Why Do We Need Theoretical Research of Evolutionary Algorithms?

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

LAMDA Group
Nanjing University, China

Theory

"A theory is a rational type of abstract thinking about a phenomenon, or the results of such thinking."

From Wikipedia

Notions of Theory in Evolutionary Computation

- Experimentally guided theory: Design an experiment to empirically study a question
- Descriptive theory: Describe/measure/quantify observations
- "Theory": Unproven claims, e.g., building block hypothesis [Goldberg, 1989]

Critiqued, even wrong [Reeves and Rowe, 2002]

- Theory: Mathematically proven results

What we mean here

Schema theorem

Schema theorem [Holland, 1975]

- To explain how the population of genetic algorithms changes in steps

Study the change of $m(H,t)$ of Simple Genetic Algorithm

$$\boxed{E[m(H,t+1)]} \geq \boxed{m(H,t)} \cdot \frac{\overline{f_H}}{\bar{f}} \cdot \left(1-\left(p_c \cdot \frac{d(H)}{n-1}\right)\right) \cdot (1-p_m)^{o(H)}$$

- Schema H is a template with “#”= “any”, which defines a subspace

01#1#

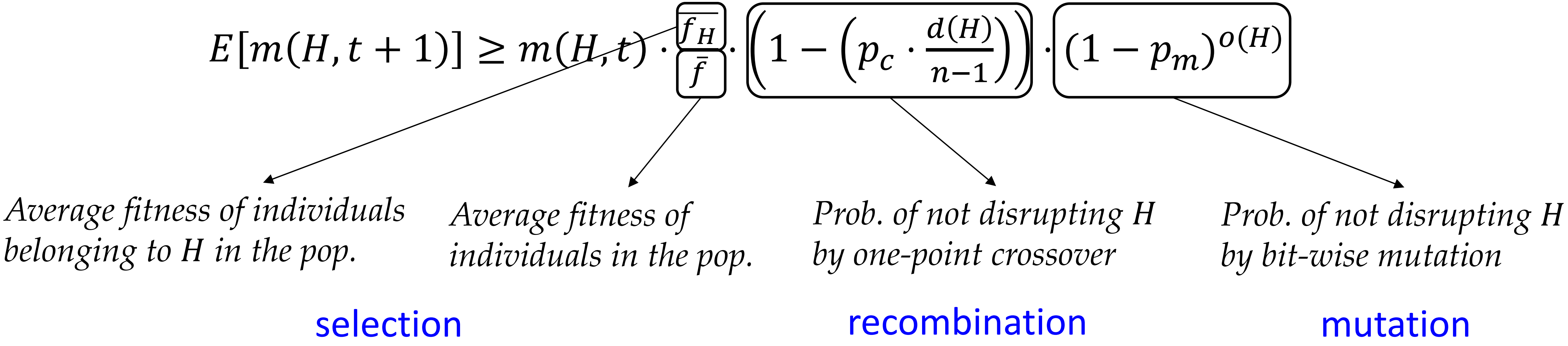
- $m(H,t)$: number of individuals belonging to schema H in the t-th population

Schema theorem

Schema theorem [Holland, 1975]

- To explain how the population of genetic algorithms changes in steps

Study the change of $m(H,t)$ of Simple Genetic Algorithm



$$E[m(H,t+1)] \geq m(H,t) \cdot \frac{\boxed{\overline{f_H}}}{\boxed{\bar{f}}} \cdot \boxed{\left(1-\left(p_c \cdot \frac{d(H)}{n-1}\right)\right)} \cdot \boxed{(1-p_m)^{o(H)}}$$

Average fitness of individuals belonging to H in the pop.

Average fitness of individuals in the pop.

Prob. of not disrupting H by one-point crossover

Prob. of not disrupting H by bit-wise mutation

selection

recombination

mutation

Schema theorem

Schema theorem [Holland, 1975]

- To explain how the population of genetic algorithms changes in steps

Study the change of $m(H,t)$ of Simple Genetic Algorithm

$$E[m(H,t+1)] \geq m(H,t) \cdot \frac{\overline{f_H}}{\bar{f}} \cdot \left(1 - \left(p_c \cdot \frac{d(H)}{n-1}\right)\right) \cdot (1-p_m)^{o(H)}$$

$$\Downarrow$$

Low-order and short schema of above-average fitness is more likely to survive

Limitation: ignoring the constructive effect of the operators; explain the local behaviors only

No free lunch theorem

No free lunch theorem [Wolpert and Macready, TEVC 1997]

- To understand the relationship between how well a black-box optimization algorithm performs and the optimization problem on which it is run

Expected Performance of an algorithm $\mathcal{A}$ iterated m times on a cost function f

$$\sum_{f} \boxed{\sum_{d_m^y} \Phi(d_m^y) P(d_m^y \mid f, m, \mathcal{A}_1)} = \sum_{f} \boxed{\sum_{d_m^y} \Phi(d_m^y) P(d_m^y \mid f, m, \mathcal{A}_2)}$$

No free lunch theorem

No free lunch theorem [Wolpert and Macready, TEVC 1997]

- To understand the relationship between how well a black-box optimization algorithm performs and the optimization problem on which it is run

$$\boxed{\sum_{f}}\sum_{d_m^y}\Phi(d_m^y)P(d_m^y \mid f,m,\mathcal{A}_1)=\boxed{\sum_{f}}\sum_{d_m^y}\Phi(d_m^y)P(d_m^y \mid f,m,\mathcal{A}_2)$$

⇩

Any two algorithms are equally good across all problems over the uniform distribution

Also hold for supervised learning algorithms [Wolpert, Neural Computation 1996]

Limitation: NOT a uniform prior in practice

What theory now we focus on?

Goals of design and analysis of algorithms

- Correctness *"Is the solution output by the algorithm always correct?"*
- Computational complexity *"How many computational resources are required?"*

For evolutionary algorithms,

- Convergence *"Does the EA find a global optimum with prob. 1 as #generations goes to infinity?"*
- Running time complexity *"How long does it take to find an (approximate) optimum?"*

Convergence

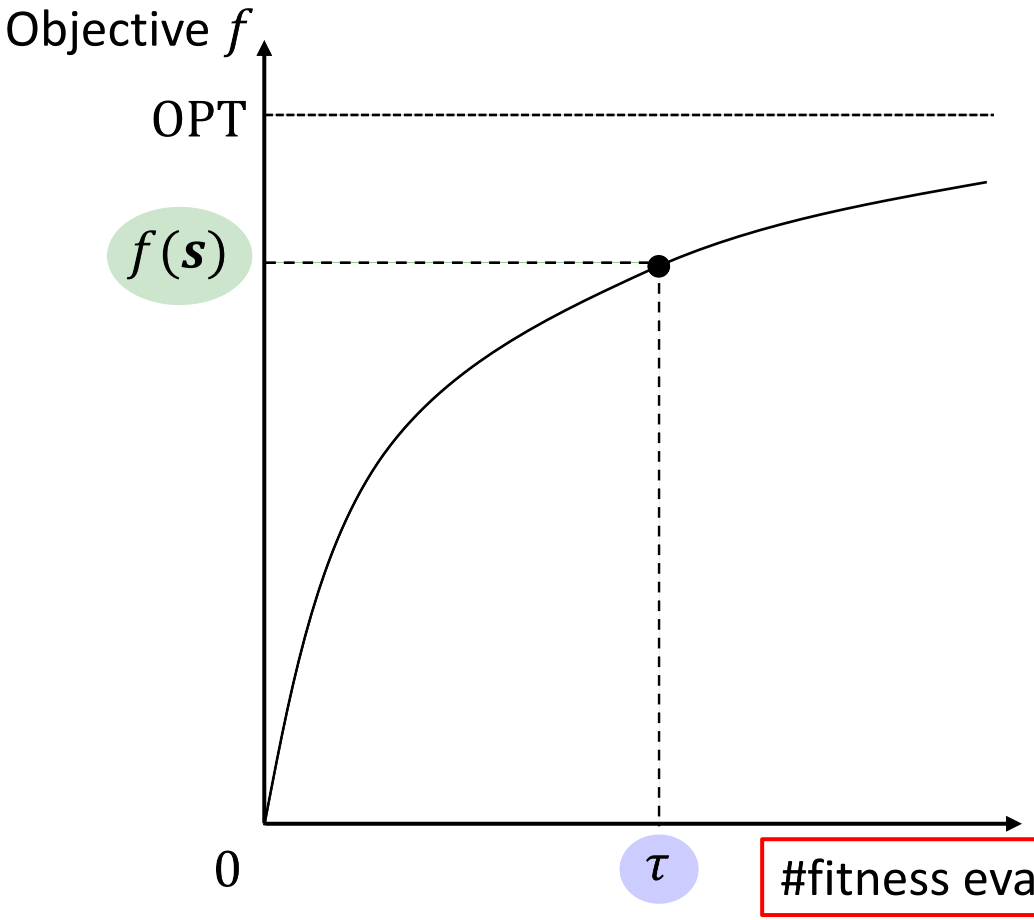
Does an EA converge to a global optimum? $\lim_{t \to +\infty} P(\xi_t \in \mathcal{X}^*) = 1$

Sufficient conditions [Rudolph, 1998]:

- Use global reproduction operators (a positive probability to reach any point)
- Preserve the best found solution (elitism)

But life is limited! How fast does it converge?

Running time complexity



#fitness evaluations

What we concern:

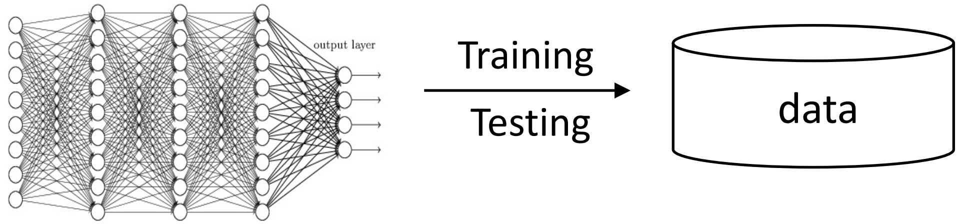
- $E[\tau]$
- $P(\tau \leq T)$

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



Running time analysis

Fitness Level

[Wegener, 2000]

I. Wegener (1950-2008)
TU Dortmund, Germany
Pioneer of EC Theory

Drift Analysis

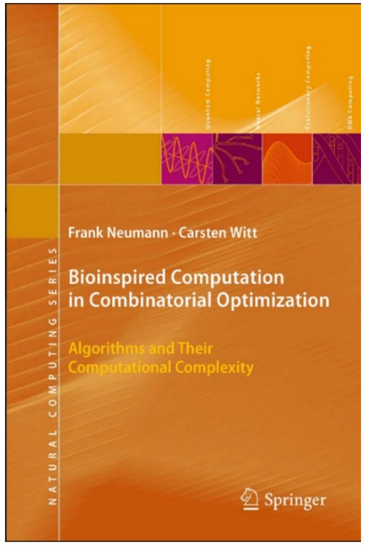
[He & Yao, AIJ’01]

X. Yao
SUSTech, China
IEEE Frank Rosenblatt Award

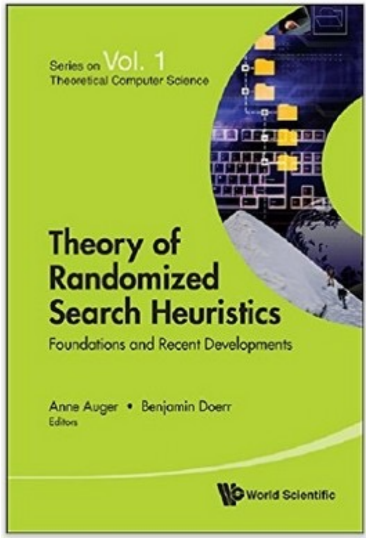
Switch Analysis

[Yu, Qian & Zhou, TEVC’15]

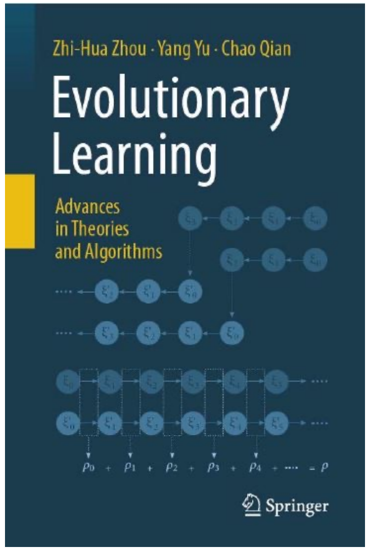
Stefan Droste, Thomas Jansen, Ingo Wegener:
A Rigorous Complexity Analysis of the (1 + 1) Evolutionary Algorithm for Separable Functions with Boolean Inputs.
Evolutionary Computation *6(2): 185-196 (1998)*



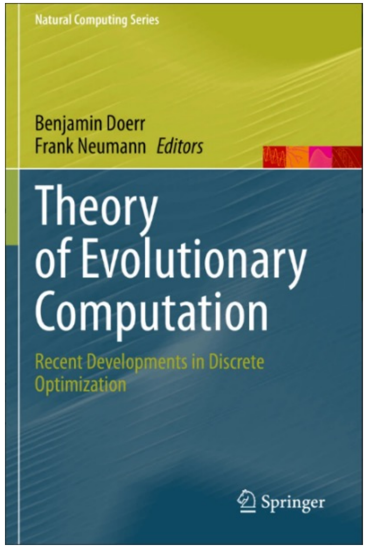
[Neumann and Witt, 2010]



[Auger and Doerr, 2011]



[Zhou, Yu and Qian, 2019]



[Doerr and Neumann, 2020]

How running time analysis can help us?

- Help understand behaviors of EAs
- Guide the design of EAs
- Generate EAs with theoretical guarantees

Example illustration: Help understand behaviors of EAs

Mutation and recombination are two characterizing features of EAs

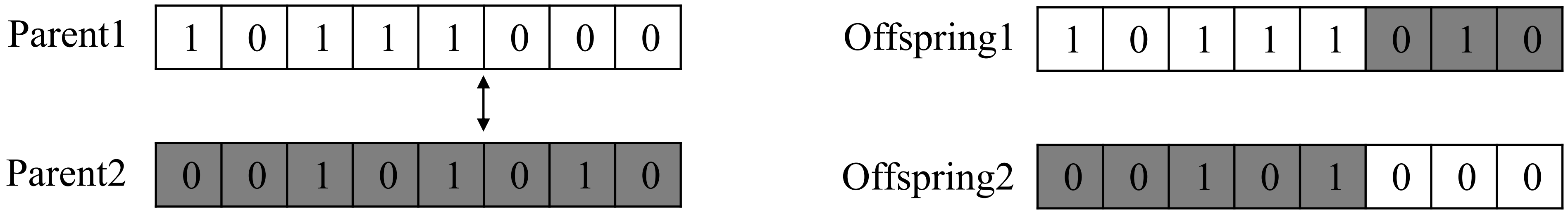
Example of mutation

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

Example illustration: Help understand behaviors of EAs

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

D. Goldberg
UIUC, USA
Pioneer of EC
building block hypothesis
1989

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 TEVC
recombination not work
2014

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

Mainly focused on single-objective optimization

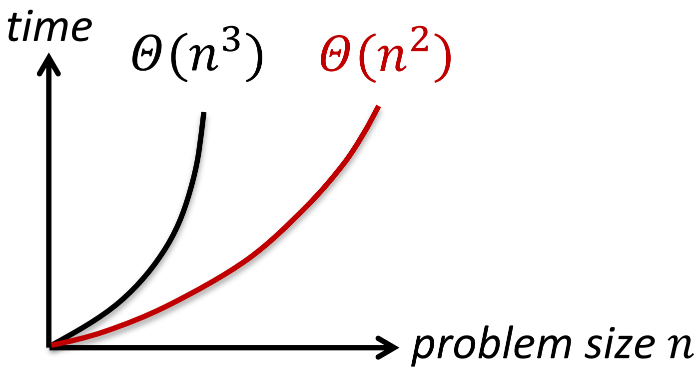
How about the influence of recombination for multi-objective optimization?

- Important applications of EAs
- More complex than single-objective optimization

Example illustration: Help understand behaviors of EAs

Theorem: For GSEMO solving the LOTZ problem

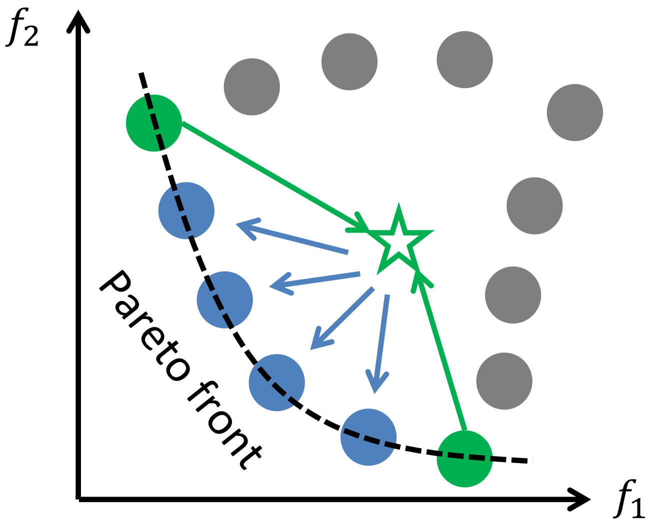
Expected running time $\Theta(n^3) \xrightarrow{\text{recombination}} \Theta(n^2)$



Our findings:

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]

Example illustration: Guide the design of EAs

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

Pareto dominance based: NSGA-II, SPEA-II, …

K. Deb, A. Pratap, S. Agarwal and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002. (Google scholar citations: 45628)

Performance indicator based: SMS-EMOA , HyPE, ….

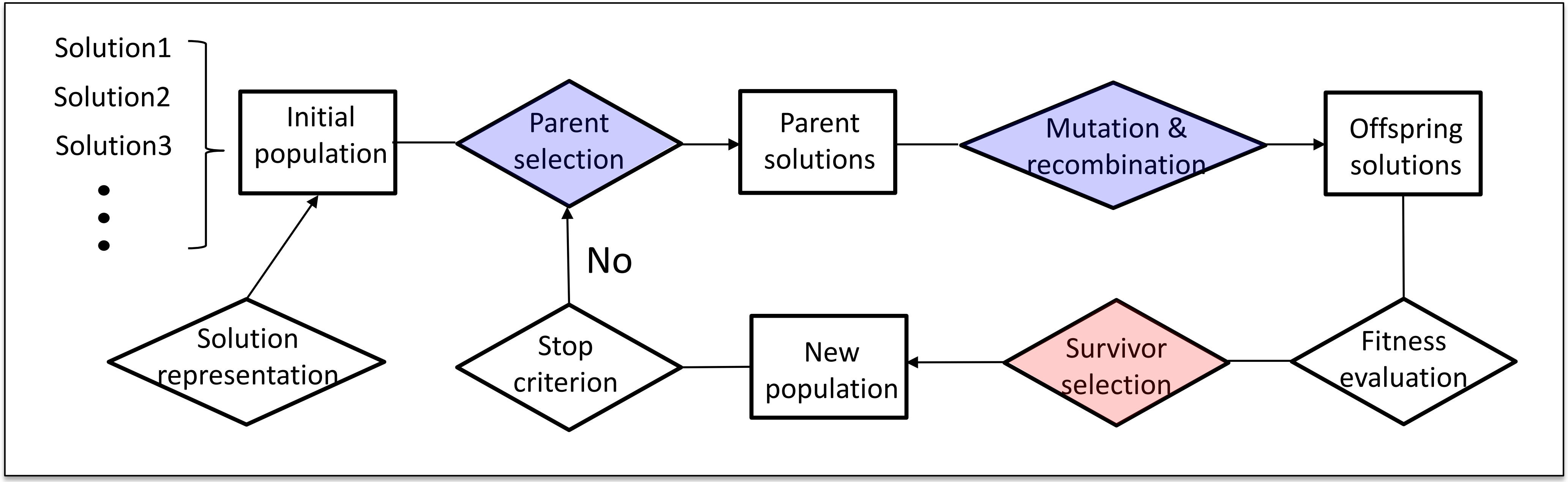
N. Beume, B. Naujoks and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 2007. (Google scholar citations: 1909)

Decomposition based: MOEA/D, ….

Q. Zhang and H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 2007. (Google scholar citations: 7515)

Example illustration: Guide the design of EAs

Two key components of MOEAs: solution generation and population update

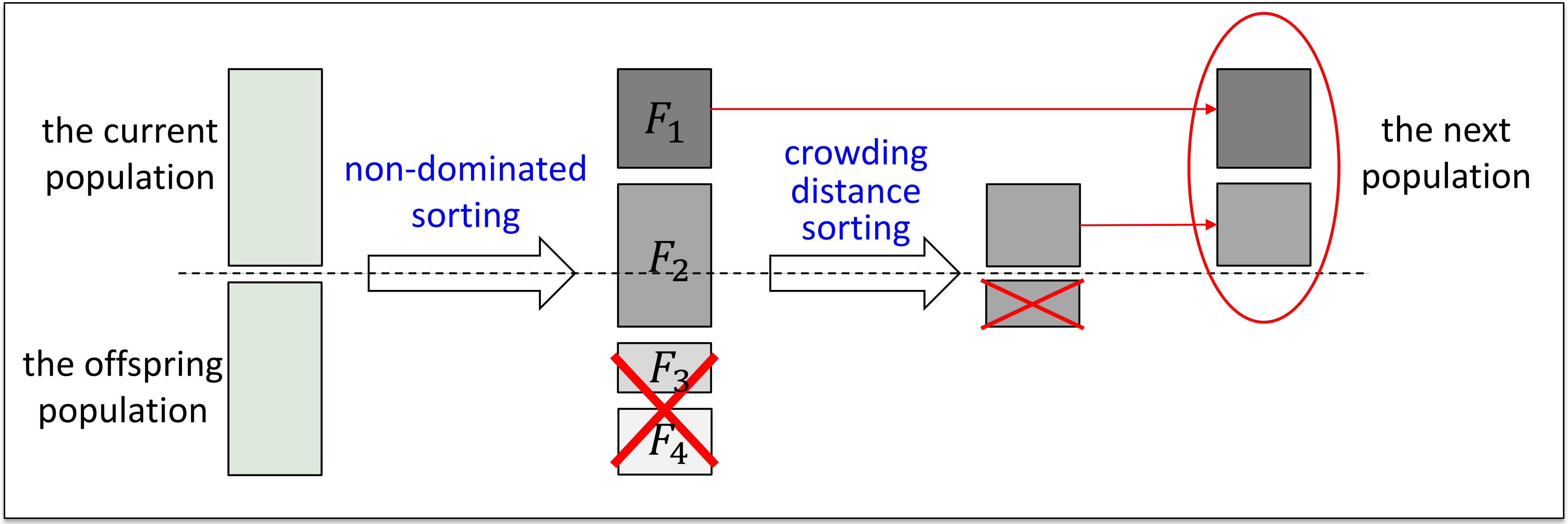


In the area of evolutionary multi-objective optimization, the research focus is mainly on population update

Example illustration: Guide the design of EAs

Population Update of NSGA-II:

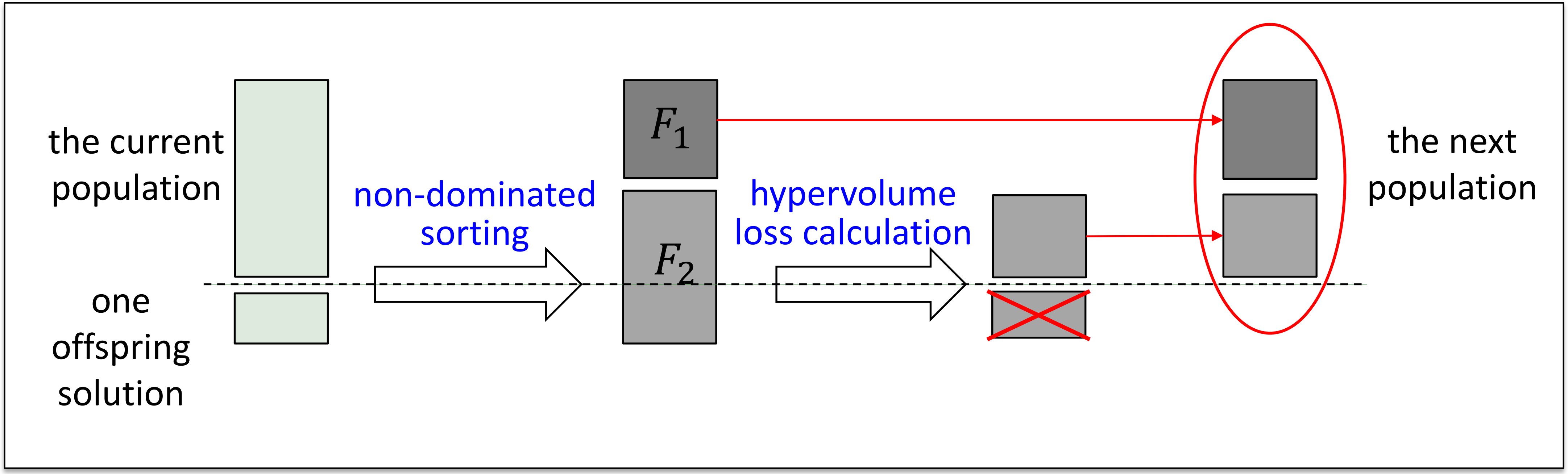
Use non-dominated sorting and crowding distance sorting to rank the solutions, and delete the worst ones



Example illustration: Guide the design of EAs

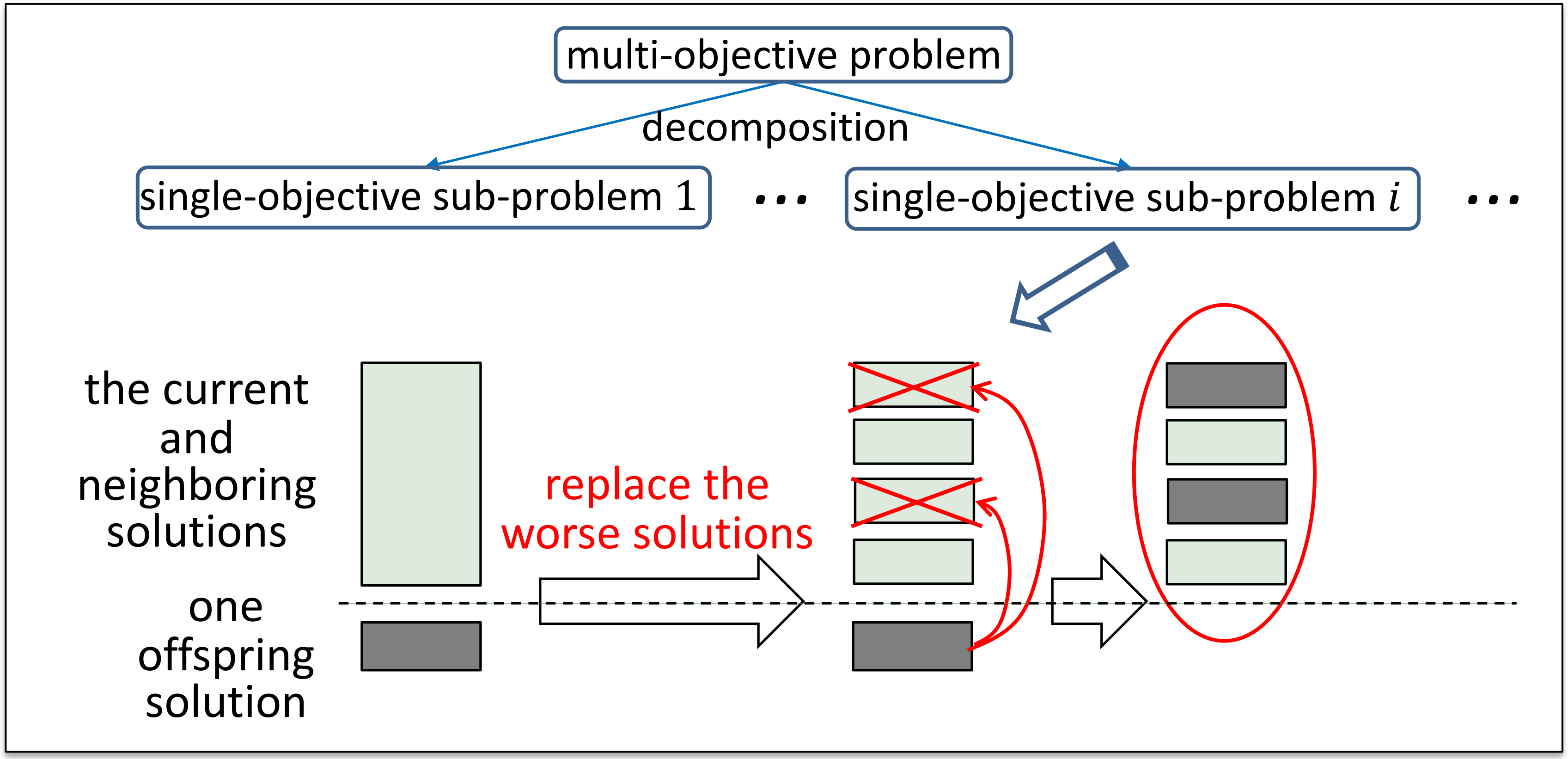
Population Update of SMS-EMOA:

Use non-dominated sorting and quality indicators (e.g., hypervolume) to rank the solutions, and delete the worst solution



Example illustration: Guide the design of EAs

Population Update of MOEA/D:



Example illustration: Guide the design of EAs

The prominent feature in population update of MOEAs: Elitism

- the next-generation population is formed by selecting the best-ranked solutions

K. Deb

"One common aspect of these ***first-generation multi-objective algorithms is that they did not use any elite-preservation operator, thereby compromising the performance*** *and was also contrary to Rudolph's asymptotic convergence proof which required the preservation of elites from one generation to the next."*

An Interview with Kalyanmoy Deb 2022 ACM Fellow

Is elitist population update always better?

NO!

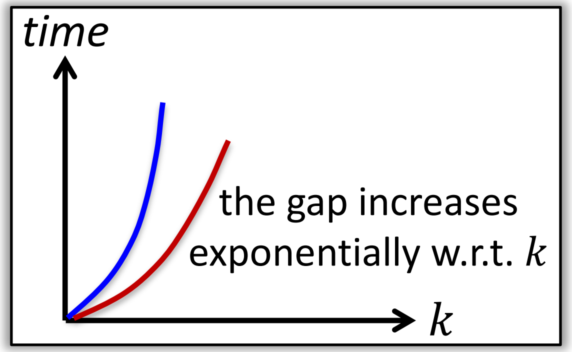
Example illustration: Guide the design of EAs

Theorem: For SMS-EMOA solving the OneJumpZeroJump problem

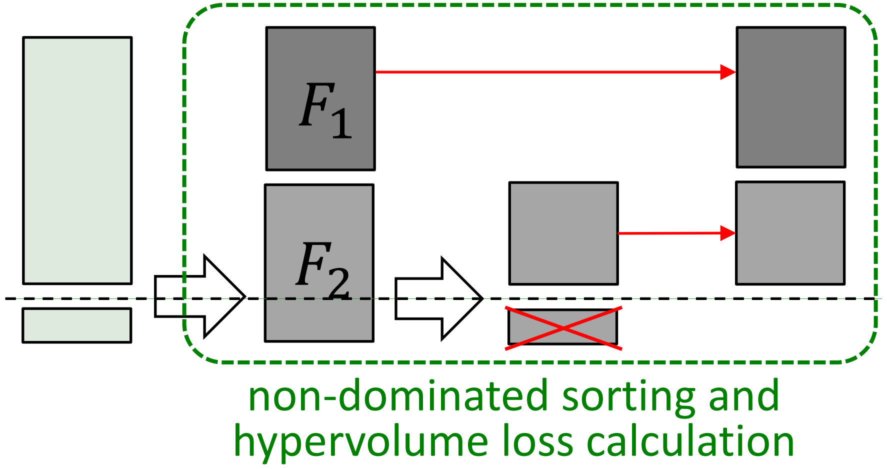
Non-elitism

Expected running time $\Omega(n^k)$ $\xrightarrow{\text{accelerate by } 2^{k/4}/\mu^2}$ $O(\mu n^k \cdot \min\{1, \mu/2^{k/4}\})$

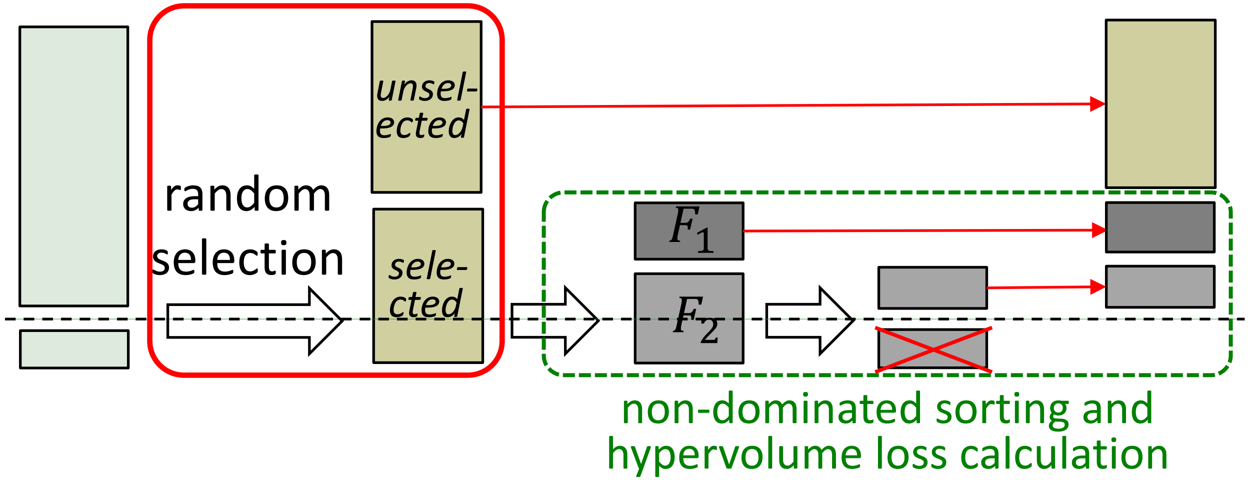
exponentially faster



Original population update of SMS-EMOA



Stochastic population update (Non-elitism)



[Bian, Zhou, Li and Qian, IJCAI 2023]

Example illustration: Guide the design of EAs

The OneJumpZeroJump problem:

$$f_1(\boldsymbol{x}) = \begin{cases} k + |\boldsymbol{x}|_1, & \mathit{if}\ |\boldsymbol{x}|_1 \leq n - k\ \mathit{or}\ \boldsymbol{x} = 1^n \\ n - |\boldsymbol{x}|_1, & \mathit{else} \end{cases}$$

$$f_2(\boldsymbol{x}) = \begin{cases} k + |\boldsymbol{x}|_0, & \mathit{if}\ |\boldsymbol{x}|_0 \leq n - k\ \mathit{or}\ \boldsymbol{x} = 0^n \\ n - |\boldsymbol{x}|_0, & \mathit{else} \end{cases}$$

Characterize a class of problems where some adjacent Pareto optimal solutions in the objective space locate far away in the decision space

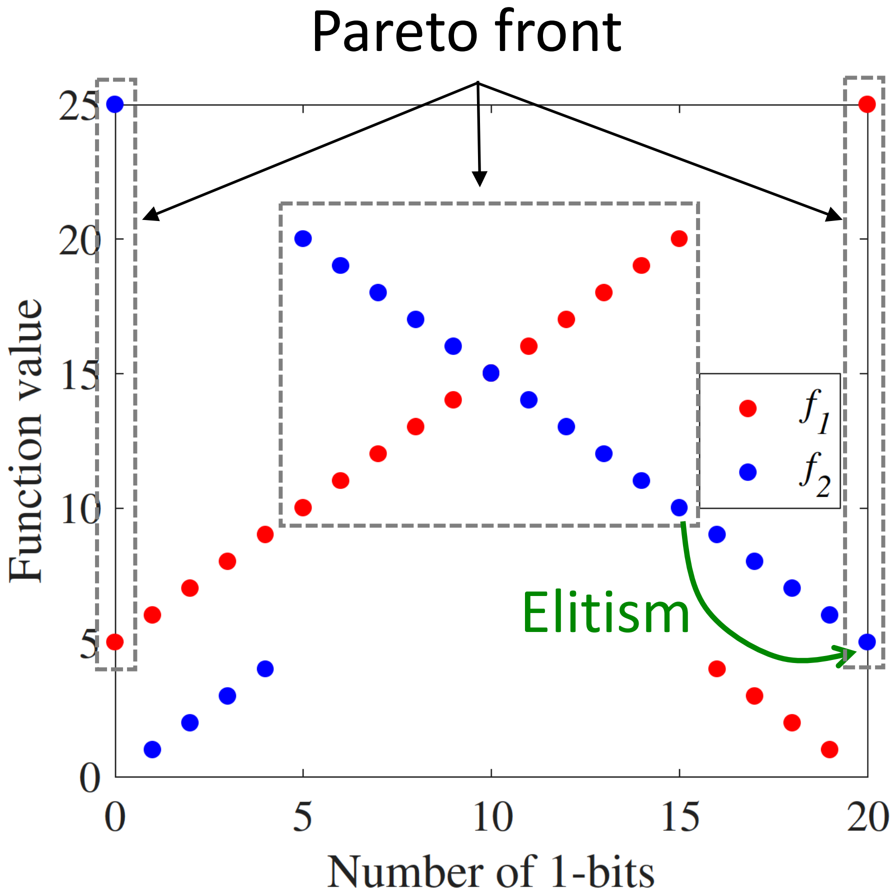


Illustration of function values when $n = 20$ and $k = 5$

[Bian, Zhou, Li and Qian, IJCAI 2023]

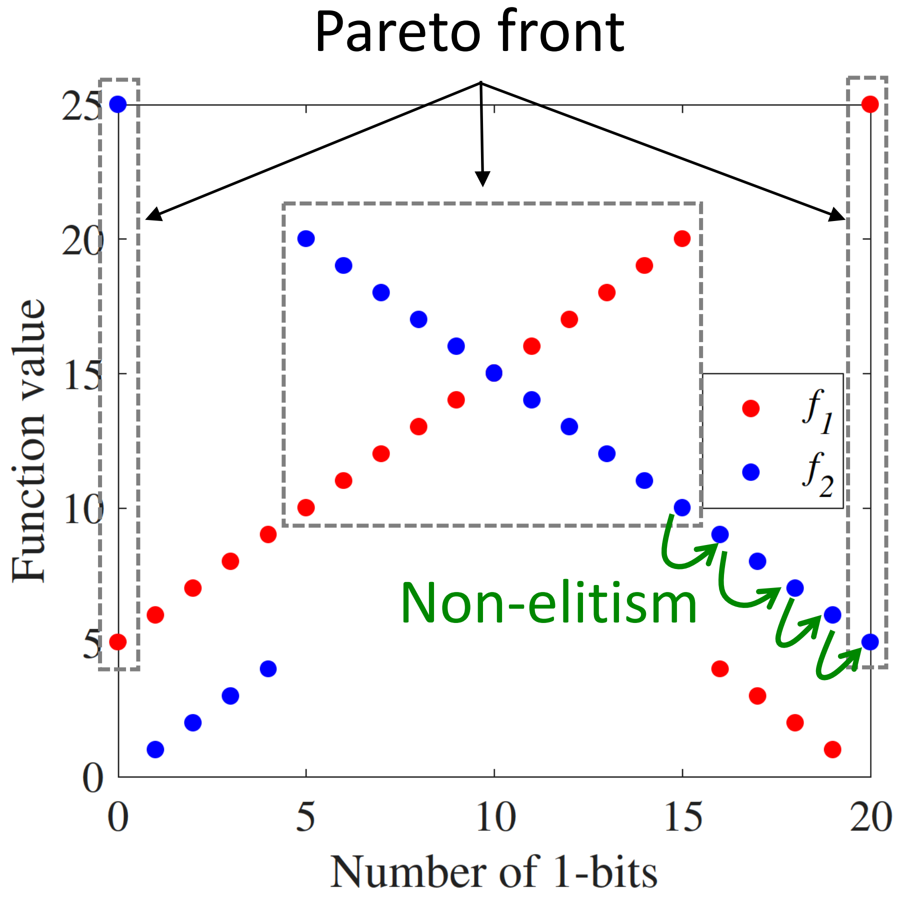
Example illustration: Guide the design of EAs

The OneJumpZeroJump problem:

$$f_1(\boldsymbol{x}) = \begin{cases} k + |\boldsymbol{x}|_1, & \textit{if } |\boldsymbol{x}|_1 \le n - k \textit{ or } \boldsymbol{x} = 1^n \\ n - |\boldsymbol{x}|_1, & \textit{else} \end{cases}$$

$$f_2(\boldsymbol{x}) = \begin{cases} k + |\boldsymbol{x}|_0, & \textit{if } |\boldsymbol{x}|_0 \le n - k \textit{ or } \boldsymbol{x} = 0^n \\ n - |\boldsymbol{x}|_0, & \textit{else} \end{cases}$$

Characterize a class of problems where some adjacent Pareto optimal solutions in the objective space locate far away in the decision space



Pareto front

Non-elitism

Illustration of function values when $n = 20$ and $k = 5$

[Bian, Zhou, Li and Qian, IJCAI 2023]

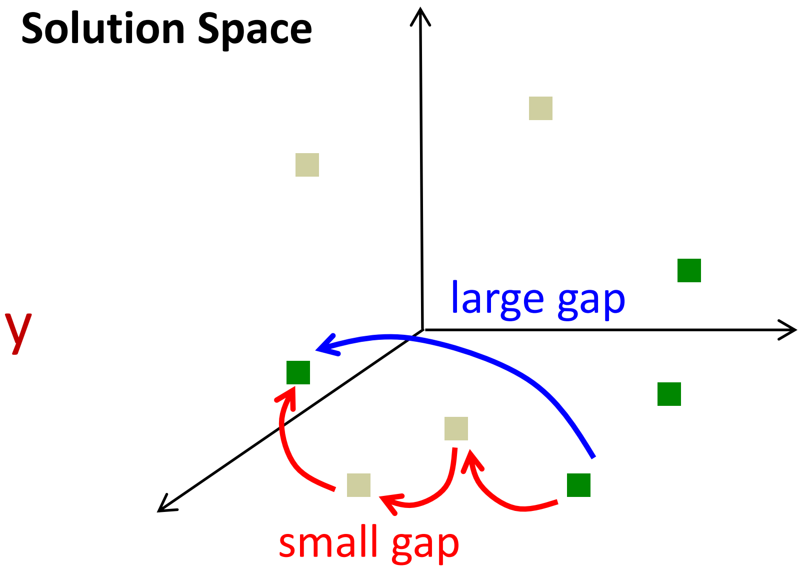
Example illustration: Guide the design of EAs

Non-elitism can make MOEAs go across inferior regions between different Pareto optimal solutions more easily, thus facilitating to find the whole Pareto front

- ➢ Elitism
  - prefers non-dominated solutions
  - if the points in the Pareto front are far away in the solution space, easy to get trapped

May hold more generally

- ➢ Non-elitism
  - allows dominated solutions to participate in the evolutionary process
  - may follow an easier path in the solution space to find points in the Pareto front



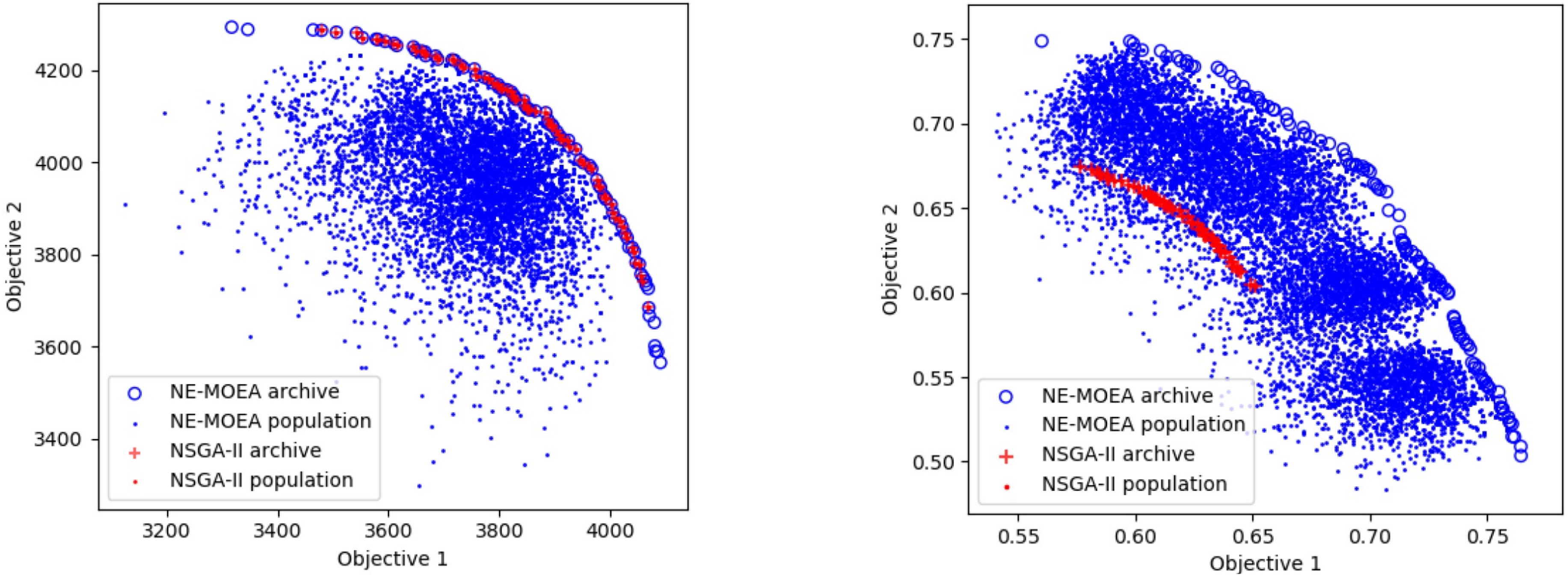
🟩 : Pareto optimal solutions
🟨 : dominated solutions

[Bian, Zhou, Li and Qian, IJCAI 2023]

Example illustration: Guide the design of EAs

Encourage the exploration of developing new MOEAs in the area

For example [Liang, Li and Lehre, arXiv'23]: NSGA-II vs. Non-elitist MOEA (NE-MOEA)



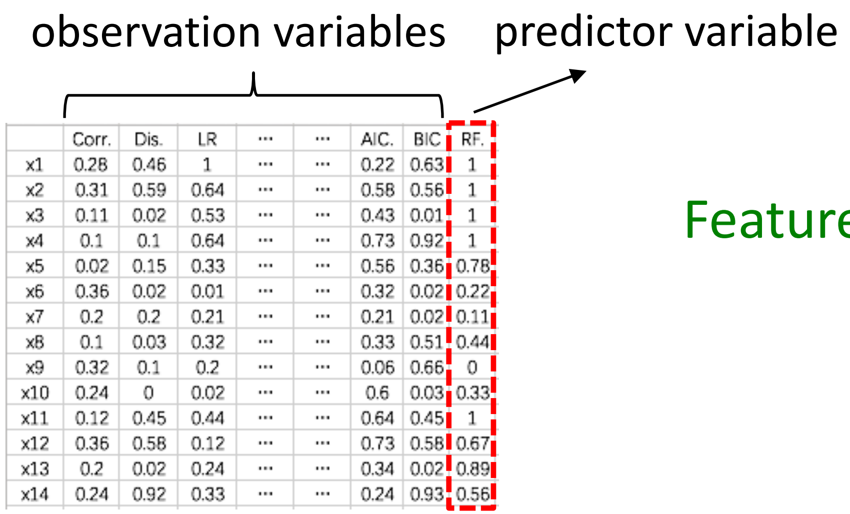
On knapsack with 100 items

On NK-Landscape with $n = 200$ and $k = 10$

[Bian, Zhou, Li and Qian, IJCAI 2023]

Example illustration: Generate EAs with theoretical guarantees

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

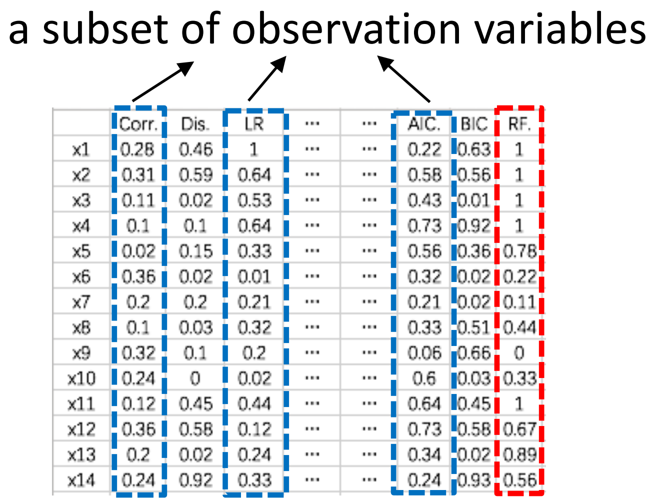


observation variables

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

Feature selection



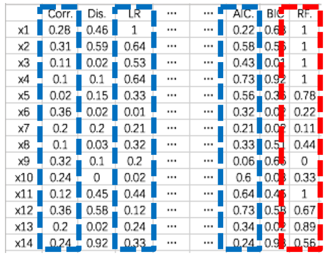
a subset of observation variables

	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

Example illustration: Generate EAs with theoretical guarantees

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

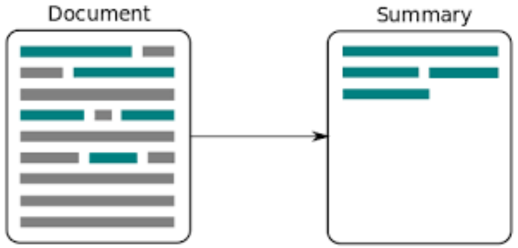
Sparse regression



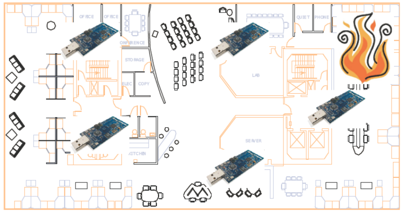
	Corr.	Dis.	LR	···	···	AIC.	BIC	RF.
x1	0.28	0.46	1	···	···	0.22	0.68	1
x2	0.31	0.59	0.64	···	···	0.58	0.56	1
x3	0.11	0.02	0.53	···	···	0.43	0.01	1
x4	0.1	0.1	0.64	···	···	0.73	0.92	1
x5	0.02	0.15	0.33	···	···	0.56	0.36	0.78
x6	0.36	0.02	0.01	···	···	0.32	0.02	0.22
x7	0.2	0.2	0.21	···	···	0.21	0.02	0.11
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	0.98	0.56

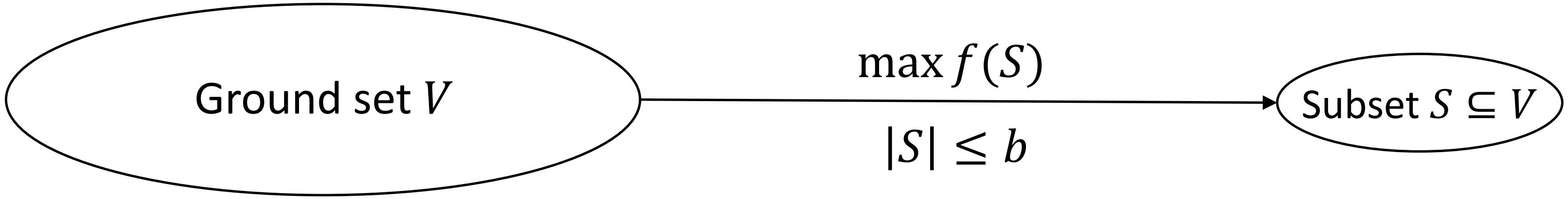
Influence maximization

Document summarization



Sensor placement





Ground set V

$\max f(S)$

$|S| \leq b$

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

Example illustration: Generate EAs with theoretical guarantees

Introduce the Pareto optimization algorithm for subset selection (POSS)

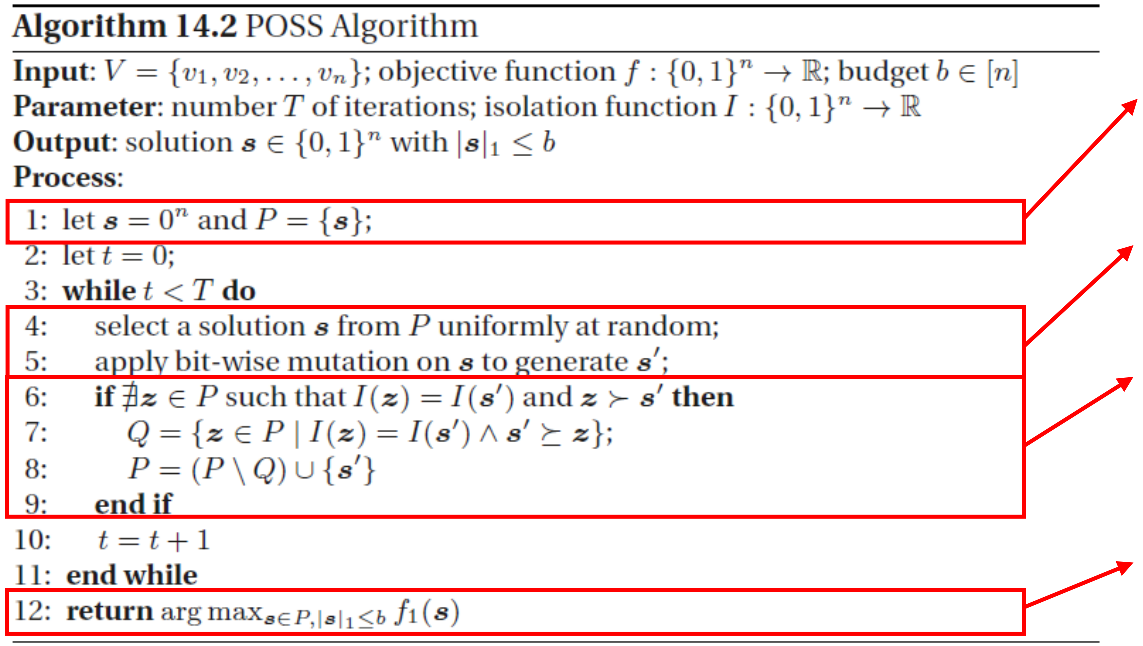
Constrained

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

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 $s\in\{0,1\}^n$ with $|s|_1\le b$
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 $\nexists z\in P$ such that $I(z)=I(s')$ and $z\succ s'$ then
        $Q=\{z\in P\mid I(z)=I(s')\wedge s'\succeq z\}$;
        $P=(P\setminus Q)\cup\{s'\}$;
    end if
    $t=t+1$
end while
return $\arg\max_{s\in P,|s|_1\le b} f_1(s)$
```

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

MOEA

Output: select the best feasible solution

[Qian, Yu and Zhou, NIPS 2015]

Example illustration: Generate EAs with theoretical guarantees

POSS can achieve the optimal polynomial-time approximation guarantee

#iterations

Theorem. *For subset selection with monotone objective functions, POSS with* $\boxed{\mathbb{E}[T] \leq 2eb^2 n}$ *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* $\gamma_{\min} = \min_{\boldsymbol{s}:|\boldsymbol{s}|_1 = b-1} \gamma_{\boldsymbol{s},b}$.

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

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

Experiments

Good reported results

Data set 1	✓
Data set 2	✓
Data set 3	✓
Data set 4	✓
Data set 5	✓

Performance on other data?

Data set 6	?
Data set 7	?
Data set 8	?
Data set 9	?
Data set 10	?

Guaranteed

Safe！

Remark: Theoretical guarantee implies worst-case performance

[Qian, Yu and Zhou, NIPS 2015]

Example illustration: Generate EAs with theoretical guarantees

[Chao Qian. Can Evolutionary Clustering Have Theoretical Guarantees? IEEE Transactions on Evolutionary Computation, in press]

Yes!

Theorem 1. *For k-center clustering, the GSEMO achieves a 2-approximation ratio in polynomial running time.*

Theorem 2. *For discrete k-median clustering, the GSEMO achieves a $\frac{1}{1-\epsilon}\left(3+\frac{2}{p}\right)$-approximation ratio in polynomial running time.*

Theorem 3. *For k-means clustering, the GSEMO achieves a $\frac{1+\epsilon}{(1-\epsilon)^2}\left(3+\frac{2}{p}\right)^2$-approximation ratio in polynomial running time.*

Theorem 4. *For β-fair discrete k-median clustering, the GSEMO achieves a $(84,7)$-bicriteria approximation ratio in polynomial running time.*

Example illustration: Generate EAs with theoretical guarantees

Approximation ratio under noise

Theorem. *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

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

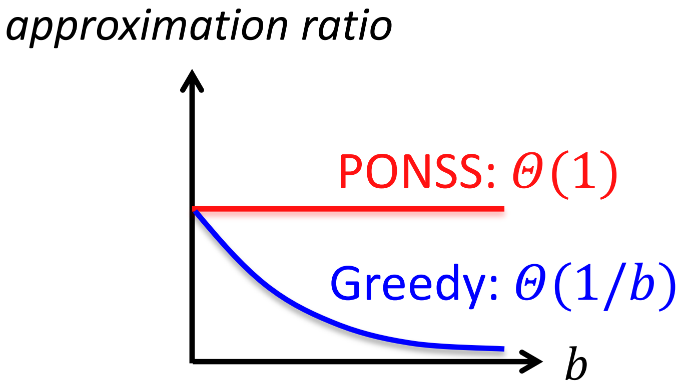
Significantly better

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

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

How running time analysis can help us?

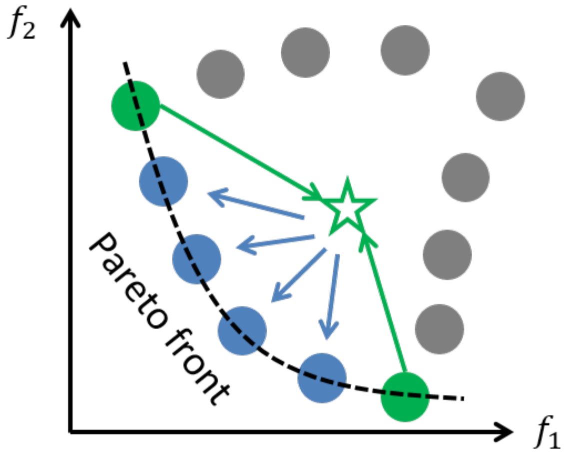
- Help understand behaviors of EAs
- Guide the design of EAs
- Generate EAs with theoretical guarantees

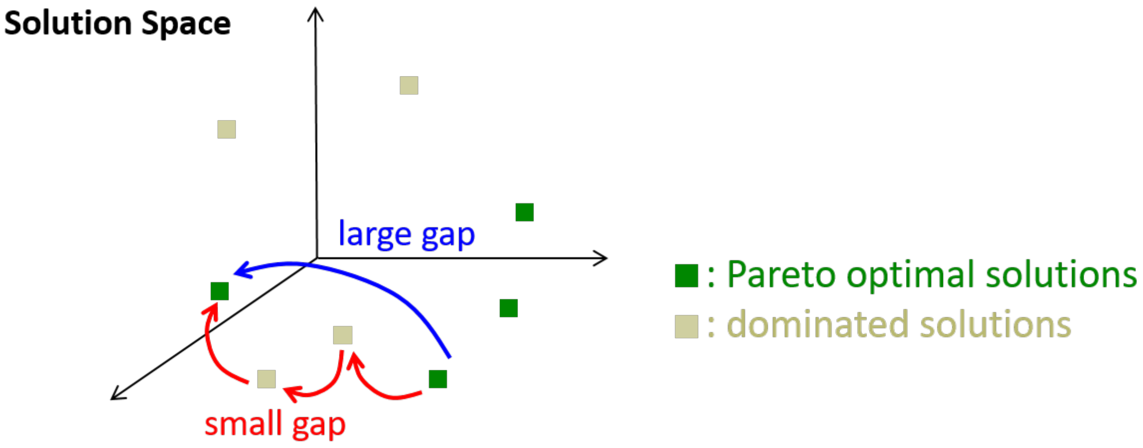
Why do theory?

Estimate the running time complexity by experiments

Why do theory? Because

- Absolute guarantee about the correctness
- Proofs (automatically) give insight in how things work





Why do theory?

Estimate the running time complexity by experiments

Why do theory? Because

- Absolute guarantee about the correctness
- Proofs (automatically) give insight in how things work
- Many results (e.g., on an algorithm/problem class) can be obtained only by theory

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

Hold for any application of subset selection, any problem size n, and any budget b

Limitations of theoretical research

Limitations: Very difficult to obtain!

Theory and experiments are complementary

- Difficult to obtain theory, **do experiments**
- Even there is theory, experiments are still needed

E.g., we derive the expected running time $O(n^2)$ by theoretical analysis

But how about the coefficient? **Do experiments**

Limitations of theoretical research

Limitations: Very difficult to obtain!

Theory and experiments are complementary

- Difficult to obtain theory, do experiments
- Even there is theory, experiments are still needed

E.g., POSS can achieve the optimal polynomial-time approximation guarantee

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

Not bad in the worst case

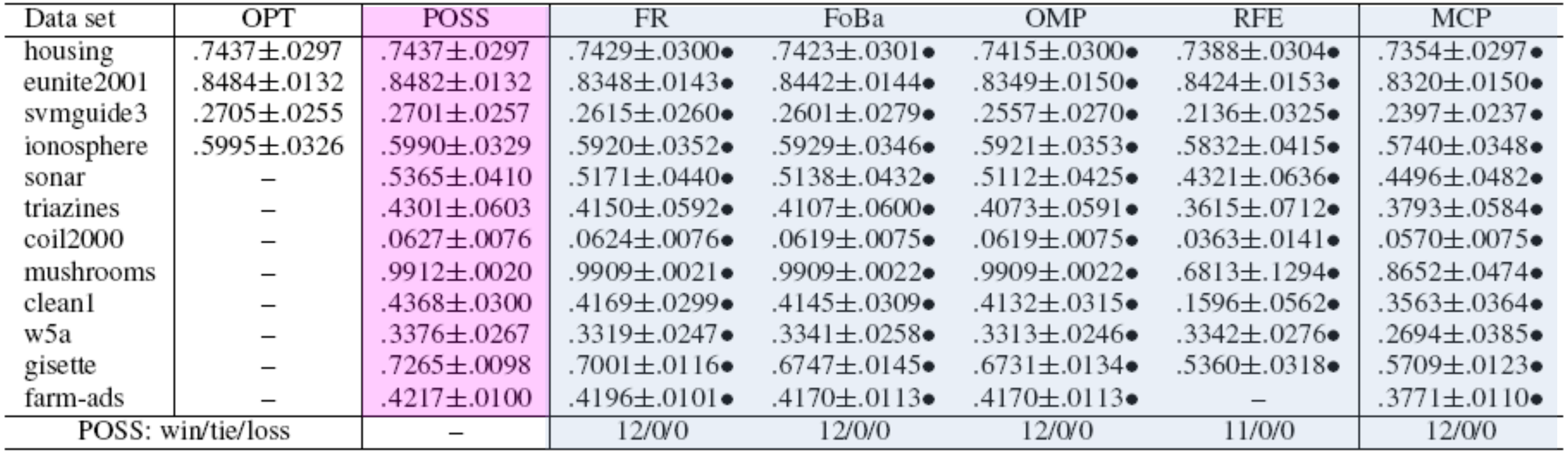
Limitations of theoretical research

E.g., POSS can achieve the optimal polynomial-time approximation guarantee

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

Not bad in the worst case

Do experiments



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

Very good
in normal cases

[Qian, Yu and Zhou, NIPS 2015]

Summary

- Schema theorem
- No free lunch theorem
- Convergence
- Running time complexity
- How theory can help us?
- Why do theory?
- Theory vs. Experiments

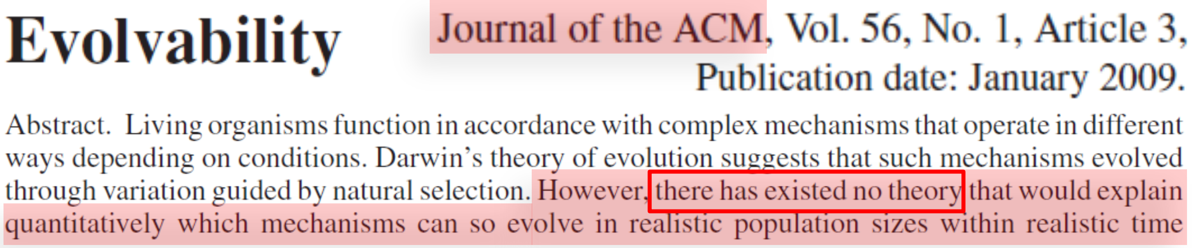
Tired?

Can I do theoretical research of evolutionary algorithms?

Theoretical analysis of evolutionary algorithms is very difficult

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: complicated

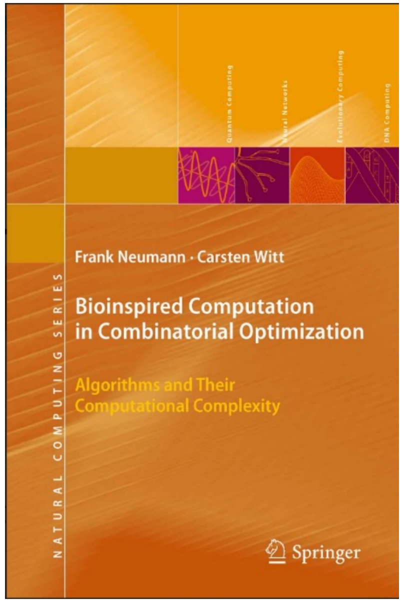
Mathematical knowledge:

- Probability Theory, Randomized Algorithms, Stochastic Processes

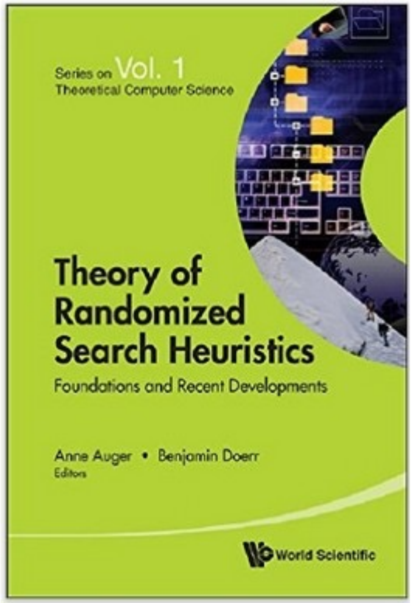
Smart: Good but not necessary!

Concentration!

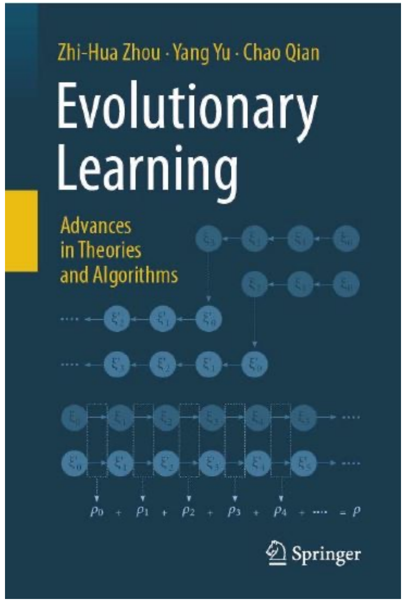
How to do theoretical research of evolutionary algorithms?



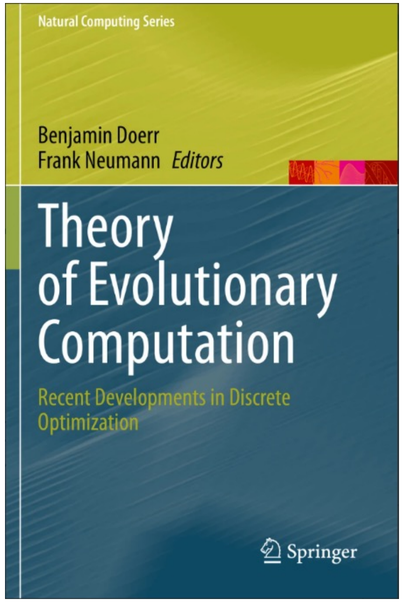
[Neumann and Witt, 2010]



[Auger and Doerr, 2011]



[Zhou, Yu and Qian, 2019]



[Doerr and Neumann, 2020]

Theoretical analysis of MOEAs may be the hottest topic in the next few years

Do useful theory!

Thank you!