南京航空航天大学

硕士研究生学位论文

姓　　名：石　　玉

专　　业：测试计量技术及仪器

研究方向：测试信息分析与处理

导　　师：于 盛 林　　教授

一九九九 年 一 月

工学硕士学位论文

遗传算法及其应用

研究方向：测试信息分析与处理

学生姓名：石　　玉　　入学时间：1996.9

指导教师：于 盛 林　教授　　答辩时间：1999.3

南京航空航天大学

一九九九年三月

摘　要

本文系统地介绍了遗传算法的基本理论和方法，重点讨论了实数遗传算法，并将之用于数字滤波。

首先，本文介绍了遗传算法的基本概念、特点、研究发展情况，对遗传算法的基本理论和方法进行了系统的说明。基本理论和方法中包括模式定理、积木块假设、算法收敛性、算法的隐并行性及算法的基本操作（编码、适应度函数设计和定标、遗传操作、结束条件等等）。

然后，本文详细讨论了实数编码的遗传算法，采用 Turbo C 语言对遗传操作编程运算，比较分析其结果，并针对遗传操作和参数选择提出了一些改进的措施。

最后，本文将实数遗传算法用于数字滤波，实验结果表明用此方法对随机噪声和脉冲型噪声都可以产生较好的效果。

在全文的结束，对课题进行了总结，并对课题今后的研究提出了一些设想。

关键词：遗传算法　遗传操作　数字滤波

Abstract

In this paper, the basic theories and methods of genetic algorithms have been systematically introduced. Real-encoding genetic algorithms have been mainly discussed and used in digital filtering.

Firstly, the basic conceptions, the features, the studies and developments of genetic algorithms are introduced. Then, the basic theories and methods that include schemata theorem, building block hypothesis, implicit parallelism, and the basic operations of genetic algorithms such as encoding, designing and scaling of adaptive functions, genetic operations, termination conditions and so on are explained systematically.

Next, real-encoding genetic algorithms are discussed in detail. Genetic operations are simulated, compared and analyzed by using Turbo C, some improvements on genetic operations and the selection of parameters are put forward.

Finally, real-encoding genetic algorithms are used in digital filtering. The results indicate that the filtering is good for both random noise and impulse noise.

At the end of the paper, research of the article have been concluded, and some ideas for future study have been put forward.

Key Words: genetic algorithm genetic operation digital filtering

目　录

第一章　绪论

§1.1　遗传算法的描述

遗传算法（Genetic Algorithms）是模拟自然界的物种进化和自然选择机制形成的一种易于操作的并行搜索方法。它首先对参数进行编码，生成一定数量（规模）的可能解，形成初始解群。其中每个解可以是一维或多维矢量，以二进制数串或实数串表示，称染色体，染色体的每一位二进制数或每一个实数值称基因。自然界生物被客观环境选择，优胜劣汰，算法中则需要设计适应度函数作为评判每个可能解性能优劣的标准，性能好的解以一定概率被选择出来作为父代参加以后的遗传操作以生成新一代数群。遗传操作主要包括染色体选择，染色体上基因的交叉和基因的变异，交叉的结果使父代的特性更集中，变异则有益于产生新个体，利于进化。生成新一代数群后，循环进行适应度评价、遗传操作等步骤，逐代优化，直至满足结束条件。

标准遗传算法的流程如图 1-1 所示。

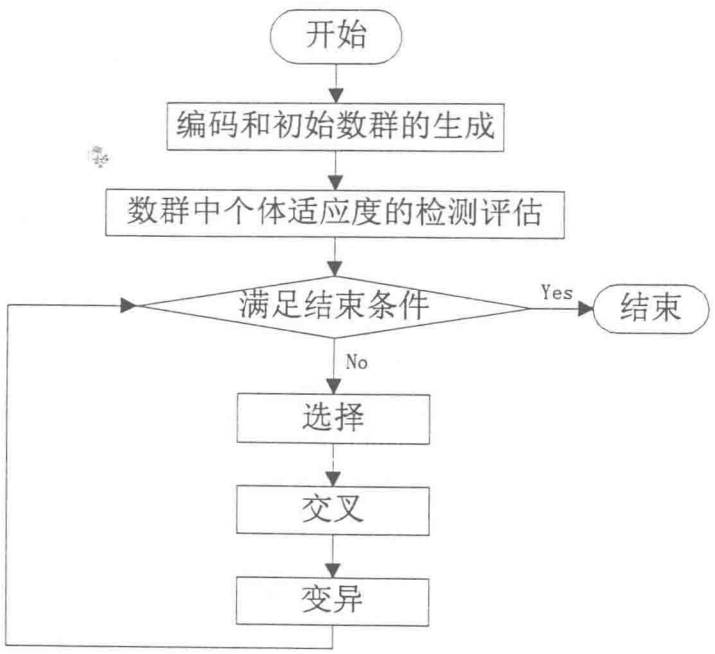


图 1-1　遗传算法基本流程

§1.2　遗传算法的特点及优越性

与传统的优化算法相比，遗传算法有如下特点：

1．遗传算法的处理对象是参数的某种编码而未必是参数本身，这使得遗传算法可以对结构对象，如集合、序列、矩阵、树、图、链、表等进行操作，从而使算法具有广泛的应用领域。

2．遗传算法不是从单个解，而是从解的群体开始搜索。

3．遗传算法利用适应度函数评估个体，无需导数或其它辅助信息。

4．遗传算法不是利用确定性规则，而是利用概率的变迁规则来引导其搜索方向。

如前所述，一方面，遗传算法的变异操作可以产生独特的个体，使搜索跳到较远的区域，保持数群的多样性；另一方面，由于是数群参与计算，即在解空间的不同区域进行多点搜索，因而能以很大的概率找到全局最优解。因此，不易陷入局部最优解是遗传算法的一个优越性。

遗传算法的另一个优越性是其隐含并行性。当算法处理一个规模为 N 的数群时，实际能反映对 cN^3 个相同结构个体的优化（c 为常数），这将在下一章遗传算法的原理中详细说明。

§1.3　遗传算法的发展简史

对遗传算法的研究始于本世纪 60 年代。当时，一些生物学家利用计算机模拟了自然遗传系统。受此启发，美国 Michigan 大学的 J. Holland 教授发现了基于适应度的人工遗传选择的基本作用，并对群体操作进行研究，于 1965 年首次提出了人工遗传操作的重要性，并将之应用到自然和人工系统中。

1967 年，Bagley 提出遗传算法这一术语，发展了与目前类似的选择、交叉、变异等遗传操作，并对染色体选择进行了详细的研究，提出了适应度定标（scaling）的概念和算法自我调整的思想，以防止早熟收敛。

同时，Rosenberg 对单细胞生物群体进行计算机仿真，Cavicchio 将遗传算法用于模式识别，Weinberg 在进行生物体计算机仿真时提出了用多层遗传算法进行参数优化，Hollstien 则首次将遗传算法用于函数优化。

1975 年，Holland 出版了专著《Adaptive in Nature and Artificial System》，书中系统地阐述了遗传算法的基本理论和方法，提出模式定理，为遗传算法奠定了数学基础。同年，De Jong 在其论文“An Analysis of the Behavior of a Class of Genetic Adaptive System”中作了大量的计算实验，把模式理论和函数优化结合起来。他建立了 De Jong 五函数测试平台，其中包括非凸函数、非连续函数、随机函数和高维函数，并且定义了遗传算法性能评价的标准。

1983 年，Goldberg 第一次把遗传算法用于实际的工程系统——煤气管道的优化。从此，遗传算法进入了蓬勃发展的时期，算法的应用研究更加广泛，理论研究也更加深入丰富。自 1985 年起，国际遗传算法会议 ICGA(International

Conference on Genetic Algorithm)每两年召开一次，有关人工智能的会议和刊物上大多有遗传算法的专题，Goldberg 的课本及其他学者的专著的出版更有力地推动了遗传算法的传播。

从 1990 年开始，以遗传算法的理论基础为中心的学术会议 FOGA(Foundation of Genetic Algorithm)隔年召开一次，类似的会议，如欧洲的 PSSN(Parallel Problem Solving form Nature)也隔年举行一次。遗传算法能有效地求解属于 NPC 类型的组合优化问题及非线性多模型、多目标的函数优化问题，因而得到多学科的广泛重视，成为寻求满意解的最佳工具之一，其应用范围也扩展到了控制、规划、设计、组合优化、图象处理、信号处理、机器人、人工生命等多个领域。

§1.4　遗传算法的理论与应用研究

遗传算法的理论研究主要有：

1．分析遗传算法的编码策略、全局收敛性和搜索效率的数学基础。

2．遗传算法的结构研究。例如双群体遗传算法，改变搜索范围的变焦遗传算法，以及调整遗传算子次序的多种算法。

3．遗传算法的遗传操作策略及其性能研究。主要是针对适应度函数的设计和三个主要遗传操作进行的改进。例如结合祖先特点的适应值求法，自我调整的适应度函数及基因操作，改进搜索性能的大变异操作等等。

4．遗传算法的参数选择。例如自适应群体规模的设定及自适应交叉、变异参数的设定。

5．遗传算法与其它算法的综合及比较研究。遗传算法可以和多种搜索方法结合，例如和最陡下降法(Steepest Descent Method)的结合，和禁忌搜索(Tabu Search)的结合，和胞腔排除法(Cell Exclusive Method)的结合，和进化算法(Evolution Algorithms)的结合、和模拟退火(Simulated Annealing)的结合等等。

遗传算法的应用研究主要包括三个部分：

1．基于遗传的优化计算。这是遗传算法最直接的应用，除了函数优化，组合优化外，神经网络已成为遗传算法应用最为活跃的领域。算法成为优化神经网络的结构权重和学习规则的有力工具。

2．基于遗传的优化编程。与优化计算不同，遗传编程(Genetic Programming)所得到的是计算机程序，它要求算法有强有力的编码表达。

3．基于遗传的机器学习。遗传学习(Genetic Learning)被应用于机器人学、模式识别、工程等方面。它把遗传算法扩展成具有独特规则生成功能的机器学习算法。

§1.5　遗传算法今后研究的主要课题

今后遗传算法研究的主要课题有：

1．优化搜索方法的研究。除了进一步改进基本理论和方法外，还要采用与

神经网络、模拟退火、专家系统等其它方法相结合的策略。

2．学习系统的遗传算法研究。包括适合遗传算法的高层次知识的表示，更通用的基于语义的操作，搜索能力的分析，更有效的体现和变动环境相互作用的适应度函数的导入等。

3．生物进化与遗传算法的研究。从生物进化的角度看，目前遗传算法理论框架尚处于核糖核酸(RNA)的水平，随着生物的超精密结构和机制正被逐渐探明，算法在吸收这些知识的基础上，从形式到内涵将受到检验并可能发生质的飞跃。

4．遗传算法的并行分布处理。遗传算法本质上有很好的并行分布处理特性，但多数的遗传操作是建立在全局统计的基础上的，在整个进化过程中所引起的通信开销较大，因而设计有效的并行遗传算法及相应的实现系统很重要。

5．人工生命与遗传算法的研究。已有一些学者用遗传算法对生态系统的演变、食物链的维持、免疫系统的进化等进行了生动的模拟。遗传算法在实现人工生命中的地位与能力是值得研究的课题。

§1.6　本研究课题的主要工作

本课题所做的主要工作有：

1．在阅读资料的基础上总结遗传算法的基本理论和方法。

2．对实数遗传算法的遗传操作和参数选择进行了详细的讨论。

3．将实数遗传算法用于静态信号的滤波。

4．将实数遗传算法用于动态信号的滤波。

第二章　遗传算法的基本理论及方法

§2.1　模式定理及隐并行性

2.1.1 模式定理

模式定理（schemata theorem）是基于二进制编码提出的，是遗传算法的理论基础，描述如下：具有低阶、短定义距以及适应值高于群体平均适应值的模式在遗传算法迭代过程中将按指数级增长。

首先介绍定理中相关的概念：

模式（schema）：基于三值字符集{0，1，*}所产生的能描述具有某些结构相似性的 0，1 字符串集的字符串，用 H 表示。符号“*”称作通配符，既可以代表“0”，也可以代表“1”。以长度为 5 的串为例， H=*1**0 描述了所有在位置 2 为“1”，位置 5 为“0”的字符串，即集合{01000，01010，01100，01110，11000，11010，11100，11110}。

模式阶（schema order）：模式 H 中确定位置的个数，记作 O(H)。上例中模式的阶为 2。

定义距（defining length）：模式 H 中第一个确定位置和最后一个确定位置之间的距离，记作$\delta(H)$。上例中模式的定义距为 3。

按模式定理，若设适应度函数为$f(x)=x^2$，x编码为二进制数串，取长度为 5 的模式 H_1=*1**0，H_2=00*01，H_2的阶为 4，定义距为 4，模式 H_1的适应度值显然大于模式 H_2的适应值，则在算法迭代过程中 H_1比 H_2更有可能按指数级增长。

设数群的规模为n，且数群中的个体两两互不相同；第t代中与模式 H 匹配的串的个数记作$m(H,t)$；第i个串被选择的概率

$$P_i = f_i / \sum_{j=1}^{n} f_j \qquad (2.1\text{-}1)$$

其中f_i是第i个个体的适应度值；与模式 H 匹配的所有个体的平均适应度值称为模式平均适应度，记作$f(H)$；则 H 在第$t+1$代中的匹配个体数$m(H,t+1)$为：

$$m(H,t+1) = m(H,t) \cdot n \cdot f(H) / \sum_{j=1}^{n} f_j \qquad (2.1\text{-}2)$$

令群体的平均适应值记作

$$\bar{f} = \sum_{j=1}^{n} f_j / n \qquad (2.1\text{-}3)$$

则模式的选择复制方程又可写为

$$m(H,t+1) = m(H,t) \frac{f(H)}{\bar{f}} \qquad (2.1\text{-}4)$$

可见，一个特定的模式按照其平均适应度与群体的平均适应值之间的比率生长，模式平均适应值高于群体平均适应值的模式在下一代中会有更多的代表串，模式适应值低于群体平均适应值的模式在下一代中的代表串将会减少。

假设模式 H 的平均适应值高于群体平均适应值，高出部分为$c\bar{f}$，c 为常数，则

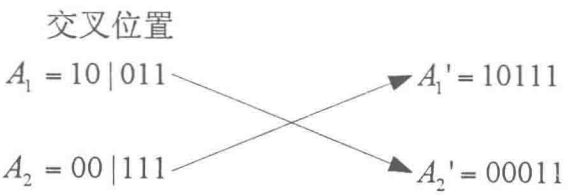
$$m(H,t+1) = m(H,t) \frac{\bar{f} + c\bar{f}}{\bar{f}} = (1+c) \cdot m(H,t) \qquad (2.1\text{-}5)$$

若从$t=0$开始，c 保持为常值，有

$$m(H,t) = m(H,0) \cdot (1+c)^t \qquad (2.1\text{-}6)$$

因此，在群体平均适应值以上（以下）的模式将会按指数增长（衰减）的方式被选择复制。

再看交叉算子对模式增长产生的影响。交叉能产生新个体，使算法到新的区域进行搜索，只讨论单点交叉的情况。它首先从群体中随机选择一对参与交叉的串，然后随机选择一个交叉位置，使交叉的两个体在这一位置前或后的内容进行交换。如个体 A_1=10011 与 A_2=00111 在第 2 点（以“|”表示）交叉的结果为：



下面以长度为 7 的串 A 和包含在其中的两个模式 H_1、H_2 为例进行说明。设随机取定的交叉位置在 3：

A=011|1000

H_1=*1*|***0　　H_2=***|10**

设 H 是包含在 A 中的模式，如果与 A 进行交叉的个体在模式 H 的确定位和 A 相同，交叉后将不破坏该模式。例如对模式 H_1 而言，确定位是第 2 位“1”

和第 7 位“0”，设与 A 交叉的个体为 B=1100000，由于 B 在第 2、7 位与 A 一致，无论交叉位置在第几点，数值交换后产生的新个体在第 2、7 位的值仍然不变，模式 H_1 未遭破坏。同样，若与 A 交叉的个体在第 4、5 位分别是“1”和“0”，交叉对 H_2 将不产生影响。不考虑上述的特殊情况，一般情况下，H_1 比较容易遭到破坏。以交叉位置在 3 为例，交叉后，H_1 在位置 2 上的“1”和在位置 7 上的“0”位于不同的新串上，而 H_2 在位置 4 和 5 上的值将不受影响地保留到新串中。H_1 的定义距大，被破坏的可能性也大，H_2 的定义距小，被破坏的可能性也小，因而 H_1 比 H_2 不易生存。也就是说，因为交叉点一般更易落在距离最远的确定位置之间，所以低定义距的模式生存率高。

设个体串长度为 l，交叉位置将在 $l-1$ 个可能位置上随机选择。如果交叉位置落在定义距之内，模式极易被破坏，被破坏的概率是 $\delta(H)/(l-1)$，若发生交叉操作的概率（交叉概率）是 p_c，则模式被破坏的概率是 $p_c \cdot \dfrac{\delta(H)}{l-1}$，而生存概率 p_s 为

$$p_s \geq 1 - p_c \cdot \frac{\delta(H)}{l-1} \tag{2.1-7}$$

H_1、H_2 由交叉算子决定的生存概率分别为 1/6 和 5/6。

将选择复制和交叉操作结合起来，当选择和交叉不相关时，模式 H 在新一代期望出现的个数是

$$m(H,t+1) \geq m(H,t) \cdot \frac{f(H)}{\bar{f}} \left[1 - p_c \cdot \frac{\delta(H)}{l-1}\right] \tag{2.1-8}$$

即模式平均适应度大于群体平均适应度又具有短定义距的模式在新一代群体中将按指数增长。

最后来看变异操作对模式保留的作用。设变异概率为 p_m，即变异以 p_m 的概率随机改变个体一个位上的值。为使模式存活，所有确定位必须保留。在变异算子作用下，个体一个基因被保留的概率为 $1-p_m$，因而对阶为 O(H)的模式而言，其生存概率为 $(1-p_m)^{O(H)}$。一般 p_m 取值很小（p_m<<1），模式存活率近似为 $1-O(H)\cdot p_m$。

综上所述，在选择、交叉、变异算子作用下，模式 H 在新一代中期望出现的次数可近似表示为：

$$m(H,t+1) \geq m(H,t) \frac{f(H)}{\bar{f}} \left[1 - p_c \frac{\delta(H)}{l-1} - O(H)p_m\right] \tag{2.1-9}$$

低阶、短定义距、高平均适应度的模式在新一代中将以指数级增长。

统计确定理论中的双角子机问题表明，要获得最优的可行解，必须保证较优解的样本数呈指数级增长。因此，模式定理满足了寻找最优解的必要条件，即遗传算法存在搜索到最优解的可能性。

2.1.2 积木块假设

积木块是具有低阶、短定义距以及高平均适应度的模式。积木块假设指出了遗传算法具备搜索到最优解的能力。它没有在理论上得到完全的证明，但被大量的事实所验证。表述为：

积木块假设（building block hypothesis）：低阶、短定义距、高平均适应度的模式（积木块）在遗传算子作用下，相互结合，能生成高阶、长定义距、高平均适应度的模式，最终可生成全局最优解。

2.1.3 隐并行性

隐并行性（implicit parallelism）指出，当遗传算法对 n 个个体进行运算时，实际隐含处理了$O(n^3)$个模式。$O(n^3)$表示正比与 n 的立方。

在串长为l，规模为n的二进制串群体中，包含2^l到$n\cdot 2^l$个模式，在单点交叉且变异率很小时，设按指数增长的模式数为n_s。

只考虑生存概率大于p_s的模式，即考虑定义距$l_s < (1-P_s)(l-1)+1$的模式。

假设计算定义距小于l_s=5 的隐含在长度$l=10$的串 A=1011100010 中的模式。先计算前 5 个位置（1011100010）的模式数。由于第 5 位固定，即计算模式%%%%1*****的个数，“%”既可表示确定位，又可表示通配符“*”，模式数为2^{l_s-1}。将底线右移一位，可计算下 5 位的模式数：1011100010。如此，对长为l的串，需将底线移动$l-l_s+1$次，从而定义距不大于l_s的模式数为$2^{l_s-1}\cdot(l-l_s+1)$。

设群体规模为n，由于群体中存在模式重复的现象，不可以用n直接乘上式表示的模式数。假设阶不小于$l_s/2$的模式不重复，则取规模$n=2^{l_s/2}$。模式数按二项分布，只计算阶不大于$l_s/2$的模式，模式数的下界n_s为：

$$n_s \geq n\cdot(l-l_s+1)\cdot 2^{l_s-2} \qquad (2.1\text{-}10)$$

代入$n=2^{l_s/2}$，有：

$$n_s = (l - l_s + 1) \cdot n^3 / 4 \qquad (2.1\text{-}11)$$

即算法处理的模式总数正比于群体规模的立方。

§2.2 编码方法

在遗传算法的特点中提到，算法的处理对象是参数的某种编码。编码是遗传算法的第一步，其方法的选择直接影响到算法的功能，对交叉操作甚至起到决定性的作用。

2.2.1 编码的评估规范

编码策略通常采用 3 个评估规范：

1．完备性　问题空间的所有点都能作为算法空间的染色体表示。

2．健全性　算法空间的染色体能对应所有问题空间的点。

3．非冗余性　染色体和问题空间点一一对应。

当然，针对具体问题，有时为了减少计算量，提高搜索效率，未必严格满足以上规范。

2.2.2 编码规则

De Jong 提出了较为客观、明确的编码原理，包括两条规则：

1．有意义积木块编码规则：所定编码应当易于生成与所求问题相关的短距低阶的积木块。

2．最小字符集编码规则：所定编码应采用最小字符集以使问题得到自然的表示或描述。

规则 1 是基于模式定理和积木块假设提出的；二进制编码则符合规则 2 的思想。

2.2.3 编码方法

传统的编码方法是用二进制编码，它在理论上易于处理，位串也容易产生和操作，几乎任何问题都可以用这种编码方法表示。

设被编码参数 a 的变化范围是 $[a_{\min}, a_{\max}]$，用 m 位二进制数 b 表示，则

$$a = a_{\min} + \frac{b}{2^m - 1}(a_{\max} - a_{\min}) \qquad (2.2\text{-}1)$$

字长 m 应在满足精度要求下尽量取小值，以减少算法的复杂性。例如 $a \in [-1,2]$，

精度要求小数点后 6 位，则定义域被分成3·1000000 个小区域，二进制字长 m=22 位($1097152 = 2^{21} < 3000000 \le 2^{22} = 4194304$)。转化关系分两步：

$$\begin{cases} (\langle b_{21}b_{20}\ldots b_0\rangle)_2=(\sum_{i=0}^{21} b_i \cdot 2^i)_{10}=a' \\ a=-1.0+a'\dfrac{3}{2^{22}-1} \end{cases} \tag{2.2-2}$$

为了满足较高的精度，二进制编码的字长将增加很多，既不方便又影响计算的速度，采用实数进行编码已越来越受到重视。实数编码避免了进制之间的转换，表示的精度高，算法速度快，并且可以表示很大的数域范围。对单参数优化，可以直接将一个实参数值作为一个染色体；对多参数优化，则将多参数的组合作为染色体，某一参数值作为其上的一个基因。

此外，还有指数编码，即将数值用科学记数法表示从而对底数和指数分别编码，以及用计算机程序进行编码等等。

从其它角度看，编码方法还包括：按数值的维数分，有一维编码、多维编码；按数据结构分，有数值编码、树结构编码；按处理问题的对象分，有符号编码、序号编码；另外还有定长染色体编码、变长染色体编码等等。

§2.3　遗传操作

标准遗传算法的遗传操作（genetic operation）包括三个基本算子：染色体的选择（selection）、交叉（crossover）、变异（mutation）。

基本遗传算子有如下特点：

1．操作在随机扰动下进行，同时这种操作是高效有向的。

2．三个算子内部的参数选择对算法的效果有很大的影响。

3．三个算子的操作方法和个体的编码方式有关。

选择的目的是选出优秀的个体以直接遗传到下一代或作为性能好的亲代进行交叉遗传。以常见的赌轮选择为例，也就是按个体适应度在群体中的比例进行选择复制。设群体规模为 n，其中个体 i 的适应度值为 f_i，则第 i 个个体被选择的概率 p_{si} 为

$$p_{si} = f_i / \sum_{j=1}^{n} f_i \tag{2.3-1}$$

交叉依据编码类型的不同有多种方法，单点交叉在模式定理的分析中已有介绍，个体 $a_1 = 1011100$，$a_2 = 1100011$ 在第三点交叉后产生后代 $a_1' = 1010011$ 和 $a_2' = 1101100$。

变异中的简单变异如下，设 $a=1100111$ 在第五点发生变异得到 a'，$a'=1100011$。

多种遗传操作方法由以下的三个表分别表示，下一章将具体分析实数遗传算法中的各种操作。

表 2-1　染色体选择操作

序号	名　称	研　究　者	备　注
1	赌轮选择	De Jong　Brindle	标准 GA 的成员
2	期望值模型	De Jong　Brindle	
3	确定式采样	Brindle	
4	有回放式余数随机采样（赌轮方式）		
5	无回放式余数随机采样(贝努利试验方式)	Brindle　Booker	应用较广
6	均匀排序	Back	
7	稳态复制	Syswerda	
8	随机比赛	Brindle	
9	复制评价	Whitley	
10	随机竞争		
11	最优串复制	De Jong　Back Greffenstette	
12	排挤选择	De Jong	

表 2-2　基因交叉操作

序号	名　称	研　究　者	适用编码
1	单点交叉	Holland　De Jong Goldberg	符号
2	双点交叉	Cavicchio　Booker Syswerda　Whitely　Yun	符号
3	均匀交叉	Syswerda	符号
4	多点交叉	De Jong　Spears	符号
5	一致交叉		符号
6	树结构交叉	Louis　Rawlin 山村　织田　小林	
7	部分匹配交换（PMX）	Goldberg	序号
8	序号交换（OX）	Davis	序号
9	循环交换（CX）	Smith	序号
10	启发式交换	Greffenstette	序号
11	基于位置的交换	Syswerda	序号
12	基于次序的交换		序号
13	简单实数串交叉算子	Michalewicz	实数
14	离散交叉算子	Michalewicz	