

实验报告

———模拟退火算法

姓名：陈恩婷

学号：19335015

日期：2020/10/11

摘要：本次实验通过实现多种局部搜索策略与模拟退火算法并进行可视化的 Matlab 程序来求解 tsp 问题，探究了不同的局部搜索策略和模拟退火算法的性能。总的来说，局部搜索策略中 inversion mutation 性能最好，模拟退火算法较单纯的局部搜索策略性能有所提升。

1. 引言

(1) 问题描述

选一个大于 100 个城市数的 TSP 问题，

1. 采用多种邻域操作的局部搜索 local search 策略求解；
2. 在局部搜索策略的基础上，加入模拟退火 simulated annealing 策略，并比较两者的效果；
3. 要求求得的解不要超过最优值的 10%，并能够提供可视化，观察路径的变化和交叉程度。

(2) 问题及方法背景介绍

旅行商问题（英语：Travelling salesman problem, TSP）是组合优化中的一个 NP 困难问题，在运筹学和理论计算机科学中非常重要。问题内容为“给定一系列城市 and 每对城市之间的距离，求解访问每一座城市一次并回到起始城市的最短回路。”

在计算机科学中，**局部搜索**是解决最优化问题的一种元启发式算法。局部搜索从一个初始解出发，然后搜索解的邻域，如有更优的解则移动至该解并继续执行搜索，否则返回当前解。局部搜索的优点是简单、灵活及易于实现，缺点是容易陷入局部最优且解的质量与初始解和邻域的结构密切相关。常见的改进方法有模拟退火、禁忌搜索等。

模拟退火法：退火是冶金专家为了达到某些特种晶体结构重复将金属加热或冷却的过程，该过程的控制参数为温度 T 。这种思想应用于许多优化问题就产生了模拟退火算法，**模拟退火法的基本思想是，在系统朝着能量减小的趋势这样一个变化过程中，偶尔允许系统跳到能量较高的状态，以避开局部极小点，最终稳定到全局最小点。**

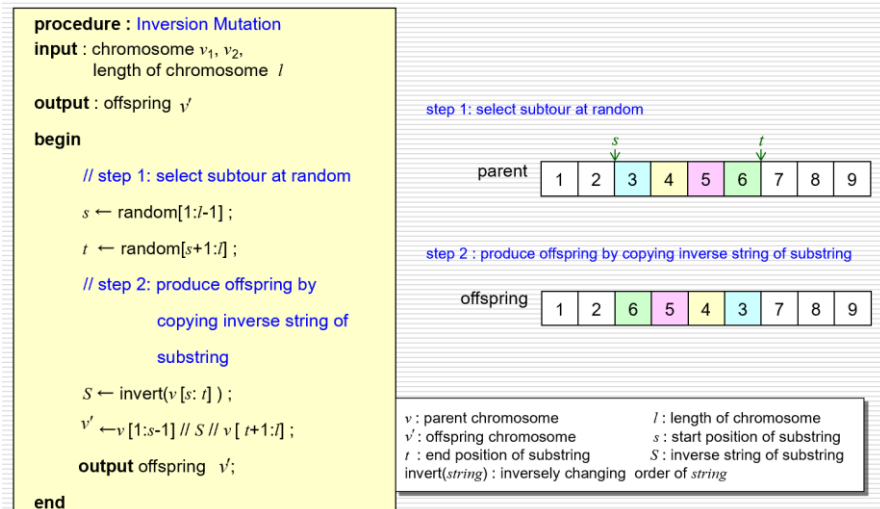
2. 实验过程

程序主要流程如下：

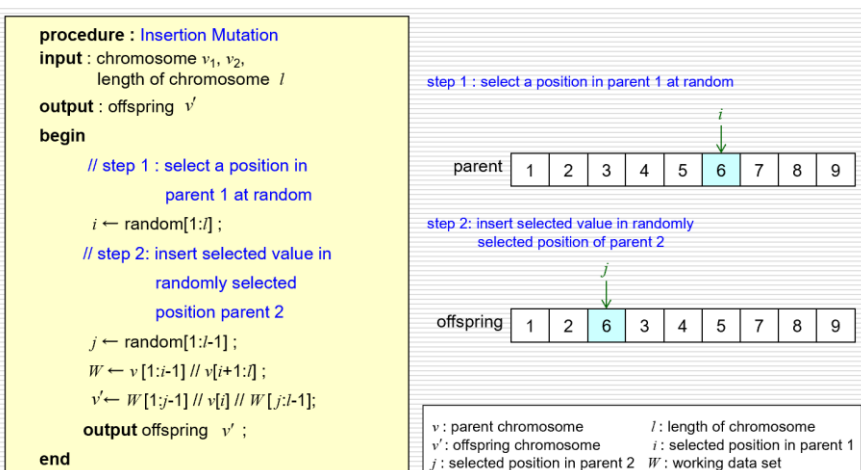
1. 从文件中读入 tsp 问题以及最优解
2. 根据用户的输入选择局部搜索策略
3. 使用局部搜索策略，或者模拟退火策略进行搜索，直到找到符合要求的近似解。

接下来分别具体解释各个局部搜索策略与搜索流程。

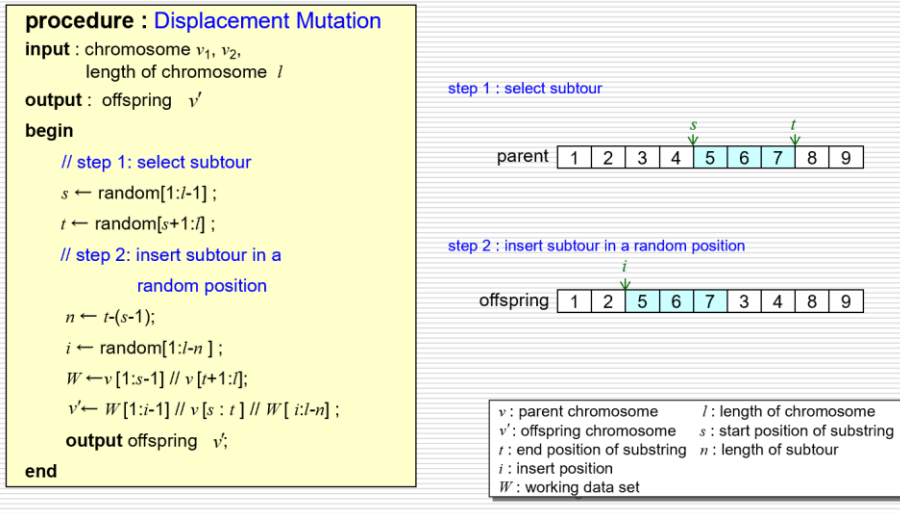
1. Inversion mutation



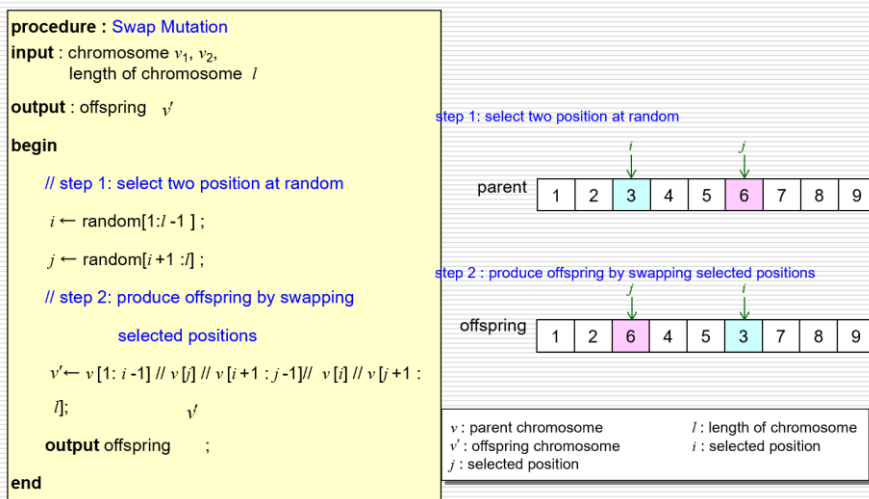
2. Insertion Mutation



3. Displacement Mutation



4. Swap Mutation



5. 局部搜索流程

tour = 随机生成的解

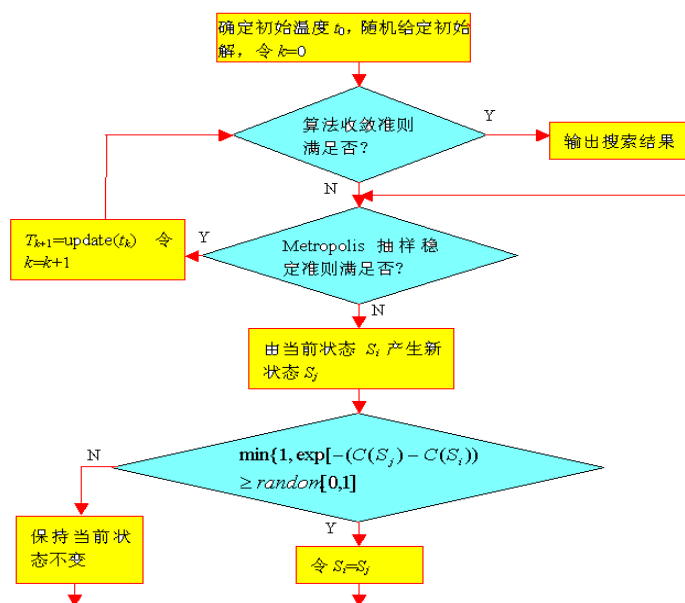
while tour 的长度不满足要求

做 1000 次局部搜索，每次生成新解时若长度更优则取代原解

end while

6. 模拟退火算法

模拟退火算法的流程图如下：



根据本实验要求，tsp 问题的模拟退火算法伪代码如下：

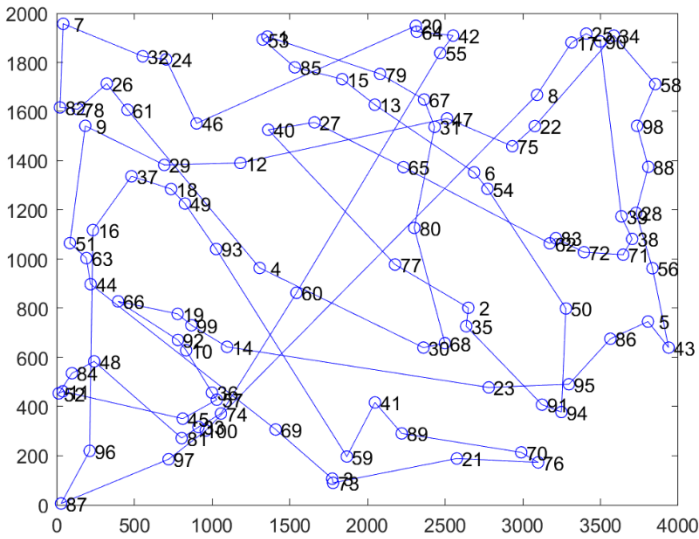
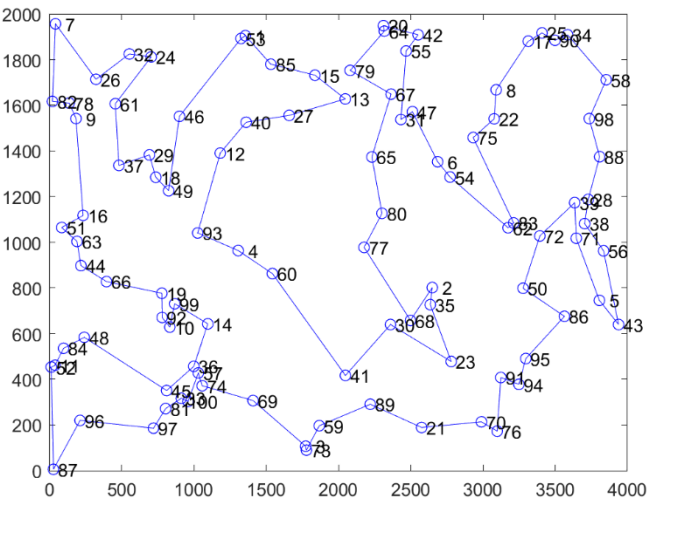
```
T = 5000;
T_end = 0.01;
while T > T_end and tour 长度 > 1.1*最优解长度
    for i = 1:1000
        new_tour = 从 tour 由局部搜索策略生成新环游
        difference = new_tour 长度 - tour 长度
        if difference < 0 or rand() <= exp(-difference/T)
            tour = new_tour
        end if
    end for
    T = T*0.98;
end while
```

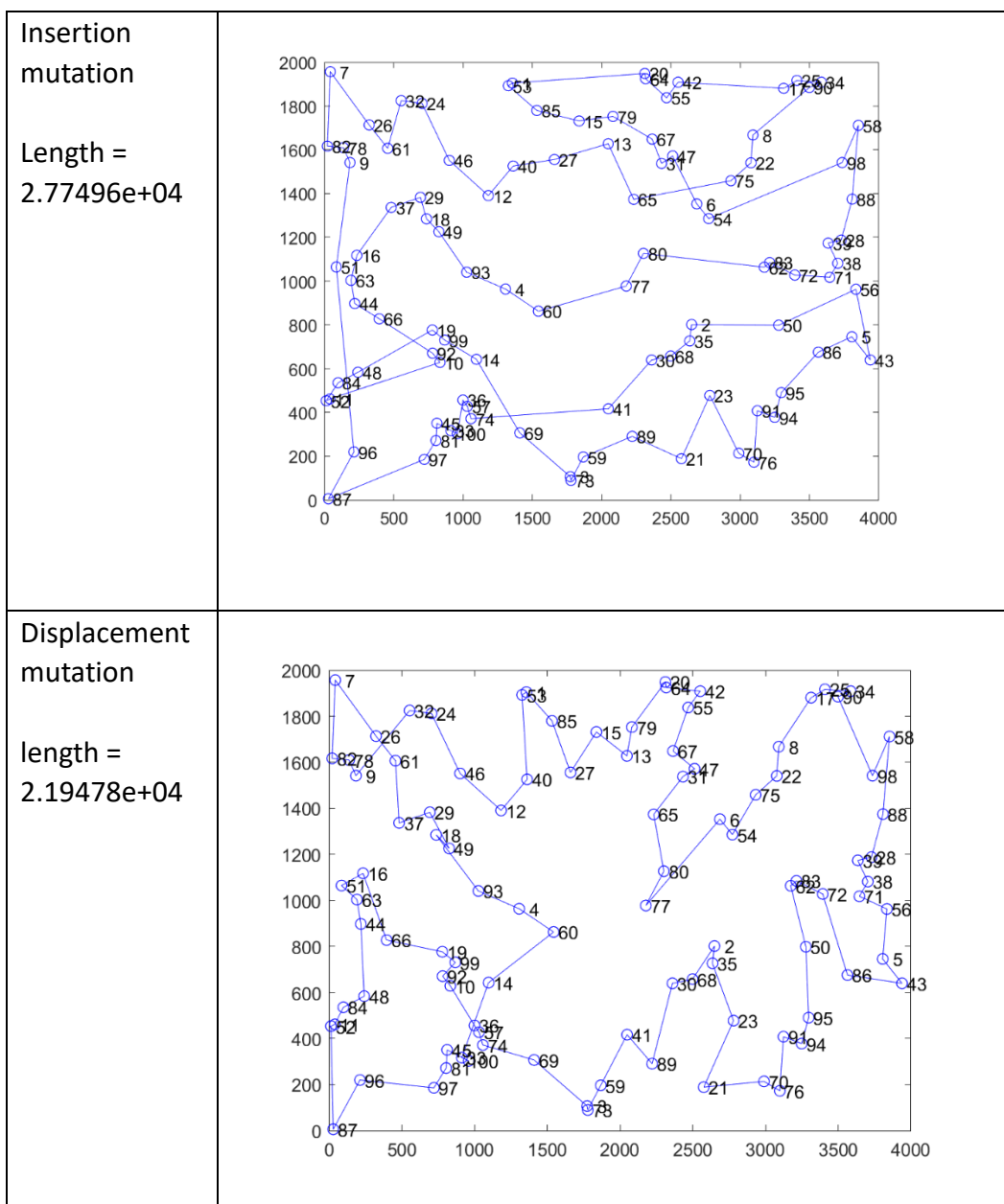
3. 结果分析

本实验在 Matlab R2020b 上实现并运行。

(1) 局部搜索策略

由于单纯的局部搜索策略运行较慢，本人在每种策略上只运行 1000 次循环，其中每次循环做 1000 次局部搜索。分别运行四种局部搜索策略，得到的结果如下：

局部搜索策略 最终总长度	可视化路径
Swap mutation Length = 4.27869e+04	
Inversion mutation Length = 2.37732e+04	



可以看到，Inversion 和 displacement 两种策略性能接近且最优，insertion 次之，swap 性能最差。

(2) 模拟退火算法

模拟退火算法用到的相关参数包括：

- 初始温度 $t_0 = 5000$
- 温度更新函数 $\text{update}(t) = 0.98 * t$
- 循环终止条件
 $T \leq T_{\text{end}}$ or $\text{tour_length}(Z, \text{tour}) \geq 20751 * 1.1$
 ($T \leq T_{\text{end}}$ 时迭代次数最大，为 650)

仿照之前的做法，对每种局部搜索策略进行模拟退火，并且把得到的迭代次

数在单纯的局部搜索程序上再次尝试比较最终路径长度，运行结果如下：

局部搜索策略 最终总长度	可视化路径
<div>Swap mutation</div> <div>iteration = 650</div> <div>Length = 2.77488e+04</div> <div>单纯局部搜索循环 650 次最终路径长度： 3.55152e+04</div>	
<div>Inversion mutation</div> <div>Iteration = 228</div> <div>Length = 2.26928e+04</div> <div>单纯局部搜索循环 228 次最终路径长度： 2.41887e+04</div>	

<p>Insertion mutation</p> <p>Iteration = 294 Length = $2.28163\text{e}+04$</p> <p>单纯局部搜索循环 294 次 最终路径长度： $2.75389\text{e}+04$</p>	
<p>Displacement mutation</p> <p>iteration = 359 length = $2.25638\text{e}+04$</p> <p>单纯局部搜索循环 359 次 最终路径长度： $2.41520\text{e}+04$</p>	

如表所示，模拟退火下所有的局部搜索策略的性能都有所提高，其中 **Swap Mutation** 性能提升最明显，虽然性能最差。其余的三种局部搜索策略性能接近，其中 **inversion mutation** 效果最好。

4. 结论

本次实验通过实现多种局部搜索策略与模拟退火算法并进行可视化，探究了不同的局部搜索策略和模拟退火算法的性能。总的来说，局部搜索策略中 **inversion mutation** 性能最好，模拟退火算法较单纯的局部搜索策略性能有所提升。本次实验还让我体会到了 **Matlab** 编程的很多好处，代码较简洁且 **debug** 很方便。

主要参考文献(三五个即可)

1. combOptGA-v1.00(1).ppt
2. 人工智能基础教程-朱福喜第二版