

遗传算法与分枝定界法求解TSP研究

杨思明, 王凤军

无锡商业职业技术学院, 江苏 无锡

Email: 52801276@qq.com

收稿日期: 2020年9月1日; 录用日期: 2020年9月11日; 发布日期: 2020年9月18日

摘要

在解决旅行商问题时, 有两种常用的方法, 即遗传算法与分枝定界法。本文使用K均值聚类改进分枝定界法, 求解给定的旅行商问题。通过运用这两种算法求解TSP进行比较, 相比之下K均值聚类优化的分枝定界法在解决旅行商问题中表现得更好。

关键词

旅行商问题, 遗传算法, 分枝定界法

Research on Solving TSP with Genetic Algorithm or Branch and Bound Method

Simin Yang, Fengjun Wang

Wuxi Institute of Commerce, Wuxi Jiangsu

Email: 52801276@qq.com

Received: Sep. 1st, 2020; accepted: Sep. 11th, 2020; published: Sep. 18th, 2020

Abstract

When solving the traveling salesman problem, there are two commonly used methods, namely genetic algorithm method, and branch and bound method. To solve the given traveling salesman problem, this paper uses K-means clustering to improve the branch and bound method. By comparing the performance of these two algorithms in solving the TSP problem, the branch and bound method with K-means clustering performs better in solving the traveling salesman problem.

Keywords

Traveling Salesman Problem, Genetic Algorithm, Branch and Bound Method

Copyright © 2020 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

旅行商问题(Travelling salesman problem, 简称 TSP), 是由 Dantzig 等人在 1959 年提出, 给定一系列城市和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路程。旅行商问题在运筹学中非常重要, 也是组合优化中的一个 NP 难问题, 在物流、旅游业中有很高的研究价值[1]。通过对参考文献进行阅读、分析发现, 求解 TSP 问题, 最常用的算法是遗传算法和分支定界法[2] [3] [4]。因此本文选用遗传算法和分支定界法两种算法求解该问题。

2. 问题描述

如图 1 所示, 以位置为 0, 0 的点为起点和终点对这 76 个点进行遍历。要求找到一条路径使机器人总运行时间最短。已知信息为: 机器人最大运行速度 1 m/s; 在每一点的停留时间与通过速度; 以及每一点的坐标。

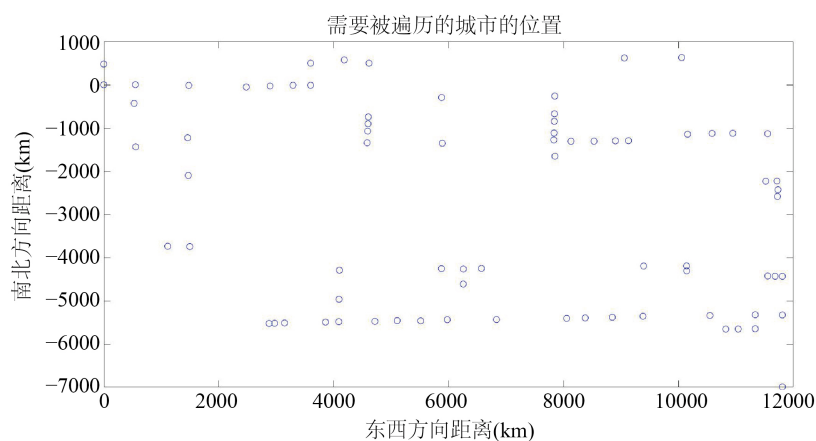


Figure 1. The distribution of cities in TSP

图 1. TSP 问题城市分布图

忽略机器人的启动与刹车的加速时间, 假设机器人全程运动速度为 1 m/s。则该问题的解等价于机器人通过旅行商问题求出的最短路径所需的时间加上机器人在所有点停留的时间。本文使用遗传算法与 K 均值聚类改进分枝定界法求解该问题。

3. 分支定界法

分支定界法是由 Richard M. Karp 在 20 世纪 60 年代发明, 成功求解含有 65 个城市的旅行商问题, 创当时的记录。这种方法不但可以求解纯整数规划, 还可以求解混合整数规划问题。分支定界法是一种

搜索与迭代的方法, 选择不同的分支变量和子问题进行分支。

如图 1 所示, 所需要遍历的点有 76 个, 除去起点后, 其全排列有 $75!$ 种可能性。经过测试, 本文所使用的计算机平台(16 G 内存, 英特尔 8 代 i7 处理器)在 1 h 内最多对 14 个城市使用分枝定界法进行求解, 因此首先要对需要遍历的点进行处理, 减少其数量。

假设有一个商人需要去中国, 日本, 美国, 巴西中的几个城市送货, 且这个商人希望能以最小的路程送完所有的货物。那这个商人必定在当前国家遍历完所有需要前往的城市后才会前往下一个国家。在两个国家之间的城市多次往返的路线不可能是最优解。因此在点分布不均匀的情况下, 对分布密集的一些点进行合并可以减少问题的复杂度。其在本质上是排除了一些路线, 其中大部分是路程非常长的路线。

本文使用聚类分析对实现对点的合并。聚类分析起源于分类学, 由于人们有时难以仅凭借经验和专业知识去对事物进行分类, 因此人们把数学工具引用到了分类学中并形成了数值分类学。之后随着多元分析技术的发展, 人们又发展出了聚类分析。从分类指标上可以分为划分法, 层次法, 基于模型的方法, 基于网格的方法, 基于密度的方法。使用 K 均值聚类将所有的点划分为指定的类数。K 均值聚类的算法需要输入分类数量 K, 利用各类别中对象的均值更新聚类相似度来完成分类。其流程如下:

1. 从 n 个数据对象任意选择 K 个对象作为初始聚类中心。
2. 计算每个对象与聚类中心的距离; 并根据最小距离重新对相应对象进行划分。
3. 重新计算每个聚类的均值作为新的聚类中心。
4. 循环 2 到 3 直到每个聚类不再发生变化为止。

对图 1 中 76 个点进行聚类分析, 结果如图 2~7 下所示:

由上图所示, 随着分类类别增加, 分类结果越来越精细, 成行成列以及距离相对较近的点被很好地划分进了同一类别。且各个类别内点的数量没有超过 14 个。求取聚类后的各类别的中心点作为各个类别的代表。将其输入到分支定界法中搜索最优路径。

分支定界算法是一种始终围绕着一颗搜索树进行的算法。其将出发点看作搜索树的根节点, 从这里出发, 分支的含义就是将大的问题分割成小的问题。而定界就是在分支的过程中检查子问题的上下界, 如果子问题不能产生一比当前最优解还要优的解, 那么砍掉这一支。直到所有子问题都不能产生一个更优的解时, 算法结束[5]。算法流程如下图所示:

由图 8 所示, 我们按顺序遍历每一种排列, 并不断更新最小距离, 当其中一个父节点的当前距离已经大于最小距离时就跳过该父节点。直到遍历完所有可行解后输出最优解。

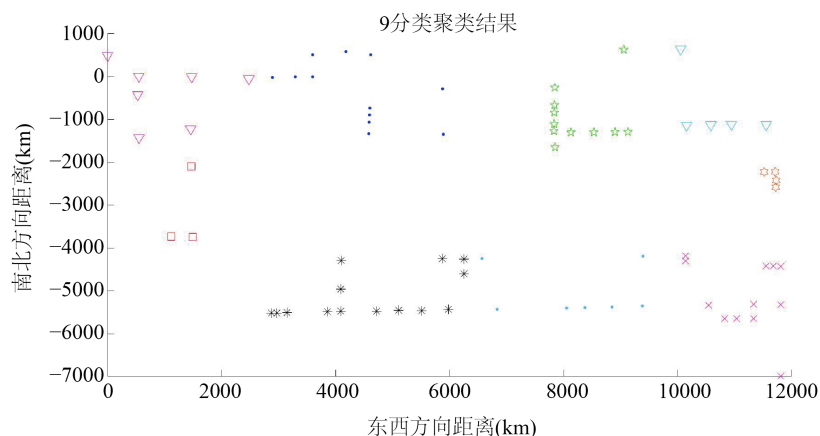


Figure 2. The result of the first classification

图 2. 分类结果 1

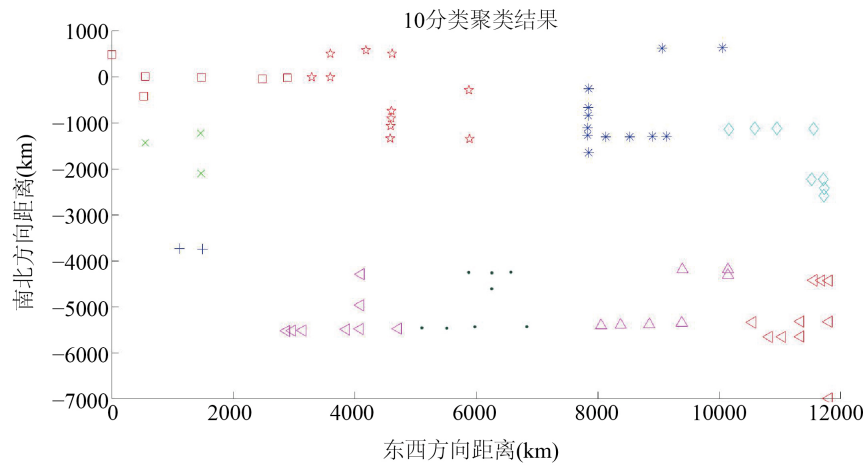


Figure 3. The result of the second classification

图 3. 分类结果 2

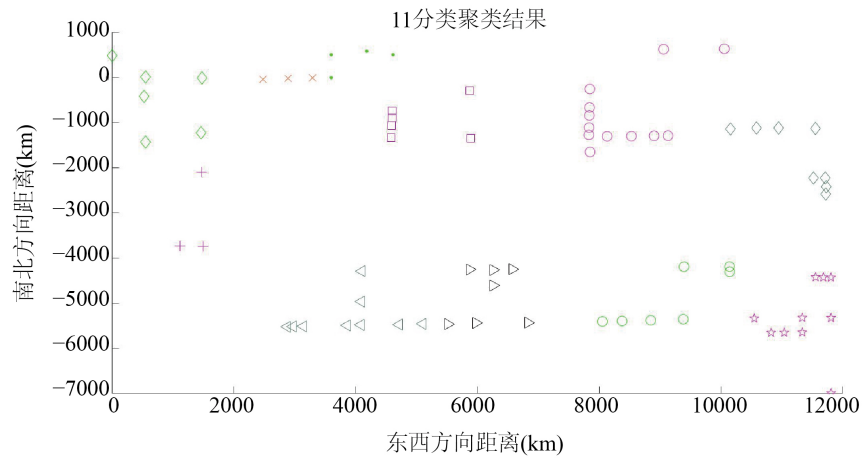


Figure 4. The result of the third classification

图 4. 分类结果 3

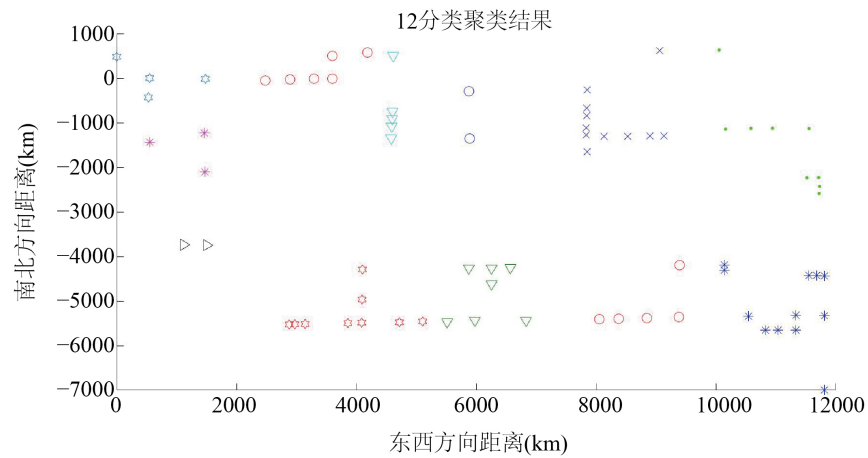


Figure 5. The result of the fourth classification

图 5. 分类结果 4

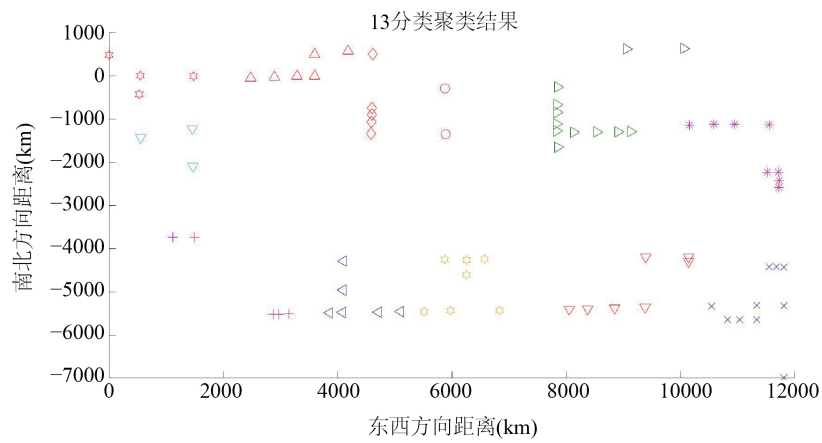


Figure 6. The result of the fifth classification
图 6. 分类结果 5

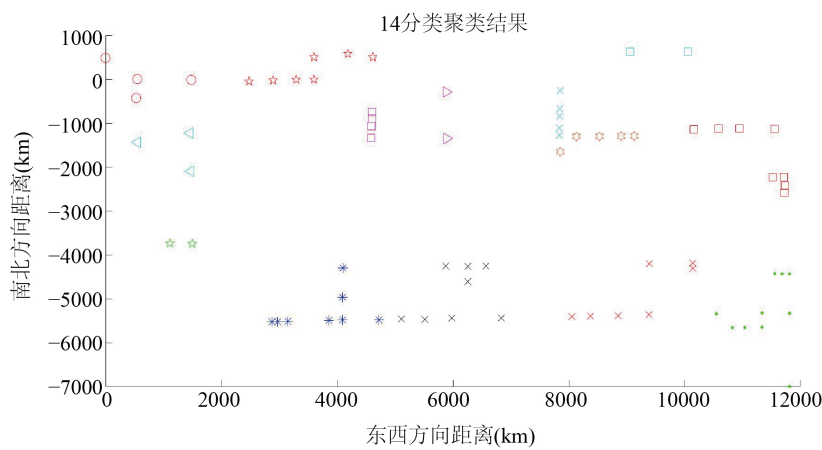


Figure 7. The result of the sixth classification
图 7. 分类结果 6

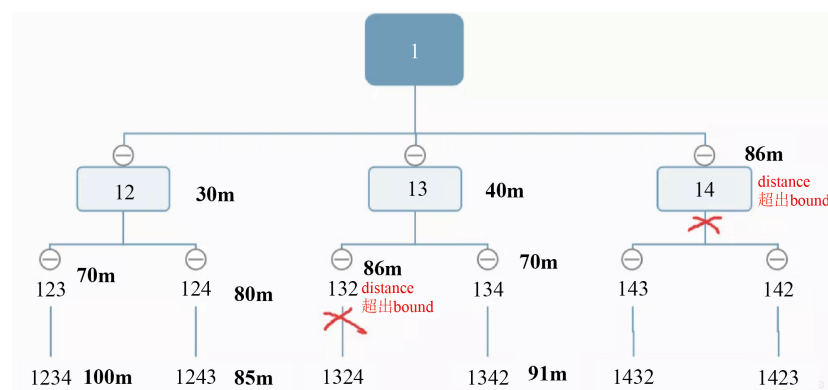


Figure 8. Schematic diagram of branch and bound method
图 8. 分支定界法示意图

将各类别的中心点输入到分支定界算法中搜索最优路径便得到了各个集合的最优连接方式。每个集合选出一个在最优路径上下一级集合中心点最近的点作为连接点。然后每个集合分别将上一级的连接点

作为起始点,自己的连接点作为终点,其余的点作为需要遍历的点输入到分枝定界法中搜索出最优路径。最后将所有路径合并便得到了一个较优解。

从图 9~12 可以看出聚类类别数量对整体路径影响不大,只对细微路径有些影响。由于 K 均值聚类每次随机选取 K 个点作为起始点,若根据城市分布人工指定起始点可能进一步减少求解出的最短路程。理论上聚类类别数越多,实际求解过程越接近于直接用分支定界算法进行求解,则求解结果离最优解应该越近。但是经过实验结果表明在聚类类别数量与点的数量差距较大时,增加聚类数量未必能提高解的精度。考虑到求解时间与求解精度,在实际应用该方式时,可以用较少的分类类别数多次运行选取最优解,若能人工指定初始点,该方法的输出将更为稳定。

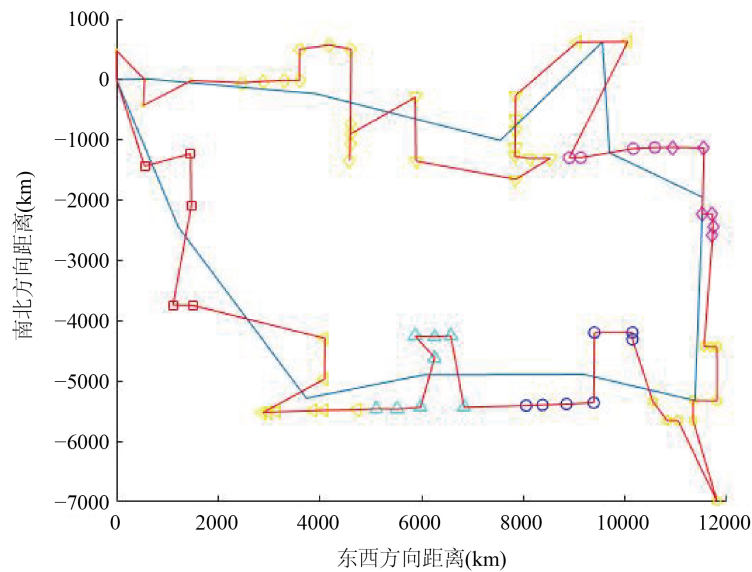


Figure 9. The final path when the number of cluster categories is 11
图 9. 聚类类别为 11 时的最终路径

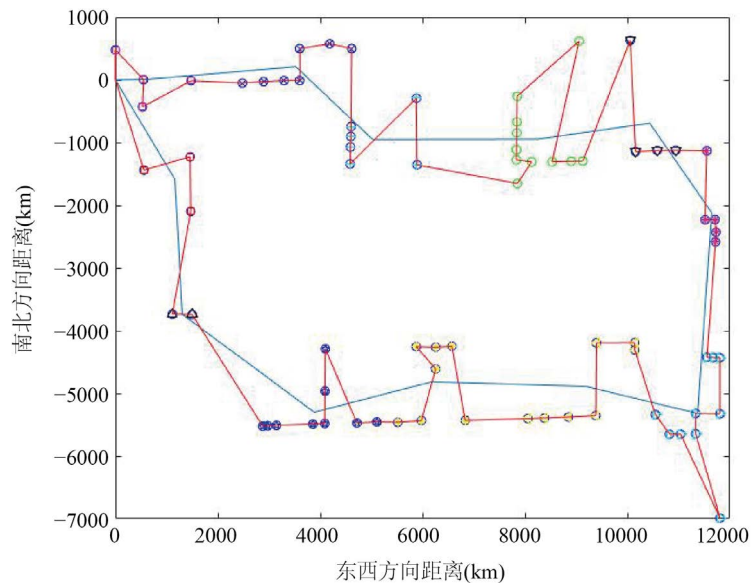


Figure 10. The final path when the number of cluster categories is 12
图 10. 聚类类别为 12 时的最终路径

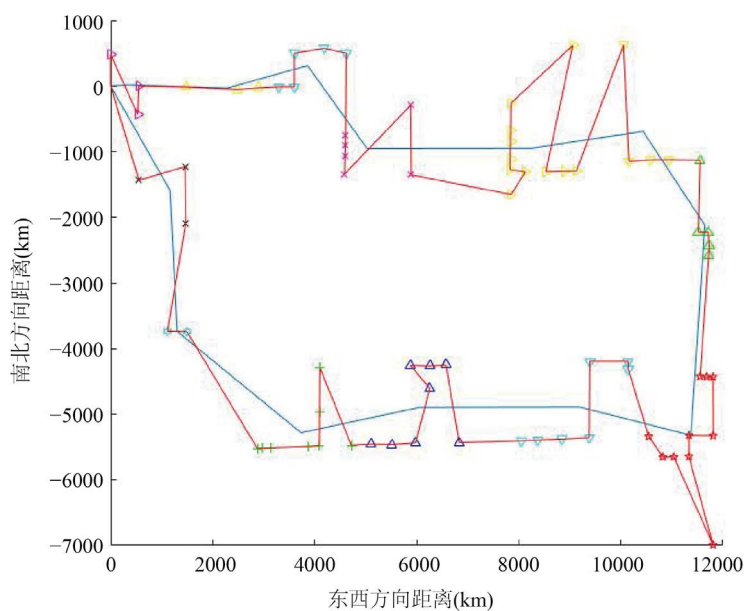


Figure 11. The final path when the number of cluster categories is 13

图 11. 聚类类别为 13 时的最终路径

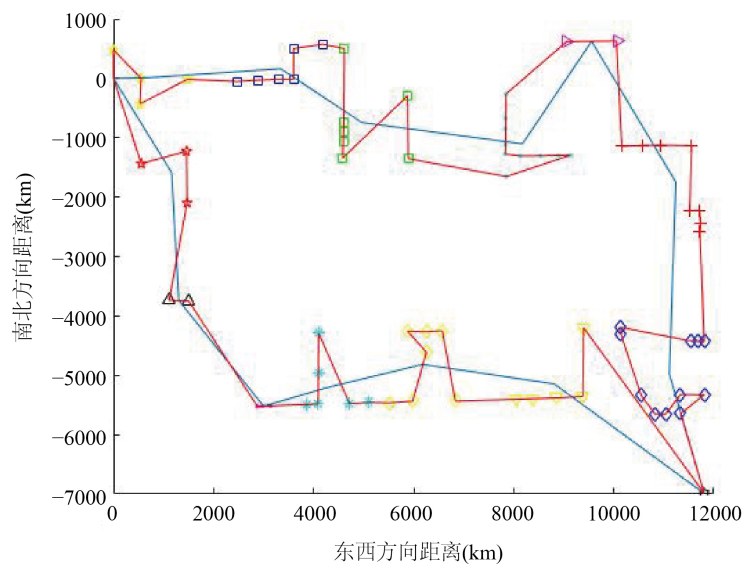


Figure 12. The final path when the number of cluster categories is 14

图 12. 聚类类别为 14 时的最终路径

4. 遗传算法

遗传算法是由 John Holland 于 20 世纪 70 年代根据大自然中生物体进化规律而设计提出的。该算法通过模拟自然进化中选择、交叉、变异现象来搜索最优解, 并已被人们广泛地应用于组合优化、机器学习、信号处理、自适应控制和人工生命等领域[6]。遗传算法流程如图 13 所示, 计算结果如图 14 所示:

在求解旅行商问题时要对遗传算法的参数编码, 适应度函数, 交叉方法进行设计, 设计结果如下:

1. 将点的遍历顺序作为编码, 即编码直接表示路径。
2. 根据每一编码计算其总路程, 并用 $100000/\log(\text{总路程} + 1)$ 作为适应度函数。

3. 在两个基因中随机选取两个交叉点, 然后根据交叉点选取两个子串直接插入到对方的交叉点中。最后删掉两个后代中各自因交叉产生的重复的点。

设置种群规模为 1000; 运行次数为 5000; 淘汰率为 30%, 交叉概率为 90%, 变异概率为 20% 进行搜索, 搜索结果如下图所示, 最短时间为 60381 s。

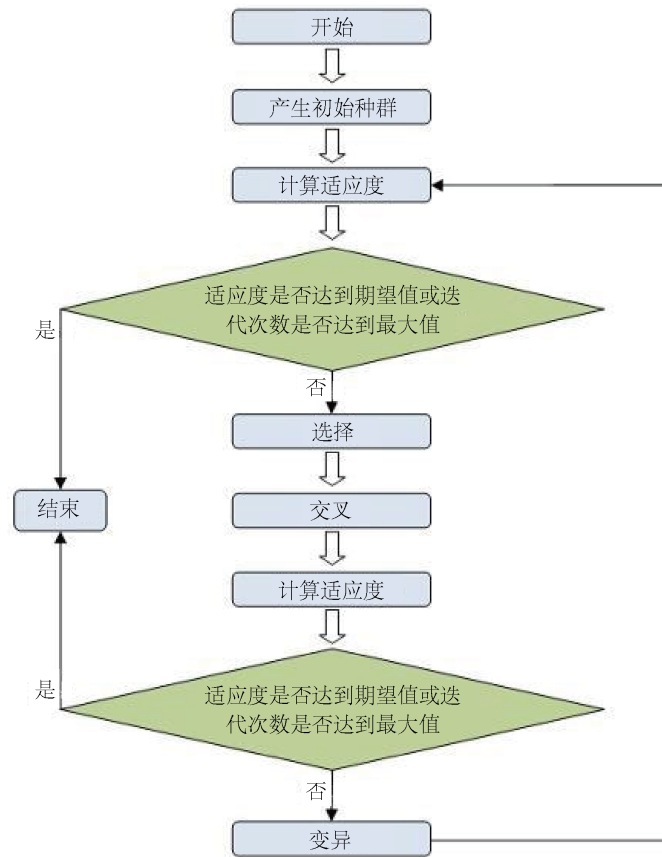


Figure 13. The flowchart of genetic algorithm
图 13. 遗传算法流程图

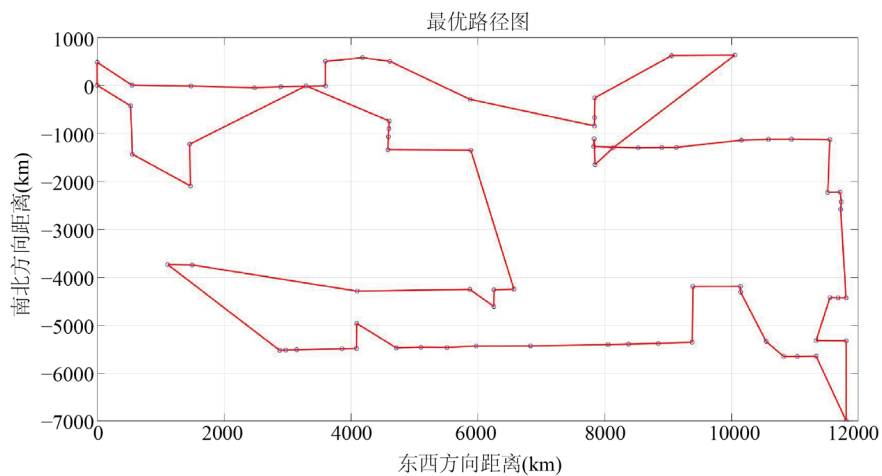


Figure 14. The outcome of genetic algorithm
图 14. 遗传算法搜索结果

5. 结论

遗传算法作为一种启发式算法有强的通用性, 但是其随机性也很强。经常需要多次运行遗传算法才能得出一个比较理想的结果。同时, 随着问题维数的增大, 遗传算法需要占用很多的内存来存储基因, 计算量也较大。遗传算法搜索结果受参数影响也较大。实际测试中若该问题采用 10000 种群规模, 进化 500 次, 其搜索结果会很差。通过两种算法求解 TSP 比较, 显而易见, 基于 K 均值聚类优化的分枝定界法, 在上述问题求解中表现得更好。

参考文献

- [1] 许冲, 钟玮, 刘欣欣. 一种求解旅行商问题的演化算法研究[J]. 闽南师范大学学报(自然科学版), 2020, 33(2): 44-48.
- [2] 姚宇威, 邓燕妮. 混合杂草遗传算法求解旅行商问题[J]. 科学技术创新, 2020(11): 40-41.
- [3] 汤雅连, 杨期江. 求解旅行商问题的蚁群优化算法参数设计[J]. 东莞理工学院学报, 2020, 27(3): 48-54.
- [4] 陈春燕, 彭阳, 许环梓, 何宇佳, 石苗. 蚁群和遗传算法在旅行路线规划中的研究[J]. 高师理科学刊, 2020, 30(7): 33-36
- [5] 王兆锐, 林剑, 张俊丽, 官静萍. 基于车公里成本的多车型车辆规划方法[J]. 物流技术, 2019(1): 82-87.
- [6] 胡士娟, 鲁海燕, 黄洋, 等. 求解工作量平衡多旅行商问题的改进遗传算法[J]. 计算机工程与应用, 2019(17): 150-155.