High Performance Dummy Fill Insertion with Coupling and Uniformity Constraints

Yibo Lin, Bei Yu, David Z. Pan

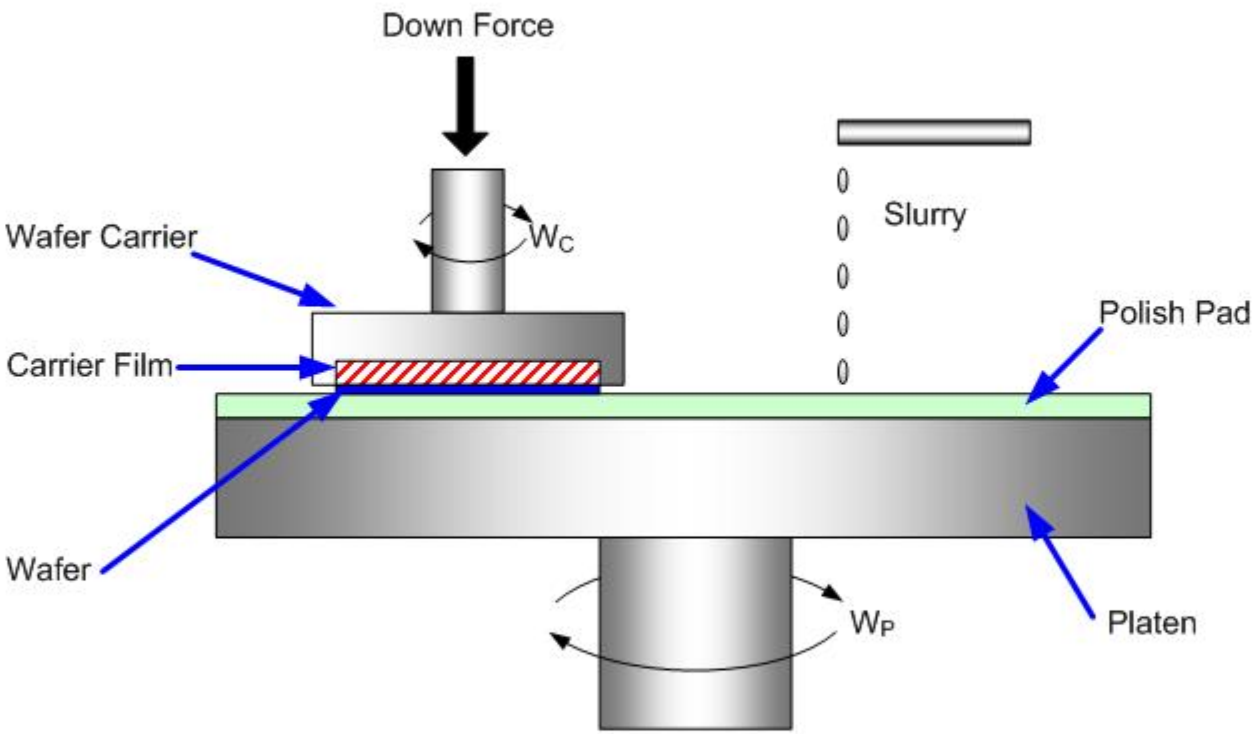
Electrical and Computer Engineering

University of Texas at Austin

Outline

- Introduction
- Problem Formulation
- Algorithms
- Experimental Results
- Conclusion

Chemical Mechanical Polishing (CMP)

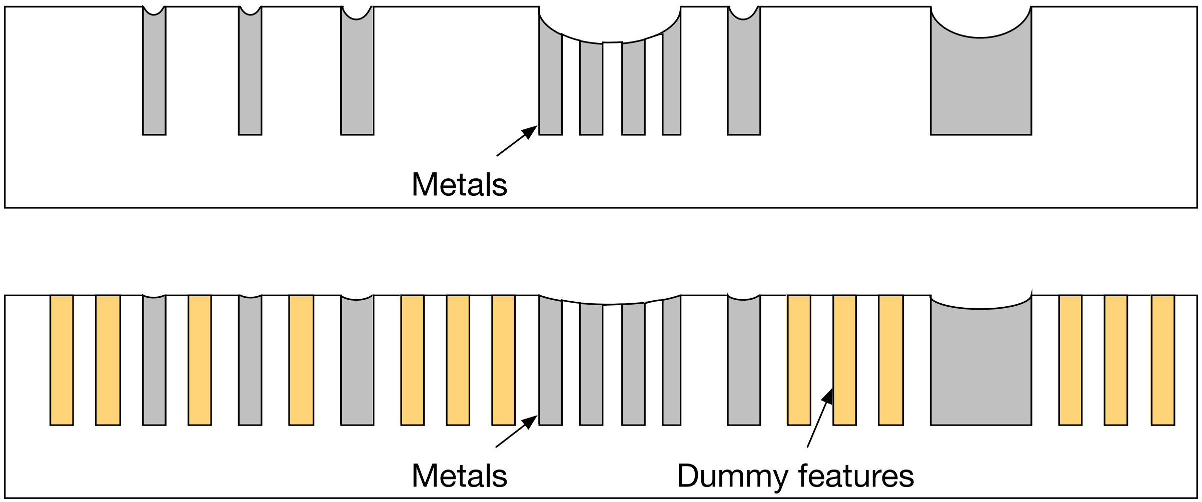


Example of CMP [source: www.ntu.edu.sg]

Uniformity

- Layout uniformity for CMP

Uneven density distribution



Even distribution

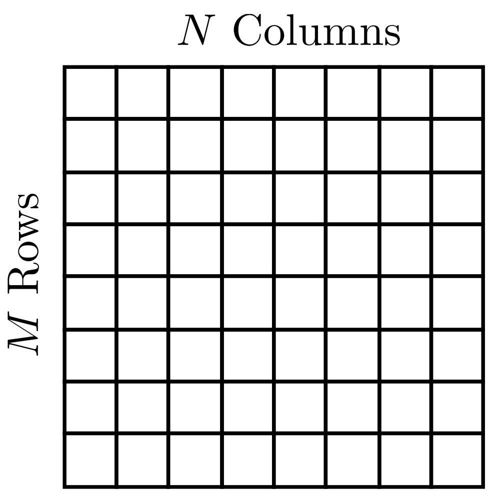
- Coupling capacitance

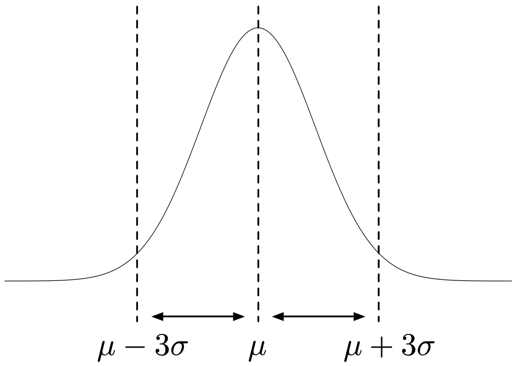
Related Works

- Minimize density variation and number of fills
  - Linear Programming (LP)
    - [Kahng+,TCAD'99]
    - [Tian+, TCAD'01]
    - [Xiang+, TCAD'08]
  - Monte Carlo and heuristic approaches
    - [Chen+, ASPDAC'00]
    - [Chen+, DAC'00]
    - [Wong+, ISQED'05]
- Minimize density variation with coupling capacitance constraints
  - ILP
    - [Chen+, DAC'03], [Xiang+, ISPD'07]

Holistic Metrics for Uniformity

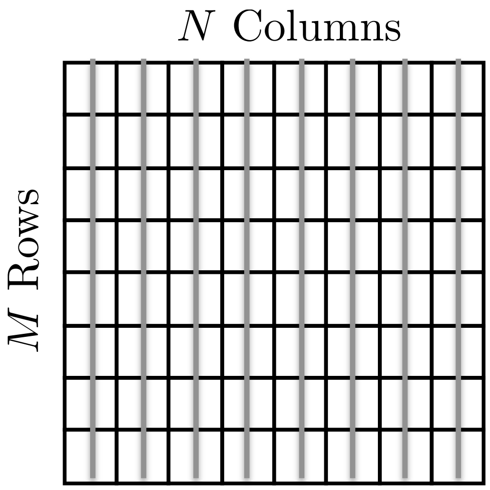
- Holistic metrics for layout uniformity from IBM (ICCAD 2014 Contest)
  - Variation (standard deviation)
  - Line hotspots
  - Outlier hotspots

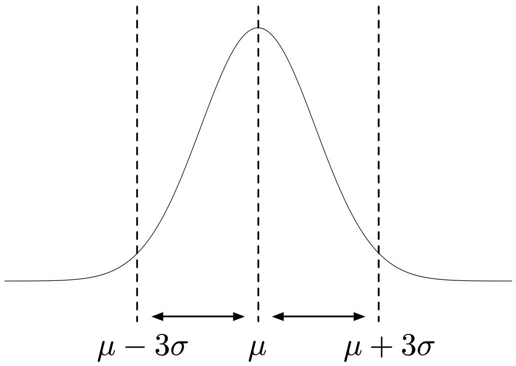




Holistic Metrics for Uniformity

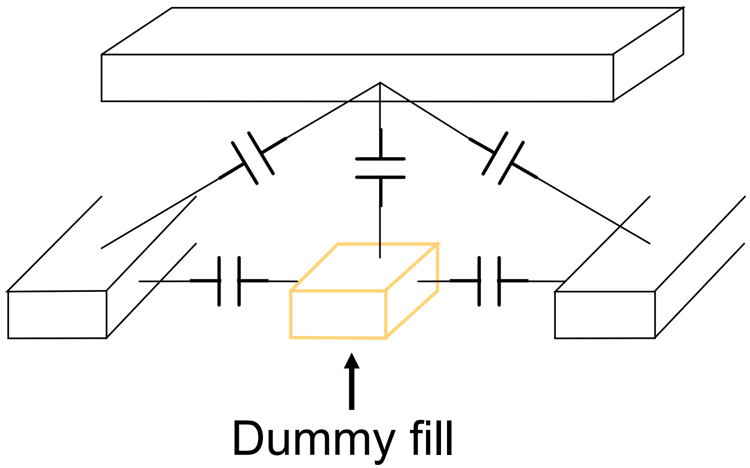
- Holistic metrics for layout uniformity from IBM (ICCAD 2014 Contest)
  - Variation (standard deviation)
  - Line hotspots
  - Outlier hotspots

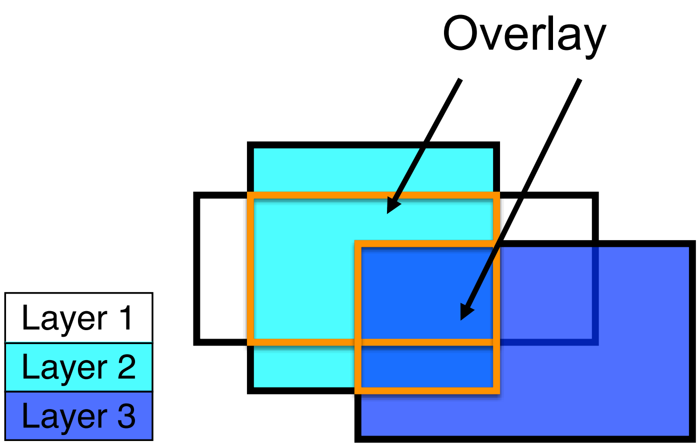




Metrics for Coupling Capacitance

- Coupling capacitance
  - Minimize overlay between layers





Problem Formulation

Based on the ICCAD 2014 contest

- Input
  - Layout with fill insertion regions
  - Signal wire density information across each window
- Quality score
  - Overlay area (20%)
  - Variation/std. dev. (20%)
  - Line hotspot (20%)
  - Outlier hotspot (15%)
  - File size for dummy fill insertion (5%)

Normalization function

$$f(x) = \max(0, 1 - \frac{x}{\beta})$$

The **higher** score, the **better**

- Overall score
  - Quality score (80%)
  - Runtime (15%)
  - Memory usage (5%)
- Output
  - Dummy fill positions and dimensions with **maximum** quality score

Outline

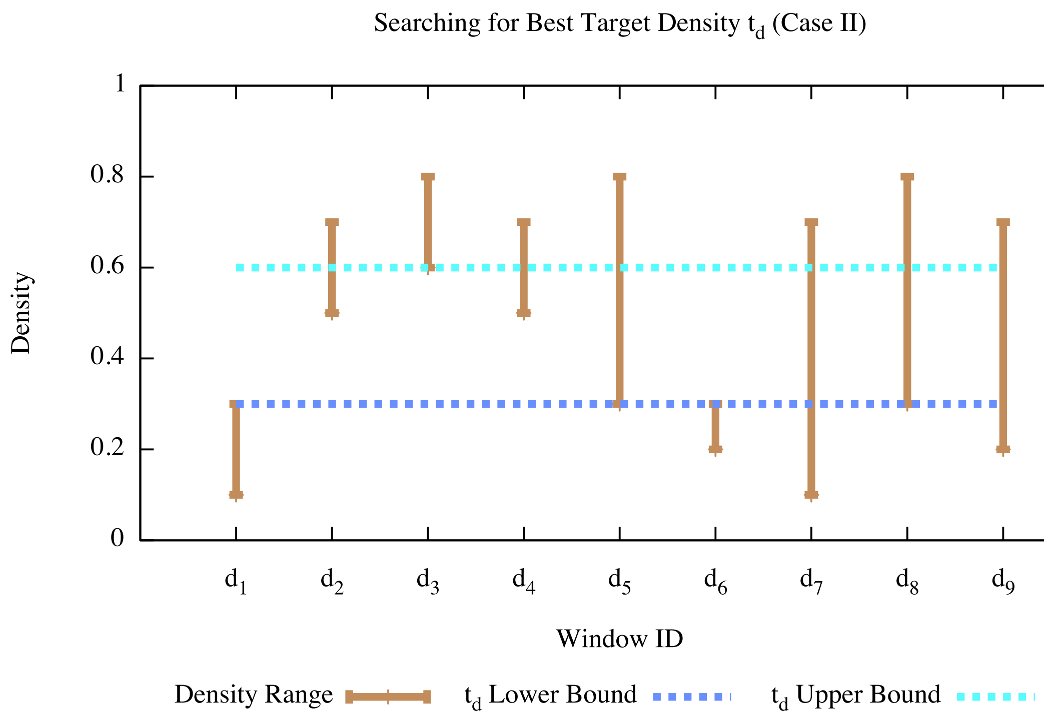
- Introduction
- Problem Formulation
- **<u>Algorithms</u>**
- Experimental Results
- Conclusion

Step 1: Density Planning

d_1	d_2	d_3
d_4	d_5	d_6
d_7	d_8	d_9

- Given density ranges of each window
- Find target density t_d for each window
- Maximize density scores

Linear scan with a small step to find best target density

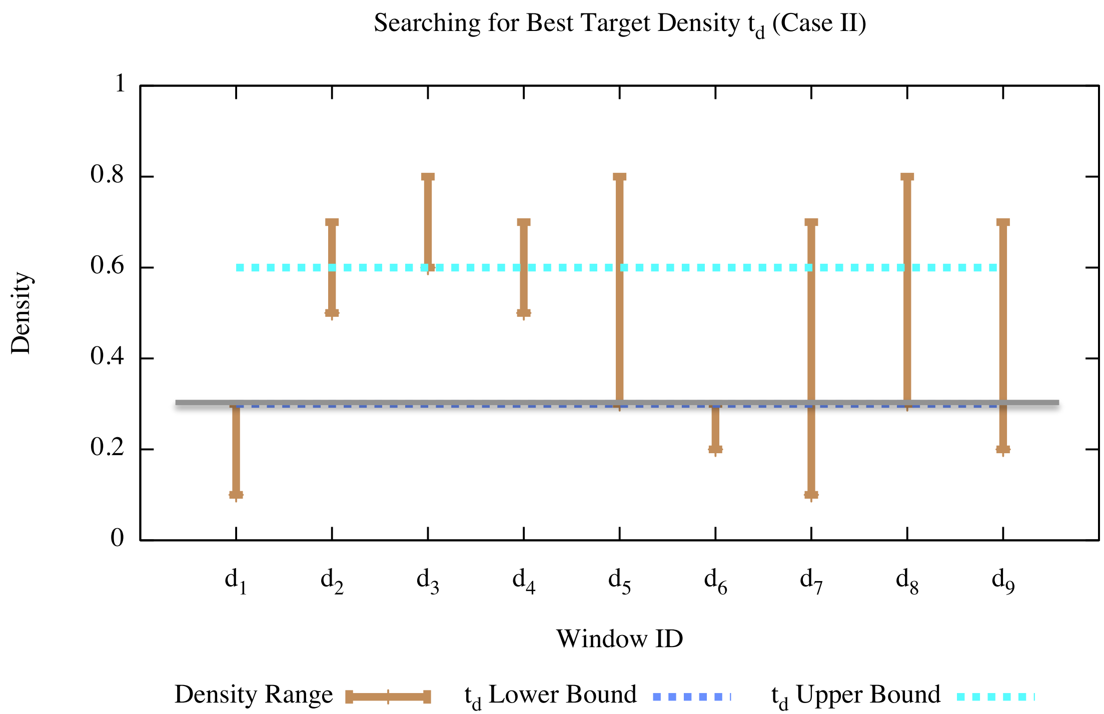


Step 1: Density Planning

0.3	0.5	0.6
0.5	0.4	0.4
0.4	0.4	0.4

- Given density ranges of each window
- Find target density t_d for each window
- Maximize density scores

Linear scan with a small step to find best target density

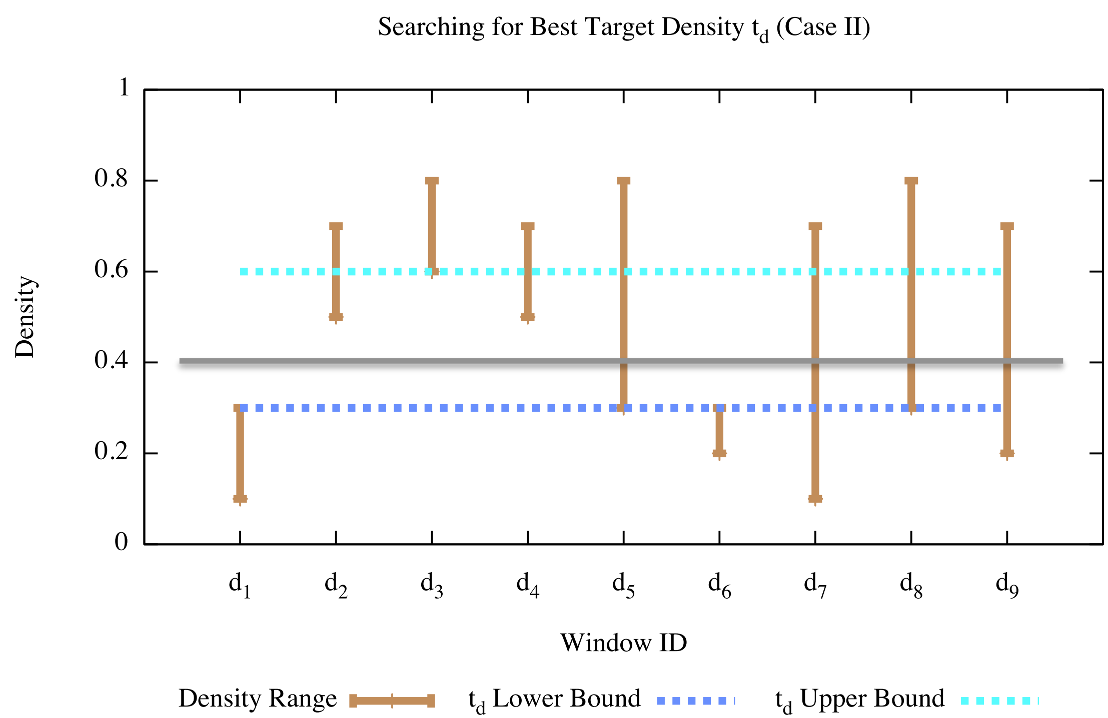


Step 1: Density Planning

0.3	0.5	0.6
0.5	0.4	0.4
0.4	0.4	0.4

- Given density ranges of each window
- Find target density t_d for each window
- Maximize density scores

Linear scan with a small step to find best target density

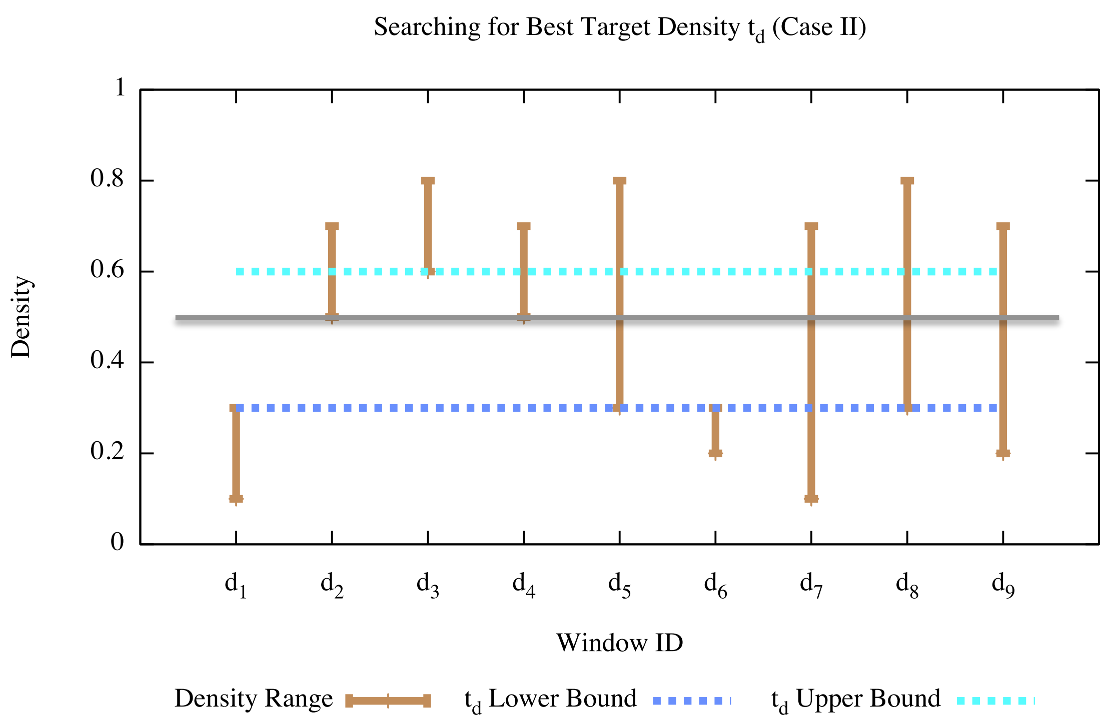


Step 1: Density Planning

0.3	0.5	0.6
0.5	0.4	0.4
0.4	0.4	0.4

- Given density ranges of each window
- Find target density t_d for each window
- Maximize density scores

Linear scan with a small step to find best target density

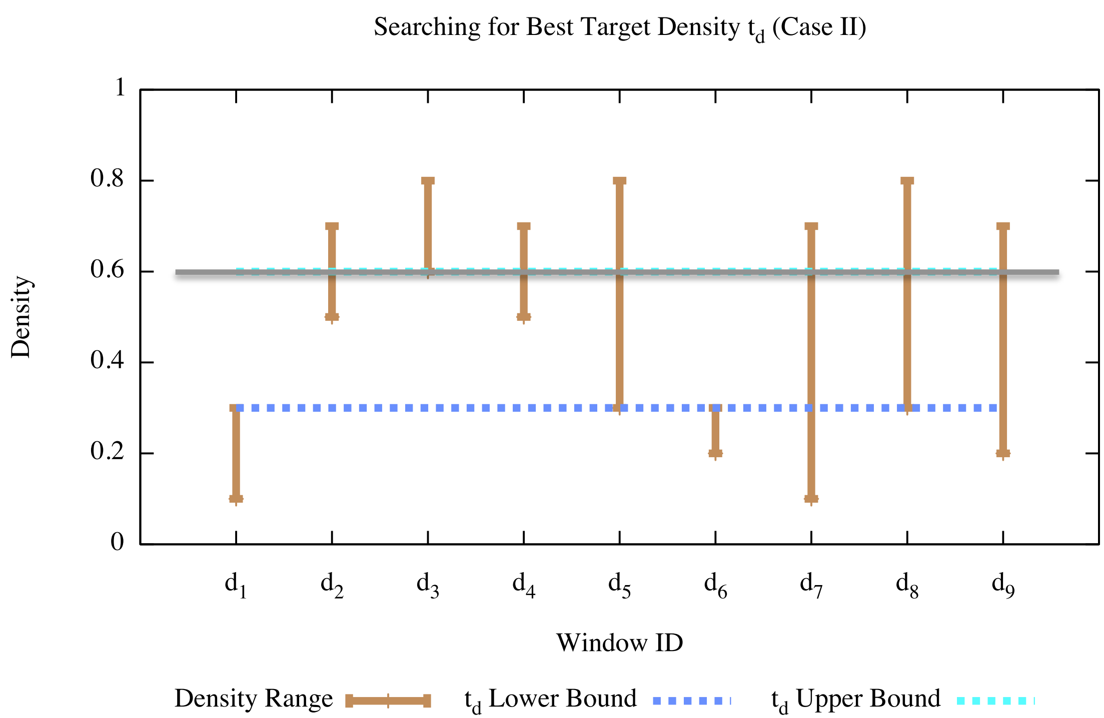


Step 1: Density Planning

0.3	0.5	0.6
0.5	0.4	0.4
0.4	0.4	0.4

- Given density ranges of each window
- Find target density t_d for each window
- Maximize density scores

Linear scan with a small step to find best target density

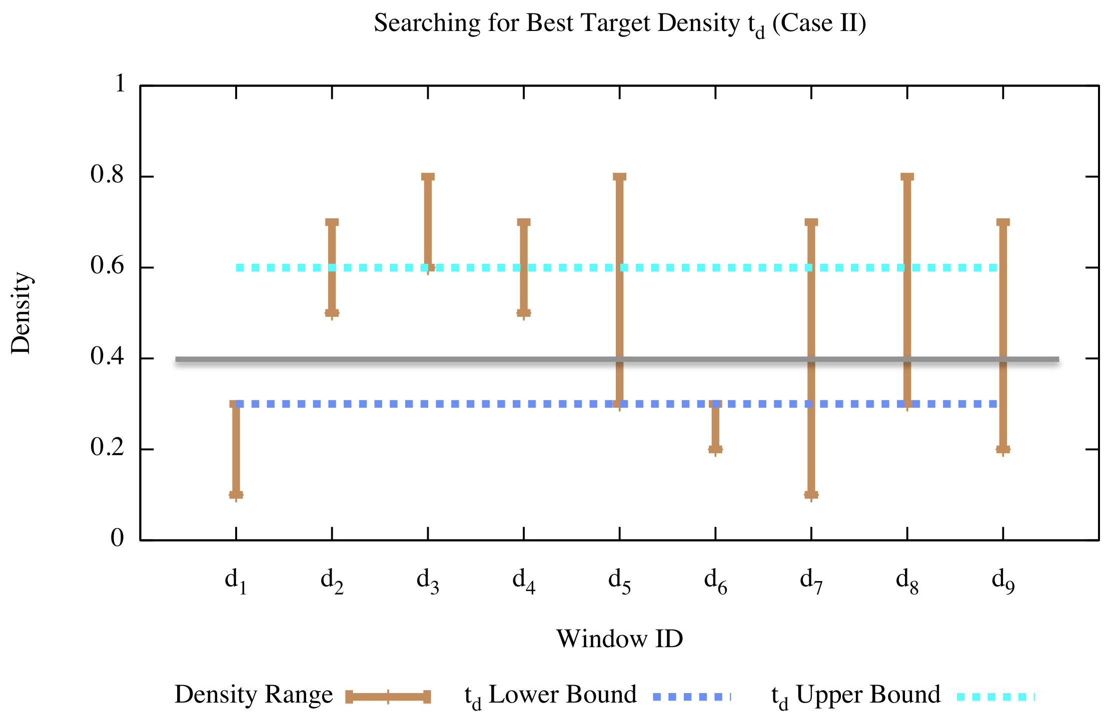


Step 1: Density Planning

0.3	0.5	0.6
0.5	0.4	0.4
0.4	0.4	0.4

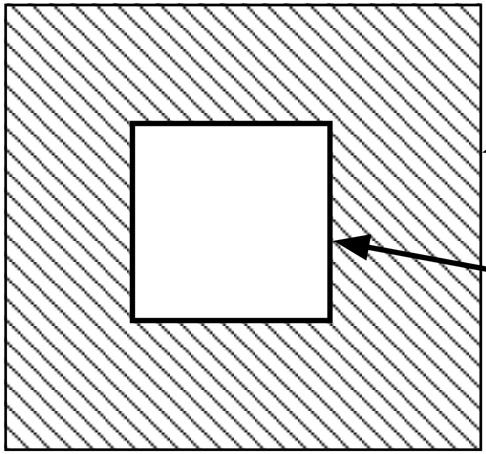
- Given density ranges of each window
- Find target density t_d for each window
- Maximize density scores

Linear scan with a small step to find best target density



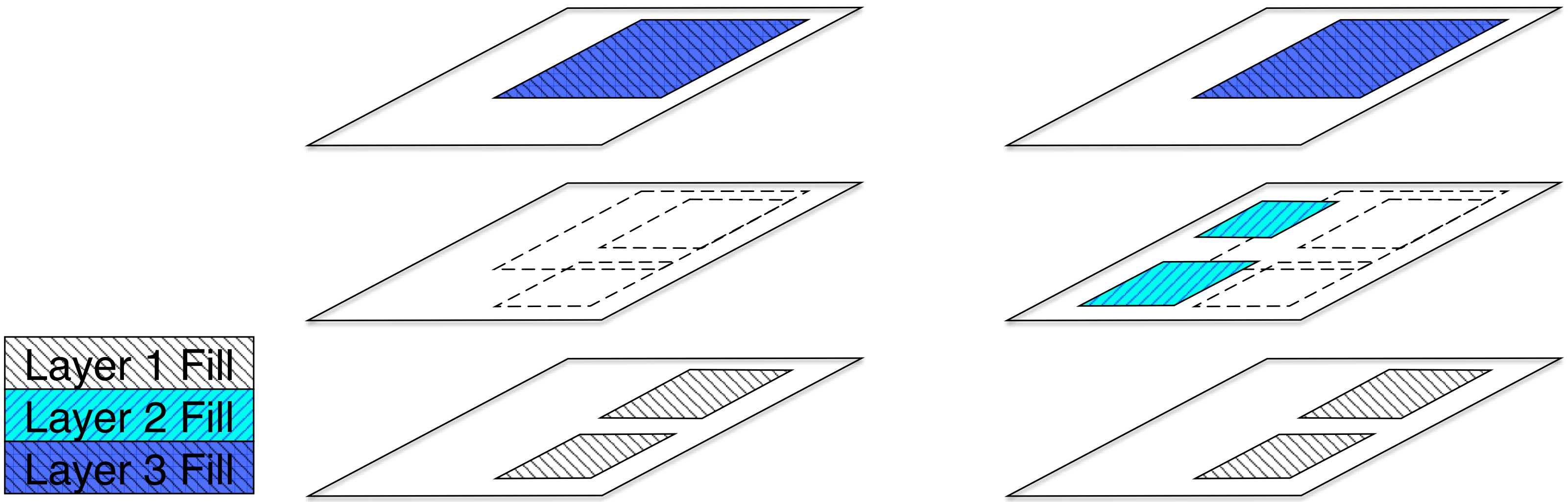
Step 2: Candidate Fill Generation

- Generate candidate fills with minimum overlay
- With the guidance of target density
- A final fill is a rectangle within a candidate fill



Step 2: Candidate Fill Generation

- Generate candidate fills with minimum overlay
- With the guidance of target density
- A final fill is a rectangle within a candidate fill

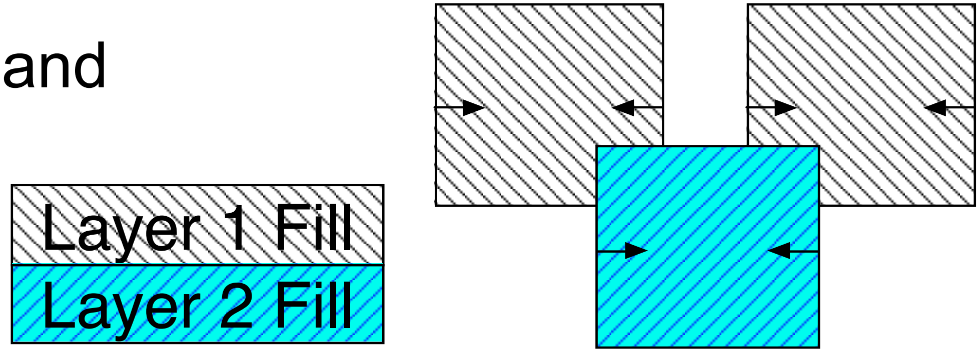


$$q = -\frac{fill\ overlay}{fill\ area} + \gamma \cdot \frac{fill\ area}{window\ area}$$

Step 3: Dummy Fill Insertion (1)

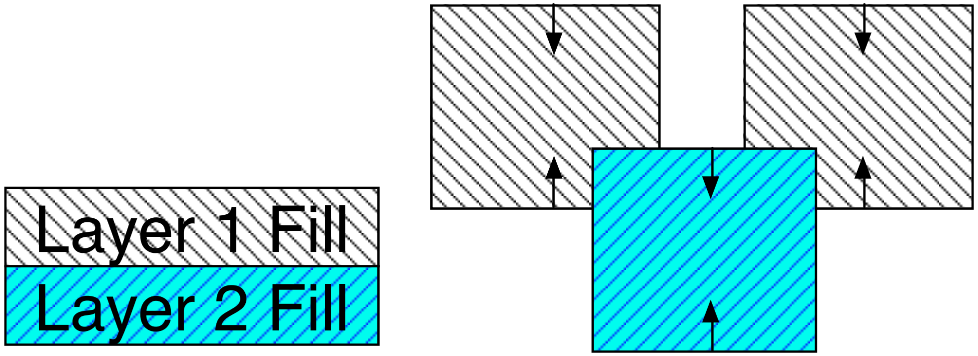
- Given a set of candidate fills
- Determine dimension of fills
- Under DRC constraints
- Minimize overlay area and density variation

Shrink in horizontal direction



- Solve in an iterative manner

Shrink in vertical direction



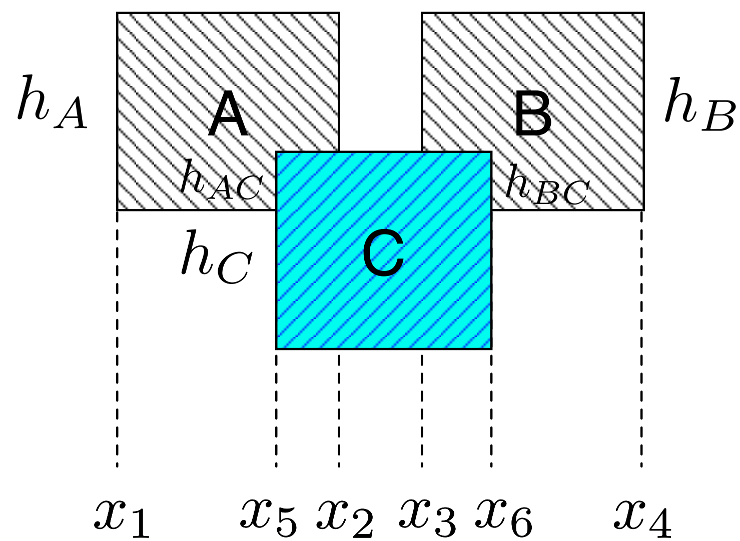
Step 3: Dummy Fill Insertion (2)

An example of the mathematical formulation in one iteration

$$
\begin{aligned}
\min \quad & \underbrace{|(x_2 - x_1)\cdot h_A + (x_4 - x_3)\cdot h_B - t_{d1}\cdot A_{win}| + |(x_6 - x_5)\cdot h_C - t_{d2}\cdot A_{win}|}_{\text{Density variation}} \\
& \underbrace{+\,(x_2 - x_5)\cdot h_{AC} + (x_6 - x_3)\cdot h_{BC}}_{\text{Overlay area}} \\
\text{s.t.} \quad & \left.\begin{array}{l}
x_2 - x_1 \geq W_{min} \\
x_4 - x_3 \geq W_{min} \\
x_6 - x_5 \geq W_{min} \\
x_3 - x_2 \geq S_{min} \\
(x_2 - x_1)\cdot h_A \geq A_{min} \\
(x_4 - x_3)\cdot h_B \geq A_{min} \\
(x_6 - x_5)\cdot h_C \geq A_{min}
\end{array}\right\}\ \text{DRC rules} \\
& \left.\begin{array}{l}
x_2 - x_5 \geq 0 \\
x_6 - x_3 \geq 0
\end{array}\right\}\ \text{Overlay constraints} \\
& l_i \leq x_i \leq u_i, \quad i = 1, 2, \ldots, 6
\end{aligned}
$$

A_{win}: area of a window
W_{min}: minimum width
S_{min}: minimum spacing
A_{min}: minimum area

Shrink in horizontal direction



Step 3: Dummy Fill Insertion (3)

$$\min \quad |(x_2 - x_1) \cdot h_A + (x_4 - x_3) \cdot h_B - t_{d1} \cdot A_{win}| + |(x_6 - x_5) \cdot h_C - t_{d2} \cdot A_{win}| + (x_2 - x_5) \cdot h_{AC} + (x_6 - x_3) \cdot h_{BC}$$

- Further relax to remove absolute operation
- Add tighter bound constraints to variables

Step 3: Dual to Min-Cost Flow

Prime

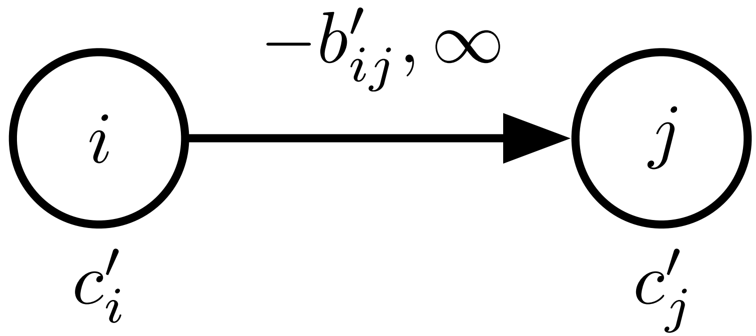
$$\min_{x_i} \sum_{i=1}^{N} c_i x_i$$

$$\text{s.t. } x_i - x_j \geq b_{ij}, \ (i,j) \in E,$$

$$l_i \leq x_i \leq u_i, \ i = 1, 2, ..., N,$$

$$x_i \in Z$$

Dual

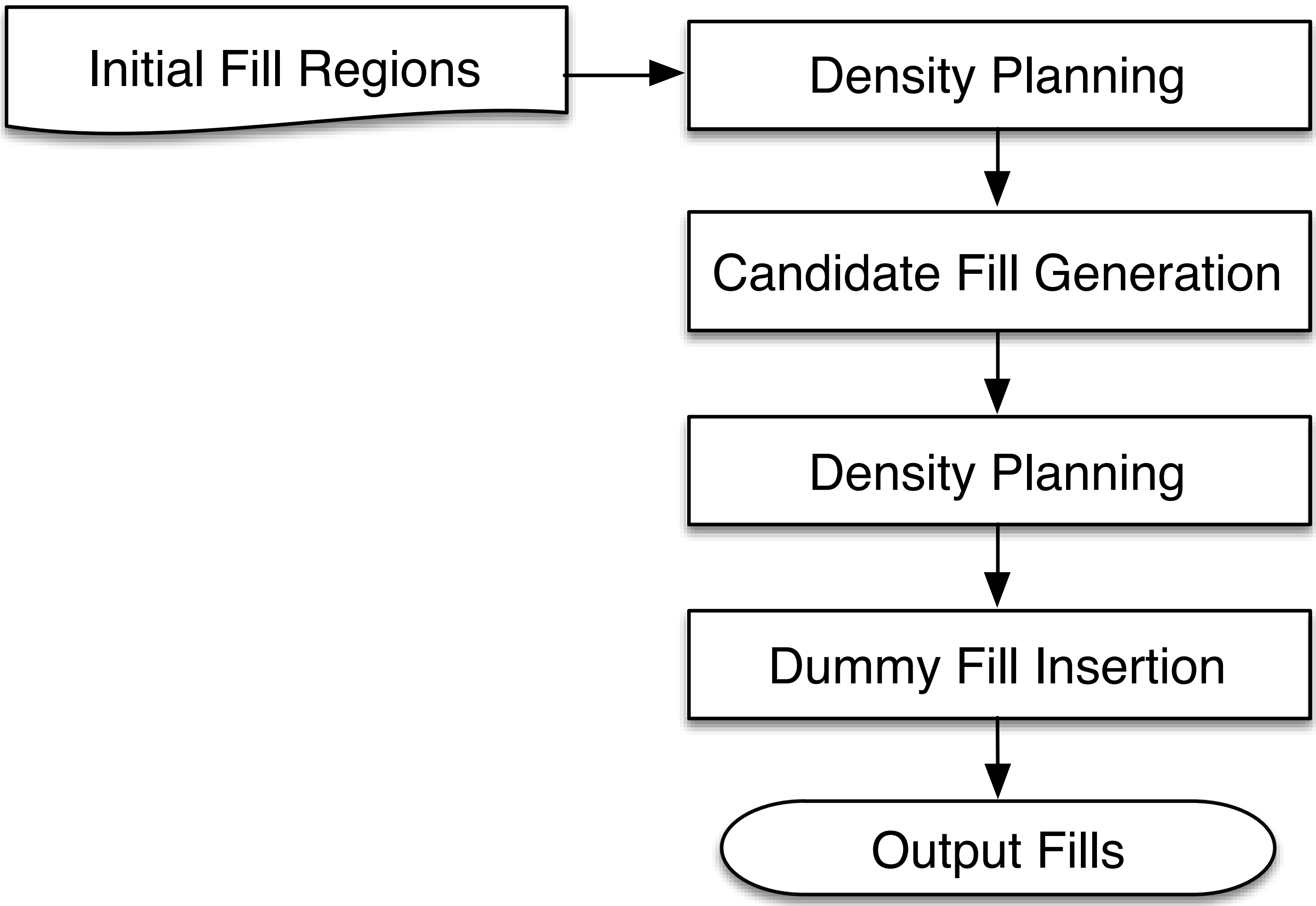


$$c'_i = \begin{cases} c_i & i = 1, 2, ..., N \\ -\sum_{i=1}^{N} c_i & i = 0 \end{cases}$$

$$b'_{ij} = \begin{cases} b_{ij} & (i,j) \in E \\ l_i & i = 1, 2, ..., N, j = 0 \\ -u_i & i = 0, j = 1, 2, ..., N \end{cases}$$

- Convert bound constraints to differential constraints
- Dual to min-cost flow

Overall Flow

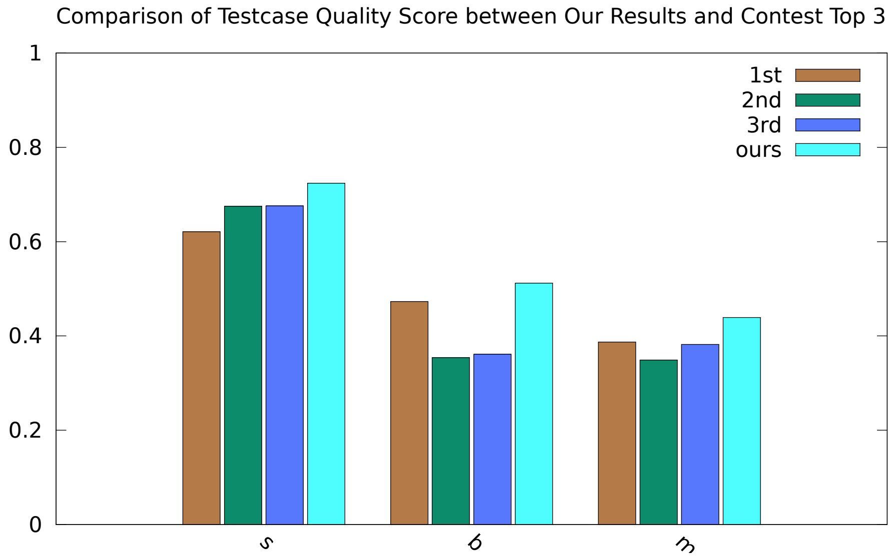


Experimental Environment

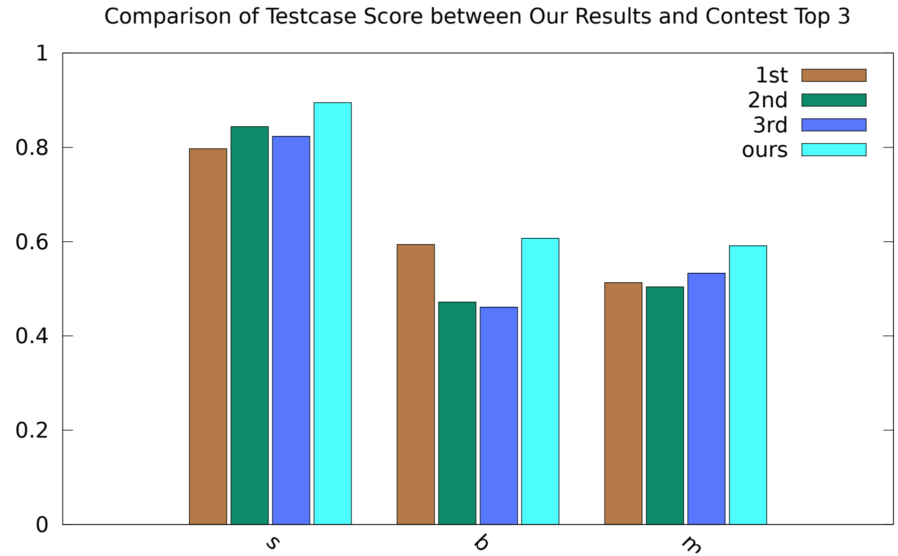
- Implemented in C++
- 8-Core 3.4GHz Linux server
- 32GB RAM
- ICCAD 2014 contest benchmarks

Experimental Results

- Compared with contest winners
  - Quality Scores (13% better than the 1st place winner)
  - Overall Scores (10% better than the 1st place winner)



Quality Sores



Overall Scores

Experimental Results

- Detailed results

Table 3: Experimental Results on ICCAD 2014 Benchmark

Design	Team	Overlay*	Variation*	Line*	Outlier*	Size*	Run-time*	Memory*	Testcase Quality	Testcase Score
s	1st	0.743	0.636	0.733	1.000	0.976	0.877	0.885	0.621	0.797
	2nd	0.743	0.909	0.967	0.975	0.103	0.846	0.831	0.675	0.844
	3rd	0.613	0.985	0.990	1.000	0.158	0.842	0.429	0.676	0.823
	ours	0.723	0.948	0.979	0.994	0.887	0.872	0.818	**0.724**	**0.895**
b	1st	0.748	0.368	0.364	0.871	0.924	0.515	0.891	0.473	0.594
	2nd	0.841	0.381	0.534	0.000	0.053	0.513	0.828	0.354	0.472
	3rd	0.576	0.485	0.601	0.000	0.568	0.554	0.339	0.361	0.461
	ours	0.685	0.499	0.470	0.953	0.765	0.351	0.852	**0.512**	**0.607**
m	1st	0.598	0.462	0.486	0.204	0.941	0.556	0.845	0.387	0.513
	2nd	0.668	0.460	0.618	0.000	0.000	0.780	0.761	0.349	0.504
	3rd	0.510	0.509	0.689	0.000	0.807	0.748	0.772	0.382	0.533
	ours	0.493	0.643	0.766	0.088	0.905	0.750	0.786	**0.439**	**0.591**

Conclusion

- Methodology for fill optimization with holistic and multiple objectives
- Validated on industry benchmarks
  - ICCAD 2014 contest benchmark
- Future work
  - Lithography related impacts

Thank you!