Stitch Aware Detailed Placement for Multiple E-Beam Lithography

Yibo Lin[1], Bei Yu[2], Yi Zou[1,3], Zhuo Li[4],
Charles J. Alpert[4], and **David Z. Pan**[1]

[1]ECE Department, University of Texas at Austin
[2]CSE Department, Chinese University of Hong Kong
[3]CEAS Department, Nanjing University
[4]Cadenace Design Systems, Inc.

This work is supported in part by NSF and SRC

Outline

- Introduction
- Previous Work
- Problem Formulation
- Stitch Aware Detailed Placement
- Experimental Results
- Conclusion

Introduction

- Technology Scaling

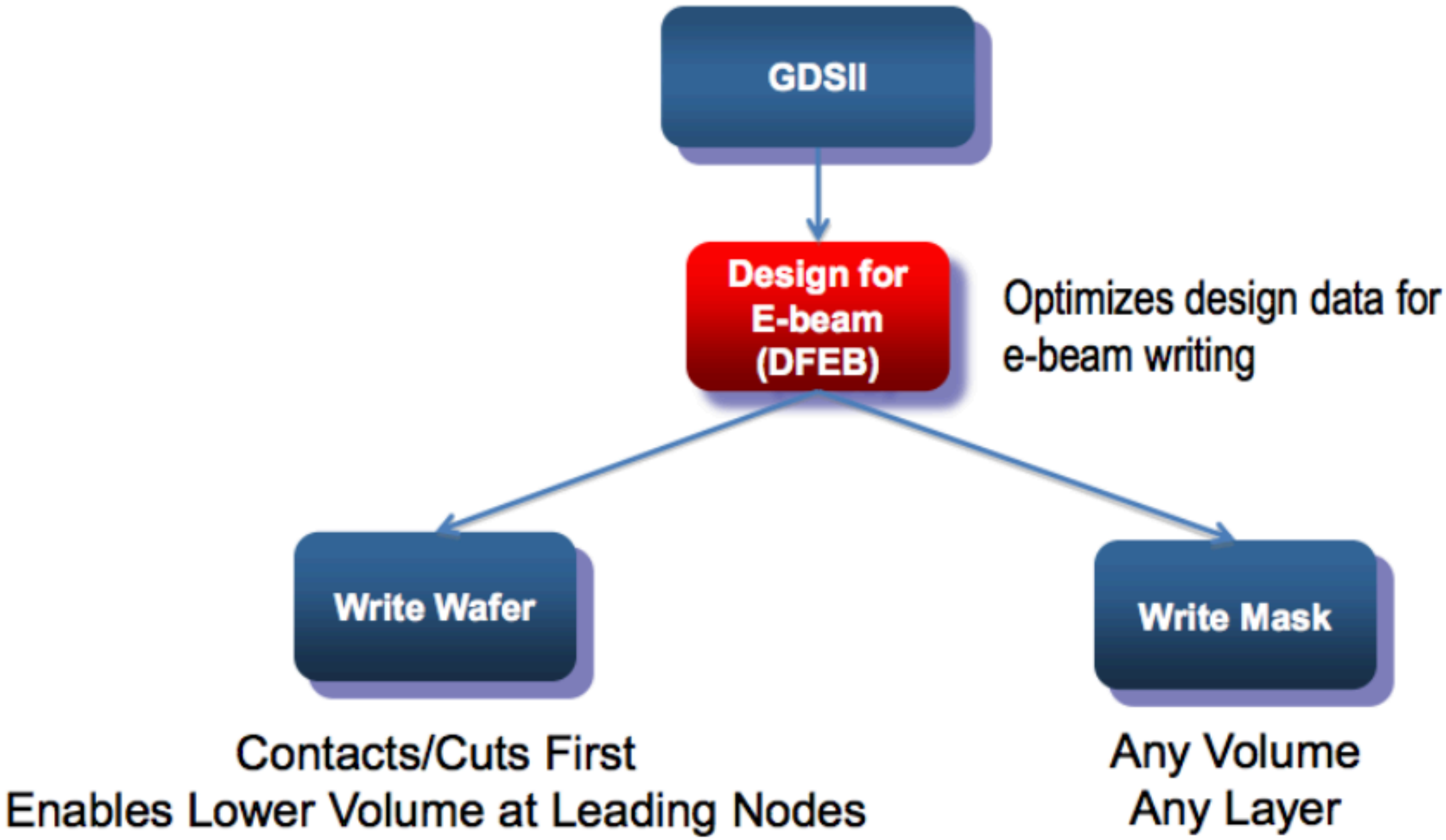
Uni-directional parallel line/space patterning techniques																			
	CD	40	38	36	34	32	30	28	26	24	22	20	18	16	14	12	10	8	6
Exposure tool	Pitch	80	76	72	68	64	60	56	52	48	44	40	36	32	28	24	20	16	12
Immersion	→																		
Immersion	→					19				20									
Immersion	→										2								
Immersion	→													5		1			
EUV	→														18				
EUV	→													4			6		
Immersion	→	Large features do not phase separate well by DSA														3	12		
ArF, EUV, E-beam	→																	11	
Nanoimprint	→															13			14
High NA EUV	→																	17	
E-beam	→		7							8				15			16		
E-beam	→	Large features do no phase separate well by DSA													9	10	12		

Consenses that technique has been used in production
Published demonstrations from potential deployable equipment show opportunity for production
Simulations, surface images, or research grade demonstration suggest potential for extendability

[Courtesy ITRS]

E-Beam Lithography

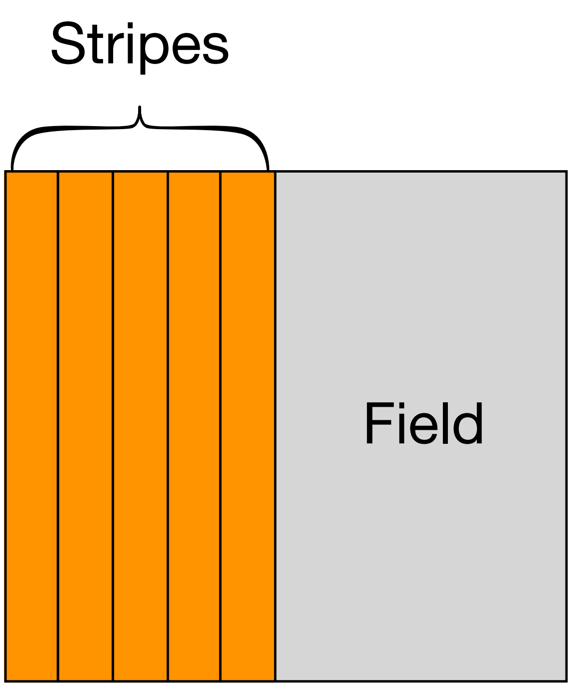
- Direct-write or mask?



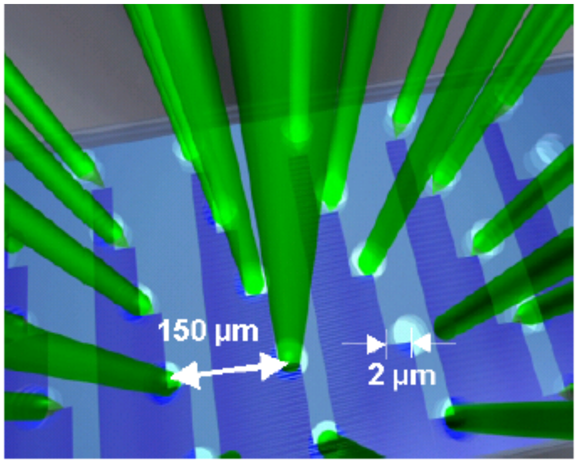
[Courtesy E-beam Initiative]

Multiple E-Beam Lithography

- Massively-Parallel e-beam writing
  - Each stripe has width of 50~200 microns
  - Stitching region has a width around 15nm [Berg+,SPIE'11]
  - Field stitching



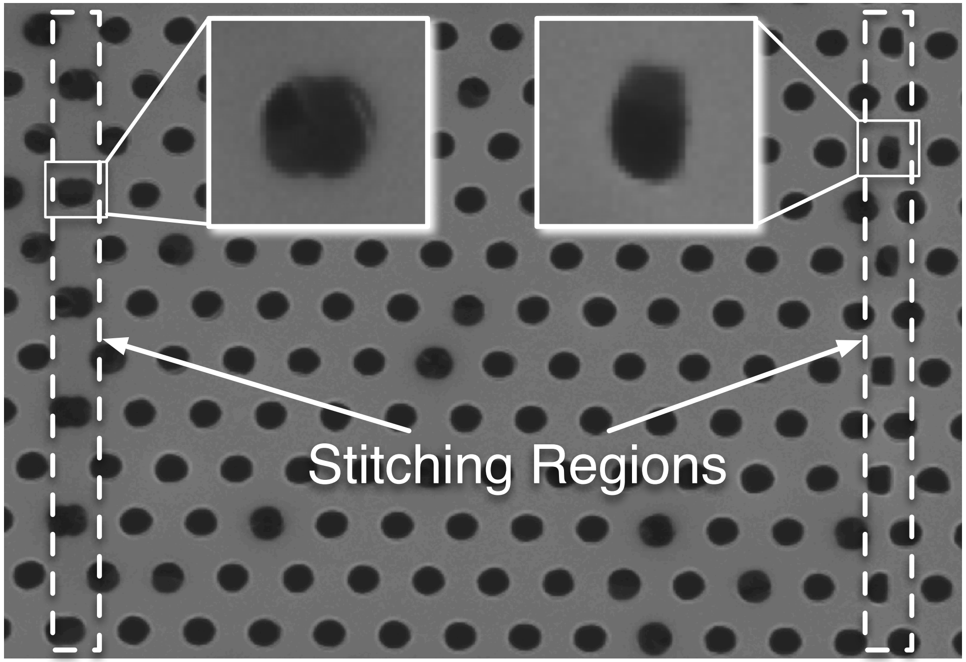
[Fang+,DAC'13]



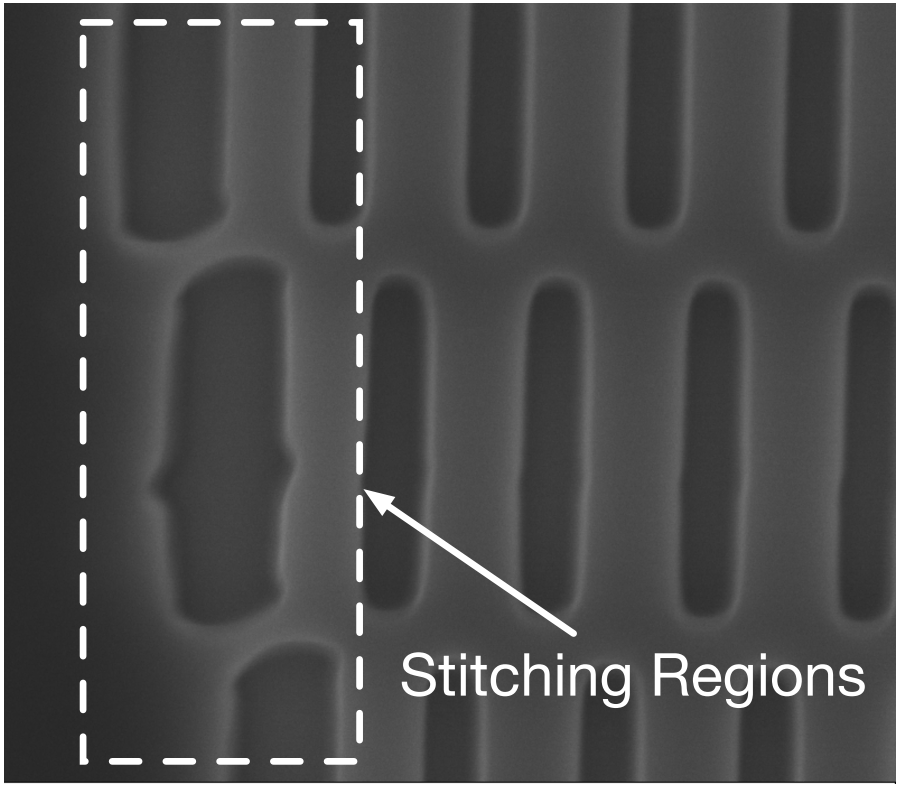
MAPPER Lithography System

Field Stitching

- SEM figures showing stitches at boundaries of beam stripes



Holes



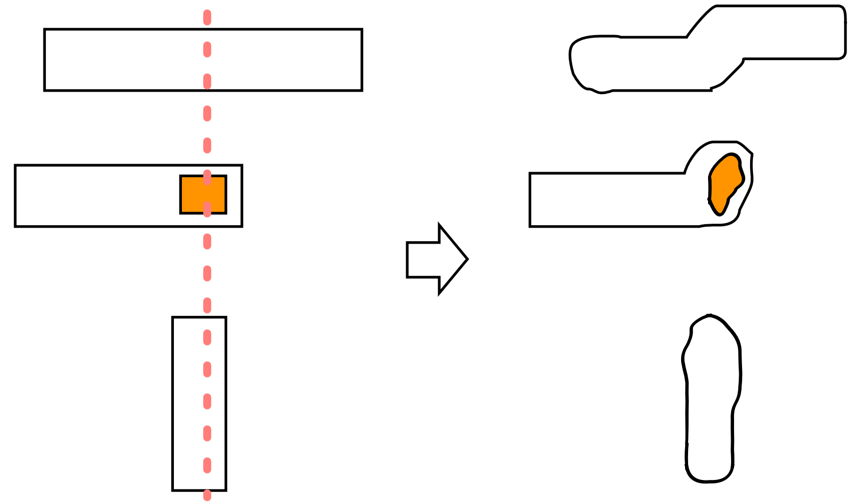
Lines

Previous Work

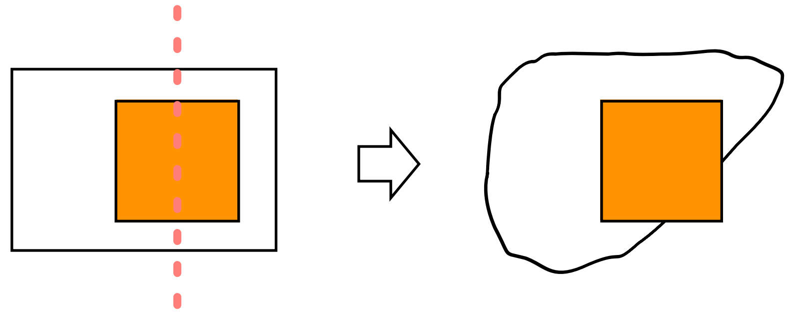
- Stitch aware routing for MEBL
  - [Fang+,DAC'13], [Liu+,TCAD'15]
- TPL aware placement
  - [Yu+,TCAD'15], [Kuang+,TVLSI'15], [Chien+,TCAD'15]
  - [Tian+,ICCAD'14], [Lin+,ISPD'15]
  - TPL applies different constraint to placement from MEBL
- No placement algorithm addressing MEBL stitch constraint yet

Stitch Errors

- Defects on vias and vertical wires

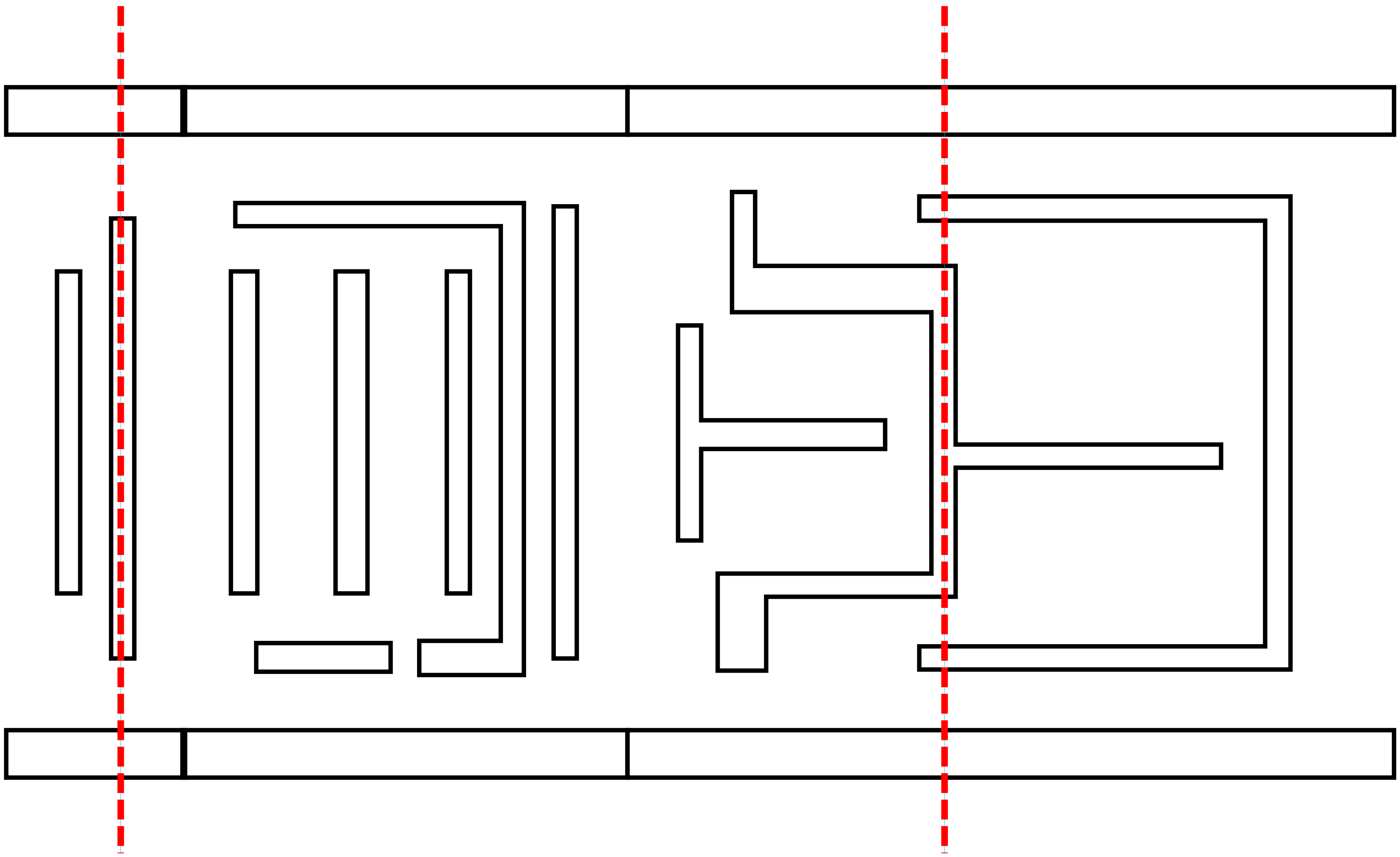


- Defects on short polygons



[Fang+,DAC2013]

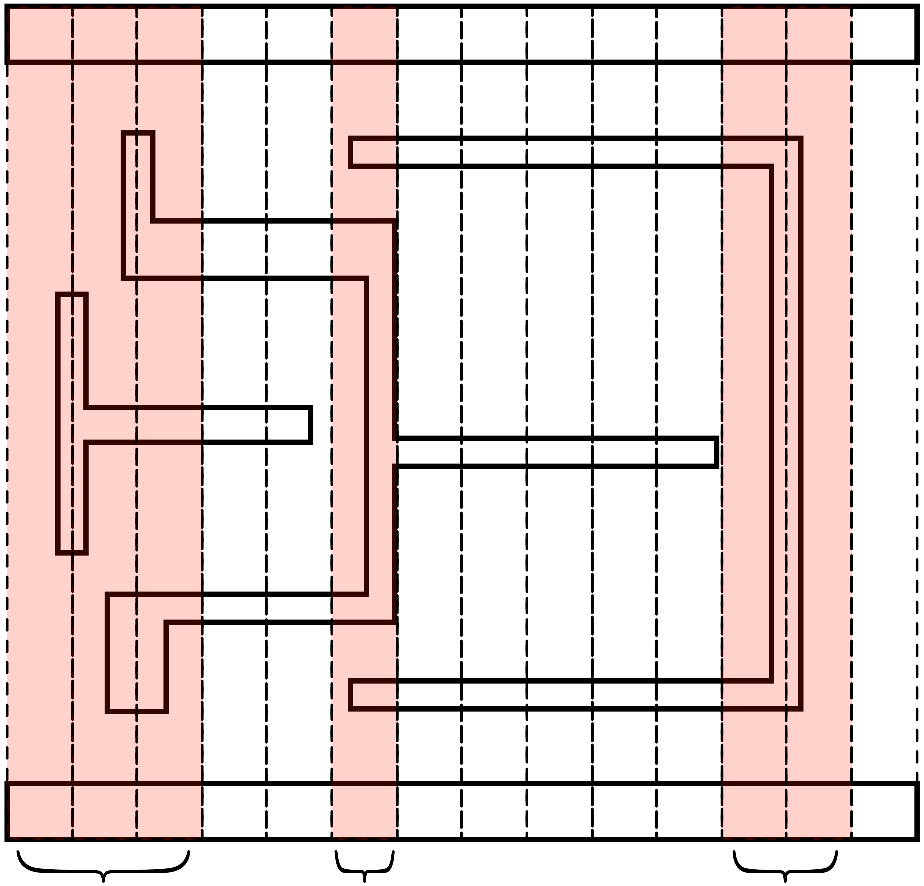
Stitch Errors within Standard Cell



Resolve stitch errors by proper placement

Dangerous Site Representation

- A cell is divided into sites (poly pitch)
- Sites that contain susceptible segments are marked as "dangerous sites"



Problem Formulation

- Input
  - Initial placement
  - Dangerous site information for each standard cell (precomputed)
- Output
  - New placement with optimized wirelength and minimum stitch errors
  - MEBL friendliness

Single Row Placement & Previous Work

- Given a set of ordered cells c_1, c_2, …, c_n, place cells horizontally to minimize objectives such as wirelength or movement
- Previous work on single row algorithm
  - Conventional objectives
    - [Brenner+,DATE’00], [Kahng+,GLSVLSI’04], Abacus [Spindler+,ISPD’08], [Taghavi+,ICCAD’10]
  - TPL awareness
    - [Yu+,ICCAD’13]: O(mnK)
    - [Kuang+,ICCAD’14]

Note: $\tau = 10, \phi = 1, \upsilon = 1$ in the experiment

Single Row Placement

- Given a set of ordered cells $c_1, c_2, \ldots, c_n$, with maximum cell displacement M
  - Minimize wirelength and stitch errors
  - An algorithm supports a cost function generalizes wirelength, movement and stitch errors

Movement

$$cost_i(p_i) = \tau \cdot WL(p_i) + \phi \cdot MOV(p_i) + \nu \cdot SP(p_i)$$

Wirelength cost

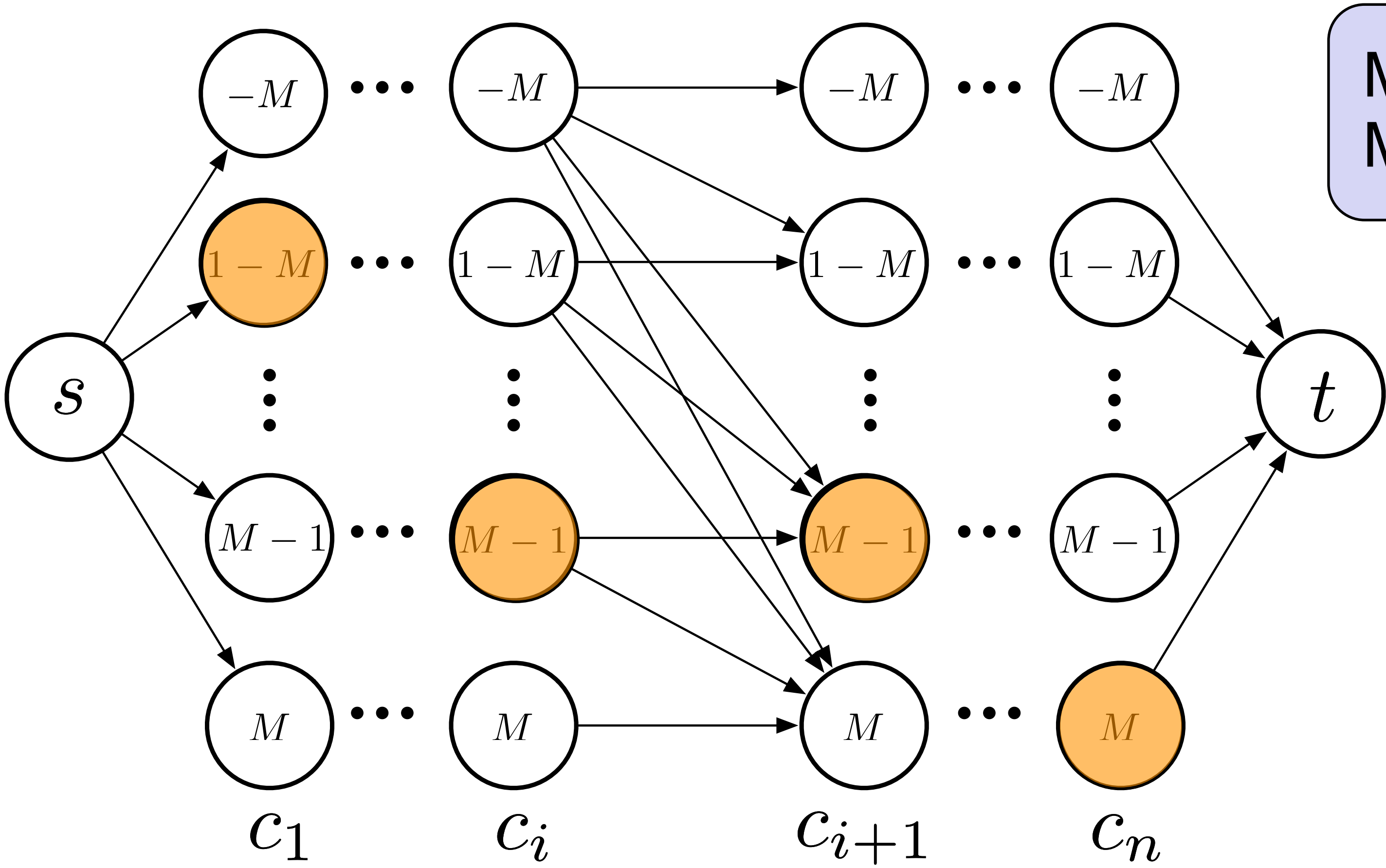
Stitch error penalty

$$SP(p_i) = \begin{cases} 0, & no\ stitch\ error \\ large\ number, & stitch\ error \end{cases}$$

Note: $\tau = 10, \phi = 1, \upsilon = 1$ in the experiment

Single Row Placement

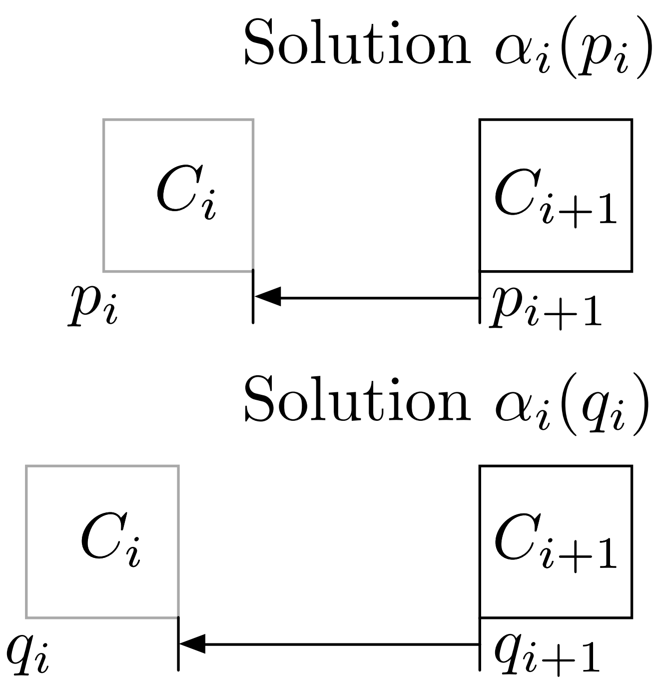
- Given a set of ordered cells $c_1, c_2, \ldots, c_n$, with maximum cell displacement M
  - Shortest path solved by dynamic programming
  - **$O(nM^2)$**

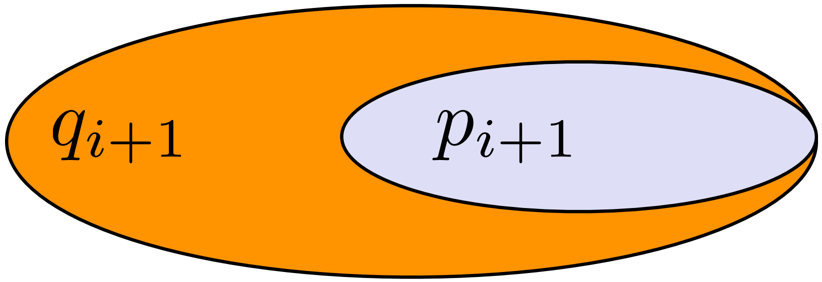


M < 10?
M > 30?

Speedup with Pruning Techniques

- Pruning technique 1
  - Let $t_i(p_i)$ denote the cost of placement solution from c_1 to c_i in which c_i is placed at p_i
  - Comparing two solutions $\alpha_i(p_i)$ and $\alpha_i(q_i)$, if $t_i(p_i) \geq t_i(q_i)$ and $p_i \geq q_i$, then $\alpha_i(p_i)$ is inferior to $\alpha_i(q_i)$.
  - Prune inferior solutions

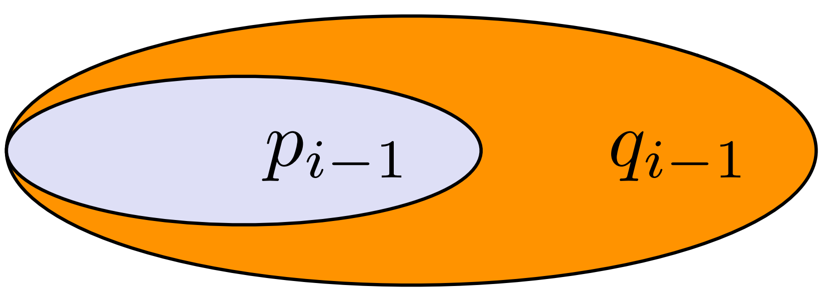




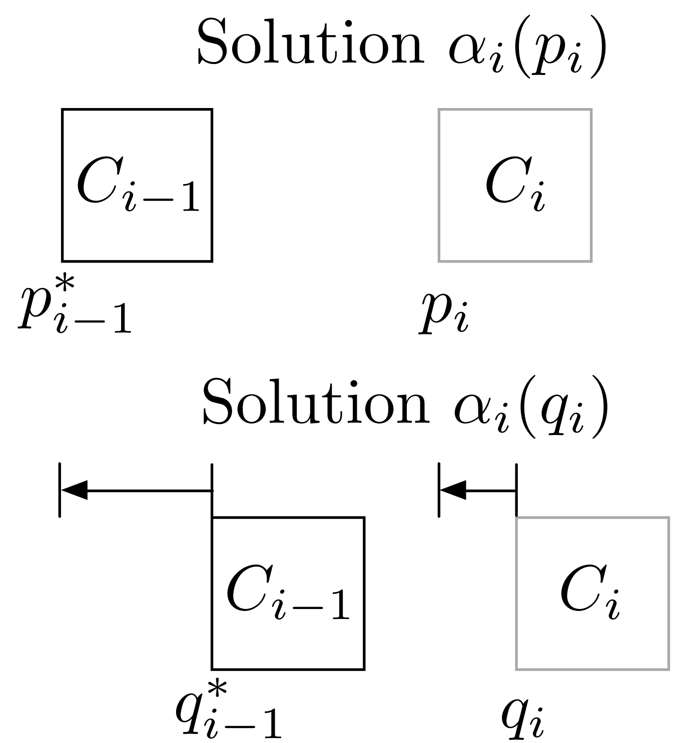
Value sets of p_{i+1} and q_{i+1}

Speedup with Pruning Techniques

- Pruning technique 2
  - Let p_{i-1}^* be the optimal position of cell c_{i-1} when cell c_i is placed at p_i
  - Let q_{i-1}^* be the optimal position of cell c_{i-1} when cell c_i is placed at q_i
  - If $q_i \geq p_i$, then $q_{i-1}^* \geq p_{i-1}^*$
  - Reduce searching ranges

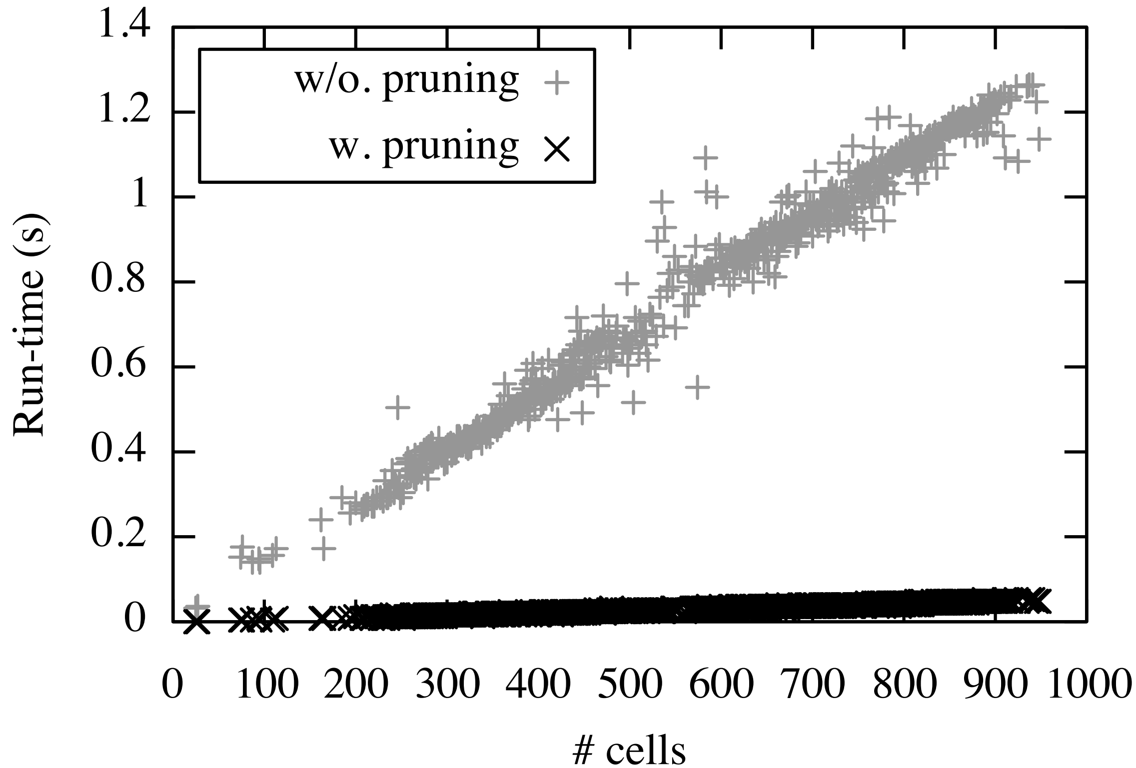


Value sets of p_{i-1} and q_{i-1}



Effectiveness of Speedup Techniques

- **O(nM)** complexity
  - Requirements: $cost_i(p_i)$ only depends on p_i
  - 30x speedup
  - Keep optimality



Resolve Stitch Errors in Dense Regions

- Global swap to smooth out density
  - $score(c_i, c_j) = \Delta sHPWL - \lambda \cdot P_{ds} - \mu \cdot P_{ov}$

Overlap penalty

sHPWL change

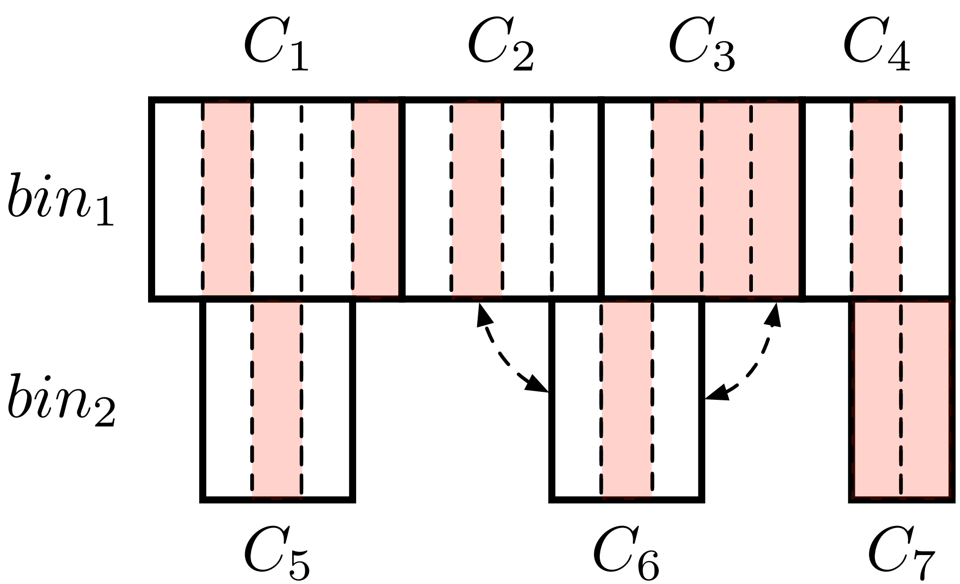
Normalized penalty of dangerous site density

$$P_{ds} = \max(0, |D'_{ds}(i) - D'_{ds}(j)| - |D_{ds}(i) - D_{ds}(j)|) \cdot A_b$$

$D_{ds}(i)$: the density of dangerous sites in bin B_i before swap

$D'_{ds}(i)$: the density of dangerous sites in bin B_i after swap

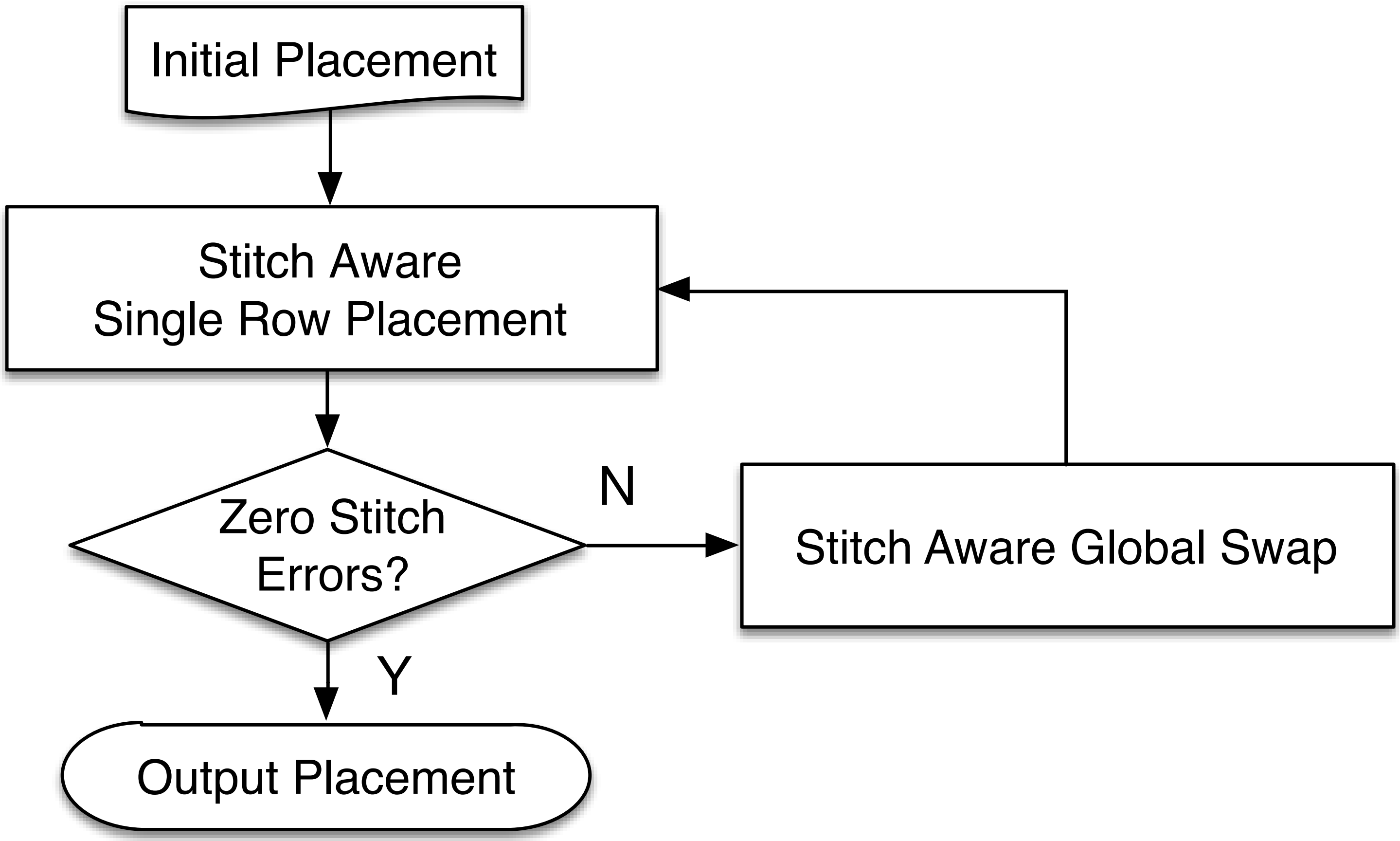
A_b: bin area



Achieve better density of dangerous sites

Note: $sHPWL = HPWL \times (1 + \alpha \times P_{ABU})$ from ICCAD 2013 Contest

Overall Flow



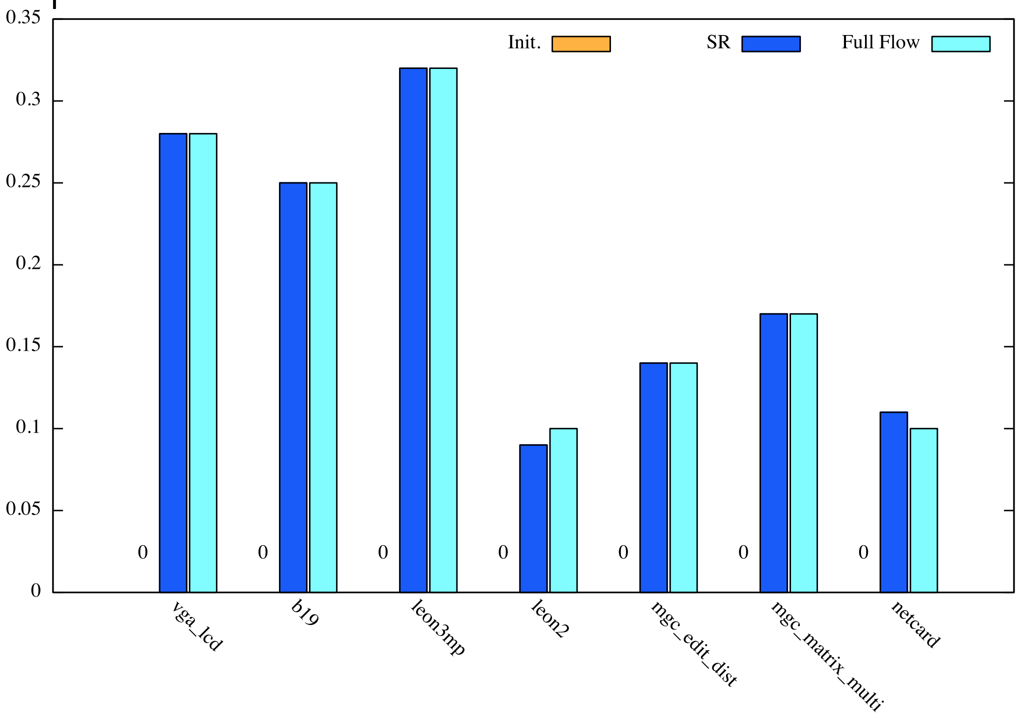
Experimental Environment Setup

- Implemented in C++
- 8-Core 3.4GHz Linux server with 32GB RAM
- ICCAD 2014 contest benchmark
  - Mapped to Nangate 15nm Standard Cell Library
  - Legalized with RippleDP [Chow+,ISPD'14]

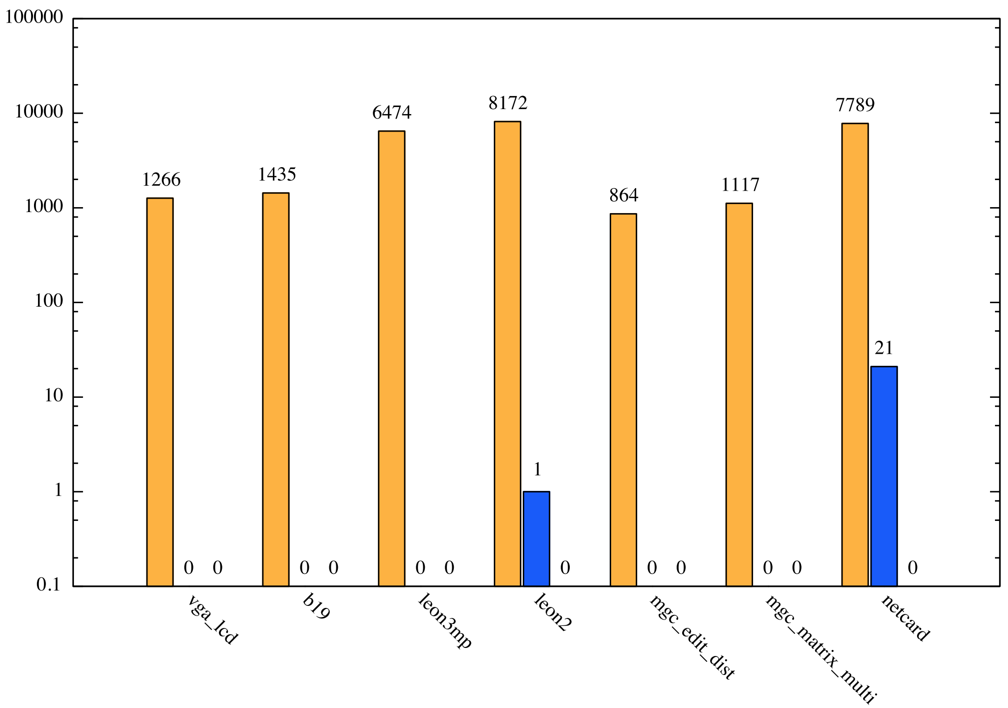
Design	#cells	#nets	#blockages
vga_lcd	165K	165K	0
b19	219K	219K	0
leon3mp	649K	649K	0
leon2	794K	795K	0
mgc_edit_dist	131K	133K	13
mgc_matrix_mult	155K	159K	16
netcard	959K	961K	12

Experimental Results

Wirelength Improvement %



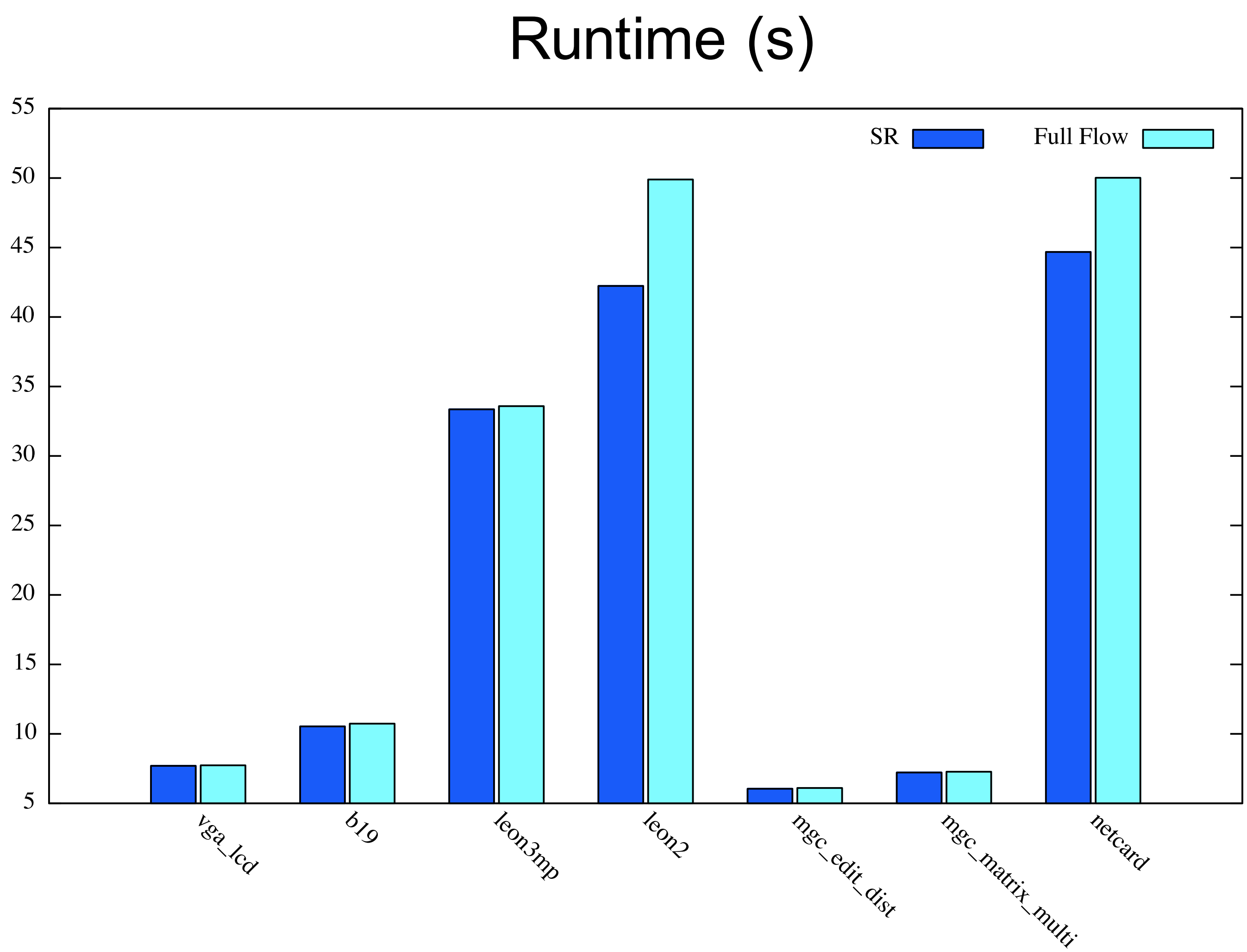
Final Stitch Errors



Init.: initial input placement
SR: single row algorithm only
Full Flow: apply full flow including single row algorithm and global swap

Runtime Comparison

- Full flow is slightly slower than SR
  - Only apply to regions still containing stitch errors



Conclusion

- Methodology to handle e-beam stitch errors during detailed placement stage
- A linear time single row algorithm with highly-adaptable objective functions
- Can be utilized in existing CAD tool on optimizing: Wire-length; Routability; Congestion, etc.
- Future work
  - Consider interaction between placement and routing for EBL friendliness

Thanks