

基于禁忌搜索的 TSP 问题求解

李彦希 彭刚

(桂林空军学院教育技术中心 广西桂林 541003)

摘 要 对禁忌搜索算法搜索原理进行简要介绍,着重讨论禁忌搜索实现的相关技术问题。通过禁忌搜索算法求解典型的 TSP(旅行商)问题,并对结果进行了简要的分析。

关键词 禁忌搜索算法; TSP; 启发式算法;

一、前言

旅行商问题(TSP)是典型的具有 NPC 计算复杂性的组合优化难题,问题基本描述是:给定 N 个城市,有一旅行商从其中一个城市出发,访问各城市一次且仅有一次后回到原出发城市,求一条最短的巡回路径。禁忌搜索算法是典型的元启发式算法之一,可以用于求解如 TSP、背包问题、生产流调度等典型的组合优化问题。本文针对 TSP 问题的特点,通过禁忌搜索算法对典型的 TSP 实例进行求解。实验表明,禁忌搜索算法对于求解 TSP 问题具有良好的收敛性和较高的搜索效率。

二、TSP 问题描述^[1]

TSP 问题即:一个商人到 n 个城市推销商品,每两个城市 i 和 j 之间的距离为 d_{ij} ,当 $d_{ij} = d_{ji}, \forall i, j$ 时,称为对称距离 TSP,否则称非对称 TSP。商人从其中一个起点城市出发,选择一条道路,保证对每个城市访问一遍且仅仅访问一遍后回到起点,且所走路径最短。TSP 问题中两两城市之间的距离 d_{ij} 可以通过距离矩阵表示。TSP 问题是 NPC (Non deterministic Polynomial Complete) 问题,目前还没有多项式时间复杂度的算法,所以,我们可以使用元启发式算法来求解,如禁忌搜索算法、模拟退火算法、遗传算法等。本文通过禁忌搜索算法求解 TSP 问题实例。

对一般的 TSP,基于图论的数学模型为:

$$\min \sum_{i \neq j} d_{ij} x_{ij} \quad (1.1)$$

$$s.t. \sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, n, \quad (1.2)$$

$$\sum_{i=1}^n x_{ij} = 1, j = 1, 2, \dots, n, \quad (1.3)$$

$$\sum_{i,j \in s} x_{ij} \leq |s|-1, 2 \leq |s| \leq n-2, s \subset \{1, 2, \dots, n\}, \quad (1.4)$$

$$x_{ij} \in \{0, 1\}, i, j = 1, \dots, n, i \neq j. \quad (1.5)$$

其中，目标(1.1)要求距离之和最小。(1.2)要求商人从城市 i 出来一次，(1.3)要求商人走入城市 j 只有一次。(1.2)和(1.3)表示每个城市经过一次。仅有(1.2)和(1.3)的约束无法避免回路的产生，一条回路是由 $k(1 \leq k \leq n)$ 个城市和 k 条弧组成，因此，(1.4)约束旅行商在任何一个城市子集中不形成回路，其中 $|s|$ 表示集合 s 中元素个数。(1.5)中的决策变量 $x_{ij} = 1$ 表示商人行走的路线包含从城市 i 到城市 j 路径， $x_{ij} = 0$ 表示商人没有选择走这条路。 $i \neq j$ 的约束可以减少变量的个数，使得共有 $n \times (n-1)$ 个决策变量。此时， $D = \{0, 1\}^{n \times (n-1)}$ ， F 为 D 中满足(1.2)，(1.3)和(1.4)的可行解。 f 为目标函数。

在本文中，根据 TSP 特点，结合禁忌搜索算法，采用第二种描述方法。固定起点，其他 $n-1$ 个点采用 2-opt 进行邻域映射。

三、禁忌搜索算法基本思想^[1]

禁忌搜索 (Tabu Search) 算法是一种典型的元启发式算法，由 Glover 在 1986 年首次提出，进而形成一套完整算法。所谓禁忌就是为了避免在局部邻域搜索算法中陷入局部最优的主要不足，而禁止重复前面的工作。禁忌搜索算法用一个禁忌表记录下已经到达过的局部最优点，在下一次的搜索中，利用禁忌表中的信息不再或有选择地搜索这些点，以此来跳出局部最优点，找到全局最优解。

该算法可简单地表示为：

Step1: 选定一个初始解 X^{now} 及给定禁忌表 $H = \Phi$;

Step2: 若满足终止规则，则终止计算；否则，在 X^{now} 的邻域 $N(H, X^{\text{now}})$ 中选出满足禁忌要求的候选集 $\text{Can_H}(X^{\text{now}})$ ；在 $\text{Can_H}(X^{\text{now}})$ 中选一个评价值最佳的解 X^{next} ， $X^{\text{now}} = X^{\text{next}}$ ；更新历史记录 H ，重复 Step2。

作为禁忌搜索算法，禁忌对象、禁忌长度、候选集等是它的主要特征，也是算法实现时应该注意的重要问题。

四、算法实现的相关技术问题

本文利用禁忌搜索算法求解 TSP 实例，其具体实现的相关技术问题如下：

1. 根据 TSP 问题的特点，结合禁忌搜索算法，采用第二种描述方法。解的集合为：

$D = \{S = (i_1, i_2, \dots, i_n) | i_1, i_2, \dots, i_n \text{ 是 } 1, 2, \dots, n \text{ 的一个排列}\}$ 。解的形式为：1, 2, ..., n 的一个排列，其中 n 为城市的数量。

2. 解的领域映射采用固定起点城市的 2-opt，即：固定起始点，后面的每两个城市进行对换进行邻域映射，邻域中的元素个数为 $C_n^2 + 1$ ；本文中采用 Hopfield/Tank 10 城市，则邻域中元素个数为： $(9 \times 8) / 2 + 1 = 37$ 。
3. 目标函数定义为巡回路径的城市间距离之和，目标函数同时作为评价函数；
4. 对于一般的禁忌搜索算法，禁忌对象的选取可以是简单解的变化，可以是解的分量的变化，还可以是目标值。对于本文的具体 TSP 实例特点，如果采用禁忌简单解的变化或者采用禁忌解的目标值，使得禁忌的范围较小，运算量大，容易陷入循环。所以，本实例中，禁忌对象定义为邻域映射时的城市交换，相当于对解的分量进行禁忌，试验结果证明达到了较好的效果；
5. 候选集元素数目、禁忌长度根据具体问题确定；本文中，针对 Hopfield/Tank 10 这一具体 TSP 问题，禁忌长度选为 11；候选集元素数目选为 12，即从邻域中 37 个元素中选择最佳的 12 个元素构成候选集；此处，选择候选集元素需要的计算量较大。
6. 为提高解的质量，防止出现循环，应用特赦规则。本文的特赦规则采用：当当前最优解未下降的次数超过给定值、或者当前候选集中的所有元素被禁时，则特赦禁忌表中的最优解，将其作为下一次迭代的初始解；
7. 终止规则：当程序运行超过给定最大搜索迭代步数，或当特赦次数超过给定最大特赦次数时，迭代搜索中止；
8. 为增强搜索空间的多样性，提高算法收敛到最优解的概率，本文应用多随机初始点策略，即分别以每一个城市作为初始点进行搜索且初始解可以随机选取。实验表明，能以较大概率收敛到最优解。

五、结果分析

本文利用禁忌搜索算法求解 Hopfield/Tank 10 这一具体 TSP 问题，城市各城市的坐标值来自文献[2]，具体数据如表 1 所示：

表 1: Hopfield/Tank 10 城市 (x, y) 坐标数据

City1=(0.4000, 0.4439)	City2=(0.2439, 0.1463)
City3=(0.1707, 0.2293)	City4=(0.2293, 0.7610)
City5=(0.5171, 0.9414)	City6=(0.8732, 0.6536)
City7=(0.6878, 0.5219)	City8=(0.8488, 0.3609)
City9=(0.6683, 0.2536)	City10=(0.6159, 0.2623)

本文采用多随机初始点策略。禁忌长度 11，候选集元素个数 12。基于传统 Hopfield/Tank 方法的神经网络求得的最好解是 2.89。本文的禁忌搜索算法求得的最优解为 2.6902(表 2)，好于已知的最优解，获优概率为 79%，平均寻优步数为：27。求解典型路径如表 2 所示：

表 2 求解典型路径

初始路径	典型路径最优解	路径目标值	寻优步数
1-2-4-5-6-9-8-7-3-10	1-4-5-6-7-8-9-10-2-3	2.690249	11
2-5-6-9-1-3-7-8-10-4	2-10-9-8-7-6-5-4-1-3	2.690249	7
3-6-5-2-1-7-4-8-9-10	3-2-10-9-8-7-6-5-4-1	2.690249	28
4-9-2-10-6-8-3-7-5-1	4-1-3-2-10-9-8-7-6-5	2.690249	9
5-6-3-7-8-2-4-1-10-9	5-4-1-3-2-10-9-8-7-6	2.690249	11
6-2-3-4-5-9-7-1-8-10	6-7-8-9-10-2-3-1-4-5	2.690249	32
7-5-8-4-2-6-9-3-1-10	7-6-5-4-1-3-2-10-9-8	2.690249	6
8-2-9-7-5-6-1-3-10-4	8-9-10-2-3-1-4-5-6-7	2.690249	13
9-7-2-5-1-4-8-10-6-3	9-8-7-6-5-4-1-3-2-10	2.690249	8
10-9-3-2-4-5-7-8-6-1	10-2-3-1-4-5-6-7-8-9	2.690249	4

典型寻优搜索过程及最优路径情况如图 1 所示。

六、小结

在本文的算法中，由于使用特赦规则、确定城市对换为禁忌对象、应用了多随机初始点策略等有效地避免了循环，使得算法能以较大概率收敛到最优解。从寻优过程图可以看出，禁忌搜索算法的寻优下降过程较快。在实际工作中，如果只要求获得满意解，则禁忌搜索算法将是一个强有力的搜索工具。然而，禁忌搜索算法同样也存在一些不足：

1. 解的质量、寻优步数与初始解有关，由表 2 数据：初始解选取不同，搜索收敛

于最优解的概率不同，获得最优解的搜索步数也不尽相同。

2. 在求解大规模 TSP 问题运行效率低。

3. 禁忌长度、候选集元素数目的大小的确定与具体问题相关。

4. 禁忌对象的选取与具体问题相关：禁忌对象可以是简单解的变化，可以是解分量的变化（体现在本文实例中是城市对换的禁忌），可以是目标值的变化。

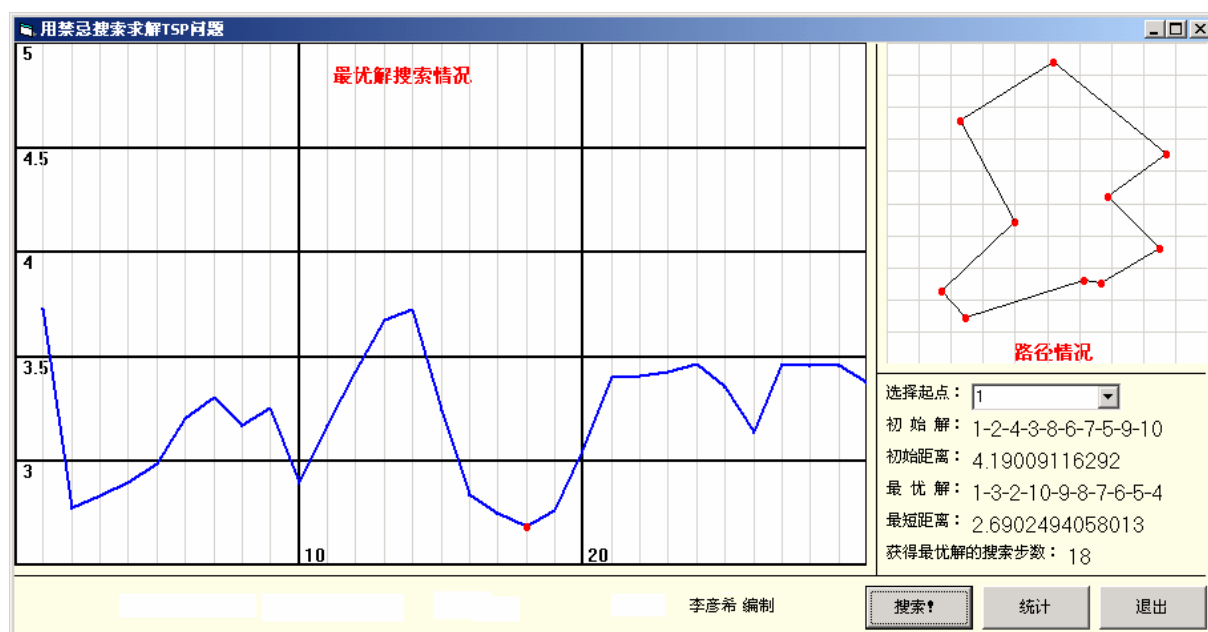


图1 典型寻优搜索过程及最优路径情况

参考文献:

- [1] 邢文训, 谢金星. 现代优化计算方法. 清华大学出版社. 1999
- [2] Hopfield J J, Tank D W. Neural computation of decision in optimization problems [J]. Biological Cybernetics, 1985, 52:141-152.

李彦希, 硕士, 桂林空军学院教育技术中心讲师, 目前主要从事网络应用研究及管理信息系统建设工作;

通信地址: 广西桂林市凯风路 90 号教育技术中心, 541003.

E-mail: gliethpaper@163.com;