# Optimal Slack-Driven Block Shaping Algorithm in Fixed-Outline Floorplanning

**Jackey Z. Yan**
Placement Tech. Group
Cadence Design Systems
San Jose, CA 95134, U.S.A.

**Chris Chu**
Department of ECE
Iowa State University
Ames, IA 50010, U.S.A.

# A Quotation

***Sometimes the questions are complicated and the answers are simple.***

**--- Dr. Seuss**
(1904 - 1991)

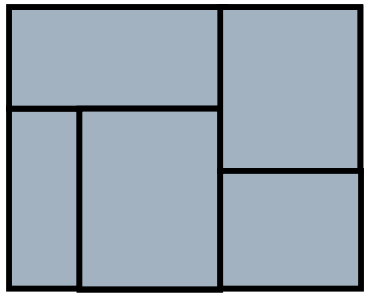
# Block Shaping in Fixed-Outline Floorplan

- **Input**
  - $n$ Blocks
    - Area $A_i$ for block $i$
    - Width bounds $W_i^{\min}$ and $W_i^{\max}$ for block $i$
    - Height bounds $H_i^{\min}$ and $H_i^{\max}$ for block $I$
  - Constraint graphs $G_h$ and $G_v$
  - Fixed-outline region

- **Output**
  - Block coordinates $(x_i, y_i)$, width $w_i$ and height $h_i$
    - All blocks inside fixed-outline region
    - All blocks without overlaps

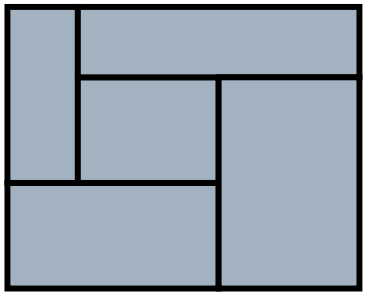
# Floorplan Representation

## Slicing Floorplan



- ☺ Excellent in shaping soft blocks (by shape curve)
- ☹ Can only represent slicing floorplans

## Non-Slicing Floorplan



- ☺ Can represent non-slicing floorplans (by sequence pair)
- ☺ Elegant representations, efficient manipulation
- ☹ Much more complicated in block shaping

# Previous Work

- T.C.Wang *et al.* **Optimal floorplan area optimization**. TCAD 1992
- P.Pan *et al.* **Area minimization for floorplans**. TCAD 1995
- S.Nakatake *et al.* **Module placement on BSG-structure and IC layout applications**. ICCAD 1996
- T.S.Moh *et al.* **Globally optimal floorplanning for a layout problem**. TCSI 1996
- M.Kang *et al.* **General floorplanning with L-shaped, T-shaped and soft blocks based on bounded slicing grid structure**. ASP-DAC 1997
- H.Murata *et al.* **Sequence-pair based placement method for hard/soft/pre-placed modules**. ISPD 1998
- F.Y.Young *et al.* **Handling soft modules in general non-slicing floorplan using Lagrangian relaxation**. TCAD 2001
- S.N.Adya *et al.* **Fixed-outline floorplanning: Enabling hierarchical design**. TVLSI 2003
- C.Lin *et al.* **A revisit to floorplan optimization by Lagrangian relaxation**. ICCAD 2006

# *SDS* Overview

- Specifically formulated for fixed-outline floorplanning
- Optimal, efficient and scalable for non-slicing floorplan
- Main contributions
  - **Basic Slack-Driven Shaping**
  - **Three Optimality Conditions**
  - **Slack-Driven Shaping (*SDS*)**
- ***Promising*** Experimental Results
  - Obtain optimal solutions for both MCNC & HB benchmarks simply by the basic *SDS.*
  - For MCNC benchmarks, 253x faster than Young's, 33x faster than Lin's, to produce results of similar quality.

# Problem Formulation

**Minimize the layout height with a fixed layout width upper bound**

- $W$
- $G_h, G_v$ (two dummy vertices 0 & $n+1$)

Minimize

$$y_{n+1}$$

subject to

$$x_{n+1} \leq W$$

$$x_j \geq x_i + w_i, \qquad \forall (i,j) \in G_h$$

$$y_j \geq y_i + h_i, \qquad \forall (i,j) \in G_v$$

$$W_i^{min} \leq w_i \leq W_i^{max}, \qquad 1 \leq i \leq n$$

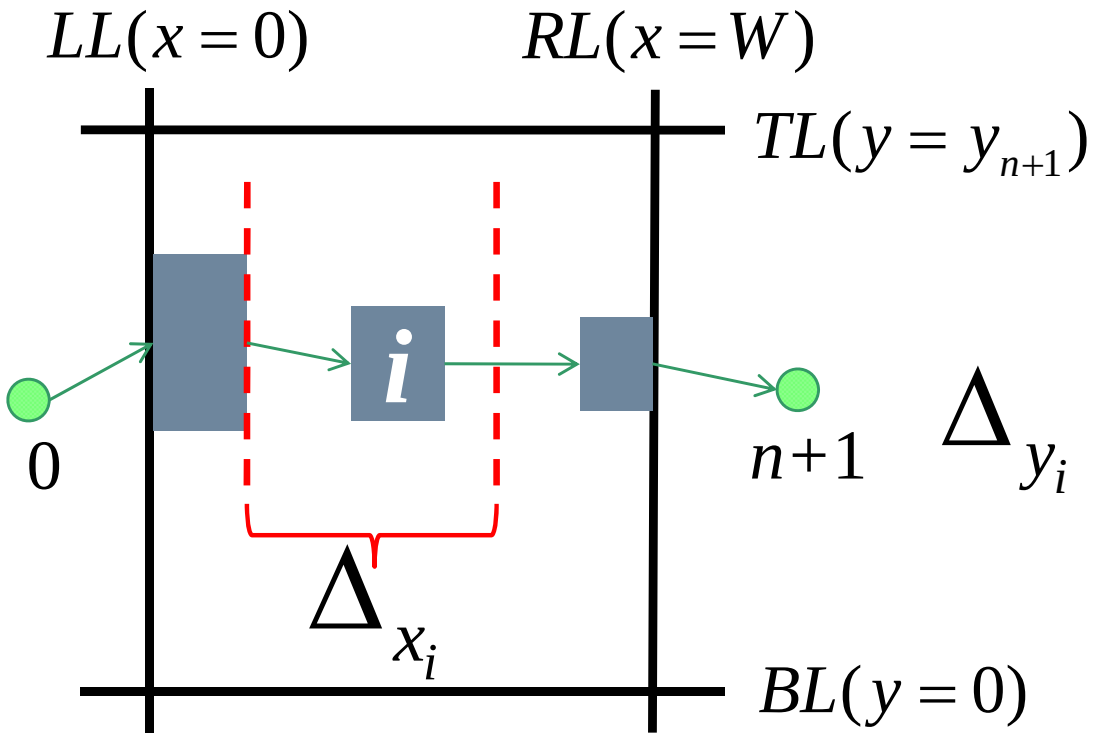
$$H_i^{min} \leq h_i \leq H_i^{max}, \qquad 1 \leq i \leq n$$

$$w_i h_i = A_i, \qquad 1 \leq i \leq n$$

$$x_0 = 0$$

$$y_0 = 0$$

# Notion of *Slack* in Floorplanning




- $G_h, G_v$
- Shape of $n$ blocks

horizontal slack

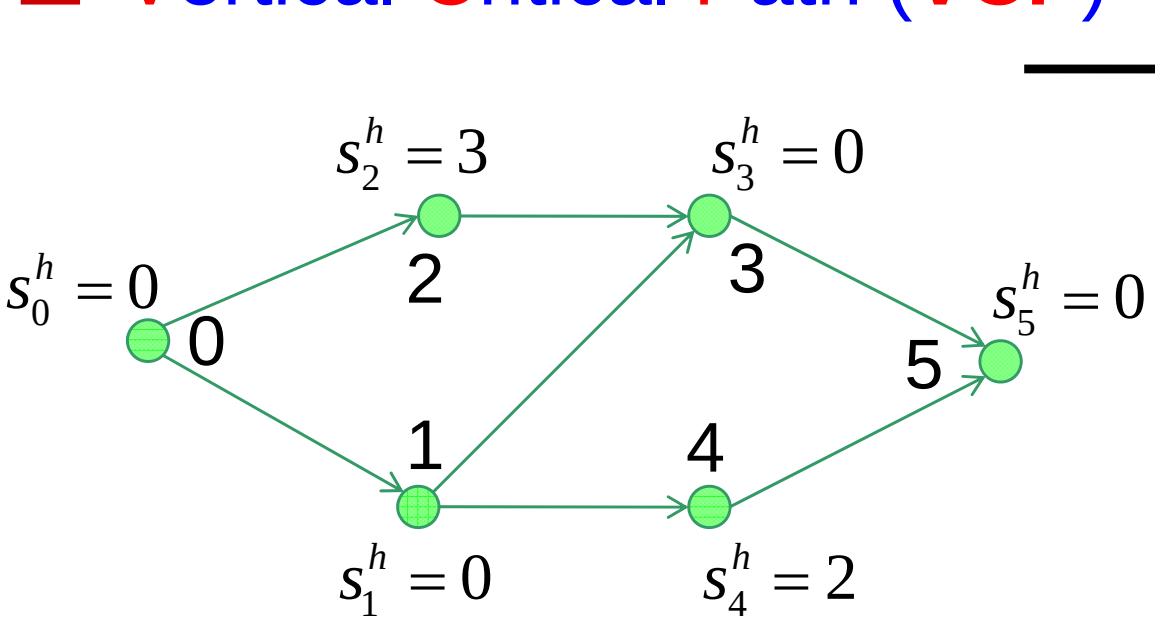
$$s_i^h = \max(0, \Delta_{x_i})$$

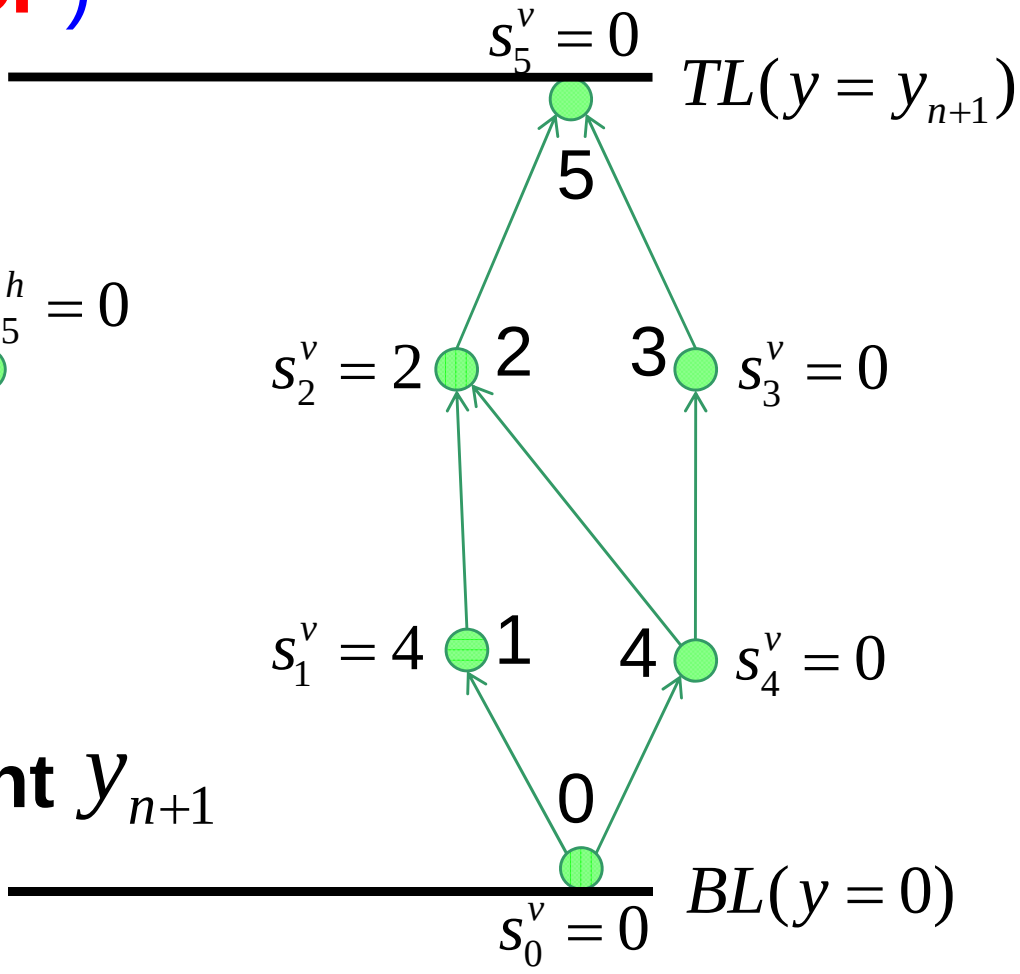
vertical slack

$$s_i^v = \max(0, \Delta_{y_i})$$

# Horizontal/Vertical Critical Path

- Horizontal Critical Path (**HCP**)
- Vertical Critical Path (**VCP**)




**Length of VCP = Layout height $y_{n+1}$**

# Basic Slack-Driven Shaping

- Soft blocks are shaped *iteratively*.
- At each iteration, apply two operations:


  - Globally distribute the total amount of slack to the individual soft block.
- Algorithm stops when there is no identified soft block to shape.
- Layout height is monotonically reducing, and layout width is bouncing, but always within the upper bound.

# Target Soft Blocks

$$\mathrm{I}:\{i \text{ is hard}\}$$

$$\mathrm{II}:\{i \text{ is soft}\}\bigcap\{s_i^h \neq 0, s_i^v \neq 0\}$$

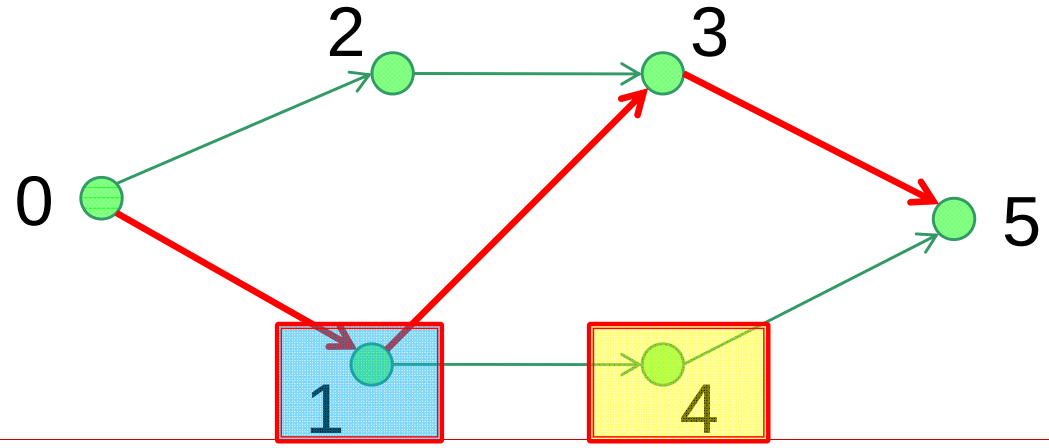
$$\mathrm{III}:\{i \text{ is soft}\}\bigcap\{s_i^h = 0, s_i^v = 0\}$$

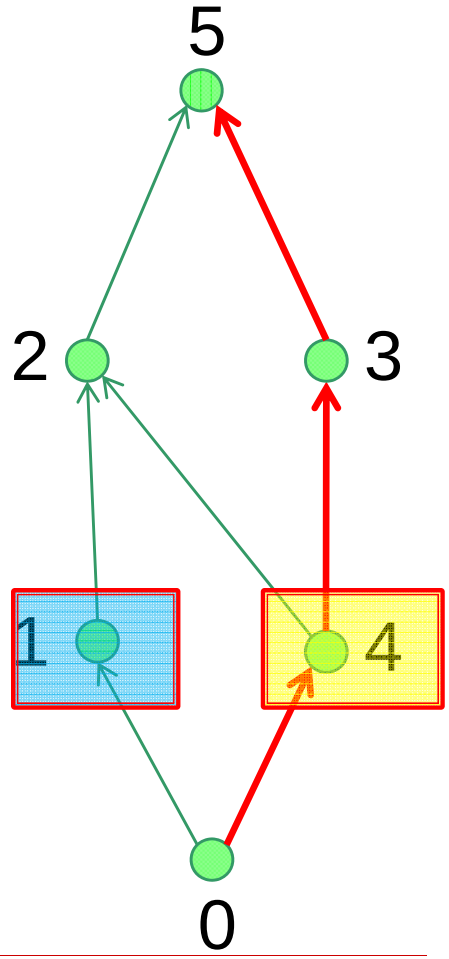
$$\mathrm{IV}:\{i \text{ is soft}\}\bigcap\{s_i^h \neq 0, s_i^v = 0\}\bigcap\{w_i \neq W_i^{\max}\}$$

$$\mathrm{V}:\{i \text{ is soft}\}\bigcap\{s_i^h \neq 0, s_i^v = 0\}\bigcap\{w_i = W_i^{\max}\}$$

$$\mathrm{VI}:\{i \text{ is soft}\}\bigcap\{s_i^h = 0, s_i^v \neq 0\}\bigcap\{h_i \neq H_i^{\max}\}$$

$$\mathrm{VII}:\{i \text{ is soft}\}\bigcap\{s_i^h = 0, s_i^v \neq 0\}\bigcap\{h_i = H_i^{\max}\}$$

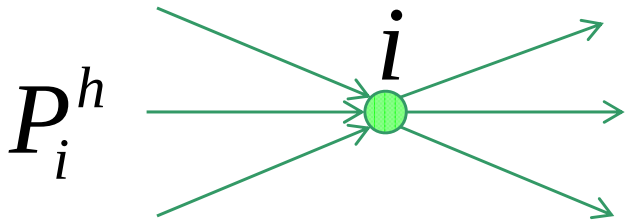


# Shaping Scheme

IV $i$ $w_i' \bullet \delta_i^h = w_i' - w_i$

VI $i$ $h_i' \bullet \delta_i^v = h_i' - h_i$




Basic *SDS*

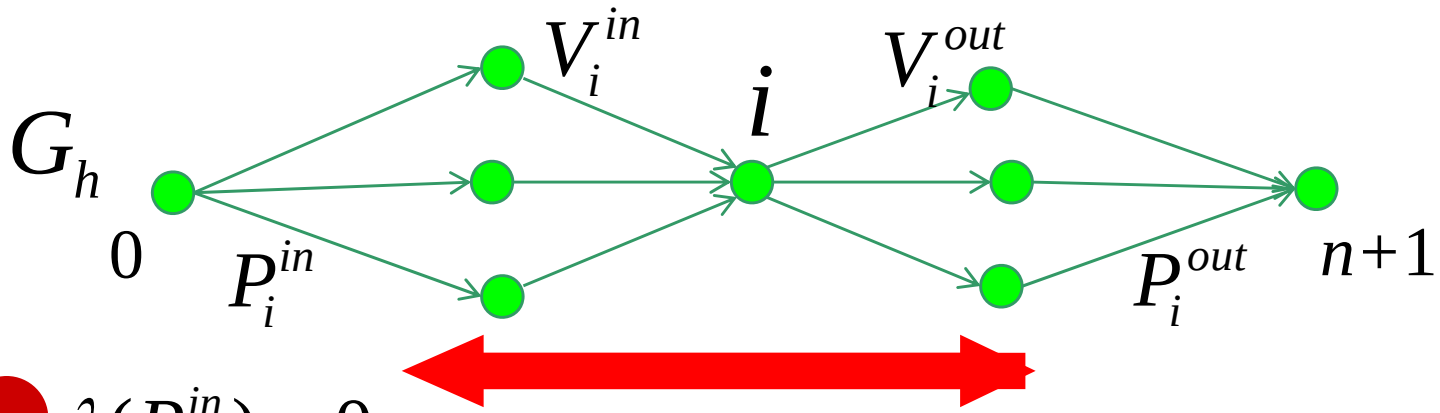
WHILE (there is target soft block)

- $\forall i \in \mathrm{IV},\ \delta_i^h = \alpha_i^h s_i^h \ \text{s.t.} \sum_{i \in p} \alpha_i^h \le 1, \forall p \in G_h$
- $\forall i \in \mathrm{VI},\ \delta_i^v = \alpha_i^v s_i^v \ \text{s.t.} \sum_{i \in p} \alpha_i^v \le 1, \forall p \in G_v$

$$\alpha_i^h = \frac{(W_i^{\max} - w_i)}{\mathrm{MAX}_{p \in P_i^h}(\sum_{k \in p}(W_k^{\max} - w_k))}$$

# Dynamic Programming Approach

$$\underset{p \in P_i^h}{\mathrm{MAX}}\left(\sum_{k \in p}\left(W_k^{\max} - w_k\right)\right) = \lambda(P_i^h)$$




- $\lambda(P_0^{in}) = 0$
- $\lambda(P_{n+1}^{out}) = 0$
- $\lambda(P_i^{in}) = \underset{j \in V_i^{in}}{\mathrm{MAX}}(\lambda(P_j^{in})) + (W_i^{\max} - w_i)$
- $\lambda(P_i^{out}) = \underset{j \in V_i^{out}}{\mathrm{MAX}}(\lambda(P_j^{out})) + (W_i^{\max} - w_i)$
- $\lambda(P_i^h) = \lambda(P_i^{in}) + \lambda(P_i^{out}) - (W_i^{\max} - w_i)$

**Linear Time!**

**Topological sorting**

↓

**Scan (source to sink)**

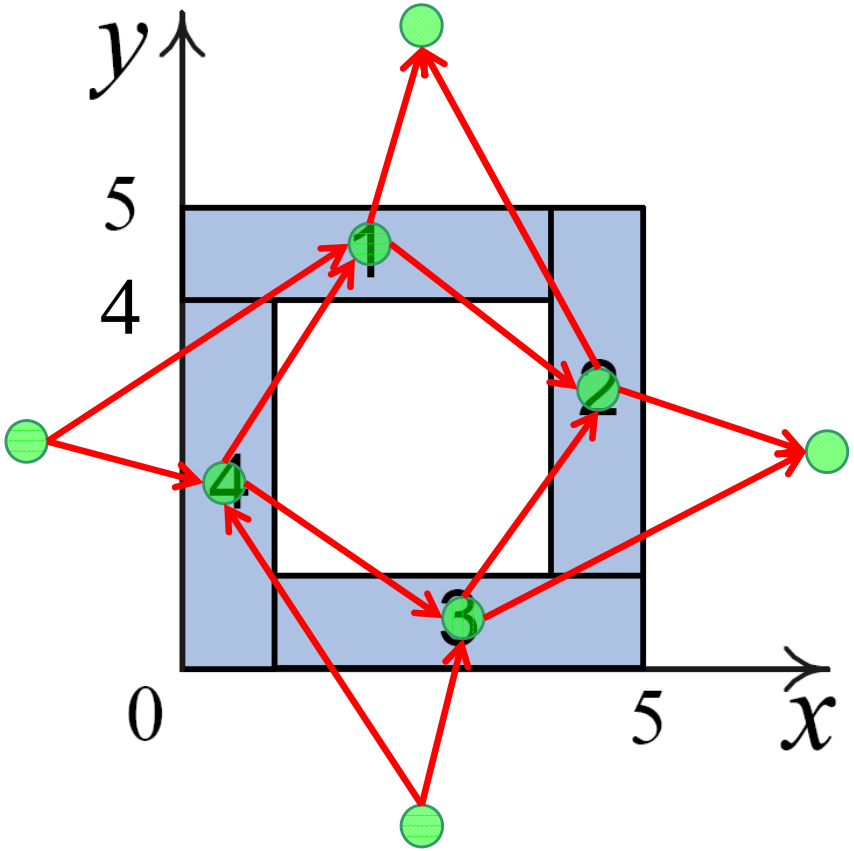
↓

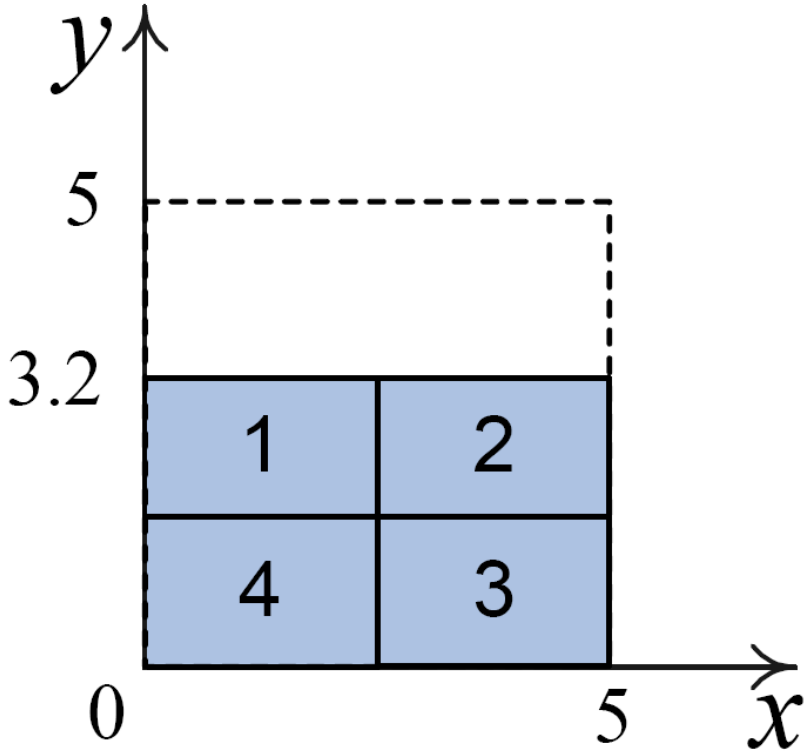
**Scan (sink to source)**

↓

**Traverse each block**

# A Non-Optimal Case



# Optimal Conditions

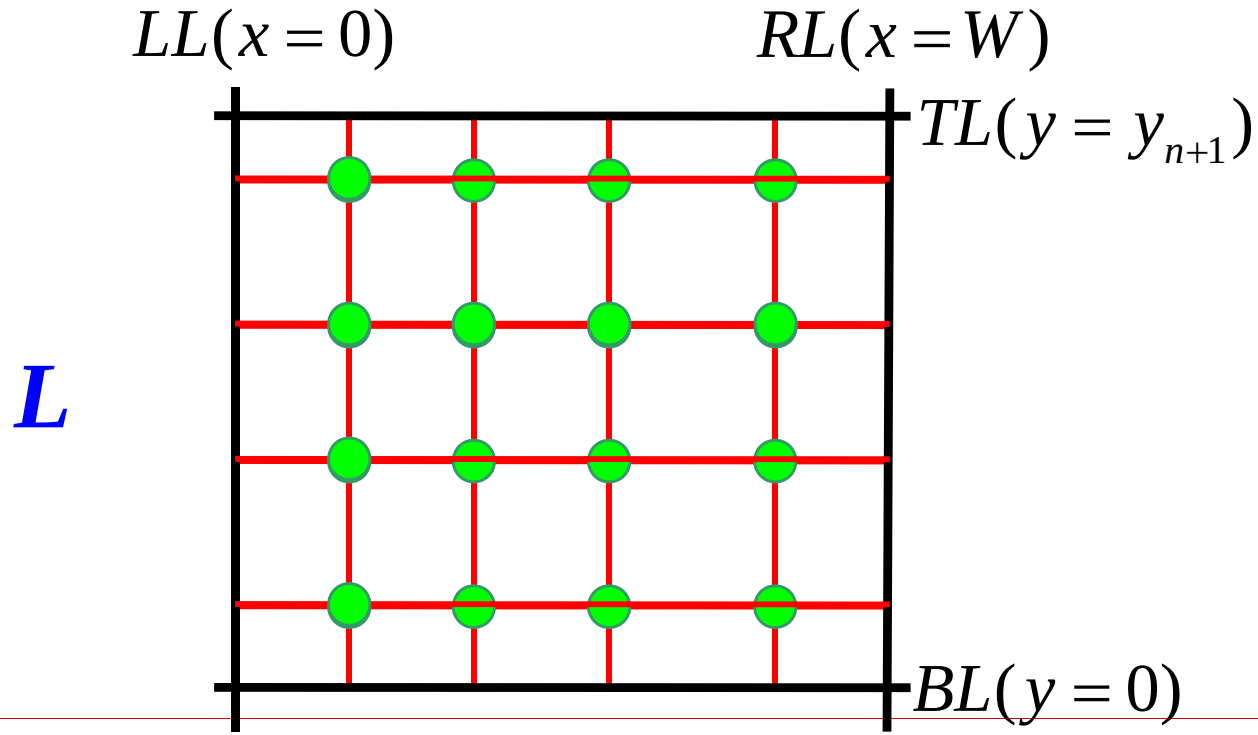
**$L$: a shaping solution generated by the basic *SDS*.**

***Hard critical path*: all blocks on this critical path are *hard* blocks.**

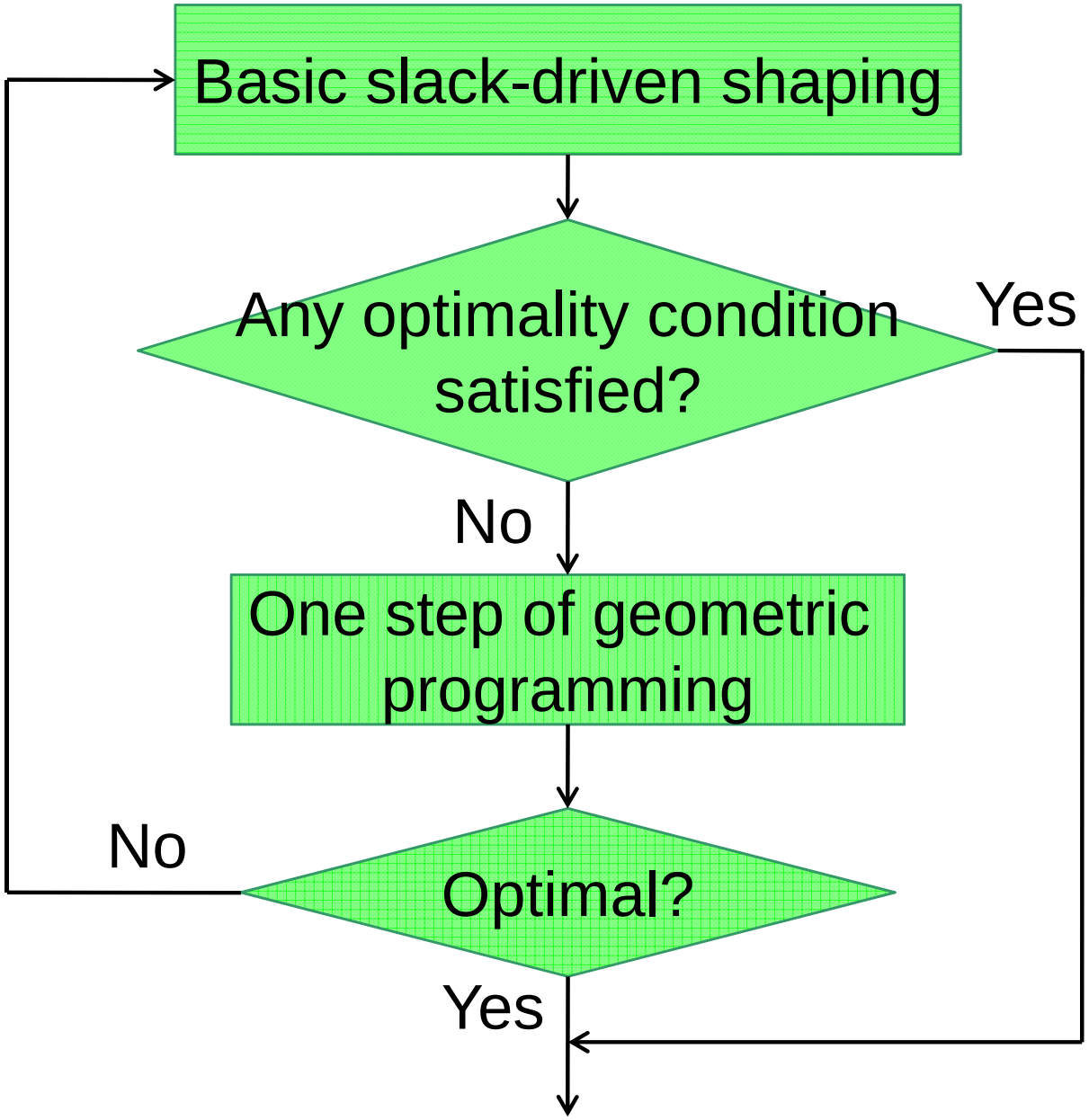
1. If there exists one hard VCP in $L$, then $L$ is optimal.

2. If there exists at most one non-hard HCP in $L$, then $L$ is optimal.

3. If there exists at most one non-hard VCP in $L$, then $L$ is optimal.



# Flow of Slack-Driven Shaping



# Experimental Results

- All experiments are run on Linux server with AMD Opteron 2.59 GHz CPU and 16 GB RAM.
- Two sets of floorplan benchmarks
  - MCNC: **9~49** soft blocks
  - HB: **500~2000** mixed of hard and soft blocks
- Aspect ratio bound of soft block is [1/3, 3].
- Input constraint graphs are provided by a floorplanner.
- In all experiments, *SDS* achieves the optimal solutions simply by the basic *SDS.*

# Experiments on MCNC Benchmarks

- Compare *SDS* with Young's and Lin's algorithms
  - Young's is minimizing the layout area.
  - Lin's is minimizing the layout half parameter.
  - *SDS* is minimizing the layout height with width upper bound.
- Procedure of Experiments
  - Conduct two groups of experiments: 1) *SDS* v.s.Young's; 2) *SDS* v.s. Lin's.
  - In each group, run each shaping algorithm 1000 times.
  - Run Young's and Lin's first, then use their resulting width as the input upper-bound width of *SDS.*
  - Compare the final result based on Young's and Lin's objective.

# Compared with Young's on MCNC

(* total shaping time of 1000 times and does not include I/O time)

| Circuit. | #. Soft Blocks | Young's [1] | | *SDS* | | *SDS* stops earlier | |
|---|---|---|---|---|---|---|---|
| | | ws (%) | Shaping Time*(s) | ws (%) | Shaping Time*(s) | ws (%) | Shaping Time*(s) |
| apte | **9** | **4.66** | **0.12** | **0.00** | **0.26** | **2.85** | **0.01** |
| xerox | **10** | **7.69** | **0.08** | **0.01** | **0.23** | **6.46** | **0.01** |
| hp | **11** | **10.94** | **0.08** | **1.70** | **0.10** | **7.96** | **0.02** |
| ami33a | **33** | **8.70** | **22.13** | **0.44** | **3.97** | **8.67** | **0.28** |
| ami49a | **49** | **10.42** | **203.80** | **1.11** | **1.86** | **9.74** | **0.20** |
| **Norm** | | **393.92** | **23.351** | **1.00** | **1.00** | **313.98** | **0.092** |

[1] F.Y.Young, C.C.N.Chu, W.S.Luk and Y.C.Wong, ***Handling soft modules in general non-slicing floorplan using Lagrangian relaxation***. TCAD 2001

# Compared with Lin's on MCNC

(* total shaping time of 1000 times and does not include I/O time)

| Circuit. | #. Soft Blocks | Lin's [1] | | SDS | | SDS stops earlier | |
|---|---|---|---|---|---|---|---|
| | | Half Para. | Shaping Time*(s) | Half Para. | Shaping Time*(s) | Half Para. | Shaping Time*(s) |
| apte | 9 | 439.319 | 0.99 | 439.305 | 0.59 | 439.179 | 0.01 |
| xerox | 10 | 278.502 | 1.24 | 278.320 | 0.30 | 278.488 | 0.12 |
| hp | 11 | 190.385 | 1.51 | 190.244 | 0.17 | 190.383 | 0.10 |
| ami33a | 33 | 215.965 | 34.85 | 215.711 | 1.45 | 215.958 | 0.46 |
| ami49a | 49 | 377.857 | 26.75 | 377.525 | 2.20 | 377.824 | 0.44 |
| Norm | | 1.001 | 10.177 | 1.000 | 1.00 | 1.001 | 0.304 |

[1] C.Lin, H.Zhou and C.Chu, ***A revisit to floorplan optimization by Lagrangian relaxation***. ICCAD 2006

# Comparison on Runtime Complexity

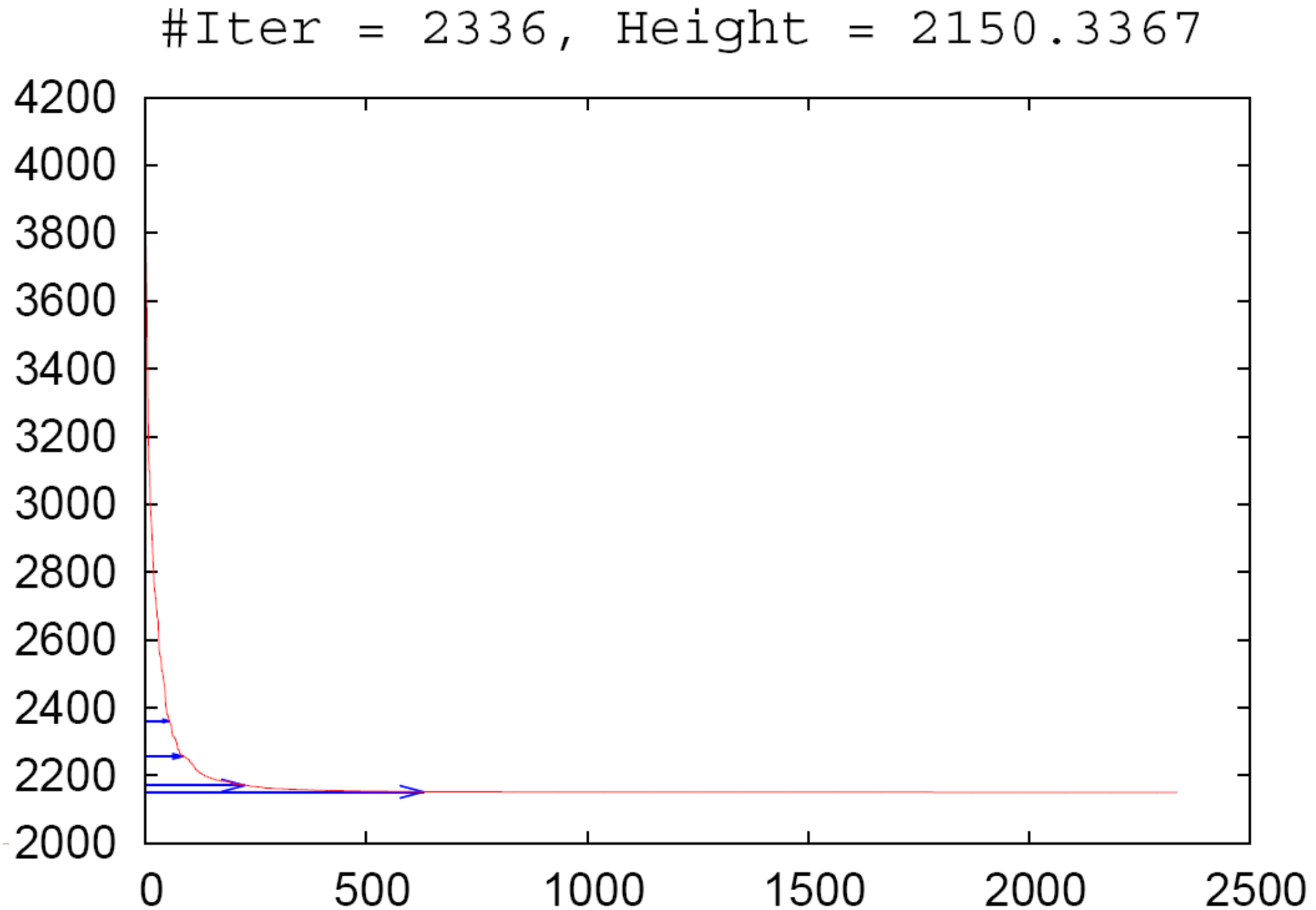
| Algorithm | Runtime Complexity |
|---|---|
| Young *et al.* | $\mathcal{O}(m^3 + km^2)$ |
| Lin *et al.* | $\mathcal{O}(kn^2 m log(nC))$ |
| Basic *SDS* | $\mathcal{O}(km)$ |

($k$ is the total number of iterations, $n$ is the total number of blocks in the design, $m$ is the total number of edges in $G_h$ and $G_v$, and $C$ is the biggest input cost.)

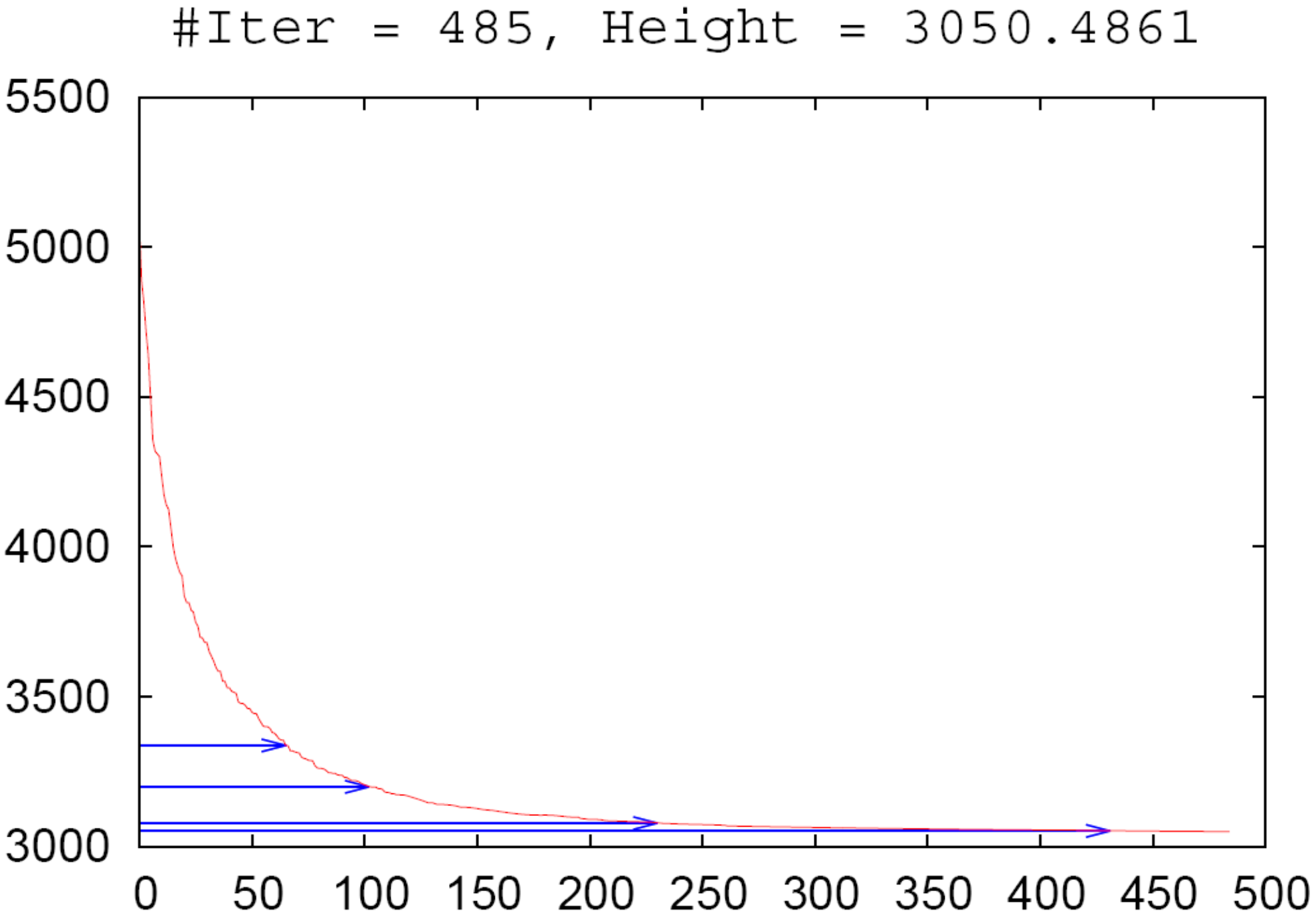
# Experiments on HB Benchmarks

- Large-scale floorplan designs (500~2000 blocks)
- Young's and Lin's algorithms cannot handle HB benchmarks
- Highlights on *SDS*'s results
  - Average convergence time: 1.18 second
  - Average total number of iterations: 1901
  - Average after 5.9%, 9.6%, 22.3% and 47.3%, layout height is within 10%, 5%, 1% and 0.1% difference from the optimal solution.

# Convergence Graph (1) [665 soft blks]



# Convergence Graph (2) [1200 soft blks]



# Conclusions

- *SDS* – efficient, scalable and optimal slack-driven shaping algorithm in fixed-outline floorplanning.
  - **Basic Slack-Driven Shaping**
  - **Optimality Conditions**
  - **Slack-Driven Shaping (*SDS*)**
- ***Promising*** Experimental Results
  - Obtain optimal solutions for both MCNC & HB benchmarks simply by the basic *SDS*.
  - For MCNC benchmarks, 253x faster than Young's, 33x faster than Lin's, to produce results of similar quality.

# Future Work

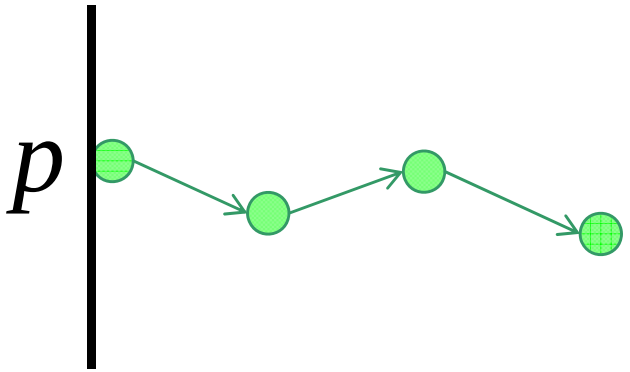
- Embed *SDS* into a floorplanner.
- Use the duality gap as a better stopping criterion.
- Propose a more scalable algorithm to replace the geometric programming method.
- Extend *SDS* to handle non-fixed outline floorplanning.
- Applied on buffer/wire sizing for timing optimization.

# Thank you!

# Shaping Scheme

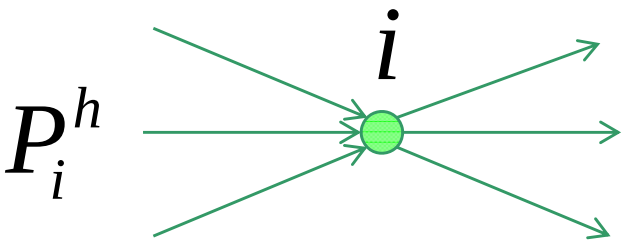
- $\delta_i^h$ : increase on $w_i$ for $i \in$ subset IV
- $i \in p$

$LL(x=0)$ $RL(x=W)$




- $s_{\max}^p = \mathrm{MAX}_{i \in p}(s_i^h)$
- $\sum_{i \in p} \delta_i^h \le s_{\max}^p$

$$\delta_i^h \le \alpha_i^p s_i^h \qquad (\sum_{i \in p} \alpha_i^p = 1, \alpha_i^p \ge 0)$$




$$\delta_i^h \le \alpha_i^p s_i^h, \forall p \in P_i^h$$

$$\alpha_i^p = \begin{cases} 0 \\ \dfrac{W_i^{\max} - w_i}{\sum_{k \in p}(W_k^{\max} - w_k)} \end{cases}$$

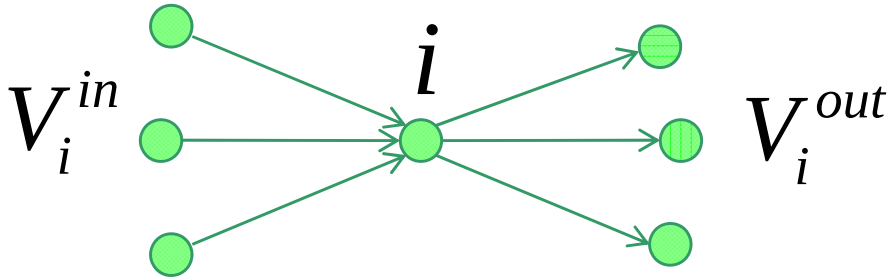
$$\delta_i^h \le \frac{(W_i^{\max} - w_i)s_i^h}{\mathrm{MAX}_{p \in P_i^h}(\sum_{k \in p}(W_k^{\max} - w_k))}$$

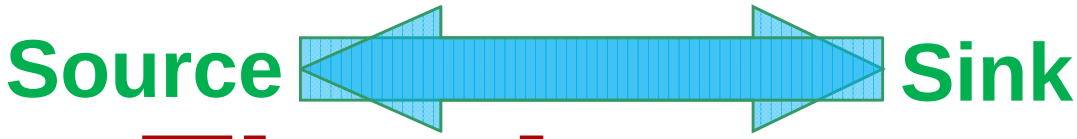
# Dynamic Programming Approach

$$\underset{p \in P_i^h}{\mathrm{MAX}}\left(\sum_{k \in p}\left(W_k^{\max} - w_k\right)\right) \;=\; \lambda(P_i^h)$$

$P_i^{in}$ : paths from source to $i$

$P_i^{out}$ : paths from $i$ to sink




**Linear Time!**

$$\begin{cases}
\lambda(P_0^{in}) = 0 \\
\lambda(P_{n+1}^{out}) = 0 \\
\lambda(P_i^{in}) = \underset{j \in V_i^{in}}{\mathrm{MAX}}(\lambda(P_j^{in})) + (W_i^{\max} - w_i) \\
\lambda(P_i^{out}) = \underset{j \in V_i^{out}}{\mathrm{MAX}}(\lambda(P_j^{out})) + (W_i^{\max} - w_i) \\
\lambda(P_i^h) = \lambda(P_i^{in}) + \lambda(P_i^{out}) - (W_i^{\max} - w_i)
\end{cases}$$

# Optimal Conditions

***L*: a shaping solution generated by the basic SDS.**

In $L$, the only remaining soft blocks that can be shaped to possibly improve $L$ are the ones at the intersection of HCP&VCP.

***Hard critical path*: all blocks on this critical path are *hard* blocks.**

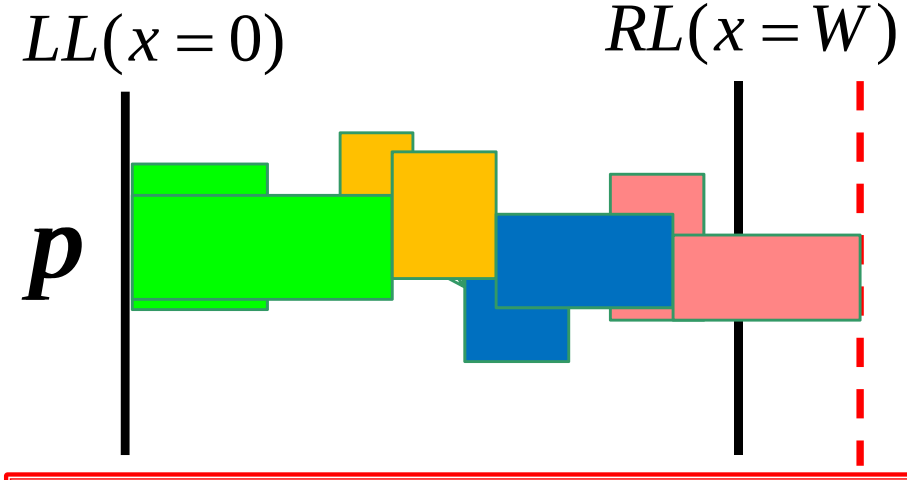
LEMMA 2. If there exists one hard VCP in $L$, then $L$ is optimal.

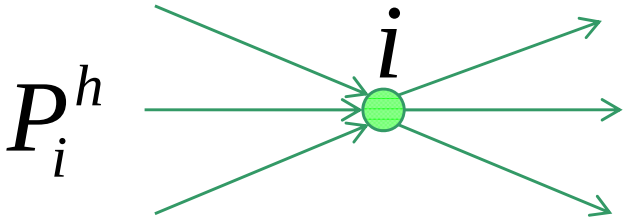
LEMMA 3. If there exists at most one non-hard HCP or at most one non-hard VCP in $L$, then $L$ is optimal.

# Shaping Scheme for blocks in IV

$i$ $w_i'$

- $\delta_i^h = w_i' - w_i$
- $i \in p$
- $s_{\max}^p = \mathrm{MAX}_{i \in p}(s_i^h)$
- ~~$\sum_{i \in p} \delta_i^h > s_{\max}^p$~~
- $\sum_{i \in p} \delta_i^h \le s_{\max}^p$




$$\delta_i^h \le \alpha_i^p s_i^h \ (\sum_{i \in p} \alpha_i^p = 1, \alpha_i^p \ge 0)$$

$$\delta_i^h \le \alpha_i^p s_i^h, \forall p \in P_i^h$$

$$\alpha_i^p = \frac{W_i^{\max} - w_i}{\sum_{k \in p}(W_k^{\max} - w_k)}$$

$$\delta_i^h \le \frac{(W_i^{\max} - w_i)s_i^h}{\mathrm{MAX}_{p \in P_i^h}(\sum_{k \in p}(W_k^{\max} - w_k))}$$