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

Evolutionary Learning

From Theory to Practice

Chao Qian

School of Artificial Intelligence
Nanjing University, China

Black-box Optimization

Solution　　Objective functions

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

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

$$h_i(\boldsymbol{s}) \le 0, \quad q+1 \le i \le m$$ Inequality constraints

Black-box Optimization: $\boldsymbol{s} \longrightarrow \blacksquare \longrightarrow f(\boldsymbol{s})$

Multi-objective

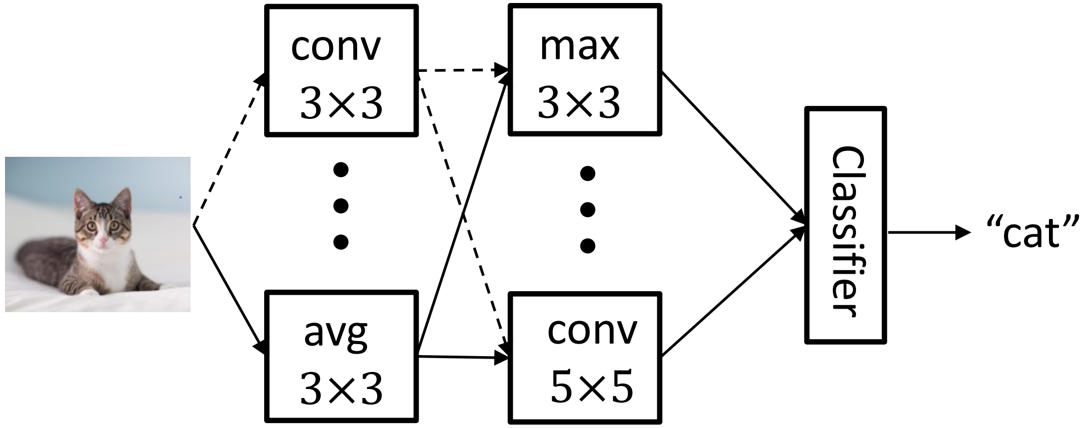
High-dimension

Goal: find good solutions using only a few $\boxed{\text{objective evaluations}}$

Usually expensive

Example: Hyper-parameter Optimization in Machine Learning

Neural architecture search



- Hyper-parameters to be optimized:
  module type, module connections, module hyper-parameters, …

  Thousands of hyper-parameters

- Objective: maximize accuracy
  Objective evaluation requires neural network training and testing, which may cost at least several hours

  Black-box Expensive

- Objective: minimize computational cost

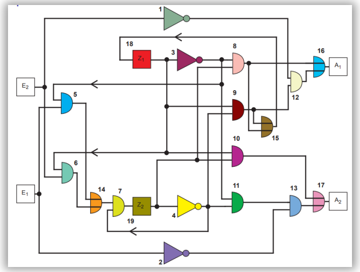
  Non-unique

应用场景：芯片布局

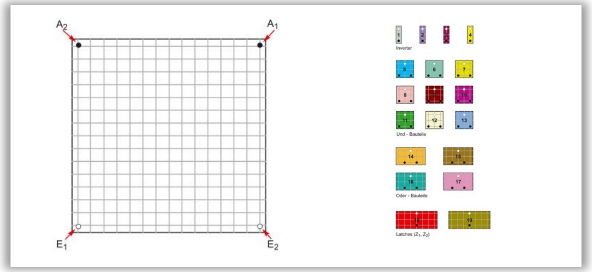
以工业制造中的

“卡脖子”问题

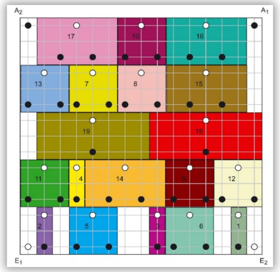
芯片布局为例



网表

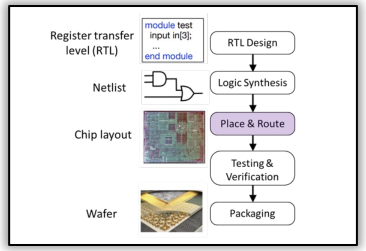


芯片版面　待放置元件



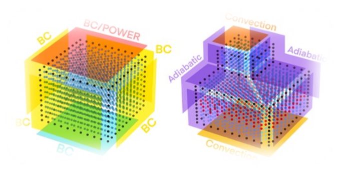
布局结果

性能评估冗长复杂



目标黑箱

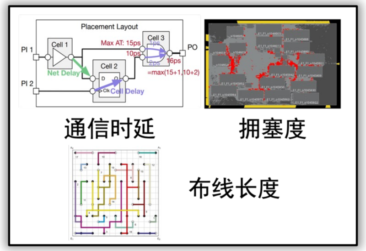
模拟仿真非常耗时



芯片热仿真

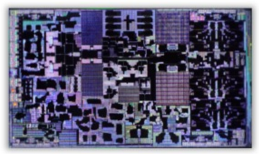
评估昂贵

涉及多个性能指标



目标多样

芯片元件数目庞大



苹果 A17pro
190 亿个晶体管

规模庞大

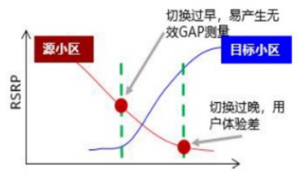
Example：华为揭榜挂帅难题

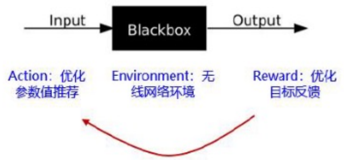
华为“揭榜挂帅”难题存在大量复杂黑箱优化

难题3：无线协同类参数优化 —超大规模黑箱多目标寻优

出题组织：无线MAE产品部　接口专家：徐志节 xuzhijie@huawei.com

- 切换类和负载均衡类(MLB)参数优化
- 基于黑箱的多参数优化问题





难题4：[AI] MIP 求解器的自学习技术

出题组织：EI服务产品部　接口专家：毛坤 maokun@huawei.com

当前结果

- **参数降维**：基于求解器开发者的专家经验，并结合XGBoost拟合随机观测的参数-性能对的代理模型所得到的Gini完成2000+参数的预筛选，实现关键参数辨识目的；
- **性能纠偏**：对已观测样本应用Box-Cox Transform实现自适应分布变换，转换成正态分布，以解决偏态分布中潜问题；
- **黑箱优化** 在采集函数优化环节，打破常规BO实现中单一采集函数的限制，集成UCB、PI、EI等多种采集函数进化（至少3目标），并从Pareto前沿中采样推荐参数，实现期望与方差的更优平衡；

难题5：硬件设计中的多黑盒参数的多目标寻优算法

出题组织：加速器技术部　接口专家：崔昕 cuixin9@huawei.com，李磊 lilei291@

技术背景

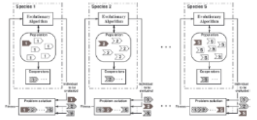
- 硬件设计中，需要找到合适的硬件设计参数（如尺寸、材料等）以得到最优的硬件产品性能。当前将硬件设计的设计参数，送入仿真工具，得到仿真后的结果。
- 设计参数：总共10~300个设计参数，包括几何相关参数，材料参数等
- 仿真结果：仿真软件会输出结果，即硬件性能指标等
- 优化目标：200~3000个仿真结果的一个加权函数值。该函数的值越小越好，最优值趋于0

难题5：[优化决策]大规模复杂网络中多参数耦合、多目标竞争下快速寻优

出题组织：公共开发部　接口专家：赖卓航 laizhuohang@huawei.com

技术挑战

方向1：基于运筹优化与AI技术的大规模黑盒参数寻优算法



协同进化策略示意图

技术挑战：
- **模型分解**：由于业务模型庞大导致优化困难，需设计模型分解策略与子模优化策略，最终达到原模型最优
- **算法自适应选择**：无线网络优化业务种类繁多，使得不同的业务模型具有数学特征和复杂度。需分析模型的特征以选择最优的优化算法，提升寻优
- **智能优化算法**：需加入基于AI的智能搜索策略，提升每个寻优算法的性能

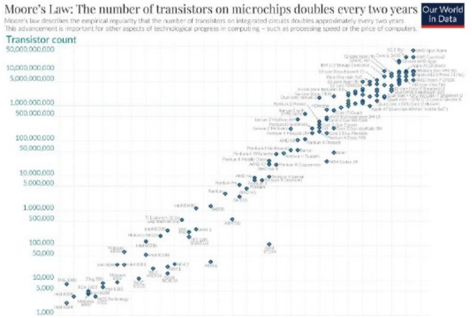
技术诉求

大规模黑盒参数优化问题高效求解：设计面向无线网络优化场景的高性能大规模黑盒参数优化方案法端到端优化时长≤ 1 min(@准实时场景)，≤ 15 min(@非实时场景)，目标数3-10个，对象数1W 100-400个寻优参数/对象

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@huaw

技术背景



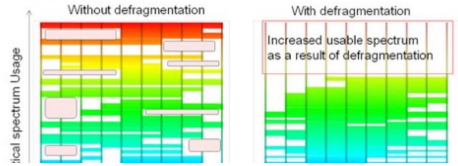
难题3：大规模混速率FlexGrid光网络多目标最优化算法

出题组织：攻坚一纵　接口专家：杨延超 yangyanchao8@huawei.com

技术挑战

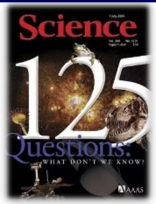
大规模混速率FlexGrid频谱多目标最优算法：千级节点万级业务场景下在2h内，获得业务成功率，数量，频谱利用率，时延等综合最优的建网解决方案。

- **NP-Hard问题**：ML-RSA问题是NP-难问题，求解难度随网络规模呈几何倍增。
- **频谱碎片问题**：由于业务是不同类型、速率、调制格式和频宽的组合，且分配频谱资源时需要遵守的一致性、连续性和不重叠约束，使得频谱资源变得离散，产生带宽大小不一的碎片，频谱利用率低下（如下图）。



应用场景：生命起源与演化

深时数字地球国际大科学计划核心科学内容

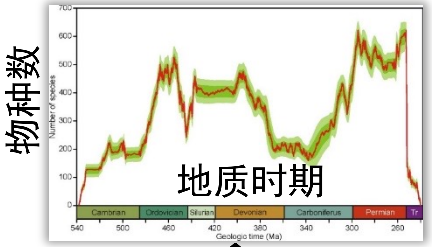


自然科学基本问题 — 生命起源与演化

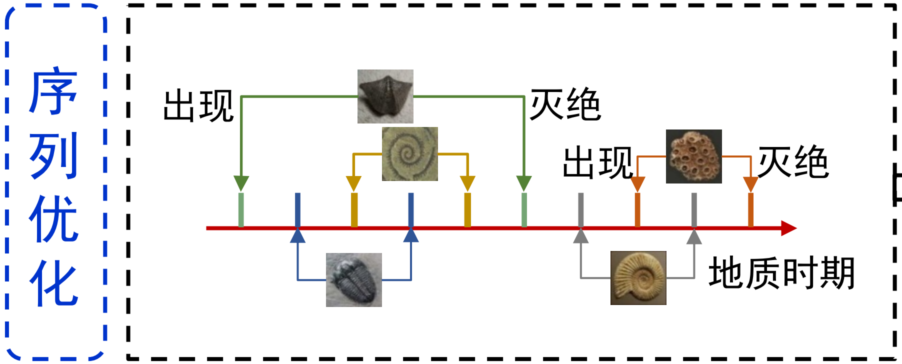
地层剖面中
海量化石记录

三叶虫、笔石、珊瑚、腕足…

生物多样性曲线



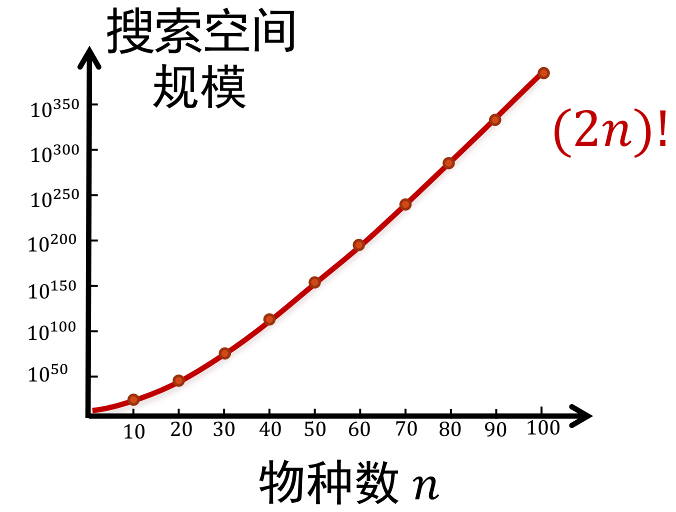
重现生命演化历史



对不同物种的“出现”和“灭绝”事件排序，
使其与地层剖面中采集到的化石记录尽可能一致

计算复杂
目标黑箱

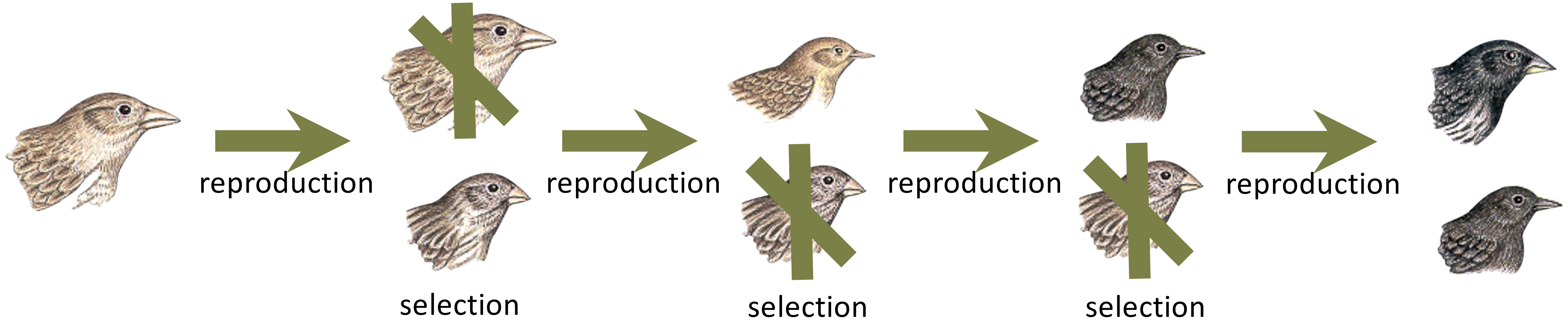
搜索空间规模
关于物种数呈指数级增长



大规模：11268个物种

Evolutionary Algorithms

Evolutionary algorithms (EAs) are a kind of randomized heuristic optimization algorithms, inspired by nature evolution (*reproduction with variation* + *nature selection*)

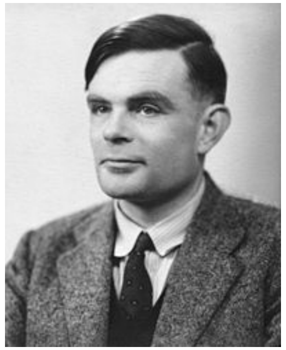


In 1950, Turing described how evolution might be used for his optimization:

building intelligent machine

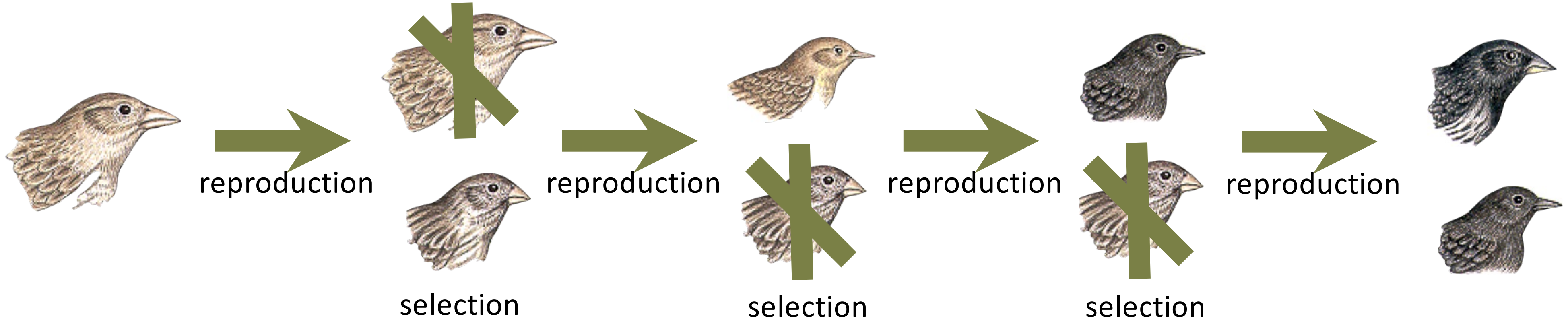
"Structure of the child machine = Hereditary material
Changes of the child machine = Mutations
Judgment of the experimenter = Natural selection"

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



Evolutionary Algorithms

Evolutionary algorithms (EAs) are a kind of randomized heuristic optimization algorithms, inspired by nature evolution (*reproduction with variation* + *nature selection*)



Many variants: genetic algorithm, evolutionary strategy, genetic programming, …

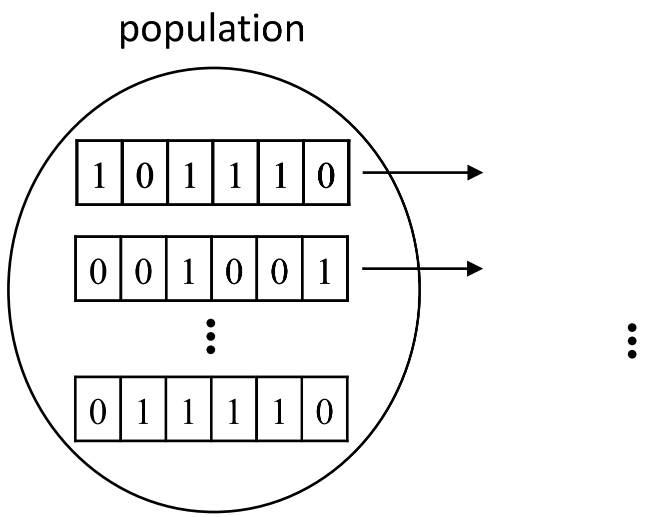
EAs also include some heuristics inspired from nature phenomena

particle swarm optimization

ant colony optimization

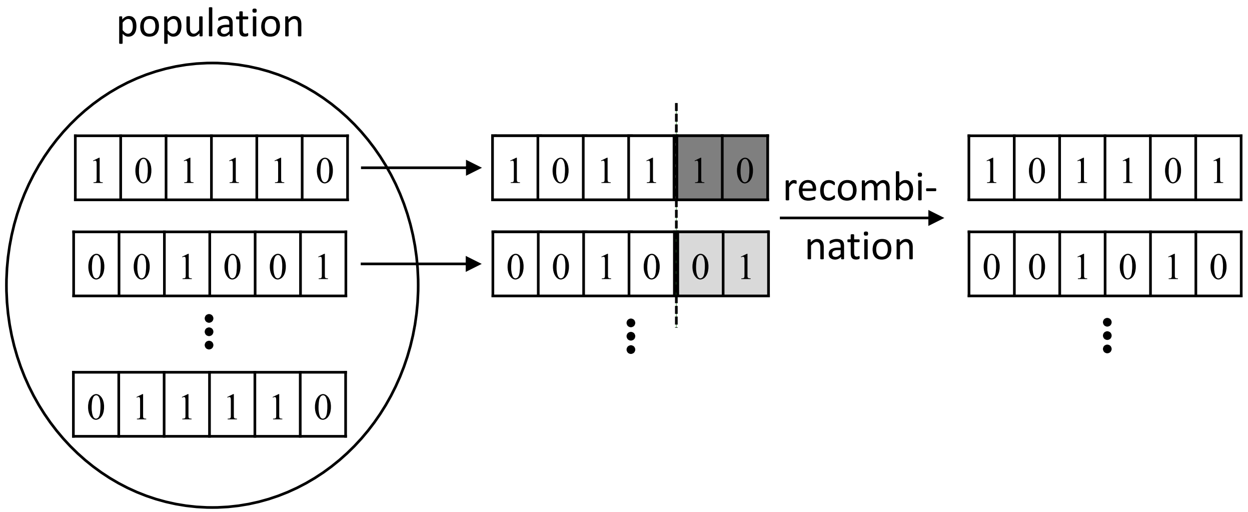
A Typical Evolutionary Process

$\arg\max_{\boldsymbol{s}} f(\boldsymbol{s})$



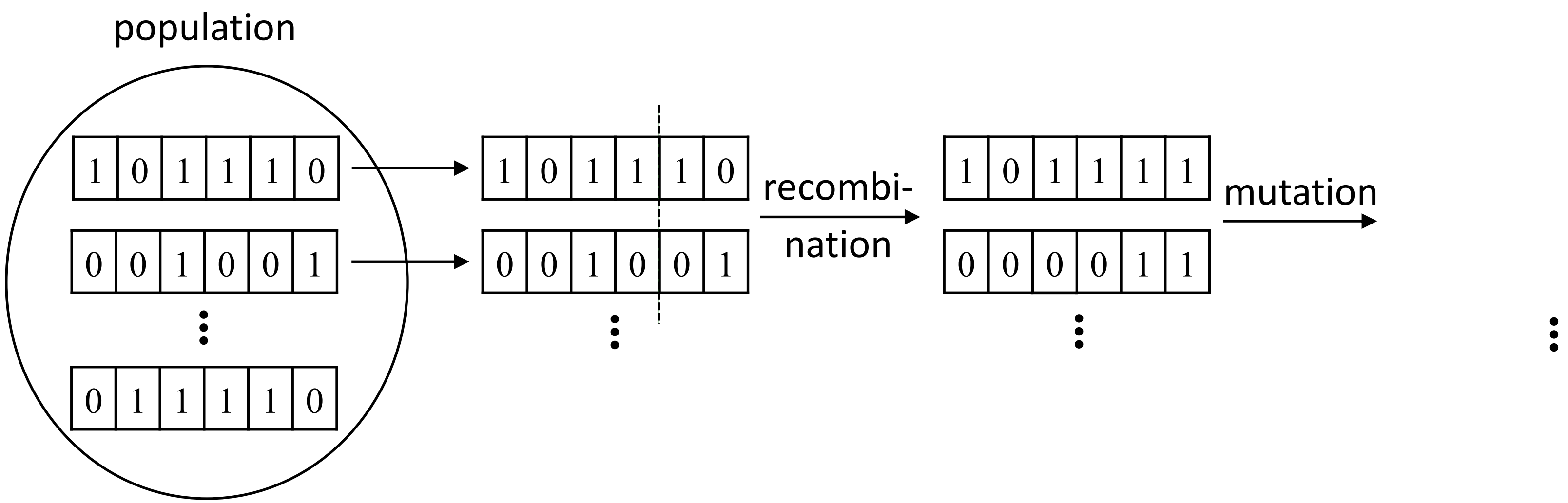
A Typical Evolutionary Process

$\arg\max_{\boldsymbol{s}} f(\boldsymbol{s})$



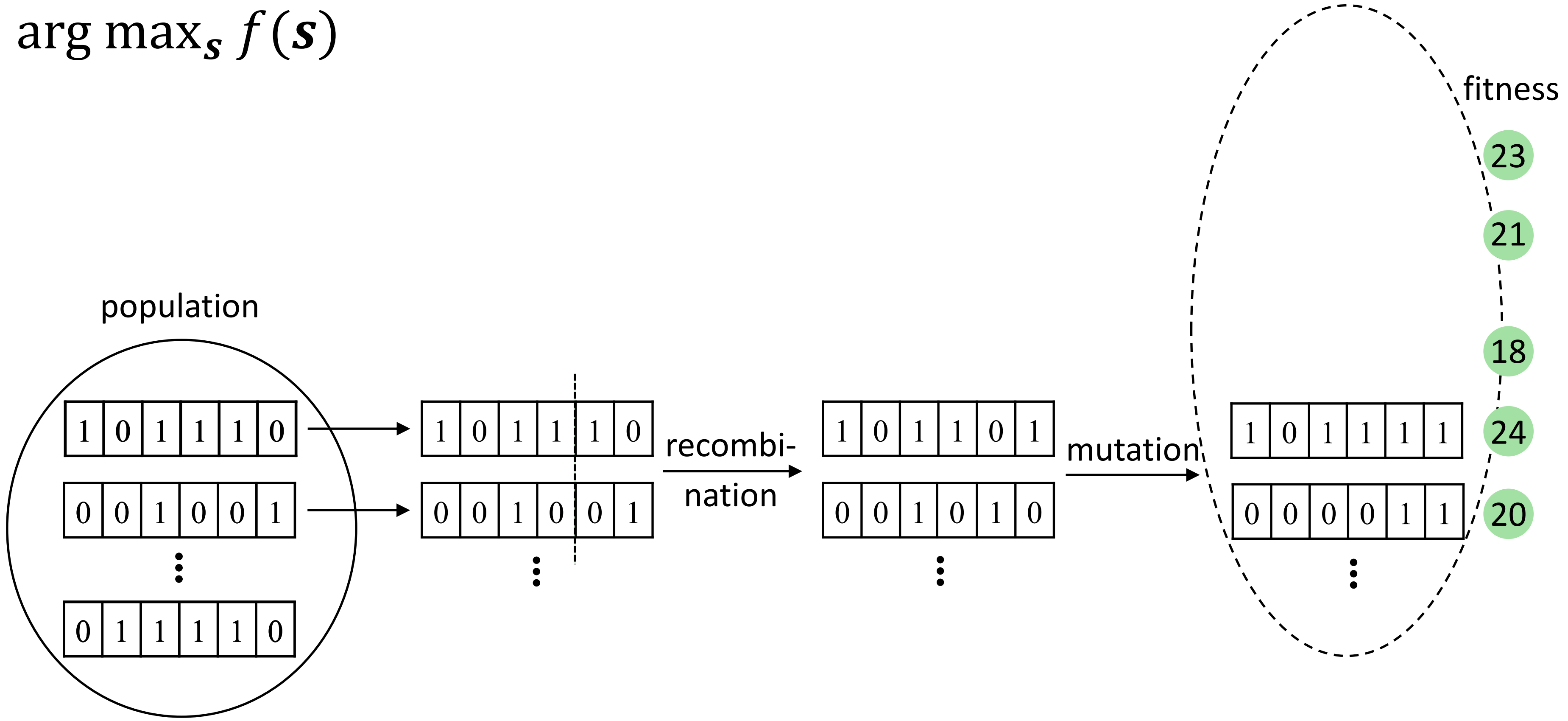
A Typical Evolutionary Process

$\arg\max_{\boldsymbol{s}} f(\boldsymbol{s})$



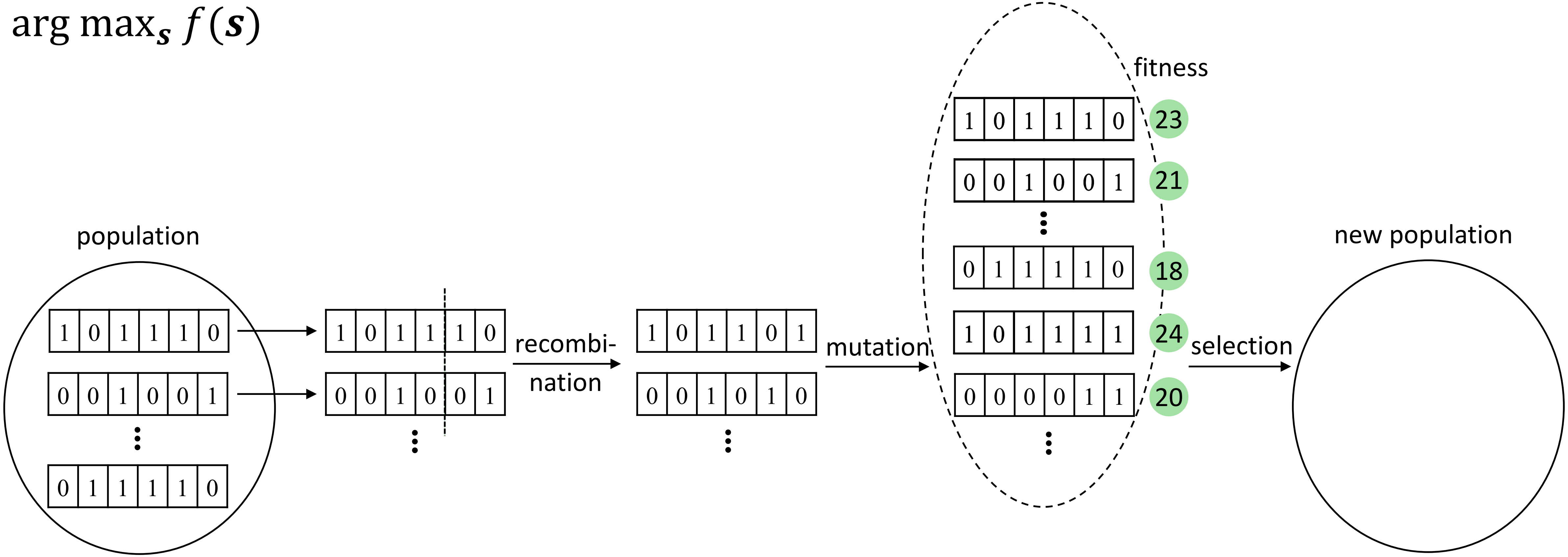
A Typical Evolutionary Process

$\arg\max_{\boldsymbol{s}} f(\boldsymbol{s})$



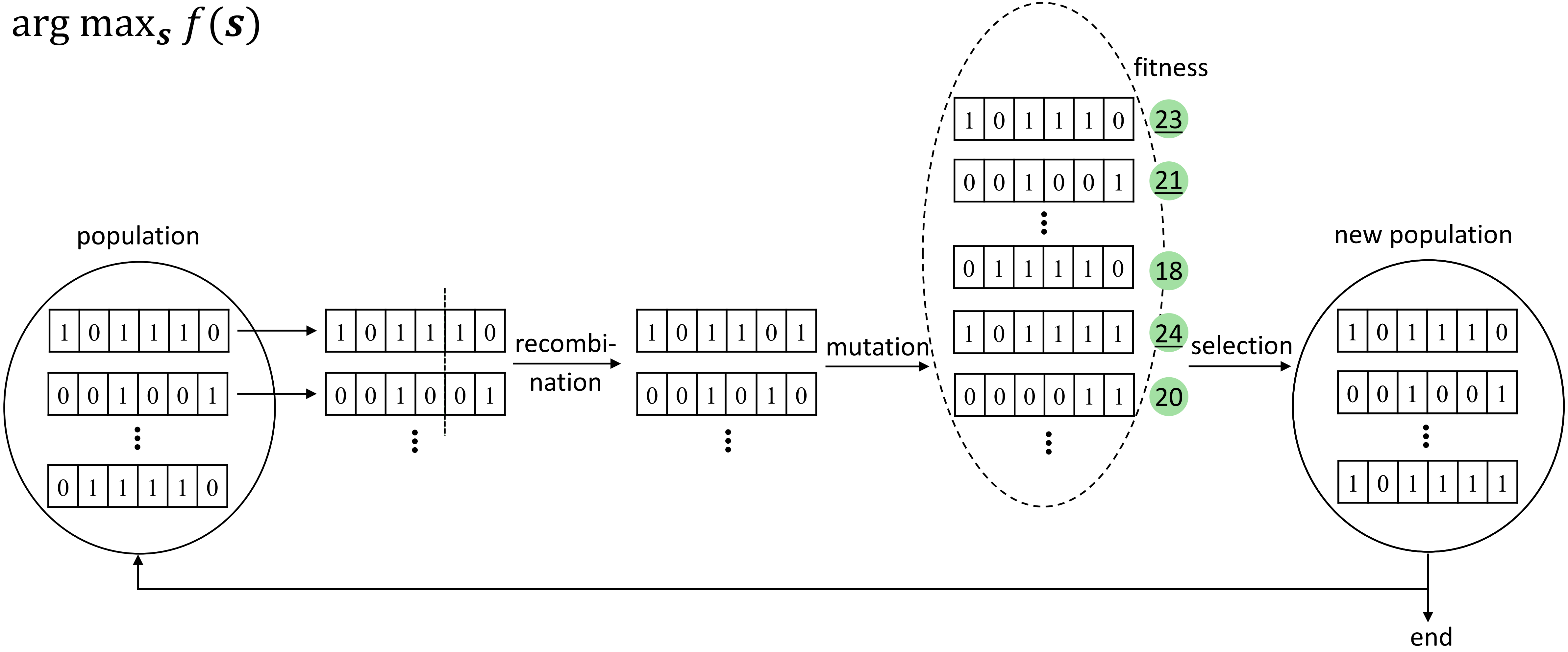
A Typical Evolutionary Process

$\arg\max_{\boldsymbol{s}} f(\boldsymbol{s})$



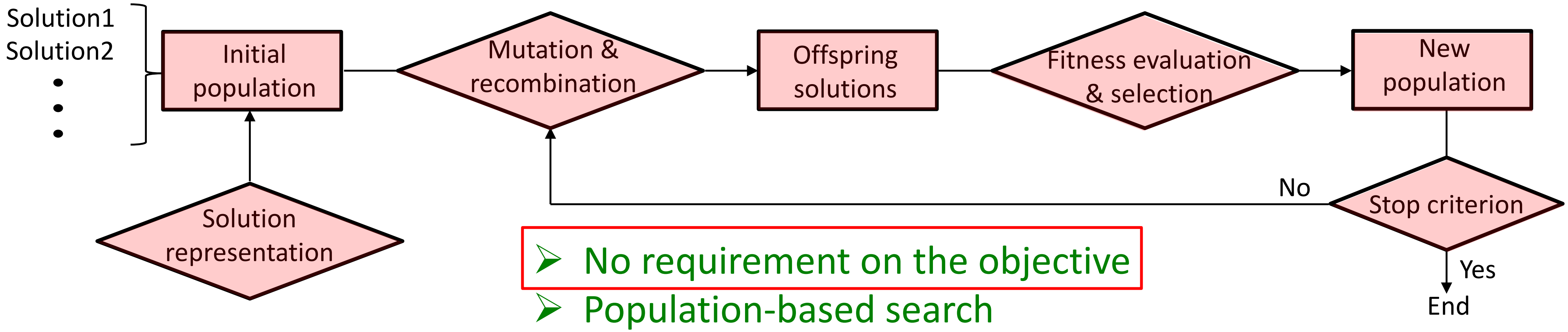
A Typical Evolutionary Process

$\arg\max_{\boldsymbol{s}} f(\boldsymbol{s})$



Evolutionary Algorithms

The general structure of EAs



- No requirement on the objective
- Population-based search

Thus, EAs can be applied to solve complicated optimization problems

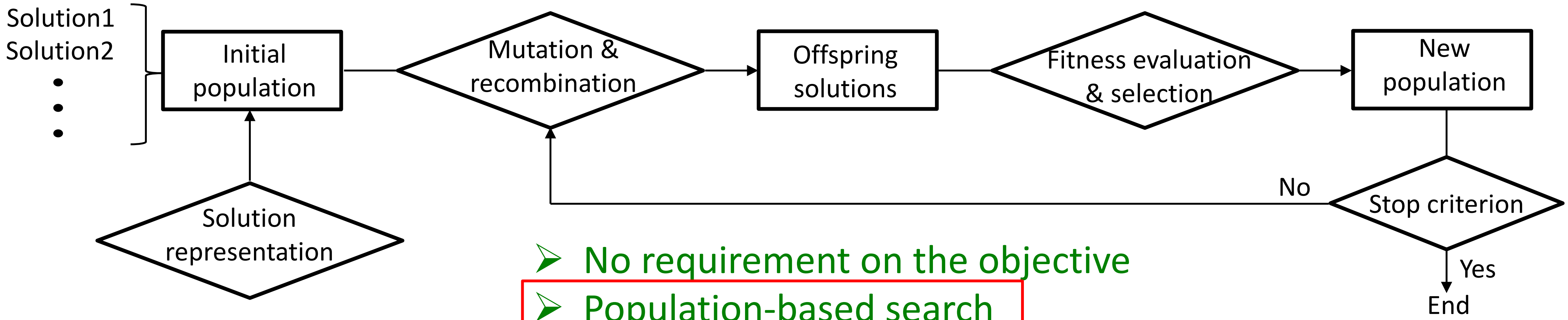
- non-differentiable, non-continuous
- without explicit objective formulation

black-box

- multiple objective functions

Evolutionary Algorithms

The general structure of EAs



- ➢ No requirement on the objective
- ➢ Population-based search

Thus, EAs can be applied to solve complicated optimization problems

- non-differentiable, non-continuous
- without explicit objective formulation
- multiple objective functions

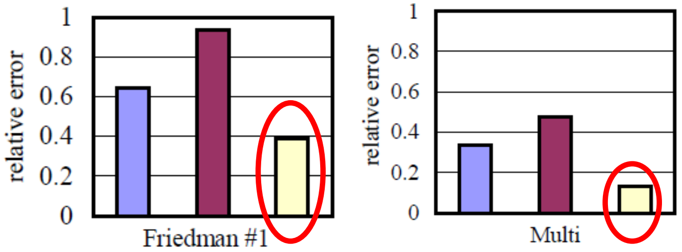
Multi-objective EAs (MOEAs)

e.g., NSGA-II [Deb et al., TEC'02]

Google scholar: 52081

Applications of Evolutionary Algorithms

Evolutionary selective ensemble



achieves smaller error by using fewer learners

[Zhou et al., AIJ'02]

Evolutionary neural architecture search

Study	Params.	C10+	C100+	Reachable?
Maxout (Goodfellow et al., 2013)	–	90.7%	61.4%	No
Network in Network (Lin et al., 2013)	–	91.2%	–	No
All-CNN (Springenberg et al., 2014)	1.3 M	92.8%	66.3%	Yes
Deeply Supervised (Lee et al., 2015)	–	92.0%	65.4%	No
Highway (Srivastava et al., 2015)	2.3 M	92.3%	67.6%	No
ResNet (He et al., 2016)	1.7 M	93.4%	72.8%†	Yes
Evolution (ours)	5.4 M 40.4 M	94.6%	 77.0%	N/A
Wide ResNet 28-10 (Zagoruyko & Komodakis, 2016)	36.5 M	96.0%	80.0%	Yes
Wide ResNet 40-10+d/o (Zagoruyko & Komodakis, 2016)	50.7 M	96.2%	81.7%	No
DenseNet (Huang et al., 2016a)	25.6 M	96.7%	82.8%	No

achieves competitive performance to the hand-designed models

[Google, ICML'17]

Evolutionary multitask learning

Model	imagenet2012	cifar100	cifar10
ViT L/16 fine-tuning (Dosovitskiy et al., 2021)	85.30	93.25	99.15
μ2Net after 5 task iterations	86.38	94.75	99.35
μ2Net after 10 task iterations	86.66	94.67	99.38
μ2Net cont. after adding VTAB-full tasks	**86.74**	94.67	99.41
μ2Net cont. after adding VDD tasks	**86.74**	94.74	99.43
μ2Net cont. after adding all 69 tasks	**86.74**	**94.95**	**99.49**

achieves competitive results on 69 public image classification tasks

[Gesmundo & Dean, 2022]

better

SOTA: 99.40% [Touvron et al., ICCV'21]

Applications of Evolutionary Algorithms

High-speed train head design

evolve

save 19% energy

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

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

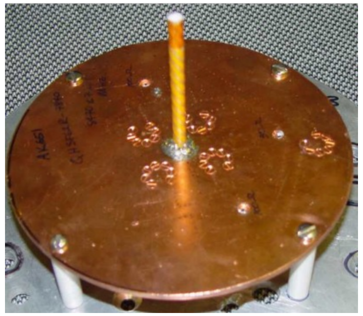
[1]*Central Japan Railway Company, Tokyo, Japan,* [2]*West Japan Railway Company, Osaka, Japan*

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.

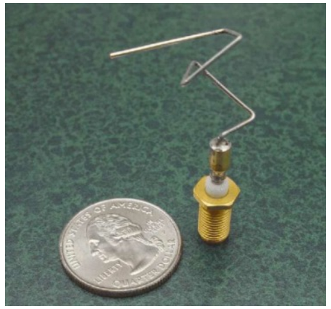
On the Tokaido Shinkansen line, Series N700 cars save 19% energy than Series 700 cars, despite a 30% increase in the output of their traction equipment for higher-speed operation (Fig. 4).

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

Antenna design



evolve



38% efficiency

93% efficiency

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

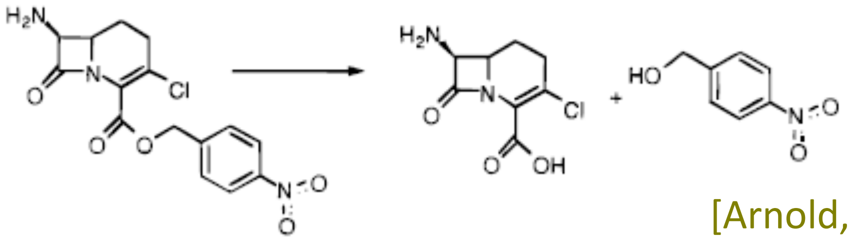
Applications of Evolutionary Algorithms

The Nobel Prize in Chemistry 2018

© Nobel Media AB. Photo: A. Mahmoud	© Nobel Media AB. Photo: A. Mahmoud	© Nobel Media AB. Photo: A. Mahmoud
Frances H. Arnold	George P. Smith	Sir Gregory P. Winter
Prize share: 1/2	Prize share: 1/4	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."

Protein design



[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

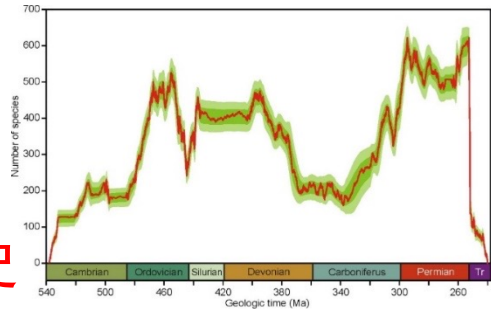
三叶虫、笔石、珊瑚、腕足…

利用化石记录

重现生命演化历史

生物多样性变化曲线

物种数



地质时期

南京大学地球科学与工程学院研究成果

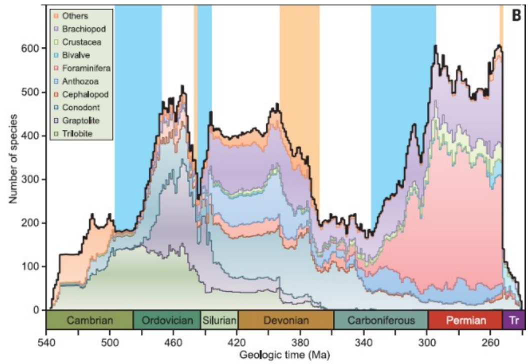
全球第一条高精度
海洋生物多样性变化曲线

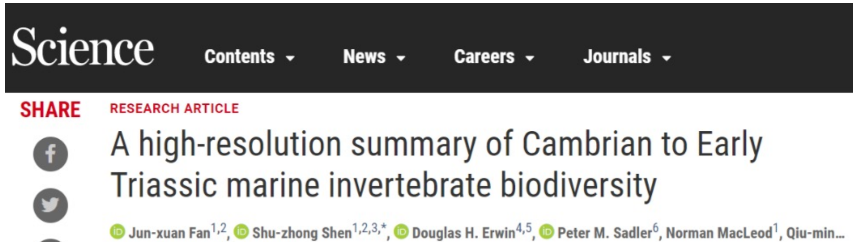
中国的地层剖面数据

3122个剖面

11268个物种

演化算法




Science Contents News Careers Journals
SHARE
RESEARCH ARTICLE
A high-resolution summary of Cambrian to Early Triassic marine invertebrate biodiversity
Jun-xuan Fan[1,2], Shu-zhong Shen[1,2,3,*], Douglas H. Erwin[4,5], Peter M. Sadler[6], Norman MacLeod[1], Qiu-min...

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

Limitations of Evolutionary Algorithms

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

Evolutionary algorithms have yielded encouraging empirical outcomes, but

Evolutionary neural architecture search [Google, ICML'17]

Study	Params.	C10+	C100+	Reachable?
Maxout (Goodfellow et al., 2013)	–	90.7%	61.4%	No
Network in Network (Lin et al., 2013)	–	91.2%	–	No
All-CNN (Springenberg et al., 2014)	1.3 M	92.8%	66.3%	Yes
Deeply Supervised (Lee et al., 2015)	–	92.0%	65.4%	No
Highway (Srivastava et al., 2015)	2.3 M	92.3%	67.6%	No
ResNet (He et al., 2016)	1.7 M	93.4%	72.8%†	Yes
Evolution (ours)	5.4 M 40.4 M	94.6%	77.0%	N/A
Wide ResNet 28-10 (Zagoruyko & Komodakis, 2016)	36.5 M	96.0%	80.0%	Yes
Wide ResNet 40-10+d/o (Zagoruyko & Komodakis, 2016)	50.7 M	96.2%	81.7%	No
DenseNet (Huang et al., 2016a)	25.6 M	96.7%	82.8%	No

2750 gpu days

Macro placement [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

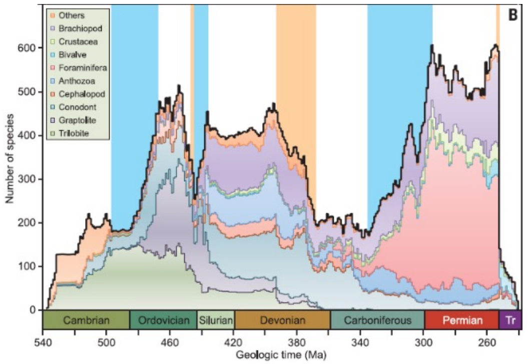
3122个剖面

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



How to improve the efficiency?

Especially for high-dimensional and expensive scenarios

Evolutionary Learning

Evolutionary Learning (EL)

Combine evolutionary algorithms and machine learning to better solve complex black-box optimization problems

- Learn surrogate models to help the optimization, e.g., preselection and Bayesian optimization
- Learn effective search subspaces
- Learn components (e.g., reproduction and selection operators) of EAs
- Learn to (dynamically) configure hyper-parameters of EAs
- Learn to select a proper EA
- Learn a universal EA

Evolutionary Learning

Evolutionary Learning (EL)

Combine evolutionary algorithms and machine learning to better solve complex black-box optimization problems

The theoretical foundation of EAs is underdeveloped

L. Valiant

Turing Award in 2010

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 …"

- EAs: Highly randomized and complex
- Problems: Complex

⇩

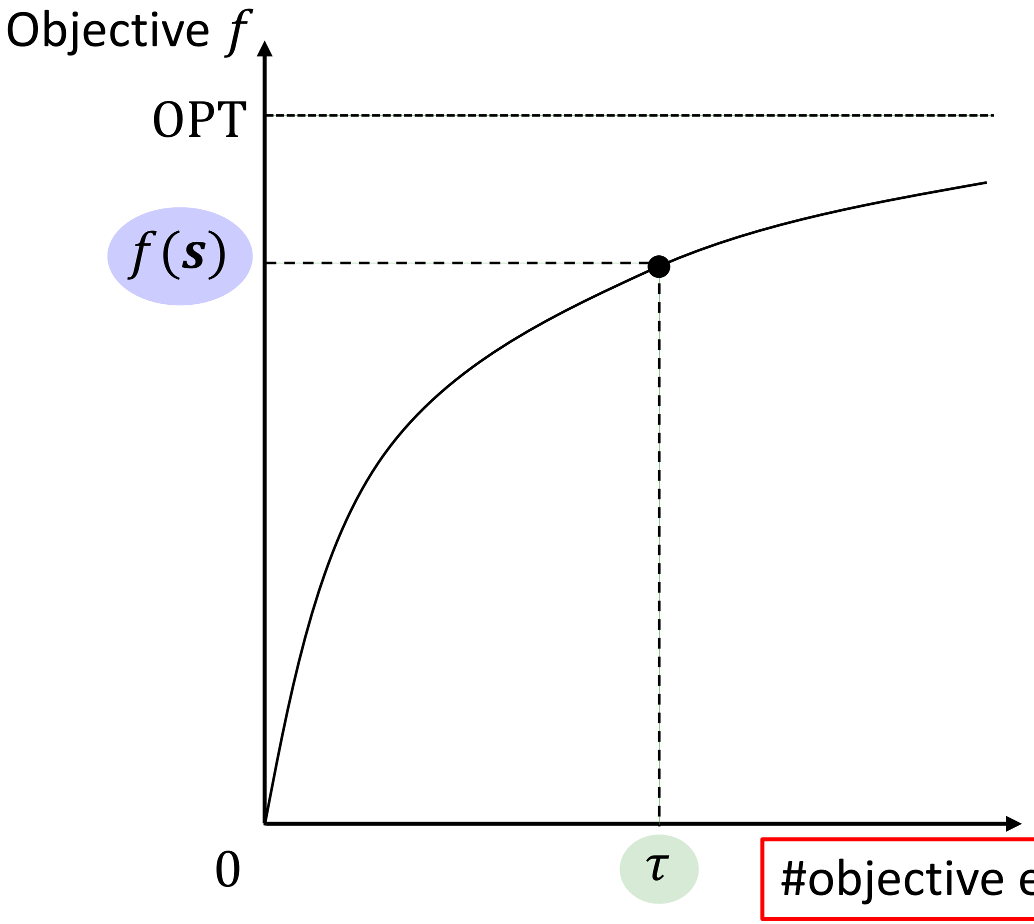
Theoretical analysis is very difficult

Outline

- **Build theoretical foundation of EAs**
  - **Theoretical analysis tools, influence analysis of major factors of EAs**
- Develop better EL algorithms
  - Efficient EL, dynamic algorithm configuration, algorithm selection, universal EL
- Apply EL to complex optimization in learning, industry, and science
  - Subset selection, electronic design automation, origin and evolution of life

Running Time Complexity

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



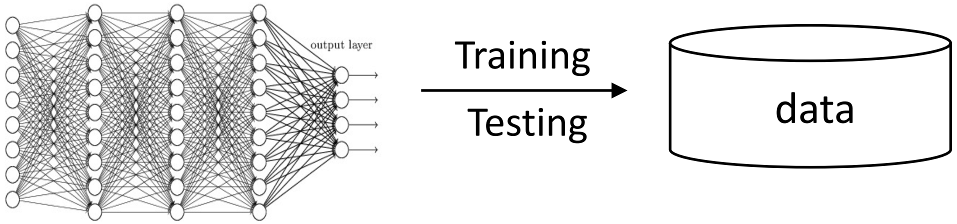
#objective evaluations

Running time τ:

#objective evaluations until finding desired solutions for the first time

the process with the highest cost of EA

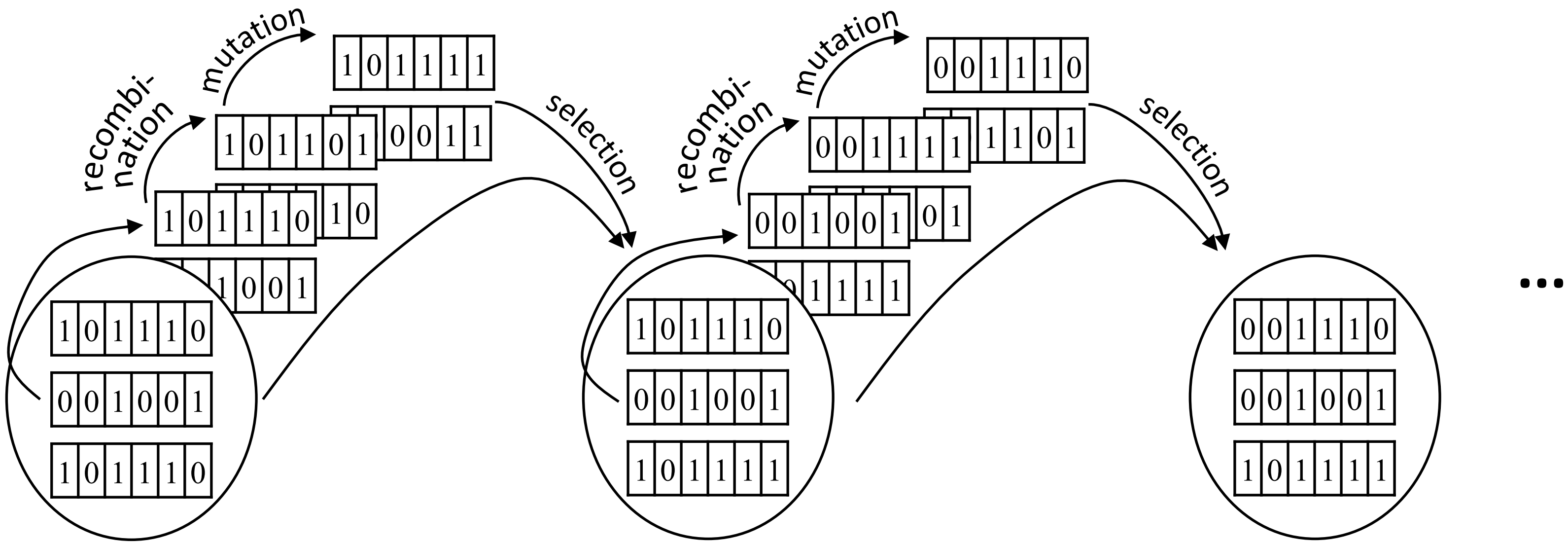
e.g., model evaluation



An Example

Hard to be analyzed directly

EA with recombination



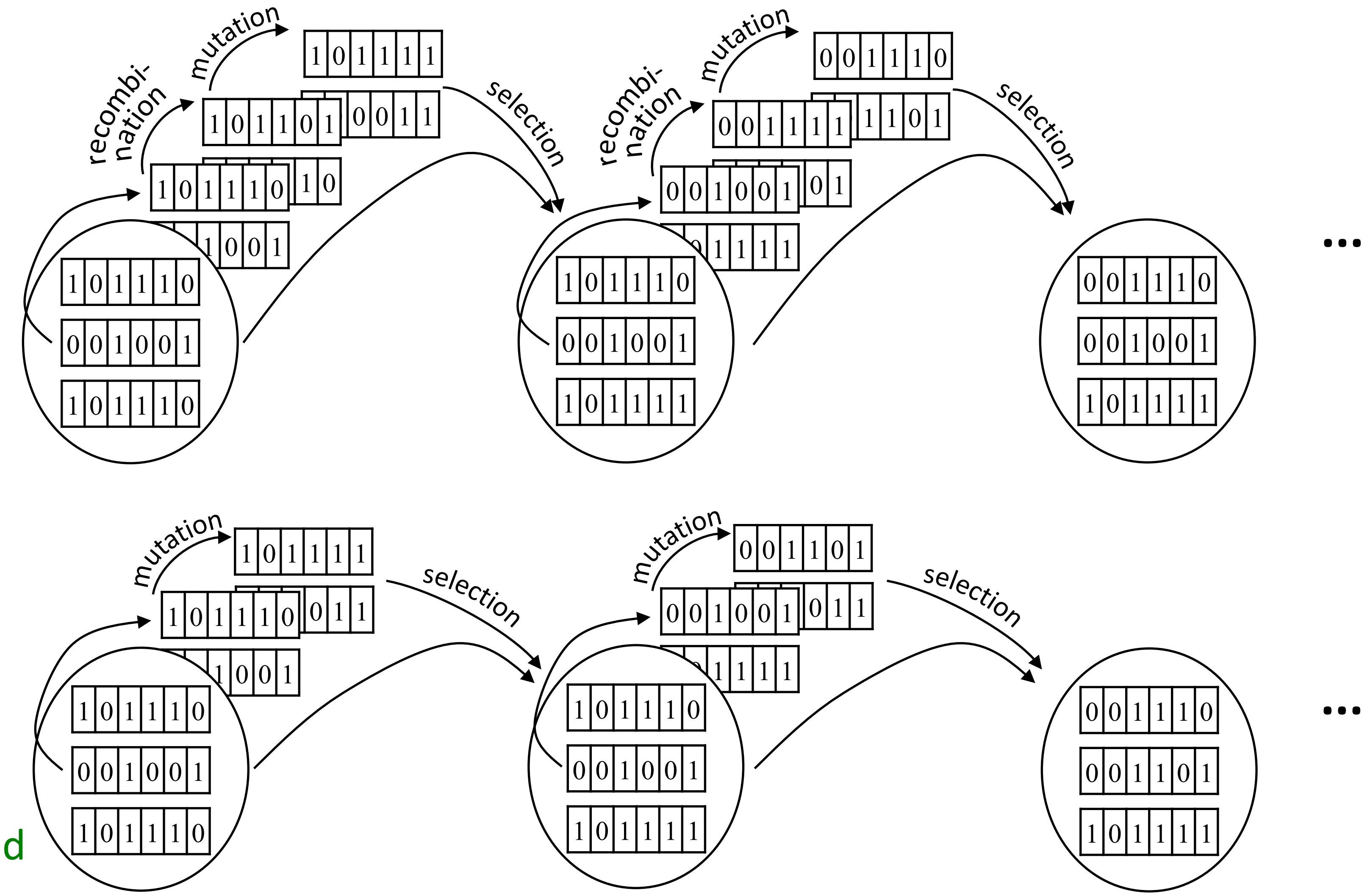
An Example

Hard to be analyzed directly

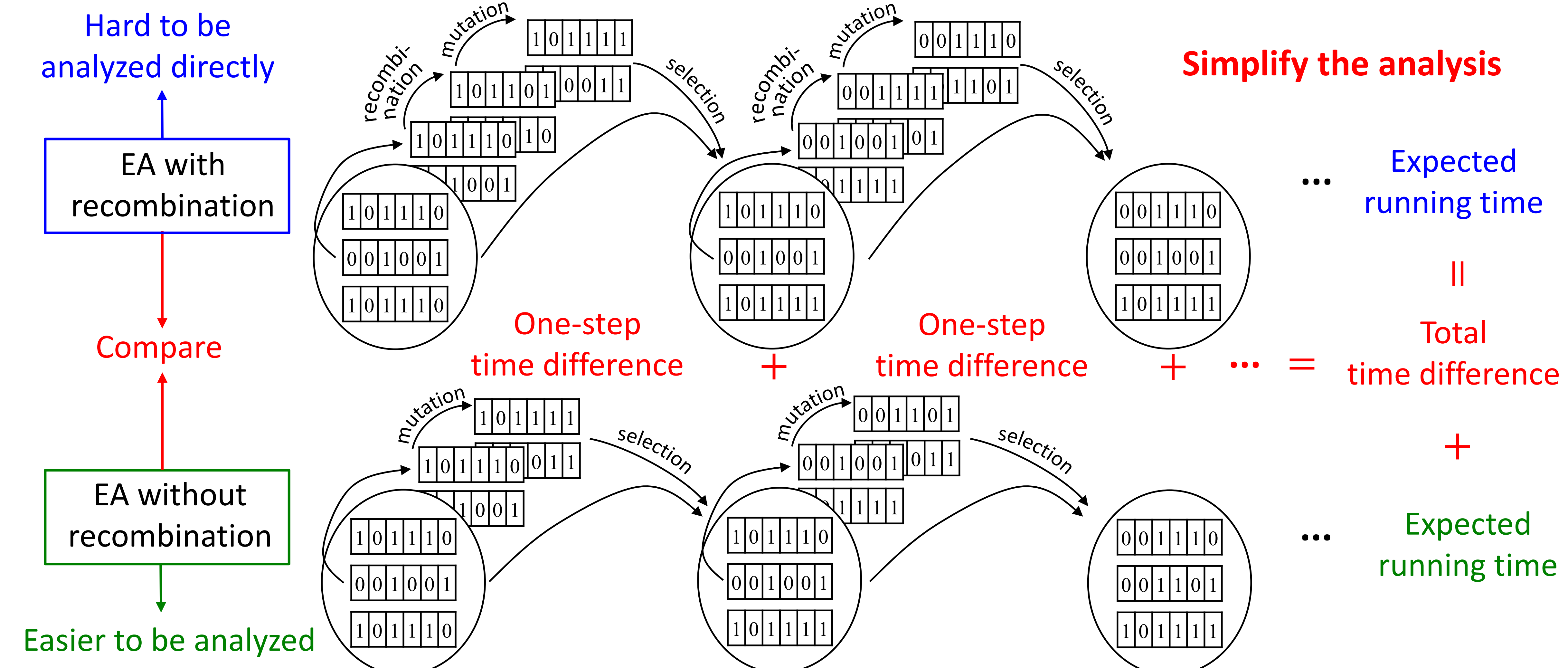
EA with recombination

EA without recombination

Easier to be analyzed

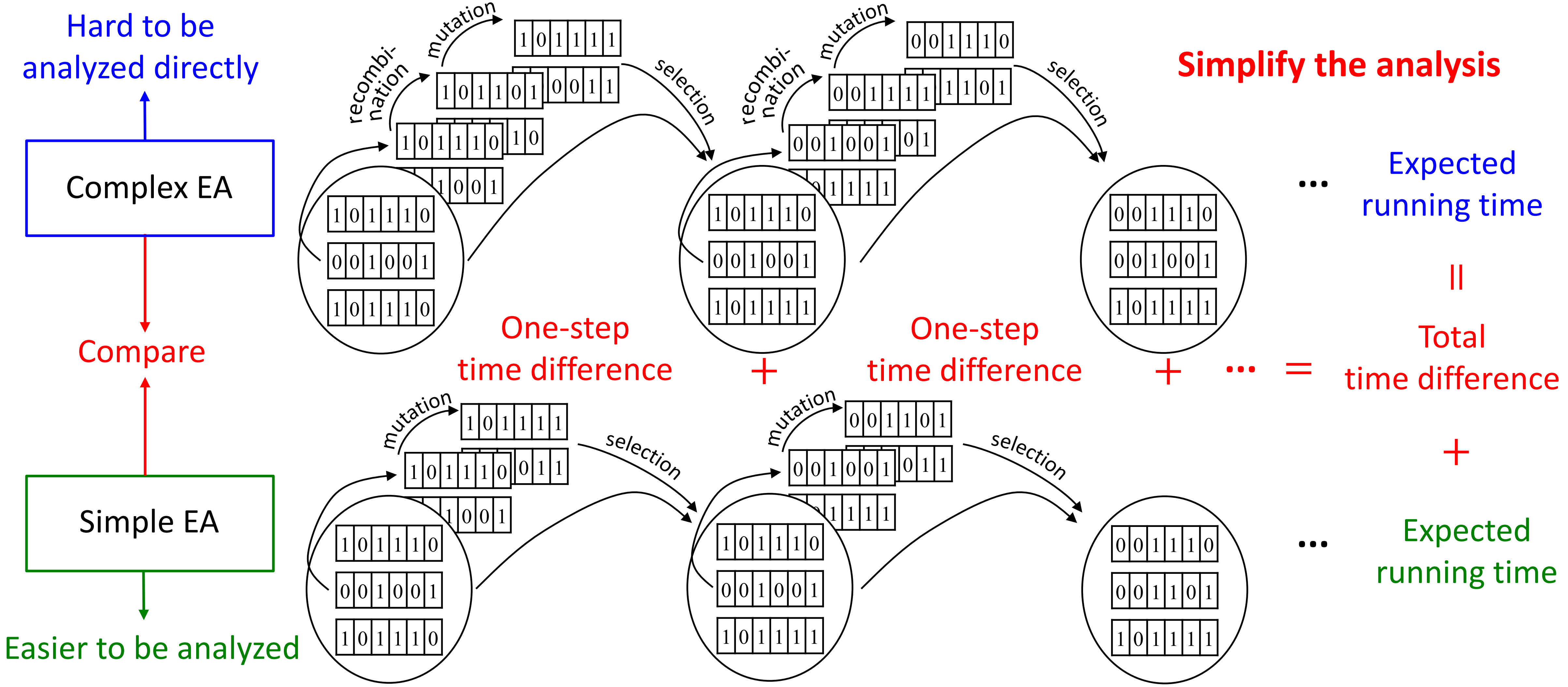


An Example



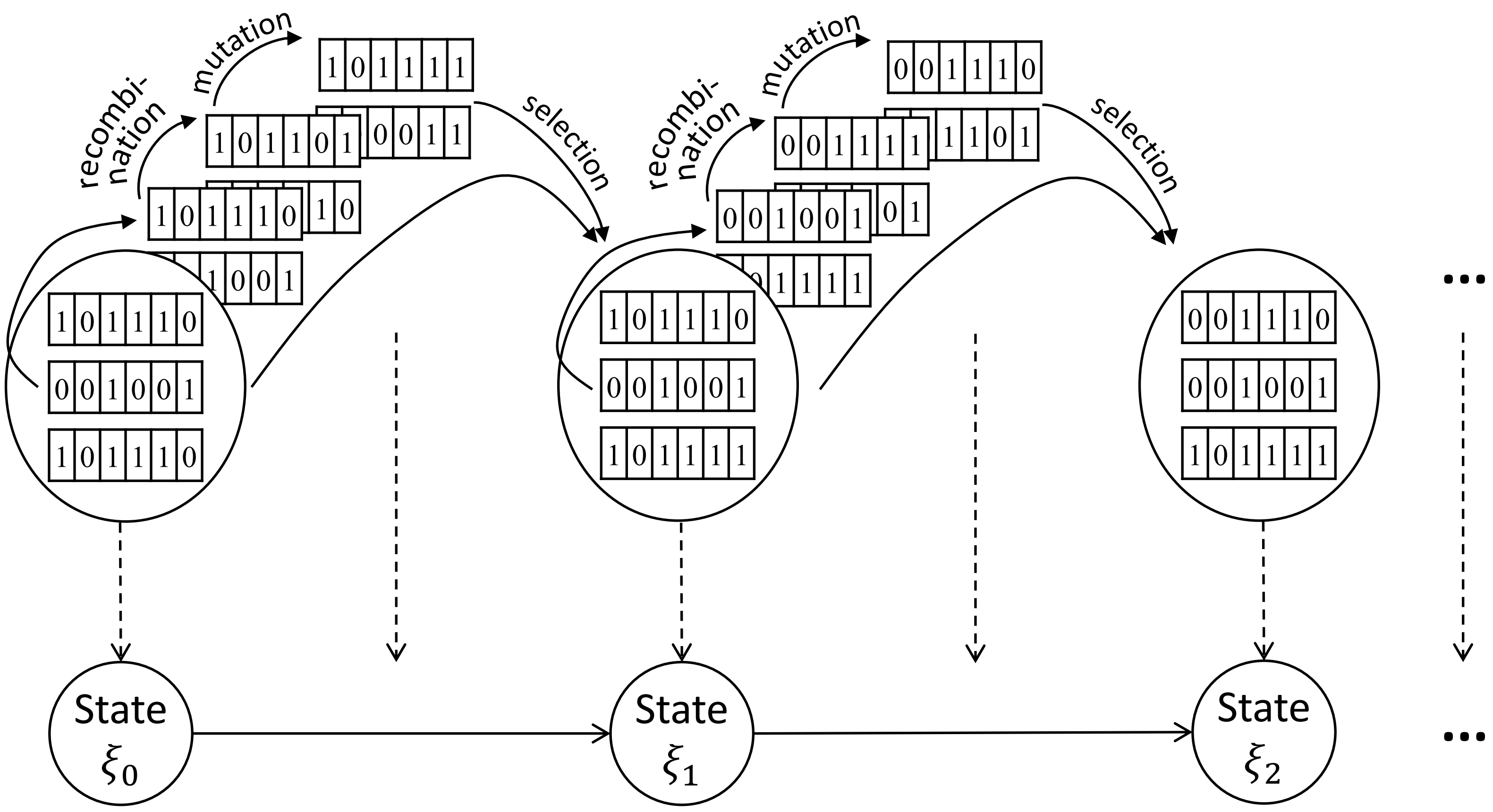
Hard to be analyzed directly
EA with recombination
Compare
EA without recombination
Easier to be analyzed
recombination
mutation
selection
One-step time difference
+
One-step time difference
+
…
=
Total time difference
Simplify the analysis
…
Expected running time
=
+
…
Expected running time

An Example


Hard to be analyzed directly
Complex EA
Compare
Simple EA
Easier to be analyzed
recombi-nation
mutation
selection
One-step time difference
+
One-step time difference
+
...
=
Simplify the analysis
Expected running time
||
Total time difference
+
Expected running time

Switch Analysis

Model an EA process as a Markov chain

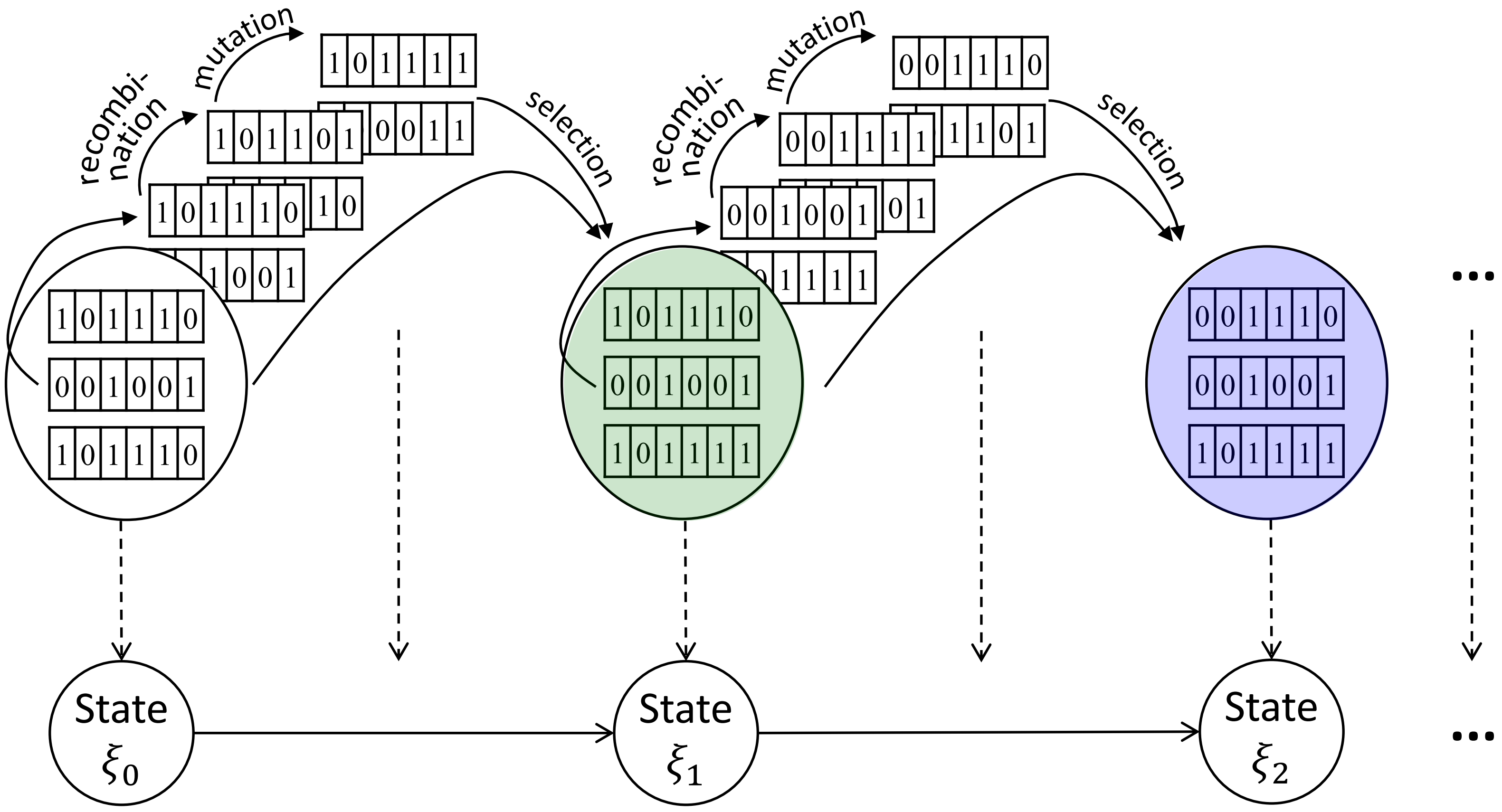


Markov chain?

[Yu, Qian and Zhou, IEEE Trans. Evolutionary Computation 2015]

Switch Analysis

Model an EA process as a Markov chain



The generation of the next population only depends on the current population

Markov property

$P(\xi_{t+1} \mid \xi_t, \dots, \xi_0) = P(\xi_{t+1} \mid \xi_t)$

[Yu, Qian and Zhou, IEEE Trans. Evolutionary Computation 2015]

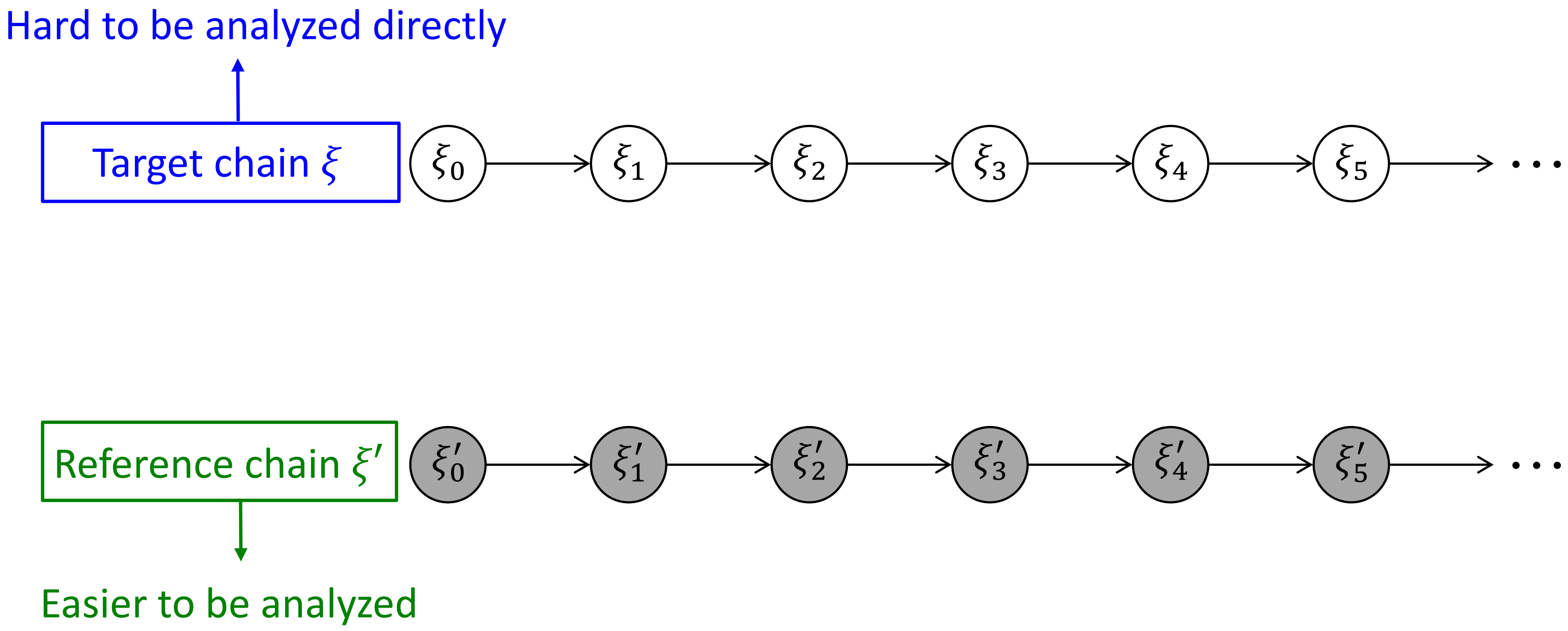
Switch Analysis

Hard to be analyzed directly

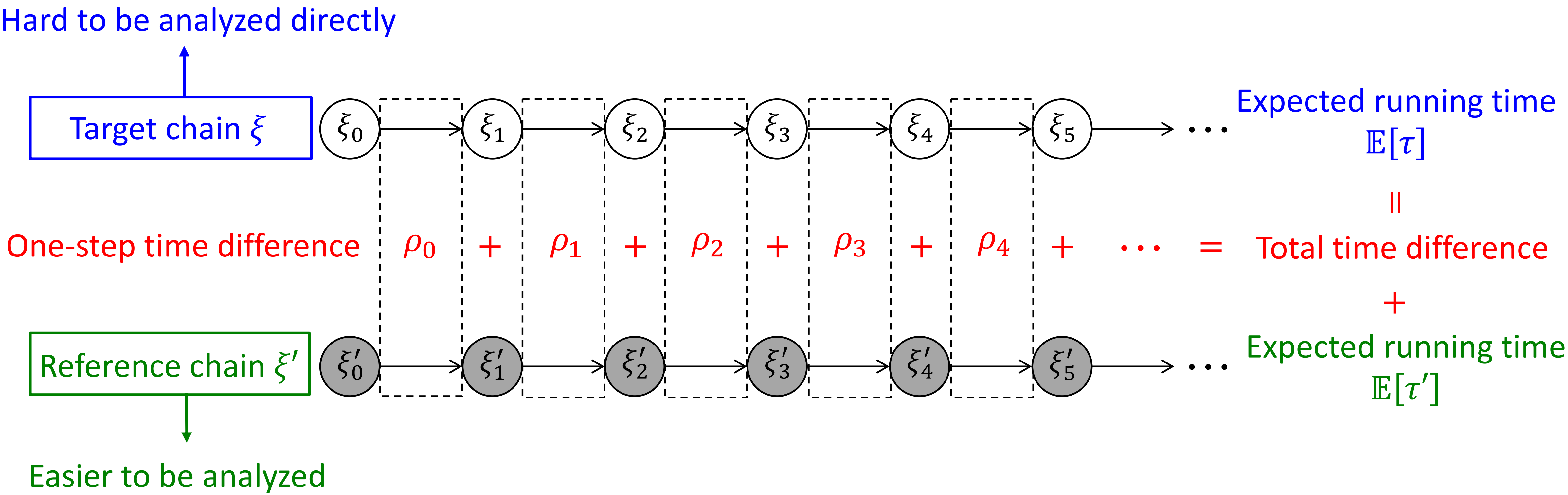
$\uparrow$

Target chain ξ $\quad \xi_0 \to \xi_1 \to \xi_2 \to \xi_3 \to \xi_4 \to \xi_5 \to \cdots$

Switch Analysis



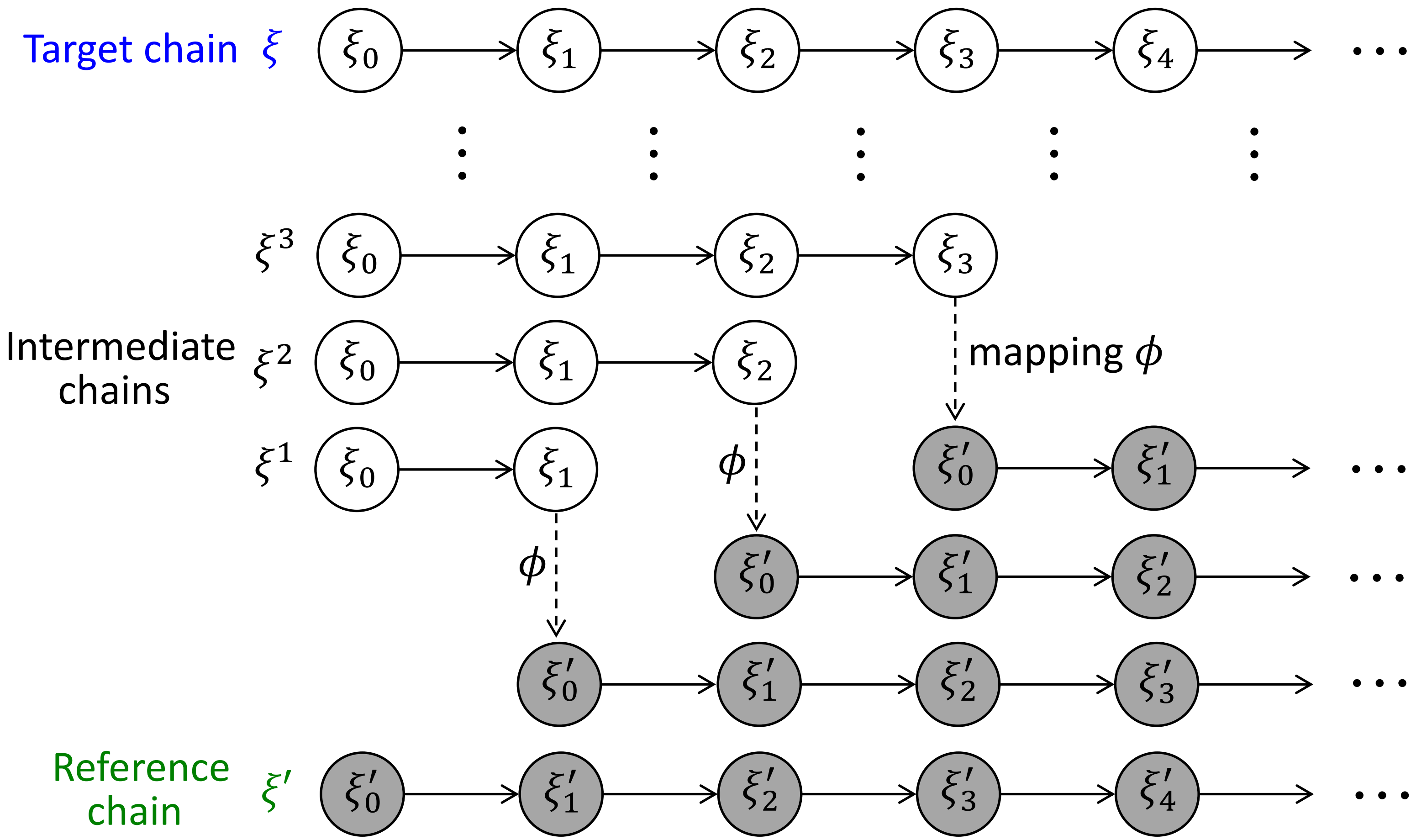
Switch Analysis



How to estimate one-step time difference ρ_t?

Switch Analysis

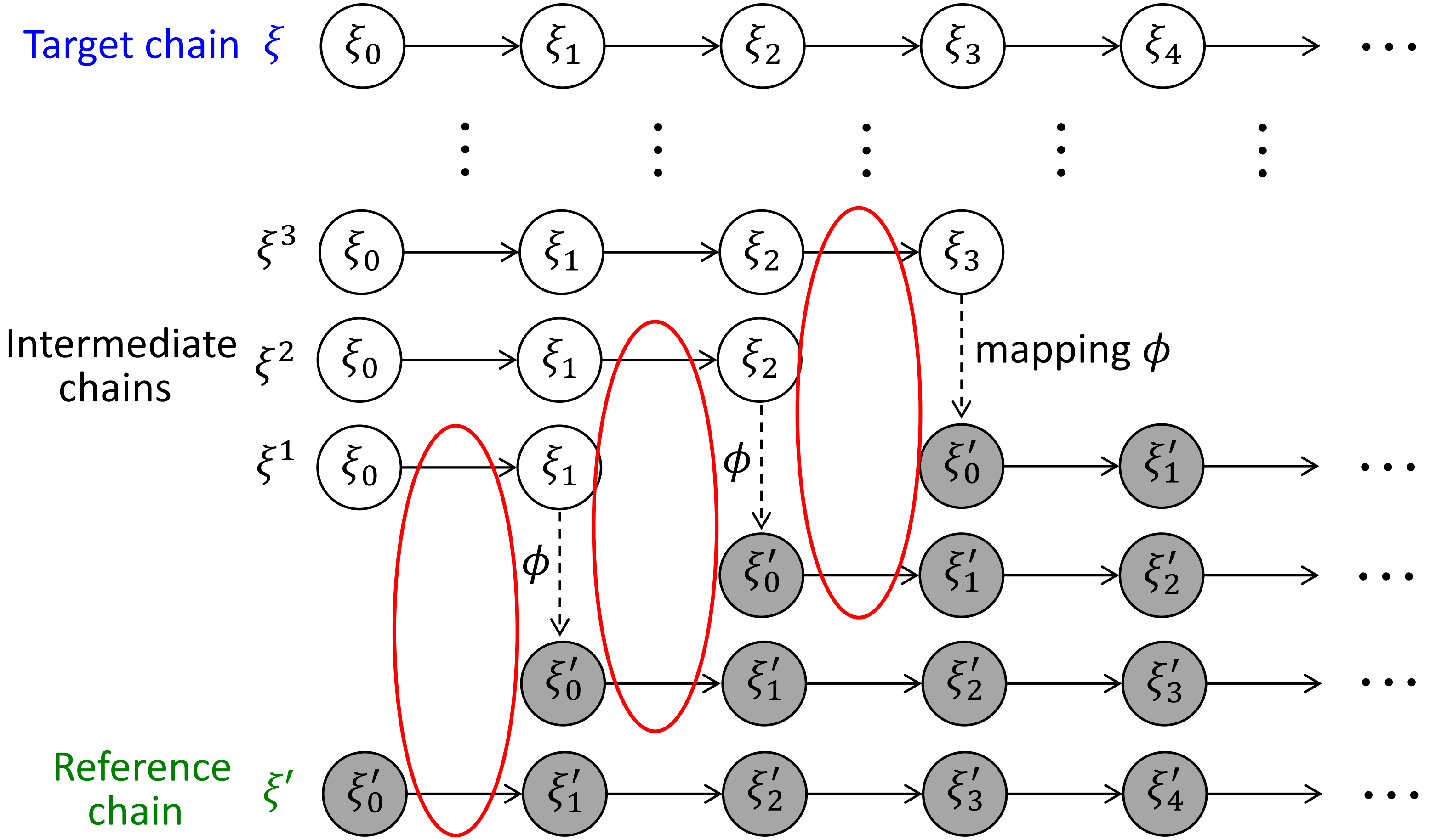
How to estimate one-step time difference ρ_t?



[Yu, Qian and Zhou, IEEE Trans. Evolutionary Computation 2015]

Switch Analysis

How to estimate one-step time difference ρ_t?



Time difference between adjacent intermediate chains

$$\mathbb{E}[\tau] - \mathbb{E}[\tau^\infty] = \rho_\infty$$

$$\vdots$$

$$\mathbb{E}[\tau^{t+1}] - \mathbb{E}[\tau^t] = \rho_t$$

$$\vdots$$

$$\mathbb{E}[\tau^3] - \mathbb{E}[\tau^2] = \rho_2$$

$$\mathbb{E}[\tau^2] - \mathbb{E}[\tau^1] = \rho_1$$

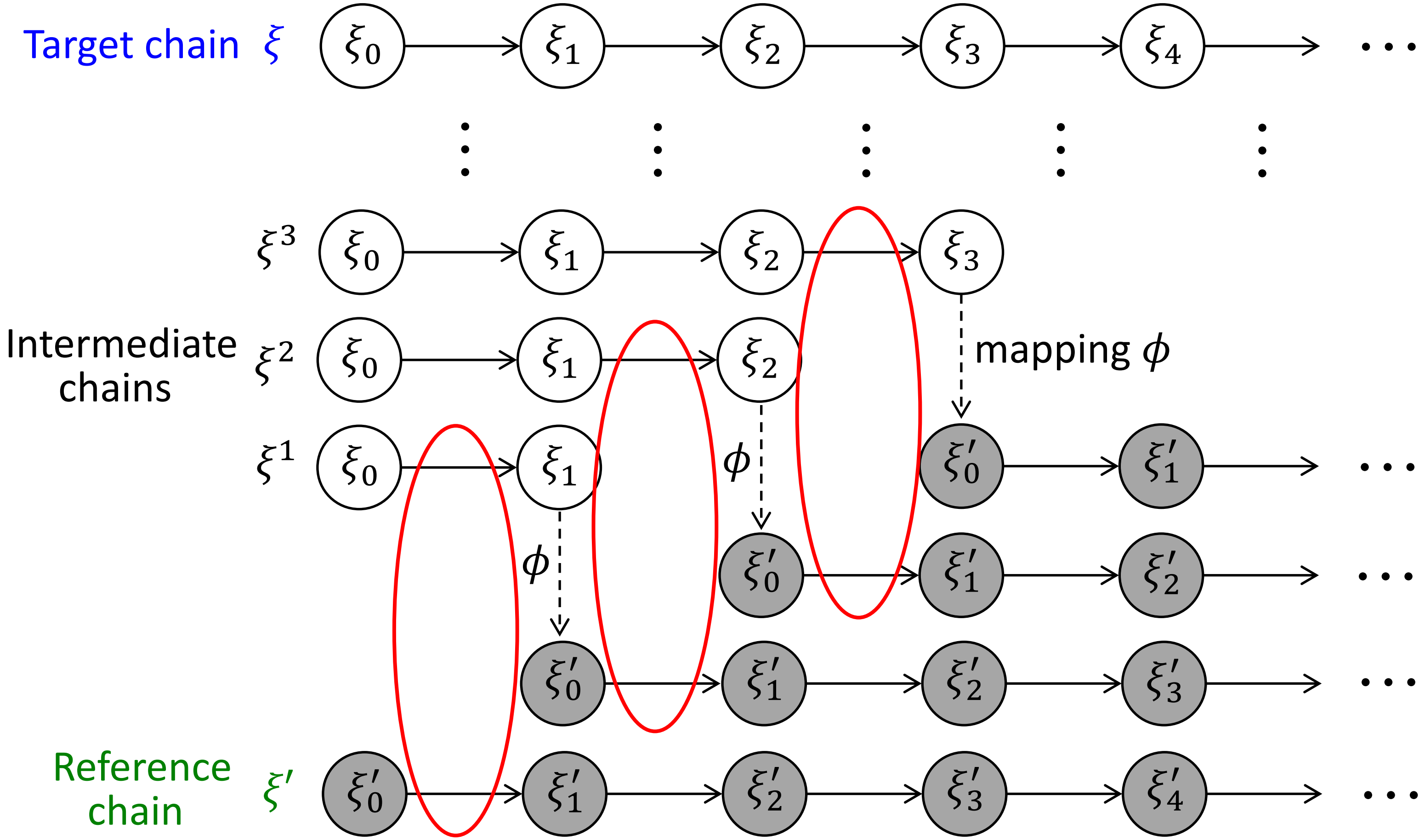
$$\mathbb{E}[\tau^1] - \mathbb{E}[\tau'] = \rho_0$$

One-step time difference

[Yu, Qian and Zhou, IEEE Trans. Evolutionary Computation 2015]

Switch Analysis

How to estimate one-step time difference ρ_t?



Time difference between adjacent intermediate chains

$$\mathbb{E}[\tau] - \cancel{\mathbb{E}[\tau^\infty]} = \rho_\infty$$
$$+$$
$$\vdots$$
$$+$$
$$\cancel{\mathbb{E}[\tau^{t+1}]} - \cancel{\mathbb{E}[\tau^t]} = \rho_t$$
$$+$$
$$\vdots$$
$$+$$
$$\cancel{\mathbb{E}[\tau^3]} - \cancel{\mathbb{E}[\tau^2]} = \rho_2$$
$$+$$
$$\cancel{\mathbb{E}[\tau^2]} - \cancel{\mathbb{E}[\tau^1]} = \rho_1$$
$$+$$
$$\cancel{\mathbb{E}[\tau^1]} - \mathbb{E}[\tau'] = \rho_0$$

Total time difference

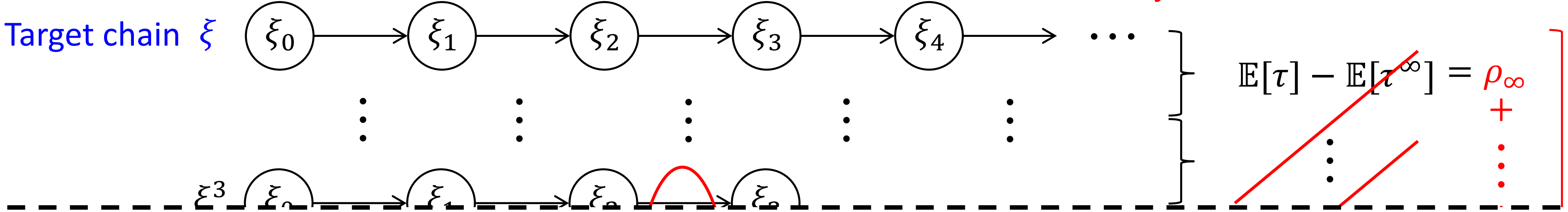
$$\mathbb{E}[\tau] - \mathbb{E}[\tau']$$

[Yu, Qian and Zhou, IEEE Trans. Evolutionary Computation 2015]

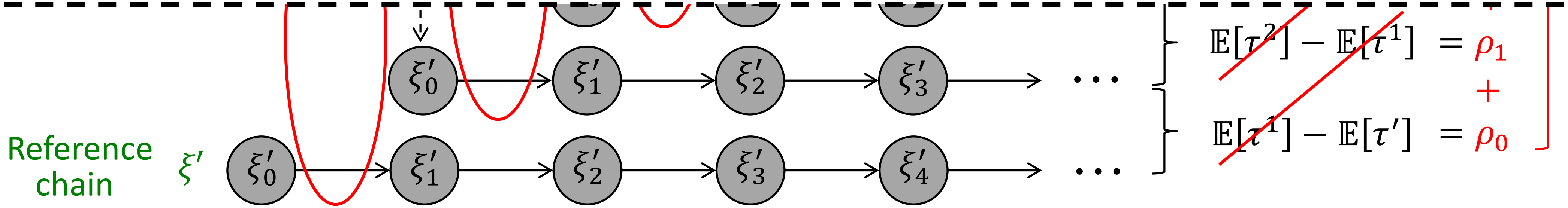
Switch Analysis

How to estimate one-step time difference ρ_t?

Time difference between adjacent intermediate chains



Theorem (Switch Analysis). *Given two absorbing Markov chains $\xi \in \mathcal{X}$ and $\xi' \in \mathcal{Y}$, a series of values $\{\rho_t \in \mathbb{R}\}_{t=0}^{+\infty}$ with $\rho = \sum_{t=0}^{+\infty} \rho_t$ and a right (or left)-aligned mapping $\phi: \mathcal{X} \to \mathcal{Y}$, if $\mathbb{E}[\tau \mid \xi_0 \sim \pi_0]$ is finite and $\forall t$: $\sum_{\mathrm{x}\in\mathcal{X},\mathrm{y}\in\mathcal{Y}} \pi_t(\mathrm{x}) P(\xi_{t+1} \in \phi^{-1}(\mathrm{y}) \mid \xi_t = \mathrm{x}) \mathbb{E}[\tau' \mid \xi'_0 = \mathrm{y}] \leq$ (or $\geq$) $\sum_{\mathrm{u},\mathrm{y}\in\mathcal{Y}} \pi_t^{\phi}(\mathrm{u}) P(\xi'_1 = \mathrm{y} \mid \xi'_0 = \mathrm{u}) \mathbb{E}[\tau' \mid \xi'_1 = \mathrm{y}] + \rho_t$, where $\pi_t^{\phi}(\mathrm{y}) = \pi_t(\phi^{-1}(\mathrm{y})) = \sum_{\mathrm{x}\in\phi^{-1}(\mathrm{y})} \pi_t(\mathrm{x})$, we have $\mathbb{E}[\tau \mid \xi_0 \sim \pi_0] \leq$ (or $\geq$) $\mathbb{E}[\tau' \mid \xi'_0 \sim \pi_0^{\phi}] + \rho$.*



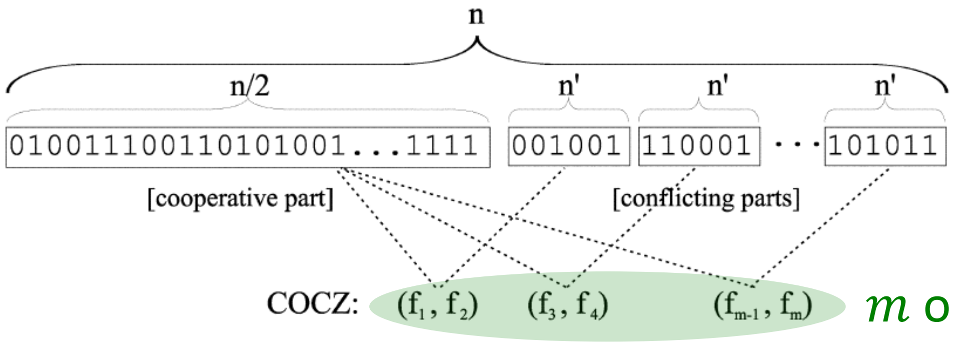
Application of Switch Analysis

Example: Analyze GSEMO solving the mCOCZ problem

GSEMO:

1. $\boldsymbol{s} \coloneqq$ randomly selected from $\{0,1\}^n$; $P \coloneqq \{\boldsymbol{s}\}$
2. Repeat until some termination criterion is met
3. Choose $\boldsymbol{s}$ from P uniformly at random
4. apply bit-wise mutation on $\boldsymbol{s}$ to generate $\boldsymbol{s}'$
5. if $\nexists\, \boldsymbol{z} \in P$ such that $\boldsymbol{z} \succ \boldsymbol{s}'$
6. $P := (P - \{\boldsymbol{z} \in P \mid \boldsymbol{s}' \succcurlyeq \boldsymbol{z}\}) \cup \{\boldsymbol{s}'\}$

mCOCZ : $\max_{\boldsymbol{s}\in\{0,1\}^n} \; (f_1(\boldsymbol{s}), f_2(\boldsymbol{s}), \ldots, f_m(\boldsymbol{s}))$



m objectives

L. Thiele

Professor, ETH Zurich
EDAA Lifetime Achievement Award

Previous results:

$O(n^{m+1})$

[Laumanns, Thiele and Zitzler, TEC'04]

tighter by n

Switch analysis:

$O(n^m)$

[Bian et al., IJCAI'18]

Influence Analysis of Recombination Operator

Mutation and recombination are two characterizing features of EAs

Example of mutation

Parent	1	0	1	1	1	0	0	0
Offspring	1	0	1	1	0	0	0	0

simulates the gene altering of a chromosome in biological mutation

Example of recombination

More complicated

Parent1	1	0	1	1	1	0	0	0
Parent2	0	0	1	0	1	0	1	0
Offspring1	1	0	1	1	1	0	1	0
Offspring2	0	0	1	0	1	0	0	0

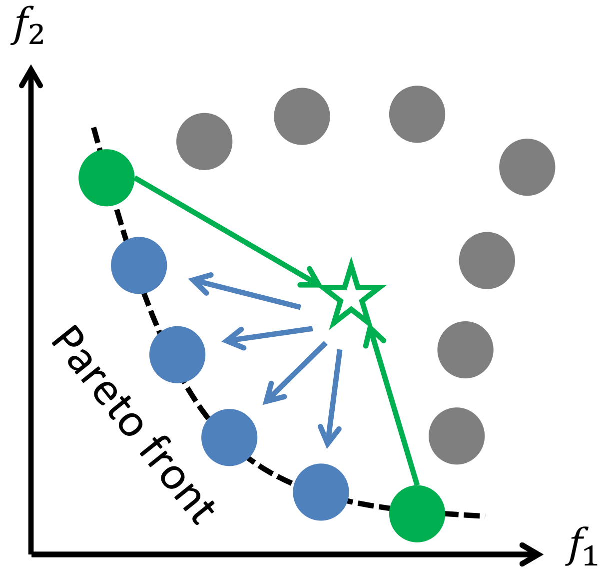
simulates the chromosome exchange phenomena in zoogamy reproductions

Influence Analysis of Recombination Operator

Our result:

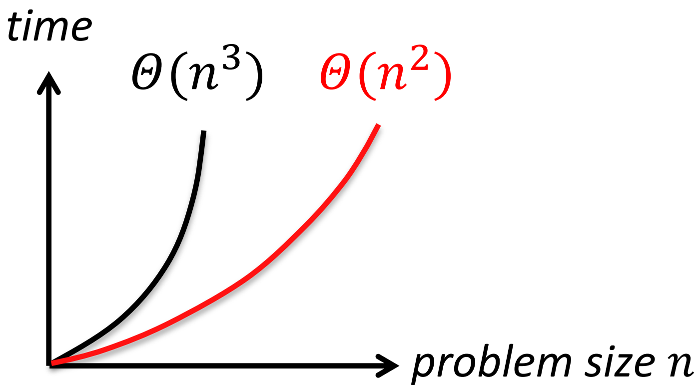
Recombination can accelerate the filling of the Pareto front by recombining diverse Pareto optimal solutions

Unique to multi-objective optimization



Example: MOEA solving the LOTZ Problem

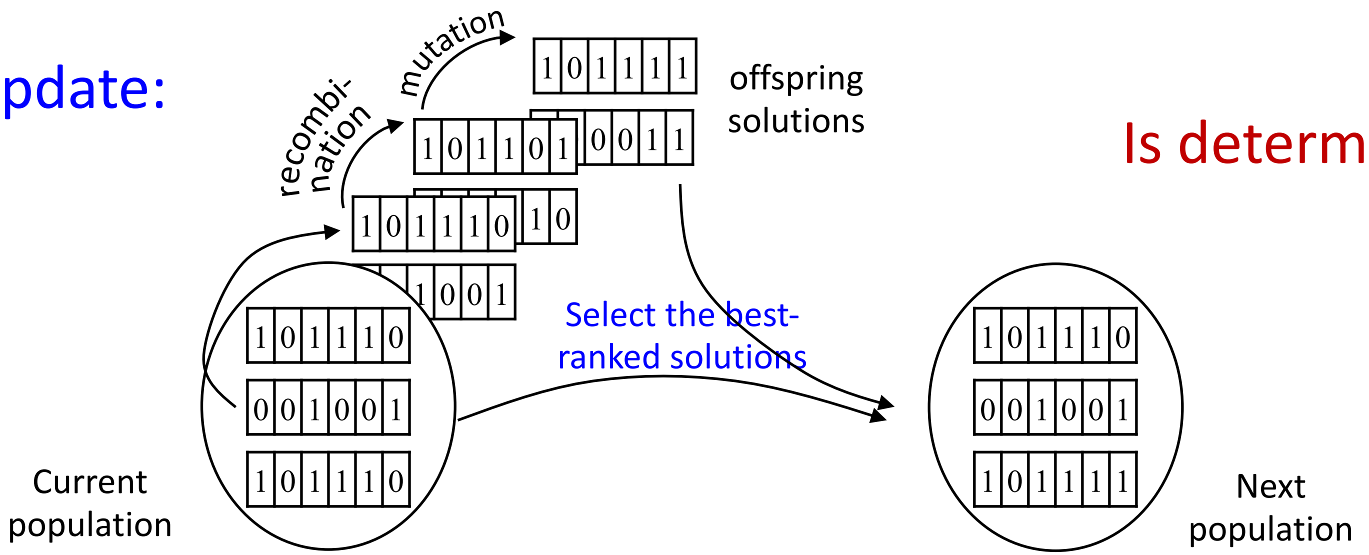
Expected running time $\Theta(n^3) \xrightarrow{\text{recombination}} \Theta(n^2)$



[Qian et al., Artificial Intelligence 2013, ACM GECCO'11 Best Theory Paper Award]

Influence Analysis of Population Update

Population update:



Is deterministic population update always better?

NO!

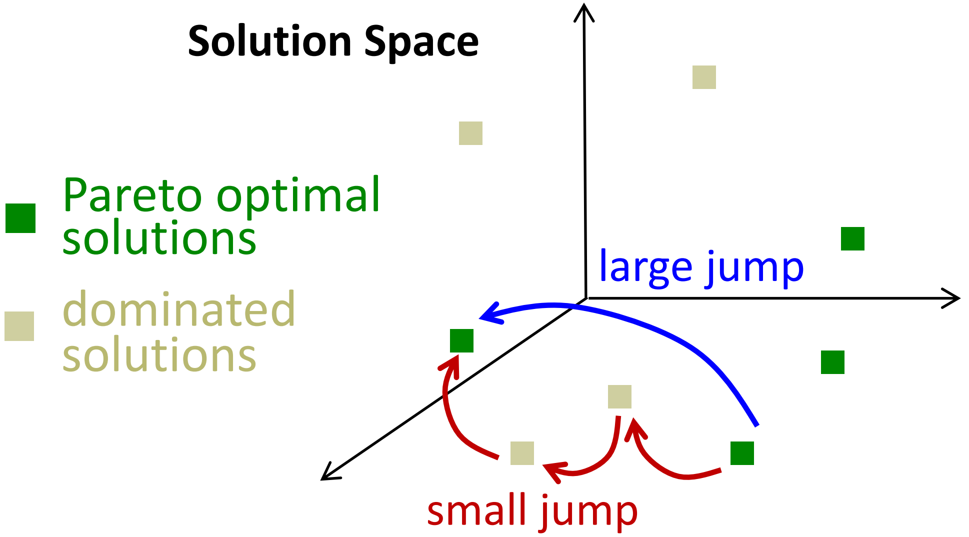
The prominent feature in population update of MOEAs: greedy and deterministic

- the next-generation population is formed by selecting the best-ranked solutions
- e.g., NSGA-II (Google scholar: 52081), SPEA-II (Google scholar: 10384), SMS-EMOA (Google scholar: 2145), MOEA/D (Google scholar: 9166), …

Influence Analysis of Population Update

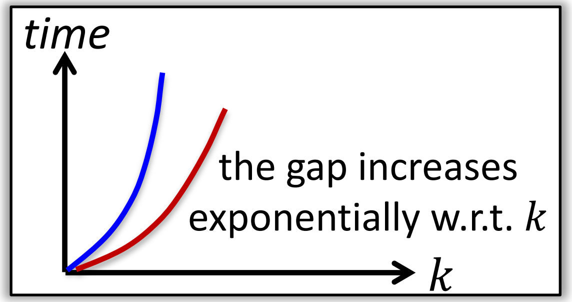
Our result:

By introducing randomness into population update, MOEAs can go across inferior regions around Pareto optimal solutions more easily



Example: SMS-EMOA solving the OneJumpZeroJump problem

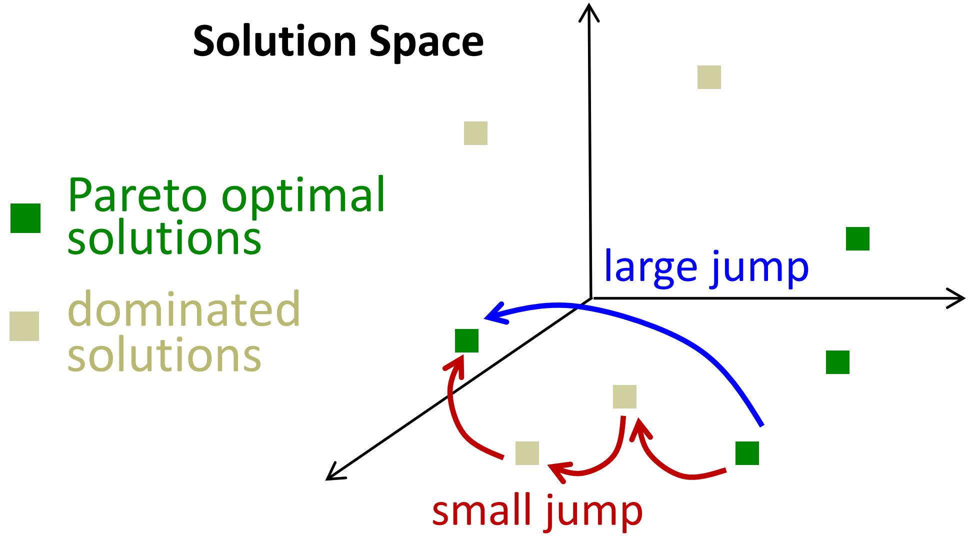
	Deterministic		Stochastic
Expected running time	$\Omega(n^k)$	accelerated by $2^{k/4}/\mu^2$	$O(\mu^2 n^k / 2^{k/4})$



Influence Analysis of Population Update

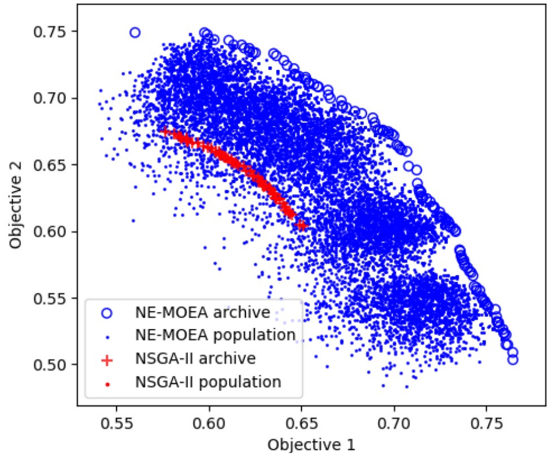
Our result:

By introducing randomness into population update, MOEAs can go across inferior regions around Pareto optimal solutions more easily

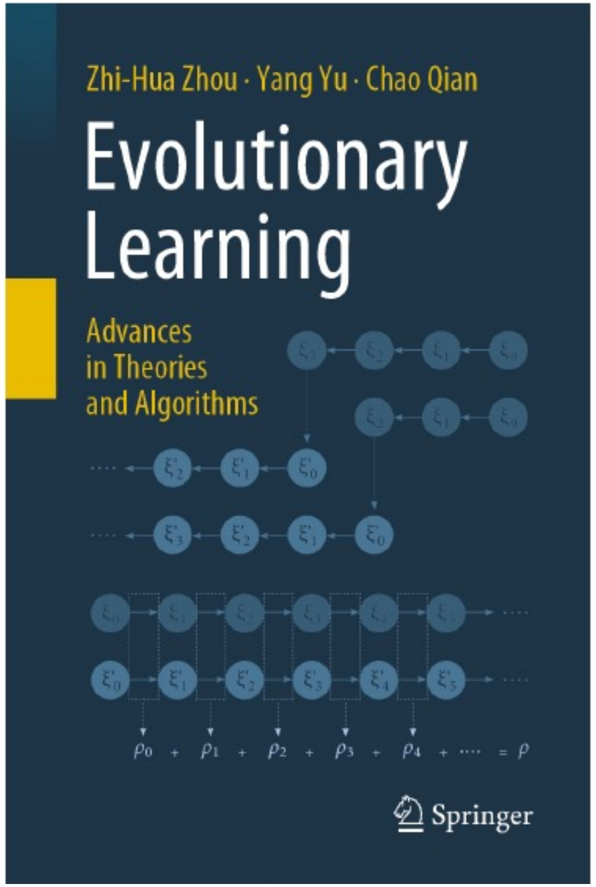


- Challenge the common practice of MOEAs, i.e., deterministic population update
- Encourage the exploration of developing new MOEAs in the area

For example, [Liang, Li and Lehre, GECCO'23]:



For details



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

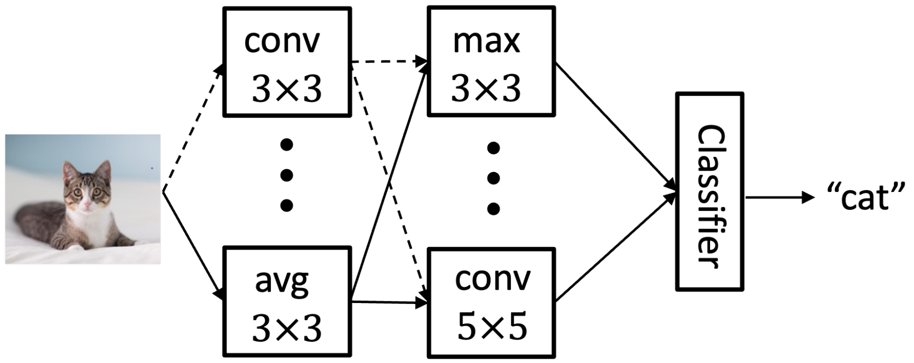
Outline

- Build theoretical foundation of EAs
  - Theoretical analysis tools, influence analysis of major factors of EAs
- **Develop better EL algorithms**
  - **Efficient EL, dynamic algorithm configuration, algorithm selection, universal EL**
- Apply EL to complex optimization in learning, industry, and science
  - Subset selection, electronic design automation, origin and evolution of life

High-dimensional Black-box Optimization

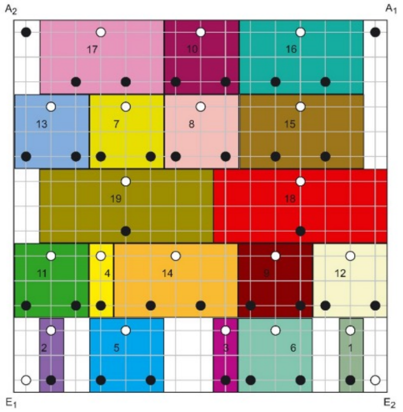
The black-box optimization problems can be high-dimensional

Neural architecture search



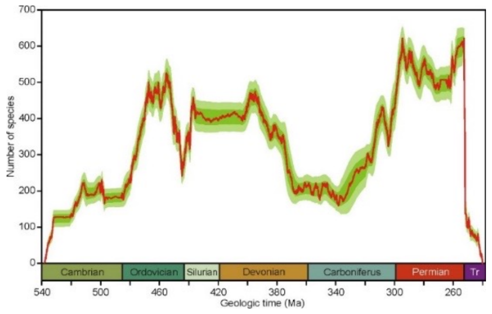
Thousands of hyper-parameters

Macro placement



Thousands of macros

Origin and evolution of life



Thousands of species

How to develop efficient EL for high-dimensional black-box optimization?

High-dimensional Black-box Optimization

Current approaches usually solve high-dimensional BBO in a low-dimensional subspace:

1. Obtain a low-dimensional subspace
2. Optimize in the low-dimensional subspace
3. Project the low-dimensional solution back to the high-dimensional space

- **Decomposition**: f can be decomposed into the sum of low-dimensional functions
  [Kandasamy et al., ICML'15; Rolland et al., AISTATS'18]
- **Embedding**: only a few dimensions affect f significantly
  [Wang et al., JAIR'16; Letham et al., NeurIPS'20]
- **Variable selection**: only a few *axis-aligned* dimensions affect f significantly
  [Li et al., IJCAI'17]

Variable Selection for High-dimensional Black-box Optimization

Variable selection: Simpler than embedding and can reduce the runtime

Dropout [Li et al., IJCAI'17] select d variables randomly and optimize the selected variables

- Select d variables randomly
- Optimize the selected variables
- Use "fill-in" strategy to obtain the unselected variables

Select d variables randomly ⇒ **Select important variables automatically**

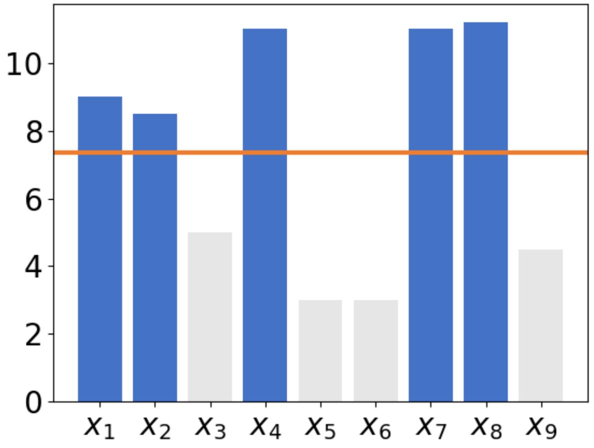
- A metric for the importance of variables
- An algorithm to select important variables

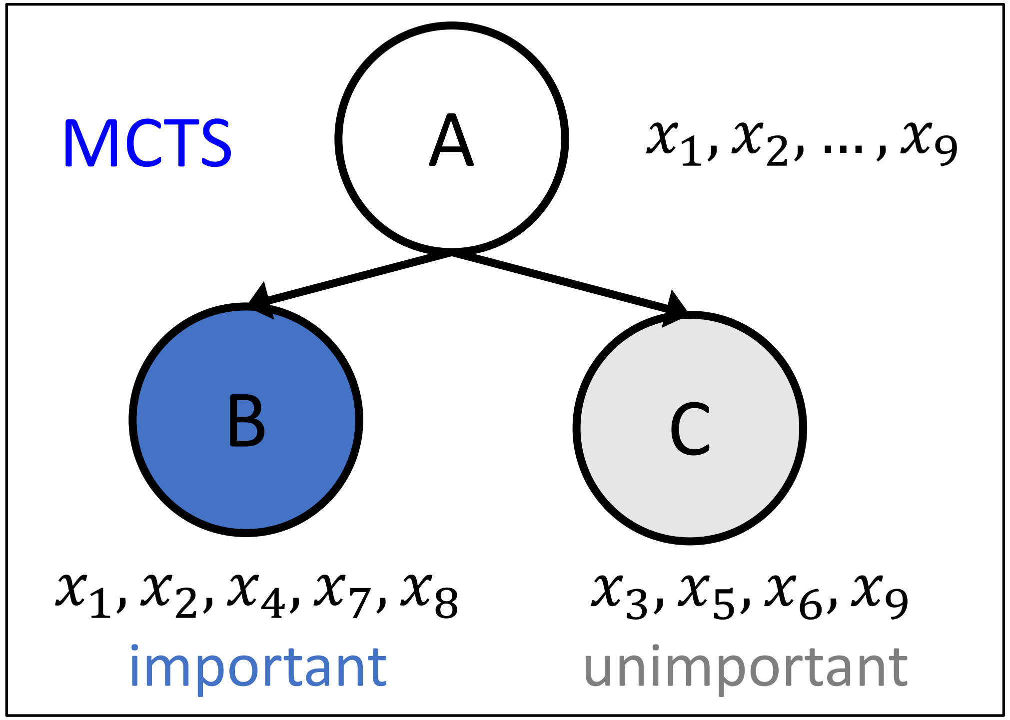
MCTS based Variable Selection

Can be combined with any BBO algorithm

Importance

variable score $\boldsymbol{s} = \dfrac{\sum_{(\mathbb{M},\mathcal{D})\in\mathbb{D}} \sum_{(\boldsymbol{x}^i, y^i)\in\mathcal{D}} y^i \cdot g(\mathbb{M})}{\sum_{(\mathbb{M},\mathcal{D})\in\mathbb{D}} |\mathcal{D}| \cdot g(\mathbb{M})} = \dfrac{\text{The sum of query evaluations using each variable}}{\text{The number of queries using each variable}}$





UCB-based selection

$$\underbrace{v}_{\text{exploitation}} + 2C_p\sqrt{\underbrace{2(\log n_p)/n}_{\text{exploration}}}$$

[Song, Xue, Huang, and Qian, NeurIPS 2022 Spotlight]

MCTS based Variable Selection

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

Theorem: $\forall \delta \in (0,1)$, let $\beta_t = 2\log\left(\frac{4\pi_t}{\delta}\right) + 2d_t \log(d_t t^2 br\sqrt{\log(\frac{4Da}{\delta})})$ and $L = b\sqrt{\log\frac{4Da}{\delta}}$, and $\{\pi_t\}_{t\geq 1}$ satisfies $\sum_{t\geq 1}\pi_t^{-1} = 1$ and $\pi_t > 0$. Let $\beta_T^* = \max_{1\leq i\leq T} \beta_t$. At iteration T,

$$R_T \leq \sqrt{C_1 T \beta_T^* \gamma_T} + 2\alpha_{max} + 2\sum_{t=1}^{T}\sum_{i\in[D]\setminus\mathbb{M}_t} \alpha_i^* Lr$$

Regret from unselected variables

Cumulative regret $R_T = \sum_{t=1}^{T}(f(x^*) - f(x^t))$

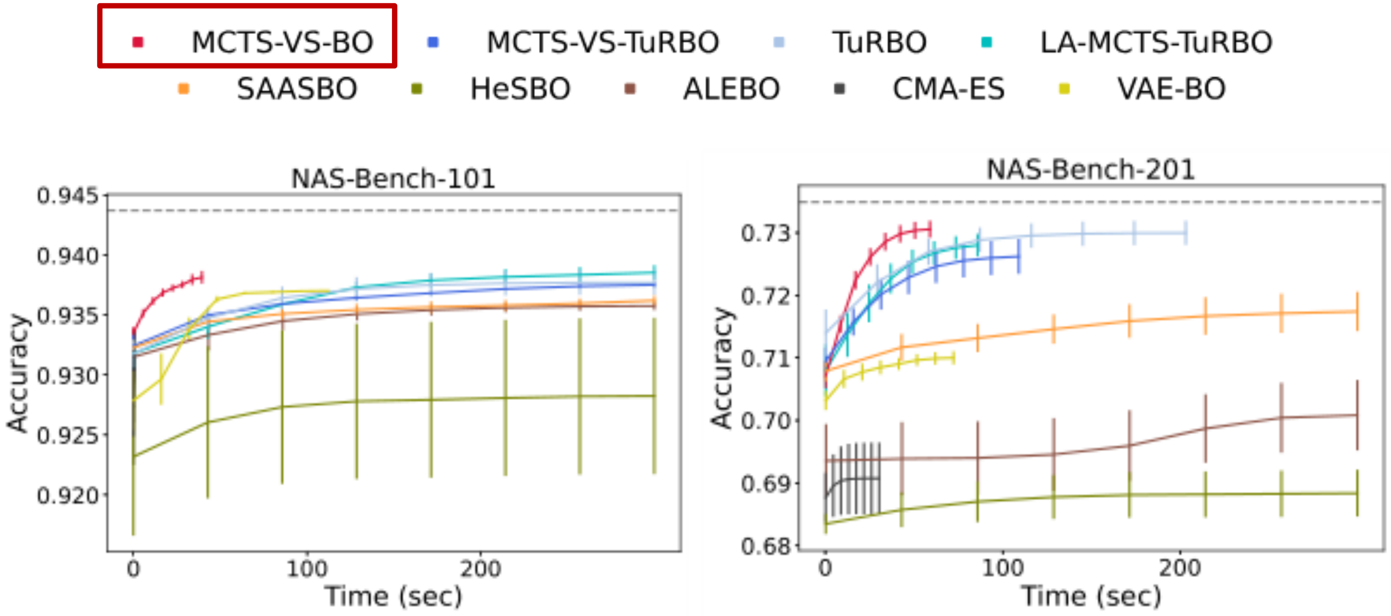
α_i^*: Importance of x_i

If important variables are selected, the cumulative regret can be reduced

MCTS based Variable Selection

Experiments by combining with Bayesian optimization

Application to neural architecture search



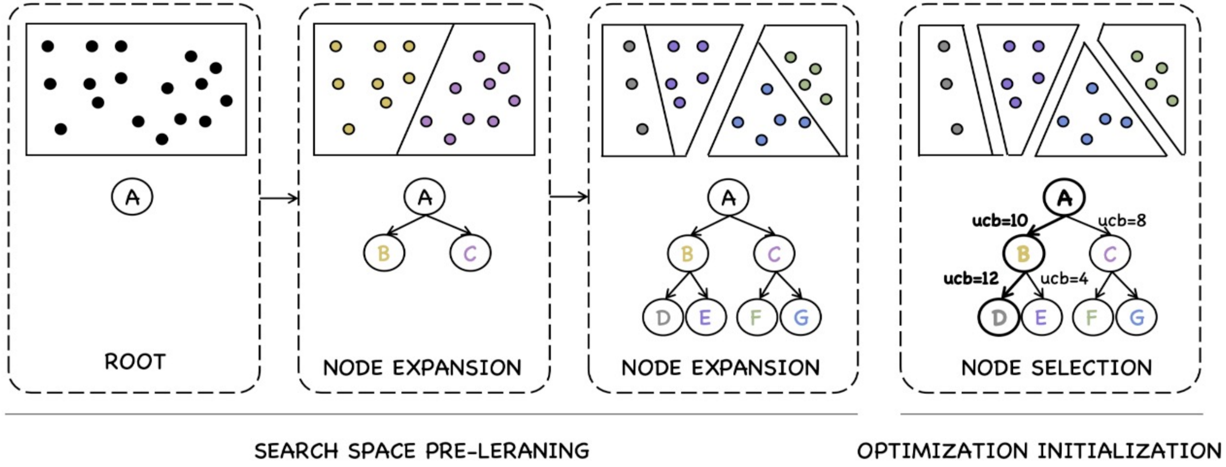
Compared to state-of-the-art methods, MCTS-VS reduces runtime significantly

[Song, Xue, Huang, and Qian, NeurIPS 2022 Spotlight]

MCTS based Space Transfer

Leverage MCTS to divide search spaces

- Use K-Means to divide the samples in a node into two clusters
- Use a binary classifier to separate the two clusters and divide the space into two nodes
- The left and right child nodes have higher and lower potential, respectively



Potential of node m: $$p_m = \boxed{\gamma^{t-1}\frac{\sum_{i\le K} w_i \bar{y}_{i,m}}{\sum_{i\le K} w_i}} + \bar{y}_{T,m}$$ Utilize data of source tasks

$\bar{y}_{i,m}$: average objective values of the samples of the i-th source task in node m

$\bar{y}_{T,m}$: average objective values of the samples of the target task in node m

[Wang, Xue, Song, Huang, and Qian, NeurIPS 2024 Spotlight]

MCTS based Space Transfer

Potential of node m: $p_m = \boxed{\gamma^{t-1} \frac{\sum_{i \leq K} w_i \bar{y}_{i,m}}{\sum_{i \leq K} w_i}} + \bar{y}_{T,m}$ Utilize data of source tasks

$\bar{y}_{i,m}$: average objective values of the samples of the i-th source task in node m

$\bar{y}_{T,m}$: average objective values of the samples of the target task in node m

w_i: weight of the i-th source task

$\mathrm{Distance}(D_i, D_T), 1 \leq i \leq K$

$\downarrow$

$\mathrm{Ranking}\ r_i, 1 \leq i \leq K$

$\downarrow$

$$w_i = \begin{cases} 1.0 - \frac{r_i}{\alpha N_m} & \text{if } r_i < \alpha N_m \\ 0.1 & \text{otherwise} \end{cases}$$

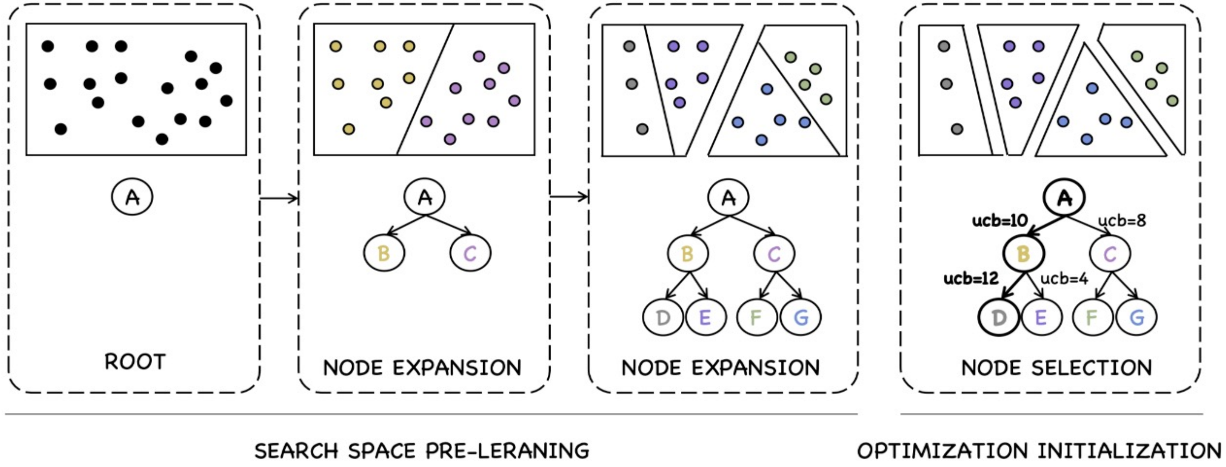
Reflect the similarity between the i-th source task and the target task

γ: decay factor The influence of source tasks decays during the optimization process

MCTS based Space Transfer

Leverage MCTS to divide search spaces

- Use K-Means to divide the samples in a node into two clusters
- Use a binary classifier to separate the two clusters and divide the space into two nodes
- The left and right child nodes have higher and lower potential, respectively

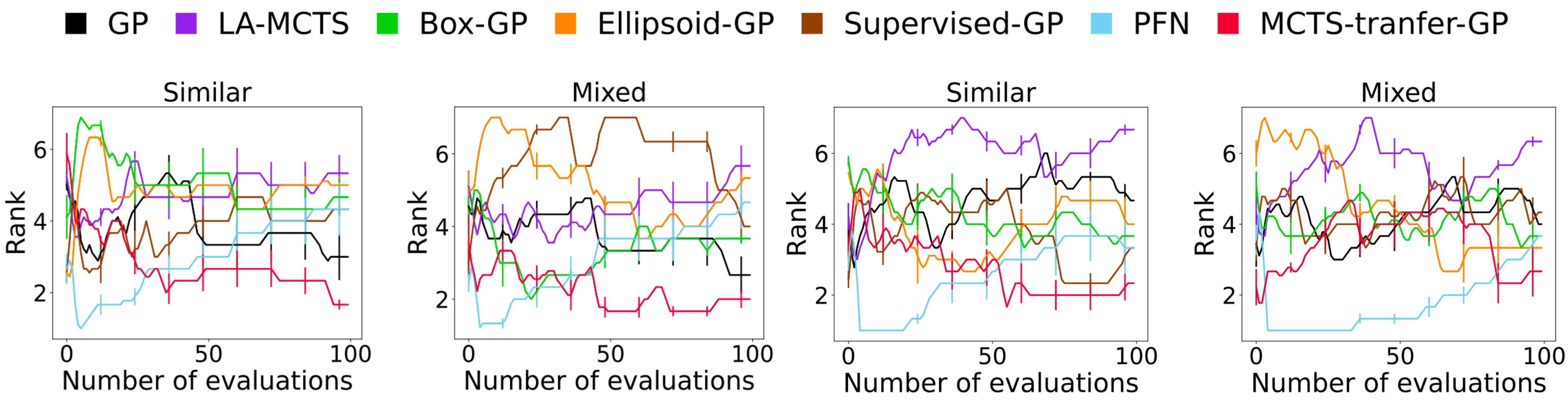


Potential of node m: $p_m = \boxed{\gamma^{t-1}\frac{\sum_{i\leq K} w_i \overline{y}_{i,m}}{\sum_{i\leq K} w_i}} + \overline{y}_{T,m}$ Utilize data of source tasks

Node selection: $\mathrm{ucb}_m = \frac{p_m}{n_m} + 2C_p\sqrt{2\log(n_p)/n_m}$ Select the node m with higher UCB from ROOT

[Wang, Xue, Song, Huang, and Qian, NeurIPS 2024 Spotlight]

MCTS based Space Transfer



(a) Design-Bench

(b) Real-world problems

Similar/Mixed transfer: learning from the data of similar (or similar and dissimilar) tasks

MCTS-transfer achieves the best average rank in high-dimensional real-world problems

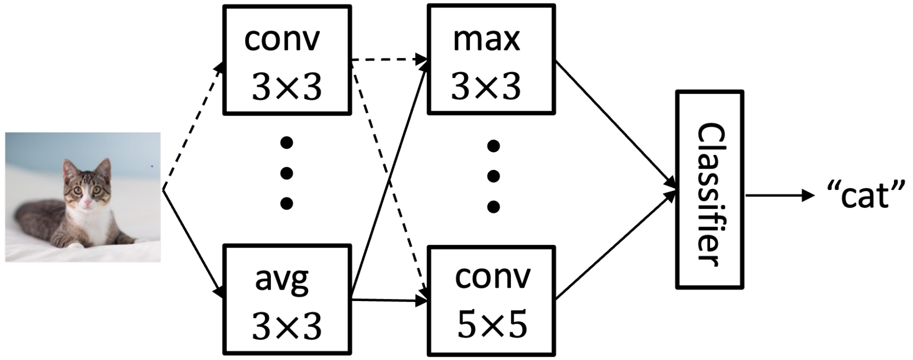
[Wang, Xue, Song, Huang, and Qian, NeurIPS 2024 Spotlight]

Expensive Black-box Optimization

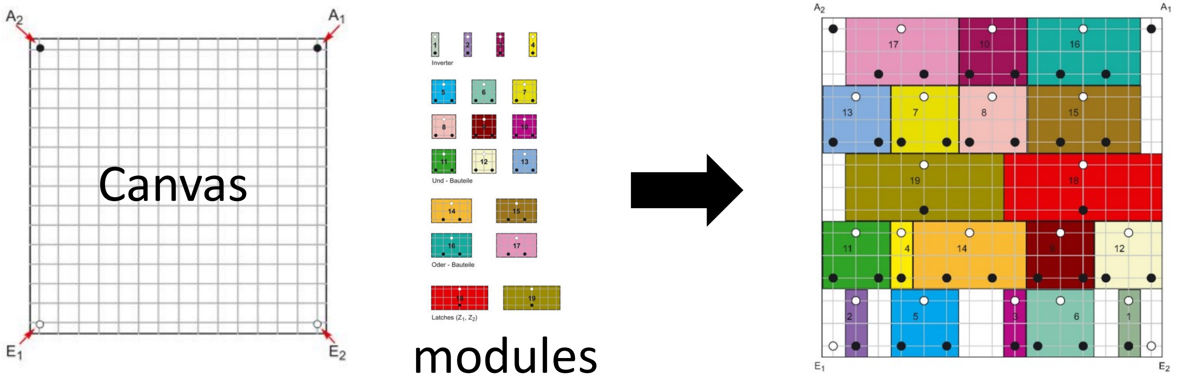
The black-box optimization problems can be expensive

Only a very limited number of evaluations (e.g., 10) are allowed

Neural architecture search



Macro placement



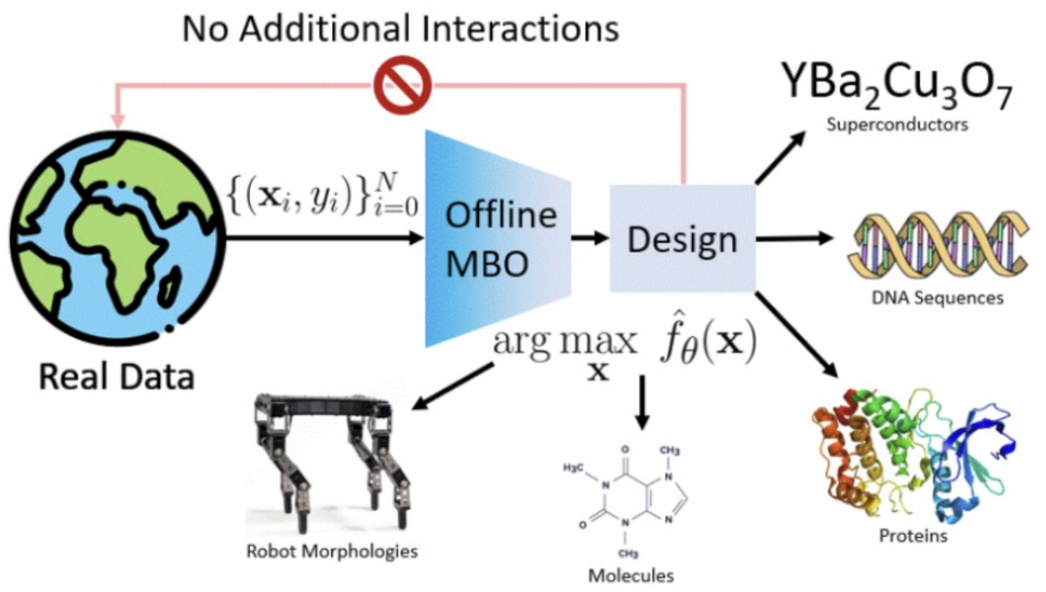
Objective evaluation requires neural network training and testing, which may cost at least several hours

Objective evaluation requires routing and simulation, which are time-consuming

How to develop efficient EL for expensive black-box optimization?

Offline Optimization

Offline optimization: Generate good solutions only using a given static data set



[Trabucco et al., ICML'22]

Current approaches:

- Forward approach ($x \rightarrow y$, surrogate model)

  [Chen et al., NeurIPS'22; Kim et al., NeurIPS'23]

- Backward approach ($y \rightarrow x$, generative model)

  [Kumar & Levine, NeurIPS'20; Krishnamoorthy et al., ICML'23]

No iterative online evaluation!

Offline Optimization

Forward approaches (mainstream):

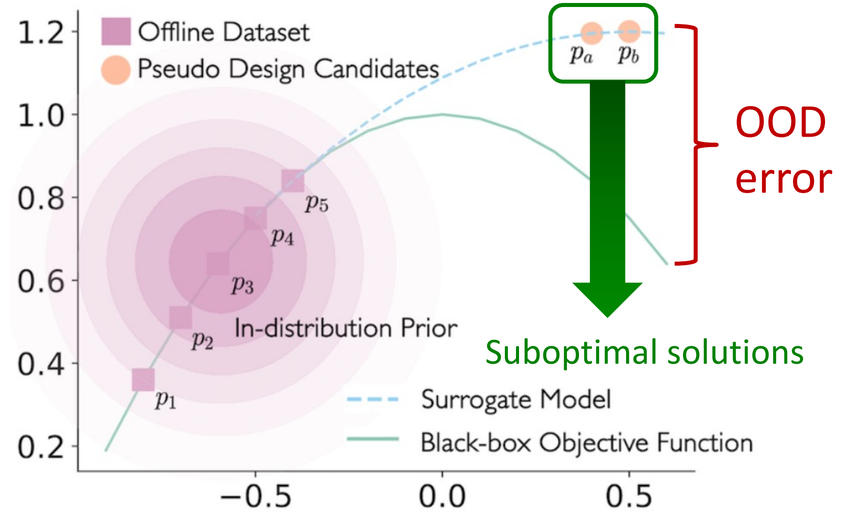
- Train a surrogate model $\hat{f}_\theta$ to predict the function values via regression:

$$\theta^* = \text{argmin}_\theta \frac{1}{N}\sum_{i=1}^{N}\left(\hat{f}_\theta(x_i) - y_i\right)^2$$

- Obtain the final solution that maximizes the model output via gradient ascent:

$$x_{t+1} = x_t + \eta \nabla_x \hat{f}_\theta(x)|_{x=x_t}$$

OOD issue



[Yuan et al., arXiv'24]

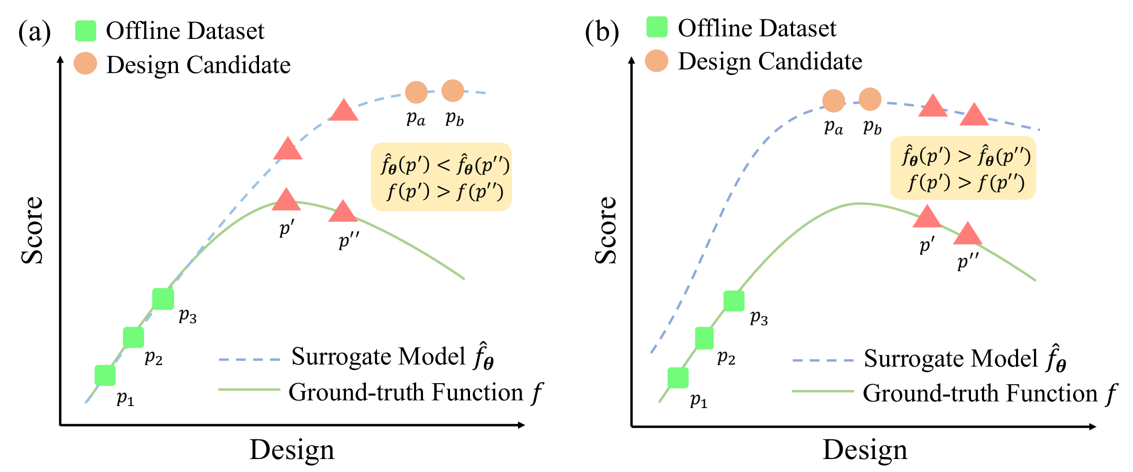
Prior works try to eliminate OOD prediction error via regularization or ensemble learning

[Trabucco et al., ICML'21; Chen et al., NeurIPS'22; Yuan et al., NeurIPS'23; Dao et al., ICML'24]

Offline Optimization by Learning to Rank

Primary goal of offline optimization:

- to select promising designs, rather than to predict their scores precisely



OOD error in order-preserving is more important than OOD prediction error

We propose a novel framework for offline optimization based on learning to rank

- Utilize data augmentation to construct training data for the ranking framework
- Train the surrogate model with ranking loss
- Search for the final solutions using gradient ascent under output adaptation

Offline Optimization by Learning to Rank

Theorem 1 (Equivalence of Optima for Order-Preserving Surrogates). *Let $\hat{f}_{\boldsymbol{\theta}}$ be a surrogate model and f the ground-truth function. A function $h : \mathbb{R} \to \mathbb{R}$ is order-preserving, if $\forall y_1, y_2 \in \mathbb{R}$, $y_1 < y_2$ iff $h(y_1) < h(y_2)$. If there exists an order-preserving h such that $\hat{f}_{\boldsymbol{\theta}}(\mathbf{x}) = h(f(\mathbf{x}))\ \forall \mathbf{x}$, then finding the maximum of f is equivalent to finding that of $\hat{f}_{\boldsymbol{\theta}}$, i.e.,* $\arg\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) = \arg\max_{\mathbf{x}\in\mathcal{X}} \hat{f}_{\boldsymbol{\theta}}(\mathbf{x})$.

- Identify the importance of the order-preserving surrogate models for offline optimization

Theorem 2 (Generalization Error Bound for LTR (Lan et al., 2009)). *Let ϕ be an increasing and strictly positive transformation function (e.g., $\phi(z) = \exp(z)$). Assume that: 1) $\forall \mathbf{x} \in \mathcal{X}$, $\|\mathbf{x}\| \le M$; 2) the ranking model f to be learned is from the linear function class $\mathcal{F} = \{\mathbf{x} \to \mathbf{w}^\top \mathbf{x} \mid \|\mathbf{w}\| \le B\}$. Then with probability $1-\delta$, the following inequality holds:* $\mathcal{O}(1/\sqrt{n})$

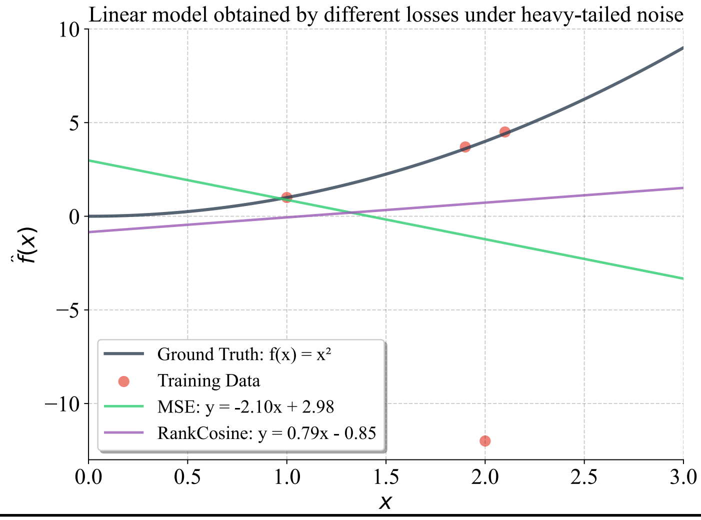
$$\sup_{f\in\mathcal{F}} \left(R_{l_\mathcal{A}}(f) - \hat{R}_{l_\mathcal{A}}(f;\mathcal{D}_R)\right) \le \boxed{4BM \cdot C_\mathcal{A}(\phi)N(\phi)/\sqrt{n}} + \sqrt{2\ln(2/\delta)/n},$$

where: 1) $\mathcal{A}$ stands for a specific LTR algorithm; 2) $N(\phi) = \sup_{z\in[-BM,BM]} \phi'(z)$, which is an algorithm-independent factor measuring the smoothness of ϕ; 3) $C_\mathcal{A}(\phi)$ is an algorithm-dependent factor, e.g., $C_{RankCosine}(\phi) = \sqrt{m}/(2\phi(-BM))$.

- The i.i.d. generalization error bound has a convergence rate of $\mathcal{O}(1/\sqrt{n})$ where n is the number of training data

A special case where pairwise ranking loss is more robust than MSE in OOD regions:

- Assume the training data suffers from heavy-tailed noise
- Assume the model to be learned is a linear model



[Tan, Xue, Lyu, Shang, Wang, Wang, Fu, and Qian, ICLR 2025]

Offline Optimization by Learning to Rank

Experiment on Design-Bench [Trabucco et al., ICML'22] :

Method	Ant	D'Kitty	Superconductor	TF-Bind-8	TF-Bind-10	Mean Rank
$\mathcal{D}$(best)	0.565	0.884	0.400	0.439	0.467	/
BO-qEI	0.812 ± 0.000	0.896 ± 0.000	0.382 ± 0.013	0.802 ± 0.081	0.628 ± 0.036	18.0 / 22
CMA-ES	**1.712 ± 0.754**	0.725 ± 0.002	0.463 ± 0.042	0.944 ± 0.017	0.641 ± 0.036	11.4 / 22
REINFORCE	0.248 ± 0.039	0.541 ± 0.196	0.478 ± 0.017	0.935 ± 0.049	**0.673 ± 0.074**	14.0 / 22
Grad. Ascent	0.273 ± 0.023	0.853 ± 0.018	0.510 ± 0.028	0.969 ± 0.021	0.646 ± 0.037	11.6 / 22
Grad. Ascent Mean	0.306 ± 0.053	0.875 ± 0.024	0.508 ± 0.019	**0.985 ± 0.008**	0.633 ± 0.030	11.2 / 22
Grad. Ascent Min	0.282 ± 0.033	0.884 ± 0.018	**0.514 ± 0.020**	0.979 ± 0.014	0.632 ± 0.027	11.5 / 22
CbAS	0.846 ± 0.032	0.896 ± 0.009	0.421 ± 0.049	0.921 ± 0.046	0.630 ± 0.039	15.5 / 22
MINs	0.906 ± 0.024	0.939 ± 0.007	0.464 ± 0.023	0.910 ± 0.051	0.633 ± 0.034	13.0 / 22
DDOM	0.908 ± 0.024	0.930 ± 0.005	0.452 ± 0.028	0.913 ± 0.047	0.616 ± 0.018	14.6 / 22
BONET	0.921 ± 0.031	0.949 ± 0.016	0.390 ± 0.022	0.798 ± 0.123	0.575 ± 0.039	15.1 / 22
GTG	0.855 ± 0.044	0.942 ± 0.017	0.480 ± 0.055	0.910 ± 0.040	0.619 ± 0.029	13.9 / 22
COMs	0.916 ± 0.026	0.949 ± 0.016	0.460 ± 0.040	0.953 ± 0.038	0.644 ± 0.052	9.5 / 22
RoMA	0.430 ± 0.048	0.767 ± 0.031	0.494 ± 0.025	0.665 ± 0.000	0.553 ± 0.000	18.3 / 22
IOM	0.889 ± 0.034	0.928 ± 0.008	0.491 ± 0.034	0.925 ± 0.054	0.628 ± 0.036	13.1 / 22
BDI	**0.963 ± 0.000**	0.941 ± 0.000	0.508 ± 0.013	0.973 ± 0.000	0.658 ± 0.000	5.9 / 22
ICT	0.915 ± 0.024	0.947 ± 0.009	0.494 ± 0.026	0.897 ± 0.050	0.659 ± 0.024	9.4 / 22
Tri-Mentoring	0.891 ± 0.011	0.947 ± 0.005	0.503 ± 0.013	0.956 ± 0.000	0.662 ± 0.012	7.7 / 22
PGS	0.715 ± 0.046	**0.954 ± 0.022**	0.444 ± 0.020	0.889 ± 0.061	0.634 ± 0.040	13.2 / 22
FGM	0.923 ± 0.023	0.944 ± 0.014	0.481 ± 0.024	0.811 ± 0.079	0.611 ± 0.008	13.2 / 22
Match-OPT	0.933 ± 0.016	0.952 ± 0.008	0.504 ± 0.021	0.824 ± 0.067	0.655 ± 0.050	8.0 / 22
RaM-RankCosine (Ours)	0.940 ± 0.028	0.951 ± 0.017	**0.514 ± 0.026**	**0.982 ± 0.012**	**0.675 ± 0.049**	**2.7 / 22**
RaM-ListNet (Ours)	0.949 ± 0.025	**0.962 ± 0.015**	**0.517 ± 0.029**	0.981 ± 0.012	0.670 ± 0.035	**2.2 / 22**

The third method: only ranks 5.9 on average

Our method equipped with two ranking losses outperforms other 20 methods with average ranks of 2.7 and 2.2

Offline Optimization by Learning to Rank

Examine the versatility of ranking loss by replacing the MSE term in regression-based methods with the best-performing ranking loss, ListNet

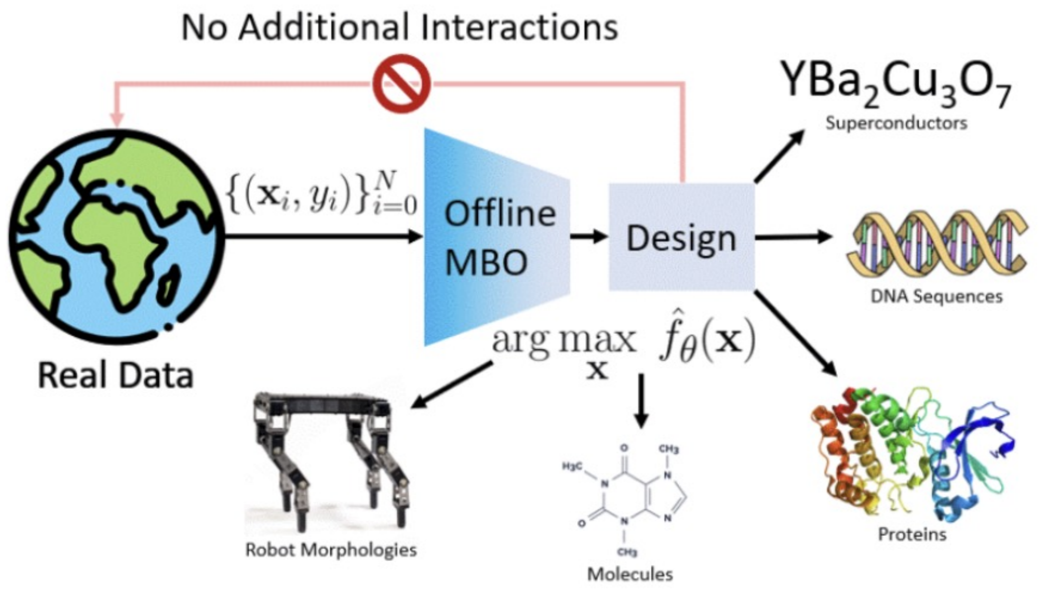
Method	Type	Ant		D'Kitty		Superconductor		TF-Bind-8		TF-Bind-10	
		Score	Gain	Score	Gain	Score	Gain	Score	Gain	Score	Gain
BO-*q*EI	MSE	0.812 ± 0.000		0.896 ± 0.000		0.382 ± 0.013		0.802 ± 0.081		0.628 ± 0.036	
	ListNet	0.812 ± 0.000	+0.0%	0.896 ± 0.000	+0.0%	0.509 ± 0.013	+33.2%	0.912 ± 0.032	+13.7%	0.653 ± 0.056	+4.0%
CMA-ES	MSE	1.712 ± 0.705		0.722 ± 0.001		0.463 ± 0.042		0.944 ± 0.017		0.641 ± 0.036	
	ListNet	1.923 ± 0.773	+12.3%	0.723 ± 0.002	+0.1%	0.486 ± 0.020	+5.0%	0.960 ± 0.008	+1.7%	0.661 ± 0.044	+3.1%
REINFORCE	MSE	0.248 ± 0.039		0.344 ± 0.091		0.478 ± 0.017		0.935 ± 0.049		0.673 ± 0.074	
	ListNet	0.318 ± 0.056	+28.2%	0.359 ± 0.139	+4.3%	0.501 ± 0.013	+4.8%	0.935 ± 0.049	+0.0%	0.673 ± 0.074	+0.0%
Grad. Ascent	MSE	0.273 ± 0.022		0.853 ± 0.017		0.510 ± 0.028		0.969 ± 0.020		0.646 ± 0.037	
	ListNet	0.280 ± 0.021	+2.6%	0.890 ± 0.019	+4.3%	0.521 ± 0.012	+2.0%	0.985 ± 0.011	+1.7%	0.660 ± 0.049	+2.2%
CbAS	MSE	0.846 ± 0.030		0.896 ± 0.009		0.421 ± 0.046		0.921 ± 0.046		0.630 ± 0.039	
	ListNet	0.854 ± 0.037	+0.9%	0.898 ± 0.009	+0.2%	0.425 ± 0.036	+1.0%	0.956 ± 0.033	+3.8%	0.642 ± 0.034	+1.9%
MINs	MSE	0.906 ± 0.024		0.939 ± 0.007		0.464 ± 0.023		0.910 ± 0.051		0.633 ± 0.032	
	ListNet	0.911 ± 0.025	+0.5%	0.941 ± 0.009	+0.2%	0.477 ± 0.019	+2.8%	0.910 ± 0.029	+0.0%	0.638 ± 0.037	+0.8%
Tri-Mentoring	MSE	0.891 ± 0.011		0.947 ± 0.005		0.503 ± 0.013		0.956 ± 0.000		0.662 ± 0.012	
	ListNet	0.915 ± 0.024	+2.7%	0.943 ± 0.004	-0.4%	0.503 ± 0.010	+0.0%	0.971 ± 0.005	+1.7%	0.710 ± 0.020	+7.3%
PGS	MSE	0.715 ± 0.046		0.954 ± 0.022		0.444 ± 0.020		0.889 ± 0.061		0.634 ± 0.040	
	ListNet	0.723 ± 0.032	+1.1%	0.962 ± 0.018	+0.8%	0.452 ± 0.042	+1.8%	0.886 ± 0.003	-0.3%	0.643 ± 0.030	+1.4%
Match-OPT	MSE	0.933 ± 0.016		0.952 ± 0.008		0.504 ± 0.021		0.824 ± 0.067		0.655 ± 0.050	
	ListNet	0.936 ± 0.027	+0.3%	0.956 ± 0.018	+0.4%	0.513 ± 0.011	+1.8%	0.829 ± 0.009	+0.6%	0.659 ± 0.037	+0.6%

The gains are always positive except two cases, clearly demonstrating the versatility of ranking loss

[Tan, Xue, Lyu, Shang, Wang, Wang, Fu, and Qian, ICLR 2025]

Offline Optimization

Offline optimization: Generate good solutions only using a given static data set



[Trabucco et al., ICML'22]

Current approaches:

- Forward approach ($x \rightarrow y$, surrogate model)

  [Chen et al., NeurIPS'22; Kim et al., NeurIPS'23]

- Backward approach ($y \rightarrow x$, generative model)

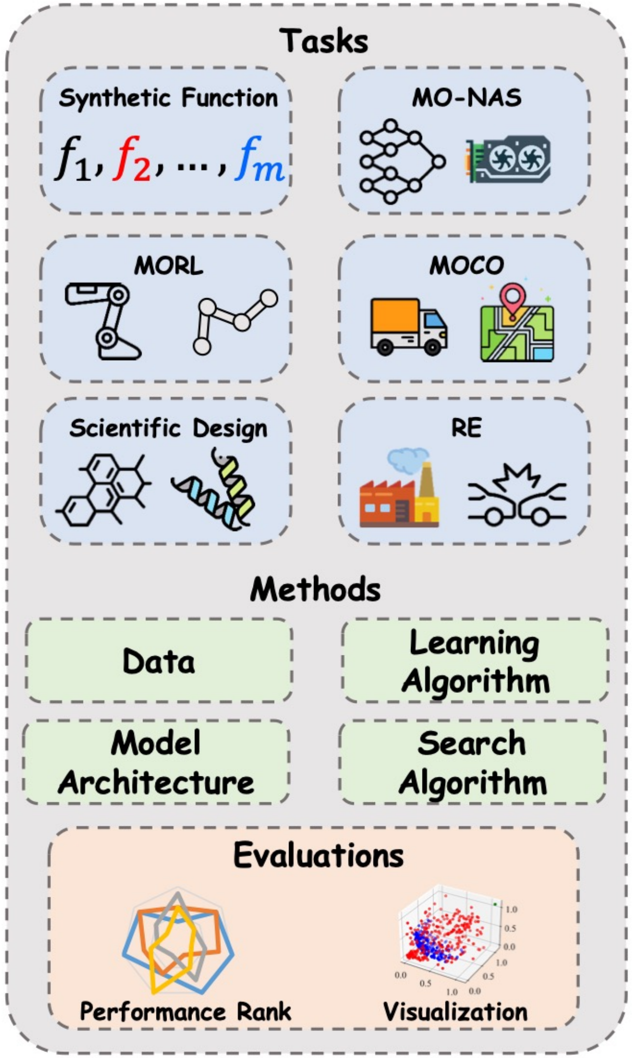
  [Kumar & Levine, NeurIPS'20; Krishnamoorthy et al., ICML'23]

No iterative online evaluation!

However, current approaches only consider single-objective scenario, while many real-world applications have **multiple objectives**

Offline Multi-objective Optimization

Propose offline multi-objective optimization for the first time



Various benchmark tasks

Task Name	Dataset size	Dimensions	# Objectives	Search space
Synthetic Function	60000	2-30	2-3	Continuous
MO-NAS	9735	6	3	Categorical
MO-Swimmer	8571	9734	2	Continuous
MO-Hopper	4500	10184	2	Continuous
MO-TSP	60000	500	2	Permutation
MO-CVRP	60000	100	2	Permutation
MO-KP	60000	200	2	Permutation
Molecule	49001	32	3	Continuous
Regex	42048	4	2	Sequence
RFP	4937	4	2	Sequence
Real-world Application	60000	3-6	2-6	Continuous & Mixed

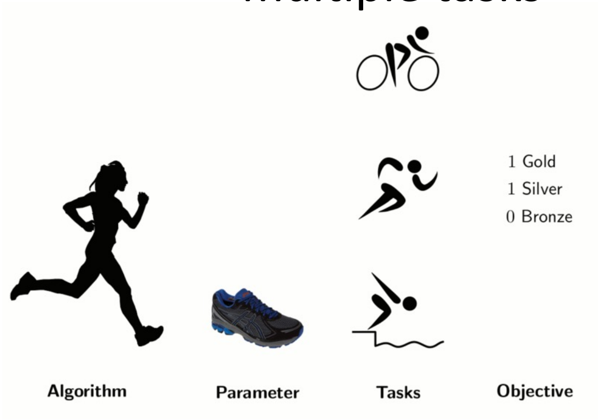
Extensive analysis

Methods	Synthetic	MO-NAS	MORL	MOCO	Sci-Design	RE	Average Rank
$\mathcal{D}$(best)	12.17 ± 0.27	12.11 ± 0.05	9.00 ± 0.50	**2.00 ± 0.14**	8.38 ± 0.38	13.13 ± 0.07	10.03 ± 0.07
End-to-End	6.91 ± 0.03	8.37 ± 0.05	7.50 ± 2.00	6.75 ± 0.46	6.75 ± 1.12	7.50 ± 0.57	7.32 ± 0.01
End-to-End + GradNorm	8.25 ± 0.56	7.71 ± 0.08	4.50 ± 1.00	7.61 ± 0.18	8.62 ± 0.50	10.53 ± 0.07	8.34 ± 0.01
End-to-End + PcGrad	7.88 ± 0.06	7.18 ± 0.39	10.50 ± 1.50	6.07 ± 0.64	8.69 ± 2.69	8.23 ± 0.17	7.51 ± 0.14
Multi-Head	6.38 ± 0.50	5.37 ± 0.37	6.25 ± 2.25	8.29 ± 0.21	9.19 ± 0.44	8.33 ± 0.40	7.00 ± 0.38
Multi-Head + GradNorm	7.78 ± 0.53	10.20 ± 0.04	11.00 ± 3.00	9.98 ± 0.30	9.06 ± 1.19	10.63 ± 0.17	9.63 ± 0.04
Multi-Head + PcGrad	8.61 ± 0.14	6.92 ± 0.55	10.50 ± 3.50	8.21 ± 0.36	9.38 ± 0.50	8.50 ± 0.17	8.09 ± 0.20
Multiple Models	**4.05 ± 0.11**	4.93 ± 0.28	9.75 ± 0.75	6.34 ± 0.27	5.62 ± 0.75	4.50 ± 0.10	5.02 ± 0.03
Multiple Models + COMs	9.81 ± 0.31	5.92 ± 0.34	7.00 ± 2.00	6.36 ± 0.50	8.38 ± 2.00	10.50 ± 0.50	8.09 ± 0.32
Multiple Models + RoMA	8.95 ± 0.05	5.00 ± 0.00	4.75 ± 2.25	8.14 ± 0.21	8.00 ± 1.38	6.30 ± 0.10	7.07 ± 0.02
Multiple Models + IOM	6.11 ± 0.36	**4.34 ± 0.34**	**3.75 ± 2.75**	4.25 ± 0.04	7.19 ± 0.44	**3.23 ± 0.03**	**4.61 ± 0.05**
Multiple Models + ICT	9.11 ± 0.27	11.92 ± 0.29	4.75 ± 0.25	9.89 ± 0.46	8.62 ± 0.75	8.43 ± 0.30	9.64 ± 0.11
Multiple Models + Tri-Mentoring	7.83 ± 0.05	11.37 ± 0.47	5.25 ± 2.75	9.50 ± 0.00	9.38 ± 1.00	6.73 ± 0.20	8.77 ± 0.21
MOBO	9.09 ± 0.47	7.18 ± 0.55	10.50 ± 0.00	13.69 ± 0.08	**5.44 ± 0.56**	6.11 ± 0.29	8.64 ± 0.37
MOBO-ParEGO	10.27 ± 0.23	11.47 ± 0.32	N/A	13.62 ± 0.04	9.44 ± 0.44	12.71 ± 0.33	11.68 ± 0.20
MOBO-JES	12.48 ± 0.05	16.00 ± 0.00	N/A	3.00 ± 0.00	7.50 ± 6.50	8.04 ± 0.37	10.30 ± 0.44

How to Configure Algorithms

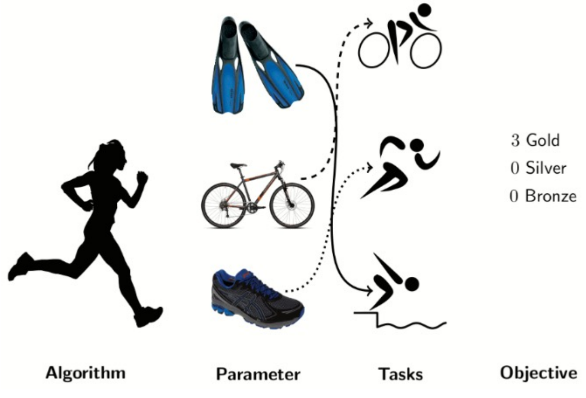
Evolutionary algorithms often have **multiple heterogeneous hyper-parameters**, whose configuration can influence the performance largely

multiple tasks



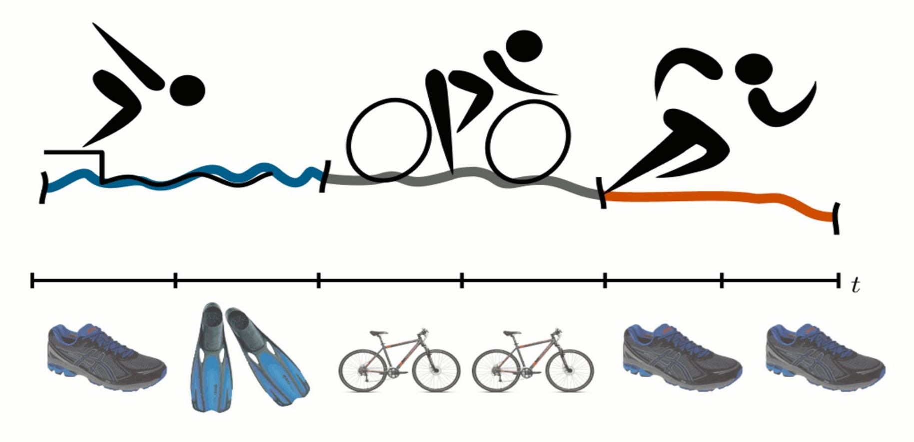
Algorithm Configuration (AC)

multiple tasks



Per-instance AC

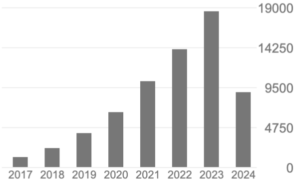
a single task



Dynamic AC (DAC)

Can we adjust multiple hyper-parameters of EAs automatically and dynamically?

Dynamic Algorithm Configuration



Automated Dynamic Algorithm Configuration

Steven Adriaensen ADRIAENS@CS.UNI-FREIBURG.DE
André Biedenkapp BIEDENKA@CS.UNI-FREIBURG.DE
Gresa Shala SHALAG@CS.UNI-FREIBURG.DE
Noor Awad AWAD@CS.UNI-FREIBURG.DE
University of Freiburg, Machine Learning Lab

Theresa Eimer EIMER@TNT.UNI-HANNOVER.DE
Marius Lindauer LINDAUER@TNT.UNI-HANNOVER.DE
Leibniz University Hannover, Institute for Information Processing

Frank Hutter FH@CS.UNI-FREIBURG.DE
University of Freiburg, Machine Learning Lab & Bosch Center for Artificial Intelligence

7.2 Limitations and Further Research

While these case studies and other previous applications provide a "proof of concept" for automated DAC, we point out that much remains to be done to unlock its full potential, and we hope that this work may serve as a stepping stone for further exploring this promising line of research. In what remains, we will discuss some of the limitations of contemporary work and provide specific directions for future research.

Jointly configuring many parameters: While static approaches are capable of jointly configuring hundreds of parameters, the configuration space in contemporary DAC is typically much smaller, often considering only a single parameter. While the configuration space is smaller, the candidate solution space (i.e., the dynamic configuration policy space) grows exponentially with the number of reconfiguration points, in the worst case, and is thus typically drastically larger than static configuration policy spaces. Although modern techniques from reinforcement learning scale much better than ever before, we still know too little about the internal structure of DAC problems to handle this exploding space of possible policies. For example, not much is known regarding interaction effects of parameters in the DAC setting. If there should be only a few interaction effects between parameters as in static AC (Hutter et al., 2014; Wang et al., 2016), learning several independent policies might be a way forward.

Open problem: How to adjust multiple heterogeneous hyper-parameters simultaneously?

Dynamic Algorithm Configuration

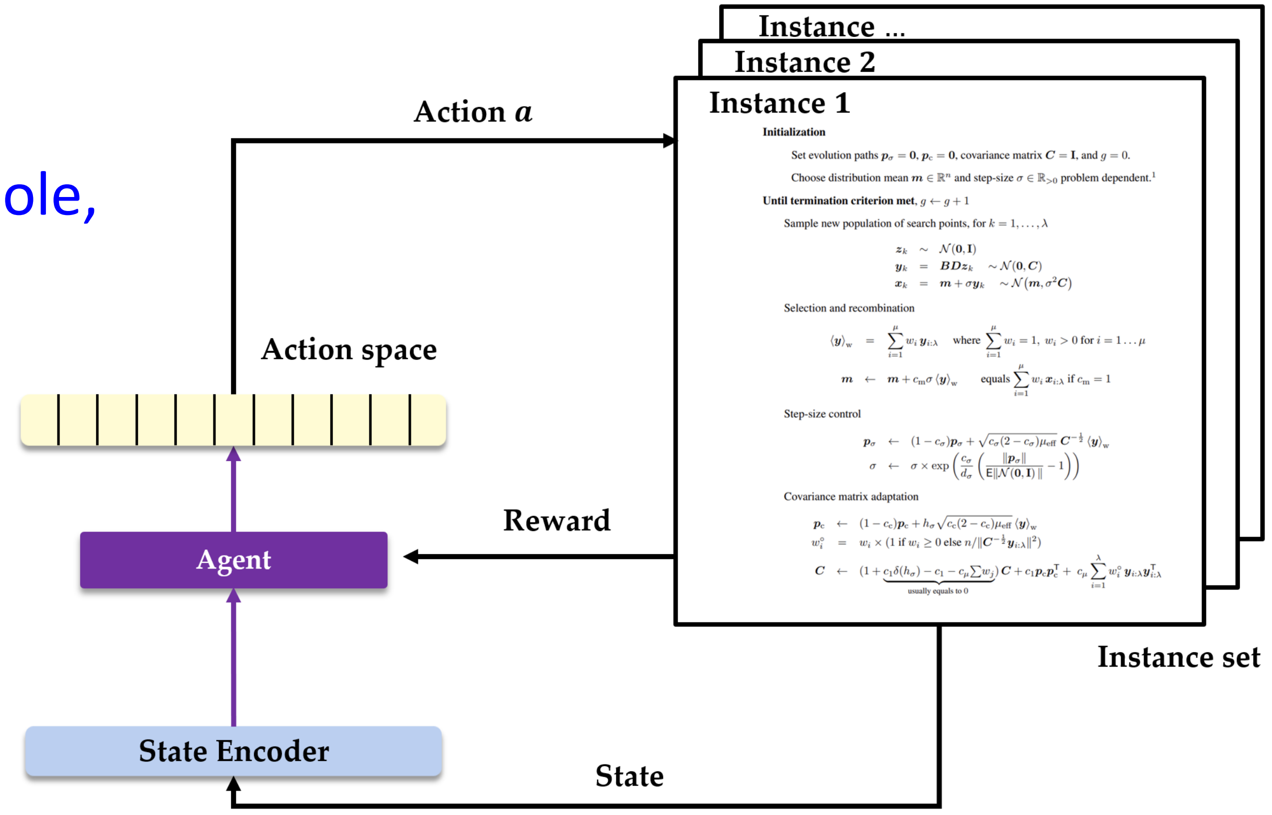
Open problem: How to adjust multiple heterogeneous hyper-parameters simultaneously?

One direct method:

Treat multiple hyper-parameters as a whole, and apply RL methods

Very difficult

because of the large action space!

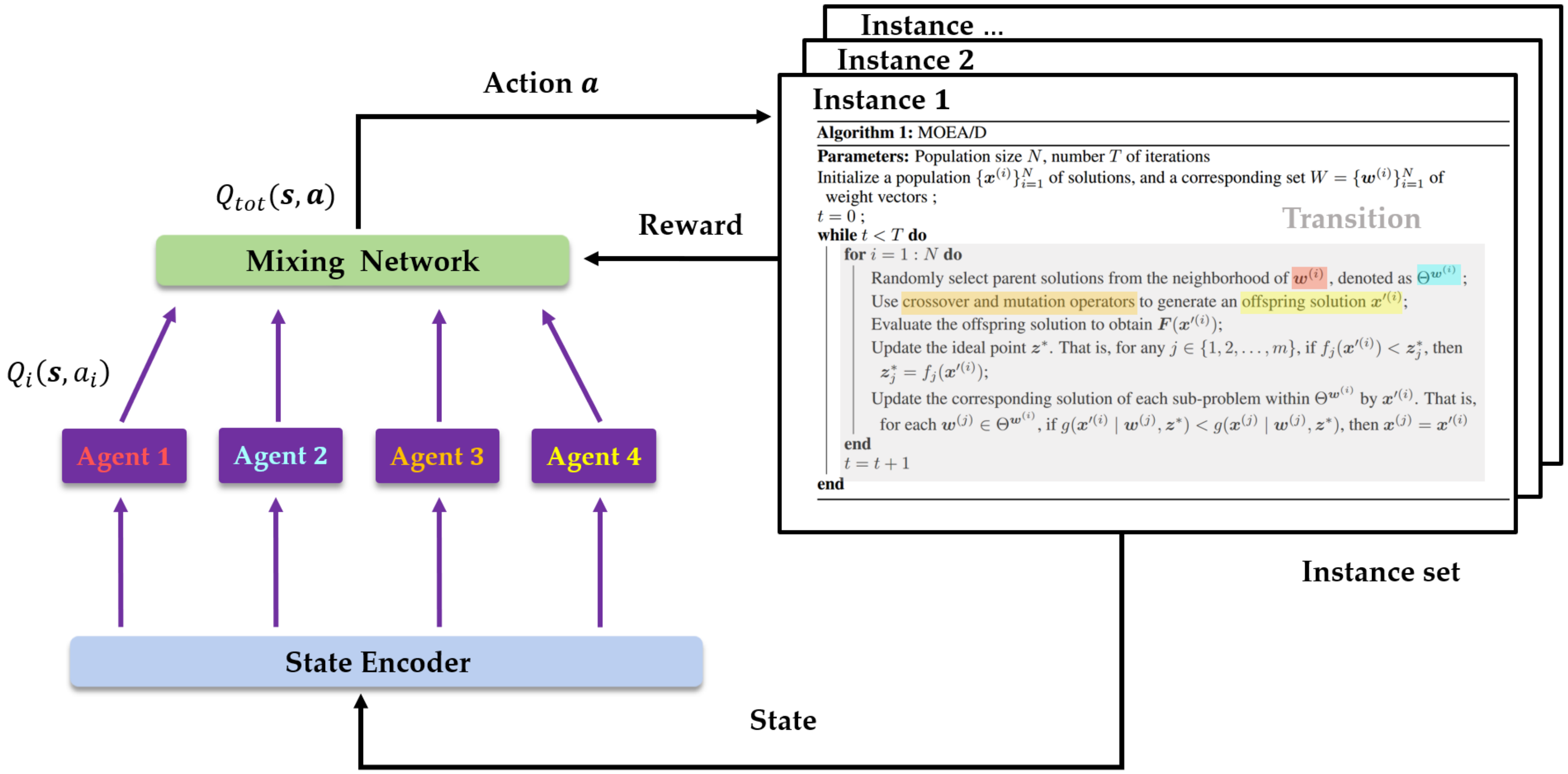


Multi-Agent Dynamic Algorithm Configuration

Open problem: How to adjust multiple heterogeneous hyper-parameters simultaneously?

Our solution:

- Cooperative multi-agent modeling
- Each agent handles one hyper-parameter



[Xue, Xu, Yuan, Li, Qian, et al. NeurIPS 2022 Spotlight]

Application of Multi-Agent DAC to MOEA/D

Table 2: IGD values obtained by MOEA/D, DQN, MA-UCB and MA-DAC on different problems. Each result consists of the mean and standard deviation of 30 runs. The best mean value on each problem is highlighted in **bold**. The symbols '+', '−' and '≈' indicate that the result is significantly superior to, inferior to, and almost equivalent to MA-DAC, respectively, according to the Wilcoxon rank-sum test with confidence level 0.05.

Problem	M	MOEA/D	DQN	MA-UCB	MA-DAC
DTLZ2	3	4.605E-02 (3.54E-04) −	4.628E-02 (2.96E-04) −	4.671E-02 (3.70E-04) −	**3.807E-02** (5.05E-04)
	5	3.006E-01 (1.55E-03) −	3.016E-01 (1.34E-03) −	3.041E-01 (1.69E-03) −	**2.442E-01** (1.26E-02)
	7	4.455E-01 (1.41E-02) −	4.671E-01 (1.15E-02) −	4.826E-01 (9.59E-03) −	**3.944E-01** (1.17E-02)
WFG4	3	5.761E-02 (5.41E-04) −	6.920E-02 (1.20E-03) −	7.165E-02 (1.83E-03) −	**5.200E-02** (1.19E-03)
	5	3.442E-01 (1.21E-02) −	2.810E-01 (6.86E-03) −	2.859E-01 (6.77E-03) −	**1.868E-01** (2.81E-03)
	7	4.529E-01 (1.79E-02) −	3.725E-01 (1.14E-02) −	3.868E-01 (1.54E-02) −	**3.033E-01** (3.66E-03)
WFG6	3	6.938E-02 (5.50E-03) −	6.834E-02 (1.78E-02) −	6.601E-02 (1.00E-02) −	**4.831E-02** (8.95E-03)
	5	3.518E-01 (2.82E-03) −	3.160E-01 (2.40E-02) −	3.359E-01 (1.47E-02) −	**1.942E-01** (6.90E-03)
	7	4.869E-01 (3.03E-02) −	4.322E-01 (2.95E-02) −	4.389E-01 (3.41E-02) −	**3.112E-01** (4.93E-03)
Train: +/−/≈		0/9/0	0/9/0	0/9/0	
DTLZ4	3	6.231E-02 (8.85E-02) ≈	**5.590E-02** (5.77E-03) −	6.011E-02 (5.08E-03) −	6.700E-02 (6.14E-02)
	5	3.133E-01 (4.45E-02) ≈	3.457E-01 (1.61E-02) −	3.492E-01 (1.69E-02) −	**2.995E-01** (2.10E-02)
	7	4.374E-01 (2.57E-02) −	4.552E-01 (1.47E-02) −	4.756E-01 (2.01E-02) −	**4.182E-01** (1.21E-02)
WFG5	3	6.327E-02 (1.10E-03) −	6.212E-02 (5.54E-04) −	6.118E-02 (7.03E-04) −	**4.730E-02** (7.89E-04)
	5	3.350E-01 (9.77E-03) −	3.077E-01 (6.36E-03) −	3.036E-01 (8.83E-03) −	**1.811E-01** (3.02E-03)
	7	4.101E-01 (2.08E-02) −	4.996E-01 (1.32E-02) −	5.024E-01 (1.38E-02) −	**3.206E-01** (8.04E-03)
WFG7	3	5.811E-02 (6.31E-04) −	5.930E-02 (7.32E-04) −	6.014E-02 (7.11E-04) −	**4.066E-02** (5.31E-04)
	5	3.572E-01 (5.47E-03) −	2.993E-01 (1.43E-02) −	3.207E-01 (1.71E-02) −	**1.858E-01** (2.12E-03)
	7	5.236E-01 (2.19E-02) −	4.576E-01 (2.38E-02) −	4.879E-01 (2.75E-02) −	**3.258E-01** (1.25E-02)
WFG8	3	8.646E-02 (3.44E-03) −	9.280E-02 (1.06E-03) −	9.612E-02 (1.48E-03) −	**7.901E-02** (1.19E-03)
	5	4.258E-01 (8.42E-03) −	3.969E-01 (1.26E-02) −	3.956E-01 (1.32E-02) −	**2.479E-01** (7.20E-03)
	7	5.816E-01 (1.30E-02) −	5.575E-01 (1.39E-02) −	5.642E-01 (1.38E-02) −	**4.127E-01** (5.93E-03)
WFG9	3	5.817E-02 (1.24E-03) −	5.628E-02 (7.29E-04) −	7.953E-02 (2.45E-02) −	**4.159E-02** (6.10E-04)
	5	3.633E-01 (1.20E-02) −	3.258E-01 (1.61E-02) −	3.396E-01 (1.55E-02) −	**1.832E-01** (7.10E-03)
	7	5.538E-01 (2.63E-02) −	5.115E-01 (2.15E-02) −	5.227E-01 (1.79E-02) −	**3.278E-01** (7.21E-03)
Test: +/−/≈		0/13/2	0/15/0	0/15/0	

Task: Adjust four hyper-parameters of MOEA/D (a popular MOEA) dynamically

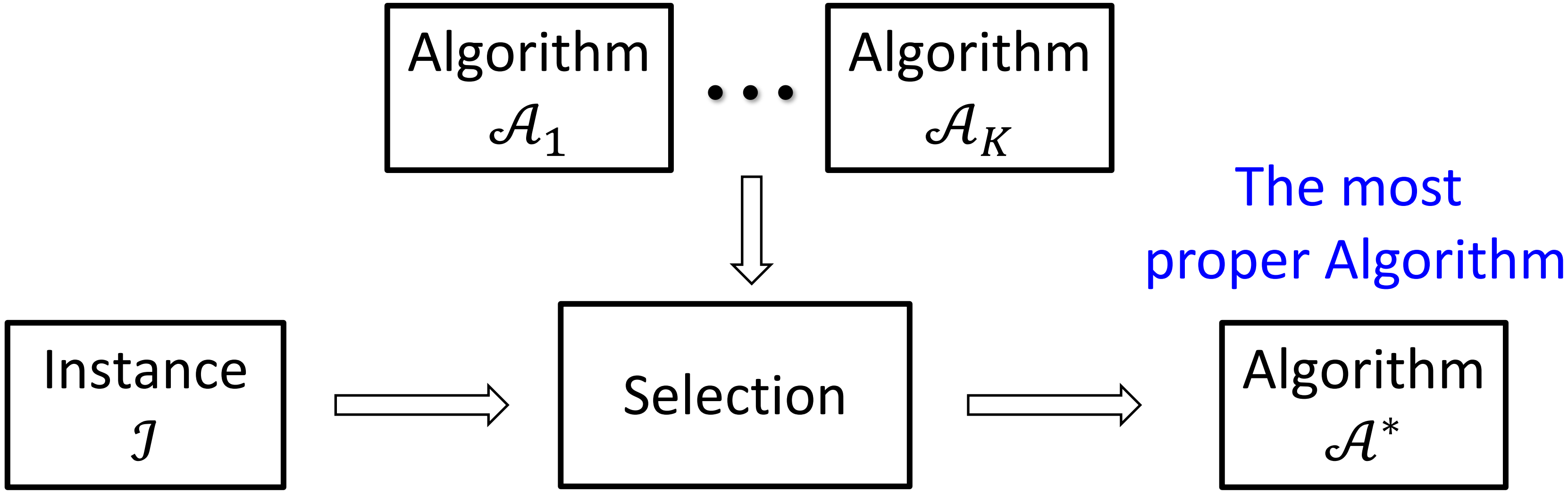
Train on DTLZ2, WFG4, and WFG6 with m objectives, and test on the other problems with m objectives

Significantly better on almost all the 24 problems

Good generalization ability

[Xue, Xu, Yuan, Li, Qian, et al. NeurIPS 2022 Spotlight]

Algorithm Selection

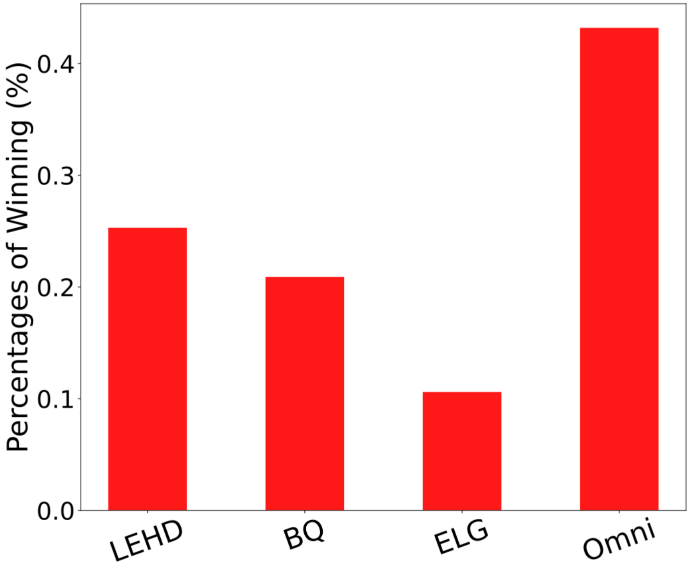


Algorithm selection has been applied in many fields, e.g., machine learning and black-box optimization

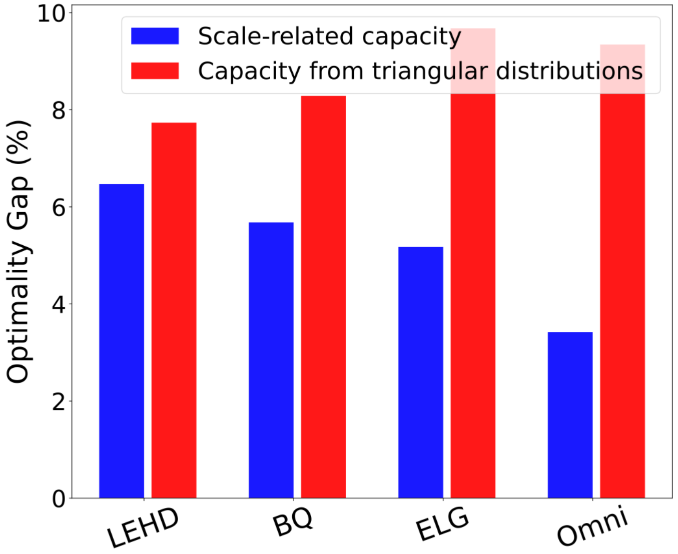
But it's new for neural combinatorial optimization!

Diversity of Neural Combinatorial Optimization Solvers

Different neural solvers demonstrate complementary performance at instance level



(a) Percentages of Winning



(b) Optimality Gap (Average)

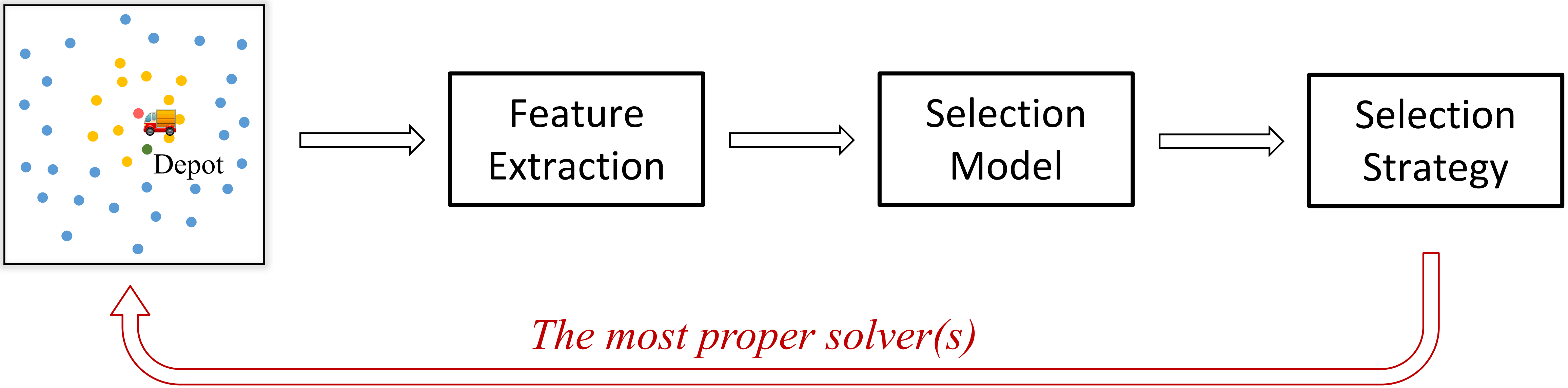
Different solvers win on different instances

The change of problem distribution almost reverses the rank of solvers

[Gao, Shang, Xue, and Qian, ICML 2025]

Neural Solver Selection for Combinatorial Optimization

We introduce the idea of solver selection into the field of NCO for the first time



Application to TSP and CVRP

As an instantiation, we apply neural solver selection to two widely studied problems: Traveling Salesman Problem (TSP) and Capacitated Vehicle Routing Problem (CVRP)

Omni [ICML'23]
BQ [NeurIPS'23]
LEHD [NeurIPS'23]
DIFUSCO [NeurIPS'23]
T2T [NeurIPS'23]
INViT [ICML'24]
ELG [IJCAI'24]

Neural solver pool for TSP

Omni [ICML'23]
BQ [NeurIPS'23]
LEHD [NeurIPS'23]
INViT [ICML'24]
ELG [IJCAI'24]
MVMoE [ICML'24]

Neural solver pool for CVRP

Application to TSP and CVRP

Feature extraction

A hierarchical graph encoder

- Adapt graph pooling to construct multi-level local views of problem instances
- Leverage the similarity of local views to boost OOD generalization

Selection model

A MLP to predict scores of neural solvers, trained by

- Classification: Treat the best solver as the ground truth label
- Ranking: Rank solvers according to their performance metrics

Selection strategies

- Greedy, top-k, rejection-based, and top-p selection

[Gao, Shang, Xue, and Qian, ICML 2025]

Experimental Results

Results on synthetic dataset with Gaussian distribution and $N \in [50,500]$

State-of-the-art individual solvers

Run all the solvers and select the best

Methods	TSP Gap	TSP Time
BQ (*3rd*)	3.00%	1.40s
T2T (*2nd*)	2.40%	1.58s
DIFUSCO (*1st*)	2.33%	1.45s
OPT	1.24%	8.93s
Selection by classification		
Greedy	1.94% (0.02%)	1.36s (0.01s)
Top-k ($k=2$)	1.53% (0.01%)	2.52s (0.04s)
Rejection (20%)	1.81% (0.01%)	1.63s (0.01s)
Top-p ($p=0.5$)	1.84% (0.03%)	1.55s (0.06s)
Selection by ranking		
Greedy	1.86% (0.01%)	1.33s (0.01s)
Top-k ($k=2$)	**1.51% (0.02%)**	2.56s (0.03s)
Rejection (20%)	1.75% (0.02%)	1.63s (0.01s)
Top-p ($p=0.5$)	1.68% (0.02%)	1.86s (0.07s)

Methods	CVRP Gap	CVRP Time
LEHD (*3rd*)	7.37%	1.01s
BQ (*2nd*)	7.20%	1.59s
Omni (*1st*)	6.82%	0.24s
OPT	4.64%	4.38s
Selection by classification		
Greedy	5.35% (0.02%)	0.64s (0.01s)
Top-k ($k=2$)	**4.81% (0.01%)**	1.87s (0.03s)
Rejection (20%)	5.19% (0.03%)	0.77s (0.01s)
Top-p ($p=0.8$)	5.16% (0.03%)	0.87s (0.08s)
Selection by ranking		
Greedy	5.31% (0.01%)	0.62s (0.01s)
Top-k ($k=2$)	4.82% (0.01%)	1.90s (0.04s)
Rejection (20%)	5.15% (0.02%)	0.74s (0.01s)
Top-p ($p=0.8$)	4.99% (0.02%)	1.03s (0.03s)

Our method is close to OPT, but achieves significant speedup

Our method outperforms the best individual solvers on TSP and CVRP, and can even consume less time on TSP

[Gao, Shang, Xue, and Qian, ICML 2025]

Experimental Results

Generalization results on out-of-distribution datasets

- Training: Synthetic dataset with Gaussian distribution and $N \in [50,500]$
- Test: **TSPLIB/CVRPLIB** with more complex distributions and $N \in [50,1002]$

State-of-the-art single solvers

Run all the solvers and select the best

Methods	TSPLIB Gap	TSPLIB Time	Methods	CVRPLIB Set-X Gap	CVRPLIB Set-X Time
BQ (*3rd*)	3.04%	1.44s	BQ (*3rd*)	10.31%	2.60s
DISFUCO (*2nd*)	2.13%	1.44s	Omni (*2nd*)	6.21%	0.38s
T2T (*1st*)	1.95%	1.74s	ELG (*1st*)	6.10%	1.31s
OPT	0.89%	9.14s	OPT	5.10%	6.81s
Selection by classification			*Selection by classification*		
Greedy	1.54% (0.05%)	1.33s (0.02s)	Greedy	5.96% (0.12%)	1.06s (0.08s)
Top-k ($k=2$)	1.22% (0.10%)	2.47s (0.02s)	Top-k ($k=2$)	5.44% (0.08%)	2.40s (0.25s)
Rejection (20%)	1.42% (0.11%)	1.54s (0.03s)	Rejection (20%)	5.83% (0.12%)	1.31s (0.09s)
Top-p ($p=0.5$)	1.49% (0.11%)	1.37s (0.02s)	Top-p ($p=0.8$)	5.79% (0.09%)	1.42s (0.17s)
Selection by ranking			*Selection by ranking*		
Greedy	1.33% (0.06%)	1.28s (0.03s)	Greedy	5.76% (0.04%)	1.31s (0.10s)
Top-k ($k=2$)	**1.07% (0.03%)**	2.48s (0.02s)	Top-k ($k=2$)	**5.39% (0.06%)**	2.56s (0.13s)
Rejection (20%)	1.26% (0.03%)	1.51s (0.04s)	Rejection (20%)	5.63% (0.05%)	1.60s (0.08s)
Top-p ($p=0.5$)	1.28% (0.04%)	1.46s (0.06s)	Top-p ($p=0.8$)	5.61% (0.03%)	1.72s (0.08s)

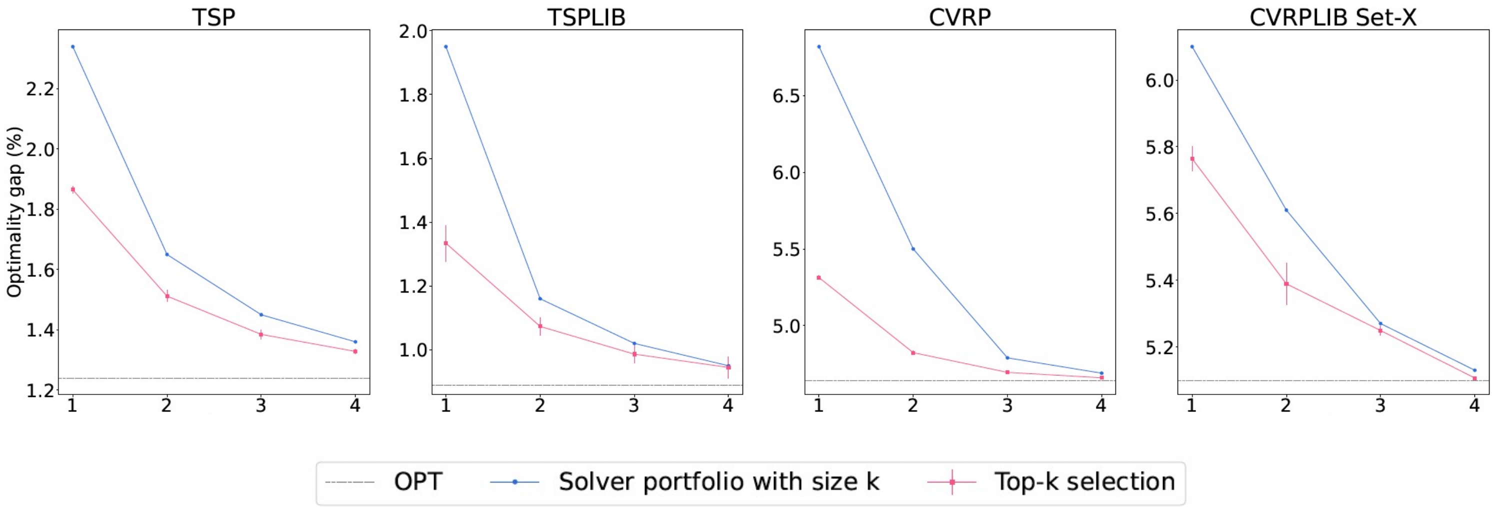
Our method is robust against the distribution shifts

[Gao, Shang, Xue, and Qian, ICML 2025]

Experimental Results

Comparison between top-k selection and solver portfolio with the same size

- Solver portfolio: The optimal solver subset obtained by enumeration



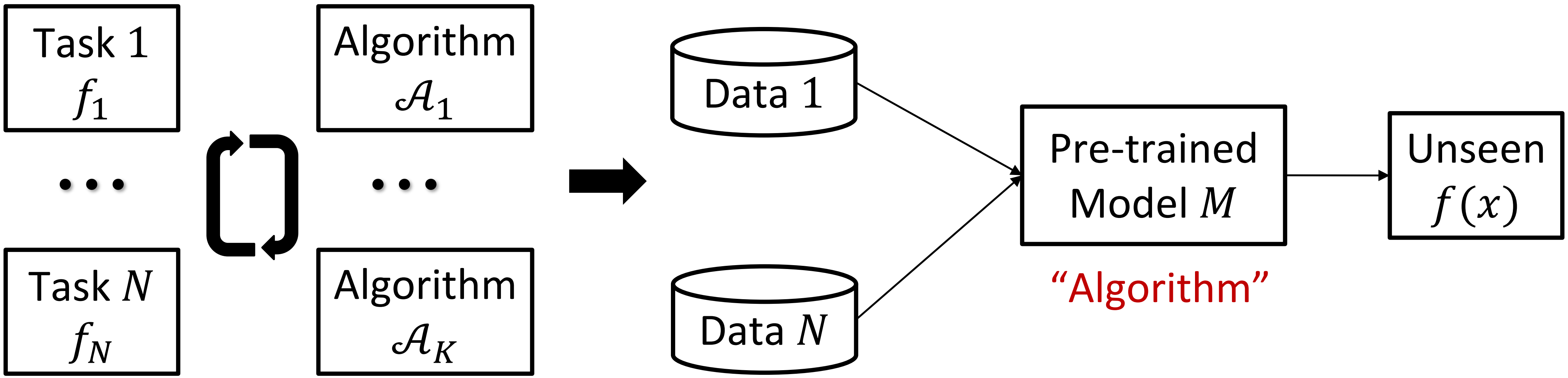
Our top-k selection consistently outperforms the size-k solver portfolio

We hope this work can open a new line for NCO. Many future works: Neural solver feature, runtime-aware selection, complementary solvers, etc.

End-to-end Learning for Black-box Optimization (BBO)

BBO: Optimize an objective function $f(x)$, with the only permission of querying $f(x)$

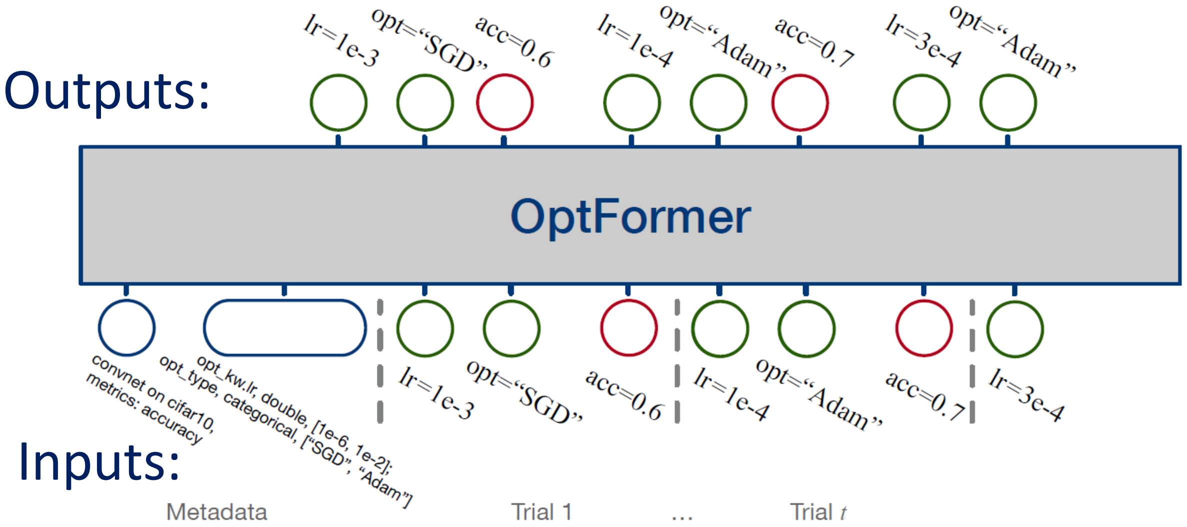
End-to-end learning for BBO: Utilize data from the task distribution $P(\mathcal{F})$ to pre-train a model M, which performs like an algorithm to optimize unseen objective functions



OPTFormer

[Google, NeurIPS'22] makes the first attempt on learning end-to-end black-box optimizers

- Convert the metadata (description of the problem and algorithm) into text
- Convert the historical optimization trajectory of classical algorithms into text
- Train the OPTFormer to learn the converted trajectories from datasets



Training $\mathcal{L}(\theta; m, h) = \sum_{n} \log P_\theta(h^{(n)} | m, h^{(1:n-1)})$

Inference $\pi(x_t | m, h_{t-1}) = \prod_{d=1}^{D} p_\theta(x_t^d | m, h_{t-1}, x_t^{(1:d-1)})$

Imitate the behavior of algorithms with an identifier of algorithm

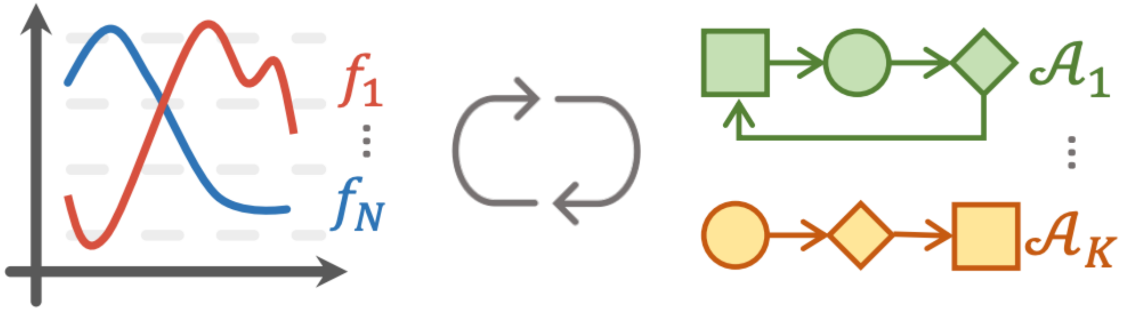
Cannot select proper algorithms automatically

Reinforced In-context BBO

Source task: $f_i \sim P(\mathcal{F})$

Behavior algorithm: $\mathcal{A}_j, j = 1, \cdots K$

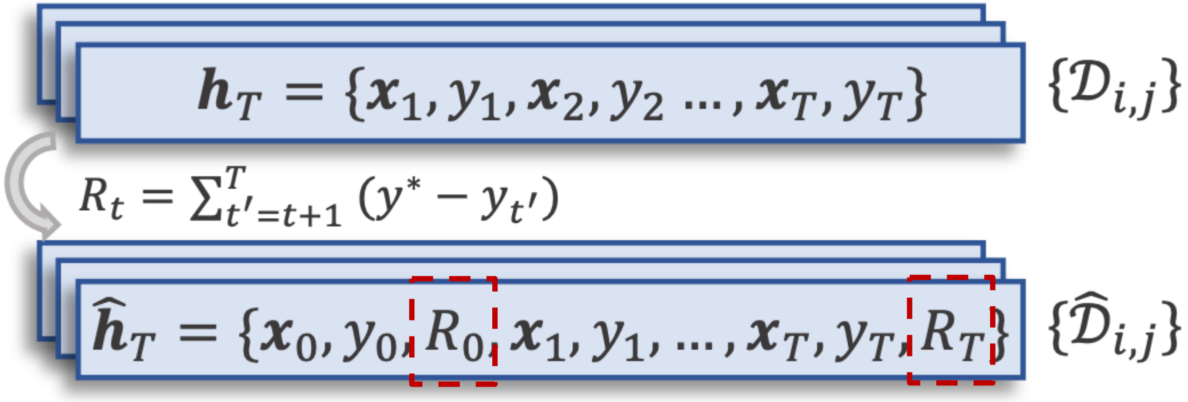
Offline dataset: $\mathcal{D}_{i,j} = \left\{\boldsymbol{h}_T^{i,j,m}\right\}_{m=1}^{M}$



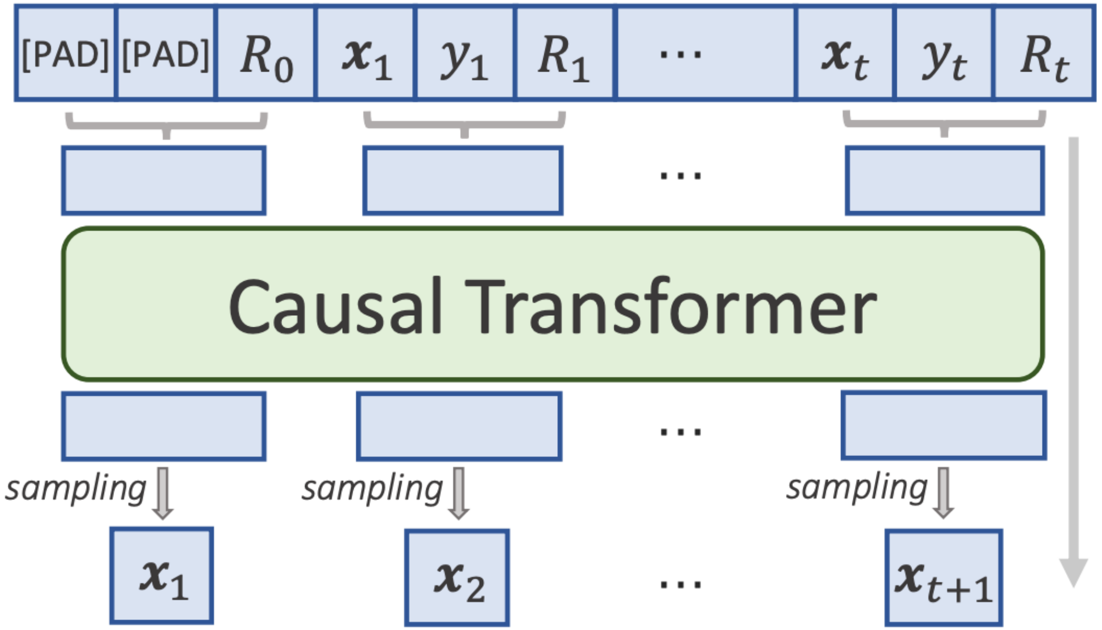
Collected by executing a behavior BBO algorithm $\mathcal{A}_j$ on f_i

Augment histories by *regret-to-go* (RTG)

$$R_t = \sum_{t'=t+1}^{T}(y^* - y_{t'})$$



Reinforced In-context BBO



(b) Training and Inference

$$\mathcal{L}_{RIBBO}(\theta) = -E_{\hat{h}_T \sim \mathcal{D}_{i,j}}\left[\sum_{t=1}^{T} \log M_\theta(x_t | \hat{h}_{t-1})\right]$$

Augmented Histories

$$R_t = \sum_{t'=t+1}^{T} (y^* - y_{t'})$$

Bring identifiability of algorithms and help generate user-desired algorithms automatically

Naïve RTG update strategy for inference $R_t = R_{t-1} - (y^* - y_t)$ Fall below 0

[Song, Gao, Xue, Wu, Li, Hao, Zhang, and Qian, IJCAI 2025]

Reinforced In-context BBO

Hindsight Regret Relabeling (HRR) for inference

Algorithm 1 Model Inference with HRR

Input: trained model $\mathcal{M}_{\boldsymbol{\theta}}$, budget T, optimum value y^*

Process:

Initialize context $\hat{\boldsymbol{h}}_0 = \{(\boldsymbol{x}_0, y_0, R_0)\}$, where $\boldsymbol{x}_0$ and y_0 are placeholders for padding and $R_0 = 0$;
for $t = 1, 2, \ldots, T$ **do**
  Generate the next query point $\boldsymbol{x}_t \sim \mathcal{M}_{\boldsymbol{\theta}}(\cdot|\hat{\boldsymbol{h}}_{t-1})$;
  Evaluate $\boldsymbol{x}_t$ to obtain $y_t = f(\boldsymbol{x}_t)$;
  Calculate the instantaneous regret $r = y^* - y_t$;
  Relabel $R_i \leftarrow R_i + r$, for each $(\boldsymbol{x}_i, y_i, R_i)$ in $\hat{\boldsymbol{h}}_{t-1}$;
  $\hat{\boldsymbol{h}}_t = \hat{\boldsymbol{h}}_{t-1} \cup \{(\boldsymbol{x}_t, y_t, 0)\}$;
end for

The immediate RTG is set as 0 to generate the most advantageous solutions

Previous RTG tokens are updated by adding the current regret

$$R_i = \sum_{t'=i+1}^{t} (y^* - y_{t'})$$

$$R_t = \sum_{t'=t+1}^{T} (y^* - y_{t'}) = 0$$

$$R_i = \sum_{t'=i+1}^{T} (y^* - y_{t'})$$

Experimental Settings

Behavior algorithms

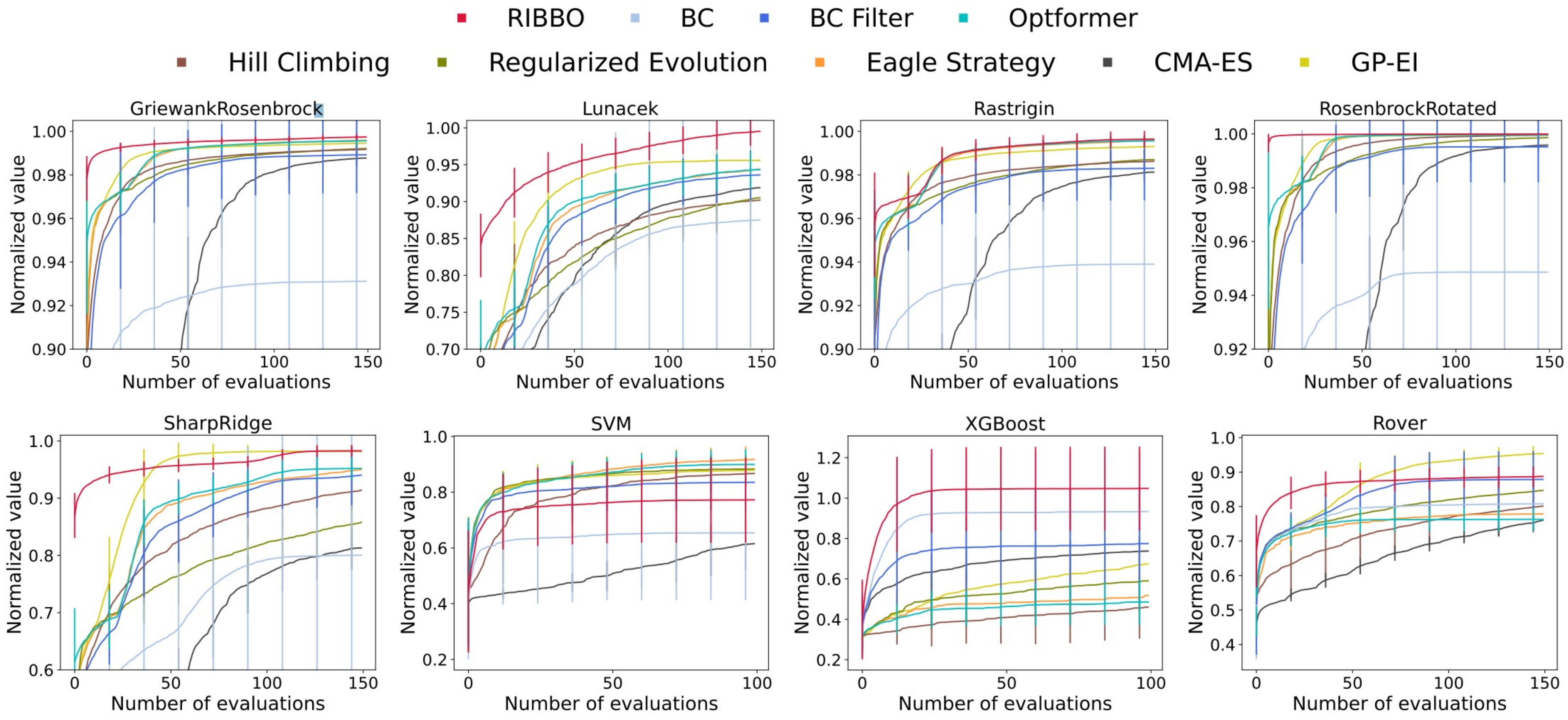
- Heuristic search, e.g., random search, shuffled grid search, hill climbing
- Evolutionary algorithms, e.g., regularized evolution, eagle strategy, CMA-ES
- Bayesian optimization, e.g., GP-EI

Benchmarks

- BBOB functions [Elhara et al., 2019]
- HPO [Arango et al., 2021]
- Robot control problems [Wang et al., 2018]

A series of transformation are used to construct training and test data sets for BBOB and robot control problems, and a training and test split is provided by the authors for HPO

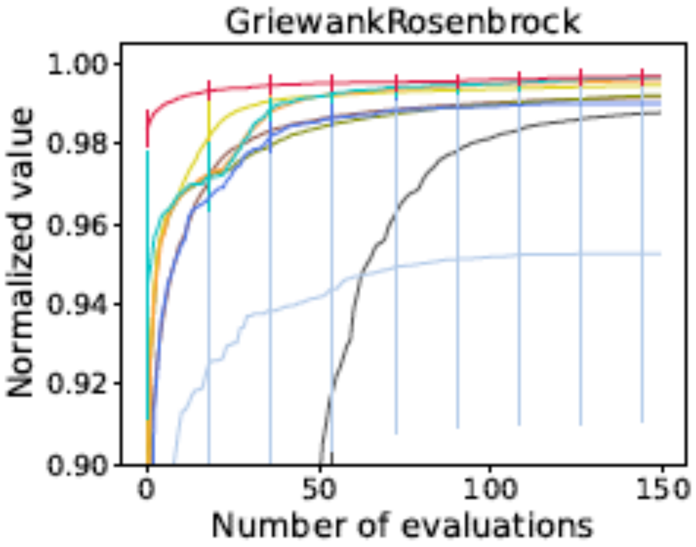
Experimental Results



RIBBO outperforms the behavior algorithms and baselines

Experimental Results

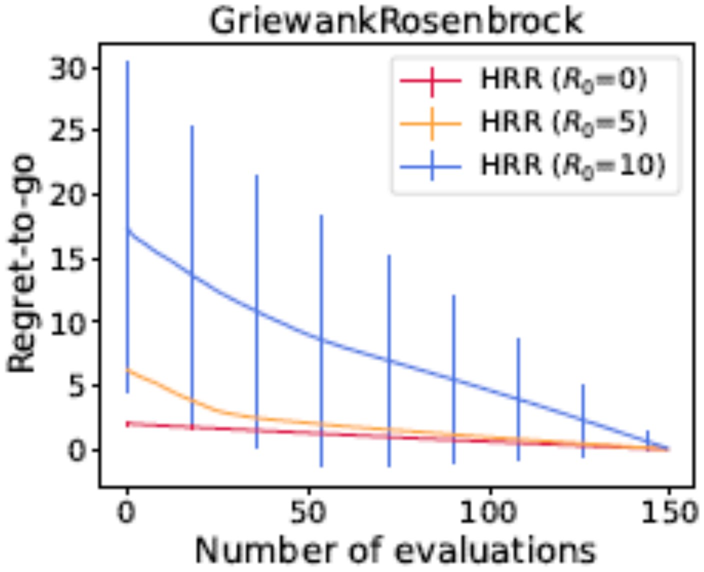
Cross-distribution generalization



RIBBO can generalize to unseen function distributions

train on 4 other function distributions and test on GriewankRosenbrock, which has different properties

Influence of initial RTG token

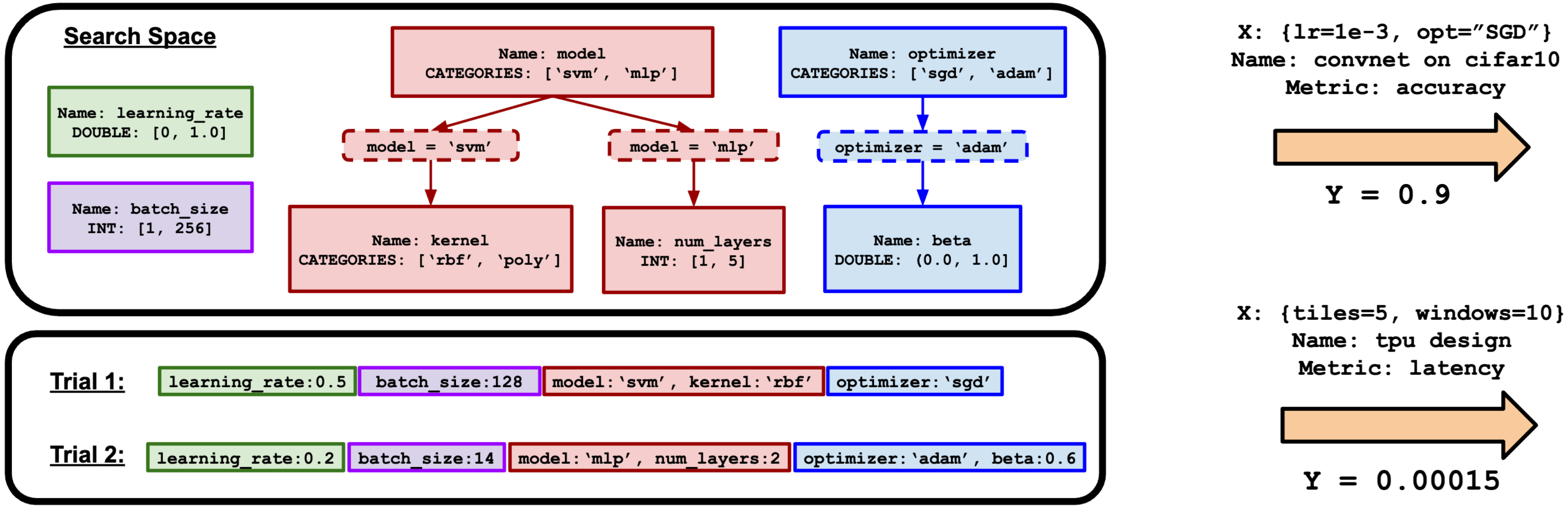


By incorporating RTG tokens into the optimization histories, RIBBO can automatically generate user-desired optimization trajectories

Universal Heterogenous BBO

Different source tasks may have heterogenous search spaces [Google, TMLR'24]

i.e., different dimensionalities, variable types



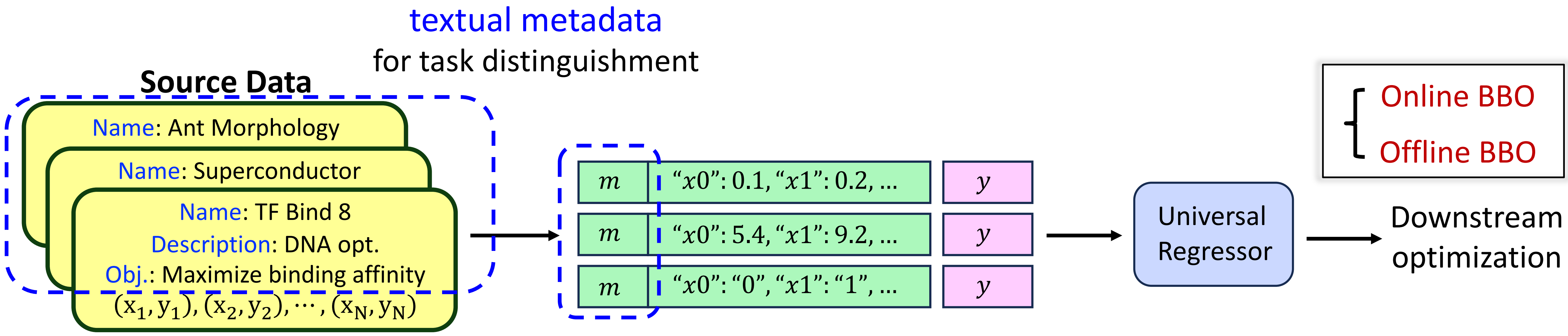
Heterogenous and dynamic search spaces

Divergent objectives

Traditional BBO algorithms cannot handle inputs from heterogeneous spaces

UniSO: Universal String-based BBO

Use string-based representation to train a language-model-based universal regressor



Key-value like solution string: overcome the heterogeneity

	Textual string input representation example
x	"x1":0.3241,"x2":64.0012,"x3":7.1285...
m	Name:"Ant Morphology". Description:"a quadruped robot morphology optimization". Objective:"to run as fast as possible"
y	<+><7><2><5><E-1> or 72.5

[Tan, Chen, Xue, Wang, Wang, Fu, and Qian, ICML 2025]

UniSO: Universal String-based BBO

Two variants of universal end-to-end regressor based on how to deal with y

P10 numeric encoding

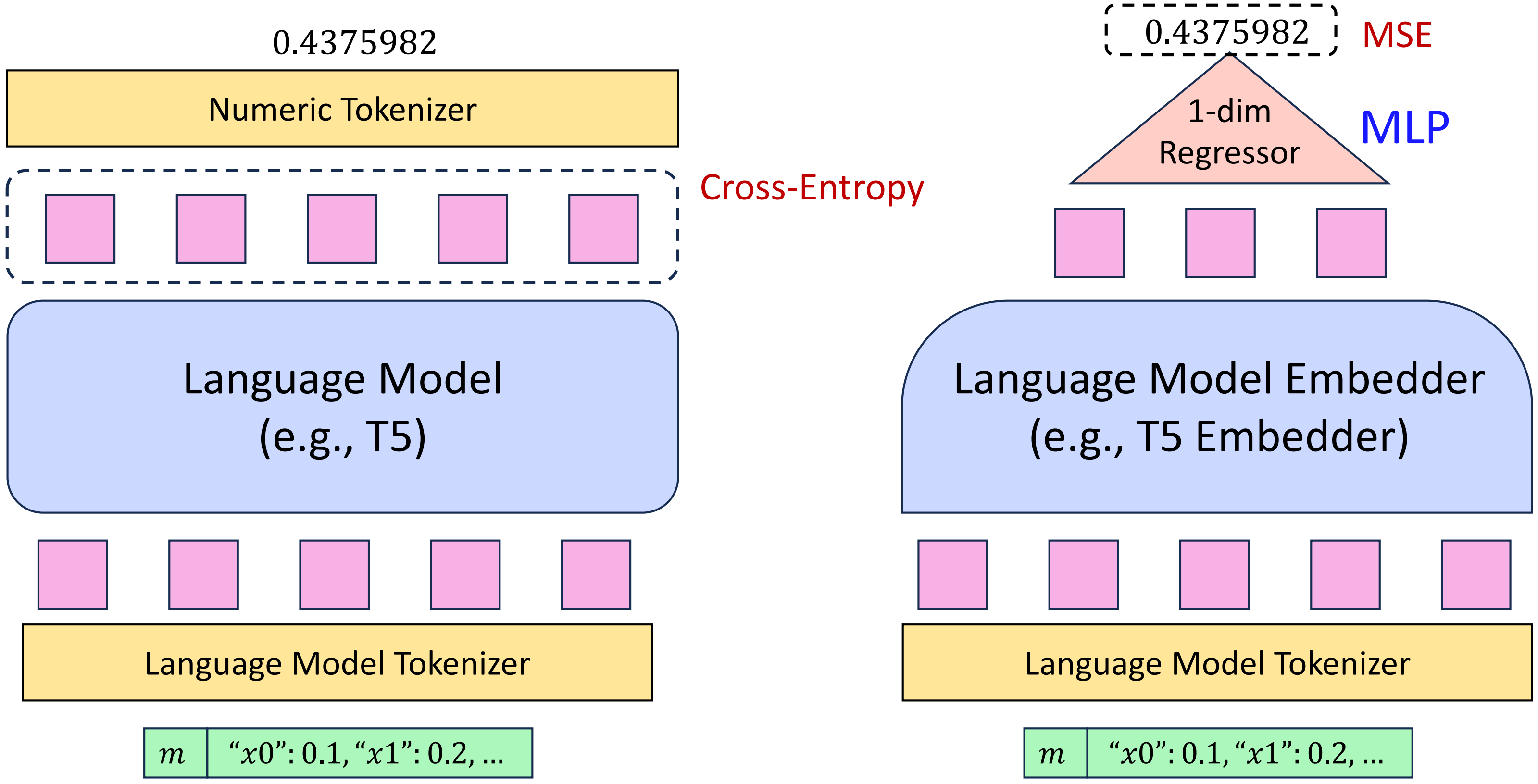
[Charton, TMLR'22]

$$y = 1.31 = 131 \times 10^{-2}$$

<+><1><3><1><E-2>

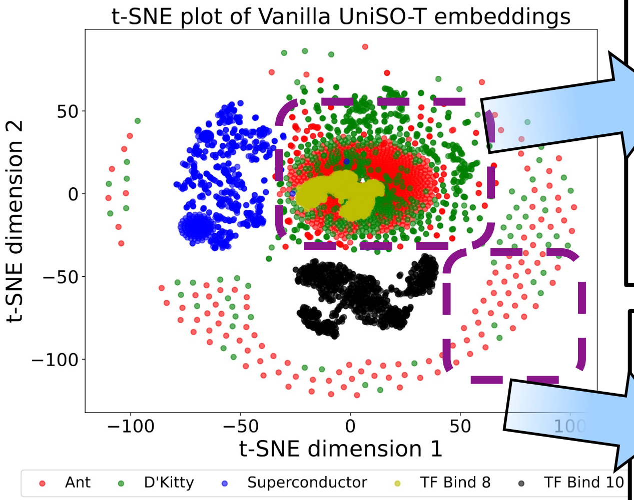
(a) UniSO-T: Token-targeted

(b) UniSO-N: Numeric-targeted



Issues of Vanilla UniSO and Improvement Techniques

tSNE of vanilla UniSO embeddings



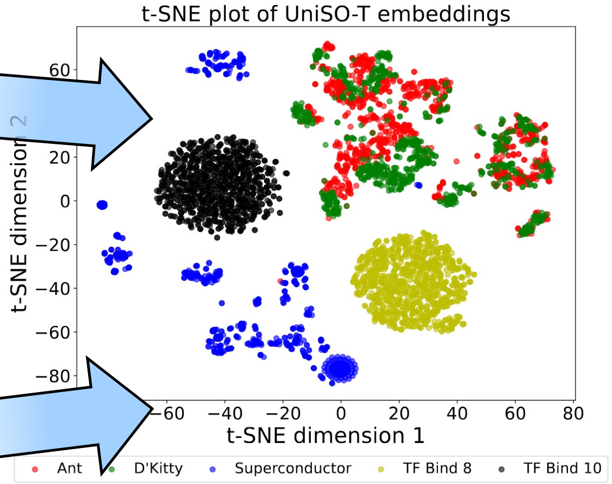
UniSO cannot effectively distinguish dissimilar tasks

- **Solution: Embedding alignment via metadata guidance**

$$\mathcal{L}_{\text{con}} = -\frac{1}{N(N-1)} \sum_{1\le i<j\le N} \hat{s}_{ij}^{m} \log \left(\frac{\exp(s_{ij}^{x}/\tau)}{\sum_{k\neq i} \exp(s_{ik}^{x}/\tau)} \right)$$

Encourage input embeddings to exhibit similar cross-task patterns to the metadata embedding

Similar tasks remain close
Dissimilar tasks are far



Non-smoothness of intra-task embeddings

- **Solution: Local Smoothness Enhancement of Embedding**

$$\mathcal{L}_{\text{lip}_{\mathcal{T}}} = \sum_{1\le i<j\le N_{\mathcal{T}}} \max\left(0, \frac{|y_i - y_j|}{\|\mathbf{z}_i - \mathbf{z}_j\|_2} - L\right), \quad \mathcal{L}_{\text{lip}} = \sum_{i=1}^{n_{\mathcal{T}}} \frac{\sum_{j=1}^{n_{\mathcal{T}}} N_{\mathcal{T}_j}}{N_{\mathcal{T}_i}} \mathcal{L}_{\text{lip}_{\mathcal{T}_i}}$$

Regularize the embedding to preserve its local Lipschitz continuity w.r.t. the objective score y

Minimizing $\mathcal{L}_{\text{lip}_{\mathcal{T}}}$ is theoretically important for optimization [Lee et al., NeurIPS'23]

[Tan, Chen, Xue, Wang, Wang, Fu, and Qian, ICML 2025]

Experimental Settings

Experiments on offline BBO (since there is a large amount of data to train the model)

- **Model structure:** T5-small encoder-decoder structure
- **Training:** Train for 200 epochs, using AdamW with a batch size of 128
- **Model-inner search:** Evolutionary algorithms (EAs)

Training tasks: Tasks from Design-Bench [Trabucco et al., ICML'22] and SOO-Bench [Qian et al., ICLR'25]

Benchmark Suite	Task	Dataset size	Variable type	# dimensions
Design-Bench	Ant	10004	CONTINUOUS	56
	D'Kitty	10004	CONTINUOUS	60
	Superconductor	17010	CONTINUOUS	86
	TF Bind 8	32898	CATEGORICAL	8 (3 categories)
	TF Bind 10	10000	CATEGORICAL	10 (3 categories)
SOO-Bench	GTOPX 2	22000	CONTINUOUS	22
	GTOPX 3	18000	CONTINUOUS	18
	GTOPX 4	26000	CONTINUOUS	26
	GTOPX 6	22000	CONTINUOUS	22

[Tan, Chen, Xue, Wang, Wang, Fu, and Qian, ICML 2025]

Experimental Results

Comparison on UniSO methods to single-task expert methods

Best

Runner up

Task	$\mathcal{D}$(best)	Single-task Experts		UniSO-T		UniSO-N	
		BN + EAs	BN + Grad	Vanilla	Improved	Vanilla	Improved
Ant	165.326	118.877 ± 127.688	229.462 ± 165.869	**275.216 ± 90.820**	**374.665 ± 56.057**	268.399 ± 82.858	269.691 ± 77.425
D'Kitty	199.363	111.205 ± 66.986	183.263 ± 62.436	**216.070 ± 23.209**	**225.752 ± 8.521**	130.655 ± 83.106	173.911 ± 46.662
Superconductor	74.000	**93.951 ± 7.039**	**97.137 ± 6.113**	86.795 ± 13.466	92.200 ± 15.209	81.266 ± 16.073	67.333 ± 16.838
TF Bind 8	0.439	**0.984 ± 0.007**	**0.959 ± 0.023**	0.940 ± 0.027	0.903 ± 0.041	0.944 ± 0.016	0.833 ± 0.005
TF Bind 10	0.005	**0.905 ± 0.326**	0.888 ± 0.229	0.830 ± 0.539	0.823 ± 0.542	0.603 ± 0.005	**0.959 ± 0.115**
GTOPX 2	-195.586	**-88.054 ± 20.878**	-128.310 ± 15.616	-132.023 ± 63.084	**-72.848 ± 9.576**	-117.022 ± 51.671	-124.995 ± 56.170
GTOPX 3	-151.190	-64.028 ± 22.678	-151.190 ± 0.000	**-60.941 ± 17.235**	**-45.602 ± 8.433**	-88.601 ± 31.865	-62.622 ± 22.261
GTOPX 4	-215.716	**-96.432 ± 10.868**	-215.716 ± 0.000	-100.943 ± 15.044	**-84.271 ± 8.307**	-99.834 ± 20.837	-110.284 ± 17.559
GTOPX 6	-112.599	-64.217 ± 14.602	-112.599 ± 0.000	-71.749 ± 28.497	**-47.794 ± 11.943**	-71.174 ± 12.932	**-57.435 ± 18.832**
Avg. Rank	/	**3.111 ± 1.728**	4.111 ± 1.792	3.667 ± 1.333	**2.111 ± 1.663**	4.222 ± 1.030	3.778 ± 1.618

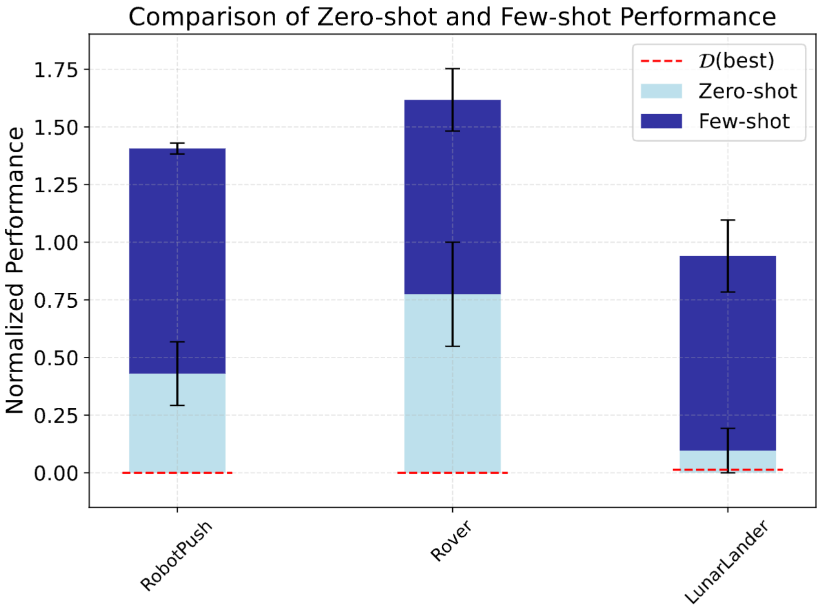
Findings:

1. All the UniSO methods outperform $\mathcal{D}$(best), showing their feasibilities
2. Performance comparison: UniSO-T > single-task experts ≈ UniSO-N
3. Our proposed techniques consistently show improvements

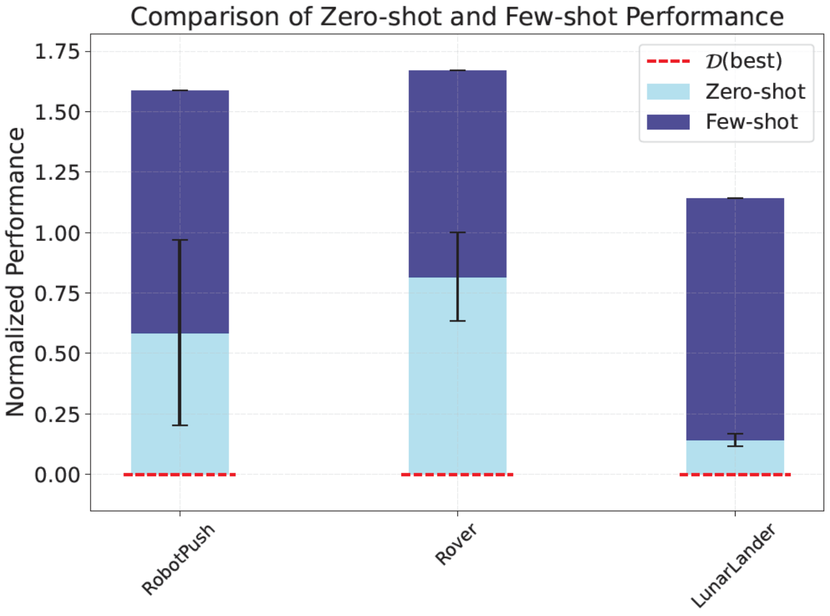
[Tan, Chen, Xue, Wang, Wang, Fu, and Qian, ICML 2025]

Experimental Results

Performance of zero-shot and few-shot fine-tuning on 3 unseen tasks



Improved UniSO-T



Improved UniSO-N

Finetune Setting

optimizer: SGD

learning rate: 2e-5

few-shot data: 100

loss: main loss

Epochs: 5

Findings: UniSO generalizes well to unseen tasks

Few-shot fine-tuning significantly improves the performance

[Tan, Chen, Xue, Wang, Wang, Fu, and Qian, ICML 2025]

成果被国家自然科学基金委报道

国家自然科学基金委员会
National Natural Science Foundation of China

请输入关键字 搜索

首 页 | 机构概况 | 政策法规 | 项目指南 | 申请资助 | 共享传播 | 国际合作 | 信息公开

当前位置：首页 >> 基金要闻 >> 资助成果

政务微信

获项目资助本科生在人工智能黑箱优化领域取得进展

日期 2025-06-03 来源：信息科学部 作者：夏莹杰、吴国政、王志衡、谢国、王乐、刘家男 【大 中 小】 【打印】 【关闭】

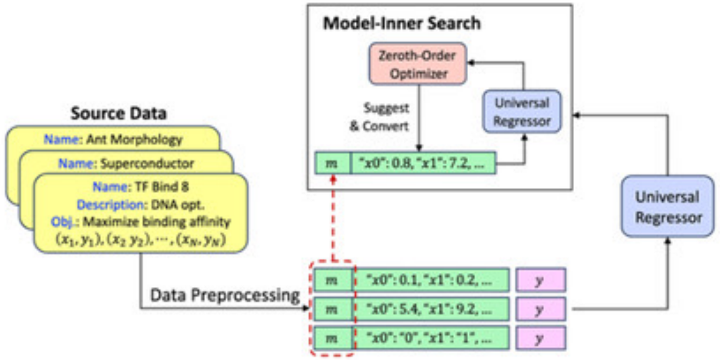


图 基于字符串表示的通用离线黑箱优化框架示意图

在国家自然科学基金青年学生基础研究项目（本科生，批准号：624B1025）的资助下，南京大学人工智能学院本科生谭荣熙在导师钱超教授指导下，在人工智能离线黑箱优化研究中取得进展，创新地提出了基于排序学习和基于字符串表示的离线黑箱优化框架，提升了多场景人工智能应用中黑箱优化的稳健性和通用性。研究成果以**第一作者身份被机器学习领域顶级国际会议第13届**International Conference on Learning Representations（ICLR'25）和**第42届**International Conference on Machine Learning（ICML'25）录用，项目代码已全部开源（论文题目和代码链接见本文附录）。

Outline

- Build theoretical foundation of EAs
  - Theoretical analysis tools, influence analysis of major factors of EAs
- Develop better EL algorithms
  - Efficient EL, dynamic algorithm configuration, algorithm selection, universal EL
- **Apply EL to complex optimization in learning, industry, and science**
  - **Subset selection, electronic design automation, origin and evolution of life**

Application I: Subset Selection

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

There are many applications of selecting a good subset from a ground set

observation variables predictor variable

	Corr.	Dis.	LR	···	···	AIC.	BIC	RF.
x1	0.28	0.46	1	···	···	0.22	0.63	1
x2	0.31	0.59	0.64	···	···	0.58	0.56	1
x3	0.11	0.02	0.53	···	···	0.43	0.01	1
x4	0.1	0.1	0.64	···	···	0.73	0.92	1
x5	0.02	0.15	0.33	···	···	0.56	0.36	0.78
x6	0.36	0.02	0.01	···	···	0.32	0.02	0.22
x7	0.2	0.2	0.21	···	···	0.21	0.02	0.11
x8	0.1	0.03	0.32	···	···	0.33	0.51	0.44
x9	0.32	0.1	0.2	···	···	0.06	0.66	0
x10	0.24	0	0.02	···	···	0.6	0.03	0.33
x11	0.12	0.45	0.44	···	···	0.64	0.45	1
x12	0.36	0.58	0.12	···	···	0.73	0.58	0.67
x13	0.2	0.02	0.24	···	···	0.34	0.02	0.89
x14	0.24	0.92	0.33	···	···	0.24	0.93	0.56

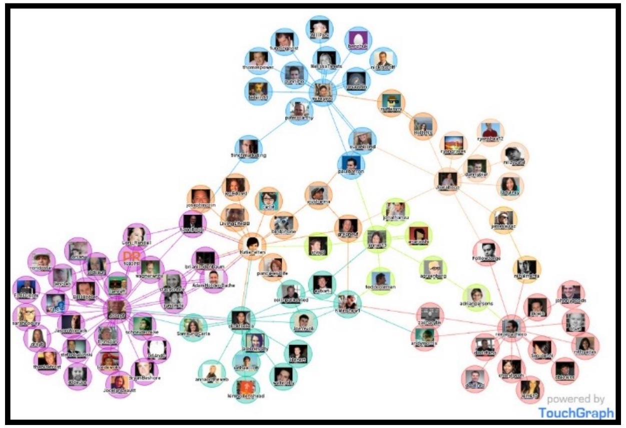
Sparse regression

a subset of observation variables

	Corr.	Dis.	LR	···	···	AIC.	BIC	RF.
x1	0.28	0.46	1	···	···	0.22	0.63	1
x2	0.31	0.59	0.64	···	···	0.58	0.56	1
x3	0.11	0.02	0.53	···	···	0.43	0.01	1
x4	0.1	0.1	0.64	···	···	0.73	0.92	1
x5	0.02	0.15	0.33	···	···	0.56	0.36	0.78
x6	0.36	0.02	0.01	···	···	0.32	0.02	0.22
x7	0.2	0.2	0.21	···	···	0.21	0.02	0.11
x8	0.1	0.03	0.32	···	···	0.33	0.51	0.44
x9	0.32	0.1	0.2	···	···	0.06	0.66	0
x10	0.24	0	0.02	···	···	0.6	0.03	0.33
x11	0.12	0.45	0.44	···	···	0.64	0.45	1
x12	0.36	0.58	0.12	···	···	0.73	0.58	0.67
x13	0.2	0.02	0.24	···	···	0.34	0.02	0.89
x14	0.24	0.92	0.33	···	···	0.24	0.93	0.56

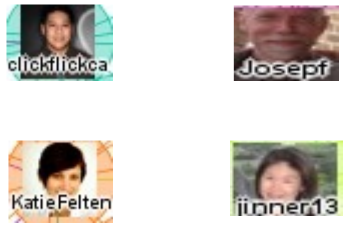
Application I: Subset Selection

There are many applications of selecting a good subset from a ground set



Influence maximization

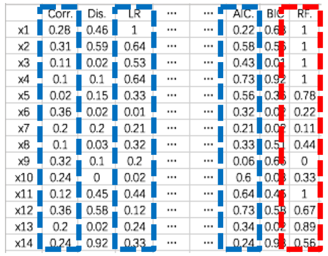
Influential users



Application I: Subset Selection

There are many applications of selecting a good subset from a ground set

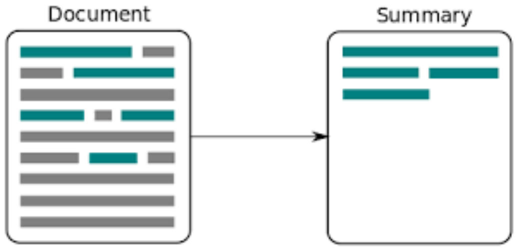
Sparse regression



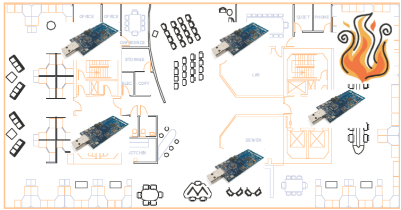
	Corr.	Dis.	LR	···	···	AIC.	BIC	RF.
x1	0.28	0.46	1	···	···	0.22	[illegible]	1
x2	0.31	0.59	0.64	···	···	0.58	[illegible]	1
x3	0.11	0.02	0.53	···	···	0.43	[illegible]	1
x4	0.1	0.1	0.64	···	···	0.73	0.92	1
x5	0.02	0.15	0.33	···	···	0.56	[illegible]	0.78
x6	0.36	0.02	0.01	···	···	0.32	[illegible]	0.22
x7	0.2	0.2	0.21	···	···	0.21	[illegible]	0.11
x8	0.1	0.03	0.32	···	···	0.33	[illegible]	0.44
x9	0.32	0.1	0.2	···	···	0.06	[illegible]	0
x10	0.24	0	0.02	···	···	0.6	[illegible]	0.33
x11	0.12	0.45	0.44	···	···	0.64	[illegible]	1
x12	0.36	0.58	0.12	···	···	0.73	[illegible]	0.67
x13	0.2	0.02	0.24	···	···	0.34	[illegible]	0.89
x14	0.24	0.92	0.33	···	···	0.24	[illegible]	0.56

Influence maximization

Document summarization



Sensor placement



Ground set V $\xrightarrow[|S| \leq b]{\max f(S)}$ Subset $S \subseteq V$

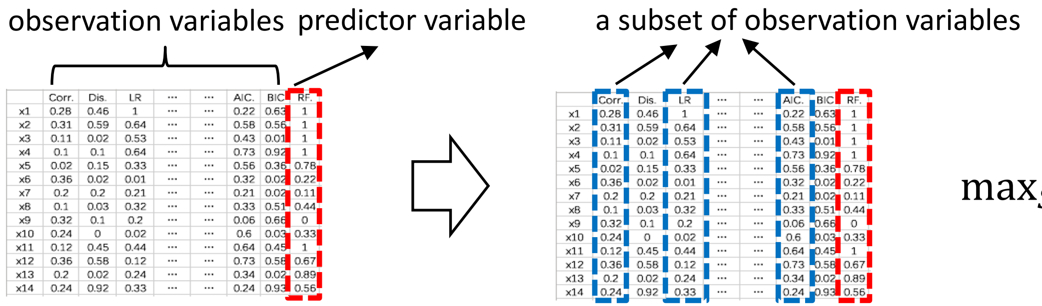
Subset Selection: Given all items $V = \{v_1, \ldots, v_n\}$, an objective function $f: 2^V \to \mathbb{R}$ and a budget b, to select a subset $S \subseteq V$ such that

$$\max_{S \subseteq V} \quad f(S) \quad \text{s.t.} \quad |S| \leq b$$

NP-hard

Application I: Subset Selection

Comparison on sparse regression



$$\max_{S \subseteq V} \quad R^2_{z,S} = \frac{\mathrm{Var}(z) - \mathrm{MSE}_{z,X}}{\mathrm{Var}(z)} \quad \text{s.t.} \quad |S| \le b$$

exhaustive search: OPT; Evolutionary algorithm: POSS; greedy algorithms: FR, FoBa, OMP, RFE; relaxation methods: MCP

Data set	OPT	POSS	FR	FoBa	OMP	RFE	MCP
housing	.7437±.0297	.7437±.0297	.7429±.0300•	.7423±.0301•	.7415±.0300•	.7388±.0304•	.7354±.0297•
eunite2001	.8484±.0132	.8482±.0132	.8348±.0143•	.8442±.0144•	.8349±.0150•	.8424±.0153•	.8320±.0150•
svmguide3	.2705±.0255	.2701±.0257	.2615±.0260•	.2601±.0279•	.2557±.0270•	.2136±.0325•	.2397±.0237•
ionosphere	.5995±.0326	.5990±.0329	.5920±.0352•	.5929±.0346•	.5921±.0353•	.5832±.0415•	.5740±.0348•
sonar	–	.5365±.0410	.5171±.0440•	.5138±.0432•	.5112±.0425•	.4321±.0636•	.4496±.0482•
triazines	–	.4301±.0603	.4150±.0592•	.4107±.0600•	.4073±.0591•	.3615±.0712•	.3793±.0584•
coil2000	–	.0627±.0076	.0624±.0076•	.0619±.0075•	.0619±.0075•	.0363±.0141•	.0570±.0075•
mushrooms	–	.9912±.0020	.9909±.0021•	.9909±.0022•	.9909±.0022•	.6813±.1294•	.8652±.0474•
clean1	–	.4368±.0300	.4169±.0299•	.4145±.0309•	.4132±.0315•	.1596±.0562•	.3563±.0364•
w5a	–	.3376±.0267	.3319±.0247•	.3341±.0258•	.3313±.0246•	.3342±.0276•	.2694±.0385•
gisette	–	.7265±.0098	.7001±.0116•	.6747±.0145•	.6731±.0134•	.5360±.0318•	.5709±.0123•
farm-ads	–	.4217±.0100	.4196±.0101•	.4170±.0113•	.4170±.0113•	–	.3771±.0110•
POSS: win/tie/loss		–	12/0/0	12/0/0	12/0/0	11/0/0	12/0/0

• denotes that POSS is significantly better by the t-test with confidence level 0.05

EA is always significantly better

[Qian, Yu and Zhou, NeurIPS 2015]

Application I: Subset Selection

EA can achieve the optimal polynomial-time approximation guarantee

Theorem 1. *For subset selection with monotone objective functions, POSS with* $\mathbb{E}[T] \leq 2eb^2 n$ *and* $I(\cdot) = 0$, *i.e., a constant function, can find a solution* $\boldsymbol{s}$ *with* $|\boldsymbol{s}|_1 \leq b$ *and* $\boxed{f(\boldsymbol{s}) \geq (1 - e^{-\gamma_{\min}}) \cdot \mathrm{OPT},}$ *where* $\gamma_{\min} = \min_{\boldsymbol{s}:|\boldsymbol{s}|_1 = b-1} \gamma_{\boldsymbol{s},b}$.

Proved to be the optimal polynomial-time approximation [Harshaw et al., ICML'19]

Under noise, EA achieves better approximation guarantees than conventional algorithms

Theorem 2. *For subset selection under multiplicative noise with the assumption Eq. (17.29), with probability at least* $(1/2)(1 - (12nb^2 \log 2b)/l^{2\delta})$, *PONSS with* $\theta \geq \epsilon$ *and* $T = 2elnb^2 \log 2b$ *finds a solution* $\boldsymbol{s}$ *with* $|\boldsymbol{s}|_1 \leq b$ *and* $f(\boldsymbol{s}) \geq \frac{1-\epsilon}{1+\epsilon}(1 - e^{-\gamma}) \cdot \mathrm{OPT}$.

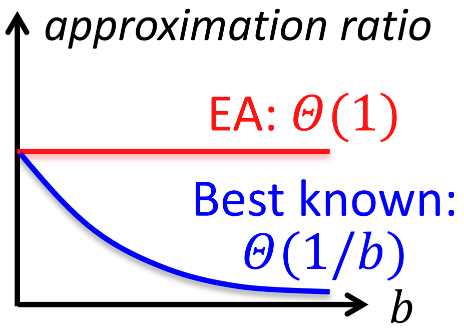
EA

$$\frac{f(S)}{\mathrm{OPT}} \geq \frac{1-\epsilon}{1+\epsilon}(1 - e^{-\gamma})$$

Significantly better

Best known [Horel and Singer, NeurIPS'16]

$$\frac{f(S)}{\mathrm{OPT}} \geq \frac{1}{1 + \frac{2\epsilon b}{(1-\epsilon)\gamma}} \left(1 - \left(\frac{1-\epsilon}{1+\epsilon}\right)^b e^{-\gamma}\right)$$



constant γ and ϵ

[Qian, Yu and Zhou, NeurIPS 2015; Qian, et al. NeurIPS 2017]

应用案例2：芯片自动化设计

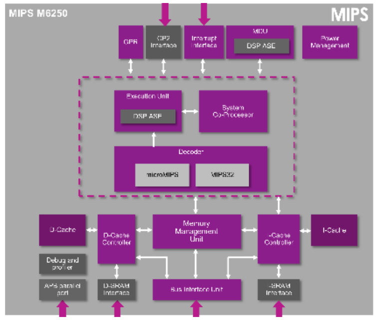
功能设计: 设计 RTL 并验证其功能 (文档 → RTL)

逻辑综合: 将RTL设计映射为网表信息 (RTL → 网表)

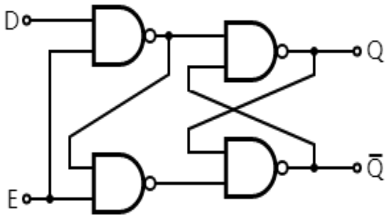
物理设计: 基于网表设计GDS物理版图 (网表 → GDS)

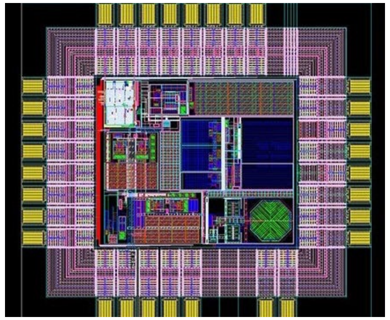
芯片制造: 通过光刻技术从GDS版图制造芯片 (GDS → 芯片产品)

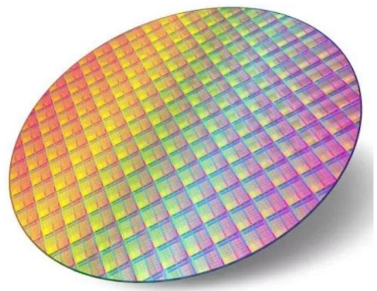
芯片设计文档 —开发→ RTL逻辑设计 —逻辑综合→ 网表 —物理设计→ GDS物理版图 —制造→ 晶圆 —封装测试→ 芯片产品

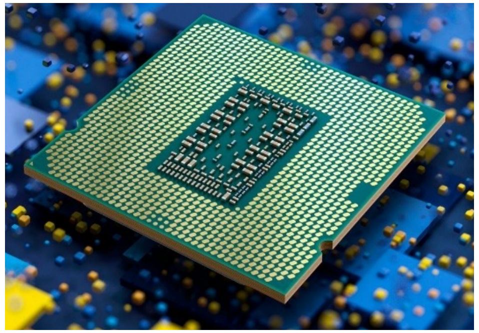



```
module alu32(Result, ALUOp, A, B, Zero) ;
  output [`ALULEN:0] Result;
  reg    [`ALULEN:0] Result;
  output Zero;
  reg    Zero;
  input  [2:0]  ALUOp;
  input  [`ALULEN:0] A, B;

  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
```



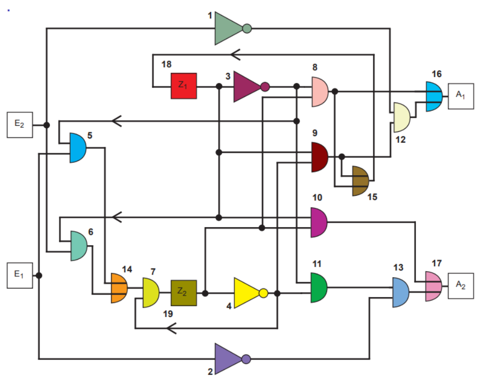




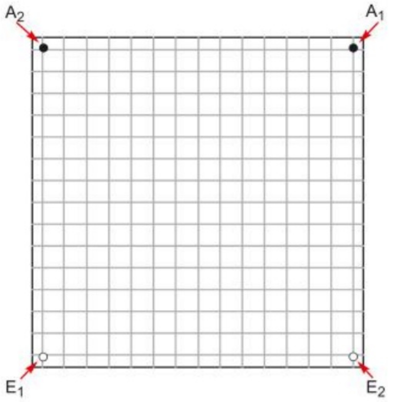


Learning And Mining from DatA
http://www.lamda.nju.edu.cn

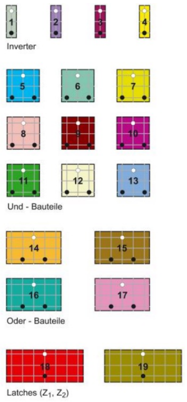
应用案例2-1：芯片元件布局



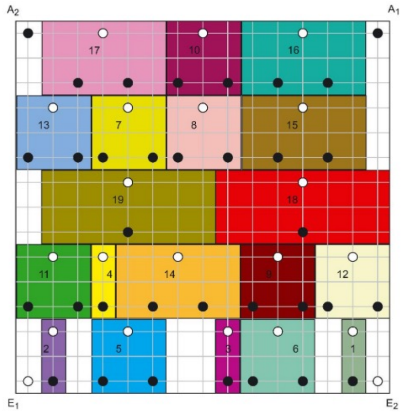
网表



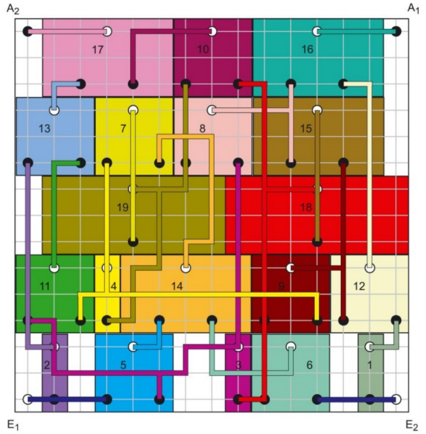
芯片版面



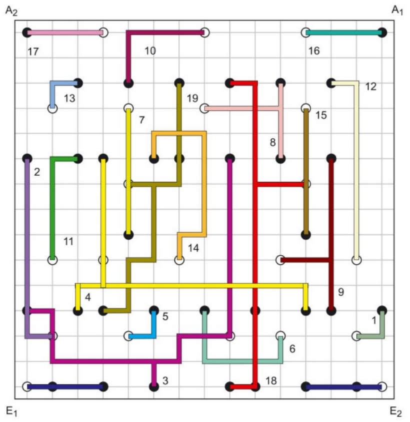
待放置元件

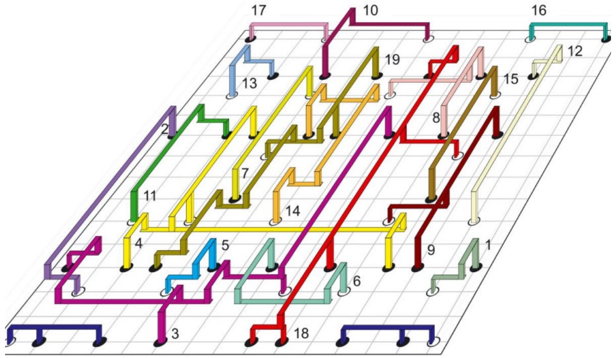


布局



布线





应用案例2-1：芯片元件布局

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

性能指标比较：线长

Table 1: $\boxed{\text{HPWL values } (\times 10^5)}$ obtained by ten compared methods on seven chips. Each result consists of the mean and standard deviation of five runs. The best (smallest) mean value on each chip is bolded. The symbols '+', '−' and '≈' indicate the number of chips where the result is significantly superior to, inferior to, and almost equivalent to WireMask-EA, respectively, according to the Wilcoxon rank-sum test with significance level 0.05.

Method	Type	adaptec1	adaptec2	adaptec3	adaptec4	bigblue1	bigblue3	bigblue4 ($\times 10^7$)	$+/-/\approx$	Avg. Rank
SP-SA [30]	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 [10]	Analytical	26.62	321.17	328.44	462.93	22.85	455.53	48.38	0/7/0	9.00
RePlace [11]	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 [25]	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 [29]	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 [13]	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 [23]	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**

[Google, Nature 2021]

我们方法

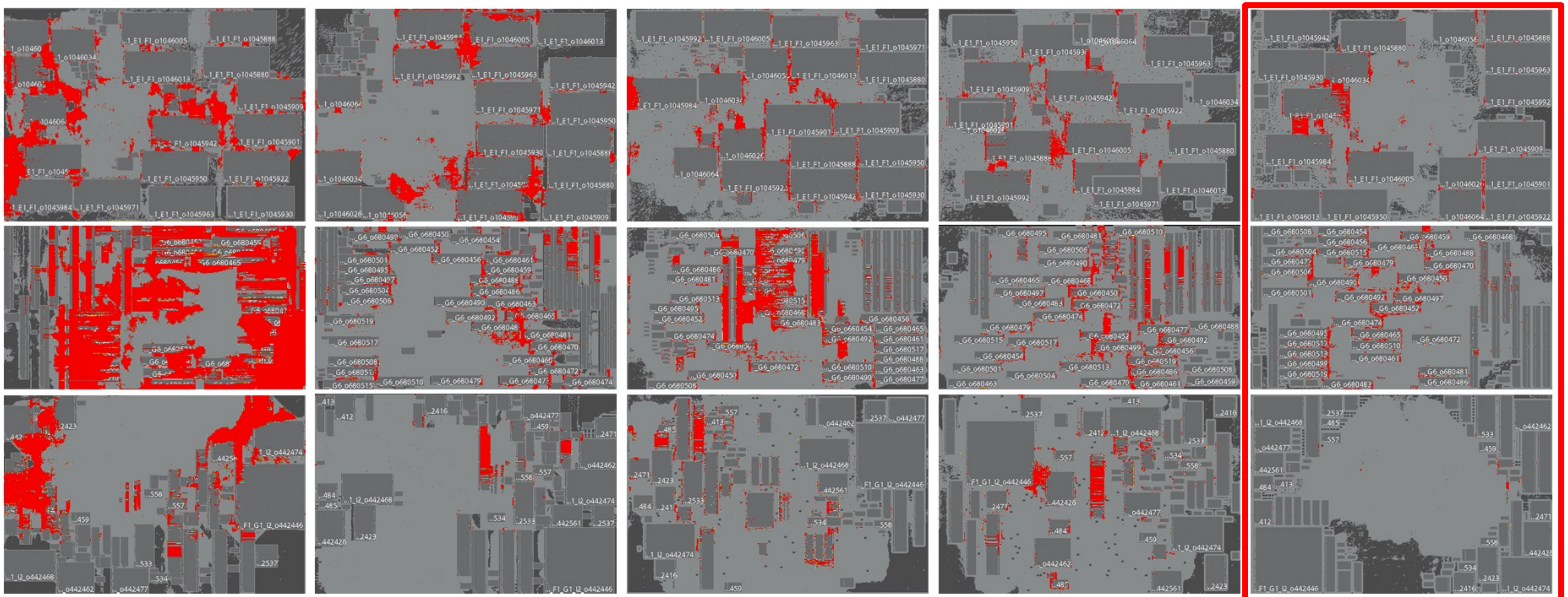
在测试的 7 个芯片上，至少 6 个显著好；较 [Google, Nature'21]，线长缩短80%

应用案例2-1：芯片元件布局

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

性能指标比较：拥塞

我们方法

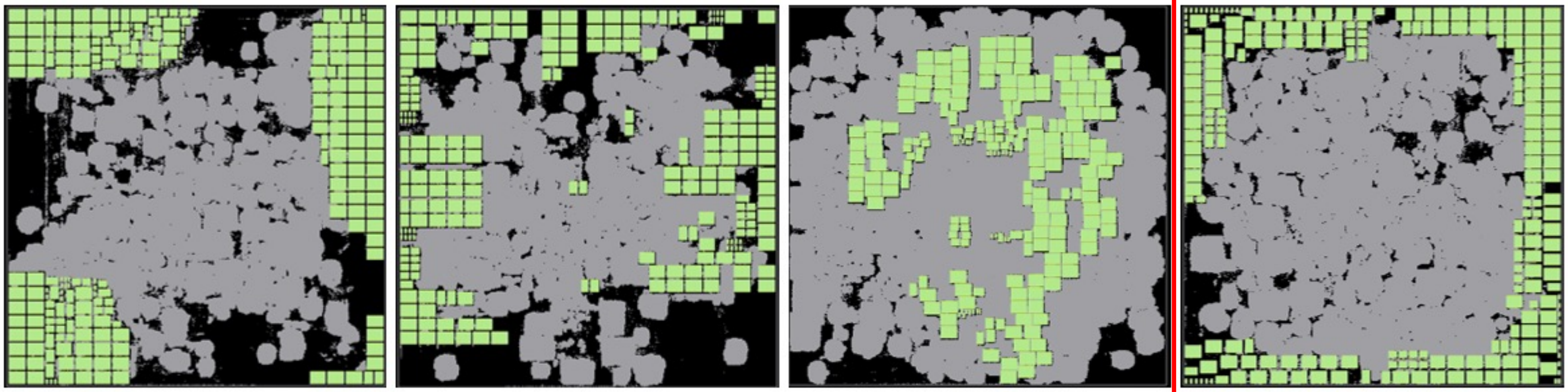


(a) DREAMPlace (b) AutoDMP (c) WireMask-EA (d) MaskPlace (e) MaskRegulate

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

应用案例2-1：芯片元件布局

性能指标比较：规整度



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

我们方法

应用案例2-1：芯片元件布局

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

性能指标比较：时序

Benchmark	DREAMPlace* [20] TNS	DREAMPlace* [20] WNS	DREAMPlace 4.0* [18] TNS	DREAMPlace 4.0* [18] WNS	Differentiable-TDP† [12] TNS	Differentiable-TDP† [12] WNS	Distribution-TDP§ [19] TNS	Distribution-TDP§ [19] WNS	Ours TNS	Ours 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**

总时序 TNS
平均提升 40.5%

最差时序 WNS
平均提升 8.3%

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

应用案例2-1：芯片元件布局

应用案例2-1：芯片元件布局


DATE 25

Design, Automation & Test in Europe

DATE Best Paper Award 2025

in Track D: Design Methods and Tools

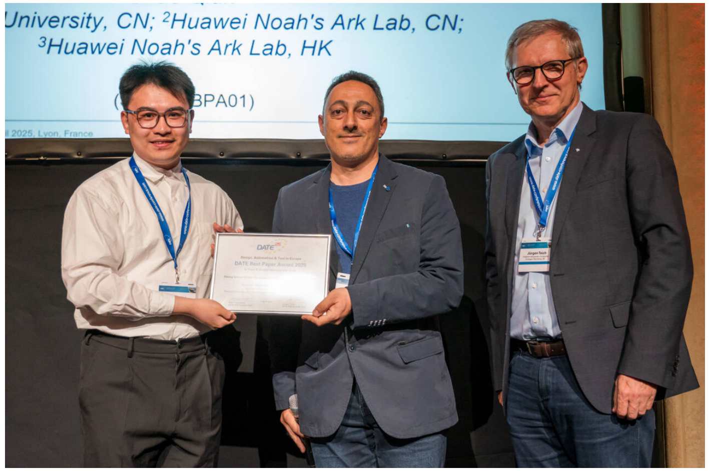
Timing-Driven Global Placement by Efficient Critical Path Extraction

Yunqi Shi[1], Siyuan Xu[2], Shixiong Kai[2], Xi Lin[1],
Ke Xue[1], Mingxuan Yuan[3], Chao Qian[1]
[1]Nanjing University, CN; [2]Huawei Noah's Ark Lab, CN; [3]Huawei Noah's Ark Lab, HK

Aida Todri-Sanial
DATE 2025 General Chair

1 April 2025

Jürgen Teich
DATE 2025 Awards Chair



获 EDA 领域顶级国际会议 DATE’25

最佳论文奖

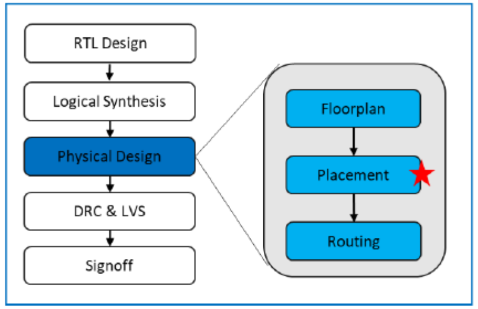
应用案例2-1：芯片元件布局

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

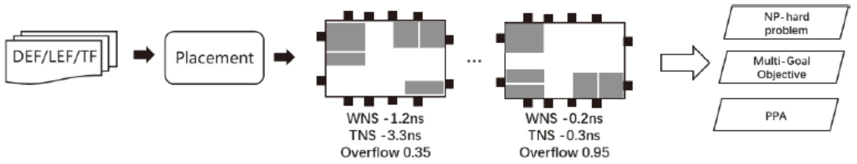
技术背景

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。

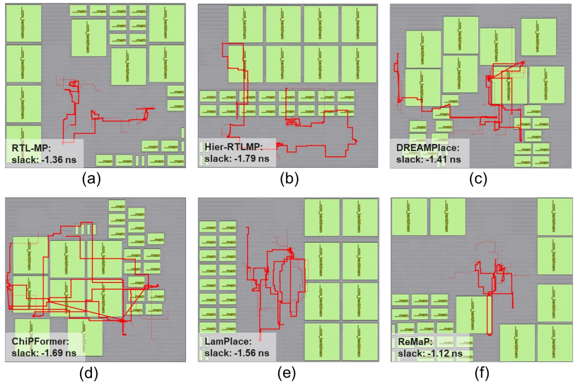


Chip Design Flow　　Auto Placement and Route (APR) Flow

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%）



a) RTL-MP
b) Hier-RTLMP
c) DREAMPlace
d) ChiPFormer
e) LamPlace
f) ReMaP（钱超）

方案对比布局时序图

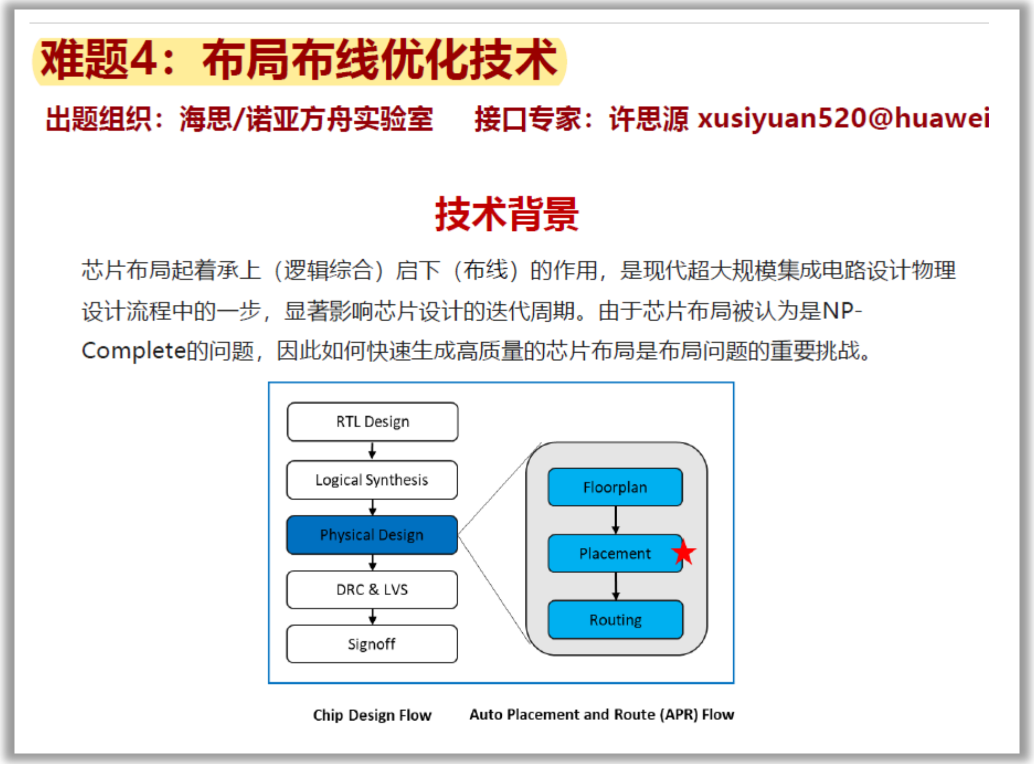
火花奖第115期

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

应用案例2-1：芯片元件布局

和华为合作攻克2D布局难题



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

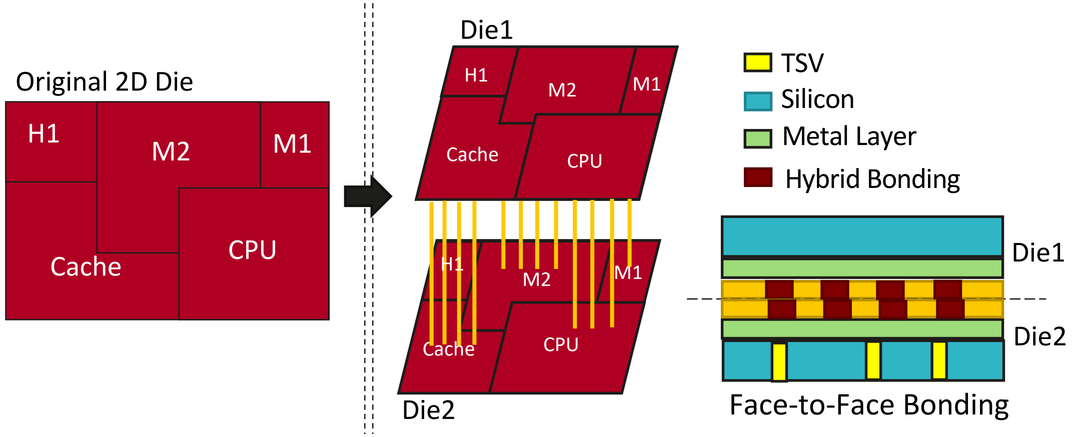
芯片布局起着承上（逻辑综合）启下（布线）的作用，是现代超大规模集成电路设计物理设计流程中的一步，显著影响芯片设计的迭代周期。由于芯片布局被认为是NP-Complete的问题，因此如何快速生成高质量的芯片布局是布局问题的重要挑战。

Chip Design Flow　Auto Placement and Route (APR) Flow

同时合作探索更为前沿的3D堆叠布局

芯片3D堆叠布局技术

芯片3D集成是一种新的设计范式，通过硅芯片或通过与嵌入式芯片的多层互连，提供了传统2D平面IC实现到3D堆叠的空间转换。

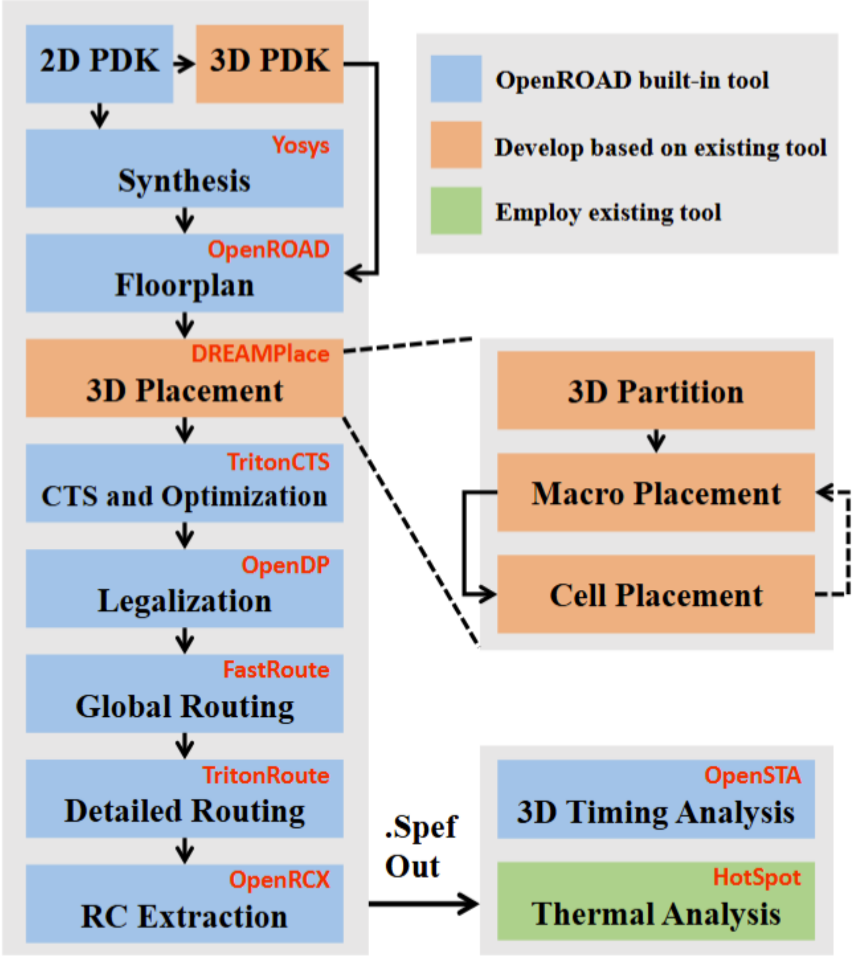


先进芯片设计 缓解 当前先进制造工艺的局限

应用案例2-1：芯片元件布局

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]

应用案例2-1：芯片元件布局

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

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 in area, routing overflow, timing and power. 3D packing also brings about thermal issues.

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

应用案例2-2：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”


今日的思想火花，明天的熊熊大火

“难题揭榜”火花奖第83期

难题	部门/学校院系	姓名/昵称	奖励金额
火花奖			
“难题揭榜”第六十三期--EDA专题第一期-难题5	南京大学人工智能学院	钱超副教授	30000 CNY
流程与IT难题第一期-难题4	先进计算与存储实验室 [2012实验室]	余恩宇 00664120	15000 CNY
HarmonyOS难题第三期-难题3	南京大学计算机科学与技术系	过洁副研究员	10000 CNY

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

应用案例3-1：生命起源与演化

自然科学四大基础科学问题之一：
生命起源与演化

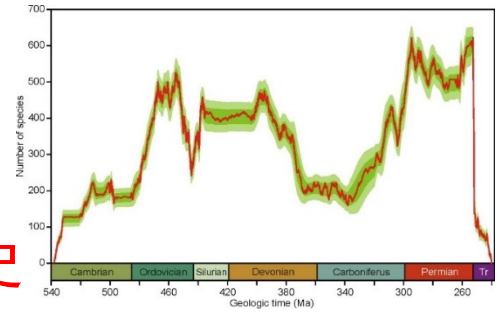
地层剖面
海量化石
记录数据

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

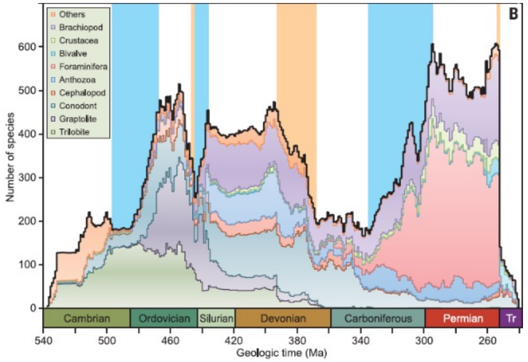
3122个剖面

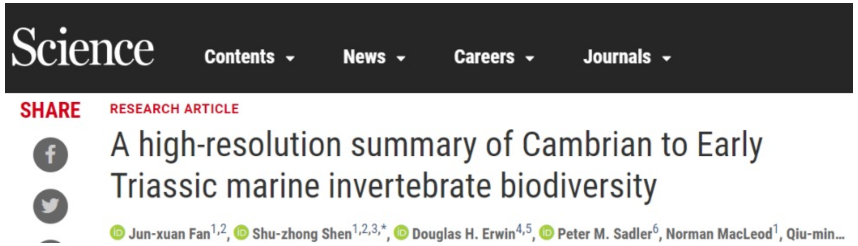
11268个物种

演化算法

“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




Science
Contents News Careers Journals
SHARE
RESEARCH ARTICLE
A high-resolution summary of Cambrian to Early Triassic marine invertebrate biodiversity
Jun-xuan Fan[1,2], Shu-zhong Shen[1,2,3,*], Douglas H. Erwin[4,5], Peter M. Sadler[6], Norman MacLeod[1], Qiu-min...

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

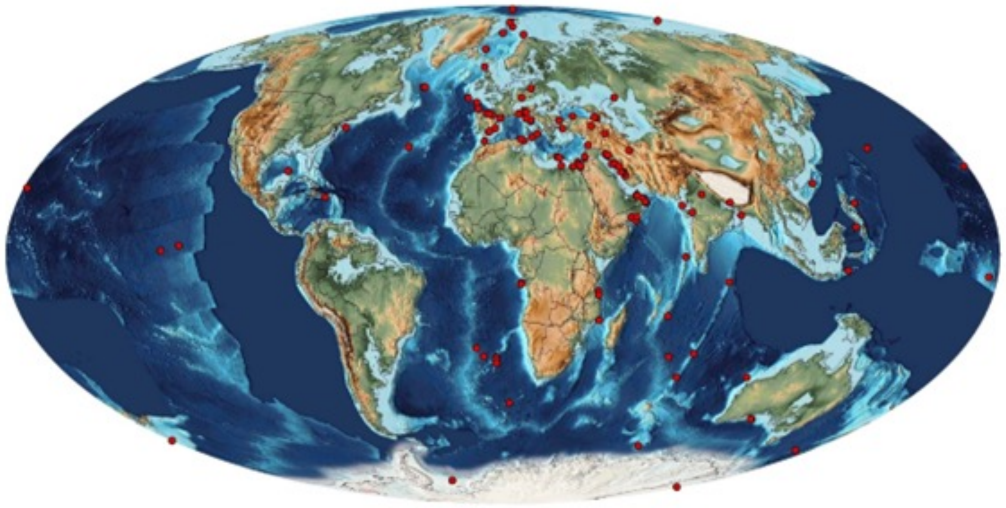
应用案例3-1：生命起源与演化

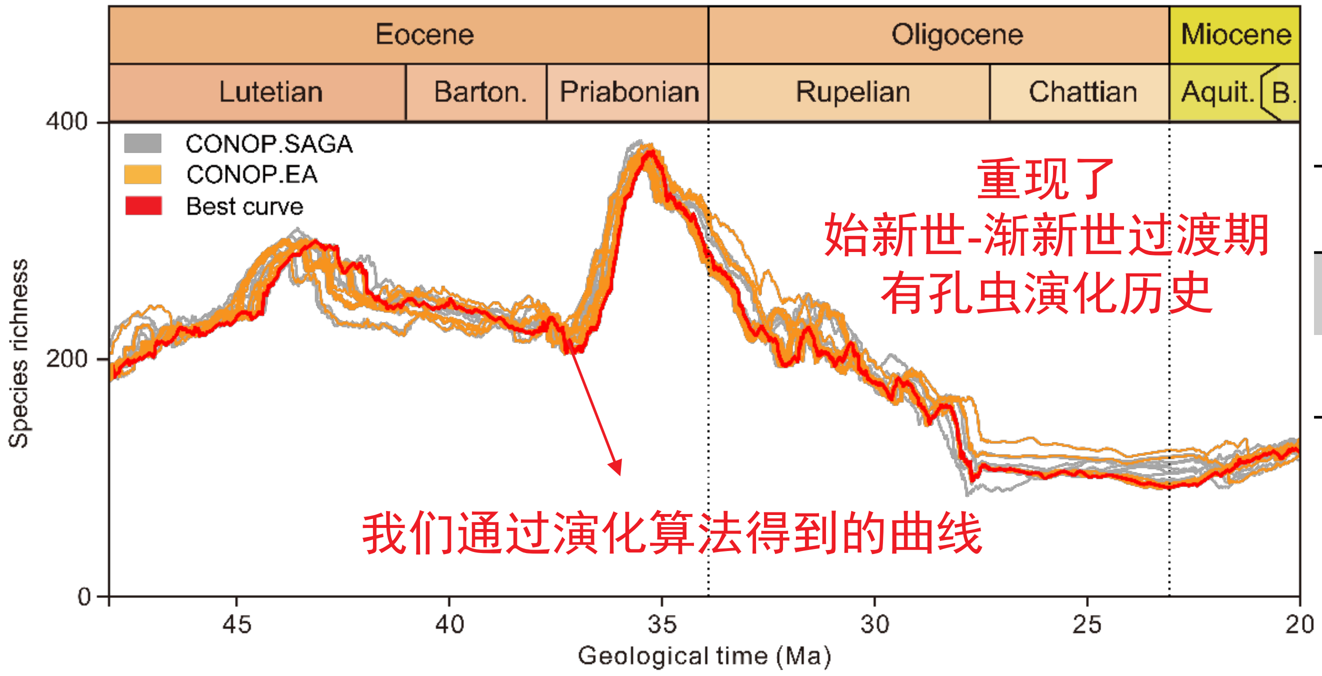
地科团队收集的数据集

数据来源广：分布于全球范围内多个站点

数据规模大：剖面数: 161；物种数: 1291

该数据集是目前全球最大规模的有孔虫数据集





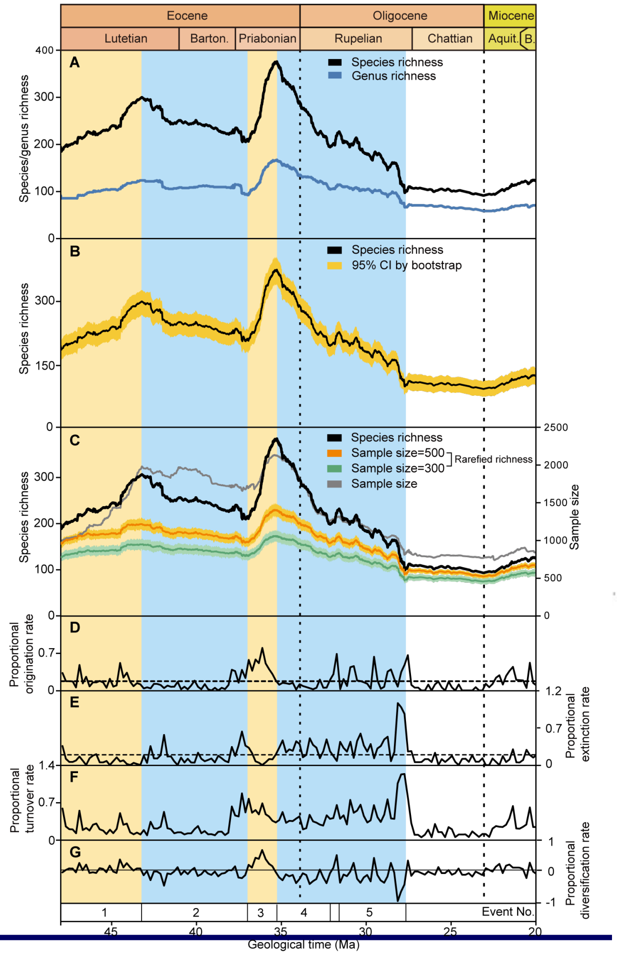
算法	目标值 范围	运行时间
以往算法	39914-41390	~40684s
新算法	39252-39778	~2376s

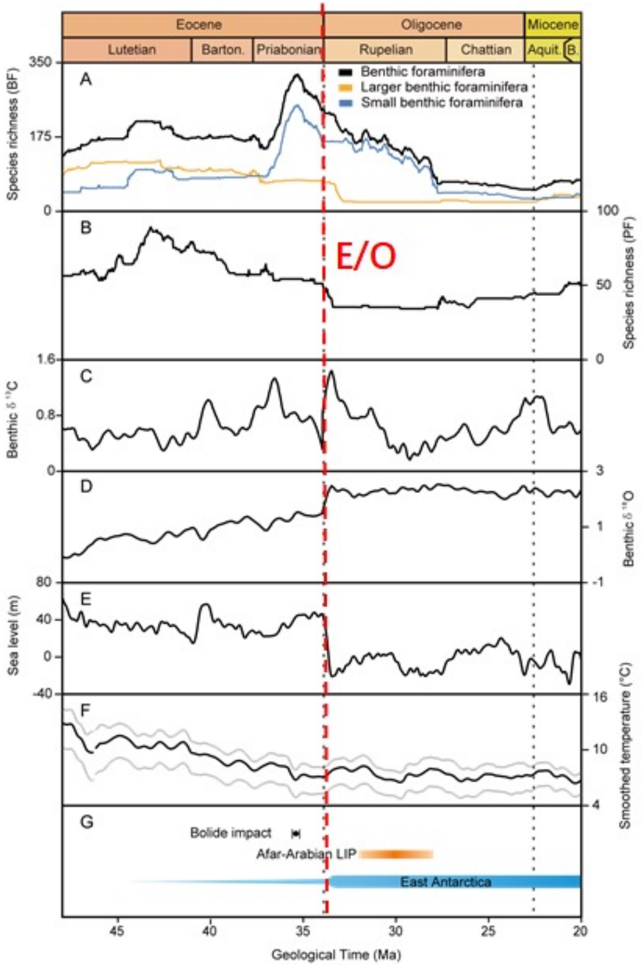
效率提升17倍，目标值更好

精准重现生命演化

应用案例3-1：生命起源与演化

高精度的有孔虫多样性变化曲线有助于精确分析其与气候环境变化的关系





浮游有孔虫和大型底栖有孔虫的灭绝与快速降温、海平面下降和碳同位素正漂移有关

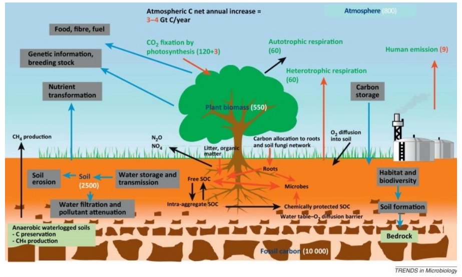
揭示E-O过渡期有孔虫
生物多样性变化的驱动机制
帮助预测生物多样性危机

近期投稿《Science》

应用案例3-2：土壤微生物源碳预测

土壤有机碳对全球碳循环有着重大影响

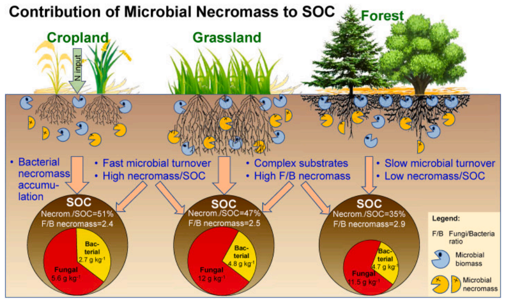
土壤有机碳：最大的陆地碳库



缓解气候变化、保证粮食安全

微生物源碳

土壤有机碳重要组成成分



如何预测其含量?

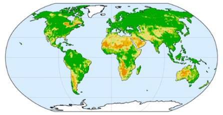
现有方法：氨基糖分析

存在问题：预测误差大

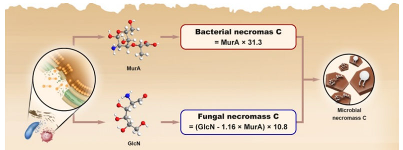
- 数据样本量有限
- 忽略细菌群组成效应

以共同第一作者发表
美国国家科学院院刊
PNAS

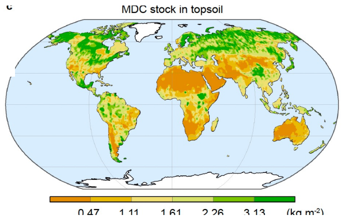
与中科院南京土壤所
合作提出基于演化学习的
土壤微生物源碳预测公式

汇编全球数据集 

联合建模、降低不确定性



使用机器学习、贝叶斯优化
精准预测关键参数



全球高精度土壤
微生物源碳分布

[Hu, Qian, Xue, et al., PNAS 2024]

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

Conclusion

- **Build theoretical foundation of EAs**
  - Theoretical analysis tools, influence analysis of major factors of EAs
- **Develop better EL algorithms**
  - Efficient EL, dynamic algorithm configuration, algorithm selection, universal EL
- **Apply EL to complex optimization in learning, industry, and science**
  - Subset selection, electronic design automation, origin and evolution of life

Collaborators

Zhi-Hua Zhou

Yang Yu

Thank you!