sizes and spacings to new PSM design rules, e.g., .15um
  - keep topology as close to original as possible
- Application to new design and to migration
  - assumes existence of a starting layout
  - hope to attain fewer violations in verification, require less manual cleanup of output layout

Compaction-Oriented Approach

- Analyze input layout
- Determine constraints for output layout
  - new PSM-induced (shape, spacing) constraints
- Compact (e.g., solve LP) with min perturbation objective
  - e.g., minimize sum of differences between old and new positions of each edge
- Key: Minimize the set of new constraints, i.e., break all odd cycles in conflict graph by deleting a minimum number of edges.

Previous Work on DF PSM Layout

- Interactive methods (Ooi et al., Moniwa et al.)
  - detect odd cycles
  - manually (in editor) widen spacing for chosen pairs
- Compaction method (Ooi et al.)
  - extract symbolic layout from mask layout
  - perform phase assignment in symbolic layout
  - apply PSM design rules to derive minimum spacings
  - compact the symbolic layout

One-Shot Phase Assignment (BACUS98)

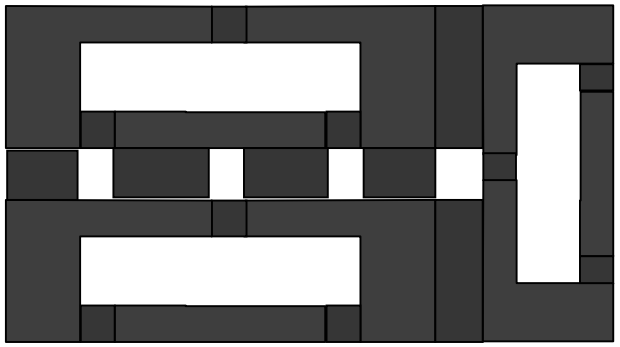
- Find conflict graph
- Find **minimum-cost** set of edges to be deleted from conflict graph for 2-colorability
- Assign phases such that only chosen conflict edges connect features of the same phase
- Compact layout with PSM design rules:
  - B-separation if features have the same phase
  - b-separation if features have different phase

The T-join Problem

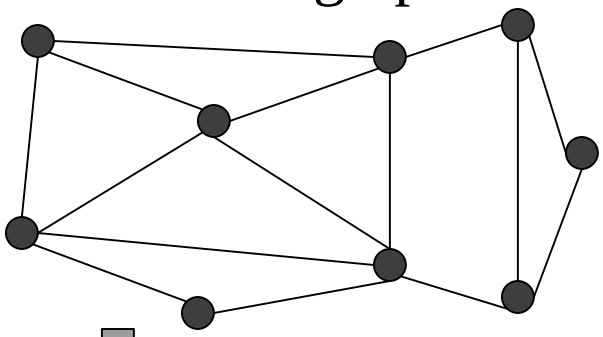
- How to delete **minimum-cost** set of edges from conflict graph G to eliminate odd cycles?
- Construct geometric dual graph D=dual(G)
- Find odd-degree vertices T in D
- Solve the **T-join problem** in D**:**
  - find min-weight edge set J in D such that
    - all T-vertices has odd degree
    - all other vertices have even degree
- Solution J corresponds to desired min-cost edge set in conflict graph G

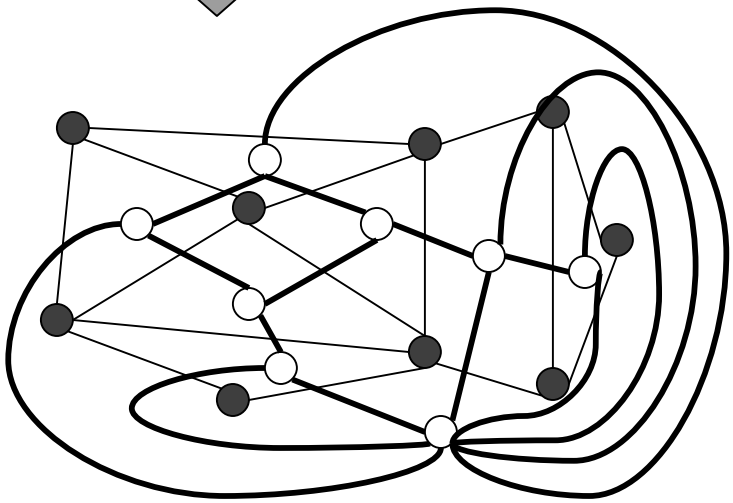
Optimal Odd Cycle Elimination

blue features/red conflicts

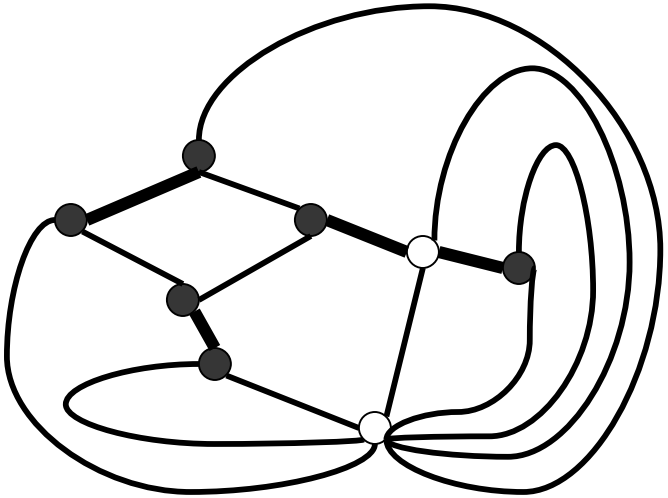


conflict graph G





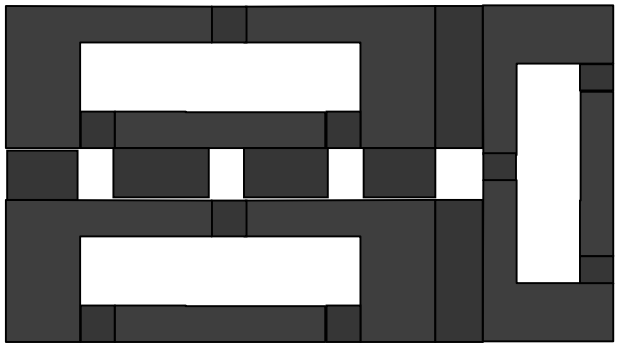
dual graph D



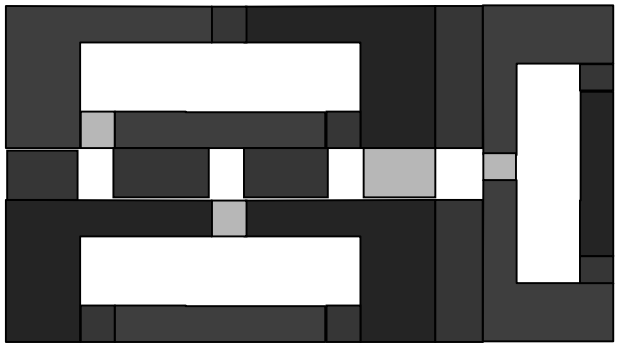
T-join odd degree nodes in D

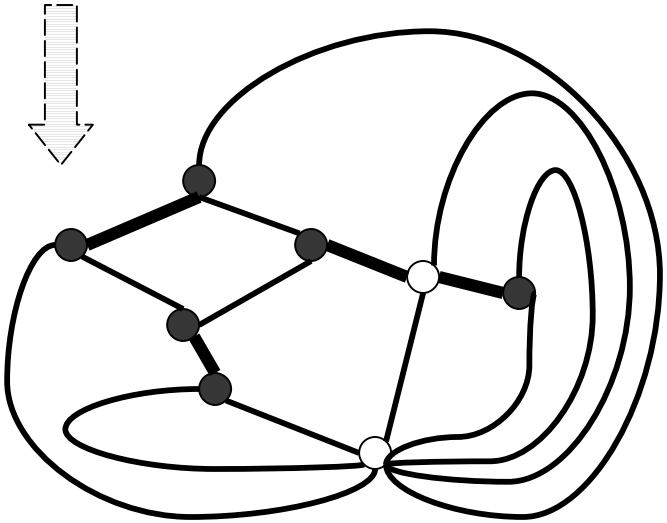
Optimal Odd Cycle Elimination

blue features/red conflicts

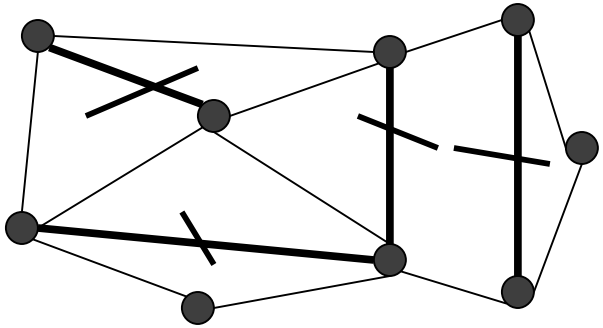


Assign phases:
only green conflicts left





T-join odd degree nodes in D



conflict graph

Outline

- Phase assignment for dark-field alternating PSM
- Minimum-perturbation PSM layout
  - previous work
  - reduction to the T-join problem
- Algorithms for T-join problems
- Implementation experience
- Conclusions

T-join Problem in Sparse Graphs

- Reduction to matching
  - construct a complete graph T(G)
    - vertices = T-vertices
    - edge costs = shortest-path cost
  - find minimum-cost perfect matching
- Typical example = sparse (not always planar) graph
  - note that conflict graphs are sparse
  - #vertices = 1,000,000
  - #edges $\approx 5 \times$ #vertices
  - # T-vertices $\approx$ 10% of #vertices = 100,000
- Drawback: finding all shortest paths too slow and memory-consuming
  - #vertices = 100,000 $\rightarrow$ #edges = 5,000,000,000

T-join Problem: Reduction to Matching

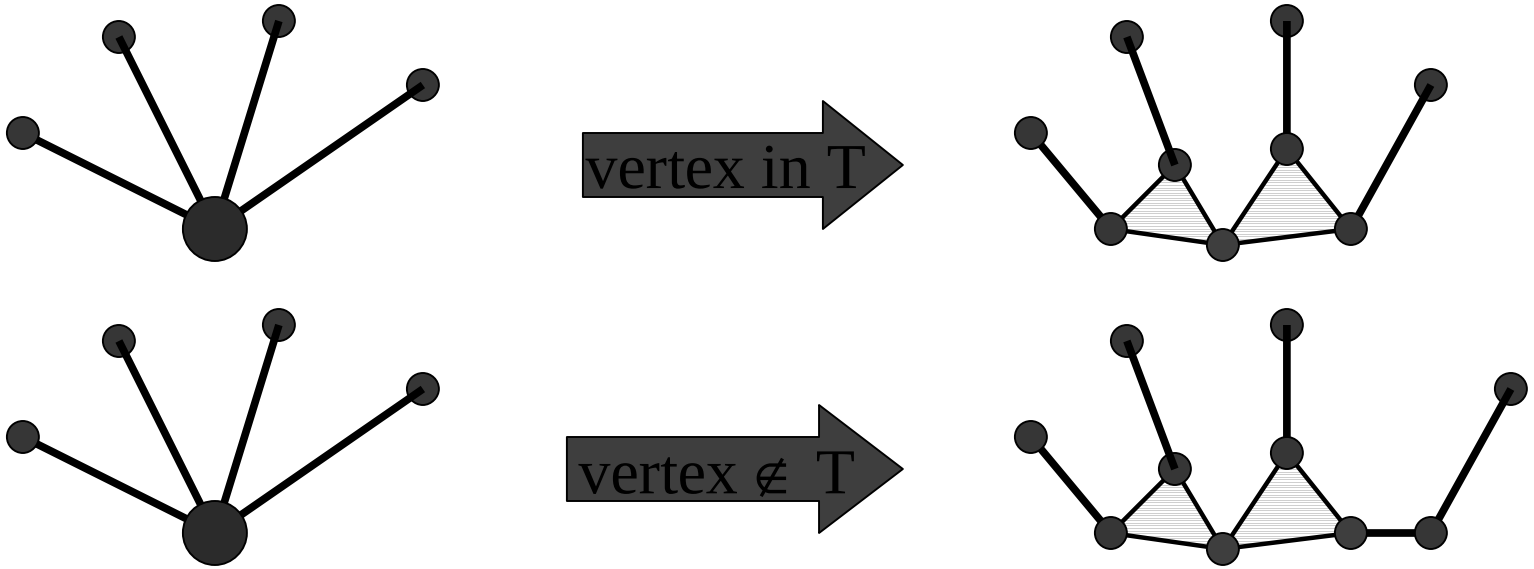
- Desirable properties of reduction to matching:
  - exact (i.e., optimal)
  - not much memory (say 2-3Xmore)
  - results in a very fast solution
- Solution: gadgets!
  - replace each edge/vertex with gadgets s.t.

    matching all vertices in gadgeted graph

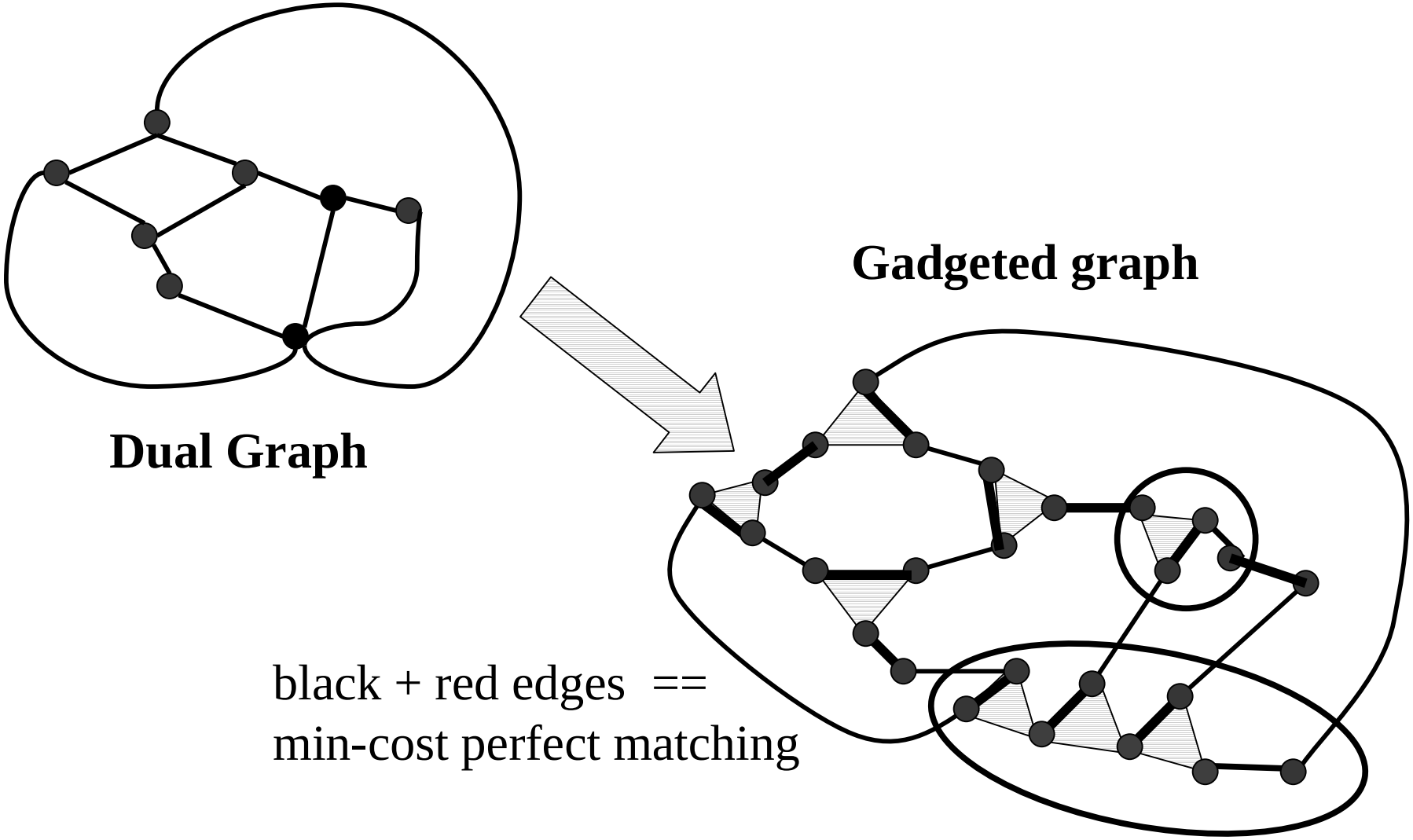
    $\Leftrightarrow$ T-join in original graph

T-join Problem: Reduction to Matching

- replace each vertex with a chain of triangles
- one more edge for T-vertices
- in graph D: m = #edges, n = #vertices, t = #T
- in gadgeted graph: 4m-2n-t vertices, 7m-5n-t edges
- cost of red edges = original dual edge costs
  cost of (black) edges in triangles = 0



Example of Gadgeted Graph



Outline

- Phase assignment for dark-field alternating PSM
- Minimum-perturbation PSM layout
  - previous work
  - reduction to the T-join problem
- Algorithms for T-join problems
- Implementation experience
- Conclusions

Data Flow

- GDSII $\rightarrow$ CIF
- CIF $\rightarrow$ internal layout representation
- New layer with phase shift $\rightarrow$ CIF

Results

	Layout1		Layout2		Layout3	
Testcase	polygons	edges	polygons	edges	polygons	edges
	3769	12442	9775	26520	18249	51402
Algorithm	edges	runtime	edges	runtime	edges	runtime
Greedy	2650	0.56	2722	3.66	6180	5.38
GW	1612	3.33	1488	5.77	3280	14.47
Exact	1468	19.88	1346	16.67	2958	74.33

- Runtimes in CPU seconds on Sun Ultra-10
- Greedy = breadth-first-search bicoloring (similar to Ooi et al.)
- GW = Goemans/Williamson95 heuristic
- Cook/Rohe98 for perfect matching

Latest improved gadgets: runtimes decrease by factor of 6

Conclusions

- New fast, optimal algorithms for minimum-cost T-join problem in sparse graphs
- Yields new approach to minimum perturbation problem; applicable to phase conflict graphs in any alternating PSM technology
- Integration with GDSII reader, polygon database, layout compactor
- More direct integrations with layout under investigation
- Preliminary results (speed, capacity) promising