

遗传算法中常用选择算子在 MATLAB 中的实现

沈崇圣

(上海应用技术学院 数理教学部, 上海, 200235)

摘要: 在数学软件 MATLAB 中, 没有遗传算法的程序, 本文编写了遗传算法中常用的四种选择算子的 MATLAB 语言程序, 并在一最优化问题上进行测试。

关键词: 遗传算法; 选择算子

中图分类号: O 242.1; TP 311

文献标识码: A

Realization in MATLAB of the Selection Operator of Genetic Algorithm

SHEN Chong-sheng

(Department of Maths and Physics Education, Shanghai Institute of Technology, Shanghai 200235, China)

Abstract: There is no genetic algorithm programme in MATLAB, and this paper programmes of four selection operators which are often used have been realized in MATLAB and used to solve a function optimization problem.

Key words: genetic algorithm; selection operator

遗传算法在很多领域得到广泛的应用, 其借助于生物进化的思想和原理, 与计算机科学相结合, 取得了很大的成功。由不同的编码方案和选择策略构成了不同的遗传算法。而在选择策略中一般总由选择算子, 杂交算子和变异算子来模仿生物进化中的遗传和变异。

在广泛应用的数学软件 MATLAB 中并没有现成的遗传算法的程序, 为了在 MATLAB 中进行遗传算法的计算, 必须对所研究的问题进行编程。本文对遗传算法中常见的四种选择算子, 用 MATLAB 语言进行编程。

在遗传算法中, 常用的选择算子有: 转盘式选择算子, 具有排名的转盘式选择算子, 随机一致选择算子和锦标赛选择算子。本文用 MATLAB 的函数实现上述四种算子, 并统一输入参数和输出参数, 在主程序中只要适当调整输入、输出参数就可以替换调用。

1 转盘式选择算子

这是遗传算法中使用最普遍的选择算子。先计算种群中每个个体的相对适应值 $\frac{f_i}{\sum f_i}$, 将其作为选择概率 p_i , 在选择时, 先产生一个 $[0, 1]$ 内的随机数 r , 如 $p_0 + p_1 + p_2 + \dots + p_{i-1} < r \leq p_1 + p_2 + \dots + p_i$ ($p_0 = 0$), 则选择个体 i 。

在以下的选择算子函数中, 输入的种群 pop 是一个 $n \times m$ 矩阵, 其中每一行是一个个体, 以二进制

编码表示。pop 中有 n 个个体。popfit 是一个 $1 \times m$ 行向量,表示 pop 中每一个个体的适应度。输出的下一代新种群为 newpop,也是一个 $n \times m$ 矩阵。

转盘式选择算子函数 select1,输入参数是种群 pop 和适应值 popfit,输出参数是选择后的新一代种群 newpop。其基本结构为:

```

计算个体相对适应值 pfit;
计算个体的相对适应值的累积和 pfit;
产生随机数向量 rs;
for i = 1:n
    for j = 1:n
        if rs(i) ≤ pfit(j)
            选择个体 j,作为新种群中的个体 i
        否则,保留新种群中的个体 i
    end
end

```

转盘式选择算子函数 select1 的 MATLAB 程序为:

```

function newpop = select1(pop, popfit)
    pfit = popfit ./ sum(popfit);
    pfit = cumsum(pfit);
    [px, py] = size(pop);
    if pfit(px, 1) < 1
        pfit(px, 1) = 1;
    end
    rs = rand(1, px);
    for i = 1:px
        ss = 0;
        for j = 1:px
            if rs(i) ≤ pfit(j)
                newpop(i, :) = pop(j, :);
                ss = 1;
            end
        end
        if ss == 1
            break;
        end
    end
end
end

```

2 具有排名的转盘式选择算子

具有排名的转盘式选择算子与转盘式选择算子的不同之处是选择概率的计算。具有排名的转盘式选择算子中的选择概率不是个体的相对适应值,而是按照个体适应值大小排列 $x_1 \geq x_2 \geq \dots \geq x_n$,然后根据线性函数计算选择概率:

$$p_i = \frac{1}{n} \left(a - \frac{bi}{n+1} \right) \quad i = 1, 2, \dots, n \quad (1)$$

式中: i 为排名序数, n 为种群规模, a, b 是常数, $1 \leq a \leq 2, b = 2(a-1)$, 通常 $a = 1.1$ 。有了选择概率,我们按照转盘式选择方式选择父体进行遗传操作。

具有排名的转盘式选择算子函数 select2,输入参数是种群 pop,适应值 popfit 和常数 a ,输出参数是选择后的新一代种群 newpop。其基本结构与转盘式选择算子的结构基本相同,其不同点是计算选择概率是用线性函数计算。

具有排名的转盘式选择算子函数 select2 的 MATLAB 程序为:

```

function newpop=select2(pop,popfit,a)
    b=2*(a-1);
    [ssfit,II]=sort(popfit);
    [px,py]=size(pop);
    I=flip(II);
    I=I';
    for i=1:px
        pfit(1,i)=(a-b*i/(px-1))/px;
    end
    pfit=cumsum(pfit);
    if pfit(1,px)<1
        pfit(1,px)=1;
    end
    rs=rand(1,px);
    for i=1:px
        ss=0;
        for j=1:px
            if rs(i)<=pfit(j)
                newpop(i,:)=pop(I(1,j),:);
                ss=1;
            end
        end
        if ss==1
            break;
        end
    end
end
end

```

3 随机一致选择算子

随机一致选择算子中的选择概率是个体的相对适应值 $p_i = \frac{f_i}{\sum f_i}$,而在选择时产生随机数的方式

不同,在转盘式选择算子中随机地产生一个 $1 \times n$ 随机向量,每次选择时使用一个分量进行选择,而在随机一致选择算子中只产生一个初始随机数,然后根据一定的步长,每次由初始随机数加上步长的倍数产生一个随机数进行选择。

随机一致选择算子函数 select3 的输入参数是种群 pop 和适应值 popfit,步长为种群规模的倒数 $\frac{1}{n}$,输出参数是新一代种群 newpop。

```

function newpop=select3(pop,popfit)
    [px,py]=size(pop);
    dpointer=1/px;
    firstp=unifrnd(0,dpointer);
    pfit=popfit./sum(popfit);
    pfit=cumsum(pfit);
    if pfit(px,1)<1
        pfit(px,1)=1;
    end
    newpop=pop;
    for i=1:px
        npointer=firstp+(i-1)*dpointer;
        if npointer>1

```

```

    npointer=1;
end
ss=0;
for j=1:px
    if npointer<pfit(j)
        newpop(i,:)=pop(j,:);
        ss=1;
    end
    if ss==1
        break;
    end
end
end
end

```

4 锦标赛选择算子

锦标赛选择算子在选择时,从种群中随机地选取 k 个个体,找出这 k 个个体中适应值最好的个体作为最优个体,这个最优个体就是下一代种群中的一个个体,这个过程重复 n 次就产生了新的种群。

锦标赛选择算子函数 select4 的输入参数是种群 pop,适应值 popfit 和竞赛规模 k ,输出参数是新种群 newpop。

```

function newpop=select4(pop,popfit,k)
[px,py]=size(pop);
for i=1:px
    tempfit=zeros(1,k);
    rs=unidrnd(px,1,k)
    for j=1:k
        tempfit(1,j)=popfit(rs(j),1);
    end
    [sffit,II]=sort(tempfit);
    newpop(i,:)=pop(rs(II(k)),:);
end

```

5 实例

对检测函数

$$\begin{aligned} \max f(x_1, x_2) &= 100(x_1^2 - x_2)^2 + (1 - x_1)^2 \\ \text{s. t. } -2.048 &\leq x_i \leq 2.048 \quad i = 1, 2 \end{aligned}$$

交叉和变异采用单点交叉算子和一致变异算子,而选择过程分别采用上面四种算子在计算机上运算。其中,采用转盘式选择算子,在第 77 代,采用具有排名的转盘式选择算子,在第 16 代,采用随机一致选择算子,在第 41 代采用锦标赛选择算子,在第 9 代得到相同的最优解: $x_1 = -2.048$, $x_2 = -2.048$ 和最优值 3 905.9。

参考文献:

- [1] 陈桂明,戚红雨,潘伟. MATLAB 数理统计[M]. 北京:科学出版社,2002.
- [2] 李胡锡,姜红. MATLAB 循序渐进[M]. 上海:上海交通大学出版社,1997.
- [3] 潘正君,康立山,陈毓屏. 演化计算[M]. 北京:清华大学出版社,南宁:广西科学技术出版社,1998.
- [4] 王小平,曹立明. 遗传算法—理论、应用与软件实现[M]. 西安:西安交通大学出版社,2002.