IEEE/ACM 2023 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN ICCAD 42nd Edition

中国科学院计算技术研究所 INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

iPL-3D: A Novel Bilevel Programming Model for Die-to-Die Placement

Xueyan Zhao[1], Shijian Chen[2], Yihang Qiu[3], Jiangkao Li[4], Zhipeng Huang[2], Biwei Xie[1], Xingquan Li[4], and Yungang Bao[1]

[1]Institute of Computing Technology, CAS,

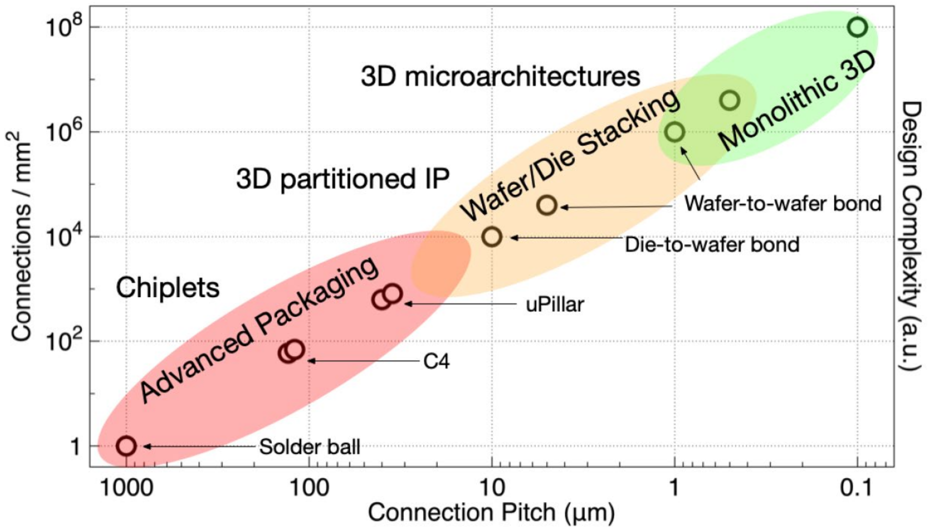
[2]Peng Cheng Laboratory,

[3]University of Chinese Academy of Sciences,

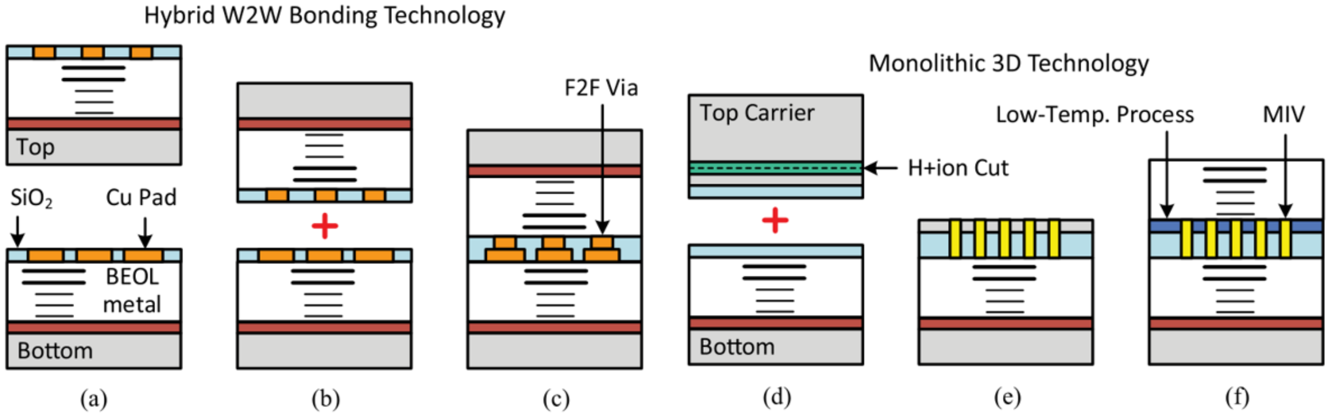
[4]Minnan Normal University,

High Interconnection Capacity Technologies

- **Primary Technologies**: W2W Hybrid Bonding or Monolithic 3-D
- **Technical features**:
  - **Heterogeneous processes** brings **cost** advantages
  - **Higher Interconnection Capacity** brings **performance** advantages



	monolithic	hybrid bonding	micro-bumping
Via/bump size	$0.3\mu m \times 0.3\mu m$	$0.5\mu m \times 0.5\mu m$	$25.0\mu m$
Via/bump pitch	$0.6\mu m$	$1.0\mu m$	$50.0\mu m$
Via/bump height	$0.1\mu m$	$0.17\mu m$	$25.0\mu m$



Advancing Chip Performance through 3D IC

- **[Kim+, DAC' 21]**: **WNS decreased 74%** with M3D compared to 2D-IC.
- **[Zhu+, TVLSI' 21]**: Cortex-A53's **frequency** is **increased by 20%** with M3D.

Table 1: Analysis of 2D and 3D designs. The **Green** means M3D **wins** and the **Red** M3D **loses**.

Cortex-A7							
flow	2D	M3D	Δ	flow	2D	M3D	Δ
clk. freq.	1.00	1.20	20.07%	tot. power	1.00	1.17	17.39%
footprint	1.00	0.50	-50.00%	sw. power	0.28	0.34	20.12%
wirelength	1.00	1.00	-0.49%	int. power	0.55	0.66	21.30%
MIV count	0	349,978	-	leak. power	0.17	0.17	0.22%
density (%)	79.40	79.26	-0.18%	logic power	0.27	0.28	4.73%
worst slack (%)	0.00	0.11	-	seq. power	0.42	0.51	19.24%
total cap	1.00	1.00	0.01%	clk. power	0.21	0.27	27.07%
pin cap	0.43	0.42	-2.07%	macro power	0.10	0.12	23.45%
wire cap	0.57	0.58	1.55%	energy per cycle	1.00	0.98	-2.23%
volt. drop (%)	6.56	8.59	30.91%	temperature (°C)	59.28	69.99	18.07%
std. cell area	1.00	1.02	2.33%				

Cortex-A53							
flow	2D	M3D	Δ	flow	2D	M3D	Δ
clk. freq.	1.00	1.21	21.02%	tot. power	1.00	1.18	18.26%
footprint	1.00	0.50	-50.00%	sw. power	0.14	0.17	17.54%
wirelength	1.00	0.97	-3.43%	int. power	0.77	0.93	20.78%
MIV count	0	588,161	-	leak. power	0.09	0.09	-2.66%
density (%)	72.54	69.92	-3.61%	logic power	0.07	0.07	-3.10%
worst slack (%)	0.00	0.00	-	seq. power	0.30	0.36	20.28%
total cap	1.00	1.00	-1.49%	clk. power	0.17	0.20	18.13%
pin cap	0.43	0.42	-3.57%	macro power	0.46	0.55	20.31%
wire cap	0.57	0.58	-0.28%	energy per cycle	1.00	0.98	-2.28%
volt. drop (%)	7.29	7.71	5.83%	temperature (°C)	54.58	67.98	24.55%
std. cell area	1.00	1.02	1.71%				

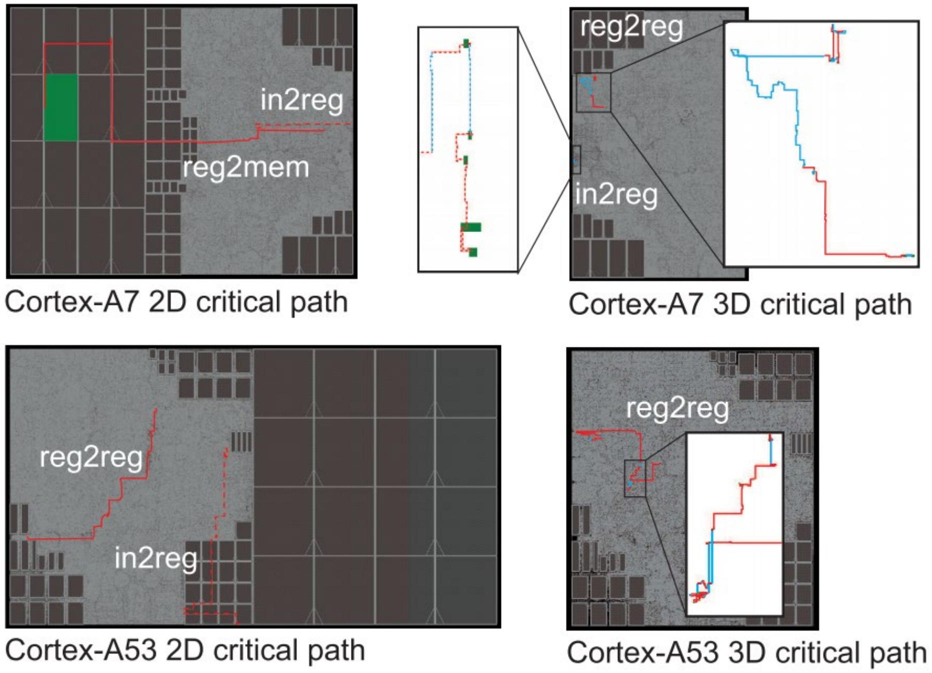


Fig. 1: Timing critical path comparisons.

Placement is Critical in 3D-IC Flow

- **The Main Decider for Variables:** Directly determine the **x and y coordinates** of the cell, while also determining **its corresponding Tier**.
- **The Main Contributor to Wirelength Reduction:** The benefits of 3D-IC mainly come from the possibility of vertical connections **reducing Critical Path Latency**.

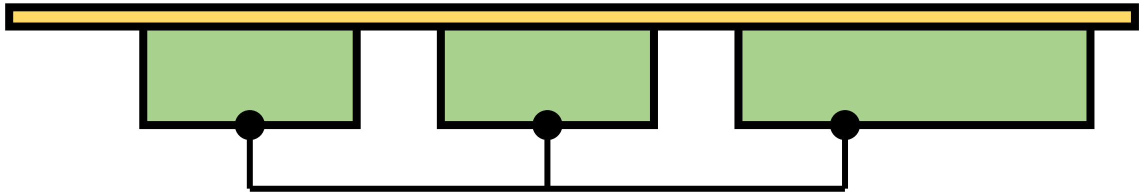


Fig. 2: 2D-IC

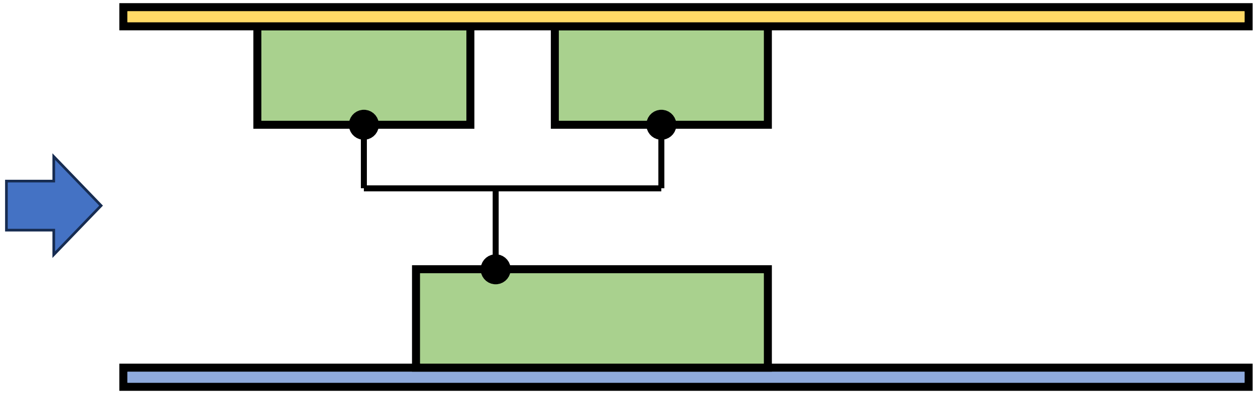
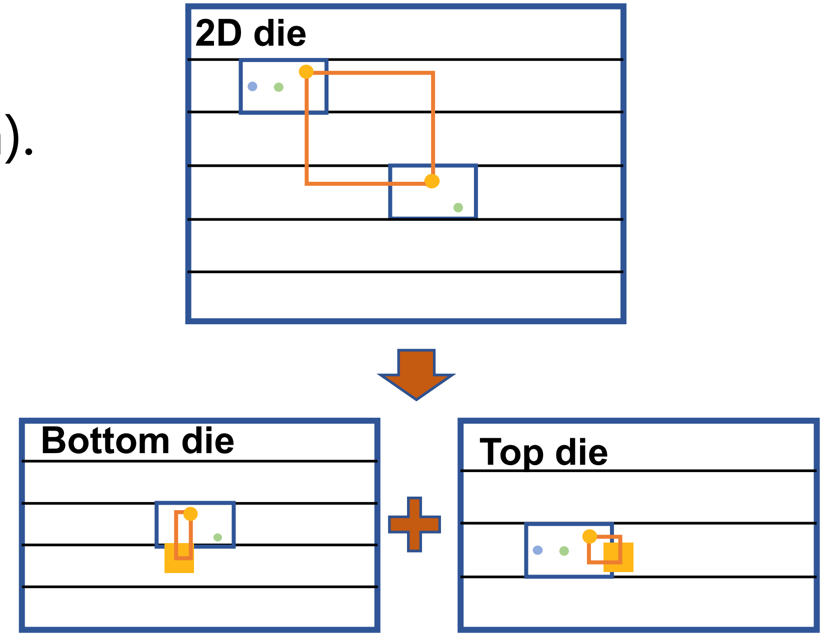


Fig. 3: 3D-IC

Problem Formulation

■ D2D Placement Problem:

- ■ Objective: Minimize 3D HPWL (Half Perimeter Wirelength).
- ■ Constraints:
  - **Heterogeneous Process** Constraint
  - **Maximum Utilization** Constraint
  - **Terminal Spacing** Constraint
  - **Cell Legality** Constraint



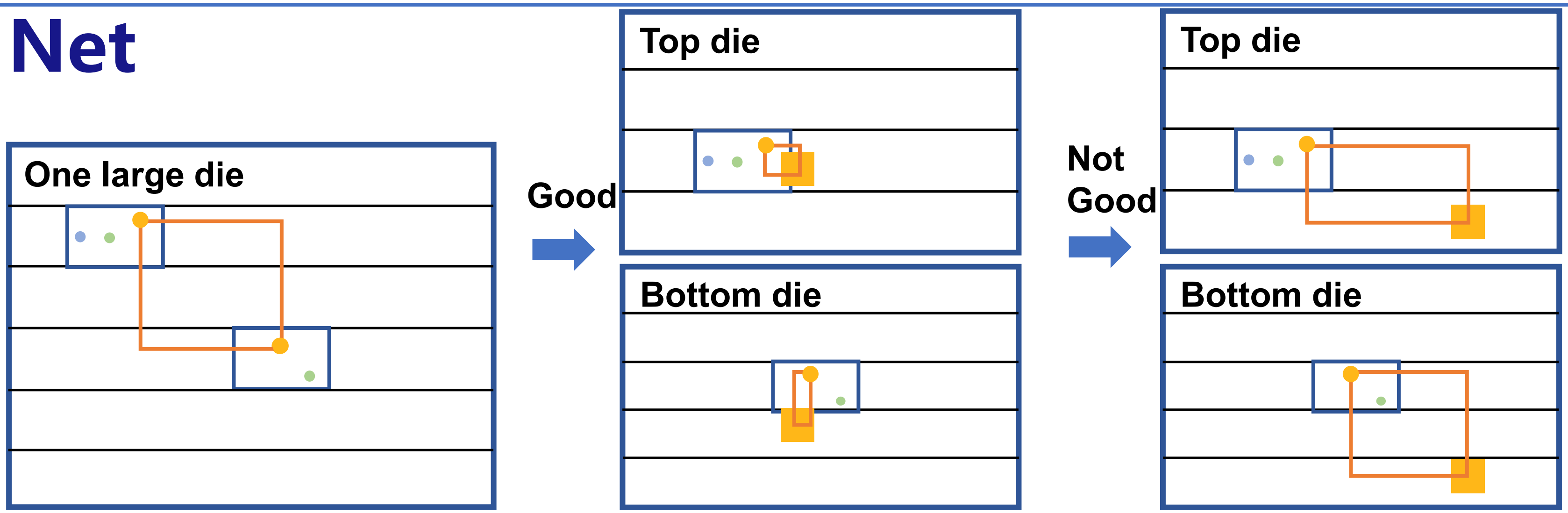
3D HPWL

■ Challenge:

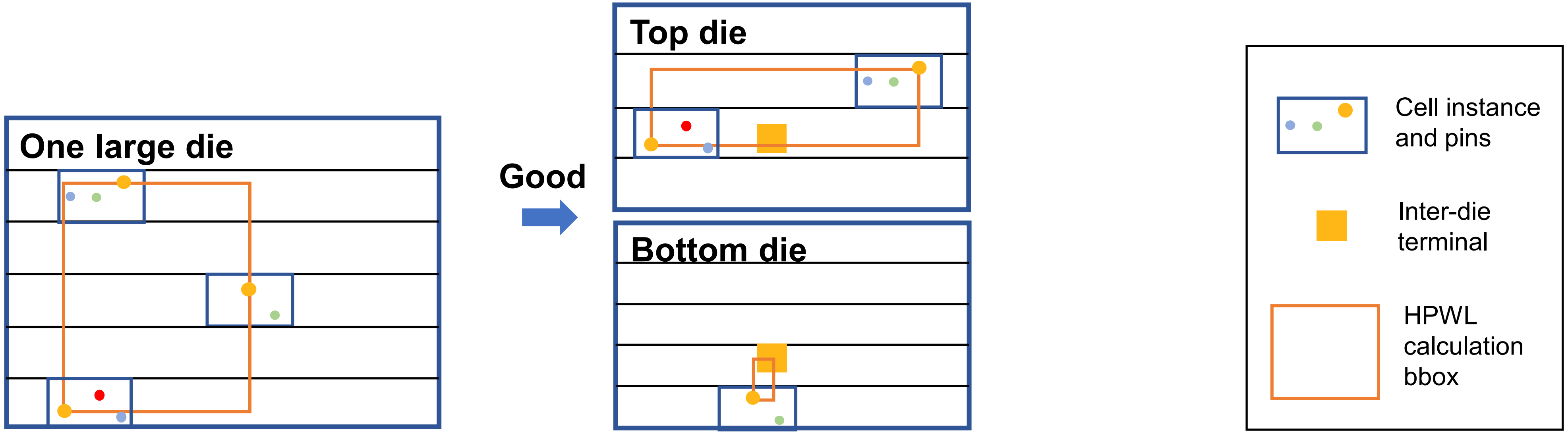
1. **New Decision Variables**.
2. **New Heterogeneous process Constraint**: Introduces significant variations for analytical calculations.
3. **New Objective Function**: Introduces the objective function for the 3D case.

Examples for 3D-HPWL

■ 2-pin Net



■ 3-pin Net



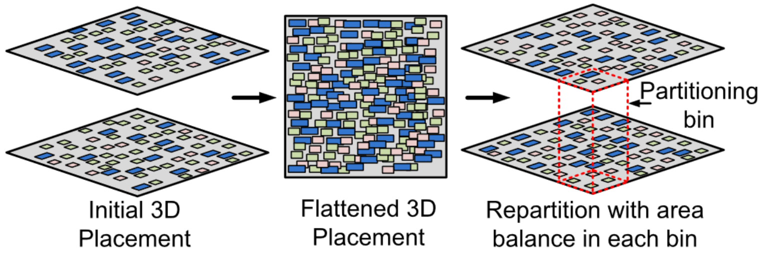
Related Works

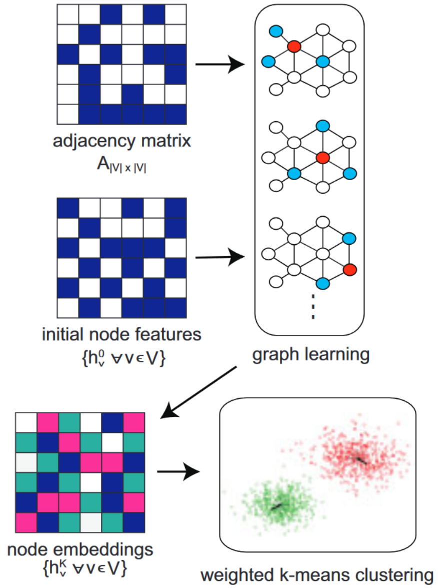
- **Bin-based Min-cut Partitioning**

  [Panth +, TCAD' 17][Panth +, ISPD' 14]:

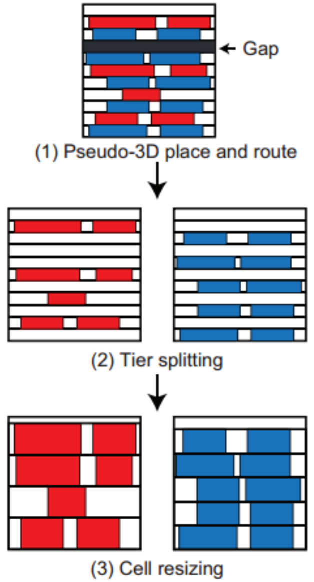
  - Method: Perform planar placement first, followed by balanced binary partitioning in each bin.
- **TP-GNN**[Lu +, DAC' 20]
  - Method: Use unsupervised learning for partitioning, aiming to consider multiple objectives.
- **Snap-3D**[Vanna-Iampikul +, TCAD' 22]
  - Method: Perform odd-even layering on legal results.
- **Existing methods have some limitations:**
  - Do not consider Partition and Placement as a **whole**.
  - Cannot handle **heterogeneous processes**.
  - Cannot consider **MIV Density**.

Bin-based min-cut





TP-GNN

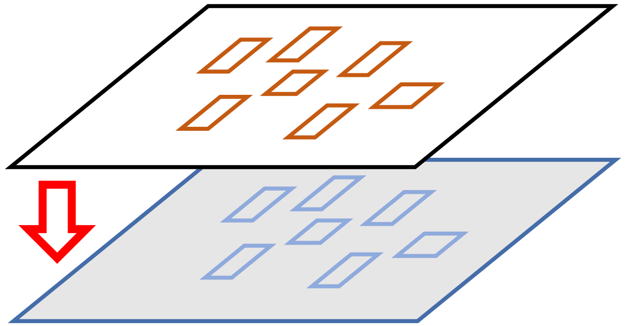


Snap-3D

Intuition of Our Works

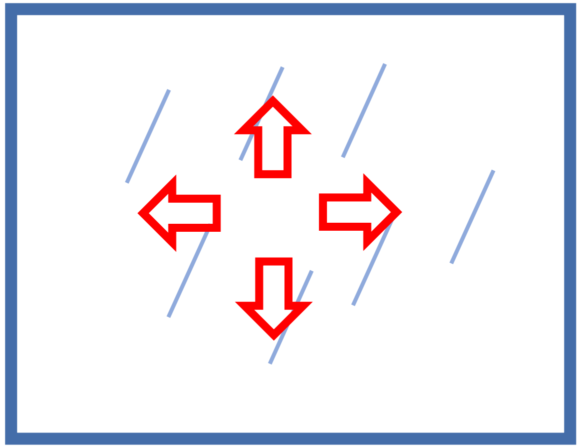
■ Requirement:

1. Consider **comprehensive objectives**, including wirelength, MIV density, etc.
2. The model can be **solved efficiently**.
3. Have a **global view of the solution space** for the overall problem.



■ Methods:

- Leverage **the natural dominance relationship** among decision variables to model the problem as a whole, **efficiently** solving the model with **comprehensive objectives**.
- Obtain the **global view** by exchanging information between two phases.



Bilevel Programming

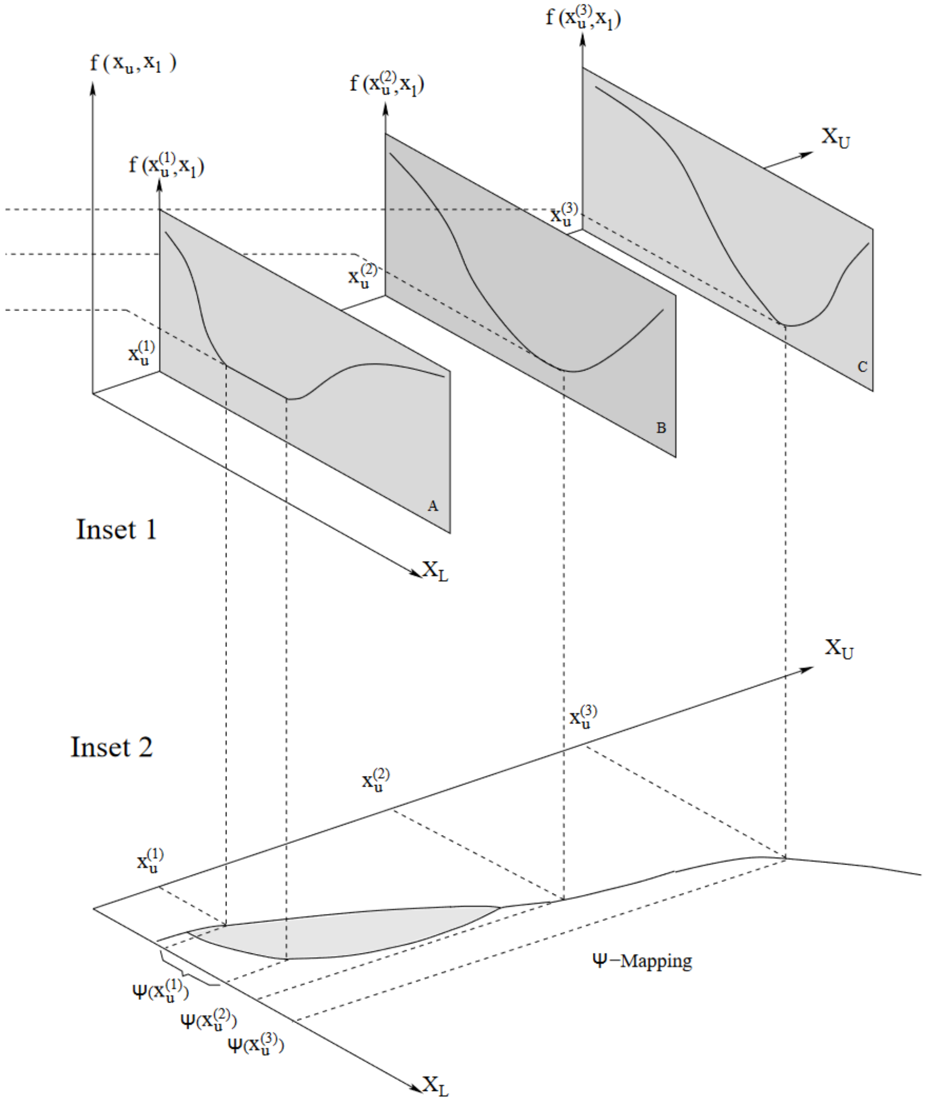
■ Definition of Bilevel Programming:

Definition 1 (Bilevel Programming). *For the upper-level objective function $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ and lower-level objective function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, the bilevel programming problem is given by*

$$\begin{aligned} \min_{x_u \in X_U, x_l \in X_L} \quad & F(x_u, x_l) \\ \text{s.t.} \quad & x_l \in \underset{x_l \in X_L}{\arg\min}\{ \; f(x_u, x_l) | \\ & \qquad g_j(x_u, x_l) \leq 0, j = 1, ..., J\} \\ & G_k(x_u, x_l) \leq 0, k = 1, ..., K, \end{aligned}$$

where $G_k : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}, k = 1, ..., K$ denote the upper-level constraints, and $g_j : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ represent the lower-level

- The **optimal solution** of Lower-level problem is the **constraint** of the upper level problem.



Original Model for D2D Placement

■ Original Model for D2D Placement:

$$
(P_0) \quad \begin{aligned}
\min_{\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{x}_t,\mathbf{y}_t} \quad & \sum_{e_j \in E} \mathrm{WL}_t(e_j; \mathbf{x}, \mathbf{y}, \mathbf{z}, x_{t_j}, y_{t_j}) + \rho\varepsilon(e;\mathbf{z}), \\
s.t. \quad & D_b(\mathbf{x}, \mathbf{y}, \mathbf{x}_t, \mathbf{y}_t, \mathbf{z}) \le M_b, \ \forall b \in S_b, \\
& \textstyle\sum_{i=1}^{n} A_1(c_i)\mathbb{I}(z_i) \le u_t \mathrm{A}, \\
& \textstyle\sum_{i=1}^{n} A_0(c_i)\mathbb{I}(1 - z_i) \le u_b \mathrm{A}, \\
& \textstyle\sum_{e_j \in E} \varepsilon(e_j; \mathbf{z}) \le N_t.
\end{aligned}
$$

$$
\varepsilon(e;\mathbf{z}) = \mathbb{I}\Big(1 - \prod_{c_i \in e}(1 - z_i) - \prod_{c_i \in e} z_i\Big)
$$

■ Important Observation:

- ■ There is a **natural dominance relationship** among decision variables.
- ■ Once z is determined, the **remaining part** is similar to the traditional **2D Placement problem**.
- ■ Traditional min-cut based methods **struggle to obtain a global view**.

✓ **Observations provided conditions for building a bilevel programming model**

Bilevel Programming Reformulation

■ Modeling:

- The **upper level variable** corresponds to z.
- The **lower level variable** corresponds to $\mathbf{x_l} = (\mathbf{x}, \mathbf{y}, \mathbf{x_t}, \mathbf{y_t})$.
- The **objective function** can be rewrite as.

$$F(\boldsymbol{z}, \boldsymbol{x}_l) = WL(\cdot) + \rho\varepsilon(\cdot)$$

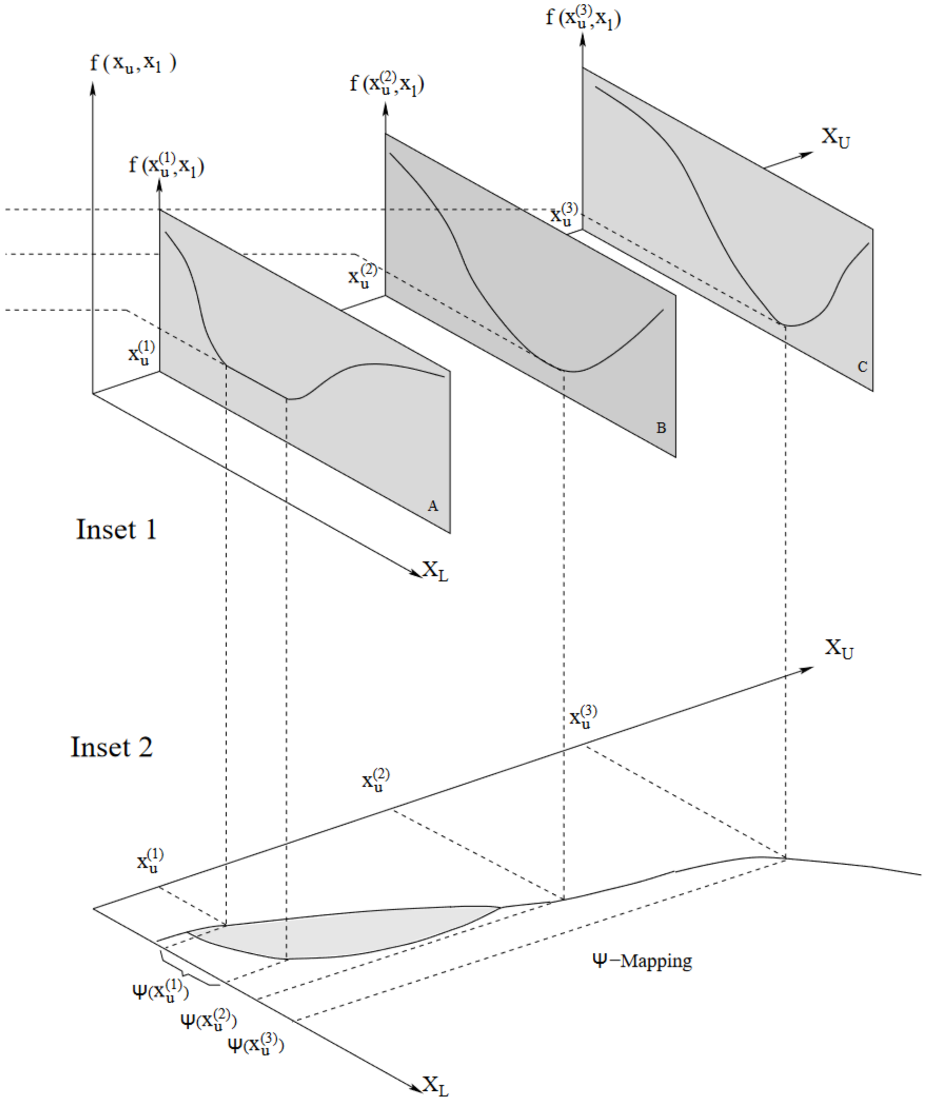
- The **lower level problem** can be defined as:

$$g(\boldsymbol{z}) = \min_{\mathbf{x_l}}\{F(\boldsymbol{z}, \boldsymbol{x}_l) | D_b(\boldsymbol{z}, \boldsymbol{x}_l) \le M_b, \forall b \in S_b\}$$

Definition 1 (Bilevel Programming)**.** *For the upper-level objective function $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ and lower-level objective function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, the bilevel programming problem is given by*

$$\begin{aligned} \min_{x_u \in X_U, x_l \in X_L} \quad & F(x_u, x_l) \\ s.t. \quad & x_l \in \arg\min_{x_l \in X_L}\{ \; f(x_u, x_l) | \\ & \qquad g_j(x_u, x_l) \le 0, j = 1, \dots, J\} \\ & G_k(x_u, x_l) \le 0, k = 1, \dots, K, \end{aligned}$$

where $G_k : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}, k = 1, \dots, K$ denote the upper-level constraints, and $g_j : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ represent the lower-level



Bilevel Programming Reformulation

Modeling:

- Use $g(z)$ instead of the original objective function:

$$\psi(\mathbf{z}) = \mathrm{argmin}_{\mathbf{x}_l}\{F(\mathbf{z}, \boldsymbol{x}_l) | D_b(\mathbf{z}, \boldsymbol{x}_l) \leq M_b, \forall b \in S_b\}$$

$$\forall\, \boldsymbol{x}_l^* \in \psi(\mathbf{z}) \Rightarrow F(\mathbf{z}, \boldsymbol{x}_l^*) = g(\mathbf{z})$$

(P1)

$$\begin{aligned} \min_{\mathbf{z},\mathbf{x}_l} \quad & F(\mathbf{z}, \boldsymbol{x}_l^*) = g(\mathbf{z}) \\ s.t. \quad & \mathbf{x}_l \in \Psi(\mathbf{z}) \\ & \textstyle\sum_{i=1}^n A_1(c_i)\mathbb{I}(z_i) \leq u_t A \\ & \textstyle\sum_{i=1}^n A_0(c_i)\mathbb{I}(1 - z_i) \leq u_b A \\ & \textstyle\sum_{e_j \in E} \varepsilon(e_j; \mathbf{z}) \leq N_t \end{aligned}$$

- The variable x_l **does not appear** in other constraints and objective. **To solve efficiently**, we split the original problem and introduce a surrogate function.

(P2)

Subproblem 1.

(SP1)

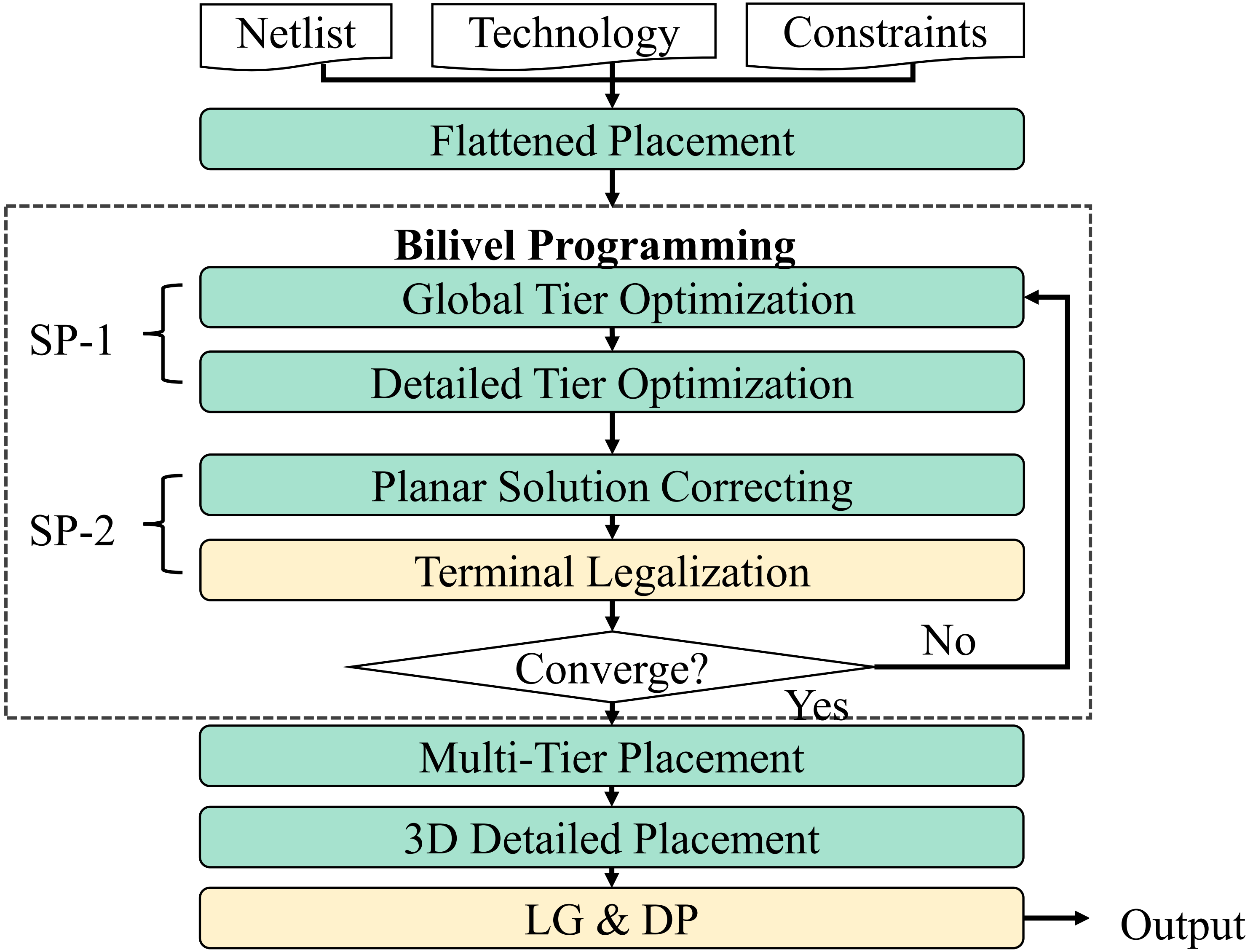
$$\begin{aligned} \min_{\mathbf{z}} \quad & \hat{g}(\mathbf{x}_l^k, \mathbf{z}) \\ s.t. \quad & \textstyle\sum_{i=1}^n A_1(c_i)\mathbb{I}(z_i) \leq u_t A \\ & \textstyle\sum_{i=1}^n A_0(c_i)\mathbb{I}(1 - z_i) \leq u_b A \\ & \textstyle\sum_{e_j \in E} \varepsilon(e_j; \mathbf{z}) \leq N_t \end{aligned}$$

Subproblem 2.

(SP2)

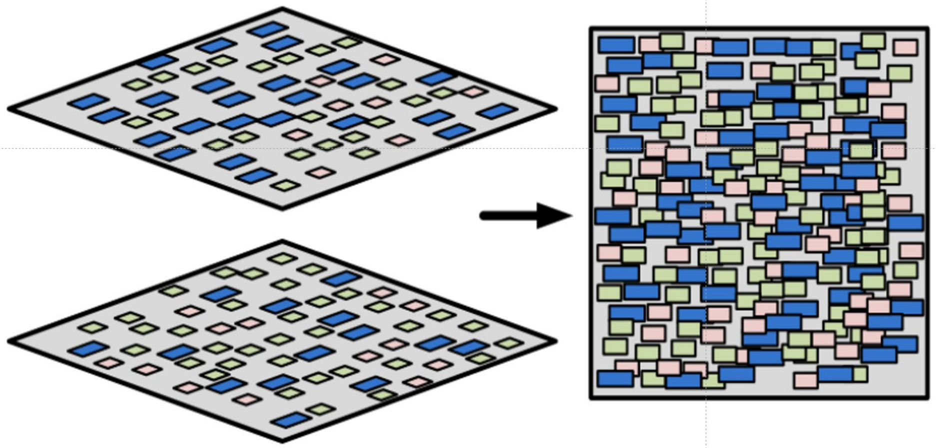
$$\mathbf{x}_l^{k+1} = \underset{\Psi(\mathbf{z}^{k+1})}{Proj}(\mathbf{x}_l^k)$$

Alternate Optimization Framework



Flattened Placement

- **Goal:**
  - Obtaining a **high-quality initial planar solution** is crucial at the beginning of iterative solving.
  - The planar solution can also **provide sufficient information for the surrogate function** $\hat{g}(\mathrm{x}_l, \mathrm{z})$.



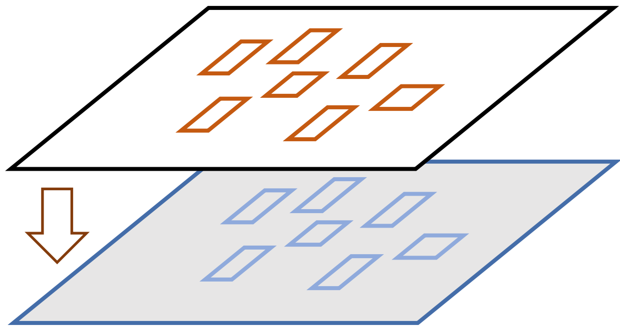
- **Method:**
  - Place all standard cells in one layer and **double the capacity of the bin**. Then solve the global placement problem to obtain $\boldsymbol{x}_{2D}$.
- **Upper bound:**
  - The quality of the optimal planar solution obtained from Flattened Placement is the upper bound for the final 3D solution.

✓ **Theorem1:** $WL(x_{2D}^*) \leq WL(x_{2\to 3D}) \leq WL(x_{3D}^*)$

Tier Optimization

■ Goal:

- **Consider MIV Density + Wirelength**: Changes in the vertical coordinates not only affect **#terminals** but also lead to **additional wirelength** changes caused by terminals.
- **Optimized from two perspectives of coarse-grained and fine-grained**: Coarse-grained can provide a relatively good initial solution, while fine-grained can further refinement.



■ Modeling:

- Transform the problem into a **search problem**.
- By **restricting the movement direction**, consider only one linear constraint, namely the knapsack constraint.
- **Cascade Terminal Legalization:** After a movement, the newly added terminals must **have valid positions** to satisfy the terminal constraint.

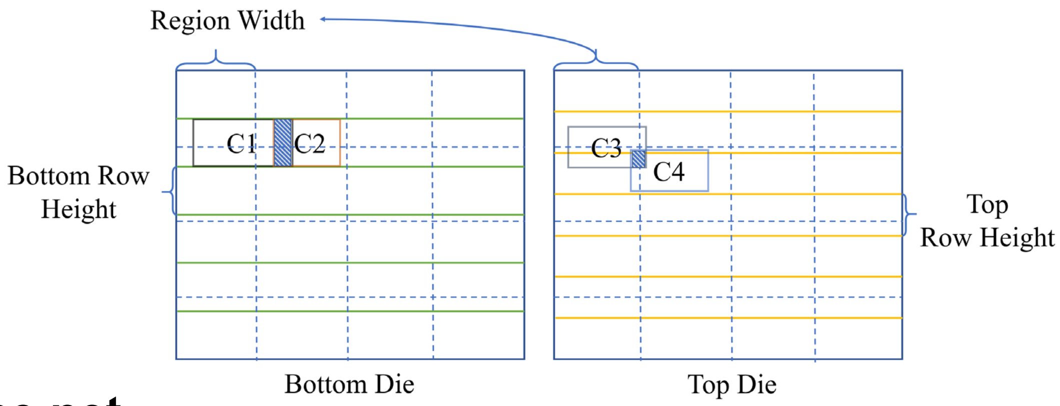
(SP1)

$$\begin{aligned} \min_{\mathbf{z}} \quad & \hat{g}(\mathbf{x}_l^k, \mathbf{z}) \\ s.t. \quad & \textstyle\sum_{i=1}^n A_1(c_i)\mathbb{I}(z_i) \le u_t A \\ & \textstyle\sum_{i=1}^n A_0(c_i)\mathbb{I}(1 - z_i) \le u_b A \\ & \textstyle\sum_{e_j \in E} \varepsilon(e_j; \mathbf{z}) \le N_t \end{aligned}$$

✓ **Optimizes (x, y) and z alternately**

Tier Optimization

- **Global Layer Optimization:**
  - **Best Improvement:**
    - Select the cell with the **highest gain** for movement.
    - Maintain priority using a **priority queue**: $\frac{p(S \cup \{c_i\}) - p(S)}{A_t(c_i)}$
    - After moving the cell, **update the priority** based on the **net** and **region relationship**.
  - **Surrogate function**:
    - **Cascade Terminal Legalization.**
    - When γ is sufficiently large, **select the region with the highest density**, and sort the remaining parts within the region based on their weights.
  - **Knapsack maximization** like priority calculation.



$$\begin{aligned} p(S \cup \{c_i\}) - p(S) = {} & \Delta\text{wirelenth} + \rho\Delta\#\text{Terminal} \\ & + \alpha\big(d(S \cup \{c_i\}) - d(S)\big) \\ & + \beta\big(o(S \cup \{c_i\}) - o(S)\big) \\ & - \gamma d(S), \end{aligned} \quad (10a)$$

where

$$d(S) = \sum_{\text{region } r} \max\left(\frac{A_r - M_r}{h_r}, 0\right),$$
$$o(S) = \sum_{i=1}^{n} \sum_{c_j \in \{c_j | \forall c_j \in V, z_j = z_i\}} \frac{\text{Overlap}(c_j, c_i)}{h_{c_i}} \quad (10b)$$

✓ **Optimizes (x, y) and z alternately**

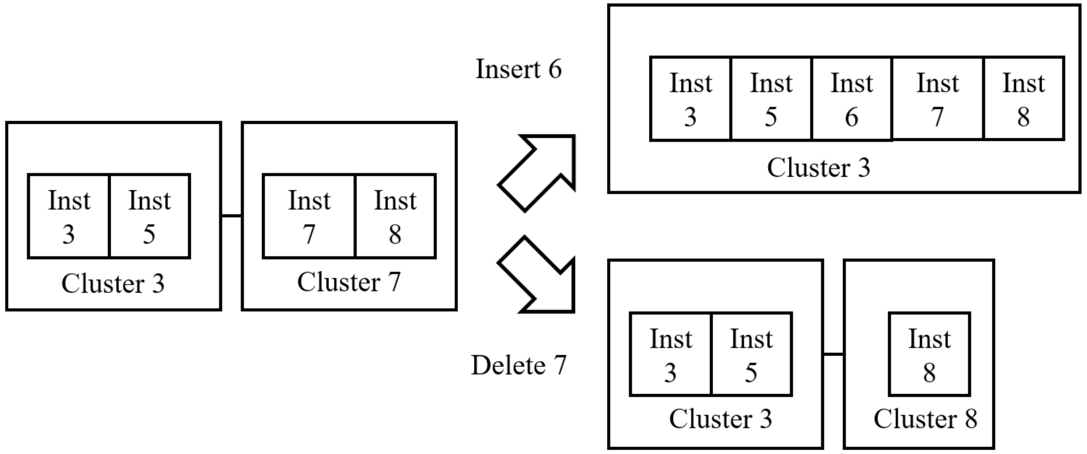
Tier Optimization

Detailed Layer Optimization

- First Improvement：
  - Select a limited number of cells for evaluation.
- **Dynamic Row-based Data Structure:**
  - Maintain the **partial order relationship** among all cells, allowing changes.
  - Implement the **insertion and deletion** of units at any position in a row.

Detailed Layer Optimization

- Dynamically maintain a legal solution for **accurate evaluation of improvements**.
- **Simple 3D Detailed Placement**:
  - Global Swap for the 3D case.
  - Quickly generate legal solutions and calculate actual gains.

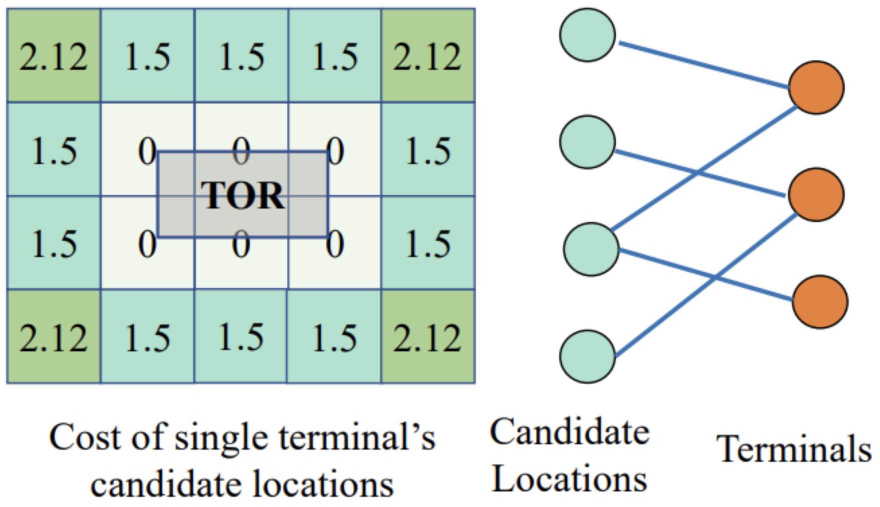


Terminal Legalization

■ Terminal Legalization:

- ■ Problem Characteristics:
  - ■ Terminals are of the same size.
  - ■ Cost calculation is independent.
- ■ Method:
  - ■ **Grid Generation:** Divide the layout into **grids** that **exactly satisfy the spacing constraint**.
  - ■ **Candidate Selection:** Select k **candidate positions around** each terminal in **its optimal region**.
  - ■ **Graph Construction and Solving**: Construct a **bipartite graph** with terminals and candidate positions, and solve it using the **network simplex algorithm**.
  - ■ **Post-processing:** Introduce **perturbations** to the placed terminals to allow for further optimization of the objective **beyond the grid.**

$$
\begin{aligned}
\min_{\mathbf{x}_t, \mathbf{y}_t} \quad & \sum_{j=1}^{m} \mathrm{WL}(e_j^- \cup \{t_j\}; \mathbf{x}_{e_j}) + \mathrm{WL}(e_j^+ \cup \{t_j\}; \mathbf{x}_{e_j}) \\
\text{s.t.} \quad & \min(|x_{t_i} - x_{t_j}|, |y_{t_i} - y_{t_j}|) \geq C, \\
& \forall i, j = 1, 2, \ldots, m.
\end{aligned}
$$



✓ **Theorem2:** $WL(x^*_{real}) \leq WL(x^*_{grid}) + 2C\#terminal$

Terminal Legalization

Terminal Legalization Upper bound:

Proof:

- From a **optimal no overlap solution** $(x_t^*,\ y_t^*)$, if you want to get a **grid solution** $(x_t',\ y_t')$, you can move the terminals **down or up** until **align the nearest grid**. At this time, the sum of all the moves in one direction is less than or equal to $\frac{mC}{2}$, and the absolute value of the slope of $WL(\cdot)$ is less than or equal to 2, so the total change in the objective function is less than or equal to $2C\#terminal$.

$$\sum_{j=1}^{m} |x'_{t_j} - x^*_{t_j}| = \min(\sum_{j=1}^{m} x^*_{t_j} \bmod C, \sum_{j=1}^{m} (C - x^*_{t_j} \bmod C)) \leq \frac{mC}{2} \quad (12)$$

✓ **Theorem2:** $WL(x^*_{real}) \leq WL\left(x^*_{grid}\right) + 2C\#\text{terminal}$

Experimental Results - Statistics

■ Public

	case1	case2	case3	case4
Die size	30 x 30	10175 x 8151	19240 x 19192	53294 x 53255
#nets	6	2644	44360	220071
#cellInsts	8	2735	44764	220845
max #inter-die terminals	4	2000	36481	183612
max u-rate of top die	80	70	78	66
max u-rate of bottom die	90	75	78	70
diff tech?	Yes	Yes	No	Yes

■ Hidden

	case2_hidden	case3_hidden	case4_hidden
Die size	11670 x 9349	17599 x 17555	55988 x 55947
#nets	2644	44360	220071
#cellInsts	2735	44764	220845
max #inter-die terminals	2000	36100	178929
max u-rate of top die	79	68	66
max u-rate of bottom die	79	78	76
diff tech?	No	Yes	Yes

Experimental Results

■ Overview:

① Compared to the top three competitors, there is an improvement in wirelength of **4.33%, 4.42%, and 5.88%,** respectively. The speed is **1.84x** faster than the first-place competitor.

② **#Terminals used is the lowest, with improvements of 79.61%, 16.74%, and 15.76% compared to the top three competitors.**

③ **The final result shows an increase in wirelength of 7.63% compared to Flatten GP (Theorem 1).**

TABLE I
EXPERIMENTAL RESULTS ON ICCAD 2022 CONTEST BENCHMARKS.

Case	Flattened	3th			2nd			1st			Ours		
	GP	HPWL	#Terminal	CPU(s)	HPWL	#Terminal	CPU(s)	HPWL	#Terminal	CPU(s)	HPWL	#Terminal	CPU(s)†
case2	1758214	2097487	**163**	10	2080647	477	14	2072075	1131	45	**1992499**	461	45
case2_hidden	2111322	2644791	**151**	9	2735158	687	15	2555461	1083	40	**2530195**	658	53
case3	26474613	33063568	14788	145	30969011	11257	437	30580336	16820	635	**30234112**	**9612**	442
case3_hidden	24200040	28372567	11211	133	27756492	8953	482	27650329	16414	412	**26939286**	**8203**	479
case4	248129463	281378079	46468	925	274026687	51480	3284	281315669	84069	2580	**267381744**	**43140**	1078
case4_hidden	272085522	307399565	58860	983	308359159	59896	3283	301193374	84728	2239	**289541474**	**51641**	1144
N.Total	-7.63%	5.88%	15.76%	0.68	4.42%	16.74%	2.32	4.33%	79.61%	1.84	**0.00%**	**0.00%**	1.00

Experimental Results—Terminal Legalization

- **Terminal Legalization:**
  - **TOR (Terminal Optimal Region):** Terminals are in the optimal positions where allows the existence of overlap.
  - **Conclusion**: In practice, the difference between the final results and the upper bound **is typically less than 0.5%. It's almost near optimal** (Theorem 2).

Case	C	#Terminal	WL	CPU(s)	TOR	Ratio
case2	200	461	1992499	1	1981785	0.54%
case2_hidden	228	658	2530195	1	2512837	0.69%
case3	100	9612	30234112	7	30141038	0.31%
case3_hidden	92	8203	26939286	5	26875050	0.24%
case4	124	43140	267381744	15	266850007	0.20%
case4_hidden	132	51641	289541474	16	288659033	0.30%

Experimental Results - Ablation Study

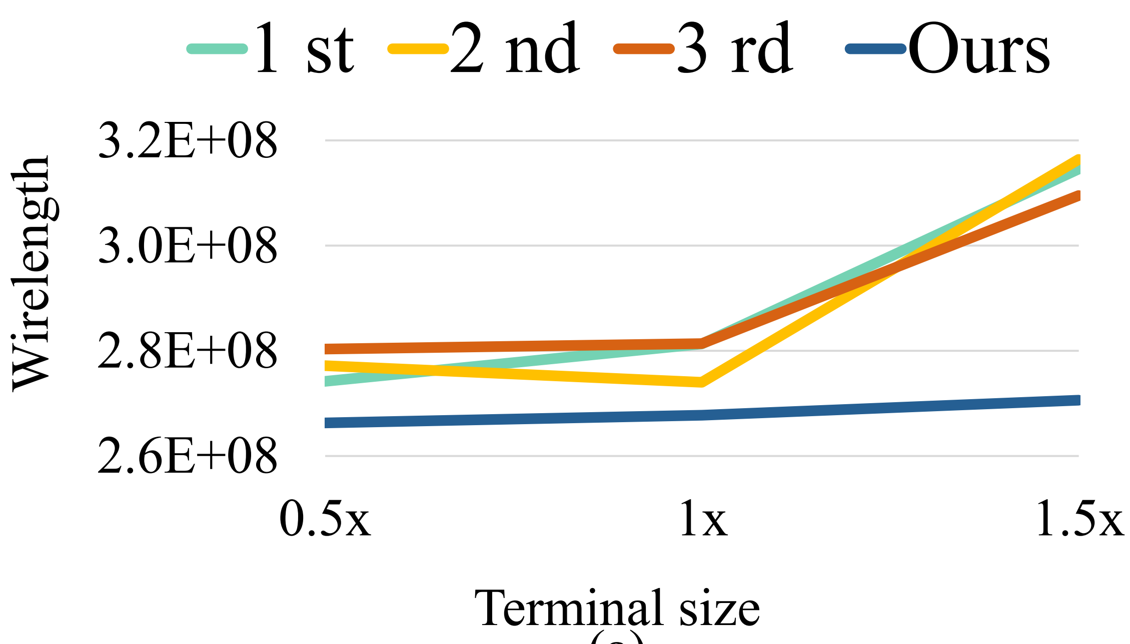
Ablation Study:

- Investigating the **impact** of **information exchange** through alternating iterations.
- **w/ Alternating Opt：** Allows alternating optimization and mutual information propagation through alternating iterations.
- **w/o. Alternating Opt：** Does not allow alternating iterations.。
- **Conclusion:** Alternating iterations enable **information exchange**, thereby **further optimizing the objective while using fewer terminals.**

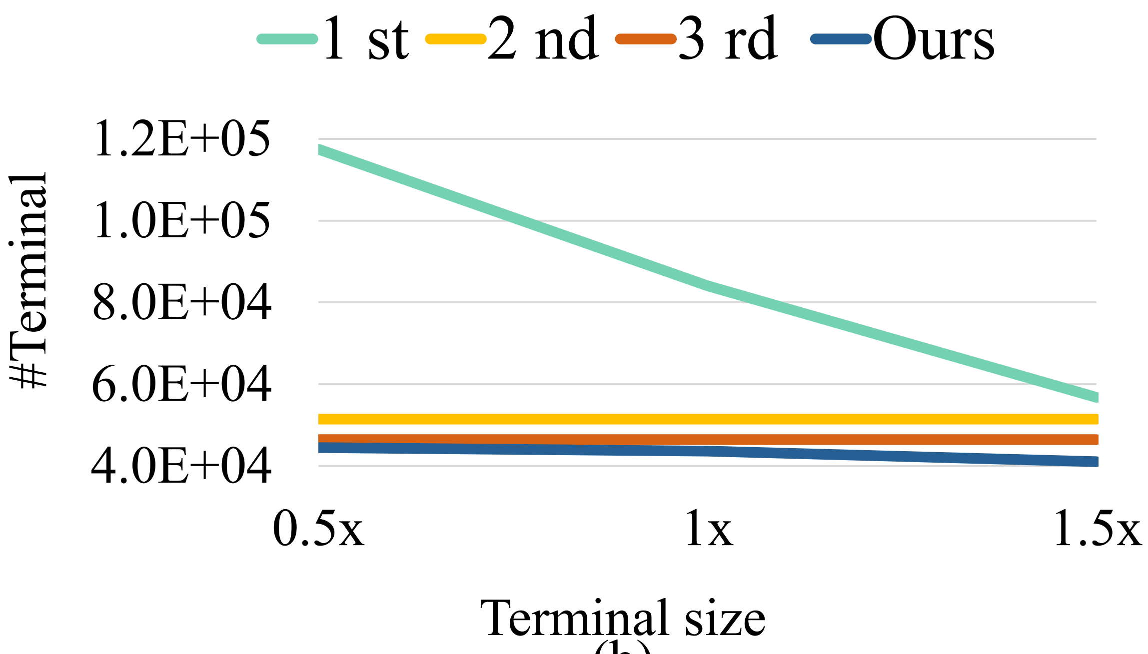
Case	w/o. Alternating Optimization.			w/ Alternating Optimization		
	HPWL	#Terminal	CPU(s)	HPWL	#Terminal	CPU(s)
case2	2032655	555	20	1992499	461	45
case2_hidden	2562890	793	19	2530195	658	53
case3	30332531	10604	135	30234112	9612	442
case3_hidden	26935732	9288	128	26939286	8203	479
case4	270042122	54112	604	267381744	43140	1,078
case4_hidden	294923683	63283	637	289541474	51641	1,144
N.Total	1.33%	21.91%	0.48	0.00%	0.00%	1.00

Experimental Results - Terminal Size Changes

- **Additional Experiment:**
  - **Left Figure:** Our method has certain advantages in both trend and quality when the terminal size changes.
  - **Right Figure:** Our algorithm can perceive the changes in terminal size and adaptively adjust the number of terminals.



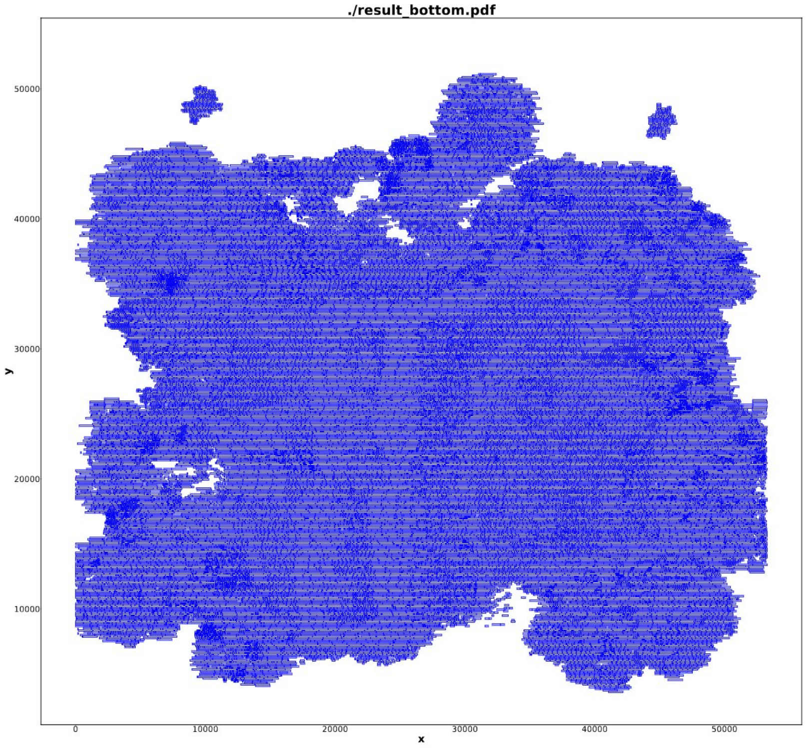
(a)



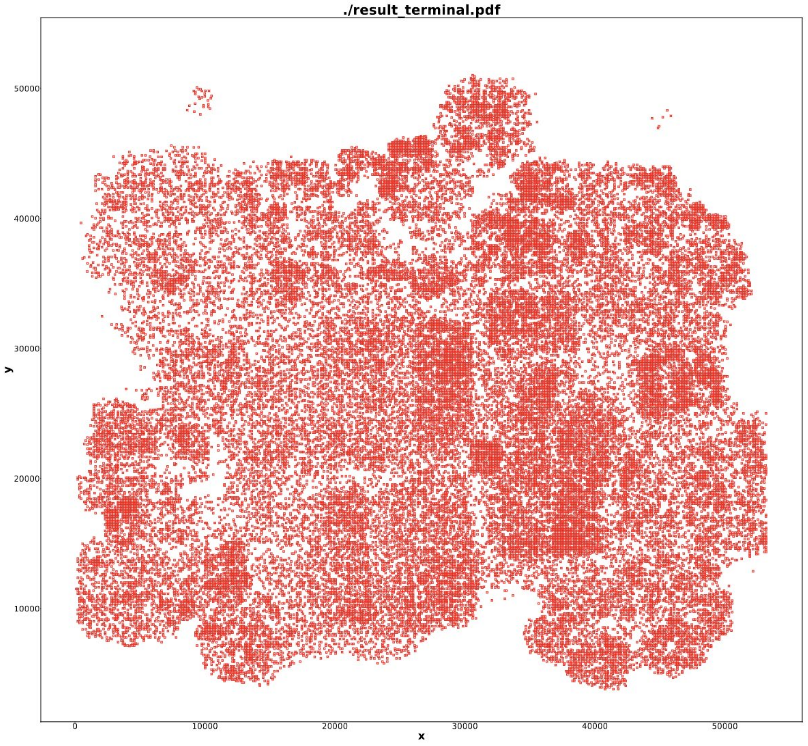
(b)

实验结果-展示

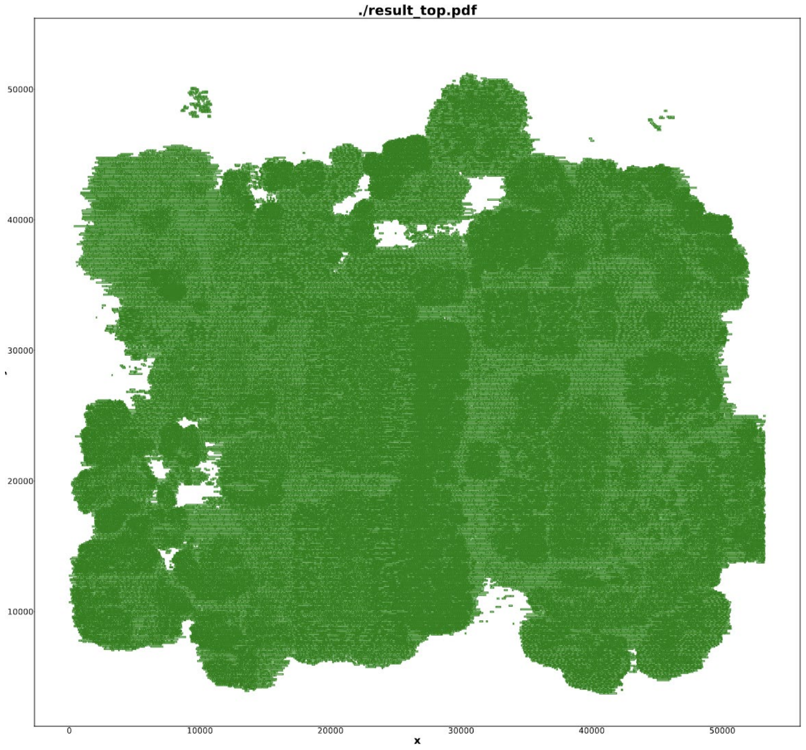
■ Case4 结果展示：



Bottom



Terminal



Top

Conclusion

- **Contributions:**
  - We propose **a novel Bilevel programming modeling** approach for the D2D Placement problem.
  - We present a **complete iterative optimization framework** to solve the Bilevel programming problem.
  - We introduce a **parallel partition algorithm** that considers comprehensive objectives, as well as a **near-optimal MIV Assignment algorithm**.
  - Compared to the top three competitors, we achieve **up to a 5.88% improvement in wirelength** and a **79.61% reduction in the number of terminals**.
- **Discussion:**
  - The analysis of the **initial solution** is still a little insufficient.
  - **Lack** of process information to assess **improvement in actual timing**.