ASP-DAC 2026
31st Asia and South Pacific Design Automation Conference

Partitioning-free 3D-IC Floorplanning

Shuo Ren, Zhen Zhuang, Rongliang Fu, Leilei Jin, Libo Shen,
Bei Yu, Tsung-Yi Ho

Department of Computer Science & Engineering
The Chinese University of Hong Kong

Jan. 22, 2026

Outline

Introduction

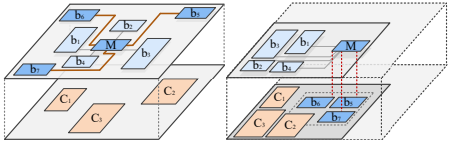
Background: Why 3D IC?

Interconnect Bottleneck in 2D ICs:

- Transistor scaling follows Moore's Law
- Interconnect scaling lags behind[a]
- Routing delay becomes the key bottleneck

3D Integration Benefits:[b]

- Stacks device layers vertically
- Transforms long intra-die wires into short inter-die connections
- Increases integration density



Comparison of 3D floorplans before and after inter-die optimization.

[a] **Kun Cao et al. (2019). "A survey of optimization techniques for thermal-aware 3D processors".** In: *Journal of Systems Architecture*.

[b] **Xueyan Zhao et al. (2025). "Toward Advancing 3D-ICs Physical Design: Challenges and Opportunities".** In: *Proc. ASPDAC*.

Three Categories of Modern 3D Integration:[1]

TSV-based[2]

- High parasitics
- Area overhead
- Sparse connections

Monolithic[3]

- Sequential processing
- High cost
- Reduced yield

F2F Hybrid Bonding[4]

- High-density links
- Low parasitic
- Our focus!

[1] **Eric Beyne (2016). "The 3-D Interconnect Technology Landscape".** In: *IEEE Design & Test*.

[2] **Tiantao Lu et al. (2017). "TSV-based 3-D ICs: Design methods and tools".** In: *IEEE TCAD* 36.10, pp. 1593–1619.

[3] **Sai Surya Kiran Pentapati and Sung Kyu Lim (2021). "Heterogeneous Monolithic 3D ICs: EDA Solutions, and Power, Performance, Cost Tradeoffs".** In: *Proc. DAC*, pp. 925–930. DOI: 10.1109/DAC18074.2021.9586246.

[4] **Kai Ma et al. (2023). "0.5 1/4m Pitch Wafer-to-wafer Hybrid Bonding with SiCN Bonding Interface for Advanced Memory".** In: *Proc. ECTC*, pp. 1110–1114.

3D Floorplanning Problem

Key Differences from 2D Floorplanning:

- Determine both block location (x, y) and die assignment z
- Handle z-axis feasibility and cross-tier density
- Account for non-uniform communication latency

Input:

- Netlist (V_b, V_p, E)
- Block dimensions (w_i, h_i)
- Design constraints (W, H)

Output:

- 3D coordinates (x_i, y_i, z_i)
- Legal block sizes
- Net partitioning

Objective: Minimize wirelength $\mathcal{W}$ and latency $\mathcal{H}$ jointly:

$$\min_{\Theta} \quad \mathcal{F}^o = (1-\mu) \cdot \mathcal{W}(A^o, D) + \mu \cdot \mathcal{H}(\mathbf{S}, L, \mathrm{MP})$$

Existing Approaches: 2D Extension Methods

Category 1: Extending 2D Representations to 3D

- **3D slicing trees**[5]
  - Hierarchical layer partitioning
- **Grouped sequence pairs**[6]
  - Encode cross-layer placement
- **FlexPlanner**[7]
  - Graph-based with deep reinforcement learning

Limitations:

- Exponential solution space growth
- Slow metaheuristics required
- Cannot easily decouple die assignment from placement

[5] Lei Cheng, Liang Deng, and Martin DF Wong (2005). "Floorplanning for 3-D VLSI design". In: *Proc. ASPDAC*, pp. 405–411.

[6] Rajeev K. Nain and Malgorzata Chrzanowska-Jeske (2011). "Fast Placement-Aware 3-D Floorplanning Using Vertical Constraints on Sequence Pairs". In: *IEEE TVLSI*.

[7] Ruizhe Zhong, Xingbo Du, Shixiong Kai, et al. (2024). "FlexPlanner: Flexible 3D Floorplanning via Deep Reinforcement Learning in Hybrid Action Space with Multi-Modality Representation". In: *Proc. NeurIPS* 37, pp. 49252–49278.

Existing Approaches: Partitioning-first Methods

Category 2: Pre-partitioning Paradigm

- **FM partitioning**[8]
  - Fiduccia-Mattheyses algorithm for block-to-die assignment
- **Spectral partitioning**[9]
  - Eigenvalue-based graph partitioning
- **Snap3D, CoPlace**[10]
  - Constrained placement-driven physical design

Limitations:

- Fixed early die assignment constrains optimization
- Cannot jointly optimize placement and assignment

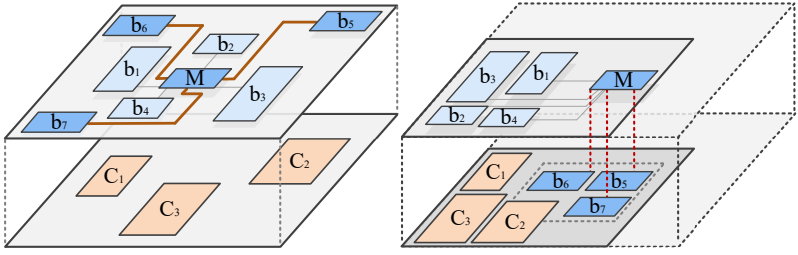
Gap: No method jointly optimizes block placement and die assignment!

[8] **L.-T. Liu et al. (1995). "A gradient method on the initial partition of Fiduccia-Mattheyses algorithm".** In: *Proc. ICCAD*, pp. 229–234.

[9] **Charles J Alpert and So-Zen Yao (1995). "Spectral partitioning: The more eigenvectors, the better".** In: *Proc. DAC*, pp. 195–200.

[10] **Bangqi Fu et al. (2024). "Coplace: Coherent placement engine with layout-aware partitioning for 3D ICs".** In: *Proc. ASPDAC*, pp. 65–70.

Why Partitioning-free Matters:



Key Insight:

- Pre-partitioning constrains highly connected modules to single tier
- Our method remaps nets across tiers for shorter connections

Our Contributions: GREAT3D

We propose GREAT3D, a partitioning-free 3D floorplanning framework:

1. **Native-3D Method**
   - Jointly performs block floorplanning and die assignment
   - Unified optimization framework without partitioning
2. **Multi-objective Formulation**
   - Balances interconnect cost and die utilization
   - Produces compact layouts with reduced wirelength
3. **Latency-aware Modeling**[11]
   - Supports early-stage performance estimation
   - CPI-based system-level latency model

Result: Up to 60% wirelength reduction on large designs!

[11] **Zhen Zhuang et al. (2023). "Multi-Product Optimization for 3D Heterogeneous Integration with D2W Bonding".** In: *Proc. ICCAD.*

Preliminaries

Half-Perimeter Wirelength (HPWL):[12]

For a net e_i connecting block set S_i:

2D HPWL Definition

$$\mathrm{WL}_{2D}(e_i) = \underbrace{\max_{b \in S_i}(x_b) - \min_{b \in S_i}(x_b)}_{\text{X-span}} + \underbrace{\max_{b \in S_i}(y_b) - \min_{b \in S_i}(y_b)}_{\text{Y-span}}$$

Pairwise Formulation: For differentiable optimization[13]

Connection-Distance Inner Product

$$\mathcal{W}(A^o, D) = \langle A^o, D \rangle = \sum_{i,j} A^o_{ij} \cdot D_{ij}$$

- A^o_{ij}: Connection strength between blocks i and j
- D_{ij}: Squared Euclidean distance between blocks

[12] **Wei Li et al. (2023). "Global Floorplanning via Semidefinite Programming".** In: *Proc. DAC*.

[13] **Chaomin Luo, Miguel F Anjos, and Anthony Vannelli (2008). "Large-scale fixed-outline floorplanning design using convex optimization techniques".** In: *Proc. ASPDAC*. IEEE, pp. 198–203.

Latency Model: CPI-based Formulation

For F2F Hybrid Bonding Systems:[14]

CPI-based Latency Objective

$$\mathcal{H}(\mathbf{S}, L, \mathrm{MP}) = \underbrace{\mathrm{CPI}_{\mathrm{sta}}}_{\text{Static CPI}} + \alpha_1 \cdot \underbrace{(\mathrm{lat}_{\mathrm{ppi}} - \mathrm{lat}_{\mathrm{sta}})}_{\text{Dynamic latency}}$$

Component Definitions:

- **Latency:** $\mathrm{lat}_{\mathrm{ppi}} = \langle \mathbf{S}, L \rangle$ (Frobenius inner product)
  - $\mathbf{S} = \sum_p N_p$: Structure matrix from net connectivity
  - $L \in \mathbb{R}^{n \times n}$: Latency matrix
- **Static CPI:** $\mathrm{CPI}_{\mathrm{sta}} = \alpha_2 \cdot \mathrm{MP} + \beta_1$
- **Wire delay:** $\beta_2 \cdot \mathrm{wirelength}_{ppi}$[15]

$\Rightarrow$ Captures inter-die communication cost in 3D ICs

[14] Zhen Zhuang et al. (2023). "Multi-Product Optimization for 3D Heterogeneous Integration with D2W Bonding". In: *Proc. ICCAD*.

[15] Yulei Zhang et al. (2010). "Prediction and comparison of high-performance on-chip global interconnection". In: *IEEE transactions on very large scale integration (VLSI) systems* 19.7, pp. 1154–1166.

Four Key Constraints for 3D Floorplanning:

1. Overlap Constraint:

$$\sum_{i<j} \text{Overlap}(b_i, b_j) \leq \epsilon_{\text{overlap}}$$

Blocks must not physically overlap

2. Aspect Ratio:

$$r_i^{\min} \leq \frac{w_i}{h_i} \leq r_i^{\max}$$

Soft blocks maintain valid shapes

3. Fixed-Outline:

$$0 \leq x_i \pm \frac{w_i}{2} \leq W$$

$$0 \leq y_i \pm \frac{h_i}{2} \leq H$$

All blocks within die boundary

4. Bonding Density:

$$|E_{\text{inter-die}}| \leq \rho_{\max} \cdot A_{\text{die}}$$

Limited vertical connections

Unified Objective Formulation

Challenge: Optimize two heterogeneous terms simultaneously

Original Multi-objective

$$\mathcal{F}^o = (1-\mu) \cdot \underbrace{\mathcal{W}(A^o, D)}_{\text{Wirelength}} + \mu \cdot \underbrace{\mathcal{H}(\mathbf{S}, L, \text{MP})}_{\text{Latency}}$$

Step 1: Expand both terms

$$\mathcal{F}^o = (1-\mu) \cdot \sum_{i,j} A^o_{ij} \cdot d_{ij} + \mu \cdot \sum_{i,j} S_{ij} \cdot L_{ij}$$

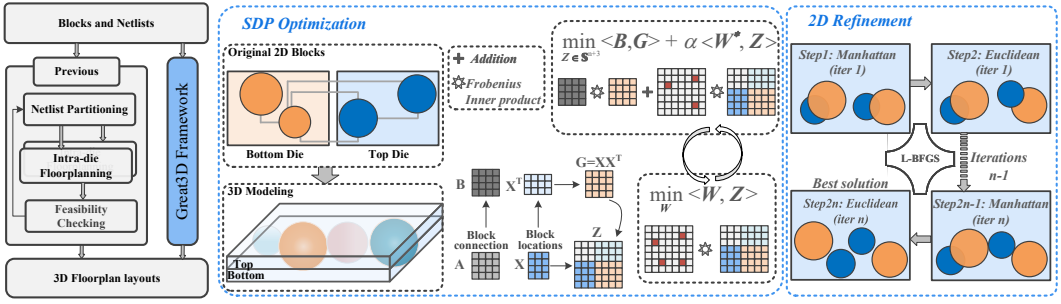
Step 2: Construct weighted connection matrix

$$A_{ij} = (1-\mu) \cdot A^o_{ij} + \mu \cdot S_{ij} \cdot \lambda_{ij}$$

Unified Objective

$$\min_{\Theta} \quad \mathcal{F} = \langle A, D \rangle = \sum_{i,j} A_{ij} \cdot d_{ij}$$

Methodology



Overview of the proposed GREAT3D optimization framework.

Two-Stage Optimization:

1. **3D SDP Optimization**: Solve unified objective in 3D space
2. **2D Refinement**: Improve layout quality per die via L-BFGS-B.

Key Idea: Encode positions via Gram matrix for SDP relaxation[16]

Gram Matrix Definition

$$\mathbf{G} = \mathbf{X}\mathbf{X}^\top \in \mathbb{R}^{n\times n}$$

where $\mathbf{X} \in \mathbb{R}^{n\times 3}$ contains 3D positions of n blocks

Distance from Gram Matrix:

$$D_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2 = G_{ii} + G_{jj} - 2G_{ij}$$

Objective Reformulation:

Inner Product Form

$$\langle \mathbf{A}, \mathbf{D} \rangle = \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{A}_{ij}\left(\mathbf{G}_{ii} + \mathbf{G}_{jj} - 2\mathbf{G}_{ij}\right)$$

$\Rightarrow$ Distance optimization via positive semidefinite matrix $\mathbf{G}$

[16] **Wei Li et al. (2023). "Global Floorplanning via Semidefinite Programming".** In: *Proc. DAC.*

Goal: Simplify the objective computation

Matrix $\mathbf{B}$ Definition

$$\mathbf{B}_{ij} = \begin{cases} \sum_{k=1}^{n} \mathbf{A}_{ik} + \sum_{k=1}^{n} \mathbf{A}_{kj}, & \text{if } i = j \\ -2\,\mathbf{A}_{ij}, & \text{if } i \neq j \end{cases}$$

Interpretation:

- **Diagonal elements** ($i = j$): Sum of all connections to/from block i
- **Off-diagonal elements** ($i \neq j$): Scaled pairwise connection strength

Property:

$$\langle \mathbf{A}, \mathbf{D} \rangle = \langle \mathbf{B}, \mathbf{G} \rangle$$

$\Rightarrow$ Enables efficient computation in SDP solver

Construct augmented matrix for SDP:

Extended Matrix Definition

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{I}_{3\times 3} & \boldsymbol{X}_{3\times n}^{\top} \\ \boldsymbol{X}_{n\times 3} & \boldsymbol{G}_{n\times n} \end{bmatrix} \in \mathbb{S}_{+}^{3+n}$$

Block Structure:

- $\boldsymbol{I}_{3\times 3}$: Identity matrix (preserves coordinate system)
- $\boldsymbol{X}_{n\times 3}$: Block position matrix
- $\boldsymbol{G}_{n\times n}$: Gram matrix (encodes pairwise distances)

Key Properties:

- $\mathbf{Z} \succeq 0$: Positive semidefinite constraint
- Top-3 eigenvectors capture dominant 3D spatial directions
- Based on Courant-Fischer principle

Two Coupled Subproblems:

Subproblem 1: Update $\mathbf{Z}$

$$\mathbf{Z}^{(k+1)} = \mathcal{P}_1(\mathbf{Z}^{(k)}, \mathbf{W}^{(k)})$$

Objective:

$$f_1(\mathbf{Z}) = \langle \mathbf{B}, \mathbf{G} \rangle + \alpha \langle \mathbf{W}^*, \mathbf{Z} \rangle$$

Placement-induced distances + regularization

Subproblem 2: Update $\mathbf{W}$

$$\mathbf{W}^{(k+1)} = \mathcal{P}_2(\mathbf{W}^{(k)}, \mathbf{Z}^{(k+1)})$$

Objective:

$$f_2(\mathbf{W}) = \langle \mathbf{W}, \mathbf{Z}^* \rangle$$

Align with current layout solution

Non-overlap Constraint:

Pairwise Distance Requirement

$$D_{ij} = \mathbf{Z}_{ii} + \mathbf{Z}_{jj} - 2\mathbf{Z}_{ij} \geq (r_i + r_j)^2, \quad \forall i \neq j$$

Ensures blocks are separated by at least sum of their radii

Dual Variable Constraints:

Normalized Semidefinite Cone

$$\mathbf{0} \preceq \mathbf{W} \preceq \mathbf{I}, \quad \text{Trace}(\mathbf{W}) = n$$

Interpretation:

- Eigenvalues of $\mathbf{W}$ lie in $(0, 1)$
- Total spectral weight is normalized to n (number of blocks)
- $\mathbf{Z} \succeq 0$: Positive semidefinite requirement

After SDP Convergence:

1. **Spectral Decomposition:**

$$\mathbf{G} = \mathbf{V}\tilde{\mathbf{V}}^{\top}$$

2. **Extract Top-3 Eigenvectors:**

$$\mathbf{V}_3 \in \mathbb{R}^{n\times 3} \quad \text{(dominant spatial directions)}$$

3. **Recover 3D Coordinates:**

$$\mathbf{X} = \mathbf{V}_3 \cdot {}^{\circ}_{\text{sqrt}}$$

Each row of $\mathbf{X} \rightarrow$ position (x_i, y_i, z_i)

Die Assignment Rule:

$z_i <$ threshold $\Rightarrow$ **Bottom Die**

$z_i \geq$ threshold $\Rightarrow$ **Top Die**

Goal: Improve per-die layout quality after die assignment

Composite Objective Function

$$f(\mathbf{x}) = \underbrace{\sum_{i<j} \mathbf{A}_{ij} \cdot \phi(\mathbf{x}_i, \mathbf{x}_j)}_{\text{Wirelength term}} + \beta \cdot \underbrace{f_{\text{pen}}(\mathbf{x})}_{\text{Spacing penalty}}$$

Mixed L1-L2 Distance Function:[17]

$$\phi(\mathbf{x}_i, \mathbf{x}_j) = \alpha_3 \|\mathbf{x}_i - \mathbf{x}_j\|_1 + (1 - \alpha_3)\|\mathbf{x}_i - \mathbf{x}_j\|_2$$

Soft Spacing Penalty:

$$f_{\text{pen}}(\mathbf{x}) = \sum_{i<j} \left[\max\left(0, d_{ij}^{\min} - \|\mathbf{x}_i - \mathbf{x}_j\|_2\right)\right]^2$$

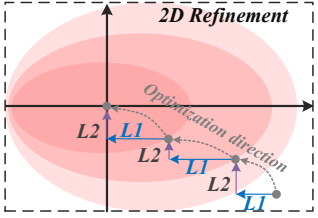
$d_{ij}^{\min}$: Minimum required distance between blocks i and j

[17] **Wei Li et al. (2023). "Global Floorplanning via Semidefinite Programming".** In: *Proc. DAC.*

Why Mix L1 and L2 Norms?

Gradient of Mixed Norm:

$$\nabla_i \phi = \alpha_3 \cdot \text{sign}(\mathbf{x}_i - \mathbf{x}_j) + (1 - \alpha_3) \cdot \frac{\mathbf{x}_i - \mathbf{x}_j}{\| \cdot \|_2}$$



Benefits:

- L1: Constant gradient magnitude
- L2: Smooth, direction-aware
- Alternating avoids local minima

Full Gradient:

$$\frac{\partial f}{\partial \mathbf{x}_i} = \sum_{j \neq i} \mathbf{A}_{ij} \cdot \nabla_i \phi - 2\beta \sum_{j \neq i} \Delta_{ij} \cdot \frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|_2}$$

where $\Delta_{ij} = \max(0, d_{ij}^{\min} - \|\mathbf{x}_i - \mathbf{x}_j\|_2)$

Limited-memory BFGS with Box Constraints

Memory-efficient quasi-Newton method for large-scale optimization[18]

Direction Update:

$$\mathbf{p} = -H_k \nabla f(\mathbf{X}_k)$$

- H_k: Inverse Hessian approximation
- Implicitly stored via history vectors

History Update:

$$s_k = \mathbf{X}_k - \mathbf{X}_{k-1}$$
$$y_k = \nabla f(\mathbf{X}_k) - \nabla f(\mathbf{X}_{k-1})$$

- Store last m pairs (s_k, y_k)
- Typically $m = 5 \sim 20$

[18] **Richard H Byrd et al. (1995). "A limited memory algorithm for bound constrained optimization".** In: *SIAM Journal on scientific computing* 16.5, pp. 1190–1208.

Armijo Line Search:[a]

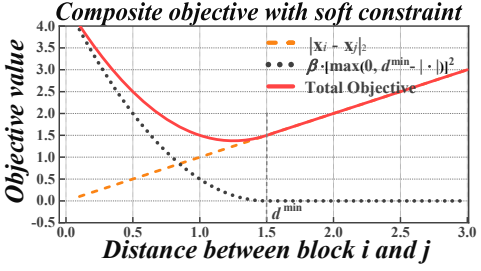
Find step size η satisfying:

$$f(\mathbf{X} + \eta\mathbf{p}) \leq f(\mathbf{X}) + c \cdot \eta \cdot \nabla f^{\top}\mathbf{p}$$

- Sufficient decrease condition
- $c \approx 10^{-4}$ (Armijo constant)
- Ensures monotonic decrease of f

[a] **Jorge Nocedal and Stephen J Wright (2006). "Nonlinear equations".** In: *Numerical Optimization*, pp. 270–302.

Composite Objective Function



Soft constraint with distance penalty

Box Constraints

Bounds enforced via gradient projection: $\mathbf{X} \in [\mathbf{l}, \mathbf{u}]$

Experiments

Experimental Setup

Benchmarks:[a]

- GSRC benchmark suite
- 5 testcases: n10, n30, n50, n100, n200
- Varying design complexity

Implementation:

- Python 3.11
- Ubuntu 22.04 LTS
- Dual Intel Xeon Gold 6426Y
- 256 GB RAM

[a] GSRC Benchmark Suite n.d.

Baseline Methods:

1. FM[a] + AR[b]
2. FM + Li[c]
3. Spectral[d] + AR
4. Spectral + Li
5. GSP[e]
6. TA3D[f]

[a] Liu et al. 1995.
[b] Luo, Anjos, and Vannelli 2008.
[c] Li et al. 2023.
[d] Alpert and Yao 1995.
[e] Nain and Chrzanowska-Jeske 2011.
[f] Lin et al. 2021.

Metrics: Wirelength (HPWL), Runtime (s), CPI (latency)

Model Parameters from Literature:

Param	Value	Ref.	Param	Value	Ref.
α_1	5×10^{-4}	Clapp et al. 2015	α_2	1.8×10^{-1}	Clapp et al. 2015
β_1	1.1×10^{-1}	Clapp et al. 2015	MP	2.4×10^{-1}	Clapp et al. 2015
CPI_{sta}	1.532×10^{-3}	Zhuang et al. 2023	β_2	9.8×10^{1}	Zhang et al. 2010

CPI Formula Recall:

$$\mathcal{H} = \text{CPI}_{\text{sta}} + \alpha_1 \cdot (\text{lat}_{\text{ppi}} - \text{lat}_{\text{sta}})$$

- Parameters calibrated for F2F hybrid bonding systems
- Based on published experimental data

Wirelength Comparison

Test	FM+AR	FM+Li	Spec+AR	Spec+Li	GSP	TA3D	Great3D
n10	43,223	48,881	24,571	28,020	36,444	41,166	**23,415**
n30	84,456	88,703	67,193	72,048	100,818	79,429	**56,542**
n50	217,731	219,508	144,133	147,835	174,368	203,847	**119,495**
n100	402,324	388,336	268,341	259,307	385,398	373,565	**166,792**
n200	880,486	852,036	446,443	408,192	397,594	832,867	**345,099**
Ratio	2.03×	2.06×	1.27×	1.29×	1.65×	1.90×	**1.00**×

Key Observations:

- GREAT3D achieves **best wirelength** across all benchmarks
- Up to 60% reduction on large-scale designs (n200)
- Average improvement: **1.45**× better than best baseline

Runtime Analysis

Runtime Comparison:

Method	Avg. Ratio	Note
FM+AR	0.49×	Fast but poor WL
FM+Li	0.52×	Fast but poor WL
Spectral+AR	0.54×	Fast but poor WL
Spectral+Li	0.58×	Fast but poor WL
GSP	3.00×	Slow, poor quality
TA3D	2.75×	Slow, poor quality
Great3D	**1.00**×	Best balance

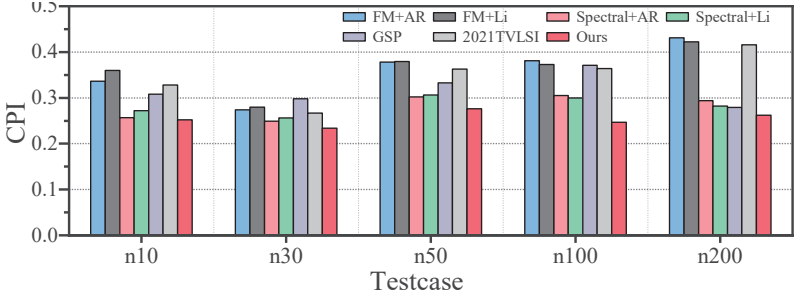
Trade-off Analysis:

- Partitioning methods run faster
- But yield significantly worse wirelength
- 3D extension methods are slower with poor scalability

Analysis:

- Partitioning methods: Fast but constrained optimization
- 3D-extension methods: Slow due to exponential search space
- GREAT3D: Competitive runtime with best quality

CPI-based Latency Model Results:[19]



Results:

- GREAT3D reduces CPI by **8.4% – 15.7%** across all benchmarks
- Average **11.2%** improvement over closest competitor (Spectral+Li)
- Co-optimizing die assignment and placement shortens critical cross-die paths

[19] **Zhen Zhuang et al. (2023). "Multi-Product Optimization for 3D Heterogeneous Integration with D2W Bonding".** In: *Proc. ICCAD.*

Why Does Great3D Outperform?

Root Cause Analysis:

Baseline 1-4 (Partitioning-first):

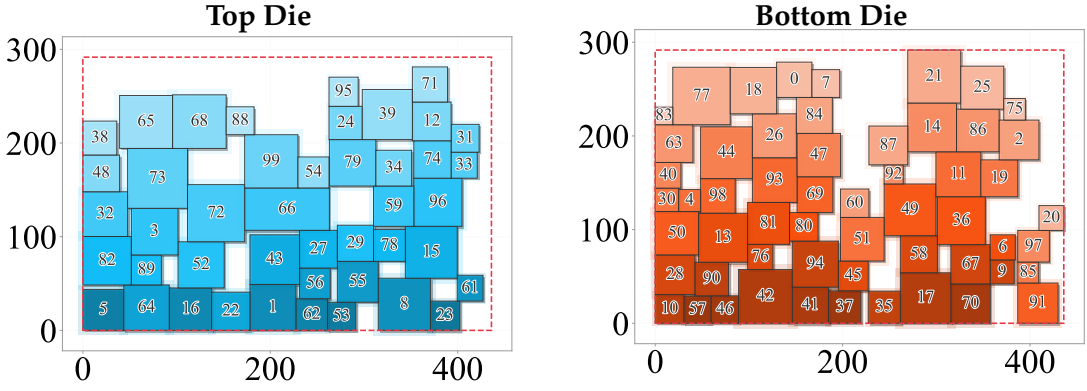
- Early die assignment is fixed
- Cannot adjust during optimization
- Suboptimal global solution

Baseline 5-6 (3D Extension):

- Exponential search space
- Slow metaheuristics
- Poor scalability $O(n! \cdot 2^n)$

Great3D Advantage:
Joint optimization in polynomial time via SDP relaxation
+ Native 3D formulation without pre-partitioning

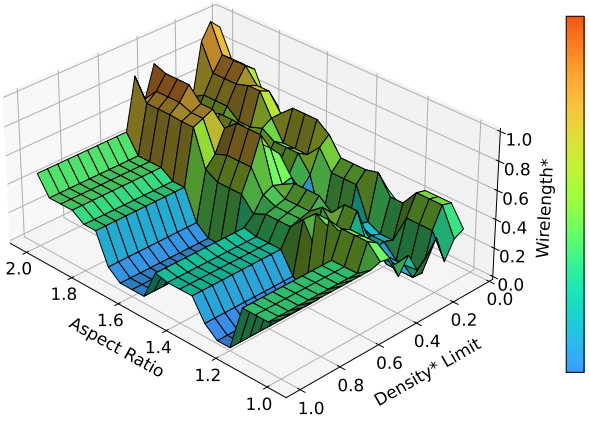
Floorplan Visualization: n100 Benchmark



Observations:

- Red dashed box: Fixed die boundary constraint (aspect ratio 4:3)
- Complementary placement across dies balances area usage
- Die assignment naturally emerges from 3D SDP optimization

Impact of Design Parameters:



Normalized wirelength surface

Findings:

- Elongated outline $\Rightarrow$ higher wirelength
- Reduced γ (inter-die limit) $\Rightarrow$ suboptimal WL
- Non-monotonic trends due to discrete blocks

Aspect Ratio:

$$\alpha = W/H \in [1.0, 2.0]$$

Key Insight: Tighter inter-die constraints degrade layout quality. Aspect ratio affects wirelength in a **layout-sensitive** manner.

Conclusion

Summary

We presented GREAT3D:

- A **partitioning-free** 3D floorplanning framework
- **Unified optimization** of block placement and die assignment
- **Two-stage approach:** 3D SDP + 2D L-BFGS refinement

Key Results on GSRC Benchmarks:

Wirelength	**CPI/Latency**	**Runtime**
Up to 60%↓	11.2% avg.↓	Competitive

Great3D achieves best Pareto efficiency among all evaluated methods!

Technical Contributions

1. **Native-3D Floorplanning**
   - No pre-partitioning required
   - Joint optimization in 3D solution space
   - Better exploration of design space

2. **Unified Objective Formulation**
   - Combined wirelength and latency optimization
   - Weighted connection-distance matching: $\mathcal{F} = \langle A, D \rangle$
   - Handles 3D-specific constraints elegantly

3. **Two-Stage Optimization**
   - SDP for global 3D placement + die assignment
   - L-BFGS-B for per-die refinement
   - Polynomial-time complexity

4. **Latency-aware Modeling**
   - CPI-based system-level model
   - Supports F2F hybrid bonding designs

Comparison with Prior Work

Feature	2D-Ext.	Part.-first	TA3D	Great3D
Joint optimization	✘	✘	✘	✓
Native 3D space	✓	✘	✓	✓
Polynomial time	✘	✓	✘	✓
Latency-aware	✘	✘	✘	✓
Scalable	✘	✓	✘	✓
Best WL	✘	✘	✘	✓

Great3D is the first to achieve:

- Joint block placement and die assignment optimization
- Native 3D formulation with polynomial complexity
- Latency-aware modeling for F2F hybrid bonding

Potential Extensions:

Thermal Management

- Heat dissipation modeling
- Temperature-aware placement
- Hotspot mitigation

Extended 3D Support

- More than 2 dies
- Heterogeneous integration
- Chiplet-based designs

Goal: Enable better 3D IC designs for future applications

Take-away Message

GREAT3D

A Partitioning-free 3D IC Floorplanning Framework

60%	11.2%	Polynomial
WL Reduction	CPI Improvement	Time Complexity

Joint optimization of placement & die assignment in native 3D space

Key Innovation:

Unified analytical framework eliminates separate partitioning stages, enabling globally optimal 3D floorplanning solutions.

THANK YOU!