MORPH: More Robust ASIC Placement for Hybrid Region Constraint Management

Jing Mai[1], Zuodong Zhang[1], Yibo Lin[1,2], Runsheng Wang[1,2], Ru Huang[1,2]

[1]Peking University, Beijing, China

[2]Beijing Advanced Innovation Center for Integrated Circuits, Beijing, China

jingmai@pku.edu.cn

Oct. 28, 2024

IEEE/ACM 2024 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN ICCAD 43rd Edition

Outline

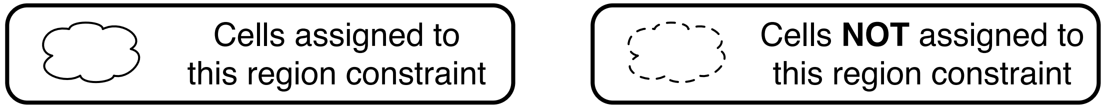
1. Introduction
2. Preliminaries
3. The `MORPH` Algorithm
4. Experimental Results
5. Conclusion & Future Work

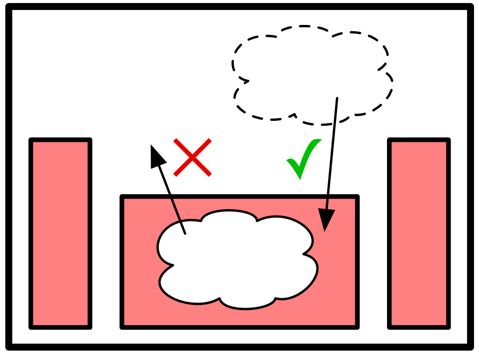
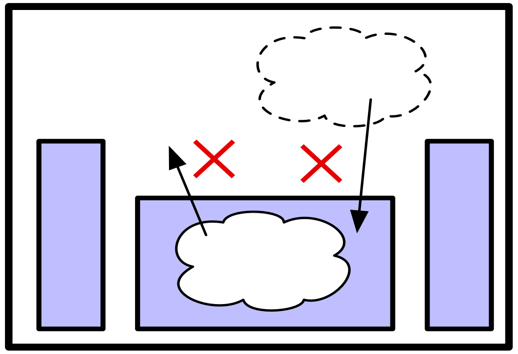
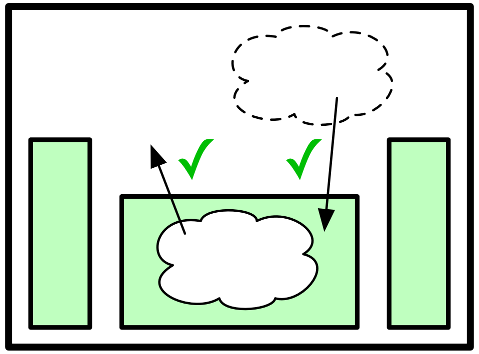
Introduction

Background

Region constraint in ASIC CAD

- An essential feature provided by modern ASIC CAD tools.
- Three categories of region constraints: **default regions**, **fence regions**, and **guide regions**.

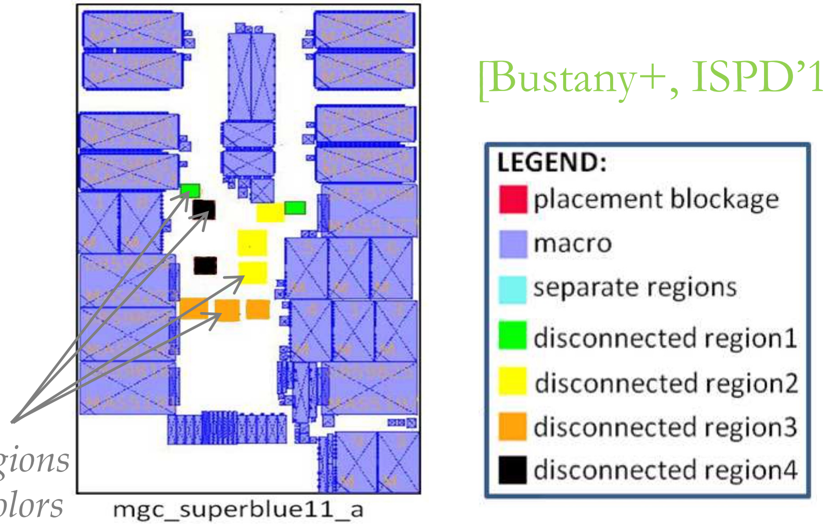


Category	Default Region	Fence Region	Guide Region
			
hard / soft constraint?	Hard All cells subject to this constraint must be placed inside the region.	Hard All cells subject to this constraint must be <u>exclusively</u> placed inside the region.	Soft Other constraints such as wirelength, can override this soft constraint.

Prior Work & Challenges

Space disconnectivity

- A region is made up of multiple spatially disjoint rectangular subregions.
- Fence-region-aware clustering technique. [Huang+, TCAD'17]
- Look-ahead region-aware rough legalization. [Darav+, TODAES'16] [Chow+, SLIP'17]
- Multi-electrostatics-based placement model. [Gu+, ICCAD'20]



Four fence regions in different colors

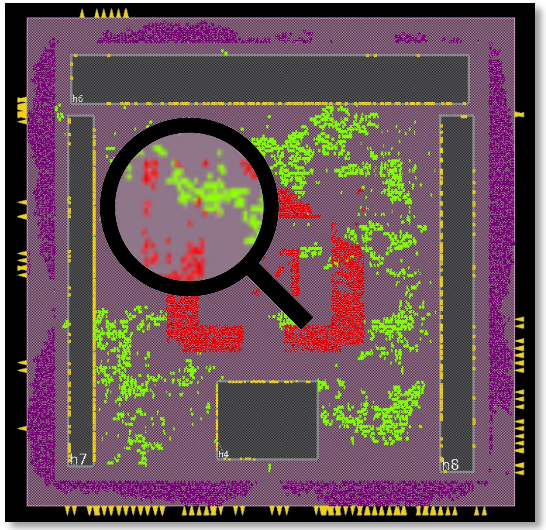
[Bustany+, ISPD'15]

Hybrid region constraints

- None of the previous work considered default regions or guide regions.
- Multi-functionality requirement.

Robustness issue

- Non-orthogonality for instances and their placeable areas (for default regions)
- Multi-region-aware placement can easily fall into local optimum and even diverge. [Gu+, ICCAD'20]



Non-orthogonality: Cells not subject to any default region constraint can be placed within default regions.

Our Contributions

MORPH: a more robust multi-electrostatics-based placement algorithm for hybrid region constraints.

- shared electrostatics model → a unified electrostatics formulation for default regions and fence regions.
- A wirelength-prioritized penalty method → balanced guide region optimization without compromising wirelength minimization.
- A modified Nesterov's accelerated LBFGS algorithm with second-order information → significantly improve the solution quality and the stability with minor runtime overhead.
- Experimental results on the ISPD 2015 benchmarks [Ismail+, ISPD'15] demonstrate that our proposed algorithm can achieve 5.6-14.3% HPWL improvement and 10-24% overflow reduction when compared to other SOTA fence region-aware placers.

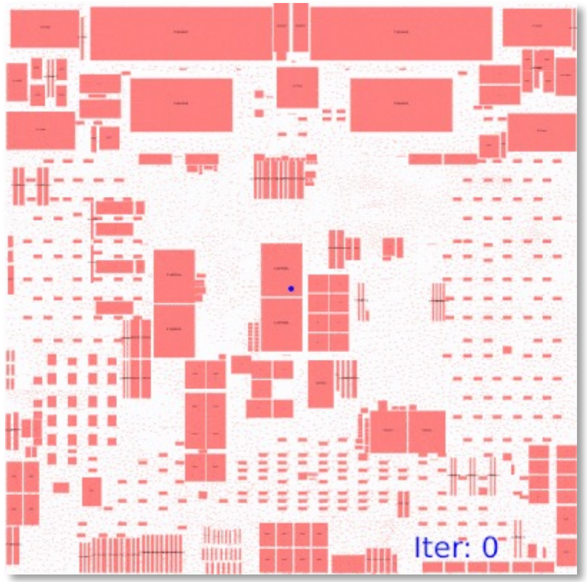
Preliminaries

Electrostatics-based Density Model

Analogy between placement density and electrostatic system `ePlace` [Lu+, TCAD’15]

Placement	Electrostatic System	
Cell	Positive Charge	
Site	Fixed Negative Charge	
Blockage / Fixed Macro	Fixed Positive Charge	
Cell Density	Charge Density	ρ
Cell Density Penalty	Electric Potential Energy	Φ
Cell Density Gradient	Electric Field	

Charge Density Map Visualization



`DREAMPlace` [Lin+, DAC’19]

Multi-Electrostatics-based Placement

Optimize wirelength while adhering to multiple density constraints, [Gu+, ICCAD'20]

$$\min_{\boldsymbol{x},\boldsymbol{y}} \widetilde{W}(x,y)$$
$$s.t.\;\; \Phi_s\left(x^{(s)}, y^{(s)}\right) \le \hat{\Phi}, \forall s \in S,$$

where S is the set of electrostatic systems.

WAWL Wirelength Model

Weighted-average (WA) wirelength model $\widetilde{W}(x,y)$ approximates half-perimeter wirelength (HPWL), [Ray+, DATE'13]

Augmented Lagrangian Method

Transfer the constrained problem into an unconstrained one, [Zhu+, DAC'18]

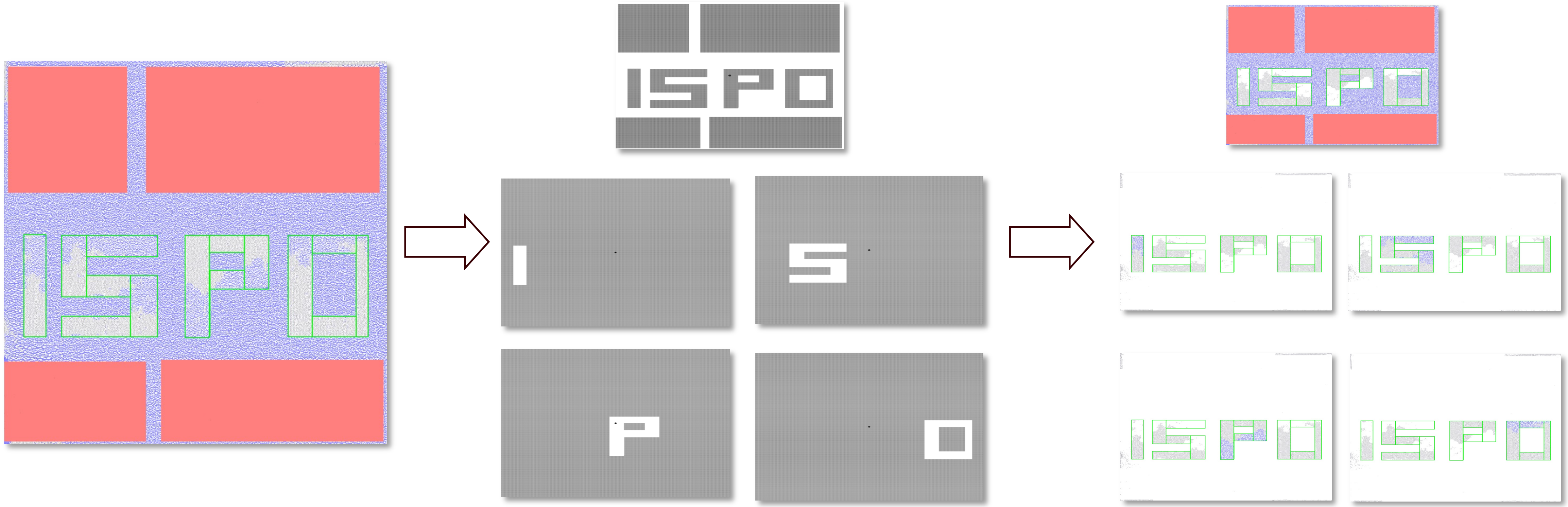
$$\min_{\boldsymbol{x},\boldsymbol{y}} \widetilde{W}(x,y) + \sum_{s\in S} \lambda_s \mathcal{D}_s$$
$$s.t.\, \mathcal{D}_s = \Phi_s + \frac{1}{2}\mathcal{C}_s {\Phi_s}^2, \forall s \in S.$$

Multi-Electrostatics-based Placement (cont'd)

Multi-Electrostatics-based Density Model

- Each fence region has a separate electrostatic system.
- #electrostatic systems = 1 + #fence regions.
- Insert artificial blockages in each electrostatic system (ES) -> block illegal areas.

Placement | **Artificial Blockages in each ES** | **Charge Density Map Visualization**



* purple: cells, red: macros, gray: artificial blockages in each electrostatic system.

The MORPH Algorithm

Multi-electrostatics-based placement for hybrid regions

Minimize wirelength while subject to default regions, fence regions, and guide regions constraints,

$$\min_{\boldsymbol{x},\boldsymbol{y}} \mathcal{L}(\boldsymbol{x},\boldsymbol{y}) = \widetilde{W}(\boldsymbol{x},\boldsymbol{y}) + \boxed{\sum_{k=0}^{K_1-1} \lambda_k \mathcal{D}_k} + \boxed{\sum_{k=K_1}^{K_1+K_2-1} \eta_k \Gamma_k(\boldsymbol{x}^{(k)},\boldsymbol{y}^{(k)}),}$$

$$\mathcal{D}_k = \Phi_k(\boldsymbol{x}^{(k)},\boldsymbol{y}^{(k)}) + \frac{1}{2}\mathcal{C}_k\Phi_k^2(\boldsymbol{x}^{(k)},\boldsymbol{y}^{(k)}),$$

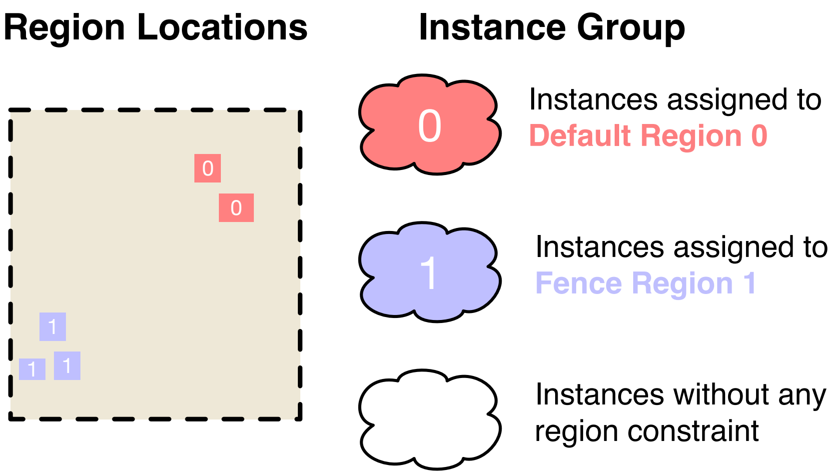
- Default regions & fence regions: **shared electrostatic model**
- Guide regions: **wirelength-prioritized penalty method**

Unified Multi-Electrostatics Formulation

Shared Electrostatics Model

- Formulate default regions and fence regions as a **unified** multi-electrostatics placement formulation.
- #electrostatic systems = 1 + #default regions + #fence regions.

Example

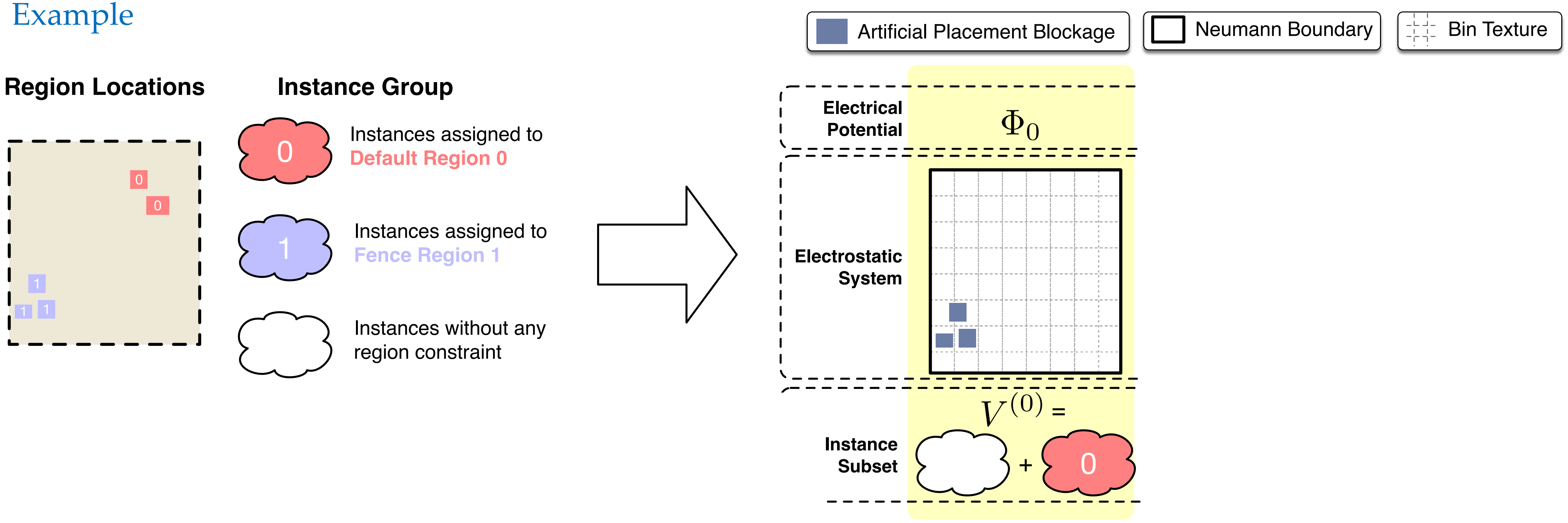


Unified Multi-Electrostatics Formulation

Shared Electrostatics Model

- Formulate default regions and fence regions as a **unified** multi-electrostatics placement formulation.
- #electrostatic systems = 1 + #default regions + #fence regions.

Example

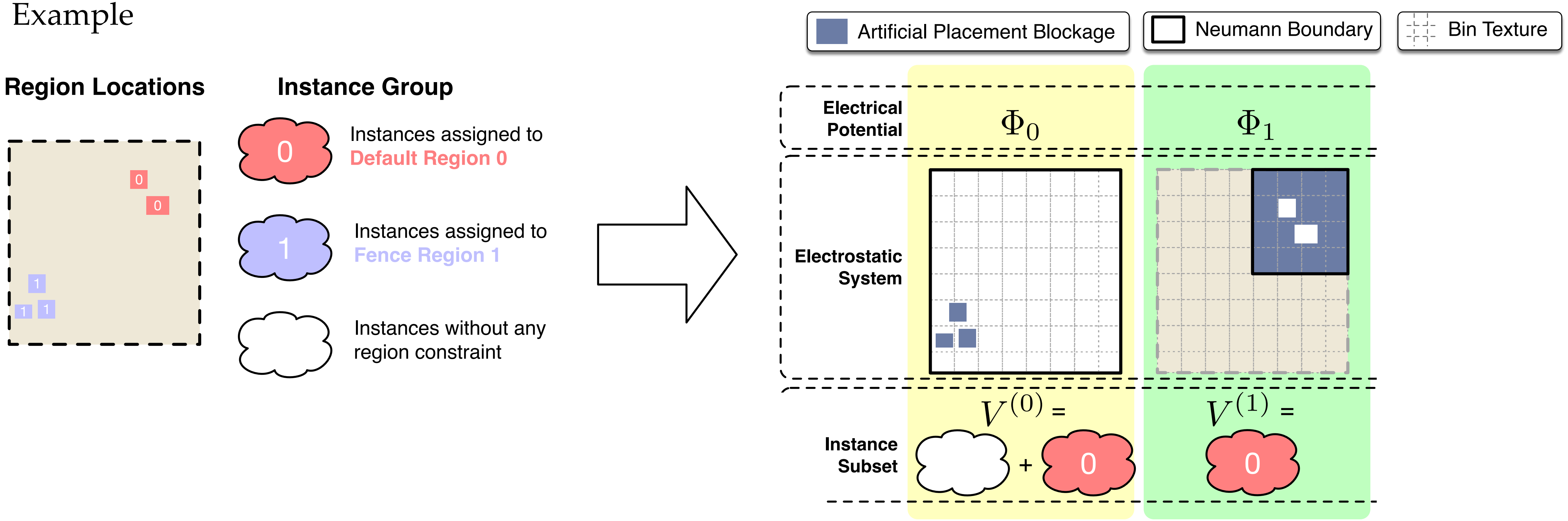


Unified Multi-Electrostatics Formulation

Shared Electrostatics Model

- Formulate default regions and fence regions as a **unified** multi-electrostatics placement formulation.
- #electrostatic systems = 1 + #default regions + #fence regions.

Example

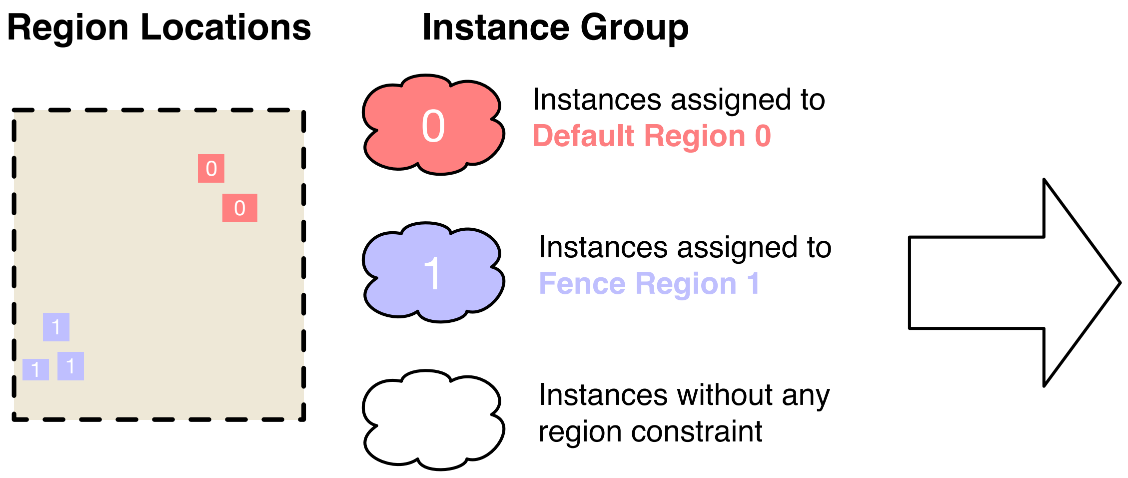


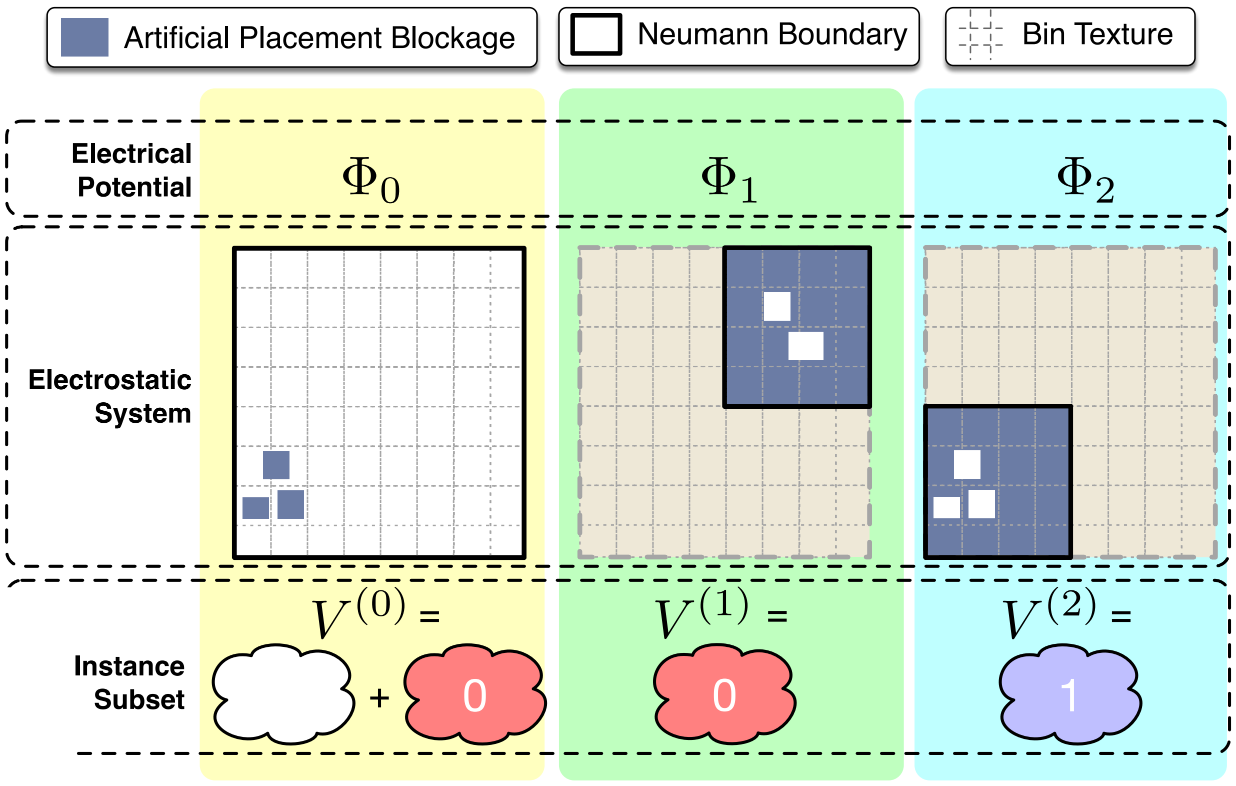
Unified Multi-Electrostatics Formulation

Shared Electrostatics Model

- Formulate default regions and fence regions as a **unified** multi-electrostatics placement formulation.
- #electrostatic systems = 1 + #default regions + #fence regions.

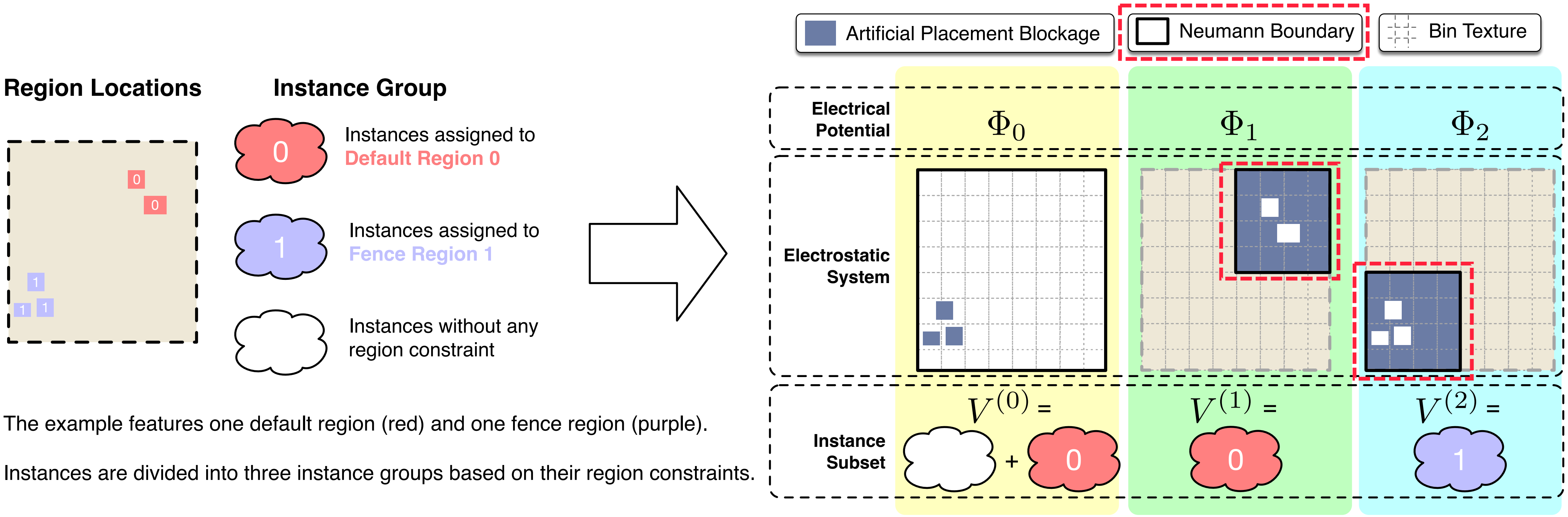
Example





Binary-lifting-based region pruning algorithm

- Trim redundant areas away from the region by binary-lifting algorithm.
- Cells in each electrostatics system can only moved within the the trimmed **Neumann boundary**.
- Reduce memory complexity in proportion to the total area of the regions $O(\sum_i A_i)$. **(lower bound of complexity)**



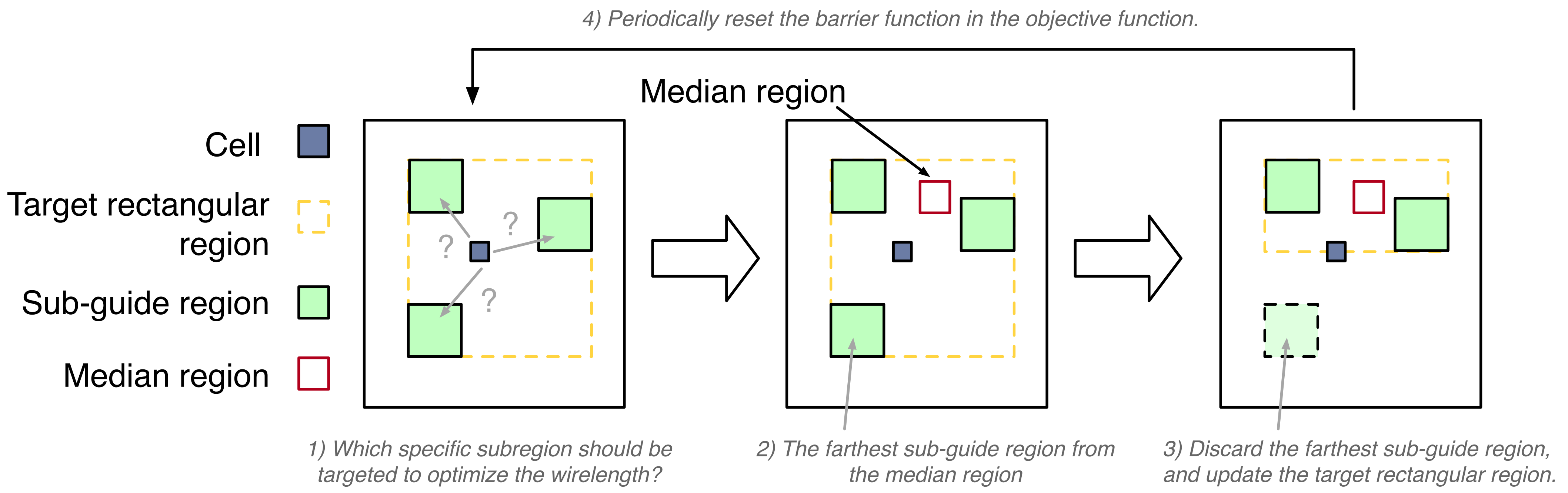
The example features one default region (red) and one fence region (purple).

Instances are divided into three instance groups based on their region constraints.

Wirelength-Prioritized Penalty Method

Tradeoff between wirelength and guide region constraints

- A barrier function to guide each cell to its target rectangular region.
- Target rectangular regions are updated periodically to discard the subregion that is farthest from the median region [Pan+, ICCAD'05]
  - i.e., the optimal region for each cell.



LBFGS Algorithm (Quasi-Newton Method)

Analytical placement problems are transformed into unconstrained optimization problems:

$$\min_{x} f(x).$$

Gradient vector (first-order): $g^{(k)} = \nabla f\,(x^{(k)})$; Hessian matrix (second-order): $H^{(k)} = \nabla^2 f\,(x^{(k)})$

Target: approximate the inverse Hessian matrix in Newton's Method:

$$x^{(k+1)} = x^{(k)} - \left[H^{(k)}\right]^{-1} g^{(k)}$$

1. Define solution difference $s_{k-1} = x^{(k)} - x^{(k-1)}$ and gradient difference $y_{k-1} = g^{(k)} - g^{(k-1)}$,

$$y_{k-1} = \nabla^2 f\left(x^{(k)}\right) s_{k-1} + \mathcal{O}(\|s_{k-1}\|^2).$$

2. Approximate the inverse Hessian matrix $T^{(k)} \approx \left[H^{(k)}\right]^{-1} = \left[\nabla^2 f\left(x^{(k)}\right)\right]^{-1}$,

$$s_{k-1} = T^{(k)} y_{k-1}.$$

3. Leverage the the recursive property of $T^{(k)}$ and truncate after m iterations to reduce memory,

$$T^{(k)} \xLongleftarrow{derive\ from} T^{(k-1)} \Leftarrow \ldots \Leftarrow T^{(k-m)} = \left[H^{(k-m)}\right]^{-1} \approx \frac{y_{k-1}^T s_{k-1}}{y_{k-1}^T y_{k-1}} I.$$

A Modified Nesterov's Accelerated LBFGS Algorithm

- Combine Nesterov's acceleration technique with LBFGS to speedup convergence.

Algorithm 3 A Modified Nesterov's Accelerated LBFGS Algorithm

Input: major solution $u^{(k)}$, reference solution $v^{(k)}$, optimization parameter $a^{(k)}$, LBFGS memory length m.
Output: $u^{(k+1)}, v^{(k+1)}, a^{(k+1)}$.
$g^{(k)}, g^{(k-1)} \leftarrow \nabla f(v^{(k)}), \nabla f(v^{(k-1)})$
$s_{k-1} \leftarrow v^{(k)} - v^{(k-1)}$ ▷ solution difference
$y_{k-1} \leftarrow g^{(k)} - g^{(k-1)}$ ▷ gradient difference
$\rho_{k-1} \leftarrow \frac{1}{y_{k-1}^T s_{k-1}}$
store $s_{k-1}, y_{k-1}, \rho_{k-1}$
if $k > m$ **then**
  remove $s_{k-m-1}, y_{k-m-1}, \rho_{k-m-1}$ from memory
end if
$\hat{T}_{k-m} \leftarrow \frac{y_{k-1}^T s_{k-1}}{y_{k-1}^T y_{k-1}} I$ ▷ approximation of $[H^{(k-m)}]^{-1}$
$d^{(k)} \leftarrow \mathrm{LBFGS}(g^{(k)}, \hat{T}_{k-m})$ ▷ descent direction
$\alpha_0 \leftarrow 1$ ▷ initial step size
$\alpha^{(k+1)} \leftarrow \textsc{LineSearch}(v^{(k)}, d^{(k)}$, starts from $\alpha_0)$
$u^{(k+1)} \leftarrow v^{(k)} - \alpha^{(k)} d^{(k)}$
$a^{(k+1)} \leftarrow \left(1 + \sqrt{4a^{(k)} + 1}\right)/2$
$v^{(k+1)} \leftarrow u^{(k+1)} + \frac{a^{(k+1)} - 1}{a^{(k+1)}}(u^{(k+1)} - u^{(k)})$
return $u^{(k+1)}, v^{(k+1)}, a^{(k+1)}$

- Divergence-aware preconditioner to mitigate gradient deviation.
  - $\mathcal{L}$: Augmented Lagrangian function.

$$\min_{\boldsymbol{x},\boldsymbol{y}} \mathcal{L}(\boldsymbol{x},\boldsymbol{y}) = \widetilde{W}(\boldsymbol{x},\boldsymbol{y}) + \sum_{k=0}^{K_1-1} \lambda_k \mathcal{D}_k + \sum_{k=K_1}^{K_1+K_2-1} \eta_k \Gamma_k(\boldsymbol{x}^{(k)}, \boldsymbol{y}^{(k)}),$$

  - $\mathcal{P} \in \mathbb{R}^{2\times|N|}$: Preconditioner for all electrostatics systems.

$$\begin{aligned}
\mathcal{P}_{0,i} &= \max\left\{1, \left[\frac{\partial^2 \mathcal{L}}{\partial x_i^2}\right]^{-1}\right\} \\
&= \max\left\{1, \left[\frac{\partial^2 \widetilde{W}}{\partial x_i^2} + \sum_{k=0}^{K_1-1} \lambda_k \frac{\partial^2 \mathcal{D}_k}{\partial x_i^2} + \sum_{k=K_1}^{K_1+K_2-1} \eta_k \frac{\partial^2 \Gamma_k}{\partial x_i^2}\right]^{-1}\right\} \\
&= \max\left\{1, \left[\#\mathrm{pins}(v_i) + \tau \sum_{k=0}^{K_1-1} \lambda_k \mathbb{I}_k(v_i)\mathrm{area}(v_i) + \sum_{k=K_1}^{K_1+K_2-1} \eta_k \frac{\partial^2 \Gamma_k}{\partial x_i^2}\right]^{-1}\right\}
\end{aligned}$$

Preconditioned gradient

$$\nabla \hat{\mathcal{L}} = \nabla \mathcal{L} \odot \mathcal{P}$$

Experimental Results

Experimental Setup

Machine

- Two Intel Xeon Platinum 8358 CPUs (2.60GHz, 32 cores) with 1024GB RAM
- One NVIDIA A800 GPU
- C++ with `LibTorch` for GPU acceleration

Benchmark Suites

- `ISPD2015-FR`: ISPD 2015 benchmark suite [Ismail+, ISPD'15] with fence regions only.
- `ISPD2015-HR`: Modified ISPD 2015 benchmark suite where some fence regions are modified into default regions and guide regions.

Placers for comparison

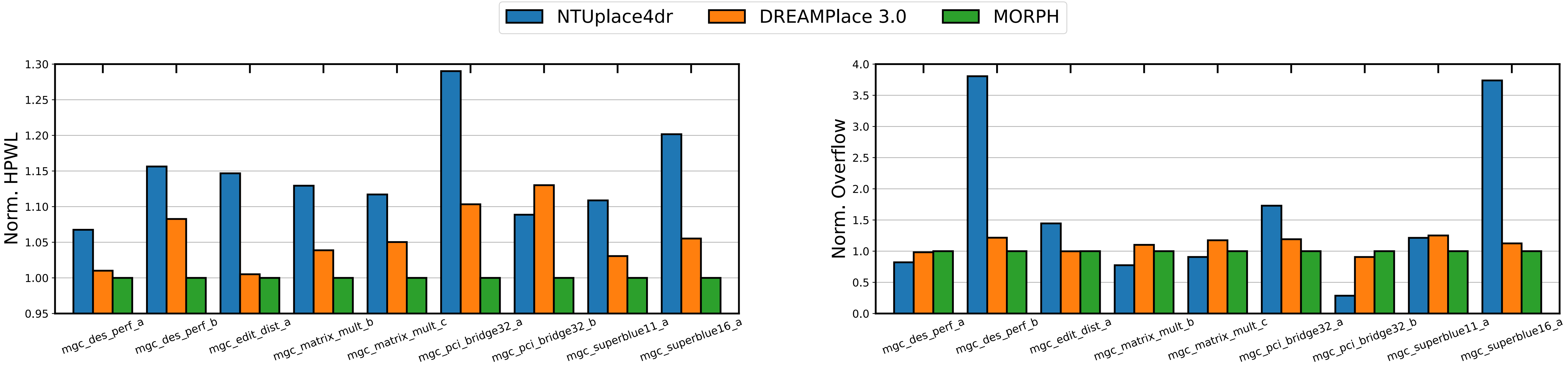
- `NTUplace4dr` [Huang+, TCAD'17]
- `DREAMPlace 3.0` [Gu+, ICCAD'20]

Statistics of `ISPD2015-FR` benchmarks and its variant `ISPD2015-HR`.

Design	#Cells	#Nets	ISPD2015-FR #Fence	ISPD2015-HR #Fence	ISPD2015-HR #Default	ISPD2015-HR #Guide
mgc_des_perf_a	108K	115K	4	1	2	1
mgc_des_perf_b	113K	113K	12	4	4	4
mgc_edit_dist_a	127K	134K	1	0	0	1
mgc_matrix_mult_b	146K	152K	3	1	1	1
mgc_matrix_mult_c	146K	152K	3	1	1	1
mgc_pci_bridge32_a	30K	34K	4	1	2	1
mgc_pci_bridge32_b	29K	33K	3	1	1	1
mgc_superblue11_a	926K	936K	4	1	1	2
mgc_superblue16_a	680K	697K	2	0	2	0

HPWL and Global Routing Overflow Comparison on `ISPD2015-FR`

- **14.3%** better HPWL than `NTUplace4dr`
- **5.6%** better HPWL than `DREAMPlace 3.0`
- **24%** smaller overflow than `NTUplace4dr`
- **10%** smaller overflow than `DREAMPlace 3.0`



Consistently achieve better HPWL results than `NTUplace4dr` and `DREAMPlace 3.0` across all cases!

Ablation Study on `ISPD2015-HR` with Hybrid Regions

Gradient Decent (GD)

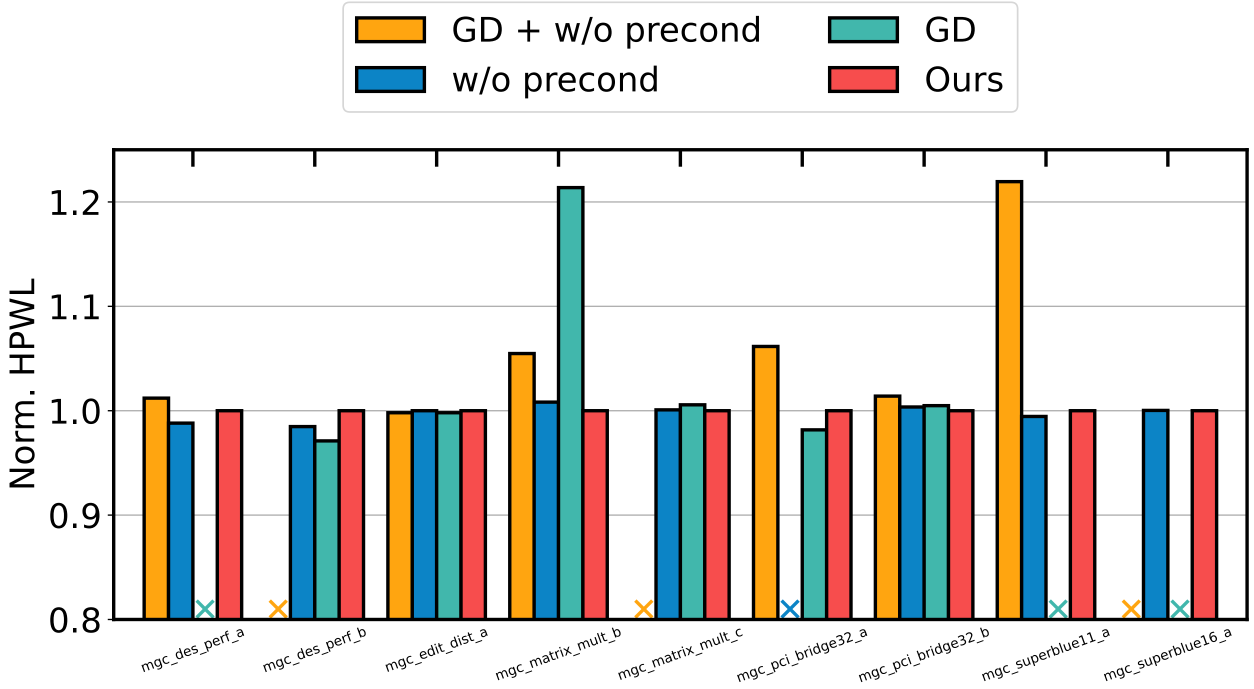
- 3/7 designs diverge
- +2.6% HPWL
- +12% Overflow
- -28% runtime

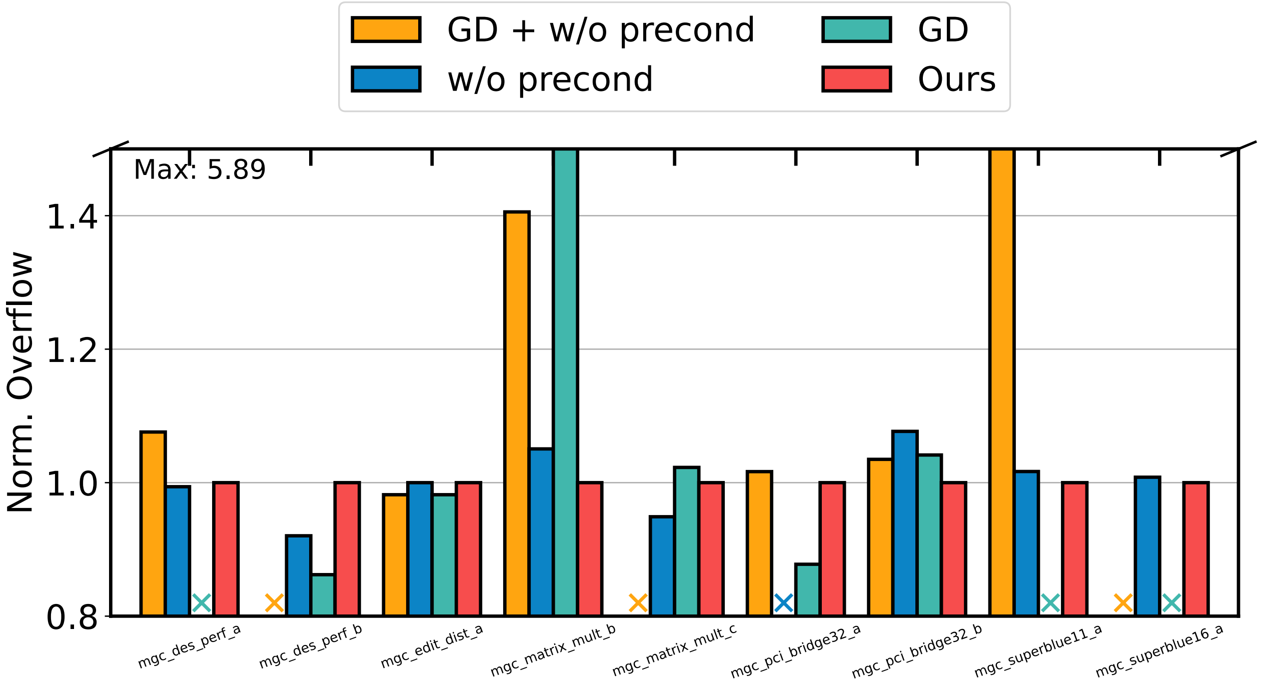
w/o precond

- 1/7 design diverge
- -0.2% HPWL
- Almost the same overflow
- +24% runtime

GD + w/o precond

- 3/7 designs diverge
- +32.7% HPWL
- +45% overflow
- -16% runtime





The Nesterov-accelerated LBFGS algorithm and preconditioner technique can significantly improve quality and robustness with minor runtime overhead.

Conclusion & Future Work

Conclusion & Future Work

- We propose MORPH, an innovative ASIC placer specifically designed to manage hybrid region constraints (i.e., default regions, fence regions, and guide regions).
  - We propose a shared electrostatics model and a binary-lifting-based region pruning algorithm that integrate hybrid region constraints into a unified multi-electrostatic formulation.
  - We propose a wirelength-prioritized penalty method to manage the tradeoff between wirelength and guide constraint penalty.
  - Our proposed Nesterov's accelerated LBFGS algorithm can improve the quality and stability with second-order information.
- Experimental results demonstrate that we achieve a 5.6-14.3% HPWL improvement and a 10-24% overflow reduction compared to previous SOTA region-aware placers.

Future Work

- More efficient hybrid-region-aware legalization.

THANK YOU!

jingmai@pku.edu.cn

Personal Website

magic3007.github.io

PEKING UNIVERSITY 1898

IEEE/ACM 2024 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN ICCAD 43rd Edition