自适应多位变异遗传算法的实现

王基一[1] 吴燕仙[2]

(浙江师大计算机学院 金华321004)[1] (金华教育学院 金华321000)[2]

Implementation of Adaptive Multiple Bit Mutation Genetic Algorithm

WANG Ji-Yi[1] WU Yan-Xian[2]

(Computer School, Zhejiang Normal University, Jinhua 321004)[1] (Jinhua Education College, Jinhua 211000)[2]

Abstract Genetic algorithm is a widely used optimization method. Crossover and mutation are two Basicl operators of the genetic algorithm. On the basis of analyzing the principles of simple genetic algorithm and discussing its existing problems of crossover point and mutation bit, this paper presents a way of the adaptive multiple bit mutation genetic algorithm, which not only can keep the population diversity but also has quicker convergence speed. The results of the multi-modal function optimization show that the adaptive multiple bit mutation genetic algorithm is practical and efficient.

Keywords Adaptive multiple bit mutation genetic algorithm, Crossover probability, Mutation probability, Convergence property

1 引言

遗传算法(Genetic Algorithms,GA)是由 Michigan 大学的 John H. Holland 教授和他的同事、学生在上世纪70年代研究出的一种搜索算法[1]。其目的是解释自然界的自适应过程及设计一个体现自然界机理的软件系统。

虽然 GA 在很多实际问题中都有成功的应用,但本身也存在着一些不足。如优化过程的收敛速度较慢,选择算子、杂交算子的寻优功能随进化迭代次数的增加而逐渐减弱,在应用中常出现早熟收敛,而陷入局部最优,这些不足阻碍了 GA 的应用推广[2]。如何改善 GA 的搜索能力和提高收敛速度是一个重要的问题,因为这是一个能否把 GA 应用到实际问题求解中的先决条件。为此人们对 SGA 进行了各种改进,如自适应遗传算法 AGA(Adaptive Genetic Algorithm)、单亲遗传算法 PGA(Partheno Genetic Algorithm)、基于生物免疫学遗传算法 IGA(Immune Genetic Algorithm)和广义遗传算法 GSAGA(Generalized Self-Adaptive Genetic Algorithm)等[2]。这些算法都不同程度地改善了遗传算法的搜索性能,但当它们在处理复杂问题的优化时,表现仍不十分理想。近几年,又有些专家提出自适应多位变异遗传算法,他们针对对个体的某一位基因进行变异,将大大制约变异算子的搜索能力,而加大变异的位数,则可以增强群体的多样性。其中,周激流等人提出的自适应多位变异遗传算法是根据个体的优劣情况决定其变异位数[3],扩展了搜索空间,改善了遗传算法的搜索性能,为进一步提高遗传算法的效率,扩大应用面提出了方向性的思想。

本文主要在遗传算法函数优化问题这一应用领域展开研究,提出利用产生随机数的方式来随机选取多位变异基因所在位置,并给出基于 MATLAB 下实现两点交叉和多位变异的算法,实例计算表明自适应多位变异遗传算法是有效和可行的。

2 自适应多位变异算子的原理

遗传算法的搜索性能主要是通过选择、交叉和变异三个操作算子来实现,尤其是变异算子,它可以增加新的搜索空间,扩大算法的搜索范围,更有可能找到全局最优解,但由于搜索范围广,收敛的速度受到影响,因而设计自适应的变异操作非常重要。传统的遗传算法和各种改进的算法在进行变异操作时,都采用随机选取一位进行翻转变值的方法,不能有效提高二进制编码的搜索性能及稳定性。周激流、丁晶、金菊良等人提出根据个体的优劣情况决定其变异位数,即适应度低的个体变异多个基因,而适应度高的个体则采用少位变异或不变异;这样可形成多种变异组合,扩大了搜索空间。

设自适应变异概率 p_m 由下式给出

$$\begin{cases} p_m = k_3 \dfrac{f_{max} - f_i}{f_{max} - f_{avg}} & f \geqslant f_{avg} \\ p_m = k_4 & f < f_{avg} \end{cases} \quad (1)$$

其中,f_{max}:群体中最大的适应度值;f_{avg}:每代群体的平均适应度值;f:两个交叉个体中最大的适应度值;f_i:要变异个体的适应度。

$0 \leqslant k_3, k_4 \leqslant 1$,用上式调整 p_m 能实现既防止过早收敛,又保护优良个体。同时,为了保证每一代的优良个体不被破坏,采用最优保存策略,使它们直接复制到下一代中。

则其变异位数由下式给出[3]:

$$MI = ((INT)(MT \times (f_{max} - f_i)/(f_{max} - f_{min}))) \quad (2)$$

其中,MT 为常数,实际应用中一般取 MT 为染色体串长的 1/4到1/3,即

$$L \times 1/4 < N < L \times 1/3$$

$f_{max} - f_{min}$ 表明当前代的适应度范围;$f_{max} - f_i$ 则表明个体的适应度值与适应度最大值之间的距离。$F = (f_{max} - f_i)/(f_{max} - f_{min})$ 则表明了个体 f_i 在当前代中的优劣程度,F 越大,则该个体质量越差;反之,则越优秀。通过式(2)就可动态地根据个体的优劣情况,决定其变异的位数:质量好的个体变异位数少,甚至无变异位;质量差的个体变异位数多,也即加

大其变异搜索范围。

由(1)和(2)式确定变异概率和变异位数，如何选取变异基因所在的位置?而变异基因位置的选取是很重要的，因为不同基因位的改变造成个体适应度的改变程度不同，也即各个基因位的重要性不是等同的。

本论文采用随机数的方式来选取变异个体的变异位，同时结合两点交叉。具体实现如下：

(1)在进行交叉操作时，采取两点交叉。先生成2个随机数 R_1，R_2，再转成十进制整数：$IR_1=INT(R_1\times L)$，$IR_2=INT(R_2\times L)$。若 $R_1<R_2$，不交换它们的值，否则 R_1 与 R_2 交换，也即 R_1 要小于 R_2。第 i 对双亲 IA1(j,k,i)和 IA2(j,k,i)的两点杂交是指将它们的二进制串中第 IR_1 位到 IR_2 位的数字段相互交换，得到两个子代个体 I`A1(j,k,I+1)和 I`A2(j,k,I+1)。

(2)在进行变异操作时，先确定个体是否发生变异，再确定要变异的个体的变异位数。变异位数为 MI，所以生成 MI+1个随机数 R_1-(R_{MI+1})，其中 R_{MI+1} 控制子代个体发生变异的可能性。对 N 个子代个体，记其二进制数为{IA(j,k,I)}，即第 I 代个体第 j 变量对应的第 k 位基因。把 R_1-(R_{MI})转化成小于 L 的整数：$IR_1=INT(R_1\times L)\cdots IR_{MI}=INT(R_{MI}\times L)$。变异率($P_m$)定义为子代个体发生变异的概率。所谓子代个体 IA(j,k,I)多点变异，是指如下变换

IA(j,k,I)：当 $R_{MI+1}\leqslant P_m$ 且 $k\in\{IR_1,IR_2,\cdots,IR_{MI}\}$ 时，原 k 位值1变为0，原 k 位值0变为1；其他情况不变。

这里，利用 $R_1,R_2,\cdots,R_{MI}$ 来随机选取子代个体串中将发生变异的 MI 个位置，利用 R_{MI+1} 来控制子代个体发生变异的可能性。

3 基于 MATLAB 的自适应多位变异算法的设计与实现

限于篇幅，本文仅列出变异操作程序，其它步骤省略

程序中变量说明：popSize 种群大小，xZomeLength 个体二进制位数+1，mutOps 变异概率，numMuts 变异的位数，opts 计算精度，bounds 变量取值范围。

```
for i=1:popSize %计算每一个个体的变异概率
  if
max(startPop(:,xZomlength)) - startPop(i,xZomelength) < (max
(startPop(:,xZomeLength)) - mean(startPop(:,xZomeLength)))
mutOps(i)=0.5 * ( max(startPop(:,xZomeLength)) - startPop(i,
xZomeLength))/(max(startPop(:,xZomeLength)) - mean(startPop
(:,xZomeLength)));
    else mutOps(i)=1.0;
    end
  end
  mt=round(numVar/4)+1;
  mN=deblank(mutFNs);
  mp=find(rand(popSize,1)<mutOps); %根据概率判断个体
  是否需要变异
  for j=1:size(mp,1)
  if (max(startPop(:,xZomeLength)) == startPop(mp(j),
    xZomeLength)) %最优个体不发生变异
    numMuts=0;
  else
   numMuts=round(mt * ((max(startPop(:,xZomeLength)) -
   startPop(mp(j),xZomeLength))/(max(startPop(:,xZome -
   Length)) - min(startPop(:,xZomeLength)))))
   %计算出要变异的个体的变异位数
     endPop(mp(j),:)=feval(mN,endPop(mp(j),:),bounds,
     [gen mutOps(mp(j))], evalFN, numMuts,opts); %调用
     变异操作函数
     end
     end
   function [parent] = binaryMutate(parent,bounds,Ops,e-
   valFN numMut,opts)
   pm=Ops(2);
   numVar = size(parent,2) - 1;
   bits=calcbits(bounds,Opts(1)); %计算出个体的二进制位
   数
   mpoint=round(rand(1,numMut) * (numVar - 1)) + 1; %随
   机确定 numMut 个变异位
   child=parent;
   child(mpoint(1,:))=abs([parent(mpoint(1,:)) - 1]); %
   变异的所在位取反
   x1=b2f(parent,bounds,bits); %计算经变异后个体的适应
   度值
 [x1 child(numVar+1)]=feval(evalFN,x1,Ops);
```

在运行遗传算法程序时，对一些参数作事先选择，它们包括种群的大小、计算精度、最大进化代数、目标函数选择等。一般说来，选择较大数目的初始种群可以同时处理更多的解，因而容易找到全局最优解，其缺点是增加了每次迭代的时间，一般取20-100之间。染色体的长度主要决定于问题求解的精度的要求，精度越高，染色体长度越长，搜索空间越大，相应种群大小设置大一些。最大进化代数作为算法终止条件，一般取100-500代。

4 自适应多位变异遗传算法的性能分析

评价遗传算法的性能，通常考虑以下方面：算法的收敛速度；搜索到全局最优解的能力；解决多峰值函数优化问题的能力等[2]。为了评价本文设计的自适应多位变异遗传算法的求解性能，采用多峰值函数优化问题进行讨论，并与自适应一位变异遗传算法、标准遗传算法进行对比。同时，遗传算法的初始种群生成是随机的，只凭几次运行结果不能说明问题，因此对所有的测试算法均独立运行30次。考虑算法收敛到全局最优点的最大迭代代数以及优化值中的最优值、平均值。

考虑如下优化问题：

f(x)=x * sin(10(x)+2，优化变量区间为[-1,2]。该函数存在多个极值点，最大极值点是 x=1.85，f(1.85)=3.85。其函数曲线如图1所示。

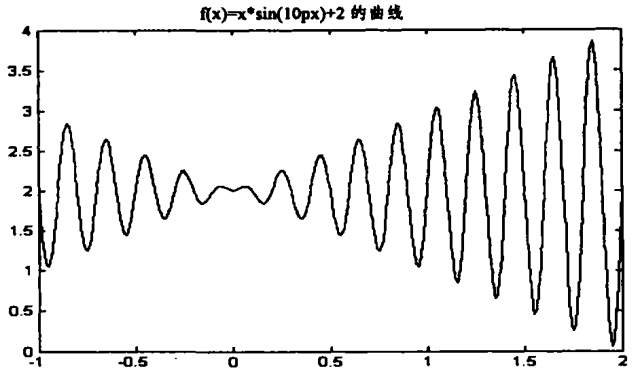

图1

4.1 算法的全局收敛性能

我们对上述函数分别用三种遗传算法进行测试，最优值结果如图2所示。其中，种群规模80，变量计算精度10^{-6}，最大进化代数200，收敛到最佳个体 x=1.8505，f(x)=3.850274时运行的代数如表1。个体对应的解与微分法预计最优解的情况吻合，最大函数值比3.85略大，可以作为问题的近似解。

表1 三种算法比较

优化方法	交叉概率与变异概率	收敛到最佳个体代数 x=1.8505，f(x)=3.850274
标准遗传算法	交叉概率0.6，变异概率0.8	第200代
自适应一位变异遗传算法	在进化过程中随个体的适应度改变	第37代
自适应多位变异遗传算法	在进化过程中随个体的适应度改变	第12代

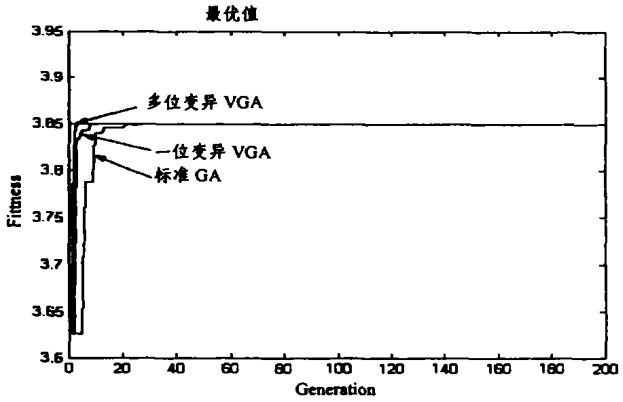

图2

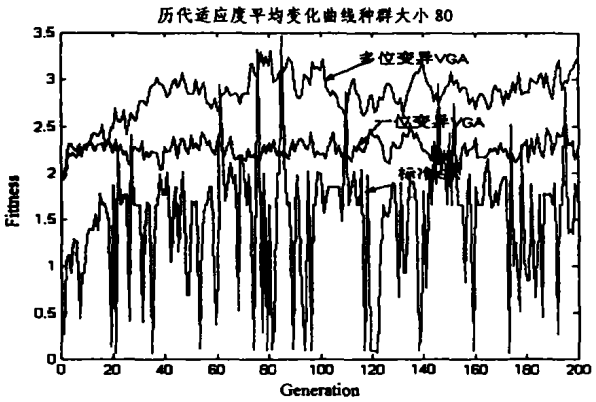

图3

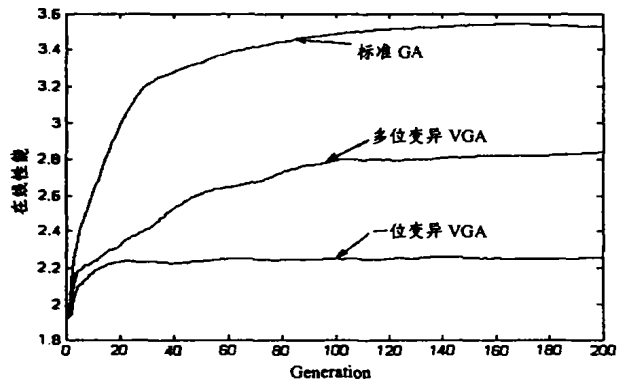

图4

三种遗传算法的历代进化的个体适应度平均值变化曲线如图3所示。

各组实验所得优化值的最优值反映了算法所能达到的最大精度，平均值反映了算法的效率。自适应多位变异遗传算法具有全局收敛性且收敛速度最快。

4.2 算法的在线特性和离线特性[6]

在线特性和离线特性。在线特性算法的动态性能；离线特性反映了算法的收敛性能。对上述函数进行测试，结果如图4和图5所示。

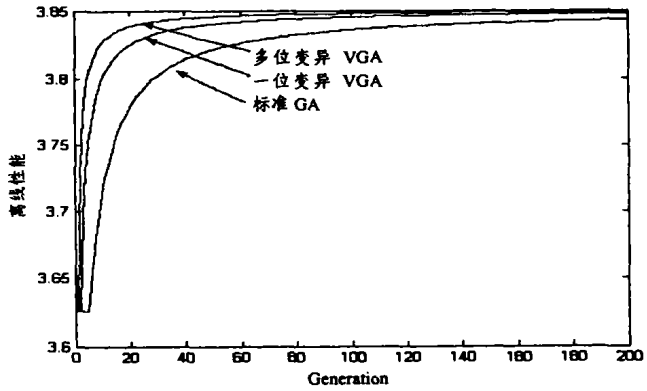

图5

从图中可以看出，自适应多位变异遗传算法的在线特性介于标准遗传算法和自适应一位变异遗传算法之间；而离线特性要优于自适应一位变异遗传算法和标准遗传算法。

结论 遗传算法的控制参数对算法本身的性能有重要的影响。基本遗传算法的控制参数在进化过程中保持不变，在多峰值函数优化时收敛速度慢。本文实现的自适应多位变异遗传算法充分考虑了遗传算子在不同进化时期的作用，改善了算法的性能，并且在多峰值函数优化求解中显示了优良的特性。

参 考 文 献

1 John H. Adaptation in Nature and Aritificial Systems . The University of Michigan Press，1975

2 王小平，曹立明. 遗传算法--理论、应用与软件实现. 西安交通大学出版社，2002，1(1)

3 周激流，丁晶，金菊良. 一种新型遗传算法及其在暴雨强度公式参数优化中的应用研究. 四川大学学报，2000，37(4)

4 Binary and Real-Valued Simulation Evolution for Matlab Copyright (C)1996 C. R. Houck, J. A. Joines, M. G. Kay

5 飞思科技产品研发中心. MATLAB 6.5辅助优化计算与设计. 电子科技出版社，2003，1(1)

6 De Jong K A. Analysis of the Behavior of a Class of Genetic Adaptive Systems [D]. University of Michigan，1975

(上接第140页)

论了历史数据对预测结果的影响。将自相关函数的概念应用于神经网络预测模型中的输入变量集选择，有明确的理论依据，通过该方法可以得到更准确的输入变量集，FFT 的采用增加了该方法的实用性和可操作性，预测误差更小，算例也验证了该方法的有效性。本文所提出的方法同样适用于城市供热负荷、城市用水量等神经网络预测模型中输入变量集的选择。

参 考 文 献

1 Vila J P, Wagner V, Neveu P. Bayesian Nonlinear Model Selection and Neural Networks: A Conjugate Prior Approach. IEEE Trans on neural networks, 2000,11(2):265～278

2 Matsui T, Iizaka T. Peak Load Forecasting Using Analyzable Structured Neural Network [A]. IEEE PES 2001 Winter Meeting, Columbus, Ohio USA, 2001

3 Drezga I, Rahman S. Input Variable Selection for ANN-Based Short-Term Load Forecasting. IEEE Trans on Power Systems, 1998, 13(4):1238～1244

4 Mastorocostas P A, Theocharis J B, Bakirtzis A G. Fuzzy Modeling for Short Term Load Forecasting Using the Orthogonal Least Squares Method. IEEE Trans on Power Systems, 1999,14(1):29～36

5 荣辉. 基于前馈神经网络的电力系统负荷预测研究:[学位论文]. 北京:清华大学，1998

6 Yuan J L, Fine T L. Neural Network Design for Small Training Sets of High Dimension. IEEE Trans on Neural networks, 1998, 9(2):266～280