UCLA COMPUTER SCIENCE DEPARTMENT

An Analytical Placer for Mixed-Size 3D Placement

Jason Cong[1,2] and Guojie Luo[1]

[1]University of California, Los Angeles

[2]California NanoSystems Institute

{ cong, gluo } @ cs.ucla.edu

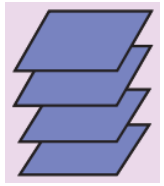
This work is partially supported by
NSF CCF-0430077 and CCF-0528583

Outline

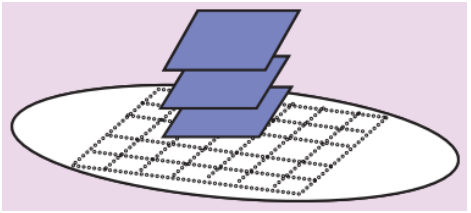
- Motivation of mixed-size 3D placement
- Review of analytical 3D placement
- Contributions
  - Analyze the limitations of analytical 3D placement
  - Develop a 3D placement flow for mixed-size circuits
  - Use the multiple-stepsize scheme to enhance analytical solvers
- Experimental results

Availability of 3D Integration Technologies

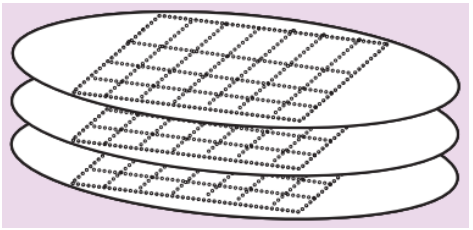
Stacking



Die-to-die

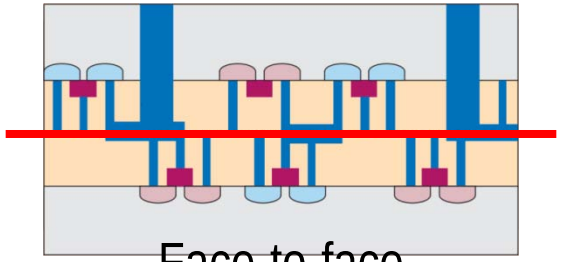


Die-to-wafer

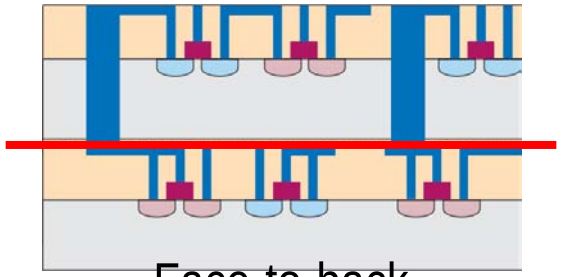


Wafer-to-wafer

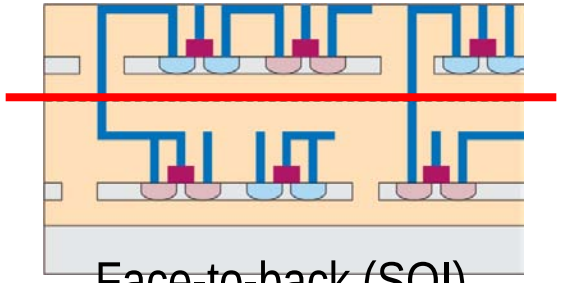
Bonding



Face-to-face

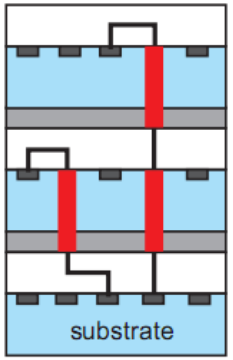


Face-to-back

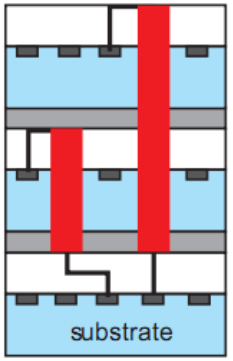


Face-to-back (SOI)

TSV



Via-first



Via-last

Image source: IBM J.'08

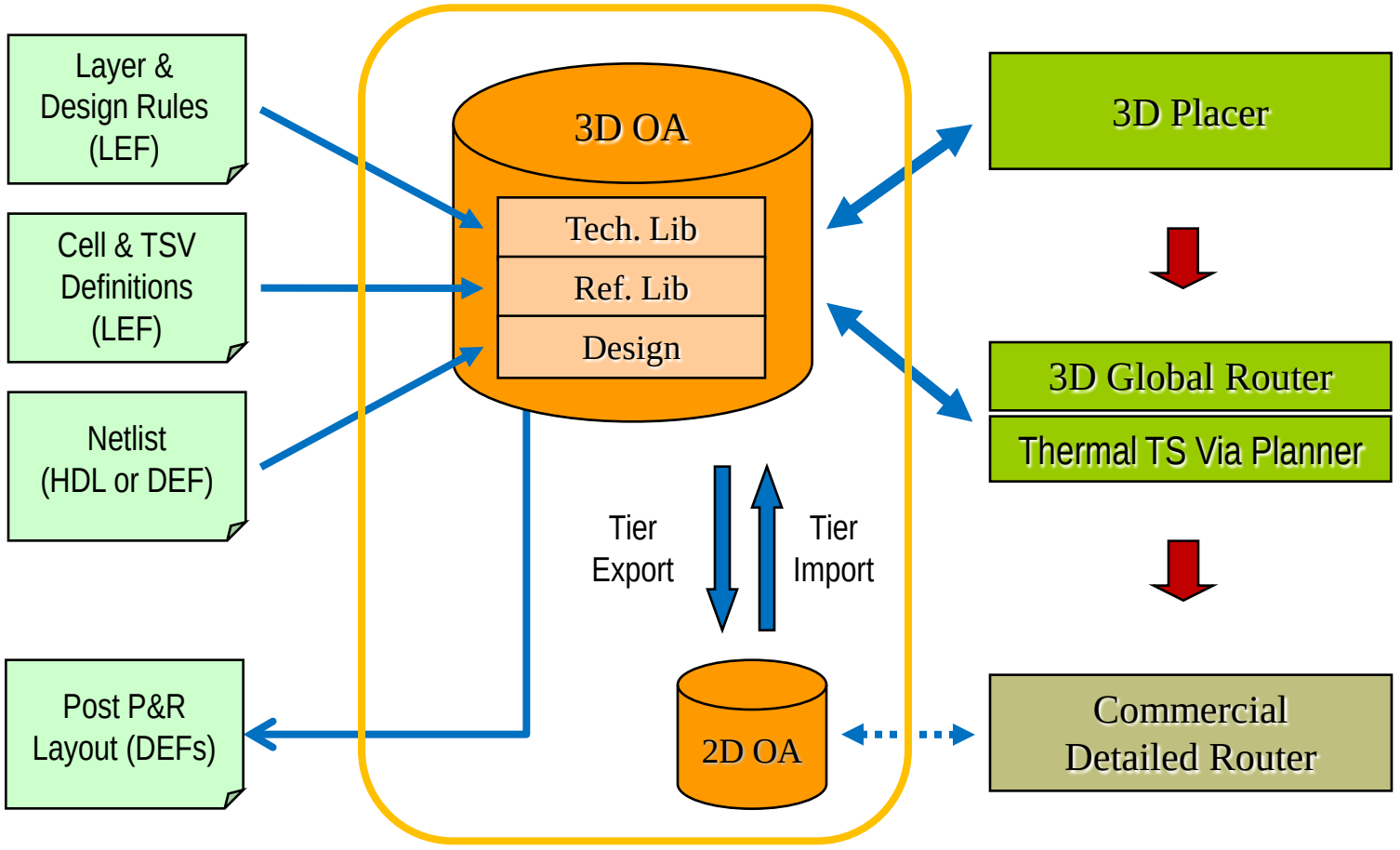
Image source: IBM, ETC'04

Image source: GaTech, ICCAD'09

Need for 3D Physical Synthesis Tools

- Availability of 3D integration technologies
- Lack of 3D EDA design tools
- 3D mixed-size placers are needed
  - Logical and physical hierarchies do not coincide [Cong, ISPD'98]
  - Advantages of a flat approach [Koehl et al., ASPDAC'03]
    - Possibility to perform global optimization
    - Reduce complexity of the design flow and data management

UCLA 3D Physical Design Flow (3D-Craft)



Review of Analytical 3D Placement – the Problem

- Variables
  - $(x_i, y_i, z_i),\ i = 1, 2, \ldots, n$
  - cell i is placed at (x_i, y_i) on layer z_i
- Objective
  - $\sum_e k_e \cdot WL_e(x, y, z) = \sum_e k_e \cdot \left(HPWL_e(x, y) + \alpha_{TSV} TSV_e(z) \right)$
  - Weighted wirelength is capable of handling timing constraints
- Constraints
  - For every pair of cells, either they are placed on different layers, or they are placed on the same layer without overlaps

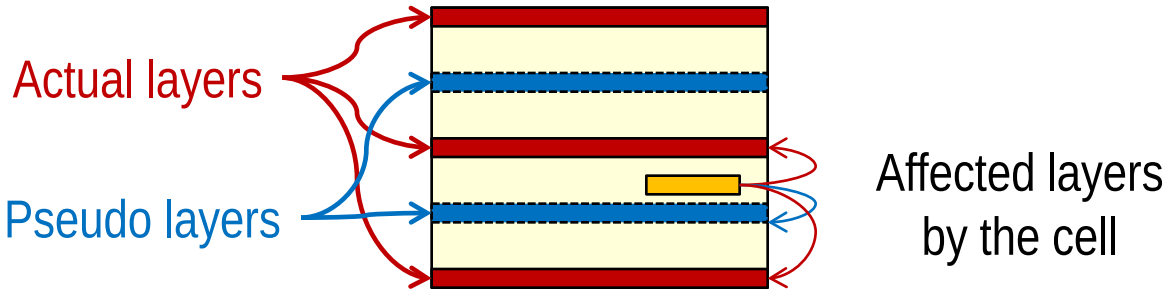
Review of Analytical 3D Placement – Formulation

- minimize $\sum_e k_e WL_e(x,y,z)$

  subject to $A_{i,j,k}(x,y,z) = C_{i,j,k}$ for all i,j,k

  $A'_{i,j,k}(x,y,z) = C'_{i,j,k}$ for all i,j,k

  - Use density constraints on actual layers and pseudo layers to enforce non-overlap constraints [Cong & Luo, ASPDAC'09]



Review of Analytical 3D Placement – the Solver

- Outer iterations – Quadratic Penalty

while (overlaps > threshold)

    minimize $F(s;\mu)$;

    increase penalty factor μ;

$$s = (x, y, z)$$

$$F(s;\mu) = \sum_e k_e WL_e(s) + \frac{\mu}{2}\sum_k \sum_{i,j} (A_{i,j,k}(s) - C_{i,j,k})^2 + \frac{\mu}{2}\sum_k \sum_{i,j} (A'_{i,j,k}(s) - C'_{i,j,k})^2$$

- Inner iterations – Gradient Descent

for($k = 1$; $\left\|\nabla F(s^{(k)};\mu)\right\| > \varepsilon$; $k++$)

    $s^{(k+1)} = s^{(k)} - \alpha \cdot \nabla F(s^{(k)};\mu)$

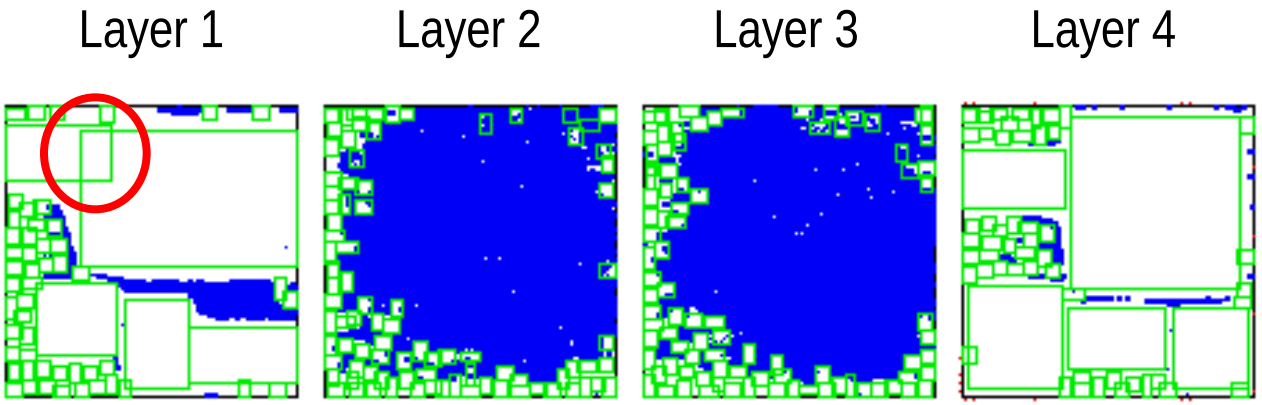
    if (diverge)

        reduce step size α;

        reset placement $k = 0$;

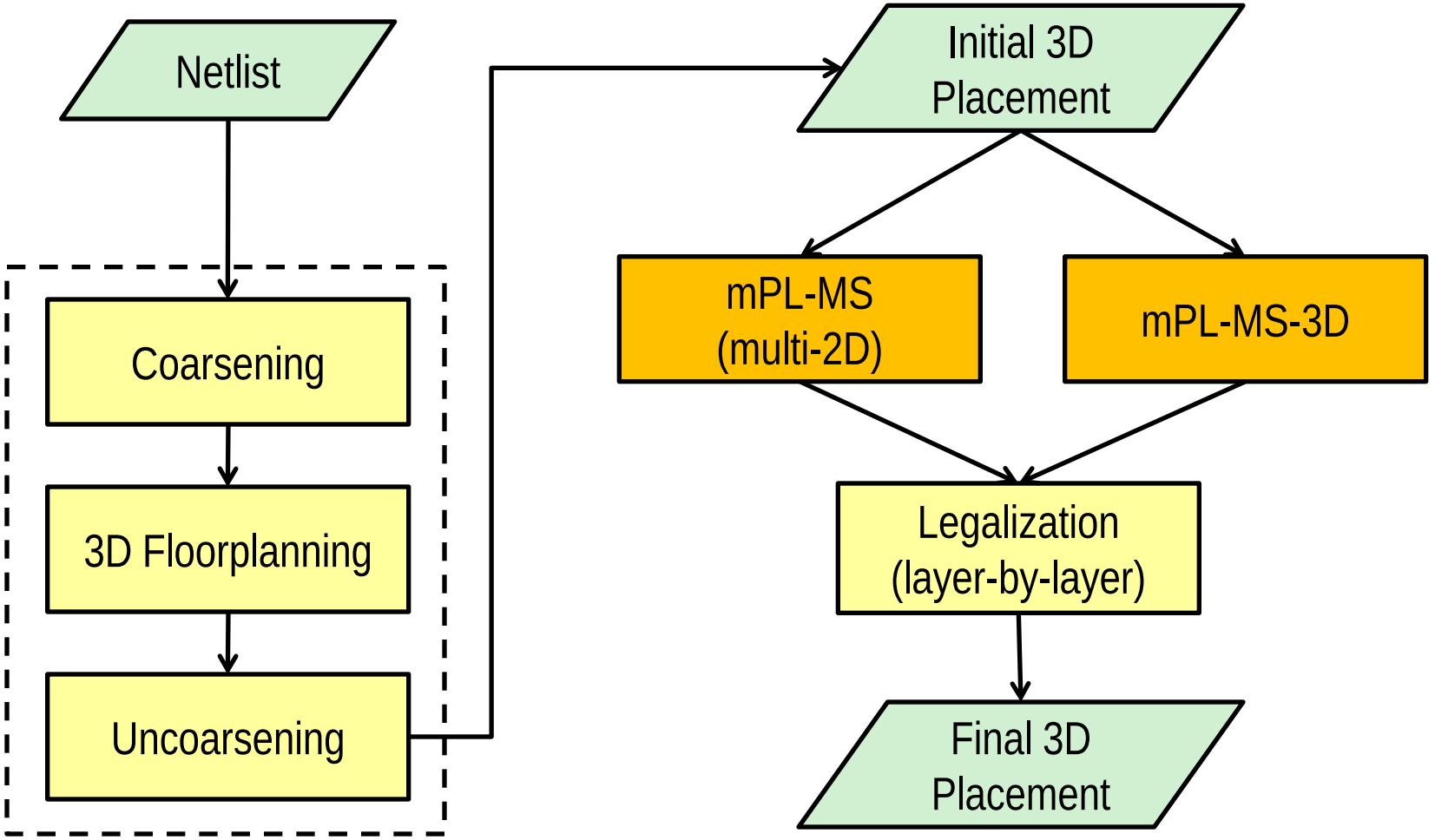
Limitations of Analytical 3D Placers

- Global placement of ibm03



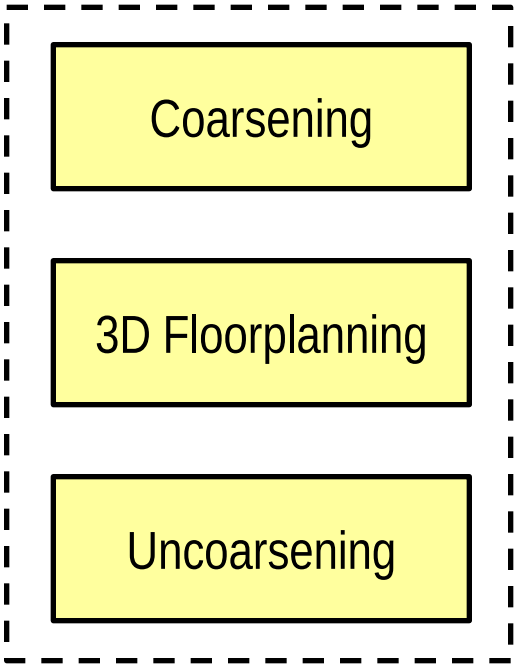
- Analytical placers roughly satisfy area density constraints
  - Remaining overlaps are left for detailed placers
    - It works fine for standard cell 3D placement
  - Overlaps between macros are difficult to remove
    - The global placement will be disturbed greatly

3D Placement Flow for Mixed-Size Circuits



Fix Large Macros Before Analytical 3D Placement

- Coarsening
  - By partitioning the netlist into a limited number (100) of partitions
- 3D floorplanning [Cong et al., ICCAD'04]
  - Performed on the coarsened netlist
    - Thus, the runtime is under control
    - 1-2 min. on a Pentium 4 to floorplan 100 partitions
- Uncoarsening
  - Restore the original netlist
  - Run 2D detailed placer layer-by-layer to optimize macros
  - Fix large macros before analytical 3D placement



Enhancement of Analytical 3D Placement

- Inner iterations – Gradient Descent method

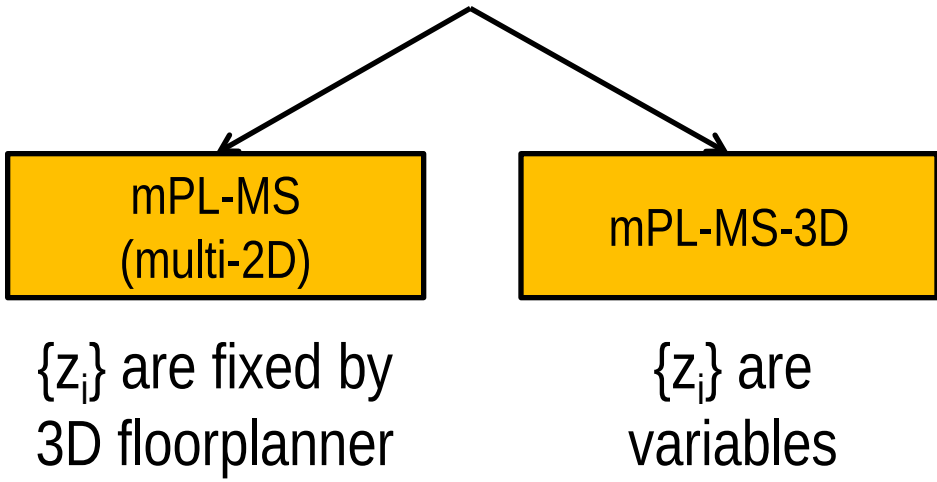
$$\text{for}(\ k=1;\ \left\|\nabla F(s^{(k)};\mu)\right\| > \varepsilon;\ k++\ )$$

$$s^{(k+1)} = s^{(k)} - \alpha \cdot \nabla F(s^{(k)};\mu)$$

if (diverge)

reduce step size α;

reset placement $k = 0$;



- Multiple-Stepsize Scheme
  - Theoretical justification
  - Experimental justification

Multiple-Stepsize Scheme

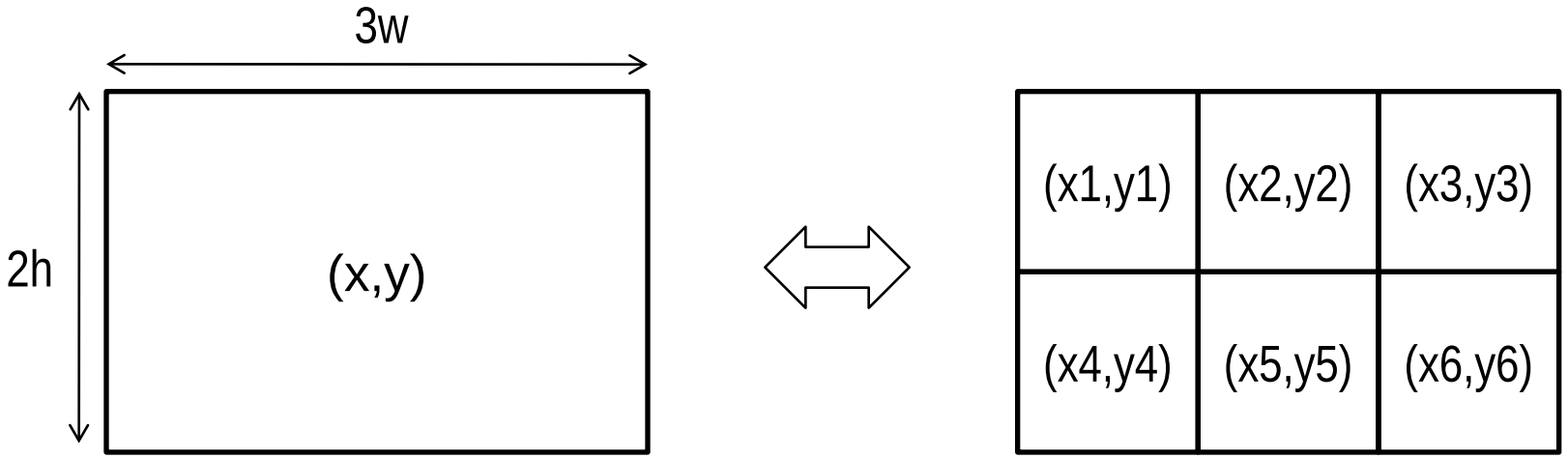
- Stepsize selection in Gradient Descent

$$s^{(k+1)} = s^{(k)} - \alpha \cdot \nabla F(s^{(k)}; \mu)$$

  - Uniform-size circuit : $\alpha_i = \alpha$, for all i
    - The size of every cell and its impact to the density penalty is identical
    - It is natural to use a single stepsize in terms of overlap removal
  - Mixed-size circuit : $\alpha_i = \alpha / A_i$, for all i
    - Larger macros have larger impact to the density penalty
      - A single stepsize scheme requires a very small stepsize
      - Thus, it is very slow to converge
    - The multiple-stepsize allows large stepsize for small cells

Multiple-Stepsize Scheme: Theoretical Justification

- A mixed-size circuit can be decomposed to a uniform-size circuit with additional constraints



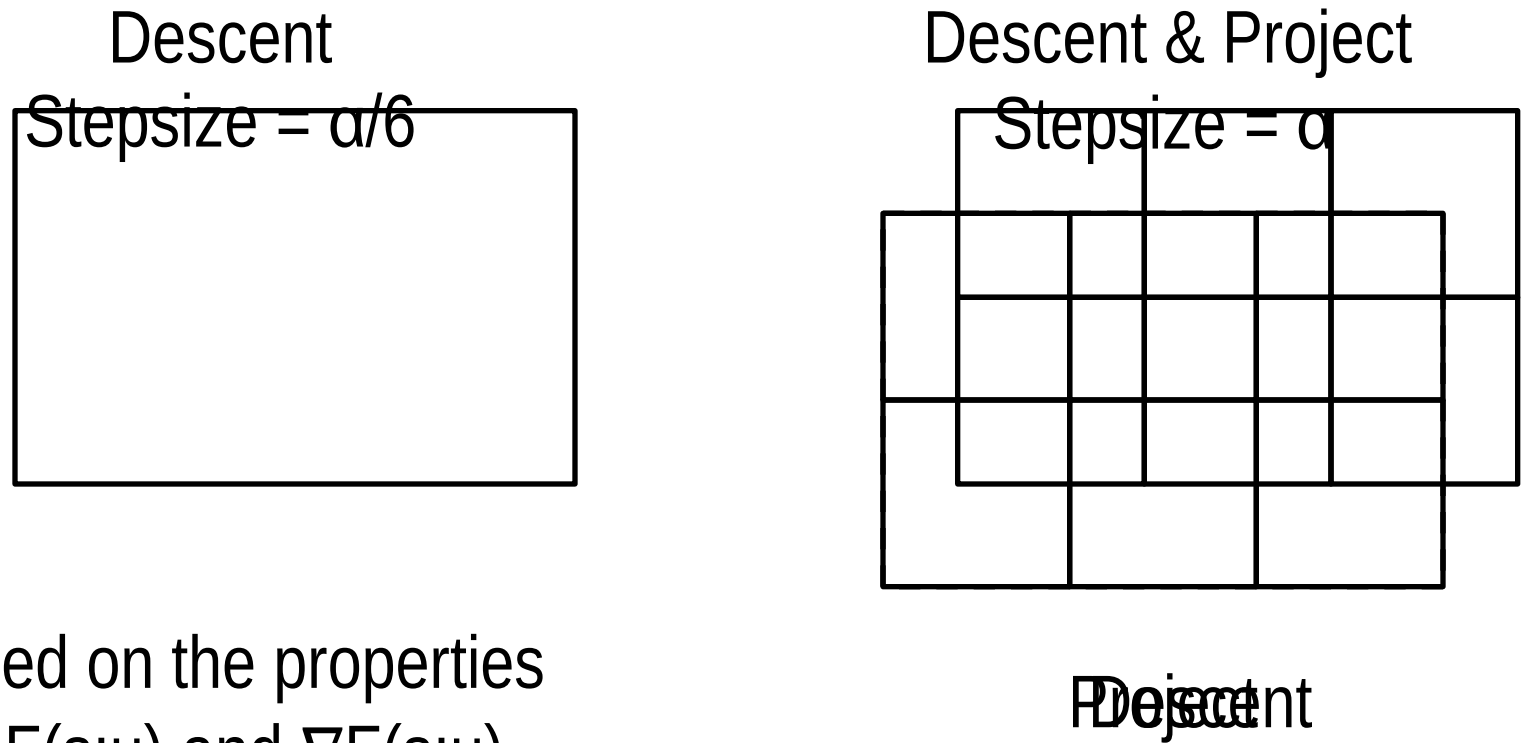
$x_1 + w = x_2$

$y_4 + h = y_1$

……

Multiple-Stepsize Scheme: Theoretical Justification

- One step of gradient descent for the macro is equivalent to one step of gradient projection for the decomposed cells



Based on the properties of F(s;μ) and ∇F(s;μ)

Multiple-Stepsize Scheme: Theoretical Justification

- In general (applicable to 2D/3D placements)
  - Given a mixed-size problem
  - Formulate it as a uniform-size problem with additional constraints
    - Use *Gradient Projection* to solve it with uniform stepsize
    - Derive an equivalent stepsize in the original problem
  - Gradient descent with multiple stepsizes
    - $\alpha_i : \alpha_j = A_i^{-1} : A_j^{-1}$

Multiple-Stepsize Scheme: Experimental Justification

- Replace a given 2D placement
  - Quality – final_HPWL : initial_HPWL
  - #iter – number of iterations to stop (overlap < 15%)
  - Comparing
    - Moderate single stepsize
    - Aggressive single stepsize
    - Aggressive multiple stepsizes

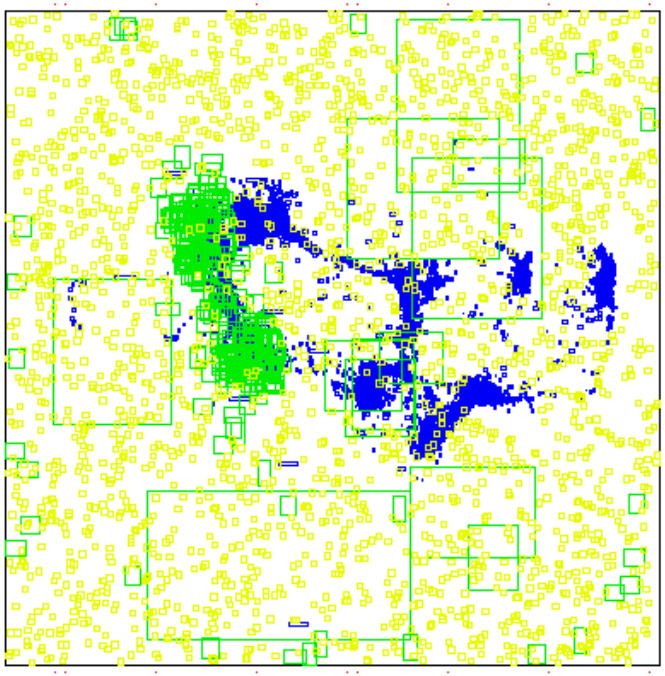
Circuit	moderate single stepsize		aggressive single stepsize		aggressive multiple stepsizes	
	quality	#iter	quality	#iter	quality	#iter
ibm01	1.01	45	1.01	45	1.01	45
ibm02	1.00	195	1.13	200	0.97	60
ibm03	0.99	200	1.12	135	0.97	70
ibm04	0.99	200	1.06	140	0.97	75
ibm05	0.99	150	0.99	70	0.99	70
ibm06	1.00	40	1.00	40	1.00	40
ibm07	1.00	200	1.09	145	0.98	65
ibm08	1.03	200	1.04	200	0.97	70
geomean	**1.01**	**112.88**	**1.06**	**94.17**	**0.99**	**57.58**

- The number of iterations is reduced
  - By allowing large stepsize for a significant number of small cells
- Quality is guaranteed
  - By limiting the stepsize of large macros for stability

Multiple-Stepsize Scheme

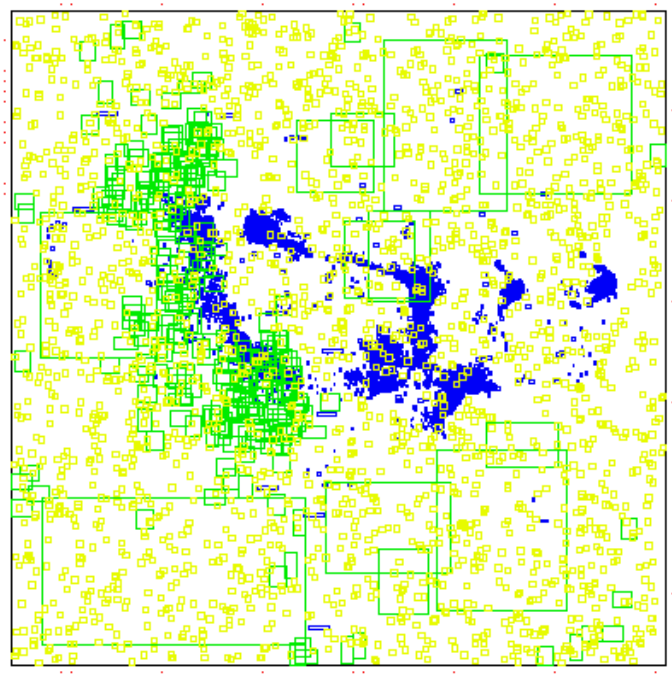
Single stepsize

At 8-th iteration:
HPWL = 2277071.
Logsumexp WL = 4401113.



Multiple stepsizes

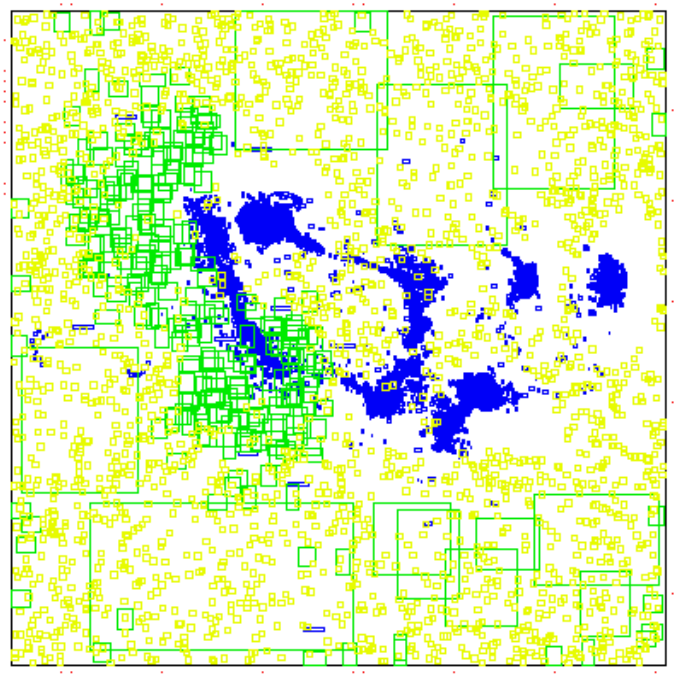
At 8-th iteration:
HPWL = 2420926.
Logsumexp WL = 4617867.



Multiple-Stepsize Scheme

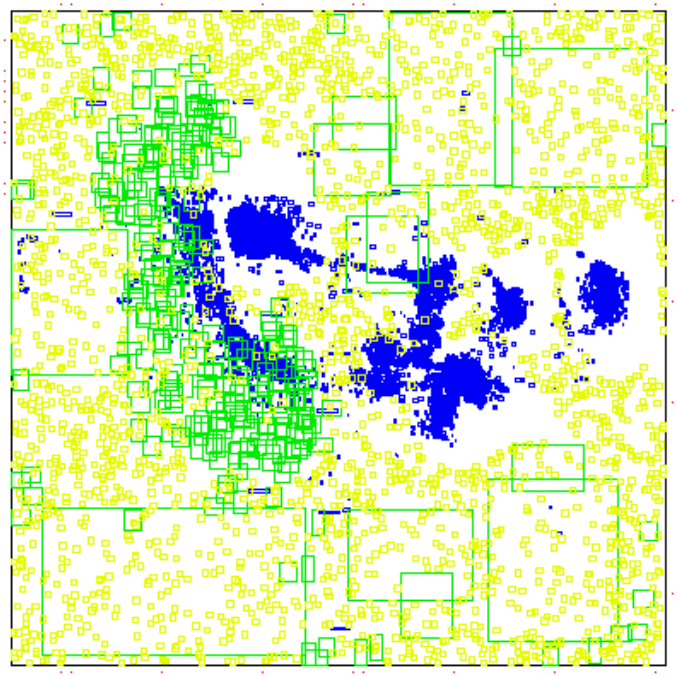
Single stepsize

At 16-th iteration:
HPWL = 2660967.
Logsumexp WL = 4731961.



Multiple stepsizes

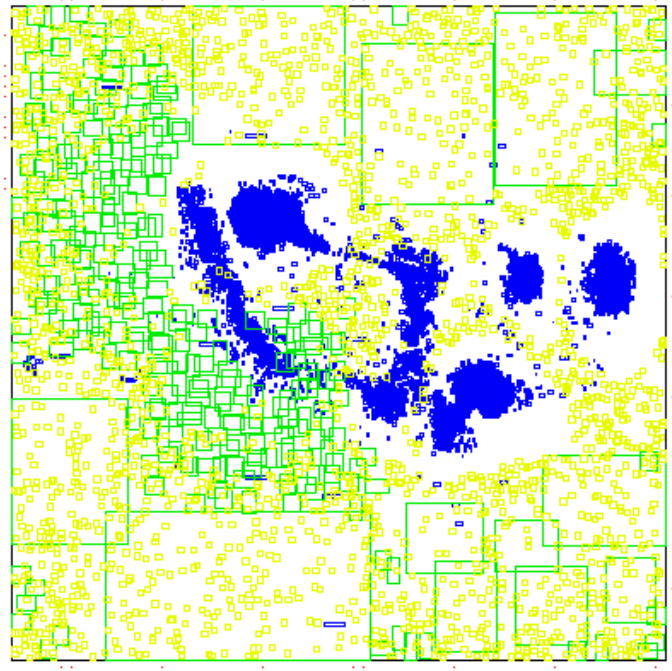
At 16-th iteration:
HPWL = 2725944.
Logsumexp WL = 4734788.



Multiple-Stepsize Scheme

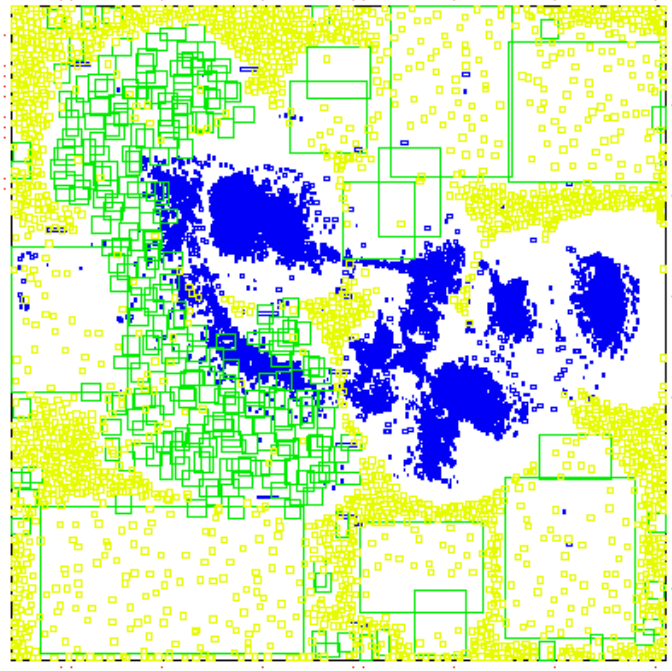
Single stepsize

At 24-th iteration:
HPWL = 3158755.
Logsumexp WL = 5019171.



Multiple stepsizes

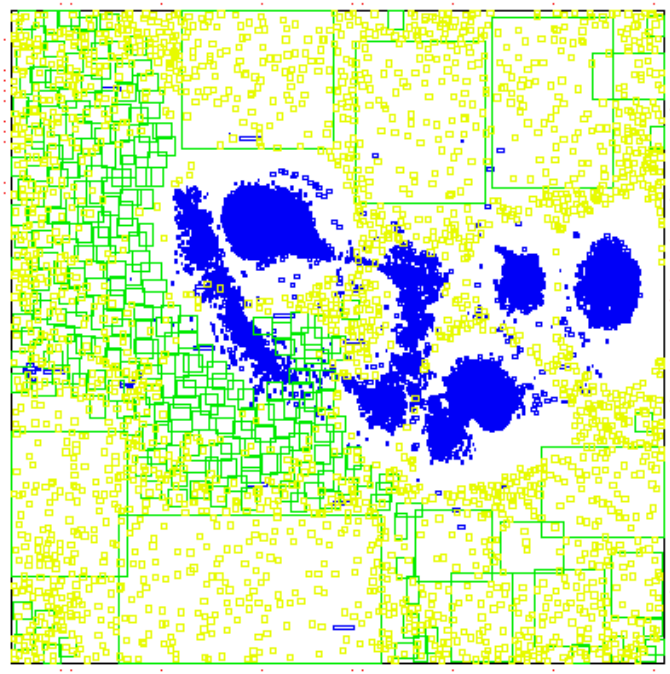
At 24-th iteration:
HPWL = 3218276.
Logsumexp WL = 5030143.



Multiple-Stepsize Scheme

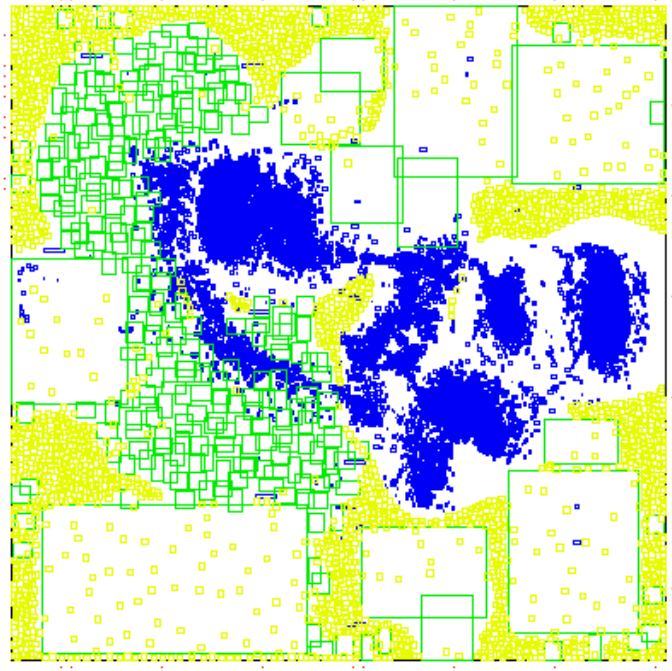
Single stepsize

At 32-th iteration:
HPWL = 3560727.
Logsumexp WL = 5228815.



Multiple stepsizes

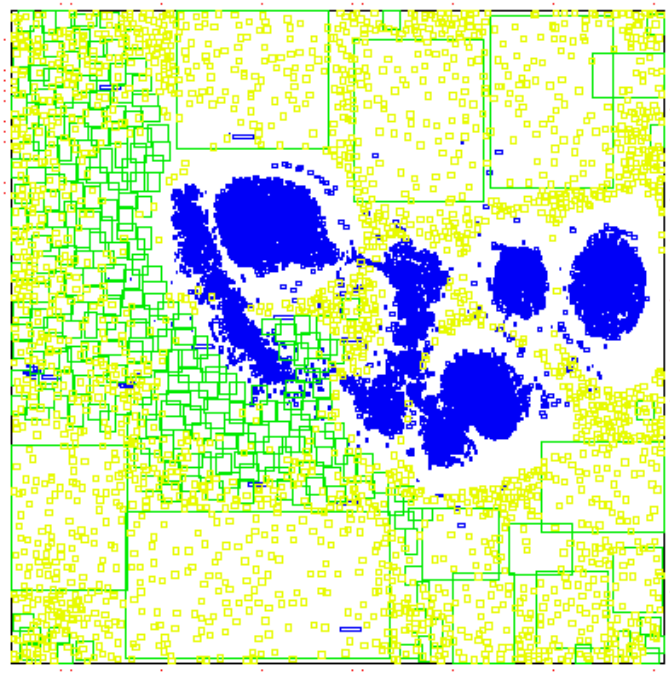
At 32-th iteration:
HPWL = 3582263.
Logsumexp WL = 5256893.



Multiple-Stepsize Scheme

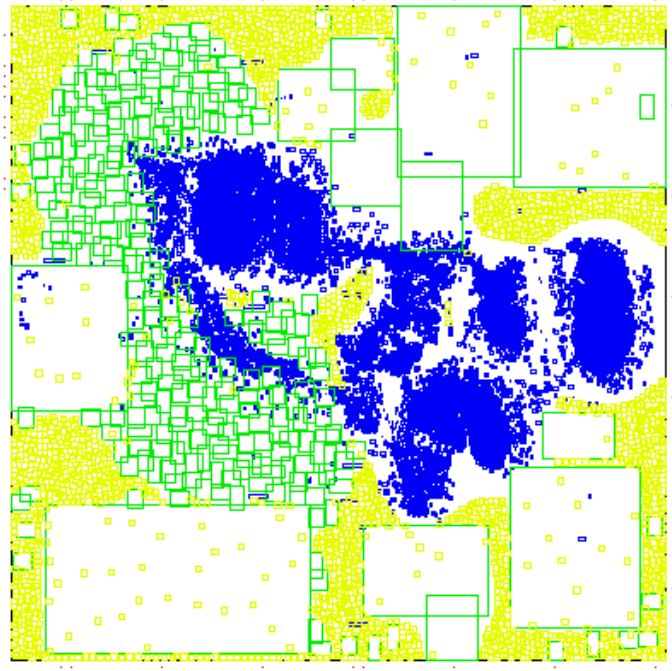
Single stepsize

At 40-th iteration:
HPWL = 3964700.
Logsumexp WL = 5462105.



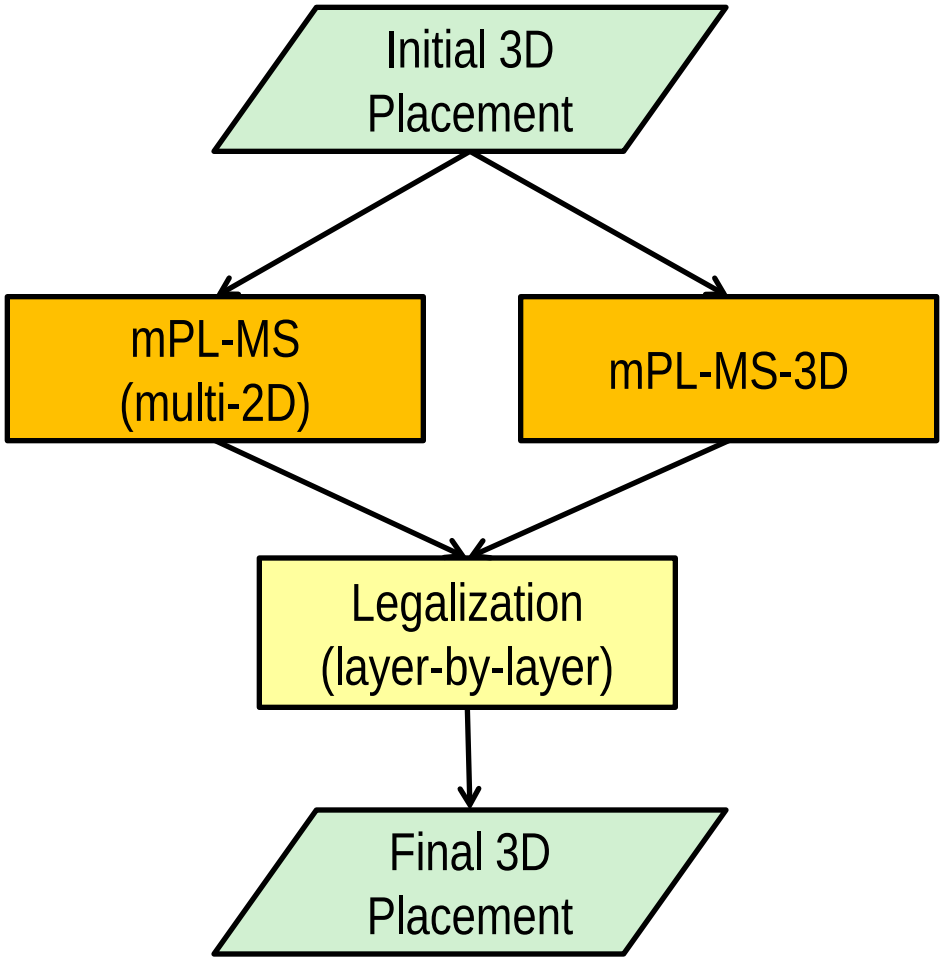
Multiple stepsizes

At 40-th iteration:
HPWL = 3810090.
Logsumexp WL = 5408312.



Experimental Results – Settings

- mPL-MS (multi-2D)
  - $\{z_i\}$ are fixed by 3D floorplanning
- mPL-MS-3D
  - $\{z_i\}$ are variables
- Large macros are fixed
- Small macros are movable



Experimental Results (4-Layer)

Circuit	mPL-MS (multi-2D)				mPL-MS-3D				Folding [ASPDAC'07]	
	gp-WL ($\times 10^6$)	dp-WL ($\times 10^6$)	TSV ($\times 10^3$)	RT (min)	gp-WL ($\times 10^6$)	dp-WL ($\times 10^6$)	TSV ($\times 10^3$)	RT (min)	dp-WL ($\times 10^6$)	TSV ($\times 10^3$)
ibm01	1.47	1.63	2.30	1.55	1.49	1.64	2.39	2.82	1.89	1.88
ibm02	4.12	3.90	3.31	4.04	3.83	3.79	4.98	5.81	4.12	3.23
ibm03	5.46	5.24	4.23	4.01	4.89	4.70	4.36	9.16	failed	failed
ibm04	6.04	5.88	4.90	4.72	5.72	5.56	5.46	9.33	6.80	3.36
ibm05	5.53	5.40	13.98	4.43	5.72	5.65	8.26	5.72	6.92	9.41
ibm06	5.02	5.09	5.62	11.90	4.77	4.86	4.87	9.02	failed	failed
ibm07	7.98	8.03	6.78	7.65	7.11	7.46	7.28	33.49	9.26	5.11
ibm08	9.65	10.00	8.95	10.68	8.12	8.48	9.40	18.25	11.79	7.01
geomean	5.06	**5.07**	5.43	5.17	4.73	**4.80**	5.45	9.03	-	-
	5.00	**5.03**	5.62	4.69	4.70	**4.81**	5.77	9.01	5.85	4.36

- mPL-MS (multi-2D) and mPL-MS-3D achieves 14.0% and 17.9% shorter wirelength than the folding method, respectively

Conclusions and Future Work

- Conclusions
  - Analyze the limitations of analytical 3D placement
  - Develop a 3D placement flow for mixed-size circuits
  - Use the multiple-stepsize scheme to enhance analytical solvers
- Future work
  - TSV density constraints
  - 3D physical hierarchy exploration

THANK YOU!