遗传算法

1. 优化问题举例
2. 参数选择
3. 执行技巧

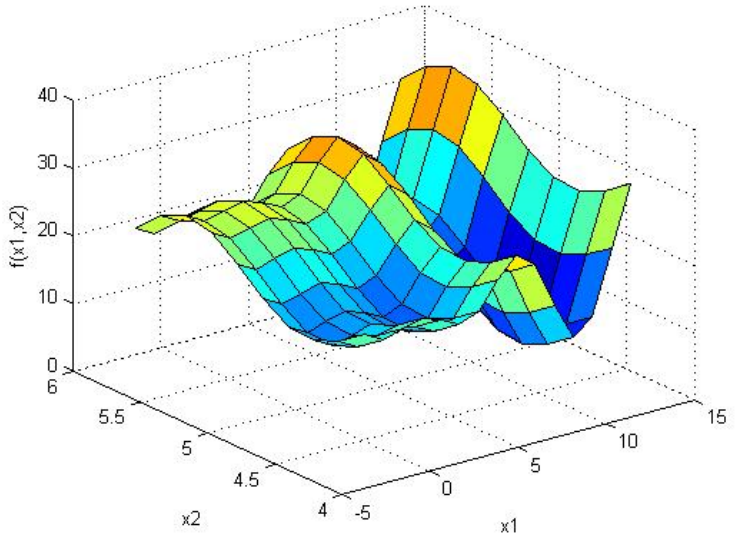
1. 优化问题举例

$$\max f(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) + x_2 \sin(20\pi x_2)$$

$$-3.0 \le x_1 \le 12.1$$

$$4.1 \le x_2 \le 5.8$$

$$\varepsilon = 10^{-4}$$



1. 优化问题举例

Step0 确定编码方案

首先，确定编码长度

$$-3.0 \le x_1 \le 12.1 \qquad m = 18$$

$$4.1 \le x_2 \le 5.8 \qquad m = 15$$

因此，染色体总长18+15=33位。

1. 优化问题举例

Step0 确定编码方案

Step1 初始化种群，令N=10，t=0

$$v_1 = [00000101010010100110111011111110]$$
$$v_2 = [00111010111001100000001010100100 0]$$
$$v_3 = [11100011100000100001010100100011 0]$$
$$v_4 = [10011011010010110100000010111001]$$
$$v_5 = [00001011110110001000111000110100 0]$$
$$v_6 = [11111010101101100000001011001100 1]$$
$$v_7 = [11010001001111100010011001110110 1]$$
$$v_8 = [00101101010000110001011001100110 0]$$
$$v_9 = [11111000101110110001110100011110 1]$$
$$v_{10} = [11110100111010101000001010110101 0]$$

1. 优化问题举例

Step2 计算个体适应度值

✓ 首先，需要对种群个体进行解码；

$v_1 = [00000101010010100110111011111110]$ $v_1 = [x_1, x_2] = [-2.687969, 5.361653]$

$v_2 = [00111010111001100000001010101001000]$ $v_2 = [x_1, x_2] = [0.474101, 4.170144]$

$v_3 = [11100011100000100001010101001000110]$ $v_3 = [x_1, x_2] = [10.419457, 4.661461]$

$v_4 = [10011011010010110100000010111001]$ $v_4 = [x_1, x_2] = [6.159951, 4.109598]$

$v_5 = [00001011110110001000111000110100 0]$ $v_5 = [x_1, x_2] = [-2.301286, 4.477282]$

$v_6 = [11111010101101100000001011001100 1]$ $v_6 = [x_1, x_2] = [11.788084, 4.174346]$

$v_7 = [11010001001111110001001100111011 01]$ $v_7 = [x_1, x_2] = [9.342067, 5.121702]$

$v_8 = [00101101010000110001011001100 1100]$ $v_8 = [x_1, x_2] = [-0.330256, 4.694977]$

$v_9 = [11111000101110110001110100011110 1]$ $v_9 = [x_1, x_2] = [11.671267, 4.873501]$

$v_{10} = [11110100111010101000001010110101 0]$ $v_{10} = [x_1, x_2] = [11.446273, 4.171908]$

1. 优化问题举例

Step2 计算个体适应度值

$$\max f(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) + x_2 \sin(20\pi x_2)$$

✓ 将目标函数直接作为个体的适应度函数；

$$v_1 = f(x_1, x_2) = f(-2.687969, 5.361653) = 19.805119$$
$$v_2 = f(x_1, x_2) = f(0.474101, 4.170144) = 17.370896$$
$$v_3 = f(x_1, x_2) = f(10.419457, 4.661461) = 9.590546$$
$$v_4 = f(x_1, x_2) = f(6.159951, 4.109598) = 29.406122$$
$$v_5 = f(x_1, x_2) = f(-2.301286, 4.477282) = 15.686091$$
$$v_6 = f(x_1, x_2) = f(11.788084, 4.174346) = 11.900541$$
$$v_7 = f(x_1, x_2) = f(9.342067, 5.121702) = 17.958717$$
$$v_8 = f(x_1, x_2) = f(-0.330256, 4.694977) = 19.763190$$
$$v_9 = f(x_1, x_2) = f(11.671267, 4.873501) = 26.401669$$
$$v_{10} = f(x_1, x_2) = f(11.446273, 4.171908) = 10.252480$$

1. 优化问题举例

Step3 通过轮盘赌选择N=10个个体放入交叉池。

✓ 首先，计算个体的适应度占总适应度的比值。

$$p_i = \frac{f(v_i)}{\sum_{j=1}^{N} f(v_j)} = \frac{f(v_i)}{f_{sum}}$$

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

1. 优化问题举例

Step3 通过轮盘赌选择N=10个个体放入交叉池。

✓ 然后，计算个体的累计适应度。

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

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

1. 优化问题举例

Step3 通过轮盘赌选择N=10个个体放入交叉池。

✓ 最后，随机产生10个[0,1]之间的实数，并选择对应的个体。

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

0.301431	0.766503
0.322062	0.881893
0.350871	0.583392
0.177618	0.343242
0.032685	0.197577

$$v_4, v_8, v_4, v_9, v_4, v_7, v_2, v_4, v_1, v_2$$

1. 优化问题举例

Step3 通过轮盘赌选择N=10个个体放入交叉池。

✓ 最后，随机产生10个[0,1]之间的实数，并选择对应的个体。

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

0.301431	0.766503
0.322062	0.881893
0.350871	0.583392
0.177618	0.343242
0.032685	0.197577

$$v_4, v_8, v_4, v_9, v_4, v_7, v_2, v_4, v_1, v_2$$

1. 优化问题举例

Step3 根据交叉概率$p_c=0.6$对选出的个体进行交叉，将两点交叉作为交叉算子，记产生的后代个体的集合为O_1。

$$v_4, v_8, v_4, v_9, v_4, v_7, v_2, v_4, v_1, v_2$$

✓ 首先，随机产生10/2个[0,1]之间的实数。

0.422062	0.650871	0.583392	0.177618	0.943242

✓ 若随机数小于等于p_c则进行两点交叉，产生两个子代个体

1. 优化问题举例

Step3 根据交叉概率$p_c=0.6$对选出的个体进行交叉，将两点交叉作为交叉算子，记产生的后代个体的集合为O_1。

$$v_4, v_8, v_4, v_9, v_4, v_7, v_2, v_4, v_1, v_2$$

0.422062	0.650871	0.583392	0.177618	0.943242

$$v_4 = [10011011010010110100000010111001]$$

$$v_8 = [00101101010000110001011001100]$$

1. 优化问题举例

Step4 根据变异概率$p_m=0.01$对交叉产生的个体进行变异，将单点变异作为变异算子，记产生的后代个体的集合为O_2。

✓ 变异概率0.01表示O_1中平均有1%个基因发生变异

✓ 若交叉产生了m个后代个体，则随机产生m*33个[0,1]之间的实数。

✓若随机数小于等于p_m，则将对应的基因位反转。

$$I=[10011011010010110100000010111001]$$

1. 优化问题举例

Step5 计算所有后代个体的适应度，选出最好的E=2个个体直接保留到下一代种群。

然后，根据轮盘赌选择N-E=8个个体进入下一代种群。

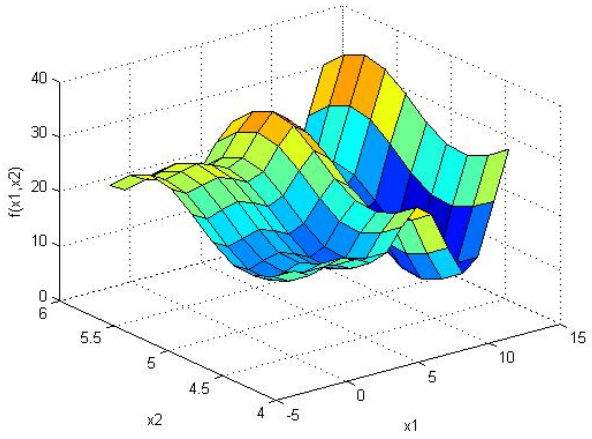
1. 优化问题举例

Step9 若进化代数t=1000，则终止算法，输出适应度最大的个体作为问题的最优解。否则，令t=t+1，转入Step2.

$$v^* = [111101111111010010111101001011010]$$

$$[x_1^*, x_2^*] = [11.6255, 5.7250]$$

$$f(x_1^*, x_2^*) = 38.8503$$



遗传算法

1. 优化问题举例
2. **参数选择**
3. 执行技巧

2. 参数选择

- 种群规模N
- 交叉概率p_c
- 变异概率p_m
- 算法终止条件

2. 参数选择

- 种群规模N

Question：N取值较大或较小时，分别对算法有什么影响？

2. 参数选择

种群规模N

- N取值较小：可以提高算法的运行速度，但却降低了种群的多样性，容易引起算法的早熟（过早收敛）。
- N取值较大：群体的多样性好，提高算法的搜索能力；但会增加算法的计算量，降低算法效率。
- 一般建议取值为20~100

2. 参数选择

- 交叉概率p_c：决定了进化过程种群参与交叉的个体平均数目（$Pc*N$）
  - 较大的p_c使搜索变得随机性太大。
  - 较小交叉概率使得发现新个体的速度太慢。
  - 一般建议取值为0.4~0.99。比较理想的方式是使用自适应的交叉概率。

2. 参数选择

- 变异概率p_m：增加群体进化的多样性，决定了进化过程中种群发生变异的基因平均个数（Pm*O*L）
  - 较大的变异概率使得算法在整个搜索空间中大步跳跃，而小的变异概率使算法聚焦于特别区域作局部搜索。
  - p_m通常取值较小（0.0001~0.5）。比较理想的方式是使用自适应的变异概率。

2. 参数选择

终止条件

- 指定终止进化代数。
  - ✓ 一般而言，事先指定进化代数只能找到问题在给定时限内所能寻求的相对满意解，但不一定是问题的最优解或较高精度的近似解。
- 依据种群稳定情况终止算法：很长时间最优解没有变化；最优解达到一定的误差等。

2. 参数选择

- 目前为止，对于进化算法的参数选择，没有成熟的准则和方法，基本上依赖经验和对所求解问题的了解。

遗传算法

1. 优化问题举例
2. 参数选择
3. 执行技巧

3. 执行技巧

- 遗传算子中融合专门领域知识
- 增加局部搜索

3. 执行技巧

- 遗传算子中融合专门领域知识
  - 举例：背包问题
  - 已知n个物体的容积及其价值分别为w_i和c_i（i=1, 2, ...,n），背包的总容量为v。问：选择哪些物品装入背包可以使背包在满足容量限制的前提下总价值最大？

3. 执行技巧

- 遗传算子中融合专门领域知识
  - 举例：背包问题

$$\max f(X) = \sum_{i=1}^{n} c_i x_i$$

$$s.t.$$

$$x_i \in \{0,1\}, \quad i = 1,2,...,n$$

$$\sum_{i=1}^{n} w_i x_i \le v$$

3. 执行技巧

- 遗传算子中融合专门领域知识
  - 举例：背包问题：贪心算法

$$\max f(X) = \sum_{i=1}^{n} c_i x_i$$

$$s.t.$$

$$x_i \in \{0,1\}, \quad i = 1,2,...,n$$

$$\sum_{i=1}^{n} w_i x_i \le v$$

将物品按照价值密度 $\rho_i = c_i / w_i, i = 1,2,...,n$ 的大小进行降序排列，按照排列好的顺序依次将物品放入背包，直到超出背包的容量限制为止。

3. 执行技巧

- 遗传算子中融合专门领域知识
  - 举例：背包问题：贪心算法

$$\max f(X) = \sum_{i=1}^{n} c_i x_i$$

$$s.t.$$

$$x_i \in \{0,1\}, \quad i = 1,2,...,n$$

$$\sum_{i=1}^{n} w_i x_i \le v$$

$n = 50; \quad v = 1000$

$C = \{220,208,198,192,180,180,165,162,160,158,$
$155,130,125,122,120,118,115,110,105,101,$
$100,100,98,96,95,90,88,82,80,77,75,73,72,$
$70,69,66,65,63,60,58,56,50,30,20,15,10,8,$
$5,3,1\}$

$W = \{80,82,85,70,72,70,66,50,55,25,50,55,40,48,$
$50,32,22,60,30,32,40,38,35,32,25,28,30,22,$
$50,30,45,30,60,50,20,65,20,25,30,10,20,25,$
$15,10,10,10,4,4,2,1\}$

3. 执行技巧

- 遗传算子中融合专门领域知识
  - 举例：背包问题：贪心算法
  - 结果：总价值：3095；总重量：996
  - 选择的物品编号：10, 40 , 17, 25, 28, 16, 19, 35, 37, 8, 26, 20, 13, 11, 24, 27, 9, 23, 41, 1, 4, 22, 6, 30, 14, 2

3. 执行技巧

- 遗传算子中融合专门领域知识
  - 举例：背包问题：混合遗传算法

$$\max f(X)=\sum_{i=1}^{n} c_i x_i$$

$$s.t.$$

$$x_i \in \{0,1\}, \quad i=1,2,...,n$$

$$\sum_{i=1}^{n} w_i x_i \le v$$

（1）编码：用长度为n的0-1字符串代表一个包装策略，即 $X=x_1,x_2,...,x_n \in \{0,1\}^n$，若 $x_i=1$，则第*i*件物品装入包中；否则，表示不装入。

（2）适应度：$f(X)=\sum_{i=1}^{n} c_i x_i$

3. 执行技巧

- 遗传算子中融合专门领域知识
  - 举例：背包问题：混合遗传算法

$$\max f(X) = \sum_{i=1}^{n} c_i x_i$$

$$s.t.$$

$$x_i \in \{0,1\}, \quad i = 1,2,...,n$$

$$\sum_{i=1}^{n} w_i x_i \le v$$

（3）交叉：两点交叉
（4）变异：单点变异
（5）选择：轮盘赌选择

注意：初始化、交叉或变异产生的个体，都不一定满足模型的全部约束条件。

3. 执行技巧

- 遗传算子中融合专门领域知识
  - 举例：背包问题：混合遗传算法

$$\max f(X)=\sum_{i=1}^{n} c_i x_i$$

$$s.t.$$

$$x_i \in \{0,1\}, \quad i=1,2,...,n$$

$$\sum_{i=1}^{n} w_i x_i \le v$$

（6）贪心变换算子：

Step1. 对所有 $x_i=1$ 的物品，按它们的价值密度排序，形成队列*b(i).*

Step2. 令 *k=1.*

Step3. 计算 $v_k=\sum_{j=1}^{k} w_{b(j)}$

Step4. 若 $v_k \le v$，令*k=k+1*,转入*Step3*. 否则，令队列b从k到最后一个元素对应的*x*编码全部置为0.

3. 执行技巧

- 遗传算子中融合专门领域知识
  - 举例：背包问题：混合遗传算法

$$\max f(X) = \sum_{i=1}^{n} c_i x_i$$

$$s.t.$$

$$x_i \in \{0,1\}, \quad i = 1,2,...,n$$

$$\sum_{i=1}^{n} w_i x_i \le v$$

$n = 50; \quad v = 1000$

$C = \{220,208,198,192,180,180,165,162,160,158,$
$155,130,125,122,120,118,115,110,105,101,$
$100,100,98,96,95,90,88,82,80,77,75,73,72,$
$70,69,66,65,63,60,58,56,50,30,20,15,10,8,$
$5,3,1\}$

$W = \{80,82,85,70,72,70,66,50,55,25,50,55,40,48,$
$50,32,22,60,30,32,40,38,35,32,25,28,30,22,$
$50,30,45,30,60,50,20,65,20,25,30,10,20,25,$
$15,10,10,10,4,4,2,1\}$

3. 执行技巧

- 遗传算子中融合专门领域知识
  - 举例：背包问题：混合遗传算法
  - 结果：总价值：3103；总重量：1000

3. 执行技巧

- 增加局部搜索
  - 举例：货郎担问题(Travelling Salesman Problem, TSP)
  - 设有n个城市，城市i和城市j之间的距离为*d(i,j)*。TSP问题是寻找最短的一条回路，要求该回路能够遍访每个城市且每个城市仅访问一次。

3. 执行技巧

增加局部搜索

- 举例：货郎担问题(Travelling Salesman Problem, TSP)

$$\min\left\{\sum_{i=1}^{n-1} d(C_i, C_{i+1}) + d(C_n, C_1)\right\}$$

（1）编码：整数编码，将城市的全排列作为个体编码；
（2）适应度：目标函数（总路径最短）的倒数；
（3）交叉：顺序交叉；
（4）变异：两点变异；
（5）选择：轮盘赌选择；
（6）局部搜索：在每个个体（解）的附近小邻域内寻找局部最优解，若找到，则更新个体；