Challenges and Solutions in Modern Circuit Placement

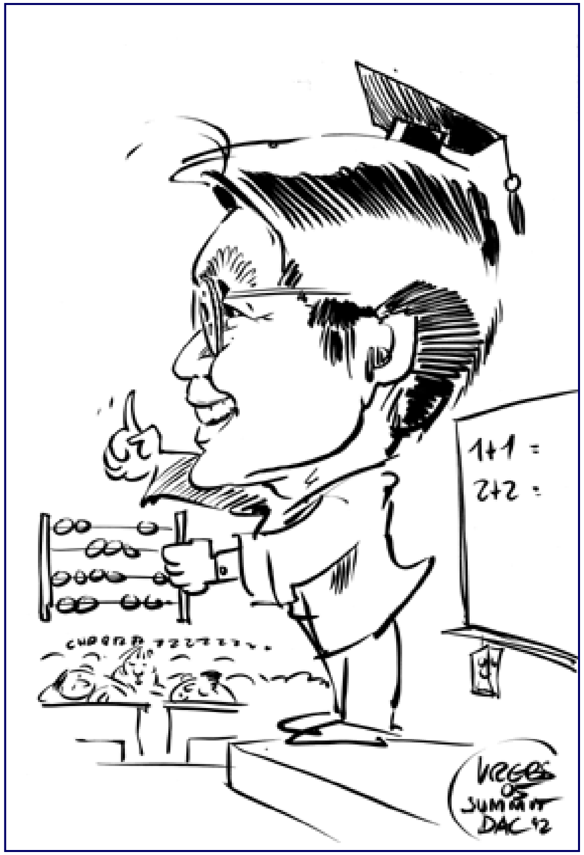
Yao-Wen Chang

ywchang@cc.ee.ntu.edu.tw

http://cc.ee.ntu.edu.tw/~ywchang

National Taiwan University

July 4, 2011



Acknowledgements

Students


Y.-C. Chang 張雲智


T.-C. Chen 陳東傑


Z.-W. Jiang 江哲維


P.-H. Yuh 喻秉鴻


T.-C. Hsu 許天彰


H.-C. Chen 陳信成


Y.-L. Chuang 莊易霖


M.-K. Hsu 徐孟楷


V. Balabanov 包偉力

- Sponsors: IBM, Intel, ITRI, NSC, MediaTek, RealTek, SpringSoft

Outline

Placement Basics

↓

Cell Placement

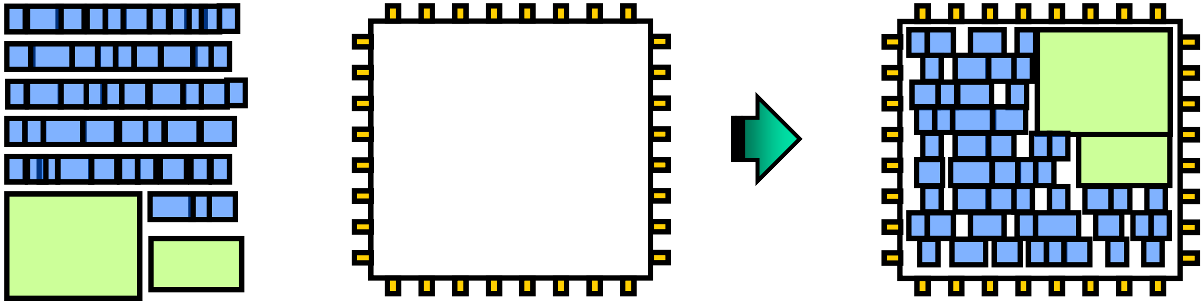
↓

Mixed-Size Placement

↓

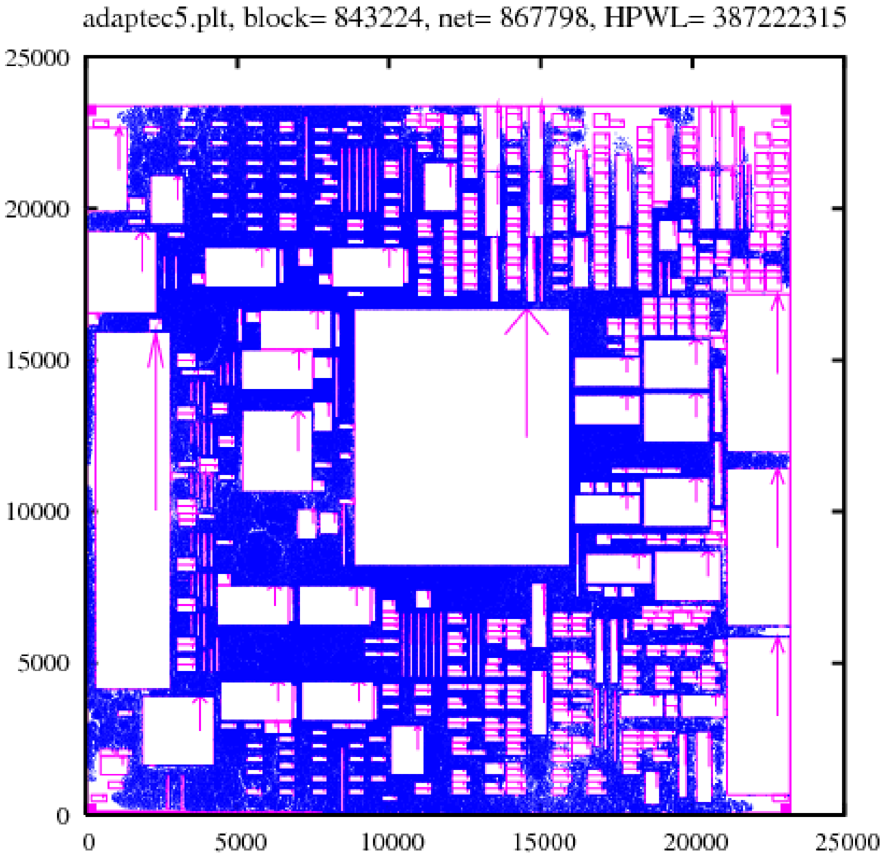
Future Research Directions

VLSI Placement



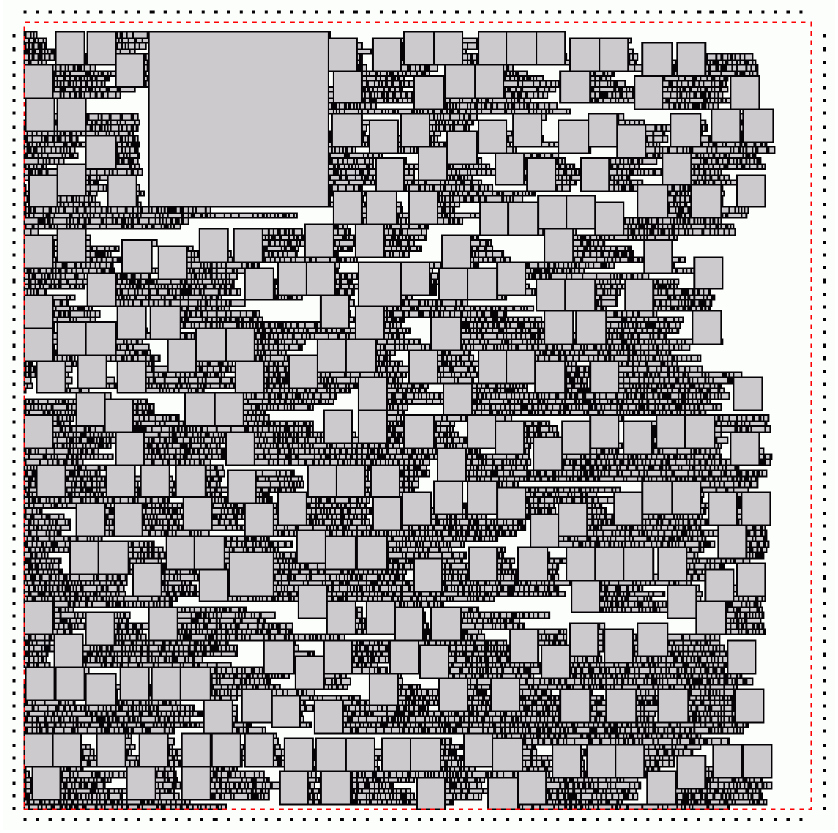
- Placement: Place objects into a fixed die s.t. no objects overlap with each other & some cost metric (e.g., wirelength) is optimized
- Attract much attention due to fast growth in design complexity and many others
  - *EETimes* (4/10/2003): far away from the optimal wirelength
  - Is still far away from optimal??
- More than 20 new academic placers since 2000
- ACM ISPD Placement Contests in 2005, 2006, and **2011**

Example Placements



842K movable cells
646 fixed macros
868K nets

ISPD98 ibm01

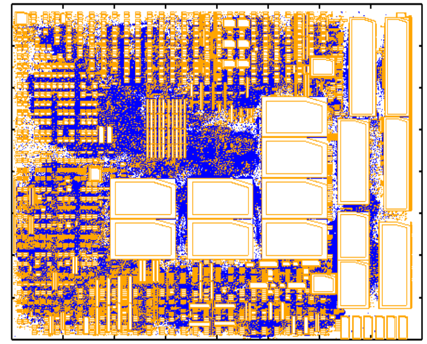


12,752 cells, 247 macros
Amax/Amin = 8416

Wires are not shown here!!

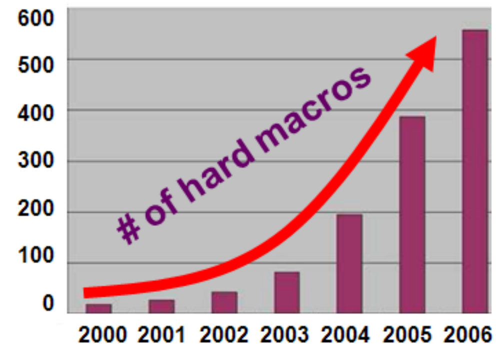
Modern Placement Challenges

- High complexity
  - Millions of objects to be placed
- Placement constraints
  - Preplaced blocks
  - Chip density, etc.
- **Mixed-size placement**
  - Hundreds/Thousands of large macros with millions of small standard cells
- **3D IC design**
  - Through-the-silicon via (TSV) induced multi-tier placement



2.5M placeable objects

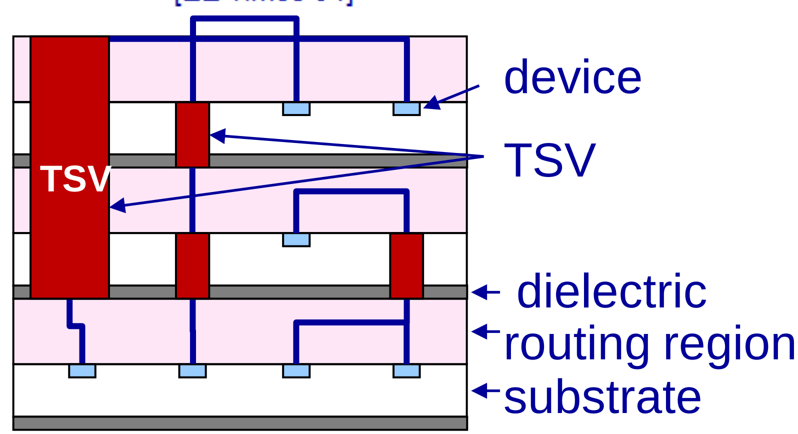
mixed-size design



# of hard macros in SoC

[EE Times'04]

Macros have revolutionized SoC design



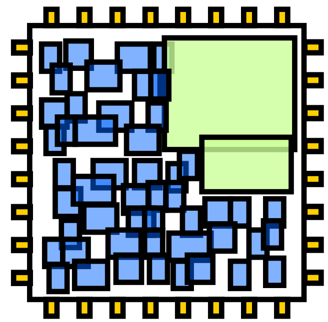
Outline

- Placement Basics
- Cell Placement
- Mixed-Size Placement
- Future Research Directions

NTUplace3 Placement Flow

- Chen, et al., “A high quality analytical placer considering preplaced blocks and density constraint,“ ICCAD-06 (TCAD-08)

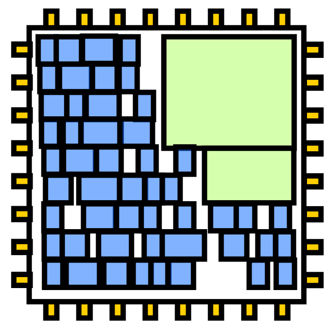
Global Placement (GP)



Computes the best position for each block to minimize the cost (e.g., wirelength), ignoring block overlaps

↓

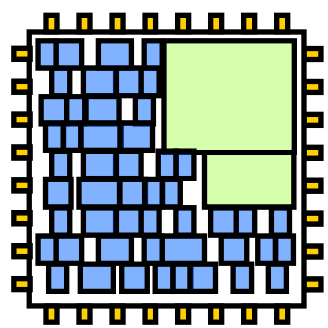
Legalization (LG)



Removes all overlaps among cells

↓

Detailed Placement (DP)



Refines the solution

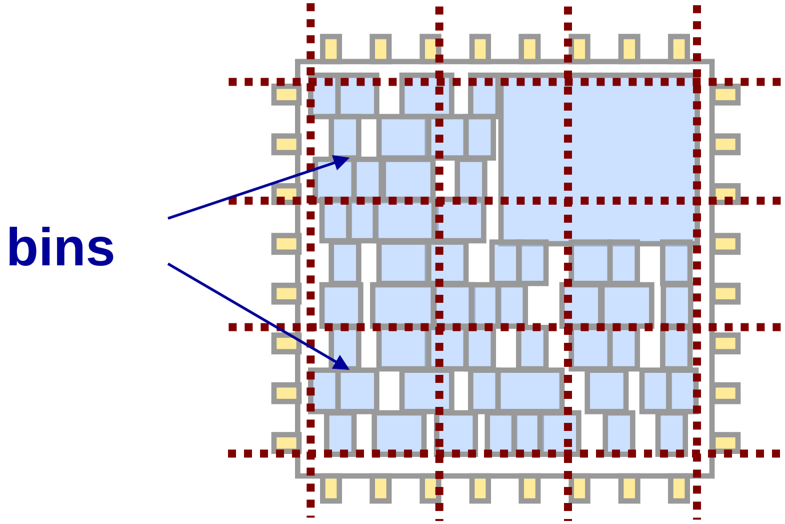
Placement with Density Constraint

- Given the chip region and block dimensions, divide the placement region into bins
- Determine (x, y) for all movable blocks

min W(x, y) // wirelength function

s.t. 1. $\text{Density}_b(x, y) \leq \text{MaximumDensity}_b$ for each bin b

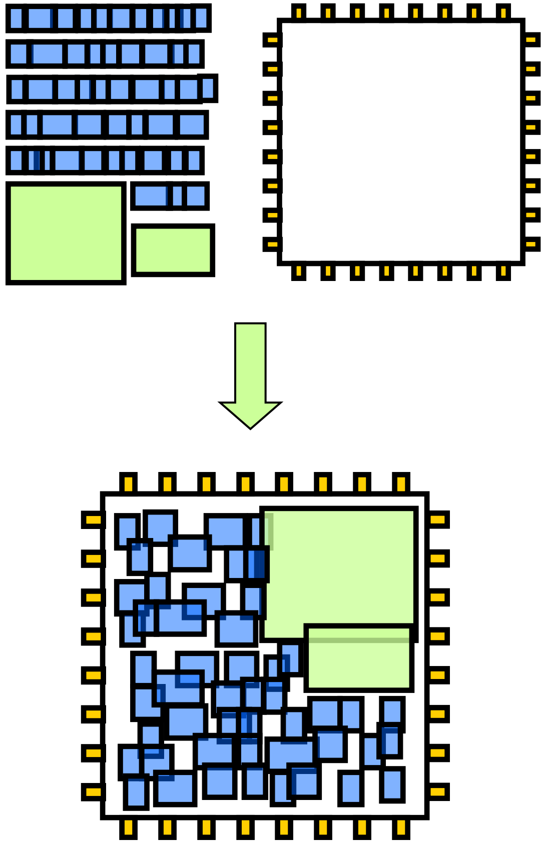
2. No overlap between blocks



$$\text{Density} = \frac{A_{block}}{A_{bin}}$$

Global Placement

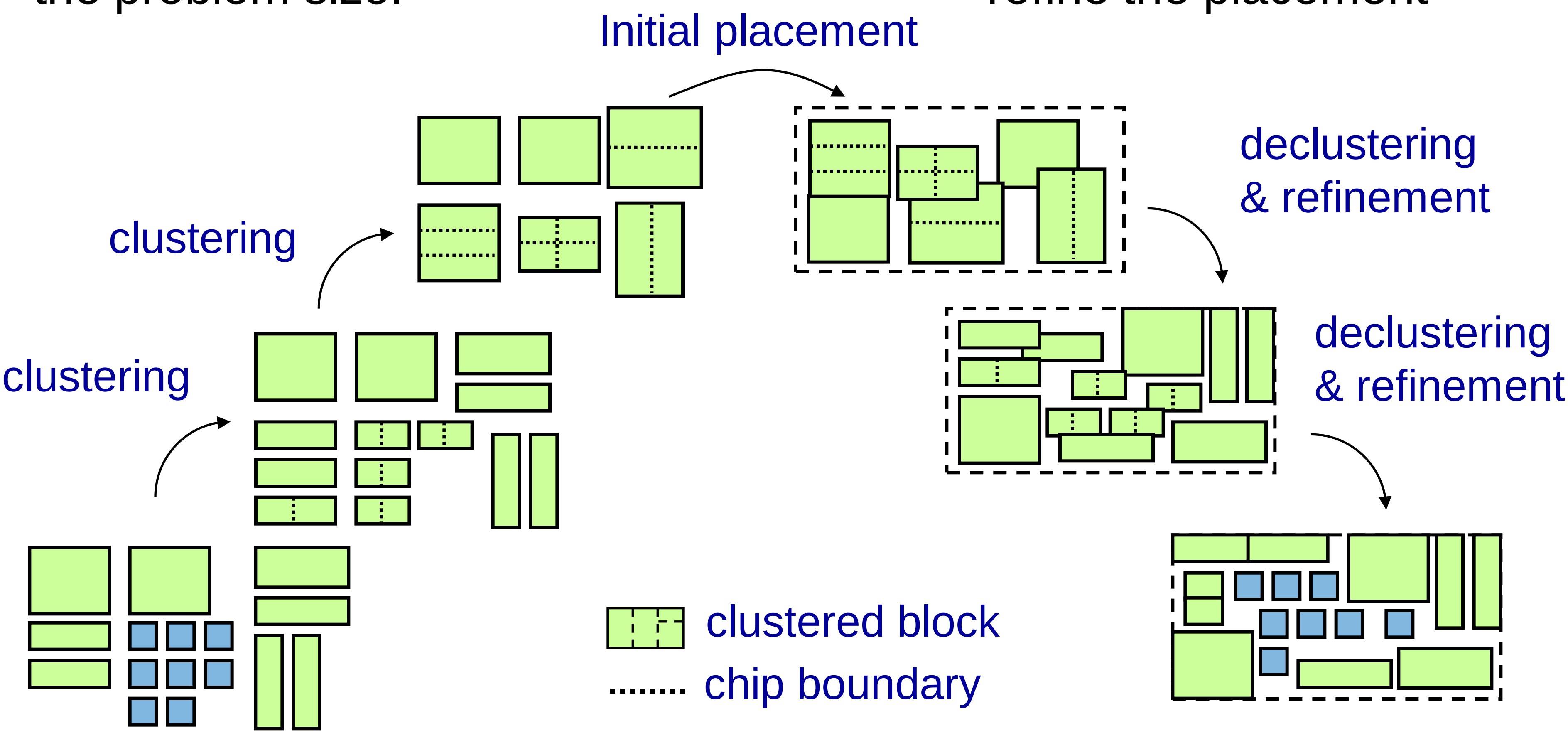
- Placement flow
  - *Global placement*
    - ***Multilevel framework***
    - ***Analytical formulation with a nonlinear objective function***
    - ***Smoothing techniques for preplaced blocks***
    - *Free-space allocation for density control*
  - Legalization
  - Detailed placement



Multilevel Global Placement

Cluster the blocks based on connectivity/area to reduce the problem size.

Iteratively decluster the clusters and further refine the placement



Analytical Placement Model

- Analytical placement during declustering
- Global placement problem (allow overlaps)

$$\textbf{min} \quad W(x, y)$$
$$\textbf{s.t.} \quad D_b(x, y) \le M_b$$

Minimize HPWL (wirelength)
D_b: density for bin *b*
M_b: max density for bin *b*

- Relax the constraints into the objective function

$$\textbf{min} \quad W(x, y) + \boldsymbol{\lambda}\Sigma(D_b(x, y) - M_b)^2$$

  - Use the gradient method to solve it
  - Increase **λ** gradually to find the optimal (x, y) under density constraint

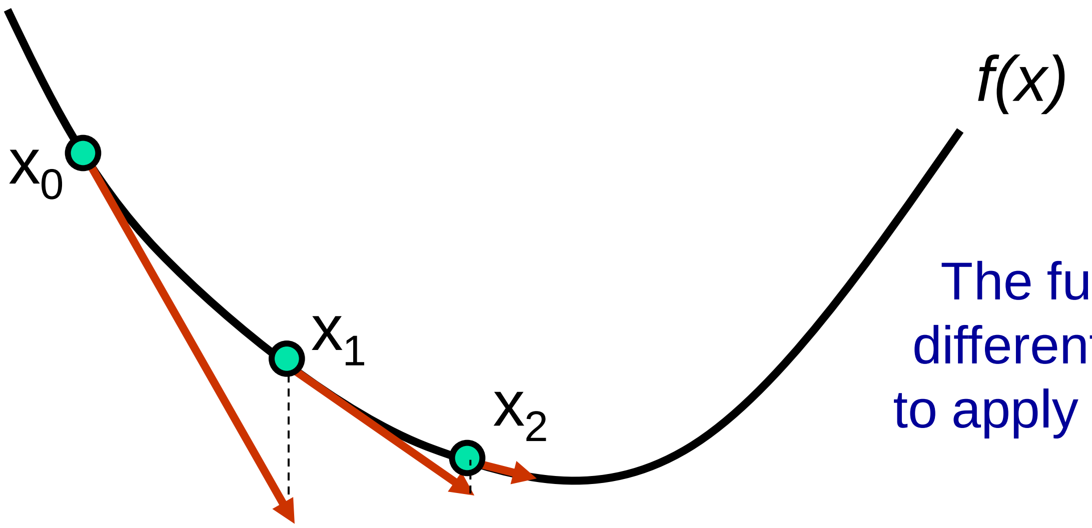
Gradient Solver

min *f(x)*

[Gradient Solver]

$x_0 \leftarrow$ initial value
Repeat until convergence

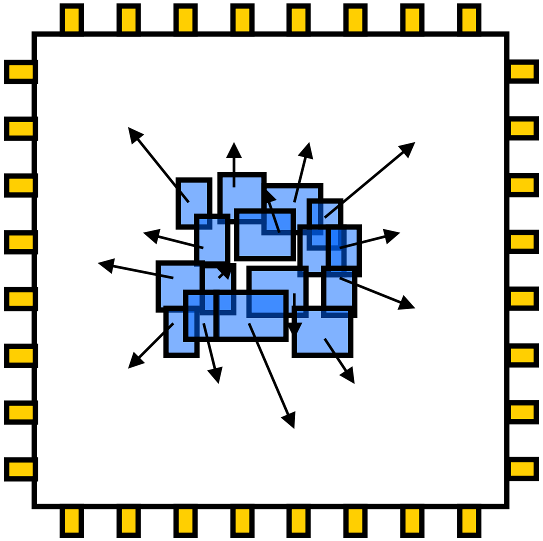
$$x_{i+1} = x_i - f'(x)|_{x=xi} * \text{stepsize}$$

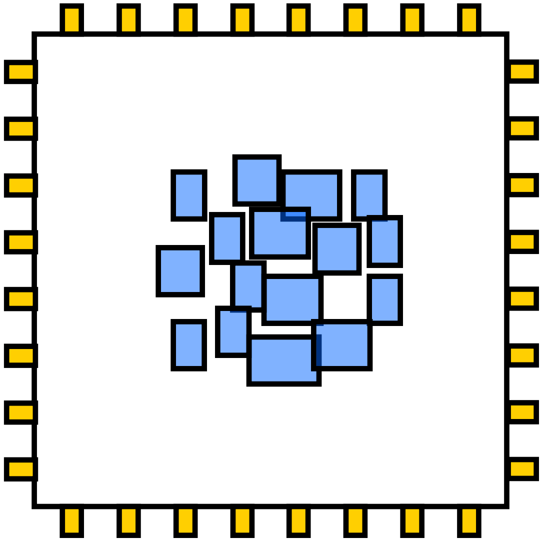


The function *f* must be differentiable and smooth to apply the gradient solver.

Dynamic Step-Size Control

- Step size is too large
  - → May not converge to a good solution
- Step size is too small
  - → Incur long running time
- Adjust the step size s.t. the average Euclidean movement of all blocks is a fixed value





step size $\alpha_k = \dfrac{s}{\|\mathbf{d}_k\|_2}$

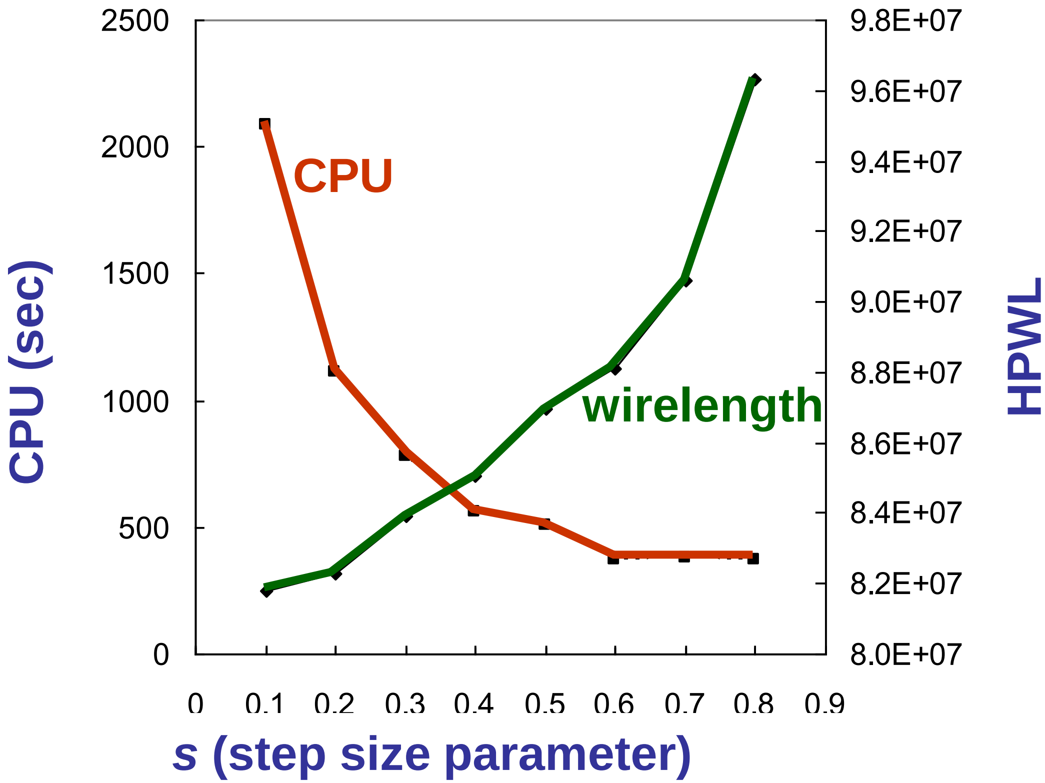
$\mathbf{d}_k$ conjugate directions

s a user-specified value

~ 0.2 * bin width

Effects of the Step-Size Parameter

- Small step size
  - Larger runtime, smaller HPWL (wirelength)
- Large step size
  - Smaller runtime, larger HPWL

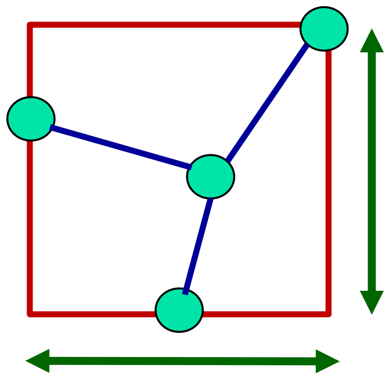


HPWL Wirelength Model

- Ideal: Half-perimeter wirelength (HPWL) model

$$W(\mathbf{x},\mathbf{y}) = \sum_{e \in E} \left(\max_{v_i, v_j \in e} |x_i - x_j| + \max_{v_i, v_j \in e} |y_i - y_j| \right)$$

  - Is not smooth and differentiable
  - Approximations: quadratic, L_p-norm, log-sum-exp, CHKS wirelength models, etc.



Log-sum-exp (LSE) Wirelength Model

- Log-sum-exp (LSE) wirelength model [Naylor et al., 2001]

$$W_{LSE}(\mathbf{x},\mathbf{y}) = \gamma \sum_{e \in E} (\ln \sum_{v_k \in e} \exp(x_k/\gamma) + \ln \sum_{v_k \in e} \exp(-x_k/\gamma) + \ln \sum_{v_k \in e} \exp(y_k/\gamma) + \ln \sum_{v_k \in e} \exp(-y_k/\gamma))$$

$$\max\{\mathbf{x}\} = -\min\{-\mathbf{x}\}; \quad \max\{\mathbf{x}\} \cong \gamma \ln \sum_{v_k \in e} \exp(x_k/\gamma)$$

  – Is an effective smooth & differentiable approximation for HPWL
  – Approaches exact HPWL when $\gamma \rightarrow 0$
  – **Has dominated modern placement for 10+ years!**

Can we do better??

Our Weighted-Average (WA) Model

$$W_{WA}(\mathbf{x},\mathbf{y}) = \sum_{e\in E}\left(\frac{\sum_{v_i\in e} x_i \exp(x_i/\gamma)}{\sum_{v_i\in e}\exp(x_i/\gamma)} - \frac{\sum_{v_i\in e} x_i \exp(-x_i/\gamma)}{\sum_{v_i\in e}\exp(-x_i/\gamma)} + \frac{\sum_{v_i\in e} y_i \exp(y_i/\gamma)}{\sum_{v_i\in e}\exp(y_i/\gamma)} - \frac{\sum_{v_i\in e} y_i \exp(-y_i/\gamma)}{\sum_{v_i\in e}\exp(-y_i/\gamma)} \right)$$

- **1st model that outperforms LSE theoretically & empirically** [Hsu, Chang, Balabanov, DAC-11]
- Weighted average of a set of x coordinates, $\mathbf{x}_e$, of a net e:
  - $X(\mathbf{x}_e)$ can approximate the maximum value of $\mathbf{x}_e$ by setting the weight function of x_i: $\mathbf{F(x_i) = \exp(x_i/\gamma)}$, a fast growing function

$$X(\mathbf{x}_e) = \frac{\sum_{v_i\in e} x_i F(x_i)}{\sum_{v_i\in e} F(x_i)} \quad \xrightarrow{\text{max}} \quad X_{\max}(\mathbf{x}_e) = \frac{\sum_{v_i\in e} x_i \exp(x_i/\gamma)}{\sum_{v_i\in e}\exp(x_i/\gamma)}$$

- Is an effective smooth, differentiable, quasiconvex function for HPWL approximation
- Approaches exact HPWL when $\gamma \rightarrow 0$

Popular Wirelength Models

HPWL

$$W(\mathbf{x}, \mathbf{y}) = \sum_{\text{net } e} \left(\max_{v_i, v_j \in e} |x_i - x_j| + \max_{v_i, v_j \in e} |y_i - y_j| \right)$$

quadratic

$$\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_{ij} [(x_i - x_j)^2 + (y_i - y_j)^2]$$

Log-sum-exp

$$\gamma \sum_{e \in E} \Big(\log \sum_{v_k \in e} \exp(x_k/\gamma) + \log \sum_{v_k \in e} \exp(-x_k/\gamma) + \log \sum_{v_k \in e} \exp(y_k/\gamma) + \log \sum_{v_k \in e} \exp(-y_k/\gamma) \Big).$$

L_p-norm

$$\sum_{e \in E} \Big(\big(\sum_{v_k \in e} x_k^p\big)^{\frac{1}{p}} - \big(\sum_{v_k \in e} x_k^{-p}\big)^{-\frac{1}{p}} + \big(\sum_{v_k \in e} y_k^p\big)^{\frac{1}{p}} - \big(\sum_{v_k \in e} y_k^{-p}\big)^{-\frac{1}{p}} \Big)$$

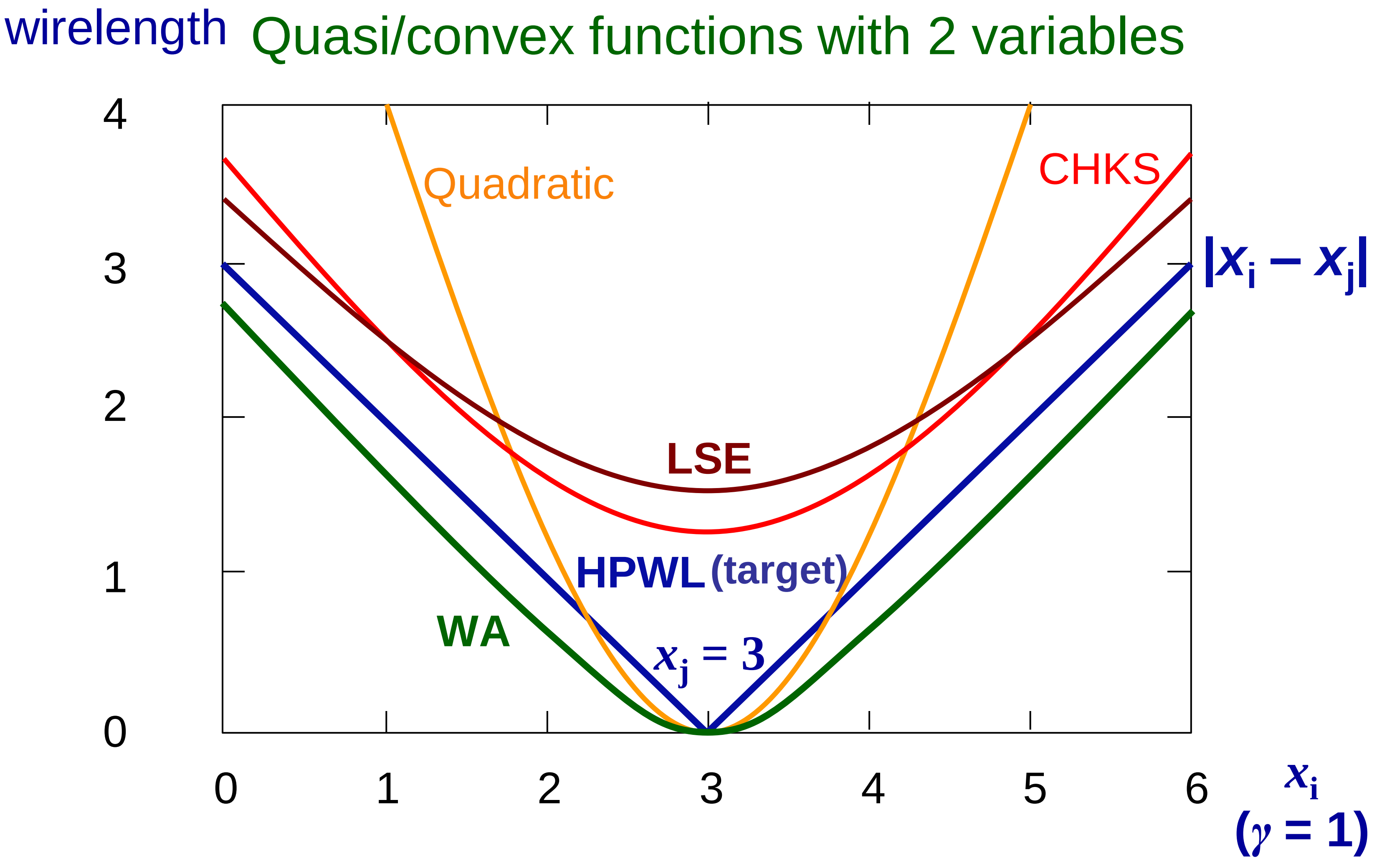
CHKS

$$CHKS(x_1, x_2) = \frac{\sqrt{(x_1 - x_2)^2 + t^2} + x_1 + x_2}{2},$$

Weighted-average

$$\sum_{e \in E} \left(\frac{\sum_{v_i \in e} x_i \exp(x_i/\gamma)}{\sum_{v_i \in e} \exp(x_i/\gamma)} - \frac{\sum_{v_i \in e} x_i \exp(-x_i/\gamma)}{\sum_{v_i \in e} \exp(-x_i/\gamma)} + \frac{\sum_{v_i \in e} y_i \exp(y_i/\gamma)}{\sum_{v_i \in e} \exp(y_i/\gamma)} - \frac{\sum_{v_i \in e} y_i \exp(-y_i/\gamma)}{\sum_{v_i \in e} \exp(-y_i/\gamma)} \right).$$

Popular Wirelength Model Comparisons



Theoretical Comparisons

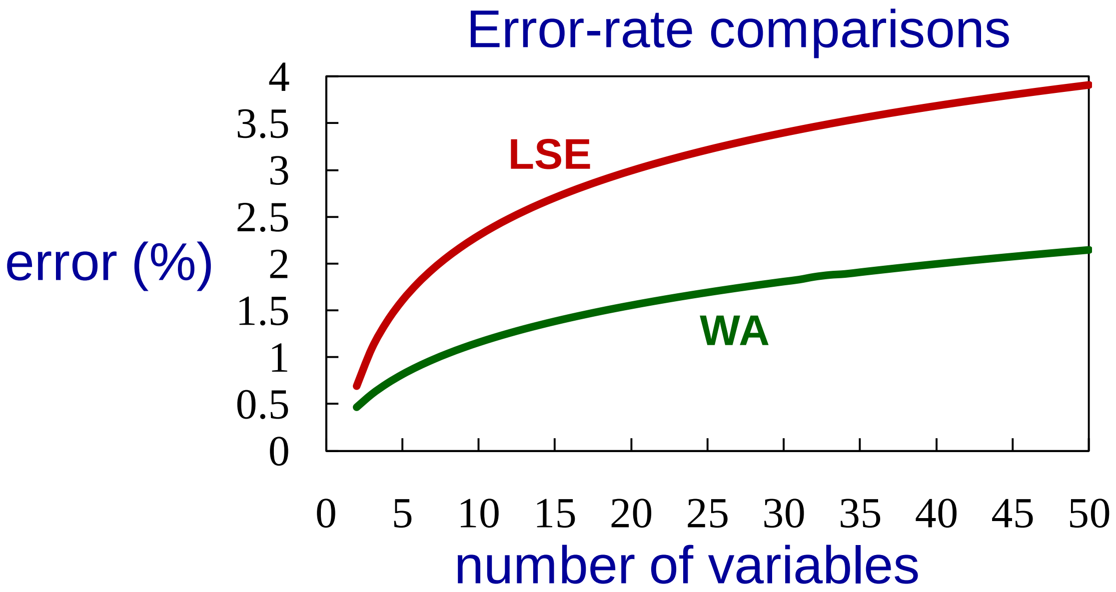
- Theorem: The estimation error bound of the WA model is

$$0 \le \varepsilon_{WA}(\mathbf{x}_e) \le \frac{\gamma(x_{\max} - x_{\min})}{1 + e^{(x_{\max} - x_{\min})/n}}.$$

$\mathbf{x}_e = \{x_i | v_i \in e\}$: a set **x** of coordinates associated with net e

- Theorem: The error upper bound of the WA model is smaller than that of the LSE model:

$$\varepsilon_{WA}(\mathbf{x}_e) \le \varepsilon_{LSE}(\mathbf{x}_e) = \gamma \ln n$$



Wirelength Model Comparison

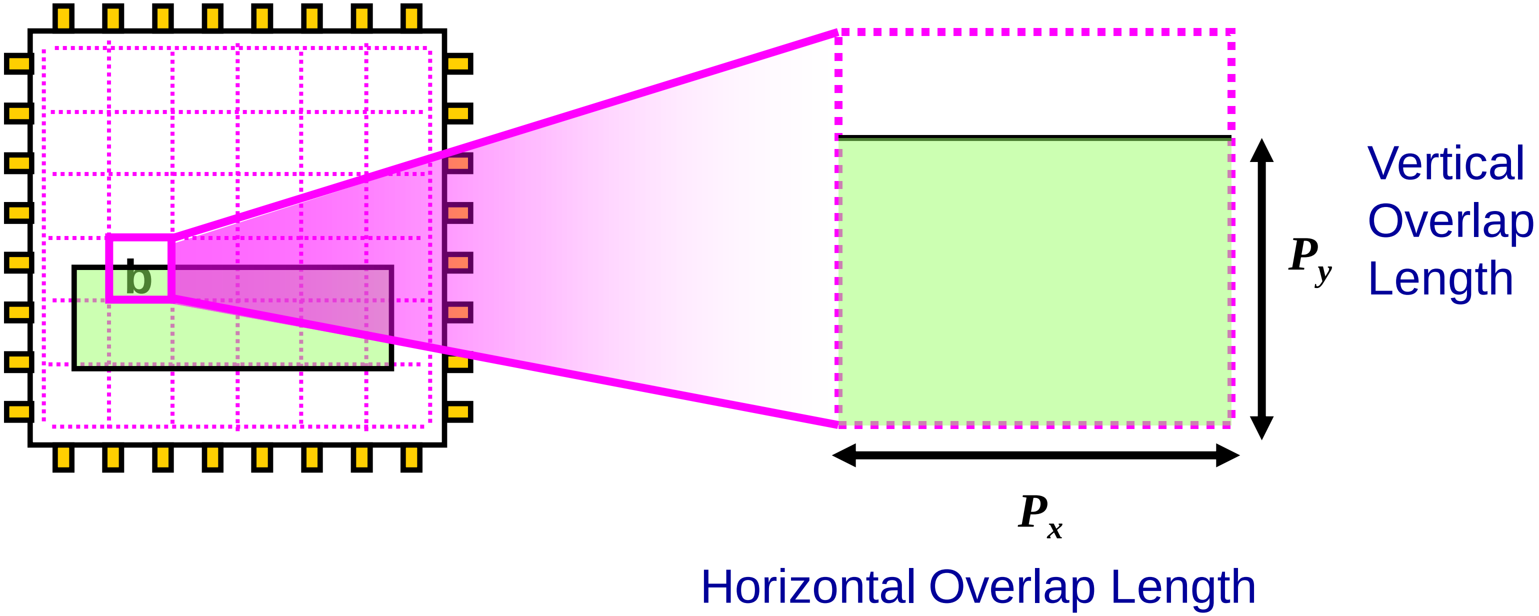
- Integrated both the LSE and WA models into NTUplace3 [ICCAD-06], a leading academic placer
- Used ISPD-06 placement benchmark circuits
  - #Cells: 330K—2481K, #Nets: 338K—2636K
- The WA model can achieve averagely 2% shorter total wirelength than the LSE model

Wirelength Model	Wirelength	CPU Time
LSE	1.000	1.000
WA	**0.980**	**1.066**

- The results show that WA outperforms LSE consistently

Density Model

- Compute the block area in each bin to obtain the bin density

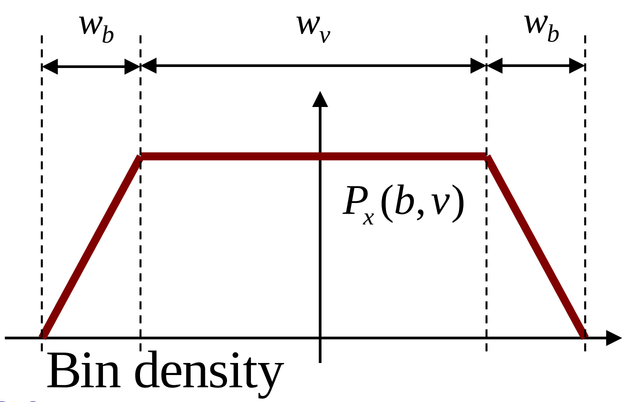


Bin density

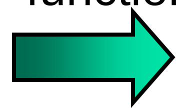
$$D_b(\mathbf{x},\mathbf{y}) = \sum_{v \in V} P_x(b,v)P_y(b,v)$$

Density Smoothing

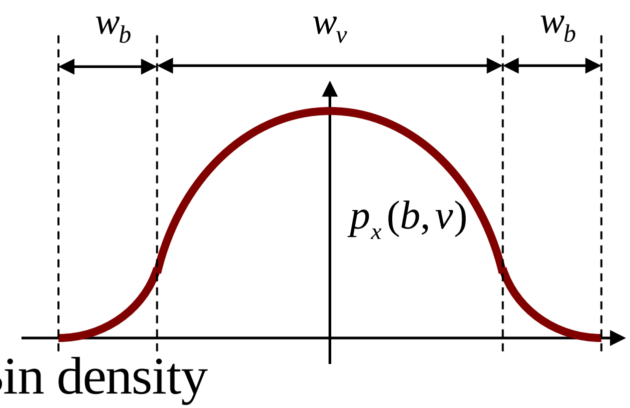
- Apply the bell-shaped function to make bin density function smooth [Kahng & Wang, ICCAD-04]



Bell-shaped smoothing function



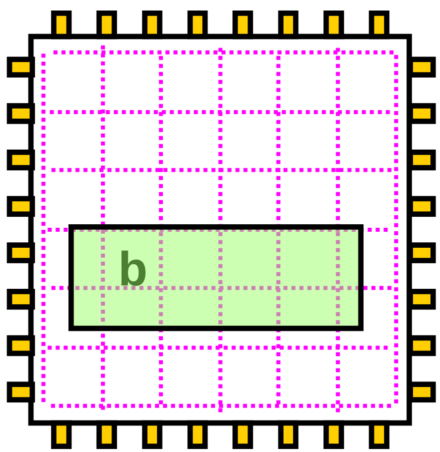
Continuous & differentiable



Overlap length function

$$D_b(\mathbf{x},\mathbf{y}) = \sum_{v \in V} P_x(b,v) P_y(b,v)$$

$$D'_b(\mathbf{x},\mathbf{y}) = \sum_{v \in V} c_v p_x(b,v) p_y(b,v)$$



$$p_x(b,v) = \begin{cases} 1 - a d_x^2, & 0 \le d_x \le w_v/2 + w_b \\ b(d_x - 2w_b - 2w_g)^2, & w_v/2 + w_b \le d_x \le w_v/2 + 2w_b \\ 0, & w_v/2 + 2w_b \le d_x, \end{cases}$$

where

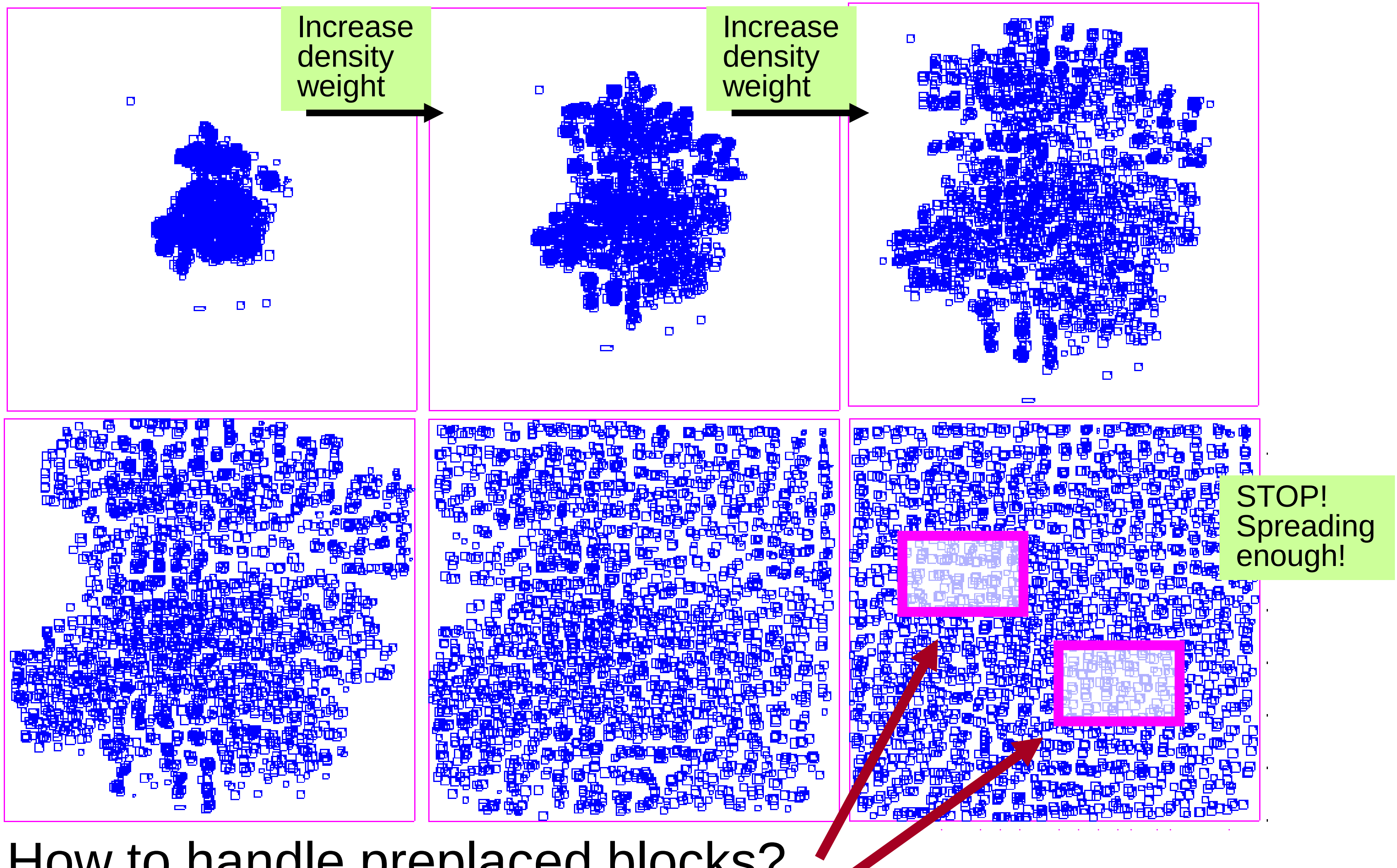
$$a = 4/((w_v + 2w_b)(w_v + 4w_b))$$
$$b = 2/(w_b(w_v + 4w_b)),$$

w_b bin width

w_v block width

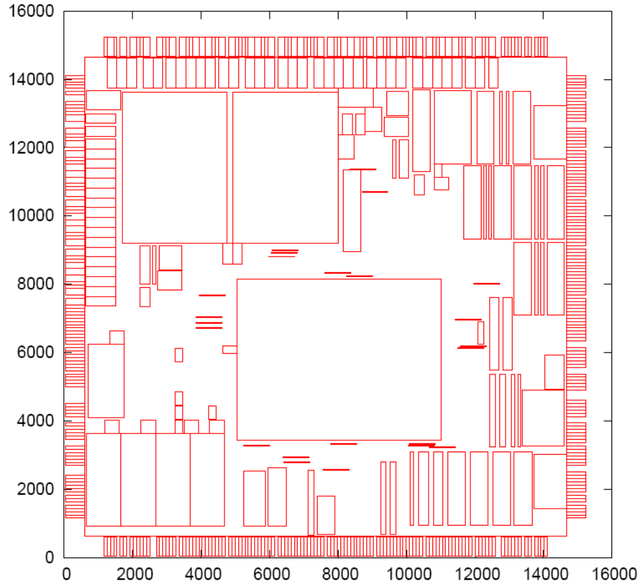
c_v normalization factor

Placement Process

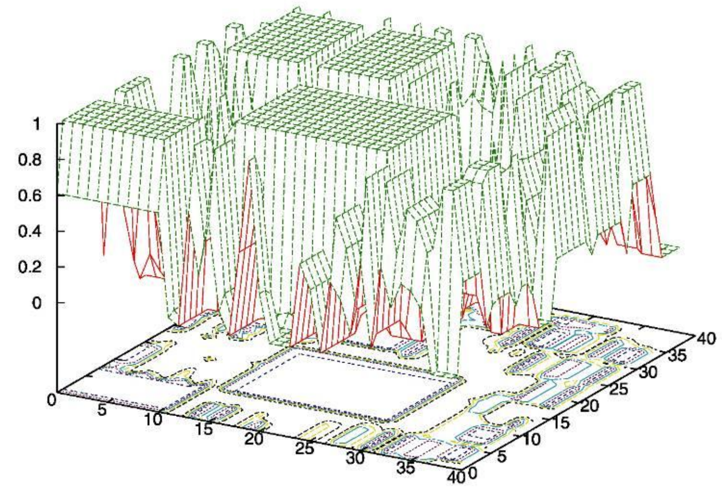


- How to handle preplaced blocks?
  - Pre-defined density makes cell spreading harder.

Density Map with Preplaced Blocks



Placement with preplaced blocks



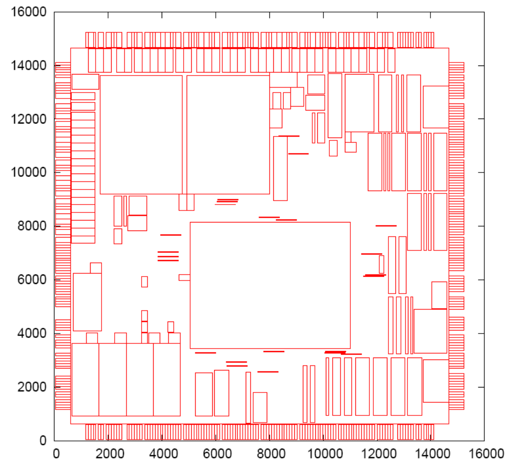
Corresponding density map

Two major problems

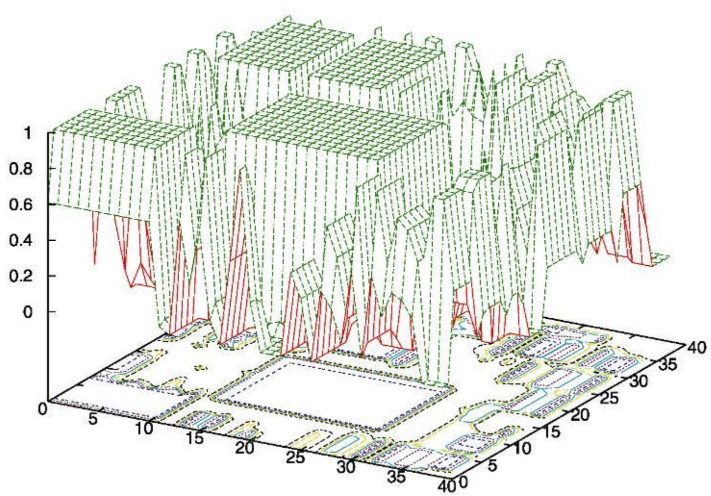
☹ the density map is not smooth

☹ some densities are too high to spread blocks over the mountains

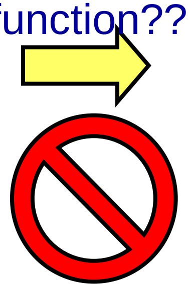
Bell-Shaped Block Smoothing??

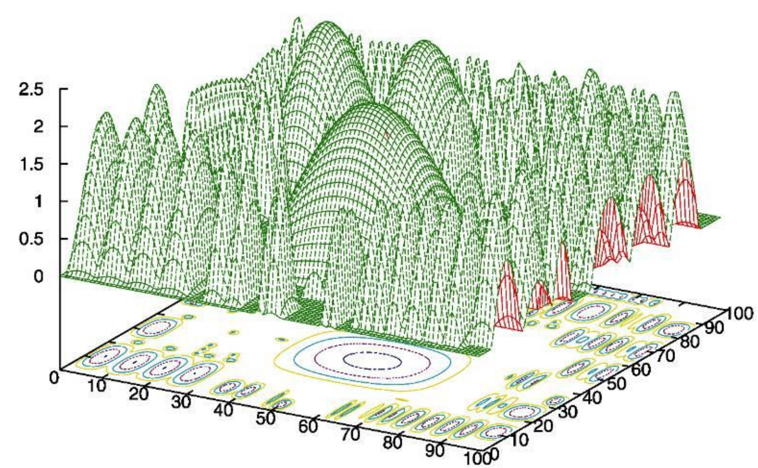


- ☹ mountain heights are quite different
- ☹ there are many valleys among mountains
- ➔ might mis-guide the movement of blocks



Apply the bell-shaped function??

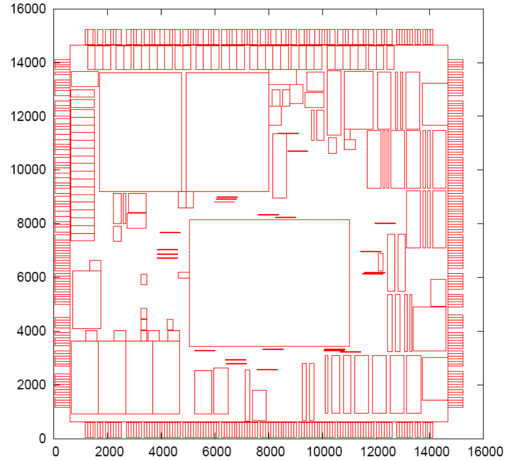


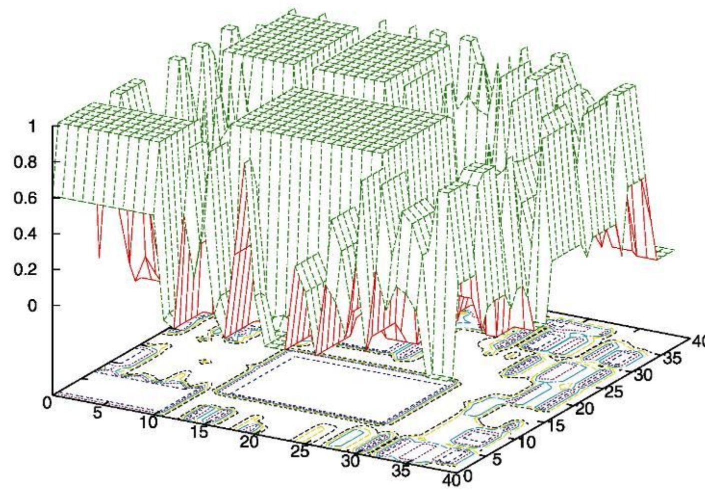


Bell-shaped smoothing

[Kahng & Wang, ICCAD-04]

Preplaced Block Smoothing

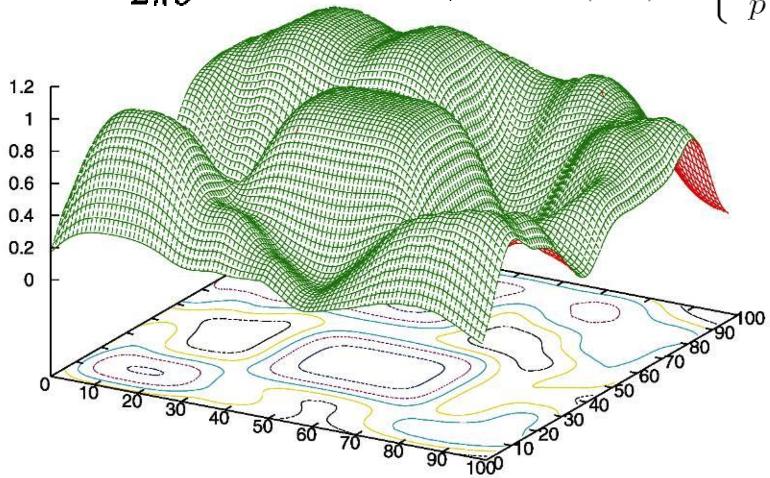




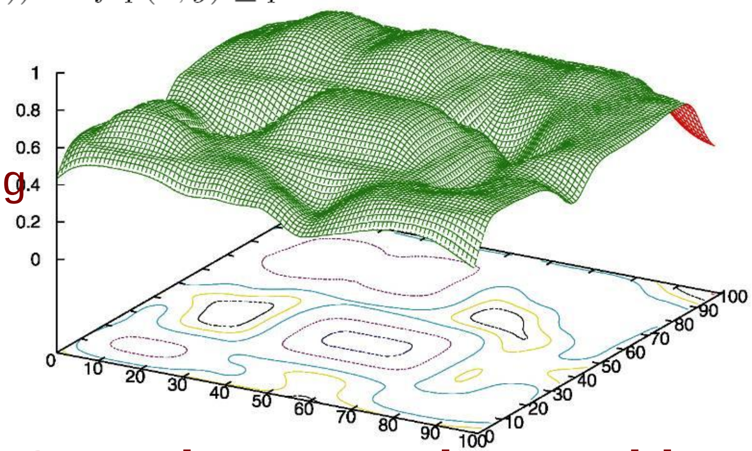
Convolute with Gaussian

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$p'(x,y) = \begin{cases} \overline{p} + (p(x,y) - \overline{p})^{\delta} & if\ p(x,y) \geq \overline{p} \\ \overline{p} - (\overline{p} - p(x,y))^{\delta} & if\ p(x,y) \leq \overline{p} \end{cases}$$



Level smoothing

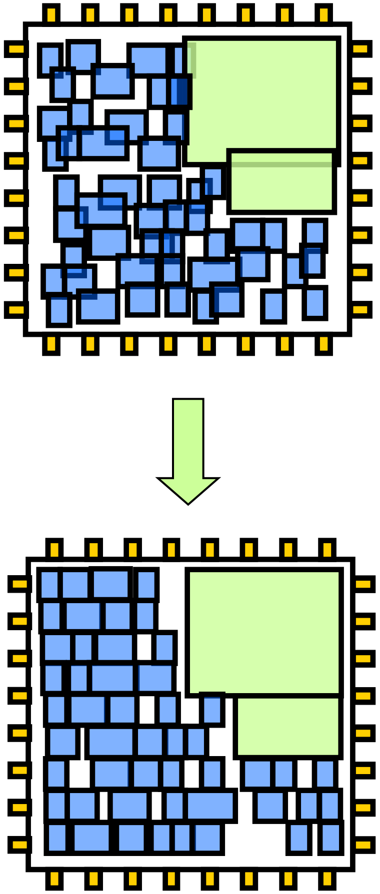


Gaussian smoothing

Gaussian + Level smoothing

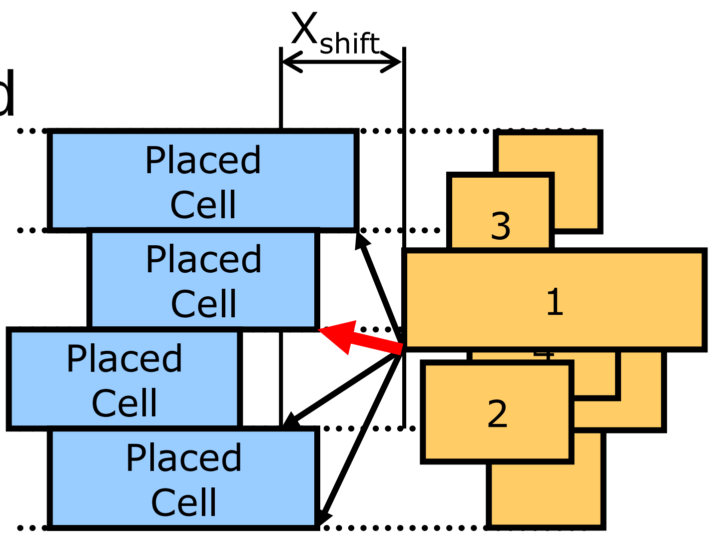
Legalization

- Placement flow
  - Global placement
  - ***Legalization***
    - ***Mixed-size legalization***
    - *Look-ahead legalization*
  - Detailed placement

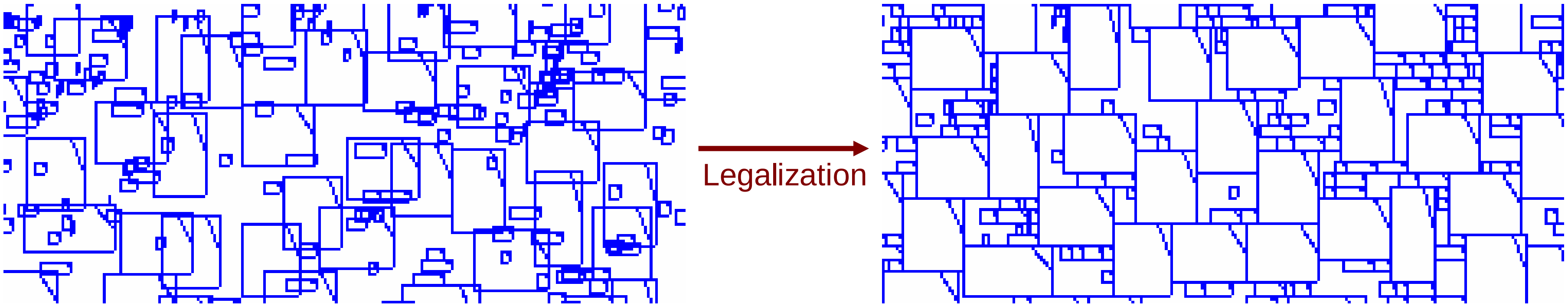


Mixed-Size Legalization

- Determine block legalization sequence by the x coordinate and block size
  - Priority = $k_1 x_i + k_2 w_i + k_3 h_i$
    - x_i: *x* coordinate of block *i*
    - $w_i(h_i)$: the width (height) of block *i*
  - Larger blocks are legalized earlier

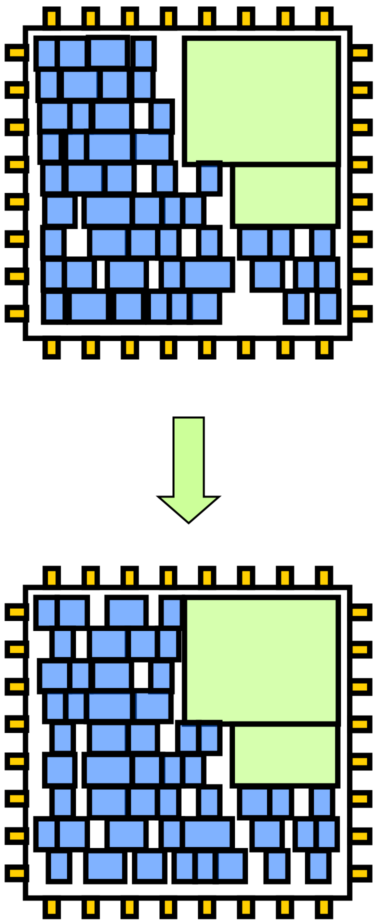


- Place block at the position with the smallest wirelength within a given range



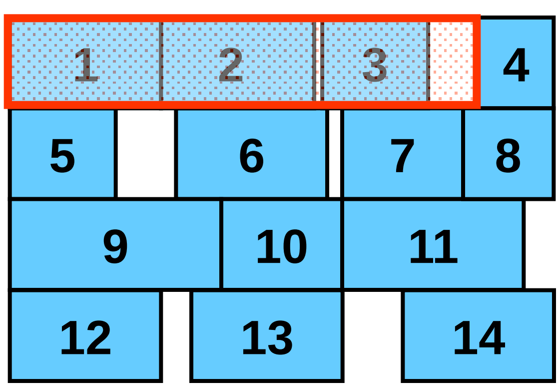
Detailed Placement

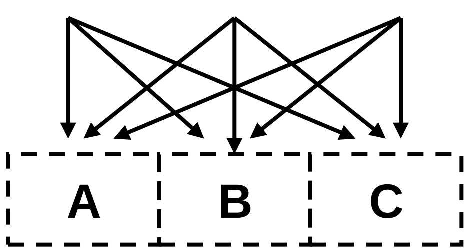
- Placement flow
  - Global placement
  - Legalization
  - ***Detailed placement***
    - ***Cell matching for wirelength minimization***
    - *Cell sliding for density optimization*



Cell Matching

- For wirelength minimization
- Steps
  1. Select a window
  2. Select blocks from the window
  3. Create a bipartite matching problem (edge weight = wirelength)
  4. Find the minimum weighted matching to optimize the wirelength
  5. Update block positions
- Handle **200-300** cells at one time
  - Compared to branch-and-bound which can handle only 6 cells at one time due to its high time complexity





Assign cells {1,2,3} to locations {A,B,C}

Demo: NTUplace3 (circuit: adaptec5)

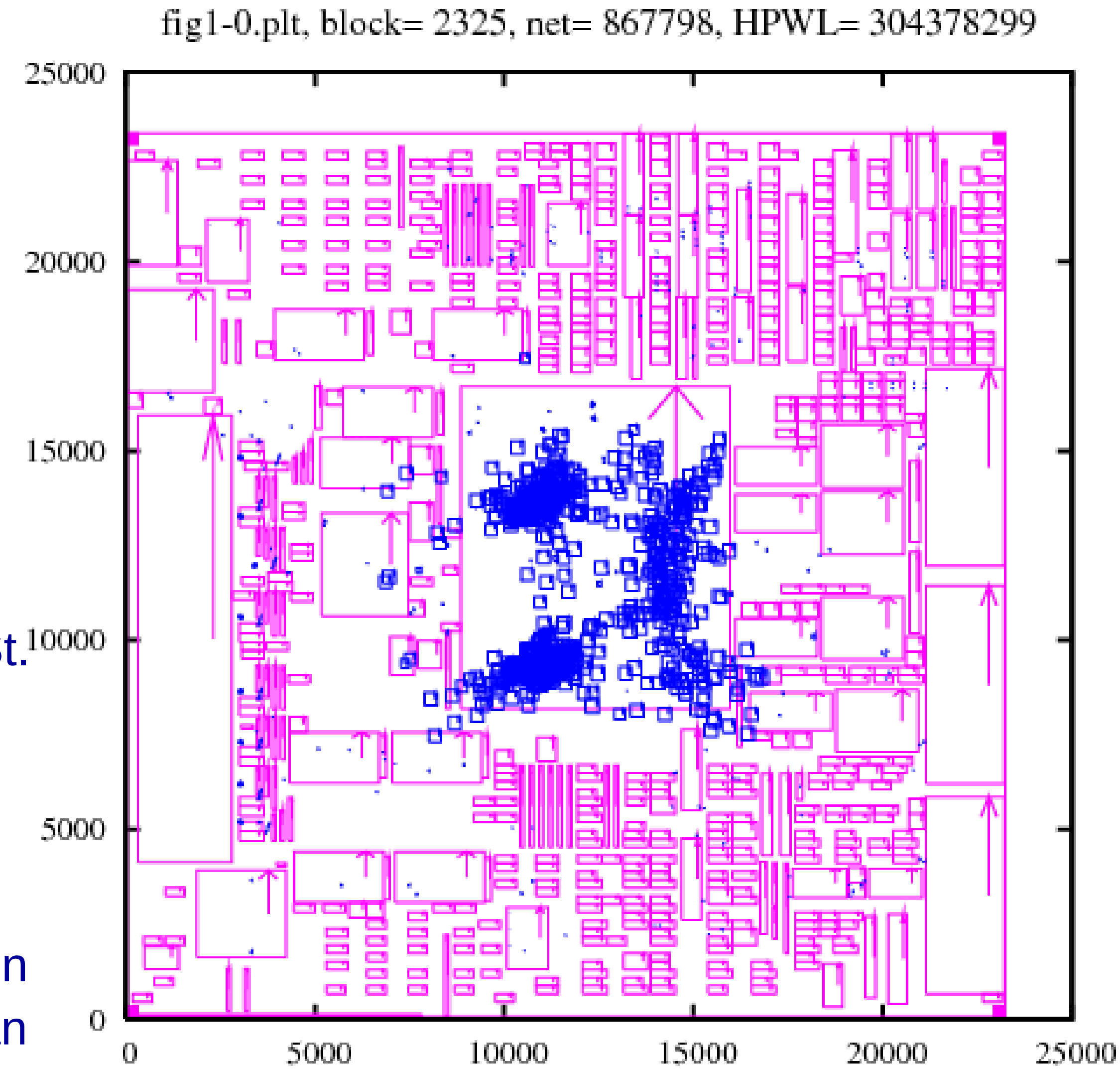
GP: 5 levels

#Movable obj.= 842k

#Fixed obj.= 646

#Nets = 868k

- ISPD-06 benchmark
  1. **NTUplace (0.99), NTU**
  2. Kraftwork (1.01), TU. Munich
  3. RQL (1.01), IBM/Iowa St.
  4. mPL6 (1.04), UCLA
  5. mFAR (1.11), UCSB
  6. APlace (1.16), UCSD
  7. Dragon (1.23), UCLA
  8. DPlace (1.36), UT-Austin
  9. Capo (1.39), U. Michigan



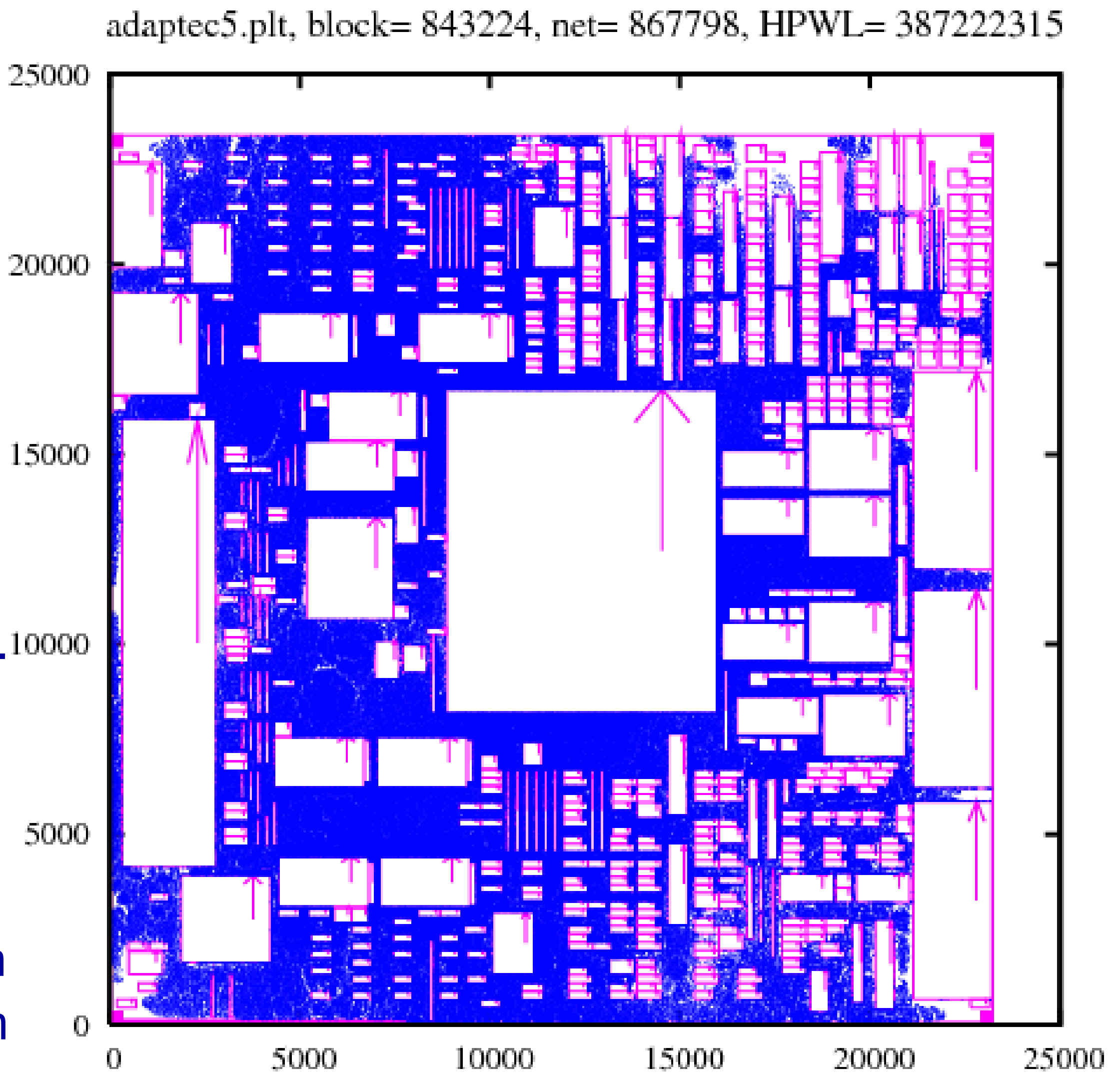
Demo: NTUplace3 (circuit: adaptec5)

GP: 5 levels

#Movable obj.= 842k

#Fixed obj.= 646

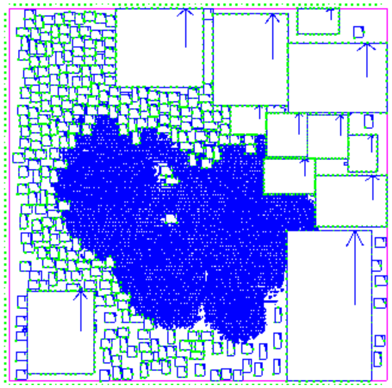
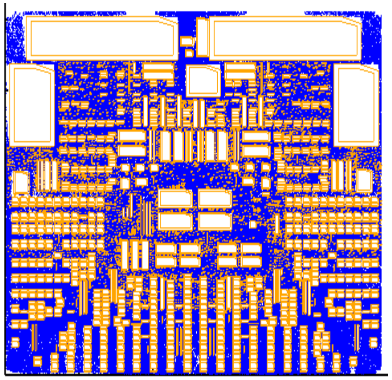
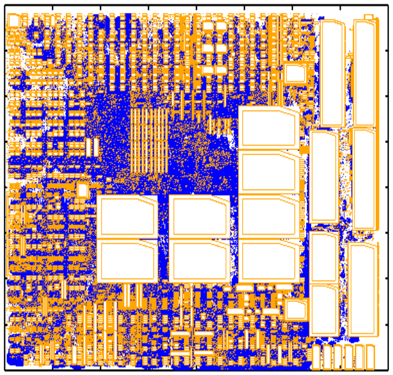
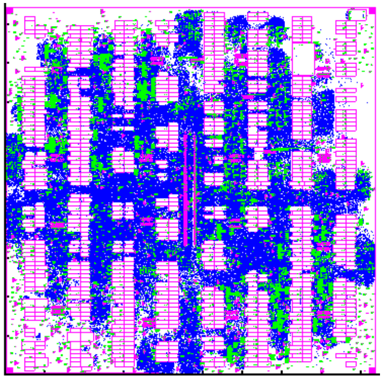
#Nets = 868k



- ISPD-06 benchmark
  1. **NTUplace (0.99), NTU**
  2. Kraftwork (1.01), TU. Munich
  3. RQL (1.01), IBM/Iowa St.
  4. mPL6 (1.04), UCLA
  5. mFAR (1.11), UCSB
  6. APlace (1.16), UCSD
  7. Dragon (1.23), UCLA
  8. DPlace (1.36), UT-Austin
  9. Capo (1.39), U. Michigan

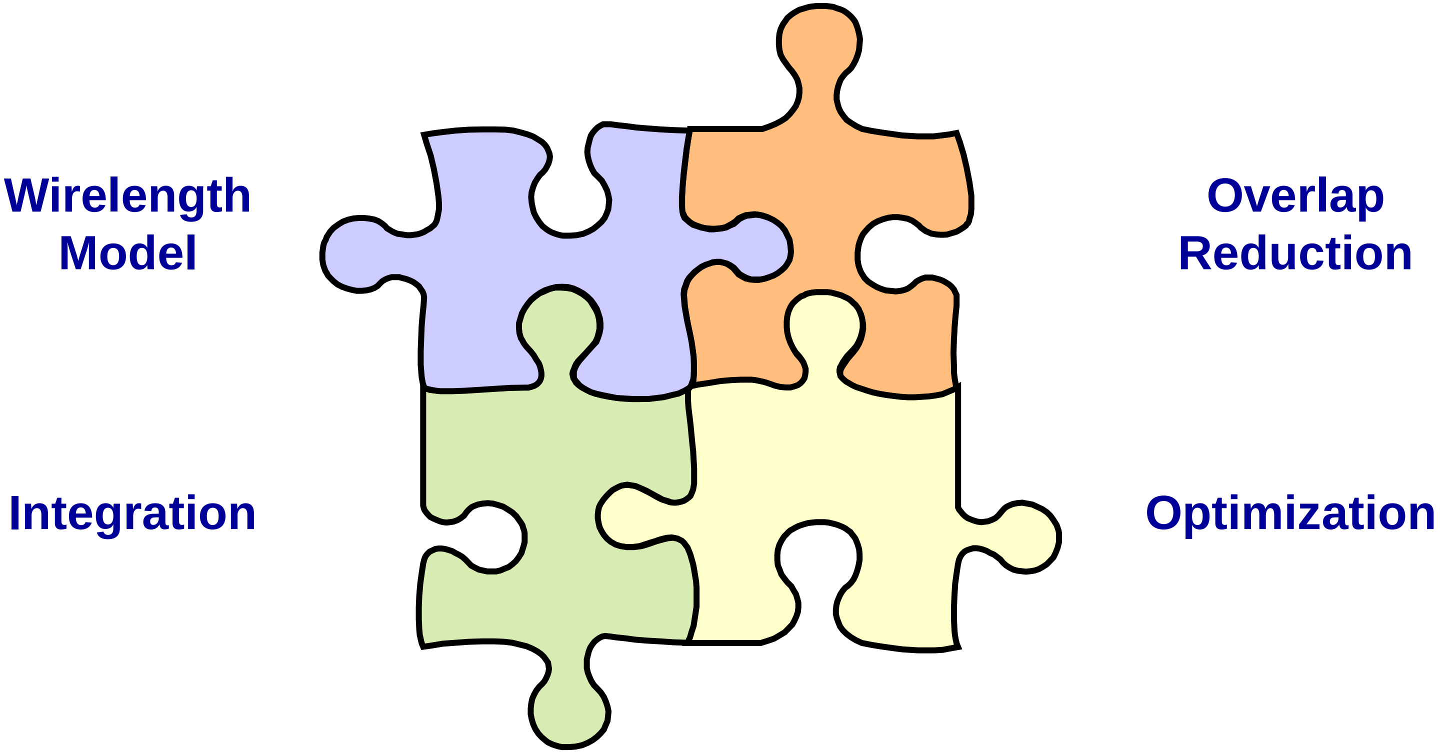
Placement Benchmarks

- Three state-of-the-art benchmark suites
  - **ICCAD-04** IBM mixed-size (18 circuits): # Mov: 12K to 210K; Utilization: 80%
  - **ISPD-05** placement contest (8 circuits): # Mov: 211K to 2.2M; # Fix: 543 to 23K; Utilization: 27% to 57%
  - **ISPD-06** placement contest (8 circuits): # Mov: 330K to **2.5M;** # Fix: 336 to 27K; Utilization: 26% to 71%; **Target density**: 50% to 90%
- NTUplace3 obtains best results for the three suites with both the WA and LSE wirelength models

Essential Issues in Analytical Placement

- Chang, Jiang & Chen, "Essential issues in analytical placement algorithms," IPSJ Trans. System LSI Design Methodology, August 2009



NTUplace3 Example

- **Wirelength Model:** **Weighted-wirelenth (WA) or log-sum-exp (LSE) function**
- **Overlap Reduction:** **Bell-shaped density model + Gaussian & Level smoothing**
- **Integration:** **Quadratic penalty method**
- **Optimization:** **Nonlinear**

Modern Academic Analytical Placers

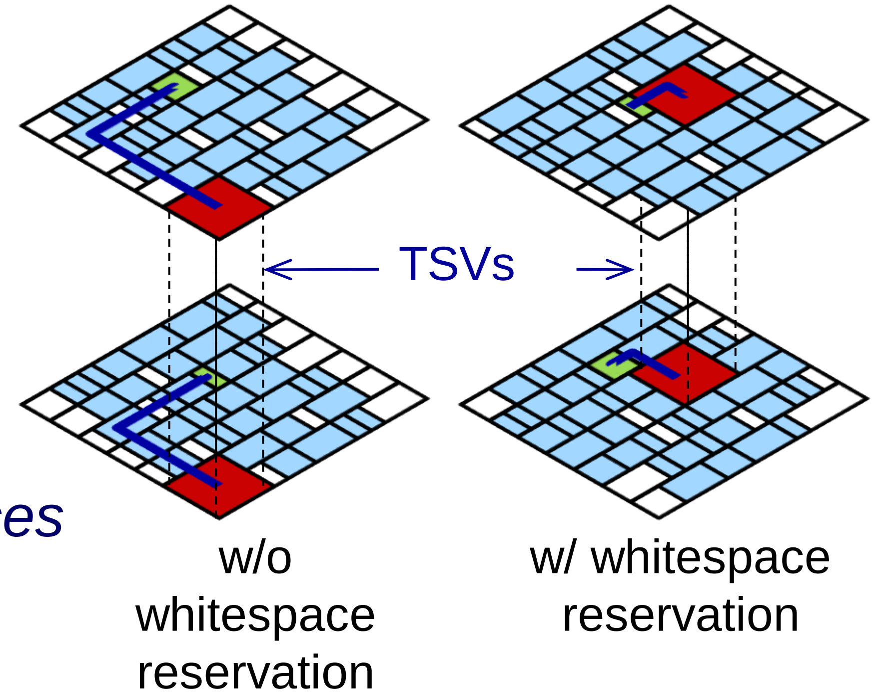
Placer	Wirelength Model	Overlap Reduction	Integration	Optimization
APlace	LSE	Density	Penalty Method	Nonlinear
BonnPlace	Quadratic	Partitioning	Region Constraint	Quadratic
DPlace	Quadratic	Diffusion	Fixed Point	Quadratic
FastPlace	Quadratic	Cell Shifting	Fixed Point	Quadratic
FDP	Quadratic	Density	Fixed Point	Quadratic
Gordian	Quadratic	Partitioning	Region Constraint	Quadratic
hATP	Quadratic	Partitioning	Region Constraint	Quadratic
Kraftwerk2	Bound2Bound	Density	Fixed Point	Quadratic
mFAR	Quadratic	Density	Fixed Point	Quadratic
mPL6	LSE	Density	Penalty Method	Nonlinear
NTUplace3	LSE	Density	Penalty Method	Nonlinear
RQL	Quadratic	Cell Shifting	Fixed Point	Quadratic
Vassu	LSE	Assignment	Fixed Point	Nonlinear

Other Combinations for New Placers?

Quadratic Wirelength function	WA/LSE function	???
Partitioning	Bell-shaped density model	???
Region constraint	Penalty method	???
Quadratic	Nonlinear	???

3D IC Placement with TSVs

- Through-silicon vias (TSVs) cause significant challenges for 3D IC placement
- TSVs
  - Connect signals between device layers in a 3D IC
  - Are usually placed at the *whitespace* among cells
  - Affect the *routing resources* and increase the overall chip or package *areas*



- Need to reserve whitespace for TSV insertion

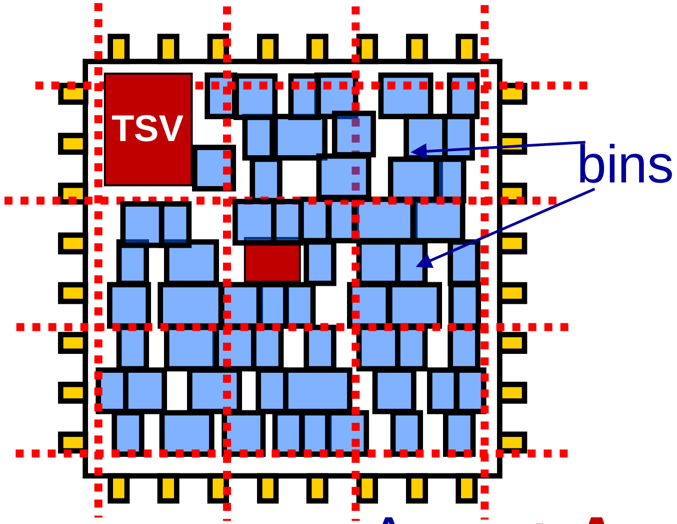
3D IC Placement Problem

- Given 3D IC layers and block dimensions, divide the placement region into bins
- Determine the layers and positions for all blocks

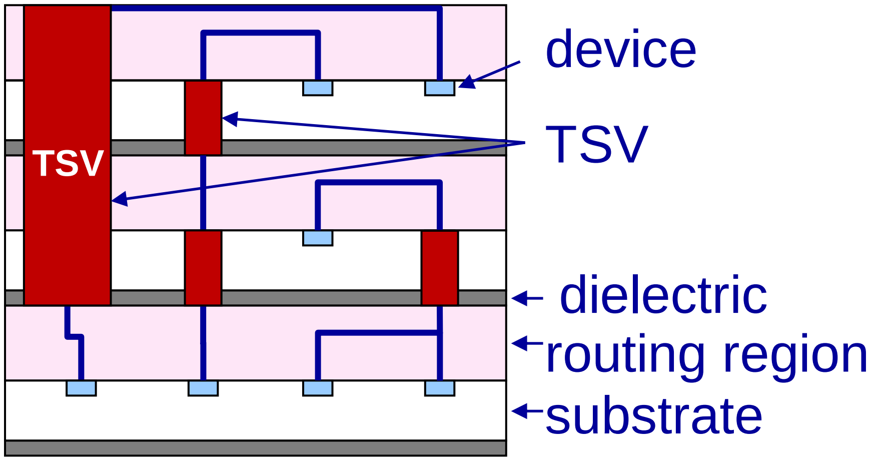
min wirelength **& TSV counts**

s.t. 1. $\text{Density}_b \leq \text{MaximumDensity}_b$ for each bin *b*

2. No overlap between blocks



$$\text{Density} = \frac{A_{block} + A_{TSV}}{A_{bin}}$$

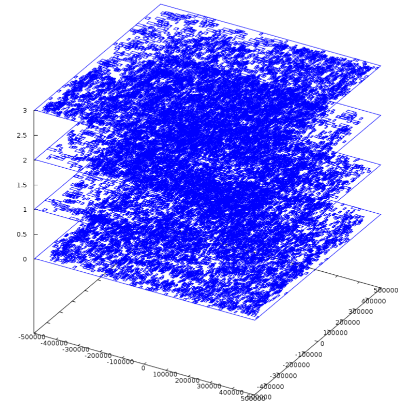


TSV-Aware 3D Analytical Placement Flow

Netlist

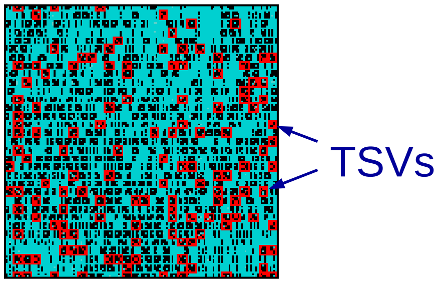
↓

3D Global Placement *based on the weighted-average (WA) wirelength model* with whitespace reservation for TSVs



↓

TSV Insertion and TSV-aware Legalization



↓

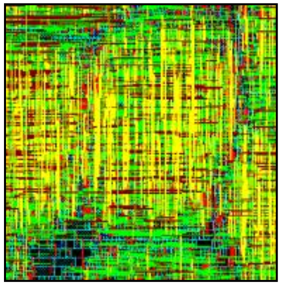
Layer-by-layer Detailed Placement

↓

3D Placed Results with Inserted TSVs

↓

Layer-by-layer Routing



3D Analytical Global Placement

- Analytical formulation

  min $\lambda_1 W(x,y) + \lambda_2 V(z)$ // minimize wirelength and TSV counts

  s.t. $(D_{b,k}(x,y,z) + T_{b,k}(x,y,z)) \le M_{b,k}, \quad 1 \le k \le K$

  K: number of layers

  $D_{b,k}$: block density function for bin *b* on layer *k*

  $T_{b,k}$: TSV density for bin *b* on layer *k*

  $M_{b,k}$: max density for bin *b* on layer *k*

- Relax the constraints into the objective function

  min $\lambda_1 W(x,y) + \lambda_2 V(z)$

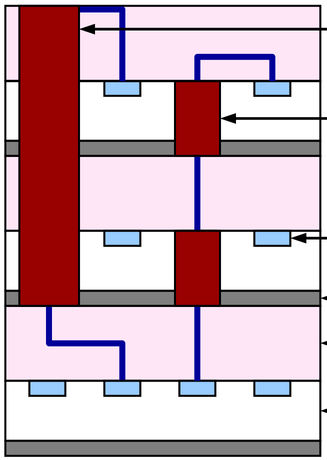
  $+ \lambda_3 (\Sigma((D_{b,k}(x,y,z) + T_{b,k}(x,y,z)) - M_{b,k})^2)$

  – Use the gradient method to solve it

  – Increase λ_3 gradually to find the desired (x,y,z)

TSV Counts

- Two types of TSVs
  - Via-first TSVs interfere with device layer only
  - Via-last TSVs interfere with both device and metal layers



- For both types of TSVs, the number of TSVs used for each net can be defined as

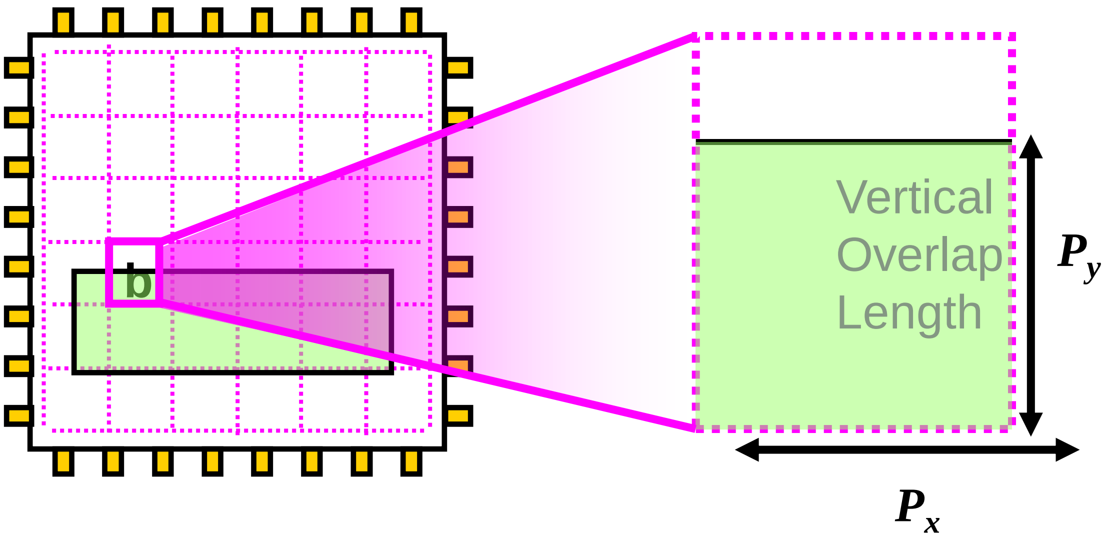
$$V(\mathbf{z}) = \sum_{e \in E} (\max_{v_i, v_j \in e} |z_i - z_j|)$$

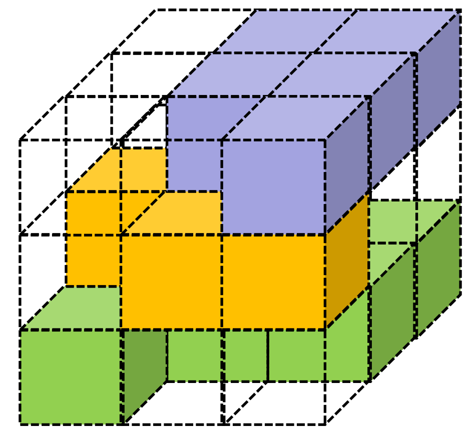
- WA approximation

$$V_{WA}(\mathbf{z}) = \sum_{e \in E} \left(\frac{\sum_{v_i \in e} z_i \exp(z_i / \gamma)}{\sum_{v_i \in e} \exp(z_i / \gamma)} - \frac{\sum_{v_i \in e} z_i \exp(-z_i / \gamma)}{\sum_{v_i \in e} \exp(-z_i / \gamma)} \right)$$

Cube Density Function

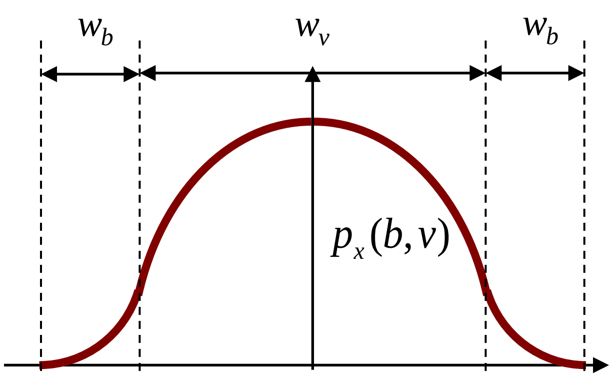
- Compute the block volume in each cube to obtain the density function for cube *b* on layer *k*





Horizontal Overlap Length

$$D_{b,k}(\mathbf{x},\mathbf{y},\mathbf{z}) = \sum_{v\in V} P_x(b,v,k)P_y(b,v,k)P_z(b,v,k)$$

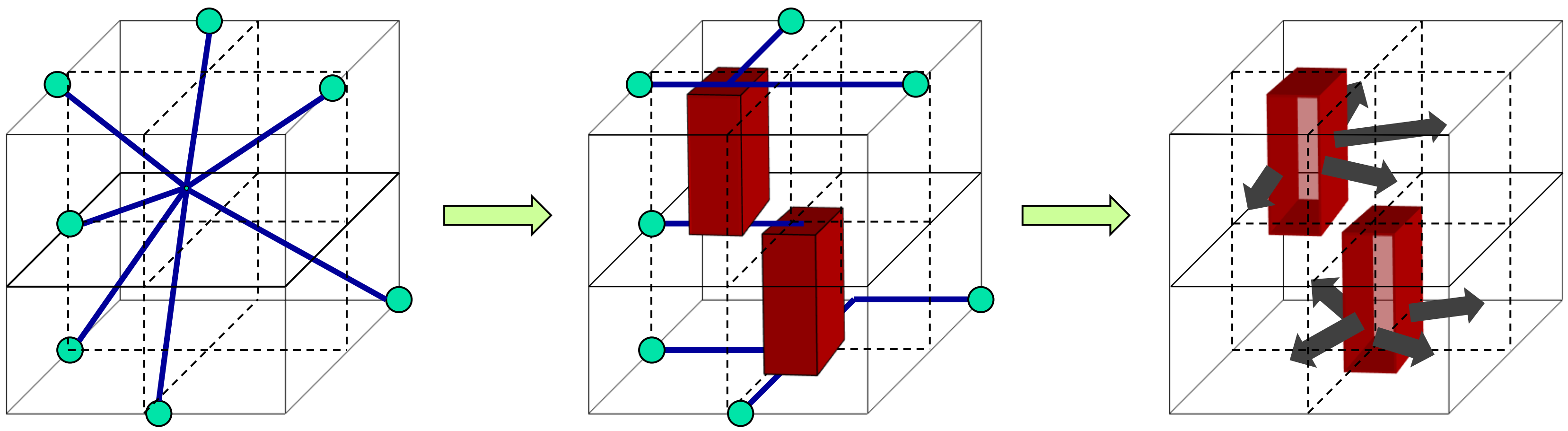


bell-shaped smoothing

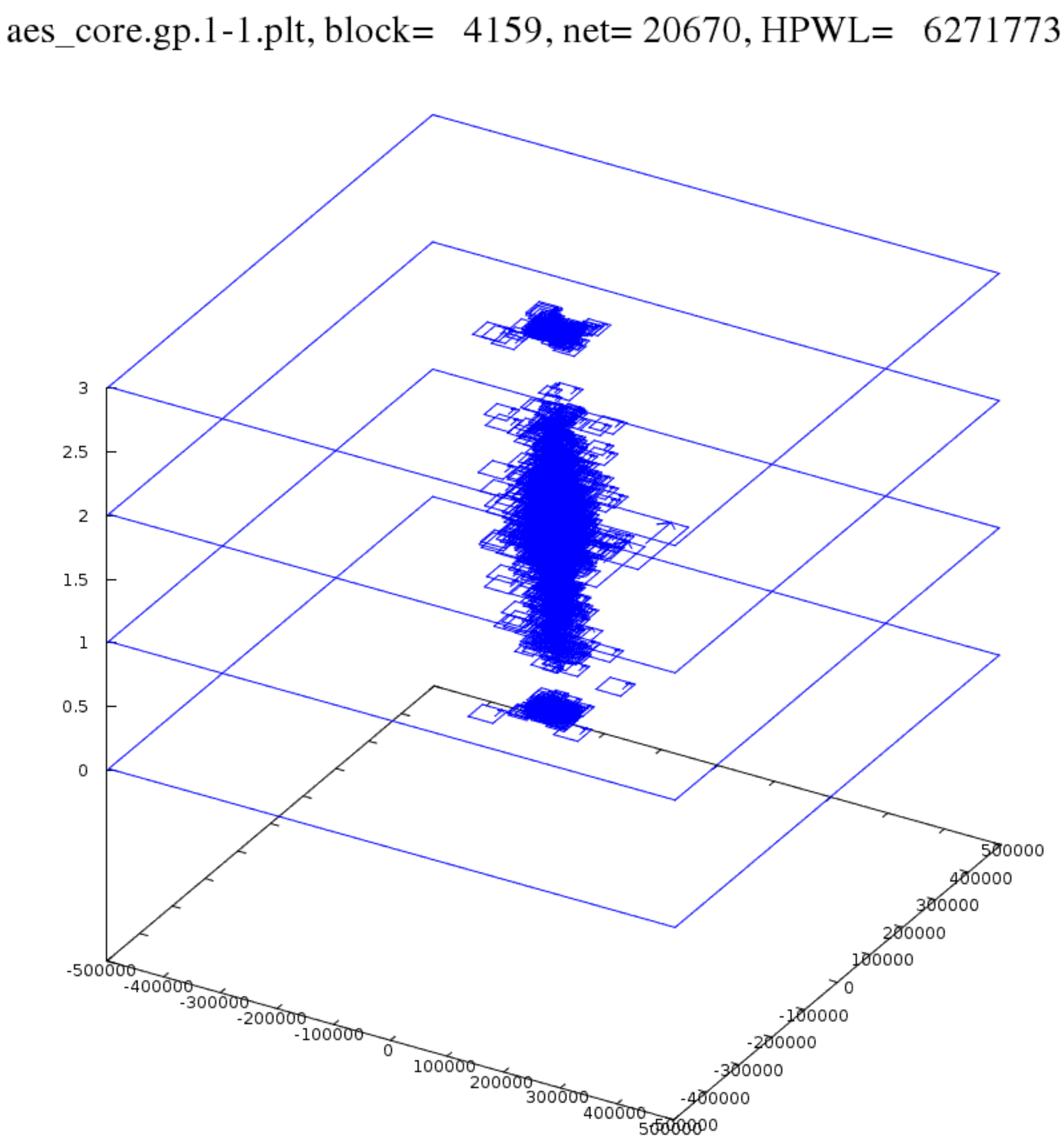
[Kahng & Wang, ICCAD-04]

Whitespace Reservation for TSVs

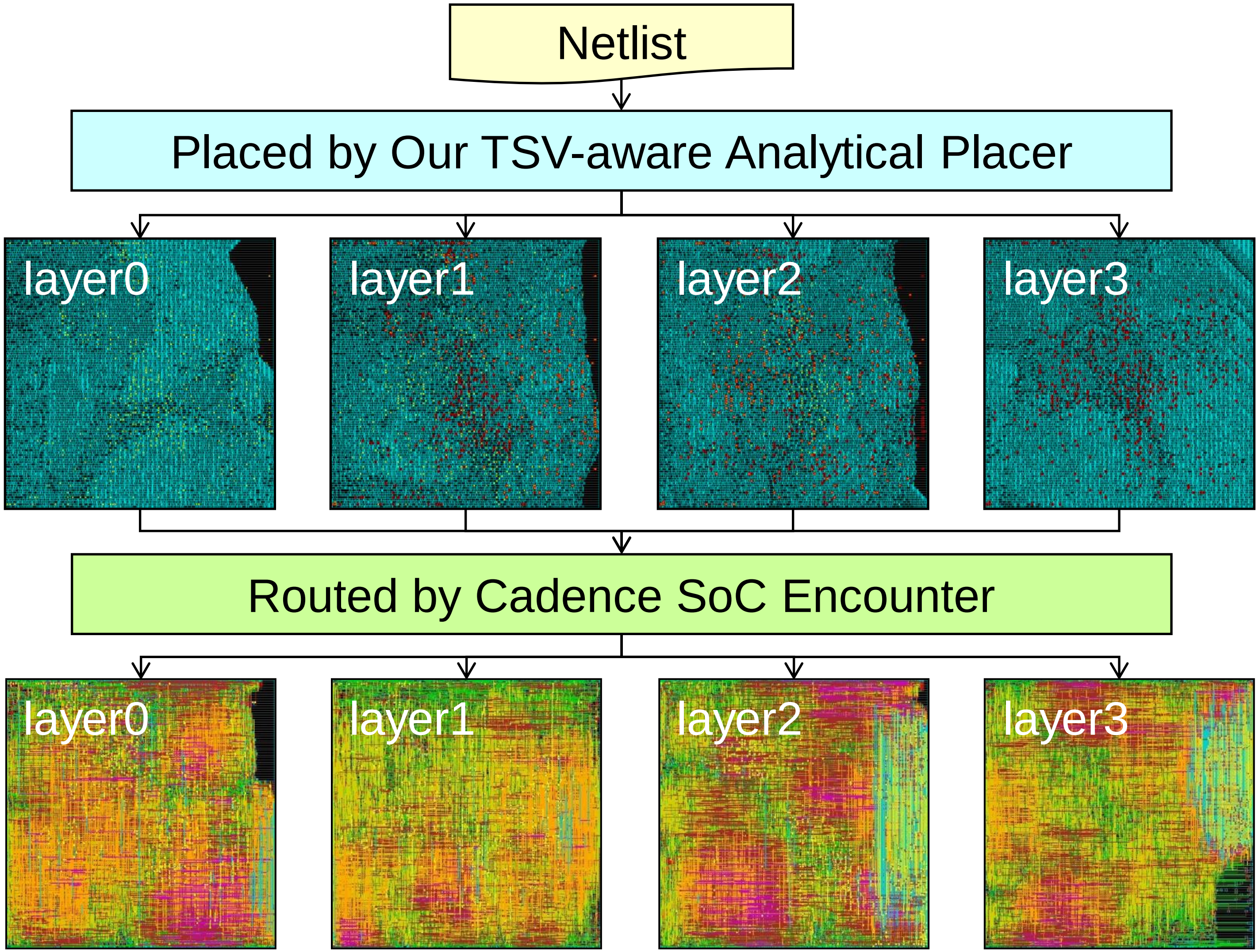
- Reserve whitespace in the bounding cube of a net for TSVs
- $D_{b,\,k}(x, y, z) + T_{b,k}(x, y, z) \leq M_{b,k}$
  - Block density $D_{b,\,k}$
  - TSV density $T_{b,k}$



Demo: 3D Analytical Placement



3D IC Placement and Routing Flow



3D IC Placement Comparisons

- Compared with [Cong & Luo, ASPDAC-09], our 3D placer can reduce the HPWL by 13% and TSV counts by 16%, with a 12X speedup

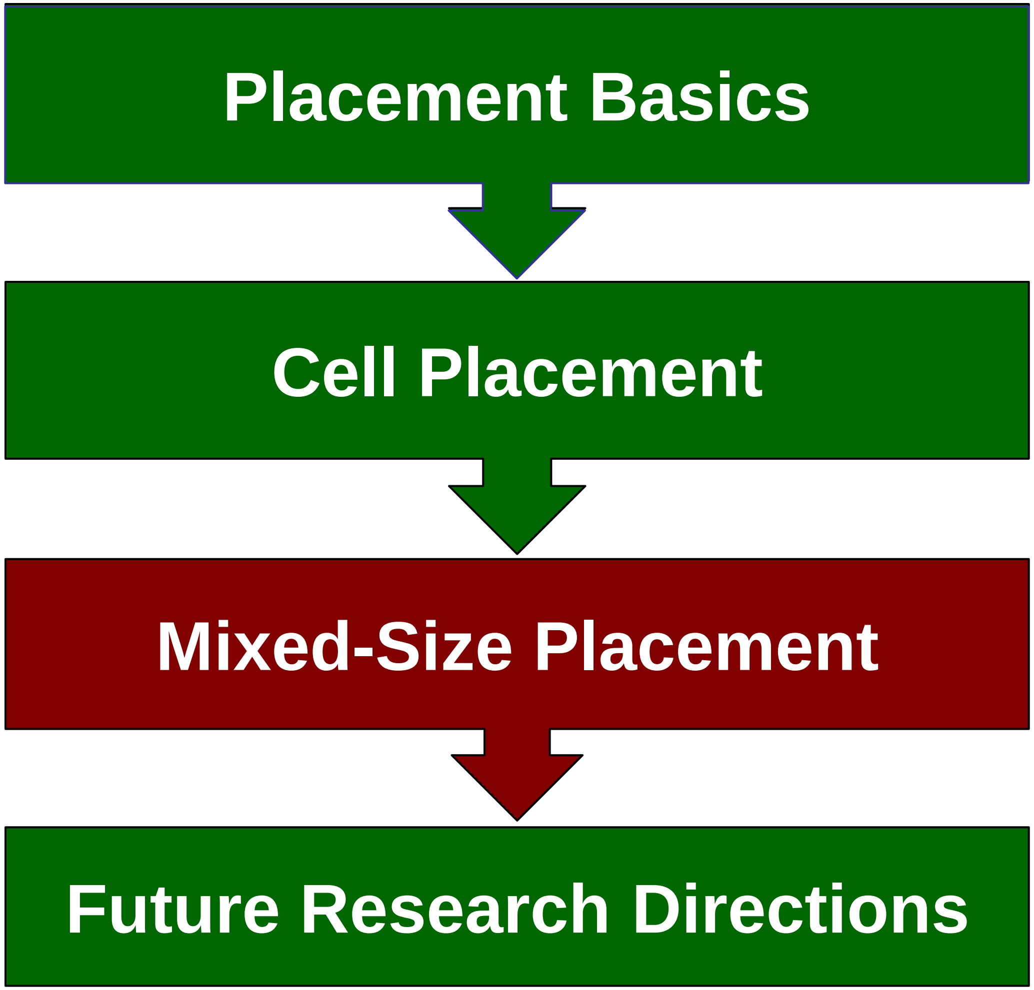
	HPWL	#TSV	Time
[Cong, ASPDAC-09]	1.00	1.00	1.00
Our placer	**0.87**	**0.84**	**0.08**

- Compared with the state-of-the-art 3D placer [Kim et al., ICCAD-09], our placer achieves 10% shorter routed wirelength, 21% fewer TSV counts, and 18% smaller total silicon area, with a 2.6X speedup

	Routed Wirelength	#TSV	Silicon Area	Time
[Kim et al., ICCAD'09]	1.00	1.00	1.00	1.00
Our placer **w/o** WR	0.93	0.80	0.83	0.38
Our placer **w/** WR	**0.90**	**0.79**	**0.82**	**0.38**

WR: whitespace reservation

Outline



Placement Basics
Cell Placement
Mixed-Size Placement
Future Research Directions

Methods on Mixed-Size Placement

Type 1: Constructive approach

- Combine floorplanning and placement
- Examples: Capo, PATOMA, FLOP

Type 2: Two-stage approach

- Perform (1) macro placement and then (2) cell placement
- Examples: MP-tree, CG

Type 3: One-stage approach

- Place macro and cell simultaneously
- Examples: mPG-MS, APlace, mPL, UPlace, NTUplace3, etc.

Type 1: Constructive Approach

Type 1: Constructive approach

- Combine floorplanning and placement
- Examples: Capo, PATOMA, FLOP

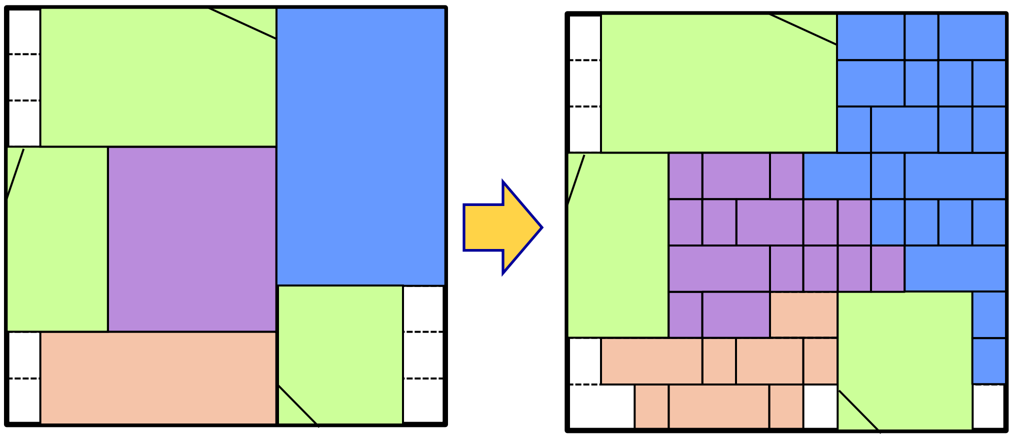
Type 2: Two-stage approach

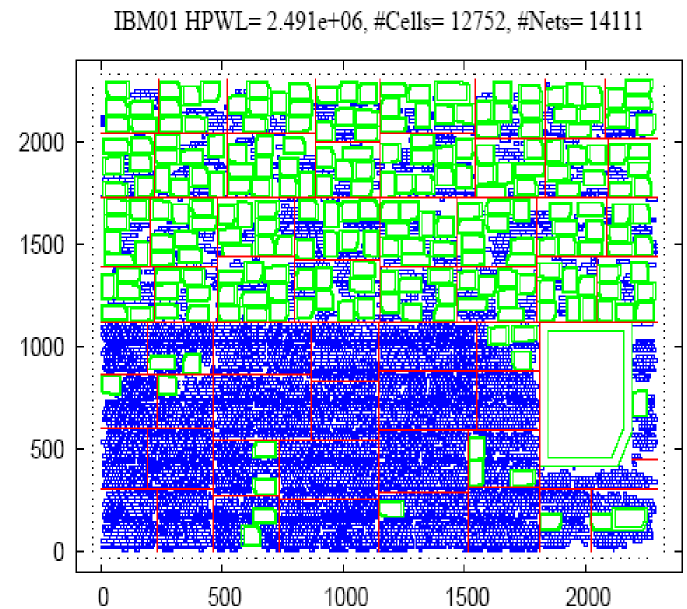
- Perform (1) macro placement and then (2) cell placement
- Examples: MP-tree, CG

Type 3: One-stage approach

- Place macros and cells simultaneously
- Examples: mPG-MS, APlace, mPL, UPlace, NTUplace3, etc.

Type 1: Constructive Approach





Capo placement

- Combine floorplanning and placement
  - Capo [ICCAD'04], PATOMA [ASPDAC'05], FLOP [DAC'09]
  - Apply recursive min-cut bi-partitioning
- Keep macros overlap-free during placement
- The solution quality is often limited

Type 2: Two-Stage Approach

Type 1: Constructive approach

- Combine floorplanning and placement
- Examples: Capo, PATOMA, FLOP

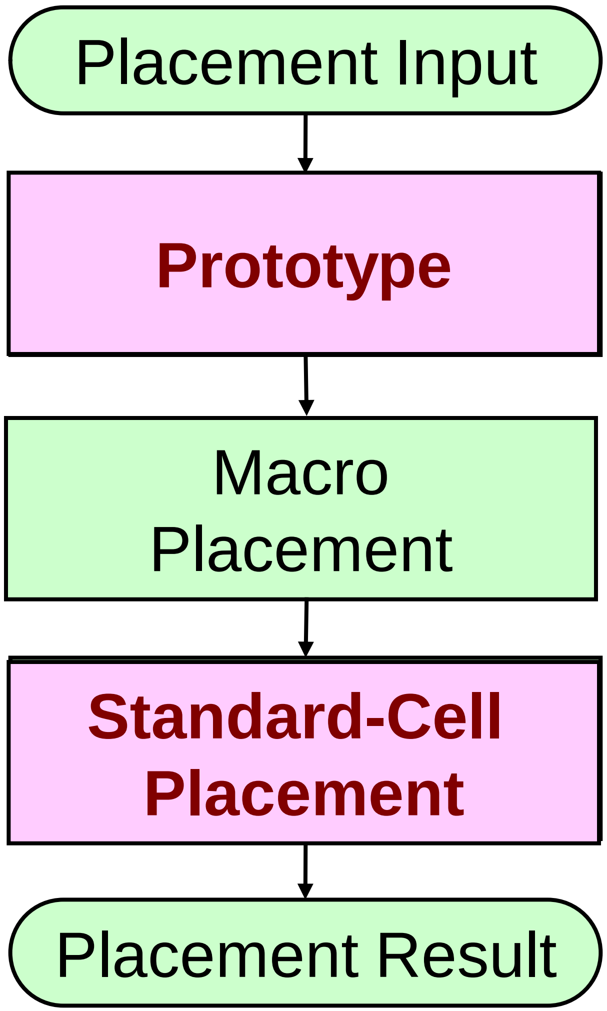
Type 2: Two-stage approach

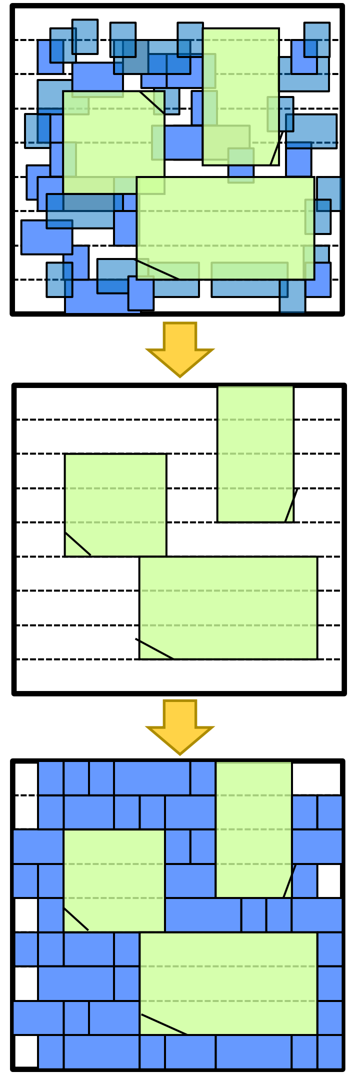
- Perform (1) macro placement and then (2) cell placement
- Examples: MP-tree, CG

Type 3: One-stage approach

- Place macros and cells simultaneously
- Examples: mPG-MS, APlace, mPL, UPlace, NTUplace3, etc.

Two-Stage Approach





wirelength optimization

NTUplace3

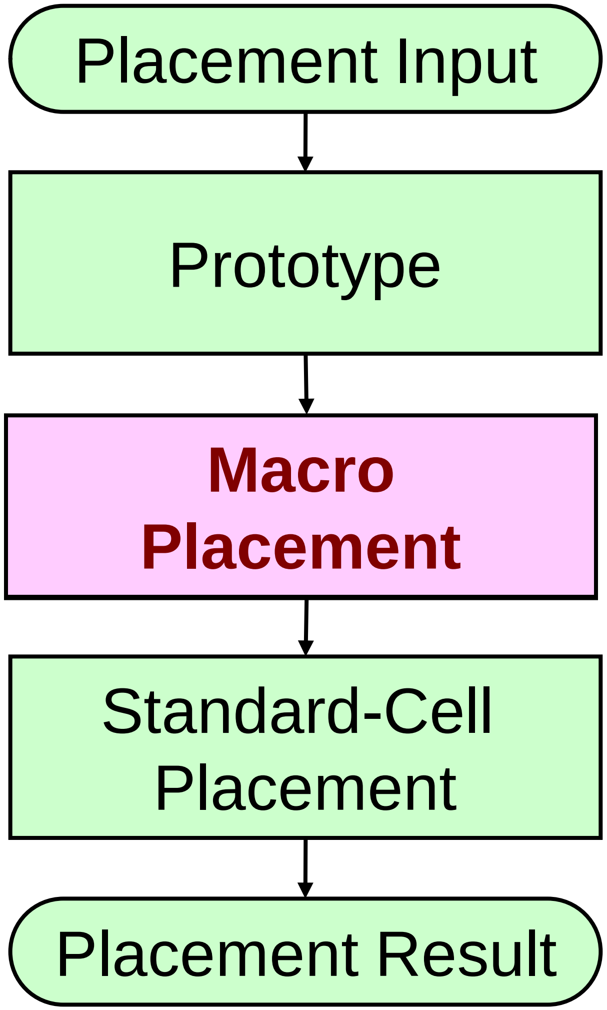
macro legalization/rotation (displacement minimization, orientation optimization, congestion optimization, etc.)

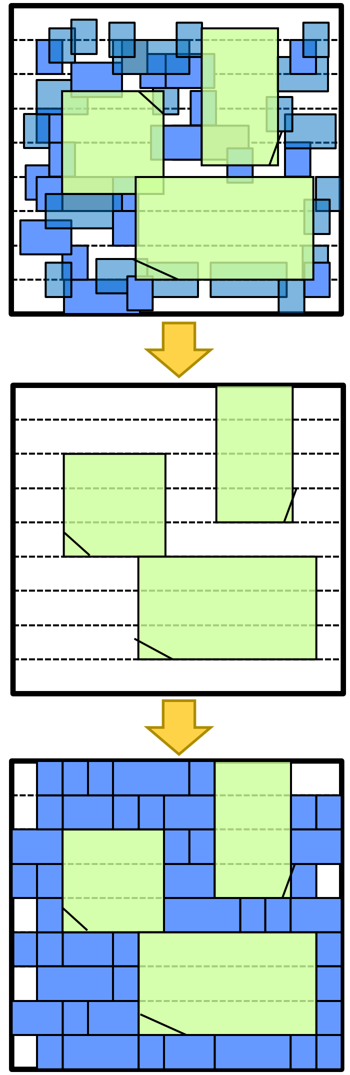
MP-tree

wirelength optimization, congestion optimization

NTUplace3

Two-Stage Approach





wirelength optimization

NTUplace3

macro legalization/rotation (displacement minimization, orientation optimization, congestion optimization, etc.)

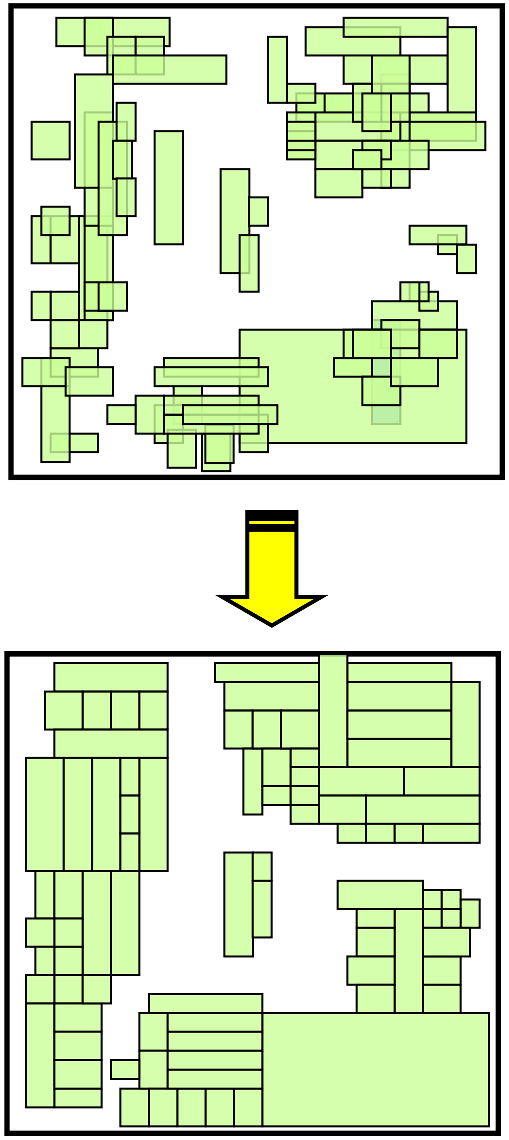
MP-tree

wirelength optimization, congestion optimization

NTUplace3

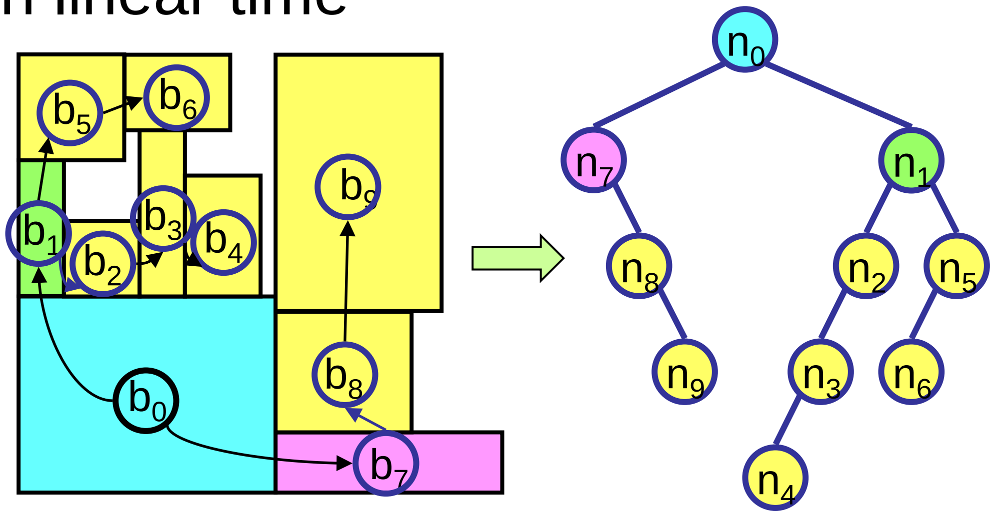
Macro Placement

- Input
  - An initial placement that considers both macros and standard cells and optimizes a simplified cost metric (e.g., wirelength)
- Objectives
  - Remove all overlaps between macros
  - Minimize macro movement (displacement)
- Popular approaches
  - Packing-based method: MP-tree [DAC'07, TCAD'08]
  - Constraint graph-based method: CG [ICCAD'08]

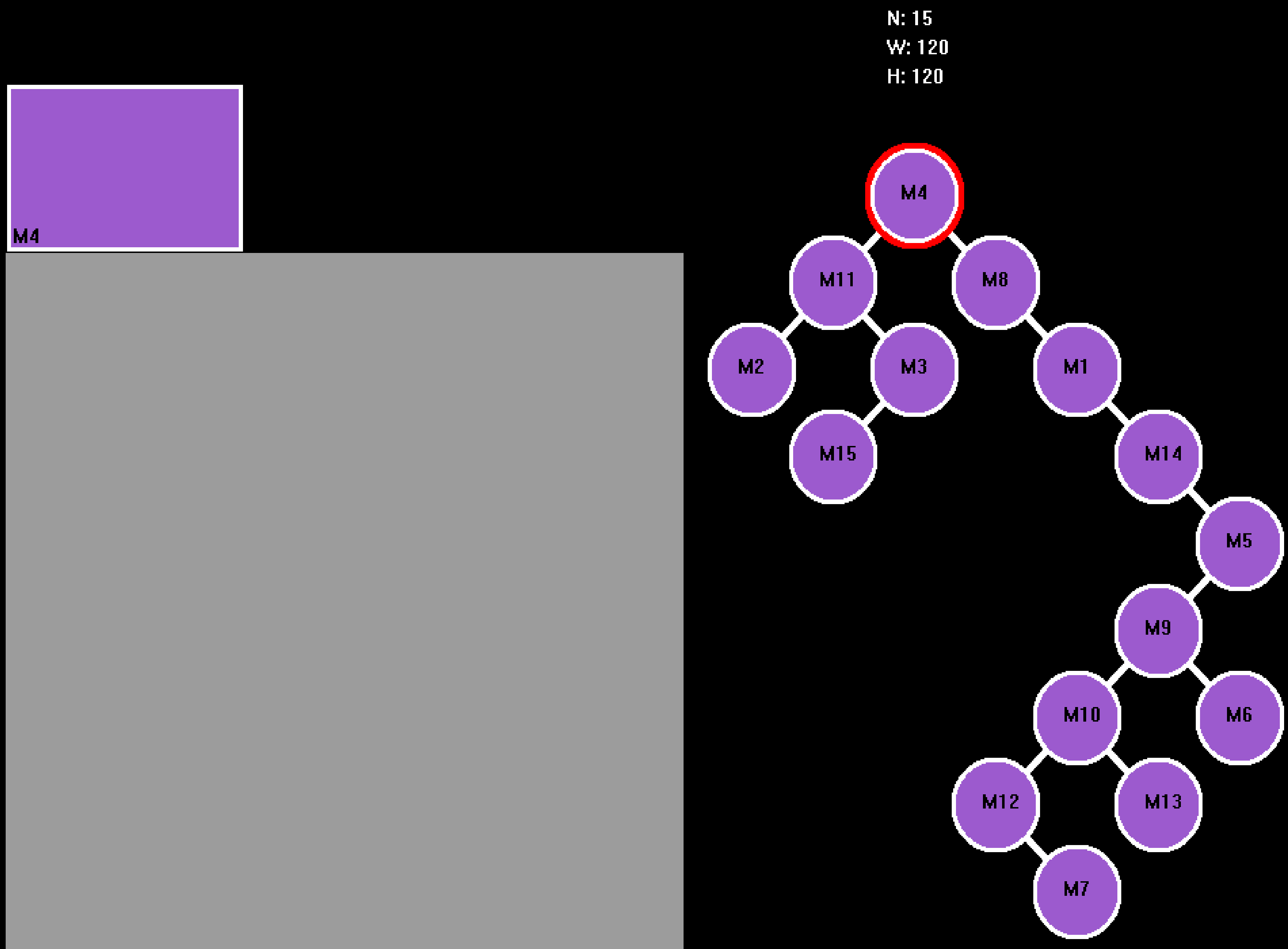


Macro Placement Using Multiple B*-Trees

- Construct an ordered binary tree (B*-tree) [Chang et al., DAC-2K]
  - Left child: the lowest, adjacent macro on the right ($x_j = x_i + w_i$)
  - Right child: the first macro above, with the same *x*-coordinate ($x_j = x_i$)
- Convert between a compacted placement and a B*-tree in linear time



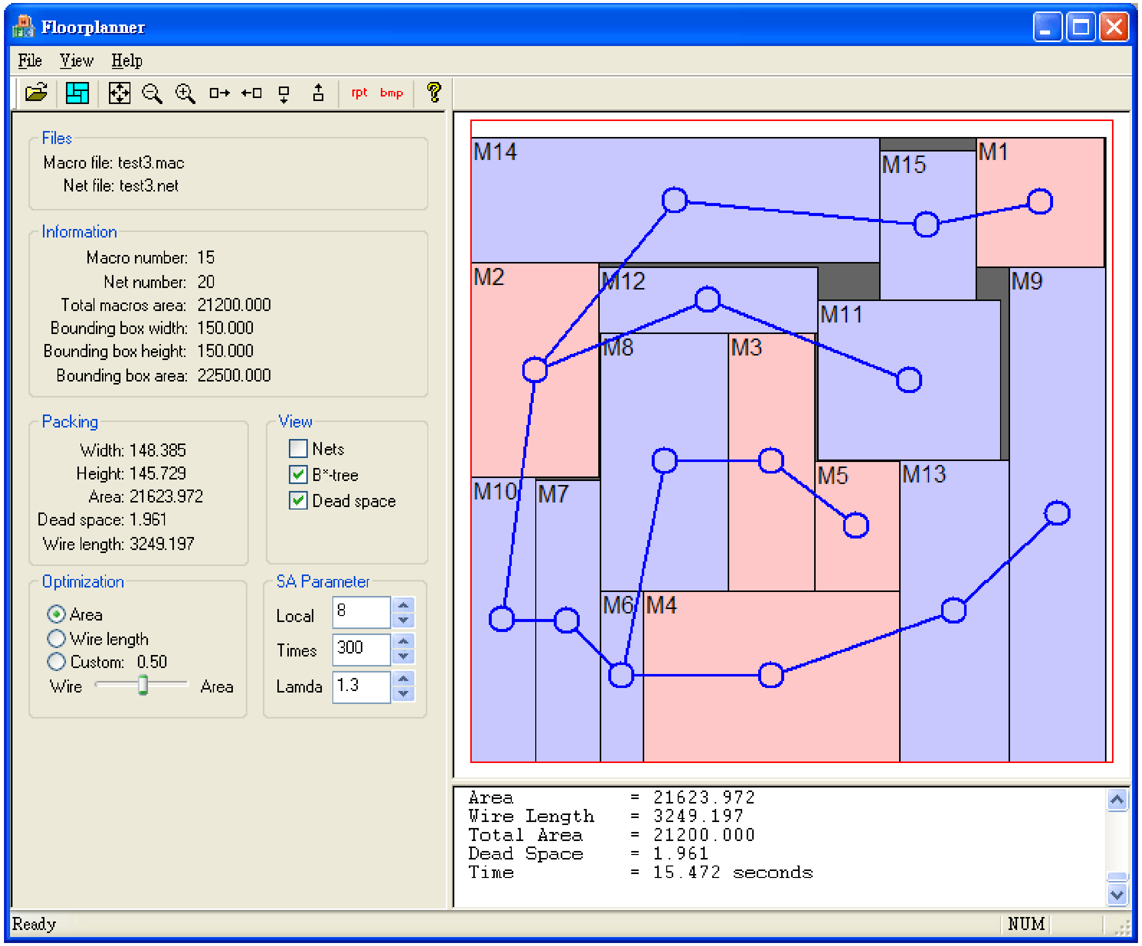
Compact to left and bottom   B*-tree


N: 15
W: 120
H: 120
M4
M4
M11
M8
M2
M3
M1
M15
M14
M5
M9
M10
M6
M12
M13
M7

B*-tree Based Placer/Floorplanner

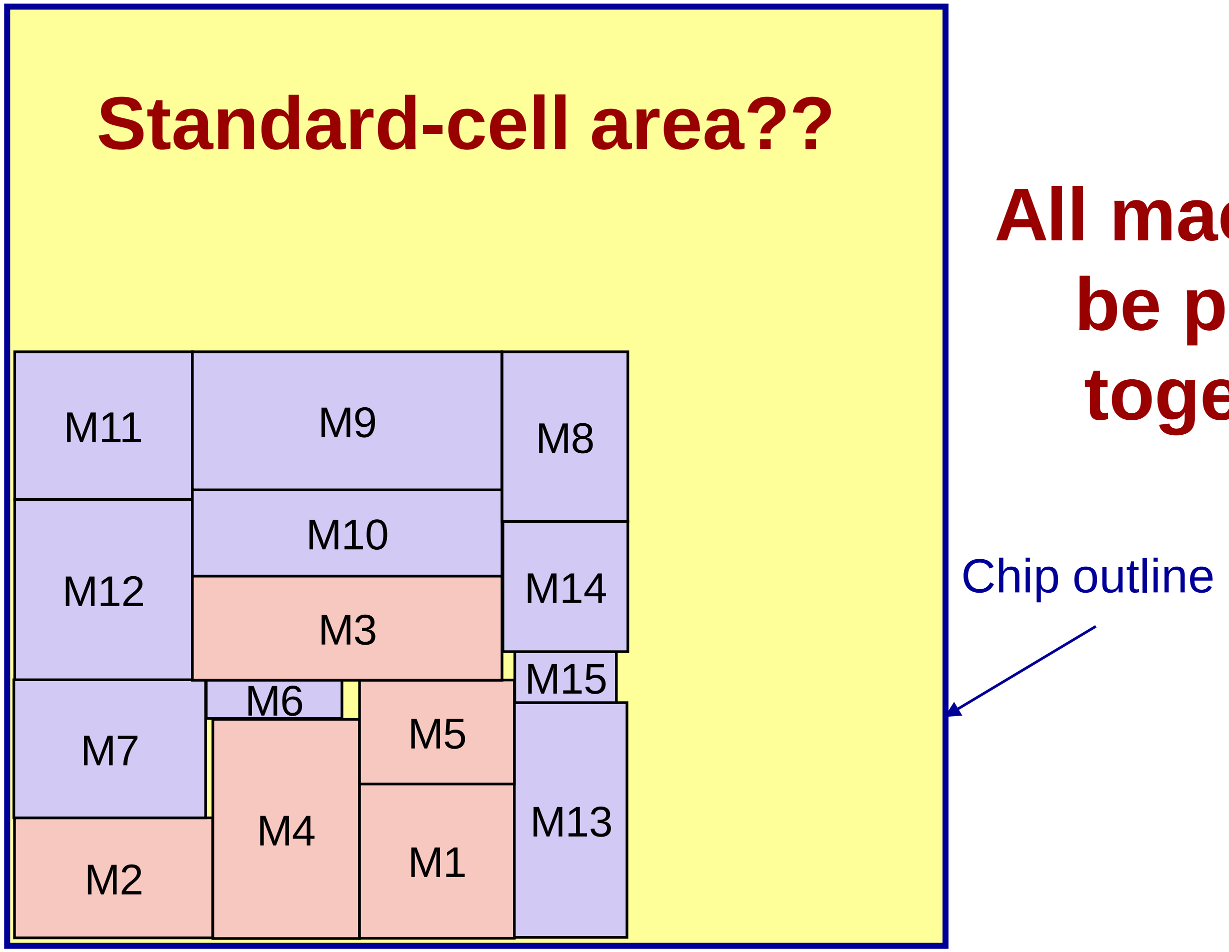
- Rated the best representation for packing in [Chan, et. al, ISPD-05]

http://eda.ee.ntu.edu.tw/research.htm/



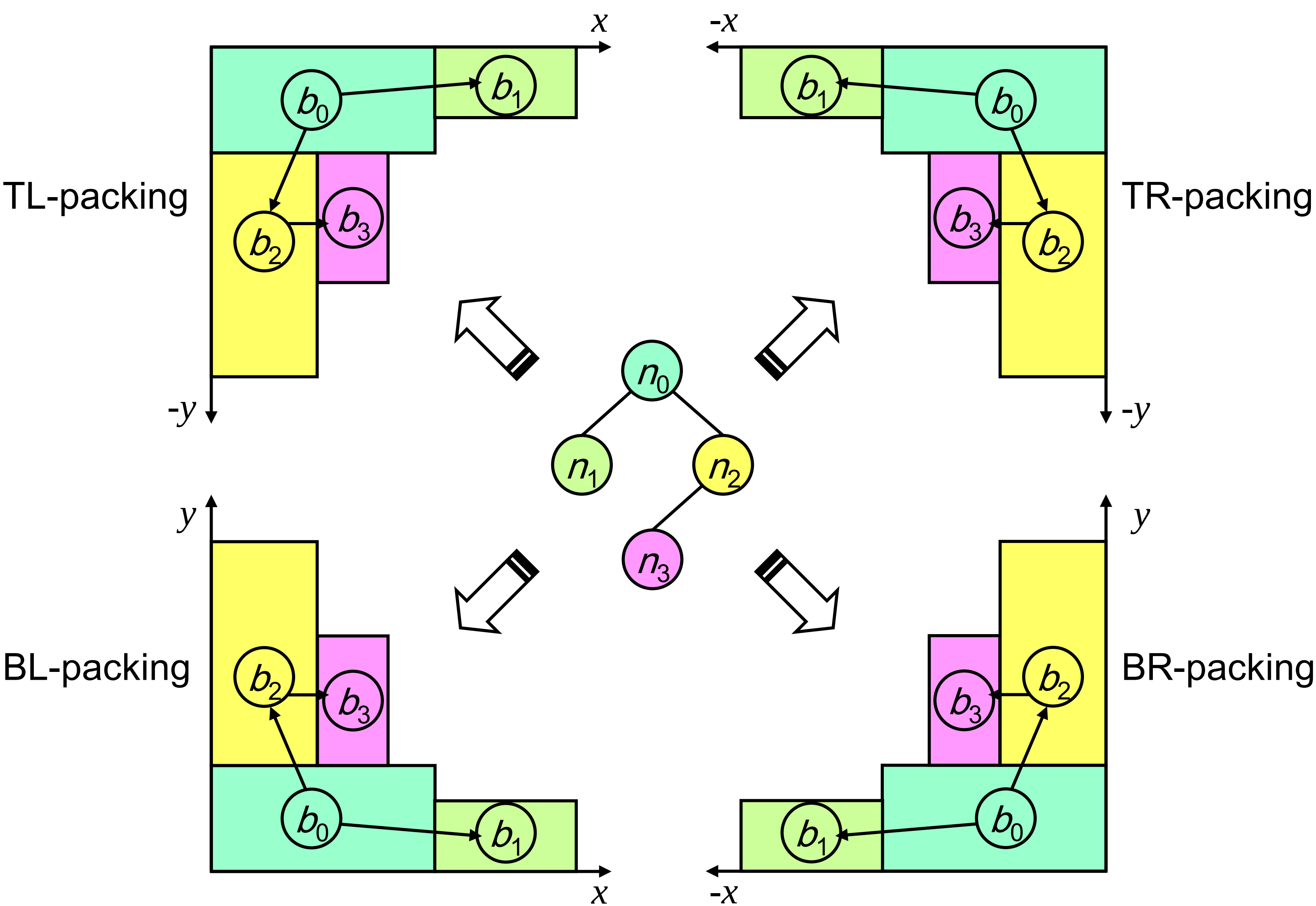
By Tung-Chieh Chen

But What If %Macro Area Is Not High?



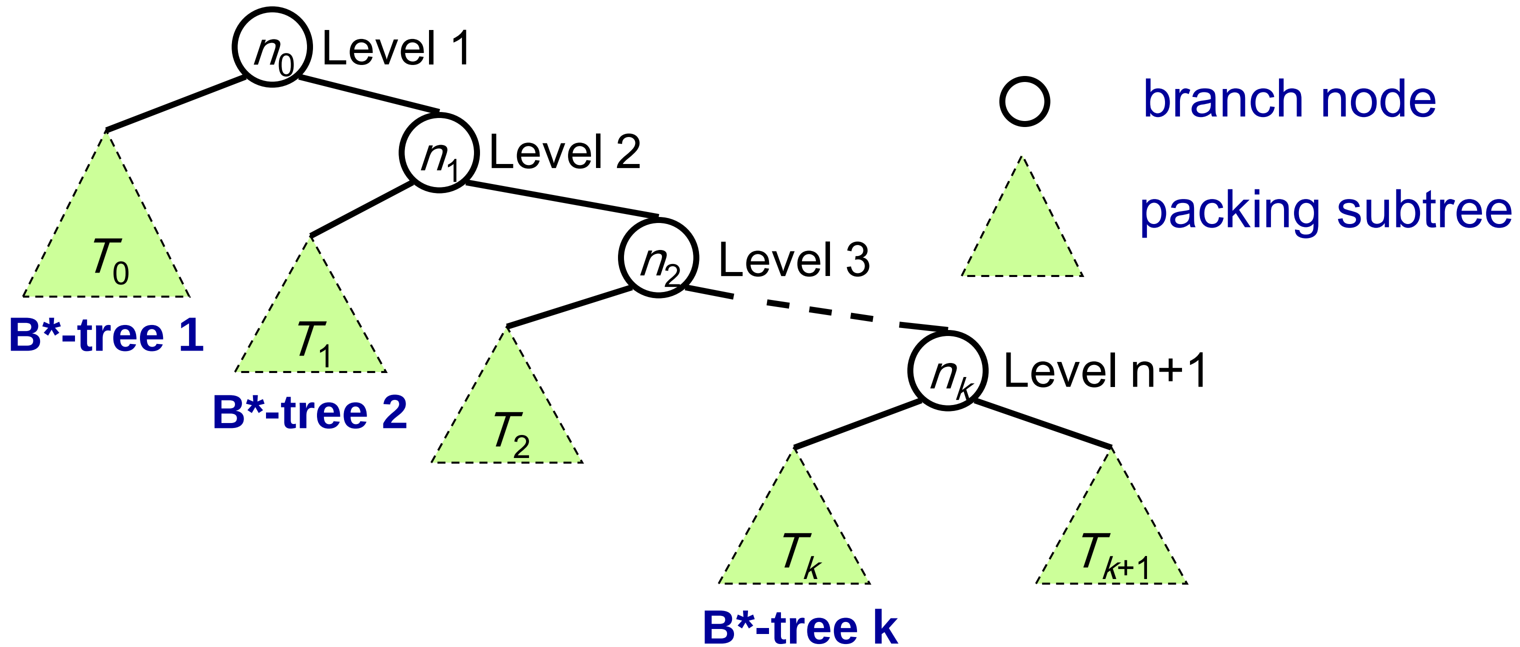
All macros will be packed together!!

Multi-Packing (MP) Tree Representation



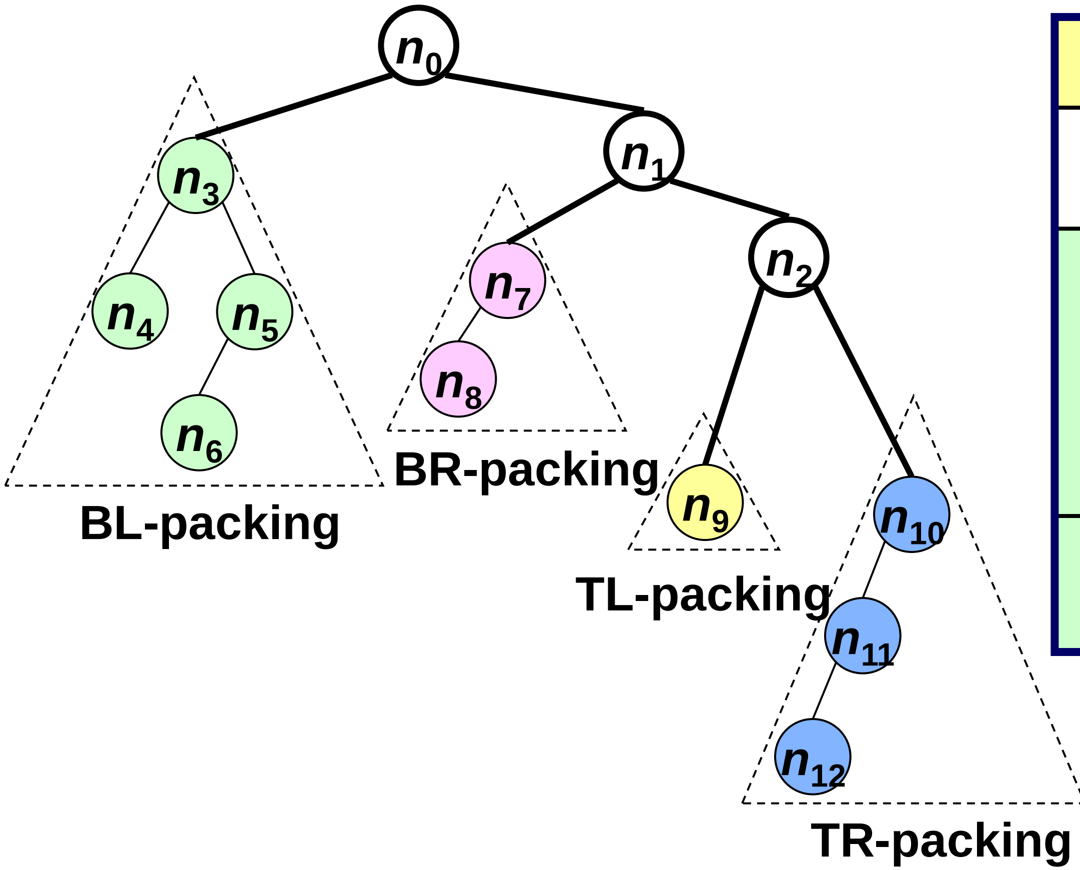
Generalized MP-Trees

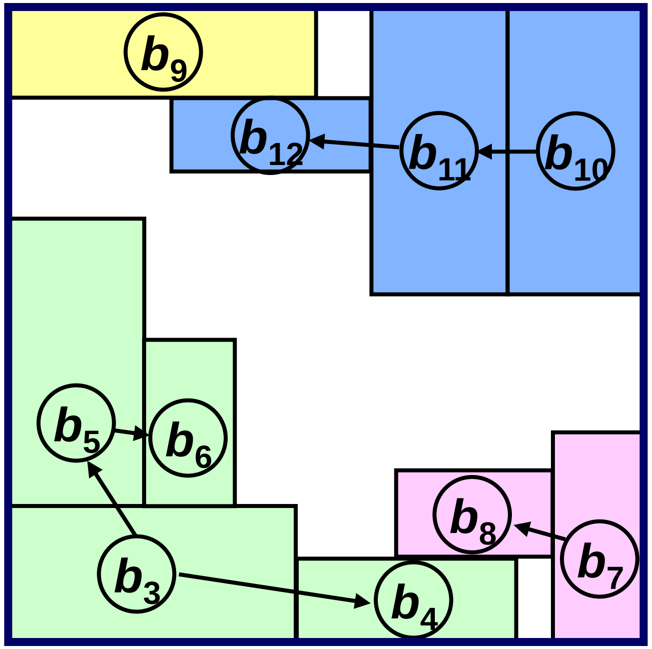
- Working on four independent packing trees may not obtain a desired solution
  - Lack global interactions among different subproblems
- Key: **Combine packing trees** packing to different corners
  - Chen et al., "MP-trees: A packing-based macro placement algorithm for modern mixed-size designs," DAC'07 & TCAD'08
- Use the right skewed branch for easier implementation



MP-tree Macro Placement Example

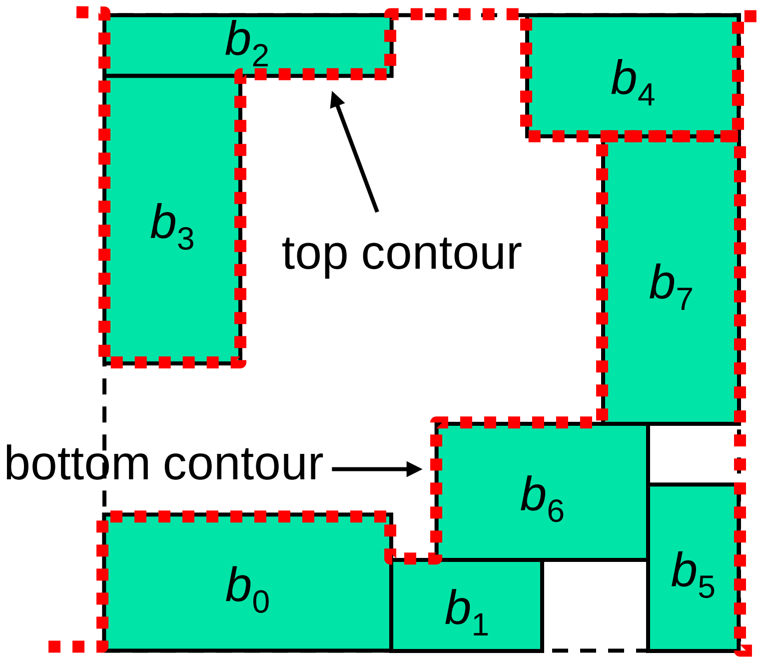
- Use four packing subtrees to handle a rectangular chip
- Applies to a placement region with any number of corners

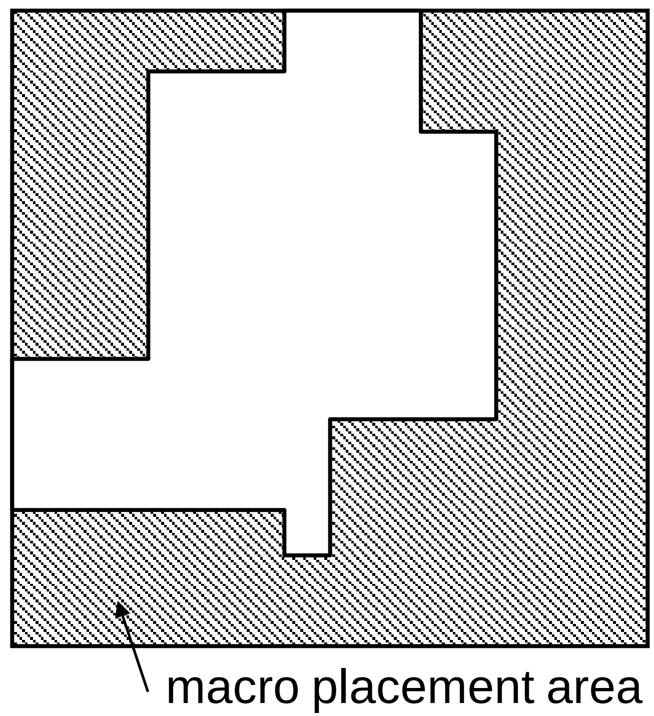




Evaluation of a Macro Placement

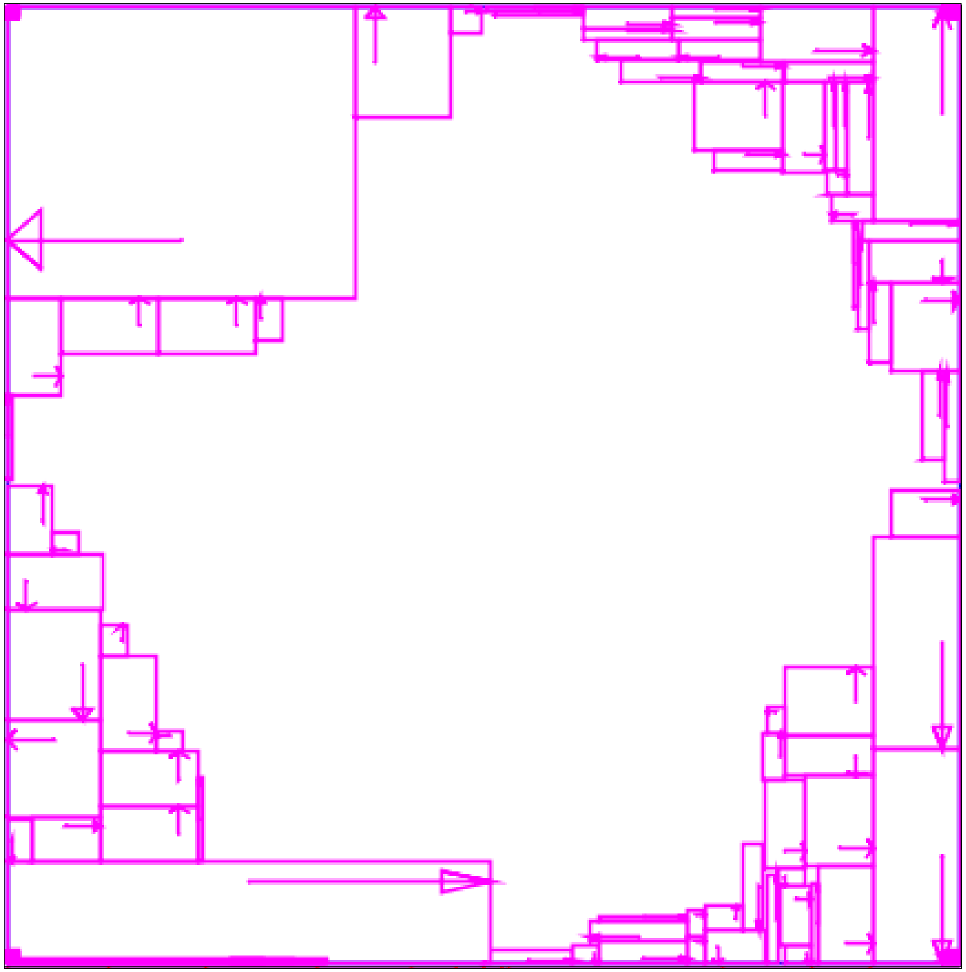
- Macro placement area
- Wirelength
- Macro displacement





Demo: MP-Tree Placer (1/2)

- Stage 1: Macro placement



Circuit: adaptec5

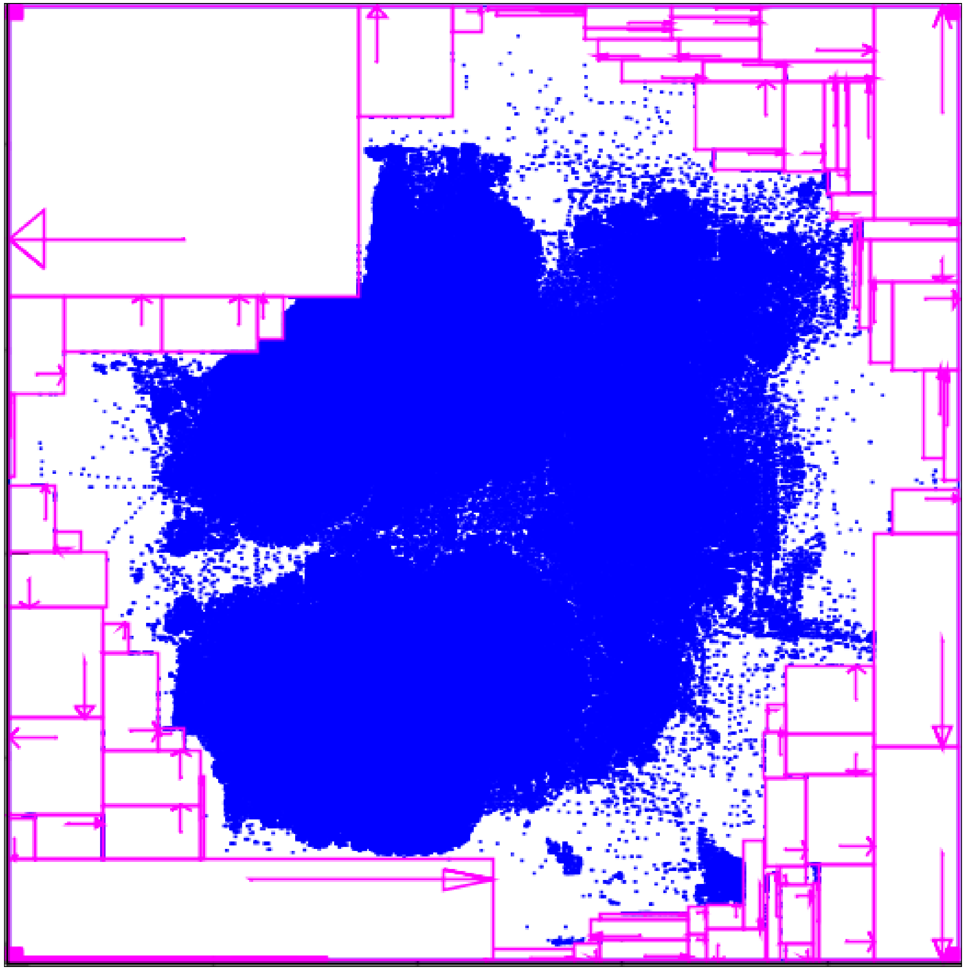
#Cell: 842k

#Net: 867k

#Macro: 76

Demo: MP-Tree Placer (2/2)

- Stage 2: Standard-cell placement



Circuit: adaptec5

#Cell: 842k

#Net: 867k

#Macro: 76

HPWL: 3.27e6

Results on the ISPD-06 Benchmarks

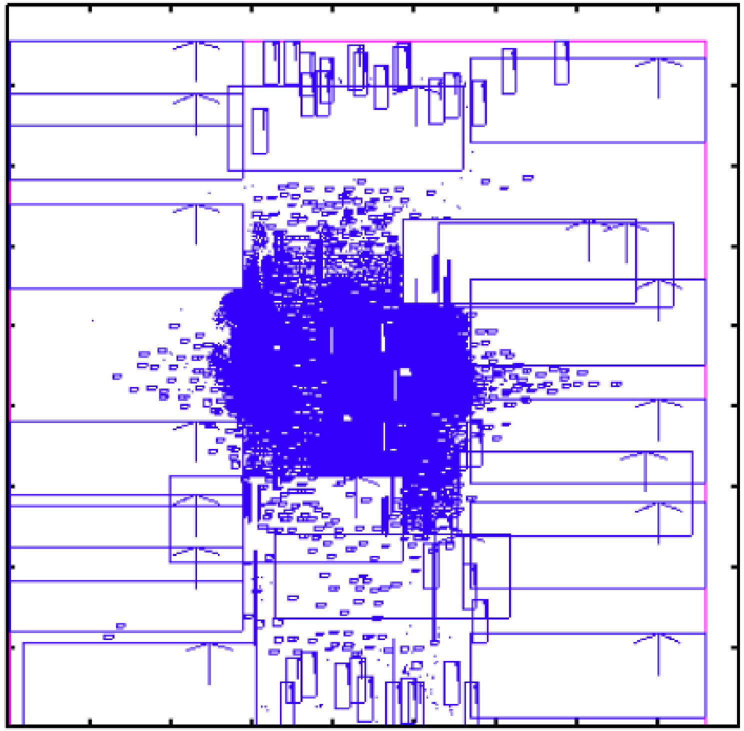
- The higher the chip utilization rate, the more the wirelength reduction.

Circuit	NTUplace3 HPWL (e7)					
	85% util		90% util		95% util	
	w/o	w/ MPT	w/o	w/ MPT	w/o	w/ MPT
adaptec5	30.55	30.40	30.29	30.48	47.25	32.30
newblue1	6.64	6.30	6.74	6.38	6.85	6.62
newblue2	20.44	21.23	20.96	19.29	25.34	20.61
newblue3	**NR**	31.21	**NR**	29.64	**NR**	38.68
newblue4	22.82	21.41	26.70	22.68	26.83	23.77
newblue5	41.09	40.21	49.12	47.97	72.56	68.14
newblue6	45.45	45.46	53.14	47.60	66.51	65.21
newblue7	111.92	114.12	**NR**	120.15	**NR**	136.87
Average	**1.00**	**0.99**	**1.00**	**0.93**	**1.00**	**0.88**

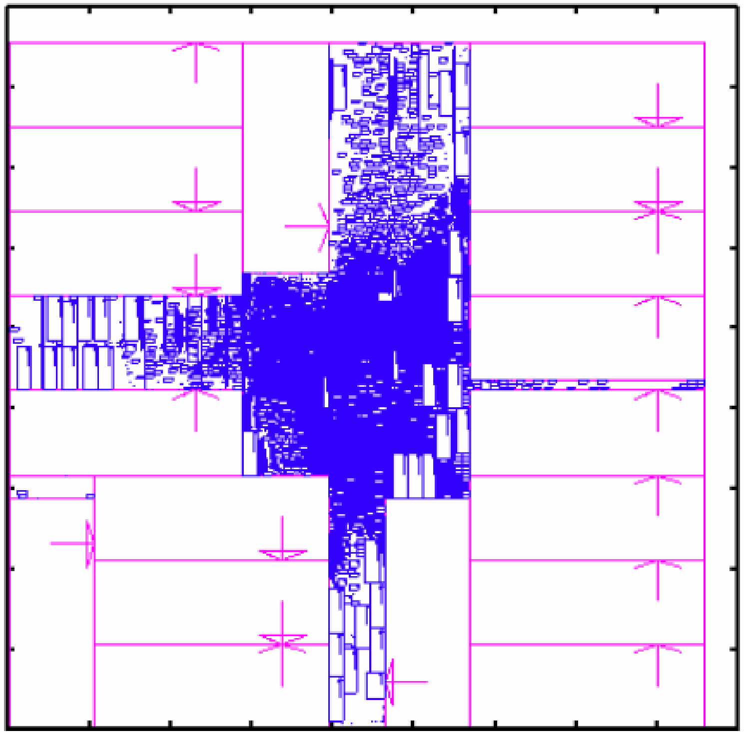
***w/o: NTUplace3 alone**
***w/ MPT: MP-tree + NTUplace3**

NR: no legal result

ISPD-06 newblue3 Layouts



NTUplace3 alone (failed to find a legal placement)



MP-tree + NTUplace3

Integration with Other Placers

- Capo 10.2: 12% wire reduction, 21% more CPU time
- mPL6: 4% wire reduction, more robust

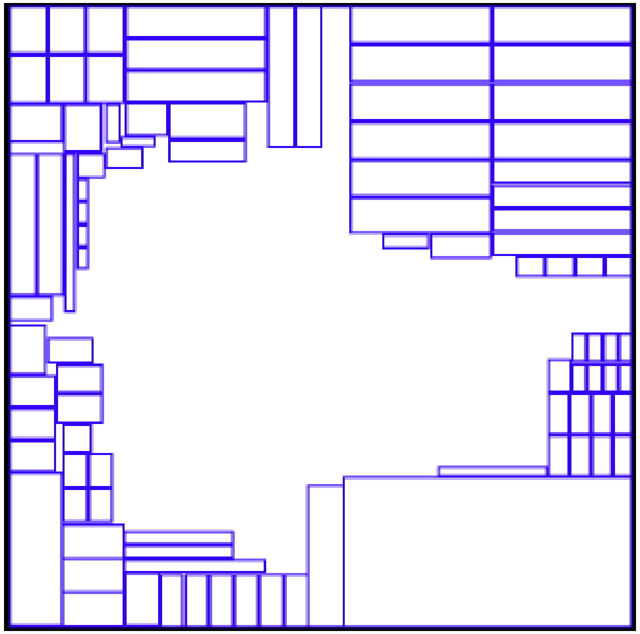
Circuit	Capo 10.2				mPL6			
	HPWL (e7)		CPU (min)		HPWL (e7)		CPU (min)	
	w/o	MPT	w/o	MPT	w/o	MPT	w/o	MPT
adaptec5	38.29	33.52	432	537	**NR**	28.72	**NR**	138
newblue1	9.56	6.71	155	109	6.45	6.18	47	47
newblue2	25.99	22.05	287	234	**NR**	18.18	**NR**	94
newblue3	33.27	34.00	263	432	**NR**	31.11	**NR**	116
newblue4	26.93	24.00	311	451	**NR**	21.04	**NR**	93
newblue5	47.07	42.96	775	894	**NR**	39.94	**NR**	239
newblue6	55.22	49.23	795	882	**NR**	45.33	**NR**	296
newblue7	119.48	107.99	1795	2752	**NR**	94.76	**NR**	588
Average	**1.00**	**0.88**	**1.00**	**1.21**	**1.00**	**0.96**	**1.00**	**0.99**

Mchip Benchmark Results

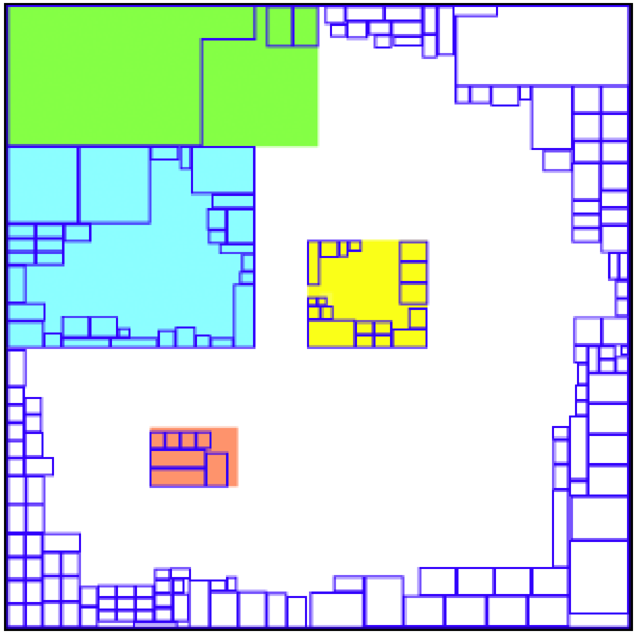
- Cell ~1320K, macro ~380, macro area ratio ~66%
- Placed HPWL is 35% shorter than Capo′s
- Routed WL is 55% shorter than Capo′s
- Compared with two leading commercial placers
  - 6% -- 56% shorter placed HPWL
  - 7% -- 67% shorter routed WL

Mchip	HPWL		Routed WL		GRC Overflow	
	Ours	Capo	Ours	Capo	Ours	Capo
Mchip1	5.26	5.84	6.13	6.56	0.7%	0.7%
Mchip2	4.72	5.65	5.34	6.65	0.1%	1.0%
Mchip3	5.26	10.00	6.02	16.90	0.1%	36.4%
Mchip4	11.76	14.12	13.27	14.16	0.1%	1.4%
Mchip5	8.92	**NA**	9.85	**NA**	0.0%	**NA**
Avg	**1.00**	**1.35**	**1.00**	**1.55**		

Mchip Benchmark Results


(a) mchip2

95 Macros


(b) mchip4

380 Macros w/
4 region constraints

Type 3: One-stage Approach

Type 1: Constructive Approach

- Combine floorplanning and placement
- Examples: Capo, PATOMA, FLOP

Type 2: Two-stage Approach

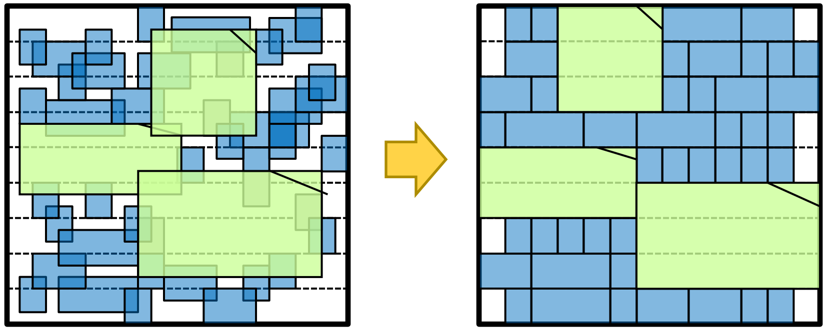
- Perform (1) macro placement and then (2) cell placement
- Examples: MP-tree, CG

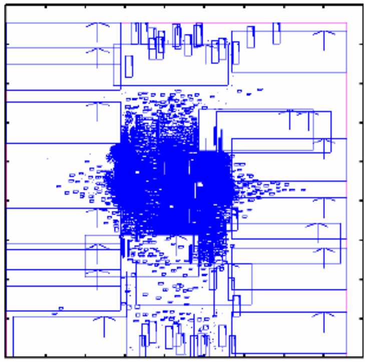
Type 3: One-stage Approach

- Place macros and cells simultaneously
- Examples: mPG-MS, APlace, mPL, UPlace, NTUplace3, etc.

One-Stage Approach

- One-stage mixed-size placers
  - Place both macros and cells simultaneously
    - mPG-MS [ASPDAC'03], APlace [ICCAD'04], mPL [ISPD'05], UPlace [ISPD'05], NTUplace3 [ICCAD'06, TCAD'08], etc.





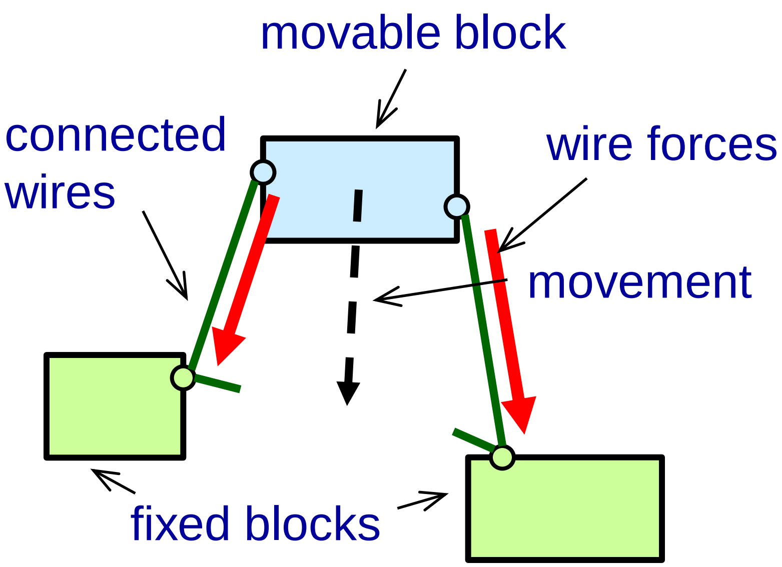
- Analytical placement
  - Has been shown to be most effective for **cell** placement
  - Key limitation: macro handling in global placement
    - **Macro rotation and legalization**

Forces in Analytical Formulation

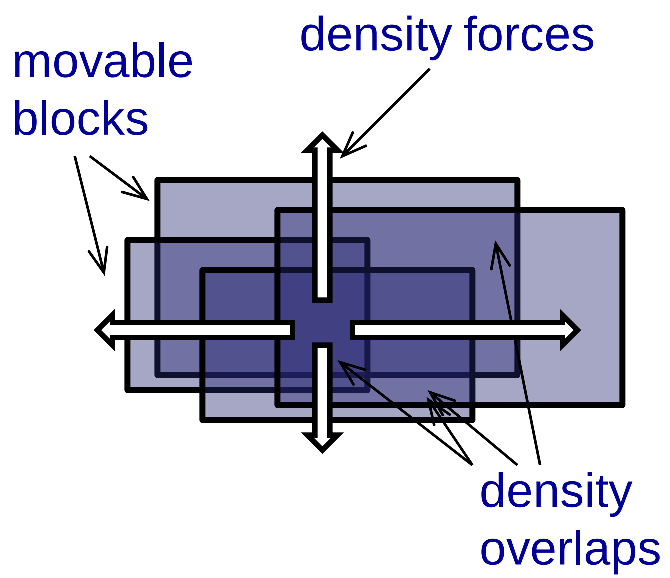
- Analytical placement formulation

$$\textbf{min} \quad \boxed{W(x, y)} + \lambda\boxed{\Sigma(D_b(x, y) - M_b)^2}$$

Wire force



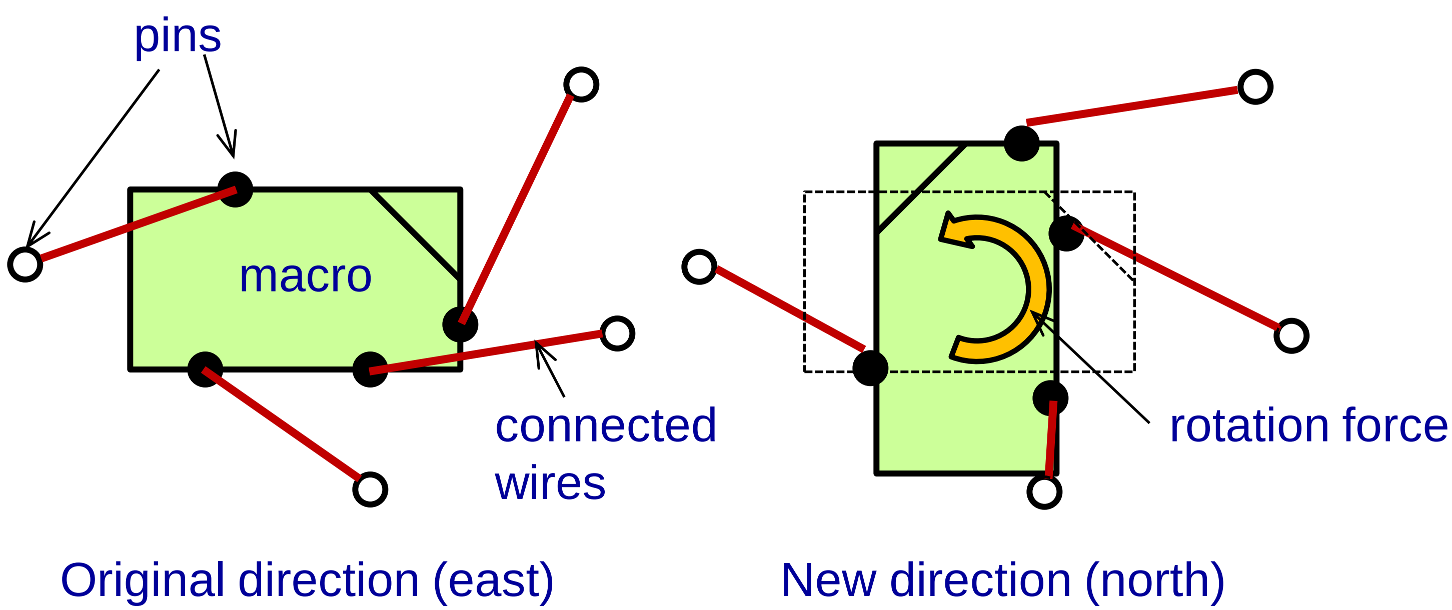
Density force



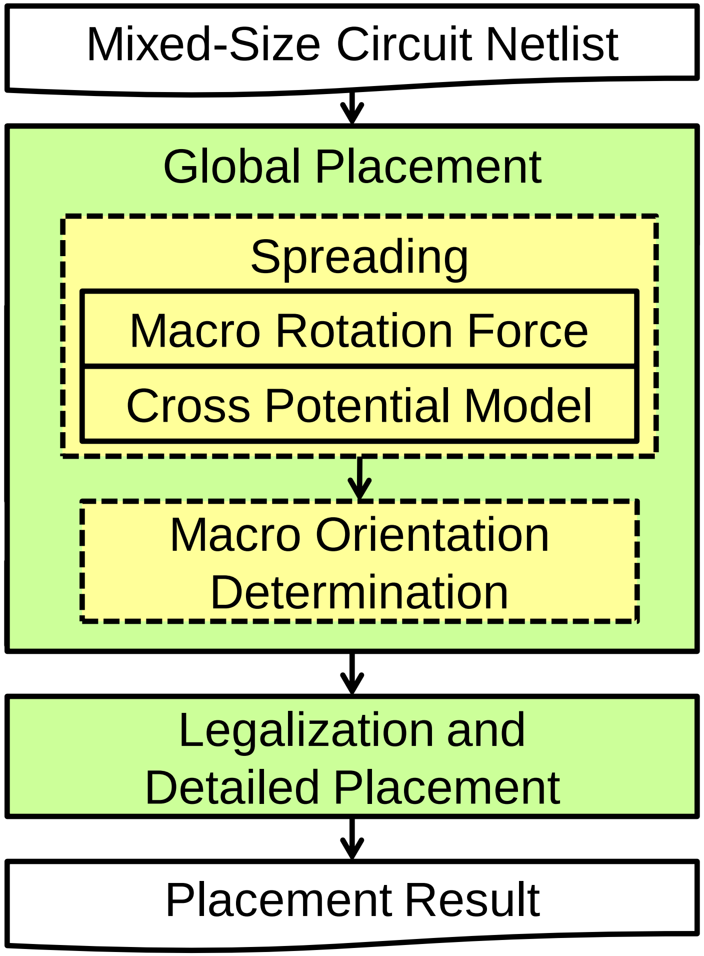
Rotation?

First Unified Approach

- Hsu & Chang [ICCAD'10] present the first attempt to rotate and legalize macros during analytical placement
- Macro rotation force
  - Is induced from wire connections, similar to wirelength gradient (wire force) for wirelength optimization



Unified Analytical Placement: NTUplace-m

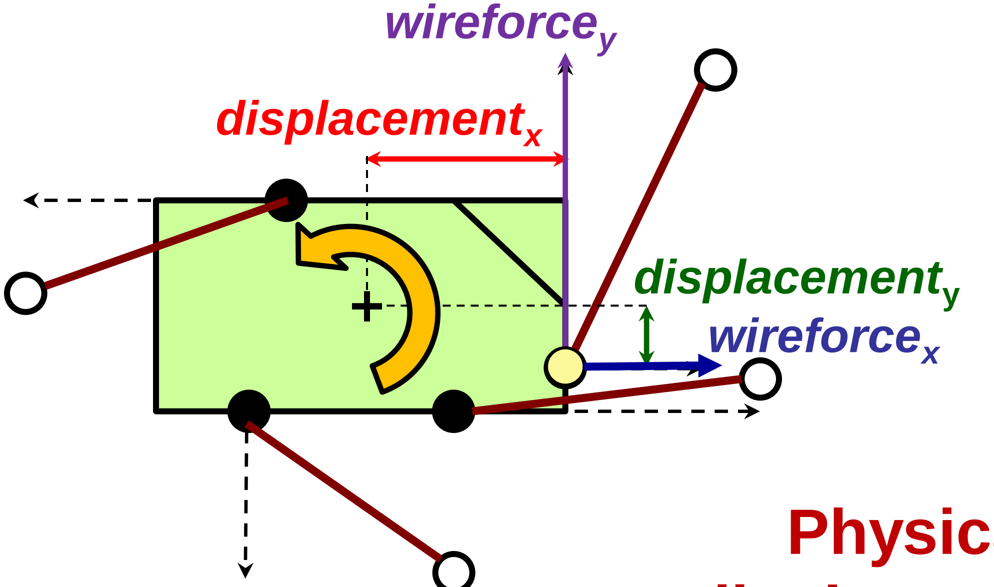


Global placement is the most critical step in placement

Rotation Force Modeling: Torque

- Model the rotation force according to wire forces
  - Use the ***torque*** concept in physics to determine the orientations of macros

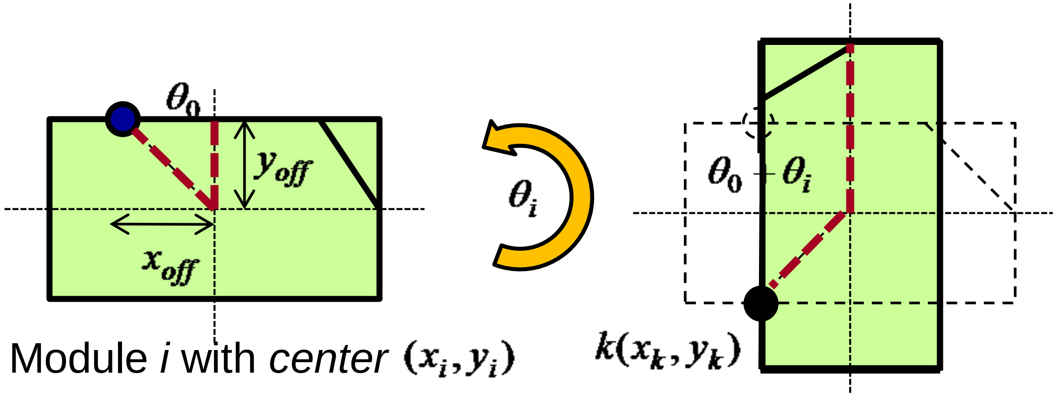
$$Torque = wireforce_x \times displacement_y + wireforce_y \times displacement_x$$



Physical displacement?

Macro Rotation Force Modeling

- New (x_k, y_k) from (x_i, y_i) after rotation by degree θ_i



Module *i* with *center* (x_i, y_i) $k(x_k, y_k)$

$$\begin{cases} x_k = x_i + x_{off}\cos\theta_i - y_{off}\sin\theta_i \\ y_k = y_i + x_{off}\sin\theta_i + y_{off}\cos\theta_i \end{cases}$$

- Rotation force: gradient of wirelength on the direction of the rotation degree => differentiate wirelength function by degree θ_i

virtual displacements

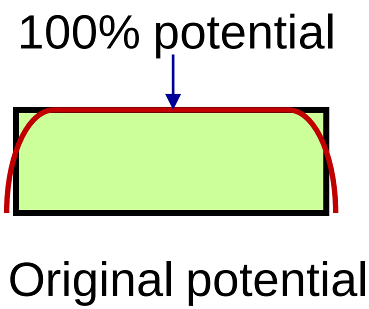
$$\frac{\partial W}{\partial \theta_i} = \frac{\partial W}{\partial x_k}\cdot\frac{\partial x_k}{\partial \theta_i} + \frac{\partial W}{\partial y_k}\cdot\frac{\partial y_k}{\partial \theta_i} = \frac{\partial W}{\partial x_k}\cdot(-x_{off}\sin\theta_i - y_{off}\cos\theta_i) + \frac{\partial W}{\partial y_k}\cdot(x_{off}\cos\theta_i - y_{off}\sin\theta_i)$$

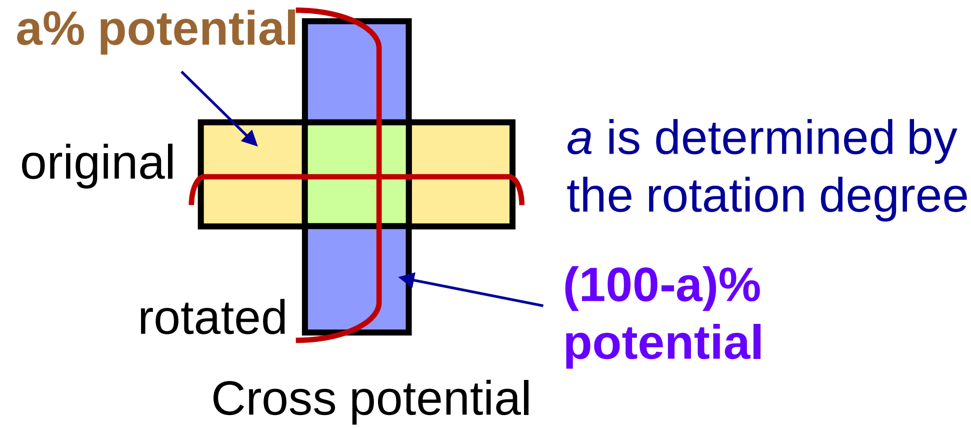
wire forces on *x* and *y* directions at pin *k*

- Continuous degree? But macro rotation is non-continuous!!
  - Cross potential model

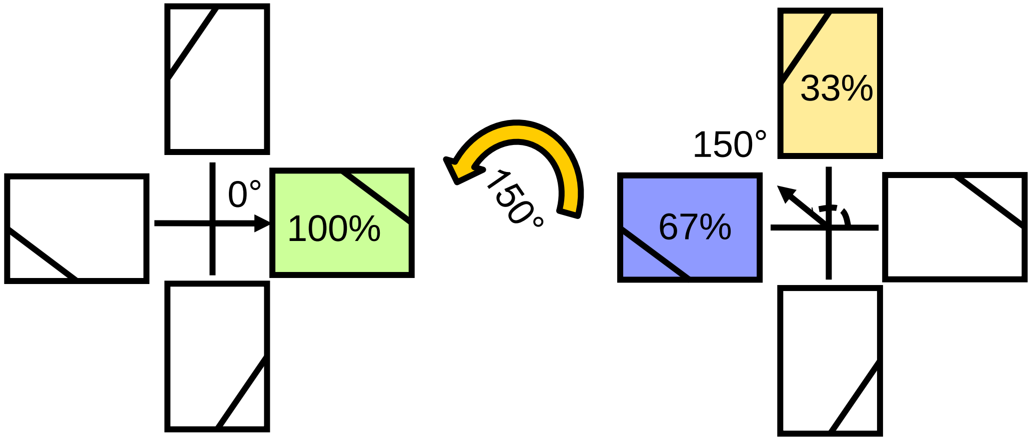
Cross Potential Model

- Consider both the original and rotated potentials



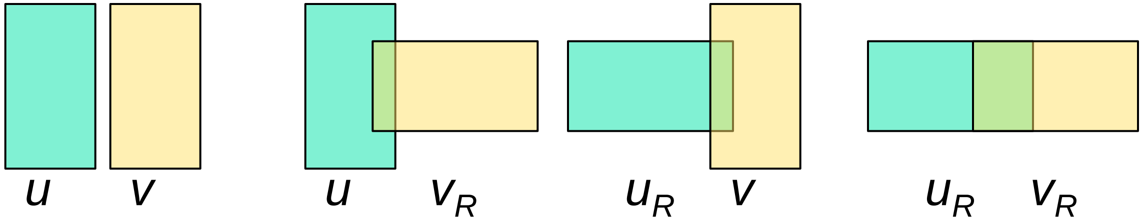


- Example



Macro Orientation Determination

- At the end of global placement, each macro is rotated to the direction with max potential and min overlaps
- Objective: minimize overlaps among macros
  - For macros u and v, there are ≦ 4 overlapping combinations



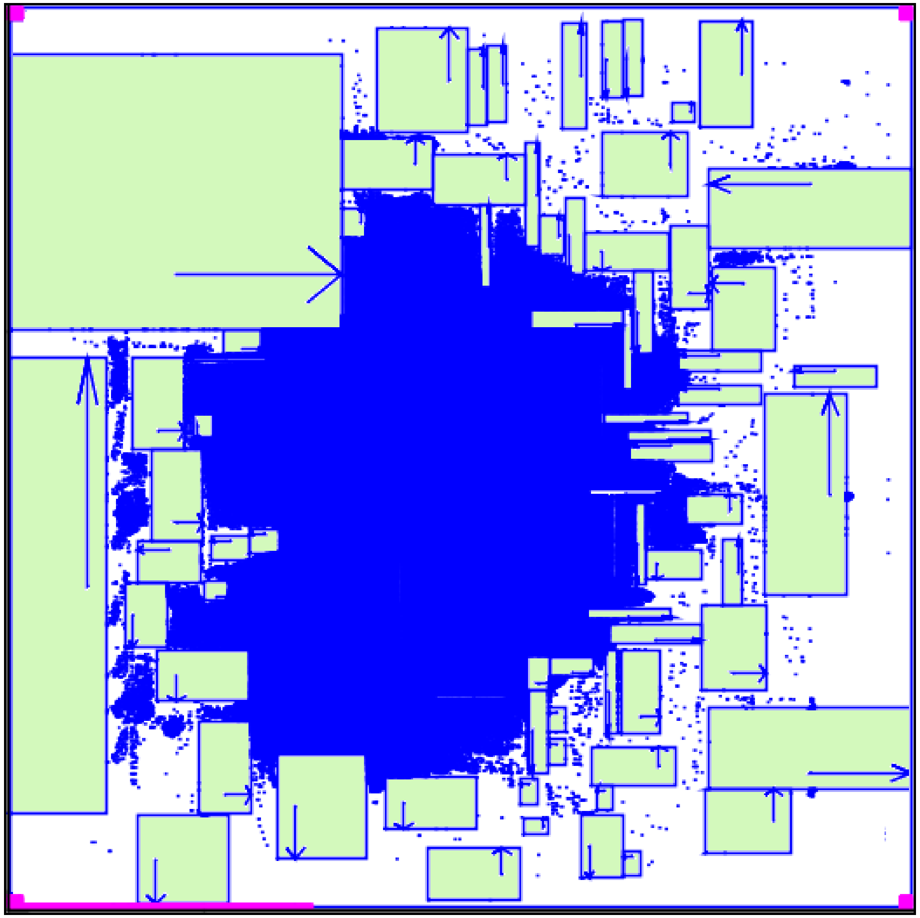
  - Overlap function for two macros u and v

$$\Psi(u, v) = (1-r_u)(1-r_v)\varphi(u, v) + (1-r_u)r_v\varphi(u, v_R) + r_u(1-r_v)\varphi(u_R, v) + r_u r_v\varphi(u_R, v_R)$$

    - u_R (v_R): rotated macro of u (v)
    - $\varphi(u,v)$: overlaps between two macros with given orientations
    - $r_u = 0$, macro u rotated by 0 (or 180) degree
      $r_u = 1$, macro u rotated by 90 (or 270) degree
  - Can use ILP to solve this problem

Demo: Unified Mixed-Size Placement

- Simultaneous macro and standard-cell placement with macro orientation handling
- At least 5% better wirelength than existing placers



Circuit: adaptec5

#Cell: 842k

#Net: 867k

#Macro: 76

HPWL: 2.86e8

Comparisons

Mixed-size placement	Pros	Cons
Constructive approach	➢ Keep macros overlap-free with floorplanning ➢ Is fast with good scalability	➢ Solution quality is usually limited by the intrinsic problems with partitioning ➢ Is less effective for spare designs
Two-stage approach	➢ Is robust in finding legal placement ➢ Is widely used in the industry ➢ Is suitable for **dense** design with higher utilizations	➢ Need a good macro placer ➢ Gaps between placements of macros and standard cells limit the quality of the final placements
One-stage approach	➢ Analytical placement is most effective for standard-cell placement ➢ Close the gap between macro and cell placement ➢ Can handle sparse designs well	➢ Special consideration for macro handling, macro legalization and rotation, are needed

Outline

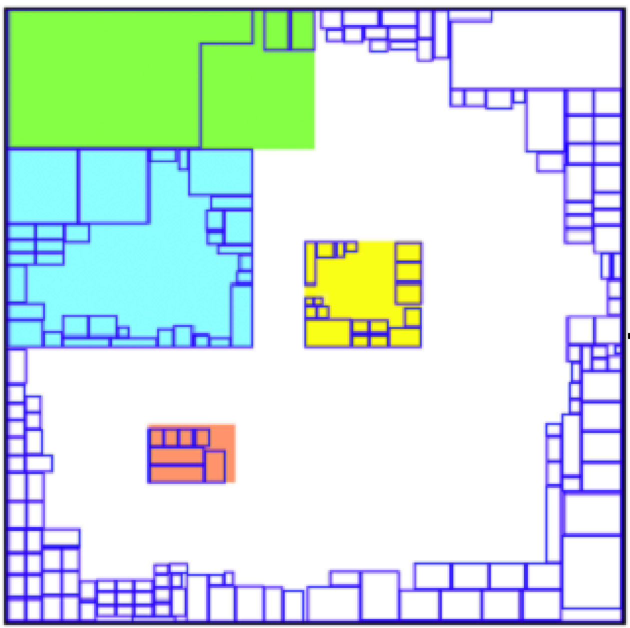
- Placement Basics
- Cell Placement
- Mixed-Size Placement
- **Future Research Directions**

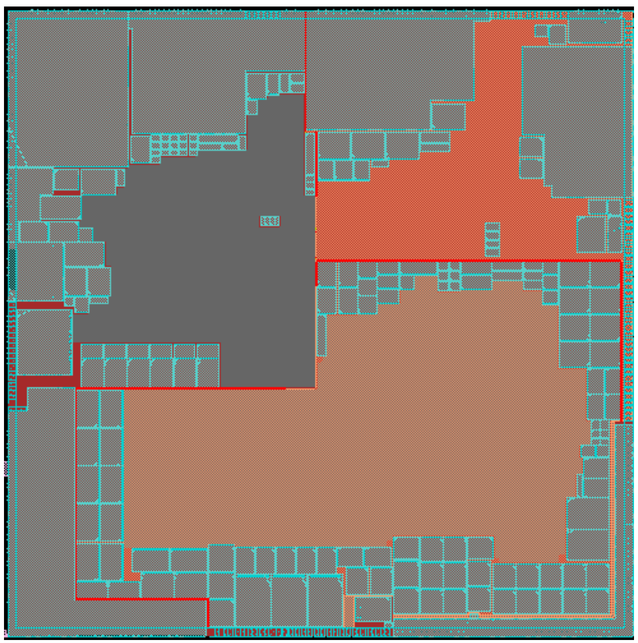
Future Research Directions

- Large-scale mixed-size placement
- Routability-driven placement
- Timing-driven placement
- Power-delivery-aware placement
- Simultaneous clock network synthesis and placement
- Manufacturability-aware placement
- Stress-aware placement
- Thermal-aware 3D IC placement

Large-Scale Mixed-Size Placement

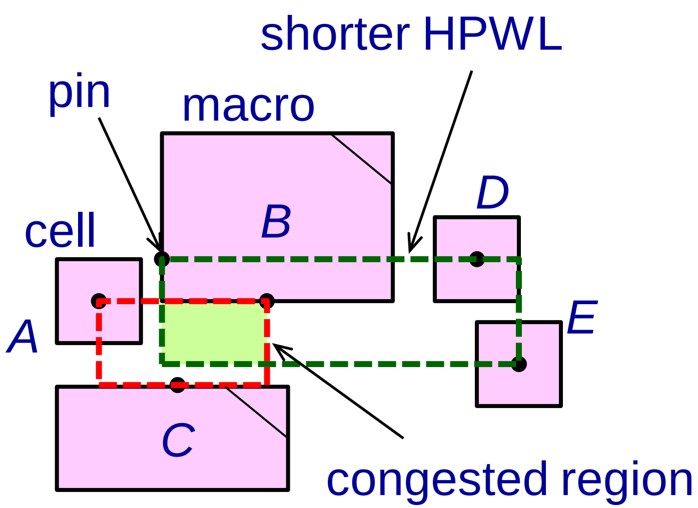
- We still have a long way to go for large-scale mixed-size placement!!
  - Find best trade-offs among existing approaches?
  - Need to consider many other placement constraints
  - Could be multiple mixed-size domains: recursive MP-trees?



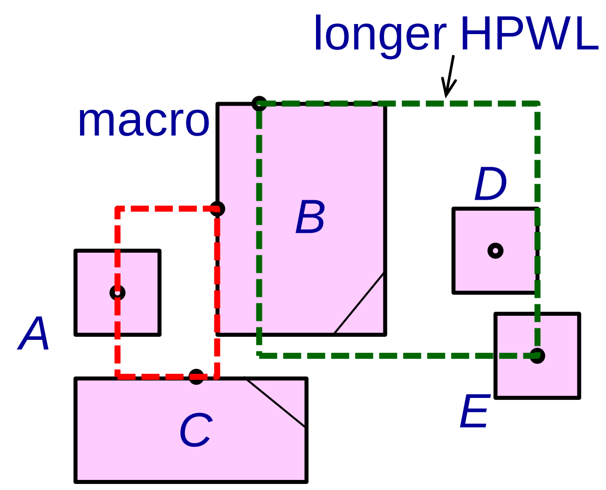


Routability-Driven Placement

- Routability issues for mixed-size placement becomes more challenging: **ISPD 2011 Contest Problem!**
  - Macro porosity, ratio of available routing resources above a macro block, and macro rotation induce more problems



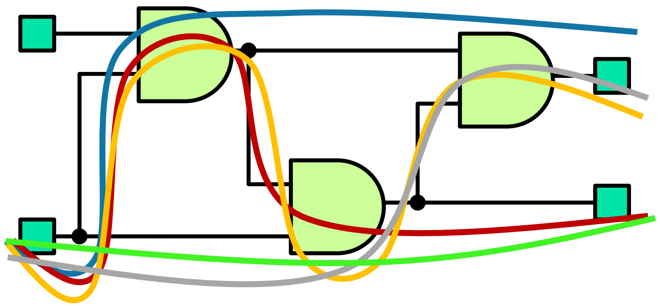
Wirelength minimization
Routing congested region occurs



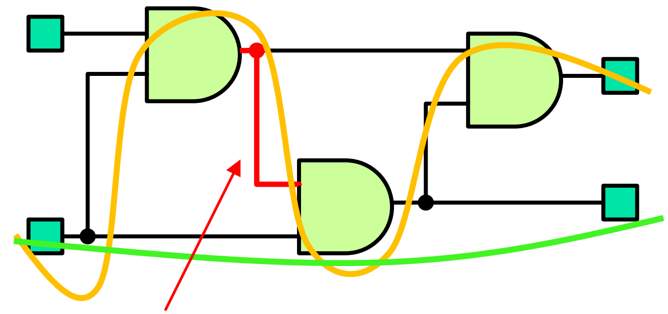
Congestion optimization
Macro B is rotated

Timing-Driven Placement

- Two major techniques for timing-driven placement
  - Path-based methods incur prohibitively time complexity due to the exponentially increasing number of paths
  - Net-based methods lack the global view of the full path



timing paths increases exponentially

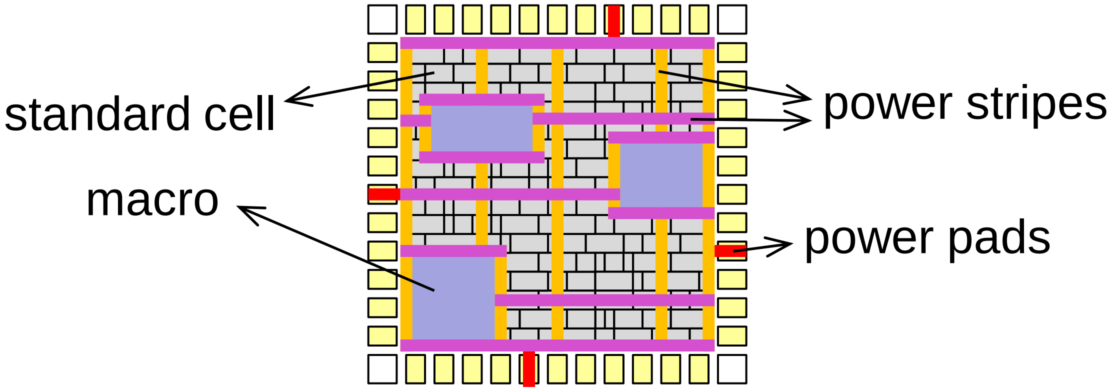


Net based method may not optimize timing effectively

- A timing optimization technique with low-complexity and high controllability is desired
- Timing-driven placement with macro awareness
  - Macros cause wirelength and routability issues
  - Timing requirements for macro blocks should be considered

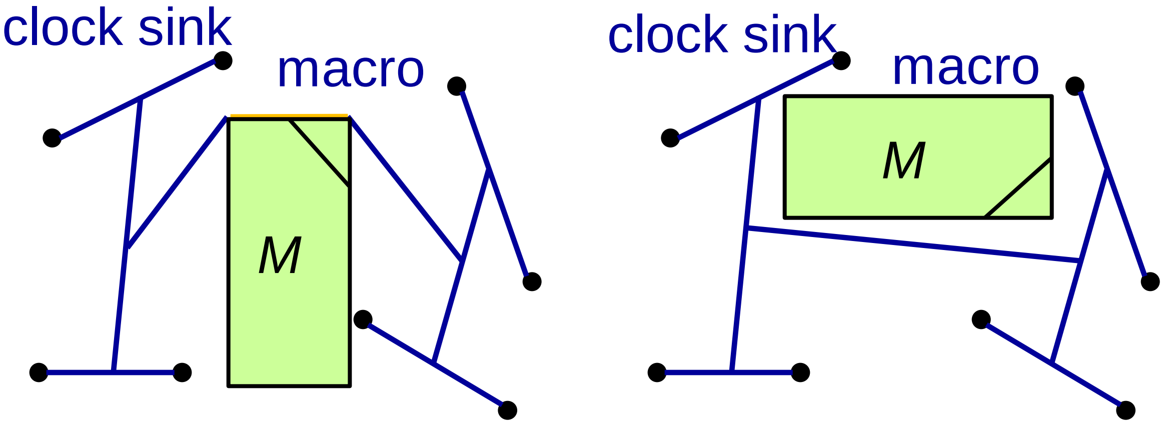
Power-Delivery-Aware Placement

- Voltage (IR) drops
  - Limit circuit performance, slow down the slew rate, and increase power consumption
  - Depend on the distance between placed macros/cells and power network
  - Should be considered during placement to reduce the power consumption
- For mixed-size designs, big macros introduce additional power rings and power stripes
  - Make power network and power delivery problems more difficult



Clock-Network-Aware Placement

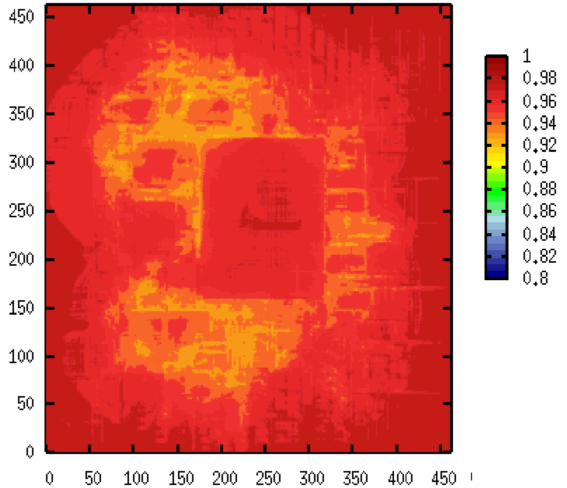
- Clock network synthesis (CNS) constructs the clock network which distributes clock signals from a source point to all the sequential elements connected it
- For modern mixed-size designs, big macros might cause obstacles for clock network synthesis



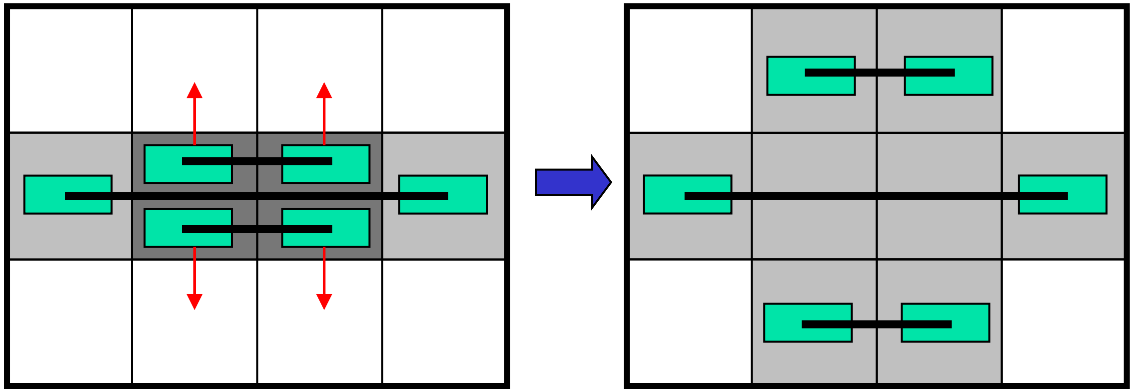
CNS designs with different macro orientations and positions

Manufacturability-Aware Placement

- Predictive Chemical Mechanical Polishing (CMP) model
  - The number of dummy fills and normalized copper thickness are functions of wire density [Cho et al, ICCAD-06]
- Wire density optimization is limited by pin locations
- Shall move cells/pins out of the high wire-density regions during placement
  - Chen, et al., ISPD-08 (TCAD, 2008)



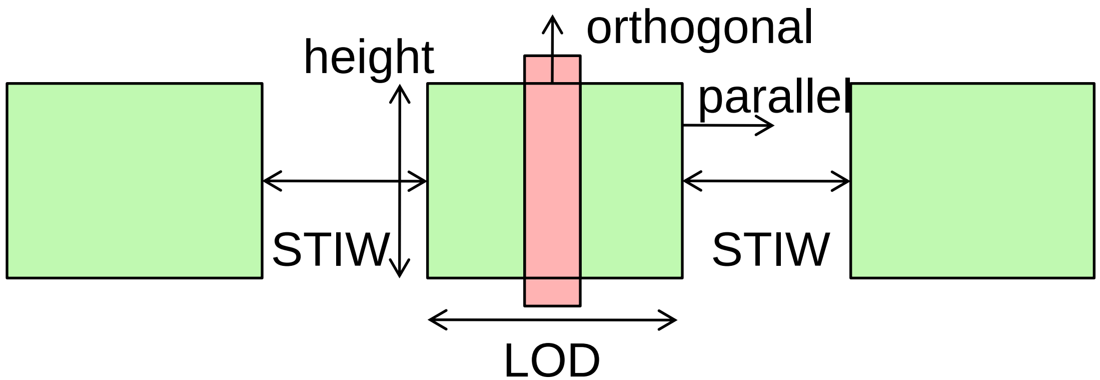
Normalized Copper Thickness Map



Better wire distribution

Stress-Aware Placement

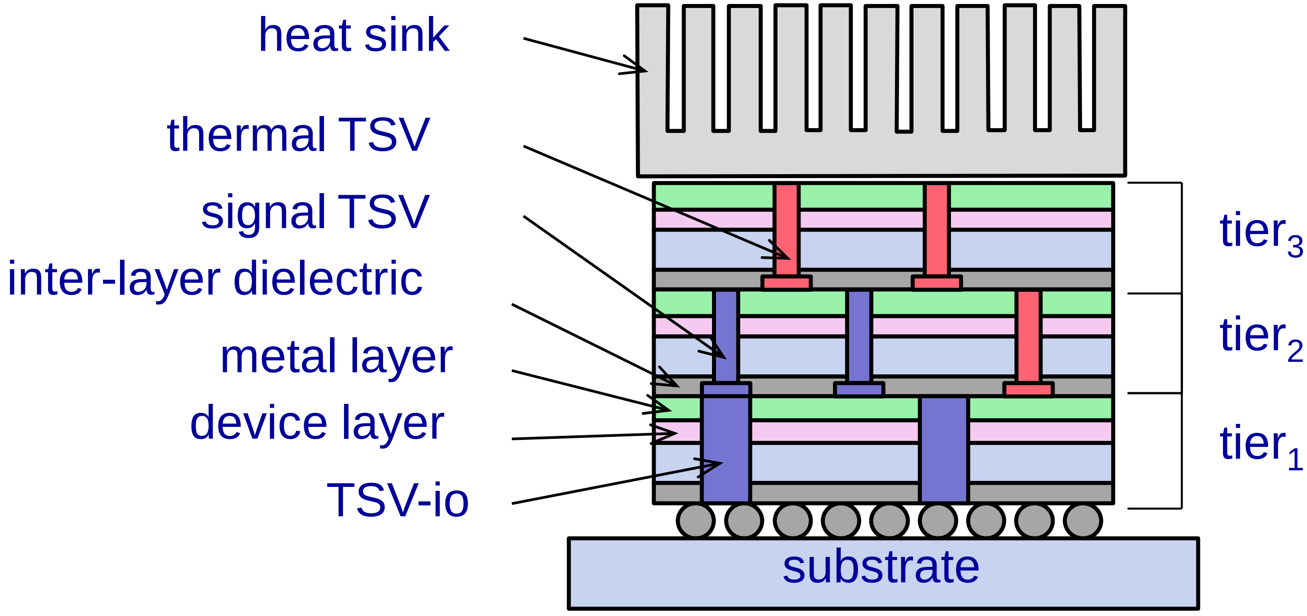
- Shallow trench isolation (STI) is the mainstream CMOS isolation technique for advanced circuit design
  - By exploiting STI wells between device active regions, STI stress can effectively improve transistor performance
    - STI width (STIW) and length of diffusion (LOD)



  - $MOB_{L,R} = \gamma[(LOD/2)^{\alpha} + \beta/STIW_{L,R}]$ [Kahng et al., ICCAD'07]
    - If STIW↑ or LOD↓, then pMOS mobility ↑
    - If STIW↓ or LOD↑, then nMOS mobility ↑
- Problem: place cells to optimize STIW between neighboring cells while achieving timing requirements

Thermal-aware 3D IC Placement

- Problem: Place cells into multiple tiers (dies) to optimize wirelength, etc.
- Important issues: reliability, thermal, routability, mixed-size design, etc



Conclusions

- Cell placement
  - NTUplace3 analytical placement framework
  - The WA wirelenth model for analytical global placement
- Mixed-size placement designs
  - Become a mainstream for modern circuit designs
  - Incur more challenges to modern circuit placement
- Major mixed-size placement approaches
  - Two-stage approach: place macros followed by standard cells
  - One-stage approach: handling macro rotation & orientation is key
  - Each has its pros and cons: trade-offs among solution quality, runtime efficiency, and utilization flexibility
- Many modern challenges, e.g.,
  - Multiple domains/objectives/constraints: routability, timing, power, CTS, stress, 3D IC designs, etc.

Keys to Our Research Solutions: CAR

Criticality

Abstraction

Restriction

Thank You!!

National Taiwan University