第7章　遗传算法

7.1　问题概述

应用微分学方法、运筹学方法求解最优化问题已经很广泛，如产品排产问题、运输调度问题、最优库存量问题，等等。这些方法在变量较少、目标与约束条件较少的情况是有效的。对于多目标下，问题的变量较大与约束条件较多的情况，应用这些方法将存在一定的局限性，甚至由于难以满足条件可能出现无解。例如，多目标下多种证券的最优组合投资问题，多目标下多个生产工艺参数的最优控制问题，都涉及到求一组较优参数的问题，并以此作为计划或控制的依据。特别地，许多问题在全局最优解难以获取时，实际只需要求满意解就可以解决问题，此时遗传算法就显示了较好的优势。

7.2　遗传算法概述

7.2.1　基本概念

遗传算法的基础是基于达尔文的进化论和 Mendel 的遗传学说。达尔文进化论最重要的是适者生存原理。它认为每一物种在发展中越来越适应环境。物种每个个体的基本特征由后代所继承，但后代又会产生一些异于父代的新变化。在环境变化时，只有那些能适应环境的个体特征方能保留下来。

Mendel 遗传学说最重要的是基因遗传原理。它认为遗传以密码方式存在细胞中，并以基因形式包含在染色体内。每个基因有特殊的位置并控制某种特殊性质。所以，每个基因产生的个体对环境具有某种适应性。基因突变和基因杂交可产生更适应于环境的后代。经过存优去劣的自然淘汰，适应性高的基因结构得以保存下来。

由于遗传算法是由进化论和遗传学机理而产生的直接搜索优化方法；故而在这个算法中要用到各种进化和遗传学的概念。一些主要概念如下：

（1）串（String）：它是个体（Individual）的形式，在算法中为二进制串，并且对应于遗传学中的染色体（Chromosome）。

（2）群体（Population）：个体的集合称为群体，串是群体的元素。

（3）群体规模（Population Size）：在群体中个体的数量称为群体规模，又称群体的大小。

(4) 基因(Gene):基因是串中的元素,基因用于表示个体的特征。例如有一个串 S=1011,则其中的 1,0,1,1 这四个元素分别称为基因,它们的值称为等位基因(Alletes)。

(5) 基因位置(Gene Position):一个基因在串中的位置称为基因位置,有时也简称基因位。基因位置由串的左边向右计算,例如在串 S=1101 中,0 的基因位置是 3。基因位置对应于遗传学中的地点(Locus)。

(6) 串结构空间 S^S:在串中,基因任意组合所构成的串的集合。基因操作是在结构空间中进行的。串结构空间对应于遗传学中的基因型(Genotype)的集合。

(7) 参数空间 S^P:这是串空间在物理系统中的映射,它对应于遗传学中的表现型(Phenotype)的集合。

(8) 适应度(Fitness):表示某一个体对于环境的适应程度。

遗传算法的主要思想是:把在解空间寻找最优解的问题,通过编码形式把解转换为生物空间对应的染色体(或个体),应用生物空间遗传机理,如交叉、变异等,对染色体或基因进行遗传进化,使进化后的染色体具有好性质,再对染色体解码转换为解空间对应的解,使这样的解逐步逼近局部乃至全局最优解。解空间与生物空间的对应如图 7.1 所示。

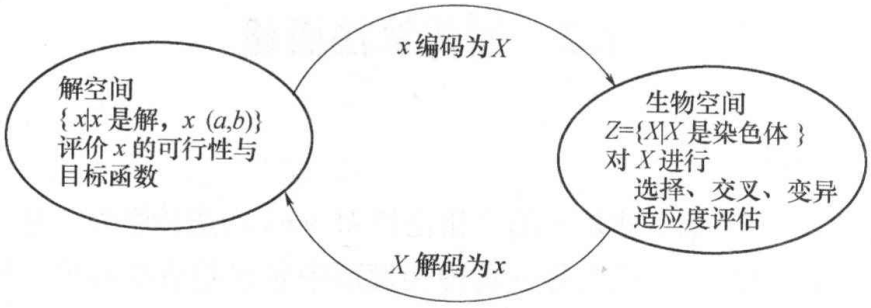

图 7.1 解空间与生物空间的对应

7.2.2 标准遗传算法

标准遗传算法(Standard GA,SGA)的基本处理流程如图 7.2 所示,是一种群体型操作。该操作以群体中的所有个体为对象,选择、交叉和变异是遗传算法区别于其他传统寻优方法的三个主要操作算子,它们构成了所谓的遗传操作。遗传算法中包含了五个基本要素:变量编码、初始群体的设定、适应度函数的设计、遗传操作设计和参数设定,这五个要素构成遗传算法的核心内容。

1) 编码

通过编码将解空间的数据表示成遗传空间的基因型串结构数据。编码一般有二进制编码和实数编码。对于问题解 X,$X=(x_1,x_2,\cdots,x_n)$,二进制编码是把 X 用 0,1 串表示,而实数编码是把 X 用一向量表示,即 $X=(x_1,x_2,\cdots,x_n)$,x_i 是实数。编码设计应适合要解决的问题,应考虑完全性、封闭性、可扩展性和复杂性等。完全性是指分

布在所有问题域的解都可能被构造出来;封闭性是指每个基因编码对应一个可接受的个体,不产生无效的个体;可扩展性是指对于具体问题,要考虑编码形式与大小影响下的解码时间;复杂性是指整体考虑基因型的结构复杂性、解码复杂性、计算复杂性等。

2) 初始群体的生成

在遗传算法处理流程中,继编码设计后的任务是初始群体的生成,并以此为起点一代代进化直到满足某种进化停止准则终止进化过程,初始群体也称为进化的初始代,即第一代。初始群体的个体一般可采用随机产生,一般群体可表示为 $Z=\{X_i \mid X_i=(x_{i1},x_{i2},\cdots,x_{in}),i=1,2,\cdots N,\}$,即 X_i 是染色体或个体,x_i 是基因或位。若是实数编码,则 $x_i \in (a_i,b_i)$,若是二进制编码编码,则 x_i 取值 1 或 0 。

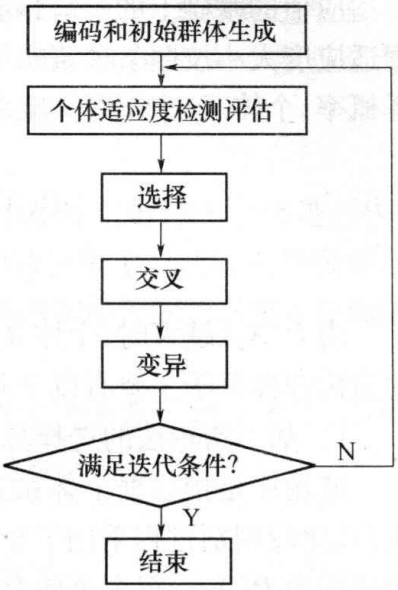

图 7.2 遗传算法的基本流程

3) 适应度函数(评估函数)

适应度函数设计是模拟自然选择,进行遗传进化操作的基础,它的评估是遗传操作的依据。适应度函数值即适应度。由于下面定义的选择概率以适应度为基础,因此适应度是非负的。几种常见的适应度函数定义为

对 $f(x)$,取 $C_{\min}$,$C_{\max}$ 使 $C_{\min}<\min\limits_{x}\{f(x)\}$,$C_{\max}>\max\limits_{x}\{f(x)\}$,$C_{\max}>0$

(1) 求 $f(x)$ 最大值问题:如果对应解 x 的个体为 X,可以定义个体 X 的适应度为

$$\mathrm{Fit}(X)=\begin{cases}-C_{\min}+f(x), f(x)>C_{\min}\\ 0, \text{其他}\end{cases}$$

或

$$\mathrm{Fit}(X)=\frac{1}{C_{\max}-f(x)}$$

(2) 求 $f(x)$ 最小值问题:可以定义个体 X 的适应度为

$$\mathrm{Fit}(X)=\begin{cases}C_{\max}-f(x), f(x)<C_{\max}\\ 0, \text{其他}\end{cases}$$

或

$$\mathrm{Fit}(X)=\frac{1}{-C_{\min}+f(x)}$$

7.2.3 选择算子

从群体中选择优胜的个体,淘汰劣质个体的操作称为选择。它是建立在群体中个

体适应值的基础上的，其目的是把优胜的个体传遗传到下一代，选择操作的实现是根据适应度大小按照某种策略从父代中挑选个体进入中间群体。选择算子设计依赖选择概率，个体 X_i 选择概率定义为

$$P(X_i) = \frac{\mathrm{Fit}(X_i)}{\sum_{j=1}^{N} \mathrm{Fit}(X_j)}, i = 1,2,\cdots,N$$

当 $P(X_i)$ 越大时，个体 X_i 被选择遗传（复制）到下一代的可能性越大。目前常用的遗传选择算子主要有以下几种：

1）基于赌轮法的选择算子

赌轮法是指根据个体被选择概率大小确定相应个体是否被遗传（复制）到下一代，其比较判别过程采用了轮盘赌的思想。设种群有 n 个个体 $X_1, X_2, \cdots, X_n$，X_i 的选择概率为 $P(X_i)$，每个个体对应 $P(X_i)$ 表示为赌轮上的某个区域，按个体数 n 转动赌轮 n 次，根据赌轮停止点区域对应的个体进行选择，个体对应赌轮区域越大被选择机会越大，计算个体被选择的数量，这些个体将按选择的数量被复制。在计算机辅实现过程中，模拟赌轮一般采用以下方法：根据个体的排序，按选择概率 $P(X_i)$ 计算积概率 $q_i = \sum_{j=1}^{i} P(X_j)$，则 $P(X_i) = q_i - q_{i-1}$，产生 n 个随机数 r_k，对每一个 r_k，若 $q_{i-1} < r_k < q_i$，则复制 X_i，可以得到选择复制后的 n 个新一代个体。可以看到，适应值越大染色体被选中（复制）的机会也越大。

2）期望值方法

把每一个体的适应度与平均适应度进行比较，以确定该个体在下一代的复制数即每个个体在下一代生存的期望数目为

$$M(X_i) = \frac{\mathrm{Fit}(X_i)}{\frac{\sum_{j=1}^{n} \mathrm{Fit}(X_j)}{n}} = \frac{n\mathrm{Fit}(X_i)}{\sum_{j=1}^{n} \mathrm{Fit}(X_j)} = nP(X_i)$$

对 M 进行取整，令 M_1 为取整后的个体复制数，则

$$M_1 = \begin{cases} [M] + 1, M - [M] > 0.5 \\ [M], M - [M] \leqslant 0.5 \end{cases}$$

由于取整问题，复制个体总数可能与原个体总数存在偏差，偏差数可以根据个最大、最小适应度进行增加或减少。

这一类选择算子虽然得到了广泛应用，但仍有其不合理之处。首先，要求适应函数大于零，由于对实际目标函数的域值无法预知，从而无法预先确定能使函数值正的转换函数，实践表明转换函数的选择对算法的性能影响也很大；另外，在基于适

值比例的选择下，一个具有很高适应值的个体很容易大量繁殖，从而导致过早收敛到局部解；最后，如果群体中各个染色体的适应值差异不大，则它们后代的个数也会基本相同，好的个体得不到更多繁殖的机会，导致算法收敛的速度变慢。

3）排序选择方法

该方法在计算每个个体的适应值后，根据适应值大小顺序对群体中个体进行排序，然后把事先设计好的概率表按序分配给个体作为各自的选择概率。此时选择概率与适应值无直接关系而仅与序号有关，其不足之处在于选择概率和序号的关系需事先确定。

4）联赛选择方法

这类算子也是基于适应值大小顺序的，但不要求适应值函数一定大于零，即可以求函数极大值问题，也可以求函数极小值问题，其基本过程如下：

（1）从上一代等概率选取 q 个染色体。

（2）从这 q 个染色体中选取最好的放入下一代群体中。

（3）重复以上各步，直到新一代群体中也包含 N 个染色体。

对于这种选择算子，一个染色体被选择的概率只取决于它在群体中的大小顺序，该染色体适应值比其他染色体大或者小多少不影响其后代的数量，从而在一定程度上避免了经选择后染色体过于集中的现象。

7.2.4 交叉算子

交叉是把两个父代个体的部分结构加以重组而生成新个体的操作。交叉的作用，是使新的群体中的个体具有多样性，扩大解的搜索空间，使个体对应的解逐步逼近局部最优解。

交叉算子设计一般要考虑三个问题：

（1）交叉概率 P_c 的确定。

（2）在 $P_c<1$ 的情况下，判别两个个体是否要交叉。

（3）对交叉的个体采用何种形式交叉。

交叉概率 P_c 的选择，一般可取 $P_c=0.5$，也可以采用方法：在指定交叉形式下，若有 P_{c0}，使进化代数最小达到终止条件，则交叉概率 P_{c0} 最优。

在 n 个个体组成的种群和给定交叉概率 P_c 的条件下，需要交叉的个体数为 $m=n\times P_c$，每代交叉时，随机抽取个体 X_i 与 X_j，产生(0,1)区间的随机数 r，如果 $r<P_c$，则表示 X_i 与 X_j 要交叉，否则不交叉，这样判别直至需要交叉个体数为 m 时停止。

在两个体需要交叉时，可采用指定交叉形式进行交叉。常用的交叉形式有以下几种：

1）单点交叉

单点交叉又叫简单交叉。具体操作是：在个体基因串中随机设定一个交叉点。实行交叉时，该点前或后的两个个体的部分结构进行互换，并生成两个新个体。当基因链码的长度为 n 时，可能有 $n-1$ 个交叉点位置。

设双亲为：$x=(x_1,x_2,\cdots,x_n)$，$y=(y_1,y_2,\cdots,y_n)$，在随机的第 K 位交叉，生成的后代为

$x=(x_1,x_2,\cdots,x_k,y_{k+1},y_{k+2},\cdots,y_n)$，$y=(y_1,y_2,\cdots,y_k,x_{k+1},x_{k+2},\cdots,x_n)$

2）两点交叉

与单点交叉类似，只是需设置两个交叉点，然后两个染色体相互交换两点之间的部分，从而生成两个新染色体。一个两点交叉的说明如下：

父辈个体 aaa | aaaa | aaa → aaa | bbbb | aaa

父辈个体 bbb | bbbb | bbb → bbb | aaaa | bbb

对于两点交叉，若染色体长为 n，则可能有 $(n-1)(n-2)/2$ 种交叉点的设置。

3）均匀交叉

均匀交叉则是依概率交换两个父辈个体基因串的每一位。其过程是：先随机的产生一个与父辈个体基因串具有同样长度的二进制串，其中 0 表示不交换，1 表示交换。这个二进制串称为交叉模版；然后根据该模板对两个父辈基因串进行交叉，得到的两个新基因串即为后代新个体。例如：

父辈个体　1：110010111000

父辈个体　2：101011101011

模板：　001101011100

后代个体　1：111011101000

后代个体　2：100010111011

由于均匀交叉在交换位时并不考虑其所在位置，破坏模式的概率较大。但另一方面它能搜索到一些前面基于点的交叉方法无法搜索到的模式。

4）基于实向量算术运算的交叉

设两向量为 x_1、x_2，则有

凸交叉：$x_1'=\lambda_1 x_1+\lambda_2 x_2$

$x_2'=\lambda_1 x_2+\lambda_2 x_1$　　$\lambda_1=\lambda_2=0.5$

仿射交叉：$x_1'=\lambda_1 x_1+\lambda_2 x_2$

$x_2'=\lambda_1 x_2+\lambda_2 x_1$　　$\lambda_1=1.5,\ \lambda_2=-0.5$

线性交叉：$x_1'=\lambda_1 x_1+\lambda_2 x_2$

$x_2'=\lambda_1 x_2+\lambda_2 x_1$　　$\lambda_1+\lambda_2\leqslant 0,\lambda_1>0,\ \lambda_2>0$

7.2.5　变异算子

变异是对群体中的个体的某些基因值的变动。变异的作用是使种群中的某些个体的基因（位）产生突变引入原种群不含有的基因，形成的新个体与其他的个体有所不同。与交叉算子的作用是使对应解逐步逼近局部最优解相比，变异算子作用是使个体对应的解逐步逼近全局最优解。

变异算子设计也要考虑三个问题：

（1）变异概率 P_m 的确定。

（2）在 $P_m<1$ 的情况下，判别个体的某基因是否要变异。

（3）对需变异的基因采用何种形式变异。

变异概率 P_m 一般为 0.01 ~ 0.001，也可以采用方法：在指定变异形式下，若有 P_{m_0}，使进化代数最小达到终止条件，则变异概率 P_{m_0} 最优。

若种群有 n 个个体，且每个体有 N 个基因，给定的变异概率为 P_m，则需要变异的基因数为 $M=n\times N\times P_m$，每代变异时，随机抽取个体 X_i 及某一基因 x_{ik}，产生(0,1)区间的随机数 r，如果 $r<P_m$，则表示 x_{ik} 要变异，否则不变异，这样判别直至需要变异的基因数为 M 时停止，一般地，一个个体变异基因数控制为最多一个是较合理的。

在个体的某基因需要变异时，可采用指定变异形式进行变异。几种常用的变异形式如下：

1）基本变异算子

基本变异算子是针对二值基因链码而言。其具体操作是：对群体中基因链码随机挑选 C 个基因位置并对这些基因位置的基因值以变异概率 P 取反，即 0 变成 1，1 变成 0。当 $C=1$ 时，表示一个基因值取反。

2）均匀变异算子

该变异方法是针对实数编码方式的。为叙述方便，设 $v=(v_1,v_2,\cdots,v_m)$ 是群体中的一个个体，$z=(z_1,z_2,\cdots,z_m)$ 是变异产生的后代。均匀性变异则是先在个体 v 中随机的选择一个分量 v_k，然后，在一个定义的区间 $[a_k,b_k]$ 中均匀随机地取一个数 v_k^1 代替 v_k 以得到 z，即 $z=(v_1,\cdots,v_k^1,\cdots,v_m)$。

3）自适应变异算子

该算子与基本变异算子的操作相似，两者之间的区别就是使用自适应变异算子时变异率不是固定不变而是随着群体中个体多样性的程度自适应调整。

7.2.6 用标准遗传算法求解函数最大值

设 $g(x)=x^2,'x\in[0,31]$，要求用标准遗传算法(SGA)求解 $g(x)$ 的最大值。即求 x_0，使 $g(x_0)=\max\limits_{x\in[0,31]}\{g(x)\}$，由于 $g(x)$ 在[0,31]连续且单调增，所以 $g(x)$ 的最大值应在边界即 $x_0=31$ 达到。下面是应用遗传算法求解，主要是体现在[0,31]中的任意解转换为染色体的选择、交叉等进化后，如何逐步逼近最优解 $x_0=31$ 的过程。遗传过程的步骤以下：

（1）编码　由于 x 的最大取值是 31，因而把变量 x 编码为 5 位长的二进制无符号整数表示形式。比如 $x=12$ 可以表示为 01100 的形式。

（2）产生种群　取群体规模为 4，即群体由 4 个个体组成。表 7.1 左端是随机产生的初始群体的成员。

表 7.1　遗传算法求 $g(x)=x^2$ 最大值

个体号	初始群体（$n=4$）	X	适应度 $f(x)=x^2$	选择概率 $f_i/\Sigma f$	实际计数	交叉点（竖线表示）	配对	交叉位置	新一代群体	X	适应度 $f(x)=x^2$
1	01101	13	169	0.14	1	0110 | 1	2	4	01100	12	144
2	11000	24	576	0.49	2	1100 | 0	1	4	11001	25	625
3	01000	8	64	0.06	0	11 | 000	4	2	11011	27	729
4	10011	19	361	0.31	1	10 | 011	3	2	10000	16	256
适应度总和（Σf）			1170	1.00	4.0						1754
平均适应度（$\Sigma f/n$）			293	0.25	1.0						439
最大适应度			576	0.46	2.0						729

（3）适应度函数定义　由于 $g(x)=x^2$ 是非负的，所以直接定义适应度函数 $\mathrm{Fit}(X)=g(X)$，为方便，记个体 X_i 的适应度为 $f_i=\mathrm{Fit}(X_i)$。评估个体时，适应度评估与解的目标评估（即使 $g(x)$ 最大）相一致，需要把基因型个体转换为解的实数形式 x，以评价解空间中的解，比如 01101 要译码转换为 24.

（4）选择操作　采用期望值方法来进行选择。计算群体中所 4 个体适应度的总和 $\sum_{i=1}^{4} f_i$，再计算每个个体选择概率 $P_i=P(X_i)=\dfrac{f_i}{\sum_{i=1}^{4} f_i}, i=1,2,3,4$，表 7.1 给出了选择 4 个个体的概率。相应的选择结果为：$M(X_1)=4\times P_1=4\times 0.14=0.56$，取整判别后个体 X_1 复制一个个体，即 X_1 被选择遗传到下一代，同理，个体 X_2 复制 2 个，个体 X_3 不复制被淘汰，X_4 复制 1 个。这反应了最优秀的个体获得了最多的生存机会（个体 X_2 被复制 2 个），最差的个体 X_3 被淘汰。被淘汰的个体 X_3 的位置以复制两个个体的 X_2 替代，将进入交叉过程。

（5）交叉操作　采用单点交叉法。表 1 中交叉点为 4，配对库的个体 X_1 与个体 X_2 配对，个体 X_3 与个体 X_4 配对，通过交叉得到两个新个体：01100 和 11001。

（6）由于变异概率与种群数、基因数都小，所以不引入变异。

（7）参数确定　遗传算法自身关键参数有 3 个，即群体大小 n、交叉概率 P_c 和变异概率 P_m。群体大小 n 太小时难以求出最优解，太大则增长收敛时间。一般 $n=30\sim160$。交叉概率 P_c 太小时难以向前搜索，太大则容易破坏高适应值的结构。一般取 $P_c=0.25\sim0.75$。变异概率 P_m 太小时难以产生新的基因结构，太大使遗传算法成了单纯的随机搜索。一般取 $P_m=0.0001\sim0.1$。

可以看到，第一代遗传进化后，尽管仅有选择与交叉，没有变异，但从解 13，24，8，19 已经通过进化，代替为解 12，25，27，16，淘汰了使目标值 $g(x)$ 较小的 8，而新的解 27 比初始解 24 更逼近最优解，$x_0=31$，结果如表 7.1 所列。如果继续进行第二次的遗

传进化，可以看到第二代进化后转换的解为 19，24，25，27，出现了解整体逐步逼近最优解 $x_0=31$ 的趋势，结果如表 7.2 所列。

表 7.2　遗传算法求 $g(x)=x^2$ 最大值，第二代遗传过程

个体号	初始群体（$n=4$）	X	适应度 $f(x)=x^2$	选择概率 $f_i/\Sigma f$	实际计数	交叉点（竖线表示）	配对	交叉位置	新一代群体	X	适应度 $f(x)=x^2$
1	01100	12	144	0.08	0	11|011	2	2	11001	25	625
2	11001	25	625	0.35	1	11|001	1	2	11011	27	729
3	11011	27	729	0.42	2	11|011	4	2	11000	24	576
4	10000	16	256	0.15	1	10|000	3	2	10011	19	361

7.2.7　遗传算法优化神经网络参数

神经网络模型主要通过输入变量的线性组合和非线性映射构造建立，其中模型参数主要是线性组合的连接权值和阈值参数等，模型参数的获取通过对样本的学习影响权值与阈值的修正。如果不采用权值与阈值的修正公式获取最优的权值与阈值参数，而是采用遗传算法获取，可以通过样本数据支持下，以神经网络的计算输出与实际输出的误差最小作为目标函数，设计相应的个体、基因、适应度函数以及选择、交叉、变异三算子，获取适合模型的一组较优参数。一般地，用遗传算法优化神经网络，主要包括三个方面：连接权的优化；网络结构的优化；学习参数的优化。

1. 遗传算法优化 XOR 问题的 BP 连接权值

设 XOR 问题为

$$Y=f(x_1,x_2)=\begin{cases}0, x_1=0, x_2=0\\ 1, x_1=0, x_2=1\\ 1, x_1=1, x_2=0\\ 0, x_1=1, x_2=1\end{cases}$$

XOR 的网络结构及参数如图 7.3 所示。

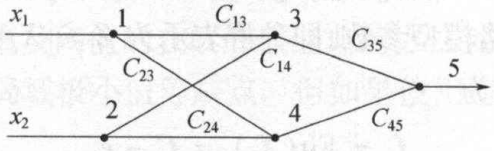

图 7.3　XOR 的网络结构及参数

建立的 BP 模型为

$$O=f\left(\sum_{i=3}^{4}C_{i5}f\left(\sum_{j=1}^{2}C_{ji}x_j-\theta_i\right)-\gamma\right) \tag{7.1}$$

式中：C_{i5}、C_{ji}、θ_i、γ 是模型参数。