NJUAI 南京大学 人工智能学院
SCHOOL OF ARTIFICIAL INTELLIGENCE, NANJING UNIVERSITY

LAMDA
Learning And Mining from DatA
http://www.lamda.nju.edu.cn

南京大學 NANJING UNIVERSITY 1902

黑箱优化赋能芯片设计自动化

钱超

南京大学
人工智能学院

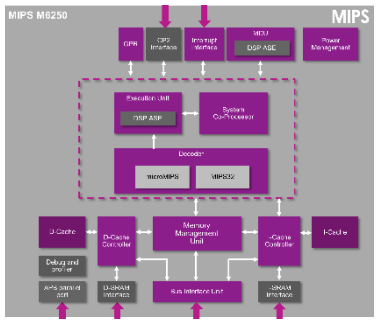
芯片自动化设计

功能设计: 设计 RTL 并验证其功能 (文档 → RTL)

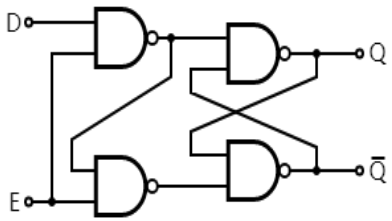
逻辑综合: 将 RTL 设计映射为网表信息 (RTL → 网表)

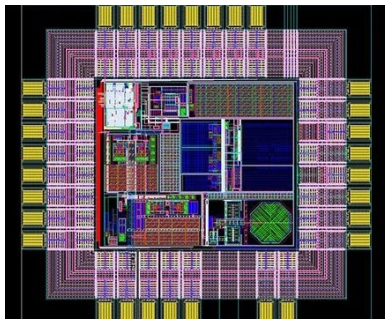
物理设计: 基于网表设计 GDS 物理版图 (网表 → GDS)

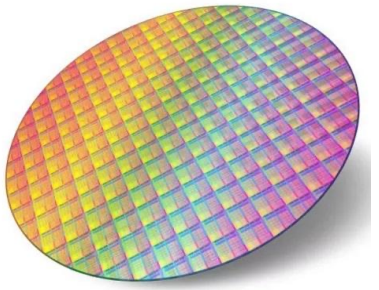
芯片制造: 通过光刻技术从 GDS 版图制造芯片 (GDS → 芯片产品)

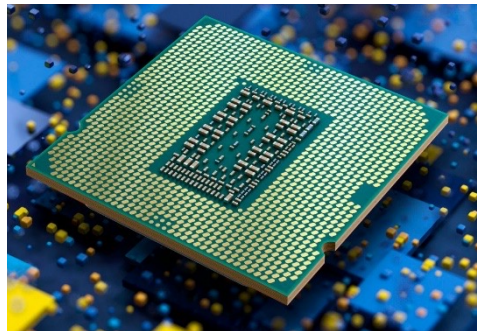



```
module alu32(Result, ALUOp, A, B, Zero) ;
  output [`ALULEN:0] Result;
  reg    [`ALULEN:0] Result;
  output Zero;
  reg    Zero;
  input  [2:0]  ALUOp;
  input  [`ALULEN:0] A, B;

  always @(A or B or ALUOp)
  begin
        case (ALUOp)
          3'b000: Result = A & B ;//and
          3'b001: Result = A | B ;//or
//       add your code here for addition, subtrac

        endcase

//      add your code here for Zero detect
  end
endmodule
```









芯片设计存在大量复杂黑箱优化问题

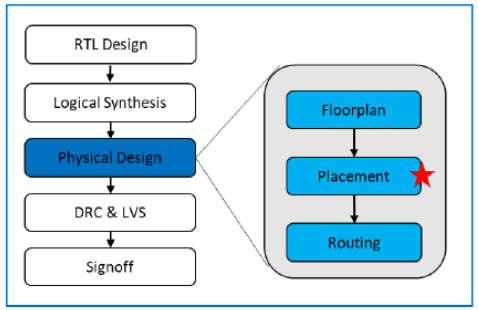
华为EDA领域的“揭榜挂帅”难题

难题4：布局布线优化技术

出题组织：海思/诺亚方舟实验室　　接口专家：许思源 xusiyuan520@huawei

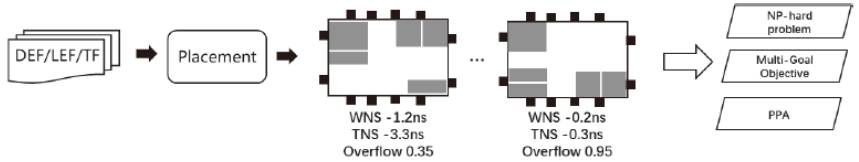
技术背景

芯片布局起着承上（逻辑综合）启下（布线）的作用，是现代超大规模集成电路设计物理设计流程中的一步，显著影响芯片设计的迭代周期。由于芯片布局被认为是NP-Complete的问题，因此如何快速生成高质量的芯片布局是布局问题的重要挑战。



Chip Design Flow　　Auto Placement and Route (APR) Flow

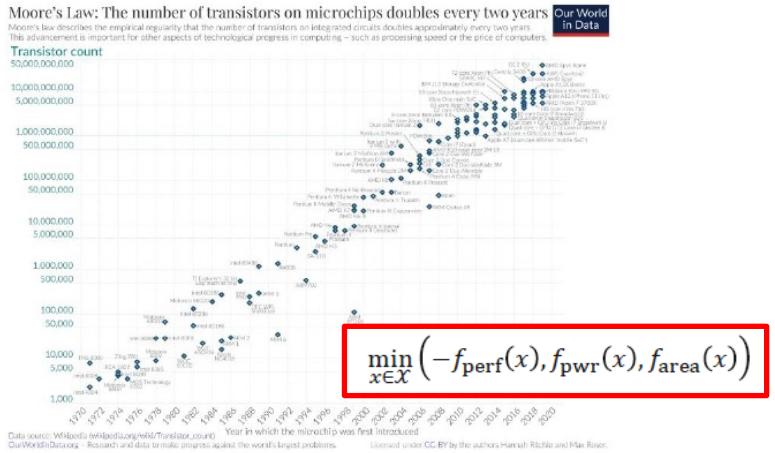
芯片布局问题可以描述为给定标准单元和宏单元的大小和连接关系，给出约束(如没有元件重叠)和优化目标(时序，拥塞，功耗，面积等)，基于特定策略确定每个单元的位置。



难题5：超高维空间多目标黑盒优化技术

出题组织：海思/诺亚方舟实验室　　接口专家：胡守博 hushoubo@huawei.

技术背景

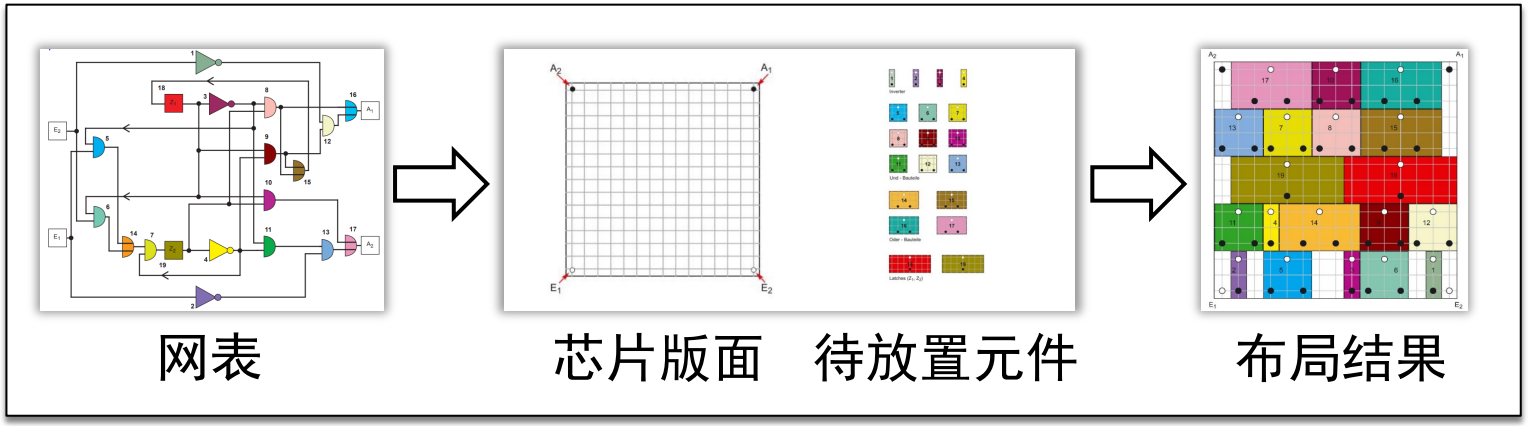


[Reference: https://en.wikipedia.org/wiki/Moore%27s_law#/media/File:Moore's_Law_Transistor_Count_1970-2020.png]

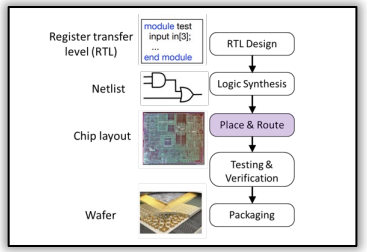
- 随着芯片制程日益演进，芯片微架构设计空间呈现几何增长导致设计空间爆炸的问题，因此无法通过穷举的方式确定最优的架构空间参数配置。
- 传统的最优参数选取依靠架构师选取有潜力的参数配置进行仿真确定，但随着设计空间的膨胀，专家经验的局限性导致人工结果距最优配置差距越来越大。
- 现有自动化最优参数寻优方案主要依赖黑盒优化算法（如贝叶斯优化，MCTS 等），但实际任务中的超高维设计空间寻优仍未很好解决，是当前主要挑战之一。

以芯片布局为例

芯片布局

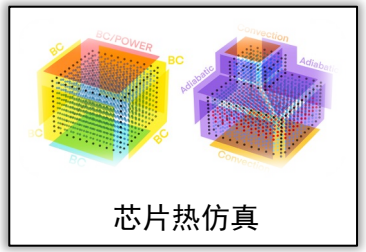


性能评估
冗长复杂



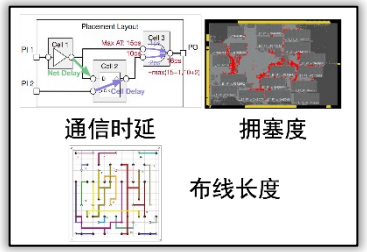
目标黑箱

模拟仿真
非常耗时



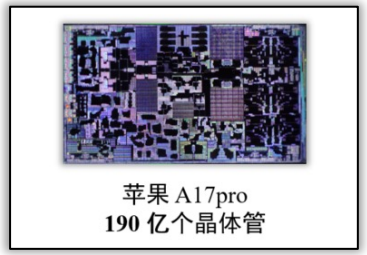
评估昂贵

涉及多个
性能指标



目标多样

芯片元件
数目庞大



规模庞大

黑箱优化

优化问题一般形式：

解 $\boldsymbol{s}$ ∈ S，目标函数 $(f_1(\boldsymbol{s}), f_2(\boldsymbol{s}), \ldots, f_m(\boldsymbol{s}))$

$$\min_{\boldsymbol{s}\in S} (f_1(\boldsymbol{s}), f_2(\boldsymbol{s}), \ldots, f_m(\boldsymbol{s}))$$

$$\text{s.t.} \quad g_i(\boldsymbol{s}) = 0, \quad 1 \le i \le q;$$
$$h_i(\boldsymbol{s}) \le 0, \quad q+1 \le i \le m$$

（约束条件）

黑箱优化：

可获取目标评估结果

$\boldsymbol{s}$ → ■ → $f(\boldsymbol{s})$

但目标函数形式未知

目标评估昂贵

优化目标多样

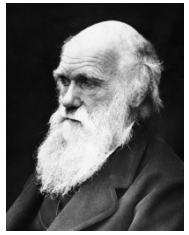
变量规模庞大

求解非常困难

希望通过少量的目标评估，获得较好的解

演化算法

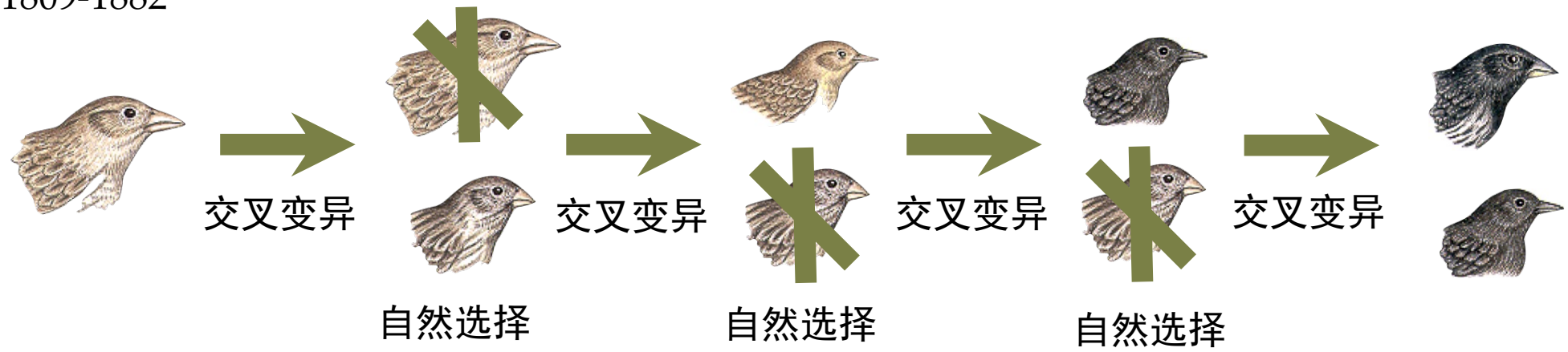
演化算法：人工智能重要研究分支，适于求解黑箱优化



达尔文

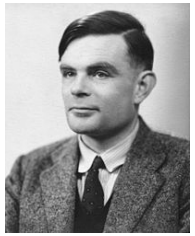
1809-1882

受自然演化过程启发产生



演化算法

演化算法：人工智能重要研究分支，适于求解黑箱优化



图灵

1912-1954

1950年，图灵描述了如何基于演化过程求解复杂问题：

building intelligent machine

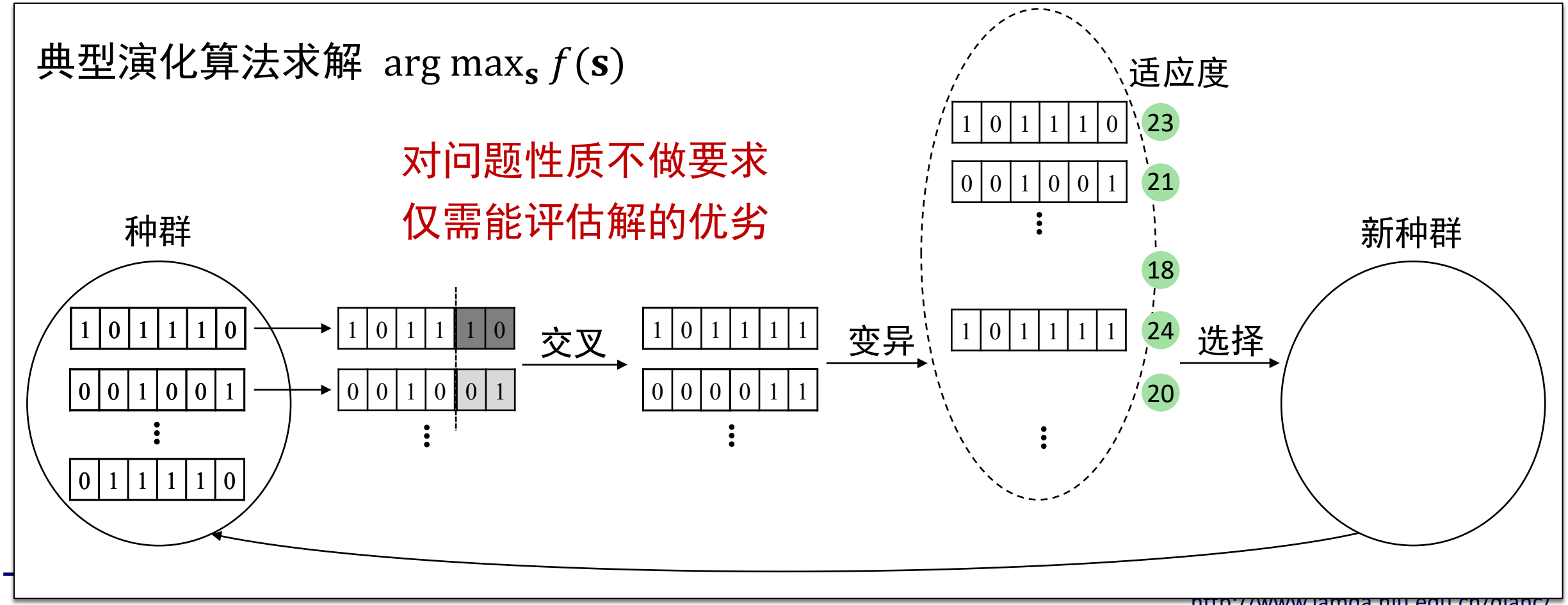
“Structure of the child machine = Hereditary material
Changes of the child machine = Mutations
Judgment of the experimenter = Natural selection”

[A. M. Turing. Computing machinery and intelligence. Mind 49: 433-460, 1950.]

被《Nature》2015年机器智能专刊列为六大代表领域之一

演化算法

演化算法：人工智能重要研究分支，适于求解黑箱优化



演化算法应用案例

日本高铁头部形状设计

Series 700

演化

Series N700

能耗节省 19%

Technological overview of the next generation Shinkansen high-speed train Series N700

M. Ueno[1], S. Usui[1], H. Tanaka[1], A. Watanabe[2]

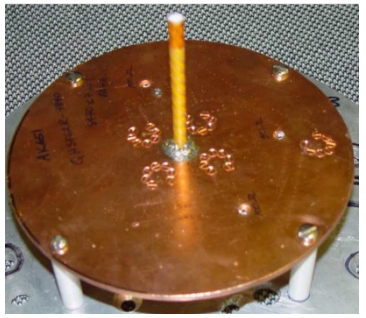
[1]Central Japan Railway Company, Tokyo, Japan, [2]West Japan Railway Company, Osaka, Japan

waves and other issues related to environmental compatibility such as external noise. To combat this, an aero double-wing-type has been adopted for nose shape (Fig. 3). This nose shape, which boasts the most appropriate aerodynamic performance, has been newly developed for railway rolling stock using the latest analytical technique (i.e. genetic algorithms) used to develop the main wings of airplanes. The shape resembles a bird in flight, suggesting a feeling of boldness and speed.

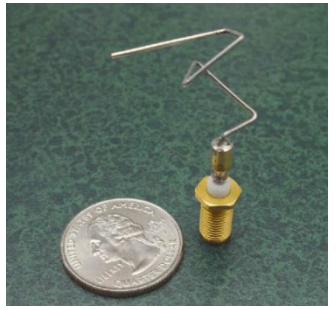
On the Tokaido Shinkansen line, Series N700 cars save 19% energy than Series 700 cars, despite a 30% increase in the output of their traction equipment for higher-speed operation (Fig. 4).

This is a result of adopting the aerodynamically excellent nose shape, reduced running

美国宇航局天线设计



演化



38%功效

93%功效

Computer-Automated Evolution of an X-Band Antenna for NASA's Space Technology 5 Mission

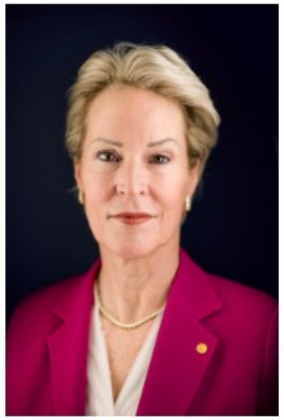
Gregory. S. Hornby Gregory.S.Hornby@nasa.gov
University Affiliated Research Center, NASA Ames Research Park, UC Santa Cruz at Moffett Field, California, 94035

Jason D. Lohn Jason.Lohn@sv.cmu.edu
Carnegie Mellon University, NASA Ames Research Park and Moffett Field, California 94035

this, different combinations of the two evolved antennas and the QHA were tried on the ST5 mock-up and measured in an anechoic chamber. With two QHAs 38% efficiency was achieved, using a QHA with an evolved antenna resulted in 80% efficiency, and using two evolved antennas resulted in 93% efficiency. Here "efficiency" means how

演化算法应用案例

The Nobel Prize in Chemistry 2018



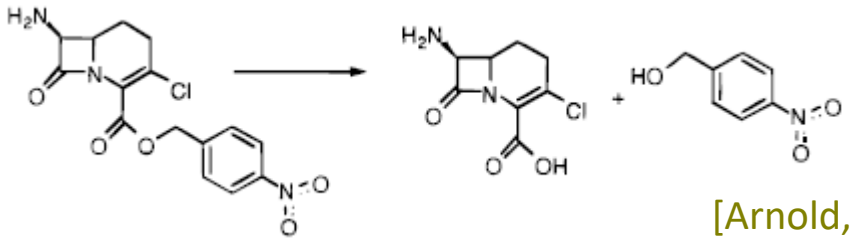
© Nobel Media AB. Photo: A. Mahmoud
Frances H. Arnold
Prize share: 1/2

© Nobel Media AB. Photo: A. Mahmoud
George P. Smith
Prize share: 1/4

© Nobel Media AB. Photo: A. Mahmoud
Sir Gregory P. Winter
Prize share: 1/4

The Nobel Prize in Chemistry 2018 was divided, one half awarded to Frances H. Arnold "for the directed evolution of enzymes", the other half jointly to George P. Smith and Sir Gregory P. Winter "for the phage display of peptides and antibodies."

蛋白质设计



[Arnold, 1998]

"Evolution—the adaption of species to different environments —has created an enormous diversity of life. Frances Arnold has used the same principles – genetic change and selection – to develop proteins that solve humankind's chemical problems. In 1993, Arnold conducted the first directed evolution of enzymes, which are proteins that catalyze chemical reactions. The uses of her results include more environmentally friendly manufacturing of chemical substances, such as pharmaceuticals, and the production of renewable fuels."

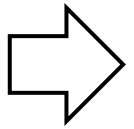
演化算法应用案例

自然科学四大基础科学问题之一：
生命起源与演化

地层剖面
海量化石
记录数据

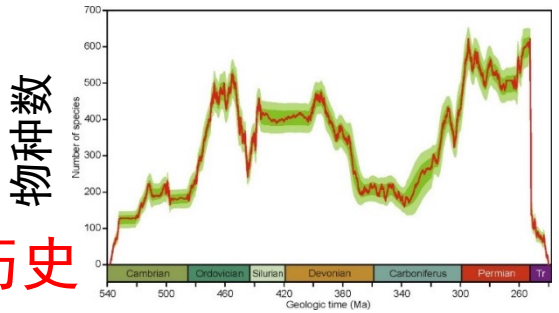
三叶虫、笔石、珊瑚、腕足…

利用化石记录



重现生命演化历史

生物多样性变化曲线



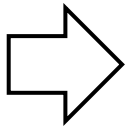
南京大学地球科学与工程学院研究成果

中国的地层剖面数据

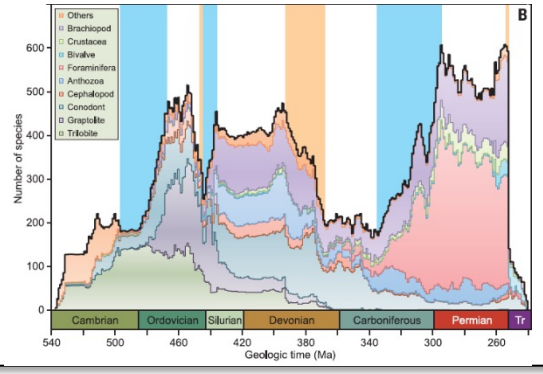
3122个剖面

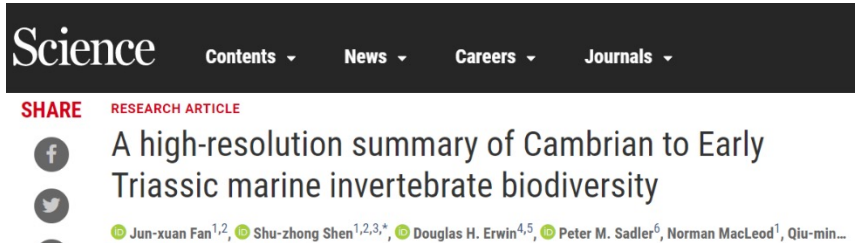
11268个物种

演化算法



全球第一条高精度海洋生物多样性变化曲线





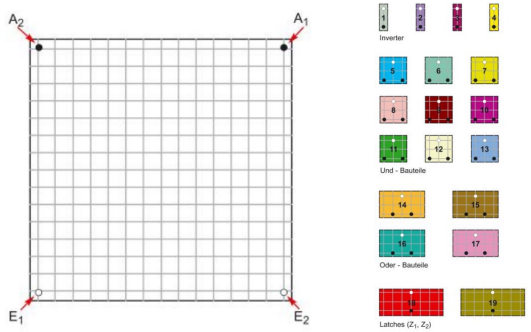
Science：“新的数据集和方法，推动整个演化生物学的变革”

Nature：“古生物学家以惊人的细节绘制地球3亿年历史”

2020 年中国科学十大进展

演化算法局限

演化算法虽能够解决复杂黑箱优化问题，但求解效率偏低



芯片布局 [Google, Nature'21]

"... it is very slow and difficult to parallelize, thereby failing to scale to the increasingly large and complex circuits of the 1990s and beyond."

生命起源与演化

[Fan et al., Science'20]

中国的地层剖面数据

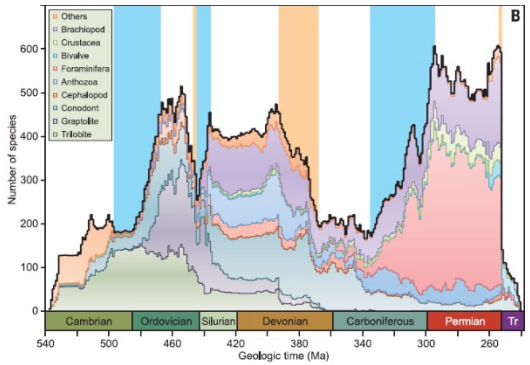
3122个剖面

11268个物种

演化算法

"天河2号"
700万核时

全球第一条高精度
海洋生物多样性变化曲线



如何提效？

演化学习

演化学习

通过结合演化算法和机器学习以更好的解决黑箱优化问题

演化算法主要基于启发式，严重缺乏理论基础，为何有效、何时有效不清楚

L. Valiant

2010年图灵奖得主

哈佛大学教授

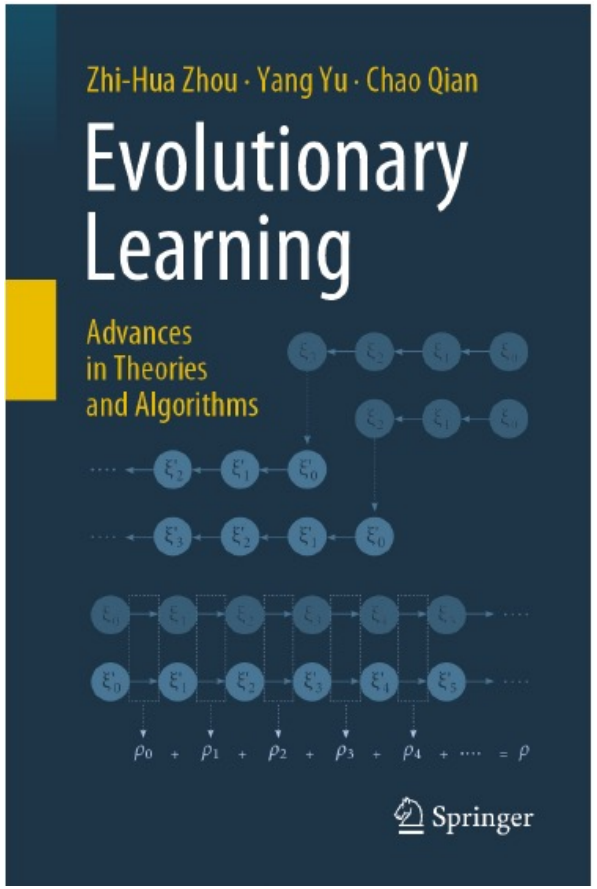
Evolvability

Journal of the ACM, Vol. 56, No. 1, Article 3,
Publication date: January 2009.

Abstract. Living organisms function in accordance with complex mechanisms that operate in different ways depending on conditions. Darwin's theory of evolution suggests that such mechanisms evolved through variation guided by natural selection. However, there has existed no theory that would explain quantitatively which mechanisms can so evolve in realistic population sizes within realistic time

"there has existed no theory that would explain quantitatively which mechanisms can so evolve in realistic population sizes within realistic time …"

我们在演化学习理论与算法方面取得进展

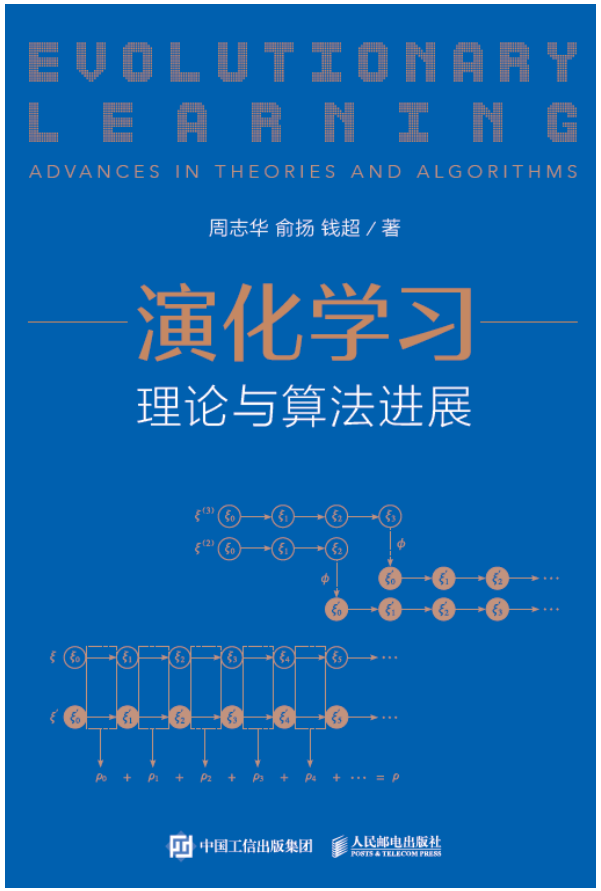


Zhi-Hua Zhou, Yang Yu, Chao Qian

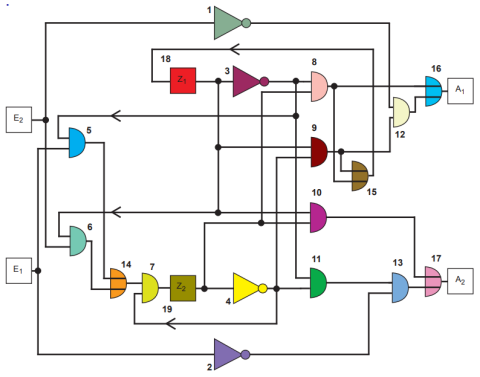
Evolutionary Learning: Advances in Theories and Algorithms

- Presents theoretical results for evolutionary learning
- Provides general theoretical tools for analysing evolutionary algorithms
- Proposes evolutionary learning algorithms with provable theoretical guarantees

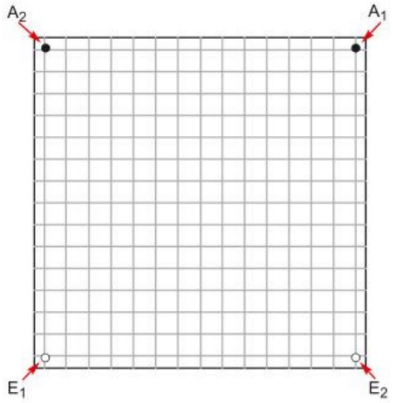
“Springer计算机学科年度最受欢迎作品”



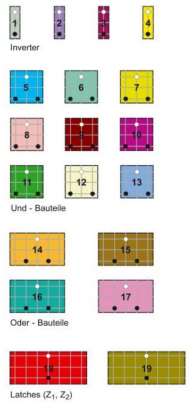
应用案例：芯片元件布局



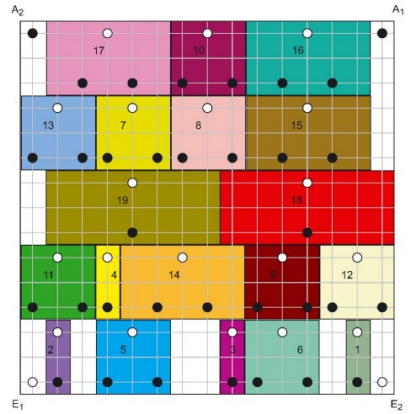
网表



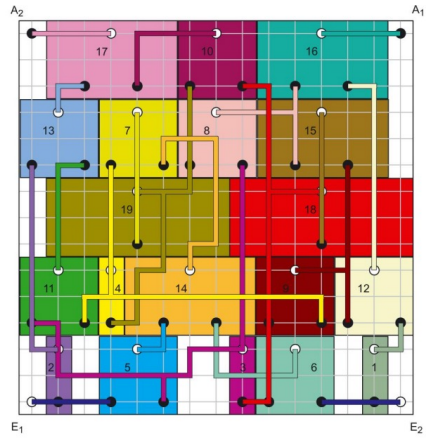
芯片版面



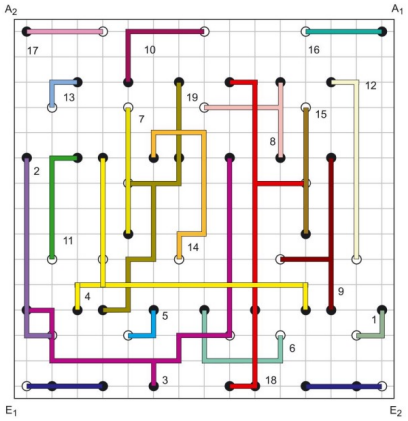
待放置元件

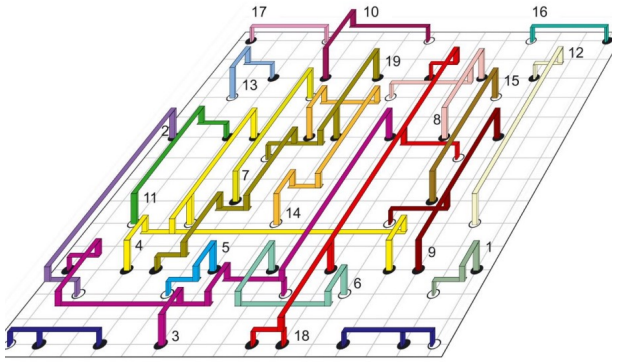


布局



布线

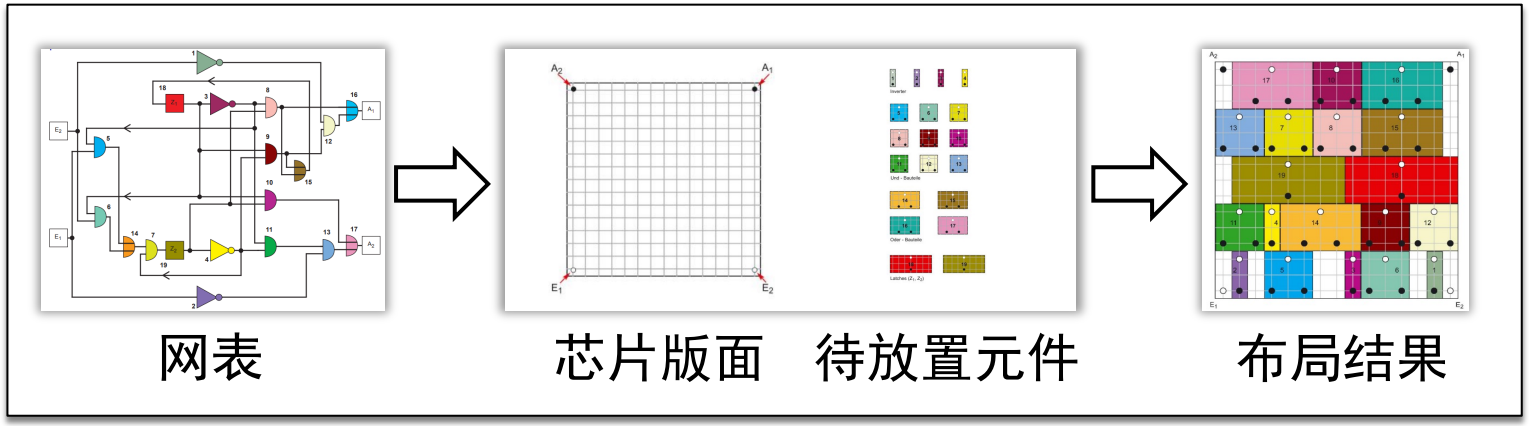




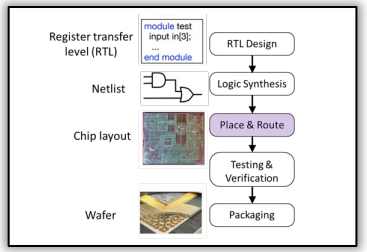
Figures are from http://www.or.uni-bonn.de/~vygen/files/buda.pdf

应用案例：芯片元件布局

芯片布局

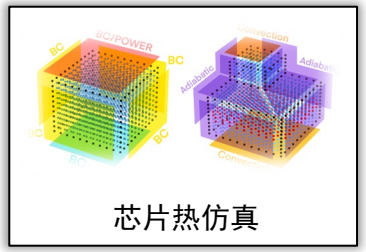


性能评估
冗长复杂



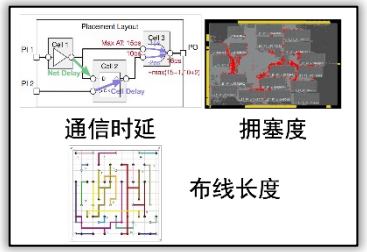
目标黑箱

模拟仿真
非常耗时



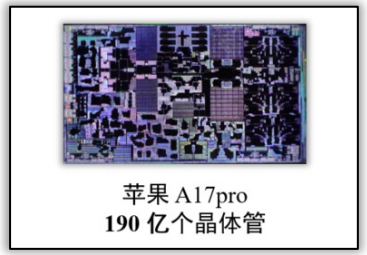
评估昂贵

涉及多个
性能指标



目标多样

芯片元件
数目庞大



规模庞大

Macro placement

Minimization objective: Half Perimeter Wire Length

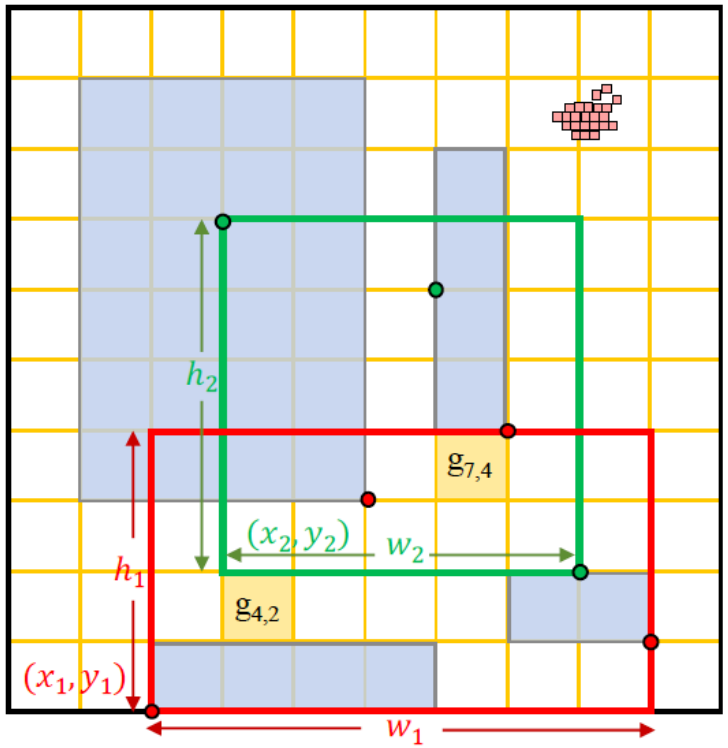
$$\mathrm{HPWL}(\boldsymbol{s}, H) = \sum\nolimits_{e_j \in E} (w_j + h_j)$$

Sum of the half perimeter of nets' bounding boxes,

which is non-differentiable

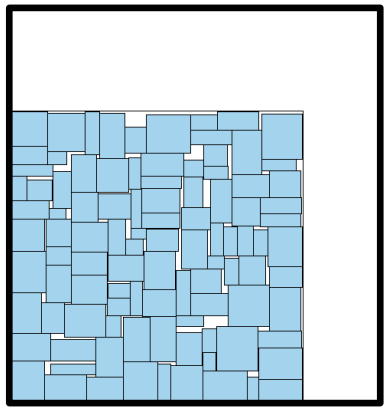
Constraint:
No-overlapping

High-dimensional:
Thousands of macros



$$\mathrm{HPWL} = w_1 + h_1 + w_2 + h_2$$

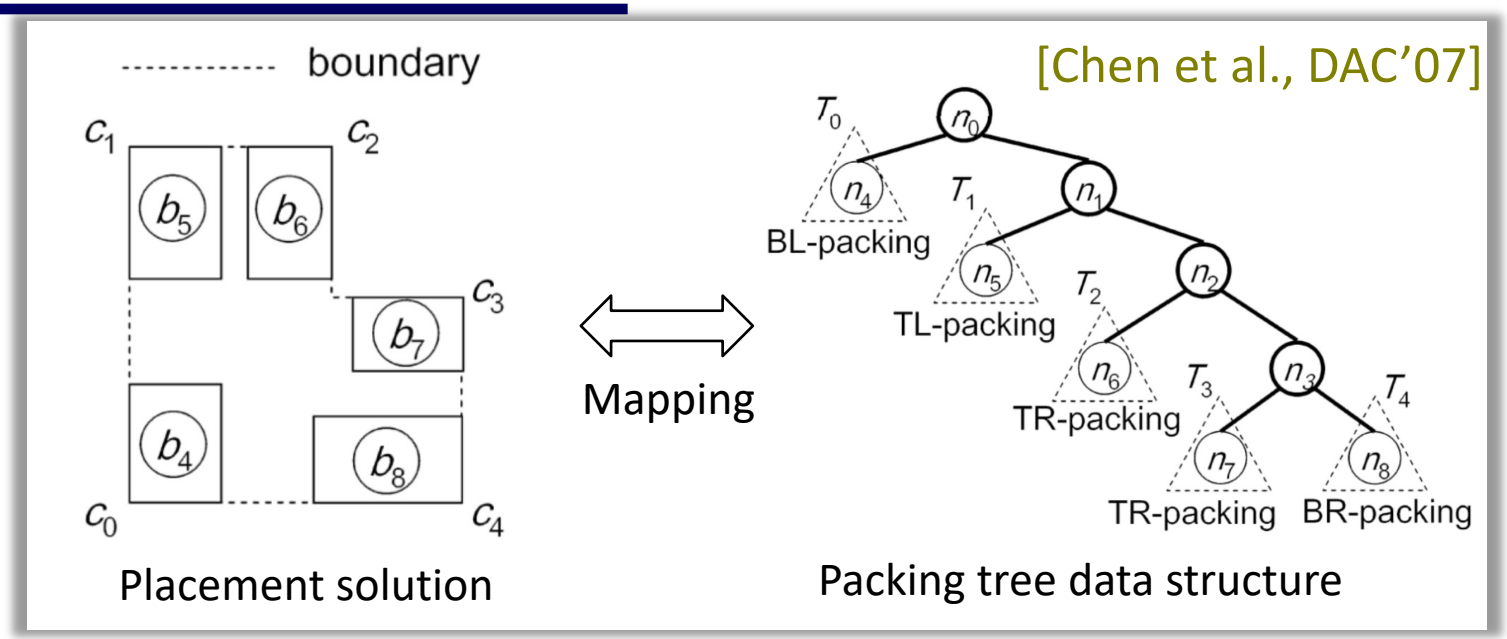
Previous methods



Classical EAs

1980s-2000s

- Packing-based solution representation
- Optimize by Evolutionary Algorithms (EAs)

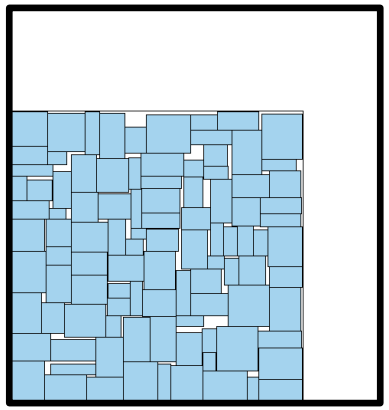


- Final performance heavily relies on the initial placement
- Low-efficiency and low-scalability

"... it is very slow and difficult to parallelize, thereby failing to scale to the increasingly large and complex circuits of the 1990s and beyond."

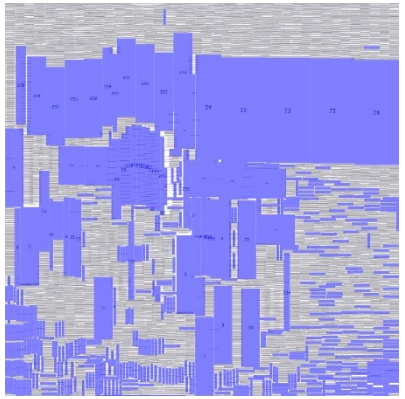
[Google, Nature'21]

Previous methods



Classical EAs

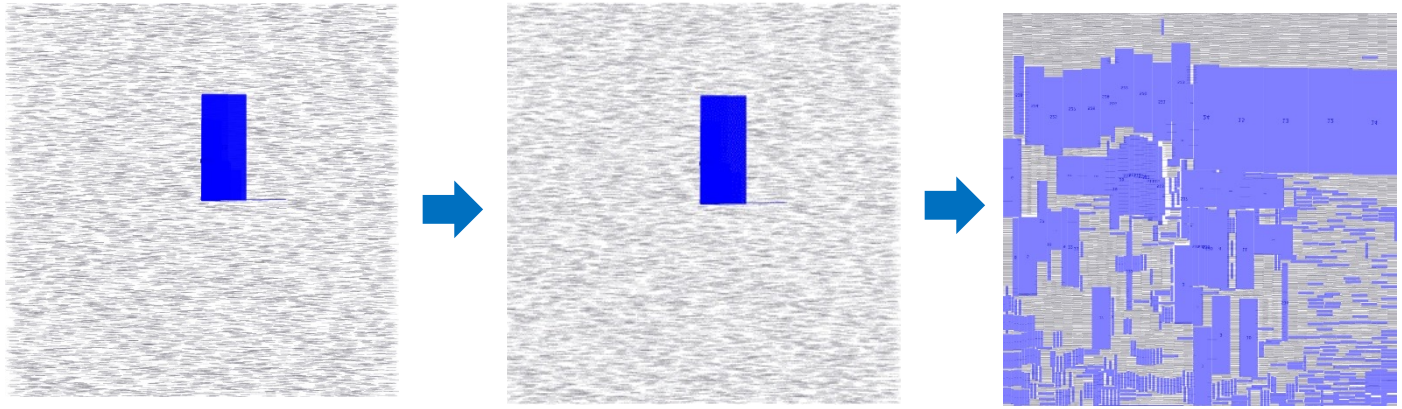
1980s-2000s



Analytical Placers

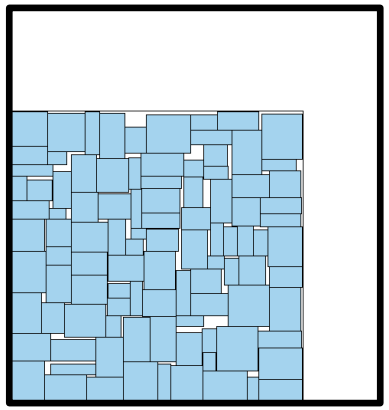
2000s-Now

- Smooth the HPWL formulation
- Optimize by gradient descent
- Overlapping
- Get stuck in local optima



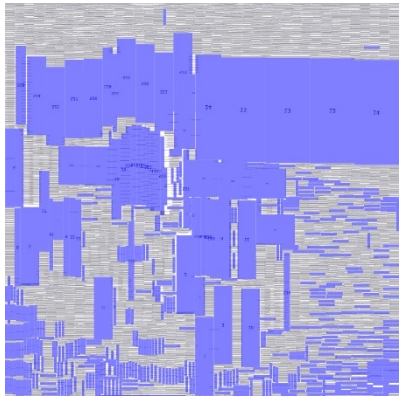
Some figures are from [Yibo Lin, DREAMPlace]

Previous methods



Classical EAs

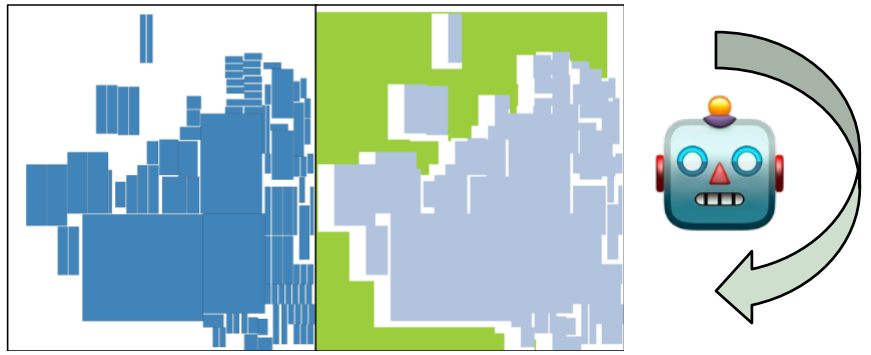
1980s-2000s



Analytical Placers

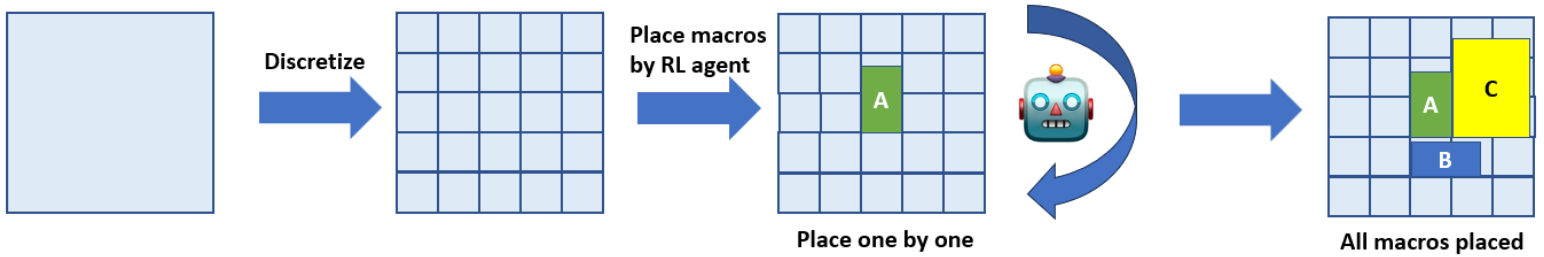
2000s-Now

[Google, Nature'21]



Reinforcement Learning Methods

2021-Now

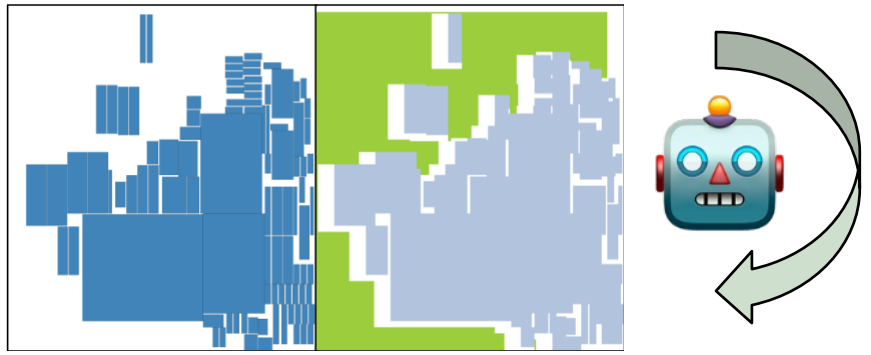


- Markov Decision Process
- Long training time
- Poor exploration

Previous methods

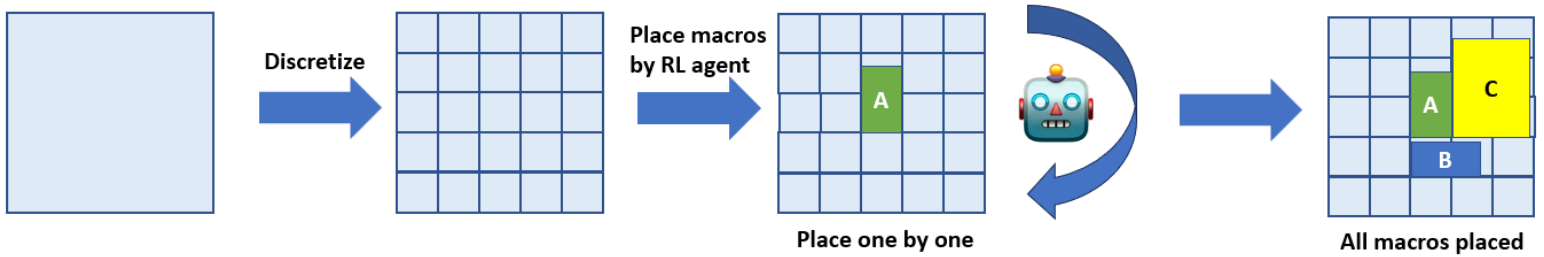
- **Graph Placement** [Google, Nature'21]
  The first RL method for macro placement, sparse reward
- **DeepPR** [Cheng et al., NeurIPS'21]
  Involve both vision and graph features, macro overlap
- **MaskPlace** [Lai et al., NeurIPS'22]
  Involve dense reward by Wiremask, converge too fast
- **ChiPFormer** [Lai et al., ICML'23]
  Involve transformer to enhance transferability, converge too fast

[Google, Nature'21]



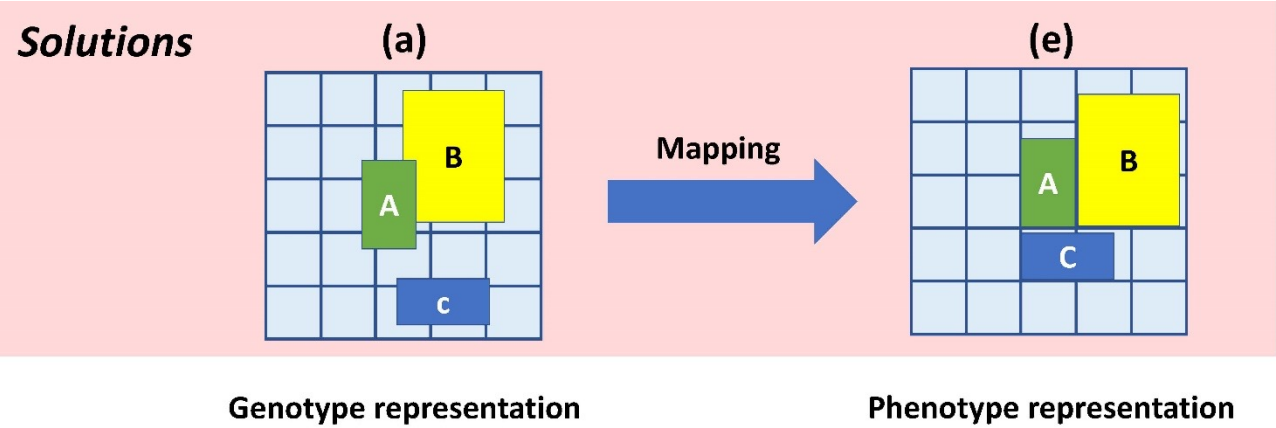
Reinforcement Learning Methods

2021-Now



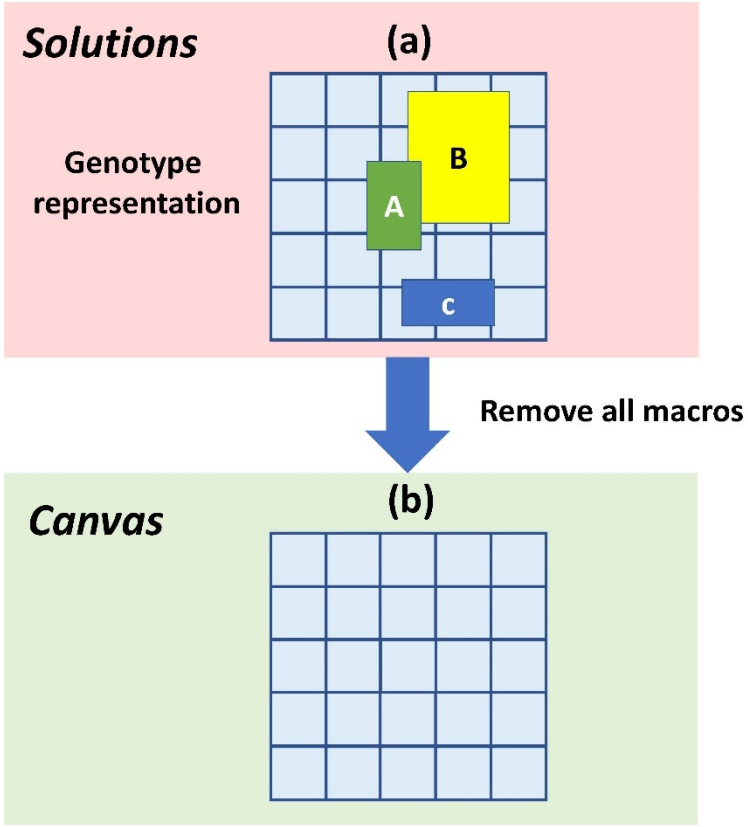
- Markov Decision Process
- Long training time
- Poor exploration

Proposed genotype-phenotype mapping



[Shi, Xue, Song, and Qian, NeurIPS 2023]

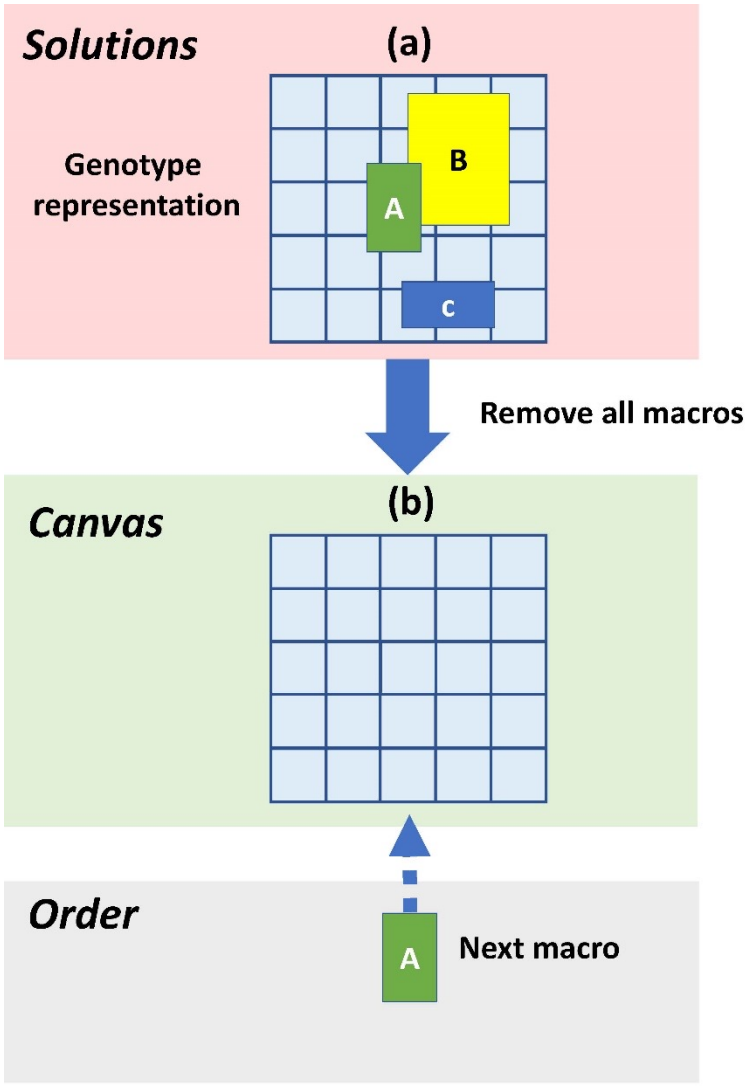
Proposed genotype-phenotype mapping



Partition the chip canvas into discrete grids

[Shi, Xue, Song, and Qian, NeurIPS 2023]

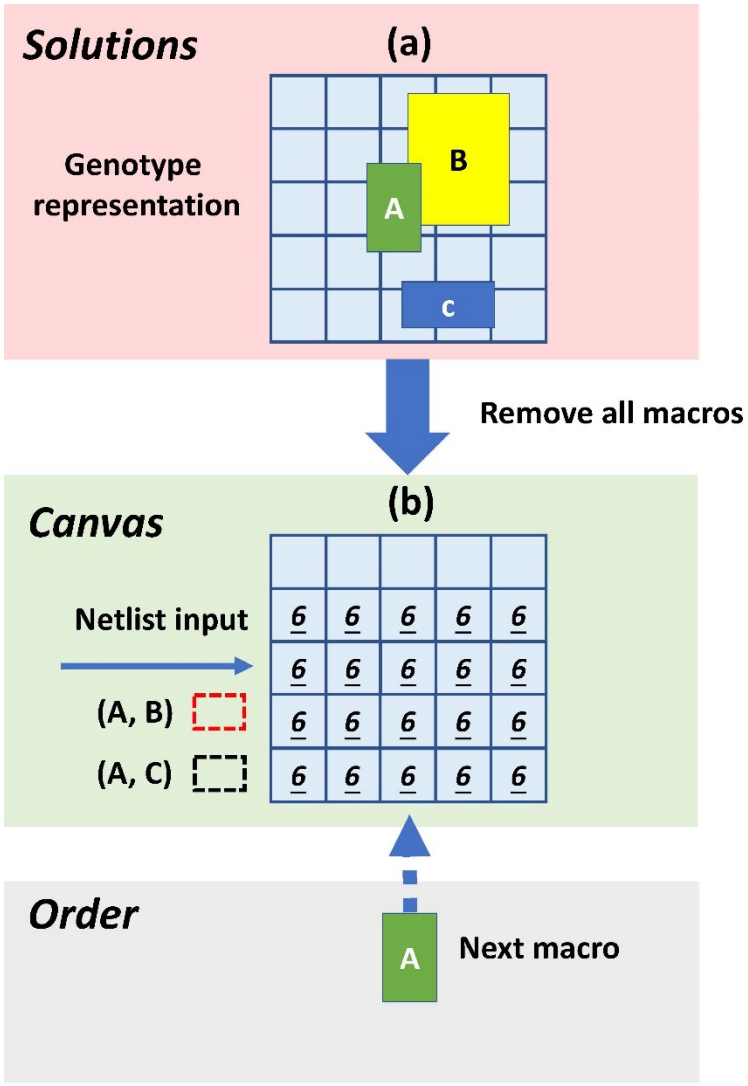
Proposed genotype-phenotype mapping



The first macro to place

[Shi, Xue, Song, and Qian, NeurIPS 2023]

Proposed genotype-phenotype mapping

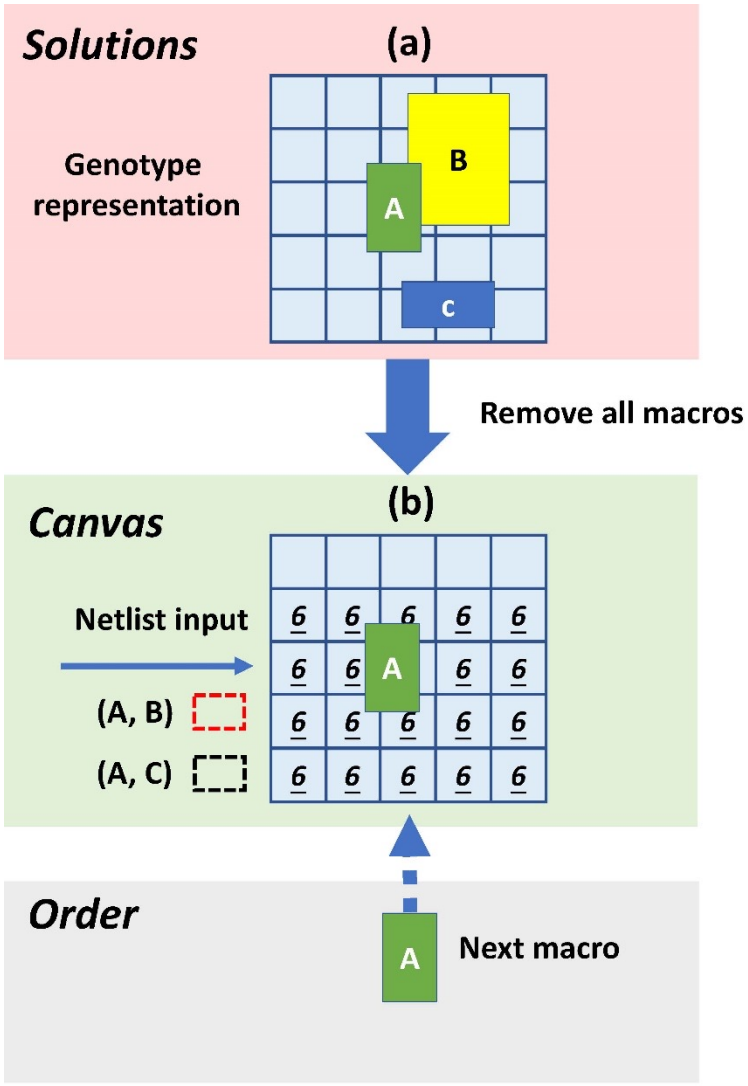


Calculate the wirelength increment after placing the macro:

- Invalid location resulting in overlap or exceeding the boundary
- **6** Wirelength increment if placing at this grid
- **6** Underlined number means the minimum increment

[Shi, Xue, Song, and Qian, NeurIPS 2023]

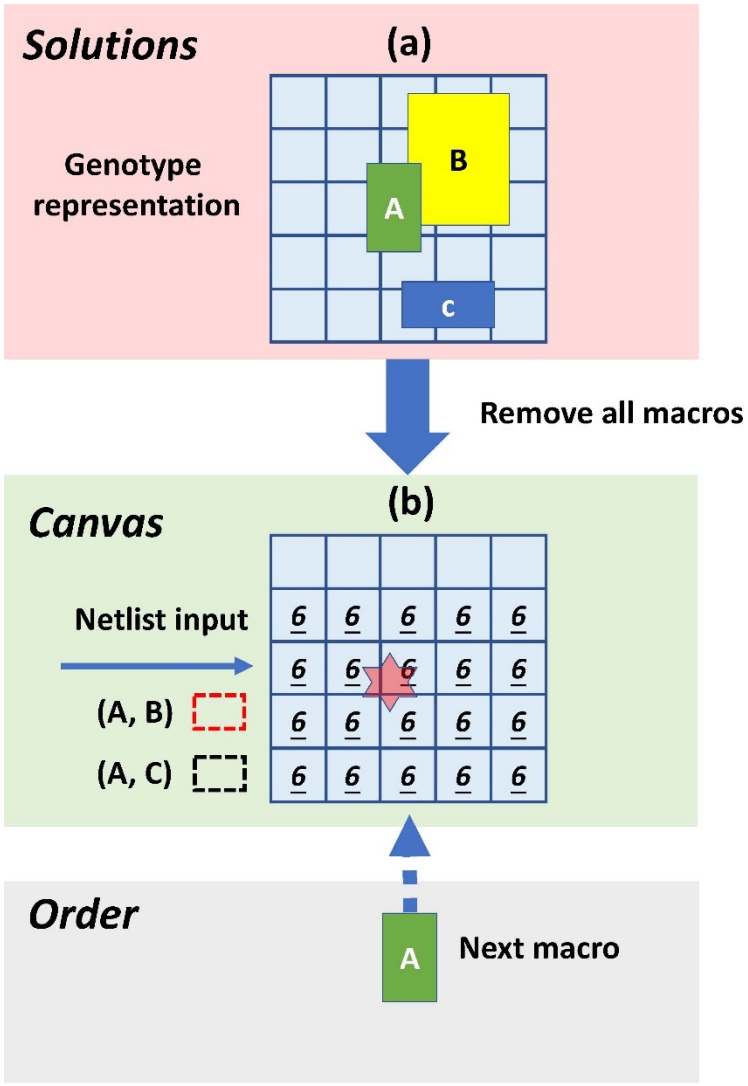
Proposed genotype-phenotype mapping



Refer to its original location

[Shi, Xue, Song, and Qian, NeurIPS 2023]

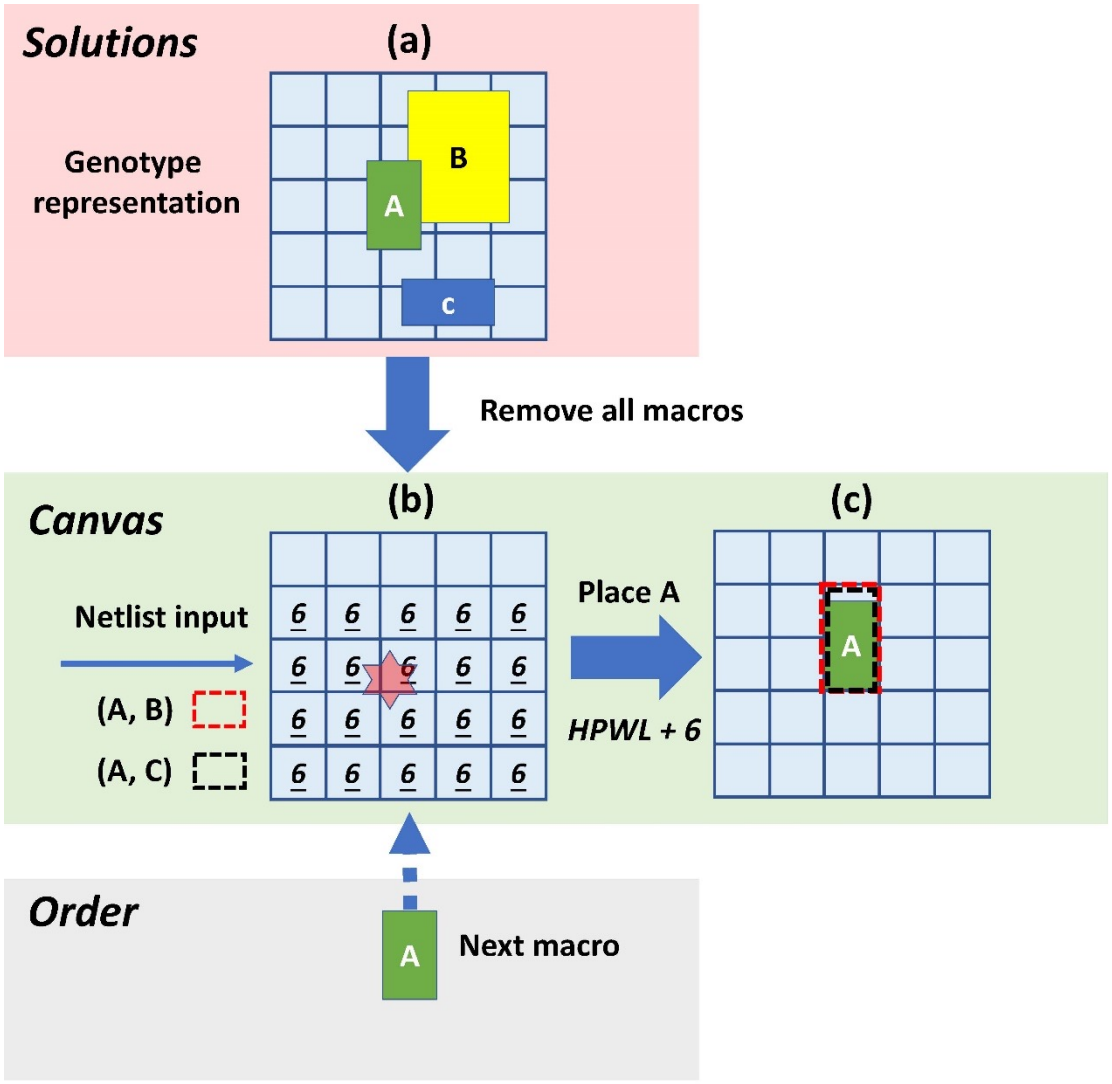
Proposed genotype-phenotype mapping



Refer to its original location

[Shi, Xue, Song, and Qian, NeurIPS 2023]

Proposed genotype-phenotype mapping

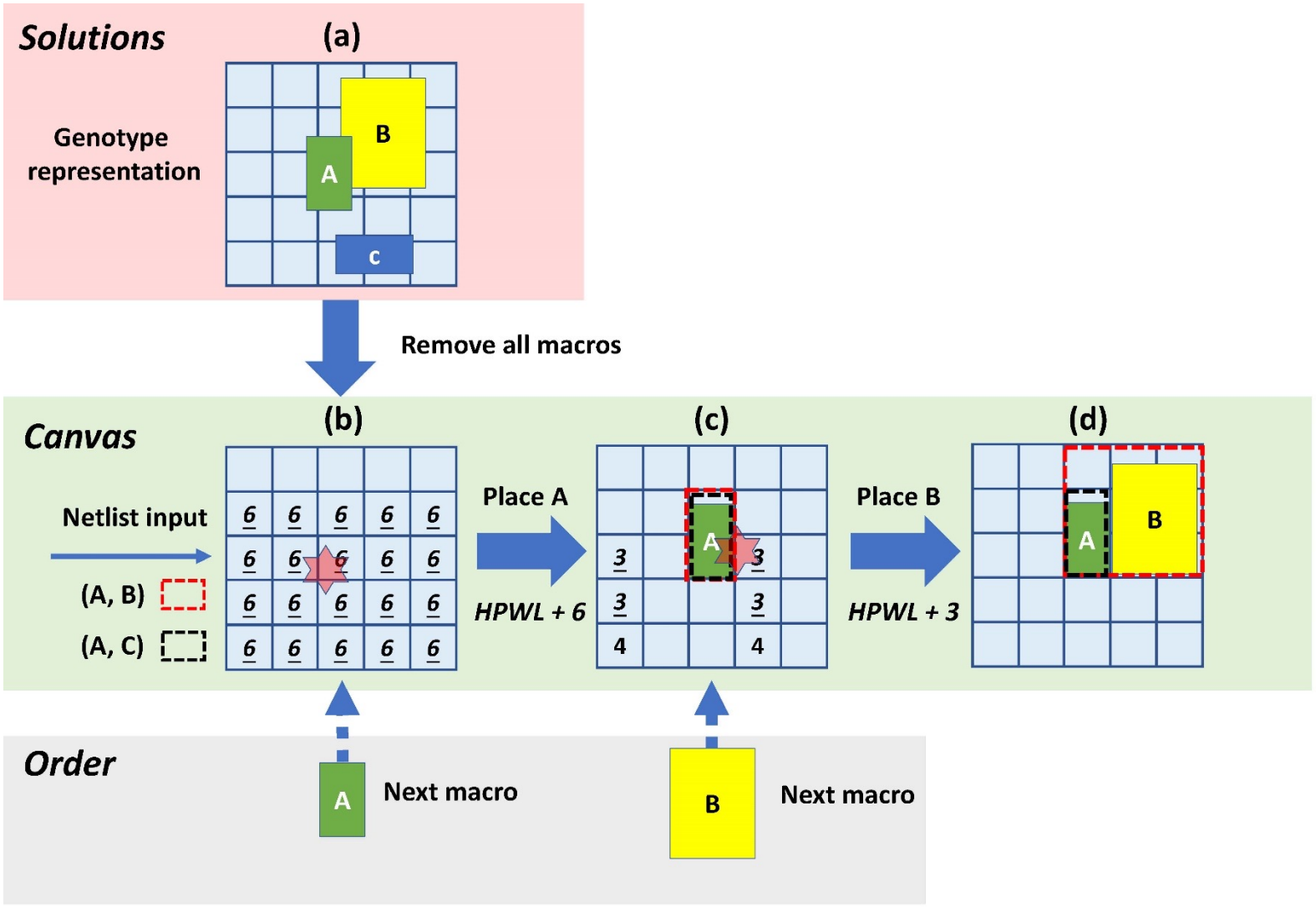


Choose the nearest best grid

Place the macro

[Shi, Xue, Song, and Qian, NeurIPS 2023]

Proposed genotype-phenotype mapping

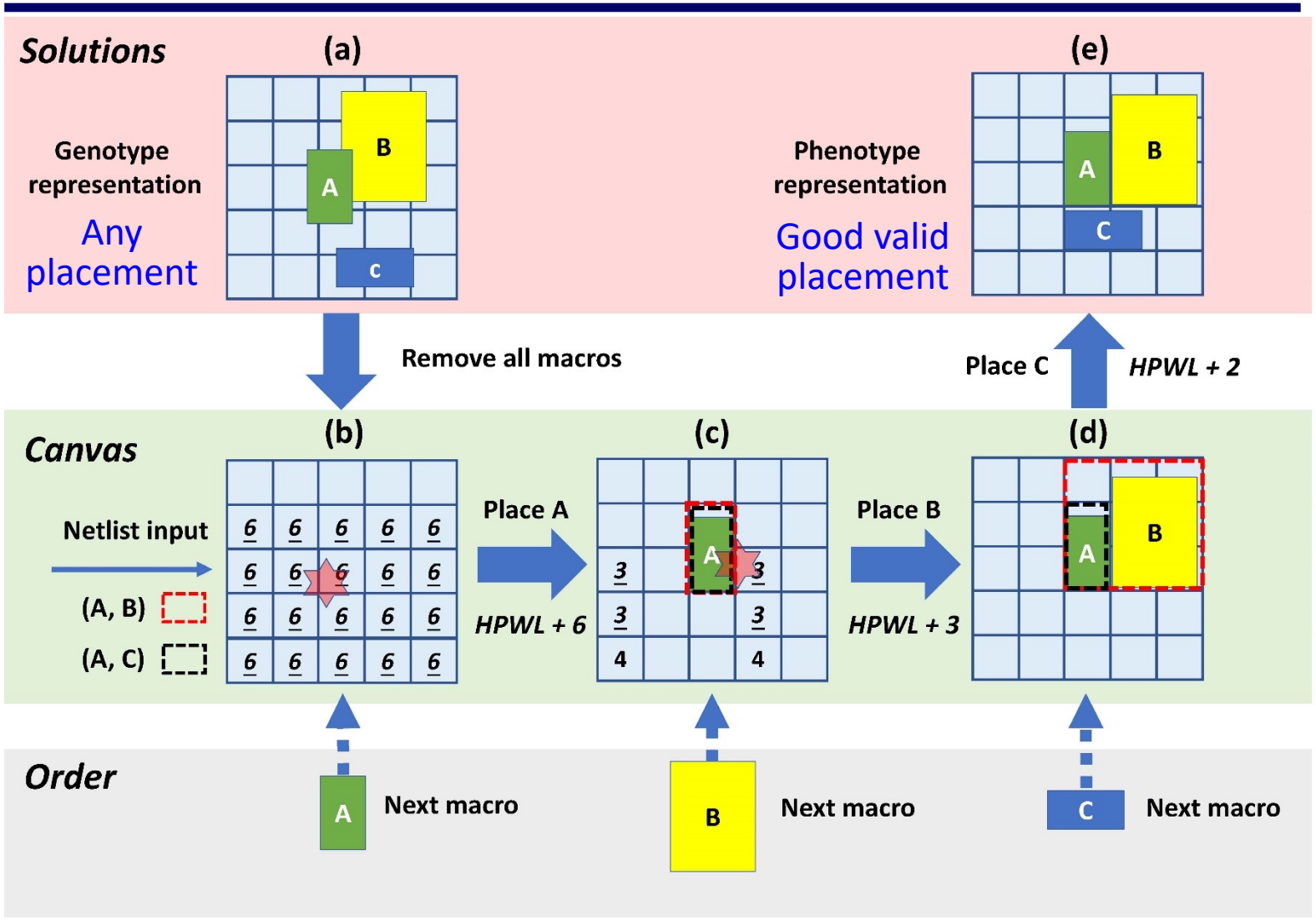


Choose the nearest best grid

Place the macro

[Shi, Xue, Song, and Qian, NeurIPS 2023]

Proposed genotype-phenotype mapping



Place macros B and C similarly

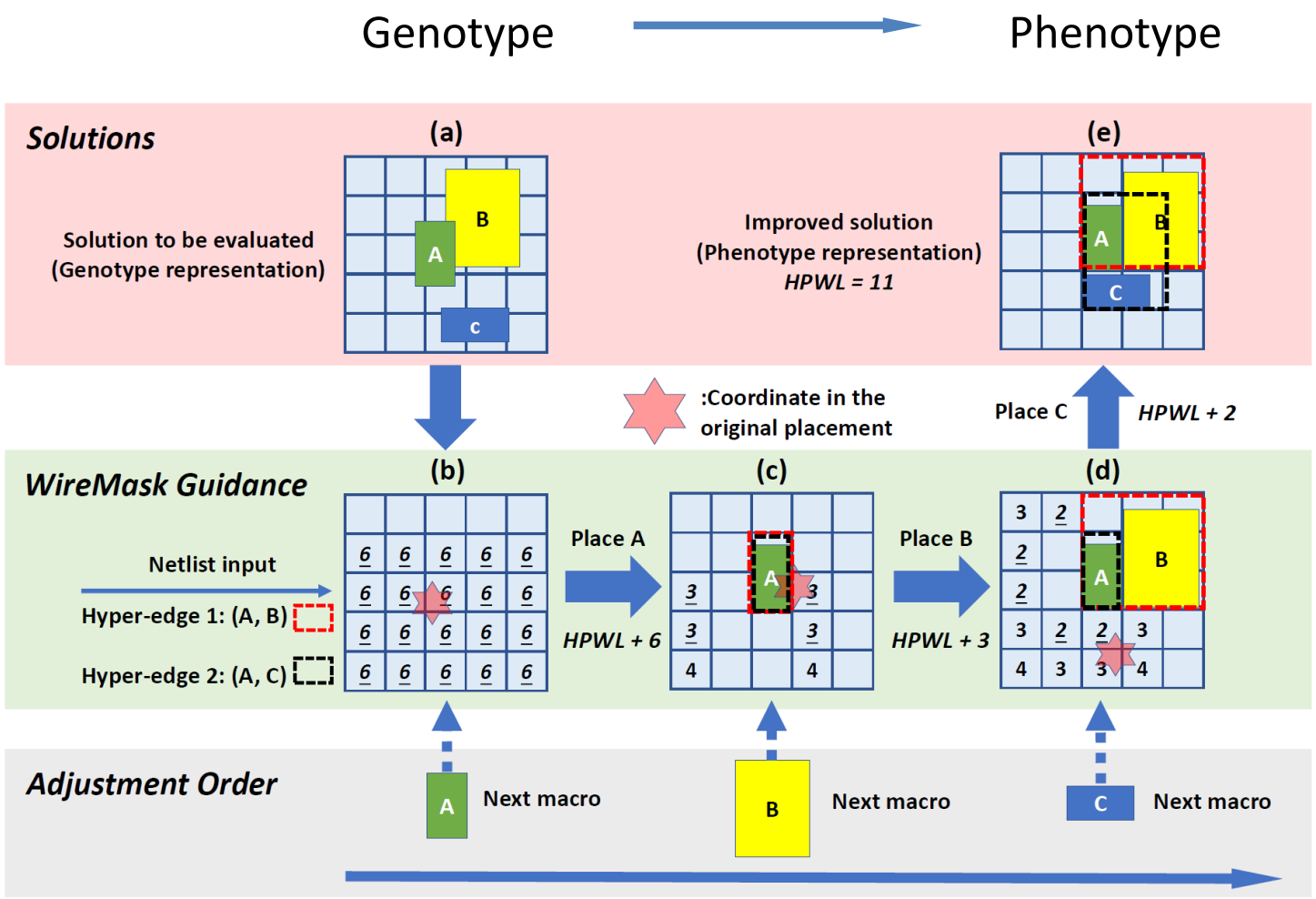
Obtain the valid phenotype placement result

A greedy mapping based on the increment of wirelength

Improve the efficiency

[Shi, Xue, Song, and Qian, NeurIPS 2023]

Optimization based on proposed mapping



Evolutionary Learning

(1+1)-EA

- **Initialization:** Randomly generate 100 genotype solutions and pick the best
- **Mutation:** Randomly exchange two macros' locations

Already performs well!

[Shi, Xue, Song, and Qian, NeurIPS 2023]

http://www.lamda.nju.edu.cn/qianc/

Comparison with previous SOTA methods

Table 1: Wirelength ($\times 10^5$) obtained by ten compared methods on seven popular ISPD contest chips.

Method	Type	adaptec1	adaptec2	adaptec3	adaptec4	bigblue1	bigblue3	bigblue4 ($\times 10^7$)	$+/-/\approx$	Avg. Rank
SP-SA [33]	Packing	18.84 ± 4.62	117.36 ± 8.73	115.48 ± 7.56	120.03 ± 4.25	5.12 ± 1.43	164.70 ± 19.55	25.49 ± 2.73	0/7/0	6.86
NTUPlace3 [12]	Analytical	26.62	321.17	328.44	462.93	22.85	455.53	48.38	0/7/0	9.00
RePlace [13]	Analytical	16.19 ± 2.10	153.26 ± 29.01	111.21 ± 11.69	37.64 ± 1.05	2.45 ± 0.06	119.84 ± 34.43	11.80 ± 0.73	1/6/0	5.28
DREAMPlace [28]	Analytical	15.81 ± 1.64	140.79 ± 26.73	121.94 ± 25.05	**37.41 ± 0.87**	2.44 ± 0.06	107.19 ± 29.91	12.29 ± 1.64	1/6/0	4.86
Graph [32]	RL	30.10 ± 2.98	351.71 ± 38.20	358.18 ± 13.95	151.42 ± 9.72	10.58 ± 1.29	357.48 ± 47.83	53.35 ± 4.06	0/7/0	9.00
DeepPR [15]	RL	19.91 ± 2.13	203.51 ± 6.27	347.16 ± 4.32	311.86 ± 56.74	23.33 ± 3.65	430.48 ± 12.18	68.30 ± 4.44	0/7/0	8.86
MaskPlace [26]	RL	6.38 ± 0.35	73.75 ± 6.35	84.44 ± 3.60	79.21 ± 0.65	2.39 ± 0.05	91.11 ± 7.83	11.07 ± 0.90	0/7/0	4.28
WireMask-RS	Ours	6.13 ± 0.05	59.28 ± 1.48	60.60 ± 0.45	62.06 ± 0.22	2.19 ± 0.01	62.58 ± 2.07	**8.20 ± 0.17**	0/5/2	2.57
WireMask-BO	Ours	6.07 ± 0.14	59.17 ± 3.94	61.00 ± 2.08	63.86 ± 1.01	2.14 ± 0.03	67.48 ± 6.49	8.62 ± 0.18	0/3/4	2.86
WireMask-EA	Ours	**5.91 ± 0.07**	**52.63 ± 2.23**	**57.75 ± 1.16**	58.79 ± 1.02	**2.12 ± 0.01**	**59.87 ± 3.40**	8.28 ± 0.25	–	**1.43**

Classical EA method

Analytical

RL

Our methods

[Google, Nature'21]

Some important baselines

- **Graph** [Nature'21]: RL method proposed by **Google**
- **DREAMPlace** [DAC'19, TCAD'21 Best Paper] : One of the most popular analytical methods
- **DeepPR** [NeurIPS'21] and **MaskPlace** [NeurIPS'22] : Two recent advanced RL methods

WireMask-EA (our proposed framework equipped with EA) achieves **the best average rank**, and **reduces wirelength by 80%** compared to [Google, Nature'21]

[Shi, Xue, Song, and Qian, NeurIPS 2023]

http://www.lamda.nju.edu.cn/qianc/

Comparison with the latest method ChiPFormer [Lai et al., ICML'23]

Table 2: Wirelength ($\times 10^5$) Compared with ChiPFormer on ten popular ISPD and ICCAD contest chips.

Benchmark	ChiPFormer (1)	WireMask-EA (1)	ChiPFormer (0.3k)	WireMask-EA (0.3k)	ChiPFormer (2k)	WireMask-EA (2k)
adaptec1	8.87 ± 0.98	**7.20 ± 0.34**	7.02 ± 0.11	**6.29 ± 0.07**	6.62 ± 0.05	**5.96 ± 0.08**
adaptec2	122.37 ± 22.61	**111.04 ± 20.09**	70.42 ± 2.67	**61.25 ± 4.10**	67.10 ± 5.46	**53.88 ± 2.53**
adaptec3	107.11 ± 8.84	**75.37 ± 2.93**	78.32 ± 2.03	**64.49 ± 1.69**	76.70 ± 1.15	**59.26 ± 1.30**
adaptec4	85.63 ± 7.52	**75.63 ± 1.30**	69.42 ± 0.54	**64.52 ± 1.81**	68.80 ± 1.59	**59.52 ± 1.71**
bigblue1	3.11 ± 0.03	**2.31 ± 0.06**	2.96 ± 0.04	**2.18 ± 0.01**	2.95 ± 0.04	**2.14 ± 0.01**
bigblue3	131.78 ± 17.36	**99.20 ± 24.69**	81.48 ± 4.83	**64.51 ± 4.15**	72.92 ± 2.56	**56.65 ± 2.81**
ibm01	4.57 ± 0.27	**3.76 ± 0.36**	3.61 ± 0.08	**2.92 ± 0.07**	3.05 ± 0.11	**2.39 ± 0.07**
ibm02	6.01 ± 0.41	**5.13 ± 0.16**	4.84 ± 0.17	**3.86 ± 0.03**	4.24 ± 0.25	**3.56 ± 0.05**
ibm03	**2.15 ± 0.17**	3.10 ± 0.12	**1.75 ± 0.07**	2.20 ± 0.11	**1.64 ± 0.06**	1.69 ± 0.11
ibm04	5.00 ± 0.14	**3.60 ± 0.17**	4.19 ± 0.11	**2.93 ± 0.11**	4.06 ± 0.13	**2.62 ± 0.04**

WireMask-EA outperforms ChiPFormer [Lai et al., ICML'23] on **9 out of 10** chips, using the same number of evaluations

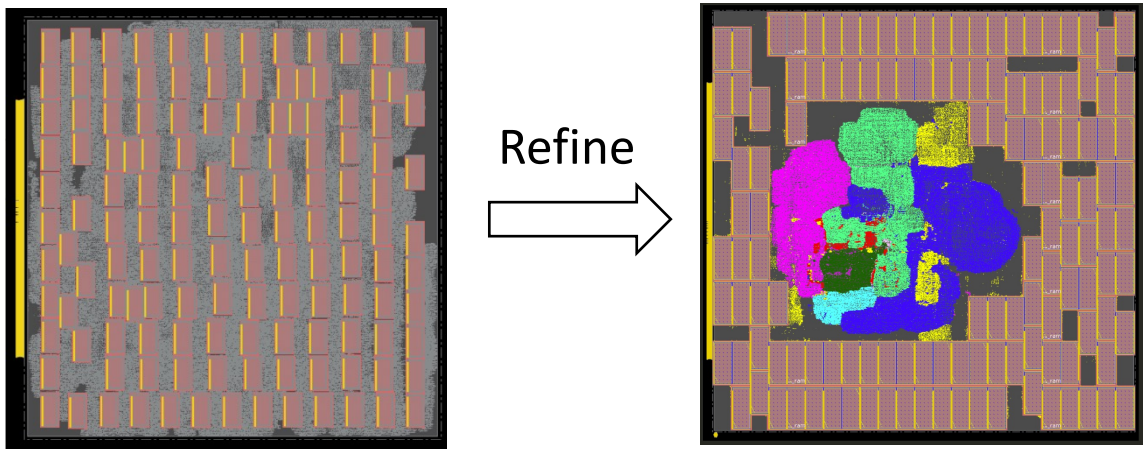
Macro Placement by Wire-Mask-Guided Black-Box Optimization

[Shi, Xue, Song, and Qian, NeurIPS 2023]

Reinforcement Learning Policy as Macro Regulator

Expert knowledge: macros should be placed to the chip periphery

- Save continuous space for standard cells
- Reduce routing congestion
- Simplify power planning



Representative works:

- **IncreMacro** [Pu et al., ISPD'24]

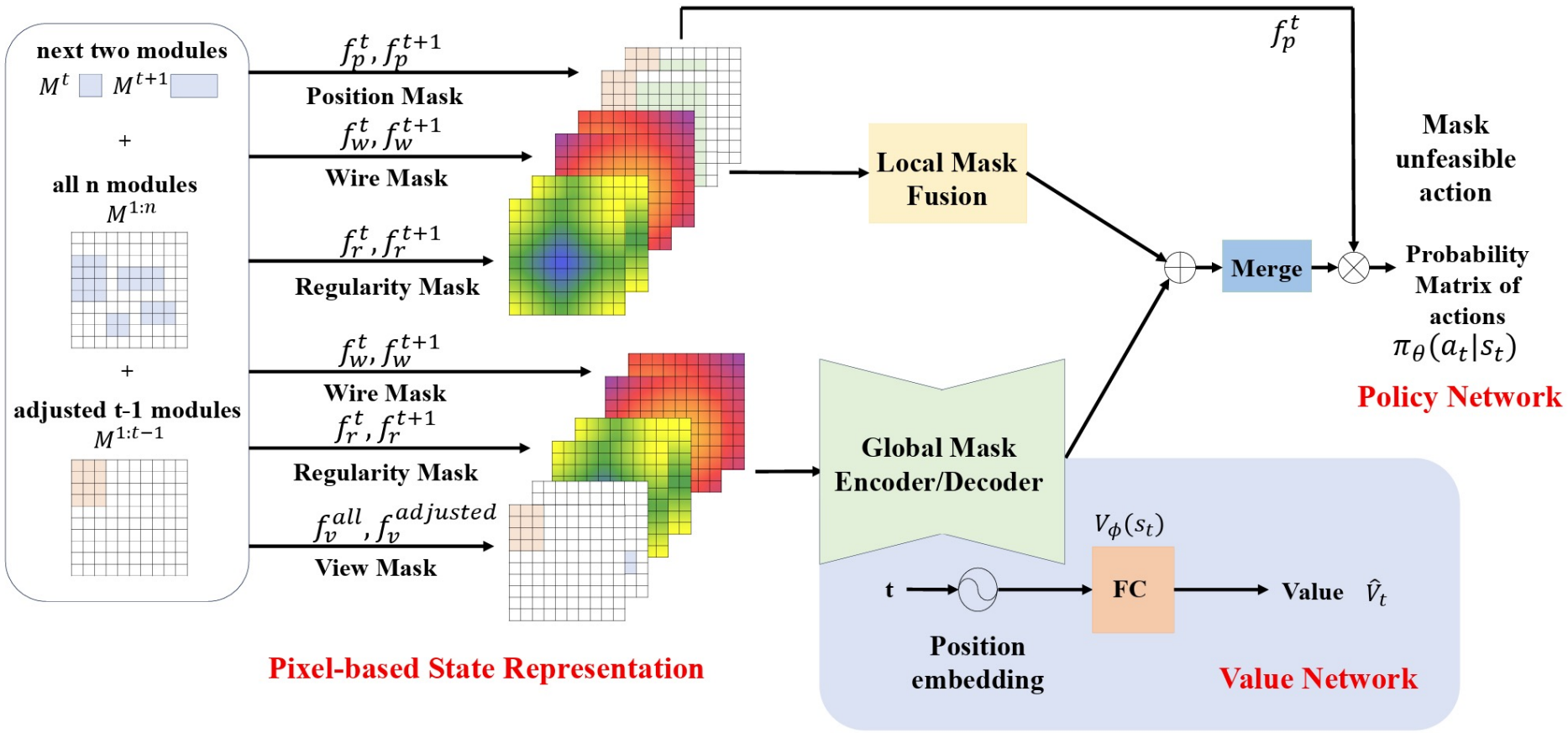
  Use analytical method to push macros to the periphery and legalize macros by linear programming

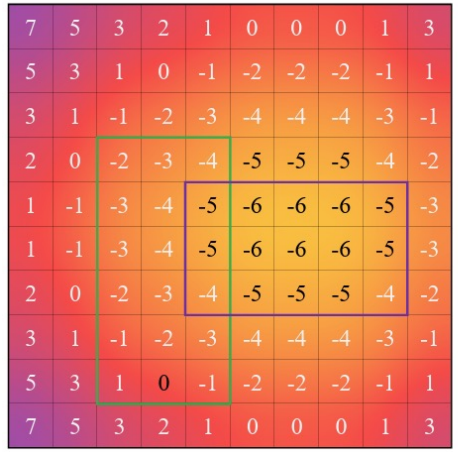
- **Macro Placement based on Simulated Evolution Algorithm (SEA)** [Lin et al., ICCAD'19]

  Use corner stitching to ensure the peripheral placement and adopt SEA to further improve it

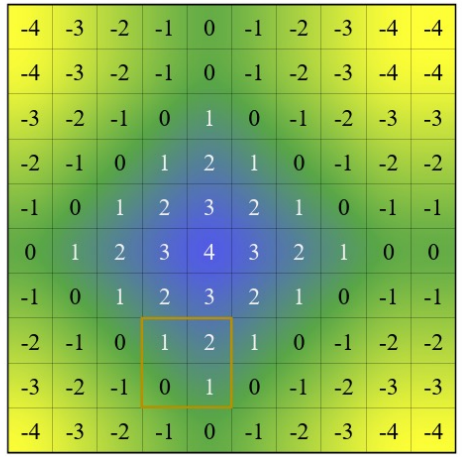
Reinforcement Learning Policy as Macro Regulator

The first RL method for macro refinement and pushing periphery





WireMask



Proposed RegularMask

[Xue, Chen, Lin, Shi, Kai, Xu, and Qian, NeurIPS 2024]

Reinforcement Learning Policy as Macro Regulator

Experiments

Rank first in all PPA metrics across 8 designs compared to recent macro placers

Benchmark	Method	Proxy metrics		PPA metrics					
		Global HPWL	Regularity	rWL	rO-H	rO-V	WNS	TNS	NVP
superblue1	DMP	8.96 ± 0.84	4.15 ± 0.04	154.23	17.15	4.48	-119.616	-2.91	3.35
	AutoDMP	8.13 ± 0.17	4.99 ± 0.08	185.60	20.99	5.73	-124.572	-3.72	3.46
	WireMask-EA	8.07 ± 0.38	4.41 ± 0.15	149.49	7.62	0.38	-67.616	-3.57	2.94
	MaskPlace	7.93 ± 0.06	4.40 ± 0.06	158.59	16.28	0.64	-72.070	-3.98	4.41
	MaskRegulate	**5.77 ± 0.05**	**3.31 ± 0.00**	**116.11**	**1.26**	**0.11**	**-60.532**	**-1.33**	**1.06**
superblue3	DMP	12.87 ± 1.73	4.43 ± 0.03	232.19	40.55	19.64	-96.904	-2.36	2.25
	AutoDMP	8.13 ± 0.69	5.49 ± 0.17	166.15	14.71	3.39	**-76.566**	**-1.12**	1.44
	WireMask-EA	9.37 ± 0.81	4.77 ± 0.23	167.67	7.81	0.32	-92.566	-1.57	2
	MaskPlace	8.90 ± 0.17	4.77 ± 0.06	177.25	9.16	0.64	-111.041	-1.77	2.02
	MaskRegulate	**7.05 ± 0.03**	**3.54 ± 0.00**	**142.89**	**1.86**	**0.18**	-83.635	-1.15	**0.97**
superblue4	DMP	6.81 ± 0.23	3.06 ± 0.01	132.16	20.62	4.87	-73.192	-1.63	2.42
	AutoDMP	4.57 ± 0.78	3.41 ± 0.06	82.94	5.43	0.21	**-48.137**	**-0.64**	1.08
	WireMask-EA	5.51 ± 0.07	3.25 ± 0.10	110.20	8.29	0.61	-83.233	-1.85	1.98
	MaskPlace	5.28 ± 0.03	3.22 ± 0.03	106.36	9.71	0.31	-67.995	-1.47	1.9
	MaskRegulate	**4.15 ± 0.06**	**2.18 ± 0.02**	**81.78**	**0.29**	**0.11**	-49.071	-0.90	**0.88**
superblue5	DMP	8.78 ± 1.47	4.84 ± 0.06	144.64	3.75	0.46	**-58.907**	**-0.68**	1.64
	AutoDMP	12.67 ± 4.09	5.79 ± 0.32	344.14	74.75	37.32	-197.175	-5.83	3.55
	WireMask-EA	10.23 ± 0.68	5.03 ± 0.15	189.84	4.06	0.41	-75.115	-1.83	2.18
	MaskPlace	9.81 ± 0.03	4.86 ± 0.04	196.79	4.79	0.37	-118.122	-2.98	2.62
	MaskRegulate	**6.94 ± 0.00**	**4.23 ± 0.01**	**137.79**	**0.02**	**0.02**	-74.83	-0.73	**1.32**

12% reduction in rWL

98% reduction in rO-H

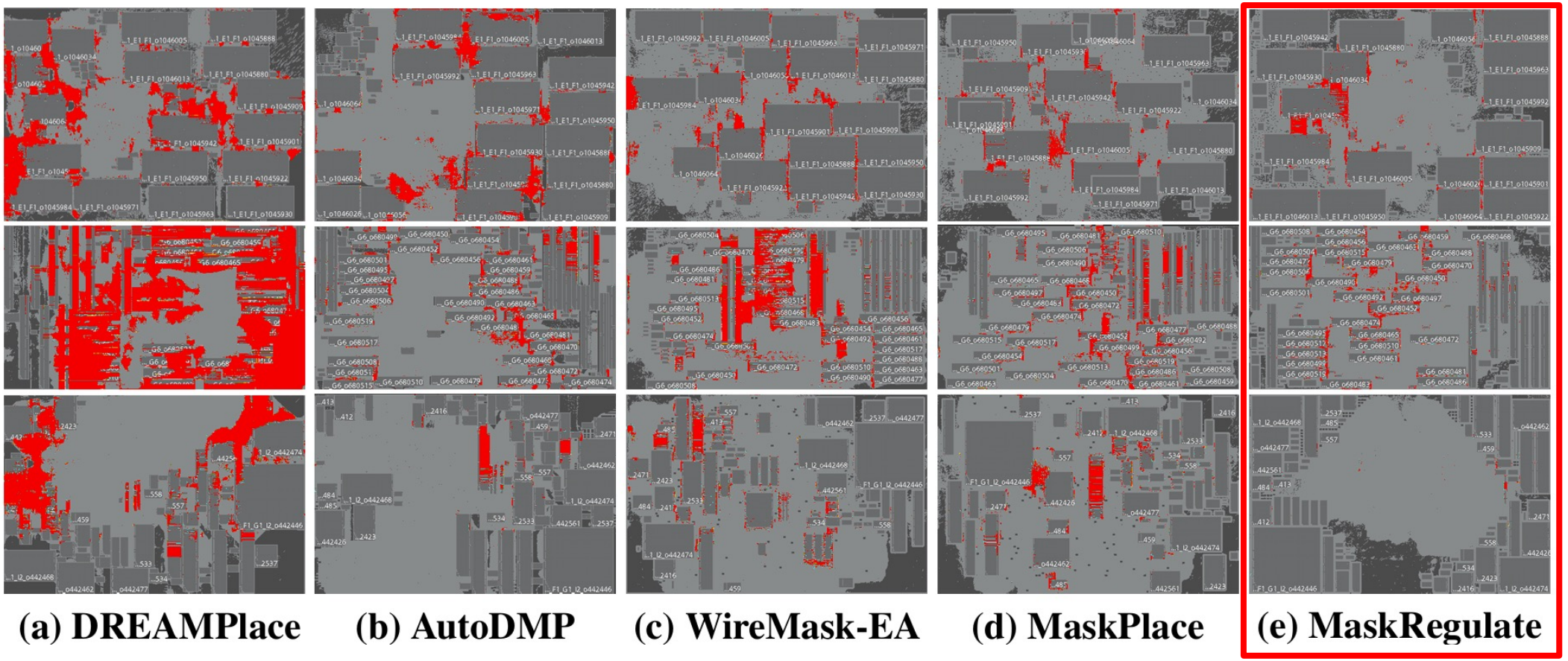
67% reduction in rO-V

[Xue, Chen, Lin, Shi, Kai, Xu, and Qian, NeurIPS 2024]

Reinforcement Learning Policy as Macro Regulator

Experiments

Reduction in routing congestion



(a) DREAMPlace (b) AutoDMP (c) WireMask-EA (d) MaskPlace (e) MaskRegulate

[Xue, Chen, Lin, Shi, Kai, Xu, and Qian, NeurIPS 2024]

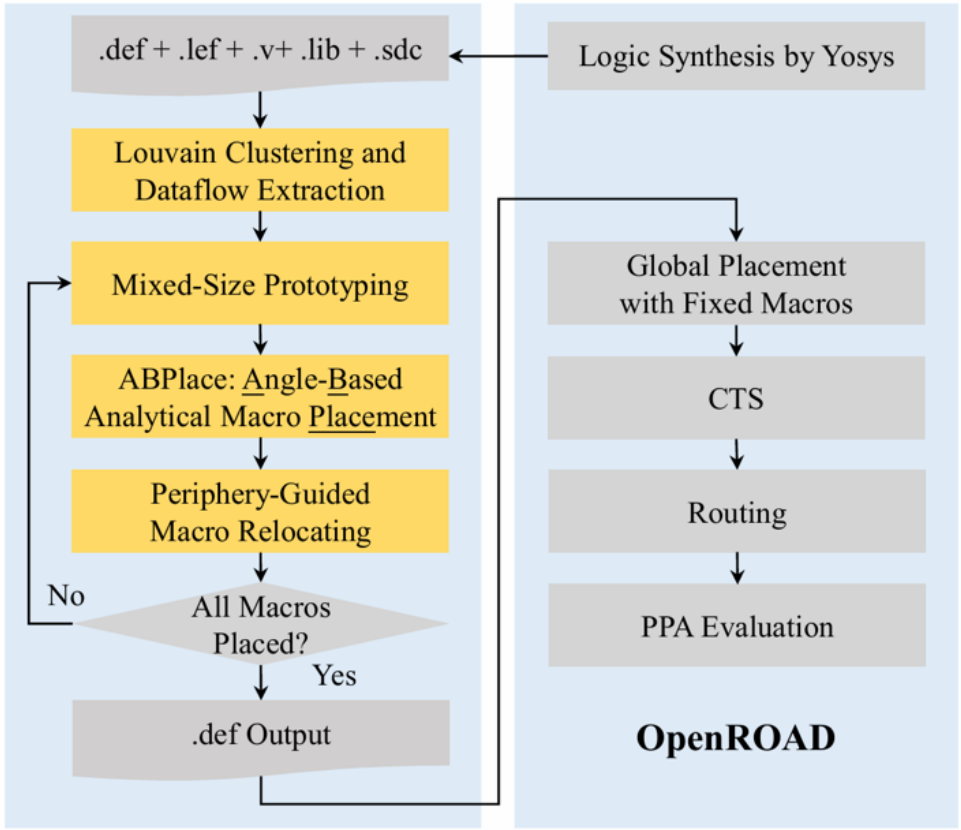
ReMaP: Macro Placement by Recursive Prototyping

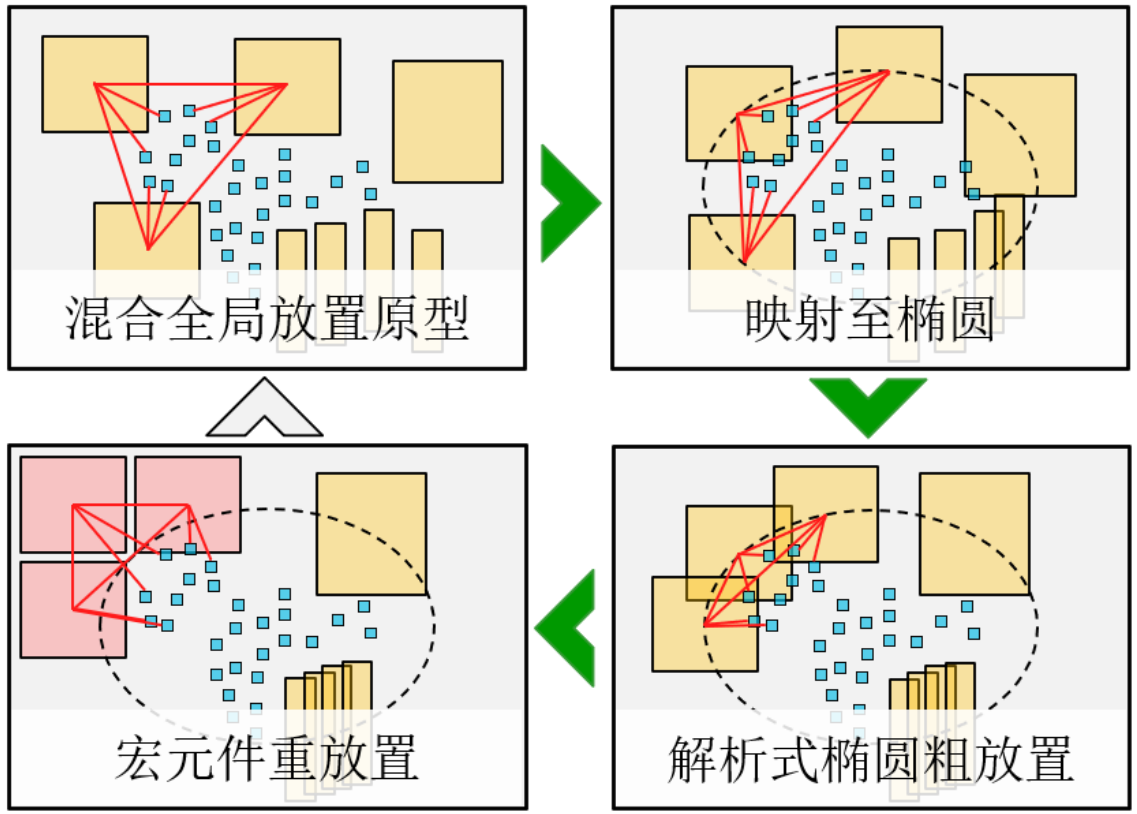
Macro refinement by reinforcement learning has certain drawbacks:

- Lacks awareness of standard cells while placing macros
- The original placement prototype may be inaccurate when several macros have been moved
- Overfitting for certain cases when training cases are limited
- Long training time

ReMaP: Macro Placement by Recursive Prototyping

Refine macros to the periphery by an ellipse formulation & recursively relocating





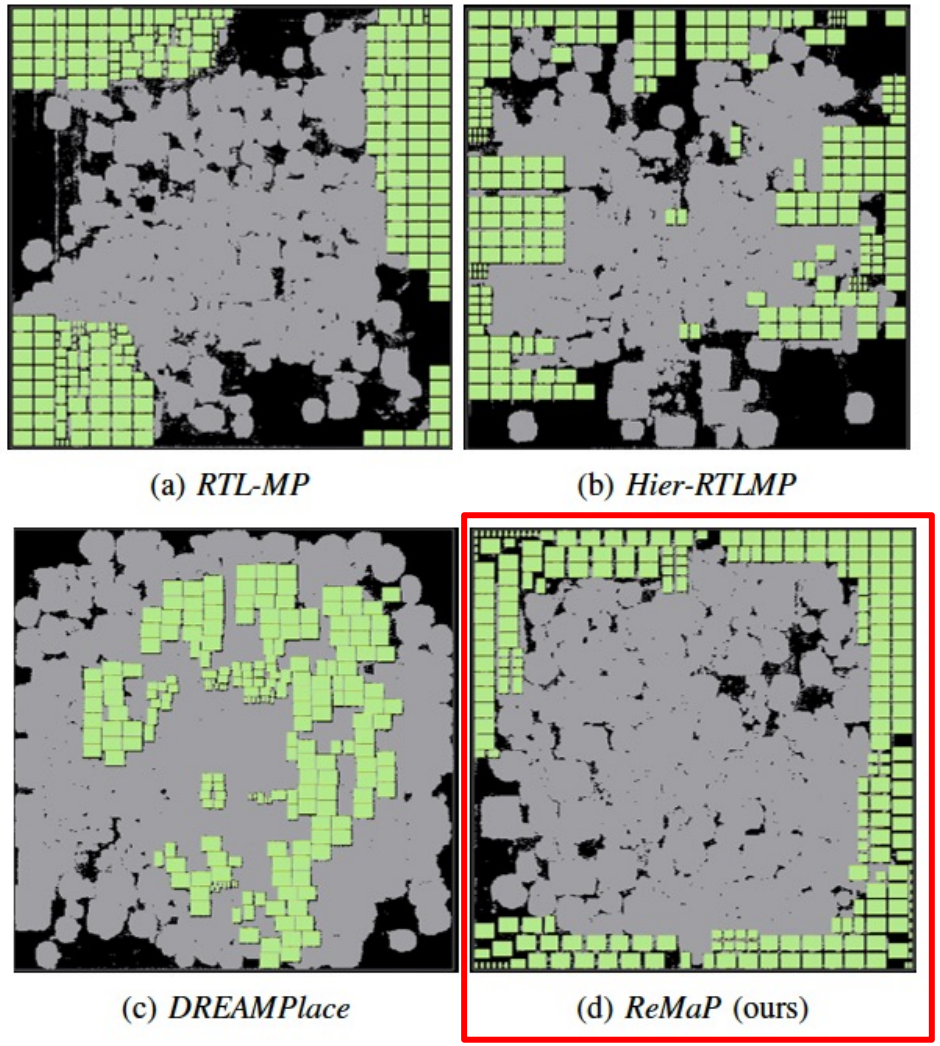
[Shi, Lin, Xu, Kai, Xue, Yuan, Qian, and Zhou, DAC 2025]

ReMaP: Macro Placement by Recursive Prototyping

LAMDA
Learning And Mining from DatA
http://www.lamda.nju.edu.cn

Experiments Best timing across all 8 cases, with an average improvement of 34% in WNS and 65% in TNS

Methods	rWL (m)	WNS (ns)	TNS (ns)	Power (W)	Overflow (#)	Runtime (s)
RTL-MP	6.76	-1.11	-2839.00	0.314	2055	1753
Hier-RTLMP	7.07	-1.17	-2967.29	0.316	0	572
DREAMPlace	7.10	-1.33	-3628.90	0.325	23	36
ReMaP (ours)	7.48	**-1.07**	**-2615.74**	0.318	24	478
RTL-MP	8.16	-2.07	-6315.44	0.479	332	4859
Hier-RTLMP	7.91	-2.23	-6830.66	0.479	100	153
DREAMPlace	7.88	-2.21	-6700.25	0.483	78	47
ReMaP (ours)	7.81	**-2.04**	**-6085.84**	0.464	7	589
RTL-MP	9.72	-4.98	-1102.72	0.475	6281	18
Hier-RTLMP	10.84	-4.75	-2111.50	0.463	75323	124
DREAMPlace	10.52	-5.15	-1216.80	0.457	6922	45
ReMaP (ours)	9.34	**-4.67**	**-1090.18**	0.454	4548	322
RTL-MP	3.36	-0.98	-125.01	0.134	25978	2
Hier-RTLMP	3.35	-0.96	-105.89	0.135	80552	17
DREAMPlace	3.80	-0.92	-111.73	0.139	23007	24
ReMaP (ours)	3.89	**-0.90**	**-98.00**	0.136	31288	389
RTL-MP	2.16	-0.41	-19.88	0.156	8841	2
Hier-RTLMP	2.17	-0.45	-23.68	0.155	34839	17
DREAMPlace	2.58	-0.65	-43.28	0.162	28885	18
ReMaP (ours)	2.65	**-0.27**	**-6.88**	0.160	15936	401



(a) *RTL-MP* (b) *Hier-RTLMP* (c) *DREAMPlace* (d) *ReMaP* (ours)

[Shi, Lin, Xu, Kai, Xue, Yuan, Qian, and Zhou, DAC 2025]

ReMaP: Macro Placement by Recursive Prototyping

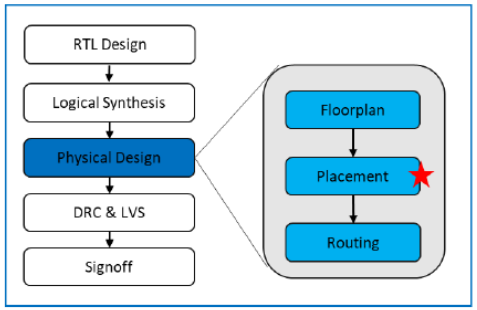
LAMDA
Learning And Mining from DatA
http://www.lamda.nju.edu.cn

难题4：布局布线优化技术

出题组织：海思/诺亚方舟实验室　　接口专家：许思源 xusiyuan520@huawei

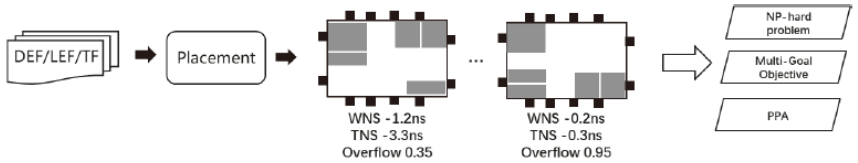
技术背景

芯片布局起着承上（逻辑综合）启下（布线）的作用，是现代超大规模集成电路设计物理设计流程中的一步，显著影响芯片设计的迭代周期。由于芯片布局被认为是NP-Complete的问题，因此如何快速生成高质量的芯片布局是布局问题的重要挑战。

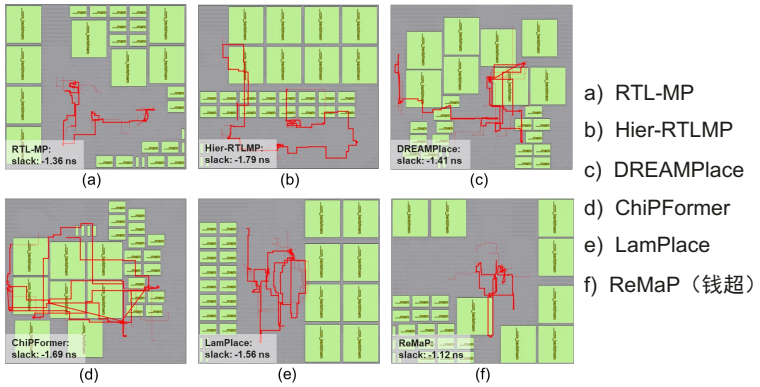


Chip Design Flow　　Auto Placement and Route (APR) Flow

芯片布局问题可以描述为给定标准单元和宏单元的大小和连接关系，给出约束(如没有元件重叠)和优化目标(时序，拥塞，功耗，面积等)，基于特定策略确定每个单元的位置。



在 6 个样例上对比SOTA方案平均提升 PPA 性能约 18.9%，平均提升拥塞指标约 46.3%，预测相关性（99%）



方案对比布局时序图

火花奖第115期

火花奖

难题期数	难题分类	难题	院校/部门	姓名
EDA专题第一期	EDA专题	布局布线优化技术	南京大学人工智能学院	钱超教授
算力会战第七期	算力会战	超长序列视频生成推理加速关键技术	浙江大学计算机科学与技术学院	李玺教授
算力会战第七期	算力会战	HPC应用骨架函数自动提取方法	浙江大学软件学院	常志豪研究员

Efficient Timing Driven Global Placement

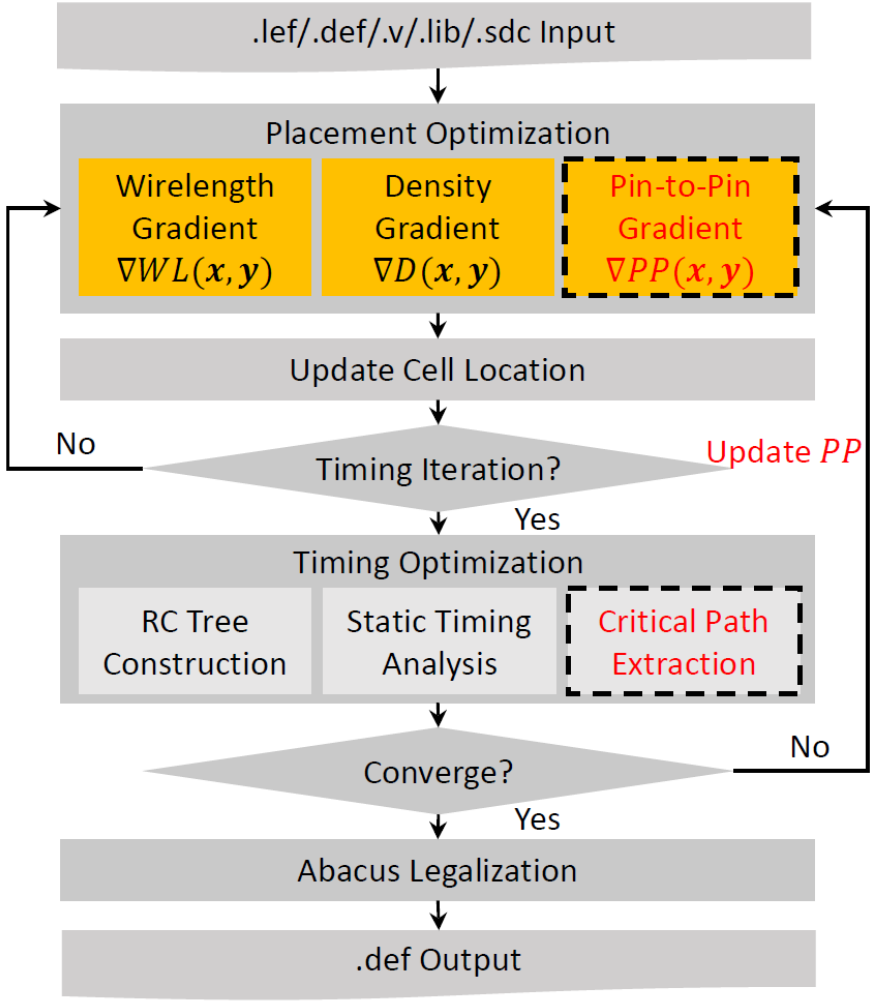
None of the above works directly target timing optimization, which determines the performance of the chip. Thus, we turn to **timing optimization in global placement.**

Representative works:

- Net-based methods [Liao et al., TCAD'24] are easy to implement, but they are coarse since optimization is performed at the net level.
- Path-based methods [Guo et al., DAC'22] suffer from scalability issues due to the poor scalability of GPU-accelerated STA.

Our contribution:

- Propose **pin-to-pin attraction** for fine-grained weighting
- Propose **efficient endpoint-based** critical path extraction
- Propose **quadratic loss** matching the timing model



[Shi, Xu, Kai, Lin, Xue, Yuan, and Qian, DATE 2025 Best Paper Award]

Efficient Timing Driven Global Placement

LAMDA
Learning And Mining from DatA
http://www.lamda.nju.edu.cn

Experiments

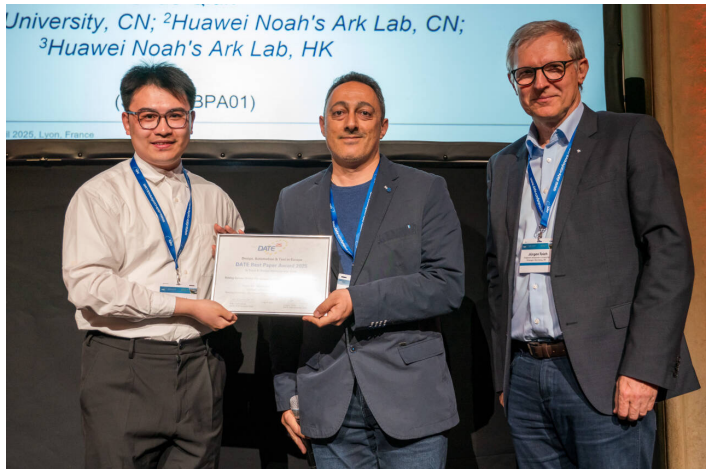
Benchmark	DREAMPlace* [20]		DREAMPlace 4.0* [18]		Differentiable-TDP† [12]		Distribution-TDP§ [19]		Ours	
	TNS	WNS	TNS	WNS	TNS	WNS	TNS	WNS	TNS	WNS
superblue1	-262.44	-18.87	-85.03	-14.10	-74.85	-10.77	-42.10	-9.26	**-17.44**	**-7.75**
superblue3	-76.64	-27.65	-54.74	-16.43	-39.43	-12.37	-26.59	-12.19	**-20.40**	**-11.82**
superblue4	-290.88	-22.04	-144.38	-12.78	-82.92	**-8.49**	-123.28	-8.86	**-82.88**	-9.17
superblue5	-157.82	-48.92	-95.78	-26.76	-108.08	-25.21	-70.35	-31.64	**-62.18**	**-24.65**
superblue7	-141.55	-19.75	-63.86	**-15.22**	-46.43	**-15.22**	-95.89	-17.24	**-43.52**	**-15.22**
superblue10	-731.94	-26.10	-768.75	-31.88	**-558.05**	**-21.97**	-691.10	-25.86	-558.14	-23.08
superblue16	-453.57	-17.71	-124.18	-12.11	-87.03	-10.85	-55.99	-12.21	**-22.90**	**-8.63**
superblue18	-96.76	-20.29	-47.25	-11.87	-19.31	-7.99	-19.23	**-5.25**	**-16.16**	-6.92
Average Ratio	6.90	2.07	2.75	1.40	2.00	1.09	1.68	1.11	**1.00**	**1.00**

Best timing on ICCAD15 contest benchmark, compared to SOTA methods, with an average improvement of 40.5% in TNS and 8.3% in WNS.

[Shi, Xu, Kai, Lin, Xue, Yuan, and Qian, DATE 2025 Best Paper Award]

http://www.lamda.nju.edu.cn/qianc/

Efficient Timing Driven Global Placement



获 EDA 领域顶级国际会议 DATE'25

最佳论文奖

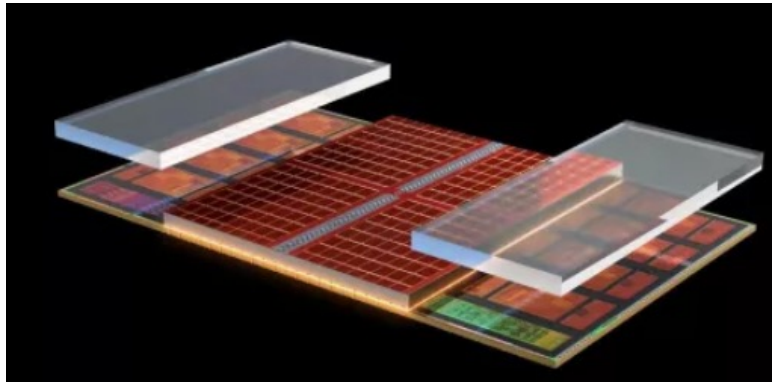
[Shi, Xu, Kai, Lin, Xue, Yuan, and Qian, DATE 2025 Best Paper Award]

Open3DBench

From 2D to 3D: Powering advanced integration

- 3D integration promises higher density as conventional fabrication nears its physical limits.
- 3D placement methodologies have attracted great attention these years, highlighted by the ICCAD'22 and ICCAD'23 3D placement contests.
  - Yet none of these contests offers a complete PDK for rigorous PPA evaluation.
  - Open-source platforms for 3D IC research are still lacking.

正与华为合作探索更为前沿的3D堆叠布局



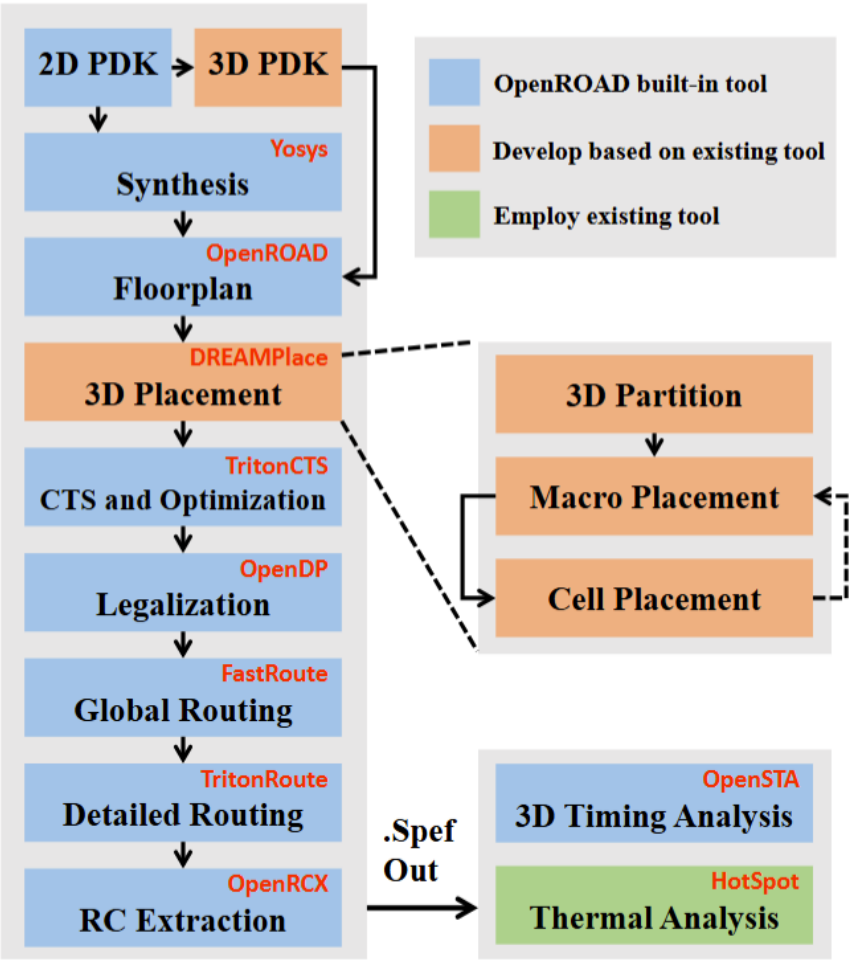
AMD 3D V-cache, ISSCC'22

[Shi, Gao, Ren, Xu, Xue, Yuan, Qian, and Zhou, arXiv:2503.12946]

Open3DBench

Open-Source Benchmark for 3D-IC PPA evaluation

- We propose an open-source 3D flow built on OpenROAD that features:
  - It converts any design supported by OpenROAD-flow-scripts into a 3D version.
  - It supports memory-on-logic 3D placement powered by the 2D DREAMPlace engine.
  - It supports 3D tier partition, placement, routing, bonding terminal assignment, timing, and thermal simulation.
  - It offers a convenient framework for evaluating any existing 3D IC methodology.



[Shi, Gao, Ren, Xu, Xue, Yuan, Qian, and Zhou, arXiv:2503.12946]

Open3DBench

Experiments

We test our framework on eight designs from OpenROAD

Designs	Methods	Area (mm^2)	rWL (m)	Overflow (#)	WNS (ns)	TNS (ns)	Power (W)	T_{max} (°C)	Runtime (s)
bp_quad	*Hier-RTLMP-2D*	12.96	46.63	3429	-3.66	-39020.00	**1.822**	66.05	8010
	DREAMPlace-2D	12.96	41.99	3968	-2.05	-31231.90	1.848	68.17	6336
	Open3D-Tiling	6.25	50.19	0	-2.62	-31124.70	1.840	69.78	7973
	Open3D-DMP	6.25	**40.39**	0	**-1.83**	**-26966.20**	1.832	66.96	7981
swerv_wrapper	*Hier-RTLMP-2D*	1.10	5.62	14428	-2.14	-1975.79	0.250	54.86	1175
	DREAMPlace-2D	1.10	5.54	9540	-1.86	-1429.90	0.254	53.48	1092
	Open3D-Tiling	0.56	3.63	0	-1.26	-972.80	**0.232**	62.17	2085
	Open3D-DMP	0.56	**3.46**	0	**-1.23**	**-958.01**	0.234	60.49	1744
3D improvements over 2D†		51.19%↑	24.06%↑	100%↑	16.24%↑	30.84%↑	5.72%↑	-10.04%↓	-24.82%↓

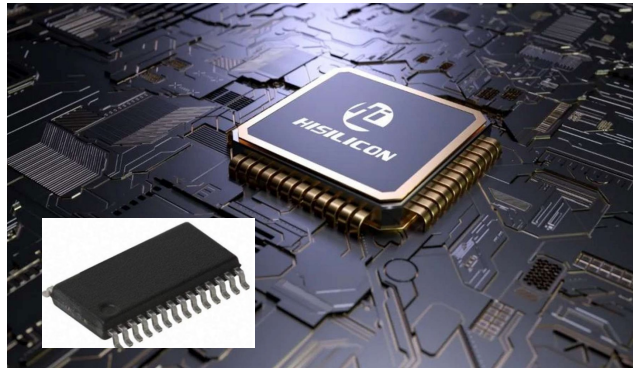
Compared to OpenROAD default 2D flow, our 3D methods show significant improvements of 50% in area, 100% in routing overflow, 30% in overall timing and 5% in power

However, 3D packing also brings about thermal issues

应用案例：芯片寄存器寻优

Chip register optimization is to maximize the performance of the application system by optimizing

- hardware parameters (various function control of registers)
- software parameters (resource scheduling of operating system)



Challenges

- Black-box: the performance is evaluated by running the system
- Expensive: one evaluation costs 20s-140s
- High-dimensional: 274 parameters

应用案例：芯片寄存器寻优

对比算法为华为自研贝叶斯优化方法HeBO，曾获**NeurIPS 2020黑箱优化比赛**冠军

工业实际数据集

达到HeBO收敛目标值

任务	HeBO 收敛值	HeBO 运行轮数	提出方法 运行轮数	效率提升
mysql	393522	293	11	26.64倍
nginx	21768	198	4	49.50倍
redis	560446	175	49	3.57倍
unixbench	107	256	29	8.83倍

华为最高价值“火花奖”

在芯片寄存器寻优的工业实际数据集上，寻优效率平均提升22.14倍

谢谢大家！