分类号 学号 M201672963

学校代码 10487 密级



**硕士学位论文**

**启发式算法求解带必经节点最短路问题研究**

|  |  |
| --- | --- |
| 学位申请人： | 张 军 琛 |
| 学科专业： | 计算机技术 |
| 指导教师： | 吕志鹏 教授 |
| 答辩日期： | 20180522 |

**A Dissertation Submitted in Partial Fulfillment of the Requirements**

**for the Degree of Master of Engineering**

**A Heuristic Algorithm for Shortest Path Problem with Must-Pass Nodes**

|  |  |
| --- | --- |
| **Candidate :** | **Zhang Junchen** |
| **Major :** | **Computer Technology** |
| **Supervisor :** | **Prof. Lv Zhipeng** |
|  |  |

**Huazhong University of Science & Technology**

**Wuhan 430074, P. R. China**

**May, 2018**

独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密□，在 年解密后适用本授权书。

本论文属于

不保密□。

（请在以上方框内打“√”）

学位论文作者签名： 指导教师签名：

日期： 年 月 日 日期： 年 月 日

# 摘 要

带必经节点的最短路问题，是在网络拓扑图中找出一条从给定起点到终点，必须访问给定必经点集合中任一节点的最短路径，属于路由问题或约束最短路问题。该问题主要出现在工业界带约束的网络资源规划问题中，在学术界目前并未得到足够的研究重视，研究成果较为有限。当必经点个数大于1时，带必经节点的最短路问题属于NP难问题，主要难点在于求解必经节点的顺序之外，还需要确定相邻必经节点之间的路径选取；必经节点之间的路径选取反过来又会影响必经节点顺序的确定，两者是相互耦合的。

本文提出一种基于k-opt边交换、候选路搜索和节点属性提升等多策略优化的多阶段启发式算法。本文先将原始问题转换为必经节点集合对应的TSP问题，使用求解TSP问题的算法确定最优必经节点顺序；相邻必经节点依靠最短路连接，如果路径中有节点被重复访问，即存在违反简单路径约束的冲突节点，则在相关必经节点之间搜索k短路作为候选路径取代最短路径，以消除冲突节点；如果搜索候选路仍无法消除冲突节点，将冲突节点属性提升为必经节点重新进行求解。通过多个阶段优化，求解得到一条在时间开销和最优性折中平衡的路径。通过与不同类型的数学规划模型求解结果对比，在随机生成以及具有特殊结构的算例集上求解测试，结果表明该启发式算法具有很高的效率和求解性能。

**关键词：**路由问题，必经节点，最短路，启发式算法，旅行商问题

# Abstract

The Shortest Path Problem with Must-Pass Nodes is to find a shortest path from the given starting node to end node in network diagram, and must access any node in a given must pass nodes set, which belongs to routing problem or constraint shortest path problem. This problem has mainly appeared in the network resource planning with constraints in industry, and it has not received sufficient research attention in academic community. The research results are limited. When the number of must pass nodes is greater than 1, the Shortest Path Problem with Must-Pass Nodes is NP-hard. The main difficulty lies in solution of the sequence of nodes that must be passed, and path selection between adjacent must pass nodes. The selection of paths between must pass nodes in turn affects determination of order of must pass nodes, and the two are coupled.

This paper presents a multi-stage heuristic algorithm based on multi-strategy such as k-opt move, candidate search and node attribute promotion. In this paper, we first convert the original problem into a TSP problem about must pass nodes. The algorithm for solving TSP problem is used to determine optimal order of must pass nodes. The neighboring must pass nodes linked by shortest path, if there are nodes that are repeatedly accessed in the path which violates the simple path constraint, a k-shortest paths is searched for related adjacent must pass nodes as candidate paths to replace the shortest path to eliminate conflicting nodes. If searching for a candidate paths still fails to eliminate the conflicting nodes, these conflicting nodes attribute is promoted as must pass nodes. Through multiple stages of optimization, the solution is to obtain a path that balances the time cost and the optimality. Comparing with the results of different types of mathematical programming models, the tests are performed on randomly generated and special set of cases. The results show that the heuristic algorithm has high efficiency and performance.

**Key words：**routing problem, must-pass nodes, shortest path, heuristic algorithm, travelling salesmen problem

# 目录

摘 要 I

Abstract II

目录 III

第1章 绪论 1

1.1 课题背景及意义 1

1.2 问题数学描述 2

1.3 国内外研究现状 3

1.4 本文主要研究内容 4

第2章 相关理论知识 6

2.1 启发式算法描述 6

2.2 局部搜索 6

2.3 变邻域搜索 6

2.4 TSP问题概述 6

2.5 问题数学建模 7

2.6 本章小结 10

第3章 基于多策略多阶段优化的启发式算法 11

3.1 算法主框架 11

3.2 和TSP问题相互转化 12

3.3 求解TSP问题 14

3.4 候选路搜索 16

3.5 节点属性提升 17

3.6 本章小结 18

第4章 实验结果分析 19

4.1 算例集生成介绍 19

4.2 数学模型结果比较 20

4.3 启发式算法结果比较 21

4.4 本章小结 21

第5章 算法分析与讨论 22

5.1 增量评估技术 22

5.2 多邻域组合 22

5.3 本章小结 22

第6章 总结与展望 23

6.1 工作总结 23

6.2 不足与展望 23

致谢 24

参考文献 25

附录作者攻读硕士学位期间发表的论文 26

# 第1章 绪论

## 1.1 课题背景及意义

带必经节点的最短路问题，主要来源于工业界。比如通信运营商的大多数业务都是路由问题，在网络资源优化配置中，部分节点会具有一些特殊的性质(如可靠性、是否中继等)，在具体点到点的业务中，受限于资源、时间、容量等条件约束，需要在一条业务中访问这些具有特殊性质的节点；实际网络中，物流配送、路线规划等问题也可以抽象为必经节点最短路问题，如取货送货地点可以抽象为必经节点，节点之间的路径上存在相应权重，目标是求解一条满足所有约束的权重之和最小的路径。在运营商的保护业务中，为了增强网络系统的高可用性，如果原来的节点发生故障，则需要启动备份节点，相当于在网络中寻路。该问题的复杂度不比TSP问题简单，至少要难于必经节点数目的TSP问题。 通过文献查阅，我们发现该问题的研究成果十分欠缺，但和该问题相关的TSP问题、以及带时间窗或资源约束的TSP变种问题的研究成果颇为丰富。

带必经节点的最短路问题(Shortest Path Problem with Must-Pass Nodes, SPPMPN)，主要来源于工业界。比如通信运营商的大多数业务都是路由问题，在网络资源优化配置中，部分节点会具有一些特殊的性质(如可靠性、是否中继等)，在具体点到点的业务中，受限于资源、时间、容量等条件约束，需要在一条业务中访问这些具有特殊性质的节点；实际网络中，物流配送、路线规划等问题也可以抽象为必经节点最短路问题，如取货送货地点可以抽象为必经节点，节点之间的路径上存在相应权重，目标是求解一条满足所有约束的权重之和最小的路径。运营商为了保证通信网络系统的高可用性，提出保护业务方案，如果业务中原路径节点发生故障，则需要启用备份路径节点，因此业务中原路径和备份路径都需要经过一些指定的节点。该问题的复杂度不比TSP问题简单，至少要难于必经节点数目的TSP问题。 通过文献查阅，我们发现该问题的研究成果十分欠缺，但和该问题相关的TSP问题、以及带时间窗或资源约束的TSP变种问题的研究成果颇为丰富。

通过文献查阅分析，SPPMPN问题最早由Saksena和Kumar[1966,]提出，文献提及了必经节点约束的必要性，同时他们也提出了基于最优性原理的算法；但是，Dreyfus[1969,]指出Saksena和Kumar的算法是不正确的，并证明了该算法的前提假设是错误的；最后他表示如果去掉简单路径的约束(即路径中可以出现回环)，则可以将SPPMPN问题转换为TSP问题进行求解，新TSP问题的维度为必经点个数+，2代表原问题中起点和终点；首先计算各必经节点以及起点终点之间的最短距离作为新问题中各节点的距离，则可以利用求解TSP问题的算法。Ibaraki[1973, ]提出了解决必经节点最短路问题的动态规划算法和分支限界算法，在文献中他指出分支限界算法比动态规划算法更高效，但是，由于两者都是精确求解算法，求解大规模问题时间开销太大。Vardhan[2009,]等人提出了解决SPPMPN问题的启发式算法，他们将求解过程分为两个阶段，第一阶段搜索必经节点的之间的所有候选路径，通过随机给定或从起点到终点通过DFS算法来求得必经节点的顺序，对于相邻必经节点之间的路径，通过网络流算法求出不经过其他必经节点的所有候选路；第二阶段采用回溯算法选择相邻必经节点的候选路径，最终得到一条满足约束的简单路径，该算法在求解候选路的过程使用网络流算法求出相邻必经节点之间的所有候选路，回溯选取候选路的时候如果出现回环时可以考虑必经点之间的其他候选路，搜索空间比较完整，但是对于必经节点的顺序选取过于简单，对一些算例该算法不能保证能求解得到最优解，在具有特殊网络拓扑图的算例甚至不能找到合法解。Gomes[2015,]等人提出了求解带保护路径的必经节点最短路问题的启发式算法，该算法一次求解相同起点和终点的两条节点分离路径，第一条路径必须满足必经节点的约束，他们在Saksena和Kumar算法的基础上做出了改进，采用k-最短路算法搜索必经节点之间的多条候选路径，但是该算法并不能保证每次都能得到合法解，并且他们的目标是得到两条节点分离的路径。Andrade[2013,2016]提出了求解该问题的数学规划模型，首先他基于生成树多边形提出了Q2模型，接着着重分析阐述了基于对偶原理的紧凑模型Q3，他对部分随机生成的算例和由TSPLIB改造的算例上进行了求解，分析模型的求解性能，指出Q3性能远优于Q2。

## 1.2 问题数学描述

带必经节点的最短路问题，数学定义如下： 给定一个有向带权图，节点集合，边集合，*E*中每条边上有非负权重，给出起点和终点，给定必经点集合，寻找一条简单路径，每一个节点都被访问一次，对于*T*中的每一条边。非必经节点，该有向图并非完全图，节点和节点之间没有边时。

如果必经节点集合包含起点终点外的所有其他节点，即满足条件，且假设起点到终点的权重为0，则SPPMPN问题转化为经典旅行商人问题(travelling salesman problem, TSP)；TSP问题是在一个完全图中找出一条访问每一个节点一次且权重之和最小的环路，即权重最小的哈密顿回路；TSP 问题是NP难问题。如果必经节点集合为空，即满足条件，则该问题转化为求节点对的最短路问题。SPPMPN问题难度至少和维度为的TSP问题相同。

## 1.3 国内外研究现状

带必经节点最短路问题最早由Saksena和Kumar[1966,]提出，文献提及了必经节点约束的必要性，同时他们也提出了基于最优性原理的算法；但是，Dreyfus[1969,]指出了该算法的问题所在，证明该算法的前提假设是错误的，最后他提出如果去掉简单路径的约束(即路径中可以出现回环)则可以将问题转换为TSP问题进行求解，新问题的维度为|S|+2，对应原问题中必经点、起点和终点；首先计算各必经节点以及起点终点之间的最短距离作为新问题中各节点的距离，则可以利用求解TSP问题的算法。Ibaraki[1973, ]提出了解决必经节点最短路问题的动态规划算法和分支限界算法，在文献中他指出分支限界算法比动态规划算法更高效，但是，由于两者都是精确求解算法，时间开销太大。Vardhan[2009,]等人提出了解决该问题的启发式算法，他们将问题的求解过程分为两个阶段，第一阶段来确定必经节点的之间的候选路径，通过随机给定或从起点到终点通过DFS算法来求得必经节点的顺序，对于相邻必经节点之间的路径，通过网络流算法求出不经过其他必经节点的所有候选路；第二阶段采用回溯算法选择相邻必经节点的候选路径，最终得到一条满足约束的简单路径，该算法在求解候选路的过程使用网络流算法求出相邻必经节点之间的所有候选路，回溯选取候选路的时候如果出现回环时可以考虑必经点之间的其他候选路，搜索空间比较完整，但是对于必经节点的顺序选取过于简单，对一些算例该算法不能保证能求解得到最优解，在具有特殊网络拓扑图的算例甚至不能找到合法解。Gomes[2015,]等人提出了求解带保护路径的必经节点最短路问题的启发式算法，该算法一次求解相同起点和终点的两条节点分离路径，第一条路径必须满足必经节点的约束，他们在Saksena和Kumar算法的基础上做出了改进，采用k-最短路算法搜索必经节点之间的多条候选路径，但是该算法并不能保证每次都能得到合法解。Andrade[2013,2016]提出了求解该问题的数学规划模型，首先他基于生成树多边形提出了Q2模型，接着着重分析阐述了基于对偶原理的紧凑模型Q3，他对部分随机生成的算例和由TSPLIB改造的算例上进行了求解，分析模型的求解性能，指出Q3性能远优于Q2。

## 1.4 本文主要研究内容

本论文主要针对硬密封结构的界面泄漏特性，采用数值方法以及理论分析，完成了以下主要内容：

第一章，主要分析了国内外在粗糙表面形貌特征表征、界面泄漏流动模型以及粗糙表面微观接触模型等相关领域的研究进展，并确定本文的章节安排。

第二章，介绍了基于随机过程理论的粗糙表面表征方法，并通过基于AR模型的二维数值滤波技术，生成了满足Gauss分布和指数自相关函数的数值粗糙表面。然后，分析了粗糙度和自相关长度对粗糙表面形貌特征的影响，以及具有加工纹理的粗糙表面的表征方法。

第三章，在已有的数值粗糙表面的基础上，使用三维图形软件构建了粗糙界面泄漏通道的几何模型。然后，采用CFD软件对泄漏通道内的流动进行数值模拟，得到相应的泄漏率。最后，定义粗糙通道与对应相同间隙高度的平直通道的泄漏率的比值为流量因子，并对其进行负指数函数形式的无量纲拟合，用以表征粗糙度对泄漏率的影响。

第四章，为了研究密封间隙高度与密封比压的关系，对粗糙表面进行了相应简化和假设。通过有限元分析软件模拟了微观粗糙峰的接触行为，得到了Roth模型中的密封系数。获得了以Roth模型为基础的密封间隙高度的求解方法。

第五章，结合前面章节的内容，给出了界面泄漏的理论预测模型。通过与其他文献中的试验数据进行比较，论证了该预测模型的有效性。最后，利用该模型分析了泄漏率的影响因素及若干提高密封性能的建议。

第六章，总结了全文研究内容及成果，并指出了现有工作的一些不足之处。

# 第2章 相关理论知识

## 2.1 启发式算法描述

## 2.2 局部搜索

## 2.3 变邻域搜索

## 2.4 TSP问题概述

TSP问题是非常著名的NP困难组合优化问题，在学术界有着非常广泛的研究，最早可以追溯到19世纪30年代。TSP问题是在一个完全图中找出权值最小的哈密顿回路。本文涉及的TSP问题都为对称TSP()问题，非对称TSP问题可以通过节点数翻倍转化为对称TSP问题。我们定义决策变量表示边是否出现在解路径中，TSP问题的经典数学规划模型如下：

*Minimize* 

subject to



文献()对比了多种不同的数学规划模型，但是对于大规模的TSP问题，由于指数级别的约束或复杂度，数学规划模型方法时间开销太大，在实际应用中难以广泛使用。

通过文献查阅，我们发现有非常多的近似算法和启发式算法用于求解TSP问题，最广泛且最有效的算法是基于边交换的2-opt算法，算法思想非常简单，是在已有的路径上删除两条边，将分割后两段路径再重新连接起来形成一条权重之和更小的环路。2-opt是一种路径改进算法，在已有路径上通过边交换策略来改进路径，如果发现有更好的解，执行该2-opt动作，在改进的路径上继续搜索。2-opt简单高效，但是也因为邻域太小而容易陷入局部最优，后来又相继提出了3-opt和k-opt等启发式算法，Lin和Kernight实现了非常高效的k-opt局部搜索算法。

根据已发布文献查证，目前求解TSP问题最高效的算法程序有：基于精确算法的Concorde求解器，基于局部搜索的LKH(LK Heuristic)算法，基于Edge Assembly

Crossover算符的遗传算法。Concorde可以求解的问题最大规模是89500，LKH可以求解的问题规模在欧几里得平面图上可达10,000 to 10,000,000，EAX算法可求解的规模达到200,000。

## 2.5 问题数学建模

带必经节点最短路问题的极限形式是TSP问题，通过查阅文献，借鉴求解TSP问题的数学规划模型方法，我们在这里实现了五个不同的数学规划模型，部分模型已经由其他学者在文献中发布，我们在这里列举出来主要是因为不同的模型具有不同的特性，我们在算例集上可以显示各个模型的求解结果差异，一方面比较模型的求解性能，另一方面有助于我们推断一些特殊结构的算例的特性，用于测试和验证我们后面提出的启发式算法的性能。

每一个数学规划模型都满足一些基本的节点度的约束，例如起点只能有出度，终点只能有入度，每个必经节点的出度和入度都必须为1，其它节点的出度和入度相等，满足流量平衡原理。以下我们先写出各个模型公共部分，再分别解释每个模型所特有约束及其背后的思想。

带必经节点最短路问题的数学规划模型公共部分如下：

subject to

模型目标是最小化所求路径上的各边权重之和。

约束(2-5)是模型中对每个节点度的约束，约束([2](#eq:2))保证路径起点的出度为1入度为0，约束(3)保证路径终点的出度为0入度为1，约束(4)限制起点终点以外节点的入度和出度相同，约束(5)保证任一必经节点都在路径上。如果模型只具有上述约束，模型可能会包含一些圈，即子回路，并不是合法解，正确的模型需要添加额外约束来避免或消除子回路的出现，不同思想的约束将产生不同的数学规划模型。

2.5.1 CM模型

第一个模型是借鉴经典的TSP问题消除子回路方法的模型(Conventional Model,)，添加约束([6](#eq:6))得到完整模型，保证对于节点的任意划分子集，该子集内不会有回路出现，需要注意的是，约束(6)的数量规模是指数级别的，*CM*总共有约束，个决策变量，相比*CM*模型，其他几个数学规划模型的约束数量都是节点个数的多项式量级，属于紧凑型模型

*CM*模型具有指数个数的约束，为了得到更为紧凑的模型，基于顺序和网络流等思想的模型被先后提出。

2.5.1 SM模型

第二个模型是顺序模型(Sequential Model, *SM*)，定义实型决策变量，代表节点在路径中被访问的顺序，所以的取值范围不会超出节点个数，约束([7](#eq:7))主要限制节点和的先后顺序；假设边(i,j)出现在一条路径中，则节点先于节点被访问，满足关系式，此时如果路径中存在子回路，继续向后遍历的时候又会回到节点，则出现节点先于节点()的情形，违反上述约束；假设边(i,j)没有出现在路径中，则节点和节点被访问顺序的差值不会超出，约束(7)自动变为冗余。模型总共有约束，个决策变量， 个实型决策变量

2.5.3 FM模型

第三个模型是基于网络流的模型(Flow Based Model,)，定义决策变量为边上的流量,约束(8)限制了网络中的边只有在最终路径上时才有流量流经，约束(9)限制所有的流量都从起点流出，且流出流量总和为个单位，约束(10)保证每经过路径上的一个点，不管是否为必经节点，流量都减少个单位，因此最终找到解时，如果不是所有节点都被经过，则剩余流量不会为0。

2.5.4 Q3模型

第四个模型是Andrade[2013,2016]提出的基于对偶性原理的模型，引入对偶变量，代表路径中当前节点距离起点的距离，模型约束大为减少，总共包含个约束，实际上约束([11](#eq:11))是冗余的，只需要约束([12](#eq:12))就能得到正确的模型，即如果边出现在路径中，则保证节点和的先后顺序即可

2.5.5 Q4模型

第五个模型是Andrade[2016]提出的，也是一种基于网络流的模型，思想来源于提出的Steiner TSP问题中的模型，但是模型从源点流出的流量只在必经节点和终点t处存在消耗，其他节点保持流量守恒，所以总共从源点流出个单位的流量，在流经非必经节点的时候流量不会减1，只在必经节点处消耗一个单位流量，最终找到解时剩余流量为0。

## 2.6 本章小结

本章基于

# 第3章 基于多策略多阶段优化的启发式算法

## 3.1 算法主框架

带必经节点最短路问题主要的难点在于确定必经节点顺序，在确定必经节点的顺序之后还需要考虑相邻必经节点之间的路径选择，必须保证选择尽可能短的路径且不能有重复访问的节点，最终才能得到一条最优的路径；但是这两者是相互耦合的，调整必经节点的顺序，可以使得路径中权重之和更优，但是在调整必经节点的顺序的时候有可能会破坏简单路径的约束。 通过简要的文献综述，我们发现带必经节点最短路问题在学术界研究内容非常少，但与其相关的类似的问题如TSP问题及其变种问题在学术界得到广泛的研究。Dreyfus[1969,]指出对于没有简单路径约束的必经点最短路问题，可以转化为TSP问题求解，必经节点通过最短路径相连，最终求解TSP问题得到的结果就是原始问题的解。我们在深入探究了该问题的结构之后，对于有简单路径约束的必经节点最短路也选择转化为TSP问题的方案来求解，以利用TSP问题的丰富研究成果。假如网络拓扑图中除起点和终点外其他所有的节点都必须经过时，则该问题退化的TSP问题；反过来，也可以认为是必经节点和起点终点即维度的TSP问题，必经节点之间的距离使用最短路算法求得，首先松弛节点不能被重复访问的约束，在求解出一条路径时再进行调整，我们称路径中的节点重复的节点个数为冲突数目。



我们通过一个实例来说明求解SPPMPN问题的启发式算法的整个过程，图(1)表示原始问题的有向带权图*G*，图*G*节点集合包含11个节点，边集*E*包含24条边，必经节点集合，起点*s*=0，终点*t*=10。

|  |  |
| --- | --- |
| a | 算法 1：主求解流程 |
|  | 1:  2:  3: |

我们的算法基于推论：给定必经节点的顺序时，各必经节点之间使用最短距离的路径更优。因此我们选择相邻必经节点之间的路径时会尽可能的选择更短路径。

## 3.2 和TSP问题相互转化

3.2.1 转为TSP问题

在转化为TSP问题后得到图，节点集合为，边集={(i,j)|i,j∈V^'}，由节点集中节点两两相连组成，边(i,j)上的权重为节点i到节点j的不经过其他必经节点的最短距离，节点不连通时权重设置为∞，如图(1)所示，其所包含的节点为原问题中的必经节点以及起点0和终点10，由于TSP问题求解得到的是一条环路，所以转化为TSP实例后，需要将起点和终点合并为一个节点(也可以看作设置终点到起点的距离为0)，在原图*G*中所有到达终点10的路径在新图中都连到起点0，如图(2)中边(9,0)实际代表原始图中(9,10)的路径长度和所经过节点，新图中节点之间的距离为原图中对应的必经节点之间的最短距离，我们使用*Dijkstra*最短路算法求解原图*G*中任意两个必经点之间的最短距离，注意路径不能经过其它必经节点，否则会破坏简单路径约束



上图示例转换为TSP问题后的图形如图所示，0代表原图*G*中起点和终点的合并节点，(0,4)有向边代表原图中从0节点到4节点的最短距离和最短路径，(4,0)有向边代表原图中从节点4到终点10的最短距离和最短路径。图()中有向边上的数据代表节点之间的最短距离和途经节点(第一行表示节点值小的节点到节点值大的节点的距离和路径，第二行为反向边)，红色标注的路径表示原图中到达终点*t*的路径，在转化后的图中连接到起点*s*。

|  |  |
| --- | --- |
| a | 算法 1：主求解流程 |
|  | 1:  2:  3: |

3.2.2 从TSP结果补充路径

在图上求解TSP问题之后我们得到一个环游，但是出现在上的边并不是图*G*对应SPPMPN问题的边，环游上的节点只是原始图中的必经节点。我们还需要将必经点之间的最短路径途经节点补充到环游中，才能得到SPPMPN问题的一个解。由于我们事先在程序GenerateTSPGraph中已经计算好图*G*上必经点集合中各点的最短距离和最短路径所经过的点，分别保存在矩阵Cost和Path中；因此我们只需要在Path中递归搜寻必经节点的前驱节点直到子段路径连通即可。



上述示例中我们得到环游T={0,4,5,9,0}，0为SPPMPN问题的起点，我们从0开始补充路径，我们首先填充0到4之间的最短路径节点，在图G中可以得到一段子路径{0,2,3,4}，其次填充节点4到5之间的最短路径节点，在图G中得到子路径{4,1,2,5}，同理得到另外的(5,9)和(9,0)两段子路径，当第二次遇到起点0时，我们要将起点替换为终点，如(9,0)，我们将0替换为终点10，填充节点9到到终点的最短路径途经节点，综上，我们最终可得一个当前解(0,2,3,4,1,2,5,7,8,9,6,7,10)。

|  |  |
| --- | --- |
|  | 算法 1：主求解流程 |
|  | 1:  2:  3: |

## 3.3 求解TSP问题

3.3.1 第一阶段

直接对转化后的图求解TSP问题时分为两个阶段进行，第一个阶段不考虑目标路径中冲突节点的个数，单纯求解TSP问题，等效于先松弛简单路径的约束，快速的确定出必经节点的顺序。如果必经节点序列通过最短路径连接时没有冲突节点出现，则算法得到了最好的解，退出。如果通过最短路连接的节点序列存在冲突节点，则还需要进一步的优化。

3.3.2 第二阶段

此时第一阶段通过求解TSP问题所得必经节点序列通过最短路径相连时存在冲突节点，即有非必经节点被多次访问。此时出现冲突主要的原因可能是必经点的顺序不合理。为了得到正确的必经节点的顺序，我们还需要进一步k-opt邻域搜索。此时不再松弛简单路的约束，在寻找k-opt的动作时除了考虑要降低路径的权重之和外，还需要考虑减少冲突节点的个数，即第二阶段的局部搜索程序目标有两个，分别是优化路径的权重和路径中冲突节点的个数，并且以降低路径中的冲突节点个数为主。该阶段求解TSP问题成功之后，如果通过最短路连接的路径中没有冲突节点个数，则算法得到了所能搜索到的最好的解，程序退出；如果在结果中还存在冲突节点，则表明必经节点之间通过最短路相连的前提下是无法通过交换必经节点的顺序来彻底消除所有冲突节点，我们还需要搜索候选路策略来解决问题。

|  |  |
| --- | --- |
|  | 算法 1：主求解流程 |
|  | 1:  2:  3: |

在我们搜索到可以减少路径权重的领域动作时，首先要判断该动作是否会减少整条路径中的冲突节点个数，我们这一阶段的目标是通过k-opt优化必经节点的顺序，使得路径中的冲突节点个数最小。



对于上述示例，通过求解TSP问题最终得到必经节点的序列T={0,4,5,9,0}

## 3.4 候选路搜索

由于求解TSP问题过程中，必经节点之间的路径都是通过最短路相连，在k-opt局部搜索过程中可能无法通过交换必经节点的顺序得到SPPMPN问题合法的解，即对于任何必经节点的序列，通过最短路相连时路径中总有被重复经过的节点，冲突节点集合C≠∅。候选路搜索策略主要消除求解TSP问题得到的路径中存在的环路，通过搜索次短路或者k短路[yen’s]来避开冲突节点，得到一个合法的解。

候选路搜索理论上最好是要搜索k短路，但是我们算法中实际只搜索次短路，我们基于假设：如果一个冲突节点无法通过必经节点之间的次短路避开，说明这个冲突节点的拓扑位置比较重要，可以通过属性提升策略处理；另外k短路会使得算法复杂并且时间复杂度变高，k短路算法随k值增大对于SPPMPN问题解的优度降低，还有原因就是当前必经节点的顺序不一定是最优的，如果不是最优的必经序列，也没必要搜索多条k短路。通过对一些算例的最优解分析，必经节点之间基本上都是通过最短路和次短路相连。

SPPMPN问题的目标是寻找一条经过所有必经节点的最短路，因此在搜索候选路的过程中我们需要尽可能的选择权重小的路径。假设冲突节点集合，对于每一个冲突节点，都存在相关联的四个必经节点，因为冲突节点是两段必经节点之间的最短路径相交引起的。假设我们在上一阶段得到SPPMPN问题的路径如下所示，为了方便表示，删除路径中除了冲突节点和与其关联的四个必经节点（可能会出现重复，如）之外的其他节点。对于冲突节点的处理，我们采用以下方法：

(m\_1,m\_2)继续以最短路相连，(m\_3,m\_4)避开冲突节点搜索最短路

(m\_1,m\_2) 避开冲突节点搜索最短路，(m\_3,m\_4)继续以最短路相连

(m\_1,m\_2) 避开冲突节点搜索最短路，(m\_3,m\_4)避开冲突节点搜索最短路

我们分为以上三类处理的主要原因是为了尽可能的利用已有的最短路，保证问题的解更优。如果尝试以上三步可以避开冲突节点，我们选取可行且最终路径权重最小的一种方式。如果三种方式均无法避开冲突节点，我们将记录下来，进入下一阶段的处理。如果我们在处理时，其候选路径和候选路上的节点发生冲突，我们将冲突节点记录下来进入下一阶段的处理。

如在示例图中我们通过上述过程可以求得路径，冲突节点集合，必经节点的顺序已经确定，我们搜索相邻必经节点之间的次短路来消除冲突节点。此处只以冲突节点2为例说明，（此处），单独在(0,4)和(4,5)之间搜次短路都无法避开冲突节点2，但(0,4)和(4,5)同时避开冲突节点2时可以找到合法路径，我们得到子段次短路径，冲突节点7处理类似，最终得到完整的路径。



搜索候选路的时间复杂度的分析，主要来自最短路算法。对于每个冲突节点，需要找四次最短路径进行比较选取。使用优化的*Dijkstra*算法，假设有*m*个冲突节点，理论上最坏时间复杂度为。

|  |  |
| --- | --- |
| a | 算法 1：主求解流程 |
|  | 1:  2:  3: |

## 3.5 节点属性提升

算法执行到该阶段表明问题比较复杂，经过前面各阶段求解仍然无法得到一条合法的路径，冲突节点集合。无法求得合法路径的原因有两种，首先是必经节点的顺序不合理，必经节点之间的路径选取总是无法避开冲突节点；其次是必经节点之间的路径选择不合理，并且在候选路搜索过程中，冲突必经点对之间候选路搜索的顺序不正确，存在先得到的次短路占用了后面将要搜索的次短路待经过节点的可能，导致无法避开冲突节点。无论实际是哪一种原因造成，我们都可以认为当前问题实例结构特殊，并且这些无法避开的冲突节点在图G中具有非常重要位置，多对必经点之间的最短路径或次短路径都要经过某些冲突节点，并且一旦将这些冲突节点从原图G中去除，则无法得到次短路。基于这些冲突节点的位置非常重要，我们假设这些点一定会出现在最优解里面，所以我们对这些冲突节点的属性进行提升，使这些冲突的非必经节点变成必经节点，即在原始图G扩充了必经节点的个数，然后重新开始整个求解流程。

节点属性提升策略主要用于确保算法可以尽可能的得到合法解；我们假设SPPMPN实例都存在合法解，求解TSP问题的算法足够强大和高效(目前可以求解的TSP问题最大规模达到200000)。假设对于所有的非必经节点全部提升属性使其为必经节点，则SPPMPN问题完全变成TSP问题，只要对于TSP问题能求得一个合法解，则SPPMPN问题也就得到了合法解。所以节点属性提升策略是通过忽视解的最优性来保证能够得到合法解，是提升算法健壮性的关键策略。如果待提升属性的冲突节点选取有误，即提升不会出现在最优解中的节点为必经节点，则只能得到近似最优的解。求解TSP问题和搜索候选路等阶段可以有效的排除这类错误节点，属性提升阶段仍然冲突的节点基本上在图G是非常特殊的节点；反过来，属性提升策略可以选出具有特殊性质的节点，指导和完善具体业务的实施。

|  |  |
| --- | --- |
| a | 算法 1：主求解流程 |
|  | 1:  2:  3: |

在上述示例中，如果删除节点0到节点1之间的边，则求解TSP问题和搜索候选路策略是无法避开冲突节点2，此时在多阶段启发式算法框架下只有对节点2进行属性提升之后才能求得SPPMPN问题的最优解。

## 3.6 本章小结

本章首先介绍了基于多策略优化的多阶段启发式算法主框架，接下来在各小节中详细描述了算法的各个流程。

# 第4章 实验结果分析

## 4.1 算例集生成介绍

带必经节点最短路问题在学术界的研究成果有限，没有公开的标准数据集。在已发表的文献中，学者们是通过随机生成网络结构和随机选取必经点集合来构造测试算例；或者是在TSPLIB、SNDlib等算例集上选取小规模算例，然后随机选取必经节点集合来产生算例集，算例节点个数以及必经节点的比例都比较小。从可以查阅到的文献中我们了解到，目前文献中测试过的规模最大的算例节点数达到500。

我们通过以下几种方式生成算例，算例类型有满足三角不等式的实际网络结构以及不满足三角不等式的虚拟网络结构，我们保证生成的算例有解。经过多次测试筛选出一部分规模不同的算例集，用于测试SPPMPN问题的数学规划模型和我们提出的启发式算法，度量算法性能。

4.1.1 随机生成

随机选择算例拓扑结构，随机选取节点对添加边，边的权值也是随机选择（可能存在重边），必经节点随机选取，算例名称以字母r开头。

4.1.2 构造结构

构造具有特殊拓扑结构的算例，比如方格图、带状图等，还有类似整体稀疏但局部是密集图的算例。算例名称以字母c开头。

4.1.3 工业算例转化

选取TSPLIB中的EUC\_2D类型的算例，节点的坐标由TSPLIB算例中给出，我们随机选取节点对添加边，边的权值为两个节点的欧式距离，必经节点随机选取。算例名称以字母t开头。

## 4.2 数学模型结果比较

我们使用数学规划求解器GUROBI 7.5.1(Academic License)在Windows Server 2012 R2 Standard(Core 2 Intel(R) Xeon(R) CPU E5-2609 v2 (2.50GHZ,32G RAM))上运行求解程序。

CM模型由于约束([eq:6])数量规模是指数级别，不能在求解开始前添加全部约束，而是采用惰性添加约束的机制，当求解的过程中出现子圈时，再设置当前路径为不合法解。

Q3模型的M选取对于模型求解性能影响较为敏感，在第一篇文献中作者选取M值为1000，我们参考文献，选取M为((|V|-1)\*MaxWeight)()，|V|表示网络拓扑图中节点个数，MaxWeight值是图中边的最大权值。

我们将五个模型的在算例集上的求解结果都列举出来，每个模型都具有不同的特性，对比求解结果我们可以分析各个模型的优缺点，同时也可以推导算例的结构特性。

模型求解结果如下图所示

从算例的求解结果来看，CM模型和Q3模型总体求解效果更好，在大多数算例上求解结果领先其他模型，虽然CM模型理论上具有指数级别的约束，但是大多情况下界会比较紧，相比其他几个数学规划模型求解速度更快；从测试结果来看Q3模型求解速度仅次于CM模型，但是Q3对于M值的选取非常敏感，M选取不恰当会严重影响模型求解速度，Grid2000算例M值升高一个数量级，求解时间呈指数上升；而基于顺序的模型SM以及基于网络流的模型FM和Q4的求解速度在大多数算例上都比较缓慢。

在算例equality中所有边的权重都相等，此时模型CM则有指数级别子回路需要消除，但是限界函数所起的实际作用非常小，求解速度会非常慢，测试结果也证实了理论推导，对于这个算例，基于网络流的模型Q4和FM模型的求解速度非常快。

通过模型求解的结果来对比和验证我们启发式算法的求解效率。

## 4.3 启发式算法结果比较

启发式算法使用C++语言实现，k-opt主要参考LKH2算法中的k-opt实现，由于在降低冲突节点个数的k-opt过程中，邻域解优度（即路径中冲突节点的个数）判断程序非常耗时，我们只是实现sequential 5-opt，没有考虑non-sequential的情形。

启发式算法求解结果显示如下：

通过算例测试结果可以看出多阶段启发式算法具有非常好的求解性能，对于大多数算例，在算法的前两个阶段就可以求解得到最优解；对于一些复杂的、结构特殊的算例，通过候选路搜索和冲突节点属性提升阶段可以求解得到一条在时效和优度俱佳的路径。该启发式算法相比模型方法，具有非常大的时间优势，特别对于大规模的算例，无论其是满足三角不等式的实际网络拓扑算例，还是不满足三角不等式的虚拟网络拓扑图，都取得非常优秀的求解结果。

## 4.4 本章小结

本章首先介绍了算例的种类和生成方法。然后对于生成的算例使用各数学规划模型求解器进行求解，对比结果分析各模型的优劣；同时根据分析结果也可以推导算例的结构构造。最后运用启发式算法求解各算例，将其结果与数学模型求解的结果进行对比，表征启发式算法的性能和效率。

# 第5章 算法分析与讨论

我们将必经节点最短路问题转换为TSP问题，利用高效的启发式算法k-opt算法进行求解TSP问题，对于求解结果在原网络图中进行寻路，如果存在冲突时寻找多条候选路调整，当这些策略都不能奏效时，采用节点属性提升策略，将发生冲突的节点提升为必经节点，保证对于有解的算例，该算法一定能求解得出合法解；通过测试分析，不同的数学规划模型在不同结构的算例上求解性能差别巨大；对于具有特殊结构的算例或来源于工业界的大规模算例，数学规划模型的求解时间非常长，不能满足实际应用需求；文中提出的启发式算法在随机生成的算例上表现出了不错的结果，对于模型求解缓慢的一些算例，在较短的时间内都能求解得到最优解或接近最优解，在时间和精度方面得到折中的方案。后续将对邻域快速评估策略进行深入研究，降低算法运行时间，提高效率。

## 5.1 增量评估技术

## 5.2 多邻域组合

## 5.3 本章小结

本章结合前面几章的内容，给出来了泄漏率理论预测模型，并通过与试验数据比较验证了模型的可靠性。同时，根据该模型分析了泄漏率随粗糙度和密封比压的变化关系，讨论了粗糙峰角对泄漏率的影响。

# 第6章 总结与展望

## 6.1 工作总结

## 6.2 不足与展望

致谢

时间的脚步总是那么匆忙，

# 参考文献

M Stedman, Keven Lindesy. Limits of surface measurement by stylus instruments[J]. Proc SPIE, 1988, 1009, 56-61.

附录作者攻读硕士学位期间发表的论文