

文章编号: 1001-0920(2002)05-0625-04

# 一种防止浮点遗传算法早熟收敛的父代选择策略

谢晓锋, 张文俊, 杨之廉  
(清华大学 微电子学研究所, 北京 100084)

**摘 要:** 通过对浮点遗传算法早熟收敛现象的分析, 提出了一种新的父代选择策略, 即使用当前代的子代个体作为下代的父代个体, 可使交叉算子持续地探索和开发新空间。引入对个体的代数保护策略, 即在它发生变异前保证有足够的演化, 可以避免对新空间不成熟的开发。通过与其它父代选择策略的对比, 并通过实验和 GENOCOP 系统比较, 表明本方法能得到较好的结果。

**关键词:** 遗传算法; 早熟收敛; 选择策略

中图分类号: TP 301.6      文献标识码: A

## A parents selection strategy fighting premature convergence in floating genetic algorithms

XIE Xiaofeng, ZHANG Wenjun, YANG Zhilian  
(Institute of Microelectronics, Tsinghua University, Beijing 100084, China)

**Abstract:** With the analysis of the premature convergence of floating genetic algorithms (FGAs), a new selection strategy is proposed by using the children individuals of current generation as the parent of next generation, which makes the crossover operator to explore and exploit in new spaces successively. A protection strategy for the evolved generations for individuals is introduced to guarantee enough evolution before the mutation occur, which avoids immature exploitation for new spaces. Compared with the results of GENOCOP system by experiment, this selection strategy provides better solutions.

**Key words:** genetic algorithms; premature convergence; selection strategy

## 1 引 言

遗传算法(GAs)是由 Holland 等创立的<sup>1</sup>, 理论基础主要来源于生物进化论和群体遗传学。由于 GA 对函数的性质几乎不作要求, 又因 GA 采用群体搜索技术, 具有高度的鲁棒性和并行性, 因此具有广泛的应用范围。

从 De Jong<sup>2</sup> 起, GA 被用来解决具有连续变量的函数优化问题; 随后, 浮点遗传算法<sup>3,4</sup> 因其具有

无需进行复杂的编码解码的优点, 而被广泛用于解决工程问题。遗传算法在解决单峰函数优化时比较有效, 但对多峰复杂函数优化问题, 往往搜索停止在未成熟阶段, 即出现早熟收敛现象<sup>5</sup>。

本文研究浮点遗传算法发生早熟收敛的原因, 提出一种新的父代选择策略来防止早熟收敛, 使得演化更加深入; 在此基础上, 利用代数保护方法得到更好的收敛效果。通过与 GENOCOP<sup>6,7</sup> 算法的比较, 证明了改进策略的有效性。

## 2 浮点遗传算法

### 2.1 算法简介

浮点遗传算法即采用浮点编码, 每个个体对应于实数空间中的一个点。其实现过程为:

- 1) 初始化种群;
- 2) 选择父代进行交叉、变异产生子代个体;
- 3) 评价子代个体适应度, 并根据排序淘汰机制更新种群;
- 4) 代数加1并重复2), 直至满足终止条件。

算法参数主要有: 种群大小  $N$ , 每代子个体数  $m$ , 变异概率  $P_m$ , 选择压力  $q$ , 演化代数  $G_n$ 。

#### 2.1.1 选择算子

选择算子用来选择父代和淘汰个体, 目的是把个体复制到下一代, 或者通过配对交叉产生新的个体再遗传到下一代。为更好地控制个体被选择的概率, 通常使用排序选择机制<sup>8</sup>: 在每一代对所有个体按适应值由高到低排序, 第  $i$  个个体的生存概率为

$$\text{prob}(i) = q(1 - q)^{i-1} \quad (1)$$

其中  $q \in (0, 1)$  为选择压力。

对于父代选择, 采用排序机制选择生存概率较高的个体作为父代。代间替换时, 在保留上一代最好个体的前提下, 子代个体按代沟方式, 由排序机制按一定比例更新群体中的部分个体, 以保证全局收敛性。

#### 2.1.2 交叉算子

交叉算子用来提供不同个体的基因交换。对两个父代点  $x_1$  和  $x_2$ , 如果它们都不是局部优值, 则在适应值较好点的邻域一定存在适应度更好的个体, 因而可由邻域交叉产生后代

$$x_3 = r(x_2 - x_1) + x_2 \quad (2)$$

其中,  $r \in (-1, 1)$  且  $r \neq 0$ ,  $x_2$  的适应值不低于  $x_1$ 。算子在  $(-1, 0)$  时为算术交叉<sup>6</sup>, 在  $(0, 1)$  时为启发式交叉<sup>9</sup>。

#### 2.1.3 变异算子

遗传算法强调的是交叉功能, 解的进化主要靠选择机制和交叉策略来完成, 变异用来使群体摆脱进化陷入搜索空间的某个局部区域。这里采用的变异为在搜索空间中取随机点, 每代中的个体按给定的概率发生变异。

由于保留每代的最优个体, 以及取变异算子的随机点是遍历的, 可得出算法是全局收敛的。

### 2.2 早熟收敛的原因

在一定代数以后, 因为选择策略采用繁殖机会

与适应值成正比的方法, 在交叉算子和选择算子作用下, 很容易导致如下问题:

1) 超级个体问题: 在进化的初始阶段, 生成了具有很高适应度的个体, 经过少数几代迭代后, 群体中的位置可能全部被该个体及其后代占据;

2) 封闭竞争问题: 即适应度较高的一组个体位于较小的封闭空间内, 由于染色体差异很小, 交叉算子将降低其搜索作用, 不能引导搜索有效地进行, 从而难以找到优值。

以上问题会导致群体分布范围有限, 搜索有可能停止在未成熟阶段, 即出现早熟收敛。解决途径是扩展当前搜索区域, 引入收敛区域外的个体, 使之成为父代个体参与演化。

为此, 在遗传算法中引入变异算子, 以使搜索进程跳出封闭竞争。实际上, 如果当前封闭区域内的点适应度较高, 那么对于一个由变异算子得到的封闭空间外的点, 它具有更高适应度的概率会很小, 在根据适应度排序的选择机制作用下, 该个体几乎没有机会成为父代参与交叉操作。这样, 交叉操作仍然由封闭区域中的点来进行, 这时算法等同于利用变异算子产生的个体进行效率不高的随机搜索。

## 3 子代复用型父代选择机制

为防止早熟收敛, 需使封闭空间外个体(无论适应度高低)有较大机会成为父代, 这样才可在封闭空间外演化。此外, 子代的适应度也可能较差, 要使其继续在新产生的搜索空间开发, 需使子代也有较大机会成为父代。如此持续数代后, 可得到该搜索空间中的优值点。

### 3.1 子代复用型父代选择

本文提出一种子代复用型父代选择机制。在初始化种群后, 再初始化其中一组点为初始种子父代。在每代中, 对每个种子父代个体以一定概率  $P_m$  发生变异, 或与另一个基于排序选择机制得到的父代个体进行交叉操作, 产生的子代个体作为下代的种子父代个体进行下一代的演化。

这样, 即使当前群体都位于一个小的封闭空间, 由变异可将个体产生在空间外部, 由此产生新的搜索空间, 然后通过该个体及其后代的多代演化来开发该空间。

### 3.2 代数保护策略

对以上子代复用选择, 变异个体的持续演化进程往往被下一次对它的变异所中断。为此, 可对每个

新变异的个体保护一定代数  $G_p$ , 在这个代数后才允许对该个体进行下一次变异操作。这样, 每个变异后的个体可至少持续演化  $G_p$  代, 从而保证它的开发能力。

### 4 测试与研究

#### 4.1 父代选择策略

对交叉算子, 需要两个父代个体, 本文对其中一个仍采用基于适应度的排序选择机制, 对另外一个父代个体做如下几种选择策略实验:

- 1) 排序选择: 以上述基本遗传算法的选择策略用作结果对比;
- 2) 随机选择: 从群体中随机选择个体, 类似于排序选择中个体生存概率相同的情形;
- 3) 分裂选择<sup>10</sup>: 将群体适应度排序后, 选取其中较好的一部分及较差的一部分;
- 4) 子代复用选择。

#### 4.2 测试与分析

对如下非线性问题<sup>6</sup> 的测试结果进行对比:

$$\begin{aligned} \min G_1(x) = & (x_1 - 10)^2 + 5(x_2 - 12)^2 + \\ & x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + \\ & x_7^4 - 4x_6x_7 - 10x_6 - 8x_7 \\ \text{s.t. } & 127 - 2x_1^2 - 3x_2^4 - x_3 - 4x_4^2 - 5x_5 = 0 \\ & 282 - 7x_1 - 3x_2 - 10x_3^3 - x_4 + x_5 = 0 \\ & 196 - 23x_1 - x_2^2 - 6x_6^2 + 8x_7 = 0 \\ & -4x_1^2 - x_2^2 + 3x_1x_2 - 2x_3^3 - 5x_6 + 11x_7 = 0 \end{aligned}$$

其中,  $-10 \leq x_i \leq 10, i = 1, 2, \dots, 7$ 。

为观察优化效果, 我们使用  $F = \log(F_{\min} - F_{\text{opt}})$  来考察算法所得最好结果逼近全局优值的能力, 其中  $F_{\min}$  代表算法获得的最好结果,  $F_{\text{opt}}$  为已知的函数优值, 取为 680 630。参数为种群大小  $N = 100$ , 每代子个体数  $m = 20$ , 选择压力  $q = 0.1$ , 演化代数取 300 代, 运算 100 次取平均。

图 1 为各种选择策略在不同变异概率  $P_m$  下得到的  $F$  的变化曲线(其中  $P_m = 1$  时, 即为随机搜索算法)。

为易于理解, 这里假设当前群体的适应值较高, 且位于很小区域内, 即已经早熟收敛, 且假定变异后的点不在封闭区域内, 但适应度较低。

对于排序选择, 变异后的个体很难得到参与演化的机会, 等同于随机搜索, 其作用不大。

对于随机选择, 每个个体都可得到平等参与演

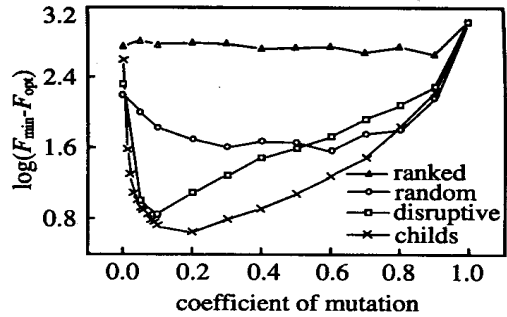


图 1 不同选择策略下  $F$  随  $P_m$  的变化曲线  
化的机会, 这意味着在封闭区域外的点也有机会参与交叉运算, 因而能搜索新的模式空间。但由于是随机选择, 并不能对新的模式进行更深入的开发。

对于分裂选择, 较差的部分个体可用来探索新空间, 而较好的部分个体可用来开发解空间。当变异概率较小时, 变异的个体不多, 与群体中的个体相比, 它们的适应度一般较差, 将作为适应度较差的一组参与交叉运算。通常是它们的父代被替换掉, 而这些子代将作为下一代的父代继续参与演化, 直到它们的子代达到或超过封闭区间的适应度, 这时的演化机制与子代复用选择相似。但当变异概率较大时, 在一些新的模式空间还没有被完全开发前, 由于更多变异新个体的加入, 开始开发另外的模式空间, 使得开发深度不够。

对于子代复用选择, 每个个体在它的下次变异发生前, 都可以对探索到的新空间中进行持续的邻域交叉搜索, 所以有能力开发出良好的解。

此外, 在对文献 6 的其它函数测试中, 同样观测到了类似的结果。

图 2 给出了子代复用选择在不同保护代数下的  $F$  随变异概率  $P_m$  的变化曲线。可以看到, 保护代数使得演化结果对变异概率  $P_m$  变得不敏感, 这也证明了对封闭区间外个体的持续演化是防止早熟收敛的主要原因, 而不是各变异个体本身。而且在合适的保

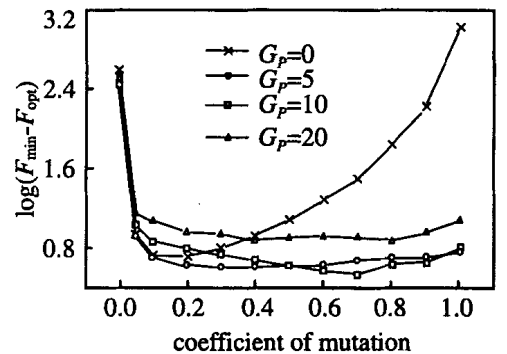


图 2 不同保护代数  $G_p$  下  $F$  随  $P_m$  的变化曲线

护代数如  $G_p$  为 5 或 10 时, 得到的结果都比不加代数保护时的结果要好。当然, 保护代数不能太高, 因为在保护代数内个体没有变异机会, 在一定演化代数后, 由于缺乏新的变异个体来探索和开发新空间而使优化能力下降。

### 4.3 与 GENOCOP<sup>6</sup> 的比较

GENOCOP 是 Michalewicz 教授等人针对带约束数值优化问题编写的, 它的独特之处在于有针对性地设计了 7 种遗传算子来进行共同演化。但 GENOCOP 的父代选择机制仍基于排序选择。

图 3 给出了对函数  $G_1(x)$  使用子代复用型父代选择的本文算法与 GENOCOP 在演化代数为 300 代时的收敛曲线的比较, 其中变异概率  $P_m = 0.1$ 。由图可见, 虽然 GENOCOP 早期的收敛速度比较快, 但在演化后期本文算法能更好地摆脱局部点而更深入地进化。

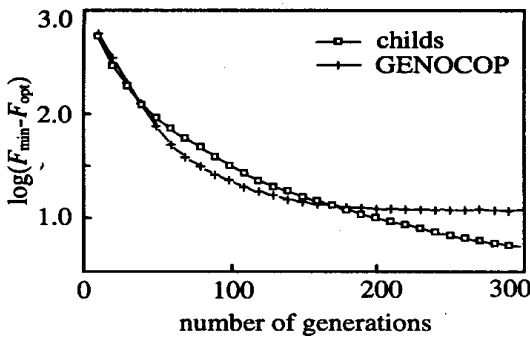


图 3 本文算法与 GENOCOP 的收敛性能比较  
利用如下函数与文献 7 结果进行对比

$$\begin{aligned}
 \text{min } G_2(x) = & x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + \\
 & (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + \\
 & 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + \\
 & 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45 \\
 \text{s.t. } & 105 - 4x_1 - 5x_2 + 3x_7 - 4x_8 = 0 \\
 & - 3(x_1 - 2)^2 - 4(x_2 - 3)^2 - \\
 & 2x_3^2 + 7x_4 + 120 = 0 \\
 & - 10x_1 + 8x_2 + 17x_7 - 2x_8 = 0 \\
 & - 2x_1^2 - 2(x_2 - 2)^2 + \\
 & 2x_1x_2 - 14x_5 + 6x_6 = 0 \\
 & 8x_1 - 2x_2 - 5x_9 + 2x_{10} + 12 = 0 \\
 & - 5x_1^2 - 8x_2 - (x_3 - 6)^2 + 2x_4 + 40 = 0 \\
 & 3x_1 - 6x_2 - 12(x_9 - 8)^2 + 7x_{10} = 0 \\
 & - 0.5(x_1 - 8)^2 - 2(x_2 - 4)^2 - \\
 & 3x_3^2 + x_6 + 30 = 0
 \end{aligned}$$

其中,  $-10 \leq x_i \leq 10, i = 1, 2, \dots, 10$ 。

问题有 3 个线性和 5 个非线性约束, 全局最小值为 24.3062091, 除最后两个外都在优化点处激活。

参数设置与文献 10 相同, 种群大小  $N = 70$ , 替换概率为 0.2, 即每代子个体数  $m = 14$ , 选择压力为 0.1, 演化代数为 4000 代, 保护代数设为 5 代, 变异率为 0.1。运算 10 次取平均值, 其测试结果比较如表 1 所示。

表 1 本文算法与 GENOCOP 的测试结果比较

algorithm	GENOCOP <sup>7,1</sup>	this GA
best solution	25.328	24.831
worst solution	32.019	29.345
average	28.943	27.056

## 5 结 论

本文研究了浮点遗传算法早熟收敛的原因, 在此基础上提出了防止早熟收敛的子代复用型父代选择机制。即将产生的子代个体作为下代的父代个体, 从而利用由变异产生位于封闭空间外的个体扩大搜索空间, 并利用多代的演化来深入地开发该空间。为防止演化进程被变异打断, 对每个新变异的个体保护一定代数  $G_p$ , 从而保证它的开发能力, 并在此方法下得到了更好的结果。本方法使得遗传算法能很好的跳出封闭空间, 有效地帮助群体持续的进化。

### 参考文献 (References):

- 1 J H Holland *A daptation in N atural and A rtificial S ystems* [M]. Ann Arbor: U niversity of M ichigan Press, 1975
- 2 K De Jong *The analysis of the behavior of a class of genetic adaptive system s* [D]. U niversity of M ichigan, 1975
- 3 C Z Janikow, Z Michalewicz *An experimental comparison of binary and floating representation in genetic algorithms* [A]. *Proc 4th ICGA* [C. Morgan Kaufman, 1991 31-36
- 4 Zeng-Ping Chen, Jian-Hui Jiang, Yang Li, et al *Non-linear mapping using real-valued genetic algorithm* [J]. *Chemometrics and Intelligent Laboratory Systems*, 1999, 45(1-2): 409-418
- 5 J Andre, P Siarry, T Dognon *An improvement of the standard genetic algorithm fighting premature convergence in continuous optimization* [J]. *Advances in Engineering Software*, 2001, (1): 49-60

---

(上接第 628 页)

- 6 Z Michalewicz, M Schoenauer Evolutionary algorithms for constrained parameter optimization problems [J]. *Evolutionary Computation*, 1996, 4(1): 1-32
  - 7 M Sakava, K Yauchi Coevolutionary genetic algorithms for nonconvex nonlinear programming problems: Revised GENOCOP III [J]. *Cybernetic and Systems*, 1998, 29(8): 885-899
  - 8 D Whitley The GENITOR algorithm and selective pressure: Why rank-based allocation of reproductive trials is best [A]. *Proc 3rd ICGA* [C]. Morgan Kaufman, 1989
  - 9 A Wright Genetic algorithms for real parameter optimization [A]. *Foundations of Genetic Algorithms* [C]. Morgan Kaufman, 1991. 205-218
  - 10 T Kuo, S-Y Hwang Using disruptive selection to maintain diversity in genetic algorithms [J]. *Applied Intelligence*, 1997, 7(3): 257-267.
-