Filling And Slotting: Analysis and Algorithms

A.B. Kahng (UCLA)
G.Robins (University of Virginia)
A. Singh (University of Virginia)
H. Wang (UCLA)
A. Zelikovsky (UCLA)

Filling and Slotting

- Manufacturing steps affecting interconnect features
  - chemical-mechanical polishing
  - optical exposure
  - resist development and etch
  - chemical vapor deposition

Filling and Slotting

- Manufacturing steps affecting interconnect features
- ***Density rules*** are imposed to minimize impact of manufacturing on device yield
  - for metal layer every10μm×10μm-window contains at least 35μm and at most 70μm metal features

Filling and Slotting

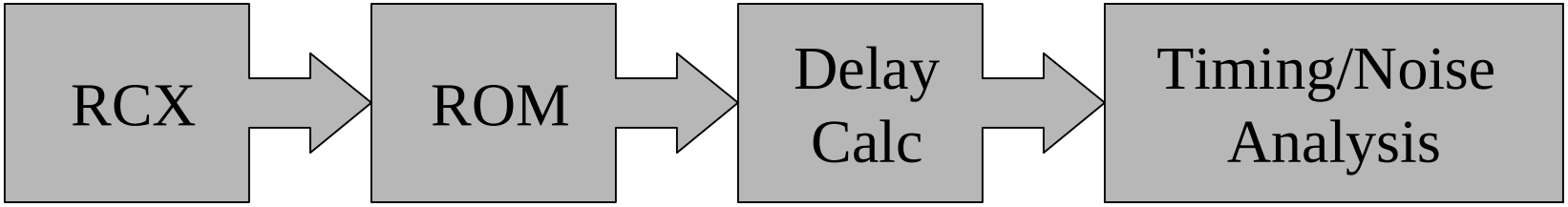
- Manufacturing steps affecting interconnect features
- ***Density rules*** are imposed to minimize impact of manufacturing on device yield
- Uniformity achieved by post-processing via
  - ***filling*** = insertion of features
  - ***slotting*** = partial deletion of features

Filling and Slotting

- Manufacturing steps affecting interconnect features
- ***Density rules*** are imposed to minimize impact of manufacturing on device yield
- Uniformity achieved by post-processing via
  - ***filling*** = insertion of features
  - ***slotting*** = partial deletion of features
- Accurate estimation of filling & slotting is required for physical design and verification tools

Need for Density Awareness in Layout

- Performance verification flow:



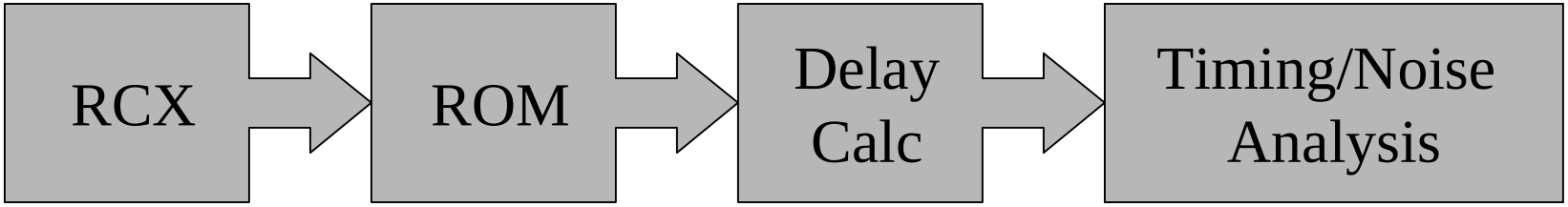
- Filling/slotting geometries affect RC extraction

VICTIM LAYER TOTAL CAPACITANCE (10^{-15} F)			
Same layer-i neighbors?	Fill layers i-1, i+1?	$\varepsilon = 3.9$	$\varepsilon = 2.7$
N	N	2.43 (1.0)	1.68 (1.0)
N	Y	3.73 (1.54)	2.58 (1.54)
Y	N	4.47 (1.84)	3.09 (1.84)
Y	Y	5.29 (2.18)	3.66 (2.18)

- Up to 1% error in extracted capacitance
- Reliability also affected (e.g. slotting of power stripes)

Need for Density Awareness in Layout

- Performance verification flow:



- Can be considered as ``single-layer'' problem

Middle Victim Conductor Total Capacitance (10^{-15} F)			
Fill layer offset	Fill geometry	$\varepsilon = 3.9$	$\varepsilon = 2.7$
N	10×1	3.776 (1.0)	2.614 (1.0)
N	1×1	3.750 (0.99)	2.596 (0.99)
Y	10×1	3.777 (1.00)	2.615 (1.00)
Y	1×1	3.745 (0.99)	2.593 (0.99)

- Caveat: contacting, active layers, other interlayer dependencies

Filling and Slotting Problem

Given a design rule-correct geometry

- of k disjoint rectilinear rectangles in $n \times n$ layout region
- a minimum feature size c
- area and/or perimeter density lower (L_a , L_p) and upper (U_a , U_p) bounds
- window size $w < n$

add fill and slot geometries into layout

- preserving circuit function and design rule-correctness
- such that every $w \times w$ window satisfies lower/upper on area and/or perimeter density

Outline

- Extremal-density window analysis
  - $O(n^2)$ exclusion-inclusion algorithm
  - $O(k^2)$ algorithm for k rectangles
  - rectangle-based analysis in fast expected time
- Fixed dissection window analysis
- Synthesis of filling and slotting patterns
  - fill synthesis for wiring type layouts
  - minimization of slotting
  - simultaneous area and perimeter bounds
- Computational experience
- Conclusions and ongoing research directions

Extremal-Density Window Analysis

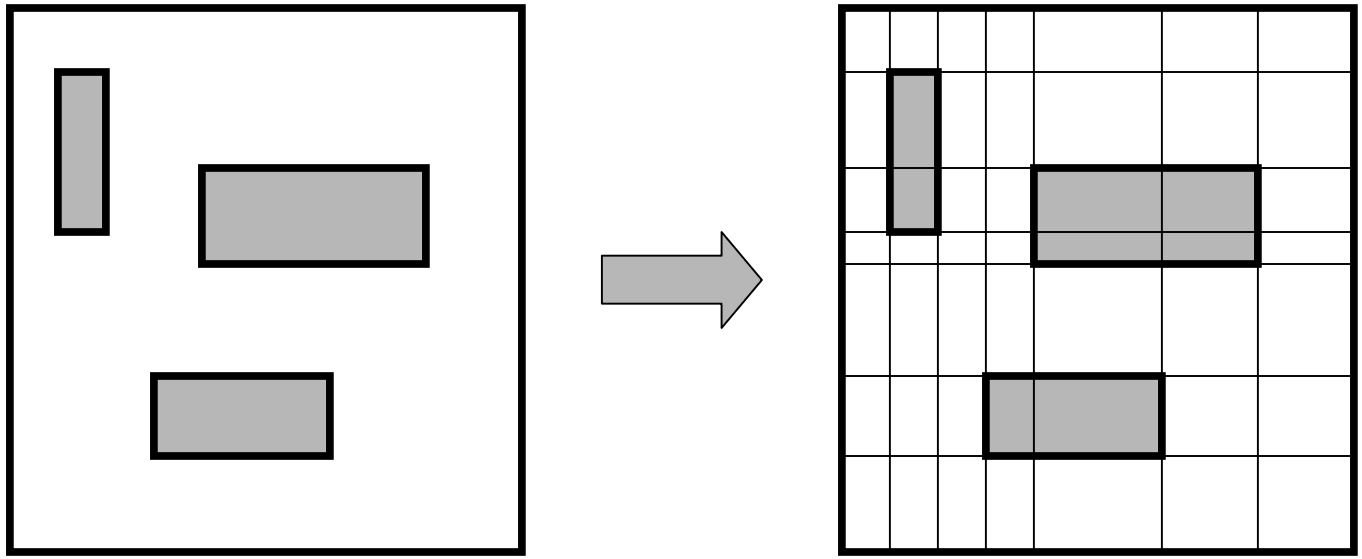
Given *k* rectangles in *n*×*n* layout, find an extremal -density *w*×*w* window in the layout

- Inclusion-exclusion $O(n^2)$ algorithm
  - initialize boolean *n*×*n* array *B* with 0's and put 1's in positions covered by rectangles
  - create integer *n*×*n* array *S*[*i*, *j*] = # 1's in *B*[1..*i*,1..*j*]
  - density *w*×*w* window at (*i*, *j*)

    = *S*[*i*+*w*, *j*+*w*] - *S*[*i*+*w*, *j*+*w*] - *S*[*i*, *j*+*w*] + *S*[*i*, *j*]

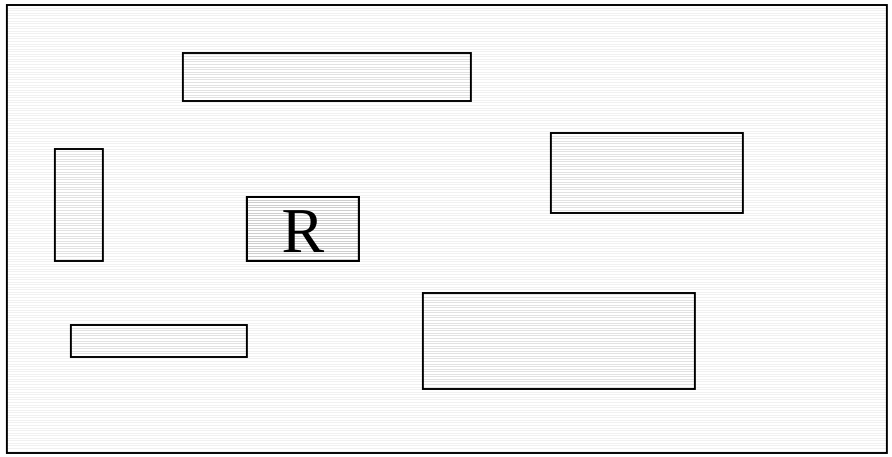
Extremal-Density Window Analysis

- Hanan's Theorem analogue: there is extremal $w \times w$ window with one corner at Hanan grid



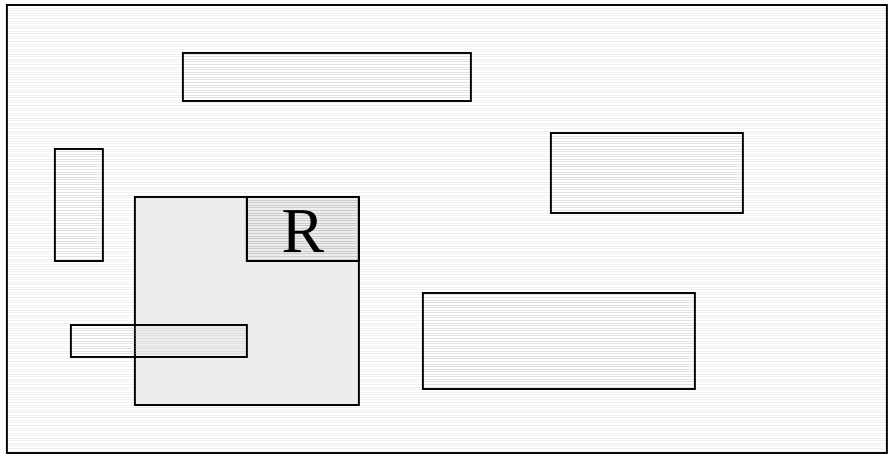
Extremal-Density Window Analysis

- $O(k^2)$ algorithm for k rectangles:
  - for each pivot rectangle R do



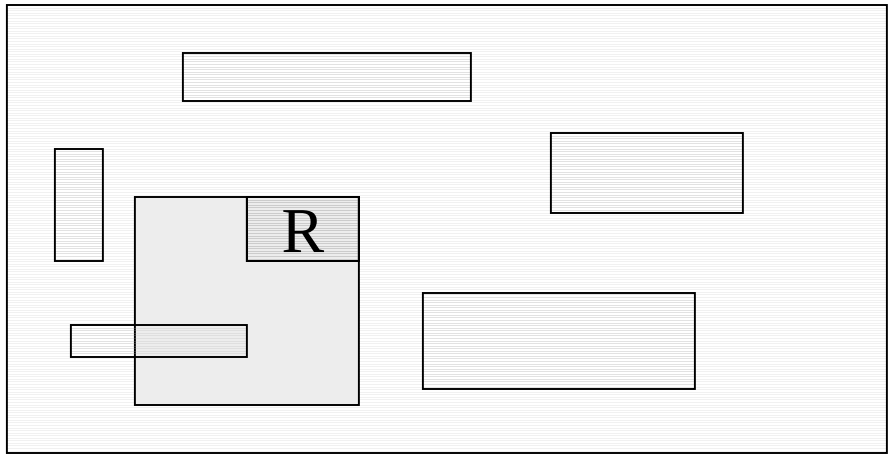
Extremal-Density Window Analysis

- $O(k^2)$ 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



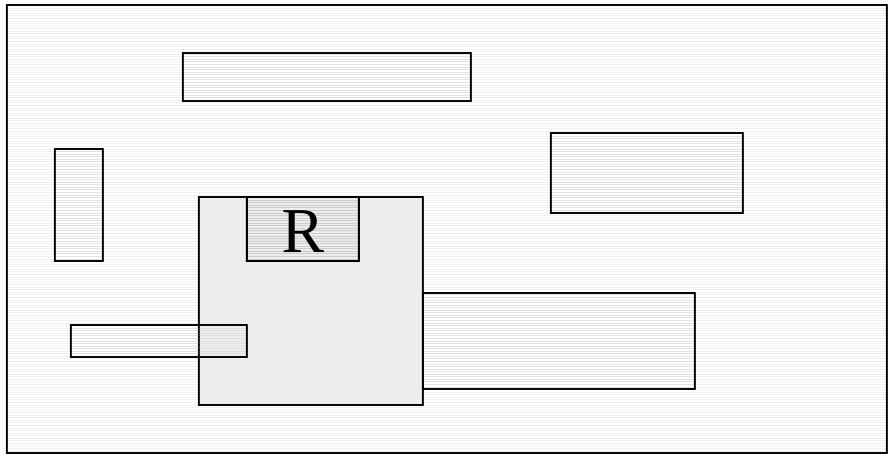
Extremal-Density Window Analysis

- $O(k^2)$ 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 till intersection with other rectangle edge
      - record changes in density



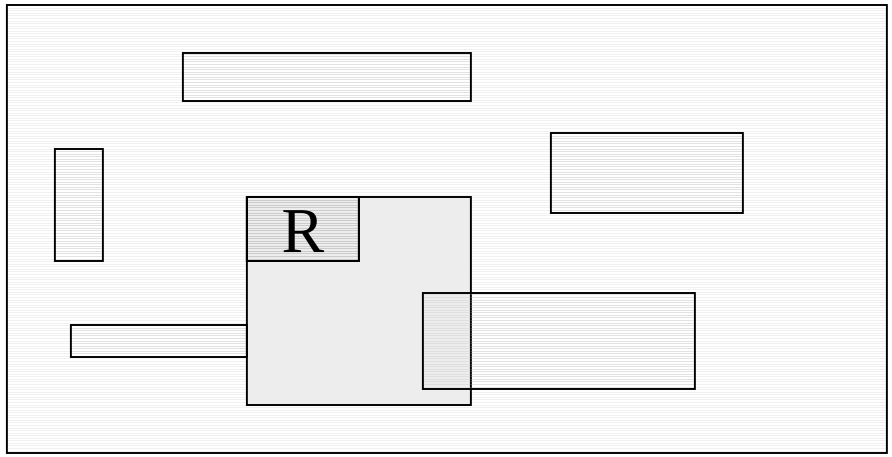
Extremal-Density Window Analysis

- $O(k^2)$ 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 till intersection with other rectangle edge
      - record changes in density



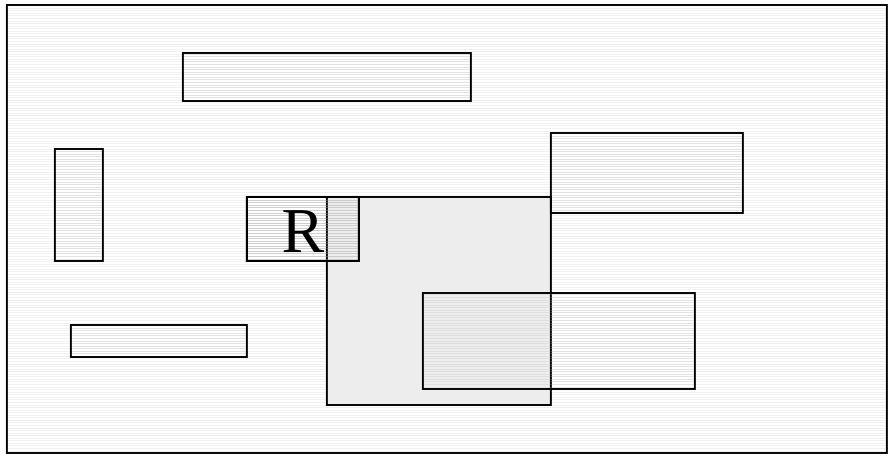
Extremal-Density Window Analysis

- $O(k^2)$ 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 till intersection with other rectangle edge
      - record changes in density



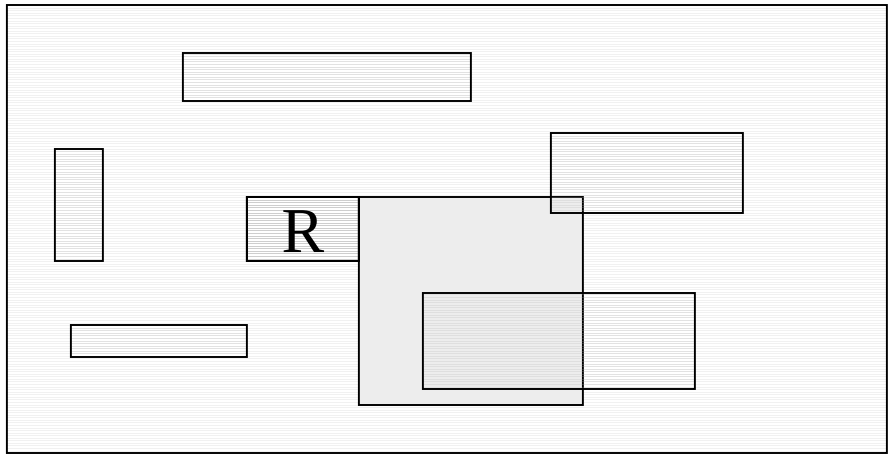
Extremal-Density Window Analysis

- $O(k^2)$ 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 till intersection with other rectangle edge
      - record changes in density



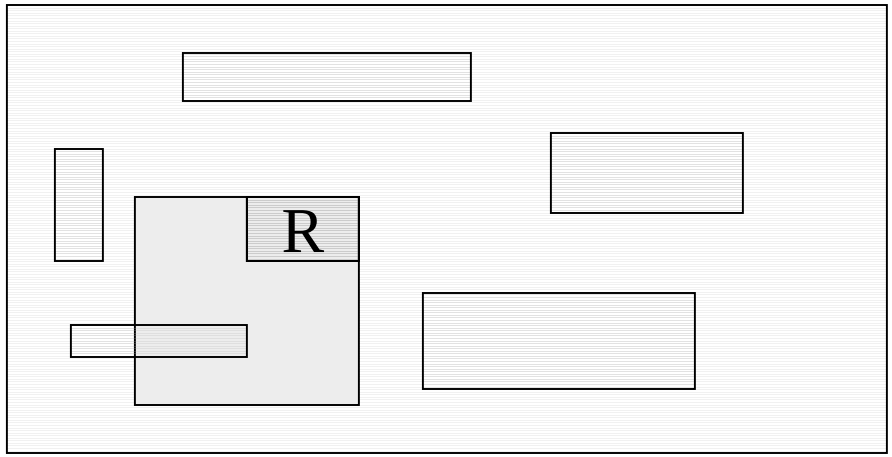
Extremal-Density Window Analysis

- $O(k^2)$ 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 till intersection with other rectangle edge
      - record changes in density



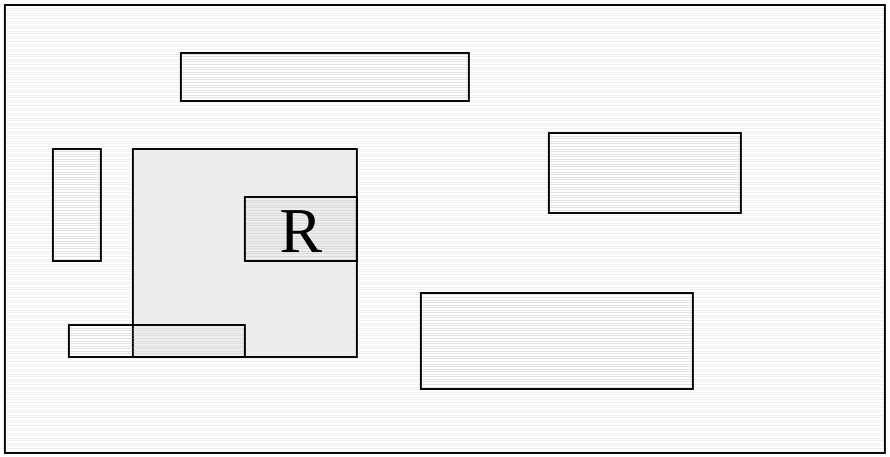
Extremal-Density Window Analysis

- $O(k^2)$ 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 till intersection with other rectangle edge
      - record changes in density
    - repeat sliding R up



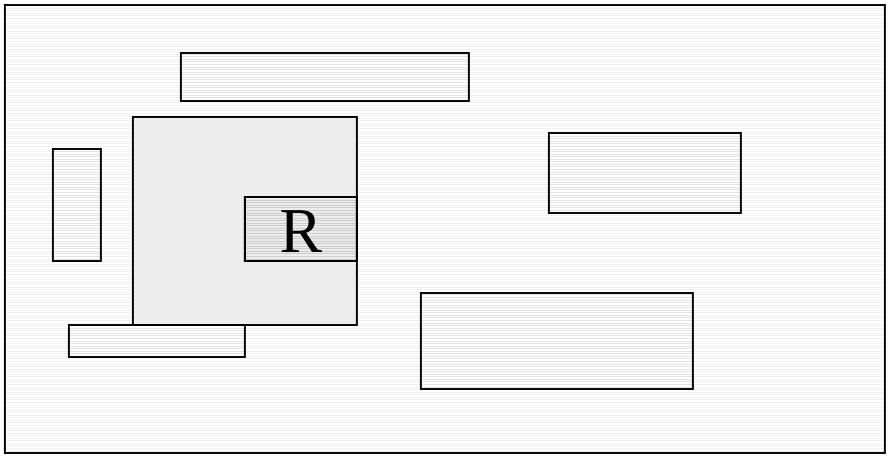
Extremal-Density Window Analysis

- $O(k^2)$ 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 till intersection with other rectangle edge
      - record changes in density
    - repeat sliding R up



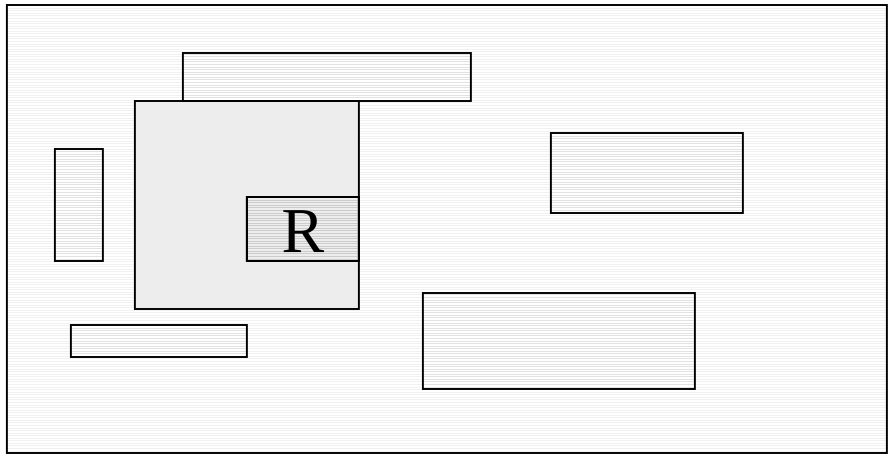
Extremal-Density Window Analysis

- $O(k^2)$ 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 till intersection with other rectangle edge
      - record changes in density
    - repeat sliding R up



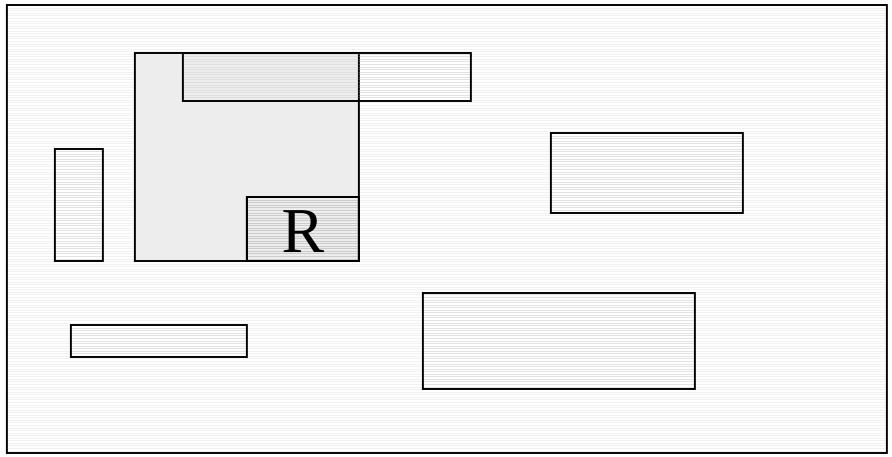
Extremal-Density Window Analysis

- $O(k^2)$ 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 till intersection with other rectangle edge
      - record changes in density
    - repeat sliding R up



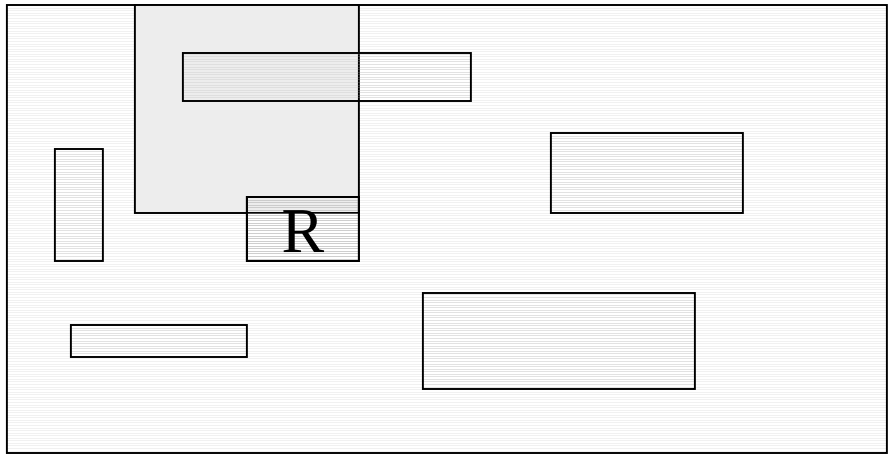
Extremal-Density Window Analysis

- $O(k^2)$ 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 till intersection with other rectangle edge
      - record changes in density
    - repeat sliding R up



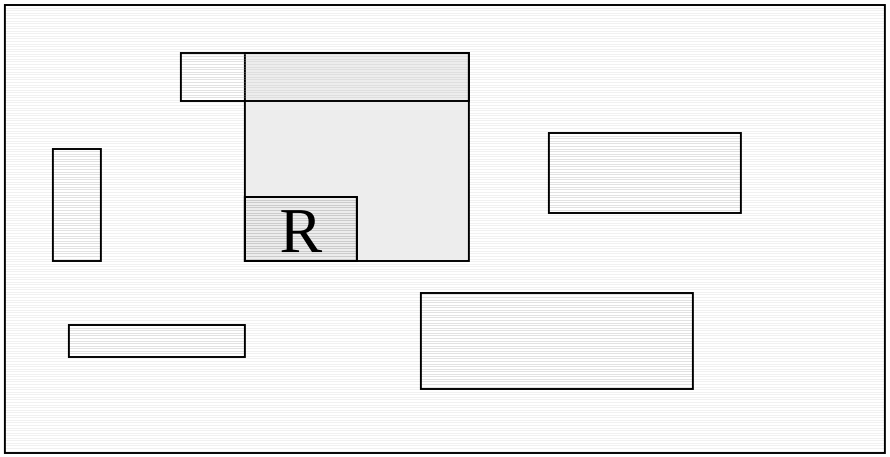
Extremal-Density Window Analysis

- $O(k^2)$ 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 till intersection with other rectangle edge
      - record changes in density
    - repeat sliding R up



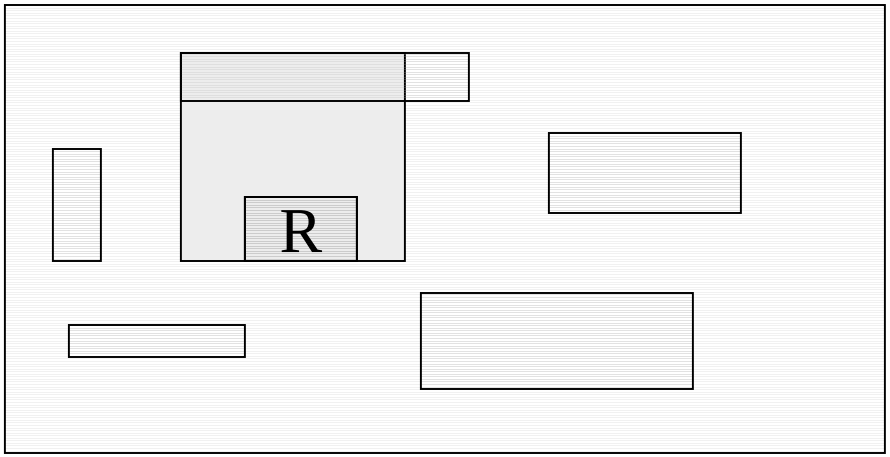
Extremal-Density Window Analysis

- $O(k^2)$ 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 till intersection with other rectangle edge
      - record changes in density
    - repeat sliding R up
    - repeat for all other orientations of W



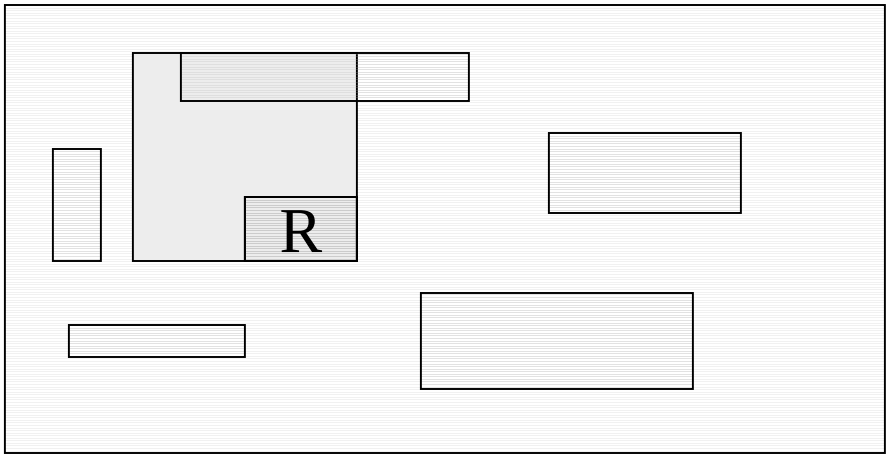
Extremal-Density Window Analysis

- $O(k^2)$ 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 till intersection with other rectangle edge
      - record changes in density
    - repeat sliding R up
    - repeat for all other orientations of W



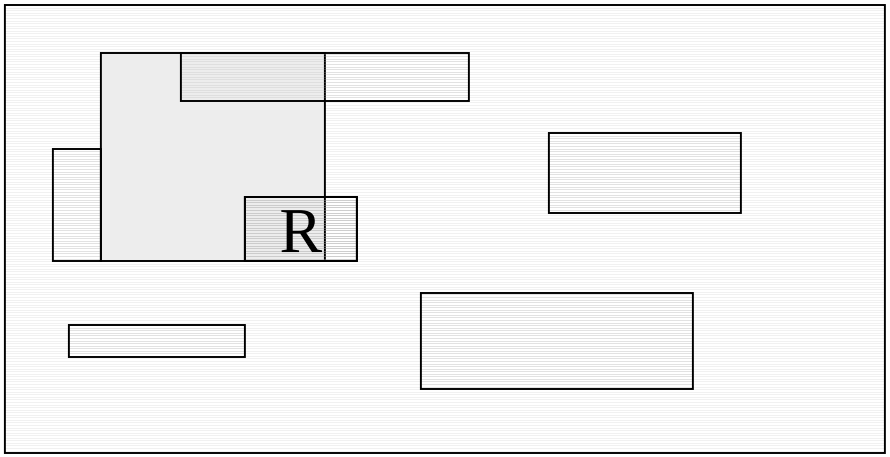
Extremal-Density Window Analysis

- $O(k^2)$ 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 till intersection with other rectangle edge
      - record changes in density
    - repeat sliding R up
    - repeat for all other orientations of W



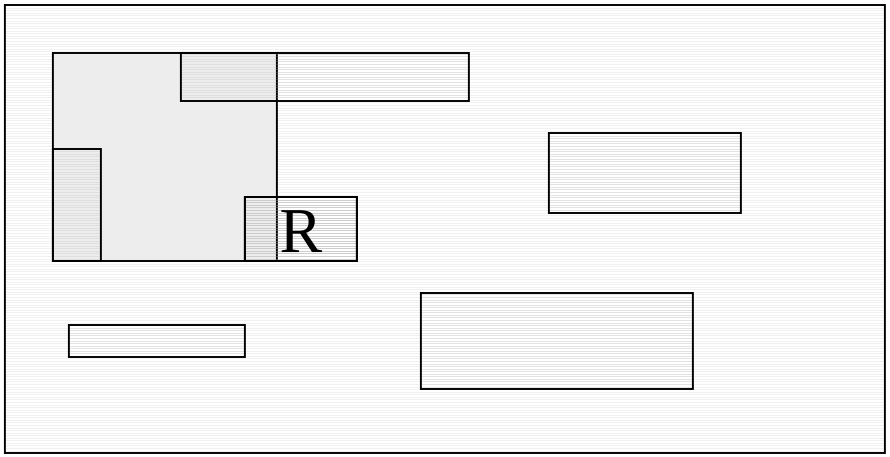
Extremal-Density Window Analysis

- $O(k^2)$ 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 till intersection with other rectangle edge
      - record changes in density
    - repeat sliding R up
    - repeat for all other orientations of W



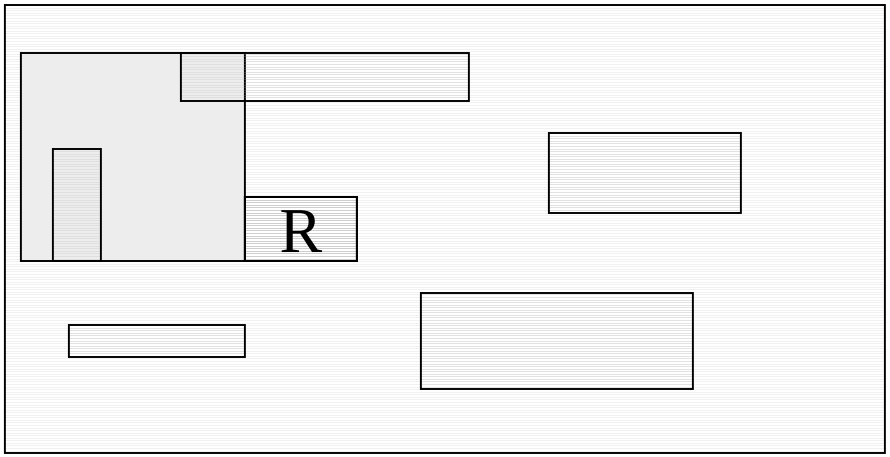
Extremal-Density Window Analysis

- $O(k^2)$ 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 till intersection with other rectangle edge
      - record changes in density
    - repeat sliding R up
    - repeat for all other orientations of W



Extremal-Density Window Analysis

- $O(k^2)$ 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 till intersection with other rectangle edge
      - record changes in density
    - repeat sliding R up
    - repeat for all other orientations of W



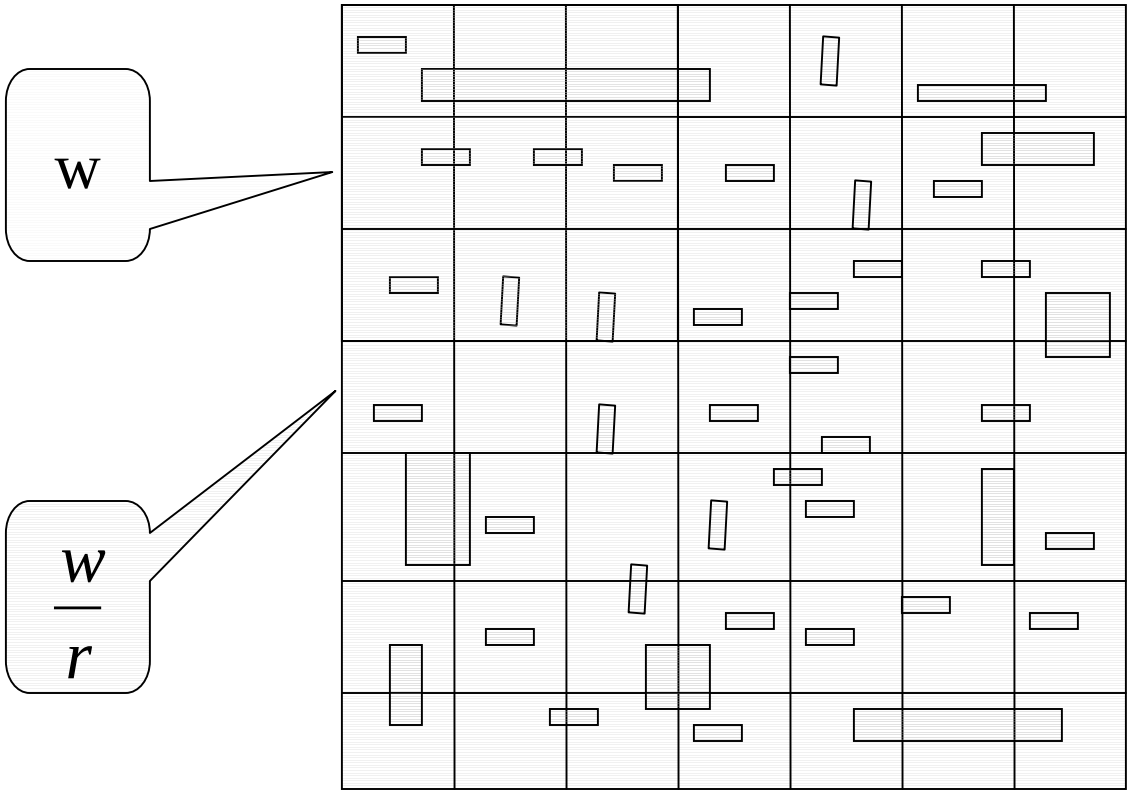
Extremal-Density Window Analysis

- Speedup via ***fixed preprocessing***
  - partition layout into $\frac{n}{w} \times \frac{n}{w}$ squares of size $w \times w$
  - list all rectangles intersecting each $w \times w$ square
- Expected runtime after preprocessing

$$O\left(\left(\frac{n}{w}\right)^2 + k\log k + \left(\frac{wk}{n}\right)^2\right) < O\left(\left(\frac{n}{w}\right)^2 + k\log k + k \cdot w^2\right)$$

Fixed-Dissection Density Analysis

Practical method = check (enforce) density only in $w \times w$ windows of fixed dissection in $\frac{w}{r} \times \frac{w}{r}$ *tiles*



Fixed-Dissection Density Analysis

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

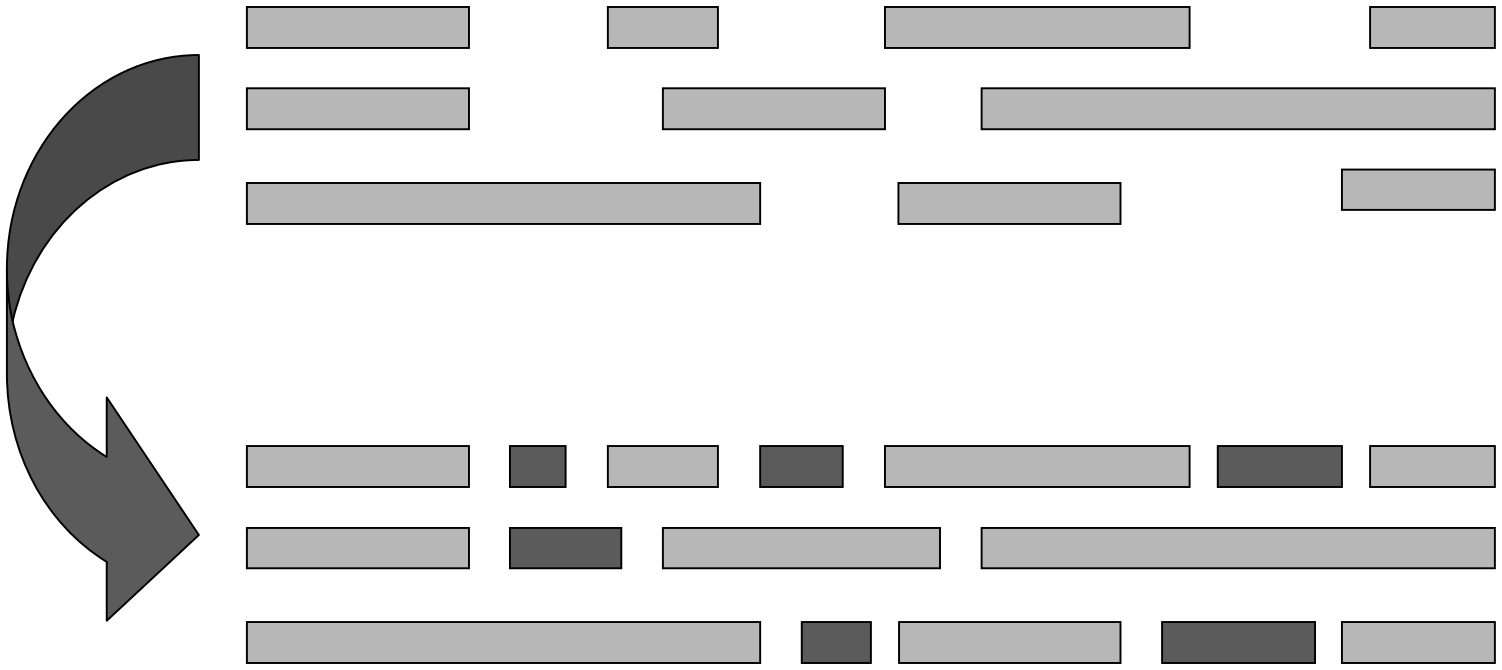
- If all fixed dissection $\frac{w}{r} \times \frac{w}{r}$ ***tiles*** has density between L_a and U_a, then ***any*** $w \times w$ window has density in the range between $\frac{(r-1)^2}{r^2} \cdot L_a + \frac{4(r-1)}{r^2} \cdot \max\{ L_a - .5, 0\} + \frac{4}{r^2} \cdot \max\{ L_a - .75, 0\}$ and $\frac{(r-1)^2}{r^2} \cdot U_a - \frac{4(r-1)}{r^2} \cdot \max\{ U_a - .5, 0\} - \frac{4}{r^2} \cdot \max\{ U_a - .25, 0\}$

Design Rules

- The minimum length rule
  - feature size $> f$
- The minimum separation rule
  - distance between filling pattern and feature $> s$
  - distance between filling elements $> s'$

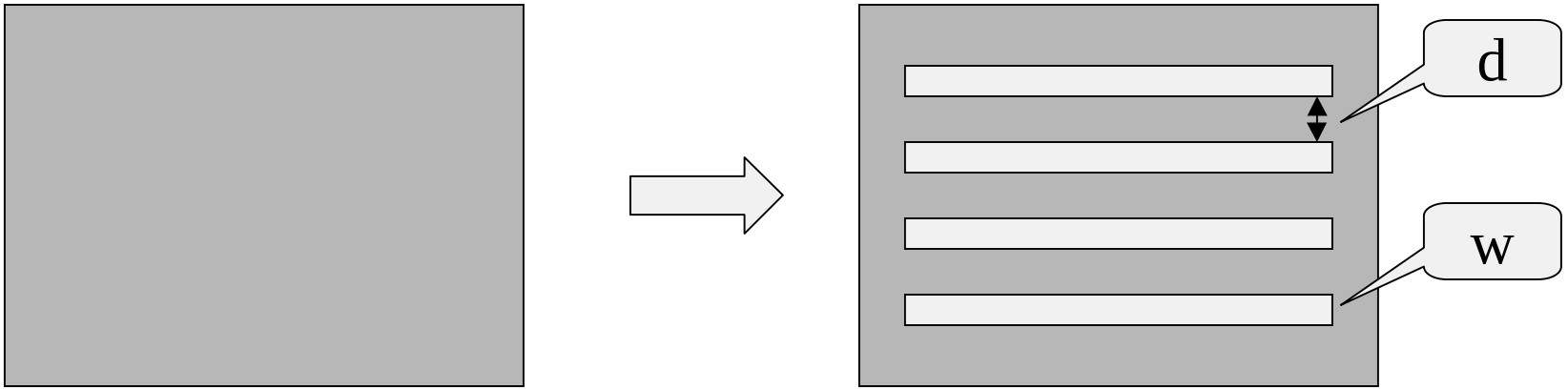
Filling for Wiring-Type Layout

- Density ≤ 50%-60% if separation ≈ row width
- $O(k \log k)$ -algorithm for filling:



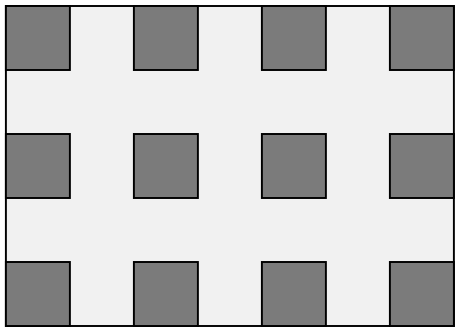
Minimization of Slotting

- Favor filling to slotting
- Slot rectangle lengthwise
  - parallel to current flow
  - of appropriate width *w* and separation *d*

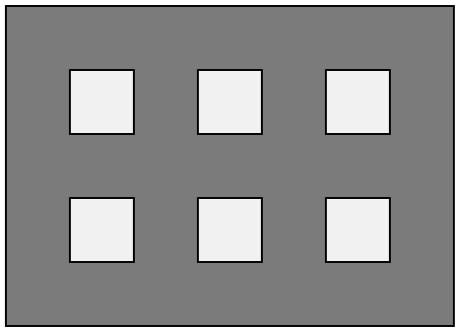


Simultaneous Area and Perimeter

- Minimum length and minimum separation rules $\rightarrow$ patterns are c -polyominoes



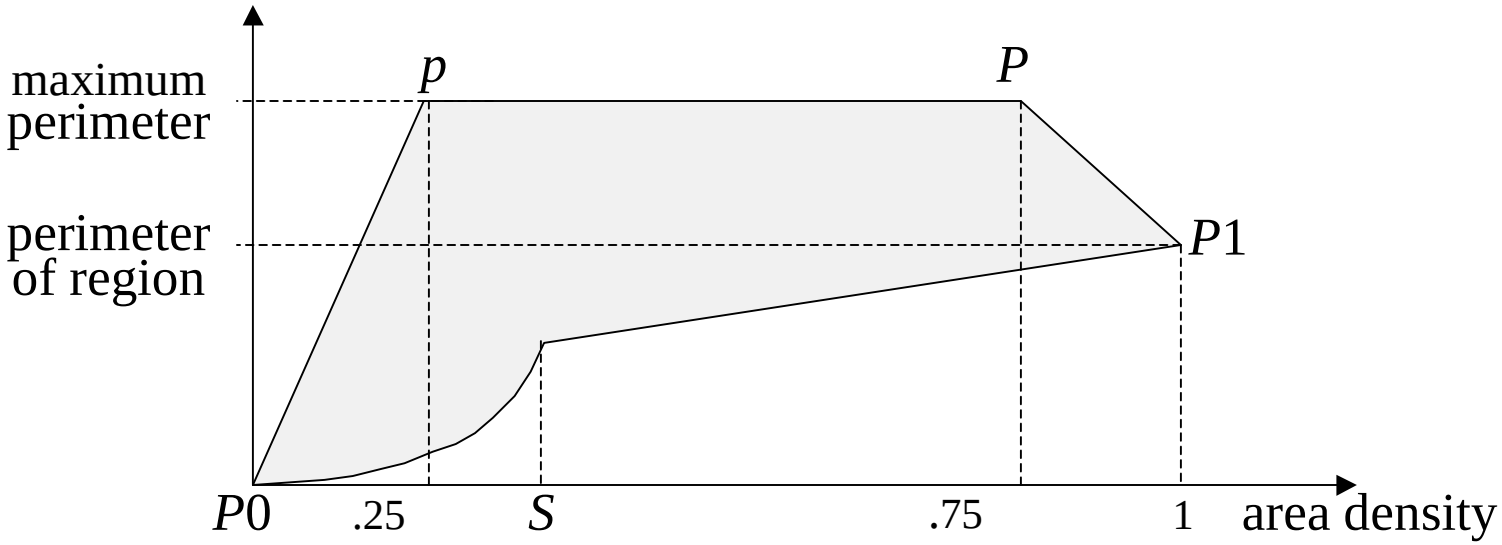
Minimum area pattern p
with maximum perimeter



Maximum area pattern P
with maximum perimeter

Simultaneous Area and Perimeter

- Realizable area and perimeter combinations
  - *P*0 is empty pattern, *P*1 is filled pattern
  - *S* is the maximum square area in region *R*



Computational Experience

Test	Runtime/Density							
k/n/w/Type	ALG3		FD(1)		FD(5)		FD(10)	
	CPU	den	CPU	den	CPU	den	CPU	den
1k/4k/20/R	0.7	.215	0.0	.0203	0.3	0.203	1.1	0.203
1k/4k/80/W	0.4	.138	0.0	.077	0.2	.092	0.7	.092
4k/4k/20/W	3.9	.4	0.1	.35	2.0	.35	10.0	.35
8k/4k/40/R	1.9	.155	0.1	.102	0.8	.15	3.2	.15
8k/2k/40/W	9.3	.478	0.2	.404	1.2	.437	4.7	.449
16k/4k/80/R	14.2	.084	0.2	.07	1.5	.083	5.9	.083
16k/4k/80/W	31.0	.410	0.2	.332	2.2	.349	8.8	.37
Benchmark 1	64.3	.477	0.0	.412	0.4	.436	1.6	.46
Benchmark 2	64.7	.352	0.6	.316	4.8	.324	46.7	.324
Benchmark 3	1430	.856	1.7	.839	12.6	.842	46.7	.852

CPU times and max window densities for ALG3 and approximate fixed dissection algorithm FD(r), r=1, 5, 10, W - wiring type, R-random small

Conclusions

- First formulation of ***filling and slotting problem***
  - layout post-processing
  - layout optimization for manufacturability
- Effective algorithms for
  - density analysis
    - exact (ALG3)
    - approximate (fixed dissection)
  - filling/slotting synthesis
- Integration in software environment
- Future work
  - more density rules and test cases
  - more efficient analysis and synthesis
  - dynamic rectangle insertion/deletion