

一种改进的自适应遗传算法

于光帅, 于宪伟

(渤海大学 数理学院, 辽宁 锦州 121013)

摘 要: 针对 IAGA 自适应遗传算法存在的未成熟收敛问题, 提出了一种改进的自适应遗传算法 (NIAGA 算法), 根据自定义判别式判断群体是否出现了未成熟收敛趋势, 由不同情况, 分别采用宏观调控与微观处理两种方法来设置交叉概率 p_c 和变异概率 p_m , 以此促使算法摆脱未成熟收敛. 仿真结果表明, 新算法有效地改善了 IAGA 算法的未成熟收敛问题, 显示出了更强的全局收敛性.

关键词: 自适应遗传算法; 交叉概率; 变异概率; 收敛性

1 引言

遗传算法^[1](Genetic Algorithm, 简称 GA) 于 1975 年被首次提出, 创始人为美国 Michigan 大学的 John Holland 教授, 它是通过模拟生物界的遗传进化过程而形成的全局优化概率搜索算法^[2]. 其初始变量是任一群体, 通过执行它特有的遗传操作——选择、交叉和变异操作使群体逐步搜索到解空间中越来越好的区域, 直至满足收敛判定依据或预设的最大迭代次数为止, 它是一种迭代式算法.

在传统遗传算法中交叉概率和变异概率为经验值, 且固定不变. 而一般认为, 若交叉概率过小由于群体规模在进化过程中缺乏多样性而极易陷入局部最优; 若交叉概率过大, 虽然保证了群体的多样性但当后期搜索到最优点临近区域时会使得个体们难以逼近最优点这将大大减慢群体的收敛速度. 而为了防止进化中途停滞, 算法设置了变异算子, 以此来增加群体的多样性, 但如果取固定的变异概率那么群体在经过多次迭代后会逐渐趋同, “近亲繁殖”由此形成, 而这会直接影响后代个体的性能.

对此, M.Srinivas 等提出了一种自适应遗传算法 (AGA)^[3], 当个体适应度较高时, 使交叉概率 p_c 和变异概率 p_m 减小; 当个体适应度较低时, 使交叉概率 p_c 和变异概率 p_m 增大, 即算法在每次迭代中, 都根据个体适应度值的不同来自适应设置 p_c 和 p_m , 这种自适应遗传算法具有更好的效率和全局最优性. 目前, 许多学者都已经认识到了动态自适应技术的优越性并发表了大量的学术论文^[4-5], 但自适应遗传算法作为一种优化方法, 它自身也存在着一定的局限性, 其中最大的不足之处就是容易出现过早收敛即未成熟收敛^[6].

未成熟收敛是指当算法还未搜索到全局满意解或最优解时, 群体所生成的后代的性能就已无法再超越其父代, 个体们彼此间几乎没有差异. 主要表现在两个方面: 群体中各个个体因趋同而停止进化; 当个体搜索到最优解附近时, 却总是被淘汰, 使得进化徘徊不前. 它的产生主要是因为迭代过程中, 尚未得到满意解或最优解, 群体就失去了多样性. 未成熟收敛是一种十分常见的现象, 且属于遗传算法所特有, 也是遗传算法的研究重点. 它与我们常说的局

收稿日期: 2014-12-27

部极值问题相似,但二者在本质上又截然不同,因为未成熟收敛提前得到的“最优解”不见得就是局部极值,而且它附有很强的随机性,人们几乎无法预见它会产生.但是为了更好地研究与应用遗传算法,我们又必须解决未成熟收敛问题,否则它将阻碍遗传算法发挥出全局寻有能力等优良性能.

2 IAGA 自适应遗传算法^[7]

IAGA 算法是对 M.Srinivas 等提出的自适应遗传算法 (AGA) 的改进,其中 p_c 和 p_m 表达式如下:

$$p_c = \begin{cases} p_{c1} - \frac{(p_{c1}-p_{c2})(f-f_{avg})}{f_{max}-f_{avg}}, & f \geq f_{avg} \\ p_{c1}, & f < f_{avg} \end{cases}$$

$$p_m = \begin{cases} p_{m1} - \frac{(p_{m1}-p_{m2})(f_{max}-f')}{f_{max}-f_{avg}}, & f \geq f_{avg} \\ p_{m1}, & f < f_{avg} \end{cases}$$

式中, f_{max} 代表群体中的最大适应度值, f_{avg} 代表群体的平均适应度值, f 代表待交叉两个个体适应度值中的较大值, f' 代表待变异个体的适应度值. $p_{c1} = 0.9$, $p_{c2} = 0.6$, $p_{m1} = 0.1$, $p_{m2} = 0.01$.

IAGA 算法使 p_c 和 p_m 随着每代个体适应度值的变化而自行调整,既保护了群体中优质的个体,又另具有最大适应度值的个体的 p_c 和 p_m 不为 0,这会使每代的最优个体不会处于一种不发生变化的状态.但根据公式,较差个体的变异能力较低,容易产生停止现象,且当群体中优良个体的数目过大时,整体会陷入停滞不前,不易跳出局部最优解,造成未成熟收敛.即以个体为单位,缺乏整体协作的精神,当出现未成熟收敛的趋势时,无法处理.

3 改进的 IAGA 算法

遗传算法是希望在保证群体有丰富的多样性的前提下,平稳顺利地搜索,最终找到满意解或最优解,并不是在此之前,群体就提前收敛,得到一个早熟且非优的所谓的“最优解”.为此,本文对 IAGA 算法进行了改进,提出了一种新的自适应遗传算法 (NIAGA),描述如下:

$$p_c = \begin{cases} 0.8 \frac{f_{max}-f_{avg}}{f_{avg}-f_{min}+\lambda} & \text{当 } \frac{f_{max}-f_{avg}}{f_{avg}-f_{min}+\lambda} < 1 \text{ 且 } M_1 > M_2 \text{ 时} \\ p_{c1} - \frac{(p_{c1}-p_{c2})(f-f_{min})}{f_{max}-f_{min}} & \text{其它} \end{cases}$$

$$p_m = \begin{cases} 0.1 \frac{f_{max}-f_{avg}}{f_{avg}-f_{min}+\lambda} & \text{当 } \frac{f_{max}-f_{avg}}{f_{avg}-f_{min}+\lambda} < 1 \text{ 且 } M_1 > M_2 \text{ 时} \\ p_{m1} - \frac{(p_{m1}-p_{m2})(f'-f_{min})}{f_{max}-f_{min}} & \text{其它} \end{cases}$$

式中, f_{min} 代表群体中的最小适应度值; M_1 代表每代个体中适应度值大于平均适应度值的个体的数目; M_2 代表每代个体中适应度值小于平均适应度值的个体的数目; λ 为一无穷小正数,主要为了防止分母等于 0; 其余部分定义同上.

当判别式 $\frac{f_{max}-f_{avg}}{f_{avg}-f_{min}+\lambda} < 1$ 时,说明该代群体的 f_{avg} 向 f_{max} 靠拢,此时群体中适应度值大于平均适应度值的个体占主导地位,而这些优秀个体被复制到下一代的几率较大,较差个体又在被逐渐淘汰,若以此下去,群体中的各个个体将非常相似,群体的多样性将急剧下降,则此时算法出现了未成熟收敛趋势,且判别式比值越小,未成熟收敛趋势越大.

而 NIAGA 算法, 对于出现未成熟收敛趋势的情况, 在表达式中, 利用指数函数的性质, 根据趋势程度大小, 宏观上对群体统一设置同一较大的 p_c 、 p_m , 以此来加大交叉与变异的力度, 使群体更多地产生新型个体, 增加群体的多样性, 从而打破该状态, 使群体不会因趋同而停止收敛, 跳出局部最优解. 而对于未出现未成熟收敛的情况, 则根据个体适应度值的不同, 自适应地调整 p_c 、 p_m , 即微观上分别对每个个体设置 p_c 、 p_m , 使得适应度值较高的个体有较小的 p_c 、 p_m , 防止优良模式被中途打破, 使之得以保留; 使得适应度值较低的个体有较大的 p_c 、 p_m , 使其劣质模式被迅速打破, 从而随着遗传操作的进行, 使得其产生的新个体若质量优良, 则会被保留, 若质量低劣, 则会被淘汰. NIAGA 算法将宏观与微观两种思想相结合, 既考虑了出现未成熟收敛时对整体的宏观调控, 又考虑了未出现未成熟收敛时对每个个体的微观处理, 这样既防止了算法尚未得到最优解就提前收敛, 又加快了搜索速度.

4 仿真实验

本文在 window7, Intel(R) Core(TM)2 Duo CPU T6600 @ 2.20GHz, RAM: 2.00GB, MATLAB2009 的环境下, 选取六个公认的具有比较意义的 benchmark 测试函数 [8], 然后在相同种群规模的条件下, 对六个测试函数相对 IAGA 和 NIAGA 两种算法分别运行 50 次求取函数最小值, 图 1 至图 6 为各函数迭代求解时得到的每步平均最优值的收敛情况, 表 1 为计算统计结果.

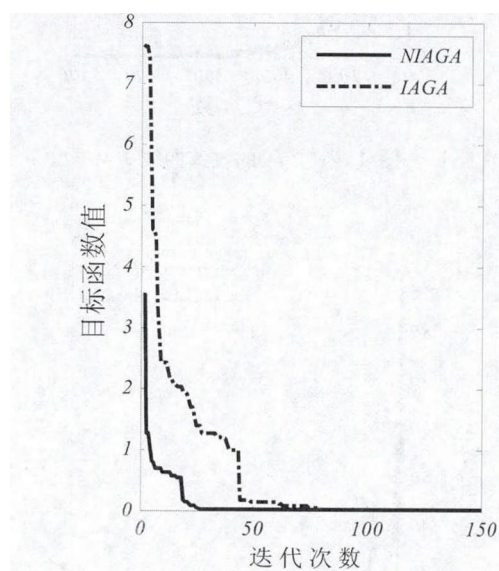


图 1 函数 f_1 的收敛曲线

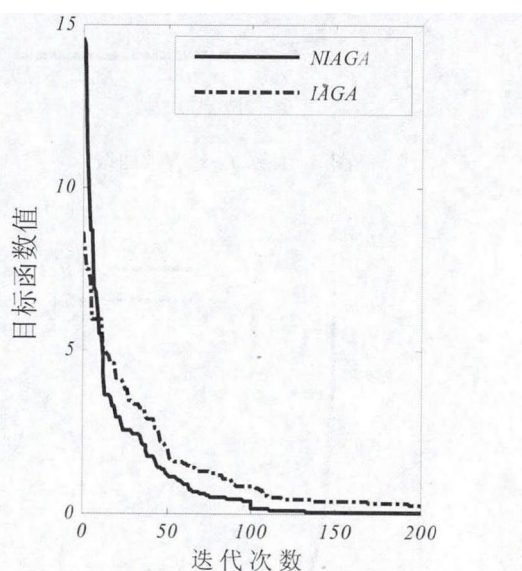


图 2 函数 f_2 的收敛曲线

其中:

$$f_1(x) = \sum_{i=1}^{10} |x_i|^{(i+1)}, [-2, 2]^{10} \quad f_2(x) = \sum_{i=1}^{20} x_i^2, [-2, 2]^{20}$$

$$f_3(x) = \sum_{i=1}^9 [(1-x_i)^2 + 100(x_{i+1} - x_i^2)^2], [-2, 2]^{10}$$

$$f_4(x) = \sum_{i=1}^{20} (x_i^2 - 10 \cos(2\pi x_i)) + 200, [-2, 2]^{20}$$

$$f_5(x) = \sum_{i=1}^{20} -x_i \sin(\sqrt{|x_i|}), [-500, 500]^{20}$$

$$f_6(x) = \sum_{i=1}^{10} \frac{x_i^2}{4000} - \prod_{i=1}^{10} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, [-500, 500]^{10}$$

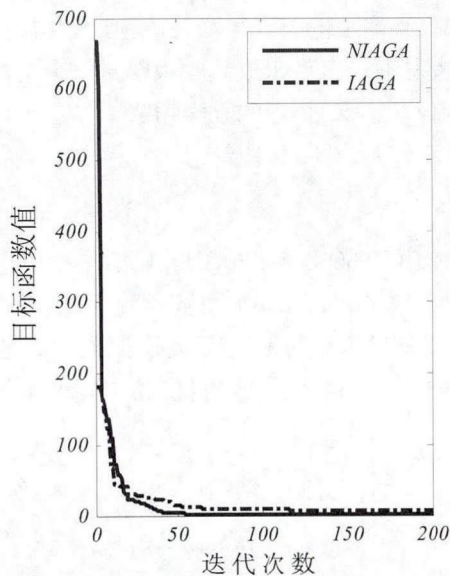
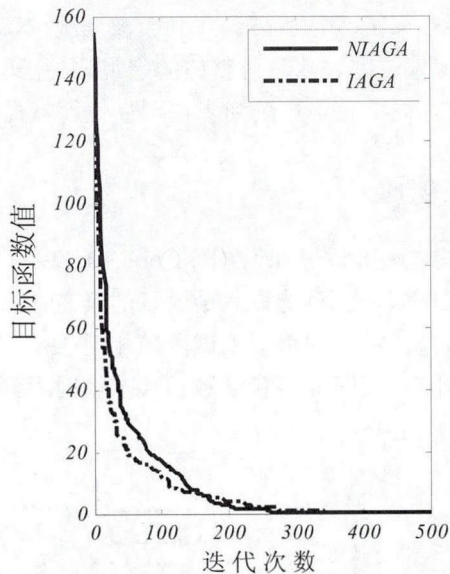
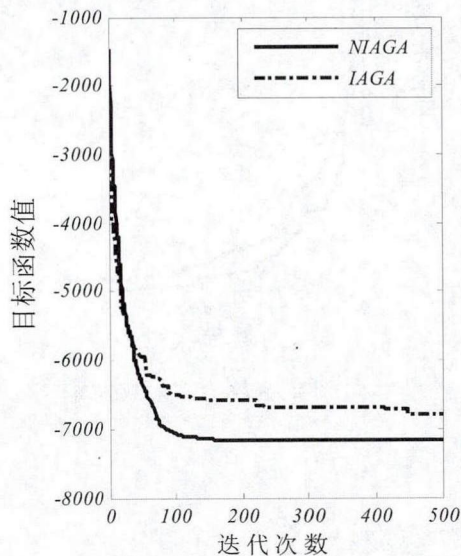
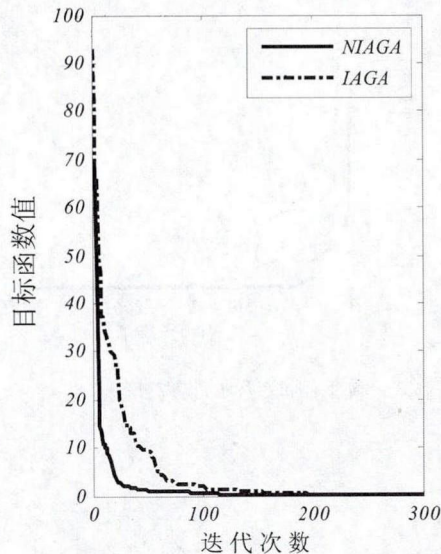
图3 函数 f_3 的收敛曲线图4 函数 f_4 的收敛曲线图5 函数 f_5 的收敛曲线图6 函数 f_6 的收敛曲线

表 1 IAGA、NIAGA 计算统计结果

| 函数 | 算法 | 最优值 | 最差值 | 平均值 |
|----------|-------|--------------|--------------|--------------|
| $f_1(x)$ | IAGA | 7.5317e-008 | 1.3287e-005 | 4.7851e-006 |
| | NIAGA | 1.0594e-010 | 2.3529e-008 | 3.0826e-010 |
| | 原始 | 0 | | |
| $f_2(x)$ | IAGA | 0.0731 | 0.3185 | 0.2551 |
| | NIAGA | 0.0036 | 0.0293 | 0.0069 |
| | 原始 | 0 | | |
| $f_3(x)$ | IAGA | 5.5861 | 7.9734 | 7.7207 |
| | NIAGA | 0.2845 | 10.7341 | 0.5169 |
| | 原始 | 0 | | |
| $f_4(x)$ | IAGA | 7.4646e-004 | 0.9963 | 0.0013 |
| | NIAGA | 1.3457e-004 | 4.0374e-004 | 1.4468e-004 |
| | 原始 | 0 | | |
| $f_5(x)$ | IAGA | -6.9097e+003 | -6.5225e+003 | -6.7984e+003 |
| | NIAGA | -7.6053e+003 | -5.2599e+003 | -7.1055e+003 |
| | 原始 | -7.6059e+003 | | |
| $f_6(x)$ | IAGA | 0.1035 | 0.4481 | 0.1504 |
| | NIAGA | 0.0360 | 0.3965 | 0.2772 |
| | 原始 | 0 | | |

由图 1 至图 6 可知, NIAGA 算法要比 IAGA 算法收敛得快. 由表 1 可知在最优值方面, NIAGA 算法均优于 IAGA 算法; 在最差值方面, 对于 $f_3(x)$ 和 $f_5(x)$, IAGA 算法优于 NIAGA 算法, 对于其他函数, NIAGA 算法均优于 IAGA 算法; 在平均值方面, 对于 $f_6(x)$, IAGA 算法优于 NIAGA 算法, 对于其他函数, NIAGA 算法均优于 IAGA 算法. 从而可以看出, NIAGA 算法所得到的解整体上要优于 IAGA 算法所得到的解好, 尤其在最优值方面, 说明 NIAGA 算法有着更强的寻优能力, 更能够跳出局部最优解, 摆脱未成熟收敛. 而且 NIAGA 算法所得到的最优值与测试函数的原始标准最优值十分接近, 在结果上具有可信性. 因此, 该算法是有效的, 可行的.

5 结束语

本文在 IAGA 算法的基础上提出了 NIAGA 算法, NIAGA 算法运用宏观与微观相结合的思想来设置交叉概率 p_c 和变异概率 p_m , 通过仿真实验证明, NIAGA 算法在未成熟收敛问题上, 有效地改进了 IAGA 算法, 显示出了更好的全局收敛性. 但如何彻底克服未成熟收敛是遗传算法最突出的一个问题, 要想从根本上解决还有待研究发现.

参考文献

[1] Holland J H. Adaptation in Natural And Artificial Systems[M]. Cambridge: M IT Press, 1975: 211-

247.

- [2] 边霞, 米良. 遗传算法理论及其应用研究进展 [J]. 计算机应用研究, 2010, 27(7): 2425-2429.
- [3] 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: 1-3
- [4] 鲍文胜, 刘晓刚. 基于自适应遗传算法的 RBF 神经网络优化算法 [J]. 山东师范大学学报, 2007, 22(3): 37-39.
- [5] 王莉. 自适应遗传算法在 PID 控制器设计中的应用西安科技大学学报 [J]. 2005, 25(1): 93-95.
- [6] 周明, 孙树栋. 遗传算法原理及应用 [M]. 北京: 国防工业出版社, 2001: 1-3.
- [7] Chen G, Wei Q. Fuzzy association rules and the extended mining algorithms[J]. Information Sciences, 2002, 147(1): 201-228.
- [8] Shelokar P S, Siarry P, Jayaraman V K, et al. Particle swarm and antcolony algorithm shybridized for improved continuous optimization[J]. Applied Mathematics and Computation, 2007, 188(1): 129-142.

An Improved Adaptive Genetic Algorithm

YU Guang-shuai, YU Xian-wei

(Collage of Mathematics and Physics, Bohai University, Jinzhou 121013, China)

Abstract: In view of the problem of the premature convergence of the IAGA adaptive genetic algorithm, an improved adaptive genetic algorithm (NIAGA algorithm) is proposed. The custom discriminant determines whether a group appeared a trend of the premature convergence, then both macroeconomic regulation and control and micro processing method are used to set the crossover probability P_c and mutation probability P_m respectively by the different situation and reduce the possibility of trapping in the premature convergence. The simulation result shows that the new algorithm improves the problem of the premature convergence of the IAGA algorithm effectively and has better global convergence.

Keywords: adaptive genetic algorithm; crossover probability; mutation probability; convergence