

机器学习驱动的多智能体路径搜寻算法综述

王祥丰, 李文浩

引用本文:

王祥丰, 李文浩. 机器学习驱动的多智能体路径搜寻算法综述[J]. 运筹学学报, 2023, 27(4): 106-135.

WANG Xiangfeng, LI Wenhao. Machine learning-driven multi-agent pathfinding: An overview[J]. *Operations Research Transactions*, 2023, 27(4): 106-135.

相似文章推荐 (请使用火狐或IE浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[运筹学在整车物流智能调度决策支持系统中的研究与应用](#)

Research and application of operations research on intelligent scheduling decision support system for automotive outbound logistics
运筹学学报. 2021, 25(3): 37-73 <https://doi.org/10.15960/j.cnki.issn.1007-6093.2021.03.003>

[非凸极小极大问题的优化算法与复杂度分析](#)

Optimization algorithms and their complexity analysis for non-convex minimax problems
运筹学学报. 2021, 25(3): 74-86 <https://doi.org/10.15960/j.cnki.issn.1007-6093.2021.03.004>

[从数值最优化方法到学习最优化方法](#)

From numerical optimization method to learning optimization method
运筹学学报. 2019, 23(4): 1-12 <https://doi.org/10.15960/j.cnki.issn.1007-6093.2019.04.001>

[多示例学习问题研究进展综述](#)

A review of multi-instance learning research
运筹学学报. 2018, 22(2): 1-17 <https://doi.org/10.15960/j.cnki.issn.1007-6093.2018.02.001>

[高维约束矩阵回归问题](#)

High-dimensional constrained matrix regression problems
运筹学学报. 2017, 21(2): 31-38 <https://doi.org/10.15960/j.cnki.issn.1007-6093.2017.02.004>

机器学习驱动的多智能体路径搜寻算法综述*

王祥丰^{1,†} 李文浩²

摘要 多智能体路径搜寻 (Multi-agent Path Finding, MAPF) 问题是多智能体系统中的核心基本问题, 被广泛应用于自动化智能仓储、自动驾驶、群体机器人等实际场景。从问题属性来看, 其关键难点在于多个智能体能够同时沿着路径行驶, 同时保证不发生碰撞, 属于 NP-难组合优化问题。然而, 上述现实世界应用需要算法能够在较短的计算时间内为大量智能体搜索高质量的无碰撞路径, 更短的路径将导致更高的系统吞吐量以及更低的操作成本, 给经典 MAPF 运筹算法带来了极大挑战。因此, 近年来大量工作开始聚焦于使用机器学习方法赋能多智能体路径搜寻问题的研究, 以期加快求解速度、提升求解质量。本文综述内容包括三部分, 包括 MAPF 问题的核心概念、优化目标、基准任务, 经典 MAPF 算法的问题建模、核心思想、算法优劣, 并从机器学习赋能程度从低到高分别介绍一系列机器学习赋能的 MAPF 算法, 并给出具体的示意图和伪代码。本文还总结了机器学习驱动的多智能体路径搜寻算法目前面临的主要挑战, 并提出未来潜在研究方向, 以期可帮助领域内的研究者, 并促进机器学习方法在经典多智能体路径搜寻领域的发展。

关键词 机器学习, 多智能体路径搜寻, 多智能体强化学习

中图分类号 O221.2

2010 数学分类号 90C47, 90C26, 90C30

Machine learning-driven multi-agent pathfinding: An overview

WANG Xiangfeng^{1,†} LI Wenhao²

Abstract The Multi-agent Path Finding (MAPF) problem is a core fundamental issue in multi-agent systems and has been widely applied in practical scenarios such as automated intelligent warehousing, autonomous driving, and swarm robotics. From the perspective of problem attributes, the key difficulty lies in enabling multiple intelligent agents to travel along paths simultaneously while ensuring no collisions occur, making it an NP-hard combinatorial optimization problem. However, the aforementioned real-world applications require algorithms to find high-quality, collision-free paths for a large number of intelligent agents within a short computation time. Shorter paths lead to higher system throughput and lower operating costs, presenting significant challenges for classical MAPF optimization algorithms. As a result, in recent years, numerous studies

收稿日期: 2023-05-04

*基金项目: 国家重点研发计划 (Nos. 2021YFA1000300, 2021YFA1000302), 国家自然科学基金 (Nos. 12071145, 11971216, 62072222), 中国博士后科学基金 (No. 2022M723039)

1. 华东师范大学计算机科学与技术学院, 数学与工程应用教育部重点实验室, 上海 200062; School of Computer Science and Engineering, East China Normal University, Key Laboratory of Ministry of Education, Shanghai 200062, China

2. 香港中文大学 (深圳) 数据科学学院, 深圳市人工智能与智能体研究院, 广东深圳 518172; School of Data Science, The Chinese University of Hong Kong, Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen 518172, Guangdong, China

† 通信作者 E-mail: xfwang@sei.ecnu.edu.cn

have begun focusing on using machine learning methods to empower MAPF research, aiming to accelerate solution speed and improve solution quality. This paper presents a comprehensive review in three parts, including the core concepts, optimization objectives, and benchmark tasks of the MAPF problem. It also covers the problem modeling, core ideas, and strengths and weaknesses of traditional MAPF algorithms. Additionally and most importantly, this paper introduces a series of machine learning-empowered MAPF algorithms with varying degrees of machine learning involvement, providing corresponding schematic diagrams and pseudocode. This paper further summarizes the major challenges currently faced by machine learning-driven MAPF algorithms and proposes potential future research directions. This is intended to assist researchers in the field and promote the development of machine learning methods in the classical MAPF domain.

Keywords machine learning, multi-agent pathfinding, multi-agent reinforcement learning

Chinese Library Classification O221.2

2010 Mathematics Subject Classification 90C47, 90C26, 90C30

多智能体路径搜寻 (Multi-agent Path Finding, MAPF) 问题是多智能体系统中的核心基本问题, 其关键难点在于多个智能体能够同时沿着路径行驶, 且保证不发生智能体碰撞^[1-3]。多智能体路径搜寻被广泛应用于自动化智能仓储^[4-6]、自动驾驶^[7-9]、群体机器人^[10-12] 等实际应用场景。多智能体路径搜寻问题作为一个典型的运筹学问题, 近年来受到学术界和工业界的广泛关注, 其是一类寻找多个智能体从起始位置到目标位置且无冲突的最优路径集合的问题。

在众多应用场景中, 最常见的就是自动化智能仓储系统, 其中多智能体路径搜寻在智慧物流领域呈现出爆发式发展趋势。国内也有多家电商企业和科技公司做出令人瞩目的成果, 例如菜鸟无人仓^[13-15]、京东云仓^[16]、Geekplus^[17-19] 等。这些自动化智能仓储系统本质上是给成百上千个智能体同时做路径搜寻, 在保证安全性的同时追求时效性, 使智能体又快又稳的到达目的地。智能体必须遵循到不同位置的无碰撞路径, 以执行不同的任务。成千上万的仓库智能体已经能够实现完全自主导航, 在自动化履行中心重新定位库存吊舱, 或在分拣中心递送包裹。

多智能体路径搜寻可看作是定义在图上的路径搜寻问题^[1,3], 其由一个连通无向图和一组智能体组成, 其中图的顶点对应于智能体所处的位置, 边则对应于智能体可以移动的位置之间的连接, 如图 1 所示。图 1 中灰色方块表示障碍物, 深色方块表示智能体的源点, 六角形表示智能体的目标点, 浅色方块及其上的数字表示每个智能体对应的单智能体规划。每个智能体在每个离散的时间步占据一个顶点, 并被赋予一个起始顶点和一个目标顶点。在两个连续的时间步之间, 每个智能体都会采取一个动作, 要么移动到相邻的顶点, 要么停留在当前顶点等待。需要注意的是, 如果两个智能体同时移动到同一个顶点或沿相反方向穿过同一条边, 就会发生碰撞。多智能体路径搜寻旨在为智能体找到从起始顶点到目标顶点的无碰撞路径, 且问题目标是 minimized 完工时间 (定义为所有智能体到达目标顶点的最大时间) 或流动时间 (定义为由所有智能体到达其目标顶点的时间之和)。

多智能体路径搜寻作为经典的运筹学问题, 已有一系列传统运筹学方法被提出以高效求解多智能体路径搜寻问题。考虑到大规模多智能体路径搜寻的最优解求解过于复杂, 传统方法一般引入多种启发式的归纳偏倚技巧来提升路径搜寻效率。具体来说, 经典的多智能体路径搜寻算法可以归纳为基于约简的算法、基于规则的算法和基于搜索的算法^[3], 简要介绍如下:

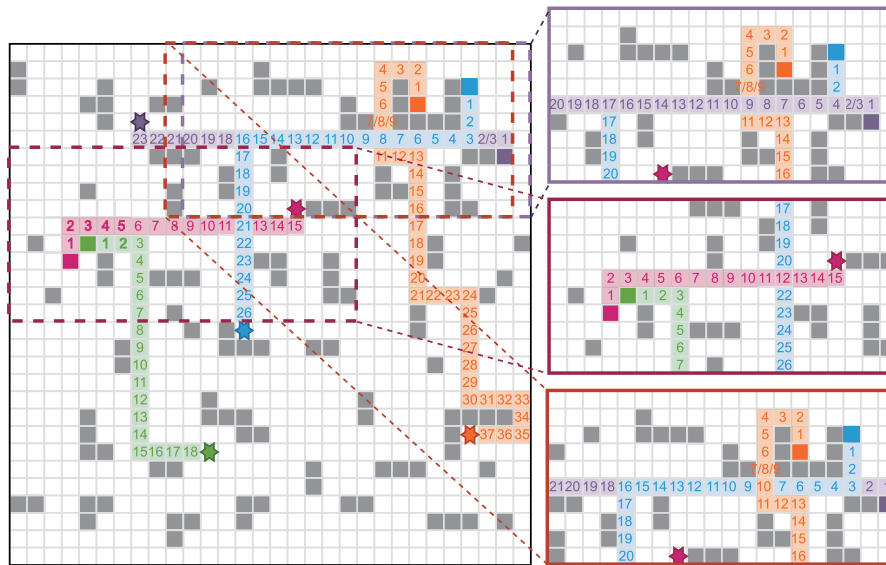


图 1 经典 MAPF 问题及对应解的示意

- 基于约简的多智能体路径搜寻算法：基于约简的多智能体路径搜寻算法将多智能体路径搜寻问题简化为其他已经研究充分的组合问题，如布尔可满足性^[20]、整数线性规划^[21]和答案集规划^[22]等；
- 基于规则的多智能体路径搜寻算法：基于规则的多智能体路径搜寻算法使用一组基本规则操作来解决多智能体路径搜寻问题，而这些基本规则操作规定了智能体在不同情况下的动作^[23-25]；
- 基于搜索的多智能体路径搜寻算法：基于搜索的多智能体路径搜寻算法通常使用启发式搜索技术来解决多智能体路径搜寻问题。用启发式搜索技术优化求解多智能体路径搜寻问题的主要挑战在于，多智能体路径搜寻问题的可能状态空间规模随着智能体数量增加呈指数级增长。经典基于搜索的多智能体路径搜寻算法包括 A*^[26]、M*^[27]、ODrM*^[28]等。

从问题属性来看，多智能体路径搜寻属于较为复杂的组合优化问题，该问题的状态空间随着问题中智能体的增多而指数增长，已被证明是 NP-难问题^[29-30]。传统方法寻找多智能体路径搜寻问题的最优解在计算上具有挑战性，尤其是对于大量智能体而言。然而，上述现实世界的应用需要算法在较短的计算时间内为大量智能体搜寻高质量的无碰撞路径，因为更短的路径将使得系统具备更高的吞吐量及更低的操作成本（仅需更少的智能体来实现相同的吞吐量）。因此，近年来大量工作开始聚焦于使用机器学习方法赋能多智能体路径搜寻问题的研究^[31-40]，以期加快求解速度、提升求解质量，我们将在文章后续部分详细介绍。

本文后续内容组织如下：第 1 节将给出经典 MAPF 问题的建模方式、关键概念以及优化目标，同时第 2 节将给出 MAPF 问题的目前标准的测试环境及数据集；第 3 节详细介绍了各类经典的 MAPF 算法，同时机器学习和多智能体路径搜寻的各类融合方法将在

第 4 节介绍, 从简单融合到深度融合分别介绍一系列机器学习赋能的多智能体路径搜寻方法。最后第 5 节对现有方法的局限性以及未来可能的研究方向进行分析, 并在第 6 节对全文进行总结。

1 问题建模

对于包含 k 个智能体 (agent) 的经典 MAPF 问题, 其输入为一个三元组 $\langle G, s, t \rangle$, 其中 $G = (V, E)$ 是一个无向图, $s : [1, \dots, k] \rightarrow V$ 将智能体映射到源点 (source vertex), $t : [1, \dots, k] \rightarrow V$ 将智能体映射到目标点 (target vertex)。假设时间是离散的, 则每个时间步内每个智能体位于图中的一个节点中, 且仅能执行单个动作 (action)。在经典 MAPF 问题中, 一个动作是一个函数 $a : V \rightarrow V$, 其中 $a(v) = v'$ 意味着如果一个智能体位于节点 v 并执行了动作 a , 则它将在下一个时间步中处于节点 v' 。每个智能体有两种动作类型: 等待 (wait) 和移动 (move)。等待表示智能体将在当前节点停留一个时间步, 移动则表示智能体从当前节点 v 移动到图中的相邻节点 v' , 即 $(v, v') \in E$ 。

对于一个动作序列 $\pi = (a_1, \dots, a_n)$ 和一个智能体 i , 我们用 $\pi_i[x]$ 表示从智能体的源点 $s(i)$ 开始, 执行前 x 个动作后智能体的位置。形式上, $\pi_i[x] = a_x(a_{x-1}(\dots a_1(s(i))))$ 。当且仅当执行序列 π 从源点 $s(i)$ 开始, 规划结果为目标点 $t(i)$ 时, 即当且仅当 $\pi_i[|\pi|] = t(i)$ 时, 序列 π 才能被称作智能体 i 的一个单智能体规划 (single-agent plan)。对应地, 整个经典 MAPF 问题的解 (solution) 则是一组 (k 个, 假定智能体的数目为 k) 单智能体规划, 其中每个智能体有且仅有一个单智能体规划^[1]。

1.1 冲突类型

MAPF 求解器的总体目标则是找到一个可以在执行时不发生碰撞的解, 即找到一组单智能体相互无碰撞的规划。为了实现这个目标, MAPF 求解器在规划时引入冲突 (conflict) 这一概念。一个经典 MAPF 问题的解是有效的 (valid), 当且仅当任意两个单智能体规划之间没有冲突。要正确定义经典的 MAPF 问题, 需要指定解中禁止哪些类型的冲突。最低要求是仅禁止边冲突 (edge conflict)。然而, 据我们所知, 所有之前处理经典 MAPF 的工作同时还禁止了顶点冲突 (vertex conflict)。因此, 我们列出了上述两种冲突的具体定义^[1]。假设 π_i 和 π_j 是一对单智能体规划, 那么有:

- 顶点冲突。当一对单智能体规划中的智能体被规划在同一时间步占用相同的顶点时, 它们之间将发生顶点冲突。形式上, 当存在时间步 x , 使得 $\pi_i[x] = \pi_j[x]$ 时, π_i 和 π_j 之间存在顶点冲突;
- 边冲突。当一对单智能体规划中的智能体被规划在同一时间步以相同的方向通过相同的边时, 它们之间将发生边冲突。形式上, 当存在时间步 x , 使得 $\pi_i[x] = \pi_j[x]$ 且 $\pi_i[x+1] = \pi_j[x+1]$ 时, π_i 和 π_j 之间存在边冲突。

除了上述两类冲突外, 还有几类在文献中较为少见的冲突定义^[1]。我们也将其陈述如下, 供读者参考。

- 追随冲突: 当一个智能体计划占据另一个智能体在上一个时间步骤中占据的顶点时, 就会发生追随冲突。这是因为存在一个时间步 x 使得 $\pi_i[x+1] = \pi_j[x]$;

- 循环冲突: 这种冲突涉及一组智能体的规划, $\pi_i, \pi_{i+1}, \dots, \pi_j$, 当每个智能体在同一时间步骤中移动到先前被另一个智能体占据的顶点时, 导致“旋转循环”模式。形式上, 如果存在一个时间步 x 满足 $\pi_i(x+1) = \pi_{i+1}(x)$ 、 $\pi_{i+1}(x+1) = \pi_{i+2}(x)$ 、 \dots 、 $\pi_{j-1}(x+1) = \pi_j(x)$ 和 $\pi_j(x+1) = \pi_i(x)$, 则存在循环冲突;
- 交换冲突: 当两个智能体在一个时间步内计划交换位置时, 就会发生交换冲突。交换冲突发生的条件是存在一个时间步 x 使得 $\pi_i[x+1] = \pi_j[x]$ 和 $\pi_j[x+1] = \pi_i[x]$ 。在 MAPF 文献中, 这一冲突有时也被归类为边冲突。

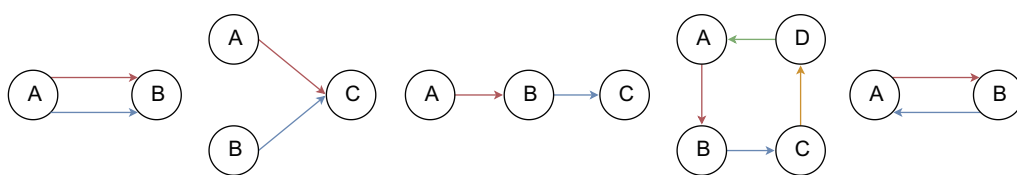


图 2 常见冲突类型示意图。从左到右依次为: 边冲突、顶点冲突、追随冲突、循环冲突以及交换冲突

上述五种冲突的示意图如图 2 所示。需要注意的是, 这些冲突定义并不代表所有可能冲突。根据它们的形式化定义, 这些冲突之间存在包含与被包含关系: (1) 禁止顶点冲突导致禁止边缘冲突; (2) 不允许追随冲突导致禁止循环冲突和交换冲突; (3) 禁止循环冲突意味着禁止交换冲突。反之: (1) 允许边缘冲突意味着顶点冲突也被允许; (2) 允许交换冲突允许循环冲突; (3) 允许循环冲突允许追随冲突。

为了正确定义一个经典的 MAPF 问题, 需要指定哪些类型的冲突是允许的。正如前面我们提到, 边冲突以及顶点冲突是文献中最为常见的两类冲突。部分研究关注具有有效载荷转移的 MAPF 问题, 允许存在交换冲突^[41]。大多数基于搜索的 MAPF 算法研究^[26,42] 禁止交换冲突, 但允许追随冲突。部分基于编译的 MAPF 算法以及将 MAPF 视为卵石运动问题的研究也禁止追随冲突^[43-44]。

1.2 智能体在目标位置的行为

在经典 MAPF 问题的解中, 智能体通常在不同的时间步到达目的地。因此, 定义智能体在最后一个智能体到达目标顶点前, 到达目标顶点后的行为是至关重要的。关于智能体在目标顶点的行为, 有两个主要假设^[1]:

- 停留在目标顶点: 在这个假设下, 智能体会停留在目标顶点, 直到所有其他智能体到达各自的目标顶点。因此, 等待在目标处的智能体可能会与在它到达目标之后经过该目标顶点的任何单智能体规划发生顶点冲突。形式上, 对于此假设, 如果存在一个时间步 $t \geq |\pi_i|$, 使得 $\pi_j[t] = \pi_i[|\pi_i|]$, 那么两个单智能体规划 π_i 和 π_j 就会发生顶点冲突;
- 到达目标顶点后消失: 根据这个假设, 智能体在到达目标后立即消失。这导致在智能体到达目标后, 该智能体的规划不再可能与其他单智能体规划发生任何冲突。

尽管在经典 MAPF 的大部分研究中都遵循“停留在目标顶点”假设, 但最近的一些研究也探讨了“到达目标顶点后消失”的情况^[45]。

1.3 优化目标

可以说, 在大多数实际的 MAPF 应用中, 有些解比其他解更好。为了评估解的质量, 关于经典 MAPF 问题的研究会引入一个目标函数。最常见的两个目标函数是最大完工时间 (makespan) 和成本和 (sum of costs)^[1]:

- 最大完工时间。让所有智能体到达目标点所需的时间步数。形式上, 对于一个经典 MAPF 的解 $\pi = \{\pi_1, \dots, \pi_k\}$, π 的最大完工时间被定义为 $\max_{1 \leq i \leq k} |\pi_i|$;
- 成本和。每个智能体到达目标所需的时间步数之和。形式上, π 的成本和被定义为 $\sum_{1 \leq i \leq k} |\pi_i|$ 。成本和也被称为流动时间 (flowtime)。

不同工作会根据实际问题选取这两个目标函数其一进行优化。值得注意的是, 若考虑到智能体在目标顶点位置的行为, 目标函数需要小心处理。具体来说, 为了了解当智能体停留在目标顶点时, 以及目标函数为成本和时, 停留在目标位置的智能体对该目标函数的影响, 有必要定义这种行为将产生何种效果。一种定义是: 当智能体停留在目标位置时, 它并不会导致成本和的增加。以往的研究通常认为, 位于目标位置的智能体会执行“等待”动作, 只要它不打算再次移动离开目标位置。

考虑这样一个场景: 智能体 i 在时间步 t 到达目标位置, 然后在时间步 t' 离开目标位置, 在时间步 t'' 返回目标位置, 并停留在目标位置, 直到所有智能体都到达各自的目标位置。该智能体的规划将为对应的解贡献总量为 t'' 的成本。

上述两种目标函数并非 MAPF 算法唯二可用的目标函数, 我们还可以定义其他目标函数。例如到达目标所需的总非等待动作数 (有时称为燃料总和) 以及智能体未位于目标顶点时所花费的总时间。基于编译的 MAPF 算法通常采用完工时间, 而以搜索为基础的 MAPF 算法大多采用成本和。也有部分工作同时设计了两类算法, 因此同时考虑了两种目标函数^[43]。此外, 也有工作^[46] 考虑在给定完工时间的约束下最大化达到目标顶点的智能体数目。

2 基准测试

在此节中, 我们将详细介绍如何通过基准测试评估一个 MAPF 算法的性能。我们首先在 2.1 节中解构一个基准任务, 对其中的关键模块进行介绍; 随后, 我们将在 2.2 节中给出两个被广泛使用的公开测试环境及数据集的介绍。

2.1 基准任务架构

MAPF 问题由一个图和一组“源点-目标顶点”对组成。因此, MAPF 的基准包括一组图, 每个图配对多个“源-目标顶点”组合。值得强调的是, 在 MAPF 问题中, 图主要都以格网的形式呈现, 这是由于 MAPF 的应用主要是在网格化的仓库或其他大型室内环境中。一方面, 常见的图, 或格网类型, 包含以下几种^[1]:

- 《龙腾世纪: 起源》(DAO) 地图: 源自《龙腾世纪: 起源》游戏, 这些格网设计位于 Sturtevant 的 movingai.com 代码仓库中^[47]。它们相对辽阔且开放, 尺寸可达 1000×1000 及更大;
- 开放式 $N \times N$ 格网: 维度为 $N \times N$ 的格网, 其中 N 的常用值为 8、16、32。这些格网允许进行智能体密度高且留有较少未被占用的顶点的实验;

- 随机障碍的 $N \times N$ 格网: 这些 $N \times N$ 格网含有随机放置的不可通行障碍物^[26];
- 仓库格网: 最近的 MAPF 研究使用由现实世界自动仓库设计启发的格网, 具有分布均匀的货架和长走廊^[48-49]。

另一方面, 智能体“源点-目标顶点”对的生成方式包含以下几种^[1]:

- 随机: 随机选择源点以及目标顶点, 并确保它们之间存在可行路径;
- 聚类: 第一个智能体的源点和目标顶点通过随机方式生成, 而后续智能体的源点和目标顶点分别从第一个智能体的源点和目标顶点的半径距离 r 以内选择;
- 指定: 对于每个智能体, 从预定义的可能“源点-目标顶点”集合中随机选择源点和目标顶点。

随机生成是最普遍的方法, 聚类生成则增加了 MAPF 问题的难度。指定生成旨在模拟自动化仓库和自动驾驶路口。在自动化仓库中, 包裹往往在特定位置由人员处理^[45,50-51]; 而在与自动驾驶路口的设定中, 指定源点和目标顶点为路口端点^[52]。

2.2 公开测试环境及数据集

本小节将介绍两个被广泛使用的公开测试环境及数据集, 分别是面向通用格网任务的 Grid-based MAPF^[1] 以及面向自动化仓库的 Aspriolo^[53]。

Grid-based MAPF 的基准*包括来自不同来源的 24 张地图:

- 真实世界的城市地图;
- 来自 Dragon Age Origins 和 Dragon Age 2 视频游戏的地图;
- 带有和无随机障碍的开放格网;
- 迷宫格网;
- 房间格网;

这些地图源自 MovingAI 寻路资源库[†]^[47]。这五类地图的示例如图 3 所示。

每张地图包含 25 个独特场景, 所有场景都由一组通过随机生成方法确定的源点和目标顶点组成。在地图的最大可达区域内随机配对点, 然后将前 1 000 个生成的问题组成场景。通过选择任意子集的源点和目标顶点, 可以从每个场景创建一系列 MAPF 问题。该基准测试流程如下: 对于选定的 MAPF 算法、地图类型和场景, 尝试按照连续顺序让场景中尽可能多的智能体达到目标顶点。具体来说, 从包含选定场景中第一对源-目标配对的两个智能体 MAPF 问题开始, 运行选定的 MAPF 算法。如果在限制的时间范围内成功完成, 继续使用前三个源-目标配对进行三个智能体 MAPF 问题, 并重复这个过程, 直到一个问题不能在限制的时间内解决。对于每个场景, 评估的算法性能定义为在限制时间内能够解决的包含最大智能体数量的场景。

*<https://movingai.com/benchmarks/mapf.html>

†<https://movingai.com/benchmarks/grids.html>

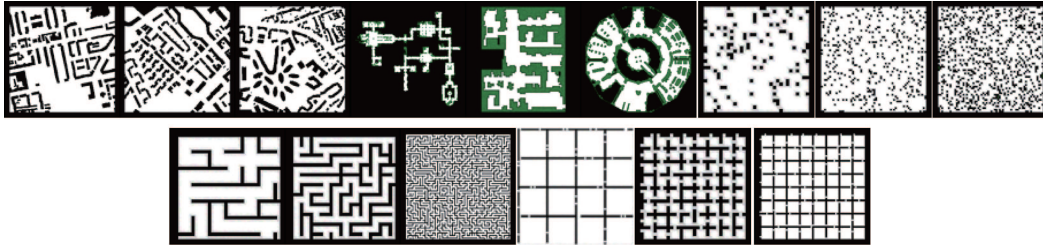


图 3 Grid-based MAPF 基准地图示意图。从左到右、从上到下依次为：城市地图、游戏地图、随机格网、迷宫格网以及放假格网

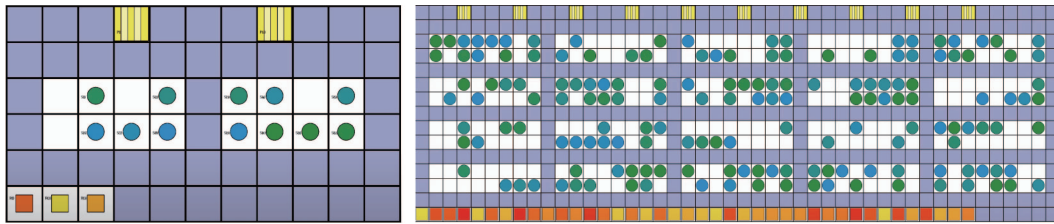


图 4 Asprilo 中的两个典型场景。从左到右依次为：完整的仓库场景（智能体在移动的同时还需搬运货物箱）以及仅移动场景（经典 MAPF 设定）

除了 Grid-based MAPF 外, Asprilo 框架也是 MAPF 研究中被广泛使用的基准测试平台。Asprilo 是一个公共平台[‡], 可模拟自动化仓库, 并为定义、生成、验证和可视化标准仓库规划问题及其解决方案提供工具。Asprilo 支持的规划问题需要智能体在仓库中将货物从一个地点运到另一个地点, 分为不同的场景, 每个场景代表一个独特的问题类型。场景 M 基本上代表 MAPF 问题, 可以作为 MAPF 算法的基准测试场景。图 4 展示了两个 Asprilo 场景: 一个完整的仓库场景 (左), 智能体必须移动货物箱; 一个仅移动场景 (右), 它是一个典型的 MAPF 问题。

3 经典多智能体路径搜寻算法

经典的 MAPF 算法可以分为三类: 基于约简的算法、基于规则的算法和基于搜索的算法。在本节中, 我们将概述这些算法, 并强调它们在完备性和最优性方面的特点。其中, 完备性包括算法在所有 MAPF 问题实例、具有特定特征的问题实例上的完备性, 或不完备性。另一方面, 最优性表示算法是否能够得到最优解, 包括关于各种目标函数的最优、有界次优或次优性。对于特定类别的问题实例来说, 如果 MAPF 算法能够保证对类别内的任何可解问题实例找到解决方案, 或者在有限的时间内判断所给问题实例是无解, 那么这种算法就被认为是完备的。

3.1 基于约简的算法

定义 1 (多智能体路径搜寻问题的布尔可满足性层编码^[20]) 当优化目标为最大完工

[‡]<https://asprilo.github.io/>

时间时, 第 x 个规则层 (时间步) 包含以下整数区间状态变量:

- 对于所有 $v \in V$, 有 $\mathcal{A}_x^v \in \{0, 1, 2, \dots, n\}$, 使得当 $\alpha_x(a_j) = v$ 时有 $\mathcal{A}_x^v = j$;
- 对于所有 $v \in V$, 有 $\mathcal{T}_x^v \in \{0, 1, 2, \dots, 2 \deg_G(v)\}$, 使得 $\mathcal{T}_x^v = 0$ 当且仅当 v 中选择了无操作; $\mathcal{T}_x^v = \sigma_v(u)$ 当且仅当在 v 中选择了具有目标 $u \in V$ 的离开操作; $\mathcal{T}_x^v = \deg_G(v) + \sigma_v(u)$ 当且仅当在 v 中选择了具有 $u \in V$ 作为源的进入操作。

以及约束:

- 对于所有 $v \in V$, 有 $\mathcal{T}_x^v = 0 \Rightarrow \mathcal{A}_{x+1}^v = \mathcal{A}_x^v$ (无操作情况);
- 对于所有 $v \in V$ (离开某个顶点的智能体情况), 有 $0 < \mathcal{T}_x^v \leq \deg_G(v) \Rightarrow \mathcal{A}_x^u = 0 \wedge \mathcal{A}_{x+1}^u = \mathcal{A}_x^u \wedge \mathcal{T}_x^u = \sigma_u(v) + \deg_G(u)$, 其中 $u = o_v^{-1}(\mathcal{T}_x^v)$;
- 对于所有 $v \in V$ (进入某个顶点的智能体情况), 有 $\deg_G(v) < \mathcal{T}_x^v \leq 2 \deg_G(v) \Rightarrow \mathcal{T}_x^u = \sigma_u(v)$, 其中 $u = \sigma_v^{-1}(\mathcal{T}_x^v - \deg_G(v))$ 。

定义 2 (多智能体路径搜寻问题的整数线性规划建模^[21]) 当优化目标为最大完工时间时, 多智能体路径搜寻问题对应的整数线性规划问题的目标函数为:

$$\max \sum_{1 \leq i \leq n} x_{i,i},$$

其中 x 表示该工作^[21] 额外引入的二元整数变量, i 表示智能体序号, j 表示顶点序号。此外, 还包含以下约束:

$$\begin{aligned} \forall e_j, \sum_{i=1}^n x_{i,j} &\leq 1, \\ \forall 1 \leq i, j \leq n, i \neq j, x_{i,j} &= 0, \\ \forall v \in G', 1 \leq i \leq n, \sum_{e_j \in \delta^+(v)} x_{i,j} &= \sum_{e_j \in \delta^-(v)} x_{i,j} \end{aligned}$$

定义 3 (多智能体路径搜寻问题的答案集编程建模^[54]) 当优化目标为最大完工时间时, 我们将多智能体路径搜寻问题表示为答案集编程 (ASP) 程序 P , 其答案集对应于问题的解。我们用一组事实 F_I 描述多智能体路径搜寻问题实例的输入 $I = (G, k, h, O, g)$: $\text{edge}(v, u)$ 表示边 $(v, u) \in E$; $\text{start}(i, v)$ 和 $\text{goal}(i, u)$ 表示每个智能体 $i \leq k$ 的源点和目标顶点 (即 $h(i) = (v, u)$); 最后 $\text{clear}(v)$ 表示 $v \in V \setminus O$ 。

多智能体路径搜寻问题的输出 (P_i, f_i) 为每个智能体 i 表征一条路径规划, 使得该智能体从其初始位置到达目标位置并避开障碍物。我们用原子 $\text{path}(i, t, v)$ 表示路径规划, 该原子指定在时间步长 t 时, 智能体 i 位于顶点 v , 形式上 $f_i(t) = v$ 。ASP 程序 P 递归地定义路径规划。智能体 i 在时间步 0 时访问的第一个顶点是其初始位置 v :

$$\text{path}(i, 0, v) \leftarrow \text{start}(i, v),$$

如果智能体 i 在步骤 $t (0 \leq t < l_i)$ 访问顶点 v , 那么智能体要么在 v 处等待, 要么沿着边 (v, u) 移动到相邻的顶点 u :

$$1\{\text{path}(i, t+1, v), \text{path}(i, t+1, u) : \text{edge}(v, u)\}1 \leftarrow \text{path}(i, t, v)。$$

同时确保智能体不经过障碍物:

$$\leftarrow \text{path}(i, t, v), \text{ not clear } (v),$$

并且每个智能体 i 都到达其目标 v :

$$\begin{aligned} &\leftarrow \text{goal}(i, v), \text{ not visit } (i, v), \\ \text{visit}(i, v) &\leftarrow \text{path}(i, t, v), \end{aligned}$$

其中, $\text{visit}(i, v)$ 表示智能体 i 的路径包含 v 。还需确保智能体之间不发生碰撞:

$$\begin{aligned} &\leftarrow \text{path}(i, t, v), \text{ path}(i', t, v) \\ &(v \in V, 1 \leq i < i' \leq k, 0 \leq t \leq l_i, l_j). \end{aligned}$$

基于约简的 MAPF 算法将 MAPF 问题转换为其他广为人知的组合问题, 如布尔可满足性^[20] (定义 1)、整数线性规划^[21] (定义 2) 以及答案集编程^[23,54] (定义 3) 等。它们在所有 MAPF 问题实例中都是完备的。关于最大完工时间这一优化目标, 这类算法可以保证最优性, 同时也可以等价转换为关于其他优化目标的最优性^[21-22,55]。它们还可以调整为提供有界次优^[56] (得到的规划在用户提供的次优性因子以内) 和次优规划^[57-59]。这类算法在处理具有较小地图大小, 且具有密集放置的机器人的 MAPF 问题实例方面表现出色。例如, 一种以整数线性规划为基础的 MAPF 求解器可以在不到 15 s 的运行时间内以最大完工时间为优化目标, 为一个 24×18 大小的格网中 60 台机器人计算出一个最优解^[59]。

3.2 基于规则的算法

基于规则的算法一般都没有形式化的运筹学定义, 使用一组原子操作来指示机器人在不同情况下的动作, 通常仅针对有限的 MAPF 问题实例类别提供完备性保证。虽然这些算法能有效地遵循预定义的原子操作, 但它们不能保证解的最优。尽管 Push and Swap^[23] 及其扩展算法^[60] 可以在不到 10 s 的时间内为 100 个智能体计算出一个解, 但在理论上并没有完备性保证。Push and Rotate^[24] 是这些算法的变种, 它在格网中至少有两个顶点没有被智能体占用的情况下对 MAPF 问题实例是完备的。TASS^[61] 在解决树 (一种特殊的图) 上的 MAPF 算法的基础上^[62], 对“可解”的树上的 MAPF 问题实例具有完备性。BIBOX^[63] 适用于至少有两个顶点未被智能体占用的双连接图上的 MAPF 问题实例, 而其变体^[64] 适用于至少有两个顶点未被占用的强连通有向图。最后, SAG^[25] 在“连通良好”的格网上是完备的, 对于最大完工时间这一优化目标, 它能够在多项式时间内提供常数因子的最优逼近保证。这类算法同样在处理具有较小地图大小, 且具有密集放置的机器人的 MAPF 问题实例方面表现出色。

3.3 基于搜索的算法

基于搜索的算法同样少见形式化的运筹学定义, 主要采用启发式搜索技术求解 MAPF 问题^[42]。智能体的数目是该类算法的主要瓶颈及核心挑战, 这是因为随着智能体数量的增加, MAPF 问题实例中可能存在的状态数量呈指数增长。下面我们介绍几类典型的启发式搜索算法。

- 基于 A* 的 MAPF 算法^[26,65-66] 旨在通过使用联合状态规划路径, 同时减少需要探索的状态空间大小来解决该类算法的核心挑战。这些算法, 同时适用于两类优化目标, 且对于所有 MAPF 问题实例都是完备的。
- 解耦 MAPF 算法^[67-71] 按照预定的或动态的智能体排序依次为单个智能体规划路径。每个智能体的路径规划在顶点和时间维度上使用 A* 搜索算法, 并将其他智能体已规划的路径视为移动障碍。尽管这类算法往往非常高效, 但缺乏最优性甚至完备性的保证。PIBT^[72] 引入了一个动态方案来确定部分智能体排序, 并确保所有智能体在二连通图上的有限时间内至少到达一次目标顶点, 但可能不一定同时到达。MAPF-LNS^[73] 采用大邻域搜索^[74] (一种局部搜索算法), 通过不断地为一组智能体的子集重新规划路径来改善次优解。该算法的变体, MAPF-LNS2^[75], 同样使用大邻域搜索反复为智能体子集重新规划路径以减少碰撞, 从而改善带碰撞的解, 直至获得无碰撞解。
- 分层 MAPF 算法在低层有效地为每个智能体规划路径, 同时通过在高层实现树搜索来动态整合这些单个智能体的规划。这种方法保证了对所有 MAPF 问题实例的完备性。增长成本树搜索算法^[76] 的目标是最小化成本和, 并进行最佳优先树搜索, 将不同智能体的到达时间组合在高层。该算法评估在低层是否存在不同到达时间组合的无碰撞规划。冲突搜索 (CBS)^[77] 是最受欢迎的最优 MAPF 算法之一, 可优化任意两个目标函数。首先, CBS 独立确定每个智能体的时间最优路径 (忽略可能的碰撞)。然后, 在高层, CBS 对二元约束树执行最佳优先搜索, 每个分支都解决了计算路径中的一个碰撞。在某些时间步内禁止特定智能体占用顶点或边。在低层, CBS 使用 A* 算法在顶点和时间维度之间重新规划路径, 遵守所施加的约束。近年来, 越来越多的 CBS 改进方法被提出, 这里罗列一些代表性工作:
 - Meta-Agent CBS^[77]: 该变体在高层动态地将多个智能体组合成一个元智能体, 并使用 A* 为这些智能体规划路径, 同时在低层考虑它们的联合状态;
 - ICBS^[78]: ICBS 优先解决产生子搜索节点的碰撞, 该子搜索节点的成本高于当前节点, 从而为 CBS 剪枝的高层搜索提供了机会;
 - CBSH^[79] 及其改进^[80]: 通过采用可接受的启发式方法, CBS 的高层最佳优先搜索得到了改进;
 - Disjoint-Splitting CBS^[81]: 这种 CBS 变体以一种特定方式扩展节点, 使得任何可行解都被仅在其子节点中的一个节点但不在两个都被接受的子树下接受, 从而减少了 CBS 高层重复搜索工作;
 - IDCBS^[82]: 将 CBS 中的高层最佳优先搜索替换为迭代深化深度优先搜索;
 - 对称破解 CBS^[83-84] 和 Mutex-Propagation CBS^[85]: 这些方法同时向子节点添加多个约束, 以便在 CBS 的高层搜索中破坏对称性。最佳对称破解 CBS 变体已经在 256×257 大小的格网上进行了仿真实验, 属于在几秒钟内求解包含 100 个智能体 MAPF 问题实例的最佳解决方案;
 - ECBS^[86] 及改进^[87-88]: 在约束树上引入了有界次优搜索, 从而产生了有界次优 CBS。最近的研究^[49] 还为 CBS 开发了约束树的任意时间有界次优搜索。

- 混合 MAPF 算法 将各种基于搜索的方法与基于约减或基于规则的技术相结合, 以提高性能和效率。SMT-CBS^[89] 将可满足性模理论集成到 CBS 的高层搜索中, 利用布尔可满足性求解器最小化任意两个优化目标之一; Lazy CBS^[90] 采用带约束规划求解器而非传统 CBS 方法进行高层搜索, 旨在最小化最大完工时间。BCP 方法^[91-92] 将混合整数规划中的分枝定界法与针对 MAPF 的对称破解技术相结合, 旨在最小化最大完工时间。基于优先级的搜索^[45] 是一种最近提出的分层解耦的算法, 在二叉优先级树上进行深度优先搜索, 以考虑所有可能的智能体排序, 该方法仅对“性质良好”的 MAPF 问题完备, 但已被证明能在仅 30 s 的时间内对包含 600 个智能体的 481×530 格网找出近似最优解; 还有一些算法将原子操作 (基于规则的 MAPF 技术) 和搜索方法结合起来, 例如 MAPP^[93] 探讨了合并单智能体规划的各种方法, 并在“可滑动”的图上求解 MAPF 问题实例。另一策略^[94] 组合图抽象上的 A* 搜索、原子操作和基于约减的方法来求解 MAPF 问题。

4 机器学习赋能多智能体路径搜寻

机器学习作为一种重要技术手段已经被广泛应用于实际问题的解决, 而机器学习显然也可以赋能多智能体路径搜寻^[3]。值得注意的是, 随着深度强化学习在多智能体系统中的快速发展, 多智能体深度强化学习方向上的各种实际应用挑战和解决方案正在浮出水面。无碰撞环境下的路径搜寻对于许多智能体快速高效地完成任务至关重要, 而使用深度强化学习的多智能体路径搜寻是多智能体系统和人工智能领域的一个新研究领域^[2], 同时也是机器学习赋能多智能体路径搜寻领域中的一个举足轻重的研究方向。本节主要综述机器学习赋能的多智能体路径搜寻方法, 从机器学习赋能程度从小到大分别介绍系列机器学习赋能多智能体路径搜寻方法范式: 使用经典方法求解多智能体路径搜寻问题、利用机器学习替换传统方法中的启发式规则、利用机器学习方法选择问题驱动的传统算法、利用机器学习及传统算法进行混合规划以及使用仅机器学习输出端到端规划路径, 其中使用经典方法求解多智能体路径搜寻问题已在第 3 节进行了详细介绍。

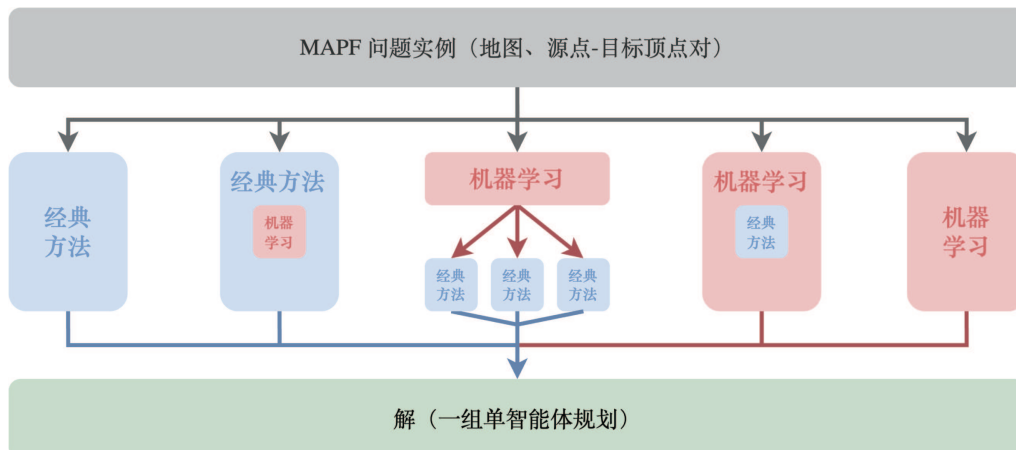


图 5 典型机器学习方法赋能经典多智能体路径搜寻方法范式

图 5 按照机器学习赋能程度从小到大的顺序, 将这些典型范式按照从左到右的顺序排序, 依次为: 使用经典方法求解多智能体路径搜寻问题、利用机器学习替换传统方法中的启发式规则、利用机器学习方法选择问题驱动的传统算法、利用机器学习及传统算法进行混合规划以及使用仅机器学习输出端到端规划路径。

4.1 利用机器学习替换传统方法中的启发式规则

启发式规则是传统多智能体路径搜寻方法中的常用基础, 而利用机器学习来替代启发式规则是深度利用机器学习提升传统方法的重要途径。对搜索节点选择、重规划智能体子集选择、任务分配策略以及优先级、数据生成等方面进行具体替代, 就可以形成机器学习赋能的传统多智能体路径搜寻方法。这类方法的优点是能够继承经典方法的完备性或最优性条件, 但同样也会被经典方法的局限性所约束。

- 搜索节点选择: Huang 等人^[36] 利用监督学习来学习搜索树的节点选择策略, 以增强传统搜索多智能体路径搜寻方法效率。该工作首先使用模仿学习和课程学习来从训练实例中迭代学习不同数量的智能体的节点选择策略, 并进一步将机器学习模块部署到基于冲突的搜索方法中。具体来说, 如算法 1 所示, Huang 等人^[36] 使用监督学习范式来学习一个自适应的 d -值, 从而对经典方法 ECBS^[86] (分层 MAPF 算法) 进行加速。在 ECBS 中, 通用节点选择方法为每个 CT 节点分配一个“ d -值”, 并总是从焦点列表中选择具有最小 d -值的 CT 节点进行扩展。以前研究中常用的节点选择策略都是基于启发式规则, 包括将冲突数量 (用 $|N_{\text{Conf}}|$ 表示) 作为 CT 节点 N 的 d -值、将具有至少一个冲突的智能体对的数量和至少与其他智能体发生冲突的智能体数量作为替代的 d -值等^[36]。

算法 1 机器学习赋能的 ECBS 算法^[36]

- 步 1 输入: 一个 MAPF 实例及次优性因子 w
- 步 2 生成一个初始解的根 CT 节点 R
- 步 3 初始化开放列表 $N \leftarrow \{R\}$
- 步 4 $LB \leftarrow RLB$, 并初始化焦点列表 $F \leftarrow \{R\}$
- 步 5 while N 不为空 do
- 步 6 $N \leftarrow$ 在 F 中 d -值最小的 CT 节点
- 步 7 if $N_{\text{Conf}} = \emptyset$
- 步 8 返回 N_{sol}
- 步 9 end if
- 步 10 从开放列表和焦点列表中删除 N
- 步 11 if $\min_{N \in N} N_{LB} > LB$
- 步 12 $LB \leftarrow \min_{N \in N} N_{LB}$
- 步 13 $F \leftarrow \{N \in N : N_{LB} \leq wLB\}$
- 步 14 end if
- 步 15 在 N_{Conf} 中选择一个冲突
- 步 16 为 N 生成两个子 CT 节点 N^1 和 N^2
- 步 17 调用低级搜索为 N^i 计算 $N_{\text{Sol}}^i, N_{\text{Cost}}^i$ 和 N_{Conf}^i , 其中 $i = 1, 2$
- 步 18 将 N^i 添加到 N , 其中 $i = 1, 2$
- 步 19 if $N_{\text{Cost}}^i \leq wLB$
- 步 20 将 N^i 添加到 F , 其中 $i = 1, 2$

步 21 返回 无解
 步 22 end while
 步 23 返回 无解

- 重规划智能体子集选择: Huang 等人^[38] 利用监督学习替换传统大型邻域搜索算法 LNS (Large Neighbourhood Search) 中的启发式规则。该工作使用监督学习来学习如何从子集集合中选择智能体的子集, 并通过重新规划提高大型邻域搜索方法 (LNS) 的质量。Ederer^[95] 利用监督学习替换传统基于冲突的搜索方法 (Conflict-Based Search) 中的启发式规则。基于冲突的搜索方法旨在通过在最佳优先搜索中迭代解决冲突来优化解决多智能体路径搜寻问题。该工作提供了在搜索过程之前找到有利分配以形成元智能体的策略, 即训练一个机器学习模型来预测有益的合并候选者。Huang 等人^[36] 同样利用监督学习替换传统基于冲突的搜索方法中的启发式规则。该工作首先利用一种用于冲突选择的预测器, 并提出了一种用于冲突选择的机器学习框架, 该框架观察预测器所做出的决策, 并学习由线性排序函数表示的冲突选择策略, 以准确地快速地模仿预测器的决策。算法 2 展示了 Huang 等人^[38] 如何利用监督学习方法赋能 LNS 算法的伪代码, 蓝色行表示相较于经典 LNS 算法新增的算法流程。可以看到, Huang 等人^[38] 采用监督学习方法学习了一个对于所有智能体的排序策略, 从而替换启发式规则进行子集的选取。

算法 2 机器学习赋能的 LNS 算法 MAPF-ML-LNS^[38]

步 1 输入: MAPF 实例 I , 排名函数 π 以及智能体集样本数 S
 步 2 $P = \{p_i : i \in [k]\} \leftarrow \text{runInitialSolver}(I)$
 步 3 初始化销毁启发式的权重 ω
 步 4 while 运行时间限制未超过 do
 步 5 $\mathcal{A} \leftarrow \emptyset$
 步 6 for $i = 1$ 到 S 执行 do
 步 7 $H \leftarrow \text{selectDestroyHeuristic}(\omega)$
 步 8 $\mathcal{A} \leftarrow \mathcal{A} \cup \text{selectAgentSet}(I, H)$
 步 9 end for
 步 10 计算所有 $A \in \mathcal{A}$ 的 $\phi(A)$
 步 11 计算所有 $A \in \mathcal{A}$ 的 $\pi(\phi(A))$
 步 12 for $A \in \mathcal{A}$ 按照 $\pi(\phi(A))$ 的降序执行 do
 步 13 $P^- \leftarrow \{p_i : a_i \in A\}$
 步 14 $P^+ \leftarrow \text{runReplanSolver}(I, A, P \setminus P^-)$
 步 15 更新销毁启发式的权重 ω
 步 16 if $\sum_{p \in P^+} l(p) < \sum_{p \in P^-} l(p)$
 步 17 $P \leftarrow (P \setminus P^-) \cup P^+$
 步 18 中断, 跳出循环
 步 19 end if
 步 20 end for
 步 21 end while
 步 22 返回 $P = 0$

- 任务分配策略: Langlois^[96] 利用监督学习方法进行任务分配后, 再使用传统搜索算

法进行路径规划。任务分配和路径规划是多智能体系统的两个问题, 其中没有一个固定算法在所有环境中提供最佳解决方案。该工作设计了一种元推理策略, 并利用监督学习离线提升该策略, 元推理策略决定何时运行任务分配算法, 并决策使用何种方法进行多智能体路径搜寻。如图 6 所示, 对于每个智能体, 其将当前局部信息 (如位置等) 传递给经典元策略, 元策略将该信息编码后传递给机器学习模型。机器学习模型根据状态编码预测该路径的成本 (以及可能产生冲突等), 并将预测结果返回给元策略。元策略最后根据所有信息进行任务分配 (即将源点与目标顶点进行匹配), 并采用经典 MAPF 算法进行求解。

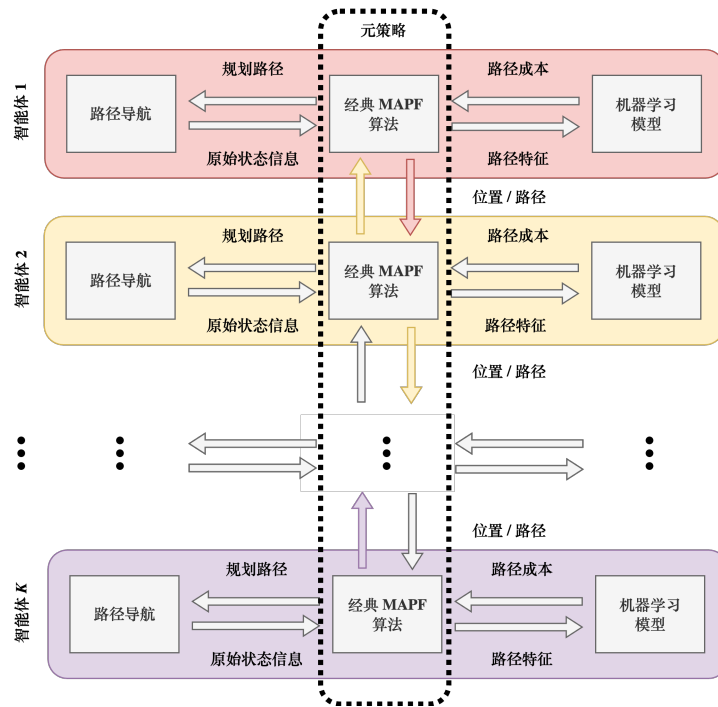


图 6 机器学习驱动的任务分配示意图^[96]

- 优先级学习: Zhang 等人^[97] 利用监督学习来学习“更优”的优先级, 并进一步使用传统基于优先级的搜索多智能体路径搜寻方法。优先级规划是一种快速而流行的解决多智能体路径搜寻框架, 但该类解决方案的质量在很大程度上取决于智能体的预测优先级。该工作提出了一个机器学习驱动的优先级学习框架, 并开发了两个机器学习算法, 分别在总优先级和部分优先级排序上进行训练。如算法 3 所示, 所有蓝色行都需要运用到算法学习到的智能体优先级关系。

算法 3 PBS 高级搜索^[97] (初始优先级排序为 \preceq_0)

步 1 $\preceq_{\text{Root}} \leftarrow \preceq_0$

步 2 $\text{Root.plan} \leftarrow \emptyset$

步 3 **for** 每一个 $i \in E[M]$ **do**

步 4 $\text{success} \leftarrow \text{更新计划}(\text{Root}, a_i)$; {如果 $\preceq_0 = \emptyset$, 则总是成功}


```

步 5   if not success then
步 6     返回 “没有解决方案”
步 7   end if
步 8   end for
步 9   Root.cost  $\leftarrow$  在 Root.plan 中的到达时间之和
步 10  STACK  $\leftarrow$  {Root}
步 11  while STACK 不为空 do
步 12     $N \leftarrow$  STACK 的顶部节点
步 13    STACK  $\leftarrow$  STACK  $\setminus$  {N}
步 14    if N 没有冲突 then
步 15      返回 N.plan
步 16    end if
步 17     $C \leftarrow$  在 N.plan 中的第一个顶点或边缘冲突  $(a_i, a_j, \dots)$ 
步 18    for 每一个  $a_i$  在  $C$  中 do
步 19       $N' \leftarrow$  新节点
步 20       $N'.plan \leftarrow N.plan$ 
步 21       $N'.constraints \leftarrow N.constraints \cup \{(a_i, \dots)\}$ 
步 22       $N' \leftarrow \{N \cup \{i\} \preceq i\}$ 
步 23      success  $\leftarrow$  更新计划( $N', a_i$ )
步 24      if success then
步 25         $N'.cost \leftarrow$  在  $N'.plan$  中的到达时间之和
步 26      end if
步 27    end for
步 28    在  $N'.cost$  的非递增顺序中将新节点  $N'$  插入到 STACK 中
步 29  end while
步 30  返回 “没有解决方案”
步 31  更新计划  $N, a_i$ 
步 32  LIST  $\leftarrow$  在部分排序集  $\{(i') \cup \{j | i' \preceq j\}, \preceq N\}$  上的拓扑排序
步 33  for 每一个  $j \in$  LIST do
步 34    if  $j = i$  或  $\exists a_k : k \preceq j, a_j$  在  $N.plan$  中与  $a_k$  冲突
步 35      更新  $N.plan$ , 通过调用为  $a_j$  进行的低级搜索, 避免与所有优先级更高的
步 36       $a_k (k \preceq j)$  冲突
步 37      if 低级搜索没有返回路径 then
步 38        返回 false
步 39      end if
步 40    end if
步 41  end for
步 42  返回 true

```

- 数据生成: Okumura 等人^[98] 利用监督学习方法预测智能体未来轨迹, 如图 7 以及图 8 所示, 以辅助传统搜索算法决策。图 7 展示了机器学习模型采用监督学习范式来预测下一个时刻智能体应该移动的位置。图 8 展示了对于新的问题实例, 该算法

首先使用训练好的机器学习模型为所有智能体生成多个路径, 最后形成所有智能体的路线图, 用以大幅减小经典 MAPF 算法的搜索空间。多智能体路径搜寻系列方法中, 一种很有前途的方法是构建近似空间的图 (称为路线图), 然后应用多智能体路径搜寻方法来输出一组无冲突路径, 但如何构建有效适用于多智能体的路线图在很大程度上仍有待探索, 路线图可以认为是传统方法中的必要信息。该论文提出了一种新的路线图概念, 称为合作定时路线图, 该路线图使每个智能体能够专注于其在潜在解决方案路径周围的重要位置, 考虑其他智能体的行为, 以避免智能体间冲突, 同时在时间方向上进行扩展, 使导出“定时”解决方案路径变得容易。为了构建该路线图, 该论文开发了一种机器学习方法, 从相关问题实例和看似合理的解决方案集中学习路线图生成模型, 从而应用该机器学习模型到新的、零样本未见的问题实例。

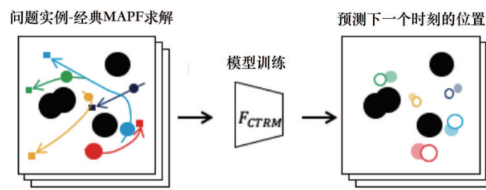


图 7 Okumura et al.^[98] 算法的训练阶段

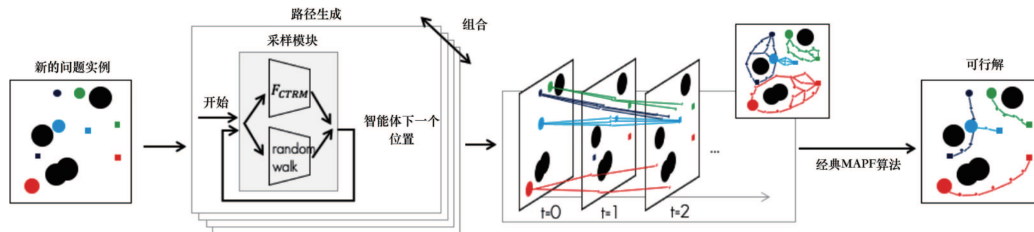


图 8 Okumura et al.^[98] 算法的推断阶段

4.2 利用机器学习方法选择问题驱动的传统算法

使用机器学习模型对经典 MAPF 算法中的关键模块进行替代, 本质还是以经典方法为主导。在本节中, 我们将介绍机器学习赋能程度更深的一类范式, 即利用机器学习方法选择问题驱动的传统算法。不同的传统方法适应于具有不同结构特征的多智能体路径搜寻问题, 且没有一个多智能体路径搜寻方法能主导所有的基准多智能体路径搜寻问题, 也没有明确的、可证明的指导方针来确定何时应该使用哪种算法。因此, 如何利用机器学习方法来选择问题驱动的传统多智能体路径搜寻方法, 是在更深层面使用机器学习技术求解多智能体路径搜寻问题的重要科学问题。这类方法的优缺点与利用机器学习替换传统方法中的启发式规则类似, 同样能够继承经典方法的完备性或最优性条件, 但同时会受到经典方法的局限性约束。但由于这类方法能够同时利用一批经典方法, 因此能够在这些方法的局限性上进行互补。

Kaduri 等人^[100] 利用监督学习方法对多智能体路径搜寻问题做分类, 并通过有针对性决定所选择使用的传统算法。该工作提出了首个多智能体路径搜寻算法选择模型, 并提出了两种学习该算法选择模型的方法, 即基于人工设计特征的标准监督学习方法和基

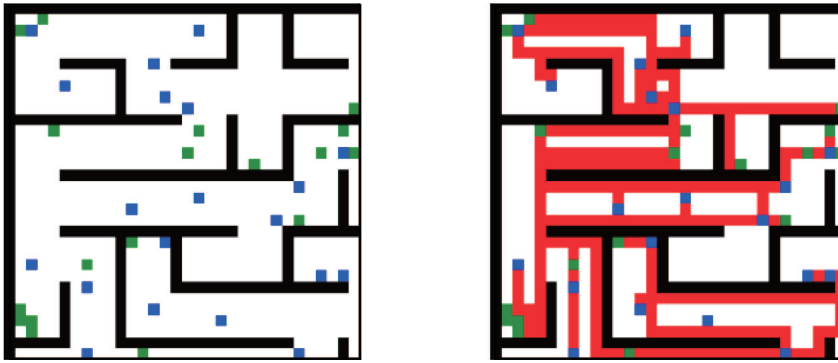


图 9 两种典型的将 MAPF 问题实例映射到图像空间的方法^[99]

于深度卷积神经网络的分类方法 (将多智能体路径搜寻问题映射到图像空间, 如图 9 所示)。图 9 中, 左图将 MAPF 实例映射到 RGB 格式的图像空间中, 对于地图中的障碍物, 该像素位置填 1, 可行动位置则填 0; 对于智能体的起始位置, 像素值设置为绿色, 目标位置则设置为蓝色, 起始位置和目标位置之间不存在对应关系^[100-101]。右图所有设定与左右一致, 但额外加入了起始位置和目标位置之间的对应关系, 即画出每一对起始位置和目标位置之间的最短路径, 不考虑该路径是否与其他路径冲突, 且路径所经过的像素设置为红色^[99,102]。该工作中候选经典 MAPF 算法都来自于基于搜索的算法, 包括基于 A* 的搜索算法^[26,65] 以及分层 MAPF 算法^[76-78,80]。Sigurdson 等人^[101] 同样利用监督学习对多智能体路径搜寻问题进行分类, 并进一步决定所采取的传统算法。该工作基于深度学习从给定多智能体路径搜寻问题的算法组合中自动选择最佳多智能体路径搜寻方法, 候选经典 MAPF 算法都来自于基于启发式搜索的方法中的解耦 MAPF 算法^[67,69,71]。Ren 等人^[99,102] 开发了基于深度卷积神经网络的 MAPFAST (Multi-Agent Path Finding Algorithm SelecTor) 方法。其在之前算法基础上, 通过给模型的实例嵌入单智能体最短路径, 以及利用补充损失函数, 来提高模型性能。该方法所选取的候选经典 MAPF 算法包括基于启发式搜索的算法中的分层 MAPF 算法^[77,79-80], 以及基于约简的方法中的布尔可满足性建模方法^[55] 和整数线性规划建模方法^[91]。Alkazzi 等人^[103] 在 MAPFAST 的基础上采用基于组合的算法选择机制, 该机制可以帮助选择最适合给定任务的算法, 同时利用候选算法的互补优势。该工作提出了一种更小更准确的基于深度学习的架构, 并旨在将其部署在智能体队伍管理系统中。该方法选取的候补经典 MAPF 算法与 Ren 等人^[99,102] 一致。

表 1 选择问题驱动的经典 MAPF 算法的机器学习算法总结

算法	候选经典算法
Kaduri 等人 ^[100]	基于 A* 的搜索算法 ^[26,65] 、分层 MAPF 算法 ^[76-78,80]
Sigurdson 等人 ^[101]	解耦 MAPF 算法 ^[67,69,71]
Ren 等人 ^[99,102]	分层 MAPF 算法 ^[77,79-80] 、布尔可满足性建模方法 ^[55] 、整数线性规划建模方法 ^[91]
Alkazzi 等人 ^[103]	分层 MAPF 算法 ^[77,79-80] 、布尔可满足性建模方法 ^[55] 、整数线性规划建模方法 ^[91]

利用机器学习技术来选择合适的匹配多智能体路径搜寻方法, 是通过最直接的方式利用机器学习赋能多智能体路径搜寻问题的解决, 然后更深入的结合是利用机器学习来增强传统方法的能力, 而不仅仅是选择传统方法。

4.3 利用机器学习及传统算法进行混合规划

直接利用机器学习技术形成端到端的多智能体路径搜寻策略, 虽然在实现方式上简单有效, 但是并未真正将机器学习技术与传统多智能体路径搜寻方法融合, 本小节将重点介绍利用结合机器学习与传统方法进行混合规划。这类方法的优点在于可以利用机器学习方法突破经典方法在问题规模以及求解时间上的限制, 但同样也无法保证算法的完备性或最优性。本小节将从传统方法引导规划、两阶段规划、启发式规则驱动等角度展开介绍。

- 传统方法引导规划: 利用传统方法生成路径规划训练数据或启发式规则训练数据, 并进一步知道损失函数或奖励函数设计。Sartoretti 等人^[33]提出了一种新的多智能体路径搜寻框架 PRIMAL, 其结合了强化学习和模仿学习来教授完全去中心化的政策, 其中智能体在一个部分可观察的世界中反应性地在线规划路径, 同时表现出隐含的协同。该框架扩展了之前在协作政策的分布式学习方面的工作, 在培训期间引入了专家多智能体路径搜寻规划师的演示, 以及仔细的奖励塑造和环境采样。Damani 等人^[104]在 PRIMAL 方法基础上设计了新的分布式强化学习框架, 即 PRIMAL²。该方法中的智能体学习完全去中心化的策略, 以便在部分可观察的世界中在线反应性地规划路径, 进一步通过识别行为和约定将之前的工作扩展到高度结构化和受限的世界, 该工作在低密度稀疏占用的世界中是有效的, 这些行为和约定可以改善隐式智能体协同, 并通过构建新的局部智能体观察和各种训练辅助工具来实现它们的学习。
- 两阶段规划: Sartoretti 等人^[31]结合多智能体强化学习 PRIMAL 和传统搜索算法 ODrM*, 其先用机器学习方法 PRIMAL 计算, 再用传统 ODrM* 来微调。Virmani 等人^[105]利用监督学习(模仿学习)和传统搜索算法 M* 形成两阶段规划框架。该工作通过采用动态耦合策略有效地绕过了维度诅咒: 首先以完全解耦的方式规划智能体, 忽略了智能体之间的潜在冲突; 然后智能体要么遵循他们各自的计划, 要么耦合在一起进行计划以解决他们之间的冲突。该工作旨在通过改进每个智能体的个人计划来减少冲突的数量(从而提高整体搜索效率)。通过利用 Visual Transformer, 该工作开发了一个基于学习的单智能体规划器, 该规划器为单个智能体进行规划, 同时关注映射的结构和可能发生冲突的其他智能体。
- 传统算法启发式规则与机器学习方法深度融合: Reijnen 等人^[34]提出了一种称为 WHCA*S-RL 的算法, 该算法将深度强化学习与启发式方法相结合, 用于求解多智能体强化学习。其中, 深度强化学习用于确定行进方向, 而启发式方法则监督基于路段的布局中的规划。Wang 等人^[106]引入了一种全局引导的强化学习方法, 该方法结合了一种新的奖励结构, 可推广到任意环境, 并将 G2RL 应用于以全分布式反应方式解决多智能体路径规划问题。Chen 等人^[107]将多智能体强化学习框架与传统搜索算法 ODrM* 相结合。该论文提出了一种解决多智能体路径搜寻问题的新方法。在这种方法中, 通过提出热监督对比损失, 将专家数据转换为监督信号, 并将其与强化学习相结合, 以教授完全分散的策略。智能体在一个部分可观察的世界中反应性地在线规划路径, 同时表现出隐含的协同, 而不与他人交流。进一步在策略网络中引入了自注意机制, 提高了策略网络从观测数据中提取智能体之间协作信息的能力。Li 等人^[18]同样将多智能体强化学习框架与传统搜索算法 ODrM* 相结合。提出了一种分布式通信学习方法, 该方法在去中心化多智能体强化学习框架内, 将隐式规划优先级纳入

通信拓扑。隐式优先级学习模块与经典的耦合规划器相结合,可以形成动态通信拓扑,也建立了有效的冲突避免机制。Wang 等人^[108]提出了一种 SCRIMP 方法,其中智能体通过依赖于基于改进的转换器的高度可扩展的全局/本地通信机制,从甚至非常小的 FOV 中学习单个策略。为了进一步为智能体人配备了基于状态价值的平局打破策略,以进一步提高对称情况下的绩效,并引入内在奖励来鼓励探索,同时缓解长期信用分配问题。

机器学习技术作为重要补充,可以从多个角度提升传统多智能体路径搜寻方法的规划效率,上述内容全面介绍了机器学习赋能的多智能体路径搜寻方法。然而机器学习技术的使用依然值得进一步提升,其可以更好地赋能多智能体路径搜寻方法。

4.4 使用机器学习端到端输出规划路径

机器学习技术可以应用到多智能体路径搜寻传统方法选择,以及传统多智能体路径搜寻方法中启发式规则替换或必要信息生成等,然而更多的机器学习驱动多智能体路径搜寻方法旨在通过机器学习技术的端到端学习能力,形成端到端规划路径能力。监督学习、强化学习、演化学习、演化博弈论等均可用于构建机器学习驱动的端到端多智能体路径搜寻方法。这类方法属于纯粹的数据驱动方法,因此其优点在于能够通过数据量与模型大小的提升来实现性能的提升,但缺点在于需要高质量的训练数据,同时也缺乏完备性或最优性保证。

- 监督学习: Abreu^[39] 利用监督学习和模仿学习技术建立端到端多智能体路径搜寻方法。多智能体路径搜寻 (MAPF) 是协同现实世界中的智能体系统所广泛需要的。新的方法转向深度学习来解决 MAPF 实例,主要使用具有高计算成本的强化学习。我们提出了一种监督学习方法,使用更小、成本更低的模型来解决 MAPF 实例。Bignoli^[109] 同样采用监督学习结合模仿学习的框架,通过利用图神经网络学习由智能体到智能体通信的局部策略来解决多智能体路径搜寻问题。Li 等人^[110] 提出了一个组合模型,自动综合智能体在受限工作空间中的本地通信和决策策略。该工作的架构由卷积神经网络 (CNN) 和图神经网络 (GNN) 组成,前者从局部观测中提取足够的特征,后者在智能体之间传递这些特征。通过模仿专家,并在只涉及局部通信和局部观测的分散规划中在线使用由此产生的模型。Li 等人^[111] 延续采用监督学习结合模仿学习的机器学习框架。在大规模多智能体系统中,找到分散的路径规划和协同解决方案是提高系统性能的关键。而图神经网络由于其在去中心化多智能体系统中学习通信策略的能力而变得流行起来。该工作通过引入一种新的消息感知图注意力机制来允许消息互相依赖,该机制确定从各种相邻智能体接收的消息中特征的相对重要性。Sinkar 等人^[112] 提出了一种解决智能体或障碍物动态添加导致冲突或路径重叠问题,其使用由 A* 算法制作的数据集上训练的深度学习模型,可以在运行时为每个智能体做出决策,同时确保它不会与任何障碍物或任何其他智能体碰撞,并在任何时间点处理环境中的各种动态变化,同时不依赖于预计算。Chen 等人^[107] 引入图神经网络和智能体之间的通信,提高分布式策略引导的智能体规划路径在复杂环境中的性能,并显著降低系统的鲁棒性。该工作通过引入 Transformer 模型设计了一种新的模仿强化学习框架,该框架使得算法能够在复杂环境中表现良好,而不依赖于智能体之间的通信。
- 无模型多智能体强化学习: Guan 等人^[40] 利用多智能体强化学习框架,并引入通信学习技术。为了提高动态环境下路径规划学习过程的有效性和效率,该工作在 Actor-

Critic 强化学习框架下引入了一种基于注意力和 BicNet 的强化多智能体路径搜寻算法。Ma 等人^[113] 同样利用多智能体强化学习框架, 并引入通信学习技术。该工作提出了一个精心设计的模型, 该模型利用附近智能体和障碍物的局部状态, 并输出每个智能体执行的最优动作, 其由三个主要组成部分: 1) 观测编码器: 其使用 CNN 提取局部部分观测, GRU 充分利用历史信息; 2) 通信模块: 其使用注意力机制将智能体的部分观测与其邻居相结合; 3) 决策模块: 其目的是输出最终的行动策略。Ma 等人^[114] 引入通信学习, 并与深度 Q 学习相结合, 为多智能体路径搜寻提供了一种新的基于学习的方法, 其中智能体通过图卷积网络实现协作。Ma 等人^[115] 同样利用通信学习建立面向多智能体路径搜寻的多智能体强化学习框架。通过深度强化学习或模仿学习进行通信学习最近被证明是解决多智能体路径搜寻的有效方法。该工作提出了一个请求-应答场景, 并提出了简单有效的决策因果通信模型, 使智能体能够选择邻居在训练和执行期间进行通信。具体地, 只有当邻居的存在导致中央智能体上的决策调整时, 该邻居才被确定为相关的和有影响力的。这种判断仅基于智能体的局部观察来学习, 因此适合于去中心化执行来处理大规模问题。Ye 等人^[116] 将强化学习与通信学习相结合, 其中智能体根据优先级选择其通信目标, 并屏蔽冗余通信链路, 进一步使用基于图神经网络的特征交互网络来实现信息聚合。该工作引入了一种额外的死锁检测机制, 以增加智能体逃离死锁的可能性。Xu 等人^[117] 与前面基于通信学习的工作不同, 其提出了一种新的基于深度强化学习的去中心化多智能体路径搜寻方法, 其结合重要的课程学习, 并使用局部和全局指导机制来帮助智能体规划可行的路径。Davydov 等人^[118] 采用值分解的多智能体强化学习框架来解决多智能体路径搜寻问题, 为了应对与学习合作行为相关挑战, 智能体需要相互让步才能完成任务, 该工作使用混合 Q 网络来补充学习个体策略。van Knippenberg 等人^[119] 提出了一种基于强化学习的多智能体路径搜寻方法来学习智能体的本地路由策略, 从而减轻了手动设计启发式机制的需要。然而该方法依赖于图卷积网络来处理任意图, 该工作提出了单智能体和多智能体强化学习方法, 展示了多智能体设置如何利用智能体财产和局部图拓扑的相似性来减少训练时间的可能性。

- 有模型多智能体强化学习: Skrynnik 等人^[37] 提出利用强化学习近端策略优化方法 PPO 学习路径规划策略, 并利用基于模型的强化学习蒙特卡罗树搜索 (MCTS) 方法实现避障。该工作提出了一种新框架, 其将问题分解为两个子任务: 达到目标和避免冲突。为了完成每一项任务, 我们利用强化学习方法, 如深度蒙特卡罗树搜索、Q 混合网络和策略梯度方法来设计将智能体的观察映射到行动的策略。接下来, 我们引入策略混合机制, 最终形成一个单一的混合策略, 允许每个智能体展现两种类型的行为——单独的行为 (达到目标) 和合作的行为 (避免与其他智能体的冲突)。我们进行了一项广泛的实证评估, 结果表明, 在这类问题上, 所提出的混合策略显著优于独立的现有强化学习方法。Zhang 等人^[13] 提出了一种具有多步前向树搜索 (MATS) 策略的去中心化多智能体强化学习 (MARL) 框架, 以做出有效的决策。
- 演化强化学习: Liu 等人^[35] 提出了一种具有进化强化学习 (MAPPER) 方法的去中心化部分可观测多智能体路径搜寻, 以学习混合动态环境中有效的局部规划策略。基于强化学习的方法在具有目标条件稀疏奖励的长期任务中通常会性能下降, 因此在全局规划器的指导下将远程导航任务分解为许多更容易的子任务, 这提高了智能体在大型环境中的性能。此外, 大多数现有的多智能体规划方法要么假设周围环

境的完美信息, 要么假设附近动态智能体的同质性, 这在实践中可能不成立。该工作使用基于图像的代表来建模动态障碍物的行为, 并在没有同质性假设的混合动态环境中训练策略。为了确保多智能体训练的稳定性和性能, 进一步提出了一种可以轻松扩展到大型复杂环境的进化训练方法。

- **演化博弈论:** Paul 等人^[32] 考虑一组同构和自治的智能体在先前未知的随机环境中导航的路径搜寻问题。该工作的解决方案基于演化博弈论的思想, 即复制表现良好的政策, 减少表现不佳的政策, 所提出的算法在计算上比深度强化学习方法快至少一个数量级。如算法 4 所示, 演化博弈论驱动的 MAPF 算法整体流程大致如下: 首先随机初始化每个智能体的路径规划策略; 其次, 采用当前策略为每个智能体随机采样一条路径, 并更新智能体在某个状态 (顶点) 执行某个动作 (上下左右以及停止不动) 的计数器; 对于成功到达目标顶点的路径, 以一定概率增大该路径上所有状态选择特定动作的概率, 否则以一定概率减小该概率; 最后根据更新后的策略重新采样, 不断循环上述过程, 直到策略收敛, 即更新前后所有状态选择所有动作的概率之差小于某个阈值。值得注意的是, 所有智能体共享相同的策略。

算法 4 演化博弈论方法求解 MAPF 问题^[32]

```

步 1 初始化:  $\pi_{\text{rand}}(a|s) \leftarrow \frac{1}{|A|}$ ,  $\text{counter}(s, a) \leftarrow -L, \forall s \in S, a \in A$ 
步 2 for  $e = 1$  to Number of Episodes
步 3 初始化轨迹:  $\tau^i = ()$ 
步 4 Sample:  $s_0^i, \dots, s_i^i \sim I$ 
步 5  $j \leftarrow 0$ 
步 6 while  $j < T$  do
步 7   for  $i = 1$  to  $N$  do
步 8     if  $s_i^j \notin \Phi$  then
步 9        $a_i^j \sim \pi_{\text{rand}}$ 
步 10       $\tau^i := \tau^i \oplus (s_i^j, a_i^j)$ 
步 11       $(s_i^j, a_i^j, s_i^{j+1}) \sim \Delta$ 
步 12     end if
步 13     if  $s_i^{j+1} \in \Phi$  then
步 14       调用  $\text{UpdateFunction}(\tau_{j+1}^i)$ 
步 15     end if
步 16   end for
步 17    $j \leftarrow j + 1$ 
步 18 end while
步 19 for  $i$  where  $s^\top \notin \Phi$  do
步 20   调用  $\text{UpdateFunction}(\tau_{j+1}^i)$ 
步 21 end for
步 22 end for
步 23  $\text{UpdateFunction}(\tau^\ell)$ 
步 24  $u = d(s^0, s^\ell)$ 
步 25 if  $s^\ell \in G$  then
步 26   if  $u \leq \eta$  then

```

```

步 27   update_prob = 1 - (1 - u)α
步 28   else
步 29   update_prob =  $\frac{1}{u}$ 
步 30   end if
步 31   if rand(0, 1) ≥ update_prob
步 32   counter( $s_k, a_k$ ) := counter( $s_k, a_k$ ) +  $\nu$ 
步 33   end if
步 34   else
步 35   if  $u \geq \beta$  then
步 36   counter( $s_k, a_k$ ) := counter( $s_k, a_k$ ) -  $\mu$ 
步 37   end if
步 38   end if

```

5 未来的挑战与发展

前文全面概述了机器学习赋能的多智能体路径搜寻方法, 并从机器学习与经典方法融合程度强弱的角度对当前的方法做了分类。本章节将进一步探讨现有机器学习驱动的多智能体路径搜寻工作面临的主要挑战, 并在此基础上讨论一些未来的研究方向^[2-3]。

- 提高算法的泛化能力。基于机器学习技术的多智能体路径搜寻问题通常采用神经网络来处理传感器数据, 虽然这些算法在训练环境中表现良好, 但缺乏从一个环境到另一个环境以及从模拟环境到真实环境的泛化能力。
- 提高算法的采样效率来加快学习速度。当环境更加复杂时, 数据量会更大, 智能体和环境之间的交互次数会更多。大多数现有基于强化学习技术的研究, 采用经验回放技术来提高算法的采样效率, 这限制了算法在数据难以收集的、现实场景中的问题求解能力。
- 设定更有效的奖励函数来提高探索效率。在路径搜寻问题中, 无模型多智能体强化学习算法主要依靠探索来找到最优策略, 但由于奖励的稀疏性, 智能体只有在达到目标时才会获得奖励, 因此智能体将在环境中执行许多无意义的探索行为。
- 增强理论认知与带约束路径搜寻问题求解能力。深入发展针对机器学习驱动的多智能体决策路径搜寻算法理论认识与收敛性保障, 并将 MAPF 技术与考虑到任务的时间约束、智能体的高阶动态约束等复杂的现实场景相结合。

6 总 结

本篇综述系统介绍了机器学习技术在经典多智能体路径搜寻问题上的应用, 从机器学习算法赋能程度角度, 对现有机器学习赋能的 MAPF 方法做了一定程度上的总结和归纳。最后, 总结了机器学习驱动的多智能体路径搜寻算法目前面临的主要挑战。我们从数学建模以及方法论两个角度, 提出一些未来潜在研究方向, 以期可帮助领域内的研究者, 并促进机器学习方法在经典多智能体路径搜寻领域的发展。首先, 从数学建模的角度, 目

前对 MAPF 问题进行了形式化建模的方法, 主要从基本的优化目标以及冲突约束的角度对问题进行建模。但机器学习赋能的 MAPF 算法更应该关注的则是算法稳健性和泛化性问题, 将这两点加入到形式化建模中的工作还较少。其次, 从方法论角度, 现有机器学习赋能的 MAPF 方法主要关注的还都是在算法设计部分。然而, MAPF 是一个非常实际的应用问题, 其整个问题求解流程包括问题建模、求解以及部署一整条管线。从运筹学角度来看, 在求解一个实际的运筹学问题过程中, 更加消耗时间和人力成本的却往往体现在问题建模以及算法部署阶段。因此, 如何利用机器学习方法赋能这两个阶段, 是更贴近现实应用的关键问题。

参考文献

- [1] Stern R, Sturtevant N, Felner A, et al. Multi-agent pathfinding: Definitions, variants, and benchmarks [J]. 2019, arXiv:1906.08291.
- [2] Qiu T, Cheng Y. Applications and challenges of deep reinforcement learning in multi-robot path planning [J]. *Journal of Electronic Research and Application*, 2021, **5**(6): 25-29.
- [3] Ma H. Graph-based multi-robot path finding and planning [J]. *Current Robotics Reports*, 2022, **3**(3): 77-84.
- [4] Hönig W, Kiesel S, Tinka A, et al. Persistent and robust execution of MAPF schedules in warehouses [J]. *IEEE Robotics and Automation Letters*, 2019, **4**(2): 1125-1131.
- [5] Li J, Tinka A, Kiesel S, et al. Lifelong multi-agent path finding in large-scale warehouses [C]//*Association for the Advancement of Artificial Intelligence*, 2021.
- [6] Varambally S, Li J, Koenig S. Which MAPF model works best for automated warehousing? [C]//*Symposium on Combinatorial Search*, 2022.
- [7] Wen L, Liu Y, Li H. CL-MAPF: Multi-agent path finding for car-like robots with kinematic and spatiotemporal constraints [J]. *Robotics and Autonomous Systems*, 2022, **150**: 103997.
- [8] Huang T, Liu J, Zhou X, et al. V2X cooperative perception for autonomous driving: Recent advances and challenges [J]. 2023, arXiv: 2310.03525.
- [9] Chandra R, Maligi R, Anantula A, et al. Optimal and efficient multiagent path finding with strategic agents for social navigation [J]. *IEEE Robotics and Automation Letters*, 2023.
- [10] Calderón-Arce C, Solis-Ortega R, Bustillos-Lewis T. Path planning on static environments based on exploration with a swarm robotics and RRG algorithms [C]//*2018 IEEE 38th Central America and Panama Convention*, 2018: 1-6.
- [11] Calderón-Arce C, Solis-Ortega R. Swarm robotics and rapidly exploring random graph algorithms applied to environment exploration and path planning [J]. *International Journal of Advanced Computer Science and Applications*, 2019, **10**(5).
- [12] Fekete S P, Keldenich P, Kosfeld R, et al. Connected coordinated motion planning with bounded stretch [J]. *Autonomous Agents and Multi-Agent Systems*, 2023, **37**(2): 43.
- [13] Zhang Y, Qian Y, Yao Y, et al. Learning to cooperate: Application of deep reinforcement learning for online AGV path finding [C]//*International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2020.
- [14] Kou N M, Peng C, Ma H, et al. Idle time optimization for target assignment and path finding in sortation centers [C]//*Association for the Advancement of Artificial Intelligence*, 2020.

-
- [15] Kou N M, Peng C, Yan X, et al. Multi-agent path planning with non-constant velocity motion [C]//*International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2019.
- [16] Guo T, Yu J. Sub-1.5 time-optimal multi-robot path planning on grids in polynomial time [J]. 2022, arXiv: 2201.08976.
- [17] Shi D, Tong Y, Zhou Z, et al. Adaptive task planning for large-scale robotized warehouses [C]//*International Conference On Data Engineering*, 2022.
- [18] Li W, Chen H, Jin B, et al. Multi-agent path finding with prioritized communication learning [C]//*International Conference on Robotics and Automation*, 2022.
- [19] Shi D, Zhou N, Tong Y, et al. Collision-aware route planning in warehouses made efficient: A strip-based framework [C]//*International Conference on Data Engineering*, 2023.
- [20] Surynek P. Towards optimal cooperative path planning in hard setups through satisfiability solving [C]//*Pacific Rim International Conference on Artificial Intelligence*, 2012.
- [21] Yu J, LaValle S M. Optimal multirobot path planning on graphs: Complete algorithms and effective heuristics [J]. *IEEE Transactions on Robotics*, 2016, **32**(5): 1163-1177.
- [22] Gómez R N, Hernández C, Baier J A. Solving sum-of-costs multi-agent pathfinding with answer-set programming [C]//*Association for the Advancement of Artificial Intelligence*, 2020.
- [23] Luna R, Bekris K E. Push and swap: Fast cooperative path-finding with completeness guarantees [C]//*International Joint Conference on Artificial Intelligence*, 2011.
- [24] De Wilde B, Ter Mors A W, Witteveen C. Push and rotate: Cooperative multiagent path planning [C]//*International Foundation for Autonomous Agents and Multiagent Systems*, 2013.
- [25] Yu J. Average case constant factor time and distance optimal multi-robot path planning in well-connected environments [J]. *Autonomous Robots*, 2020, **44**(3-4): 469-483.
- [26] Standley T. Finding optimal solutions to cooperative pathfinding problems [C]//*Association for the Advancement of Artificial Intelligence*, 2010.
- [27] Wagner G, Choset H. A complete multirobot path planning algorithm with performance bounds [C]//*Institute of Electrical and Electronics Engineers*, 2011.
- [28] Ferner C, Wagner G, Choset H. Optimal multirobot path planning in low dimensional search spaces [C]//*International Conference on Robotics and Automation*, 2013.
- [29] Surynek P. An optimization variant of multi-robot path planning is intractable [C]//*Association for the Advancement of Artificial Intelligence*, 2010.
- [30] Yu J, LaValle S. Structure and intractability of optimal multi-robot path planning on graphs [C]//*Association for the Advancement of Artificial Intelligence*, 2013.
- [31] Sartoretti G, Koenig S, Choset H. A combined learning-and search-based approach to complete multi-agent path finding [C]//*International Joint Conference on Artificial Intelligence*, 2019.
- [32] Paul S, Deshmukh J V. Multi agent path finding using evolutionary game theory [J]. 2022, arXiv: 2212.02010.
- [33] Sartoretti G, Kerr J, Shi Y, et al. Primal: Pathfinding via reinforcement and imitation multi-agent learning [J]. *IEEE Robotics and Automation Letters*, 2019, **4**(3): 2378-2385.

- [34] Reijnen R, Zhang Y, Nuijten W, et al. Combining deep reinforcement learning with search heuristics for solving multi-agent path finding in segment-based layouts [C]//*Symposium Series on Computational Intelligence*, 2020.
- [35] Liu Z, Chen B, Zhou H, et al. Mapper: Multi-agent path planning with evolutionary reinforcement learning in mixed dynamic environments [C]//*Institute of Electrical and Electronics Engineers*, 2020.
- [36] Huang T, Dilkina B, Koenig S. Learning node-selection strategies in bounded suboptimal conflict-based search for multi-agent path finding [C]//*International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2021.
- [37] Skrynnik A, Yakovleva A, Davydov V, et al. Hybrid policy learning for multiagent pathfinding [J]. *IEEE Access*, 2021, **9**: 126034-126047.
- [38] Huang T, Li J, Koenig S, et al. Anytime multi-agent path finding via machine learning-guided large neighborhood search [C]//*Association for the Advancement of Artificial Intelligence*, 2022.
- [39] Abreu N. Efficient deep learning for multi agent pathfinding [C]//*Association for the Advancement of Artificial Intelligence*, 2022.
- [40] Guan H, Gao Y, Zhao M, et al. AB-mapper: Attention and BicNet based multi-agent path planning for dynamic environment [C]//*Institute of Electrical and Electronics Engineers*, 2022.
- [41] Ma H, Tovey C, Sharon G, et al. Multi-agent path finding with payload transfers and the package-exchange robot-routing problem [C]//*Association for the Advancement of Artificial Intelligence*, 2016.
- [42] Felner A, Stern R, Shimony S, et al. Search-based optimal solvers for the multiagent pathfinding problem: Summary and challenges [C]//*Annual Symposium on Combinatorial Search*, 2017.
- [43] Surynek P, Felner A, Stern R, et al. An empirical comparison of the hardness of multi-agent path finding under the makespan and the sum of costs objectives [C]//*Annual Symposium on Combinatorial Search*, 2016.
- [44] Barták R, Zhou N F, Stern R, et al. Modeling and solving the multi-agent pathfinding problem in picat [C]//*International Conference on Tools with Artificial Intelligence*, 2017.
- [45] Ma H, Harabor D, Stuckey P J, et al. Searching with consistent prioritization for multi-agent path finding [C]//*Association for the Advancement of Artificial Intelligence*, 2019.
- [46] Ma H, Wagner G, Felner A, et al. Multi-agent path finding with deadlines [C]//*International Joint Conference on Artificial Intelligence*, 2018.
- [47] Sturtevant N R. Benchmarks for grid-based pathfinding [J]. *IEEE Transactions on Computational Intelligence and AI in Games*, 2012, **4**(2): 144-148.
- [48] Ma H, Koenig S. AI buzzwords explained: Multi-agent path finding [J]. *AI Matters*, 2017, **3**(3): 15-19.
- [49] Cohen L, Greco M, Ma H, et al. Anytime focal search with applications [C]//*International Joint Conference on Artificial Intelligence*, 2018.
- [50] Cohen L, Uras T, Koenig S. Feasibility study: Using highways for bounded-suboptimal multi-agent path finding [C]//*Annual Symposium on Combinatorial Search*, 2015.

-
- [51] Ma H, Hönig W, Kumar T S, et al. Lifelong path planning with kinematic constraints for multi-agent pickup and delivery [C]//*Association for the Advancement of Artificial Intelligence*, 2019.
- [52] Švancara J, Vlk M, Stern R, et al. Online multi-agent pathfinding [C]//*Association for the Advancement of Artificial Intelligence*, 2019.
- [53] Gebser M, Obermeier P, Otto T, et al. Experimenting with robotic intra-logistics domains [J]. *Theory and Practice of Logic Programming*, 2018, **18**(3-4): 502-519.
- [54] Erdem E, Kisa D, Oztok U, et al. A general formal framework for pathfinding problems with multiple agents [C]//*Proceedings of the AAAI Conference on Artificial Intelligence*, 2013, **27**(1): 290-296.
- [55] Surynek P, Felner A, Stern R, et al. Efficient SAT approach to multi-agent path finding under the sum of costs objective [C]//*European Conference on Artificial Intelligence*, 2016.
- [56] Surynek P, Felner A, Stern R, et al. Modifying optimal SAT-based approach to multi-agent path-finding problem to suboptimal variants [C]//*Annual Symposium on Combinatorial Search*, 2017.
- [57] Surynek P. Reduced Time-expansion graphs and goal decomposition for solving cooperative path finding sub-optimally [C]//*International Joint Conference on Artificial Intelligence*, 2015.
- [58] Wang J, Li J, Ma H, et al. A new constraint satisfaction perspective on multiagent path finding: Preliminary results [C]//*International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2019.
- [59] Han S D, Yu J. Integer programming as a general solution methodology for path-based optimization in robotics: Principles, best practices, and applications [C]//*Institute of Electrical and Electronics Engineers*, 2019.
- [60] Sajid Q, Luna R, Bekris K. Multi-agent pathfinding with simultaneous execution of single-agent primitives [C]//*Annual Symposium on Combinatorial Search*, 2012.
- [61] Khorshid M, Holte R, Sturtevant N. A polynomial-time algorithm for nonoptimal multi-agent pathfinding [C]//*Annual Symposium on Combinatorial Search*, 2011.
- [62] Masehian E, Nejad A H. Solvability of multi robot motion planning problems on trees [C]//*Institute of Electrical and Electronics Engineers*, 2009.
- [63] Surynek P. A novel approach to path planning for multiple robots in bi-connected graphs [C]//*International Conference on Robotics and Automation*, 2009.
- [64] Botea A, Bonusi D, Surynek P. Solving multi-agent path finding on strongly biconnected digraphs [J]. *Journal of Artificial Intelligence Research*, 2018, **62**: 273-314.
- [65] Goldenberg M, Felner A, Stern R, et al. Enhanced partial expansion A* [J]. *Journal of Artificial Intelligence Research*, 2014, **50**: 141-187.
- [66] Wagner G, Choset H. Subdimensional expansion for multirobot path planning [J]. *Artificial intelligence*, 2015, **219**: 1-24.
- [67] Silver D. Cooperative pathfinding [C]//*Proceedings of the First Artificial Intelligence and Interactive Digital Entertainment Conference*, 2005.

- [68] Sturtevant N, Buro M. Improving collaborative pathfinding using map abstraction [C]//*Proceedings of the First Artificial Intelligence and Interactive Digital Entertainment Conference*, 2006.
- [69] Wang K H C, Botea A, et al. Fast and memory-efficient multi-agent pathfinding [C]//*The International Conference on Automated Planning and Scheduling*. 2008, **8**: 380-387.
- [70] Bnaya Z, Felner A. Conflict-oriented windowed hierarchical cooperative A* [C]//*International Conference on Robotics and Automation*, 2014.
- [71] Sigurdson D, Bulitko V, Yeoh W, et al. Multi-agent pathfinding with real-time heuristic search [C]//*IEEE Conference on Computational Intelligence and Games*, 2018.
- [72] Okumura K, Machida M, Défago X, et al. Priority inheritance with backtracking for iterative multi-agent path finding [J]. *Artificial Intelligence*, 2022, **310**: 103752.
- [73] Li J, Chen Z, Harabor D, et al. Anytime multi-agent path finding via large neighborhood search [C]//*International Joint Conference on Artificial Intelligence*, 2021.
- [74] Shaw P. Using constraint programming and local search methods to solve vehicle routing problems [C]//*International Conference on Principles and Practice of Constraint Programming*, 1998.
- [75] Li J, Chen Z, Harabor D, et al. MAPF-LNS2: Fast repairing for multi-agent path finding via large neighborhood search [C]//*Association for the Advancement of Artificial Intelligence*, 2022.
- [76] Sharon G, Stern R, Goldenberg M, et al. The increasing cost tree search for optimal multi-agent pathfinding [J]. *Artificial intelligence*, 2013, **195**: 470-495.
- [77] Sharon G, Stern R, Felner A, et al. Conflict-based search for optimal multi-agent pathfinding [J]. *Artificial Intelligence*, 2015, **219**: 40-66.
- [78] Boyarski E, Felner A, Stern R, et al. Icbs: The improved conflict-based search algorithm for multi-agent pathfinding [C]//*Annual Symposium on Combinatorial Search*, 2015.
- [79] Felner A, Li J, Boyarski E, et al. Adding heuristics to conflict-based search for multi-agent path finding [C]//*The International Conference on Automated Planning and Scheduling*, 2018.
- [80] Li J, Felner A, Boyarski E, et al. Improved heuristics for multi-agent path finding with conflict-based search [C]//*International Joint Conference on Artificial Intelligence*, 2019.
- [81] Li J, Harabor D, Stuckey P J, et al. Disjoint splitting for multi-agent path finding with conflict-based search [C]//*The International Conference on Automated Planning and Scheduling*, 2019.
- [82] Boyarski E, Felner A, Harabor D, et al. Iterative-deepening conflict-based search [C]//*International Joint Conference on Artificial Intelligence*, 2021.
- [83] Li J, Harabor D, Stuckey P J, et al. Symmetry-breaking constraints for gridbased multi-agent path finding [C]//*Association for the Advancement of Artificial Intelligence*, 2019.
- [84] Li J, Harabor D, Stuckey P J, et al. Pairwise symmetry reasoning for multi-agent path finding search [J]. *Artificial Intelligence*, 2021, **301**: 103574.
- [85] Zhang H, Li J, Surynek P, et al. Multi-agent path finding with mutex propagation [J]. *Artificial Intelligence*, 2022, **311**: 103766.

-
- [86] Barer M, Sharon G, Stern R, et al. Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem [C]//*Annual Symposium on Combinatorial Search*, 2014.
- [87] Cohen L, Uras T, Kumar T S, et al. Improved solvers for bounded-suboptimal multi-agent path finding [C]//*International Joint Conference on Artificial Intelligence*, 2016.
- [88] Li J, Ruml W, Koenig S. Eecbs: A bounded-suboptimal search for multi-agent path finding [C]//*Association for the Advancement of Artificial Intelligence*, 2021.
- [89] Surynek P. Unifying search-based and compilation-based approaches to multiagent path finding through satisfiability modulo theories [C]//*Annual Symposium on Combinatorial Search*, 2019.
- [90] Gange G, Harabor D, Stuckey P J. Lazy CBS: Implicit conflict-based search using lazy clause generation [C]//*The International Conference on Automated Planning and Scheduling*, 2019.
- [91] Lam E, Le Bodic P, Harabor D, et al. Branch-and-cut-and-price for multi-agent path finding [J]. *Computers and Operations Research*, 2022, **144**: 105809.
- [92] Lam E, Le Bodic P. New valid inequalities in branch-and-cut-and-price for multiagent path finding [C]//*The International Conference on Automated Planning and Scheduling*, 2020.
- [93] Wang K, Botea A. MAPP: A scalable multi-agent path planning algorithm with tractability and completeness guarantees [J]. *Journal of Artificial Intelligence Research*, 2011, **42**: 55-90.
- [94] Ryan M. Constraint-based multi-robot path planning [C]//*International Conference on Robotics and Automation*, 2010.
- [95] Ederer I. Enhancing meta-agent conflict-based search for the multi-agent pathfinding problem with informed merging and heuristics [D]. Republik Österreich: Technische Universität Wien, 2021.
- [96] Langlois S T. Decentralized multiagent metareasoning applications in task allocation and path finding [D]. Maryland: University of Maryland, College Park, 2021.
- [97] Zhang S, Li J, Huang T, et al. Learning a priority ordering for prioritized planning in multi-agent path finding [C]//*Annual Symposium on Combinatorial Search*, 2022.
- [98] Okumura K, Yonetani R, Nishimura M, et al. CTRMs: Learning to construct cooperative timed roadmaps for multi-agent path planning in continuous spaces [C]//*International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2022.
- [99] Ren J, Sathiyarayanan V, Ewing E, et al. MAPFAST: A deep algorithm selector for multi agent path finding using shortest path embeddings [C]//*International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2021.
- [100] Kaduri O, Boyarski E, Stern R. Algorithm selection for optimal multi-agent pathfinding [C]//*The International Conference on Automated Planning and Scheduling*, 2020.
- [101] Sigurdson D, Bulitko V, Koenig S, et al. Automatic algorithm selection in multiagent pathfinding [J]. 2019, arXiv: 1906.03992.
- [102] Ren J, Sathiyarayanan V, Ewing E, et al. Automatic optimal multi-agent path finding algorithm selector [C]//*Association for the Advancement of Artificial Intelligence*, 2021.
- [103] Alkazzi J M, Rizk A, Salomon M, et al. MAPFASTER: A faster and simpler take on multi-agent path finding algorithm selection [C]//*Institute of Electrical and Electronics Engineers*, 2022.

-
- [104] Damani M, Luo Z, Wenzel E, et al. PRIMAL₂: Pathfinding via reinforcement and imitation multi-agent learning-lifelong [J]. *IEEE Robotics and Automation Letters*, 2021, **6**(2): 2666-2673.
- [105] Virmani L, Ren Z, Rathinam S, et al. Subdimensional expansion using attention-based learning for multi-agent path finding [J]. 2021, arXiv: 2109.14695.
- [106] Wang B, Liu Z, Li Q, et al. Mobile robot path planning in dynamic environments through globally guided reinforcement learning [J]. *IEEE Robotics and Automation Letters*, 2020, **5**(4): 6932-6939.
- [107] Chen L, Wang Y, Miao Z, et al. Multi-agent path finding using imitation-reinforcement learning with transformer [C]//*IEEE International Conference on Robotics and Biomimetics*, 2022.
- [108] Wang Y, Xiang B, Huang S, et al. SCRIMP: Scalable communication for reinforcement-and imitation-learning-based multi-agent pathfinding [J]. 2023, arXiv: 2303.00605.
- [109] Bignoli A. Graph attentional neural network in multi-agent pickup and delivery problems [D]. Milano: Politecnico di Milano, 2022.
- [110] Li Q, Gama F, Ribeiro A, et al. Graph neural networks for decentralized multirobot path planning [C]//*Institute of Electrical and Electronics Engineers*, 2020.
- [111] Li Q, Lin W, Liu Z, et al. Message-aware graph attention networks for large-scale multi-robot path planning [J]. *IEEE Robotics and Automation Letters*, 2021, **6**(3): 5533-5540.
- [112] Sinkar M, Izhan M, Nimkar S, et al. Multi-agent path finding using dynamic distributed deep learning model [C]//*The IEEE International Conference on Communication, Information and Computing Technology*, 2021.
- [113] Ma J, Lian D. Attention-cooperated reinforcement learning for multi-agent path planning [C]//*Database Systems for Advanced Applications*, 2022.
- [114] Ma Z, Luo Y, Ma H. Distributed heuristic multi-agent path finding with communication [C]//*International Conference on Robotics and Automation*, 2021.
- [115] Ma Z, Luo Y, Pan J. Learning selective communication for multi-agent path finding [J]. *IEEE Robotics and Automation Letters*, 2021, **7**(2): 1455-1462.
- [116] Ye Z, Li Y, Guo R, et al. Multi-agent pathfinding with communication reinforcement learning and deadlock detection [C]//*International Conference on Intelligent Robotics and Applications*, 2022.
- [117] Xu Y, Li Y, Liu Q, et al. Multi-agent pathfinding with local and global guidance [C]//*International Conference on Networking, Sensing and Control*, 2021.
- [118] Davydov V, Skrynnik A, Yakovlev K, et al. Q-mixing network for multi-agent pathfinding in partially observable grid environments [C]//*The Royal Australian Chemical Institute*, 2021.
- [119] Van Knippenberg M, Holenderski M, Menkovski V. Time-constrained multi-agent path finding in non-lattice graphs with deep reinforcement learning [C]//*Asian Conference on Machine Learning*, 2021.