

一种具有良好多样性的遗传算法

陈 烨

(四川省内江市第六中学 641000)

摘 要 本文提出了通过控制群体多样性来防止实数编码的遗传算法成熟前收敛,并根据多样性在实数编码遗传算法中的体现,引入新的用于控制算法收敛速度的算子。经试验证明,改进后的算法在计算过程中能保持良好的多样性,且具有较好的收敛性。

关键词 遗传算法, 多样性, 收敛性, 适应值

A Genetic Algorithm with Fine Diversity

CHEN Ye

(No.6 Middle School, Neijinag, Sichuan 641000)

Abstract To prevent Real-Number Coded prematurity of genetic algorithm, this paper presents an effective method by controlling the diversity of population. In view of the diversity in Real-Number Coded genetic algorithm, it introduces a new operator to control the speed of convergence. The result of the experiment shows that the new method keeps diversity well and has good astringency.

Keywords genetic algorithm, diversity, astringency, fitness

1 引言

遗传算法(Genetic Algorithm, GA)是一种广泛用于求解最优化问题的算法。它通过模拟生物的进化过程来寻求问题的最优解。作为求解最优化问题的算法,遗传算法具有对目标函数要求低,简单通用等优点。因此,它在许多领域得到了广泛的应用,并已形成一套较完整的算法体系。但是仍有许多问题需要研究。遗传算法虽在求解许多其他方法很难求解的问题方面表现出了良好的性能,但是,对于一些更难得问题,如求一类峰值较多,变化较大的函数时,还存在当算法还未达到全局最优解时就已经收敛的问题,即成熟前收敛问题(prematurity)。这个问题直接关系到算法求解问题的质量和速度。许多人对它进行了研究。

成熟前收敛问题的解决可以从多方面入手。如对个体的进化进行引导(Culture 算法,基于生命周期引导^[1]等方法),调整群体的多样性(diversity)。就我目前所见,防止成熟前收敛效果很明显的一种算法就是文献[2]介绍的算法。这种方法依据群体多样性来调节新群体替换就群体的概率,以此来解决成熟前收敛问题。该文献给出了二进制编码下群体多样性的定义,并根据这个定义,对经典 GA 算法作了改进,试验结果证明这样改进获得的方法有效地控制了种群的多样性,有效地防止了计算过程中的成熟前收敛。这说明通过调整群体的多样性能够很好地解决成熟前收敛问题。但是,文献[2]的这种改进是基于二进制编码遗传算法的。由于实数编码的算法便于引入与问题领域相关的启发性信息以增加算法的搜寻能力,而且能有效地解决二进制编码的 Hamming 悬崖(Hamming Cliffs)的问题,遗传算法也逐渐采用实数编码^[3]。因此,本文提出了实数编码下群体多样性的度量方法,并根据群体多样性决定产生的新群体是否替换就群体,以解决实数编码的遗传算法的成熟前收敛问题。

2 成熟前收敛的原因

对于经典遗传算法,优化问题的求解过程由问题的解所对应的个体的适应值(fitness)来引导。这些个体是从问题定义域中随机抽取的样本,由所抽取的样本对应的适应值的分布情况代表问题的解的分布情况,这就存在抽样误差。由此引导群体进化,必然会产生误差,容易把群体引入局部最优值,造成算法的早熟收敛。除非算法收敛较慢,抽取的样本较多,能够较准确地反映函数的变化情况,否则,由于所计算的样本不能较准确地反映被优化函数的特征,进化求解的过程中,算法就容易收敛到局部最优值。因此,是否发生早熟收敛是与算法的收敛速度相关的。算法收敛速度快,则容易发生早熟收敛;算法收敛速度慢,则容易发生早熟收敛。

用遗传算法求解问题时,通常需要算法收敛。文献[2]中已证明,只要算法收敛,不论它收敛到何值,群体的多样性都会减为零。且当多样性减为零时,算法收敛。由此可知,多样性可用于衡量算法收敛的程度,也可以通过控制多样性来控制算法的收敛速度,从而防止算法收敛到局部最优值。

3 算法的改进

根据上面所述,需要用一个参数来描述群体的多样性。

我们先来讨论各种操作对多样性的影响。

对于下面的最优化问题:

$$\max\{f(x), x \in D \subset \mathbb{R}^n\}$$

其中,目标函数 $f(x) : \mathbb{R} \rightarrow \mathbb{R}$ 经过调整为适合用遗传算法求解的适应函数 $F(x)$,则该优化问题可改写为:

$$\max\{F(x), x \in S\}, \text{ 其中, } S \subset \mathbb{R}^n$$

一种较常用的遗传算法求解此问题的步骤如下:

算法 1:

```
{
  随机初始化种群  $P(0), t=0$ ;
  计算  $P(0)$  中个体的适应值;
  While (不满足终止准则) do
  {
    根据个体的适应值使用下式计算种群内个体的选择概率  $p_i$ :
    
$$P_i = F_i / \sum F_i \quad (1)$$

    根据  $p_i$  从  $P(t)$  中选择  $N_1 (< N)$  个个体;
    对所选择个体进行交叉;
    根据变异概率对交叉后的个体进行变异;
    用产生的  $N_1$  个个体替换掉  $P(t)$  中  $N_1$  个较差的个体,生成新种群  $P(t+1)$ ;
    计算  $P(t+1)$  中个体的适应值,  $t=t+1$ ;
  }
}
```

该算法中,有三个主要算子:选择、交叉、变异算子。选择算子从就群体中抽取个体,问后面的操作做准备,它不直接对群体多样性产生影响。交叉算子对选择出的个体进行杂交。其主要作用是保持群体中有价值的信息,尽量让适应值高的个体保持下去,并使算法有收敛的趋势。它使群体的多样性逐渐减小。变异算子则是为保持群体多样性而设置的。它能降低群体多样性减小的速度。这些操作都缺少对多样性的控制。

设 X 表示群体, P 为群体规模, X_i 为群体中第 i 个个体,其中,最大个体为 X_m ,对

应的适应值为 F_m ，平均适应值为 F_{avg} ，则群体的多样度越大， $1 - F_{avg}/F_m$ 就越大，群体多样度越小， $1 - F_{avg}/F_m$ 就越小，反之亦然。因此，群体的多样度可用下式来度量：

$$D(X) = 1 - F_{avg}/F_m \quad (2)$$

根据上式就可以通过限制和引导 $D(X)$ 值的变化来控制遗传算法的收敛速度。解决普通遗传算法缺少对多样性控制的问题。由上面的分析可知，交叉算子是使遗传算法收敛的主要因素，因此，本文介绍的改进算法就是根据 $D(X)$ 值的变化来限制交叉算子的作用，以达到防止算法成熟前收敛的目的。

原算法在进化过程前期，群体的多样性迅速减小。在优化峰值较多的复杂函数时，就往往会出现成熟前收敛。因此，进化过程前期需要防止算法过快收敛，保持群体的多样性。原算法在进化过程的后期，群体多样性几乎减为零，算法收敛。因此，改进后的算法不能影响进化过程后期的收敛。本文中的改进算法引入了一个变量 p_r 。用该变量表示旧群体被新群体替换的概率。这个概率与新旧群体多样度的变化率和算法的当前迭代次数有关。多样度比旧群体减少的太多或增大的太多的新群体取代旧群体的概率很小。随着迭代次数增大，对多样度的变化的控制越来越强。 P_r 的具体表达式如下：

$$P_r = D(P(t)) - D(P(t+1))^{C/t} \quad (3)$$

改进后的算法如下：

算法 2：

```
{
    确定有关参数：群体规模  $N$ ，迭代次数  $G$ ，操作个体数  $N_1$  ( $< N$ )，常数  $C$  等；
    随机初始化种群， $t=0$ ；
    计算  $P(0)$  中个体的适应值；
    While (不满足终止准则) do
    {
        根据个体的适应值使用下式计算种群内个体的选择概率  $p_i$ ：
            
$$P_i = F_i / F_t$$

        根据  $p_i$  从  $P(t)$  中选择  $N_1$  ( $< N$ ) 个个体；
        对所选择个体进行交叉；
        根据变异概率对交叉后的个体进行变异；
        把  $P(t)$  复制到  $P(t+1)$ ；
        用产生的  $N_1$  个个体替换掉  $P(t+1)$  中  $N_1$  个较差的个体，生成新种群  $P(t+1)$ ；
        计算  $P(t+1)$  中个体的适应值；
        根据下式计算群体被替换的概率：
            
$$P_r = D(P(t)) - D(P(t+1))^{C/t}$$

        其中， $C$  为常数；
        根据  $p_r$  决定是否用新群体替换旧群体；
         $t=t+1$ ；
    }
}
```

4 实验结果及分析

为了检验本文所述算法的有效性，选用以下函数进行测试：

$$\begin{aligned} f_1(x) &= (1 - 2 \sin(3 - x)^{20} + \sin(20 - x)^{20})^{20} & (0 \leq x \leq 1) \\ f_2(x) &= |(1 - x)x^2 \sin(200 - x)| & (0 \leq x \leq 1) \end{aligned}$$

函数 f_1 和 f_2 都是多峰的函数。函数 f_1 在定义域内有 20 个极大、极小值。函数 f_2 在其定义域内有 200 个极大、极小值。这两个函数都是普通算法很难找到全局极值的。它们的图象见图 1 和图 2。

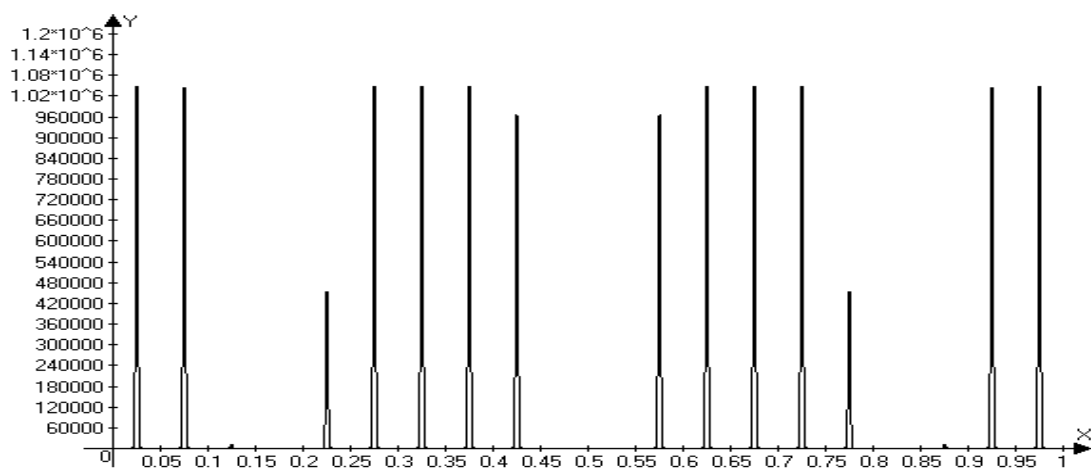


图 1 f_1 图象

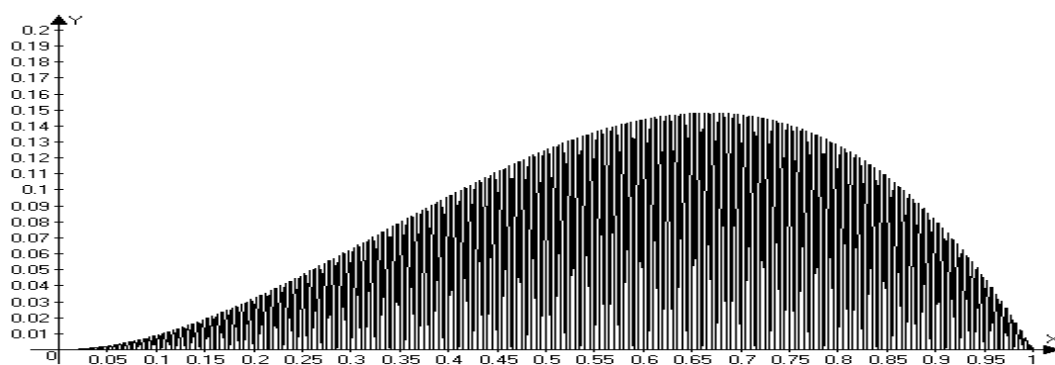


图 2 f_2 图象

第一组实验：用 f_1 作目标函数。用本文所述算法 1 与算法 2 比较。取适应函数 $F_1=f_1$ 。模拟实验所选用的参数如下：

两算法都选用实数编码，群体规模都为 50，迭代次数为 $G=200$ ，取 $N_1=20$ 。对算法 2， $C=10$ 。两种算法各运行十次。其平均适应值变化情况及算法 2 的一幅个体分布图如下：

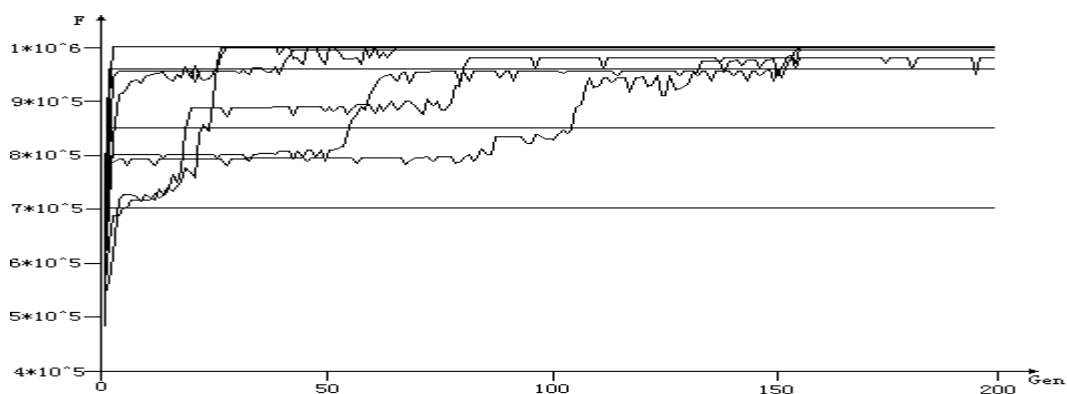


图 3 改进前遗传算法平均适应值图（10 次）

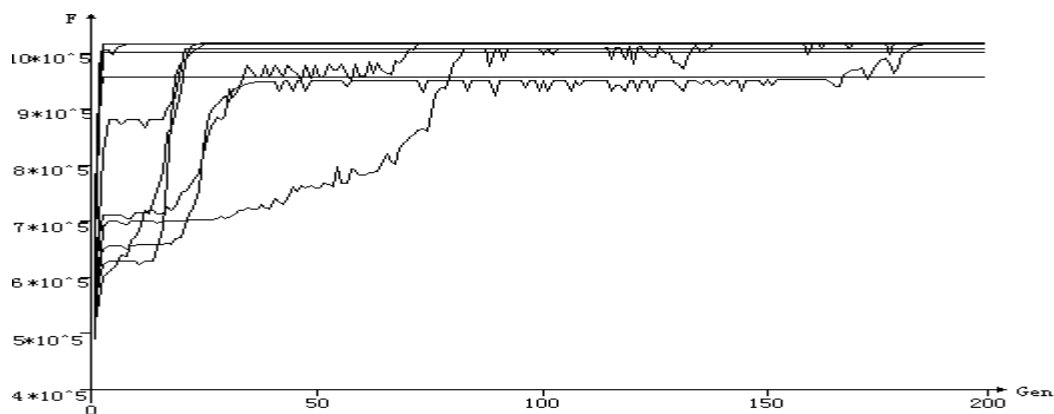


图4 改进后遗传算法平均适应值图(10次)

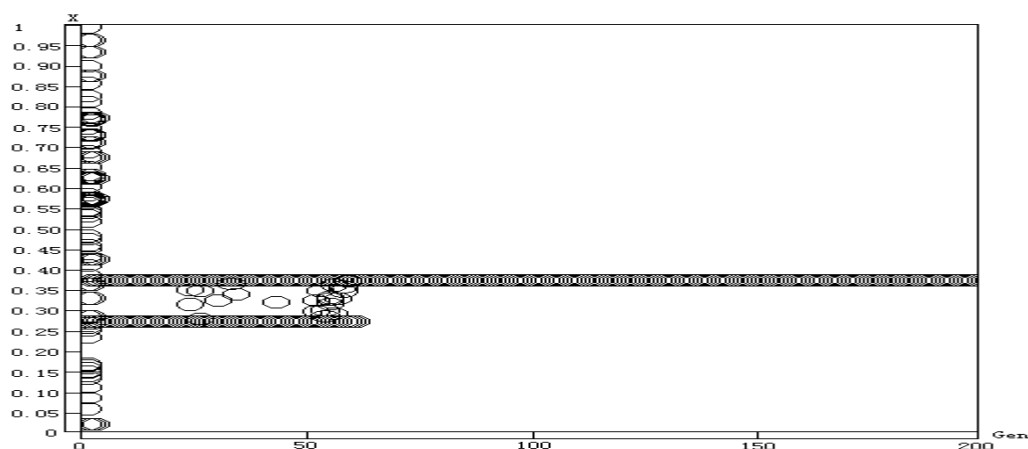


图5 改进算法的一次进化过程的个体分布图

从图中可以看出：改进前的算法不容易收敛到全局最优解，改进后的算法一般都能收敛到全局最优解。且改进后的算法在进化过程前期，个体较分散，具有较好的多样性。在后期，个体趋于集中，算法开始收敛，收敛速度较快。这就达到了本文提出的目标。

第二组实验：优化的目标函数为 f_2 。取适应函数 $F_2=f_2$ 。用本文算法与文献[4]中两组试验的结果对比。本文算法使用的参数出迭代次数 $G=6$ 外，其余参数与第一组试验相同。

表1 三种算法计算 $f_2(x)$ 20次的最大值的均值

算 法	$f_2(x)$ 计算结果	
	迭代次数	最大值
普通 GA 法	80	0.128699
文献[4]改进的 GA 算法	8	0.148092
本文提出的 GA 算法	6	0.147474373996504

从上表看出，改进后的算法收敛到全局最大值的速度明显改善。

5 讨论

5.1 关于多样度量方法

本文中用的多样度用群体平均适应度与最优适应度的比值来度量。为了使多样性越好，计算出的多样度越大，将多样度计算式定义为 1 减去上述比值。这种定义的优点是简单方便，

计算多样性消耗的时间较少。为了提高算法的收敛性能,还可将多样度的度量公式定义为群体适应值的均方差:

$$X = \frac{(F_i - (F_i/n))^2}{n} \quad (4)$$

然后用一个一次函数将上述计算结果映射到区间(0,1)上。

这样定义后除计算时间增长以外几乎没有其它缺点。

5.2 关于算法2中的参数C的选择

C只大小关系到算法对群体多样性控制的迭代次数范围。当C=0时,算法对多样性的控制减为零。当C值更大时,算法对多样性的控制的时间(这个时间对应于算法2中的变量t)更长。

因此,可根据以下公式计算,

$$dD^{C/t} = 1 \quad (5)$$

其中,C为待定系数,t取算法所需控制的时间长度(以迭代次数表示),dD为算法允许的多样度的变化量的最大值。如,算法需要第t代时多样性变化0.1,则dD取0.1。

6 结论

本文首先分析了GA算法出现成熟前收敛的原因,指出其中需要改进之处。在此基础上,提出了实数编码下群体多样性的定义。并根据此定义,对普通遗传算法作了改进。通过模拟比较,结果令人满意。

参考文献

- [1]曹先彬,许凯等.基于生命期引导的生态进化模型.软件学报,2000.11(6):823~828
(Cao Xian-bin, Xu Kai, etc. Ecological Evolution Model Guided by Life Period. Journal of Software. (in Chinese) 2000,11(6): 823~828)
- [2]李书全,赵良英等.一种防止遗传算法成熟前收敛的有效算法.系统工程理论与实践,1999.19(5)
(Li Shu-quan, Zhao Liang-ying, etc. An Effective Method of Preventing Prematurity of Genetic Alogorithm. System Engineering Theory & Practice. (in Chinese) 1999,19(5))
- [3]潘正君,康立山等.演化计算.北京:清华大学出版社,1998.7
(Pan Zheng-jun, Kang Li-shan, etc. Evolutionary Computation. Tsinghua University Press. (in Chinese) 1998.7)
- [4]张铃,张钹.遗传算法机理研究.软件学报,2000.11(7):945~952
(Zhang Ling, Zhang Bo. Research on the Mechanism of Genetic Algorithms. Journal of Software. (in Chinese) 2000,11(7):945~952)
- [5]阎平凡,张长水.人工神经网络与模拟进化计算.北京:清华大学出版社,2000.11:357~415
(Yan Ping-fan, Zhang Chang-shui. ANN & Simulation Evolutionary Computation. Tsinghua University Press. (in Chinese) 2000.11:357~415)
- [6]Nirwan Ansari, Edwin Hou 著,李军,边肇祺译.用于最优化的计算智能.北京:清华大学出版社,1999.12
(Nirwan Ansari, Edwin Hou, Translated by LiJun, BianZhao-qi. Computational Intelligence for Optimization, Tsinghua University Press. (in Chinese) 1999.12)

作者简介:

陈 烨,男,1984年11月出生,现正在内江市第六中学高 二级五班学习
已在《电脑开发与应用》2000年8期上发表论文《中学几何学习软件的设计与实现》
目前对演化计算和神经网络较感兴趣。

电话: 0832 - 2247220

E-mail: chenlixuan0@163.com 或 chenlixuano@163.com

通讯地址: 四川省内江市第六中学 高零二级五班 陈烨 收

邮政编码: 641000