2017 中兴捧月杯 Dijkstra 派论文

刘航, 徐安, 陈安志

摘要：根据$Dijkstra$派赛题，本文将问题抽象为：给定一个无向图G，已知起点s、终点e、必经点集R、必经边集B、禁忌边集F和访问节点数限制k,要求寻找一条从点s到点e的最短路径，同时满足以下几个限制条件:（1）不能经过禁忌边集F中的任意边（2）必须经过必经点集R中的所有顶点（3）必须经过必经边集B中的所有边（4）路径访问的节点数量不超过k。本文通过图压缩的方法将此问题转化为旅行商问题（TSP），并采用两种方法求解TSP模型：指派模型结合分支定界；$lpsovle$求解器结合线性规划。实验证明，本文所采取的方法能高效解决上述问题，得到满足条件的最优解。

关键词：TSP；分支定界；指派问题；线性规划；$lpsovle$

1 问题模型构建与分析

本文将赛题抽象为一类通用问题：已知无向图$G=(V,A)$，V为顶点集，A为边集，求从指定起点到达终点的最短路径，该路径满足以下四个限制条件：（1）不能经过禁忌边集F中的任意边（2）必须经过必经点集R中的所有顶点（3）必须经过必经边集B中的所有边（4）路径访问的节点数量不超过k。图中的每个节点可以访问多次，对同一节点的重复访问按重复次数计算。

针对限制条件（1），由于禁忌边是不允许访问的边，本文的处理方法是直接将禁忌边从图中删除，即可保证求解出来的路径不包含任何禁忌边。

针对限制条件（2），本文的基本思路是将其转化为旅行商问题（Traveling Salesman Problem, TSP）。旅行商问题即给定一系列城市和每对城市之间的距离（一个无向图），求解访问每座城市一次并回到起始城市的最短回路。本问题与TSP有几点不同：①本问题中每个节点可以访问多次，而TSP模型中每个节点可只能访问一次；②本问题中只有部分节点是必须经过，TSP是要经过图中的所有节点；③本问题中是到达指定终点，TSP是回到起点。根据这三点不同，本文的转化方法是对原图进行压缩，建立一个新图，对新图的TSP求解等价于对本问题的求解。

原图的压缩方法如下：对于原图中所有必经节点（包括起点和终点），求出每两个的节点间的最短路径，建立一个新图，新图中只包含这些必经节点，每两个节点之间的边权值为所求最短路径权值；对终点进行处理，增加一条从终点到起点的边，并让从终点出来的边只能到达起点。经过图压缩后，新图中的所有顶点都是必经节点，且到达终点后只能回到起点，因此，得到的新图完全满足TSP模型。对新图求解TSP，每个必经节点只经过了一次，与原问题节点可重复经过不符，但可以证明；新图 TSP 模型的最优解一定也是原问题的最优解。证明如下：

第一，在原问题的最优解中，任何两个相邻必经节点之间的访问路径，一定是这两个必经节点的最短路径。这是显而易见的。如果求解的路径中两个必经节点间的访问路径不是最短路径，那么用最短路径代替它，一定可以得到更短的路径。这里由于每个节点可以重复经过，所以不存在最短路径被占用的问题。第二，压缩后的新图是一个满足三角不等式的完全图，即图中任意两个节点间存在路径，且图中任意三个顶点构成的三条边满足两边之和大于等于第三边。这个性质可以根据前文提到的压缩方法得出。第三，对于满足三角不等式的完全图，遍历所有点的最短回路中，一定不存在重复节点。这个性质可以用反证法证明：假设路径中存在重复结点，如$a-b-c$路径段，b为重复节点，由于图为完全图，且满

足三角不等式即$bc \le ab + bc$，则将此路径段用路径$a - c$代替可以得到更短的路径。根据以上三点性质以及前文所述压缩图的方法，可以证明新图TSP模型的最优解一定也是原问题的最优解。

针对限制条件(3)，本文采取的方式是将必经边转换为必经点：①将必经边的两个端点设置为必经点；②在该必经边上加一个节点，并将这个节点设置为必经点。在TSP模型中，只要路径经过这个新增的必经点，则路径一定经过这条必经边。经过转化，限制条件（3）可以规约到限制条件（2）。

针对限制条件（4），本文采用逐次迭代增加边权重的方法来解决。具体如下：首先求出满足限制条件（1）（2）（3）的最优解，如果得到的路径节点数大于最大节点数k，则将无向图中的每条边都增加 1，然后继续求解满足限制条件（1）到（3）的解，如此迭代，直到求得的解路径的节点数少于k（即满足限制（4））。这样迭代的原理是：图中的每个边的权重都增加后，节点数多的路径，总路径权重值增加的多，这样迭代会选出节点数越来越少的路径。

综上所述，本文求解问题的方法如图 1（处理限制条件（1）到（3））和图 2 所示（处理限制条件（4））。

2 指派模型结合分支定界求解 TSP

2.1 数学模型

给定一个无向图$G = (V, A)$，其顶点集合$V = \{1, \ldots n\}$，边集合$A = \{(i,j)|i,j = 1, \ldots n\}$，$c_{ij}$表示对应的边的费用，则$TSP$模型可以表示为以下的形式[1]：

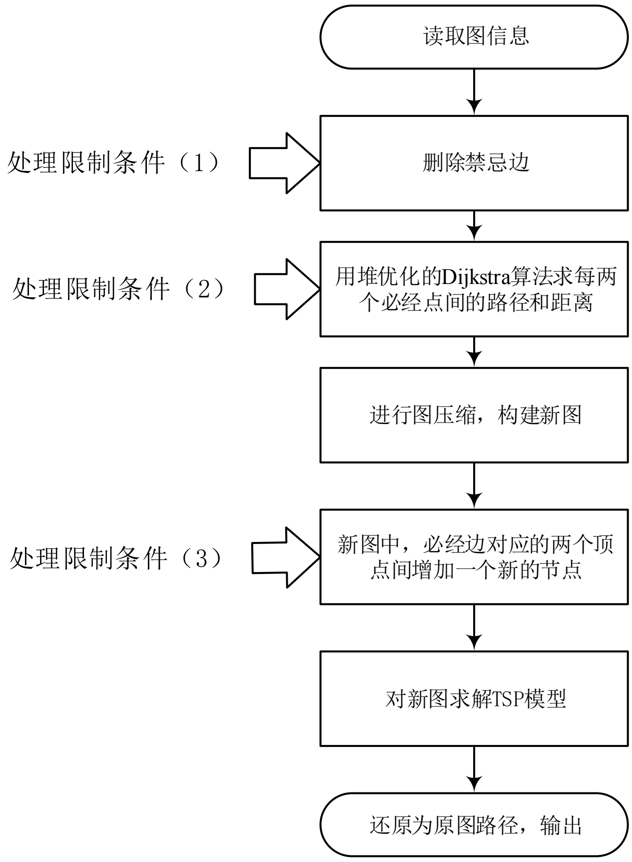

图 1 算法流程 1，处理限制条件（1）到（3）

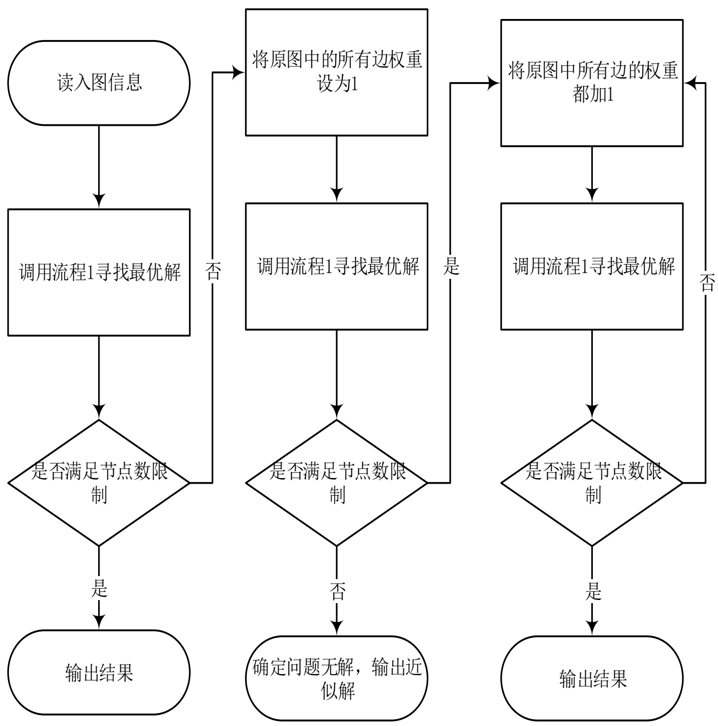

图 2 算法流程 2，处理限制条件（4）

$$\text{Min} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \tag{1}$$

$$\sum_{i \in V} x_{ij} = 1, \qquad \forall j \in V \tag{2}$$

$$\sum_{j \in V} x_{ij} = 1, \qquad \forall i \in V \tag{3}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \le |S| - 1, \qquad \forall S \subset V, S \neq \emptyset \tag{4}$$

$$x_{ij} \in \{0,1\}, \qquad \forall\, i, j \in V \tag{5}$$

其中，方程（1）为模型的目标函数，方程（2）到（5）为约束条件，满足约束条件的一个可行解是，找到一个访问了图中所有顶点的环，即哈密顿环[2]。如果边出现在解中（即路径中出现从顶点i到顶点j），则$x_{ij} = 1$，否则，$x_{ij} = 0$，最优解是找到一个成本最小的哈密顿环。如图 3，

	①	②	③	④	⑤	⑥	⑦
①	0	53	2	18	39	27	22
②	53	0	55	43	64	52	48
③	2	55	0	20	41	29	24
④	18	43	20	0	29	24	15
⑤	39	64	41	29	0	29	36
⑥	27	52	29	24	29	0	9
⑦	22	48	24	15	36	9	0

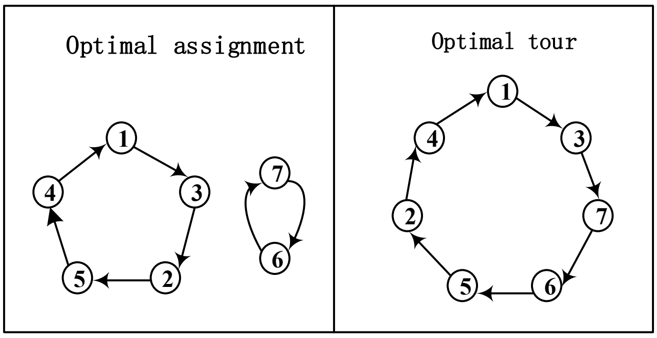

图 3 含有 7 个点的模型示例，表格为费用矩阵，方框中为指派问题的最优解（val(AP) = 168）和 TSP 最优解（val(TSP) = 171）

包含一个费用矩阵的示例，其TSP最优路径如右侧方框中所示。这个例子将被用作解释整个算法。

方程（1）（2）（3）（5），可以当作一个最小指派问题(Assignment Problem，AP)[3]的模型，方程（4）表示只能有一个环，约束的解释如图 4 所示。

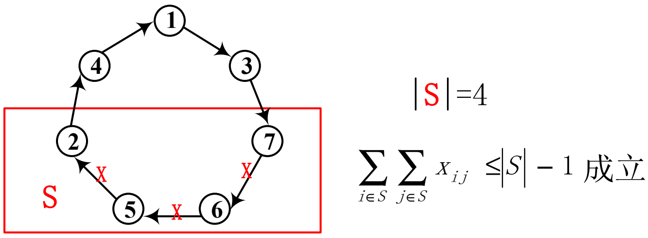

图 4 只形成一个环约束示意图

用$val(AP)$表示最小指派问题的解（即路径费用和）。指派模型可以保证每个点都被访问，且只访问一次（即约束方程（2）（3）），但不能保证路径是一条连续的环，它可能有多个环；最小指派问题的解与TSP问题的最优解之间存在如下关系：

$$val(AP) \le val(TSP), \qquad \forall\, i, j \in V \tag{6}$$

这是因为TSP模型比AP模型多了一个约束条件，即TSP模型要求只形成一个换（约束方程（4））；如图 3 所示的例子，最小指派的解为 168，而TSP模型的解为 171。

根据前面的讨论，我们可以把TSP模型转化为AP模型，而AP模型可能形成多个环，本文采用分支定界的方法来解决多环问题。整个算法的流程如下：

（1）根据图的信息，得到费用矩阵；

（2）由费用矩阵求解AP模型，并计算求解后的环数量；

（3）若环的数量等于 1，则求解成功；若环的数量大于 1，则用分支定界法，进行分支和定界，继续求解AP模型，直到求解成功。

2.2 指派问题求解

指派问题是在加权二分图中寻找最大或最小加权匹配的问题，本文中用到的是最小指派问题。指派问题可以在多项式时间内求解，匈牙利算法就是一种可以在多项式时间内求解指派问题的组合优化算法，它由美国数学家哈罗德·库恩于 1955 年提出。

匈牙利算法的时间复杂度为$O(n^4)$，但后来多位科学家发现可以修改算法达到$O(n^3)$运行时间，此后该算法被称为 Kuhn-Munkres 算法（即

KM 算法)。

本文求解AP模型所用的方法就是KM算法，关于KM算法的原理和具体实现，本文限于篇幅，不作具体说明，详见参考文献[4](组员刘航的博客总结)。

2.3 分支定界

分支定界法是一种经典的在解空间进行枚举搜索的技术，我们可以用它得到TSP问题的最优解。

2.3.1 下界

根据方程(1.6)，AP模型求解的值肯定小于TSP模型的最优解，因此，对费用矩阵求解一次AP模型之后，得到的解就是整个解空间的下界。

求解一次AP模型后，若得到多个环，则对节点数最少的环进行分支，对这个环上的每个边，分别禁止它出现在下一次求AP模型求解的路径上（即进行破坏处理）。由此可以得到一颗解搜索树，搜索树上的每个解节点包含上次AP模型求解的值和下次AP求解时应该破开的边。在这颗搜索树上进行搜索，可以得到TSP问题的最优解。

2.3.2 上界

在某一次AP模型求解之后，若只存在一个环，则这次AP求得的解（路径费用和）为最优解的上界。对于解空间中，$val(AP)$大于上界的那些解节点，不必再往下进行分支，因为分支后的点求解出来的值只会比上界大。根据上界的值，我们可以避免不必要的分支，极大的提高了算法的性能，更快的找到最优解。

2.3.3 分支规则

本算法最核心的地方在于分支规则。定义必须边集合（Required Arc Set）和禁止边集合（Forbidden Arc Set），禁止边集合包含禁止出现在下次AP求解路径上的边，必须边集合包含必须出现在下次AP求解路径上的边。

一次AP求解之后，对节点数最少的环进行分支，对这个环的每个边，将它加入禁止边集合，得到一个新的解节点，为了防止重复搜索，对前一个加入禁止边集合的边，本次分支时将它加入必须边集合。如图 5，一次AP求解之后，得到解节点$vertex\ 0$，它包含两个环，环$6-7-6$节点数最少，对它进行分支。首先，禁止边(7,6)，求解AP模型；然后，禁止边(6,7)，同时让边(7,6)为必须边，再进行AP模型求解。

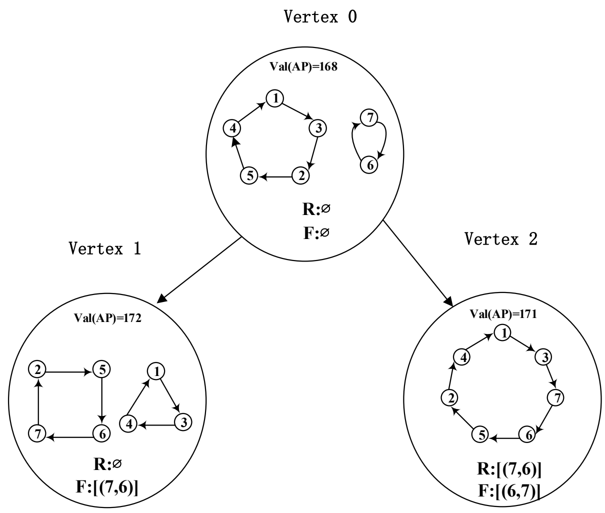

图 5 图 3 例子的解节点搜索树。每个解节点包含了AP模型的路径、必经边集合（R）、禁止边集合（F）

图 5 为整个分支定界求解的过程示意图，首先，对费用矩阵求解AP模型，得到路径费用和$val(AP)=168$，然后对只含有两个点的环进行分支，得到$vertex\ 1$和$vertex\ 2$，$vertex\ 1$ 求解AP模型后，得到包含两个环的解，$val(AP)=172$；vertex 2 求解 AP 模型后，得到只包含一个环的解，此时$val(AP)=171$，由此更新上界为 171。由于$vertex\ 1$的费用和已经高于上界，所以不需要继续往下分支，因此得到TSP问题最优解为171，路径如图中$vertex\ 2$所示。

3 线性规划求解 TSP

求解TSP有很多种算法，其中，线性规划是一种非常有效的方法。目前市面上存在许多开源和商业求解器，可以非常高效的求解线性规划问题。因此，本文对 TSP 问题提供线性规划的解法。

3.1 数学模型

TSP 可以被规范化为一个整数规划问题：

$$\text{Min} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (7)$$

$$\sum_{i \in V} x_{ij} = 1, \quad \forall j \in V \quad (8)$$

$$\sum_{j \in V} x_{ij} = 1, \quad \forall i \in V \quad (9)$$

$$u_i \in Z \quad i = 1, \dots, n$$

$$u_i - u_j + n * x_{ij} \le n - 1 \quad 2 \le i \ne j \le n \quad (10)$$

$$x_{ij} \in \{0,1\}, \quad \forall\, i, j \in V \quad (11)$$

其中，u_i是辅助变量，记录访问节点的先后顺序，表示第i个节点是第几个访问的。此整数规划模型与 2.1 节所述模型基本相同，区别在于对只有一个环的约束。本模型中，约束（10）限定了只存在一个环，证明如下：

如果某个环中不包含城市 1，设这个环中的节点为$i, \dots, i+k-1$，令$x_{i,i+1} = x_{i+1,i+2} = \cdots = x_{i+k-1,i} = 1$，则有$k$个满足方程（10）的不等式：

$$u_i - u_{i+1} \le n - 1$$

$$u_i - u_{i+1} \le n - 1$$

$$\dots$$

$$u_i - u_{i+1} \le n - 1$$

叠加后，得：$nk \le (n-1)k$，产生矛盾。因此，所有的环必须包含节点 1，所以满足方程（10）的可行解只能形成一个环。

3.2 模型求解

在 3.1 节所述模型中，一共使用了$n^2 + n$个变量，$n^2 + 2n$个约束条件。它是一个整数规划的模型，即所有未知数都为整数的线性规划问题。

求解线性规划可以使用求解器，求解器分商业版和开源版，例如 cplex、Gurobi、XPRESS、lpsolve 等。考虑到版权等问题，本文使用 lpsolve 求解器[5]来解决上述整数规划问题。Lpsolve 是一款开源求解器，求解速度中等偏上。（实际上，我们也使用了 Gurobi 求解器来测试，发现速度比 Lpsolve 快很多。但是 Gurobi 是一款商业求解器，考虑到版权问题，我们没有提供 Gurobi 版的代码。）

4 总结

本文把赛题抽象，归为一类通用问题，并将这类问题转化为TSP模型，并证明了这种转化的正确性。在此基础上，本文采用两种方法求解TSP：第一，将TSP转化为指派问题模型，结合分支定界法来求解；第二，利用线性规划求解TSP，并用开源求解器lpsolve求解线性规划。

参考文献：

[1] Miller D L, Pekny J F. Exact Solution of Large Asymmetric Traveling Salesman Problems[J]. Science. 1991, 251(4995):754-761.

[2] https://en.wikipedia.org/wiki/Hamiltonian_path，哈密顿环

[3] https://en.wikipedia.org/wiki/Assignment_problem,指派问题（任务分配问题）

[4] http://liuhang.net.cn, KM 算法原理和具体实现

[5] http://lpsolve.sourceforge.net/5.5/, lpsolve 求解器

说明：

算法的性能测试和用例设计分析在文档《程序说明及用例分析测试.pdf》中，**一定要看呀！**

团队成员信息及分工在文件夹根目录中的《说明.txt》中。