



同濟大學
TONGJI UNIVERSITY

硕士学位论文

空调输配系统（风）自动化设计研究

姓 名：陈志森

学 号：1930254

所在院系：机械与能源工程学院

学科门类：工学

学科专业：供热、供燃气、通风及空调工程

指导教师：许鹏 教授

二〇二二年五月

同济大学



同濟大學
TONGJI UNIVERSITY

A dissertation submitted to

Tongji University in conformity with the requirements for

the degree of Master of Engineering

**Research on Automatic Design of HVAC
Transmission and Distribution (air) System**

Candidate: Chen Zhisen

Student Number: 1930254

School/Department: School of Mechanical Engineering

Discipline: Civil Engineering

Major: Heating, Ventilation and Air Conditioning

Supervisor: Prof. Peng Xu

May, 2022

同济大学

空调输配系统(网)自动化设计研究

陈志森

同济大学

同济大学

摘要

传统的空调系统风管设计主要依靠经验,受项目时间限制一般无法进行精确设计,同时为了应对上游设计的变更,暖通工程师通常需要花费大量时间修改图纸,简单的重复性劳动消耗了大量的人力并且容易产生设计或绘图错误。追求速度的设计习惯通常无法兼顾设计方案的效益,当前的设计方式耗费了大量劳动力却容易产生不合理的设计结果,造成了巨大的能源浪费和糟糕的用户体验,而实现风系统的设计自动化可以有效解决上述问题。风管设计自动化的主要研究难点是实现风口节点和管道的自动布置,并根据优化算法生成最优的设计方案。现有管道连接算法主要聚焦于水管系统,而风口节点隐含垂直方向的开口,因而只有特定的连接方式能符合施工条件,现有的管道连接算法无法生成合理的风口连接方案;同时现有的最优化算法也无法在优化过程中计算水力特性。

针对上述算法需求,本文提出了一种风管自动布置方法,该方法主要包括风管布置区域提取、风口连接和快速水力计算三个部分。首先,为了确定风管干管拓扑的求解域,本文假定风管干管布置于走廊中心线上,并根据走廊边界切割方法提出了走廊中心线提取算法;为了确定风管支管的求解域,本文通过风口选型和布置的限制条件将风口设计和校核过程进行结合,实现了风口的自动布置与选型,确定了风口连接算法的目标节点。其次,为了快速筛选出符合实际工程的风口连接方案,本文基于风管连接的规则提出了风口连接算法,该算法以字符串表示风口的连接关系,并通过字符串的计算筛选风口连接方案。最后,针对风口连接拓扑图,本文采用深度优先搜索算法建立了有向图的水力计算模型,并明确了节点和边的属性及其计算公式,该模型可以通过综合优化目标实现风管连接方案的水力稳定性、管道耗材和输配阻力等参数的综合评价,筛选出最优的设计方案。

为了验证算法的有效性,本文测试了六个不同案例,结果表明:风口连接算法可以筛除超过 99.9 %的不合理方案,且运行时长在 0.1s 以内;当风口连接方案的个数少于 5000 时,快速水力计算过程的运行时长在 5min 以内,满足工程需求。基于风口连接算法和快速水力计算模型,在明确管道布置区域后,本文以房间代表节点来连接风管的干管和支管,同时提出了局部阻力构件的位置调整算法,并基于 Python 平台实现了四栋建筑标准层的空调系统风管自动设计的全过程。

关键词: 风管自动设计;管路连接算法;快速水力计算;路径寻优算法;

同济大学

ABSTRACT

Traditional HVAC duct design mainly relies on engineers' experience and generally cannot be designed accurately due to project time constraints. Also, HVAC engineers must repeatedly change the duct design in response to changes in upstream architecture design. Duct drawing modifications are repetitive, simple, and error prone, and can be fully automated by the program. The design habit of pursuing speed usually fails to guarantee the efficiency of duct network, and the current design approach consumes a lot of manpower but with unreasonable design results, resulting in huge energy waste and poor user experience. Automated duct design can reduce the design workload and ensure optimal duct connections. The main research challenge in the automation of duct design is to generate the connections of the outlet nodes and to select the optimal solution by traversal algorithm. Most previous studies have focused on the water system, mainly on algorithms for generating pipe routing. Whereas, duct nodes are open nodes, implying a dimension in the vertical direction, thus only a specific connection can meet the construction conditions. The algorithms appropriate for pipes can produce unreasonable connections for duct routing. Also, the existing algorithms cannot calculate the hydraulic characteristics during the optimization process.

In response to the above algorithm requirements, this paper proposes an automatic duct design method, which mainly includes three parts: duct arrangement area extraction, air outlet connection and fast pressure calculation. Firstly, to determine the solution domain of the main duct topology, this paper assumes that the main duct is arranged on the corridor centreline and proposes a corridor centreline extraction algorithm according to the corridor boundary cutting method. For the determination of the solution domain of the branch duct, the design and calibration processes are combined to automatically determine the outlet size and arrangement with some constraints, as well as to determine the target node of the outlet connection algorithm. Secondly, for quickly filtering the outlet nodes connection schemes that accord with the actual project, this paper proposes a rule-based traversal algorithm, which represents the connection relationship of air outlets by strings and filters the connection scheme by the calculation of strings. Finally, the presented fast resistance calculation model

based on the depth-first search algorithm is applied to perform fast calculations in directed graph. The properties of nodes and edges and their calculation formulas are clarified. This model simultaneously computes the pressure characteristics, duct material consumption, and operational resistance in the resistance calculation process.

The methodology is tested on six different building layouts for the automatic design of ducts in a room and the results showed that: the rule-base traversal algorithm is effective in filtering connection solutions and can remove greater than 99.9% of the schemes according to the duct connection, and the running time is within 0.1s; when the number of schemes selected by the rule-based traversal algorithm is less than 5000, the optimisation running time is within 5 min, which meets the engineering requirements. Based on the rule-based traversal algorithm and fast pressure calculation model, after specifying the duct arrangement area, this paper connects the main ducts and branch ducts using room representative nodes, and proposes the position adjustment algorithm of local resistance components. The whole process of automatic duct design of HVAC system for standard floors of four buildings is realized based on Python platform.

Keywords: Automated design; Duct routing; Fast resistance calculation; Traversal algorithm;

目录

第 1 章 引言.....	1
1.1 研究背景及意义.....	1
1.1.1 建筑节能需求.....	1
1.1.2 空调系统辅助设计现状.....	2
1.1.3 研究意义.....	5
1.2 空调输配系统自动化研究现状.....	6
1.2.1 管路布置算法.....	6
1.2.2 水力计算.....	9
1.2.3 国内外研究总结.....	12
1.3 研究内容及技术路线.....	13
1.3.1 研究内容.....	13
1.3.2 技术路线和文章架构.....	13
第 2 章 风管布置前处理.....	15
2.1 概述.....	15
2.2 风管干管布置区域提取算法.....	15
2.3 风口自动布置与选型.....	19
2.4 本章小结.....	22
第 3 章 基于规则的风口连接算法.....	23
3.1 概述.....	23
3.2 模型建立.....	23
3.2.1 风口连接问题简化.....	24
3.2.2 风口连接规则.....	25
3.3 风口连接算法.....	28
3.3.1 风口预处理.....	29
3.3.2 输入参数.....	29
3.3.3 风口行连接方案计算.....	30
3.3.4 节点位置调整及障碍规避.....	36
3.4 本章小结.....	38
第 4 章 快速水力计算模型.....	39
4.1 概述.....	39

4.2	风管选型算法.....	39
4.3	有向图的水力计算.....	41
4.3.1	算法输入和相关计算公式.....	41
4.3.2	计算过程.....	42
4.4	本章小结.....	46
第 5 章	算法测试与评估.....	47
5.1	概述.....	47
5.2	优化目标.....	47
5.3	输入参数设置.....	48
5.4	房间内风管连接案例.....	49
5.5	算法结果评估.....	53
5.6	本章小结.....	54
第 6 章	空调系统风管生成案例.....	55
6.1	概述.....	55
6.2	房间内风管连接.....	55
6.2.1	孤立房间的代表节点生成.....	55
6.2.2	房间合并.....	58
6.3	空调系统干管连接.....	59
6.3.1	走廊房间的代表节点生成.....	60
6.3.2	风管干管布置.....	61
6.3.3	局部阻力构件位置调整.....	63
6.4	案例应用.....	65
6.5	本章小结.....	68
第 7 章	结论与展望.....	69
7.1	结论与成果总结.....	69
7.2	课题创新点.....	70
7.3	局限性与展望.....	70
	参考文献.....	71
	致谢.....	77
	个人简历、在读期间发表的学术论文与研究成果.....	79

第 1 章 引言

1.1 研究背景及意义

1.1.1 建筑节能需求

世界能源消耗的快速增加已经引起了人们广泛的关注，供能困难、能源枯竭和环境破坏（臭氧层破坏、全球变暖、气候变化等）等问题日益严重^[1]，国际能源署（International Energy Agency, IEA）公布的数据表明，建筑用能占全球能源消耗的 40%^[2]。美国能源信息署（U.S. Energy Information Administration, U.S. EIA）估计，随着工业化和城市化进程的加快，建筑能耗将以相应的速度继续增加^[3]。

如果能在建筑设计阶段准确预测建筑能耗，则可以通过优化能耗过大的设计达到节能效果。建筑能耗受诸多因素影响，如环境气候条件、建筑结构和热力特性、照明和暖通空调系统的运行、人行为等。多因素的综合影响使得准确预测建筑能耗变得非常困难^[4]。如果在建筑设计初期将上述影响因素的参数与其他能耗模拟工具进行共享，建筑能耗预测的复杂性和难度将大大降低。研究表明，建筑能源模型的搭建时间占整个能耗模拟所需时间的一半，如果将建筑设计阶段的所有参数直接与其他模拟平台进行搭接共享，建筑能耗模拟的效率将得到显著提升^[5]。美国国家标准与技术研究所（National Institute of Standards and Technology, NIST）估计，每年至少浪费 150 亿美元用于处理建筑设计软件的关联操作问题^[6]。因此，在建筑设计过程中保证软件之间的协同操作和数据共享是非常必要的，许多企业和政府都针对该领域进行深入研究^[7]。

建筑中通常会设计空调系统，尤其对于夏季炎热地区，保持室内空气质量至关重要^[8]。建筑用能中大部分（54%）为供热和制冷用能^[2]，更高效的暖通空调系统可以极大地提高能源利用效率。但是，由于空调系统的设计需求过大导致设计时间非常有限，同时由于节能技术的进步以及可再生能源应用的增加，暖通空调系统的复杂性提高，设计师缺乏足够的时间了解和掌握新技术。设计人员和研究人员通常采用基于近似值和一般性假设的设计方法，导致设计结果不符合实际要求，造成能源浪费。

1.1.2 空调系统辅助设计现状

传统的空调系统设计过程主要包括负荷计算，设备选型，管道设计等，如图 1.1 所示。在实际工程中，空调设计周期短，任务重，且需要与建筑等上游设计进行配合，需要不断更改设计结果。设计师通常采用经验设计，容易导致设计结果不合理，造成空调系统能耗过大。为了减少空调系统设计不合理造成的能源浪费和缩短空调系统的设计时间，许多团队研究和开发了对应的设计方法和软件工具，使设计变得高效。一般来说，HVAC 系统设计工具主要由空调制造商开发，作为建筑设计软件的插件或独立组件^[9]。

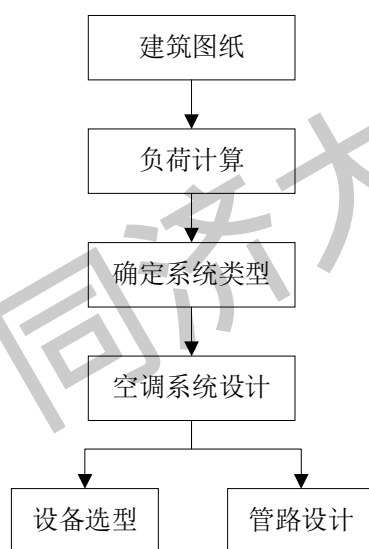


图 1.1 空调系统设计流程

空调系统负荷计算的准确性直接影响设计结果的合理性，是空调系统优化设计过程中的关键一步。Verbai Z 等学者^[10]从设计参数取值的角度研究降低空调系统能耗的方法。他们利用 50 年的室外干球温度数据建立累积频率图，提出了 99% 和 99.5% 置信度的新设计值并将其应用到空调系统设计过程中，取得了非常好的节能效果。ASHARE (American Society of Heating, Refrigerating and Air-Conditioning Engineers) 发布了五种建筑冷负荷计算方法^[11]，包括总等效温差/时间平均法 (TETD/TA)、传递函数法 (TFM)、冷负荷温差/太阳能冷负荷/冷却负荷系数法 (CLTD/SCL/CLF)、热平衡法 (HBM) 和辐射时间序列法 (RTSM)。Mao C 等学者^[12-15]对上述五种冷负荷计算方法的准确性进行了研究，结果表明，HBM 是最准确的方法，其次是 RTSM、TFM、TETD/TA 方法和 CLTD/SCL/CLF

方法。由于实际设计过程中,采用不同负荷计算方法产生的结果差异非常大,为了安全起见,设计师通常以计算结果的最大值作为设计值,导致系统过大。为探究不同负荷计算方法的差异性,Nadeem T B 等人^[16]研究了采用 CLTD 或者 HAP 软件(基于 TFM 方法开发的软件)计算空调系统冷负荷的结果差异,并分析了产生偏差的原因:HAP 软件中的缺省值与实际数值不符。由于 CLTD 方法中的所有设计数值都是根据 ASHRAE 标准精确选择的,因此得到的结果更加真实可靠。基于 CLTD 方法,Nadeem 团队进一步开发了一个空调冷负荷计算软件。随着计算机技术的进步,为了更好地辅助暖通工程师进行空调系统设计,负荷计算软件逐渐更新迭代并开始快速发展。据统计,目前全世界建筑负荷计算软件超过一百种,如美国 BLAST、DOE-2、EnergyPlus,英国 ESP-r,中国 DeST 等。DOE-2 是开发最早应用也最广泛的计算软件之一,并作为计算核心衍生了一系列模拟软件,如 eQuest, VisualDOE, EnergyPro 等;EnergyPlus 是美国能源部支持开发的新一代建筑负荷计算软件,以此为核心开发的软件有 DesignBuilder 等;DeST 是以 AutoCAD 为图形界面的建筑负荷计算软件。朱丹丹^[17]、周欣^[18]等学者从计算结构、主要设备模型及控制策略几方面对 DOE-2、EnergyPlus 和 DeST 三种主要软件进行了对比和分析,结果表明,三种软件的负荷计算结果差异不仅受软件本身的影响,更加取决于使用者对软件操作的熟练度和输入参数。

传统空调系统的设备选型过程是标准化的,主要由 ASHRAE 制定。标准方法是通过典型设计日参数和安全系数来进行设备选择,该方法容易导致系统过大,使初投资过大和系统常年偏离最佳工况运行。为了优化空调系统冷机的选型过程,Kang Y Z 等人^[19]提出了一种基于蒙特卡洛和动态模拟的冷机选型方法。在考虑照明强度、用电设备负荷和人员密度等不确定参数的影响下,兼顾冷机的全周期成本和室内的热舒适度,综合确定出冷机的最佳设计容量。Lee W L 等人^[20]通过对香港 50 栋商业建筑的空调系统设计方案和 186 台冷机的实际运行数据进行分析,确定冷机容量与系统能源性能无关,进而建立了冷机数量与能源效率的相关模型,该模型可以快速帮助设计师权衡能源效率和其他因素(机房空间、经济投入等)的重要性,在设计过程中对空调系统的配置进行优化。为了优化地源热泵与热激活系统相结合的建筑的空调系统设计,Picard D 等学者^[21]开发了一个 Python 工具。该工具在给定建筑参数、空调负荷以及空调设备库的情况下,采用基于标准、制造商数据和基本假设的静态计算,可以遍历计算所有设计方案需要的初投资以及在对方案下的二氧化碳总排放量,通过对比选择最优的空调系统设计结果。为了探究不确定性因素影响下的空调系统设计方法,Huang P 等人^[22]基于多标准决策方法建立了一种新的空调系统设计模型。该模型在设计中考虑了

建筑物理参数、设计气象参数以及空调负荷波动的不确定性，并且能够在设计阶段从多个性能指标和客户的要求和偏好方面评估设计的性能。Sun Y M 等人^[23]采用不确定性分析方法对不同来源的不确定性因素进行综合量化，并用量化指标代替传统设计过程中的安全系数，同时提出了一种基于概率的敏感性分析方法，用于识别导致不确定性的重要个体因素或群体因素。

然而，暖通空调系统设计必须基于建筑设计，是一项复杂的综合任务，需要来自不同学科的设计师协同设计。Kennington J 和 Monaghan P E^[24]指出，这种复杂性是导致建筑行业在提高能源效率方面进展甚微的主要原因之一。研究人员认为，通过使用基于综合设计的系统设计方法，可以节约 70%左右的能源^[24-26]，该综合设计方法的目的是找到各种能耗影响因素在所有约束条件下的最优解。但现有的多数空调系统设计工具都缺乏必要的连续性和整体性，不能用于综合设计^[27]。目前，已有研究人员尝试通过开发集成设计工具来解决上述问题^[28-31]。集成设计工具的开发思路主要分为两类：第一类是开发专门用于整合 HVAC 系统和建筑热模拟的软件；第二类是通过中央数据库在设计软件之间传输数据。第一类集成设计工具的开发研究目前主要基于建筑信息模型（BIM，Building Information Modeling）平台。在建筑的设计阶段使用 BIM 有助于设计师发现设计问题并协同多个学科之间进行设计^[32]。为了实现建筑和 HVAC 系统的综合设计，Choi M 等学者^[33]提出了一个变制冷剂流量空调系统（VRF）的设计模块，实现了在 BIM 环境中的空调系统设备选型和管道布置，同时可以分析设计结果的节能效率。Sha H J 和 Wang H X 等人^[34,35]提出了一个基于 BIM 的暖通空调系统半自动设计的概念性框架，欲以人工智能技术取代目前基于人工的空调系统设计。这个框架包括以下自动化过程：建筑信息模型（BIM）简化、建筑能源模型（BEM）生成、负荷计算、设备选型、空调输配系统设计和系统图生成，如图 1.2 所示。

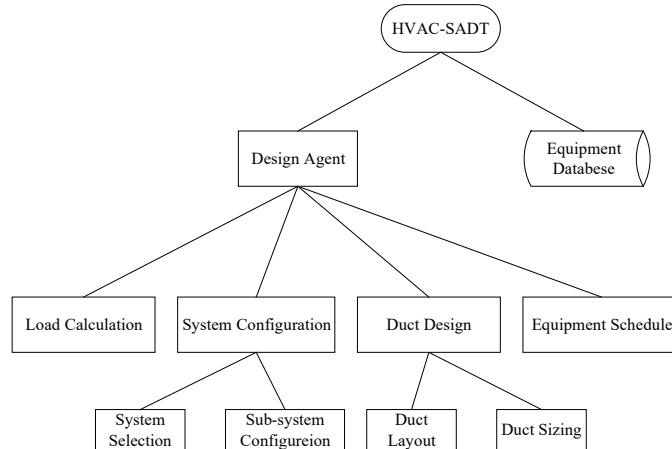


图 1.2 暖通空调半自动设计工具的构成^[34]

1.1.3 研究意义

空调系统设计的自动化是当前行业发展的前沿,也是暖通行业发展的必然趋势。多年以来,许多学者和专家在该领域进行研究,取得了一些重要的成果,但目前依然没有真正实现空调系统的自动化设计过程。输配系统作为空调系统设计中的重要部分,关乎整个系统的节能性以及用户的舒适性,但目前关于风系统自动化设计的研究依然缺乏,其中涉及多方面原因,故本课题拟对空调输配系统(风)设计的自动化展开研究,其研究意义如下:

(1) 顺应空调系统设计自动化发展的必然性

随着计算机科学技术的急剧发展,人类生活进入人工智能时代。理论上,一切可重复性强,创造性低的工作都可由机器执行,从而大大减少人的工作量。在传统的建筑行业中,存在着大量的重复劳动,过去,由于各个专业工作的相关性,专业之间的协调过程需要反复进行,设计人员花费大量时间绘制图纸,并对图纸进行反复修改,Alfred W.在1902年为纽约证券交易所设计第一个空调系统时,用时一年才完成图纸。在手工绘图年代,大量的人力支出用于制图,用于实际创新性劳动的时间不多。随着部分辅助设计软件的研发,与其他行业的协同问题得到有效解决,但设计人员依然需要重复进行图纸绘制和修改过程。一项针对设计院工程师的调研显示,目前,在部分工作由计算机辅助完成的情况下,实现一栋建筑面积为1万平方米的住宅建筑暖通空调系统设计工作需要投入至少10(人·天),实现一栋建筑面积为5万平方米的办公建筑暖通空调系统设计工作需要投入至少40(人·天),实现一栋建筑面积为10万平方米的综合体建筑暖通空调系统设计工作需要投入至少135(人·天)。由此可见,计算机辅助绘图技术在暖通空调设计领域的广泛应用对于设计效率的提升巨大,但依然还存在大量重复性劳动有待进一步实现自动化。空调系统的设计方案很大程度上由建筑的构造和建筑类型决定,工作需要的创新性不高,因此,空调系统设计过程由机器替代是历史必然,开展空调系统设计自动化的相关研究具有历史前瞻性。

(2) 解决空调风系统的现有设计问题

空调系统设计过程中,设计人员操作存在许多不合理步骤,为了赶工程进度,空调系统的管道布置往往采用经验设计,并未进行方案比对,管道布置方案确定后,即便管网系统存在较大的水力失衡现象,也不会对设计方案进行优化,而采用增加平衡阀进行管网平衡调节,造成管网阻力过大,增加能耗。由设计人员手工进行设计方案的对比过程繁琐,耗时长,同时也是创新性极低的重复性劳动,因此,实现空调风系统管道布置的自动化具有重要意义。

1.2 空调输配系统自动化研究现状

1.2.1 管路布置算法

管道布置是空调输配系统设计优劣的重要评价依据,空调输配系统的设计要求是保证管道布局的短、直和简单,同时追求最佳的输配性能。有关管道布置算法的研究已经超过 50 年,最先提出的算法主要应用于障碍物影响下的管道连接情况,算法的主要思路是通过权重的设置,对障碍物进行规避,常见的算法有迷宫算法、逃逸算法、Dijkstra 算法、A*算法、粒子群算法、蚁群算法和遗传算法。

尽管生活中常见的管网都是枝状连接,在管道连接问题的研究初期,学者们普遍聚焦于两点间的管道连接。1961 年, Lee^[36]提出了迷宫算法(Maze algorithm),将计算区域划分为均匀网格并对障碍物所在网格进行标记,在无障碍物网格中进行搜索过程。所有的单元网格根据其与起点的相邻与否用不同的权重进行标记,迭代一次后,以下一个网格作为起始节点重复搜索过程,直到到达目标所在网格。迷宫算法可以保证有解,但需要大量的运行内存^[37], William R P^[38]和 Mitsuta T 等学者^[39]将迷宫算法应用于机房水管的空间连接方案求解,以实现机房水管管道的自动设计。1969 年, Hightower^[40]提出了逃逸算法(Escape algorithm),该算法先给起点划定两条垂线边界,再试图找到一个逃逸线可以延申到起点边界外的节点,以该点为下一个搜索过程的起点重复搜索,直到逃逸线穿过终点。逃逸算法较于迷宫算法计算速度快,内存需求小,但不能保证有解^[37]。Traub H S 等学者^[41]将迷宫算法和逃逸算法应用于概念性机房自动布置过程中的水管连接,初步实现了在一定设备布局规则下的管道连接。部分学者则将管道布置问题视为节点间的最短路径问题,在节点间建立连通图并采用最短路径算法求解管道布置方案。基于 Dijkstra^[42]算法, Reginaldo G^[43]采用整数编程和线性网络优化的方法,提出了一个优化机房布局和管道连接的策略; Wangdahl G E 等人^[44]开发并演示了一种用于两个指定端点之间的管道布置的动态编程方法,为管道自动布置奠定了算法基础。Nicholson T^[45] 则提出了一种新的方法来寻找连通图中两点之间的最短路线。算法先对初始起点和终点的相邻连接边进行搜索,将产生最短边的节点(起点或终点)的所有连接边放入已遍历的边集合,并将最短边上的另一个节点更新为下一个搜索过程的起点或者终点,以更新后的起点和终点作为下一个计算步长中的初始节点进行搜索,同时将搜索过的边放入已遍历的边集合,当起点和终点联通时,对已遍历的边集合中的其他联通边的权重进行判断,确定起点和终点联

通路线是否为最短,该算法比采用线性或动态编程求解更高效。A*算法是路径规划领域中常用的一种算法,基于机器人路径规划技术,Zhu 和 Latombe^[46]提出了一个管道自动布置系统,将管道路线视为刚性物体(机器人)留下的路径,对搜索域进行网格划分后使用 A*算法求解最优的管道连接方案。作者在该自动布置系统中同时考虑了管道实际布置中的其他约束条件,包括建造工艺,施工和运行维护的便利性等。1999 年,学者 Ito^[47]尝试用遗传算法^[48](GA, Genetic Algorithm)实现机房布局设计中的管道路线的交互式规划,以确定水管连接的最佳方案。该算法过程同样需要对求解域进行网格划分,并赋予每个网格单元一个潜力值。位于墙边的单元的潜力值最低,因为路径更有利,而障碍物单元的潜力值则相应地高。该方法通过代表管道路线的基因的进化先产生一个管路布置方案,设计者评估后对其进行修改,进行另一次 GA 优化,重复该过程直到设计出合适的路线。

1976 年,Newell^[49]在 Nicholson 的算法基础上,率先开始了分支管道布置算法的研究。随后, Park 和 Storch^[50]沿用网格划分计算域的方法,将枝状管网视为包含末端分叉和中间分叉的管道结合体,提出了一种船舶内枝状管网自动布置的方法。该方法主要实现多条管道的综合布置,分支连接则需要人工参与才能实现,计算前需要输入管线信息,以区分主管和分支管线,同时需要人工确定分支节点位置。樊江等人^[51]用网格划分三维空间,利用改进的迷宫算法解决了单点对多点的分支管路连接问题,实现了航空发动机的管网自动敷设。文中基于迷宫算法先求得其中两个点之间的最短路径,将这条最短路径上所有网格点作为源点,下一个端点作为目标点再进行最短路径计算,生成多个点之间的最短路径,如此反复,直到连接上所有的目标点为止,管道敷设次序遵循先粗后细。虽然迷宫算法的运算速率较低,但是航空发动机内空间较小,需要的计算量较少,文中通过实践验证敷设一条管路只需要 100ms (SGI O2 的工作站上),能满足工程需求。

随着算法的发展,求解一对多节点的连接算法也逐渐更新迭代,除了前面述及的确定性算法外,分支管道路由问题还广泛使用启发式算法来解决,主要包括粒子群优化算法^[52](PSO, Particle Swarm Optimization)、蚁群算法^[53](ACO, Ant Colony Optimization)和遗传算法。PSO 算法将鸟群的个体位置或食物当作优化问题的解,利用群体中个体与最优个体以及个体之间的信息交互,引导整个群体中个体在保留自身多样性信息的同时,朝向群体最优个体收敛,通过不断地更新逐渐找到最优解^[54]。2006 年,Asmara 和 Nienhuis^[55]将 PSO 和 Dijkstra 算法^[42]结合起来,解决分支管道的连接问题。PSO 算法用来优化管道的连接顺序,每个粒子对应一种排序方式;Dijkstra 算法则用于确定每根管道的最短连接方式。该方法可以解决多个系统管道的布置问题,但随着管路数量增加,计算时间增加明显,

有 n 条管道则需要求解 $n!$ 次。枝状管网的连接问题也可以转化成多个节点间的最小斯坦纳树^[56]问题，即在目标节点外再添加其他节点（斯坦纳点）使连接结果最优。管道连接的斯坦纳节点通常位于 Hanan^[57]提出的直角斯坦纳树网格节点上。Liu 和 Wang^[58,59]使用 PSO 算法寻找多节点的最小斯坦纳树，实现了航空发动机表面上的管道自动布置。该方法以斯坦纳节点的个数和坐标作为粒子，每个粒子对应一种斯坦纳节点选择方式，之后采用最小生成树算法求解粒子对应的最短路径，逐渐更新粒子以求解最优的斯坦纳节点组合方式并生成目标节点的最小斯坦纳树。蚁群算法的研究模型源于对真实蚂蚁觅食行为的模拟。蚂蚁在觅食过程中，会在所经路径上释放出一种可挥发的信息素，不同蚂蚁个体通过感知信息素的存在及其强度指导自己的移动方向。蚂蚁个体间通过感知信息素交换路径信息，最终整个蚁群在这种自组织作用下搜索出巢穴与食物源之间的最优路径^[60]。为了获得了单个枝状管网的最小斯坦纳树结果，Hu 等人^[61]和 Luyet 等人^[62]使用蚁群优化方法对目标节点的 Hanan 网格进行搜索，在每个目标节点处生成一只蚂蚁进行路径搜索，当两只蚂蚁相遇时，一只蚂蚁死去，将两只蚂蚁走过的路径进行合并，作为另一只蚂蚁走过的路，并将蚂蚁的位置调整到走过的路径上与其他蚂蚁距离最近的节点，继续搜索直到路径中的蚂蚁数为 1。随后对生成的路径进行修剪使其符合管路输配规则，并更改路径中每条边的信息素浓度值，重新生成蚁群进行搜索，直到找到全局最优解。邬君^[63]、Jiang W^[64,65]、Wu L^[66]和 Liu Q^[67]等学者则将分支管路系统视为多个独立管路的结合，将每个独立分支视为一个种群，采用蚁群算法进行独立支管的优化，再采用多种群协同进化算法对总体管路布局进行优化，实现了不同场合下三维空间的枝状管网连接。遗传算法基于生物体进化规律，将自变量编码成个体，在求解域中随机产生多个个体组成种群，在每一次迭代中，根据适应度函数评估种群中的所有成员，然后从当前种群中用概率方法选取适应度最高的个体产生新一代种群。在这些选中的个体中，一部分保持原样直接进入下一代种群，其他则应用遗传方法（交叉或变异）产生新个体，最终迭代进化出最优种群^[68]。遗传算法应用于分支管网计算时，通常将一个分支管路分割成若干起点相同而终点不同的单管，对每个单管使用遗传算法搜索最优路径后，合并所有单管的连接方案，再综合筛选生成最优的枝状管网布置方案。单管连接的遗传算法将搜索域中的一个网格坐标对应一个基因，路径所经过的网格坐标组合则对应一条染色体。杂交是指在两个父本中随机选择两个点（不包括起点和目标点）作为新的起点和目标点，重新生成一个连续的中间路径，每个父代被随机选择的点切割成两部分，将父本前段和中间路径以及另一父本后段连接生成

杂交子代。变异是指从一条父染色体中随机选择两个点作为起点和目标点，随机产生一个新的中间路径取代父染色体中两个选定点之间的部分，生成变异子代。基于上述流程，Ren T^[69]、Sui H T^[70]、Niu W T^[71]和 Dong Z R^[72]等学者在船舶管道设计中应用改进的遗传算法，在各自提出的单管布置顺序算法下，实现了障碍空间中的枝状管网自动布置。

在空调系统设计领域，相关的管道连接算法研究较少。为了在建筑设计前期能对建筑性能进行全面模拟，Brahme R 等人^[73]基于管道设计规则建立启发式算法，以实现建筑空调区域的自动划分和区域内部干管的自动布置，区域内部端点的连接以及区域间的管道连接则采用 Dijkstra 最短路算法计算。Medjdoub B 等人^[74]采用枚举法计算建筑机房内的管道连接方案，并设计了交互式设计程序。Brès A 等人^[75]在确定管道的节点后，通过添加可能连接的中间节点构建连通图，通过最小生成树算法对所有连接结果进行遍历，计算最优的管道布置方案。Medjdoub B 等人^[76]主要研究了风管系统的自动布置问题，他们使用“分支和约束”^[77]算法实现了风机盘管系统新风管道的半自动设计，设计过程需要依靠用户经验来选择部分连接方案。Stanczak M 等人^[78]介绍了两种在三维空间中自动设计管道系统的方法，以实现各种约束下的管道连接。该方法将管网系统视为直管段和弯管的结合体，并建立了所有可能使用到的正交和非正交弯管列表，通过遍历列表生成所有连接方案，再分别通过线性程序和混合整数线性程序进行最优方案计算。

1.2.2 水力计算

水力计算是空调风系统管道设计过程中的重要环节，目的是经济合理地确定风管尺寸以保证实际风量与设计要求相符，同时计算管网的总阻力以用于风机选型。传统的风管水力计算方法有假定流速法、压损平均法和静压复得法^[79]，单纯依靠人工计算时，通常采用计算表形式逐段地确定管道尺寸并计算其对应的阻力值，此过程需要手动输入大量设计参数，工作量大且容易出错。随着计算机技术的发展，市面上逐渐出现水力计算的辅助工具，以实现管网系统的自动水力计算。国内广泛应用的水力计算软件主要有中国建筑科学研究院建筑工程软件研究所研发的 PKPM 系列软件中的空调水力计算软件 HCVAV^[80]、北京天正软件股份有限公司的天正暖通软件^[81]和广联达科技股份有限公司（北京鸿业）的 BIMSpace 机电模块^[82]，上述风管水力计算软件都需要用户预先布置设备和管道并设置水力计算方法，计算结果则包含风管尺寸以及阀门、三通和弯头等管道附件的阻力计算结果。国外的水力计算软件由于工程设计的差异而在国内鲜有应用，但其水力计算

流程和方法与国内软件基本相同。

基于商业目的，水力计算软件的算法并不透明，但算法涉及的基础核心数据是各个部件的阻力系数，阻力系数的正确与否决定了水力计算结果是否可靠。古往今来，已有大量学者开展关于风系统构件阻力系数的研究，风管阻力根据特性可分为沿程阻力和局部阻力两类。直管段的沿程阻力通常采用 Darcy Weisbach 公式 (1.1) 进行计算，其中沿程阻力系数依据雷诺数的大小选择对应的经验公式进行计算，如 Colebrook 公式 (1.2)、Moody 公式 (1.3)、Churchill 公式 (1.4) 等。

$$\Delta P_f = \lambda \frac{l}{d} \left(\frac{1}{2} \rho V^2 \right) \quad (1.1)$$

$$\frac{1}{\sqrt{f}} = -2 \log \left(\frac{\varepsilon}{3.7d} + \frac{2.51}{Re \sqrt{f}} \right) \quad (1.2)$$

$$\lambda = 0.0055 \left[1 + \left(20000 \frac{K}{d} + \frac{10^6}{Re} \right)^{\frac{1}{3}} \right] \quad (1.3)$$

$$\lambda = 8 \left[\left(\frac{8}{Re} \right)^{12} + \frac{1}{(A+B)^{1.5}} \right]^{\frac{1}{12}} \quad (1.4)$$

$$\text{其中, } A = \left[-2.457 \times \ln \left(\frac{7}{Re} \right)^{0.9} + \frac{0.27 \varepsilon}{d} \right]^{16}, \quad B = \frac{37530^{16}}{Re}$$

风管的局部阻力系数主要采用设计手册中的数值，常用的设计手册主要有前苏联学者 Idelchik 编著的《Handbook of Hydraulic Resistance》^[83]、美国 ASHRAE 手册^[11]、英国 CIBSE 指南^[84]以及我国学者编著的《全国通用通风管道配件图表》^[85]和《实用供热空调设计手册》^[86]，上述设计手册中的局部阻力系数大多通过实验的方法测定。上世纪六十年代，Idelchik 首先开展有关弯管流动特性的研究并确定了影响阻力的因素。随后，D.S.Miller^[87]实验测定了弯管和三通在不同的弯曲角度、横截面积、曲率半径和雷诺数下的阻力系数。国内方面，周谟仁等学者^[88]研究了弯管的流态并分析了局部阻力的产生机理。秦慧敏等人^[89]分析了三通局部阻力系数后，提出了影响三通局部阻力系数的主要因素，认为在分流三通中，支路的局部阻力系数主要受该支路与主管的流量比和面积比影响，与另一支路的参数无关；同时发现以流速比为自变量拟合分流三通的局部阻力系数的效果优于采用面积比或流量比。自上世纪以来，随着计算流体力学（CFD, Computational Fluid Dynamics）的发展以及计算效率的提高，数值仿真方法逐渐被应用于风管构件的局部阻力系数研究中。为了测试 CFD 仿真是否可以替代实验确定风管构

件的局部阻力系数, ASHRAE 发起过一次国际学术竞赛 (RP141493), 吸引 CFD 领域的研究者在未知实测数据的情况下采用 CFD 仿真计算案例的局阻系数, 但所有参赛学者都无法将每个测试案例的预测误差控制在 15% 以内。因此, 实际工程中, 水力计算软件中的风管局部阻力系数的参考值主要是设计手册中的实验值。

设计手册中的局部阻力系数通常是孤立阻力部件的试验测试值, ASHRAE 120 标准^[11]在测定局部构件上下游压差时, 规定上下游的测压截面与构件之间需有一段较长的直管段。然而, 实际风系统中局部构件之间的连接管大多不满足上述规定, 经过上游构件的气流通常无法恢复到均匀的充分发展流便进入下一个构件。这就使得两个局部构件近距离相连时, 二者的总局部阻力并不等于两个孤立构件的局部阻力之和。因此, 国内外的研究者开展了构件相邻连接时总局部阻力特性的研究, 此类研究主要通过实验和模拟相结合方式, 得到了部分构件结合情况下的阻力系数如表 1.1 所示。尽管构件相邻连接会对风管管网的阻力特性产生影响, 但相关的研究依然不够全面和准确。现有研究只涉及了少量的局部阻力构件结合情况, 且在相关研究中并未给出具体的工程参考数值。因此, 在实际设计过程中, 现有的水力计算软件暂未考虑相邻局部阻力构件对管网阻力的影响。

表 1.1 有关相邻构件对局部阻力系数影响的研究汇总^[90]

研究者	构件类型	连接形式	与孤立 (或间距较远) 加和值的偏差
Mumma 等人 ^[91] (1995)	渐扩管、弯管	特殊连接	0 ~ +27%
Atkin 和 Shao ^[92] (2000)	弯管	U 型连接 S 型连接	-5.4% ~ +8.1%
Sami 和 Cui ^[93] (2004)	弯管	U 型连接 S 型连接 T 型连接 Z 型连接	-35% ~ 0
Mylaram 和 Idem ^[91] (2005)	弯管	S 型连接	-4.8% ~ +37.4%
Lakshmiraju 和 Cui ^[94] (2006)	弯管	U 型连接 S 型连接 T 型连接 Z 型连接	-18% ~ 0
Ai 和 Mak ^[95] (2013)	类风阀构件	直线串联	-36.8% ~ -2.0%
Sleiti 等人 ^[96-98] (2017)	弯管	U 型连接 Z 型连接	-22.67% ~ +22.2%

1.2.3 国内外研究总结

现有的管路连接算法研究主要以水管为研究对象，目前已有多种算法能实现平面或空间中的管道避障和连接，但现有算法都基于求解域的网格划分方法，故要求建筑结构较为规整，同时网格大小的设定强依赖于经验，因而算法的鲁棒性不高。虽然水管的连接方式与风管类似，但水管连接算法会产生不合理的风管连接结果。风管连接算法中的目标节点是风口，由于风管是开式系统，风口节点隐含着垂直方向上的出口，如图 1.3 所示，同一个拓扑连接在两种管网系统中表示不同部件，在水管连接中表示弯头，但在风管连接中则产生空间三通，此部件在实际工程中并不存在。其次，由于空气和水的特性不同，四通普遍存在于风管系统中，但由于导致水力失衡而基本不应用于水管系统，因此水管连接算法通常也无法生成四通连接。

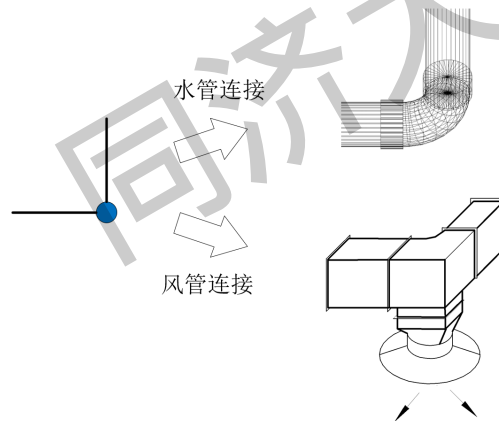


图 1.3 相同拓扑连接在风管和水管中的差异

目前有关空调风系统的管线布置算法只能初步生成风口和管道的连接关系，优化布置方案过程则以管长最短和弯头个数最少为目标，计算过程中结合障碍物信息可以进行管线障碍规避，但暂时依然没有在生成管线布置拓扑时考虑水力稳定性和管道选型的自动设计算法。对于空调风系统而言，设计方案更重要的是保证水力稳定性，水力失调将导致温度失控影响用户体验，同时也造成能源浪费。

综上所述，现有的风管连接研究无法满足工程需求，而已被深入研究的水管连接算法无法用于风管设计。同时，现有的管道连接算法求解最优方案时需要固定的节点和边的权值，但节点之间的水力特性会随着方案的不同而变化，采用现有的算法思路无法在优化过程中考虑管网的水力稳定性。因此，一个综合考虑多种因素的空调输配系统（风）自动设计算法依然有待深入研究。

1.3 研究内容及技术路线

1.3.1 研究内容

本课题研究对象为空调输配系统中的风管系统。研究目标为：针对空调风管系统设计，综合考虑影响风系统设计的各种因素，提出适用于送风、回风以及新风系统的自动设计算法。为了实现该目标，本文将研究问题细化为：

- a. 如何在满足实际工程需求的条件下实现风口节点的自动连接？
- b. 如何在风管自动布置中综合考虑造价、运行费用和水力稳定性等因素？
- c. 如何对管道自动布置算法进行求解速度优化？

1.3.2 技术路线和文章架构

根据研究内容，针对具体的研究问题，本课题需要进行以下工作：

a. 确定风管管网拓扑求解域：空调系统的风管管网在相同楼层中布置高度变化不大，因此在建筑平面图中完成管路布置后再进行高度调节可以简化计算，本文主要研究二维平面图中的风管自动布置方法。建筑平面图属于二维图，而风管连接方案可视为节点连接的拓扑图，平面中有无穷多种拓扑连接方案，无法直接在平面区域求解特定的拓扑图，因此需要先确定风管管网拓扑的求解域。本文将风管连接分为走廊中的干管和房间内的支管两部分，干管拓扑的求解域设定为走廊中心线，建筑中走廊的形状多种多样，因此需要研究一种通用的走廊中心线提取方法；房间内的支管布置区域则主要由风口位置决定，因此，实现空调风系统的自动化设计需要先研究风口自动布置算法。当风口位置确定后，需要确定支管拓扑的求解域，本文结合工程设计习惯在风口连接算法中确定支管的求解区域。

b. 风口连接算法研究：当明确管网布置拓扑的求解域后，需要结合风管设计习惯进行风口连接算法研究。由于风管系统的水力特性在所有节点连接后才能确定，因此本文需要研究一个生成所有符合工程的风口连接方案的风口连接算法。

c. 快速水力计算模型研究：确定风口连接拓扑后，需要进行风管系统的水力计算。由于本文生成的拓扑图缺乏适用的水力计算方法，同时生成的拓扑图数量巨大，为了减少计算时间，本课题需要研究一种快速水力计算模型。

d. 综合水力稳定、造价和运行费用等多种因素：快速水力计算过程可以确定所有影响因素的取值，进而筛选最优连接方案。因此需要研究一种设计方案的综合评价方法，该评价方法可以根据不同设计目标而自动调节各个因素的权重。

基于上述研究工作的分析，本文的技术路线如图 1.4 所示。

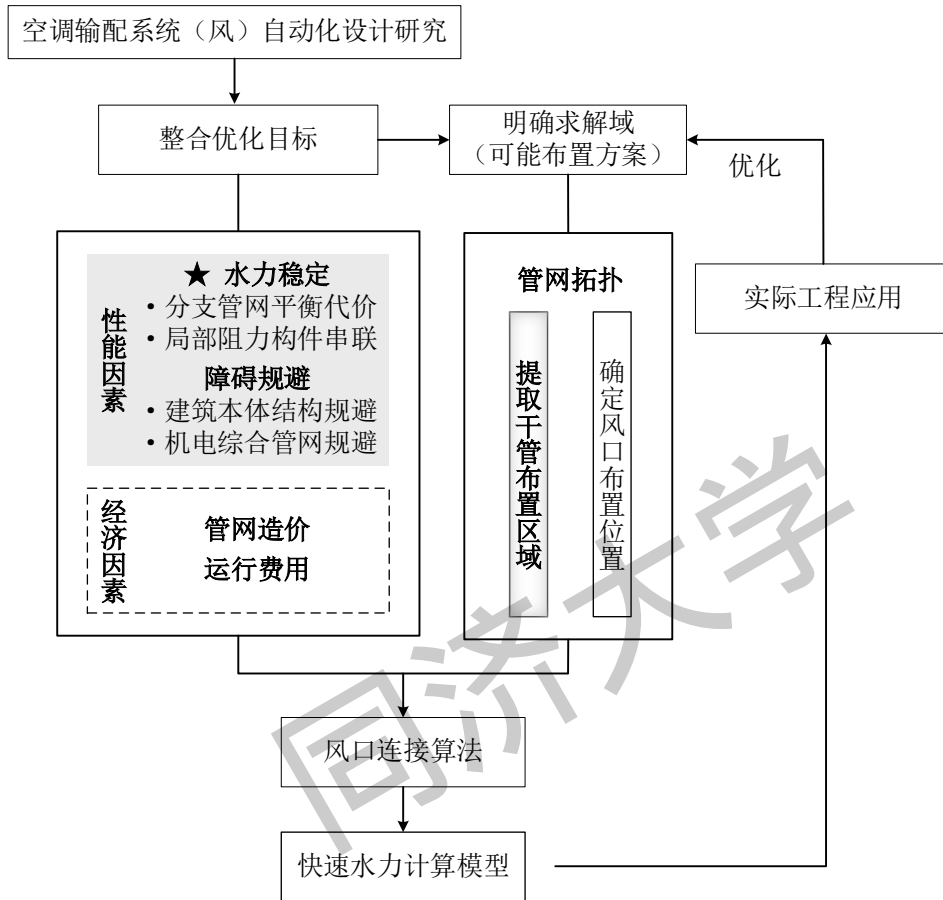


图 1.4 技术路线

本文第二章介绍了风管布置的前处理部分，包括对干管布置区域的提取以及风口的自动布置，属于技术路线中的明确求解域部分；第三章详细介绍了风口连接算法的建立过程；第四章提出了用于风口连接有向拓扑图的快速水力计算模型；第五章是风口连接算法和快速水力计算模型的测试与评估部分，在该章节提出了优化目标的整合方法；第六章将本文提出的风管系统自动设计方法应用于实际建筑中；第七章为本文的结论与展望。

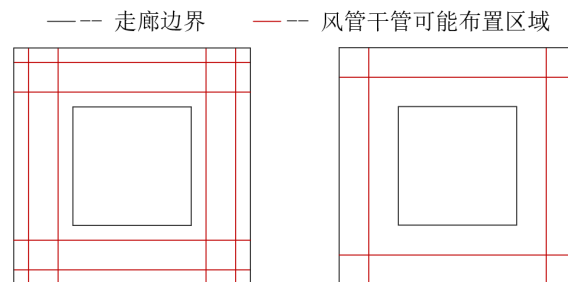
第2章 风管布置前处理

2.1 概述

传统空调系统的风管设计过程包括风口选型及定位、风管连接和水力计算等过程。人工进行风管布置可以直接进行障碍物规避，将风管布置在合理位置，但需要通过计算机程序完成布管过程，则需要先明确管道布置的区域，即确定管道连接拓扑的求解域。房间内部风管可以根据房间轮廓节点和风口节点建立连通图，并通过算法求解风管布置方案；但是房间外部的风管则通常无法确定其输配节点，因而无法建立网格来确定布置方案。本章是实现风管系统管道布置过程的前处理部分，主要介绍从建筑的走廊轮廓中提取风管干管布置区域的算法，为风管干管的布置算法提供求解域，同时在本章中提出风口的定位和选型算法，确定房间内部风管连接算法的目标节点。

2.2 风管干管布置区域提取算法

在实际工程中，为了施工和维修方便，空调系统的风管干管一般沿走廊敷设，将处理后的空气送往与走廊连通的房间内。风管干管一般沿着输配房间的墙线布置，若采用与实际工程相同的布置方法进行风管干管的布置方案计算，如图 2.1a，干管的可能布置方案随着走廊数量的增多而成倍增加，大大增加了计算难度，为了减少计算量，本文将房间外部风管的求解域限定在走廊中心线上（图 2.1b）。由于走廊的宽度一般不大，沿中心线布置和沿墙布置的管长差异很小，对干管布置方案的优劣性影响不大，在确定最优的干管连接方案后，用户可以通过实际建筑情况对干管进行适当偏移，以满足工程需求。



a.实际工程中风管干管的布置位置；b.本文规定的风管干管布置位置

图 2.1 风管干管的布置位置

通过干管布置位置的限定,本文生成的风管干管布置方案走廊中心线的关系为:风管干管布置方案 \subseteq 走廊中心线。

因此,在风管干管布置方案确定前,需要先提取走廊的中心线。走廊中心线的提取是平面图向拓扑图转化的过程,提取多边形的中心线在其他领域中通常采用直骨架算法(straight skeleton),二维平面上简单多边形的直骨架^[99],其算法逻辑如图 2.2 所示,已知多边形 $P = \{p_1, p_2, \dots, p_{n-1}, p_n\}$, 各顶点 p_i 沿着其邻边夹角的平分线向 P 内部等速移动所形成的直线轨迹构成了 P 的直骨架 $S(P)$ 。其过程会发生两类事件:边事件和分裂事件。边事件是指多个顶点在缩进的过程中会合成一点从而多个多边形的边消失成点;分裂事件是指凹顶点在缩进过程中将对边分割成两条边,从而原多边形被分裂成多个多边形。原多边形上的边定义为轮廓边,由原多边形的边与边的角平分线所组成的边称作骨架边,原多边形上的顶点为轮廓顶点,由骨架边相交而成的顶点为骨架顶点。直骨架由骨架顶点和骨架边组成,对于给定多边形是唯一的。

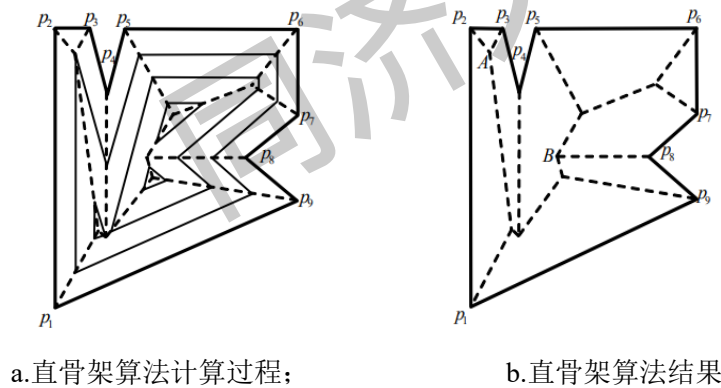
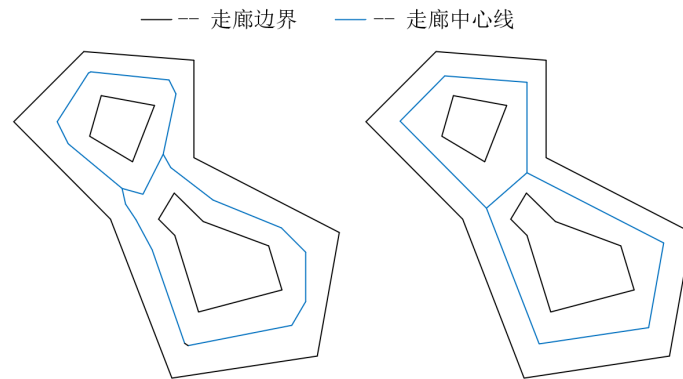


图 2.2 平面多边形的直骨架提取^[99]

直骨架算法可以较为精确地提取多边形的中心线,但该算法的计算量较大,对多边形的每条边都计算其相对的中心线,应用于走廊中心线提取时,由于其提取过于精确,可能导致某条走廊的一边有几何变化时,提取的中心线是多段折线,如图 2.3a 所示。而风管干管布置追求短、平、直,在相同多边形下,需要提取出的中心线不需要遵循严格的中心位置,但需要尽量减少弯折,如图 2.3b 所示。由于建筑设计形状的多样化,走廊几何中存在各种影响走廊中心线提取的形状,提取过程会受凹点或凸点影响,导致中心线提取算法的通用性差。直接采用直骨架算法产生的结果不太适用于风管布置,对结果进行修改则需要额外设定其它判别条件,提出通用的修改算法比较困难。因此,本文提出一个走廊中心线提取算法,以提取出满足风管布置的走廊中心线拓扑。



a.采用直骨架算法生成的中心线； b.风管敷设所需要的中心线

图 2.3 直骨架算法生成的中心线与布管所需

用于布置风管干管的走廊中心线的提取目标是：用最少的连通线段覆盖所有房间。为了保证线段数量最少，则需要保证每条中心线都尽量长，因此，本文从走廊外轮廓的最长边开始，依据边的长短对走廊多边形进行切割。本文将每次切割过程中涉及的走廊外轮廓边称为基准边，每次切割过程包括选择基准边、确定切割点和切割多边形三个环节。

切割的对象为上一次切割产生的多边形，切割的基准边是多边形中属于走廊外轮廓的最长边，根据基准边可以切割出一个覆盖走廊外轮廓且属于走廊区域的四边形。为了使中心线较好地反映走廊趋势，切割后的四边形最好可以提取出与基准边平行的中心线，故本文规定切割边与基准边平行。为了保证切割后生成的四边形在走廊区域内，需要确定一个切割点，使经过该点的切割边都在走廊区域内且尽可能长。为了使切割次数最少，切割点应处于走廊边界上。切割点需要满足通过该节点生成的切割边与基准边形成的四边形完全位于走廊区域内，符合该要求的节点具有以下特征：节点与基准边的两个端点的连线都位于走廊内部，本文依据该特征对所有节点进行筛选，当有多个节点满足要求时，选择与基准边垂直距离最近的节点可以得到最长的切割边。确定切割节点后，在切割节点处作基准边的平行线，使平行线穿过整个走廊区域，此时平行线与走廊边界会产生多个交点，在所有交点形成的边中选择位于走廊区域内且边长最长的边作为切割边，通过切割边对多边形区域进行分割，生成一个两边平行的四边形和若干个多边形，对所有非三角形进行搜索切割，直到所有走廊外边都被遍历或被切割。

切割形成多个四边形后，再生成对应的中心线拓扑，中心线拓扑必须连通且边尽可能少。优先生成最长的中心线可以尽快连通所有四边形，生成的边数最少；优先遍历已完成计算的四边形所相邻的四边形可以保证中心线的连通状态。因此，

中心线的提取过程为：从基准边最长的四边形开始，搜寻其相邻四边形，逐渐遍历所有四边形生成连通的中心线拓扑。每个四边形先以非基准边和非切割边的两条边中点连线作为走廊中心线，该中心线平行于基准边和切割边，将中心线在走廊区域内部延伸至走廊边界，将其添加到已生成的中心线中。当某个四边形生成的中心线与已生成的多条中心线不相连时（图 2.4b），需要添加其他边，使所有中心线连通。添加的边位于四边形与其相邻图形的共享边上，添加边与共用边垂直，当共用边较长时（大于 3 米），无法确定风管干管的最优连接方式，因此添加两条边，添加边与共用边端点距离需满足风管的安装需求；当共用边较短时，在共用边中点处添加一条边。对所有四边形进行中心线生成和联通后，可以得到走廊中心线的连通拓扑图（图 2.4c）。

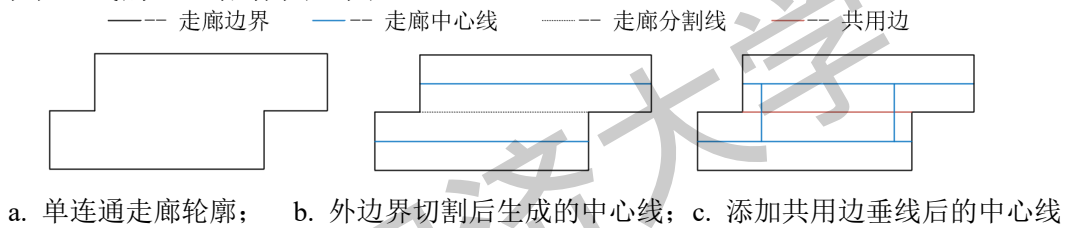


图 2.4 单连通走廊的中心线提取过程

复联通域的走廊中心线提取过程与单连通走廊相似，在走廊外轮廓遍历切割后生成的中心线连通图可以覆盖所有外部走廊，但是会存在内部走廊未被覆盖的情况（图 2.5a），因此需要再进行中心线添加。内部边界是否被已生成的中心线拓扑覆盖的判定依据为：在外边界线切割后剩余的多边形中（图 2.5b）进行内边界线可切割节点搜索，当某条内边的所有可切割节点都处于走廊边界时，该条边无法被中心线覆盖。此时，找到与该边垂直距离最近的可切割节点，作该节点与内边的两个端点的连线，连接两条线段中点，并延长至与生成的走廊中心线相交，生成完整的走廊中心线拓扑（如图 2.5c）。走廊中心线的提取过程如表 2.1 所示。

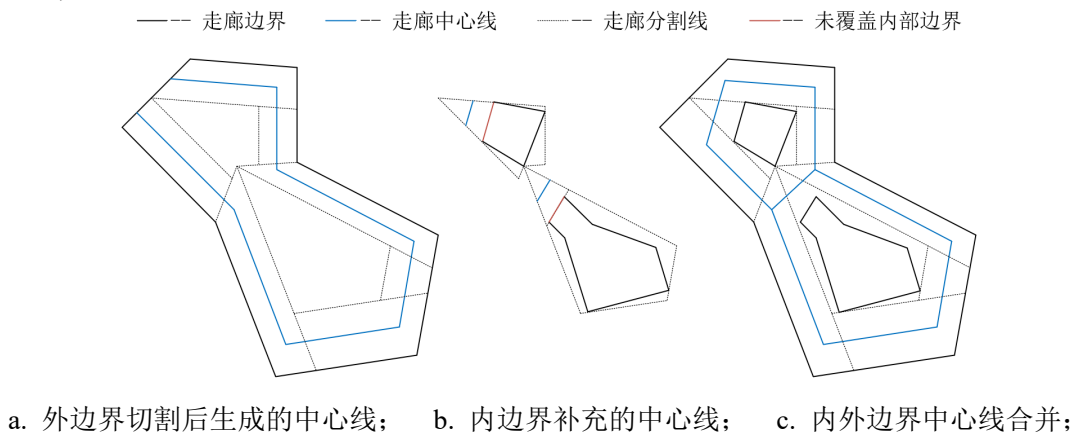


图 2.5 复联通走廊的中心线提取过程

表 2.1 走廊中心线提取算法伪代码

输入：走廊轮廓图 $G=(E,V)$ ，二维多边形
 输出：走廊中心线连通图 $G'=(E',V')$ ，拓扑图

```

1:  $G_{外} := (E_{外}, V_{外})$ , 走廊多边形的外轮廓;  $G_{内} := (E_{内}, V_{内})$ , 走廊多边形的内轮廓;
2:  $G_{四边形} := \{ \}$ , 走廊区域切割后形成的四边形的集合;  $E_{基准} :=$  切割过程的基准边;
3:  $G_{切割} :=$  切割时形成的两边平行的四边形;  $E_{切割} :=$  切割边;  $V_{切割} :=$  切割节点;
4: While  $E_{外}$  and  $\text{len}(G) > 3$  do
5:    $E_{基准} := \max(E_{外})$ 
6:   find  $V_{切割}$  in  $V$ 
7:   find  $E_{切割}$  in  $G$ 
8:   split  $G$  by  $E_{切割}$ 
9:    $G := G - G_{切割}$ 
10:   $G_{外} := G_{外} - G_{切割}$ 
11:   $G_{四边形} := G_{四边形} + G_{切割}$ 
12: end while
13:  $G' := \{ \}$ , 走廊中心线拓扑;  $G_{用} :=$  已经遍历后四边形
14:  $G_{相邻} :=$  相邻接的四边形;  $E_{平行} :=$  与切割边和基准边平行的中心线;
15:  $E_{垂直} :=$  与两个四边形的共同边垂直相交且与共同边端点距离为安装距离的连通线;
16: while  $G_{四边形}$  do
17:   if not  $G'$  then
18:     find  $G_{切割max}$  generated by  $\max(E_{基准})$ 
19:      $G' := (E_{平行} \text{ of } G_{切割max})$ 
20:      $G_{四边形} := G_{四边形} - G_{切割max}$ 
21:   else then
22:     for  $G_{相邻}$  of  $G_{用}$  do
23:        $G' := G' + (E_{平行} \text{ of } G_{相邻})$ 
24:       if  $E_{平行}$  of  $G_{相邻}$  not connect with  $G'$  then
25:          $G' := G' + (E_{垂直} \text{ of } G_{相邻})$ 
26:       end if
27:        $G_{四边形} := G_{四边形} - G_{相邻}$ 
28:        $G_{用} := G_{用} + G_{相邻}$ 
29:     end for
30:   end if
31: end while

```

2.3 风口自动布置与选型

风管干管布置区域提取后，需要进行风管支管的连接方案计算，支管的连接目标节点是房间风口，在已有的自动布置算法中，主要采用人工确定风口的布置

位置和型号。根据风口安装间距，一个区域可以产生多种符合要求的方案，设计师对每个方案再进行风速校核，从而确定一个设计方案，设计和校核过程需要人工反复进行计算和表格查询等简单重复性劳动。为减少人工参与，提高空调系统辅助设计工具的自动化程度，本文结合风口布置过程的实际要求，通过公式推导，将设计和校核过程进行整合，实现了风口的自动布置和选型过程。

本文的风口自动布置与选型算法适用于矩形房间，由于风管送风系统通常采用散流器顶送模式，因而该算法以散流器顶送方式的相关公式进行计算。根据设计手册^[86]，风口布置方案需要考虑的因素有：

- (1) 为了能够施工操作，两个散流器的安装距离不应小于设定值 D_{\min} ；
- (2) 风口负责区域的长宽比应当小于设定值 R_{\max} ；
- (3) 为了保证送风气流对房间的覆盖范围，送风射程与送风区域长度之比不应小于设定值 n_{\min} ；
- (4) 室内平均风速不能大于设定值 \bar{v}_{\max} ；
- (5) 为了防止产生过大噪音，散流器颈部的出口风速不能大于设定值 v_{\max} ；

假设某个矩形房间的长、宽和高分别为 L 、 W 和 H （单位：m），该房间需要的总送风量为 V_{supply} （单位： m^3/s ），为调整每个风口送风区域的长宽比，使其满足要求，将房间的长和宽边分别进行 i 和 j 等分， $i, j = 1, 2, 3, \dots$ ，考虑风口之间的安装间距，由限制条件（1）可得：

$$\begin{cases} \frac{L}{i} \geq D_{\min} \\ \frac{W}{j} \geq D_{\min} \end{cases} \quad (2.1)$$

通过公式变换，可得房间风口的行和列放置的风口数 i 和 j 的取值范围为：

$$1 \leq i \leq \frac{L}{D_{\min}}; 1 \leq j \leq \frac{W}{D_{\min}}; i, j \in N \quad (2.2)$$

为保证每个风口负责区域的长宽比，由限制条件（2）可得：

$$\frac{1}{R_{\max}} \leq \frac{\left(\frac{L}{i}\right)}{\left(\frac{W}{j}\right)} = \frac{L}{W} \times \frac{j}{i} \leq R_{\max} \quad (2.3)$$

$$\frac{W}{LR_{\max}} \leq \frac{j}{i} \leq \frac{WR_{\max}}{L} \quad (2.4)$$

公式 2.4 可以计算出一个房间中满足几何要求的所有风口布置方案, 每个方案对应一种房间的几何划分方式, 每种方案下的风口负责区域的几何条件为:

$$\begin{cases} V'_{supply} = \frac{V_{supply}}{i \times j} \\ L' = \frac{L}{i} \\ W' = \frac{W}{j} \end{cases} \quad (2.5)$$

其中, V'_{supply} 是一个风口的送风量, m^3/s ; L' 和 W' 分别为一个风口负责区域的长和宽, 单位为 m 。

当确定风口的个数以及布置位置后, 需要进行风口的选型和送风参数的校核。由风口选型设计手册可知^[86], P. J. 杰克曼(Jackman) 对圆形多层锥面型散流器或盘式散流器进行实验, 综合实验结果, 提出散流器平送时射流速度的衰减方程:

$$\frac{v_x}{v_s} = \frac{K\sqrt{F}}{x + x_0} \quad (2.6)$$

其中, v_x 为距离散流器中心水平距离为 x 处的最大风速, m/s ; K 为送风口常数, 多层锥面型散流器为 1.4, 平盘式散流器为 1.1; F 为散流器的有效面积, m^2 ; x_0 为自散流器中心算起到射流外观原点的距离, 对于多层锥面型为 0.07 m 。 v_s 为散流器的送风速度, m/s , 由风口流速和风量的关系, 散流器的送风速度为:

$$v_s = \frac{V'_{supply}}{F} \quad (2.7)$$

为了保证送风气流对房间的覆盖范围, 送风射程与送风区域的长的比值需要在限定范围内。气流射程是指送风射流的轴心速度下降到 0.5 m/s 时, 该点至风口的水平(或垂直)距离, 由公式 2.6, 满足限制条件 (3) 的风口参数为:

$$n = \frac{x|_{v_x=0.5}}{L_{\max}} = \frac{KV'_{supply}}{0.5L_{\max}\sqrt{F}} - \frac{x_0}{L_{\max}} \geq n_{\min} \quad (2.8)$$

其中, L_{\max} 是风口送风区域的长, m , $L_{\max} = \max(L', W')$ 。

为避免产生过强的吹风感, 需要对风口送风后的室内平均风速进行控制, 即满足限定条件 (4)。P. J. 杰克曼提出, 室内平均风速与房间尺寸和主气流射程有关, 其经验公式为:

$$\bar{v} = \frac{0.381nL_{\max}}{[L_{\max}^2 + H^2]^{0.5}} \leq \bar{v}_{\max} \quad (2.9)$$

由公式 2.8 和公式 2.9 整合变换后可得，散流器的有效面积满足以下公式：

$$\left[\frac{0.381KV'_{supply}}{0.5 \times \left(\frac{1}{4} + \frac{H^2}{L_{max}^2} \right)^{0.5} v_{max} + 0.1905x_0} \right]^2 \leq F \leq \left[\frac{KV'_{supply}}{0.5(n_{min}L_{max} + x_0)} \right]^2 \quad (2.10)$$

为了满足风口运行过程中的噪音要求，由限制条件（5）， $v_s \leq v_{max}$ 可得：

$$\frac{V'_{supply}}{v_{max}} \leq F \quad (2.11)$$

根据上述算法，由公式 2.5、公式 2.10 和公式 2.11 可直接计算符合设计要求的风口定位和风口尺寸，当有多个可选方案时，为了施工便利，采用风口个数最小的设计方案。其他类型的风口自动布置与选型过程与散流器相似，只需替换对应的射流衰减方程和房间内空气平均速度公式即可进行计算。

对于其他多边形区域，本文采用陈喆^[100]提出的最小包络矩形方法进行切割。最小包络矩形的求解算法分为凸多边形和凹多边形的情况。对于凹多边形，需要将其切割成多个凸多边形后采用最小包络矩形方法进行风口布置。沿凹点与房间投影所在的走廊中心线形成的垂线方向对凹多边形进行分割，将分割后的多个凸多边形视为多个房间，每个房间的风量按面积进行分配。对于非矩形的凸多边形，采用其最小包络矩形作为风口布置区域，完成风口自动布置与选型。

2.4 本章小结

本章为风管布置前处理部分，主要进行风管干管求解域的确定以及进行风口定位和选型。为了减少风管布置过程的计算量，本文将风管干管的布置区域限定为走廊中心线，并提出了针对单连通区域和复连通区域的走廊多边形中心线自动提取算法，该算法以走廊边界的边长为切割依据，将多边形切割成多个两边平行的四边形，再通过中心线的连通算法连接所有四边形的中心线，生成满足风管布置要求的连通的走廊中心线拓扑。确定风管干管布置区域后，本章为风口连接算法确定了连接目标，即确定了风口的布置位置以及风口型号。根据工程中风口布置需要满足的限定条件，通过公式推导与整合，得到适用于计算机执行的风口选型和定位算法，减少了表格查询过程，实现了风口的自动布置与选型。

第3章 基于规则的风口连接算法

3.1 概述

本章介绍了基于规则的风口连接算法的具体实现过程，主要包括从设计图纸中建立风口连接模型、从实际工程中提取风口连接规则、以及基于规则遍历风口连接方案的详细过程。该算法的主要目的是通过风口连接规则对风口连接拓扑进行快速筛选，去除不符合工程要求和明显非最优连接的方案，将风口最优连接方案的求解域大幅度缩小，减少计算量，使遍历计算时间缩减到工程可接受范围。

3.2 模型建立

本节介绍了风口连接模型的建立过程，包括风口连接问题的简化和风口连接规则的提取。图 3.1a 是某大空间全空气系统的风管设计图，从管道的设计结果可以将风管设计问题简化成节点的连接问题，对节点类型和位置进行分析后，可以将平面上的节点连接简化成特定节点之间的拓扑问题，从而实现风口连接问题的简化。通过风口的连接特性以及风管配件的特性，可以提炼出风口连接过程中必然遵循的规则，利用这些规则作为风口节点拓扑的筛选条件，可以大幅减少最优连接方案挑选过程的计算量。

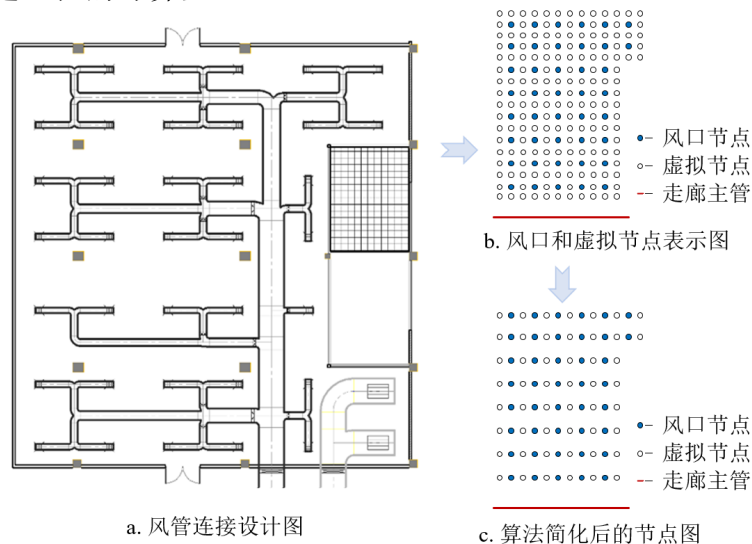


图 3.1 风管连接算法涉及的节点

3.2.1 风口连接问题简化

风口连接是指将房间内的风口连接到风管干管，其数学过程是将平面上的多个节点连接到一条线段，如果不限定管线布置的区域和产生的中间连接节点的位置，平面上连接一条线段和多个节点有无穷种可能并且无法求解。因此，本文基于风管的实际工程设计习惯，对风口连接问题进行简化，将平面问题转化成多个特定节点之间的拓扑连接问题，进而实现风口的连接。根据风管的设计图纸（图 3.1a），可以总结出风口连接方案必然遵循以下规律：

a. 向房间内输送气体的风管干管通常设立在走廊中，且房间内的风口都在干管的某一侧（上侧或者下侧）。通过简单的相对位置转换，可以将干管位置变换到所有风口的下侧，此时，房间内的空气输送可以视为从干管连接点开始，逐行风口往上进行气体输配。

b. 为了便于施工安装，风管连接相对于风管主管（或走廊）都是横平竖直连接，因此，可能的连接节点相对固定，并且均匀分布在风口周围。

基于风口的上述两个特性，风口连接过程可以视为在一个固定的点集中寻找最优的连接拓扑问题，固定的点集中包含必须被连接的风口节点（图 3.1b 中蓝色节点）和可能被连接的虚拟节点（图 3.1b 中白色节点），虚拟节点环绕在风口节点周围，每个风口节点周围都有四个虚拟节点。通过引入虚拟节点，风口连接问题从空间连接转化成二维的节点拓扑生成问题。

为了求解最优的连接方案，需要比较所有连接方案的水力特性，然而，任意两个节点之间的权重都随着连接方案的变化而变化，因此，单个方案的水力特性只能在连接所有的风口后才能确定。直接在点集中遍历并对比所有的连接方案来获取最优的设计方案在理论上是可行的，但是由于该过程需要非常大的计算量和数据内存，对于实际工程来说不太现实。假如一个房间有 $n_x \times n_y$ 个风口，产生的风管连接点集有 $(2n_x + 1)(2n_y + 1)$ 个节点，此时节点之间的连接边的数量为 $8n_x n_y + 2n_x + 2n_y$ ，如图 3.2 所示，两个节点之间存在连接和不连接两种状态，则所有节点产生的连接方案有 $2^{8n_x n_y + 2n_x + 2n_y}$ 种，当风口个数增加时，直接计算的计算量将指数型增长，因此直接对所有连接方案进行遍历不切实际。同时，许多并不满足风口连接要求的方案也被存储到运行内存中并进行方案比对，增加了计算软件和硬件的负担，也需要再建立繁琐的筛选规则。为了解决此问题，本文建立了一个去除不合理连接方案的遍历算法，在对方案进行水力计算前快速地筛选出适合实际工程的风管连接方案，减少计算量。

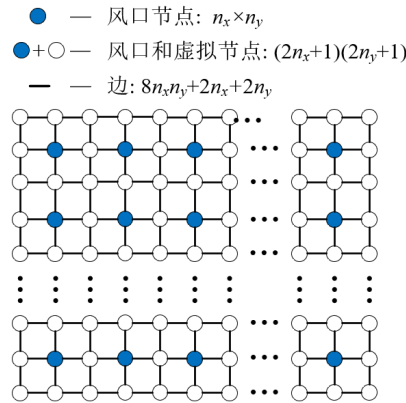


图 3.2 n_x 行 n_y 列风口节点的可能连接方式

3.2.2 风口连接规则

为了过滤掉不符合最优设计的风口连接方案，需要建立一定的连接规则。风口连接方案的筛选规则可以根据筛选目的分为两种：一种是筛选不满足风口连接的方案的规则，另一种是去除非最优连接的方案的规则。风口的连接必须遵循一定的物理规律以及满足风管的连接和安装要求，因此相关的筛选规则都是客观的，所有筛选过程都必须遵循。而非最优方案的筛选则是根据风管的优化目标来制定的，属于相对主观的人为规定，在不同优化目标下的筛选规则可能发生变化，本文基于传统的风管设计目标，将风管连接最优方案的评价指标定为水力平衡、运行阻力和管道耗材。

根据实际的风口连接要求和风管的输配特性，可以提取出风口连接过程必须遵循的连接规则如下：

- a. 风管连接拓扑中的每个节点都有且只有一个空气输入口，但可以存在多个输出口。风管设计主要目标是对人居环境进行空气输送，只有一个送风来源可以保证空气流量调节的稳定性。传统的空调系统设计通常都是此种形式。
- b. 连接拓扑中的所有支管的终点节点一定是风口节点，不可以将虚拟节点作为支管的终点。
- c. 为避免空间三通（图 1.3），风口节点不能存在两个垂直连接边（图 3.3）。

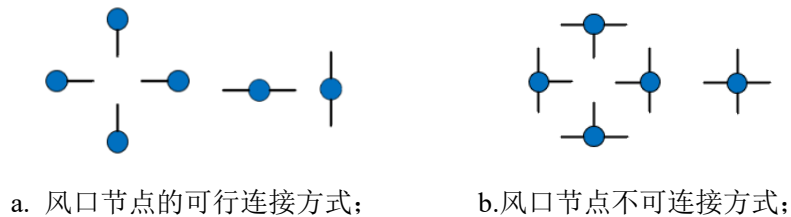


图 3.3 风口节点的可行连接与不可连接方式

为了便于解释去除非最优连接方案的规则，需要先对节点的连接过程以及本文定义的名词进行说明。由 3.2.1 节中风口连接必然遵循的规律 a 可得，通过坐标变换，风管干管位于所有风口节点的下方，风口的连接过程则变成从风管干管处逐行向上连接风口。通过图 3.1b 中每行节点的特性，可以将所有的节点以行划分为两种类型：虚拟节点行和风口行。虚拟节点行是指全部节点都是虚拟节点的一行节点；风口行是包含风口的一行节点。根据节点行类型的不同，风口的连接又可以分为虚拟行连接和风口行连接两种形式（图 3.4）。虚拟行连接是指位于单行风口之间不存在连接关系，所有风口连接到同一行虚拟节点行。风口行连接是相同行风口由该行的虚拟节点进行连接，或者不同行风口之间直接连接。

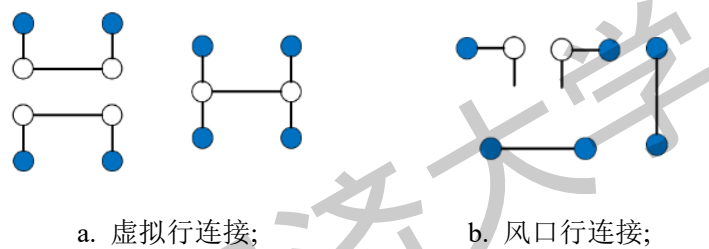


图 3.4 风口节点的两种连接方式

两行风口的连接方式也可能是混合型连接，既是部分风口按照风口行连接，一部分风口以虚拟行连接方式。如图 3.5。但是混合连接与纯粹的虚拟行连接结果进行对比，会产生更多的不平衡节点（图中红色节点），与纯风口行连接相比，需要耗费更多的管材。因此，考虑混合型连接会产生非常多的非最优方案，不利于提高计算速度。

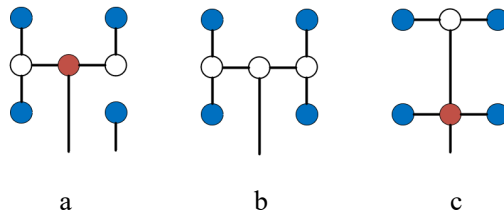


图 3.5 三种连接方式对比；a. 混合型连接；b. 虚拟行连接；c. 风口行连接；

两行风口的虚拟行连接方式有四种，如图 3.6 所示，图 3.6a 表示的是两行风口都连接到风口中间的虚拟节点行上，本文将该连接方式称为中间虚拟行连接，图 3.6b-d 可以视为两个单行风口的虚拟行连接。与其他三种连接方式相比，中间虚拟行连接明显是最优的连接方式（平衡性最好、管道耗材最少）。图 3.6c-d 中绿框所表示的是单行风口的两种虚拟行连接，两种连接的水力平衡性一样，但由

于风管干管位于风口的下方，图 3.6c 中的连接方式更加节省管材，同时输配距离更短，沿程阻力更小，因此，图 3.6c 的连接方式对于单行风口的虚拟行连接来说是最优的，本文将该连接方式命名为下虚拟行连接。

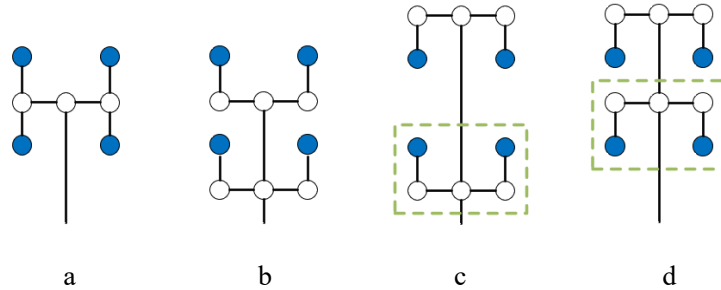
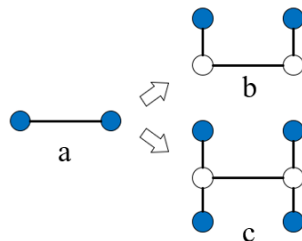


图 3.6 两行风口的虚拟行连接情况：a. 中间虚拟行连接；b-d. 两行风口的单行风口连接；

基于以上分析，本文用于筛除非最优连接方案的规则有：

- 两行风口的连接方式不考虑混合型连接，亦即两行风口的连接过程中，同一风口行的所有风口必须连接到相同的虚拟节点行上。
- 两行相邻的风口行不能同时进行单行虚拟行连接（图 3.6b-d），当两行风口都是虚拟行连接时，只能以中间虚拟行连接方式（图 3.6a）。
- 单行风口进行虚拟行连接时，只考虑下虚拟行连接方式。

在上述三个约束规则下，以虚拟行连接的风口只存在两种形式：单行风口的虚拟行连接采用下虚拟行连接方式；两行风口的虚拟行连接采用中间虚拟行连接方式，如图 3.7b-c 所示。这两种虚拟行连接都可以采用单行风口的风口行连接表示，如图 3.7a 所示。通过风口行连接表示虚拟行连接可以在遍历过程中舍去虚拟行节点，即需要进行遍历计算的节点集从图 3.1b 简化成图 3.1c，同时，在计算风口的连接拓扑时，两行风口进行虚拟行连接时，可以直接合并成单行风口，再计算风口行连接结果。节点数量的减少可以快速降低风口连接拓扑的计算量，极大地提高了计算速度。在所有节点的连接拓扑计算结束后，将表示虚拟行连接的结果进行对应还原（a 还原成 b 或者 c），便可以体现风口的虚拟行连接结果。



a. 单行风口行连接；b. 下虚拟行连接；c. 中间虚拟行连接；

图 3.7 用风口行连接表示虚拟行连接。

3.3 风口连接算法

基于 3.2.2 节中提出的风口连接规则，本节建立了风口连接拓扑的遍历算法，该算法的目标为快速搜索节点集中符合风口连接规则的方案。该算法的流程如图 3.8 所示，主要分为三个模块：(a) 风口预处理；(b) 输入参数计算；(c) 风口行连接方案计算。风口预处理阶段将节点集用字符串进行表示，以字符表示风口类型和连接状态。输入参数是指单行风口中的必须连接节点和不能连接节点位置。第三个模块是计算单行风口的所有可行的连接结果。在完成一行风口行的连接结果计算后，对下一行风口重复模块 b 和 c 中的计算，直到最后一行风口行。模块 b 和 c 在运行过程中筛除不合理的连接方案，b 生成符合两行风口行连接的输入参数，c 则将单行风口行的不合理连接进行筛除。

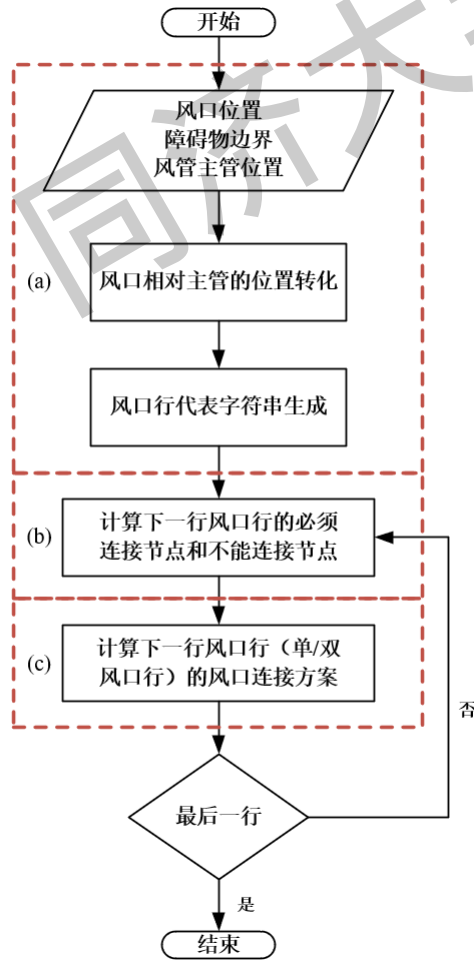


图 3.8 风口连接算法计算流程图

3.3.1 风口预处理

风口预处理是指根据风口节点的位置和风口与风管干管的相对位置进行节点的字符串表示。首先以风管干管方向建立直角坐标系，风口节点一般相对于干管（走廊）平行且均匀布置，与干管平行方向称为行，与干管垂直方向称为列。对于部分非规则房间，每行风口的个数可能不同，同时为了规避房间内部结构，也可能导致风口节点不在同一列。计算过程中节点的位置坐标还可能由于小数点偏差，导致行和列无法严格遵循数学上的平行与垂直关系。为了符合实际工程精度，本文设置了 0.25 米的节点位置偏差，当两个风口节点的水平坐标在该偏差内，则认为处于同一列，当垂直坐标处于该偏差内，则认为处于同一行。

通过以上处理，可以将风口点集分成整齐的 n_y 行 n_x 列。对该节点集在风口行位置添加虚拟节点。假如某个位置出现风口节点缺失，将该位置以虚拟节点进行填补。此时可以得到每行节点个数相同的点集，点集的形式为 $n_y \times (2n_x+1)$ 。为了区分风口和虚拟节点，本文以字符“1”和“0”分别表示风口和虚拟节点，则每个风口行可以由一串字符串表示节点间的相对位置（比如：“01010... .010”），字符串的长度为 $2n_x+1$ 。图 3.9 展示了风口缺失时的虚拟节点添加过程和节点集的字符串表示过程。

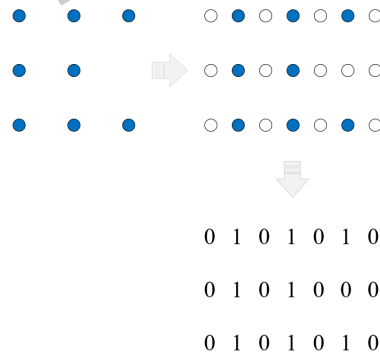


图 3.9 风口节点和虚拟节点的字符串表示

3.3.2 输入参数

当逐行计算风口行的连接方案时，上一行风口的连接情况将直接影响下一行风口的连接，具体而言，上一行风口的连接结果会产生必须接入下一行的节点，本文称为必须连接节点；上一行风口的连接结果也会产生不能连接到下一行的节点，本文称为不能连接节点。

基于 3.2.2 节建立的风口连接规则，必须连接节点是指在上一行连接中位于

连接终端的虚拟节点，虚拟节点不能作为风管连接结果中的最后一个节点，所以必须将其连接到下一行（图 3.10）。同时，用户可以规定风管的干管入口，该节点则为第一行风口节点的必须接入节点。由于风口节点不能同时出现两条垂直连接的边，才可以避免形成空间三通，则上一行连接结果中被水平连接的风口就是下一行风口行的不能连接节点（图 3.11a）。由于上一行风口连接结果中未被连接的虚拟节点没有供气接入，也无法为后面的风口行进行气体输配，因此，上一行风口中所有未被连接的虚拟节点也属于下一行风口行的不能连接节点（图 3.11b）。

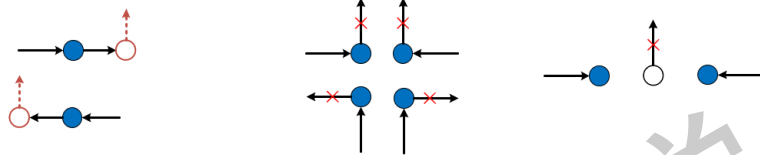


图 3.10 必须连接节点： a. 产生空间三通的风口节点； b. 未被连接的虚拟节点；

图 3.11 不能连接节点。

不能连接节点和必须连接节点作为某行风口行计算的输入参数，必须在上一行风口连接方案确定后才可以确定。对于同一风口行，上一行风口的连接结果不同，输入参数也会变化，因此只能在算法的遍历过程中计算输入参数。对于第一行风口行，当用户不规定输入节点时，其不能连接节点和必须连接节点都是空值。

3.3.3 风口行连接方案计算

在确定了风口行的不能连接节点和必须连接节点后，则可以开始进行单行风口行的连接方案计算。风口行连接方案的计算可以分为三个部分：a. 进口字符串计算；b. 进口字符串筛选；c. 风口段连接方案生成。根据必须连接节点和不能连接节点可以计算出符合两行风口的接入情况，本文同样采用字符串表示接入情况，称为进口字符串，其表示了两行风口连接的位置和连接节点的数量。符合两行风口连接规则的进口字符串不一定满足单行风口行的连接需求，因此需要对进口字符串再进行判断，筛除不符合单行风口连接规则的字符串。一个进口字符串在单行风口行中可能产生多个可行的风口连接方案，因此需要遍历计算所有合理的风口连接方案。图 3.12 展示了风口连接方案计算的流程。

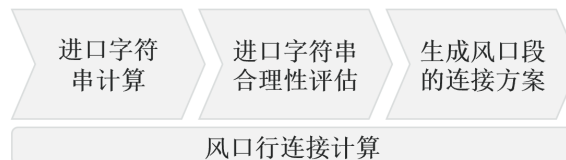


图 3.12 单行风口连接计算流程图

3.3.3.1 进口字符串计算

为了与风口行字符串保持一致，本文将每行风口的接入口也以字符串表示。与风口字符串类似，字符“1”表示所在节点与上一行风口连接，字符“0”表示没有连接。风口行的连接情况可以通过一个与风口字符串相同长度的字符串表示。

对于没有风口缺失的一行风口行，进口字符串表示的连接情况需要满足两行风口的连接规则，亦即满足不能连接节点和必须连接节点所表示的节点状态：不能连接节点对应位置的字符为“0”；必须连接节点对应位置的字符为“1”。为了保证每个节点都最多只有一个接入口，对一个含有 n 个风口的风口行，其进口字符串必须满足以下公式：

$$S_i = \begin{cases} 1, & i \in C, C \in F \\ 0, & \text{else} \end{cases}, \quad i = 1, 2, \dots, 2n+1 \quad (3.1)$$

其中， S_i 是指进口字符串中第 i 个节点对应的字符值，取值为 0 或者 1。 C 是所有连接节点的位置信息集合； F 是包含所有满足要求的连接节点位置信息的集合， C 是 F 中的一个元素，相当于其中一种合理的接入方式。当 F 为空集时，表示没有可行的连接方案。集合 F 满足下列公式：

$$F = \{X \cup M \mid X \subset P, |X| \leq n - |M|\} \quad (3.2)$$

其中， M 是必须连接节点的位置信息集合； X 是其他可能连接节点中被连接的节点位置信息集合； P 是所有可能连接节点的位置信息集合。所有可能连接节点是指除去必须连接节点和不能连接节点外的其他节点，这些节点没有明确的连接要求，因此会产生被连接和不被连接两种状态。 X 是 P 的一个子集，表示一种可能连接节点的连接情况。 P 集合可以通过公式 3.3 表示：

$$P = O \setminus (M \cup N) \quad (3.3)$$

其中， N 是不能连接节点的位置信息集合； O 是单行风口的所有节点的位置信息集合，可以表示为：

$$O = \{1, 2, \dots, 2n+1\} \quad (3.4)$$

对于下虚拟行连接和中间虚拟行连接，只需要考虑一个入口的连接情况，因此，求解符合连接规则的进口字符串的限定条件与求解风口行连接不同，满足要求的连接节点位置信息的集合 F 变为：

$$F = \{X \cup M \mid X \subset P, |X| \leq 1 - |M|\} \quad (3.5)$$

为了保持算法计算过程的一致性，对于有风口缺失的风口行也采用上述方法求解进口字符串。在求解前，先对风口行字符串进行一定的变化，使其满足进口字符串的判断规则。在本文中，对于具有风口缺失情况的风口行采用字符串替代

的方式进行连接方案求解。风口缺失位置必然会产生大于等于三个“0”字符连续排列的情况，将连续的三个以上的“0”字符以一个“0”字符替代，本文将该过程称为字符串的塌缩。塌缩后的字符串与没有风口缺失时的风口字符串形式相同，都是“0”和“1”间隔分布，可以通过公式 3.1 计算塌缩后的进口字符串。

一行风口可能在不相邻的部分具有多个风口缺失，此时利用塌缩字符串计算的进口字符串与之后的风口字符串长度不同，储存的位置信息对应不上。因此，假如单行风口塌缩后，计算完该行的进口字符串后，需要对进口字符串包含的信息再次解析，主要是分析其包含的下一行风口的必须连接点和不能连接点信息。

3.3.3.2 进口字符串筛选

对于下虚拟行连接和中间虚拟行连接，生成的进口字符串只有一个接入口，同时所有风口节点都连接到对应的虚拟行节点上，此时的风口连接结果满足所有连接规则。对于风口行连接来说，由于可能有多个接入口，并且接入口直接连接到风口行的对应节点上，满足两行风口间连接要求的进口字符串也可能导致单行风口内无法合理地连接风口。接入节点位置的分布不合理可能导致部分风口被“孤立”，没有接入口可以为其供气；也可能导致部分接入口无法连接到风口上，成为供气中断点，而此类接入口通常为上一行的某个虚拟节点，由于虚拟节点不可以作为风管的终端节点，因此，该进口字符串不满足连接规则。如：风口行字符串为“01010”，进口字符串为“11000”时，进口字符串满足进口字符串的计算规则，但是以该形式的进口节点会导制第二个风口成为孤立风口，也会导致第一个接入口没有可以连接的风口，成为供气断点。因此，需要对计算得到的风口字符串再进行筛选，使其表示的接入情况可以满足 3.2 节建立连接规则。

在风口行连接中，假如一个风口被前一行风口直接连接后，为了避免空间三通的形成，该风口不能再进行水平方向上的连接，此时的风口接入点只能给一个风口进行气体输配，本文将接入点位于风口节点位置的情况称为风口直连。当接入点位于风口行的虚拟节点时，一个接入口可以同时给同一行的多个风口供气，并且可能产生多种供气组合。由于接入点类型的不同会产生不同的连接方式，本文将进口字符串在风口直连位置进行分解，将一个进口字符串拆分为多个不含风口直连情况的字符段和若干个风口直连节点。当进口字符串不含风口直连时，则不进行拆分。风口直连节点只有唯一的连接情况且满足单行风口的连接规则，因此不需要再进行分析，不含风口直连的进口字符段可能产生多种连接情况，同时也可能产生不符合风口连接规则的接入状态，因此需要更深入地讨论。

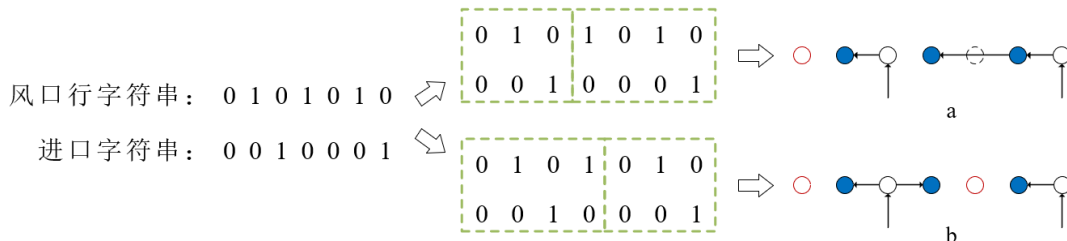
对于一个进口字符串而言，假如该字符串符合单行风口的连接规则，则其拆解后形成的每个不包含风口直连的字符段都符合单行风口的连接规则。利用进口字符串的特性可以对每个字符段进行判断，辨别一个进口字符串的是否符合单行风口的连接规则。对于每个字符段，都需要满足字符“1”的个数大于 1 且同时小于字符段所对应的风口个数。表 3.1 是对每个不含风口直连的字符段的判断算法伪代码，当算法返回的结果为 True 时，该字符段满足单行风口的连接要求。

表 3.1 筛选进口字符串算法

输入： S , 进口字符串
输出： True 或者 False, S 的可行性, True 为满足连接要求, False 为不满足
1: $S_p :=$ 用于储存 S 的部分字符段的集合
2: for each character c in S do
3: if c 是 "1" and 位置索引为偶数 then
4: $S_p :=$ 在 c 处对 S 进行分段并删除 c 后产生的字符段
5: end for
6: for 每个字符段 s in S_p do
7: if s 中 "1" 的个数 $> (s$ 的字符个数 $- 1) / 2$ then
8: return False
9: end if
10: end for
11: return True

3.3.3.3 风口段连接方案生成

由于下虚拟行连接、中间虚拟行连接和风口直连都只有一个接入口，因此其对应的输配方案只有一种。但对于不包含风口直连的字符段来说，接入口可以有多个，且接入口的分配可能有多种方式，因此，一段进口字符串会产生多种符合规则的风口连接方案，如图 3.13 所示。因此，需要在遍历过程中对每个不含风口直连的字符段进行可行方案计算。以下计算只讨论某一段风口段的连接过程，为了便于说明，下文采用的位置信息是风口段的相对位置，即第一个节点的位置为 1，而不是采用风口段在整个进口字符串中的绝对位置作为其位置信息。因此，实际算法中，需要再添加相对位置信息和绝对位置信息的转化过程。



进口字符串与风口的连接过程也可以视为风口的分配问题，风口的分配也必须符合前文所建立的所有规则：每个风口都必须被分配给相邻的某个接入口，不能跨过接入口进行连接；同时每个接入口都必须与一个以上风口进行连接。对于一段不含风口直连的进口字符串 S' ，假设其包含 n' 个风口，则符合上述规则的风口分配结果可以用公式 3.6 表示：

$$Dis = \{(Y_1, Y_2, \dots, Y_m) \mid \bigcup_{i=1}^m Y_i = \{2, 4, \dots, 2n'\}, \bigcap_{i=1}^m Y_i = \Phi, Y_i \in Z, \forall y \in Y_i, Adj_{i_i}^- < y < Adj_{i_i}^+, i = 1, 2, \dots, m\} \quad (3.6)$$

其中， Dis 表示包含所有符合连接规则的风口分配结果的集合，该集合中的每个元素 (Y_1, Y_2, \dots, Y_m) 都表示一种可行的风口分配情况； m 为进口字符串中接入口的个数； I 是所有接入口节点的位置信息集合； Y_i 是被分配到第 i 个接入口 I_i 的风口节点位置信息集合； Y_i 属于集合 Z ，集合 Z 包含所有的风口分配结果； $Adj_{i_i}^-$ and $Adj_{i_i}^+$ 分别表示第 i 个接入口的左相邻接入口和右相邻接入口的位置信息。

I 、 m 和 Z 可以通过以下公式进行计算：

$$I = \{x \mid S'_x = 1, x \in \{1, 3, \dots, 2n' + 1\}\} \quad (3.7)$$

$$m = |I| \quad (3.8)$$

$$Z = \{Y \mid Y \subset \{2, 4, \dots, 2n'\}, Y \neq \Phi\} \quad (3.9)$$

$Adj_{i_i}^-$ 和 $Adj_{i_i}^+$ 则可以由以下公式表示：

$$Adj_{i_i}^- = \begin{cases} 1, & L_{i_i} = \Phi \\ \max(L_{i_i}), & \text{else} \end{cases} \quad (3.10)$$

$$Adj_{i_i}^+ = \begin{cases} 2n' + 1, & R_{i_i} = \Phi \\ \min(R_{i_i}), & \text{else} \end{cases} \quad (3.11)$$

其中， L_{i_i} 和 R_{i_i} 分别表示第 i 个接入口左边和右边的所有接入口的位置信息集合，可以由下列公式计算得到：

$$L_{i_i} = \{x \mid x \in I, x < I_i\} \quad (3.12)$$

$$R_{i_i} = \{x \mid x \in I, x > I_i\} \quad (3.13)$$

除了风口行中的风口节点具有多种分配方式外，风口行中未被连接的虚拟节点（图 3.13 中红色圆圈，本文称为自由虚拟节点）也存在多种分配方式，与风口不同的是，自由虚拟节点没有必须被连接的限制，因此，一个自由虚拟节点通常具备三种合理的连接状态：未连接、连接到左边风口节点和连接到右边风口节点。

一段不含风口直连的进口字符串与风口行进行连接后产生的所有自由虚拟节点可以通过公式 3.14 进行计算， NV 表示自由虚拟节点的位置信息集合。

$$NV = O' \setminus C_{A,V} \quad (3.14)$$

其中, Q' 包含进口字符串中所有节点的位置信息, 节点位置是相对位置信息, 因此, $O' = \{1, 2, \dots, 2n' + 1\}$; $C_{A,V}$ 是包含所有风口节点位置信息和部分虚拟节点位置信息的集合, 此类虚拟节点不属于自由虚拟节点, 是位于接入口位置或处于风口行连接中的两个相连风口中间的虚拟节点 (如图 3.13 中所示虚线表示的虚拟节点), $C_{A,V}$ 可以通过以下公式计算得到:

$$C_{A,V} = \bigcup_{i=1}^m \{x \mid \min(Y_i \cup \{I_i\}) \leq x \leq \max(Y_i \cup \{I_i\}), x \in O'\} \quad (3.15)$$

对于第 i 个自由虚拟节点, 其连接情况可以由下列公式表示:

$$PV_i = \{\{NV_i - 1, NV_i\}, \{\}, \{NV_i, NV_i + 1\}\} \quad (3.16)$$

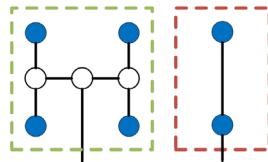
基于自由虚拟节点的连接特性, 一段风口段中的所有自由虚拟节点的所有连接组合情况可以通过下列公式表示:

$$CV = \{U \mid U = \bigcup_{i=1}^k \{W \mid W \subset PV_i, |W| = 1\}, \bigcap_{i=1}^k U_i = \Phi, \bigcup_{i=1}^k U_i \subset O'\} \quad (3.17)$$

其中, CV 是包含所有自由虚拟节点的可行的连接方案集合; k 是自由虚拟节点的个数, $k = |NV|$; CV 的每个元素 U 也是一个集合, 包含所有自由虚拟节点的一种连接方式; U_i 是 U 中的第 i 个元素, 和集合 W 相同, 表示第 i 个自由虚拟节点的一种连接方式, 是集合 PV_i 的其中一个子集, W 和 U_i 的元素个数都为 1; 限制条件 $\bigcap_{i=1}^k U_i = \Phi$ 保证了任意两个自由虚拟节点不同时连接到相同的风口节点上; 位于字符串两端的自由虚拟节点只有两种连接方式: 不被连接或连接到左/右边风口节点, 使用条件 $\bigcup_{i=1}^k U_i \subset O'$ 来保证位于端点的自由虚拟节点的特殊性。

以单行风口表示下虚拟行连接和中间虚拟行连接时, 自由虚拟节点只能位于进口字符段的端点且不能为接入节点。因此, 自由虚拟节点的位置信息集合为: $NV = \{x \mid x \in \{1, 2n' + 1\}, x \notin I\}$ 。虚拟节点的分配过程和连接结果的求解都和不含风口直连的风口段的过程一致, 遵循公式 3.15-3.17。

前面的所有算法过程都是建立在一行风口只按风口行连接或者虚拟行连接的假设上进行的。但是实际风管布置过程中, 两行风口连接时, 同时应用风口行连接和虚拟行连接会产生非常优质的连接结果, 如图 3.14 所示。此处风口行连接和虚拟行连接的组合与 3.2.2 节中讨论的混合型连接不同, 该连接方式只针对两行风口, 且两行风口行对应位置上的所有风口都采用相同连接方式, 本文将这种连接方式称为组合连接。



a. 虚拟行连接; b. 风口行连接;

图 3.14 单行风口行连接中包含两种连接方式

组合连接可以通过风口行字符串的分段来实现，将拆分后形成的多个风口段按照相邻风口段采用不同连接方式的规则来计算。两行风口段的拆分必须满足每个风口段都包含一个以上的风口，且每个风口段中的风口位置都是连续的。对于含有 n 个风口的风口行，满足上述要求的所有风口段拆分结果的集合 C_{om} 为：

$$C_{om} = \bigcup_{m=1}^n \{(H_1, \dots, H_m) \mid \bigcup_{i=1}^m H_i = O, \bigcap_{i=1}^m H_i = \Phi, H_i = \{\min H_i, \dots, \max H_i\}, i = 1, \dots, m\} \quad (3.18)$$

其中， O 是所有风口节点的位置信息集合， $O = \{1, 2, \dots, n\}$ ； m 是接入口个数； H_i 是拆分后的第 i 段风口段所包含的风口位置信息集合；限制条件 $\bigcup_{i=1}^m H_i = O$ 保证所有风口都被分配到接入口；限制条件 $\bigcap_{i=1}^m H_i = \Phi$ 保证每个风口只在一个风口段内；限制条件 $H_i = \{\min H_i, \dots, \max H_i\}$ 保证每一段风口段中的风口节点位置都是连续的。当进行组合连接时，需要保证两个相邻风口段之间的虚拟节点不同时连接到这两个风口段上。

对于一个具有 n_y 行风口的节点集，由于两行风口可以进行虚拟行连接，或者进行风口行连接，因此，任意两行相邻风口行都可以进行组合，并在计算过程中以单行风口表示。所有风口行的组合可以用以下公式表示：

$$R_{com} = \bigcup_{n'_y = \lceil n_y / 2 \rceil}^{n_y} \{(R_1, \dots, R_{n'_y}) \mid R_i \subset \{1, \dots, n_y\}, \max R_i - \min R_i = |R_i| - 1, i = 1, \dots, n'_y\} \quad (3.19)$$

R_{com} 是所有风口行组合结果的集合，其中的一个元素表示一种风口行组合方式。 n'_y 是组合后用于计算的风口行数，其最小取值情况是最大限度地进行中间虚拟行连接，最小取值为 $\lceil n_y / 2 \rceil$ ，符号“ $\lceil \rceil$ ”表示向上取整函数，其最大取值情况是不进行虚拟行连接，最大取值为 n_y 。组合前后的行数位置信息用集合 R_i 表示， i 是组合后的行数， R_i 中的元素是组合前的行数，该集合的元素个数为 1 或者 2。限制条件 $\max R_i - \min R_i = |R_i| - 1$ 保证了进行组合的两行风口相邻。

对风口行组合后的每个结果都进行逐行计算，当遍历筛选完一行风口行的所有连接方案后，计算每一个连接方案所对应的下一行风口行的输入参数，再计算下一行风口行的所有连接方案。将该过程重复直到最后一行风口，则可以得到符合所有风口连接规则的节点连接拓扑。

3.3.4 节点位置调整及障碍规避

在前面几节进行的风口节点连接方案计算过程中，风口节点的位置是相对位置，通过风口位置的行和列作为位置信息，因此得到的节点连接拓扑也是相对位置连接拓扑。在实际风管设计时，需要考虑每段管道的阻力特性，因此，需要将

节点的位置对应到现实坐标上,才可以进行水力计算。由于风口的定位是固定的,需要确定的是风口之间的虚拟节点的实际坐标。在 3.1 节建立模型时认为虚拟节点均匀分布在风口之间,但是实际工程中可以对部分虚拟节点进行适当地偏移,以节省管材;风口的定位会考虑障碍物的影响,但认为虚拟节点处于风口节点中间时没有考虑障碍规避,同时,节点之间的连接边也可能与障碍物发生重叠,因此需要对虚拟节点和连接拓扑的边进行碰撞检测和位置调整。

在实际工程中,虚拟节点的位置主要有三个,如图 3.15 所示。当虚拟节点处于某段连接的端点时,为了节省管材,实际布置管道时会将虚拟节点与风口的距离调整为安装弯头和连接风管所需要的最小安装距离 $D_{install}$;当虚拟节点左右都连接到风口时,为了保证风口风量输配的平衡性,一般以两个风口连线的中点作为虚拟节点的实际位置。

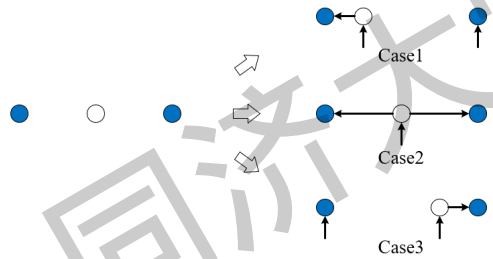


图 3.15 不同情况下虚拟节点的实际位置

由于实际建筑的房间中可能存在障碍物(如柱子),两个节点之间连接的风管可能会与之发生碰撞。为了避免得到的最优设计方案无法用于实际工程,在节点连接拓扑转化为节点的现实坐标连接过程中需要考虑对障碍物的规避。障碍物与连接方案的重叠可分为两种情况,即边重叠和节点重叠。为了避免节点位置变动的可能性过多导致计算的判定过程过于复杂,本文将连接拓扑中的节点分为固定连接节点和可变连接节点(图 3.16)。固定连接节点是指其位置和连接方向(水平或垂直)都不可改变的节点,此类节点的位置变动会对其上游或者下游节点的位置产生连锁影响,具体是指非端点的风口、三通和四通。可变连接节点是指其位置或连接方向可以改变的节点,此类节点的位置变动只影响其相邻的节点位置,同时该影响不会扩散到其他节点,对应的实际节点是位于端点的风口和弯头。

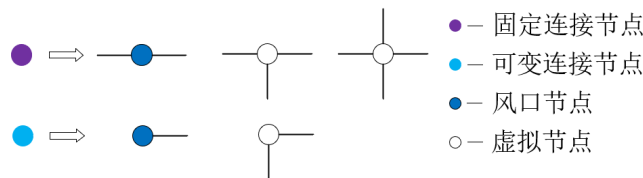


图 3.16 节点所对应连接情况

根据边两端节点的类型可以将边与障碍物的碰撞分为三种情况。当边的两个节点都是固定节点时，只能移动边的一部分进行障碍规避，如图 3.17a，边的移动距离为与障碍物的最小安装距离；当边的其中一个节点是可变连接节点时，同时移动部分边和可变动节点进行边的避障（图 3.17b），此方案是局部阻力构件最少的方案；当边的两个端点都是可变动连接节点时，直接偏移边是最优的避障方案（图 3.17c）。

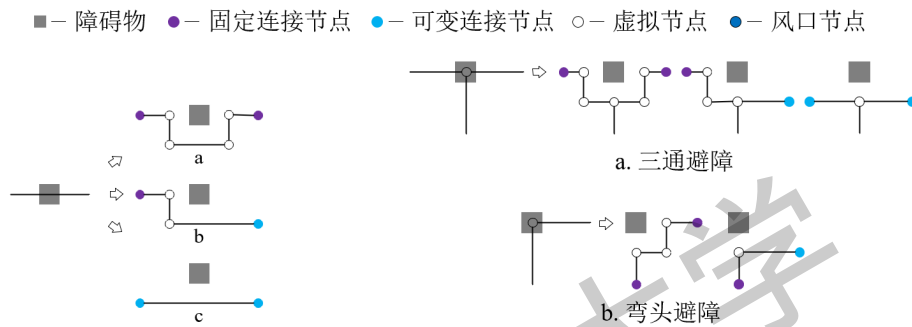


图 3.17 三种边的避障方案 图 3.18 虚拟节点的障碍规避方式

由于风口的布置会避开障碍物，与建筑结构重叠的节点只能是虚拟节点。在风口连接方案中，虚拟节点相当于风管中的弯头、三通或四通。当四通与障碍物重叠时，调整四通节点的位置无法规避障碍物，与四通相连的其他边依然会发生碰撞。因此，存在四通与障碍物碰撞的方案都是无法进行工程施工的方案，将其进行筛选。弯头和三通的位置调整如图 3.18 所示。根据其相连节点的类型进行最适合的位置调整。三通的避障主要考虑两个平行分支，处理方式与边的碰撞类似。节点和边的偏移都是按照水平或垂直偏移，使用最小安装距离作为偏移距离。

3.4 本章小结

本章主要介绍了房间内风口连接拓扑的遍历算法，算法的目标是快速筛选出符合实际工程的风口连接方案。基于实际案例，本章建立了房间内风口连接的节点模型，同时为了实现算法目标，提出了快速筛选规则。规则主要分为两类，一类是风口连接必须遵循的客观规则，不符合该规则的方案无法实现风管系统的输配功能；另一类规则是为了筛除非最优方案而定义的主观规则，不符合该规则的连接方案也可以实现风量输配，但是会造成水力失衡、管材耗费大和运行阻力大等问题。基于建立的规则，本章详细地介绍了风口连接遍历算法的实现过程，主要包含风口预处理、输入参数计算、风口行连接方案计算和节点位置调整等过程。

第4章 快速水力计算模型

4.1 概述

通过风口连接算法计算后,单个房间内的所有满足连接规则的风口连接拓扑都被筛选出来,在确定所有节点的坐标后,可以开始进行最优布置方案的选取。在实际工程中,风管的设计主要关注管网系统的水力平衡、管材消耗和运行阻力大小等问题。暖通工程师所关注的是多个目标综合下的最优设计结果,而这些目标的具体参数只能在管道连接方案的水力计算后才可以得到。传统的空调系统水力计算需要反复进行查表和插值,无法应用于大量的计算案例上。因此,本章基于现有的水力计算方法和一些合理的假设,提出了一个适用于有向图的快速水力计算模型,主要解决在众多风口连接拓扑中挑选最优连接方案的问题,同时实现多目标规划。快速水力计算模型主要分为两部分:风管选型和有向图的水力计算。

4.2 风管选型算法

由于风管设计过程需要考虑管道耗材,对一个连接方案来说,当节点的位置和节点间的连接关系确定后,影响管道耗材的因素就是管道的尺寸。因此,在本文建立的快速水力计算模型中考虑了风管尺寸的选择。传统的空调系统风管设计是通过风管输配的风量和设计风速求得风管的横截面积,在标准风管尺寸表中进行面积匹配来确定具体的风管尺寸。对于数量较大的节点拓扑来说,大量的查表比对过程会严重影响算法的运行速度。同时,挑选最优设计方案是多个连接结果进行比较的过程,当计算方式相同时,相对精确的数值就可以体现方案之间的差别,筛选过程中对每个方案的风管尺寸都进行精确设计会导致计算成本过高,且意义不大,因此本文提出了快速风管选型算法,在方案筛选时省略查表和插值过程。本文的风管选型算法只适用于矩形风管的尺寸选型,基于以下假设:所有风管尺寸中最短边的长度为 l_{\min} ,风管的其他边长 l' 都为最短边长的 α 的整数次方倍,即:

$$l' = \alpha^i l_{\min}, i \in \mathbb{N} \quad (4.1)$$

其中, \mathbb{N} 为整数集合; α 为本文规定的长宽比。进行风管设计时,风管横截面积

的计算公式为：

$$A = \frac{L}{v_{\text{set}}} \quad (4.2)$$

其中， A 是对应风量和设计风速下的风管横截面积， m^2 ； L 是风管的风量， m^3/s ； v_{set} 是风管的设计风速， m/s 。

基于前述风管边长的假设，根据几何知识，任意风管的横截面积是最小风管横截面积 l_{min}^2 的 α 的整数次方倍，即：

$$A_s = \alpha^x l_{\text{min}}^2, x \in \mathbb{N} \quad (4.3)$$

其中， A_s 表示风管的标准尺寸对应下的风管横截面积， m^2 。风管的实际面积应该尽量接近设计风速下的风管横截面积，即 $A \approx A_s$ 。通过数学变换，可得：

$$x = \left[\frac{\lg L - \lg v_{\text{set}} - 2 \lg l_{\text{min}}}{\lg \alpha} \right] \quad (4.4)$$

其中， x 表示风管选型后的实际面积对应的幂次数；符号 $[\]$ 表示取整函数。

风管设计的目标包括实现最小的输配阻力同时节省空间。当矩形风管的横截面积相同时，长宽比越小，风管的水力半径越大，单位管长的沿程阻力越小。而对于施工安装而言，矩形风管的宽（三维中即为风管厚度）越小，安装占据的空间越小，吊顶高度可以增加，空间利用率更大，因此，风管的长宽比不能太大也不宜为 1。考虑到以上要求，本文将风管的长宽比规定为：

$$\frac{l}{w} = \begin{cases} \alpha, & x = \text{奇数} \\ \alpha^2, & x = \text{偶数} \end{cases} \quad (4.5)$$

此时，风管的长 l 、宽 w 和周长 Pe 为：

$$l = \begin{cases} \alpha^{\frac{x+1}{2}} l_{\text{min}}, & x = \text{奇数} \\ \alpha^{\frac{x+1}{2}} l_{\text{min}}, & x = \text{偶数} \end{cases} \quad (4.6)$$

$$w = \begin{cases} \alpha^{\frac{x-1}{2}} l_{\text{min}}, & x = \text{奇数} \\ \alpha^{\frac{x-1}{2}} l_{\text{min}}, & x = \text{偶数} \end{cases} \quad (4.7)$$

$$Pe = 2(l + w) = \begin{cases} 2l_{\text{min}} (\alpha^{\frac{x+1}{2}} + \alpha^{\frac{x-1}{2}}), & x = \text{奇数} \\ 2l_{\text{min}} (\alpha^{\frac{x+1}{2}} + \alpha^{\frac{x-1}{2}}), & x = \text{偶数} \end{cases} \quad (4.8)$$

基于风管长宽比为固定数值的假设,在风管的输配风量和设计风速确定的情况下,采用公式 4.6–4.8 可以快速进行风管选型。实际计算中,可以通过改变 α 和 l_{\min} 的数值来满足不同的设计需求。

4.3 有向图的水力计算

在完成风管节点连接拓扑生成和节点位置坐标的确定后,需要对连接方案进行水力计算,传统风系统的管网水力计算是采用最不利环路末端风口逐步向前计算每根支管水力特性的方法,该过程可以采用深度优先搜索算法实现。将风口的连接拓扑图转化成与风量输配方向相对应的有向图,再规定节点和边的相关特性和计算方法,便可以计算得到每个风口连接方案的水力特性。

4.3.1 算法输入和相关计算公式

在开始进行水力计算前,需要将风口连接拓扑转化成有向图,同时确定每个风口的风量和其他相关的计算公式。快速水力计算模型的输入参数是包含所有节点位置信息、节点风量的风口连接有向图。节点的位置信息是指风管的相关部件在空间中的坐标,节点处的风量值是指节点给空间的送风量或者回风量。所有风口节点的风量为风量设计值;虚拟节点对应的是局部阻力构件,在风管系统中只起输送空气作用,节点的进出口处风量相同,因此,所有虚拟节点的风量值为 0。

布置方案的水力稳定性用于描述支管之间的阻力大小关系,本文以参数 UB 来描述支管间的相对稳定性,该特性属于节点的特性,值为 0 或 1。当节点的支管间不平衡率大于允许值 B_{un} 时,认为该节点为不平衡节点, UB 取值为 1;当节点支管间不平衡率小于等于 B_{un} 时,认为该节点为平衡节点, UB 取值为 0。为了避免受计算机保存数值的小数位数偏差影响, B_{un} 的取值不宜太小,否则容易得到所有节点都不平衡的结果。同时,为了更好地地区分方案间的优劣, B_{un} 的取值也不宜过大,否则容易导致被判定为最平衡连接的方案个数过多,造成计算量过大。本文通过实际的计算过程测试,发现 $B_{un}=0.001$ 时效果最理想。有关 UB 的详细计算过程见 4.2.2 节。

在水力计算过程中,需要计算每段管道的阻力和管道耗材,相关的计算公式通过设计手册^[86]获得,如表 4.1 所示:

表 4.1 水力计算过程相关公式

计算项	公式或相关描述
风管类型及材料	矩形风管，镀锌钢管，绝对粗糙度为 0.15 mm
沿程阻力计算公式	$\Delta P_f = 1.05 \times 10^{-2} \times D^{-1.21} \times (Fedge.V)^{1.925} \times Fedge.L$ $D = \frac{2lw}{l+w}$
局部阻力计算公式	$\Delta P_l = \xi \frac{Sedge.V^2}{2g}$
局部阻力系数 ξ	变径 $\xi = 0.065 \frac{Fedge.A}{Sedge.A} - 0.036$
	弯头 $\xi = 0.11$
	三通 旁通管: $\xi = 0.5 \left(\frac{Sedge.V}{Fedge.V} \right)^2 + 1$
	直通管: $\xi = 0.35 \left(1 - \frac{Sedge.V}{Fedge.V} \right)^2$
四通	视为两个三通相连

4.3.2 计算过程

当算法的输入参数和相关的计算公式确定后，开始计算风管连接方案的水力特性。本文中风管的水力计算过程包含每段管段的风管选型、局部阻力计算、沿程阻力计算和每个节点的平衡性计算。每段风管的上述所有参数都是风量和风速的函数，即：

$$Duct_{\text{参}} = f(L, v) \tag{4.9}$$

每段风管的设计风速为设定值，因此在计算过程中，对其他参数产生影响的变量为风管的输配风量，只要确定风量便可求解其他参数。每段风管的风量等于其下游所连接的支管风量之和，因此，计算每段管段的风量都必须先得到其下游所连接支管的风量值。该问题可以采用深度优先搜索算法进行求解。

深度优先搜索算法(Depth First Search, DFS)是一种经典的搜索算法，属于图算法的一种。其过程简要来说是对每一个可能的分支路径深入到不能再深入为止，而且每个节点只能访问一次。其采用了回溯的思想，可通过根节点向下游搜索遍

历至所有节点，搜索算法如下：

- a. 随机访问与根节点相连的一个子节点；
- b. 将当前节点视为根节点，重复步骤 a，直到当前节点不存在相应的子节点；
- c. 从当前节点向上游回溯至具有未被访问子节点的父节点，随机访问父节点的其他未被访问的子节点，重复步骤 b。
- d. 重复步骤 c，直到所有节点被访问，停止搜索算法。

深度优先搜索算法在访问节点时，需要进行节点以及节点之间的连接边的属性计算。为了便于理解，采用图 4.1 进行过程描述，节点和边的所有属性如图所示。图中 Node 表示当前访问的节点，其上下游节点分别为 Fnode 和 Snode，与上下游节点相连的边分别为 Fedge 和 Sedge。由于可以有多个下游节点，因此用 Snode_j 和 Sedge_j 分别表示 Node 的第 j 个相连的子节点和边。

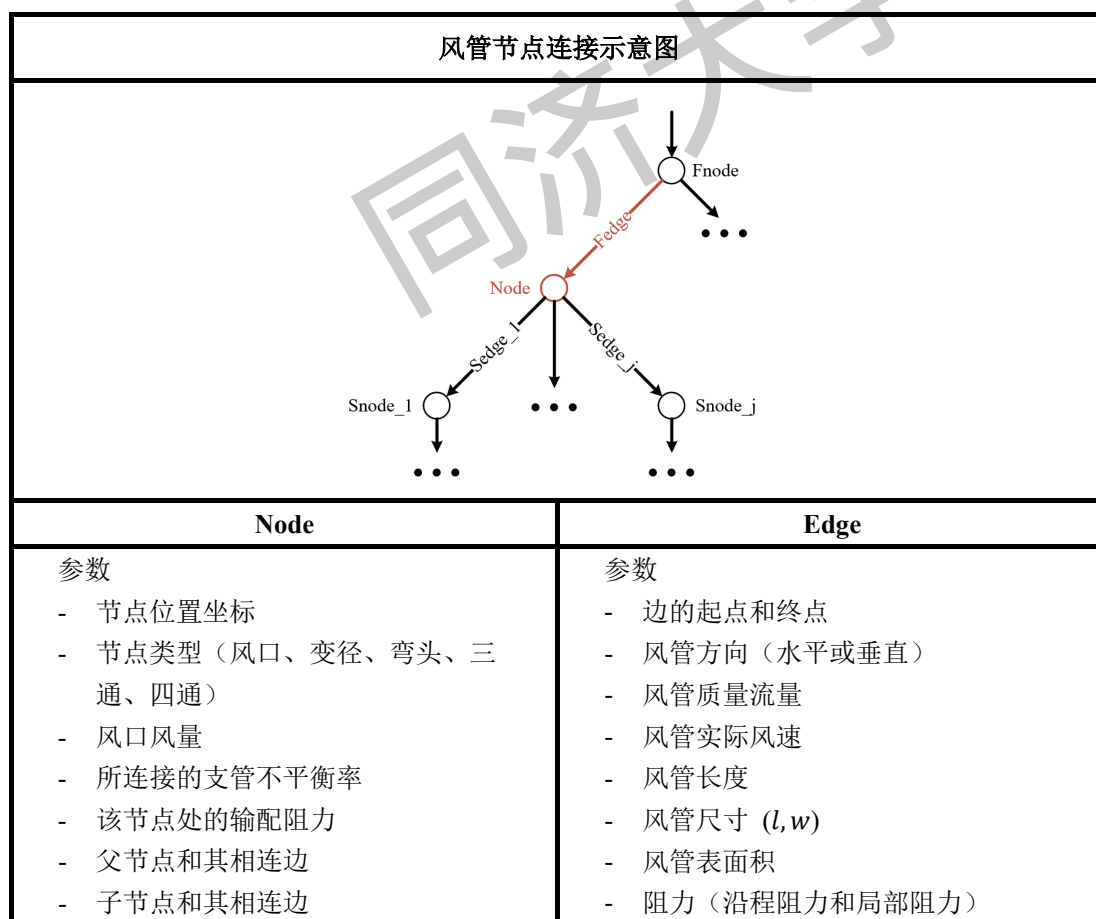


图 4.1 风管连接有向图的节点和边的属性

当采用深度优先搜索算法进行节点访问时，当前访问节点 Node 的子节点进行的所有计算过程与 Node 相同。假设在完成所有 Snode 的计算后访问 Node，此

时需要计算当前节点的属性值以及其相连边 *Sedge* 和 *Fedge* 的属性值。为了保证下游节点回溯时计算的一致性和稳定性，当前节点计算的所有参数的输入值都必须为预设值或者为上一节点计算的输出值，亦即，每个节点处进行计算的输出参数都必须包含所有输入的子节点的同名参数。由于风管水力计算过程中与所有水力特性变量都相关的唯一的未知变量是风管的风量，*Node* 的计算输入变量包含 *Sedge* 的风量，输出参数一定具有 *Fedge* 的风量。计算公式为：

$$Fedge.F = Node.F + \sum_{i=1}^j Sedge_i.F \quad (4.10)$$

其中，*Fedge.F*、*Node.F* 和 *Sedge_i.F* 都为对应边和节点的风量。

在 *Fedge* 的风量已知的情况下，通过预设参数和已访问节点的输出参数可以求得 *Fedge* 的其他参数。由于所有节点访问过程中的输入参数和输出参数保持一致，*Node* 和 *Snode* 在各自计算过程中输出参数相同。相关参数的计算公式如下：

$$Fedge.S = (l, w) = f(Fedge.F, l_{\min}, v_{set}, \alpha) \quad (4.11)$$

$$Fedge.L = \sqrt{(Fnode.X - Node.X)^2 + (Fnode.Y - Node.Y)^2} \quad (4.12)$$

$$Fedge.SA = 2 \times (l + w) \times Fedge.L \quad (4.13)$$

$$Fedge.V = \frac{Fedge.F}{Fedge.A} \quad (4.14)$$

$$Fedge.FR = f(Fedge.F, Fedge.S, Fedge.L) \quad (4.15)$$

其中，*Fedge.S* 为 *Fedge* 对应风管的尺寸，具体的计算公式为公式 4.6 - 4.8；*Fnode.X*、*Fnode.Y*、*Node.X* 和 *Node.Y* 为对应节点的横纵坐标值；*Fedge.L* 为 *Fedge* 对应的风管长度；*Fedge.SA* 为 *Fedge* 对应风管的表面积；*Fedge.V* 为 *Fedge* 对应风管的实际风速；*Fedge.FR* 为 *Fedge* 对应风管的沿程阻力，具体公式为表 4.1 中的沿程阻力计算公式。

由于节点所连接的下游支管需要的输配阻力不同，为了保证风量输配的可靠性，风机基于最大输配阻力进行选型。因此，将节点的下游支管中的最大阻力值作为节点的一个属性进行储存，作为衡量方案优劣的一个参数，记为节点的阻力值 *Node.R*，其相关计算公式为：

$$Sedge.LR = f(Fedge.V, Fedge.S, Sedge.V, Sedge.S) \quad (4.16)$$

$$Sedge.R = Sedge.LR + Sedge.FR \quad (4.17)$$

$$Node.R = \max(Sedge_i.R + Snode_i.R), i = 1, \dots, j \quad (4.18)$$

其中, $Sedge.LR$ 为下游边的局部阻力, 由于 $Sedge$ 的局部阻力与 $Fedge$ 的参数相关, 只能在进行 $Fedge$ 的计算后才可以求解; $Sedge.R$ 是下游边的总阻力, 等于下游边沿程阻力 $Sedge.FR$ 和局部阻力之和, 由于 $Sedge.LR$ 只能在访问 $Node$ 时求解, $Sedge.R$ 也只能在 $Node$ 节点的水力计算过程计算; $Snode.i.R$ 是子节点的阻力值, 由于在计算 $Node$ 步骤中可以计算得到 $Node.R$ 的值, 则在子节点的访问过程中同样可以输出其节点阻力值。同理, $Fedge.FR$ 的值可以在访问 $Node$ 过程中求解, 则 $Sedge.FR$ 也可以在子节点的搜索过程中进行求解。

为了评估风管连接方案的水力稳定性, 本文采用访问节点的下游支管间的阻力偏差作为衡量参数, 即为 4.2.1 节中提出的节点的 UB 属性, 其计算公式为:

$$Node.UB = \begin{cases} 0, & \forall i \in [1, j], \frac{Node.R - (Sedge.i.R + Snode.i.R)}{Node.R} \leq B_{un} \\ 1, & else \end{cases} \quad (4.19)$$

综上, 在访问节点中进行运算所涉及的输入参数、预设参数和输出参数如表 4.2 所示。访问节点的输出参数包含所有输入的子节点的同名参数 (F 、 R 和 FR), 可以保证算法的稳定性。进行深度优先搜索算法时, 遍历求解所有子节点的对应参数后, 回溯到父节点进行相同的求解过程, 逐个节点求解直至完成风管起始节点的属性求解。

表 4.2 访问节点 $Node$ 时运算所涉及的参数

预设参数	输入参数	输出参数
		上游边风量 $Fedge.F$
访问节点坐标($Node.X, Node.Y$)		访问节点阻力 $Node.R$
父节点坐标($Fnode.X, Fnode.Y$)	下游边风量 $Sedge.F$	上游边沿程阻力 $Fedge.FR$
访问节点风量 $Node.F$		下游边局部阻力 $Sedge.LR$
上游边设计风速 v_{set}	子节点阻力 $Snode.R$	下游边阻力 $Sedge.R$
最小管长 l_{min}		上游边风管尺寸 $Fedge.S$
风管长宽比 α	下游边沿程阻力 $Sedge.FR$	上游边风管表面积 $Fedge.SA$
		上游边实际风速 $Fedge.V$
		访问节点不平衡性 $Node.UB$

4.4 本章小结

本章主要介绍了适用于风口连接拓扑图的快速水力计算模型。为了避免传统空调系统风管设计过程的查表插值过程，本文基于风管长宽比遵循一定规律的假设，提出了风管选型算法，可以在确定风管风量和设计风速条件下快速进行风管选型计算。传统风管水力计算是从风管最不利支路末端逐渐往前进行风管选型和水力计算的过程，为了实现该过程在风口连接拓扑中的应用，本文将连接拓扑变换成以风管干管与房间支管连接处为起始节点的有向图，图中各节点的初始参数为节点坐标和节点风量。在明确算法输入后，本文以深度优先搜索算法对风口连接有向图进行节点和边的参数计算。对风口连接有向图进行水力计算的最终目的是对比风口连接方案的优劣，本章明确了表征连接方案优劣性质的参数，并确定了参数的从属问题（属于边或者节点）。为了保证深度优先搜索算法的稳定性和可运行性，本章建立的快速水力计算模型在每个访问节点都保证运算的输出参数包含所有输入的子节点的同名参数（边的风量 F 、边的沿程阻力 FR 和节点的阻力 R ）。同时，本章对访问节点处所有参数的计算过程进行了详细的说明和分析。

第5章 算法测试与评估

5.1 概述

在介绍完风口连接算法和快速水力计算模型后,本章主要对上述两个算法进行测试和评估,以判断该算法的有效性和适用对象。前文建立的两个算法实现的是单个房间内的风管布置最优方案计算,本章从不同优化目标、不同风口个数和不同风口排列方式角度对本文建立的算法进行测试。

5.2 优化目标

在对有向图进行阻力特性计算后,可以根据不同的目标选择风口的最佳连接。在实际工程中,风管设计最佳方案的目标可以是水力稳定性最高(Most Balanced Pressure, MBP),管道耗材最少(Least Use of Duct Material, LUM),以及输配阻力最小(Minimum Distribution Resistance, MDR)。上述三个优劣性评价参数可以通过风口连接有向图的节点和边的特性直接计算,计算公式如下:

$$MBP = \min(\sum Node.UB) \quad (5.1)$$

$$LUM = \min(\sum edge.SA) \quad (5.2)$$

$$MDR = \min(Node_{start}.R) \quad (5.3)$$

其中, $Node_{start}$ 是每个连接方案的起始节点; $\sum Node.UB$ 为单个方案的不平衡率; $\sum edge.SA$ 为单个方案的管道表面积; $Node_{start}.R$ 为单个方案所需的输配阻力。

由于不同设计项目所要求的风管设计最优目标可能不同,用户可以根据实际需求对上述三个参数进行重要性排序。当两个方案的相同优化目标值处于一定的差异率 η 范围内时,认为这两个方案在该目标下的优劣性一致,则需要通过下一个优化目标的比较进行方案选择,差异率的计算公式为:

$$\eta' = \frac{|Value_1 - Value_2|}{\max(Value_1, Value_2)} \quad (5.4)$$

其中, η' 为两个方案的实际差异率; $Value_1$ 和 $Value_2$ 为两个方案的优化目标所对应的取值。当 $\eta' \leq \eta$ 时,认为两个方案在某个优化目标下是相同的。

本文关注的三个优化目标根据其特点可分为稳定性目标（MBP）和经济性目标（LUM, MDR）。对它们进行排序组合，可以产生四种综合目标：PMR - [MBP, LUM, MDR]; PRM - [MBP, MDR, LUM]; MRP - [LUM, MDR, MBP]和 RMP - [MDR, LUM, MBP]。

5.3 输入参数设置

本文以 5.1 节中提出的四种综合目标进行房间内的风口连接方案生成测试。本文采用的建筑结构和房间尺寸都符合建筑设计要求^[101]，房间的长度符合建筑模数，以 4.5 米、9 米或 12 米作为房间的边长。建筑的梁结构会在高度方向对风管布置产生影响，但可以通过风管的高度调整进行梁的避让，对风管的平面连接方案不产生影响。建筑结构中对风管布置方案影响最大的是柱结构，本文在房间平面中添加了以 9 米间距排列的柱网，以检查风管连接算法规避障碍物的能力。房间中的风口采用风口布置算法进行均匀布置，风口间距为 4-6 米。算法测试过程涉及的输入参数值如表 5.1 所示：

表 5.1 房间内风口连接算法测试的输入参数

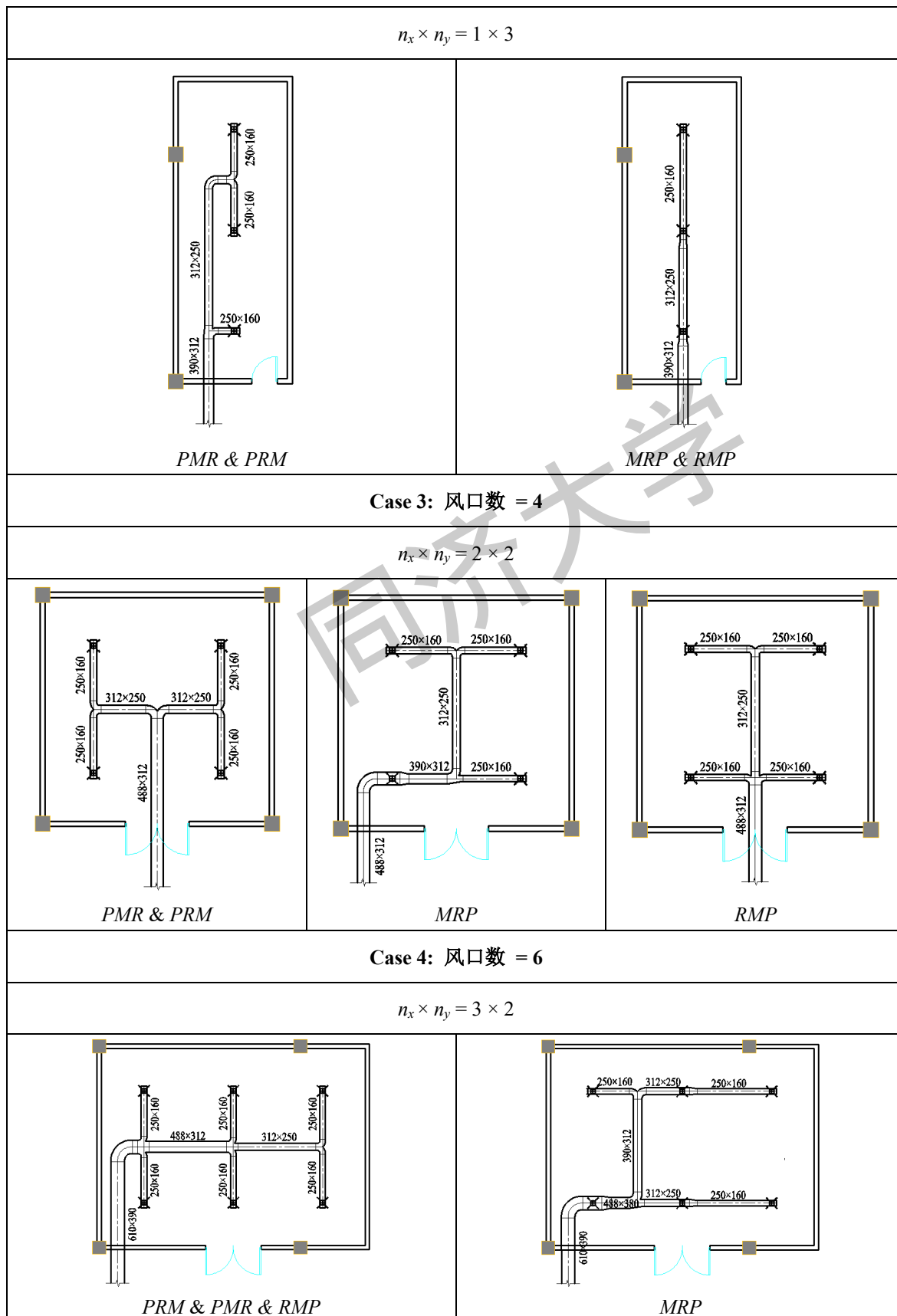
参数符号	数值	说明 ^[86]
α	1.25	房间内的风管通常比较小，因此采用比较小的长宽比计算出的风管尺寸也可以满足安装要求，同时，风管截面越接近正方形，风管的沿程阻力更小。
l_{\min}	100 mm	ASHRAE 设计手册中矩形风管的最短边的长度。
$Node.F$	0.08 m ³ /s	当房间内噪音等级为 35 ~ 50 dB 时，风口的设计风速为 1.5 ~ 2.5 m/s，本文选择 2 m/s 作为风口设计风速。假设风口为工程中常用的型号(200 mm × 200 mm)，可以计算得到风口的质量流量。
v_{set}	2 m/s	当房间的噪音等级为 35 ~ 50 dB 时，风管支管的设计风速为 2 ~ 3 m/s，本文选择中间值作为风管的设计风速。
$D_{install}$	1 m	风管局部阻力构件之间的安装距离。
$D_{arrange}$	5 m	风口布置间距。
B_{un}	0.001	不平衡率越小，计算结果越精确，但是计算速度会明显下降；不平衡率越大，会导致一些不平衡的方案被认为平衡。本文采用比局部阻力值小一个数量级的数值作为不平衡率。
η	0.001	方案差异率越小，计算结果越精确，但会导致计算速率降低；方案差异率太大则会导致计算的最优方案个数太多。本文采用比优化目标小一个量级的数值作为方案差异率。

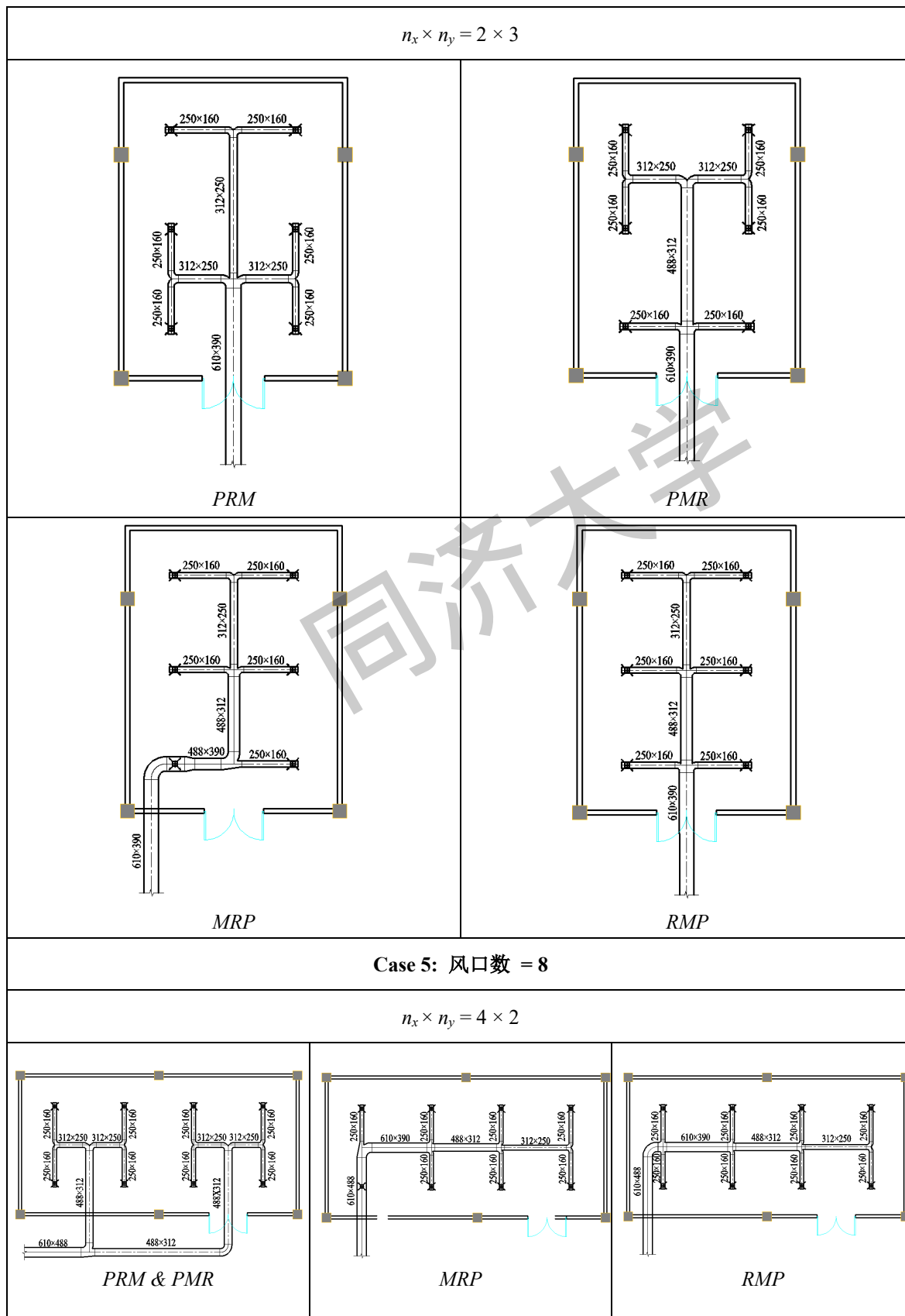
5.4 房间内风管连接案例

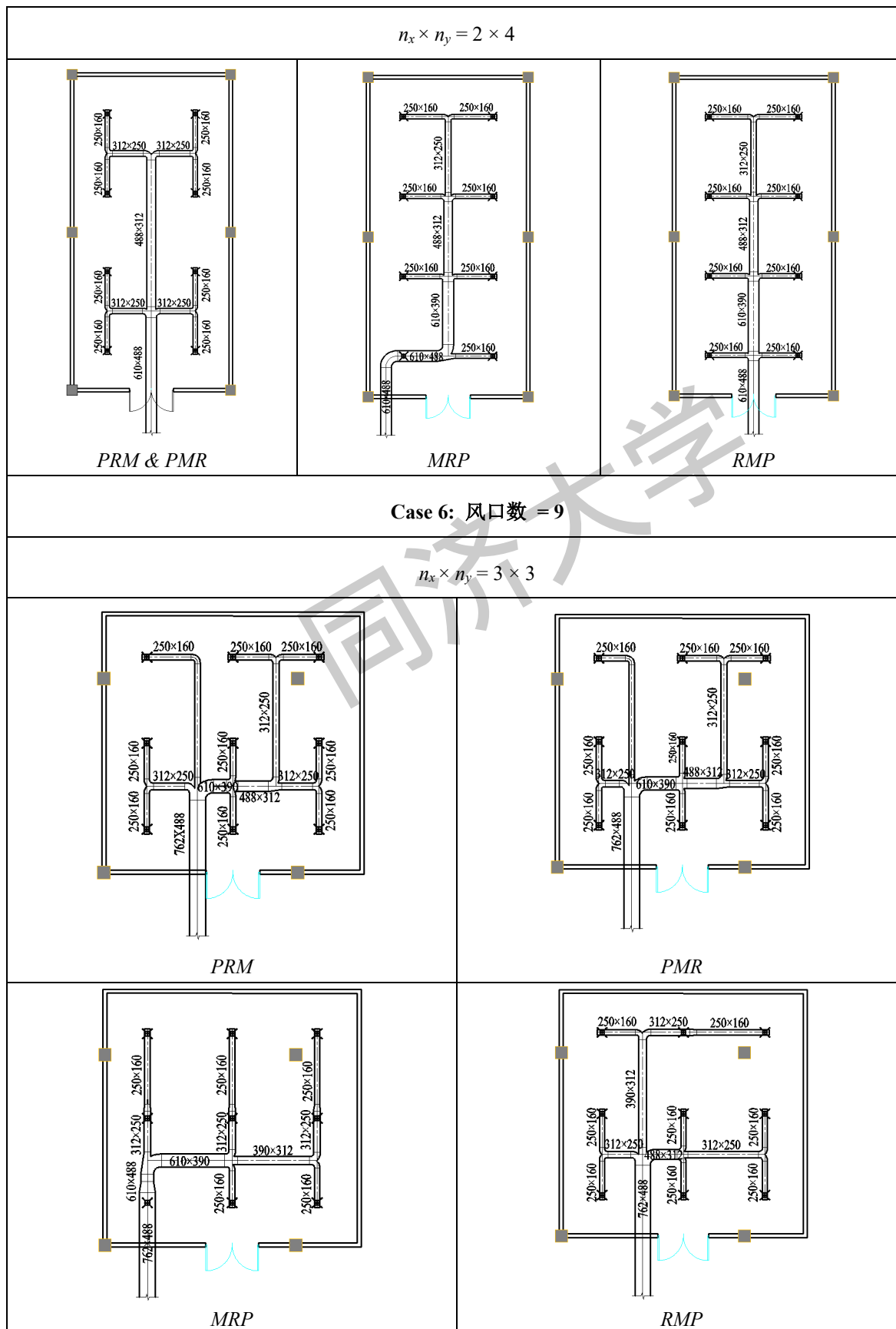
本节对风口个数不同的 6 个案例进行测试，每个案例中又分为风口行列排布不同的情况。走廊默认处于房间平面的下方，且位于走廊中的风管干管的气体输配方向是从左向右。在四种综合目标下的最优布置方案如表 5.2 所示：

表 5.2 四种综合优化目标下的最优房间风口连接结果

Case 1: 风口数 = 2		
$n_x \times n_y = 1 \times 2$		
$n_x \times n_y = 2 \times 1$		
Case 2: 风口数 = 3		
$n_x \times n_y = 3 \times 1$		







5.5 算法结果评估

采用 Python 软件对风口连接算法和快速水力计算模型进行实现，有向图的转化采用 Shapely 库进行。5.3 节中测试的 6 个案例的方案筛选结果和计算用时如表 5.3 所示。

表 5.3 风口连接算法和快速水力计算模型的运行结果

风口数 ($n_x \times n_y$)	直接进行遍历 的方案个数	风口连接算法筛 选后的方案数	运行时长 (s)	
			风口连接算法	快速水力计算模型
1×2	4.19×10^6	7	0.0020	0.011
2×1	4.19×10^6	11	0.0010	0.034
1×3	4.29×10^9	19	0.0020	0.042
3×1	4.29×10^9	39	0.0030	0.375
2×2	1.10×10^{12}	79	0.0060	0.606
2×3	2.88×10^{17}	646	0.0430	13.861
3×2	2.88×10^{17}	343	0.0160	12.780
2×4	7.56×10^{22}	5226	0.3651	271.249
4×2	7.56×10^{22}	4360	0.1700	631.052
3×3	1.93×10^{25}	8160	0.4611	1740.523

根据几种情况的计算结果，风口连接算法和快速水力计算模型具有以下特点：

a) 本文建立的风口连接算法在过滤风口节点的连接方面非常有效。基于本文提出的风口连接规则，可以筛除超过 99.9% 的不合理方案，且在进行大量方案筛选时仍能保持较快的运算速度（计算时长的数量级为 0.1s）。

b) 风口连接算法受风口个数增加的影响较小，但筛选的适合风口连接的方案个数随风口数的增加而快速增加。快速水力计算模型需要进行节点的搜索和回溯，同时需要确定虚拟节点的实际位置（3.2.4 节）并将节点连接拓扑转化成有向图，其运行时间随着连接方案的增加呈指数增长。当风口连接算法筛选后的方案个数少于 5000 个时，快速水力计算过程运行时长在 5 分钟以内，此时才能凸显自动设计的优势。

c) 最佳设计方案随着综合优化目标的变化而改变。以稳定性为首要目标时，

以管道耗材或运行阻力为次级目标产生的结果差别不大,许多情况下设计方案相同;当以经济性为首要目标时,以管道耗材或运行阻力为第一目标生成的最佳布置方案通常差别很大。

d) 用固定的风管长宽比进行快速阻力计算可能会产生非标准的风管尺寸,由于算法中风管长宽比的设定值为标准风管长宽比的众数,所以采用算法计算的风管尺寸与标准风管型号非常接近。在计算得到最佳的运行方案后,再为非标准尺寸的风管进行标准型号匹配,该过程对设计方案的水力特性影响微弱,风口连接的最优设计方案不会因此改变。

e) 由于水力计算过程随着方案个数增加而运行速度显著降低,本文提出的风口连接的最优方案计算方法有一定局限:当风口数量过大时,算法运行时间过长。对于有 9 个以上风口的房间,采用算法进行风口连接与人工绘图相比在工时上没有优势,但对于大多数房间(小于 400 平方米)的风管布置,这种方法是有效的。

基于上述特点,本文提出的风口连接最优方案生成方法(包含风口连接算法和快速水力计算模型)可以适用于大部分房间的风口连接方案生成。由于关于风管自动设计的研究较少,笔者已知的唯一研究是 Medjdoub 和 Bi 的风管半自动设计方法^[76],他们将房间内的风管定义为局部管道,所提出的算法解决案例 3 的局部管道布置需要大约 20 秒。一个有经验的设计师在使用辅助设计软件的情况下,处理同样的案例需要几分钟的时间。然而,本文提出的算法运行时间不到 1s,这在设计速度上有很大的提高。当房间数量增加时,这一优势将更加明显。

5.6 本章小结

本章对第三章建立的风口连接算法和第四章提出的快速水力计算模型进行了测试和评估,将风口连接最优方案生成方法在 6 个案例中进行测试。由于风管的设计目标可能发生变化,本章将设计师最关注的三个目标(水力稳定性、管道耗材、管网运行阻力)进行整合,提出了四种综合优化目标以及每种目标的计算方法。在实际进行算法测试时,为了保证测试的有效性,输入参数的设置非常关键,本章详细介绍了测试过程的房间大小、柱网间距以及其他设计参数的取值。在明确测试的输入后,本章展示了基于 Python 环境的风口连接最优方案生成方法的运行结果,并对风口连接算法和快速水力计算模型的运行时间分别进行了分析,在与其他研究成果和人工设计的耗时进行对比后,对算法进行了评估。

第6章 空调系统风管生成案例

6.1 概述

前文介绍了适用于单个房间的风管连接最优方案生成方法,为了实现空调风系统的自动设计,还需要处理多个房间之间的连接问题,在满足所有房间的气体输配需求前提下同时实现多个房间之间的最优连接。当涉及多个房间内的风管连接时,每个房间内的风管设计目标与系统的整体设计目标一致,即5.2节提出的四个综合目标。当确定风管系统的机房和各个房间的风口位置后,理论上可以直接对机房节点和风口节点进行连接方案遍历,但由于整个系统的房间数通常较多,遍历方案个数远远超过现有计算能力,因此,需要对风管系统的连接问题进行简化。工程中,空调系统的干管大多布置于走廊,由走廊向房间进行气体输配,本文将风管连接过程分为房间内风口连接和空调系统干管连接两部分,对各个部分分别进行最优化设计,以达到空调风系统的整体设计最优的目的。

6.2 房间内风管连接

房间内的风管连接是在确定风管干管位置和干管内的气体输送方向后,计算房间内的风口最优连接方案的过程。前文建立的风口连接最优方案生成方法需要保证风管干管可以直接对房间进行气体输配,亦即房间需要与干管所在的走廊相邻,但在实际建筑中,可能存在不与走廊直接相邻的房间,风管需要穿过其他房间对其进行间接气体输送,本文将不与走廊直接相邻的房间称为孤立房间,将与走廊相邻的房间称为走廊房间。为了满足前文建立的算法运行条件,本文将风管穿越房间和孤立房间进行捆绑送风,即将一个走廊房间与若干个与该房间相邻的孤立房间视为一个房间进行风口连接,本文将此过程称为房间的合并。

6.2.1 孤立房间的代表节点生成

当房间合并时,由于孤立房间的边长通常与走廊房间的边长存在差异,风口节点的布置存在错位,将两个相邻房间视为同一房间后直接进行风口的最优连接方案计算会导致风口划分时产生过多行和列,且每一行和每一列节点都存在大量

虚拟节点（如图 6.1a），直接对形成的点集进行计算会造成快速水力计算过程运行缓慢，导致计算时间过长。同时，由于两个房间之间具有共用墙等建筑结构，直接对两个房间的所有风口进行连接方案计算容易导致风管布置在墙线上（如图 6.1b 中红色线段表示的管道与墙线重叠），导致设计方案无法施工。

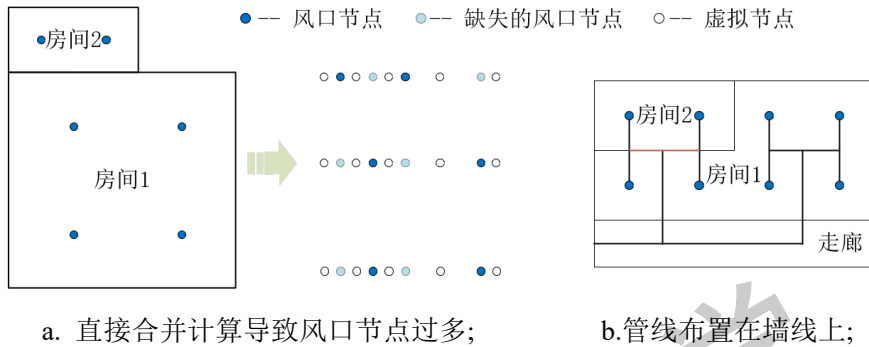
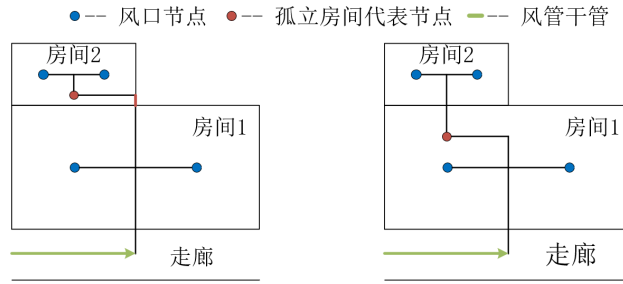


图 6.1 不能直接合并两个房间计算的原因示意图;

为了避免计算量过大以及保障设计方案的合理性，本文用一个房间代表节点表示孤立房间的风管连接方案特性来进行房间合并，此时，等效于给走廊房间添加了一个节点，相对于直接进行两个房间的节点连接，此方法可以显著降低节点的数量，提高运算效率。房间代表节点是能体现房间内风管连接方案水力稳定性、管道耗材和输配阻力的一个节点，同时，为了保证管道合并后的连贯性，该节点必须为孤立房间风管连接方案中的某个节点。

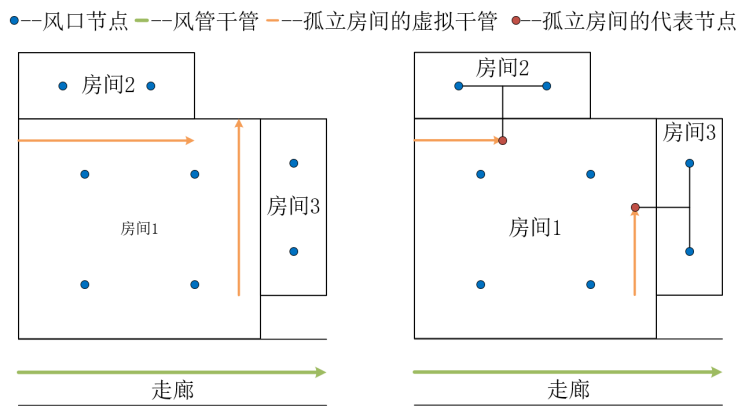
对于孤立房间而言，其最优连接方案中的节点可以在孤立房间内，也可以在走廊房间内。如图 6.2a 所示，当孤立房间的代表节点在孤立房间内时，将孤立节点添加到走廊房间内进行计算时，最优的布置方案可能与墙线重合（图中的红色线段表示的管道），因此，孤立房间的代表节点位置应该处于走廊房间内。当孤立房间代表节点处于走廊房间的风口点集内部时，孤立房间代表节点连接到孤立房间的管道会与走廊房间内的风管发生碰撞，为了保证管道连接的合理性，孤立房间代表节点在走廊房间的位置必然处于走廊房间的风口点集的外部。因此，孤立房间代表节点的位置有两种布置方式：在走廊房间最外围的风口行或者列上；或者单独布置成行或者列。孤立房间代表节点布置在走廊房间最外围的风口行或者列上时，某些案例可能发生孤立房间代表节点与走廊房间内的风口位置重叠的情况，此时，一个节点无法同时表示两个节点的特性，导致算法失效，如图 6.2b 所示，房间 2 为孤立房间，其最优连接方案上的节点若在房间 1 中的风口行时，风口节点与孤立房间的代表节点发生重合，导致无法计算房间 1 的最优连接方案，而孤立房间的代表节点单独成行或者列时，不会产生上述情况。



a. 代表节点在孤立房间内部; b. 代表节点在走廊房间内;
图 6.2 孤立房间代表节点放置位置讨论

孤立房间的面积通常不会太大，一根风管足够承担其所需的风量；走廊房间向孤立房间进行气体输送需要经过一面共用墙，只采用一根管道穿墙的设计方案可以减少工程量；同时，孤立房间代表节点的个数越少，走廊房间内添加的节点越少，风管连接方案计算速度越快，故本文规定孤立房间的代表节点数量为 1。

由于计算孤立房间的风口最优连接方案也需要走廊干管作为输入，因此需要在计算孤立房间前为其确定一条虚拟的走廊干管。由于代表节点位于走廊房间内，同时在走廊房间的风口点集外部，因此，虚拟走廊干管可以布置在房间共用墙和风口点集之间，如图 6.3a 所示。为了避免孤立房间的风管连接方案与墙线重叠，虚拟走廊干管的长度为共用墙的长度。为了保证孤立房间与走廊房间的风管连接方案合并后为全局最优，虚拟走廊干管的方向需要与走廊干管气体输配的方向对应。当虚拟干管与走廊干管平行时，其方向与走廊干管方向相同；当虚拟干管与走廊干管不平行时，气体从靠近走廊干管一端输送至较远端。



a. 孤立房间虚拟干管的位置和方向; b. 孤立房间代表节点的生成
图 6.3 孤立房间的虚拟干管和代表节点

当确定虚拟走廊干管的位置和方向后，采用风管连接最优方案生成方法计算孤立房间的最优风口连接方案，孤立房间的代表节点为最优连接方案中的干管与支管的连接节点（如图 6.3b 所示）。

6.2.2 房间合并

孤立房间和走廊房间的合并过程如图 6.4 所示，确定孤立房间的虚拟主管位置和方向后，计算孤立房间的最优风口连接方案。对每个方案确定其对应的代表节点后，将代表节点以风口属性添加到走廊房间中，对每个添加后的节点集进行最优连接方案计算，得到合并房间的最优布置方案。

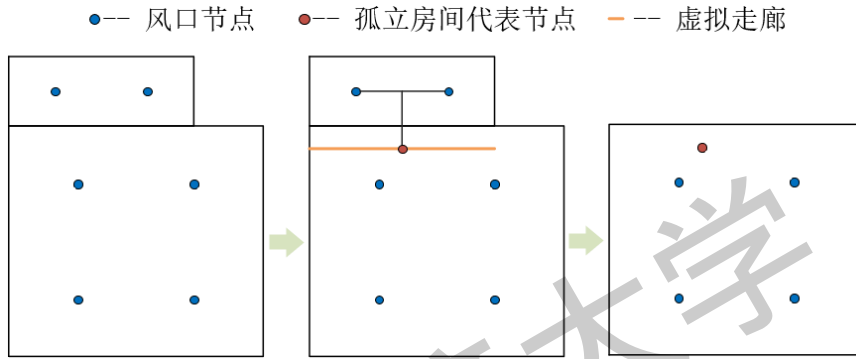
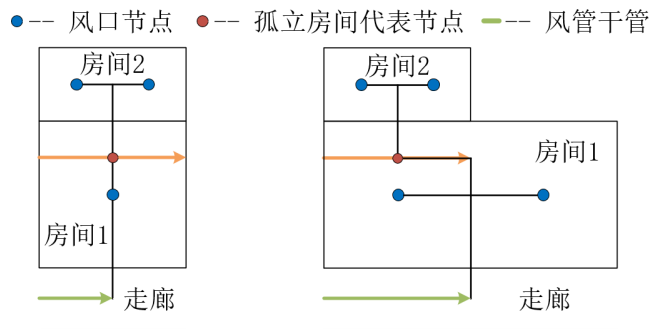


图 6.4 房间组合时的风口连接过程

由于孤立房间的代表节点是从虚拟走廊主管向孤立房间输配的连接节点，同时孤立房间的代表节点个数设置为 1，因此该节点的类型属性是弯头，其水力特性也是通过弯头的计算公式计算得到。但当孤立房间的代表节点添加到走廊房间后，由于代表节点视为额外添加的风口，房间合并后，孤立房间代表节点会产生两种连接结果，如图 6.5 所示，图 6.5a 表示孤立房间的代表节点在房间合并后的连接方案中是虚拟连接节点；图 6.5b 表示孤立房间的代表节点在房间合并后的连接方案中是弯头。



a. 孤立房间代表节点连接后作为弯头； b. 孤立房间代表节点连接后作为弯头；

图 6.5 孤立房间代表节点实际表示的局部阻力构件

由于孤立房间的代表节点在实际连接中可能是弯头或者虚拟连接点，其相对应的节点参数也会根据其表示的节点类型而有所改变。代表节点需要表示该节点

所连接的管道的水力特性参数，具体包含风量、不平衡率、管道表面积和管网的输配阻力。由于代表节点作为弯头或者虚拟节点连接只会影响其阻力的计算，故该节点的其他参数计算公式如式 6.1 - 6.4 所示：

$$Rnode_{\text{孤立}}.pos = Node.pos \quad (6.1)$$

$$Rnode_{\text{孤立}}.F = \sum Snode.F \quad (6.2)$$

$$Rnode_{\text{孤立}}.UB = \sum Snode.UB \quad (6.3)$$

$$Rnode_{\text{孤立}}.SA = \sum Sedge.SA \quad (6.4)$$

其中， $Rnode_{\text{孤立}}$ 表示孤立房间的代表节点，孤立房间连接方案中与其对应的节点为 $Node$ ， $Rnode_{\text{孤立}}$ 的特性与 $Node$ 相关； pos 为节点的位置坐标； F 为风量； UB 为不平衡率； SA 为管道的表面积； $Snode$ 表示 $Node$ 所连接的所有子节点； $Sedge$ 表示 $Node$ 与其子节点以及其子节点之间的所有连接边。

当孤立房间的代表节点作为虚拟连接节点时，该节点不是局部阻力构件，因此不具备局部阻力；当孤立房间的代表节点作为弯头时，该节点的阻力值包含 $Node$ 的局部阻力。因此， $Rnode_{\text{孤立}}$ 所连接管道的输配阻力计算公式为：

$$Rnode_{\text{孤立}}.R = \begin{cases} Sedge_{\text{connect}}.FR + \sum Snode.R, & Rnode_{\text{孤立}} \text{ 是虚拟连接点} \\ Node.R + \sum Snode.R, & Rnode_{\text{孤立}} \text{ 是弯头连接点} \end{cases} \quad (6.5)$$

其中， R 表示对应节点的阻力值，包含与该节点直接相连的子边 $Sedge_{\text{connect}}$ 的沿程阻力和局部阻力； FR 表示沿程阻力。

6.3 空调系统干管连接

当完成每个房间最优连接方案的计算后，需要确定走廊干管与走廊房间内支管的最优连接方案，即确定空调系统的干管连接。空调系统通常具有多个走廊房间，每个房间的内部风管连接方案比较复杂，在计算空调系统干管连接的过程中，采用代表节点表示房间内的风口连接方案可以简化寻优过程。生成走廊房间的代表节点后，在走廊中心线区域内求解最优的风管干管布置方案，完成干管布置。由于干管布置过程没有考虑实际施工，布置方案中可能存在局部阻力构件距离过近，故需要再进行局部阻力构件的位置调整。将调整后的风管干管连接方案与房间风口连接方案进行结合，即可得到空调风系统的最优风管布置方案。

6.3.1 走廊房间的代表节点生成

由于走廊房间内的风管连接方案通常具有多个节点和多条边，直接将所有房间内的风管与走廊干管进行连接后进行水力计算求取最优设计方案会导致计算量过大，因此，与房间组合过程相似，本文采用代表节点来表示合并房间的风管水力特性。对减少了节点数量的干管连接方案进行水力计算的计算速度更快，同时，直接使用房间内风管最优连接方案的计算结果，可以避免房间内部管网水力特性的重复计算，极大地减少了计算耗时。

走廊房间的代表节点与孤立房间代表节点的选取方式不同。走廊房间代表节点不能位于走廊干管上，因为风管系统的干管会出现一个节点输配走廊两边房间的情况（三通或四通），此时采用一个连接点无法同时表征两个走廊房间的水力特性，如图 6.6 所示，房间 1 与房间 2 的风口布置和风量需求都相同，且处于同一走廊的两侧，此时，单个房间的风管连接最优方案一致，风管干管向两个房间的接入点也相同。若采用孤立房间代表节点的选取方式，一个房间接入点需要同时表征房间 1 和房间 2 的风管水力特性，与快速水力计算模型中关于节点属性的规定不符。同时，若走廊房间的代表节点位于干管上，在干管与房间代表节点的连接结果中，房间接入点处于直线连接，此时无法判断走廊房间代表节点对应的局部阻力构件的类型，导致无法计算走廊房间代表节点的局部阻力特性。

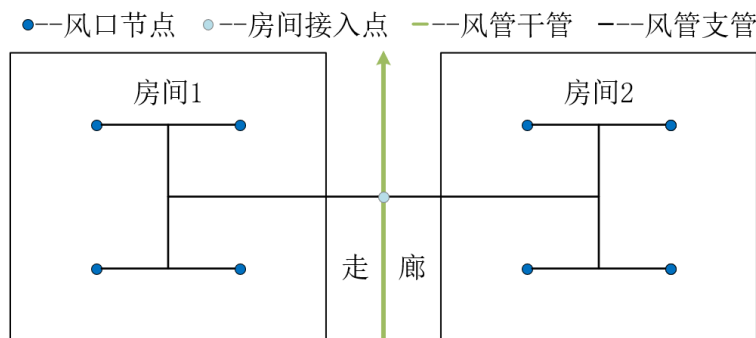


图 6.6 走廊房间代表节点与孤立房间代表节点选取方式不同的原因

为了能直接采用快速水力计算模型计算干管与走廊房间代表节点的连接方案的水力特性，本文将房间接入点的子节点作为走廊房间的代表节点，如图 6.7 所示。一个走廊房间可能有多个代表节点，同时，代表节点可以是风口节点也可以为虚拟节点。

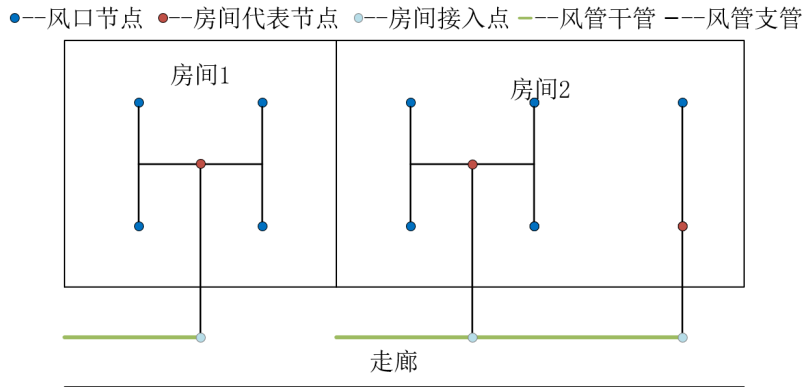


图 6.7 走廊房间的代表节点

走廊房间代表节点的属性包括：该节点后（含该节点）的所有风量，该节点后的所有节点的不平衡率，该节点后的所有输配阻力，以及该节点后所连接的管道表面积。具体的节点属性计算公式如公式 6.6–6.10 所示：

$$Rnode_{走廊} \cdot pos = Node \cdot pos \quad (6.6)$$

$$Rnode_{走廊} \cdot F = Node \cdot F + \sum Snode \cdot F \quad (6.7)$$

$$Rnode_{走廊} \cdot UB = Node \cdot UB + \sum Snode \cdot UB \quad (6.8)$$

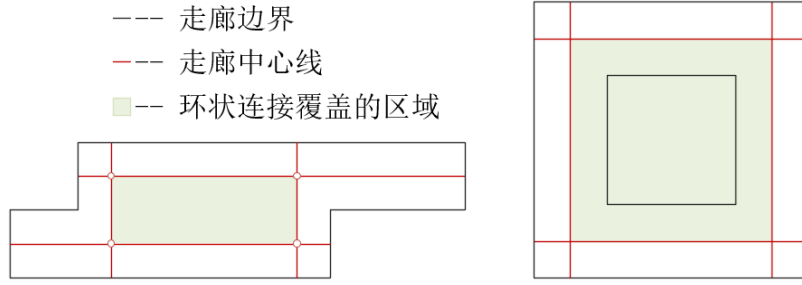
$$Rnode_{走廊} \cdot R = Node \cdot R + \sum Snode \cdot R \quad (6.9)$$

$$Rnode_{走廊} \cdot SA = \sum Sedge \cdot SA \quad (6.10)$$

其中， $Rnode_{走廊}$ 表示走廊房间的代表节点，走廊房间连接方案中与其对应的节点为 $Node$ ， $Rnode_{走廊}$ 的特性与 $Node$ 相关； pos 为节点的位置坐标； F 为风量； UB 为不平衡率； R 为输配阻力； SA 为管道表面积。 $Snode$ 表示 $Node$ 节点后的所有子节点； $Sedge$ 表示 $Node$ 节点与其子节点以及所有子节点之间相连接的边。

6.3.2 风管干管布置

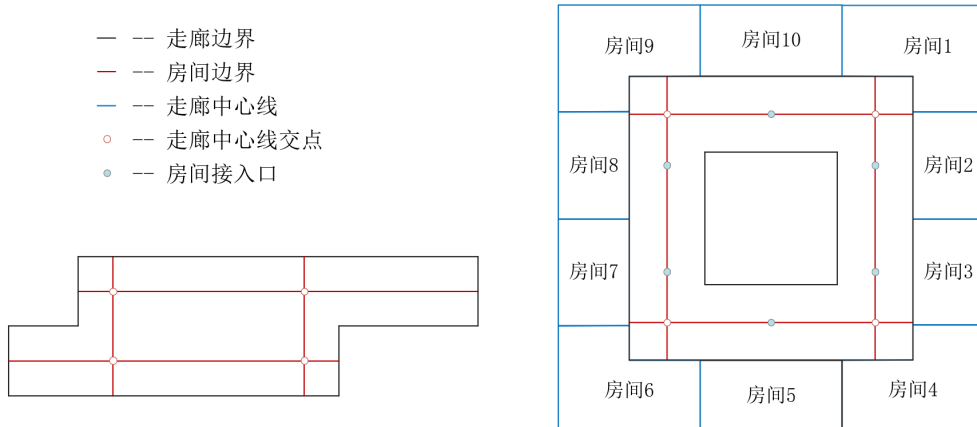
走廊房间代表节点确定后，需要将其连接到走廊干管中并确定风管干管的连接方案。风管干管的布置以 2.2 节提取的走廊中心线拓扑为求解域，以所有走廊房间的代表节点作为连接目标，采用快速水力计算模型计算最优的管道连接方案。由于走廊中心线的提取结果可能存在环状连接，而风管的连接结果应为枝状连接，不能存在环路，故风管干管的布置需要进行走廊中心线拓扑的破坏过程。走廊中心线的连接结果可能存在两类环状连接，如图 6.8 所示，图 6.8a 表示内部环状连接，指环状连接形成的区域都在走廊几何内部；图 6.8b 为外部环状连接，环状连接形成的区域中有部分不属于走廊区域。



a. 内部环状连接的走廊中心线; b. 外部环状连接的走廊中心线;

图 6.8 走廊中心线连接拓扑的环状连接分类

内部环状连接的走廊中心线一般产生于单条异形走廊内，该走廊无法采用一条风管直管段给所有走廊房间送风；外部环状连接则产生于多条联通走廊中。由于风管通常设置为枝状管网，故需要在确定连接方案前进行走廊中心线连接拓扑的破坏处理。破坏的基本原则是删除多余的边，但由于内部环状连接和外部环状连接进行风量输送的性质不同，两种环路的破坏过程也有所区别。破除内部环状连接（内部破坏）的目的是确定一条走廊的管道联通方式，在提取环路拓扑（图 6.9a）后删除对应的边即可实现破坏目的，贯穿异形走廊的联通管道可以对所有房间进行气体输配。破除外部环状连接（外部破坏）的目的是确定环状走廊中干管的终点，以求取在环状走廊中的最优的枝状管网布置方式。与内部破坏不同，外部环状连接的每一条边所表示的干管需要给多个房间进行气体输送，直接删除某条边会导致部分房间无法连接到风管干管上，故外部破坏需要保证每个房间的气体输配。本文将位于外部环状连接中的房间接入点添加到环状连接中，对房间接入点和外部环状连接的交点形成的环状拓扑（图 6.9b）进行破坏处理，以保障每个房间都能被风管干管连接。



a. 走廊中心线的内部环状破环节点; b. 走廊中心线的外部环状破环节点;

图 6.9 两种走廊环路的破坏

对添加房间入口节点后形成的走廊中心线拓扑删除边的具体步骤如下：

(1) 提取连接方案中的环所包含的边。拓扑中的环路个数符合以下规则：

$$n_{loop} = n_{edge} - n_{node} + 1 \quad (6.11)$$

其中， n_{loop} 为连接拓扑中的环的个数； n_{edge} 为连接拓扑的边数； n_{node} 为连接拓扑的节点数。连接管路中的环路可通过深度优先搜索算法得到，当节点的子节点和已完成搜索的前几层的父节点相同时则为环路中的节点；

(2) 根据环路个数 n_{loop} 进行边的修剪，删除原有连接结果的 n_{loop} 条边。将所有环路中的边按每组 n_{loop} 条边进行组合，将原存在环路的连接拓扑中对应的边进行删除，对删除后的连接拓扑重新计算其环路个数，所有环路个数为 0 的方案都满足风管干管的布置需求。

由于走廊中心线的线段未必都以风管形式连接到房间中，因此，经过破坏处理的走廊中心线依然无法直接进行连接方案的水力计算，需要删除多余的连接线段。风管连接方案中的所有支管的连接末端都是风口，在风管干管的布置过程中则为房间代表节点。将破坏后的走廊中心线连接拓扑中末端不是房间代表节点的线段删除，则可得到对应情况下的风管干管布置方案。由于走廊中心线的破坏可产生多种连接结果，一个走廊中心线经过破坏和多余线段删除后可得到多个风管布置方案，最优的风管干管布置方案可采用第四章的快速水力计算模型计算得到。

6.3.3 局部阻力构件位置调整

进行房间代表节点与走廊主管连接方案的水力特性计算后，可以筛选得到最优的风管干管连接图。但是，在前文述及的连接走廊房间代表节点与走廊主管的过程中未考虑房间之间的风管接入节点的间距，会导致生成的连接方案中接入节点距离过近，反映到实际风管布置上则为局部阻力构件之间距离过近，当考虑各个构件之间的大小和安装间距后，可能导致生成的连接方案无法实际施工。因此，需要在生成风管干管的连接方案后，对局部阻力构件进行一定的调整。判断两个局部阻力构件间是否需要调整间距的依据是：两个局部阻力构件的距离是否大于 1.5 倍的连接风管宽度，当小于该值时认为在实际施工过程中由于局部阻力构件距离过近而无法准确地安装两个构件。

局部阻力构件距离过近的情况主要产生于走廊两侧房间之间以及走廊拐角的多个房间之间，调整局部阻力构件位置的主要思路是合并节点或者加大节点间距。由于合并两个局部阻力构件可以减少风管总体的局部构件个数，同时对连接方案的水力稳定性影响不大，故优先考虑合并两个接入节点。局部阻力构件之间

的合并不能改变房间的风管输入方向，只能对房间进口风管进行平移，同时两个节点可以组合生成另一个对应的局部阻力构件，上述条件可具体为判断依据：平移的风管与另一个节点的所有支管不重叠。当两个接入点满足合并条件后，根据节点的类型特性可以确定移动哪个节点进行合并。为了降低合并算法的计算量和提高其通用性，本文规定两个接入点的合并是某个节点向另一个节点合并，不考虑同时更改两个节点坐标再搜索最优合并方案的方式。当两个节点不符合合并条件时，改变其中一个节点的坐标，使两个节点之间的距离满足安装要求。

为了避免单个节点的调整反复影响到其相邻节点，导致算法对几个节点进行反复调整而陷入死循环，本文以机房节点为起点，按空气输送方向逐个接入点进行判断并修改位置，保证算法遍历的单向性。当两个接入节点需要合并时，默认将后面节点合并到前一个节点，以减少管材消耗，当后面节点由于自身特性无法移动时，再移动前一个节点进行合并。当风管干管连接图中的节点位置进行调整后，对应的房间连接方案中的其他节点也需要进行调整，将调整后的房间连接方案逐个连接到走廊主管连接图上，便可形成空调风系统的管道布置方案。具体的局部阻力构件的位置调整算法见表 6.1。

表 6.1 局部阻力构件的位置调整算法伪代码

输入：图 $G_{走廊} = (E_{走廊}, V_{走廊})$ ，经过水力计算的风管主管与房间代表节点的连接图
图 $G_{房间} = (E_{房间}, V_{房间})$ ，经过水力计算的最优的房间风管连接方案
输出：图 $G' = (E', V')$ ，用于施工的房间风口节点和走廊主管连接图，初始 $G' := G_{走廊}$
<pre> 1: for $\forall v_1, v_2 \in V_{走廊}$ do 2: $l_{v_1, v_2} :=$ 点 v_1 和 v_2 之间的距离; $w_{mainduct} :=$ 点 v_1 和 v_2 之间的风管主管宽度 3: if $l_{v_1, v_2} < 1.5 w_{mainduct}$ then 4: $Type_v :=$ 节点 v 的对应的局部阻力构件类型 5: $B_v :=$ 三通的旁通管或弯头中连接房间的管道的单位方向向量 6: $D_v :=$ 三通的直通管的单位方向向量 7: $v_{change}, v_{stay} :=$ 分别表示需要进行位置调整的节点和保持原来位置的节点 8: if v_2 in the corners of the corridor centerline then 9: $(v_{stay}, v_{change}) := (v_2, v_1)$ 10: else if $Type_{v_2} = \text{"Turn"}$ and $Type_{v_1} = \text{"Tee"}$ and $B_{v_1} // B_{v_2}$ and $B_{v_1} \neq B_{v_2}$ then 11: $(v_{stay}, v_{change}) := (v_2, v_1)$ 12: else then 13: $(v_{stay}, v_{change}) := (v_1, v_2)$ 14: end if 15: $Combine :=$ 调整后的两个节点是整合还是分离的参数，值为 True 或者 False 16: if $Type_{v_1} = \text{"Four way"}$ or $Type_{v_2} = \text{"Four way"}$ then 17: $Combine :=$ False </pre>

续表 6.2 局部阻力构件的位置调整算法伪代码

```

18:     else if  $Type_{v1} = \text{"Tee"}$  and  $Type_{v2} = \text{"Tee"}$  then
19:         if not ( $D_{v1} // D_{v2}$  and  $B_{v1} \neq B_{v2}$ ) then
20:              $Combine = \text{False}$ 
21:         else then
22:              $Combine := \text{True}$ 
23:         else then
24:              $Combine = \text{True}$ 
25:         end if
26:          $Gv_{change}, Gv_{stay} := v_{change}$  和  $v_{stay}$  对应的房间风管连接方案,  $Gv_{change}, Gv_{stay} \in G_{\text{房间}}$ 
27:         if  $Combine = \text{True}$  then
28:             Change the coordinates of  $v_{change}$  to  $v_{stay}$  in  $G'$ 
29:         else then
30:             Change the coordinates of  $v_{change}$  to a distance of  $1.5 W_{mainduct}$  from  $v_{stay}$ 
31:         end if
32:         Adjust the coordinates of the relevant nodes in  $Gv_{change}$ 
33:          $G' := G' + Gv_{change} + Gv_{stay}$ 
34:     else then
35:          $Gv_1, Gv_2 := v_1$  和  $v_2$  分别对应的房间风管连接方案,  $Gv_1, Gv_2 \in G_{\text{房间}}$ 
36:          $G' := G' + Gv_1 + Gv_2$ 
37:     end if
38: end for

```

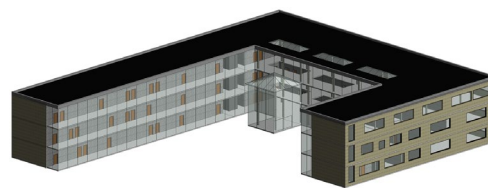
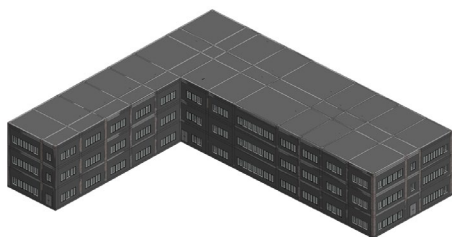
6.4 案例应用

本文提出的风管系统自动设计方法应用于四栋办公建筑中, 房间功能包含办公室、会议室、走廊、和设备用房, 其中还有部分房间为非空调房间。从建筑的 Revit 模型提取 IDF 文件, 进行风管自动设计后, 将设计结果返回 Revit 中查看管道布置的平面图和空间布置图, 本文采用提出的自动设计方法, 设计了四栋建筑在风机盘管系统下的新风管道以及在全空气系统时的送风管道。各个建筑标准层的实际设计结果如图 6.10 所示。

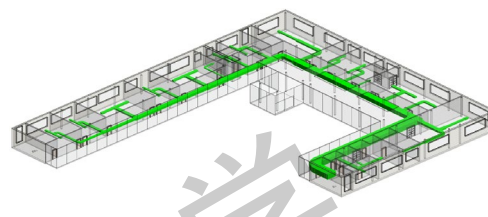
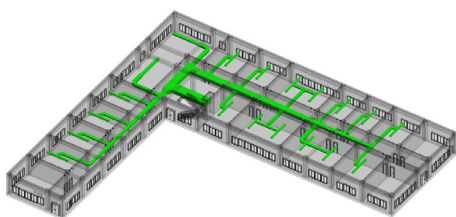
Case 1

Case 2

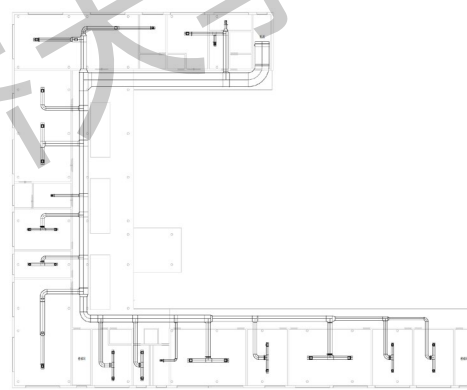
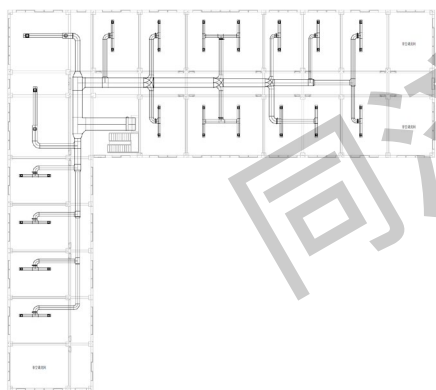
建筑
外形



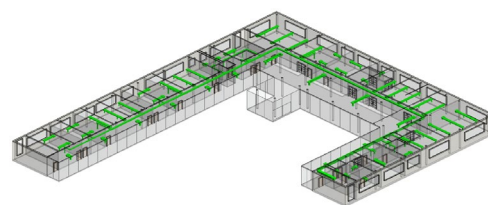
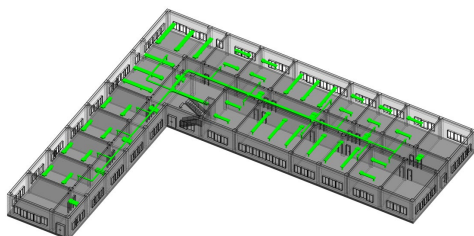
全空
气系
统送
风管
(3D)



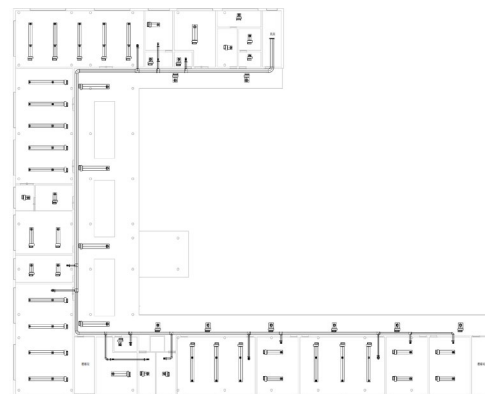
全空
气系
统送
风管
(2D)



风机
盘管
系统
风管
(3D)



风机
盘管
系统
风管
(2D)



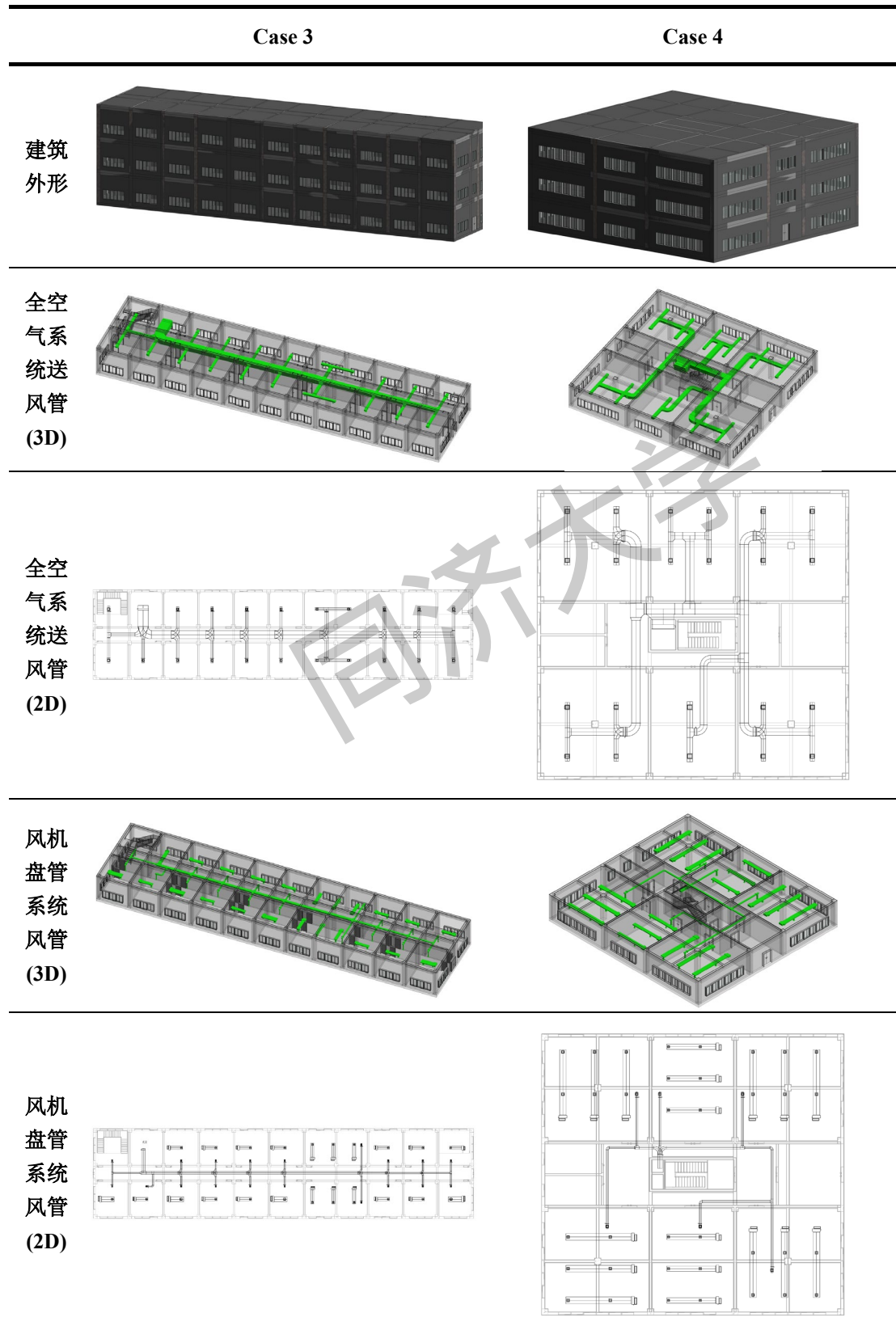


图 6.10 风管自动设计算法的案例应用

6.5 本章小结

本章通过实际案例对本文提出的空调系统风管自动布置方法进行验证，并详细介绍了将房间内风口连接拓扑遍历算法和快速水力计算模型应用于整个空调系统的风管自动布置的过程中需要完成的其他步骤。为了减少计算量，本文将房间内的风管和走廊处的干管分为两部分生成，以房间代表节点作为房间风管连接方案的表征节点，减少干管连接方案计算过程涉及的节点个数的同时实现走廊干管与房间支管的正确连接。生成的走廊主管连接方案再通过局部阻力构件调整算法对因间距过近而无法施工的部件进行调整，将调整后的各个房间风管连接方案添加到走廊干管连接方案中，完成整个风系统的管道生成。在明确风系统管道自动设计的所有流程后，采用 Python 软件对上述流程进行代码实现，并将其应用于四栋案例建筑的空调风系统设计。

第7章 结论与展望

7.1 结论与成果总结

本文提出了一种空调系统风管自动设计的新方法,首次实现了风系统设计的全自动化。与传统的人工设计相比,该方法可以通过对所有方案的精确计算筛选最优的设计方案,同时将设计时间缩短了数十倍。由于设计过程完全自动化,可以避免人为因素产生的错误,更能保证设计方案的质量。本文深入研究了风系统自动化设计过程存在的难题,具体为管道布置区域、风口连接算法、快速水力计算算法以及综合优化目标等相关内容,并获得了以下结论及成果:

- (1) 提出了风管网拓扑求解域的计算方法。基于风管干管布置区域的假设,本文提出了较为通用的建筑走廊中心线提取算法,该算法以边长为切割依据,将不规则多边形切割为多个两条边平行的四边形,继而进行中心线提取,文中同时详细介绍了单连通和复连通区域走廊的中心线提取过程。
- (2) 提出了风口连接算法。为了快速筛选出符合实际工程的风口连接方案,本文基于实际案例,建立了房间内风口连接的节点模型,并以字符串表示风口的连接关系,通过字符串的计算实现风口连接方案筛选,筛选后的方案再通过节点和边的避障方法,完成风管的障碍规避。该算法考虑了虚拟连接节点数量的不确定性和风口的特殊性,可以筛除超过 99.9 %的不合理方案,且在进行大量方案筛选时仍能保持较快的运算速度。
- (3) 建立了快速水力计算模型。基于风管长宽比为定值的假定,本文推导了风管的选型公式,在确定风管风量和设计风速后可快速进行风管选型。为了计算设计方案的水力特性,针对风口连接拓扑图,本文采用深度优先搜索算法建立了有向图的水力计算模型,确定了节点和边的属性以及计算公式。该模型可快速计算大量风口连接方案的水力特性,当风口连接方案的个数少于 5000 个时,快速水力计算过程总运行时长在 5 分钟以内。
- (4) 实现了实际建筑中的风管自动设计。基于风口连接算法和快速水力计算模型,在明确管道布置区域后,本文以房间代表节点来连接风管的干管和支管部分,同时提出了局部阻力构件的位置调整算法。文中详细介绍了算法实际应用过程中需要解决的问题以及对应的方案,通过 Python 软件代码实现了风系统自动设计的整个过程,并对四栋建筑的标准层进行了测试。

7.2 课题创新点

- (1) 本研究创新性地提出了一种风口连接算法，该算法可以快速筛选出满足工程需求的所有风口连接方案，同时兼顾障碍规避。与已有算法相比，该算法无需对计算区域进行网格划分，并通过虚拟节点的实际位置变动实现了风管局部阻力构件的灵活连接。该算法可筛除超过 99.9% 的不合理方案，且运算速度较快（计算时长的数量级为 0.1s）。
- (2) 针对风口连接拓扑图，本文提出了适用于有向图的水力计算模型，并详细介绍了每个迭代步长内的节点及边的相关属性的计算方法。该水力计算模型可以通过综合优化目标实现风管连接方案的水力稳定性、管道耗材和输配阻力等参数的综合评价，筛选出最优的设计方案。综合目标中各个因素的优先级可由暖通工程师根据设计目标自行定义。

7.3 局限性与展望

风系统管网的自动设计需要兼顾运算时间，在相较人工设计有绝对时间优势时才具备足够动力在行业中应用。本文提出的风管自动布置方法可以快速完成大部分建筑中的风管设计，但对于具有高大空间的建筑，当一个房间内风口个数过多时，其运行时长与内存消耗超过了工程可接受范围。同时，本课题主要研究散流器顶送形式的风系统的自动设计，还未研究其他末端形式的适配与调整方法。因此，本课题未来的研究方向也主要围绕上述两个问题展开：

- (1) 现有的风口连接算法筛选后剩余的方案较多，当风口连接算法筛选后的方案数小于 5000 时，快速水力计算过程运行时长为 5 分钟以内，因此可以考虑添加更严格的筛选规则，减少符合条件的方案。
- (2) 对于包含多个风口的大空间，可以尝试将其划分为多个小空间再进行风口连接。由于风口计算时间与风口数量成指数型关系，采用多个小空间分别计算后再整合的方法可以有效减少运算量。
- (3) 本文风管干管布置建立在走廊连通所有房间的基础上，对于大空间或者异形建筑，需要研究无走廊连通时的风管干管的布置区域提取算法。
- (4) 其他风口形式的管道通常由于末端的改变而具备特定的设计要求，故需要研究风口连接算法的适配方案，同时在设计过程中需要添加风管系统与其他系统的管路之间的碰撞和规避的相关研究。

参考文献

- [1] Pérez-Lombard L, Ortiz J, Pout C. A review on buildings energy consumption information[J]. *Energy and Buildings*, 2008, 40(3): 394–398.
- [2] IEA. Key World Energy Statistics[R]. Key World Energy Statistics, Paris: International Energy Agency, 2021.
- [3] EIA. International Energy Outlook[M]. 2017, 2021.
- [4] Zhao H X, Magoulès F. A review on the prediction of building energy consumption[J]. *Renewable and Sustainable Energy Reviews*, Elsevier Ltd, 2012, 16(6): 3586–3592.
- [5] Eddy Krygiel B N. BIM Green: Successful sustainable design with building information modeling[M]. Wiley Publishing, 2008, 37(4).
- [6] Gallaher M P, O'Connor A C, Dettbarn J L, 等. Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry. National Institute of Standards and Technology (NIST)[J]. National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, Tech. Rep., 2004: 04–867.
- [7] Lu Y, Wu Z, Chang R, 等. Building Information Modeling (BIM) for green buildings: A critical review and future directions[J]. *Automation in Construction*, Elsevier, 2017, 83(August): 134–148.
- [8] Suziyana M D, Nina S N, Yusof T M, 等. Analysis of heat gain in computer laboratory and excellent centre by using CLTD/CLF/SCL method[J]. *Procedia Engineering*, Elsevier B.V., 2013, 53: 655–664.
- [9] Trčka M, Hensen J L M. Overview of HVAC system simulation[J]. *Automation in Construction*, 2010, 19(2): 93–99.
- [10] Verbai Z, Kocsis I, Kalmár F. Outdoor dry bulb heating design temperatures for Hungary[J]. *Energy*, 2015, 93: 1404–1412.
- [11] Refrigerating and Air Conditioning Engineers, Inc. Handbook of Fundamentals[M]. Atlanta: ASHRAE, 2017.
- [12] Mao C L, Haberl J, Baltazar J C. Peak Heating/Cooling Load Design Methods: How We Got To Where We Are Today In The U.S.[J]. 2013.
- [13] Mao C L. Analysis of Building Peak Cooling Load Calculation Methods for Commercial Buildings in the United States[J]. 2016.
- [14] Mao C L, Baltazar J C, Haberl J S. Literature review of building peak cooling load methods in the United States[J]. *Science & Technology for the Built Environment*, 2017: 0–0.
- [15] Mao C L, Baltazar J-C, Haberl J S. Comparison of ASHRAE peak cooling load calculation methods[J]. *Science and Technology for the Built Environment*, Taylor & Francis, 2019, 25(2): 189–208.
- [16] Nadeem T B, Ahmed A, Naqvi A A, 等. Designing of Heating, Ventilation, and Air Conditioning (HVAC) System for Workshop Building in Hot and Humid Climatic Zone Using

- CLTD Method and HAP Analysis: A Comparison[J]. *Arabian Journal for Science and Engineering*, 2022.
- [17] 朱丹丹, 燕达, 王闯, 等. 建筑能耗模拟软件对比: DeST、EnergyPlus and DOE-2[J]. *建筑科学*, 2012(S2 vo 28): 213–222.
- [18] 周欣, 燕达, 洪天真, 等. 建筑能耗模拟软件空调系统模拟对比研究[J]. *暖通空调*, 2014, 44(04): 113-122+131.
- [19] Kang Y Z, Augenbroe G, Li Q, 等. Effects of scenario uncertainty on chiller sizing method[EB/OL]. /2022-03-17. <https://www.sciencedirect.com/science/article/pii/S1359431116322360>.
- [20] Lee W L, Lee S H. Developing a simplified model for evaluating chiller-system configurations[J]. *Applied Energy*, 2007, 84(3): 290–306.
- [21] Picard D, Helsen L. Economic optimal HVAC design for hybrid GEOTABS buildings and CO2 emissions analysis[J]. *Energies*, 2018, 11: 314.
- [22] Huang P, Huang G S, Wang Y. HVAC system design under peak load prediction uncertainty using multiple-criterion decision making technique[J]. *Energy and Buildings*, 2015, 91: 26–36.
- [23] Sun Y M, Gu L, Wu C F J, 等. Exploring HVAC system sizing under uncertainty[EB/OL]. /2022-03-17. <https://www.sciencedirect.com/science/article/pii/S0378778814005088>.
- [24] Kennington J, Monaghan P F. COMBINE: The HVAC-design prototype[J]. *Building and Environment*, 1993, 28(4).
- [25] Amor R, Hosking J, Donn M. Integrating design tools for total building evaluation[J]. *Building and Environment*, 1993, 28(4).
- [26] Clarke J A, Maver T W. Advanced design tools for energy conscious building design: Development and dissemination[J]. *Building and Environment*, 1991, 26(1).
- [27] Rousseau P G, Mathews E H. Needs and trends in integrated building and HVAC thermal design tools[J]. *Building and Environment*, 1993, 28(4): 439–452.
- [28] Clarke J A, Maver T W. Advanced design tools for energy conscious building design: Development and dissemination[J]. *Building and Environment*, 1991, 26(1): 25–34.
- [29] Ellis M W, Mathews E H. Needs and trends in building and HVAC system design tools[J]. *Building and Environment*, 2002, 37(5): 461–470.
- [30] Donnelly J, Flynn J, Monaghan P F. Integration of energy simulation & ventilation design tools via an object oriented data model[J]. *Renewable Energy*, 1994, 5(5): 1190–1192.
- [31] Al-Homoud M S. Optimum thermal design of air-conditioned residential buildings[J]. *Building and Environment*, 1997, 32(3): 203–210.
- [32] Alireza, Golabchi, Manu, 等. Automated building information modeling for fault detection and diagnostics in commercial HVAC systems[J]. *Facilities*, 1983.
- [33] Choi M, Cho S, Lim J, 等. Design framework for variable refrigerant flow systems with enhancement of interoperability between BIM and energy simulation[J]. *Journal of Mechanical Science and Technology*, 2018, 32(12): 6009–6019.
- [34] Sha H J, Xu P, Yang Z W. IFC based semi-automated design tool for HVAC central system: A

- general framework[J]. IOP Conference Series: Earth and Environmental Science, 2019, 238: 012074.
- [35] Wang H X, Xu P, Sha H J, 等. BIM-based automated design for HVAC system of office buildings—An experimental study[J]. Building Simulation, 2022, 15(7): 1177–1192.
- [36] Lee C Y. An Algorithm for Path Connections and Its Applications[J]. IRE Transactions on Electronic Computers, 1961, EC-10(3): 346–365.
- [37] Jian S K, Hong E Z. Efficient routing algorithm[J]. Computer-Aided Design, 1987, 19(7): 375–379.
- [38] William R P. Development of a Three-dimensional Pipe Routing Algorithm[D]. Lehigh University, 1975.
- [39] Mitsuta T, Kobayashi Y, Wada Y, 等. A knowledge-Based Approach To Routing Problems In Industrial Plant Design[A]. France: 1987: 237–256.
- [40] Hightower D W. A solution to line routing problems on the continuous plane[A]. Papers on Twenty-five years of electronic design automation[C]. New York, NY, USA: Association for Computing Machinery, 1988: 11–34.
- [41] Traub H S, Köster M, Holtkötter T, 等. Conceptual plant layout[J]. Computers & Chemical Engineering, 1998, 22(3): s499–s504.
- [42] Dijkstra E W. A note on two problems in connexion with graphs[J]. Numerische Mathematik, 1959, 1(1): 269–271.
- [43] Reginaldo G. Optimization of process plant layout[D]. The University of Wisconsin - Madison, 1993.
- [44] Wangdahl G E, Pollock S M, Woodward J B. Minimum-Trajectory Pipe Routing[J]. Journal of Ship Research, 1974, 18(1): 46–49.
- [45] Nicholson T. Finding the Shortest Route between Two Points in a Network[J]. The Computer Journal. Section A / Section B, 1966, 9(3): 275–280.
- [46] Zhu D J, Latombe J C. Pipe routing-path planning (with many constraints)[A]. 1991, 3: 1940–1947.
- [47] Ito T. Genetic algorithm approach to piping route path planning[J]. Journal of Intelligent Manufacturing, 1999, 10(1): 103–114.
- [48] Holland J H. Adaptation in natural and artificial systems[M]. Ann Arbor: University of Michigan Press, 1975.
- [49] Newell R G. An Interactive Approach to Pipe Routing in Process Plants.[A]. Ljubljana, Yugoslavia: North-Holland Publishing Co., 1971, 2: 1080–1085.
- [50] Park J-H, Storch R L. Pipe-routing algorithm development: case study of a ship engine room design[J]. Expert Systems with Applications, 2002, 23(3): 299–309.
- [51] 樊江, 马枚, 杨晓光. 航空发动机外部管路自动敷设研究[J]. 机械设计, 2003, 20(7): 21–23.
- [52] Kennedy J, Eberhart R. Particle swarm optimization[A]. Proceedings of ICNN'95 - International Conference on Neural Networks[C]. Australia: IEEE, 1995, 4: 1942–1948.
- [53] Dorigo M, Maniezzo V, Colomi A. Positive Feedback as a Search Strategy[J]. Tech rep., 91-016, Dip Elettronica, Politecnico di Milano, Italy, 1999.

- [54] 冯茜, 李擎, 全威, 等. 多目标粒子群优化算法研究综述[J]. 工程科学学报, 2021, 43(06): 745–753.
- [55] Asmara A, Nienhuis U. Automatic piping system in ship[J]. Proceedings of the 5th International Conference on Computer and IT Applications in the Maritime Industries, 2006: 269–280.
- [56] Hakimi S. Steiner's Problem in Graphs and Its Implications[J]. Networks, 1971, 1(2): 113–133.
- [57] Hanna M. On steiner's problem with rectilinear distance[J]. Siam Journal on Applied Mathematics, 1966, 14(2): 255–265.
- [58] Liu Q, Wang C. A discrete particle swarm optimization algorithm for rectilinear branch pipe routing[J]. Assembly Automation, Emerald Group Publishing Limited, 2011, 31(4): 363–368.
- [59] Liu Q, Wang C G. Multi-terminal pipe routing by Steiner minimal tree and particle swarm optimisation[J]. Enterprise Information Systems, Taylor & Francis, 2012, 6(3): 315–327.
- [60] 乔东平, 裴杰, 肖艳秋, 等. 蚁群算法及其应用综述[J]. 软件导刊, 2017, 16(12): 217–221.
- [61] Hu Y, Jing T, Feng Z, 等. ACO-Steiner: Ant Colony Optimization Based Rectilinear Steiner Minimal Tree Algorithm[J]. Journal of Computer Science and Technology, 2006, 21(1): 147–152.
- [62] Luyet L, Varone S, Zufferey N. An Ant Algorithm for the Steiner Tree Problem in Graphs[A]. M. Giacobini. Applications of Evolutionary Computing[C]. Berlin, Heidelberg: Springer, 2007, 4448: 42–51.
- [63] 邬君, 林焰, 纪卓尚, 等. 基于协同进化的舰船分支管路系统路径优化研究[J]. 船海工程, 2008(04): 135–138.
- [64] Jiang W Y, Lin Y, Chen M, 等. An ant colony optimization-genetic algorithm approach for ship pipe route design[J]. International Shipbuilding Progress, 2014, 61(3): 163–183.
- [65] Jiang W Y, Lin Y, Chen M, 等. A co-evolutionary improved multi-ant colony optimization for ship multiple and branch pipe route design[J]. Ocean Engineering, 2015, 102(1): 63–70.
- [66] Wu L, Tian X, Wang H Y, 等. Improved ant colony optimization algorithm and its application to solve pipe routing design[J]. Assembly Automation, 2019, 39(1): 45–57.
- [67] Liu Q, Tang Z, Liu H, 等. Integrated Optimization of Pipe Routing and Clamp Layout for Aeroengine Using Improved MOALO[J]. International Journal of Aerospace Engineering, London: Hindawi Ltd, 2021, 2021: 6681322.
- [68] 常洪江. 遗传算法综述[J]. 电脑学习, 2010(03): 115–116.
- [69] Ren T, Zhu Z L, Dimirovski G, 等. A new pipe routing method for aero-engines based on genetic algorithm[J]. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 2013, 228(2): 424–434.
- [70] Sui H, Niu W. Branch-pipe-routing approach for ships using improved genetic algorithm[J]. Frontiers of Mechanical Engineering, 2016, 11(3): 316–323.
- [71] Niu W T, Sui H T, Niu Y X, 等. Ship Pipe Routing Design Using NSGA-II and Coevolutionary Algorithm[J]. Mathematical Problems in Engineering, London: Hindawi Ltd,

- 2016, 2016: 7912863.
- [72] Dong Z R, Bian X Y. Ship Pipe Route Design Using Improved A* Algorithm and Genetic Algorithm[J]. Ieee Access, Piscataway: Ieee-Inst Electrical Electronics Engineers Inc, 2020, 8: 153273–153296.
- [73] Brahme R, Mahdavi A, Lam K P, 等. Complex building performance analysis in early stages of design: a solution based on differential modelling, homology-based mapping, and generative design agents.[A]. Building Simulation, 2001, 2: 661–668.
- [74] Medjdoub B, Richens P, Barnard N. Generation of variational standard plant room solutions[J]. Automation in Construction, 2003, 12(2): 155–166.
- [75] Brès A, Judex F, Suter G, 等. A Method for Automated Generation of HVAC Distribution Subsystems for Building Performance Simulation[A]. San Francisco: International Building Performance Simulation Association, 2017: 1833–1842.
- [76] Medjdoub B, Bi G. Parametric-based distribution duct routing generation using constraint-based design approach[J]. Automation in Construction, 2018, 90: 104–116.
- [77] Carpaneto G, Dell’Amico M, Toth P. Algorithm 750: CDT: a subroutine for the exact solution of large-scale, asymmetric traveling salesman problems[J]. ACM Transactions on Mathematical Software, 1995, 21(4): 410–415.
- [78] Stanczak M, Pralet C, Vidal V, 等. Optimal Pipe Routing Techniques in an Obstacle-Free 3D Space[M]. 2020: 79.
- [79] 付祥钊. 流体输配管网[M]. 第4版. 北京: 中国建筑工业出版社, 2018.
- [80] 邱勇云, 邱相武, 赵志安. PKPM 智能空调水力计算软件开发[J]. 全国暖通空调制冷2010年学术年会论文集, 中国浙江杭州: 中国建筑学会暖通空调分会、中国制冷学会空调热泵专业委员会, 2010: 256.
- [81] 天正暖通软件[J]. 建设科技, 2012(Z1): 65.
- [82] 尚辰超. BIMSpace 在机电工程中的应用体会[J]. 安装, 2015(07): 20–21.
- [83] Idelchik I E. Handbook of Hydraulic Resistance[M]. R Malyavskaya G., 译. .
- [84] CIBSE. CIBSE Guide C: Reference Data (2007)[M]. London: The Chartered Institution of Building Services Engineers, 2007.
- [85] 上海市工业设备安装公司. 全国通用通风管道配件图表[M]. 北京: 中国建筑工业出版社, 1979.
- [86] 陆耀庆. 实用供热空调设计手册[M]. 第2版. 北京: 中国建筑工业出版社, 2008.
- [87] Miller D S. Internal flow : a guide to losses in pipe and duct systems[M]. Bedford: British Hydromechanics Research Association, 1971.
- [88] 周谟仁. 流体力学 泵与风机[M]. 第3版. 北京: 中国建筑工业出版社, 1994.
- [89] 秦慧敏. 关于通风管三通的局部阻力系数问题[J]. 建筑技术通讯(暖通空调), 1980(03): 10–13.
- [90] 王秋润. 空调风系统管网仿真调平方法[D]. 上海: 同济大学, 2019.
- [91] Mylaram N K, Idem S. Pressure Loss Coefficient Measurements of Two Close-Coupled HVAC Elbows[J]. HVAC&R Research, 2005, 11(1): 133–146.
- [92] Atkin S M, Shao L. Effect on pressure loss of separation and orientation of closely coupled HVAC duct fittings[J]. Building Services Engineering Research and Technology, 2000, 21(3):

- 175–178.
- [93] Sami S, Cui J. Numerical Study of Pressure Losses in Close-Coupled Fittings[J]. *Hvac&R Research*, 2004, 10(4): 539–552.
- [94] Lakshmiraju M, Cui J. Laminar Pressure Loss Coefficient in Close Coupled Fittings[A]. Chicago: Fluids Engineering, 2006: 713–719.
- [95] Ai Z T, Mak C M. Pressure Losses across Multiple Fittings in Ventilation Ducts[J]. *The Scientific World Journal*, 2013, 2013(3): 195763.
- [96] K. Sleiti A, Saleh M, Idem S. Detailed velocity profiles in close-coupled elbows - measurements and CFD predictions (RP-1682)[J]. *Science and Technology for the Built Environment*, 2017, 23(1): 1212–1223.
- [97] Salehi M, Idem S, K. Sleiti A. Experimental determination and CFD predictions of pressure loss in close-coupled elbows (RP-1682)[J]. *Science and Technology for the Built Environment*, 2017, 23(7): 1132–1141.
- [98] Salehi M, K. Sleiti A, Idem S. Study to identify CFD models for use in determining HVAC duct fitting loss coefficients[J]. *Science and Technology for the Built Environment*, 2017, 23(1): 181–191.
- [99] 张勇. 平面多边形直骨架及软件实现算法[J]. *计算机光盘软件与应用*, 2012(04): 165–166.
- [100] 陈喆. 暖通空调系统自动化设计方法研究[D]. 上海: 同济大学, 2020.
- [101] 樊振和. 建筑构造原理与设计[M]. 第4版. 天津: 天津大学出版社, 2011.

致谢

书至此处，意味着硕士三年已结束，抽盲与查重都幸运通过，到了写下离别话语的时候。从未想过硕士阶段会结束得如此匆忙与遗憾，受疫情影响，和很多朋友都来不及告别，情绪也比较低沉，对好多出乎意料的事感到些许失望。但是，最近总能想起复试那天被录取时的狂喜，考研两年，一切都值得！

进入许门大概是这几年最幸运的一件事。许老师温文儒雅，思维敏捷，对专业前沿总保持着极大的兴趣与热忱，脚踏实地地坚持自己的学术追求，对我的生活习惯与学术观念产生极大影响，每周的头脑风暴与论文分享更是不断地拓宽我的认知。许老师日常相处非常随和，对我们生活上也关心体恤，亦师亦友。我时长窃喜能成为许老师的学生，非常非常感谢许老师这三年的照顾！希望多年后我也可以成为与您一样优秀的学者。真诚地感谢各位师兄师姐师弟师妹和同门，与你们的快乐玩耍绝对是这几年来灿烂的回忆！非常感谢大宝师兄将我“捞”进课题组并带我开始学术研究，也非常感谢将我带入羽毛球圈和篮球圈，让我认识了不少好朋友，非常遗憾一直没能完成约定的球赛。陈喆师兄是我进课题组时最初交流的人，平静随和，热心助人，总能拥有最好用的小工具，总能给我们带来新颖的知识点，感谢师兄在我申请博士过程和课题研究过程的帮助！智博师兄思维敏锐，聪敏好学，时常可以在课题组一起谈天说地，感谢师兄给我取的“森教练”外号。在课题组熬夜时总有泓名师兄修机器人作陪，感谢师兄与我一起畅想未来，也很遗憾今年没法完成在学校钓一条鱼的约定。坤坤师兄在大宝毕业后是唯一的篮球伙伴，单挑多场，上一年羡慕你的快活，现在我也将要离开。鑫鑫师兄总是默默地处理了许多事，当着项目协调者，非常感谢为我们承担了诸多杂事！感谢沙师姐给我们带来许多古早轶事，帮我们普及了很多人际知识，给我们枯燥的生活添加了许多乐趣。顾师姐活泼有趣，给我们带来了许多游戏与欢乐，一起毕业快乐！重庆妹子罗为师姐大方开朗，热情外向，给我们带来了许多欢乐，也紧密着课题组的关系。非常感谢肖桐老乡不时的关心，一定要保持开心，憧憬的五月线下见大概又黄了，希望还有机会一起打球。感谢活泼小郭在重要事件上的提醒与分享，平时闲聊度过了不少烦闷日子，作为课题组的两个大胃王，咱去香港再吃！感谢 Carry 贴心的答辩资料分享，谢谢平时出游时和肖桐一起承担着拍照重任，照片和视频都做得非常棒！谢谢小贾日常的关心，永远能记得你准点发来的生日歌，我们真的要走了，你别哭。感谢关航帮我处理繁忙杂乱的夭大财务，我

们终于熬过来了！感谢张亚珂师妹帮我修改大小论文以及画了非常好看的图（非彩虹屁），非常感谢日常的论文进度监督，终于不用再算作业数了。感谢夏壮师妹接过了我的烫手代码，看着你无奈的样子我很抱歉，非常感谢帮我减少任务！感谢齐梓轩师弟陪我熬夜联动改代码，请多多保重身体！感谢李达忍受着我的每周极限操作，谢谢你默默承担着自动设计的最终压力！非常感谢朱博对我的未来分析，鞭策我写小论文，在图书馆的一大段时间里陪我解闷聊天。很遗憾，毕业前见不上，最后一次大餐也无了。非常非常感谢各位的陪伴，青海、太原、西安、丽水，我会时常想起这些美妙的时光，同门之谊，没齿难忘，江湖再会！

除了课题组，也非常感谢其他老师和朋友在这三年的帮助。首先非常感谢潘毅群老师和王慧龙师兄在我博士申请期间给予的帮助！非常感恩遇上袁晓磊，帮我开启了第一篇文章，将自己的论文经验毫无保留地与我分享，往后的学术生涯一定保持交流，也祝你新婚快乐！感谢羽毛球院队的小伙伴们，波波、冰冰、小彭、聪哥、鱼、卧龙、林付凌、阿曼、乐乐、陈逸山、蜘蛛侠、亮仔、标哥和大卫，非常怀念与你们一起大汗淋漓的日子，烦恼和忧愁都不会在球场停留。非常感谢本科室友在我日常 emo 时的排解，每次聊天都能有所宽慰。

一路走来，时常因为无法回馈年迈的父母而感到愧疚，非常感谢父母的理解与信任，谢谢哥哥们一直帮我撑起家庭重担，鼓励我继续深造，让我能一直放心前行，成为更好的自己。非常非常感谢谢天同学多年的陪伴与理解，承受着我大量的坏心情与臭脾气，相信我们马上就能拥有一起幻想过的生活，我会继续努力！

最后，感谢您能读到这里，衷心希望这三年的工作能够对暖通行业的发展起到一定的推动作用，如果对文章内容有疑问，非常欢迎与我一起讨论！

二零二二年五月，于同济友园十七号楼

个人简历、在读期间发表的学术论文与研究成果

个人简历:

陈志森, 男, 1995 年 12 月生。

2018 年 7 月毕业于西安建筑科技大学 建筑环境与能源应用专业 获学士学位。

2019 年 9 月入同济大学读硕士研究生。

已发表论文:

- [1] **Chen Zhisen**, Guan Hang, Yuan Xiaolei, et al. Rule-based generation of HVAC duct routing[J]. Automation in Construction, 2022, 139: 104264.
<https://doi.org/10.1016/j.autcon.2022.104264> (SCI, Q1, IF: 7.7)
- [2] Chen Yongbao, Guo Mingyue, **Chen Zhisen**, et al. Physical energy and data-driven models in building energy prediction: A review[J]. Energy Reports, 2022, 8: 2656–2671.
<https://doi.org/10.1016/j.egy.2022.01.162> (SCI, Q1, IF: 6.87)
- [3] Chen Yongbao, **Chen Zhisen**, Chen Zhe, et al. Dynamic modeling of solar-assisted ground source heat pump using Modelica[J]. Applied Thermal Engineering, 2021, 196: 117324.
<https://doi.org/10.1016/j.applthermaleng.2021.117324> (SCI, Q1, IF: 5.295)
- [4] Yuan Xiaolei, **Chen Zhisen**, Liang Yuming, et al. Heating energy-saving potentials in HVAC system of swimming halls: A review[J]. Building and Environment, 2021, 205: 108189.
<https://doi.org/10.1016/j.buildenv.2021.108189> (SCI, Q1, IF: 6.456)
- [5] **陈志森**,陈喆,张亚珂,郭明月,许鹏.空调系统风管连接方案自动设计.建筑节能(中英文), 2022, 50(06).

已公开发明专利:

- [1]许鹏,**陈志森**,李梦西,陈喆,陈永保.一种短时电力负荷组合式预测方法[P].上海市: CN112865093A, 2021-05-28.
- [2]许鹏,陈喆,陈永保,**陈志森**.一种居住建筑多能互补的需求相应控制方法[P].上海市: CN112968445A, 2021-06-15.
- [3]许鹏,郭明月,何睿凯,**陈志森**,陈喆,陈永保.一种管道井寻优算法[P].上海市: CN112241563A, 2021-01-19.
- [4]许鹏,肖桐,陈喆,**陈志森**,郭明月,沙华晶.一种基于拓扑分层抽象的空调水系统原理图自动生成图方法[P].上海市: CN112865093A, 2020-10-20.

研究及项目经历:

- [1] 2019 年 9 月~2021 年 6 月: 多能互补集成优化的分布式能源系统示范 (项目编号: 2017YFB0903400), 国家重点研发项目(十三五), 主要参与人
- [2] 2019 年 9 月~2021 年 6 月: 基于 BIM 的绿色建筑运营优化关键技术研发 (项目编号: 2018YFC0705900), 国家重点研发项目(十三五), 参与人

同济大学学位论文原创性声明

本人郑重声明：所提交的学位论文《空调输配系统（风）自动化设计研究》，是本人在导师指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或者没有公开发表的作品的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名：陈志森

日期：2022年05月18日

同济大学学位论文版权使用授权书

本人完全了解同济大学关于收集、保存、使用学位论文的规定，同意如下各项内容：按照学校要求提交学位论文的印刷本和电子版本；学校有权保留学位论文的印刷本和电子版，并采用影印、缩印、扫描、数字化或其它手段保存论文；学校有权提供目录检索以及提供本学位论文全文或者部分的阅览服务；学校有权按有关规定向国家有关部门或者机构送交论文的复印件和电子版；允许论文被查阅和借阅。学校有权将本学位论文的全部或部分内 容授权编入有关数据库出版传播，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。

本学位论文属于（在以下方框内打“√”）：

保密，在_____年解密后适用本授权书。

不保密。


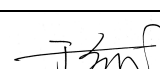
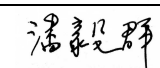
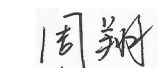
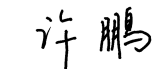

学位论文作者签名：陈志森

指导教师签名：许鹏

日期：2022年05月18日

日期：2022年05月20日

七、学位论文答辩委员会决议

姓 名	陈志森	学 号	1930254	所在学科/专业	供热、供燃气、通风及空调工程
指导教师	许鹏	答辩日期	2022年6月1日	答辩地点	腾讯会议
论文题目	空调输配系统（风）自动化设计研究				
<p>陈志森同学的硕士学位论文《空调输配系统（风）自动化设计研究》以风管系统自动设计为对象展开相关研究，为空调系统全自动设计的实现提供了方法。</p> <p>论文提出了一种风管自动布置方法，主要包括风管布置区域提取、风口连接和快速水力计算三个部分，通过对六个不同案例的测试验证了算法的有效性，并基于 Python 平台实现了四栋建筑标准层的空调系统风管的自动设计。</p> <p>论文研究目标明确，结构完整，条理清晰，模型及论证合理，结论可信。论文工作量大，体现了作者具有扎实的基础理论和专业知识，具备了从事科学研究工作和解决实际工程问题的能力。</p> <p>陈志森同学在答辩过程中，表述完整，思路清晰，能正确回答答辩委员提出的问题。论文达到了工学硕士学位论文要求。经无记名投票表决，四位答辩委员中，四位同意并建议授予陈志森同学工学硕士学位，并推荐申请优秀硕士学位论文。</p> <p style="text-align: right;">答辩委员会主席签名：  2022 年 06 月 01 日</p>					
<p>答辩委员会共 <u>4</u> 人，经表决，<u>4</u> 人建议授予申请人硕士学位。根据《同济大学学位授予工作细则》^[注]（在□内划“√”）：</p> <p><input type="checkbox"/> 申请人可在一年内修改论文，申请重新答辩一次。 <input checked="" type="checkbox"/> 建议授予申请人硕士学位。 <input type="checkbox"/> 建议不授予申请人硕士学位。</p>					
<p><input checked="" type="checkbox"/> 推荐为同济大学优秀硕士学位论文。</p>					
答辩委员会成员签名	职 务	姓 名	职 称	单 位	签 名
	主 席	于航	教授	同济大学	
	委 员	潘毅群	教授	同济大学	
		周翔	教授	同济大学	
		许鹏	教授	同济大学	
	秘 书	赵美	工程师	同济大学	

注：根据《同济大学学位授予工作细则》第十一条规定：

1. 申请人获得全体答辩委员会成员三分之二以上（含）同意票，为建议授予申请人硕士学位；
2. 申请人获得全体答辩委员会成员二分之一以上（含）、三分之二以下（不含）同意票，申请人可在一年内修改论文，申请重新答辩一次；
3. 申请人获得全体答辩委员会成员二分之一以下（不含）同意票，为建议不授予申请人硕士学位。