

遗传算法的特点及其应用

● 省、市： 上海市
● 学 校： 复旦附中
● 姓 名： 张 宁

- ✿ 遗传算法的基本概念
- ✿ 简单的遗传算法
 - 选择、交换、变异
- ✿ 遗传算法应用举例
 - 子集和问题
 - TSP(旅行商) 问题
- ✿ 结束语



遗传算法的基本概念

- ✿ 遗传算法（ **Genetic Algorithms** ，简称 **GA** ）根据适者生存，优胜劣汰等自然进化规则来进行搜索计算和问题求解。

对许多用传统数学难以解决或明显失效的复杂问题，特别是优化问题， **GA** 提供了一个行之有效的新途径。



简单的遗传算法

- ❁ **GA** 把每一个可能的解编码为一个向量，称为一个染色体，向量的每一个元素称为基因。所有染色体组成群体。并按预定的目标函数对每个染色体进行评价，根据其结果给出一个适应度的值。
- ❁ 算法开始时先随机地产生一些染色体，计算其适应度，根据适应度对诸染色体进行选择、交换、变异等遗传操作，剔除适应度低的染色体，留下适应度高的染色体。
- ❁ 由于新群体的成员是上一代群体的优秀者，因而在总体上优于上一代。**GA** 就这样反复迭代，直至满足某种预定的优化指标。上述 **GA** 的工作过程可用图1简要描述。



选择运算使用比较普遍的一种是适应度比例法。其实就是将适应度值视为其权值，权值大的被选中的概率也大。它与各染色体适应度成比例。某一染色体被选中的概率为

$$P_c = f(x_c) / \sum f(x_i)$$

式中 \mathbf{x}_i 为种群中第 \mathbf{i} 个染色体对应的数字串， $\mathbf{f}(\mathbf{x}_i)$ 是第 \mathbf{i} 个染色体的适应度值， $\sum f(x_i)$ 是种群中所有染色体的适应度值之和。显然，此法要求染色体的适应度应为正值。



交 换

复制操作虽然能够从旧种群中选择出优秀者，但不能创造新的染色体，因此，遗传算法的开创者提出了交换操作。即：在匹配集中任选两个染色体，随机选择一点或多点交换点位置，交换双亲染色体交换点右边的部分，即可得到两个新的染色体数字串。



变异运算用来模拟生物在自然界的遗传环境中由于各种偶然因素引起的基因突变，它以很小概率随机地改变遗传基因的值。在染色体以二进制编码的系统中，它随机地将染色体的某一个基因由 **1** 变成 **0**，或由 **0** 变成 **1**。通过变异操作，可以使搜索能在尽可能大的空间中进行，获得质量较高的优化解答。



遗传算法应用举例

✿ 子集和问题

✿ **TSP**(旅行商) 问题



GA 在子集和问题上的应用

- ❁ 子集和问题 **SUBSET_SUM**：给定正整数集合 **S** 和一个整数 **t**，判定是否存在 **S** 的一个子集使得 **S'** 中整数的和为 **t**。
- ❁ 我们已知道该问题是一个 **NP**-完全问题。在实际应用中，我们常遇到的是最优化子集和问题。在这种情况下，我们要找出 **S** 的一个子集 **S'**，使得其和不超过 **t**，但又尽可能接近于 **t**。



子集和问题

- ❁ 下面用遗传算法来解决：
- ❁ 我们可以用 n 位二进制数来表示每个染色体。每一位，用 0 、 1 表示是否属于子集。

我们将染色体所表示的子集的元素和与所给 t 的差异记为适应度，即令染色体 \mathbf{x} 的每一位为 \mathbf{x}_i ，所表示元素的值为 \mathbf{s} ，则 $\sum_{x_i \neq 0} s_i - t$

但是经过实践后发现由于适应度相对差异较小，使得适应度非常接近，难以区分染色体的优劣，使得遗传进化变得非常缓慢，且 $\mathbf{f}(\mathbf{x})$ 可能为负值，因此还需对适应度函数做一下变换，才可以适合本题的要求。

令 $\mathbf{f}(\mathbf{k})$ 为当前群体中所有染色体适应度的最大值

$$\mathbf{f}'(\mathbf{x}) = |\mathbf{f}(\mathbf{k}) - \mathbf{f}(\mathbf{x})|$$

所以适应度为 $\mathbf{f}'(\mathbf{x})$ 。



子集和问题

- ❁ 选择时可以用前面所介绍的适应度比例法，但可能会因为偶然情况使得优秀的染色体没有子孙。因此，我在这里采用确定性选择法，先计算群体中每个串的生存概率 $f_i = \frac{P_i}{\sum_{j=1}^n P_j}$ ， $1 \leq j \leq n$ ，然后计算期望复制数 $e_i = P_i * n$ ，式中： n 为群体中染色体的数目。根据 e_i 的值给每个染色体串分配一个复制数。
- ❁ 交换运算与前述相同，不过若进行单点交换有可能使得两个染色体在交换时产生的差异过大，使得遗传变得不稳定，优秀的染色体不能遗传到下一代。因此可以采用多点交换。
- ❁ 变异运算时，只需注意变异概率的取值，至于具体算法如前面所述。



子集和问题

- ✿ 在本题中的一些数值不妨取值如下：
- ✿ 种群长度（染色体个数）：**20**
- ✿ 选择概率：**0.9**
- ✿ 变异概率：**0.1**
- ✿ 结束条件：当前最优解在 **100** 代遗传后仍未改变，或已取到最优解



TSP(旅行商)问题

GA 在 TSP (旅行商) 问题求解中的应用

- ✿ 设存在 N 个城市, D_{ij} 表示城 i 与城 j 之间的距离, $D_{ij}=D_{ji}$, 现在要求一条遍历所有 N 个城市, 且不走重复路的最短路径 (最短哈密尔顿圈)。
- ✿ 这是一个典型 **NP-** 完全问题。传统解法对此都并不太奏效下面我们试着用遗传算法来解决这道题目。



TSP(旅行商)问题

我们先采用十进制编码，每个染色体由按一定顺序排列的 **N** 个城市的序号组成，表示一条可能的旅行路径。适应度为一条旅行路径对应的距离，路径越短的染色体适应度越高。例如，取 **N=10**，城市代号为 **1** 至 **10**。

例如种群中的染色体：**2 8 4 10 5 1 7 3 6 9**

表示一条旅行路径

2→8→4→10→5→1→7→3→6→9→2 其总路径长
 $\sum D_{ij} = D_{28} + D_{84} + D_{410} + D_{105} + D_{51} + D_{17} + D_{73} + D_{36} + D_{69} + D_{92}$

我们可以采用非负变换，把最小化优化目标函数变换为以最大值为目标的适应度函数，可以如下定义：

$$f(x) = c_{\max} - \sum D_{ij}$$

其中 **c_{max}** 为可以取为进化过程中路径长度的最大值，或者为了保证 **f(x)** 为正而预先设定为一个与种群无关的常数。



TSP(旅行商)问题

- ✿ 关于选择运算，可以考虑前面介绍的确定性选择法。
- ✿ 此处的交换运算不同于前，因为两个染色体，若进行简单的交换运算，可能会使得染色体所表示路径中会重复经过同一城市，即同一染色体中的两个基因有着相同的城市编号。因此须改进交换运算。我们可以采用部分匹配交换运算（**PMX**）。

当然也可以不进行交换运算，而直接进行变异运算，因为本例中的变异运算本就隐含了交换的含义。

- ✿ 变异操作与二进制编码时不同，从群体中随机抽取一个染色体，随机抽取两个基因，将两者交换，即颠倒城市次序。
- ✿ 算法的主要部分已经讨论完了，但是还有一点值得提出的，由于遗传算法是一种不断优化的搜索算法，因此，我们可以用贪心算法构造初始群。



TSP(旅行商)问题

- ❁ 题目中的一些数值不妨取值如下：
- ❁ 种群长度（染色体个数）：**20**
- ❁ 变异概率：**0.1**
- ❁ 结束条件：当前最优解在 **100** 代遗传后仍未改变。



遗传算法应用举例



通过两道例题，我们在遗传算法原有简单定义上，加以扩充，介绍了若干高级遗传算法在具体实例中的应用，旨在打开思想的局限性，不仅仅在原有简单定义下做文章，而是充分发挥想象，对于不同问题，采取不同对策，在遗传算法的框架下，安排使用合理的算法，改进原有算法，或与原有算法相结合。这样，才能充分发挥遗传算法的长处解决问题。



结 束 语

- ❁ 遗传算法的原理是简单的，但是如何熟练运用遗传算法却并不是一个简单的问题，理论要切合实际，对于不同问题，遗传算法要稍加变化，就如同画龙点睛一般，切忌不可生搬硬套。我认为遗传算法应当与现有优化算法结合，可产生比单独使用遗传算法或现有优化算法更好，更实用的算法。
- ❁ 本文的主要目的还是让大家对遗传算法能够有一个初步的了解，这样对大家进行深入的实践是有帮助的。希望大家通过本文的指导，能够将遗传算法熟练应用在各个方面。



图

一

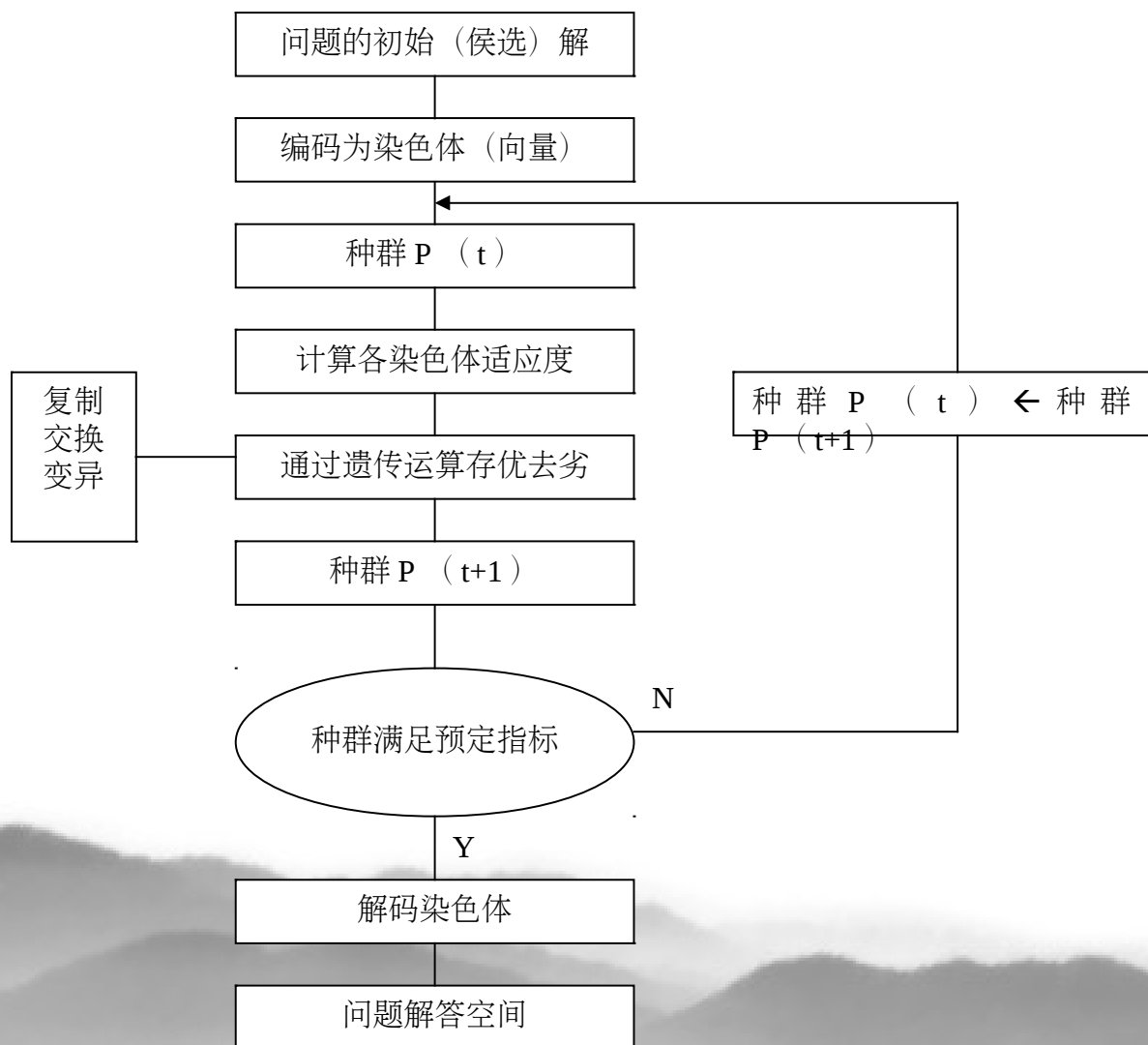


图 1 遗传算法工作原理示意图



多点交换

- ❁ 不过在本例中使用多点交换，只需进行两点交换即可，其实两点交换与单点交换是类似的。
- ❁ 设有两个父辈染色体 **A** 和 **B**：

A： 10100100101110

B： 01001110110001

设两个交换点选择如下：

A： 10100|10010|1110

B： 01001|11011|0001

则两点交换运算就是交换染色体 **A** 和染色体 **B**，两个交换点之间的部分，则交换结果如下：

A'： 10100110111110

B'： 01001100100001



部分匹配交换运算

- 部分匹配交换运算先在两父染色体串中各产生两个交换点，把这两点之间的区域定义为匹配区域，再对两个匹配区域中的基因通过对应匹配置换。例如，两父染色体串为：

A: 2 8 4 10 * 5 1 7 3 * 6 9

B: 5 6 7 1 * 10 2 8 3 * 9 4

符号“*”表示交换点。

以染色体 **A** 为例，其匹配区域中有四个元素 **5**，**1**，**7**，**3**，与染色体串 **B** 中的四个元素 **10**，**2**，**8**，**3**，逐一匹配，即通过在染色体串 **A** 内进行元素间的置换，使得匹配区域内这四个元素换为 **10**，**2**，**8**，**3**。为此，染色体串 **A** 中的元素。作如下置换：

