Towards Evolutionary Approximate Optimization for Machine Learning

Yang Yu
(俞扬)

National Key Laboratory for Novel Software Technology
Nanjing University, China

joint work with (alphabetic order):

Mr. Yi-Qi Hu, Dr. Chao Qian, Mr. Hong Qian, Mr. Jing-Cheng Shi,
Prof. Ke Tang, Prof. Xin Yao, Prof. Zhi-Hua Zhou

南京大學
NANJING UNIVERSITY

SKL
国家重点实验室
STATE KEY LABORATORY

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

Machine learning

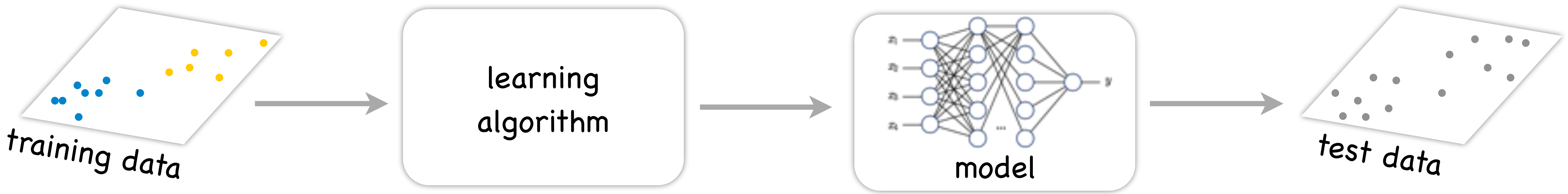
machine learning
is in the center of
artificial intelligence

Machine learning

machine learning
is in the center of
artificial intelligence

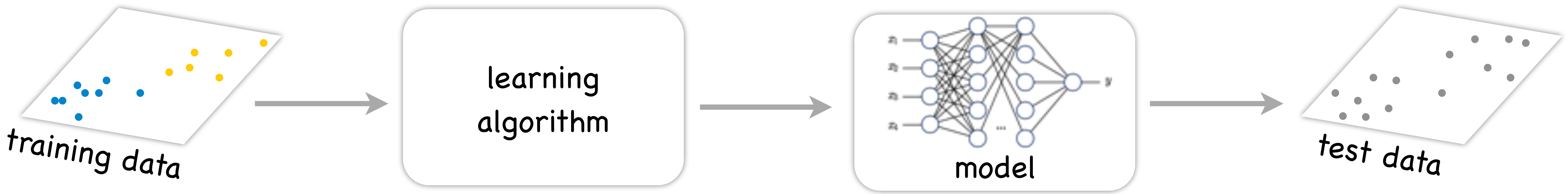
Optimization in machine learning

A typical learning task:



Optimization in machine learning

A typical learning task:

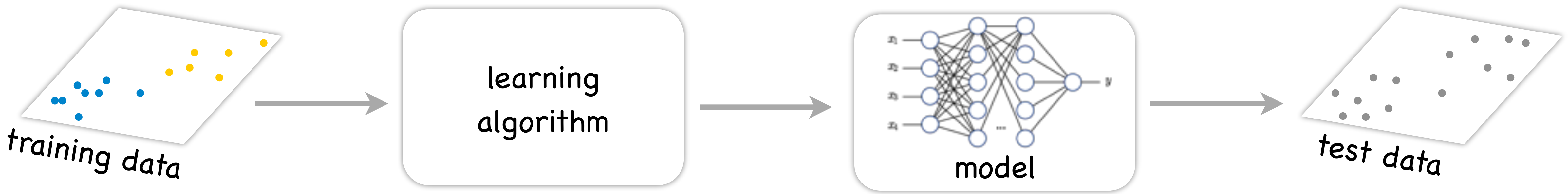


Components [Domingos, CACM'12]:

machine learning = representation + evaluation + optimization

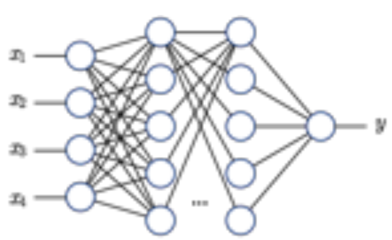
Optimization in machine learning

A typical learning task:



Components [Domingos, CACM'12]:

machine learning = representation + evaluation + optimization

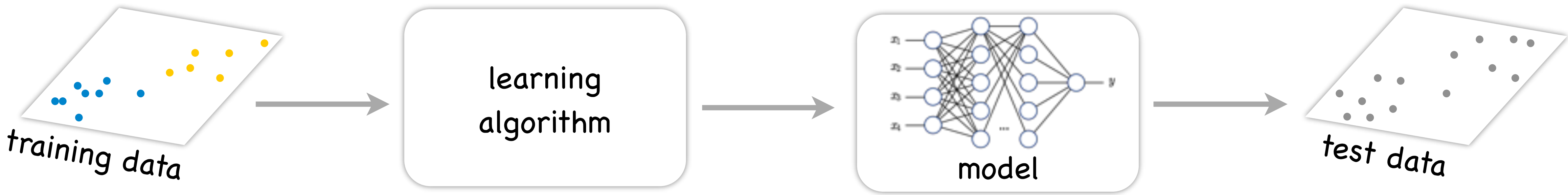


$0/1\ \text{error} + \|w\|_0$

gradient

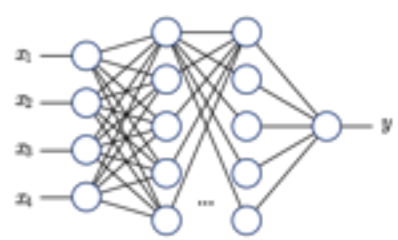
Optimization in machine learning

A typical learning task:



Components [Domingos, CACM'12]:

machine learning = representation + evaluation + optimization



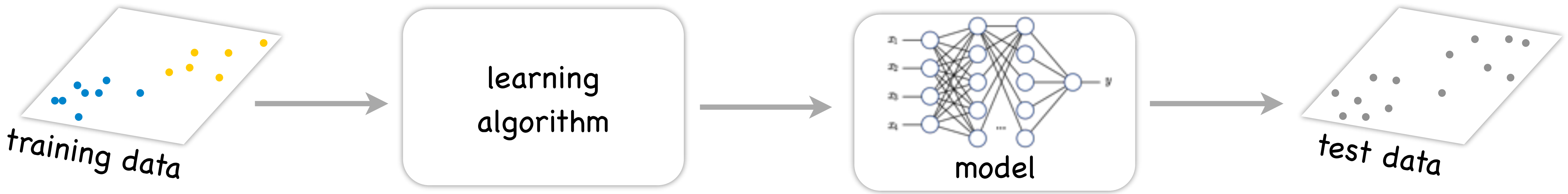
$0/1 \text{ error} + \|w\|_0$

gradient

non-linear ⟶ non-convex

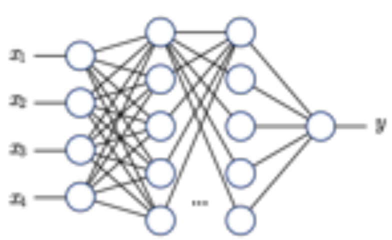
Optimization in machine learning

A typical learning task:



Components [Domingos, CACM'12]:

machine learning = representation + evaluation + optimization



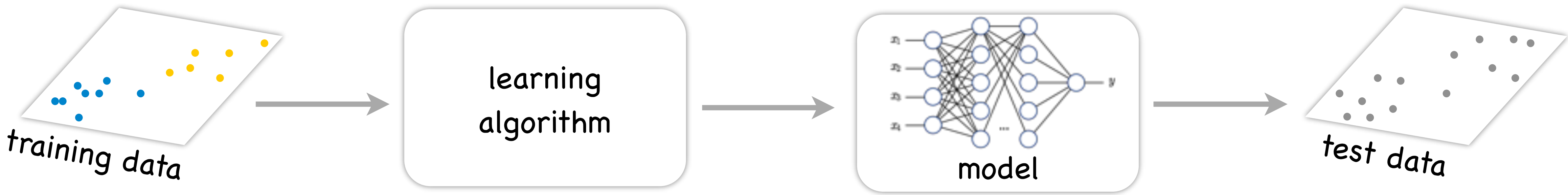
$0/1\ \text{error} + \|w\|_0$

gradient

non-linear → non-convex

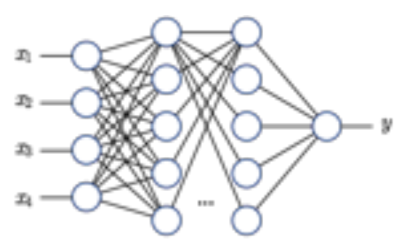
Optimization in machine learning

A typical learning task:



Components [Domingos, CACM'12]:

machine learning = representation + evaluation + optimization



$0/1 \text{ error} + \|w\|_0$

gradient

convex loss functions are noise-sensitive [Long and Servedio, MLJ'00]

convex regularizations are not consistent [Fan and Li, JASA'01]

non-linear → non-convex

can we have more powerful optimization tools?

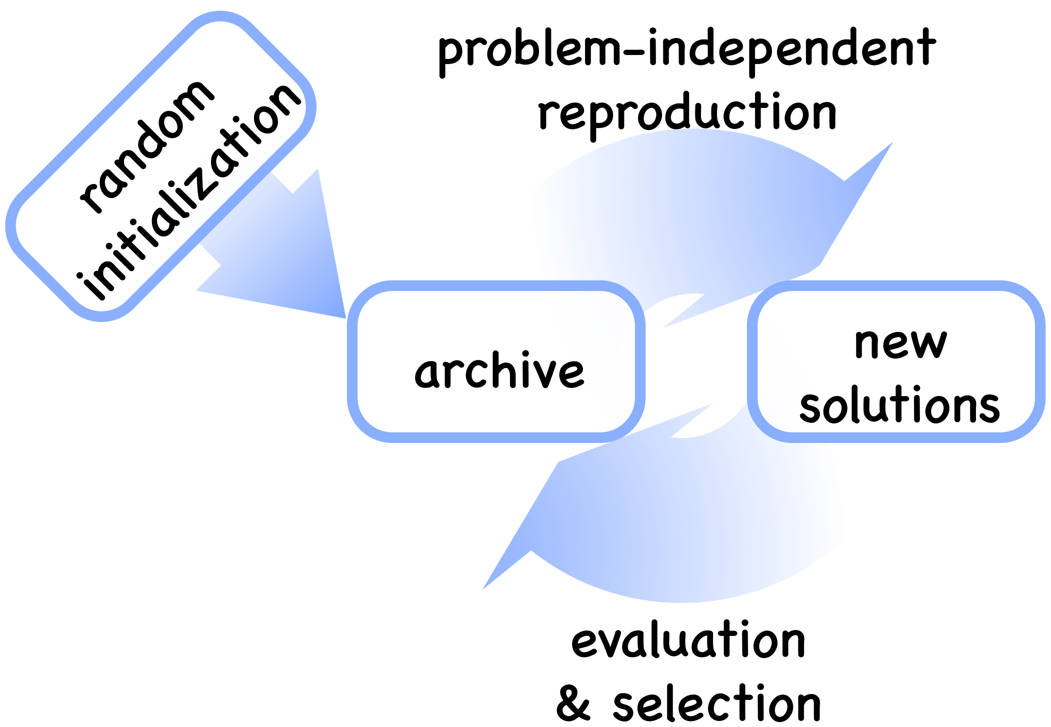
Evolutionary algorithms

Genetic Algorithms [J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

Evolutionary Strategies [I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming [L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms ...



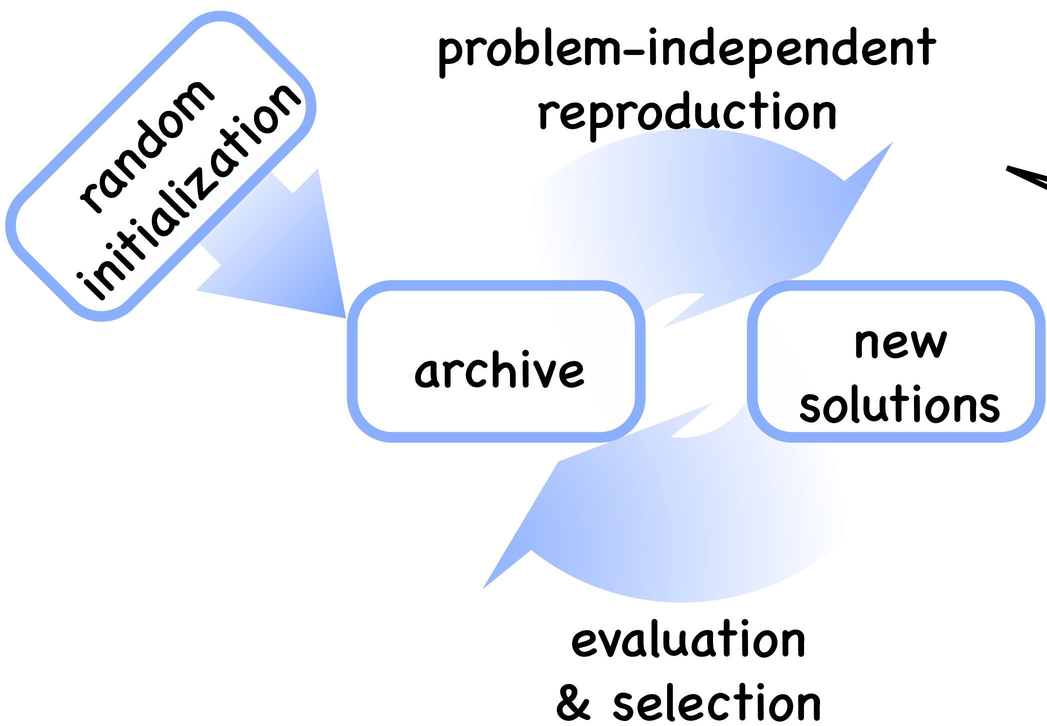
Evolutionary algorithms

Genetic Algorithms [J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

Evolutionary Strategies [I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming [L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms ...



for binary vector:

mutation: [1,0,0,1,0] → [1,1,0,1,0]

crossover: [1,0,0,1,0] + [0,1,1,1,0]

→ [0,1,0,1,0] + [1,0,1,1,0]

for real vector:

mutation: $x = x + \delta, \delta \sim \mathcal{N}(0,1)$

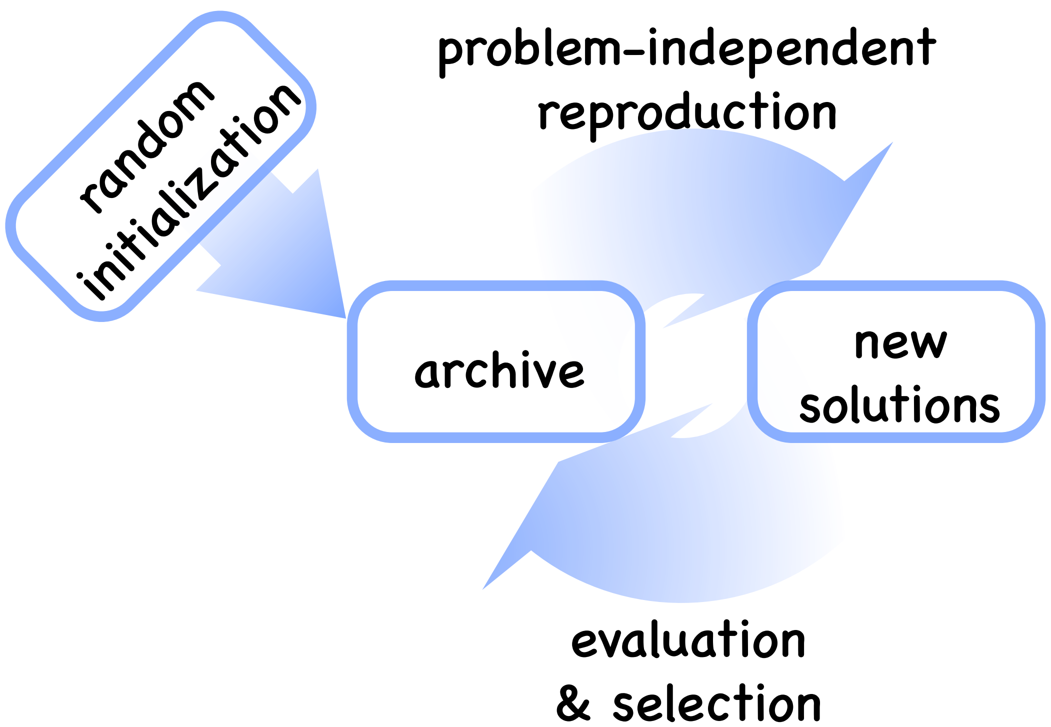
Evolutionary algorithms

Genetic Algorithms [J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

Evolutionary Strategies [I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming [L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms …



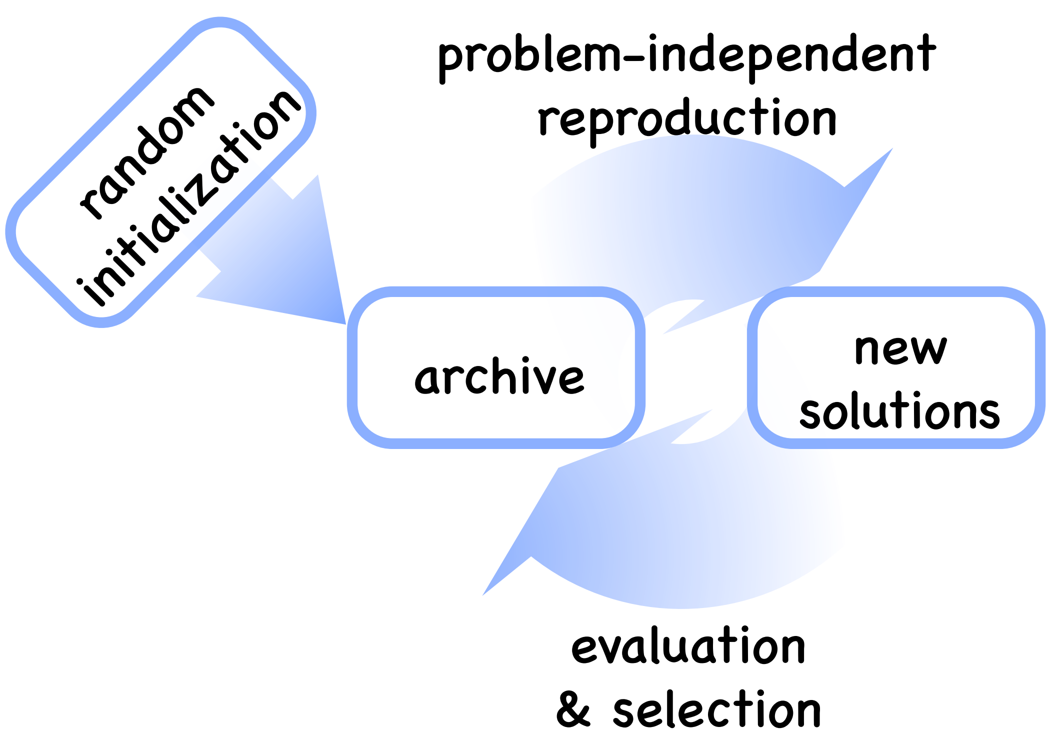
Evolutionary algorithms

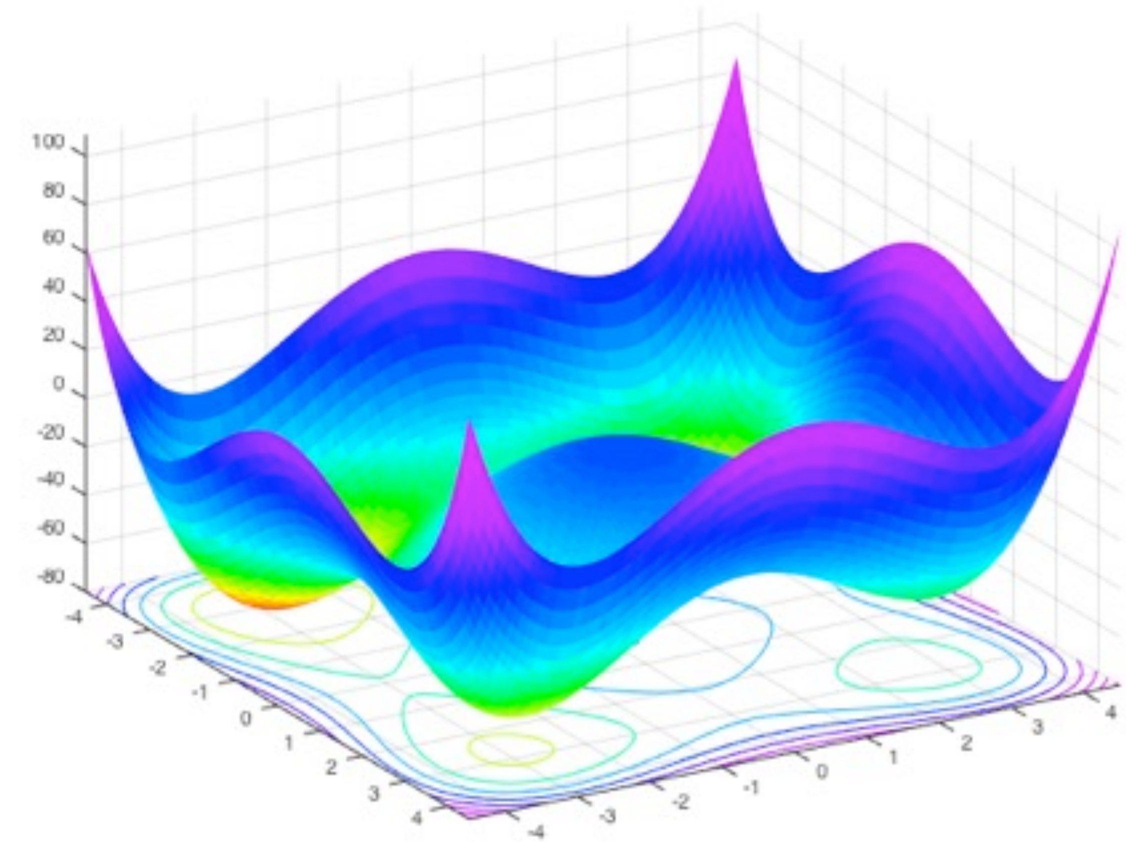
Genetic Algorithms [J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

Evolutionary Strategies [I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming [L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms …





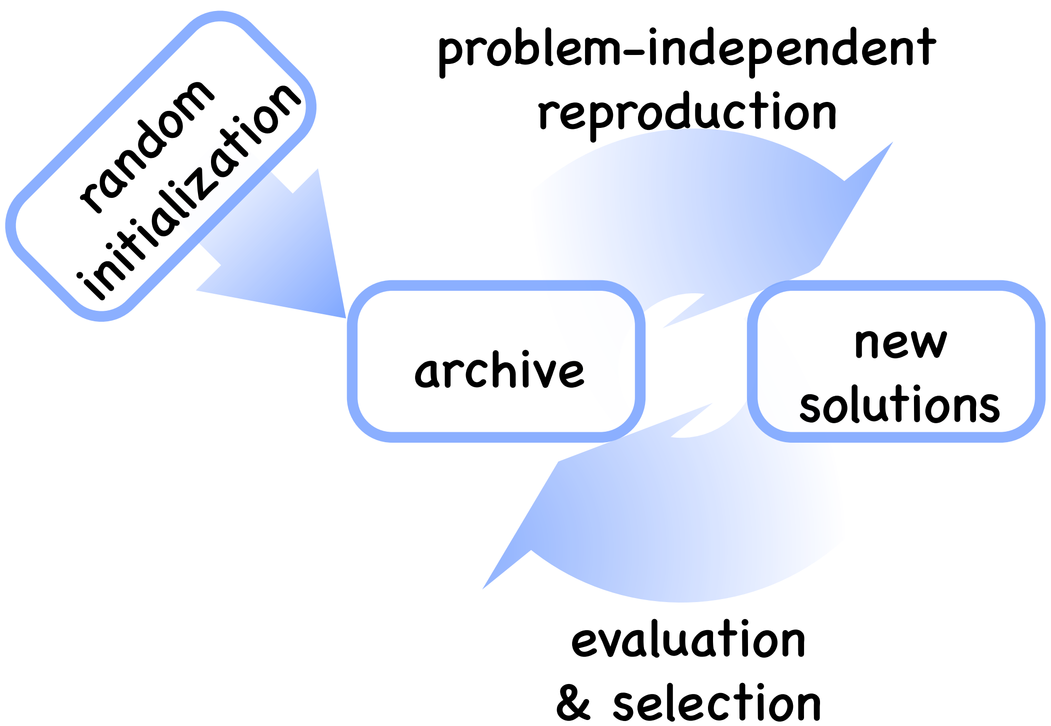
Evolutionary algorithms

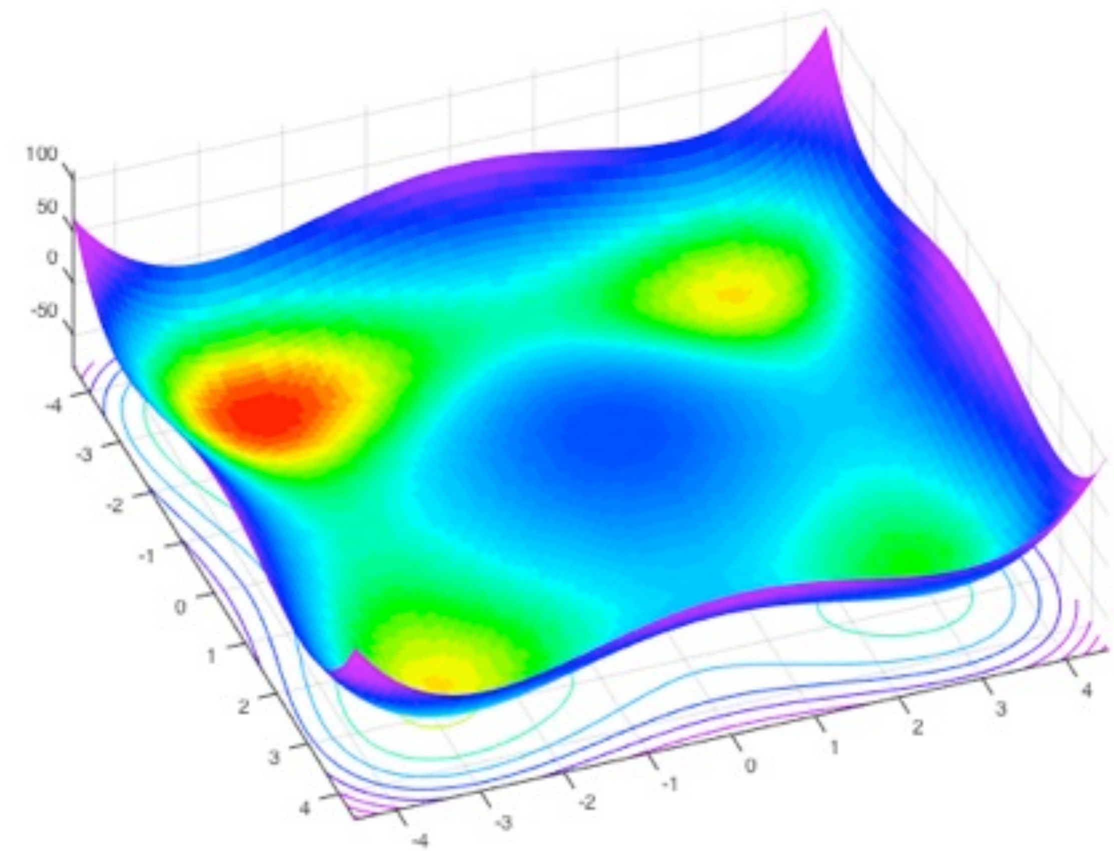
Genetic Algorithms [J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

Evolutionary Strategies [I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming [L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms …





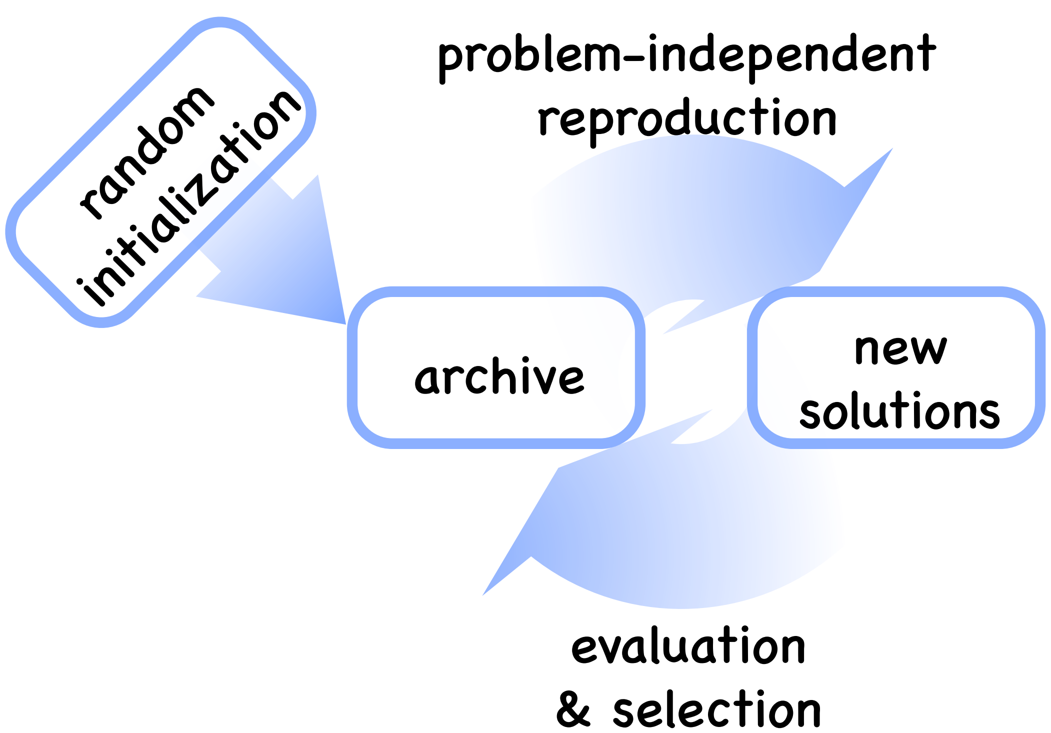
Evolutionary algorithms

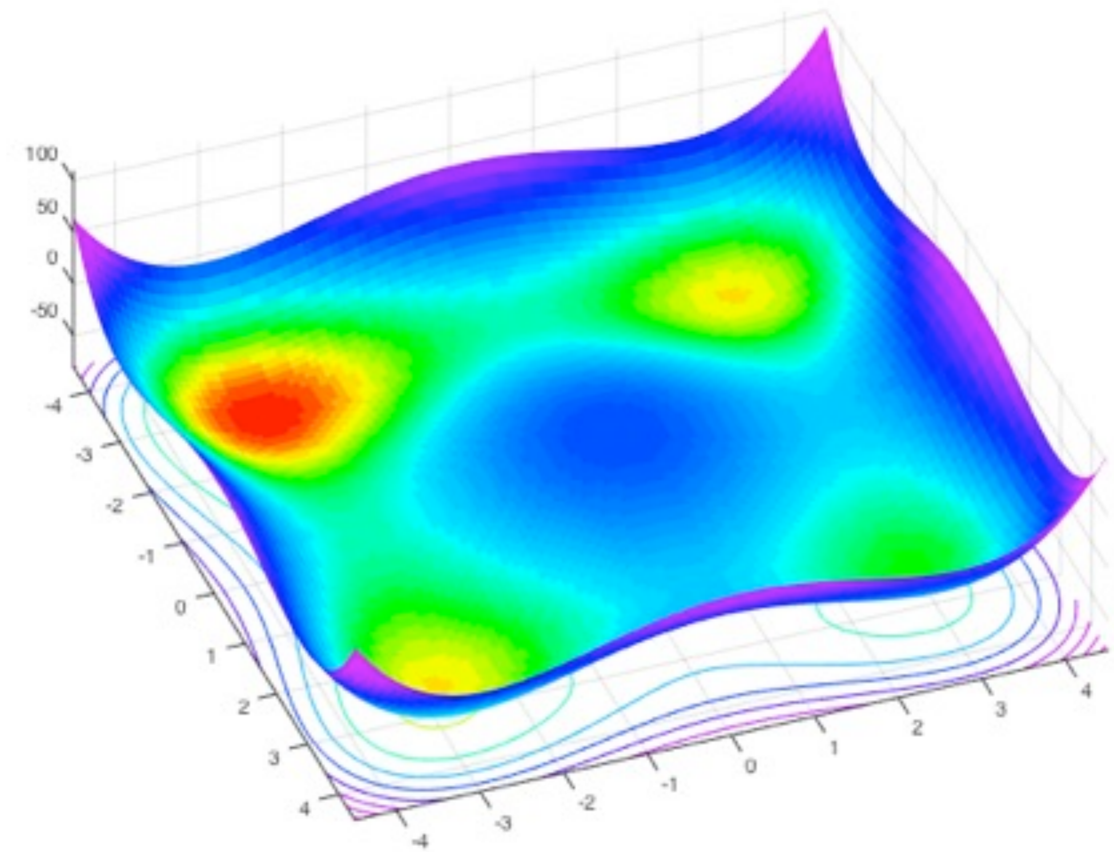
Genetic Algorithms [J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

Evolutionary Strategies [I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming [L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms ...





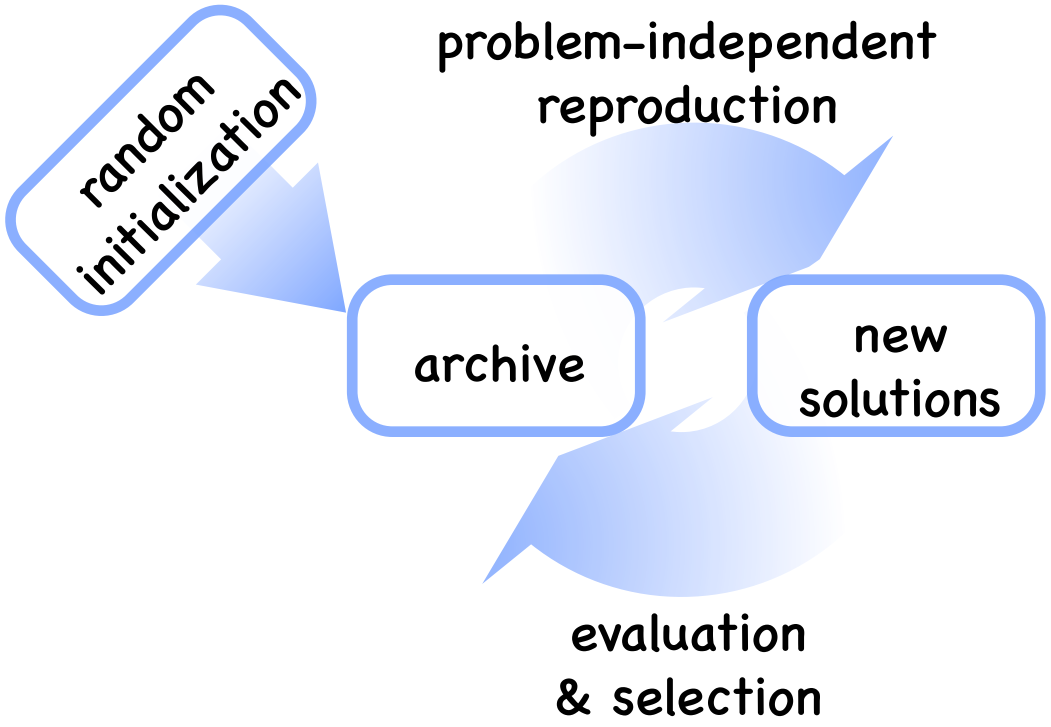
Evolutionary algorithms

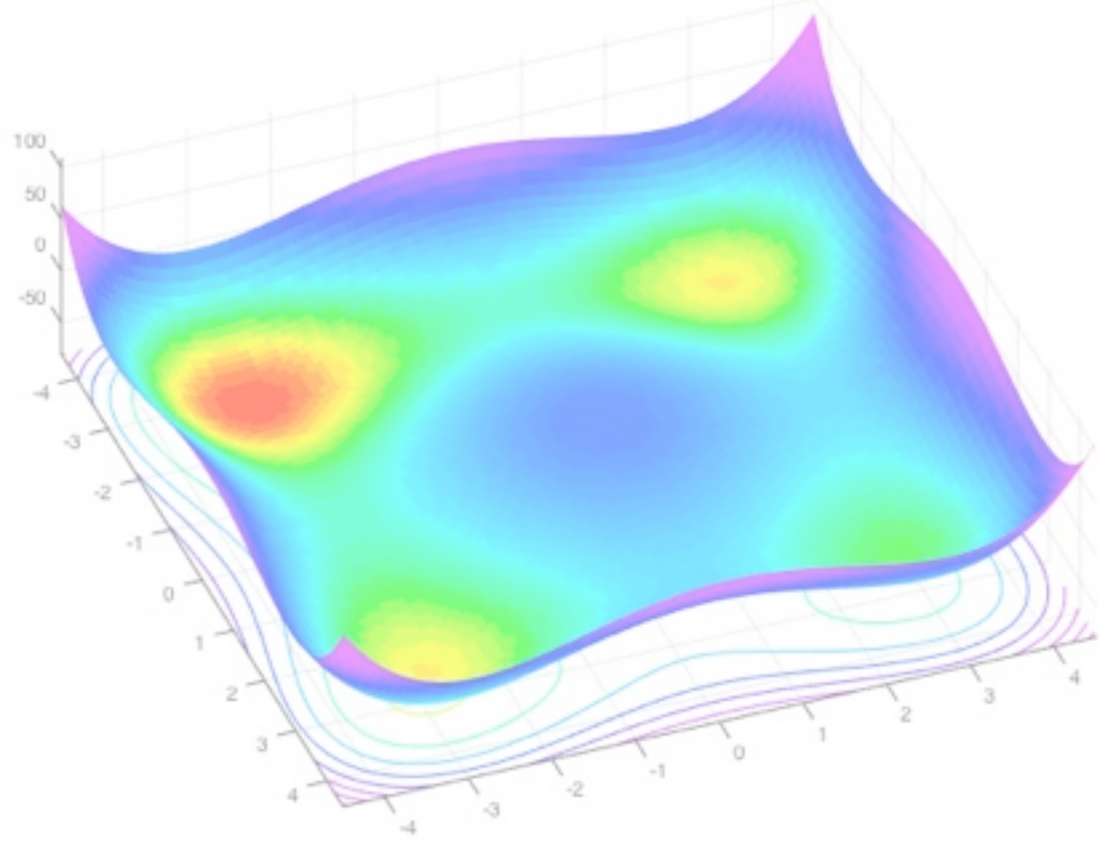
Genetic Algorithms [J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

Evolutionary Strategies [I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming [L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms …





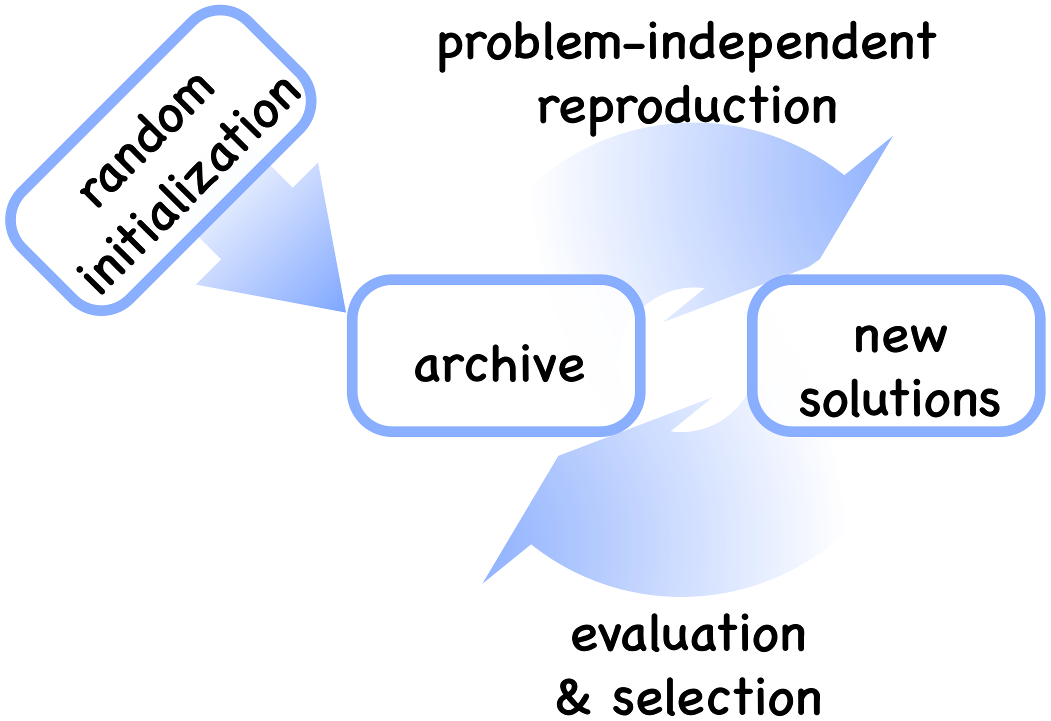
Evolutionary algorithms

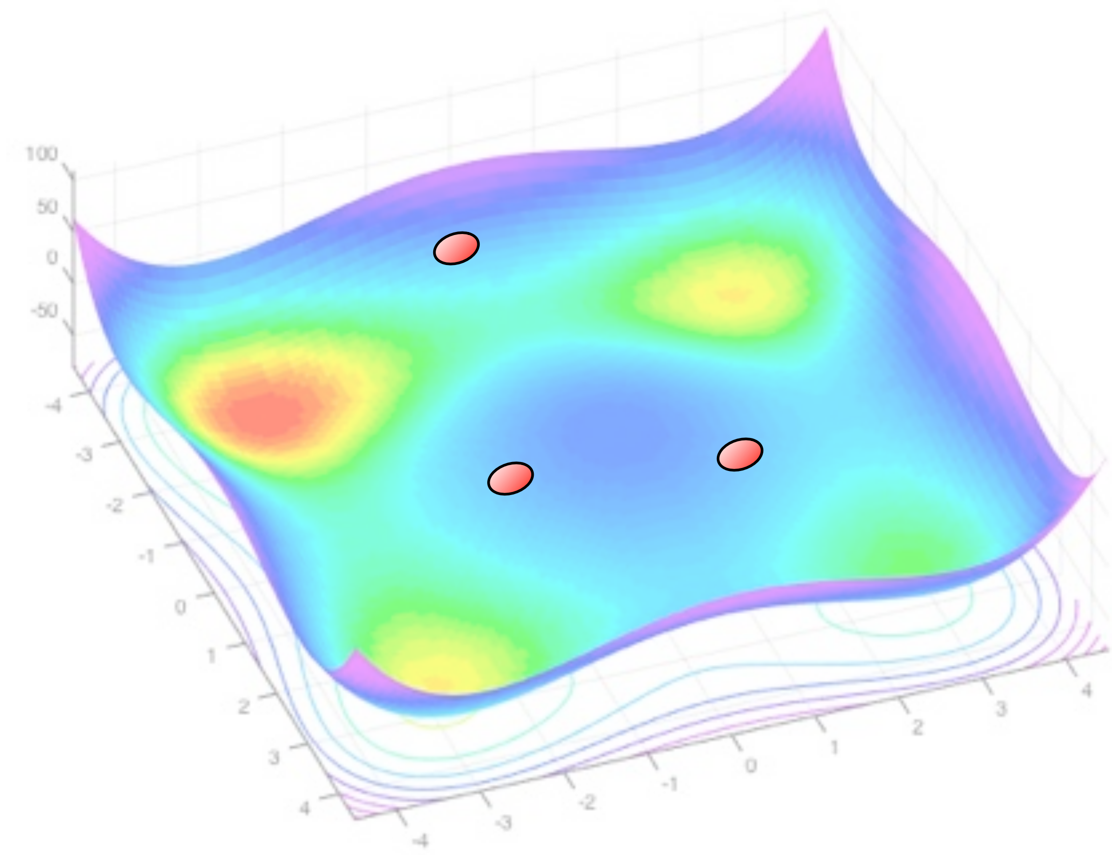
Genetic Algorithms [J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

Evolutionary Strategies [I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming [L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms ...





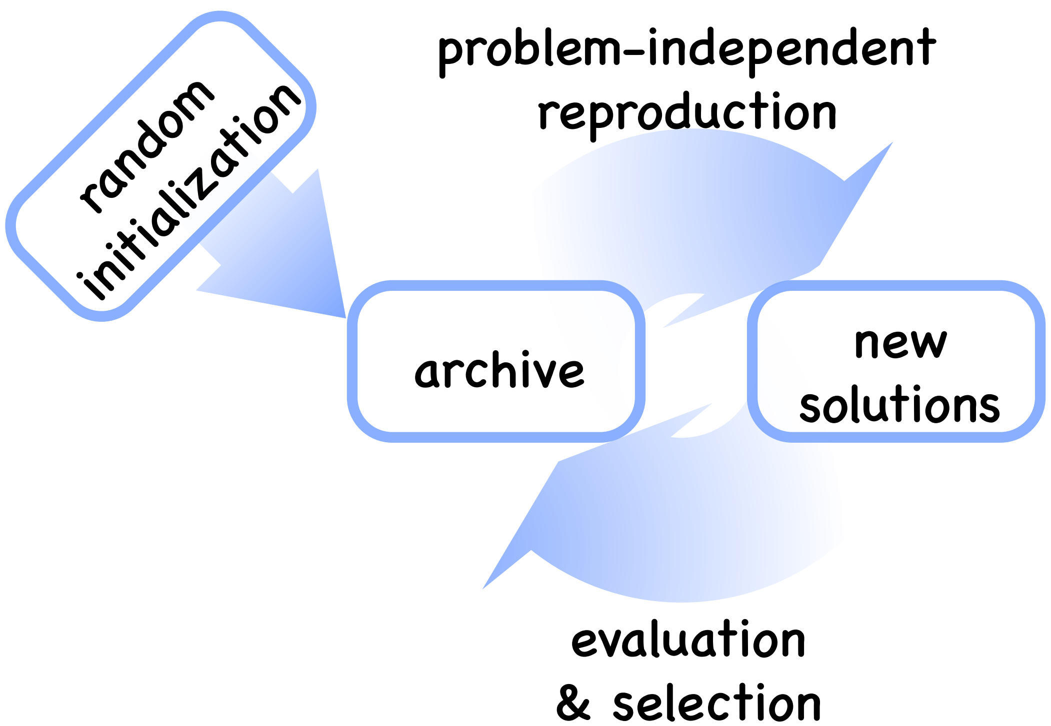
Evolutionary algorithms

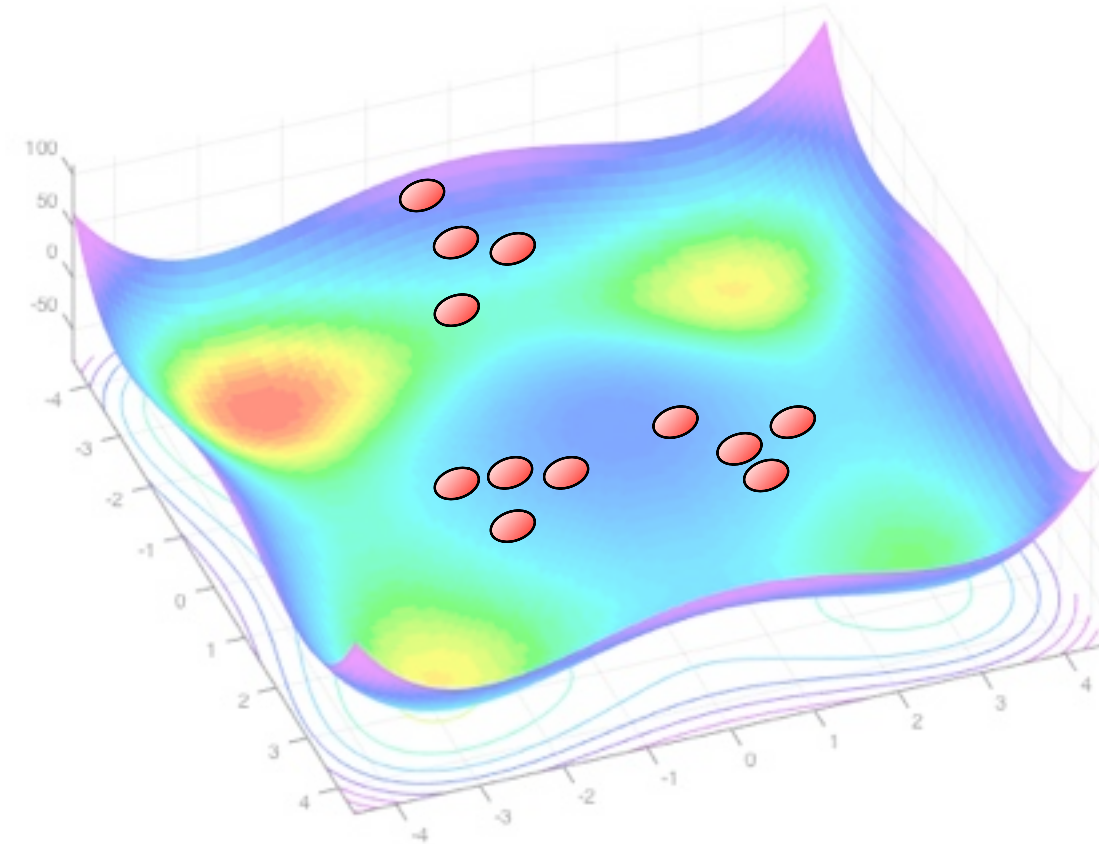
Genetic Algorithms [J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

Evolutionary Strategies [I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming [L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms ...





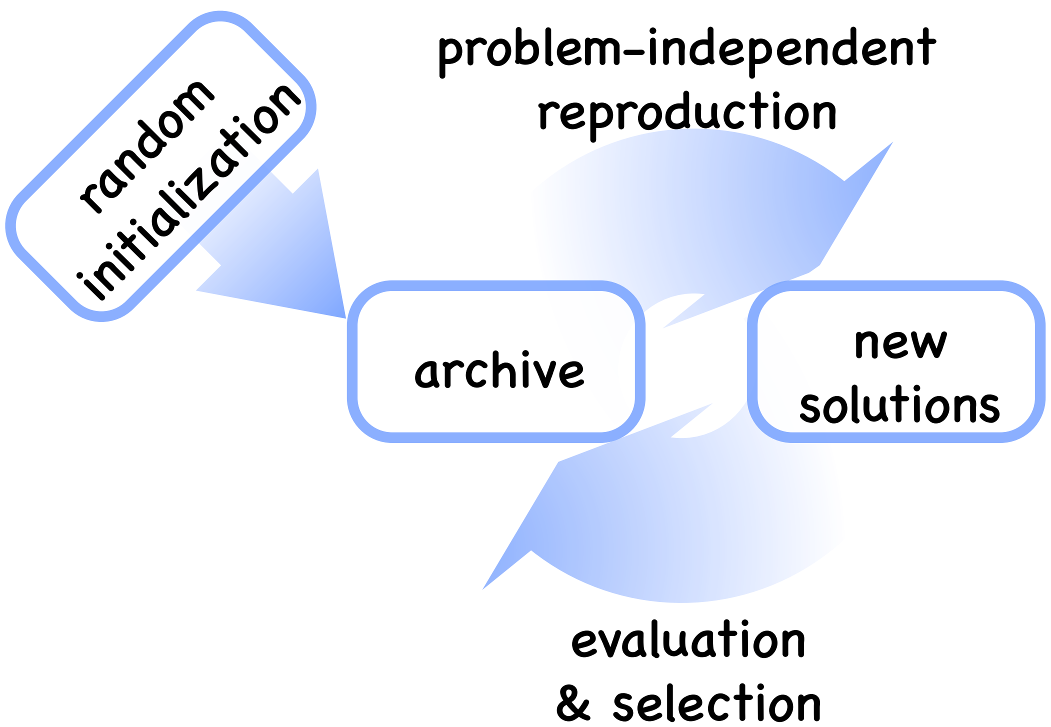
Evolutionary algorithms

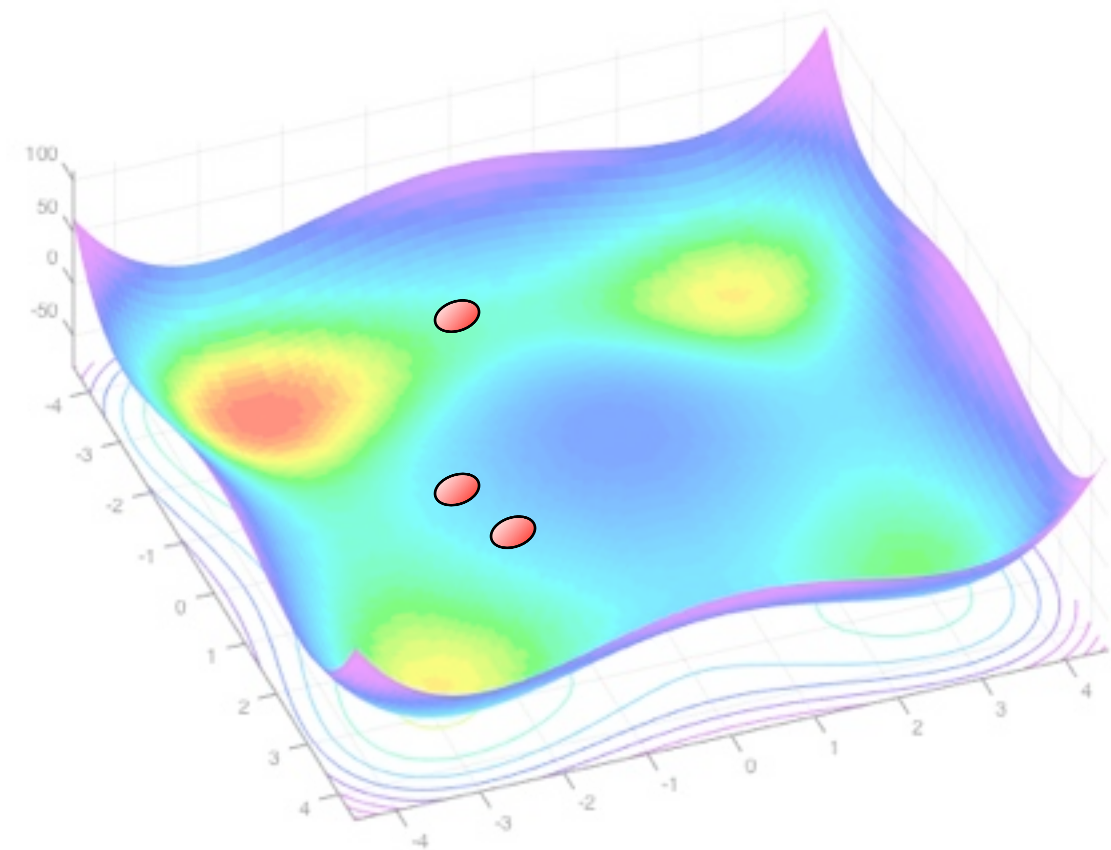
Genetic Algorithms [J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

Evolutionary Strategies [I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming [L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms …





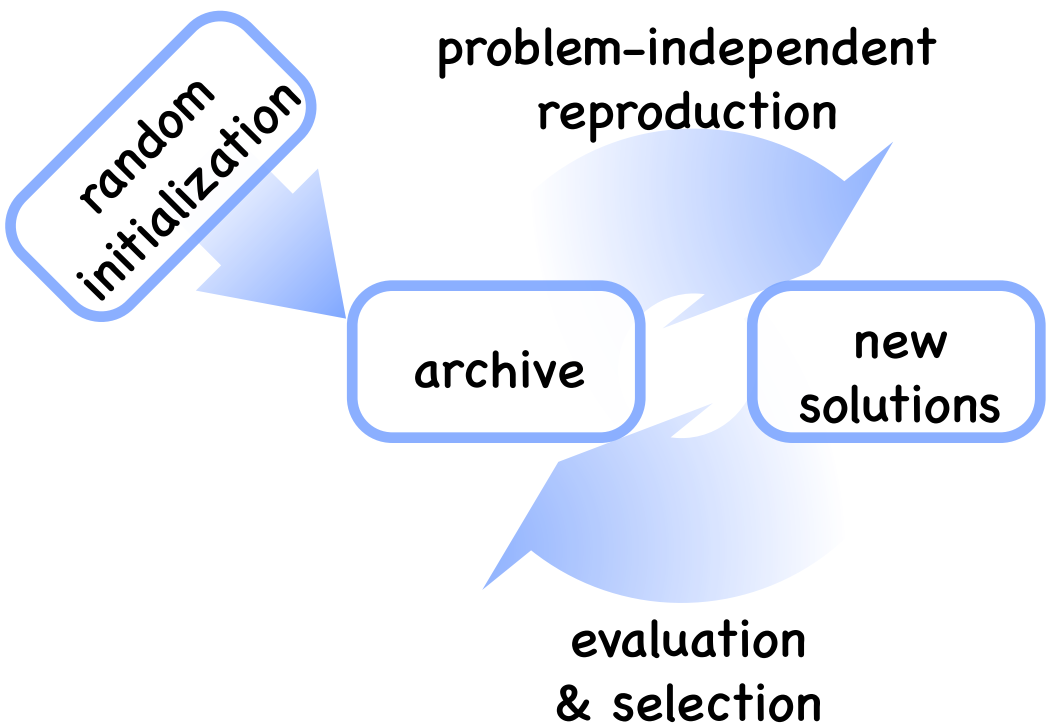
Evolutionary algorithms

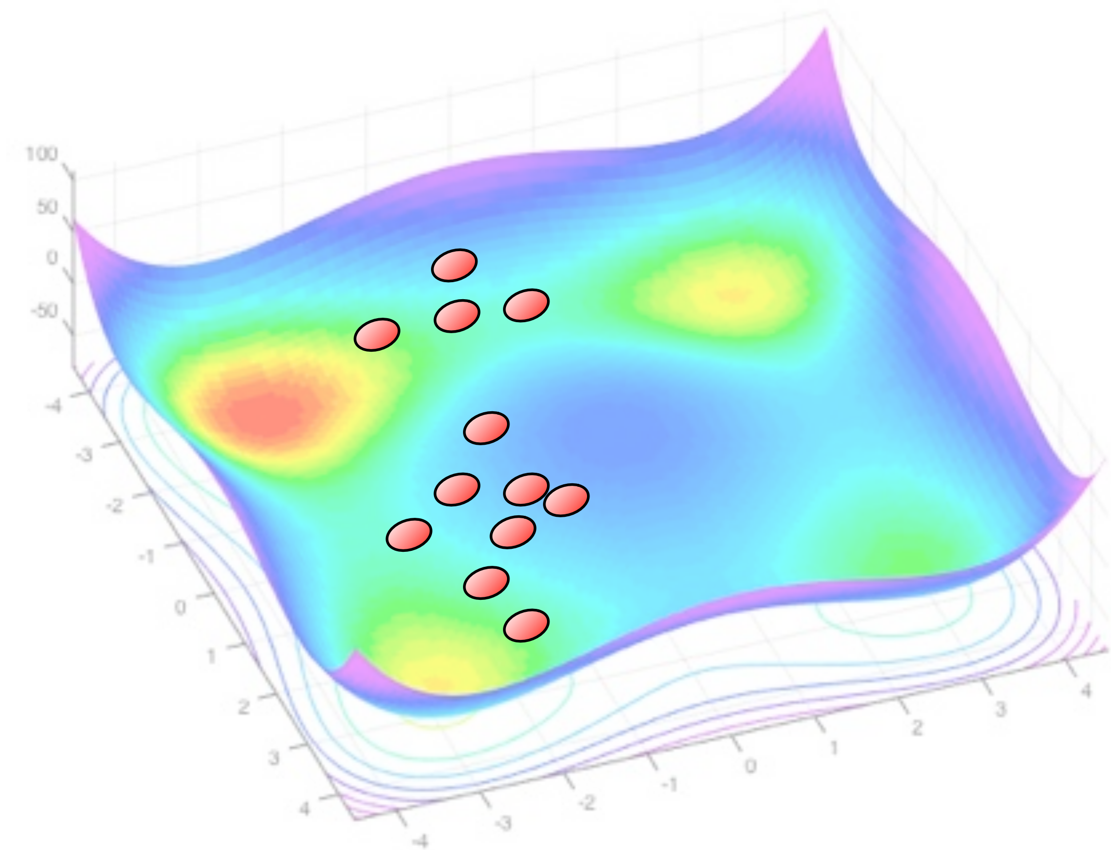
Genetic Algorithms [J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

Evolutionary Strategies [I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming [L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms …





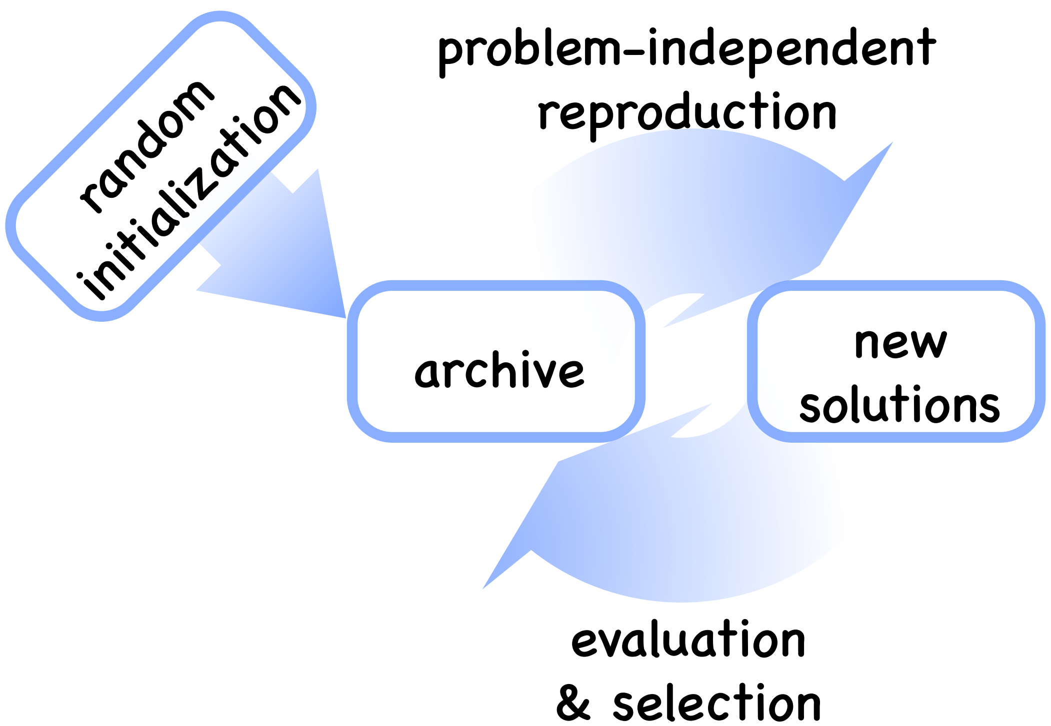
Evolutionary algorithms

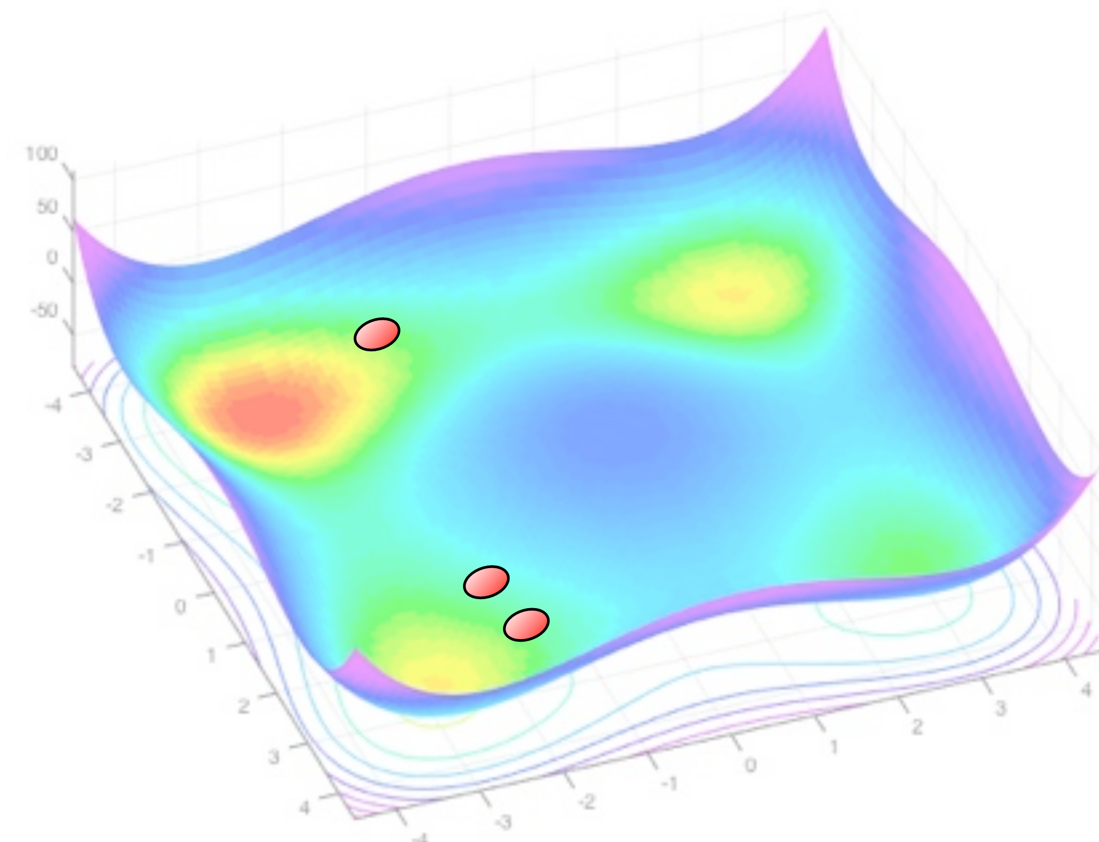
Genetic Algorithms [J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

Evolutionary Strategies [I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming [L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms …





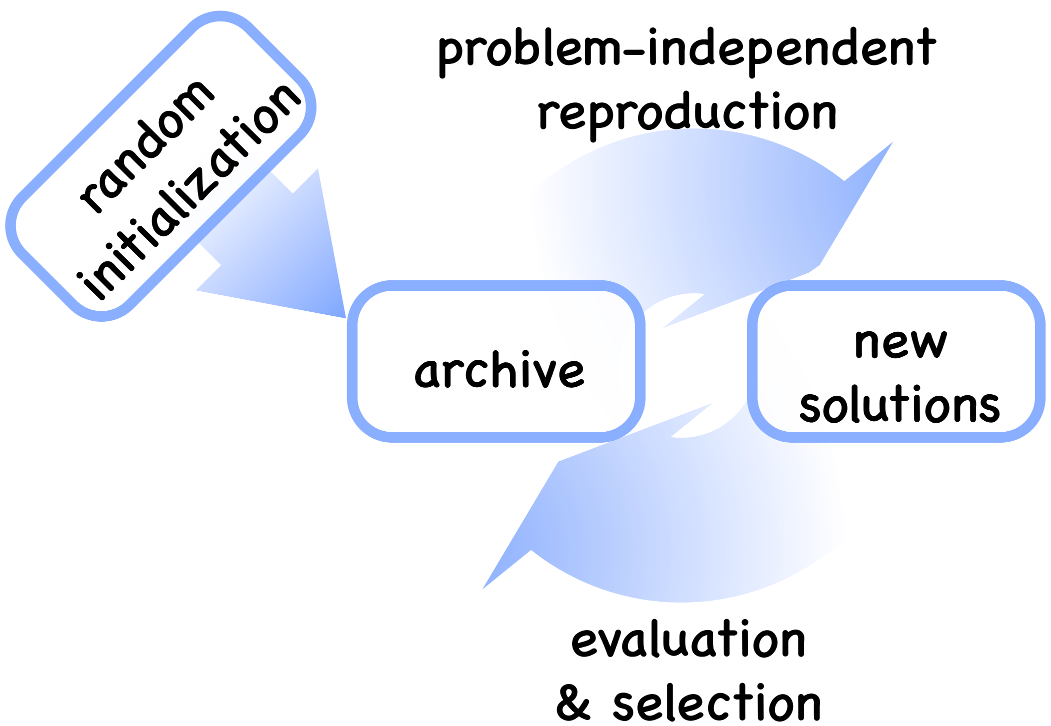
Evolutionary algorithms

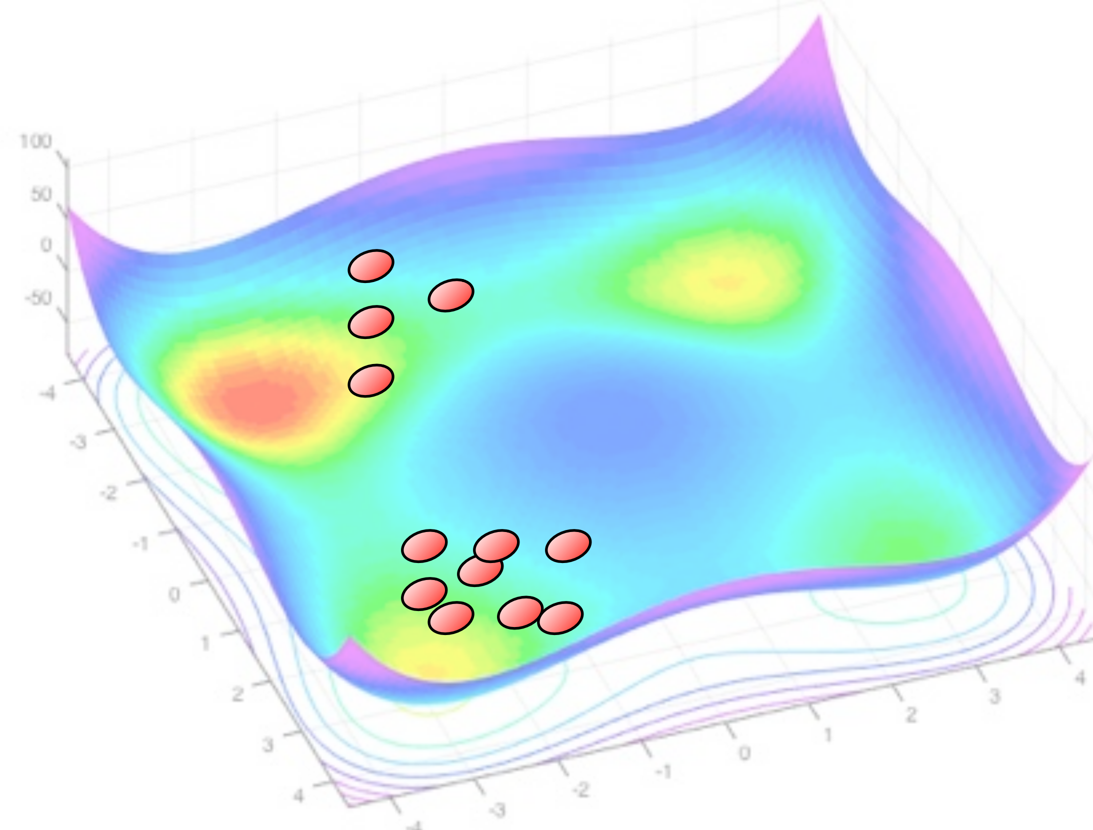
Genetic Algorithms [J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

Evolutionary Strategies [I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming [L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms …





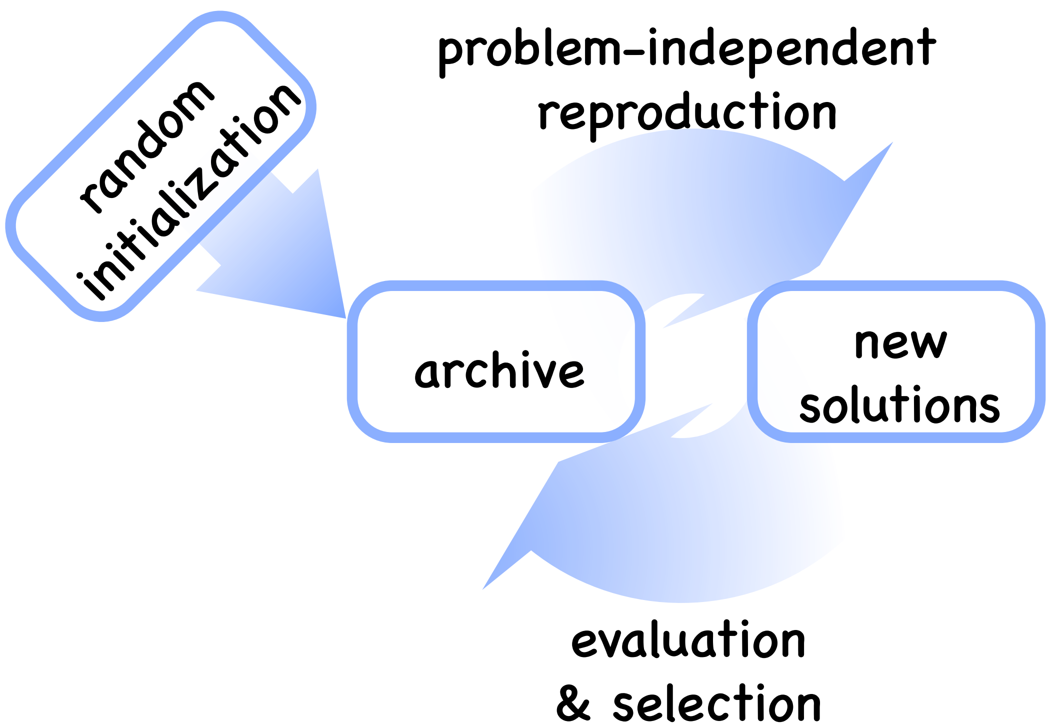
Evolutionary algorithms

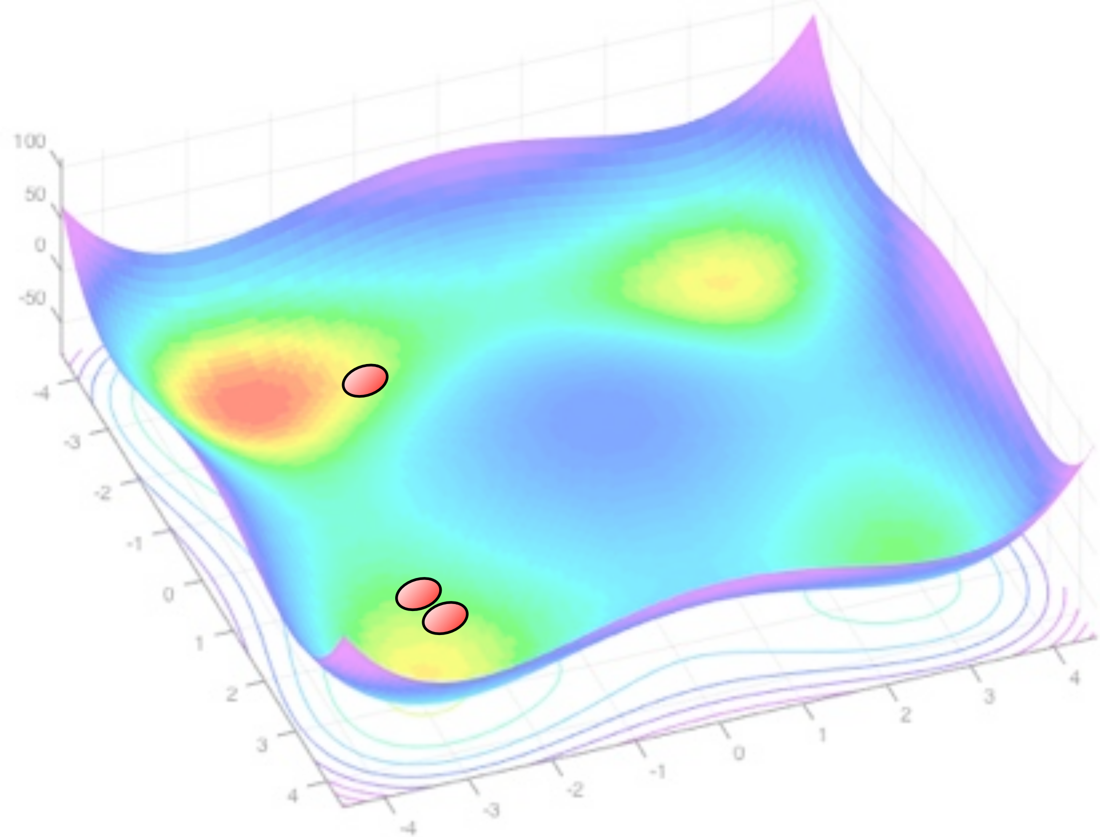
Genetic Algorithms [J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

Evolutionary Strategies [I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming [L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms ...





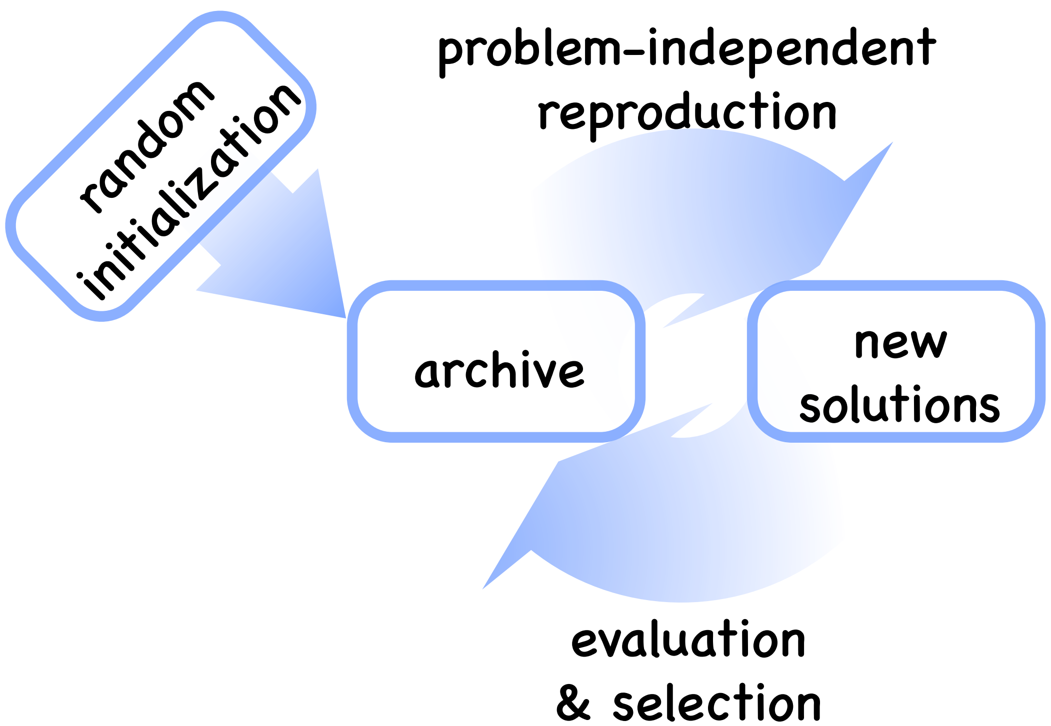
Evolutionary algorithms

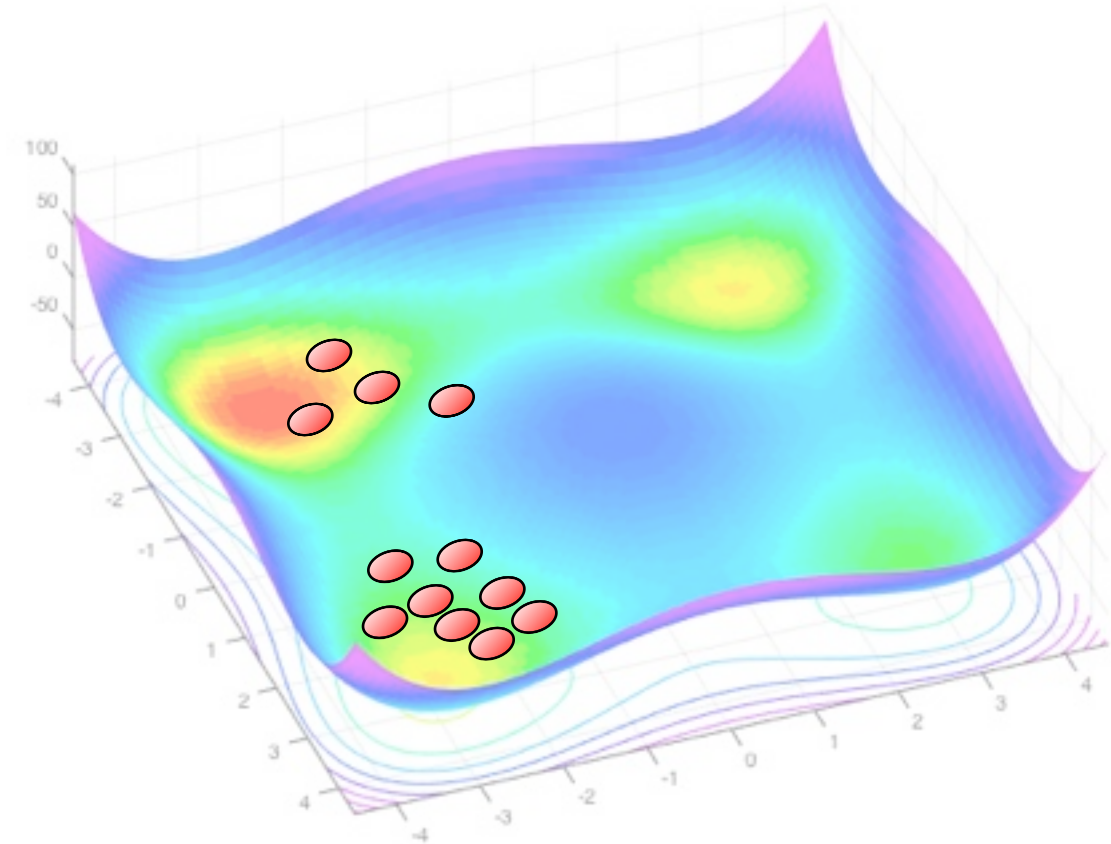
Genetic Algorithms [J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

Evolutionary Strategies [I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming [L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms …





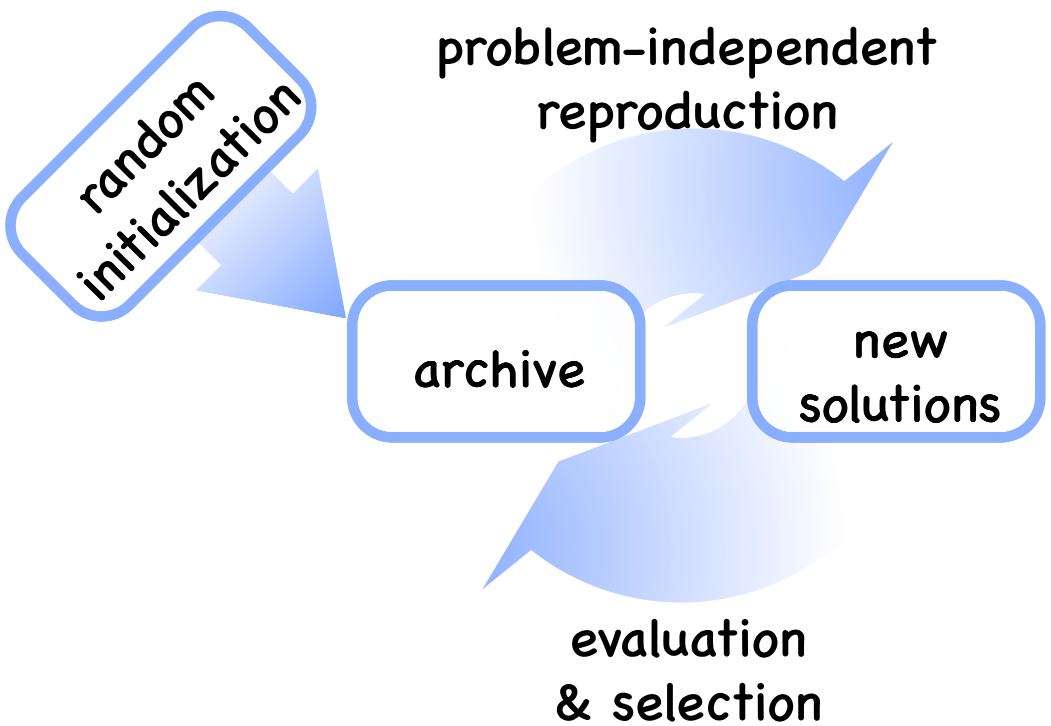
Evolutionary algorithms

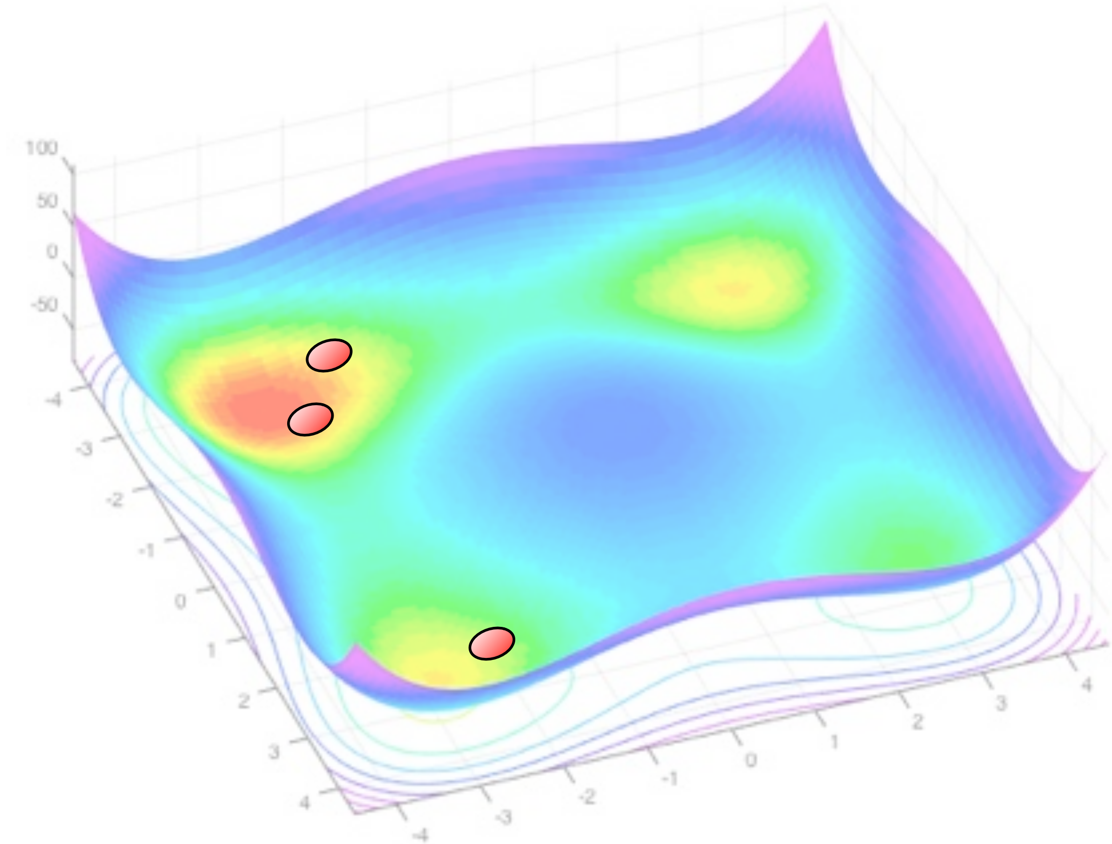
Genetic Algorithms [J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

Evolutionary Strategies [I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming [L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms …





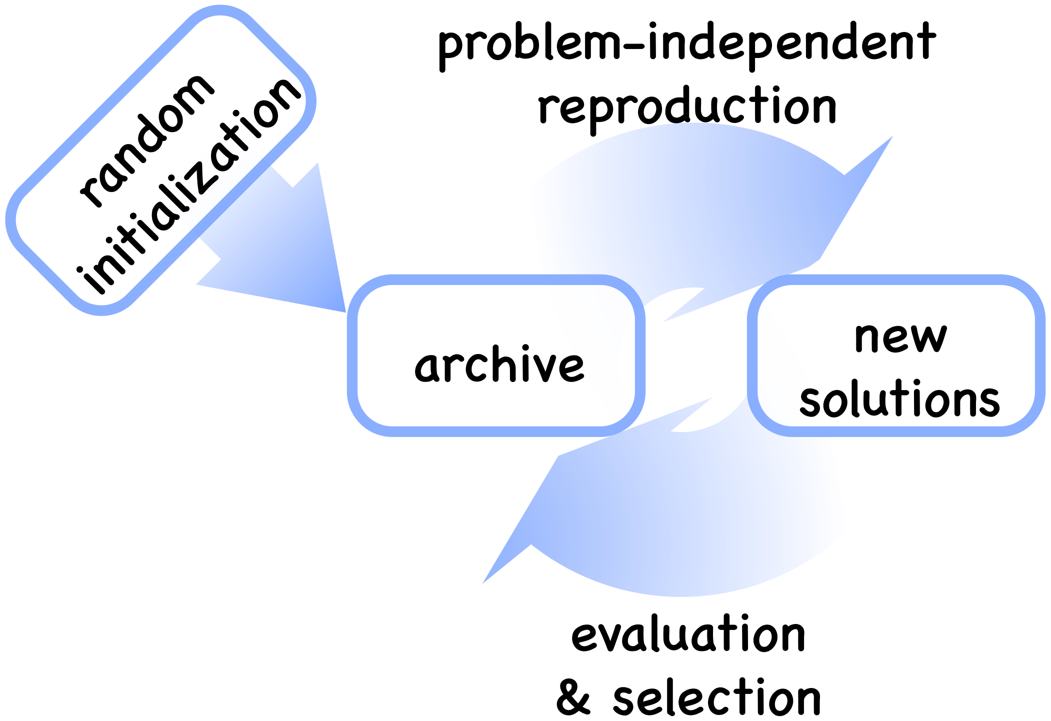
Evolutionary algorithms

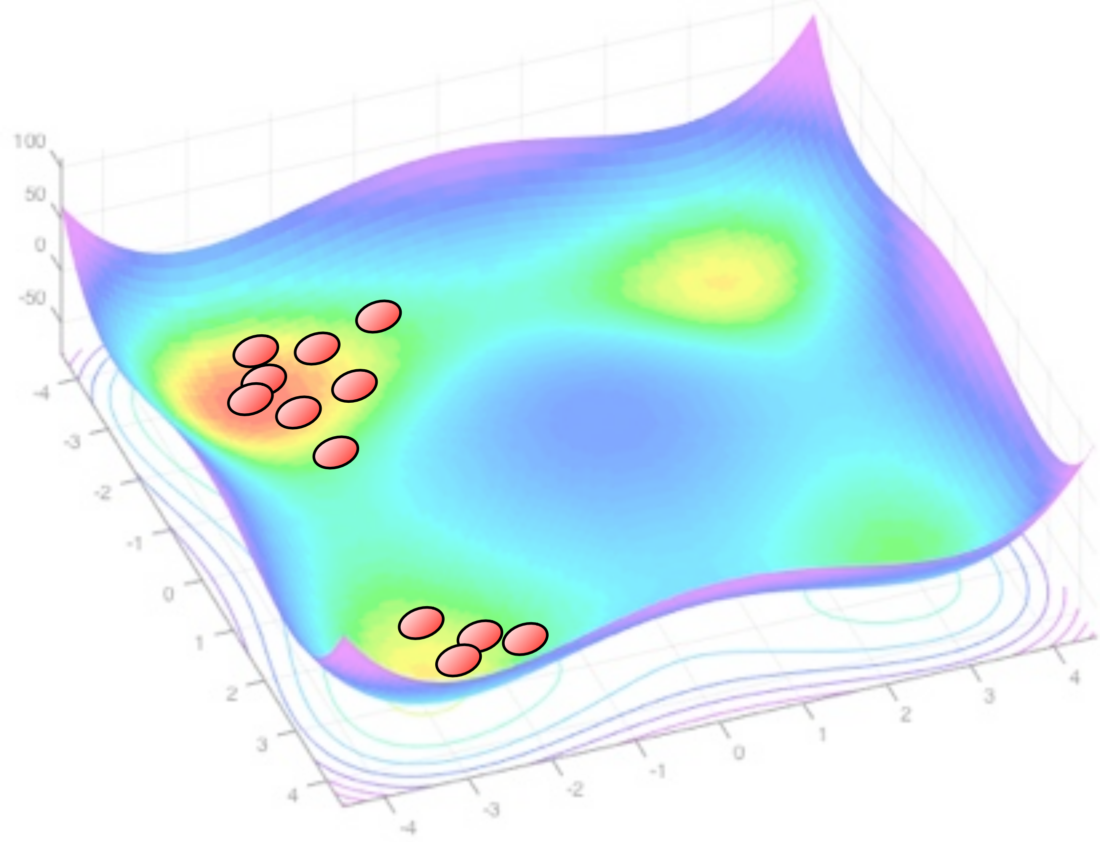
Genetic Algorithms [J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

Evolutionary Strategies [I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming [L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms …





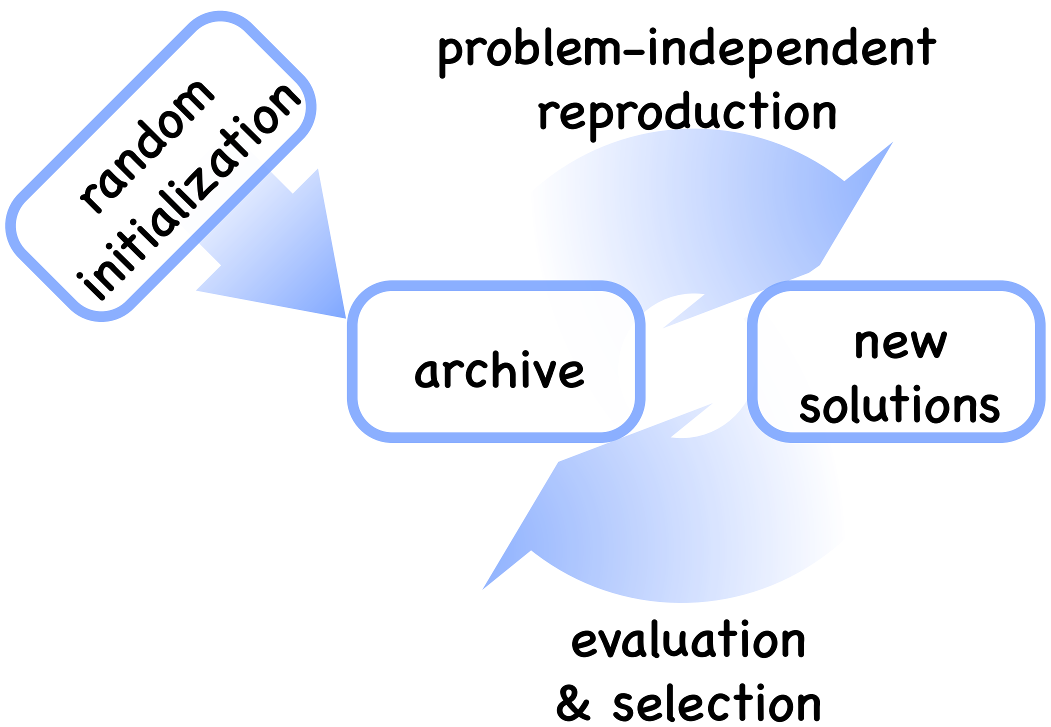
Evolutionary algorithms

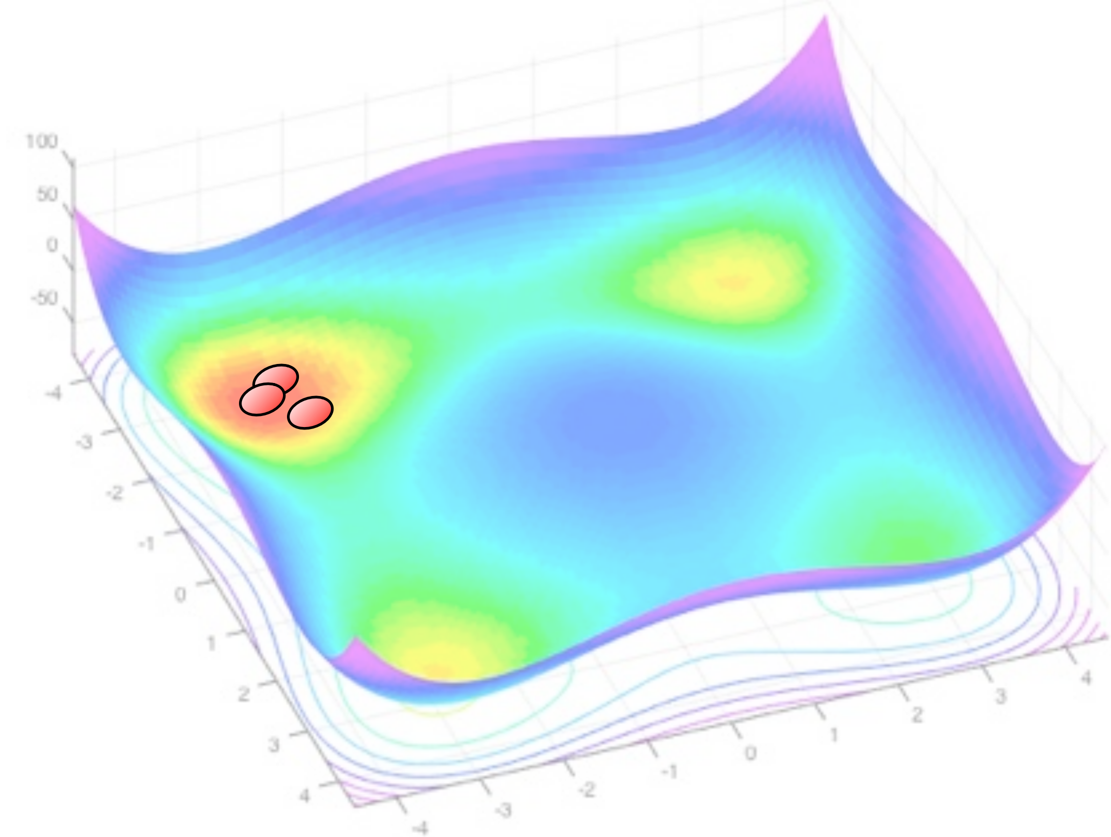
Genetic Algorithms [J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

Evolutionary Strategies [I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming [L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms …





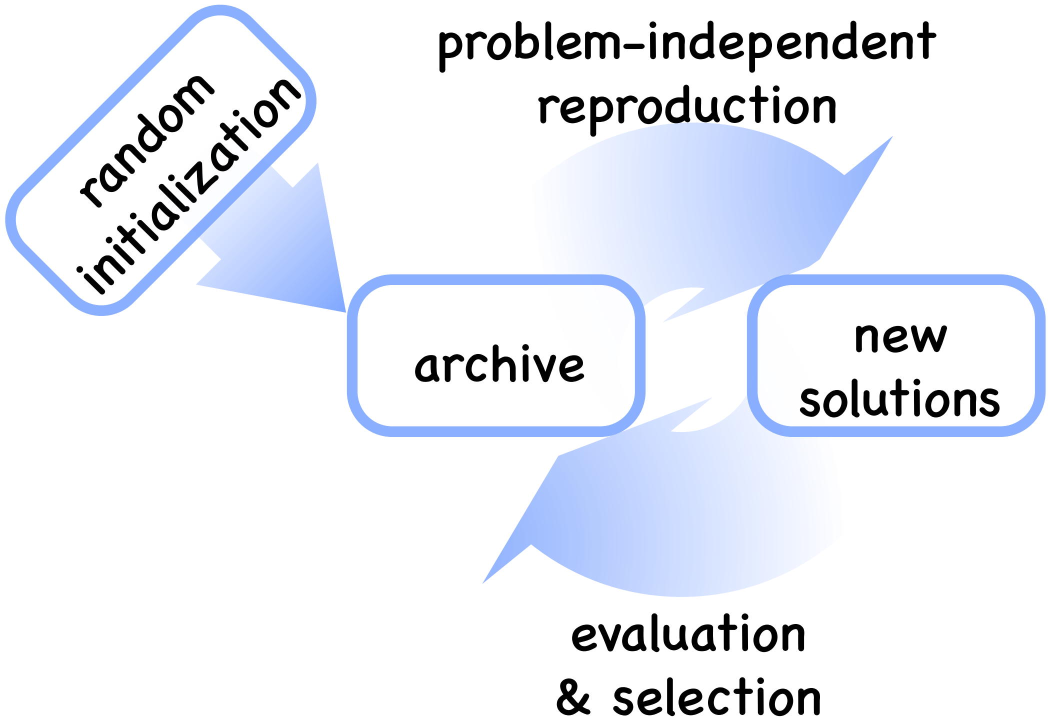
Evolutionary algorithms

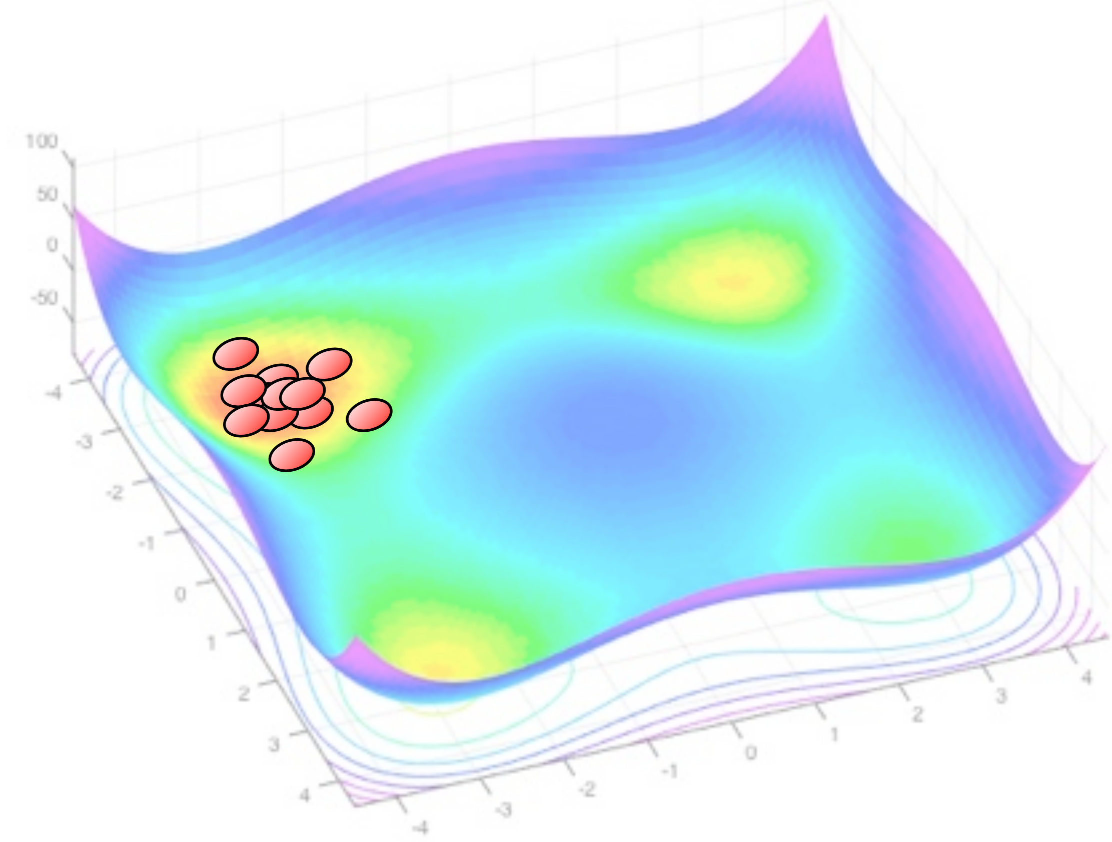
Genetic Algorithms [J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

Evolutionary Strategies [I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming [L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms …





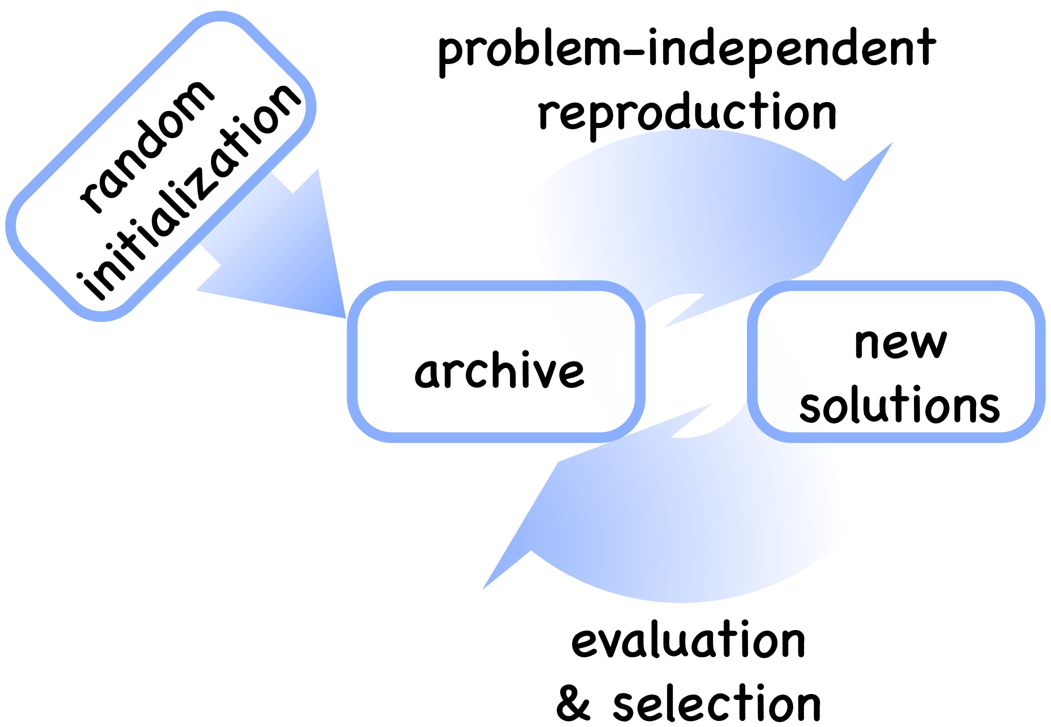
Evolutionary algorithms

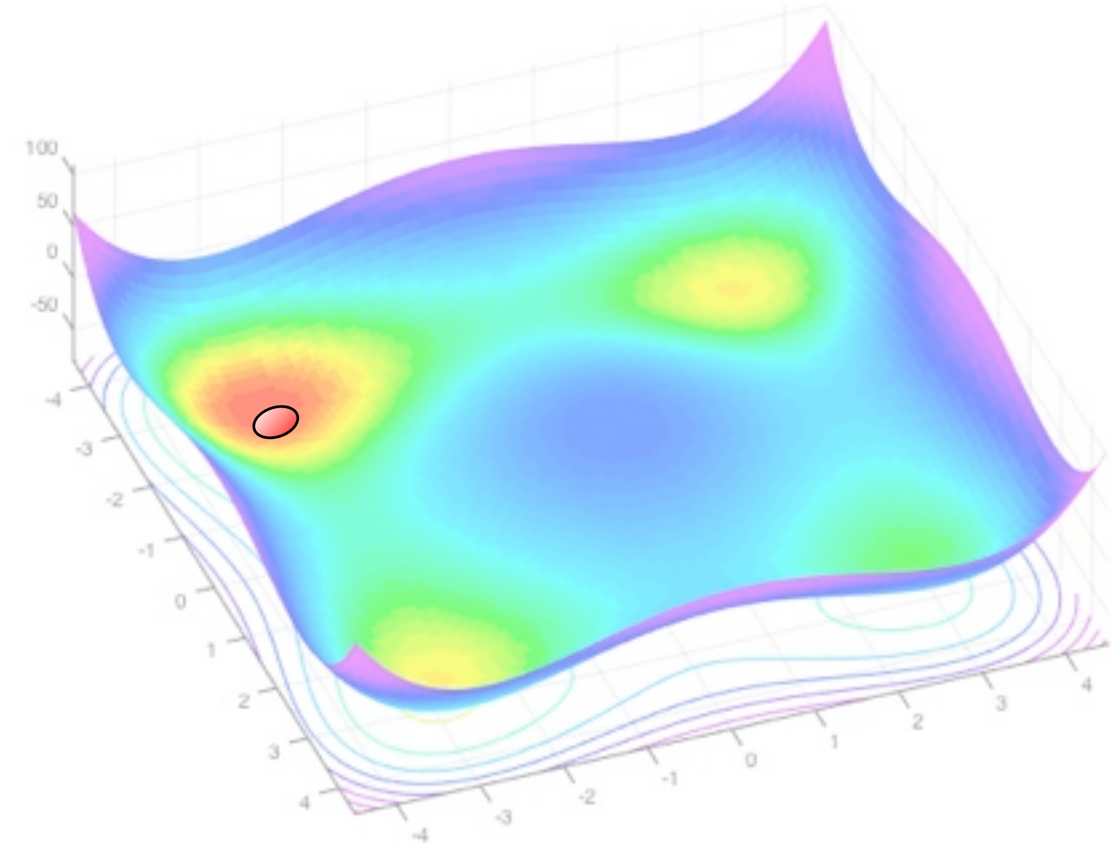
Genetic Algorithms [J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

Evolutionary Strategies [I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming [L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms …





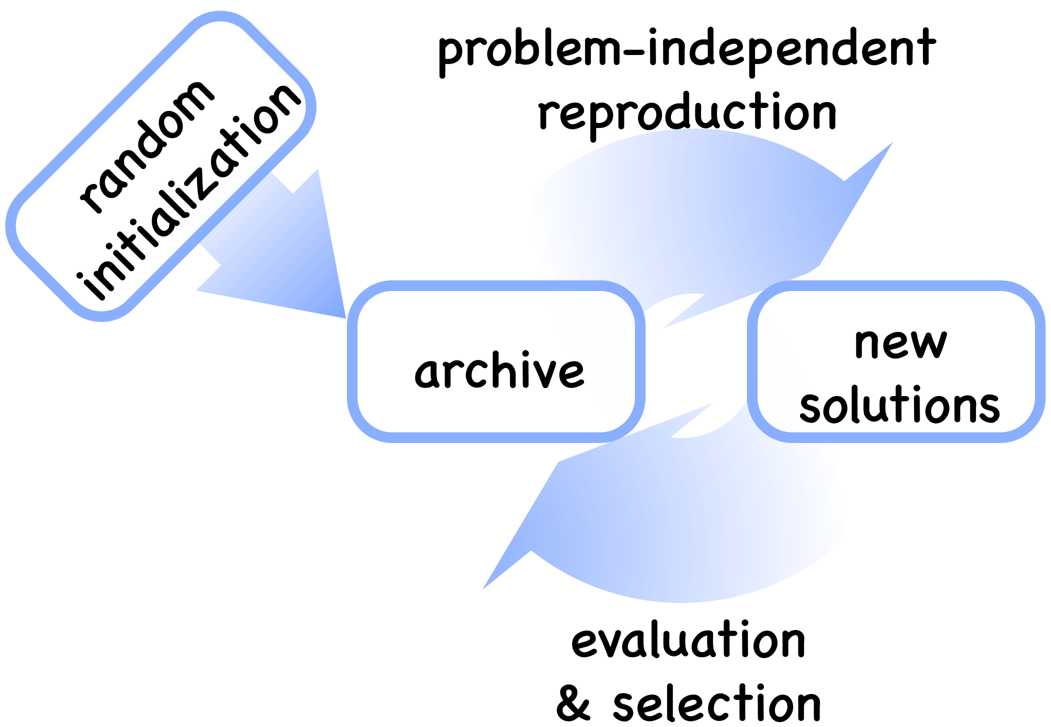
Evolutionary algorithms

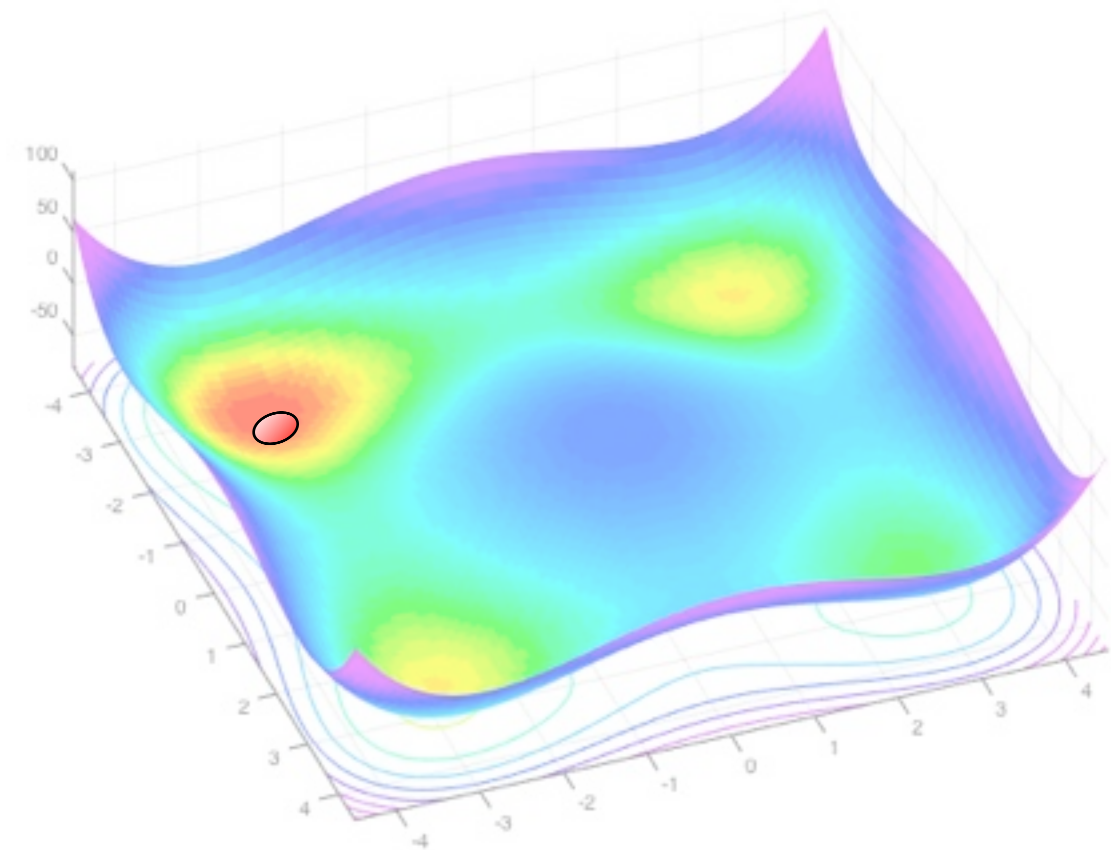
Genetic Algorithms [J. H. Holland. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.]

Evolutionary Strategies [I. Rechenberg. **Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologischen Evolution**. Fromman-Hozlboog Verlag, Stuttgart, 1973.]

Evolutionary Programming [L. J. Fogel, A. J. Owens, M. J. Walsh. **Artificial Intelligence through Simulated Evolution**, John Wiley, 1966.]

and many other nature-inspired algorithms ...





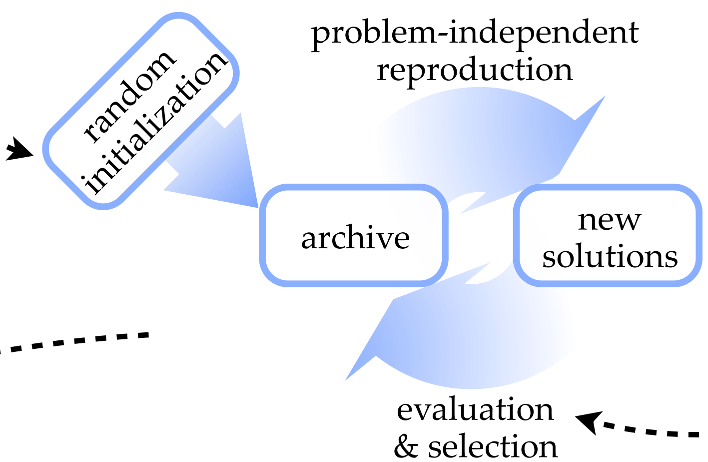
only need to evaluate solutions ⇒ calculate f(x) !

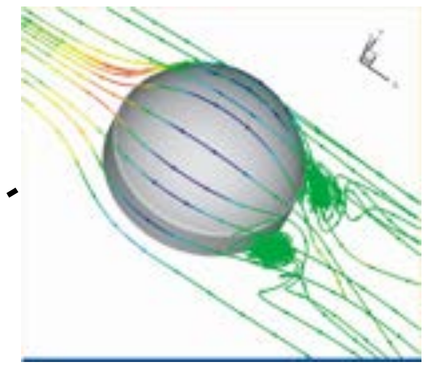
Application of evolutionary algorithms

Series 700

Application of evolutionary algorithms

Series 700

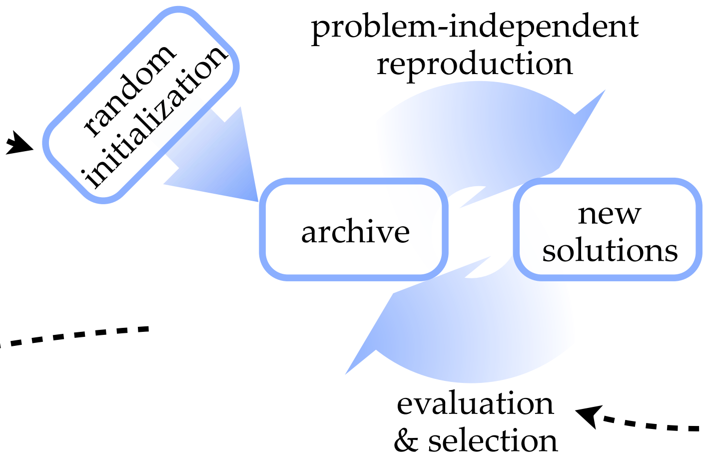


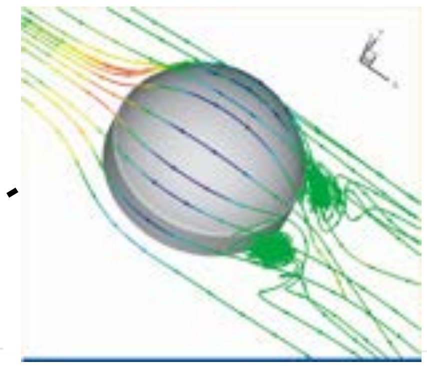


Series N700

Application of evolutionary algorithms

Series 700





Series N700

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*

Abstract

In March 2005, Central Japan Railway Company (JR Central) has completed prototype

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.

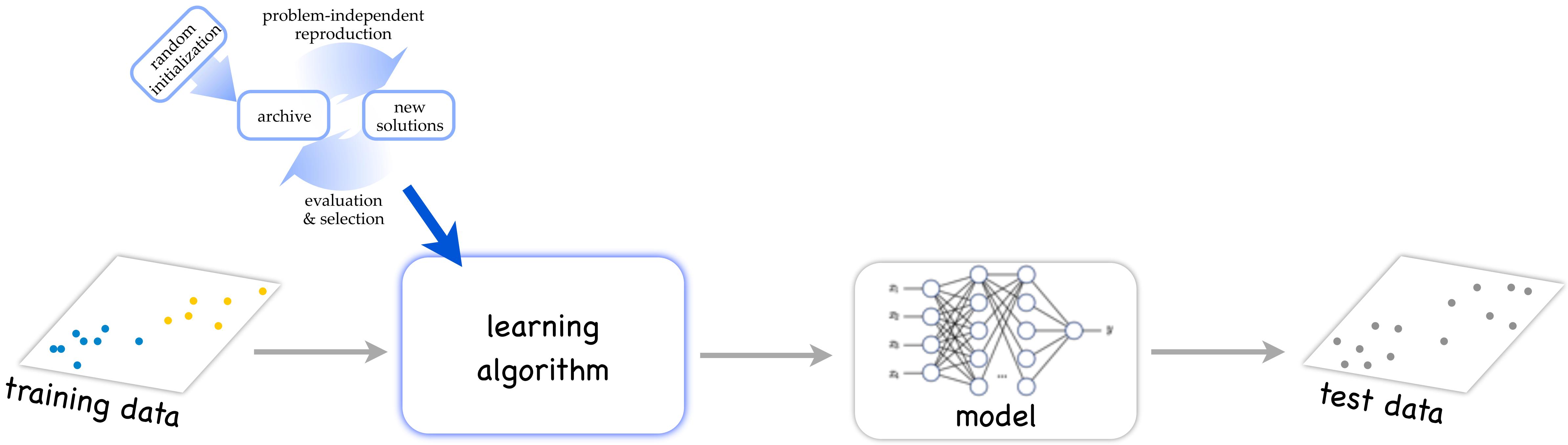
On the Tokaido Shinkansen line, Series N700 cars save 19% energy than Series 700 cars, ... a 30% increase in the output of their traction equipment for higher-speed operation (Fig. 4).

This is a result of adopting the aerodynamically excellent nose shape, reduced running ... thanks to the drastically smoothened car body and under-floor equipment, effective

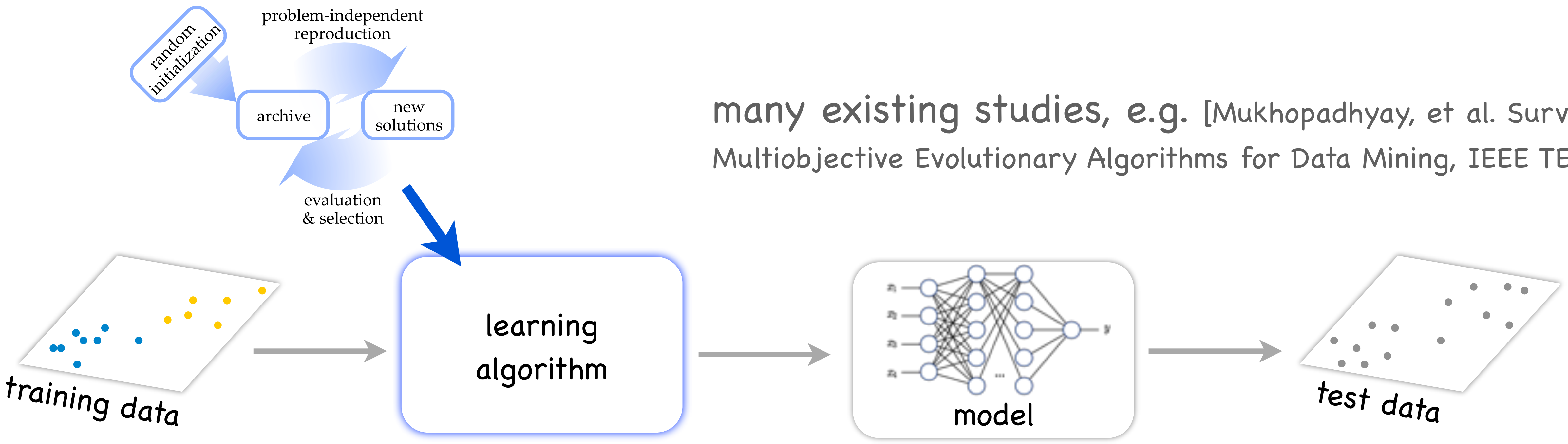
this nose ... has been newly developed ... using the latest analytical technique (i.e. **genetic algorithms**)

N700 cars save **19%** energy ... **30%** increase in the output... This is a result of adopting the ... nose shape

Evolutionary algorithm + machine learning

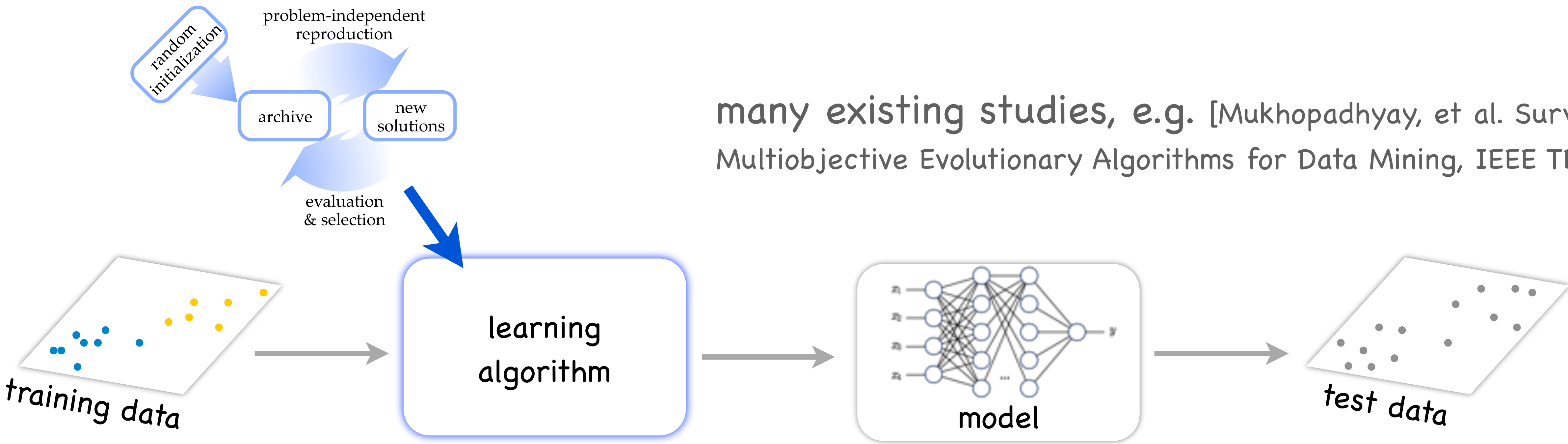


Evolutionary algorithm + machine learning



many existing studies, e.g. [Mukhopadhyay, et al. Survey of Multiobjective Evolutionary Algorithms for Data Mining, IEEE TEC'14]

Evolutionary algorithm + machine learning



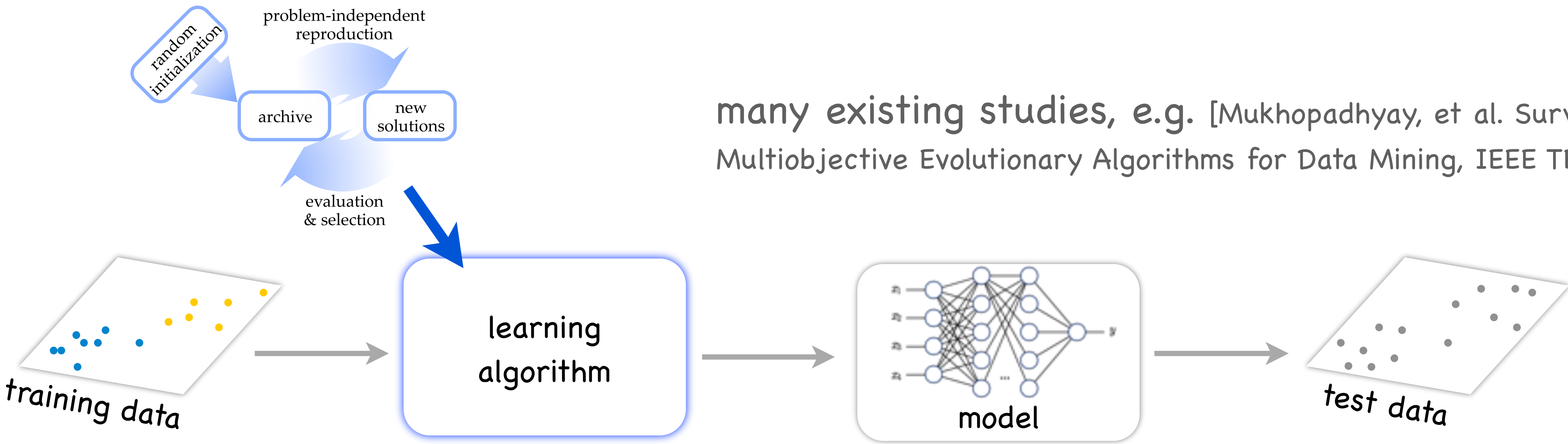
many existing studies, e.g. [Mukhopadhyay, et al. Survey of Multiobjective Evolutionary Algorithms for Data Mining, IEEE TEC'14]

machine learning:
approximate solution
can be sufficient

evolutionary algorithm:
suitable for solving
approximate solutions

Evolutionary algorithm + machine learning

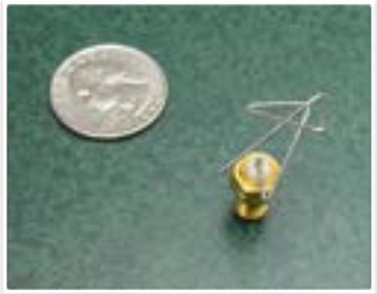
many existing studies, e.g. [Mukhopadhyay, et al. Survey of Multiobjective Evolutionary Algorithms for Data Mining, IEEE TEC'14]



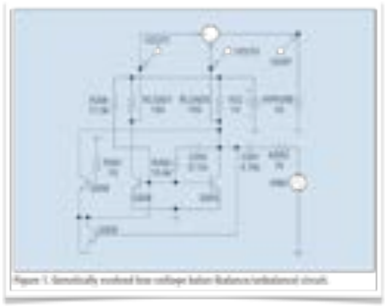
machine learning:
approximate solution can be sufficient

evolutionary algorithm:
suitable for solving approximate solutions

"...save 19% energy ... 30% increase in the output..."

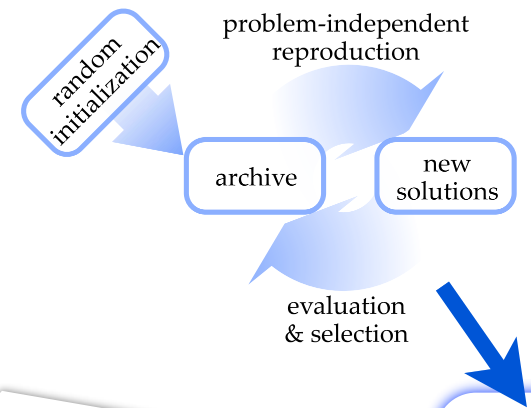


"...38% efficiency ... resulted in 93% efficiency..."

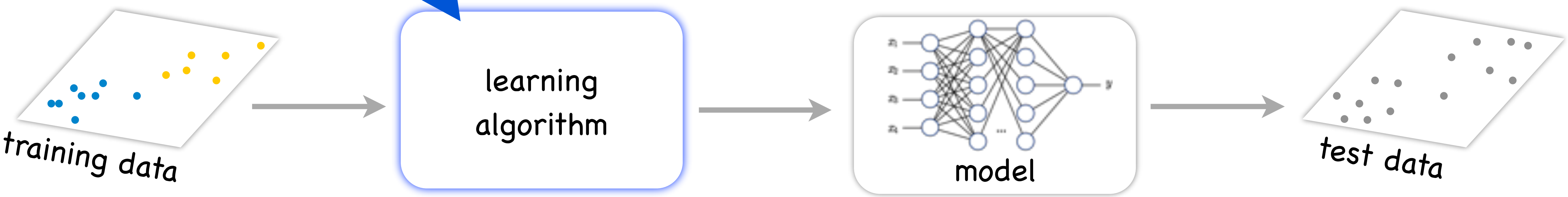


"... roughly a fourfold improvement..."

Evolutionary algorithm + machine learning



many existing studies, e.g. [Mukhopadhyay, et al. Survey of Multiobjective Evolutionary Algorithms for Data Mining, IEEE TEC'14]

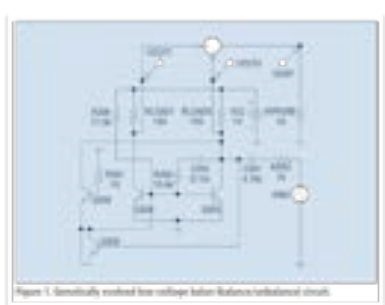


machine learning: approximate solution can be sufficient

evolutionary algorithm: suitable for solving approximate solutions

"...save 19% energy ... 30% increase in the output..."

"...38% efficiency ... resulted in 93% efficiency..."



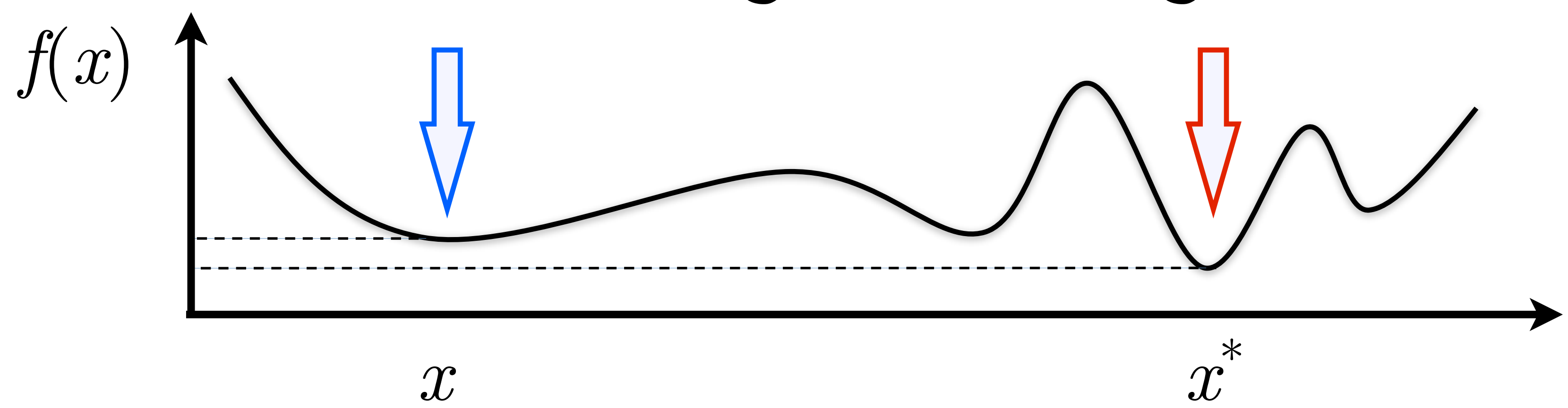
"... roughly a fourfold improvement..."

For maximum matching, a simple EA takes exponential time to find an optimal solution, but $O(n^{2\lceil 1/\epsilon \rceil})$ time to find a $(1+\epsilon)$-approximate solution [Giel and Wegener, STACS'03]

Exact v.s. approximate

approximate optimization: obtain good enough solutions

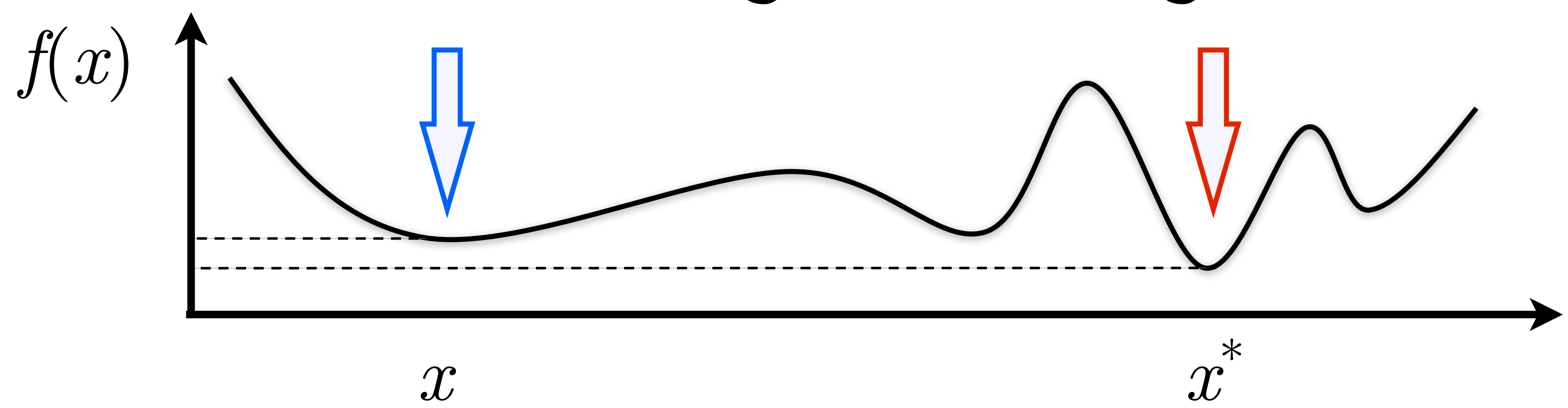
with a close-to-opt. objective value



Exact v.s. approximate

approximate optimization: obtain good enough solutions

with a close-to-opt. objective value



measure of the goodness: (for minimization)

approximation ratio:

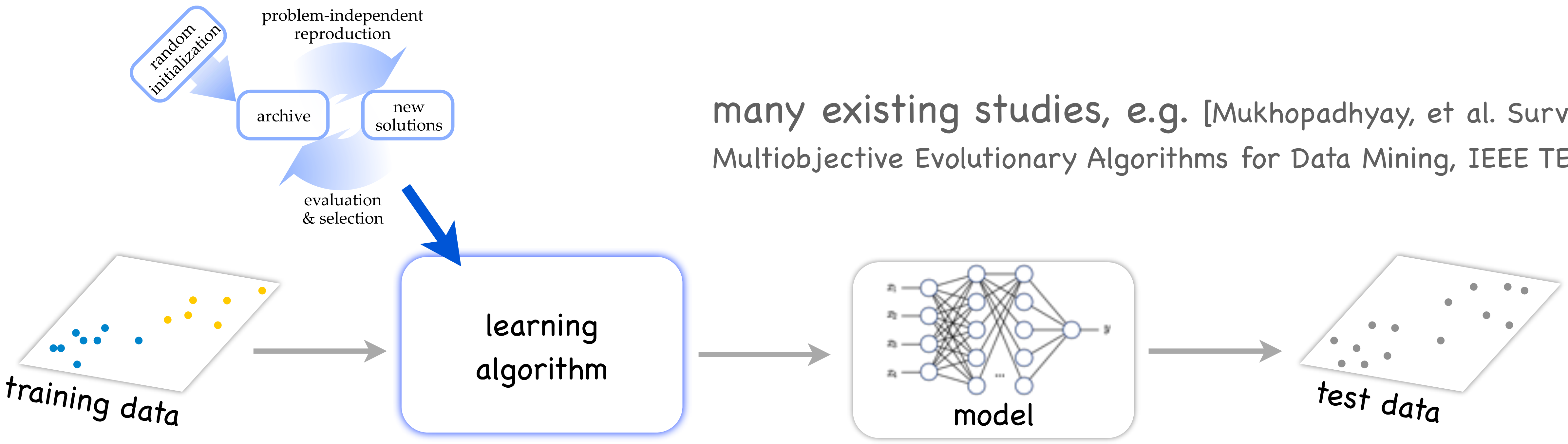
$$\frac{f(x)}{f(x^*)} \geq 1$$ is called the approximation ratio of x

x is an r-approximate solution

simple regret:

$$f(x) - f(x^*) \geq 0$$ is called the simple regret of x

Evolutionary algorithm + machine learning



many existing studies, e.g. [Mukhopadhyay, et al. Survey of Multiobjective Evolutionary Algorithms for Data Mining, IEEE TEC'14]

Challenges:

- **theoretical supports**
- competitors of domain-specific algorithms
- large-scale optimization tasks
- ...

Outline

Subset selection problem

and Pareto optimization

Local Lipschitz continuous problem

and classification-based optimization

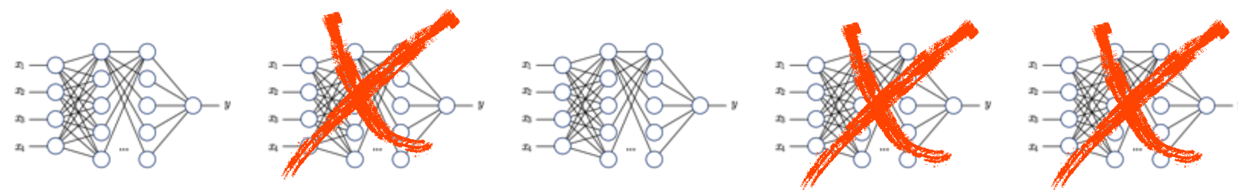
Selection problems in learning

Selection problems in learning

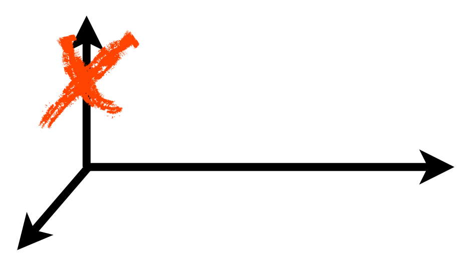
model selection

Selection problems in learning

model selection

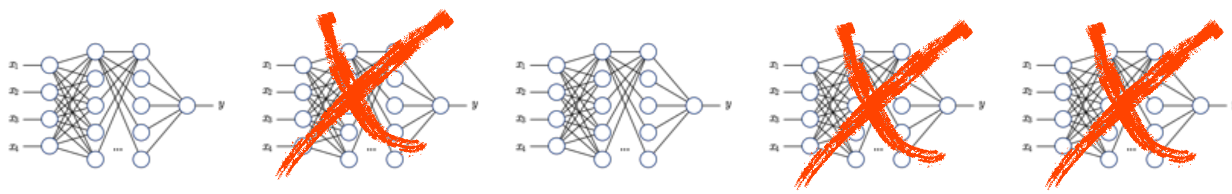


feature selection

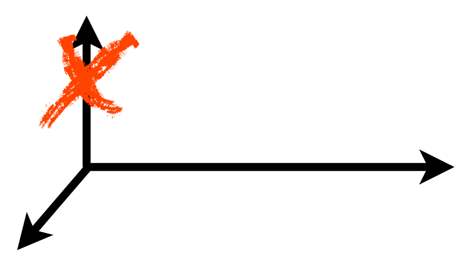


Selection problems in learning

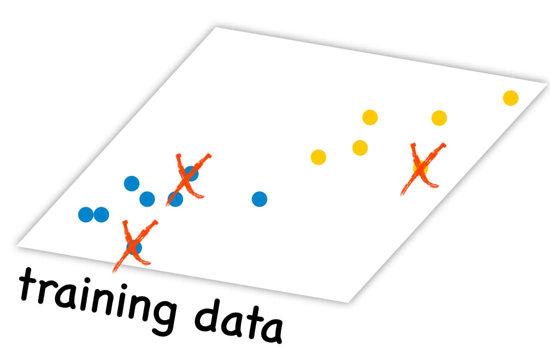
model selection



feature selection

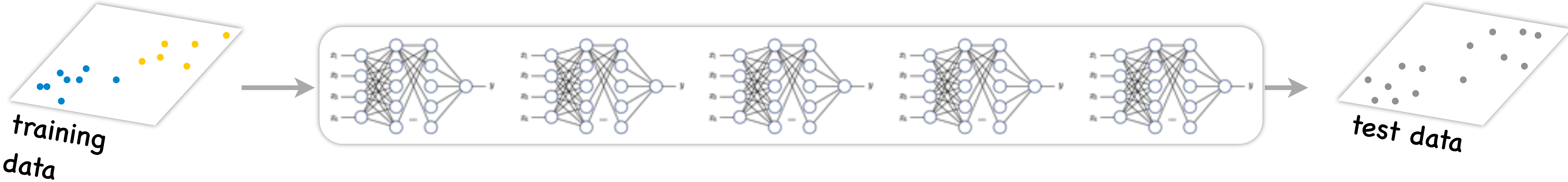


sample selection



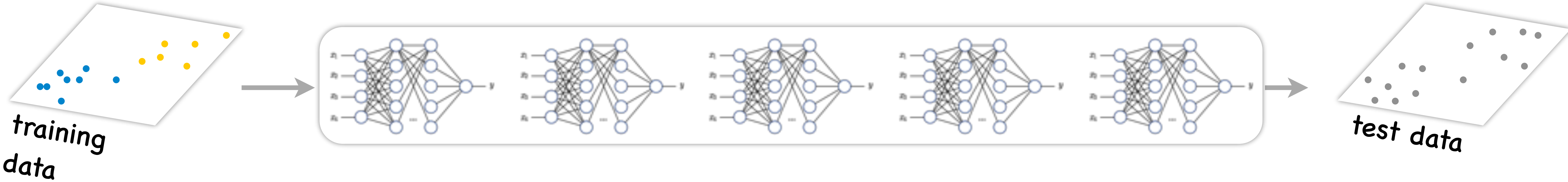
Example: selective ensemble

Ensemble: [M. P. Perrone: *Pulling it all together: Methods for combining neural networks*. **NIPS**'94]

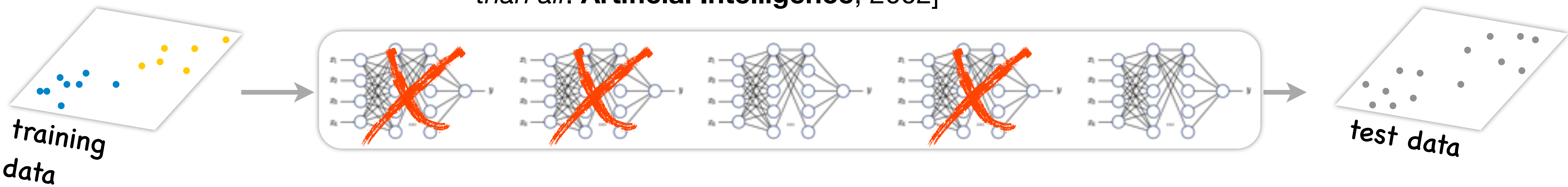


Example: selective ensemble

Ensemble: [M. P. Perrone: *Pulling it all together: Methods for combining neural networks*. **NIPS**'94]

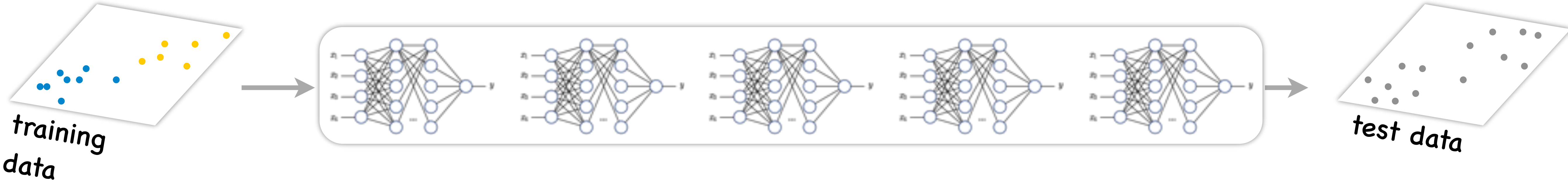


Selective ensemble: [Z.-H. Zhou, J. Wu, and W. Tang. *Ensembling neural networks: Many could be better than all*. **Artificial Intelligence**, 2002]

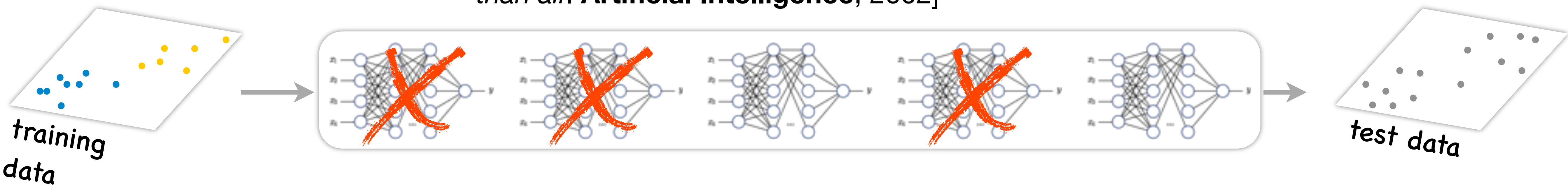


Example: selective ensemble

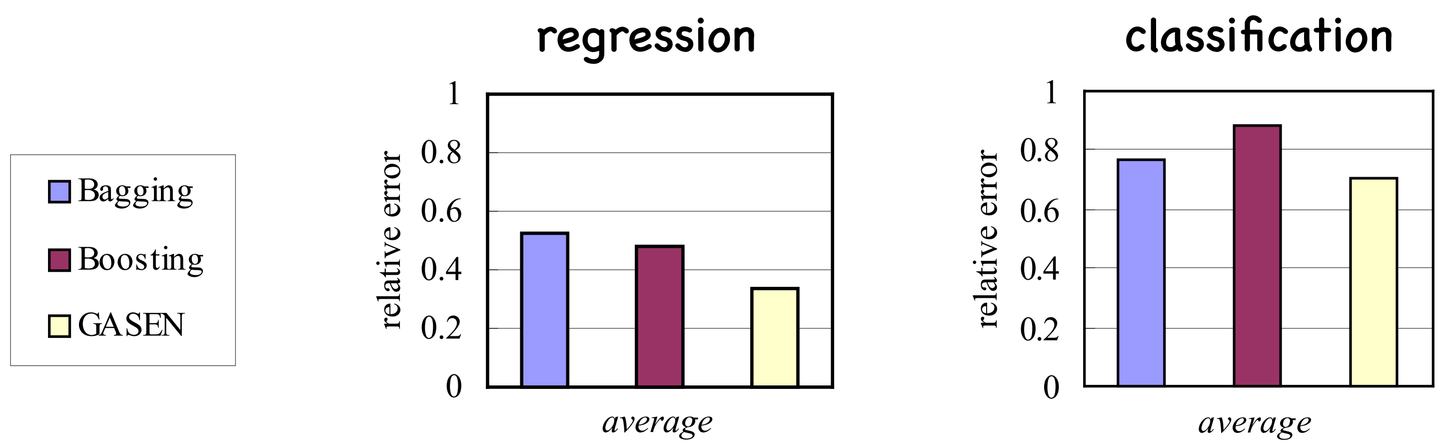
Ensemble: [M. P. Perrone: *Pulling it all together: Methods for combining neural networks*. **NIPS**'94]



Selective ensemble: [Z.-H. Zhou, J. Wu, and W. Tang. *Ensembling neural networks: Many could be better than all*. **Artificial Intelligence**, 2002]



GASEN: a **genetic algorithm** approach:



figures from [Zhou et al., AIJ'02]

two major branches: Optimization-based methods
Ordering-based methods

Subset selection problem

a set $V = \{X_1, X_2, \ldots, X_n\}$
a function $f : 2^V \to \mathbb{R}$

given a subset size restriction k
optimize the function within the subset size:

$$\arg\min_{S \subseteq V} f(S) \quad s.t. \quad |S| \leq k$$

Subset selection problem

a set $V = \{X_1, X_2, \ldots, X_n\}$
a function $f : 2^V \to \mathbb{R}$

given a subset size restriction k
optimize the function within the subset size:

$$\arg\min_{S \subseteq V} f(S) \quad s.t. \quad |S| \leq k$$

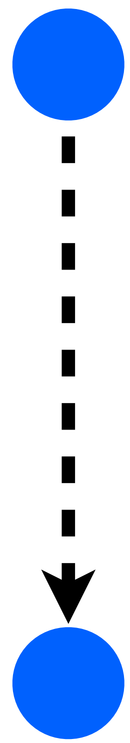
greedy algorithm

convex relaxation

heuristic search

Pareto Optimization

$$\arg\min_x f(x)$$

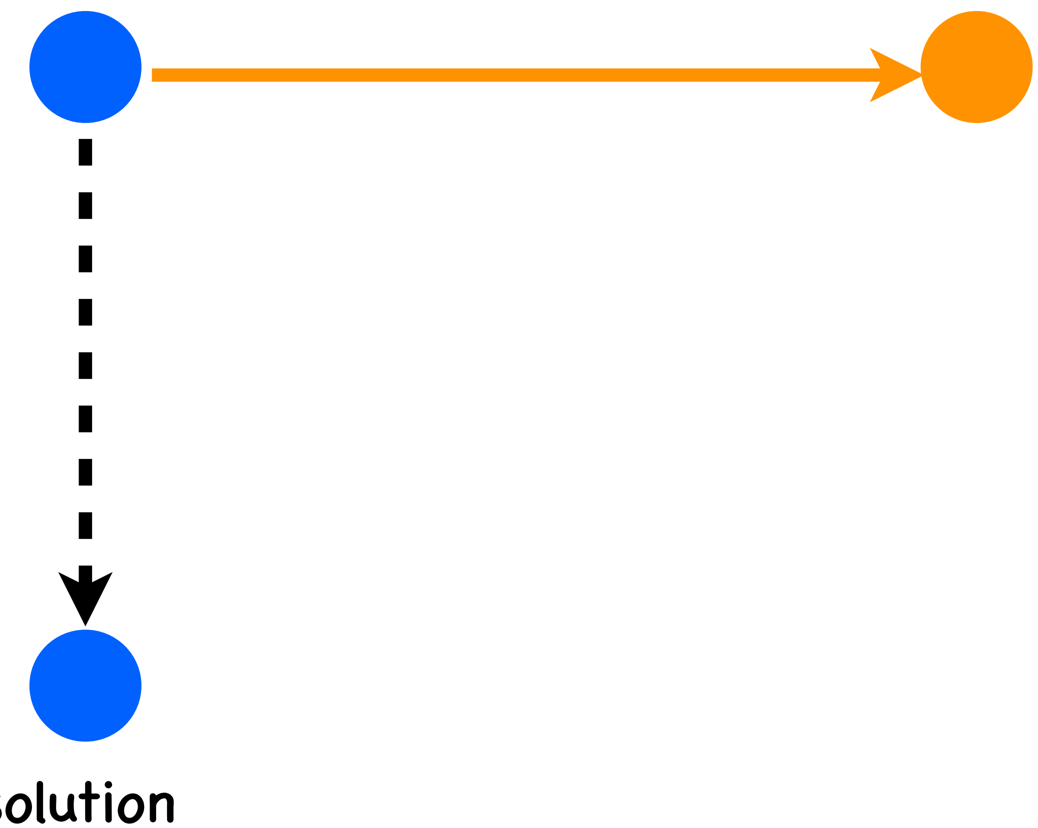


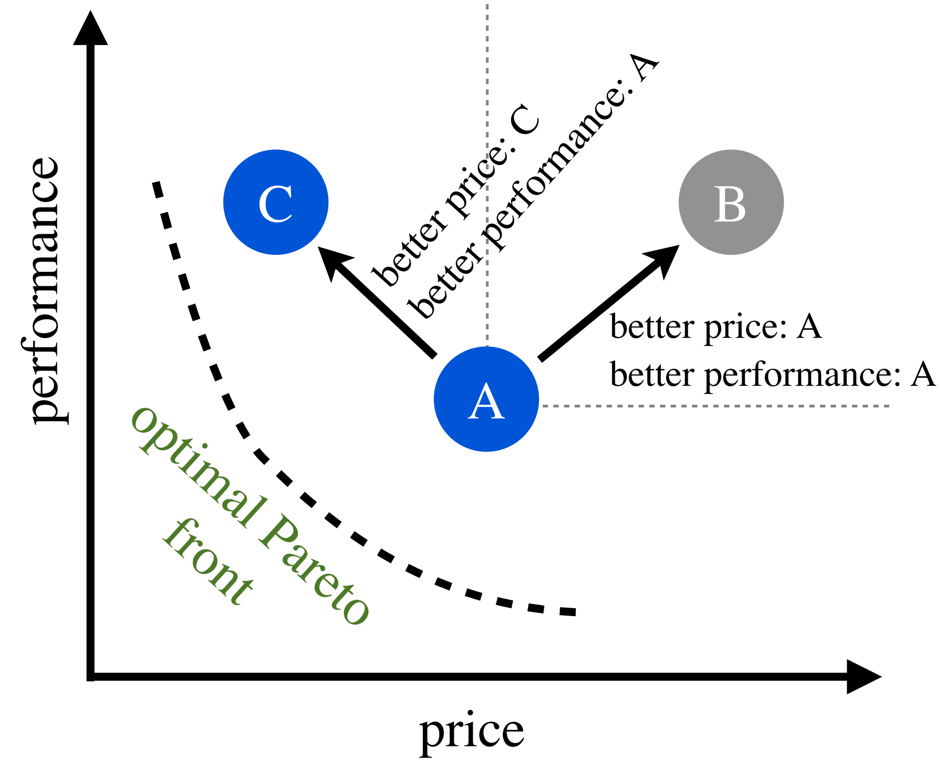
solution

Pareto Optimization

$$f = g_1 + g_2$$

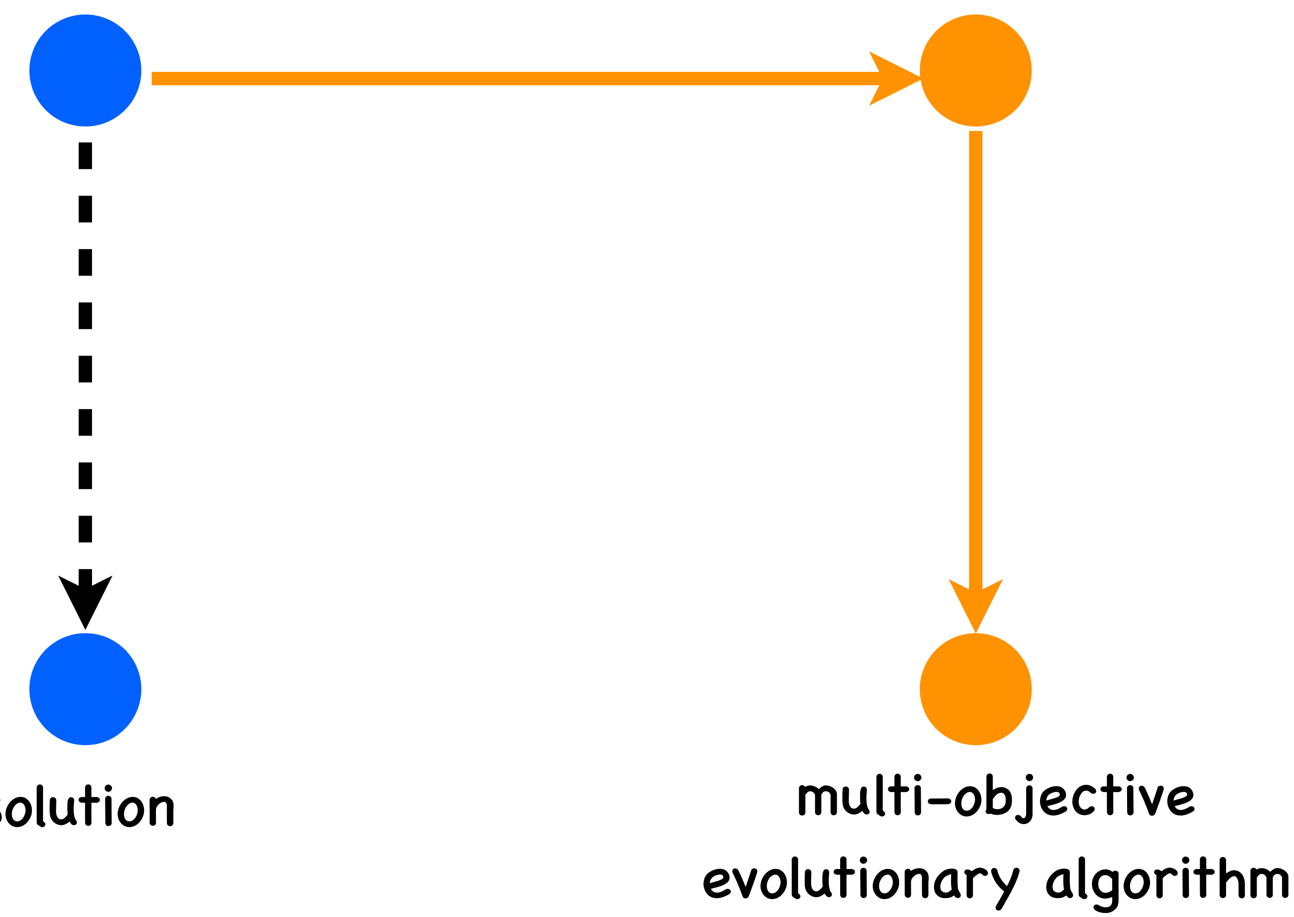
$$\arg\min_x f(x) \quad \Rightarrow \quad \arg\min_x (g_1(x), g_2(x))$$

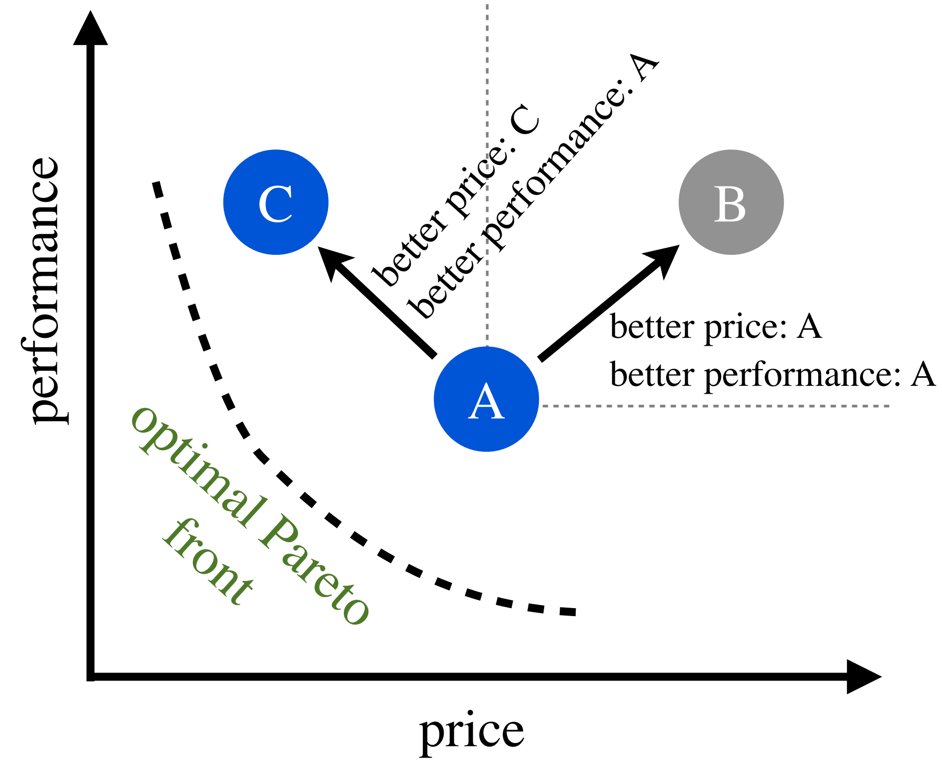


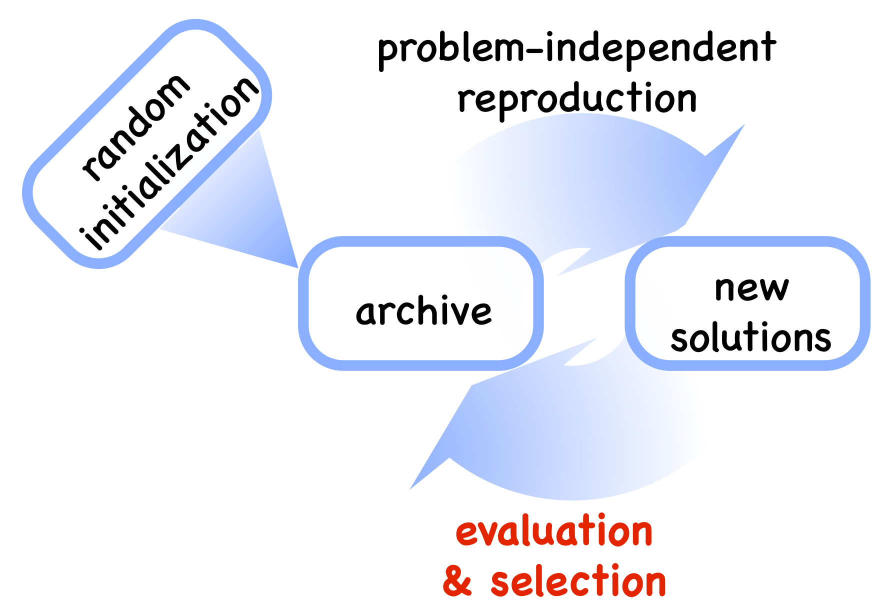


Pareto Optimization

$$\arg\min_x f(x) \overset{f = g_1 + g_2}{\Rightarrow} \arg\min_x (g_1(x), g_2(x))$$



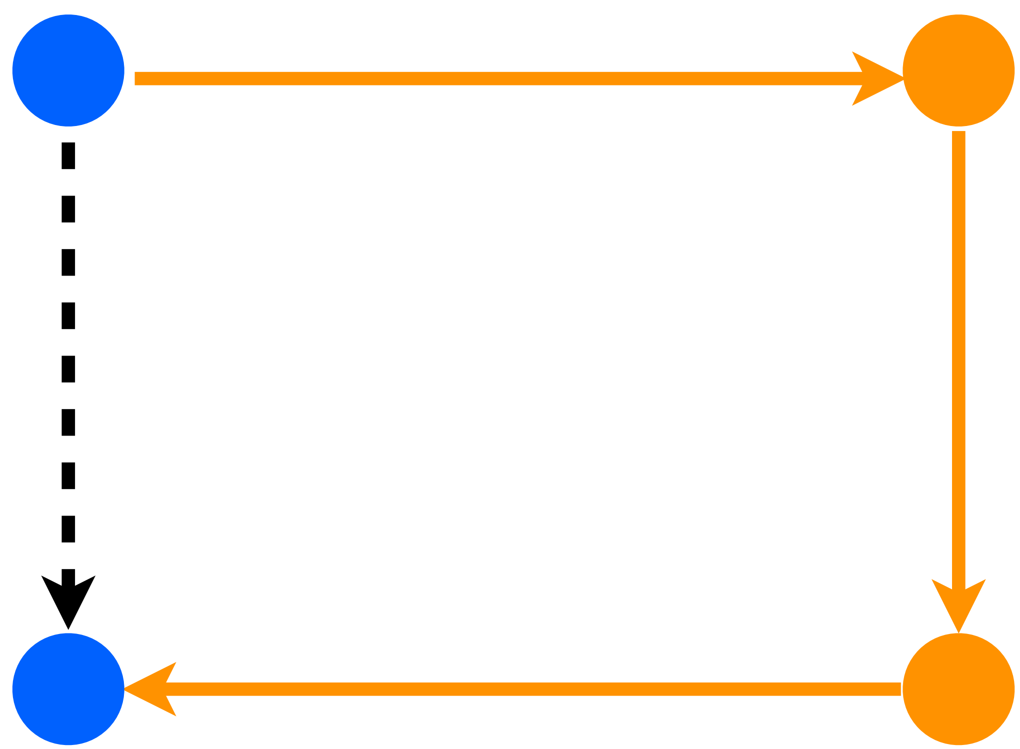




Pareto Optimization

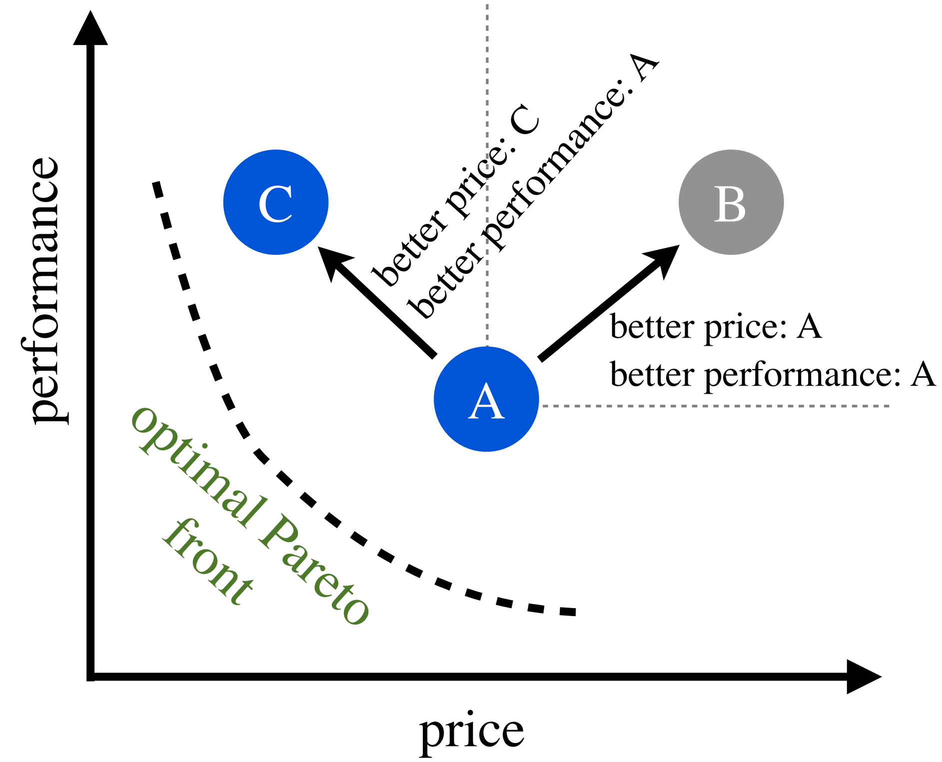
$$f = g_1 + g_2$$

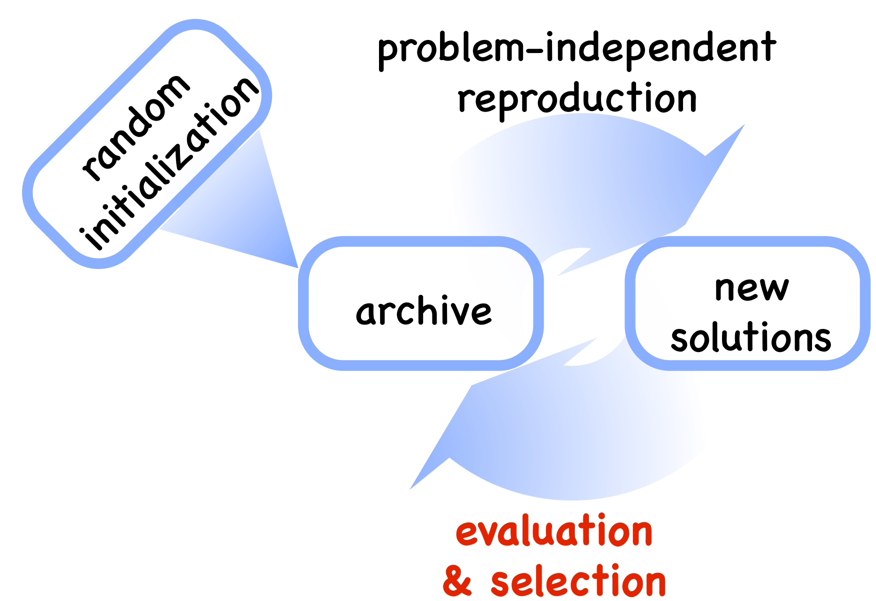
$$\arg\min_x f(x) \quad \Rightarrow \quad \arg\min_x (g_1(x), g_2(x))$$



solution

multi-objective evolutionary algorithm

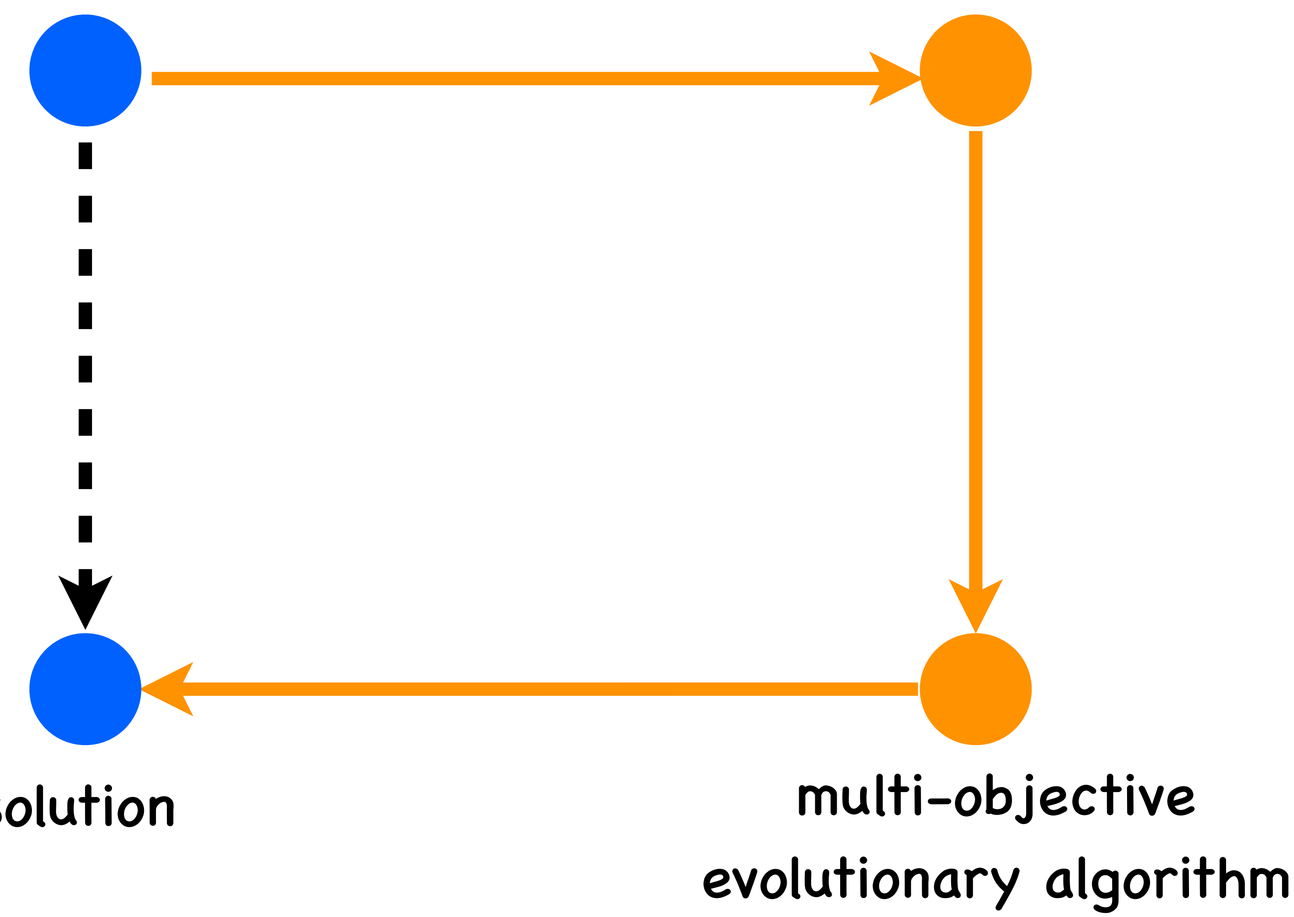


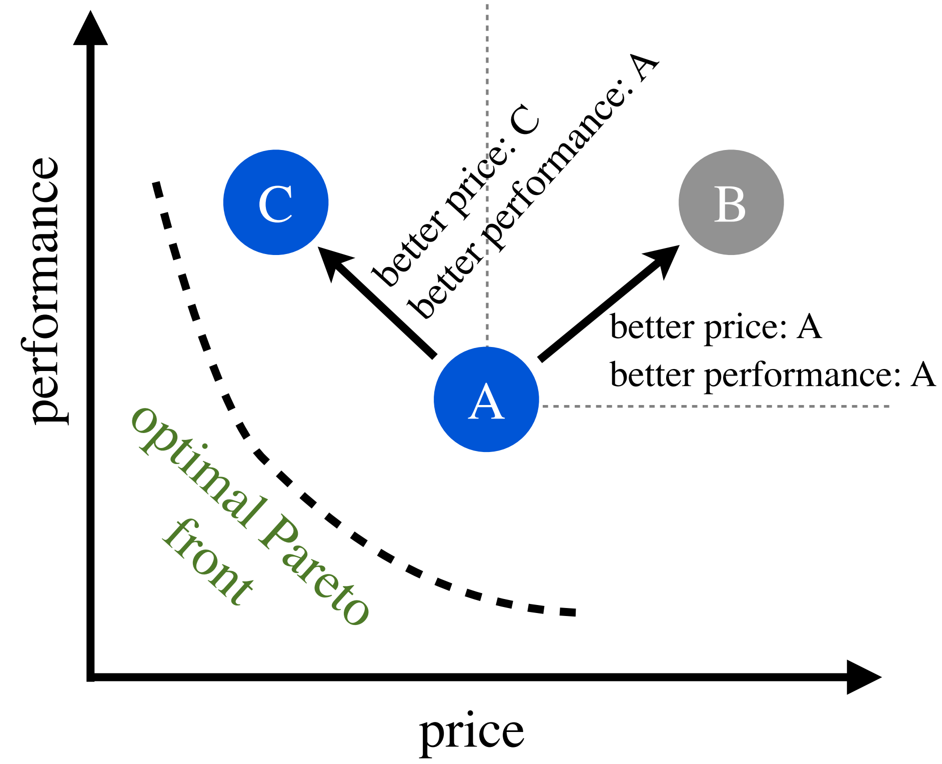


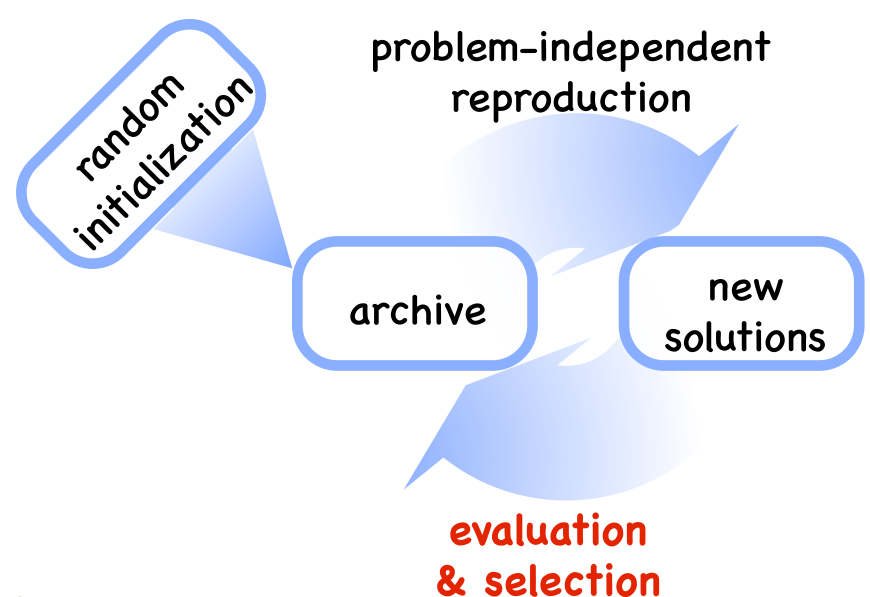
Pareto Optimization

$$f = g_1 + g_2$$

$$\arg\min_x f(x) \quad \Rightarrow \quad \arg\min_x (g_1(x), g_2(x))$$







As good as greedy algorithm on minimum vertex cover problem [Friedrich et al., ECJ'10]

SEIP framework

[Y. Yu, X. Yao, and Z.-H. Zhou. *On the approximation ability of evolutionary optimization with application to minimum set cover*. **Artificial Intelligence**, 2012.]

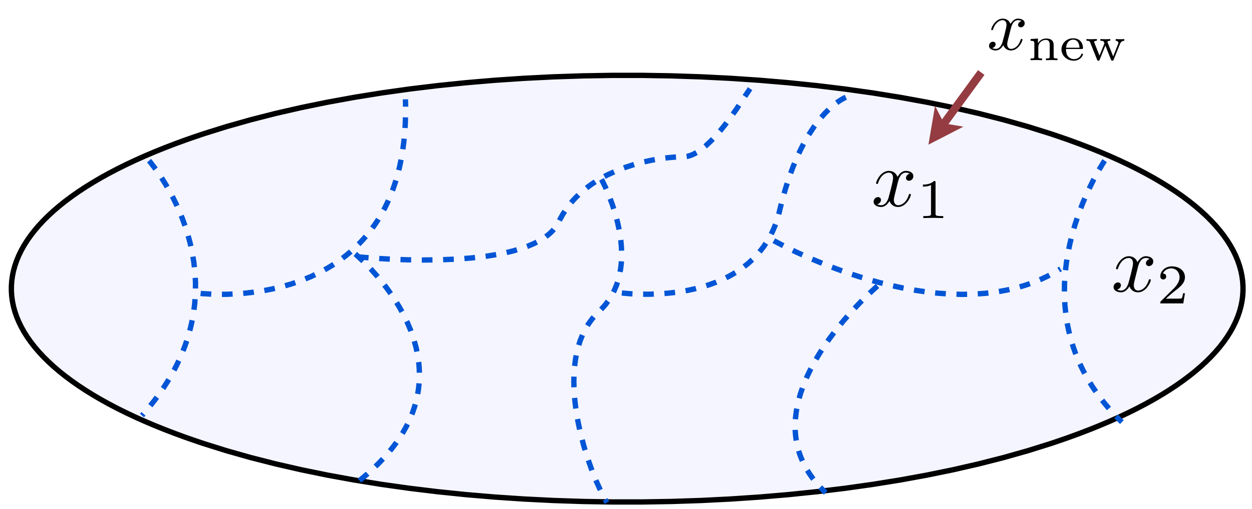
Pareto optimization is covered by the SEIP framework

SEIP framework

[Y. Yu, X. Yao, and Z.-H. Zhou. *On the approximation ability of evolutionary optimization with application to minimum set cover*. **Artificial Intelligence**, 2012.]

Pareto optimization is covered by the SEIP framework

Isolation function: isolates the competition among solutions



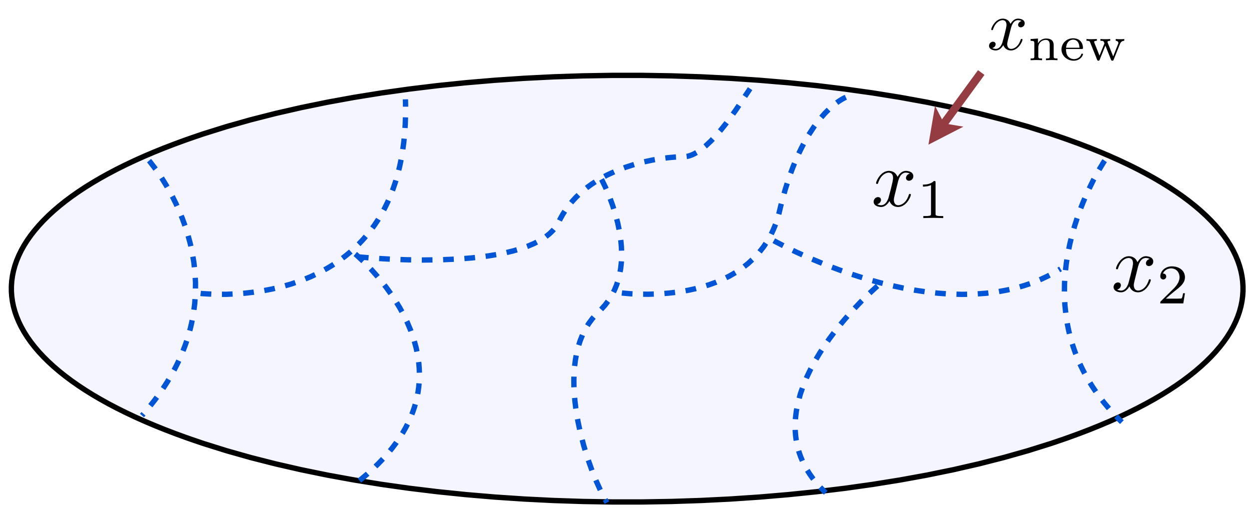
Properly configured isolation $\Rightarrow$ the multi-objective reformulation

SEIP framework

[Y. Yu, X. Yao, and Z.-H. Zhou. *On the approximation ability of evolutionary optimization with application to minimum set cover*. **Artificial Intelligence**, 2012.]

Pareto optimization is covered by the SEIP framework

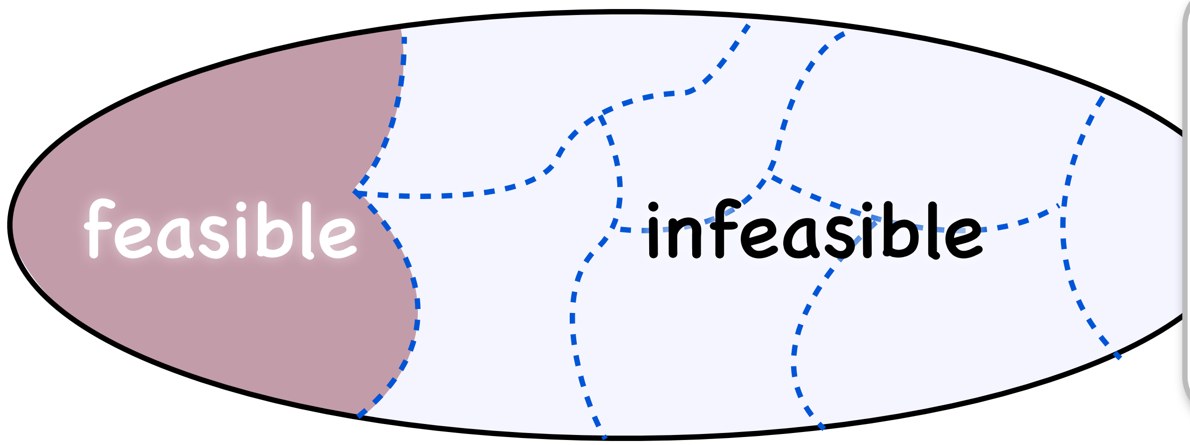
Isolation function: isolates the competition among solutions



Properly configured isolation $\Rightarrow$ the multi-objective reformulation

Partial ratio:

measures infeasible solutions



Definition 4 (Partial reference function)

Given a set $[q]$ *and a value* v*, a function* $\mathcal{L}_{[q],v} : 2^{[q]} \to \mathbb{R}$ *is a partial reference function if*

1) $\mathcal{L}_{[q],v}([q]) = v$,

2) $\mathcal{L}_{[q],v}(R_1) = \mathcal{L}_{[q],v}(R_2)$ *for all* $R_1, R_2 \subseteq [q]$ *such that* $|R_1| = |R_2|$.

Definition 5 (Partial ratio)

Given a minimization problem $(n, f, \mathcal{C})$ *and an isolation function* μ*, the partial ratio of a (partial) solution* $\boldsymbol{x}$ *with respect to a corresponding partial reference function* $\mathcal{L}$ *is*

$$\text{p-ratio}(\boldsymbol{x}) = \frac{f(\boldsymbol{x})}{\mathcal{L}(\mu(\boldsymbol{x}))},$$

and the conditional partial ratio of $\boldsymbol{y}$ *conditioned on* $\boldsymbol{x}$ *is*

$$\text{p-ratio}(\boldsymbol{x} \mid \boldsymbol{y}) = \frac{f(\boldsymbol{y} \mid \boldsymbol{x})}{\mathcal{L}(\mu(\boldsymbol{y}) \mid \mu(\boldsymbol{x}))},$$

where $f(\boldsymbol{y} \mid \boldsymbol{x}) = f(\boldsymbol{x} \cup \boldsymbol{y}) - f(\boldsymbol{x})$ *and* $\mathcal{L}(\mu(\boldsymbol{y}) \mid \mu(\boldsymbol{x})) = \mathcal{L}(\mu(\boldsymbol{y}) \cup \mu(\boldsymbol{x})) - \mathcal{L}(\mu(\boldsymbol{x}))$.

SEIP framework

[Y. Yu, X. Yao, and Z.-H. Zhou. *On the approximation ability of evolutionary optimization with application to minimum set cover*. **Artificial Intelligence**, 2012.]

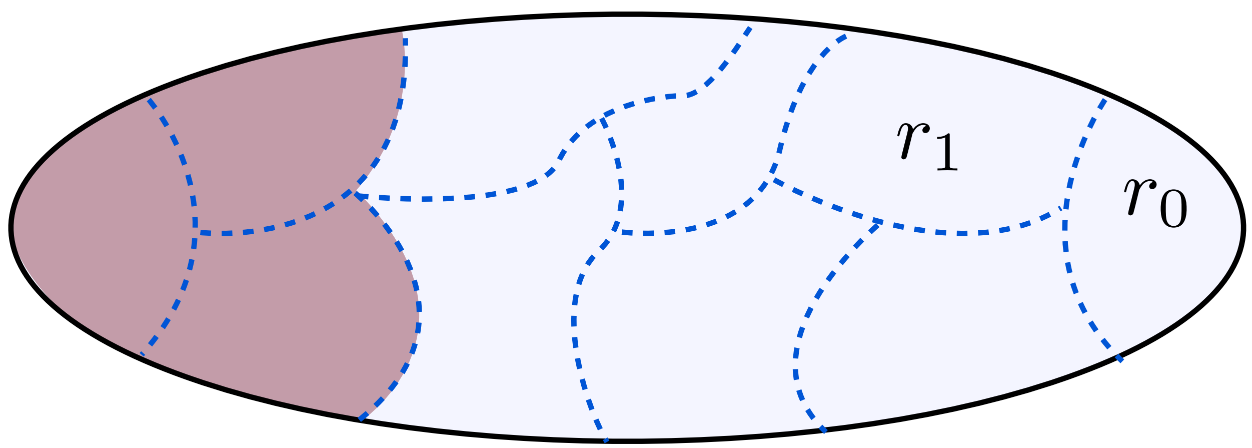
Theorem

SEIP can find $(\sum_{i=0}^{q-1} r_i)$-approximate solutions in $O(q^2 n^c)$ time

best conditional partial ratio in isolations

number of isolations

size of an isolation



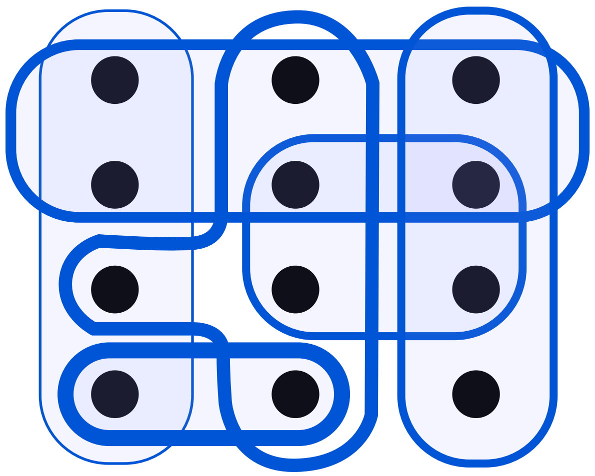
Pareto optimization $\Rightarrow$ balance q and c

Example

[Y. Yu, X. Yao, and Z.-H. Zhou. *On the approximation ability of evolutionary optimization with application to minimum set cover*. **Artificial Intelligence**, 2012.]

On minimum set cover problem

a typical NP-hard problem for approximation studies



n **elements in** E

m **weighted sets in** C

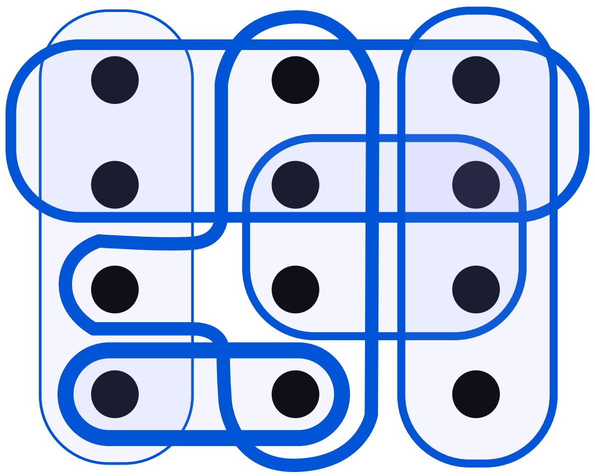
k is the size of the largest set

Example

[Y. Yu, X. Yao, and Z.-H. Zhou. *On the approximation ability of evolutionary optimization with application to minimum set cover*. **Artificial Intelligence**, 2012.]

On minimum set cover problem

a typical NP-hard problem for approximation studies



n elements in E

m weighted sets in C

k is the size of the largest set

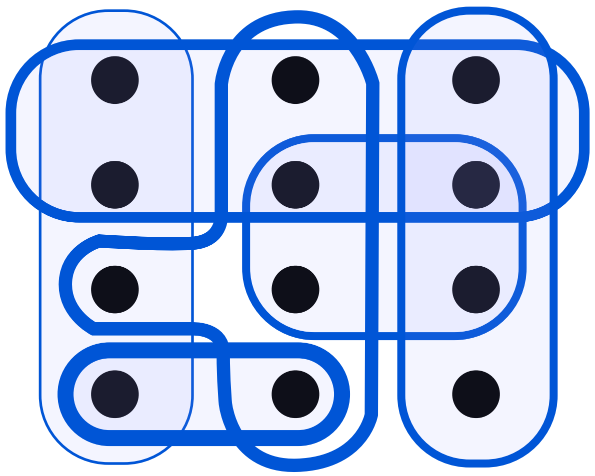
For unbounded minimum set cover problem:

SEIP finds H_n-approximate solutions in $O(mn^2)$ time

Example

[Y. Yu, X. Yao, and Z.-H. Zhou. *On the approximation ability of evolutionary optimization with application to minimum set cover*. **Artificial Intelligence**, 2012.]

On minimum set cover problem a typical NP-hard problem for approximation studies



n elements in E

m weighted sets in C

k is the size of the largest set

For unbounded minimum set cover problem:

SEIP finds H_n-approximate solutions in $O(mn^2)$ time

For minimum k-set cover problem:

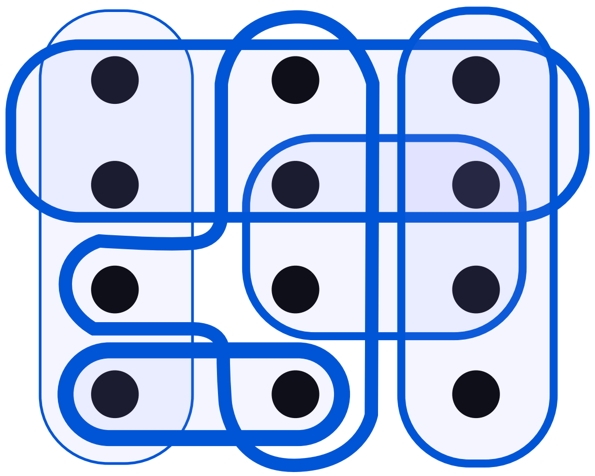
SEIP finds $(H_k - \frac{k-1}{8k^9})$-approximate solutions in $O(m^{k+1}n^2)$ time

Example

[Y. Yu, X. Yao, and Z.-H. Zhou. *On the approximation ability of evolutionary optimization with application to minimum set cover*. **Artificial Intelligence**, 2012.]

On minimum set cover problem

a typical NP-hard problem for approximation studies



n elements in E

m weighted sets in C

k is the size of the largest set

For unbounded minimum set cover problem:

SEIP finds H_n-approximate solutions in $O(mn^2)$ time

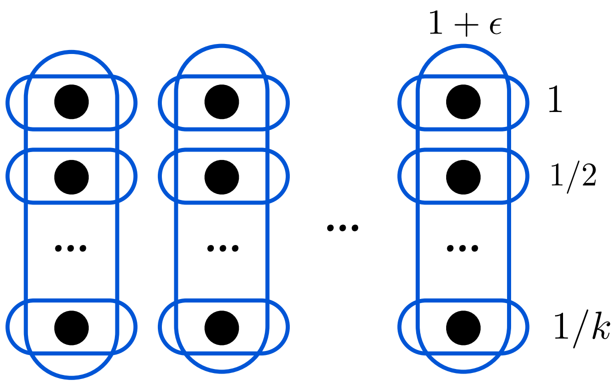
For minimum k-set cover problem:

SEIP finds $(H_k - \frac{k-1}{8k^9})$-approximate solutions in $O(m^{k+1}n^2)$ time

Pareto optimization can be the best-so-far approximation algorithm

Example

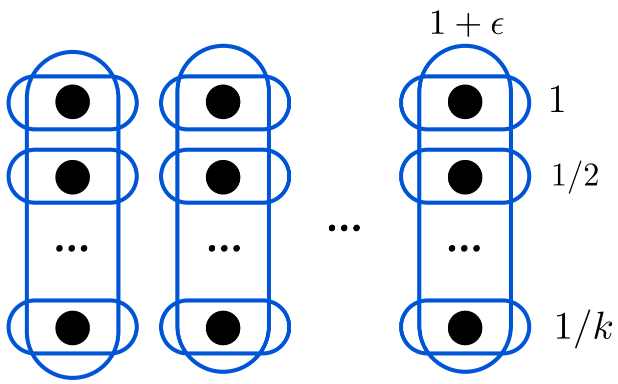
[Y. Yu, X. Yao, and Z.-H. Zhou. *On the approximation ability of evolutionary optimization with application to minimum set cover*. **Artificial Intelligence**, 2012.]



Greedy algorithm: bad! no better than H_k

Example

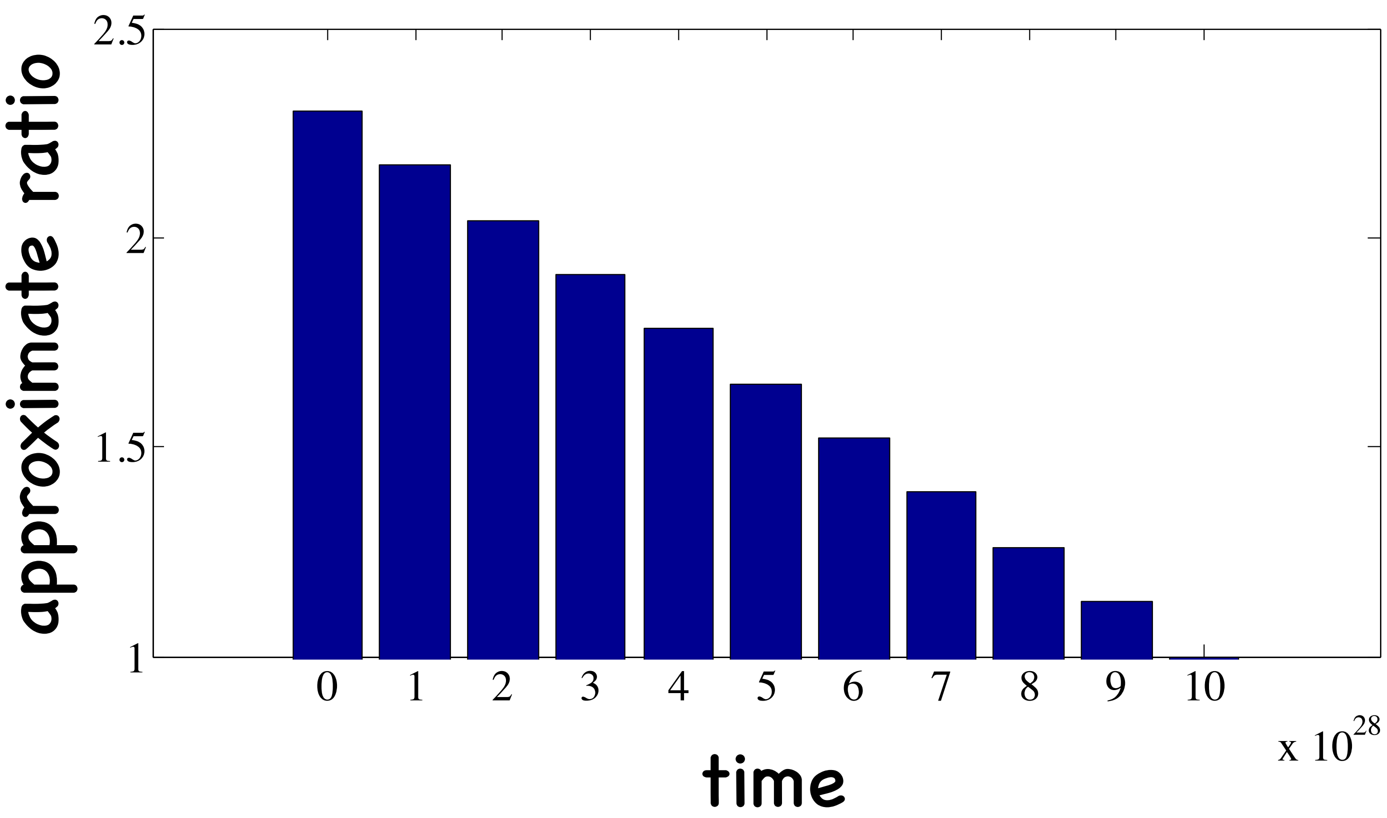
[Y. Yu, X. Yao, and Z.-H. Zhou. *On the approximation ability of evolutionary optimization with application to minimum set cover*. **Artificial Intelligence**, 2012.]



Greedy algorithm: bad! no better than H_k

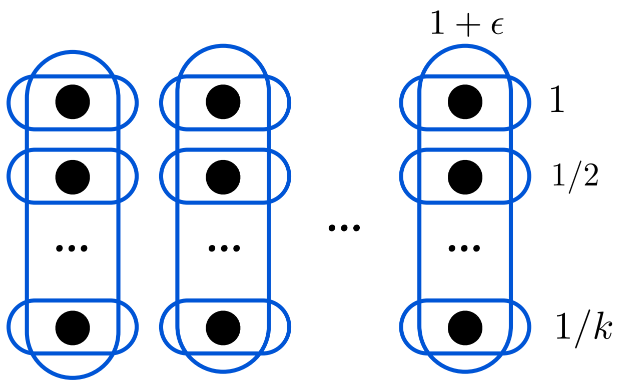
SEIP: for $c = 0, 1, \ldots, \frac{n}{k}$, $(H_k - c\frac{k}{n}(H_k - 1))$-approximate solutions in $O(mn^2 + cn^2m^{k+1}/k)$ time

(anytime algorithm)



Example

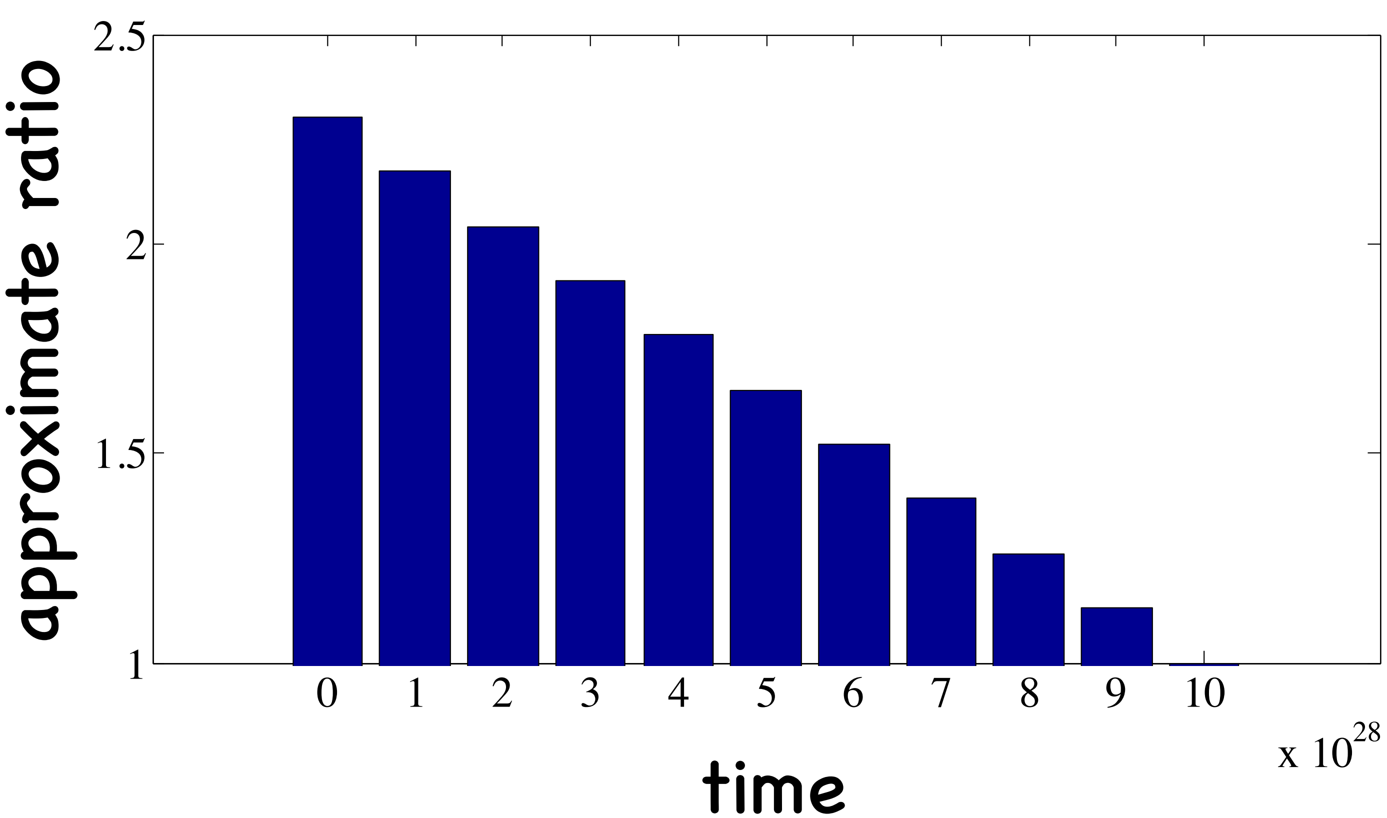
[Y. Yu, X. Yao, and Z.-H. Zhou. *On the approximation ability of evolutionary optimization with application to minimum set cover*. **Artificial Intelligence**, 2012.]



Greedy algorithm: bad! no better than H_k

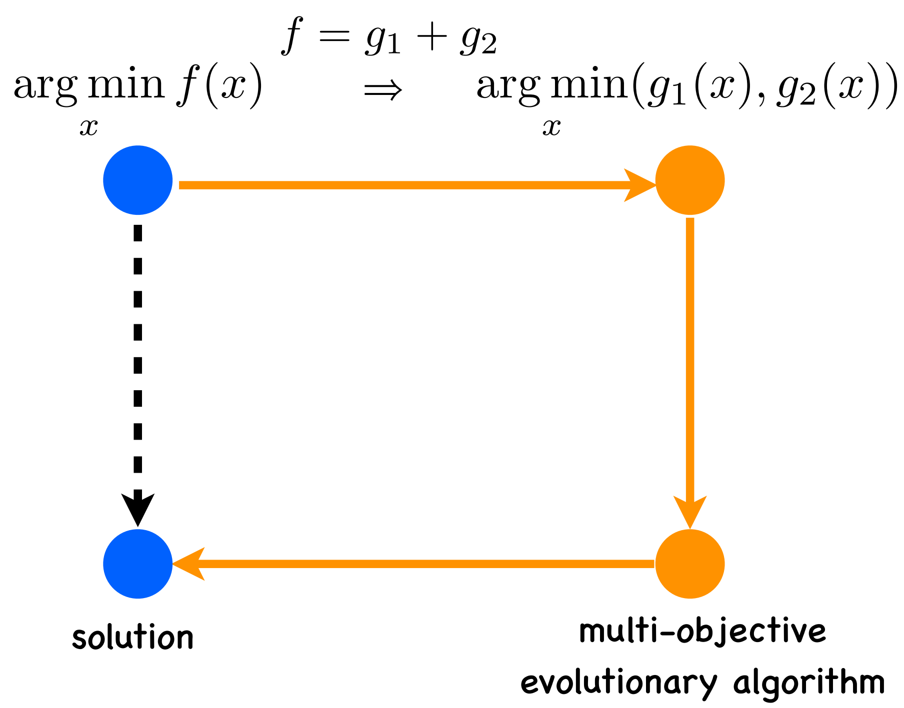
SEIP: for $c = 0, 1, \dots, \frac{n}{k}$, $(H_k - c\frac{k}{n}(H_k - 1))$-approximate solutions in $O(mn^2 + cn^2m^{k+1}/k)$ time

(anytime algorithm)



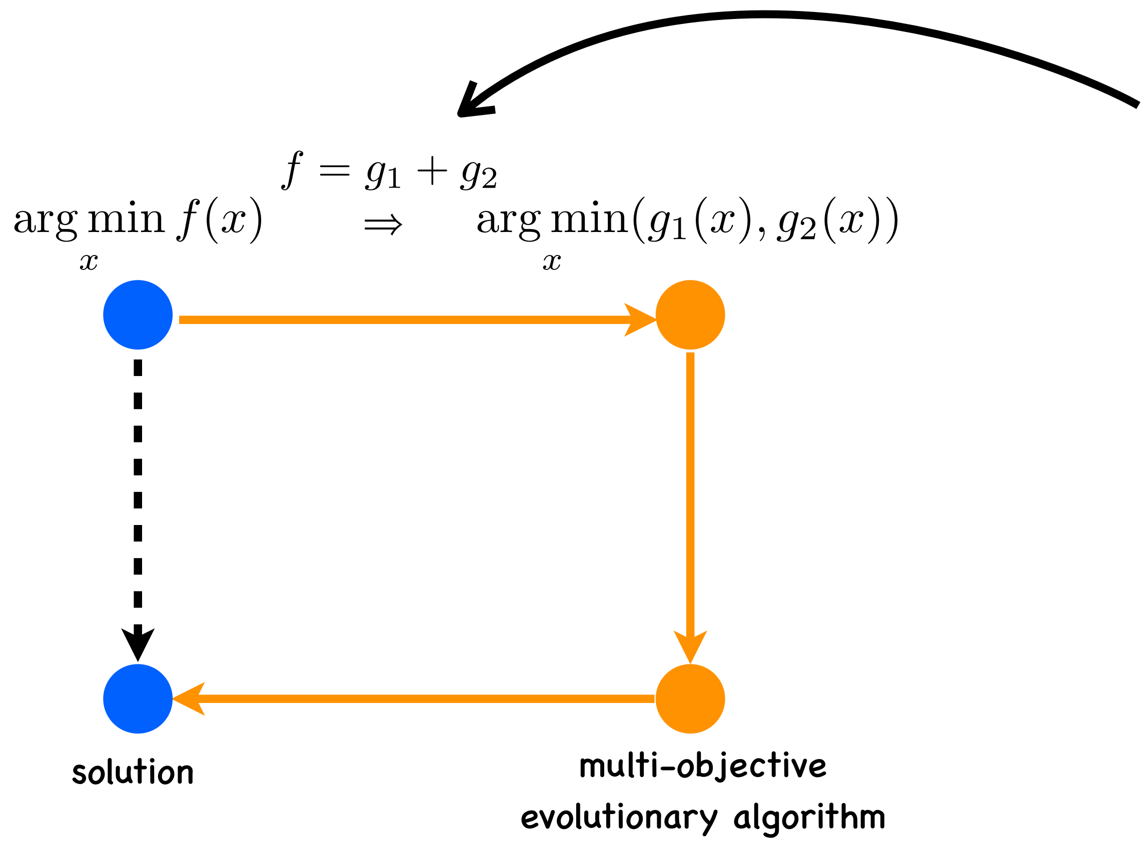
Pareto optimization can be the best-so-far approximation algorithm, with practical advantages

A question before using



how to decompose f ?

A question before using



how to decompose f ?

Subset selection is a constrained problem.

$$\arg\min_{S \subseteq V} f(S) \quad s.t. \quad |S| \leq k$$

For constrained optimization problems

For constrained optimization problems

Constrained optimization: $\arg\min_{\boldsymbol{x}\in\{0,1\}^n} \quad f(\boldsymbol{x})$

$$\begin{aligned} \textit{subject to} \quad & g_i(\boldsymbol{x}) = 0 \quad \textit{for } 1 \leq i \leq q, \\ & h_i(\boldsymbol{x}) \leq 0 \quad \textit{for } q+1 \leq i \leq m, \end{aligned}$$

For constrained optimization problems

Constrained optimization:

$$\begin{aligned} &\arg\min_{\boldsymbol{x}\in\{0,1\}^n} && f(\boldsymbol{x}) \\ &\textit{subject to} && g_i(\boldsymbol{x}) = 0 \quad \textit{for } 1 \le i \le q, \\ & && h_i(\boldsymbol{x}) \le 0 \quad \textit{for } q+1 \le i \le m, \end{aligned}$$

Penalty Function method:

$$\arg\min_{\boldsymbol{x}\in\{0,1\}^n} f(\boldsymbol{x}) + \lambda \sum_{i=1}^{m} f_i(\boldsymbol{x})$$

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

For constrained optimization problems

Constrained optimization:

$$\arg\min_{\boldsymbol{x}\in\{0,1\}^n} \quad f(\boldsymbol{x})$$
$$\textit{subject to} \quad g_i(\boldsymbol{x}) = 0 \quad \textit{for } 1 \leq i \leq q,$$
$$h_i(\boldsymbol{x}) \leq 0 \quad \textit{for } q+1 \leq i \leq m,$$

Penalty Function method:

$$\arg\min_{\boldsymbol{x}\in\{0,1\}^n} f(\boldsymbol{x}) + \lambda \sum_{i=1}^{m} f_i(\boldsymbol{x})$$

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

Pareto Optimization method :

$$\arg\min_x f(x) \quad \xRightarrow{f = g_1 + g_2} \quad \arg\min_x (g_1(x), g_2(x))$$

$$\arg\min_{\boldsymbol{x}\in\{0,1\}^n} \left(f(\boldsymbol{x}), \sum_{i=1}^{m} f_i(\boldsymbol{x})\right)$$

For constrained optimization problems

Constrained optimization: $\arg\min_{\boldsymbol{x}\in\{0,1\}^n} \quad f(\boldsymbol{x})$

$$\textit{subject to} \quad g_i(\boldsymbol{x}) = 0 \quad \textit{for } 1 \le i \le q,$$
$$h_i(\boldsymbol{x}) \le 0 \quad \textit{for } q+1 \le i \le m,$$

Penalty Function method:

$$\arg\min_{\boldsymbol{x}\in\{0,1\}^n} f(\boldsymbol{x}) + \lambda \sum_{i=1}^{m} f_i(\boldsymbol{x})$$

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

Pareto Optimization method :

$$\arg\min_x f(x) \quad \overset{f = g_1 + g_2}{\Rightarrow} \quad \arg\min_x (g_1(x), g_2(x))$$

$$\arg\min_{\boldsymbol{x}\in\{0,1\}^n} \left(f(\boldsymbol{x}), \sum_{i=1}^{m} f_i(\boldsymbol{x})\right)$$

can Pareto optimization be better?

Problem Class 1

[C. Qian, Y. Yu and Z.-H. Zhou. *On Constrained Boolean Pareto Optimization*. **IJCAI'15**]

Minimum Matroid Problem

matroid

Let $|\cdot|$ denote the size (i.e., cardinality) of a set. A matroid is a pair (U, S), where U is a finite set and $S \subseteq 2^U$, satisfying

(1) $\emptyset \in S$; (2) $\forall A \subseteq B \in S, A \in S$;

(3) $\forall A, B \in S, |A| > |B| : \exists e \in A - B, B \cup \{e\} \in S$.

rank: $r(A) = \max\{|B| \mid B \subseteq A, B \in S\}$

Problem Class 1

[C. Qian, Y. Yu and Z.-H. Zhou. *On Constrained Boolean Pareto Optimization*. **IJCAI'15**]

Minimum Matroid Problem

matroid

Let $|\cdot|$ denote the size (i.e., cardinality) of a set. A matroid is a pair (U, S), where U is a finite set and $S \subseteq 2^U$, satisfying

(1) $\emptyset \in S$; (2) $\forall A \subseteq B \in S, A \in S$;

(3) $\forall A, B \in S, |A| > |B| : \exists e \in A - B, B \cup \{e\} \in S$.

rank: $r(A) = \max\{|B| \mid B \subseteq A, B \in S\}$

minimum matroid optimization

given a matroid (U,S), let x be the subset indicator vector of U

$$\arg\min_{\boldsymbol{x} \in \{0,1\}^n} w(\boldsymbol{x}) = \sum\nolimits_{i=1}^{n} w_i x_i \quad s.t. \quad r(\boldsymbol{x}) = r(U)$$

e.g. minimum spanning tree, maximum bipartite matching

Problem Class 1

[C. Qian, Y. Yu and Z.-H. Zhou. *On Constrained Boolean Pareto Optimization*. **IJCAI'15**]

Minimum Matroid Problem

- the worst problem-case average-runtime complexity
- solve optimal solutions

Problem Class 1

[C. Qian, Y. Yu and Z.-H. Zhou. *On Constrained Boolean Pareto Optimization*. **IJCAI'15**]

Minimum Matroid Problem

- the worst problem-case average-runtime complexity
- solve optimal solutions

For the Penalty Function Method

$$\Omega(r^2 n(\log n + \log w_{\max}))$$

Problem Class 1

[C. Qian, Y. Yu and Z.-H. Zhou. *On Constrained Boolean Pareto Optimization*. **IJCAI'15**]

Minimum Matroid Problem

- the worst problem-case average-runtime complexity
- solve optimal solutions

For the Penalty Function Method

$$\Omega(r^2 n(\log n + \log w_{\max}))$$

For the Pareto Optimization Method

$$O(rn(r + \log n + \log w_{\max}))$$

Problem Class 2

[C. Qian, Y. Yu and Z.-H. Zhou. *On Constrained Boolean Pareto Optimization*. **IJCAI'15**]

Minimum Cost Coverage

Monotonic submodular function

Let $U = \{e_1, e_2, \ldots, e_n\}$ be a finite set. A set function $f : 2^U \to \mathbb{R}$ is monotone and submodular iff $\forall A, B \subseteq U$, $f(A) \leq f(B) + \sum_{e \in A-B} (f(B \cup \{e\}) - f(B))$

Problem Class 2

[C. Qian, Y. Yu and Z.-H. Zhou. *On Constrained Boolean Pareto Optimization*. **IJCAI'15**]

Minimum Cost Coverage

Monotonic submodular function

Let $U = \{e_1, e_2, \ldots, e_n\}$ be a finite set. A set function $f : 2^U \to \mathbb{R}$ is monotone and submodular iff $\forall A, B \subseteq U$, $f(A) \leq f(B) + \sum_{e \in A-B}(f(B \cup \{e\}) - f(B))$

minimum cost coverage problem

given U, let **x** be the subset indicator vector of U, given a monotone and submodular function f , and some value $q \leq f(U)$

$$\mathop{\arg\min}_{\boldsymbol{x} \in \{0,1\}^n} \ w(\boldsymbol{x}) = \sum\nolimits_{i=1}^{n} w_i x_i \quad s.t. \quad f(\boldsymbol{x}) \geq q$$

e.g. minimum submodular cover, minimum set cover

Problem Class 2

[C. Qian, Y. Yu and Z.-H. Zhou. *On Constrained Boolean Pareto Optimization*. **IJCAI'15**]

Minimum Matroid Problem

- the worst problem-case average-runtime complexity
- solve H_q-approximate solutions

Problem Class 2

[C. Qian, Y. Yu and Z.-H. Zhou. *On Constrained Boolean Pareto Optimization*. **IJCAI'15**]

Minimum Matroid Problem

- the worst problem-case average-runtime complexity
- solve H_q-approximate solutions

For the Penalty Function Method

$$\text{at least exponential w.r.t. } n,\ q \text{ and } \log w_{\max}$$

Problem Class 2

[C. Qian, Y. Yu and Z.-H. Zhou. *On Constrained Boolean Pareto Optimization*. **IJCAI'15**]

Minimum Matroid Problem

- the worst problem-case average-runtime complexity
- solve H_q-approximate solutions

For the Penalty Function Method

$$\text{at least exponential w.r.t. } n,\ q \text{ and } \log w_{\max}$$

For the Pareto Optimization Method

$$O(qn(\log n + \log w_{\max} + q))$$

Problem Class 2

[C. Qian, Y. Yu and Z.-H. Zhou. *On Constrained Boolean Pareto Optimization*. **IJCAI'15**]

Minimum Matroid Problem

- the worst problem-case average-runtime complexity
- solve H_q-approximate solutions

For the Penalty Function Method

$$\text{at least exponential w.r.t. } n,\ q \text{ and } \log w_{\max}$$

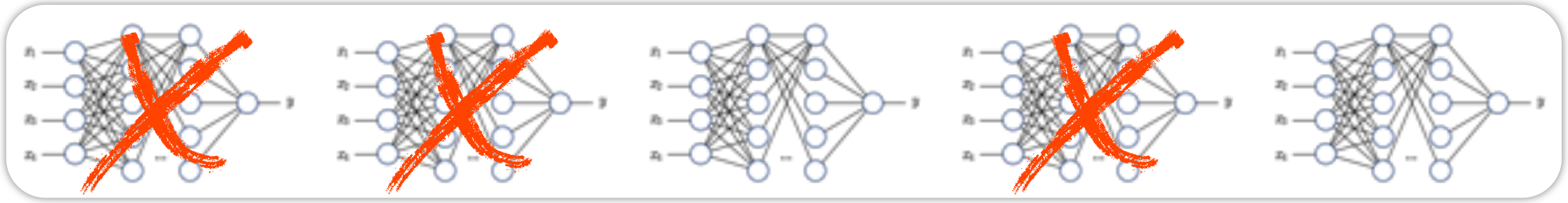
For the Pareto Optimization Method

$$O(qn(\log n + \log w_{\max} + q))$$

Pareto optimization can be much better than penalty method

App. 1: selective ensemble

Selective ensemble:



Previous approaches

Ordering-based methods

error minimization [Margineantu and Dietterich, ICML'97]

OEP diversity-like criterion maximization [Banfield et al., Info Fusion'05] [Martínez-Munõz, Hernańdez-Lobato, and Suaŕez TPAMI'09]

combined criterion [Li, Yu, and Zhou, ECML'12]

Optimization-based methods

semi-definite programming [Zhang, Burer and Street, JMLR'06]

quadratic programming [Li and Zhou, MCS'09]

genetic algorithms [Zhou, Wu and Tang, AIJ'02]

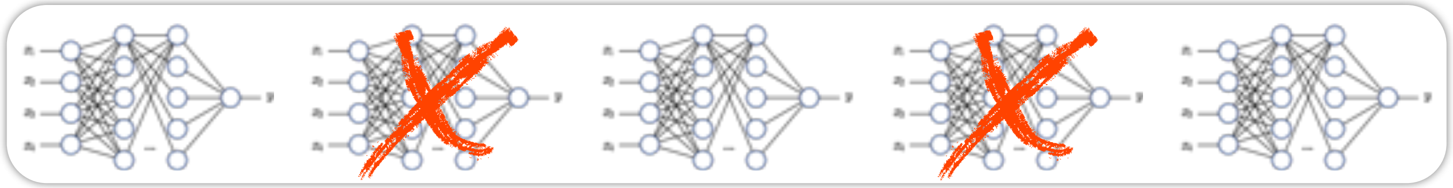
SEP

artificial immune algorithms [Castro et al., ICARIS'05]

App. 1: selective ensemble

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Ensemble Pruning*. **AAAI'15**]

by Pareto optimization method:



$$\arg\min_{\text{model subset}} \ \ell(\text{model subset}) \quad s.t. \quad \text{subset size} \leq k$$

$\downarrow$ reduce error

$\downarrow$ reduce size

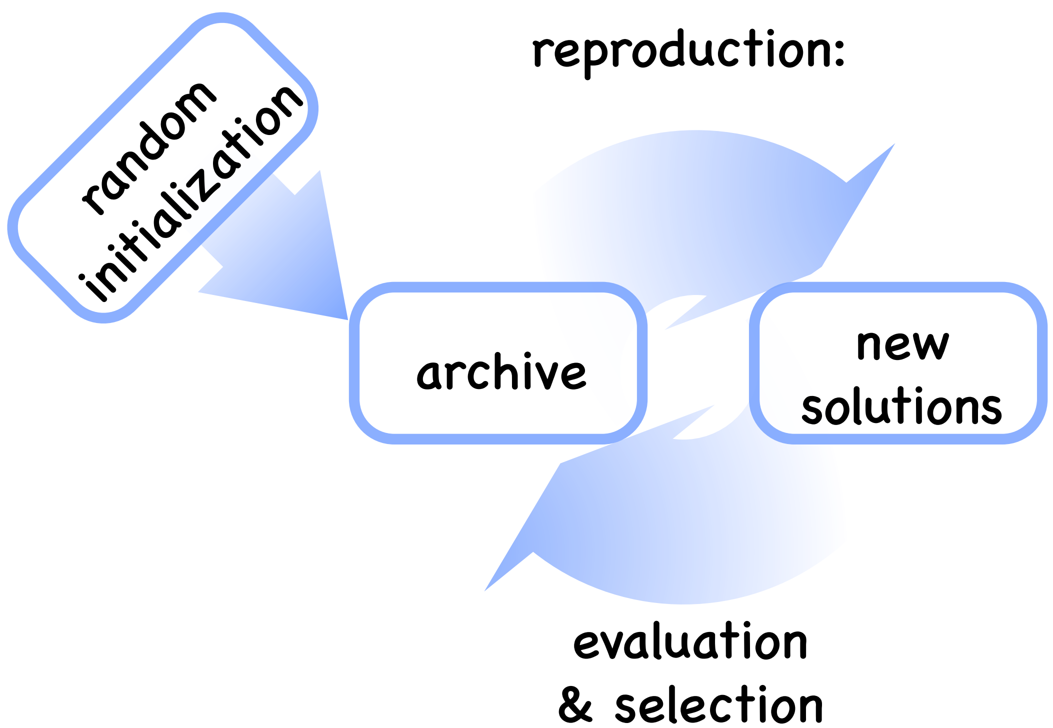
selective ensemble can be divided into two goals

App. 1: selective ensemble

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Ensemble Pruning*. **AAAI'15**]

Pareto Ensemble Pruning (PEP):

1. random generate a pruned ensemble, put it into the archive
2. loop
| 2.1 pick an ensemble randomly from the archive
| 2.2 randomly change it to make a new one
| 2.3 if the new one is not dominated
| | 2.3.1 put it into the archive
| | 2.3.2 put its good neighbors into the archive
3. when terminates, select an ensemble from the archive

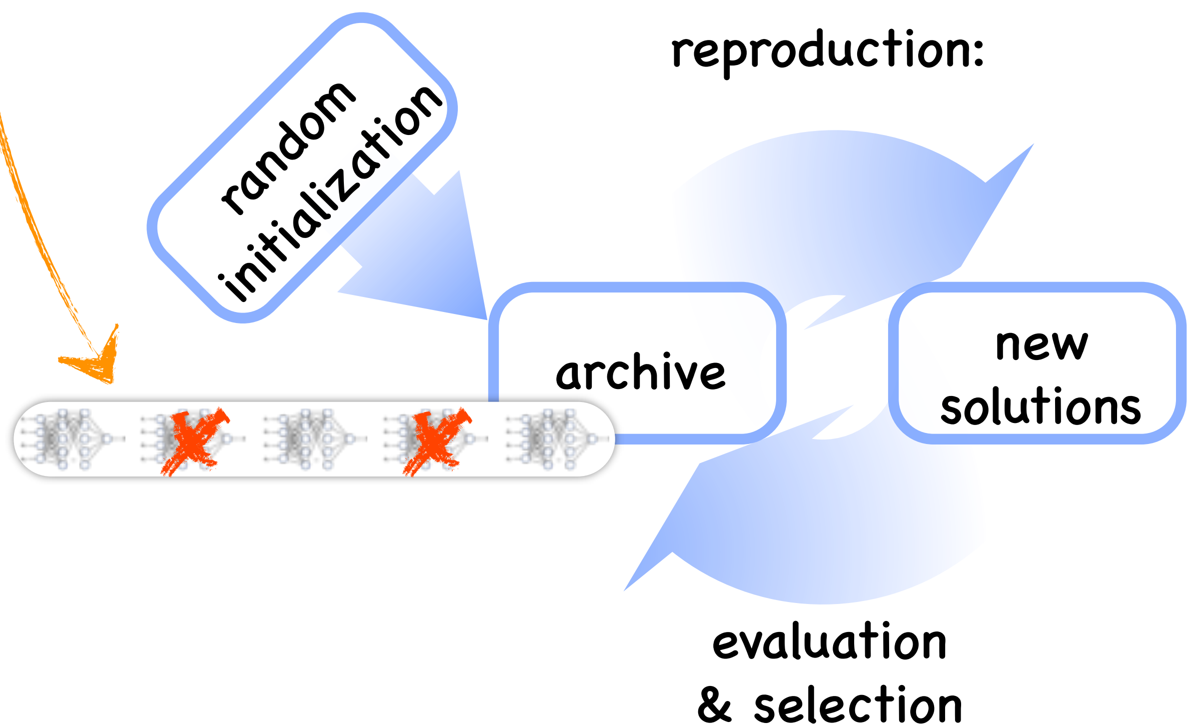


App. 1: selective ensemble

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Ensemble Pruning*. **AAAI'15**]

Pareto Ensemble Pruning (PEP):

1. random generate a pruned ensemble, put it into the archive
2. loop
   - 2.1 pick an ensemble randomly from the archive
   - 2.2 randomly change it to make a new one
   - 2.3 if the new one is not dominated
     - 2.3.1 put it into the archive
     - 2.3.2 put its good neighbors into the archive
3. when terminates, select an ensemble from the archive

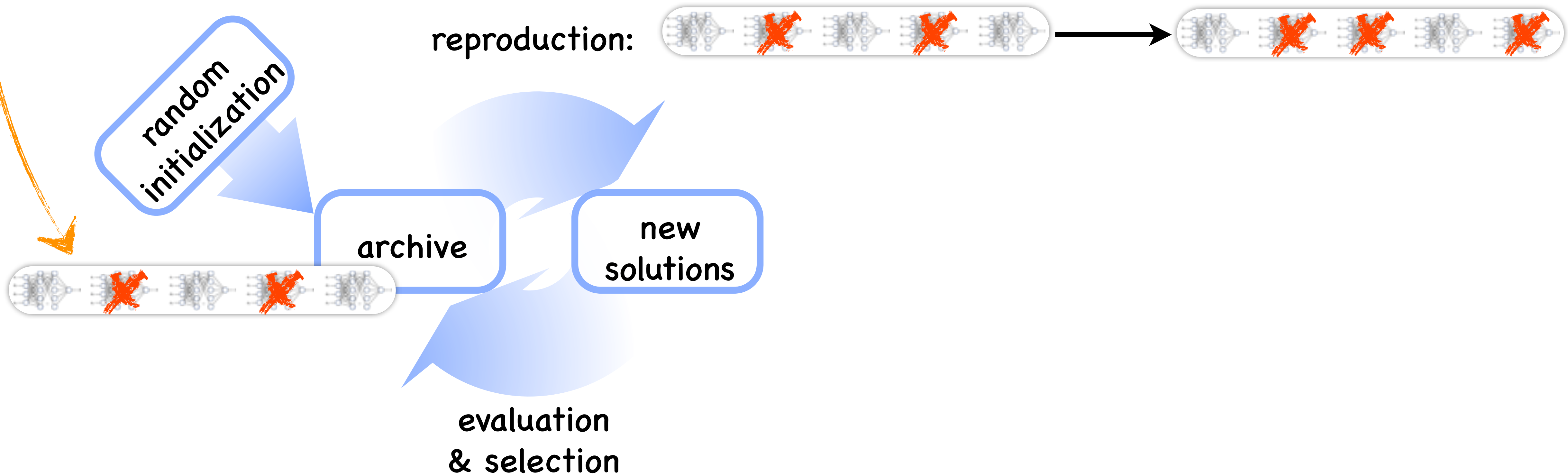


App. 1: selective ensemble

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Ensemble Pruning*. **AAAI'15**]

Pareto Ensemble Pruning (PEP):

1. random generate a pruned ensemble, put it into the archive
2. loop
| 2.1 pick an ensemble randomly from the archive
| 2.2 randomly change it to make a new one
| 2.3 if the new one is not dominated
| | 2.3.1 put it into the archive
| | 2.3.2 put its good neighbors into the archive
3. when terminates, select an ensemble from the archive

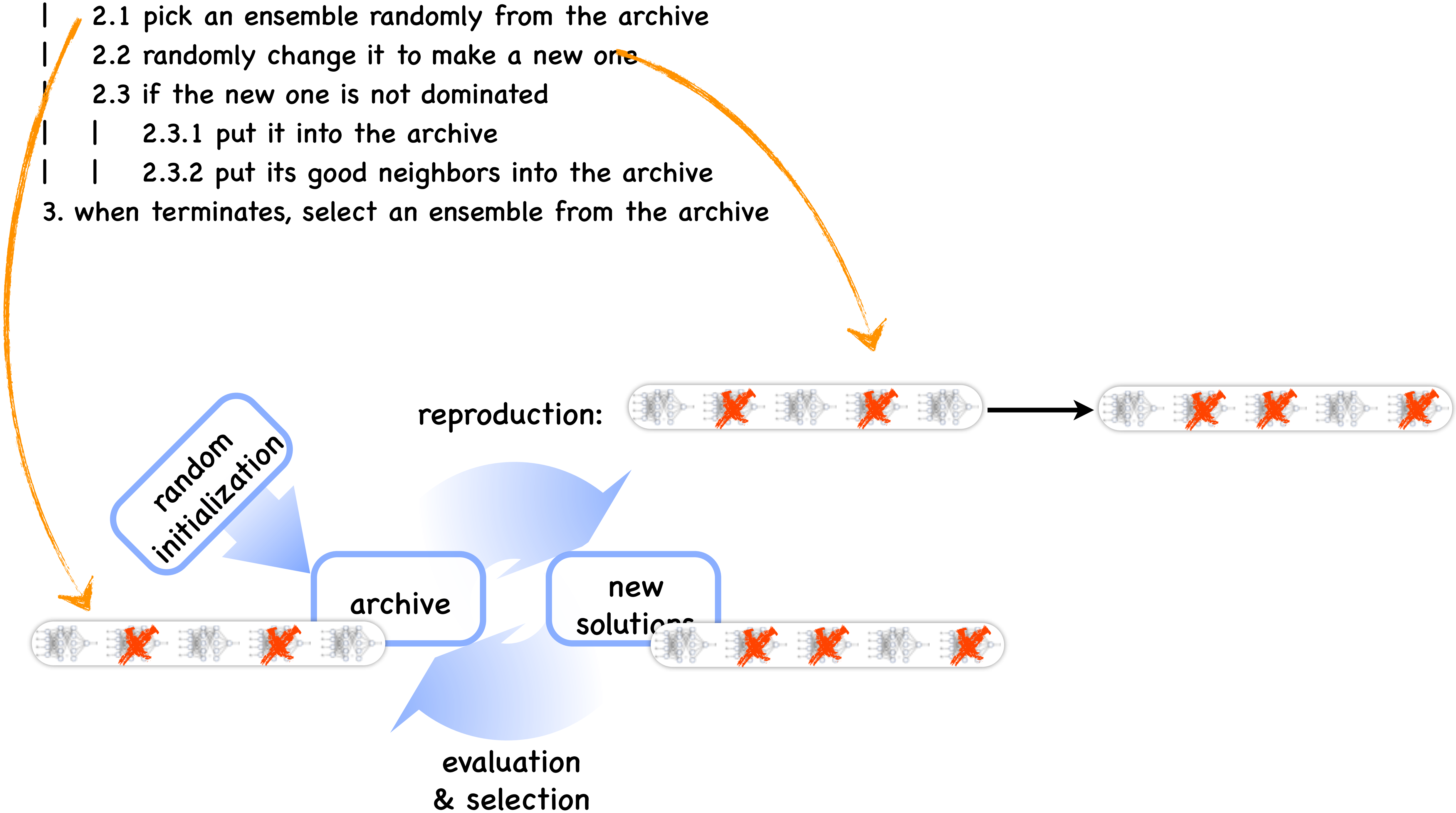


App. 1: selective ensemble

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Ensemble Pruning*. **AAAI'15**]

Pareto Ensemble Pruning (PEP):

1. random generate a pruned ensemble, put it into the archive
2. loop
| 2.1 pick an ensemble randomly from the archive
| 2.2 randomly change it to make a new one
| 2.3 if the new one is not dominated
| | 2.3.1 put it into the archive
| | 2.3.2 put its good neighbors into the archive
3. when terminates, select an ensemble from the archive

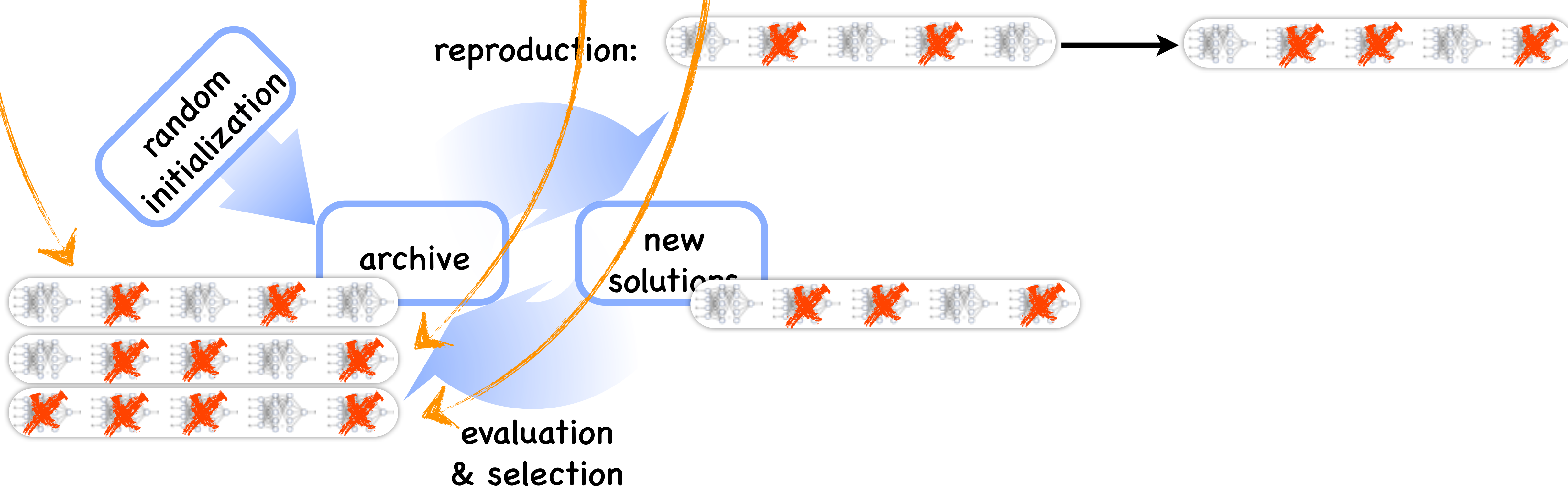


App. 1: selective ensemble

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Ensemble Pruning*. **AAAI'15**]

Pareto Ensemble Pruning (PEP):

1. random generate a pruned ensemble, put it into the archive
2. loop
| 2.1 pick an ensemble randomly from the archive
| 2.2 randomly change it to make a new one
| 2.3 if the new one is not dominated
| | 2.3.1 put it into the archive
| | 2.3.2 put its good neighbors into the archive
3. when terminates, select an ensemble from the archive



Theoretical advantages

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Ensemble Pruning*. **AAAI'15**]

Previously, hard to perform theoretical comparison

Now:

Theoretical advantages

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Ensemble Pruning*. **AAAI'15**]

Previously, hard to perform theoretical comparison

Now:

✓ PEP is at least as good as ordering-based methods

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

Theoretical advantages

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Ensemble Pruning*. **AAAI'15**]

Previously, hard to perform theoretical comparison

Now:

- ✓ PEP is at least as good as ordering-based methods
- ✓ PEP can be better than ordering-based methods

Situation 1.

$$\exists H' \subseteq H, |H'| = 3 \wedge \forall g,h \in H', \mathit{diff}(g,h) = \mathit{err}(g) + \mathit{err}(h);$$

$$\exists h^* \in H - H', \begin{cases} \mathit{err}(h^*) < \min\{\mathit{err}(h) | h \in H'\}, \\ \forall h \in H', \mathit{diff}(h, h^*) < \mathit{err}(h) + \mathit{err}(h^*); \end{cases}$$

$$\forall g \in H - H' - \{h^*\}, \mathit{err}(g) > \max\{\mathit{err}(h) | h \in H'\}$$
$$\wedge\, \mathit{err}(g) + \mathit{err}(h^*) - \mathit{diff}(g, h^*) >$$
$$(\min + \max)\{\mathit{err}(h) + \mathit{err}(h^*) - \mathit{diff}(h, h^*) | h \in H'\}.$$

Theorem 2. *In Situation 1, OEP using Eq.1 finds a solution with objective vector $(\geq 0, \geq 3)$ where the two equalities never hold simultaneously, while PEP finds a solution with objective vector $(0, 3)$ in $O(n^4 \log n)$ expected time.*

Theoretical advantages

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Ensemble Pruning*. **AAAI'15**]

Previously, hard to perform theoretical comparison

Now:

✓ PEP is at least as good as ordering-based methods

✓ PEP can be better than ordering-based methods

✓ PEP/ordering-based methods can be better than the direct use of heuristic search

Situation 2.
$\exists H' \subseteq H,\ |H'| = n-1 \ \wedge\ \forall g,h \in H', \mathit{diff}(g,h) = 0;$
$\mathit{err}(H - H') < \mathit{err}(h \in H').$

Theorem 3. *In Situation 2, OEP using Eq.1 finds the optimal solution in $O(n^2)$ optimization time, while the time of SEP is at least $2^{\Omega(n)}$ with probability $1 - 2^{-\Omega(n)}$.*

Theoretical advantages

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Ensemble Pruning*. **AAAI'15**]

Previously, hard to perform theoretical comparison

Now:

✓ PEP is at least as good as ordering-based methods

✓ PEP can be better than ordering-based methods

✓ PEP/ordering-based methods can be better than the direct use of heuristic search

Empirical comparison

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Ensemble Pruning*. **AAAI’15**]

Pruning bagging base learners with size 100

Test Error									
Data set	PEP	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	.195±.038	.235±.049●	**.187±.044**	.201±.038	.199±.044	.226±.048●	.194±.044	.196±.032
house-votes	.045±.019	**.041±.013**	.047±.016	.043±.018	.044±.017	.045±.017	.048±.018●	.045±.017	**.041±.012**
ionosphere	.088±.021	.092±.025	.117±.022●	.086±.021	**.084±.020**	.089±.021	.100±.026●	.085±.021	.093±.026
kr-vs-kp	**.010±.003**	.015±.007●	.011±.004	**.010±.004**	**.010±.003**	.011±.003	.011±.005	.011±.003	.012±.004
letter-ah	.013±.005	.021±.006●	.023±.008●	.015±.006●	**.012±.006**	.015±.006	.017±.007●	.014±.005	.017±.006●
letter-br	**.046±.008**	.059±.013●	.078±.012●	.048±.012	.048±.014	.048±.012	.057±.014●	.048±.009	.053±.011●
letter-oq	.043±.009	.049±.012●	.078±.017●	.046±.011	.042±.011	.042±.010	.046±.011	**.041±.010**	.044±.011
optdigits	**.035±.006**	.038±.007●	.095±.008●	.036±.006	**.035±.005**	.036±.005	.037±.006●	**.035±.006**	**.035±.006**
satimage-12v57	**.028±.004**	.029±.004	.052±.006●	.029±.004	**.028±.004**	.029±.004	.029±.004	.029±.004	.029±.004
satimage-2v5	**.021±.007**	.023±.009	.033±.010●	.023±.007	.022±.007	**.021±.008**	.026±.010●	.022±.008	**.021±.008**
sick	**.015±.003**	.018±.004●	.018±.004●	.016±.003	.017±.003●	.016±.003●	.017±.003●	.016±.003	.017±.004●
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
PEP: count of direct win		**17**	**20**	**15.5**	12.5	**17**	**20**	12.5	**15.5**
Ensemble Size									
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	–	–	**13.9±4.2**	24.7±16.3●	15.3±10.6	17.7±20.0	**13.9±5.9**	42.0±6.2●
heart-statlog	**9.3±2.3**	–	–	11.4±5.0●	17.9±11.1●	13.2±8.2●	13.6±21.1	11.3±2.7●	44.2±5.1●
house-votes	**2.9±1.7**	–	–	3.9±4.0	5.5±3.3●	4.7±4.4●	5.9±14.1	4.1±2.7●	46.5±6.1●
ionosphere	**5.2±2.2**	–	–	7.9±5.7●	10.5±6.9●	8.5±6.3●	10.7±14.6●	8.4±4.3●	48.8±5.1●
kr-vs-kp	**4.2±1.8**	–	–	5.8±4.5	10.6±9.1●	9.6±8.6●	7.2±15.2	7.1±3.9●	45.9±5.8●
letter-ah	**5.0±1.9**	–	–	7.3±4.4●	7.1±3.8●	8.7±4.7●	11.0±10.9●	7.8±3.6●	42.5±6.5●
letter-br	**10.9±2.6**	–	–	15.1±7.3●	13.8±6.7●	12.9±6.8	23.2±17.6●	11.3±3.5	38.3±7.8●
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	–	–	25.0±9.3	25.2±8.1	**21.4±7.5**	46.8±23.9●	25.0±8.0	41.4±7.6●
satimage-12v57	**17.1±5.0**	–	–	20.8±9.2●	22.1±10.3●	21.2±10.0●	37.6±24.3●	18.1±4.9	42.7±5.2●
satimage-2v5	**5.7±1.7**	–	–	6.8±3.2	7.6±4.2●	10.9±7.0●	26.2±28.1●	7.7±3.5●	44.1±4.8●
sick	**6.9±2.8**	–	–	7.5±3.9	10.9±6.0●	11.5±10.0●	8.3±13.6	11.6±6.7●	44.7±8.2●
sonar	11.4±4.2	–	–	**11.0±4.1**	20.6±9.3●	13.9±7.1	20.6±20.7●	14.4±5.9●	43.1±6.4●
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	13.2±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
PEP: count of direct win		–	–	**17**	**19.5**	**18**	**17.5**	**16**	**20**

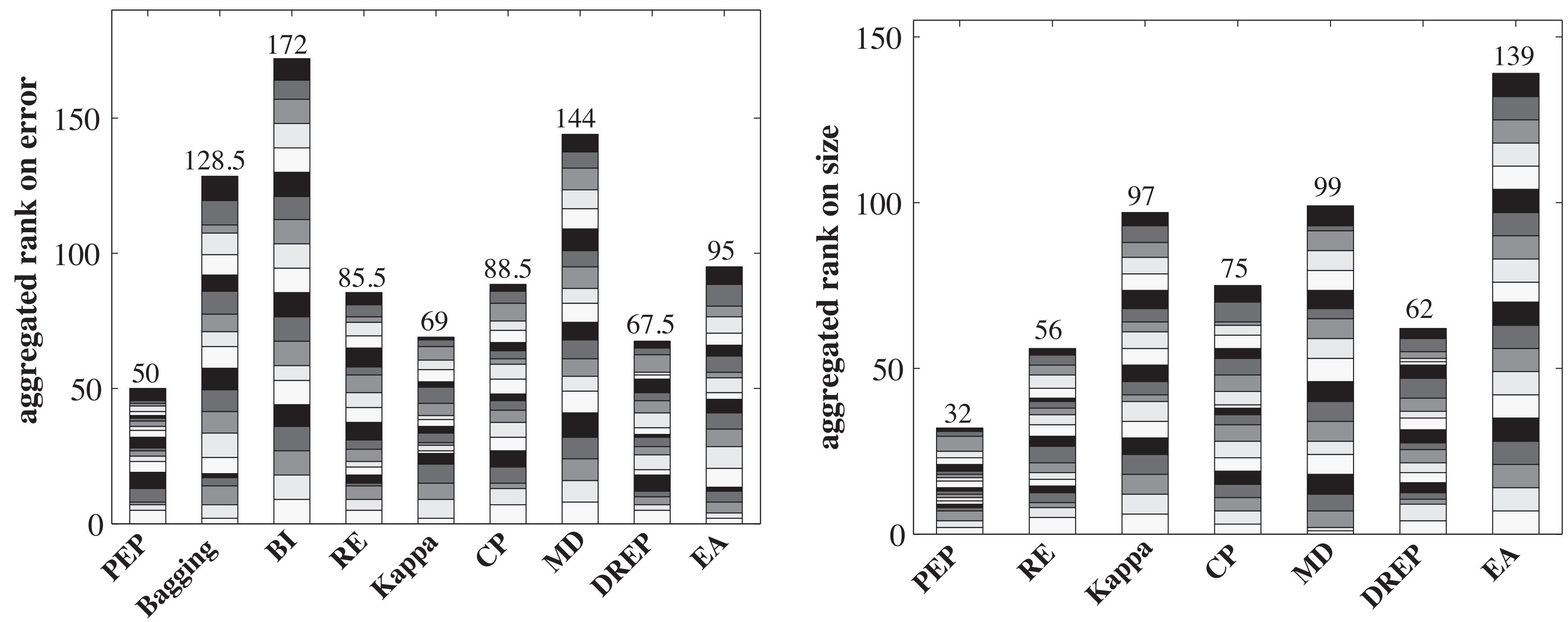
Empirical comparison

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Ensemble Pruning*. **AAAI'15**]

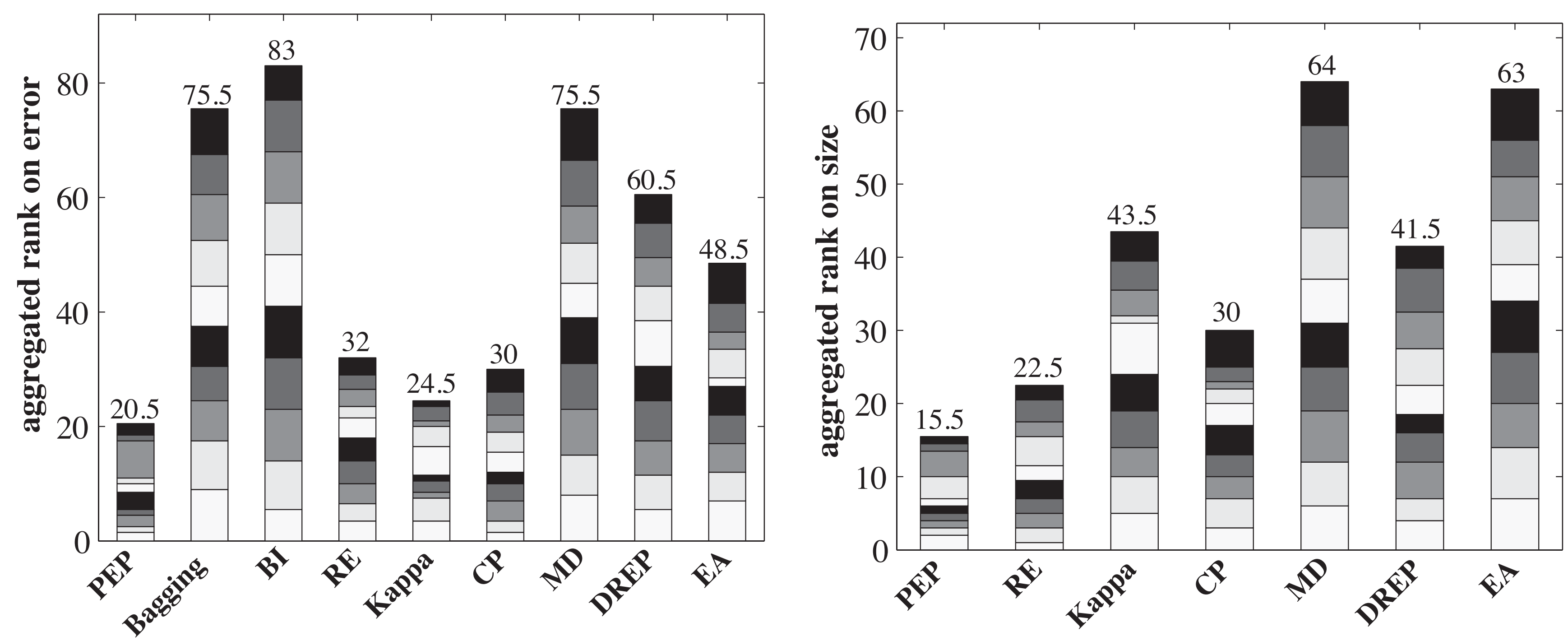
Pruning bagging base learners with size 100

on error

on size



(a) on 20 binary data sets



(b) on 10 multiclass data sets

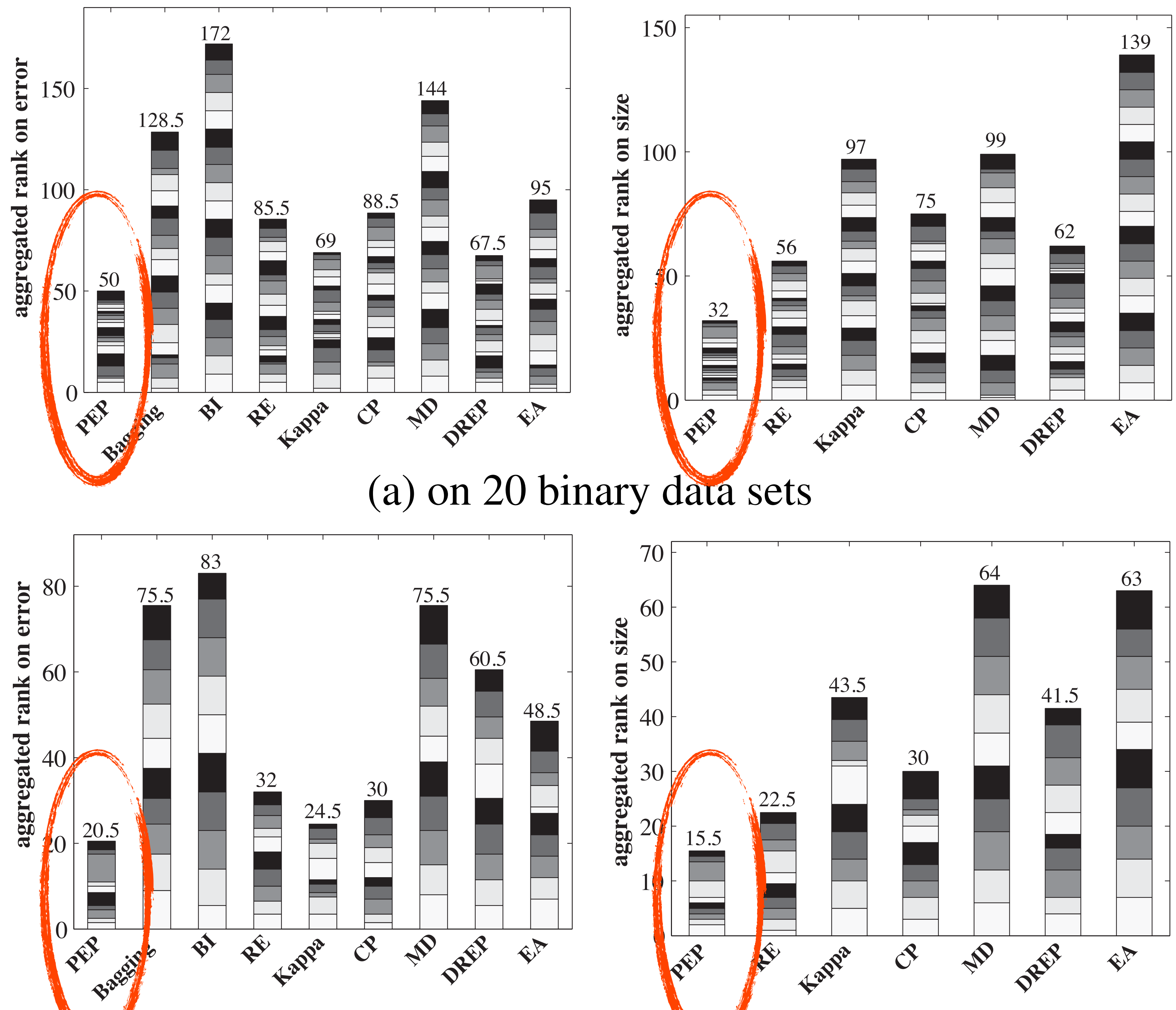
Empirical comparison

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Ensemble Pruning*. **AAAI'15**]

Pruning bagging base learners with size 100

on error

on size



(a) on 20 binary data sets

(b) on 10 multiclass data sets

App 2: Sparse regression

Regression: $\arg\min_{\boldsymbol{w}} \sum_{(\boldsymbol{x},y)\in D} (\boldsymbol{w}^\top \boldsymbol{x} - y)^2$

Sparse regression (sparsity k): another subset selection problem

$$\arg\min_{\boldsymbol{w}} \sum_{(\boldsymbol{x},y)\in D} (\boldsymbol{w}^\top \boldsymbol{x} - y)^2 \quad s.t. \quad \|\boldsymbol{w}\|_0 \leq k$$

$\|w\|_0$ denotes the number of non-zero elements in w

App 2: Sparse regression

Regression: $\arg\min_{\boldsymbol{w}} \sum_{(\boldsymbol{x},y)\in D} (\boldsymbol{w}^\top \boldsymbol{x} - y)^2$

Sparse regression (sparsity k): another subset selection problem

$$\arg\min_{\boldsymbol{w}} \sum_{(\boldsymbol{x},y)\in D} (\boldsymbol{w}^\top \boldsymbol{x} - y)^2 \quad s.t. \quad \|\boldsymbol{w}\|_0 \leq k$$

$\|\boldsymbol{w}\|_0$ denotes the number of non-zero elements in w

Previous methods

Greedy methods [Gilbert et al.,2003; Tropp, 2004]

Forward (FR)

Current best approximation ratio: $1 - e^{-\gamma}$ on R^2 [Das and Kempe, ICML'11]

Forward-Backward (FoBa), Orthogonal Matching Pursuit (OMP) ...

Convex relaxation methods [Tibshirani, 1996; Zou & Hastie, 2005]

$$\arg\min_{\boldsymbol{w}} \sum_{(\boldsymbol{x},y)\in D} (\boldsymbol{w}^\top \boldsymbol{x} - y)^2 \quad s.t. \quad \|\boldsymbol{w}\|_1 \leq k$$

Our approach

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Optimization for Subset Selection*. **NIPS'15**]

by Pareto optimization method:

$$\arg\min_{\boldsymbol{w}} \sum_{(\boldsymbol{x},y)\in D} (\boldsymbol{w}^\top \boldsymbol{x} - y)^2 \quad s.t. \quad \|\boldsymbol{w}\|_0 \leq k$$

Our approach

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Optimization for Subset Selection*. **NIPS'15**]

by Pareto optimization method:

$$\arg\min_{\boldsymbol{w}} \sum_{(\boldsymbol{x},y)\in D} (\boldsymbol{w}^\top \boldsymbol{x} - y)^2 \quad s.t. \quad \|\boldsymbol{w}\|_0 \leq k$$

reduce MSE **reduce size**

sparse regression can be divided into two goals

Our approach

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Optimization for Subset Selection*. **NIPS'15**]

by Pareto optimization method:

$$\arg\min_{\boldsymbol{w}} \sum_{(\boldsymbol{x},y)\in D} (\boldsymbol{w}^\top \boldsymbol{x} - y)^2 \quad s.t. \quad \|\boldsymbol{w}\|_0 \leq k$$

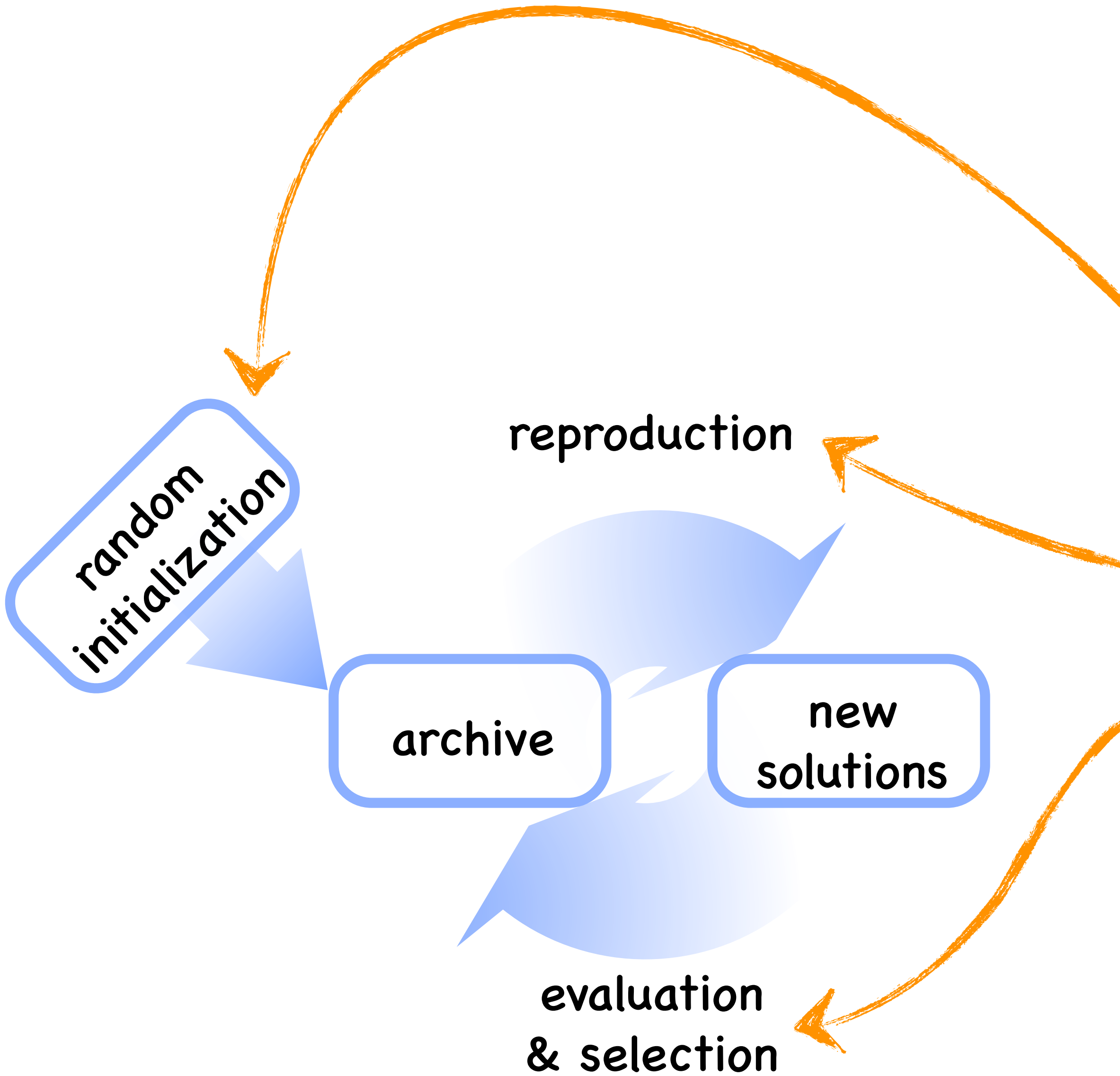
reduce MSE **reduce size**

sparse regression can be divided into two goals

$$\boldsymbol{s}.o_1 = \begin{cases} +\infty, & \boldsymbol{s} = \{0\}^n, \text{ or } |\boldsymbol{s}| \geq 2k \\ f(\boldsymbol{s}), & \text{otherwise} \end{cases}, \qquad \boldsymbol{s}.o_2 = |\boldsymbol{s}|.$$

Our approach

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Optimization for Subset Selection*. **NIPS'15**]



Algorithm 2 POSS

Input: all observation variables $V = \{X_1, \ldots, X_n\}$, a given criterion f and an integer parameter $k \in [1, n]$
Parameter: the number of iterations T and an isolation function $I : \{0,1\}^n \to R$
Output: a subset of V with at most k variables
Process:

Let $\boldsymbol{s} = \{0\}^n$ and $P = \{\boldsymbol{s}\}$.
Let $t = 0$.
while $t < T$ **do**
  Select $\boldsymbol{s}$ from P uniformly at random.
  Generate $\boldsymbol{s}'$ from $\boldsymbol{s}$ by flipping each bit of $\boldsymbol{s}$ with probability $\frac{1}{n}$.
  if $\nexists \boldsymbol{z} \in P$ such that $I(\boldsymbol{z}) = I(\boldsymbol{s}')$ and $\big((\boldsymbol{z}.o_1 < \boldsymbol{s}'.o_1 \wedge \boldsymbol{z}.o_2 \leq \boldsymbol{s}'.o_2)$ or $(\boldsymbol{z}.o_1 \leq \boldsymbol{s}'.o_1 \wedge \boldsymbol{z}.o_2 < \boldsymbol{s}'.o_2)\big)$ **then**
    $Q = \{\boldsymbol{z} \in P \mid I(\boldsymbol{z}) = I(\boldsymbol{s}') \wedge \boldsymbol{s}'.o_1 \leq \boldsymbol{z}.o_1 \wedge \boldsymbol{s}'.o_2 \leq \boldsymbol{z}.o_2\}$.
    $P = (P \setminus Q) \cup \{\boldsymbol{s}'\}$.
  end if
  $t = t + 1$.
end while
return $\arg\min_{\boldsymbol{s} \in P, |\boldsymbol{s}| \leq k} f(\boldsymbol{s})$

Theoretical advantages

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Optimization for Subset Selection*. **NIPS'15**]

Is POSS as good as the previously best method (FR) ?

Theoretical advantages

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Optimization for Subset Selection*. **NIPS'15**]

Is POSS as good as the previously best method (FR) ?

✓ Yes, POSS can achieve the same approximation ratio

Theorem 1. *For sparse regression, POSS with* $E[T] \leq 2ek^2n$ *and* $I(\cdot) = 0$ *(i.e., a constant function) finds a set* S *of variables with* $|S| \leq k$ *and* $R^2_{Z,S} \geq (1 - e^{-\gamma_{\emptyset,k}}) \cdot OPT$.

Theoretical advantages

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Optimization for Subset Selection*. **NIPS'15**]

Is POSS as good as the previously best method (FR) ?

✓ Yes, POSS can achieve the same approximation ratio

Theorem 1. *For sparse regression, POSS with $E[T] \leq 2ek^2n$ and $I(\cdot) = 0$ (i.e., a constant function) finds a set S of variables with $|S| \leq k$ and $R^2_{Z,S} \geq (1 - e^{-\gamma_{\emptyset,k}}) \cdot OPT$.*

Can POSS be better ?

Theoretical advantages

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Optimization for Subset Selection*. **NIPS'15**]

Is POSS as good as the previously best method (FR) ?

✓ Yes, POSS can achieve the same approximation ratio

Theorem 1. *For sparse regression, POSS with $E[T] \leq 2ek^2n$ and $I(\cdot) = 0$ (i.e., a constant function) finds a set S of variables with $|S| \leq k$ and $R^2_{Z,S} \geq (1 - e^{-\gamma_{\emptyset,k}}) \cdot OPT$.*

Can POSS be better ?

✓ Yes, POSS can solve exact solutions on problem subclasses, while FR cannot

Theorem 2. *For the Exponential Decay subclass of sparse regression, POSS with $E[T] \in O(k^2n^2 \log n)$ and $I(\boldsymbol{s} \in \{0,1\}^n) = \min\{i \mid s_i = 1\}$ can find the optimal solution.*

Empirical comparison

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Optimization for Subset Selection*. **NIPS'15**]

select 8 features, report R^2 (the larger the better), average over 100 runs

data set	#inst	#feat	data set	#inst	#feat
housing	506	13	*coil2000*	9000	86
eunite2001	367	16	*mushrooms*	8124	112
svmguide3	1284	21	*clean1*	476	166
ionosphere	351	34	*w5a*	9888	300
sonar	208	60	*gisette*	7000	5000
triazines	186	60	*farm-ads*	4143	54877

Empirical comparison

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Optimization for Subset Selection*. **NIPS'15**]

select 8 features, report R^2 (the larger the better), average over 100 runs

data set	#inst	#feat	data set	#inst	#feat
housing	506	13	*coil2000*	9000	86
eunite2001	367	16	*mushrooms*	8124	112
svmguide3	1284	21	*clean1*	476	166
ionosphere	351	34	*w5a*	9888	300
sonar	208	60	*gisette*	7000	5000
triazines	186	60	*farm-ads*	4143	54877

Data set	OPT	POSS	FR	FoBa	OMP	L1
housing	.7437±.0297	.7437±.0297	.7429±.0300•	.7423±.0301•	.7415±.0300•	.7230±.0330•
eunite2001	.8484±.0132	.8482±.0132	.8348±.0143•	.8442±.0144•	.8349±.0150•	.8183±.0247•
svmguide3	.2705±.0255	.2701±.0257	.2615±.0260•	.2601±.0279•	.2557±.0270•	.2247±.0241•
ionosphere	.5995±.0326	.5990±.0329	.5920±.0352•	.5929±.0346•	.5921±.0353•	.5173±.0408•
sonar	–	.5365±.0410	.5171±.0440•	.5138±.0432•	.5112±.0425•	.3309±.0652•
triazines	–	.4301±.0603	.4150±.0592•	.4107±.0600•	.4073±.0591•	.2665±.0691•
coil2000	–	.0627±.0076	.0624±.0076•	.0619±.0075•	.0619±.0075•	.0379±.0076•
mushrooms	–	.9912±.0020	.9909±.0021•	.9909±.0022•	.9909±.0022•	.8191±.0891•
clean1	–	.4368±.0300	.4169±.0299•	.4145±.0309•	.4132±.0315•	.2058±.0437•
w5a	–	.3376±.0267	.3319±.0247•	.3341±.0258•	.3313±.0246•	.1066±.0347•
gisette	–	.7265±.0098	.7001±.0116•	.6747±.0145•	.6731±.0134•	.4471±.0236•
farm-ads	–	.4240±.0093	.4215±.0093•	.4190±.0106•	.4190±.0106•	.2942±.0212•
POSS: win/tie/loss		–	12/0/0	12/0/0	12/0/0	12/0/0
average rank		1	2.5	2.83	3.67	5

Empirical comparison

[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Optimization for Subset Selection*. **NIPS'15**]

select 8 features, report R^2 (the larger the better), average over 100 runs

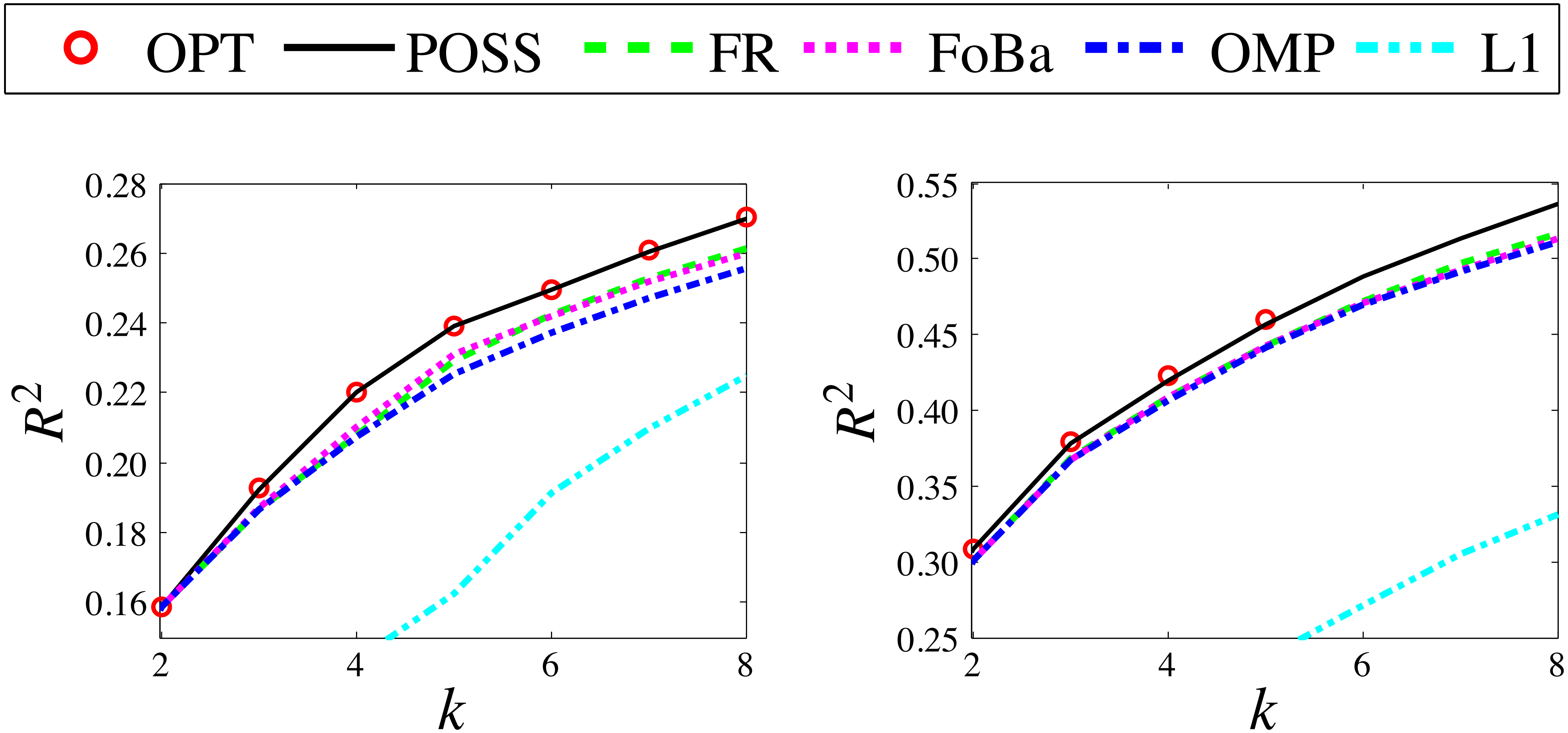
data set	#inst	#feat	data set	#inst	#feat
housing	506	13	*coil2000*	9000	86
eunite2001	367	16	*mushrooms*	8124	112
svmguide3	1284	21	*clean1*	476	166
ionosphere	351	34	*w5a*	9888	300
sonar	208	60	*gisette*	7000	5000
triazines	186	60	*farm-ads*	4143	54877

Data set	OPT	POSS	FR	FoBa	OMP	L1
housing	.7437±.0297	.7437±.0297	.7429±.0300•	.7423±.0301•	.7415±.0300•	.7230±.0330•
eunite2001	.8484±.0132	.8482±.0132	.8348±.0143•	.8442±.0144•	.8349±.0150•	.8183±.0247•
svmguide3	.2705±.0255	.2701±.0257	.2615±.0260•	.2601±.0279•	.2557±.0270•	.2247±.0241•
ionosphere	.5995±.0326	.5990±.0329	.5920±.0352•	.5929±.0346•	.5921±.0353•	.5173±.0408•
sonar	–	.5365±.0410	.5171±.0440•	.5138±.0432•	.5112±.0425•	.3309±.0652•
triazines	–	.4301±.0603	.4150±.0592•	.4107±.0600•	.4073±.0591•	.2665±.0691•
coil2000	–	.0627±.0076	.0624±.0076•	.0619±.0075•	.0619±.0075•	.0379±.0076•
mushrooms	–	.9912±.0020	.9909±.0021•	.9909±.0022•	.9909±.0022•	.8191±.0891•
clean1	–	.4368±.0300	.4169±.0299•	.4145±.0309•	.4132±.0315•	.2058±.0437•
w5a	–	.3376±.0267	.3319±.0247•	.3341±.0258•	.3313±.0246•	.1066±.0347•
gisette	–	.7265±.0098	.7001±.0116•	.6747±.0145•	.6731±.0134•	.4471±.0236•
farm-ads	–	.4240±.0093	.4215±.0093•	.4190±.0106•	.4190±.0106•	.2942±.0212•
POSS: win/tie/loss		–	12/0/0	12/0/0	12/0/0	12/0/0
average rank		1	2.5	2.83	3.67	5

Empirical comparison

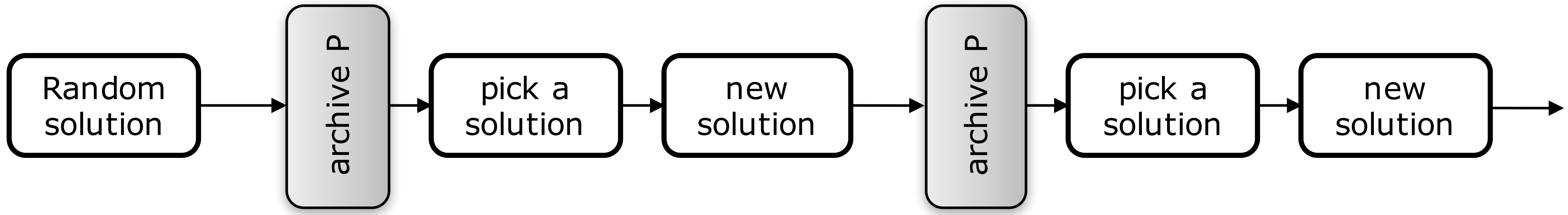
[C. Qian, Y. Yu and Z.-H. Zhou. *Pareto Optimization for Subset Selection*. **NIPS'15**]

Comparison optimization performance with different sparsities



(a) on *svmguide3* (b) on *sonar*

Extension: parallel Pareto optimization

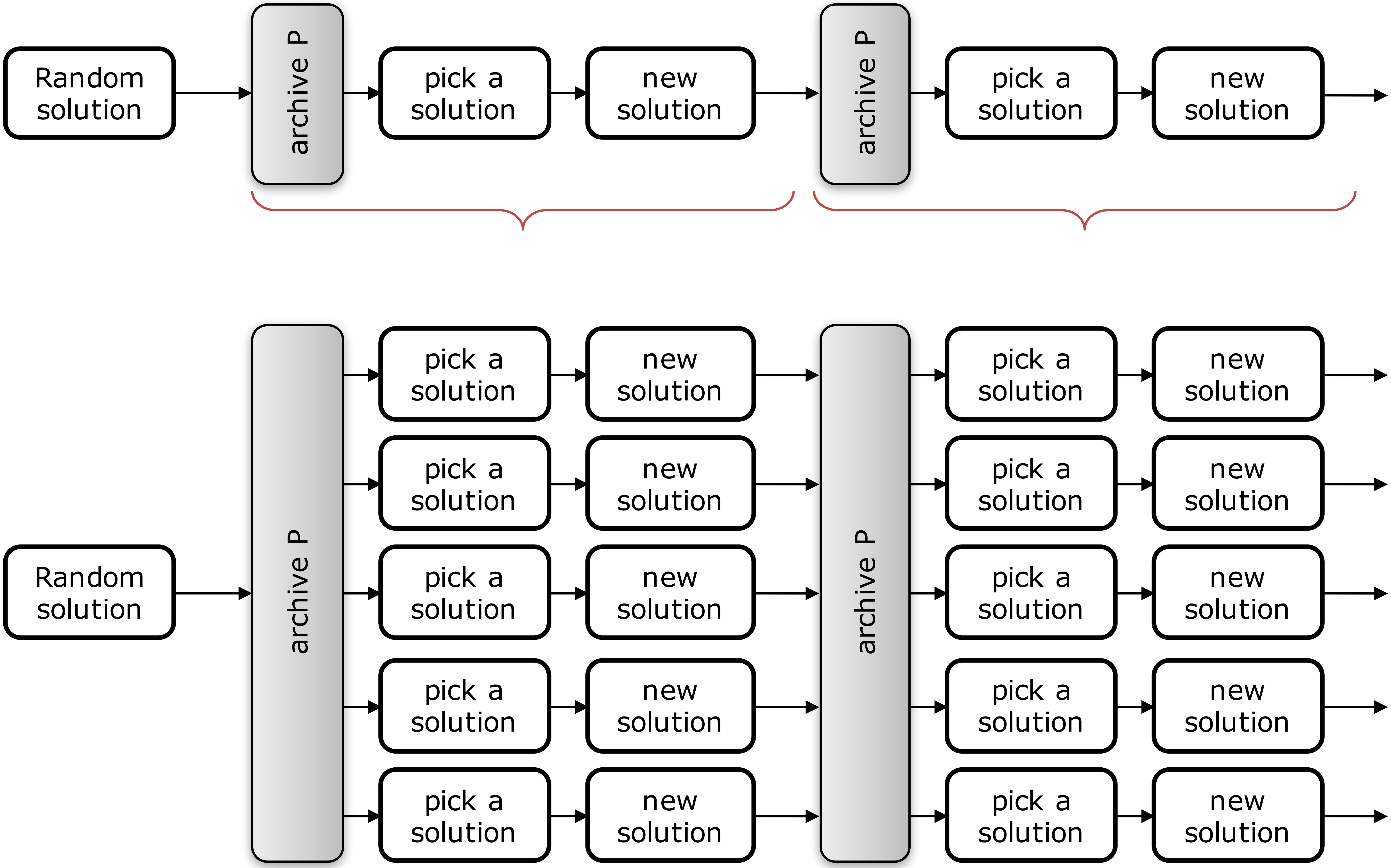


1. randomly generate a solution,
   and put it into the archive P;
2. loop
   | 2.1 pick a solution randomly from P;
   | 2.2 randomly change it to make a new one;
   | 2.3 if the new one is not ``strictly worse”
   | | 2.3.1 put it into P;
   | | 2.3.2 remove *worse* solutions from P;
3. when terminates, select the best feasible
   solution from P.

faster?

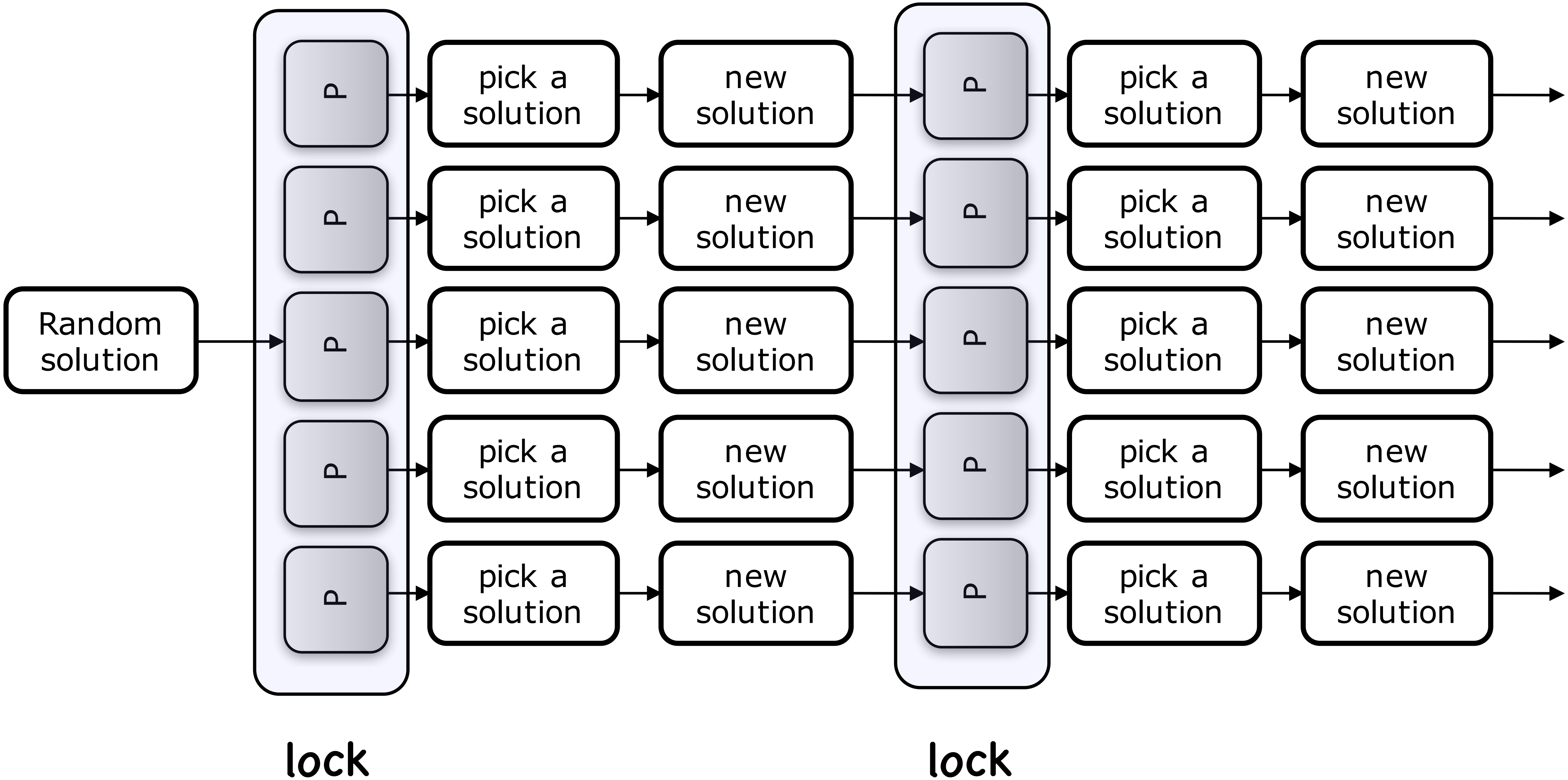
Parallel POSS

[C. Qian, J.-C. Shi, Y. Yu, K. Tang and Z.-H. Zhou. *Parallel Pareto Optimization for Subset Selection*. **IJCAI'16**]



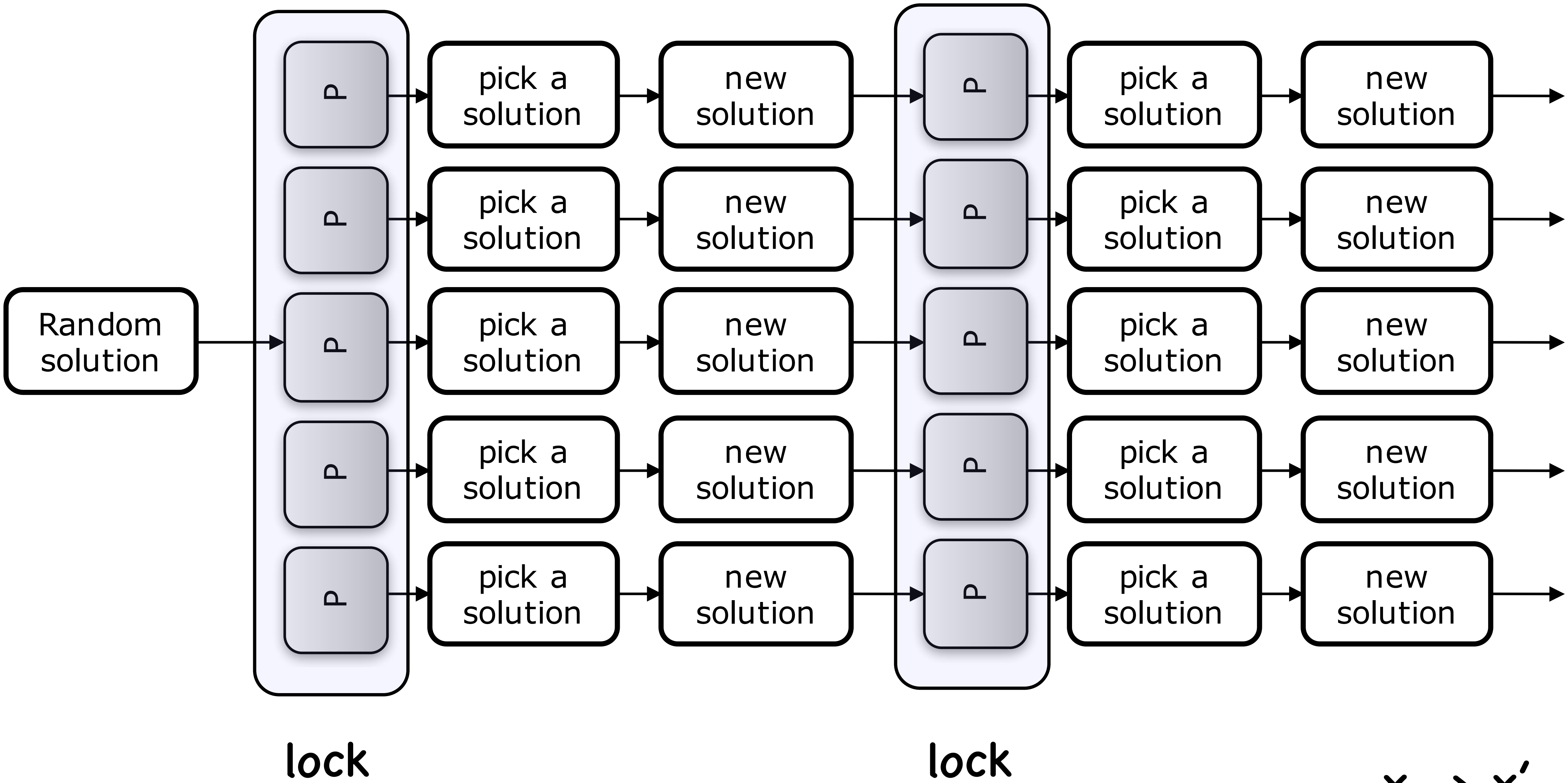
Parallel POSS

[C. Qian, J.-C. Shi, Y. Yu, K. Tang and Z.-H. Zhou. *Parallel Pareto Optimization for Subset Selection*. **IJCAI'16**]



Parallel POSS

[C. Qian, J.-C. Shi, Y. Yu, K. Tang and Z.-H. Zhou. *Parallel Pareto Optimization for Subset Selection*. **IJCAI'16**]

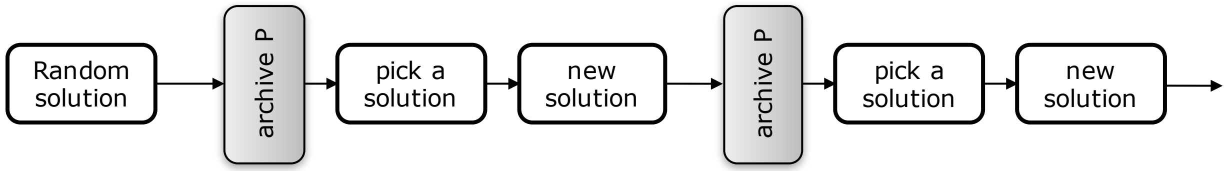


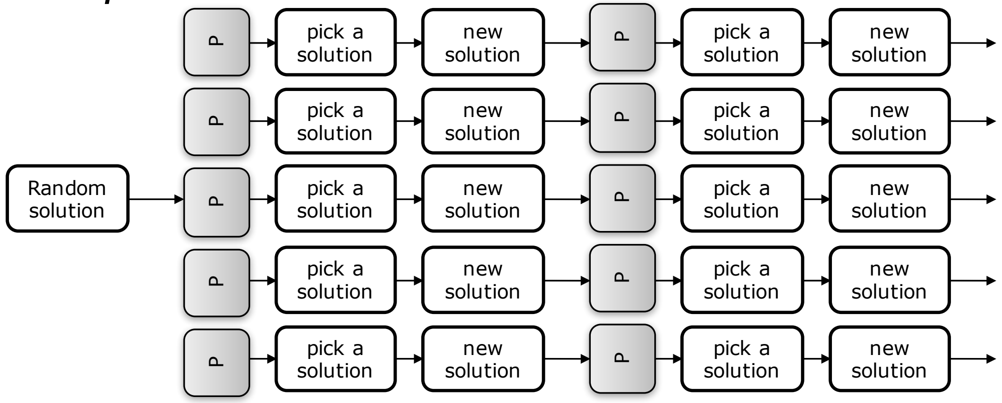
possible difference: POSS: $x \to x' \to x''$ PPOSS: $x \to x'$, $x \to x'$, $x \to x'$

can the parallelization preserve the effectiveness?

Parallel POSS

[C. Qian, J.-C. Shi, Y. Yu, K. Tang and Z.-H. Zhou. *Parallel Pareto Optimization for Subset Selection*. **IJCAI'16**]





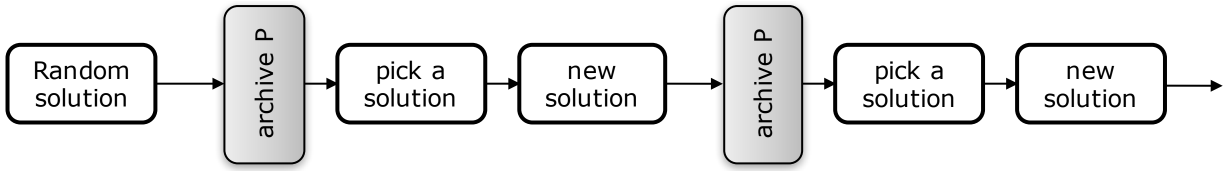
Theorem 1. For maximizing a monotone function under the set size constraint, the expected number of iterations until PPOSS finds a solution s with $|s| \leq k$ and $f(s) \geq (1 - e^{-\gamma_{\min}}) \cdot OPT$, where $\gamma_{\min} = \min_{s:|s|=k-1} \gamma_{s,k}$, is

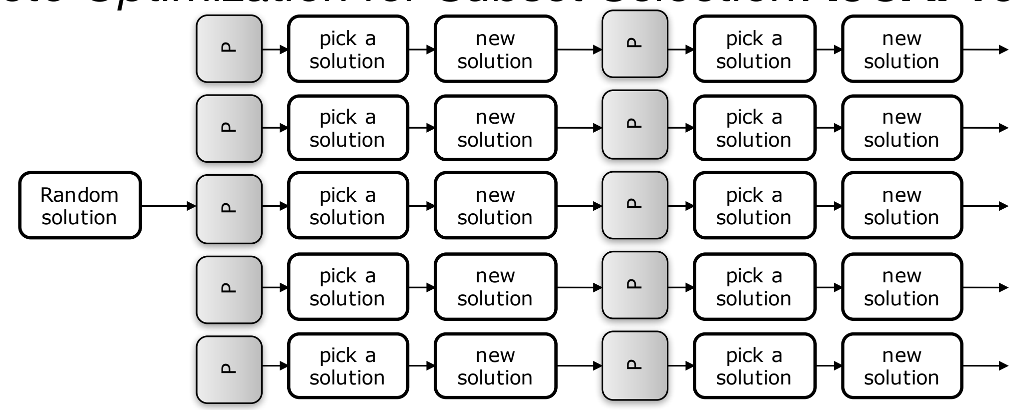
$$(1) \text{ if } N = o(n), \text{ then } \mathbb{E}[T] \leq 2ek^2 n/N;$$

When the number of processors is less than the number of variables, the number of iterations can be reduced linearly w.r.t. the number of processors

Parallel POSS

[C. Qian, J.-C. Shi, Y. Yu, K. Tang and Z.-H. Zhou. *Parallel Pareto Optimization for Subset Selection*. **IJCAI'16**]





Theorem 1. For maximizing a monotone function under the set size constraint, the expected number of iterations until PPOSS finds a solution s with $|s| \leq k$ and $f(s) \geq (1 - e^{-\gamma_{\min}}) \cdot OPT$, where $\gamma_{\min} = \min_{s:|s|=k-1} \gamma_{s,k}$, is

$$(2) \text{ if } N = \Omega(n^i) \text{ for } 1 \leq i \leq k, \text{ then } \mathbb{E}[T] = O(k^2/i);$$

$$(3) \text{ if } N = \Omega(n^{\min\{3k-1,n\}}), \text{ then } \mathbb{E}[T] = O(1).$$

With increasing number of processors, the number of iterations can be continuously reduced, eventually to a constant

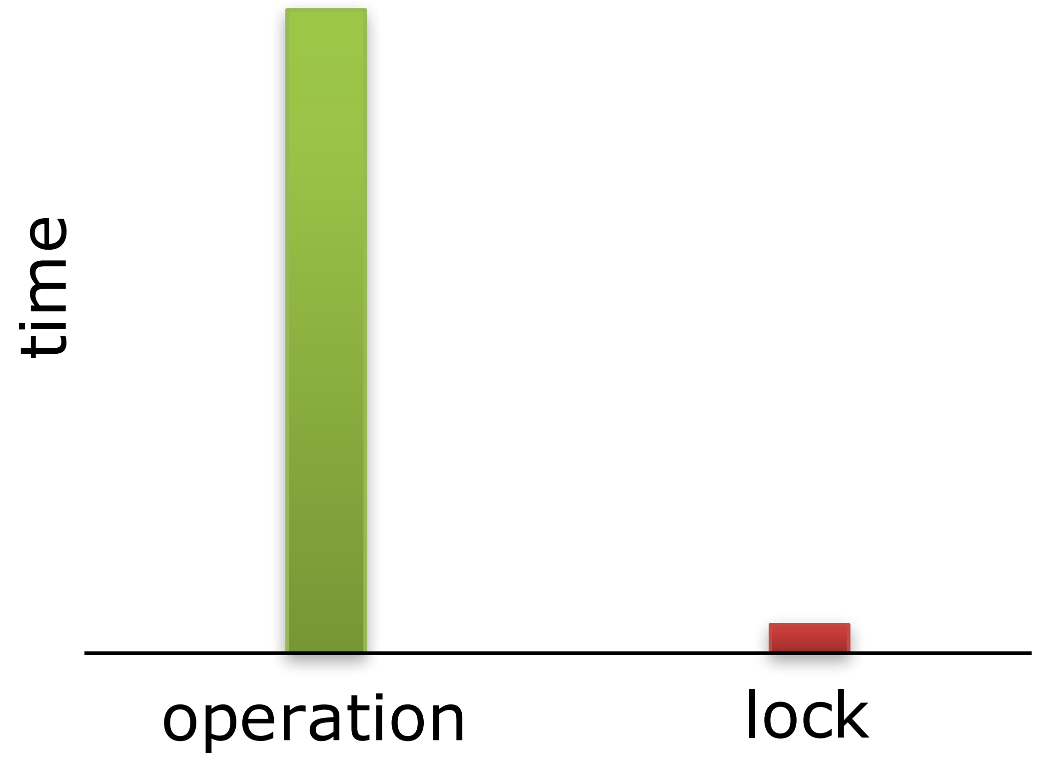
Parallel POSS

[C. Qian, J.-C. Shi, Y. Yu, K. Tang and Z.-H. Zhou. *Parallel Pareto Optimization for Subset Selection*. **IJCAI'16**]

Theorem 2. The expected difference between the running time of each iteration for POSS and PPOSS is

$$\mathbb{E}[t_{pposs} - t_{poss}] \leq (N-1) \cdot t_u + c/2.$$

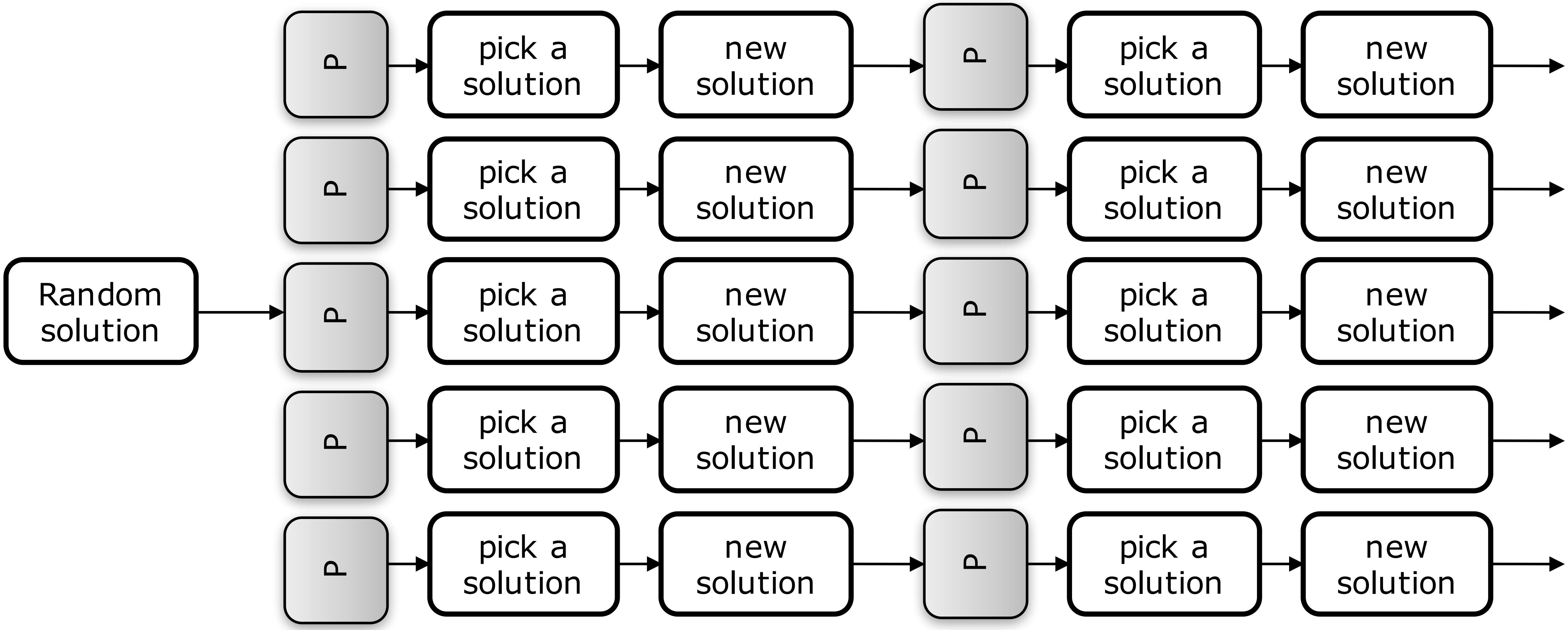
With a good approximation guarantee, the runtime decreases nearly linearly w.r.t. the number of processors



Lock-free version

[C. Qian, J.-C. Shi, Y. Yu, K. Tang and Z.-H. Zhou. *Parallel Pareto Optimization for Subset Selection*. **IJCAI'16**]

PPOSS-asy



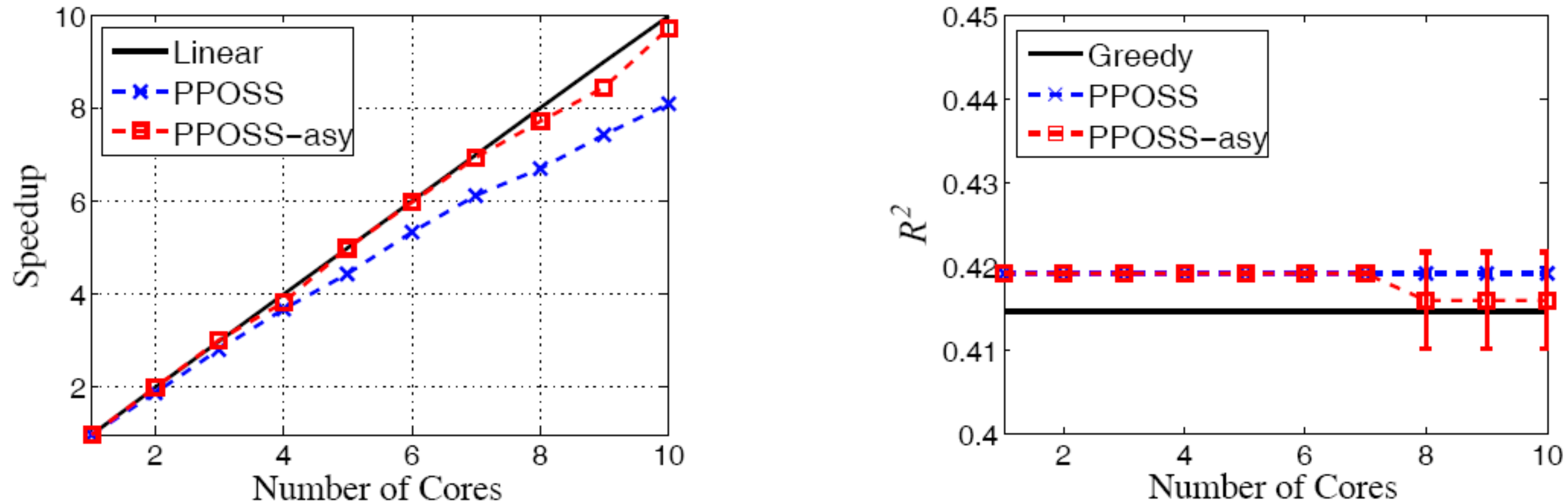
no lock

no lock

Experiments

[C. Qian, J.-C. Shi, Y. Yu, K. Tang and Z.-H. Zhou. *Parallel Pareto Optimization for Subset Selection*. **IJCAI'16**]

the asynchronous version of PPOSS　　the best previous algorithm [Das & Kempe, ICML'11]



(f) on *farm-ads* (4143 instances, 54877 features)

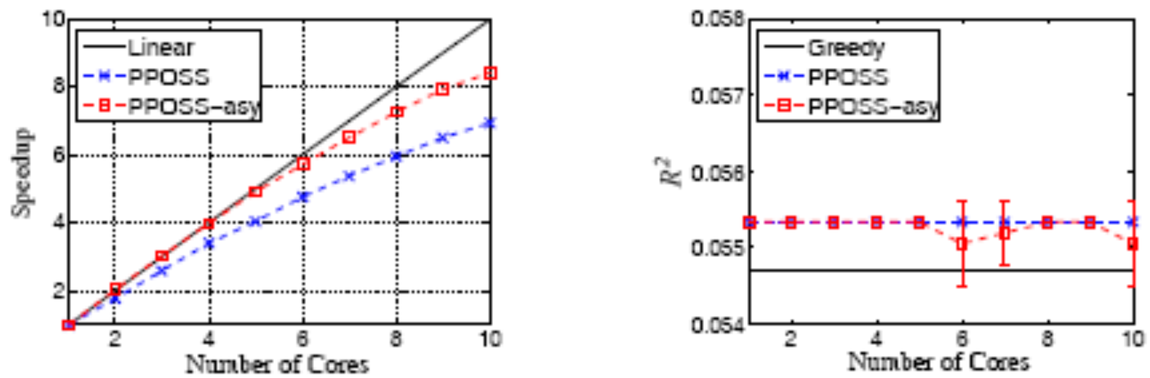
PPOSS (blue line): achieve speedup around 8 when the number of cores is 10; the R^2 values are stable

PPOSS-asy (red line): achieve better speedup (avoid the lock cost)**;**
the R^2 values are slightly worse (the noise from lock-free)

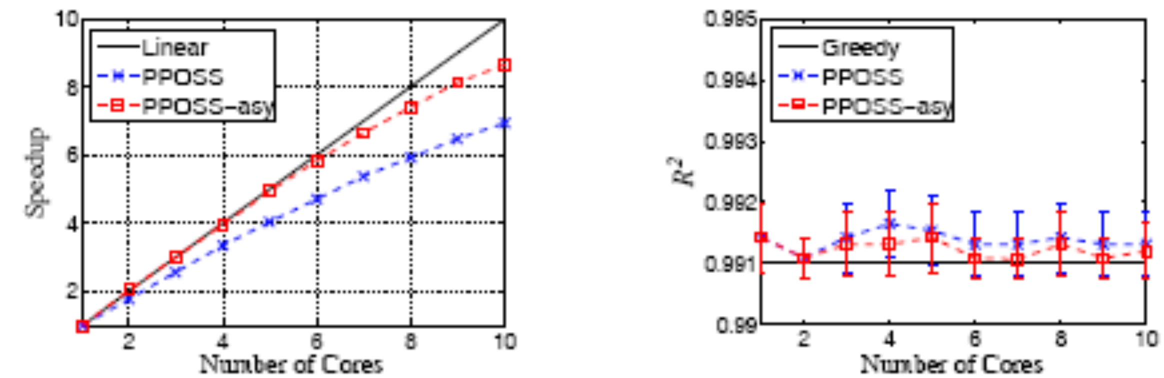
Experiments

[C. Qian, J.-C. Shi, Y. Yu, K. Tang and Z.-H. Zhou. *Parallel Pareto Optimization for Subset Selection*. **IJCAI'16**]

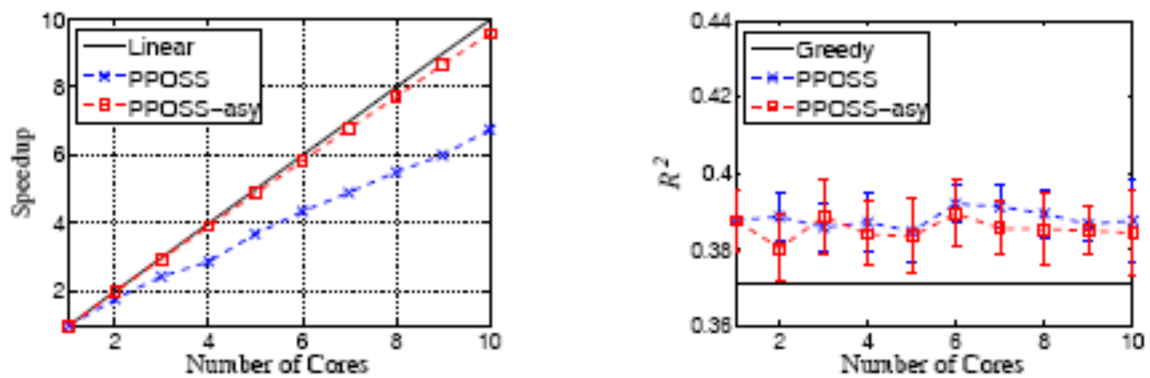
Compare the speedup as well as the solution quality measured by R^2 values with different number of cores



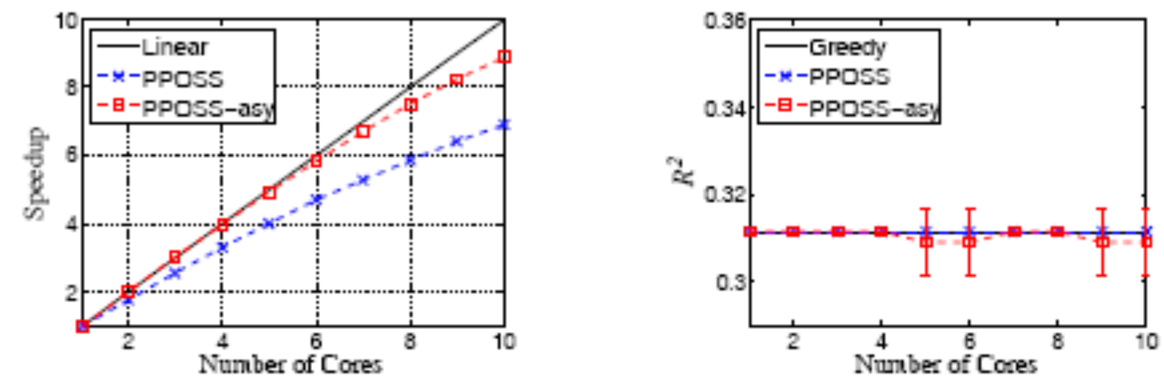
(a) on *coil2000* (9000 instances, 86 features)



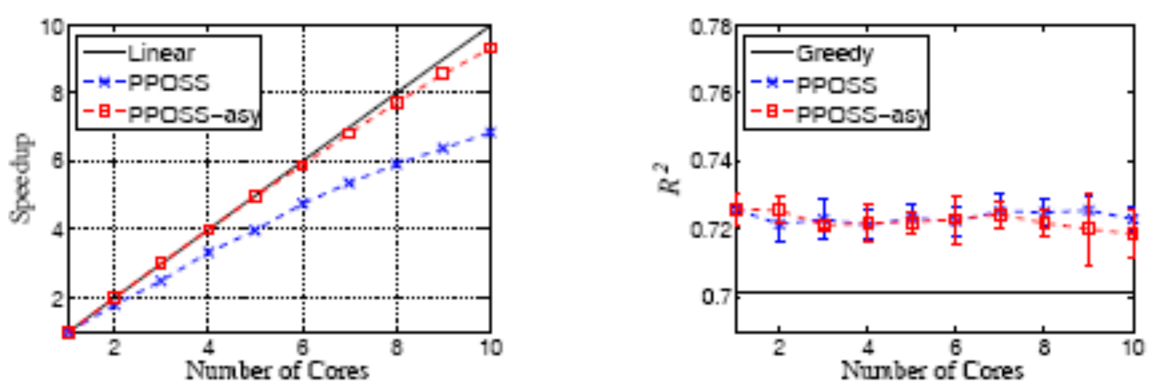
(b) on *mushrooms* (8124 instances, 112 features)



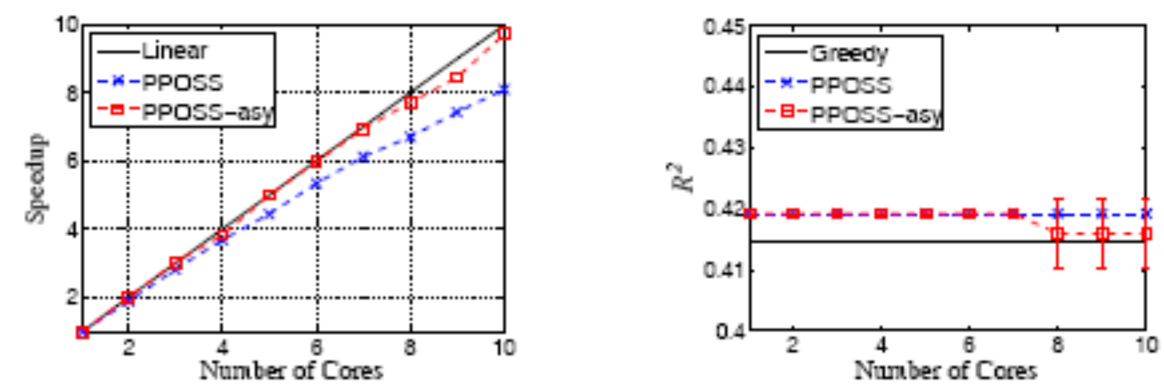
(c) on *clean1* (476 instances, 166 features)



(d) on *w5a* (9888 instances, 300 features)



(e) on *gisette* (7000 instances, 5000 features)



(f) on *farm-ads* (4143 instances, 54877 features)

References for Pareto optimization

- Yang Yu, Xin Yao, and Zhi-Hua Zhou. *On the approximation ability of evolutionary optimization with application to minimum set cover*. **Artificial Intelligence**, 2012, 180-181:20-33.

- Chao Qian, Yang Yu and Zhi-Hua Zhou. *Pareto ensemble pruning*. In: **Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)**, Austin, TX, 2015, pp.2935-2941.

- Chao Qian, Yang Yu and Zhi-Hua Zhou. *On constrained Boolean Pareto optimization*. In: **Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'15)**, Buenos Aires, Argentina, 2015, pp. 389-395.

- Chao Qian, Yang Yu and Zhi-Hua Zhou. *Subset selection by Pareto optimization*. In: **Advances in Neural Information Processing Systems 28 (NIPS'15)** , Montreal, Canada, 2015.

- Chao Qian, Jing-Cheng Shi, Yang Yu, Ke Tang and Zhi-Hua Zhou. *Parallel Pareto optimization for subset selection*. In: **Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI'16)**, New York, NY, 2016

Outline

Subset selection problem

and Pareto optimization

Local Lipschitz continuous problem

and classification-based optimization

Local Lipschitz continuous functions

binary space:

Given $f \in \mathcal{F}$, let x^* be a global minimum of f, for all $x \in X$, if $X = \{0,1\}^n$, then there exist positive constants $\beta_1, \beta_2, L_1, L_2$ such that

$$L_2\|x - x^*\|_H^{\beta_2} \leq f(x) - f(x^*) \leq L_1\|x - x^*\|_H^{\beta_1};$$

continuous space:

if X is a compact continuous domains, then there exist positive constants $\beta_1, \beta_2, L_1, L_2$ such that

$$L_2\|x - x^*\|_2^{\beta_2} \leq f(x) - f(x^*) \leq L_1\|x - x^*\|_2^{\beta_1}.$$

Local Lipschitz continuous functions

binary space:

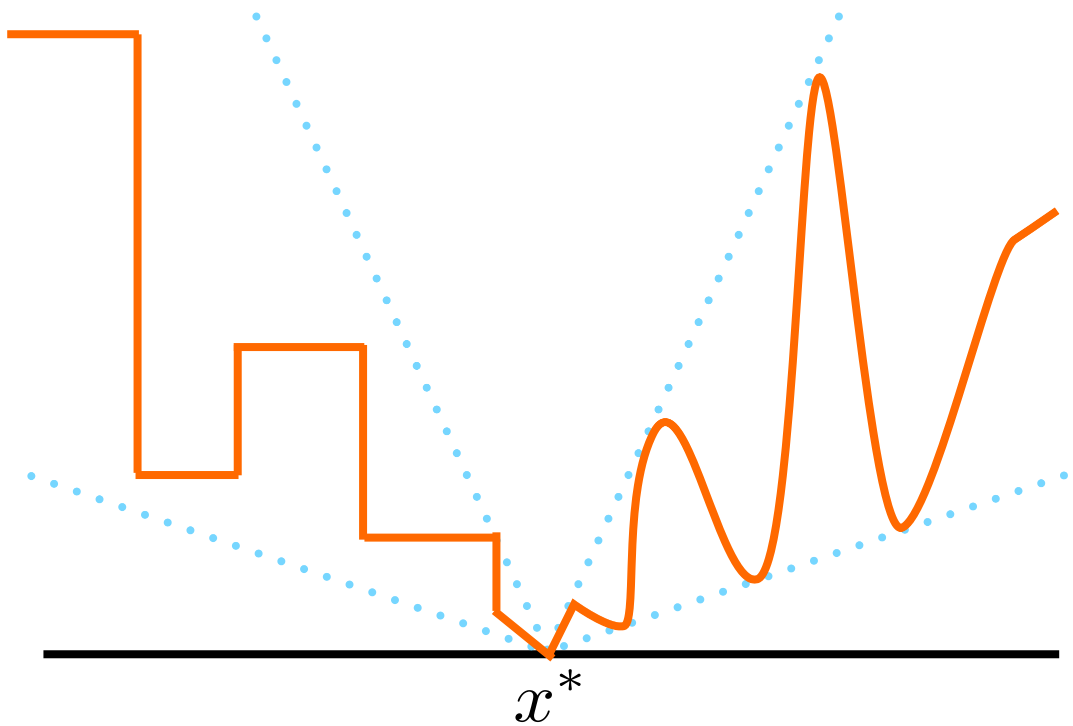
Given $f \in \mathcal{F}$, let x^* be a global minimum of f, for all $x \in X$, if $X = \{0,1\}^n$, then there exist positive constants $\beta_1, \beta_2, L_1, L_2$ such that

$$L_2\|x - x^*\|_H^{\beta_2} \leq f(x) - f(x^*) \leq L_1\|x - x^*\|_H^{\beta_1};$$

continuous space:

if X is a compact continuous domains, then there exist positive constants $\beta_1, \beta_2, L_1, L_2$ such that

$$L_2\|x - x^*\|_2^{\beta_2} \leq f(x) - f(x^*) \leq L_1\|x - x^*\|_2^{\beta_1}.$$

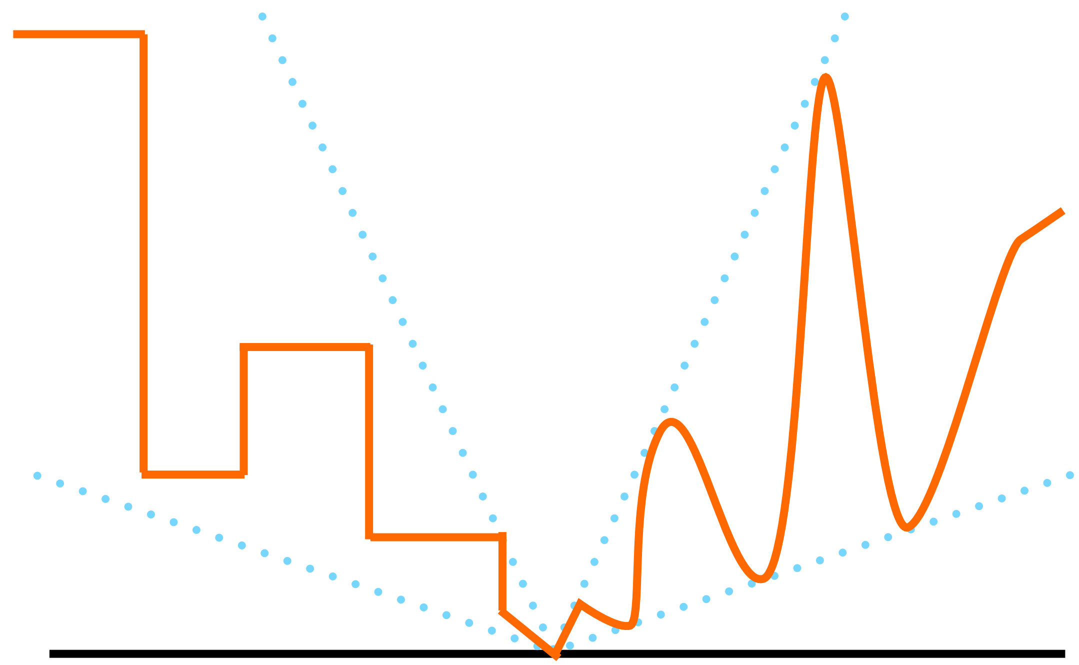


Local Lipschitz continuous functions

binary space:

Given $f \in \mathcal{F}$, let x^* be a global minimum of f, for all $x \in X$, if $X = \{0,1\}^n$, then there exist positive constants $\beta_1, \beta_2, L_1, L_2$ such that

$$L_2\|x - x^*\|_H^{\beta_2} \leq f(x) - f(x^*) \leq L_1\|x - x^*\|_H^{\beta_1};$$



continuous space:

if X is a compact continuous domains, then there exist positive constants $\beta_1, \beta_2, L_1, L_2$ such that

$$L_2\|x - x^*\|_2^{\beta_2} \leq f(x) - f(x^*) \leq L_1\|x - x^*\|_2^{\beta_1}.$$

A branch-and-bound method, optimistic optimization, can be proved to be efficient for this problem [Munos, Foundation and Trends in Machine Learning'14]

A general model-based optimization

Input:

$\epsilon > 0$: Approximation level
$T \in \mathbb{N}^+$: Number ofiterations
$m_0, \ldots, m_T \in \mathbb{N}^+$: Number of samples
$\lambda \in [0,1]$: Balancing parameters
$\mathcal{L}$: Learning algorithm
$\mathcal{T}$: Distribution transformation of hypothesis

Procedure:

1: Collect $S_0 = \{x_1, \ldots, x_{m_0}\}$ by i.i.d. sampling from the uniform distribution over$X$
2: $\tilde{x} = \mathrm{argmin}_{x \in S_0} f(x)$
3: Initialize the hypothesish_0
4: $T_0 = \emptyset$
5: for $t = 1$ to T do
6: Construct $T_t = \{(x_1, y_1), \ldots, (x_{m_{t-1}}, y_{m_{t-1}})\}$, where $x_i \in S_{t-1}$ and $y_i = f(x_i)$
7: $h_t = \mathcal{L}(T_t, T_{t-1}, h_{t-1}, t)$, the learning step
8: Initialize S_t from T_t
9: for $i = 1$ to m_t do
10: Sample x_i from $\begin{cases} \mathcal{T}_{h_t}, & \text{with probability } \lambda \\ \mathcal{U}_X, & \text{with probability } 1-\lambda \end{cases}$
11: $S_t = S_t \cup \{x_i\}$
12: end for
13: $\tilde{x} = \mathrm{argmin}_{x \in S_t \cup \{\tilde{x}\}} f(x)$
14: end for
15: return $\tilde{x}$

A general model-based optimization

Start with random solutions

Evaluate solutions

Learn a model

Sample new solutions:

from the model and

from the whole solutions space

with a balancing probability

Record the best-so-far solution

Return the best-so-far solution

Input:

$\epsilon > 0$: Approximation level
$T \in \mathbb{N}^+$: Number ofiterations
$m_0, \ldots, m_T \in \mathbb{N}^+$: Number of samples
$\lambda \in [0, 1]$: Balancing parameters
$\mathcal{L}$: Learning algorithm
$\mathcal{T}$: Distribution transformation of hypothesis

Procedure:

Collect $S_0 = \{x_1, \ldots, x_{m_0}\}$ by i.i.d. sampling from the uniform distribution over$X$
$\tilde{x} = \mathrm{argmin}_{x \in S_0} f(x)$
Initialize the hypothesish_0
$T_0 = \emptyset$
for $t = 1$ to T do
  Construct $T_t = \{(x_1, y_1), \ldots, (x_{m_{t-1}}, y_{m_{t-1}})\}$,
    where $x_i \in S_{t-1}$ and $y_i = f(x_i)$
  $h_t = \mathcal{L}(T_t, T_{t-1}, h_{t-1}, t)$, the learning step
  Initialize S_t from T_t
  for $i = 1$ to m_t do
    Sample x_i from $\begin{cases} \mathcal{T}_{h_t}, & \text{with probability } \lambda \\ \mathcal{U}_X, & \text{with probability } 1 - \lambda \end{cases}$
    $S_t = S_t \cup \{x_i\}$
  end for
  $\tilde{x} = \mathrm{argmin}_{x \in S_t \cup \{\tilde{x}\}} f(x)$
end for
return $\tilde{x}$

A general model-based optimization

Consider any functions F over compact solution spaces with bounded value range

Optimization performance measure

A general model-based optimization

Consider any functions F over compact solution spaces with bounded value range

Optimization performance measure

DEFINITION 1 ((ϵ,δ)-Query Complexity)
Given $f \in \mathcal{F}$, an algorithm $\mathcal{A}$, $0 < \delta < 1$ and $\epsilon > 0$, the (ϵ,δ)-query complexity is the number of calls to f such that, with probability at least $1-\delta$, $\mathcal{A}$ finds at least one solution $\tilde{x} \in X \subseteq \mathbb{R}^n$ satisfying

$$f(\tilde{x}) - f(x^*) \leq \epsilon,$$

where $f(x^*) = \min_{x \in X} f(x)$.

The number of evaluations until an (additive) approximate solution is found with a probability

Theoretical characterization

We can bound the query complexity:

Given $f \in \mathcal{F}$, $0 < \delta < 1$ and $\epsilon > 0$, the (ϵ,δ)-query complexity of a classification-based optimization algorithm is upper bounded by

$$O\left(\max\left\{\frac{1}{(1-\lambda)|D_\epsilon| + \lambda\overline{\mathbf{Pr}}_h} \ln\frac{1}{\delta}, \sum_{t=1}^{T} m_{\mathbf{Pr}_{h_t}}\right\}\right),$$

where $\overline{\mathbf{Pr}}_h = \frac{1}{T}\sum_{t=1}^{T}\mathbf{Pr}_{h_t}$

D_ϵ be the area of the target solutions

$\mathbf{Pr}_{h_t}$ be the success probability by sampling from the model at iteration t

$m_{\mathbf{Pr}_{h_t}}$ be the sample size required to have $\mathbf{Pr}_{h_t}$ success probability

Theoretical characterization

We can bound the query complexity:

Given $f \in \mathcal{F}$, $0 < \delta < 1$ and $\epsilon > 0$, the (ϵ,δ)-query complexity of a classification-based optimization algorithm is upper bounded by

$$O\left(\max\left\{\frac{1}{(1-\lambda)|D_\epsilon| + \lambda\overline{\mathbf{Pr}}_h}\ln\frac{1}{\delta}, \sum_{t=1}^{T} m_{\mathbf{Pr}_{h_t}}\right\}\right),$$

where $\overline{\mathbf{Pr}}_h = \frac{1}{T}\sum_{t=1}^{T}\mathbf{Pr}_{h_t}$ unknown due to unspecified model learning

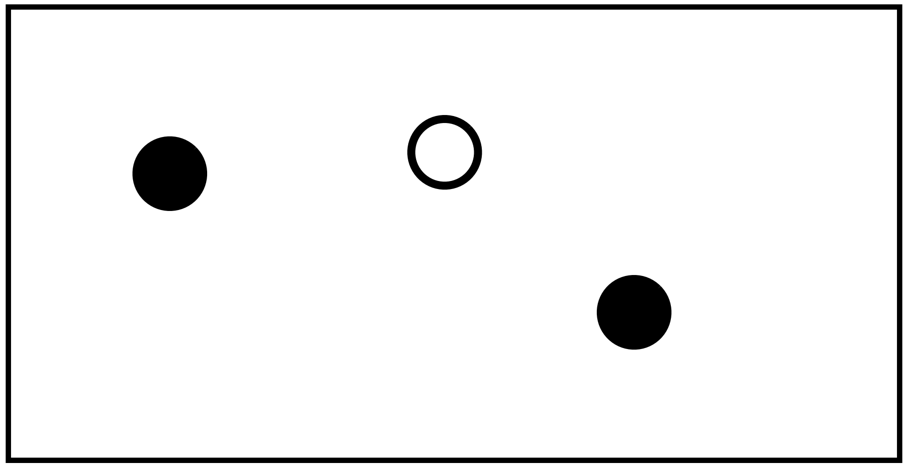
D_ϵ be the area of the target solutions

$\mathbf{Pr}_{h_t}$ be the success probability by sampling from the model at iteration t

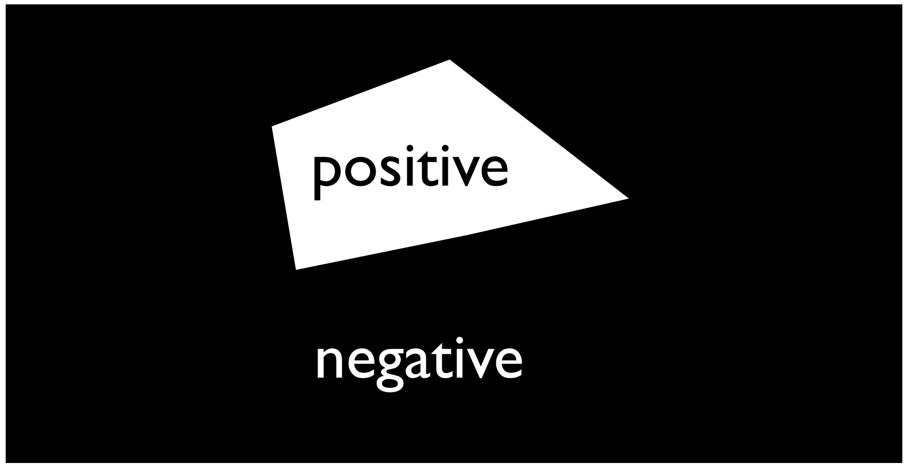
$m_{\mathbf{Pr}_{h_t}}$ be the sample size required to have $\mathbf{Pr}_{h_t}$ success probability

Classification model

from positive and negative examples



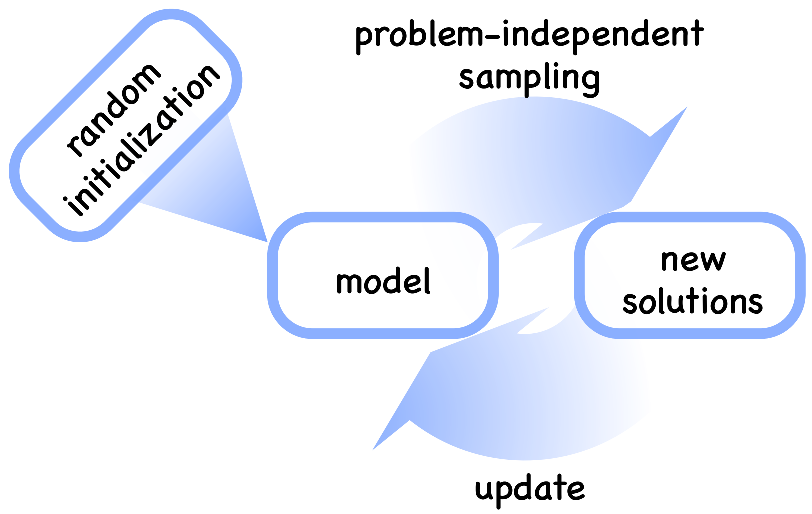
classify the space into two classes: {positive, negative}



with bounded generalization error

$$R_{\mathcal{D}_t} \leq \hat{R}_{\mathcal{D}_t} + \sqrt{\frac{8}{m}(d \log \frac{2em}{d} + \log \frac{4}{\eta})}$$

Classification-base optimization

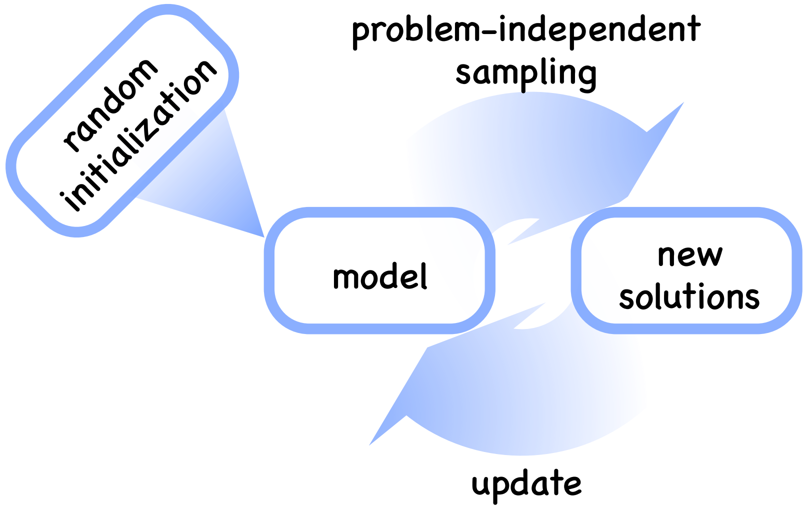


Model: classifier

Sampling: uniformly from positive area

Update: learn a new classifier

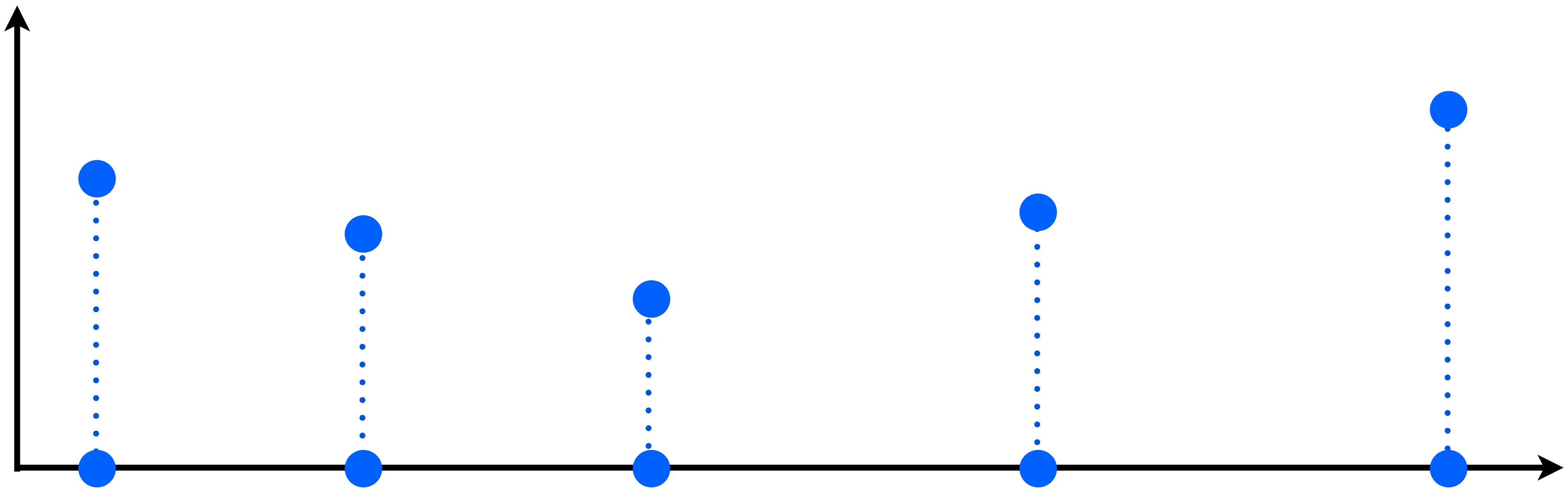
Classification-base optimization



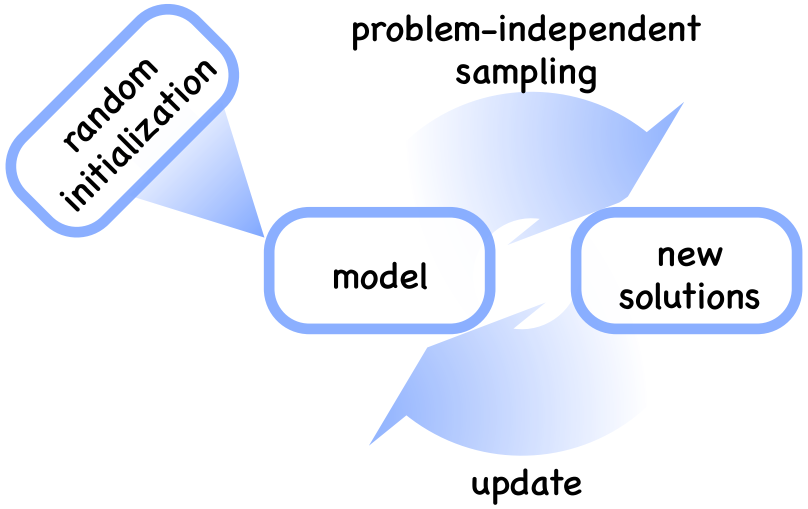
Model: classifier

Sampling: uniformly from positive area

Update: learn a new classifier



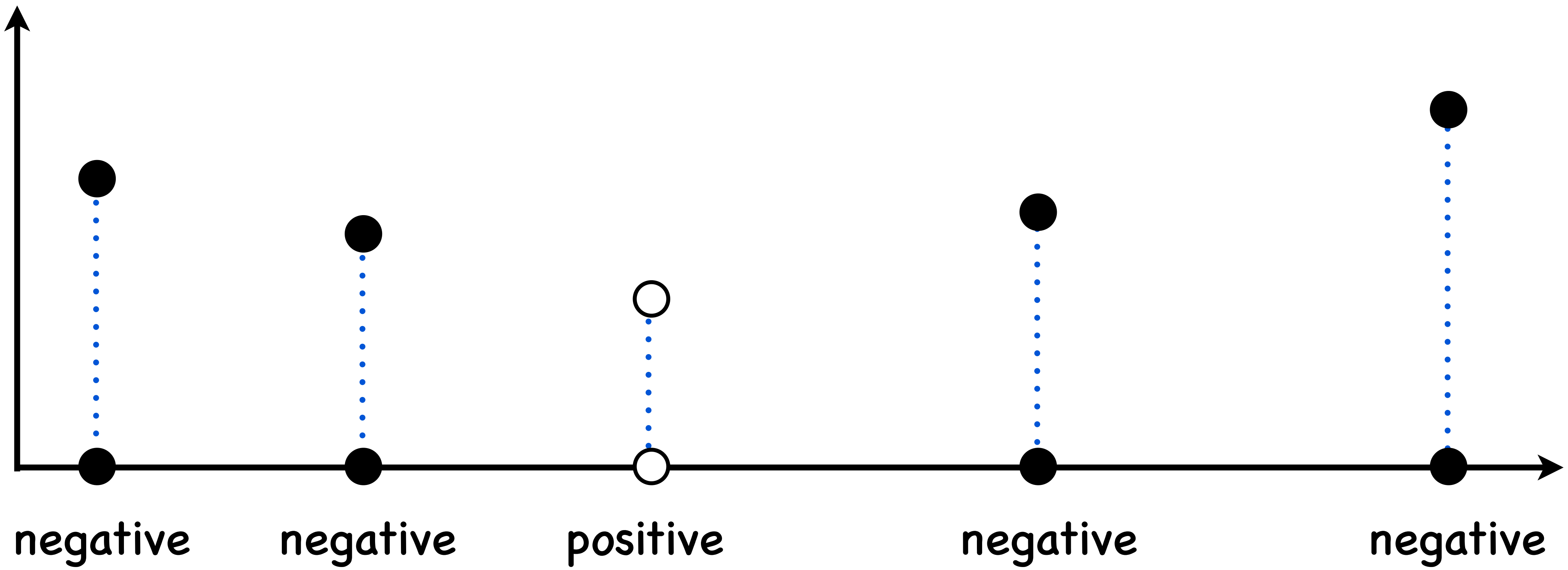
Classification-base optimization



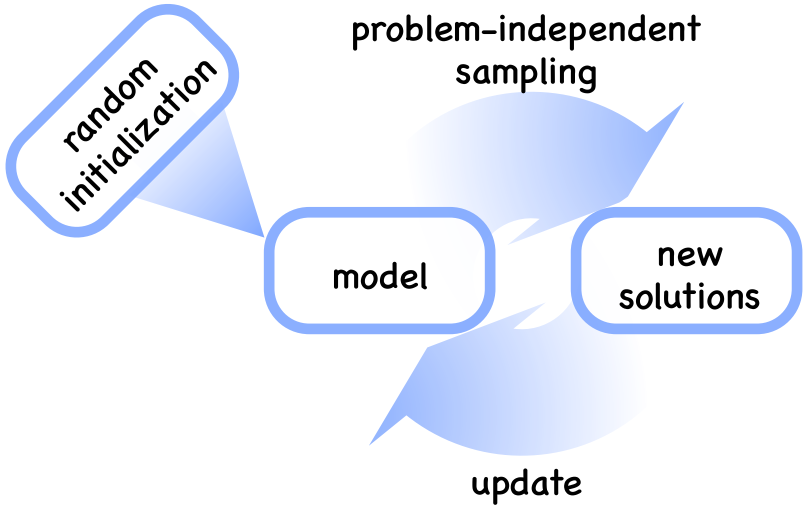
Model: classifier

Sampling: uniformly from positive area

Update: learn a new classifier



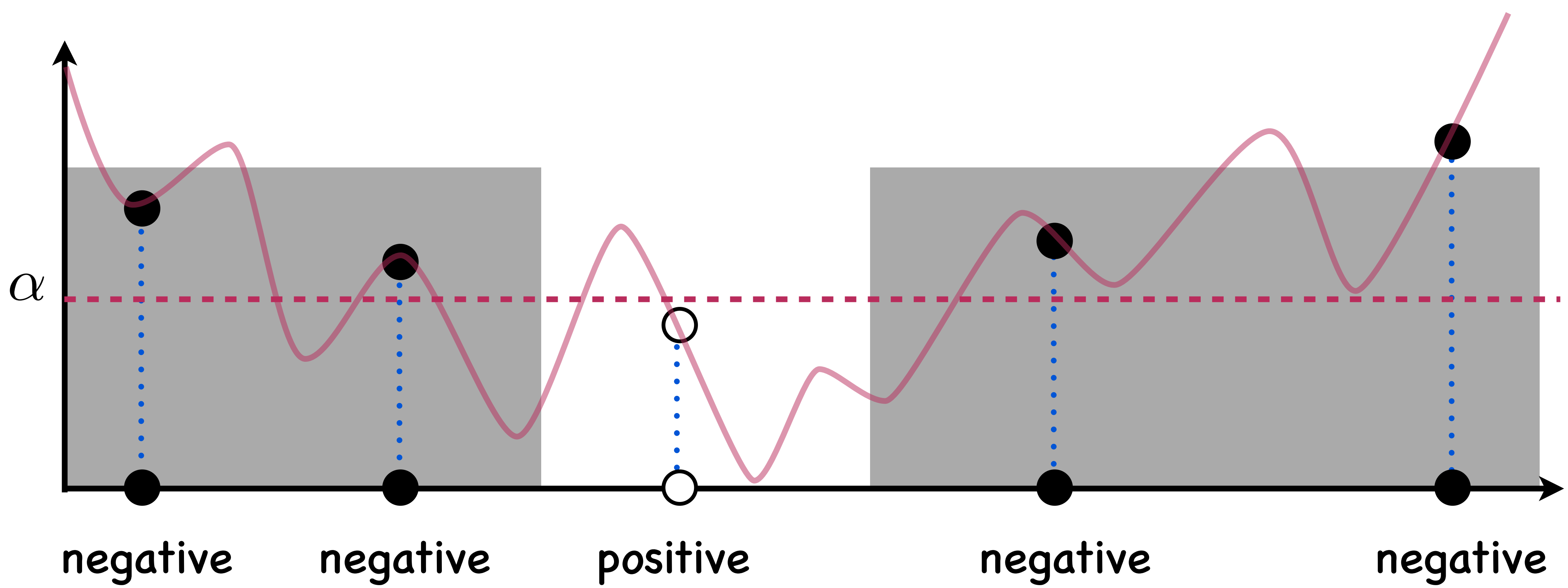
Classification-base optimization



Model: classifier : bounded error

Sampling: uniformly from positive area

Update: learn a new classifier



Classification-based optimization

Start with random solutions

Evaluate solutions

and prepare training data

Learn a classification model

Sample a new solution:

Record the best-so-far solution

Return the best-so-far solution

Algorithm 1 classification-based optimization

Input:

f: Objective function to be minimized;
$\mathcal{C}$: A binary classification algorithm;
$\lambda \in [0, 1]$: Balancing parameter;
$\alpha_1 > \ldots > \alpha_T$: Threshold for labeling;
$T \in \mathbb{N}^+$: Number of iterations;
$m \in \mathbb{N}^+$: Sampled size;
`Sampling`: Sampling subprocedure.

Procedure:

Collect $S_0 = \{x_1, \ldots, x_m\}$ by i.i.d. sampling from $\mathcal{U}_X$;
let $\tilde{x} = \mathrm{argmin}_{x \in S_0} f(x)$;
for $t = 1$ to T **do**
  Construct $B_t = \{(x_1, y_1), \ldots, (x_m, y_m)\}$,
  where $x_i \in S_{t-1}$ and $y_i = \mathrm{sign}[\alpha_t - f(x_i)]$
  Let $S_t = \emptyset$
  for $i = 1$ to m **do**
    $h_t = \mathcal{C}(B_t)$, where $h_t \in \mathcal{H}$
    $x_i = \mathtt{Sampling}(h_t, \lambda)$, and let $S_t = S_t \cup \{x_i\}$
  end for
  $\tilde{x} = \mathrm{argmin}_{x \in S_t \cup \{\tilde{x}\}} f(x)$
end for
return $\tilde{x}$ and $f(\tilde{x})$

Theorem

Theorem 1
Given $f \in \mathcal{F}$, $0 < \delta < 1$ and $\epsilon > 0$, if a classification-based optimization algorithm has error-target θ-dependence and γ-shrinking rate, its (ϵ,δ)-query complexity is upper bounded

$$O\left(\frac{1}{|D_\epsilon|}\left((1-\lambda)+\frac{\lambda}{\gamma T}\sum_{t=1}^{T}\frac{1-Q\cdot R_{\mathcal{D}_t}-\theta}{|D_{\alpha_t}|}\right)^{-1}\ln\frac{1}{\delta}\right),$$

where $Q = 1/(1-\lambda)$.

Theorem

THEOREM 1
Given $f \in \mathcal{F}$, $0 < \delta < 1$ and $\epsilon > 0$, if a classification-based optimization algorithm has error-target θ-dependence and γ-shrinking rate, its (ϵ,δ)-query complexity is upper bounded

$$O\left(\frac{1}{|D_\epsilon|}\left((1-\lambda)+\frac{\lambda}{\gamma T}\sum_{t=1}^{T}\frac{1-Q\cdot R_{\mathcal{D}_t}-\theta}{|D_{\alpha_t}|}\right)^{-1}\ln\frac{1}{\delta}\right),$$

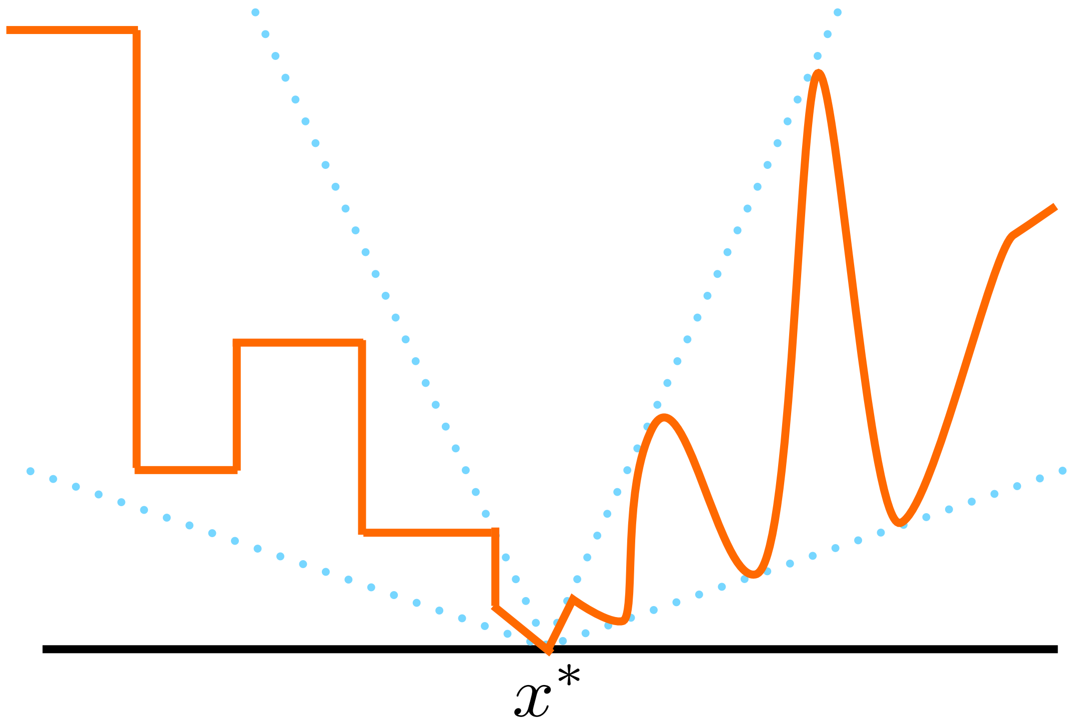
where $Q = 1/(1-\lambda)$.

smaller θ the better: the classifier should be highly randomized

smaller γ the better: the learnt positive area should be small

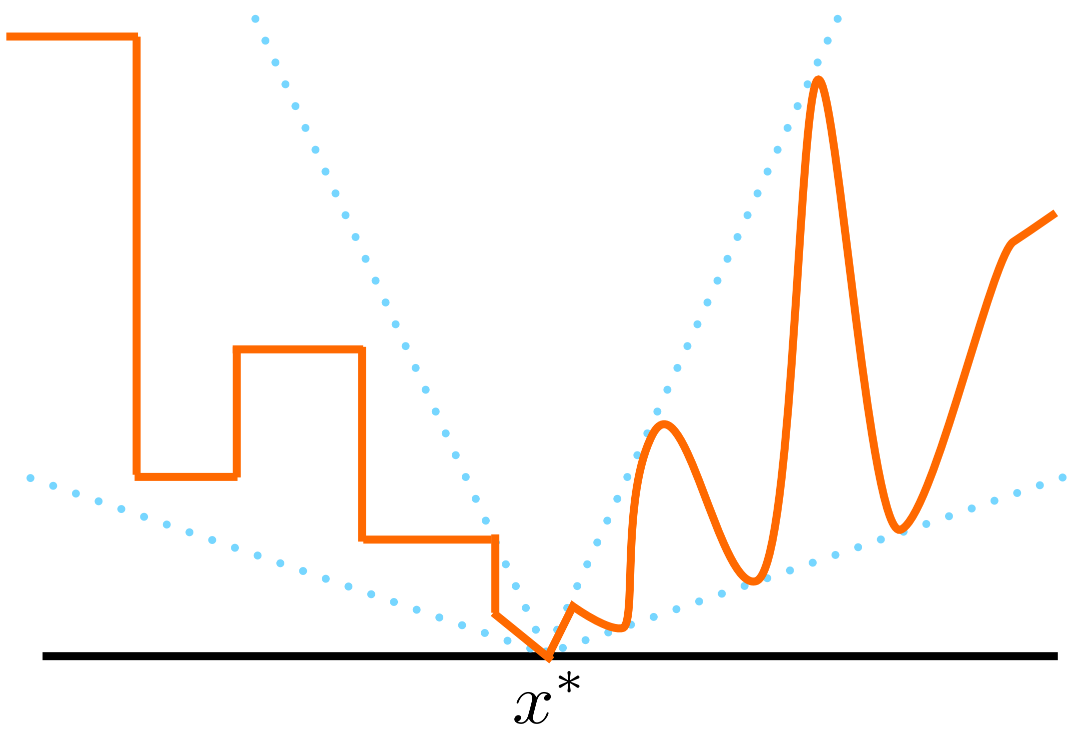
Corollaries

On local Lipschitz continuous functions



Corollaries

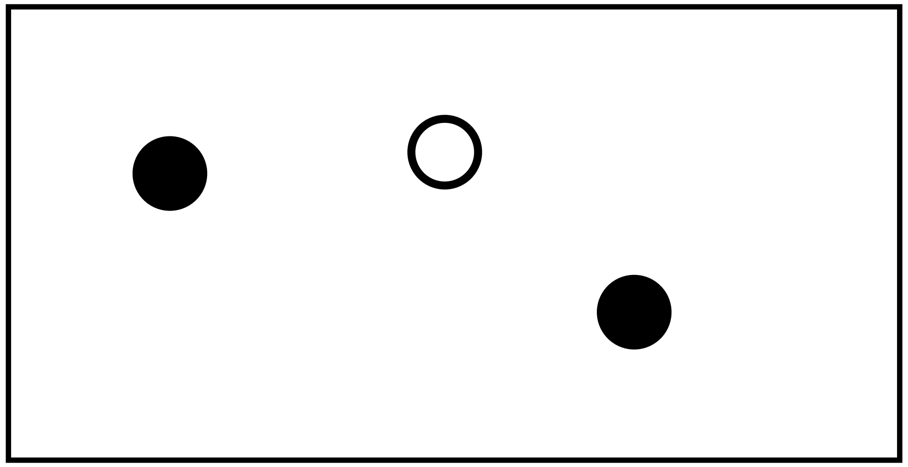
On local Lipschitz continuous functions



COROLLARY 2
In compact continuous domains X, given $f \in \mathcal{F}_L^{\beta_1, L_1, \beta_2, L_2}$, $0 < \delta < 1$ and $\epsilon > 0$, for a classification-based optimization algorithm using a classification algorithm with convergence rate $\widetilde{\Theta}(\frac{1}{m})$, under the conditions that error-target dependence $\theta < 1$ and shrinking rate $\gamma > 0$, its (ϵ,δ)-query complexity belongs to $\texttt{poly}(\frac{1}{\epsilon}, n, \frac{1}{\beta_1}, \beta_2, \ln L_1, \ln \frac{1}{L_2}) \cdot \ln \frac{1}{\delta}$.

classification-based optimization is efficient for local Lipschitz functions

Classification model design



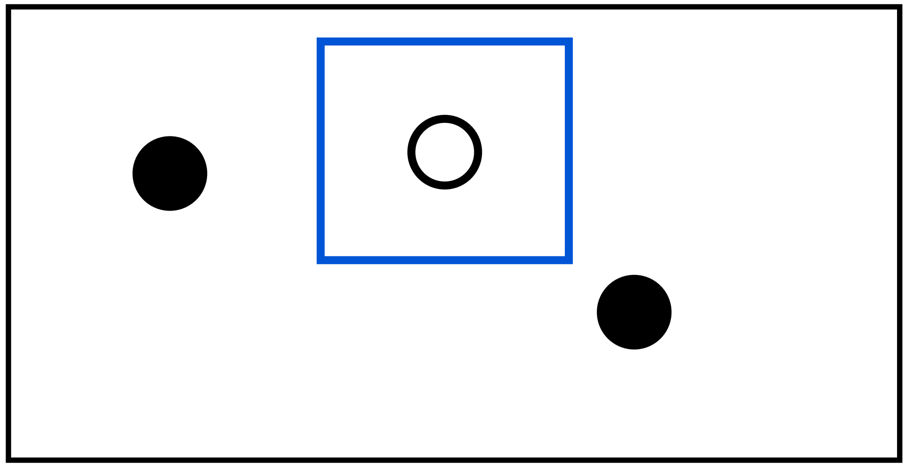
Classification model design

Considerations

1. a classifier with a
   samplable positive area

Implementation:

learn an axis-parallel region



Classification model design

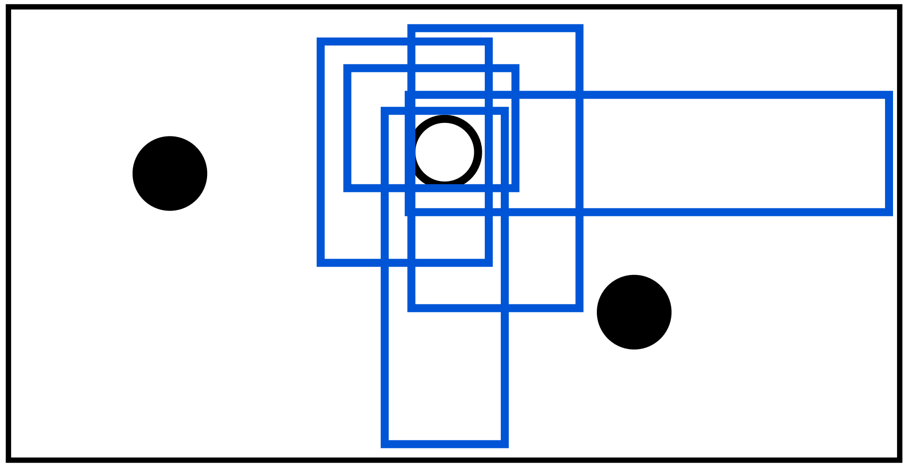
Considerations:

1. a classifier with a samplable positive area
2. smaller θ -> less dependent

Implementation:

learn an axis-parallel region

with randomness



Classification model design

Considerations:

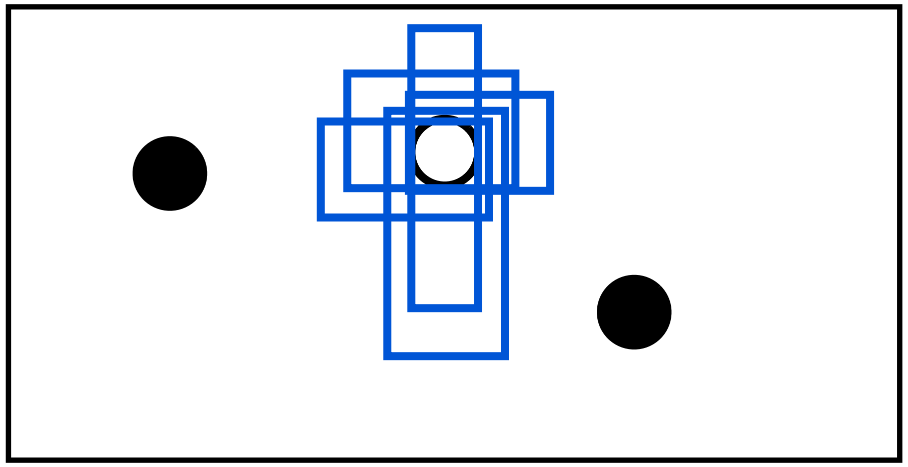
1. a classifier with a samplable positive area
2. smaller θ -> less dependent
3. smaller γ -> small positive area

Implementation:

learn an axis-parallel region

with randomness

as small as possible



randomized coordinate shrinking classification (RACOS) cation algorithm for $X = \{0,1\}^n$ or $[0,1]^n$

Classification model design

randomized coordinate shrinking classification algorithm (RACOS)

Input:

t: Current iteration number;
B_t: Solution set in iteration t;
X: Solution space ($\{0,1\}^n$ or $[0,1]^n$);
I: Index set of coordinates;
$M \in \mathbb{N}^+$: Maximum number of uncertain coordinates.

Procedure:

$B_t^+ =$ the positive solutions in B_t
$B_t^- = B_t - B_t^+$
Randomly select $x_+ = (x_+^{(1)}, \ldots, x_+^{(n)})$ from B_t^+
Let $D_{h_t} = X$, $I = \{1, \ldots, n\}$
while $\exists x \in B_t^-$ s.t. $h_t(x) = +1$ **do**
  if $X = \{0,1\}^n$ **then**
    $k =$ randomly selected index from the index set I
    $D_{h_t} = D_{h_t} - \{x \in X \mid x^{(k)} \neq x_+^{(k)}\}$, $I = I - \{k\}$
  end if
  if $X = [0,1]^n$ **then**
    $k =$ randomly selected index from the index set I
    $x^- =$ randomly selected solution from B_t^-
    if $x_+^{(k)} \geq x_-^{(k)}$ **then**
      $r =$ uniformly sampled value in $(x_-^{(k)}, x_+^{(k)})$
      $D_{h_t} = D_{h_t} - \{x \in X \mid x^{(k)} < r\}$
    else
      $r =$ uniformly sampled value in $(x_+^{(k)}, x_-^{(k)})$
      $D_{h_t} = D_{h_t} - \{x \in X \mid x^{(k)} > r\}$
    end if
  end if
end while
while $\#I > M$ **do**
  $k =$ randomly selected index from the index set I
  $D_{h_t} = D_{h_t} - \{x \in X \mid x^{(k)} \neq x_+^{(k)}\}$, $I = I - \{k\}$
end while
return h_t

for discrete domain

learn an axis-parallel region with randomness

for continuous domain

as small as possible

Experiments

RACOS: a classification-based optimization algorithm

SOO: a branch-and-bound algorithm
REMBO: a Bayesian optimization algorithm
CMAES: an evolutionary algorithm

test cases
- clustering tasks
- classification tasks

On clustering

clustering a dataset $\mathcal{V} = \{v_1, \ldots, v_n\}$

similarity between two instances $W_{p,q} = \exp\left(-\|v_p - v_q\|_2^2 / \sigma^2\right)$

normalized min-cut $f(A_1, A_2) = \sum_{i}^{2} \frac{1}{\#A_i} \sum_{p \in A_i, q \notin A_i} W_{p,q}$

(NP-hard)

solution: binary vector representing the bipartition

On clustering

results with 30n evaluations

repeat 30 times independently

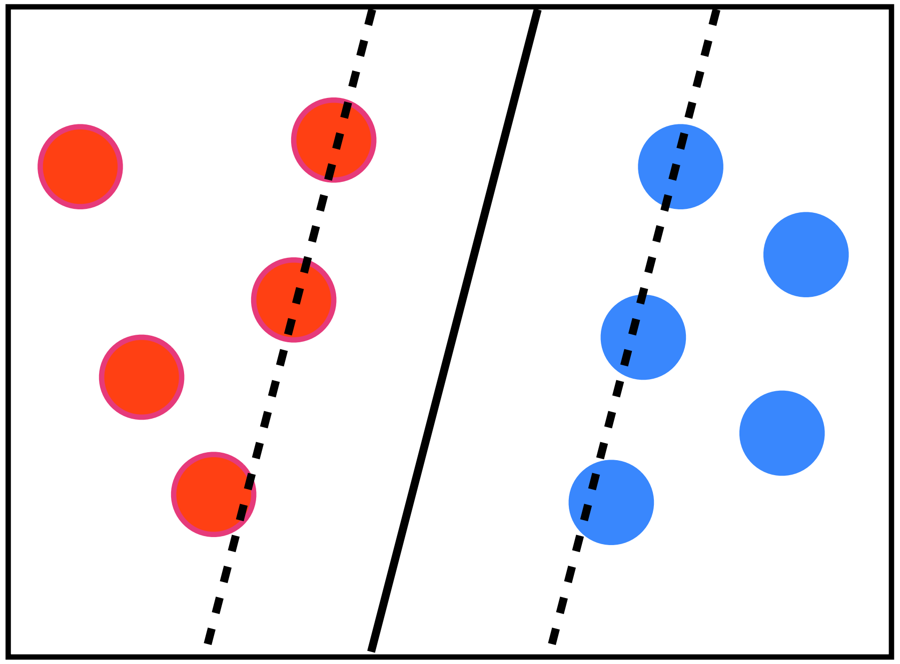
t-test with confidence level 5%

data sets: Sonar, Heart, Ionosphere, Breast Cancer, German

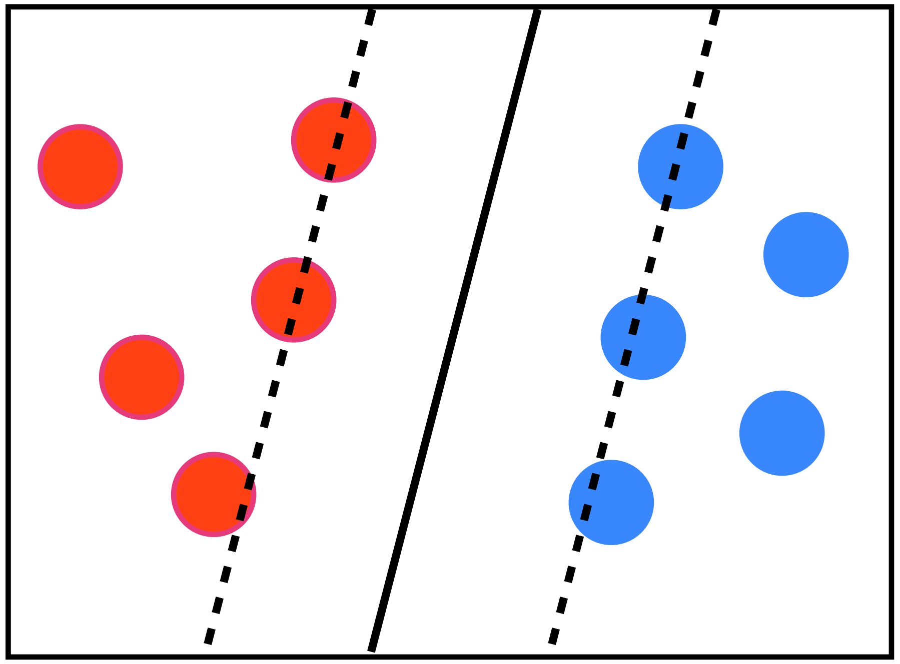
instances: 208, 270, 351, 683, 1000

Algorithm	*Sonar*	*Heart*	*Ionosphere*	*Breast Cancer*	*German*	w/t/l to RACOS
USC	3.91±0.00•	79.67±0.00•	54.21±0.00•	200.62±0.00•	239.00±0.00•	0 / 0 / 5
GA	3.14±0.74	**57.31**±0.46	55.71±3.74•	189.52±1.26	205.61±1.80•	0 / 3 / 2
RLS	4.07±0.82•	58.81±0.45•	58.74±2.81•	192.63±1.62•	207.36±2.11•	0 / 0 / 5
UMDA	7.40±2.26•	58.76±1.02•	61.77±4.54•	193.58±3.56•	212.83±1.08•	0 / 0 / 5
CE	8.00±1.35•	58.75±1.39•	63.71±3.41•	188.76±3.77	209.57±1.96•	0 / 1 / 4
RACOS	**2.88**±0.63	57.45±0.89	**50.01**±2.80	**187.55**±3.01	**192.11**±2.51	- / - / -

On classification with Ramp loss

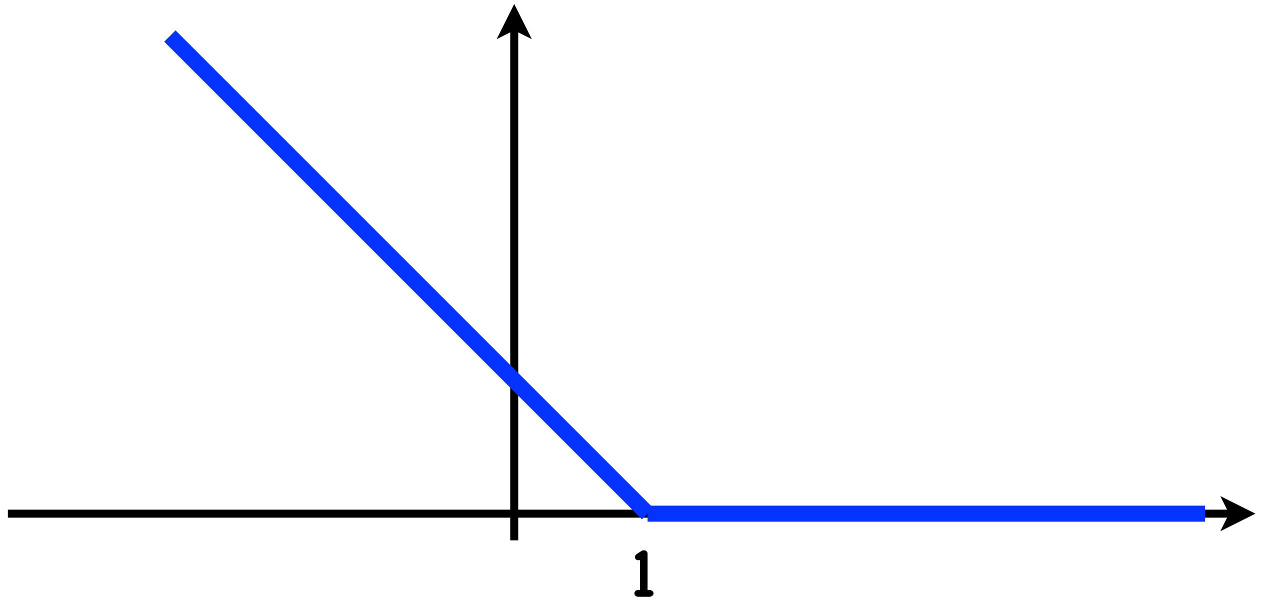


On classification with Ramp loss

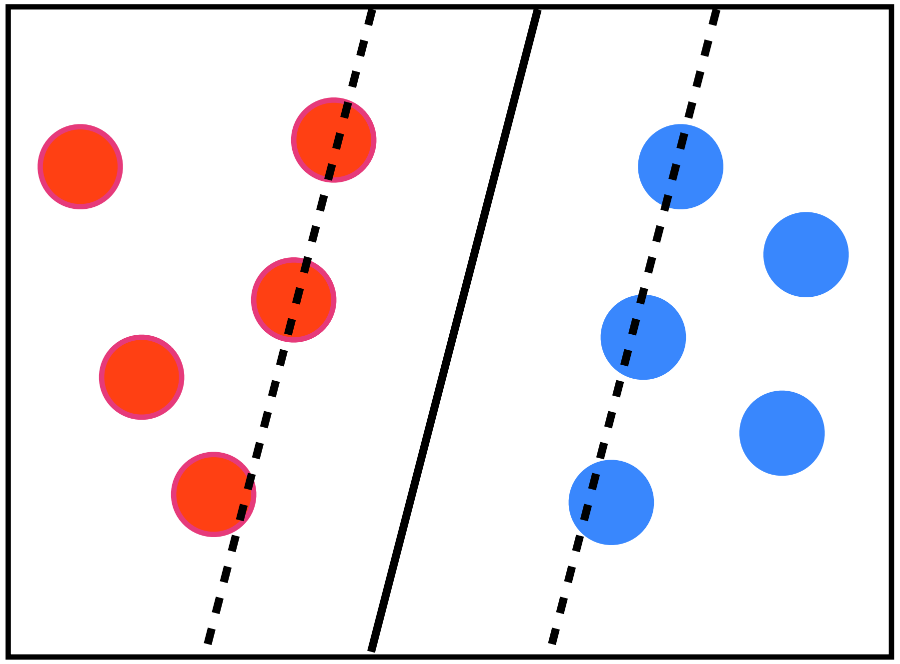


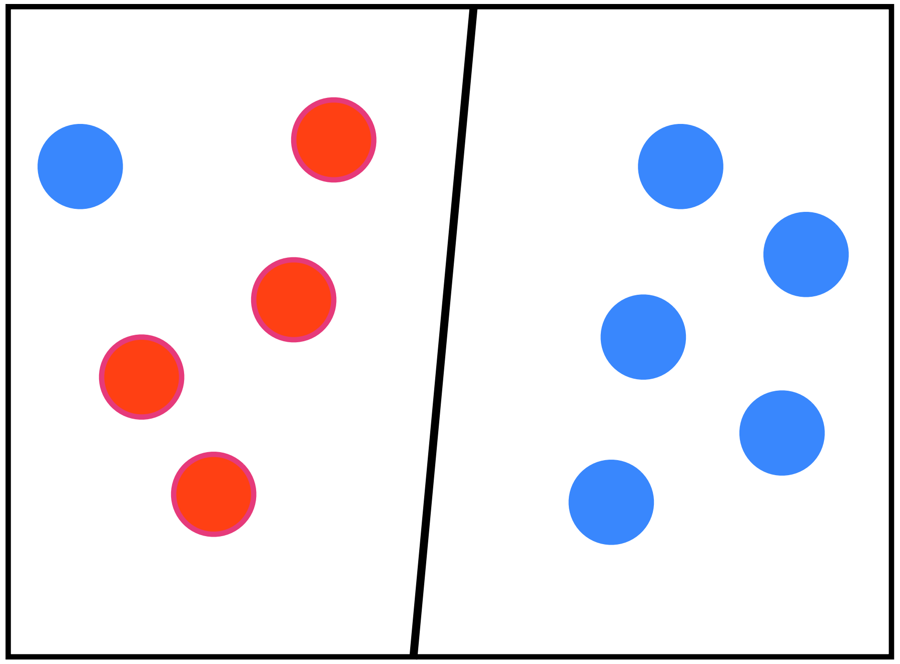
the loss function for linear SVM

$$f(w,b) = \frac{1}{2}\|w\|_2^2 + C\sum_{\ell}^{L} \max\{0,\ 1 - y_\ell(w^\top v_\ell + b)\}$$



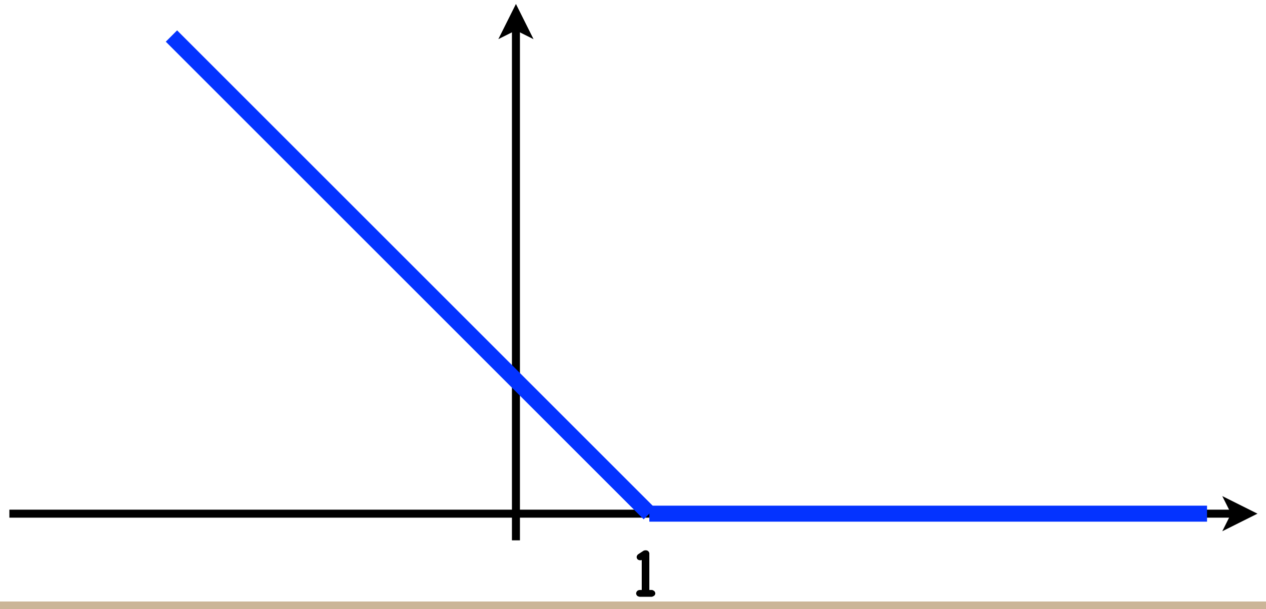
On classification with Ramp loss





the loss function for linear SVM

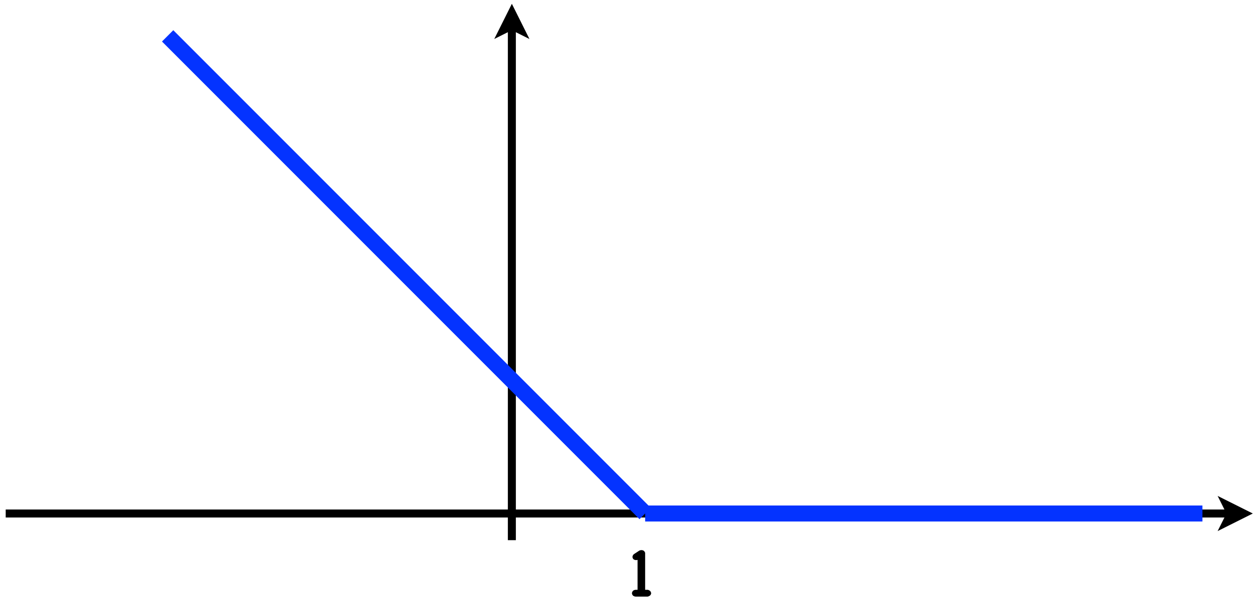
$$f(w,b) = \frac{1}{2}\|w\|_2^2 + C\sum_{\ell}^{L} \max\{0,\ 1 - y_\ell(w^\top v_\ell + b)\}$$



On classification with Ramp loss

the loss function for linear SVM

$$f(w,b) = \frac{1}{2}\|w\|_2^2 + C\sum_{\ell}^{L}\max\{0,\ 1 - y_\ell(w^\top v_\ell + b)\}$$



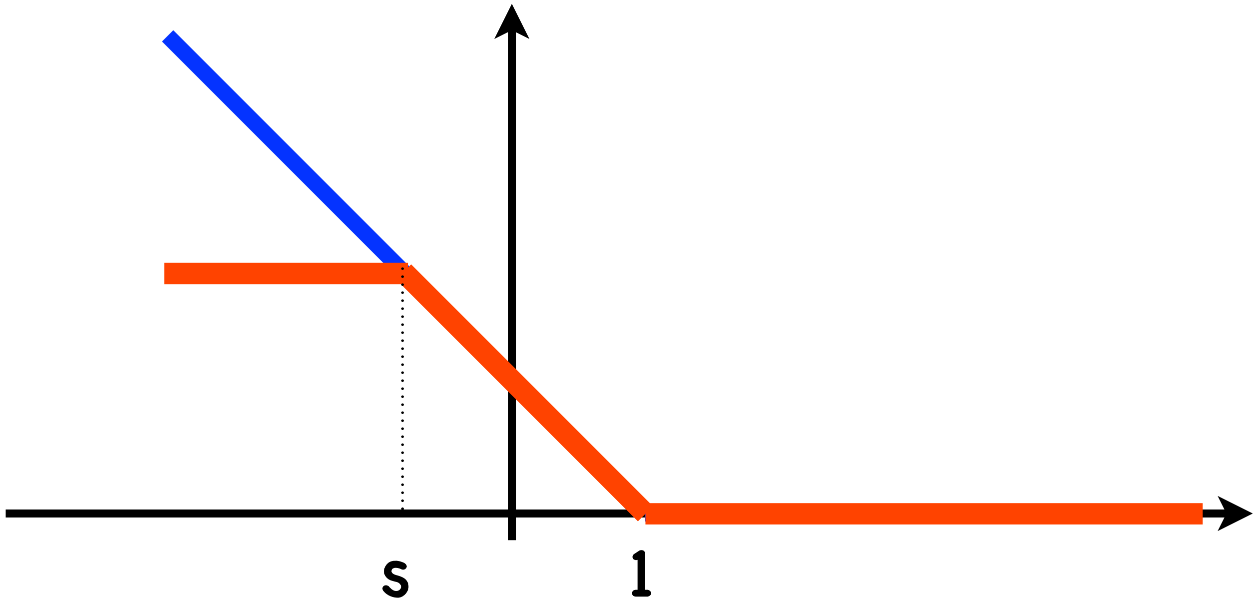
On classification with Ramp loss

the loss function for linear SVM

$$f(w,b) = \frac{1}{2}\|w\|_2^2 + C\sum_{\ell}^{L} \max\{0,\ 1 - y_\ell(w^\top v_\ell + b)\}$$

the loss function using Ramp loss

$$f(w,b) = \frac{1}{2}\|w\|_2^2$$
$$+ C\sum_{\ell}^{L}\Big(\max\{0,\ 1 - y_\ell(w^\top v_\ell + b)\} - \max\{0,\ s - y_\ell(w^\top v_\ell + b)\}\Big)$$



On classification with Ramp loss

the loss function for linear SVM

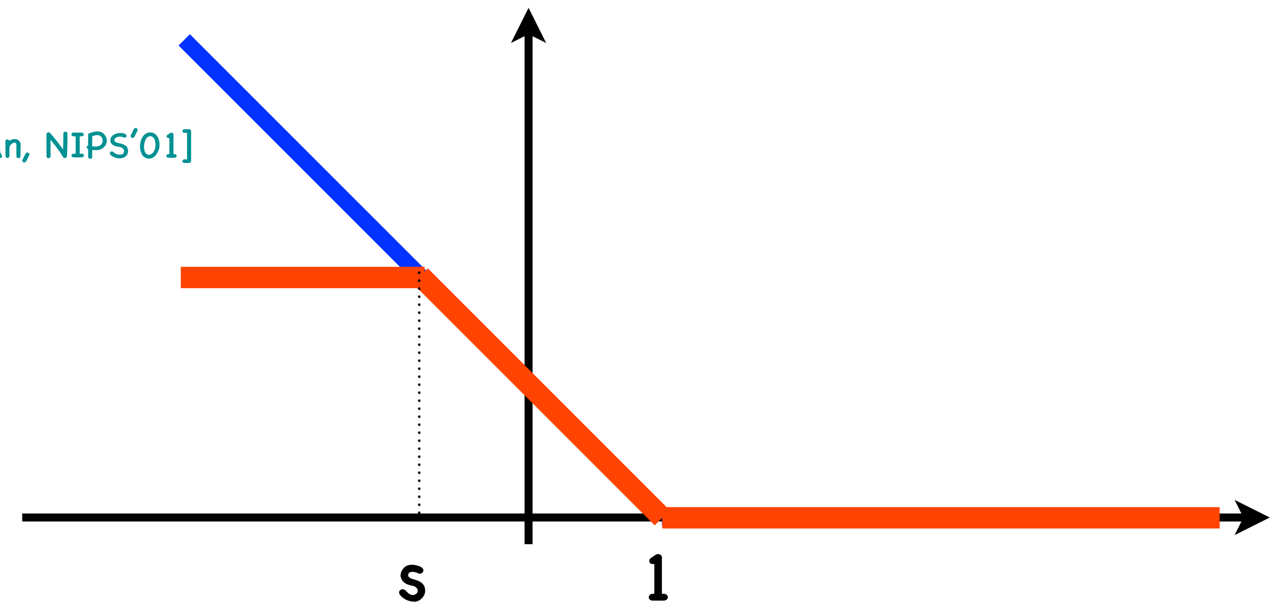
$$f(w,b) = \frac{1}{2}\|w\|_2^2 + C\sum_{\ell}^{L} \max\{0,\ 1 - y_\ell(w^\top v_\ell + b)\}$$

the loss function using Ramp loss

$$f(w,b) = \frac{1}{2}\|w\|_2^2$$
$$+ C\sum_{\ell}^{L}\Big(\max\{0,\ 1 - y_\ell(w^\top v_\ell + b)\} - \max\{0,\ s - y_\ell(w^\top v_\ell + b)\}\Big)$$

previous solution: CCCP [Yuille and Rangarajan, NIPS'01]

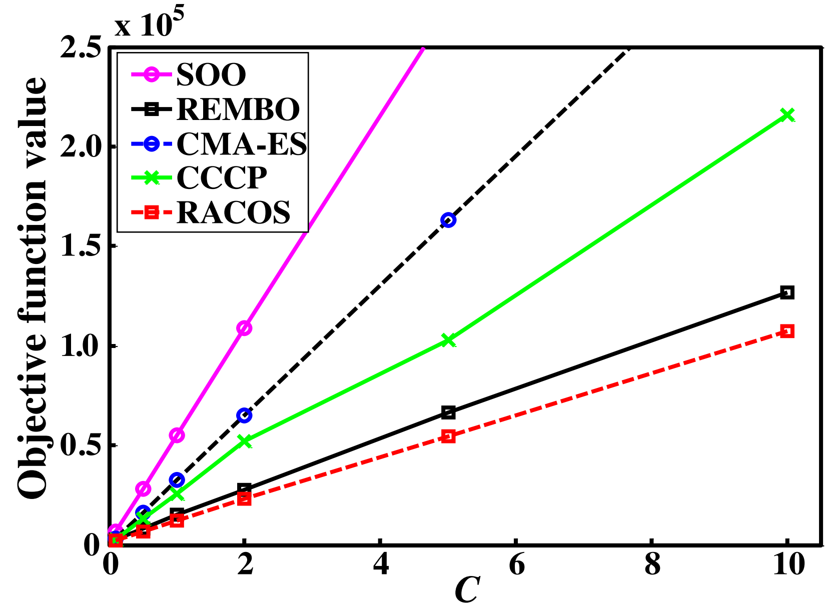
- relax the concave part to be linear
- gradient decent



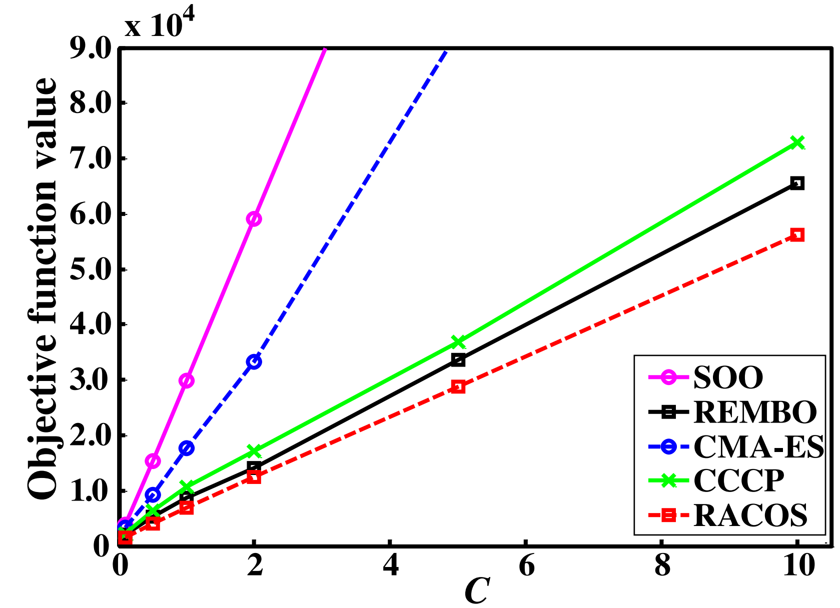
On classification with Ramp loss

with 40n evaluations

n=124

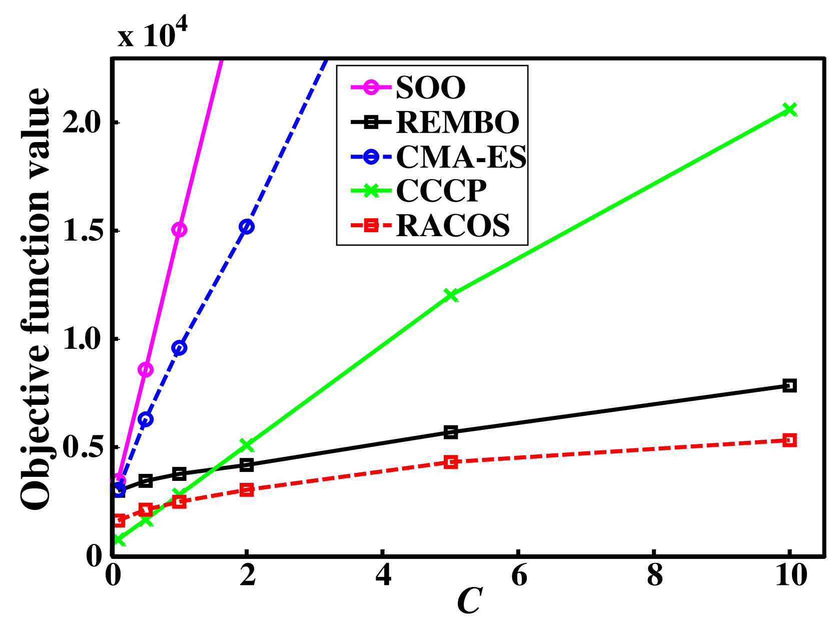


(a) on *Adult*, $s = -1$

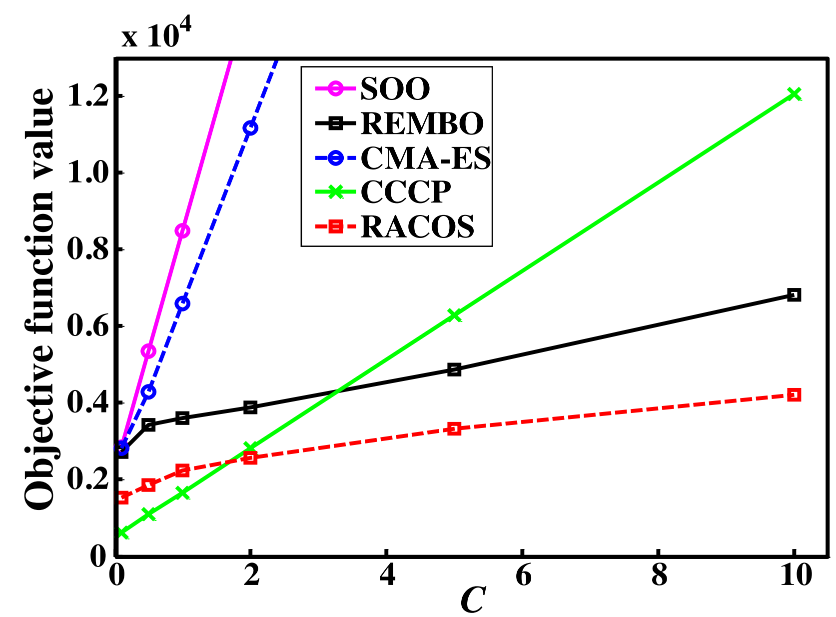


(b) on *Adult*, $s = 0$

n=257

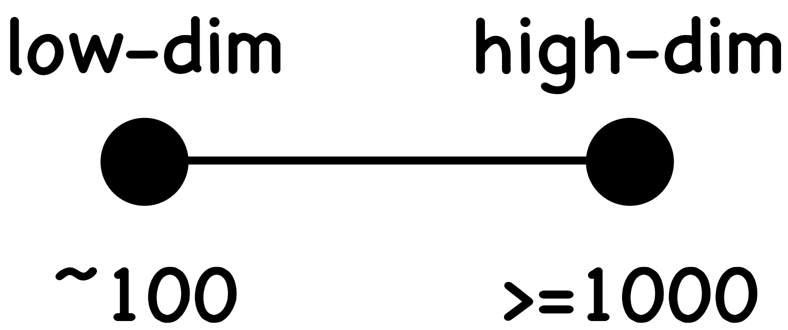


(c) on *USPS+N*, $s = -1$



(d) on *USPS+N*, $s = 0$

Extension 1: High-dimensional optimization



Extension 1: High-dimensional optimization

derivative-free optimization methods are hard to scale:

- too slow to calculate in high-dimensions
- too slow to converge in high-dimensions

low-dim high-dim

~100 >=1000

Extension 1: High-dimensional optimization

derivative-free optimization methods are hard to scale:

- too slow to calculate in high-dimensions
- too slow to converge in high-dimensions

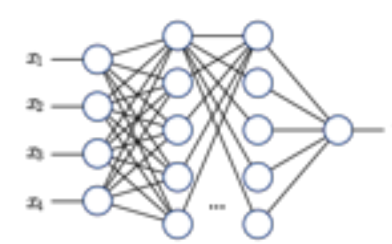
low-dim	high-dim	high-dim in ML
~100	>=1000	10 billion

Extension 1: High-dimensional optimization

derivative-free optimization methods are hard to scale:

- too slow to calculate in high-dimensions
- too slow to converge in high-dimensions

low-dim	high-dim	single machine	high-dim in ML
~100	>=1000	>1 million linear model	10 billion

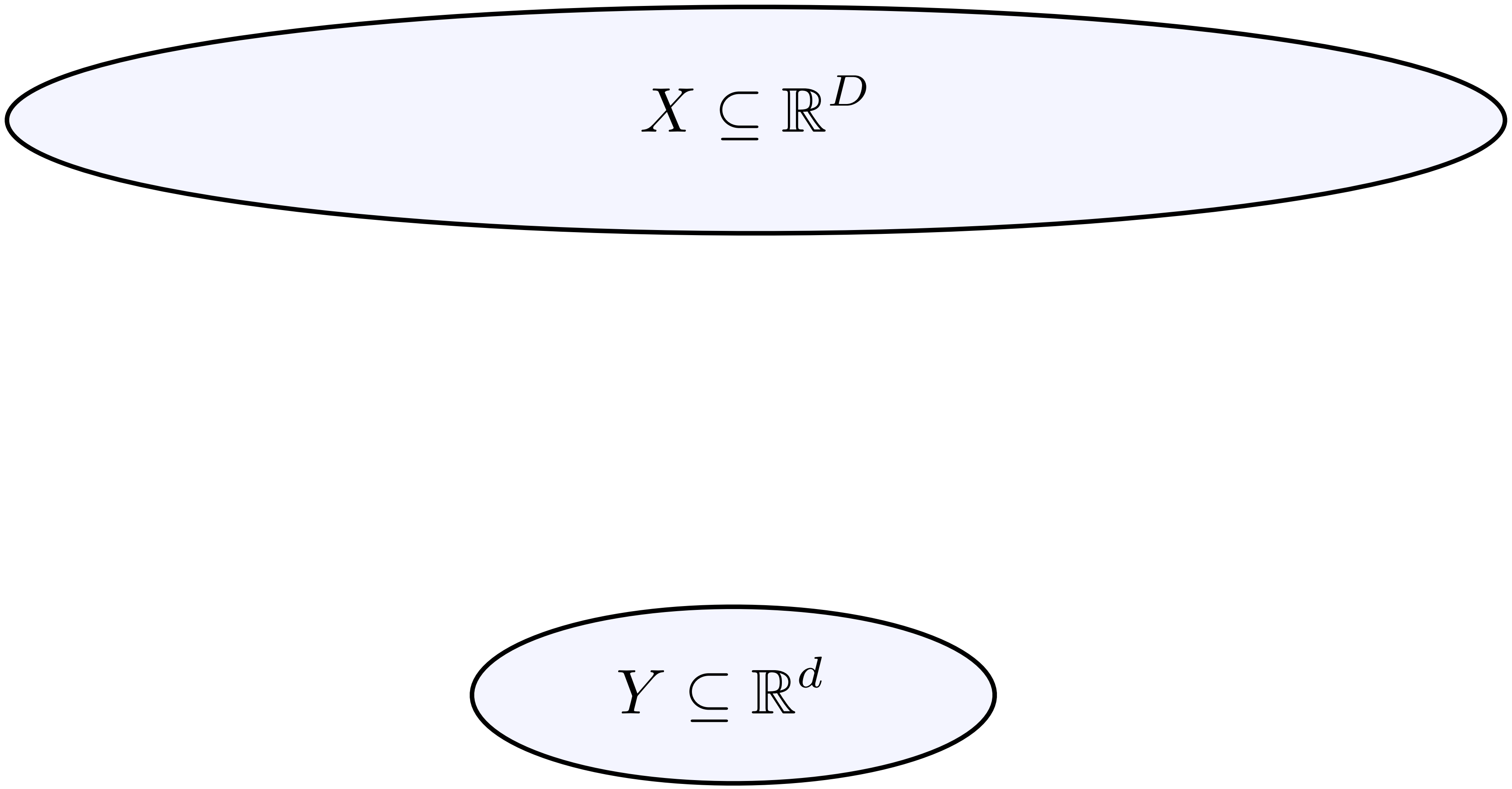


Random embedding

$$X \subseteq \mathbb{R}^D$$

$$\min_{x \in X} f(x)$$

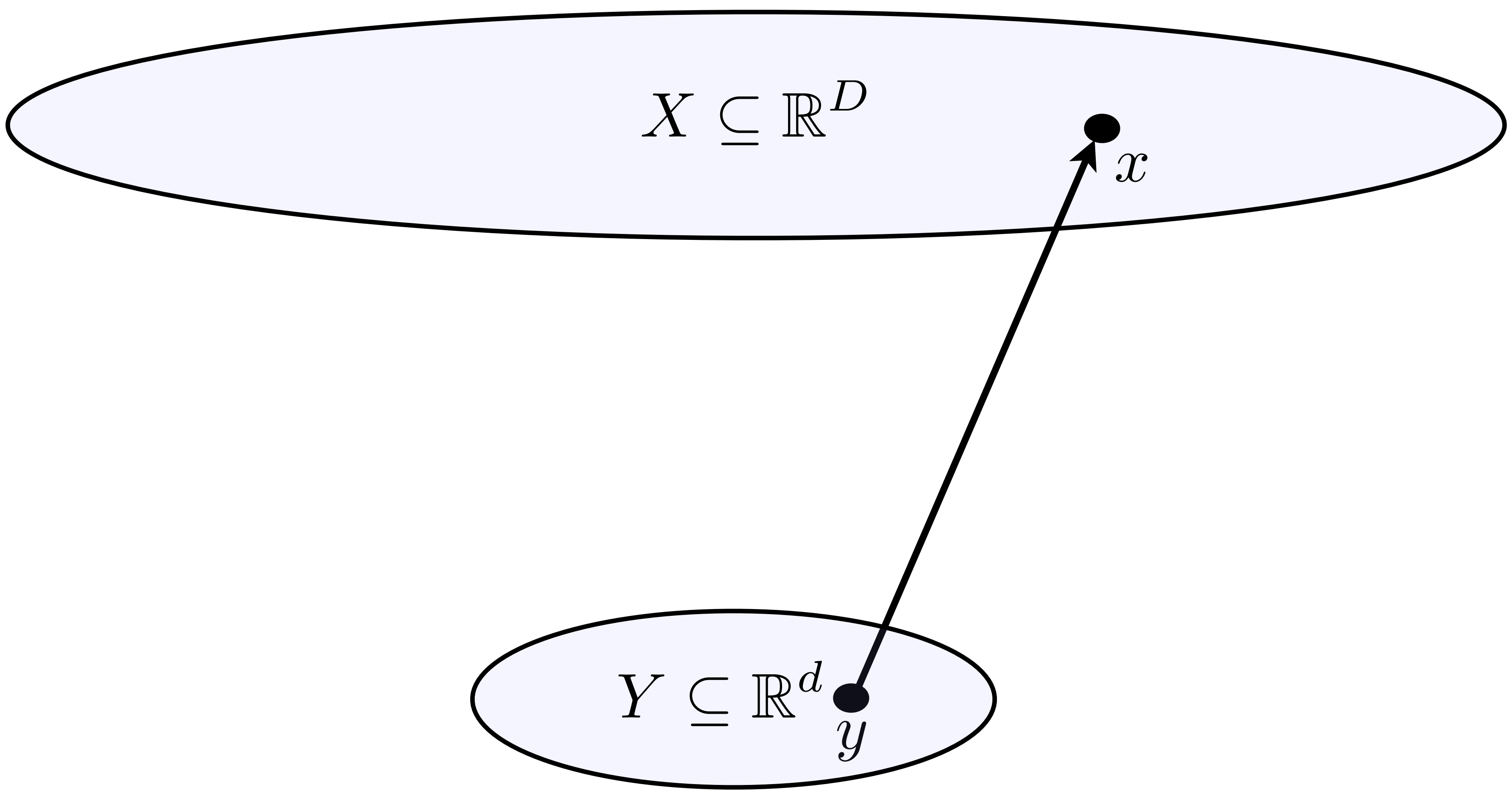
Random embedding



$$\min_{x \in X} f(x)$$

separate the search space and the evaluation space

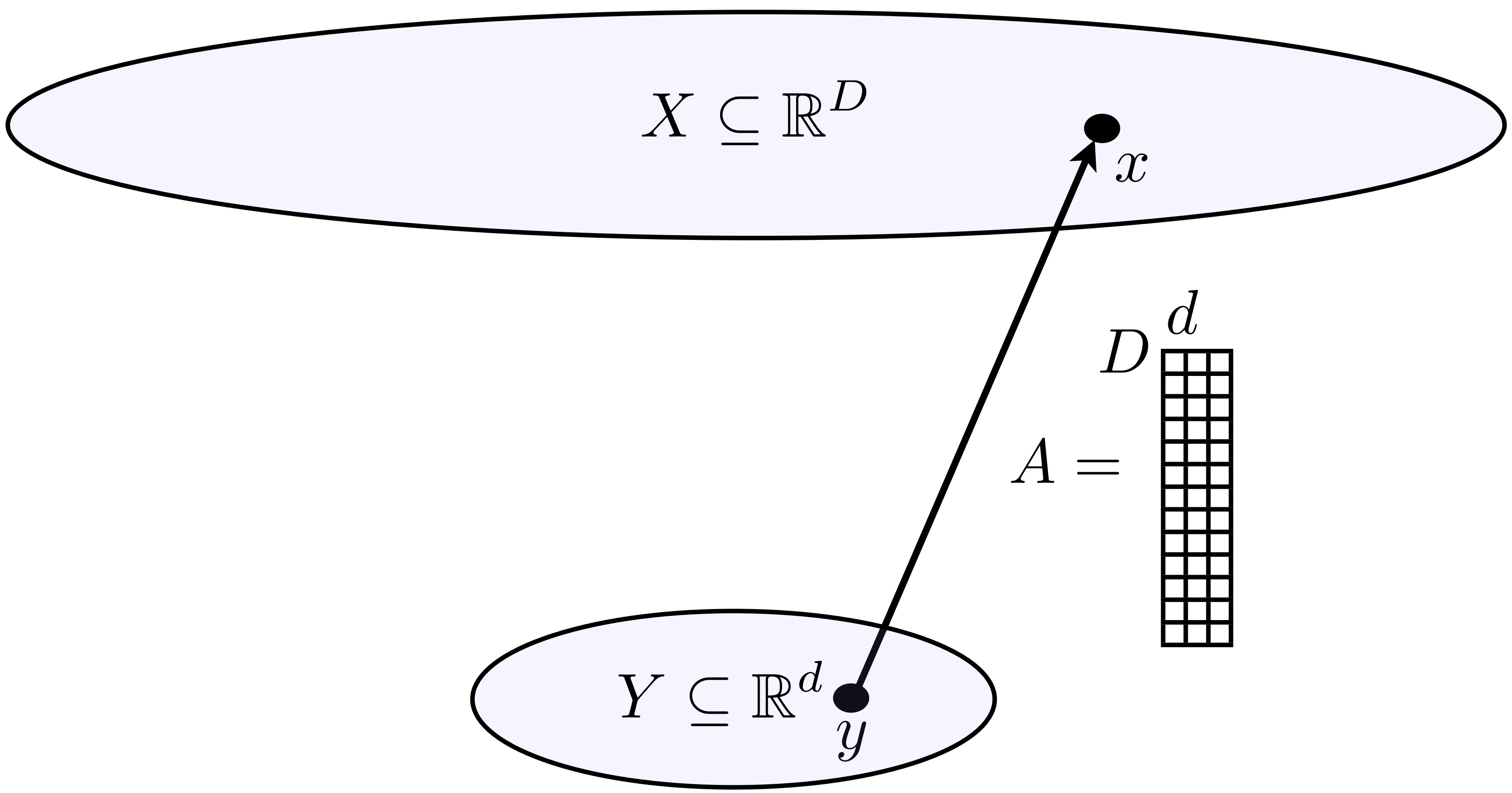
Random embedding



$$\min_{x \in X} f(x)$$

separate the search space and the evaluation space

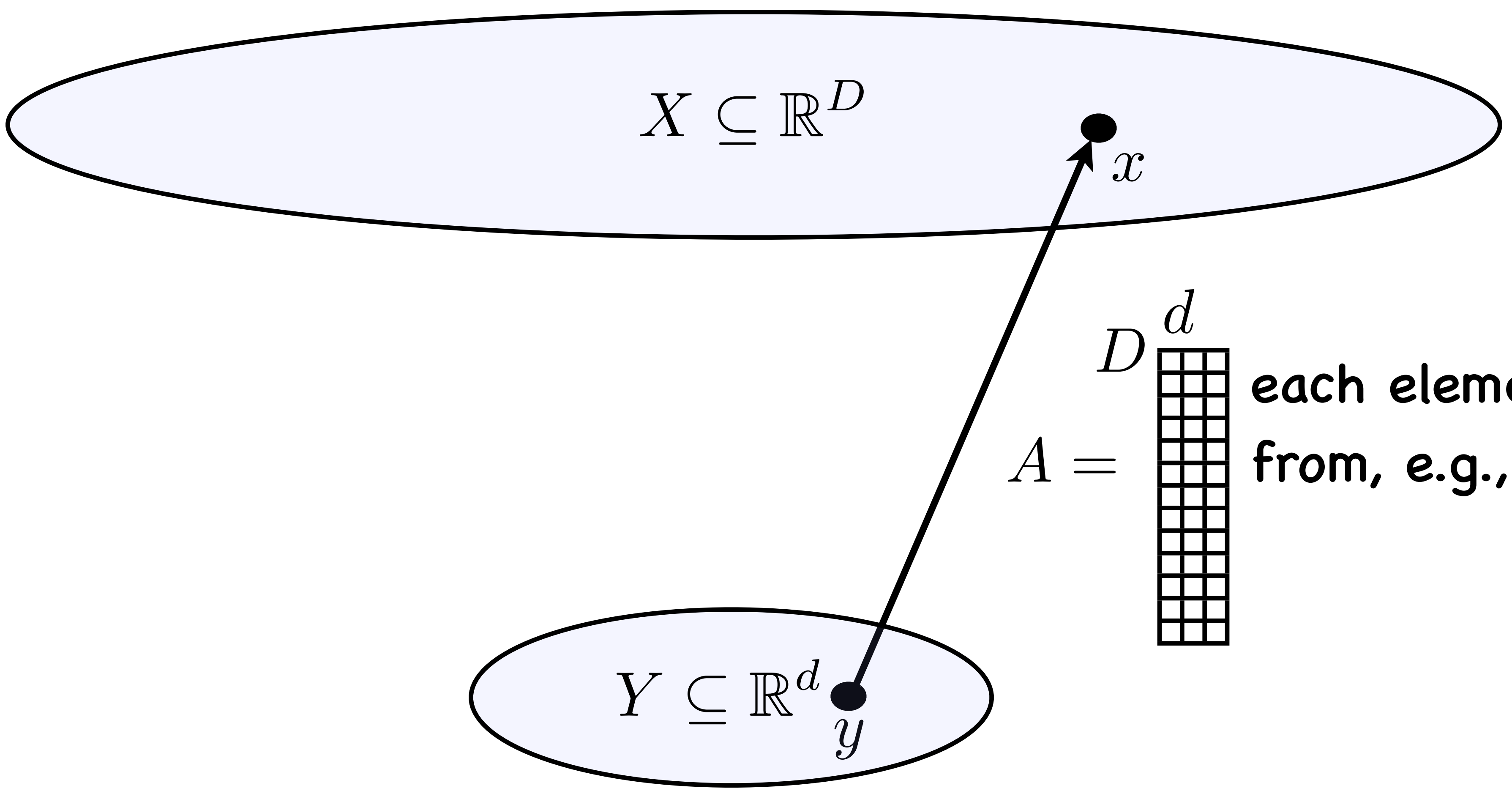
Random embedding



$$\min_{x \in X} f(x)$$

separate the search space and the evaluation space

Random embedding

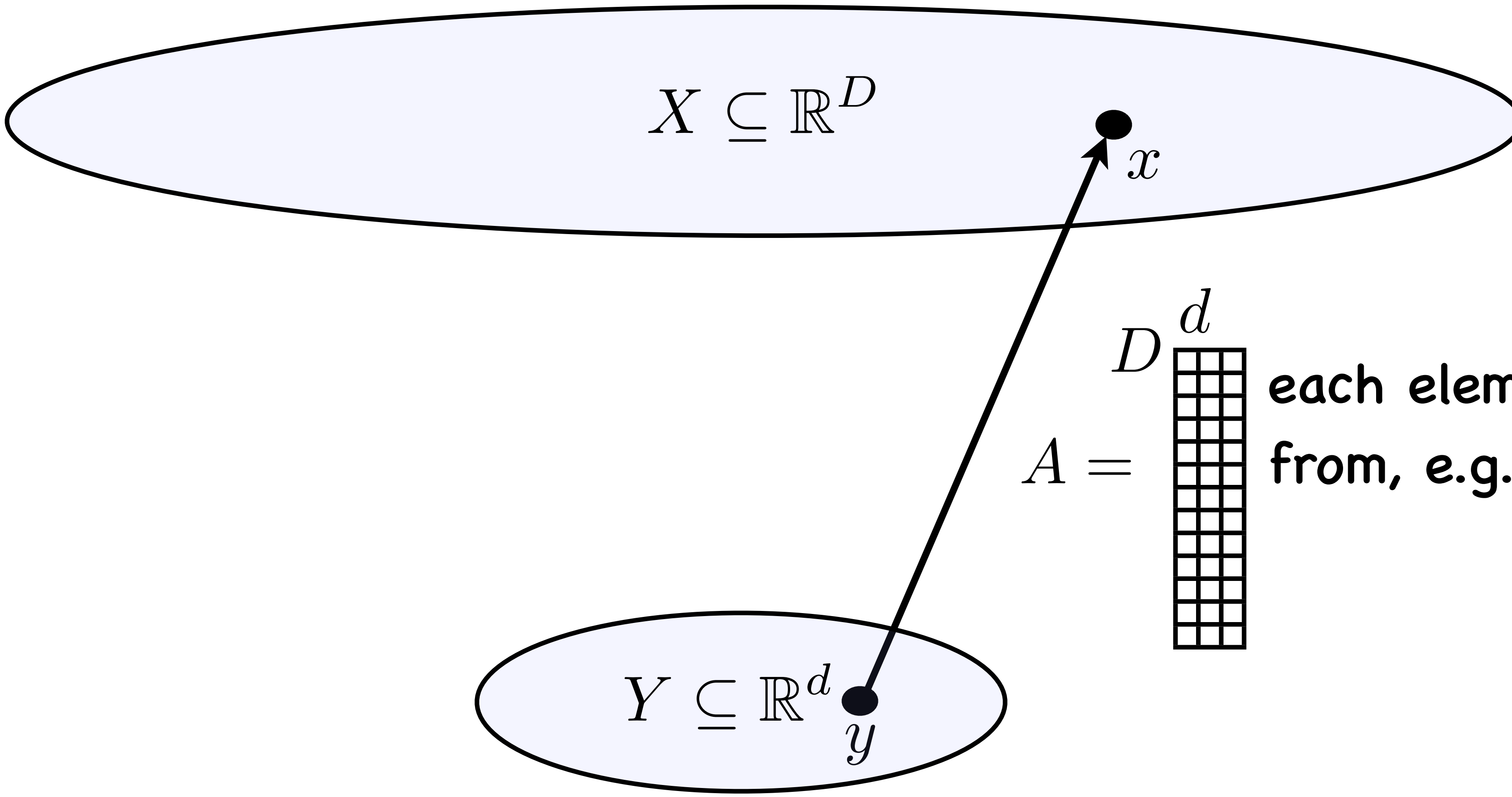


$$\min_{x \in X} f(x)$$

each element is randomly sampled from, e.g., a Gaussian distribution

separate the search space and the evaluation space

Random embedding



$$\min_{x \in X} f(x)$$

each element is randomly sampled from, e.g., a Gaussian distribution

$$\min_{y \in Y} f(Ay)$$

separate the search space and the evaluation space

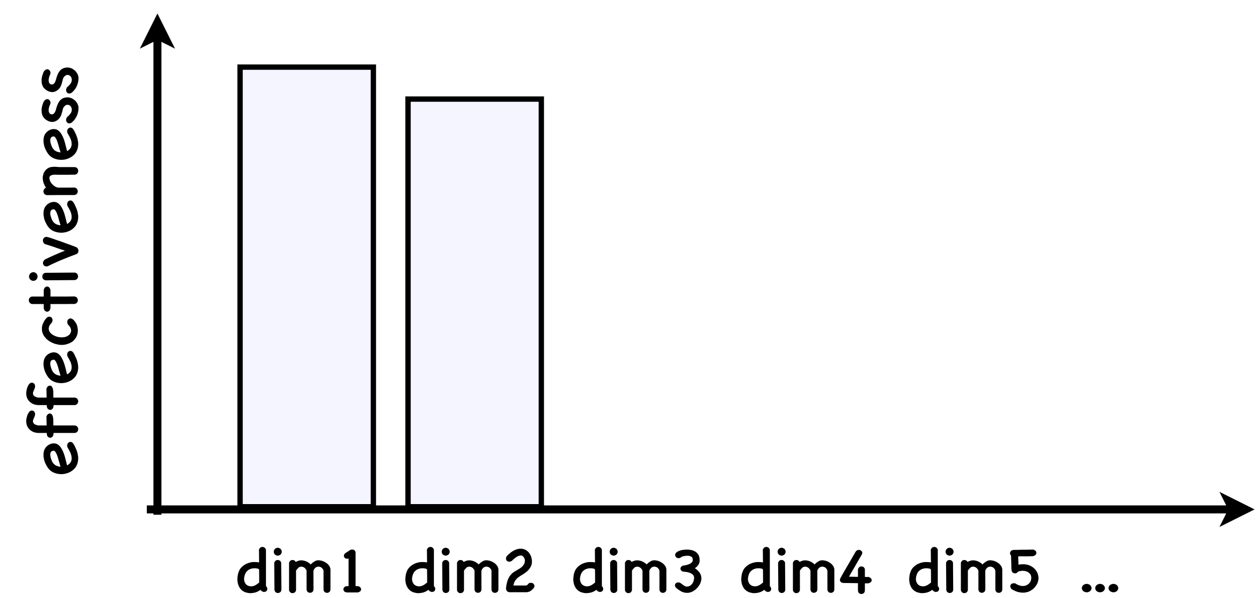
Problems with a low effective dimension

Effective dimension:

A function $f : \mathbb{R}^D \to \mathbb{R}$ is said to have **effective dimension** d_e with $d_e < D$, if there exists a linear subspace $\mathcal{V} \subseteq \mathbb{R}^D$ with dimension d_e such that for all $\boldsymbol{x} \in \mathbb{R}^D$, we have $f(\boldsymbol{x}) = f(\boldsymbol{x}_e + \boldsymbol{x}_c) = f(\boldsymbol{x}_e)$, where $\boldsymbol{x}_e \in \mathcal{V} \subseteq \mathbb{R}^D$, $\boldsymbol{x}_c \in \mathcal{V}^\perp \subseteq \mathbb{R}^D$ and $\mathcal{V}^\perp$ denotes the orthogonal complement of $\mathcal{V}$.

[Wang et al., IJCAI'13]

after some linear rotation



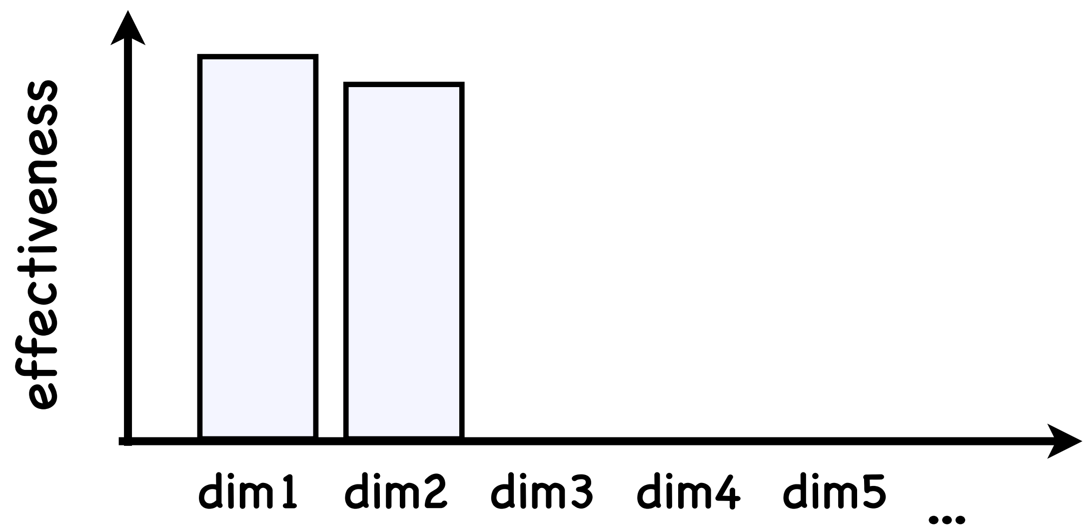
RE + low effective dimension

Given a function $f : \mathbb{R}^D \to \mathbb{R}$ with effective dimension d_e, and a random matrix $\boldsymbol{A} \in \mathbb{R}^{D \times d}$ with independent entries sampled from $\mathcal{N}$ where $d \geq d_e$, then, with probability 1, for any $\boldsymbol{x} \in \mathbb{R}^D$, there exists a $\boldsymbol{y} \in \mathbb{R}^d$ such that $f(\boldsymbol{x}) = f(\boldsymbol{A}\boldsymbol{y})$.

[Wang et al., IJCAI'13]

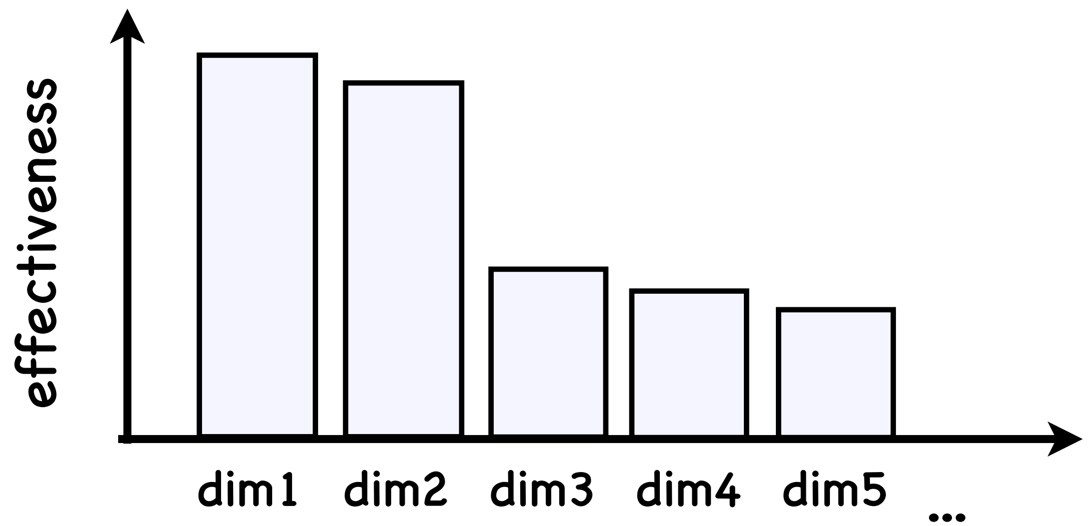
$$\exists \boldsymbol{y}^* \in \mathbb{R}^d \text{ such that } f(\boldsymbol{A}\boldsymbol{y}^*) = f(\boldsymbol{x}^*)$$

the optimal solution is not out of the search space



Random embedding is good for problems with low effective dimensions

What if a problem has no low effective dimension ?



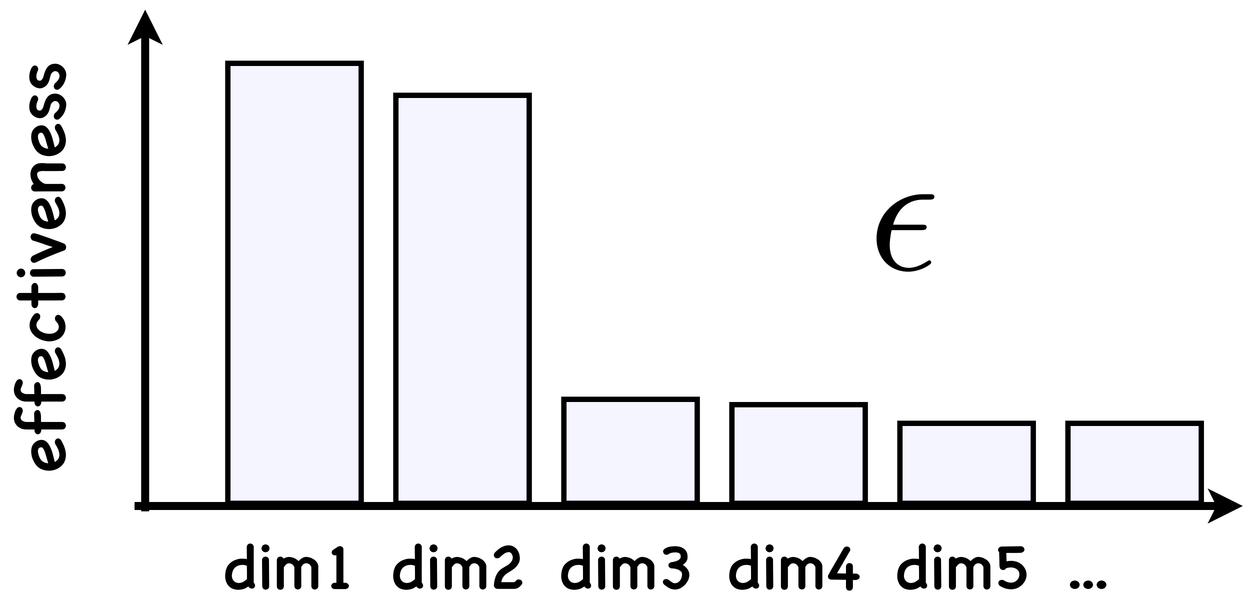
Extend the problems

effective dimension -> ε-effective dimension

For any $\varepsilon > 0$, a function $f : \mathbb{R}^D \to \mathbb{R}$ is said to have an **ε-effective subspace** $\mathcal{V}_\varepsilon$, if there exists a linear subspace $\mathcal{V}_\varepsilon \subseteq \mathbb{R}^D$ s.t. for all $\boldsymbol{x} \in \mathbb{R}^D$, we have $|f(\boldsymbol{x}) - f(\boldsymbol{x}_\varepsilon)| \leq \varepsilon$, where $\boldsymbol{x}_\varepsilon \in \mathcal{V}_\varepsilon$ is the orthogonal projection of $\boldsymbol{x}$ onto $\mathcal{V}_\varepsilon$. Let $\mathbb{V}_\varepsilon$ denote the collection of all the ε-effective subspaces of f, and $\dim(\mathcal{V})$ denote the dimension of a linear subspace $\mathcal{V}$.

We define the **optimal ε-effective dimension** of f as $d_\varepsilon = \min_{\mathcal{V}_\varepsilon \in \mathbb{V}_\varepsilon} \dim(\mathcal{V}_\varepsilon)$.

after some linear rotation



RE revisit

The embedding gap:

Given a function $f: \mathbb{R}^D \to \mathbb{R}$ with optimal ε-effective dimension d_ε, and any random matrix $\boldsymbol{A} \in \mathbb{R}^{D\times d}$ $(d \geq d_\varepsilon)$ with independent entries sampled from $\mathcal{N}$, then, with probability 1, for any $\boldsymbol{x} \in \mathbb{R}^D$, there exists $\boldsymbol{y} \in \mathbb{R}^d$ such that

$$|f(\boldsymbol{x}) - f(\boldsymbol{A}\boldsymbol{y})| \leq 2\varepsilon$$

Random embedding can be applied !

RE revisit

The embedding gap:

Given a function $f : \mathbb{R}^D \to \mathbb{R}$ with optimal ε-effective dimension d_ε, and any random matrix $\boldsymbol{A} \in \mathbb{R}^{D \times d}$ $(d \geq d_\varepsilon)$ with independent entries sampled from $\mathcal{N}$, then, with probability 1, for any $\boldsymbol{x} \in \mathbb{R}^D$, there exists $\boldsymbol{y} \in \mathbb{R}^d$ such that

$$|f(\boldsymbol{x}) - f(\boldsymbol{A}\boldsymbol{y})| \leq 2\varepsilon$$

Random embedding can be applied !

This gap cannot be compensated by optimization

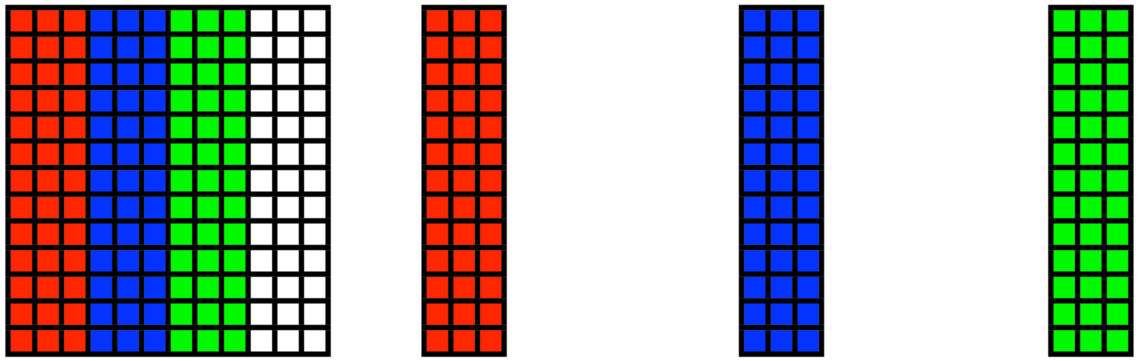
$$f(\boldsymbol{A}\tilde{\boldsymbol{y}}) - f(\boldsymbol{x}^*) = f(\boldsymbol{A}\tilde{\boldsymbol{y}}) - \inf_{\boldsymbol{y} \in \mathbb{R}^d} f(\boldsymbol{A}\boldsymbol{y}) + \inf_{\boldsymbol{y} \in \mathbb{R}^d} f(\boldsymbol{A}\boldsymbol{y}) - f(\boldsymbol{x}^*)$$

$$\leq \theta + 2\epsilon$$

optimization gap + embedding gap

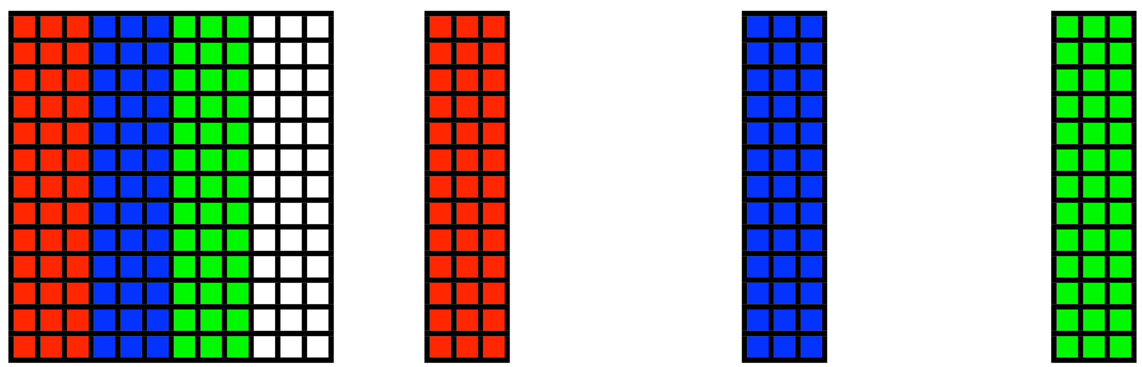
Sequential random embedding (SRE)

$$x = Ay = A^{(1)}y_1 + A^{(2)}y_2 + A^{(3)}y_3 \ldots$$



Sequential random embedding (SRE)

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{y} = \boldsymbol{A}^{(1)}\boldsymbol{y}_1 + \boldsymbol{A}^{(2)}\boldsymbol{y}_2 + \boldsymbol{A}^{(3)}\boldsymbol{y}_3 \ldots$$



Sequential random embedding

- Firstly, generate a random matrix $\boldsymbol{A}^{(1)}$, solve $\tilde{\boldsymbol{y}}_1 = \mathrm{argmin}_{\boldsymbol{y}} f(\boldsymbol{A}^{(1)}\boldsymbol{y})$ with some derivative-free method. Let $\tilde{\boldsymbol{x}}_1 = \boldsymbol{0}$ and $\tilde{\boldsymbol{x}}_2 = \boldsymbol{A}^{(1)}\tilde{\boldsymbol{y}}_1$;
- Secondly, generate a random matrix $\boldsymbol{A}^{(2)}$, solve $\tilde{\boldsymbol{y}}_2 = \mathrm{argmin}_{\boldsymbol{y}} f(\tilde{\boldsymbol{x}}_2 + \boldsymbol{A}^{(2)}\boldsymbol{y})$. Update the current solution $\tilde{\boldsymbol{x}}_3 = \tilde{\boldsymbol{x}}_2 + \boldsymbol{A}^{(2)}\tilde{\boldsymbol{y}}_2$;
- In the following steps, it acts like the second step that performs the optimization.

Sequential random embedding (SRE)

Theoretical property:

- Assumption 1: functions with optimal ε-effective dimension
- Assumption 2: Local Holder Continuity

$$f(\boldsymbol{x}) - f(\boldsymbol{x}^*) \leq L \cdot \|\boldsymbol{x} - \boldsymbol{x}^*\|_2^\alpha \text{ with } \alpha > 0$$

- Assumption 3:

$$\|\hat{\boldsymbol{x}}_i - \boldsymbol{A}^{(i)}\tilde{\boldsymbol{y}}_i\| / \|\hat{\boldsymbol{x}}_i\| \leq (1/5) \cdot \|\hat{\boldsymbol{x}}_i\| / \|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_i\|$$

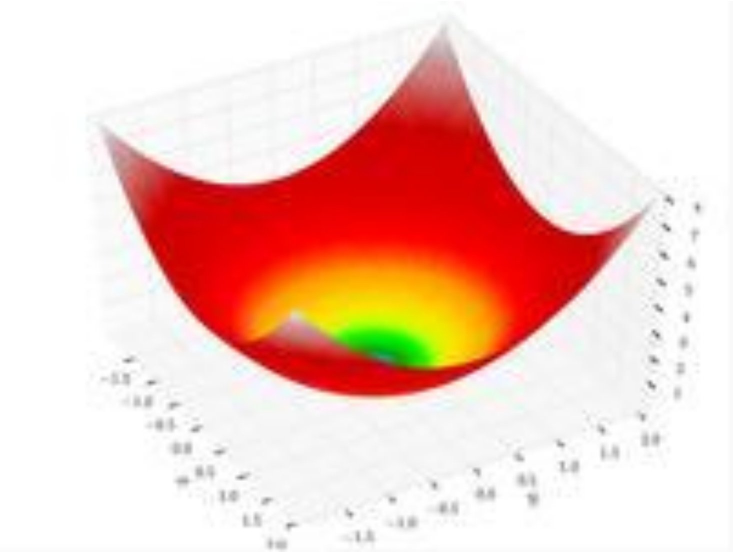
$\hat{\boldsymbol{x}}_i$ is the orthonormal projection of $\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_i$ onto the subspace $\mathcal{S}_i = \{\boldsymbol{A}^{(i)}\boldsymbol{y} \,|\, \boldsymbol{y} \in \mathbb{R}^d\}$

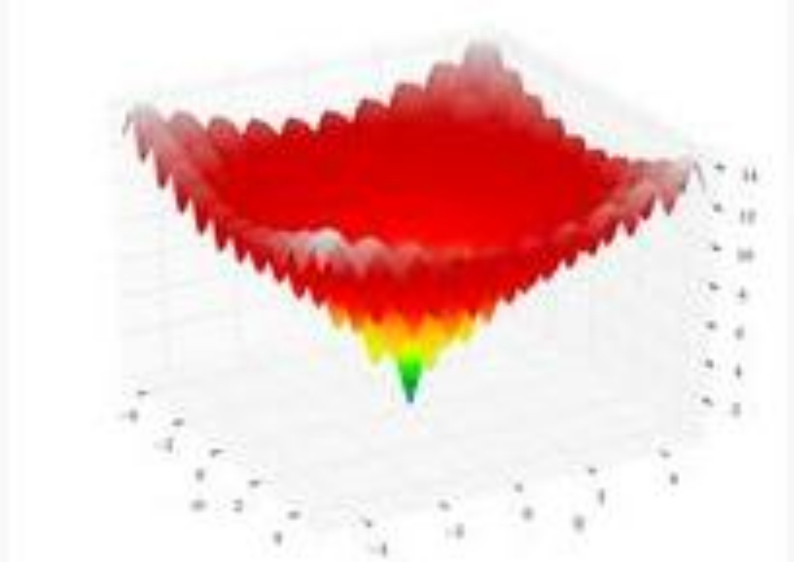
SRE could reduce the embedding gap strictly in each step.

$$\|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_i\| > \|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}_{i+1}\|$$

Experiments

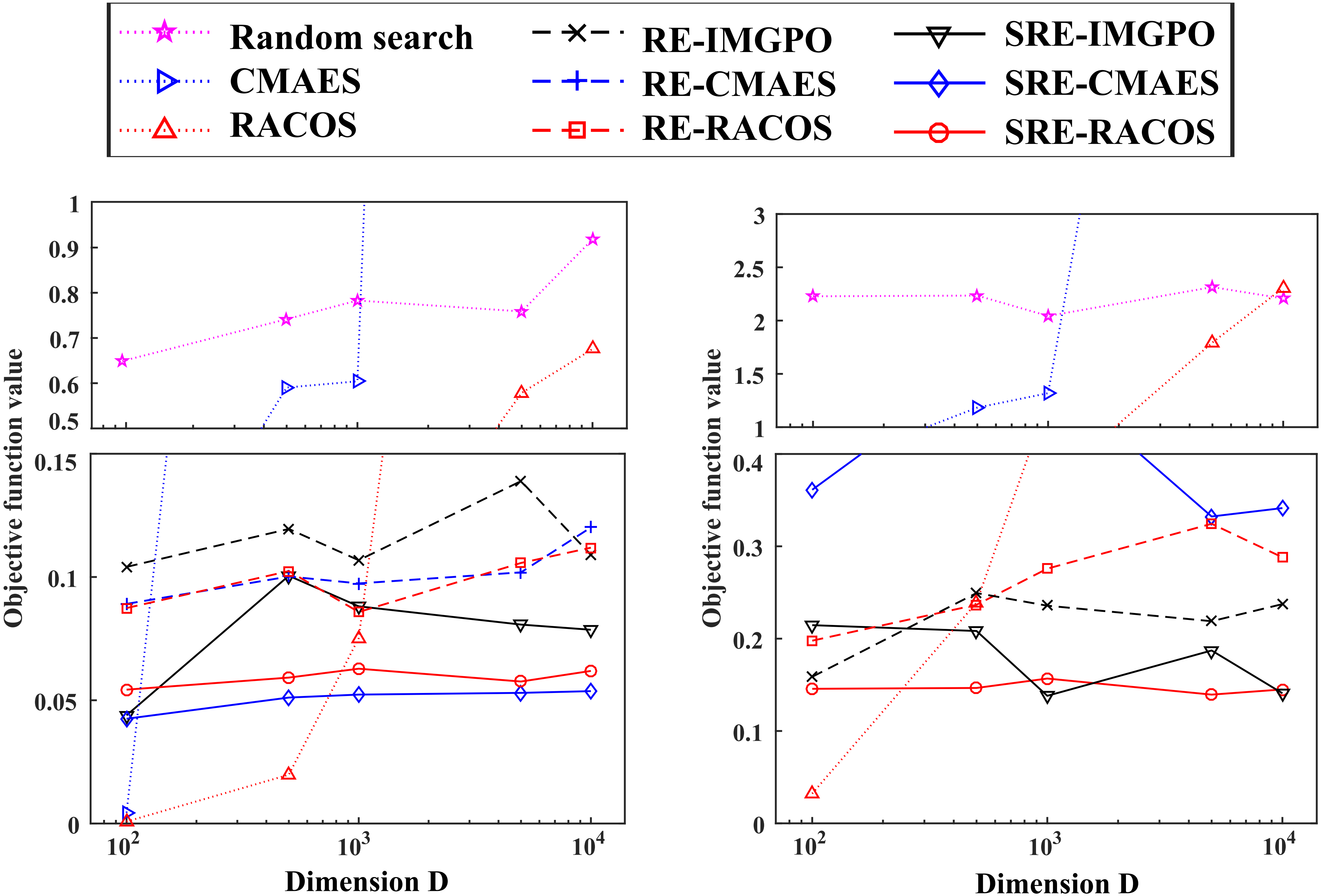
Synthetic functions: extend to high-dim by adding variables with small effect





- set $\mathcal{X} = [-1,1]^D, \ \mathcal{Y} = [-1,1]^d$
- compared methods:
  - Random Search, CMAES, RACOS
  - RE-IMGPO, RE-CMAES, RE-RACOS
  - SRE-IMGPO, SRE-CMAES, SRE-RACOS

On scalability over D



$n = 10000$

$d = 10$

$m = 5$

(a) on Sphere function (b) on Ackley function

Applications in classification

the loss function for linear SVM

$$f(w,b) = \frac{1}{2}\|w\|_2^2 + C\sum_{\ell}^{L} \max\{0,\ 1 - y_\ell(w^\top v_\ell + b)\}$$

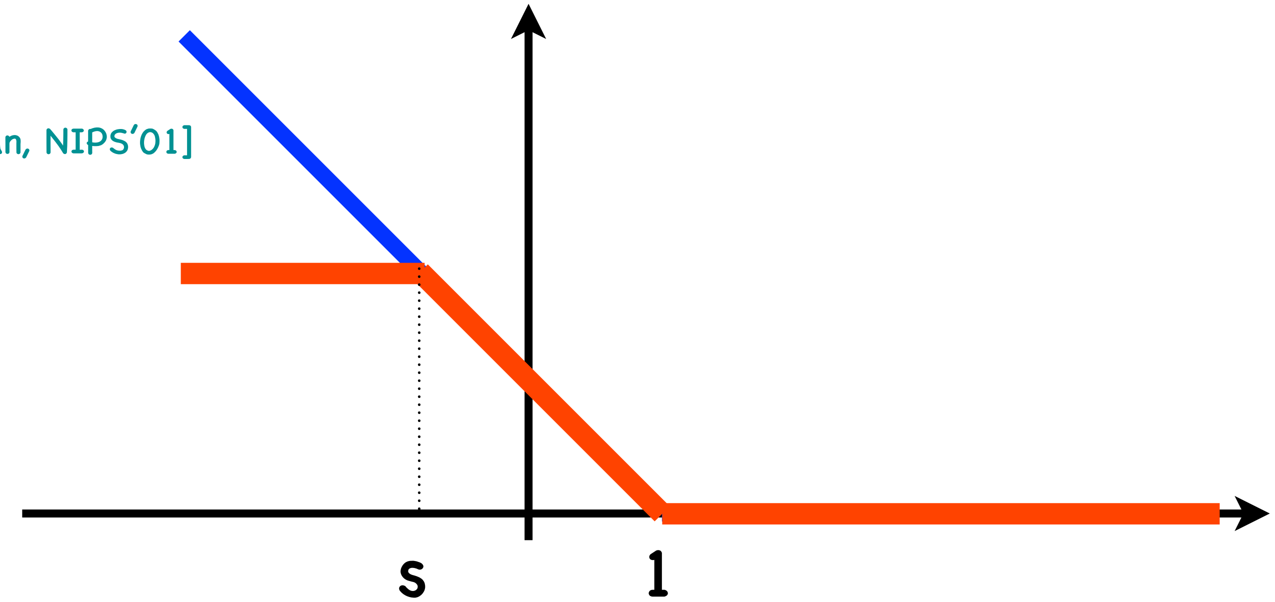
the loss function using Ramp loss

$$f(w,b) = \frac{1}{2}\|w\|_2^2$$

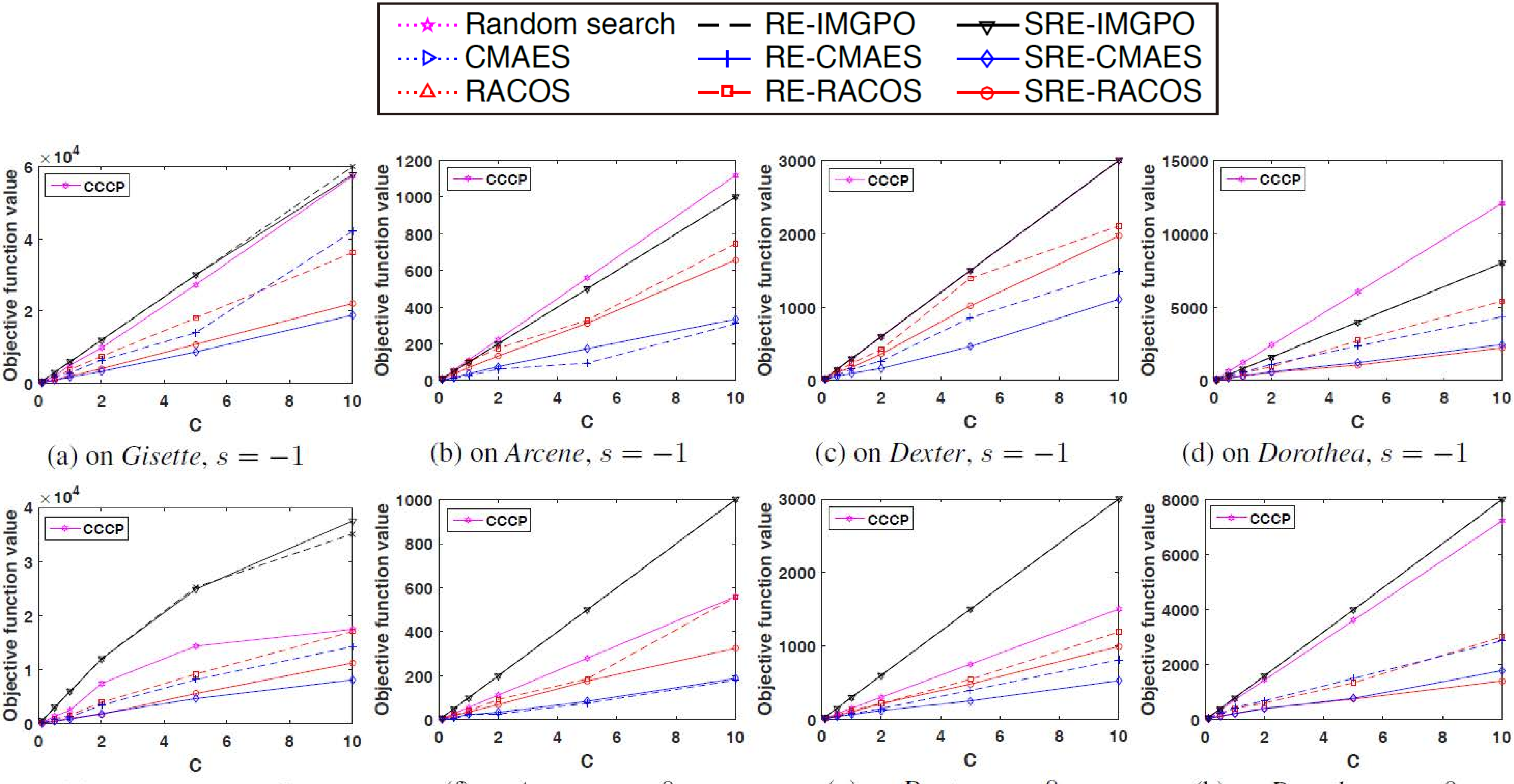
$$+ C\sum_{\ell}^{L} \Big(\max\{0,\ 1 - y_\ell(w^\top v_\ell + b)\} - \max\{0,\ s - y_\ell(w^\top v_\ell + b)\} \Big)$$

previous solution: CCCP[Yuille and Rangarajan, NIPS'01]

- relax the concave part to be linear
- gradient decent



Results



(a) on *Gisette*, $s = -1$ (b) on *Arcene*, $s = -1$ (c) on *Dexter*, $s = -1$ (d) on *Dorothea*, $s = -1$

(e) on *Gisette*, $s = 0$ (f) on *Arcene*, $s = 0$ (g) on *Dexter*, $s = 0$ (h) on *Dorothea*, $s = 0$

D=10,000 **D=5,000** **D=20,000** **D=100,000**

References for classification-based optimization

- Yang Yu, Hong Qian, and Yi-Qi Hu. Derivative-free optimization via classification. In: **Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)**, Phoenix, AZ, 2016.

- Hong Qian, Yang Yu. Scaling simultaneous optimistic optimization for high-dimensional non-convex functions with low effective dimensions. In: **Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)**, Phoenix, AZ, 2016.

- Hong Qian, Yi-Qi Hu and Yang Yu. Derivative-free optimization of high-dimensional non-convex functions by sequential random embeddings. In: **Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI'16)**, New York, NY, 2016.

Conclusion

Subset selection problem

and Pareto optimization

- can be shown to be the currently best approximation algo.
- extension: parallel version
- useful in ensemble selection, sparse regression, etc.

Local Lipschitz continuous problem

and classification-based optimization

- shown to be efficient for local Lipschitz continuous problems
- extension: high-dimensional optimization
- extension: sequential optimization (unpublished)
- useful in robust classification, reinforcement learning, etc.

THANK YOU!

yuy@nju.edu.cn
http://cs.nju.edu.cn/yuy