

多智能体路径规划研究综述^①

张浩珂^{1,3}, 丁丽萍², 张 珩²

¹(南京信息工程大学 软件学院, 南京 210044)

²(中国科学院 软件研究所, 北京 100190)

³(中国科学院大学南京学院, 南京 211135)

通信作者: 张 珩, E-mail: zhangheng17@iscas.ac.cn



摘 要: 多智能体路径规划 (multi-agent path finding, MAPF) 旨在为多个智能体规划无冲突路径以优化协同任务性能. 本文综述了 MAPF 的研究现状, 涵盖算法分类、应用场景和未来趋势, 并探讨了其在大规模动态环境中的挑战. 首先, 本文详细介绍了 MAPF 的定义; 接着, 分类总结了基于搜索、智能仿生、采样及强化学习的路径规划算法; 最后, 分析了各类算法的优缺点及适用场景. 本文旨在帮助研究者了解 MAPF 技术的发展现状及未来方向, 推动相关研究进展.

关键词: 多智能体; 路径规划算法; 移动机器人; 研究综述

引用格式: 张浩珂, 丁丽萍, 张珩. 多智能体路径规划研究综述. 计算机系统应用, 2026, 35(1): 1-18. <http://www.c-s-a.org.cn/1003-3254/10074.html>

Review of Multi-agent Path Finding

ZHANG Hao-Ke^{1,3}, DING Li-Ping², ZHANG Heng²

¹(School of Software, Nanjing University of Information Science and Technology, Nanjing 210044, China)

²(Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)

³(University of Chinese Academy of Sciences, Nanjing, Nanjing 211135, China)

Abstract: Multi-agent path finding (MAPF) aims to plan conflict-free paths for multiple agents to optimize collaborative task performance. This study reviews the current state of MAPF research, including algorithm classification, application scenarios, and future trends, while discussing the challenges in large-scale dynamic environments. First, the study provides a detailed introduction to the definition of MAPF. Then, it categorizes and summarizes path planning algorithms based on search, bio-inspired methods, sampling, and reinforcement learning. Finally, the study analyzes the advantages and disadvantages of each algorithm and their applicable scenarios. This review aims to help researchers understand the current developments and future directions of MAPF technology, and to promote further progress in this field.

Key words: multi-agent; path planning algorithm; mobile robot; research review

多智能体路径规划 (multi-agent path finding, MAPF) 指在共享环境中为多个智能体规划无冲突的路径, 以高效协同完成指定任务. 随着智能化、自动化系统的快速发展, 其应用逐渐广泛, 例如仓储机器人需协调搬运货物, 避免碰撞并提升整体效率; 无人机集群在灾区搜救或农业巡检任务中需高效协作以防止空中冲突;

智能交通领域中的自动驾驶车辆则通过路径协作规划有效减少交通拥堵与事故风险. 因此, 多智能体路径规划不仅要考虑智能体间的碰撞避免与全局性能指标优化, 还需具备实时性与鲁棒性以应对动态复杂环境. 针对这些问题, 研究人员提出了基于图搜索的集中式算法、优化数学模型、分布式协作计算, 以及近年来兴

① 基金项目: 国家自然科学基金 (62002350); 电子数据取证系列产品研发及产业化项目 (2020AG007)

收稿时间: 2025-07-16; 修改时间: 2025-08-13, 2025-09-04; 采用时间: 2025-09-15; csa 在线出版时间: 2025-11-26

CNKI 网络首发时间: 2025-11-27

起的强化学习与博弈论方法等. 本文将系统梳理当前多智能体路径规划的研究进展, 分析不同方法的优缺点, 并展望其未来的发展趋势与挑战.

1 MAPF 问题定义与核心

1.1 MAPF 问题定义

MAPF 旨在为多个智能体在共享环境中规划无冲突的路径, 以优化协同任务的整体性能. 其目标是在确保智能体之间不发生碰撞的同时, 优化整体路径. Sharon 等人^[1]对 MAPF 进行了全面的综述, 特别提出了基于冲突的搜索算法, 该方法通过解决智能体间的冲突来实现路径规划. MAPF 问题的核心是让多个智能体从各自的起点到达各自的终点, 同时避免冲突. 路径规划问题中常见的冲突类型包括以下几种, 如图 1 所示. 顶点冲突: 智能体 1 和智能体 2 在时间步 t 都试图占据位置 C , 即两个智能体在同一个时间步同时到达同一个节点时, 就会发生顶点冲突; 边缘冲突: 智能体 1 与智能体 2 相向移动, 在同一时间步试图穿越同一条边, 导致边缘冲突; 交换冲突: 智能体 1 和智能体 2 试图在同一时间交换位置, 导致交换冲突; 依赖冲突: 智能体 1 需要等待智能体 2 离开 B 节点才能继续移动, 如果路径规划中未处理好时间依赖关系, 将导致依赖冲突.

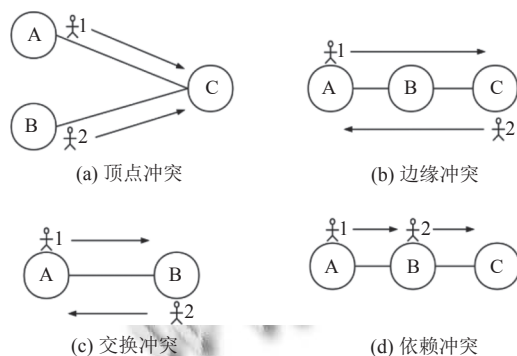


图 1 常见冲突类型

问题定义 (MAPF): 给定多智能体集合 $\mathcal{A} = \{1, 2, \dots, N\}$, 离散时间步 $t \in \{0, 1, \dots, T\}$, $x_i^t \in V$ 表示智能体 i 在时间 t 的位置, 在满足无碰撞约束的前提下, 实现最优化的目标: $\min \sum_{i=1}^N \left(T_i + \sum_{t=0}^{T_i-1} c(x_i^t, x_i^{t+1}) \right)$. 其中, T_i 为智能体 i 的到达时间, $c(\cdot)$ 为移动代价函数.

1.2 MAPF 问题的复杂性与现实场景分析

MAPF 问题本质上是一个组合优化问题, 其复杂

性随场景条件的变化而显著增加.

1.2.1 问题复杂性分析

MAPF 问题已被证明是典型的 NP 难问题, 即使在简化条件 (如二维网格、无动态障碍) 下, 问题规模也会随智能体数量的增加呈指数级增长, 导致理论最优解的计算代价极高, 难以在实际中快速求解. 这种复杂性的本质来源于 3 个因素: 一是智能体数量增加所带来的状态空间爆炸; 二是环境复杂度 (如障碍物的密集程度和空间连通性) 显著扩大搜索空间; 三是动态不确定性因素要求实时调整路径规划, 使问题的计算难度进一步提高. 因此, 在实际应用中通常会采用启发式、近似或分布式算法以实现问题的高效求解.

1.2.2 现实场景分析与核心挑战

实际应用场景通常具有动态性、不确定性、协同性及多目标权衡等特性, 这些特性直接影响 MAPF 算法的设计与优化方向. 动态环境下, 智能体需实时应对环境变化, 如自动驾驶路况、无人机风场波动. 核心挑战在于实现快速重规划与自适应调整; 不确定环境中, 智能体面临感知不完全、定位误差和通信延迟等问题, 如灾后救援、军事无人机任务, 关键难点在于确保路径的可靠性与群体鲁棒性; 协作场景则要求多智能体之间高效协同, 如仓储机器人协作、无人机编队, 着重于优化分布式决策与通信机制, 降低计算复杂度; 此外, 实际任务常涉及多目标权衡, 如路径长度、时间成本、能耗与安全性等多维优化问题, 如城市交通与物流配送, 需要算法具备平衡多目标需求的能力.

1.3 相关综述对比

现有关于 MAPF 的综述, 如 Wang 等人^[2]对多智能体强化学习在路径规划中的应用综述, 虽然涵盖了相关领域的研究进展, 但未能深入探讨大规模动态环境下 MARL 算法的扩展性及其在实际路径规划任务中的适用性, 尤其是在算法的实时性和多智能体协作方面的挑战. 此外, 许多现有综述未能有效结合大语言模型 (LLM) 与 MARL, 忽略了 LLM 在路径规划中的潜力, 尤其是在跨领域协同和决策优化方面的贡献. 本文在现有综述的基础上, 特别是从动态环境中的 MARL 可扩展性角度, 深入分析了现有算法的局限性, 提出 LLM 与 MARL 结合的思路, 推动了 MAPF 算法在动态、复杂环境中的应用与优化, 为未来的研究提供了新的视角和方向.

2 基于搜索的路径规划算法

2.1 广度优先搜索算法和深度优先搜索算法

广度优先搜索 (BFS) 是一种逐层遍历的算法, 利用队列依次访问邻居节点并标记为已访问, 在无权图或网格图中常用于寻找从起点到目标点的最短路径. 在 MAPF 中, BFS 可以检测智能体在同一时间步是否试图访问相同节点或边, 避免顶点冲突或边缘冲突, 并通过重新规划路径快速生成替代路径. 然而, BFS 在大规模场景中内存需求较大, 效率较低.

深度优先搜索 (DFS) 是一种优先深入分支的算法, 从起始节点开始沿当前路径尽可能深入, 直到达到分支尽头或没有未访问的邻居, 再回溯继续搜索. DFS 适用于连通性检测、路径搜索和拓扑排序等场景, 但不保证最短路径. 在 MAPF 中, DFS 常用于迷宫求解或路径可达性检查, 适合复杂区域中的快速判断. 然而, DFS 无法保证最优路径, 且在大规模或复杂图结构中容易陷入深层搜索, 降低效率.

BFS 和 DFS 都是用于图或树遍历的算法, 在 MAPF 中常用于解决小规模或结构较简单的场景问题. 它们都需要记录已访问的节点, 但两者在实现和应用上有所不同, 图 2 为二者的工作方式. BFS 通过队列逐层遍历节点, 能够找到无权图中的最短路径, 适合解决分层问题; DFS 通过递归或栈优先深入分支, 适合连通性检测和路径搜索, 但不保证最短路径. 此外, BFS 在宽图中可能占用更多空间, 而 DFS 在深图中可能导致较大的递归栈开销.

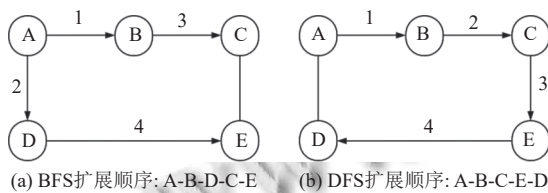


图2 以 BFS 与 DFS 基准路径规划的探索示例

2.2 Dijkstra 算法

Dijkstra 算法是一种经典的单源最短路径算法, 通过贪心策略从起点开始逐步扩展, 适用于图中边权为非负的情况. 它可用于 MAPF 中, 为每个智能体独立规划最短路径, 保证单智能体的最优路径. 然而, Dijkstra 算法无法直接处理多智能体间的路径冲突, 需要额外的冲突检测和路径调整, 增加计算复杂度. 在动态环境中, 算法可以实时更新边权, 重新计算受环境变化影响的智能体路径, 但在大规模或多智能体场景中, 其效率

可能受到影响, 且不适合实时规划.

Dijkstra 算法作为一种单源最短路径算法, 其效率问题在处理大型或稠密图时尤为突出, 这限制了其在某些复杂场景下的应用. 近年来, 针对 Dijkstra 算法的不足, 研究者提出了一些改进方法. 例如, 为了让最短路径搜索更快, Haeupler 等人^[3]于 2025 年证明了双向 Dijkstra 在许多情况下几乎是最优的, 也就是说它检查的边数已经少到没有别的算法能更大幅度减少了, 因此在工程和理论上都非常值得采用. 此外, 为了减少 Dijkstra 中频繁取最小值、更新堆的开销, Benomar 等人^[4]提出了学习增强型优先队列: 利用机器学习模型预测下一个可能的最小元素, 把正确性保证和经验加速结合起来, 从而让 Dijkstra 在实际运行时更快.

2.3 A*算法

A*算法是一种基于启发式搜索的高效路径规划算法, 通过结合实际成本和启发式估计来评估路径, 从而快速找到最优路径. 其核心在于维护开放列表 (OPEN) 和关闭列表 (CLOSE), 每次选择具有最低 $f(n)=g(n)+h(n)$ 值的节点进行扩展, 其中 $g(n)$ 是从起点到当前节点的实际代价, $h(n)$ 是从当前节点到终点的启发式估计代价, 算法流程如图 3 所示. 常用的启发函数包括曼哈顿距离和欧几里得距离等. A*算法的性能受到启发函数选择的影响, 启发函数需满足两个性质: 可采纳性: 启发函数的估计值不大于实际最短路径代价, 保证算法的最优性; 一致性: 启发函数满足三角不等式, 确保算法在扩展节点时不会重复访问已探索节点, 从而提高搜索效率.

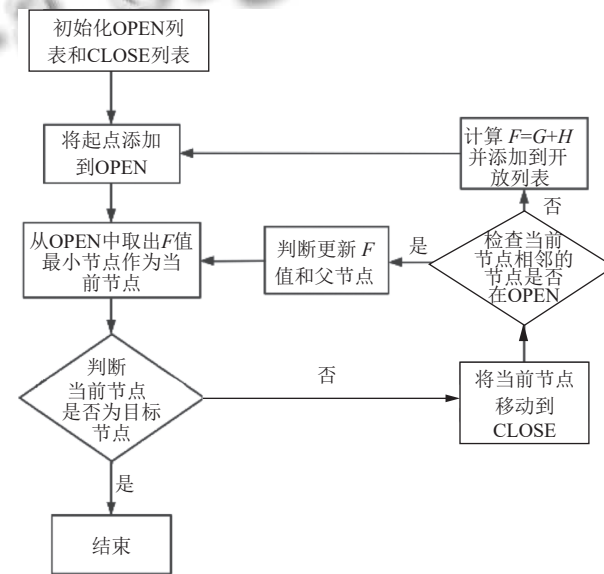


图3 A*算法流程图

相较于 Dijkstra 算法, A*算法具有高搜索效率, 能够显著减少搜索空间, 保证最优路径. 该算法易于扩展, 例如加入时间维度可处理动态环境或多智能体冲突检测问题. 然而, A*算法难以直接处理智能体间的路径冲突, 需要额外的协调机制, 增加计算复杂度. 在 MAPF 中, A*面临多个智能体的路径冲突问题, 常用改进方法包括优先级顺序法、合作 A*和时间扩展图法. 优先级顺序法为智能体分配优先级以避免冲突, 合作 A*通过智能体间的信息交换进行路径规划, 时间扩展图法则通过引入时间维度解决智能体冲突.

A*算法已广泛应用于机器人导航、游戏开发、物流与交通、医疗等领域. 在机器人导航领域, Koenig 等人^[5]将 A*用于自主机器人路径规划, 并与动态路径规划算法 D* Lite 进行了对比研究. Liu 等人^[6]提出改进型 A*, 结合人工势场和随机逃逸策略来减少节点访问、提升效率; 在医疗领域, Pehlivanoglu 等人^[7]于 2024 年提出新的手术路径规划框架, 用医学图像为不同手术器械生成安全路径; 在智能机器人方向, Liu 等人^[8]于 2024 年把深度强化学习与 A*结合, 在嵌入式平台上实现更快、更优的路径规划.

针对 A*的计算效率问题, 研究者们提出了多种改进方法, 如 Korf^[9]提出的 IDA*算法, 结合深度优先搜索和 A*优化内存使用; Liu 等人^[10]引入多搜索策略与动态自适应加权的启发式评估, 缩小有效搜索范围、降低节点扩展数, 从而提高计算效率; Mukherjee 等人^[11]提出并行化边评估, 在边评估代价高的场景显著提速, 并给出最优性、有界次优性保证. 未来对 A*算法的改进应侧重于提高计算效率, 特别是在动态环境和多智能体场景中的路径规划, 优化冲突解决机制和计算资源的利用.

2.4 JPS 算法

JPS 算法是一种优化的路径规划算法, 针对规则网格图, 通过剪枝和跳跃技术显著减少 A*扩展的节点数量, 提升搜索效率. 其核心思想是仅在路径中必须拐弯或被障碍物限制时生成跳点, 从而减少对非关键节点的检查. 跳点是必须检查的关键节点, 如转弯点或强制邻居节点, 后者是指在某个方向跳跃时必须检查的节点, 如图 4, N 为 M 的强制邻居. 与 A*相比, JPS 避免了对每个网格单元的逐一检查, 尤其在稠密网格中表现出色. 然而, JPS 在动态环境和不规则网格图上的适应性较差, 且在障碍物密集的情况下性能下降. 其代码

如算法 1.

算法 1. Jump point search (JPS)

输入: *start*, *goal*.

输出: *path*.

1. Initialize *OpenSet* with *start*

2. while *OpenSet* is not empty do

3. *current* \leftarrow node with lowest *f* in *OpenSet*

4. if *current* = *goal* then

5. return *path(current)*

6. for each direction from *current* do

7. *successor* \leftarrow *jump(current, d, g)*

8. if *successor* is valid then

9. Update *g*, *f*, parent for *successor*

10. Add *successor* to *OpenSet*

11. end for

12. return failure

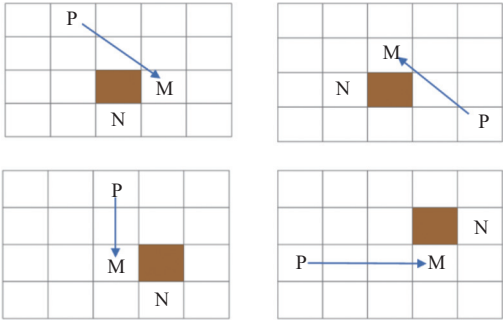


图 4 多智能体路径巡检中的强制邻居节点图

JPS 方法在多个领域被广泛应用, 如仓储机器人路径规划、游戏开发中的单位移动、城市道路网路径搜索和无人机导航等. 例如在机器人导航中, Harabor 等人^[12]通过 JPS 算法优化了仓储机器人在网格化环境中的路径规划, 使其在复杂地图上的搜索更高效; 在游戏开发中, Sturtevant 等人^[13]通过将 JPS 与 A*算法进行比较, 证明了 JPS 在大规模地图中的显著性能提升, 尤其在即时战略游戏和开放世界游戏中的单位移动方面表现优异.

在 MAPF 中, JPS 作为低层搜索器为每个智能体规划路径, 结合时空网格进行冲突检测并在发现冲突后重新规划路径. 尽管 JPS 在静态规则网格上显著提高了搜索效率, 但在动态环境和全局协调方面仍有限, 需要与 CBS 等高层冲突解决机制结合.

针对 JPS 算法的不足, 研究者们提出了多种改进方法, 如 Guo 等人^[14]提出的双向动态 JPS 算法, 通过自适应启发式函数优化了动态环境中的路径搜索效率.

Nobes 等人^[15]扩展了 JPS 算法至三维空间, 提高了高维环境中的路径规划效率. Zhao 等人^[16]通过引入几何约束, 提出了增强障碍物感知能力的 JPS 改进方法, 有效减少了冗余计算, 提升了路径规划性能. 未来对 JPS 算法的改进应集中在提升其对动态环境的适应性和高维空间的扩展能力, 同时优化在障碍物密集场景中的效率.

2.5 D*算法

D*算法是专为动态环境下的路径规划设计的, 与 A*算法不同, D*能在环境变化时高效更新路径, 无需重新规划. 其特点包括逆向搜索、增量更新和代价动态调整, 使其能快速适应动态环境. D*的基本流程包括初始化地图、从目标点反向规划路径、在环境变化时动态更新路径代价, 最终生成最优路径. 由于其高效性, D*广泛应用于机器人导航、无人机和自动驾驶领域, 尤其是在复杂的动态环境中.

2.5.1 D* Lite 算法

D* Lite 算法^[5]是 2002 年提出的一种高效的动态路径规划算法, 它可以看作是对经典 D*算法的简化和优化, 同时保留了 D*的核心优势, 即增量更新和对动态环境的快速适应能力. D* Lite 的设计灵感来源于另一种增量路径规划算法 LPA*, 它通过对路径搜索过程进行增量式更新, 仅在环境发生变化时更新受影响的部分路径, 而无需重新规划整个路径. D* Lite 的主要特点在于其简单易用的逻辑结构, 它不需要实现复杂的启发式搜索机制, 而是以一种更直观、更简化的方式实现增量更新.

D* Lite 核心思想是基于启发式搜索和增量搜索, 通过维护一个优先级队列和两种 key 值来动态调整路径. 当环境发生变化时, D* Lite 无需从头规划路径, 而是仅对受影响的节点重新计算代价并更新优先级队列, 从而高效地生成新路径. 这种方法结合了启发式搜索的高效性和增量算法的灵活性, 非常适合机器人导航等需要快速应对环境变化的场景.

2.5.2 Field D*算法

Field D*^[17]是一种针对路径平滑问题的改进版本. 传统的 D*和 D* Lite 算法通常在离散网格地图上进行路径搜索, 因此生成的路径往往呈现出离散的拐角或不自然的运动轨迹, 这在实际机器人运动中可能会导致不必要的运动耗能或执行误差. Field D*引入了线性插值技术, 使得路径可以在离散网格间生成连续的平

滑轨迹, 更加贴近机器人或无人车辆的实际运动轨迹.

Field D*的核心改进在于: 在规划路径时考虑网格节点间的连续性, 通过局部插值计算, 使路径更自然, 减少了后续的路径平滑处理需求. 尽管引入了插值计算, 但 Field D*的计算效率并未显著下降, 仍然能够满足动态环境中的实时性需求. 这种改进使得 Field D*特别适合需要高精度路径的机器人导航、自动驾驶以及无人机飞行规划任务.

2.5.3 Anytime D*算法

Anytime D*是一种专为实时路径规划任务优化的算法, 由 Likhachev 等人^[18]提出. 该算法的特点是能够在有限时间内快速生成次优解, 并在剩余时间内逐步优化路径, 最终收敛到最优解. 这种方法特别适用于时间敏感的动态环境, 例如机器人需要在移动过程中快速响应环境变化的场景.

Anytime D*的核心思想是通过初步搜索生成一个可行路径, 确保机器人能够立即开始运动, 同时继续优化路径质量. 与原始 D*相比, Anytime D*在动态环境中表现出更高的实时性, 但生成的初始解可能并非最优. 它的主要应用场景包括时间约束较强的动态路径规划任务, 例如无人机飞行、自动驾驶以及应急响应机器人导航等.

2.5.4 D*系列算法在 MAPF 应用

D*系列算法在 MAPF 中应用广泛, 主要通过每个智能体独立运行 D*算法来实现路径规划, 并在动态环境中实时更新路径. 当环境发生变化时, 智能体会动态调整路径以避免障碍物. 但由于 D*算法本身不考虑智能体之间的冲突, 在多智能体场景中, 可能会出现路径冲突. 为了解决这个问题, 通常采用冲突检测与解决策略, 如等待、重新规划、优先级控制等. 此外, D* Lite、Field D*和 Anytime D*等变种算法在不同场景中提供了不同的优势, D* Lite 更易于集成并快速响应环境变化, Field D*生成平滑路径适合复杂移动需求, Anytime D*则能快速提供初步路径并逐步优化, 适用于大规模、多变环境下的多任务系统.

2.6 基于搜索的路径规划算法对比

每个算法在计算效率、可扩展性和环境适应性方面各有特点. BFS 和 DFS 实现简单, 适用于小规模或结构化场景, 但在大规模复杂环境下效率较低, 规划时间较长. Dijkstra 算法能够保证全局最优路径, 但计算量较大, 随着智能体数量或环境规模增加, 运行速度明

显下降。A*算法通过引入启发式函数,在保持最优性的同时大幅提升搜索效率,因此被广泛应用于自动驾驶、游戏AI等实时路径规划。JPS针对规则网格环境进行了节点扩展优化,大幅提升搜索速度,在静态结构化场景中表现最好。D* Lite及其变种专为动态或部分已知环境设计,可在障碍物或地图变化时实现实时路径重规划,非常适合应用于月球车、火星车等自主机器人。

如图5可以看出,随着智能体数量的增加,BFS和DFS的平均规划时间上升最快,难以满足大规模场景下的实时性需求。Dijkstra算法虽然能够保证路径的全局最优,但其计算复杂度较高,规划时间也随智能体数量线性甚至超线性增长。A*算法凭借启发式优化,显著降低了搜索时间,在各规模下表现优于Dijkstra。JPS通过对A*的节点扩展进行剪枝,极大提升了搜索效率,在所有算法中表现最快,特别适合规则网格环境下的大规模路径规划。D* Lite在静态场景下与A*接近,但其优势在于能够应对环境变化,实现动态重规划,适合动态或部分未知环境。总体而言,JPS和A*在静态结构化环境下兼顾了效率与路径质量,而D* Lite则为动态任务提供了更好的适应性,BFS和DFS则仅适合小规模、低复杂度应用。

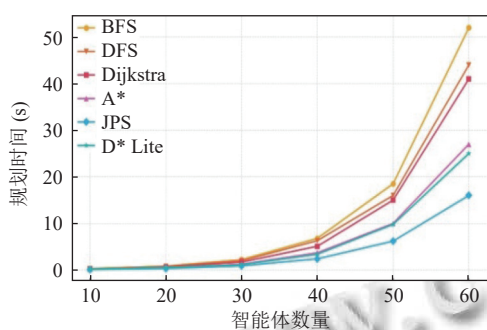


图5 搜索算法在多智能体环境规划时间对比分析

3 基于智能仿生的路径规划算法

3.1 蚁群算法

蚁群算法是一种启发式优化的路径规划方法,灵感源于蚂蚁通过释放信息素来协同寻找路径。算法模拟蚂蚁在图中移动并释放信息素,引导其他智能体选择高概率路径,进而实现全局最优路径搜索。如图6所示。

在MAPF中,蚁群算法具有较强的全局搜索能力,适用于复杂环境下的协同路径优化。然而,算法的计算

成本较高,特别是在路径冲突检测和信息素更新方面,且初期收敛速度较慢,容易陷入局部最优。如图7所示,实验对比了蚁群算法(ACO)、A*算法和GA算法的性能,结果显示,蚁群算法在初期收敛较慢,且在多智能体场景中有可能出现局部最优现象,导致路径代价在后期停滞。虽然信息素机制增强了搜索能力,但过高的信息素浓度可能导致算法过早收敛。因此,改进信息素更新策略以提高收敛速度和全局寻优能力是蚁群算法研究的重要方向。

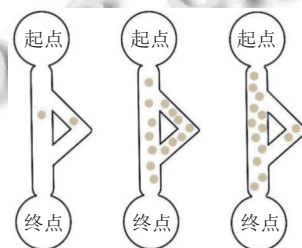


图6 基于蚁群算法的MAPF

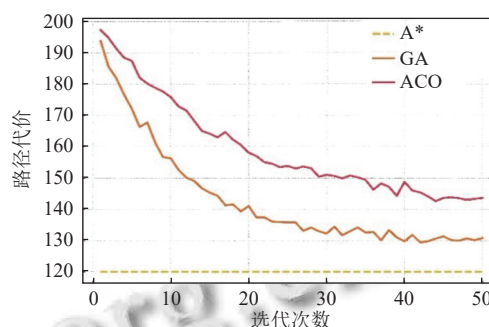


图7 不同路径规划算法的收敛曲线对比分析

蚁群算法被广泛应用于路径优化、网络路由和物流配送等领域。Dorigo^[19]首次提出蚁群算法并应用于旅行商问题,Di等人^[20]进一步将其应用于动态网络路由优化,有效提高了通信网络的传输效率。近年来,蚁群算法的改进研究主要集中在提升算法在动态环境中的适应性和搜索效率。例如,Shan等人^[21]提出了一种结合自适应蚁群优化算法与动态窗口法的路径规划方法,而Li等人^[22]结合深度强化学习和自适应蚁群算法解决动态路径规划问题。Song等人^[23]也提出了一种改进的ACO与DWA结合的路径规划策略,以提高动态环境中的避障性能。

蚁群算法在MAPF中展现了优异的表现,能够为多个智能体规划无冲突的路径。该算法天然适合MAPF,因为它模拟多个“蚂蚁”并行寻找路径,并通过信息素

进行协同、避免冲突. 蚁群算法的优点是能够进行全局并行优化, 适用于大规模、多目标和复杂障碍环境, 且能够处理多智能体间的冲突与协同, 适合离散空间, 也可扩展到连续空间.

3.2 粒子群算法

粒子群算法 (PSO) 由 Kennedy 等人^[24]于 1995 年提出, 基于群体智能的全局优化算法. 算法模拟鸟群觅食行为, 每个粒子代表一个潜在解, 基于自身历史最优和全局历史最优来调整位置和速度, 从而逼近最优解. PSO 的更新机制包括速度更新 (由惯性权重、认知部分和社会部分共同决定) 和位置更新 (根据速度调整粒子位置). 在多智能体路径规划中, PSO 通过粒子迭代更新路径, 最终找到无碰撞的最优路径.

在仿真模拟中, PSO 用于优化 20 架无人机的路径, 实验环境为障碍密度 15% 的三维空间. 结果显示, PSO 方法无碰撞率达到 97%, 路径长度较传统算法缩短 12%, 任务完成时间减少 18%. 与蚁群算法相比, PSO 在复杂障碍和高密度智能体环境中收敛速度更快, 但在高维动态环境下可能陷入局部最优. PSO 的优点包括实现简单、参数少、全局搜索能力强, 但在高维空间和动态环境中表现不稳定.

在 MAPF 中 PSO 因计算效率高、实现简单, 广泛应用于函数优化、作业调度、路径规划等领域. 例如, 在高压输电线路巡检任务中, PSO 为多架无人机规划同步路径, 优化了路径选择, 避免空中冲突, 提高了巡检效率与安全性.

近年来, PSO 在多目标优化和动态环境适应性等方面取得显著进展. 例如, Zheng 等人^[25]提出的 ADNE-PSO 算法通过对抗分解和邻域进化提高了动态环境中的适应性. Li 等人^[26]提出的 PSO-SAVL 算法通过动态调整速度限制提升了高维问题的性能, Liu 等人^[27]提出了一种混合级别集粒子群优化 LS-PSO 框架, 通过结合 PSO 与级别集法有效地处理了动态流场中的路径规划问题. 未来对粒子群优化的改进应集中在提升其在高维动态环境中的适应性和收敛速度, 尤其是在处理复杂约束和多目标优化问题时.

3.3 遗传算法

遗传算法 (GA) 是一种基于自然选择和遗传机制的全局优化算法, 由 Holland^[28]于 1975 年提出. 算法模仿生物进化过程, 通过选择、交叉和变异操作对种群中的个体进行优化, 逐步逼近最优解, 算法流程如图 8.

个体的适应度值衡量其解的优劣, 选择运算模拟“优胜劣汰”, 交叉和变异操作增加种群多样性, 避免陷入局部最优.

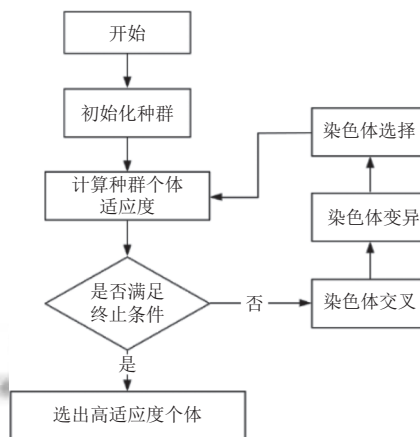


图 8 GA 流程图

在大规模场景下, GA 展现出潜在优势. 以一项假设性的大型仓储物流调度仿真实验为例: 针对 100 台搬运机器人进行路径优化, 与 A* 和 PSO 相比, GA 的碰撞事件数分别减少约 60% 与 30%, 平均路径长度降低约 8%, 总任务完成时间缩短约 13%. 该示例旨在说明, 当面向多目标、复杂障碍与高密度智能体时, GA 有望在安全性与效率之间取得更优折中; 实际收益仍取决于编码方式、约束建模、参数设置与问题规模等因素.

GA 广泛应用于组合优化、机器学习和工程设计, 具有强大的全局搜索能力、适应复杂非线性和多目标的优势. 但其缺点是收敛较慢、计算开销大、容易陷入局部最优. GA 适用于机器人群协作、无人车编队和多智能体避障任务, 在多机器人物流调度中, 通过路径编码和进化, 降低路径重叠和拥堵, 提升吞吐量并减少能耗.

为克服 GA 的不足, 研究者提出了多种改进方法. 例如, Goldberg^[29]提出的精英保留策略通过保留适应度最高的个体, 避免优质解丢失. 近年来, Wang 等人^[30]提出的基于 GA 的安全协同控制策略提高了多智能体系统的稳定性和鲁棒性. 协同进化的遗传算法 (CEGA) 在多目标优化中表现良好. 这些改进证明了 GA 在复杂优化问题中的广泛应用前景.

3.4 模拟退火算法

模拟退火算法 (SA) 是一种基于物理退火过程的全局优化方法, 灵感来源于固体高温退火时能量降低

的现象. 算法从初始解出发, 通过邻域搜索生成新解, 并根据 Metropolis 准则以一定概率接受劣解, 从而避免局部最优. 随着“温度”降低, 接受劣解的概率减少, 最终实现全局收敛, 如图 9 所示.

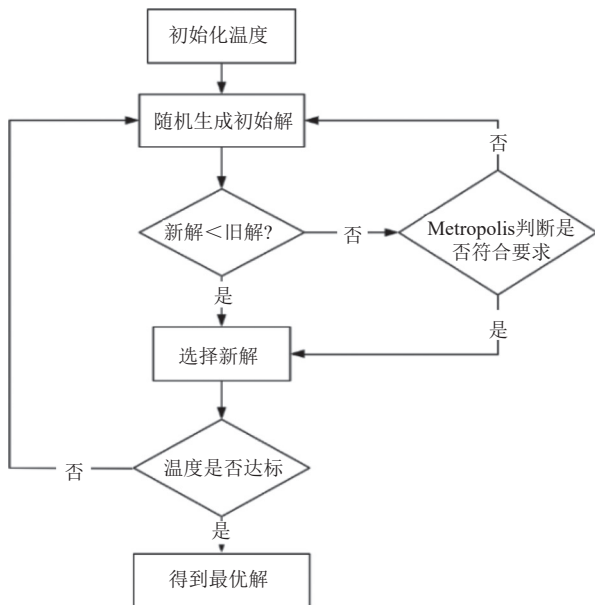


图9 模拟退火算法流程

在 MAPF 场景中, 模拟退火展现出良好效果. 以智慧园区多机器人巡逻为例, 10 台安防机器人在路径优化后, 仿真结果显示巡检覆盖率达 100%, 平均等待时间降低 23%; 与 PSO 和 GA 相比, SA 在高障碍、动态环境下达到全局最优的概率约提升 10%. 需要指出的是, SA 的收敛时间通常长于 PSO 与 GA, 更适用于对解的质量与全局搜索能力要求较高, 并且可接受较长规划时间的任务; 若任务频繁变更, 可结合增量式重启与温度重设策略以提升适应速度.

模拟退火广泛应用于组合优化、路径规划和芯片设计等领域. Kirkpatrick 等人^[31]首次提出该算法, 并成功应用于芯片布局问题. 其加速版本在图像处理中的应用也表现出更高的效率. 在多智能体全局路径规划中, 模拟退火通过允许次优解跳出局部最优, 结合全局扰动和迭代优化, 适用于高维、复杂障碍环境. 针对算法收敛速度慢、计算成本高的局限性, 研究者们提出了多种改进. 例如, Luo 等人^[32]提出的自适应初始化和多邻域搜索策略显著提高了网络入侵检测系统中的效率. Wang 等人^[33]则提出了结合贪婪算法和模拟退火的混合算法, 优化了医疗环境中的资源配置. 未来对

模拟退火算法的改进应侧重于提升其在动态和大规模环境中的计算效率, 以及加速收敛速度, 同时优化温度调节和扰动机制.

模拟退火算法在 MAPF 中的全局寻优表现出色, 尤其适用于高维、复杂环境, 能够通过随机扰动和概率接受机制跳出局部最优, 探索更优的无冲突路径. 它不仅适用于单智能体路径优化, 还能扩展至多智能体路径集合的联合优化, 尤其在解空间大、障碍密集和目标动态变化的场景中表现突出.

4 基于采样的路径规划算法

4.1 RRT 算法

RRT 算法是一种基于随机采样的路径规划算法, 其核心思想是通过随机采样快速扩展一棵树, 使其逐步探索自由空间, 从而找到从起点到目标点的可行路径. RRT 的工作机制包括以下步骤: 从起点开始, 随机从配置空间中生成一个采样点 q_{rand} , 找到树中距离该采样点最近的节点 q_{near} , 沿着 q_{near} 到 q_{rand} 的方向以固定步长生成一个新节点 q_{new} , 如果从 q_{near} 到 q_{new} 的路径无碰撞, 则将其加入树中并继续扩展, 直到找到连接到目标点的路径, 代码如算法 2. RRT 算法通过随机采样联合目标状态, 逐步向目标扩展路径, 同时进行无碰撞和动态约束检测, 确保路径规划的可行性, 这种方法非常高效, 能够快速找到一条可行解, 但通常生成的路径质量较差, 缺乏平滑性.

算法 2. RRT

输入: $start, goal, max_iter, step$.

输出: $path$.

```
1.  $T \leftarrow \{start\}$ 
2. for  $i = 1$  to  $max\_iter$  do
3.    $q_{rand} \leftarrow RandomPoint()$ 
4.    $q_{near} \leftarrow Nearest(T, q_{rand})$ 
5.    $q_{new} \leftarrow Extend(q_{near}, q_{rand}, step)$ 
6.   if  $NoCollision(q_{near}, q_{new})$  then
7.     Add  $q_{new}$  to  $T$ 
8.   if  $Dist(q_{new}, goal) < step$  then
9.     return  $path(T, q_{new})$ 
10. return fail
```

4.1.1 RRT-Connect

RRT-Connect 是对 RRT 的改进, 旨在提升路径搜索效率, 特别是在复杂环境中的应用. 其创新点在于采用双向扩展策略: 分别从起点和目标点生成两棵树, 通

过扩展树并尝试将它们连接起来,快速找到可行路径。每轮扩展时,随机采样一个点,让一棵树向采样点延伸,另一棵树连接到新节点,从而迅速合并两棵树。这种策略显著减少了搜索时间,尤其在狭窄通道等复杂环境中表现突出。

在多智能体场景中, RRT-Connect 的重点是快速生成可行路径,而非最优路径。它适用于高维空间中的路径规划,但需要额外处理路径冲突,且生成的路径通常为次优。

为了进一步优化路径质量,研究者们提出了多种改进方法。例如, Xu^[34]综述了基于 RRT 的改进算法,并讨论了其在焊接、装配和救援机器人中的应用; Lei 等人^[35]提出了多策略融合的 RRT 算法 MSF-RRT,提升了路径搜索效率和质量; Dong 等人^[36]提出 FHQ-RRT* 算法,通过动态稀疏采样和路径重连策略,提升了路径质量; Huang 等人^[37]提出的多策略双向 RRT 算法 MS-BI-RRT*,优化了路径规划效率。

4.1.2 RRT*

RRT*是 RRT 的改进版本,旨在解决原始 RRT 无法生成最优路径的问题。它引入了路径优化机制,通过在每次扩展时寻找邻居节点并连接到代价最小的节点,确保路径代价递减。RRT*通过迭代优化路径,保证渐近最优性,虽然计算时间较长,但生成的路径更短且平滑,适合需要高质量路径的场景。

RRT*在路径规划中重新检查邻居节点,动态优化路径树结构。尽管计算复杂度较高,但它是第 1 个在采样数量趋于无穷时能够渐近收敛于全局最优解的采样路径规划算法。RRT*特别适合机器人运动规划和自动驾驶等应用。

随着 RRT*的提出,研究者们对其进行了多方面的改进。Ding 等人^[38]提出了 EP-RRT*算法,结合了 RRT-Connect 的贪婪启发式方法,通过在路径扩展区域内进行启发式采样,快速探索环境并获得可行路径。该算法在狭窄通道环境中表现出较低的收敛速度和节点利用率。此外, Wang 等人^[39]提出了一种基于人工势场引导采样和模糊自适应扩展的 RRT 算法 APF-GFARRT,用于解决消毒机器人在密集环境中的路径规划问题,提高了路径质量和搜索成功率。其他研究还尝试将 RRT*应用于动态高维复杂场景中,进一步增强了适用范围。

4.1.3 Informed RRT*

Informed RRT*是在 RRT*基础上的进一步改进,

针对 RRT*在高维空间中全局随机采样效率较低的问题提出了一种更加智能的采样策略。在 RRT*中,树的扩展需要在整个自由空间中采样,这会导致许多无效采样点,尤其是在当前路径较优时。Informed RRT*通过限制采样范围,仅在与当前最优路径相关的椭圆区域内采样,从而显著提高了采样效率。随着搜索的进行,最优路径长度逐步缩短,椭圆区域也会逐渐收缩,进一步聚焦于可能包含最优路径的区域。在 MAPF 中, Informed RRT*保留了 RRT*的渐近最优性,同时通过更加智能的采样策略减少了无效探索,使其在高维复杂环境中的效率优势更加显著,适用于多智能体的高维、复杂环境路径规划。

4.1.4 RRT 优化算法对比及应用

RRT 类算法的发展过程可以理解为不断在“搜索速度-路径质量-收敛性”之间做平衡。

RRT: 基础版快速探索随机树,强调快速找到一条从起点到目标的可行路径,但路径往往曲折、冗长,未必接近最优。

RRT-Connect: 改进了扩展策略,采用双树并尽可能快速“拉近”距离,显著提高了搜索效率,常用于高维复杂空间。

RRT*: 在 RRT 基础上加入重连和重布线机制,每插入新节点时检查邻域并选择代价最小的父节点,同时优化已有连接,保证渐近最优性,即样本数趋近无穷时路径趋于最优。

Informed RRT*: 在 RRT*的基础上,利用启发式将采样限制在可能改进解的区域,避免在无关空间继续浪费采样,大幅加快收敛到高质量解的速度。

RRT 改进算法在提高效率、路径质量和适应复杂环境方面表现更优,如图 10 对比了 RRT 系列算法在 MAPF 中的主要性能指标,包括路径生成时间、路径长度以及规划成功率。

图 10 显示了不同 RRT 算法在最优路径代价、成功率和搜索耗时上的明显差异。具体来说,与基本的 RRT 算法相比, RRT*和 RRT-Connect 在路径代价和搜索效率方面进行了优化。在这 3 个指标中, Informed RRT*的表现最为出色,尤其在提升路径质量和降低搜索耗时方面,其优势尤为明显。为更好地解释这些差异,我们结合了具体实验数据来展示不同算法的性能。例如,在图 10 中,我们提供了每个算法在 3 项指标上的具体数值对比。

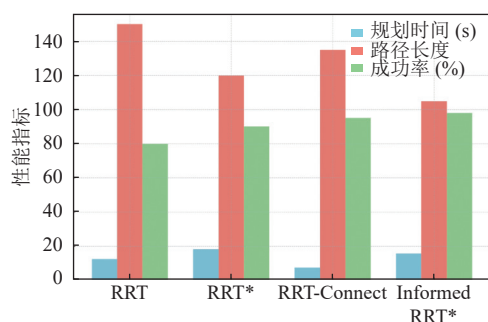


图10 RRT系列算法性能对比

路径长度: Informed RRT*相较于RRT*和RRT-Connect, 减少了约15%的路径长度, 这表明其在路径优化方面具有更高的效率。

成功率: 成功率方面, RRT*、RRT-Connect和Informed RRT*均表现较好, 其中Informed RRT*的成功率最高, 表明它在更复杂环境下的路径规划能力最强。

规划时间: RRT*算法规划时间最长, RRT-Connect在这方面表现最优, 显著减少了规划时间。具体来说, RRT-Connect的规划时间比RRT减少约20%, 显示了其更高的采样效率和收敛速度。

这些数据验证了Informed RRT*在提高路径质量、减少搜索节点数以及缩短搜索时间方面的优势, 证明了其更高的采样效率和收敛速度。

在实际应用中, RRT及其变种被广泛用于高自由度、多障碍物环境下的路径规划。例如, 在服务机器人穿越医院或写字楼等复杂室内环境时, RRT-Connect通过双向扩展迅速生成初始可行路径, 适用于路径实时性要求高的场合; RRT*常用于工业机器人或无人车辆高精度运动任务, 通过多次重连优化生成更为平滑且接近最优的路径。Informed RRT*在自动驾驶车辆进行城市道路避障和泊车路径优化时表现出色, 能在动态约束下高效收敛到高质量解。

4.2 PRM 算法

概率路图 (PRM) 是一种适用于高维自由空间路径搜索的采样算法, 广泛用于机器人路径规划。PRM通过在自由空间中随机采样生成节点, 并用局部路径规划连接这些节点, 形成无向图。该图表示了从起点到目标点的可行路径。PRM的工作流程分为两阶段: 构图阶段和查询阶段。在构图阶段, 算法随机采样并验证点的可行性, 将可行点连接成稀疏图; 查询阶段通过A*或Dijkstra算法在图中搜索最优路径。

在MAPF中, PRM通过随机采样生成无碰撞点,

构建稀疏图, 并使用搜索算法为每个智能体规划路径。同时, 优先级调度或联合搜索可解决多智能体冲突。PRM适用于高维复杂环境, 具有灵活性和良好的扩展性, 但可能面临采样密度不足导致路径不连通的问题, 且全局最优性较弱, 且多智能体协调较为复杂。该算法常用于机器人集群或无人机群的路径规划。

为提升PRM性能, Karaman等人^[40]提出了PRM算法, 通过优化采样策略使路径渐进最优。Liu等人^[41]提出的PRM-D方法结合PRM与D*算法, 提高了路径规划的成功率和速度, 特别适用于动态环境。Shukla等人^[42]则提出了基于Levy飞行的狭窄通道采样方法, 提升了在狭窄区域的采样质量和路径规划的完整性。

PRM在实际应用中表现优异。例如, 在大型仓库自动分拣系统中, PRM通过随机采样生成导航路网, 结合A*搜索优化各搬运机器人的路径, 提升运输效率。在无人机群导航任务中, PRM构建三维飞行通路图, 为多架无人机规划安全的飞行路径, 实现大范围协同目标覆盖。

5 多智能体强化学习在MAPF中的应用

多智能体强化学习 (MARL) 在MAPF中广泛应用, 核心思想是通过强化学习框架让多个智能体在共享或部分可观测环境中自主学习协同策略, 避免冲突并实现全局最优路径。与传统方法相比, MARL能在复杂动态环境中通过经验优化策略, 提高鲁棒性与泛化能力, 如图11。研究者利用去中心化训练与集中式批评 (如MADDPG)、价值分解方法 (如QMIX) 等手段, 解决多智能体间的耦合与通信问题, 显著提升大规模MAPF任务的效率与路径质量。MARL已经成为推动MAPF研究的重要方向, 并在机器人群体协作、无人机编队、仓储物流等领域展现广阔前景。与传统启发式算法相比, MARL不依赖环境建模, 通过交互与探索, 在动态、信息不完备和目标冲突环境下学习高效路径规划和决策策略。

5.1 MARL 方法基本原理

MARL是一种让多个智能体在同一环境中通过与环境和彼此的交互, 不断学习并优化各自或整体行为策略的技术。与单智能体强化学习 (RL) 不同, MARL最核心理论本质在于每个智能体所面临的环境是非平稳的, 即每个智能体的最优策略会随着其他智能体策略的变化而动态变化。这意味着, 单个智能体在学习

过程中,环境状态转移和奖励函数会不断变化,从而增加了学习任务的复杂度和不确定性。

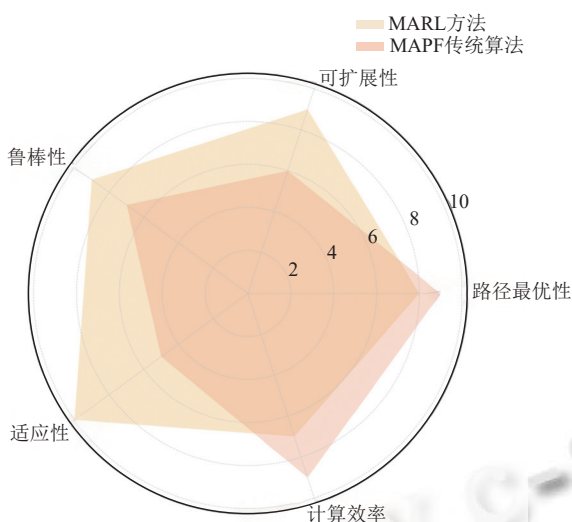


图 11 MARL 方法与传统算法对比图

从博弈论的角度来看, MARL 通常被建模为马尔可夫博弈。在这种博弈模型下,环境的状态转移不仅依赖于当前状态和单体动作,还受到所有智能体联合动作的影响。相较于单智能体 RL 中的马尔可夫决策过程 (MDP) 模型, MARL 中的最优策略不再是静态的,而是随着智能体群体策略的调整而不断演化。系统只有在所有智能体策略达到 Nash 均衡时,才实现了整体意义上的稳定和平衡。

MARL 方法主要分为两大类,一类是独立学习,即每个智能体将其他智能体的行为视为环境噪声,独立进行策略优化;另一类是集中式训练-分布式执行 (CTDE),在训练阶段利用全局信息进行联合优化,执行阶段各自独立决策。典型算法如 MADDPG^[43]、QMIX^[44]等,可以有效提升多智能体系统的协作与决策效率,支持协作、竞争、通信等多种任务。

然而, MARL 中的非平稳性也带来了收敛性难以保证、训练样本效率低、泛化与鲁棒性不足等挑战。如何应对这种动态博弈环境下的策略震荡、奖励稀疏和信用分配问题,是当前 MARL 理论和应用创新的关键研究方向。

5.2 代表性 MARL 在 MAPF 中的应用

5.2.1 MADDPG

MADDPG 属于 CTDE 框架下的深度确定性策略梯度方法,将每个智能体建模为一个具有独立策略网络的智能体,训练时使用全局信息,包含所有智能体状

态与动作,执行时仅依赖各自局部观测。这样能有效处理多智能体之间的依赖性与非平稳性。在 MAPF 中, MADDPG 可通过连续动作空间优化智能体的运动轨迹,实现路径平滑与碰撞规避。训练阶段通过全局视角引导各智能体学会协同让步、避障与路径分配。推理阶段各自独立,仅依赖本地观测,具备分布式自主决策能力。

在无人机编队、自主机器人集群、仓储物流机器人等任务中, MADDPG 被用于优化多机体避障、分布式任务分配与能量消耗最小化等目标。相关研究表明, MADDPG 在障碍密集、高动态场景下能够实现优于传统方法的无冲突路径分配与协作效率提升,如图 12 所示。

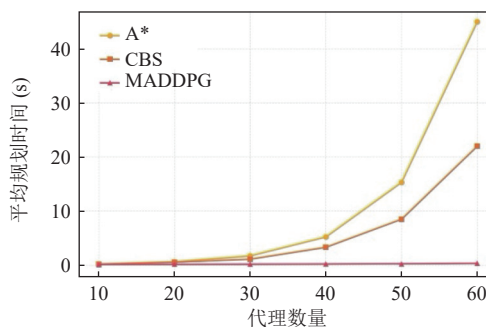


图 12 MADDPG 与传统算法效率对比

5.2.2 QMIX

QMIX 是一种值函数分解的 MARL 方法。其核心思想是将全局联合 Q 值 (整体任务收益) 分解为各智能体的局部 Q 值的可加非线性组合,确保个体策略提升有利于整体目标优化。QMIX 的混合网络保证了分解过程的单调性,从而实现去中心化执行与中心化训练的完美结合。在 MAPF 中,每个智能体根据自身观测学习独立的动作价值函数,全局联合优化总路径成本或协作目标。可应用于调度无人车、物流机器人集群等场景,提升路径规划效率、缩短整体运行时间。

QMIX 被广泛应用于大规模 MAPF,如大型仓库 AGV (自动引导车) 调度、城市交通信号优化等任务。研究表明, QMIX 在处理上百个智能体的协同路径规划问题时,表现出高效的学习与良好的可扩展性。

5.2.3 LNS2+RL

LNS2+RL^[45]是一种结合了大邻域搜索和强化学习的方法,专门用于 MAPF 任务中。LNS2 是一种基于局部搜索的优化方法,通过大范围内重新调整路径来提

高路径质量, 广泛应用于解决复杂的约束优化问题. LNS2+RL 将强化学习与 LNS2 的强大搜索能力相结合, 在低级别路径重规划时使用 MARL 进行智能体协作决策, 在高密度任务中通过 LNS2 快速调整路径, 减少碰撞并优化路径成本.

在 MAPF 中, LNS2+RL 通过结合 MARL 和 LNS2, 能够有效应对复杂的路径规划问题, 特别是高密度、多障碍物的场景. MARL 部分通过协同学习, 每个智能体在低级别的路径规划中进行决策, 而 LNS2 则在后期阶段通过优化大邻域内的路径, 进一步提升路径质量, 确保在复杂环境中能够快速找到全局最优或近似最优的路径.

LNS2+RL 已广泛应用于大规模的路径规划问题, 如城市交通流优化、无人机编队调度和自动驾驶系统中的多车调度等. 研究表明, LNS2+RL 在处理数百个智能体的协同路径规划任务时, 表现出色, 能够在短时间内实现高效的路径规划, 并且具有很好的可扩展性.

5.2.4 LLM-MARL

近年来, 大语言模型 (LLM) 的崛起为智能体决策带来了新的契机, 研究者开始探讨将 LLM 引入 MARL, 利用其知识和语言能力来提升多智能体系统的智能水平. LLM 可通过多种方式与 MARL 结合: 首先, 作为策略模型直接生成智能体的决策, 或作为世界模型模拟环境与其他智能体的行为; 其次, LLM 可以作为通信机制, 使智能体通过自然语言进行有效信息交换; 最后, LLM 还可作为元学习器, 在多智能体系统中优化学习过程. 通过这些路径, LLM 增强了 MARL 的灵活性和表现力, 尤其在复杂环境下的适应能力.

代表性项目如 Voyager, 该项目作为 LLM-MARL 的先行者, 充分利用 GPT-4 等大型语言模型在决策生成中的优势. 在 Minecraft 这类开放世界环境中, Voyager 通过让智能体与 GPT-4 进行交互, 生成任务执行代码, 帮助智能体自主进行环境探索和任务完成. 另一个重要的研究方向是 ChatArena, 这是一个针对多智能体对话和协作的研究平台. ChatArena 采用了基于自然语言的通信机制, 让多个智能体在共享环境中进行信息交换, 从而实现任务协作和竞争. 这些系统的成功证明了 LLM 在提升多智能体系统决策和协作能力方面的潜力. 此外, MetaGPT 和 CAMEL 等项目则采用了角色扮演式的多智能体系统, 这些系统通过给每个智能体赋予明确的角色, 促进智能体之间的高效协作. 在这些框

架中, 每个智能体根据其角色职责与其他智能体进行互动, 角色可以是事先定义的, 例如“产品经理”或“工程师”, 或者是任务动态生成的. 这种方法的优势在于让每个智能体集中精力处理与其角色相关的任务, 从而提高了系统的任务解决效率.

在奖励设计方面, LLM 通过其强大的文本生成和语言理解能力, 可以自动化地生成和调整奖励函数, 从而优化智能体的行为策略. Zhu 等人^[46]提出了基于 LLM 辅助的 MARL 方法, 其核心思想是通过结合 LLM 的语言理解能力和 MARL 的强化学习框架, 动态生成奖励函数, 从而增强多智能体系统的协作性和适应性, 减少人工干预并提高系统在复杂环境中的学习效率, 如图 13 所示. 这一方法为 MARL 在复杂任务中的应用开辟了新的方向, 尤其在处理非结构化和动态变化的环境时具有显著优势.

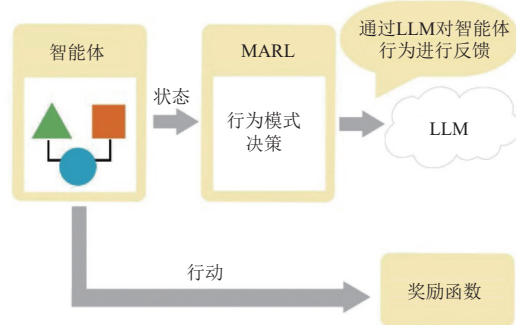


图 13 LLM-MARL 框架

此外, Kwon 等人^[47]则研究了 LLM 作为奖励函数设计工具的潜力, 提出了通过自然语言反馈来交互式指定奖励的框架. 该方法能有效地通过用户反馈调整奖励设计, 从而提高多智能体系统的协同效果. 此类方法不仅减轻了人工调试奖励函数的负担, 还能在复杂场景中提高智能体之间的合作效率. Sun 等人^[48]提出一种 CARD 框架, 旨在利用 LLM 自动生成和优化强化学习任务中的奖励函数. 该方法通过迭代生成和改进奖励函数代码, 提高了奖励设计的效率和质量.

通过引入 LLM, 研究者能够在没有明确编程指令的情况下, 快速生成并调整奖励函数, 极大提升了多智能体系统在动态环境下的学习能力和协作表现.

尽管 LLM-MARL 展现出强大潜力, 但仍面临推理成本高、环境泛化差、可解释性不足等挑战. 未来研究将侧重于提升系统的计算效率、强化智能体的环境适应性及增强模型可控性. 通过解决这些问题, LLM-

MARL 有望在更多实际应用场景中获得成功,并推动人工智能向更加通用智能迈进。

5.2.5 GNN-MARL

随着神经网络和 MARL 相结合的研究逐步深入, GNN-MARL 框架在多智能体系统中逐渐成为一种重要的解决方案,特别是在路径规划和协作任务中展现了巨大的潜力。GNN 通过图结构的建模能力,能够在智能体间建立复杂的关系网络,使得每个智能体在进行决策时可以基于其他智能体的状态与行为进行动态调整,从而提升整体协作效率和适应性。

Liu 等人^[49]提出的 GNN-MARL 方法将神经网络的优势与强化学习结合,采用图卷积网络优化智能体之间的协作策略,动态生成与调整智能体的行为模式。通过这种方式,系统能够在复杂和动态环境中更高效地优化多智能体的协作,尤其是在大规模任务和动态变化的环境中,这一方法表现出色,为 MARL 在复杂应用中的进一步发展提供了新方向。此外, Pan 等人^[50]提出的 Qedgix 框架结合了 GNN 和 QMIX 算法,针对多无人机网络中的信息年龄管理问题进行优化。该方法利用 GNN 从无人机、用户及邻近无人机中提取信息,实现有效的轨迹规划。通过优化部分可观察马尔可夫决策过程, Qedgix 显著提高了收敛速度,并降低了用户的平均 AoI 值,尤其在动态环境中表现优异。

通过结合 GNN,研究人员无需全局模型即可在局部信息下实现智能体协作,从而提升了多智能体系统在动态环境中的学习与决策能力。无论是无人机编队、自动驾驶车辆,还是仓储机器人, GNN-MARL 都展现了显著的应用潜力,特别是在复杂、多变的环境中优化路径规划和提升智能体之间的协作效率。

5.2.6 联邦学习

联邦学习^[51]是一种去中心化的学习方法,通过让多个智能体或设备在本地进行模型训练,避免了原始数据的共享,从而有效地保护了数据隐私。这一方法尤其在自动驾驶、智能交通等领域展现了巨大的应用潜力。在实时路径规划任务中,联邦学习通过将模型训练过程分布到各个智能体或设备上,每个智能体根据自身的经验更新本地模型,最后通过共享模型参数协同工作。图 14 展示了 3 辆车(智能体)如何通过本地训练和共享模型更新,在联邦学习系统中协作,最终形成一个全局模型,帮助每辆车优化路径规划。

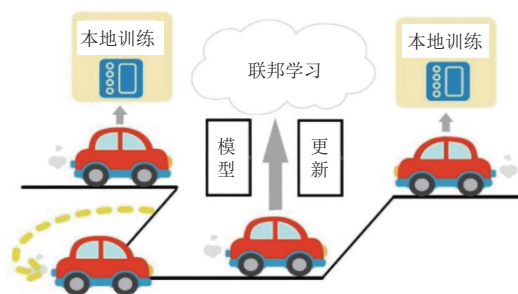


图 14 联邦学习系统

在多智能体系统中,联邦学习的优势尤为明显。以 Wang 等人^[52]提出的 FLCAV 框架为例,该框架利用车辆网络建立联邦深度神经网络,通过分布式数据进行训练,避免了数据聚合,有效保护了隐私并减少了通信成本。在一项实验中,FLCAV 框架在多辆自动驾驶汽车之间进行路径规划协作时,相比传统集中式学习方法,通信开销减少了约 30%。此外,该框架还能够提高路径规划的精度,增强车辆的实时反应能力,特别是在动态交通环境中。通过本地训练和共享模型更新,FLCAV 框架使得不同车辆能够在不共享原始数据的情况下,协同优化路径规划。

在车道分割任务中, Eid 等人^[53]提出的 FedLane 方法进一步展示了联邦学习的应用潜力。在该方法中,研究人员利用 U-Net、ResUNet 和 ResUNet++ 模型进行训练,并通过联邦学习在多个车辆间共享更新的模型。在实时推理和模型更新过程中, FedLane 显著提高了车道分割的精度。例如, U-Net 模型的 Dice 系数从 0.9429 提升至 0.9794, ResUNet 从 0.9291 提升至 0.9854, ResUNet++ 从 0.9079 提升至 0.9675。这些结果表明,联邦学习能够在复杂交通环境下提升自动驾驶系统的感知能力。

另一项研究中, FedRAV^[54]框架在自动驾驶任务中的表现也证明了联邦学习的优势。FedRAV 专注于自动驾驶中的交通物体分类任务,通过分层的联邦学习框架,在多个智能体之间共享模型更新。实验表明, FedRAV 相较于现有的传统联邦学习算法,准确率提升了至少 3.69%。这项实验验证了联邦学习在提高自动驾驶决策精度和适应多样化环境中的潜力,尤其在处理复杂道路情况和动态障碍物时表现出色。

通过上述案例和实验结果可以看出,联邦学习在 MAPF 中的优势主要体现在隐私保护、降低通信成本、适应多样化环境等方面。联邦学习作为一种去中

心化的学习方法,在实时路径规划中的应用具有广阔前景.通过分布式训练和共享模型更新,联邦学习不仅能够有效保护数据隐私,还能够提升实时路径规划的精度和适应性.通过 FLCAV、FedRAV 和 FedLane 等典型案例,我们可以看到联邦学习在自动驾驶和智能交通中的应用价值.未来,随着联邦学习技术的不断优化,它将在多智能体协作、实时决策等领域提供更强大的支持.

5.2.7 SIGMA

随着深度学习和几何优化技术的不断发展,基于几何的 MAPF 方法逐渐成为研究的热点.特别是在多智能体协作方面, Liao 等人^[55]提出了 SIGMA 算法,其核心思想是通过引入层叠体理论来增强多智能体系统的协作效率.该方法通过局部共识和信息共享,使得智能体能够在没有中心控制的情况下,基于自己的局部感知与其他智能体进行有效合作,从而实现高效的路径规划和避障.

通过将几何信息和深度强化学习相结合, SIGMA 在处理复杂、多任务和动态环境时,表现出了比传统方法更强的适应性和鲁棒性.这一方法不仅增强了多智能体系统的适应性和协作性,还在处理动态障碍物和复杂地形时展现出显著的优势.通过 SIGMA,智能体能够在本地独立进行训练,并通过几何结构进行全局优化,有效减少了路径规划中的冲突问题,提升了整体效率.该方法为 MAPF 的研究开辟了新的方向,尤其在自动驾驶、无人机编队等应用领域中,提供了更高效的解决方案.

SIGMA 不仅展现了强大的可扩展性,还能有效处理大规模的多智能体协作任务,避免了传统路径规划方法中的计算瓶颈. SIGMA 的几何化 MARL 框架为未来复杂环境中的多智能体协作提供了坚实的基础.

5.3 MARL 在 MAPF 中的优势与挑战

尽管 MARL 方法在理论与仿真中展现出优异性能,但实际部署面临诸多挑战.例如, MARL 模型通常需要大量样本和高性能计算资源进行训练,导致计算成本较高,且模型训练结果对环境变化敏感,泛化和迁移能力有限.实际部署时,需考虑模型在不同场景间的适应性、训练样本效率、通信延迟及故障容错等问题.

以 MADDPG 算法为例,在仓储机器人集群中的实验显示,应用该算法后系统整体吞吐量提升约 18%,碰撞率下降至 1% 以下,但训练时间显著增加,需数十

万次交互才能收敛. QMIX 算法在大型智能停车场无人车调度仿真中,通过分解全局 Q 值优化单车决策,使车辆平均等待时间缩短 30%,但在实际部署中对环境建模精度和通信带宽有较高要求.这些案例表明, MARL 方法具有显著提升多智能体协同效率的潜力,但在大规模复杂系统落地时,需持续优化其训练效率、泛化鲁棒性和现实适应性.

6 MAPF 算法实际应用

MAPF 算法已在实际工程和社会经济重大场景中广泛落地,极大提升了自动化系统的智能水平和协作效率.不同类型的算法因其技术特性,在不同行业和应用领域得到了针对性应用.以下结合实际案例与典型算法进行系统阐述.

6.1 基于搜索的路径规划算法的工程应用

基于搜索的路径规划算法因其理论完备性和全局最优性,广泛应用于静态或结构化环境. BFS 与 DFS 常用于迷宫求解或教育机器人等简单场景; Dijkstra 算法被广泛用于 GPS 导航与通信路由. A* 结合启发式搜索,适用于游戏 AI 和自动驾驶的局部路径规划,实现高效实时决策; JPS 在规则网格中优化 A* 的扩展效率,常用于仓储机器人和无人机路径生成. D* 及其变体专为动态环境设计,能实现路径的实时更新,已应用于月球车、火星车等自主机器人.例如在 DARPA Urban Challenge 大赛中,多车静态环境下普遍采用 A* 搜索作为局部或全局路径生成模块. NASA 火星探测车 Mars Rover 自动避障和实时路径调整使用了 D* Lite 算法进行增量更新. 以及 Google 地图早期智能交通路网最短路径、地图服务导航等.

6.2 基于智能仿生路径规划算法的产业实践

智能仿生算法通过模拟自然界的协作与进化机制,具备出色的全局搜索能力,适用于复杂、多目标和动态环境下的多智能体路径规划. 蚁群算法常用于智能交通与仓储调度,提升系统协同与鲁棒性; PSO 算法在无人机编队与 AGV 调度中表现出高效收敛能力; GA 擅长处理物流配送与车队分工等多目标优化问题; 模拟退火算法则在大规模动态任务中有效规避局部最优. 实际应用方面,如新加坡樟宜机场行李车调度、中国神华煤矿与西门子工厂车辆调度、大疆无人机航线规划、ABB 机械臂路径优化等,均成功引入仿生算法提升多智能体协同效率与规划质量.

6.3 基于采样的路径规划算法的行业应用

采样类路径规划算法因其对高维复杂空间的强适应性,广泛应用于智能制造与自动化领域. RRT 及其变种算法如 RRT-Connect、Informed RRT*可在不完全建模环境中高效生成无碰撞路径,广泛应用于机器人手臂、AGV 导航与自动驾驶车辆路径优化. PRM 算法通过构建稀疏图实现多智能体协同,常应用于仓储机器人和智能工厂的群体路径调度与多目标优化. 该类算法已在波士顿动力 Atlas 机器人、Waymo 自动驾驶系统、NASA Robonaut 机械臂等项目中落地应用,并作为 OMPL 与 ROS MoveIt! 等主流平台的标准模块,显著提升了复杂系统下的路径生成效率、任务响应速度与避障能力.

6.4 MARL 在 MAPF 场景中的算法应用

MARL 技术已在智能仓储、城市交通、应急无人机、医疗配送、智能制造等关键社会场景落地应用,典型算法包括 QMIX、MADDPG、CoLight、IDQN、MAPPO 等. 以 QMIX 为代表的分布式值分解方法在大规模 AGV 调度、无人机集群协作等任务中取得了显著效率提升; MADDPG 等 CTDE 类算法则在多机器人协作与复杂动态环境下表现出更强的自适应性和实时性. 相关方法已通过公开论文、企业部署和仿真实验在各类实际工程系统中得到验证,推动了 MAPF 技术的工程化和社会化应用进程.

具体案例包括: 华为仓储实验室使用 VDN/QMIX 算法实现 50+仓储机器人在复杂动态环境中的高效避障与分拣; 2021 年河南洪灾中, 无人机集群巡检团队

在 Gazebo+ROS 仿真平台上用 MADDPG 训练多无人机执行灾区搜索, 实现高效避障和无重叠覆盖; 擎朗医疗配送机器人在医院环境落地应用了独立 DQN 和分布式策略相结合的方法, 成功解决了医院走廊的动态障碍和多机器人交汇问题.

7 总结

本节按照 MAPF 技术演化路径对各类算法进行了简要归纳, 面向 MAPF 任务进行了系统综述. 在经典与现代方法方面, 我们比较了采样算法在搜索速度、路径质量、收敛性等方面的权衡与应用场景, 总结了仿生优化方法在群体协作和大规模调度中的优势及改进方向, 并强调其在降低拥堵、碰撞与能耗方面的工程价值. 前沿部分聚焦于范式融合, GNN-MARL 利用图结构建模多智能体关系与通信, 显著增强动态环境下的协作决策能力; 联邦学习和几何化的 SIGMA 框架进一步提升了可扩展性与鲁棒性, 在自动驾驶、无人机编队等大规模场景中减少冲突并提高效率.

与已有综述相比, 本文特别关注了动态环境下 MARL 算法的可扩展性与多智能体协作问题, 并率先探讨了将大语言模型与 MARL 融合以改进路径规划的新思路, 在跨领域协同和决策优化方面填补了现有研究的空白. 并且本文分析了各类算法的适用场景与优缺点, 如表 1, 指出了大规模动态环境下 MAPF 的主要挑战. 各类方法各有特点, 实际应用中需根据环境复杂度、动态性、计算资源和路径要求综合选择最优方案.

表 1 MAPF 算法归纳

算法类型	代表算法	计算效率	路径质量	扩展性	动态适应性	应用场景	主要瓶颈
搜索类	A*、D*、JPS	较快	较优	有限	一般	静态/小型地图	难以扩展至多智能体
仿生类	GA、ACO、PSO	较慢	一般	适中	一般	复杂多目标优化	易陷入局部最优、收敛慢
采样类	RRT、PRM	较快	一般	有限	较强	高维/连续空间	路径质量次优、高维低效
MARL	MADDPG、QMIX	一般	较优	较强	较强	动态/大规模协同	收敛性与泛化难题
融合前沿	MARL+GNN/LLM	较快	较优	较强	极强	多模态/大模型/自适应	算力与数据需求高、可解释性弱

8 现存问题与未来研究展望

目前, MAPF 问题面临着效率低下、冲突避免策略不完善、实时性与适应性不足、通信与协作机制待优化及多样化目标整合等挑战. 未来研究需致力于开发高效算法、提升分布式路径规划能力、增强动态环境适应性、改进通信与协作策略, 并探索跨领域算法融合的可能性, 同时在实际应用中进行算法验证, 以推

动 MAPF 领域的发展. 针对传统搜索算法复杂度高的问题, 可以在现有算法基础上改进算法, 在保证次优解的同时减少无效搜索, 实现效率与质量的平衡. 此外, 随着人工智能技术的快速发展, 基于多智能体深度强化学习的路径规划算法展现出广阔前景, 其高效性能能够显著提升路径规划的响应速度和多智能体的协同能力. 未来研究主要集中在以下 4 个方向.

(1) 多算法融合的研究. 环境复杂性和动态性的逐渐增加, 单一类型的路径规划算法难以满足实际需求, 多算法融合成为研究的热点方向. 例如, 将基于搜索的算法与智能仿生算法结合, 利用搜索算法的全局最优性和智能仿生算法的动态适应能力, 实现更高效的路径规划. 此外, 采样算法可以结合智能仿生方法, 通过优化采样策略提高路径质量和计算效率. 这类多算法的协同优化能够应对复杂环境下的路径规划需求.

(2) 动态环境下的实时规划. 动态环境下的路径规划是一个具有挑战性的问题, 尤其是在目标和障碍物不断变化的场景中, 如自动驾驶、无人机导航. 未来的研究可以重点关注动态更新策略和实时响应机制, 比如通过改进 D* 算法或结合深度强化学习, 快速调整规划路径. 同时, 动态环境中的不确定性建模也将是重要研究方向.

(3) 高维空间的规划优化. 机器人应用场景的不断拓展, 如机械臂控制、无人机编队, 高维自由空间的路径规划需求显著增加. 基于采样的算法虽然适合高维场景, 但在路径平滑性和计算效率上还存在改进空间. 未来研究可以聚焦于高效采样策略和路径后处理优化, 例如通过学习优化采样分布或结合机器学习预测采样点, 提高高维规划问题的解决效率.

(4) 智能化与深度学习的融合. 深度学习在路径规划中的应用潜力巨大. 例如, 基于深度神经网络的地图表示和路径预测, 可以从复杂环境中直接生成全局或局部路径, 减少传统规划算法的依赖. 深度强化学习则可以通过交互学习实现路径规划的动态适应性, 特别是在未知环境或无模型的情况下. 未来研究将进一步探索深度学习与传统算法的结合, 提高路径规划的智能化水平.

(5) 近年来, MAPF 领域正呈现出与多种前沿技术深度融合的发展趋势. MARL 与 GNN、LLM 等结合, 推动了多智能体在大规模、复杂环境下的高效协同和知识泛化能力. 例如, GNN 能实现多智能体系统中高阶依赖关系和动态拓扑建模, LLM 则为路径规划策略设计、知识迁移和复杂环境下的智能决策提供了强大支撑. 此外, 数字孪生技术的引入, 使得 MAPF 能够在虚实融合的环境中进行全流程仿真与调优, 有效提升方案的安全性和鲁棒性. 边缘计算协同为分布式多智能体实时规划、低延迟决策等应用场景提供了算力保障, 推动算法向实际工程部署迈进. 当前, MAPF 还面

临着可解释性、可复现性和大规模可扩展性等新兴需求, 未来有必要重点关注跨模态感知、知识推理和模型泛化能力等方向, 以支撑其在智慧物流、智能制造、应急救援等重大社会经济场景的落地应用.

参考文献

- 1 Sharon G, Stern R, Felner A, *et al.* Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence*, 2015, 219: 40–66. [doi: [10.1016/j.artint.2014.11.006](https://doi.org/10.1016/j.artint.2014.11.006)]
- 2 Wang SY, Xu HZ, Zheng YH, *et al.* Where paths collide: A comprehensive survey of classic and learning-based multi-agent pathfinding. *arXiv:2505.19219*, 2025.
- 3 Haeupler B, Hladik R, Rozhoň V, *et al.* Bidirectional Dijkstra's algorithm is instance-optimal. *Proceedings of the 8th Symposium on Simplicity in Algorithms*. New Orleans: SOSA, 2025. 202–215.
- 4 Benomar Z, Coester C. Learning-augmented priority queues. *Proceedings of the 38th International Conference on Neural Information Processing Systems*. Vancouver: Curran Associates Inc., 2024. 3945.
- 5 Koenig S, Likhachev M. D* Lite. *Proceedings of the 8th National Conference on Artificial Intelligence*. Edmonton: American Association for Artificial Intelligence, 2002. 476–483.
- 6 Liu HB, Cao JJ, Wang ZY. Research on path planning of mobile robot in complex environment. *Discover Applied Sciences*, 2025, 7(4): 330. [doi: [10.1007/s42452-025-06713-y](https://doi.org/10.1007/s42452-025-06713-y)]
- 7 Pehlivanoglu MK, Ay EC, Eker AG, *et al.* A new surgical path planning framework for neurosurgery. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 2024, 20(1): e2576.
- 8 Liu H, Shen Y, Yu SJ, *et al.* Deep reinforcement learning for mobile robot path planning. *arXiv:2404.06974*, 2024.
- 9 Korf RE. Depth-first iterative-deepening: An optimal admissible tree search. *Artificial Intelligence*, 1985, 27(1): 97–109. [doi: [10.1016/0004-3702\(85\)90084-0](https://doi.org/10.1016/0004-3702(85)90084-0)]
- 10 Liu C, Wu L, Li GX, *et al.* Improved multi-search strategy A* algorithm to solve three-dimensional pipe routing design. *Expert Systems with Applications*, 2024, 240: 122313. [doi: [10.1016/j.eswa.2023.122313](https://doi.org/10.1016/j.eswa.2023.122313)]
- 11 Mukherjee S, Aine S, Likhachev M. ePA*SE: Edge-based parallel A* for slow evaluations. *Proceedings of the 15th International Symposium on Combinatorial Search*. Vienna: SOCS, 2022. 136–144.
- 12 Harabor D, Grastien A. Improving jump point search. *Proceedings of the 24th International Conference on*

- Automated Planning and Scheduling. Portsmouth: ICAPS, 2014. 128–135.
- 13 Sturtevant NR, Geisberger R. A comparison of high-level approaches for speeding up pathfinding. Proceedings of the 6th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. Stanford: AAAI Press, 2010. 76–82.
- 14 Guo R, Quan XB, Bao CC. Research on mobile robot path planning based on an improved bidirectional jump point search algorithm. Electronics, 2025, 14(8): 1669. [doi: [10.3390/electronics14081669](https://doi.org/10.3390/electronics14081669)]
- 15 Nobes TK, Harabor D, Wybrow M, *et al.* The JPS pathfinding system in 3D. Proceedings of the 15th International Symposium on Combinatorial Search. Vienna: SOCS, 2022. 145–152.
- 16 Zhao SZ, Harabor D, Stuckey PJ. Reducing redundant work in jump point search. Proceedings of the 16th International Symposium on Combinatorial Search. Prague: SOCS, 2023. 128–136.
- 17 Ferguson D, Stentz A. Field D*: An Interpolation-based path planner and replanner. In: Thrun S, Brooks R, Durrant-Whyte H, eds. Robotics Research. Berlin: Springer, 2007. 239–253.
- 18 Likhachev M, Ferguson D, *et al.* Anytime dynamic A*: An anytime, replanning algorithm. Proceedings of the 15th International Conference on Automated Planning and Scheduling (ICAPS). Monterey: AAAI Press, 2005. 262–271.
- 19 Dorigo M. Optimization, learning and natural algorithms. [Ph.D. Thesis]. Milano: Politecnico di Milano, 1992.
- 20 Di Caro G, Dorigo M. AntNet: Distributed stigmergetic control for communications networks. Journal of Artificial Intelligence Research, 1998, 9: 317–365. [doi: [10.1613/jair.530](https://doi.org/10.1613/jair.530)]
- 21 Shan DR, Zhang SS, Wang XF, *et al.* Path-planning strategy: Adaptive ant colony optimization combined with an enhanced dynamic window approach. Electronics, 2024, 13(5): 825. [doi: [10.3390/electronics13050825](https://doi.org/10.3390/electronics13050825)]
- 22 Li XC, Ruan ZK, Ou Y, *et al.* Adaptive deep ant colony optimization-asymmetric strategy network twin delayed deep deterministic policy gradient algorithm: Path planning for mobile robots in dynamic environments. Electronics, 2024, 13(20): 4071. [doi: [10.3390/electronics13204071](https://doi.org/10.3390/electronics13204071)]
- 23 Song BY, Tang SM, Li Y. A new path planning strategy integrating improved ACO and DWA algorithms for mobile robots in dynamic environments. Mathematical Biosciences and Engineering, 2024, 21(2): 2189–2211. [doi: [10.3934/mbe.2024096](https://doi.org/10.3934/mbe.2024096)]
- 24 Kennedy J, Eberhart R. Particle swarm optimization. Proceedings of the 1995 IEEE International Conference on Neural Networks (ICNN). Perth: IEEE, 1995: 1942–1948.
- 25 Zheng JH, Zhang ZY, Zou J, *et al.* A dynamic multi-objective particle swarm optimization algorithm based on adversarial decomposition and neighborhood evolution. Swarm and Evolutionary Computation, 2022, 69: 100987. [doi: [10.1016/j.swevo.2021.100987](https://doi.org/10.1016/j.swevo.2021.100987)]
- 26 Li XZ, Mao KZ, Lin FF, *et al.* Particle swarm optimization with state-based adaptive velocity limit strategy. arXiv: 2308.00936, 2023.
- 27 Liu C, Li X, Miao JG, *et al.* Day-night energy-constrained path planning for stratospheric airships: A hybrid level-set particle swarm optimization (LS-PSO) framework in dynamic flows. Aerospace, 2025, 12(5): 417. [doi: [10.3390/aerospace12050417](https://doi.org/10.3390/aerospace12050417)]
- 28 Holland JH. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. Ann Arbor: University of Michigan Press, 1975.
- 29 Goldberg DE. Genetic Algorithms in Search, Optimization, and Machine Learning. Reading: Addison-Wesley, 1989.
- 30 Wang X, Yang DS, Dolly DRJ, *et al.* Genetic algorithm-based secure cooperative control for high-order nonlinear multi-agent systems with unknown dynamics. Journal of Cloud Computing, 2024, 13: 1. [doi: [10.1504/IJCC.2024.136277](https://doi.org/10.1504/IJCC.2024.136277)]
- 31 Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. Science, 1983, 220(4598): 671–680. [doi: [10.1126/science.220.4598.671](https://doi.org/10.1126/science.220.4598.671)]
- 32 Luo YF, Chen RH, Li CT, *et al.* An improved binary simulated annealing algorithm and TPE-FL-LightGBM for fast network intrusion detection. Electronics, 2025, 14(2): 231. [doi: [10.3390/electronics14020231](https://doi.org/10.3390/electronics14020231)]
- 33 Wang ZX, Huang YB, Zhang YK, *et al.* Improved genetic algorithm based on greedy and simulated annealing ideas for vascular robot ordering strategy. PLoS One, 2025, 20(2): e0306990. [doi: [10.1371/journal.pone.0306990](https://doi.org/10.1371/journal.pone.0306990)]
- 34 Xu T. Recent advances in rapidly-exploring random tree: A review. Heliyon, 2024, 10(11): e32451. [doi: [10.1016/j.heliyon.2024.e32451](https://doi.org/10.1016/j.heliyon.2024.e32451)]
- 35 Lei SJ, Li TY, Gao XC, *et al.* Research on improved RRT path planning algorithm based on multi-strategy fusion. Scientific Reports, 2025, 15(1): 13312. [doi: [10.1038/s41598-025-92675-5](https://doi.org/10.1038/s41598-025-92675-5)]

- 36 Dong XX, Wang YJ, Fang C, *et al.* FHQ-RRT*: An improved path planning algorithm for mobile robots to acquire high-quality paths faster. *Sensors*, 2025, 25(7): 2189. [doi: [10.3390/s25072189](https://doi.org/10.3390/s25072189)]
- 37 Huang YR, Jiang WX, Xu SY. A multi strategy bidirectional RRT* algorithm for efficient mobile robot path planning. *Scientific Reports*, 2025, 15(1): 29501. [doi: [10.1038/s41598-025-13915-2](https://doi.org/10.1038/s41598-025-13915-2)]
- 38 Ding J, Zhao YX, Huang X, *et al.* An improved RRT* algorithm for robot path planning based on path expansion heuristic sampling. *Journal of Computational Science*, 2023, 67: 101937. [doi: [10.1016/j.jocs.2022.101937](https://doi.org/10.1016/j.jocs.2022.101937)]
- 39 Wang HT, Zhou XL, Li JY, *et al.* Improved RRT* algorithm for disinfecting robot path planning. *Sensors*, 2024, 24(5): 1520. [doi: [10.3390/s24051520](https://doi.org/10.3390/s24051520)]
- 40 Karaman S, Frazzoli E. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 2011, 30(7): 846–894. [doi: [10.1177/0278364911406761](https://doi.org/10.1177/0278364911406761)]
- 41 Liu CY, Xie SB, Sui X, *et al.* PRM-D* method for mobile robot path planning. *Sensors*, 2023, 23(7): 3512. [doi: [10.3390/s23073512](https://doi.org/10.3390/s23073512)]
- 42 Shukla S, Kumar L, Bera T, *et al.* A levy flight based narrow passage sampling method for probabilistic roadmap planners. *arXiv:2107.00817*, 2021.
- 43 Lowe R, Wu Y, Tamar A, *et al.* Multi-agent actor-critic for mixed cooperative-competitive environments. *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Long Beach: Curran Associates Inc., 2017. 6382–6393.
- 44 Rashid T, Samvelyan M, De Witt CS, *et al.* Monotonic value function factorisation for deep multi-agent reinforcement learning. *The Journal of Machine Learning Research*, 2020, 21(1): 178.
- 45 Wang YT, Duhan T, Li JY, *et al.* LNS2+RL: Combining multi-agent reinforcement learning with large neighborhood search in multi-agent path finding. *Proceedings of the 39th AAAI Conference on Artificial Intelligence*. Philadelphia: AAAI Press, 2025. 23343–23350.
- 46 Zhu GB, Zhou R, Ji WL, *et al.* LAMARL: LLM-aided multi-agent reinforcement learning for cooperative policy generation. *IEEE Robotics and Automation Letters*, 2025, 10(7): 7476–7483. [doi: [10.1109/LRA.2025.3577527](https://doi.org/10.1109/LRA.2025.3577527)]
- 47 Kwon M, Xie SM, Bullard K, *et al.* Reward design with language models. *Proceedings of the 11th International Conference on Learning Representations*. Kigali: ICLR, 2023.
- 48 Sun SJ, Liu RZ, Lyu JF, *et al.* A large language model-driven reward design framework via dynamic feedback for reinforcement learning. *Knowledge-based Systems*, 2025, 326: 114065. [doi: [10.1016/j.knsys.2025.114065](https://doi.org/10.1016/j.knsys.2025.114065)]
- 49 Liu ZH, Zhang JY, Shi EY, *et al.* Graph neural network meets multi-agent reinforcement learning: Fundamentals, applications, and future directions. *IEEE Wireless Communications*, 2024, 31(6): 39–47. [doi: [10.1109/MWC.015.2300595](https://doi.org/10.1109/MWC.015.2300595)]
- 50 Pan YH, Wang XC, Xu ZY, *et al.* GNN-empowered effective partial observation MARL method for AoI management in multi-UAV network. *IEEE Internet of Things Journal*, 2024, 11(21): 34541–34553. [doi: [10.1109/JIOT.2024.3447774](https://doi.org/10.1109/JIOT.2024.3447774)]
- 51 Alshammari GK, Abubakar A, Ahmed NMOS, *et al.* Federated learning-based semantic segmentation for lane and object detection in autonomous driving. *arXiv:2504.18939*, 2023.
- 52 Wang S, Li CY, Ng DWK, *et al.* Federated deep learning meets autonomous vehicle perception: Design and verification. *IEEE Network*, 2023, 37(3): 16–25. [doi: [10.1109/MNET.104.2100403](https://doi.org/10.1109/MNET.104.2100403)]
- 53 Eid Kishawy MM, Abd El-Hafez MT, Yousri R, *et al.* Federated learning system on autonomous vehicles for lane segmentation. *Scientific Reports*, 2024, 14(1): 25029. [doi: [10.1038/s41598-024-71187-8](https://doi.org/10.1038/s41598-024-71187-8)]
- 54 Zhai YJ, Zhou PZ, He YP, *et al.* FedRAV: Hierarchically federated region-learning for traffic object classification of autonomous vehicles. *Proceedings of the 20th International Conference on Mobility, Sensing and Networking (MSN)*. Harbin: IEEE, 2024. 892–899.
- 55 Liao SH, Xia WH, Cao YH, *et al.* SIGMA: Sheaf-informed geometric multi-agent pathfinding. *Proceedings of the 2025 IEEE International Conference on Robotics and Automation (ICRA)*. Atlanta: IEEE, 2025. 1–7.

(校对责编: 李慧鑫)