

控制与决策

Control and Decision

一种基于改进冲突搜索的多机器人路径规划算法

张洪琳, 吴耀华, 胡金昌, 张健

引用本文:

张洪琳, 吴耀华, 胡金昌, 张健. 一种基于改进冲突搜索的多机器人路径规划算法[J]. *控制与决策*, 2023, 38(5): 1327–1335.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2022.0729>

您可能感兴趣的其他文章

Articles you may be interested in

[基于信息物理系统的多分拣移动机器人调度策略](#)

Multi-sorting mobile robot scheduling strategy based on cyber-physical system

控制与决策. 2023, 38(1): 281–288 <https://doi.org/10.13195/j.kzyjc.2021.1343>

[基于改进蚁群和DWA算法的机器人动态路径规划](#)

Robot dynamic path planning based on improved ant colony and DWA algorithm

控制与决策. 2022, 37(9): 2211–2216 <https://doi.org/10.13195/j.kzyjc.2021.1804>

[基于改进生物激励神经网络算法的多移动机器人协同变电站巡检作业](#)

Multi-mobile robot cooperative inspection operation based on improved biological excitation neural network algorithm in substation

控制与决策. 2022, 37(6): 1453–1459 <https://doi.org/10.13195/j.kzyjc.2020.1714>

[一种基于免疫机理的确定性移动机器人路径规划算法](#)

A path planning algorithm of deterministic mobile robot based on immune mechanism

控制与决策. 2021, 36(10): 2418–2426 <https://doi.org/10.13195/j.kzyjc.2020.0059>

[凸优化与A*算法结合的路径避障算法](#)

Convex optimization and A-star algorithm combined path planning and obstacle avoidance algorithm

控制与决策. 2020, 35(12): 2907–2914 <https://doi.org/10.13195/j.kzyjc.2019.0351>

一种基于改进冲突搜索的多机器人路径规划算法

张洪琳¹, 吴耀华^{1†}, 胡金昌², 张健¹

(1. 山东大学 控制科学与工程学院, 济南 250061; 2. 山东师范大学 商学院, 济南 250013)

摘要: 针对智能仓储环境下多载位自主移动机器人集群拣选-配送路径规划问题, 提出一种改进型基于冲突搜索的多智能体路径规划算法. 在模型方面, 采用多载位机器人替代KIVA机器人, 建立以最小化拣选-配送时间以及无效路径比为目标的数学规划模型. 在算法方面, 首先, 提出一种基于优先级规则的多智能体冲突消解加速策略; 然后, 设计基于动态规划的单机机器人拣选序列优化算法; 最后, 设计考虑转向惩罚的增强A*算法搜索机器人最优路径. 实验结果表明: 所提出模型与KIVA系统相比有较大优越性; 所提出算法能够有效缩短拣选-配送时间、减少无效路径时间.

关键词: 多智能体路径规划; 基于冲突的搜索; 多载位机器人; 冲突消解; 最小路径时间; 路径规划

中图分类号: TP18

文献标志码: A

DOI: 10.13195/j.kzyjc.2022.0729

引用格式: 张洪琳, 吴耀华, 胡金昌, 等. 一种基于改进冲突搜索的多机器人路径规划算法[J]. 控制与决策, 2023, 38(5): 1327-1335.

A multi-robot path finding algorithm based on improved conflict search

ZHANG Hong-lin¹, WU Yao-hua^{1†}, HU Jin-chang², ZHANG Jian¹

(1. School of Control Science and Engineering, Shandong University, Jinan 250061, China; 2. Business School, Shandong Normal University, Jinan 250013, China)

Abstract: An improved multi-agent path finding algorithm based on the conflict search algorithm is proposed for the path planning problem of picking-delivery with mobile robot clusters in the smart storage environment. In terms of the model, a mathematical planning model with the objective of minimizing the picking-delivery time and the invalid path ratio is established by using multi-carrier robots instead of KIVA robots. In terms of the algorithm, firstly a priority rule-based multi-intelligent conflict resolution acceleration strategy is proposed. Then a single-robot picking sequence optimization algorithm based on dynamic planning is designed. Finally, an enhanced A* algorithm with the consideration of turning penalty is proposed to search for the optimal robot path. The experimental results show that the proposed model is superior to the KIVA system. The proposed algorithm can effectively shorten the picking-delivery time and reduce the invalid path time.

Keywords: multi-agent path finding; conflict-based search; multi-load robot; conflict resolution; minimum path time; path planning

0 引言

随着近十几年电商平台的快速发展,指数级增长的海量订单对仓储和物流系统的效率提出了越来越高的要求;此外,日益增长的人工成本使得自动化和智能化成为物流系统的发展趋势.在此背景下,无人化、智能化立体仓储成为物流领域的热点.自主移动式仓储机器人作为可完全替代人工的高效工具,成为智能仓储系统的研究热点.在基于多机器人系统的

智能仓储领域,美国亚马逊、日本大福、国内兰剑智能、极智嘉、菜鸟物流、京东物流等知名电商和物流企业正在加速布局.

KIVA机器人是亚马逊子公司Kiva Systems研发的一种载重式自主移动机器人.在KIVA系统中,大量的机器人在接收指令后,自主移动至目标货架,托举货架至拣选台^[1].自亚马逊公司首次将KIVA机器人应用于仓储系统以来,多机器人路径规划(multi-

收稿日期: 2022-05-01; 录用日期: 2022-09-03.

基金项目: 山东省自然科学基金项目(ZR2020MF085); 国家自然科学基金项目(62273204); 山东省自然科学基金青年项目(ZR2022QF109).

责任编辑: 杨涛.

†通讯作者. E-mail: MIKE.WU@263.net.

*本文附带电子附录文件,可登录本刊官网该文“资源附件”区自行下载阅览.

robot path finding, MRPF) 成为领域内的研究热点. MRPF 是多智能体路径规划 (multi-agent path finding, MAPF) 在机器人路径规划领域的一个分支. 文献[2] 将 MAPF 问题定义为: 多个智能体由各自的当前位置出发, 规划路径到达各自的目标点, 同时消除或尽量减少和避免机器人路径间的冲突. 文献[3] 进一步研究和拓展 MAPF 问题, 提出了一种全周期 KIVA 调度和 MAPF 框架, 该框架考虑多机器人系统的动态任务特性, 完善了 MAPF 系统. 此外, 也有学者结合多层穿梭车系统^[4] 与 KIVA 系统, 设计了新型仓储机器人系统^[5].

在 MAPF 算法方面, 已有许多代表性成果. 文献[6] 提出了一种基于启发式的初始路径搜索算法, 并结合模拟退火算法进行路径优化; 文献[7] 提出了一种解决大型停车场路径规划问题的增强启发式 A* 算法; 文献[8] 提出了一种 MS* 精确算法, 该算法能够求解 20 台机器人执行 50 个任务时的最优无冲突路径; 文献[9] 提出了一种鲁棒算法, 用于优化考虑不可预测障碍的 MAPF 问题; 文献[10] 考虑机器人运动学特性, 针对真实场景下机器人运动学约束, 提出了一种 MAPF-POST 算法, 在减少冲突的同时避免路径重规划导致的算法低效问题; 文献[11] 提出了一种融合交通规则与动态地图预约机制的增强 A* 算法, 实现仓储环境下多机器人路径协同规划; 文献[12] 研究以最小化分拣作业周期为目标的大规模 AGV 调度问题, 提出了一种基于通用变邻域搜索的优化算法; 文献[13-14] 提出了一种基于强化学习的 PRIMAL 框架, 该框架可实现 1 024 个智能体路径协同规划, 并通过引入分布式强化学习策略, 进一步提出 PRIMAL 2 框架, 将调度规模扩展至 2 048 个智能体; 文献[15] 提出了一种基于简化地图的区域采样 RRT* 算法以解决传统 RRT* 算法^[16] 搜索效率低的问题, 实现全局路径快速规划; 文献[17-18] 提出了一种基于冲突搜索 (conflict-based search, CBS) 两层算法, 通过高层冲突二叉树搜索以及低层 A* 路径搜索实现 MAPF 问题的优化求解, 并证明 CBS 算法的最优性和完备性. 自首次被提出以来, 由于具备良好的适应性和可行性, CBS 算法及其改进型算法逐渐成为 MAPF 领域内最流行的算法. 文献[19] 提出了一种改进型基于优先级的搜索 (priority-based search, PBS) 算法, 该算法通过在高层搜索中增加优先级约束的方式, 加速算法搜索速度. 文献[20] 提出了一种改进型 PBS 算法, 用于协调和优化多机器人轨迹. 文献[21-22] 提出了 2 种改进型 PBS 算法, 分

别用于优化 tMAPF 问题和 L-MAPF 问题. 但是上述关于 PBS 的文献中均缺少优先级判定的相关规则. 文献[23] 提出了一种以提高算法搜索速度为目标的增强 CBS (enhanced CBS, ECBS) 算法. 在此基础上, 文献[24] 通过融合明确估计方法的 CBS (explicit estimation CBS, EECBS) 算法, 通过在线学习确定高层搜索树的扩展方式. 随后, 为进一步提高 ECBS 成功率, 文献[25] 在 EECBS 的基础上进一步提出了一种嵌套式 CBS (nested CBS, NCBS) 算法. 文献[26] 提出了一种求解无人机系统航迹规划问题的带有绕行助力策略的改进型 CBS (improved CBS with helpful bypass, ICBS-HB). 文献[27] 提出了一种连续时间下的 CBS (continuous-time CBS, CCBS) 算法, 摆脱了 CBS 算法对离散化时间的依赖. 文献[28] 提出了一种融合“高速公路”策略^[29] 的 ECBS 算法, 该算法能够有效缩短 CBS 搜索时间. 此外, 文献[20] 提出了一类基于优先级规划 (prioritized planning, PP) 的 MAPF 算法, 设计并证明了几种确定优先级的规则.

综上, 尽管关于 MAPF 的研究已取得一些成果, 但是仍然存在一定局限性, 主要体现为: 1) 目前 MAPF 研究集中在 KIVA 系统路径优化, 是一种“货架到人”的拣选模式. 关于多载位自主移动机器人 (multi-load auto-mobile robot, MLAMR) 系统“货箱到人”模式的研究极少. 2) 尽管 MLAMR 拣选-配送过程的 MAPF 问题具有较高工程价值和学术意义, 但是鲜有相关研究报道. 3) 已有研究大多假设机器人匀速行驶, 未考虑不同负载状态、不同运动状态的机器人速度变化以及机器人转向因素对系统运行时间的影响. 针对现有研究的不足, 本文提出一种针对仓储环境下 MLAMR 集群拣选-配送 MAPF 问题, 建立一种新型拣选-配送模型, 设计一种改进型 CBS 算法, 并通过仿真实验验证 MLAMR 模型的有效性以及改进型 CBS 算法的优越性.

1 数学模型

1.1 问题描述

智能化立体仓库的拣货流程如图 1 所示: 包含 N 件货物的 M 个订单由 K 台同型号多载位自主移动机器人 (multi-load auto-mobile robot, MLAMR) 完成拣货-配送, 如图 2 所示. 与 KIVA (如图 2(a) 所示) 不同, MLAMR (如图 2(b) 所示) 单次可执行多个货箱的拣货-配送任务, 而 KIVA 执行单个货架的选取和配送任务. 即 MLAMR 执行“货箱到人”的柔性化拣选模式, KIVA 执行“货架到人”的粗放式拣选模式. 在被指派一个或多个任务后, MLAMR 由各自的起始

点出发,规划无冲突路径依次到达目标货物所在的货架,伸缩臂提升至目标层抓取货箱并存放至空闲载位.在完成所有拣货任务后,规划路径到达指定的拣选台,整个流程分为拣货和配送2个子过程.目标是 minimized 拣选完成时间以及无效路径占总路径的比例.定义该问题为多载位自主移动机器人集群拣选-配送问题 (pick and delivery problem with multi-load auto-mobile robot vlusters, PDPMLAMRC). 为方便分析和研究,做出如下假设.

- 1) 所有机器人型号相同,参数相同;
- 2) 机器人在空载和不同负载下运行时速度不同;
- 3) 机器人直行和转向时速度不同;
- 4) 每件货物的拣货过程有且只有1个机器人执行和完成;
- 5) 机器人被指派的货物数量不大于载位数量;
- 6) 不考虑机器人运行时产生的电能消耗;
- 7) 所有通道均为单行道;
- 8) 考虑机器人路径冲突.

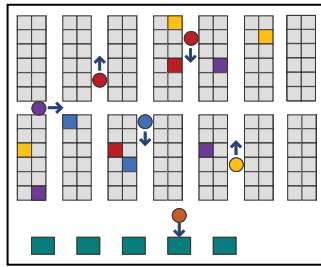


图1 拣选-配送示意图

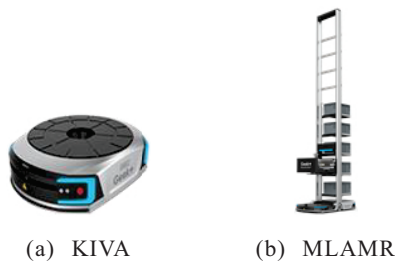


图2 KIVA和MLAMR实物

1.2 符号和变量

变量、集合和决策变量如表1所示.

表1 符号和变量

符号	定义
O	订单集合, $O = O_1, O_2, \dots, O_M$, 共 M 个订单
T	所有订单中的货物集合, $T = T_1, T_2, \dots, T_N$, 共 N 件货物
T^{O_i}	第 i 个订单中包含的货物集合, $T^{O_i} = T_1^{O_i}, T_2^{O_i}, \dots, T_l^{O_i}$, $T_l^{O_i}$ 为第 i 个订单中的第 l 件货物
R	机器人集合, $R = R_1, R_2, \dots, R_K$, 共 K 台机器人
R^w	处于工作状态的机器人数量
L	机器人载位数量

表1 (续)

符号	定义
Q_{R_j}	分配给第 j 个机器人的货物集合, $Q_{R_j} = T_{R_j}^1, T_{R_j}^2, \dots, T_{R_j}^i$
q_{R_j}	Q_{R_j} 包含的货物数量, $q_{R_j} \leq 1$
N	所有货物的坐标集合, $N = N_{T_1}, N_{T_2}, \dots, N_{T_N}$
N_{T_i}	货物 T_i 的坐标, $N_{T_i} = (P_{T_i}, C_{T_i}, H_{T_i})$, $P_{T_i}, C_{T_i}, H_{T_i}$ 分别为排坐标、列坐标、层坐标
G	所有拣选台的集合, $G = G_1, G_2, \dots, G_i, \dots, G_F$, 共 F 个拣选台
N_{G_i}	拣选台 G_i 的坐标, $N_{G_i} = (P_{G_i}, C_{G_i})$, P_{G_i}, C_{G_i} 分别为排坐标、列坐标
NG	所有拣选台的坐标集合, $NG = N_{G_1}, N_{G_2}, \dots, N_{G_i}, \dots, N_{G_F}$
$N_{R_i}^t$	机器人 R_i 在 t 时刻的位置坐标, $N_{R_i}^t = (P_{R_i}^t, C_{R_i}^t)$
v_0^s	机器人空载直行速度
v_0^t	机器人空载转向速度
v_l^s	机器人负载为 l 时的直行速度
v_l^t	机器人负载为 l 时的转向速度
$p_{Q_{R_j}}$	机器人 j 完成所有货物拣货所行驶的路径长度
$d_{Q_{R_j}}$	机器人 j 配送所行驶的路径长度
P_{R^w}	所有机器人拣选过程的总路径长度
D_{R^w}	所有机器人配送过程的总路径长度
W_{R^w}	所有机器人拣选-配送过程的总路径长度
a	机器人转向惩罚因子
b	机器人等待惩罚因子
α	0-1 变量, 当机器人处于任务执行状态时为1, 空闲时为0
β	0-1 变量, 当机器人移动时为1, 原地等待时为0
δ	0-1 变量, 当机器人原地等待时为1, 移动时为0
γ	0-1 变量, 当机器人直行时为1, 否则为0
θ	0-1 变量, 当机器人转向时为1, 否则为0
$X_{R_j}^{T_i}$	0-1 变量, 当 R_j 执行 T_i 的拣选-配送时为1, 否则为0
$x_{R_j}^{T_i}$	0-1 变量, 当 R_j 行驶到 N_{T_i} 节点坐标位置时为1, 否则为0
$Y_{R_j}^{G_i}$	0-1 变量, 当 R_j 的配送目标拣选台为 G_i 时为1, 否则为0
$y_{R_j}^{N_{G_i}}$	0-1 变量, 当 R_j 行驶到 N_{G_i} 拣选台坐标时为1, 否则为0
$z_{N_{R_j}^t}^{N_{R_i}^t}$	0-1 变量, 当 R_i 和 R_j 在同一时刻占用同一坐标时为1, 否则为0

1.3 数学规划模型

构建如下数学规划模型:

$$\min F = C_1 + C_2. \tag{1}$$

$$\text{s.t. } R^w \leq K; \tag{2}$$

$$q_{R_j} \leq L; \tag{3}$$

$$\sum_{R_j \in R} \sum_{T_i \in T} X_{R_j}^{T_i} = 1; \tag{4}$$

$$\sum_{R_j \in R} \sum_{T_i \in T} x_{R_j}^{T_i} = 1; \tag{5}$$

$$\sum_{R_j \in R} \sum_{G_i \in G} Y_{R_j}^{G_i} = 1; \tag{6}$$

$$\sum_{R_j \in R} \sum_{N_{G_i} \in NG} y_{R_j}^{N_{G_i}} = 1; \quad (7)$$

$$\sum_{R_i, R_j \in R} z_{N_{R_j}^t}^{N_{R_i}^t} = 0; \quad (8)$$

$$N_{R_j}^t \cap N_{R_i}^t = \emptyset; \quad (9)$$

$$p_{Q_{R_j}} \geq \sum v_0^s \Delta t + \sum v_0^t \Delta t + \sum v_l^s \Delta t + \sum v_l^t \Delta t + \sum w_p \Delta t; \quad (10)$$

$$\sum_{j=1}^{R^w} p_{Q_{R_j}} \leq P_{R^w}; \quad (11)$$

$$d_{Q_{R_j}} \geq \sum v_l^s \Delta t + \sum v_l^t \Delta t + \sum w_d \Delta t; \quad (12)$$

$$\sum_{j=1}^{R^w} d_{Q_{R_j}} \leq D_{R^w}; \quad (13)$$

$$P_{R^w} + D_{R^w} \leq W_{R^w}; \quad (14)$$

$$W_{R^w} \geq \alpha \beta \left(\gamma \sum v^s \Delta t + \theta \sum v^t \Delta t \right); \quad (15)$$

$$\alpha \in \{0, 1\}, \beta \in \{0, 1\}, \delta \in \{0, 1\}, \gamma \in \{0, 1\},$$

$$\theta \in \{0, 1\}, X_{R_j}^{T_i} \in \{0, 1\}, x_{R_j}^{T_i} \in \{0, 1\},$$

$$Y_{R_j}^{G_i} \in \{0, 1\}, y_{R_j}^{G_i} \in \{0, 1\}, z_{N_{R_j}^t}^{N_{R_i}^t} \in \{0, 1\}. \quad (16)$$

其中

$$C_1 = t_s + t_t + t_w, \quad (17)$$

$$C_2 = \frac{at_t + bt_w}{t_s + t_t}, \quad (18)$$

$$t_s = \alpha \beta \sum \gamma, \quad (19)$$

$$t_t = \alpha \beta \sum \theta, \quad (20)$$

$$t_w = \alpha \beta \sum \delta \geq \sum w_p \Delta t + \sum w_d \Delta t. \quad (21)$$

式(1)为最小化拣选-配送时间以及低效路径时间与有效路径时间比之和;式(2)为处于工作状态的机器人数量不能超过机器人总数;式(3)为机器人载重约束;式(4)为1件货物有且只有1个机器人执行拣选-配送任务;式(5)为判断机器人是否到达目标货物所在位置;式(6)为1个机器人对应唯一的目标拣选台;式(7)为判定机器人是否到达目标拣选台;式(8)为同一时刻不允许2个机器人出现在同一坐标;式(9)为机器人的时空冲突约束;式(10)为机器人 j 的拣选路径长度不小于空载直行路径、空载转向路径、负载直行路径、负载转向路径与等待成本和;式(11)为所有机器人拣货总路径不小于每个机器人的拣货路径和;式(12)为机器人 j 的配送路径长度不小于负载直行、负载转向与等待成本和;式(13)为所有机器人配送总路径不小于每个机器人的配送路径和;式(14)为拣选-配送总路径不小于拣选路径与配送路径和;

式(15)为拣选-配送总路径不小于直行路径与转向路径和;式(16)为决策变量的取值约束;式(17)为拣选-配送时间等于直行时间、转向时间与等待时间和;式(18)为低效路径时间与有效路径时间比,其中 a 、 b 分别为转向时间和等待时间的惩罚因子;式(19)~(21)分别为直行时间、转向时间和等待时间。

2 改进型基于冲突搜索的MAPF算法

基于冲突的搜索(conflict-based search, CBS)是一种2层算法,分为高层搜索和底层搜索.在高层,首先分别规划每个机器人的最短路径;然后建立二叉冲突树(conflict tree, CT)记录所有机器人路径上的时空冲突;最后对二叉树上产生冲突的节点对应的2个或多个机器人施加约束,解决节点冲突.在低层,调用A*、D*等路径规划算法,在CT约束下重新规划单机器人路径.就仓储系统而言,CBS是一种解决KIVA系统MAPF问题的高效算法.由于所提出模型中多载位机器人系统与KIVA系统有较大区别,为了更好地求解PDPMLAMRC问题,在文献[17]的基础上进行针对性改进.改进思路为:1)针对CBS高层搜索易陷入维数爆炸的不足,设计结合优先级规则的冲突消除策略,提升算法效率;2)根据多载位机器人的多目的地特性,从TSP角度出发,设计增强A*算法优化单机器人路径;3)在低层搜索上,引入转向惩罚机制,引导机器人减少因转向次数过多导致的路径时间增长.

2.1 CBS算法

CBS算法的冲突消解以及路径规划流程如图3所示.1号、2号2个机器人由各自起点出发,到达对应目标点.在该算例中,CBS算法步骤如下.

1)调用低层算法,规划单机器人最优路径:1号机器人一条最优路径 $P_1 = \{B_1, B_2, B_3, C_3\}$,2号机器人的一条最优路径 $P_2 = \{A_2, B_2, C_2\}$.

2)搜索时空冲突点,分别对1号和2号添加禁行约束.对1号添加禁行约束后,调用低层算法更新路径 $P'_1 = \{B_1, C_1, C_2, C_3\}$,2号机器人路径仍然为 $P_2 = \{A_2, B_2, C_2\}$;对2号机器人添加禁行约束后,更新路径 $P'_2 = \{A_2, A_1, B_1, C_1, C_2\}$.

3)在 (P'_1, P_2) 和 (P_1, P'_2) 两组路径中继续搜索冲突,重复添加禁行约束操作,直至路径无时空冲突.

4)得到3组无冲突路径: $(P_1, P_2) = (\{B_1, B_2, B_3, C_3\}, \{A_2, A_1, B_1, C_1, C_2\})$, $(P_1, P_2) = (\{B_1, B_1, B_2, B_3, C_3\}, \{A_2, B_2, C_2\})$, $(P_1, P_2) = (\{B_1, B_2, B_3, C_3\}, \{A_2, A_2, B_2, C_2\})$.选择路径长度最短的组合,即 $(P_1, P_2) = (\{B_1, B_1, B_2, B_3, C_3\}, \{A_2, B_2, C_2\})$ 或

$$(P_1, P_2) = (\{B_1, B_2, B_3, C_3\}, \{A_2, A_2, B_2, C_2\}).$$

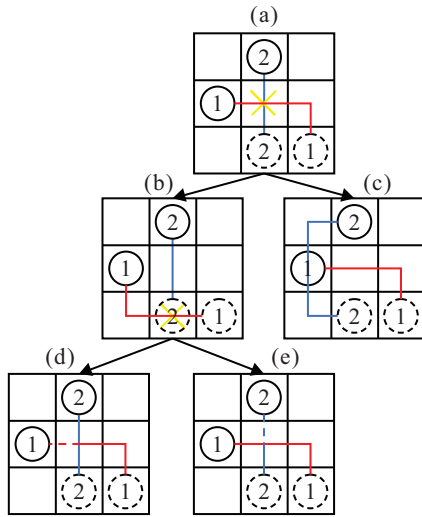


图3 CBS算法搜索步骤

2.2 改进型CBS算法

2.2.1 基于优先级规则的冲突消解策略

CBS执行高层搜索时,通过在冲突处分别给冲突双方施加禁行约束,生成子节点的方式消解冲突.尽管这保证了算法的完备性,但是随着状态空间的增大,这种穷举的方法效率较低.基于此,本节提出一种基于优先级规则的冲突消解策略(prioritized conflict-free strategy, PCS).PCS的原理是为每个机器人设置优先级,在冲突产生时,按照优先级降序进行冲突消解,即高优先级机器人路径不变,低优先级机器人等待或重新规划路径.

图4为PCS的一个算例.在该例中,1号、2号机器人优先级分别为1、2.初始路径分别为: $P_1 = \{B_1, B_2, B_3, A_3\}$, $P_2 = \{C_2, B_2, A_2\}$. P_1 、 P_2 在 B_2 处发生冲突.根据PCS规则,1号避让2号,得到无冲突路径 $P'_1 = \{B_1, B_1, B_2, B_3, A_3\}$, $P'_2 = \{C_2, B_2, A_2\}$.

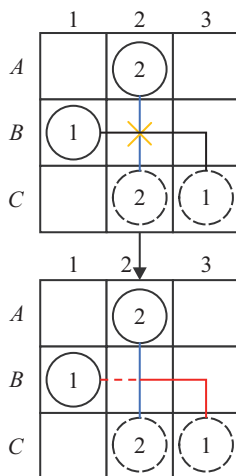


图4 PCS策略算例

在现有基于优先级规划的MAPF研究中,优先级的设置规则均采用随机赋值方法^[19-22].对于以单个

货架为目的地的KIVA机器人而言,随机权重是高效的优先级确定方式;但是在MLAMR仓储环境下,每台MLAMR单次执行多个货箱的拣选任务,随机赋值方法已不适合用.根据PDPMLAMRC的特点,本文定义了一种MLAMR优先级规则.

定义1 基于拣选-配送曼哈顿距离降序排列的优先级规则(priority rule in descending order based on Manhattan distance of pick-delivery, PRDCBMDP-D).

由文献[30]可知,曼哈顿距离公式为

$$c = |x_1 - x_2| + |y_1 - y_2|, \quad (22)$$

其中 (x_1, y_1) 、 (x_2, y_2) 分别为起点和终点坐标.由于多载位机器人有多个目的地,任务数量为 H 的机器人的拣货路径以及配送路径的曼哈顿距离为

$$C = \sum_{i=1}^H c. \quad (23)$$

特别地,当发生冲突的机器人路径曼哈顿距离相等时,按照任务数量降序确定优先级;曼哈顿距离以及任务数量均相等时,判定机器人优先级相同并随机选择避让方和优先方.

2.2.2 基于动态规划的MLAMR拣货顺序优化

单台MLAMR的拣货-配送过程如图5所示.机器人 R 由起点出发,拣选1、2、3号货箱并配送至 P_4 拣选台.这一过程符合指定起点和终点的TSP问题的特征.多载位机器人的载位数量有限,单次可执行的任务数有限,即TSP途径的节点数量有限,因此解空间较小,可采用动态规划算法进行拣选顺序的快速最优化排序.

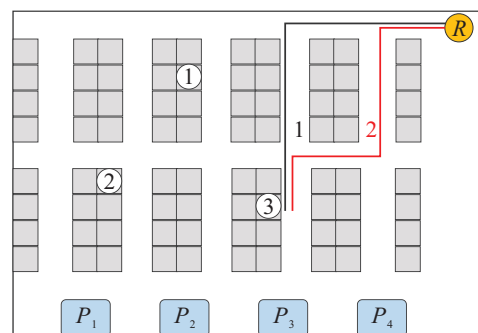


图5 单台MLAMR拣选-配送示意图

首先计算起点与各目标货箱、目标拣选台的曼哈顿距离;然后采用动态规划的逆序解法确定拣选顺序,递推公式为

$$\begin{cases} f_j(x_j) = \min_{u_j \in D_j} \{d(x_j, x_j + 1) + f_{j+1}(x_j + 1)\}, \\ f_G(x_G) = 0, G \in \{j, j-1, \dots, 1\}. \end{cases} \quad (24)$$

其中: $x_j + 1 = u_j(x_j)$, 即从当前状态 x_j 出发,在所

有可能的决策集合 D_j 中选择决策 u_j , 到达下一状态 x_{j+1} ; $f_G(x_G)=0$ 为边界条件, 表示拣选台作为终点.

2.2.3 基于改进型A*低层搜索算法

A*算法是最流行的CBS低层搜索算法. A*算法的路径搜索方式为: 由起始点出发, 机器人每次选择下一步所有可行节点, 估算所有可行节点的路径时间代价, 选择路径时间代价最小的节点进行移动; 机器人重复上述步骤, 直到移动至目标点. 在KIVA环境下, A*算法可高效地优化路径. 但是考虑多载位机器人分段和多目标的运行特征, 常规A*算法已不适用. 基于此, 本节设计适用于多载位机器人路径优化的改进型A*算法.

在第2.2.2节的单MLAMR拣货顺序约束下, 单机路径规划变为分段式规划. 假设拣货顺序为1-2-3, 此时整个拣选-配送过程的路径被分为4段: $R-1, 1-2, 2-3, 3-P_4$. 对于每一分段, 均采用A*算法进行路径规划, 假设机器人由前一节点 $n-1$ 移动至当前节点 n , 该段路径时间代价为

$$f(n) = g(n) + h(n), \quad (25)$$

$$g(n) = \frac{d_{1 \rightarrow n}}{v_{R_i}}, \quad (26)$$

$$h(n) = \frac{d_{n-G}}{v_{R_j}}. \quad (27)$$

其中: $g(n)$ 为机器人由起始点至当前节点 n 的路径时间, $h(n)$ 为由当前节点 n 至目标点 G 的估计路径时间.

在实际生产场景中, 还需进一步考虑机器人转向对路径时间的影响. 机器人在转向时需要经历减速、角度调整、加速3个过程, 与直行相比, 这一过程会消耗相对长的时间, 导致系统效率下降. 如图5所示, 机器人 R 由起点出发, 执行3号货箱的拣货任务. 图5中1、2所示为2条可行路径. 尽管1、2号路径长度相等, 但是2号路径转向次数过多, 导致路径长度相同的情况下需要比1号路径花费更多时间. 基于此, 在路径时间代价公式中添加转向惩罚因子, 促使机器人选择转向次数更少的路径, 改进路径代价公式为

$$f(n) = g(n) + h(n) + \sum R^t, \quad (28)$$

其中 $\sum R^t$ 为机器人转向次数.

2.2.4 改进型基于冲突搜索的MAPF算法

综上, 改进型PCBS算法的流程如图6所示, 其步骤如下.

step 1: 确定每个机器人的优先级, 采用动态规划算法确定单机器人拣货顺序;

step 2: 采用改进型A*算法搜索单机器人最优路

径;

step 3: 生成多机器人路径冲突树;

step 4: 基于优先级的冲突消解;

step 5: 得到无冲突路径.

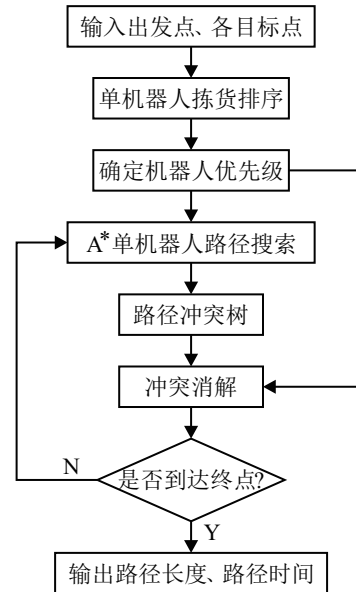


图6 PCBS算法流程

2.2.5 算法复杂度分析

改进型PCBS算法经过单机拣货排序、优先级确定、A*路径搜索和冲突检测、冲突消除4个步骤. 在订单任务数量为 t , 机器人数量为 n , 单台机器人平均任务数为 $m = t/n$ 的情况下, 单机拣货排序的计算复杂度为 $O(m^2)$; 优先级确定的计算复杂度为 $O(mn + n^2)$; A*路径搜索和冲突检测的计算复杂度为 $O(nm^2 \log m)$; 冲突消除的计算复杂度为 $O(n^2m^2 \log m)$. 由于 $m = t/n$, 令 $mn = t$, 各阶段计算复杂度分别为 $O(m^2)$ 、 $O(t + n^2)$ 、 $O(tm \log m)$ 、 $O(t^2 \log m)$. 同时, 因单台机器人载位数量通常为3~5个, m 对于算法复杂度无影响. PCBS算法的计算复杂度为 $O(t^2 + n^2)$.

3 实验分析

3.1 实验设置

本节使用Python语言编写和实现. 在配置为Intel i7-11700k CPU @ 3.6 GHz, NVIDIA GeForce RTX 2060 Graphics Board @ 14000 MHz, 16 GB 内存的电脑上运行. 实验以山东某物流科技公司代运营的某电商平台华南大仓部分时段订单数据为任务, 对算法进行测试. 为验证MLAMR模型的有效性以及所提出PCBS算法的性能, 设计如下2组实验.

1) 相同负载能力下KIVA与MLAMR系统性能比较实验, 测试在地图规模为70_70、80_80、90_90、100_100环境下, 分别执行300、400、500、600个拣选

任务的系统时间和路径长度. 根据MLAMR载位数量不同,在载位数分别为3、4以及5的情况下,设置MLAMR与KIVA数量比为10:30、10:40、10:50进行实验. 2种系统均使用PCBS算法进行实验.

2) MLAMR系统下PCBS算法与CBS算法^[17]和EECBS算法^[24]的MAPF性能对比实验. 测试在地图规模为200_200, MLAMR数量为20、40、60、80、100, 单车载位数量为5, 任务数量为600、700、800、900、1000环境下进行. 在算法对比实验中,式(16)中惩罚因子设置为 $a=1, b=2$.

表2 KIVA与MLAMR对比实验

地图_任务规模	70_70_300		80_80_400		90_90_500		100_100_600		
	<i>l</i>	<i>t</i>	<i>l</i>	<i>t</i>	<i>l</i>	<i>t</i>	<i>l</i>	<i>t</i>	
载位数量	10_3	4283	490	6232	627	14020	910	21154	1257
	10_4	3876	410	5990	551	11894	786	18023	1074
	10_5	3192	390	5346	508	10310	710	16102	916
KIVA数量	30	7630	485	11972	761	24956	1507	63145	2990
	40	7804	374	10183	529	19653	869	47804	1691
	50	7765	361	13646	504	20718	832	39610	1458

表2中:*l*、*t*分别为机器人拣选-配送路径长度和路径时间. 对实验数据进行分析得到以下结论: 1) 在所有地图_任务规模以及不同载位数量下, MLAMR拣选-配送的路径长度均小于KIVA系统; 2) 在70_70_300、80_80_400规模下, MLAMR与KIVA运行时间基本持平; 3) 在90_90_500、100_100_600规模下, MLAMR在路径长度和运行时间上均明显优于KIVA; 4) 随着问题规模的增大, MLAMR的优势愈发明显; 5) MLAMR“货箱到人”模式有效, 且在中大规模地图和任务环境下的实用性较强.

3.2.2 PCBS算法性能实验

在每组任务_机器人规模下均重复实验20次, 实验结果如表3所示. 对实验数据进行分析得出如下结果: 1) 在所有规模实验中, CBS算法表现最差, 仅在任务数为900的实验中路径时间少于EECBS. 2) 任务数为600、700、800时, EECBS与所提出PCBS算法在路径时间上基本持平, 任务数为900、1000时, 所提出PCBS路径时间与EECBS相比有明显减少, 表明所提出PCBS算法在大规模问题求解中具有较大优势. 3) 在所有规模实验下, PCBS算法所得无效路径比均最小, 通过对式(16)分析可知, 所提出算法对路径中的转向次数减少和等待时间缩短有效. 4) 随着任务规模增大, EECBS与CBS的路径时间差距逐渐缩小, 在900_60、900_80、900_100规模下CBS路径时间均少于EECBS; 在1000_60、1000_80、1000_100规模下

3.2 实验分析

3.2.1 MLAMR与KIVA对比实验

为验证MLAMR“货箱到人”模式的有效性, 与KIVA“货架到人”模式进行对比实验. 为排除不同算法对实验的干扰, 本节不运行PCBS算法, 而是选择CBS算法^[17]进行实验. 在CBS算法中, 除针对MLAMR的多目的地属性进行相应改动外, 未对路径搜索算法进行其他改动. 实验在每个问题规模下均重复进行20次, 取平均值作为对比数据. 实验结果如表2所示.

CBS所得路径时间与EECBS路径时间基本持平, 表明EECBS在大规模问题求解策略上仍然有改进空间. 5) 在所有规模实验中, PCBS算法运行时间均明显少于CBS和EECBS算法, 在中大规模问题下, PCBS算法的搜索效率优势更加明显. 6) PCBS算法有效, 且在大规模问题上表现出较大优越性.

表3 PCBS算法性能测试结果

问题规模	CBS							
	路径时间			无效路径比			运行时间	
	bst	wst	avg	bst	wst	avg	avg	
600	20	8907	9132	9035	0.59	0.65	0.62	29.67
	40	6824	7071	6988	0.65	0.72	0.68	22.53
	60	3806	4193	4097	0.77	0.89	0.83	86.49
	80	2597	2982	2708	0.92	1.05	1.02	189.02
	100	1905	2114	2089	1.28	1.39	1.33	422.31
700	20	13599	14331	14008	0.63	0.73	0.67	61.88
	40	11816	13197	12613	0.77	0.85	0.81	103.61
	60	10175	11931	10974	0.8	0.94	0.89	147.97
	80	7610	8309	8062	0.99	1.16	1.04	522
	100	6555	7342	6901	0.81	1.07	0.99	708.09
800	40	17148	18450	17967	0.74	1.01	0.87	589.2
	60	14671	16077	15380	0.91	1.11	1.05	557.08
	80	12033	13980	12843	1.28	1.5	1.38	2409.33
	100	9886	11281	10519	1.43	1.77	1.64	3852.1
900	60	20064	23490	21030	1.06	1.38	1.29	1285.75
	80	15983	20595	18497	1.62	1.74	1.7	3491.6
	100	13211	14800	13990	1.49	1.74	1.68	7940.25
1000	60	26533	29246	27990	1.28	1.6	1.44	5209.3
	80	23979	26802	25889	1.88	2.11	1.94	17358
	100	20195	23220	21467	1.71	1.9	1.83	39150.42

表3 (续)

问题规模	EECBS							
	路径时间			无效路径比			运行时间	
	bst	wst	avg	bst	wst	avg	avg	
600	20	7816	8945	8305	0.4	0.63	0.52	30.05
	40	5463	6021	5708	0.48	0.67	0.6	31.94
	60	3124	3708	3409	0.62	0.89	0.74	28.1
	80	1792	2431	2015	0.81	1.06	0.92	28.47
	100	1438	2019	1759	0.99	1.25	1.1	39.65
700	20	11023	13174	12794	0.52	0.78	0.61	91.08
	40	11478	12980	11680	0.61	0.81	0.69	86.22
	60	8913	10634	9593	0.67	0.98	0.8	107.35
	80	6763	6892	6810	0.87	1.04	0.92	116.14
	100	3461	4556	4185	1.05	1.39	1.13	110.71
800	40	15233	18750	16534	0.58	0.97	0.81	588
	60	12209	14153	12918	0.89	1.12	0.97	514.63
	80	9071	11900	9698	1.07	1.46	1.22	416.91
	100	8480	9312	9073	1.29	1.73	1.49	794.8
900	60	19147	24085	21978	0.94	1.38	1.16	631.85
	80	18428	23251	19160	1.28	1.7	1.41	821.9
	100	13710	17964	15318	1.19	1.73	1.58	1740.32
1000	60	23259	39406	27381	1.14	1.78	1.39	1901
	80	24550	29741	26000	1.36	2.2	1.81	5428.2
	100	21012	24199	22763	1.71	2.15	1.93	8071.49

问题规模	PCBS							
	路径时间			无效路径比			运行时间	
	bst	wst	avg	bst	wst	avg	avg	
600	20	7845	9126	8435	0.4	0.48	0.41	15.26
	40	5409	6113	5880	0.4	0.53	0.45	14.23
	60	2903	3149	3068	0.43	0.61	0.53	14.07
	80	1754	1928	1805	0.63	0.73	0.69	10.8
	100	1301	1765	1480	0.82	0.9	0.87	10.33
700	20	12120	15070	13092	0.4	0.47	0.43	26.52
	40	11043	12602	11837	0.49	0.62	0.57	22.37
	60	9276	10831	9650	0.61	0.87	0.71	22.15
	80	5453	6528	5716	0.7	0.77	0.73	19.94
	100	3022	3790	3398	0.59	0.71	0.66	19.01
800	40	12347	16723	14735	0.39	0.65	0.54	201.69
	60	10866	12095	11368	0.6	0.76	0.7	185.34
	80	9038	11082	9772	0.79	1.08	0.91	116.45
	100	8780	9906	9290	0.67	0.89	0.76	170.56
900	60	17370	19812	18796	0.66	0.91	0.73	472.1
	80	15824	19109	17951	0.73	1.02	0.81	290.48
	100	12299	14701	12920	0.71	0.98	0.83	147.32
1000	60	21901	25683	24537	0.74	1.01	0.83	1254.75
	80	19798	21788	21054	0.99	1.35	1.29	1008.25
	100	18033	21045	19983	1.07	1.21	1.16	613

4 结论

随着智能化仓储快速发展,基于KIVA的“货架到人”模式已无法满足拣选需求.针对这一现象,本文提出了一种基于MLAMR“货箱到人”模式的新型拣选-配送MAPF问题,建立数学规划模型,设计了一种PCBS算法进行优化求解.在PCBS高层搜索上,提出了一种MLAMR优先级规则,并基于该规则设计了

冲突消解策略.在低层,首先提出了一种基于动态规划的单MLAMR拣货顺序确定方法;然后引入转向惩罚策略,设计了增强A*算法进行单MLAMR路径搜索.通过实验发现:1)MLAMR“货箱到人”模式有效,且在中、大规模问题中比KIVA“货架到人”模式表现更好;2)所提出PCBS算法能够有效缩短系统路径时间,减少无效路径时间.

参考文献(References)

- [1] Li X W, Hua G W, Huang A Q, et al. Storage assignment policy with awareness of energy consumption in the Kiva mobile fulfilment system[J]. *Transportation Research Part E: Logistics and Transportation Review*, 2020, 144: 102158.
- [2] Stern R, Sturtevant N, Felner A, et al. Multi-agent pathfinding: Definitions, variants, and benchmarks[J/OL]. 2019, arXiv: 1906.08291.
- [3] Ma H, Li J Y, Kumar T K S, et al. Lifelong multi-agent path finding for online pickup and delivery tasks[J/OL]. 2017, arXiv: 1705.10868.
- [4] 马文凯, 吴耀华, 吴颖颖, 等. 基于进化算法的跨巷道多层穿梭车仓储系统的研究[J]. *机械工程学报*, 2019, 55(8): 216-224.
(Ma W K, Wu Y H, Wu Y Y, et al. Research on cross-aisles multi-shuttle warehouse system based on evolutionary algorithm[J]. *Journal of Mechanical Engineering*, 2019, 55(8): 216-224.)
- [5] Roy D, Nigam S, de Koster R, et al. Robot-storage zone assignment strategies in mobile fulfillment systems[J]. *Transportation Research Part E: Logistics and Transportation Review*, 2019, 122: 119-142.
- [6] Ganeshmurthy M S, Suresh G R. Path planning algorithm for autonomous mobile robot in dynamic environment[C]. *The 3rd International Conference on Signal Processing, Communication and Networking*. Chennai, 2015: 1-6.
- [7] Cheng L P, Liu C X, Yan B. Improved hierarchical A-star algorithm for optimal parking path planning of the large parking lot[C]. *IEEE International Conference on Information and Automation*. Hailar, 2014: 695-698.
- [8] Ren Z Q, Rathinam S, Choset H. MS: A new exact algorithm for multi-agent simultaneous multi-goal sequencing and path finding[C]. *IEEE International Conference on Robotics and Automation*. Xi'an, 2021: 11560-11565.
- [9] 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.
- [10] Wen L C, Liu Y, Li H L. CL-MAPF: Multi-agent path finding for car-like robots with kinematic and spatiotemporal constraints[J]. *Robotics and Autonomous Systems*, 2022, 150: 103997.
- [11] 张丹露, 孙小勇, 傅顺, 等. 智能仓库中的多机器人

- 协同路径规划方法[J]. 计算机集成制造系统, 2018, 24(2): 410-418.
(Zhang D L, Sun X Y, Fu S, et al. Cooperative path planning in multi-robots for intelligent warehouse[J]. Computer Integrated Manufacturing Systems, 2018, 24(2): 410-418.)
- [12] 郭超, 陈香玲, 郭鹏, 等. 基于通用变邻域搜索的多AGV分拣调度优化[J]. 河北科技大学学报, 2021, 42(5): 523-534.
(Guo C, Chen X L, Guo P, et al. General variable neighborhood search for the multi-AGV scheduling problem with sorting operations[J]. Journal of Hebei University of Science and Technology, 2021, 42(5): 523-534.)
- [13] Sartoretti G, Kerr J, Shi Y F, et al. PRIMAL: Pathfinding via reinforcement and imitation multi-agent learning[J]. IEEE Robotics and Automation Letters, 2019, 4(3): 2378-2385.
- [14] Damani M, Luo Z Y, 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.
- [15] 许万, 杨晔, 余磊涛, 等. 一种基于改进RRT*的全局路径规划算法[J]. 控制与决策, 2022, 37(4): 829-838.
(Xu W, Yang Y, Yu L T, et al. A global path planning algorithm based on improved RRT*[J]. Control and Decision, 2022, 37(4): 829-838.)
- [16] 阮晓钢, 周静, 张晶晶, 等. 基于子目标搜索的机器人目标导向RRT路径规划算法[J]. 控制与决策, 2020, 35(10): 2543-2548.
(Ruan X G, Zhou J, Zhang J J, et al. Robot goal guide RRT path planning based on sub-target search[J]. Control and Decision, 2020, 35(10): 2543-2548.)
- [17] Sharon G, Stern R, Felner A, et al. Conflict-based search for optimal multi-agent pathfinding[J]. Artificial Intelligence, 2015, 219: 40-66.
- [18] 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.
- [19] Ma H, Harabor D, Stuckey P J, et al. Searching with consistent prioritization for multi-agent path finding[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33: 7643-7650.
- [20] Čáp M, Novák P, Kleiner A, et al. Prioritized planning algorithms for trajectory coordination of multiple mobile robots[J]. IEEE Transactions on Automation Science and Engineering, 2015, 12(3): 835-849.
- [21] Vainshtein D, Solovey K, Salzman O. Multi-agent terraforming: Efficient multi-agent path finding via environment manipulation[J/OL]. 2022, arXiv: 2203.10540.
- [22] Madar N, Solovey K, Salzman O. Leveraging experience in lifelong multi-agent pathfinding[J/OL]. 2022, arXiv: 2202.04382.
- [23] 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 7th Annual Symposium on Combinatorial Search. Prague, 2014: 19-27.
- [24] Li J Y, Ruml W, Koenig S. EECBS: A bounded-suboptimal search for multi-agent path finding[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35(14): 12353-12362.
- [25] Chan S, Li J, Harabor D, et al. Nested ECBS for bounded-suboptimal multi-agent path finding[C]. Proceedings of the IJCAI Workshop on Multi-Agent Path Finding. Yokohama, 2020.
- [26] Ai B, Jiang J C, Yu S S, et al. Multi-ggent path finding with heterogeneous edges and roundtrips[J]. Knowledge-Based Systems, 2021, 234: 107554.
- [27] Andreychuk A, Yakovlev K, Surynek P, et al. Multi-agent pathfinding with continuous time[J]. Artificial Intelligence, 2022, 305: 103662.
- [28] Cohen L, Uras T, Koenig S. Feasibility study: Using highways for bounded-suboptimal multi-agent path finding[J]. Proceedings of the International Symposium on Combinatorial Search, 2021, 6(1): 2-8.
- [29] Wang K, Botea A. Fast and memory-efficient multi-agent pathfinding[C]. Eighteenth International Conference on Automated Planning & Scheduling. Sydney, 2008: 380-387.
- [30] 朱军, 唐万奇, 李凯. 基于曼哈顿距离模糊C聚类及粒子群优化的中继车辆选择算法[J]. 安徽大学学报: 自然科学版, 2021, 45(4): 35-40.
(Zhu J, Tang W Q, Li K. The relay vehicle selection algorithm based on Manhattan distance-based fuzzy C clustering and particle swarm optimization[J]. Journal of Anhui University: Natural Science Edition, 2021, 45(4): 35-40.)

作者简介

张洪琳(1993—), 男, 博士生, 从事多智能体路径规划、智能仓储系统任务调度等研究, E-mail: 201920522@mail.sdu.edu.cn;

吴耀华(1963—), 男, 教授, 博士生导师, 从事自动分拣系统、集成化物流系统规划、设计和仿真等研究, E-mail: MIKE.WU@263.net;

胡金昌(1992—), 男, 讲师, 博士, 从事生产调度、物流系统优化等研究, E-mail: 621116@sdu.edu.cn;

张健(1968—), 女, 讲师, 博士, 从事仓储拣选系统优化、物流系统优化等研究, E-mail: zhangjian@sdu.edu.cn.

(责任编辑: 魏冰)