

# 第一章 绪 论

## 1.1 遗传算法的生物学基础

生物在自然界中的生存繁衍，显示出了其对自然环境的优异自适应能力。受其启发，人们致力于对生物各种生存特性的机理研究和行为模拟，为人工自适应系统的设计和开发提供了广阔的前景。遗传算法（Genetic Algorithms，简称 GAs）就是这种生物行为的计算机模拟中令人瞩目的重要成果。基于对生物遗传和进化过程的计算机模拟，遗传算法使得各种人工系统具有优良的自适应能力和优化能力。遗传算法所借鉴的生物学基础就是生物的遗传和进化。

### 1.1.1 遗传与变异

世间的生物从其亲代继承特性或性状，这种生命现象就称为遗传（Heredity），研究这种生命现象的科学叫做遗传学（Genetics）<sup>[1]</sup>。由于遗传的作用，使得人们可以种瓜得瓜、种豆得豆，也使得鸟仍然是在天空中飞翔，鱼仍然是在水中遨游。

构成生物的基本结构和功能单位是细胞（Cell）。细胞中含有的一种微小的丝状化合物称为染色体（Chromosome），生物的所有遗传信息都包含在这个复杂而又微小的染色体中。遗传信息是由基因（Gene）组成的，生物的各种性状由其相应的基因所控制，基因是遗传的基本单位。细胞通过分裂具有自我复制的能力，在细胞分裂的过程中，其遗传基因也同时被复制到下一代，从而其性状也被下一代所继承。经过生物学家的研究，现在人们已经明白控制并决定生物遗传性状的染色体主要是由一种叫做脱

氧核糖核酸 (Deoxyribonucleic Acid, 简称 DNA) 的物质所构成, 除此之外, 染色体中还含有很多蛋白质。DNA 在染色体中有规则地排列着, 它是个大分子的有机聚合物, 其基本结构单位是核苷酸。每个核苷酸由四种称为碱基的环状有机化合物中的一种、一分子戊糖和磷酸分子所组成。许多核苷酸通过磷酸二酯键相结合形成一个长长的链状结构, 两个链状结构再通过碱基间的氢键有规律地扭合在一起, 相互卷曲起来形成一种双螺旋结构。另外, 低等生物中还含有一种叫做核糖核酸 (Ribonucleic Acid, 简称 RNA) 的物质, 它的作用和结构与 DNA 类似。基因就是 DNA 或 RNA 长链结构中占有一定位置的基本遗传单位。生物的基因数量根据物种的不同也多少不一, 小的病毒只含有几个基因, 而高等动植物的基因却以数万计, DNA 中, 遗传信息在一条长链上按一定的模式排列, 亦即进行了遗传编码。一个基因或多个基因决定了组成蛋白质的 20 种氨基酸的组成比例及其排列顺序。遗传基因在染色体中所占据的位置称为基因座 (Locus), 同一基因座可能有的全部基因称为等位基因 (Allele)。某种生物所特有的基因及其构成形式称为该生物的基因型 (Genotype), 而该生物在环境中呈现出的相应的性状称为该生物的表现型 (Phenotype)。一个细胞核中所有染色体所携带的遗传信息的全体称为一个基因组 (Genome)。

细胞在分裂时, 遗传物质 DNA 通过复制 (Reproduction) 而转移到新产生的细胞中, 新细胞就继承了旧细胞的基因。有性生殖生物在繁殖下一代时, 两个同源染色体之间通过交叉 (Crossover) 而重组, 亦即在两个染色体的某一相同位置处 DNA 被切断, 其前后两串分别交叉组合而形成两个新的染色体。另外, 在进行细胞复制时, 虽然概率很小, 但也有可能产生某些复制差错, 从而使 DNA 发生某种变异 (Mutation), 产生出新的染色体。这些新的染色体表现出新的性状。如此这般, 遗传基因或染色体在遗传的过程中由于各种各样的原因而发生变化。

### 1.1.2 进化

生物在其延续生存的过程中，逐渐适应于其生存环境，使得其品质不断得到改良，这种生命现象称为进化（Evolution）。生物的进化是以集团的形式共同进行的，这样的—个团体称为群体（Population），组成群体的单个生物称为个体（Individual），每一个个体对其生存环境都有不同的适应能力，这种适应能力称为个体的适应度（Fitness）。达尔文（Darwin）的自然选择学说（Natural Selection）构成了现代进化论的主体<sup>[2]</sup>。自然选择学说认为，通过不同生物间的交配以及其他一些原因，生物的可能发生变异而形成一种新的生物基因，这部分变异了的基因也将遗传到下一代。虽然这种变化的概率是可以预测的，但具体哪一个个体发生变化却是偶然的。这种新的基因依据其与环境的适应程度决定其增殖能力，有利于生存环境的基因逐渐增多，而不利干生存环境的基因逐渐减少。通过这种自然的选择，物种将逐渐地向适应于生存环境的方向进化，从而产生出优良的物种。

### 1.1.3 遗传与进化的系统观

虽然人们还未完全揭开遗传与进化的奥秘，既没有完全掌握其机制，也不完全清楚染色体编码和译码过程的细节，更不完全了解其控制方式，但遗传与进化的以下几个特点却为人们所共识：

（1）生物的所有遗传信息都包含在其染色体中，染色体决定了生物的性状。

（2）染色体是由基因及其有规律的排列所构成的，遗传和进化过程发生在染色体上。

（3）生物的繁殖过程是由其基因的复制过程来完成的。

（4）通过同源染色体之间的交叉或染色体的变异会产生新的物种，使生物呈现新的性状。

（5）对环境适应性好的基因或染色体经常比适应性差的基因或染色体有更多的机会遗传到下一代。

## 1.2 遗传算法简介

遗传算法是模拟生物在自然环境中的遗传和进化过程而形成的一种自适应全局优化概率搜索算法。它最早由美国密执安大学的 Holland 教授提出,起源于 60 年代对自然和人工自适应系统的研究<sup>[3]</sup>。70 年代 De Jong 基于遗传算法的思想在计算机上进行了大量的纯数值函数优化计算实验<sup>[4]</sup>。在一系列研究工作的基础上,80 年代由 Goldberg 进行归纳总结,形成了遗传算法的基本框架<sup>[5]</sup>。

### 1.2.1 遗传算法概要

对于一个求函数最大值的优化问题(求函数最小值也类同),一般可描述为下述数学规划模型:

$$\begin{cases} \max & f(X) \\ \text{s.t.} & X \in R \end{cases} \quad (1-1)$$

$$R \subseteq U \quad (1-2)$$

$$(1-3)$$

式中,  $X = [x_1, x_2, \dots, x_n]^T$  为决策变量,  $f(X)$  为目标函数, 式 (1-2)、(1-3) 为约束条件,  $U$  是基本空间,  $R$  是  $U$  的一个子集。满足约束条件的解  $X$  称为可行解, 集合  $R$  表示由所有满足约束条件的解所组成的一个集合, 叫做可行解集合。它们之间的关系如图 1-1 所示。

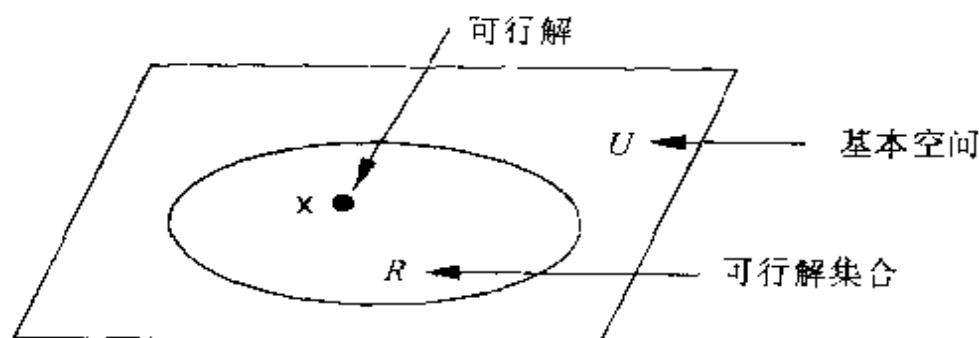


图 1-1 最优化问题的可行解及可行解集合

对于上述最优化问题，目标函数和约束条件种类繁多，有的是线性的，有的是非线性的；有的是连续的，有的是离散的；有的是单峰值的，有的是多峰值的。随着研究的深入，人们逐渐认识到在很多复杂情况下要想完全精确地求出其最优解既不可能，也不现实，因而求出其近似最优解或满意解是人们的主要着眼点之一。总的来说，求最优解或近似最优解的方法主要有三种：枚举法、启发式算法和搜索算法。

(1) 枚举法。枚举出可行解集合内的所有可行解，以求出精确最优解。对于连续函数，该方法要求先对其进行离散化处理，这样就有可能产生离散误差而永远达不到最优解。另外，当枚举空间比较大时，该方法的求解效率比较低，有时甚至在目前最先进的计算工具上都无法求解。

(2) 启发式算法。寻求一种能产生可行解的启发式规则，以找到一个最优解或近似最优解。该方法的求解效率虽然比较高，但对每一个需要求解的问题都必须找出其特有的启发式规则，这个启发式规则无通用性，不适合于其他问题。

(3) 搜索算法。寻求一种搜索算法，该算法在可行解集合的一个子集内进行搜索操作，以找到问题的最优解或近似最优解。该方法虽然保证不一定能够得到问题的最优解，但若适当地利用一些启发知识，就可在近似解的质量和求解效率上达到一种较好的平衡。

随着问题种类的不同，以及问题规模的扩大，要寻求到一种能以有限的代价来解决上述最优化问题的通用方法仍是一个难题。而遗传算法却为我们解决这类问题提供了一个有效的途径和通用框架，开创了一种新的全局优化搜索算法。

遗传算法中，将  $n$  维决策向量  $X = [x_1, x_2, \dots, x_n]^T$  用  $n$  个记号  $X_i$  ( $i = 1, 2, \dots, n$ ) 所组成的符号串  $X$  来表示：

$$X = X_1 X_2 \cdots X_n \Rightarrow X = [x_1, x_2, \dots, x_n]^T$$

把每一个  $X_i$  看作一个遗传基因，它的所有可能取值称为等位基因，这样， $X$  就可看做是由  $n$  个遗传基因所组成的一个染色体。

一般情况下, 染色体的长度  $n$  是固定的, 但对一些问题  $n$  也可以是变化的。根据不同的情况, 这里的等位基因可以是一组整数, 也可以是某一范围内的实数值, 或者是纯粹的一个记号。最简单的等位基因是由 0 和 1 这两个整数组成的, 相应的染色体就可表示为一个二进制符号串。这种编码所形成的排列形式  $X$  是个体的基因型, 与它对应的  $X$  值是个体的表现型。通常个体的表现型和其基因型是一一对应的, 但有时也允许基因型和表现型是多对一的关系。染色体  $X$  也称为个体  $X$ , 对于每一个个体  $X$ , 要按照一定的规则确定出其适应度。个体的适应度与其对应的个体表现型  $X$  的目标函数值相关联,  $X$  越接近于目标函数的最优点, 其适应度越大; 反之, 其适应度越小。

遗传算法中, 决策变量  $X$  组成了问题的解空间。对问题最优解的搜索是通过对染色体  $X$  的搜索过程来进行的, 从而由所有的染色体  $X$  就组成了问题的搜索空间。

生物的进化是以群体为主体的。与此相对应, 遗传算法的运算对象是由  $M$  个个体所组成的集合, 称为群体。与生物一代一代的自然进化过程相类似, 遗传算法的运算过程也是一个反复迭代过程, 第  $t$  代群体记做  $P(t)$ , 经过一代遗传和进化后, 得到第  $t+1$  代群体, 它们也是由多个个体组成的集合, 记做  $P(t+1)$ 。这个群体不断地经过遗传和进化操作, 并且每次都按照优胜劣汰的规则将适应度较高的个体更多地遗传到下一代, 这样最终在群体中将会得到一个优良的个体  $X$ , 它所对应的表现型  $X$  将达到或接近于问题的最优解  $X^*$ 。

生物的进化过程主要是通过染色体之间的交叉和染色体的变异来完成的。与此相对应, 遗传算法中最优解的搜索过程也模仿生物的这个进化过程, 使用所谓的遗传算子 (genetic operators) 作用于群体  $P(t)$  中, 进行下述遗传操作, 从而得到新一代群体  $P(t+1)$ 。

●选择 (selection): 根据各个个体的适应度, 按照一定的规则或方法, 从第  $t$  代群体  $P(t)$  中选择出一些优良的个体遗传

到下一代群体  $P(t+1)$  中。

●交叉 (crossover): 将群体  $P(t)$  内的各个个体随机搭配成对, 对每一对个体, 以某个概率 (称为交叉概率, crossover rate) 交换它们之间的部分染色体。

●变异 (mutation): 对群体  $P(t)$  中的每一个个体, 以某一概率 (称为变异概率, mutation rate) 改变某一个或某一些基因座上的基因值为其等的等位基因。

### 1.2.2 遗传算法的运算过程

图 1-2 所示为遗传算法的运算过程示意图。

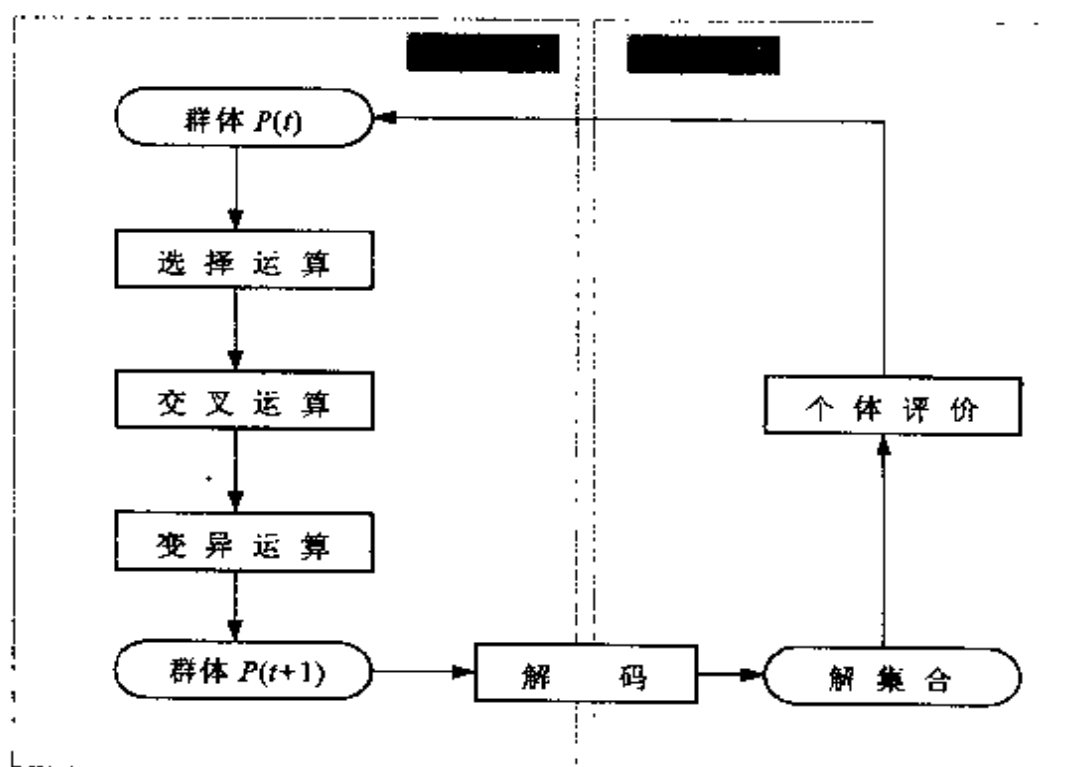


图 1-2 遗传算法的运算过程示意

由该图可以看出, 使用上述三种遗传算子 (选择算子、交叉算子、变异算子) 的遗传算法的主要运算过程如下所述。

**步骤一: 初始化。**设置进化代数计数器  $t \leftarrow 0$ ; 设置最大进化代数  $T$ ; 随机生成  $M$  个个体作为初始群体  $P(0)$ 。

**步骤二: 个体评价。**计算群体  $P(t)$  中各个个体的适应度。

**步骤三: 选择运算。**将选择算子作用于群体。

**步骤四：交叉运算。**将交叉算子作用于群体。

**步骤五：变异运算。**将变异算子作用于群体。群体  $P(t)$  经过选择、交叉、变异运算之后得到下一代群体  $P(t+1)$ 。

**步骤六：终止条件判断。**若  $t \leq T$ ，则： $t \leftarrow t+1$ ，转到步骤二；若  $t > T$ ，则以进化过程中所得到的具有最大适应度的个体作为最优解输出，终止计算。

### 1.2.3 遗传算法的手工模拟计算示例

为更好地理解遗传算法的运算过程，下面用手工计算来简单地模拟遗传算法的各个主要执行步骤，如表 1-1 所示。

**【例】**求下述二元函数的最大值：

$$\begin{cases} \max & f(x_1, x_2) = x_1^2 + x_2^2 \\ \text{s. t.} & x_1 \in \{0, 1, 2, \dots, 7\} \\ & x_2 \in \{0, 1, 2, \dots, 7\} \end{cases}$$

现对其主要运算过程作如下解释：

(1) 个体编码。遗传算法的运算对象是表示个体的符号串，所以必须把变量  $x_1$ 、 $x_2$  编码为一种符号串。该例题中， $x_1$  和  $x_2$  取 0~7 之间的整数，可分别用 3 位无符号二进制整数来表示，将它们连接在一起所组成的 6 位无符号二进制整数就形成了个体的基因型，表示一个可行解。例如，基因型  $X = 101110$  所对应的表现型是： $X = [5, 6]^T$ 。个体的表现型  $x$  和基因型  $X$  之间可通过编码和解码程序相互转换。

(2) 初始群体的产生。遗传算法是对群体进行的进化操作，需要给其准备一些表示起始搜索点的初始群体数据。本例中，群体规模的大小取为 4，即群体由 4 个个体组成，每个个体可通过随机方法产生。一个随机产生的初始群体如表 1-1 中第②栏所示。

(3) 适应度计算。遗传算法中以个体适应度的大小来评定各个个体的优劣程度，从而决定其遗传机会的大小。本例中，目标函数总取非负值，并且是以求函数最大值为优化目标，故可直接



利用目标函数值作为个体的适应度。为计算函数的目标值，需先对个体基因型  $X$  进行解码。表 1-1 中第③、④栏所示为初始群体中各个个体的解码结果，第⑤栏所示为各个个体所对应的目标函数值，它也是个体的适应度，第⑤栏中还给出了群体中适应度的最大值和平均值。

表 1-1 遗传算法的手工模拟计算

① 个体编号 $i$	② 初始群体 $P(0)$	③ $x_1$	④ $x_2$	⑤ $f_i(x_1, x_2)$	⑥ $f_i / \sum f_i$	
1	011101	3	5	34	<div><math>\sum f_i = 143</math> <math>f_{\max} = 50</math> <math>f = 35.75</math></div>	0.24
2	101011	5	3	34		0.24
3	011100	3	4	25		0.17
4	111001	7	1	50		0.35
⑦ 选择次数	⑧ 选择结果	⑨ 配对情况	⑩ 交叉点位置	⑪ 交叉结果	⑫ 变异点	⑬ 变异结果
1	011101	1-2 3-4	1-2; 2 3-4; 4	011001	4	011101
1	111001			111101	5	111111
0	101011			101001	2	111001
2	111001			111011	6	111010
⑭ 子代群体 $P(1)$	⑮ $x_1$	⑯ $x_2$	⑰ $f_i(x_1, x_2)$	⑱ $f_i / \sum f_i$		
011101	3	5	34	<div><math>\sum f_i = 235</math> <math>f_{\max} = 98</math> <math>f = 58.75</math></div>	0.14	
111111	7	7	98		0.42	
111001	7	1	50		0.21	
111010	7	2	53		0.23	

(4) 选择运算。选择运算（或称为复制运算）把当前群体中适应度较高的个体按某种规则或模型遗传到下一代群体中。一般要求适应度较高的个体将有更多的机会遗传到下一代群体中。本例中，我们采用与适应度成正比的概率来确定各个个体复制到下一代群体中的数量。其具体操作过程是：先计算出群体中所有

个体的适应度的总和  $\sum f_i$ ；其次计算出每个个体的相对适应度的大小  $f_i / \sum f_i$ ，如表 1-1 中第⑥栏所示，它即为每个个体被遗传到下一代群体中的概率，每个概率值组成一个区域，全部概率值之和为 1；最后再产生一个 0 到 1 之间的随机数，依据该随机数出现在上述哪一个概率区域内来确定各个个体被选中的次数。如表 1-1 中第⑦、⑧栏所示为一随机产生的选择结果。

(5) 交叉运算。交叉运算是遗传算法中产生新个体的主要操作过程，它以某一概率相互交换某两个个体之间的部分染色体。本例采用单点交叉的方法，其具体操作过程是：先对群体进行随机配对，如表 1-1 中第⑨栏所示为一种随机配对情况；其次随机设置交叉点位置，如表 1-1 中第⑩栏所示为一随机产生的交叉点位置，其中的数字表示交叉点设置在该基因座之后；最后再相互交换配对染色体之间的部分基因。表 1-1 中第⑪栏所示为交叉运算的结果。

例如，若第 3 号和第 4 号个体在第 4 个基因座之后进行交叉运算，则可得到两个新的个体：

第 3 号个体：1 0 1 0	∥	1 1	交叉操作	→	1 0 1 0	∥	0 1
第 4 号个体：1 1 1 0	∥	0 1			1 1 1 0	∥	1 1

可以看出，其中新产生的个体“111011”的适应度较原来两个个体的适应度都要高。

(6) 变异运算。变异运算是对于个体的某一个或某一些基因座上的基因值按某一较小的概率进行改变，它也是产生新个体的一种操作方法。本例中，我们采用基本位变异的方法来进行变异运算，其具体操作过程是：首先确定出各个个体的基因变异位置，如表 1-1 中第⑫栏所示为随机产生的变异点位置，其中的数字表示变异点设置在该基因座处；然后依照某一概率将变异点的原有基因值取反。表 1-1 第⑬栏所示为变异运算结果。

例如，若第 3 号个体的第 2 个基因座需要进行变异运算，则可产生出一个新的个体：

第3号个体: 1 0 1 0 0 1  $\xrightarrow{\text{第2位变异}}$  1 1 1 0 0 1

对群体  $P(t)$  进行一轮选择、交叉、变异运算之后可得到新一代的群体  $P(t+1)$ 。如表 1-1 第⑭栏所示。表中第⑬、⑯、⑰、⑱栏还分别表示出了新群体的解码值、适应度和相对适应度, 并给出了适应度的最大值和平均值等。从表 1-1 中可以看出, 群体经过一代进化之后, 其适应度的最大值、平均值都得到了明显的改进。事实上, 这里已经找到了最佳个体“111111”。

需要说明的是, 表中第②、⑦、⑨、⑩、⑫栏的数据是随机产生的。这里为了更好地说明问题, 我们特意选择了一些较好的数值以便能够得到较好的结果, 而在实际运算过程中有可能需要一定的循环次数才能达到这个最优结果。

### 1.3 遗传算法的特点

为解决各种优化计算问题, 人们提出了各种各样的优化算法, 如单纯形法、梯度法、动态规划法、分枝定界法等。这些优化算法各有各的长处, 各有各的适用范围, 也各有各的限制。遗传算法是一类可用于复杂系统优化计算的鲁棒搜索算法, 与其他一些优化算法相比, 它主要有下述几个特点<sup>[5]</sup>:

(1) 遗传算法以决策变量的编码作为运算对象。传统的优化算法往往直接利用决策变量的实际值本身来进行优化计算, 但遗传算法不是直接以决策变量的值, 而是以决策变量的某种形式的编码为运算对象。这种对决策变量的编码处理方式, 使得我们在优化计算过程中可以借鉴生物学中染色体和基因等概念, 可以模仿自然界中生物的遗传和进化等机理, 也使得我们可以方便地应用遗传操作算子。特别是对一些无数值概念或很难有数值概念, 而只有代码概念的优化问题, 编码处理方式更显示出了其独特的优越性。

(2) 遗传算法直接以目标函数值作为搜索信息。传统的优化

算法不仅需要利用目标函数值，而且往往需要目标函数的导数值等其他一些辅助信息才能确定搜索方向。而遗传算法仅使用由目标函数值变换来的适应度函数值，就可确定进一步的搜索方向和搜索范围，无需目标函数的导数值等其他一些辅助信息。这个特性对很多目标函数是无法或很难求导数的函数，或导数不存在的函数的优化问题，以及组合优化问题等，应用遗传算法时就显得比较方便，因为它避开了函数求导这个障碍。再者，直接利用目标函数值或个体适应度，也可使得我们可以把搜索范围集中到适应度较高的部分搜索空间中，从而提高了搜索效率。

(3) 遗传算法同时使用多个搜索点的搜索信息。传统的优化算法往往是从解空间中的一个初始点开始最优解的迭代搜索过程。单个搜索点所提供的搜索信息毕竟不多，所以搜索效率不高，有时甚至使搜索过程陷于局部最优解而停滞不前。遗传算法从由很多个体所组成的一个初始群体开始最优解的搜索过程，而不是从一个单一的个体开始搜索。对这个群体所进行的选择、交叉、变异等运算，产生出的乃是新一代的群体，在这之中包括了很多群体信息。这些信息可以避免搜索一些不必搜索的点，所以实际上相当于搜索了更多的点，这是遗传算法所特有的一种隐含并行性。

(4) 遗传算法使用概率搜索技术。很多传统的优化算法往往使用的是确定性的搜索方法，一个搜索点到另一个搜索点的转移有确定的转移方法和转移关系，这种确定性往往也有可能使得搜索永远达不到最优点，因而也限制了算法的应用范围。而遗传算法属于一种自适应概率搜索技术，其选择、交叉、变异等运算都是以一种概率的方式来进行的，从而增加了其搜索过程的灵活性。虽然这种概率特性也会使群体中产生一些适应度不高的个体，但随着进化过程的进行，新的群体中总会更多地产生出许多优良的个体，实践和理论都已证明了在一定条件下遗传算法总是以概率 1 收敛于问题的最优解。当然，交叉概率和变异概率等参数也会影响算法的搜索效果和搜索效率，所以如何选择遗传算法

的参数在其应用中是一个比较重要的问题。而另一方面,与其他一些算法相比,遗传算法的鲁棒性又会使得参数对其搜索效果的影响会尽可能地低。

## 1.4 遗传算法的发展

遗传算法起源于对生物系统所进行的计算机模拟研究。早在本世纪 40 年代,就有学者开始研究如何利用计算机进行生物模拟的技术,他们从生物学的角度进行了生物的进化过程模拟、遗传过程模拟等研究工作。进入 60 年代后,美国密执安大学的 Holland 教授及其学生们受到这种生物模拟技术的启发,创造出了一种基于生物遗传和进化机制的适合于复杂系统优化计算的自适应概率优化技术——遗传算法。下面是在遗传算法的发展进程中一些关键人物所做出的一些主要贡献。

### 1. J. H. Holland

60 年代, Holland 认识到了生物的遗传和自然进化现象与人工自适应系统的相似关系,运用生物遗传和进化的思想来研究自然和人工自适应系统的生成以及它们与环境的关系,提出在研究和设计人工自适应系统时,可以借鉴生物遗传的机制,以群体的方法进行自适应搜索,并且充分认识到了交叉、变异等运算策略在自适应系统中的重要性。

70 年代初, Holland 教授提出了遗传算法的基本定理——模式定理 (Schema Theorem),从而奠定了遗传算法的理论基础。模式定理揭示出了群体中的优良个体(较好的模式)的样本数将以指数级规律增长,因而从理论上保证了遗传算法是一个可以用来寻求最优可行解的优化过程。1975 年, Holland 出版了第一本系统论述遗传算法和人工自适应系统的专著《自然系统和人工系统的自适应性 (Adaptation in Natural and Artificial Systems)》<sup>[3]</sup>。

80 年代, Holland 教授实现了第一个基于遗传算法的机器学习系统——分类器系统 (Classifier Systems, 简称 CS),开创了

基于遗传算法的机器学习的新概念, 为分类器系统构造出了一个完整的框架<sup>[6]</sup>。

## 2. J. D. Bagley

1967 年, Holland 的学生 Bagley 在其博士论文中首次提出了“遗传算法”一词<sup>[7]</sup>, 并发表了遗传算法应用方面的第一篇论文。他发展了复制、交叉、变异、显性、倒位等遗传算子, 在个体编码上使用了双倍体的编码方法。这些都与目前遗传算法中所使用的算子和方法相类似。他还敏锐地意识到了在遗传算法执行的不同阶段可以使用不同的选择率, 这将有利于防止遗传算法的早熟现象, 从而创立了自适应遗传算法的概念。

## 3. K. A. De Jong

1975 年, De Jong 在其博士论文中结合模式定理进行了大量的纯数值函数优化计算实验, 树立了遗传算法的工作框架, 得到了一些重要且具有指导意义的结论<sup>[4]</sup>。例如, 对于规模在 50~100 的群体, 经过 10~20 代的进化, 遗传算法都能以很高的概率找到最优或近似最优解。他推荐了在大多数优化问题中都较适用的遗传算法的参数, 还建立了著名的 De Jong 五函数测试平台, 定义了评价遗传算法性能的在线指标和离线指标。

## 4. D. J. Goldberg

1989 年, Goldberg 出版了专著《搜索、优化和机器学习中的遗传算法 (Genetic Algorithms in Search, Optimization and Machine Learning)》<sup>[5]</sup>。该书系统总结了遗传算法的主要研究成果, 全面而完整地论述了遗传算法的基本原理及其应用。可以说这本书奠定了现代遗传算法的科学基础, 为众多研究和发展遗传算法的学者所瞩目。

## 5. L. Davis

1991 年, Davis 编辑出版了《遗传算法手册 (Handbook of Genetic Algorithms)》一书, 书中包括了遗传算法在科学计算、工程技术和社会经济中的大量应用实例<sup>[8]</sup>。这本书为推广和普及遗传算法的应用起到了重要的指导作用。

6.J.R.Koza

1992年, Koza将遗传算法应用于计算机程序的优化设计及自动生成, 提出了遗传编程 (Genetic Programming, 简称 GP) 的概念<sup>[9,10]</sup>。他将一段 LISP 语言程序作为个体的基因型, 把问题的解编码为一棵树, 基于遗传和进化的概念, 对由树组成的群体进行遗传运算, 最终自动生成性能较好的计算机程序。Koza成功地把他提出的遗传编程的方法应用于人工智能、机器学习、符号处理等方面。

## 1.5 遗传算法的应用

遗传算法提供了一种求解复杂系统优化问题的通用框架, 它不依赖于问题的具体领域, 对问题的种类有很强的鲁棒性, 所以广泛应用于很多学科<sup>[8,11,12,13]</sup>。下面是遗传算法的一些主要应用领域:

(1) 函数优化。函数优化是遗传算法的经典应用领域, 也是对遗传算法进行性能评价的常用算例。很多人构造出了各种各样的复杂形式的测试函数, 有连续函数也有离散函数, 有凸函数也有凹函数, 有低维函数也有高维函数, 有确定函数也有随机函数, 有单峰值函数也有多峰值函数等。用这些几何特性各具特色的函数来评价遗传算法的性能, 更能反映算法的本质效果。而对于一些非线性、多模型、多目标的函数优化问题, 用其他优化方法较难求解, 而遗传算法却可以方便地得到较好的结果。

(2) 组合优化。随着问题规模的增大, 组合优化问题的搜索空间也急剧扩大, 有时在目前的计算机上用枚举法很难或甚至不可能求出其精确最优解。对这类复杂问题, 人们已意识到应把主要精力放在寻求其满意解上, 而遗传算法是寻求这种满意解的最佳工具之一。实践证明, 遗传算法对于组合优化中的 NP 完全问题非常有效。例如, 遗传算法已经在求解旅行商问题、背包问题、装箱问题、图形划分问题等方面得到成功的应用。

(3) 生产调度问题。生产调度问题在很多情况下所建立起来的数学模型难以精确求解,即使经过一些简化之后可以进行求解,也会因简化得太多而使得求解结果与实际相差甚远。而目前在现实生产中也主要是靠一些经验来进行调度。现在遗传算法已成为解决复杂调度问题的有效工具,在单件生产车间调度、流水线生产车间调度、生产规划、任务分配等方面遗传算法都得到了有效的应用。

(4) 自动控制。在自动控制领域中有很多与优化相关的问题需要求解,遗传算法已在其中得到了初步的应用,并显示出了良好的效果。例如用遗传算法进行航空控制系统的优化、使用遗传算法设计空间交会控制器、基于遗传算法的模糊控制器的优化设计、基于遗传算法的参数辨识、基于遗传算法的模糊控制规则的学习、利用遗传算法进行人工神经网络的结构优化设计和权值学习等,都显示出了遗传算法在这些领域中应用的可能性。

(5) 机器人学。机器人是一类复杂的难以精确建模的人工系统,而遗传算法的起源就来自于对人工自适应系统的研究,所以机器人学理所当然地成为遗传算法的一个重要应用领域。例如,遗传算法已经在移动机器人路径规划、关节机器人运动轨迹规划、机器人逆运动学求解、细胞机器人的结构优化和行为协调等方面得到研究和应用。

(6) 图像处理。图像处理是计算机视觉中的一个重要研究领域。在图像处理过程中,如扫描、特征提取、图像分割等不可避免地会存在一些误差,这些误差会影响图像处理的效果。如何使这些误差最小是使计算机视觉达到实用化的重要要求。遗传算法在这些图像处理中的优化计算方面找到了用武之地,目前已在模式识别、图像恢复、图像边缘特征提取等方面得到了应用。

(7) 人工生命。人工生命是用计算机、机械等人工媒体模拟或构造出的具有自然生物系统特有行为的人造系统。自组织能力和自学习能力是人工生命的两大主要特征。人工生命与遗传算法有着密切的关系,基于遗传算法的进化模型是研究人工生命现象



的重要基础理论。虽然人工生命的研究尚处于启蒙阶段，但遗传算法已在其进化模型、学习模型、行为模型、自组织模型等方面显示出了初步的应用能力，并且必将得到更为深入的应用和发展。人工生命与遗传算法相辅相成，遗传算法为人工生命的研究提供了一个有效的工具，人工生命的研究也必将促进遗传算法的进一步发展。

(8) 遗传编程。Koza 发展了遗传编程的概念，他使用了以 LISP 语言所表示的编码方法，基于对一种树型结构所进行的遗传操作来自动生成计算机程序。虽然遗传编程的理论尚未成熟，应用也有一些限制，但它已成功地应用于人工智能、机器学习等领域。

(9) 机器学习。学习能力是高级自适应系统所应具备的能力之一。基于遗传算法的机器学习，特别是分类器系统，在很多领域中都得到了应用。例如，遗传算法被用于学习模糊控制规则，利用遗传算法来学习隶属度函数，从而更好地改进了模糊系统的性能；基于遗传算法的机器学习可用来调整人工神经网络的连接权，也可用于人工神经网络的网络结构优化设计；分类器系统也在学习式多机器人路径规划系统中得到了成功的应用。