第 16 卷 第 6 期　　控 制 与 决 策　　2001 年 11 月
Vol 16 No. 6　　CON TROL AND DECIS ION　　Nov. 2001

文章编号: 1001-0920(2001)06-0926-04

适应值共享拥挤遗传算法

于歆杰, 王赞基
(清华大学 电机工程与应用电子技术系, 北京 100084)

摘　要: 保持遗传算法在演化过程中的种群多样性, 是将遗传算法成功应用于解决多峰优化问题和多目标优化问题的关键。适应值共享遗传算法和拥挤遗传算法分别从不同角度改善了遗传算法的搜索能力, 是寻找多个最优解的常用算法。将这两种算法的优点加以结合, 提出适应值共享拥挤遗传算法。数值测试结果表明, 该算法比标准适应值共享遗传算法和确定性拥挤遗传算法具有更强的搜索能力。

关键词: 遗传算法; 适应值共享; 拥挤; 多峰优化
中图分类号: TP 18　　**文献标识码**: A

Fitness Sharing Crowding Genetic Algorithm

YU Xin-jie, WANG Zan-ji
(Department of Electrical Engineering, Tsinghua University, Beijing 100084, China)

Abstract: By combining fitness sharing method and crowding method in selection stage and replacement stage of genetic algorithms respectively, a fitness sharing crowding genetic algorithms is proposed. It combines the advantages of both fitness sharing genetic algorithms and crowding genetic algorithms in searching ability. The suggested algorithm is proved to be more efficient in solving benchmark multimodal optimization problems than standard fitness sharing genetic algorithm and crowding genetic algorithm.

Key words: genetic algorithms; fitness sharing; crowding; multimodal optimization

1 引　　言

遗传算法是模拟生物界“适者生存”进化原则的一种现代优化算法。该算法的最大特点在于它始终维持一群解, 即生物界“种群”的概念, 因而, 对于一些多峰函数的极值求解或多目标的优化问题, 比其它优化算法具有一定的优势。但是, 在种群规模有限或选择压力不合适的情况下, 遗传算法也可能只收敛到一个解, 这就是所谓的“遗传漂移”现象[1, 2]。解决“遗传漂移”的关键在于保持遗传算法演化过程中的种群多样性(Population Diversity)。遗传算法通常由选择、复制和替换三个过程组成。如果没有特殊干预手段, 在遗传算法进行的早期, 种群比较丰富; 到了晚期, 种群就会比较单一。如何使遗传算法既能收敛又保持相对丰富的种群多样性, 是遗传算法理论研究的关键问题之一。

适应值共享遗传算法(Fitness Sharing Genetic Algorithms)[3, 4]是使遗传算法保持种群多样性的一种有效的改进方法。在多峰函数求解的遗传算法中, 通常把解空间中峰周围的子空间比作生物生长的小生境, 把峰周围的个体比作在该小生境中繁衍的物

收稿日期: 2000-06-26; **修回日期**: 2000-10-20
作者简介: 于歆杰(1973—), 男, 北京人, 讲师, 博士生, 从事现代优化算法、电路与系统理论等研究; 王赞基(1946—), 男, 福建福清人, 教授, 博士生导师, 博士, 从事电工理论与新技术领域的教学与科研工作。

种。所谓适应值共享, 就是将该小生境中所有个体的适应值按照物种的规模以一定的方式降低。显然, 如果某个小生境中有较多的个体, 那么该小生境中所有个体的适应值将以较大幅度降低。由于适应值共享遗传算法是在遗传算法的选择过程之前用适应值共享的思想处理每个个体的适应值, 因此, 小规模物种的被选择概率会比适应值共享之前有所提高, 从而增加了种群的多样性。

拥挤遗传算法(Crowding Genetic Algorithms)是保持遗传算法种群多样性的另一种有效的改进方法。其基本思想是当物种繁衍到一定规模, 以至于生存空间变得非常拥挤时, 新产生的个体要想生存, 就必须与种群中的其它个体进行竞争。拥挤遗传算法的实质就是在替换过程中采用拥挤思想。竞争方式的不同以及评价个体生存能力的判据不同, 便形成了各种不同类型的拥挤遗传算法, 如子个体与父个体进行竞争的预选择方法; 考虑父子个体间距离和适应值的确定性拥挤方法[5,6]; 从当前种群中随机选出 W 个个体与新个体进行竞争的方法[7]; 考虑个体适应值影响的限制性锦标选择方法[8]; 将竞争机制转移到旧个体中的替换方法[9] 等。一般来说, 采用拥挤思想的遗传算法可以较好地维持种群多样性, 但由于拥挤思想强调的是在替换过程中实现优胜劣汰, 不太重视选择过程的作用, 因此算法的收敛能力较弱。

适应值共享遗传算法是在选择过程之前使用共享思想, 而其替换过程则直接将新个体群全部替换老个体群; 拥挤遗传算法在替换过程中使用拥挤思想, 而其选择过程则采用顺序选择或随机选择。如果能将适应值共享思想和拥挤思想有机结合, 分别在遗传算法的选择过程中调整个体的适应值, 而在替换过程中加入比较竞争手段, 则可取长补短, 提高算法的搜索能力。这便是本文提出的适应值共享拥挤遗传算法的基本思想。

2　适应值共享拥挤遗传算法

适应值共享拥挤遗传算法的出发点是在选择过程和替换过程中均采用相应的策略来维持合理的种群多样性, 从而有可能使遗传算法既保证收敛又能找到多个峰。该算法的基本内容是在标准遗传算法的基础上, 在选择阶段之前, 利用适应值共享的思想进行适应值调整; 而在替换阶段, 则采用确定性拥挤的思想。算法的主要步骤如下。

Step1　用适应值共享的方法计算每个个体的共享后适应值。

设 $d(i,j)$ 是两个个体 i 和 j 之间的距离, 则它们之间的共享函数为

$$\mathrm{sh}(d(i,j)) = \begin{cases} 1 - \left(\dfrac{d(i,j)}{\sigma}\right)^{\alpha}, & d(i,j) < \sigma \\ 0, & \text{其　它} \end{cases} \tag{1}$$

其中, σ 为事先指定的峰半径, 通常是已知的或假设的; α 为控制共享函数形状的参数, 通常 $\alpha = 1$, 即线性共享函数。在得到所有个体的共享函数值之后, 可由下式计算个体的小生境数

$$m_i = \sum_{j=1}^{N} \mathrm{sh}(d(i,j)), \quad i = 1, 2, \cdots, N \tag{2}$$

其中 N 是个体的数目, 代表种群的规模。显然, 某个体有较大的小生境数便意味着该个体的周围聚集有较多的个体。然后, 计算共享后个体的适应值

$$f_i' = f_i / m_i \tag{3}$$

其中 f_i 为个体 i 共享前的适应值。随后进行的选择过程将使用共享后个体的适应值。如果某个物种有较多的个体, 那么该物种中所有个体的适应值将以较大的幅度降低, 从而鼓励有较少个体的物种繁衍。假设第 i 个物种中所有个体的适应值均为 f_i, N_i 表示该物种的个体数, k 代表小生境的数量, 则适应值共享的稳定状态可以表示为

$$f_i / N_i = f_j / N_j$$

其中 $i \neq j$, 且 $\sum_{i=1}^{k} N_i = N$。这就意味着在稳定状态下, 个体的分布依赖于峰的数量和峰的高度, 而不是收敛到唯一的峰上。

Step2　采用随机误差比较小的选择方法进行选择过程。

Step3　随机地选取个体对进行杂交和变异, 产生新个体对 c_1 和 c_2 (其对应的父个体对为 p_1 和 p_2)。

Step4　在替换阶段采取确定性拥挤的思想来决定下一代中的个体。下面以 c_1, c_2, p_1, p_2 的竞争为例说明具体方法。

如果 $[d(p_1,c_1) + d(p_2,c_2)] \leq [d(p_1,c_2) + d(p_2,c_1)]$, 则

　　如果 $f(c_1) > f(p_1)$, 则用 c_1 替换 p_1, 否则保留 p_1

　　如果 $f(c_2) > f(p_2)$, 则用 c_2 替换 p_2, 否则保留 p_2

否则

　　如果 $f(c_2) > f(p_1)$, 则用 c_2 替换 p_1, 否则

保留 p_1

如果 $f(c_1) > f(p_2)$，则用 c_1 替换 p_2，否则保留 p_2

其中，N 是种群规模，$d(i,j)$ 是个体 i 和个体 j 之间的距离。

在计算个体共享后适应值阶段，适应值共享拥挤遗传算法中每个个体均需要计算 N 次与其它个体之间的距离。对于 N 个个体来说，距离计算的时间复杂度为 $O(N^2)$。在实施新老个体竞争阶段，适应值共享拥挤遗传算法中每个个体需要计算两次距离，距离计算的时间复杂度为 $O(N)$。因此，适应值共享拥挤遗传算法的总体距离计算时间复杂度为 $O(N^2)$。

拥挤遗传算法由于仅采用顺序或随机的选择方法，使优秀个体的优势在下一代中不能得到完全体现；而适应值共享拥挤遗传算法因其采取随机误差较小的选择方法，可以充分体现“适者生存”的生物进化原则，因而提高了求解的质量。

3 适应值共享拥挤遗传算法的测试

为充分比较和考察适应值共享遗传算法(SH)、确定性拥挤遗传算法(DC)和适应值共享拥挤遗传算法(SC)的搜索能力，本文采用两个峰高相同的标准问题作为上述三种方法的测试问题。

3.1 对测试问题的描述

问题 1[10]

$$F_1(x) = \sin^6[5\pi(x^{0.75} - 0.05)] \tag{4}$$

该问题的定义域为[0, 1]，包含 5 个不均匀分布的峰，分布在 0.080, 0.247, 0.451, 0.681 和 0.934 处，峰高为 1.0。

问题 2[11]

$$f(x_0, x_1, \cdots, x_{29}) = \sum_{i=0}^{4} u\left(\sum_{j=0}^{5} x_{6i+j}\right) \tag{5}$$

其中，$\forall x_k \in \{0, 1\}, k = 0, 1, \cdots, 29$，$u(s)$ 定义为

$$u(s) = \begin{cases} 1, & s \in \{0, 6\} \\ 0, & s \in \{1, 5\} \\ 0.360\,384, & s \in \{2, 4\} \\ 0.640\,576, & s = 3 \end{cases} \tag{6}$$

该问题也被称为复杂欺骗性问题，其搜索空间规模为 10^9，峰的个数为 10^6，而全局峰的个数仅有 32 个。之所以称作复杂欺骗性问题，不仅因为该问题巨大的局部峰个数，而且因为众多局部峰包围着全局峰，这就使得遗传算法的杂交算子和变异算子很容易产生局部峰。峰分布在(000 000, 000 000, 000 000, 000 000, 000 000)，(000 000, 000 000, 000 000, 000 000, 111 111)，…，(111 111, 111 111, 111 111, 111 111, 111 111) 上，峰高为 5.0。

3.2 算法参数的选取

在测试过程中，所有算法均取相同的参数。具体的算法参数列于表 1。

表 1 算法的测试参数

	问题 1	问题 2
比例变换类型	不采用	指数比例变换
距离度量手段	欧几里得距离	汉明距离
种群规模	60	400
染色体长度	30	30
峰半径(SH, SC)	0.1[4]	6[11]
个体收敛到峰的判据	个体与峰之间的距离小于 0.02	在峰上

在测试过程中，所有算法均采用 SUS 选择方式[12]，单点杂交，杂交概率为 1.0，变异概率为 0.0，最大遗传代数均为 50。在 SH 算法和 SC 算法中使用了限制交配策略[3]。每种算法对每一个问题都进行 20 次优化，每次优化的初始种群均随机选取。三种算法的初始种群相同，排除了随机误差对优化结果的影响。

算法性能的判据有 4 个，分别用来评价算法不同方面的搜索能力：判据 1 是算法找到的全局峰个数，判据值越大，则算法越优秀；判据 2 是类 X^2 判据[3]，判据值越小，则种群中的个体在不同峰上的分布越均匀，算法也越优秀；判据 3 是收敛到全局峰的个体数；判据 4 是算法的运行时间(单位：s)，判据值越小，则算法效率越高。较大的判据 1 和较大的判据 3 表明算法的搜索能力较强，而较小的判据 1 和较大的判据 3 则表明算法出现了“早熟收敛”现象。

三种算法均在 MATLAB 环境中实现，运行于赛扬 466CPU 和 128M 内存的计算机中，所有性能判据均由算法的 20 次计算平均得到。

3.3 测试结果

表 2 为三种算法对两个问题的平均优化结果。

表 2 算法的测试结果

	问题 1			问题 2		
	SH	DC	SC	SH	DC	SC
判据 1	4.8	4.7	4.6	22.0	7.5	25.1
判据 2	3.7	9.2	3.4	17.0	75.2	14.1
判据 3	45	15.6	48	245	400	355
判据 4	71.1	27.6	73.8	2 023.1	912.08	2 040.1

问题 1 代表峰半径不相同的情况, 难度较低, 此时三种算法的优化质量相似。SH 运行时间几乎与 SC 运行时间相同, 而DC 的运行时间则非常短。DC 种群分布的情况劣于其它两种算法的种群分布。

问题 2 是复杂欺骗性问题, 非常难于优化。SH 的解质量尚可, 但运行时间较长。DC 仅能找到 7.5 个全局峰, 说明其搜索能力较弱, 这与 Sareni 的研究结果一致[10]。SC 找到全局峰的数量最多, 而且种群分布比较均匀, 但运行时间也比较长。

4 结　论

本文将适应值共享思想和拥挤思想结合起来, 提出了适应值共享拥挤遗传算法。该算法的时间复杂度与标准适应值共享遗传算法相同。从算法的结构来说, 任何对标准适应值共享遗传算法时间复杂度的改进均可用于适应值共享拥挤遗传算法。由于该算法结合了标准适应值共享遗传算法和确定性拥挤遗传算法在调整种群多样性方面的优点, 因而具有较高的搜索能力, 种群分布比较均匀。对于简单和复杂标准测试问题的计算表明, 适应值共享拥挤遗传算法对问题的依赖性较弱, 可以应用的范围比较广。特别是对于复杂的优化问题, 该方法明显优于标准适应值共享遗传算法和确定性拥挤遗传算法。

参考文献:

[1] Goldberg D E. Genetic algorithms in search, optimization and machine learning[M]. New York: Addison-Wesley, 1989.

[2] Mahfoud S W. Genetic drift in sharing methods[A]. Proc 1st IEEE Conf Evolutionary Computation[C]. NJ: IEEE Press, 1994. 67-72.

[3] Deb K, Goldberg D E. An investigation of niche and species formation in genetic function optimization[A]. Proc 3rd Int Conf Genetic Algorithms[C]. CA: Morgan Kaufmann, 1989. 42-50.

[4] Goldberg D E, Richardson J. Genetic algorithms with sharing for multimodal function optimization[A]. Proc 2nd Int Conf Genetic Algorithms and Their Applica-tions[C]. NJ: Lawrence Erlbaum, 1987. 41-49.

[5] Mahfoud S W. Crowding and preselection revisited[A]. Proc 2nd Conf Parallel Problem Solving from Nature[C]. Amsterdam, 1992. 27-36.

[6] Mahfoud S W. Crossover interactions among niches[A]. Proc 1st IEEE Conf Evolutionary Computation[C]. NJ: IEEE Press, 1994. 188-193.

[7] De Jong K A. Analysis of the behavior of a class of genetic adaptive systems[D]. Univ of Michigan, 1975.

[8] Harik G. Finding multimodal solutions using restricted tournament selection[A]. Proc 6th Int Conf Genetic Algorithms[C]. CA: Morgan Kaufmann, 1995. 24-31.

[9] Cedeno W. The multi-niche crowding genetic algorithem: Analysis and applications[D]. Univ of California, 1995.

[10] Sareni B, Krahenbuhl L. Fitness sharing and niching methods revisited[J]. IEEE Trans on Evol Comput, 1998, 2(3): 97-106.

[11] Goldberg D E, Deb K, Horn J. Massive multimodality, deception and genetic algorithms[A]. Proc 2nd Conf Parallel Problem Solving from Nature[C]. Amsterdam, 1992. 15-25.

[12] Baker J E. Reducing bias and inefficiency in the selection algorithm[A]. Proc 2nd Int Conf Genetic Algorithms and Their Applications[C]. NJ: Lawrence Erlbaum, 1987. 14-21.