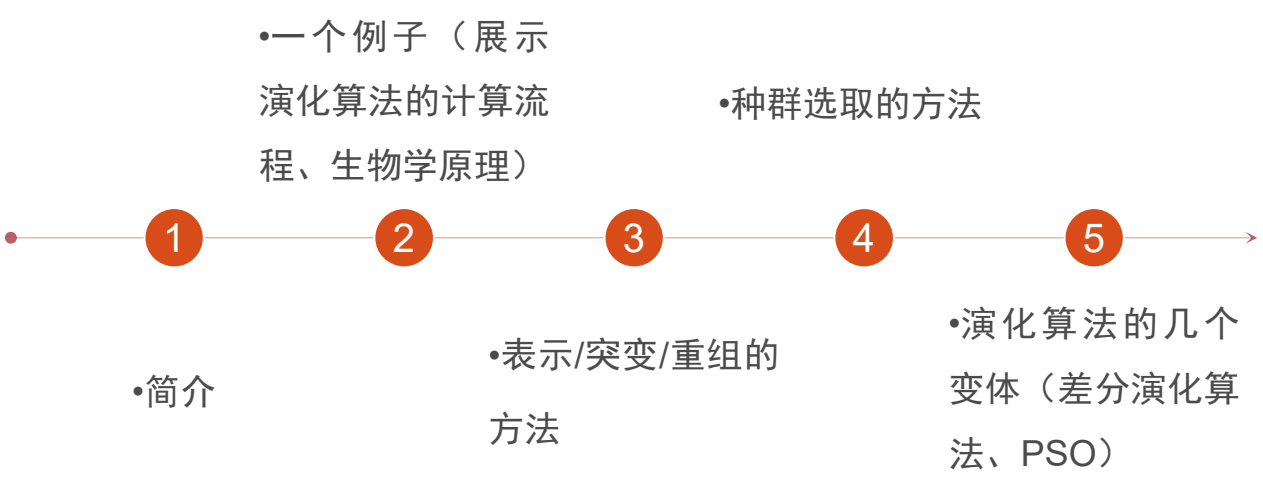
演化算法

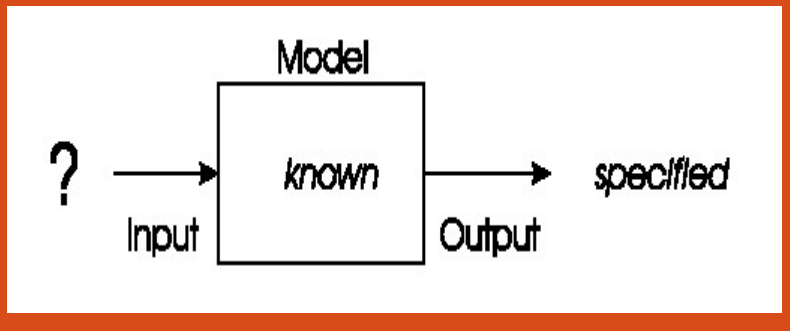
2016年7月28日 王卯宁

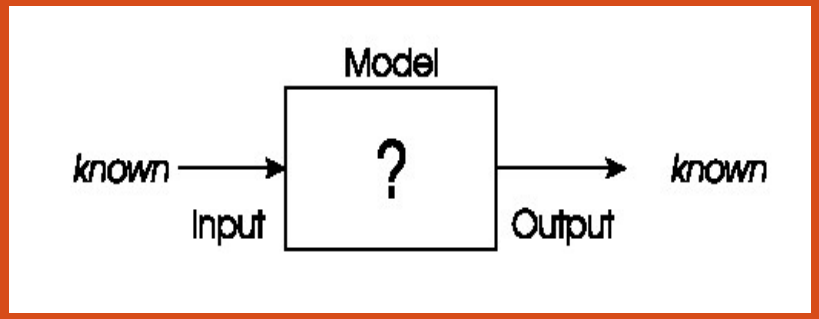
内容


•一个例子（展示演化算法的计算流程、生物学原理）
•种群选取的方法
1
2
3
4
5
•简介
•表示/突变/重组的方法
•演化算法的几个变体（差分演化算法、PSO）

简介-可以用演化算法来求解的问题

组合最优化问题
-背包问题
-8棋子放置问题
-旅行商人问题
-天线设计问题
Learning问题
-优化模型参数
-模型选取





简介-演化算法的思想

- 一种生成-测试型算法：

-生成：用种群中个体的变异和重组来生成新的个体

-测试：根据适者生存原理选取能够存留、保存特征的个体

- The population provides the algorithm with a means of defining a nonuniform probability distribution function (p.d.f.) governing the generation of new points from S. This p.d.f. reflects possible interactions between points in S which are currently represented in the population.
- This potentially complex p.d.f. contrasts with the globally uniform distribution of blind random search, and the locally uniform distribution used by many other stochastic algorithms such as simulated annealing and various hill-climbing algorithms.
- 用物种进化的思想来逼近潜在的复杂分布

一个例子

How Does the Simple EA Work

Let's use the simple EA to maximise the function $f(x) = x^2$ with x in the *integer* interval $[0, 31]$, i.e., $x = 0, 1, \cdots, 30, 31$.

The first step of EA applications is *encoding* (i.e., the representation of chromosomes). We adopt binary representation for integers. Five bits are used to represent integers up to 31. Assume that the population size is 4.

1. Generate initial population at random, e.g., 01101, 11000, 01000, 10011. These are *chromosomes* or *genotypes*.
2. Calculate fitness value for each individual.
   (a) Decode the individual into an integer (called *phenotypes*),

$$01101 \rightarrow 13, 11000 \rightarrow 24, 01000 \rightarrow 8, 10011 \rightarrow 19;$$

一个例子

(b) Evaluate the fitness according to $f(x) = x^2$,

$$13 \to 169, 24 \to 576, 8 \to 64, 19 \to 361.$$

3. Select two individuals for crossover based on their fitness. If roulette-wheel selection is used, then

$$p_i = \frac{f_i}{\sum_j f_j}.$$

Two offspring are often produced and added to an intermediate population. Repeat this step until the intermediate population is filled. In our example,

$$p_1(13) = 169/1170 = 0.14 \quad p_2(24) = 576/1170 = 0.49$$
$$p_3(8) = 64/1170 = 0.06 \quad p_4(19) = 361.1170 = 0.31$$

Assume we have $crossover(01101, 11000)$ and $crossover(10011, 11000)$. We may obtain offspring 0110 0 and

一个例子

1100 1 from $crossover(01101, 11000)$ by choosing a random crossover point at 4, and obtain 10 000 and 11 011 from $crossover(10011, 11000)$ by choosing a random crossover point at 2. Now the intermediate population is

$$01100, 11001, 10000, 11011$$

4. Apply mutation to individuals in the intermediate population with a *small* probability. A simple mutation is bit-flipping. For example, we may have the following new population $P(1)$ after random mutation:

$$0110\mathbf{1}, 11001, \mathbf{0}0000, 11011$$

5. Goto Step 2 if not stop.

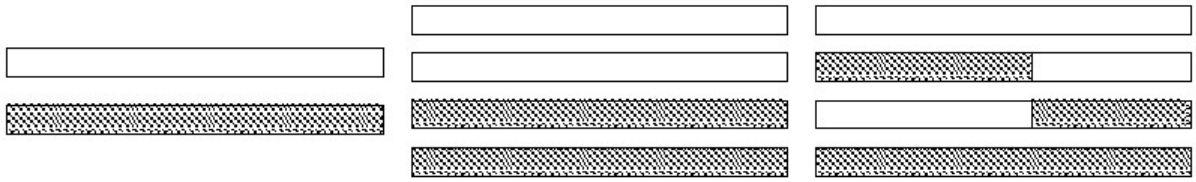
一个例子

String no.	Initial population	x Value	Fitness $f(x) = x^2$	$Prob_i$	Expected count	Actual count
1	0 1 1 0 1	13	169	0.14	0.58	1
2	1 1 0 0 0	24	576	0.49	1.97	2
3	0 1 0 0 0	8	64	0.06	0.22	0
4	1 0 0 1 1	19	361	0.31	1.23	1
Sum			1170	1.00	4.00	4
Average			293	0.25	1.00	1
Max			576	0.49	1.97	2

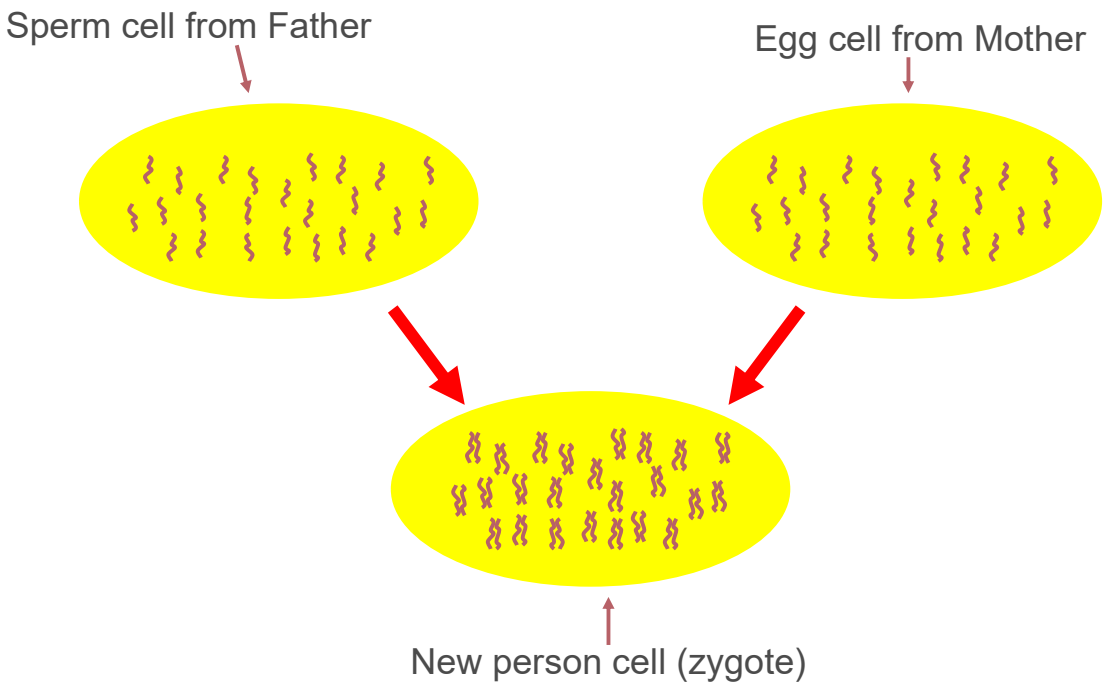
String no.	Mating pool	Crossover point	Offspring after xover	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 | 1	4	0 1 1 0 0	12	144
2	1 1 0 0 | 0	4	1 1 0 0 1	25	625
2	1 1 | 0 0 0	2	1 1 0 1 1	27	729
4	1 0 | 0 1 1	2	1 0 0 0 0	16	256
Sum					1754
Average					439
Max					729

String no.	Offspring after xover	Offspring after mutation	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 0	1 1 1 0 0	26	676
2	1 1 0 0 1	1 1 0 0 1	25	625
2	1 1 0 1 1	1 1 0 1 1	27	729
4	1 0 0 0 0	1 0 1 0 0	18	324
Sum				2354
Average				588.5
Max				729

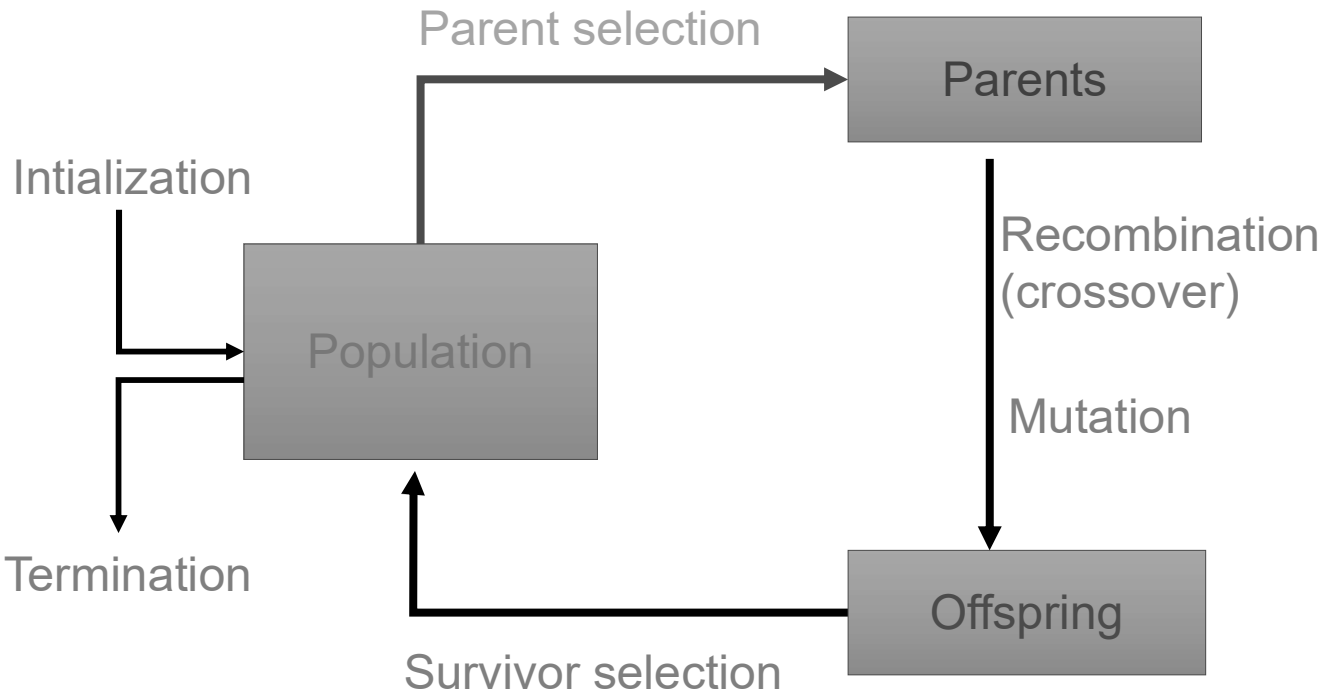
基因重组



新个体的形成


Sperm cell from Father
Egg cell from Mother
New person cell (zygote)

演化算法的流程


Parent selection
Parents
Intialization
Recombination
(crossover)
Population
Mutation
Termination
Offspring
Survivor selection

演化算法是指一个算法族

A

不同的演化算法区别于：

- 历史背景
- 表示形式
- 变异/重组的方法
- 个体生存的选取方法

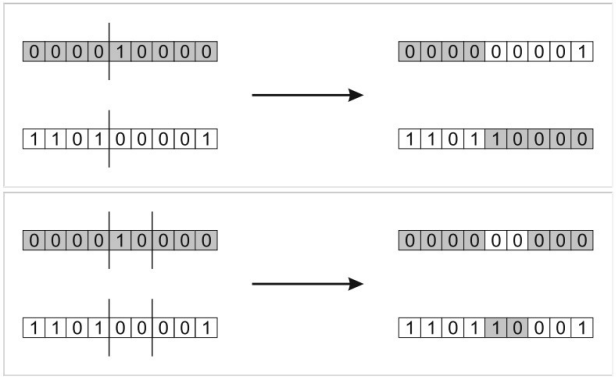
表示/突变/重组的方法　1.二元序列表示

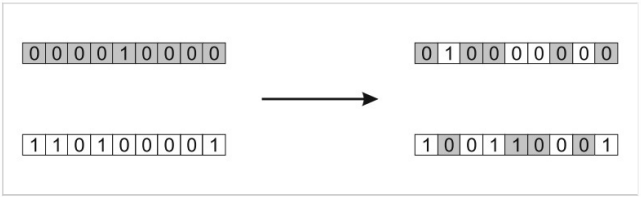
例子：求解背包问题

变异：

Fig. 4.1. Bitwise mutation for binary encodings

重组：





表示/突变/重组的方法　2.实数/浮点数序列表示

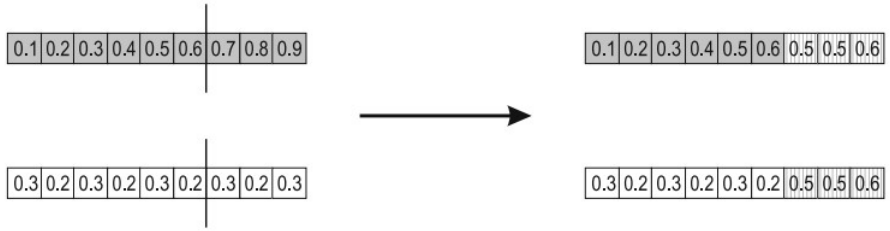
例子：参数选取问题

变异：　$\langle x_1, \ldots, x_n \rangle \to \langle x'_1, \ldots, x'_n \rangle,$　　$p(\Delta x_i) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(\Delta x_i - \xi)^2}{2\sigma^2}}$

改进形式：自适应的选取 σ

重组：　Child 1: $\langle x_1, \ldots, x_k, \alpha \cdot y_{k+1} + (1-\alpha) \cdot x_{k+1}, \ldots, \alpha \cdot y_n + (1-\alpha) \cdot x_n \rangle.$

Child 2 is analogous, with x and y reversed

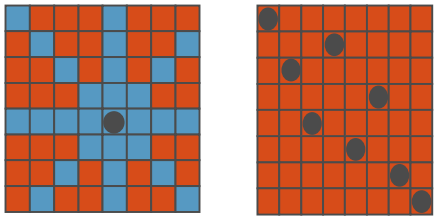


表示/突变/重组的方法　3.整数序列表示

例子：刚才展示的整数约束条件的最优化问题

变异、重组都可以看成是二元序列或离散的实数序列的变体

表示/突变/重组的方法　4.置换表示

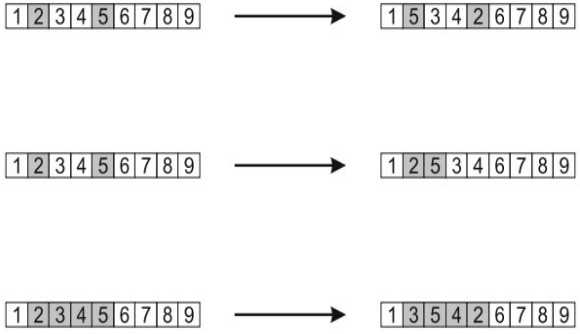


例子：

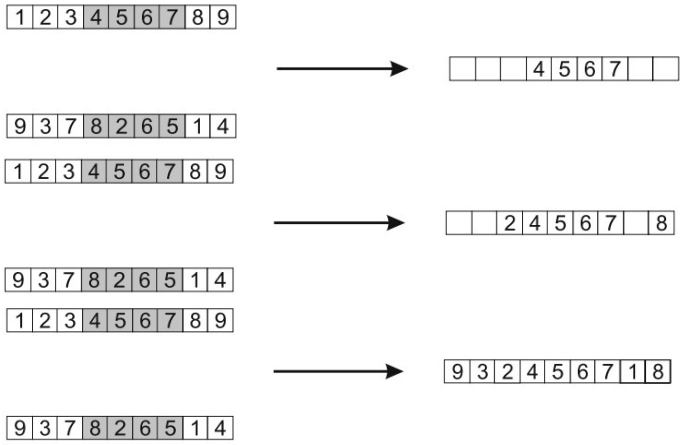
-N*N的棋盘放置N枚棋子，使得每行、每列、每个对角线都只有一个棋子

-旅行商人问题：走过N个城市，每个城市只经过一次，求最短的路线

变异：



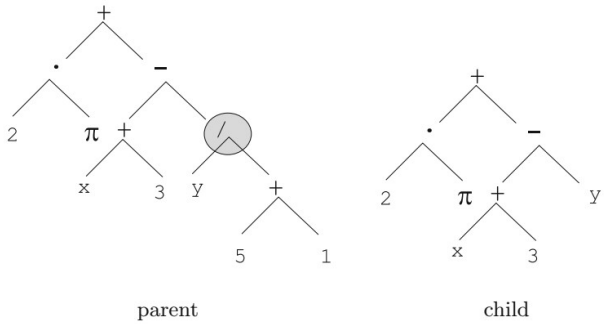
重组：



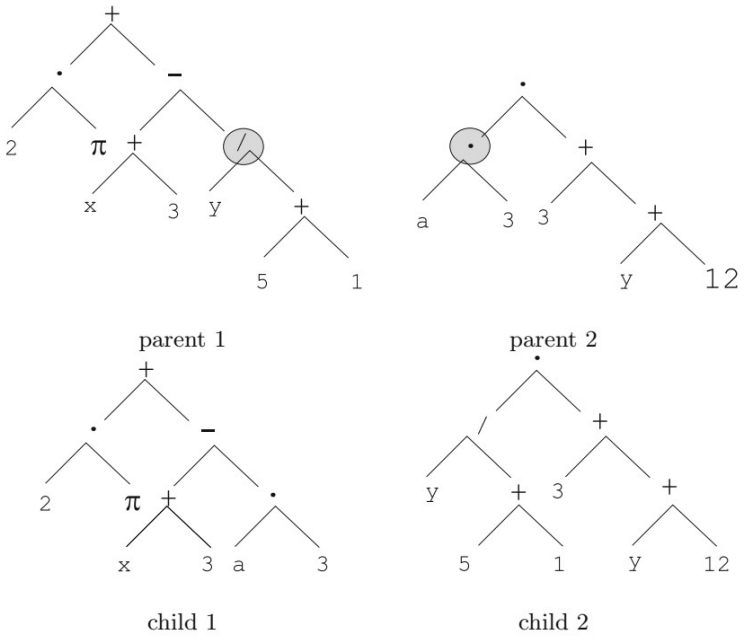
表示/突变/重组的方法　5.树表示

例子：找最优的表达式

变异：



重组：



树表示的一些例子

- an arithmetic formula:

$$2 \cdot \pi + ((x + 3) - \frac{y}{5 + 1}),$$

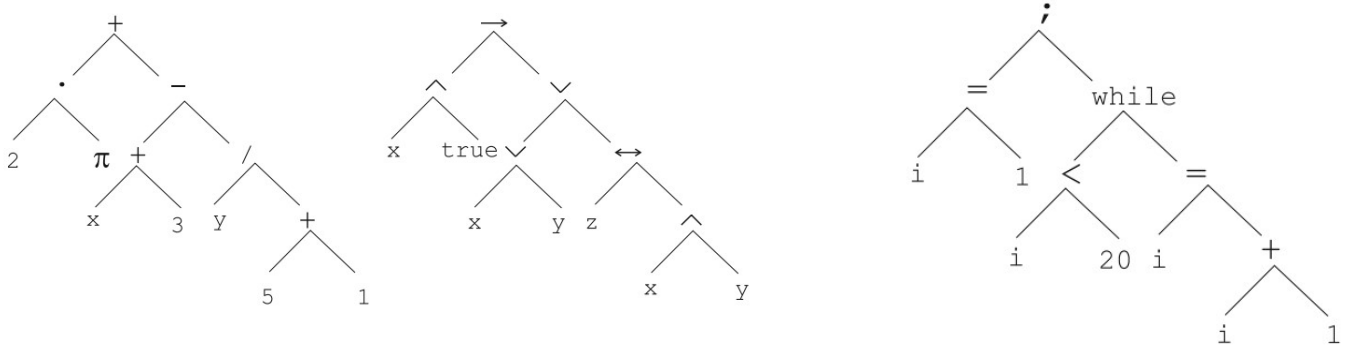
- a logical formula:

$$(x \wedge true) \rightarrow ((x \vee y) \vee (z \leftrightarrow (x \wedge y))),$$

- the following program:

```
i = 1;
while (i < 20)
{
        i = i+1;
}
```

树表示的一些例子



种群控制的方法

种群控制包括：

-一个个体可以被选为parent的概率
fitness proportional selection (FPS)：$P_{FPS}(i) = f_i / \sum_{j=1}^{\mu} f_j$
改进形式：$f'(x) = max(f(x) - (\bar{f} - c \cdot \sigma_f), 0)$

-一个新生成的个体被保留在群体中的概率
全部替换上一代种群
部分替代上一代种群
保持物种多样性 $F'(i) = \frac{F(i)}{\sum_j sh(d(i,j))}$

演化算法的变体-差分演化算法

- 变异：$\overline{x}' = \overline{x} + \overline{p}$

$$\overline{p} = F \cdot (\overline{y} - \overline{z})$$

演化算法的变体-PSO（Particle Swarm Optimisation）

- 基因为 $\langle \overline{x}, \overline{p} \rangle$
- 每个个体都发生变异 $\overline{x}' = \overline{x} + \overline{p}'$

$$\overline{v}' = w \cdot \overline{v} + \phi_1 U_1 \cdot (\overline{y} - \overline{x}) + \phi_2 U_2 \cdot (\overline{z} - \overline{x})$$

总结

- 演化算法可以用来求解优化问题，包括对结构和参数的优化
- 演化算法是指一个算法族，针对不同结构的问题，应建立不同的表示，采用相应的变异重组方式
- 演化算法没有完整的理论证明过程，但是对算法中的部分步骤可以进行分析
- 演化算法对生物进化过程的模拟并不精确，但这是一种使得物种进化的方式；改进演化算法的方法也不仅限于生物学