

快速寻优的遗传交叉策略

陈小平¹, 石 玉², 于盛林²

(1. 苏州大学 通信与电子工程系, 苏州 215021; 2. 南京航空航天大学 测试工程系, 南京 210016)

摘要: 介绍了遗传算法中一种新的交叉策略, 它利用逼近方法决定遗传算法中子代个体的位置. 采用这种交叉策略可以使子代快速地向更高适应度的区域移动, 实验数据说明使用这种交叉策略的遗传算法可以极大地提高算法获得最优解的速度.

关键词: 遗传算法; 交叉策略; 逼近

中图分类号: TP301.6

文献标识码: A

Genetic crossover strategy to obtain optimal solutions quickly

CHEN Xiao-ping¹, SHI Yu², YU Sheng-lin²

(1. Department of Communication and Electronics Engineering, Suzhou University, Suzhou 215021, China;

2. Department of Measurement and Testing Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, China)

Abstract: This paper presents a crossover strategy that utilizes an approximation concept to steer the location of progeny in genetic algorithm. The use of such a crossover strategy provides the capacity of quickly moving progeny toward regions with improved fitness. Experiment results demonstrated that genetic algorithms with this crossover strategy could greatly accelerate the speed on obtaining optimal solutions.

Key words: genetic algorithms; crossover strategy; approximation

1 引言(Introduction)

遗传算法(genetic algorithm, GA)是 Darwin 生物进化理论的哲学思想启发的搜索技术. GA 的基本概念和理论框架, 是由 Holland 首先提出的^[1], 并由 Goldberg 作了广泛的探索. 模式定理证明了 GA 具有全局搜索的能力. GA 中的交叉类似于生物繁殖过程中基因成分的交换, 这种操作算子被认为是贡献于 GA 的全局搜索性能的最重要的因素, 尽管变异算子在一定程度上也有这种贡献. 基于概率的交叉是鲁棒的, 这是因为它不受特定问题的限制. 然而遗传算法的搜索方向在交叉算子中很少被强调. 为了快速获得高性能的候选解, 搜索方向应能指示出 GA 在一定领域中潜在的前进方向. 交叉算子中随机数的使用常常会引起候选解在搜索空间中处于随机的位置, 如果最优解所在区域距当前搜索区域很远又不能预先识别的话, 这就降低了发现最优解的速度, 尤其是对于多重优化的问题.

由于在交叉算子中常常忽略考虑如何给出指导性的搜索方向, 不能使子代群体向高质量区域快速

移动, GA 需要进化代数的增加就会不可避免地导致成本增高和时间浪费, 这被认为是 GA 的主要弱点^[2]. 为了使 GA 能够与其它优化方法竞争, 就必须改善 GA 的性能, 改进交叉算子就是改善 GA 性能的一种手段. 本文研究了利用逼近方法定位子代群体的交叉策略, 此策略强调了 GA 潜在的搜索方向使得子代群体能在此方向前进, 快速搜索到其他高质量的区域, 通过几个测试函数的实验例子以说明这种策略的有效性.

2 交叉算子中的逼近方法(The approximation method in crossover)

考虑具有连续参数的最大化问题

$$f_{\max}(x), x_i^L \leq x_i \leq x_i^U. \quad (1)$$

此处 x 是参数向量, x_i^L 和 x_i^U 分别是 x_i 的上界和下界. 在交叉操作中使用逼近方法有两个主要原因: 1) 估计潜在的搜索方向; 2) 充分利用两个交叉父体提供的遗传信息. 在没有先验知识的情况下, 这样的潜在搜索方向通常是不清楚的. 但是, 利用适应度 $f(x)$ 的值估计这种方向是可能的. 一般地, 交叉的两

个父体之间的适应度 $f(x)$ 是不相同的,两个父体之间适应度差异不仅在复制过程中有贡献,而且可能为生成更好质量的子代指示潜在的搜索方向.因此,利用适应度差异生成子代的过程可以增加遗传信息的利用,这种信息在普通的交叉算子中一般是被忽略的.

如果搜索方向可线性逼近,利用父体之间的适应度差,规范化搜索方向 d 可以表示为

$$d = (f_{\text{high}} - f_{\text{low}})/f_{\text{high}}. \quad (2)$$

此处 f_{high} 和 f_{low} 表示交叉对中两父体的适应度, $f_{\text{high}} > f_{\text{low}}$. 由于 GA 的工作原理是基于适应度的随机搜索,等式(2)是一合理的估计.如果一个个体沿着方向 d 从区域(f_{low})向另一个区域(f_{high})前进,该个体的适应度可以得到改善.这个估计搜索方向的几何表示如图1所示.

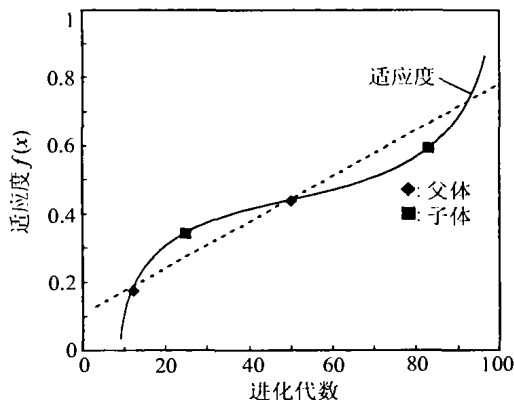


图1 搜索方向的线性逼近

Fig. 1 The linear approximation in searching direction

一些研究人员^[3,4]曾研究过用这种线性逼近在GA中建立指导的搜索方向,后代个体位于

$$x^{t+1} = x_{\text{low}}^t + \alpha(x_{\text{high}}^t - x_{\text{low}}^t)d. \quad (3)$$

此处 x^{t+1} 表示后代在 $t+1$ 代时位置, $\alpha \in [0,1]$ 是决定后代位置的一随机距离系数.通过在两个候选者之间连一条线,沿这条线随机地选择两个点生成后代个体(如图1所示)完成交叉操作.

然而,式(3)的交叉策略不允许后代个体处于 $[x_{\text{low}}^t, x_{\text{high}}^t]$ 区界之外.后代个体位置的这种限制没有最大化改善后代个体的基因模式的程度,限制了整个群体的搜索速度,并有未成熟收敛于局部最优的可能,后代若要跳出 $[x_{\text{low}}^t, x_{\text{high}}^t]$ 之界,只能借助于遗传算法中的变异操作.鉴于此,本文提出了一种新的交叉策略,选择合适的 α 值,最大化基因模式的改善.

本文提出的交叉策略如下:

$$\begin{cases} x_1^{t+1} = x_{\text{low}}^t + \alpha_1(x_{\text{high}}^t - x_{\text{low}}^t)d, \\ x_2^{t+1} = x_{\text{low}}^t + \alpha_2(x_{\text{high}}^t - x_{\text{low}}^t)d. \end{cases} \quad (4)$$

式中 α_1 是 $[0, 1]$ 上均匀分布的随机数, α_2 是 $[1, 2]$ 上均匀分布的随机数,这样 x_1^{t+1} 将处于 $[x_{\text{low}}^t, x_{\text{high}}^t]$ 界内, x_2^{t+1} 就有可能处于 $[x_{\text{low}}^t, x_{\text{high}}^t]$ 界外.式(4)有以下几个重要特点:

1) 引进了后代个体基因模式的线性改善,即一个弱的字符串(适应度为 f_{low} 的染色体)能向着一强的字符串(适应度为 f_{high} 的染色体)改善其基因类型.对于一固定的 α ,两父体之间的大的距离 $x_{\text{high}}^t - x_{\text{low}}^t$ 和大的适应度差 $f_{\text{high}} - f_{\text{low}}$ 对改善有大的贡献,这样在交叉期间弱字符串比强字符串有更多的改善;

2) 通过修改个体字符串的参数向量 x 搜索新染色体的能力,只要所有的个体没有相同的适应度,新的模式(它不出现在它们的父体中)就能产生,新模式的引进保持了群体中基因的多样性,这样避免了未成熟收敛于局部最优的可能;

3) 克服了后代个体位置的限制;

4) 本身具备变异的效果,可以省略变异算子.

3 测试函数及实验结果(Testing functions and experiment results)

本节将使用本文提出的逼近交叉策略的GA应用于一类测试函数,并与采用其它交叉策略的遗传算法进行了比较.选取3个具有相当复杂度的测试函数^[5],这些测试函数都包含多个已知的极大值.所选函数如下,

$$f_1(x, y) = 0.5 - \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{(1 + 0.001(x^2 + y^2))^2},$$

$$-10 < x, y < 10,$$

$$f_2(x, y) = \left(4 - 2.1x^2 + \frac{x^4}{3}\right)x^2 + xy + (-4 + 4y^2)y^2,$$

$$-10 < x, y < 10,$$

$$f_3(x, y) = -(x^2 + 2y^2 - 0.4\cos(3\pi x) - 0.6\cos(4\pi y)),$$

$$-10 < x, y < 10.$$

函数 f_1 有无数个局部极大点,但只有一个 $(0,0)$ 为全局最大点,最大值为1.此函数的最大峰周围有两圈脊,它们的取值分别为0.990284和0.962776,因此优化过程中很容易停滞在这些局部极大点.函数 f_2 有6个局部极小点,有两个点 $(-0.0898, 0.7126)$ 和 $(0.0898, -0.7126)$ 为全局最小点,最小值为 -1.031628 .函数 f_3 是一个多峰函数,但只有一个全局最大点 $(0,0)$,最大值为1.所用实验方法为:

M_1 ——二进制字符串编码遗传算法,交叉概率 $p_c = 0.9$,变异概率 $p_m = 0.01$,按比例选择并保留最优个体;

M_2 ——实数编码遗传算法,交叉概率 $p_c = 0.9$,变异概率 $p_m = 0.01$,按比例选择并保留最优个体;

M_3 ——采用本文提出的交叉策略的实数遗传算法,即式(4),交叉概率 $p_c = 1$,无变异算子,按比例选择并保留最优个体.需要指出的是,采用式(4)的交叉策略,子代的个体也有可能越出定义区间 $[x_i^L, x_i^U]$.算法中进行判别,若子代个体越出定义区间,则放弃此个体,用式(4)重新生成子代个体.

表 1 三种实验方法结果之比较

Table 1 Results comparison between three experiment methods

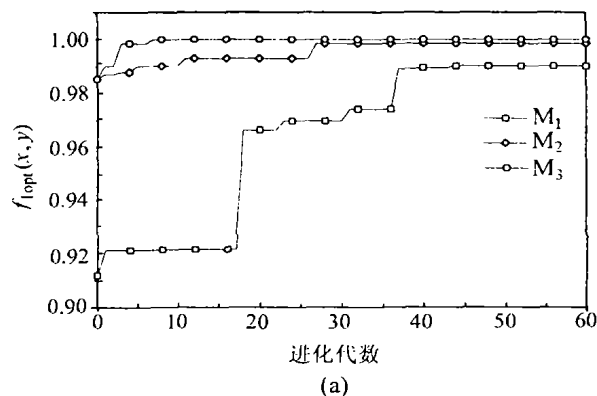
f	M	x	y	$f(x, y)$	t/s	g	$p(\%)$
f_1	M_1	0.9487300	2.9917000	0.990284	29.82	91	0
	M_2	0.0360951	0.0150883	0.998469	14.61	48	20
	M_3	-0.0022775	-0.0021200	0.999990	7.23	28	90
f_2	M_1	-0.0898438	0.7128910	-1.031628	52.13	84	65
	M_2	-0.1049010	0.6912210	-1.026773	13.29	42	10
	M_3	0.0998189	-0.7107720	-1.031193	7.69	27	95
f_3	M_1	0	0	1	71.67	94	100
	M_2	-0.0125966	0.0022719	0.996771	13.62	45	60
	M_3	0.0024325	-0.0041858	0.999024	7.36	31	100

函数 f_1 采用 M_1 方法运行多次均收敛于局部最优 0.990284 且运行时间远大于其他两种方法的运行时间. M_2 方法的效果要比 M_1 方法好,但其收敛于未成熟解的概率非常大.而 M_3 方法从其最终收敛结果和运行时间来看,是成功的,虽然没有完全达到最大值 1,这是由于 M_3 的编码方法也是实数编码.函数 f_2 的测试结果, M_1 方法的收敛概率尚可,这是由于局部最小点较少的缘故,但其运行时间很长.函数 f_3 由于局部极大点较少,采用 M_1 方法,结果都能找到全局最大值 1,但运行时间远长于 M_3 方法.由表中得出的结论是遗传算法采用本文提出的交叉策略的效果从运行时间和收敛概率两方面都要优于其它两种方法.

图 2 给出 f_1, f_2 两个测试函数在三种不同实验方法 M_1, M_2, M_3 下的性能比较,横坐标为算法进化代数,非运行时间.纵坐标为最优个体对应的函数值 $f_{\text{opt}}(x, y)$.图中只画出了算法前 60 代的进化曲线,这是由于在算法程序中采用了连续 20 代最优个体无变化则结束算法的准则,也就是说算法结束前 20 代已收敛于全局最优或局部最优,之后进化曲线为一水平线.图中清楚地表明采用本文提出的交叉策略(M_3 方法)的遗传算法的性能明显优于采用另两

三种算法中群体规模均为 100,总进化代数均设为 100,允许最优解性能无改善的代数为 20,即连续运行算法 20 代,最优解没有改善则结束算法.在 PII300 微机上(内存 64M)分别运行三种算法 20 次,为了有比较性, M_2 与 M_3 方法的初始群体取为一致.所得结果如表 1 所示,表中 $x, y, f(x, y)$ 为算法结束时的典型收敛值, t 表示算法平均运行时间(单位为秒), g 表示平均运行代数, p 为收敛到全局最优解的概率.

种方法的遗传算法的性能, M_3 方法收敛于全局最优的速度要明显快于 M_1 和 M_2 方法,这是由于提出的交叉策略具有指导搜索方向的能力,后代向着适应度改善区域连续前进的事实.换句话说,通过将适应度差异引入交叉操作过程,后代将沿着成功的搜索方向进化,此方向可以从父代中识别出来.对于最优解处于不规则位置的特殊问题,如上述三个测试函数,这样的连续前进的能力可以快速地移动到最优解附近的搜索点,结果是加快发现最优解的速度.通过比较, M_1, M_2 方法不能发现到潜在的搜索方向,从而使得最好个体及整个群体适应度改善的速度受到限制.



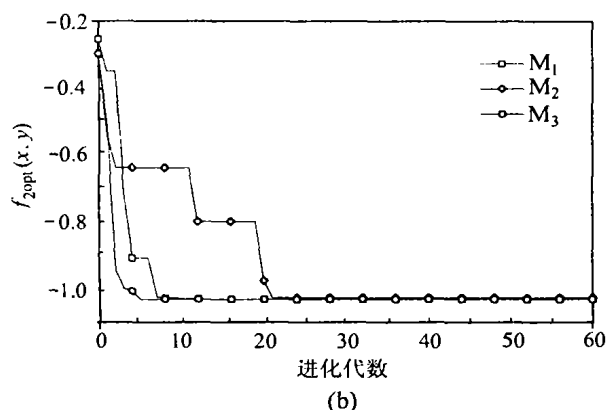


图 2 三种实验方法性能之比较

Fig. 2 Performance comparison between three experiment methods

4 结束语 (Conclusion)

本文探讨了一种新的遗传交叉策略,它使用一个粗糙的但似乎是合理的逼近方法.几个不同的测试函数运行结果说明该交叉策略可以定位后代个体在合理的搜索方向上而不失去基因的多样性,可以使算法快速搜索到高质量解,而传统的 GA 不能提供合理的搜索方向使得最好个体及整个群体适应度改善的速度受到限制.采用这种交叉策略的 GA 在实际应用中有一定的参考价值.

参考文献 (References)

- [1] Holland J H. Adaptation in Natural and Artificial Systems [M]. Ann Arbor, MI: University of Michigan, 1975
- [2] Hajela P. Genetic search — an approach to the nonconvex optimization problem [J]. AIAA Journal, 1990, 28(3): 704 ~ 711
- [3] Wright A. Genetic algorithm for real parameter optimization [A]. The First Workshop on the Foundations of Genetic Algorithms and Classifier Systems [C]. Bloomington, Morgan Kaufmann: Indiana University, 1990, 205 ~ 218
- [4] Pham D T, Jin G. Genetic algorithm using gradient-like reproduction operator [J]. Electronics Letters, 31(18): 1558 ~ 1559
- [5] Hou Gexian, Wu Chengke. The performance analysis for genetic algorithm [J]. Control and Decision, 1999, 14(3): 257 ~ 260 (in Chinese)

本文作者简介

陈小平 1965 年生, 2001 年于南京航空航天大学获工学博士学位. 现在苏州大学从事教学科研工作. 研究领域为进化计算, 神经网络, 计算机测控. Email: chenxp26@public1.sz.js.cn

石 玉 1974 年生, 1999 年于南京航空航天大学获硕士学位. 现为南京航空航天大学博士研究生. 研究领域为进化计算, 信号处理.

于盛林 1941 年生, 1965 年毕业于南京航空学院. 现为南京航空航天大学测试工程系教授, 博士生导师. 研究领域为信号处理, 混沌分形理论及应用.