On Simulated Annealing in EDA

A tribute to Prof. C. L. Liu at ISPD-2012

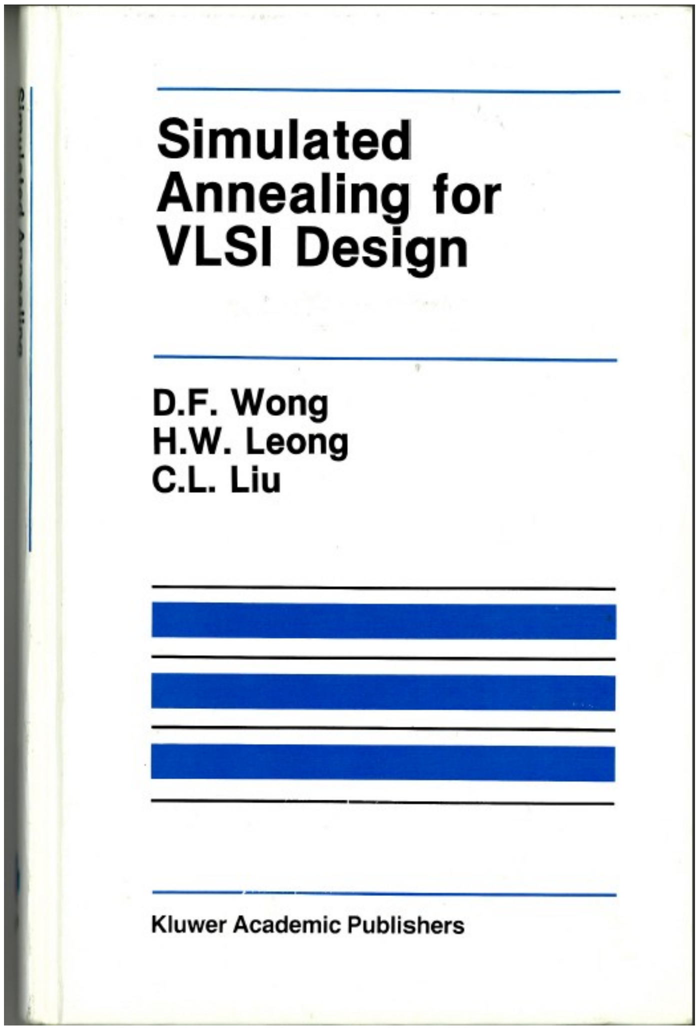
Martin D.F. Wong

Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign

ICCAD Panel on Simulated Annealing

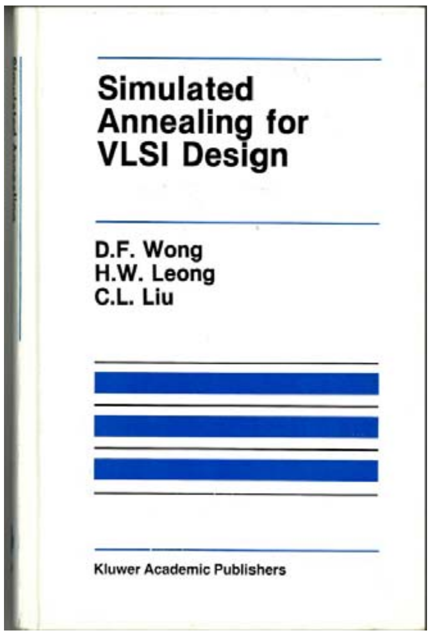
1987 ICCAD Panel: “Is Simulated Annealing Practical for CAD?”

SA Research in Liu's Group



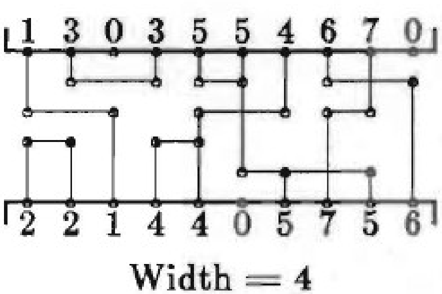
1988

Preface of the Book

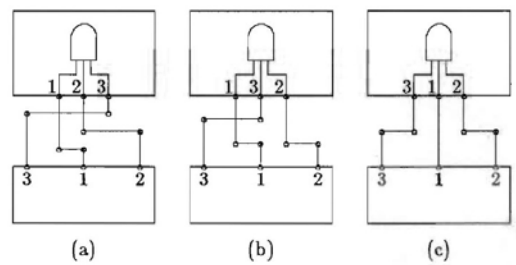


"We hope that our experiences with the techniques we employed, some of which indeed bear certain similarities for different problems, could be useful as hints and guides for other researchers in applying the method To the solutions of other problems."

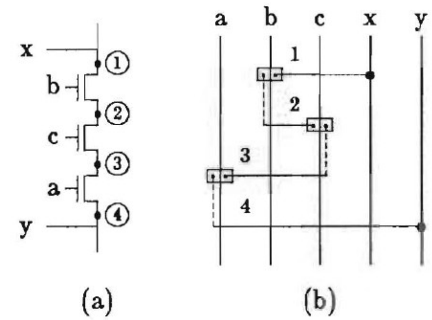
Studied Many Problems



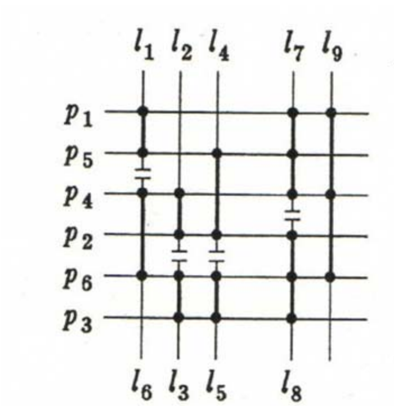
Channel Routing
ICCAD-85



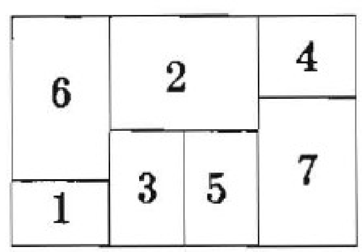
Pin Assignment
ICCD-85, INTEGRATION-87



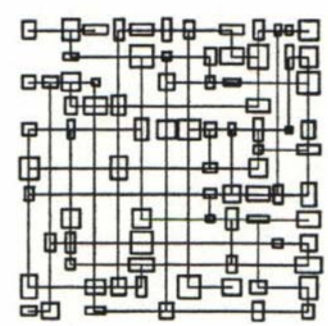
Gate Matrix Layout
ICCAD-86



PLA Folding
CICC-85,JSSC-87



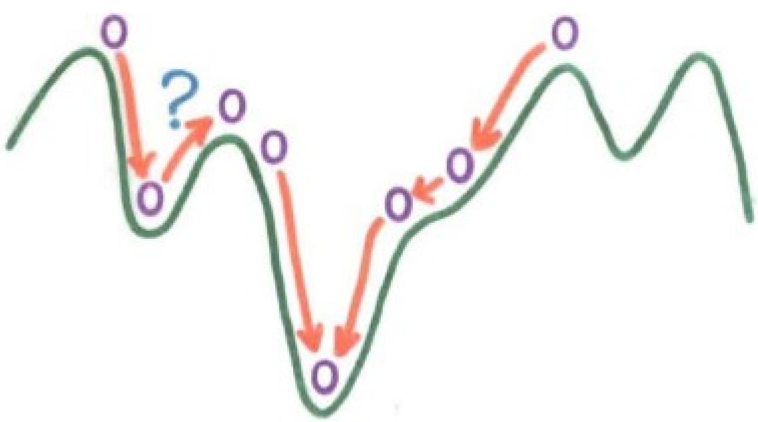
Floorplan Design
DAC-86,ICCAD-87



Array Optimization
DAC-87

SIMULATED ANNEALING

- Local Search
- Non-zero probability for "up-hill" moves
- Probability* depends on magnitude of "up-hill" movement
- Probability* depends on total search time



$$^*\mathrm{Prob}(S \rightarrow S') = \begin{cases} 1 & \text{if } \Delta C \leq 0 \\ e^{-\frac{\Delta C}{T}} & \text{if } \Delta C > 0 \end{cases}$$

$T = T_0, T_1, T_2, \cdots\cdots$

$(T_i = r^i T_0,\ r < 1)$

DAC-86

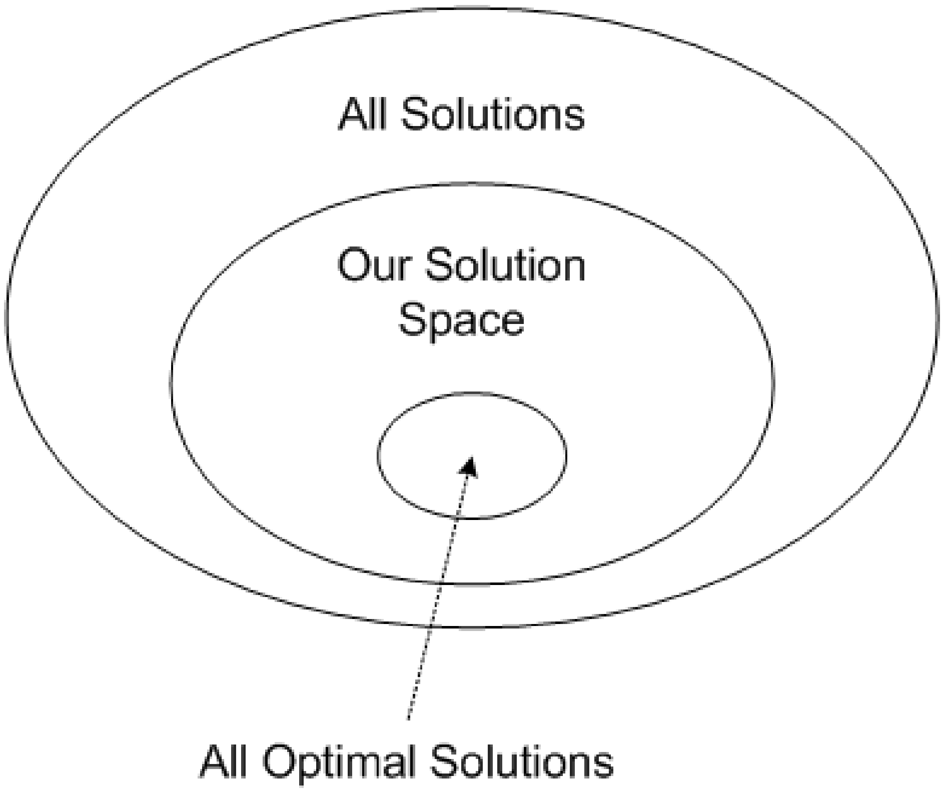
Basic Ingredients for SA

- Solution Space
- Neighborhood Structure
- Cost Function
- Annealing Schedule

DAC-86

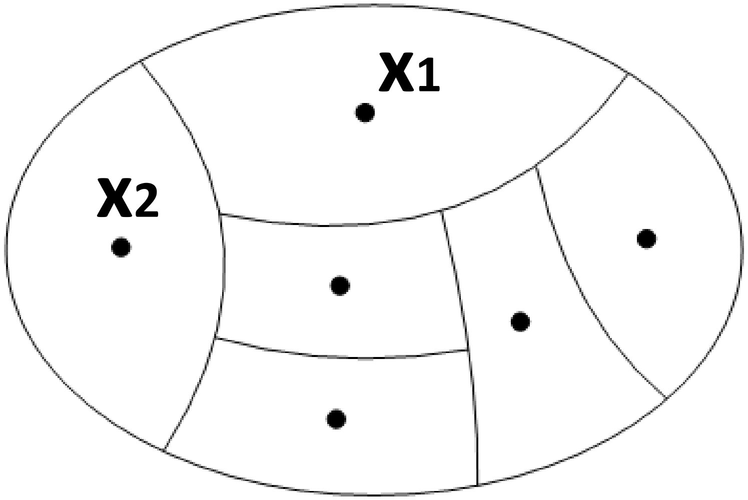
Methodology

- Significant reduction in solution space size
- Keep optimal solutions

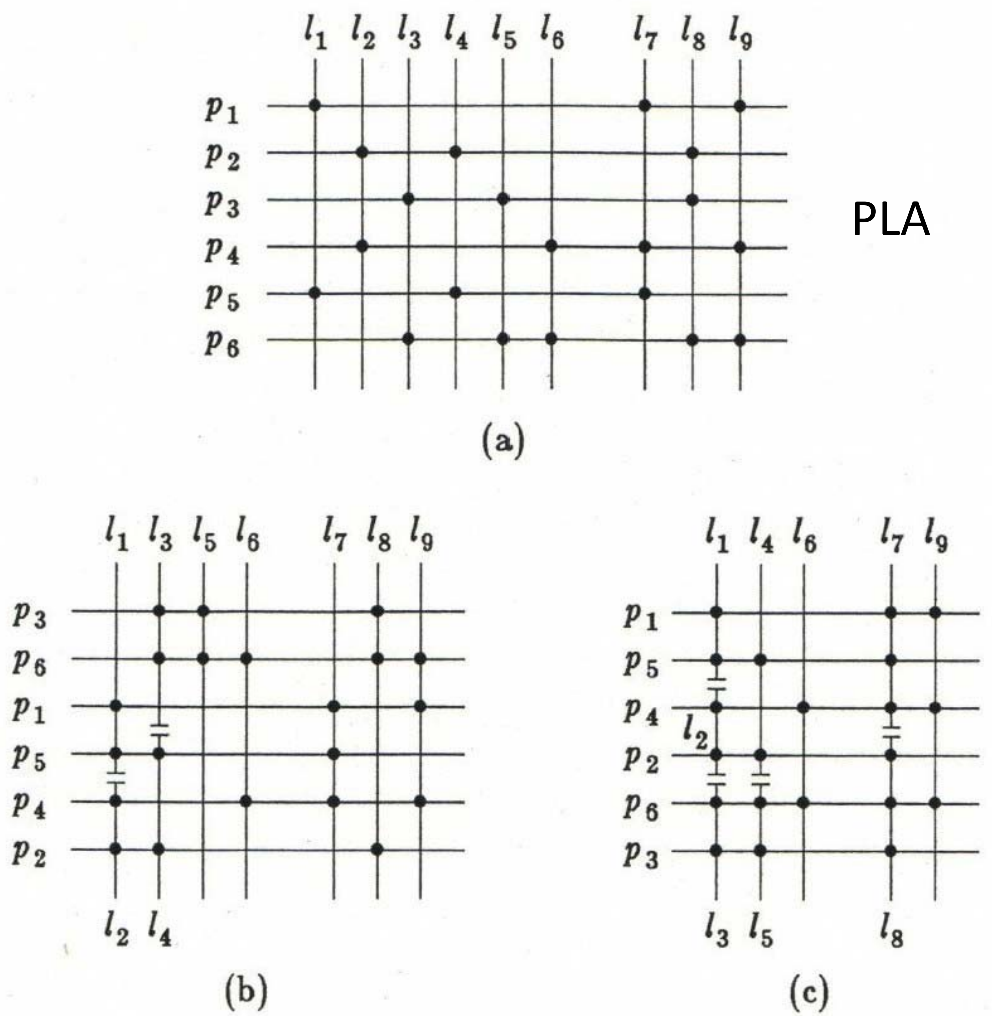


Methodology

- Solution space partitioning: $\mathbf{S} = \mathbf{S_1} + \mathbf{S_2} + \ldots + \mathbf{S_n}$
- Each $\mathbf{S_k}$ is a tractable optimization problem
- min $\mathbf{S}$ = min { $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}$ } where $\mathbf{x_k}$ = min $\mathbf{S_k}$
- New solution space $\mathbf{S'}$ = { $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}$ } =
  { $\mathbf{S_1}$, $\mathbf{S_2}$, … , $\mathbf{S_n}$ }
- Encoding for { $\mathbf{S_1}$, $\mathbf{S_2}$, … , $\mathbf{S_n}$ }



PLA Folding



PLA

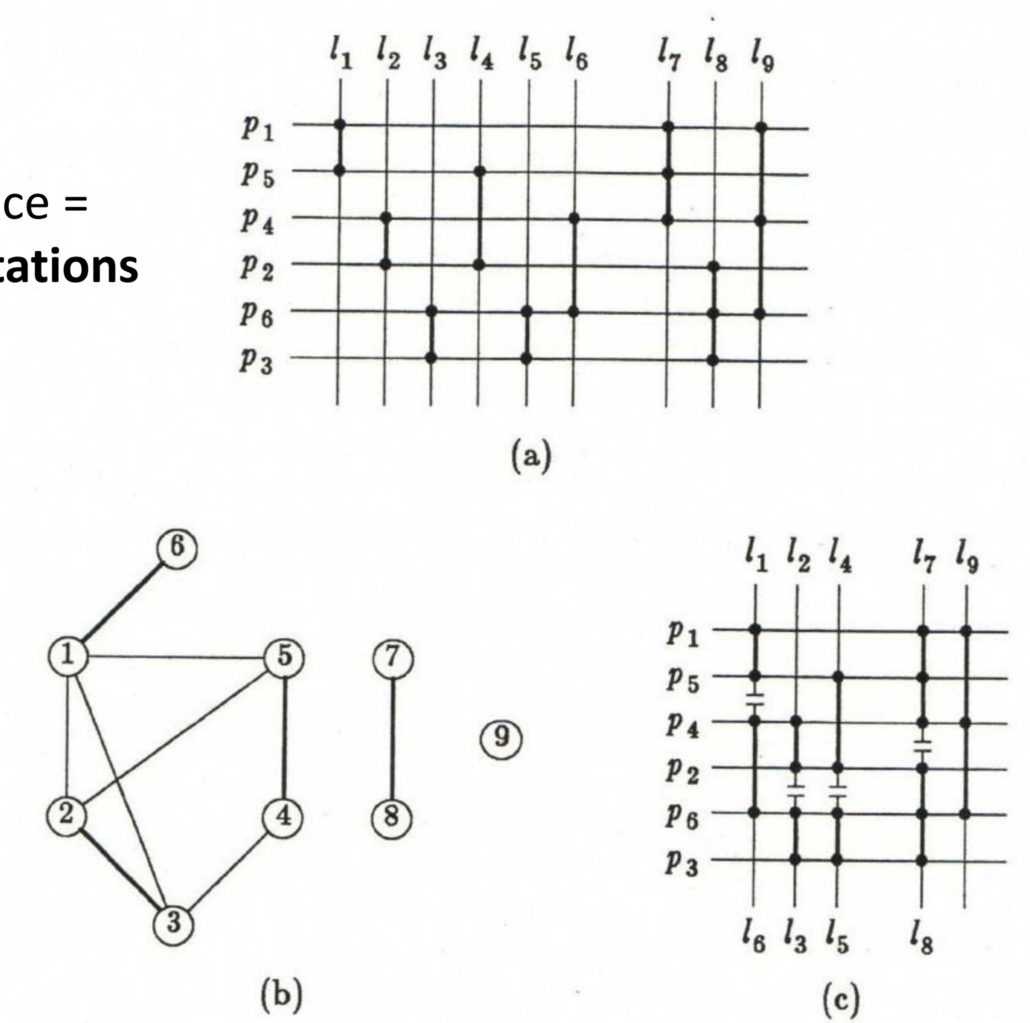
simple folding

multiple folding

PLA Folding

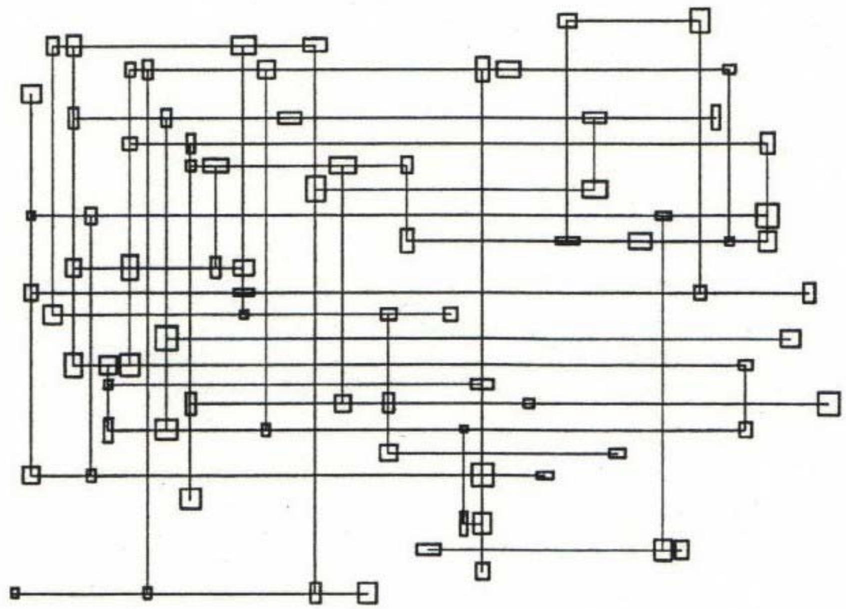
Solution Space =
Row Permutations

maximum
matching

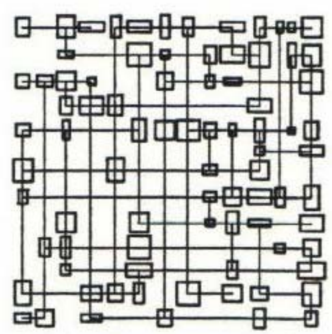


simple folding

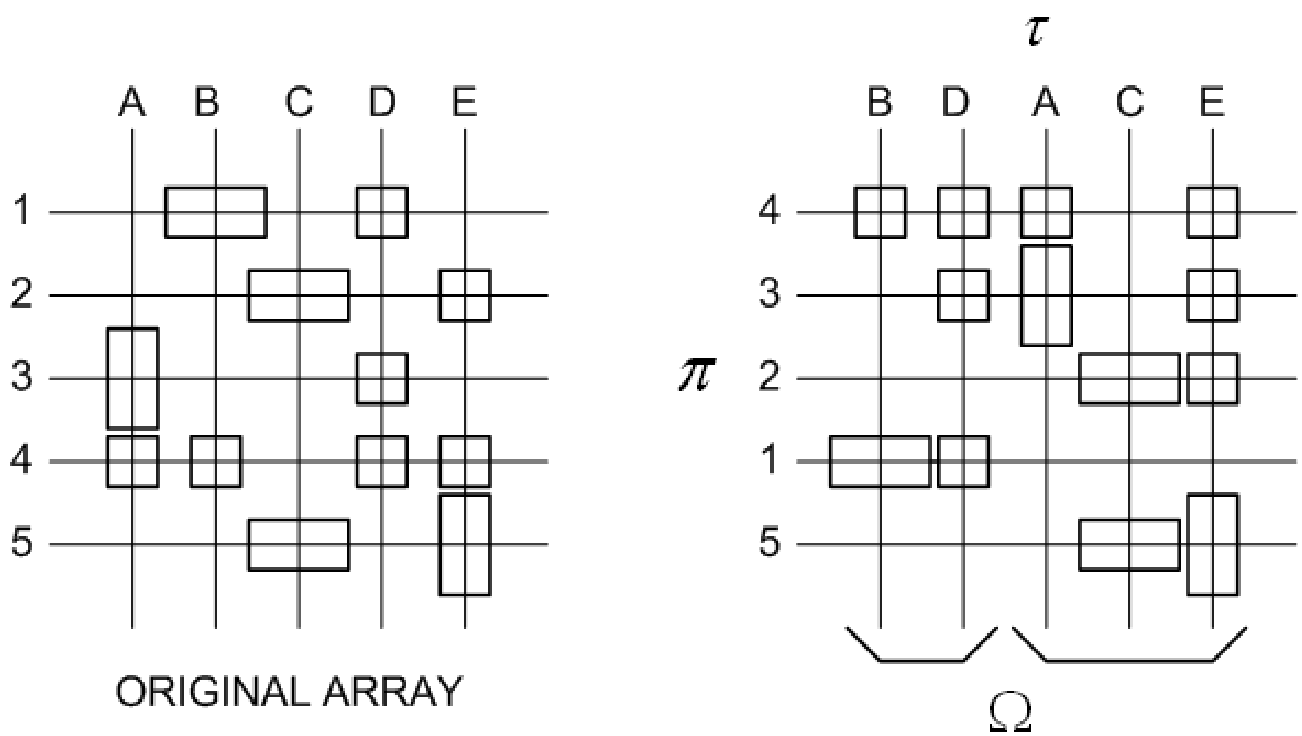
Array Optimization



Row Permutations +
Column Permutations +
2D Compactions



Array Optimization



Solution Space = $\left\{\left(\pi, \tau, \Omega\right)\right\}$

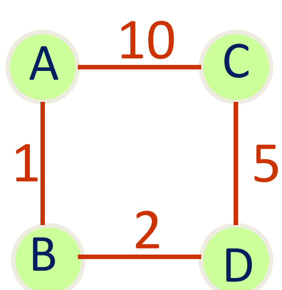
Floorplan Design

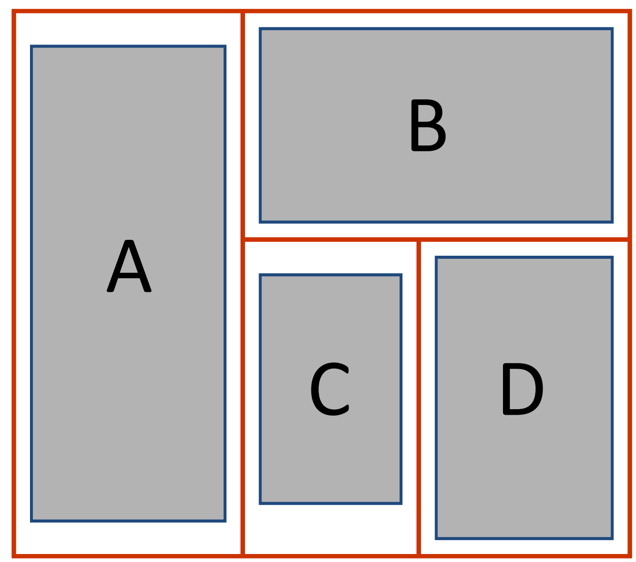
Pack modules on a rectangular chip to optimize total area, interconnect cost and other performance measure.

Module:

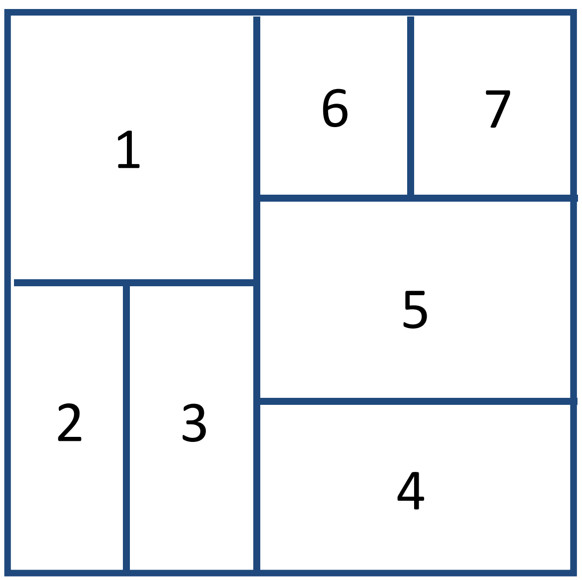
- Hard modules
- Soft modules

Connectivity:

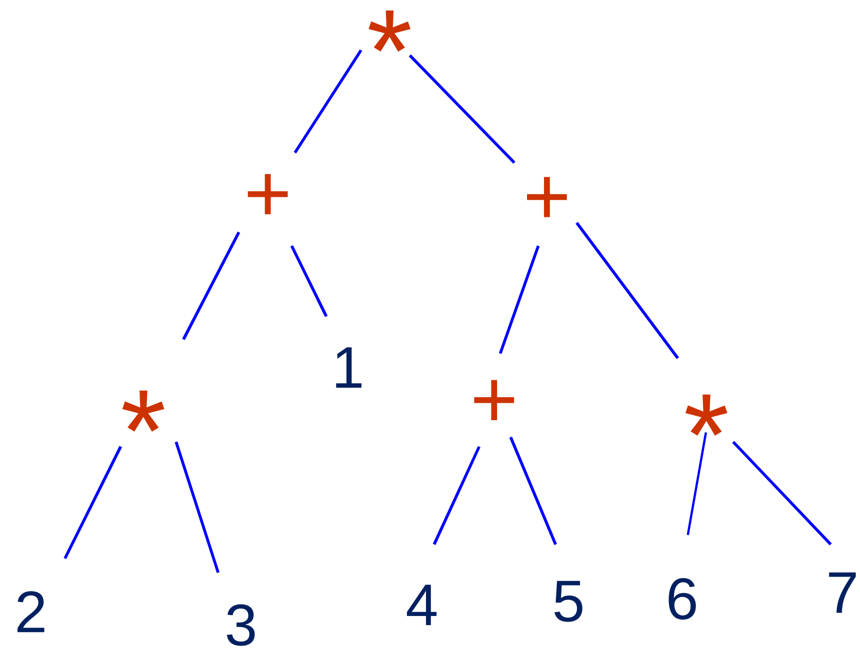




Algorithm



Slicing Floorplan

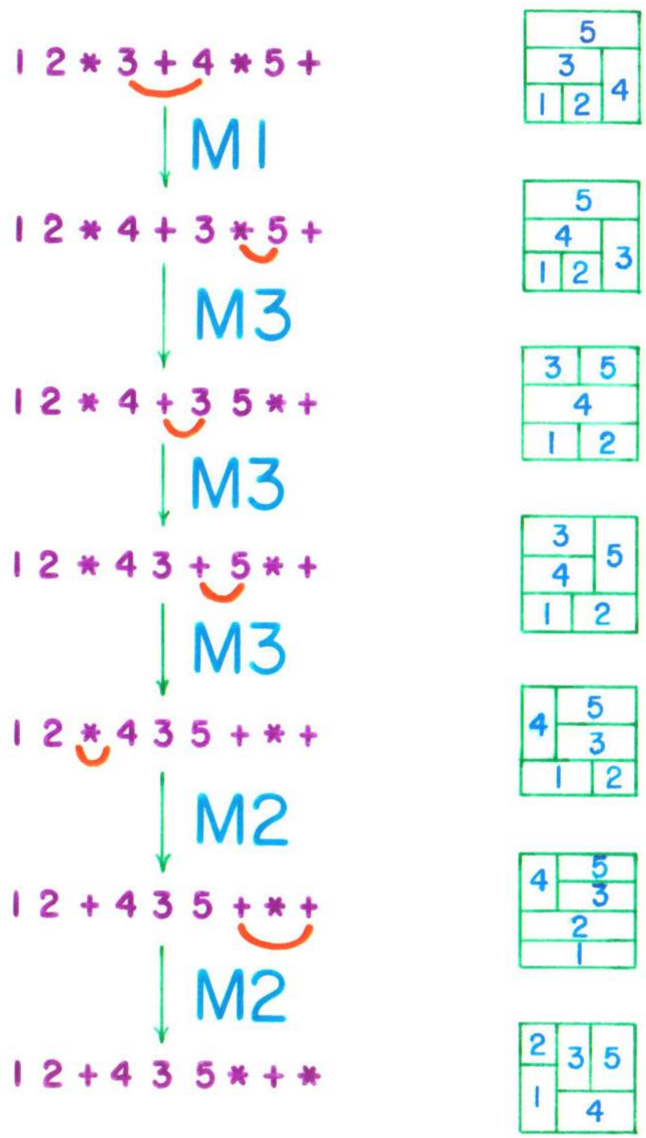


Slicing Tree

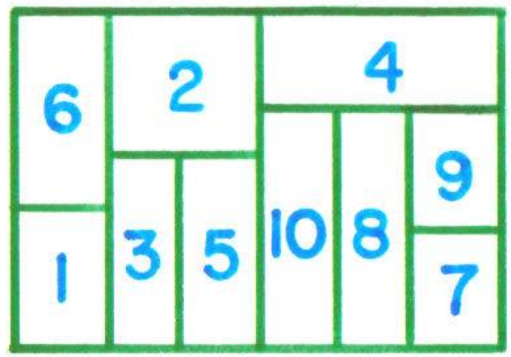
2 3 * 1 + 4 5 + 6 7 * + *

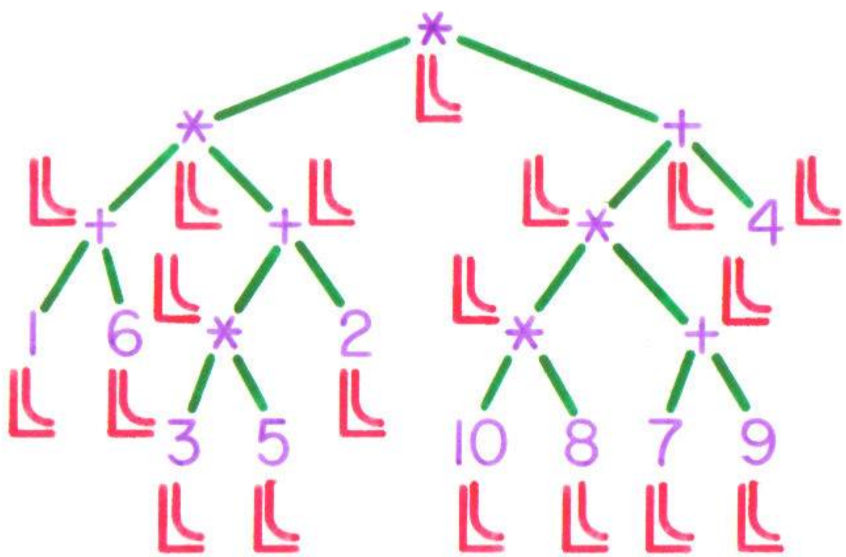
Polish Expression

Algorithm



Algorithm





How good are slicing floorplans?

Results for Soft Blocks

Experimental results => slicing is good for soft modules

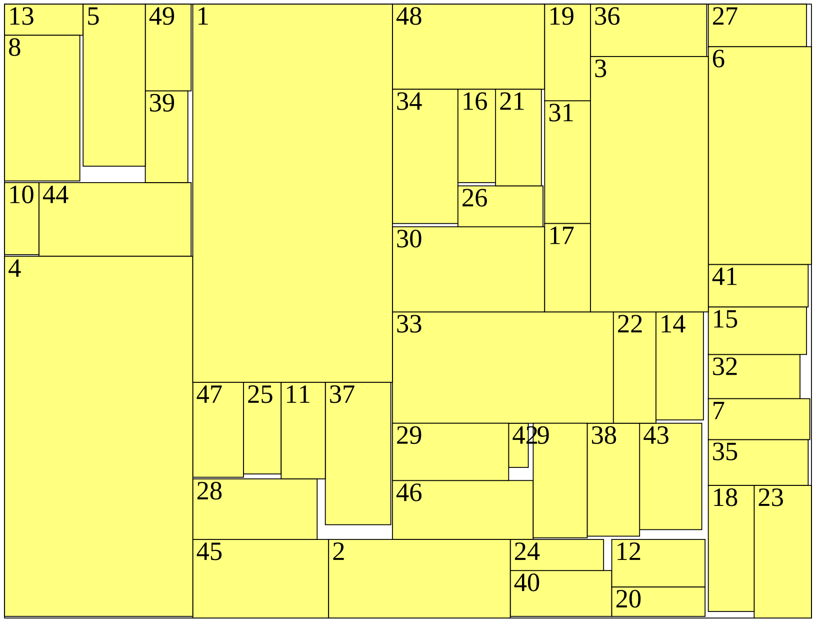
Circuit	*No. of Modules*	*runtime(s)*	*deadspace(%)*
apte	9	0.31	0.74
xerox	10	0.38	0
hp	11	0.45	0
ami33	33	3.22	0.01
ami49	49	6.93	0.13

*all modules have aspect ratio between 0.5 and 2

Results for Hard Blocks

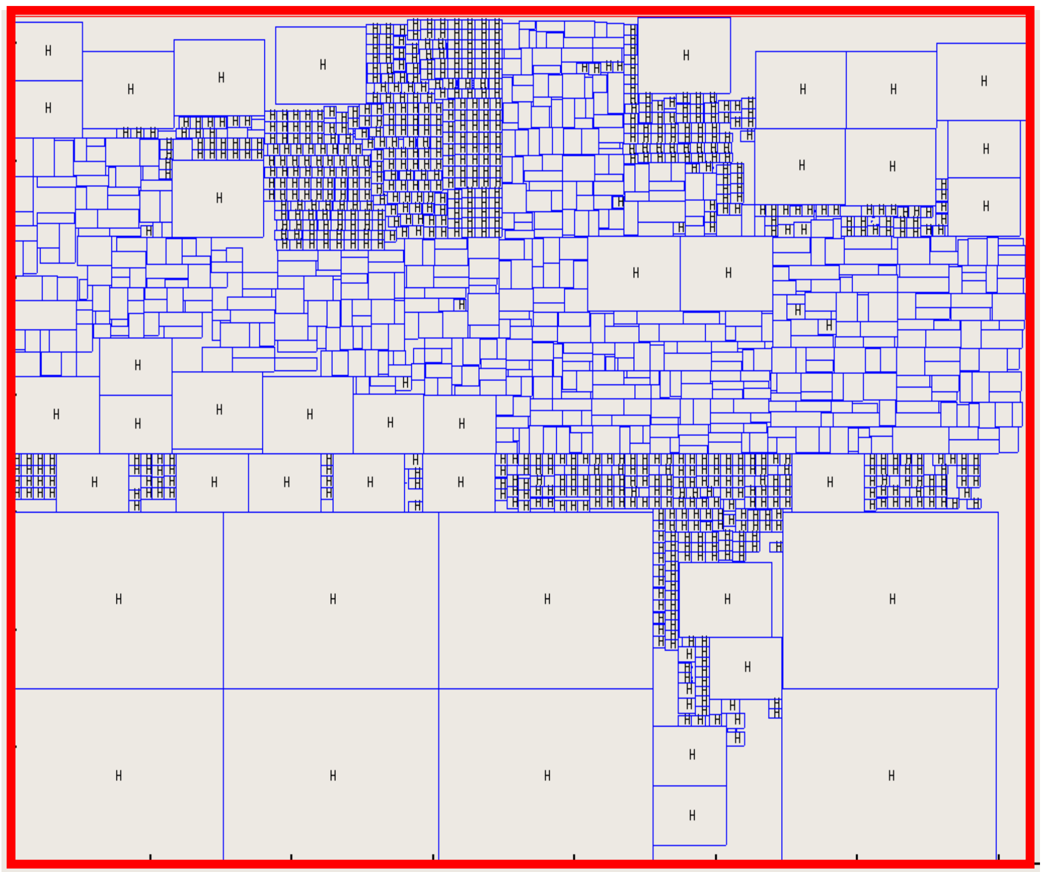
MCNC benchmark	Problem Size	Fast-SP Area	ECBL Area	Enhanced Q-seq Area	TBS Area	Enhanced O-tree Area	Slicing Area
apte	9	46.92	45.93	46.92	47.44	46.92	46.92
xerox	10	19.80	19.91	19.93	19.78	20.21	20.20
hp	11	8.947	8.918	9.03	8.48	9.16	9.03
ami33	33	1.205	1.192	1.194	1.196	1.242	1.183
ami49	49	36.5	36.70	36.75	36.89	37.73	36.24

- Excellent results by slicing for the largest MCNC benchmarks (Cheng, Deng, Wong, ASPDAC 2005)



Results on Large Benchmarks

- Yan and Chu, DAC-2008
- Slicing approach produced best results on GSRC & HB large benchmarks



Theoretical Analysis

Theorem [Young and Wong ISPD-97]

Given a set of soft blocks of total area A_{total} , maximum area A_{max} and shape flexibility $r \geq 2$, there exists a slicing floorplan F of these blocks such that:

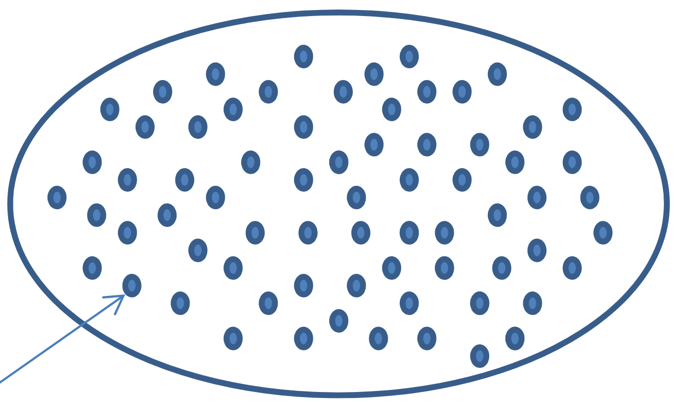
$$area(F) \leq \min\left\{(1+\frac{1}{\sqrt{r}}), \frac{5}{4}, (1+\alpha)\right\} A_{total}$$

where $$\alpha = \sqrt{\frac{2A_{max}}{rA_{total}}}$$

Can we do better?

Conjecture: For each non-slicing floorplan, there exists a slicing floorplan with "similar" area and topology.

Are slicing floorplans dense ?

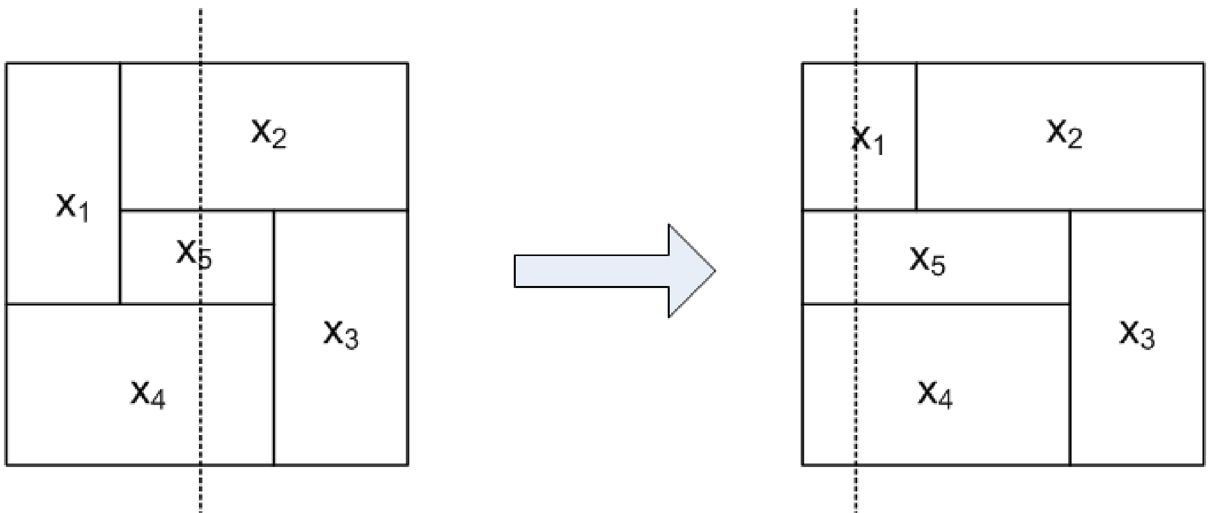


slicing floorplan

Wheel Floorplans with Squared Blocks

Lemma Given any wheel floorplan with 5 squared blocks, there is a "neighboring" slicing floorplan with equal/smaller area.

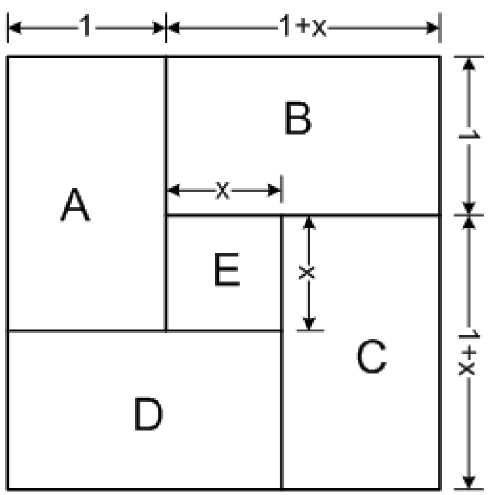
- It is not possible that x1 > x2 and x2 > x3 and x3 > x4 and x4 > x1. Otherwise, x1 > x1!
- We may assume x1 ≤ x2. It is easy to see that there is a "neighboring" slicing floorplan which is smaller!

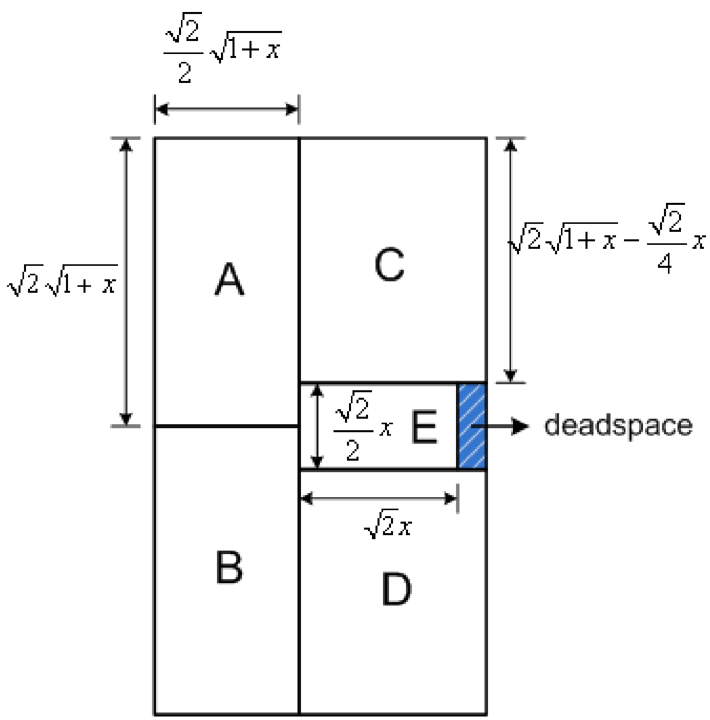


$$x_1+x_5+x_4 \leq x_2+x_5+x_4$$

Tightly Packed Wheel Floorplans

- Tightly packed wheel floorplans
  - 5 blocks: A, B, C and D identical, E is a square
  - 0< x ≤ 1; block aspect ratio ∈ [1/2, 2]
- Neighboring slicing floorplan
  - A, B and E: aspect ratio = 2

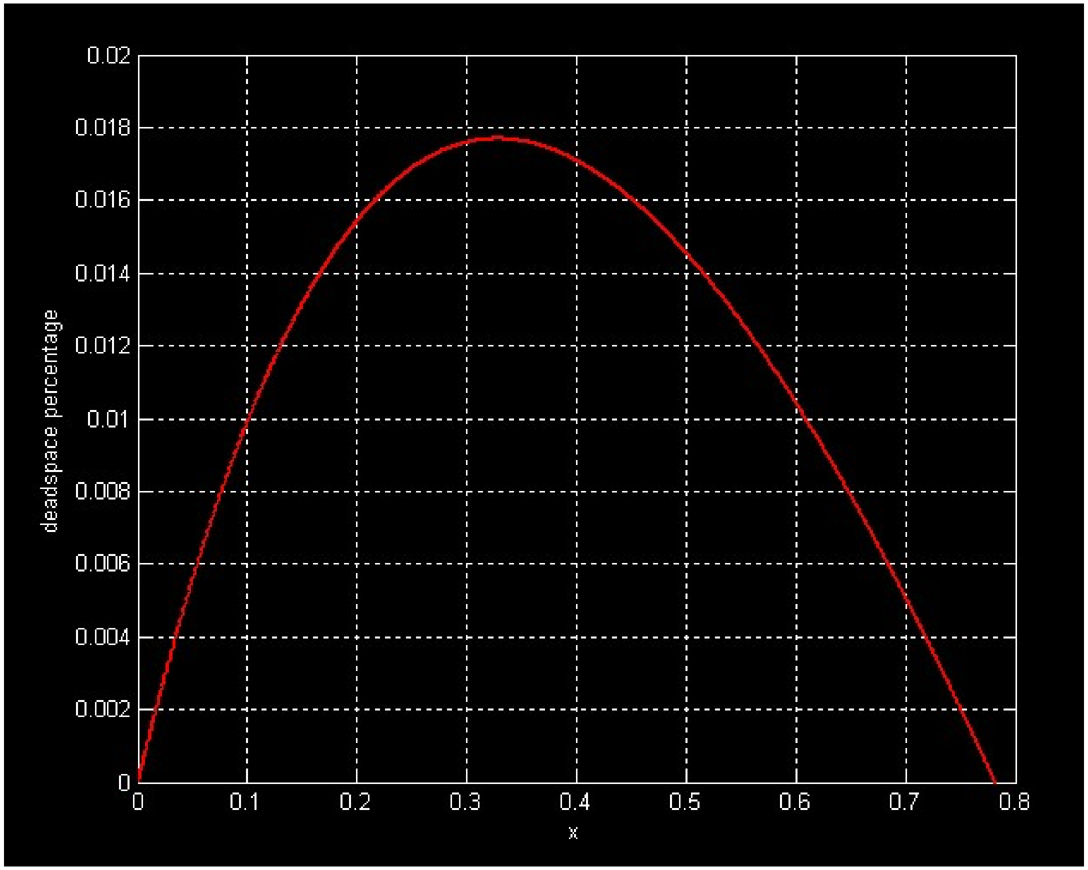




Tightly Packed Wheel Floorplans

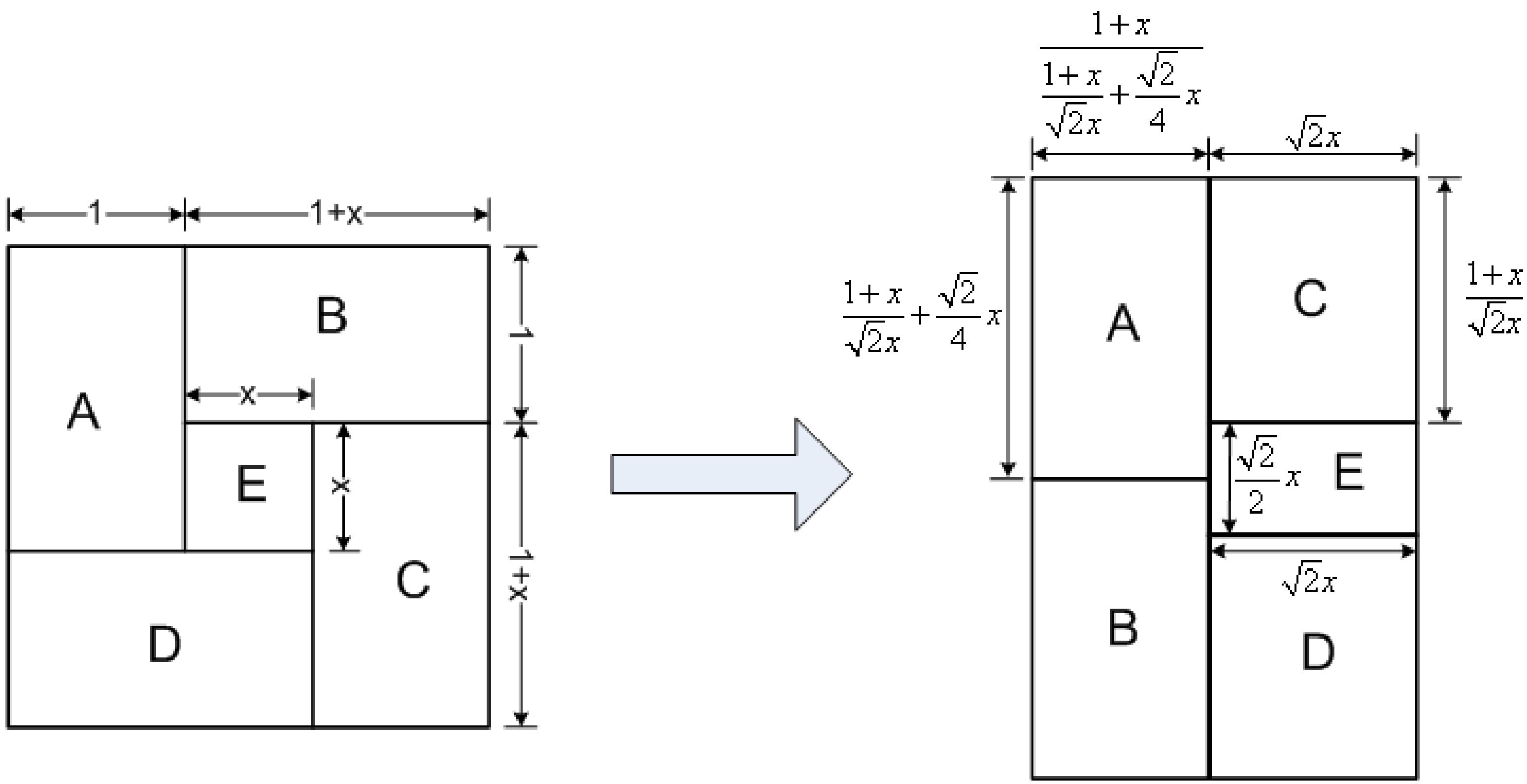
$$\textbf{Area Increase} = \frac{\left(\dfrac{1+x}{\sqrt{2}\sqrt{1+x}-\dfrac{\sqrt{2}}{4}x}-\sqrt{2}x\right)\dfrac{\sqrt{2}}{2}x}{4(1+x)+x^2}$$

- Area increase is 0% when x = 0.783
- Max area increase is 1.77% when x = 0.328



Tightly Packed Wheel Floorplans

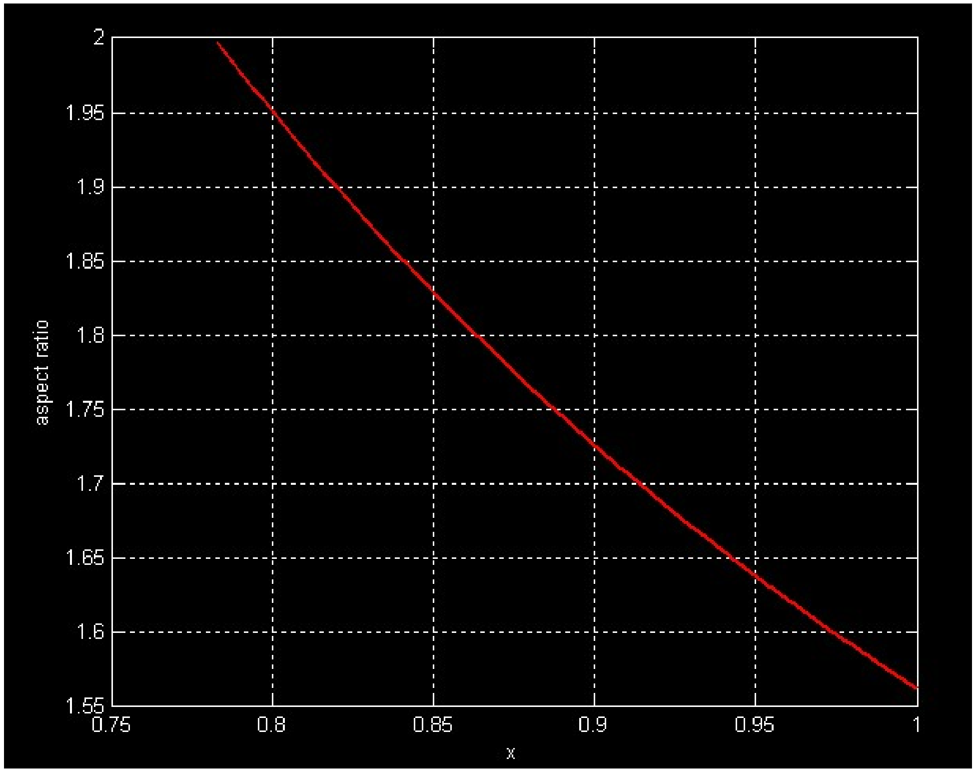
- When $0.783 \leq x \leq 1$
  - The slicing floorplan can be packed with zero dead-space
  - Adjust aspect ratios of A, B, C and D
  - Aspect ratio of E is 2



Tightly Packed Wheel Floorplans

R = aspect ratio of A & B = $\dfrac{\left(\dfrac{1+x}{\sqrt{2x}}+\dfrac{\sqrt{2}}{4}x\right)^2}{1+x}$

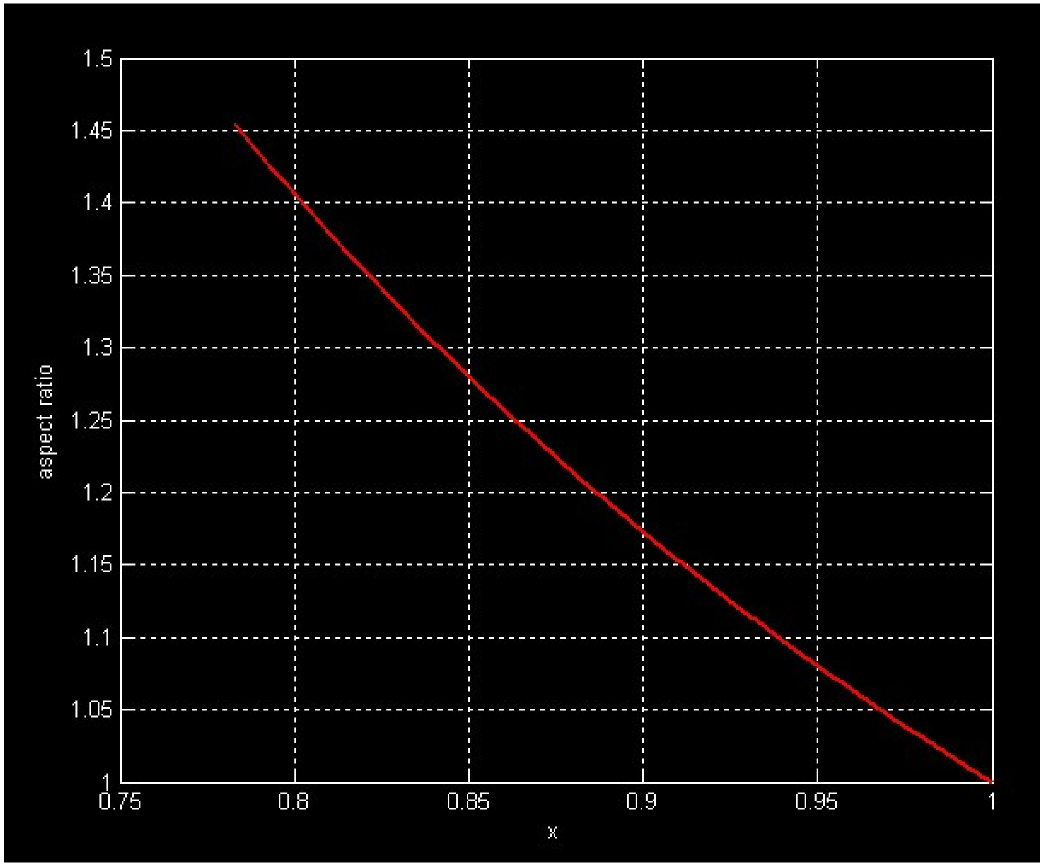
R ∈ [1.5625,2]   when x ∈ [0.783,1]



Tightly Packed Wheel Floorplans

S = aspect ratio of C &D = $\dfrac{1+x}{2x^2}$

S ∈ [1,1.455] when x ∈ [0.783, 1]



Conclusion

- Solution space partitioning: **S = S1 + S2 + … + Sn**
- New solution space **S'** = { **x1**, **x2**, **…**, **xn** } = { **S1** , **S2** , … , **Sn** }
- Encoding for { **S1** , **S2** , … , **Sn** }
- Significant reduction in solution space size
- Keep optimal solutions