



## 遗传算法的 TSP（旅行商问题）的求解

学生姓名：宗满意

指导老师：乔立红

### 摘 要

TSP 问题是典型的NP完全问题, 遗传算法是求解NP完全问题的一种常用方法。本文针对解决TSP 问题, 在MATLAB中用遗传算法施行对TSP问题进行了求解, 进行了选择、交叉和变异算子进行了算法设计, 最后在MATLAB软件上进行编程实现。最后探讨了遗传算法解决旅行商问题自身具备的特点[1]。

**关键词：**遗传算法；TSP问题；MATLAB软件



---

## SOLVING TSP (Travelling Salesman Problem ) BASED ON GENETIC ALGORITHM

Author : Zong Man-yi

Tutor : Qiao Li-hong

### Abstract

TSP( Traveling Salesman Problem) is a typical NP complete problem ,genetic algorithm is the perfect method for solving NP complete problem. This paper use genetic algorithm in the MATLAB software to solve the a typical TSP problem . It probes into the realization of genetic operator program through TSP solving by genetic algorithm , design the each function of each genetic operator(select, intercross, mutate). Finally ,We programm in Matlab language and discuss the characteristic of genetic algorithm in solving TSP

**Key words :** genetic algorithm; TSP Matlab ;



目录

引言..... 4

1 GA 概述..... 4

2 旅行商问题(TSP) ..... 4

3 用遗传算法解决旅行商问题 ..... 5

4 论文的主要构成 ..... 5

遗传算法的设计..... 6

1 问题分析..... 6

2 总体设计 ..... 7

3 详细设计 ..... 8

3.1 编码与随机和初始群体生成.....8

3.2 城市位置及距离矩阵和适应度函数 .....8

3.4 选择.....9

3.4 交叉.....9

3.5 变异..... 10

3.6 群体的跟新和终止条件 ..... 10

MATLAB 编程验证 ..... 11

1MATLAB 计算..... 11

2 算法分析优化..... 13

结论..... 15

参考文献..... 16



## 引言

### 1 GA 概述

遗传算法 (Genetic Algorithm, GA) 是由美国 J. Holland 教授提出的一类借鉴生物界自然选择和自然遗传机制的随机化搜索算法。它起源于达尔文的进化论, 是模拟达尔文的遗传选择和自然淘汰的生物进化过程的计算模型。其主要特点是群体搜索策略和群体中个体之间的信息交换, 搜索不以梯度信息为基础。它尤其适用于处理传统搜索方法难于解决的复杂和非线性问题, 可广泛应用于组合优化、机器学习、自适应控制、规划设计和人工生命等领域。作为一种全局优化搜索算法, 遗传算法以其简单通用、鲁棒性强、适于并行处理以及应用范围广等特点, 使其成为 21 世纪智能计算核心技术之一。进入 80 年代, 遗传算法迎来了兴盛发展时期, 无论是理论研究还是应用研究都成了十分热门的话题[2]。

### 2 旅行商问题 (TSP)

一个有名的组合优化问题就是旅行商问题 (Travelling Salesman Problem, TSP)。TSP 问题是典型的、易于描述却难以处理的组合优化问题。由于遗传算法方法的本质是处理复杂问题的一种鲁棒性强的启发性随机搜索算法, 故而利用遗传算法求解这类问题是有效的。

假设平面上有  $n$  个点代表  $n$  个城市的位置, 寻找一条最短的闭合路径, 使得可以遍历每一个城市恰好一次。这就是旅行商问题。旅行商的路线可以看作是对  $n$  个城市所设计的一个环形, 或者是对一系列  $n$  个城市的排列。由于对  $n$  个城市所有可能的遍历数目可达  $(n-1)!$  个, 因此解决这个问题需要  $O(n!)$  的计算时间。假设每个城市和其他任一城市之间都以欧氏距离直接相连。也就是说, 城市间距可以满足三角不等式, 也就意味着任何两座城市之间的直接距离都小于两城市之间的间接距离。是一个典型的优化组合问题[3]。



## 3 用遗传算法解决旅行商问题

目前针对TSP问题有许多种解法，较为常用的算法有神经网络法、列表寻优法、二叉树描述法、模拟退火法和遗传算法等等。遗传算法是近几年发展起来的一种崭新的全局优化算法，它借用了生物遗传学的观点，通过选择、遗传、变异和免疫等作用机制，使每个个体的适应性提高。由于其全局搜索的特性，遗传算法在解决TSP问题中有着其他算法所没有的优势。提出的TSP问题包括31个城市的编号和各自的位置。本文针对这个TSP问题进行了遗传算法的编程和求解。

## 4 论文的主要构成

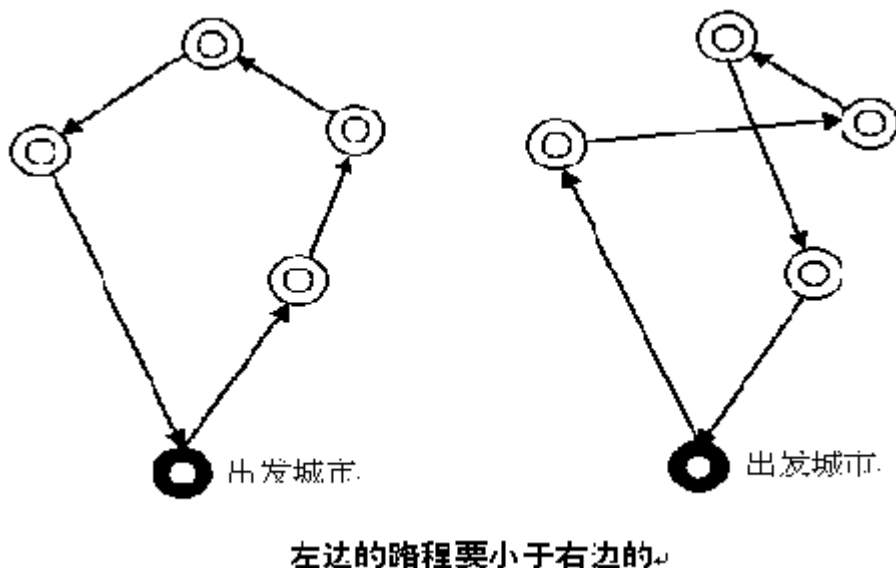
本论文在 MATLAB 中用遗传算法施行对 TSP 问题进行了求解，进行了选择、交叉和变异、免疫等算子进行了算法设计，最后在 MATLAB 软件上进行编程实现。

## 遗传算法的设计

### 1 问题分析

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 $X = \{1, 2, \dots, n\}$ （ $X$ 的元素表示对 $n$ 个城市的编号）的一个排列

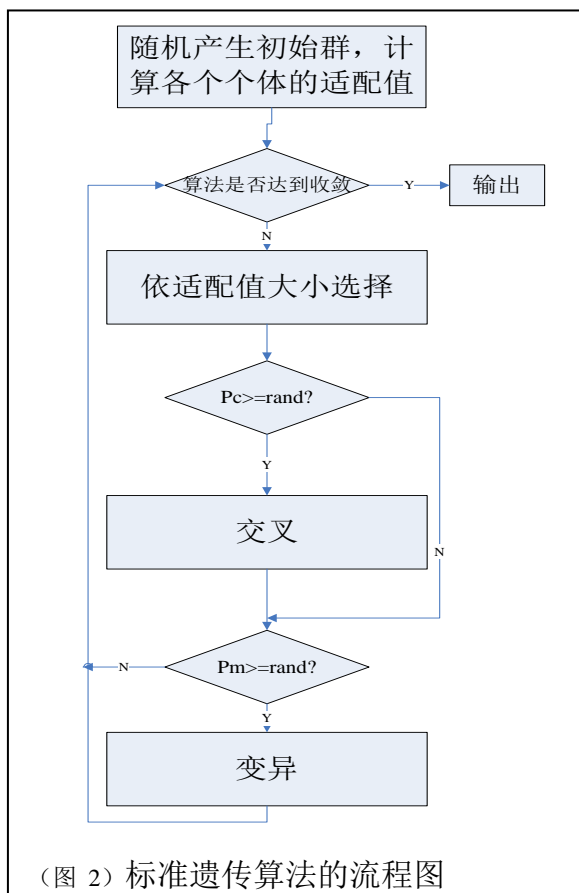
$\pi(X) = \{V_1, V_2, \dots, V_n\}$ ，使 $len = \sum d(V_i, V_{i+1}) + d(V_1, V_n)$ 取最小值，式中的 $d(V_i, V_{i+1})$ 表示城市 $V_i$ 到城市 $V_{i+1}$ 的距离。



(图 1) [3]

本实例中设定了我国31个省会城市，当一个旅行商遍历了所有城市后回到出发城市，求其最短路径的走法。一个最容易想到的方法是利用排列组合的方法把所有的路径都计算出来，并逐一比较，选出最小的路径。虽然该方法在理论上是可行的，但路径的个数与城市的个数成指数增长，当城市个数较大时，该方法的求解时间是难以忍受的，甚至是不可能完成的，31个城市所有路径个数是 $30!$ 个。由于其全局搜索的特性，遗传算法在解决TSP问题中有着其他算法所没有的优势。本文针对这个TSP问题进行了遗传算法的编程和求解。

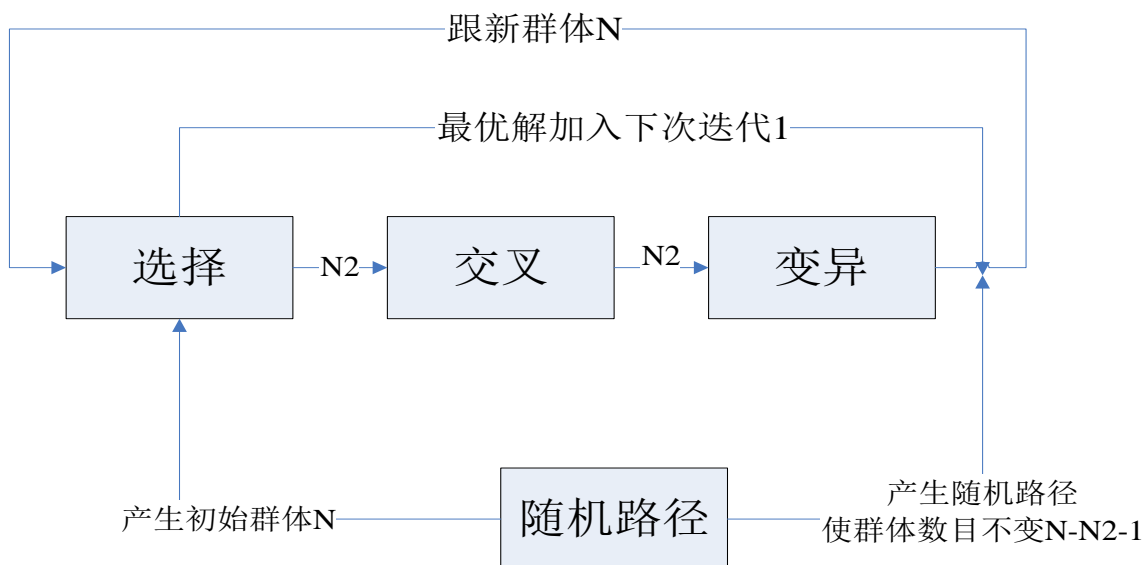
## 2 总体设计



标准的遗传算法包括群体的初始化，选择，交叉，变异操作。其主要步骤可描述如下[1]：

- (1) 随机产生一组初始个体构成的初始种群，并评价每一个个体的适配值。
- (2) 判断算法的收敛准则是否满足。若满足输出搜索结果；否则执行以下步骤。
- (3) 根据适配值大小以一定方式执行选择操作。
- (4) 按交叉概率 $P_c$ 执行交叉操作。
- (5) 按变异概率 $P_m$ 执行变异操作。
- (6) 返回步骤（2）。

### 本TSP问题的遗传算法总体设计





(图3) 总体设计

说明: (1) 本算法的判断结束准则是固定指定了迭代的次数当算法达到迭代次数时, 算法结束, 输出当前的最优解

(2) 在根据适配值计算并选择的时候, 记录下来的当前最优值, 在变异后加入跟新的群体, 保证新的迭代循环中TSP解越来越好 (不会变差)。

(3) 在选择的一种操作是拿最优的K个替换最差的K个个体, 本例是按适配值选择, 并使群体数目变少, 当每次变异操作后, 产生随机路径补充群体是群体数目不变, 再次循环, 一定程度上防止因初始群体的选择问题而陷入局部最优。

### 3 详细设计

#### 3.1 编码与随机和初始群体生成

给所有城市编码, 以城市的遍历次序作为遗传算法的编码。在MATLAB中使用 `randperm(N)` 产生一个  $1 \times N$  的矩阵 ( $N$  为所有城市的个数, 本例中为31) 为一个随机路径。利用  $n \times N$  矩阵存储  $n$  个随机群体产生初始群体。

#### 3.2 城市位置及距离矩阵和适应度函数

城市的位置为编译前指定, 也可以使用随机生成的坐标参数。距离矩阵使用一个  $N \times N$  矩阵  $D$  存储,  $D(i, j)$  代表城市  $i$  和城市  $j$  之间的距离。

$$D(i, j) = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}$$

在该问题的求解中, 用距离的总和来衡量适应度, 距离的总和越大, 适应度越小, 进而探讨求解结果是否最优。

每个个体 (每条路径距离) 总合计算公式为:

```
len=D(1,N);
```

```
for i=1:(N-1)
```

```
    len=len+D(i,i+1);
```

```
end
```

`len` 纪录总路径格式  $n \times 1$  `len(i)` 代表第  $i$  个个体 (路径) 距离总合。



### 3.4 选择

选择操作是为了避免有效基因的损失，使高性能的个体得以更大的几率生存，从而得到全局收敛性和计算效率。

本例中使用的适配值函数为

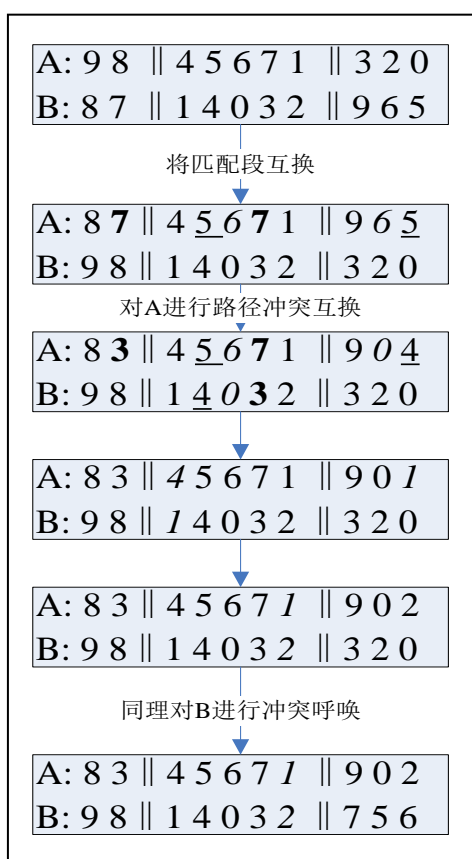
$$\text{fitness}(i, l) = (1 - ((\text{len}(i, l) - \text{minlen}) / (\text{maxlen} - \text{minlen} + 0.0001)))^m;$$

% maxlen , minlen 为最大和最短路径

利用 fitness > rand 选择个体，将个体，将较小路径(适应度较大)个体选择下来。但这种算法的群体的个体数目变小并且较优的个体个数较少，可能收敛的数目变慢，在算法调试的过程中证明了这一点[4]。

### 3.4 交叉

交叉操作用于组合出新的个体，再借空间中进行有效的搜索，同时降低有效模式的破坏概率[1]。



本实例中交叉采用部分匹配交叉策略,其基本实现的步骤是:

步骤1: 随机选取两个交叉点

步骤2: 将两交叉点中间的基因段互换

步骤3: 将互换的基因段以外的部分中与互换后基因段中元素冲突的用另一父代的相应位置代替，直到没有冲突。

本例中路径实例如图交叉点为2, 7, 交换匹配段后A中冲突的有7、6、5, 在B的匹配段中找出与A匹配段中对应位置的值7-3, 6-0, 5-4, 继续检测冲突直到没有冲突。对B做同样操作, 得到最后结果。

(图 4)



### 3.5 变异

变异操作通过随机改变个体中某些个体的某些基因而产生新个体，有助于增加种群的多样性，避免早熟收敛（非全局最优）。

本例中变异操作使用互换操作算子（SWAP），即随机交换染色体中两个不同基因编码的位置，SWAP 相对于 INV（逆序操作）和 INS（插入操作）更有利于算法的大范围搜索。

对于一个 31 路径的染色体的变异操作，依变异概率确定是否变异，随机选择路径上的两个城市进行交换。例如 变异交换位置为 2 和 8



### 3.6 群体的跟新和终止条件

为保证计算随迭代次数的增加，最优解越来越好，本例种每次变异后将每次的上次迭代最优解加入群体，防止因交叉或变异而是最优解失去，出现退化现象。

为保持群体数目不变，变异后产生随机解加入群体（本算法群体数目在选择中可能发生减少）。

终止条件最常用的有事先给定一个最大进化步数，或者是判断最优化值是否连续若干步没有明显变化两种。本实例中只选择了第一种方法[6]。

## MATLAB 编程验证

### 1 MATLAB 计算

#### 1 初始值 (图 5)

种群个数 $n=100$ ;迭代次数 $C=1000$ ;

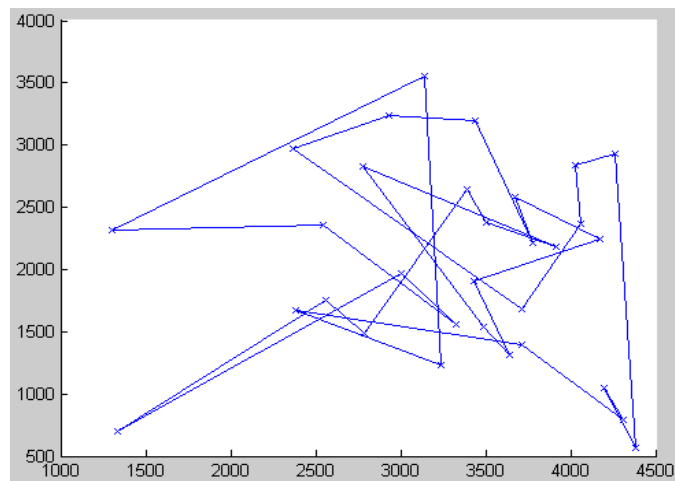
交叉概率 $P_c=0.9$ ;变异概率 $P_m=0.2$ ;

初始种群中的一个随机值:

$R = [ 29 \ 13 \ 15 \ 16 \ 4 \ 17 \ 22 \ 12 \ 2 \ 18$   
 $31 \ 5 \ 25 \ 26 \ 11 \ 6 \ 28 \ 3 \ 14 \ 30 \ 20 \ 1 \ 9$   
 $21 \ 10 \ 24 \ 8 \ 7 \ 23 \ 19 \ 27 ]$

$Rlength = 4.4308e+004$

迭代 1000 次后路径如图  $Rlength = 3.0902e+004$  发现算法收敛很慢



#### 2 初始值 (图 6)

种群个数 $n=100$ ;迭代次数 $C=2000$ ;

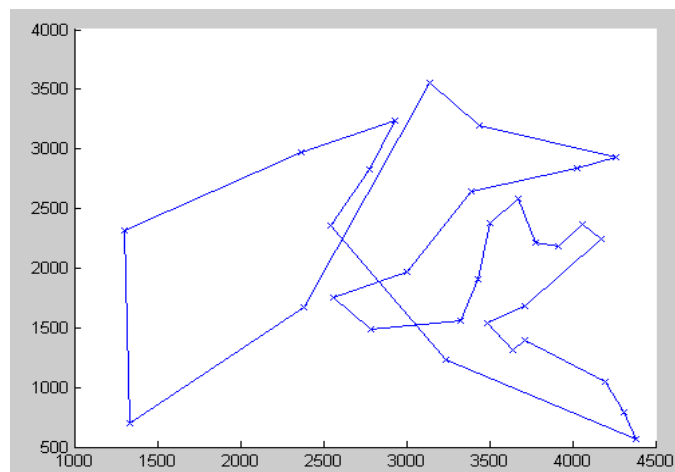
交叉概率 $P_c=0.9$ ;变异概率 $P_m=0.2$ ;

初始种群中的一个随机值:

$Rlength = 4.3506e+004$

迭代 2500 次后路径如图

$Rlength = 1.9497e+004$  趋向于收敛



#### 3 初始值 (图 7)

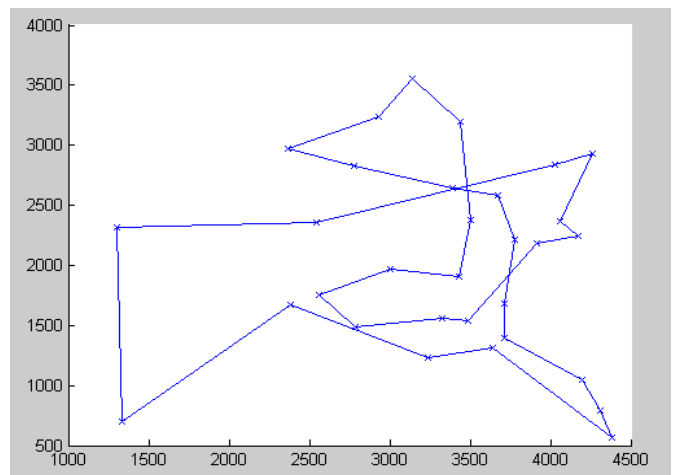
种群个数 $n=100$ ;迭代次数 $C=2500$ ;

交叉概率 $P_c=0.9$ ;变异概率 $P_m=0.2$ ;

初始种群中的一个随机值:

$Rlength = 4.3506e+004$

迭代 2500 次后路径如图  $Rlength = 1.8754e+004$  趋向于收敛





4初始值: (图 8)

种群个数 $n=100$ ;迭代次数 $C=3000$ ;

交叉概率 $P_c=0.9$ ;变异概率 $P_m=0.2$ ;

$Rlength = 4.4721e+004$

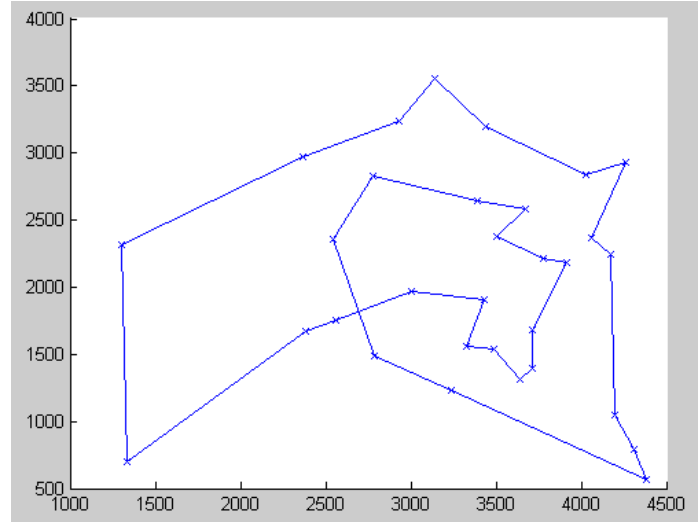
迭代 3000 次路径如图  $Rlength = 1.6957e+004$

次序为  $R = 29\ 13\ 7\ 10\ 9\ 8\ 3\ 18\ 22\ 21$

$26\ 28\ 27\ 31\ 1\ 15\ 14\ 12\ 11\ 23\ 6\ 5$

$2\ 4\ 16\ 17\ 19\ 24\ 20\ 25\ 30$

路径较小可能已经为最优解



5初始值: (图9)

种群个数 $n=100$ ;迭代次数 $C=3500$ ;

交叉概率 $P_c=0.9$ ;变异概率 $P_m=0.2$ ;

$Rlength = 4.4721e+004$

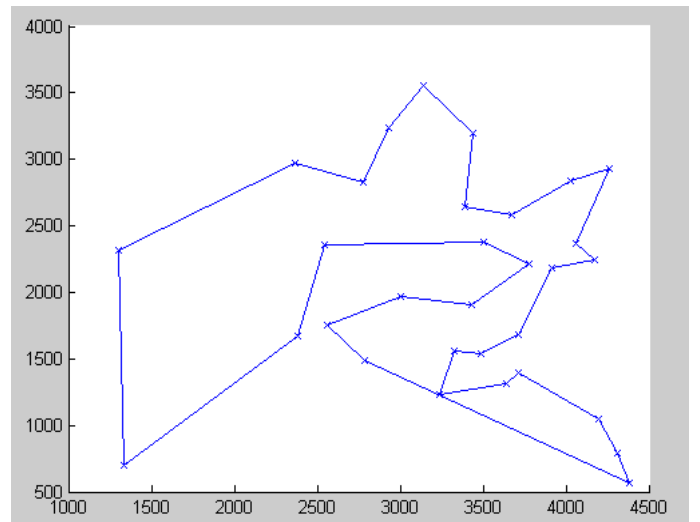
迭代 3500 次路径如图  $Rlength = 1.7093e+004$

次序为  $R = 25\ 20\ 21\ 22\ 18\ 3\ 17\ 16\ 5$

$6\ 7\ 2\ 4\ 8\ 9\ 10\ 13\ 12\ 11\ 23\ 19\ 24$

$29\ 14\ 15\ 1\ 31\ 30\ 27\ 28\ 26$

路径较小可能已经为最优解



城市 坐标矩阵为 $a=[1304\ 2312;3639\ 1315;4177\ 2244;3712\ 1399;3488\ 1535;3326$

$1556; 3238\ 1229;4196\ 1044;4312\ 790;4386\ 570;3007$

$970;2562\ 1756; 2788\ 1491;2381\ 1676;1332\ 695;3715$

$1678;3918\ 2179;4061\ 2370;3780\ 2212;3676\ 2578;4029$

$2838;4263\ 2931;3429\ 1908;3507\ 2376; 3394\ 2643;3439$

$3201;2935\ 3240;3140\ 3550;2545\ 2357;2778\ 2826; 2370$

$2975]$ ;

可以认为该算法在迭代 3000 左右时,已经发生了收敛,对于本例给出一个解为  $R = 29$

$13\ 7\ 10\ 9\ 8\ 3\ 18\ 22\ 21\ 26\ 28\ 27\ 31\ 1\ 15\ 14\ 12\ 11\ 23\ 6\ 5\ 2\ 4\ 16\ 17\ 19\ 24\ 20\ 25$

30

## 2 算法分析优化

1. 选择操作。本例中算法收敛较慢，改进选择操作如下，选定  $K$  个最大适应度的个体，用来代替适应度最小的  $K$  个个体，保持群体的数目不变。此方法可以使群体的平均适应度大幅提高，可能使其收敛速度加快。

试验中使用该种选择操作，发现速度有加快迹象

6初始值：（图 10）

种群个数 $n=100$ ；迭代次数

$C=1000$ ；

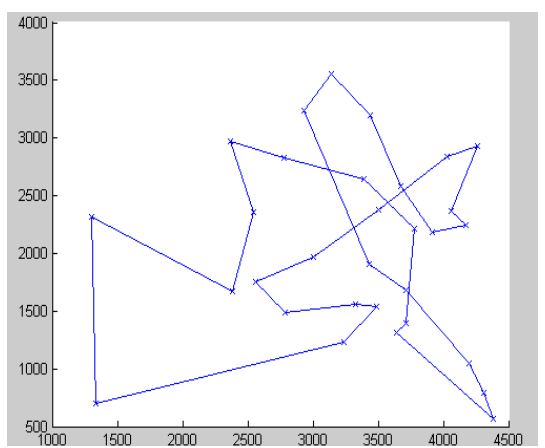
交叉概率 $P_c=0.9$ ；变异概率

$P_m=0.2$ ；

$Rlength=1.9575e+004$

与图6相比收敛速度又加快迹  
象

（本次迭代次数为 1000 次）



7初始值：（图 11）

种群个数 $n=100$ ；迭代次数

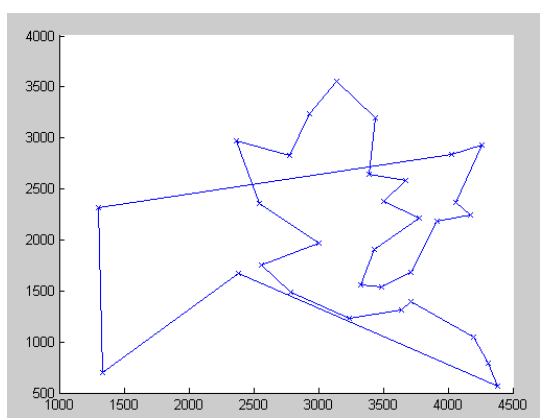
$C=4000$ ；

交叉概率 $P_c=0.9$ ；变异概率

$P_m=0.2$ ；

$Rlength=1.8588e+004$

与图 8, 9 相比收敛结果基本正  
确。证明优化方式正确。



3. 每次交叉或变异操作以后，检验子代的适应度。如果子代发生退化，取消操作，用父代代替子代。这种方法可能带来的问题有①虽然收敛迭代次数可能减少，但每次运算的计算量增加，增加了运算的时间②不利于种群的多样化，可能收敛解非最优解。解决方法根据某一较小概率 0.1-0.2 有选择的进行检验和替换。

本例中算法已经进行了设计

4. 免疫遗传算法。免疫遗传算法在变异以后加入免疫算子，可以大幅减少迭代次数。针对 TSP 问题，免疫遗传算子设计为先用一个  $n \times 2$  的矩阵存储每一个顶点（城市）与其相应的最近城市的编号。如阿  $a[i,1]$  的最近城市  $a[i,2]$ 。根据 TSP 问题而言，



在最终的解决方案中，即在最佳路径中必然包括（或载很大程度上包括）了相邻城市间距离的最短路径。将城市  $a[i,2]$  移到  $a[i,1]$  之后，进行免疫检测，如果个体发生退化，取消免疫。免疫遗传算法可以大幅加快迭代的收敛速度减少迭代次数。本实例中没有进行免疫设计[4]。

5. 终止条件。终止条件最常用的有事先给定一个最大进化步数，或者是判断最优化值是否连续若干步没有明显变化两种。本实例中只选择了第一种方法。可以设定一百次记录一次最优解，当连续三次的解不变（或相差不大）输出迭代结果。本实例中没有设计。



---

## 结论

论文用遗传算法对 TSP 问题进行了求解，熟悉遗传算法地算法流程，证明了遗传算法在求解 TSP 问题时，具有可行性，MATLAB 在进行算法优化编程时具有一定的优势。遗传算法在设计过程中要照顾好两个原则①子代要具有父代优良的基因信息即继承性②子代要保持群体的多样性，即变异。虽然这两个原则在设计过程中经常会出现冲突，我们必须统筹的把握。既要实现算法的快速运算，又要实现解的最优。本论文认为对标准遗传算法有进行改经的必要，以提高其求解能力。



---

## 参考文献

- [1] 王凌著, 智能优化算法及其应用, 清华大学出版社, 2001
- [2] 陈国良, 王煦法, 庄镇圈等. 遗传算法及其应用, 北京: 人民邮电出版社, 1996
- [3] 谢胜利等, 基于遗传算法的旅游商问题求解, 温州师范学院学报; 2002. 1. 23
- [4] 阮怀忠, 张建中, 基于改进遗传算法的 TSP 问题求解, 安徽建筑工业学院学报 2003. 11
- [5] 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
- [6] 高经纬, 张煦等, 求解TSP 问题的遗传算法实现, 计算机时代, 2004年第2 期