第六章　遗传算法的数学理论

6.1　模 式 定 理

6.1.1　模式

遗传算法通过对群体中多个个体的迭代搜索来逐步找出问题的最优解。这个搜索过程是通过个体之间的优胜劣汰、交叉重组和突然变异等遗传操作来实现的，在这个搜索过程中，哪种个体更容易生存，哪种个体更容易被淘汰掉呢？

从第一章中所给出的求 $f(x_1, x_2) = x_1^2 + x_2^2$ 的最大值这个例子来看，4 个初始个体经过一代遗传和进化运算之后得到 4 个新的个体，如图 6-1 所示。

个体编码串	适应度
011101	34
101011	34
011100	25
111001	50
平均适应度	35.75

第 1 代群体

个体编码串	适应度
011101	34
111111	98
111001	50
111010	53
平均适应度	58.75

第 2 代群体

图 6-1　群体进化过程的一个片段

通过对上述过程的观察可以看出，新一代个体的编码串组成结构与其父代个体的编码串组成结构之间有一些相似的结构联系。如第 2 代群体中的个体 111001、111010，与其父代个体之一 111001 在编码串的前半部分数码一致、结构相似，并且该父

代个体的适应度要高于群体中全部个体的平均适应度。

由此我们可以看到，遗传算法处理了一些具有相似编码结构模板的个体。若把个体作为某些相似模板的具体表示的话，对个体的搜索过程实际上就是对这些相似模板的搜索过程。这样，就需要引入一个描述这种相似模板的新概念——模式（Schema）。

【定义 6.1】模式表示一些相似的模块，它描述了在某些位置上具有相似结构特征的个体编码串的一个子集。

不失一般性，以二进制编码方式为例，个体是由二值字符集 $V=\{0, 1\}$ 中的元素所组成的一个编码串，而模式却是由三值字符集 $V_1=\{0, 1, *\}$ 中的元素所组成的一个编码串，其中“*”表示通配符，它既可被当作“1”，也可被当作“0”。

例如，模式 $H=11**1$ 描述了长度为 5，且在位置 1、2、5 取值为“1”的所有字符串的集合 {11001, 11011, 11101, 11111}；模式 $H=00***$ 描述了由 8 个个体所组成的集合 {00000, 00001, 00010, …, 00111}；而模式 $H=11011$ 所描述的个体集合是由它自身组成的，即 {11011}。由这些例子可以看出，模式的概念使得我们可以简明地描述具有相似结构特点的个体编码字符串。

在进行遗传算法的理论分析时，有时需要估算模式的数量。在一个编码字符串中往往隐含着多种不同的模式，定义在长度为 l 的二进制编码字符串上的模式共有 3^l 个，更为一般地，定义在含有 k 个基本字符的字母表上的长度为 l 的字符串中的模式共有 $(k+1)^l$ 个，在长度为 l、规模为 M 的二进制编码字符串群体中，一般包含有 $2^l\sim M\cdot 2^l$ 个模式。另一方面，不同的模式所能匹配的字符串的个数也是不同的。

在引入模式概念之后，遗传算法的本质是对模式所进行的一系列运算，即通过选择算子将当前群体中的优良模式遗传到下一代群体中，通过交叉算子进行模式的重组，通过变异算子进行模式的突变。通过这些遗传运算，一些较差的模式逐步被淘汰，而一些较好的模式逐步被遗传和进化，最终就可得到问题的最优

解。

为定量地估计模式运算，下面再引入两个概念：模式阶和模式定义长度。

【定义 6.2】 在模式 H 中具有确定基因值的位置数目称为该模式的模式阶（Schema Order），记为 $o(H)$。

对于二进制编码字符串而言，模式阶就是模式中所含有的 1 和 0 的数目，例如，$o(10**0**)=3$，$o(*******1)=1$。当字符串的长度固定时，模式阶数越高，能与该模式匹配的字符串（称为样本）数就越少，因而该模式的确定性也就越高。

【定义 6.3】 模式 H 中第一个确定基因值的位置和最后一个确定基因值的位置之间的距离称为该模式的模式定义长度（Schema Defining Length），记为 $\delta(H)$。

例如，$\delta(11*0**)=3$，$\delta(0***1)=4$。而对于 $H=****1$、$H=0*****$、$H=******1*$之类的模式，由于它们只有一位确定的基因值，这个位置既是第一个确定基因值位置，也是最后一个确定基因值位置，所以规定它们的模式定义长度为 1，如 $\delta(**0****)=1$。

6.1.2 模式定理

由前面的叙述我们可以知道，在引入模式的概念之后，遗传算法的实质可看作是对模式的一种运算。对基本遗传算法（SGA）而言，也就是某一模式 H 的各个样本经过选择运算、交叉运算、变异运算之后，得到一些新的样本和新的模式。

假设在进化过程中的第 t 代时，当前群体 $P(t)$ 中能与模式 H 匹配的个体数（样本数）记为 $m(H,t)$，下一代群体 $P(t+1)$中能与模式 H 匹配的个体数记为 $m(H,t+1)$。下面对基本遗传算法在选择算子、交叉算子和变异算子的连续作用下，模式 H 的样本数 $m(H,t)$ 的变化情况进行分析。

1. 选择算子的作用

基本遗传算法中的选择运算使用的是比例选择算子。将当前

群体中适应度的总和记为 $F(t)=\sum_{i}F(A_i)$ 在这个算子的作用下，与模式 H 所匹配的各个个体 A_i 能够平均复制 $M\cdot F(A_i)/F(t)$ 个个体到下一代群体中，即：

$$\begin{aligned} m(H,t+1) &= \sum_{A_i\in H\cap P(t)} \frac{M\cdot F(A_i)}{F(t)} \\ &= \sum_{A_i\in H\cap P(t)} \frac{M\cdot f(H,t)}{F(t)} \\ &= m(H,f)\frac{M\cdot f(H,t)}{F(t)} \\ &= m(H,f)\frac{f(H,t)}{\bar{F}(t)} \end{aligned} \tag{6-1}$$

式中，$f(H,t)$ 是第 t 代群体中模式 H 所隐含个体的平均适应度；$\bar{F}(t)=F(t)/M$ 是第 t 代群体的平均适应度。

若再假设模式 H 的平均适应度总是高出群体平均适应度的 C 倍，则式（6-1）可改写为：

$$m(H,t+1)=m(H,t)\cdot(1+C) \tag{6-2}$$

由此可见，$m(H,t)$ 为一等比级数，其通项公式为：

$$m(H,t)=m(H,0)\cdot(1+C)^t \tag{6-3}$$

由式（6-3）可知，

●若 $C>0$，则 $m(H,t)$ 呈指数级增长；

●若 $C<0$，则 $m(H,t)$ 呈指数级减少。

由此可得到下述结论：在选择算子作用下，对于平均适应度高于群体平均适应度的模式，其样本数将呈指数级增长；而对于平均适应度低于群体平均适应度的模式，其样本数将呈指数级减少。

2. 交叉算子的作用

这里以常用的单点交叉算子为例进行研究。

假设有如图 6-2 所示的一个模式：

隐含在该模式中的样本与其他个体进行交叉操作时，根据交叉点的位置不同，有可能破坏该模式，也有可能不破坏该模式而

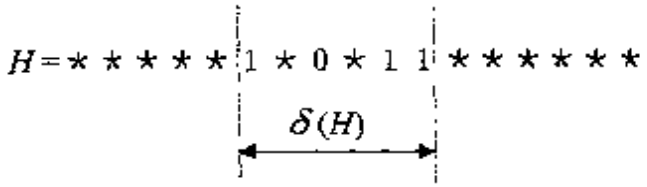

图 6-2　模式及其定义长度

使其继续生存到下一代群体中。下面估算该模式生存概率 p_s 的下界。

显然，当随机设置的交叉点在模式的定义长度之内时，将有可能破坏该模式（当然，根据与之交叉的配对个体所属模式情况也可能不破坏该模式）；而当随机设置的交叉点在模式的定义长度之外时，肯定不会破坏该模式。再考虑到交叉操作本身是以交叉概率 p_c 发生的，所以模式 H 的生存概率下界为：

$$p_s \geqslant 1 - p_c \cdot \delta(H) / (l-1) \tag{6-4}$$

这样，经过选择算子和交叉算子作用之后，模式 H 的样本数满足下式：

$$m(H, t+1) \geqslant m(H, t) \cdot (1+C) \cdot \left[1 - p_c \cdot \frac{\delta(H)}{l-1}\right] \tag{6-5}$$

由式（6-5）可知，在其他值固定的情况下（$C>0$），

- $\delta(H)$ 越小，则 $m(H, t)$ 越容易呈指数级增长；
- $\delta(H)$ 越大，则 $m(H, t)$ 越不容易呈指数级增长。

3. 变异算子的作用

这里以常用的基本位变异算子为例进行研究。

此时，若某一模式被破坏，则必然是模式描述形式中通配符“*”之处的某一基因值发生了变化，其发生概率是：

$$1 - (1 - p_m)^{o(H)}$$

当 $p_m \ll 1$ 时，有：

$$1 - (1 - p_m)^{o(H)} \approx o(H) \cdot p_m$$

由此可知，在变异算子作用下，模式 H 的生存概率大约是：

$$p_s \approx 1 - o(H) \cdot p_m \tag{6-6}$$

由式（6-6）可知：

●o（H）越小，模式 H 越易于生存；

●o（H）越大，模式 H 越易于被破坏。

这样，综合上述式（6-1）、式（6-4）、式（6-6），并忽略一些极小项，则在比例选择算子、单点交叉算子、基本位变异算子的连续作用下，群体中模式 H 的子代样本数为：

$$m(H, t+1) \geqslant m(H, t) \cdot \frac{f(H, t)}{\bar{F}(t)} \cdot \left[1 - p_c \cdot \frac{\delta(H)}{l-1} - o(H) \cdot p_m\right] \tag{6-7}$$

由式（6-7）就可得到下述定理[3]：

【模式定理】 遗传算法中，在选择、交叉和变异算子的作用下，具有低阶、短的定义长度，并且平均适应度高于群体平均适应度的模式将按指数级增长。

模式定理阐述了遗传算法的理论基础，它说明了模式的增加规律，同时也给遗传算法的应用提供了指导作用。

6.2 积木块假设与遗传算法欺骗问题

6.2.1 积木块假设

模式定理说明了具有某种结构特征的模式在遗传进化过程中其样本数将按指数级增长，这种模式就是具有低阶、短的定义长度，且平均适应度高于群体平均适应度的模式。这种类型的模式被称为基因块或积木块（Building Block）。

之所以称之为积木块，是由于遗传算法的求解过程并不是在搜索空间中逐一地测试各个基因的枚举组合，而是通过一些较好的模式，像搭积木一样，将它们拼接在一起，从而逐渐地构造出适应度越来越高的个体编码串。

模式定理说明了积木块的样本数呈指数级增长，亦即说明了用遗传算法寻求最优样本的可能性，但它并未指明遗传算法一定

能够寻求到最优样本。而积木块假设却说明了遗传算法的这种能力。

【积木块假设】个体的基因块通过选择、交叉、变异等遗传算子的作用，能够相互拼接在一起，形成适应度更高的个体编码串。

积木块假设说明了用遗传算法求解各类问题的基本思想，即通过基因块之间的相互拼接能够产生出问题更好的解。基于模式定理和积木块假设，就使得我们能够在很多应用问题中广泛地使用遗传算法的思想。

需要说明的是，虽然积木块假设并未得到完整而严密的数学证明，但大量的应用实践说明了其有效性。

6.2.2 遗传算法欺骗问题

提到积木块假设，这里就必须说明一下遗传算法欺骗问题[75,76]（GA Deceptive Problem）。应用实践表明，存在着一类用遗传算法难以求解的问题，这类称为“GA-难”的问题往往不满足积木块假设，即由基因块之间的拼接，往往会欺骗遗传算法，使其进化过程偏离最优解。

各种研究结果表明，属于“GA-难”的问题一般包含有孤立的最优点，即在这个最优点周围是一些较差的点，从而使得遗传算法较难通过基因之间的相互拼接而达到这个最优点的模式。实际上，目前也尚无解决这类问题的较好的方法或策略。所幸的是，现实所遇到的各种应用问题中，很少有这种奇怪的性质。

6.3 隐含并行性

在遗传算法的运行过程中，每代都处理了 M 个个体，但由于一个个体编码串中隐含有多种不同的模式，所以算法实质上却是处理了更多的模式。

以二进制编码符号串为例，长度为 l 的编码串中隐含有 2^l 种模式，这样，规模为 M 的群体中就可能隐含有 $2^l \sim M \cdot 2^l$ 种