第 18 卷 第 3 期 控 制 与 决 策 2003 年 5 月
Vol. 18 No. 3 Control and Decision May 2003

文章编号: 1001-0920(2003)03-0361-03

基于异位交叉的遗传算法的研究

钟国坤, 曾 碧, 余永权
(广东工业大学 计算机学院, 广东 广州 510090)

摘 要: 针对目前遗传算法搜索速度较慢的问题, 对提高遗传算法收敛速度的不同方法进行了分析。提出一种加快收敛速度的异位交叉算子, 并给出算法仿真实验。仿真结果表明, 这种交叉算子可比一般的对等位交叉算子更有效地提高收敛速度, 且不易陷入局部最优解。具有实现简单、易于应用及鲁棒性强的特点。

关键词: 遗传算法; 交叉算子; 收敛速度
中图分类号: TP301.6 文献标识码: A

Study of genetic algorithms based on different location crossover

ZHONG Guo-kun, ZENG Bi, YU Yong-quan
(Faculty of Computer, Guangdong University of Technology, Guangzhou 510090, China)

Abstract: Aimed at the problems of slow convergence speed of genetic algorithm, a variety of methods are analysed to improve convergence speed of genetic algorithm. A method of different location crossover is proposed to quicken the convergence to the optimal set. Simulation results demonstrate that the different location crossover is more efficient to improve convergence speed than the standard crossover of genetic algorithm. The method is easy to realize and has better robustness.

Key words: Genetic algorithms; Crossover operator; Convergence of speed

1 引 言

遗传算法(GA)是一种基于生物进化论和遗传学机理的概率搜索方法。对于各种优化计算问题, 单纯形法、梯度法、动态规划法等都有各自的优点适应范围及限制。由于遗传算法中不包含待解决问题所特有的形态, 因此它被用于很多目标函数无法求取或很难求导数的函数优化问题以及组合优化等问题时, 就显得较为方便。虽然遗传算法具有以上优点, 但由于遗传算法存在早熟和收敛速度问题, 影响了它的应用, 因此如何提高遗传算法的搜索能力和收敛速度是一个重要的问题。虽然对此已提出不同的改进方法, 如自调整适应度、自调整交叉、变异概率、交叉位置非等概率选取和保存最优个体等。这些方法在不同程度上都可提高收敛速度, 但对于不同问题收敛速度的提高差别很大。

由于交叉算子在遗传算法中起着非常重要的作用, 它对于遗传算法的收敛速度有很重要影响。本文提出了一种提高收敛速度的方法, 引出一种新的交叉算子——异位交叉。为了研究这种交叉算子的收敛速度, 通过对多个函数的实验, 并与采用对等位交叉算子结合一些改进方法(自调整适应度, 自调整交叉、变异概率, 保存最优个体)的效果进行比较, 通过反复试验, 发现它比一般的交叉算子更能有效地推进进化, 加快收敛速度。

2 遗传算法优化的关键技术

遗传算法通过对生物遗传和进化过程中的选择、交叉、变异机理的模仿, 完成对问题最优解的自适应搜索过程。把待优化问题的参数编码成二进制

收稿日期: 2002-02-24; 修回日期: 2002-06-06。

作者简介: 钟国坤(1978—), 男, 广东东莞人, 硕士生, 从事模糊控制、神经网络的研究; 曾碧(1963—), 女, 广东广州人, 副教授, 从事模糊控制、神经网络等研究。

位串的形式, 然后由若干个位串形成一个初始种群作为待求问题的候选解, 通过适应度评价, 适应度高的优质个体被复制保留, 适应度低的个体被淘汰。通过交叉和变异, 产生新个体, 交叉是核心部分, 交叉后, 子代的基因链是父代的继承与重组, 任一子代中的适应度都比父代双亲高。但只用交叉, 可能使种群陷入局部最优解, 而变异则可能使之跳出, 由于变异概率过高可能会破坏优良个体, 所以变异操作的概率要很小。遗传算法的理论基础是图式定理, 在选择、交叉和变异算子作用下, 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。而积木块假设则说明了遗传算法寻找最优样本的能力。

遗传算法中, 采用不同的编码方法, 能有效提高收敛速度。最初模型采用二进制编码, 它得到早期的理论(Schema 定理, 最小字母表原理) 支持, 但当要求的精度较高时, 则有以下问题: ① 二进制编码存在连续函数离散化误差; ② 当个体编码串长度较短时, 达不到精度要求; ③ 太长时, 搜索空间急剧扩大。例如, 对一个有 10 个变量的问题求解, 如果每个变量用 8 位二进制表示, 其中符号位 1 位, 整数 3 位, 小数 4 位, 则精度为 $1/2^4 = 1/16$。所以要提高精度就要增加小数位位数, 如取变量为 16 位, 则个体长度为 $10 \times 16 = 160$, 搜索空间 $2^{160} \approx 1.46 \times 10^{48}$。在如此大的搜索空间寻优, 一定会使遗传算法的运行性能变差, 而且每增加一位, 搜索空间就按指数增加。为了克服二进制编码的缺点, 提出个体的浮点数编码方法。所谓浮点数编码, 是指个体的每个基因值用某一范围内的一个浮点数表示, 个体的编码长度等于其变量个数(这里是权系数和阈值), 因为编码变了, 所以交叉和变异算子也不同, 交叉的条件为 $a + b = a' + b'$, 其中 a 和 b 为父代, a' 和 b' 为子代, 也就是说父代交叉后得到子代个体的数值和等于父代的数值和。变异算子是变异基因 c 在变量区间内变成 c', 其中变异 c' 与 c 符合正态分布。此外, 还有多值编码、区间编码、指数编码等多种编码, 它们各有优缺点, 要根据具体问题采取不同的编码。

遗传算法的选择算子也是优化的关键技术之一。选择算子是将优良个体保留下来, 如果选择算子对优良个体保留的概率过大, 将会造成几个较好的个体迅速占据整个种群(即收敛入局部解) 的情况; 如果概率过小, 则会与纯粹的随机搜索没区别。为提高选择算子的效率, 于是提出自调整适应度方法, 保证不同时期都能按合适的概率保留种群的优良个体, 保证种群的多样性。另外, 遗传算法中的交叉算子作为遗传算法的核心, 对于遗传算法的优化起重要作用, 下面对此进行详细介绍。

3　提高遗传算法收敛速度的交叉算子

从遗传算法的寻优过程可以看出, 遗传算法是一个以一定概率随机搜索的优化过程, 其选择、交叉和变异算子都是以概率方式进行。交叉、变异的概率过高, 将会破坏优良个体, 使遗传算法变成纯粹的随机搜索; 过低则容易陷入局部最优解, 收敛速度慢。所以交叉和变异的参数选择将影响算法的搜索结果和搜索效率。对于如何改进算法本身以提高求解问题速度, 通过人们在实践中不断调整交叉和变异概率, 提出了自调整交叉和变异概率算法, 即保留最优解不参加交叉和变异操作。但其处理复杂问题时, 由于问题的规模较大, 往往造成搜索空间大, 收敛速度不明显的现象。因此本文提出异位交叉运算的方法, 把交叉和变异在一次操作运算中完成, 以便能更有效地提高遗传算法的收敛速度。

遗传算法中的交叉运算, 是遗传算法区别于其他进化算法的重要特征, 是在遗传过程中产生新个体的主要方法。好的交叉算子既能保留优良个体的优良模式, 又会产生更多更好的个体模式, 并能较快地找到最优点, 提高收敛速度。

常用的交叉算子有单点交叉和两点交叉。单点交叉是指在个体编码串中随机设置一个交叉点, 然后在该点互相交换两个配对个体的部分染色体。这种交叉算子破坏个体形状的可能性小, 但产生新个体的可能性也小, 所以对于复杂的问题这种交叉算子收敛速度很慢。两点交叉是指在个体编码串中随机设置两个交叉点, 然后进行部分基因交换。如图 1 所示。这种交叉对于保存优良模式和产生新个体比较适中。再配合自调整适应度和自调整交叉、变异概率, 收敛速度就有所提高。但收敛速度的改善还很有限, 往往满足不了要求, 因而在此引出新的算子 —— 异位交叉算子, 如图 2 所示。

经反复研究和实验证明, 这种交叉能有效保留个体优良模式并产生更多更优的新个体, 与上述对

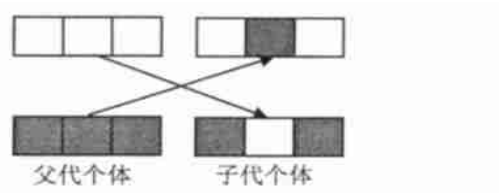

图 1　两点交叉

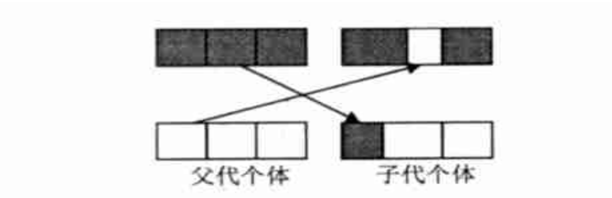

图 2　异位交叉

等交叉相比, 具有很大的优势, 它能大幅度提高遗传算法的收敛速度。

4　模拟试验

本文采用如下 3 个典型测试函数检测本文提出的异位交叉算子的性能。在此二进制编码, 个体长度 48 位, 种群个体数 100。对等位交叉使用自调整适应度, 自调整交叉、变异概率。异位交叉使用固定交叉概率 0.25, 变异概率 0.05。

4.1　测试函数 1

$$f_1(x_1, x_2, x_3) = \sum_{i=1}^{3} x_i^2, \quad -5.12 \le x_i \le 5.12$$

它为平方和函数, 只有一个极小点 $f_1(0, 0, 0) = 0$。实验中迭代 20 次, 求平均值, 结果列于表 1。

表 1　函数 1 仿真结果

	平均迭代代数	平均输出偏差	最大迭代代数
对等位交叉	244.7	3.5e-6	557
异位交叉	25.7	6.64e-7	40

4.2　测试函数 2

$$f_2(x_1, x_2) = 0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$$
$$-100 \le x_i \le 100$$

它是一个多峰值函数, 在其定义域内只有一个全局极小点 $f_2(0, 0) = 0$。采用上述的参数迭代最大代数 10 000 次, 重复迭代 15 次, 结果如表 2 所示。

表 2　函数 2 仿真结果

	未收敛次　数	平均迭代次数	平均输出偏差	最大迭代代数
对等位交叉	12	8 624.3	0.003 587 2	- -
异位交叉	0	1 446	3.29e-9	7 188

注: 在对等位交叉时, 因为陷入局部解不收敛, 所以最大迭代代数无法给出。

4.3　测试函数 3

用遗传算法求神经网络的权系数, 这个正向前馈神经网络实现正弦曲线为

$$y = -0.44\sin(2\pi x) + 0.5, \quad 0 \le x \le 1$$

神经网络结构为 1-4-4-4-1。

这是一个复杂的组合优化问题, 待求未知量有 53 个(权系数和阈值), 这里采用 16 位二进制为每个权系数编码, 每个个体字符串长度为 848 位。适应度函数为

$$f = \frac{1}{E} = \frac{1}{\left\{\sum_{i=0}^{10} [\mathrm{out}(i) - y(i)]^2\right\}}$$

其中: out(i) 是神经网络输出, $y(i)$ 是理论期望值。

对于如此大的搜索空间($2^{849} = 3.7 \times 10^{255}$), 如果要有效地找到更优良的图式, 找出最优解, 搜索的代数很大, 但用异位交叉则能很快地组合出更多更优良图式, 且搜索速度很快。实验也是用自调整适应度, 自调整交叉、变异概率的对等位交叉, 与非异位交叉进行比较。所得实验结果如表 3 所示。

表 3　函数 3 仿真结果

	1 000 代平均最大适应度(误差)	10 万代最大适应度(误差)
对等位交叉	9.076(0.110 2)	187.813(0.005 32)
异位交叉	39.979 8(0.025 01)	1 319.655 8(7.578e-4)

5　结　　论

本文对于遗传算法中的交叉算子, 提出一种新的算子——异位交叉。通过实验分析表明, 对于一些简单函数优化或多峰值函数优化, 以及复杂的组合优化(求神经网络权系数), 用一般的对等位交叉算子收敛速度比用异位交叉算子慢, 异位交叉表现出明显的优越性, 它能有效地提高遗传算法的运行效率和全局收敛性。

参考文献(References):

[1] 余永权. 神经网络模糊逻辑控制[M]. 北京: 电子工业出版社, 1999.

[2] 徐丽娜. 神经网络控制[M]. 哈尔滨: 哈尔滨工业大学出版社, 1999.

[3] 史奎凡, 陈月辉. 提高遗传算法收敛速度的方法[J]. 信息与控制, 1998, 8(4): 289-293.
(Shu Kuifan, Chen Yuehui. A method of improving the convergence speed of genetic algorithm[J]. *Information and Control*, 1998, 8(4): 289-293.)

[4] 张彤, 张华, 王子才. 浮点数编码的遗传算法及其应用[J]. 哈尔滨工业大学学报, 2000, 32(4): 59-61.
(Zhang Tong, Zhang Hua, Wang Zicai. Float encoding genetic algorithm and its application[J]. *J of Harbin Institute of Technology*, 2000, 32(4): 59-61.)

[5] 周明, 孙树栋. 遗传算法原理及应用[M]. 北京: 国防工业出版社, 1999.

[6] 楼顺天, 施阳. 基于 Matlab 的系统分析与设计——神经网络[M]. 西安: 西安电子科技大学出版社, 1999.