

带密度加权的自适应遗传算法^①

聂文亮, 蔡黎, 邱刚, 李春莉

(重庆三峡学院 信号与信息处理重点实验室, 重庆 404000)

摘要: 为了改善传统自适应遗传算法收敛速度慢、易陷入局部最优解的情况, 提出了带密度加权的自适应遗传算法. 该算法基于种群的分布密度, 动态调整遗传算法的交叉概率和变异概率, 并且在算法中使用了保留最佳个体法. 实验结果表明: 该算法在破坏种群局部稳定性、跳出局部极值的同时, 又能以较快的速度收敛于全局最优, 提高了算法的实用性和鲁棒性.

关键词: 遗传算法; 自适应; 密度加权

引用格式: 聂文亮, 蔡黎, 邱刚, 李春莉. 带密度加权的自适应遗传算法. 计算机系统应用, 2018, 27(1): 137-142. <http://www.c-s-a.org.cn/1003-3254/6133.html>

Adaptive Genetic Algorithm with Density Weighted

NIE Wen-Liang, CAI Li, QIU Gang, LI Chun-Li

(Signal and Information Processing Key Lab, Chongqing Three Gorges University, Chongqing 404000, China)

Abstract: The traditional adaptive genetic algorithm is slow in convergence and easy to fall into the local optimal solution. In order to resolve this problem, an adaptive genetic algorithm with density weighted is put forward in this study. Based on distribution density of population, this new algorithm can dynamically change the crossover probability and mutation probability of genetic algorithm, and combine with the best individual method. The results of the experiment show that the new algorithm can change the stability of local population, speed up the convergence, and improve its robustness and application.

Key words: genetic algorithm; adaptive; density weighted

遗传算法 (Genetic Algorithm, GA) 是由美国 Michigan 大学的 Hoiand 教授于 1975 年首先提出, 是一种借鉴生物界自然选择和自然遗传机制的随机化搜索算法^[1]. 由于遗传算法通用性强, 被广泛应用于非线性、多目标、多变量、复杂的自适应系统中.

但是传统的遗传算法收敛速度慢、容易陷入局部最优, 尤其在计算复杂问题时无法求出最优解, 所以国内外学者对其进行了改进. 目前, 对遗传算法的改进主要体现在三个方面: 1) 对遗传算子进行动态调整, 最早由 Srinvas 等人提出^[2,3]; 2) 对遗传算法的个体进行优化, 比如在文献^[4]中提出个体可进化的适应度函数评

价机制; 3) 采用 1) 和 2) 结合的方法改进遗传算法, 比如文献^[5]提出的带基因修复的自适应遗传算法.

虽然近年来有大量的改进遗传算法被提出, 但是课题组在研究遗传算法时发现, 如果能够根据种群个体的分布情况动态调整遗传算子, 可以增强种群个体的多样性, 同时有利于加快算法收敛. 因此, 本文提出了一种带密度加权的自适应遗传算法—DWAGA (Adaptive Genetic Algorithm with Density Weighted), 该算法可以根据种群分布情况动态调整交叉和变异概率, 使得算法具有跳出局部极大值加快收敛速度的能力, 同时本文还对该算法的改进过程、方法进行了详细的

① 基金项目: 重庆市教委基金 (KJ1601022); 重庆高校创新团队建设计划 (CXTDX201601034)

收稿时间: 2017-03-28; 修改时间: 2017-04-20; 采用时间: 2017-04-24; csa 在线出版时间: 2017-12-22

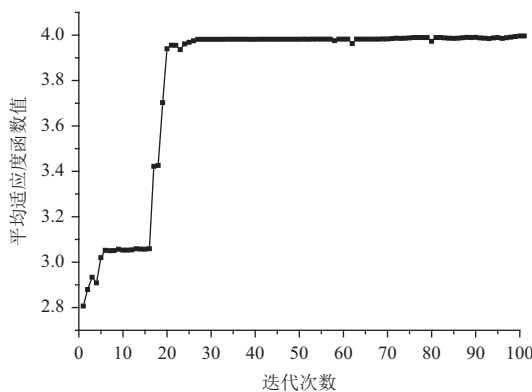
分析,最后通过求解三个函数的最优解验证了算法的有效性.

1 问题提出

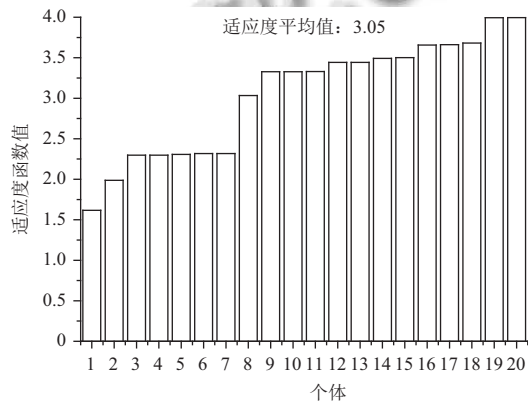
采用传统自适应遗传算法对如公式(1)所示函数进行最优求解,并对其个体分布情况进行分析.

$$y = -(x-1)^2 + 4; x \in [-1, 3] \quad (1)$$

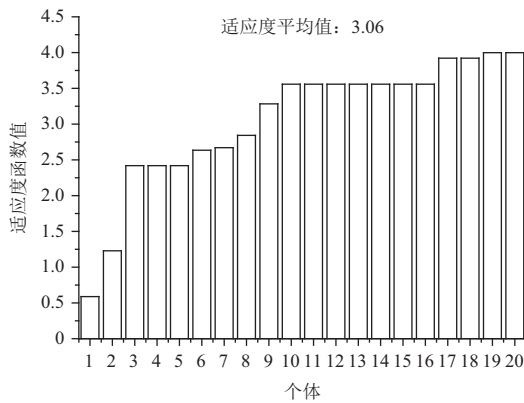
从图1可以看出,遗传算法在第5代到第18代,



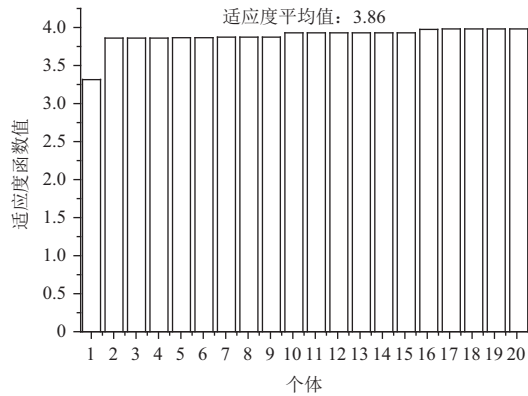
(a) 每代平均适应度函数值分布图



(b) 第5代种群个体适应度分布图



(c) 第16代种群个体适应度分布图



(d) 第50代种群个体适应度分布图

图1 种群个体适应度分布图

2 改进的遗传算法——DWAGA

2.1 确定种群密度

种群初始化后,对其个体按其适应度值由小到大的顺序进行排序,如图2所示.



图2 种群适应度值分布图

以及第25代-第95代之间停留了较长的时间,虽然也进行交叉、变异等遗传操作,但是由于种群过于集中于某一区域,导致个体已近似相当,即使进行了遗传操作,但是变化不大.尤其在算法迭代后期,特别容易陷入局部最优解.

因此,本文提出一种可以根据种群分布特点动态调整遗传算子的自适应遗传算法,该算法旨在破坏种群的局部稳定性,保持种群个体的多样性,同时加快算法的收敛速度.

图2中, M 表示种群规模, f_{\max} 表示当代种群中个体的最大适应度值, f_{\min} 表示当代种群中个体的最小适应度值, f_{avg} 表示当代种群个体的平均适应度值. 为了便于进行下一步的描述,做如下定义.

群体范围:

$$H = f_{\max} - f_{\min}$$

某个体到适应度平均值处的距离 (f 表示要变异个体的适应度值):

$$h = |f - f_{avg}|$$

中心区域半径:

$$\delta = \frac{H}{M-1}$$

到适应度平均值处距离小于 δ 的种群个数:

$$N(\delta, f_{avg})$$

种群中心区域密度:

$$\rho = \frac{N(\delta, f_{avg})}{M}$$

2.2 自适应参数调整

从模式理论^[1]可知,当选择策略确定后影响遗传算法收敛性的主要因素是由遗传算子来决定的. Srinvas 首先提出了自适应遗传算法^[2],其思想是当种群个体适应度值低于种群平均适应度值时,说明个体性能较差,增大交叉概率 P_c 和变异概率 P_m ,使得该个体被破坏或者淘汰;相反,当种群个体适应度值高于种群平均适应度值时,说明种群个体优良,相应地减小 P_c 和 P_m ,使该个体得以保存到下一代中.其中计算交叉概率和变异概率的公式如下:

$$P_c = \begin{cases} P_c \frac{(f_{max} - f')}{f_{max} - f_{avg}}, (f' \geq f_{avg}) \\ P_c, (f' < f_{avg}) \end{cases} \quad (2)$$

$$P_m = \begin{cases} P_m \frac{(f_{max} - f')}{f_{max} - f_{avg}}, (f' \geq f_{avg}) \\ P_m, (f' < f_{avg}) \end{cases} \quad (3)$$

式中, f_{max} 表示当代种群中个体的最大适应度值, f_{min} 表示当代种群中个体的最小适应度值, f_{avg} 表示当代种群个体的平均适应度值, f' 表示要变异个体的适应度值.

从公式 (2) 和公式 (3) 中看出,当种群适应度值等

$$P_c = \begin{cases} P_{cmax} \cdot A\rho, (h \leq \delta) \\ P_{cmax} - \frac{(P_{cmax} - P_{cmin})(f_{max} - f')}{f_{max} - f_{avg} + \alpha} \cdot A\rho, (h > \delta, f' < f_{avg}) \\ P_{cmax} - \frac{(P_{cmax} - P_{cmin})(f_{max} - f')}{f_{max} - f_{avg} + \alpha} \cdot A(1-\rho), (h > \delta, f' \geq f_{avg}) \end{cases} \quad (5)$$

式 (4) 中 A 为常数,其值大小反映了对种群中心区域密度的重视程度. α 为极小的常数,防止分母为零导致的错误,通常取值为极小的数. $0 < P_{cmin} < P_{cmax} < 1$,本文中 $P_{cmax}=0.9, P_{cmin}=0.4$. ρ 为种群中心区域密度.其他符号的意义同前不变.

当 h 小于等于 δ 时,种群个体适应度分布不均匀,在

于最大适应度值时,交叉概率和变异概率的值为零,这样就过分地保留了种群在前期进化阶段适应度值最大的个体,使得种群进化过程缓慢,极易陷入局部最优解.基于这个原因,段玉倩等人对上述自适应遗传算法进行了改进^[3].

如图 3 所示,在改进的自适应遗传算法中,设置种群交叉概率的最大值和最小值,且最小值不等于零.在种群个体为最大适应度值时,仍可以以最小的交叉概率来进行操作,这样就使得种群个体不会处于一种停滞不变的状态.计算交叉概率的公式如下式 (4),变异概率计算公式类似,省略.

$$P_c = \begin{cases} P_{cmax} - (P_{cmax} - P_{cmin}) \frac{(f_{max} - f')}{f_{max} - f_{avg}}, (f' \geq f_{avg}) \\ P_{cmax}, (f' < f_{avg}) \end{cases} \quad (4)$$

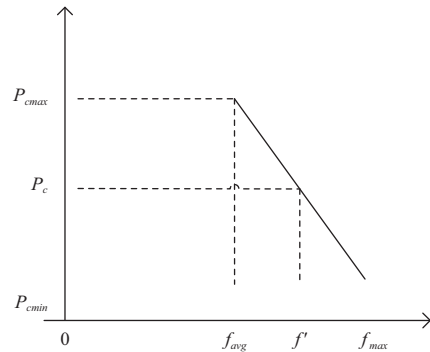


图 3 自适应遗传算法交叉概率取值图

公式 (3) 虽然改善了公式 (1) 和 (2) 中存在的不足,但仍缺乏对种群整体分布的分析,尤其是若种群个体过分集中于某一区域,则容易陷入局部最优解.因此,本文在上述自适应遗传算法的基础上^[3-5],提出带密度加权的自适应遗传算法.依据种群中心区域密度 ρ 对 P_c 作出修正,如公式 (5).

个体适应度平均值处(中心区域)集中,容易陷入局部最优解.因此,采用增大中心区域个体的交叉概率,破坏该区域个体的稳定性,使得该个体被淘汰或者在交叉过程中产生新的个体,以此实现跳出局部极值的能力.

当 h 大于 δ 时,种群个体适应度分布均匀,对于适应度低于平均适应度的个体,取较高的 P_c ,使该个体被

淘汰掉;而高于群体平均适应度的个体,取较低的 P_c 使个体得以保护进入下一代.

同理可以得到自适应遗传算法变异概率的计算公式,如公式(6).

$$P_m = \begin{cases} P_{m\max} \cdot A\rho, (h \leq \delta) \\ P_{m\max} - \frac{(P_{m\max} - P_{m\min})(f_{\max} - f)}{f_{\max} - f_{\text{avg}} + \alpha} \cdot A\rho, (h > \delta, f < f_{\text{avg}}) \\ P_{m\max} - \frac{(P_{m\max} - P_{m\min})(f_{\max} - f)}{f_{\max} - f_{\text{avg}} + \alpha} \cdot A(1 - \rho), (h > \delta, f \geq f_{\text{avg}}) \end{cases} \quad (6)$$

式中, $0 < P_{m\min} < P_{m\max} < 1$, 本文中 $P_{m\max} = 0.1, P_{m\min} = 0.01$. 其他符号的意义同前不变.

数值随自变量变化如图5所示.此函数将用以考查算法在全局最优点被局部最优解包围时从局部最优跳离的能力.

3 DWAGA 算法性能分析

3.1 评估函数^[6,7]

3.1.1 一元函数

函数表达式为:

$$F_1(x) = x \sin(10\pi x) + 2.0, \quad x \in [-1, 2] \quad (7)$$

该函数为一元多峰值函数,其函数值随自变量变化如图4所示.在其定义域内有全局最大值3.85,在整个定义域内,该函数还存在多个极大值,呈台阶式分布.该一元函数用来考查算法在存在多个极值时的搜寻能力.

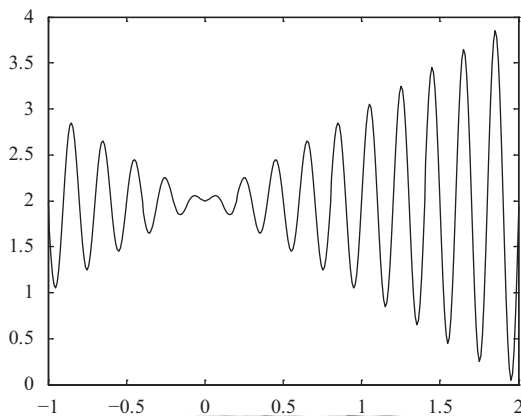


图4 一元函数图像

3.1.2 Schaffer 函数

函数表达式为:

$$F_2(x_1, x_2) = 0.5 - \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}, \quad x_1, x_2 \in [-10, 10] \quad (8)$$

该函数在其定义域内有最大值 $F_2(0, 0) = 1$,但是在极大点附近有由全局次优点 $F_2(x_1, x_2) = 0.99$ 形成的圈脊,如果算法局部搜索能力较弱,则极易收敛于次优点.其函

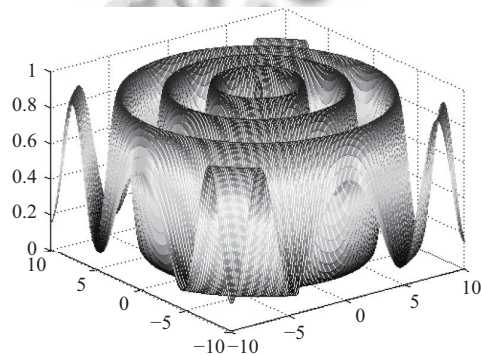


图5 Schaffer 函数图像

3.1.3 De Jone's 函数

函数表达式为:

$$F_3(x_1, x_2) = 0.002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6}, \quad x_i \in [-40, 40],$$

$$a_{ij} = \begin{bmatrix} -32, -16, 0, 16, 32, \dots, -32, -16, 0, 16, 32 \\ -32, -32, -32, -32, -32, \dots, 32, 32, 32, 32, 32 \end{bmatrix}_{2 \times 25} \quad (9)$$

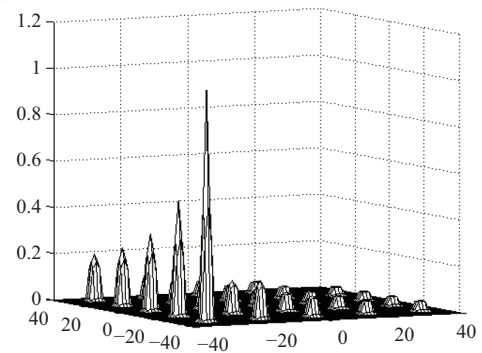


图6 De Jone's 函数图像

该函数为二维区域的多峰值函数,在其定义域内共有25个局部极大值,呈跳跃状分布在独立的区域内,极大值点自变量之间变化幅值大,极易使算法陷入局

部最大值点而停止进化,其函数值随自变量变化如下图6所示.其中,该函数的全局最大值点为 $F_3(-32, -32)=1$.该函数用来考查算法在存在多个极值的二维函数中的寻优能力.

3.2 实验结果与分析

为了验证算法的有效性,文中采用表1中的测试条件,对上述3个函数进行500次寻优计算,在收敛次数、平均收敛代数、以及最佳适应度值三方面对自适应遗传算法(Adaptive Genetic Algorithm, AGA)、改进的自适应遗传算法(Improve Adaptive Genetic Algorithm, IAGA),以及本文提出的带密度加权的自适应遗传算法(Adaptive Genetic Algorithm with Density Weighted, DWAGA)进行了统计和比较,详见表2,表3,表4.

表1 测试条件

	F_1	F_2	F_3
种群规模	30	100	300
编码方式	二进制	二进制	二进制
编码长度	16	18	20
选择操作	轮盘赌	轮盘赌	轮盘赌
交叉概率	0.5	0.4-0.9	0.4-0.9
变异概率	0.1	0.01-0.1	0.01-0.1
迭代次数	150	150	150
计算精度	10^{-4}	10^{-4}	10^{-4}

表2 F_1 收敛性能

算法	收敛次数	平均收敛代数	平均收敛值
AGA	457	38	3.8494
IAGA	463	27	3.8500
DWAGA	495	13	3.8501

表3 F_2 收敛性能

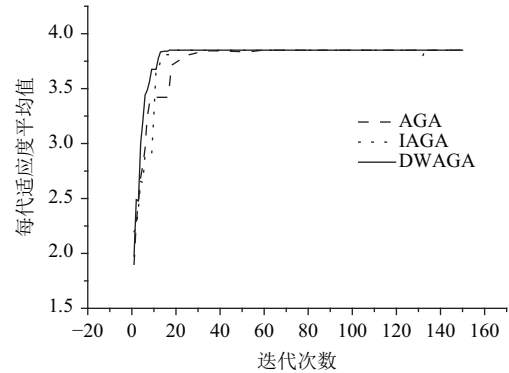
算法	收敛次数	平均收敛代数	平均收敛值
AGA	449	56	0.9989
IAGA	486	34	0.9997
DWAGA	497	25	0.9999

表4 F_3 收敛性能

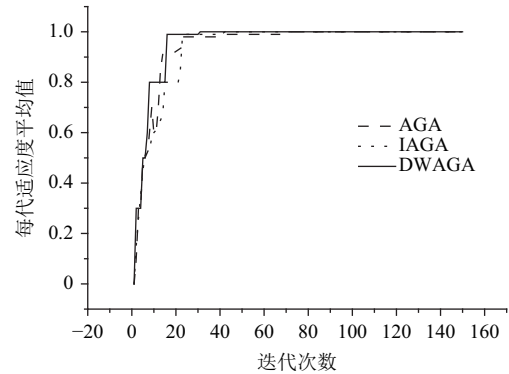
算法	收敛次数	平均收敛代数	平均收敛值
AGA	451	28	0.9992
IAGA	479	21	0.9999
DWAGA	499	17	1.0000

从表2、表3、表4的实验结果可以看出,在3个测试函数下,文中提出的DWAGA算法收敛总次数最多.同时,改进的自适应遗传算法收敛代数明显快于标准的自适应遗传算法,其中本文提出的DWAGA收敛速度最快.

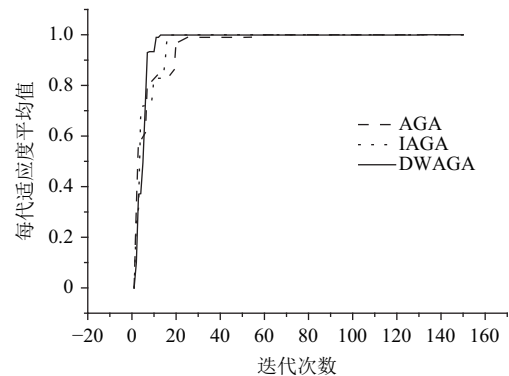
同时,为了分析AGA、IAGA、DWAGA这3种算法在整个迭代周期上的性能,文中分别列出了在不同测试函数下3种算法的适应度平均值分布图,如图7所示.



(a) F_1 函数 AGA、IAGA、DWAGA 每代适应度平均值曲线



(b) F_2 函数 AGA、IAGA、DWAGA 每代适应度平均值曲线



(c) F_3 函数 AGA、IAGA、DWAGA 每代适应度平均值曲线

图7 AGA、IAGA、DWAGA 算法在三种测试函数下每代适应度平均值曲线图

从图7中可以看出,AGA算法在迭代中期长期保持不变,IAGA和DWAGA算法都得到了改善,其中DWAGA算法改善最为明显,收敛速度最快.同时,文中三个测试函数的适应度函数就是函数的表达式,因

此都是求解相应函数的全局最大值点,通过图7的比较,AGA和IAGA算法都多次停留在局部极值处,但是本文提出的DWAGA算法能够快速跳出局部极值,向最大值点处收敛,有效地提高了算法的收敛速度。

4 结论

通过对传统的自适应遗传算法进行了深入研究,针对其收敛速度慢、难以跳出局部极大值的情况,本文提出DWAGA算法进行改善,通过对三个函数求解最优解的实验,表明该算法在收敛速度、平均收敛次数,以及全局最大值的搜寻能力上都明显优于传统自适应遗传算法,从而验证该算法的有效性和鲁棒性。

参考文献

- 1 Holland JH. *Adaptation in natural and artificial system: An introductory analysis with applications to biology, control and artificial intelligence*. Cambridge, MA: MIT Press, 1992.
- 2 Srinivas M, Patnaik LM. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 1994, 24(4): 656–667. [doi: 10.1109/21.286385]
- 3 段玉倩, 贺家李. 遗传算法及其改进. *电力系统及其自动化学报*, 1998, 10(1): 39–52.
- 4 林明玉, 黎明, 周琳霞. 基于可进化性的自适应遗传算法. *计算机工程*, 2010, 36(20): 173–175. [doi: 10.3969/j.issn.1000-3428.2010.20.061]
- 5 刘冀成, 胡雅毅. 带基因修复策略的自适应遗传算法. *计算机应用*, 2006, 26(6): 1401–1402, 1405.
- 6 Wang SH, Lu ZY, Wei L, *et al.* Fitness-scaling adaptive genetic algorithm with local search for solving the multiple depot vehicle routing problem. *Simulation*, 2016, 92(7): 601–616. [doi: 10.1177/0037549715603481]
- 7 Deng Y, Liu Y, Zhou DY. An improved genetic algorithm with initial population strategy for symmetric TSP. *Mathematical Problems in Engineering*, 2015, 2015: 212749.
- 8 曲志坚, 张先伟, 曹雁锋, 等. 基于自适应机制的遗传算法研究. *计算机应用研究*, 2015, 32(11): 3222–3225, 3229. [doi: 10.3969/j.issn.1001-3695.2015.11.004]