

文章编号: 1001-0920(2009)12-1801-04

一种基于多样化成长策略的遗传算法

袁煜明, 范文慧, 杨雨田, 赵常宏, 孙元栋

(清华大学 自动化系, 北京 100084)

摘要: 遗传算法中, 部分局部最优个体常常会抑制种群中其他个体的成长, 使进化停滞, 结果陷入局部最优. 对此, 提出一种基于多样化成长策略的遗传算法, 通过采用迁移杂交、多态变异和群体突变等方法, 对种群中不同适应度的个体给予不同的进化成长策略, 以保持个体间的多样性, 从而增强算法的寻优能力. 对改进算法的算法复杂度进行了分析, 并通过算例表明, 多样化成长策略的各个部分均对改善寻优结果发挥了作用.

关键词: 遗传算法; 多样化成长; 迁移杂交; 多态变异; 群体突变

中图分类号: TP273

文献标识码: A

Genetic algorithm based on diversified development strategy

YUAN Yu-ming, FAN Wen-hui, YANG Yu-tian, ZHAO Chang-hong, SUN Yuan-dong

(Department of Automation, Tsinghua University, Beijing 100084, China. Correspondent: YUAN Yu-ming, E-mail: yym03@mails.tsinghua.edu.cn)

Abstract: In genetic algorithms, some local optimum individuals often inhibit the development of other individuals, so the evolution stagnates with a result of local optimum. Therefore, a genetic algorithm based on diversified development strategy is proposed, which assigns different individuals of different fitness with different development strategy, so as to diversify the population, and enhance the algorithm's optimizing ability through migrated crossover, divergent mutation, collective saltation. The complexity of the algorithm is analyzed. Benchmark example shows that all parts of the strategy play roles for optimum search.

Key words: Genetic algorithm; Diversified development; Migrated crossover; Divergent mutation; Collective saltation

1 引言

遗传算法(GA)是 20 世纪 70 年代由美国 Holland 提出的模仿生物进化过程的优化方法. 其主要思想是基于 Darwin 的生物进化论和 Mendel 的遗传学. 遗传算法基于自然选择, 在计算机上模拟生物进化机制进行寻优搜索^[1-3]. 用遗传算法进行优化计算时, 只需要适应度值, 而不需要导数信息, 也不需要设计空间或函数的连续性条件, 只要遗传进化中每一代都包含上一代的最优个体, 算法就可以执行搜索寻优. 因而遗传算法在优化中得到了广泛的应用^[4].

遗传算法自身也存在不足, 未成熟收敛是其中一个很重要的问题. 具体而言, 部分局部最优个体常常会主导整个种群的进化, 其他有发展潜力的个体, 也即有可能进化到全局最优解的个体则较难被选择到下一代, 使整个进化停滞, 导致最终结果陷入局部

最优^[5]. 如何改善遗传算法的搜索能力, 提高算法的收敛速度, 使其更好地应用于实际问题的解决, 是各国研究者一直探索的主要课题^[6].

针对上述缺陷, 有人从适应度定标^[7]、修改遗传算法的控制参数着手^[8], 或使用自适应的参数计算方法^[9], 将遗传算法和局部搜索算法(如模拟退火、爬山法等)相结合^[10]. 但这些方法仍然存在问题, 如适应度定标方法并不具备普遍适用性, 而遗传算法仍未能很好地解决局部搜索和全局采样之间的矛盾.

为此, 本文提出一种基于多样化成长策略的遗传算法, 通过迁移杂交、多态变异和群体突变等方法, 对于不同种群以及种群中的不同适应度的个体, 给予不同的进化成长策略, 以保持个体间的多样性, 从而增强算法的寻优能力. 仿真算例表明, 多样化成长策略的各个部分均对改善寻优结果发挥了作用.

收稿日期: 2009-01-15; **修回日期:** 2009-04-02.

基金项目: 国家自然科学基金项目(60474059); 国家 863 计划项目(2006AA04Z160).

作者简介: 袁煜明(1984—), 男, 浙江湖州人, 硕士, 从事复杂产品设计、协同优化的研究; 范文慧(1966—), 男, 吉林扶余县人, 副教授, 博士, 从事复杂产品设计、协同优化等研究.

2 多样化成长策略

2.1 迁移杂交

多种群并行遗传算法是近年来提出的一种改进遗传算法性能的较好方法,该方法通过若干个种群的分别发展来扩大搜索空间.而如何在保证各个种群独立性的基础上实现各种群的协同发展,是多种群并行遗传算法研究的重要问题^[11].

本文提出了迁移杂交策略,使得各个种群的精英得以互相借鉴而拓宽进化方向,又不影响各种群原本的进化方向.具体实现方法如下:

- 1) 半数种群采用轮盘赌选择,半数种群采用排序选择;
- 2) 每隔若干代进行一次迁移杂交,在轮盘赌选择的种群与排序选择的种群之间随机配对进行;
- 3) 每次迁移杂交,将配对种群中各自的最优个体进行均匀交叉,产生子个体;
- 4) 用子个体替换掉种群中两个最差个体.

2.2 多态变异

通常的变异操作都是设置一个变异概率,对种群中的每个个体按该概率进行变异.但变异概率的设置是一个难题:变异概率过小,使解有一定的局限性,遍历性较差;变异概率较大,使得进化的随机性增大,也不容易得到稳定的解^[12].

Srinivas 等提出一种自适应遗传算法^[9],使交叉和变异概率与种群适应度的集中度相关联.但该算法对同一个体的各二进制位依然赋予统一的变异概率,在陷入局部最优时仍难以跳出.

针对二进制编码时,染色体不同位对应的权重差别很大,本文借鉴三角波平移的方式,提出了多态变异的策略.根据适应度大小,对于同一个体不同二进制位,赋予不同的变异概率,从而使得适应度高的个体变异步长较小,在最优点附近进行微调以达到最优点;适应度低的个体变异步长较大,以试图找到全局其他的最优点.具体实现方法如下:

- 1) 对种群中的个体按适应度排序;
- 2) 根据适应度序与染色体长度确定每个个体的主变异位,将该位的变异权重值设置为染色体长度值;
- 3) 从主变异位往两侧每移一位,变异权重值减1;
- 4) 在设置完该个体所有位的变异权重值后,按变异权重值的比例给每一位确定变异概率;
- 5) 进行变异操作.

以8位染色体、每个种群有160个个体为例,图1给出了每一位的变异概率示意.

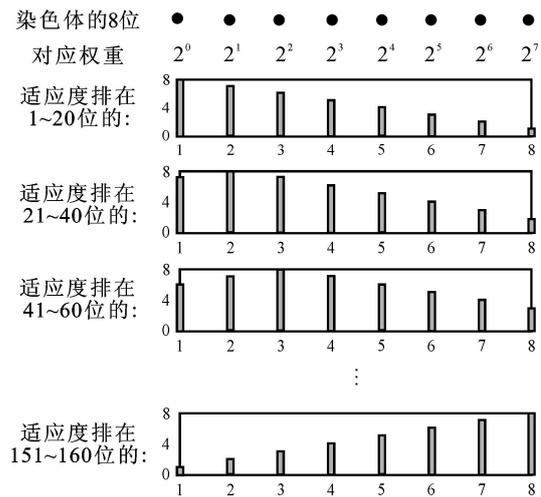


图1 多态变异的变异概率分配示意

2.3 群体突变

以上两种策略已基本避免了陷入局部最优,但仍会偶尔出现.为此,本文提出了群体突变策略.如果种群最优解持续若干代不变,则认为其无法自行再进化,有可能陷入局部最优,需对种群进行颠覆性的改变,从而跳出可能陷入的局部最优点.具体实现方法为:

- 1) 每一代均对最优解进行检查,如果连续若干代均不变,则在选择操作后触发群体突变;
- 2) 群体突变时,小部分精英(用户可设定比例)将直接保留到下一代,避免原有最优解的流失;
- 3) 群体突变时,精英外的个体均采用一个比原本大得多的变异概率进行变异,一般来说变异概率在0.5左右,从而使这些个体均被颠覆;
- 4) 由于群体突变已经是大概率变异了,该代的变异操作将被省去.

2.4 策略与流程

对于种群中不同适应度的个体,本文采用了不同的成长策略,具体如表1所示.

表1 不同个体的成长策略

个体	适应度	策略	常规操作	触发操作
领头	最大	每代均保留,与其他种群领头相互借鉴	极小步长变异	迁移杂交
精英	较大(一小部分)	用变异进行微调	较小步长变异	无
平民	其他	用变异进行较大幅度调节,必要时进行颠覆	较大步长变异,随适应度减小而增大	群体突变

基于多样化成长策略的遗传算法,主要在产生下一代的方法上与传统遗传算法不同,具体流程如图2所示.

传统的遗传算法在产生下一代上主要包括选

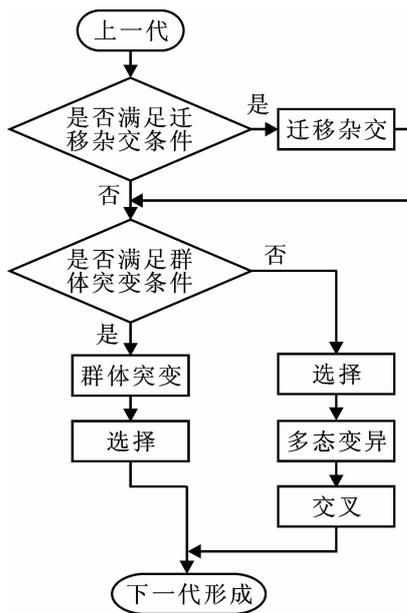


图 2 基于多样化成长策略的遗传算法流程

择、交叉和变异 3 个步骤。而基于多样化成长策略的遗传算法需先判断是否需要执行迁移杂交,如需要,则执行操作;然后判断是否需要执行群体突变,如需要,则执行操作;再进行选择、多态变异和交叉;如果已经执行了群体突变,则可以跳过多态变异操作。

2.5 算法复杂度分析

首先对算法复杂度分析中用到的参数进行说明。

Popsize: 种群中的个体数量;

Popnum: 种群数目;

N_m : 迁移杂交的间隔代数;

N_r : 群体突变的间隔代数;

N_t : 总运行代数, 设其为常量。

对于迁移杂交, 每次操作增加的算法复杂度为 $O(\text{Popnum})$, 而每次间隔 N_m 代进行一次, 因此, 总共增加的算法复杂度为 $N_t/N_m * O(\text{Popnum})$ 。

对于多态遗传, 每次操作增加的算法复杂度为

$O(\text{Popsize} * \text{Popnum})$, 且每代均进行, 因此, 总共增加的算法复杂度为 $N_t * O(\text{Popsize} * \text{Popnum})$ 。

对于群体突变, 由于其算法复杂度与变异相当, 且替代了原来的变异操作, 算法复杂度的增加仅为每代中检验是否触发群体突变, 总共增加的算法复杂度为 $N_t * O(1)$ 。

综上, 可以得到多样化成长策略所增加的算法复杂度为 $N_t * O(\text{Popsize} * \text{Popnum})$, 为线性增加, 不会造成运行时间成倍增加。

3 算例验证

3.1 测试算例

为检验多样化成长策略的遗传算法的寻优性能, 本文采用如下测试案例:

$$g_1(x) = x_1^2 - x_2^2 - 0.01 \leq 0,$$

$$g_2(x) = x_1^2 + x_2 + 1 \geq 0,$$

$$-100 \leq x_1, x_2 \leq 100,$$

$$f(x) = \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{(1 + 0.001(x^2 + y^2))^2} - 0.5 \rightarrow \min.$$

以上是 Shaffer's F_6 函数, 其中只有一个 $(0, 0)$ 为全局最小。最小值周围有一个圈脊, 取值均为 -0.990283 , 很容易停滞在此局部最小点。

3.2 测试结果

对于传统遗传算法、多样化成长策略的遗传算法、以及去掉算法中任一部分后的算法, 总共 5 种算法进行测试, 每种均测试 10 次, 表 2 列出了 5 种算法 10 次中的最差结果, 以及 10 次的正确率。

由表 2 可以看到, 对于传统遗传算法, 或是省略了任意一部分的算法, 正确率均无法达到 100%, 最坏情况与理想目标值存在偏差, 都容易陷入局部最优。

图 3, 图 4 给出了两种算法的最差结果下, $x_1 \sim x_4$ 的历史迭代图。

理想结果应当是两个变量最后均取 0。但从图中可以看到, 传统遗传算法的结果中两个设计变量

表 2 算例测试结果

指标	传统遗传算法	多态变异 + 群体突变	迁移杂交 + 群体突变	迁移杂交 + 多态变异	完全的多样化成长策略的遗传算法: 迁移杂交 + 多态变异 + 群体突变	
运行时间 /s	1.5	2.3	10.5	18.0	14.1	
迭代次数	391	473	319	347	333	
最差结果	最优解	$X_1 = 2.022$ $X_2 = -2.401$	$X_1 = -1.058$ $X_2 = 2.955$	$X_1 = 0.5708$ $X_2 = 3.086$	$X_1 = -1.047$ $X_2 = 2.959$	$X_1 = -4.768 \times 10^{-5}$ $X_2 = 4.768 \times 10^{-5}$
	目标函数值	$Y = -0.990284$	$Y = -0.990284$	$Y = -0.990284$	$Y = -0.990284$	$Y = -1.000000$
正确率 /%	30	80	80	90	100	
理想目标值	$Y = -1.000000$	$Y = -1.000000$	$Y = -1.000000$	$Y = -1.000000$	$Y = -1.000000$	

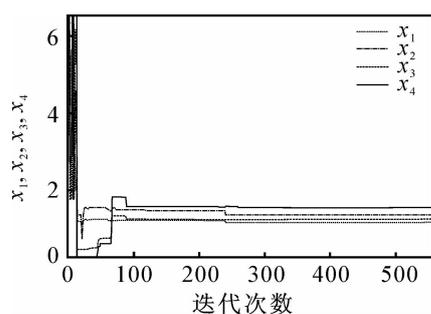


图3 传统遗传算法

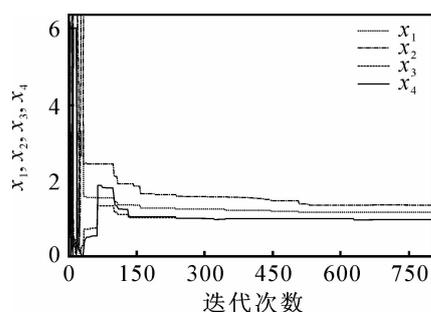


图4 多样化成长策略的遗传算法

最后也不会都达到 0。

由此验证了,本文提出的迁移杂交、多态变异和群体突变 3 部分均对改善寻优结果发挥了作用。

4 结 论

针对遗传算法中存在的未成熟收敛、陷入局部最优问题,本文提出了一种基于多样化成长策略的改进遗传算法来改进寻优性能,包括迁移杂交、多态变异、群体突变 3 部分。算例验证表明,3 部分均对改善寻优结果发挥了作用;而算法复杂度的分析表明,改进的遗传算法的算法复杂度仅是线性增加。

需要指出的是,当前的方法仅适用于单目标优化问题。如果用以解决多目标优化问题,由于领头与精英的定义均有变化,无法直接照搬算法。下一步将对多样化成长策略相关的参数进行调整,使其可以应用到非支配排序遗传算法(NSGA)中,从而可有效地解决多目标优化问题。

参考文献(References)

- [1] Holland J H. Adaptation in nature and artificial system [M]. Ann Arbor: University of Michigan Press, 1975.
- [2] De Jong K A. The analysis of the behavior of a class of genetic adaptive systems[D]. Ann Arbor: University of Michigan, 1975.
- [3] Goldberg D E. Genetic algorithms in search,

optimization and machine learning [M]. Boston: Addison-Wesley Longman Press, 1989.

- [4] 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.
(Zhou M, Sun S D. Principle and application of genetic algorithms [M]. Beijing: National Defense Industry Press, 1999.)
- [5] 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.
(Wang X P, Cao L M. Genetic algorithm — Theory, application and software implementation [M]. Xi'an: Xi'an Jiaotong University Press, 2002.)
- [6] 葛继科, 邱玉辉, 吴春明, 等. 遗传算法研究综述[J]. 计算机应用研究, 2008, 25(10): 2911-2916.
(Ge J K, Qiu Y H, Wu C M, et al. Summary of genetic algorithms research [J]. Application Research of Computers, 2008, 25(10): 2911-2916.)
- [7] Kreinovich V, Quintana C, Fuentes O. Genetic algorithms—What fitness scaling is optimal[J]. Cyber and Systems, 1993, 24(1): 9-26.
- [8] Eiben A E, Hinterding R, Michalewicz Z. Parameter control in evolutionary algorithms[J]. IEEE Trans on Evolutionary Computation, 1993, 3(2): 124-141.
- [9] Srinivas M, Patnaik L M. Adaptive probabilities of crossover and mutation in genetic algorithms[J]. IEEE Trans on Systems, Man and Cybernetics, 1994, 24(4): 656-667.
- [10] Orvosh D, Davis L. Using a genetic algorithm to optimize problems with feasibility constraints[C]. Proc of 1st IEEE Conf on Evolutionary Computation. Orlando, 1994: 548-553.
- [11] 巩敦卫, 孙晓燕. 变搜索区域多种群遗传算法[J]. 控制理论与应用, 2006, 23(2): 256-260.
(Gong D N, Sun X Y. Multi-population genetic algorithm with variational search areas [J]. Control Theory and Applications, 2006, 23(2): 256-260.)
- [12] 张焯, 崔杜武, 黑新宏, 等. 一种改进变异控制策略的遗传算法研究[J]. 西安理工大学学报, 2002, 18(1): 54-57.
(Zhang Y, Cui D W, Hei X H, et al. An improved mutation control method of genetic algorithms [J]. J of Xi'an University of Technology, 2002, 18(1): 54-57.)