

文章编号: 1671-9824(2017)02-0041-04

贪心算法在 TSP 问题中的应用

来学伟

(三门峡职业技术学院 信息传媒学院, 河南 三门峡 472000)

摘要: 用贪心算法对 TSP 问题进行求解, 首先介绍了贪心算法的设计思想和基本原理及基本过程, 然后利用两种贪心算法对 TSP 问题进行求解, 贪心算法可体现进化计算的优越性。

关键词: 贪心算法; TSP 问题; 算法

中图分类号: TP301.6 **文献标识码:** A

复杂问题通过贪心法被划分为 n 个相对比较容易的局部最优解问题。相对当前解来说, 任何一个解都是对当前解的扩展, 如此反复, 直到得到最完整的解。退一步来说, 即使得不到整体的最优解, 也是整体最优解的近似解^[1]。

1 贪心法的设计思想

从策略上来说, 贪心法在解决问题时比较浅显。仅仅根据当前获得的信息进行判断选择, 而且不考虑将来出现什么结果, 一旦这个判断选择确定, 就不会再更改。因为贪心法不从整体最优考虑, 因此, 它做出的选择判断仅仅是局部的最优。所有的局部选择判断并不是总能得到整体最优解。相对来说, 它可以得到近似最优解。针对只能通过蛮力法穷举获得的问题最优解, 贪心法不失为一个改进的好办法^[2]。

想法 1: TSP 问题的贪心策略可以采用最近邻点策略: 从任意城市出发, 每次在没有到过的城市中选择最近的一个, 直到经过了所有的城市, 最后回到出发城市。

如图 1 所示, 是一个无向图的代价矩阵, 从代价矩阵顶点 1 出发, 按照最近邻点的贪心策略, 得到的路径是 $1 \rightarrow 4 \rightarrow 3 \rightarrow 5 \rightarrow 2 \rightarrow 1$, 总代价是 14, 求解过程如图 2 所示:

$$C = \begin{pmatrix} \infty & 3 & 3 & 2 & 6 \\ 3 & \infty & 7 & 3 & 2 \\ 3 & 7 & \infty & 2 & 5 \\ 2 & 3 & 2 & \infty & 3 \\ 6 & 2 & 5 & 3 & \infty \end{pmatrix}$$

图 1 无向图的代价矩阵

算法 1: 设图 G 中 n 个顶点的编号为 $\{1, 2, \dots, n\}$, C_{ij} 表示顶点从 i 到顶点 j 的代价 ($1 \leq i, j \leq n$), 集合 V 存储图的顶点, 集合 P 存储经过的边, 从顶点 w 出发采用最近邻点策略求解 TSP 问题的算法如下。

最近邻点策略求解 TSP 问题

输入: 无项带权图 $G = (V, E)$, 顶点 w

收稿日期: 2016-03-23

作者简介: 来学伟 (1981—), 男, 河南灵宝人, 讲师, 硕士, 研究方向: 计算机软件开发。

输出:回路长度 TSPLength

1. 初始化: $P = \{ \}$; $TSPLength = 0$;
2. $u = w$; $V = V - \{w\}$;
3. 循环直到集合 P 中包含 $n - 1$ 条边
 - 3.1 查找与顶点 u 邻接的最小代价边 (u, v) 并且 v 属于集合 V ;
 - 3.2 $P = P + \{(u, v)\}$; $V = V - \{v\}$; $TSPLength = TSPLength + c_{uv}$;
 - 3.3 输出经过的路径 u
4. 输出 $TSPLength + c_{uv}$.

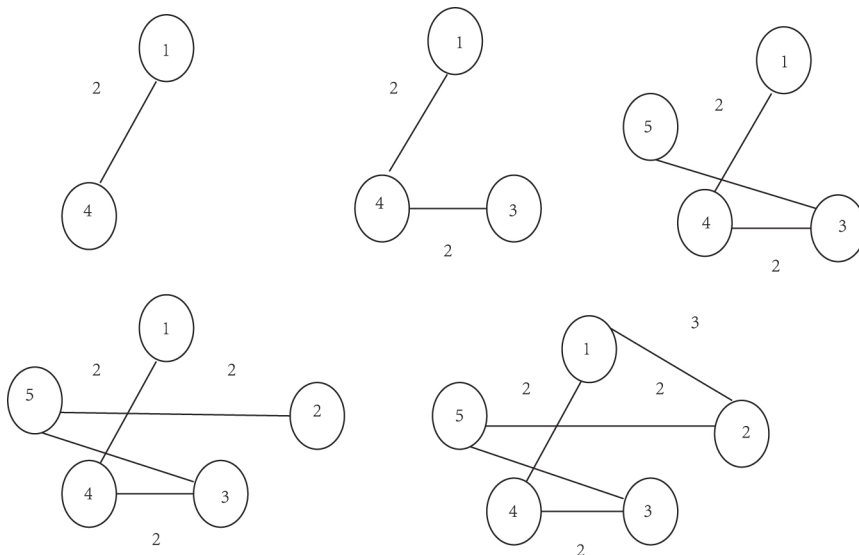


图 2 求解过程

算法分析:

经计算,该算法的时间性能为 $O(n^2)$,每选择一次都要经过 n 次查询满足条件的最短边,而总共要完成 $n - 1$ 次该操作,因此,该算法的时间性能是 $O(n^2)$.

特别指出的是,使用最近邻点贪心法得到的 TSP 问题的结果不一定是最优解.而且近似解类似于最优解的类似程度难于保证^[3].

算法实现:设函数 TSP1 实现最近邻点贪心策略求解 TSP 问题.数组 $arc[n][n]$ 存储图中各边的代价, $TSPLength$ 存储最短哈密顿回路(经过所有顶点一次且仅一次的回路)的长度.标志数组 $flag[n]$ 表示某顶点是否已加入哈密顿回路^[4].算法用 C++ 语言描述如下:

```
int TSP1 (int arc [n] [n] ,int w)
{
    int edgeCount = 0 ,TSPLength = 0;
    int min u ,v;
    int flag [n] = {0};
    u = w; flag [w] = 1;
    while ( edgeCount < n - 1 )
    {
        min = 100;
        for (int j = 0; j < n; j++)
        {
            if ( (flag [j] == 0) && (arc [u] [j] != 0) && (arc [u] [j] < min) )
            {
```

```

        v = j; min = arc[u][j];
    }
}
TSPLength += arc[u][v];
flag[v] = 1; edgeCount++;
cout << u << " - - " << v << endl;
u = v;
}
cout << v << " - - " << w << endl;
return (TSPLength + arc[u][w]);
}

```

想法2: TSP问题的贪心策略可采用最短连接策略: 每次在整个图的范围内选择最短边加入解集中, 但是, 要保证加入解集中的边最终形成一个哈密顿回路. 因此, 当从剩余边集 E' 中选择一条边 (u, v) 加入解集合 S 中, 应满足以下条件:

1. 边 (u, v) 是边集 E' 中代价最小的边;
2. 边 (u, v) 加入解集合 S 后, S 中不产生回路;
3. 边 (u, v) 加入解集合 S 后, S 中不产生分枝;

如图1所示, 是一个无向图的代价矩阵, 从代价矩阵顶点1出发, 按照最短链接的贪心策略, 得到的路径 $1 \rightarrow 4 \rightarrow 3 \rightarrow 5 \rightarrow 2 \rightarrow 1$, 总代价是14, 求解过程如图3所示.

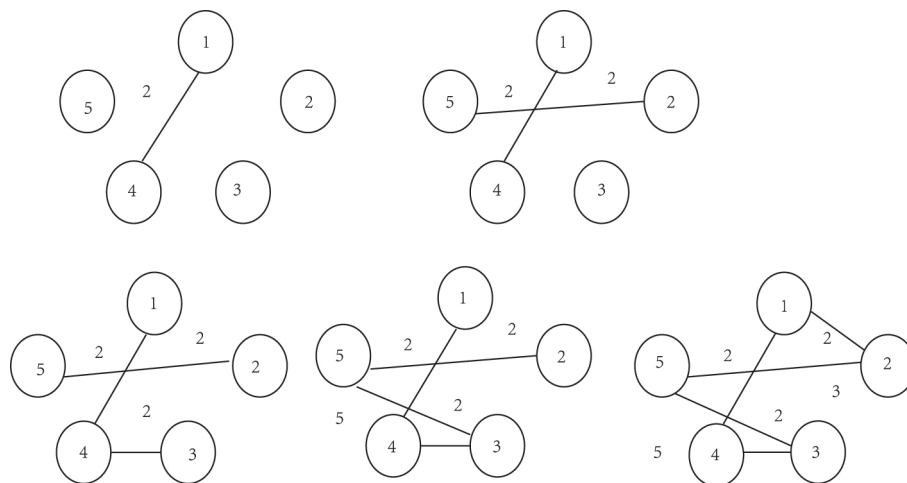


图3 求解过程

算法2: 设图 G 中 n 个顶点的编号为 $(1, 2, \dots, n)$, C_{ij} 表示顶点 i 到顶点 j 的代价 $(1 \leq i, j \leq n)$, 集合 E' 是候选边集合 (即所有未选取的边), 集合 P 存储经过的边, 最短链接策略求解 TSP 问题的算法如下.

算法: 最短链接策略求解 TSP 问题

输入: 无向带权图 $G = (V, E)$

输出: 回路长度 $TSPLength = 0$

1. 初始化: $P = \{ \}$; $TSPLength = 0$;
2. $E' = E$;
3. 循环直到集合 P 中包含 $n - 1$ 条边
 - 3.1 在 E' 中选取最短边 (u, v) ;
 - 3.2 $E' = E' - \{(u, v)\}$;
 - 3.3 如果 (顶点 u 和 v 在 P 中不连通 and 不产生分枝) 则

$$P = P + \{ (u, v) \}; \text{TSPLength} = \text{TSPLength} + c_{uv};$$

4. 输出 TSPLength.

算法分析 2

在上述算法中,共进行 $n-1$ 次贪心选择,在每一次贪心选择中,如果顺序查找 E' 中的最短边 (u, v) ,则算法的时间性能是 $O(n^2)$. 如果采用堆排序的方法将集合 E' 中的边建立堆,则可以将选取最短边的操作提高到 $O(\log_2 n)$ ^[5],可以用并查集的操作考察两个顶点是否连通以及是否会产生分枝,其时间性能为 $O(\log_2 n)$. 因此,算法的时间性能为 $O(n \log_2 n)$.

2 结语

通过对 TSP 问题的算法进行分析研究,我们是使用贪心算法,采用对图中各边代价进行编码的手段,得到 TSP 问题的求解.也就是说,准备到 n 个城市去旅游,开始从某一个城市出发,走完所有的城市,而且每个城市都只走一遍,最后回到开始出发的城市,得出总路程最短的方案.该算法降低了时间复杂性,而且对货物配送路线、电路布线等问题有实际的指导意义^[6].

参考文献:

- [1] 韩立. 基于学分制的排课算法的分析与设计[J]. 电脑编程技巧与维护, 2011(12): 139, 144.
- [2] 陆国锋. 基于多约束多目标的旅游路线推荐及关键算法研究[D]. 长沙: 国防科技大学, 2013.
- [3] 王楠. 基于模型诊断中测量方法的研究与实现[D]. 吉林: 吉林大学, 2006.
- [4] 宋磊. 覆盖阵列 CAN 上界的推导[D]. 上海: 上海交通大学, 2008.
- [5] 程美英, 钱乾. 连续空间二元粒子群算法理论研究综述[J]. 计算机应用研究, 2012, 30(4): 983.
- [6] 吴陈, 张吉缘, 王绍雷. 旅行商问题的动态规划与子集编码求解法的研究与实现[J]. 信息与电脑, 2015(18): 52.

Application of Probabilistic Algorithm in the TSP problem

LAI Xuewei

(College of Information Media, Sanmenxia Vocational-Technical College, Sanmenxia 472000, China)

Abstract: To solve TSP problem applying the ant colony algorithm, firstly the paper introduces the ant colony algorithm design idea and basic principle and basic process, and then uses two ant colony algorithm to solve TSP problem. The ant colony algorithm can reflect the evolutionary computation superiority.

Keywords: ant colony algorithm; TSP problem; algorithm

责任编辑: 赵秋宇