基于信息物理系统的多分拣移动机器人调度策略

廉胤东[1], 刘彦汝[1], 杨启帆[1], 杨奕斌[1], 谢 巍[1,2†]

(1. 华南理工大学 自动化科学与工程学院，广州 510640；
2. 广东省电力系统网络安全企业重点实验室，广州 510663)

摘 要: 本文主要研究了基于信息物理系统建模的多分拣移动机器人 (multi-SMR) 调度策略. 首先, 在基于实际应用场景的拓扑地图建模中加入了新的路径弧时间损耗指标, 实现了对货物不均匀比例和多机器人拥堵状态的精确估计. 其次, 提出了一种改进的启发式路径规划算法, 并在路径评估过程中增加了目的地距离和时间损耗指标. 最后, 完整的调度过程以分层式结构部署在信息物理系统模型中, 包括控制层的时间损耗指标更新、交通管制监测，以及物理层的分布式路径规划和机器人状态更新. 仿真实验验证了本文方法的可靠性, 结果表明, 改进调度策略进一步提升了系统分拣效率, 降低了计算成本, 有效解决了机器人拥堵和安全问题.

关键词: 多移动机器人；信息物理系统 (CPS)；智慧分拣；启发式规划；分层调度；拓扑地图建模

中图分类号: TP242.6 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2021.1343 **开放科学 (资源服务) 标识码 (OSID):**

Multi-Sorting Mobile Robot Scheduling Strategy Based on Cyber-Physical System

LIAN Yin-dong[1], *LIU Yan-ru*[1], *YANG Qi-fan*[1], *YANG Yi-bin*[1], *XIE wei*[1,2†]

(1. School of Automation Science and Engineering，South China University of Technology, Guangzhou 510640，China；
2. Guangdong Provincial Key Laboratory of Power System Network Security，Guangzhou 510663，China)

Abstract: This paper mainly studies the CPS-based multi-sorting mobile robot (multi-SMR) scheduling strategy. First, a novel time cost is added to the topology map modeling, realizing the accurate estimation of the uneven proportion of cargos and the congestion state of multiple robots. Second, an improved path planning algorithm is proposed, and the path evaluation process adds the destination distance and time cost indicators. Finally, the complete scheduling process is deployed in the CPS model in a hierarchical structure, including the time cost update and traffic control of the control layer and the distributed path planning and robot status update of the physical layer. Simulation experiments verify the reliability of the proposed method. Results show that the improved scheduling strategy further improves the system sorting efficiency, reduces the calculation cost, and solves the problem of robot congestion.

Keywords: Multiple mobile robots；cyber-physical system (CPS)；smart sorting；heuristic planning；hierarchical scheduling；topology map modeling

0 引 言

多移动机器人系统能够在大幅提高工作效率和精确度的前提下降低人为错误和潜在的安全风险[1], 其在物流分拣领域的应用是广泛和成熟的, 如亚马逊、阿里巴巴、京东、DHL 等[2]. 基于信息物理系统 (Cyber-Physical System, 以下简称 CPS) 和物联网 (Internet of Things, IoT) 的多智能机器人协作技术日益成熟. CPS 通过无线通信网络将控制机构与执行机构有机联系在一起, 消除了有线连接和固定网络结构的限制, 增强了系统的灵活性和可拓展性[3].

本文主要研究物流分拣领域的多分拣移动机器人 (multi-sorting mobile robot, 以下简称 multi-SMR) 系统. 传统的建模架构如图 1 所示[4]. 其中, 路径规划的主要目的是根据分拣任务和机器人状态进行最优的资源调度 (包括机器人和路径资源), 同时保证系统效率和安全. 在物理层, 机器人的车载处理器是传感器和执行部件数据的集成处理与交互中心, 其计算过程可以视为在系统边缘进行. 在 CPS 架构下, 移动机器人通过多个传感器 (如 RFID, 摄像头, 雷达) 感

收稿日期: 2021-07-31；**录用日期:** 2021-10-27.
基金项目: 国家自然科学基金项目 (61973125).
†通讯作者. E-mail: weixie@scut.edu.cn.

知环境, 获取实时的位置信息和自身状态. 网络层是控制层和物理层的连接枢纽, 实现可靠的数据传输.

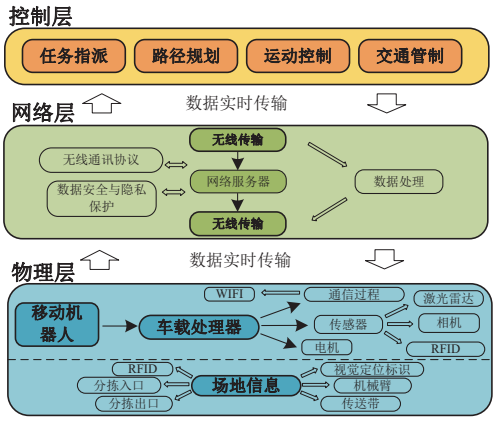


图 1 ：基于 CPS 的 multi-SMR 系统模型

Multi-SMR 系统调度过程中, 路径规划将直接影响系统可靠性. 事实上, 规划性能取决于多种因素, 包括场景地图建模、规划评价指标、算法部署架构等. 当前研究和应用中仍然存在以下问题: (1) 不同包裹对应的目的地比例不均, 且不可避免, 考虑调度过程中货物的不均衡性是必要的; (2) 物流系统的场地面积大, 而机器人比例相对较低, 提高机器人密度与调度策略密切相关; (3) 机器人任务过程存在时间层面的高度不确定性, 需要从算法结构上进行优化.

针对上述问题, 本文的主要改进包括: (1) 增加新的时间损耗评价机制, 用以评估货物分配不均和解决机器人运行过程可能出现的时间不确定问题. 具体的, 本文方法考虑了货物数量的不均衡比例, 据此对多移动机器人的运动路径进行拓扑建模, 并在启发式路径规划过程中加入时间和距离约束; (2) 利用 CPS 模型改进了算法部署结构, 其中, 路径规划算法被独立部署在物理层的每台移动机器人上, 而控制层主要负责约束状态的更新和冲突避让.

1 相关研究

近期的研究提出了多种方案, 包括 CPS 架构下的物流分拣系统[4], 以及启发式规划方法[5]. 对于不同应用场景, 采用不同的评价指标是合理的[1].

文献 [4] 构建了基于 CPS 的车间物料搬运模型, 物理层的物理实体独立执行任务, 而调度和控制算法全部集中在控制层, 当移动机器人规模较大时, 大规模数据传输将影响调度效率. Dai 等人提出了一种基于数据驱动的 CPS 自治管理方法[6], 确保了执行层任务数据对决策过程的约束, 实现了对物理实体的高效管理和控制, 保证了执行机构的任务可靠性. Lin 和 Yang 提出了一种适用于 CPS 建模下物流系统的云边雾 (cloud-edge fog) 计算[7], 算法将部分计算任务部署在系统边缘 (通常为物理层), 有效降低了计算延迟. 此外, CPS 系统的通信过程也被深入研究[8].

在规划方面, 郭昆仑等人提出了一种改进的 Dijkstra 算法[9], 增加了可同时规划的 AGV 数量, 而单次规划效率较低. 进一步的, 相关研究已经验证了启发式规划算法在移动机器人系统中的有效性, 如 A* 和 D* 算法[5,10]. 王云峰等人提出了一种改进 A* 算法[11], 降低了 AGV 运动过程的频繁转向和冲突. Digani 等人提出了一类适用于多工业移动机器人系统的地图建模和调度策略[5], 首先利用模型预测控制 (MPC) 对拥堵状态进行评估, 进而采用 D* 算法实现路径规划. 分层调度策略进一步细化了规划任务, 而上述过程仍集中在主控单元内, 这将影响系统的计算效率和实时性. 李明辉等人提出了一种适用于自动化物料运输系统的路径规划策略[12], 将车辆经过路径的时间损耗设置为评价依据, 在任务部署、冲突避让方面具有良好效果. 然而, 时间损耗被设为常值, 不适用于 AGV 数量较多的情景. 孙兆臣等人提出了一种考虑任务优先级的改进 A* 算法[13], 实现了在时间层面的评估, 验证了时间损耗评估的可靠性. 文献 [10] 提出了一种基于视觉定位估计和路径时间损耗的改进 A* 规划算法, 其中, 启发函数的评估过程考虑了未来路径可能产生的路径时间耗费, 这将有利于对全局路径资源的评估. 然而, 利用单一时间指标进行评估的过程可能造成 AGV 的无序运动.

2 Multi-SMR 系统调度策略

2.1 基于不均衡路径时间损耗的拓扑地图建模

针对货物不均衡问题, 一类主要的策略是增加交通流密度指标[4,5], 这通常在拥塞较严重时才产生有效的控制信号. 相较于空间的拥堵, 时间指标能够更好地反映 multi-SMR 运动时的任务和空间位置特性. 图 2 展示了根据实际场景建立的拓扑地图, 主要由路径弧和转向节点组成. 路径节点具有独立的坐标信息. 此外, 分拣出入口通常环绕在路径弧之中.

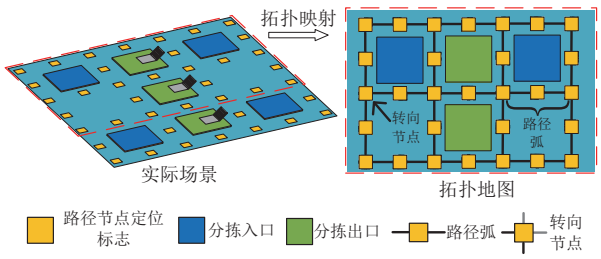


图 2 ：应用场景拓扑地图建模

定义 1 $\mathcal{R}=\{\mathcal{N},\mathcal{P},\mathcal{C}(t)\}$ 表示包含不均衡时

间约束的拓扑地图模型, 其中 $\mathcal{N} = \{1, 2, \cdots, n\}$, $\mathcal{P} = \{p_{m,n} | m, n \in \mathcal{N}\}, \mathcal{C}(t) = \{c_{mn}(t) | m, n \in \mathcal{N}, t \in [t_i(m), t_f(n)]\}$ 分别表示路径节点集合, 路径弧集合, 以及路径时间损耗集合. $t_i(m)$ 和 $t_f(n)$ 表示机器人进入和离开路径弧 $p_{m,n}$ 的时间节点.

此外, 令 $\mathcal{X} = \{x_{ij} | i, j \in \mathcal{N}\}$ (i, j 是路径弧两端的节点, 并且 i 为起始节点) 表示分拣入口 s 与分拣出口 d 之间的完整路径集合, 这包含了该路径上的所有路径弧. 弧 $x_{ij} \in \mathcal{P}$, 其取值范围为 $x_{ij} = \{0, 1\}$, 其中数值 1 和 0 分别表示该段弧在规划路径中存在与否. 因此, 对于每一台移动机器人, 期望的最优路径应该是节点连续并且时间损耗最小的. 最小时间损耗路径的约束可以表示为:

$$\begin{aligned} \min \quad & \mathcal{C} = \sum c_{ij} x_{ij}, \\ \text{s.t.} \quad & \forall c_{ij} x_{ij} \geq 0, \\ & \sum (x_{sj}) - \sum (x_{js}) = 1 \ (j \in N, j \neq s), \\ & \sum (x_{ij}) - \sum (x_{ji}) = 0 \ (i \in N, i \neq s, i \neq d), \\ & \sum (x_{dj}) - \sum (x_{jd}) = -1 \ (j \in N, j \neq d), \end{aligned} \tag{1}$$

其中 c_{ij} 表示每段路径弧的时间损耗, $\mathcal{C}$ 是总的损耗. 考虑 x_{ij} 的取值限定为 0 或 1, 约束 $\forall c_{ij} x_{ij} \geq 0$ 保证了损耗指标的非负性, 这有利于具体的实现. 其他三项约束保证了规划路径的连续性和可计算性.

由于货物数量不均匀, 不同路径弧被占用的时间是不同的, 这进一步反映了系统内的拥塞状态. 根据定义 1, 路径弧之间是相互独立的, 可以将每段路径弧的时间损耗视为基于时间的独立随机过程. 因此, 在确定规划路径的最小时间损耗前, 应计算拓扑地图中所有弧时间损耗的分布情况.

对于公式 (1) 约束下的最优路径, 总的时间损耗应为:

$$\mathcal{C}(t_0) = \sum c_{ij}(t) x_{ij}, \tag{2}$$

且总体损耗的概率分布函数为:

$$F_{\mathcal{C}}(t_0) = \prod F_{c_{ij}}(x_{ij}, t)^{x_{ij}}, \tag{3}$$

其中 t_0 表示 multi-SMR 系统的启动时间, $F_{c_{ij}}(x_{ij}, t)$ 表示每段路径弧独立的路径时间损耗分布函数. 在计算过程中, 每段弧上产生的时间损耗具有随机性. 受到机器人空间位置的影响, 期望的路径损耗分布 (密度) 函数可能具有时变、非线性特性.

引理 1 对于独立的随机变量 $\mathcal{R}_1$ 和 $\mathcal{R}_2$, 可以通过其期望值大小进行比较. 若 $\mathcal{R}_1$ 和 $\mathcal{R}_2$ 的期望值为 $E[\mathcal{R}_1]$ 和 $E[\mathcal{R}_2]$, 则 $E[\mathcal{R}_1] \leq E[\mathcal{R}_2]$ 等价于 $P(\mathcal{R}_1 \leq \mathcal{R}_2) \geq 0.5$.

根据引理 1[14], 若可以利用公式 (3) 获得损耗的最小概率分布值, 或者直接从统计量获取期望损耗 $E[c_{ij}(t) x_{ij}]$ 时, 则可以通过比较总的期望损耗来确定最优路径.

由于时间损耗具有不确定性, 精确建模求解是较为困难的, 本文采用统计求解方法 (统计量的正态分析[14]). 在正态分析中, 偏度和峰度反映了统计数据的三阶和四阶特性, 这是判断周期内统计数据正态性的重要指标. 在实验部分, 偏度和峰度的统计值在 0 ± 0.5 范围内是合理的. 此外, Kolmogorov-Smirnov (K-S) 和 Shapiro-Wilk (S-W) 检验也需要进一步验证, 其检验置信度应该高于 0.05.

2.2 改进启发式路径规划算法

本文结合拓扑地图建模对 A* 算法进行改进.

定义 2 改进 A* 算法的启发 (损失) 函数为:

$$\mathcal{F}(k, t_s) = \mathcal{G}(k, t_s) + \mathcal{H}(k, t_s), \tag{4}$$

其中 t_s 记录机器人进入起始节点的时间, 并且在重新规划时更新. k 表示当前所在的节点位置.

与传统 A* 算法 ($f(n) = g(n) + h(n)$, 其中 n 表示当前节点位置) 类似, 改进的启发函数包含对已生成路径损耗的记录 $\mathcal{G}(k, t_s)$, 以及下一步规划中的路径损耗估计 $\mathcal{H}(k, t_s)$. 具体的, $\mathcal{G}(k, t_s)$ 描述了从起始位置到当前节点的真实损耗. 若 $\mathcal{D}_p$ 表示起始位置与当前位置间的 Manhattan 距离[10], 机器人的平均移动速度为 v_a, 弧时间损耗为 $\mathcal{T}_p$, 转向时间损耗为 $\mathcal{T}_c$, 那么 $\mathcal{G}(k, t_s)$ 可以表示为:

$$\mathcal{G}(k, t_s) = \mathcal{D}_p / v_a + \sum_1^l \mathcal{T}_p + \sum_1^h \mathcal{T}_c, \ (\text{包含转向}), \tag{5}$$

以及

$$\mathcal{G}(k, t_s) = \mathcal{D}_p / v_a + \sum_1^l \mathcal{T}_p, \ (\text{无转向}), \tag{6}$$

其中 l 为已经过路径中的弧节点数, h 为已经过路径中的转向节点数. $\mathcal{G}(k, t_s)$ 包含两种状态.

对未来路径的评估 $\mathcal{H}(k, t_s)$ 可以表示为:

$$\mathcal{H}(k, t_s) = \mathcal{D}_d / v_a + \sum_1^r E[c_{ij}(t)], \tag{7}$$

其中 $\mathcal{D}_d$ 表示当前位置与目标节点间的 Manhattan 距离, r 为评估路径中的弧节点数, $E[c_{ij}(t)]$ 为拓扑地图中弧的时间损耗期望值. 算法 1 描述了路径规划的实现过程, 搜索过程包含 OPEN 和 CLOSE 两个列表, 用于从起始点逐个进行节点评估, 并保存路径. 评估过程比较了每个节点的启发函数值, 最优路径具有最小损耗值. 注意 t_s 更新前后的时间差值可以作为统计量, 表示该段弧的被占用时间.

算法 1 改进 A* 路径算法 (单台移动机器人):

输入: 地图信息 Map, 弧损耗集合 $\mathcal{C}(t)\{c_{mn}\}$, OPEN 表, CLOSE 表, 起始节点 S, 终点节点 D;

输出: 路径集合 $\mathcal{X}\{x_{ij}\}$;
运算变量: $\mathcal{F}(k,t_s), \mathcal{G}(k,t_s), \mathcal{H}(k,t_s), t_s$;
初始化: OPEN & CLOSE = null, t_s= 0;
1 Record the current t_s; //记录当前时刻
2 $S.\mathcal{G}(k,t_s) = 0$; //S 节点的 $\mathcal{G}$ 值
3 CLOSE.add(S); //添加 S 到 CLOSE 表
4 CLOSE.add(S.NextNode); //S 的相邻节点
5 **while** D not in CLOSE **do**
6 Node i = CLOSE.LatestNode; //最近加入 CLOSE 的节点
7 **for** count = 1:size(Map.(i).NeighborNode), count++ **do**
8 OPEN.add(Map.(i).NeighborNode); //i 的相邻节点
9 $i.\mathcal{F}(k,t_s) = i.\mathcal{G}(k,t_s) + i.\mathcal{H}(k,t_s)$;
10 **end**
11 Sort(OPEN); //根据 $\mathcal{F}$ 值从小到大排序;
12 CLOSE.add(OPEN.LatestNode); //添加 OPEN 表中的最新节点到 CLOSE 表
13 Clean OPEN; //结束 i 附近节点的搜索
14 **end**
15 return $\mathcal{X}\{x_{ij}\}$ = CLOSE;

2.3 基于 CPS 的分层调度算法部署

调度策略中路径规划、交通管制等算法的协同工作将直接影响系统的整体性能. 当前研究中[4,5], 算法的实现过程通常是集中式的, 如图 1 所示.

路径规划算法直接影响了机器人和场景地图间的关系, 这将在实验部分进一步说明. 采用分层式调度的目的是进一步划分规划过程的子任务, 包括评价指标的生成、更新, 以及具体的路径计算. 从计算复杂度来看, 路径计算需要考虑所有机器人和拓扑地图节点, 这将占用大部分计算资源.

交通管制 (冲突避让) 取决于移动机器人的传感器信息. 发生冲突时, 物理层的机器人会立即停车并向控制层上报冲突信息, 等待解除冲突的控制命令. 随着工业和制造业智能化程度的提升 (如更大量的图形计算和传输), 物理层中物理实体的自治性和网络层的可靠通信需求进一步增加.

CPS 有效连接了物理层和控制层, 每一台机器人被分布式地部署在系统的计算边缘, 其计算过程都是独立的. 本文尝试将分层调度过程部署在 CPS 的不同模块中, 如图 3 所示. 地图信息 (时间损耗更新) 和冲突避让算法部署在控制层, 这是所有物理实体共享的信息. 路径规划过程独立部署在每一台机器人中 (物理层), 边缘计算和运动指令执行是分布式的. 因此, 每台机器人与控制层独立交互, 获取指令信息并自主规划最优路径.

文献 [4] 讨论并验证了物理实体与控制层之间实时交互信息的可靠性, 在移动机器人上部署计算任务是可行的. 此外, 改进部署方式的一个明显优势是算力的均衡分布, 控制层中的大部分规划计算被重新分配至边缘计算中. 同时, 交通管制策略在控制层中实现, 保证了全局安全.

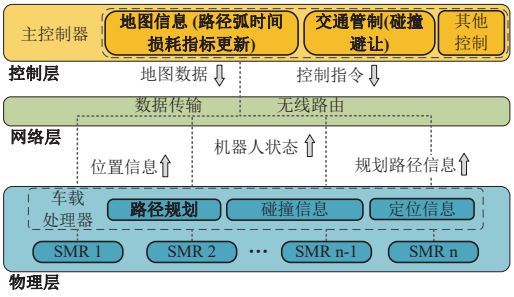


图 3 ：基于 CPS 的改进分层调度架构

3 对比实验与分析

3.1 实验配置和准备工作

如图 4 所示, 本文实验在自主开发的软件平台进行. 实验过程主要对控制层的主控制器和物理层的 multi-SMR 系统进行模拟. 软件在两台计算机上同时执行 (Windows 10 操作系统, Intel i7-8750H 2.20GHz CPU, 16.0GB RAM), 分别对应控制层和物理层.



图 4 ：模拟实验软件平台

实验地图在图 2 的基础上进行扩展, 形成 8×14 规模的标准栅格地图. 地图包含 314 个路径节点, 包括 202 个弧节点, 路径弧长度设置为 2m. 此外, 地图模拟了 8 个分拣入口和 36 个分拣出口. 移动机器人的数量设置为 29 台, 其速度为 1m/s, 装载和卸货时间分别为 20s 和 10s. 仿真实验中的地图规模、车辆数量配置均参考当前的研究[2,5] 和应用[15]. 此外, 为了模拟货物不均衡情况, 8 个分拣入口的货物产生比例被设置为 9:7:5:3:2:1:1:1. 对比实验选择了四组算法进行模拟, 具体配置如表 1 所示. 模型 1 采用了传统 A* 算法, 其主要规划指标为当前位置与目标节点的最短距离. 模型 2[5] 采用了分层式规划方式, 模型预测控制 (MPC) 首先用于评估拓扑地图中的区域车辆密度, 获得的拥堵值将作为改进 D* 算法的规划指标. D* 算法根据场景内车流密度的变化实时重新规划. 模型 3[10] 同样采用了分层规划方式, 分别对下一节点位置和路径时间损耗进行估计, 改进 A* 算法的评

估过程考虑了机器人在地图网络中的占用时间. 此外, 冲突避让算法考虑了估计节点的时间窗序列. 实验默认模型 3 直接检测到下一节点的坐标, 没有增加视觉解码过程. 模型 4 采用了本文设计的改进方法. 在实验过程中, 机器人之间的死锁被记录, 并通过反向重新规划进行解除. 任务分配过程选择距离分拣入口最近的移动机器人.

3.2 实验结果和分析

初始化阶段, 路径弧损耗设定为 0.9s, 转向时间损耗设置为 1s. 初始化时间不会影响后续的弧时间损耗统计. 实验首先记录了系统运行 12 小时内的时间损耗情况, 数据表明, 不均衡货物分配对不同位置弧的时间损耗具有明显影响. 随着统计时间的增加, 部分弧损耗的统计量呈现正态分布. 表 2 记录了系统运行 15 分钟内两段路径弧的正态分析数据. 可以看出, 峰度和偏度的统计量趋近于 0, 而 K-S 和 S-W 测试的显著性高于 0.05. 此外, 分拣入口附近的弧呈现出较明显的拥堵状态. 根据总体的统计数据, 超过 80% 的路径弧在每 15 分钟周期内呈现正态特性. 对于正态数据, 周期内的正态均值被设置为新的弧时间损耗, 用于路径规划过程的评估. 对于非正态数据, 周期内的时间损耗平均数被设置为新的损耗值.

图 5 展示了三项关键系统指标的实验结果 (基于 15 组实验数据, 每组模拟实验的时间超过 4 小时). 分拣效率指多移动机器人在每小时内完成的分拣任务数. 模型 4 的分拣效率达到模型 1 的 1.73 倍、模型 2 的 1.23 倍、模型 3 的 1.13 倍. 模型 3 中改进 A* 算法的启发函数考虑了路径时间损耗指标, 这与模型 4 的估计过程相似. 模型 4 考虑了时间和距离约束, 机器人在朝向目的地方向的前提下尽可能避免了拥堵.

产生停车时间 [图 5(b)] 的主要原因是路径拥堵. 模型 1 仅考虑了最短距离, 多台机器人在短时间内大量集中于分拣出入口, 产生较严重的拥堵. 模型 2、模型 3 和模型 4 的规划评价过程分别基于交通拥堵状态和路径被占用时间, 其直接目的是根据交通状态规划下一步的路径, 因此能够有效缓解拥堵. 一旦出现拥堵和排队, 模型 2 系统将立刻指派机器人至附近的空闲路径中, 大幅减少交通拥堵. 但是, 当多个拥堵位置相对靠近、或包围在空闲区域外部时, multi-SMR 会被同时重新规划到有限的路径资源中, 造成死锁. 相比模型 2 对区域内拥堵的评估, 模型 4 对每一条路径弧的时间损耗进行了更详细的统计和评估, 这在保证机器人流动性的前提下提高了任务效率. 在模型 3 中, 基于队列表的冲突解除方法增加了整体系统的停车时间, 降低了系统内的车辆流通性.

平均规划时间表示每规划 1000 条路径花费的时间. 模型 1 和 4 在规划时间上明显优于模型 2. 模型 1 采用的 A* 算法具有较好的时效性, 在实验拓扑地图中, 已知两点间 A* 算法的时间复杂度为 $O(n)$. 模型 2 的规划时间分配在 MPC 评估和 D* 算法中. 对于细分的 m 个子地图区域, 采用 MPC 进行拥堵值评估的复杂度为 $O(m) < O(n)$. 改进 D* 算法首先利用 Dijkstra 算法 (复杂度为 $O(n^2)$) 进行反向搜索, 接着利用 A* 算法确认最优的路径节点, 总体的复杂度为 $O(n^3)$. 注意当启发指标 (区域密度) 发生变化时, 算法会多次使用 A* 重新规划路径, 导致了规划时间的大幅增加. 模型 4 中, 时间损耗更新针对所有弧节点 p, 且满足 $O(p) < O(n)$. 在规划方面, 改进算法保持了 A* 的复杂度 $O(n)$. 整体的算法复杂度应该接近 $O(n^2)$. 事实上, 基于 CPS 的算法部署改变了原算法的时间复杂度, 路径规划算法被部署至物理层, 其复杂度对于控制层来说降低为 $O(1)$. 此外, 弧时间损耗的固定更新周期也避免了类似模型 2 中实时更新的复杂状态. 基于 CPS 的算法部署架构也可以进一步解释模型 1 在规划计算时间上的最优性. 模型 3 不具备 CPS 架构, 这在一定程度上增加了规划时间.

此外, 实验统计了 8 个分拣入口及其附近路径弧的拥堵情况. 表 3 展示了模型 1 和模型 4 在实验过程中的统计数据. 在模型 1 中, 相较于入口 5–8, 1–4 的拥堵情况更明显. 模型 4 有效缓解了高货物比例入口的时间损耗, 避免了因货物不均衡而造成的拥堵.

最后, 实验对比了不同调度策略对机器人密度的影响, 如图 6 所示. 当机器人数量为 10 时, 模型 1 具有最高的任务效率. 随着机器人数量增加 (10-20), 模型 2、模型 3 和模型 4 的任务效率快速上升, 且模型 2 具有更明显的优势. 此时系统内开始出现排队和拥堵, 模型 2 将拥堵车辆指派至附近空闲区域, 缓解了拥堵, 并且增加了任务效率. 模型 1 在此阶段产生了较大的拥堵和排队. 模型 3 的任务效率高于模型 4, 这是由于队列表增加了有序性. 当 SMR 数量进一步增加 (20-40), 基于路径时间损耗的评估方式体现出最优性. 实验过程中, 模型 2 相对频繁地进行重新规划, 导致 multi-SMR 在高任务比的分拣口和路段上产生拥堵, 进而降低了任务效率. 此段范围内模型 3 和模型 4 体现出更好的规划和任务效率. 机器人数量超过 40 台时, 模型 1 中产生了大量不可解除的死锁, 导致系统停滞甚至是完全停止. 随后的过程中 (40-80), 机器人数量接近地图可容纳范围的饱和值. 从任务

效率曲线来看, 模型 2、模型 3 和模型 4 的增幅下降, 并逐步达到饱和值, 随后开始下降. 通过上述实验可知, 本文的设计方法是可靠的, 改进算法有效缓解了机器人的不均衡状态, 并且提升了机器人密度.

4 结论

本文设计了一种基于信息物理系统 (CPS) 的调度策略, 主要贡献包括: (1) 基于时间的拓扑地图建模和路径规划过程优化了路径利用效率, 缓解了拥堵问题; (2) 新的路径评价指标 (目的地距离和时间损耗) 细化了评估过程, 有利于提升系统安全性, 具有实际意义; (3) 基于 CPS 的算法部署、分层式架构和边缘计算共同提高了系统的实时性能和任务效率.

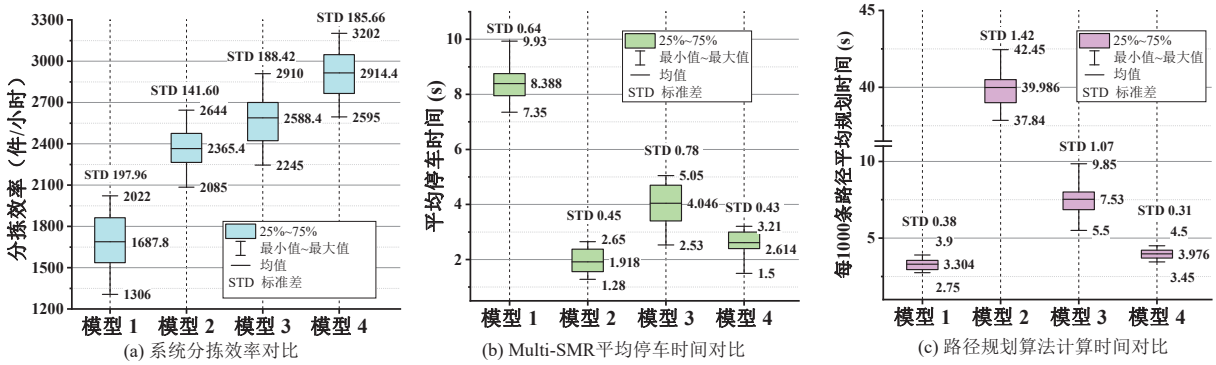
在下一步工作中, 系统中的任务分配策略和时间损耗更新规则将被着重考虑. 此外, multi-SMR 运行的能耗规律和节能规划将被进一步研究.

表 1 四组对比算法配置

	路径规划算法	交通管制 (冲突避让)	启发式	分层调度	CPS 架构	主要评价指标
模型 1	传统 A*(物理层)	传感器信息 (控制层)	具备	不具备	具备	最短路径
模型 2	MPC + 改进 D*(控制层)	传感器信息 (控制层)	具备	具备	不具备	区域车辆密度 (拥堵状态)
模型 3	节点估计 + 改进 A*(控制层)	队列表时间窗 (控制层)	具备	具备	不具备	路径时间、队列表
模型 4	地图信息 (控制层)+ 改进 A*(物理层)	传感器信息 (控制层)	具备	具备	具备	路径弧时间损耗

表 2 弧损耗统计量正态分析

指标 \ 弧编号	1 (靠近分拣入口)			2		
	统计量 (ms)	标准误差 (ms)	显著性	统计量 (ms)	标准误差 (ms)	显著性
95% 均值置信区间	14446.0386 & 14652.8244	/	/	3075.0539 & 3260.8737	/	/
均值	14549.4315	51.4500	/	3167.9638	46.2300	/
最大值	15498.7300	/	/	4086.5700	/	/
最小值	13619.1700	/	/	2572.6400	/	/
全距范围	1879.5600	/	/	1513.9300	/	/
Skewness	.068	.337	/	.263	.355	/
Kurtosis	.721	.662	/	.037	.762	/
K-S	.083	/	.200*	.075	/	.200*
S-W	.984	/	.720	.500	/	.692



(a) 系统分拣效率对比 (b) Multi-SMR平均停车时间对比 (c) 路径规划算法计算时间对比

图 5 ：Multi-SMR 系统关键指标数据对比

表 3 分拣入口及相邻路径平均弧时间损耗 (ms)

分拣入口 \ 路径弧	分拣入口路径	相邻路径弧 1	相邻路径弧 2	相邻路径弧 3	相邻路径弧 4
分拣入口 1 (模型 1)	24525	17357	15104	18353	12394
分拣入口 2 (模型 1)	24437	19957	17104	17700	10394
分拣入口 3 (模型 1)	22133	15000	13100	16421	12235
分拣入口 4 (模型 1)	20037	14021	11039	13521	12047
分拣入口 5 (模型 1)	15428	12124	10362	11541	9521
分拣入口 6 (模型 1)	16425	10255	7036	8544	8021
分拣入口 7 (模型 1)	13524	9051	4328	8065	6857
分拣入口 8 (模型 1)	12023	6511	4421	5569	3842
分拣入口 1 (模型 4)	17954	16325	15420	17744	15426
分拣入口 2 (模型 4)	18835	15004	15210	16234	14320
分拣入口 3 (模型 4)	16540	14474	14021	16453	13572
分拣入口 4 (模型 4)	14233	13351	12844	13524	13501
分拣入口 5 (模型 4)	12550	13324	11517	12521	11003
分拣入口 6 (模型 4)	11320	11047	11523	10521	10021
分拣入口 7 (模型 4)	12475	11544	10023	10541	10058
分拣入口 8 (模型 4)	12755	12033	10074	11541	10732

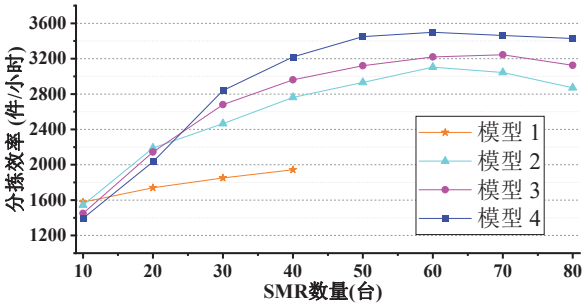


图 6 ：Multi-SMR 密度对任务效率影响对比

参考文献 (References)

[1] 徐翔斌, 马中强. 基于移动机器人的拣货系统研究进展 [J]. 自动化学报, 2020, 46: 1-20.
(Xu X B, Ma Z Q. Robotic mobile fulfillment systems: state-of-the-art and prospects[J]. Acta Automatica Sinica, 2020, 46: 1-20.)

[2] Kehoe B, Patil S, Abbeel P, Goldberg K. A survey of research on cloud robotics and automation[J]. IEEE Transactions on Automation Science and Engineering, 2015, 12(2): 398-409.

[3] Liu Y, Peng Y, Wang B L, Yao S R, Liu Z H. Review on cyber-physical systems[J]. IEEE/CAA Journal of Automatica Sinica, 2017, 4(1): 27-40.

[4] Zhang Y F, Xi D, Zhu Z F, Lv J X. CPS-based smart control model for shopfloor material handling[J]. IEEE Transactions on Industrial Informatics, 2018, 14(4): 1764-1775.

[5] Digani V, Hsieh M A, Sabattini L, Secchi C. Coordination of multiple AGVs: a quadratic optimization method[J]. Autonomous Robots, 2019, 43(3): 539-555.

[6] Dai W B, Dubinin V N, Christensen J H, Vyatkin V, Guan X P. Toward self-manageable and adaptive industrial cyber-physical systems with knowledge-driven autonomic service management[J]. IEEE Transactions on Industrial Informatics, 2017, 13(2): 725-736.

[7] Lin C C, Yang J W. Cost-efficient deployment of fog computing systems at logistics centers in industry 4.0[J]. IEEE Transactions on Industrial Informatics, 2018, 14(10): 4603–4611.

[8] 汪慕峰, 胥布工. DoS 干扰攻击下的信息物理系统状态反馈稳定 [J]. 控制与决策, 2019, 34(8): 1681-1687.
(Wang M F, Xu B G. State feedback stabilization of cyber-physical system under DoS jamming attacks[J]. Control and Decision, 2019, 34(8): 1681-1687.)

[9] 郭昆仑, 朱瑾. 基于多智能体系统的自动化码头多 AGV 无冲突路径规划 [J]. 制造业自动化, 2021, 43(08): 83-89.
(Guo K L, Zhu J. Multi-AGV path planning in automated terminals based on multi-agent system[J]. Manufacturing Automation, 2021, 43(08): 83-89.)

[10] 廉胤东, 谢巍. 基于视觉引导多 AGV 系统的改进 A* 路径规划算法 [J]. 控制与决策, 2021, 36(08): 1881-1890.
(Lian Y D, Xie W. Improved A* path planning algorithm for vision-guided multi-AGV system[J]. Control and Decision, 2021, 36(08): 1881-1890.)

[11] 王云峰, 曹小华, 郭兴. 基于改进 A* 算法和系统短期状态预测的仓储 AGV 路径规划方法 [J/OL]. 计算机集成制造系统, 2021. https://kns.cnki.net/kcms/detail/11.5946.TP.20210809.1137.008.html
(Wang Y F, Cao X H, Guo X. Warehouse AGV path planning method based on improved A* algorithm and system short-term state prediction[J/OL]. Computer Integrated Manufacturing Systems, 2021. https://kns.cnki.net/kcms/detail/11.5946.TP.20210809.1137.008.html)

[12] 李明辉, 徐方, 宋吉来. 整体式自动化物料运输系统防拥堵调度策略 [J]. 计算机工程与设计, 2021, 42(08): 2395-2401.
(Li M H, Xu F, Song J L. Anti-congestion scheduling strategy of unified automatic material handling system[J]. Computer Engineering and Design, 2021, 42(08): 2395-2401.)

[13] 孙兆臣, 孙晓琴, 杨文. 智能化仓库的 AGV 路径规划与紧急避障系统 [J]. 山东农业大学学报 (自然科学版), 2021, 52(03): 483-488.
(Sun Z C, Sun X Q, Yang W. AGV path planning and emergency obstacle avoidance system in intelligent warehouse[J]. Journal of Shandong Agricultural University (Natural Science Edition), 2021, 52(03): 483-488.)

[14] 盛骤. 概率论与数理统计 (第三版)[M]. 北京: 高等教育出版社, 2001: 70-110.
(Sheng Z. Probability theory and mathematical statistics (The 3rd edition)[M]. Beijing: Higher Education Press, 2001: 70-110.)

[15] 吴菁芃. 亚马逊仓库 Kiva 机器人的应用分析与前景展望 [J]. 物流技术与应用, 2015, 20(10): 159-164.
(Wu J P. Application analysis and prospects of Kiva robot in Amazon warehouse[J]. Logistics & Material Handling, 2015, 20(10): 159-164.)

作者简介

廉胤东 (1992–), 男, 博士生, 从事移动机器人调度、信息物理系统的研究, E-mail: aulianyd@mail.scut.edu.cn;

刘彦汝 (1998–), 女, 硕士生, 从事移动机器人调度算法的研究, E-mail: auliuyr@mail.scut.edu.cn;

杨启帆 (1997–), 男, 硕士生, 从事移动机器人导航与规划的研究, E-mail: yqfyokihang@163.com;

杨奕斌 (1999–), 男, 硕士生, 从事传感器定位与机器人的研究, E-mail: anchalyyb@163.com;

谢巍 (1974–), 男, 教授, 博士生导师, 从事智能机器人、机器视觉、鲁棒控制的研究, E-mail: weixie@scut.edu.cn.