第6卷 第1期 集美大学学报（自然科学版） Vol.6 No.1
2000年3月 Journal of Jimei University(Natural Science) Mar. 2001

[文章编号] 1007－7405(2001)01－0014－06

基于MATLAB的遗传算法程序设计及TSP问题求解

储理才

（集美大学基础教学部，福建 厦门 361021）

[摘要] 首先分析了用Matlab语言设计遗传算法程序的优越性，接着以遗传算法求解TSP问题为例，深入讨论了各个遗传算子的程序实现，并通过分析实验数据，得到各个遗传算子在搜索寻优过程中所起的作用，最后指出了用Matlab语言编程同用其它高级程序语言编程的差异所在.

[关键词] Matlab语言；遗传算法；程序设计；TSP

[中图分类号] O 224 [文献标识码] A

0 引言

当前，有关遗传算法的研究方兴未艾，由于其基础理论尚未有突破性进展，因此要想知道一个新的方案效果到底如何，一个有效途径就是利用计算机进行模拟．目前，人们大多采用C语言编写这类程序[1]．C语言作为一种高效的编程语言有它的许多优点，但对于编写遗传算法程序来说，笔者认为，Matlab语言比它更为合适．遗传算法诸多算子（如选择、交叉、变异等），都是针对所谓染色体进行的，而染色体实质上是一个向量，可将其看成一个$1\times n$的矩阵，因此这些算子实质上是一些矩阵运算，而Matlab的基本数据单元就是一个维数不加限制的矩阵，在这种编程环境下，无需用户考虑大量的有关矩阵的运算该采用何种算法等低层问题，更不必深入了解相应算法的具体细节，因而对用户算法语言方面的要求十分宽松，用它编写遗传算法程序，比用C等其它高级语言要简单、灵活、快捷，程序篇幅也将缩小许多．

1 TSP问题的遗传算法程序设计

TSP (Traveling Salesman Problem)问题是一个NP完全问题，近几年来，基于遗传算法求解TSP问题的研究相当活跃[1~3]，在遗传算法研究中，TSP问题已被广泛地用于评价不同的遗传操作及选择机制的性能，因此，将求解TSP问题的遗传算法编制成程序进行计算机模拟有

[收稿日期] 2000－05－06
[基金项目] 福建省自然科学基金项目（F9810013）.
[作者简介] 储理才（1969－），男，讲师，硕士，从事遗传算法与模糊系统的研究.

重要意义．TSP问题的描述十分简单，简言之，就是寻找一条最短的遍历 n 个城市的最短路径，或者说搜索自然数子集 $X = |1,2,\cdots,n|$（X 的元素表示对 n 个城市的编号）的一个排列 $\pi(X) = |V_1, V_2, \cdots, V_n|$，使

$$T_d = \sum_{i=1}^{n-1} d(V_i, V_{i+1}) + d(V_1, V_n)$$

取最小值，式中的 $d(V_i, V_{i+1})$ 表示城市 V_i 到城市 V_{i+1} 的距离．

笔者对遗传算法方法求解TSP问题，用Matlab语言作过实验研究．现就笔者所用的遗传算法框架，介绍其算法的程序实现过程．

1.1　编码与适应值函数

以城市的遍历次序作为遗传算法的编码，适应度函数取为哈密顿圈的长度的倒数．在群体初始化，交叉操作和变异操作中均考虑TSP问题的合法性约束条件（即对所有城市要作到不重不漏）．

1.2　控制参数、城市位置及距离矩阵

为见其名知其义，分别用 lchrom、popsize、maxgen、pcross、pmutation 表示染色体长度（城市数）、群体规模、遗传代数、交叉概率、变异概率．在平面上随机生成 lchrom 个点（数对）表示城市位置，再计算出两城市之间的距离，构成距离矩阵 distmatrix，其 Matlab 程序为

```
%  xy是一个lchrom行，2列的矩阵，每一行表示一个城市的位置．
X= []; Y= []; n=1;
while n<=lchrom,
    a=rand*1.4+rand;
    X=[X;real(a)];Y=[Y;imag(a)];
    n=n+1;
end;
xy=[x  y];
%  计算任意两城市间的距离，构造距离矩阵．
distmatrix=zeros(lchrom);
for count1=1:lchrom,
  for count2=1:count1,
    x1=xy(count1,1); y1=xy(count1,2);
    x2=xy(count2,1); y2=xy(count2,2);
    distmatrix(count1,count2)=sqrt((x1-x2)^2+(y1-y2)^2);
    distmatrix(count2,count1)=distmatrix(count1,count2);
  end;
end;
```

1.3　生成初始群体

系统内建函数randperm(n)随机产生一个由自然数1到 n 组成的全排列，调用它来产生初始群体．

```
    n=1; pop=zeros(popsize);
while n<=popsize,
    pop(n,:)=randperm(lchrom);
```

```
    n = n + 1;
end;
```

1.4 计算个体对应的哈密顿圈长和适应值的自定义函数

```
%  p表示一个染色体（遍历城市的次序），函数返回该遍历次序所对应的哈密顿圈长和适应值
function[Length,Fitness] = PathLenFit(p,distmatrix);
npts = size(p,2);
Length = sum(distmatrix([(p-1) * npts + p([end 1:(end-1)])]));
Fitness = 1/Length;
```

1.5 选择、交叉、变异

采用赌轮选择机制，在每一代运算过程中，个体被选中的概率与其在群体中的相对适应度成正比.

采用如下的交叉方法：

1）随机在串中选择一个交配区域，如两父串选定为

oldp1 = 1 2 | 3 4 5 6 | 7 8 9，　　oldp2 = 9 8 | 7 6 5 4 | 3 2 1

2）将 oldp2 的交配区域加到 oldp1 的前面，oldp1 的交配区域加到 oldp2 的前面，得

oldp12 = 7 6 5 4 | 1 2 3 4 5 6 7 8 9，　　oldp21 = 3 4 5 6 | 9 8 7 6 5 4 3 2 1

3）对 oldp12，oldp21 自交配区域后依次删除与交配区相同的城市码，得到最终的两子串为

newp1 = 7 6 5 4 1 2 3 8 9，　　newp2 = 3 4 5 6 9 8 7 2 1

对应的程序为

```
function [newp1,newp2] = crossover(oldp1,oldp2);
crossj1 = floor((length(oldp1) - 1) * rand) + 1;
crossj2 = floor((length(oldp1) - 1) * rand) + 1;
minjcross = min(crossj1,crossj2);
maxjcross = max(crossj1,crossj2);
segment1 = oldp1(minjcross:maxjcross);
segment2 = oldp2(minjcross:maxjcross);
oldp12 = eliminate(oldp1,segment2);                    %其中 eliminate 是自定义函数
oldp21 = eliminate(oldp2,segment1);
newp1 = [segment2 oldp12];
newp2 = [segment1 oldp21];
```

自定义函数 elim = eliminate (x, y)，其功能为删除 x 向量中与 y 向量元素相同的元素，例如：x = [1 2 5 3 4]，y = [4 2]，则 eliminate (x, y) 返回 [1 5 3]

```
function elim = eliminate(x,y);
for n = 1:length(y),
    x = x(find(x~ = y(n)));
end;
elim = x;
```

变异策略采取随机选取两个点，将其对换位置.

```
%minmutation，maxmutation 是两个变异点的位置
```

```
pold([minmutation maxmutation]) = pold([maxmutation minmutation]);
pnew = pold;
```

1.6　进化逆转操作

为改善遗传算法的局部搜索能力，在选择、交叉、变异之后引进连续多次的进化逆转操作．所谓逆转，如染色体（1－2－3－4－5－6）在区间 2－3 和区间 5－6 两处发生断裂，断裂片段又以反向顺序插入，于是逆转后的染色体变为（1－2－5－4－3－6）．这里的“进化”是指逆转算子的单方向性，即只有经逆转后，适应值有提高的才接受下来，否则逆转无效．

可见，这里的进化逆转操作可以认为是一种朝着改进的方向的特殊的变异操作，它具有变异操作的某些效果，但它并不能完全替代变异操作．因为变异操作在可行解空间中动作范围较宽，它能使迭代过程突破局部最优圈而跳到另一个搜索空间，同变异算子相比较，它类似于一种基于邻域的局部搜索，起到改进遗传算法局部搜索能力的作用．

逆转操作可由下述语句实现：

```
%mininverse，maxinverse 是两断裂点
order = 1:lchrom;
order(mininverse:maxinverse) = order(maxinverse: -1:mininverse);
pnew = pold(order);
```

1.7　算法的流程

一个包含选择、交叉、变异及进化逆转的遗传算法流程如下所示．

1）确定遗传参数，产生初始群体；

2）计算适应度；

3）群体更新：选择、交叉、变异、进化逆转；

4）满足终止条件否？否，转 2；是，转 5

5）输出结果，结束．

另外，Matlab 还具有强大的图形功能，并且提供了丰富的图形界面函数，当前最佳路径可以在屏幕上很轻松地显示出来，从而动态地观察当前最佳路径的演变情况．限于篇幅，程序就不再列出了．

2　求解 TSP 问题的遗传算法中各类遗传算子的作用研究

本文在应用 Matlab 编制遗传算法程序用于求解 TSP 问题的过程中，发现逆转算子起着特殊的作用．为了便于研究，本文将 100 个城市设置成排列整齐的 10 行 10 列的方阵，行距、列距皆为 25．显然此问题的最短哈密顿圈长为 2500，其次为 2520.71 ．考虑三种策略：

策略 1：有选择、交叉、变异算子，无进化逆转算子；

策略 2：无交叉、变异算子，有选择、进化逆转算子；

策略 3：选择、交叉、变异、进化逆转算子都有．

取遗传代数为 2000，群体规模为 100，交叉概率、变异概率分别取为 0.9，0.05（就策略 1 策略 3 来说），每种策略各进行 10 次实验，结果如表 1，2 所示．

表1 三种策略的收敛情况

收敛情况	策略1	策略2	策略3
收敛到最优解次数	0	0	7
收敛到次优解次数	0	0	3
最短哈密顿圈长(10次试验的平均数)	7391.16	2581.79	2506.21

表2 策略2、策略3收敛时的代数

策略	收敛代数									
策略2	331	247	500	528	382	533	479	439	435	226
策略3	1145	1066	933	1814	1510	1217	965	1465	1168	1925

从表1，2可看出，只有选择、交叉、变异算子，而无逆转算子（策略1）时，不易收敛，得到的解与最优解相比，差距很大．这说明选择、交叉、变异算子搜索最优解的能力较差；而只有逆转算子（策略2），则容易导致过早收敛到局部最优解；只有将两者结合起来（策略3），才能起到很好的效果．

在遗传算法中，交叉算子被认为是起核心作用的遗传操作，通过交叉，遗传算法的搜索能力会得以飞跃提高[1]．那么为什么在求解TSP问题中，交叉算子所显示的搜索能力反而不及逆转算子呢？究其原因，是由于TSP问题本身以及编码所引起的．在本文的编码中．每一个染色体即对应一个TSP环游，如果染色体码串的顺序发生变化，则环游路径也随之改变，因此，TSP问题解关键地方就是码串的顺序．对照文中的交叉算子．可以发现，纵使两个亲代完全相同，通过交叉，仍然会产生不同于亲代的子代，且子代的码串排列顺序与亲代有较大的差异．交叉算子的这种变异效果所起的作用有两个方面，一方面它能起到维持群体内一定的多样性的作用，避免陷入局部最优解；但是另一方面，它却不利于子代继承亲代的较多信息，特别是当进化过程进入到后期，群体空间中充斥着大量的高适应度个体，交叉操作对亲代的较优基因破坏很大，使子代难以继承到亲代的优良基因，从而使交叉算子的搜索能力大大降低．

同交叉算子相比较、逆转算子更能够使子代继承亲代的较多信息．假设码串为a b c d e f g,在b和c，e和f之间发生两处断裂再逆转插入，则新码串为a b e d c f g，此时子代中a b段，e d c段，f g段与亲代对应片段顺序完全一样（e d c与c d e顺序对于环游路径长度来说是等价的），只是在断裂点的两端环游次序发生了变化，而且本文中逆转是单方向的，即只接受朝着好的方向的逆转，因此它搜索最优解的能力强于交叉算子．基于同样的分析，在遗传算法的后期，用连续多次的单方向的变异也可以达到同样的效果．计算机模拟也证实了这一点．

3 应用Matlab编程应注意的问题

从程序设计的方法看，在Fortran和C等高级语言中，编制的程序是过程程序(Procedural Program)．编制程序时总要精心地设置一些局部变量、循环控制结构和子程序等，过程程序的优点是有利于充分发挥个人的智慧和才能．

在Matlab中，编制的程序以函数程序(Functional Program)为主．编程的重点放在要解决

的问题的要求上，编制程序就是调用和组合相应的系统函数、实际上，这是 Matlab 与常见程序设计语言的一个重要差别．使用函数而不用编程，有效地减少了嵌套调用的冗余计算，简化或改变了递归的形式，使用户脱离了编程中细微而又繁琐的一面、如果找不到解决问题的系统函数，则必须编制函数或程序，但在可能的情况下、应当尽量考虑使用系统函数，尤其是对矩阵或数组的操作、

Matlab 的基本数据单元就是一个维数不加限制的矩阵，系统提供了大量处理矩阵的函数，使人可以整体性地考虑和描述计算问题．例如要给一个矩阵 M 的每一个元素加 1，只要写一句 "$M = M + 1$" 就可以了、将数据成批进行处理；而在其它程序语言里需要用一个循环结构实现这个工作，类似的例子在前文中也大量存在．熟悉其它编程语言的人初用 Matlab，往往不自觉地用老的方法考虑问题，总是想到要一点一点地处理数据，这样做往往不能发挥 Matlab 的优势，在编制程序时会花费太多不必要的时间，而且运算速度还很低、因此在运用 Matlab 编程时，整体性地考虑问题是非常重要的．

[参 考 文 献]

[1] 陈国良，王煦法、庄镇泉，等．遗传算法及其应用 [M]．北京：人民邮电出版社、1996．

[2] 赵赫、杜端甫．遗传算法求解旅行推销员问题时算子的设计与选择 [J]．系统工程理论与实践、1998、18 (2)：62-65．

[3] 刘克胜，邵华，曹先彬，等．基于免疫算法的 TSP 问题求解 [A]．1999 中国智能自动化学术会议论文集（下册）[C]．北京：清华大学出版社，1999．1168-1174．

[4] 张延华，许阳明．高技术计算环境——Matlab 使用指南 [M]．北京：科学技术文献出版社，1998．

Genetic Algorithm Programming Based on Matlab and TSP Solving

CHU Li-cai

(Dept. of Basic Courses、Jimei University，Xiamen 361021，China)

Abstract: Firstly，this paper analyses the advantages of using Matlab Language to devise genetic algorithm program．Secondly，it probes into the realization of genetic operator program through TSP solving by genetic algorithm，finding the function of each genetic operator by analyzing the experimental data．Finally，it points out the differences between programming by Matlab language and programming by other program languages．

Key words: Matlab；genetic algorithm；programming；TSP