

DOI: 10.19650/j.cnki.cjsi.J2412507

面向多到一任务交接的多机器人协作路径规划算法*

毛剑琳¹, 贺志刚¹, 张书凡², 李睿祺², 张凯翔²

(1. 昆明理工大学信息工程与自动化学院 昆明 650500; 2. 昆明理工大学机电工程学院 昆明 650500)

摘要:针对传统的多机器人路径规划算法处理任务形式单一、非必要损耗大等问题,本文提出一种多组多到一任务处理方式的协作动态优先级安全间隔路径规划算法(Co-DPSIPP)。首先,该算法以最小化路径总长度为目标,采用模拟退火、扩散搜索等方法确定各组机器人的任务交接点;然后,采用改进的安全间隔路径规划算法为所有机器人进行分段路径规划;进一步针对部分不合理任务交接点会造成区域性拥塞并导致求解失败的问题,设计群组优先级与中间点动态调整规划策略。最后,在4种基准地图上的测试结果显示,相较于协作基于冲突搜索算法(Co-CBS),本文提出的算法在求解成功率上平均可提升73%,在运行时间和路径总长度上平均分别可减少56%和5%。实验结果证明,本文算法为多组多到一任务场景下的多机器人协作路径规划问题提供了更为灵活且扩展性更强的解决方案。

关键词:多机器人;路径规划;安全间隔;任务交接点;群组优先级;中间点

中图分类号: TP242 TH166

文献标识码: A

国家标准学科分类代码: 520.20

Multi-robot collaborative path planning algorithm for many-to-one task handover

Mao Jianlin¹, He Zhigang¹, Zhang Shufan², Li Ruiqi², Zhang Kaixiang²

(1. Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China;

2. Faculty of Mechanical and Electrical Engineering, Kunming University of Science and Technology, Kunming 650500, China)

Abstract: To solve the problems of traditional multi-robot path planning algorithms dealing with a single form of task and large non-essential loss, this article proposes a multi-group many-to-one task processing mode of cooperative dynamic priority safe interval path planning algorithm (Co-DPSIPP). Firstly, the algorithm utilizes the simulated annealing and diffusion search to determine the task handover point of each group of robots with the objective of minimizing the total path length. Then, the improved safe interval path planning algorithm is used to carry out the segmented path planning for all the robots. Furthermore, to deal with the problem that some irrational task handover points may cause regional congestion and lead to solution failure, a cluster prioritization and intermediate point dynamic adjustment planning strategy is designed. Finally, the test results on four benchmark maps show that, compared with the cooperative conflict-based search algorithm (Co-CBS), the proposed algorithm can improve the solution success rate by 73% on average, and reduce the running time and total path length by 56% and 5% on average, respectively. The experimental results show that the proposed algorithm provides a more flexible and scalable solution for the collaborative path planning problem of multi-robot in multi-group many-to-one task scenarios.

Keywords: multi-robot; path planning; safe interval; task handover points; cluster priority; intermediate point

0 引言

多机器人系统具备调度灵活、任务执行成本低和种

类多等特点,可以高效协作完成单机器人难以胜任的繁杂任务^[1]。而多智能体路径查找技术^[2](multi-agent path finding, MAPF)作为多机器人系统应用的一个重要分支,已经有很多学者在各自的研究领域提出了不同类型

的路径规划算法,并被应用到仓储物流^[3-4]、矿井抢险探测^[5]、军事行动^[6]等领域^[7]中。

多机器人路径规划算法主要包括环境已知和环境未知的两类算法,在环境已知的多机器人路径规划方法中比较具有代表性的算法是层级协作 A* 搜索(hierarchical cooperative A*, HCA*)算法^[8]、基于冲突搜索(conflict-based search, CBS)算法^[9]、基于安全间隔进行路径搜索的多机器人路径规划算法。这些智能搜索算法搜索空间随机器人数量的增加呈指数倍递增,如何采用有效的改进策略对节点扩展过程进行优化从而提高搜索效率是该领域的研究者最为关注的问题。而环境未知的多机器人路径规划方法中比较具有代表性的方法有仿生智能搜索^[10]、深度强化学习^[11]等,这些方法需要学习足够的策略以应对未知的环境。本文主要讨论环境已知的多机器人路径规划算法,该类算法一般由规划层和协调层构成。规划层主要是采用 A*、基于安全间隔路径规划(safe interval path planning, SIPP)^[12]等算法进行单机路径规划,而协调层则负责多机器人路径协调、避免冲突。规划层和协调层的规划策略均深刻影响着规划结果,相关学者在这两方面进行了大量的改进和创新。在规划层, Yakovlev 等^[13]提出了任意角度安全间隔路径规划(any-angle safe interval path planning, AA-SIPP)算法,该算法中机器人移动不再局限于几个固定方向,而是可以直接到达任意斜角位置节点而不受角度约束,因此能有效减少路径长度。后来, Yakovlev 等^[14]进一步对 SIPP 算法节点扩展方式进行改进,提出了具有重复状态的加权 SIPP(weight SIPP with duplicate states, Wd-SIPP)算法,该算法在每次节点扩展时保留最优及次优节点,有效提升了算法在复杂环境中的求解成功率。而在协调层, Barer 等^[15]针对 CBS 算法以代价最优方式进行路径搜索导致规划效率低、求解时间长等问题,提出了增强基于冲突搜索(enhanced CBS, ECBS)算法。该算法允许机器人进行次优搜索,从而可以减少冲突树分支、缩减算法运行时间。

随着问题的扩展, Ma 等^[16-17]提出了解决多智能体取货和交付(multi-agent pickup and delivery, MAPD)问题的算法。尽管这些算法中机器人规划包含了取货和交付两个过程,但是每个机器人仍然是独立完成的任务,未考虑通过机器人间的协作完成任务来提升方案质量。Yakovlev 等^[18]关注了系统中可能存在不同类型机器人共同执行任务的路径规划问题,并采用近似估算方法处理机器人之间的冲突。Rathi 等^[19]考虑到基于静态优先级进行规划易导致算法求解失败的问题,提出了根据机器人每一步规划的自由度指数和距离设定规划顺序的动态优先级规划算法,提升了算法的求解能力。Greshler^[20]基于 CBS 算法框架提出了协作基于冲突搜索(cooperative

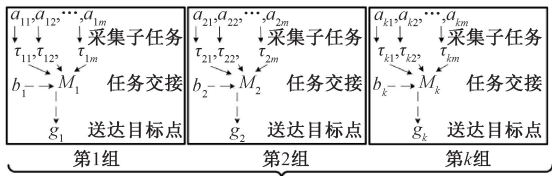
CBS, Co-CBS)算法,实现了单机与单机协作配送的多机器人路径规划。该算法利用改进的多值决策图(multiple-valued decision diagram, MDD)辅助搜索,可获得较满意的任务交接点。但是,该算法仍存在不足,首先,该算法以全局搜索的方式获取协作任务交接点导致计算困难,求解效率不高;其次是 Co-CBS 算法获得的解决方案在路径长度上仍有较大提升空间;最后是该算法目前仅实现单机与单机的协作,协作机器人数量有待提升。

综上,考虑到复杂应用背景下多机器人协作可以完成一些单机无法完成的任务,且能有效降低单机单任务带来的非必要能耗问题。本文在文献[16-20]研究的基础上,搭建了新的解决框架并提出了新的任务交接点搜索策略,提出了协作动态优先级安全间隔路径规划(cooperative dynamic priority safe interval path planning, Co-DPSIPP)算法。在该算法中,本文采用基于优先级的解耦规划框架,依次为各协作组机器人规划路径,以确保算法能够快速返回规划方案。同时,利用模拟退火、扩散搜索等方法来确定各协作组机器人的任务交接点,旨在使所获得解决方案的路径总长度最短。此外,本文采用具有中间点的安全间隔路径规划算法,依次为各协作组机器人进行分段路径规划,以确保机器人之间能够准确、高效进行任务交接。针对不合适的汇集点和优先级设置可能导致路径规划失败的问题,算法还进行了群组优先级和中间点的动态调整,以期获得更高的求解成功率。

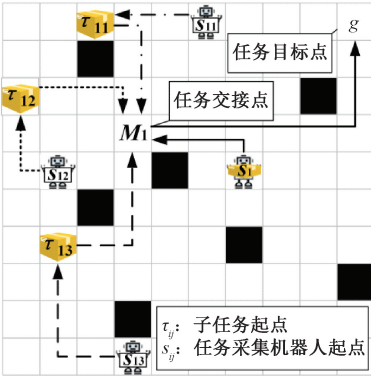
1 问题描述

1.1 Co-MAPF 问题设定

Co-MAPF 问题是指在一个无向图 $G = (V, E)$ 中,给定一个任务集合 $T = \{T_i | 0 < i \leq k\}$ 和机器人群组集合 $R = \{\text{group}_i | 0 < i \leq N_{\text{group}}\}$,由 $m + 1$ 个机器人组成的协作组 group_i ,通过组内协作完成对应分组任务 T_i 中的路径规划任务。其中, $\text{group}_i = \{(a_{i1}, a_{i2}, \dots, a_{im}), b_i\}$, $T_i = \{(\tau_{i1}, \tau_{i2}, \dots, \tau_{im}), g_i\}$; a_{ij} 表示第 i 组内第 j 个子任务采集机器人, b_i 是第 i 组的任务汇集机器人; τ_{ij} 表示第 i 组内第 j 个子任务的起点, g_i 表示第 i 组所有任务的共同目标点。多机器人协作工作方式如图 1 所示,各分组内子任务采集机器人 a_{ij} 和任务汇集机器人 b_i 都有一个自身的起始位置 s_{ij} 和 s_i 。协作组内的子任务采集机器人 a_{ij} 需先从起点 s_{ij} 规划到达任务起点 τ_{ij} 进行货物采集,然后从 τ_{ij} 处规划到达任务交接点 M_i 处,将货物交接给任务汇集机器人 b_i 即可离开去执行新的任务;而任务汇集机器人 b_i 则需先从 s_i 处规划到达 M_i 处与各子任务采集机器人进行货物交接,等待所有货物交接完成后,再从 M_i 点规划到达该组所有子任务的共同目标点 g_i 。



(a) 多机器人组内协作、组间协同的无冲突路径规划

(a) 群组协作机器人路径规划
(a) Path planning for cluster collaborative robots

(b) 单协作组机器人路径规划

(b) Path planning for single collaborative group robots

图1 多机器人协作路径规划

Fig. 1 Diagram of multi-robot collaborative path planning

Co-MAPF 算法规划返回的有效路径方案 $P = \{((p_{a11}, \dots, p_{a1m}), p_{b1}), ((p_{a21}, \dots, p_{a2m}), p_{b2}) \dots ((p_{ak1}, \dots, p_{akm}), p_{bk})\}$, 它由所有单机器人的无冲突路径 p_i 共同构成, 且 $p_i = \{(v_i, t_i) \mid i = 0, 1, 2, \dots, \pi_i\}$ 。其中, v_i 是机器人 i 在时间步 t_i 时机器人所处的栅格位置, p_{aij}, p_{bi} 则表示子任务采集机器人和任务汇集机器人的路径; π_i 是第 i 个机器人从起点到目标点所需的时间代价, 相邻栅格移动时间步代价为 1, 在原地等待则需耗费整数倍的时间步代价。另外, 此问题的基本假设如下:

- 1) 在协作机器人进行任务交接时, 单个协作组 $team_i = \{((a_{i1}, a_{i2}, \dots, a_{im}), b_i), ((\tau_{i1}, \tau_{i2}, \dots, \tau_{im}), g_i)\}$ 只允许存在一个任务交接点 M_i (也称汇集点);
- 2) 各协作组机器人数量、种类一致;
- 3) 协作任务 T_i 已知且固定 (默认上层任务分配系统已完成了最优任务分配), 不可变更。

1.2 Co-MAPF 问题描述

Co-MAPF 作为 MAPF 问题的延伸, 以最小化路径总长度 $length$ 、最大完工时间 $makespan$ 、路径总代价 $flowtime$ 为优化目标^[21-22], Co-MAPF 问题描述如下:

优化目标:

$$\min flowtime = \sum_{i=1}^n \pi_i + \sum_{j=1}^{N_{group}} \pi_j \quad (1)$$

$$\min makespan = \max(\pi_1, \pi_2, \pi_3, \dots, \pi_{n+N_{group}}) \quad (2)$$

$$\min length = \sum_{i=1}^n length_i + \sum_{j=1}^{N_{group}} length_j \quad (3)$$

约束条件:

$$\forall v_l \notin L_{obstacles} \quad (4)$$

$$(v_i = v_{j-1}) \cap (v_j = v_{l-1}) = \emptyset \quad (5)$$

$$v_i \in p_i, v_j \in p_j, i \neq j \quad (6)$$

$$p_i \cap p_j = \emptyset, i, j = 1, 2, \dots, n + N_{group}, i \neq j \quad (6)$$

其中, 式(1)中 $flowtime$ 是所有机器人的完成任务的时间代价之和, 式(2)中 $makespan$ 是所有机器人当中时间代价最大的机器人的完工时间, 式(3)中 $length_i$ 是第 i 个机器人的路径长度, 它与 $flowtime$ 一样以“步”为单位, 相邻两个栅格之间的路径长度记为 1 步, 一般情况下 $length \leq flowtime$ 。在约束条件设定中, 式(4)要求所有路径节点不包含障碍物栅格, 以避免机器人与障碍物碰撞。此外, 式(5)要求两个机器人不能在相邻栅格、相同时间步互换位置, 以防止两个机器人发生边缘冲突。同时, 式(6)要求两个机器人不能在相同时间步占据同一个栅格位置, 以防发生顶点冲突。

2 协作动态优先级安全间隔路径规划算法 (Co-DPSIPP algorithm)

针对上述 Co-MAPF 问题, 本文首先以 Wd-SIPP 算法^[14]作为基础算法来实现多机器人协作路径规划, 并将其命名为协作安全间隔路径规划 (cooperative safe interval path planning, Co-SIPP) 算法。该算法核心包括任务交接点求解、各组协作机器人进行任务交接的协作路径规划两大环节。为提升算法性能, 第 2.3 节融入了群组优先级与中间点动态优化策略, 得到 Co-DPSIPP 算法。

2.1 任务交接点求解策略

本文首先利用点群中位中心原理^[23]求解任务交接点。该原理的定义是: 给定平面上存在某个点 e , 使得该点 e 到其余 N 个点的欧式距离之和 L 最小, 那么称该点为该点群的中位中心。点群中位中心位置 (x_e, y_e) 的表达式为 $L = \min \sum_{i=1}^N \sqrt{(x_e - x_i)^2 + (y_e - y_i)^2}$, 而图 2(a) 为点群中位中心 e 的示意图。

对于点 e , 本文使用模拟退火算法 (simulate anneal arithmetic, SAA) 来求解, 其核心思想是在栅格地图上, 从所有起始点的算数平均中心出发, 逐步迭代直至找到与各点的欧式距离之和的最小解。具体步骤如下:

- 1) 初始化: 设置初始点为算数平均中心点 e_0 , 即 $e_0 = (x_{e0}, y_{e0})$, $x_{e0} = \sum_{i=1}^N x_i / N$, $y_{e0} = \sum_{i=1}^N y_i / N$ 。初始距离之和 $L = \sum_{i=1}^N \sqrt{(x_i - x_{e0})^2 + (y_i - y_{e0})^2}$;
- 2) 通过 $x_{e'} = x'_{e1} / y'_{e1}$, $y_{e'} = x'_{e2} / y'_{e2}$ 获得新参考点 e' , 并重复 3) ~ 6) 步。其中, $x'_{e1} = x_{e1} + \sum_{i=1}^N (x_i / L)$, $y'_{e1} = y_{e1} +$

$\sum_{i=1}^N (1/L); x'_{e2} = x_{e2} + \sum_{i=1}^N (y_i/L), y'_{e2} = y_{e2} + \sum_{i=1}^N (1/L);$
 $x_{e1}, y_{e1}, x_{e2}, y_{e2}$ 均初始化为 0, 而 $x'_{e1}, y'_{e1}, x'_{e2}, y'_{e2}$ 是 $x_{e1}, y_{e1}, x_{e2}, y_{e2}$ 每次迭代后的新值;

3) 产生新解 $L' = \sum_{i=1}^N \sqrt{(x_i - x_{e'})^2 + (y_i - y_{e'})^2};$

4) 计算目标函数 $\Delta L = L' - L;$

5) 以概率 P_e 决定是否接受 $L',$

$$P_e = \begin{cases} 1, & L' < L \\ e^{-(L'-L)/L}, & L' \geq L \end{cases};$$

6) 当 ΔL 小于设定的阈值 0.01, 终止并返回点 $e'.$

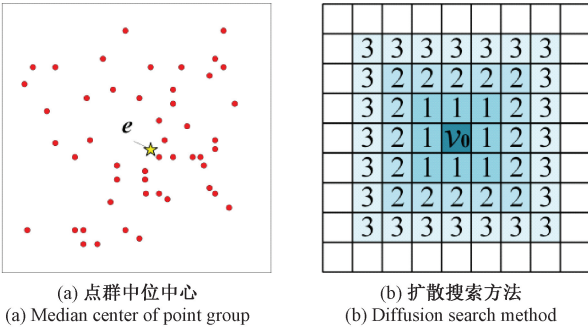


图 2 点群中位中心及扩散搜索方法

Fig. 2 Diagram of the median center in the point group and diffusion search method

根据上述方法, 本文仅需接收每组任务的起始坐标 $(\tau_{i1}, \tau_{i2}, \dots, \tau_{im})$ 、任务汇集机器人的初始位置 s_i 以及任务目标点 g_i 构成的集合作为输入。随后, 通过迭代计算, 就可以获取如图 2(a) 所示的到达集合中各点欧式距离最短的任务交接点 M_0 。

随后, 鉴于机器人 4 邻域(上下左右)移动特性以及地图中潜在障碍物的存在, 可能阻碍机器人直达 M_0 , 因此, 继续利用图 2(b) 所示的扩散搜索方法对 M_0 的周围栅格进行更深入评估。具体而言, 如果协作组机器人规划到达某个邻域栅格(v_i) 的总路径长度相较于规划到达初始中心点 v_0 的总路径长度更短, 则将该邻域栅格 v_i 点选定为新的中心点 v_0' 继续进行搜索。直到协作机器人规划到达邻域栅格(v_i) 的路径长度总和均大于等于到达中心点 v_0' 的路径长度总和时, 将 v_0' 作为路径长度总和最短的任务交接点 M 返回。

模拟退火、扩散搜索两种方法的结合能够有效引导算法迅速锁定路径总长度最小的汇合点 M , 有效避免了对远离 M 的栅格进行冗余搜索, 从而显著缩减了算法的搜索状态空间, 提升算法的求解效率。

2.2 多机器人协作路径规划方法

任务交接点明确之后, 为确保任务交过程在 M 处顺利实现, 各组机器人应该首先规划所有子任务采集机

器人 a_i 的路径, 随后规划任务汇集机器人 b 的路径。采用这种规划方式有以下 2 个主要原因: 1) 在优先级规划框架中, 一个机器人规划成功后其轨迹就已锁定, 在下一个机器人开始规划之前将其规划好的路径信息更新进全局安全间隔列表 (SafeInterval), 就可以避免后续冲突; 2) 任务汇集机器人的最终路径规划需依赖并整合本组内所有子任务采集机器人 a_i 的路径信息, 确保所有子任务在 M 处成功交接后, 方才启动从交接点 M 至最终任务目标点 g 的路径规划。此外, 本文和 Co-CBS 保持一致, 假设任务交接过程需要 1 个时间步来完成。

在具体规划流程中, 严格遵循群组优先级顺序 $\{group_1, group_2, \dots, group_k\}$ 逐一进行。每个协作组内的机器人采用融入中间点的 iSIPP 算法进行有序、高效的规划, 以确保任务交接过程顺畅进行。首先将各个子任务采集机器人 a_i 的初始目标点设定为对应的子任务起点位置 τ_i , 一旦规划至该起点 τ_i , 立即将当前规划机器人的目标点更新为 2.1 节计算得到的对应任务交接点 M , 并继续规划至新的目标点。在此过程中持续更新各子任务采集机器人从机器人起点到达任务交接点的最大时间代价 $cost_{max}$ 。

同时, 任务汇集机器人的初始目标点被直接设定为任务交接点 M , 待其规划至 M 点后, 立即计算当前代价 $cost$ 并与 $cost_{max}$ 比较, 依据比较结果灵活调整后续策略。若满足条件 $cost > cost_{max}$, 则在等待 1 个时间步后, 直接将任务汇集机器人目标点切换至最终任务目标点 g ; 若不满足, 即 $cost \leq cost_{max}$, 则要求任务汇集机器人 b 在 M 处执行 $cost_{max} - cost + 1$ 时间步的等待动作, 随后再调整其目标点为 g 点。最终, 任务汇集机器人从 M 点规划至 g 点, 完成该组的整个协作路径规划任务。

2.3 群组优先级与中间点动态调整策略

Co-SIPP 算法在进行协作路径规划时, 面对任务量增加或地图环境复杂化, 交接点的区域拥塞会加剧群组机器人之间的路径耦合, 进而增加求解失败的风险。无论失败源自子任务采集机器人还是任务汇集机器人, 均会导致整组规划的失败。传统的优先级调整策略在应对规划失败问题时, 常采用确定性重新调度^[24]方法来提升求解成功率。该方法通过优先规划失败的机器人, 确保其优先获得解决方案, 随后再规划其他机器人, 经过有限次调整后, 通常能有效达成规划目标。

受上述方法启发, 为解决 Co-SIPP 算法在复杂环境下求解成功率不高的问题, 本文设计了群组优先级动态调整策略。与以往仅针对单机器人优先级调整的方法不同, 该策略着重于将整个协作组内的机器人优先级集体提升, 同时维持组内机器人原有的优先级顺序。此设计旨在确保任务汇集机器人在协作组中始终处于最低的优先级位置, 以确保其能顺利完成组内所有子任务的汇总

工作。如图3(a)所示,当某一协作组规划遭遇失败时,该组机器人将被整体前置至优先级序列的最前端,优先进行路径规划。此外,为避免无效规划的循环发生,即避免重复利用已知无效的群组优先级组合,本文设计了一个优先级列表 L_{Priority} 来存储并管理这些优先级。在规划前,系统会检查当前生成的优先级是否已存在于列表 L_{Priority} 中,对于已存在的优先级则不再采用,而是将刚调整的协作组在优先级序列中后移1位,形成新的优先级组合,直至找到一个全新的、未尝试过的优先级组合,才继续进行求解尝试。

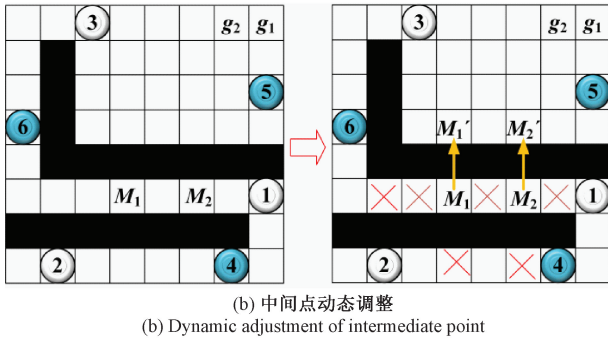
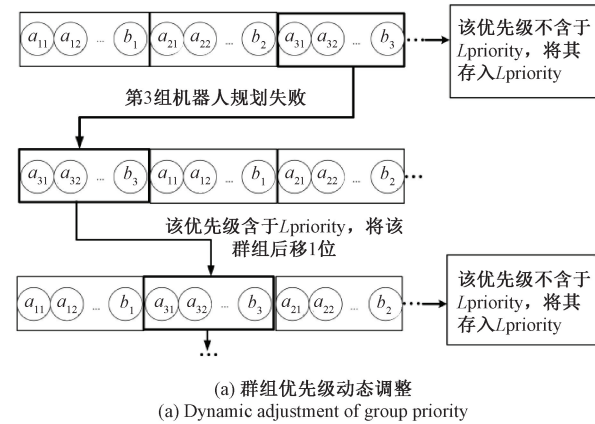


图3 群组优先级、中间点动态调整

Fig. 3 Diagram of dynamic adjustment of group priority and intermediate point

若在规定的时间内,所有可能的优先级组合均被尝试且未能求解成功,算法将终止求解过程,以避免无意义的资源消耗和时间浪费。

在狭窄环境或机器人密集的场景中,若任务交接点的位置设定不当,单纯依赖优先级调整往往难以克服路径冲突与空间限制。鉴于此,本文进一步提出了一种针对中间点位置的动态优化策略。当依据当前设定的优先级及中间点无法得出可行解时,本策略将在调整优先级并重启求解流程的同时,摒弃先前尝试过的中间点,转而探索新的任务交接点。这些新的任务交接点通过图2(b)所示的扩散搜索方法获取,旨在探索其他有效的

解决方案。如图3(b)左图所示,1~3号机器人构成的协作组1与4~6号机器人构成的协作组2因中间点 M_1 和 M_2 初始设定不合理,在分别优先规划时均会遭遇路径阻塞,导致求解失败,此问题无法通过优先级调整方式解决。而图3(b)右图则展示了动态调整中间点的效果:通过扩散搜索,逐步将中间点调整至更合适的位置(如 M_1' 和 M_2'),从而确保两个协作组均能找到无碰撞的可行路径。图中“×”标记了调整过程中被排除的、先前尝试过的无效中间点位置,这些点被记录在一个中间点列表 L_{inter} 中,以便在后续求解中避免重复尝试,加速求解过程。

上述中间点动态调整策略与优先级调整相辅相成,共同构成了 Co-DPSIPP 算法的关键部分,有效提升了算法在复杂场景下的求解成功率与效率。

2.4 Co-DPSIPP 算法伪代码与流程图

Co-DPSIPP 算法的伪代码如算法1、2所示,其中算法2(子模块)为底层单机规划算法。

算法1: Co-DPSIPP 算法

输入: $Team = \{team_i \mid 0 < i \leq N_{\text{group}}\}, Map, T_0$

输出: $Paths, \text{length}$

```

1: while  $t < T_0$  do
2:   for  $team_i$  in  $Team$  do
3:      $M_i \leftarrow$  模拟退火、扩散搜索等方法计算任务交接点 ( $team_i$ )
4:      $a_{ij} \leftarrow \tau_{ij}, b_i \leftarrow M_i$  % 协作机器人的初始目标点设置
5:      $newGoal \leftarrow M_i, g_i$  % 准备各机器人后续进行替换的最终目标点
6:     for  $agent$  in  $team_i$  do
7:        $res, path_i \leftarrow$  底层路径规划 ( $agent, id, Map, newGoal$ )
8:       if  $res = \text{successful}$  then
9:         更新安全间隔  $SafeInterval$  和路径列表  $Paths$ 
10:      end if
11:      if  $res = \text{failed}$  then
12:         $Team \leftarrow newTeam \leftarrow$  群组确定性重排 ( $team_i, Team$ )
13:        初始化安全间隔  $SafeInterval$  和路径列表  $Paths$ 
14:      end if
15:    end for
16:  end for
17:  return  $Paths, \text{length}$  % 求解成功返回路径方案
18: end while
19: return  $\emptyset, 0$  % 到限定时间仍未求解成功停止求解

```

算法 2(子模块): iSIPP 算法

输入: $agent, id, Map, newGoal, Replaced = False$

输出: $res, path$

```

1: while  $curNode \neq agentGoal$  do
2:   使用  $A^*$  扩展节点  $node$ , 将起点放入 OPEN 列表
3:   if  $OPEN = \emptyset$  then
4:     return  $failed, \emptyset$  % 底层求解失败
5:   end if
6:   if not  $Replaced$  then
7:     if  $curNode = agentGoal$  then
8:       if  $agentFinalGoal = g_i$  then % 任务汇集机
          器人
9:         根据具体情况执行对应时间步的等待
          动作
10:      end if
11:       $agentGoal = newGoal_{id}$  % 进行目标点替换
12:       $Replaced = True$  % 目标替换后对应标识
          符置为  $True$ 
13:    end if
14:  end if
15:  if  $Replaced$  then
16:    if  $curNode = agentGoal$  then
17:      return  $successful, path$ 
18:    end if
19:  end if
20: end while

```

图 4 展示了 Co-DPSIPP 算法的流程图,其中边框加粗的 3 个矩形部分是本文的主要研究内容。核心在于“利用模拟退火与扩散搜索等方法求解协作组的任务交接位置”,这一环节对算法的求解效率及规划方案的质量起着决定性作用。通过智能优化等手段,该部分能够快速计算出协作机器人的任务交接位置,避免了对全局地图进行搜索,有效缩小了搜索范围并减少了算法的运行时间。此外,分段路径规划以及群组优先级、中间点的动态调整策略也是该算法的重要构成部分。分段路径规划方法关乎协作机器人能否准确且高效地到达汇合位置进行任务交接;而群组优先级和中间点的动态调整策略则对提升算法的求解能力和鲁棒性具有显著影响。

3 实验分析

为更好且公平地评价算法,本文将多机器人协作任务交接模式分为单机与单机协作(1-to-1)、多机与单机协作(m -to-1)两种。实验 1 首先利用 Wd-SIPP^[14]、

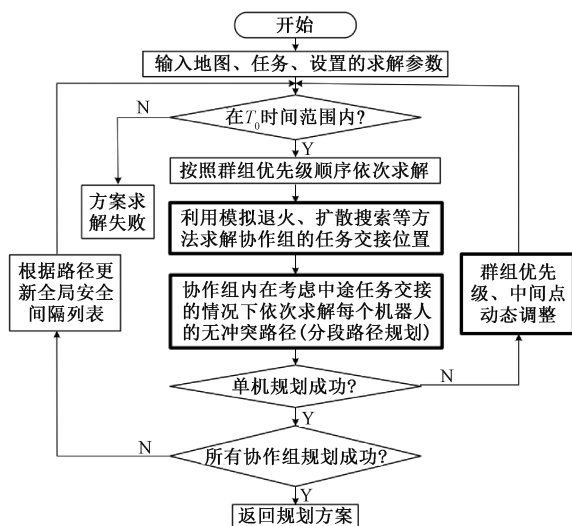


图 4 Co-DPSIPP 算法流程

Fig. 4 Co-DPSIPP algorithm flowchart

Co-CBS^[20]以及 Co-DPSIPP 算法进行“1-to-1”对比实验,旨在凸显本文算法相较于 Co-CBS 及 Wd-SIPP 算法在性能上的独特优势与不足。然后,实验 2 进行了 Wd-SIPP、Co-SIPP 以及 Co-DPSIPP 算法的“ m -to-1”对比实验,不仅分析了本文提出的多机器人协作路径规划算法在求解效率上的优越性,还探讨了引入群组优先级及中间点动态调整策略对算法求解成功率的影响。

本文实验在配备 3.6 GHz 英特尔酷睿 i7 处理器、32 GB 内存的 Windows10 系统电脑上执行,所有算法均以 python 实现。规划时长 T_0 设置为 3 min,若求解器无法在 T_0 时间内返回有效解,则该实例求解被视为失败。实验场景为图 5 所示的 4 种基准地图^[2],分别为巢穴地图、空白地图、迷宫地图及仓储地图,以全面考验算法在不同环境下的适应性。

在协作任务设置中,直接利用基准地图的附带的坐标来定义任务,若任务涉及多个机器人协作,则相应增加坐标组以明确各机器人的起始点与目标点。具体而言,首组坐标定义了首个子任务的起点与任务目标点,随后各组坐标依次表示子任务采集机器人的起点、任务汇集机器人起点以及剩余子任务采集机器人的起点与子任务起点。实验采用递进式设计,每个地图均进行 5 组实验,每组实验的协作任务组数量 N_{group} 逐渐增加。每组实验内又包含 25 个独立实例,每个实例均配置了不同的坐标,以确保实验结果的广泛性与代表性。在性能评估方面,本文聚焦于路径总长度($length$)、最大完工时间($makespan$)、运行时间($runtime$)3 个关键指标,仅对所有算法均能成功求解的算例计算平均值进行比较,以确保比较的公平性。而算法求解成功率 $S_r = N_{successful} / (N_{successful} + N_{fail})$,其中 $N_{successful}$ 是该组测试中求解

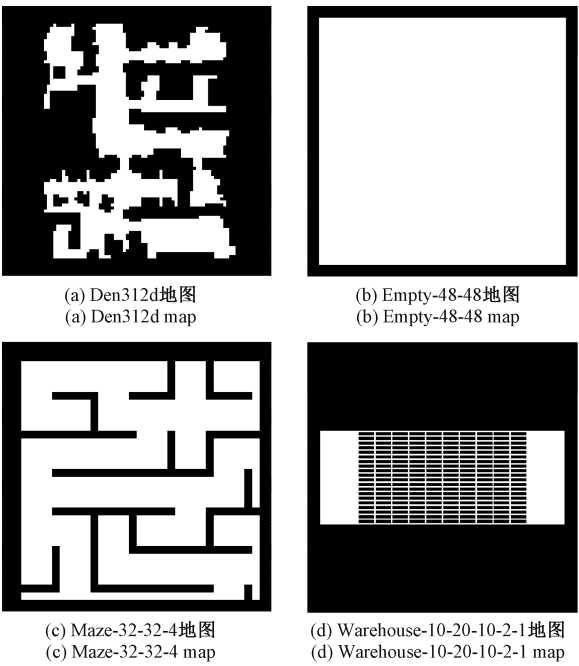


图5 巢穴、空白、迷宫、仓储基准地图
Fig. 5 Nest, empty, maze, warehouse benchmark maps

成功的实例数, N_{fail} 是该组测试中求解失败的实例数, 求

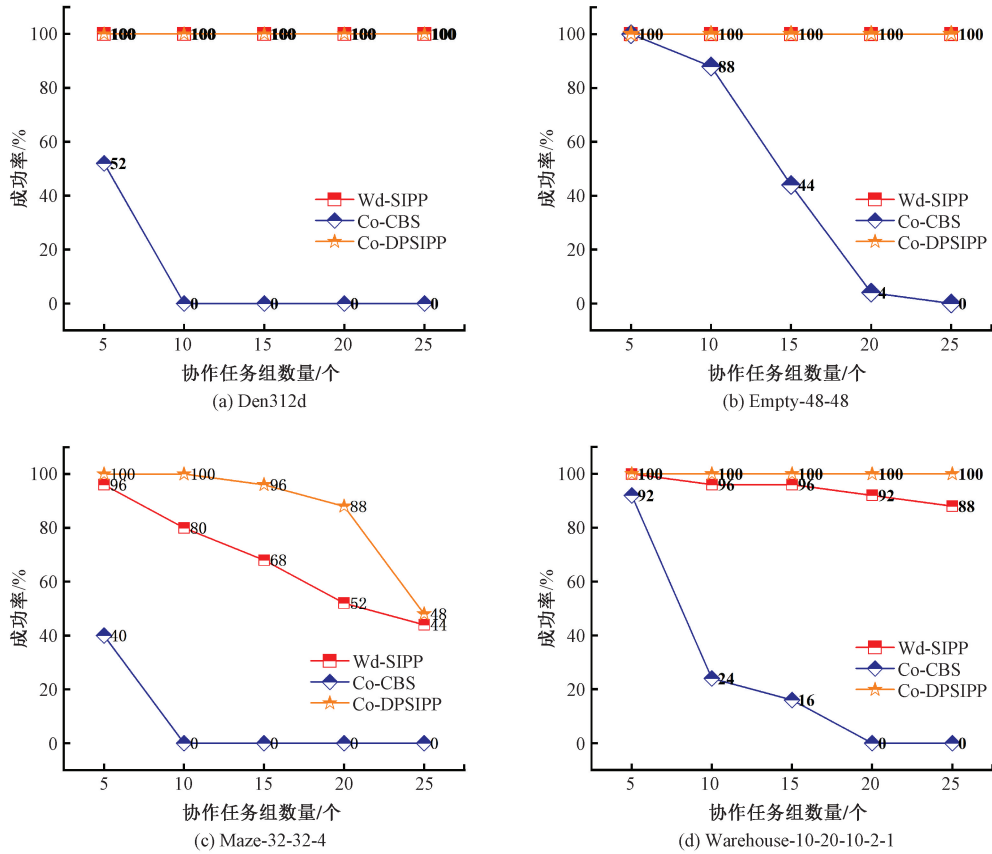


图6 1-to-1协作中3种算法的求解成功率
Fig. 6 Solution success rates of three algorithms in 1-to-1 collaboration

解成功率 S_r 越高则表明算法的求解效率越高、求解能力越强。

3.1 实验 1:1-to-1 协作中几种算法对比

在4种基准地图中,图6及表1展示了Wd-SIPP、Co-CBS与本文提出的Co-DPSIPP算法的测试结果。如图6所示,在迷宫与仓储等复杂狭窄环境中,Co-DPSIPP算法相较于Wd-SIPP算法,在求解成功率上具有一定的优势,在迷宫和仓储地图中分别实现了18.4%和5.6%的增幅。这一优势主要归因于Co-DPSIPP算法通过优化任务分配,使任务汇集机器人能够更有效地协调后续运输任务,从而有效缓解了多机器人之间的拥堵现象,进而提升了整体任务的执行效率与成功率。

进一步对比Co-DPSIPP与Co-CBS算法,图6结果显示在4种基准地图中,Co-DPSIPP算法的求解成功率分别实现了89.6%、52.8%、78.4%、73.6%的显著提升。同时值得注意的是,在可求解最大协作任务组数量的表现上,Co-DPSIPP算法远超过25个任务组,表明该算法能够支持超过50个机器人在上述地图中高效协同作业,同时维持较高求解成功率。相反,Co-CBS算法在巢穴与迷宫地图上的表现严重受限,最大协作任务组数量不足10个,平均求解成功率低于50%,这直接反映了其在处

表1 1-to-1 协作中比较3种算法路径方案质量

Table 1 Quality comparison of three algorithmic path solutions in 1-to-1 collaboration

地图	N_{group}	路径长度/步			最大完工时间/步			算法运行时间/秒		
		Wd-SIPP	Co-CBS	Co-DPSIPP	Wd-SIPP	Co-CBS	Co-DPSIPP	Wd-SIPP	Co-CBS	Co-DPSIPP
den312d	5	737.8	737.5	663.9	147.2	147.8	153.0	0.85	41.43	0.64
	5	460.2	449.1	412.3	89.1	89.2	94.8	0.40	7.04	0.32
	10	940.2	904.6	834.2	100.4	100.3	106.1	0.74	15.06	0.63
	15	1 425.2	1 366.5	1 267.2	111.9	111.9	115.2	1.27	57.25	0.98
	20	1 725.3	1 661.4	1 575.4	99.4	99.0	114.3	1.52	47.75	1.71
maze-32-32-4	5	595.1	562.66	554.7	126.6	126.2	143.2	0.49	42.85	0.37
warehouse-10-20-10-2-1	5	1 134.7	1 116.3	1 041.3	213.2	213.3	224.6	1.36	54.74	0.99
	10	2 336.2	2 198.6	2 105.5	237.5	237.8	247.6	2.74	108.78	1.82
	15	3 334.5	3 154.2	3 128.2	229.7	228.2	249.5	4.27	154.39	3.33

理复杂多机器人协作路径规划问题时在求解效率上的不足。这一局限主要因为 Co-CBS 算法在寻找任务交接点及规划协作路径时计算复杂度过高,导致难以在限定时间内生成有效的解决方案。

图6的测试结果有力证明了在限定时间内,本文提出的 Co-DPSIPP 算法在求解成功率和支撑大规模协作任务组方面均显著优于 Co-CBS 算法,同时在狭窄场景及大量机器人协同作业的场景下,相较于 Wd-SIPP 算法也展现出了独特的优势。

另外,通过表1数据可知,本文提出的算法在平均路径总长度上相较于 Wd-SIPP 算法在上述4个地图中分别可减少10.0%、10.33%、6.8%、8.0%。而相较于 Co-CBS 算法分别可减少10.0%、7.1%、1.4%、4.0%。在实际应用中,更短的路径意味着可以降低能耗,使机器人能够执行更多任务,提升整体作业效率与收益。在求解效率方面,本文算法的平均求解时间均低于 Wd-SIPP 和 Co-CBS 算法,特别是与 Co-CBS 相比,其求解时间仅为后者的1/20或更低,更满足系统对高效性的严苛要求。然而,值得注意的是,Co-DPSIPP 算法在最大完工时间上的表现略逊于 Wd-SIPP 和 Co-CBS 算法,这主要是由于在任务交接过程中任务汇集机器人需要在交接点处等待进行任务交接所致。尽管如此,该算法的分段路径规划设计确保了等待期间不会影响已完成任务交接的机器人继续去执行新的任务,从而维持了系统整体的作业流畅性并降低了整体能耗。此外,Co-CBS 算法与 Wd-SIPP 算法相比,在空白、迷宫、仓储地图中,在路径总长度上分别实现了3.5%、5.4%、4.3%的缩减;在最大完工时间上各有优势,而在求解效率上则不及 Wd-SIPP 算法。

综上所述,本文提出的算法通过创新的任务交接点求解策略,能够快速定位汇集点。相较于全局搜索的 Co-CBS 算法,不仅大幅缩短了求解时间,还保持了较高的求

解成功率与路径优化效果。这一策略有效克服了 Co-CBS 在复杂场景下因搜索空间庞大而导致的求解困难问题,为 Co-MAPF 问题提供了更为高效、实用的解决方案。另外,本文提出的方法相较于 Wd-SIPP 算法,可以缩短路径总长度和算法求解时间,且部分特殊情况下可以提高求解成功率。

3.2 实验2: m -to-1 协作中几种算法对比

在多机与单机协作中,本文对 Wd-SIPP、Co-SIPP 以及 Co-DPSIPP 3种算法进行了对比。需要说明的是,由于设置不同的 m 值实验结果相似,本文仅选取 $m=9$ 呈现实验结果,具体如图7和表2所示。在算法求解工程中,若某实例使用 Co-SIPP 算法能够直接求解,Co-DPSIPP 算法求解时就无需进行优先级和任务交接点的动态调整,这表明 Co-DPSIPP 算法获得的路径方案包含了 Co-SIPP 获得的路径方案。因此,在评估路径的优化效果时,本文主要比较 Wd-SIPP 与 Co-DPSIPP 两种算法。

从表2的数据中可以看出,Co-DPSIPP 算法在路径总长度上表现出了明显优势。与 Wd-SIPP 算法相比,在4个不同的地图环境中,Co-DPSIPP 算法规划的路径总长度分别减少了12.2%、11.5%、7.8%、7.8%,有效减少了机器人的移动距离且优化了多机器人系统的整体效率。这一成果归功于“多到一”协作模式下,任务汇集机器人对于任务采集机器人后续路径的的整合与优化。

在求解时间方面,Co-DPSIPP 算法在空白和仓储环境中相比 Wd-SIPP 表现出优势,但在巢穴和迷宫等复杂地形中则稍显劣势。结合图7结果,可以推断,在巢穴和迷宫这类高复杂度地图中,Co-SIPP 算法本身即面临较高的求解难度,易出现求解失败的情况。因此,Co-DPSIPP 在这些场景下可能需要更频繁地进行优先级和任务交接点的动态调整,以寻找可行解,从而导致求

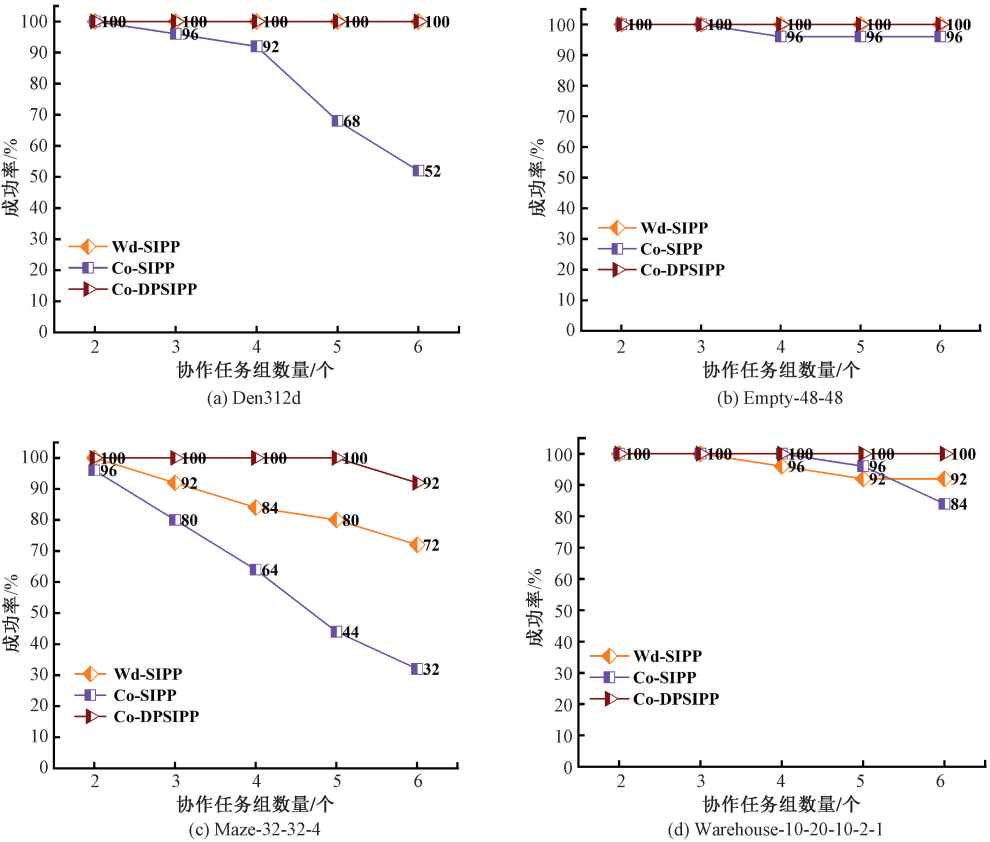


图 7 9-to-1 协作中 3 种算法的求解成功率

Fig. 7 Solution success rates of three algorithms in 9-to-1 collaboration

解时间的增加。而最大完工时间的增加是由于任务汇集机器人需要统一进行本组所有子任务的汇总工作导致。这些结果揭示了算法在不同环境条件下的性能差异,为后续算法优化提供了明确的方向。

3.3 实物测试

为评估 Co-DPSIPP 算法在实物系统上的性能,在 CoCube 机器人系统上进行了 2-to-1 协作路径规划方案的测试。CoCube 机器人如图 8 所示,测试地图如表 3 中给出的过程图所示,深色为障碍物区域,浅色为非障碍物区域。实验中仅设置一组协作任务,由 2 个子任务采集机器人(1 号和 2 号)和 1 个任务汇集机器人(3 号)协作完成。 τ_1 、 τ_2 为子任务起点, m 为任务交接点, G 为 3 个机器人的共同任务目标点。实验中,考虑到实验室配备的 CoCube 机器人的实际情况,设置子任务采集机器人到达任务交接点 m 相邻栅格就可以将其货物交接给任务汇集机器人,然后由任务汇集机器人将其移动到任务目标点 G 。

测试结果如表 3 所示,3 个协作机器人可以按照规划的路径方案顺利执行。图中,从机器人起点开始且带箭头的线条是对应机器人的路径。两个子任务采集机器



图 8 CoCube 机器人

Fig. 8 CoCube robots

人均在 6 时间步进行任务交接,随后,3 号任务汇集机器人单独移动到任务目标点 G 。若采用传统的 MAPF 单机单任务路径规划策略,情况则大为不同。该策略要求所有机器人独立规划直接到达 G 点的路径,此举不仅增加了机器人的整体能耗,还因路径重叠而埋下了冲突隐患。同时,先抵达 G 点的机器人返程容易对其他后续到达 G 点的机器人造成阻碍,并加剧多机器人系统内的碰撞风险。相比之下,本文所提出的算法通过优化路径规划方式,有效避免了上述诸多潜在问题,确保了多机器人协作任务的高效完成。

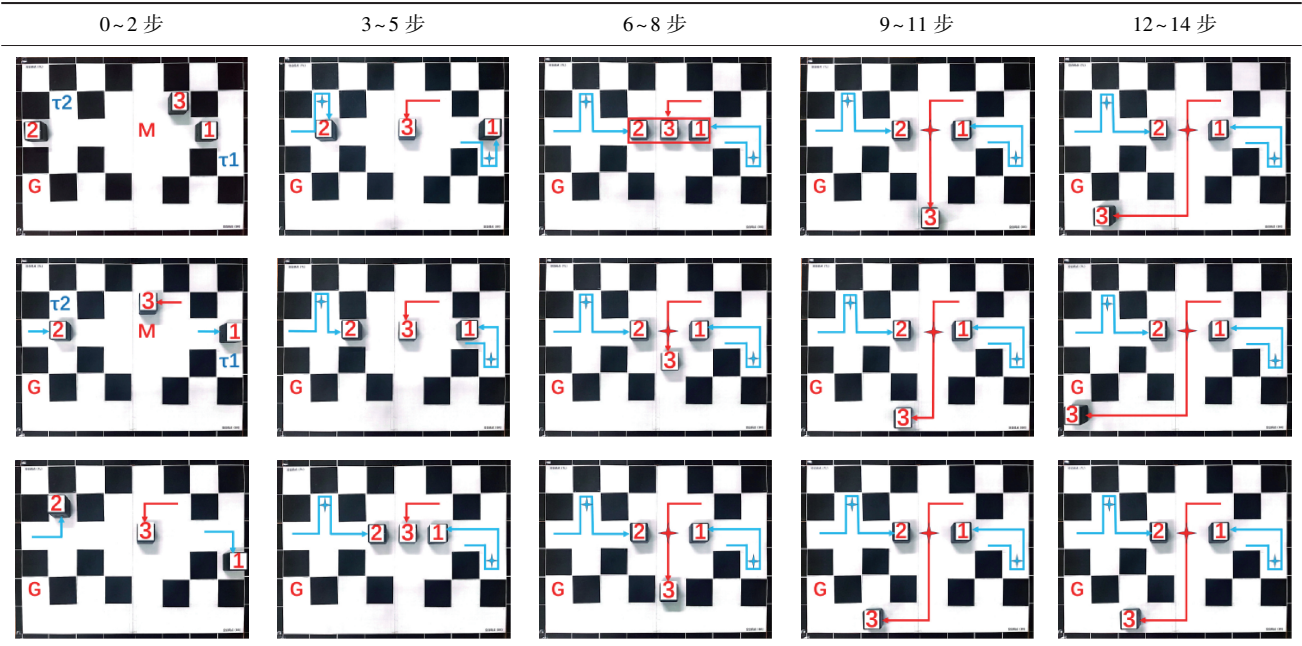
表 2 9-to-1 协作中比较 2 种算法路径方案的质量

Table 2 Comparing the quality of two algorithmic path solutions in a 9-to-1 collaboration

地图	N_{group}	路径长度/步		最大完工时间/步		算法运行时间/s	
		Wd-SIPP	Co-DPSIPP	Wd-SIPP	Co-DPSIPP	Wd-SIPP	Co-DPSIPP
den312d	5	2 032. 2	1 781. 3	182. 4	207. 9	1. 96	1. 42
	10	3 125. 6	2 689. 8	201. 2	221. 2	3. 64	4. 83
	15	4 172. 2	3 669. 8	206. 5	232. 3	5. 83	5. 88
	20	5 189. 3	4 585. 8	207. 3	236. 8	7. 83	15. 97
	25	6 233. 1	5 567. 5	210. 1	246. 7	10. 31	19. 40
empty-48-48	5	1 200. 5	1 046. 1	109. 1	121. 4	0. 85	0. 58
	10	1 814. 6	1 593. 2	113. 5	128. 8	1. 38	0. 92
	15	2 422. 2	2 138. 2	119. 2	134. 9	1. 99	1. 33
	20	3 024. 3	2 704. 5	121. 6	137. 7	3. 04	1. 74
	25	3 631. 8	3 249. 9	124. 5	143. 7	3. 93	2. 24
maze-32-32-4	5	1 568. 6	1 411. 3	171. 5	196. 2	1. 23	3. 33
	10	2 339. 4	2 106. 6	190. 6	206. 8	2. 17	5. 88
	15	3 154. 1	2 851. 1	203. 2	222. 6	3. 42	13. 41
	20	3 905. 3	3 629. 7	208. 9	245. 6	5. 09	28. 63
	25	4 693	4 499. 7	222. 4	263. 2	8. 04	35. 84
warehouse-10-20-10-2-1	5	2 838. 5	2 562. 4	279. 1	315. 1	2. 53	1. 89
	10	4 327. 3	3 936. 8	294. 3	324. 6	4. 64	2. 88
	15	6 062. 9	5 652. 2	310. 4	340. 3	7. 22	4. 32
	20	7 770. 1	7 281. 9	326. 4	352. 5	11. 68	5. 75
	25	9 606. 6	8 878. 7	333. 7	411. 2	14. 09	8. 04

表 3 简单的 2-to-1 协作路径规划中各时间步对应的多机器人位置

Table 3 Multi-robot positions corresponding to each time step in simple 2-to-1 collaborative path planning



4 结 论

传统多机器人路径规划算法常遵循单机配置单任务模式,且预设机器人已位于任务起点,仅需规划起点至目标点的路径。然而,多机协作完成复杂任务的路径规划问题则考虑更为周全,不仅要实现多机器人间的协同作业,还需额外规划机器人从当前位置至任务起点的无冲突路径。本文巧妙的运用模拟退火等方法来快速求解任务交接点,有效减少了地图搜索范围、提升了求解效率。此外,当面临求解器难以正常求解的复杂问题时,本文还创新性地引入了群组优先级调整与中间点动态优化策略,有效提高了算法的求解成功率。

仿真结果表明,本文提出的 Co-DPSIPP 算法相较于 Wd-SIPP 算法,路径总长度最低减少 6.8%,最高达 12.2%。在与先进算法 Co-CBS 的对比中,Co-DPSIPP 求解成功率最高可提升 89.6%,可求解协作组数量更是 Co-CBS 算法的 5 倍以上。此外,Co-DPSIPP 算法在求解时间上相较于 Co-CBS 大幅缩短,且在路径总长度上可实现高达 10% 的优化。这一系列突出表现充分证明了 Co-DPSIPP 算法相较于 Wd-SIPP 和 Co-CBS 算在效率、路径优化及求解能力上的显著优势,能更高效生成切实可行的路径方案。最后,实物测试结果表明,CoCube 机器人可以按照本文算法提供的路径方案顺利执行,验证了该算法获得的解决方案的可行性。

参考文献

- [1] KUDO F, CAI K. A TSP-based online algorithm for multi-task multi-agent pickup and delivery [J]. IEEE Robotics and Automation Letters, 2023, 8(9): 5910-5917.
- [2] STERN R, STURTEVANT N R, FELNER A, et al. Multi-agent pathfinding: Definitions, variants, and benchmarks [C]. Proceedings of the International Symposium on Combinatorial Search, 2019, 10(1): 151-158.
- [3] CHEN X, LI Y J, LIU L T. A coordinated path planning algorithm for multi-robot in intelligent warehouse [C]. International Conference on Robotics and Biomimetics, 2019: 2945-2950.
- [4] ZHANG ZH, CHEN J, ZHAO W B. Multi-AGV route planning in automated warehouse system based on shortest-time Q-learning algorithm [J]. Asian Journal of Control, 2024, 26(2): 683-702.
- [5] DAI L L, WANG H G, PAN Q K. An effective multi-objective evolutionary algorithm based on decomposition for a robot rescue path planning problem [C]. 2022 41st Chinese Control Conference (CCC), 2022: 2065-2070.
- [6] ZHANG J, HE Y, PENG Y N, et al. Cooperative path planning for adversarial target based on neural network and artificial potential field [C]. 2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC), 2018: 1-5.
- [7] DRESNER K, STONE P. A multiagent approach to autonomous intersection management [J]. Journal of Artificial Intelligence Research, 2008, 31: 591-656.
- [8] 张凯翔, 毛剑琳, 向凤红, 等. 基于讨价还价博弈机制的 B-IHCA* 多机器人路径规划算法 [J]. 自动化学报, 2023, 49(7): 1483-1497.
ZHANG K X, MAO J L, XIANG F H, et al. B-IHCA* multi robot path planning algorithm based on bargaining game mechanism [J]. Journal of Automation, 2023, 49(7): 1483-1497.
- [9] SHARON G, STERN R, FELNER A, et al. Conflict-based search for optimal multi-agent pathfinding [J]. Artificial Intelligence, 2015, 219: 40-66.
- [10] 汪繁荣, 杜力, 徐光辉. 基于改进蚁群算法的分布式多机器人协同路径规划 [J]. 中南民族大学学报(自然科学版), 2023, 42(5): 650-657.
WANG F R, DU L, XU G H. Distributed multi robot collaborative path planning based on improved ant colony algorithm [J]. Journal of Central South University for Nationalities (Natural Science Edition), 2023, 42(5): 650-657.
- [11] TAI L, PAOLO G, LIU M. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation [C]. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017: 31-36.
- [12] PHILLIPS M, LIKHACHEV M. Sipp: Safe interval path planning for dynamic environments [C]. 2011 IEEE International Conference on Robotics and Automation, 2011: 5628-5635.
- [13] YAKOVLEV K, ANDREYCHUK A. Any-angle pathfinding for multiple agents based on SIPP algorithm [C]. Proceedings of the International Conference on Automated Planning and Scheduling, 2017, 27: 586-594.
- [14] YAKOVLEV K, ANDREYCHUK A, STERN R. Revisiting bounded-suboptimal safe interval path planning [C]. Proceedings of the International Conference on Automated Planning and Scheduling, 2020, 30: 300-304.
- [15] BARER M, SHARON G, STERN R, et al. Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem [C]. Proceedings of the International Symposium on Combinatorial Search, 2014,

- 5(1): 19-27.
- [16] MA H, HONIG W, KUMAR T K S, et al. Lifelong path planning with kinematic constraints for multi-agent pickup and delivery [C]. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33(1): 7651-7658.
- [17] LIU M H, MA H, LI J Y, et al. Task and path planning for multi-agent pickup and delivery [C]. Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, 2019: 1152-1160.
- [18] YAKOVLEV K, ANDREYCHUK A, VOROBYEV V. Prioritized multiagent path finding for differential drive robots [C]. European Conference on Mobile Robots, 2019: 1-6.
- [19] RATHI A M, ROHITH G, VADALI M. Prioritized path planning of multiple autonomous vehicles in urban environments [C]. Proceedings of the 2021 5th International Conference on Advances in Robotics, 2021: 1-6.
- [20] GRESHLER N, GORDON O, SALZMAN O, et al. Cooperative multi-agent path finding: Beyond path planning and collision avoidance [C]. 2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS), 2021: 20-28.
- [21] 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]. Proceedings of the International Symposium on Combinatorial Search, 2016, 7(1): 145-146.
- [22] MA H, WAGNER G, FELNER A, et al. Multi-agent path finding with deadlines [C]. 27th International Joint Conference on Artificial Intelligence, IJCAI 2018, 2018: 417-423.
- [23] DAS S K, ROY S K, WEBER G W. An exact and a heuristic approach for the transportation-p-facility location problem [J]. Computational Management Science, 2020, 17(3): 389-407.
- [24] ANDREYCHUK A, YAKOVLEV K. Two techniques that enhance the performance of multi-robot prioritized path

planning [C]. Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, 2018: 2177-2179.

作者简介



毛剑琳, 1997 年于昆明理工大学获得学士学位, 2000 年于昆明理工大学获得硕士学位, 2007 年于上海交通大学获得博士学位, 现为昆明理工大学教授, 主要研究方向为通信网络资源分配与协议优化、网络化控制系统研究、智能优化及调度算法等。

E-mail: 1318524654@qq.com

Mao Jianlin received her B.Sc. degree from Kunming University of Science and Technology in 1997, M.Sc. degree from Kunming University of Science and Technology in 2000, and Ph.D. degree from Shanghai Jiaotong University in 2007. She is currently a professor at Kunming University of Science and Technology. Her main research interests include communication network resource allocation and protocol optimization, networked control system research, intelligent optimization and scheduling algorithms, et al.



贺志刚 (通信作者), 2022 年于华东交通大学获得学士学位, 现为昆明理工大学硕士研究生, 主要研究方向为多机器人路径规划。

E-mail: 3224528984@qq.com

He Zhigang (Corresponding author) received his B.Sc. degree from East China Jiaotong University in 2022. He is currently a master student at Kunming University of Science and Technology. His main research interest is multi-robot path planning.



张书凡, 2018 年于浙江大学获得学士学位, 现为昆明理工大学硕博连读的博士研究生, 主要研究方向为多机器人鲁棒路径规划。

E-mail: 20233103006@stu.kust.edu.cn

Zhang Shufan received his B.Sc. degree from Zhejiang University in 2018. He is currently a Ph.D. student at Kunming University of Science and Technology. His main research interest is multi-robot robust path planning.