实数编码遗传算法中常用变异算子的Matlab实现及应用

黄卫华[1],许小勇[1],范建坤[2]
(1.云南民族大学数学与计算机科学学院,云南 昆明 650031;2.昆明理工大学理学院,云南 昆明 650031)

【摘 要】本文对实数编码遗传算法最常用的四种变异算子进行介绍,将其编写成Matlab程序,并在最优化问题上进行测试。

【关键字】实数编码;遗传算法;变异算子;Matlab
【中图分类号】O242.1 【文献标识码】A 【文章编号】1003-2673(2007)01-0077-02

遗传算法具有广泛的应用领域,它借助于生物进化的思想和原理与计算机科学相结合,在解决实际问题中得到了很好的应用。遗传算法一般由选择、交叉、变异构成,变异发生的可能性较小。实数编码遗传算法中常用的变异算子有:均匀变异、边界变异、非均匀变异、高斯变异等四种算子[1]。本文针对这个常用的四种变异算子进行了详细介绍,并用Matlab的函数实现了上述四种变异算子,并给出了应用实例,在实例中都能得到全局最优解,且收敛速度快,充分利用各变异算子的优点。

1 均匀变异算子

均有变异(UniformMutation)操作是指分别用符合某一范围内均匀分布的随即数,以某一较小的概率来替换个体编码串中各个基因座上的原有基因值。均匀变异的具体操作过程是:

(1) 依次指定个体编码串中的每各个基因座为变异点;

(2) 对每一个变异点,以概率 p_m 从对应基因的取值范围内取一随机数来来代替原有基因值。

假设有一个个体为 $X=X_1X_2\cdots X_k\cdots X_l$,若 X_k 为变异点,其取值范围为 $[U^k_{min},U^k_{max}]$,在该点对个体X进行均有变异后,可得到一个新的个体 $X=X_1X_2\cdots X_k`\cdots X_l$,其中变异点的新基因值是:$X_k`=U^k_{min}+r(U^k_{max}-U^k_{min})$。式中,r为范围内符合均匀分布的一个随机数。均匀变异算子函数Uniform_Mutation的Matlab程序为(符号说明:pop为种群,pm为变异概率,bound为参数变化范围组成的矩阵,下同):

```
function [mpop]=Uniform_mutation(pop,pm,bound)
[m,n]=size(pop);
mpop=pop;
for i=1:m
   mpoint=round(rand*(n-1))+1;
   x=pop(i,mpoint);
   if rand<pm
      y=bound (1,mpoint)+rand* (bound (2,mpoint)-bound(1,
mpoint));
    mpop(i,mpoint)=y;
   end
end
```

2 边界变异算子

边界变异算子(Boundary Mutation)是均匀变异操作的一个变形遗传算法。在进行边界变异操作时,随机地取基因座的二个对应边界基因之一取代替原有基因值。

在进行由 $X=X_1X_2\cdots X_k\cdots X_l$ 向 $X=X_1X_2\cdots X_k`\cdots X_l$ 的边界变异时,若变异点 $X_k$ 处的基因值取值范围为 $[U^k_{min},U^k_{max}]$,则新的基因 $X_k`$ 由下式确定:$X_k`=\begin{cases}U^k_{min} & \text{if random}(0,1)=0\\ U^k_{max}, & \text{if random}(0,1)=1\end{cases}$,

式中 if random(0,1) 表示以均等概率从0、1中任取其一。边界变异算子函数B_Mutation的Matlab程序为:

```
function [mpop]=B_mutation(pop,pm,bound)
[m,n]=size(pop);
mpop=pop;
for i=1:m
  mpoint=round(rand*(n-1))+1;
  if rand<pm
    if rand<0.5
      mpop(i,mpoint)=bound(1,mpoint);
    else
      mpop(i,mpoint)=bound(2,mpoint);
    end
  end
end
```

3 非均匀变异算子

非均匀变异的具体操作过程于均匀那变异相似,但它重点搜索原个体附近的微小区域。在进行由 $X=X_1X_2\cdots X_k\cdots X_l$ 向 $X=X_1X_2\cdots X_k`\cdots X_l$ 的非均匀变异操作时,若变异点 $X_k$ 处的基因值取值范围为 $[U^k_{min},U^k_{max}]$,则新的基因 $X_k`$ 由下式确定:

$$X_k`=\begin{cases}X_k+\triangle(t,U^k_{max}-X_k) & \text{if random}(0,1)=0\\ X_k-\triangle(t,X_k-U^k_{min}), & \text{if random}(0,1)=1\end{cases},$$

式中,$\triangle(t,y)$,(y表示 $U^k_{max}-X_k$ 和 $X_k-U^k_{min}$)表示[0,y]范围内符合非均匀分布的一个随机数,要求随着进化代数t的增加,$\triangle(t,y)$ 接近于0的概率也逐渐增加。例如,$\triangle(t,y)$ 可按下式定义:$\triangle(t,y)=y.(1-r^{(1-t/T)b})$,式中r为[0,1]范围内符合均匀分布的一个随机数,T时最大进化代数,b时一个系统参数,它决定了随机数扰动对进化代数t的依赖程度。非均匀变异算子函数Nonuniform_mutation的Matlab程序为:

【作者简介】黄卫华(1979-),女,河南中牟人,在读研究生,主要从事信息代数、半群与粗糙集的学习与研究。

```
function [mpop]=Nonuniform_mutation(pop,pm,bound,t,T)
b=2;%可取 2～5
[m,n]=size(pop);
mpop=pop;
for i=1:m
  mpoint=round(rand*(n- 1))+1;
  x=pop(i,mpoint);
  a=(1- t/T);
  if rand<pm
    if rand<0.5
      r=rand;
      y=x+(bound(2,mpoint)- x)*(1- r.^(a*b));
      mpop(i,mpoint)=y;
    else
      r=rand;
      y=x- (x- bound(1,mpoint))*(1- r.^(a*b));
      mpop(i,mpoint)=y;
    end
  end
end
```

4 高斯变异算子

高斯变异(Gaussian Mutation)是改进遗传算法对重点搜索区域大局部搜索性能的另一种变异操作方法。所谓高斯变异操作是指进行变异操作时,用符合均值为 μ、方差为 σ^2 的正态分布的一个随机数来替换原有基因值。由正态分布的特性可知,高斯变异也时重点搜索原个体附近大某个局部区域。高斯变异的具体操作过程于均匀变异相似。

具体实现高斯变异时,符合正态分布的随机数 Q 可由一些符合均匀分布的随机数 $r_i(i=1,2,\cdots,12)$,则符合 $N(\mu,\sigma^2)$正态分布的一个随机数 Q 可由此求得:$Q=\mu+\sigma(\sum_{i=1}^{12} r_i-6)$,在进行由 $X=X_1X_2\cdots X_k\cdots X_l$ 向 $X=X_1X_2\cdots X_k`\cdots X_l$ 的高斯变异操作时,若变异点 $X_k$ 处的基因值取值范围为 $[U^k_{min},U^k_{max}]$,并假设 $\mu=\frac{U^k_{min}+U^k_{max}}{2}$,$\sigma=\frac{U^k_{max}-U^k_{min}}{6}$,则新的基因 $X_k`$ 由下式确定:

$X_k`=\frac{U^k_{min}+U^k_{max}}{2}+\frac{U^k_{max}-U^k_{min}}{6}(\sum_{i=1}^{12} r_i-6)$。高斯变异算子函数 Gaussian_muatation 的 Matlab 程序为:

```
function [mpop]=Gaussian_mutation(pop,pm,bound)
[m,n]=size(pop);
mpop=pop;
for i=1:m
  r=rand(1,12);
  s=sum(r);
  mpoint=round(rand*(n- 1))+1;
  u=(bound(1,mpoint)+bound(2,mpoint))/2;
  v=(bound(2,mpoint)- bound(1,mpoint))/6;
  q=u+v*(s- 6);
  if rand<pm
     mpop(i,mpoint)=q;
   end
end
```

5 应用实例

在一下两个应用实例中[2],选择算子采用最优保存策略与具有排名的转盘式相结合,交叉算子采用均匀交叉算子,为充分利用上述四种变异算子的优点,将进化代数 T 分为四个阶段,当前进化代数记为 t,当 0< t≤T/4 时采用均匀变异算子,T/4< t≤T/2 时采用非均匀变异算子,T/2< t≤3T/4 时采用高斯变异算子,3T/4< t≤T 采用边界变异算子。参数设置如下:最大进化代数 T=240,种群数目 M=120,交叉概率 $P_c=0.85$,变异概率 $P_m=0.01$。测试函数如下:

(1) $\max f(x_1,x_2)=100(x_1^2-x_2)^2+(1-x_1)^2$,
$s.t -2.048\le x_i\le 2.048, i=1,2$

运行结果:程序在 43 代搜索到了全局最优解 $x_1=-2.048$,$x_2=-2.048$ 最优解,和最优值 3905.926。

进化代数与最优函数值变化曲线为:

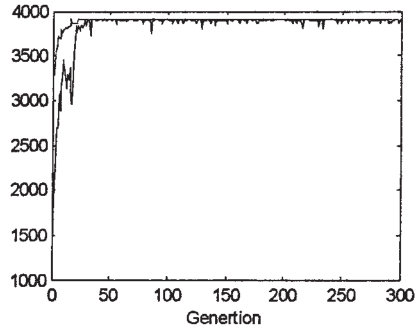

图 1 进化代数与最优函数值变化曲线

(2) $\max f(x_1,x_2)=4-(x_1^2+2x_2^2-0.3\cos(3\pi x_1)-0.4\cos(4\pi x_2))$, $s.t -1.024\le x_i\le 1.024, i=1,2$

运行结果:程序在 44 代搜索到了全局最优解,最优解 $x_1=0.2234^*10^{-8}$, $x_2=0.1213^*10^{-8}$(精确解为(0,0))和最优值 4.7,进化代数与最优函数值变化曲线为:

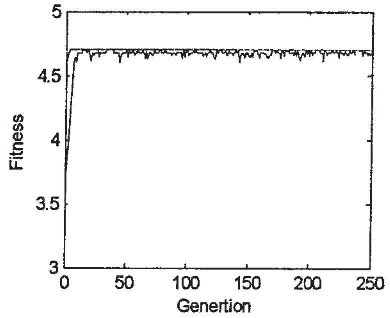

图 2 进化代数与最优函数值变化曲线

通过以上两个实例可以看出,将各变异算子融合在一起发挥各算子的优点,为改善遗传算法、提高遗传算法的收敛速度提供了一条有益的思路。

参考文献

[1] 周明,孙树栋.遗传算法原理及应用[M].北京,国防工业出版社,1999.

[2] 王成栋,张优云.基于实数编码的自适应伪并行遗传算法[J].西安交通大学学报,2003.(7):707- 710.

[3] 刘则毅.科学计算技术与 Matlab[M].科学出版社, 2001.