

5 重组

进化算法中的重组有时俗称为“交叉”，但系统地看，重组包含了交叉。重组算法是改进进化算法最有效的环节，它通过结合交配群体中包含的遗传信息产生新的个体。因为进化算法中有二进制编码、实值编码、排列编码、树编码等，因此必须也有与编码方式相适应的不同的重组算法。

重组概率：

如果某个重组算子又称为“交叉”，那么“重组概率”在其看来可以称作“交叉概率”。重组概率主要有 2 种，一种是**染色体重组概率**，即对于任意一组配对的染色体而言，其发生重组的概率；另一种是**条件重组概率**，又称**片段重组概率**，它是指在满足染色体重组概率的条件下，重组算子在染色体上发生作用的最小片段（“最小作用片段”）将要发生重组的概率。与变异算子的变异概率不同，不同的重组算子所侧重强调哪一种重组概率会有所区别，在使用的时候要看清楚是指哪一种概率。

有关“最小作用片段”详细的相关定义如下：

定义 1. 给定一个重组算法，当它作用在染色体上的某片段时，如果该片段不能再细分成互相独立的多个片段，那么称这个片段为该重组算子在染色体上的**最小作用片段**。

例如，在模拟二进制交叉中，该交叉算子作用与染色体的最小片段为染色体上的每一位元素。因此它的“最小作用片段”的长度为 1。在该交叉算子中，最小作用片段对应的交叉概率固定地设为 0.5，因此在该交叉算子中，侧重强调的是第一种交叉概率，即染色体发生交叉的概率。

又例如，在差分进化算法的二项式分布交叉中，交叉概率指的是上文所说的第二种，即最小作用片段发生交叉的概率。而第一种概率则固定地设为 1。

某些交叉算子还会设立一种额外的交叉概率，比如均匀分布交叉。它里面不研究上文所说的两种交叉概率（即把它们都固定地设为 1），而是设立了额外的一个交叉概率：比如交叉得到的 child 1 的染色体的某一位有 0.5 的概率等于 parent 1 染色体的相同位置的元素；另外 0.5 的概率等于 parent 2 染色体的相同位置的元素。

下面介绍几种经典的重组算法：

1) 重组算法的代表——离散重组算法 (Discrete recombination):

离散重组算法在个体间执行变量值的交换，在生成交配个体时，交配个体中每个变量可以等概率地挑选一个父个体对应变量作为自身的值。其几何特征表现如下，离散重组产生了父代所在的超立方体的角：

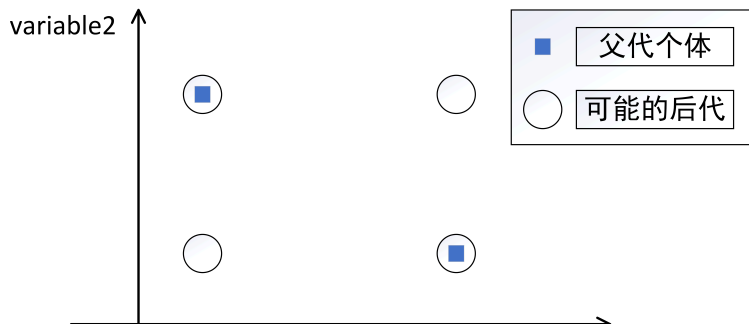


图 1 离散重组产生的可能的后代在解空间上的位置

考虑以下两个个体，每个个体有 3 个变量 (3 维)：

父个体 1: 13 24 5

父个体 2: 124 3 24

生成的 child 1 和 child 2 的染色体分别可以是：124 24 24 和 13 24 5。

与离散重组非常相似的是均匀分布交叉。但稍有不同的是在均匀分布交叉中，一旦确定了交叉后的 child 1 的染色体的元素值，那么 child 2 的染色体元素值也随之确定。

2) 实数值重组 (Real valued recombination):

实值重组算法可以实现对元素为实数值的染色体之间的重组，它包括中间重组、线性重组、扩展线性重组、模拟二进制交叉等。

2.1) 中间重组：

中间重组是一种仅适用于实数变量个体的重组算法。这里后代的变量值是在父辈变量的区间上选择的。生成子代个体的公式如下：

$$Var_i^{Child} = Var_i^{Parent1} \cdot \alpha_i + Var_i^{Parent2} \cdot (1 - \alpha_i), i \in [1, 2, ..., N]$$

其中， α_i 是 $[-d, 1 + d]$ 之间的随机数，它是一个随机均匀选择的比例因子。

参数 d 的值代表可能产生的后代的区域大小。 $d = 0$ 表示后代的变量值的区域大小与父代是一样的，此时称为“(标准的) 中间重组”。但是，由于后代的大多数变量不是在可能区域的边界上生成的，因此变量所覆盖的面积有可能会越来越小。因此，仅用 $d = 0$ 的标准中间重组就会发生这种变量空间收缩现象。因此，通过设置更大的 d 值可以防止这种现象。一般设置 $d = 0.25$ ，此时可以在统计学上保证后代的变量值的范围不会缩小。如图所示：

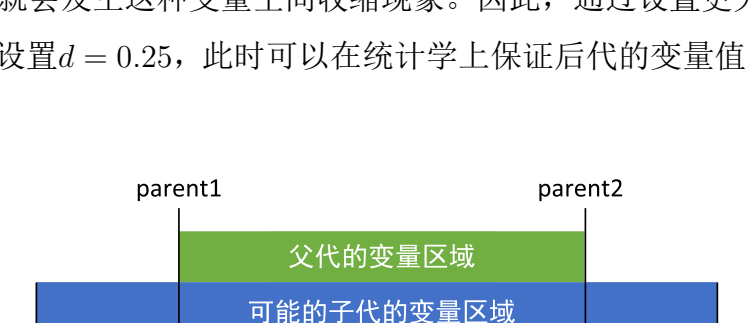


图 2 中间重组下父代与子代的变量值的区域范围比较

例如父个体染色体为：

父个体 1: 0.4 1.2 -0.3

父个体 2: 0.2 0.7 0.6

生成的子个体染色体可以是：0.3 0.9 0.4。

中间重组能够稍微超出父代所在的超立方体的边界，如图所示：

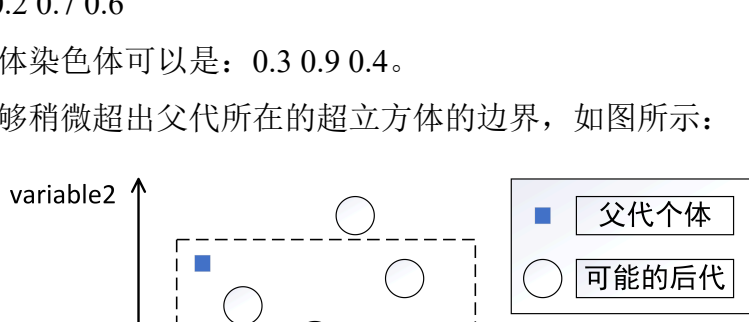


图 3 中间重组后可能的子代变量在解空间的位置

2.2) 线性重组：

线性重组类似与中间重组。产生子代个体的公式如下：

$$Var_i^{Child} = Var_i^{Parent1} \cdot \alpha + Var_i^{Parent2} \cdot (1 - \alpha), i \in [1, 2, ..., N]$$

对比中间重组可见，线性重组只有一个 α ，即所有个体的 α 是一样的。并且 α 同样是 $[-d, 1 + d]$ 之间的随机数，代表随机均匀选择的比例因子。

对于 d 的解析跟中间重组是一样的。

线性重组产生的后代，其变量是在父代所在的直线上，如图所示：

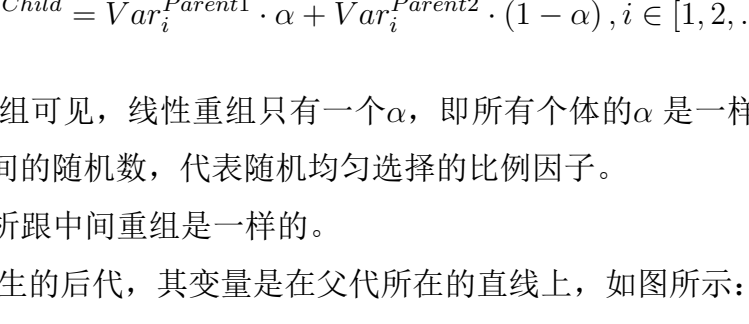


图 4 线性重组后可能的子代变量在解空间的位置

另外还有扩展线性重组，其详细介绍可以参见文献 [1]。

3) 模拟二进制交叉：

模拟二进制交叉 (Simulated Binary Crossover, SBX)[3] 由父代 x_a 和 x_b 生成以下后代：

$$\left. \begin{aligned} y_1(k) &\leftarrow \frac{1}{2} [(1 + \beta_k) x_a(k) + (1 - \beta_k) x_b(k)] \\ y_2(k) &\leftarrow \frac{1}{2} [(1 - \beta_k) x_a(k) + (1 + \beta_k) x_b(k)] \end{aligned} \right\}, 1 \leq k \leq n$$

其中 n 为染色体长度， β_k 是由下面的概率密度函数生成的随机数：

$$\text{PDF}(\beta) = \begin{cases} \frac{1}{2} (\eta + 1) \beta^\eta, 0 \leq \beta \leq 1 \\ \frac{1}{2} (\eta + 1) \beta^{-(\eta + 2)} \beta > 1. \end{cases}$$

其中 η 是任意的非负实数。 η 的取值必须不小于 0。可以用下面的公式得到 β ：

$$r \leftarrow U[0, 1], \beta \leftarrow \begin{cases} (2r)^{\frac{1}{\eta + 1}}, r \leq 0.5, \\ (2 - 2r)^{-\frac{1}{\eta + 1}}, r > 0.5 \end{cases}$$

当然也可以采用其他方法来得到 β ，比如采用服从标准正态分布的随机数，此时模拟二进制交叉就变成了“正态分布交叉”[4]。

细心观察不难发现，为什么“模拟二进制交叉”和“正态分布交叉”都被称为是“交叉”，而前面的“中间重组”、“线性重组”称为“重组”，但除了中间过程所用到的方法不同外，它们整体上看又比较相似，且有区别与单纯的值互换重组 (即严格意义上的交叉)。此外，如果翻看了下一节 (变异)，会发现差分变异从整体上看也跟上面的若干重组算法有着很大的相似之处，但差分变异被称作是变异而非重组。实际上，差分变异加上交叉算子可以被称为一种重组算子，有些流派甚至直接称差分变异为一种交叉算子，这其实是可以接受的。

如果要严格对这些算子分类，那目前来看是没多大必要的，在很多文献中“重组”和“交叉”两个概念是相互混用的，至于差分变异与重组，尽管有着很大相似性，但若真要究其本质，重组也可以算作一种变异，因此本文档也不对其进行重新划分，继续把差分变异归为变异，以沿袭传统的说法。而对于“模拟二进制交叉”、“正态分布交叉”以及类似的非单纯值互换的重组，尽管其原名是“交叉”，但本文仍把它们归类为“重组”而非“交叉”。而对于“离散重组”，它虽然是值互换的重组，本应当严格归类为“交叉”，但考虑到重组包含交叉这一理念，因此继续把其归类为“重组”。事实上，一切交叉算子都可称其为重组算子。甚至一切重组算子也可以不太严谨地直接称其为交叉算子，这都是可以接受的。

4) 值互换重组——交叉 (Values exchanged recombination——crossover):

这种重组方式就是两个父代个体交换染色体片段产生新个体的过程，因此也称为交叉操作。

交叉操作中，个体的染色体编码可以是实值、也可以是二进制，比如：

[0 1 0 1 1 0 1] 和 [1.1 2.0 3.4 2.7 1.6] 这些类型的染色体都可以进行交叉操作，一般更多地用在二进制或格雷码编码的个体中。

交叉有单点、双点和多点交叉。它们是根据交叉点的数量分类的。其中多点交叉的示意图如下：

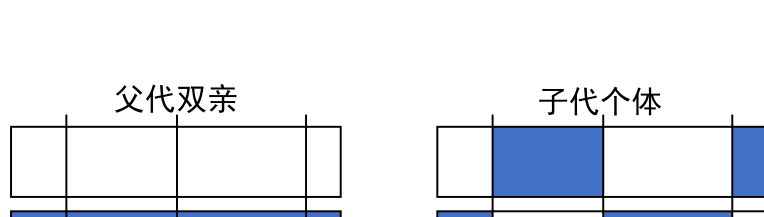


图 5 多点交叉

此外还有均匀交叉、洗牌交叉，这里就不一一赘述了。这里要介绍一下“减少代理”交叉和部分匹配交叉：

4.1) 减少代理交叉 (Crossover with reduced surrogate):

上面的交叉算法的交叉结果可能会产生和父代性状一样的个体，如果在遗传算法中想让交叉得到的子代中更多的个体拥有与父代个体不一样的性状，这时就可以用减少代理的交叉算法。

减少代理交叉算法尽可能地产生全新性状的个体，这是通过限制交叉点的位置来实现的——控制交叉点只出现在父代两个交叉个体的基因值不同的地方。

4.2) 部分匹配交叉 (PMX, partially matched crossover):

对于组合优化问题，染色体中每一位的值都是独一无二的，称为“排列编码”。这意味着不能使用简单的交叉算法。PMX 可基于单点交叉也可基于两点交叉。本文档默认它是基于两点交叉的，因此它先随机产生两个交叉点，定义两点之间的基因片段为匹配区域，交换 2 个父代的匹配区域的基因，并对匹配区域外出现的重复基因进行替换。详见文献 [2]

文献 [2] 中还详细讲述了循环交叉 (CX, cycle crossover)、顺序交叉 (OX1, order crossover)、基于顺序的交叉 (OX2, order-based crossover)、基于位置的交叉 (POS, position-based crossover)、投票重组交叉 (VR, voting recombination crossover) 以及位置交替交叉 (AP, alternating-position crossover)。在文中能进一步找到其详细算法的描述文档，这里就不一一赘述了。

参考文献

[1] Muhlenbein H . The Breeder Genetic Algorithm-a provable optimal search algorithm and its application[C]// Applications of Genetic Algorithms, IEE Colloquium on. IEEE Xplore, 1994.

[2] Khan I H. Assessing Different Crossover Operators for Travelling Salesman Problem[J]. International Journal of Intelligent Systems Technologies & Applications, 2015, 7(11):19-25.

[3] Agrawal R B , Deb K , Agrawal R B . Simulated Binary Crossover for Continuous Search Space[J]. Complex Systems, 2000, 9(3):115-148.

[4] ZHANG Min. Research on evolutionary algorithms for constrained optimization and multiobjective optimization[D]. Hefei: University of Science and Technology of China, 2008.