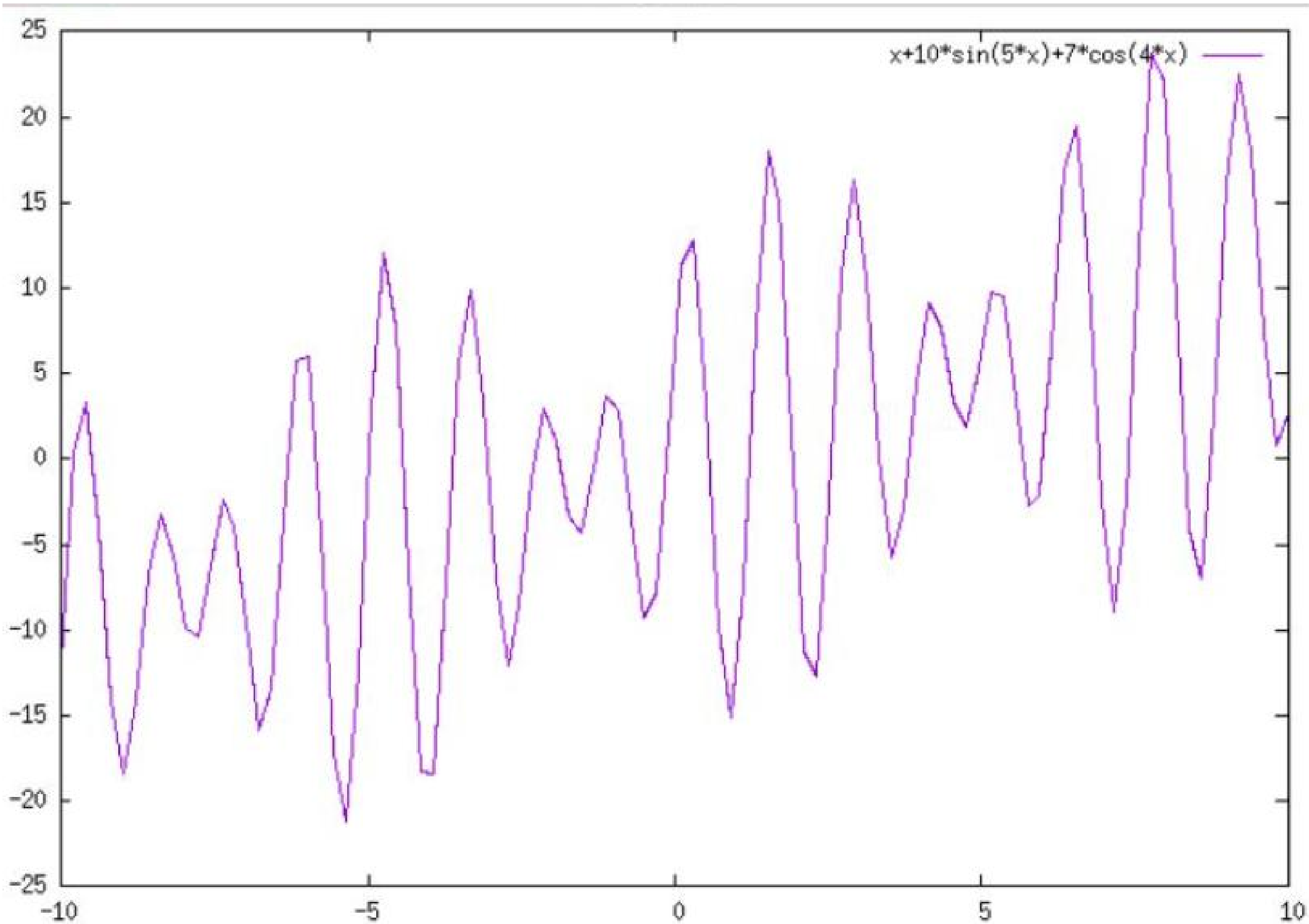
遗传算法

(Genetic Algorithm,GA)

问题背景：

求解函数 f(x) = x + 10*sin(5*x) + 7*cos(4*x) 在区间[0,9]的最大值。



3 遗传算法

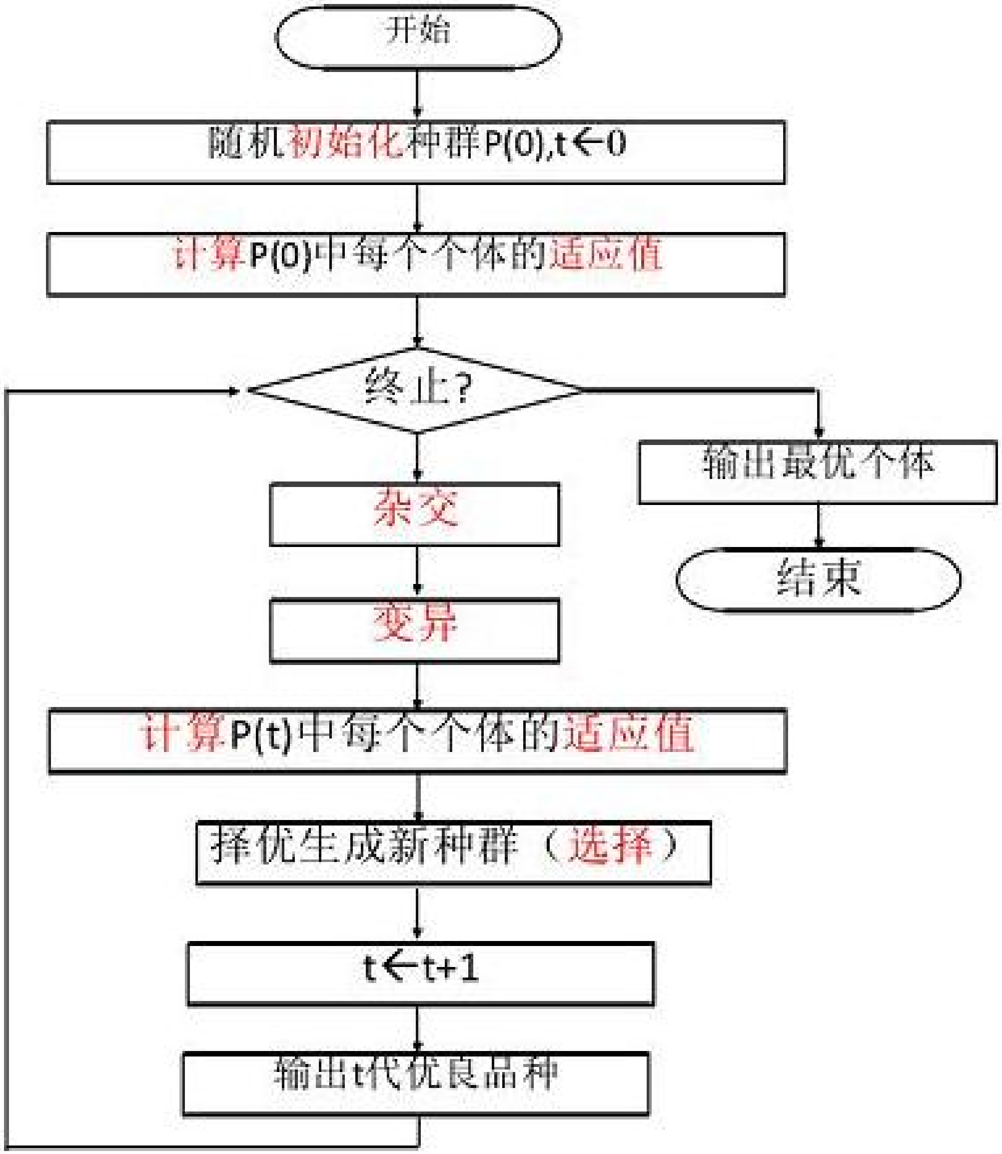
①该算法是根据大自然中生物体进化规律设计提出。

②模拟达尔文自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法

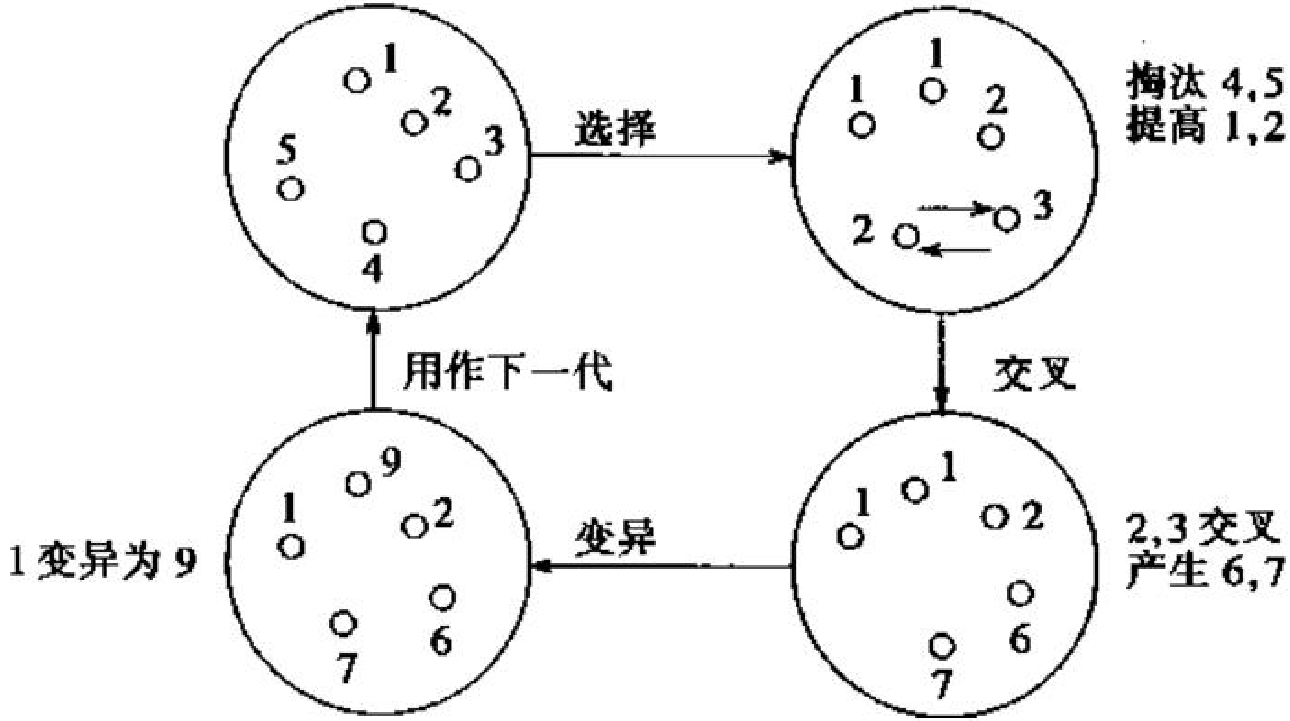
③该算法通过数学的方式,利用计算机仿真运算,将问题的求解过程转换成类似生物进化中的染色体基因的交叉、变异等过程。

④在求解较为复杂的组合优化问题时,相对一些常规的优化算法,通常能够较快地获得较好的优化结果

3.1 遗传算法的基本思路



遗传算法执行过程


1
2
3
5
4
选择
1
1
2
3
2
淘汰 4,5
提高 1,2
用作下一代
交叉
9
1
2
6
7
1变异为9
变异
1
1
2
6
7
2,3 交叉
产生 6,7

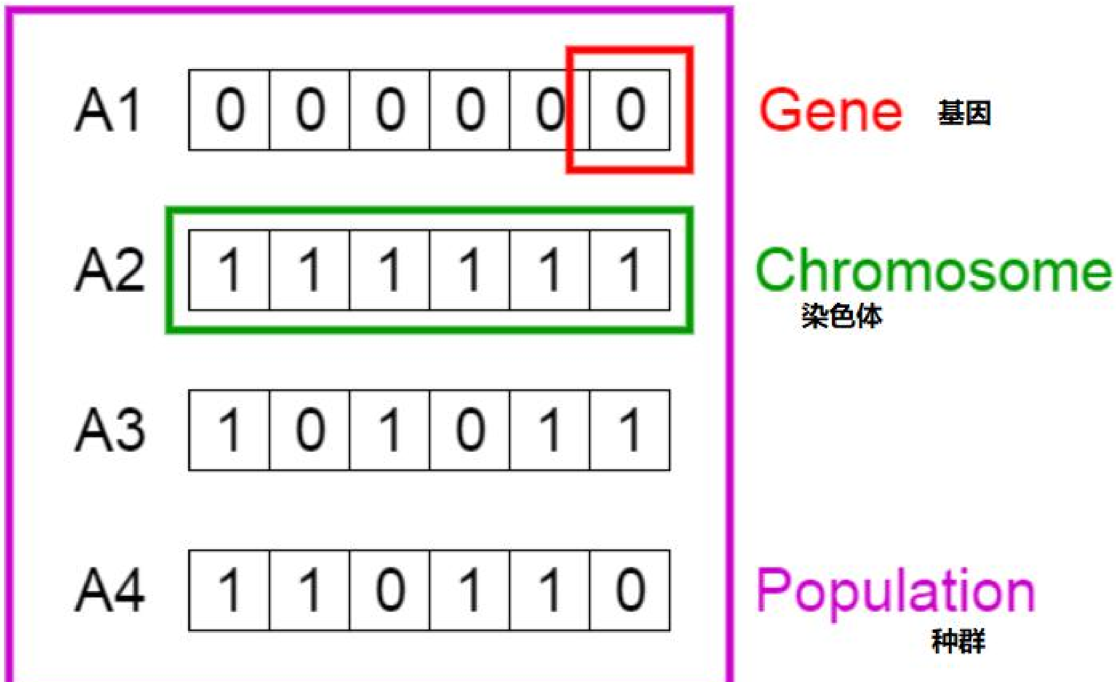
3.2 遗传算法基本概念

个体与种群(群体)

- 个体：模拟生物个体， 个体也即可行解，对应染色体
- 种群(population)：模拟生物种群，由若干个体组成的群体， 可行解集。

染色体与基因

染色体（chromosome）：
可行解的编码表示(二进制编码)。
可行解编码的分量，称为基因（gene）。



适应度与适应度函数

- 适应度(fitness)：生物个体对环境的适应能力。
- 适应度函数(fitness function)：遗传算法用来评价个体(解)优劣的数学函数。

适应度函数值越大，解的质量越好

遗传操作

亦称遗传算子(genetic operator)，关于染色体的运算。

遗传算法中有三种遗传操作:

- 选择(selection)
- 交叉(crossover，亦称交换、交配或杂交)
- 变异(mutation，亦称突变)

选择算子

遗传算法使用选择运算实现对个体进行优胜劣汰操作：

① 适应度高的个体被遗传到下一代群体中的概率大；

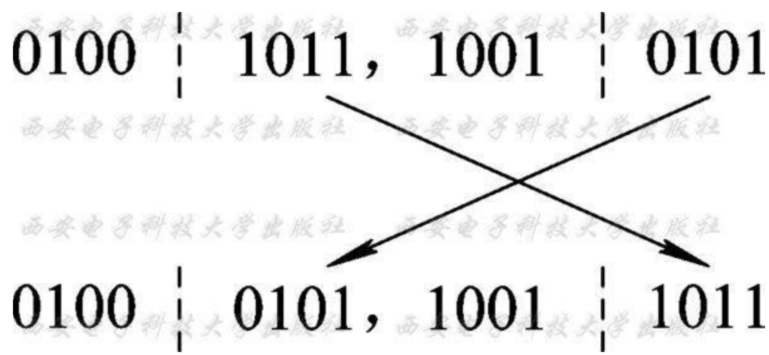
② 不产生新个体

经典的选择算子采用轮盘赌选择方法。

交叉

互换两个染色体某些位上的基因, 是产生新个体的主要方法

例如, 设染色体 s_1=01001011, s_2=10010101, 交换其后4位基因, 即



s_1'=01000101, s_2'=10011011

可以看做是原染色体s_1和s_2的子代染色体。

变异

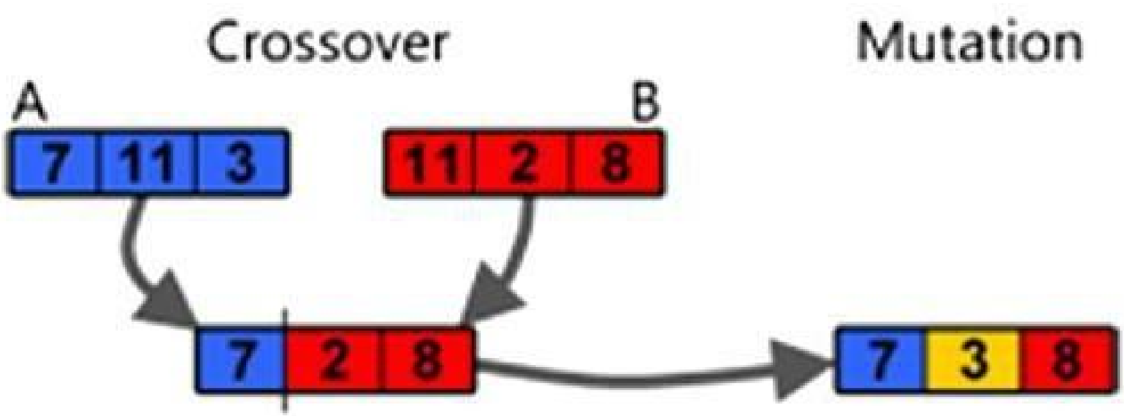
改变染色体某个(些)位上的基因。

例如，设染色体 s=11001101，将其第三位上的0变为1，

即

$$s=11\underline{0}01101 \rightarrow 11\underline{1}01101 = s'$$。

s'也可以看做是原染色体s的子代染色体。

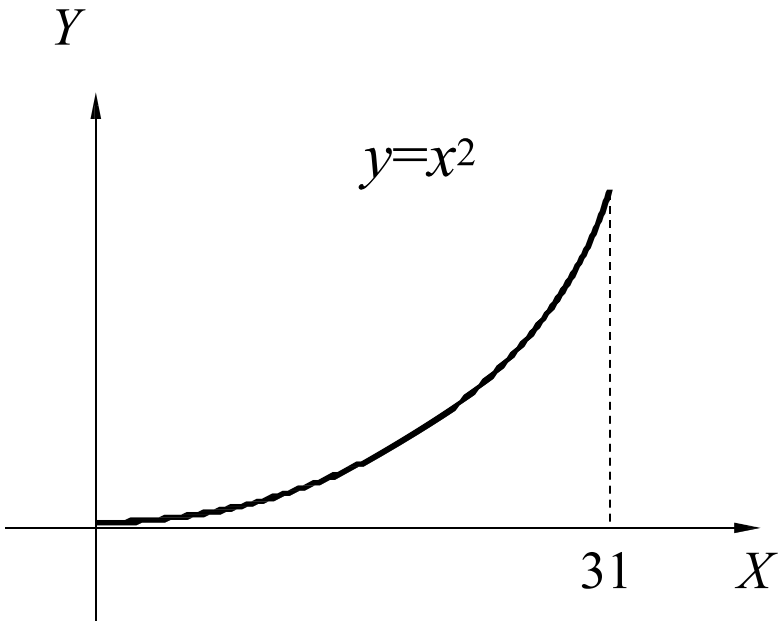


生物遗传概念在遗传算法中的对应关系

生物遗传概念	遗传算法中的作用
适者生存	在算法停止时，最优目标值的解有最大的可能被保留
个体(individual)	解
染色体(chromosome)	解的编码（字符串，向量等）
基因(gene)	解中每一分量的特征（如各分量的值）
适应性(fitness)	适应函数值
群体(population)	选定的一组解（其中解的个数为群体的规模）
种群(reproduction)	根据适应函数值选取的一组解
交配(crossover)	通过交配原则产生一组新解的过程
变异(mutation)	编码的某一个分量发生变化的过程

4 遗传算法应用举例

例1 利用遗传算法求解区间［0,31］上的二次函数$y=x^2$的最大值。



分析

原问题可转化为：

- 在区间［0, 31］中搜索能使y取最大值的点a的问题。
- 则［0, 31］ 中的点x即是个体,
- 函数值$f(x)$可作为x的适应度函数，
- 区间［0, 31］是一个(解)空间 。

给出个体x的适当染色体编码, 该问题可用遗传算法解决。

解：

(1) **设定种群规模, 编码染色体，产生初始 种群。**

将种群规模设定为4；用5位二进制数编码个体；

取下列个体组成初始种群S_1:

s_1= 13 (01101), s_2= 24 (11000)

s_3= 8 (01000), s_4= 19 (10011)

（2）定义适应度函数,取适应度函数：$f(x)=x^2$

(3) **计算各代种群中的各个体的适应度，并对其进行遗传操作,直到适应度最高的个体出现为止。**

(即31（11111）)

首先计算种群S_1中各个体的适应度$f(s_i)$

$s_1 = 13(01101)$, $s_2 = 24(11000)$

$s_3 = 8(01000)$, $s_4 = 19(10011)$

易求得

$$f(s_1) = f(13) = 13^2 = 169$$

$$f(s_2) = f(24) = 24^2 = 576$$

$$f(s_3) = f(8) = 8^2 = 64$$

$$f(s_4) = f(19) = 19^2 = 361$$

再计算种群S_1中各个体的选择概率。 选择概率的计算公式为：

$$P(x_i)=\frac{f(x_i)}{\sum_{j=1}^{N} f(x_j)}$$

由此可求得

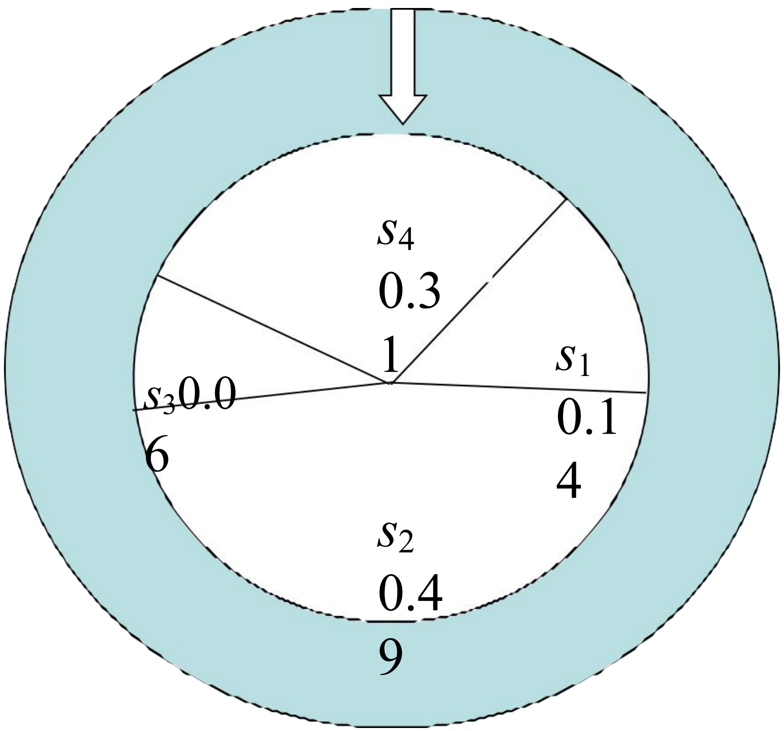
$$P(s_1)=P(13)=0.14$$

$$P(s_2)=P(24)=0.49$$

$$P(s_3)=P(8)=0.06$$

$$P(s_4)=P(19)=0.31$$

赌轮选择法



赌轮选择示意

① 在［0, 1］区间内产生一个均匀分布的随机数r。

② 若$r \le q_1$,则染色体x_1被选中。

③ 若$q_{k-1} < r \le q_k (2 \le k \le N)$, 则染色体$x_k$被选中。其中$q_i$称为染色体$x_i$ (i=1, 2, …, n)的**积累概率**, 其计算公式为

$$q_i = \sum_{j=1}^{i} P(x_j)$$

选择

设从区间［0, 1］中产生4个随机数如下:

$$r_1 = 0.450126,\ r_3 = 0.572496,$$

$$r_2 = 0.110347，\ r_4 = 0.98503$$

染色体	适应度	选择概率	积累概率	选中次数
s_1=01101	169	0.14	0.14	1
s_2=11000	576	0.49	0.63	2
s_3=01000	64	0.06	0.69	0
s_4=10011	361	0.31	1.00	1

于是，经选择操作后得群体 S_1：

s_1'=11000（24），　　s_2'=01101（13）

s_3'=11000（24），　　s_4'=10011（19）

交叉

设交叉率p_c=100%，即S_1中的全体个体都参加交叉运算。

设s_1'与s_2'配对，s_3'与s_4'配对。

分别交换后两位基因，得新染色体集S_2：

s_1'=11001（25），　　s_2'=01100（12）

s_3'=11011（27），　　s_4'=10000（16）

变异

设变异率p_m=0.001， 则群体S_1共有

5×4×0.001=0.02 位基因可以变异。

0.02位显然不足1位，所以本轮遗传操作

不做变异。

于是，得到第二代种群S_2:

s_1=11001（25）, s_2=01100（12）

s_3=11011（27）, s_4=10000（16）

群体 S_1:

s_1'=11000（24）， s_2'=01101（13）

s_3'=11000（24）， s_4'=10011（19）

第二代种群S_2中各染色体的情况

染色体	适应度	选择概率	积累概率	估计的 选中次数
s_1=11001	625	0.36	0.36	1
s_2=01100	144	0.08	0.44	0
s_3=11011	729	0.41	0.85	2
s_4=10000	256	0.15	1.00	1

假设这一轮选择操作中，新种群S_2中的

4个染色体都被选中，则得到群体：

s_1'=11001（25），　　s_2'= 01100（12）

s_3'=11011（27），　　s_4'= 10000（16）

做交叉运算，让s_1'与s_2'，s_3'与s_4'分别交换后三基因，得

s_1'=11100（28）, s_2'= 01001（9）

s_3'=11000（24）, s_4'= 10011（19）

这一轮仍然不会发生变异。

于是，得第三代种群S_3：

s_1=11100（28）, s_2=01001（9）

s_3=11000（24）, s_4=10011（19）

第三代种群*S*中各染色体的情况

染色体	适应度	选择概率	积累概率	估计的选中次数
s_1=11100	784	0.44	0.44	2
s_2=01001	81	0.04	0.48	0
s_3=11000	576	0.32	0.80	1
s_4=10011	361	0.20	1.00	1

设这一轮的选择结果为：

s_1'=11100（28），s_2'=11100（28）

s_3'=11000（24），s_4'=10011（19）

做交叉运算，让s_1'与s_4'，s_2'与s_3'分别交换后两位基因，得

s_1''=11111（31），s_2''=11100（28）

s_3''=11000（24），s_4''=10000（16）

这一轮仍然不会发生变异。

于是，得第四代种群S_4：

s_1=11111（31）， s_2=11100（28）

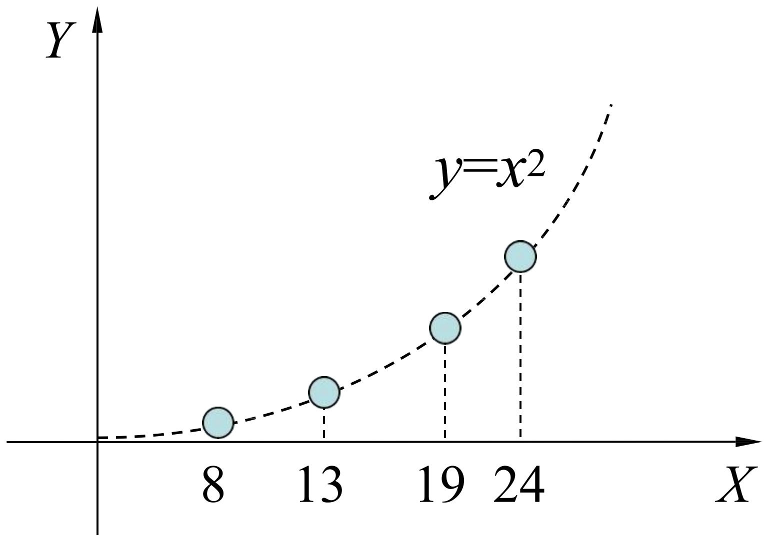
s_3=11000（24），s_4=10000（16）

显然，在这一代种群中已经出现了适应度最高的
个体 s_1=11111。
则遗传操作终止， 将个体“11111”作为最终结果输出。

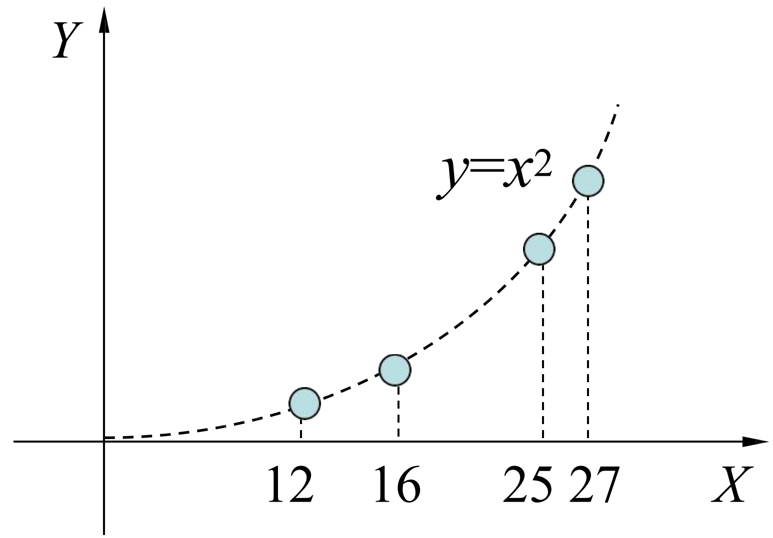
将染色体“11111”解码， 即得所求的最优解：31。

将31代入函数$y=x^2$中，即得原问题的解，

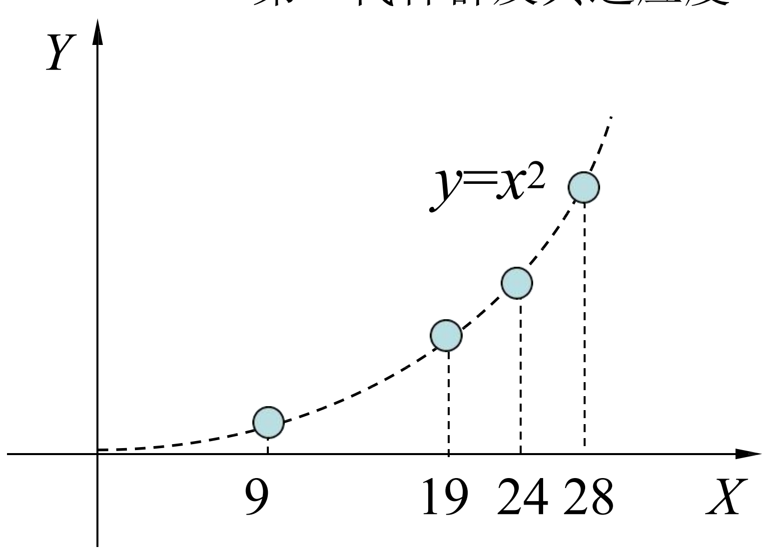
即 函数$y=x^2$的最大值为961。



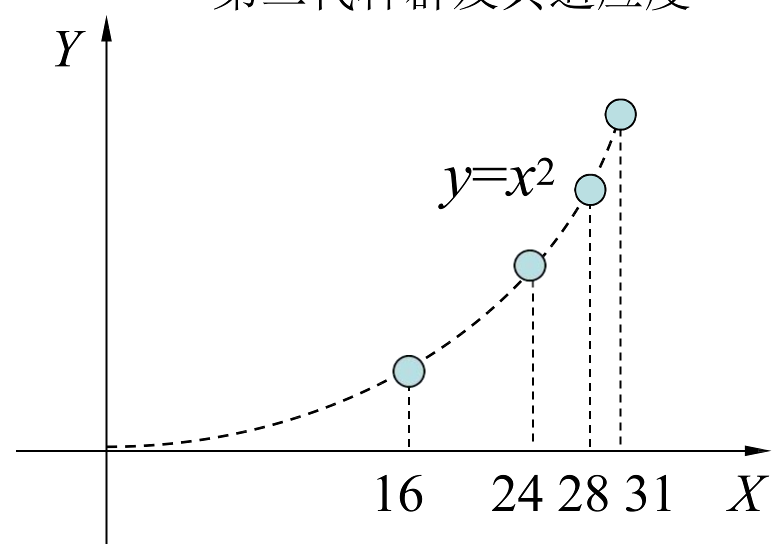
第一代种群及其适应度



第二代种群及其适应度



第三代种群及其适应度



第四代种群及其适应度

4 遗传算法主要步骤

- 首先初始化个体与种群
- 其次对优化问题的解进行编码
- 第三是适应函数的构造和应用
- 第四是染色体的结合:遗传算子（选择、交叉、变异）
- 最后运行参数。

（1）NP ： 种群规模

（2）G ： 遗传运算的终止迭代条件

（3）P_c ： 交叉概率

（4）P_m ： 变异概率

适应度函数及其尺度变换

- **适应度函数的重要性**

  适应度函数的选取直接影响遗传算法的收敛速度以及能否找到最优解。

  一般而言，适应度函数是由目标函数变换而成的，对目标函数值域的某种映射变换称为适应度的尺度变换（**fitness scaling**）。

◆ 几种常见的适应度函数

✓ **直接转换**

若目标函数为最大化问题：Fit (f (x))= f (x)

若目标函数为最小化问题：Fit (f (x))= - f (x)

✓ **界限构造法1**

若目标函数为最大化问题：

$$Fit(f(x)) = \begin{cases} f(x) - c_{\min}, & f(x) > c_{\min} \\ 0, & \text{其他} \end{cases}$$

式中，$c_{\min}$为$f(x)$的最小估计值。

若目标函数为最小化问题：

- **适应度函数的幂函数变换法**

  $f' = f^{k}$

  k与所求优化相关

- **适应度函数的指数变换法**

  $f' = e^{-af}$

  a决定了复制的强制性，其值越小，复制的强制性就越趋向于那些具有最大适应性的个体。

参数说明

（1）NP：种群规模

群体规模将影响遗传优化的最终结果以及遗传算法的执行效率。当群体规模*NP*太小时，遗传优化性能一般不会太好。采用较大的群体规模可以减小遗传算法陷入局部最优解的机会，但较大的群体规模意味着计算复杂度较高。一般 *NP*取10～200。

（2）G：遗传算法终止迭代次数

终止进化代数*G*是表示遗传算法运行结束条件的一个参数，它表示遗传算法运行到指定的进化代数之后就停止运行，并将当前群体中的最佳个体作为所求问题的最优解输出。一般视具体问题而定，*G*的取值可在100～1000之间。

（3）P_c：交叉概率

交叉概率P_c控制着交叉操作被使用的频度。较大的交叉概率可以增强遗传算法开辟新的搜索区域的能力，但高性能的模式遭到破坏的可能性增大；若交叉概率太低，遗传算法搜索可能陷入迟钝状态。一般P_c取0.25～1.00。

（4）P_m： 变异概率

变异在遗传算法中属于辅助性的搜索操作，它的主要目的是保持群体的多样性。一般低频度的变异可防止群体中重要基因的可能丢失，高频度的变异将使遗传算法趋于纯粹的随机搜索。通常P_m取0.001～0.1。

基本遗传算法

步1 给定种群规模N，交叉率P_c和变异率P_m，

适应度函数 $f(x)$， 迭代次数T； t=0；

步2 随机产生N个个体$s_1, s_2, \ldots, s_N$，组成初始种群$S=\{s_1, s_2, \ldots, s_N\}$，

步3 计算S中每个个体的适应度f()；

步4 若终止条件满足，则取S中适应度最大的个体作为所求结果。

若不满足则执行算子操作

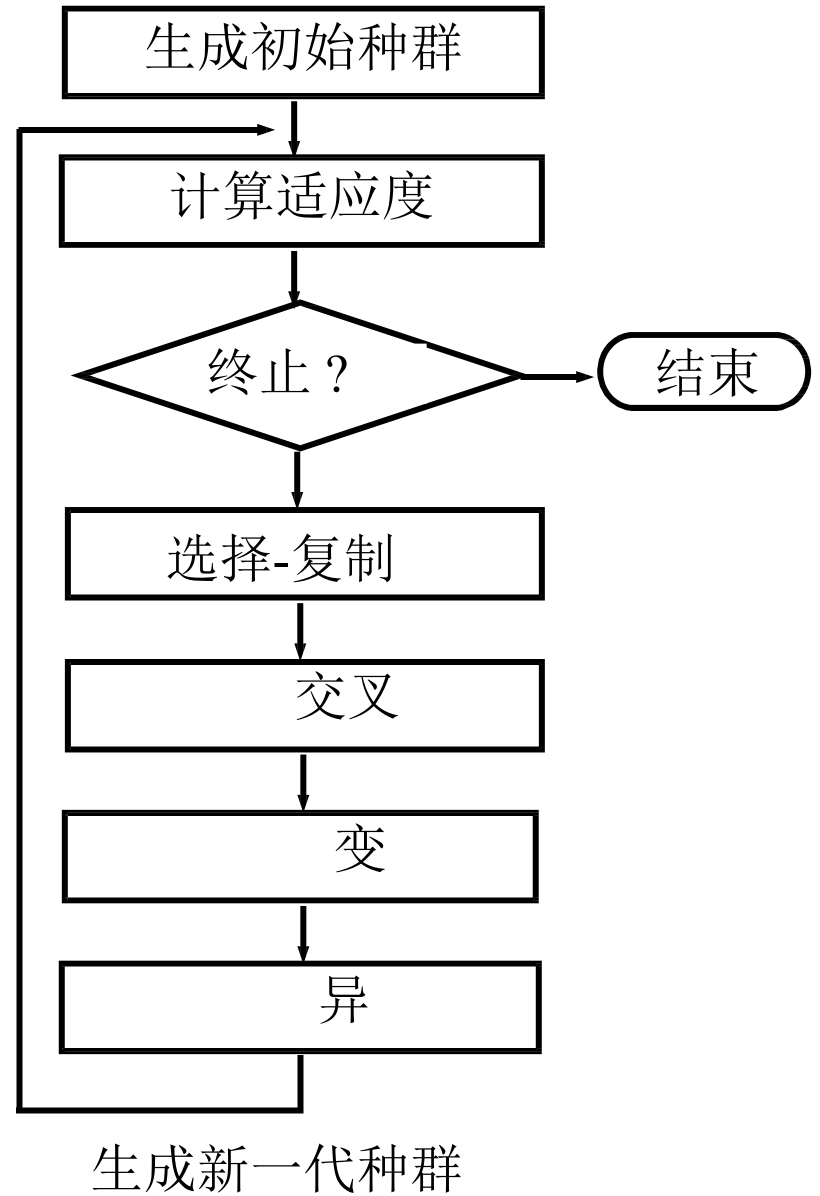
步4.1 按**选择概**率$P(x_i)$，从S中随机选定个体，组成群体S_{t1}

步4.2 按**交叉率**P_c所得个体数c，从S_{t1}中随机确定c个染色体，进行交叉操作，得群体S_{t2}；

步4.3 按**变异率**P_m决定的变异次数m，从S_{t2} 中随机确定m个染色体，进行变异操作，得群体S_{t3}；

步4.4 将群体S_{t3}作为新一代种群，即用S_{t3}代替S，$t = t+1$，转步3

遗传算法基本流程图



例2 背包问题

已知 l 个物品的质量及其价值分别为 $w_i(i=1,2,\cdots,l)$ 和 $v_i(i=1,2,\cdots,l)$，

背包的最大载重量为C，因此，背包问题的数学建模为：
背包在最大载重限制内所装物品的总价值最大即，

$$\max_{x} \sum_{i=1}^{l} v_i x_i,$$

$$\text{s.t.} \quad \sum_{i=1}^{l} w_i x_i, \leq C, \quad x_\text{i} \in \{0,1\}, i = 1,2,\cdots,l$$

，x_i 为0–1决策变量，表示物品 i 是否被装包，如果是，则 $x_i=1$，否则 $x_i=0$。

问题分析

假设现在有1个背包，5个物品，背包最大承重6kg

表1.1 5个物品的质量和价值数据

物品序号	质量/kg	价值/元
1	5	12
2	2	8
3	1	4
4	4	15
5	3	6

所有背包方案列举如下：

表1.2 装包方案

装包方案序号	装包物品编号集合	装包物品总质量/kg	装包物品总价值/元
1	1	5	12
2	2	2	8
3	3	1	4
4	4	4	15
5	5	3	6
6	1、3	6	16
7	2、3	3	12
8	2、4	6	23
9	2、5	5	14
10	2、3、5	6	18
11	3、4	5	19
12	3、5	4	10

一共有12种装包方案，其中方案8的装包物品总价值是最大

0-1表示背包方案

0：表示物品没有被装进背包
1：表示物品被装进背包

可将上述12种方案用0-1数字进行表示
例如，方案10中，

物品2、3、5装进包中，而物品1、4没有装进包中，

方案10可以用01101表示，

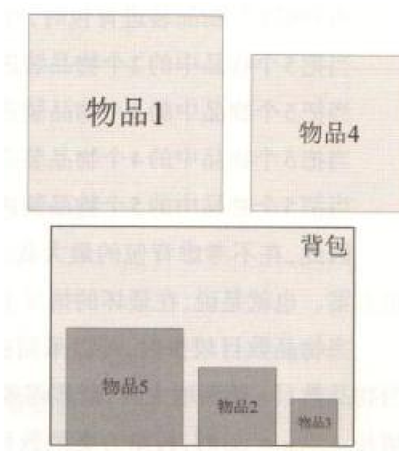


图1.1 方案10

算法的编码

表1.3　装包方案形式转换

方案序号	装包物品编号集合	形式转换后的装包方案	装包物品总质量/kg	装包物品总价值/元
1	1	10000	5	12
2	2	01000	2	8
3	3	00100	1	4
4	4	00010	4	15
5	5	00001	3	6
6	1、3	10100	6	16
7	2、3	01100	3	12
8	2、4	01010	6	23
9	2、5	01001	5	14
10	2、3、5	01101	6	18
11	3、4	00110	5	19
12	3、5	00101	4	10

求解策略

1 如何编码

2 如何处理约束条件

3 如何选择个体

4 如何进行交叉

5 如何进行变异

STEP 1 初始化种群

随机生成含5个基因的n个染色体，
即，5个{0，1}数组成的个体，并且判断是否超过最大载重量
例如：10条染色体

1	0	0	0	0
0	1	0	1	0
0	0	1	0	0
0	0	1	1	0
1	0	0	0	0
0	1	1	0	0
0	0	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	0	0

```
round(rand(1,n));
```

STEP 2 计算适应度函数值，执行选择算子

[12 23 4 19 12 12 6 23 18 12]

[0.0851 0.1631 0.0284 0.1348 0.0851 0.0851 0.0426 0.1631 0.1277 0.0851]

执行选择操作后的种群个数为9，

1	0	0	0	0
0	1	0	1	0
0	0	1	0	0
0	0	1	1	0
1	0	0	0	0
0	1	1	0	0
0	0	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	0	0

10->9

0	1	0	1	0
0	1	0	1	0
0	0	1	1	0
1	0	0	0	0
0	1	1	0	0
0	1	1	0	1
0	1	0	1	0
0	1	0	1	0
0	1	0	1	0

STEP 3 执行交叉算子

交叉概率为0.9， 随机生成交叉位置

执行交叉操作后的种群

0	1	0	1	0
0	1	0	1	0
0	0	1	1	0
1	0	0	0	0
0	1	1	0	0
0	1	1	0	1
0	1	0	1	0
0	1	0	1	0
0	1	0	1	0

10->9

0	1	0	1	0
0	1	0	1	0
1	0	0	1	0
0	0	1	0	0
0	1	1	0	0
0	1	1	0	1
0	1	0	1	0
0	1	0	1	0
0	1	0	1	0

STEP 4 执行变异算子

变异概率为0.08， 随机生成变异位置

执行变异操作后的种群：未发生变异

0	1	0	1	0
0	1	0	1	0
1	0	0	1	0
0	0	1	0	0
0	1	1	0	0
0	1	1	0	1
0	1	0	1	0
0	1	0	1	0
0	1	0	1	0

$\Rightarrow$

0	1	0	1	0
0	1	0	1	0
1	0	0	1	0
0	0	1	0	0
0	1	1	0	0
0	1	1	0	1
0	1	0	1	0
0	1	0	1	0
0	1	0	1	0

STEP 5 更新种群

对得到的种群个体进行约束处理：满足约束条件选取适应度值最大的个体添加，更新种群

0	1	0	1	0
0	1	0	1	0
1	0	0	1	0
0	0	1	0	0
0	1	1	0	0
0	1	1	0	1
0	1	0	1	0
0	1	0	1	0
0	1	0	1	0

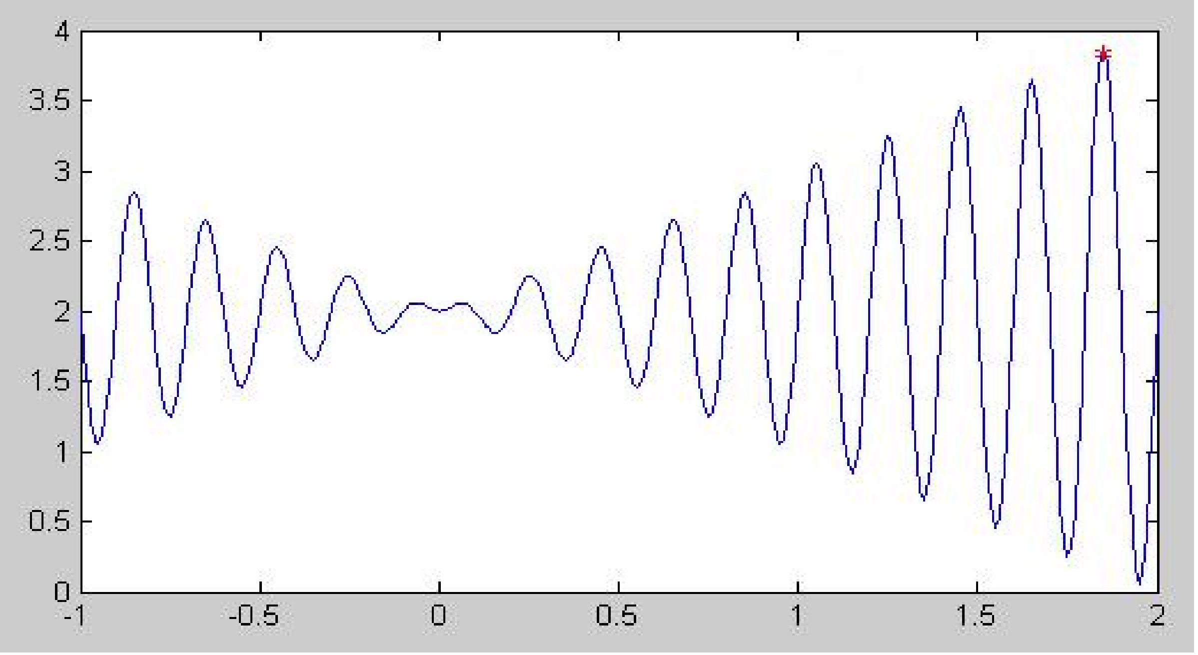
→

0	1	0	1	0
0	1	0	1	0
0	1	0	1	0
0	1	0	1	0
0	0	1	0	0
0	1	1	0	0
0	1	1	0	1
0	1	0	1	0
0	1	0	1	0
0	1	0	1	0

例3

一元函数求最大值：

$$f(x) = x\sin(10\pi * x) + 2.0 \qquad x \in [-1,2]$$



- **编码**

表现型：x

基因型：二进制编码（串长取决于求解精度）

<u>串长与精度之间的关系</u>：

若要求求解精度到6位小数，区间长度为2-(-1)＝3，即需将区间分为3/0.000001=3×10^{6}等份。

$$2097152 = 2^{21} < 3000000 < 2^{22} = 4194304$$

所以编码的二进制串长应为22位。

- **产生初始种群**

  产生的方式：随机

  产生的结果：长度为22的二进制串

  产生的数量：种群的大小（规模），如30，50，…

  1111010011100001011000
  1100110011101010101110
  1010100011110010000100
  1011110010011100111001
  0001100101001100000011
  0000011010010000000000
  ……

- **计算适应度**

本例：直接用目标函数作为适应度函数

①将某个体转化为[-1,2]区间的实数：

$s=<1000101110110101000111> \rightarrow x=0.637197$

②计算x的函数值（适应度）：

$f(x)=x\sin(10\pi x)+2.0=2.586345$

- **计算适应度**

<u>二进制与十进制之间的转换</u>：

1，将一个二进制串（$b_{21}b_{20}\ldots b_0$）转化为10进制数：

$$(b_{21}b_{20}\cdots b_0)_2 = (\sum_{i=0}^{21} b_i \cdot 2^i)_{10} = x'$$

2，x'对应的区间[-1,2]内的实数：

$$\frac{x-(-1)}{2-(-1)} = \frac{x'-1}{2^{22}-1}$$

- **遗传操作**

  选择：轮盘赌选择法；

  交叉：单点交叉；

  变异：小概率变异

- **模拟结果**

  <u>**设置的参数：**</u>

  种群大小50；交叉概率0.75；变异概率0.05；最大代数200。

  <u>**得到的最佳个体：**</u>

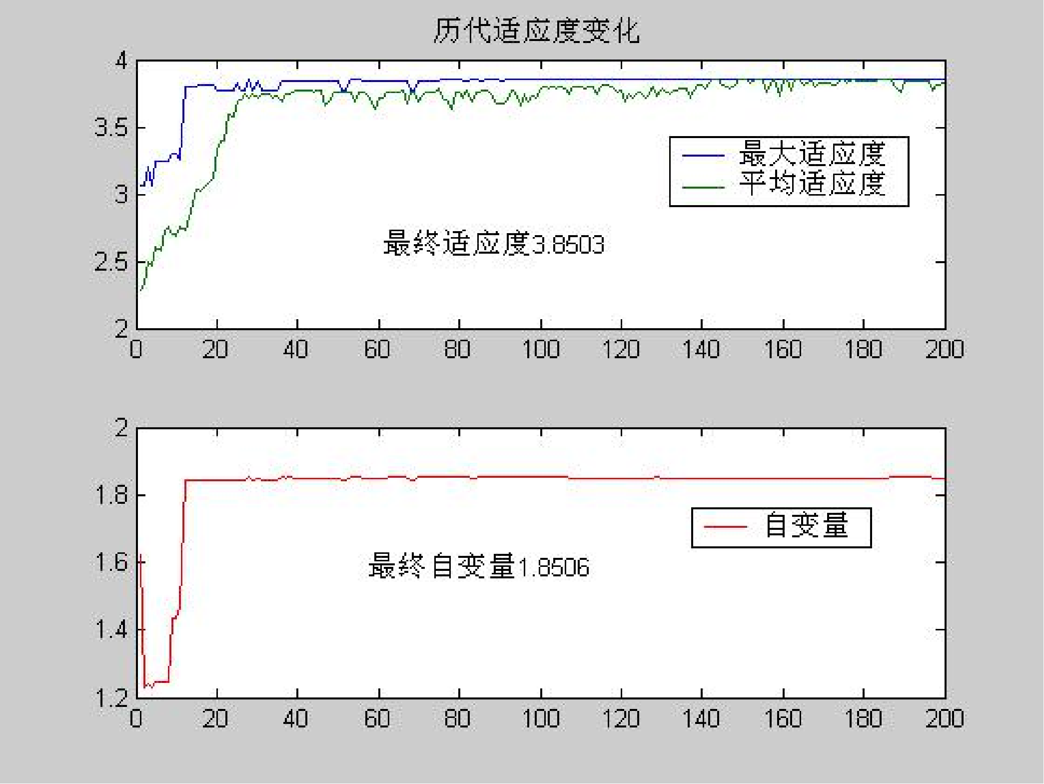
  s_{max}=<1111001100111011111100>;

  x_{max}=1.8506;

  $f(x_{max})$=3.8503;

- **模拟结果**

<u>进化的过程：</u>



世代数	自变量	适应度
1	1.4495	3.4494
9	1.8395	3.7412
17	1.8512	3.8499
30	1.8505	3.8503
50	1.8506	3.8503
80	1.8506	3.8503
120	1.8506	3.8503
200	1.8506	3.8503

5.2 算法的设计与实现

- **主程序**

```
%初始化
clear


bn=22;   %个体串长度
inn=50;   %初始种群大小
gnmax=200;    %最大代数
pc=0.75;   %交叉概率
pm=0.05;   %变异概率
```

主程序

```
%产生初始种群
s=round(rand(inn,bn));

%计算适应度,返回适应度f和累积概率p
[f,p]=objf(s);
```

```
gn=1;
while gn<gnmax+1
  for j=1:2:inn
    %选择操作
    seln=sel(s,p);
    %交叉操作
    scro=cro(s,seln,pc);
    scnew(j,:)=scro(1,:);
    scnew(j+1,:)=scro(2,:);
    %变异操作
    smnew(j,:)=mut(scnew(j,:),pm);
    smnew(j+1,:)=mut(scnew(j+1,:),pm);
  end
```

◆ 主程序

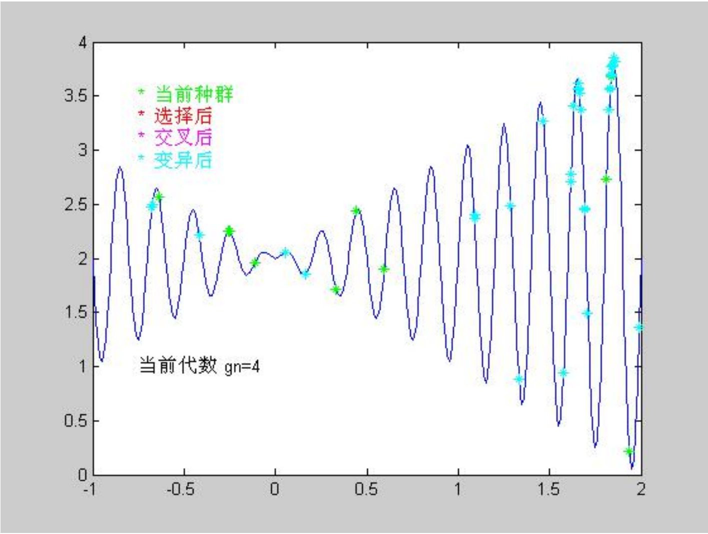
```
s=smnew;  %产生了新的种群
%计算新种群的适应度
[f,p]=objf(s);

%记录当前代最好和平均的适应度
[fmax,nmax]=max(f);
fmean=mean(f);
ymax(gn)=fmax;
ymean(gn)=fmean;
```

- **主程序**

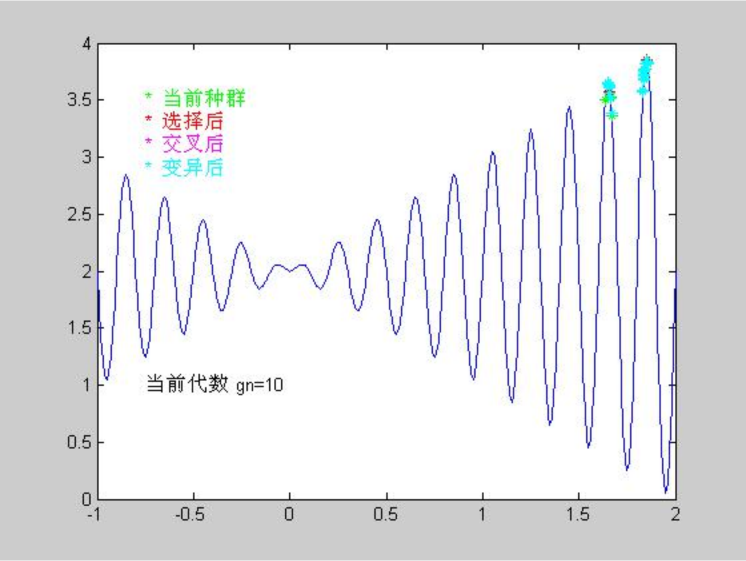
```
  %记录当前代的最佳个体
  x=n2to10(s(nmax,:));
  xx=-1.0+x*3/(power(2,bn)-1);
  xmax(gn)=xx;

  gn=gn+1
end
gn=gn-1;
```

- **主程序**

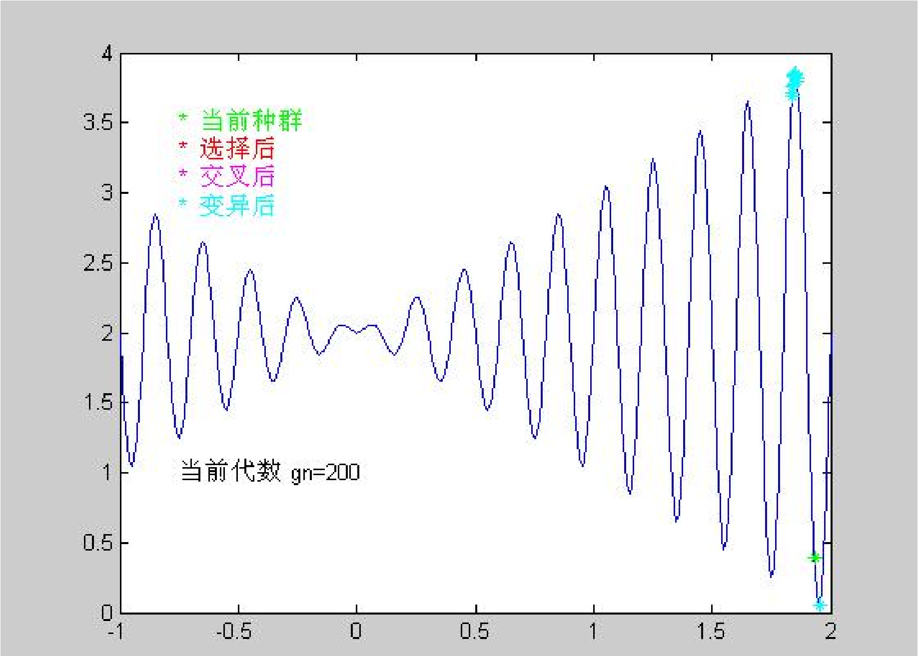
```
%绘制曲线
subplot(2,1,1);
plot(1:gn,[ymax;ymean]);
title('历代适应度变化','fonts',10);
legend('最大适应度','平均适应度');
string1=['最终适应度',num2str(ymax(gn))];
gtext(string1);
subplot(2,1,2);
plot(1:gn,xmax,'r-');
legend('自变量');
string2=['最终自变量',num2str(xmax(gn))];
gtext(string2);
```



* 当前种群
* 选择后
* 交叉后
* 变异后
当前代数 gn=4



* 当前种群
* 选择后
* 交叉后
* 变异后
当前代数 gn=10



* 当前种群
* 选择后
* 交叉后
* 变异后
当前代数 gn=200

5.3、遗传算法的特点

1　群体搜索，易于并行化处理；

2　不是盲目穷举，而是启发式搜索；

3　适应度函数不受连续、可微等条件的约束，适用范围很广。

6、理论基础

1、遗传算法的数学基础

2、遗传算法的局限性

3、遗传算法的改进

6.1 遗传算法的数学基础

模式 H 描述种群中染色体相似性的字符串。

例如：$\left.\begin{array}{l}1001101110\\1001001010\\1111101110\\1001111110\end{array}\right\}$染色体子集 $\left.\begin{array}{l}10101\\10111\\11100\end{array}\right\}$染色体

1**1**1**10 — 模式*H*　　1*1** — 模式*H*

- 定义1：模式 H 中确定位置的个数称为模式 H 的阶，记作 O(H)。例如O(10**1)=3 。
- 定义2：模式 H 中第一个确定位置和最后一个确定位置之间的距离称为模式H的定义距，记作 δ(H)。例如 δ(10**1)=4 。

模式的阶和定义距的意义

模式阶用来反映不同模式间确定性的差异，

模式阶数越高， 模式的确定性就越高，所匹配的样本数就越少。
在遗传操作中，即使阶数相同的模式，也会有不同的性质，
模式的定义距就反映了这种性质的差异。

模式定理

具有低阶、短定义距以及平均适应度高于种群 平均适应度的模式在子代中呈指数增长。

- 结论：在选择、交换、变异的作用下，位数少 $o(H)$、定义长度短 $\delta(H)$、适应度高 $f(H,t)$ 的模式，其下一代的存活率会增大。

积木块假设

积木块：具有短定义距、低阶以及高平均适应度的模式

积木块假设：

个体的积木块通过选择，交叉，变异等算子作用，能够互相结合在一起，形成高阶，长距，高平均适应度的个体编码串。

注：模式定理保证了较优模式的样本数呈指数增长，从而使遗传算法找到全局最优解的可能性存在；

积木块假设则指出了在 遗传算子的作用下，能生成全局最优解。

6.2 GA的局限性

1. 若群体规模太小，含有的信息量少，不能使算法得到充分发挥，
2. 若群体规模大，包含的信息量也大，但计算次数会激剧增加，因而限制了算法的使用。
3. GA中最重要的遗传算子——交叉算子使群体中的个体具有局部相似性，父代染色体的信息交换量小，从而使搜索停滞不前；
4. 另一方面是变异概率又太小，以至于不能使搜索转向其它的解空间进行搜索。

程序作业：

使用遗传算法求以下函数的最大值点

$$f(x)=11\sin(6x)+7\cos(5x), x\in[-\pi,\pi]$$

请写出算法流程及用matlab实现