

6 变异

变异是指通过改变染色体中的一部分元素来形成新的染色体的过程。它能够提高种群的多样性，降低进化算法陷入局部最优解的风险。

变异概率：

变异概率  $p$  是用于控制变异发生几率的变量，变异概率主要有 2 种，一种是**染色体变异概率**，即对于任意一条染色体而言，其发生变异的概率（默认为 1）；另一种是**条件变异概率**，又称**片段变异概率**，它是指在满足染色体变异概率的条件下，变异算子在染色体上发生作用的最小片段（“最小作用片段”）将要发生变异的概率。一般来说进化算法只考虑第二种变异概率，(前一种变异概率默认为 1)，因此第二种变异概率我们简称其为“变异概率”，其详细的相关定义与定理如下：

**定义 1.** 给定一个变异算法，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变异算子在染色体上的**最小作用片段**。

例如：对于二进制染色体的变异算子，它是作用在染色体的一个位（即一个“比特”）——将 0 变 1 或 1 变 0。因此染色体的位就是这种变异的最小作用片段。

又例如：对于两点互换变异算子，当它作用在一个排列编码的染色体上时，从染色体上随机选择两个位进行元素互换，因此该变异的最小作用片段为整条染色体。

**定理 1.** 变异算子最小作用片段的长度由具体的变异算子决定，这个长度可以是固定的也可以是不固定的。

**定理 2.** 给定一个变异算法，其在染色体上的最小作用片段可以有多个，长度可以唯一也可以不唯一。

**定义 2.** 给定一个变异算法，在其最小作用片段上发生这种变异的概率称为该变异算子的**片段变异概率**，进化算法中主要侧重强调这类变异概率，因此把它简称“变异概率”。

**定理 3.** 给定一个变异算法，其所有的最小作用片段对应的变异概率可以是相同的也可以是不相同的。

**定义 3.** 给定一个变异算法，把其有多少个发生了变异的最小作用片段的期望值称为该染色体的**变异期望次数**。

例如：对于多项式变异算子，假如想要一条染色体上有 2 个位的元素发生多项式变异，那么此时的变异期望次数是 2。

**定理 4.** 给定一条染色体和一个变异算子，该染色体的变异期望次数等于所有最小作用片段对应的变异概率之和。

例如：多项式变异算子常常会默认设定变异概率等于一个整数/染色体长度。这个整数其实就等于染色体发生多项式变异的位数的期望值。

下面介绍几种常见的变异算法。

1) 二进制染色体突变：

在二进制进化算法中，变异非常简单，只需根据变异概率反转染色体的每一位元素(0->1 或 1->0) 即可。

2) 连续进化算法的变异：

连续进化算法是指染色体的每一位是实数。此时变异的方法非常多，常见的有：

2.1) 均匀变异：

均匀变异是指变异的结果服从均匀分布。它包括以当前值为中心点的均匀变异以及以搜索域中央为中心点的均匀变异。均匀变异提供一个  $\alpha$  参数来调节变异的大小。 $\alpha$  确定了均匀变异的范围。

2.2) 高斯变异：

高斯变异是指变异的结果服从正太分布。它包括以当前值为中心点的高斯变异以及以搜索域中央为中心点的高斯变异。高斯变异提供一个  $\sigma$ (正态分布的标准差) 来控制变异的大小。但它并不像均匀变异那样可以严格限制变异的范围，在大样本情况下，变异结果落在中心点附近长度为  $\alpha$  的邻域的概率约为 68.27%；落在中心点附近长度为  $3\alpha$  的邻域的概率约为 99.73%。

2.3) 多项式变异：

在多项式变异中，当前元素值以一定概率加上一个服从多项式概率分布的值得到一个邻近值，详见文献 [1]。

2.4) 差分变异：

差分变异操作将一个可缩放且随机选择的向量差分量加到基向量中。下面的式子展示了将三个不同且随机选择的向量结合以产生一个变异向量  $v_{i,g}$ ：

$$v_{i,g} = x_{r_0,g} + F \cdot (x_{r_1,g} - x_{r_2,g})$$

其中  $F$  是缩放因子，值取正实数，一般取 (0,1] 之间的随机数，它可以控制种群的进化率。

$r_0$  是基向量索引，它可以通过多种选择方式来确定 (详见前面的“选择”一节)。 $r_0$  一般是随机选取的且不同于目标向量索引  $i$  的索引。差分向量的索引  $r_1$  和  $r_2$  也彼此互不相同，并且它们也不同于基向量和目标向量的索引。下面是在一个 2 维决策空间中构造变异向量  $v_{i,g}$  的示意图：

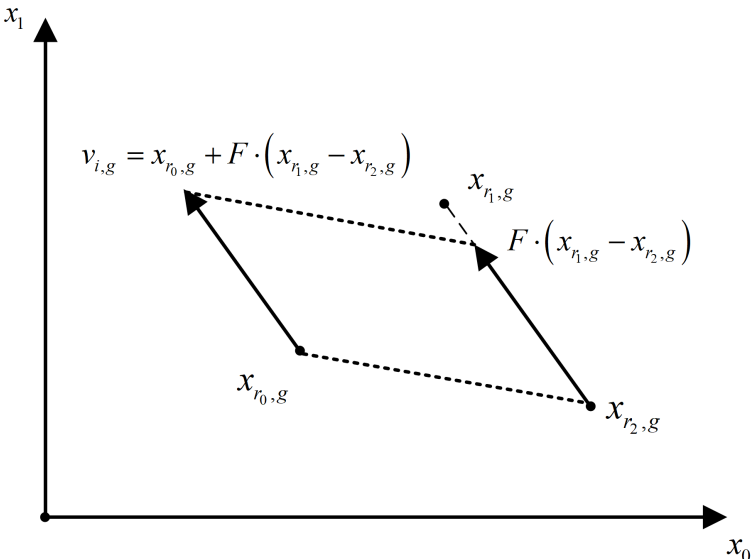


图 1 差分变异示意图

差分变异跟上面的变异算子有个很大的不同之处是它需要参照不止一个个体染色体来进行变异。这跟重组是极为相似的，因此有些流派直接称差分变异为一种交叉算子。本文档认为差分变异实质上是重组概率为 1 的三父体重组算法，因为若加上差分变异之后所需要进行的按概率和父代个体的交叉操作之后，整体上看就是一个完全意义上的重组算子了。正因为单是就差分变异而言，它不需要重组概率，因此仍将其归类为“变异”。

3) 整数值突变：

对于元素为整数值的染色体的突变，一种普遍的做法是先按实数值进行突变，然后四舍五入进行取整。另外一种方法是专门设计变异长度为整数且服从一定分布的整数值变异算子。但后者通用性不强，而且往往需要跟实际问题结合起来专门设计相应的算子，才有更好的表现。

4) 互换突变 (Exchanged mutation)：

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”。这意味着进行突变操作后依然要保持染色体的这个特征。互换突变算法在染色体上随机确定两个基因片段并进行互换。这两个片段必须包含相同数量的基因。如图所示：

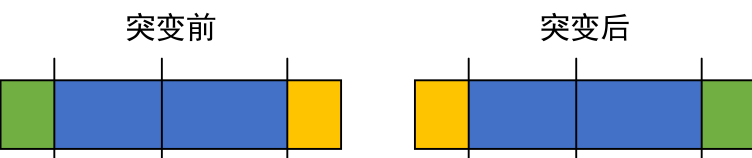


图 2 互换突变

除了这些经典突变方法外，另外在进化策略中，还有能够调整步长的突变方法，以及一些自适应的变异算子，在这里就不展开介绍了。

参考文献

[1] K. Deb and M. Goyal, A combined genetic adaptive search (GeneAS) for engineering design, Computer Science and informatics, 1996, 26: 30-45.