Dummy Fill Optimization for Enhanced Manufacturability

Yaoguang Wei and Sachin S. Sapatnekar

Department of Electrical and Computer Engineering
University of Minnesota

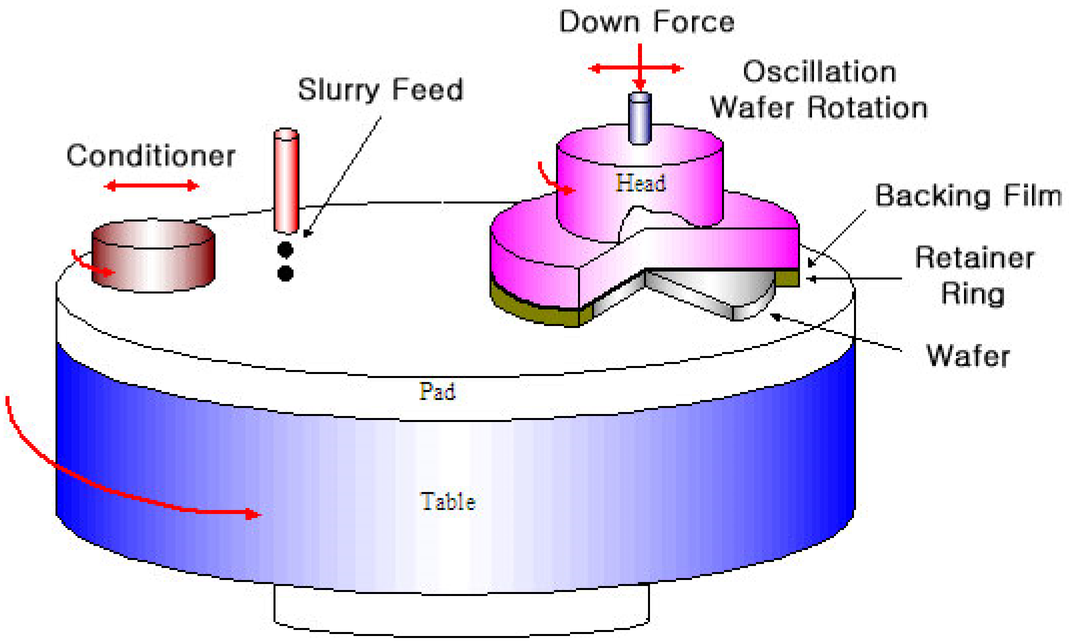
ISPD 2010

Outline

- Introduction
- Flow of the routing algorithm
- Cost functions
- Experimental results
- Conclusion

Chemical mechanical polishing (CMP)

- CMP: chemical and mechanical means to polish wafer



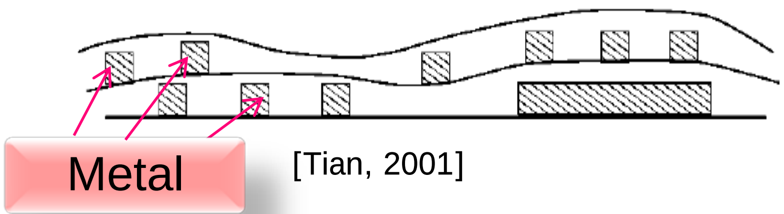
CMP process

CMP machine

[http://www.ceramic.hanyang.ac.kr/paik/cmp.htm]
[http://strasbaugh.com/cm/Products/CMP/nTREPID%20300mm%20PRODUCTION%20CMP.html]

Variations caused by oxide CMP

- Oxide CMP used to polish the interlayer dielectric (ILD)
- Variation in the post-CMP ILD thickness



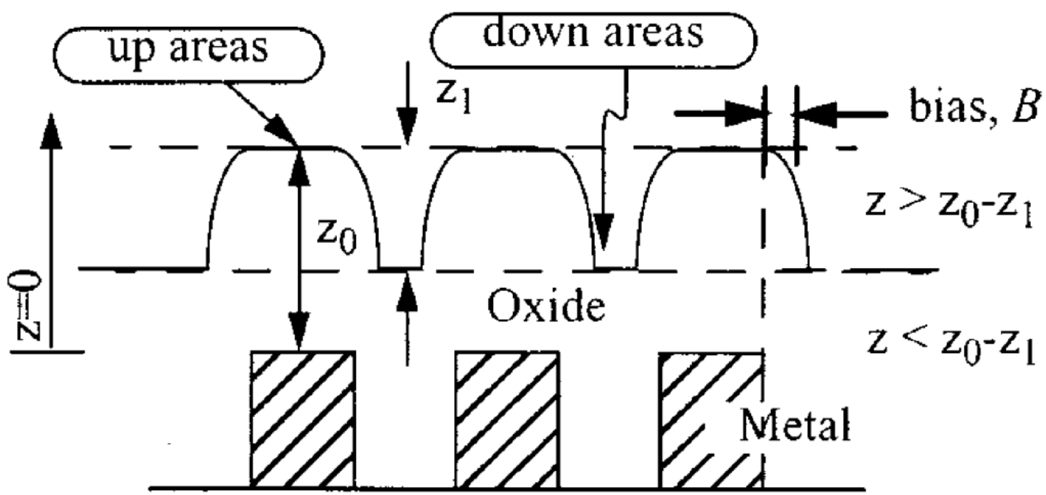
[Tian, 2001]

Depth of focus variation, and finally lower the performance and yield

e.g., CMP variation can lead to 30% delay variation [Mehrotra, 2000]

- Oxide CMP model in [Ouma, 2002]
  - Post-CMP ILD thickness z ~ effective pattern density ρ

$$z = c_0 + z_1\rho$$

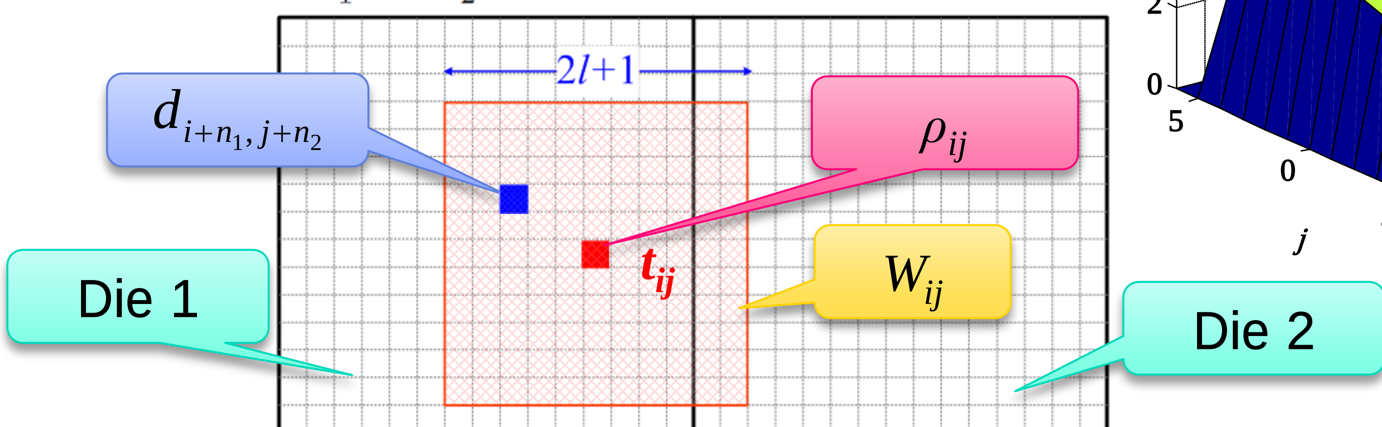


[Ouma, 2002]

Effective pattern density (EPD)

- Partition the layout to tiles
- For each tile t_{ij}
  - **Initial pattern density (IPD)** $\boldsymbol{d_{ij}}$: Density of metal in a tile
  - Assume the die is periodically repeated, and then d_{ij} is periodic.
  - **Effective pattern density (EPD)** $\boldsymbol{\rho_{ij}}$: Weighted sum of d_{ij} over a weighting window W_{ij} with size $2l+1$

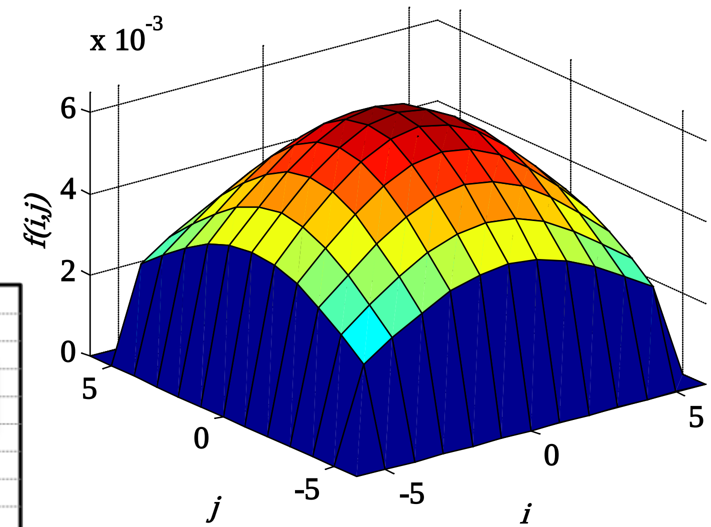
$$\rho_{ij} = \sum_{n_1=-l}^{l} \sum_{n_2=-l}^{l} d_{i+n_1,j+n_2} \times f(n_1, n_2)$$



Discretized Gaussian function

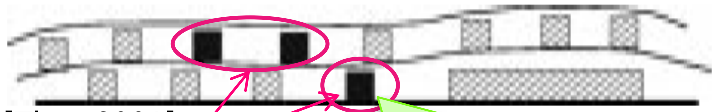
$$f(i,j) = \frac{1}{2\pi l^2} \exp\left(-\frac{i^2+j^2}{2l^2}\right)$$

Weighting function with l=5



Reduce CMP variation

- Add dummy fill



[Tian, 2001]

Dummy feature

Minimize dummy fill

Reduce EPD variation, and ILD thickness variation

Increase routing cost or coupling capacitance

e.g., floating dummy fill may increase the coupling capacitance up to 90% [Kahng, 2006]

- Dummy filling problem formulation
  - Insert minimal amount of dummy fill (D) to make

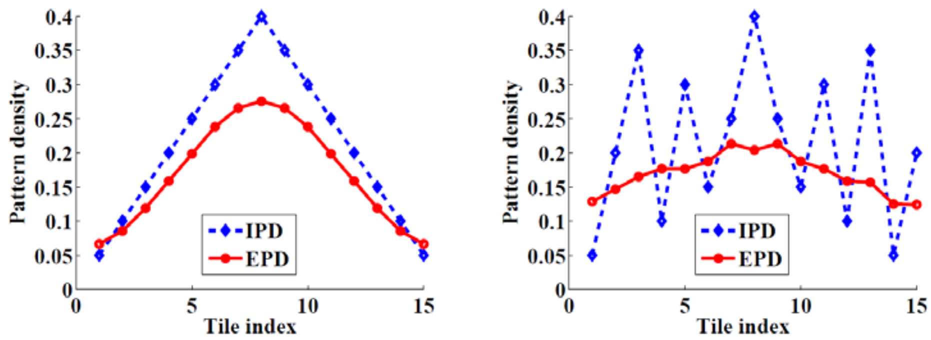
$$\mathrm{ran}(\rho) = \rho_H - \rho_L \leq \epsilon$$

(ρ = EPD = effective pattern density)

  - D is a good metric
- Use dummy filling algorithm in [Tian, 2001]
- CMP-aware routing
  - Generate more uniform distribution of wires
  - Reduce variation and D

Previous work and our contributions

- Previous work of CMP-aware routing
  - No CMP model used
    - Control the maximal initial pattern density [Li, 2005]
  - The metrics used have limitations
    - Minimize the gradient of initial pattern density (IPD) [Chen, 2009]



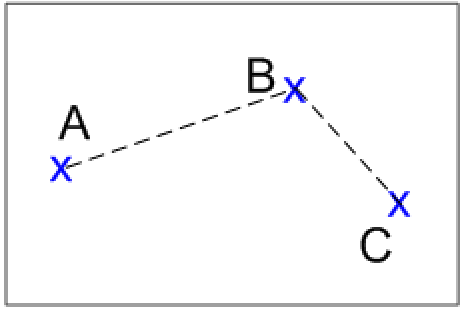
(a) Layout I: IPD gradient ≤ 0.05. EPD variation = 0.209.
(b) Layout II: IPD gradient ≥ 0.10. EPD variation = 0.089.

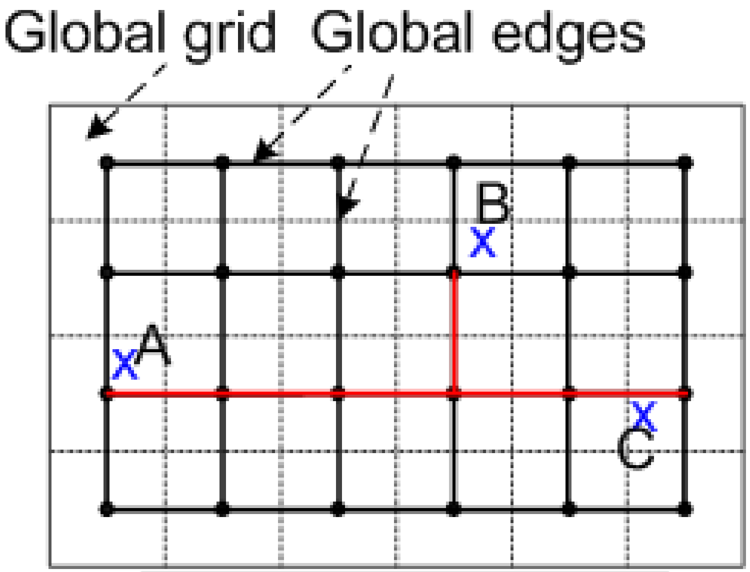
EPD
=
effective pattern density

  - Impact of routing on the neighboring regions not considered [Yao, 2007][Jia, 2008]
- Our contributions
  - Use accurate CMP model and metric D (D = amount of required dummy fill)
  - Optimize dummy fill directly

Global routing model

- Input
  - Information for nets/interconnects
- Output
  - Paths on the global routing graph
- Objective: minimize overflow, wire length and amount of dummy fill D





Global routing graph

Outline

- Introduction
- Flow of the routing algorithm
- Cost functions
- Experimental results
- Conclusion

Algorithm flow

- Based on NTHU-Route 2.0 (NTHR) [Chang, 2008]

NTHR's flow

1. Initial stage
2. Main stage
3. Refinement stage
4. Layer assignment

Our flow

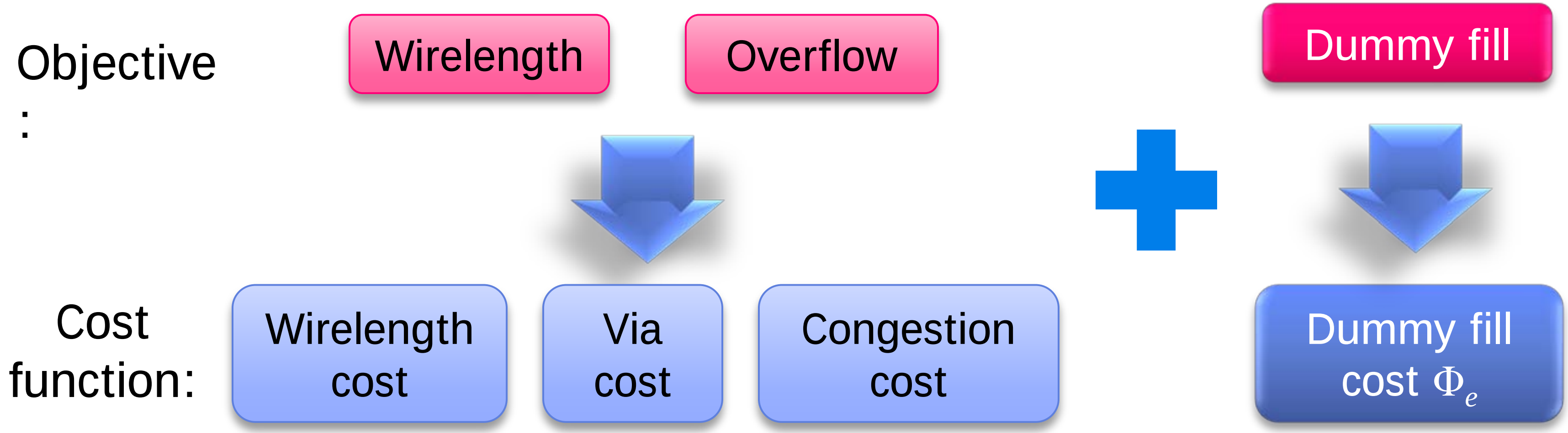
1. Initial stage
2. Main stage with **dummy fill cost function**
3. Refinement stage with **dummy fill cost function**
4. EPD postprocessing: RRR the nets passing through tiles related to ρ_H
5. Layer assignment

Outline

- Introduction
- Flow of the routing algorithm
- Cost functions
- Experimental results
- Conclusion

Big picture of cost function

- Achievement of objectives dependent on cost functions



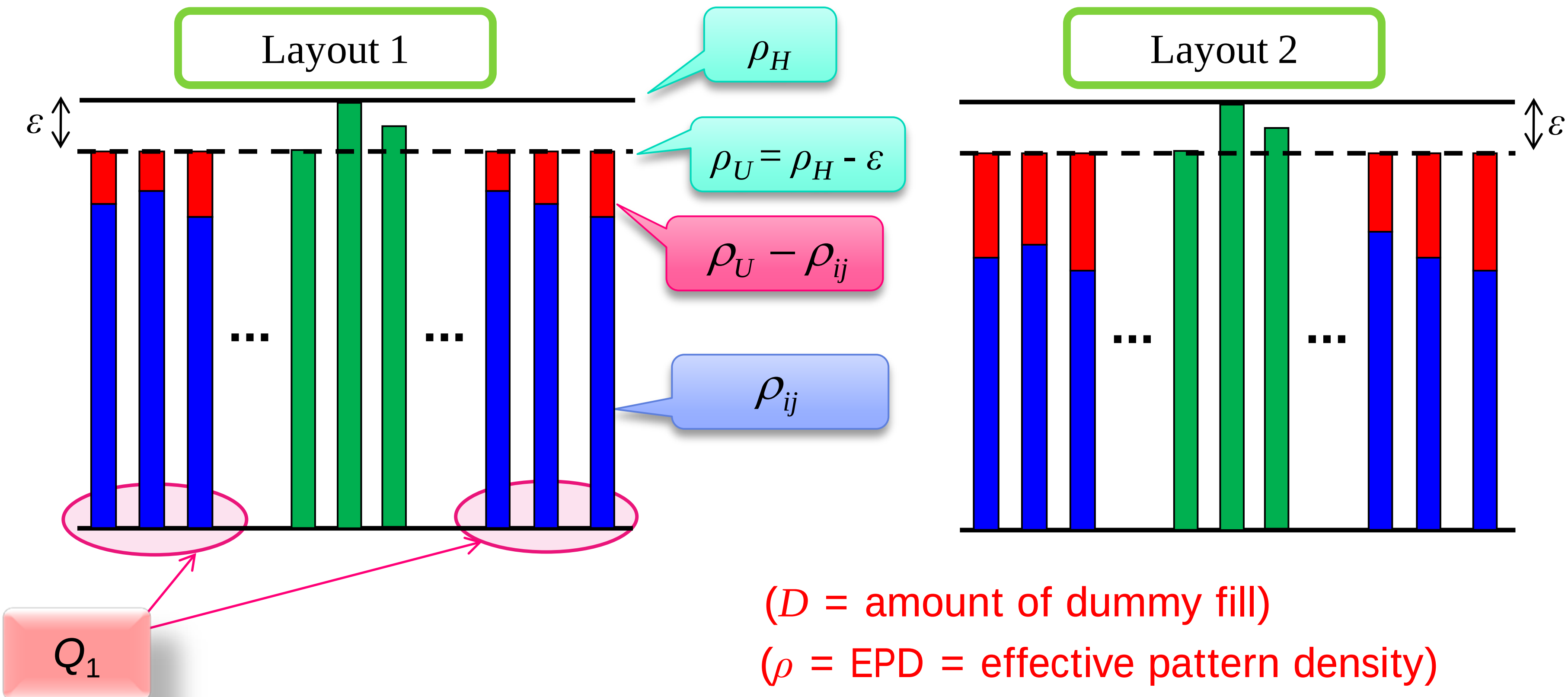
- Dummy fill cost Φ_e
  - Efficient
  - Effective
- Metric: D (*D* = amount of dummy fill)
  - Linear programming used
  - Several minutes required to compute D

Metric Γ: A surrogate for *D*

- Metric Γ:

$$\Gamma = \sum_{t_{ij} \in Q_1} (\rho_U - \rho_{ij})$$

Γ approximate indicator of amount of dummy fill

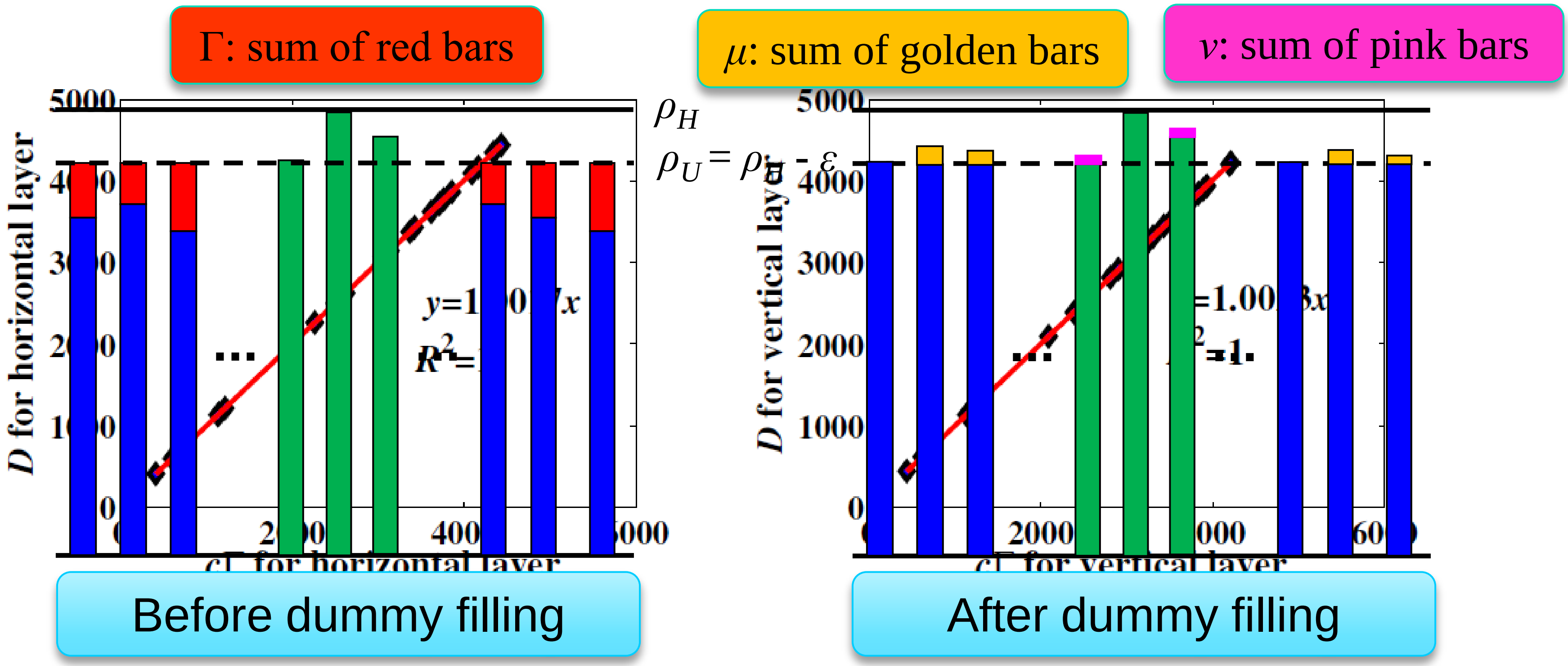


(D = amount of dummy fill)

(ρ = EPD = effective pattern density)

Metric Γ: A surrogate for *D*

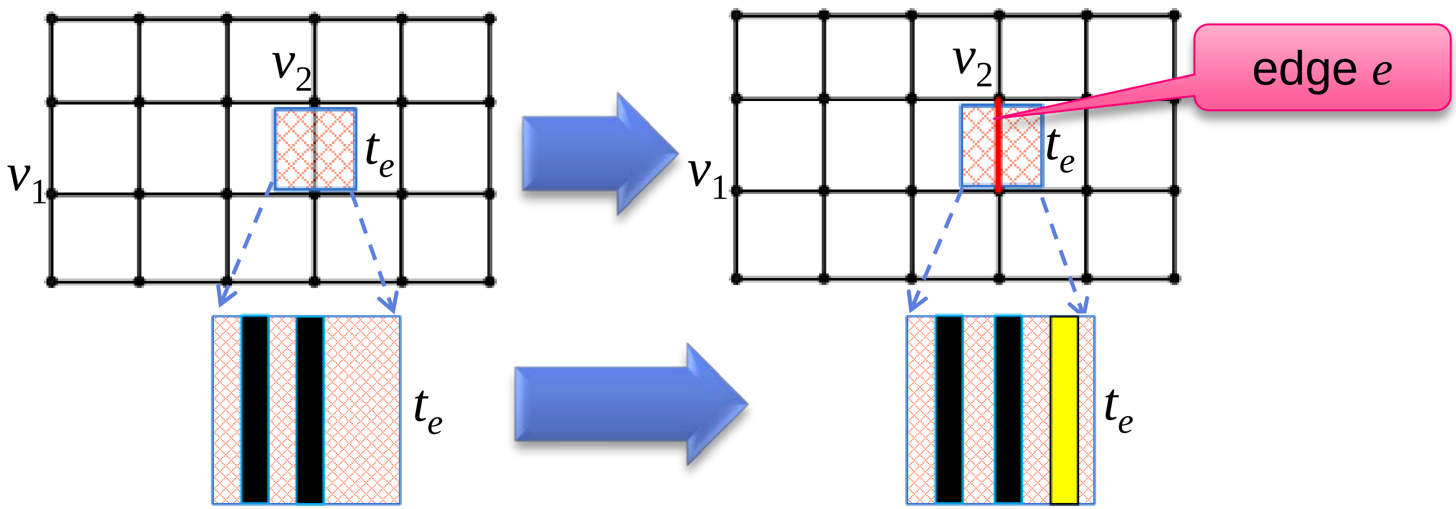
- Theorem 1: $D = c\,(\Gamma + \mu + \nu)$, where *c* is constant



- $\mu+\nu$ much smaller than Γ
  - e.g., for circuit newblue2, $(\mu + \nu)\ :\ \Gamma\ =\ 1\ :\ 483$
- Important conclusion: $D \approx c\Gamma$ (*D* = amount of dummy fill)

A surrogate for Γ

- Γ still too complex to be used
- The impact of routing on Γ and D (D = amount of dummy fill)
- Theorem 2:



$$\Delta\Gamma \geq |Q_1|\,\Delta\rho_H - c_1$$

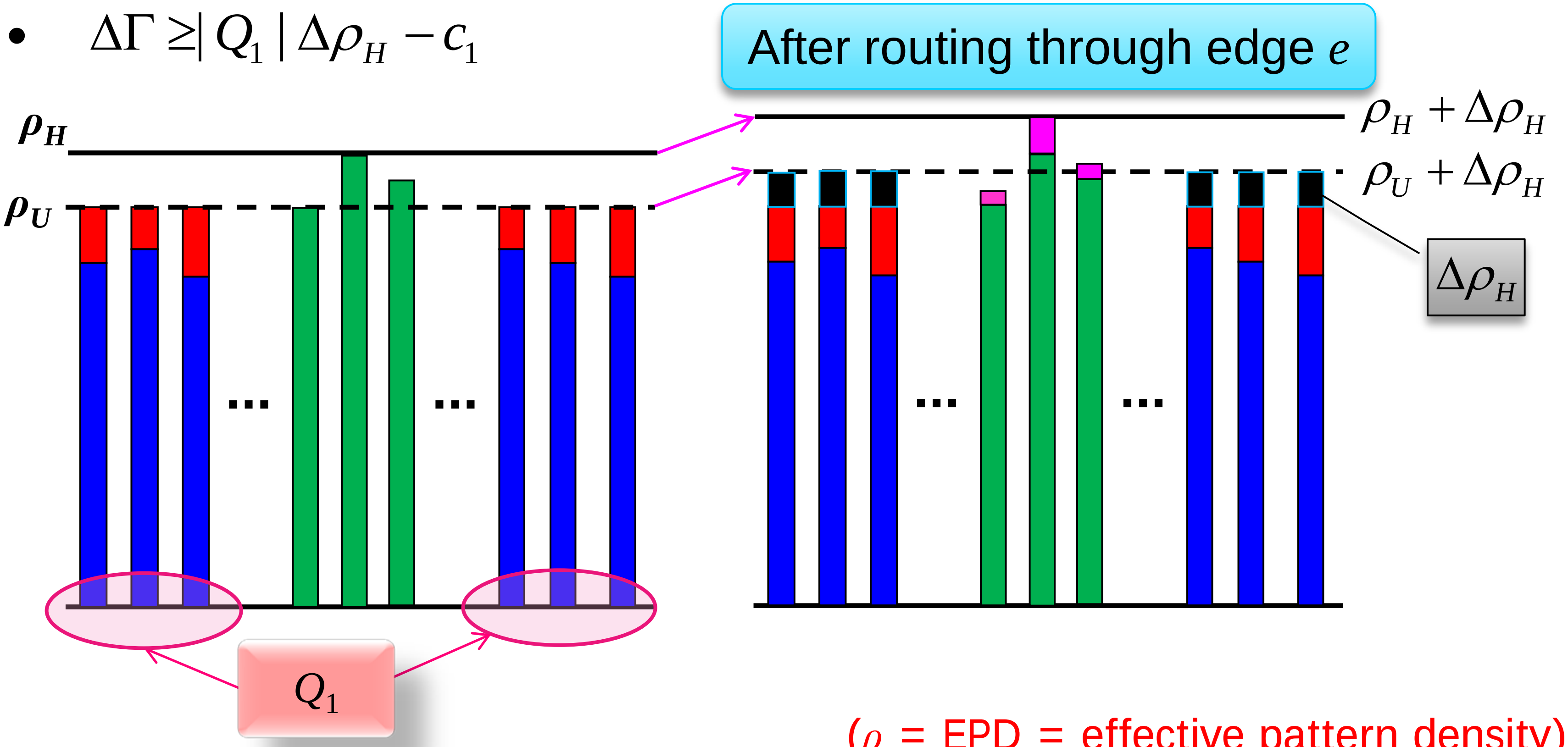
(c_1 = a constant)

(ρ_H = maximal effective pattern density)

(Q_1 = set of blue tiles)

A surrogate for Γ

- $\Delta\Gamma \geq |Q_1|\,\Delta\rho_H - c_1$



(Q_1 = set of blue tiles)

(ρ = EPD = effective pattern density)

(D = amount of dummy fill)

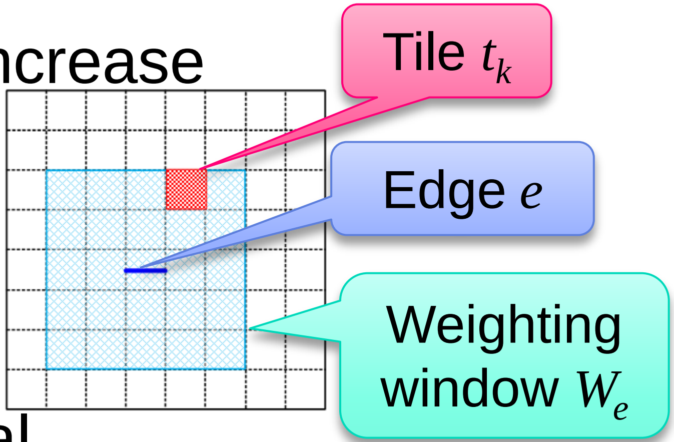
- $|Q_1|$ is about 85% of the total number of tiles
- D highly sensitive to the change in ρ_H ($D \approx c\Gamma$)
- Minimizing ρ_H is a good surrogate for minimizing D

Cost function

- Compute dummy fill cost Φ_e
  - Purpose: constrain the increase of ρ_H

- Cost component 0: penalty for direct increase of ρ_H after routing through edge e

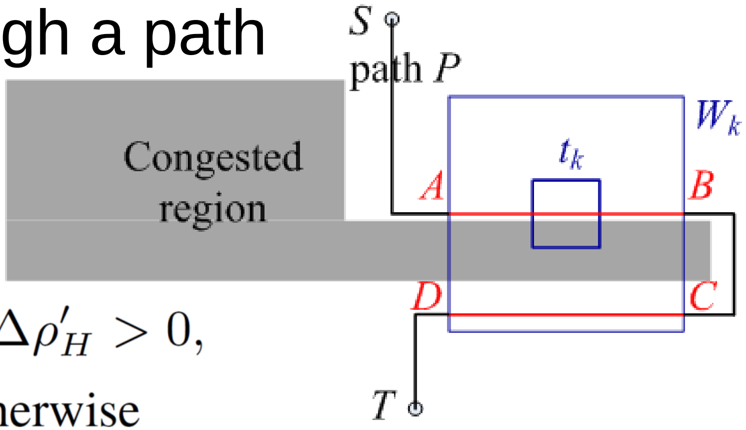
$$\Omega_e = \begin{cases} \exp\left(\frac{\Delta\rho_H}{\Delta_\rho^M}\right) & \text{if } \Delta\rho_H > 0, \\ 0 & \text{otherwise} \end{cases}$$



- Cost component 1: penalty for potential increase of ρ_H after routing through a path

$$\Delta\rho'_H = \max_{t_k \in W_e} (\rho_k + J_M - \rho_H)$$

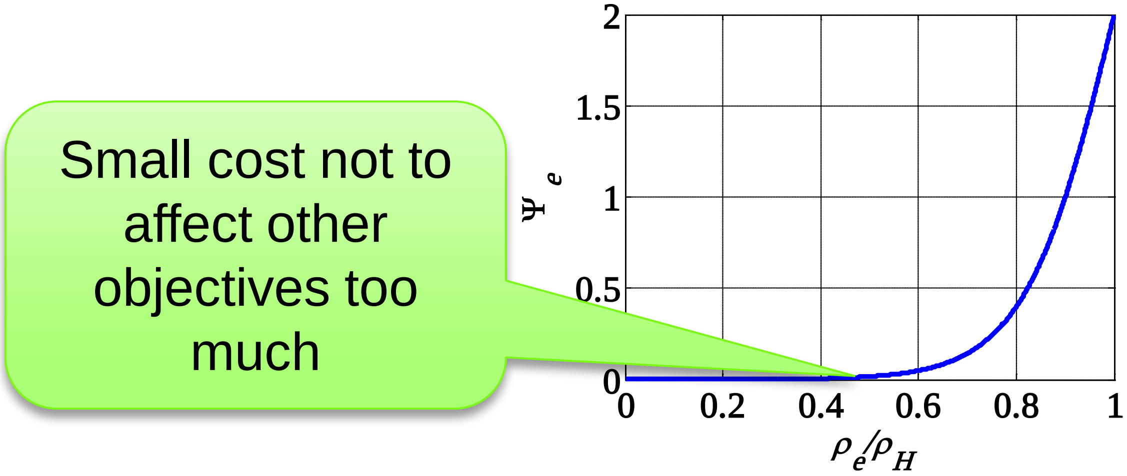
$$\Theta_e = \begin{cases} \exp\left(\frac{\Delta\rho_k}{\Delta_\rho^M}\right) \cdot \exp\left(p_0 \cdot \frac{\Delta\rho'_H}{\Delta_\rho^M}\right) & \text{if } \Delta\rho'_H > 0, \\ 0 & \text{otherwise} \end{cases}$$



Cost function

- Cost component 2: penalty for routing through the edges in the tiles with large EPD

$$\Psi_e = \frac{p_1}{1 + e^{p_2(1-\rho_e/\rho_H)}}$$



- Total dummy fill cost: $\Phi_e = \Psi_e + \Theta_e$
- Integrate dummy fill cost function into the original router

Outline

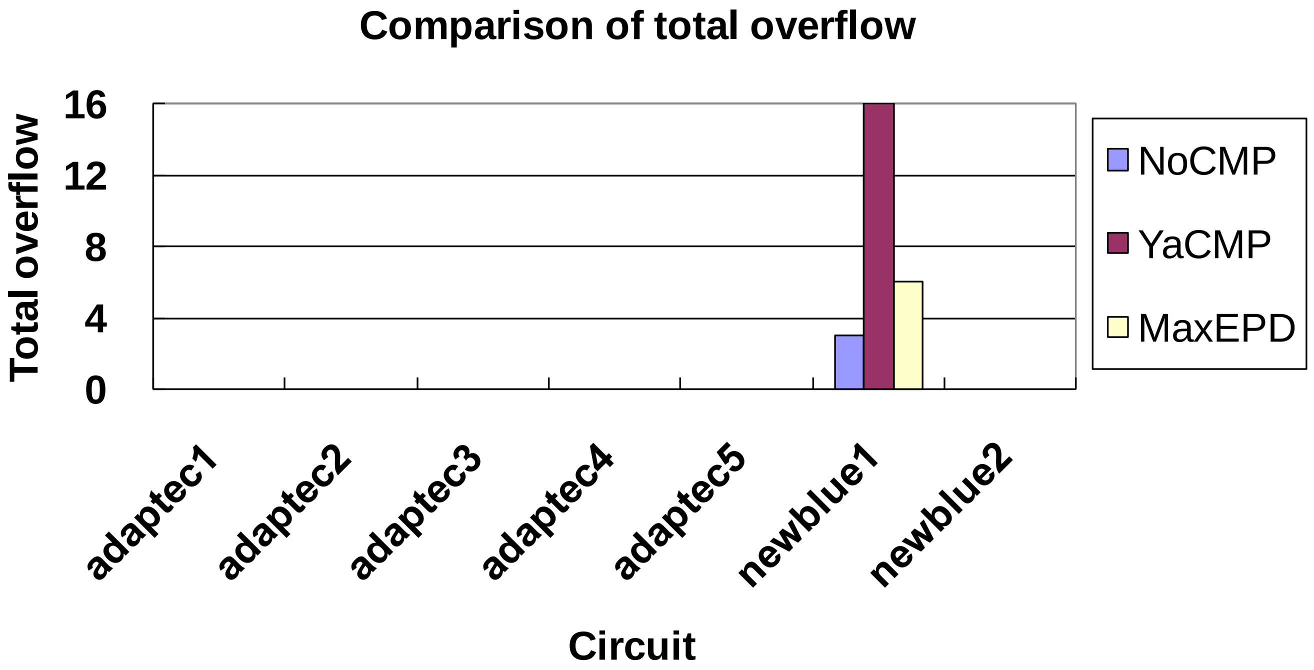
- Introduction
- Flow of the routing algorithm
- Techniques to optimize dummy fill
- Experimental results
- Conclusion

Experimental setup

- Platform
  - A 64-bit Linux machine with an Intel Core(TM)2 Duo 3.00GHz CPU and 8GB memory
- Major assumptions
  - 0.13um technology
  - Size of weighting window 1mm
  - The space between the dies on the wafer is negligible
- ISPD 2007 benchmark
- Comparison
  - NoCMP: the original NTHU router
  - MaxEPD: our algorithm
  - YaCMP: replacing the cost function we propose in MaxEPD by the cost function in [Yao, 2007] (our implementation)
- Parameters tuned using newblue2

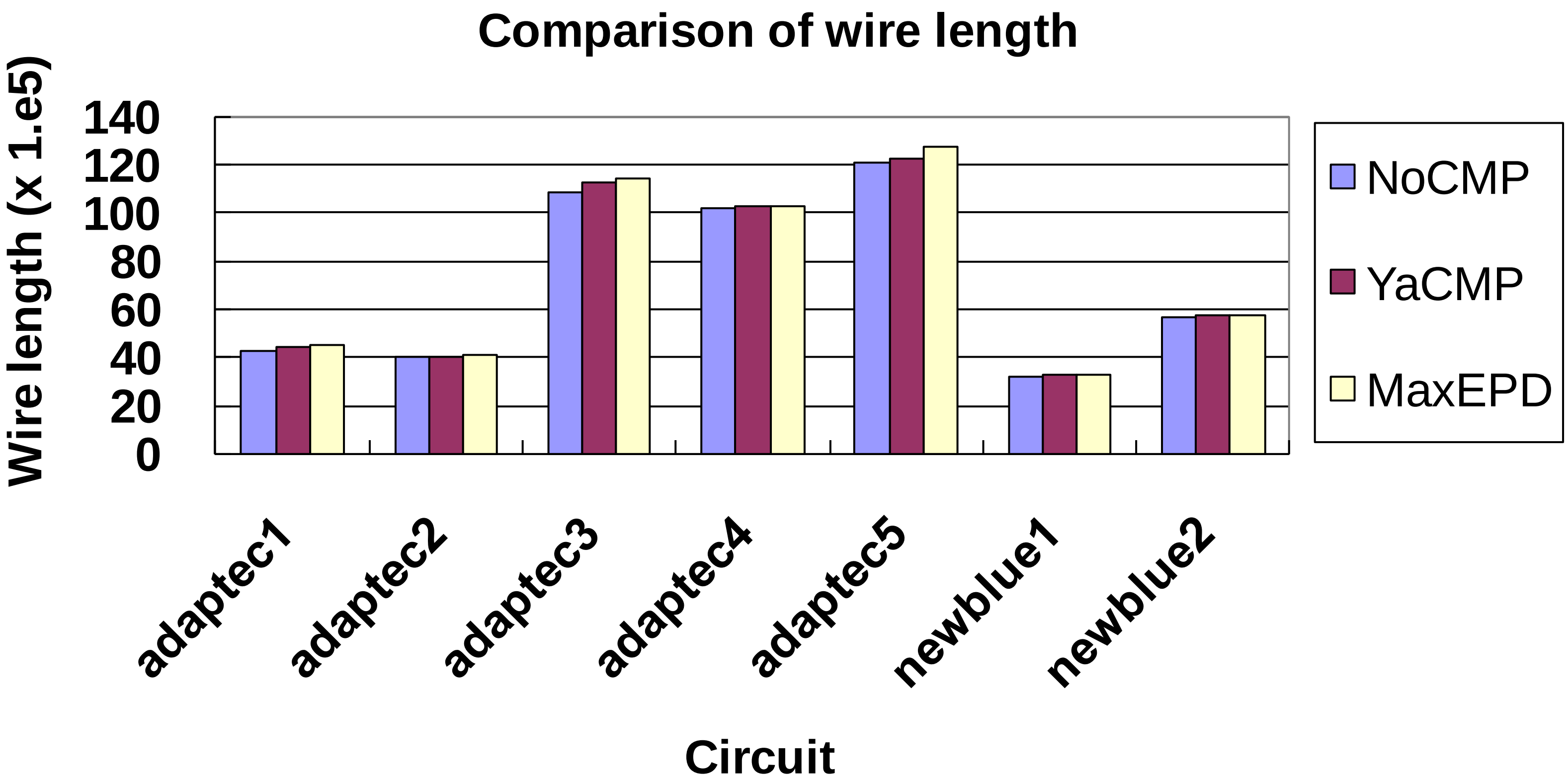
Routing results

- Overflow
  - Circuit newblue1 is difficult
  - Maximal overflow is 1
  - The overflows can be eliminated in later stage with the reserved capacity



Routing results

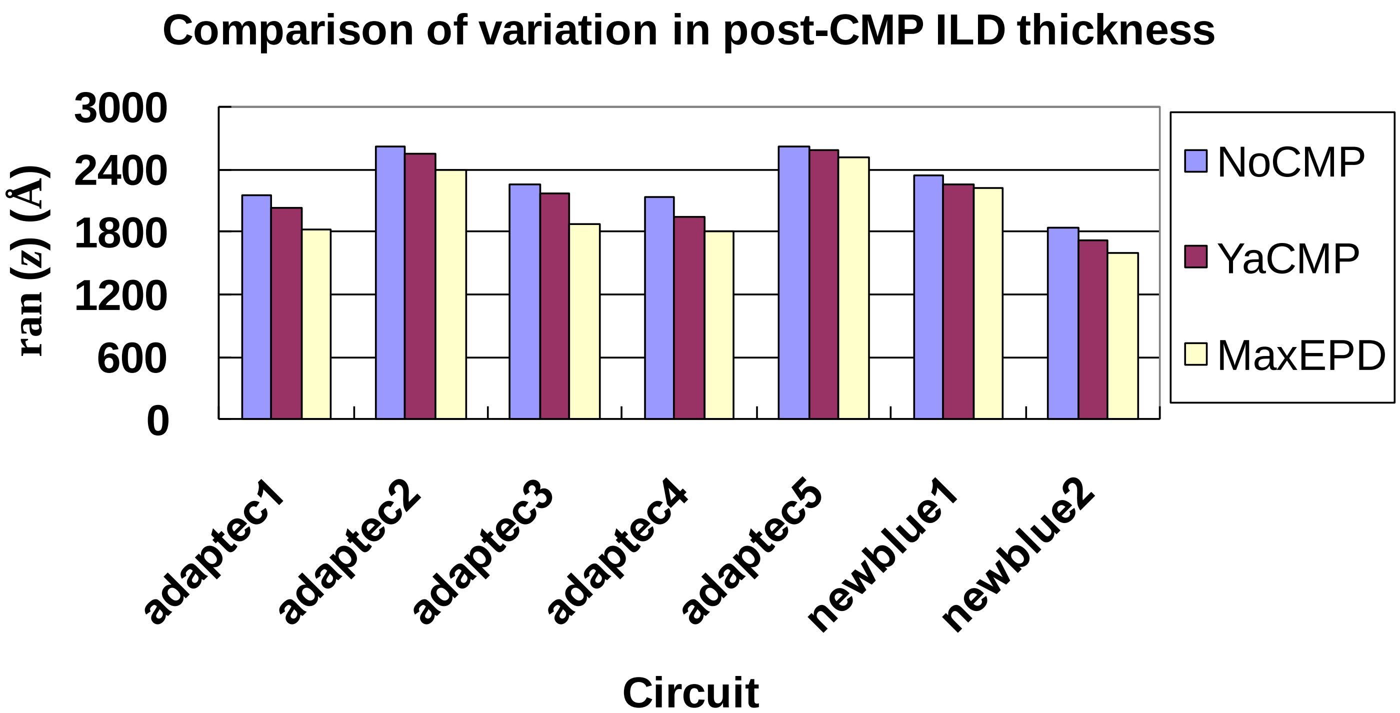
- Wire lengths are similar



Routing results

- Variation in the post-CMP ILD thickness

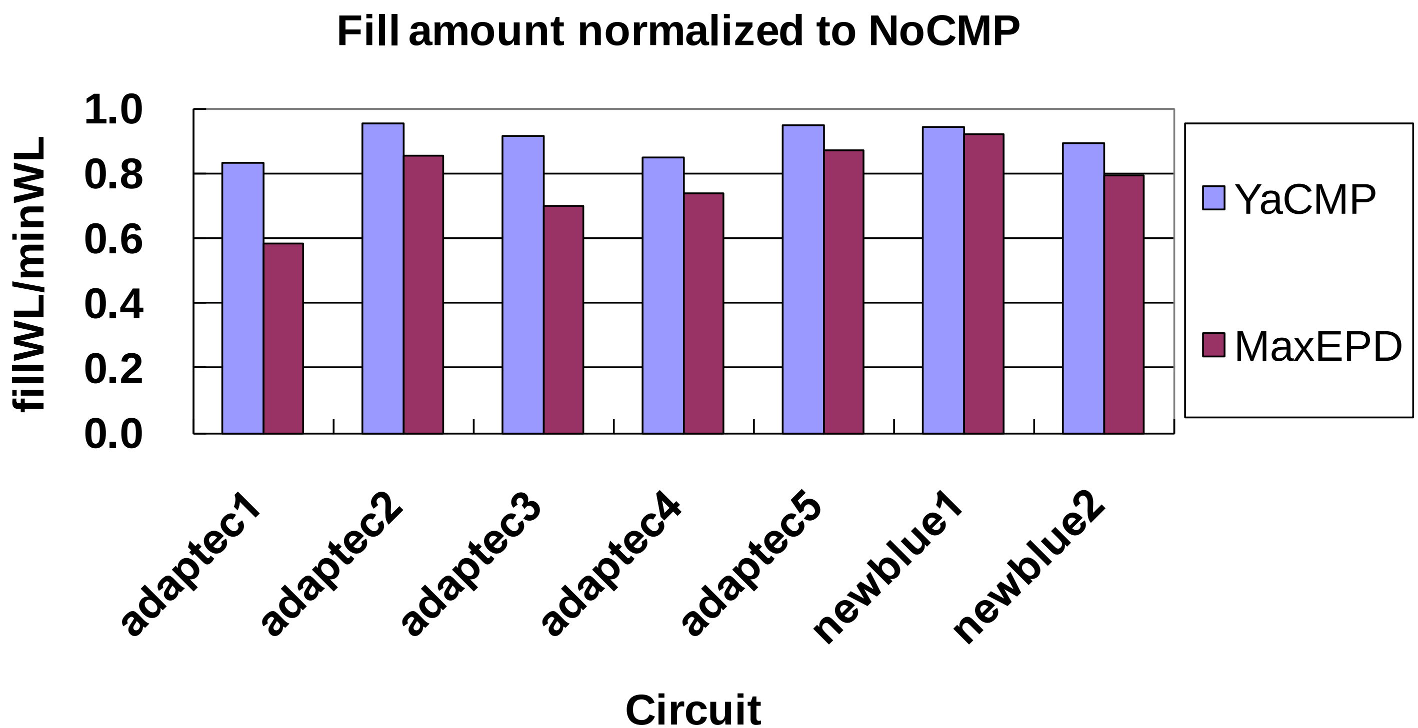
$$\mathrm{ran}(z) = z_1 \cdot \mathrm{ran}(\rho) \quad \text{with } z_1\text{=7000 Å}$$



Compared to NoCMP, reduced by 16.7% (max) and 11.1% (average);
Compared to YaCMP, reduced by 13.5% (max) and 7.0% (average).

Dummy fill results

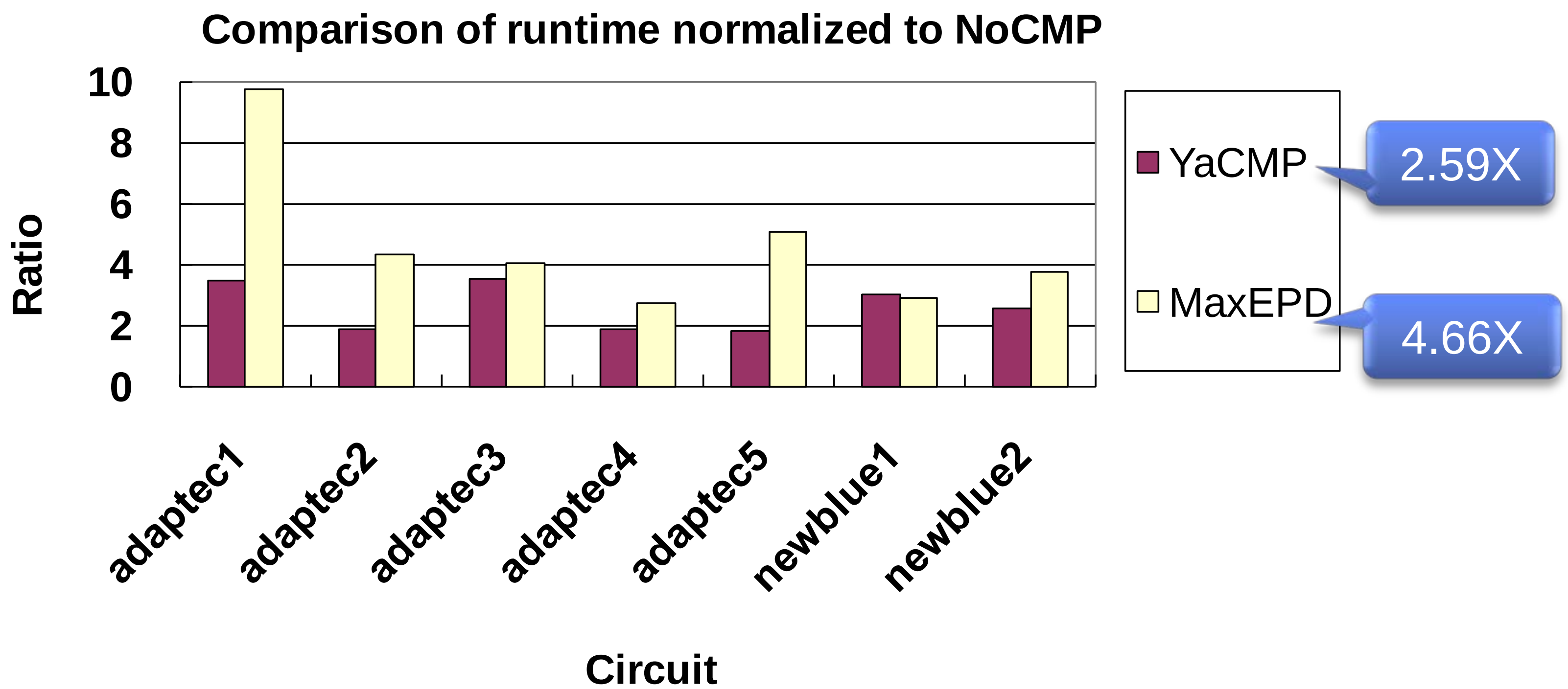
- Amount of dummy fill D
  - Computed using dummy filling algorithm in [Tian, 2001]



Compared to NoCMP, reduced by 41.5% (max) and 22.0% (average);
Compared to YaCMP, reduced by 23.6% (max) and 14.1% (average).

Routing results

- Runtime



- Runtime in absolute terms for MaxEPD
  - Less than 1 hour
  - Reasonable

Conclusion

- Develop a global routing algorithm optimizing the amount of dummy fill.
- Minimizing maximal effective pattern density is a good surrogate for minimizing the amount of dummy fill.
- Propose effective cost functions to perform dummy fill optimization.
- Experimental results show that our algorithm can reduce the amount of dummy fill up to 41.5% with ~4x runtime overhead, compared with original router.

Thank You!

Any questions?

Algorithm flow

- Based on NTHU-Route 2.0 (NTHR) [Chang, 2008]

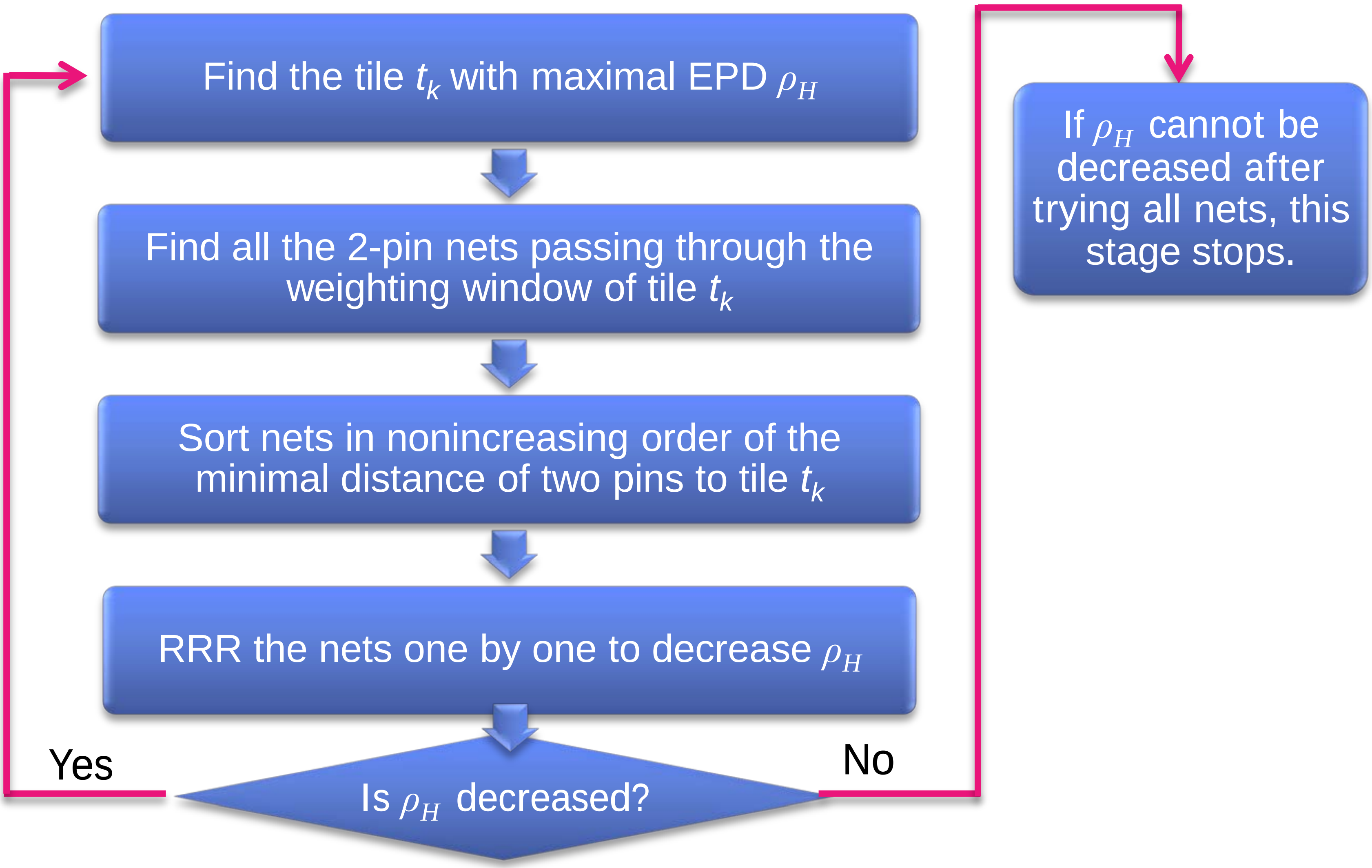
NTHR's flow:

1. Initial stage
2. Main stage
3. Refinement stage
4. Layer assignment

Our flow:

1. Initial stage
2. Main stage with **dummy fill cost function**
3. Refinement stage with **dummy fill cost function**
4. EPD postprocessing: RRR the nets passing through tiles related to ρ_H
5. Layer assignment

EPD postprocessing stage



Why not care about ρ_L

- The change in D after routing through edge e
- $\Delta\rho_H > 0$, we will not use this edge
- $\Delta\rho_H = 0$, maximal ΔD is -1 wire track
  - Many choice of edges
  - No matter how ρ_L changes
  - No need to make efforts to increase ρ_L
- Similar argument for routing through a path