



同濟大學

TONGJI UNIVERSITY

硕士学位论文

(学术学位)

空调风系统管路自动生成方法研究

姓 名：李俊杰

学 号：2230292

学 院：机械与能源工程学院

学科门类：工学

一级学科：土木工程

二级学科：供热、供燃气、通风及空调工程

研究方向：建筑节能技术

指导教师：许鹏教授

联合培养单位：

二〇二五年五月



同濟大學
TONGJI UNIVERSITY

A thesis submitted to

Tongji University in partial fulfillment of the requirements for
the degree of Master of Engineering

**Research on the Automatic Generation
Method of Air Conditioning Duct System
Layout**

Candidate: Junjie Li

Student Number: 2230292

School/Department: School of Mechanical Engineering

Categories: Engineering

First-level Discipline/Degree: Civil Engineering

Second-level Discipline/Degree's Field: Heating, Gas
Supply, Ventilation, and Air Conditioning Engineering

Research Fields: Building Energy Saving Technology

Supervisor: Prof. Peng Xu

May 2025

摘要

随着建筑智能化水平的不断提高，风系统作为建筑环境控制的核心子系统，其设计的自动化、智能化程度日益成为研究热点。传统暖通空调风系统设计过程复杂，严重依赖人工经验和既定规则，难以满足现代建筑在空间布局复杂化与性能多目标化背景下的设计需求。本文面向风系统中支管路径规划与管径水力性能优化两个核心环节，提出了一种融合强化学习与多目标优化算法的管路自动化设计方法。

首先，针对风系统路径规划任务的特点，构建了基于二维空间网络的强化学习环境，引入离散方向与连续步长相结合的混合动作建模方式，增强智能体对空间的灵活探索能力。为引导智能体稳定学习路径策略，设计了包含行为有效性与路径性能的复合奖励函数体系：前者用于判定路径是否连通且合法，后者将路径长度、阻力与系统不均衡性转化为初投资与运行能耗成本，作为训练中优化的综合目标。开发了一套具有代表性的样本环境数据集，涵盖矩形、L型、T型等多类房间结构，并建立了多指标加权融合难度评估方法用于量化不同空间场景对路径生成的挑战程度。

其次，构建了融合空间感知与时序建模能力的混合神经网络结构，基于残差卷积模块提取输入网格的结构语义特征，通过坐标增强机制提升网络对绝对位置的感知能力，并利用 LSTM 实现路径规划过程中状态序列的时序建模。在训练方面，本文采用 PPO 算法进行策略优化，并结合 Optuna 框架完成超参数的自动搜索与优化。实验结果表明，在中低难度环境下智能体能够有效学习路径规划策略，验证了方法的可行性。

最后，在水力性能优化阶段，针对路径生成后的风系统拓扑结构，建立了基于 NSGA-II 的多目标管径优化算法，优化目标包括管道面积、系统阻力和阻力不平衡性。通过定义管径可行域、设计编码解码机制与修复算子，构建了完整的进化优化流程。案例验证表明所提出方法可在多个性能指标之间实现合理权衡，生成的 Pareto 解优于传统基于假定流速法的结果，尤其是在管道面积指标和不平衡性方面有显著改善。

关键词：空调风系统设计，管道布线，路径规划，多目标优化

ABSTRACT

With the increasing level of building intelligence, the automation and intelligence of air duct system design—an essential subsystem in heating, ventilation, and air conditioning (HVAC)—have become a research hotspot. Traditional HVAC duct design processes are complex and heavily reliant on manual experience and predefined rules, which makes it difficult to meet the demands of modern buildings characterized by complex spatial layouts and multi-objective performance requirements. This study focuses on two core aspects of air duct system design: branch duct path planning and hydraulic performance optimization. It proposes an automated duct design method that integrates reinforcement learning with multi-objective optimization algorithms.

Firstly, a reinforcement learning environment was constructed in a 2D grid space tailored to the characteristics of air duct path planning. A hybrid action modeling strategy combining discrete directions and continuous step sizes was introduced to enhance the ability of agent to flexibly explore spatial environments. To enable stable policy learning, a compound reward function was designed, including both behavioral validity and path performance components. The former ensures the generated path is valid and connected, while the latter translates path length, pressure loss, and system imbalance into lifecycle costs encompassing initial investment and operational energy consumption. A representative dataset of environmental samples was developed, covering various room structures such as rectangular, L-shaped, and T-shaped rooms. In addition, a difficulty evaluation method based on multi-indicator fusion was proposed to quantify the challenge level of path planning in different spatial scenarios.

Secondly, a hybrid neural network architecture with spatial perception and temporal modeling capabilities was developed. Residual convolution modules were used to extract semantic features from grid inputs, while coordinate augmentation was introduced to improve absolute position awareness. An LSTM module was applied to model temporal dependencies in the path construction process. For training, the PPO algorithm was adopted for policy optimization, and the Optuna framework was used for automated hyperparameter search and tuning. The experimental results demonstrate that the agent successfully learns basic path extension strategies in medium and low difficulty environments, validating the feasibility of the proposed method.

Finally, in the hydraulic performance optimization phase, a multi-objective duct sizing method based on the NSGA-II algorithm was proposed, targeting pipe surface area, system resistance, and resistance imbalance. A complete evolutionary optimization process was established, including feasible domain definition, encoding-decoding mechanism, and solution repair operators. Case studies demonstrate that the proposed approach achieves a reasonable trade-off among performance objectives, with the generated Pareto front significantly outperforming the conventional assumed velocity method, especially in terms of duct area and balance performance.

Key Words: air duct system design, duct routing, path planning, multi-objective optimization

目录

摘要.....	I
ABSTRACT.....	II
目录.....	IV
第 1 章 引言.....	1
1.1 研究背景及研究意义.....	1
1.1.1 研究背景.....	1
1.1.2 研究意义.....	3
1.2 研究现状.....	4
1.2.1 建筑设计自动化.....	4
1.2.2 管道路径规划.....	6
1.2.3 强化学习算法.....	9
1.2.4 研究现状总结.....	10
1.3 研究内容及技术路线.....	11
第 2 章 风系统管路生成输入准备.....	14
2.1 概述.....	14
2.2 走廊主管布置方法.....	14
2.3 风口布置算法.....	16
2.4 本章小结.....	19
第 3 章 风系统支管路径生成问题建模与环境构建.....	20
3.1 概述.....	20
3.2 风系统支管路径生成问题建模.....	20
3.2.1 空间网格表示.....	20
3.2.2 状态空间建模.....	22
3.2.3 动作空间建模.....	22
3.2.4 状态转移与路径生成逻辑.....	23
3.2.5 奖励函数.....	25
3.3 风系统支管路径生成训练环境集构建.....	28
3.3.1 训练环境生成方法.....	28
3.3.2 多指标加权融合的训练环境难度评估方法.....	32
3.4 本章小结.....	34
第 4 章 风系统支管路径生成策略模型训练.....	36
4.1 概述.....	36
4.2 网络结构设计.....	36
4.2.1 共享特征提取模块.....	36

4.2.2 策略输出模块.....	37
4.3 强化学习算法选择与训练流程.....	39
4.3.1 PPO 算法原理	39
4.3.2 回报计算与优势估计.....	40
4.3.3 损失函数.....	41
4.3.4 模型训练.....	41
4.4 本章小结.....	43
第 5 章 风系统管路水力性能多目标优化.....	45
5.1 概述.....	45
5.2 风系统管路水力性能优化问题建模.....	45
5.2.1 风系统管路生成结果储存格式.....	45
5.2.2 风系统管路水力计算模型.....	47
5.2.3 风系统管路水力性能优化目标.....	49
5.3 风系统管路水力性能优化方法与流程.....	50
5.3.1 决策变量定义与个体编码.....	50
5.3.2 设计解生成与修复机制.....	52
5.3.3 多目标优化算法.....	53
5.4 本章小结.....	57
第 6 章 风系统管路自动生成方法验证.....	58
6.1 概述.....	58
6.2 风系统支管生成案例验证.....	58
6.3 风系统管路水力性能优化验证.....	60
6.4 本章小结.....	68
第 7 章 结论与展望.....	69
7.1 结论.....	69
7.2 局限性与展望.....	70
参考文献.....	71
附录 A 支管路径生成环境代码（节选）	77
附录 B 管路水力性能优化代码（节选）	80
致谢.....	84
个人简历、在读期间发表的学术成果.....	85

第1章 引言

1.1 研究背景及研究意义

1.1.1 研究背景

在“双碳”战略背景下，我国正大力推进建筑能效提升与绿色低碳发展。而在建筑能耗中，尤其是在运行阶段的能耗中，暖通空调（Heating, Ventilation, and Air Conditioning, HVAC）系统的能耗占据了十分重要的部分。根据国际能源署（International Energy Agency, IEA）的数据，2022 年，建筑运营占全球最终能源消耗的 30%，占全球能源相关排放的 26%^[1]。美国能源信息署（U.S. Energy Information Administration, U.S. EIA）的研究报告指出，在商业建筑中，暖通空调系统占建筑总能耗的 52%^[2]，尤其在办公楼、商场、医院等高负荷建筑中，空调系统往往成为主要的能耗来源^[3]。因此，提升暖通空调系统的设计质量与运行效率，对于降低建筑能耗、推动碳达峰与碳中和目标的实现具有重要意义。

在暖通空调系统中，空调风系统作为核心子系统之一，不仅直接负责将经过调节的空气输送至各个功能区域，并且在实现室内热湿平衡中发挥着至关重要的作用。当前，空调风系统的设计在满足建筑环境舒适性的同时，还需兼顾能源消耗和运行效率，二者之间往往存在一定的矛盾。现有的系统设计方法在管路布局、风量分配及气流组织等方面，往往依赖于传统的经验设计和人工干预，这不仅导致设计过程耗时耗力，同时也难以全面考虑到系统运行过程中复杂多变的情况。特别是在当前追求绿色节能和智能化建筑管理的背景下，传统设计方法在提高送风效率、降低能耗以及提升用户舒适性等方面的局限性日益凸显。由此，迫切需要从方法论层面进行根本性的革新。

目前的空调风系统设计方法主要有以下几点问题：

（1）风系统设计流程复杂，人工参与度高

传统的暖通空调设计流程包含负荷计算、系统形式确定、设备选型、管道设计、水力计算、出图等，流程复杂且环环相扣，依赖于设计人员的经验与工程判断。随着过去几十年来电子信息技术的发展，整个建筑设计行业从原先的在纸上手绘图纸，到后来借助各种计算机辅助设计（Computer Aided Design, CAD）软件进行二维图纸的绘制，再到近年来的借助建筑信息模型（Building Information Modeling, BIM）的发展，设计师可以借助 BIM 软件进行三维的设计，这极大地提升了各专业协调的效率。尽管如此，由于暖通空调系统设计本身的复杂性以及

对于设计质量要求的提高，暖通空调系统的设计仍然是一项耗时耗力的工作。

一方面，暖通空调系统设计需要较高的技术门槛，设计人员不仅需要有相关的专业背景知识，还需要有丰富的工程经验，并且熟悉相关的设计标准和规范。另一方面，由于建筑图纸反复修改、建筑体量大且结构复杂、甲方要求变更、各专业协调等原因，暖通图纸需要根据多方的意见反复地进行修改。因此，暖通空调系统的设计在专业门槛高、技术性强的同时，其中大多数的工作又有很强的重复性，导致整个设计流程耗时长、重复性高、生产效率低下。一项针对暖通空调系统设计的调研发现^[4]，对于从业经验少于5年的工程师来说，在整个暖通空调系统的设计流程中，出图是修改次数最多，也是耗时最长的一项，平均需要经历9.27次修改，耗费的时间长达125.64小时。

（2）基于规则的方法智能化程度低，难以适应复杂设计需求

目前关于暖通空调输配系统管道的自动生成已有一定研究进展，但现有方法主要依赖于预设规则^[5,6]或启发式算法^[7-9]，从而存在明显局限性。

首先，这些方法在生成设计方案时往往只能保证解的可行性，难以实现最优解，即在风管总长度、系统压损以及成本投入之间难以达到全面权衡。由于各项指标存在内在的矛盾关系，预设规则难以同时兼顾这些方面的优化，从而导致生成的管路布局在实际运行中可能表现出局部最优而非全局最优的特征。其次，当前方法对于形态多样且结构复杂的不规则建筑适应性不足。当面临不规则建筑或者大空间布局时，传统基于规则的方法往往难以灵活应对建筑结构的多变性，从而影响风管布置的整体效果。最后，这类方法的可扩展性较差。由于预设的规则集往往针对特定类型的建筑或系统进行了优化设计，其普适性不高，难以适应新项目或新建筑类型的要求。随着建筑功能和设计要求的不断升级，传统的规则集不仅迁移困难，而且在维护和更新上也面临较高成本。

因此，传统基于规则的布管策略难以满足当今建筑设计日益复杂化和定制化的需求，亟待引入更具智能性和自适应能力的方法体系，例如结合机器学习、人工智能优化算法以及动态参数调整的新型智能设计平台，以提升设计的自动化水平和适应性。

（3）风系统设计需在多目标性能指标之间进行优化，设计难度高

风系统的性能优劣不仅体现在各个风口是否有效连通，更取决于整个系统水力性能的合理性与整体优化效果。具体而言，该优化问题主要涉及三个关键维度：首先，管道面积直接决定了材料用量和安装成本，是初期投资的重要组成部分；其次，系统阻力则对风机功耗和运行能耗产生显著影响，直接关系到系统的运营成本 and 能源效率；最后，阻力平衡度则反映系统送风均衡性，其调控稳定性对用户的舒适性具有决定性作用^[10-12]。这三者之间既相互制约，又互为影响，构成了

一个典型的多目标优化问题。

在此背景下,传统设计方法主要依赖经验公式和人工不断迭代来寻求合理折中的解决方案^[13,14]。然而,当面对大规模、复杂的实际系统时,传统方法不仅难以快速找到理想解,而且往往存在设计周期长以及优化精度不足的问题。为解决这一系列挑战,亟需发展新型的多目标优化设计方法,该方法能够在保证送风效果的同时,通过系统化的建模和优化算法精细调控系统参数,实现风系统水力性能的优化。这可能包括引入优化算法,以期在初投资与运行费用、系统效率与使用舒适性之间寻求最优平衡,从而推动暖通空调系统在节能设计和智能控制方向上的突破。

1.1.2 研究意义

面向当前建筑空调风系统设计普遍存在设计流程复杂、高度依赖人工经验、智能化水平低等关键问题,本研究具有以下三方面的重要意义:

(1) 响应建筑节能战略,推动绿色低碳目标实现

在“双碳”战略背景下,建筑节能成为实现国家能源转型的关键路径之一,而暖通空调系统作为建筑运行能耗的核心组成部分,其设计优化对能耗水平具有决定性影响。风系统作为暖通子系统中的关键环节,其路径布局和参数配置直接关系到风机能耗、系统平衡性与运行成本。通过构建自动化、智能化的风系统设计方法,可从源头上提升建筑能效水平,助力绿色建筑推广和建筑行业低碳转型。

(2) 降低人工负担,提高设计效率与质量稳定性

现有风系统设计流程依赖设计人员进行大量手工操作与反复修改,特别是在建筑空间复杂、设计需求变更频繁的项目中,路径布置和参数调整极为耗时,重复性劳动强、出图周期长,严重制约了暖通设计的生产效率。本研究通过引入强化学习与优化算法,实现路径生成与参数设定的自动化,有效减少人工干预,缩短设计周期,提升出图效率,同时增强方案的一致性与标准化程度,显著缓解当前工程实践中耗时、费力、低效的设计痛点。

(3) 突破规则方法的瓶颈,构建可拓展的智能设计机制

尽管近年来已有部分研究尝试将布管规则嵌入CAD/BIM平台以实现一定程度的自动布线,但大多仍基于固定规则和启发式搜索方法,且在面对多变的结构条件与设计需求时缺乏足够的适应性与扩展性。本研究旨在构建一种具有良好可拓展性的风系统设计机制,使其能够适应多种建筑类型与设计边界条件下的管路生成任务,在满足规范约束的基础上,支持风系统设计流程的标准化与智能化,从而推动风系统布管设计从规则驱动向数据驱动和策略驱动的范式转型。

1.2 研究现状

1.2.1 建筑设计自动化

随着数字建造和智能化建筑技术的快速发展,建筑设计自动化已成为当前建筑工程领域的重要研究方向之一。建筑设计过程传统上以人为主导,需整合结构、建筑、暖通、电气等多个专业的设计逻辑,任务高度复杂、交叉繁多。图 1.1 展示了暖通空调系统设计的流程示意图,设计过程通常复杂且耗时。自动化设计的提出,旨在借助算法、数据和模型的力量,提升建筑设计效率、减少人工干预,并在更高层次上实现系统性能的协同优化。

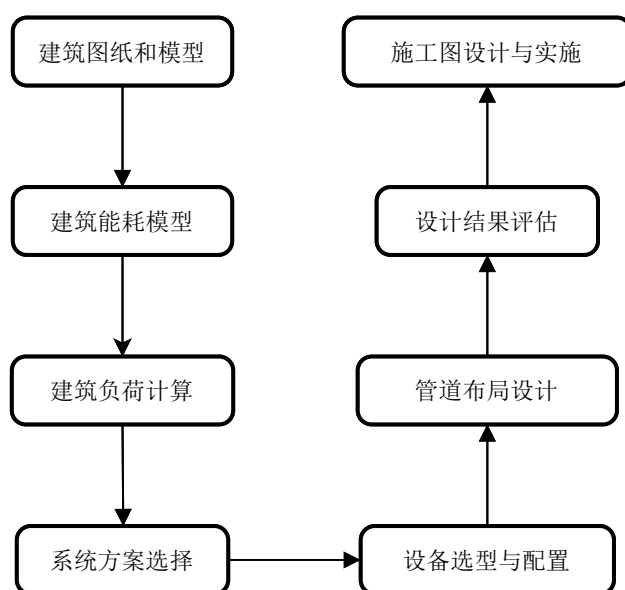


图 1.1 暖通空调系统设计流程示意图

BIM 的普及为建筑设计自动化奠定了数据基础与协同平台。BIM 不仅能表达三维构件几何关系,还包含材料、空间、设备、负荷等多维信息,是连接建筑设计、施工与运维的核心载体。基于 BIM 的建筑能耗建模尤其适用于早期设计阶段,可以将最合适且最具成本效益的节能设计方法融入到整体建筑设计流程中^[15]。借助 BIM 平台,可以实现多专业模型的集成管理与冲突检测,为后续自动布管、设备布置、路径规划等任务提供结构边界与空间条件的约束支持^[16]。Sha 等人提出了一种集成半自动化设计工具的通用框架,用于 HVAC 系统设计,包含负荷计算、系统选型、风管设计和设备表生成 4 个模块^[17]。Wang 等人利用 BIM 建立了自动设计全流程框架,包含建筑信息模型简化、建筑能耗模型生成与负荷计算、HVAC 系统拓扑生成与设备选型以及系统图生成等自动化流程^[4],旨在实现整个设计流程的自动化,以取代目前暖通空调系统高度依赖人工的设计流程。

HVAC 系统设计的负荷模拟的关键元素包括几何图形、材质、空间类型和热分区等, 这些基本元素的数据从建筑信息模型到建筑能耗模型 (Building Energy Model, BEM) 交互方法主要包括基于 gbXML (Green Building XML, 绿色建筑可扩展标记语言) [18,19] 和基于 IFC (Industry Foundation Class, 工业基础类) [20,21] 的方法。Yang 等通过基于修补平面图上的拉伸重建模型, 从而避免了建模误差和无关细节, 以支持以 gbXML 格式记录轻量级几何图形 [22]。Bracht 等人通过 gbXML 预测住宅的热负荷, 指出不同 BIM 创作工具的 gbXML 文件之间存在不一致的问题 [23]。Trani 等人开发了一套操作流程和方法, 将 BIM 模型中的信息用 IFC 标准导出供能耗模拟软件读取, 实现从 BIM 模型到 BEM 模型的转换 [24]。Ying 等人开发了一个自动化系统, 用于验证 IFC 二级空间边界的语法和语义正确性、几何正确性和一致性 [25], 并提出了使用 IFC 模型中建筑元素和空间的多种几何表示来计算二级空间边界的几何和语义信息新方法 [26]。近年来随着大语言模型 (Large Language Model, LLM) 的发展, 许多研究聚焦于使用大语言模型来生成建筑能耗模型, 旨在提高模型性能和减少工程工作量 [27-29]。

一些研究总结了 BIM 在能源模拟应用中面临的一些挑战, 例如互操作性问题、缺乏标准以及缺乏扩展现有 BIM 模式的简便方法和可用的相应解决方案 [30-32]。基于 BEM 的建筑能耗模拟 (Building Energy Simulation, BES) 可用于预测建筑全年运行能耗、负荷变化趋势与系统响应行为, 为系统选型、管道设计与节能策略制定提供科学依据。Trčka 总结了 HVAC 系统设计和分析模拟工具的分类及其适用问题, 并就如何根据模拟目标选择合适的 HVAC 建模方法提出了建议 [33]。

随着建筑性能驱动设计理念的兴起, 越来越多的研究尝试将优化算法嵌入到建筑空调系统的设计流程中, 将优化结果作为系统选型的指导。在系统层面, 基于建筑负荷需求与能耗模拟结果, 采用多目标优化方法对暖通空调系统进行选型与配置, 优化目标包括能效比、设备容量、初投资与生命周期成本等 [34-36], 从而在满足负荷与舒适性前提下, 平衡能效与经济性, 为建筑节能提供有效支撑。Feng 等人提出了将优化算法对超高层建筑空调系统拓扑结构、组件尺寸和控制设置进行优化, 并在能源消耗和成本方面进行了评估 [37]。

在设备布置和管路设计层面, 优化方法也逐步被引入到末端设备布置和风水系统的布置中。Pouya 等人提出了一个集成优化方法生成通风口、风道末端和管道的具体位置, 以方便制造和提高组装效率 [38]。Chen 等人采用基于规则的遍历方法, 考虑管道压力平衡、施工障碍、减少空气阻力和最大限度地降低施工成本来生成空调风系统管道路径 [5]。Liang 等人将基于规则的分治法引入管道布线设计, 实现了暖通空调管道系统设计的完全自动化 [6]。Dong 等人开发了一种多联机系统制冷剂管道自动布线方法, 利用蚁群优化 (Ant Colony Optimization, ACO)

实现制冷剂管道的自动生成^[9]。在建筑其他机电系统领域, Jin 等人提出了一种火灾报警系统自动化设计框架, 包含信息预处理、设备布置、系统回路生成、设备布线和结果生成五个自动化流程^[39]。

1.2.2 管道路径规划

管路设计问题根据管路的数量和类型分为单管路设计、多管路设计以及分支管路设计, 而暖通空调风系统管道属于典型的分支管路设计问题。目前对于管道设计的研究方法主要是将管道设计问题转化为路径规划问题进行研究。而对于路径规划问题, 国内外学者针对不同的子问题提出了很多不同的方法, 包括迷宫算法、Dijkstra 算法、A*算法等, 以及采用粒子群算法 (Particle Swarm Optimization, PSO)、蚁群算法和遗传算法 (Genetic Algorithm, GA) 等优化方法对管路进行优化。

迷宫算法由 C. Y. Lee 于 1961 年提出^[40], 这是一种基于广度优先搜索的算法, 因此通常能够找到最短路径, 但是速度很慢并且需要大量内存。1959 年, E. W. Dijkstra 提出 Dijkstra 算法, 其主要是通过贪心原则逐个遍历最小子节点, 然后利用松弛方法去优化路径选择, 最终将最优路径存放到可读列表当中, 以此来解决最优路径规划问题^[41]。但是 Dijkstra 算法无法解决存在负权边的问题。Bellman-Ford 算法是求解单源最短路径问题的一种算法, 由 Richard Bellman^[42]和 Lester Ford^[43]分别于 1958 年和 1956 年提出。它的原理是对图进行松弛操作, 得到所有可能的最短路径, 其优于 Dijkstra 算法的方面是边的权值可以为负数、实现简单, 缺点是时间复杂度过高, 但算法可以进行若干种优化, 提高效率。Yefim Dinitz 使用 Bellman-Ford 和 Dijkstra 算法的混合算法, 改进了具有稀疏负成本边分布的图的 Bellman-Ford 的运行时间界限^[44]。Jeong, IK 提出了一种基于图形处理单元 (GPU) 的距离矢量路由协议中使用的 Bellman-Ford 算法的实现, 基于 GPU 的多线程并行运算提升了 Bellman-Ford 算法的效率^[45]。

A*算法于 1968 年提出, 它是一种结合了深度优先搜索 (Depth First Search, DFS) 和广度优先搜索 (Breadth First Search, BFS) 的启发式搜索算法^[46], 结合了 Dijkstra 算法的最短路径搜索特性和贪婪搜索的启发式信息, 以有效地找到从起点到目标点的最短路径。A*算法通过维护一个开放列表, 选择估价值最小的节点来搜索, 同时使用启发式函数来估计到目标的代价。在 A*算法的基础上, 提出了 D*算法, 即 Dynamic A*算法, 包括 D*^[47]、Focussed D*^[48]、D* Lite^[49]等算法。D*算法和 A*类似, 不同的是, 其代价的计算在算法运行过程中可能会发生变化。D*算法允许在路径规划过程中进行增量更新, 以反映环境的变化, 当环境变化时, D*算法根据新的信息重新计算路径的成本, 以适应新的条件。2002 年,

Sven Koenig 提出 PLA*算法^[50]，是一种基于 A*算法的增量启发式搜索算法，用来处理动态环境下从给定起始点到给定目标点的最短路径问题。Field D*^[51]扩展了 D 和 D* Lite，是一种基于插值的规划算法，它使用线性插值来有效地生成低成本路径，从而消除不必要的转向。并且支持动态地图，可以实时更新障碍物的位置和属性，而不必重新计算整个路径。它使用势场（Potential Field）来表示地图上各个位置的吸引力或斥力，以便路径规划时考虑动态障碍物。

1991 年，D. Zhu 等将 A*算法应用于管道设计，将管道线路被视为刚性物体（机器人）留下的路径，使用机器人路径规划技术自动设计管道布局的实现系统^[52]。2018 年，马江涛等提出基于局部二次学习概率路径图算法（Local Twice Learning Probabilistic Roadmap Method, LTL-PRM）的分支管路自动布局方法，以解决非正交分支管路的自动布局问题^[53]。该方法首先采用回溯算法计算出端点之间的最小斯坦纳树，并对其中的斯坦纳点位置进行优化处理，从而获得管路系统的分支点位置，在此基础上通过局部二次学习构建路径图，并利用 Dijkstra 算法搜索出各分支管路的最短路径，最后对搜索出的路径进行优化，获得分支管路的最终布局结果。2020 年，Farhan Ajiwaskita 等采用 Dijkstra 算法进行了船舶机舱管路优化，考虑多个约束条件和惩罚条件，目标函数不仅考虑了管道长度最小化，同时考虑弯曲次数和交叉次数^[54]。2021 年，Stanczak 等使用加权 A*搜索算法与线性规划相结合来解决在 3D 连续空间内管道布线问题^[55]。该方法将管道布线问题表述为布线计划空间中的最短路径问题，其中布线计划由组成管道的部件以及施加在这些部件上的几何约束来指定。

遗传算法是一种启发式搜索和优化算法，受到自然界生物进化的启发^[56]。它用于解决复杂的优化和搜索问题，其中可能存在多个候选解，以寻找最佳解或近似最佳解。遗传算法的工作原理基于生物遗传学的概念，包括选择、交叉、变异和适应性。通过不断迭代选择、交叉和变异操作，遗传算法在解空间中搜索，逐渐改进个体的适应度，从而找到更好的解。遗传算法在解决复杂优化问题，如旅行商问题、机器学习模型参数优化等方面具有广泛的应用。

蚁群算法是一种启发式搜索和优化算法，受到蚂蚁寻找食物路径行为的启发^[57]。蚁群算法用于解决组合优化问题，如旅行商问题和调度问题，以寻找最优或近似最优解。蚁群算法模拟了蚂蚁群体的协作行为，通过信息素和启发式信息来引导搜索，以寻找优化问题的解。蚁群算法也具有自适应性，可以适应动态环境和不完全信息的情况。

粒子群优化是一种启发式搜索和优化算法，灵感来自鸟群或鱼群等自然群体的集体行为^[58]。PSO 算法用于解决各种优化问题，包括连续和离散优化问题，以寻找最优解或接近最优解。PSO 算法的核心思想在于通过模仿群体中个体之间的

信息共享和协同行为来寻找全局最优解。粒子的速度和位置更新过程模拟了每个粒子受到个体经验和群体经验的影响，以帮助更好地搜索解空间。

基于以上优化方法以及展开了许多管道设计的研究。Liu 等提出了一种基于 Steiner 树理论的分支管道布线算法，利用粒子群优化解决带障碍的欧几里德施泰纳最小树问题，实现航空发动机表面非直线管道的布线^[59,60]。Jiang 等提出一种基于 ACO 和 GA 的船舶管路设计问题的优化方法^[61]，在蚁群算法的计算过程中嵌入遗传算子，包括遗传算法中的交叉操作和变异操作，以提高算法的计算性能，以及一种进化改进多蚁群优化（Co-Evolutionary Improved Multi-Ant Colony Optimization）算法，针对船舶管道布线的特点，改进了 ACO 算法，用于解决单管道布线问题，采用具有协同进化机制的多蚁群优化算法来解决多支管路由问题^[62]。Liu 等提出了一种改进的动态自适应蚁群算法（Improved Dynamic Adaptive ACO, IDAACO）用于解决布管设计问题，包括四种新颖的机制，即带有方向信息的启发式策略、自适应伪随机传输策略、改进的局部信息素更新机制和改进的全局信息素更新机制^[63]。Liu 等提出了一种走管与夹钳布局集成优化方法，解决装配约束和两者之间的设计耦合问题^[64]。Ren 等^[65]、Sui 等^[66]采用改进的基于遗传算法进行了船舶的管道设计优化。Niu 等结合迷宫算法、非支配排序遗传算法 II（Nondominated Sorting Genetic Algorithm II, NSGA-II）和合作协同进化非支配排序遗传算法 II（Co-evolutionary NSGA-II）提出了一种处理管道路线规划问题的优化程序^[67]。

部分研究尝试将优化算法与路径规划方法相结合，实现路径生成过程中的性能引导与多目标协调，这类路径和优化融合策略在一定程度上突破了传统启发式搜索算法的局限。Waghoo 等人定量评估了三种多目标的路径规划算法在建筑工地应用的性能，包括 Dijkstra、A*和遗传算法^[68]。2020 年，Zongran Dong 等结合 A*算法与遗传算法，提出了 A*-GA-Router 方法求解船舶管道设计^[69]。2022 年，Gunawan 等提出了一种基于 Dijkstra 和遗传算法的寻径算法的船舶机舱管道自动支撑系统^[70]。该方法从弯管成本、穿越管成本、管架成本降低、管系约束、计算时间缩短、管道路径设计程序等方面考虑，寻找最短路径。采用 Dijkstra 的最短路径算法，根据船舶机舱布局确定起点与目标点之间的最短路径，采用遗传算法确定管道施工顺序。2022 年，Zong-ran 等提出了一种基于多目标蚁群优化（Multi-Objective ACO）的船舶管道路径设计方法，详细介绍了改进算法的关键技术，如蚂蚁移动方向、启发式信息、信息素更新规则、禁忌搜索和后退策略等^[71]。在此基础上，给出了多目标蚁群优化框架及其内部蚂蚁路径规划过程，给出了基于帕累托解和人为选择的多分支管道布线方法。2023 年，Yan Lin 等人^[72]提出了一种基于混合维度的多目标协同粒子群优化算法（Hybrid Dimensions Multi-objective

Cooperative PSO, HDMCPSO) 用于船舶管道航路设计, 该算法采用多粒子维度策略和基于多目标的每个子群代表的协同更新机制粒子群优化。该框架能够兼容地解决设施简化和管道布局两个独立的非确定性多项式 (Non-deterministic Polynomial, NP) 难题, 并获得高质量的帕累托解。

1.2.3 强化学习算法

强化学习是一种学习方法, 其中智能体通过试错方法与其周围环境交互, 并根据从先前交互中收到的奖励信号学习最佳行为策略^[73]。强化学习智能体的学习过程模仿了人类或动物的学习方法。近年来, 强化学习已成为解决复杂顺序决策任务的有效技术。强化学习的示意图如图 1.2 所示, 其主要流程可以描述为: 智能体 (Agent) 通过与环境 (Environment) 交互进行学习, 交互包括行动 (Action), 奖励 (Reward), 状态 (State)。智能体在每一时刻根据当前状态选择动作, 环境根据该动作反馈新的状态及奖励值, 智能体据此优化行为策略, 智能体在于环境的交互中学习策略, 通过试错方式不断调整策略, 以最大化长期累积奖励。

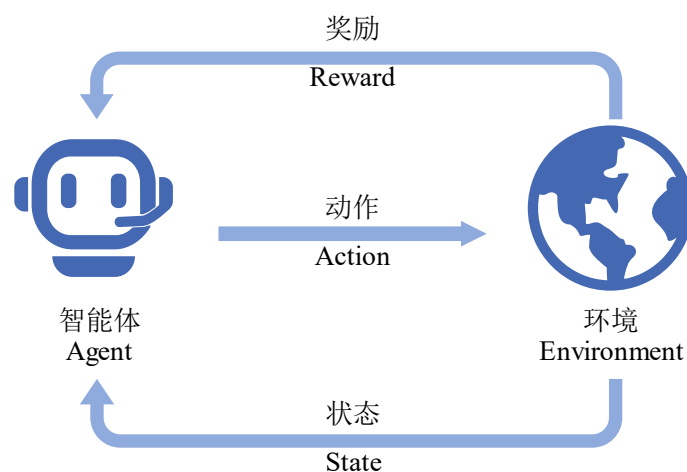


图 1.2 强化学习示意图

Q-learning^[74]是智能体学习如何在受控马尔可夫域中采取最佳行动的一种简单方法, 相当于一种动态规划的增量方法。Q-learning 算法将智能体所有状态-动作对存储在 Q 表中, 并通过引用 Q 表来搜索最优解。然而, 它在表示高维状态时受到计算复杂性的限制。Volodymyr 等^[75]利用深度神经网络来开发一种新型人工智能体, 称为深度 Q 网络, 使用深度神经网络逼近 Q 值函数, 使其能够处理高维和离散动作空间, 可以使用端到端强化学习直接从高维感觉输入中学习成功的策略。

无模型强化学习方法不依赖完整的环境模型, 而是通过智能体的交互经验进行策略评估与更新。典型方法包括蒙特卡罗 (Monte Carlo, MC) 方法和时序差分

(Temporal Difference, TD) 学习^[76]。相较于有模型方法, 无模型强化学习更适用于系统动力学难以建模或信息不完全的复杂环境。

深度强化学习 (Deep Reinforcement Learning, DRL) 结合了深度学习的特征表达能力与强化学习的策略优化机制, 能够在连续状态和动作空间下处理非线性、高维和不确定性的动态系统问题。在无需准确系统建模的前提下, DRL 可通过离线或在线方式, 从环境交互样本中学习近似最优策略, 成为近年来解决复杂智能控制与优化问题的有效工具^[77]。由于上述优点, DRL 已广泛应用于路径规划应用中, 例如无人机路径规划问题^[78]。此外强化学习已经广泛应用于工业过程控制^[79]、自动驾驶车辆^[80]、机器人^[81]、交通信号控制^[82]、能源管理^[83]、计算机视觉^[84]、医学^[85]等领域。

深度强化学习作为一种具备强大策略学习能力的智能方法, 为路径规划问题提供了新的解决范式, 尤其在非结构化、多约束环境中的自适应路径生成方面展现出独特优势。Sun 等人将强化学习用于机器人动态路径规划, 所提出的方法实现了清洁效率最高和对通行的影响最小^[86]。Li 等人引入了一个基于强化学习的超启发式框架, 求解多台自动导引车的任务分配和路径规划, 以最小化行程时间^[87]。深度强化学习还常被用于紧急灾害救援中人的卡车-无人机协同路径问题^[88]、空中交通管理系统飞行航线规划问题^[89]和智能仓库或工厂中自动导引车的实时路径规划^[90]。

在建筑能源领域, 深度强化学习已经广泛应用于暖通空调系统控制^[91]、热水系统控制^[92]、光伏储能系统控制^[93]、需求侧能源管理^[94]以及需求响应策略优化^[95]等任务中, 显著提升了建筑运行能效与系统控制智能化水平。

Chen 等将深度强化学习应用于解决三维空间的管道布局问题, 通常抽象空间的几何特征来建立训练环境, 并基于管道长度、弯头和安装距离三个约束定义奖励函数, 通过智能体与环境之间的交互收集数据并训练 DRL 模型^[96]。实验结果表明, DRL 模型可以比传统算法在更短的时间内完成空间管道布局任务, 同时保证高质量的布局结果。

综上所述, 深度强化学习作为一种具备强大泛化能力和策略优化能力的智能方法, 在路径规划、能源管理和建筑空间布局等多个工程应用领域均展现出广泛的适用性与研究价值, 尤其在处理非结构化、多约束、动态环境下的路径生成任务中, 展现出传统方法难以比拟的优势。

1.2.4 研究现状总结

以上内容系统梳理了建筑空调风系统设计相关领域的研究现状, 涵盖建筑设计自动化、管路路径规划、以及深度强化学习在路径生成与暖通空调系统中的应

用。现有研究显示，BIM 与能耗模型的集成成为建筑设计流程自动化提供了数据基础，相关研究已实现从模型构建到系统选型、负荷模拟的初步自动化流程。然而，在系统布置与管路设计层面，仍主要依赖规则方法与人工经验，缺乏有效的智能化策略指导。

路径规划作为管路设计中的核心问题，已有研究将 A*、Dijkstra 等传统图搜索算法，或遗传算法、蚁群算法等优化方法应用于管道路径生成任务。这些方法虽然在特定场景下取得一定成效，但普遍存在泛化能力弱、适应复杂建筑结构能力差、难以实现多目标优化等问题。

随着深度强化学习技术的发展，其在高维、多约束路径生成问题中的优势日益凸显。DRL 能够通过与环境的交互不断学习近似最优策略，具备良好的自适应性及泛化能力，已被广泛应用于机器人导航、交通路径规划及复杂空间布线等任务中。在建筑能源系统控制方面，DRL 也表现出优越的能效调控能力与策略优化效果。已有初步研究将其应用于三维空间管道布设任务，验证了其在提高效率与布局质量方面的潜力。

然而，现有风系统布管相关研究大多依赖预设规则或启发式算法，虽可在特定场景中实现路径的可行性布置，但普遍存在难以获得最优解、适应性差、扩展性弱等显著问题。基于规则的方法在面对结构非规则下的设计场景时鲁棒性不足，缺乏应对复杂建筑环境下布管任务的通用能力。因此，亟需一种具备自适应能力、能够统筹路径生成与性能优化、并适应多样化建筑场景的智能化布管方法，以突破传统基于规则设计范式的局限，推动空调风系统设计向高效、智能、协同的方向发展。

1.3 研究内容及技术路线

本研究聚焦于空调风系统管路设计，提出一种融合深度强化学习与多目标优化的风系统管路自动设计方法。本研究将传统风系统布管任务结构化拆解为走廊主管布置与房间支管布置两个子任务。具体研究围绕路径生成建模、强化学习策略训练、系统水力优化三个方面展开，研究内容如下：

（1）支管路径生成任务建模

本研究将房间内支管路径生成问题建模为路径规划任务，并从路径规划的角度对问题进行结构化定义。将布管问题形式化为路径规划问题，首先明确路径生成的目标对象、空间定义、风口连接关系与管道连通约束；其次设计状态空间，将房间空间、风口位置、当前连接状态等信息编码为输入状态；动作空间则采用方向+步长的组合方式，描述管道生成过程；基于初始投资和运行费用建立奖励

函数，使智能体能够通过与环境交互学习最优布管策略，为后续策略模型训练与环境生成奠定基础。

（2）支管路径自动生成策略训练方法设计

在结构化建模的基础上，设计支管路径策略的学习框架，构建适用于路径规划问题的策略网络结构。结合输入状态特征，提取空间信息，并构建用于动作决策的策略输出头。选择合适的深度强化学习算法进行策略训练，并通过合理的超参数优化机制，提高训练效率与策略收敛性，实现支管路径自动生成的高效、稳定学习机制。

（3）风系统管路水力性能的多目标优化

在路径生成完成的基础上，进一步构建风系统水力性能的多目标优化模型。以系统管道总面积、路径最大阻力与阻力均衡性为优化目标，建立管径配置优化流程。采用非支配排序遗传算法（NSGA-II）进行解空间搜索，获取帕累托最优解集，从而在系统运行能效、成本与风量平衡之间实现性能协同。最终形成一套兼顾几何可行性与系统性能的风系统管路优化设计方法。

本研究的技术路线如图 1.3 所示。

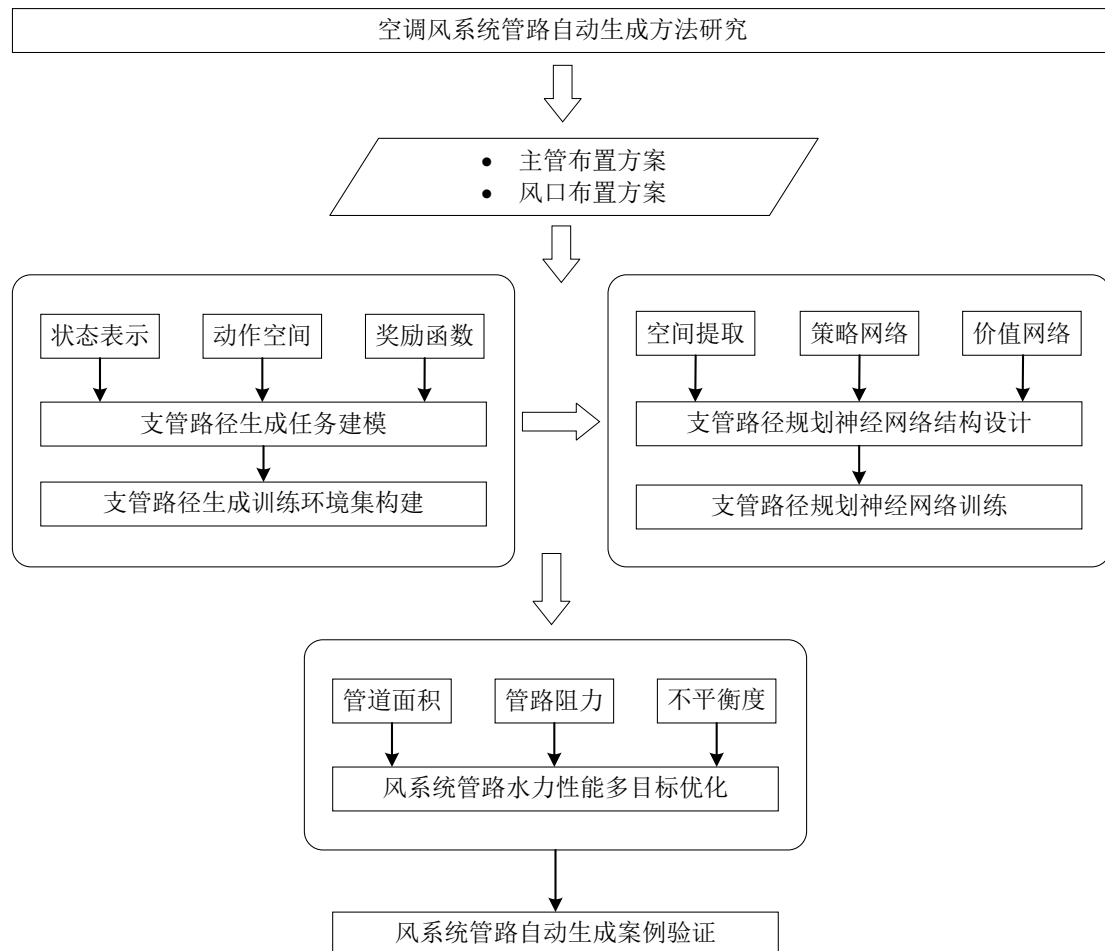


图 1.3 课题技术路线图

本论文共分为七章，各章内容安排如下：

第一章介绍了研究的背景与意义，综述了国内外在建筑设计自动化、路径规划算法及强化学习方法方面的研究现状，明确了本研究的主要内容与技术路线。第二章对风系统管路设计任务进行了结构化拆解，分别介绍了走廊主管布置方法与风口布置规则。第三章围绕房间支管路径规划问题展开建模，提出了路径生成任务的形式化表达方法，并构建了多样化的路径训练环境集，以支持后续策略学习。第四章详细介绍了路径生成策略的网络结构设计、训练方法与参数配置。第五章构建了基于管道面积、系统阻力与阻力平衡度的多目标优化模型，提出了风系统水力性能优化的建模方法与求解流程，优化生成管道的管径配置。第六章选取实际建筑场景进行系统验证，展示了自动布管与优化结果，评估所提出方法的适用性与工程可行性。第七章对本研究的工作进行了总结，梳理了主要研究成果，并对未来在建筑系统智能设计方面的研究方向进行了展望。

第2章 风系统管路生成输入准备

2.1 概述

本章旨在根据风系统路径生成任务的特征对任务进行拆解，并完成路径规划输入的准备工作的。

建筑平面图通常具有较大的空间尺度，其面积范围可从数百至数千平方米不等。若直接在完整的平面图范围内进行管线路径搜索，不仅会显著扩大搜索空间，还将造成路径规划计算的效率下降与资源消耗增加。因此，为了降低求解难度、提升计算效率，在路径生成之前有必要基于建筑空间特征对管道生成任务进行合理的结构化拆解与局部建模。

在典型建筑布局中，走廊通常作为连接多个独立功能房间的公共交通空间，人员需经由走廊进入各个房间。因此，从功能逻辑与空间布局的角度出发，可将管路系统划分为主管与支管两个层级：主管用于从空调机房延伸至走廊并沿走廊布置，作为系统的主气流输送路径；支管则从主管延伸进入各个房间，实现对末端风口的连接与气流输送。大多数房间的送风路径通过支管与走廊中的主干线连接。

因此，本研究将空调风系统的路径规划任务划分为两阶段子任务：走廊主管布置与房间支管布置。该问题拆解策略不仅有助于约束路径规划的空间范围，降低路径搜索的复杂度，同时也便于在后续路径建模与优化中采用分模块设计与协同优化的处理方式。

在路径规划任务拆解的基础上，在进行风管路径规划之前，还需明确风口的位置与选型。风口作为风系统的末端构件，其空间分布直接影响路径规划的目标区域与连接关系。因此，本章还将介绍风口布置的基本原则与处理方法，为后续路径建模与策略学习提供必要的边界条件与输入约束。

综上，本章将分别从主管布置规则与风口布置策略两个方面展开，完成风系统自动设计前的任务拆解与输入准备工作。

2.2 走廊主管布置方法

在暖通空调系统设计中，主管道作为空气输送的主干通道，承担着将空调气流从设备井输送至各功能房间的重要任务。在实际建筑工程中，主管道通常沿走

廊区域布设，既便于施工与后期维护，也有助于与房间支管进行集中式连接，形成清晰高效的送风网络结构。尤其在公共建筑、办公楼和住宅楼等多房间场景中，走廊往往作为人员流通与功能分区的中枢通道，其沿线区域具备良好的管道布置条件，是暖通系统主管优先布设的空间位置。

建筑平面图中走廊空间常呈细长、窄幅形态，若直接在走廊全域内进行路径搜索，将显著增加计算复杂度，导致规划效率低下且生成路径可控性差。因此，为提升路径生成效率、简化计算模型，本研究将主管道布设的求解范围限定于走廊中心线区域，使用一种基于多边形分割的走廊中心线提取方法，用于快速获取路径规划的主通道骨架。

该方法主要基于对走廊几何轮廓的分段处理与中心线建模，能够从不规则走廊结构中提取出连续、可连接、工程可行的主干路径，如图 2.1 所示。

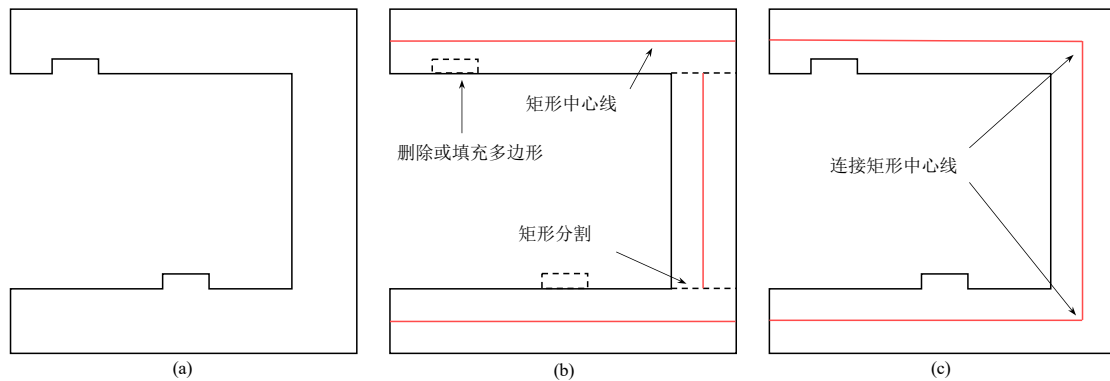


图 2.1 走廊中心线生成示意图

具体包括以下步骤：

步骤 1：走廊边界的规范化处理。在实际建筑设计中，走廊虽在整体上呈现为若干矩形通道的组合，但其边界往往受到结构柱、墙体厚度、门洞或家具布置的影响，存在凹角、突起或缺口等几何不规则性。这些细节将对路径规划的分割稳定性与中心线连续性造成不利影响。因此，首先需对走廊边界轮廓进行预处理操作，采用填充、修补等方式规整其几何形态，将其转换为近似规则的封闭多边形，以便于后续的图形分割与中心路径生成。

步骤 2：走廊区域的多边形分割。对处理后的走廊边界轮廓进行多边形分割，主要方法为在凹点处采用边界延长线进行分割操作。

对于一个顶点按照逆时针顺序排列的多边形，对每一个顶点 v_i ，其前一个顶点为 v_{i-1} ，后一个顶点为 v_{i+1} ，通过判断其叉积的正负判断该点是凹点或是凸点，如式(2.1)：

$$u \times v = (v_i - v_{i-1}) \times (v_{i+1} - v_i) \quad (2.1)$$

若叉积大于 0 则该点为凸点（外角）若小于 0，则该点为凹点。每个凹点均由两条边界边构成，延长线存在两个可能方向，通常在标准走廊形态下分割结果较为一致，因此可任选其中之一执行。通过该方式可将整个走廊区域划分为一系列几何规则、结构独立的矩形片段。

步骤 3：局部中心线提取。考虑到走廊区域大多沿长边方向展开，为提高中心线的代表性与几何合理性，提取每个分割矩形的长边方向作为中心线，作为该片区内的主管路径候选线段。中心线提取遵循对称性原则，确保所生成路径在空间分布上保持居中布设。

步骤 4：整体中心线拼接与路径构建。将相邻分割区域的中心线段进行拓展与连接：若相邻线段具有交点，则延伸连接；若两段互为平行，则插入垂直连接线以保持路径连通性。最终构建形成一条完整、连续且结构清晰的走廊中心线，可作为主管布线的计算主路径。

步骤 5：中心线偏移与避障修正。为提升路径的工程可实施性，避免路径穿越障碍物或设备空间，结合 BIM 模型的障碍信息，对路径与实际建筑结构之间的冲突关系进行检查与调整，并对生成中心线进行适度偏移处理，使其更靠近房间墙体一侧布设，从而确保布管方案在物理空间中具有良好的可行性与施工条件。

通过上述方法提取的中心线，不仅有效规避了对整张平面图的大规模搜索，同时也为路径生成任务提供了清晰的主干结构基础。该中心线区域将在后续主管路径生成、水力计算与管径优化中作为计算域输入。

2.3 风口布置算法

在风系统设计中，风口作为将经过主管道输送的空气最终分配至各个房间的重要末端构件，其合理布置直接影响到室内气流分布、系统均衡性及能效水平。鉴于不同房间功能和使用要求对气流量、压力分布等存在差异，风口布置不仅要满足设计规范，更需兼顾各区域的舒适性与经济性。本研究中采用基于扫描的风口布置算法^[97]，自动确定风口的最优位置和数量，为整体风系统设计提供准确、高效的输入。其核心思路在于利用扫描线遍历建筑底面多边形确定风口安装位置，根据房间负荷对风口进行选型，并结合经验参数对布置方案进行动态调整，其示意图如图 2.2 所示，流程图如图 2.3 所示。下面详细介绍该算法的主要步骤与关键原理。

（1）空间参数提取与参考基准确定

首先，对待布置房间的底面进行处理，提取其多边形轮廓，并通过最小外接正交矩形的求解，获得房间在建筑朝向正交轴系下的长边 L_x 及短边 L_y 长度。同

时,为避免不规则房间在重心计算中出现偏差(例如L型或回型房间的重心可能位于房间外部),采用房间内部的内心作为房间的基准点。这种方法在能够准确反映房间的有效使用面积的同时,还为后续扫描线方向的确定提供了参考。

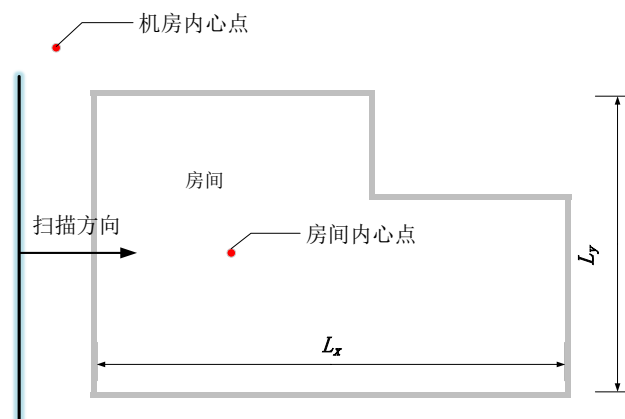


图 2.2 风口布置算法示意图

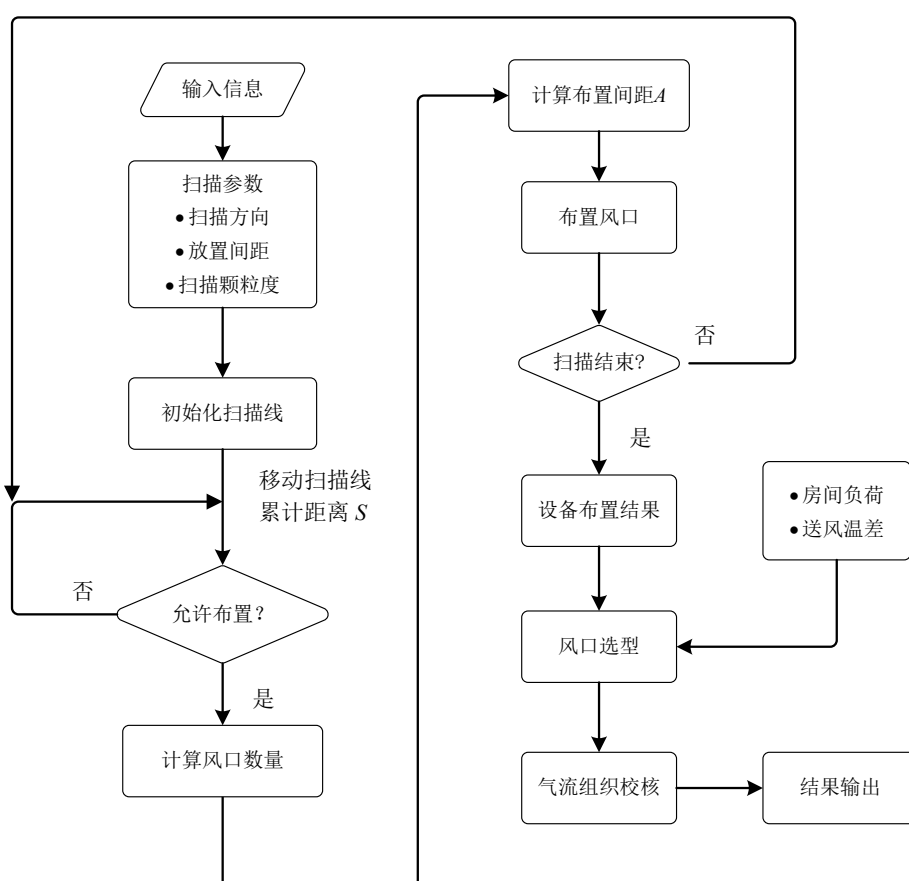


图 2.3 风口布置算法流程图

(2) 扫描线生成与有效扫描区确定

基于建筑朝向,将房间的最小正交外接矩形作为基本空间单元,沿着预设的

扫描方向生成扫描线，通常将建筑正交轴方向预设为扫描方向。扫描线定义为沿固定方向平行移动的直线，其作用是对房间底面多边形进行扫描。扫描线的长度为沿垂直于扫描方向的最小外接正交矩形边长乘以一个扩展系数，例如 1.1，确保能够覆盖整个房间实际可布置区域。截取与房间边界重合的区域作为有效扫描线。扫描线沿扫描方向移动的步长为可调参数，即扫描颗粒度，设定为 0.1 m，以保证在扫描过程中逐步累积距离，便于后续判断布置条件。

(3) 扫描过程与风口布置判断

在有效扫描线上，从起始点开始，每移动一个预设步长，累计的扫描距离 S 随之增加。布置间距 A 是一个预先定义的参数，例如对于吊顶式散流器， A 的值通常控制在 3~5 m 之间。当累计扫描距离首次达到预设距离的一半，即 $A/2$ 时，触发布置信号，此时开始检查当前扫描线经过的位置是否满足布置条件，例如是否存在障碍物、距离墙体或其他设备足够安全。在接收到允许布置信号后，根据当前扫描线在房间多边形内的具体位置，确定是否可以布置送风风口。特别地，在处理实际工程场景时，需考虑走廊、柱体等建筑构件可能对扫描线连续性的影响；当扫描线因局部空间限制无法连续达到预定布置距离时，需要动态调整布置间距，并继续扫描直至找到满足条件的布置位置。

在布置时，需要根据当前有效扫描线长度和对应方向的布置间距来确定布置的风口个数，再根据实际布置的风口个数来更新布置间距的值。布置完成后，扫描累计路程 S 被清零，直到达到所设置的布置距离 A ，该扫描线上才能重新变成允许，进行下一次布置。

在水平平面内确定风口位置后，依据房间净高和预留的吊顶距离，计算出送风风口的安装高度。

(4) 风量计算与风口选型

在确认风口的布局之后，需要根据房间负荷和设计温差来计算房间内的送风量，并以此为依据对风口进行选型。在稳态设计条件下，送风量按照焓差进行计算，在实际工程中，可以按照式(2.2)进行计算。

$$Q = \frac{3600L}{\rho c_p \Delta T} \quad (2.2)$$

式中： Q 为房间的总送风量， m^3/h ； L 为房间的负荷， W ； ρ 为空气密度，取 $1.2 \text{ kg}/\text{m}^3$ ， c_p 为空气比热容，取 $1005 \text{ J}/(\text{kg} \cdot \text{K})$ ， ΔT 为室内外空气的焓差对应的温差。

得到房间的总送风量之后，根据风口数量得到单个风口的理论送风量，再根据表 2.1 风口选型样本，筛选出风量符合要求的风口型号，并计算得到每一个风口的风速。

表 2.1 风口选型样本

序号	长/mm	宽/mm	风管高度/mm	风管宽度/mm	最大风量/m ³ /h	最小风量/m ³ /h
1	200	150	150	200	540	324
2	350	200	200	350	1260	756
3	400	300	300	400	2160	1296
4	600	400	400	600	4320	2592
5	700	450	450	700	5670	3402
6	800	500	500	800	7200	4320
7	950	650	650	950	11115	6669
8	1100	750	750	1100	14850	8910
9	1250	850	850	1250	19125	11475
10	1350	900	900	1350	21870	13122
11	1500	1000	1000	1500	27000	16200

(5) 气流组织校核

为了确保布置方案既满足气流组织要求,也能达到工程可行性要求,采用气流组织检查,对布置结果进行验证。以房间内平均风速经验公式为依据,计算公式如式(2.3)。

$$v_{pj} = \frac{0.38 \ln W}{\sqrt{W^2 / 4 + H^2}} \quad (2.3)$$

式中: v_{pj} 表示室内平均风速, m/s; W 为房间长度, m; H 为房间净高, m; n 为射程与房间长度之比。

根据风速设计要求, 分别需满足夏季 $v_{pj} < 0.3\text{m/s}$ 和冬季 $v_{pj} < 0.2\text{m/s}$ ^[98]。若测算得出的风速超出要求, 在预设的调整范围内对扫描线参数及放置间距 A 进行动态调整, 并重新布置直至满足设计规范。

2.4 本章小结

本章围绕风系统管路设计任务的拆解与输入准备展开讨论, 通过对建筑平面图中走廊与房间功能区域的剖析, 本研究将整体管线布置任务有效划分为走廊主管布置和房间支线布置两个子任务, 从而大幅缩小了路径规划的搜索空间, 提高了计算效率。随后, 针对主管道布置, 本章详细介绍了基于多边形分割的走廊中心线提取方法, 包括走廊边界的规范化处理、多边形分割、局部中心线的生成以及中心线拼接等步骤, 从而形成了一条连续且工程可行的主管路径。此外, 本章介绍了基于扫描的风口布置方案, 实现了从空间参数提取、扫描线生成、风口布置选型到气流组织校核的自动化流程, 为后续的空调风系统整体设计提供了准确且高效的末端设备分布方案。

第3章 风系统支管路径生成问题建模与环境构建

3.1 概述

本章主要围绕支管路径生成问题展开,旨在将房间内支管布置任务形式化为路径规划问题,明确划分状态空间、动作空间及奖励函数,并对关键设计约束进行建模,并构建一个可扩展的训练环境,为后续智能策略的学习与验证提供基础。基于实际建筑空间的多样性,构建包含多种典型房间布局的训练环境集,反映复杂多变的实际应用情况,为智能体在不同情景下学习有效路径策略提供数据支撑。

3.2 风系统支管路径生成问题建模

支管路径规划是实现风系统自动化布管设计的关键步骤之一,其目标是从房间入口连接至送风口,生成一条满足连通性、工程可行性与气流组织要求的路径。在实际设计中,房间几何结构不规则、障碍物分布复杂,使得路径搜索成为一个高维、约束多的优化问题。

为实现高效可扩展的建模与求解,本研究将支管路径规划问题离散化为基于网格的强化学习路径规划问题。通过将房间投影为二维网格,融合建筑空间属性、障碍物、风口与入口分布等信息,将设计任务形式化为状态-动作交互过程,并引入奖励机制指导路径策略学习。

3.2.1 空间网格表示

在将房间内支管路径生成问题转化为一个路径规划问题的过程中,空间状态的表示起着至关重要的作用。由于建筑平面图本质上是一种连续且常常具有不规则形状的空间描述,为了使得复杂的环境信息能够被有效地捕捉和处理,同时使强化学习等方法能够在离散状态空间上进行高效探索,本研究采用了空间网格表示。

首先,以房间最小外接矩形为边界,将房间在二维空间中离散化为若干固定尺寸(例如 0.1 m)的网格单元,每个网格表示一个局部空间区域。这些空间信息构成了路径规划中环境状态的基础输入,如图 3.1 所示。

为提升状态信息的表达能力与可学习性,本研究将每个网格单元的属性表示为一个多维通道向量,采用热独编码方式进行表征。热独编码是一种常用的将类

别型数据转换为数值型数据的离散特征向量化方法，其主要原理是，对于 N 个类别，将每个类别表示为一个长度为 N 的向量，其中所属类别的位置为 1，其他位置为 0。对于本研究的空间网格表示，每个空间的状态分为 7 个类别，分别是当前所在位置、风口位置、管道路径、管道入口、可通行的房间区域、柱子或其他障碍区域以及房间外空间，如图 3.2 所示。其中管道入口定义为房间底面多边形与走廊多边形的相交边，风管通过这个相交边从走廊进入房间内。

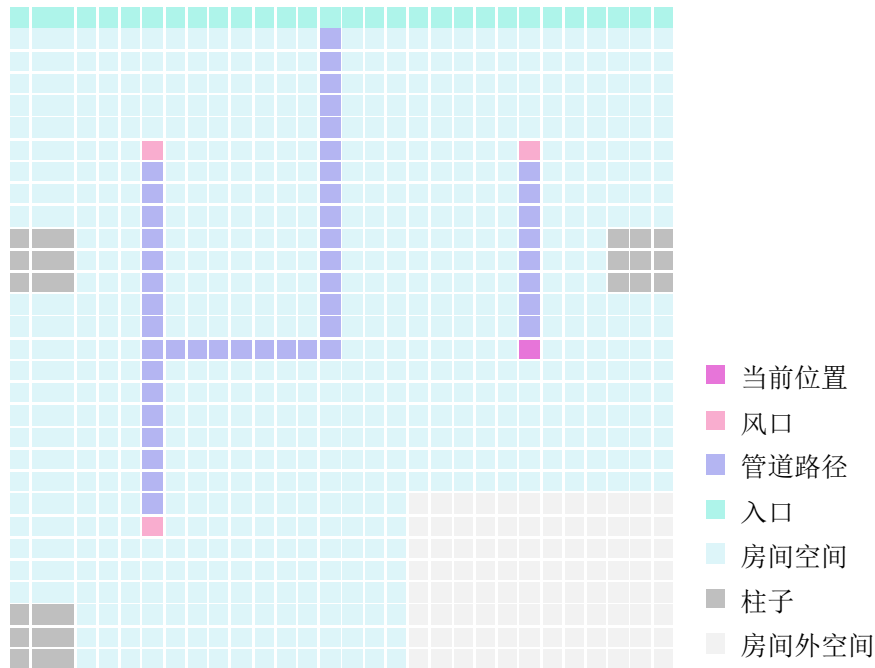


图 3.1 空间网格表示示意图

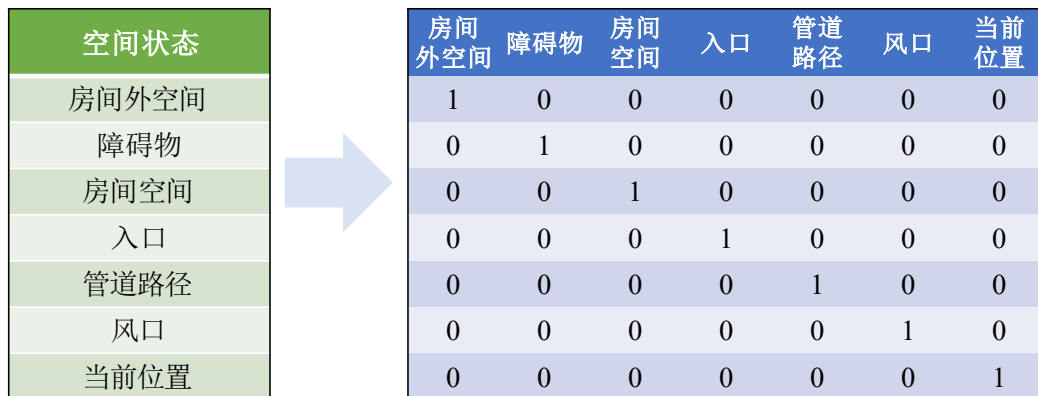


图 3.2 热独编码示意图

热独编码使得每类空间元素（如入口、风口、障碍、路径等）在张量中具有明确、互不混淆的通道表示，有利于神经网络准确识别不同类型的空间信息，避免数值重叠带来的歧义。多通道表示形式可直接输入至卷积神经网络，允许模型利用局部空间特征进行高效特征提取与策略学习。并且热独编码结构可灵活扩展，

具有良好的兼容性和可维护性。

3.2.2 状态空间建模

在将风系统支管路径生成问题形式化为智能体决策过程时,构建合理的状态空间表示至关重要。状态空间应充分表达建筑空间的拓扑结构、障碍分布、设计目标位置以及智能体的当前位置等信息,同时保证表达结构对神经网络输入友好,具备可扩展性与通用性。

因此直接将空间网格表示张量作为状态空间表示,并且为让策略网络感知到风口的风量信息,对每个网格点的状态向量进行扩展,进一步引入风量信息作为状态的一部分。即在原有的7维热独编码基础上,新增一维用于表示送风口的设计风量,从而构成一个8维状态向量。其中前7维为标准热独编码表示的该位置的类别信息,第8维为表示风口风量大小连续值。在风口处,该维度存储设计风量值,其余位置设为0。

该结构既保留了空间语义的清晰性,又引入了工程参数信息,通过风量维度使策略网络能够识别不同送风口的重要性,增强对设计目标的感知能力;采用张量结构表示整个房间状态,可直接作为深度神经网络输入,支持批处理和空间卷积,保持网络输入结构友好。

将所有网格单元的状态向量组合形成一个三维状态张量作为智能体决策的输入,动态随路径更新而变化,如式(3.1)所示。

$$S_t \in \mathbb{R}^{H \times W \times C} \quad (3.1)$$

式中 H 和 W 分别为栅格在垂直与水平方向上的离散单元数量, C 为通道数量,这里为8。

3.2.3 动作空间建模

在基于强化学习的路径规划框架中,动作空间的设计直接决定了智能体探索策略的表达能力和搜索效率。针对风系统支管路径生成任务的物理特点和网格化空间结构,本研究设计了一种混合型动作空间,结合离散方向决策与连续步长调控,以实现路径构型的灵活控制和表达能力提升。

每一个动作 a_t 表示智能体在时间步 t 所选择的移动策略,每个动作可表示为一个二元组,如式(3.2)。

$$a_t = (d_t, v_t) \quad (3.2)$$

式中: $d_t \in \{0,1,2,3\}$, 为方向变量,表示沿上、下、左、右四个正交方向中的一个; $v_t \in [0,1]$, 为步长变量,表示沿该方向移动的相对距离,为归一化的连续值,

乘以该方向长的网格总数得到实际移动长度。

连续步长的引入使路径生成具有更高的柔性与分辨率，避免单纯的离散策略在细节建模上的局限，能够生成更加平滑且工程可行的路径。步长可根据局部空间结构动态调整，例如在空旷区域快速穿越、在狭小区域精细规避，有助于提高搜索效率。所提出的混合动作空间为风系统支管路径的高效生成提供了合理策略表示与执行接口，满足实际工程中的灵活布管需求，为强化学习模型提供了良好的表达结构基础。

3.2.4 状态转移与路径生成逻辑

在基于强化学习的路径规划建模中，状态转移机制定义了智能体与环境交互后状态的更新规则，是驱动整个决策过程演化的关键。针对本研究中的空调风系统支管路径生成问题，状态转移不仅决定智能体的空间移动行为，还直接关系到管道路径的生成、拓展与终止策略的有效性，路径生成流程示意图如图 3.3 所示。

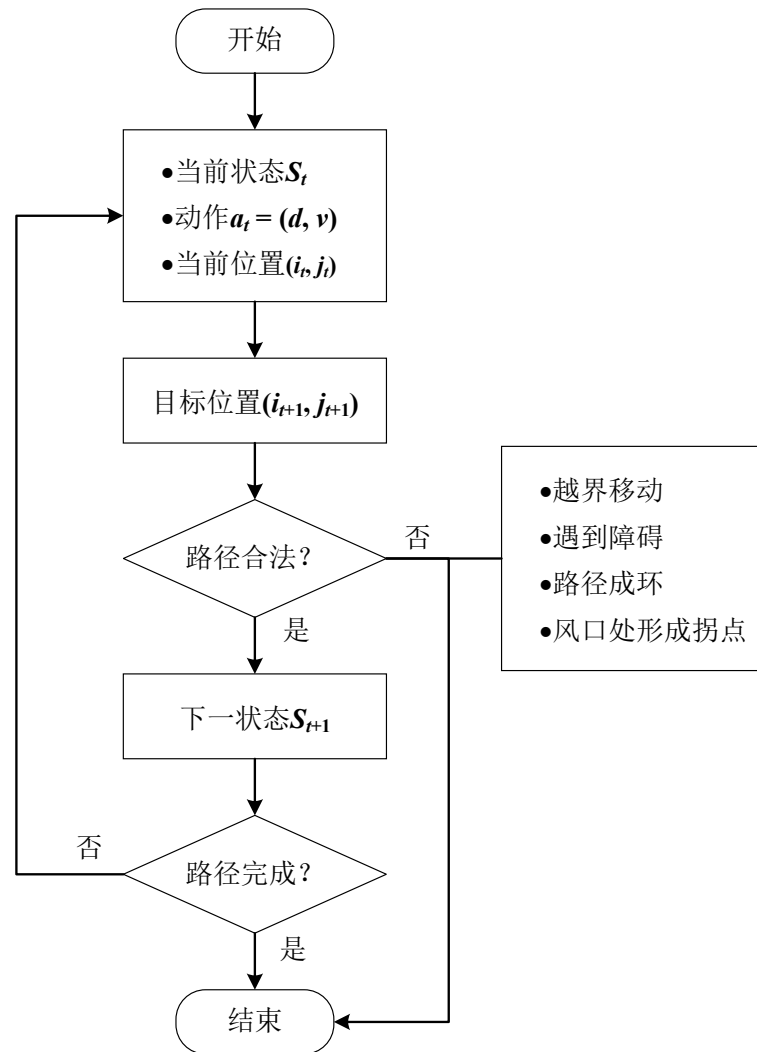


图 3.3 路径生成的流程图

设当前时间步为 t ，状态张量为 S_t ，智能体根据当前状态 S_t 选择动作 a_t ，并在环境中执行此动作，由环境计算并生成下一个状态 S_{t+1} 。

在路径生成过程中，智能体初始位置被设定在房间中距离入口最远的送风口位置。其目标是生成一条路径，最终连接至任意一个入口点。考虑到房间送风支管应独立控制，且为降低材料与能耗成本，每个房间仅允许一根支管连接至入口点。因此，在第一个风口成功连接入口后，后续风口的连接目标则切换为已构建完成的路径，形成树状结构而非网状回路。

动作执行的本质是对智能体位置的更新。若当前位置为 (i, j) ，则根据方向变量 d_t 和归一化步长变量 v_t ，计算移动后的目标位置 (i_{t+1}, j_{t+1}) ，目标位置的计算公式如式(3.3)。

$$(i_{t+1}, j_{t+1}) = (i, j) + \lfloor v_t \cdot L_d \cdot \delta_d \rfloor \quad (3.3)$$

式中： L_d 为该方向对应的最大步长，即网格总数； $\delta_d \in \mathbb{Z}^2$ 为方向索引向量，例如上为 $(-1, 0)$ ，右为 $(0, 1)$ ； $\lfloor \cdot \rfloor$ 表示取整操作。

在计算得到目标位置后，需要判断从移动的起始点到目标点路径上的所有点的状态，如果路径上的所有点都是房间内的可通行点，则更新当前位置，并把路径上的点状态更新为路径点，构成新的状态 S_{t+1} 。

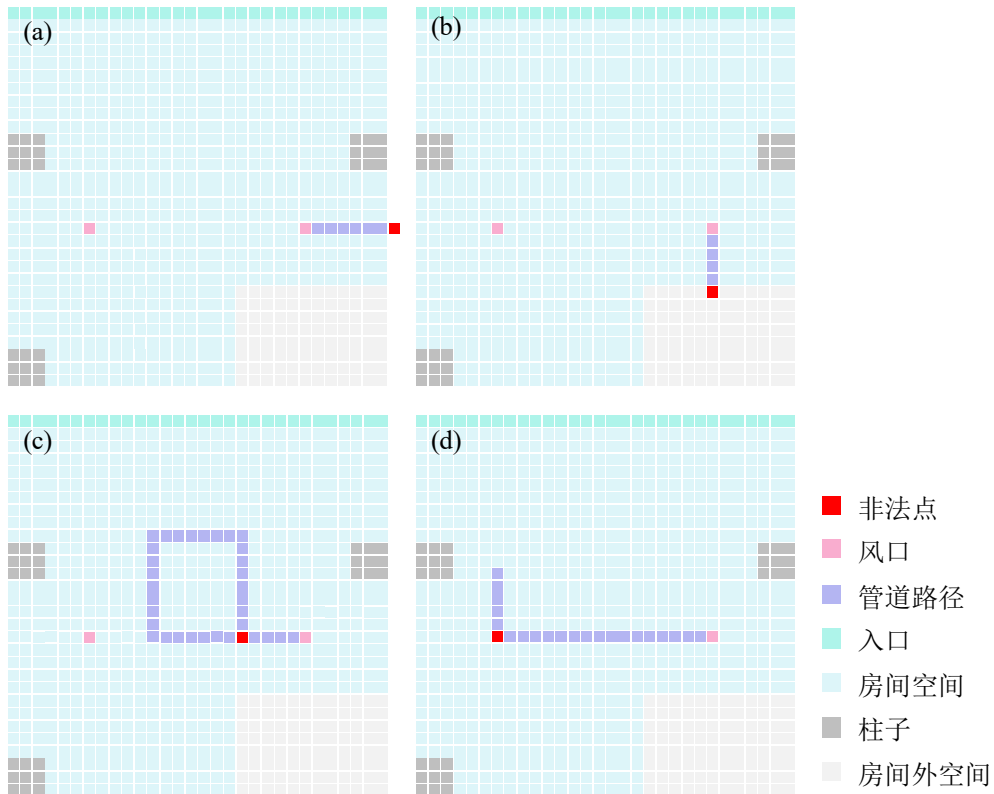


图 3.4 非法状态示意图，(a)越界移动，(b)遇到障碍，(c)路径成环，(d)风口处形成拐点

在路径构建过程中,为确保路径的合法性与工程可行性,需对动作执行过程进行边界与障碍校验。若出现以下任意情形,判定为非法状态,立即终止当前路径构建,并给予智能体负向反馈,如图 3.4:

- (1) 越界移动: 任意路径点超出状态张量边界;
- (2) 遇到障碍: 任意路径点为障碍点或房间外的点;
- (3) 路径成环: 构建的路径与其他路径形成封闭环路, 出现无效的气流输送路径;
- (4) 风口处形成拐点: 在路径转弯处连接风口, 影响送风稳定性与布置合理性。

在触发非法状态时,环境会向智能体返回较大的负奖励,并强制进入终止状态,以引导策略收敛于可行、优化的路径行为。

路径构建完成的终止条件通常包括以下几种形式,满足任意条件即视为一次路径规划尝试结束:

- (1) 成功连接所有风口, 判定所风口点均已被路径连通;
- (2) 路径达到最大步数限制, 即动作执行次数超过预设阈值;
- (3) 动作执行进入非法状态, 如遇障碍、越界、成环等不合法行为。

其中,路径被认定为成功的标准为:从设定初始点出发,所生成的路径未穿越任何非法区域,最终将全部送风口有效连通至入口或已连接主路径,且不违反任何设计约束。在强化学习训练过程中,成功路径将获得正向奖励,用鼓励策略收敛至高质量、工程可行的路径解决方案。该过程配合基于奖励信号的策略更新机制,引导智能体在复杂空间环境中学习最优布管路径,满足房间结构、设备位置及施工可行性等多重约束条件。

3.2.5 奖励函数

在强化学习框架中,奖励函数用于量化智能体在每一动作后的行为优劣,是引导策略学习的核心信号来源。一个合理的奖励函数应能够明确区分成功与失败、优与劣的路径决策,并能够在局部反馈稀疏的情况下提供适当的梯度信息,提升策略学习的稳定性和效率。

针对本文提出的风系统支管路径生成任务,奖励函数设计需同时满足以下三个目标:

- (1) 鼓励智能体生成从风口出发连接至入口或主路径的有效路径;
- (2) 惩罚非法路径行为,如越界、穿越障碍、成环等;
- (3) 在路径合法的前提下,优化路径性能指标,包括初投资与运行成本。

本研究将风系统支管路径生成奖励函数设计划分为两类:动作有效性奖励与

路径性能奖励，分别对应动作合法性判定与路径工程质量评估。

3.2.5.1 行为有效性奖励

该类奖励用于引导智能体在路径生成过程中避免错误行为、实现目标连通。其核心思想是对完成路径任务给予正向激励，对不合法或失败行为给予惩罚。

智能体从风口出发，通过序列决策生成路径，若路径成功连接至入口或既有路径网络，未触发任何非法操作，则视为一次成功路径，根据该路径所覆盖的送风口数量 N_{succ} 给予基础完成奖励，以鼓励有效连通行为：

$$R_{\text{succ}} = 4000 \times N_{\text{succ}} \quad (3.4)$$

若在路径生成过程中触发非法状态，则判定当前路径无效，立即给予固定的惩罚性奖励：

$$R_{\text{fail}} = -2500 \quad (3.5)$$

在连续步长控制机制下，为防止智能体以极小的步长反复移动或生成极短的管道而无法为配件留出空间，需要对小于阈值的步长行为加入惩罚性奖励 R_{fail} 。小步长阈值设置为当前风口所需连接管道的最小长或宽。此项惩罚抑制了无效路径扩展行为，有助于提升路径生成效率与连通稳定性。

3.2.5.2 路径性能奖励

在风系统路径规划任务中，系统性能通常涉及多个彼此制约的评价维度，如路径长度、管道阻力、风量分配均衡性及材料用量等。这些指标在物理属性和度量方式上差异显著，难以在同一评价体系下实现统一度量与权衡。例如，路径长度主要影响风管材料和施工成本；管道阻力决定系统能耗与风机负载；风量分布的不均衡性则可能导致需要配置额外的调节装置，增加局部成本并影响运行稳定性。

为解决上述异质指标融合优化的问题，本文提出一种成本统一建模策略，将多个关键性能指标转化为具有工程经济意义的费用指标，以风系统全生命周期成本（Life Cycle Cost, LCC）作为路径优劣的综合评价指标，并据此构造强化学习中的路径性能奖励。

具体地，路径性能成本被拆解为两部分：初投资成本和运行能耗成本，分别对应系统的施工阶段与运行阶段。此类设计不仅提升了奖励函数的可解释性和实用性，也使策略优化过程具有工程经济导向。

（1）初投资成本

风管系统的成本通常按照风管的展开面积计算^[99]，同时为了评估系统平衡性

的影响，在初投资成本中加入因风量不均衡带来的调节装置成本。

风管成本的计算公式如式(3.6)所示。

$$C_{\text{duct}} = \sum_{i=1}^n \rho_{\text{duct}} \cdot A_i \cdot L_i \quad (3.6)$$

式中： ρ_{duct} 为单位面积管道的成本，取 160 元/m²； A_i 为第 i 段管道的展开周长，m； L_i 为第 i 段管道的长度，m。

当系统中由于路径差异导致风量分布不均时，需在部分支路配置调节阀进行气流调节以保证末端送风的均衡性。

调节阀的安装成本依赖于其所连接风管的水力直径。本研究基于工程样本数据，采用多项式回归，根据管道的水力直径对调节阀的成本进行拟合，计算公式如式(3.7)。

$$C_{\text{valve}} = \sum_{i=1}^m a D_{h,i}^2 + b D_{h,i} + c \quad (3.7)$$

式中： $D_{h,i}$ 为第 i 段管道的水力直径，m； a ， b ， c 为经验拟合系数，根据造价样本数据，如图 3.5 所示，取 $a = 320.76$ ， $b = 256.62$ ， $c = 75.588$ 。

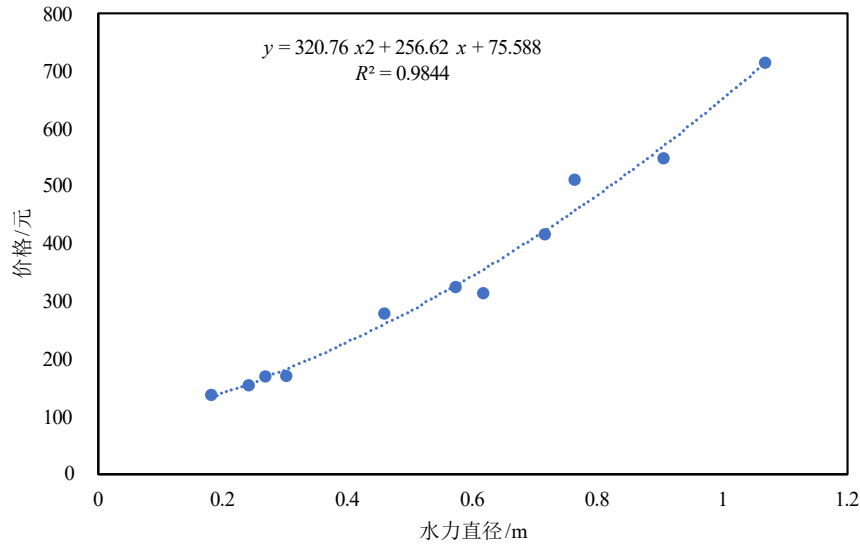


图 3.5 调节阀成本拟合结果

则最终初投资总成本为：

$$C_{\text{inv}} = C_{\text{duct}} + C_{\text{valve}} \quad (3.8)$$

(2) 运行能耗成本

运行成本主要来源于风机在全年运行过程中为克服管道和配件阻力所消耗的电能。运行能耗由系统阻力和风机效率决定，全年运行电能成本计算公式见式

(3.9)。

$$C_{\text{ope}} = \frac{Q \cdot P}{3600 \cdot \eta_f \cdot \eta_m} T \cdot \rho_{\text{electricity}} \cdot \tau \quad (3.9)$$

式中： Q 为管道或配件中通过的风量， m^3/h ； P 为管道或配件的压力损失， Pa ； η_f 为风机效率，取 0.75； η_m 为电机效率，取 0.9； T 为全年运行时间，取每年 3000 小时； $\rho_{\text{electricity}}$ 为电价，取 0.7 元/kWh； τ 为运行年限，取 20 年。

在强化学习奖励函数中，将最终路径性能总成本的负值作为回报信号，如式 (3.10)。

$$R_{\text{path}} = -(C_{\text{inv}} + C_{\text{ope}}) \quad (3.10)$$

此奖励机制使得强化学习智能体在实现路径连通的同时，能主动学习在不同工程情境下的投资和运行折中策略，以获得更优的设计质量与经济性。

3.3 风系统支管路径生成训练环境集构建

为训练具备泛化能力的强化学习模型，需构建多样性丰富、结构合理的训练环境集。本文通过随机生成房间几何、多种障碍物与边界配置，构建出面向风系统支管路径规划任务的二维训练样本集。训练环境以二维多边形表示房间平面轮廓，包含如下三个组成部分：房间空间轮廓、内部结构障碍（如柱子）、可达入口边界。通过系统化的空间采样机制，实现了具有代表性的房间形态和布置模式，为策略网络提供丰富的结构信息输入与路径探索挑战。

3.3.1 训练环境生成方法

为训练具备泛化能力的强化学习模型，需构建多样性丰富、结构合理的训练环境集。本研究基于工程现实中的建筑空间特征，设计了一套自动化的房间样本生成流程，用于支管路径生成训练。每一个房间实例均包含以下四个要素：房间轮廓多边形、结构障碍（柱子）、入口段以及风口布置，伪代码如表 3.1 所示。

主要步骤如下：

(1) 房间轮廓多边形生成

房间轮廓的几何形状决定了智能体路径规划的空间约束。本文设计了三类典型房型生成方式，包括矩形、L 型和 T 型，并通过概率加权随机采样获得多样形状组合，矩形、L 型和 T 型房间生成的概率分别为 60%、20% 和 20%。具体流程如下：

1. 尺寸采样与约束检查：通过设定长宽范围与最大长宽比，对主区域与延

- 伸区域分别进行尺寸采样，确保生成几何满足建筑常规构造逻辑。
2. 组合与合并：将主区域和附加部分（L型转角、T型侧翼）合并为完整房间形状；
 3. 几何清洗与正交性验证：通过为几何实体加入缓冲区，清洗可能存在的几何异常（如交叠、碎裂），并对边界线方向进行逐一判断，确保边界为水平或垂直线段，满足建筑设计的正交性要求；
 4. 镜像变换：为增强训练样本的空间多样性，对生成形状随机进行水平、垂直或双向镜像操作。

表 3.1 环境样本集合生成伪代码

输入：样本数量 N ，最大尝试次数 $MaxAttempts$
输出：环境样本集合 $EnvSet$
<pre> 1: $EnvSet \leftarrow \emptyset$ 2: for $i = 1$ to N do 3: $ShapeType \leftarrow$ 从 $\{Rectangle, L_shape, T_shape\}$ 中按概率随机采样 4: if $ShapeType == "Rectangle"$ then 5: $w, h \leftarrow$ 随机采样宽高（范围为 $[3.0, 12.0]$，长宽比 ≤ 3.0） 6: $RoomPolygon \leftarrow$ 生成矩形：四点 $(0,0)-(w,0)-(w,h)-(0,h)$ 7: else if $ShapeType == "L_shape"$ then 8: 主段尺寸 $\leftarrow$ 在长宽比限制下采样主段 $w_1, h_1 \in [3.0, 8.0]$ 9: L型臂尺寸 $\leftarrow$ 根据剩余空间采样 $w_2 \in [3.0, 12.0 - w_1], h_2 \in [3.0, 12.0 - h_1]$ 10: $RoomPolygon \leftarrow$ 主段+臂段 构造 L 型多边形 11: else if $ShapeType == "T_shape"$ then 12: 主厅尺寸 $\leftarrow w_1, h_1 \in [3.0, 8.0]$ 13: 侧厅尺寸 $\leftarrow w_2 \in [3.0, 12.0 - w_1], h_2 \in [3.0, 12.0 - h_1]$ 14: $RoomPolygon \leftarrow$ 主段+侧翼段 构造 T 型多边形 15: end if 16: $RoomPolygon \leftarrow$ 几何清洗，正交检查，随机镜像 17: $R, W \leftarrow$ 随机偏移量 $\in [0, 1]$，柱子边长 $\in [0.5, 0.8]$ 米 18: $S_x, S_y \leftarrow x/y$ 方向柱网间距 $\in [6.0, 10.0]$ 米 19: $Columns \leftarrow$ 基于规则柱网遍历 $RoomPolygon$ 内部空间，并去除重叠或越界 20: $SelectedEdge \leftarrow$ $RoomPolygon$ 的所有边按边长加权随机采样一条边 21: $Entry \leftarrow$ 从 $SelectedEdge$ 随机截取长度大于 2 的线段 22: $Diffusers \leftarrow$ 生成风口 23: $Env \leftarrow \{polygon: RoomPolygon, columns: Columns, entry: Entry, diffusers: Diffusers\}$ 24: $EnvSet \leftarrow EnvSet \cup \{Env\}$ 25: end for 26: return $EnvSet$ </pre>

（2）结构柱生成

结构柱作为典型的障碍物，显著影响路径连通性与布局复杂度。本研究在每个房间内自动生成若干规则柱体，具体流程如下：

1. 生成参数采样：包括纵横方向的柱间距（6~10 m）、柱网起始位置（根据间距随机偏移）、柱子边长（0.5~0.8 m）等均从随机区间中抽样获得。
2. 布置逻辑：以规则网格方式对房间内部进行遍历布置，排除穿出房间边界或与房间边界贴近位置，确保柱体均位于有效空间内部。
3. 集合输出：将所有符合条件的柱子以 shapely 库的 Polygon 格式返回，用于后续构建空间网格时标记为障碍区域。



图 3.6 房间生成结果

(3) 入口段生成

入口定义了风系统主管进入房间的连接点，是路径规划的边界条件之一。入口生成流程如下：

1. 候选边采集：选取房间外轮廓边为候选边，计算其边界长度。
2. 有效边筛选：筛除长度小于 2 m 的边界线段，仅保留能够布置合理入口宽度的边。
3. 边段截取：在候选边上随机截取一段长度大于 2 m 的连续线段，作为入口段，以 shapely 库的 LineString 格式返回。

入口段既模拟了从公共空间（如走廊）进入房间的连接界面，也为路径生成任务提供明确的目标边界。

（4）送风风口布置

在房间底面多边形、柱子和入口段生成后，根据 2.3 节的风口自动布置算法生成风口。

按照以上的房间生成算法构建了 1000 个不同结构的房间样本，构建具有代表性和多样性的强化学习训练环境集。图 3.6 所示为部分生成样本的可视化结果，样本涵盖了矩形、L 型和 T 型等典型房间外轮廓。这些样本不仅包含房间边界信息，还包含了柱子（灰色区域）、入口段（房间边界上的线段）与风口布置（图中的点）等必要构件，确保环境生成具有结构复杂性和路径规划约束性。

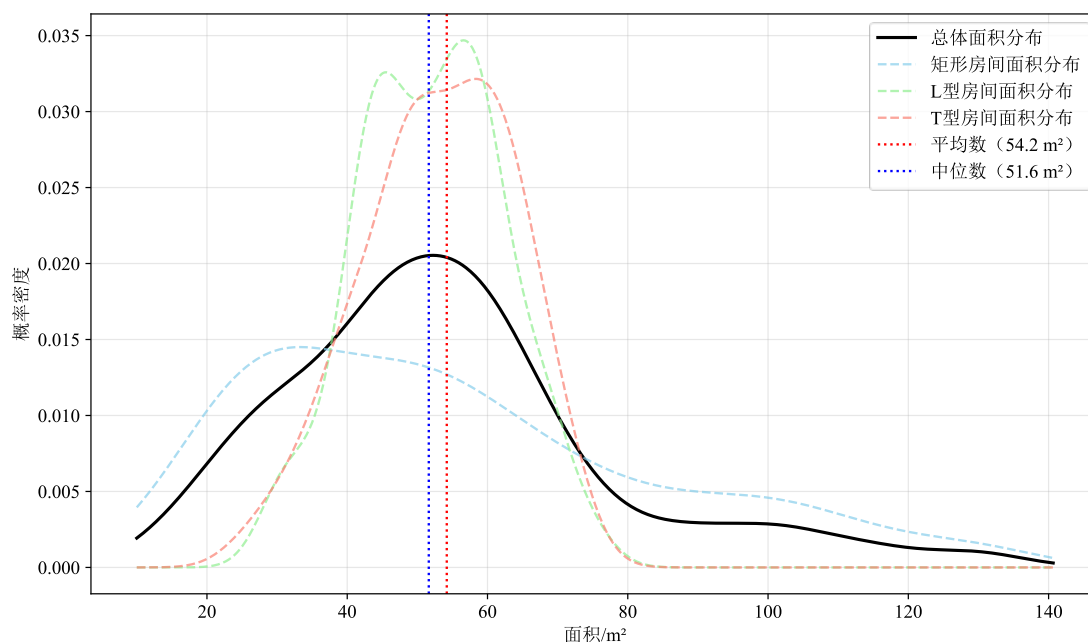


图 3.7 生成房间面积分布

图 3.7 展示了所有生成样本的面积分布统计结果。可以观察到，所生成房间面积主要集中在 $40\sim70\text{ m}^2$ 之间，均值约为 54.2 m^2 ，中位数为 51.6 m^2 。三类结构形状的面积分布上，矩形房间分布较为分散，L 型与 T 型房间的面积相对集中，

峰值位置更靠近均值。训练样本在保持几何结构可控的同时，也一定程度上体现出建筑设计中真实的房间面积分布趋势。

该训练环境集可以用于支管路径规划策略的训练与验证。通过模拟不同的建筑形状和布置特征，强化学习智能体将面临多样化的布局挑战，从而学习更具鲁棒性和适应性的路径规划策略。

3.3.2 多指标加权融合的训练环境难度评估方法

在风系统支管路径生成任务中，房间结构复杂度、风口分布密度、障碍位置等因素显著影响路径规划的难度。特别是在不规则空间、多风口分散布局、障碍密集或入口远离目标区域的情境下，路径生成面临搜索空间剧增、规划策略不稳定以及训练收敛缓慢等挑战。为了提高智能体在不同情境下的学习效率与策略稳定性，需要对环境的任务难度进行评估，以此来指导训练流程。根据这种评估，智能体可以先从简单环境入手，逐步增加任务难度，使强化学习过程遵循由易到难的学习路径，从而避免智能体陷入早期训练阶段探索失败、梯度稀疏或策略崩溃等问题。

本文设计了一种多指标加权融合环境难度评估方法，从空间复杂性、入口可达性与风口分布三个维度综合衡量样本的难度，如式(3.11)所示。该评估方法不仅能够为环境的难度排序提供依据，还可以为后续的训练策略制定提供有力支持。通过这种方式，训练过程能够根据环境的难度逐步调整，确保智能体在不同难度级别的环境中有效学习。

$$D_{\text{total}} = \gamma(D_{\text{space}} + D_{\text{entry}} + D_{\text{diffuser}}) \quad (3.11)$$

其中： D_{total} 为环境总难度； D_{space} 、 D_{entry} 、 D_{diffuser} 分别为空间复杂度、入口可达性与风口分布相关难度分量； γ 为尺寸因子修正项，计算公式为式(3.12)。

$$\gamma = 1 + \frac{\log(h \cdot w)}{k} \quad (3.12)$$

其中： h 和 w 分别为网格的高度和宽度尺寸； k 为缩放系数，取 10。

空间复杂度指标衡量房间结构与障碍物布置对路径选择造成的影响，由障碍密度 d_{obs} 与空间熵 H 组成。障碍密度定义为柱子区域所占房间面积的比例，空间熵度量空间利用率的离散性，计算公式列于(3.13)和(3.14)。

$$D_{\text{space}} = \beta_1 d_{\text{obs}} + \beta_2 H \quad (3.13)$$

$$H = -\sum_{i=1}^N p_i \log(p_i) \quad (3.14)$$

其中：房间被划分为 N 个子区块， p_i 表示第 i 个区块中房间空间的占比； β_1 、 β_2 为权重系数，取值为 0.15。

入口可达性指标衡量风口至入口的平均路径成本，计算公式如式(3.15)。

$$D_{\text{entry}} = \beta_3 d_{\text{mean}} + \beta_4 \sigma_d \quad (3.15)$$

其中： d_{mean} 为风口与入口归一化平均距离，即将所有风口至入口的最短路径长度归一化后的平均距离； σ_d 为距离标准差，衡量各风口至入口距离的波动性，反映路径分布的不均衡性； β_3 、 β_4 为权重系数，取值分别为 0.20、0.05。

风口分布指标综合考虑了风口在空间中的分布特性，评估风口的空间布局对路径规划难度的影响，评估中综合考虑分散指数、覆盖密度、数量影响和障碍邻近性四项因素，计算公式如式(3.15)所示。

$$D_{\text{diffuser}} = \beta_5 I_{\text{disp}} + \beta_6 \rho_{\text{cover}} + \beta_7 \xi_{\text{num}} + \beta_8 \phi_{\text{obs}} \quad (3.16)$$

其中： I_{disp} 为分散指数，衡量风口在空间中的离散程度，计算公式如式(3.17)； ρ_{cover} 为覆盖密度，用风口的凸包面积衡量其在空间中的覆盖范围，归一化为总房间网格面积比，计算公式如式(3.18)； ξ_{num} 表示风口数量的影响，通过 Sigmoid 函数将风口数量归一化，反映风口数量增加对路径连通复杂度的非线性影响，计算公式如(3.19)； ϕ_{obs} 为障碍邻近性，通过卷积近似每个风口邻域内的障碍密度，并取平均作为障碍影响因子，计算公式如式(3.20)； β_5 、 β_6 、 β_7 、 β_8 为权重系数，取值分别为 0.15、0.10、0.10、0.10。

$$I_{\text{disp}} = \frac{\sqrt{N} d_{\text{diff}}}{\sqrt{h^2 + w^2}} \quad (3.17)$$

$$\rho_{\text{cover}} = \frac{A_{\text{hull}}}{h \cdot w} \quad (3.18)$$

$$\xi_{\text{num}} = \text{Sigmoid} \left(N / \left\lfloor \frac{h \cdot w}{100} \right\rfloor \right) \quad (3.19)$$

$$\phi_{\text{obs}} = \frac{1}{N} \sum_{i=1}^N D_{\text{obs}}(d_i) \quad (3.20)$$

其中： N 为风口总数； d_{diff} 是风口之间欧氏距离的均值， A_{hull} 是风口点构成的最小凸包面积 $D_{\text{obs}}(d_i)$ 表示风口 d_i 周围 10×10 区域的障碍密度。

通过这种多维度的难度评估体系，能够全面衡量路径规划中的几何结构复杂性、目标可达性以及布局复杂性等关键因素。房间的面积、风口的数量、障碍物的密度等因素都会显著影响难度的高低。最终的难度指标可以为后续的训练提供

有效的度量依据，帮助智能体在不同难度级别的环境中进行合理的训练调整。

根据以上计算方法，计算了 1000 个训练样本的难度，图 3.8 展示了 1000 个训练样本计算所得的环境难度分布情况。可以看出，总体难度分布呈现近似靠近难度均值的双峰形态，表明样本集中在中间两个典型复杂度段之间。不同形状房间的难度分布具有显著差异：矩形房间分布较集中且偏向低难度区间；而 T 型与 L 型房间则展现出更高的平均难度值与更广的分布区间，体现了形状结构对路径规划任务的显著影响。

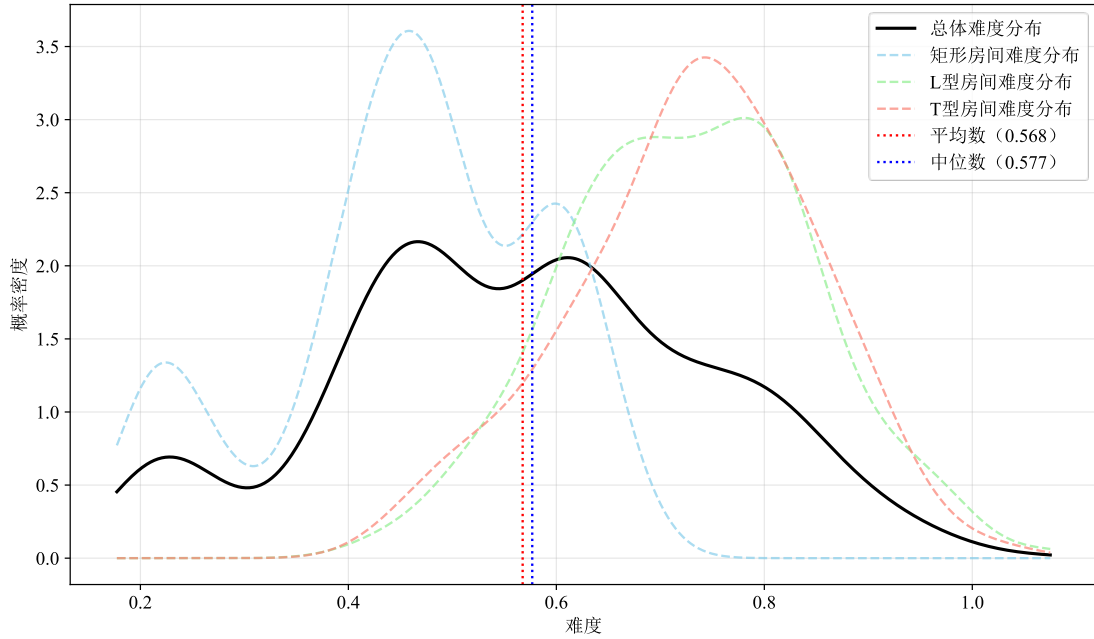


图 3.8 生成的房间难度评估结果分布

总体来看，生成的训练环境集在几何结构、目标可达性与布局复杂性方面具有丰富的多样性和层次性，为后续强化学习路径规划策略的学习与泛化提供了扎实的数据基础。

3.4 本章小结

本章围绕空调风系统支管路径生成问题，构建了基于强化学习的建模框架与训练环境，为后续策略训练奠定了基础。

首先，提出了一种基于正交网格的空间离散化建模方法，并采用热独编码和风量信息构建了具有表达性与通用性的状态表示。这一方法不仅能够有效地表达不同的空间特征，还能适应多种复杂的风系统设计需求。在动作空间方面，设计了结合离散方向与连续步长的混合动作模型，以增强路径生成策略的灵活性与表达能力。

其次，定义了清晰的路径生成流程，并构建了符合工程约束与性能评价标准的奖励函数体系，包括行为有效性奖励和路径性能奖励。其中路径性能通过初投资与运行能耗统一建模为经济指标，有效兼顾了设计合理性与能效目标。

在训练环境构建方面，本文提出了一种多种房间形态的随机样本生成方法，包括矩形、L型和T型的正交多边形构型，并包括了内部障碍物（柱子）生成、入口段提取以及风口位置生成策略，形成了一套可扩展的训练样本生成机制。同时，为提高策略泛化能力与训练效率，本章还设计了包含空间复杂度、入口可达性和风口分布特征等多指标的加权融合环境难度评估模型，通过量化评估不同环境的难度，为后续训练过程提供了科学的难度调控依据。

综上所述，本章完成了从问题建模到训练环境构建的全过程设计，建立了风系统支管路径生成任务的强化学习建模框架，并建立了具备多样性和层次性的训练环境集。该环境集具有不同难度等级，能够有效支持后续路径生成策略的训练与评估。

第4章 风系统支管路径生成策略模型训练

4.1 概述

在上一章中，本文完成了风系统支管路径规划问题建模与训练环境的构建，建立了基于空间网格状态表示与混合动作机制的智能体交互框架，并构建了具备多样空间形态与路径规划难度的样本环境库。本章将再此基础上进一步聚焦于神经网络结构设计与路径策略的学习方法。

本章首先针对风系统路径规划问题进行了深度神经网络结构设计，结合空间特征提取与策略价值联合建模需求，构建了一种具有位置感知与策略表达能力的神经网络模型。随后，选用主流的近端策略优化算法(Proximal Policy Optimization, PPO)作为智能体的策略优化方法，结合基于蒙特卡洛估计的回报与优势函数，构建了多阶段训练流程，并进行了参数调优与性能评估。

4.2 网络结构设计

针对风系统路径规划任务具有明显的空间拓扑特征与动作时序依赖性的特点，本文设计了一种融合空间卷积与时间建模的混合神经网络架构，实现对输入环境状态的有效编码，并输出用于控制的混合动作策略。整体模型采用特征共享与任务分支相结合的模块化设计思路，分为主要两个部分：共享特征提取模块、和策略输出模块。

4.2.1 共享特征提取模块

在路径规划任务中，环境状态的空间感知能力对动作选择的合理性至关重要。为有效提取输入网格中的结构信息、障碍分布与风口位置等关键元素，本文设计了一套多层残差卷积网络作为特征编码主干结构，并引入了位置感知机制以增强网络对绝对空间信息的理解能力。共享特征提取模块结构如图 4.1 所示。

原始输入状态张量通道包括风口、障碍、路径等多类空间特征。考虑到路径规划任务中的位置依赖性显著，本文采用了一种坐标增强策略，在原始输入通道中额外拼接了归一化位置编码，包括横向坐标 x 、纵向坐标 y ，以及相对于中心的径向坐标 r 。在保持输入空间结构不变的前提下，为每个像素点引入位置信息，从而提升卷积网络的空间辨识能力。

考虑到房间网格可能因为房间的绝对尺度的差异而导致形状不同,因此采用可以接受任意形状输入的全卷积网络对空间特征进行提取。特征编码主干网络由四个残差卷积模块(Residual Block)级联组成。残差卷积模块可以对输入进行下采样和空洞卷积。下采样可以压缩空间维度的同时提取初级边界与形状信息;空洞卷积可以以较少的计算代价扩大感受野,增强模型对全局结构的感知能力。卷积后输出为通道数固定为256的中间特征。

为进一步压缩特征表示并增强全局信息感知能力,网络在残差模块输出后接入自适应平均池化层,得到序列特征向量。该特征向量作为时序建模与策略生成的基础表达,不仅包含了结构语义与障碍信息,同时坐标增强的引入可以使得模型具备更强的绝对定位能力。

同时考虑到为了在训练时对多个环境的状态轨迹进行批处理,需要将不同形状的网络填充为统一尺寸,为使模型具备对房间绝对几何尺寸的感知能力,进一步引入了尺寸嵌入模块,将原始空间的宽度与高度作为额外输入,经由全连接网络映射为固定维度的形状特征,并在后续与卷积特征拼接。

风系统支管路径的生成过程是一个序列决策问题,即智能体在多步状态转移中逐步延展路径、连接风口。由于每一步的动作决策受到历史轨迹的影响。为此,本文在特征提取模块之后引入循环神经网络结构,对路径构建过程中的状态序列进行建模,提升策略的时序感知与行为连贯性。具体而言,采用结构紧凑、稳定性优异的长短期记忆网络(Long Short-Term Memory, LSTM)作为时序建模模块。该模块以上一阶段卷积特征输出为输入,将其映射至隐藏状态,并更新内部状态,以捕捉路径历史信息与当前状态之间的动态关系。

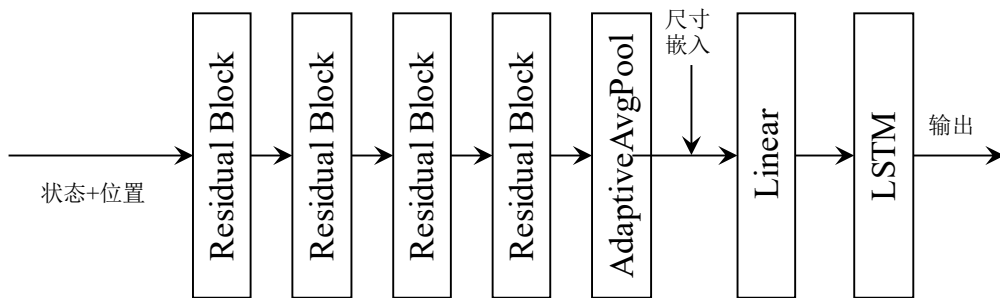


图 4.1 共享特征提取模块

4.2.2 策略输出模块

在得到时序编码后的特征表示后,本文采用经典的演员-评论家(Actor-Critic)框架,分别设计了用于策略输出的动作(Actor)模块和用于状态评估的价值(Critic)模块。Actor 网络包含两个子分支:一是离散动作头部,输出当前状态下在四个方向(上、下、左、右)上的动作概率分布;二是连续动作头部,生成对应方向

上的归一化步长的分布参数，包括均值和标准差。策略输出模块结构如图 4.2 所示。

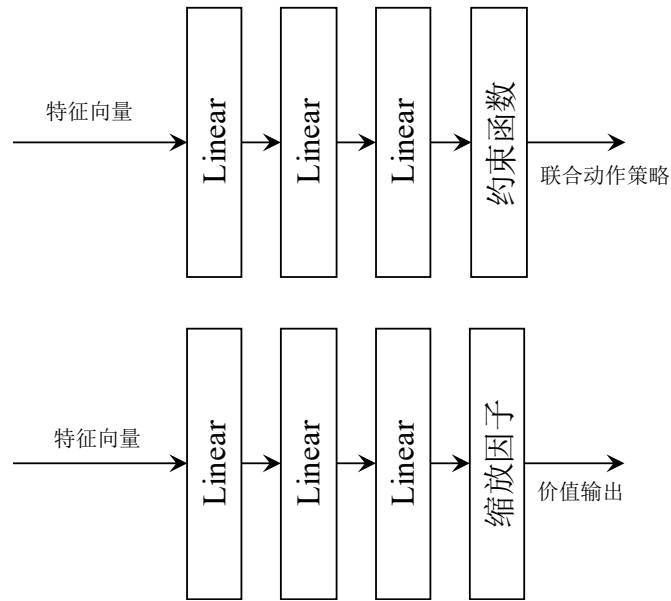


图 4.2 策略输出模块

离散动作部分通过多层感知机生成 4 维向量，经 Softmax 函数转换为概率分布。Softmax 函数是多分类任务中常用的激活函数，用于将一个向量的实数值转换为概率分布，输出每个类别的概率。对于输入向量 $z = [z_1, z_2, \dots, z_n]$ ，Softmax 函数的输出通过公式(4.1)计算。

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^n e^{z_j}} \quad (4.1)$$

连续部分则通过独立的前馈网络分别输出均值和标准差，并使用 Sigmoid 与 softplus 函数进行约束，保证其范围合理稳定，从而构成一个联合动作策略，实现对方向与步长的联合建模。Sigmoid 是一种 S 形激活函数，可以将输出值范围限制在 0 到 1 之间，计算公式为式(4.2)。Softplus 函数是一个平滑的近似 ReLU (Rectified Linear Unit) 函数，可以将确保输出为正值，并且可以有效地避免 ReLU 函数的死区问题，计算公式见式(4.3)。3 种激活函数的示意图见图 4.3。

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (4.2)$$

$$\text{Softplus}(x) = \log(1 + e^x) \quad (4.3)$$

Critic 网络则以共享的时序特征为输入，通过多层感知机输出当前状态的价值估计。该模块用于目标值计算和优势函数估计，由于不同动作环境奖励分布差

异数值差异较大，Critic 网络也是训练中强化学习梯度的主要来源。为了提升训练稳定性，Critic 输出值乘以一个可训练的尺度因子，以自适应调整预测值的范围，避免值函数过大或过小对梯度传播带来负面影响。

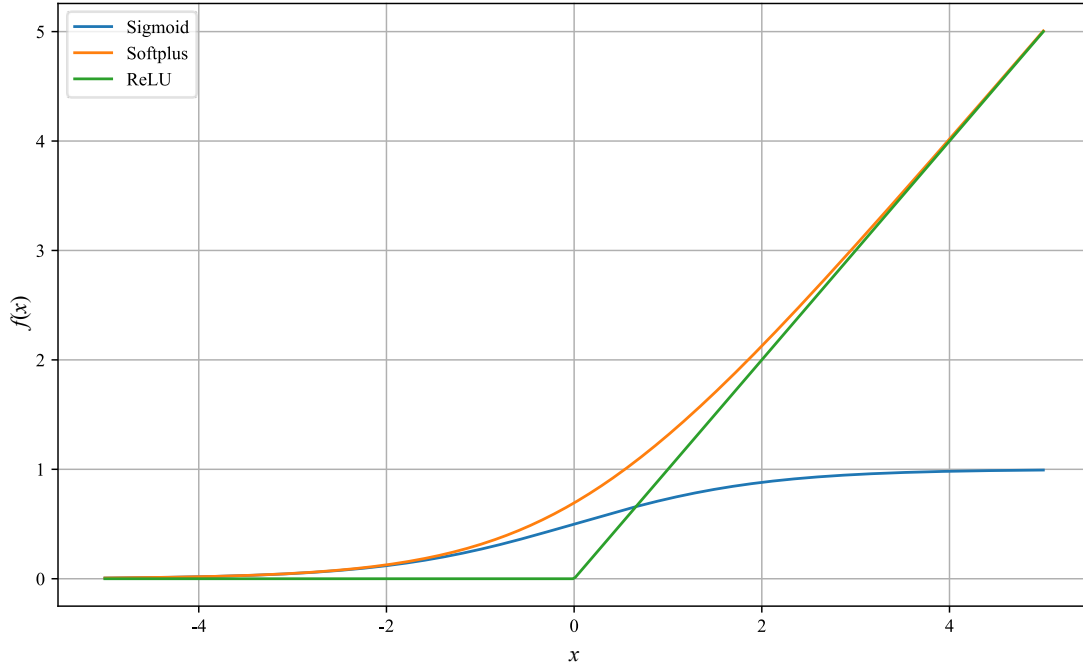


图 4.3 激活函数

4.3 强化学习算法选择与训练流程

为了实现风系统路径规划任务中高效、稳定的策略学习，采用 PPO 算法作为主要的强化学习策略优化方法。PPO 结合了策略梯度与价值函数逼近的优点，在保持训练稳定性的同时，具备较强的样本效率与泛化能力。该算法通过 Actor-Critic 结构结合蒙特卡洛回报估计与优势函数，引导智能体策略向更优方向迭代，已被广泛应用于复杂决策任务。

本节将从以下四个方面展开：PPO 算法原理、回报与优势估计方法、损失函数设计以及完整训练流程。

4.3.1 PPO 算法原理

PPO 是一类改进的策略梯度算法，它在更新策略时引入了剪切概率比率机制（Clipped Surrogate Objective），以避免策略在单次更新中发生过大的偏移，从而提高训练的稳定性。其核心思想是，通过比较当前策略与旧策略对同一动作的概率比，动态约束策略更新幅度^[100]。PPO 的目标函数为式(4.4)。

$$L^{CLIP}(\theta) = \mathbb{E}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)] \quad (4.4)$$

$$r_t(\theta) = \frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \quad (4.5)$$

其中： $r_t(\theta)$ 是新旧策略的比率； $\hat{A}_t$ 是优势函数； ϵ 是剪切参数，控制更新幅度。

PPO 通过引入剪切项来限制策略的更新。当策略变动超出剪切范围时，优化目标被限制为保守更新，有效平衡了策略改进与稳定性的关系。

4.3.2 回报计算与优势估计

强化学习中，智能体通过对未来奖励的估计进行策略更新。回报（Return）表示从某个状态开始，智能体所能获得的总奖励；而优势（Advantage）则衡量某个动作相比于平均水平的表现。通过合理的回报和优势估计，智能体能够调整策略以提高整体的长期回报。

使用蒙特卡洛方法计算每个状态下的回报。蒙特卡洛方法依赖于完整的轨迹数据。在每个轨迹结束后，根据观测到的回报来更新价值函数估计，计算公式如式(4.6)。

$$R_t = \sum_{k=0}^{T-t} \gamma^k r_{t+k} \quad (4.6)$$

其中： γ 是折扣因子，控制着未来奖励对当前决策的影响； r_{t+k} 是时间步 $t+k$ 的即时奖励。

优势函数衡量的是智能体在某状态下采取某动作所带来的实际回报相较于基准值的表现，计算公式如(4.7)。

$$A_t = R_t - V(s_t) \quad (4.7)$$

其中： $V(s_t)$ 是状态价值函数。

在实际训练过程中，优势估计的分布可能会产生较大的波动，特别是在环境较为复杂或奖励反馈不稳定的情况下。为了提高训练的稳定性和收敛速度，对优势值进行了归一化处理。未经归一化的优势值可能会导致梯度在训练过程中变化过大，进而引起参数更新的不稳定。通过归一化，优势值被拉回到一个相对较小的范围内，使得梯度更新变得更加稳定。此外，不同环境和任务的奖励分布和尺度不同，优势归一化能使得算法更好地适应各种环境。

优势归一化的公式是式(4.8)。

$$A'_t = \frac{A_t - \mu}{\sigma} \quad (4.8)$$

其中： μ 是优势值的均值； σ 是优势值的标准差。

4.3.3 损失函数

PPO 算法的损失函数由三部分组成：策略损失、值函数损失与熵损失。值函数损失旨在逼近状态的真实价值，帮助智能体估计每个状态下的长期回报，用于训练 Critic 网络。通过最小化状态价值预测值和实际回报之间的均方误差（Mean Squared Error, MSE）来进行优化，见式(4.9)。

$$L^V(\theta) = \mathbb{E}_t[(V(s_t) - R_t)^2] \quad (4.9)$$

熵损失用于鼓励智能体在策略选择时保持一定的随机性。在强化学习的训练过程中，如果智能体过早地收敛到一个确定的策略，可能会导致探索不足，陷入局部最优解。熵损失通过鼓励探索，从而确保智能体能够在训练中保持足够的探索性，计算公式是式(4.10)。

$$L^H = -\beta H(\pi) \quad (4.10)$$

其中： $H(\pi)$ 是当前策略分布的熵； β 是熵损失系数，是一个控制探索和利用平衡的超参数，通过调节熵损失系数平衡探索和利用，确保智能体在策略收敛之前能充分探索环境，学习到更为有效的策略。

结合策略损失、值函数损失和熵损失，最终的总损失函数为式(4.11)。

$$L_{total} = L^{CLIP} + c_1 L^V + L^H \quad (4.11)$$

其中： c_1 是值函数损失系数，为了平衡策略损失和价值损失在反向传播的时候对梯度的贡献，需要将值函数损失乘一个系数，防止某部分损失在反向传播过程中主导网络的更新。

4.3.4 模型训练

强化学习的核心在于智能体与环境的交互过程中，通过不断试探与学习形成对动作策略的优化。本节详细描述训练流程，包括数据采集、优势计算、损失函数优化以及超参数调优过程。训练目标是使智能体能够通过学习有效策略，实现高效、合理的路径生成。

为提高训练效率与策略学习稳定性，本文仅在环境难度小于 0.5 的情形下开展训练。这些低难度环境主要包含结构较为简单、风口数目较少、障碍物排布规律性较强的空间结构，有助于智能体在早期阶段掌握基本策略以及状态和动作的映射关系。

训练流程如下：

（1）数据采集

智能体在当前策略下与环境进行交互，生成多个完整轨迹。每条轨迹包含多个时间步的状态（state）、动作（action）、奖励（reward）、动作概率（log prob）、

是否终止标记（done）和掩码（mask）等信息。这些数据构成训练所需的批次，作为后续策略更新的依据。

（2）计算回报与优势估计

采用蒙特卡洛回报方式估算未来累计奖励，进而计算优势函数，用于衡量当前动作相较于平均策略的优越程度。为提升训练稳定性，优势值在每个批次内进行了均值标准差归一化处理，从而避免梯度估计过大或过小的波动影响。

（3）损失函数计算

在每次策略更新中，使用 PPO 算法的目标函数作为损失设计基础，包含策略损失、价值损失和熵损失。总损失为三者加权求和，通过反向传播联合优化共享网络、策略网络和 Critic 网络参数。

（4）网络更新与优化器选择

本文采用 Adam 优化器对各子网络参数进行更新。Adam 融合了动量法与自适应学习率机制，能在面对稀疏梯度或非平稳目标函数时保持较高的优化效率，已广泛用于深度强化学习任务中。

模型的性能往往受超参数选择的影响极大。为了提升模型的训练效率和最终性能，超参数调优成为不可忽视的一环。为了在多个超参数组合中找到最优配置，本文采用了 Optuna 库进行自动化的超参数优化。Optuna 是一个自动化超参数优化框架，它基于贝叶斯优化原理，通过不断探索不同的超参数组合来优化目标函数，并且能够在大量的超参数空间中有效地搜索最佳解。

参考相关研究以及预实验初步确定了超参数的搜索范围，再使用 Optuna 库进行优化搜索。为平衡搜索时间与效率，共进行 100 轮搜索，每轮训练 5000 步，以平均奖励为评估指标。各超参数的搜索范围及取值和优化结果见表 4.1。

表 4.1 超参数优化结果

超参数	搜索范围	优化结果	说明
ppo_epochs	3~5	3	PPO 训练的轮数
batch_size	8, 16, 32	8	批量大小
gamma	0.95~0.99	0.97	回报的折扣因子
clip_param	0.1~0.3	0.28	剪切阈值
c1	1e-6~1e-4	7.32e-5	Critic 网络的权重参数
entropy_coef	0.01~0.001	0.007	熵正则化系数
shared_lr	1e-5~1e-3	8.17e-5	共享网络的学习率
actor_lr	1e-6~1e-4	7.80e-5	Actor 网络的学习率
critic_lr	1e-5~1e-3	4.57e-4	Critic 网络的学习率

图 4.4 展示了训练过程中策略成功率和损失曲线的变化情况。从训练曲线可以看出，模型在训练初期成功率快速上升，从 0.2 提升至 0.6 以上，说明智能体在短时间内便初步学习到了一定的策略结构。在中后期阶段，成功率持续稳步上

升，最终收敛至 0.92 左右，表明智能体在当前训练设定下具有良好的收敛性能和策略生成能力。

损失曲线中，初期损失值较大，在 1 万步以内迅速下降，随后整体保持在一个较低且稳定的范围内，偶有小幅波动。训练过程中损失函数中价值损失在训练初期占据主导地位，因此这一阶段的损失快速下降主要归因于 Critic 模块较快地学会了对环境状态进行准确的价值评估。在策略尚未大幅更新的早期，准确的状态价值估计能够显著提升优势函数估计的质量，进而稳定引导后续策略优化过程。随着训练的进行，损失曲线逐步趋于平稳，波动幅度减小，表明策略更新已进入细化阶段。

需要特别指出的是，本研究在模型训练阶段主要采用了难度等级低于 0.5 的中低复杂度环境样本，未涵盖具有较复杂空间结构或多风口分布的高难度场景。这一设置主要出于训练稳定性与样本效率的综合考虑。本研究的主要目标在于探索并验证所提出的神经网络结构与强化学习算法在风系统路径生成与水力性能优化中的可行性，因此优先选择中低难度样本有助于明确评估模型在基础任务中的表现，保证训练过程的可控性。

实验中观察到，智能体在中低复杂度环境中已能够稳定学习到具备一定普适性的路径生成策略，表明所构建的算法框架具备良好的学习能力和基础有效性。模型在更复杂的布局环境中的泛化能力尚需进一步验证。

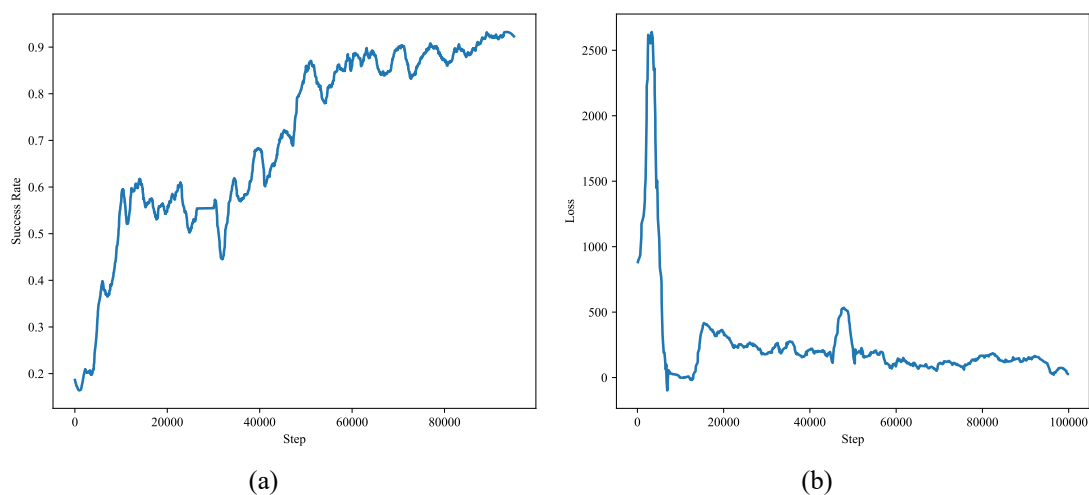


图 4.4 训练结果，(a)成功率曲线，(b)损失曲线

4.4 本章小结

本章围绕风系统支管路径生成任务，设计并实现了基于强化学习的路径策略训练方法。首先，结合风系统路径规划的结构特性与时序依赖，构建了具备空间

感知与时序建模能力的混合神经网络结构。该网络以残差卷积模块提取网格状态中的结构语义特征，并通过坐标增强机制强化对绝对空间位置的感知能力，同时引入 LSTM 模块对路径构建过程中的状态序列进行建模，确保策略生成过程具备合理的历史依赖性与行为连贯性。

随后，本文采用近端策略优化（PPO）算法作为路径策略训练的主干方法，建立了完整的策略评估与更新流程。为提升模型性能并减少手动调参的工作负担，引入了 Optuna 超参数优化框架，利用贝叶斯优化策略在高维超参数空间中自动搜索最优组合。在训练阶段，选取了路径生成难度相对较低的环境作为样本，重点验证所提出方法在典型场景中的适用性与有效性。实验结果表明，模型能够稳定地学习到路径延展的基础策略，具备一定的空间感知与时序控制能力。

第5章 风系统管路水力性能多目标优化

5.1 概述

在完成空调风系统支管路径的自动生成之后，与主管进行连接得到整个环路的管道设计结果。为实现系统运行性能与经济性的协同提升，需进一步对已生成的管路系统进行水力性能优化。风系统作为一个典型的分支式流体输送网络，其设计质量不仅依赖于合理的路径规划，还受到风管尺寸配置的影响。在多房间、多风口复杂布局下，如何在确保送风效果的基础上，同时降低系统压力损失与初投资成本，成为设计中亟需解决的核心问题。

本章面向路径已定的风系统，围绕管径参数的多目标优化展开建模与求解研究。首先，基于路径拓扑结构构建水力性能模型，综合考虑风管面积、系统阻力与送风不平衡度等性能指标，明确管径对系统运行效果的影响机理。随后，引入进化计算框架，将优化问题转化为典型的多目标优化问题，并采用非支配排序遗传算法（NSGA-II）对风管截面尺寸进行全局搜索。

本章的研究旨在实现风系统管路设计由路径可行向性能优化的跃升，进一步完善路径与参数一体化的自动设计方法体系。优化后的系统在保证设计合理性与气流均衡的基础上，能够有效降低能耗与成本，提升建筑暖通系统的整体运行效能与节能水平。

5.2 风系统管路水力性能优化问题建模

5.2.1 风系统管路生成结果储存格式

在空调风系统设计任务中，路径规划与水力性能优化的基础是对风管网络拓扑结构及其物理属性的准确建模和存储。风管系统天然具有分支性和方向性等拓扑特征：空气从空调机房或管井等起始节点出发，经由主干管和若干分支管道，最终送达多个末端风口。在此过程中，气流沿既定路径传播，经过三通、四通等分支节点对风量进行分配，不同路径上存在显著的风阻差异，影响系统平衡性与能效水平。

这一结构不仅包含了拓扑连接关系，还涉及风口、管段、配件等多类实体及其属性，如风量、尺寸、阻力、流速等，要求数据存储方法既要表达空间连接关系，又要支持对属性的查询、计算与动态更新。

针对上述需求，本文将路径生成结果抽象为有向图（Directed Graph）结构，以形式化、结构化的方式存储风系统的管路拓扑和运行参数，如图 5.1 所示。

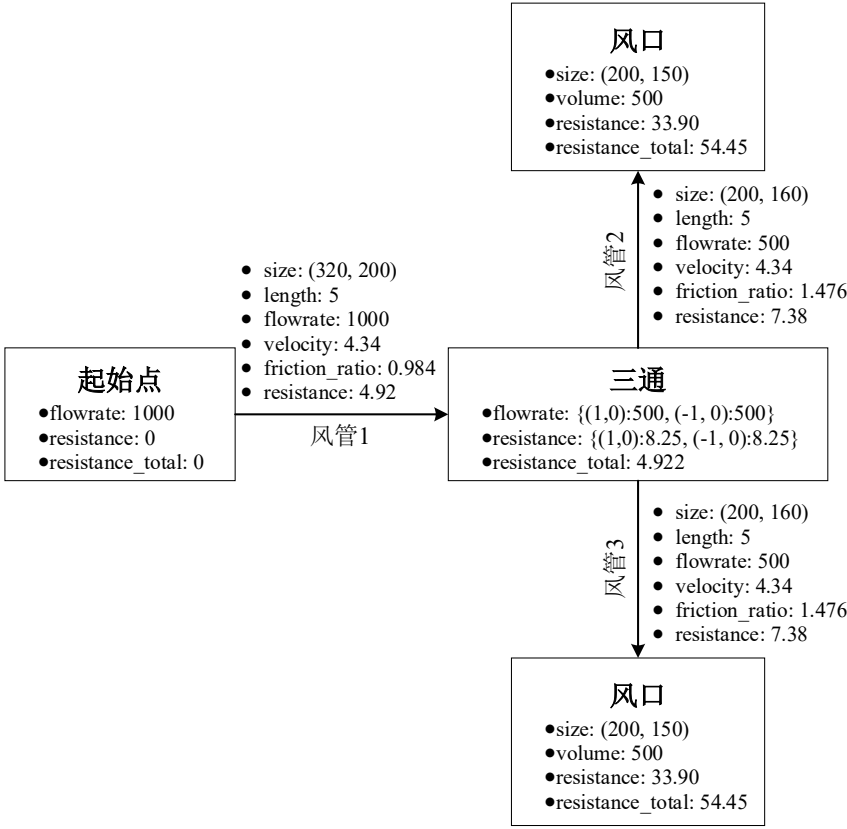


图 5.1 图结构存储风管设计结果示意图

表 5.1 图结构存储的数据及类型

类别	属性	字段名称	数据类型	说明
节点	类型	type	字符串	节点类型，包括风口、弯头、三通、四通、起始点等
	尺寸	size	元组	储存风口的尺寸，其他配件的尺寸通过访问其上下边获得，单位为 mm
	风量	volume	浮点数	风口节点的设计风量，m ³ /h
	流量	flowrate	浮点数或字典	节点的流量，三通/四通为后继方向与风量的映射字典，m ³ /h
	阻力	resistance	浮点数或字典	节点的局部阻力，风口为末端阻力，三通/四通为后继方向与阻力映射字典，Pa
	总阻力	resistance_total	浮点数	从初始节点到当前节点的总阻力，Pa
边	尺寸	size	元组	管道的横截面尺寸，mm
	长度	length	浮点数	管道的长度，m
	流量	flowrate	浮点数	管道的风量，m ³ /h
	风速	velocity	浮点数	管内的风速，m/s
	比摩阻	friction_ratio	浮点数	管道单位长度摩阻，Pa/m
	阻力	resistance	浮点数	管道阻力，Pa

该图结构中的节点 (Node) 代表送风系统中的关键设备或管道连接构件, 如送风口、弯头、三通等; 边 (Edge) 代表各节点之间的风管连接段, 携带与管段尺寸、风量、阻力等相关的物理量信息。相关的参数和数据类型列于表 5.1。

图结构节点与边之间的有向关系与实际气流方向一致, 可直接映射系统层次与分支结构, 不仅具备良好的工程解释性和计算兼容性, 还为后续的水力计算、性能评价与多目标优化提供了清晰的数据基础和灵活的操作接口。

5.2.2 风系统管路水力计算模型

在空调风系统设计中, 准确评估管道网络的水力特性是确保风量输配合理、系统运行高效的关键。管网中的气流在输送过程中会因摩擦、局部构件以及路径几何变化产生压力损失, 从而对风机选型、能耗评估和风口性能产生直接影响。因此, 需建立合理的水力模型以评估风系统中各类管段与节点的阻力特性。针对前面所构建的有向图结构, 系统性地建立了涵盖风管、弯头、三通、风口等构件的水力计算模型, 包含沿程阻力和局部阻力的计算, 并通过广度优先搜索的方式计算环路阻力。

5.2.2.1 沿程阻力计算

沿程阻力是由于气流与风管壁面之间的粘性摩擦引起的压力损失, 对于一段直管段, 其沿程摩擦阻力可由达西公式给出, 如式(5.1)。

$$\Delta P_f = f \cdot \frac{l}{D_h} \cdot \frac{\rho v^2}{2} \quad (5.1)$$

其中: f 为摩阻系数, 无量纲; l 为管道长度, m, D_h 为管道的水力直径, m, 对于矩形风管, 计算公式如式(5.2); ρ 为空气密度, 取 1.2 kg/m^3 ; v 为管道内的风速, m/s, 根据管道内的流量计算, 如(5.3)所示。

$$D_h = \frac{2ab}{a+b} \quad (5.2)$$

$$v = \frac{Q}{ab} \quad (5.3)$$

在沿程阻力计算模型中, 雷诺数是判断流动状态的关键参数, 计算公式为式(5.4)。当管道内的流体雷诺数 $\text{Re} < 2000$ 时, 为层流状态, 摩阻系数的计算公式为式(5.5); 当 $\text{Re} > 2000$ 时, 为湍流状态, 采用 Haaland 近似公式^[101], 即式(5.6), 计算摩阻系数。

$$\text{Re} = \frac{\rho v D_h}{\mu} \quad (5.4)$$

$$f = \frac{64}{\text{Re}} \quad (5.5)$$

$$f = (-1.8 \lg[(\frac{\varepsilon}{3.7D_h})^{1.11} + \frac{6.9}{\text{Re}}])^{-2} \quad (5.6)$$

其中： μ 为空气动力黏度，取 $1.81 \times 10^{-5} \text{ Pa}\cdot\text{s}$ ； ε 为风管内壁粗糙度，取 0.15 mm 。

5.2.2.2 局部阻力计算

局部阻力指的是由于气流流经管道中弯头、三通、四通、变径等非直管构件而产生的附加压力损失，计算公式为式(5.7)。

$$\Delta P_l = K \cdot \frac{\rho v^2}{2} \quad (5.7)$$

其中： K 为局部阻力系数。

根据设计手册^[101]，对于 90° 圆弧弯头， K 的取值为 0.25 ；而三通和四通的局部阻力系数需要查表确定。因此，本研究采用三次样条插值（Cubic Spline）查表确定三通和四通的局部阻力系数。

样条插值是一种常用的数值拟合方法，通过使用分段多项式来逼近函数，具有以下优点^[102]：

- （1）能够实现对给定数据点之间函数关系的平滑逼近；
- （2）可保证函数在各子区间上的连续性和导数连续性，通常为一阶、二阶连续；
- （3）相较于多项式拟合，可避免高阶振荡现象，更适用于局部数据插值。

5.2.2.3 水力计算流程

以图结构表示的风系统可视作由起点向末端风口逐步递进的有向网络，采用广度优先搜索实现环路阻力的计算，计算的伪代码列于表 5.2，具体流程为：

- （1）遍历所有边，计算长度、风速、尺寸、摩阻，写入边的属性；
- （2）遍历所有节点，判断节点类型，如风口、三通、四通等，选择对应模型计算局部阻力，写入节点属性；
- （3）识别初始节点，即入度为 0 的节点，初始化总阻力为 0 ；
- （4）遍历子节点，计算当前节点的局部阻力与连接边的沿程阻力；
- （5）将其与父节点的总阻力相加，得到当前节点总阻力；
- （6）遍历至末端节点。

上述流程实现了风系统路径结构中任意末端风口处总阻力的精确计算，具备良好的拓扑适应性与可扩展性，为后续的水力性能优化提供基础。

表 5.2 有向图水力计算伪代码

输入: 风系统图结构 $G(V, E)$
输出: 经过水力计算的风系统图结构 $G'(V, E)$

```

1: for each edge  $(u, v)$  in  $E$  do
2:    $size \leftarrow$  计算风管尺寸
3:    $velocity \leftarrow$  计算风速
4:    $friction \leftarrow$  计算单位摩阻
5:   更新  $G[u][v]$  中的属性:  $size, velocity, friction\_ratio$ 
6: end for
7: for each node  $n$  in  $V$  do
8:   根据  $n.type$  计算阻力, 写入  $G[n].resistance$  属性
9: end for
10:  $source \leftarrow$  查找入度为 0 的入口节点
11:  $G[source].total\_resistance \leftarrow 0$ 
12: 初始化队列  $Q \leftarrow [source]$ 
13: while  $Q$  非空 do
14:    $parent \leftarrow Q.pop()$ 
15:   for each  $child$  in  $G.successors(parent)$  do
16:      $r\_edge \leftarrow G[parent][child].friction \times G[parent][child].length$ 
17:      $r\_local \leftarrow G[parent].resistance[child]$  (若为 dict) 或  $G[parent].resistance$ 
18:      $total \leftarrow G[parent].total\_resistance + r\_local + r\_edge$ 
19:      $G[child].total\_resistance \leftarrow total$ 
20:      $Q.push(child)$ 
21:   end for
22: end while

```

5.2.3 风系统管路水力性能优化目标

空调风系统的性能表现对建筑能源效率与室内热舒适性具有重要影响。风系统布管不仅需满足所有送风末端的连通性和可施工性, 还应在材料消耗、系统阻力、送风均衡性等多个性能维度间实现协调优化。为将工程评价指标转化为可优化的数学目标函数, 本文从系统初投资、运行成本与气流均衡性三个角度出发, 定义以下三项优化目标:

(1) 风管展开面积最小化

风管初期投资成本与其展开面积密切相关。考虑风管通常为矩形截面, 其展开面积可由周长与长度乘积表示。设风系统图结构中边 (u, v) 表示一段风管, 其尺寸为 (w_i, h_i) , 长度为 L_i , 则展开面积目标函数定义如式(5.8)。

$$f_1 = \sum_{i=1}^N 2(w_i + h_i) \cdot L_i \quad (5.8)$$

该指标衡量了整体风管系统所需的材料面积, 反映施工材料的使用成本与布

管经济性。

(2) 系统最大阻力最小化

在风系统运行过程中，最不利送风末端的总阻力直接决定了风机的选型与运行负荷，较高的系统阻力将导致风机能耗显著提升，因此需要最小化最不利环路的阻力，其定义如式(5.9)。

$$f_2 = \max_{j \in D} R_{\text{total},j} \quad (5.9)$$

其中： D 表示所有送风末端集合； $R_{\text{total},j}$ 表示送风末端 j 的总阻力。

(3) 系统水力不均衡度最小化

风量分配均衡性是衡量风系统舒适性与稳定性的重要指标。若各送风末端存在明显阻力差异，则将导致风量分布失衡，影响末端的送风效果和温度控制性能。采用送风口总阻力的变异系数，即标准差与均值之比，衡量系统不均衡度，如式(5.10)。

$$f_3 = \frac{\sigma(R_{\text{total}})}{\mu(R_{\text{total}})} \quad (5.10)$$

其中： $\sigma(R_{\text{total}})$ 为所有送风口节点总阻力的标准差； $\mu(R_{\text{total}})$ 为所有送风口节点总阻力的均值。

综上，风系统管路水力性能优化问题被形式化为如式(5.11)的三目标最小化问题。

$$\min F = (f_1, f_2, f_3) \quad (5.11)$$

其中目标函数分别对应风管展开面积、系统最大阻力与阻力不均衡性。

5.3 风系统管路水力性能优化方法与流程

在完成风系统路径构建与水力性能建模的基础上，本节进一步引入多目标进化算法对管径参数进行联合优化，旨在在系统初投资、阻力水平与平衡性能之间实现综合权衡。采用以管段尺寸为核心的个体编码策略，配合可行域构造与解码修复机制，建立完整的变量表示、适应度函数与优化流程，在满足工程可行性的前提下寻求更具全局性的设计解。

5.3.1 决策变量定义与个体编码

在风系统管路水力性能优化问题中，优化目标的实现依赖于对管道设计参数的合理调整，因此选择管道的边长作为决策变量，并设计适用于进化计算的个体编码结构，如图 5.2 所示。

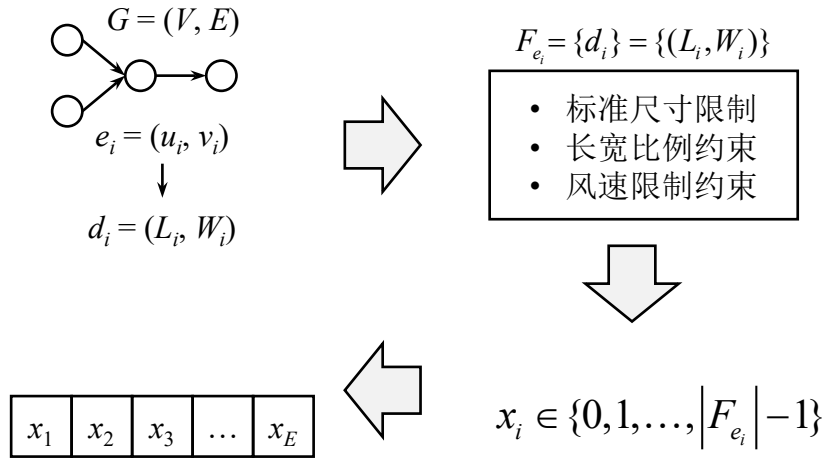


图 5.2 个体编码示意图

风系统在水力分析建模阶段被抽象为一个有向图结构 $G = (V, E)$ ，其中每个边 $e \in E$ 表示一段风管。对于每条边 $e_i = (u_i, v_i)$ ，其设计变量为风管的长宽尺寸对如式(5.12)所示。

$$d_i = (L_i, W_i) \quad (5.12)$$

其中： L_i ， W_i 为风管截面的长和宽，mm，需要满足如下限制：

(1) 标准尺寸限制：在实际工程中，风管的制作和安装需遵循统一的工业标准。非标准尺寸不仅增加加工难度、材料浪费和施工成本，也可能无法适配现有的支吊架、风口及调节装置，影响工程的一体化协调。将边长尺寸限定在标准尺寸集合之中，能够有效确保优化结果在实际施工中的制造与采购。

(2) 长宽比例约束：风管的长宽比例关系到其结构稳定性、制造工艺及气流组织效果。若长宽比过大，风管截面呈扁平形态，在风压作用下容易发生形变甚至共振，降低系统可靠性；同时也会增加法兰连接、吊装支撑的结构复杂度。采用长宽比例约束可以显著提高优化结果的结构稳健性，避免生成极端而不可行的断面设计，增强系统整体的结构稳定性与安装便利性。

(3) 风速限制约束：风速是风系统运行安全与舒适性的关键参数。过高风险不仅会导致较大的压降和风机能耗，还会引发噪声与振动等问题，从而降低室内环境品质。引入风速限制条件，是确保系统运行效率与环境舒适性的基础前提。

因此，变量空间是一个具有不规则约束的离散组合域，每条边 e_i 的可行域表示为式(5.13)。

$$F_{e_i} = \{(L, W) \in D^2 \mid L \geq W, \frac{L}{W} \leq r_{\max}, L \cdot W \geq \frac{Q_i \times 10^6}{3600 \times v_{\max}}\} \quad (5.13)$$

其中： D 表示标准风管尺寸集合； $r_{\max}$ 表示长宽比上限，取 3.0； Q_i 为该风管段的

风量, m^3/h ; $v_{\max}$ 为管道内的风速上限值, 对于主管选择 8 m/s , 支管为 6.5 m/s ^[101]。

为适配进化优化算法的操作机制, 本文采用基于索引映射的整数编码策略。将图中所有边按固定顺序编号为 $e_1, e_2, \dots, e_E$, 则每一个个体, 即候选解, 可编码为一个长度为 E 的整数向量, 如式(5.14)。

$$\mathbf{x} = [x_1, x_2, \dots, x_E], x_i \in \{0, 1, \dots, |F_{e_i}| - 1\} \quad (5.14)$$

其中: x_i 是第 i 条边的取值索引。对于每个 x_i , 有:

$$F_{e_i}[x_i] = (L_i, W_i) \quad (5.15)$$

这种编码方式使用整数索引而非实际尺寸表示, 降低编码复杂度; 可以适应不同规模图结构与不等长可行域; 可直接基于索引在每个边的可行解域中执行替换或扰动, 便于交叉和变异操作。

5.3.2 设计解生成与修复机制

为了实现风系统管路水力性能的优化, 优化算法需在可行的尺寸组合空间中搜索最优设计解。由于设计变量是基于图结构中每条风管边的尺寸选择, 个体的结构编码需映射到一套具有物理含义的管路布置方案。本研究提出一套包含个体解码、图结构构建、水力参数更新、可行性修复等步骤的设计解生成机制, 确保候选解不仅结构合法, 而且满足风系统工程特性。

在遗传编码中, 每个个体是长度为 E 的整数向量, 每个位置 x_i 对应于图中边 $e_i = (u_i, v_i)$ 的尺寸选择索引。为了评估该个体的水力性能, 需要将其映射回风系统图结构, 即为每条边分配具体的尺寸配置, 该过程称为解码, 具体步骤如下:

(1) 原始图结构复制: 对基础风管图 $G_0 = (V, E)$ 进行深拷贝, 保留结构不变, 仅更新边和节点属性;

(2) 映射尺寸配置: 对于每条 e_i , 依据个体编码 x_i , 提取其在可行解域中的尺寸配置 (L_i, W_i) 并写入图中;

(3) 更新边的参数: 依据新尺寸, 调用水力计算模块更新每条边的风速、比摩阻和阻力;

(4) 更新节点参数: 以广度优先搜索方式, 从入口点出发逐级向下游节点更新节点的阻力和总阻力。

风系统路径具有层级拓扑结构, 每个节点的上下游边的尺寸应满足一定的递减规则或尺寸一致性, 避免出现主干尺寸小于支干或弯头前后尺寸不匹配等工程问题。由于随机初始化或进化操作可能产生不满足物理约束的尺寸配置, 本研究结合系统结构特征, 引入如下两类修复规则:

(1) 尺寸单调性修复: 即对于每个有任意出边 (n, c) 和唯一入边 (p, n) 的节点,

出边和入边的尺寸需满足式(5.16)。

$$L_{n,c} \leq L_{p,n}, W_{n,c} \leq W_{p,n} \quad (5.16)$$

当前尺寸不满足上述条件，则从可行域中重新选择满足该约束的最接近配置。

(2)弯头尺寸一致性修复：风系统中的弯头前后管道通常需保持尺寸一致，以确保气流平稳过渡与连接构件的标准化制造。设当前节点为弯头节点，其唯一入边为 (p, n) ，出边为 (n, c) ，进行式(5.17)的强制约束。

$$L_{n,c}, W_{n,c} \leq L_{p,n}, W_{p,n} \quad (5.17)$$

若出边当前选择的尺寸不符合该约束，则自动替换为与入边完全一致的配置，若在可行域中不存在该配置，则选择最接近该尺寸的合法解。

修复机制示意图如图 5.3 所示，其中(a)为尺寸单调性修复；(b)为弯头一致性修复。

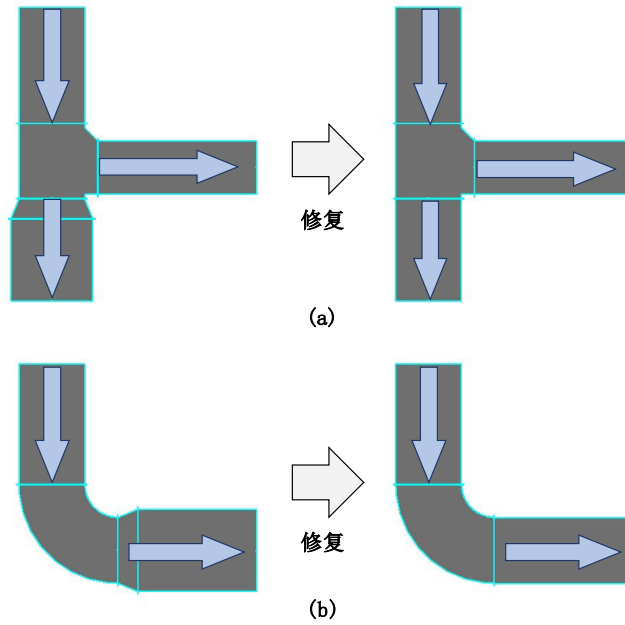


图 5.3 修复机制示意图

为高效实现上述修复规则，使用基于广度优先搜索图遍历的解修复流程，从入口节点出发，进行拓扑顺序遍历。该修复策略在维持进化多样性的同时，有效防止不合理基因传播至下一代，提升优化效率与结果有效性。

5.3.3 多目标优化算法

考虑到目标函数间存在相互制约关系，例如增大管道边长会引起管道面积的增加，但是可以减小管道阻力。在这样的情形下，本研究引入非支配排序遗传算法 II (NSGA-II) 作为优化框架，以系统性探索在不同目标间的折中方案。

NSGA-II 是一种广泛应用于工程设计问题的多目标进化算法，其核心思想是通过种群演化逐步逼近 Pareto 最优前沿（Front），在目标冲突的空间中寻找一组彼此互不支配的高质量解^[103]。Pareto 前沿表示在所有目标之间达到最优权衡的非支配解集合。一个解是 Pareto 最优的，表示没有其他解在所有目标上都优于它，即它是不可被支配的，如图 5.4 所示。相较于传统启发式方法，NSGA-II 能在单次运行中输出一组多样化的优化候选方案，为后续的工程决策提供充分的灵活性与可选性。

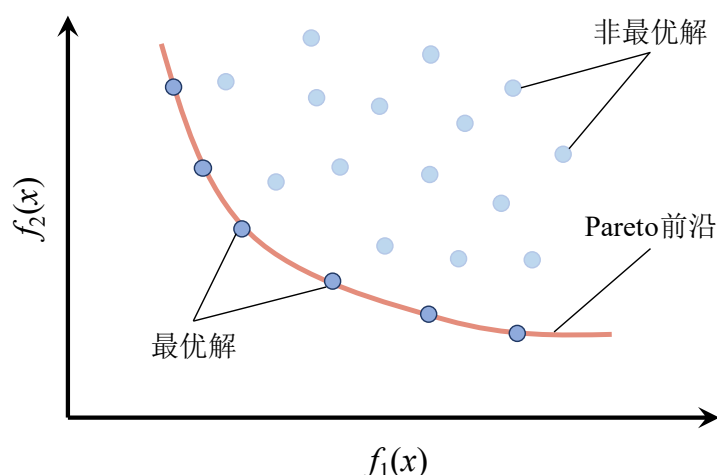


图 5.4 Pareto 前沿示意图

该算法主要依赖如下两个机制：

- （1）非支配排序（Non-dominated Sorting）：将种群划分为若干支配等级（Front），等级越高的个体被更少的解所支配，表示在多个目标上表现更优；
- （2）拥挤距离（Crowding Distance）维护机制：用于评估解在目标空间中的分布稀疏程度，鼓励解的多样性，防止种群早熟收敛。

本研究在 DEAP（Distributed Evolutionary Algorithms in Python）框架下实现了完整的 NSGA-II 优化流程。DEAP 是一个用 Python 编写的灵活、模块化、功能强大的进化计算框架，广泛用于实现遗传算法、遗传编程、差分进化、多目标优化等元启发式算法。

NSGA-II 优化流程图如图 5.5 所示，具体步骤如下：

- （1）可行域构建：依据每条风管边的流量与最大允许风速，计算其在标准管径库下的所有合法尺寸组合，形成个体基因的取值空间。
- （2）算法算子初始化：对一系列关键算子进行初始化，包括种群结构、个体编码、遗传操作及目标评价函数等。
- （3）构建参考点与超体积（Hypervolume, HV）函数初始化

在多目标优化中,不同目标往往具有数量级差异和物理含义不一致性,直接比较原始目标值会导致目标之间的权重失衡,从而影响优化算法对 Pareto 前沿的探索能力。为了解决这一问题,并实现对多目标解集质量的有效评估,在优化过程中引入归一化参考点与超体积指标,构建系统化的优化进展度量机制。

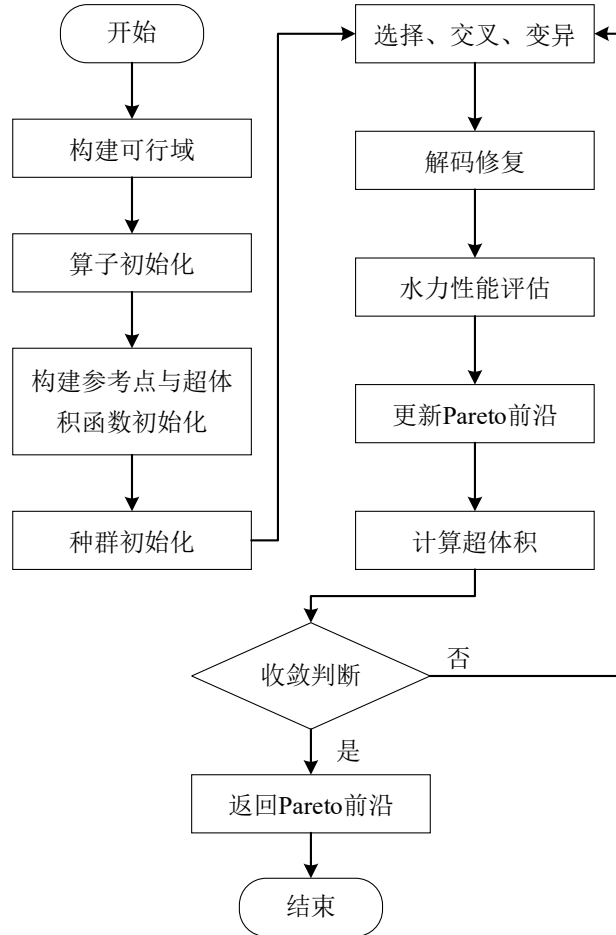


图 5.5 NSGA-II 多目标优化流程图

评估不同解集在目标空间中的覆盖范围需要一个与目标无关、统一标准的归一化基准,即参考点。参考点的位置必须位于所有当前 Pareto 解之外,并且其构建不依赖特定解,而是根据问题属性和最劣解估计产生,用于将原始目标向量转换为统一比例下的度量空间。

在本研究中,参考点的构建策略如下:对面积目标,每条风管边选择面积最大的尺寸组合,计算管道总展开面积;对阻力目标,选择阻力最大的尺寸组合,进行水力计算后提取最不利风口总压损;对于阻力均衡性,固定为常数 1,如式 (5.18)所示。

$$z^{\text{ref}} = (f_1^{\max}, f_2^{\max}, 1.0) \quad (5.18)$$

超体积是多目标优化中广泛使用的质量指标,其定义是当前非劣解集与参考

点之间在目标空间中构成的超立方体体积，计算公式如(5.19)。

$$HV(\mathcal{P}, z^{\text{ref}}) = \lambda \left(\bigcup_{f \in \mathcal{P}} [f, z^{\text{ref}}] \right) \quad (5.19)$$

其中： $\mathcal{P}$ 是一组归一化目标向量集合， $\mathcal{P} = \{f_1, \dots, f_n\}$ ； λ 表示 Lebesgue 测度； $[f, z^{\text{ref}}]$ 是由解 f 与参考点 z^{ref} 构成的正轴对齐的超矩形。为了避免各指标在数值尺度上的差异影响，将每项指标使用参考点的对应指标值进行归一化后，计算归一化超体积。

(4) 种群初始化：按照设定的种群规模生成初始种群。

(5) 遗传循环流程

在完成种群初始化后，NSGA-II 算法进入主循环阶段，通过交替执行选择、交叉、变异、适应度评估与非支配排序选择等一系列遗传操作，不断更新并优化种群，从而逼近 Pareto 最优前沿。交叉和变异示意图如图 5.6 所示。

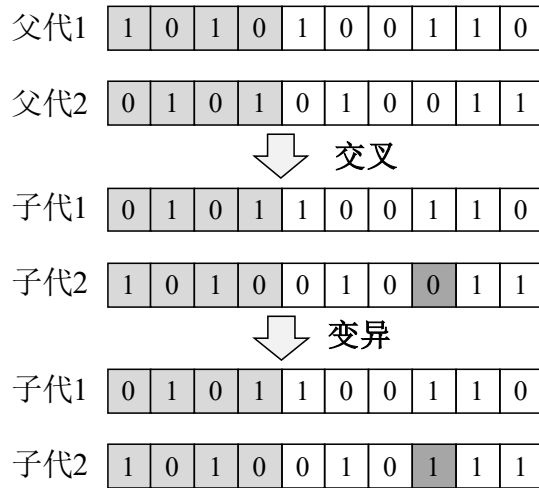


图 5.6 交叉变异示意图

其整体流程如下：

- 选择操作 (Selection): 应用 NSGA-II 选择算子对当前种群进行排序，并保留精英个体；
- 交叉操作 (Crossover): 以设定的概率在选定的父代个体对之间进行，采用均匀交叉策略，提升搜索空间探索能力，保持遗传多样性；
- 变异操作 (Mutation): 对交叉后的个体进行微扰调整，以预先定义的概率对个体的某些基因位进行变异，即对重新从边的尺寸选择索引可行域中采样一个新值；
- 适应度评估 (Evaluation): 对变异后的新个体重新进行目标函数适应度评估。

(6) 早停策略

为了避免不必要的计算资源浪费,同时确保算法在性能改进趋于饱和时及时停止,引入了基于超体积指标的早停策略。

设当前第 t 代的归一化超体积为 $HV(t)$, 每一代的改进幅度为式(5.20)。

$$\Delta HV(t) = HV(t) - HV_{\text{best}} \quad (5.20)$$

其中: HV_{best} 为历史最优的超体积。当存在连续 T 代满足式(5.21)。

$$\Delta HV(t) < \varepsilon, \text{ for } t \in [g - T + 1, g] \quad (5.21)$$

即说明连续 T 代中当前解集相较历史最优未取得显著性能提升,判定算法已近收敛,触发提前终止。其中 g 为当前代数, T 为耐受代数; ε 为超体积最小改进阈值。

此外,为防止超体积波动但尚未真正收敛,设定最大代数限制。

5.4 本章小结

本章围绕风系统管路设计中水力性能优化的关键问题,系统提出并实现了一种基于多目标进化算法的优化框架。该方法以风管尺寸为决策变量,兼顾系统初投资与运行性能等多个目标,实现了风系统管径的多目标优化。主要工作总结如下:

问题建模方面,本研究基于图结构对风系统布局结果进行了抽象表示,将风口、弯头、三通等构件建模为节点,风管建模为边,在此基础上构建了面向节点与边的属性描述体系,并制定了水力性能的详细计算流程,包括沿程阻力与局部阻力的统一建模与计算。

优化目标方面,综合考虑系统生命周期成本与运行效能,明确提出了三个优化目标:系统总管道面积(反映初投资)、最大送风末端阻力(衡量最不利环路运行压力)与阻力标准差归一值(评估阻力均衡性)。通过目标函数设计实现了工程性能指标的经济转化与可比性提升。

优化方法方面,采用非支配排序遗传算法(NSGA-II)作为求解策略,并结合风系统约束特点设计了编码解码机制、可行域筛选、约束修复算子与适应度评估函数,实现了对风管尺寸组合的高效搜索。引入超体积指标与早停策略,用于优化过程的效果评价与收敛控制。

第6章 风系统管路自动生成方法验证

6.1 概述

本章通过具体案例展示风系统管路自动生成方法的初步应用效果,以及风系统管路水力性能优化的验证结果。对于支管路径生成方法,选择了部分简单环境进行测试,测试结果表明模型已经初步学习到了基本管道路径规划策略,并指出在更复杂环境中的挑战和优化方向。此外,基于一个真实建筑,展示了风系统管路水力性能优化的验证结果,所提出方法可在多个性能指标之间实现合理权衡,相比基于假定流速法的设计结果有明显改善。

6.2 风系统支管生成案例验证

本节展示了风系统支管路径生成的案例结果,验证模型在不同环境下的路径规划能力与适应性。所采用的训练与测试环境基于第3章提出的环境生成方法构建。

测试阶段选取了典型的简单结构样本,并使用训练完成的策略模型进行路径生成。结果表明,在较为简单的房间结构中,例如风口较少的矩形或轻度非规则的L型空间中,模型能够稳定生成从入口到所有风口的可行路径。这表明模型已经初步学习到了基本管道路径规划策略。

然而,在更复杂的房间环境中,如存在多个风口或形状较不规则的场景中,模型训练的难度增大,路径生成的成功率显著下降。主要问题表现为路径不连通、超出边界、穿越障碍等。该现象反映出目前的训练策略在应对高复杂度环境方面存在局限,可能与以下因素相关:

(1) 训练时间不足。强化学习算法依赖大量的交互样本与策略迭代才能在复杂策略空间中收敛,尤其在具备长序列决策需求的路径规划任务中更为显著。复杂环境中的路径规划解空间呈指数增长,智能体需花费更长时间探索有效策略。然而目前模型训练样本主要来自中低难度环境,并且训练轮次有限,智能体尚未充分经历多样化的复杂环境交互过程,导致策略未能掌握多目标连接和复杂路径等高级行为模式,最终表现为路径生成失败或策略无效。

(2) 奖励函数设计与优化目标限制了模型的探索激励与策略引导能力。在复杂房间中,风口往往布局分散,路径生成任务具有明显的稀疏反馈特征,即只

有当路径成功连接至入口或已有路径时才可获得正向奖励,这使得随机初始化策略难以获得正向回报。缺乏更具中间反馈性质的辅助目标限制了模型在高复杂度状态空间中的有效学习能力。

(3) 网络结构在表达复杂空间拓扑结构与全局策略协同方面能力仍显不足。当前所采用的神经网络结构尽管在中小规模网格结构上具有良好表达力,但在处理具有多目标(多个风口)、复杂平面布局时存在一定信息瓶颈。

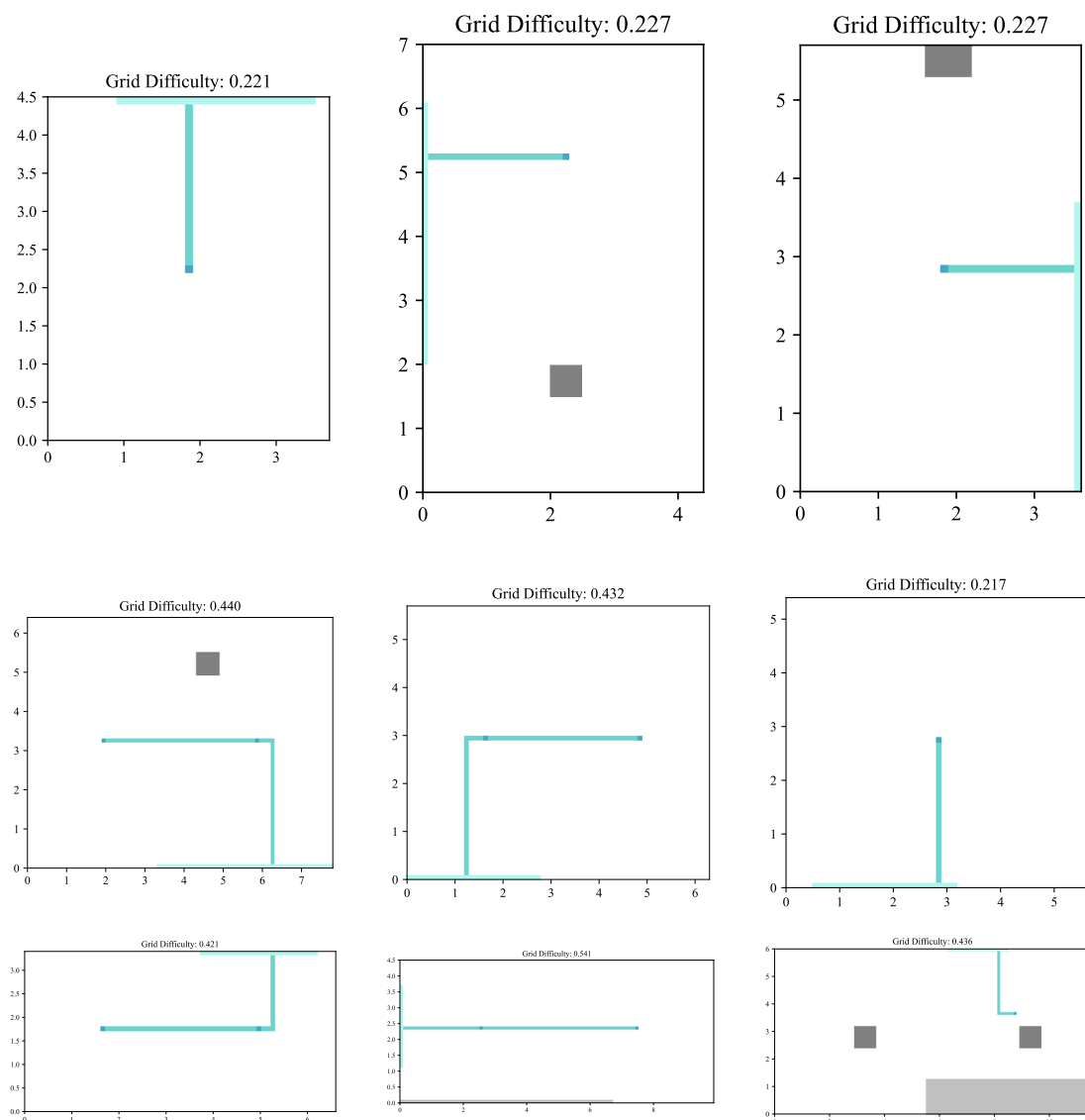


图 6.1 支管生成案例

针对训练策略在复杂环境中表现不佳的问题,以下提出若干具有针对性的解决方案,以提升模型在高复杂度场景下的表现能力:

(1) 增强奖励机制,缓解稀疏反馈问题。复杂环境中的主要挑战之一是回报信号稀疏,可以考虑增强中间反馈和过程引导信号,提升智能体在复杂情境下

的学习效率与探索能力。

(2) 优化神经网络结构。当前模型基于残差卷积与 LSTM 的架构，在面对多目标布局与复杂空间拓扑结构时，其信息捕获与长期依赖建模能力可能不足。为此，可以考虑引入注意力机制，通过聚焦关键区域增强模型对空间中重要元素（如风口、障碍分布）的感知能力；以及引入 Transformer 模块，在长序列决策中实现更稳定的策略学习。

(3) 利用专家经验引导强化学习学会探索。专家经验可以来源于人工设计路径、基于规则的算法或历史工程项目中的典型布管方式。将这些经验性策略引入智能体的学习过程，有助于提供可行解样本，还可以提升策略搜索的方向性和目标相关性，缓解随机初始化模型在稀疏奖励问题下的训练困难。

尽管如此，本文所提出的强化学习方法在中低复杂度环境中已展现出良好的学习效果与路径规划能力。这一成果证明了所提方法在路径策略建模上的理论可行性，也为后续扩展至更高复杂度任务奠定了坚实的基础，初步实现了从传统规则驱动向数据驱动路径设计模式的转变，为风系统自动化设计提供了一种可推广的思路与实现框架。

6.3 风系统管路水力性能优化验证

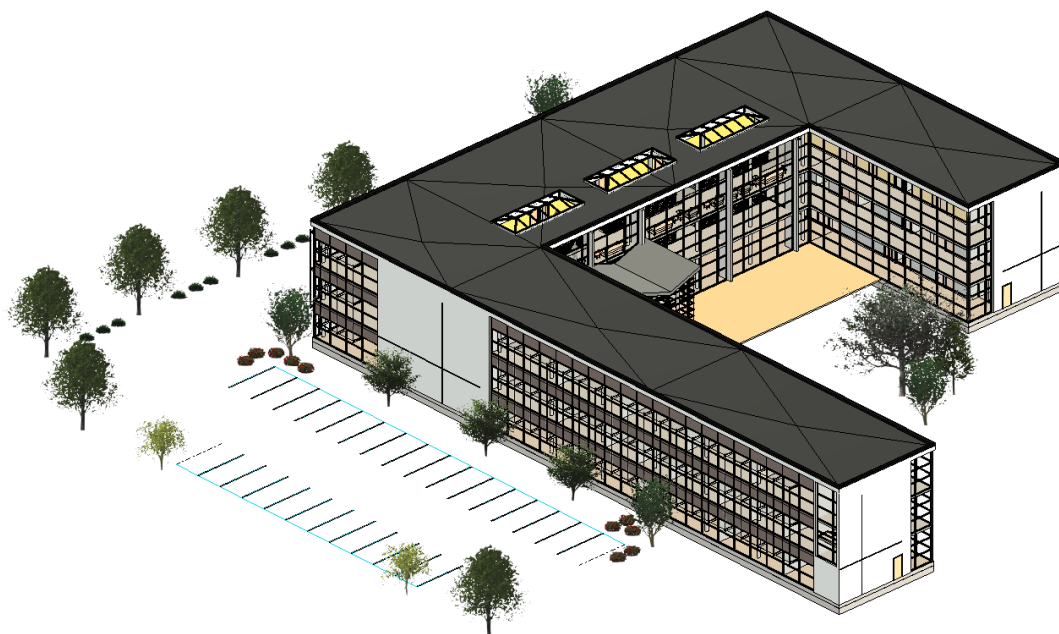
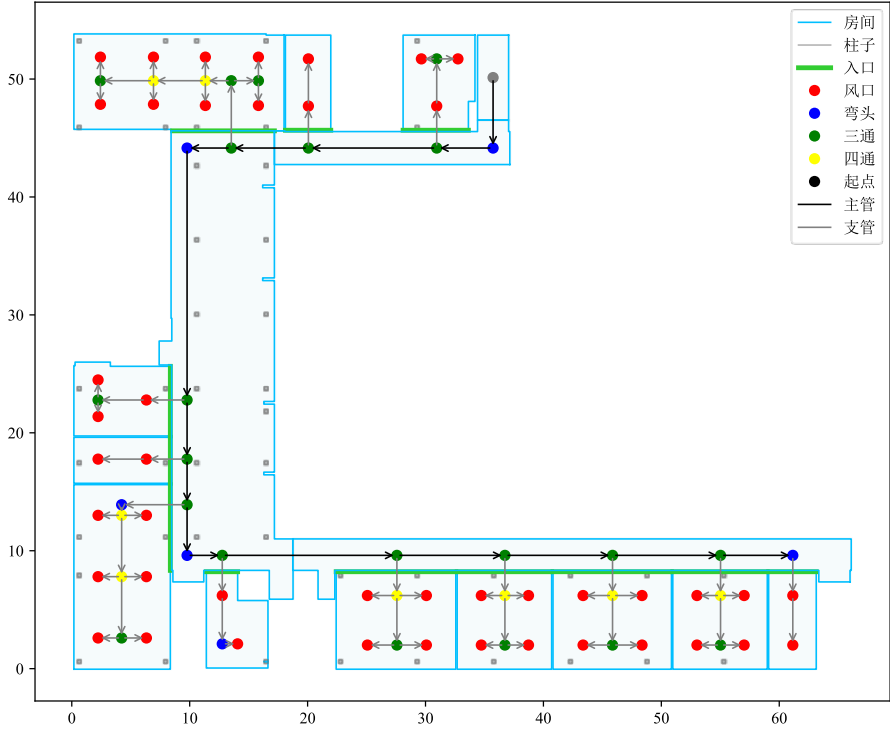


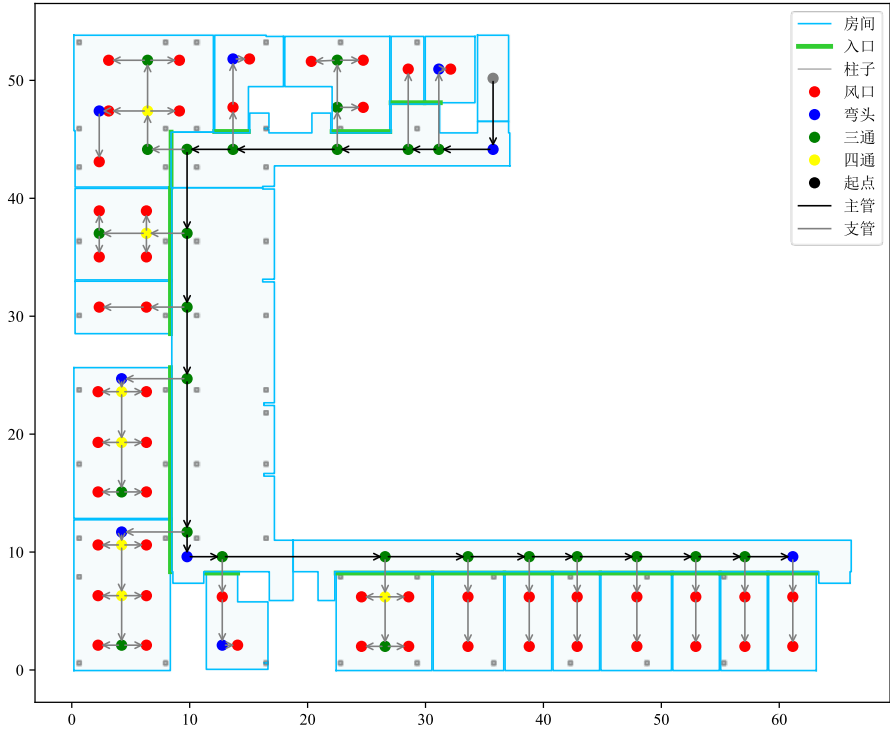
图 6.2 测试建筑模型

为验证风系统管路水力性能优化算法的有效性，本章选取了一栋真实办公建

筑作为测试对象，并应用优化方法对其风系统管路进行优化。测试建筑的 Revit 模型如图 6.2 所示。该建筑为办公建筑，地上 3 层，内部的房间包括办公室和会议室。



(a)一层



(b)二层

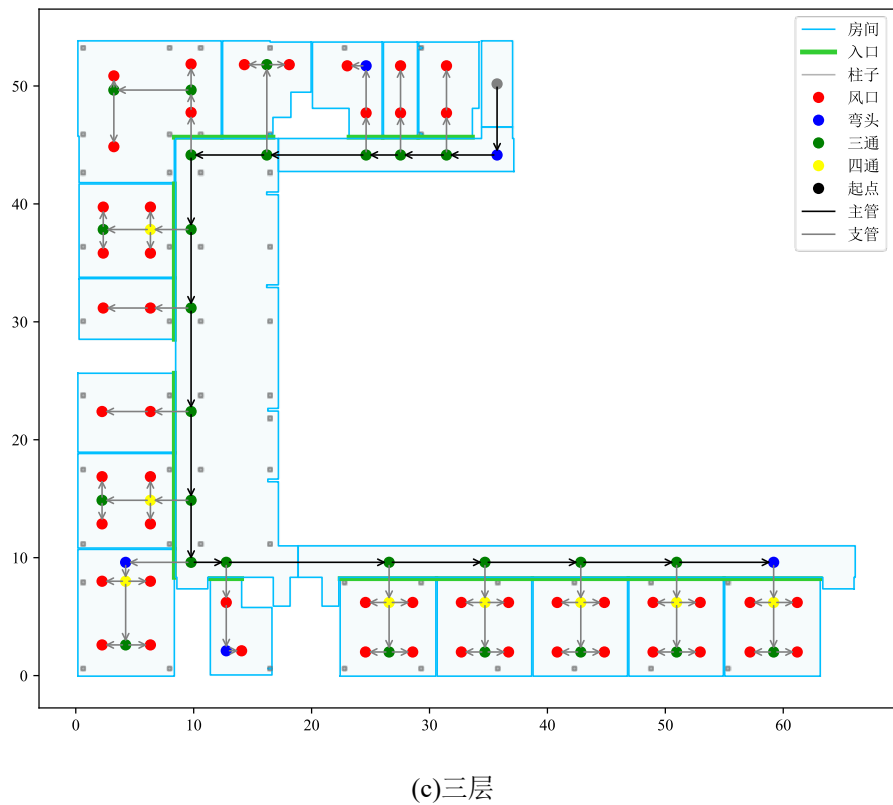


图 6.3 测试建筑各环路平面布置图

该建筑的风系统设计被分为三个环路，每个楼层作为一个独立的环路进行设计。主管按照第 2 章中的走廊主管布置方法生成，支管采用第 4 章的策略网络生成。由于对于部分较为复杂的房间布局，特别是风口较多的房间，智能体尚未能够学会有效的路径规划策略，因此采用基于规则^[5]的方法生成管道路径。主管和支管生成之后，将其拼接成整个环路的风系统的有向无环拓扑图，表示环路的风管设计结果，如图 6.3 所示。各环路风口和风管数量如表 6.1 所示。

表 6.1 各环路风口和风管数量

环路	风口数量	风管数量
一层	44	80
二层	50	92
三层	50	90

得到环路设计结果后，基于第 5 章的方法，开展基于 NSGA-II 算法的管路水力性能优化实验。优化目标包括：管道展开面积、系统最大阻力、不平衡性三项核心性能指标，这些指标直接关系到风系统的效率与稳定性。NSGA-II 多目标优化算法的各参数的取值见表 6.2。

表 6.2 NSGA-II 多目标优化算法参数取值

参数	取值
种群数量	300
最大代数	1000
耐受代数	20
最小改善阈值	$10e-5$
交叉概率	0.9
变异概率	0.1

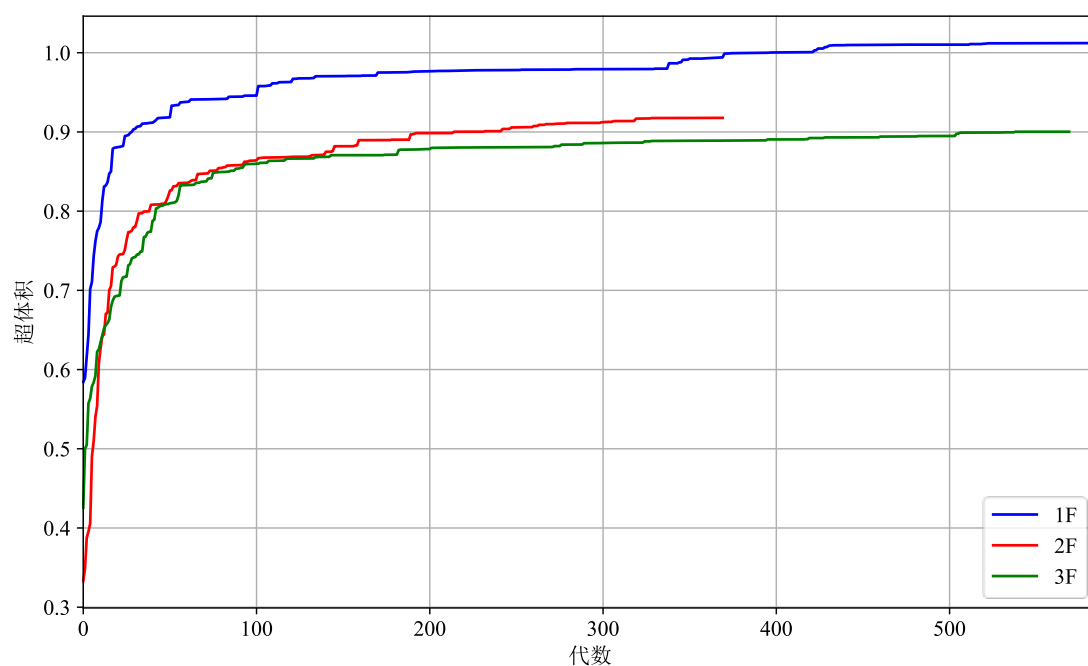


图 6.4 管路水力性能优化案例种群收敛曲线

图 6.4 展示了风系统管路多目标优化过程中超体积指标随迭代代数的变化曲线，分别对应建筑一层、二层和三层风管环路的优化过程。超体积作为衡量解集在目标空间中覆盖范围和多样性的重要指标，能够反映优化种群在多目标维度上整体性能的提升趋势。

从图中可以看出，三组测试中种群在前 50 代内超体积均呈现快速增长，随后进入相对平稳的收敛阶段，表明算法能够在初期快速探索到多个具有代表性的非支配解，并在随后的迭代中持续改善。比较不同楼层的超体积演化趋势可以观察到：一层环路具有最优的收敛性能，其超体积在 400 代以内稳定提升至 0.99 左右，显示出解集在面积、阻力、均衡性三目标间取得了良好的折中优化；二层环路在前期提升速度快，最终超体积趋于 0.91，略低于一层，说明其目标间存在一定冲突性，对搜索空间提出了更高的挑战；三层环路的超体积最终稳定在 0.90 左

右。三个环路收敛曲线的差异可能是由于解空间维度随着管道和风口数量的增加而增加，导致优化算法在选择最优解时需要更多的时间来进行搜索与探索。

图 6.5 展示了针对测试建筑中不同楼层风系统管路所获得的多目标优化解集在三个性能指标维度下的分布情况，其中每一个点表示一个非支配解，代表在该配置下三个目标的权衡结果。图中三组楼层的 Pareto 解均形成了典型的边界带状结构，说明优化过程有效识别并保留了在不同目标组合下具备代表性的高质量解集。其中一层环路（图 a）的 Pareto 解集分布广度和密度较高，呈现出明显的面积-阻力反比关系，表明在该问题规模下优化器能够稳定探索出多样化解，且目标间权衡结构清晰。

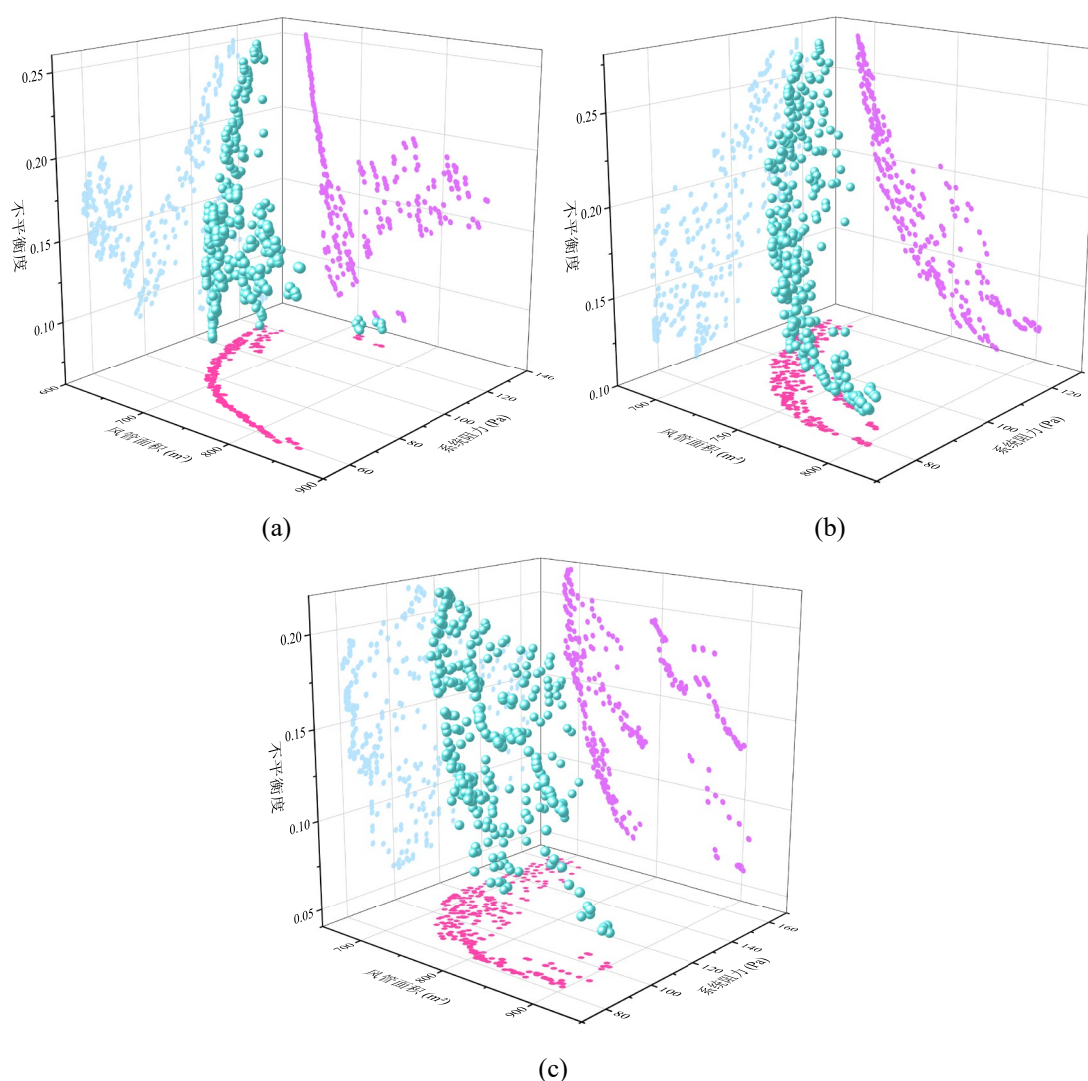


图 6.5 测试建筑各环路 Pareto 前沿图，(a)一层环路，(b)二层环路，(c)三层环路

图 6.6 展示了三组 Pareto 前沿解集中三个性能指标——风管面积、系统阻力与不平衡度的均值与标准差统计结果。图中三角形标注点代表假定流速法下计算

得到的设计方案对应的性能指标。假定流速法进行水力计算时设计参数设定为主管内的推荐流速 6 m/s，支管推荐流速 4 m/s，管道的长宽比为 1.5。

对比分析可见，在管道面积指标上，三组 Pareto 前沿解集相较于假定流速法的参考设计结果都有明显改善；在不平衡度指标上，三层环路的不平衡度均值比参考点改善超过 50%，其余两个环路也有显著改善；但是在阻力指标上，大部分的 Pareto 最优解都比假定流速法计算的参考值高，这说明在追求管道面积最小化与风量均衡性提升的过程中，算法可能在一定程度上牺牲了部分系统阻力性能，从而实现整体性能的协同优化。平均而言，Pareto 最优解相较参考结果在风管面积上改善 9.79%，在不平衡度上改善 40.01%。在值得注意的是，一层与二层环路中，假定流速法的解被部分 Pareto 解完全支配，意味着在所有三个目标维度上均存在更优的设计组合，进一步印证了 NSGA-II 多目标优化方法在管道水力优化问题上全局搜索能力与综合性能平衡方面的优势。

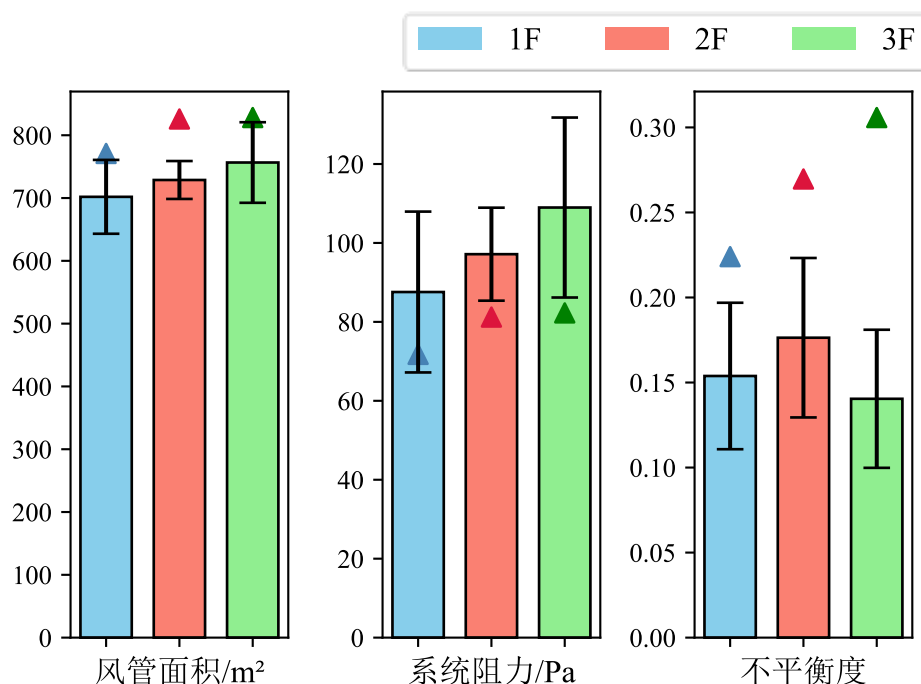
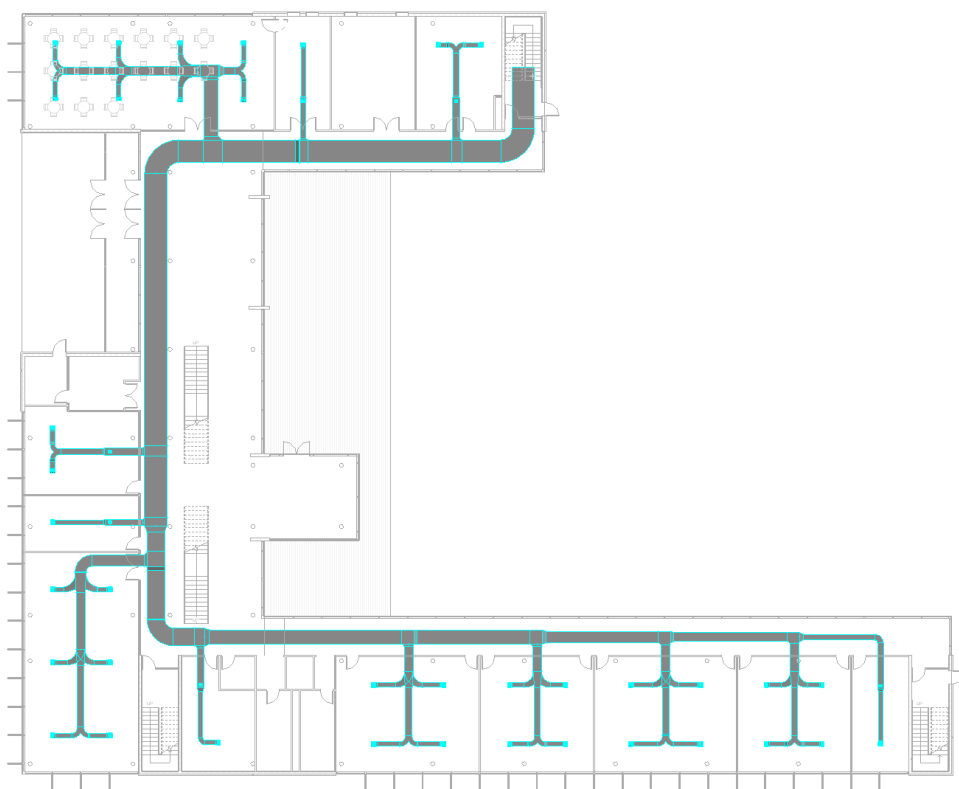
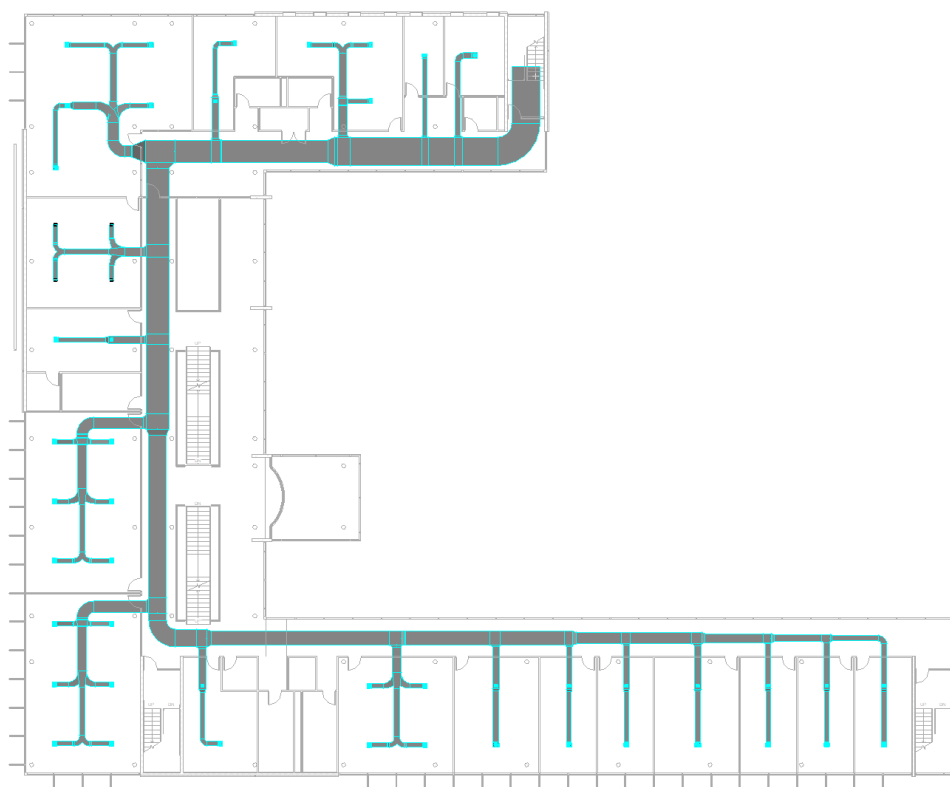


图 6.6 测试建筑各环路 Pareto 最优解的评估指标

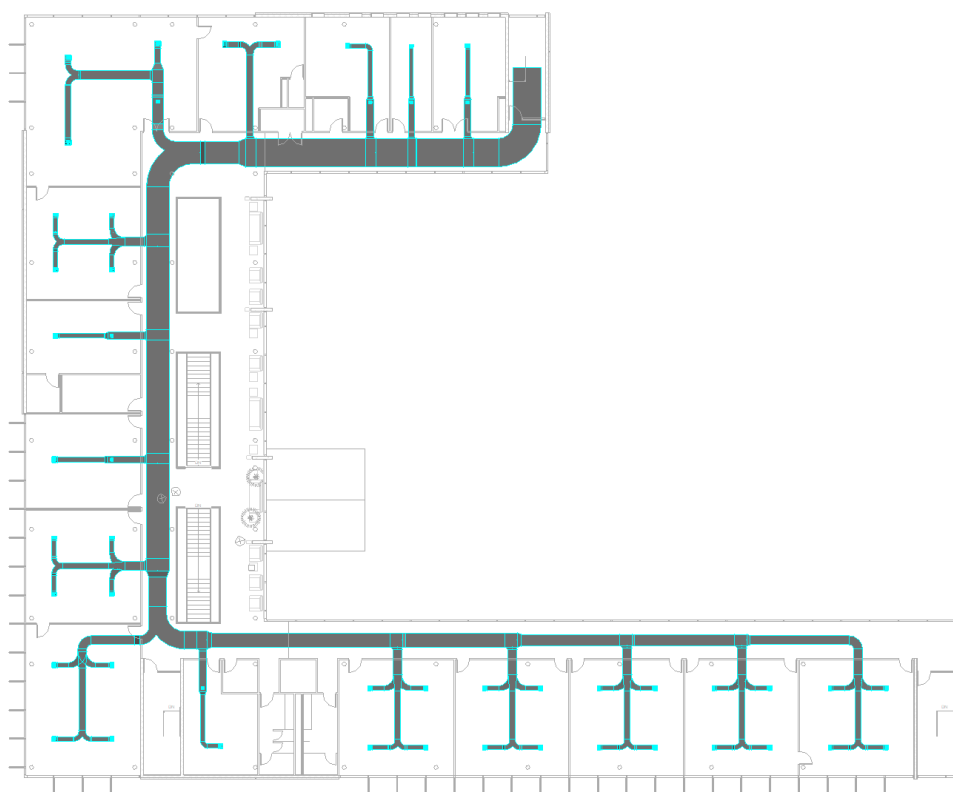
图 6.7 所示为测试建筑的风管设计结果，模型通过 Revit 二次开发接口调用设计选型结果，在 Revit 中自动生成，实现了风系统设计与三维建模流程的高效集成。Revit 二次开发接口是 Autodesk Revit 提供的一套用于扩展和定制其功能的 API 工具，支持开发者通过 C#、VB.NET 等编程语言构建定制化的应用程序和插件，从而满足建筑设计、施工与工程领域中的特定需求，提升建模效率与自动化程度。



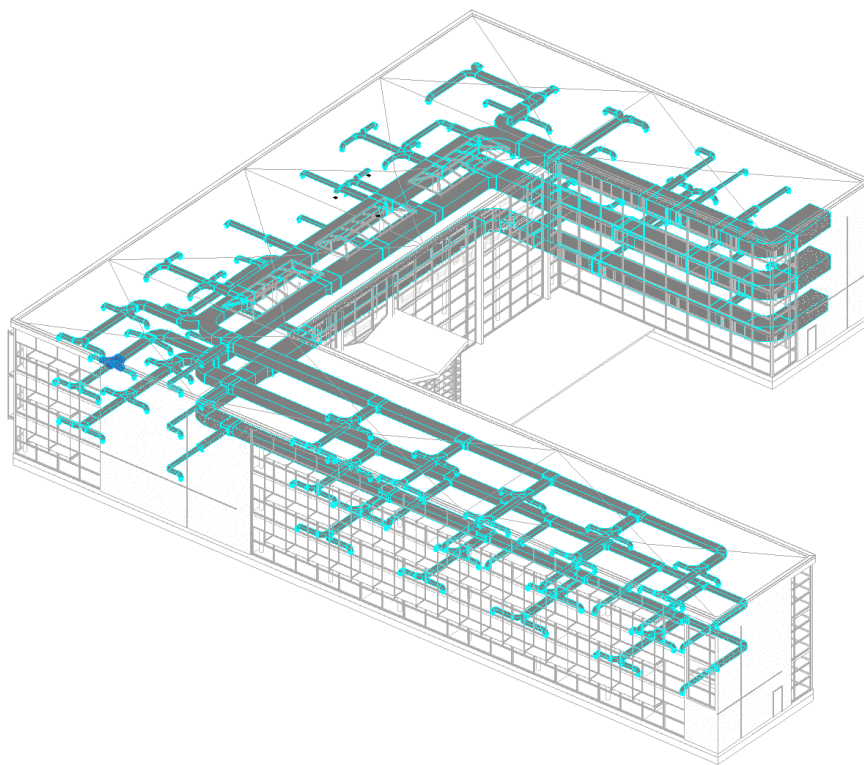
(a)一层平面图



(b)二层平面图



(c)三层平面图



(d)三维视图

图 6.7 测试建筑风管设计结果 Revit 模型

在多目标优化阶段，通过 NSGA-II 算法获得了一组 Pareto 最优解集，每个解代表在风管面积、系统阻力和不平衡性等性能目标之间不同的权衡选择。这些解在数学上互不支配，提供了多种可行的设计方案供工程实践参考。在实际工程中，设计人员可根据项目的优先目标进行选择：如在材料成本敏感项目中优先选择面积最小方案；在运行能效关键场景中倾向于选择阻力较低方案；而在系统调试与运行稳定性要求较高的项目中，优先考虑阻力均衡性较优的方案。此外，还可结合生命周期成本分析等经济性评估方法，对 Pareto 解集中的备选方案进行综合评分，从而确定最适宜的最终设计。本研究的价值在于提供了策略生成与多解筛选的协同框架，为实际设计提供科学的备选基础。

6.4 本章小结

本章主要通过案例验证了风系统管路自动生成和水力性能优化的效果。首先，在风管支管路径生成案例中，使用简单环境验证了支管生成结果。模型在简单环境下能够成功生成路径，体现了模型对局部空间结构的初步感知与策略学习能力。但在面对复杂环境时，模型的成功率较低，表明现有训练策略仍有提升空间。针对这一问题，提出了可以通过增加训练时间、优化奖励设计以及改进网络结构等策略，以提升模型在复杂环境中的表现。

在水力性能优化方面，本文提出的基于 NSGA-II 的多目标优化方法能够在管道面积、系统阻力与阻力均衡性三大性能指标之间实现有效的权衡。优化算法展现出良好的收敛能力与解集多样性，能够在满足工程约束的前提下搜索出结构合理、性能协调的风管配置方案。与传统规则设计方法相比，该方法在降低管道冗余、改善系统不平衡度方面具有明显优势，在多目标冲突较强的设计场景中表现出更优的全局搜索能力和工程适应性。最后，通过 Revit 二次开发接口，实现了根据优化选型结果自动生成风管布局的功能。

总体而言，本章的案例验证了本文所提出方法在风系统设计自动化与性能优化中的有效性，为实际工程设计提供了科学多样的决策支持，同时也揭示了当前方法在面对复杂任务时的性能瓶颈，明确了进一步改进路径策略泛化能力的研究方向。

第7章 结论与展望

7.1 结论

传统暖通风系统设计方法面临设计流程复杂、设计质量依赖人工经验和既定规则、难以同时满足多目标性能要求等突出挑战。针对这一问题，本文以风系统支管路径生成与管径水力优化为研究对象，提出了一套基于深度强化学习与多目标进化算法相结合的风系统管路自动化设计方法，并在典型建筑案例中进行了验证，取得了以下研究成果：

- (1) 针对风系统路径设计中的布局复杂性与目标多样性问题，构建了二维空间网格表示的强化学习训练环境，结合风口位置、障碍物分布等建筑要素，设计了具有实际工程意义的状态编码方法。提出了一种结合离散方向选择与连续步长调控的混合动作空间建模方式，有效提升了路径生成的表达能力与决策灵活性。在此基础上，设计了环境集生成机制，能够自动生成矩形、L型、T型等不同空间拓扑的房间样本，从而形成覆盖多种几何形态与路径规划难度的环境集。此外，提出了量化路径规划复杂度的难度评估指标体系，为后续的策略训练和性能分析提供了评价基础。
- (2) 根据支管路径的时序决策特性，构建了一种融合空间感知与时序建模能力的混合神经网络结构。模型采用坐标增强卷积网络提取空间状态特征，结合建筑尺寸信息的形状嵌入模块提升对房间几何变化的适应能力，并引入 LSTM 模块捕捉路径构建过程中的历史依赖特征。在强化学习训练方面，基于 PPO 算法构建完整的训练流程，在中低难度房间中取得较优的路径生成表现，初步验证了方法的可行性。
- (3) 提出了基于 NSGA-II 算法的风系统管路水力性能优化方法，构建了完整的多目标进化优化流程。基于风系统图结构，构建了管道水力计算模型与三目标优化框架（面积、阻力、不平衡度），设计了可行域构建、个体编码与修复机制，结合解码映射与性能评估函数，实现了进化过程中的自动约束处理与解质量控制。优化结果表明，该方法能够生成在多性能指标之间达到良好权衡的设计解集，优于传统基于假定流速法的设计结果。
- (4) 通过案例验证了风系统管路自动生成和水力性能优化的效果。在支管路径生成方面，模型在低复杂度房间中已初步学习到有效的空间策略。但在面对风口数量多、房型结构复杂的场景时，成功率显著下降，表明当前模型

在稀疏反馈与多目标控制下的泛化能力仍需提升。在水力性能优化方面，针对每层风系统拓扑图，使用基于 NSGA-II 算法的三目标优化框架，实现了对管道面积、系统阻力与阻力均衡度的协同权衡。实验结果表明，优化算法在多目标性能提升方面具有良好表现，优于传统假定流速法所得到的设计解，平均而言，在风管面积上改善 9.79%，在不平衡度上改善 40.01%。

7.2 局限性与展望

尽管本文提出的基于深度强化学习与多目标优化的风系统管路自动设计方法在多个方面取得了积极进展，并通过实际建筑案例初步验证了其可行性，但从整体实现效果来看，仍存在若干局限性，亟需在后续研究中进一步改进与拓展。

- (1) 复杂环境下的训练策略仍显不足。目前模型的训练主要针对结构简单、风口数量较少的房间环境，能够较好地学习路径延展策略。然而，在风口布局分散、障碍密集或形状不规则等复杂环境中的训练策略有待进一步研究。未来应加强对训练样本的覆盖性，优化奖励函数设计，引入策略引导机制或模仿专家示范，提高模型对复杂空间任务的适应能力。
- (2) 路径生成与水力性能尚未实现协同优化。由于建筑尺度大、空间网格细、状态空间维度高，直接联合建模会显著增加训练与求解难度，因此本研究将路径生成与水力优化拆分为两个阶段处理，并且主管布局未与支管布局协同进行。未来研究可考虑将主管与支管的布局、路径与管径优化纳入统一优化框架中，推动风系统整体性能的联合提升。
- (3) 未覆盖不同空调系统和风口类型。本研究主要针对顶部送风的办公室类空间进行建模，但在实际工程中存在不同类型的空调系统和风口形式。当前路径与管径设计方法缺乏对不同系统形式和末端的适配能力。未来可通过引入房间功能、末端风口类型等语义信息，扩展至多样化的空调系统类型与使用场景，提升模型的通用性与应用范围。
- (4) 集成管线碰撞规避与多系统协同机制。建筑风系统通常需与水、电、消防等多系统协同布置，存在空间冲突风险。目前本文聚焦于风系统内部的路径构建与性能优化，尚未纳入多专业之间的空间协同建模。未来可引入 BIM 模型中的空间体积信息，集成多专业模型进行空间关系解析与碰撞检测，实现动态避让与路径重构，提升方案的施工可行性与系统协调性，推动风水电一体化的智能协同设计方法研究。

参考文献

- [1] IEA. Tracking Clean Energy Progress 2023[R]. Paris: International Energy Agency, 2023.
- [2] U.S. Energy Information Administration. Commercial Buildings Energy Consumption Survey (CBECS), 2018[R]. Washington, D.C.: U.S. Department of Energy, 2022.
- [3] Pérez-Lombard L, Ortiz J, Pout C. A review on buildings energy consumption information[J]. Energy and Buildings, 2008, 40(3): 394-398.
- [4] Wang H, Xu P, Sha H, et al. BIM-based automated design for HVAC system of office buildings—An experimental study[J]. Building Simulation, 2022, 15(7): 1177-1192.
- [5] Chen Z, Guan H, Yuan X, et al. Rule-based generation of HVAC duct routing[J]. Automation in Construction, 2022, 139: 104264.
- [6] Liang R, Wang C, Wang P, et al. Realization of rule-based automated design for HVAC duct layout[J]. Journal of Building Engineering, 2023, 80: 107946.
- [7] Kabbara Z, Jorens S, Matbouli H, et al. Heuristic optimization for designing centralized air distribution systems in non-residential buildings[J]. Energy and Buildings, 2023, 292: 113161.
- [8] Kabbara Z, Jorens S, Seuntjens O, et al. Simulation-based optimization method for retrofitting HVAC ductwork design[J]. Energy and Buildings, 2024, 307: 113991.
- [9] Dong L, Lin B, Jiang Y, et al. Automatic Generation of Pipe Routing for Variable Refrigerant Flow Air Conditioning System[J]. Journal of Building Engineering, 2025, 100: 111755.
- [10] Jorens S, Sörensen K, Verhaert I, et al. Air distribution system design optimization in non-residential buildings: Problem formulation and generation of test networks[J]. Journal of Building Engineering, 2017, 12: 60-67.
- [11] Jorens S, Verhaert I, Sörensen K. Design optimization of air distribution systems in non-residential buildings[J]. Energy and Buildings, 2018, 175: 48-56.
- [12] Kabbara Z, Jorens S, Ahmadian E, et al. Improving HVAC ductwork designs while considering fittings at an early stage[J]. Building and Environment, 2023, 237: 110272.
- [13] 黄翔. 空调工程 第3版[M]. 北京: 机械工业出版社, 2017.
- [14] 王汉青. 通风工程 第2版[M]. 北京: 机械工业出版社, 2018.
- [15] Gao H, Koch C, Wu Y. Building information modelling based building energy modelling: A review[J]. Applied Energy, 2019, 238: 320-343.
- [16] Tang X, Zhang J, Liang R. The design of heating, ventilation, and air conditioning systems based on building information modeling: A review from the perspective of automatic and intelligent methods[J]. Journal of Building Engineering, 2024, 82: 108200.
- [17] Sha H, Xu P, Yang Z. IFC based semi-automated design tool for HVAC central system: A general framework[J]. IOP Conference Series: Earth and Environmental Science, 2019, 238(1): 012074.
- [18] Chong A, Xu W, Chao S, et al. Continuous-time Bayesian calibration of energy models using BIM and energy data[J]. Energy and Buildings, 2019, 194: 177-190.
- [19] Xu W, Chong A, Lam K P, et al. A New BIM To BEM Framework: The Development And Verification Of An Open-source gbXML To EnergyPlus Translator For Supporting Building Life Cycle Performance Analysis[C]. Proceedings of Building Simulation 2019: 16th

- Conference of IBPSA: Vol. 16. Rome, Italy: IBPSA, 2019: 2659-2666.
- [20] Bazjanac V, Maile T. IFC HVAC interface to EnergyPlus-A case of expanded interoperability for energy simulation[J]. 2004.
- [21] O'Sullivan B, Keane M. Specification of an IFC based intelligent graphical user interface to support building energy simulation[J]. 2005.
- [22] Yang Y, Pan Y, Zeng F, et al. A gbXML Reconstruction Workflow and Tool Development to Improve the Geometric Interoperability between BIM and BEM[J]. Buildings, 2022, 12(2).
- [23] Bracht M K, Melo A P, Lamberts R. A metamodel for building information modeling-building energy modeling integration in early design stage[J]. Automation in Construction, 2021, 121: 103422.
- [24] Trani M L, Ruschi M, Ripamonti F. Cross information and BIM interoperability tools for energy retrofit analysis[J]. IOP Conference Series: Earth and Environmental Science, 2021, 863(1): 012049.
- [25] Ying H, Lee S. A rule-based system to automatically validate IFC second-level space boundaries for building energy analysis[J]. Automation in Construction, 2021, 127: 103724.
- [26] Ying H, Lee S. Generating second-level space boundaries from large-scale IFC-compliant building information models using multiple geometry representations[J]. Automation in Construction, 2021, 126: 103659.
- [27] Zhang L, Chen Z, Ford V. Advancing building energy modeling with large language models: Exploration and case studies[J]. Energy and Buildings, 2024, 323: 114788.
- [28] Jiang G, Ma Z, Zhang L, et al. EPlus-LLM: A large language model-based computing platform for automated building energy modeling[J]. Applied Energy, 2024, 367: 123431.
- [29] Zhang L, Ford V, Chen Z, et al. Automatic building energy model development and debugging using large language models agentic workflow[J]. Energy and Buildings, 2025, 327: 115116.
- [30] Kamel E, Memari A M. Review of BIM's application in energy simulation: Tools, issues, and solutions[J]. Automation in Construction, 2019, 97: 164-180.
- [31] Bahar Y N, Pere C, Landrieu J, et al. A Thermal Simulation Tool for Building and Its Interoperability through the Building Information Modeling (BIM) Platform[J]. Buildings, 2013, 3(2): 380-398.
- [32] Jin R, Zhong B, Ma L, et al. Integrating BIM with building performance analysis in project life-cycle[J]. Automation in Construction, 2019, 106: 102861.
- [33] Trčka M, Hensen J L M. Overview of HVAC system simulation[J]. Automation in Construction, 2010, 19(2): 93-99.
- [34] Pargas-Carmona L A, Silva J A M D, Sant'Anna Â M O, et al. An optimization scheme for chiller selection in cooling plants[J]. Journal of Building Engineering, 2022, 49: 104066.
- [35] Torres Y D, Calvo R R, Herrera H H, et al. Procedure to obtain the optimal distribution cooling capacity of an air-condensed chiller plant for a hotel facility conceptual design[J]. Energy Reports, 2021, 7: 622-637.
- [36] Freeland D, Hall C, Nassif N. Optimizing the Design of Chilled Water Plants in Large Commercial Buildings[C]. Uzochukwu G A, Schimmel K, Kabadi V, et al. Proceedings of the 2013 National Conference on Advances in Environmental Science and Technology. Cham: Springer International Publishing, 2016: 187-194.
- [37] Feng F, Fu Y, Hou J, et al. Optimizing the topologies of heating, ventilation, and air-conditioning water systems in supertall buildings: A pilot study[J]. Science and Technology for

- the Built Environment, 2018, 24(4): 371-381.
- [38] Baradaran-Noveiri P, Liu H, Han S, et al. Parametric-based design optimization of air distribution system in panelized construction[J]. Journal of Building Engineering, 2022, 51: 104254.
- [39] Jin R, Li J, Wang H, et al. Automatic device arrangement and wiring for fire alarm systems in metro stations: A case study[J]. Journal of Building Engineering, 2024, 97: 110617.
- [40] Lee C Y. An Algorithm for Path Connections and Its Applications[J]. IRE Transactions on Electronic Computers, 1961, EC-10(3): 346-365.
- [41] Dijkstra E W. A note on two problems in connexion with graphs[J]. Numerische Mathematik, 1959, 1(1): 269-271.
- [42] Richard B. On a routing problem[J]. Quarterly of Applied Mathematics, 1958, 16(1): 87-90.
- [43] Ford L R. Network Flow Theory.[M]. Santa Monica, CA: RAND Corporation, 1956.
- [44] Dinitz Y, Itzhak R. Hybrid Bellman–Ford–Dijkstra algorithm[J]. Journal of Discrete Algorithms, 2017, 42: 35-44.
- [45] Jeong I K, Uddin J, Kang M, et al. Accelerating a Bellman–Ford Routing Algorithm Using GPU[C]. Park J J (Jong H, Zomaya A, Jeong H Y, et al. Frontier and Innovation in Future Computing and Communications. Dordrecht: Springer Netherlands, 2014: 153-160.
- [46] Hart P, Nilsson N, Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths[J]. IEEE Transactions on Systems Science and Cybernetics, 1968, 4(2): 100-107.
- [47] Stentz A. Optimal and efficient path planning for partially-known environments[C]. Proceedings of the 1994 IEEE International Conference on Robotics and Automation. 1994: 3310-3317 vol.4.
- [48] Stentz. The Focussed D* Algorithm for Real-Time Replanning[C]. 1995.
- [49] Koenig S, Likhachev M. D*lite[C]. Eighteenth National Conference on Artificial Intelligence. USA: American Association for Artificial Intelligence, 2002: 476-483.
- [50] Koenig S, Likhachev M, Furcy D. Lifelong Planning A*[J]. Artificial Intelligence, 2004, 155(1): 93-146.
- [51] Ferguson D, Stentz A. Field D*: An Interpolation-Based Path Planner and Replanner[C]. Thrun S, Brooks R, Durrant-Whyte H. Robotics Research. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007: 239-253.
- [52] Zhu D, Latombe J C. Pipe routing-path planning (with many constraints)[C]. Proceedings. 1991 IEEE International Conference on Robotics and Automation. 1991: 1940-1947 vol.3.
- [53] 马江涛, 刘检华, 徐联杰, 等. 基于 LTL-PRM 算法的分支管路自动布局方法[J]. 机械工程学报, 2018, 54(15): 160-170.
- [54] Farhan A, Gunawan, Yanuar. Pipe-routing optimization using system engineering methodology in ship engine room[C]. AIP Conference Proceedings: Vol. 2227. 2020: 020022.
- [55] Stanczak M, Pralet C, Vidal V, et al. A Pipe Routing Hybrid Approach Based on A-Star Search and Linear Programming[C]. Stuckey P J. Integration of Constraint Programming, Artificial Intelligence, and Operations Research. Cham: Springer International Publishing, 2021: 179-195.
- [56] Holland J H. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence[M]. The MIT Press, 1992.
- [57] Stützle T. Ant Colony Optimization[C]. Ehrgott M, Fonseca C M, Gandibleux X, et al. Evolutionary Multi-Criterion Optimization. Berlin, Heidelberg: Springer Berlin Heidelberg,

- 2009: 2-2.
- [58] Bonyadi M R, Michalewicz Z. Particle Swarm Optimization for Single Objective Continuous Space Problems: A Review[J]. *Evolutionary Computation*, 2017, 25(1): 1-54.
- [59] Liu Q, Wang C. A discrete particle swarm optimization algorithm for rectilinear branch pipe routing[J]. *Assembly Automation*, 2011, 31(4): 363-368.
- [60] Liu Q, and C W. Multi-terminal pipe routing by Steiner minimal tree and particle swarm optimisation[J]. *Enterprise Information Systems*, 2012, 6(3): 315-327.
- [61] Jiang W Y, Lin Y, Chen M, et al. An ant colony optimization-genetic algorithm approach for ship pipe route design[J]. *International Shipbuilding Progress*, 2014, 61: 163-183.
- [62] Jiang W Y, Lin Y, Chen M, et al. A co-evolutionary improved multi-ant colony optimization for ship multiple and branch pipe route design[J]. *Ocean Engineering*, 2015, 102: 63-70.
- [63] Wu L, Tian X, Wang H Y, et al. Improved ant colony optimization algorithm and its application to solve pipe routing design[J]. *Assembly Automation*, 2019, 39(1): 45-57.
- [64] Liu Q, Tang Z, Liu H, et al. Integrated Optimization of Pipe Routing and Clamp Layout for Aeroengine Using Improved MOALO[J]. *International Journal of Aerospace Engineering*, 2021, 2021(1): 6681322.
- [65] Ren T, Zhu Z L, Dimirovski G, et al. 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: 424-434.
- [66] Sui H, Niu W. Branch-pipe-routing approach for ships using improved genetic algorithm[J]. *Frontiers of Mechanical Engineering*, 2016, 11(3): 316-323.
- [67] Niu W, Sui H, Niu Y, et al. Ship Pipe Routing Design Using NSGA-II and Coevolutionary Algorithm[J]. *Mathematical Problems in Engineering*, 2016, 2016(1): 7912863.
- [68] Soltani A R, Tawfik H, Goulermas J Y, et al. Path planning in construction sites: performance evaluation of the Dijkstra, A*, and GA search algorithms[J]. *Advanced Engineering Informatics*, 2002, 16(4): 291-303.
- [69] Dong Z, Bian X. Ship Pipe Route Design Using Improved A* Algorithm and Genetic Algorithm[J]. *IEEE Access*, 2020, 8: 153273-153296.
- [70] Gunawan, Hamada K, Kunihiro K, et al. Automated Pipe Routing Optimization for Ship Machinery[J]. *Journal of Marine Science and Application*, 2022, 21(2): 170-178.
- [71] Dong Z ran, Bian X yi, Zhao S. Ship pipe route design using improved multi-objective ant colony optimization[J]. *Ocean Engineering*, 2022, 258: 111789.
- [72] Lin Y, Zhang Q. A multi-objective cooperative particle swarm optimization based on hybrid dimensions for ship pipe route design[J]. *Ocean Engineering*, 2023, 280: 114772.
- [73] Shakya A K, Pillai G, Chakrabarty S. Reinforcement learning algorithms: A brief survey[J]. *Expert Systems with Applications*, 2023, 231: 120495.
- [74] Watkins C J C H, Dayan P. Q-learning[J]. *Machine Learning*, 1992, 8(3): 279-292.
- [75] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. *Nature*, 2015, 518(7540): 529-533.
- [76] Sutton R S, Barto A G. Reinforcement Learning: An Introduction[J]. *IEEE Transactions on Neural Networks*, 1998, 9(5): 1054-1054.
- [77] Arulkumaran K, Deisenroth M P, Brundage M, et al. Deep Reinforcement Learning: A Brief Survey[J]. *IEEE Signal Processing Magazine*, 2017, 34(6): 26-38.
- [78] Huang H, Yang Y, Wang H, et al. Deep Reinforcement Learning for UAV Navigation Through

- Massive MIMO Technique[J]. IEEE Transactions on Vehicular Technology, 2020, 69(1): 1117-1121.
- [79] Nian R, Liu J, Huang B. A review On reinforcement learning: Introduction and applications in industrial process control[J]. Computers & Chemical Engineering, 2020, 139: 106886.
- [80] Kiran B R, Sobh I, Talpaert V, et al. Deep Reinforcement Learning for Autonomous Driving: A Survey[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(6): 4909-4926.
- [81] Singh B, Kumar R, Singh V P. Reinforcement learning in robotic applications: a comprehensive survey[J]. Artificial Intelligence Review, 2022, 55(2): 945-990.
- [82] Noaeen M, Naik A, Goodman L, et al. Reinforcement learning in urban network traffic signal control: A systematic literature review[J]. Expert Systems with Applications, 2022, 199: 116830.
- [83] Chen X, Qu G, Tang Y, et al. Reinforcement Learning for Selective Key Applications in Power Systems: Recent Advances and Future Challenges[J]. IEEE Transactions on Smart Grid, 2022, 13(4): 2935-2958.
- [84] Le N, Rathour V S, Yamazaki K, et al. Deep reinforcement learning in computer vision: a comprehensive survey[J]. Artificial Intelligence Review, 2022, 55(4): 2733-2819.
- [85] Zhou S K, Le H N, Luu K, et al. Deep reinforcement learning in medical imaging: A literature review[J]. Medical Image Analysis, 2021, 73: 102193.
- [86] Sun C, Tol R van der, Melenhorst R, et al. Path planning of manure-robot cleaners using grid-based reinforcement learning[J]. Computers and Electronics in Agriculture, 2024, 226: 109456.
- [87] Li K, Liu T, Kumar P N R, et al. A reinforcement learning-based hyper-heuristic for AGV task assignment and route planning in parts-to-picker warehouses[J]. Transportation Research Part E: Logistics and Transportation Review, 2024, 185: 103518.
- [88] Peng W, Wang D, Yin Y, et al. Multi-agent deep reinforcement learning-based truck-drone collaborative routing with dynamic emergency response[J]. Transportation Research Part E: Logistics and Transportation Review, 2025, 195: 103974.
- [89] Baneshi F, Cerezo-Magaña M, Soler M. Network-level aircraft trajectory planning via multi-agent deep reinforcement learning: Balancing climate considerations and operational manageability[J]. Expert Systems with Applications, 2025, 271: 126604.
- [90] Ho G T S, Tang Y M, Leung E K H, et al. Integrated reinforcement learning of automated guided vehicles dynamic path planning for smart logistics and operations[J]. Transportation Research Part E: Logistics and Transportation Review, 2025, 196: 104008.
- [91] Xue W, Jia N, Zhao M. Multi-agent deep reinforcement learning based HVAC control for multi-zone buildings considering zone-energy-allocation optimization[J]. Energy and Buildings, 2025, 329: 115241.
- [92] Yin L, Xiong Y. Incremental learning user profile and deep reinforcement learning for managing building energy in heating water[J]. Energy, 2024, 313: 133705.
- [93] Hu Z, Gao Y, Sun L, et al. A novel reinforcement learning method based on generative adversarial network for air conditioning and energy system control in residential buildings[J]. Energy and Buildings, 2025, 336: 115564.
- [94] Kumaresan S S, Jeyaraj P R. Smart building transferable energy scheduling employing reward shaping deep reinforcement learning with demand side energy management[J]. Journal of Building Engineering, 2025, 104: 112316.
- [95] Madani S, Pineau P O, Charlin L, et al. Towards Sustainable Energy Use: Reinforcement Learning for Demand Response in Commercial Buildings[J]. Energy and Buildings, 2025:

- 115721.
- [96] Chen Y, Zheng Z, Lin J R. Automatic Design Method of Building Pipeline Layout Based on Deep Reinforcement Learning[C]. 30th International Conference on Intelligent Computing in Engineering. London, UK, 2023: 42-51.
- [97] Qi Z, Jin R, Li J, et al. BIM-Based Automated Multi-Air Distribution Layout Generation for Office Buildings: A Case Study[J]. Buildings, 2023, 13(7): 1819.
- [98]民用建筑供暖通风与空气调节设计规范: GB 50736-2012[S]. 北京: 中国建筑工业出版社, 2012.
- [99]张国栋. 通风空调工程概预算手册(第二版)[M]. 北京: 中国建筑工业出版社, 2014.
- [100] Schulman J, Wolski F, Dhariwal P, et al. Proximal Policy Optimization Algorithms[J]. arXiv preprint arXiv:1707.06347, 2017.
- [101] 陆耀庆. 实用供热空调设计手册[M]. 北京: 中国建筑工业出版社, 2008.
- [102] de Boor C. A Practical Guide to Spline[M]. New York: Springer, 1978.
- [103] Srinivas N, Deb K. Multiojective Optimization Using Nondominated Sorting in Genetic Algorithms[J]. Evolutionary Computation, 1994, 2(3): 221-248.

附录 A 支管路径生成环境代码（节选）

```
import numpy as np
import torch
import torch.nn.functional as F
from typing import Tuple, List, Dict
import networkx as nx

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

class State:
    def __init__(self, roomEnv: "RoomEnv") -> None:
        self.grid = roomEnv.grid_world

    @staticmethod
    def makeBatch(states: List["State"]) -> torch.Tensor:
        # 获取当前批次最大空间尺寸
        max_D = max(state.grid.size(0) for state in states)
        max_H = max(state.grid.size(1) for state in states)
        processed = []
        for state in states:
            tensor = state.grid # 原始形状 (H, W, C)
            D, H, W = tensor.shape
            pad_D = max_D - D
            pad_H = max_H - H
            if pad_D > 0 or pad_H > 0:
                pad = (0, 0, 0, pad_H, 0, pad_D)
                padded = F.pad(tensor, pad, mode='constant', value=0)
                padded[D:, :, 0] = 1.0
                padded[:, H:, 0] = 1.0
            else:
                padded = tensor
```

```

        processed.append(padded)
    return torch.stack(processed)

class Action:
    action_space = ["↑", "↓", "←", "→"]
    directions = np.array([[-1, 0], [1, 0], [0, -1], [0, 1]]) # 上下左右

    def __init__(self, direction: int, value: float) -> None:
        self.direction = torch.tensor([direction]).to(device).long().view(1)
        self.value = torch.tensor([value]).to(device).float().view(1)

    def __str__(self) -> str:
        return f'{self.action_space[self.direction.item()]}: {self.value.item():.2f}'

    @staticmethod
    def randomAction() -> "Action":
        direction = random.randint(0, 3)
        value = random.random() # 随机生成 0 到 1 之间的浮点数
        return Action(direction, value)

    @staticmethod
    def makeBatch(actions: List["Action"]) -> torch.Tensor:
        directions = torch.cat([action.direction for action in actions], dim=0)
        values = torch.cat([action.value for action in actions], dim=0)
        return directions, values # 返回两个张量

class RoomEnv:
    def __init__(self, grid: torch.Tensor, info: Dict = {}):
        defaults = {
            'operation_years': 20,
            'grid_size': 0.1,
            'minxy': (0, 0),
            'difficulty': 0,
            'id': 0

```

```

    }
    for key, default in defaults.items():
        setattr(self, key, info.get(key, default))
    self.resistance_tolerance = 0.15
    self.tensor_size: Tuple[int, int, int] = tuple(self.grid_world.shape)
    self.height: int = self.tensor_size[0]
    self.width: int = self.tensor_size[1]
    self.channels: int = 8
    self.t_max = 10
    self.entry_direction = 'v' if torch.all((e_y := torch.where(grid[..., self.ENTRY]
== 1)[1]) == e_y[0]) else 'h'
    self.reset()

def reset(self):
    self.total_reward = 0
    self.t = 0
    self.connect_entry = False
    self.current_state = State(self)
    self.reset_grid()

def reset_grid(self):
    # 初始化 route
    grid = self.grid_world
    mask_route = grid[..., self.ROUTE] == 1
    mask_current = grid[..., self.CURRENT] == 1
    grid[..., self.ROUTE][mask_route] = 0    # ROUTE 通道置零
    grid[..., self.CURRENT][mask_current] = 0 # CURRENT 通道置零
    grid[..., self.ROOMSPACE][mask_current] = 1 # ROOMSPACE 通道置
    grid[..., self.ROOMSPACE][mask_route] = 1 # ROOMSPACE 通道置一
    self.route_grid = torch.zeros(self.height, self.width, dtype=torch.int32,
device=device)

```

附录 B 管路水力性能优化代码（节选）

```
DUCT_SIZE = [120, 160, 200, 250, 320, 400, 500, 630, 800, 1000, 1250, 1600, 2000,
2500, 3000, 3500, 4000]
L_W_RATIO_MAX = 3
```

```
def get_feasible_configs_for_edge(graph, u, v, duct_size = DUCT_SIZE,
ratio_max=L_W_RATIO_MAX):
```

```
    edge_data = graph[u][v]
    flowrate = edge_data.get('flowrate', 0)
    v_max = velocity_max(graph, u, v)
    Q = flowrate / 3600.0
    A_req = Q / v_max
    threshold = A_req * 1e6
    feasible = []
    for L in duct_size:
        for W in duct_size:
            if L >= W:
                if (L / W) < ratio_max:
                    if L * W >= threshold:
                        feasible.append((L, W))
    return feasible
```

```
def evaluate_design(graph):
```

```
    total_area = 0
    for u, v, data in graph.edges(data=True):
        length = data.get('length')
        size = data.get('size')
        total_area += (size[0]+size[1])/1000*2*length
    values = [data.get('total_resistance', 0)
               for node, data in graph.nodes(data=True)
               if data.get('type') == 'Diffuser']
```

```

mean_val = sum(values) / len(values)
variance = sum((x - mean_val) ** 2 for x in values) / len(values)
std_dev = math.sqrt(variance)
return total_area, max(values), std_dev/mean_val

# 编码个体：将图中各边的管径编码为个体基因
def init_individual(icls, feasible_domains, edge_list, graph_org):
    individual = []
    for edge in edge_list:
        domain = feasible_domains.get(edge, [])
        if domain:
            individual.append(random.randint(0, len(domain) - 1))
        else:
            individual.append(0)
    repaired = repair_operator_feasible(individual, graph_org, edge_list,
feasible_domains)
    return icls(repaired)

# 解码个体：把个体中的基因映射到图中各边的管径设计
def decode_individual(individual, graph_org, edge_list, feasible_domains):
    graph = copy.deepcopy(graph_org)
    for i, edge in enumerate(edge_list):
        domain = feasible_domains.get(edge, [])
        if domain:
            config = domain[individual[i]]
            u, v = edge
            graph[u][v]['size'] = config
    update_by_size(graph)
    return graph

# 交叉算子：均匀交换各个位置基因
def cx_uniform_feasible(ind1, ind2, indpb):
    for i in range(len(ind1)):
        if random.random() < indpb:

```

```

        ind1[i], ind2[i] = ind2[i], ind1[i]
    return ind1, ind2

# 变异算子：每个基因以一定概率随机变为在该边可行域中的另一个索引
def mut_uniform_feasible(individual, feasible_domains, edge_list, indpb):
    for i in range(len(individual)):
        domain = feasible_domains.get(edge_list[i], [])
        if domain and random.random() < indpb:
            individual[i] = random.randint(0, len(domain)-1)
    return individual

# NSGA-II 优化算法
def nsga2_optimize(graph_org, ngen=1000, pop_size=300, cxpb=0.9, mutpb=0.1,
                   early_stop_patience=20, early_stop_threshold=1e-5,):
    edge_list = list(graph_org.edges())
    feasible_domains = get_all_edges_feasible_domains(graph_org)
    toolbox.register("select", tools.selNSGA2)
    toolbox.unregister("individual")
    toolbox.register("individual", lambda: init_individual(creator.Individual,
feasible_domains, edge_list, graph_org))
    toolbox.unregister("evaluate")
    toolbox.register("evaluate", eval_individual, graph_org=graph_org, edge_list =
edge_list, feasible_domains = feasible_domains)
    toolbox.unregister("mutate")
    toolbox.register("mutate", mut_uniform_feasible, feasible_domains =
feasible_domains, edge_list = edge_list, indpb=0.1)
    toolbox.register("population", tools.initRepeat, list, toolbox.individual)
    ref_point = compute_reference_point(graph_org, feasible_domains)
    normalized_ref_point = np.array([1.1, 1.1, 1.1])
    hv_calculator = HV(ref_point=normalized_ref_point)
    hv_history = []
    no_improve_count = 0
    best_hv = -float('inf')
    pop = toolbox.population(n=pop_size)

```

```

for ind in pop:
    ind.fitness.values = toolbox.evaluate(ind)
for gen in range(ngen):
    offspring = toolbox.select(pop, len(pop))
    offspring = list(map(toolbox.clone, offspring))
    for child1, child2 in zip(offspring[::2], offspring[1::2]):
        if random.random() < cxpb:
            toolbox.mate(child1, child2)
            del child1.fitness.values, child2.fitness.values
    for mutant in offspring:
        if random.random() < mutpb:
            toolbox.mutate(mutant)
            del mutant.fitness.values
    invalid_ind = [ind for ind in offspring if not ind.fitness.valid]
    for ind in invalid_ind:
        ind.fitness.values = toolbox.evaluate(ind)
    pop = toolbox.select(pop + offspring, pop_size)
    current_front = tools.sortNondominated(pop, k = len(pop), first_front_only =
True)[0]
    front_obj = [ind.fitness.values for ind in current_front]
    front_array = np.array(front_obj, dtype=float)
    normalized_front = front_array / np.array(ref_point)
    current_hv = hv_calculator.do(normalized_front)
    hv_history.append(current_hv)
    if current_hv - best_hv < early_stop_threshold:
        no_improve_count += 1
    else:
        best_hv = current_hv
        no_improve_count = 0
    if no_improve_count >= early_stop_patience:
        break
    pareto_front = tools.sortNondominated(pop, k=len(pop), first_front_only=True)[0]
    return pareto_front, gen, hv_history

```

致谢

转眼间到了硕士研究生生涯的最后阶段，三年的时光是如此短暂又漫长，短暂到我仍然清晰而真切地记得每一个欢乐、惊喜、难忘和眼含热泪的瞬间，漫长到我无法再回想起任何一个独自对着发光的屏幕思考、创造和抓耳挠腮的夜晚。但是相信所有经历过的艰辛和痛苦都将会成为滋养我终生的养分，为我铸造新的血肉。

感谢导师的指导和充分信任。感谢课题组的伙伴对我学习和生活上的照顾。感谢生命旅途中所有给予我关心和帮助的朋友们。

感谢我的父母，在所有人都希望我解决问题、出成果的时候，你们只是希望我能多吃点饭。

感谢在我的耳机里一直陪伴我的音乐人们，让我所有渺小、脆弱、敏感的情绪都有地方安放，让我普通而平凡的生活不至于了无生趣。

感谢不期而遇的人们。我们收藏好每个片刻珍贵的记忆，未来再一起分享。

2025 年 5 月

个人简历、在读期间发表的学术成果

个人简历:

李俊杰, 男, 2000 年 12 月生。

2022 年 6 月毕业于同济大学 建筑环境与能源应用工程专业 获学士学位。

2022 年 9 月入同济大学攻读硕士学位。

已发表论文:

- [1] Li J, Guan H, Xu P. Automatic design of HVAC water system pipelines based on multi-agent ant colony algorithm[C]. Proceedings of ASim Conference 2024: 5th Asia Conference of IBPSA: Vol. 5. Osaka, Japan: IBPSA-Asia, 2024: 182-189.
- [2] Jin R, Li J, Wang H, et al. Automatic device arrangement and wiring for fire alarm systems in metro stations: A case study[J]. Journal of Building Engineering, 2024, 97: 110617.
- [3] Qi Z, Jin R, Li J, et al. BIM-Based Automated Multi-Air Distribution Layout Generation for Office Buildings: A Case Study[J]. Buildings, 2023, 13(7): 1819.
- [4] 贾乾然, 许鹏, 李俊杰. 一种基于深度学习的空调机房设备状态检测算法[J]. 建筑科学, 2023, 39(02): 162-170.

研究及项目经历:

- [1] 基于轨道交通车站 BIM 的暖通自动化研究, 横向课题, 主要参与人
- [2] 基于 BIM 的地铁车站火灾报警系统自动化设计及环控柜原理图自动化设计研究, 横向课题, 主要参与人
- [3] 基于 BIM 的多联机系统自动化设计, 横向课题, 参与人