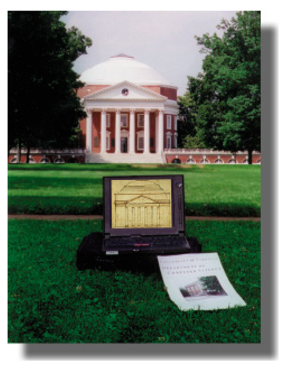
Filling Algorithms and Analyses for Improved VLSI Manufacturability



Department of
Computer Science
School of Engineering
University of Virginia
Charlottesville, Virginia 22904 (434) 982-2207

Andrew Kahng
Gabriel Robins
Anish Singh
Alex Zelikovsky

This research has appeared in:

Kahng, A. B., Robins, G., Singh, A., and Zelikovsky, A., Filling Algorithms and Analyses for Layout Density Control, to appear in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1999.

Kahng, A. B., Robins, G., Singh, A., Wang, H., and Zelikovsky, A., Filling and Slotting: Analysis and Algorithms, Proc. International Symposium on Physical Design, Monterey, California, April, 1998, pp. 95-102.

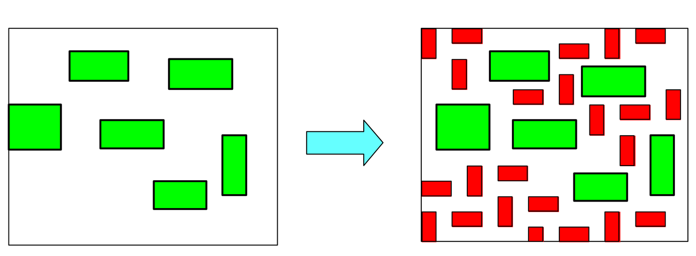
Kahng, A. B., Robins, G., Singh, A., and Zelikovsky, A., New and Exact Filling Algorithms for Layout Density Control, Proc. VLSI Design Conference, Goa, India, January 1999, pp. 106-110.

Kahng, A. B., Robins, G., Singh, A., and Zelikovsky, A., New Multi-Level and Hierarchical Algorithms for Layout Density Control, Proc. Asia and South Pacific Design Automation Conference, Hong Kong, January 1999, pp. 221-224; nominated for **Best Paper Award**.

www.cs.virginia.edu/robins

Metal Filling

- VLSI manufacturing steps (e.g., CMP) have varying affect on device and interconnect features
- *Density rules*: minimize impact of variation on yield and improve predictability
- Uniformity achieved by post-processing via *filling* i.e. insertion of features:



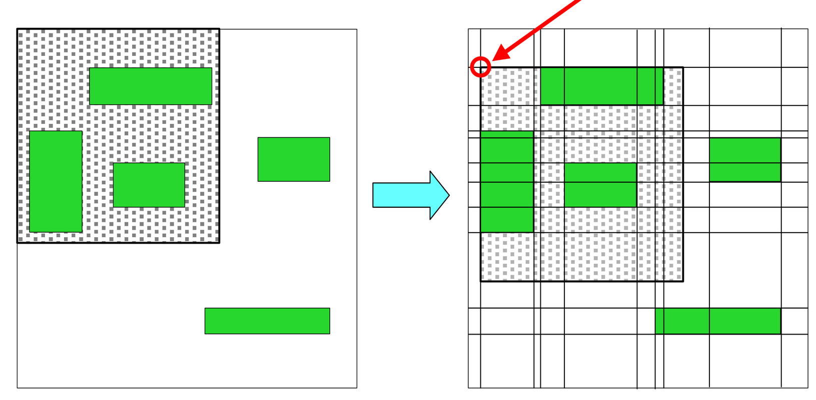
Problem Formulation

Given: a design rule-correct geometry with:
- k disjoint rectangles in $n \times n$ layout region
- a minimum feature size c
- area density lower and upper bounds
- window size $w < n$

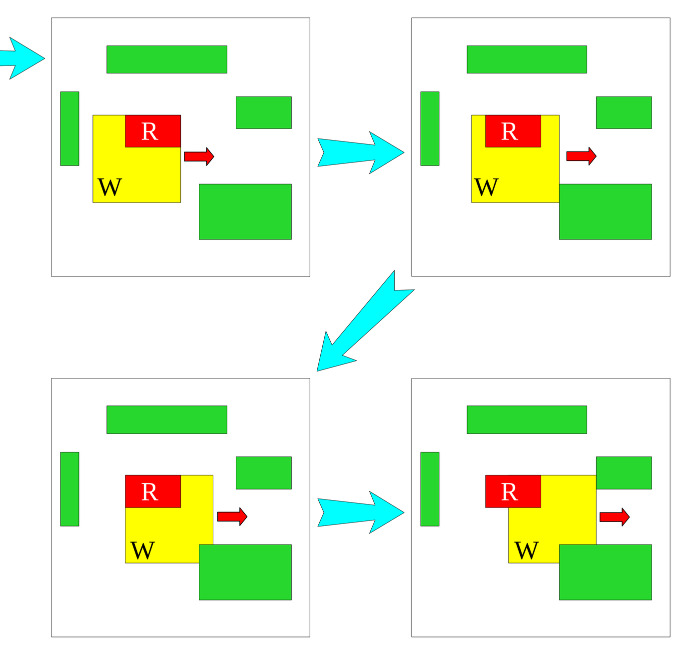
Goal: add fill geometries into layout while:
- preserving circuit function & design rule-correctness
- all $w \times w$ windows satisfy lower/upper density bounds

Extremal-Density Window Analyses

- Theorem (Hanan analogue): there is extremal $w \times w$ window with a corner at Hanan gridpoint

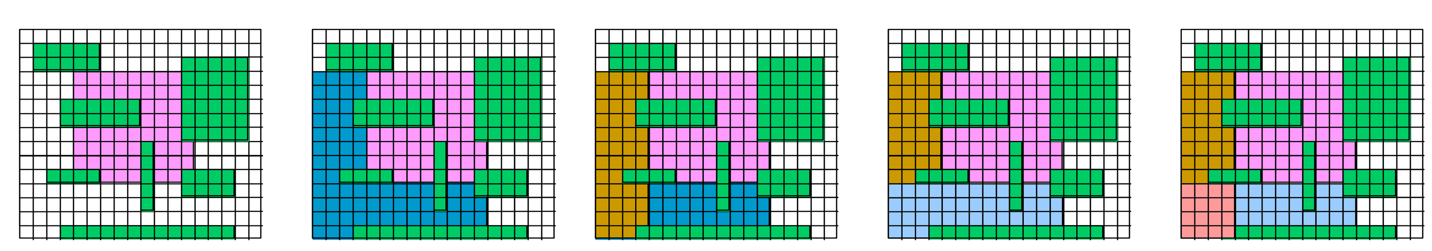


$O(k^2)$ -time algorithm for k rectangles:



- For each pivot rectangle R do:
  - Find the density of $w \times w$ window W that abuts R on top and right
  - While W intersects R do:
    - slide W right untill intersection with next rectangle edge
    - record changes in density as you go
  - Repeat for R, this time sliding up
  - Repeat for all other orientations of W
- Speedup via ***fixed preprocessing*:**
  - Partition layout into $\frac{n}{w} \times \frac{n}{w}$ squares of size $w \times w$ each
  - Record all rectangles intersecting each $w \times w$ square
  - Proceed as above
  - Expected runtime: $O\left(\left(\frac{n}{w}\right)^2 + k\log k + \left(\frac{wk}{n}\right)^2\right)$

Given: k rectangles in $n \times n$ layout
Find: extremal-density $w \times w$ window
- Inclusion-exclusion -based $O(n^2)$ -time algorithm:

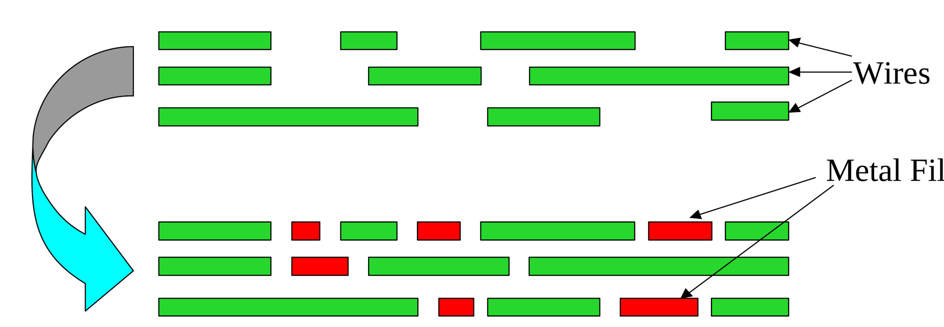


- initialize boolean $n \times n$ array B with 0's and 1's
- create integer $n \times n$ array $S[i, j]$ = # 1's in $B[1..i, 1..j]$
- density $w \times w$ window at (i, j) = $S[i+w, j+w]$ - $S[i, j+w]$ - $S[i+w, j]$ + $S[i, j]$

Metal Fill Types

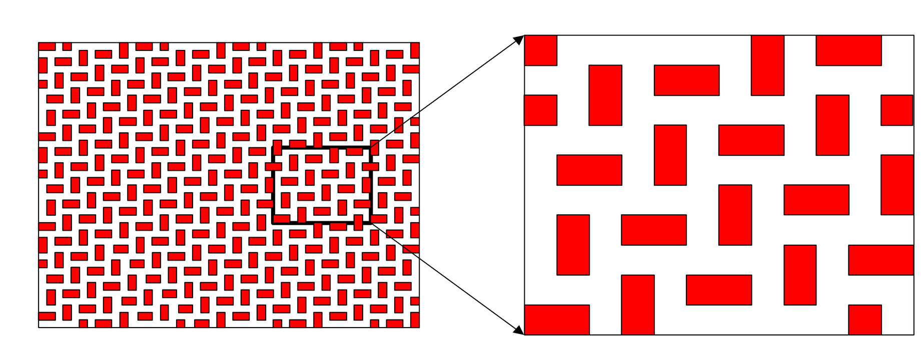
Wiring Layers

- Density $\le$ 50%-60% if separation $\approx$ row width
- $O(k \log k)$ -algorithm for filling

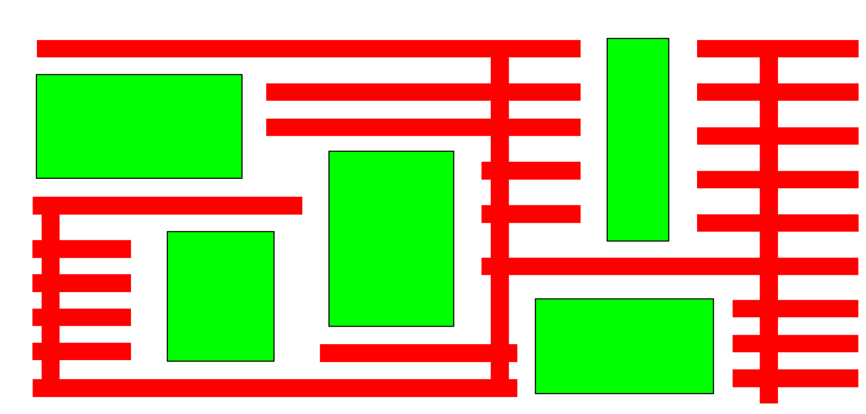


Basket Weaving

- Fill pattern should not consist of regular grid geometries .
- Idea: basket-weave pattern using modulo arithmetic:

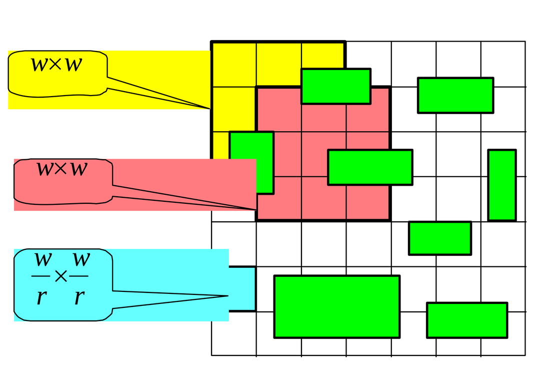


Grounded Fill



Fixed-Dissection Density Analysis

Practical method: check / enforce density constraints only in $w \times w$ windows of fixed dissection with $\frac{w}{r} \times \frac{w}{r}$ *tiles*



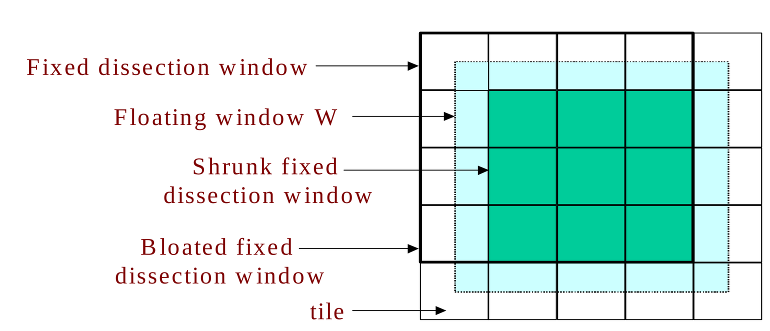
Theorem: If all fixed dissection $w \times w$ windows has density between U and L, then any $w \times w$ window has density in the range between $L - \left(\frac{1}{r} - \frac{1}{4r^2}\right)$ and $U + \left(\frac{1}{r} - \frac{1}{4r^2}\right)$

Theorem: If all fixed dissection $\frac{w}{r} \times \frac{w}{r}$ tiles have density between L and U then any $w \times w$ window has density in the range between $\frac{(r-1)^2}{r^2} \cdot L + \frac{4(r-1)}{r^2} \cdot \max\{L - \frac{1}{2}, 0\} + \frac{4}{r^2} \cdot \max\{L - \frac{3}{4}, 0\}$

and $\frac{(r-1)^2}{r^2} \cdot U - \frac{4(r-1)}{r^2} \cdot \max\{U - \frac{1}{2}, 0\} - \frac{4}{r^2} \cdot \max\{U - \frac{1}{4}, 0\}$

Multi-Level Density Analysis

- Any floating window within a fixed dissection is contained within a
  - bloated $w(1+\frac{1}{r})$ x $w(1+\frac{1}{r})$ window, and also contains a
  - window of size $w(1-\frac{1}{r})$ by $w(1-\frac{1}{r})$
- Maximum area of floating windows is bounded by that of a bloated window.

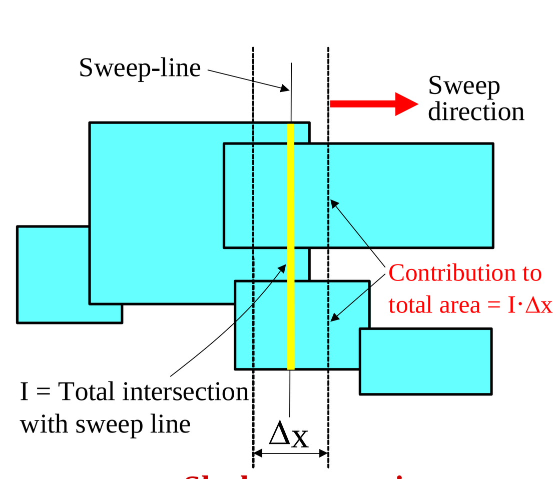


Input: n x n layout, window size and accuracy
Output: maximum area density of w x w window

```
Accuracy = ∞;  r=1
SurvivingTiles = all w/r × w/r tiles
while ( Accuracy > desiredAccuracy)
{
    Find maximum standard window area using surviving tiles (Max)
    Find maximum bloated window area using surviving tiles (BloatMax)
    Delete tiles that do not belong to any bloated window with area > Max
    Accuracy = BloatMax / Max
    r = 2r  // replace surviving tiles by 4 sub-tiles
}
Output maximum window area = (BloatMax + Max) / 2
```

Linear Programming Approach to Computing Optimal Fill Amount

- Filling problem for a fixed r-dissection:
  - Given r-dissection with tiles of size $\frac{w}{r} \times \frac{w}{r}$
  - Given Area and Slack of each tile (T_{ij})
  - The fill amount (P_{ij}) in each tile should satisfy:
    - $0 \le P_{ij} \le Slack(T_{ij})$
    - $\sum P_{ij} \le \max\{UW^2\text{-area}(W), 0\}$ for all wxw windows
- Min variation formulation seeks to maximize the minimum window density: Maximize (min (Area(T_{ij}) + P_{ij}))



Slack computation

Linear Program

Maximize M subject to:

$$P_{ij} \ge 0 \quad i,j = 1,\ldots,\frac{nr}{w} - 1$$

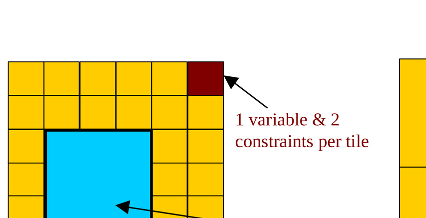
$$P_{ij} \le Slack(T_{ij}) \quad i,j = 1,\ldots,\frac{nr}{w} - 1$$

$$\sum_{s=i}^{i+r-1}\sum_{t=j}^{j+r-1} P_{st} \le U.w^2 - Area(W_{ij}) \quad i,j = 1,\ldots,\frac{nr}{w} - r + 1$$

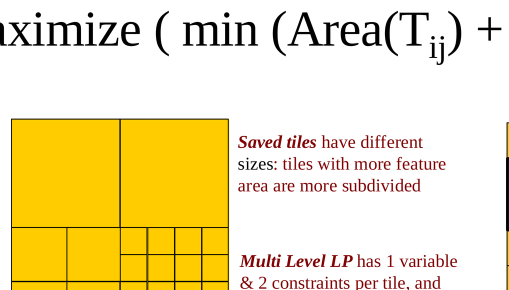
$$M \le Area(W_{ij}) + \sum_{s=i}^{i+r-1}\sum_{t=j}^{j+r-1} P_{st} \quad i,j = 1,\ldots,\frac{nr}{w} - r + 1$$

where:

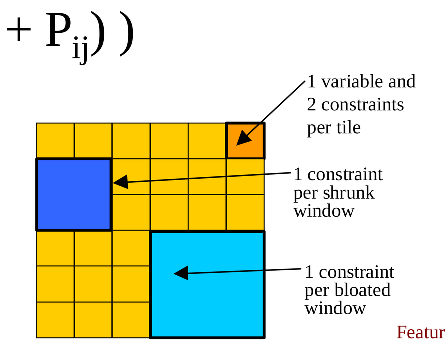
$$Area(W_{ij}) = \sum_{s=i}^{i+r-1}\sum_{t=j}^{j+r-1} Area(T_{st})$$



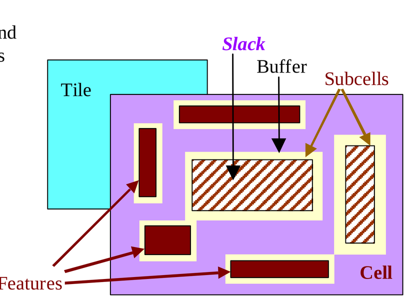
Fixed-dissection LP



Multi-level LP



Floating Deviation LP



Hierarchical LP

Computational Results

VLSI Layer Benchmarks

Benchmark	N=layout size	K= # rectangles	W= window size
L1	12,500	49,506	31,250
L2	11,200	76,423	28,000
L3	11,120	133,201	28,000

Fixed-Dissection Density Analysis

Benchmark	r	Max Density	CPU Time
L1	4	.2125	2.9
L1	8	.2170	9.2
L2	4	.1791	4.5
L2	8	.1791	14.5
L3	4	.2895	8.0
L3	8	.2910	25.1

Multilevel Density Analysis

Benchmark	Accuracy	Max Density	CPU Time
L1	2%	.2184	2.8
L1	3%	.2184	2.8
L2	2%	.1830	6.9
L2	3%	.1829	3.8
L3	2%	.2925	7.1
L3	3%	.2911	6.6

Fixed Dissection LP Fill Generation

Benchmark	r	LP Generation CPU	LP Solve CPU	M	Fill CPU	Total CPU
L1	2	4.1	0.1	.2192	3.3	7.6
L1	4	4.0	0.4	.2192	3.2	7.6
L1	8	10.3	18.3	.2189	3.3	31.9
L2	2	2.8	0.1	.1816	5.2	8.0
L2	4	5.2	1.7	.1704	5.0	11.9
L2	8	15.8	41.5	.1631	5.2	62.5
L3	2	5.2	0.1	.2640	8.3	13.5
L3	4	9.4	0.8	.2606	8.0	18.2
L3	8	27.2	24.4	.2553	8.1	59.7