

实数编码遗传算法的评述

田小梅, 龚 静

(湖南生物职业技术学院, 湖南 衡阳 421005)

摘 要: 遗传算法是一种基于自然选择和遗传变异等生物进化机制的全局优化搜索算法. 使用实数表示基因的实数编码遗传算法常用于求解连续函数优化等问题. 本文阐述了实数编码遗传算法研究的有关工作进展, 讨论了算法框架及特点, 对实数编码遗传算法中常用的选择、交叉和变异算子进行了比较全面的形式化描述, 并介绍了其未来研究方向.

关键词: 遗传算法; 实数编码遗传算法; 选择; 交叉; 变异

中图分类号: TP18

文献标识码: A

文章编号: 1671 - 6361(2005)01 - 025 - 07

On Overview of Real - Coded Genetic Algorithm

TIAN Xiao - mei, Gong Jing

(Hunan Environment - Biological Polytechnic, Hengyang 421005 Hunan)

Abstract: Genetic algorithm is a global optimization search method which is based on biological evolutionary mechanism such as natural selection, heredity and mutation. Real - Coded Genetic Algorithm which denotes gene with real number is often used to solve continuous function optimization problem. This paper expatiates on the progresses in real - coded genetic algorithm research field, discusses algorithm skeleton and characteristic, formally depicts selection, crossover and mutation operators which are often used in real - coded genetic algorithm, and introduces some future research work.

Key words: genetic algorithm; real - coded genetic algorithm; selection; crossover; mutation

遗传算法作为一种通用性好、鲁棒性强的启发式随机搜索优化算法, 广泛地应用于自动控制、组合优化、图像处理、机器人、人工生命、机器学习、人工智能和工程设计等领域中. 尤其是当搜索空间很大、问题非常复杂或对问题域的先验知识很少时, 使用经典搜索工具如枚举法、启发式方法等不适宜的情况下, 遗传算法提供了一种效率高且有效的求解问题的合理方法. 实数编码遗传算法的发展时间并不长, 但对于参数是连续型变量的数值优化问题, 直接采用实数表示基因是特别自然的. 它与传统遗传算法(二进制编码遗传算法)不同的是: 直接采用实数表示基因, 等位基因就是实数的取值, 染色体则是一个实数向量, 染色体的长度即此实数向量的大小. 目前, 实数编码遗传算法已引起了越来越多的专家学者的关注, 而且取得了大量的理论研究成果和应用研究成果.

收稿日期: 2004 - 11 - 08

基金项目: 湖南环境生物职业技术学院院长科研基金资助(203 - 01)

作者简介: 田小梅(1971 -), 女, 湖南隆回人, 讲师. 研究方向: 遗传算法

1 遗传算法编码

按照遗传算法的工作流程,当用遗传算法求解问题时,必须在目标问题实际表示与遗传算法的染色体结构之间建立联系,即确定编码.由于遗传算法计算过程的鲁棒性,它对编码的要求并不苛刻.实际上,大多数问题都可以采用基因呈一维排列的定长染色体表现形式,尤其是基于 $\{0,1\}$ 符号集的二进制编码形式.由于编码形式决定了交叉算子的操作方式,编码问题往往称作编码-交叉问题.因此,作为遗传算法流程中第一步的编码是遗传算法中需要认真研究的问题,很多专家提出了各种编码方法^[8].

1.1 二进制编码(binary encoding)

此编码将问题空间的参数表示为基于 $\{0,1\}$ 字符集构成的染色体位串.染色体 X 可形式描述为: $X = (x_1, x_2, x_3, \dots, x_n)$, $x_i = (x_{i1}, x_{i2}, \dots, x_{im})$, $x_{ij} \in \{0,1\}$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$.这种方式最常采用,采用二进制编码的遗传算法称为二进制编码遗传算法(Binary-Coded Genetic Algorithm,简称BCGA).它具有以下的特点:①简单易行;②符合最小字符集编码规则;③便于用模式定理进行分析,因为模式定理就是以二进制编码为基础提出的.

采用这一编码形式的遗传算法进行数值优化时,可以通过改变编码长度,协调搜索精度和搜索效率之间的矛盾.

目前,二进制编码还有一些其他的变体形式,如格雷(Gray)编码、二倍体(diploid)编码等.

1.2 实数编码(real-number encoding)

实数编码直接使用问题变量进行编码.染色体 X 的形式为: $X = (x_1, x_2, x_3, \dots, x_n)$, $x_i \in R$, $i = 1, 2, \dots, n$.使用这种编码的遗传算法称为实数编码遗传算法(Real-Coded Genetic Algorithm,简称为RCGA).实数编码遗传算法具有精度高,便于大空间搜索等优点.本文仅限于概括介绍实数编码遗传算法的发展及已有的研究成果.

1.3 大字符集编码(large alphabet encoding)

除了上述两种编码形式之外,可以结合实际问题的特征采用 D 进制数($D > 2$)或字符集来表示长度为 L 的染色体基因串,如染色体 $X = (x_1, x_2, x_3, \dots, x_n)$, $x_i \in \{0, 1, 2, \dots, D-1\}$.

1.4 序列编码(sequence encoding)

采用GA求解TSP问题时,用排列法进行编码似乎更为自然合理.如有10个城市的TSP问题,城市序号为 $\{1, 2, \dots, 10\}$,则编码位串: $(1, 3, 4, 2, 6, 7, 10, 5, 9, 8)$ 表示按照如下次序“1→3→4→2→6→7→10→5→9→8”依次访问各个城市.

1.5 树编码(tree-based encoding)

树编码是一种非定长编码方式,其表示空间是开放的.在搜索过程中树可以自由地增长,但是不形成更具有结构化和层次性的问题解,实际应用中往往加以限制.

1.6 自适应编码(adaptive encoding)

事先选择合适的固定编码方式对潜在的遗传算法用户来说是一个难题.事实上,提出最合适的编码同解决问题本身一样困难.因此一些专家就采用了自适应编码的方法,让遗传算法在进化过程中同时调整编码.树编码也可看作是一种自适应编码,因为染色体长度随环境变化而动态地变化.

2 实数编码遗传算法的发展和研究概况

遗传算法(genetic algorithm, GA)是一类借鉴自然界生物种类的遗传和进化过程而形成的一种自适应全局优化搜索算法,它的主要特点是群体搜索策略和群体中个体之间的信息交换,搜索不依赖于梯度信息.它尤其适用于处理传统搜索方法难以解决的复杂和非线性问题,可广泛用于组合优化、机器学习、自适应控制、规划设计和人工生命等领域,是21世纪智能计算中的关键技术之一.遗传算法研究的兴起是在80年代末和90年代初期,但它的历史起源可追溯到60年代.早在1967年Bagley就提出了“遗传算法”一词并发表了第一篇有关遗传算法应用的论文“采用遗传算法的六子棋评估函数中的参数搜索”.从60年代至70年代中期,这一阶段对遗传算法的研究主要侧重于对一些复杂操作的研究,如自动博弈、生物系统模拟、模式识别和函数优化等,这是一个无明确目标的发展时期,缺乏带有指导性的理论和计算工具的开拓.这种现象直到70年代中期由于Holland和De Jong的创造性研究成果的发表才得以改

观.遗传算法的基本理论和基本方法最初由美国 Michigan 大学的 J. Holland 教授于 1975 年首先提出,同年 De Jong 针对他提出的 5 个测试函数对简单遗传算法的基本参数及其部分改进 GA 的搜索性能进行了细致的分析研究.从此,遗传算法不断得到改进和提高,并已在实际应用领域中取得了巨大的成功,目前它已成为进化计算(evolutionary computation)的一个研究热点.

随着采用二进制编码的标准遗传算法在各个领域广泛应用的同时,研究人员发现,在处理高维连续搜索空间问题和对问题的解要求较大数值精度的时候,二进制编码遗传算法遇到了难以解决的困难.例如对于一个定义域为 $[-500, 500]$ 的含 100 个变量的函数优化问题,若要求小数点后六位小数的数值精度,二进制位串的长度将达 3000,这样,产生的搜索空间的大小约为 10^{1000} .另一个问题就是当一个变量仅含有有限个(非 2^n 个)离散有效值时,部分二进制代码是冗余的,它们经解码后对应的值不在参数的定义域内,为了处理经交叉和变异操作后产生的这些冗余代码而不影响遗传算法的效率,必须采用一些专门机制,如固定重映射(fixed remapping)、随机重映射(random remapping)或概率重映射(probabilistic remapping)技术,将冗余代码映射为有效代码,这样,必将增加算法实现的复杂性,而且可能影响父代个体(下简称父代)与子代个体之间的信息传递量.自 1989 年 Lucasius 和 Kateman^[1]首次将实数编码用于化学统计学中探寻拟合核磁共振数据的发夹型 DNA 的结构以来,很多学者(如 Davis, Wright, Goldberg, Radcliffe, Michalewica, Mühlenbein, Deb 等)为克服上述二进制编码遗传算法存在的缺陷,在研究中纷纷使用实数编码方法以更有效地使用遗传算法处理现实优化问题.

3 实数编码遗传算法框架描述

实数编码遗传算法的搜索过程可描述如下:

Procedure RCGA

Begin

初始化:

{

选择遗传操作的类型,确定所有的参数值;

设置进化代数计数器 $t = 0$;

随机生成初始种群 $p(0) = \{X_1, X_2, X_3, \dots, X_N\}$, N 为种群大小

}

度量:计算初始种群 $P(0)$ 中每个个体的适应度值;

while (终止条件不满足) do

{

交叉:对 $P(t)$ 进行交叉操作生成种群 $p'(t+1)$

变异:对 $p'(t+1)$ 进行变异操作生成种群 $p''(t+1)$

度量:计算种群 $p''(t+1)$ 中每个个体的适应度值;

选择:对 $(Q \cup p''(t+1))$ 进行选择操作生成新种群 $P(t+1)$, 其中 Q 代表 $P(t)$ 的某个子集或空集.

$t = t + 1$;

}

End

其中选择、交叉、变异是遗传算法的基本遗传操作.

4 实数编码遗传算法的特点

实数编码遗传算法的使用使得遗传算法的研究推进了一大步,近十多年来受到了广大专家和学者的关注,相关的理论与应用研究层出不穷.实数编码遗传算法除了具备二进制编码遗传算法的所有特点,如简单、通用、鲁棒性强、适于并行分布处理等之外,实数编码遗传算法还有如下一些特点:

(1)直接使用实数作为染色体参与遗传操作,无需特定的编码与解码过程,因此降低了算法实现的复杂度,提高了算法的执行效率,尤其是当处理大规模复杂问题、高维数值优化问题或子目标个数较多的多目标优化问题时,算法的效率更能得到体现.

(2)用实数编码可以消除二进制编码存在的“海明悬崖”(Hamming cliffs)问题。

(3)实数编码遗传算法中使用的交叉和变异算子适应于不同的进化方法,如进化规划及进化策略。同样,进化规划及进化策略中的相关算子也可用于实数编码遗传算法。

(4)算法中可以利用连续变量函数的渐变性(graduality),这里的“渐变性”指的是变量值的微小变化所引起的对应函数值的变化也是微小的。由于这个特点,使得遗传算法具有局部调节功能。例如 RCGA 的非一致变异(non uniform mutation)算子就具有比 BCGA 中变异算子更快更合理地对解群进行局部调节的功能,而在 BCGA 中由于海明悬崖效应的存在这种调节是很困难的。

(5)使用实参数作为编码,使得变量可以取大定义域甚至是未知定义域成为可能;而这在二进制编码表示染色体的算法实现中是很难做到的,也就是说假定染色体长度保持不变,增大定义域就意味着牺牲数值精度。

(6)对于具非平凡约束条件的问题,实数编码使得问题域知识较易并入 RCGA 中,指导种群朝正确的搜索方向进化。

(7)根据 Antonisse 的理论,语言越富于表达力,它提供的自适应机制越强,使用实数编码染色体的基因型和表现型相似,其所能达到的表达能力级别是很高的。

(8)使用实数编码的染色体,其繁殖新个体的方式更灵活。对于二进制编码遗传算法,由于编码的限制,可供使用的交叉和变异算子的种类非常有限;而使用实数编码时能用的交叉和变异算子种类则相对较丰富,同时也使得算法具有更好的搜索能力。

5 实数编码遗传算法的基本操作

实数编码遗传算法包括三个基本操作:选择、交叉和变异。这些基本操作又有许多不同的方法^[3,4,6,8],下面逐一进行介绍。

5.1 常用选择算子

遗传算法的原理从本质上来说基于达尔文的自然选择学说。选择提供了遗传算法种群进化的驱动力。在过去的 20 多年中提出了许多选择方法。实数编码遗传算法中常用的选择方法有如下几种:

5.1.1 轮盘赌选择(roulette wheel selection) 这是一种最基本的、最常用的选择机制。在这种选择机制中,个体每次被选中的概率与其在整个种群中的相对适应度成正比,故又称适应度比例选择法。设群体大小为 n ,其中个体 ind 的适应度值为 F_{ind} ,则 ind 被选择的概率为 P_{ind} 为: $P_{ind} = F_{ind} / \sum_{j=1}^n F_j$ 由上式可看出,个体适应度值越大,其被选择的概率就越高。

5.1.2 $(\mu + \lambda)$ 选择($(\mu + \lambda)$ selection) 这种选择策略最初在演化策略中使用,现遗传算法中亦经常使用,其选择标准是根据父代种群的 μ 个个体用交叉(重组)或变异操作产生 λ 个子代个体,然后将这 $\mu + \lambda$ 个个体进行比较再在其中选取 μ 个最优者。

5.1.3 锦标赛选择(tournament selection) 其操作思想是,从种群中任意选择一定数量(称为竞赛规模,一般取 2)的个体,其中适应度最高的个体保存到下一代。这个过程反复执行,直到保存到下一代的个体数达到预先设定的数目为止。

5.1.4 排序选择(rank-based model) 此选择方法在计算每个个体的适应度后,根据适应度大小顺序对种群中全部个体排序,然后按事先设计好的概率分配表取对应序号的概率值作为各个体的选择概率。

5.1.5 期望值方法(expected value model) 在这种机制中,先按期望值 $n \cdot F_{ind} / \sum_{j=1}^n F_j$ 的整数部分安排个体被选中的次数,而对其期望值 $n \cdot F_{ind} / \sum_{j=1}^n F_j$ 的小数部分再按确定方式、赌轮方式或贝努利试验方式确定该个体再次被选中与否及其选中次数^[3]。

5.1.6 共享(sharing) 此机制主要适用于多峰函数的优化及多目标优化领域,用于维持种群的多样性。共享函数用于确定每个个体在种群中的共享度,一个个体的共享度等于该个体与种群内的各个其它个体之间的共享函数值的总和。

5.1.7 精英保留选择机制(elitist model) 这方法的思想是把种群中适应度最高的个体不进行配对交叉而直接复制到下一代中。此种选择操作又称复制。此方法一般与其他选择方法结合使用。

5.1.8 截断选择(truncation selection) 这种方法只从种群中选择一定比例的最好个体作为父体,然后均匀随机地对这些父体进行交叉配对直至子代个数等于种群大小。

5.2 常用交叉算子

前面已经提到,实数编码遗传算法与二进制编码遗传算法相比,可供使用的交叉和变异算子种类丰富^[5],算法具有更好更灵活的繁殖新个体的能力。

假设参与交叉的两个父体分别表示为: $p_1 = (p_1^1, p_1^2, \dots, p_1^n)$, $p_2 = (p_2^1, p_2^2, \dots, p_2^n)$ 。

5.2.1 平坦交叉(flat crossover) 由两个父体 P_1, P_2 产生一个子代个体: $C_1 = (c_1, c_2, \dots, c_n)$, 其中 c_i 是随机均匀产生的在区间 $[p_i^1, p_i^2]$ 内的一个数值。

5.2.2 简单交叉(simple crossover) 首先随机产生一个交叉位置 $i, i \in \{1, 2, \dots, n-1\}$, 然后按下述方式产生两个新的个体 C_1 和 C_2 :

$$C_1 = (p_1^1, p_2^1, \dots, p_1^i, p_2^i, \dots, p_1^n)$$

$$C_2 = (p_2^1, p_1^1, \dots, p_2^i, p_1^i, \dots, p_2^n)$$

5.2.3 算术交叉(arithmetical crossover)^[2] 通过此方式产生的个体 C_1 和 C_2 如下:

$$C_k = (c_1^k, \dots, c_i^k, \dots, c_n^k), k = 1, 2$$

其中 $c_i^1 = \lambda p_1^1 + (1-\lambda)p_2^1, c_i^2 = \lambda p_2^1 + (1-\lambda)p_1^1$. 其中 λ 可以是常量, 此时即为均匀算术杂交, 也可以随着世代数而变化, 此时称非均匀算术杂交。

5.2.4 线性交叉(linear crossover) 首先产生三个新的个体, 然后从中选择两个最好的个体作为子代个体。

$$C_k = (c_1^k, \dots, c_i^k, \dots, c_n^k), k = 1, 2, 3,$$

$$c_i^1 = \frac{1}{2} p_1^1 + \frac{1}{2} p_2^1, c_i^2 = \frac{3}{2} p_1^1 - \frac{1}{2} p_2^1,$$

$$c_i^3 = -\frac{1}{2} p_1^1 + \frac{3}{2} p_2^1.$$

5.2.5 混合交叉(BLX- α crossover) 此算子每次产生一个子代个体: $C_1 = (c_1, c_2, \dots, c_n)$, 其中 c_i 是在区间 $[p_{\min} - I \cdot \alpha, p_{\max} + I \cdot \alpha]$ 中随机均匀选取的数值, $p_{\max} = \max(p_1^1, p_2^1), p_{\min} = \min(p_1^1, p_2^1), I = p_{\max} - p_{\min}$ 。

5.2.6 离散交叉(discrete crossover) 离散交叉又分为部分离散交叉和整体离散交叉。部分离散交叉即是在组成父体的向量分量中选择一部分分量(如一个分量或从某分量以后的所有分量, 或者是随机选择的一部分分量), 然后交换这些分量以形成后代。如下述形式的部分离散交叉与简单交叉相同。

$$C_1 = (p_1^1, p_2^1, \dots, p_1^i, p_2^i, \dots, p_1^n)$$

$$C_2 = (p_2^1, p_1^1, \dots, p_2^i, p_1^i, \dots, p_2^n)$$

整体离散交叉是以 1/2 的概率交换 p_1, p_2 的所有分量, 这有点像二进制编码时的均匀性杂交。 $C_k = (c_1^k, \dots, c_i^k, \dots, c_n^k), k = 1, 2$, 其中, c_i^k 是从集合 $\{c_i^1, c_i^2\}$ 中随机均匀选取的一个数值。

5.2.7 启发式交叉(Wright's heuristic crossover) 假设 P_1 是具有最好适应度值的父体, 则生成的新个体 $C = (c_1, \dots, c_2, \dots, c_n)$, 其中 $c_i = r \cdot (p_1^1 - p_1^2) + p_1^1, r$ 是一个位于 $[0, 1]$ 区间中的一个随机数。

5.2.8 模拟二进制交叉(simulated binary crossover, SBX)^[7] 先通过下式计算父体相应分量的 β_{qi} 值: $\beta_{qi} =$

$$\left\{ \begin{array}{ll} (2u_i)^{\frac{1}{\eta_e+1}}, & u_i \leq 0.5 \\ \frac{1}{2(1-u_i)^{\frac{1}{\eta_e+1}}}, & \text{其他} \end{array} \right\}, \text{其中 } \eta_e \text{ 是由用户确定的分布指数, 可取任意非负实数. } \eta_e \text{ 较大时, 产生的后代个}$$

体离父体较近的概率非常高, 反之, η_e 较小时, 产生的后代个体离父体较远的概率较大。

子代个体由下式得出:

$$c_i^1 = 0.5[(1+\beta_{qi})p_1^1 + (1-\beta_{qi})p_2^1], c_i^2 = 0.5[(1+\beta_{qi})p_2^1 + (1-\beta_{qi})p_1^1].$$

5.2.9 单峰正态分布交叉(unimodal normally distributed recombination, UNDX) 设参与交叉的父体有 u 个, 为 $\vec{x}^i (i=1, 2, \dots, u)$. 首先从中随机选取 $(u-1)$ 个父体, 设为 $\vec{x}^i (i=1, 2, \dots, u-1)$, 计算它们的均值向量 \vec{g} . 由 \vec{g} 计算 $(u-1)$ 个方向向量 $\vec{d}^{(i)} = \vec{x}^i - \vec{g}$; 求出 $\vec{e}^{(i)} = \vec{d}^{(i)} / |\vec{d}^{(i)}|$; 然后计算与所有 $\vec{e}^i (i=1, 2, \dots, u-1)$

1) 正交的向量 $(\hat{x}^{(u)} - \hat{g})$ 的长度 D ; 进而计算 $\hat{e}^{(j)} (j = u, \dots, n)$, $\hat{e}^{(j)}$ 是与被所有 $\hat{e}^{(i)}$ 覆盖的子空间正交的子空间的正交基, n 是向量 \hat{x} 的大小. 最后, 子代由下式得出: $\hat{y} = \hat{g} + \sum_{i=1}^u w_i |d^{(i)}| \hat{e}^{(i)} + \sum_{i=1}^u v_i D \hat{e}^{(i)}$, 其中, w_i, v_i 分别是服从均值为 0, 方差为 $\sigma_x^2 \sigma_y^2$ 的正态分布变量.

5.2.10 单纯形交叉 (simplex recombination, SPX) 为了提高进化算法的局部搜索能力, Renders 和 Bersini 把传统优化中的单纯形算法用于构造交叉算子. 其方法为: 设有 $u (u > 2)$ 个父体 $\hat{x}^i (i = 1, 2, \dots, u)$ 参与杂交, 选出最好的个体 (设为 $\hat{x}^{(b)}$, 用 b 表示) 和最差的个体 (设为 $\hat{x}^{(w)}$, 用 w 表示), 然后计算出除 $\hat{x}^{(w)}$ 以外的父体的质心 $c: c = \sum_{i=1}^{u-1} \frac{\hat{x}^{(i)}}{u-1}$, 由质心算出最差个体的反射点 $r = c + (c - w)$, 经过比较函数值 $f(b), f(w), f(r)$, 最后可确定杂交产生的后代为其中适应度较好的个体.

5.2.11 父体中心交叉 (parent-centric recombination, PCX) 设参与交叉的父体有 u 个, 为 $\hat{x}^i (i = 1, 2, \dots, u)$. 首先计算它们的均值向量 \hat{g} ; 每产生一个后代时, 以等概率从 u 个父体中选取一个父体, 设为 $\hat{x}^{(p)}$, 计算方向向量 $d^{(p)} = \hat{x}^{(p)} - \hat{g}$; 然后计算出其他 $(u-1)$ 个父体与 $d^{(p)}$ 的垂直距离 D_i 和平均值 \bar{D} ; 最后, 子代由下式得出:

$$\hat{y} = \hat{x}^{(p)} + w_\zeta |d^{(p)}| + \sum_{i=1, i \neq p}^u w_\eta \bar{D} \hat{e}^{(i)}$$

其中, w_ζ, w_η 分别是服从均值为 0, 方差为 σ_x^2, σ_y^2 的正态分布变量. $\hat{e}^{(i)}$ 是覆盖垂直于 $d^{(p)}$ 的子空间的 $(u-1)$ 个正交基.

另外, 还有很多其他的杂交算子, 如几何杂交 (geometrical crossover); 王宇平、刘海林和 Y. W. Leung 把一种叫均匀设计的数学实验方法用于设计杂交算子, 提出了均匀杂交 (uniform crossover) 算子; Herrera 等提出的基于模糊连结的交叉算子 (Fuzzy Connectives Based Crossover, FCB).

5.3 常用变异算子

假设拟变异的个体表示为: $p = (p_1, \dots, p_i, \dots, p_n)$, 拟变异的基因表示为 $c_i, c_i \in [a_i, b_i]$; 变异后的基因表示为 c'_i .

5.3.1 随机变异 (random mutation) 从区间 $[a_i, b_i]$ 中随机均匀选取.

5.3.2 非一致变异 (non-uniform mutation)^[2] 为了提高变异算子对局部范围的搜索能力, Michalewicz 把参与变异的分量作随机扰动, 这个扰动在进化初期变化范围较大, 随着进化代数的增加扰动变化范围逐渐减小. 变异后的个体按下式计算:

$$c'_i = \begin{cases} c_i + \Delta(t, b_i - c_i) & \text{if } \gamma = 0 \\ c_i - \Delta(t, c_i - a_i) & \text{if } \gamma = 0 \end{cases}$$

其中 γ 是取值为 0/1 的随机数, t 是当前代次, T 是最大进化代数, $\Delta(t, y) = y(1 - r^{(1-\frac{1}{T})^b})$, r 是区间 $[0, 1]$ 中的一个随机数, b 是由用户选取的参数, 由它控制算法对代数的依赖度.

5.3.3 多项式变异 (polynomial mutation) 首先计算 δ 值, 如下式:

$$\delta = \begin{cases} (2u)^{\frac{1}{\eta_m+1}} - 1 & \text{if } u < 0.5 \\ 1 - [2(1-u)]^{\frac{1}{\eta_m+1}} & \text{if } u \geq 0.5 \end{cases}$$

u 是一个 $(0, 1)$ 区间内的随机数, η_m 是由用户选取的分布指数. 然后, 按下式计算变异后的个体基因值:

$$c'_i = c_i + \delta \Delta_{\max}, \Delta_{\max} \text{ 是最大允许扰动范围.}$$

5.3.4 Mühlenbein 变异 (Mühlenbein mutation) 按下式计算 $c'_i: c'_i = c_i \pm \text{range}_i \gamma$, range_i 定义变异范围, 通常设置为 $0.1 \cdot (b_i - a_i)$. “+”、“-”符号各按 0.5 的概率的选取, $\gamma = \sum_{k=0}^{15} \alpha_k 2^{-k}$, $\alpha_k \in \{0, 1\}$, 其取 1 值的概率为 $1/16$, 即 $p(\alpha_k = 1) = 1/16$.

5.3.5 边界变异 (boundary mutation) 变异后的个体基因值在集合 $\{a_i, b_i\}$ 中随机选取.

5.3.6 高斯变异 (Gaussian mutation) 高斯变异是比较常用的一种变异算子, 最初是在进化策略中使用, 现也广泛用于遗传算法. 这种变异的方式是, 对要进行变异的个体 $p = (p_1, \dots, p_i, \dots, p_n)$, 确定 n 个服从均值为 0、方差是 σ_i 的正态分布的随机变量 $N(0, \sigma_i) (i = 1, 2, \dots, n)$, 变异后的子代 $C = (c_1, c_2, \dots,$

c_n)的分量是:

$$c_i = p_i + N(0, \sigma_i), i = 1, 2, \dots, n$$

由正态分布的特性可知,高斯变异是对变异个体的附近区域进行重点搜索的一种变异算子。

5.3.7 柯西变异(Cauchy mutation) Yao X 和 Liu Y 提出的柯西变异是高斯变异算子的一种变形,用服从 Cauchy 分布的随机数代替服从正态分布的随机数,即:

$$c_i = p_i + \delta_i, i = 1, 2, \dots, n$$

其中, δ_i 是规模参数 $t=1$ 、服从 Cauchy 分布的随机变量。以原点为中心的一维 Cauchy 密度函数是:

$$f_t(x) = \frac{1}{\pi} \frac{t}{t^2 + x^2}, -\infty < x < \infty,$$

这里 $t > 0$, 称为规模参数。由 Cauchy 分布密度函数的性质,可推得 Cauchy 变异的搜索范围比 Gaussian 变异要大,因而这种变异更容易调节局部最优值。

5.3.8 自适应性变异(adaptive mutation)^[8] 潘正君等将模拟退火思想引入到非均匀变异算子中,提出了自适应性变异算子。其方法如下:设 $p = (p_1, \dots, p_i, \dots, p_n)$ 是解空间的一个个体, $f(P)$ 是它的适应度值,

F 为所求问题的最大适应度值的一个粗略上限或当前群体中的最大适应度值,则定义 $T = 1 - \frac{f(p)}{F}$ 为该个体的变异温度。个体 P 的变异方式类似非均匀变异算子,只是把其中的进化代数 t 改为 $\Delta(T, y) = y(1 - r^T)$ 。

该算子对适应度较好的解在其较小的范围内搜索,而对较差的解搜索领域较大。

6 研究展望

(1)为实数编码遗传算法中交叉算子和变异算子的选定建立标准。目前,已设计了很多种不同的交叉算子和变异算子,在解决某一具体问题,该选择何种交叉和变异算子会得出较好的结果还没有任何标准,仅凭经验与探索。

(2)现有的多目标遗传算法对于较高维多目标优化问题收敛性差或根本不收敛,如何将实数编码遗传算法应用于多目标优化问题的求解,将是一个非常前途的研究领域。

(3)很多学者都对使用多亲交叉算子的多亲遗传算法进行了研究,用实验证明了其较好的搜索性能。新的多亲交叉算子的设计将会是一个较好的研究方向。

(4)由于 RCGA 直接使用实数向量作为染色体,其繁殖新个体的方式非常灵活,新遗传算子的设计也较具吸引力。

参考文献:

- [1] Lucasius, C. B., G. Kateman. Applications of genetic algorithms in chemometrics[A]. Proceedings of the Third International Conference on Genetic Algorithms[C]. CA: Morga Kaufman Publishers, Inc., 1989, 170 ~ 176.
- [2] Michalewicz, Z. . Genetic Algorithms + Data Structures = Evolution Program[M]. Springer - Verlag, New York, 1992.
- [3] 陈国良, 王煦法, 庄镇泉, 王东生. 遗传算法及应用[M]. 北京: 人民邮电出版社, 1996, 117 ~ 121.
- [4] 王小平, 曹立明. 遗传算法 - 理论、应用与软件实现[M]. 西安: 西安交通大学出版社, 2002, 21 ~ 38.
- [5] Herrera, F., Lozano, M. and Verdegay, J. L. . Tackling real - coded genetic algorithms: Operators and tools for behavioural analysis [J]. Artificial Intelligence Review, 1998, 12(4), 265 ~ 319.
- [6] 玄光男, 程润伟. 遗传算法与工程优化[M]. 北京: 清华大学出版社, 2004.
- [7] Deb, K. and Agrawal, R. B. . Simulated binary crossover for continuous search space[J]. Complex Systems, 1995, 9(2), 115 ~ 148.
- [8] 潘正军, 康立山, 陈毓屏. 演化计算[M]. 北京: 清华大学出版社, 1998.