

一种新的改进遗传算法及其性能分析

徐 璐, 涂承宇

(北京工业大学电子工程系, 北京 100022)

摘 要: 本文提出了一种改进遗传算法(IGA), 通过加入混合选择、小范围竞争择优的交叉变异操作, 以及过滤相似个体, 并动态补充新个体的操作, 提高全局搜索性能和收敛速度。测试试验表明了算法性能优于许多现有算法。

关键词: 遗传算法; 收敛速度; 过滤; 竞争择优

中图分类号: TP418 **文献标识码:** A **文章编号:** 0372-2112 (2001) 07-0902-03

A New Improved Genetic Algorithm and Its Property Analysis

XU lu, TU Cheng-yu

(Dept. of Electrical Engineering, Beijing Polytechnic University, Beijing 100022, China)

Abstract: IGA (Improved Genetic Algorithm), a new genetic algorithm, is proposed in this paper. With five additional steps which are hybrid selection operator, competition among brothers, greedy operation, filter operation, and dynamic supply new individuals, IGA can improve global convergence property and convergence speed. The experimental results show IGA has great advantage over many existing genetic algorithms.

Key words: genetic algorithm, convergence speed, filter operation, competition among brothers

1 引言

近年来, 遗传算法的研究成为热点, 广泛应用于许多领域如神经网络、模式识别、自动控制等。但是该算法中全局搜索性能和收敛速度影响了遗传算法的应用。现有的一些改进算法大多只局限于 p_c 、 p_m (交叉、变异概率) 的改进, 忽略了算法整体设计。但是从 De Jong 的分析^[1]中可以看出, 算法的搜索性能、收敛速度不仅与 p_c 、 p_m 有关, 而且与群体规模以及遗传操作算子的设计、群体生成策略等都有关。因此本文提出一种新的改进算法(Improved GA, IGA), 从算法整体设计入手, 加入混合选择、小范围竞争择优、局部寻优的贪心算子、过滤相似个体、动态补充新子代个体等新操作, 兼顾全局搜索性能和遗传算法的收敛速度。试验结果表明, 这些新操作使改进算法(IGA)的性能大大提高, 明显优于许多现有的方法。

2 改进算法(Improved GA, IGA)

(1) 混合选择 本文提出将赌轮选择和排序选择结合的混合选择机制。方法是, 首先将个体的适应度从大到小进行排序, 然后执行赌轮选择。这样, 既有利于进化后期保持群体多样性, 又避免选择概率与适应度偏离的问题。

(2) 小范围竞争择优的交叉、变异操作 本文从加快收敛速度、全局搜索性能两方面考虑, 受自然界中家庭中兄弟间竞争现象的启发, 加入小范围竞争、择优操作。方法是, 将某一对父母 A、B 进行 n 次(3~5次)交叉、变异操作, 生成 $2n$ 个

不同的个体 ab_1 、 ab_2 、…… ab_{2n} , 选出其中一个最高适应度的个体 ab^* 送入子代对列中。反复随机选择父母对, 直到生成设定个数的子代个体为止。这种方法实质是在相同父母的情况下, 预先加入兄弟间的小范围的竞争择优机制。

另一方面, 在 SGA 算法中, 一对父母 X 、 Y 经遗传操作后产生一对子代个体 xy_1 、 xy_2 , xy_1 、 xy_2 随后都被放入子代对列。当进行新一轮遗传操作时, xy_1 、 xy_2 可能作为新的父母对进行交叉配对, 即“近亲繁殖”。而加入小范围竞争择优机制后, 每一对父母个体的子代只保留一个个体生存, 减小了在下一代中出现这一问题的几率。

(3) 局部寻优的贪心操作 局部寻优操作能进一步提高优化质量和搜索效率。因此, 本文提出将局部寻优的随机方法与改进算法结合。方法是, 对每一代的适应度值最大的前几个个体, 进行该个体邻域范围内进行数次随机搜索, 如果找到更优值, 则替代原值; 否则, 无变化。

(4) 过滤相似个体操作 未成熟收敛是遗传算法中不可忽视的现象, 主要表现为群体中所有的个体都陷于同一极值而停止进化。目前, 多种改进方法集中于对交叉、变异概率的自适应改变。变异操作可增加新的搜索空间, 以利于找到全局最优解; 但是增加变异会影响收敛的速度。因此, 本算法为加快收敛速度, 不采用改变变异概率的方法增加群体的多样性, 而采用滤除相似个体操作, 减少基因的单一性。

删除相似个体的过滤操作为: 对子代个体按适应度值排

序,依次计算适应度差值小于阈值 δ 的相近个体间的广义海明距离(相同长度的以 a 为基的两个字符串中对应位不相同的数量称为两者间的广义海明距离)。如果同时满足适应度差值小于阈值 δ ,广义海明距离小于阈值 d ,就滤除其中适应度值较小的个体。 δ 、 d 应适当选取,以提高群体的多样性。阈值过大可能将优秀个体滤除,过小可能起不到过滤作用。

(5) 动态补充新子代个体 过滤操作后,需要引入新个体。从实验测试中发现,如果采用直接随机生成的方式产生新个体,适应度值都太低,而且对算法的全局搜索性能增加并不显著(例如,对于复杂的多峰函数很难跳出局部最优值)。

因此,提出从优秀的父代个体中变异产生的方法。该法将父代中适应度较高的 m 个个体随机进行若干次变异,产生出新个体,加入子代对列。这些新个体继承了父代较优个体的模式片断,并产生新的模式,易于与其他个体结合生成新的较优子代个体。而且增加的新个体的个数与过滤操作删除的数量有关。如果群体基因单一性增加,则被滤除的相似个体数目增加,补充的新个体数目随之增加;反之,则只少量滤除相似个体,甚至不滤除,补充的新个体数目也随之减少。这样,动态的解决了群体由于缺乏多样性而陷入局部解的问题。

除了以上 5 个步骤之外,改进算法还加入了保留多个最优父代 $N(1-G)$ 个个体的群体构成策略。算法的执行步骤如下:

- (1) 确定控制参数及代沟 G ,初始化群体,群体规模为 N ;
- (2) 判断是否满足收敛条件,满足则输出结果,退出;否则继续;
- (3) 父代群体适应度排序,保留父代群体中最佳的 $N(1-G)$ 个体;
- (4) 混合选择、小范围竞争择优的交叉变异操作,直到子代个数达到设定数目 NG ;
- (5) 对新子代群体进行适应度排序;
- (6) 局部寻优的贪心操作,如果找到更优值,则替代原值;否则,无变化。
- (7) 过滤操作:滤除相似个体对中较差的个体,剩下 l 个个体;
- (8) 若 $l = N$,转入步骤 2;否则动态补充新子代个体,生成 $N-l$ 个新个体放入子代,转入步骤 2。

3 改进算法试验结果及分析

由于遗传算法中大量的随机操作,通常采用一些典型测试函数来检测算法的实际效率。本文选取了 4 个具有代表性的测试函数,对改进算法进行测试。

(1) $F_1(x) = \exp(-0.001x) \cos^2(0.8x) \quad x > 0$ (1)
全局最大值为 $f_m = 1.0$, $x_m = 0$ 。

(2) $F_2(x, y) = -(x^2 + 2y^2 - 0.3 \cos(3x) - 0.4 \cos(4y)) + 4x, y \in [-1, 1]$ (2)
此函数有多个局部极大值,其中全局最大点为 $(0, 0)$,最大值为 4.7。

(3) Camel 函数 F_3

$$F_3(x, y) = (4 - 2.1x^2 + \frac{x^4}{3})x^2 + xy + (-4 + 4y^2)y^2 \quad (3)$$

此函数有 6 个局部极小点,其中有两个 $(-0.0898, 0.7126)$ 和 $(0.0898, -0.7126)$ 为全局极小,最小值为 -1.031628 , $x, y \in [-10, 10]$

(4) Schaffer's 函数 F_4

$$F_4(x, y) = 0.5 - \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{[1 + 0.001(x^2 + y^2)]^2} \quad (4)$$

$x, y \in [-100, 100]$

此函数有无数局部极值点。只有一个全局最大点 $(0, 0)$,最大值为 1。全局极大值附近的次极值点为 0.990283,是最大值周围的一圈脊,极易使算法陷入局部极值。

本文采用 $F_1 \sim F_4$ 函数将改进算法(IGA)与 SGA、保留最佳个体法(OMSGA)的性能进行比较。在编程实现时,对其中的参数(如阈值 δ , d 等)采用多次测量取经验值的方法。试验结果如下(码长 20,群体规模 100,进化代数 100,交叉概率 $p_c = 0.9$,变异概率 $p_m = 0.1$):

3.1 算法收敛频率

不同算法独立运行 100 次的平均结果见表 1。容差区间 $[f^* - 0.00025, f^* + 0.00025]$ (f^* 为全局极值)。

表 1 SGA、OMSGA、IGA 函数 $F_1 \sim F_4$ 收敛频率比较(单位:次)

函数 \ 算法	SGA	OMSGA	IGA
F_1	26	100	100
F_2	0	35	100
F_3	0	6	100
F_4	0	0	99

从表中可见,改进算法对于 $F_1 \sim F_3$,特别是函数 F_4 的收敛效果明显优于 SGA、OMSGA 以及文献[3]的全局收敛占 88%,文献[4]的全局收敛占 59~66%的效果。可见改进算法能很好地解决复杂的多峰函数寻优问题,避免陷入局部最优,收敛频率远远优于其他算法。

3.2 算法收敛速度

不同算法在相同容差区间条件下(同上),独立执行 100 次函数 $F_1 \sim F_4$ 的收敛速度、最好解见表 2。

表 2 SGA、OMSGA、改进算法(IGA)函数 $F_1 \sim F_4$ 收敛速度比较(单位:秒)

函数	最大值 (最小值)	SGA		OMSGA		改进算法(IGA)	
		平均收敛时间	最好解	平均收敛时间	最好解	平均收敛时间	最好解
F_1	1	0.2	0.999981	0.05	1.000000	0.12	1.000000
F_2	4.7	#	#	3.1	4.699999	0.38	4.700000
F_3	-1.031628	#	#	14.8	-1.031625	0.62	-1.031628
F_4	1	#	#	#	#	4.95	0.999999

表中 # 表示进化 500 代后该算法仍无法收敛。

从表中可见,对于简单函数 F_1 ,IGA 的收敛时间多于 OMSGA,但是对于复杂函数 $F_2 \sim F_4$,IGA 的收敛速度和精度远远优于 OMSGA、SGA。虽然执行一次 IGA 迭代比 SGA、OMSGA 复杂,但是 IGA 达到收敛所需的进化代数远远少于

SGA、OMSGA, 因此收敛速度明显加快. 而且, 越是复杂的函数, 改进算法收敛速度快的优势越明显.

3.3 在线性能、离线性能

通过对 $F_1 \sim F_4$ 函数的测试, 改进算法显示了很好的在线性能和离线性能. 选取多峰函数 F_2 、全局最大值点附近具有无数局部极值的函数 F_4 的在线性能、离线性能测试曲线, 如图 1~2 所示. ($f_e(t)$ 选用平均适应度). 从图中可以看出改进算法的性能远远优于 SGA、OMSGA, 在进化的前几代就能迅速逼近最优值.

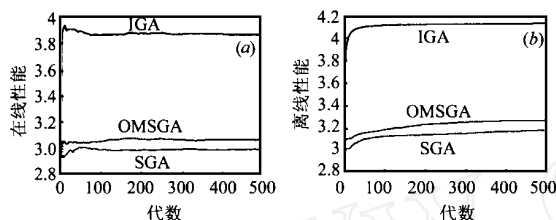


图 1 函数 F_2 在线性能 (a); 离线性能 (b)

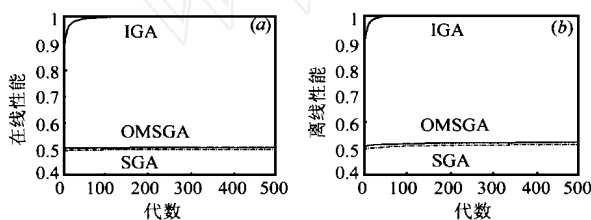


图 2 函数 F_4 在线性能 (a); 离线性能 (b)

4 分析及结论

从以上的试验结果可以看出, 在相同的控制参数下, 改进算法的收敛频率、收敛时间及精度等性能都远远优于 SGA、保留最佳个体法和许多现有算法. 而且越是复杂的函数, 算法的优势越明显.

综上所述, 由于改进算法将各个操作要素综合考虑, 使算法能在保持全局搜索特性的基础上, 大大加快收敛速度. 保持多个最佳个体的群体策略, 使优秀个体不易被破坏; 混合选

择、小范围竞争择优的遗传操作使算法的全局搜索性能大大增强; 局部寻优的贪心算子提高了优化质量; 过滤操作、动态补充新子代个体的操作使得当群体多样性降低的时候, 能加入新个体、新模式, 避免了陷入局部最优. 这些步骤的结合作用, 大大提高了算法的全局搜索性能和收敛速度, 性能明显优于现有的许多遗传算法, 并且可以应用于实时控制.

参考文献:

- [1] De Jong K A. An Analysis of the Behaviour of a Class of Genetic Adaptive Systems [M]. Univ. of Michigan, 1975.
- [2] 陈国良, 等. 遗传算法及其应用 [M]. 人民邮电出版社, 1996.
- [3] 任庆生, 叶中行. 进化计算的收敛速度 [J]. 上海交通大学报, 1999, 33(6): 671 - 673.
- [4] 雷德明. 自调整遗传算法 [J]. 系统工程与电子技术, 1999, 21(11): 70 - 71.
- [5] 李海民, 吴成柯. 自适应变异遗传算法及其性能分析 [J]. 电子学报, 27(5): 90 - 92.

作者简介:



徐璐 女. 1973 年生于北京. 1997 年获北京工业大学电子工程系学士学位, 同年就读北京工业大学电子工程系研究生至今. 研究方向为智能化信息处理、遗传算法及其应用.



涂承宇 男. 1960 年生于北京. 北京工业大学电子工程系副教授, 硕士. 1997 年至 1998 年日本名古屋大学访问学者. 主要研究方向包括智能化信息处理、人工智能技术等.