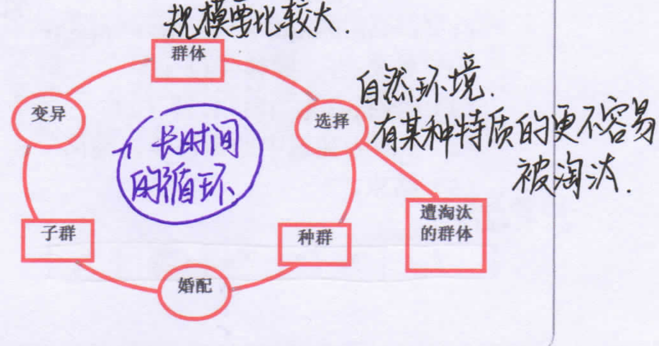
高级搜索2.

9

遗传算法

- 达尔文进化论：“物竞天择、适者生存”
- 70年代由美国的密执根大学的Holland教授首先提出
- 遗传算法作为一种有效的工具，已广泛地应用于最优化问题求解之中。

生物进化与遗传算法



生物进化与遗传算法之间的对应关系

生物进化中的概念	遗传算法中的作用
环境	适应函数
适应性	适应函数值
适者生存	适应函数值最大的解被保留的概率最大
个体	问题的一个解
染色体	解的编码
基因	编码的元素
群体	被选定的一组解
种群	根据适应函数选择的一组解
交叉	以一定的方式由双亲产生后代的过程
变异	编码的某些分量发生变化的过程

针对某一个个体

对应关系.

遗传算法的三个主要操作

- 选择 优胜劣汰
- 交叉 繁衍后代
- 变异 基因变异.

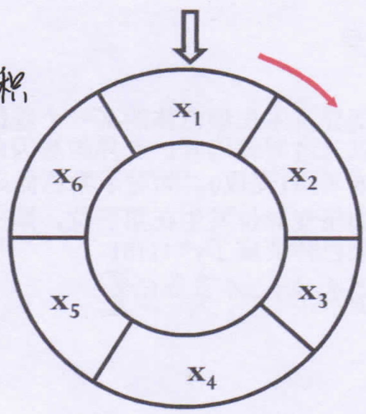
选择

- “轮盘赌”法：
设群体的规模为N，$F(x_i)(i=1, ..., N)$是其中N个染色体的适应值。则第i个染色体被选中的概率由下式给出：

$$p(x_i)=\frac{F(x_i)}{\sum_{j=1}^{N}F(x_j)}$$

模拟优胜劣汰.

每个x的面积有大有小.



模拟“轮盘赌”算法

只random一次. 可运行多次抽出多个. (有放回的抽取).

（1）r=random(0, 1)，i=0，s=$p(x_0)$；
（2）如果s≥r，则转（4）；
（3）s=s+$p(x_i)$，i=i+1, 转（2）
（4）x_i即为被选中的染色体，输出i
（5）结束。

把轮盘展开

x_1	x_2	x_3	x_4	x_5	x_6

↑ r

用得少但调试时可能用 等情况.

“确定性”法

给每个个体匹配一个染色体.

对于规模为N的群体，一个选择概率为$p(x_i)$的染色体x_i被选择次数的期望值$e(x_i)$：

$$e(x_i) = p(x_i)N$$

对于群体中的每一个x_i，首先选择 $\lfloor e(x_i) \rfloor$ 向下取整. 次。这样共得到 $\sum_{i=1}^{N} \lfloor e(x_i) \rfloor$ 个染色体。然后按照$e(x_i)-\lfloor e(x_i) \rfloor$从大到小对染色体排序，依次取出 $N-\sum_{i=1}^{N} \lfloor e(x_i) \rfloor$ 个染色体，这样就得到了N个染色体。 缺2个则给$e(x_i)-\lfloor e(x_i) \rfloor$最大和第二大的$x_i$各添一个.

给了N则每个x_i对应的个体数确定.

交叉

随机挑对.

- 交叉发生在两个染色体之间，由两个被称之为双亲的父代染色体，经交叉以后，产生两个具有双亲的部分基因的新的染色体。当染色体采用二进制形式编码时，交叉过程是以这样一种形式进行的：

随机产生一个交叉位置.

交叉位置

	交叉前		交叉后	
父	$a_1\ a_2 \cdots a_i \mid a_{i+1} \cdots a_n$	⇨	$a_1\ a_2 \cdots a_i\ b_{i+1} \cdots b_n$	孩子1
母	$b_1\ b_2 \cdots b_i \mid b_{i+1} \cdots b_n$		$b_1\ b_2 \cdots b_i\ a_{i+1} \cdots a_n$	孩子2.

变异

- 变异发生在染色体的某一个基因上，当以二进制编码时，变异的基因由0变成1，或者由1变成0。如对于染色体x=11001，如果变异位发生在第三位，则变异后的染色体变成了y=11101。

随机产生一个变异位置.

遗传算法

问题越复杂N越大. 产生后代的概率较大.

（1）给定群体规模N，交叉概率p_c和变异概率p_m，t=0; 较小. 多少代.
（2）随机生成N个染色体作为初始群体；
（3）对于群体中的每一个染色体x_i分别计算其适应值$F(x_i)$； 没有特别的规定…(一般是t>某确定值).
（4）如果算法满足停止准则，则转（10）；
（5）对群体中的每一个染色体x_i计算概率；
（6）依据计算得到的概率值，从群体中随机的选取N个染色体，得到种群； 选择.

并更新到目前为止的max. Loop.

不是一共N个吗…

看第三页的第5张PPT.
N个染色体可能取值,
然后为N个个体都匹配一个染色体.
可能多个个体匹配同一个染色体,
也有可能有染色体没有被任何一个个体匹配.

参与交叉的就不进入新群体了.

（7）依据交叉概率p_c从种群中选择染色体进行交叉，其子代进入新的群体，种群中未进行交叉的染色体，直接复制到新群体中；

（8）依据变异概率p_m从新群体中选择染色体进行变异，用变异后的染色体代替新群体中的原染色体；

（9）用新群体代替旧群体，t=t+1，转（3）；

第x代.

（10）进化过程中适应值最大的染色体，经解码后作为最优解输出；不一定产生于最后一代中.

是在整个进化过程中的最大值.

（11）结束。

收敛性定理：

如果在代的进化过程中，遗传算法每次保留到目前为止的最好解，并且算法以交叉和变异为其随机化操作，则对于一个全局最优化问题，当进化代数趋于无穷时，遗传算法找到最优解的概率为1。

例：求函数的最大值

$$f(x) = x^2$$

其中x为[0, 31]间的整数

编码：采用二进制形式编码

由于x的定义域是[0, 31]间的整数，刚好可以用5位二进制数表示，因此可以用5位二进制数表示该问题的解，即染色体。如00000表示x＝0，10101表示x＝21，11111表示x＝31等

其实二进制未必直接与对应的十进制直接对应,只需与染色体的可能取值有一一对应关系即可.

- 适应函数：
  直接使用函数f(x)作为适应函数。
- 假设群体的规模N＝4，交叉概率p_c＝100%，变异概率p_m＝1%。
- 设随机生成的初始群体为：
  01101，11000，01000，10011
- 选择方法：“确定性”法

第0代情况表

序号	群体	适应值	选择概率（%）	期望次数	选中次数
1	01101	169	14.44	0.58	1
2	11000	576	49.23	1.97	2
3	01000	64	5.47	0.22	0
4	10011	361	30.85	1.23	1

x_b=11000，f(x)=576

第0代种群的交叉情况

序号	种群	交叉对像	交叉位	子代	适应值
1	01101	2	4	01100	144
2	11000	1	4	11001	625
3	11000	4	2	11011	729
4	10011	3	2	10000	256

第1代情况表

序号	群体	适应值	选择概率（%）	期望次数	选中次数
1	01100	144	8.21	0.33	0
2	11001	625	35.62	1.42	1
3	11011	729	41.56	1.66	2
4	10000	256	14.60	0.58	1

x_b=11011，f(x)=729

第1代种群的交叉情况

序号	种群	交叉对像	交叉位	子代	适应值
1	11001	2	3	11011	729
2	11011	1	3	11001	625
3	11011	4	1	10000	256
4	10000	3	1	11011	729

第1代种群的编译情况

序号	种群	交叉对像	交叉位	子代	适应值
1	11001	2	3	11011	729
2	11011	1	3	11101 (变异)	841
3	11011	4	1	10000	256
4	10000	3	1	11011	729

第2代情况表

序号	群体	适应值	选择概率（%）	期望次数	选中次数
1	11011	729	28.53	1.14	1
2	11101	841	32.92	1.32	1
3	10000	256	10.02	0.40	1
4	11011	729	28.53	1.14	1

x_b=11101，f(x)=841

第2代种群的交叉情况

序号	种群	交叉对像	交叉位	子代	适应值
1	11011	2	3	11001	625
2	11101	1	3	11111	961
3	10000	4	4	10001	289
4	11011	3	4	11010	676

第3代情况表

序号	群体	适应值	选择概率（%）	期望次数	选中次数
1	11001	625	24.50	0.98	1
2	11111	961	37.67	1.51	2
3	10001	289	11.33	0.45	0
4	11010	676	26.50	1.06	1

x_b=11111，f(x)=961

第3代种群的交叉情况

序号	种群	交叉对像	交叉位	子代	适应值
1	11001	2	3	11011	729
2	11111	1	3	11101	841
3	11111	4	4	11110	900
4	11010	3	4	11011	729

第4代情况表

序号	群体	适应值	选择概率（%）	期望次数	选中次数
1	11011	729			
2	11101	841			
3	11110	900			
4	11011	729			

x_b=11111，f(x)=961

是第三代里的.

- 前面例题的输出？

输出的不是最后一代中的最好结果，而是进化过程中得到的最好结果。

x_b=1111，f(x)=961

遗传算法的特点

- （1）遗传算法是一个随机搜索算法，适用于数值求解具有多参数、多变量、多目标等复杂的最优化问题。
- （2）遗传算法对待求解问题的指标函数没有什么特殊的要求，比如不要求诸如连续性、导数存在、单峰值假设等。甚至于不需要显式的写出指标函数。
- （3）在经过编码以后，遗传算法几乎不需要任何与问题有关的知识，唯一需要的信息是适应值的计算。也不需要使用者对问题有很深入的了解和求解技巧，通过选择、交叉和变异等简单的操作求解复杂的问题，是一个比较通用的优化算法。
- （4）遗传算法具有天然的并行性，适用于并行求解。

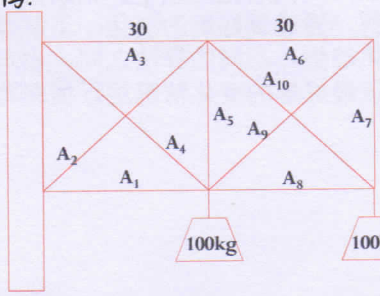
交叉、选择等可以并行.

遗传算法的实现问题

- 编码
- 评价
- 适应函数
- 交叉规则
- 停止条件

编码举例：十杆桁架问题

墙.



怎么优化A1到A10,让它们刚好能支撑重物.

假设每个杆的截面积在0.1至10之间，在该范围内，有16个可能的取值。这样我们可以用4位二进制向量表示截面积的可能取值，其中0000表示0.1，1111表示10，余下的14位二进制向量表示其他的截面积的可能取值。这样10个杆，共用40位二进制向量表示一个十杆桁架问题的染色体。

不一定是真实的二进制.

例：

A1 A2

0010 1110 0001 0011 1011

0011 1111 0011 0011 1010

A9 A10.

编码举例：旅行商问题

- 对于n个城市的旅行商问题，可以用一个矩阵来表示一个可能解。

$$
\begin{array}{c|cccc}
 & 1 & 2 & 3 & 4 \\
\hline
A & 0 & 1 & 0 & 0 \\
B & 1 & 0 & 0 & 0 \\
C & 0 & 0 & 0 & 1 \\
D & 0 & 0 & 1 & 0
\end{array}
$$

去4个城市的次序.

B→A→D→C

- 如果按行展开该矩阵，则该可能解可以用一个4×4的二进制向量表示为：

0100100000010010

不太合理.

二进制表示存在的问题

采用这样的表示方法，对于n城市的旅行商问题，至少需要用n×n位二进制向量表示一个可能的旅行路线。一个n×n位二进制向量，所有可能的编码个数为 $2^{n\times n}$，而一个对称的n城市旅行商问题的可能解个数为n!/2，只占编码个数非常小的比例。以n=10为例，编码个数为可能解个数的7.0×10²³倍。可能解在整个状态空间中，是非常稀疏的，交叉和变异所产生的是大量的非可能解。

A-B与B-A距离一样.

每行每列各只有一个1.

遗传算法的效果就不是很好.

遗传算法的评价

- 定理4给出了当进化代数趋于无穷时，遗传算法找到最优解的概率为1。即保证了遗传算法的收敛性。但在实际计算时，希望随时了解遗传算法的进展情况，监视算法的变化趋势。

可以人为观察，如果下降了可以暂停看看是否饱和.

① 当前最好法

- 该方法在每一代进化过程中，记录得到的最好解，通过最好解的变化，了解算法的变化趋势。不同的算法之间，也可以通过该最好解的变化情况进行横向比较。

是从第1代到目前为止的最好吗?

② 在线比较法

- 该方法用当前代中染色体的平均指标函数值来刻划算法的变化趋势。计算方法如下：

$$v_{on\_line} = \frac{1}{T}\sum_{t=1}^{T} f(t)$$

其中T为当前代中染色体的个数 。