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

Multi-objective Evolutionary Learning

Advances in Theories and Algorithms

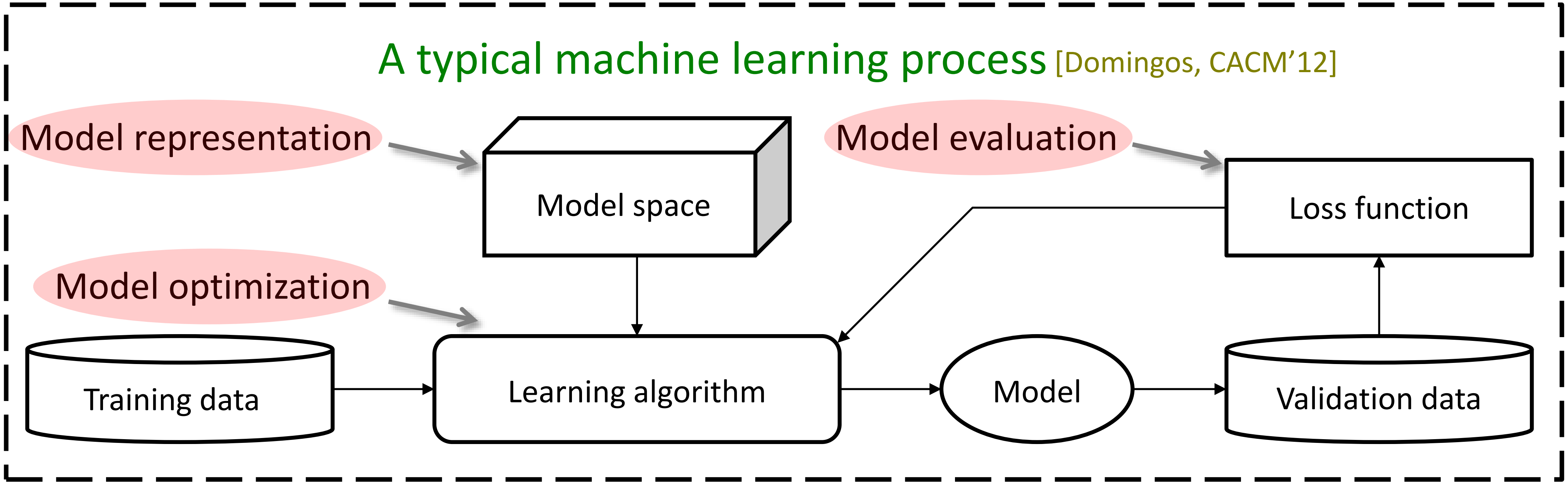
Chao Qian

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

LAMDA Group
Nanjing University, China

Machine Learning

Machine learning aims at learning generalizable models from data

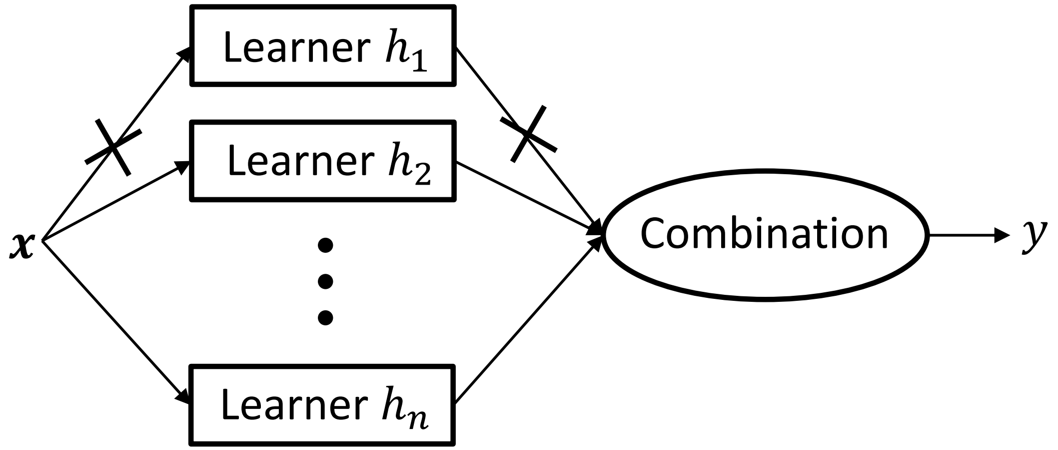


Thus, a machine learning problem is often formulated as an optimization problem

The resulting optimization problems are usually complicated, where the objective can be non-differentiable, non-continuous, non-unique and have many local optima

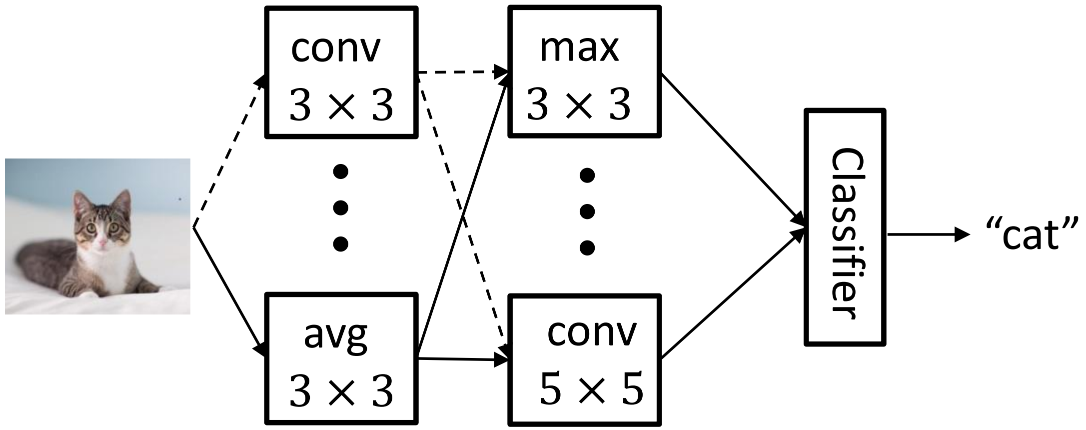
Selective ensemble

- Max: generalization performance
- Min: number of selected learners



Neural architecture search

- Max: accuracy
- Min: computation cost



Multi-objective optimization tries to optimize multiple objectives simultaneously

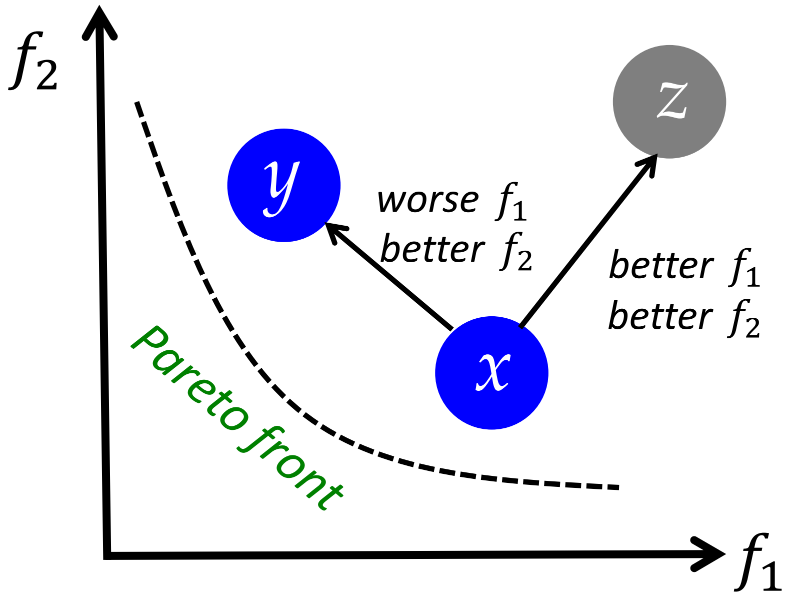
$$\min_{\boldsymbol{s}\in S}\ (f_1(\boldsymbol{s}), f_2(\boldsymbol{s}), \ldots, f_m(\boldsymbol{s}))$$

$\boldsymbol{x}$ dominates $\boldsymbol{z}$:

$$f_1(\boldsymbol{x}) < f_1(\boldsymbol{z}) \wedge f_2(\boldsymbol{x}) < f_2(\boldsymbol{z})$$

$\boldsymbol{x}$ is incomparable with $\boldsymbol{y}$:

$$f_1(\boldsymbol{x}) > f_1(\boldsymbol{y}) \wedge f_2(\boldsymbol{x}) < f_2(\boldsymbol{y})$$



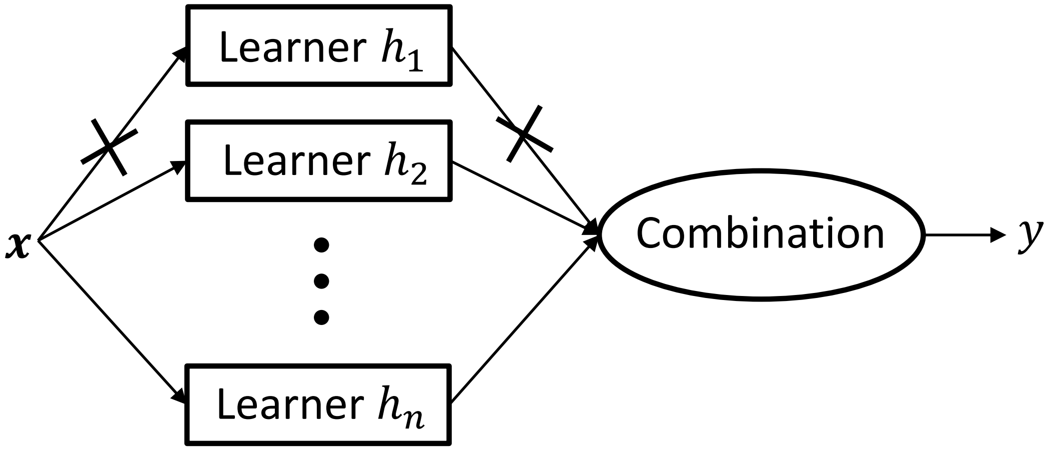
Much more complicated than single-objective optimization

Machine Learning

The resulting optimization problems are usually complicated, where the objective can be non-differentiable, non-continuous, non-unique and have many local optima

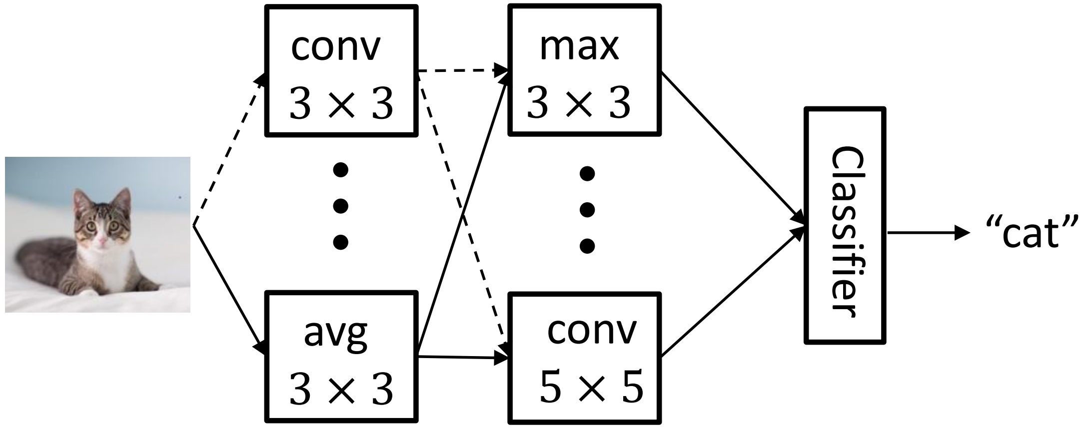
Selective ensemble

- Max: generalization performance
- Min: number of selected learners



Neural architecture search

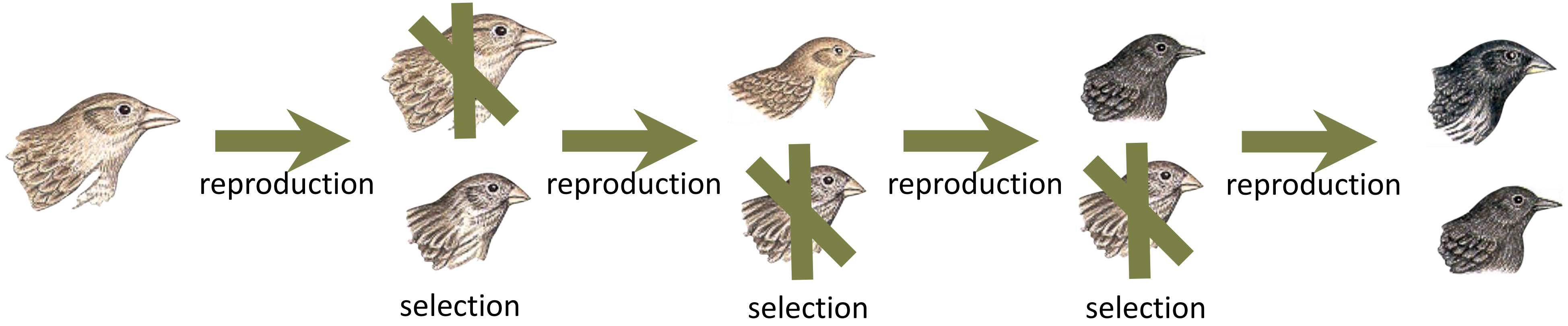
- Max: accuracy
- Min: computation cost



Thus, the conventional optimization algorithms such as gradient descent may fail, while other powerful optimization algorithms are needed

Evolutionary Algorithms

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

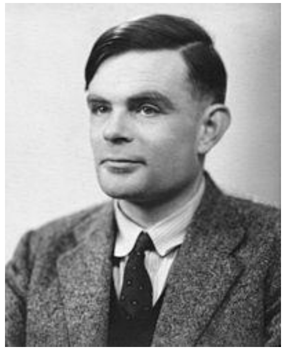


In 1950, Turing described how evolution might be used for his optimization:

building intelligent machine

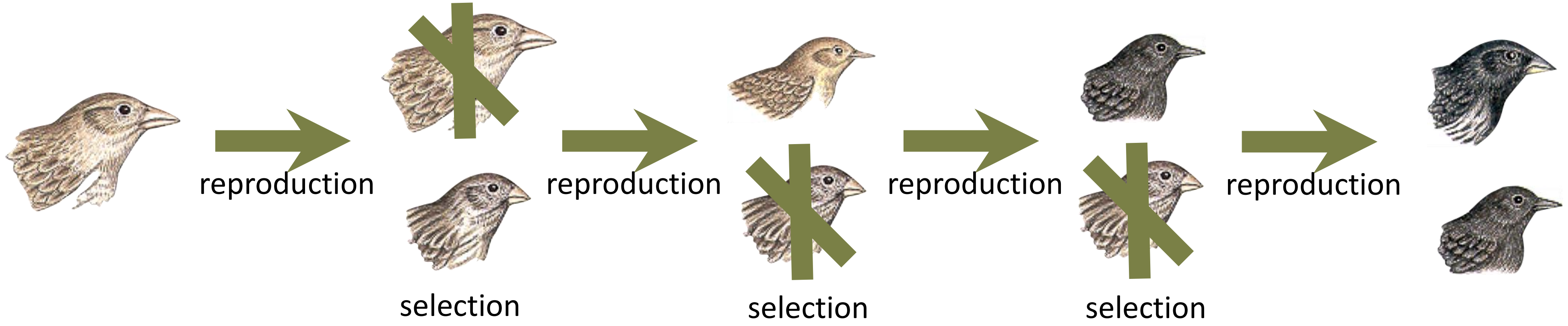
"Structure of the child machine = Hereditary material
Changes of the child machine = Mutations
Judgment of the experimenter = Natural selection"

[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, …

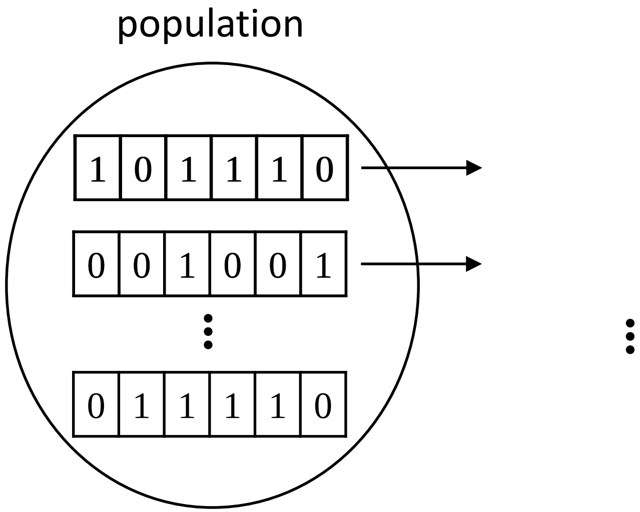
EAs also include some heuristics inspired from nature phenomena

particle swarm optimization

ant colony optimization

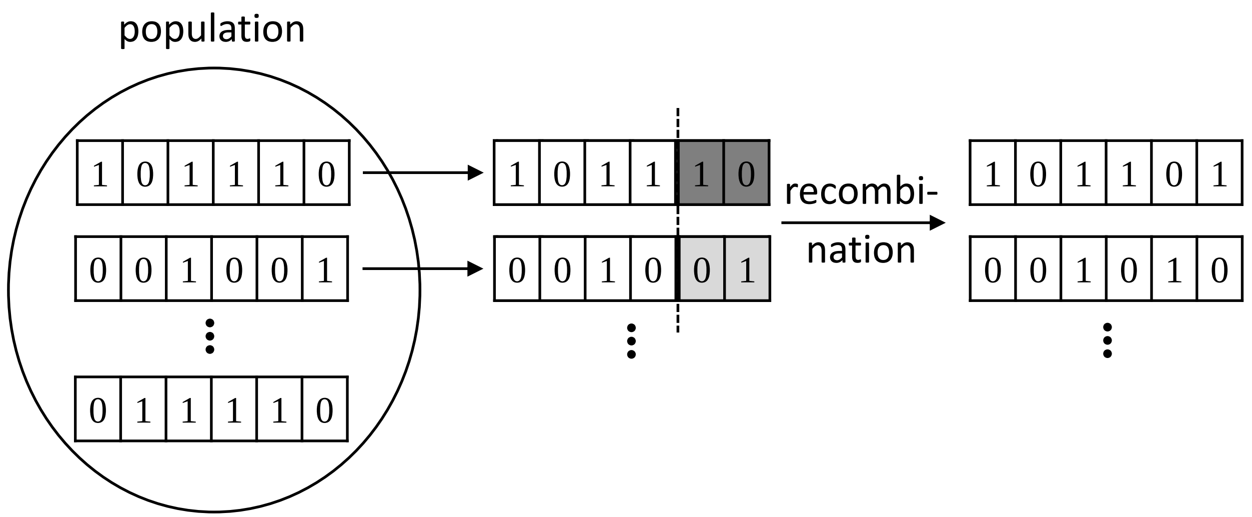
A typical evolutionary process

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



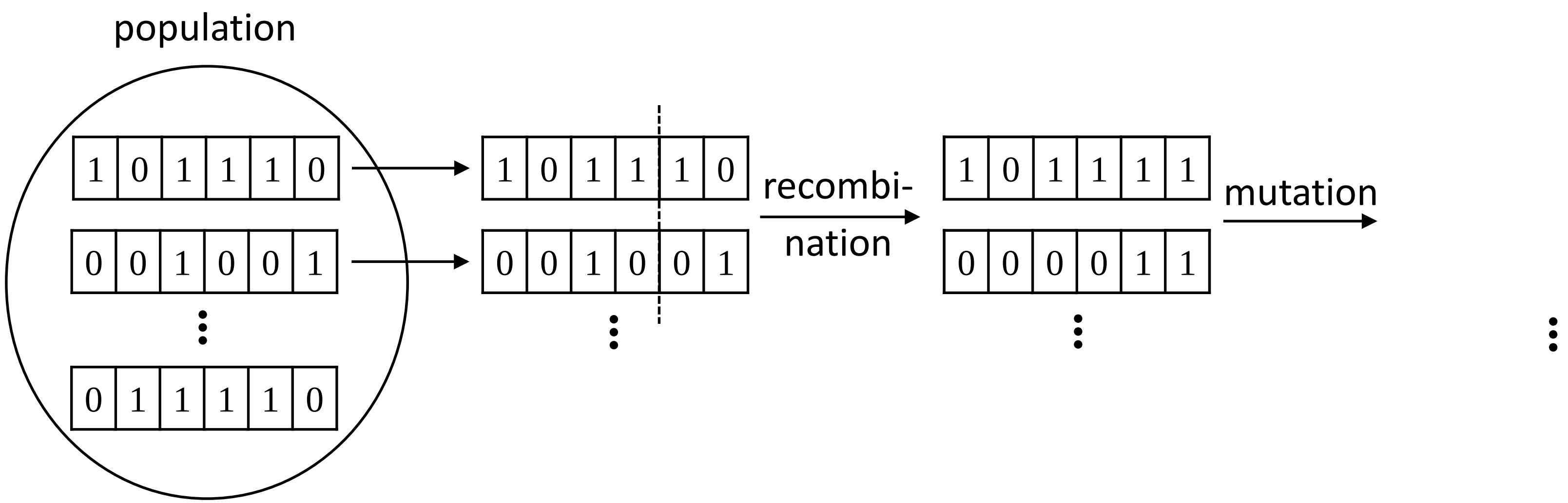
A typical evolutionary process

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



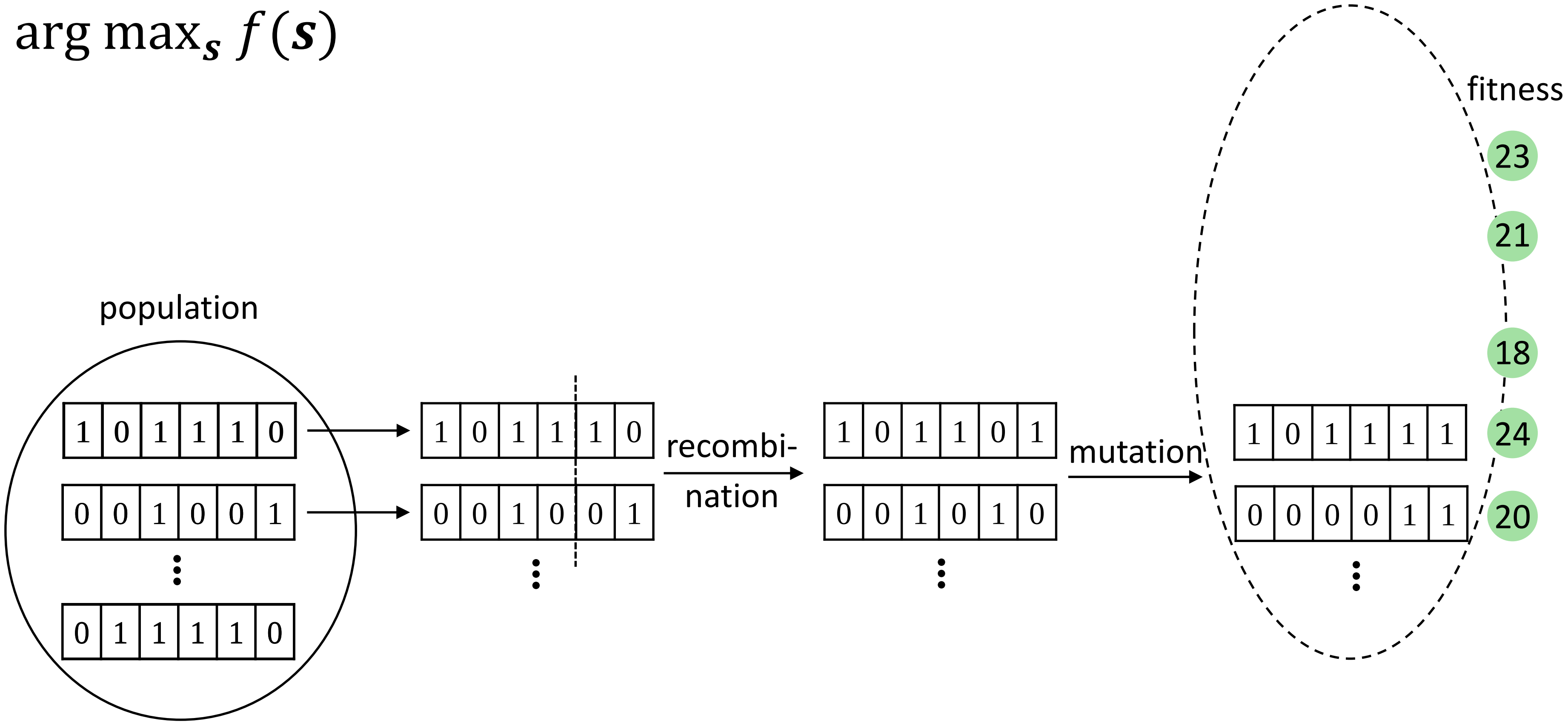
A typical evolutionary process

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



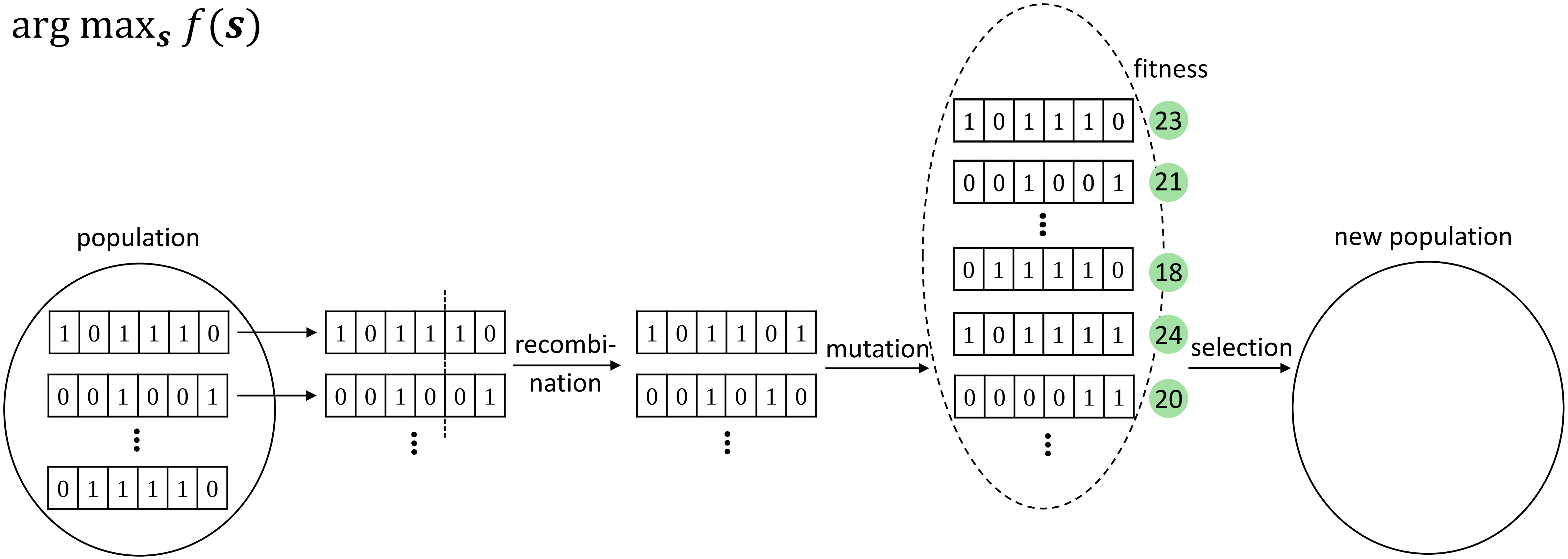
A typical evolutionary process

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



A typical evolutionary process

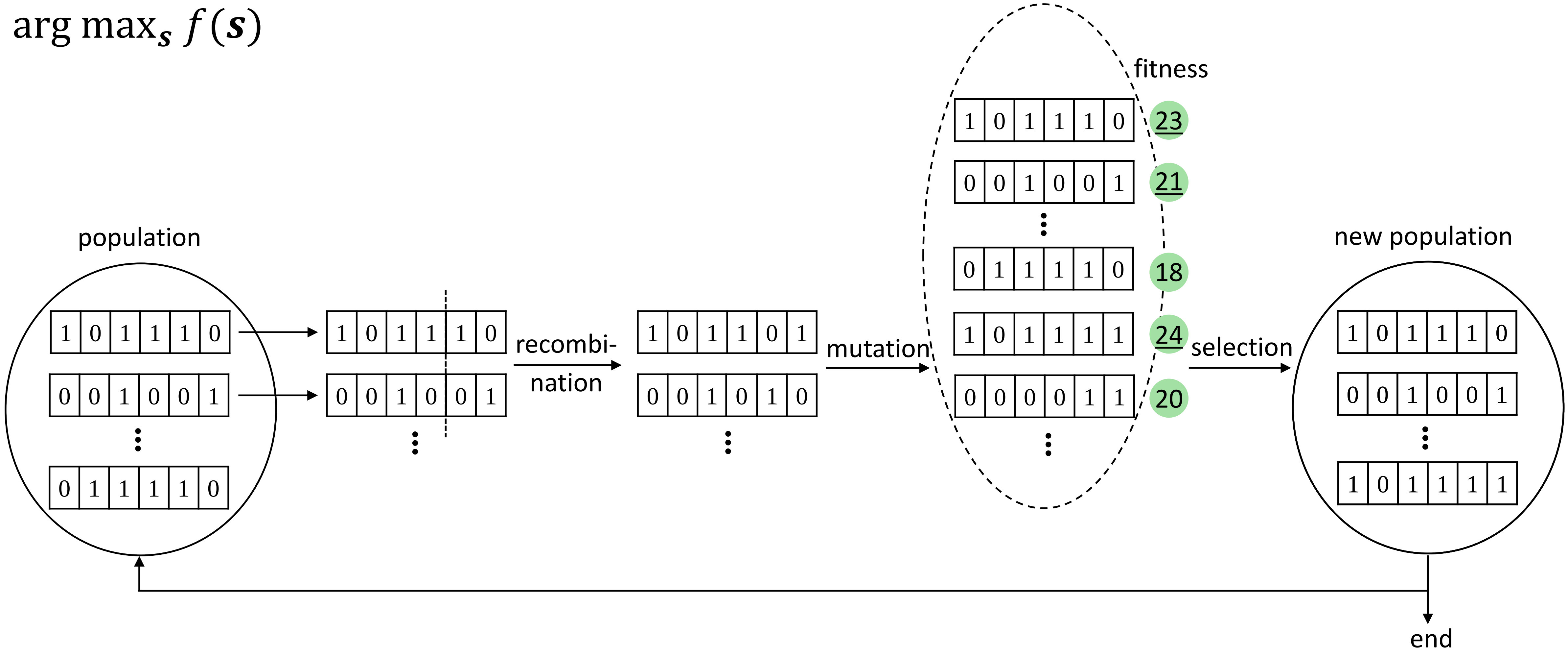
$\arg\max_{\boldsymbol{s}} f(\boldsymbol{s})$



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

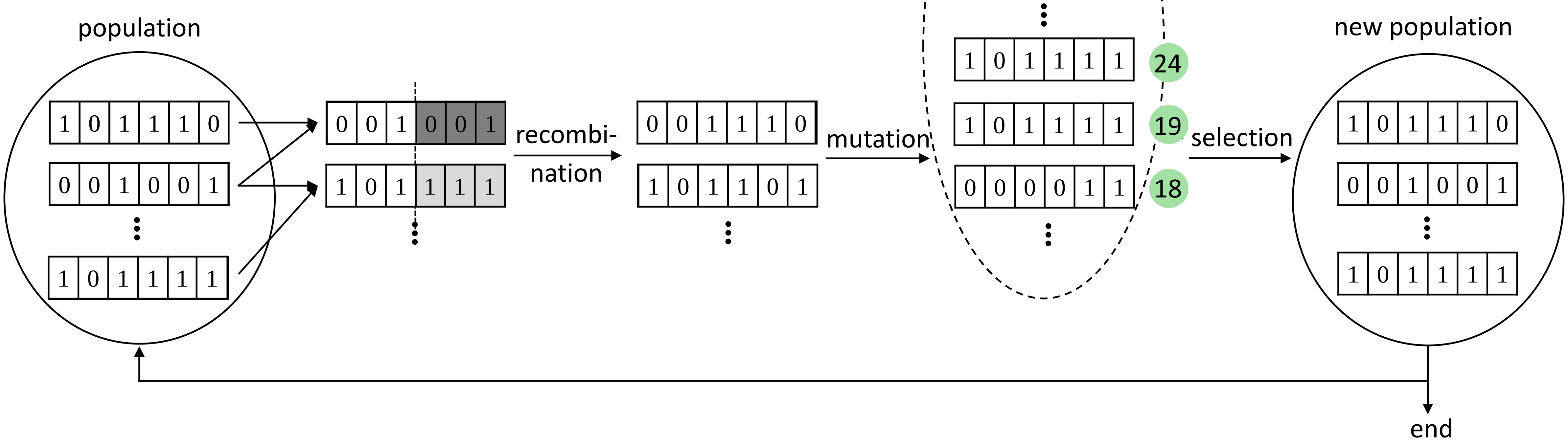
A typical evolutionary process

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



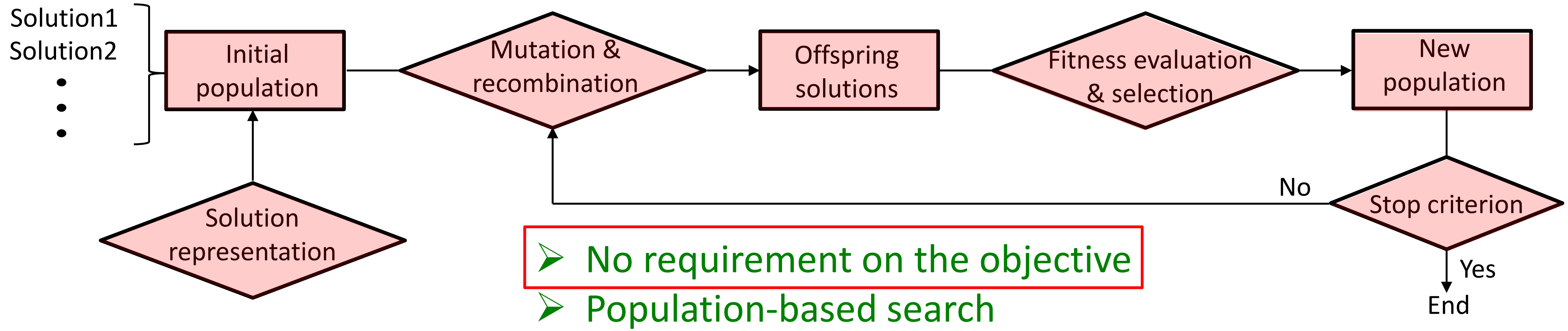
A typical evolutionary process

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



Evolutionary Algorithms

The general structure of EAs



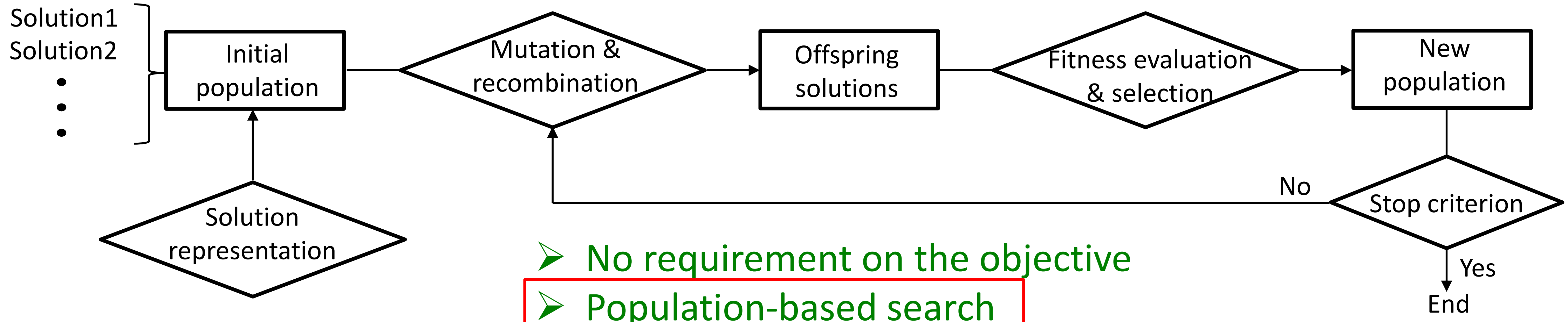
- ➢ No requirement on the objective
- ➢ Population-based search

Thus, EAs can be applied to solve complicated optimization problems

- non-differentiable, non-continuous
- without explicit objective formulation
- 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: 41646

Applications of Evolutionary Algorithms

High-speed train head design


Series 700

evolve


Series N700

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]

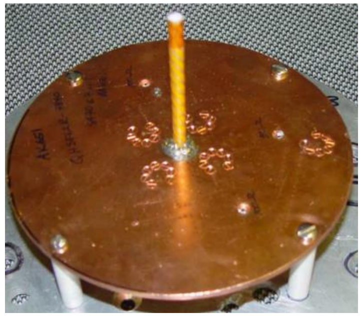
[1] *Central Japan Railway Company, Tokyo, Japan,* [2] *West Japan Railway Company, Osaka, Japan*

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.

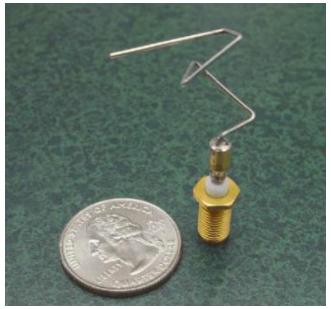
On the Tokaido Shinkansen line, Series N700 cars save 19% energy than Series 700 cars, despite a 30% increase in the output of their traction equipment for higher-speed operation (Fig. 4).

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

Antenna design


38% efficiency

evolve


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

Frances H. Arnold

Prize share: 1/2

© Nobel Media AB. Photo: A. Mahmoud

George P. Smith

Prize share: 1/4

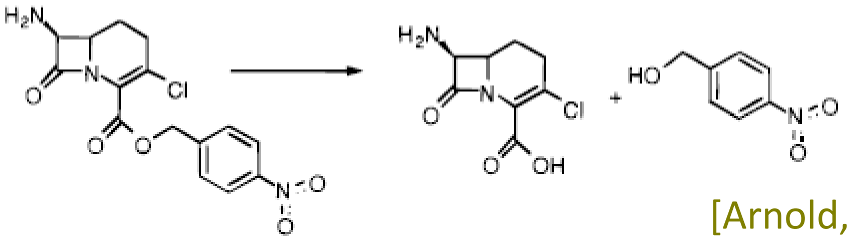
© Nobel Media AB. Photo: A. Mahmoud

Sir Gregory P. Winter

Prize share: 1/4

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

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."

Applications of Evolutionary Algorithms

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

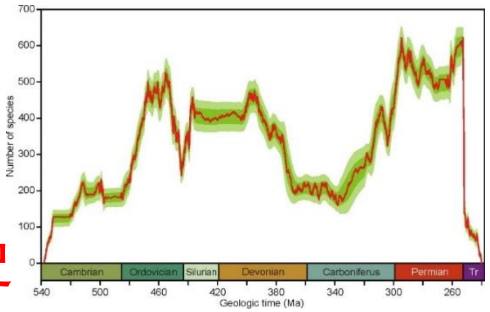
三叶虫、笔石、珊瑚、腕足…

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

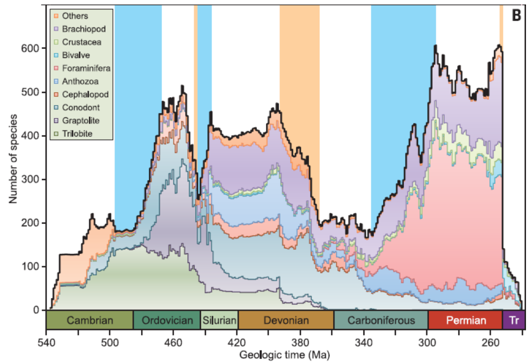
全球第一条高精度
海洋生物多样性变化曲线

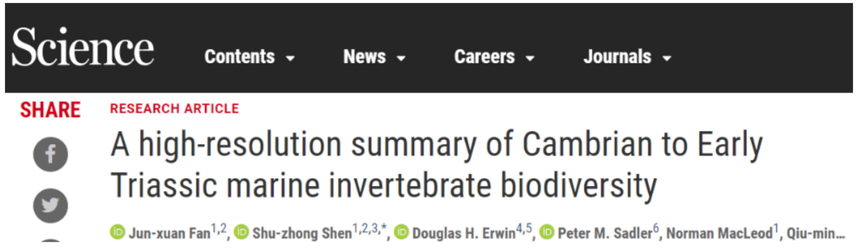
中国的地层剖面数据

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 年中国十大科技进展

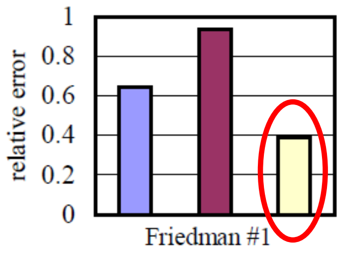
Multi-objective Evolutionary Learning

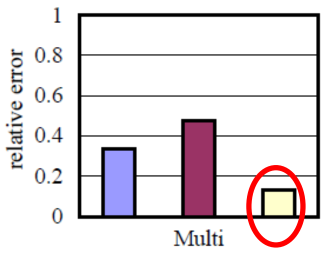
Multi-objective evolutionary learning

applies MOEAs to solve multi-objective optimization problems in machine learning

Multi-objective evolutionary learning has yielded encouraging empirical outcomes, e.g.,

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

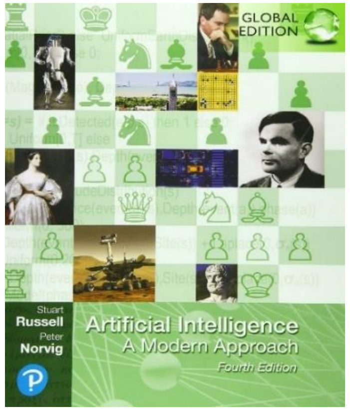
achieves competitive performance to the hand-designed models

[Real et al., ICML'17]

Why not popular?

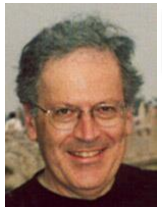
Multi-objective Evolutionary Learning

The theoretical foundation of MOEAs is underdeveloped



Artificial Intelligence: A Modern Approach

"… At present, it is not clear whether the appeal of genetic algorithms arises from their performance or from their aesthetically pleasing origins in the theory of evolution. Much work remains to be done to identify the conditions under which genetic algorithm perform well."



L. Valiant

图灵奖得主

哈佛大学教授

Evolvability

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

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

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

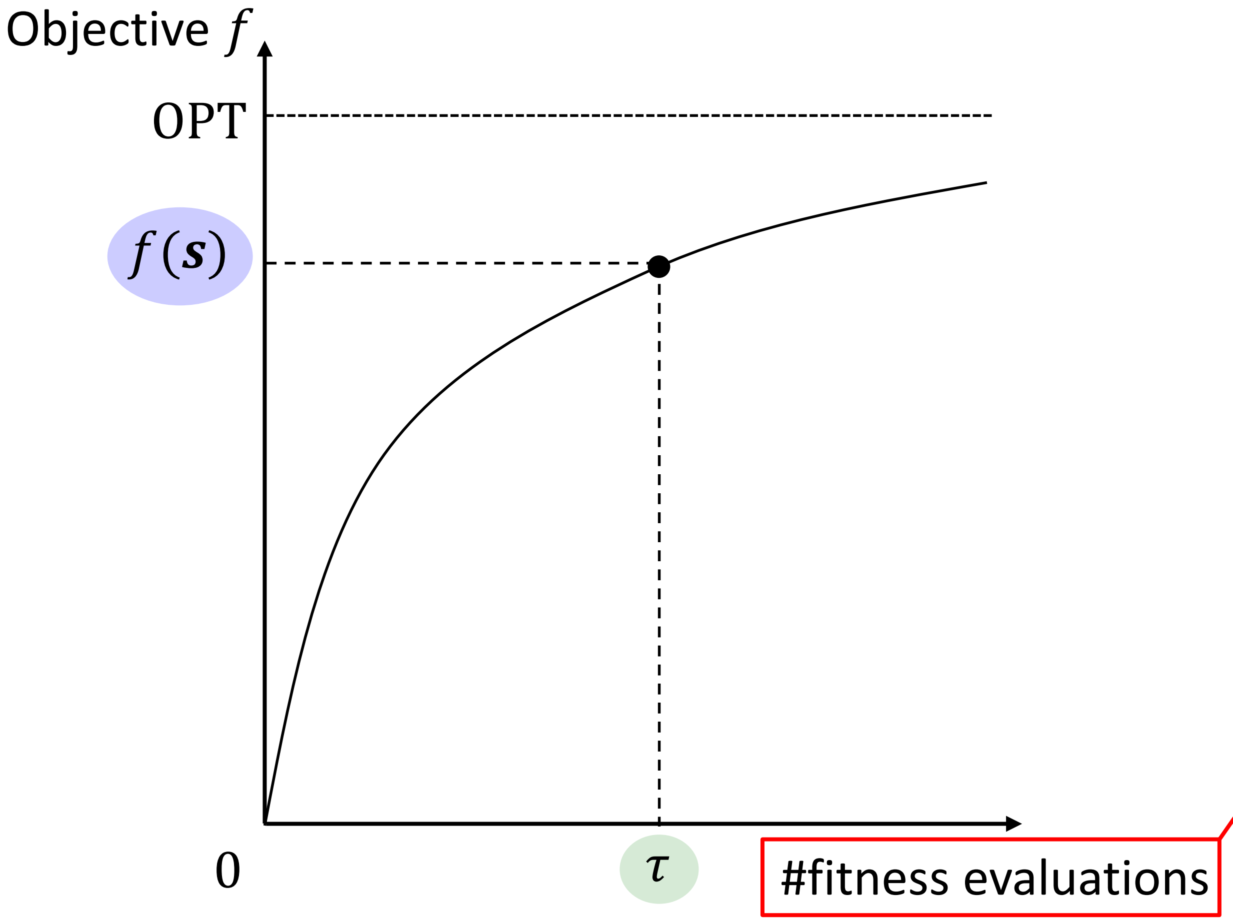
Theoretical analysis is very difficult

- MOEAs: highly randomized and complex
- Problems: complicated

Outline

- ❑ Introduction
- ❑ **Theoretical analysis tools for MOEAs**
- ❑ Theoretical perspectives of MOEAs
  - ➢ Recombination operator, constrained optimization, noisy optimization
- ❑ Multi-objective evolutionary learning algorithms
  - ➢ Selective ensemble, subset selection
- ❑ Conclusion

Running Time Complexity

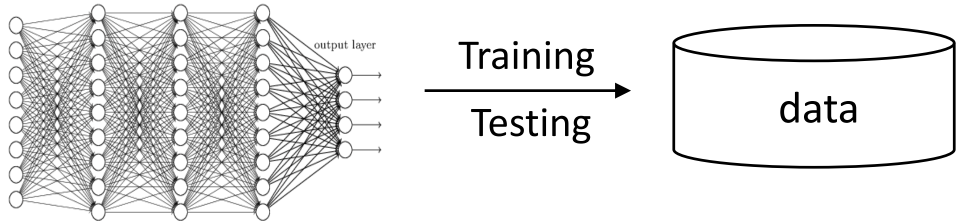


Running time τ:

#fitness evaluations until finding desired solutions for the first time

the process with the highest cost of EA

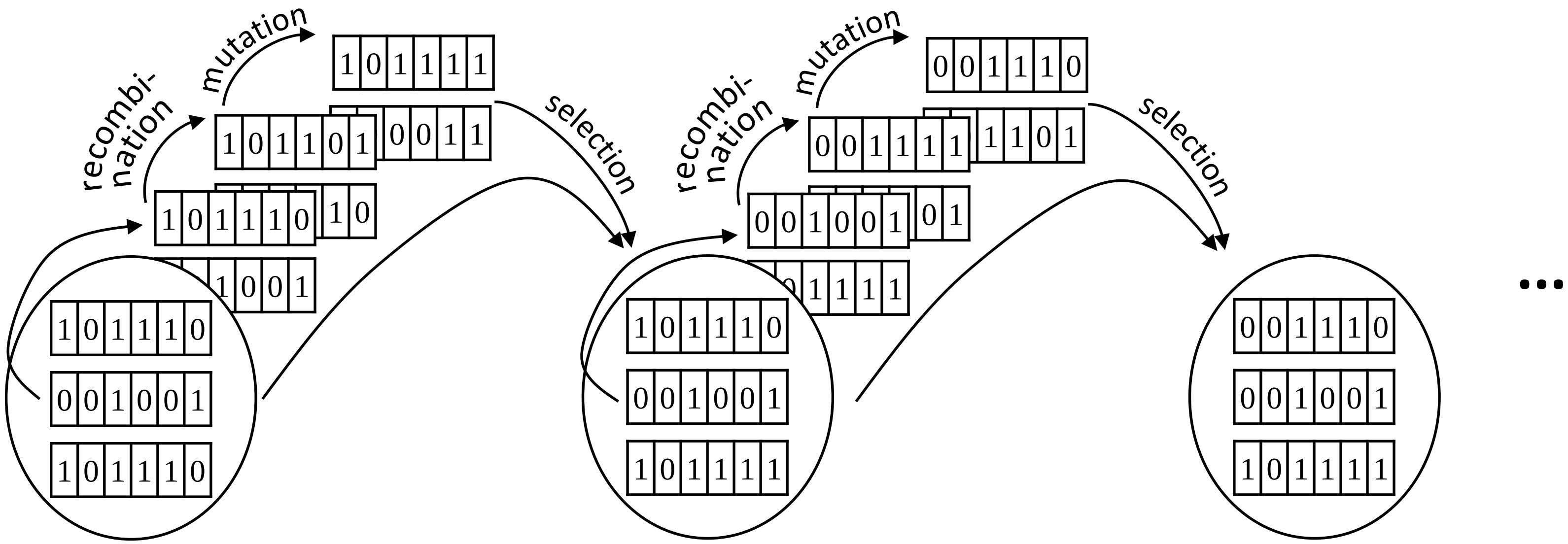
e.g., model evaluation



An Example

Hard to be analyzed directly

EA with recombination



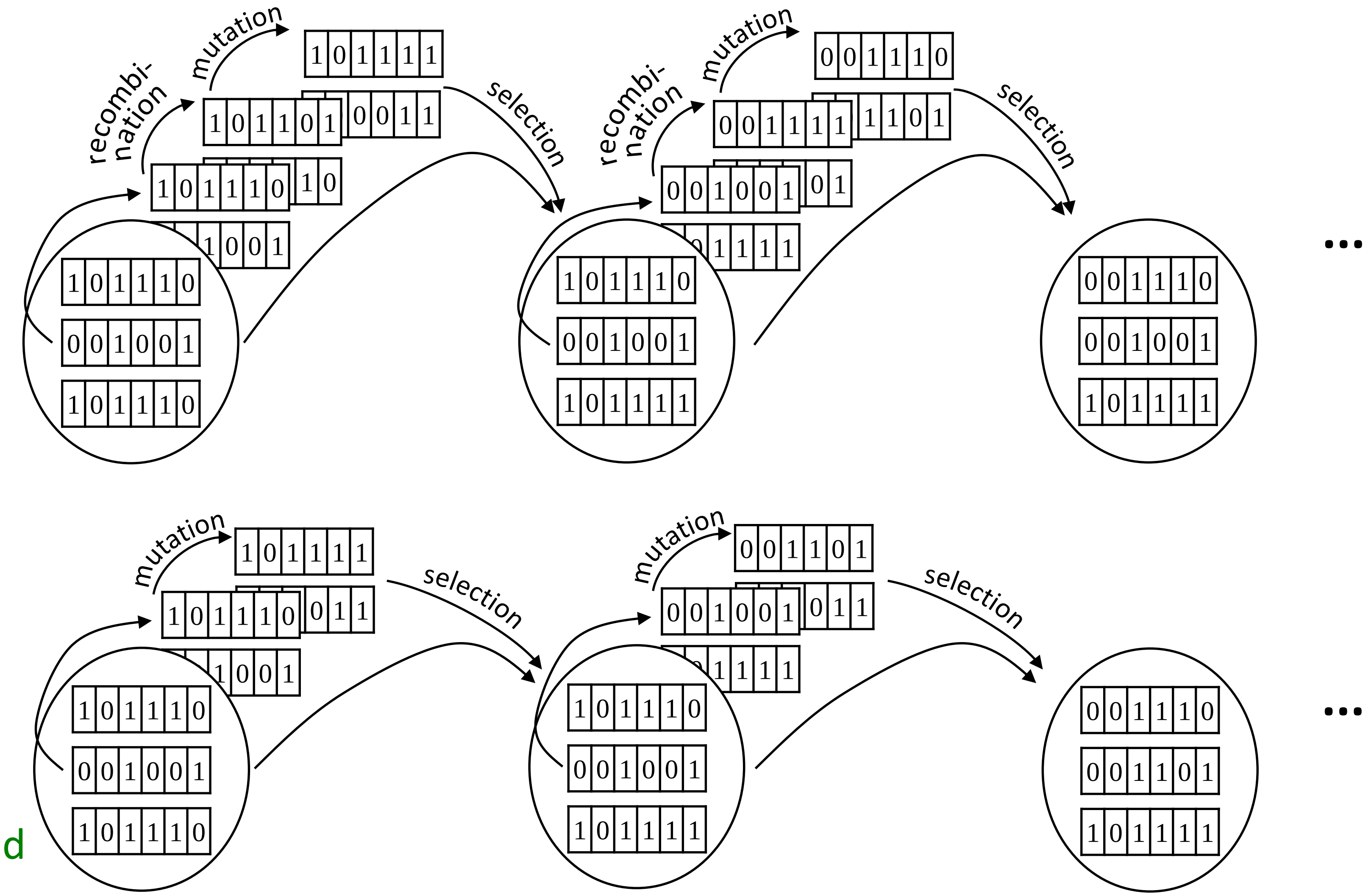
An Example

Hard to be analyzed directly

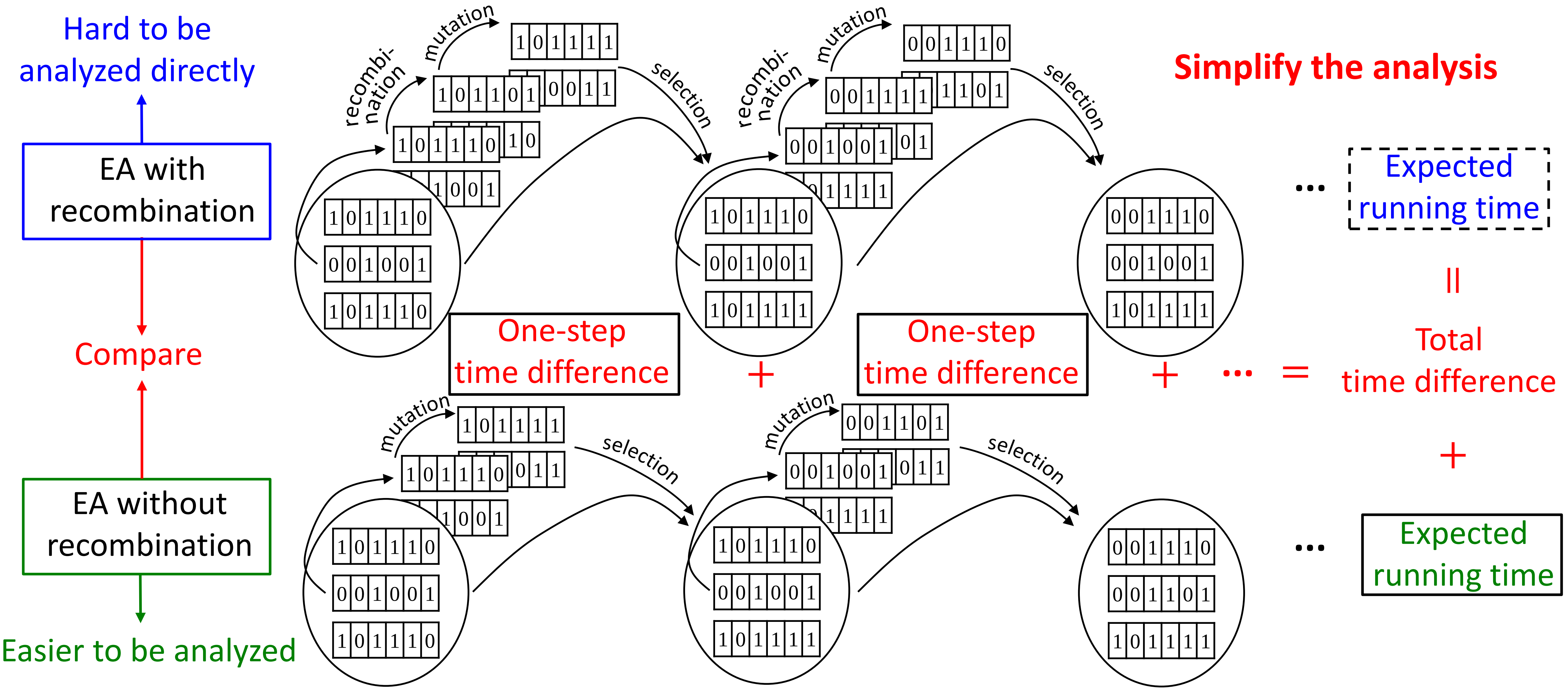
EA with recombination

EA without recombination

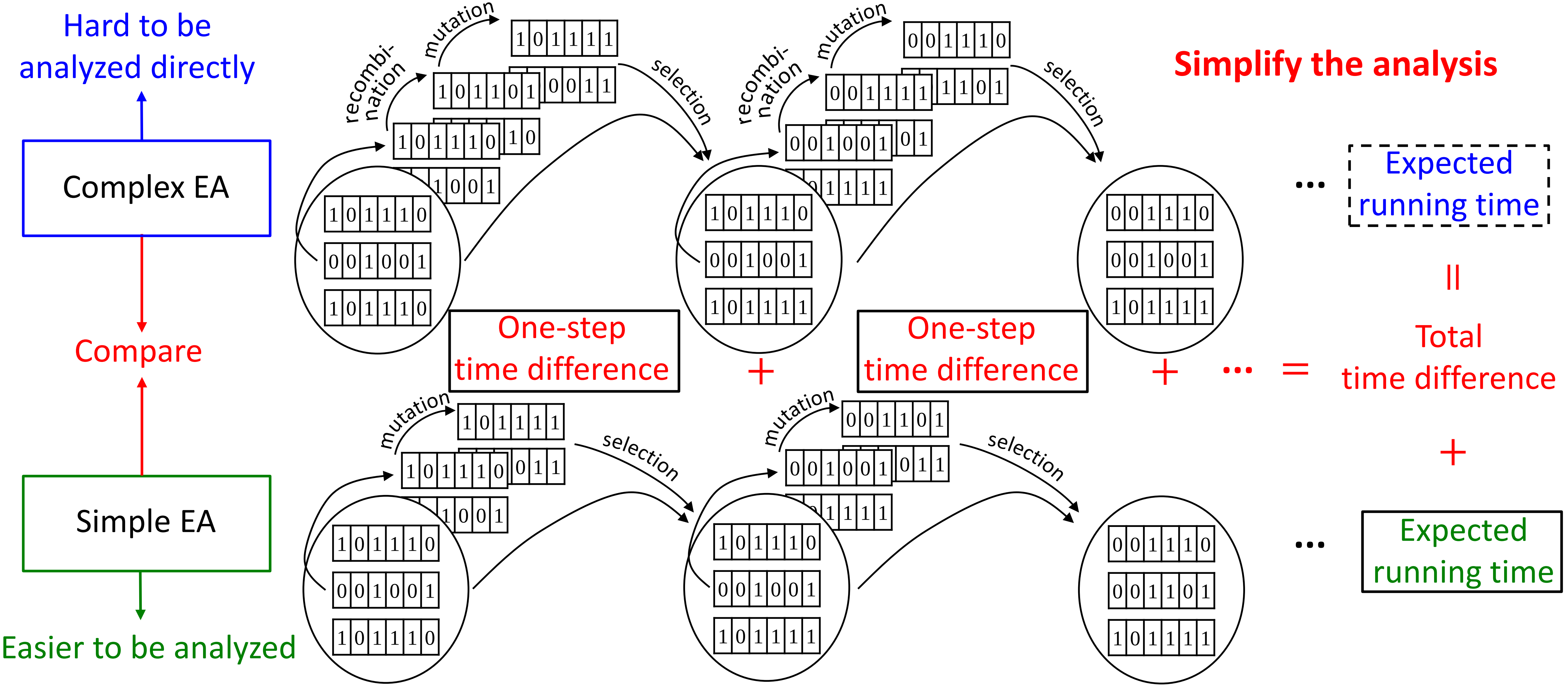
Easier to be analyzed



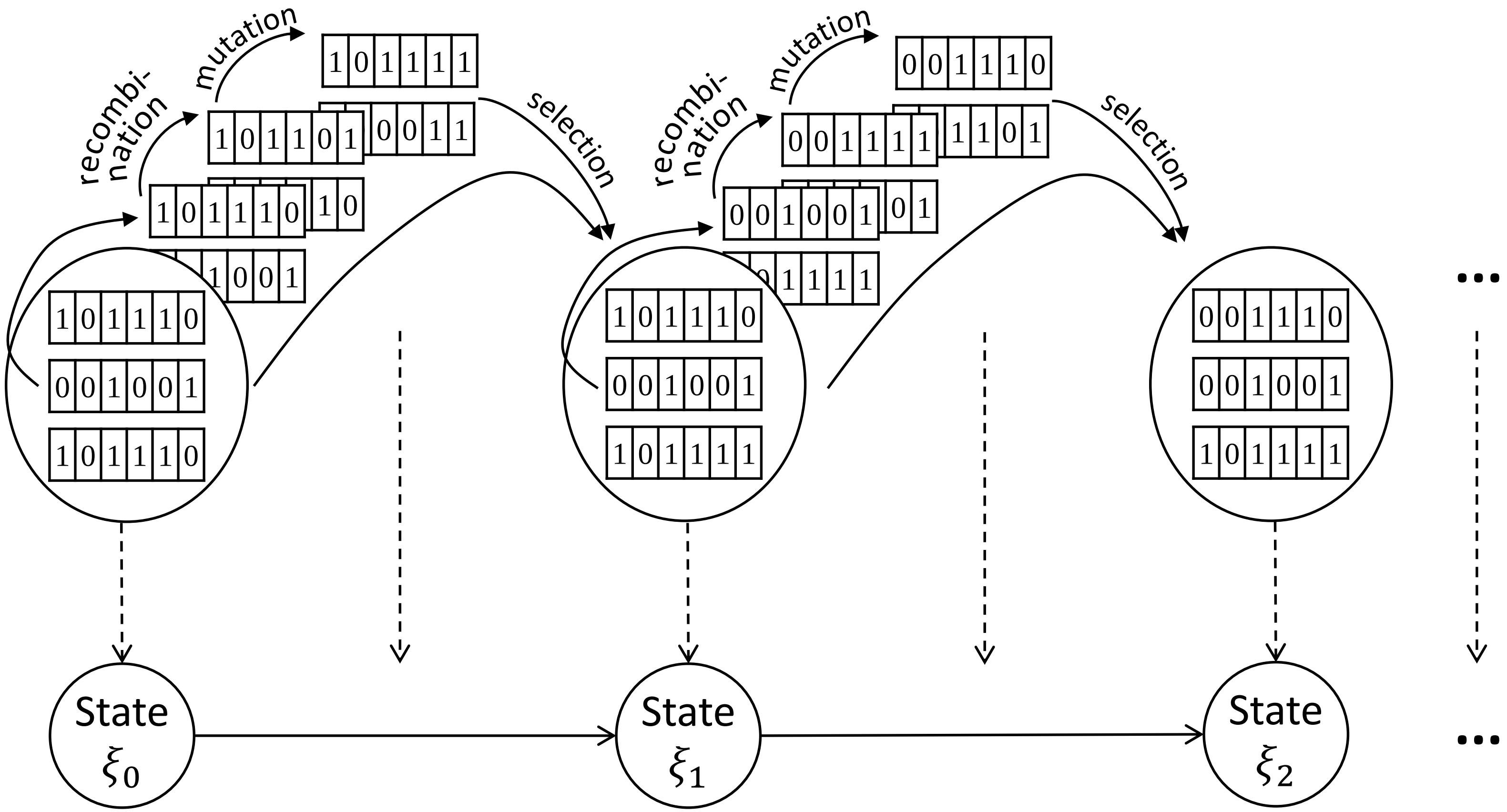
An Example


Hard to be analyzed directly
EA with recombination
Compare
EA without recombination
Easier to be analyzed
recombi-nation
mutation
selection
One-step time difference
+
One-step time difference
+
…
=
Total time difference
Simplify the analysis
Expected running time
=
+
Expected running time
101111
101101
0011
101110
10
1001
101110
001001
001110
001111
1101
001001
01
1111
101111
001101
011

An Example


Hard to be analyzed directly
Complex EA
Compare
Simple EA
Easier to be analyzed
recombi- nation
mutation
selection
1 0 1 1 1 0
0 0 1 0 0 1
1 0 1 1 1 0
1 0 1 1 1 1
0 0 1 0 0 1
1 0 1 1 1 1
0 0 1 1 1 0
0 0 1 0 0 1
1 0 1 1 1 1
0 0 1 1 0 1
One-step time difference
+
...
=
Simplify the analysis
Expected running time
=
Total time difference
+
Expected running time

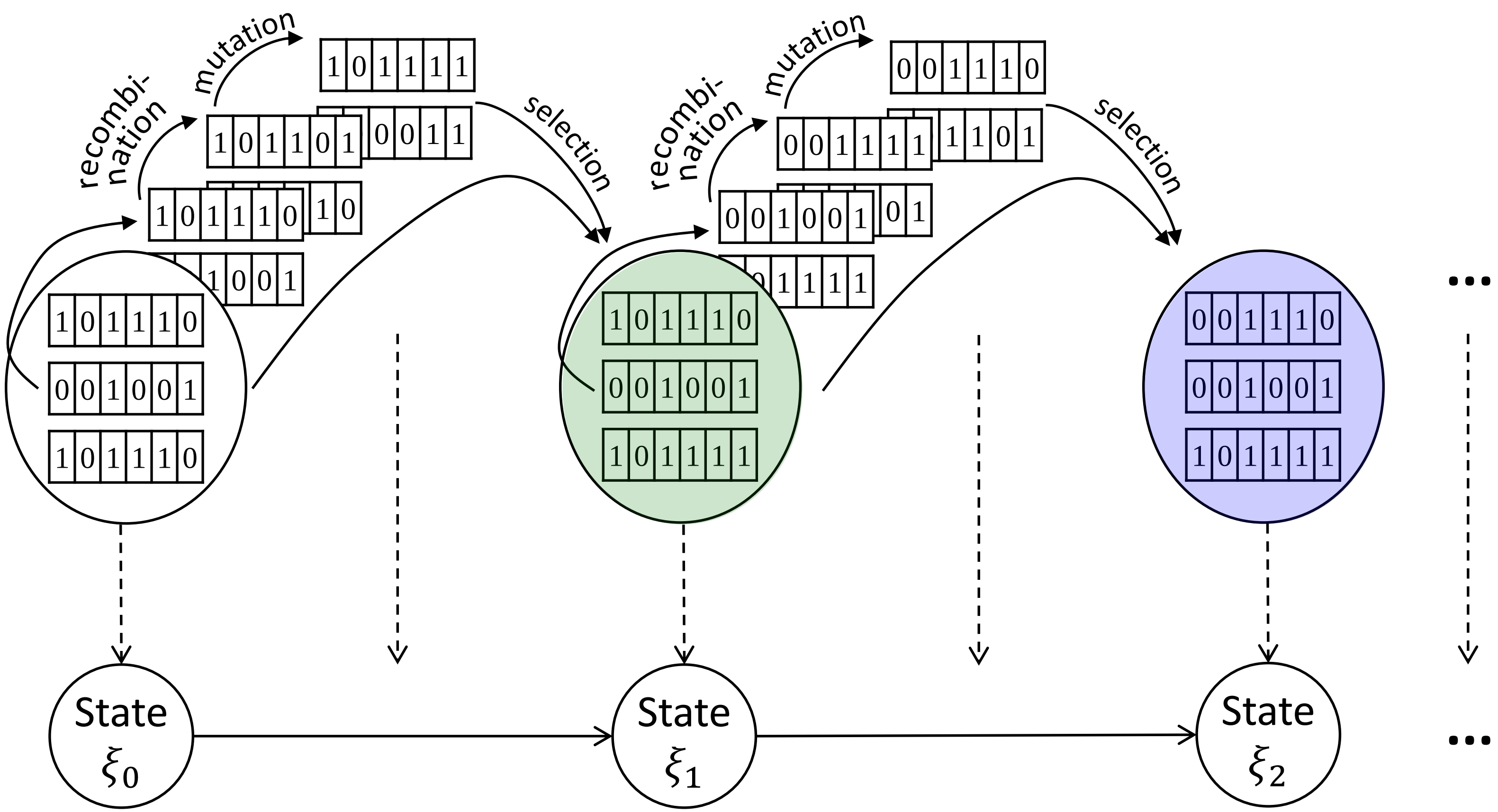
Model an EA process as a Markov chain



Markov chain?

Switch Analysis

Model an EA process as a Markov chain



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

Markov property

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

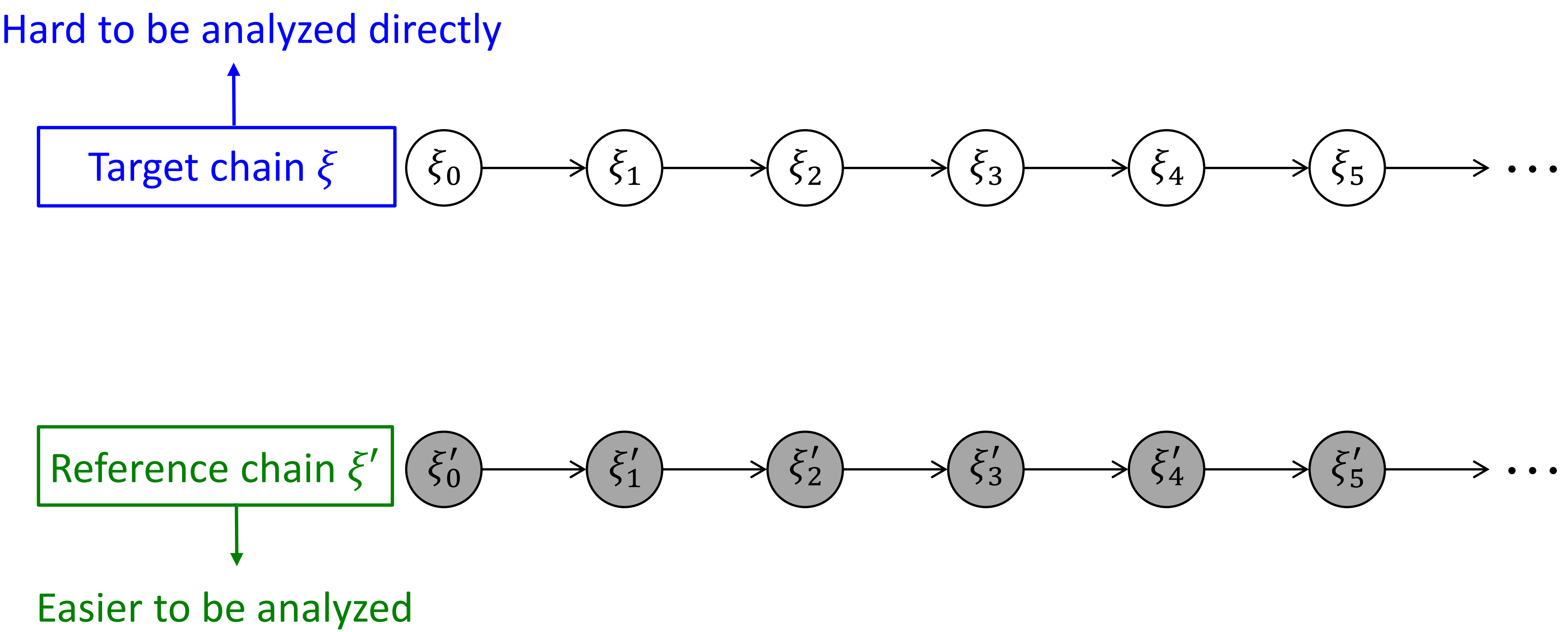
Switch Analysis

Hard to be analyzed directly

$\uparrow$

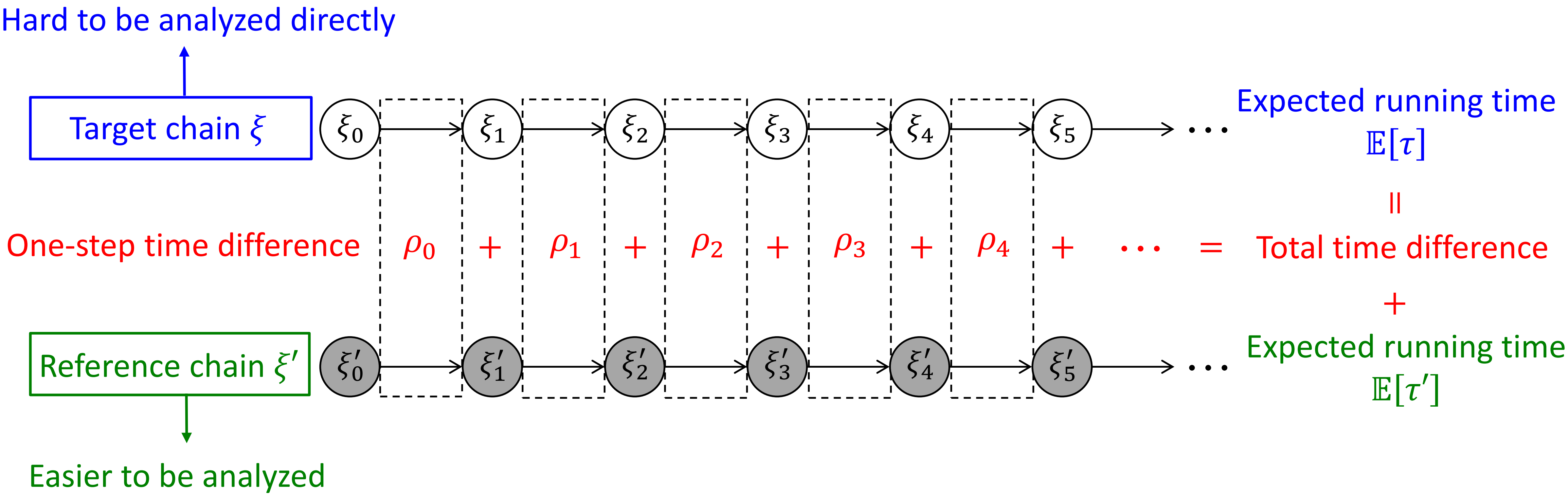
Target chain ξ $\quad \xi_0 \rightarrow \xi_1 \rightarrow \xi_2 \rightarrow \xi_3 \rightarrow \xi_4 \rightarrow \xi_5 \rightarrow \cdots$

Switch Analysis



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

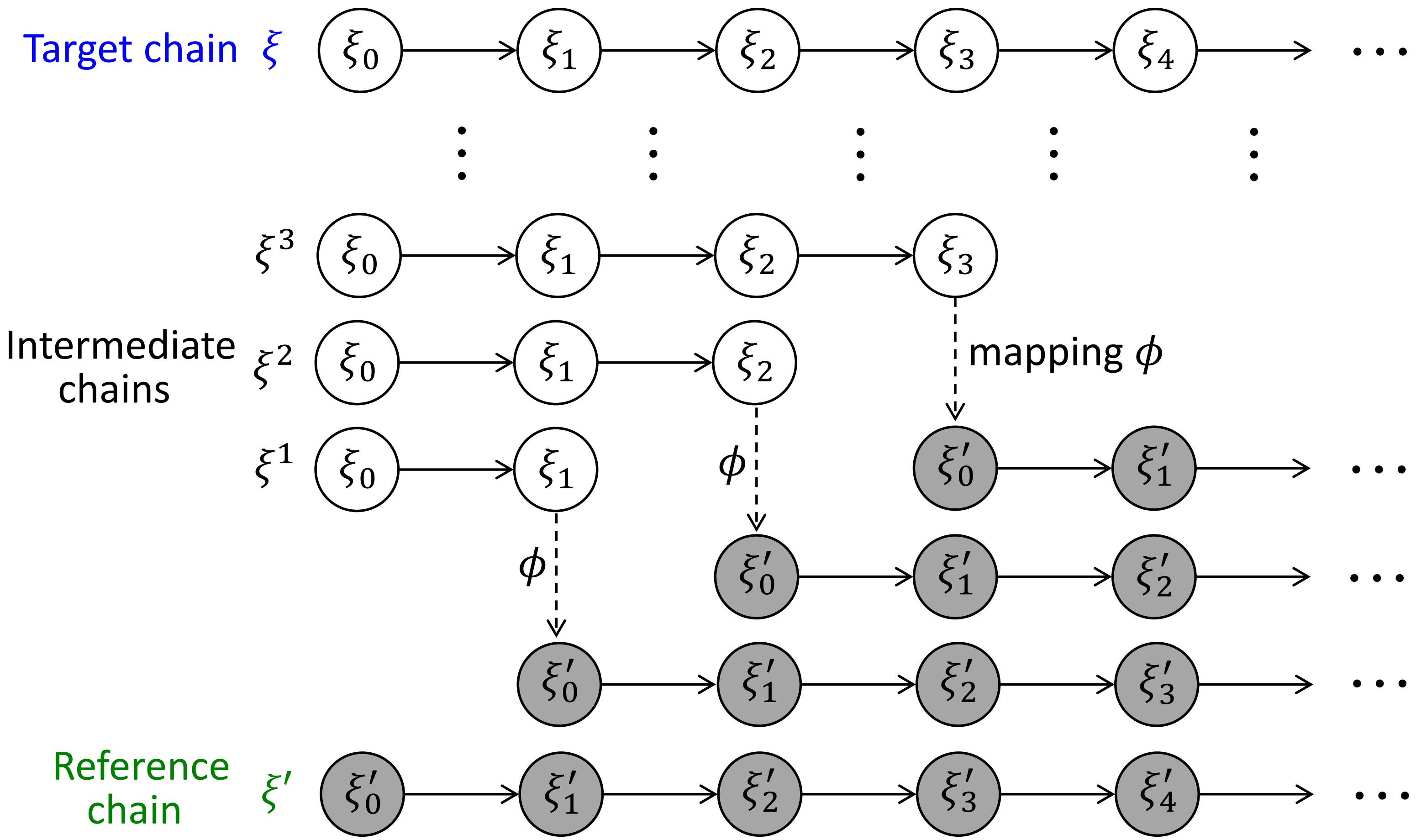
Switch Analysis



How to estimate one-step time difference ρ_t?

Switch Analysis

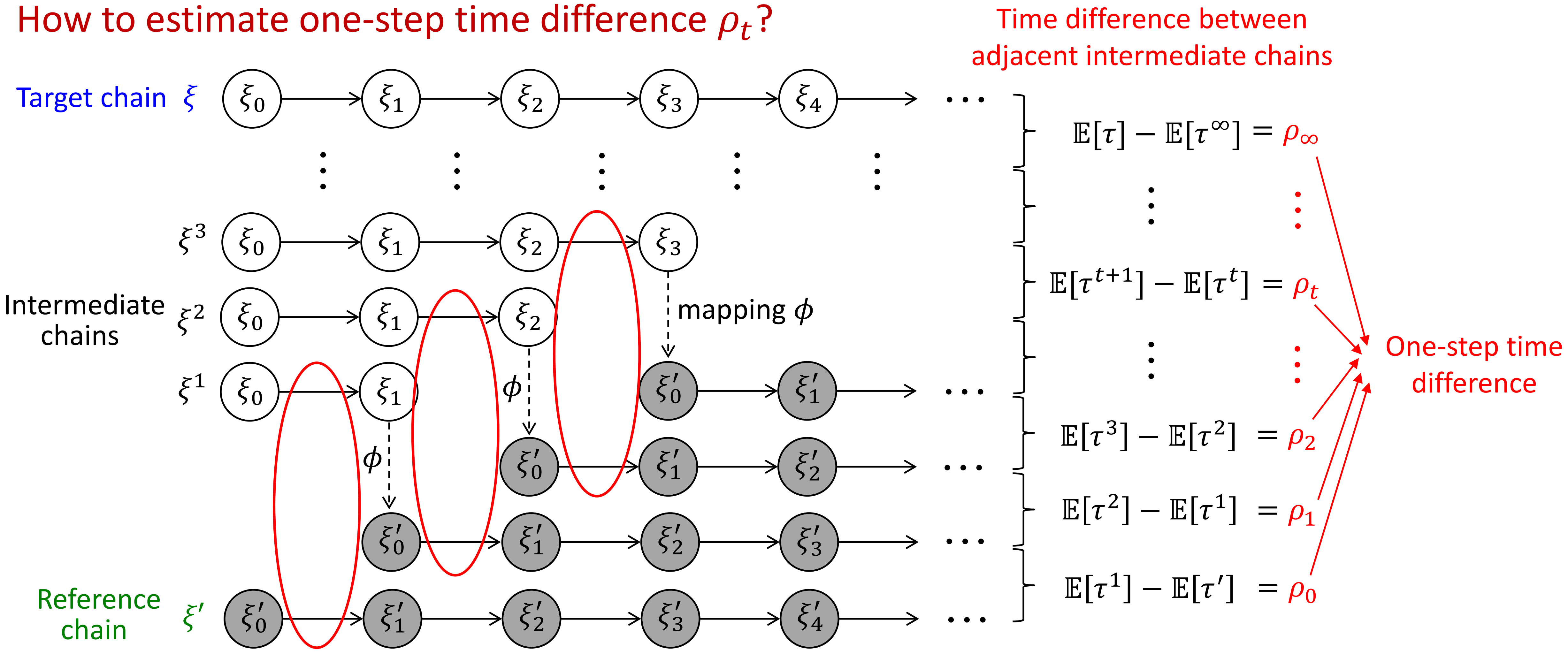
How to estimate one-step time difference ρ_t?



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

Switch Analysis

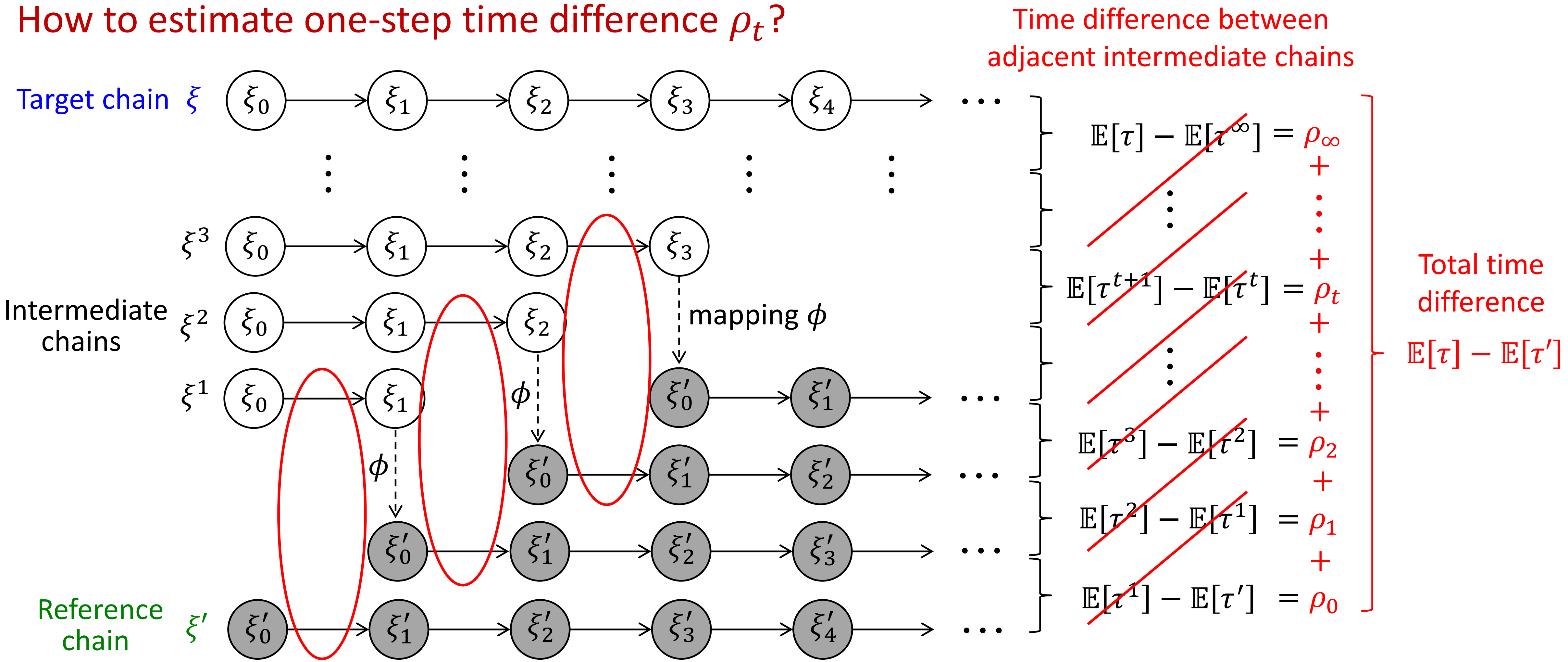
How to estimate one-step time difference ρ_t?



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

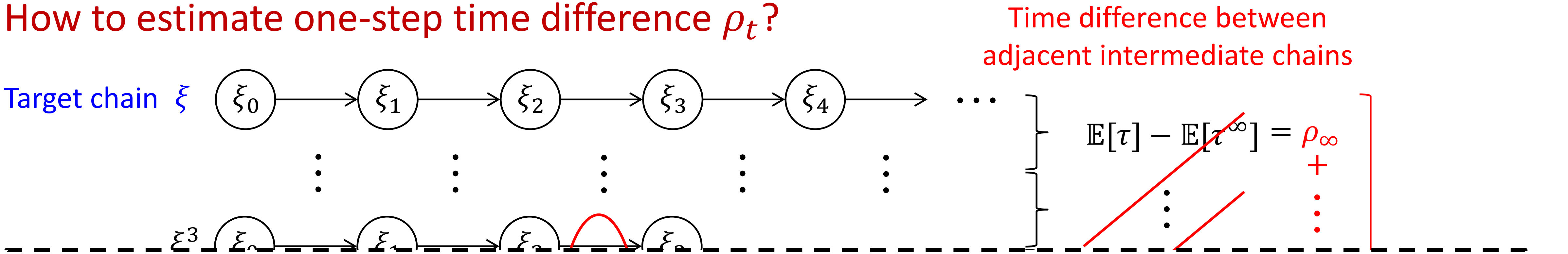
Switch Analysis

How to estimate one-step time difference ρ_t?

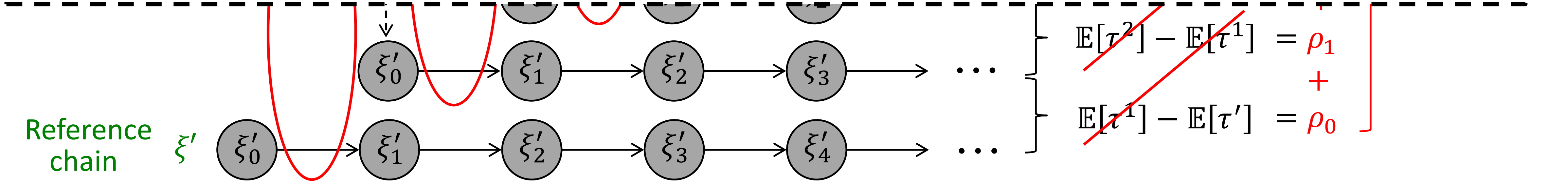


Switch Analysis

How to estimate one-step time difference ρ_t?



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* $\geq$) $\sum_{\mathrm{u}, \mathrm{y} \in \mathcal{Y}} \pi_t^\phi(\mathrm{u}) P(\xi'_1 = \mathrm{y} \mid$ ... $\tau']$



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

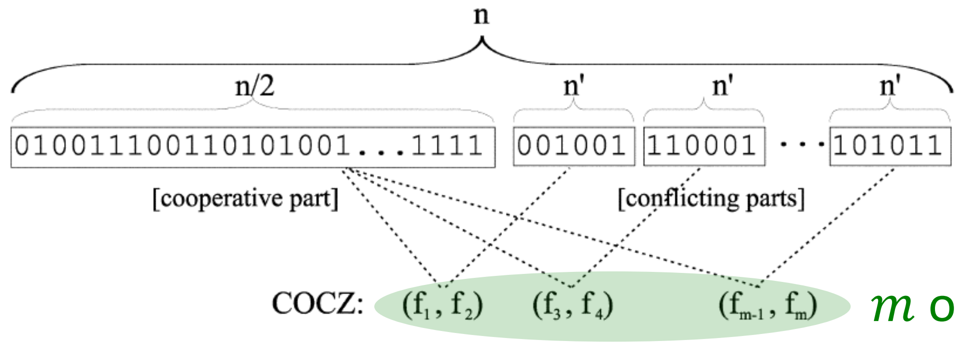
Application of Switch Analysis

Example: Analyze GSEMO solving the mCOCZ problem

GSEMO:

1. $\boldsymbol{s} :=$ randomly selected from $\{0,1\}^n$; $P := \{\boldsymbol{s}\}$
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 : $\max_{\boldsymbol{s}\in\{0,1\}^n} \ (f_1(\boldsymbol{s}), f_2(\boldsymbol{s}), \dots, f_m(\boldsymbol{s}))$



m objectives

L. Thiele
Professor, ETH Zurich
Member of Academia Europaea

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]

Outline

- ❑ Introduction
- ❑ Theoretical analysis tools for MOEAs
- ❑ **Theoretical perspectives of MOEAs**
  - ➢ **Recombination operator, constrained optimization, noisy optimization**
- ❑ Multi-objective evolutionary learning algorithms
  - ➢ Selective ensemble, subset selection
- ❑ Conclusion

Recombination

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

Recombination

Most theoretical studies focused on EAs with mutation, while only a few included recombination, which is difficult to be analyzed due to the irregular behavior

J. Holland
UMich, USA
Pioneer of EC
building block hypothesis
1975

J. Rowe
UOB, UK
AE of TCS
“building block” fails
2002

B. Doerr
EP, France
AE of AIJ
recombination helpful
2008

P. S. Oliveto
UOS, UK
AE of TEC
recombination not work
2014

D. Sudholt
UOP, Germany
AE of ECJ
recombination helpful
2017

Mainly focused on single-objective optimization

How about the influence of recombination for multi-objective optimization?

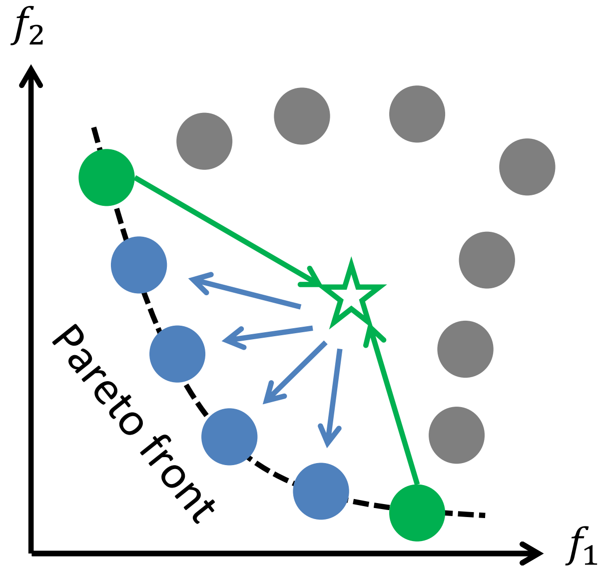
- Often involved in machine learning
- More complex than single-objective optimization

Recombination

Our result:

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

Unique to multi-objective optimization



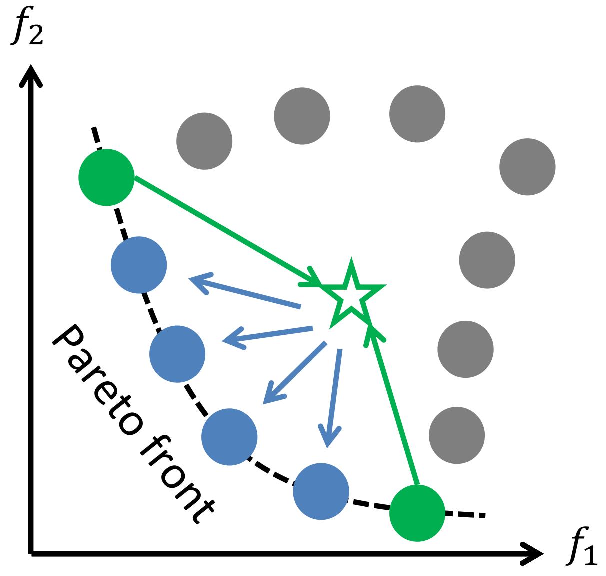
[Qian et al., Artificial Intelligence 2013, ACM GECCO'11 Best Theory Paper Award]

Recombination

Our result:

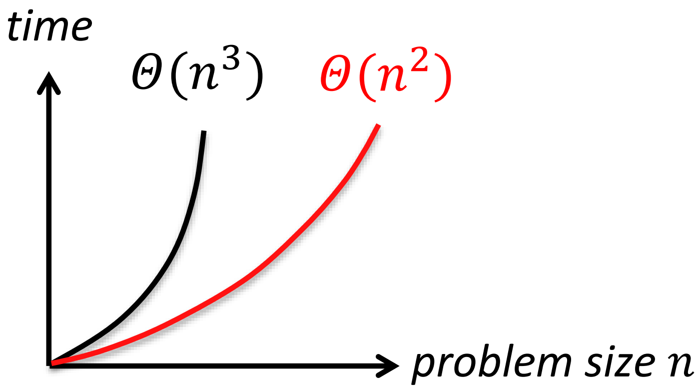
Recombination can accelerate the filling of the Pareto front by recombining diverse Pareto optimal solutions

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]

Constrained Optimization

The optimization problems in machine learning often come with constraints

e.g., to avoid overfitting, one often needs to minimize the error of a model, while constraining the model complexity

General formulation of constrained optimization:

$$\min_{\boldsymbol{s}\in S} \quad \boxed{f(\boldsymbol{s})} \rightarrow \text{objective function}$$
$$\text{s.t.} \quad \boxed{g_i(\boldsymbol{s}) = 0, \quad 1 \le i \le q;} \rightarrow \text{equality constraints}$$
$$\boxed{h_i(\boldsymbol{s}) \le 0, \quad q+1 \le i \le m} \rightarrow \text{inequality constraints}$$

The goal is to find a feasible solution minimizing the objective f

Remark: A solution is (in)feasible if it does (not) satisfy the constraints

How to deal with constraints for EAs?

The penalty function method transforms the original constrained optimization problem into an unconstrained optimization problem [Hadj-Alouane and Bean, OR'97]

constrained

$$\begin{aligned} \min \quad & f(\boldsymbol{s}) \\ \mathrm{s.t.} \quad & g_i(\boldsymbol{s}) = 0, \quad 1 \le i \le q; \\ & h_i(\boldsymbol{s}) \le 0, \quad q+1 \le i \le m \end{aligned}$$

$\Rightarrow$

unconstrained

$$\min \ f(\boldsymbol{s}) + \lambda \sum_{i=1}^{m} \boxed{f_i(\boldsymbol{s})}$$

constraint violation degree

$$f_i(\boldsymbol{s}) = \begin{cases} |g_i(\boldsymbol{s})| & 1 \le i \le q \\ \max\{0, h_i(\boldsymbol{s})\} & q+1 \le i \le m \end{cases}$$

How to deal with constraints for EAs?

Pareto optimization transforms the original constrained optimization problem into a bi-objective optimization problem [Coello Coello, 2002]

Constrained optimization: *optimize f under some constraints*

Transform

Bi-objective optimization: *optimize f and a constraint-related objective simultaneously*

An example

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

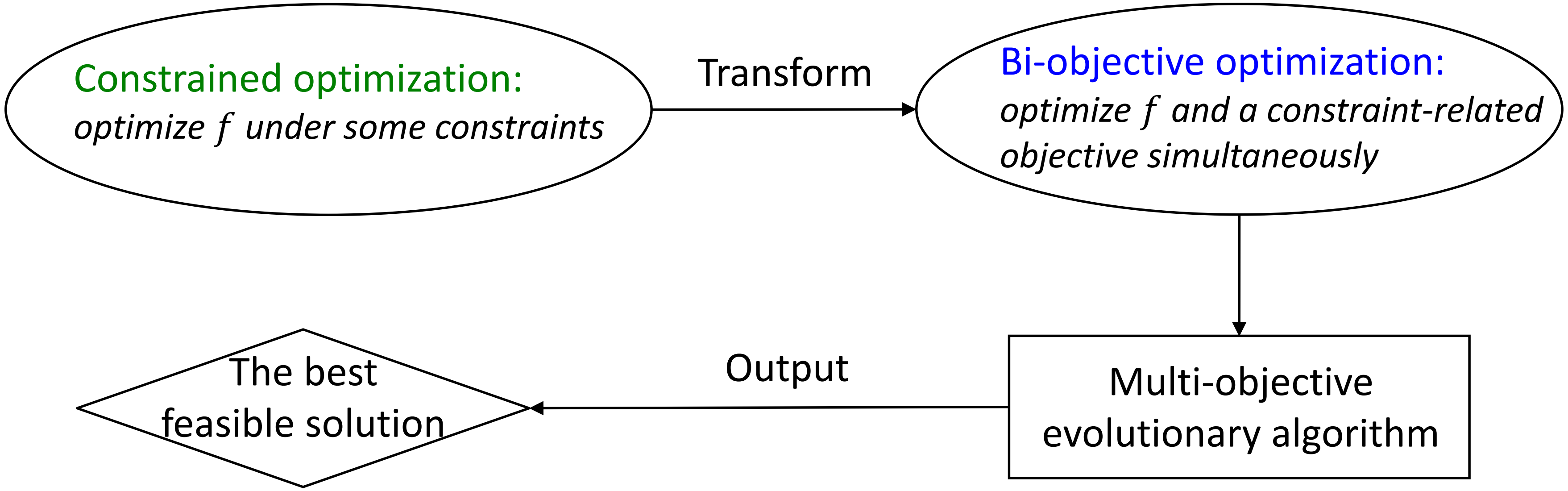
$$\Rightarrow \quad \min \ \left(f(\boldsymbol{s}),\ \textstyle\sum_{i=1}^{m} \boxed{f_i(\boldsymbol{s})}\right)$$

constraint violation degree

$$f_i(\boldsymbol{s}) = \begin{cases} |g_i(\boldsymbol{s})| & 1 \le i \le q \\ \max\{0, h_i(\boldsymbol{s})\} & q+1 \le i \le m \end{cases}$$

How to deal with constraints for EAs?

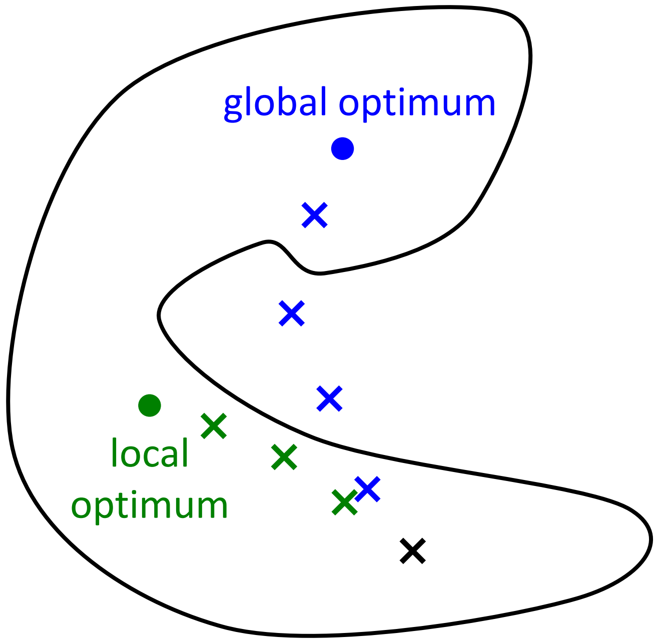
Pareto optimization transforms the original constrained optimization problem into a bi-objective optimization problem [Coello Coello, 2002]



Constrained Optimization

Our result: Pareto optimization can be better by exploiting infeasible solutions

- Penalty function
  - prefers feasible solutions
  - if initialized far from the global optimum, easy to get trapped by local optimum
- Pareto optimization
  - allows infeasible solutions to participate in the evolutionary process naturally
  - follows a shortcut from infeasible space to feasible space to find good solutions



[Qian, Yu and Zhou, IJCAI 2015]

Constrained Optimization

Our result: Pareto optimization can be better by exploiting infeasible solutions

Example: Minimum set cover problem **One of Karp's 21 NP-complete problems**

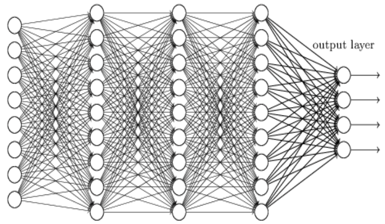
	Expected running time
Penalty function	exponential
Pareto optimization	$O(mn(\log n + \log w_{max} + m))$

[Qian, Yu and Zhou, IJCAI 2015]

Noisy Optimization

The objective (i.e., fitness) evaluation in machine learning is often disturbed by noise

model evaluation



Training → Testing

limited data

- Exact fitness value: $f(\boldsymbol{s})$
- Noisy fitness value: $f^{\mathrm{n}}(\boldsymbol{s})$ e.g., $f(\boldsymbol{s}) + N(0, \sigma^2)$

How to reduce the negative influence of noise?

Threshold selection [Markon et al., CEC'01]

accepts an offspring solution only if its fitness becomes better by at least a threshold τ

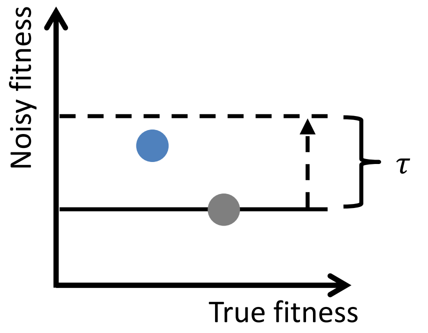
$$f^{\mathrm{n}}(\boldsymbol{s}) > f^{\mathrm{n}}(\boldsymbol{s}') \quad \Longrightarrow \quad f^{\mathrm{n}}(\boldsymbol{s}) > f^{\mathrm{n}}(\boldsymbol{s}') + \tau$$

Its effectiveness is not yet clear

Noisy Optimization

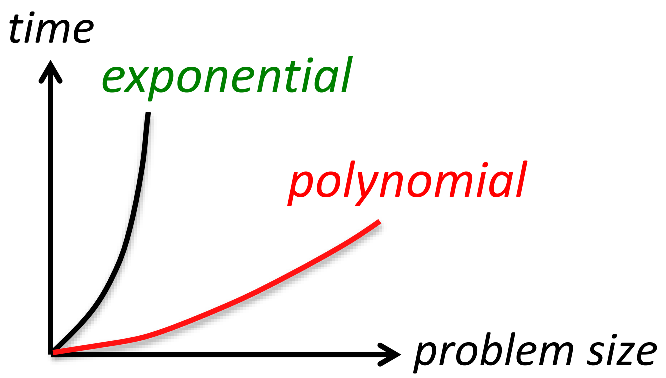
Our result: Threshold selection can bring robustness against noise

reduces the risk of deleting a good solution



Example: (1+1)-EA solving the OneMax problem under noise

Expected running time: exponential $\xrightarrow[\text{selection}]{\text{threshold}}$ polynomial



[Qian, Yu and Zhou, Evolutionary Computation 2018]

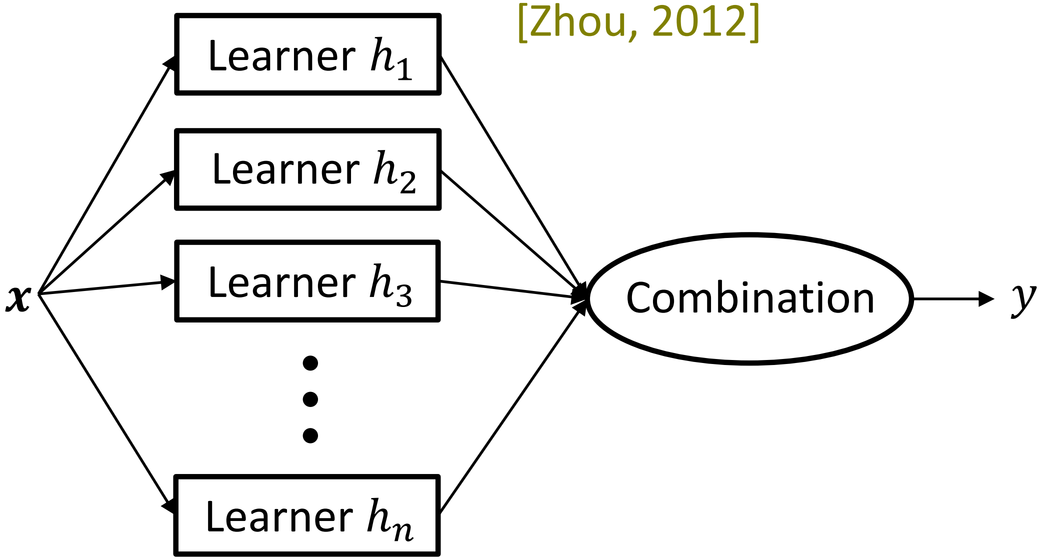
Outline

- Introduction
- Theoretical analysis tools for MOEAs
- Theoretical perspectives of MOEAs
  - Recombination operator, constrained optimization, noisy optimization
- **Multi-objective evolutionary learning algorithms**
  - **Selective ensemble, subset selection**
- Conclusion

Selective Ensemble

Ensemble learning

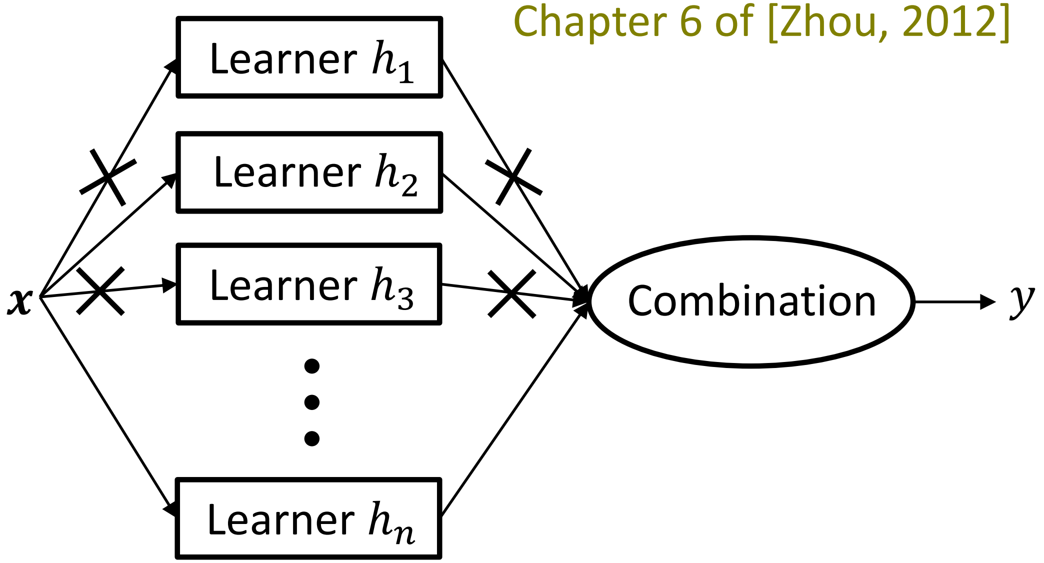
[Zhou, 2012]



- achieves better performance than a single learner

Selective ensemble

Chapter 6 of [Zhou, 2012]



- achieves better performance than the complete ensemble
- reduces storage and improve efficiency

Selective Ensemble

Selective ensemble naturally bears two goals: maximize the generalization performance; minimize the number of selected learners

Previous methods can be roughly categorized into two branches

- **Ordering-based selective ensemble methods (OSE):**

  e.g., error minimization [Margineantu and Dietterich, ICML'97], diversity-like criterion maximization [Martínez-Munõz et al., TPAMI'09], combined criterion [Li et al., ECML'12]

- **Single-objective optimization-based methods (SOSE):**

  e.g., genetic algorithms [Zhou et al., AIJ'02]

$$\max_{\underbrace{\boldsymbol{s}}_{\text{A subset of learners}}\in\{0,1\}^n} \underbrace{\boxed{f(\boldsymbol{s})}}_{\text{Generalization performance}}$$

⇒ Genetic algorithm

No theoretical guarantee

Pareto Optimization for Selective Ensemble

Introduce the Pareto optimization algorithm for selective ensemble (POSE)

Algorithm 13.3 POSE Algorithm

Input: trained individual learners $H = \{h_i\}_{i=1}^{n}$; objective $f : 2^H \to \mathbb{R}$; criterion $eval$
Output: subset of H
Process:

let $\boldsymbol{g}(\boldsymbol{s}) = (f(\boldsymbol{s}), -|\boldsymbol{s}|_1)$ be the bi-objective formulation;
let $\boldsymbol{s}$ = a solution uniformly and randomly selected from $\{0,1\}^n$;
let $P = \{\boldsymbol{s}\}$;
while criterion is not met **do**
  select a solution $\boldsymbol{s}$ from P uniformly at random;
  apply bit-wise mutation on $\boldsymbol{s}$ to generate $\boldsymbol{s}'$;
  if $\nexists \boldsymbol{z} \in P$ such that $\boldsymbol{z} \succ \boldsymbol{s}'$ **then**
    $P = (P \setminus \{\boldsymbol{z} \in P \mid \boldsymbol{s}' \succeq \boldsymbol{z}\}) \cup \{\boldsymbol{s}'\}$;
    $Q = \mathrm{VDS}(f, \boldsymbol{s}')$;
    for $\boldsymbol{q} \in Q$
      if $\nexists \boldsymbol{z} \in P$ such that $\boldsymbol{z} \succ \boldsymbol{q}$ **then**
        $P = (P \setminus \{\boldsymbol{z} \in P \mid \boldsymbol{q} \succeq \boldsymbol{z}\}) \cup \{\boldsymbol{q}\}$
      end if
    end for
  end if
end while
return $\arg\min_{\boldsymbol{s} \in P} eval(\boldsymbol{s})$

Bi-objective formulation:

$$\max_{\boldsymbol{s} \in \{0,1\}^n} \left(f(\boldsymbol{s}), -|\boldsymbol{s}|_1\right)$$

Max generalization performance   Min #learners

Initialization: randomly generate a solution, put it into the population P

Reproduction: pick a solution randomly from P, and mutate it to generate a new one

Evaluation & selection: if the new solution is not dominated, put it and its good neighbors into P

MOEA

Output: select a final solution

[Qian, Yu and Zhou, AAAI 2015]

Theoretical Results

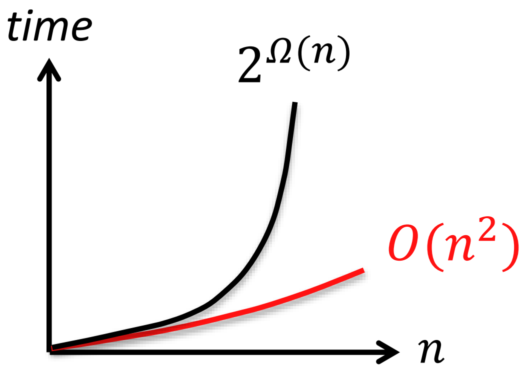
POSE can do better than ordering-based methods

Theorem 1. *For any objective and any size, POSE within* $O(n^4 \log n)$ *expected running time can find a solution weakly dominating that generated by OSE at the fixed size.*

Theorem 2. *For Example 13.1, OSE using Eq. (13.2) finds a solution with objective vector* $(\leq 0, \leq -3)$ *where the two equalities never hold simultaneously, whereas POSE finds a solution with objective vector* $(0, -3)$ *in* $O(n^4 \log n)$ *expected running time.*

POSE can do better than single-objective optimization-based methods

Theorem 3. *For Example 13.2, OSE using Eq. (13.2) finds the optimal solution in* $\boxed{O(n^2)}$ *running time, whereas the running time of SOSE is at least* $\boxed{2^{\Omega(n)}}$ *with probability* $1 - 2^{-\Omega(n)}$.



The first evolutionary learning algorithm with theoretical guarantee!

[Qian, Yu and Zhou, AAAI 2015]

Empirical Results

Pruning bagging base learners with size 100

Comparison on test error

baseline methods ordering-based methods single-objective optimization-based methods

previous EA without theoretical guarantee

Test Error

Data set	POSE	Bagging	BI	RE	Kappa	CP	MD	DREP	EA
australian	.144±.020	**.143±.017**	.152±.023•	.144±.020	**.143±.021**	.145±.022	.148±.022	.144±.019	**.143±.020**
breast-cancer	**.275±.041**	.279±.037	.298±.044•	.277±.031	.287±.037	.282±.043	.295±.044•	**.275±.036**	**.275±.032**
disorders	**.304±.039**	.327±.047•	.365±.047•	.320±.044•	.326±.042•	.306±.039	.337±.035•	.316±.045	.317±.046•
heart-statlog	.197±.037								
house-votes	.045±.019								
ionosphere	.088±.021								
kr-vs-kp	**.010±.003**								
letter-ah	.013±.005								
letter-br	**.046±.008**	.059±.013•	.078±.012•	.048±.012	.048±.014	.048±.012	.057±.014•	.048±.009	.053±.011•
letter-oq	.043±.009								
optdigits	**.035±.006**								
satimage-12v57	**.028±.004**								
satimage-2v5	**.021±.007**								
sick	**.015±.003**								
sonar	**.248±.056**	.266±.052	.310±.051•	.267±.053•	.249±.059	.250±.048	.268±.055•	.257±.056	.251±.041
spambase	**.065±.006**	.068±.007•	.093±.008•	.066±.006	.066±.006	.066±.006	.068±.007•	**.065±.006**	.066±.006
tic-tac-toe	.131±.027	.164±.028•	.212±.028•	.135±.026	.132±.023	.132±.026	.145±.022•	**.129±.026**	.138±.020
vehicle-bo-vs	**.224±.023**	.228±.026	.257±.025•	.226±.022	.233±.024•	.234±.024•	.244±.024•	.234±.026•	.230±.024
vehicle-b-v	**.018±.011**	.027±.014•	.024±.013•	.020±.011	.019±.012	.020±.011	.021±.011•	.019±.013	.026±.013•
vote	.044±.018	.047±.018	.046±.016	.044±.017	**.041±.016**	.043±.016	.045±.014	.043±.019	.045±.015
count of the best	12	2	0	2	7	1	0	5	5
POSE: count of direct win		17	20	15.5	12.5	17	20	12.5	15.5

POSE achieves the smallest error on 60% (12/20) of the data sets, while other methods are no more than 35% (7/20)

POSE is better than any other method on more than 60% (12.5/20) data sets

POSE is never significantly worse

• and ○ denote that POSE is significantly better and worse, respectively, by the t-test with confidence level 0.05

Empirical Results

Pruning bagging base learners with size 100

Comparison on ensemble size

ordering-based methods

single-objective optimization-based methods

	Ensemble Size						
Data set	POSE	RE	Kappa	CP	MD	DREP	EA
australian	10.6±4.2	12.5±6.0	14.7±12.6	11.0±9.7	**8.5±14.8**	11.7±4.7	41.9±6.7•
breast-cancer	8.4±3.5	8.7±3.6	26.1±21.7•	8.8±12.3	**7.8±15.2**	9.2±3.7	44.6±6.6•
disorders	14.7±4.2						
heart-statlog	**9.3±2.3**						
house-votes	**2.9±1.7**						
ionosphere	**5.2±2.2**						
kr-vs-kp	**4.2±1.8**						
letter-ah	**5.0±1.9**						
letter-br	**10.9±2.6**						
letter-oq	**12.0±3.7**	13.6±5.8	13.9±6.0	12.3±4.9	23.0±15.6•	13.7±4.9	39.3±8.2•
optdigits	22.7±3.1						
satimage-12v57	**17.1±5.0**						
satimage-2v5	**5.7±1.7**						
sick	**6.9±2.8**						
sonar	11.4±4.2						
spambase	17.5±4.5	18.5±5.0	20.0±8.1	19.0±9.9	28.8±17.0•	**16.7±4.6**	39.7±6.4•
tic-tac-toe	14.5±3.8	16.1±5.4	17.4±6.5	15.4±6.3	28.0±22.6•	**13.6±3.4**	39.8±8.2•
vehicle-bo-vs	16.5±4.5	15.7±5.7	16.5±8.2	**11.2±5.7◦**	21.6±20.4	**3.2±5.0◦**	41.9±5.6•
vehicle-b-v	**2.8±1.1**	3.4±2.1	4.5±1.6•	5.3±7.4	**2.8±3.8**	4.0±3.9	48.0±5.6•
vote	**2.7±1.1**	3.2±2.7	5.1±2.6•	5.4±5.2•	6.0±9.8	3.9±2.5•	47.8±6.1•
count of the best	12	2	0	2	3	3	0
POSE count of direct win		17	19.5	18	17.5	16	20

previous EA without theoretical guarantee

POSE achieves the smallest size on 60% (12/20) of the data sets, while other methods are no more than 15% (3/20)

POSE is better than any other method on at least 80% (16/20) data sets

POSE is never significantly worse, except two losses on vehicle-bo-vs

• and ○ denote that POSE is significantly better and worse, respectively, by the t-test with confidence level 0.05

[Qian, Yu and Zhou, AAAI 2015]

Subset Selection

There are many other 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

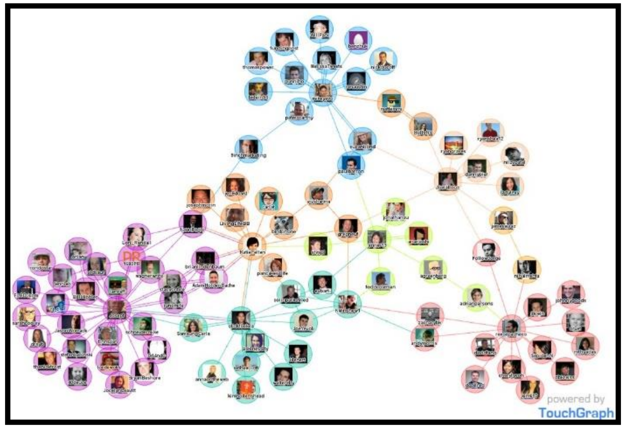
Sparse regression

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

Subset Selection

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

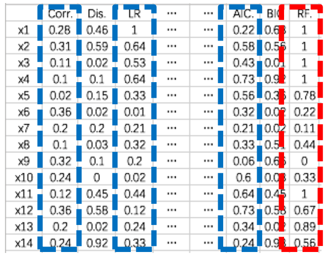


Influence maximization

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

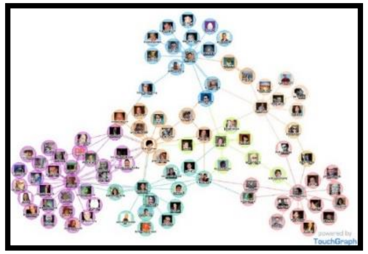
There are many other applications of selecting a good subset from a ground set

Sparse regression

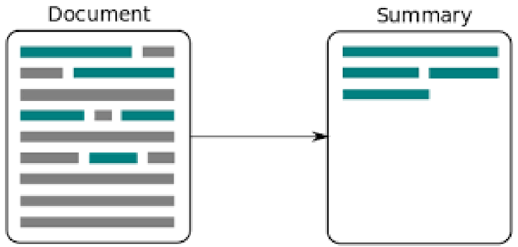


	Corr.	Dis.	LR	···	···	AIC.	BIC	RF.
x1	0.28	0.46	1	···	···	0.22	0.63	1
x2	0.31	0.59	0.64	···	···	0.58	0.56	1
x3	0.11	0.02	0.53	···	···	0.43	0.01	1
x4	0.1	0.1	0.64	···	···	0.73	0.92	1
x5	0.02	0.15	0.33	···	···	0.56	0.36	0.78
x6	0.36	0.02	0.01	···	···	0.32	0.02	0.22
x7	0.2	0.2	0.21	···	···	0.21	0.02	0.11
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	0.08	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

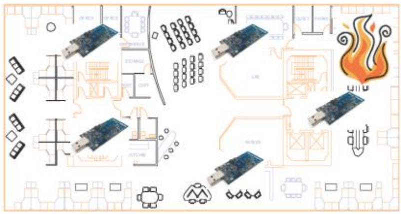
Influence maximization



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

Pareto Optimization for Subset Selection

Introduce the Pareto optimization algorithm for subset selection (POSS)

Constrained: $\max_{S \subseteq V} \; f(S) \quad \text{s.t.} \quad |S| \le b$

Transformation ⇒

Bi-objective: $\min_{S \subseteq V} \; (-f(S), |S|)$

Algorithm 14.2 POSS Algorithm

Input: $V = \{v_1, v_2, \ldots, v_n\}$; objective function $f : \{0,1\}^n \to \mathbb{R}$; budget $b \in [n]$
Parameter: number T of iterations; isolation function $I : \{0,1\}^n \to \mathbb{R}$
Output: solution $\boldsymbol{s} \in \{0,1\}^n$ with $|\boldsymbol{s}|_1 \le b$
Process:

```
let s = 0^n and P = {s};
let t = 0;
while t < T do
    select a solution s from P uniformly at random;
    apply bit-wise mutation on s to generate s';
    if ∄z ∈ P such that I(z) = I(s') and z ≻ s' then
        Q = {z ∈ P | I(z) = I(s') ∧ s' ⪰ z};
        P = (P \ Q) ∪ {s'}
    end if
    t = t + 1
end while
return arg max_{s∈P, |s|_1 ≤ b} f_1(s)
```

MOEA:

- Initialization: put the special solution 0^n into the population P
- Reproduction: pick a solution randomly from P, and mutate it to generate a new one
- Evaluation & selection: if the new solution is not dominated, put it into P and delete bad solutions

Output: select the best feasible solution

[Qian, Yu and Zhou, NIPS 2015]

Theoretical Results

POSS can achieve the optimal polynomial-time approximation guarantee

#iterations

Theorem 14.1. *For subset selection with monotone objective functions, POSS with* $\boxed{\mathbb{E}[T] \le 2eb^2 n}$ *and* $I(\cdot) = 0$*, i.e., a constant function, can find a solution* $\boldsymbol{s}$ *with* $|\boldsymbol{s}|_1 \le b$ *and* $\boxed{f(\boldsymbol{s}) \ge (1 - e^{-\gamma_{\min}}) \cdot \mathrm{OPT},}$ *where* $\gamma_{\min} = \min_{\boldsymbol{s}:|\boldsymbol{s}|_1 = b-1} \gamma_{\boldsymbol{s},b}$.

$\forall S \subseteq T \subseteq V\colon\ f(S) \le f(T)$

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

D. Kempe

Prof., USC
General Chair of STOC'18

Previously obtained by the greedy algorithm [Das and Kempe, ICML'11]

ICML'11 Distinguished Paper Award

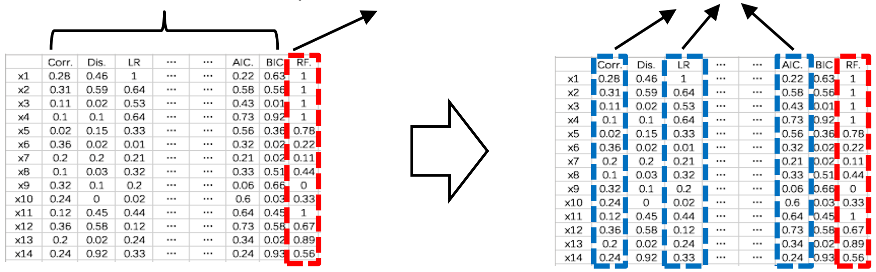
Remark: Approximation guarantee implies worst-case performance

In practice, POSS can do better than the greedy algorithm by escaping from local optima

Theorem 14.2. *For the Exponential Decay subclass of sparse regression, POSS using* $\mathbb{E}[T] = O(b^2(n-b)n\log n)$ *and* $I(\boldsymbol{s} \in \{0,1\}^n) = \min\{i \mid s_i = 1\}$ *can find an optimal solution, while the greedy algorithm cannot.*

Empirical Results

Comparison on sparse regression



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

exhaustive search — greedy algorithms — relaxation methods

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

POSS is always significantly better

[Qian, Yu and Zhou, NIPS 2015]

Previous analyses assume that the objective function can be evaluated exactly

However, only a noisy value can be obtained in many applications of subset selection

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

- Computing the R^2 objective is very expensive
- Estimation by using a set of limited data brings noise

Influence maximization

- Computing the influence spread objective is #P-hard
- Estimation by simulating random diffusion brings noise

Consider a general noise model: $(1-\epsilon)\cdot f(S) \leq f^{\mathrm{n}}(S) \leq (1+\epsilon)\cdot f(S)$

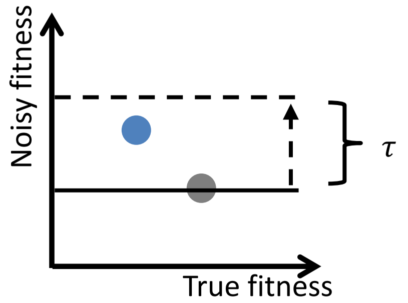
Pareto Optimization for Noisy Subset Selection

Inspired by the robustness of threshold selection against noise

accepts an offspring solution only if its fitness becomes better by at least τ

$$f^{\mathrm{n}}(S) \geq f^{\mathrm{n}}(S') \quad \Rightarrow \quad f^{\mathrm{n}}(S) \geq f^{\mathrm{n}}(S') + \tau$$

reduce the risk of deleting a good solution



Introduce the Pareto optimization algorithm for noisy subset selection (PONSS)

Algorithm 14.2 POSS Algorithm

Input: $V = \{v_1, v_2, \ldots, v_n\}$; objective function $f : \{0,1\}^n \to \mathbb{R}$; budget $b \in [n]$
Parameter: number T of iterations; isolation function $I : \{0,1\}^n \to \mathbb{R}$
Output: solution $\boldsymbol{s} \in \{0,1\}^n$ with $|\boldsymbol{s}|_1 \leq b$
Process:

```
let s = 0^n and P = {s};
let t = 0;
while t < T do
    select a solution s from P uniformly at random;
    apply bit-wise mutation on s to generate s';
    if ∄z ∈ P such that I(z) = I(s') and z ≻ s' then
        Q = {z ∈ P | I(z) = I(s') ∧ s' ⪰ z};
        P = (P \ Q) ∪ {s'}
    end if
    t = t + 1
end while
return arg max_{s∈P, |s|_1 ≤ b} f_1(s)
```

modifies the domination-based comparison of POSS

POSS

$$S \succcurlyeq S' \Leftrightarrow \begin{cases} f^{\mathrm{n}}(S) \geq f^{\mathrm{n}}(S') \\ |S| \leq |S'| \end{cases}$$

PONSS

$$S \succcurlyeq S' \Leftrightarrow \begin{cases} f^{\mathrm{n}}(S) \geq \dfrac{1+\theta}{1-\theta} f^{\mathrm{n}}(S') \\ |S| \leq |S'| \end{cases}$$

$\theta \in [0,1)$

[Qian, Shi, Yu, Tang and Zhou, NIPS 2017]

Theoretical Results

Approximation ratio under noise

Theorem 16.1. *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* $f(\boldsymbol{s}) \geq \frac{1-\epsilon}{1+\epsilon}(1-e^{-\gamma})\cdot \mathrm{OPT}$.

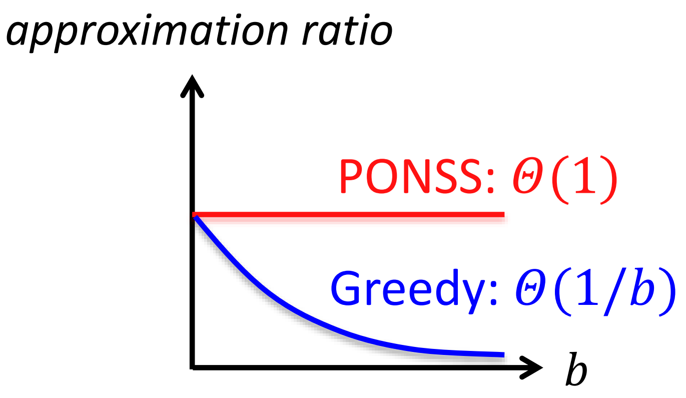
PONSS $$\frac{f(S)}{\mathrm{OPT}} \geq \frac{1-\epsilon}{1+\epsilon}(1-e^{-\gamma})$$

Significantly better

Greedy [Horel and Singer, NIPS'16]

Y. Singer
Gordon McKay Prof., Harvard

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



constant γ and ϵ

EAs achieve better approximation guarantees than conventional algorithms

Large-scale Subset Selection

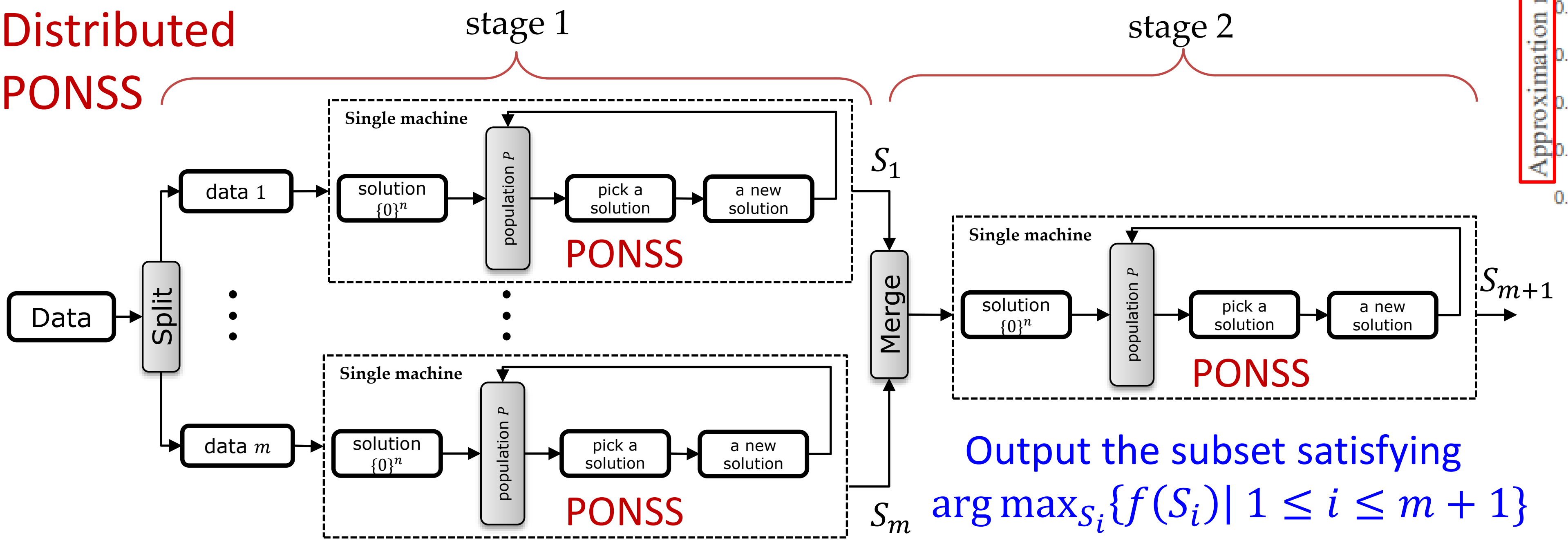
The applications of subset selection are often **large-scale**

Millions of 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

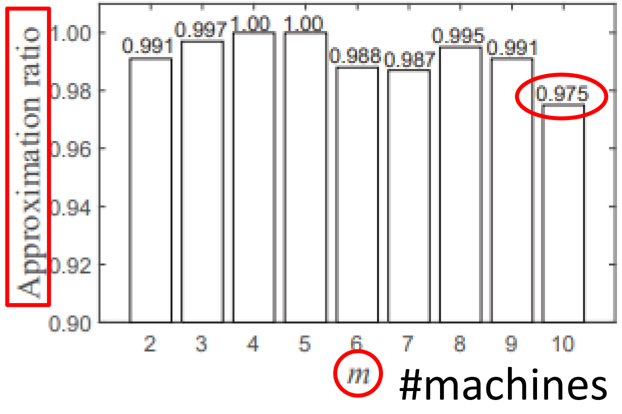
Millions of social network users

Distributed PONSS



Output the subset satisfying $\arg\max_{S_i}\{f(S_i) \mid 1 \le i \le m+1\}$

Empirical Results



very close to the centralized algorithm

[Qian, IEEE Trans. Evolutionary Computation 2020]

How about the performance of POSS under dynamic environments?

Ground set
$V = \{v_1, \dots, v_n\}$

high "quality" & "diversity"

$|S| \le b$

Subset $S \subseteq V$

Yes

Result Diversification $\quad \arg\max_{X \subseteq V} f(S) + \lambda \cdot div(S) \quad s.t. \quad |S| \le b$

quality diversity

A. Borodin
Prof., Univ. of Toronto
Member of the Royal Society of Canada

Open problem: When the objective changes dynamically, is it possible to maintain the (1/2)-approximation ratio in polynomial running time? [Borodin et al., PODS'12]

Outline

- ❑ Introduction
- ❑ Theoretical analysis tools for MOEAs
- ❑ Theoretical perspectives of MOEAs
  - ➢ Recombination operator, constrained optimization, noisy optimization
- ❑ Multi-objective evolutionary learning algorithms
  - ➢ Selective ensemble, subset selection
- ❑ **Conclusion**

Conclusion

Can we build theoretical foundation of multi-objective evolutionary learning?

Yes

Part I: Analysis Methodology

Propose a general theoretical analysis tool, i.e., switch analysis, for MOEAs

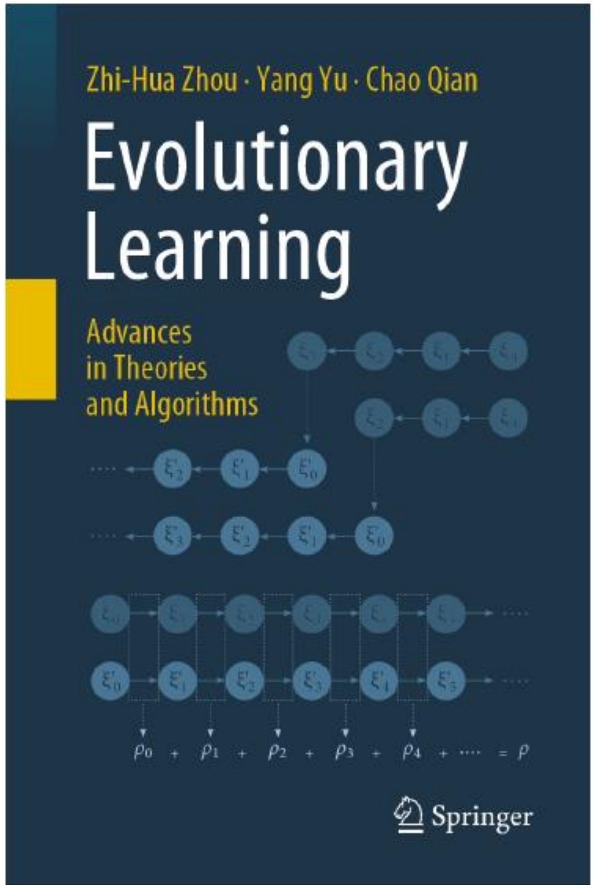
Part II: Theoretical Perspectives

Derive theoretical results of MOEAs about operators, constraints and noise

Part III: Learning Algorithms

Develop theoretically grounded multi-objective evolutionary learning algorithms

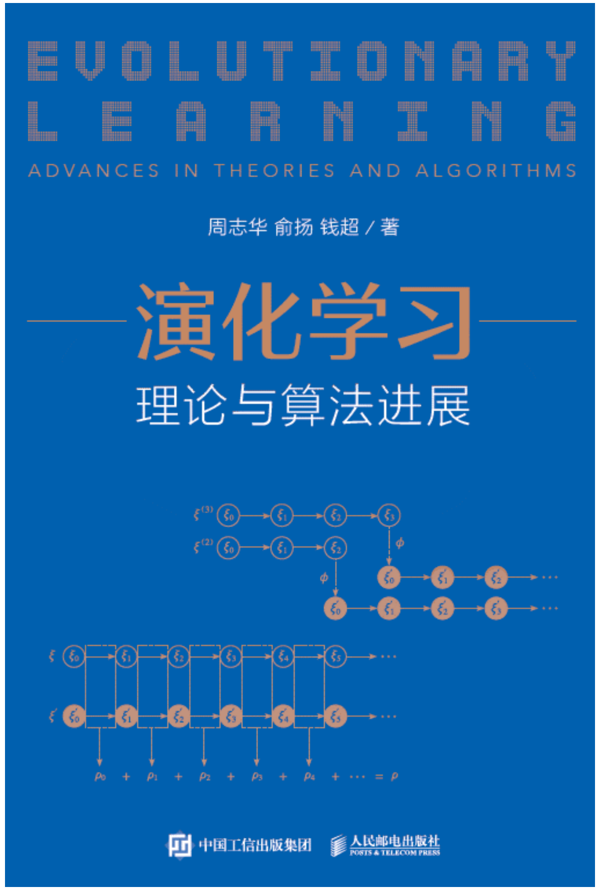
For details



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



Thanks