

# 基于禁忌表的定位算法求解 TSP 问题<sup>\*</sup>)

雷开友<sup>1</sup> 邱玉辉<sup>1</sup> 刘光远<sup>2</sup> 贺 一<sup>1,3</sup>

(西南师范大学计算机与信息科学学院 重庆 400715)<sup>1</sup> (西南师范大学电子信息工程学院 重庆 400715)<sup>2</sup>  
(重庆师范大学现代信息管理系 重庆 400047)<sup>3</sup>

**摘要** 本文提出了一种基于禁忌表的定位算法求解 TSP 问题的快速、高效近似算法。这种算法结合了禁忌搜索算法中禁忌表及大规模构造算法和定位改进算法求解规模较大的 TSP 问题。计算机实例仿真证明,算法在求解质量和求解速度两方面高于著名的启发式算法的解。该算法针对 TSP 问题提出,是非常有效的。  
**关键词** 禁忌搜索,禁忌表,TSP 问题,大规模构造算法,定位改进算法

## Position-Fixed Based on Tabu List Algorithm to TSP

LEI Kai-You<sup>1</sup> QIU Yu-Hui<sup>1</sup> LIU Guang-Yuan<sup>2</sup> HE Yi<sup>1,3</sup>

(Faculty of Computer & Information Science, Southwest-China Normal University, Chongqing 400715)<sup>1</sup>

(School of Electronic & Information Engineering, Southwest-China Normal University, Chongqing 400715)<sup>2</sup>

(Dept. of Modern Information Management Science, Chongqing Normal University, Chongqing 400047)<sup>3</sup>

**Abstract** This paper proposes a fast and effective approximate algorithm—position-fixed based on tabu list algorithm, Which incorporates the tabu list in tabu search algorithm, SizeScale-construction algorithm and position-fixed improvement algorithm to solve the large-scale traveling salesman problem. Position-fixed based on tabu list algorithm is specially devised for TSP, the experimental results show that the algorithm outperforms the known best ones in quality of solution and running speed compared to the famous heuristic algorithm.

**Keywords** Tabu search, Tabu list, TSP, SizeScale-construction, Position-fixed improvement

## 1 引言

TSP(traveling salesman problem)问题是一个著名 NP-Hard 组合优化问题<sup>[1,5-8]</sup>,广泛应用于 VLSI 芯片设计、电路板布局、机器人控制、车辆选路等领域。问题描述如下:给定  $n$  个城市及两两城市之间的距离,求一条经过各城市一次且仅一次的最短路线。其图论描述为:给定图  $G=(V, E, W)$ ,其中  $V$  为顶点集,  $E$  为边集,  $W$  为边权集(各顶点间连接距离),要求确定一条长度最短的 Hamilton 回路,即遍历所有顶点一次且仅一次的最短回路。

TSP 问题描述非常简单,但最优化求解很困难,如枚举法寻优,其算法复杂性呈指数增长,即所谓的“组合爆炸”。所以,针对 TSP 问题求解采用有效的环路构造算法和环路改进算法等启发式算法或者把模拟退火、神经网络、遗传算法、禁忌搜索算法、蚂蚁算法、竞争算法等最优化新方法运用于 TSP 问题的求解。

本文结合一种大规模环路构造算法(SizeScale-construction)和一种环路改进算法(tour improvement)及禁忌搜索算法(tabu search)提出了基于禁忌表(tabu list)的定位算法。

## 2 算法的提出

求解 TSP 问题,体现在求解质量和求解速度两方面上,求解质量为所得路径的长度比已知最短路径高出的长度或百分比,求解速度为找到近优解或最优解的时间,一般用 CPU 时间来度量。本文算法在两个方面表现较好。

### 2.1 插入算法产生初始解

环路构造算法之一是插入算法,常见的是单个城市插入算法和大规模(批量)城市插入算法。

单个城市插入算法常见有两种描述。第一种描述如下:生成所有  $n$  个城市的一个随机全排列,用它来构成初始循环队列(简称初始队列),任取三个城市,形成一个最小回路,在余下的城市中,任取一城市,插入到上述回路中,不同的插入位置会导致不同的旅行次序,最终的插入位置必须使得插入以后的旅行路径的长度最小(也就是遵循长度最小增量原则),重复任取余下城市,直至所有城市插入完毕。第二种描述如下:生成所有  $n$  个城市的一个随机全排列,用它来构成初始循环队列(简称初始队列),取初始队列中前三个城市来构成一个最小回路,把初始队列中的第四个城市插入这个回路,不同的插入位置会导致不同的旅行次序,最终的插入位置必须使得插入以后的旅行路径的长度最小。初始队列中剩余城市的第五个、第六个城市……依次插入,直到最后一个城市插入这个队列为止。

大规模(批量)城市插入算法描述如下:首先得到一个只包含部分城市的子环路,每次向该环路中插入一批余下的城市得到新的环路,得到新的环路后,再通过快速改进算法如 2-opt 或 LK 搜索优化该环路,优化环路时仍遵循最小增量原则,重复取一批余下城市,直至所有城市插入完毕。

本文用大规模(批量)城市插入算法产生初始解。

### 2.2 定位算法

为了使所得的解进一步被优化,需要用下面所述的基于禁忌表的定位算法来进行局部优化。先用大规模(批量)城市插入算法得到一个近优解(在近优解中,城市的排列也接近最

<sup>\*</sup> 本文受到教育部科学技术重点项目(NO. 104262 和重庆市科委基金项目 2003-7881)共同资助。雷开友 副教授,博士研究生,研究方向为神经网络与计算智能。邱玉辉 教授,博士生导师,长期从事非单调推理、近似推理、神经网络、机器学习和分布式人工智能的教学和研究工作。刘光远 教授,硕士生导师,主要研究方向为神经网络、模糊控制及现代优化方法。贺 一 讲师,博士研究生,研究方向为神经网络与计算智能。

优队列),然后从所得近优解的局部截取一段一段重复的旅行路径,再按单个插入算法的思想对所截取的局部路径进行优化,见图1,这是一种环路改进算法,截段分解极大地降低原TSP问题的规模,而并不降低搜索的性能。为了使优化后的局部路径能够合理地接回到近优解,必须保证这个局部路径

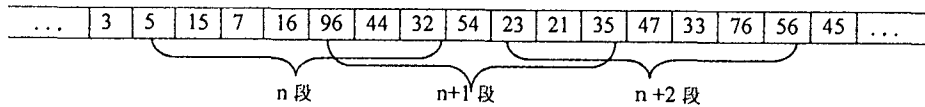


图1 每段为七个城市的局部截取重复的旅行路径图

为什么要局部截取一段一段重复的旅行路径呢?第一,避免每段首尾城市由于位置固定后不能移动,故不能参加本段的局部优化,但可参加下一段局部优化;第二,避免城市数目往往不能被小于它的任何整数除尽,余数为零,故要局部截取每段段长相同的段就较难,给编程及数据存储带来困难。例如,FL417有417个城市,假如每段20个城市,最后余下17个城市给编程及数据存储带来了困难。

2.3 禁忌表在定位算法中的使用

在禁忌搜索算法(TS)中,构成算法的设计框架为邻域结构、禁忌表、禁忌长度、候选集、特赦规则等,其中,禁忌表用来避免迂回搜索,即避免重复搜索,节约搜索时间,从而最终实现全局优化<sup>[1~4]</sup>。文[10~11]中对TSP问题的观察发现:多个不同的近优解的交集会以很大的概率出现在全局最优解中,即定义L1和L2是图G=(V,E,W)的TSP环路,定义P(L1,L2)=E(L1)∩E(L2)为L1和L2的公共边的集合,称为由L1和L2得到的部分解,这个部分解就是近优解的交集,它是很难局部优化改善的。因此在一个近优解中有一些段(部分解)是很难局部优化改善的。当用定位算法搜索时,建立一个禁忌表,设定一迭代步数及禁忌长度,若超过一定的迭代步数,某段局部路径不能改善,则把此某段局部路径装入禁忌表,不再搜索,继续下一段局部路径搜索,直至所有局部路径段搜索完毕且符合结束条件。由于在定位算法中使用了禁忌表,我们把整个算法叫作基于禁忌表的定位算法(Position-Fixed Based on Tabu List Algorithm, PFTLA)。

2.4 基于禁忌表的定位算法流程图(图2)

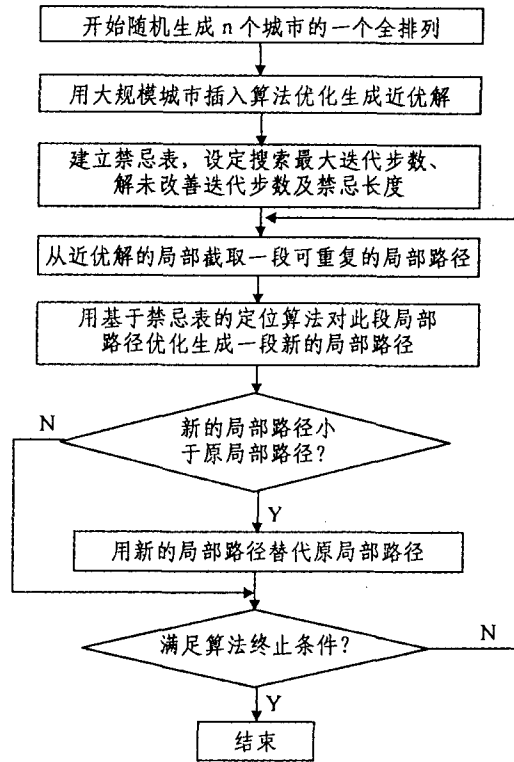


图2

表2 文[12]中采用自适应冷却进度表时的结果

| 实例     | 模拟退火算法     |         | 两阶段模拟退火算法 |         |
|--------|------------|---------|-----------|---------|
|        | 平均cpu时间(s) | 平均路径长度  | cpu时间(s)  | 平均路径长度  |
| Lin318 | 1722.98    | 42869.6 | 703.22    | 42835.0 |
| Att532 | 10104.70   | 88105.5 | 6044.17   | 88162.1 |

表3 文[8]中采用嵌套插队算法计算的结果

| 实例     | 平均       |         | 最好       |       | 最差       |       |
|--------|----------|---------|----------|-------|----------|-------|
|        | cpu时间(s) | 长度      | cpu时间(s) | 长度    | cpu时间(s) | 长度    |
| Lin318 | 235.2    | 42548.0 | 234.9    | 42354 | 235.7    | 42748 |
| Att532 | 361.5    | 88090.6 | 361.2    | 87548 | 362.3    | 89716 |

表4 本文采用基于禁忌表的定位算法结果

| 实例     | 平均       |         | 最好       |       | 最差       |       |
|--------|----------|---------|----------|-------|----------|-------|
|        | cpu时间(s) | 长度      | cpu时间(s) | 长度    | cpu时间(s) | 长度    |
| Lin318 | 184.3    | 42528.2 | 177.5    | 42332 | 195.8    | 42939 |
| Att532 | 289.5    | 88075.7 | 283.2    | 87622 | 302.7    | 89176 |

比较表1、表2、表3、表4结果显示:对于Lin318、Att532基于禁忌表的定位算法都能用更短的时间,却取得了比嵌套

3 计算机实例仿真及结果分析

3.1 用基于禁忌表的定位算法对Lin318、Att532的求解结果

用来测试基于禁忌表的定位算法性能的实例是:Lin318(318个城市)、Att532(532个城市)。它们的原始数据来源于网址ftp://softlib.rice.edu的目录/pub/tsplib下的TSPLIB。

采用经典冷却进度表时,用模拟退火算法(SA)和两阶段模拟退火算法(TSSA)计算所得结果如表1所示<sup>[12]</sup>;采用自适应冷却进度表时,用模拟退火算法(SA)和两阶段模拟退火算法(TSSA)计算所得结果如表2所示<sup>[12]</sup>;采用嵌套插队算法(NQJA)计算的结果如表3所示<sup>[8]</sup>;用基于禁忌表的定位算法计算的结果如表4所示;所有平均结果都是运行20次的平均结果<sup>[8]</sup>。

表1 文[12]中采用经典冷却进度表时的结果

| 实例     | 模拟退火算法     |         | 两阶段模拟退火算法 |         |
|--------|------------|---------|-----------|---------|
|        | 平均cpu时间(s) | 平均路径长度  | cpu时间(s)  | 平均路径长度  |
| Lin318 | 125.98     | 43295.2 | 87.63     | 43375.1 |
| Att532 | 651.70     | 89555.9 | 345.49    | 89648.6 |

插队算法约好的平均路径长度;对于 CPU 的运行时间,基于禁忌表的定位算法每次所用时间的变化很小,而模拟退火算法和两阶段模拟退火算法,CPU 所用时间的变化却较大,特别是当采用自适应冷却进度表时,变化就更大。显然,在不降低解的质量的前提下,基于禁忌表的定位算法明显优于模拟退火算法和两阶段模拟退火算法,也优于嵌套插队算法。

### 3.2 用基于禁忌表的定位算法对 FL417 的求解结果

FL417(417 个城市)问题是一个已知最优解长度为 11861 的 TSP 问题,用最优解长度为 11861 标准来判断所得解的质量,更客观地说明问题。文[13]中,用嵌套划分算法(NP 算法)和其他启发式算法相结合(2-opt, 3-opt)计算了 FL417 问题,结果如表 5 所示。文[8]中,用嵌套插队算法对 FL417 计算 10 次所得的结果如表 6 所示。用基于禁忌表的定位算法对 FL417 的求解结果如表 7 所示。

表 5 文[13]用嵌套划分算法和启发式算法结合(2opt, 3opt)计算 FL417 的结果

| 方法              | 路径质量=(最优解-已获解)/最优解 | CPU 时间(s) |
|-----------------|--------------------|-----------|
| 2-opt, 起始路径随机选择 | 14.6%              | 112       |
| NP 算法结合 2-opt   | 14.0%              | 35        |
|                 | 4.9%               | 4199      |
| 3-opt, 起始路径随机选择 | 0.78%              | 206298    |
| NP 算法结合 3-opt   | 1.36%              | 25368     |
|                 | 0.51%              | 133129    |

表 6 文[8]用嵌套插队算法计算 10 次所得的结果

| 实例    | 平均    |           | 最好     |           | 最差     |           |
|-------|-------|-----------|--------|-----------|--------|-----------|
|       | 路径质量  | cpu 时间(s) | 路径质量   | cpu 时间(s) | 路径质量   | cpu 时间(s) |
| FL417 | 0.31% | 240       | 0.084% | 238       | 0.548% | 242       |

表 7 基于禁忌表的定位算法计算 10 次所得的结果

| 实例    | 平均     |           | 最好     |           | 最差     |           |
|-------|--------|-----------|--------|-----------|--------|-----------|
|       | 路径质量   | cpu 时间(s) | 路径质量   | cpu 时间(s) | 路径质量   | cpu 时间(s) |
| FL417 | 0.312% | 121.2     | 0.105% | 112.3     | 0.539% | 134.5     |

(上接第 192 页)

模型 2 为

$$y = \frac{\sin(x_1) \sin(x_2)}{x_1 x_2} \quad (22)$$

其中  $-5 \leq x_1, x_2 \leq 5$ , 数据点数为 400, 噪声为正态分布  $N(0, 0.04)$ , 孤立点为 20 个高斯分布的随机点,  $t_{max} = 40, C_{max} = 40$ 。仿真结果如图 2 所示,其中黑实线为模型轨迹,方框点为带有噪声和孤立点的输入数据,黑点为处理后的轨迹。

结论 本文采用 RCA 的模糊学习思想,对算法中遗忘函数进行了合理的修正,实现了算法的稳健收敛。由仿真结果可以看出,该算法在输入数据有噪声和孤立点的干扰下依然能自动地向理想的模型逼近。

### 参考文献

1 Bezdek J C. Pattern Recognition With Fuzzy Objective Function

比较表 5、表 6、表 7 结果显示:基于禁忌表的定位算法所求解的质量明显好于文[13]结果,也好于文[8]嵌套插队算法结果。说明基于禁忌表的定位算法是解决 TSP 问题的一种非常有效的算法。

结论 本文提出基于禁忌表的定位算法是快速、高效的近似算法,在合理的计算时间内求解较大规模 TSP 问题,实例验证在求解质量和求解速度方面表现较好。基于禁忌表的定位算法是一种很有竞争力的求解 TSP 问题的算法。

### 参考文献

- 1 王凌. 智能优化算法及其应用[M]. 北京:清华大学出版社,2001
- 2 Glover F, Laguna M. Tabu Search. Boston: Kluwer Academic publishers,1997
- 3 贺一,刘光远. 变异操作对禁忌搜索性能的影响研究[J]. 计算机科学,2002,29(5)
- 4 贺一,刘光远,邱玉辉. Tabu search 中集中性和多样性的自适应搜索策略. 计算机研究与发展[J]. 2004,41(1):162~166
- 5 周培德. 货郎担问题的几何解法[J]. 软件学报,1995(6)
- 6 于志伟,陶波,王元美. 一种竞争算法及其在组合优化问题上的应用[J]. 软件学报,1998(10)
- 7 鄢烈祥. 用列队竞争法解旅行商问题[J]. 运筹与管理,1999(3)
- 8 翟东海,靳蕃. 用嵌套插队算法解决 TSP 问题[J]. 运筹与管理,2003,12(04)
- 9 高国华,沈林成,常文森. 求解 TSP 的空间锐化模拟退火算法[J]. 自动化学报,1999,25(3)
- 10 万颖瑜,周智,陈国良,顾钧. SizeScale: 求解旅行商问题(TSP)的新算法. 计算机研究与发展[J],2002,39(10)
- 11 邹鹏,周智,陈国良,顾钧. 求解 TSP 问题的多级归约算法[J]. 软件学报,2003,14(1)
- 12 James M, James C P. A fast method for generalized starting temperature determination in homogeneous two-stage simulated annealing system [J]. Computer and Operation research, 1999, 26: 481~503
- 13 Shi Leyuan, et al. New parallel randomized algorithms for the traveling sales man problem [J]. Operation and research, 1999, 26: 371~394

Algorithms. New York: Plenum Press, 1981

- 2 Frigui H, Krishnapuram R. A robust competitive clustering algorithm with applications in computer vision. IEEE Trans. Pattern Anal. Mach. Intell, 1999, 21: 450~465
- 3 Hampel F R, Ronchetti E M, Rousseeuw P J, Stahel W A. Robust Statistics: The Approach Based on Influence Functions. New York: John Wiley & Sons, 1986
- 4 Hawkins D M. Identification of Outliers. London, U. K.: Chapman & Hall, 1980
- 5 Sugeno M, Yasukawa T. A fuzzy-logic-based approach to qualitative modeling. IEEE Trans. Fuzzy Syst, 1993, 1: 7~31
- 6 Chuang C-C, Su S-F, Chen S-S. Robust TSK Fuzzy Modeling for Function Approximation With Outliers. IEEE Trans. Fuzzy Syst, 2001, 9: 810~821