

基于复杂网络的管道巡检机器人路径规划方法

赵陈浩, 余水龙*

北华大学电气与信息工程学院, 吉林 吉林

收稿日期: 2026年4月23日; 录用日期: 2026年5月22日; 发布日期: 2026年5月29日

摘要

管道内巡检机器人的路径规划是降低巡检能耗与时间成本的关键环节。为解决管道全覆盖巡检问题, 文章提出了一种基于经典中国邮递员问题(CPP)框架的系统性工程解决方案。该方法首先将管道结构抽象为复杂网络模型, 随后提取其最小生成树以获取网络的最优连通骨架。在此基础上, 针对原始网络图进行欧拉化改造, 通过匹配奇度节点并添加权值最小的重复边, 将非欧拉图转化为欧拉图。最终引入Hierholzer算法求解欧拉回路, 生成一条能够无遗漏覆盖所有管道且总权值最小的巡检路径。实验结果表明, 与传统路径规划策略相比, 该方法显著提升了巡检机器人的路径利用率与运行性能, 为管道巡检任务提供了一种高效、可行的技术方案。

关键词

管道, 巡检机器人, 复杂网络, 路径规划

Path Planning Method for Pipeline Inspection Robots Based on Complex Networks

Chenhao Zhao, Shuilong Yu*

College of Electrical and Information Engineering, Beihua University, Jilin Jilin

Received: April 23, 2026; accepted: May 22, 2026; published: May 29, 2026

Abstract

Path planning for in-pipe inspection robots is a key technology for reducing energy consumption and time costs during inspection tasks. To address the full coverage inspection problem in pipelines, this paper presents a systematic engineering solution based on the classic Chinese Postman Problem (CPP)

*通讯作者。

framework. In this approach, the pipeline structure is first modeled as a complex network. The minimum spanning tree of the network is then extracted to obtain the optimal connected backbone. Based on this, an Eulerian transformation is applied to the original network graph: by matching odd-degree nodes and adding minimal-weight duplicated edges, the non-Eulerian graph is converted into an Eulerian graph. Finally, the Hierholzer algorithm is employed to find an Eulerian circuit, generating an inspection route that covers all pipeline segments without omission and minimizes total weight. Experimental results demonstrate that, compared with traditional methods, the proposed solution significantly enhances path efficiency and operational performance of inspection robots, offering an effective and feasible technical solution for pipeline inspection applications.

Keywords

Pipeline, Inspection Robot, Complex Network, Path Planning

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

管道内巡检机器人已经被广泛应用于供水管道[1]、天然气管道[2]等管网的定期检查与维护工作,以此保障管道运行安全。而管内巡检机器人的路径规划是该领域的重要研究方向,涉及如何在复杂的管道环境中实现高效、安全的机器人导航。路径规划方法主要分为全局路径规划方法和局部路径规划方法两类。全局路径规划主要用于已知环境中,旨在找到从起点到目标点的最优路径。其研究经历了不同阶段的发展:在早期,Dijkstra算法和A算法为静态环境下的最短路径规划奠定了基础,其中A算法通过引入启发式函数提高了搜索效率。随后,人工势场法通过引力和斥力实现路径规划,但存在局部最优的问题,研究重点在于改进方法。遗传算法通过模拟自然选择优化路径规划,适应性较强。进入2010年后,粒子群优化算法模拟群体行为,实现了全局路径优化,而深度强化学习结合动态环境,自适应地规划路径,成为近年来的热点。多目标路径规划则侧重于在复杂环境中优化多个约束条件的路径选择,推动了这一领域的发展[3]-[5]。Park等学者提出了一种管道内机器人在不同管道结构和直径下的导航方法,并给出了用于估计机器人在管道内位置和方向的跟踪算法[6]。

局部路径规划方法的研究随着时间的推移不断发展,呈现出多样化的特点。在早期,研究主要聚焦基于人工势场法的路径规划,相关论文探讨了改进势场函数以提升算法性能。随后,动态窗口法(DWA)成为研究热点,通过实时计算机器人运动学模型优化路径规划,提高了算法的实时性和避障能力。进入2010年后,时间弹性带(TEB)算法逐渐受到关注,其通过优化路径的时间和空间分布生成平滑轨迹,相关研究集中在复杂环境中的应用效果。近年来,深度强化学习技术被广泛应用于局部路径规划,相关论文探讨了机器人如何通过与环境交互学习最优策略,特别是在动态和未知环境中的表现。此外,改进的快速搜索随机树(RRT)算法也成为研究重点,通过路径后处理和平滑技术提升路径规划的效率和质量。Kazeminasab等学者提出了一种基于粒子滤波与两相运动控制器的管道机器人智能导航方法。通过粒子滤波算法对管道构型进行识别,准确区分直线段与非直线段。在直线路径导航阶段,机器人采用线性二次型调节器结合比例-积分-导数控制器实现运动稳定性控制与速度跟踪;在非直线路径导航阶段,则通过运动轨迹生成器规划最优路径,引导机器人沿预定轨迹行进。该方法通过双模控制策略实现了管道环境下的精准导航[7]。从复杂网络的视角审视管道全覆盖巡检问题,其本质可

归结为图论中的经典问题——中国邮递员问题(Chinese Postman Problem, CPP), 即寻求遍历网络中所有边至少一次且总权值最小的回路。当网络本身为欧拉图时, 最优解即为其欧拉回路; 当网络为非欧拉图时, 则需通过复制部分边将其转化为欧拉图, 再进行求解[8][9]。本文正是基于这一经典理论框架, 针对管道巡检的具体工程需求, 提出了一套系统性解决方案。该方案首先构建管网的复杂网络模型, 通过最小生成树确定网络的拓扑骨架, 随后对原始图进行欧拉化改造, 最终利用欧拉回路算法生成全覆盖的最优巡检路径。这一工作为 CPP 在管道机器人路径规划领域的工程化应用提供了完整的实现范式。

2. 路径规划方法

2.1. 管道复杂网络模型构建

复杂网络是一种用于描述和分析复杂系统中元素及其关系的数学工具, 它通过将系统中的个体抽象为节点, 个体间的相互作用抽象为边, 构建出一个网络结构。复杂网络通常具有小世界特性、无标度特性以及社团结构。经典模型包括随机网络模型, 用于简单随机系统; 小世界网络模型, 兼具高聚类性和短路径特性; 无标度网络模型, 反映“强者愈强”的特性。复杂网络模型在社会网络、生物网络和技术网络等领域有广泛应用, 能够帮助研究者理解系统的整体行为和局部特性, 为解决实际问题提供理论支持[10][11]。

为了构建管道复杂网络模型, 首先将管道中的交接点(如交叉点或终端点)定义为网络的节点, 如图 1 所示。以实际管道为网络的边, 连接相邻的交接点, 并记录其长度、材质等属性; 然后, 根据通行难度为每条边分配权值, 管道最常见的两种架设方式分为两种, 一种是水平架设, 另一种是垂直架设, 如图 2 所示。以线路的长度为权值基础, 将垂直管道的通行难度设置为水平管道的 3 倍, 权值越高表示通行难度越大。利用邻接矩阵构建复杂网络模型, 节点表示交接点, 边表示管道, 权值标注在边上, 最终形成网络结构。

特别地, 关于边权值的设定依据, 本研究将垂直管道的通行难度设置为水平管道的 3 倍, 这一参数选择具有明确的物理意义与工程背景。对于轮式或履带式管道机器人而言, 在水平管道中行驶时主要克服滚动摩擦阻力, 能耗相对较低; 而在垂直管道中爬升时, 机器人不仅需要克服重力势能做功, 还需提供足够的附着力以防止下滑, 其电机输出扭矩与能量消耗通常呈指数级增长。根据文献[7]实际电机驱动模型估算, 垂直爬升的单位距离能耗约为水平直行的 2.5 至 3.5 倍。为了在不失一般性的前提下简化模型并突出垂直障碍对路径选择的惩罚效应, 本文选取了折中的 3 倍权重系数。这一设定确保了最小生成树算法在连接节点时会优先选择水平铺设的管道分支, 从而生成更符合实际低能耗巡检需求的拓扑骨架。

从复杂网络的角度, 上述管道网络可形式化定义为加权无向网络 $G=(V,E,W)$, 其中, $V=\{v_1,v_2,\dots,v_n\}$ 为节点集, 对应管道系统中的交接点或终端; $E=\{e_1,e_2,\dots,e_m\}\subseteq V\times V$ 为边集, 对应连接相邻节点的管道段; $W:E\rightarrow\mathbb{R}^+$ 为权值函数, 表示机器人通过该段管道的难度或能耗成本。对于任意边 $e=(v_i,v_j)\in E$, 其权值由管道架设方式决定:

$$w(e)=\begin{cases} L(e), & \text{水平架设} \\ 3*L(e), & \text{垂直架设} \end{cases}$$

其中, $L(e)$ 为边 e 所对应管道的实际物理长度。该权值设定体现了垂直管道爬升所需克服的重力势能与额外附着力需求。网络 G 的拓扑结构可用邻接矩阵 $A=[a_{ij}]_{n\times n}$ 表示, 其元素定义为:

$$a_{ij}=\begin{cases} w(v_i,v_j), & \text{若}(v_i,v_j)\in E \\ 0, & \text{若}i=j\text{或}(v_i,v_j)\notin E \end{cases}$$

通过上述建模过程, 原始管道物理布局被转化为可供复杂网络算法处理的数学结构, 为后续最小生成树识别与欧拉回路求解奠定了形式化基础。

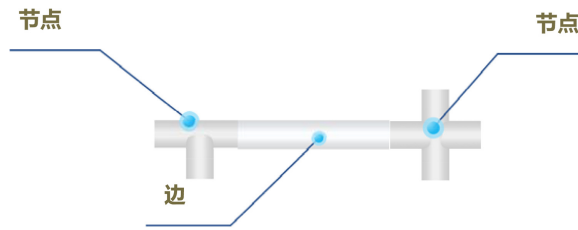


Figure 1. Nodes and edges of the pipeline network model

图 1. 管道网络模型的节点和边

图 1 展示了管道物理结构与复杂网络模型之间的映射关系。在实际管道系统中, 管道的交汇点、分叉口或终端被抽象为网络中的“节点”(如图中蓝色圆点所示), 而连接这些节点的实际管道段则被抽象为网络中的“边”(如图中灰色连线所示)。通过这种抽象, 复杂的物理管网被简化为一个由节点和边构成的拓扑结构, 为后续运用复杂网络分析算法奠定了基础。

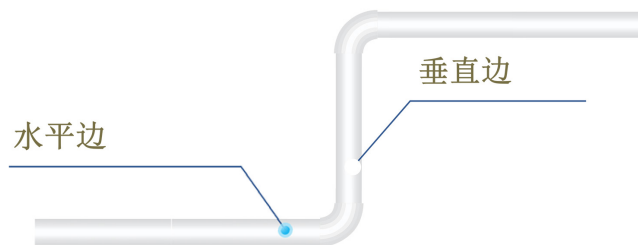


Figure 2. Types of pipeline installation

图 2. 管道架设类型

图 2 示意了管网中两种典型的管道架设方式: 水平架设与垂直架设。在复杂网络模型中, 这两种架设方式对应着不同的边权值。由于机器人在垂直管道中爬升需要克服重力并消耗更多能量, 本研究将垂直管道的权值设定为相同长度水平管道的 3 倍。这一权值差异化的处理, 使得后续的路径优化算法能够有效规避高能耗的垂直通道, 倾向于选择更为经济的水平路径。

2.2. 最小生成树识别

采用 Kruskal 算法识别最小生成树。该算法通过逐步将权重最小的边加入生成树中, 构建一棵权值最小的生成树。

- 1) 将图中的所有边按权值从小到大排序。
- 2) 初始化生成树为空。
- 3) 遍历排序后的边。
- 4) 如果加入当前边不会形成环(通过并查集判断), 则将其加入生成树。

重复此过程直到生成树包含 $|V| - 1$ 条边, 此时 T 即为图 G 的最小生成树。

设图 $G = (V, E)$, 其中 V 是节点集合, E 是边集合。目标是找到权值和最小的边子集 $T \subseteq E$, 使得 T 连接 V 中的所有节点且不形成环。每条边 $e \in E$ 具有权值 $w(e)$ 。最小生成树 $T \subseteq E$ 的数学目标为:

$$\min_T \sum_{e \in T} w(e)$$

约束条件为： T 是 G 的一个连通无环子图，且包含所有节点 V 。

Kruskal 算法的执行过程可形式化描述如下：

1) 将边集 E 按权值 $w(e)$ 升序排序，得到序列 $\{e_1, e_2, \dots, e_m\}$ ，其中 $w(e_1) \leq w(e_2) \leq \dots \leq w(e_m)$ 。

2) 初始化 $T = \emptyset$ ，为每个节点 $v \in V$ 创建独立的并查集。并查集数据结构通过两个核心操作实现环的高效检测：**Find** (x) 返回节点 x 所属连通分量的代表元，**Union** (x, y) 将两个连通分量合并。采用路径压缩和按秩合并优化后，每次 **Find** 和 **Union** 操作的均摊时间复杂度接近常数级 $O(\alpha(|V|))$ ，其中 α 为反阿克曼函数。因此，Kruskal 算法处理边集的时间复杂度主要由排序步骤决定，为 $O(|E| \log |E|)$ 。

3) 依次遍历每条边 $e_i = (u, v)$ ：

- 若节点 u 和 v 不属于同一连通分量(即 $\text{Find}(u) \neq \text{Find}(v)$)，则将 e_i 加入 T ，并合并两分量(**Union** (u, v))。
- 否则跳过该边。

4) 重复步骤 3 直至 $|T| = |V| - 1$ 。

该算法的正确性基于贪心选择性质，即每一步选取当前可选边中权值最小且不构成环的边，最终得到的 T 必为图 G 的最小生成树。

2.3. 优化遍历路径

为了实现所有管道段的无遗漏高效遍历，首先提取网络的最小生成树作为最优连通骨架。然而，最小生成树本身通常并非欧拉图，无法直接提供不重复遍历所有边的欧拉回路。因此，下一步需对原始网络图进行欧拉化改造，而非对生成树本身。具体而言，识别原始图中度数为奇数的节点，通过添加重复边的方式将这些奇度节点两两配对连接，使得所有节点度数均变为偶数，从而将原非欧拉图转化为欧拉图。重复边的选择以总权值最小为原则，以保证最终巡检路径的总成本最低。由于最小生成树是连通无环网络且为非欧拉图，其中部分节点的度数可能为奇数，因此需找到所有奇度节点，并以两两配对的方式添加额外的边，从而使每个节点的度数均为偶数。为尽可能降低路径总权值，这些新增的边应该选择权值最小的路径。在构造网络成为欧拉图后，应用 Hierholzer 算法，该算法通过迭代性地构建欧拉回路并回溯填充路径，确保每条边仅访问一次且最终形成一个闭合的回路，从而实现起点和终点一致的最优路径遍历，该过程精准平衡了节点和边的访问次数与路径权值的最小化。从计算复杂度的角度分析，本文所提出的路径规划方法主要包含三个步骤：最小生成树求解、奇度节点匹配及欧拉回路构建。采用 Kruskal 算法求解最小生成树的时间复杂度为 $O(|E| \log |E|)$ ；奇度节点最小权完美匹配可通过带权匹配算法(如 Edmonds 算法)在 $O(|V|^3)$ 时间内完成；Hierholzer 算法求解欧拉回路的时间复杂度为 $O(|E|)$ 。由于管道网络通常为稀疏网络($|E| \approx |V|$)，整体算法在包含百余节点的中等规模管网中可在毫秒级时间内完成规划，完全满足实际工程中离线全局规划或在线重规划的实时性需求。

将最小生成树 T 转化为欧拉图的过程可借助复杂网络中的匹配理论进行严格描述。首先，找出 T 中所有度数为奇数的节点，记为集合 $O \subseteq V$ 。根据握手引理， $|O|$ 必为偶数。为使 T 成为欧拉图，需在 O 中的奇度节点之间添加一组边，使得每个节点的度数均变为偶数，且新增边的总权值最小。该问题等价于在完全网络 $K_{|O|}$ (节点集为 O ，边权为原网络中对应节点间的最短路径长度)上求解最小权完美匹配。为获取任意两节点间的最短路径长度，本文采用 Floyd-Warshall 算法计算原网络 G 中所有节点对的最短路径距离。对于规模为 $|V|$ 的管道网络，该算法的时间复杂度为 $O(|V|^3)$ ，鉴于管网通常为稀疏网络且节点数有限，该计算在工程上完全可行。记 $D = \{d(u, v)\}$ 为求得的距离矩阵，其中 $d(u, v)$ 即为节点 u 与 v 之间的最短路权值。设匹配结果为 $M = \{(u_1, v_1), (u_2, v_2), \dots, (u_k, v_k)\}$ ，其中 $k = |O|/2$ 。对于每一对匹配节点 (u_i, v_i) ，在 T 中添加其对应最短路径上的所有边(若边已存在，则视为重复边)，从而得到

欧拉图 G' 。具体操作上, 对于匹配对 (u_i, v_i) 的最短路径 $P_i = \{e_{i1}, e_{i2}, \dots, e_{ik}\}$ 若某条边 e_{ij} 在原图 G 中仅有一条实例, 则将其复制一份作为平行边加入图 G' ; 若该边已被之前的匹配对复制过, 同样再次复制。这一处理确保了最终图 G' 中所有节点的度数均为偶数, 满足欧拉图的存在性条件。记所有添加的重复边构成的集合为 E_{dup} , 则欧拉化后图 G' 的边集为 $E' = E \cup E_{dup}$, 总边数 $|E'| = |E| + \sum_{i=1}^k |P_i|$, 其中 $k = |O|/2$ 为匹配对数。

该构造过程的优化目标为:

$$\min \sum_{(u,v) \in M} d(u,v)$$

其中, $d(u,v)$ 表示节点 u 与 v 在原图 G 中的最短路径长度。

得到欧拉图 G' 后, 应用 Hierholzer 算法求解欧拉回路。算法流程如下:

1) 任选起始节点 $v_0 \in V$, 初始化路径栈 $P = [v_0]$ 和回路序列 $C = []$ 。

2) 当 P 非空时:

① 令当前节点 $v = \text{top}(P)$ 。

② 若 v 在 G' 中仍有未访问的出边 $e = (v, u)$, 则标记 e 为已访问, 将 u 压入 P , 并令 $v \leftarrow u$ 。

③ 否则, 弹出 P 的栈顶节点, 并将其追加至 C 的末尾。

3) 最终 C 中的节点序列即为一条欧拉回路, 依次遍历该序列即完成对管道网络所有边的无遗漏巡检。

上述过程确保了每条边恰好被访问一次, 且总巡检权值达到理论下界。对于具有 $|E|$ 条边的图, Hierholzer 算法的时间复杂度为 $O(|E|)$, 在工程应用中具有较高的计算效率。需指出, 若图 G' 非连通(即存在多个连通分量), 上述基本算法需稍作调整: 若当前连通分量的所有边均被访问后, 图中仍有未访问的边, 则需从任意包含未访问边的节点重新开始, 构建子回路, 并最终将所有子回路拼接为完整的欧拉回路。然而, 由于最小生成树 T 保证了所有节点的连通性(T 连通所有节点, 添加的重复边仅连接 T 中已有节点), G' 必为连通欧拉图, 因此上述复杂情况在本文方法中不会出现。

3. 结果与讨论

为了验证本方法的有效性, 构建了一个管网复杂网络模型, 该模型包含 80 个节点和 120 条边, 其中垂直架构管网边 40 条, 如图 3 所示, 图中蓝色的圆形代表网络节点, 即管网中的交叉点, 灰色的边线表示边, 即管网的通道, 边上的值的边的权值。

图 3 展示了用于实验验证的仿真管网复杂网络模型全貌。该模型包含 80 个节点和 120 条边, 清晰地呈现了管道系统中错综复杂的连接关系。图中边的长度与标注的数值代表了机器人通过该段管道所需付出的代价(权值)。通过构建这样一个具有一定规模和复杂度的网络模型, 能够有效检验本文所提路径规划方法在真实管网环境中的适用性与鲁棒性。

管网复杂网络模型的基本拓扑属性参数如表 1 所示, 其中网络密度是指网络中实际存在的边数与理论上可能存在的最大边数之比。它反映了网络中节点之间连接的紧密程度。密度值范围在 0 到 1 之间, 值越接近 1, 表示网络越稠密; 值越接近 0, 表示网络越稀疏。平均度是网络中所有节点的度的平均值 [12], 表示网络中每个节点平均连接的其他节点数。平均接近度衡量的是网络中节点到其他所有节点的平均距离的倒数 [13]。接近度越高, 表示节点与其他节点的距离越短, 信息传播效率越高 [14]。平均介数反映了一个节点在网络中作为其他节点之间桥梁的作用 [15]。它通过计算节点出现在其他节点对之间最短路径上的次数来衡量。介数越高, 表示该节点在网络中具有更高的控制力或中介作用。网络聚类系数衡量的是一个节点的邻居之间相互连接的程度 [16]。对于某个节点, 其聚类系数定义为邻居之间实际存在的

边数与可能存在的最大边数之比[17]。

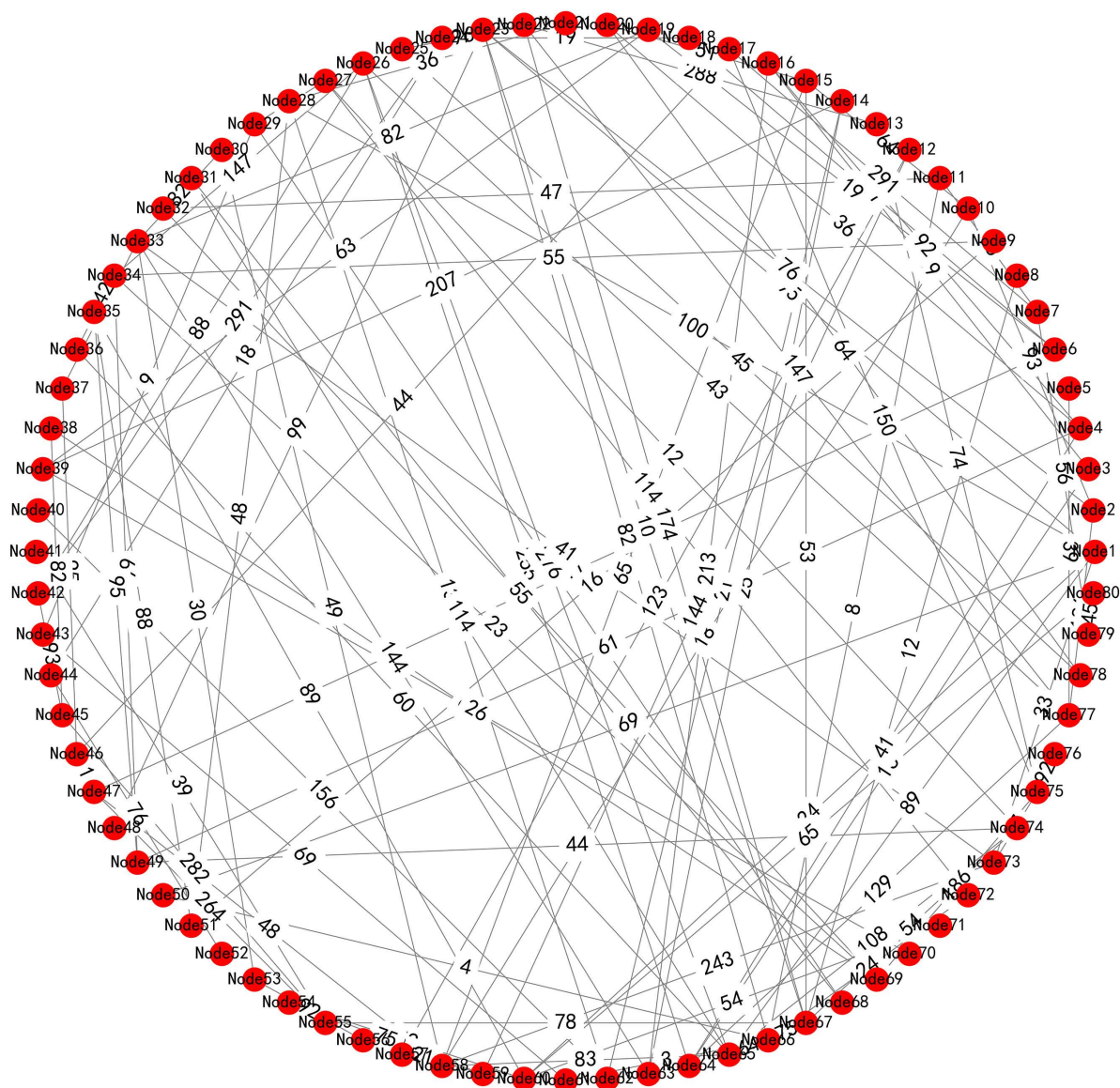


Figure 3. Complex network model of the pipeline system

图 3. 管网复杂网络模型

Table 1. Basic topological parameters of the pipeline network complex model

表 1. 管道网络复杂模型的基本拓扑参数

网络密度	平均度	平均接近度	平均介数	网络聚类系数
0.03797	3	0.24293	0.04143	0.0191

构建了网络模型以后,提取了该网络的最小生成树,如图 4 所示。然后按照第 2 节介绍的方法识别出了最优巡检线路,如图 5 所示,巡检顺序是从表格的第一行第一列的节点 1 开始,按照表格从左到右、从上到下的顺序扫描,也就是 1, 50, 66…。巡检全路径的权值和为 7486。

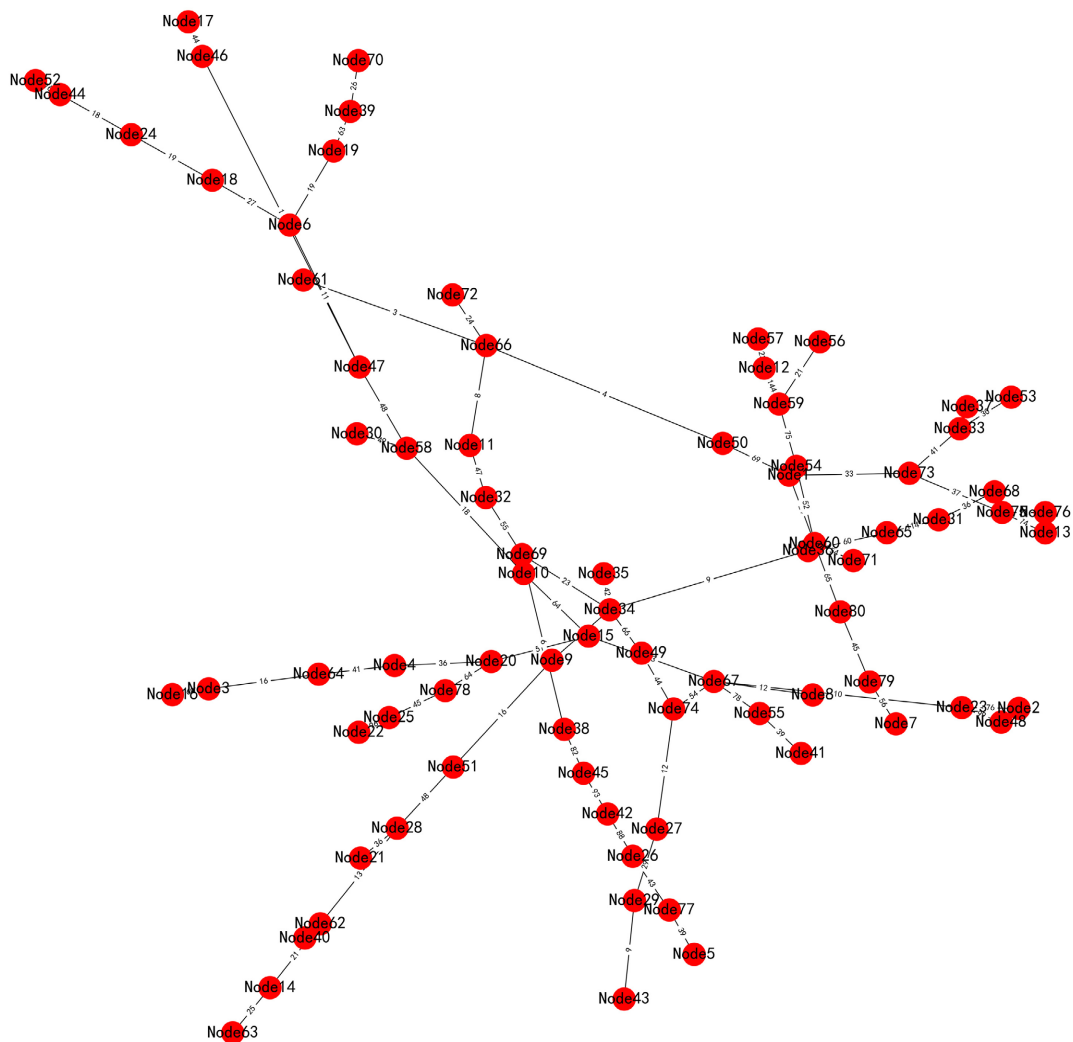


Figure 4. Minimum spanning tree
图 4. 最小生成树

图 4 呈现了从原始复杂网络(图 3)中提取出的最小生成树结构。该生成树保留了连接所有节点的必要边,且确保了总权值最小。可以看到,相比于原始网络,最小生成树剔除了大量冗余的环路连接,形成了一个不含环的树状拓扑。这一骨架结构为后续构建高效的欧拉回路、规划无遗漏巡检路径提供了最优的连通基础。

图 5 以直观的形式给出了最终生成的优化巡检路径序列。巡检机器人按照表格从左至右、从上至下的顺序依次访问节点(例如从节点 1 开始,前往节点 50、66 等),即可完成对管网所有边的遍历。该路径是应用 Hierholzer 算法在欧拉化后的网络上求解得出的,它确保了每条管道仅被巡检一次,从而实现了总巡检成本(权值和 7486)的理论最小值。

为了进一步证明本方法的有效性,将本方法与随机路径法、深度优先法、广度优先法、最大介数法进行了比较。随机路径法是每次选择下一个路径时完全随机,深度优先法是一种按照尽可能深的方式探索图或树结构的遍历方法,它优先访问节点的子节点,然后再回溯到父节点,寻找尚未访问的其他节点。广度优先法是一种按照逐层扩展的方式探索图或树结构的方法,从起始节点开始,优先访问当前层的所有节点,再逐层向下探索。最大介数法优先访问介数最大的边。各方法获得的巡检路径权值和的归一化值如图 6 所示,图中 Eulerian、Random、BFS、Max Betweenness 分别代表本文方法、随机路径法、广度

优先法和最大介数法。由图可知, 本文方法获得的巡检路径权值和归一化值小于其他方法, 表明本方法在路径总成本上优于其他几种对比方法。

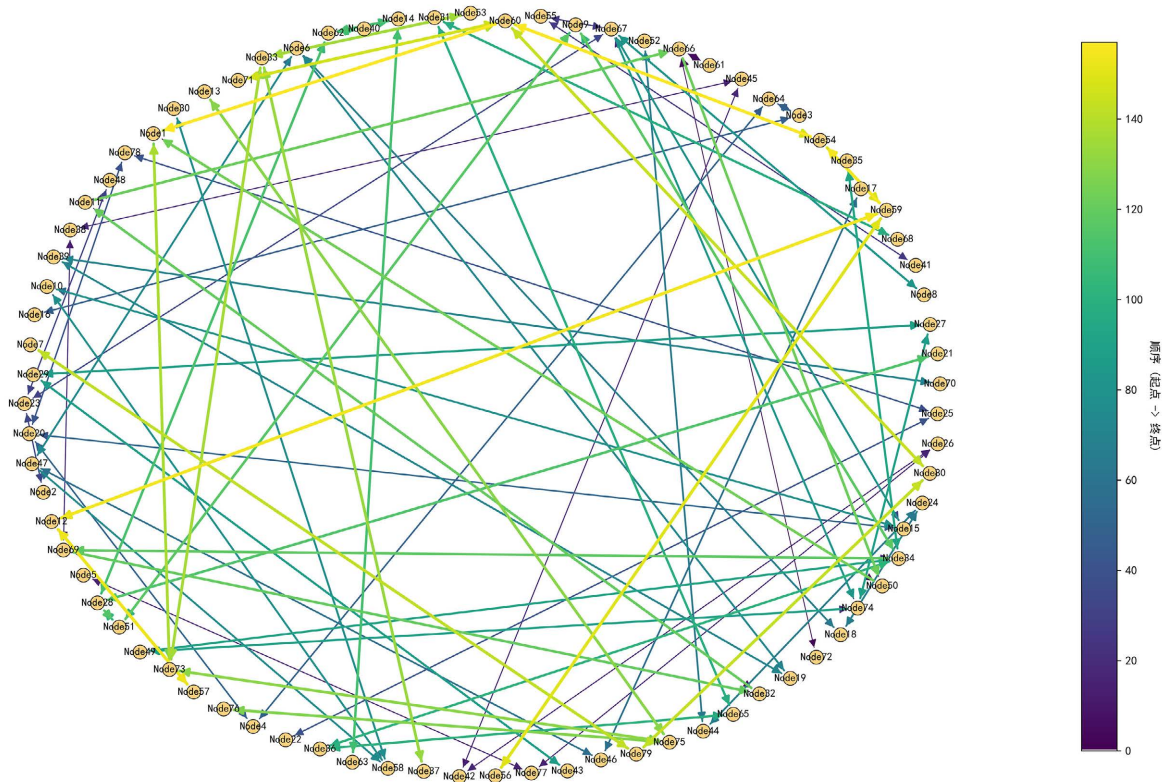


Figure 5. Inspection path
图 5. 巡线路径

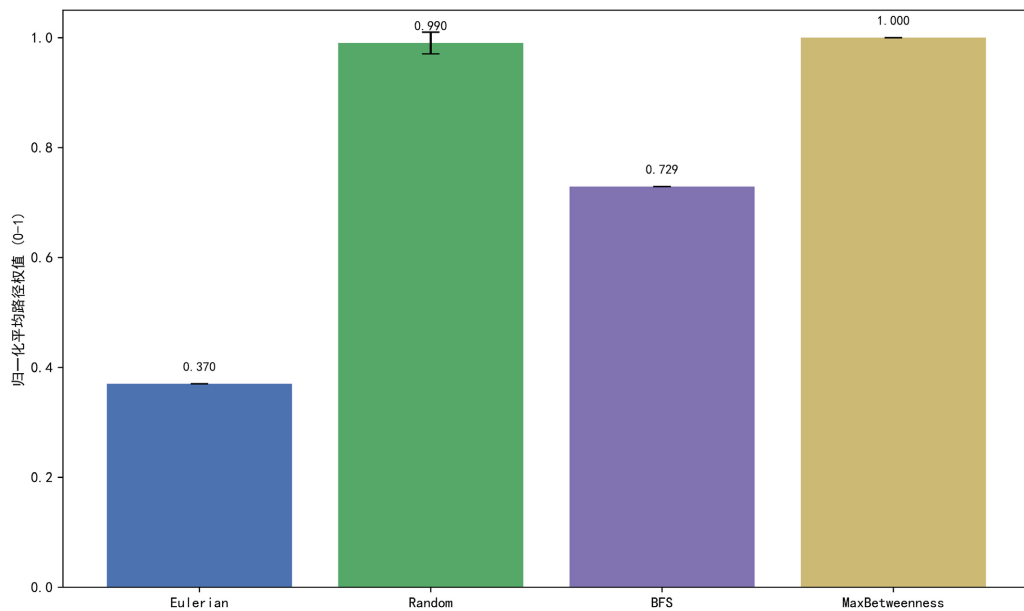


Figure 6. Comparison of normalized inspection path weights for different methods
图 6. 不同方法巡检路径权值

图 6 通过柱状图对比了本文方法(Eulerian)与随机路径法(Random)、广度优先法(BFS)及最大介数法(Max Betweenness)在归一化巡检总成本上的差异。归一化值越低, 代表路径规划方案越优。从图中可以直观地看出, 本文方法的成本柱高显著低于其他三种对比方法, 这有力地证明了基于复杂网络欧拉回路思想的路径规划策略能够最大程度地压缩巡检过程中的额外开销, 实现巡检效率的大幅提升。

为量化各方法的路径规划性能, 本文采用以下两个自定义指标:

路径重复率(η_R): 定义为巡检过程中重复经过的管道所对应的权值之和占总巡检权值的比例, 即

$$\eta_R = \frac{W_{\text{total}} - W_{\text{network}}}{W_{\text{total}}} \times 100\%$$

其中, W_{total} 为该方法规划路径的总巡检权值, W_{network} 为管道网络所有边的原始权值总和(即理论下界, 本文中 $W_{\text{network}} = 7486$)。

归一化效率(η_E): 定义为理论最小总权值与实际总巡检权值之比, 用于衡量路径规划方案接近理论下界的程度, 即

$$\eta_E = \frac{W_{\text{network}}}{W_{\text{total}}}$$

该指标取值介于 0 与 1 之间, η_E 表示路径为完全无重复的最优欧拉回路, 值越小则效率越低。图 6 中所采用的“巡检路径权值和的归一化值”即为 $W_{\text{total}}/W_{\text{network}}$, 是该指标的倒数形式, 因此本文方法对应的归一化值最小(为 1)。

为了进一步量化不同路径规划方法的实际性能差异, 本研究基于生成的路径数据, 从路径重复率与巡检总里程两个维度对四种方法进行了深入对比, 具体数据如表 2 所示。

Table 2. Comparison of performance indicators for different path planning methods

表 2. 不同路径规划方法的性能指标对比

方法名称	W_{network}	η_R	η_E
本文方法(Eulerian)	7486	0.0%	1.00
广度优先法(BFS)	14,747	49.2%	0.51
最大介数法(Max Betweenness)	20,225	63.0%	0.37
随机路径法(Random)	20,419	63.3%	0.37

注: 路径重复率定义为(总巡检权值 - 网络所有边权值总和)/总巡检权值。

为保证对比的公平性, 各对比方法均进行了面向管道全覆盖巡检的适配设计。随机路径法(Random)在每一步从未访问的邻接边中随机选取前进方向; 若当前节点所有邻接边均已访问, 则通过 Dijkstra 算法计算前往最近含未访问边节点的最短路径, 沿该路径行进并继续巡检。深度优先法(DFS)按照尽可能深入的原则探索网络, 优先访问当前节点的未访问子节点, 仅在无未访问邻接边时回溯至父节点。广度优先法(BFS)按照逐层扩展的原则遍历网络, 从起始节点开始, 优先访问同层所有节点的未访问邻接边, 再进入下一层。最大介数法(Max Betweenness)优先选择边介数最大的管道段作为行进方向, 边介数定义为网络中所有节点对之间的最短路径中经过该边的次数。上述方法均以同一节点(节点 1)为起点, 当完成所有边的覆盖后, 通过最短路径返回起点以形成闭合巡检回路。

从表 2 可以看出, 本文提出的基于复杂网络的规划方法具有明显的性能优势。首先, 在总巡检权值方面, 本方法仅 7486, 相比 BFS、最大介数法和随机法分别降低了约 49.2%、63.0% 和 63.3%, 显著减少了机器人的能源消耗与作业时间。其次, 在路径重复率方面, 本文方法实现了 0% 的重复率, 即 Hierholzer

算法成功构建了一条纯粹的欧拉回路, 使得机器人在完成全管道覆盖巡检时, 没有重复经过任何一条管道。相比之下, BFS 的路径重复率为 49.2%, 而最大介数法和随机路径法则分别高达 63.0% 和 63.3%, 这意味着机器人在实际作业中需要往返经过已检查的管道段, 造成了极大的能源与时间浪费。BFS 作为图遍历的经典方法, 其设计目标是保证节点的访问顺序而非边的覆盖效率。在管网这类稀疏图中, 单纯的 BFS 完成后仅能覆盖与生成树相关的少量边, 其余大量边需要通过回溯或重复访问来覆盖, 这正是导致其路径重复率超过 50% 的根本原因。最大介数法和随机路径法由于缺乏对边覆盖的全局优化机制, 重复率进一步攀升至 60% 以上。本文方法通过将图转化为欧拉图, 从根本上消除了回溯的必要性, 从而实现了零重复的全覆盖。综上, 本方法不仅在理论上找到了巡检路径总成本的下界解, 在具体数据指标上也展现了显著的高效性。

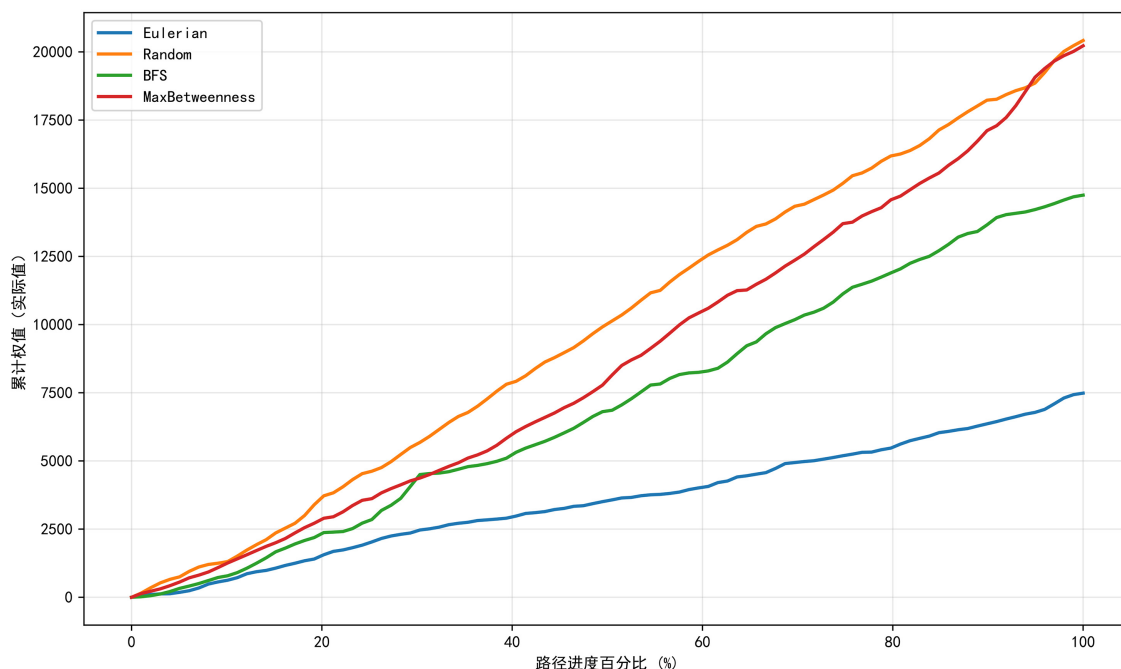


Figure 7. Cumulative weight growth curves during the inspection process

图 7. 路径进度权重增长过程

四种方法在巡检过程中的累计权值增长曲线如图 7 所示。横坐标为巡检进度的归一化值(已访问边的数量占总边数的比例), 纵坐标为累计权值。从图中可以看出, 本文方法(Eulerian)的累计权值随进度增长最为平缓, 且在任意进度节点处的累计成本均低于其他三种方法。相比之下, 最大介数法在巡检初期即产生了较高的权值积累, 随机路径法的增长曲线波动剧烈且最终成本偏高。这一结果表明, 本文方法不仅在最终总成本上最优, 且在整个巡检过程中始终保持高效的边访问顺序, 有助于机器人在有限的续航能力内完成更长距离的巡检任务。

4. 结论

本文基于经典中国邮递员问题的理论框架, 针对管道巡检路径优化问题, 提出了一套完整的系统性解决方案。通过将管道系统抽象为复杂网络模型, 结合最小生成树提取与图的欧拉化改造, 成功构建了能够无遗漏覆盖所有管段且总成本最优的巡检路径。实验结果表明, 与随机路径法、深度优先法、广度优先法和最大介数法相比, 本方法在巡检效率方面具有显著优势。这表明基于复杂网络理论的路径规划

方法能够更加高效地提升巡检机器人的工作性能, 为管道巡检系统的优化提供了重要参考。

尽管本方法在静态管网的全覆盖路径规划中取得了理想效果, 但实际管道环境往往存在动态变化和复杂约束, 因此仍有若干值得深入探索的方向。

首先, 针对管道拓扑动态变更的问题(如阀门临时关闭、局部堵塞或障碍物), 可构建“离线全局优化 + 在线局部重规划”的混合框架。在该框架下, 离线阶段采用本文方法生成全局最优欧拉回路作为基准路线; 在线执行过程中, 机器人利用自身搭载的视觉、激光雷达等传感器实时感知环境变化, 当检测到原路径不可通行时, 立即触发局部重规划模块。局部重规划可基于动态窗口法(DWA)或时间弹性带(TEB)算法, 在当前位置与基准路径上下可行节点之间生成局部避障路径; 待绕过障碍后, 机器人重新接入全局欧拉回路, 继续完成剩余边的巡检。该混合框架既能保证整体路径的最优性, 又能赋予系统对动态事件的实时响应能力。

其次, 为进一步提高规划结果的工程可执行性, 可将更多机器人实际约束融入图模型的权值设计中。例如: 1) 转弯成本: 管道交叉口处的转弯操作需机器人减速并重新定位, 可在节点处引入转弯惩罚项, 将边权值扩展为 $w'(e) = w(e) + \lambda \cdot \text{turn}(e_i, e_j)$, 其中 $\text{turn}(e_i, e_j)$ 表示从边 e_i 进入边 e_j 时的转向角度代价, λ 为权重系数; 2) 续航里程约束: 巡检机器人的单次续航里程有限, 可将最大巡检长度 L_{\max} 作为约束条件, 将原问题转化为带容量约束的弧路径问题(CARP), 通过在网络中合理设置充电或停靠节点, 将欧拉回路分割为若干满足续航限制的子回路。这些约束的引入将使规划路径更贴近机器人的实际运动能力和能源配置, 提升方案的现场部署价值。

此外, 对于未知或部分可观测的管道环境, 可将本方法与同时定位和建图(SLAM)技术相结合。机器人在首次进入未知管道时, 利用 SLAM 构建增量式拓扑地图, 并在后台持续运行本文所提的路径规划算法。当新区域被探测后, 拓扑图动态更新, 最小生成树与欧拉回路也随之在线调整, 逐步逼近全局最优的全覆盖巡检路径。这种“边探索边规划”的策略有望将本方法的应用范围从已知环境拓展至未知环境。

综上所述, 基于复杂网络与 CPP 理论的巡检路径规划方法具备坚实的理论根基和显著的效率优势, 通过与动态环境适应、实际运动约束及在线建图等技术的深度融合, 有望在更广泛的管道巡检实际场景中发挥关键作用。

基金项目

本研究受到国家级大学生创新创业项目: “吸盘侠”管内全自动巡检机器人(202410201001)资助。

参考文献

- [1] Aitken, J.M., Evans, M.H., Worley, R., Edwards, S., Zhang, R., Dodd, T., *et al.* (2021) Simultaneous Localization and Mapping for Inspection Robots in Water and Sewer Pipe Networks: A Review. *IEEE Access*, **9**, 140173-140198. <https://doi.org/10.1109/access.2021.3115981>
- [2] An, J., Lee, G., Oh, I., Moon, H. and Ryew, S. (2017) Navigation-Oriented Design for In-Pipe Robot in Recursively Divided Sampling Space with Rapidly Exploring Random Tree. *Journal of Mechanical Science and Technology*, **31**, 5987-5995. <https://doi.org/10.1007/s12206-017-1143-8>
- [3] Pang, M., Wang, X. and Ma, L. (2022) Transit Route Planning for Megacities Based on Demand Density of Complex Networks. *Promet—Traffic & Transportation*, **34**, 13-23. <https://doi.org/10.7307/ptt.v34i1.3752>
- [4] Park, S.J., Lee, K. and Yang, J. (2021) Navigating Optimal Treaty-Shopping Routes Using a Multiplex Network Model. *PLOS ONE*, **16**, e0256764. <https://doi.org/10.1371/journal.pone.0256764>
- [5] Pan, X. and Wang, H. (2018) Resilience of and Recovery Strategies for Weighted Networks. *PLOS ONE*, **13**, e0203894. <https://doi.org/10.1371/journal.pone.0203894>
- [6] Park, J. and Yang, H. (2023) Pipeline Mapping with Crawler-Type In-Pipe Robot Feature. *Journal of Mechanical Science and Technology*, **37**, 5015-5020. <https://doi.org/10.1007/s12206-023-0908-5>
- [7] Kazeminasab, S., Janfaza, V., Razavi, M. and Banks, M.K. (2021) Smart Navigation for an In-Pipe Robot through Multi-

-
- Phase Motion Control and Particle Filtering Method. 2021 IEEE *International Conference on Electro Information Technology (EIT)*, Mt. Pleasant, 14-15 May 2021, 342-349. <https://doi.org/10.1109/eit51626.2021.9491887>
- [8] Mei-Ko, K. (1962) Graphic Programming Using Odd or Even Points. *Chinese Mathematics*, **1**, 273-277.
- [9] Edmonds, J. and Johnson, E.L. (1973) Matching, Euler Tours and the Chinese Postman. *Mathematical Programming*, **5**, 88-124. <https://doi.org/10.1007/bf01580113>
- [10] Barabási, A. (2012) Luck or Reason. *Nature*, **489**, 507-508. <https://doi.org/10.1038/nature11486>
- [11] Newman, M.E.J. (2003) The Structure and Function of Complex Networks. *SIAM Review*, **45**, 167-256. <https://doi.org/10.1137/s003614450342480>
- [12] Wang, X., Trajanovski, S., Kooij, R.E. and Van Mieghem, P. (2016) Degree Distribution and Assortativity in Line Graphs of Complex Networks. *Physica A: Statistical Mechanics and Its Applications*, **445**, 343-356. <https://doi.org/10.1016/j.physa.2015.10.109>
- [13] Du, Y., Gao, C., Chen, X., Hu, Y., Sadiq, R. and Deng, Y. (2015) A New Closeness Centrality Measure via Effective Distance in Complex Networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, **25**, Article ID: 033112. <https://doi.org/10.1063/1.4916215>
- [14] López-Rourich, M.A. and Rodríguez-Pérez, F.J. (2023) Efficient Data Transfer by Evaluating Closeness Centrality for Dynamic Social Complex Network-Inspired Routing. *Applied Sciences*, **13**, Article 10766. <https://doi.org/10.3390/app131910766>
- [15] Barthélemy, M. (2004) Betweenness Centrality in Large Complex Networks. *The European Physical Journal B*, **38**, 163-168. <https://doi.org/10.1140/epjb/e2004-00111-4>
- [16] Zhang, P., Wang, J., Li, X., Li, M., Di, Z. and Fan, Y. (2008) Clustering Coefficient and Community Structure of Bipartite Networks. *Physica A: Statistical Mechanics and Its Applications*, **387**, 6869-6875. <https://doi.org/10.1016/j.physa.2008.09.006>
- [17] Zhou, B., Yan, X., Lv, Y. and Xuan, Q. (2024) Adversarial Attacks on Clustering Coefficient in Complex Networks. *IEEE Transactions on Circuits and Systems II: Express Briefs*, **71**, 2199-2203. <https://doi.org/10.1109/tcsii.2023.3337896>