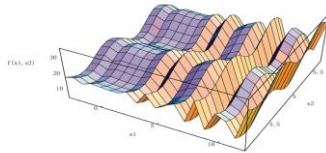


遗传算法实例

• 无约束优化问题：

(精度达到小数点后面四位)

$$\max f(x_1, x_2) = 21.5 + x_1 \cdot \sin(4\pi x_1) + x_2 \cdot \sin(20\pi x_2)$$
$$\text{s. t. } -3.0 \leq x_1 \leq 12.1$$
$$4.1 \leq x_2 \leq 5.8$$



编码与解码

• 二进制表示

■ $x_j \in [a_j, b_j]$ ，精度至少是小数点后面四位

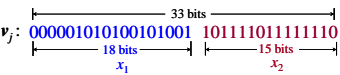
■ 编码长度： $2^{l-1} < (b_j - a_j) \times 10^4 \leq 2^l - 1$

$$x_1: (12.1 - (-3.0)) \times 10,000 = 151,000$$
$$2^{17} < 151,000 \leq 2^{18}, \quad m_1 = 18 \text{ bits}$$

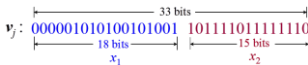
$$x_2: (5.8 - 4.1) \times 10,000 = 17,000$$
$$2^{14} < 17,000 \leq 2^{15}, \quad m_2 = 15 \text{ bits}$$



编码长度： $m = m_1 + m_2 = 18 + 15 = 33 \text{ bits}$



■ 解码公式： $x_j = a_j + \text{decimal}(\text{substring}_j) \times \frac{b_j - a_j}{2^{l_j} - 1}$



	Binary Number	Decimal Number
x_1	000001010100101001	5417
x_2	1011110111111110	24318



$$x_1 = -3.0 + 5417 \times \frac{12.1 - (-3.0)}{2^{18} - 1}$$
$$= -2.687969$$

$$x_2 = 4.1 + 24318 \times \frac{5.8 - 4.1}{2^{15} - 1}$$
$$= 5.361653$$

• 初始种群以及个体相应解码：

- $v_1 = [00000101010010100110111101111110] = [x_1, x_2] = [-2.687969 \quad 5.361653]$
 $v_2 = [001110101110011000000010101001000] = [x_1, x_2] = [0.474101 \quad 4.170144]$
 $v_3 = [111000111000001000010101001000110] = [x_1, x_2] = [10.419457 \quad 4.661461]$
 $v_4 = [100110110100101101000000010111001] = [x_1, x_2] = [6.159951 \quad 4.109598]$
 $v_5 = [00001011110110001000111000101000] = [x_1, x_2] = [-2.301286 \quad 4.477282]$
 $v_6 = [111110101011011000000010110011001] = [x_1, x_2] = [11.788084 \quad 4.174346]$
 $v_7 = [110100010011111000100110011101101] = [x_1, x_2] = [9.342067 \quad 5.121702]$
 $v_8 = [001011010100001100010110011001100] = [x_1, x_2] = [-0.330256 \quad 4.694977]$
 $v_9 = [111110001011101100011101000111101] = [x_1, x_2] = [11.671267 \quad 4.873501]$
 $v_{10} = [111101001110101010000010101101010] = [x_1, x_2] = [11.446273 \quad 4.171908]$

■ 适应度评价：

$$\text{eval}(v_k) = f(x_1, x_2, \dots, x_n) \quad (k = 1, 2, \dots, \text{popSize})$$

如： $v_1 = [00000101010010100110111101111110] = [x_1, x_2] = [-2.687969 \quad 5.361653]$

$$(x_1 = -2.687969, x_2 = 5.361653)$$



$$\text{eval}(v_1) = f(-2.687969, 5.361653) = 19.805119$$

种群适应度评价

- $\text{eval}(v_1) = f(-2.687969, 5.361653) = 19.805119$
 $\text{eval}(v_2) = f(0.474101, 4.170144) = 17.370896$
 $\text{eval}(v_3) = f(10.419457, 4.661461) = 9.590546$
 $\text{eval}(v_4) = f(6.159951, 4.109598) = 29.406122$
 $\text{eval}(v_5) = f(-2.301286, 4.477282) = 15.686091$
 $\text{eval}(v_6) = f(11.788084, 4.174346) = 11.900541$
 $\text{eval}(v_7) = f(9.342067, 5.121702) = 17.958717$
 $\text{eval}(v_8) = f(-0.330256, 4.694977) = 19.763190$
 $\text{eval}(v_9) = f(11.671267, 4.873501) = 26.401669$
 $\text{eval}(v_{10}) = f(11.446273, 4.171908) = 10.252480$

显然 v_4 是最优的个体， v_3 是最差的个体

• Selection

- 轮盘赌选择 (roulette wheel) :
- 通过适应度来计算个体被选择的概率

• step 1: 计算种群中所有个体的适应度值之和 $F = \sum_{k=1}^{popSize} eval(v_k)$;

• step 2: 计算每个个体 v_k 的选择概率 p_k ;

$$p_k = \frac{eval(v_k)}{F}, \quad k = 1, 2, \dots, popSize$$

• step 3: 计算每个个体 v_k 的累积概率 q_k ;

$$q_k = \sum_{j=1}^k p_j, \quad k = 1, 2, \dots, popSize$$



• step 4: 随机产生N个 [0, 1] 的随机数 r_i ;

• step 5: 对于每一个 r_i : 如果 $r_i \leq q_1$, 选择第一个个体 v_1 ; 否则, 如果 $q_{k-1} < r_i \leq q_k$, 选择第 k 个个体 v_k ($2 \leq k \leq popSize$).

46

• 轮盘赌选择

step 1: 计算种群中所有个体的适应度值之和

$$F = \sum_{k=1}^{10} eval(v_k) = 178135372$$

step 2: 计算每个个体 v_k 的选择概率 p_k

$$\begin{aligned} p_1 &= 0.111180, & p_2 &= 0.097515, & p_3 &= 0.053839, & p_4 &= 0.165077, \\ p_5 &= 0.088057, & p_6 &= 0.066806, & p_7 &= 0.100815, & p_8 &= 0.110945, \\ p_9 &= 0.148211, & p_{10} &= 0.057554 \end{aligned}$$

step 3: 计算每个个体 v_k 的累积概率 q_k

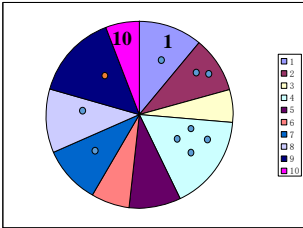
$$\begin{aligned} q_1 &= 0.111180, & q_2 &= 0.208695, & q_3 &= 0.262534, & q_4 &= 0.427611, \\ q_5 &= 0.515668, & q_6 &= 0.582475, & q_7 &= 0.683290, & q_8 &= 0.794234, \\ q_9 &= 0.942446, & q_{10} &= 1.000000 \end{aligned}$$

step 4: 随机产生N个 [0, 1] 的随机数 r_i

$$\begin{aligned} &0.301431, & 0.322062, & 0.766503, & 0.881893, & 0.350871, \\ &0.583392, & 0.177618, & 0.343242, & 0.032685, & 0.197577 \end{aligned}$$

47

$$\begin{aligned} q_1 &= 0.111180, & q_2 &= 0.208695, & q_3 &= 0.262534, & q_4 &= 0.427611, \\ q_5 &= 0.515668, & q_6 &= 0.582475, & q_7 &= 0.683290, & q_8 &= 0.794234, \\ q_9 &= 0.942446, & q_{10} &= 1.000000 \end{aligned}$$



$q_5 < r_1 = 0.301432 \leq q_6$,
选中 v_6
 $q_5 < r_2 = 0.322062 \leq q_6$,
选中 v_6
 $q_7 < r_3 = 0.766503 \leq q_8$,
选中 v_8
.....

random number r				
0.301431,	0.322062,	0.766503,	0.881893,	0.350871,
0.583392,	0.177618,	0.343242,	0.032685,	0.197577

48

最终, 新种群包含如下染色体 :

$$\begin{aligned} v_1' &= [100110110100101101000000010111001] & (v_4) \\ v_2' &= [100110110100101101000000010111001] & (v_4) \\ v_3' &= [001011010100001100010110011001100] & (v_6) \\ v_4' &= [111110001011101100011101000111101] & (v_6) \\ v_5' &= [100110110100101101000000010111001] & (v_4) \\ v_6' &= [11010001001111100010011001101101] & (v_7) \\ v_7' &= [001110101110011000000010101001000] & (v_2) \\ v_8' &= [100110110100101101000000010111001] & (v_4) \\ v_9' &= [000001010100101001101110111110] & (v_1) \\ v_{10}' &= [001110101110011000000010101001000] & (v_2) \end{aligned}$$

49

• 交叉算子 (单点交叉)

- 在交叉过程中, 根据交叉概率 p_c 判断每组染色体对是否进行交叉。
- 随机选择一个交叉点:

交叉点位于第17个基因位

$$\begin{aligned} v_1 &= [100110110100101101000000010111001] \\ v_2 &= [0011101011100110000000101001000] \end{aligned}$$

$$\begin{aligned} c_1 &= [1001101101001011000000010101001000] \\ c_2 &= [001110101110011001000000010111001] \end{aligned}$$

50

• 变异 (单点变异)

对每条染色体的每一位都产生 [0, 1] 间的随机数, 根据变异概率 p_m , 判断该位是否需要变异。

在第16个基因位进行变异

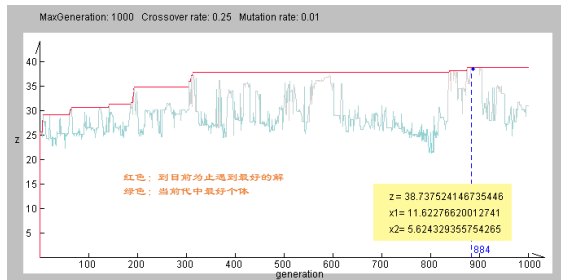
$$\begin{aligned} v_1 &= [100110110100101101000000010111001] \\ c_1 &= [100110110100101000000101001000] \end{aligned}$$

51

• 下一代种群

$v_1' = [100110110100101101000000010111001], f(6.159951, 4.109598) = 29.406122$
 $v_2' = [100110110100101101000000010111001], f(6.159951, 4.109598) = 29.406122$
 $v_3' = [001011010100001100010110011001100], f(-0.330256, 4.694977) = 19.763190$
 $v_4' = [111110001011101100011101000111101], f(11.907206, 4.873501) = 5.702781$
 $v_5' = [100110110100101101000000010111001], f(8.024130, 4.170248) = 19.91025$
 $v_6' = [1101000100111110001001100110110101], f(9.34067, 5.121702) = 17.958717$
 $v_7' = [100110110100101101000000010111001], f(6.159951, 4.109598) = 29.406122$
 $v_8' = [100110110100101101000000010111001], f(6.159951, 4.109598) = 29.406122$
 $v_9' = [00000101010010100110111101111110], f(-2.687969, 5.361653) = 19.805199$
 $v_{10}' = [001110101110011000000010101001000], f(0.474101, 4.170248) = 17.370896$

52



54

• 进化结果

- 算法运行1000代后终止。
- 在第884代可以得到如下所示的最优解:

$$\begin{aligned} \max \quad & f(x_1, x_2) = 21.5 + x_1 \cdot \sin(4\pi x_1) + x_2 \cdot \sin(20\pi x_2) \\ \text{s. t.} \quad & -3.0 \leq x_1 \leq 12.1 \\ & 4.1 \leq x_2 \leq 5.8 \end{aligned}$$

$eval(v^*) = f(11.622766, 5.624329) = 38.737524$
 $x_1^* = 11.622766$
 $x_2^* = 5.624329$
 $f(x_1^*, x_2^*) = 38.737524$

53

填空题 4分

设置

正常使用填空题需3.0以上版本浏览器

种群中共有4个个体:S1, S2, S3, S4, 假设每个个体的选择概率分别为 $p_1=0.28$, $p_2=0.24$, $p_3=0.16$, $p_4=0.32$ 。若选择阶段随机生成的4个随机数分别为0.6846, 0.5198, 0.7612, 0.2403, 请回答每个随机数分别选中的是哪个个体(只写个体的序号)?

0.6846选中 [填空1], 0.5198选中 [填空2], 0.7612选中 [填空3], 0.2403选中 [填空4]。

作答

55

进化算法的分类

进化算法:

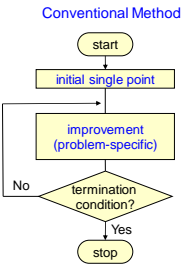
- 1 进化规划
- 2 进化策略
- 3 遗传算法

56

讨论: 和传统方法相比, 进化算法有哪些优势?

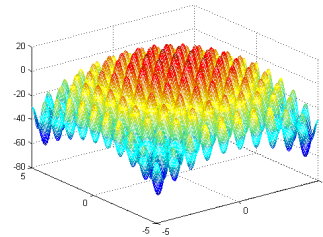
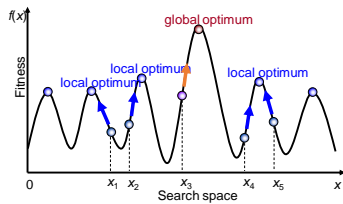
□ 传统方法 (point-to-point approach)

- 共轭梯度法、拟牛顿法、单纯形方法
- 特点:
 - 经典的优化搜索算法往往是基于梯度的, 梯度方向提高个体性能
 - 单点搜索
 - 局部最优



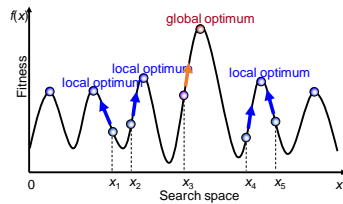
57

- 进化算法同时进行解空间的多点搜索。
- 传统的优化算法往往从解空间的一个初始点开始搜索，这样容易陷入局部极值点。进化算法进行群体搜索，而且在搜索的过程中引入遗传运算，使群体又可以不断进化。这些是进化算法所特有的一种隐含并行性。

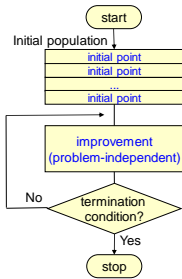


$$f(x) = -\sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad S = [-5.12, 5.12]^n$$
$$f(x^*) = 0 \quad x^* = (0, 0, \dots, 0)$$

- 进化算法使用概率搜索技术。
- 进化算法属于一种自适应概率搜索技术，其选择、交叉、变异等运算都是以一种概率的方式来进行的，从而增加了其搜索过程的灵活性以及种群的多样性。



- 进化算法直接以目标函数值作为搜索信息。
- 传统的优化算法往往不只需要目标函数值，还需要目标函数的导数等其它信息。这样对许多目标函数无法求导或很难求导的函数，进化算法就比较方便。



遗传算法的特点

- 1) 遗传算法是对决策变量的编码进行操作，这样提供的参数信息量大，优化效果好；
- 2) 遗传算法是从许多点开始并行操作，因而可以有效地防止搜索过程收敛于局部最优解。同时这一特点使得它可以采用大规模并行计算来提高计算速度；
- 3) 遗传算法通过目标函数来计算适应度值，而不需要其导数和附加信息，从而对问题的依赖性小，应用范围较广。
- 4) 遗传算法的择优规则是由概率决定的，而非确定性的。这也决定了它是一种高效启发式搜索，而非盲目地穷举或完全随机搜索。
- 5) 遗传操作简单，易于实现，这一特点也使得它能很方便与其他方法混合。

讨论：进化算法适用于什么问题？

- 进化算法只需计算目标函数的值即可，对优化问题本身的性质要求是非常低的，不会像数学优化算法那样往往依赖于一大堆的条件，例如是否为凸优化，目标函数是否可导等。
- 所以，对问题结构确定的优化问题，有充分的关于优化问题的信息来利用的时候，数学优化一般来说有优势，例如线性规划，二次规划，凸优化等等。反之，使用进化算法更有优势。
- 对于一些数学优化目前不能彻底解决的问题例如NP-hard问题，进化算法也有很大的应用前景。

小结

• 遗传算法的组成

- 编码机制
- 种群初始化
- 适应度函数
- 遗传算子
 - (选择、交叉、变异)

64

算法启示

• 遗传算法的组成

- 编码机制
- 种群初始化
 - (1) 群体搜索 > 个体搜索
(团队合作)
- 适应度函数
 - (2) 概率操作, 交叉变异 → 群体多样性
(生物多样性)
- 遗传算子
 - (选择、交叉、变异)

65

Copyright by Lrc&Mch