南京大学 人工智能学院

SCHOOL OF ARTIFICIAL INTELLIGENCE, NANJING UNIVERSITY

Heuristic Search and Evolutionary Algorithms

Lecture 13: Evolutionary Algorithms Made Faster by Surrogate Models

Chao Qian （钱超）

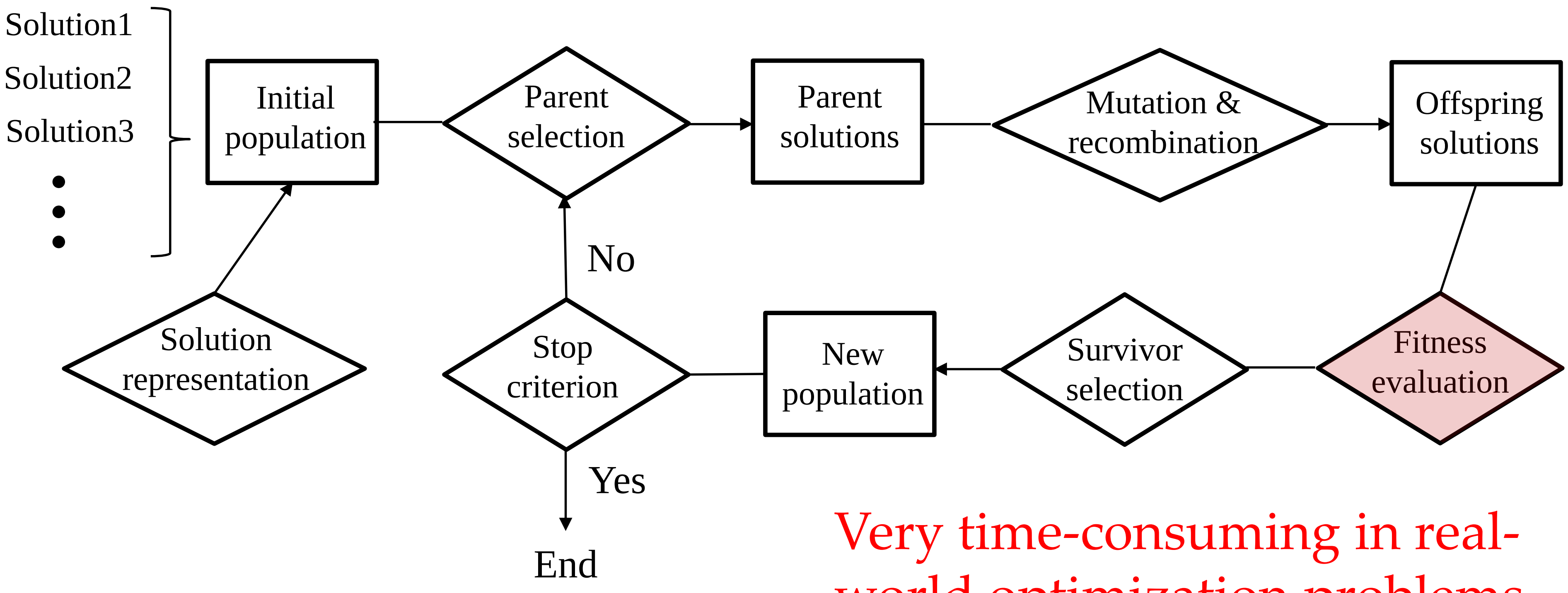
Associate Professor, Nanjing University, China

Email: qianc@nju.edu.cn

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

Expensive fitness evaluation

General structure of evolutionary algorithms

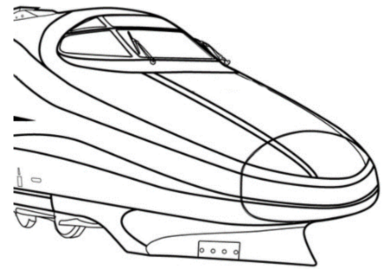


Very time-consuming in real-world optimization problems

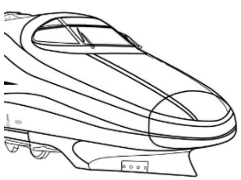
Examples

Optimize the efficiency of the train head

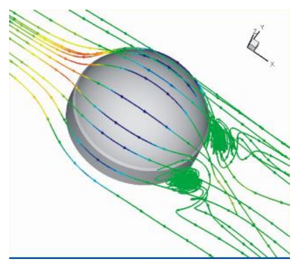
parameterize



represented as a vector of parameters



x_i

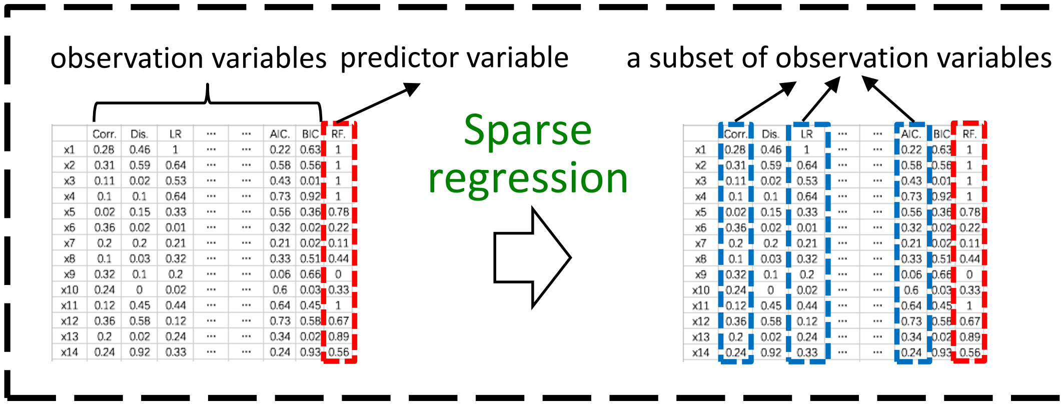


test by simulation

$f(x_i)$

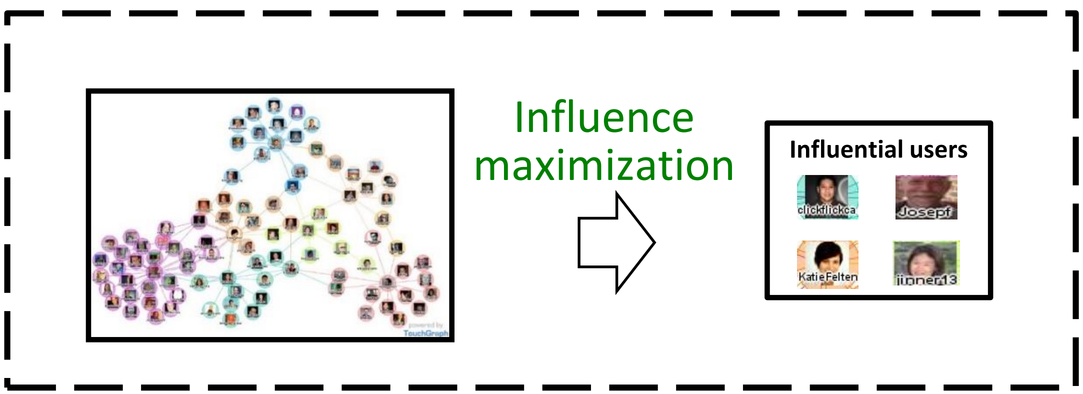
time-consuming

Examples



Hundreds of millions of instances

Computing the R^2 objective is very expensive



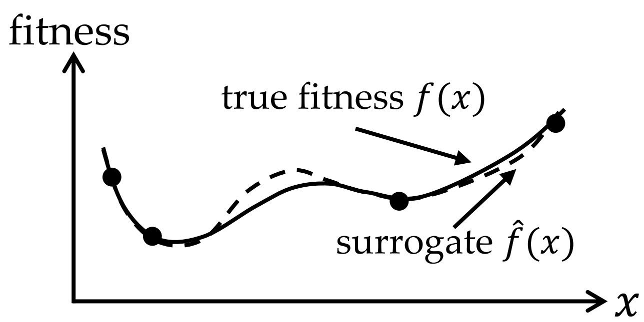
Computing the influence spread objective is #P-hard

Estimated by the average of 10,000 random diffusions

Very expensive

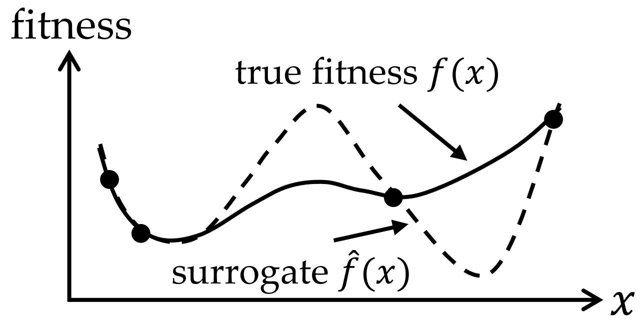
Surrogate models

Collect some data points to build a surrogate model



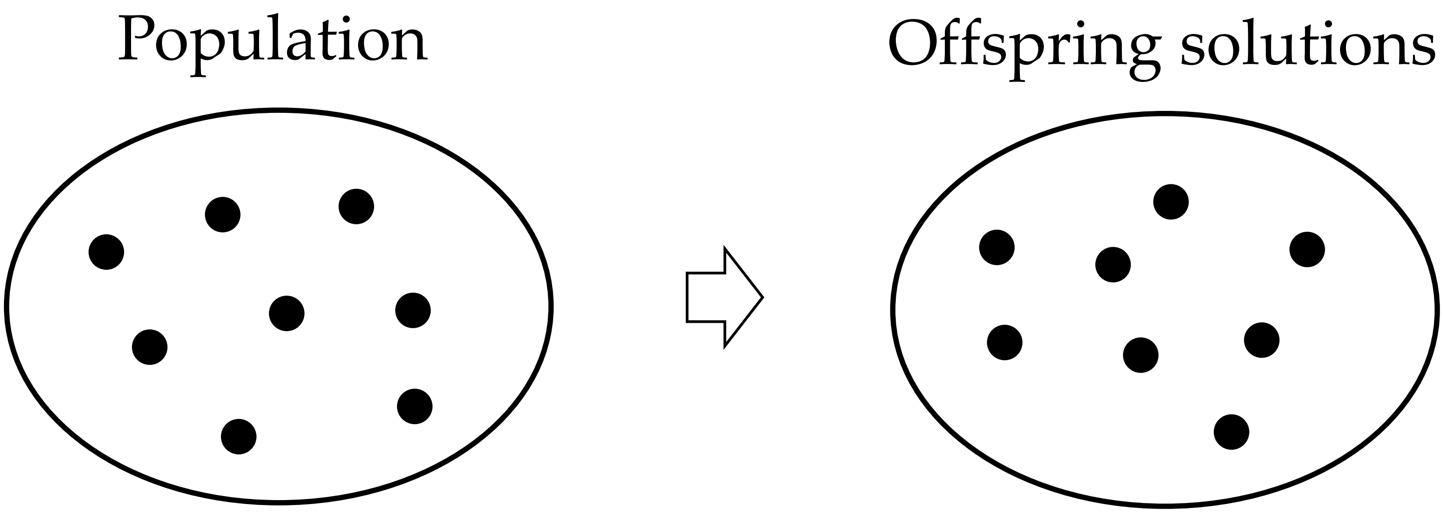
Use the surrogate to approximate the true fitness function

However,



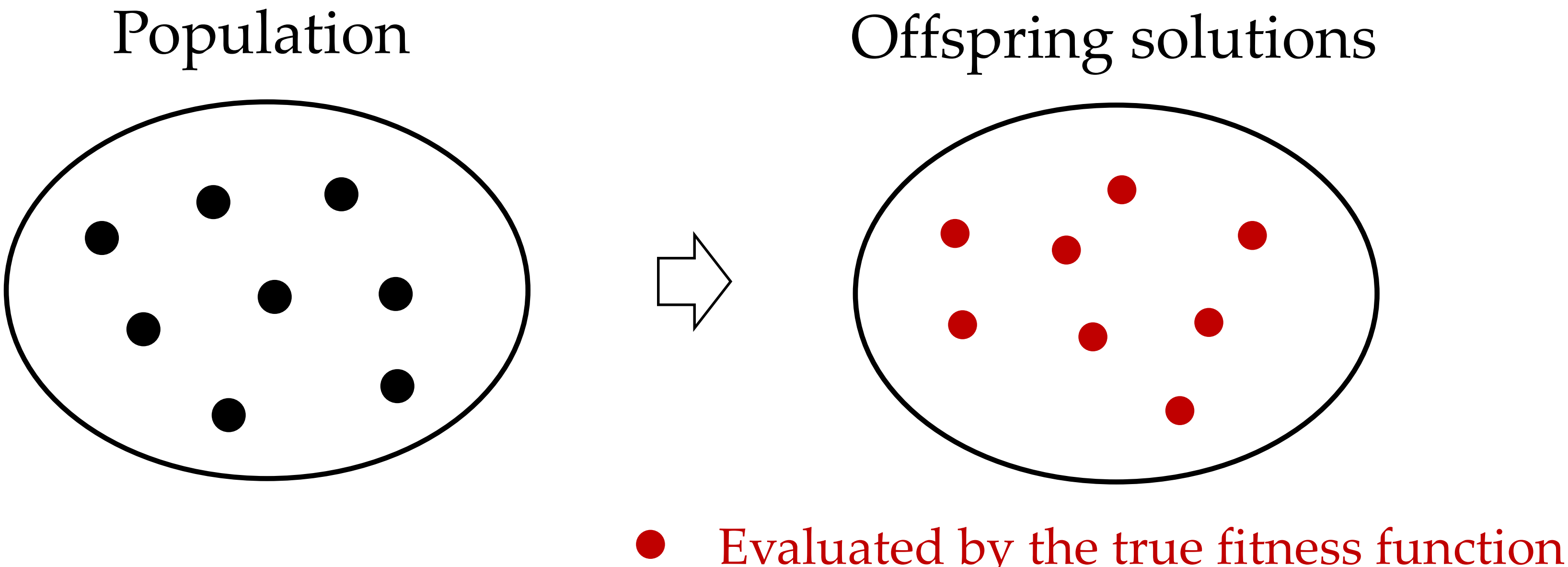
How to use surrogates

We should use surrogate models carefully



How to use surrogates

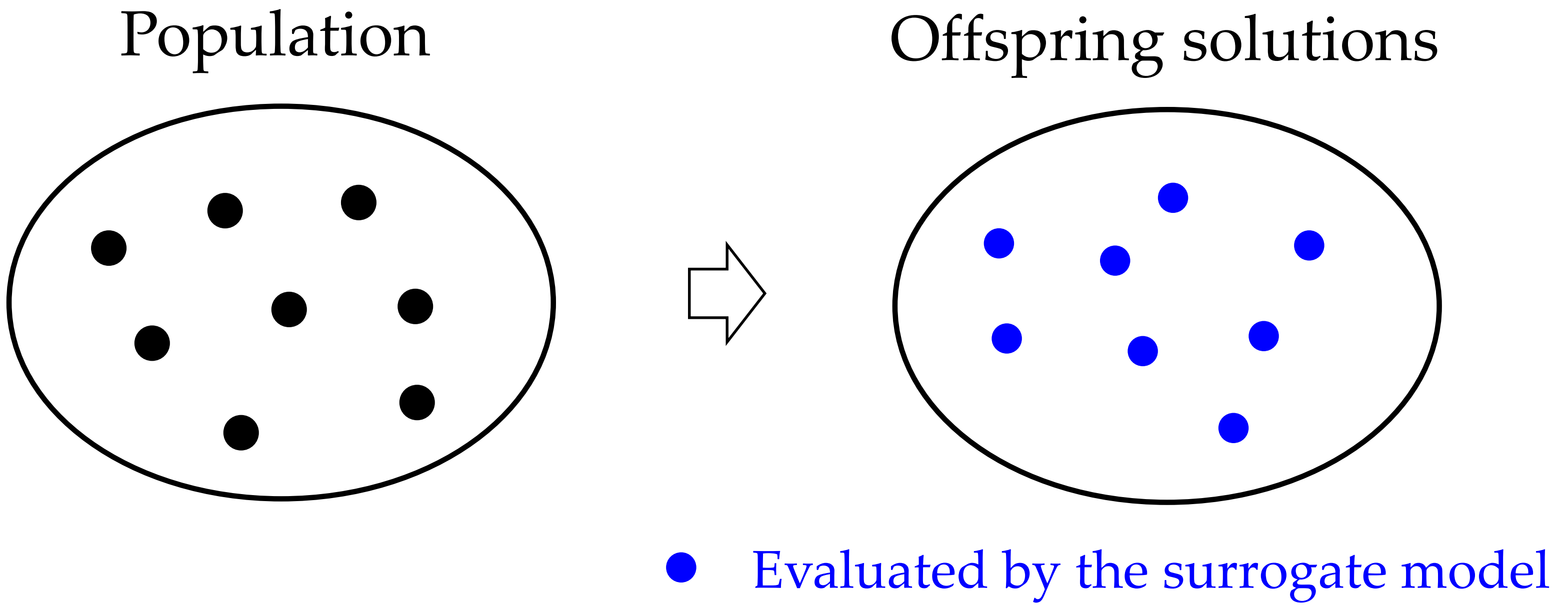
We should use surrogate models carefully



Too expensive

How to use surrogates

We should use surrogate models carefully



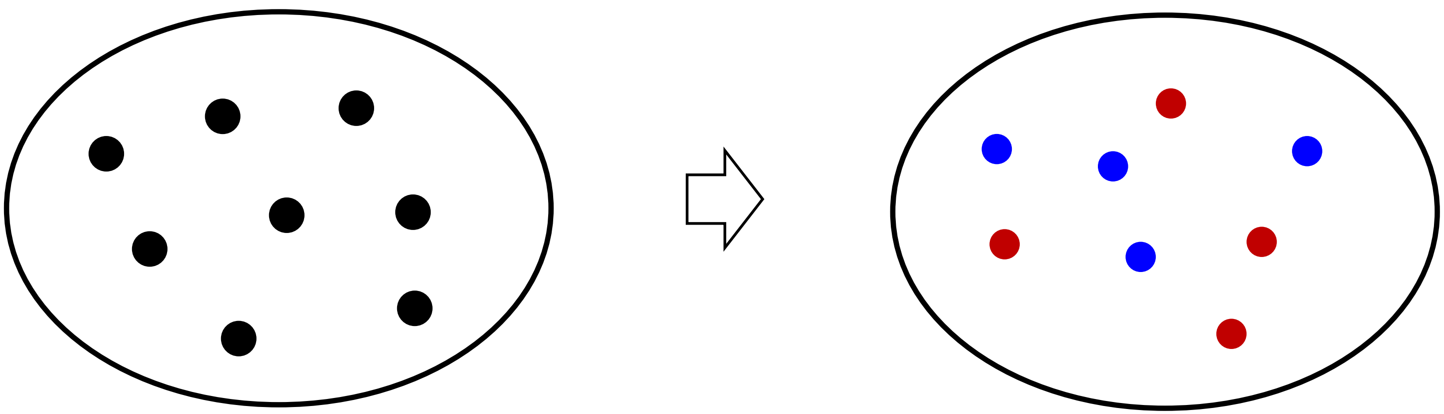
Too risky

How to use surrogates

We should use surrogate models carefully

Population

Offspring solutions



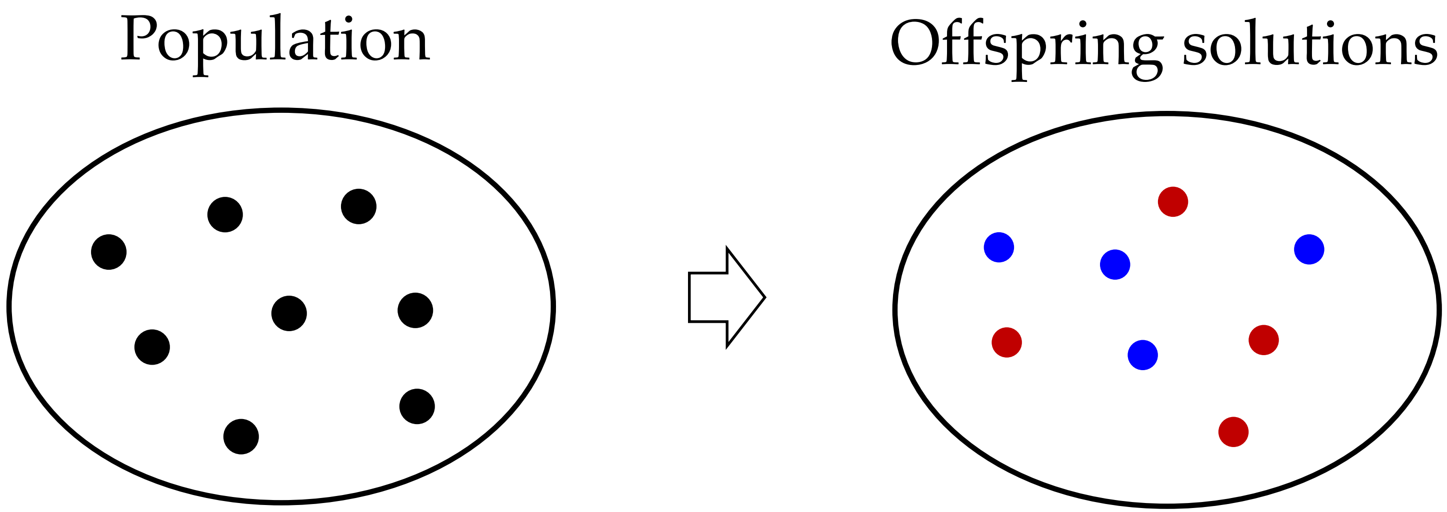
● Evaluated by the surrogate model

● Evaluated by the true fitness function

Which solutions are ● or ● ?

How to use surrogates

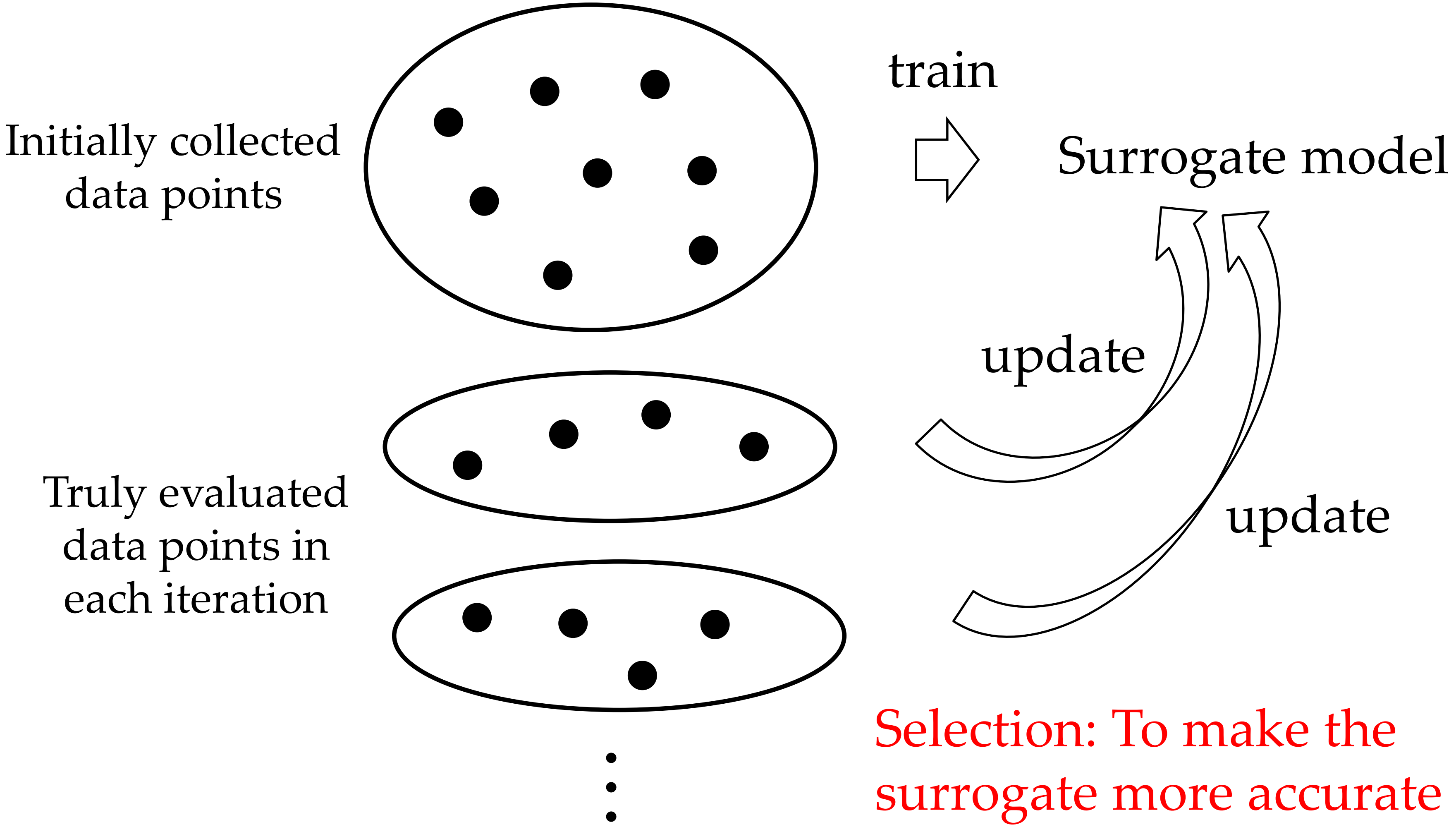
We should use surrogate models carefully



Random strategy

- Select some offspring solutions randomly to be evaluated using the true fitness function
- Evaluate the remaining ones using the surrogate model

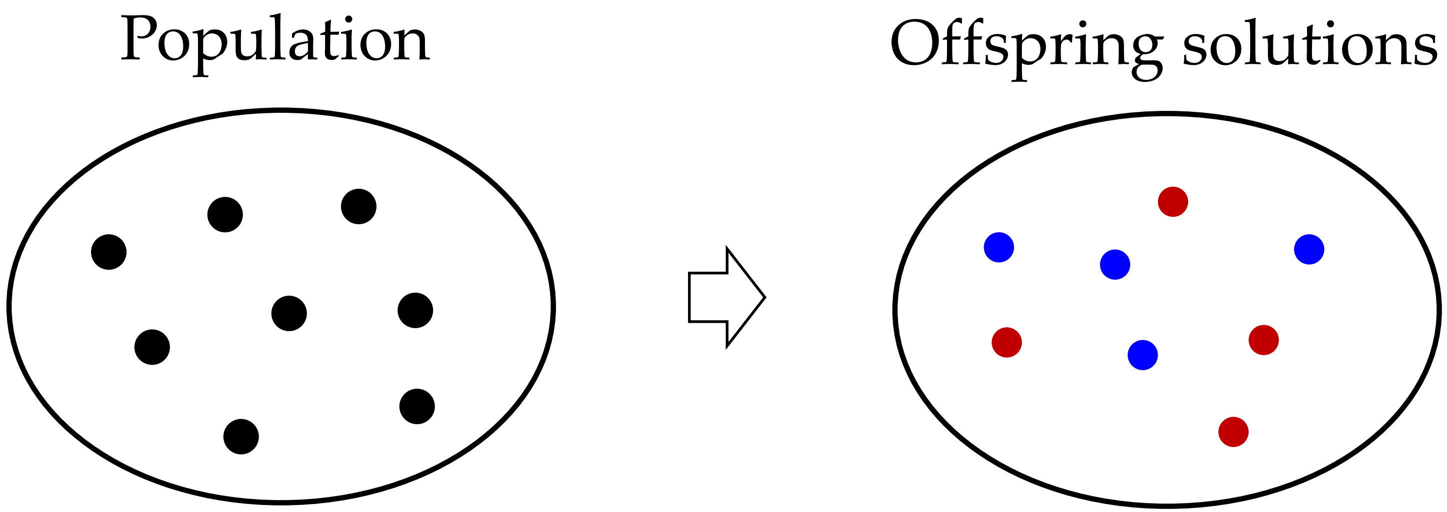
Surrogate models during evolution



Selection: To make the surrogate more accurate

How to use surrogates

We should use surrogate models carefully

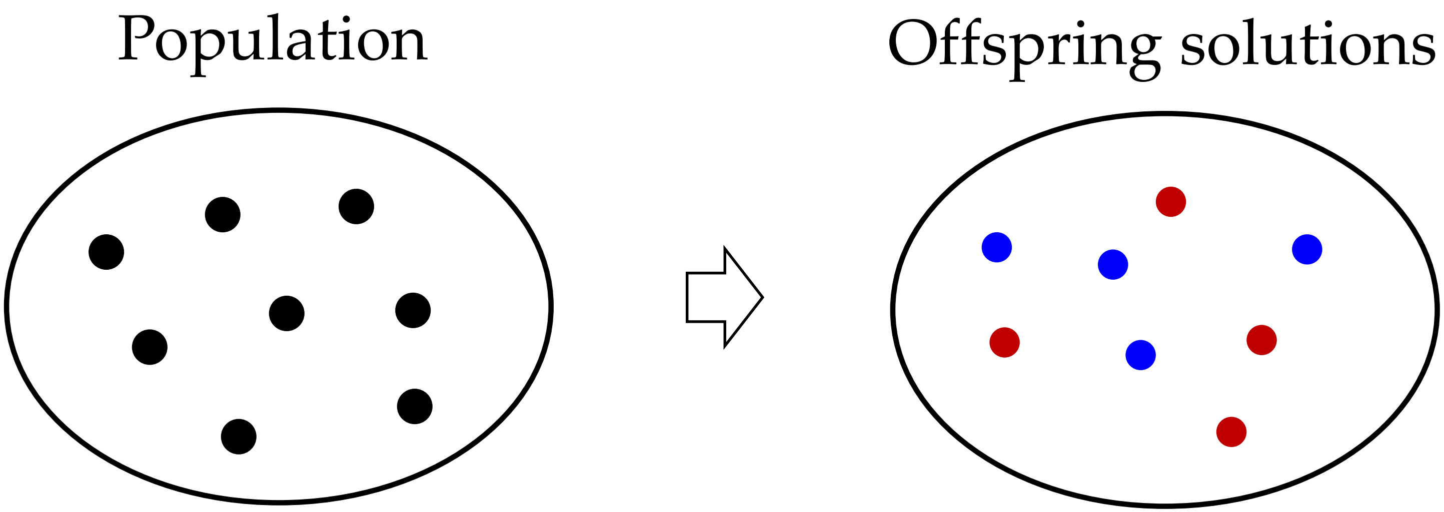


Best strategy

- Evaluate all N offspring solutions using the surrogate model
- Re-evaluate the best $N' < N$ offspring solutions using the true fitness function

How to use surrogates

We should use surrogate models carefully

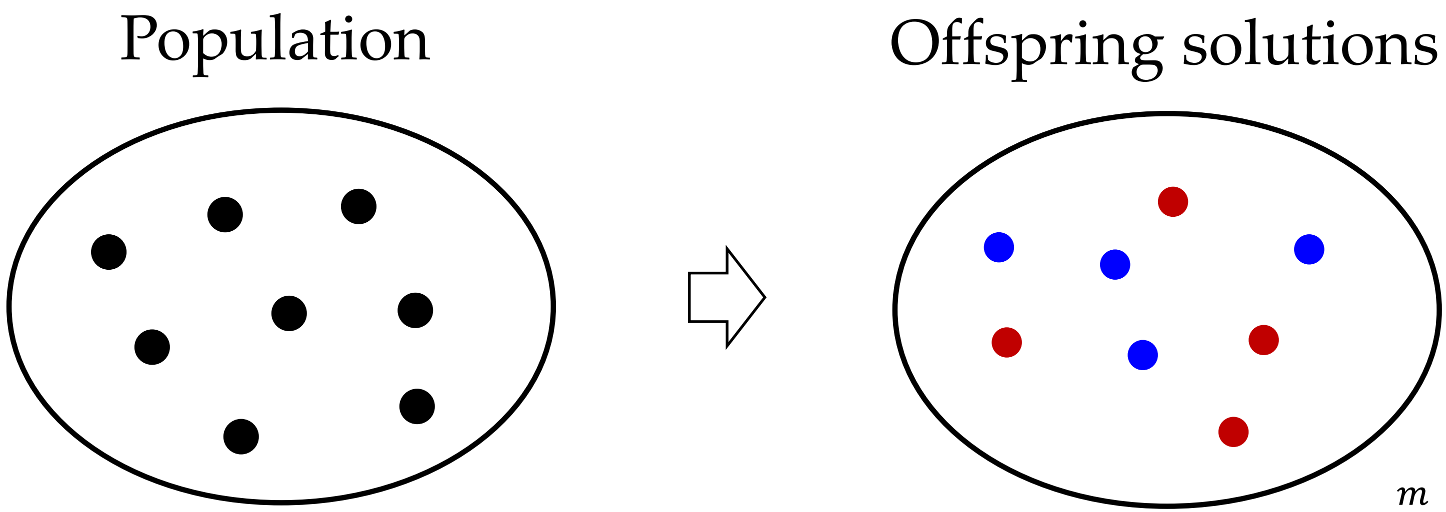


Clustering based strategy

- Group the offspring solutions into a number of clusters
- Select some representative solutions from each cluster to be evaluated using the true fitness function
- Evaluate the remaining ones using the surrogate model

How to use surrogates

We should use surrogate models carefully

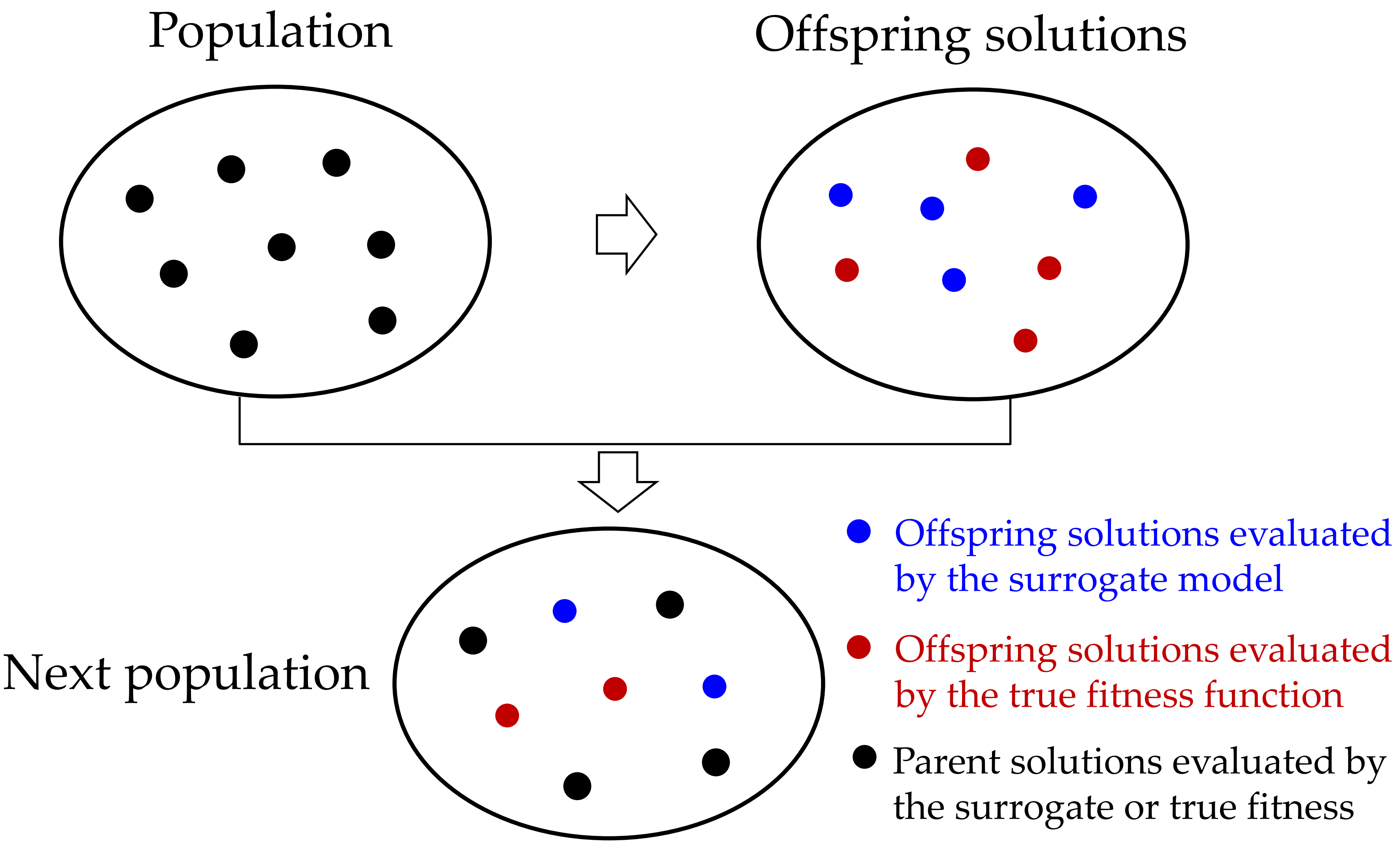


$$\delta_i = \sum_{j=1}^{m} \|x_i - x_j\|$$

Uncertainty based strategy

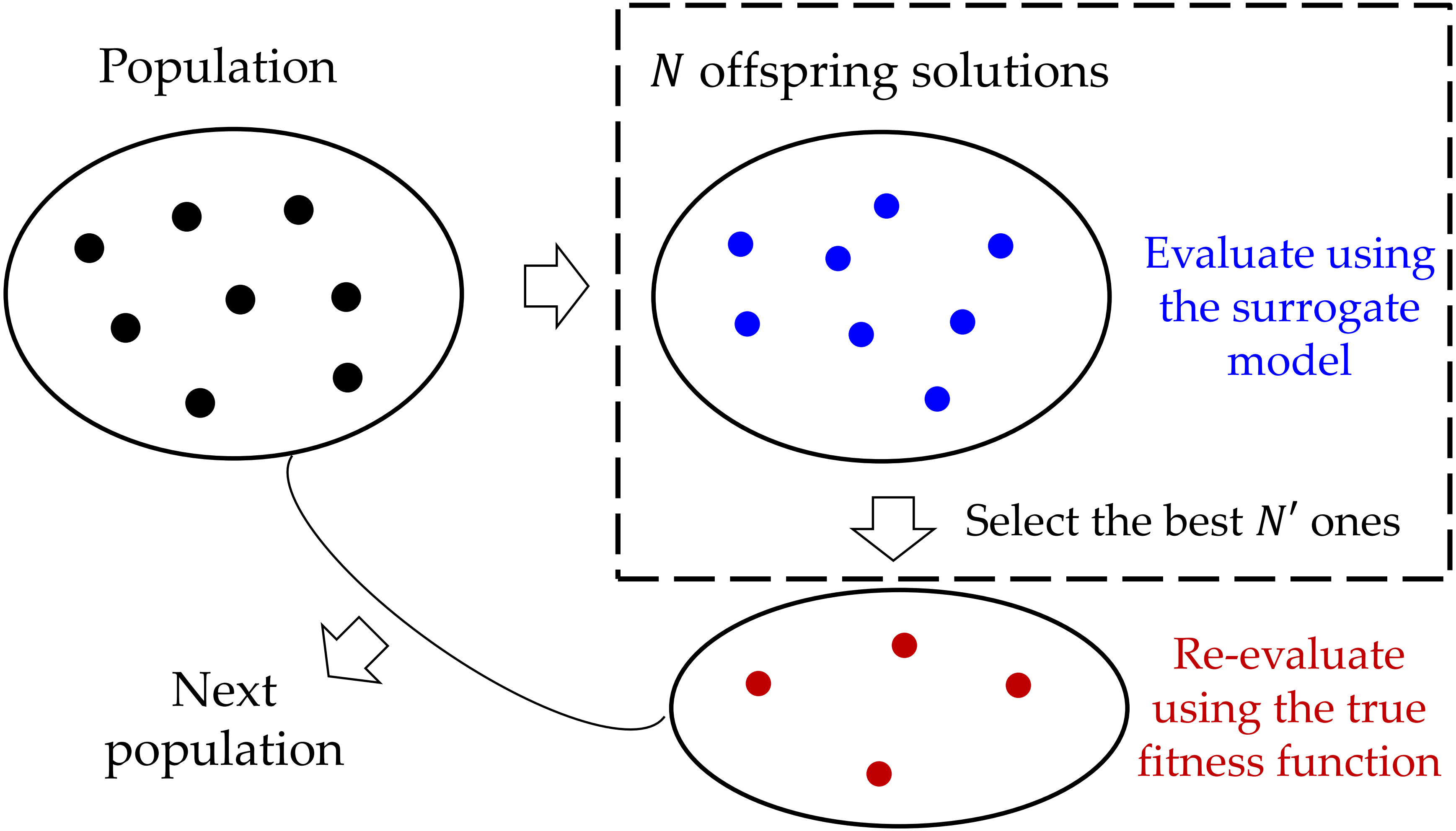
- Select some "uncertain" offspring solutions to be evaluated using the true fitness function
- Evaluate the remaining ones using the surrogate model

How to use surrogates



How to use surrogates

Preselection strategy



An example of preselection-based EA

[Hao et al., TEvC'20]

Algorithm 1 EA Framework With Preselection

Input: N: the population size;
M: the trial vectors size;
$MaxFES$: maximum number of function evaluations;

Output: the best found solution in the population;

Initialize the population $P = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ and evaluate them by the real objective function f;
Set number of fitness evaluations $fes = N$;
while $fes \leq MaxFES$ **do**
  Train a surrogate model based on P :

$$\mathfrak{M} = \textbf{SurrogateTrain}(P);$$

  for each $\mathbf{x} \in P$ **do**
    Generate M trail solutions $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M$ for the parent solutions $\mathbf{x}$ by generation operators;
    Choose the potential trial solution :

$$\mathbf{u}^* = \textbf{PreSelection}(\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M\}, \mathfrak{M});$$

    Evaluate $\mathbf{u}^*$ by the real objective function f;
    if $f(\mathbf{u}^*) < f(\mathbf{x})$ **then**
      Set $\mathbf{x} = \mathbf{u}^*$;
    end if
    Set $fes = fes + 1$;
  end for
end while
Output the best solution in P.

Initialization

Train and update the surrogate model

Genenerate M offspring solutions for each parent solution

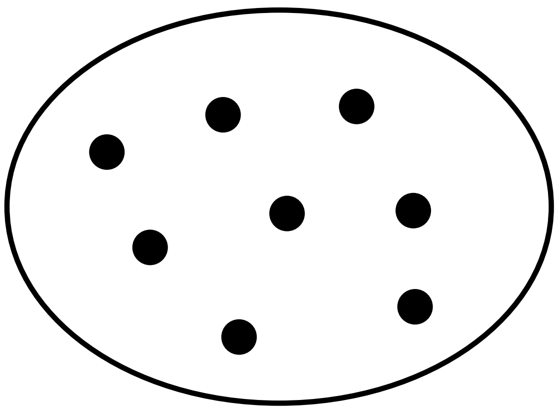
Use the surrogate to select the best offspring solution

Re-evaluate the best offspring

Replace the parent if the offspring is better

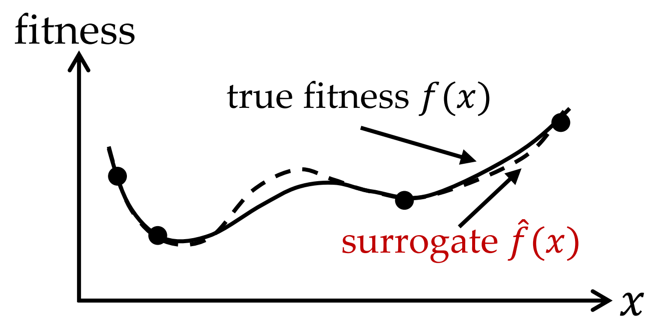
What surrogate models can we use?

Truly evaluated data points



train

Surrogate model



Neural Network

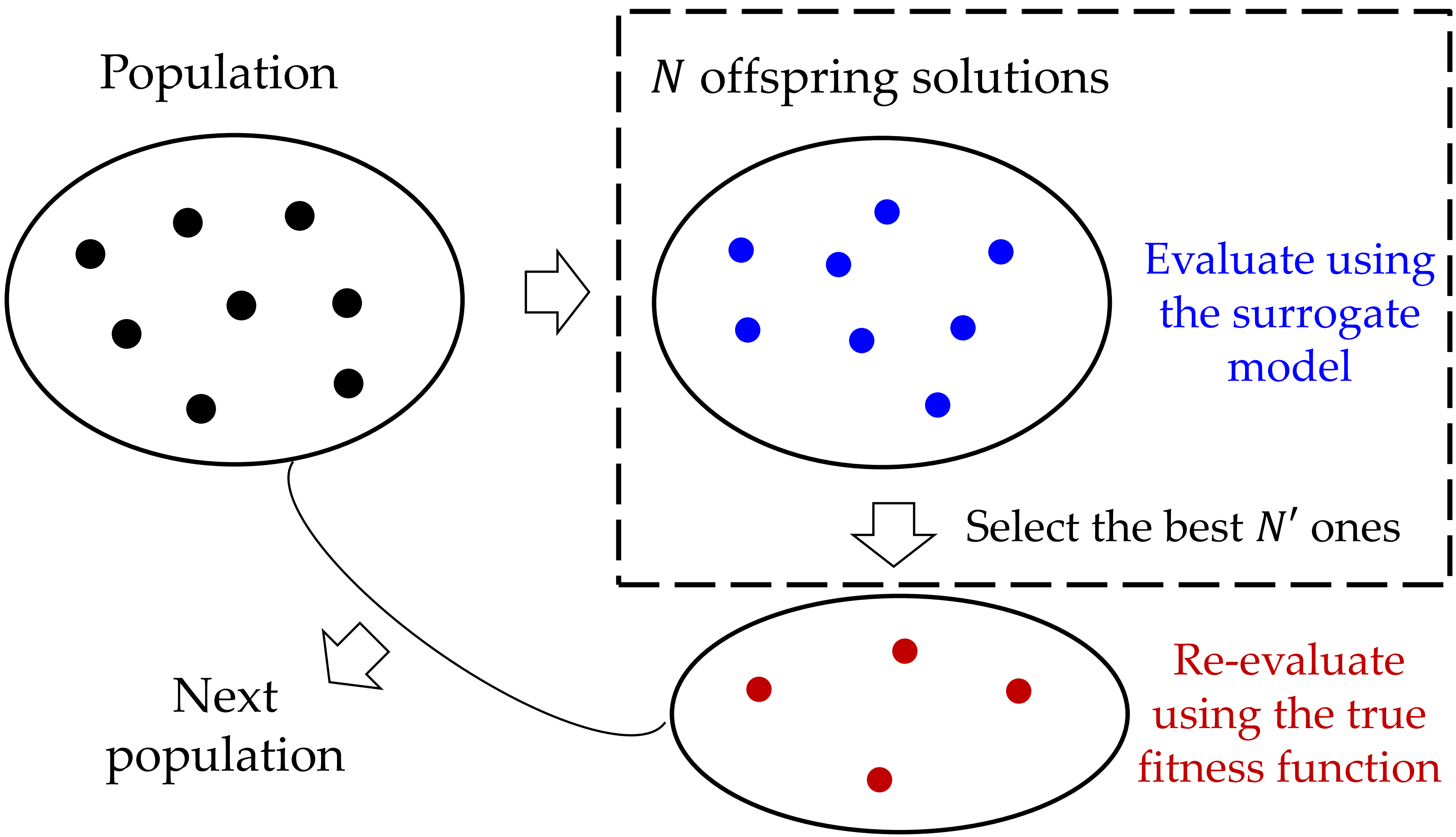
SVM Decision Tree

AdaBoost Gaussian Process

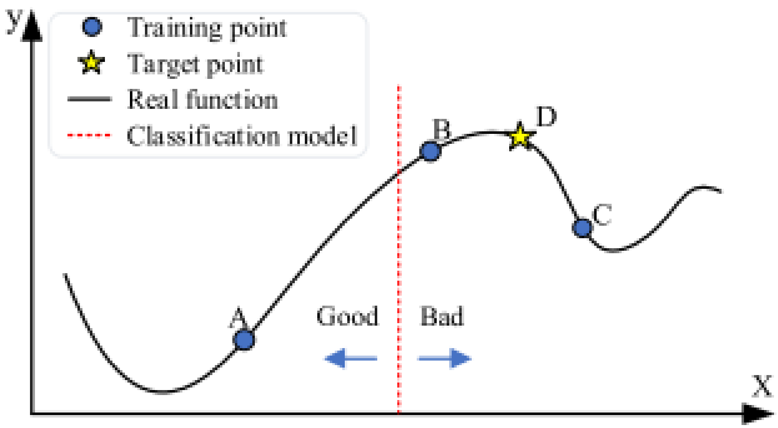
Random Forest ……

Any machine learning model which can be used for regression

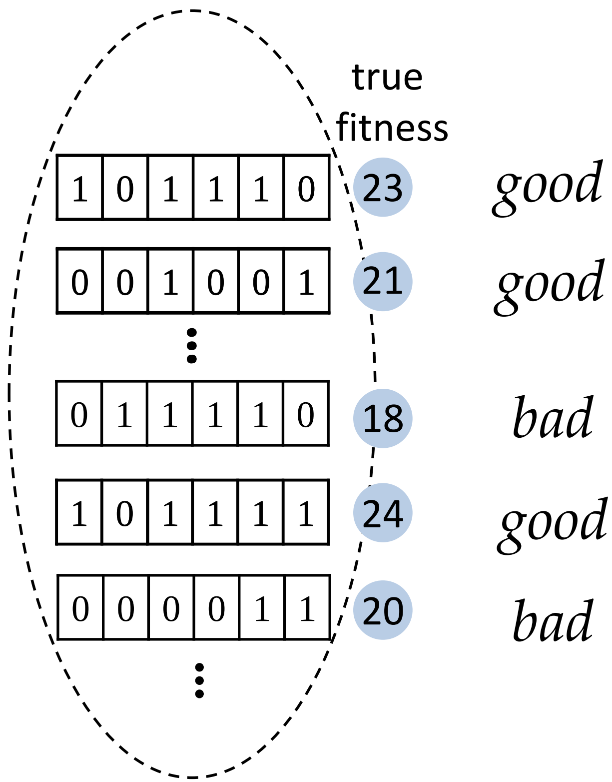
How to use surrogates - Preselection



Binary classification based preselection



How to get the training data?



Surrogate model:

Predict whether a solution is good or bad

Binary classification based preselection

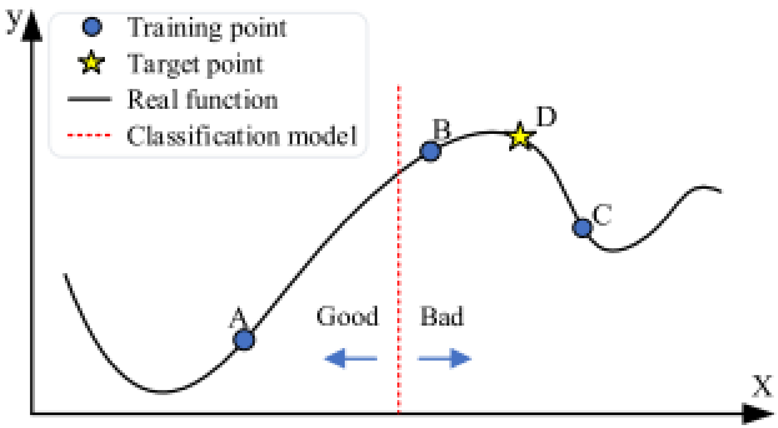
An example

from Aiming Zhou's talk

Algorithm 1: Framework of CPS-EA

// Initialization
Initialize the population $P = \{x^1, x^2, \cdots, x^N\}$;
// Main loop
while *termination condition is not satisfied* **do**
    // Sample definition
    Assign each $x \in P$ a label $l \in \{+1, -1\}$;
    // Model building
    Train a classifier $l = Bclass(x)$ based on the data set $\{< x, l > | x \in P\}$;
    foreach $x \in P$ **do**
        // Offspring generation
        Generate M candidate offspring individuals $Y = \{y^1, \cdots, y^M\}$;
        // Offspring solutions labeling and selection
        Predict their labels by the classifier;
        Set $V = \{y \in Y | Bclass(y) == +1\}$;
        Reset $V = Y$ if $V = \emptyset$;
        Randomly choose $y \in V$ as the offspring individual of x;
        // Environmental selection
        if $f(y) < f(x)$ **then**
            Set $x = y$;
        end
    end
end

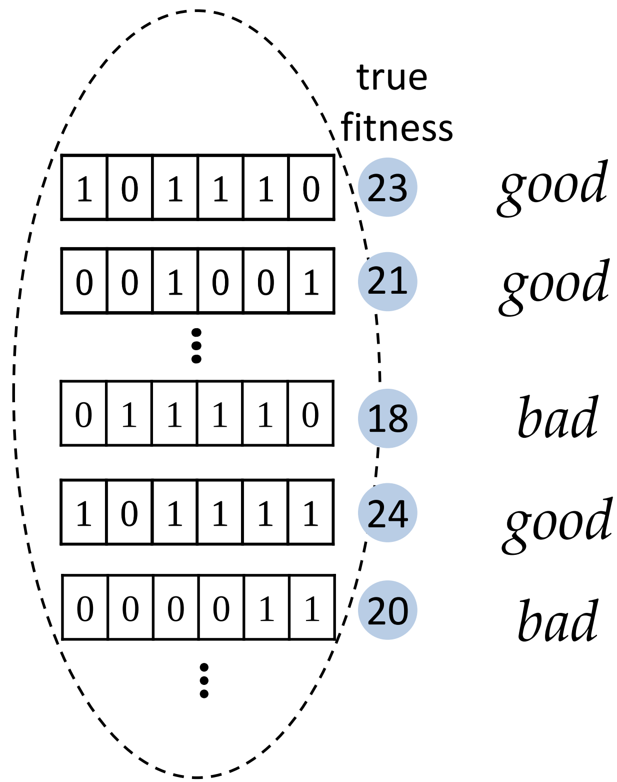
Fuzzy classification based preselection



Surrogate model:

Predict the probability of a solution being good

How to get the training data?



Fuzzy classification based preselection

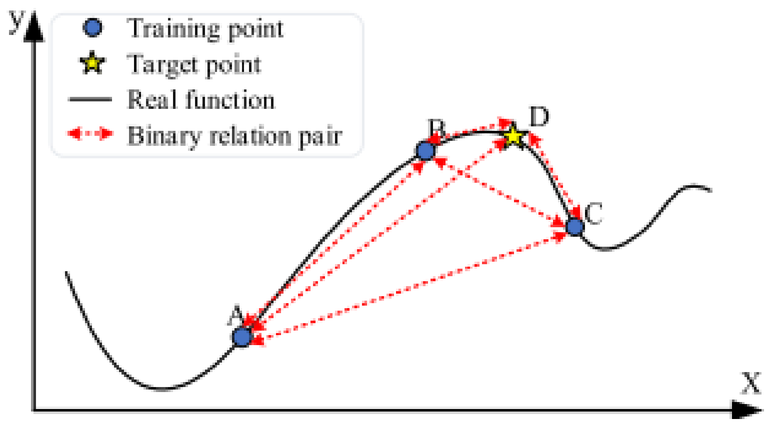
An example

Algorithm 1: FCPS-EA Framework

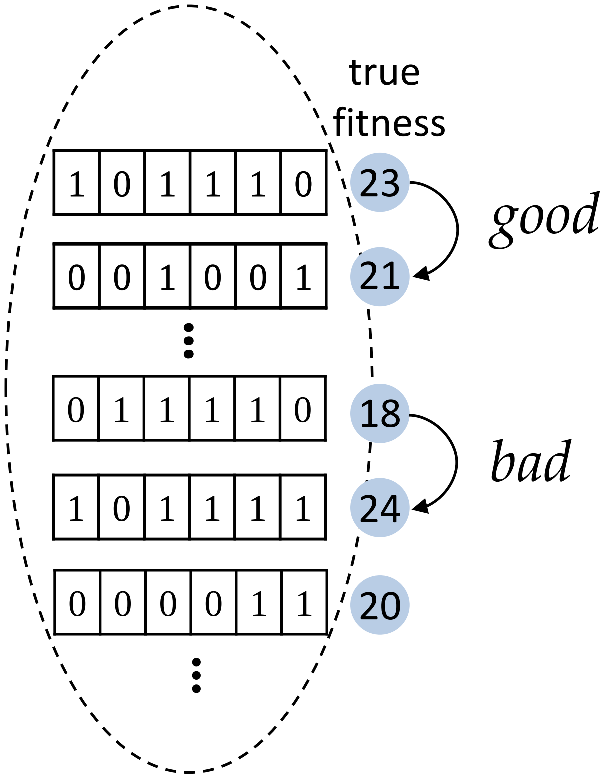
// Initialization
Initialize the population $P = \{x^1, x^2, \cdots, x^N\}$;
// Main loop
while *termination condition is not satisfied* **do**
    // Training Set definition
    Assign each individual $x \in P$ a label $l \in \{+1, -1\}$;
    // Model building
    Train a fuzzy classifier model $m = Fclass(x)$ based on the data set $\{< x, l > | x \in P\}$;
    foreach $x \in P$ **do**
        // Candidate solution generation
        Sample candidate solutions $Y = \{y^1, \cdots, y^M\}$;
        // Candidate solutions labeling and selection
        Predict the membership of $y \in Y$ by $m = Fclass(y)$;
        Let $V \subseteq Y$ contain candidate solutions with maximal membership degree belongs to 'promising' class;
        Randomly choose $y \in V$ as offspring x;
        // Environmental selection
        if $f(y) < f(x)$ **then**
            Set $x = y$;
        end
    end
end

[Zhou et al., AAAI'19]

Binary relation classification based preselection



How to get the training data?



Surrogate model:

Predict whether a solution is better than another one

Binary relation classification based preselection

An example [Hao et al., TEvC'20]

Algorithm 1 EA Framework With Preselection

Input: N: the population size;
M: the trial vectors size;
$MaxFES$: maximum number of function evaluations;
Output: the best found solution in the population;

Initialize the population $P = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ and evaluate them by the real objective function f;
Set number of fitness evaluations $fes = N$;
while $fes \leq MaxFES$ **do**
    Train a surrogate model based on P :

$$\boxed{\mathfrak{M} = \textbf{SurrogateTrain}(P);}$$

    for each $\mathbf{x} \in P$ **do**
        Generate M trail solutions $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M$ for the parent solutions $\mathbf{x}$ by generation operators;
        Choose the potential trial solution :

$$\boxed{\mathbf{u}^* = \textbf{PreSelection}(\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M\}, \mathfrak{M});}$$

        Evaluate $\mathbf{u}^*$ by the real objective function f;
        if $f(\mathbf{u}^*) < f(\mathbf{x})$ **then**
            Set $\mathbf{x} = \mathbf{u}^*$;
        end if
        Set $fes = fes + 1$;
    end for
end while
Output the best solution in P.

$$\mathfrak{M} = \text{ClassifierTrain}(\{\langle[\mathbf{x}_1, \mathbf{x}_2], l\rangle | \mathbf{x}_1, \mathbf{x}_2 \in P\})$$

$$\mathbf{u}^* = \arg \max_{u \in \{\mathbf{u}_1, \ldots, \mathbf{u}_M\}} \sum_{\mathbf{u}_i \neq \mathbf{u},\ i \in 1, \ldots, M} \text{Predict}([\mathbf{u}, \mathbf{u_i}], \mathfrak{M})$$

Advantage of surrogate models

Surrogate-assisted EA　　A variant of EA

Instance	n	Btree-CoDE		CoDE	
		Median	mean$_{std}$	Median	mean$_{std}$
LZG01	5	2.83e–160(+)	3.54e–159$_{9.83e-159}$ (+)	5.32e–75	1.72e–74$_{2.40e-74}$
	10	2.65e–61(+)	3.94e–60$_{8.90e-60}$ (+)	3.74e–33	6.50e–33$_{7.53e-33}$
	20	3.17e–29(+)	4.84e–29$_{8.90e-60}$ (+)	8.01e–17	7.61e–17$_{7.53e-33}$
	30	5.95e–20(+)	9.46e–20$_{1.42e-19}$ (+)	9.20e–12	8.57e–12$_{5.18e-12}$
LZG02	5	0.00e+00(∼)	0.00e+00$_{0.00e+00}$ (∼)	0.00e+00	0.00e+00$_{0.00e+00}$
	10	0.00e+00(+)	7.68e–29$_{2.36e-28}$ (+)	9.17e–15	6.61e–14$_{1.71e-13}$
	20	4.63e–03(+)	6.38e–03$_{7.19e-03}$ (+)	5.16e+00	5.07e+00$_{6.24e-01}$
	30	1.32e+01(+)	1.32e+01$_{1.21e+00}$ (+)	2.00e+01	1.99e+01$_{5.31e-01}$
LZG03	5	8.88e–16(∼)	8.88e–16$_{0.00e+00}$ (∼)	8.88e–16	8.88e–16$_{0.00e+00}$
	10	4.44e–15(∼)	4.09e–15$_{1.12e-15}$ (∼)	4.44e–15	4.44e–15$_{0.00e+00}$
	20	6.13e–14(+)	7.37e–14$_{5.71e-14}$ (+)	2.73e–08	2.87e–08$_{9.38e-09}$
	30	4.90e–10(+)	5.73e–10$_{2.82e-10}$ (+)	3.74e–06	4.27e–06$_{1.54e-06}$
LZG04	5	0.00e+00(∼)	0.00e+00$_{0.00e+00}$ (∼)	0.00e+00	0.00e+00$_{0.00e+00}$
	10	0.00e+00(+)	0.00e+00$_{0.00e+00}$ (+)	5.12e–10	3.83e–07$_{1.20e-06}$
	20	0.00e+00(+)	1.23e–03$_{3.89e-03}$ (∼)	4.24e–13	1.64e–10$_{2.51e-10}$
	30	0.00e+00(+)	9.86e–04$_{3.12e-03}$ (∼)	8.07e–10	2.00e–09$_{2.56e-09}$
+/–/∼	5	1/0/3	1/0/3		
	10	3/0/1	3/0/1		
	20	4/0/0	3/0/1		
	30	4/0/0	3/0/1		

Using surrogate models can improve the performance of EAs

Better/Worse/Similar, compared with CoDE

[Hao et al., ICIC'18]

Bayesian Optimization

Algorithm 1 BO Framework

Input: iteration budget T

Process:

let $D_0 = \emptyset$;

for $t = 1 : T$ **do**

  $\boxed{\boldsymbol{x}_t = \arg\max_{\boldsymbol{x} \in \mathcal{X}} acq(\boldsymbol{x});}$

  evaluate f at $\boldsymbol{x}_t$ to obtain y_t;

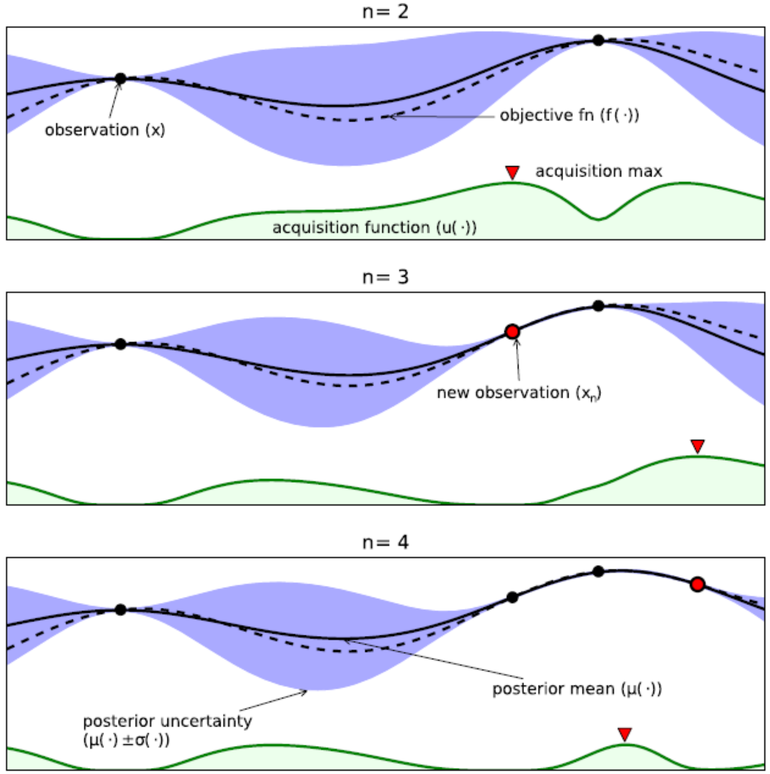
  augment the data $D_t = D_{t-1} \cup \{(\boldsymbol{x}_t, y_t)\}$ and update the GP model

end for

regards the f value at each data point as a random variable, and assumes satisfying a joint Gaussian distribution

Surrogate model: Gaussian process

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



Solid line: surrogate

Dotted line: true fitness

Bayesian Optimization

Typical acquisition functions

PI: prob. of a new x better than the best x^+ genereated-so-far

$$
\begin{aligned}
PI(x) &= P\big(f(x) \geq f(x^+)\big) \\
&= P\left(\frac{f(x) - u(x)}{\sigma(x)} \geq \frac{f(x^+) - u(x)}{\sigma(x)}\right) \\
&= 1 - \Phi\left(\frac{f(x^+) - u(x)}{\sigma(x)}\right) \\
&= \Phi\left(\frac{u(x) - f(x^+)}{\sigma(x)}\right)
\end{aligned}
$$

Φ: cumulative distribution function of standard Gaussian distribution

Bayesian Optimization

Typical acquisition functions

EI: expectation of improvement, i.e., $\max\{0, f(x) - f(x^+)\}$

⇩ ?

$$EI(x) = \begin{cases} \big(\mu(x) - f(x^+)\big)\Phi(Z) + \sigma(x)\varphi(Z) & \text{if } \sigma(x) > 0 \\ \max\{0,\ \mu(x) - f(x^+)\} & \text{if } \sigma(x) = 0 \end{cases}$$

$$Z = \frac{\mu(x) - f(x^+)}{\sigma(x)}$$

Φ: cumulative distribution function of standard Gaussian distribution

φ: probability density function of standard Gaussian distribution

Bayesian Optimization

Derivation of EI: Since $f(x) \sim N(\mu(x), \sigma(x)^2)$, for $\sigma(x) > 0$,

$$EI(x) = E[\max\{0, f(x) - f(x^+)\}\ ]$$

$$= \int_{f(x^+)}^{+\infty} \left(f(x) - f(x^+)\right) \frac{1}{\sqrt{2\pi}\sigma(x)} \exp\left(-\frac{\left(f(x) - \mu(x)\right)^2}{2\sigma(x)^2}\right) df(x)$$

$$= \int_{\frac{f(x^+)-\mu(x)}{\sigma(x)}}^{+\infty} \left(\sigma(x)Y + \mu(x) - f(x^+)\right) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{Y^2}{2}\right) dY \quad (Let\ Y = \frac{f(x) - \mu(x)}{\sigma(x)})$$

$$= \left(\mu(x) - f(x^+)\right)\Phi(Z) - \int_{\frac{f(x^+)-\mu(x)}{\sigma(x)}}^{+\infty} \sigma(x) \frac{1}{\sqrt{2\pi}}\, d \exp\left(-\frac{Y^2}{2}\right) \quad (Let\ Z = \frac{\mu(x)-f(x^+)}{\sigma(x)})$$

$$= \left(\mu(x) - f(x^+)\right)\Phi(Z) \text{-} (\sigma(x) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{Y^2}{2}\right))\Big|_{\frac{f(x^+)-\mu(x)}{\sigma(x)}}^{+\infty}$$

$$= \left(\mu(x) - f(x^+)\right)\Phi(Z) + \sigma(x)\varphi(Z)$$

Bayesian Optimization

Derivation of EI: Since $f(x) \sim N(\mu(x), \sigma(x)^2)$, for $\sigma(x) = 0$,

$$f(x) = \mu(x), almost\ surely,$$

$$\Downarrow$$

$$EI(x) = \max\{0,\ \mu(x) - f(x^+)\}$$

$$EI(x) = \begin{cases} \big(\mu(x) - f(x^+)\big)\Phi(Z) + \sigma(x)\varphi(Z) & \text{if } \sigma(x) > 0 \\ \max\{0,\ \mu(x) - f(x^+)\} & \text{if } \sigma(x) = 0 \end{cases}$$

$$Z = \frac{\mu(x) - f(x^+)}{\sigma(x)}$$

Φ: cumulative distribution function of standard Gaussian distribution

φ: probability density function of standard Gaussian distribution

Bayesian Optimization

Typical acquisition functions

UCB: weighted sum of posterior mean and variance

Exploitation

Exploration

$$UCB(x) = \mu(x) + \kappa \cdot \sigma(x)$$

Summary

- How to use surrogate models
  - Random strategy
  - Best strategy
  - Clustering based strategy
  - Uncertainty based strategy
  - Preselection

- Bayesian optimization

References

- Y. C. Jin, H. D. Wang and C. L. Sun. Data-driven Evolutionary Optimization. Chapter 5.
- E. Brochu, V. M. Cora and N. De Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv preprint arXiv:1012.2599, 2010.
- H. Hao, J. Y. Zhang and A. M. Zhou. A comparison study of surrogate model based preselection in evolutionary optimization. In: International Conference on Intelligent Computing. Springer, Cham, 2018, p. 717-728.
- H. Hao, J. Y. Zhang, X. F. Lu and A. M. Zhou, Binary relation learning and classifying for preselection in evolutionary algorithms. IEEE Transactions on Evolutionary Computation, 2020, 24(6), 1125-1139.

References

- Y. C. Jin. A comprehensive survey of fitness approximation in evolutionary computation. Soft computing, 2005, 9(1), 3-12.
- Y. C. Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. Swarm and Evolutionary Computation, 2011, 1(2), 61-70.
- B. Shahriari, K. Swersky, Z. Y. Wang, R. P. Adams and N. De Freitas, Taking the human out of the loop: A review of Bayesian optimization. Proceedings of the IEEE, 2015, 104(1), 148-175.
- J. Y. Zhang, A. M. Zhou, K. Tang, and G. X. Zhang, Preselection via classification: A case study on evolutionary multiobjective optimization. Information Sciences, 2018, 465, 388-403.