New and Exact Filling Algorithms for Layout Density Control

A. B. Kahng, G. Robins, A. Singh, A. Zelikovsky

(UCLA and UVa)

http://vlsicad.cs.ucla.edu

Supported by Cadence Design Systems, Inc. and the Packard Foundation

Outline

- CMP
- Filling problem
- Fixed-dissection regime
- Optimal LP formulation
- Synthesis of filling patterns
- Computational experience
- Summary and research directions

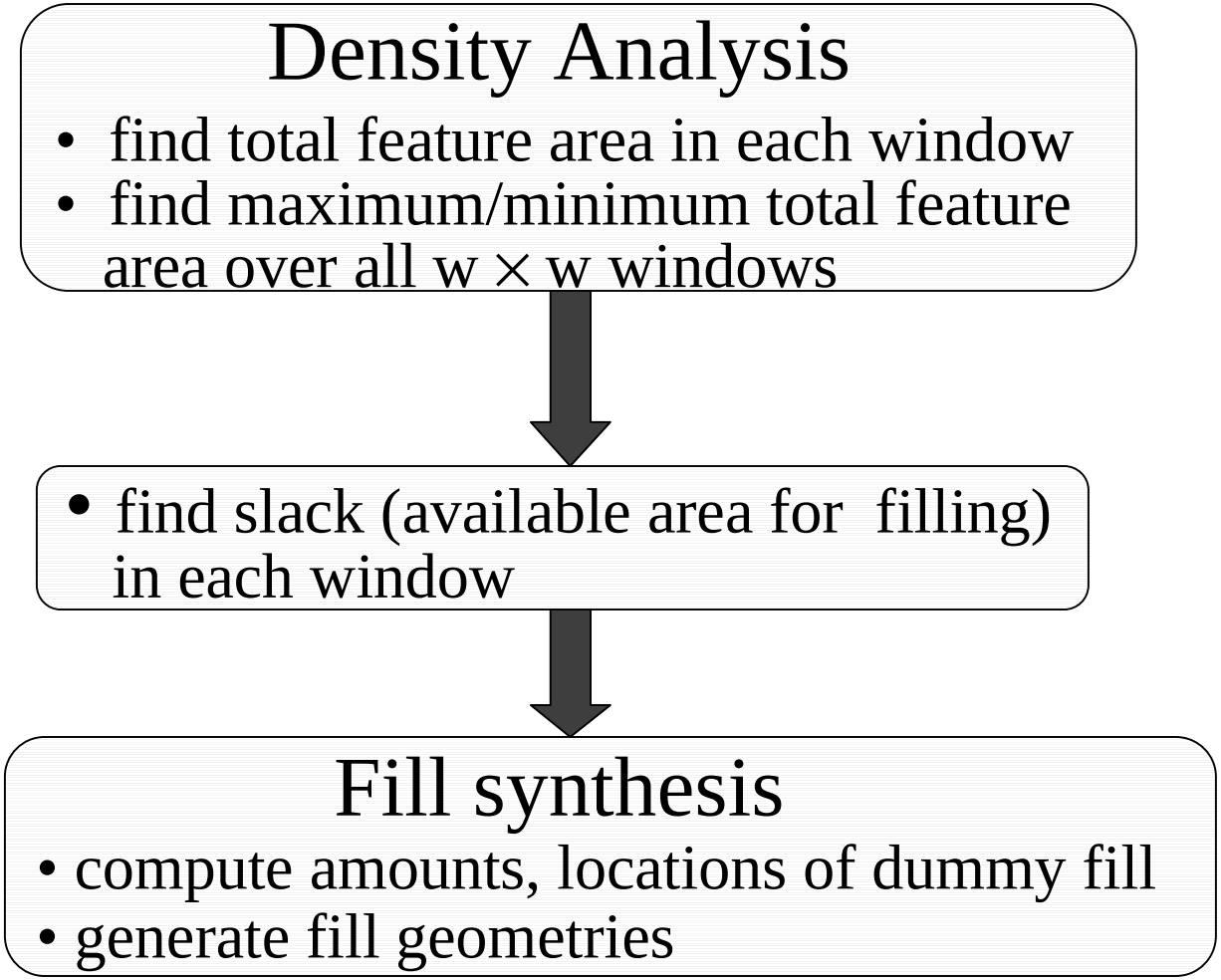
Chemical-Mechanical Polishing (CMP)

- Rotating pad polishes each layer on wafers to achieve planarized surfaces
- Non-uniform feature density affects CMP:
  - overpolishing of empty areas (***dishing***)
  - underpolishing of dense areas
- Density variation must be tightly controlled

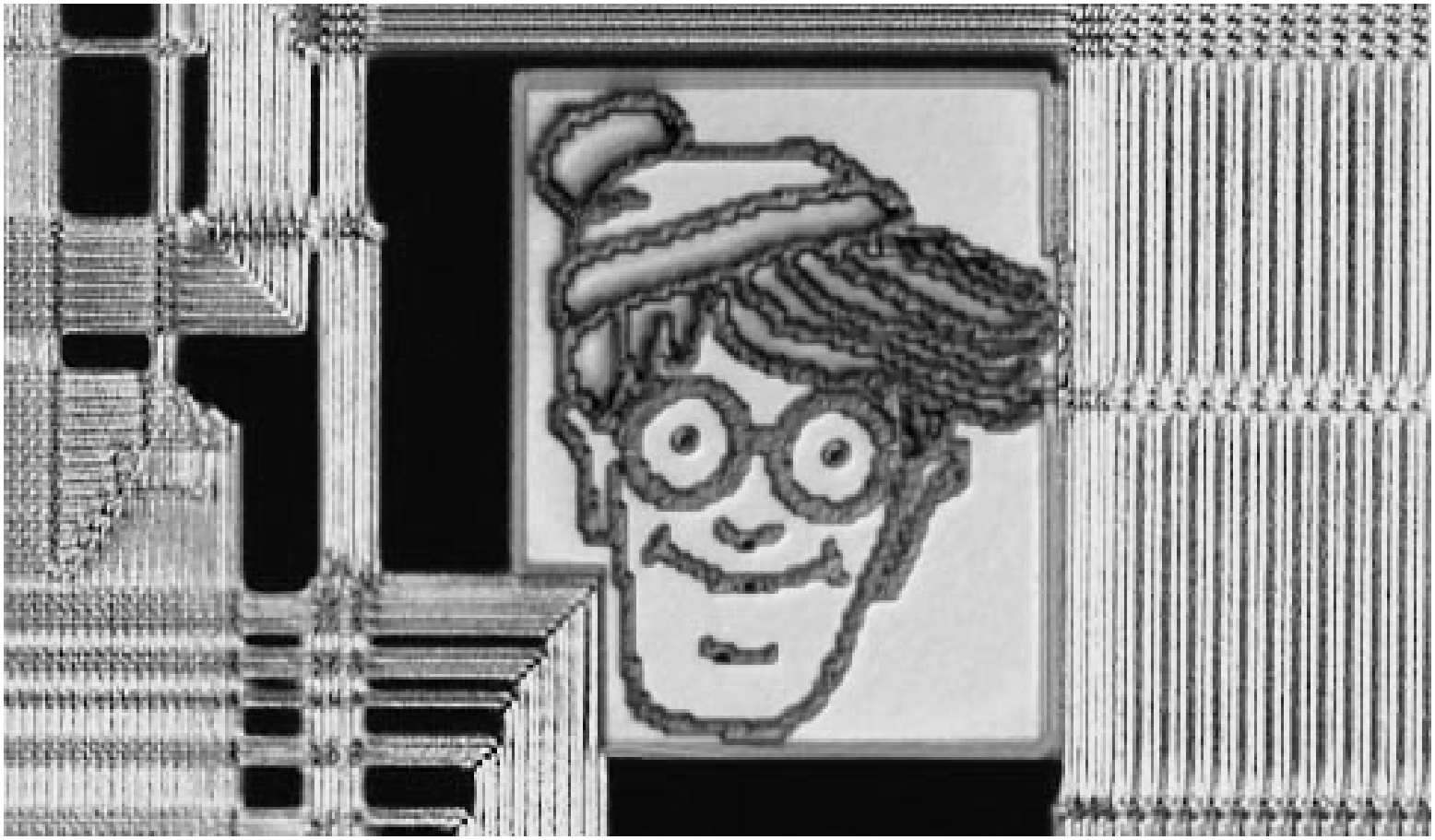
Density Rules

- Modern foundry rules specify layout density bounds to minimize impact of CMP on yield
- Density rules control local feature density for $w \times w$ windows
  - e.g., for metal layer every 2000um $\times$ 2000um window must be between 35% and 70% filled
- Filling = insertion of "dummy" features to improve layout density
  - typically via layout post-processing in PV / TCAD tools
  - affects vital design characteristics (e.g., RC extraction)
  - accurate knowledge of filling is required during physical design and verification

Layout Density Control Flow



Waldo-Fill



Limitations of Current Techniques

- Current techniques for density control have three key weaknesses:

  (1) only the average ***overall*** feature density is constrained, while local variation in feature density is ignored

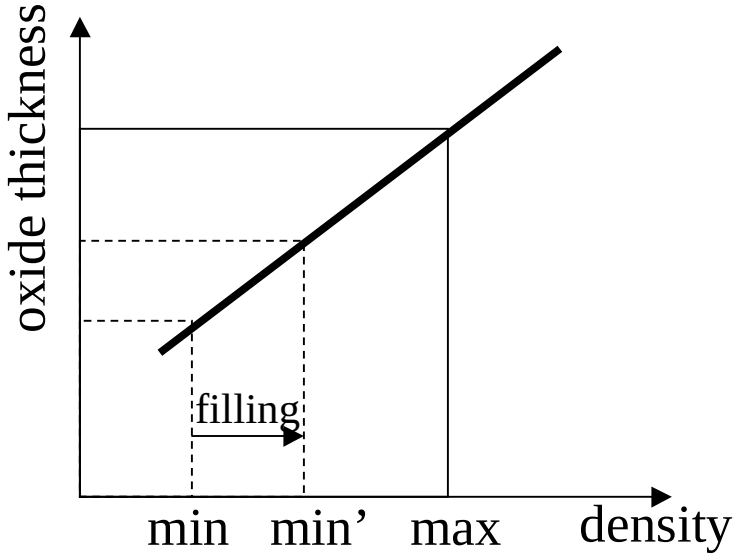
  (2) density analysis does not find ***true*** extremal window densities - instead, it finds extremal window densities only over fixed set of window positions

  (3) fill insertion into layout does not minimize the maximum variation in window density

- This work solves problems (1) and (3)

Min-Variation Objective

- Relationship between oxide thickness and local feature density



- Minimizing variation in window density over layout preferable to satisfying lower and upper density bounds

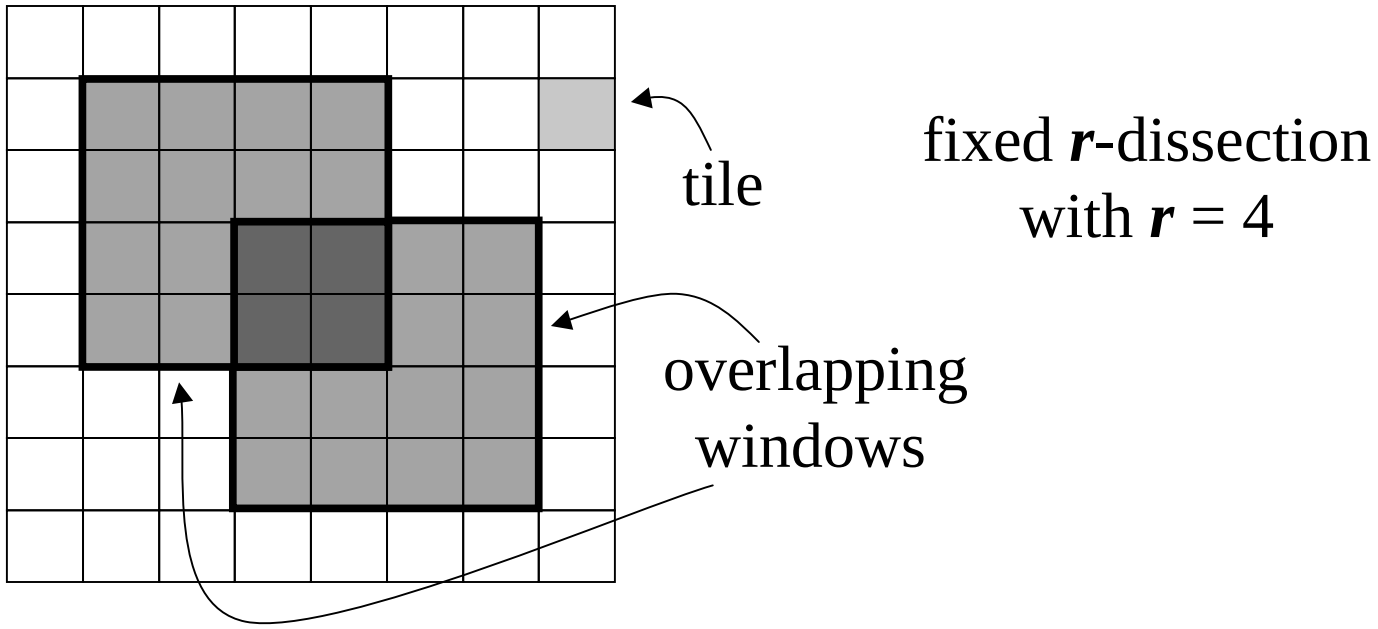
Filling Problem

- **Given** design rule-correct layout

  of k disjoint rectilinear features in $n \times n$ region
- **Find** design rule-correct **filled** layout
  - no fill geometry is added within distance B of any layout feature
  - no fill is added into any window that has density $\geq \boldsymbol{U}$
  - minimum window density in the filled layout is maximized (or has density $\geq$ lower bound $\boldsymbol{L}$)

Fixed-Dissection Regime

- Feature density is imposed only for ***fixed*** set of $w \times w$ windows
- Layout partitioned by r^2 fixed dissections
- Each $w \times w$ window is partitioned in r^2 ***tiles***



Fixed-Dissection Regime

- Let all $\boldsymbol{w} \times \boldsymbol{w}$ windows of fixed $\boldsymbol{r}$-dissection have density $\leq \boldsymbol{U}$
- An arbitrary $\boldsymbol{w} \times \boldsymbol{w}$ window has

  $$\text{density} \leq \min\{1, U + 1/r - 1/(4r^2)\}$$

- This bound is tight

Filling Problem in Fixed-Dissection Regime

- **Given**
  - fixed ***r***-dissection of layout
  - total feature area ***area***[T] in each tile T
  - ***slack***[T] = area available for filling in T
  - maximum window density ***U***
- **Find** total fill area p[T] to add in each T
  - any $\boldsymbol{w} \times \boldsymbol{w}$ window W has density $\leq \boldsymbol{U}$
  - $\min_W \sum_{T \in W}$ (area[T] + p[T]) is maximized

Optimal Solution Via Linear Programming

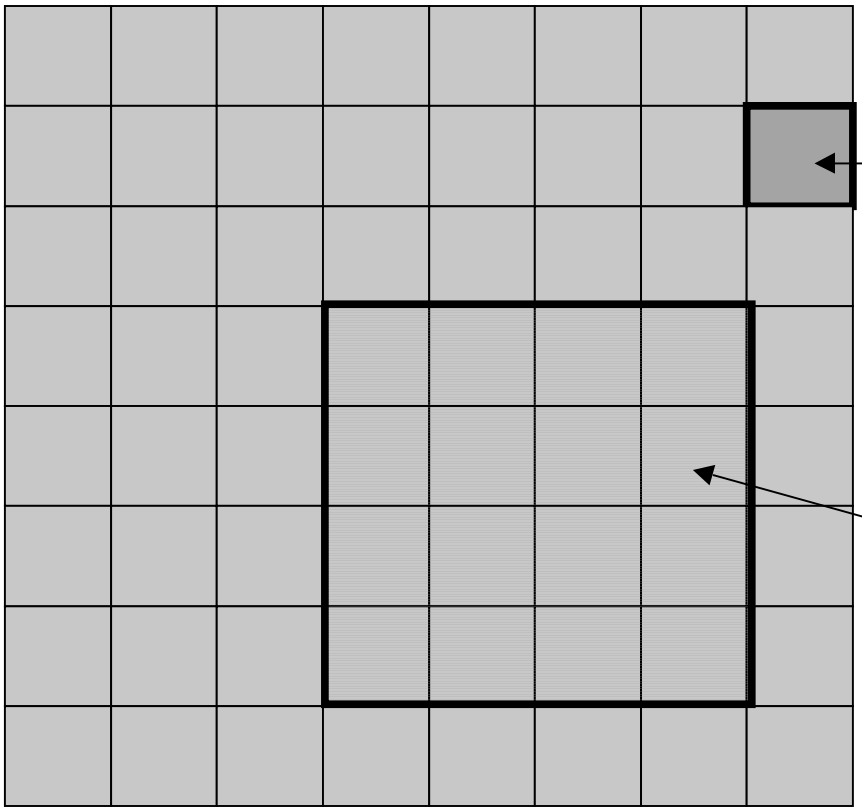
- Maximize M (lower bound on window density)
- subject to:
  - For any tile T: $0 \leq p[T] \leq \textbf{\textit{pattern}} \times slack[T]$
  - For any window W:

$$\sum_{T \in W} p[T] \leq U \times w^2$$

$$M \leq \sum_{T \in W} (p[T] + area[T])$$

(***pattern*** = max achievable pattern area density)

Optimal Solution via Linear Programming

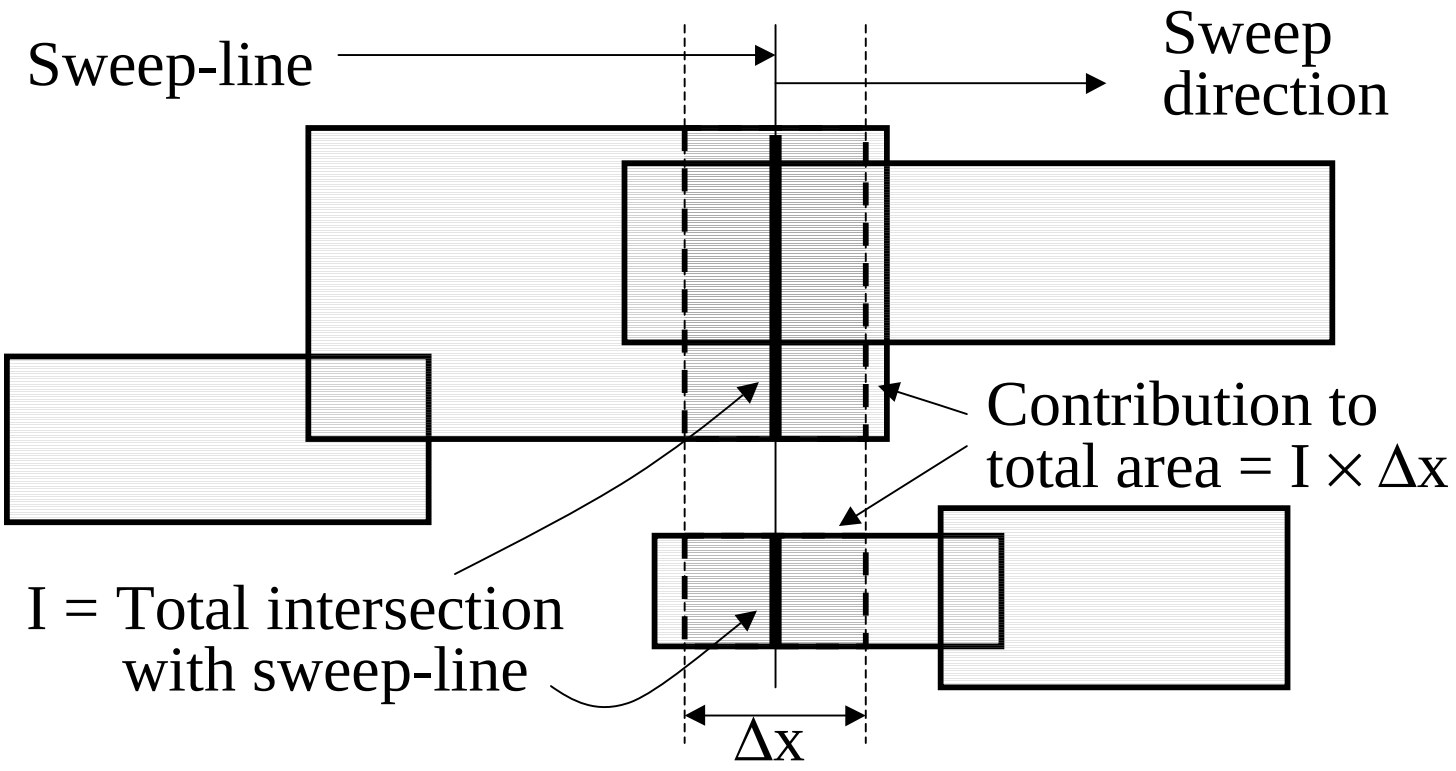


one variable and two constraints per tile

two constraints per window

Slack Computation

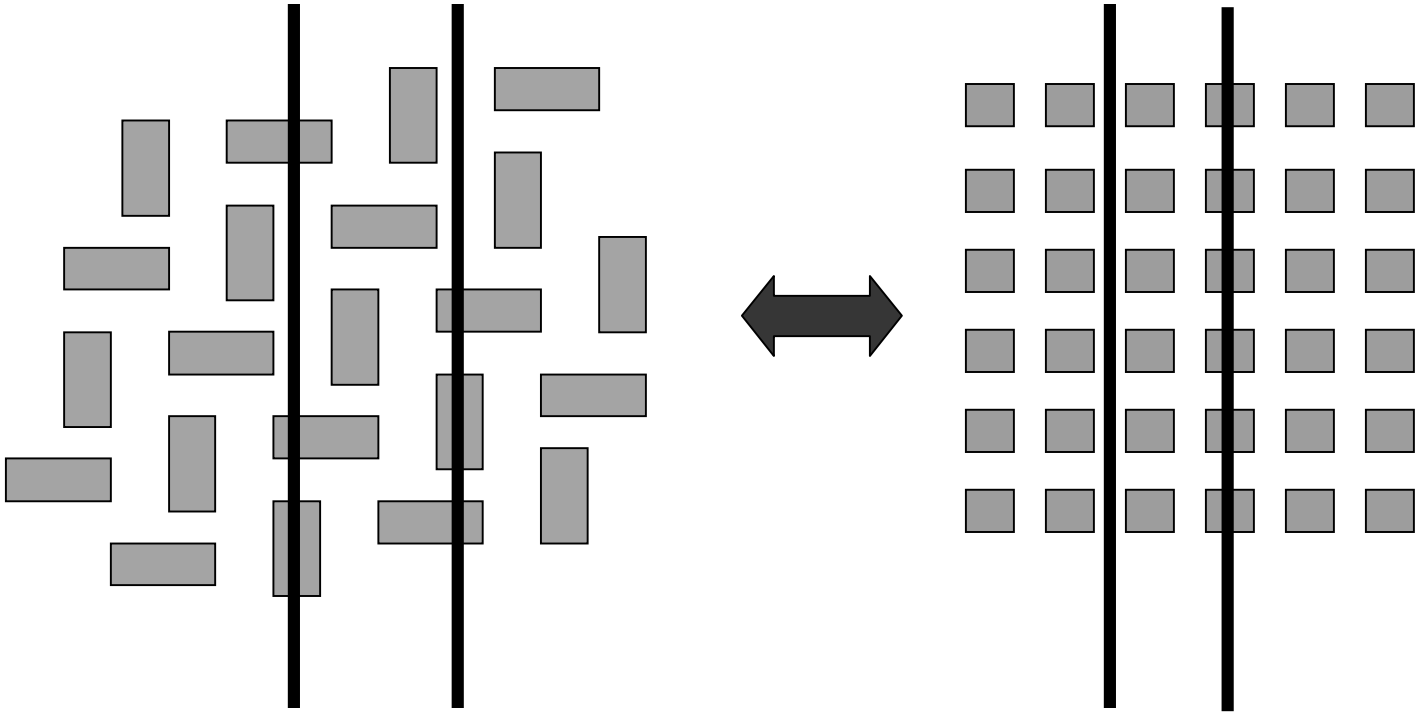
- Sweep-line technique
- Measure of union of rectangles



Synthesis of Filling Patterns

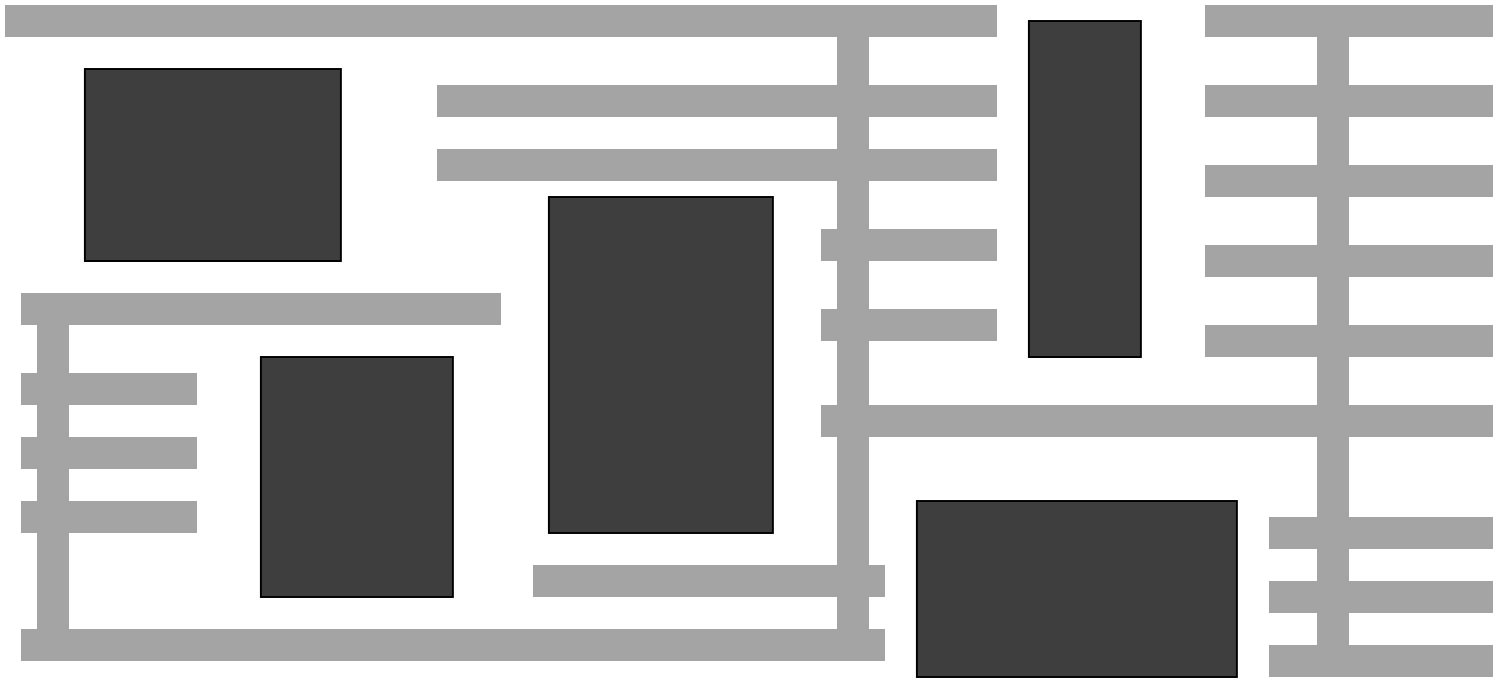
- Given area of filling pattern p[i,j], insert filling pattern into tile T[i,j] ***uniformly*** over available area
- Desirable properties of filling pattern
  - uniform coupling to long conductors
  - either grounded or floating

Basket-Weave Pattern



Each vertical/horizontal crossover line has same overlap capacitance to fill

Grounded Pattern



Fill with horizontal stripes,

then span with vertical lines

Experimental Testbed

- GDSII $\rightarrow$ CIF
- Internally developed geometric processing engines
- All runtimes are CPU seconds, Sun Ultra-10
- Industry test cases

benchmark	layout size	# rectangles	window size
L1	125,000	49,506	31,250
L2	112,000	76,423	28,000
L3	112,000	133,201	28,000

Computational Experience

Density analysis, LP generation and solution, filling generation for fixed-***r*** dissections

benchmark	r	Max Density	CPU time	M	generation	solution	filling
L1	2	0.202	1.3	0.219	4.3	0.1	3.3
L1	4	0.212	2.9	0.219	4.1	0.4	3.2
L1	8	0.217	9.2	0.219	10.3	18.3	3.3
L2	2	0.161	2.1	0.182	2.8	0.1	5.2
L2	4	0.179	4.5	0.171	5.2	1.7	5.1
L2	8	0.179	14.5	0.163	15.8	41.5	5.2
L3	2	0.288	3.6	0.264	5.2	0.1	8.3
L3	4	0.289	8.1	0.261	9.4	0.8	8.1
L3	8	0.291	25.1	0.255	27.2	24.4	8.1

Summary of Contributions

- New ***min-variation*** objective for synthesis of fill geometries
- Within fixed-dissection regime, optimal solution via linear programming for min-variation objective
- Criteria for fill pattern synthesis (both floating and grounded)
- Experimental results showing very fast runtimes

Ongoing Research Directions

- Improved heuristics or exact algorithms for the min-variation formulation
- Solutions to the Filling Problem in the continuous (as opposed to fixed-dissection) regime
- Robust implementations of our linear-programming based fill synthesis methods
- Calibration of our methods with data and density control requirements from industry partners