

基于时空窗的多机器人协同路径规划算法研究

钟志峰, 张智浩, 易晓婕, 严诗玛, 孙玉坤

(湖北大学 人工智能学院, 湖北 武汉 430062)

摘要: 为进一步提高带窗口的层次化协作A*算法(WHCA*)进行多机器人路径规划时的效率, 提出减少邻域扩展的策略, 即重新定义机器人的邻域搜索范围以减少搜索空间、降低搜索时间, 进而提高算法效率。同时, 为了应对运行过程中出现的机器人失联状况, 提出以决策博弈为核心的改进WHCA*算法DG-WHCA*, 用于处理复杂场景下多机器人之间的碰撞和阻塞问题。在机器人移动配送系统仿真框架RAWSim-O下的仿真实验结果表明, 改进算法相较于原始WHCA*算法的搬运效率提高了3.14%, 行驶距离降低了2.17%; 在有机器人失联的情况下, 改进算法能有效避免碰撞与阻塞, 相较原始WHCA*算法的搬运效率提高了3.69%。

关键词: WHCA*; 邻域; 机器人失联; 决策树; 博弈论

DOI: 10.11907/rjdk.241729

中图分类号: TP301.6

文献标识码: A

开放科学(资源服务)标识码(OSID):

文章编号: 1672-7800(2025)009-0055-07



Research on Multi-robot Collaborative Path Planning Algorithm Based on Spatiotemporal Windowing

ZHONG Zhifeng, ZHANG Zhihao, YI Xiaojie, YAN Shima, SUN Yukun

(School of Artificial Intelligence, Hubei University, Wuhan 430062, China)

Abstract: To further improve the efficiency of the windowed hierarchical collaborative A* algorithm (WHCA*) in multi robot path planning, a strategy is proposed to reduce neighborhood extension, which involves redefining the robot's neighborhood search range to reduce search space and time, thereby improving the algorithm's efficiency. At the same time, in order to cope with the situation of robot disconnection during operation, an improved WHCA* algorithm DG-WHCA* with decision game as the core is proposed to handle collision and blockage problems between multiple robots in complex scenarios. The simulation experiment results under the RAWSim-O framework for robot mobile delivery system show that the improved algorithm improves the handling efficiency by 3.14% and reduces the driving distance by 2.17% compared to the original WHCA* algorithm; In the case of robot disconnection, the improved algorithm can effectively avoid collisions and blockages, with a 3.69% increase in handling efficiency compared to the original WHCA* algorithm.

Key Words: WHCA*; neighborhood; robot lost contact; decision tree; game theory

0 引言

智慧仓储中的移动机器人拣货系统(Robotic Mobile Fulfillment Systems, RMFS)是一种基于自动导引车(Automatic Guided Vehicle, AGV)的转送拣货系统^[1]。目前, RMFS中主要使用解耦型算法使其在无碰撞的情况下以最小总时间代价为多个机器人规划路径, 传统的解耦型算法

将RMFS简化为多个独立机器人的单机规划问题^[2-3]。代表性的解耦型算法是由Sharon等^[4]提出的冲突搜索(Conflict-Based Search, CBS)算法。该算法将问题巧妙地分解为冲突处理和单机路径规划两部分, 在大规模环境中表现出较好的性能。然而, CBS算法在处理复杂的冲突场景时会由于搜索空间的爆炸性增长而性能下降。为此, Chan等^[5]提出增强型冲突搜索(Enhanced Conflict-Based Search, ECBS)算法。该算法引入聚焦搜索策略, 通过在上

收稿日期: 2024-08-21

扫描二维码阅读全文:

基金项目: 湖北省技术创新专项(2018ACA13)

作者简介: 钟志峰(1971-), 男, 湖北大学人工智能学院教授, 研究方向为人工智能、信号处理、系统集成; 张智浩(1998-), 男, 湖北大学人工智能学院硕士研究生, 研究方向为仓储机器人路径规划算法设计。本文通讯作者: 张智浩。



下层搜索中添加 FOCAL 列表来缩小搜索范围,同时引入寻求全局最优解的思想,以提高多智能体路径规划的效率和性能。然而,ECBS 算法在面临大量障碍物和大量机器人时会因 FOCAL 列表管理的复杂性、全局最优解计算的困难性以及参数设置的不合理性导致性能快速下降。Likhachev 等^[6]提出的快速规划路径(Fast Attemptable Route Planner, FAR)算法能够在规划过程中动态调整和优化路径,但在初始参数设定不合理或存在较大偏差时会需要更长时间收敛,甚至无法找到最优解。Lam 等^[7]提出的偏执成本(Biased Cost Pathfinding, BCP)算法将多机器人路径规划问题分解为多个独立的子问题,每个子问题可以单独求解,且允许针对每个子问题采用不同的优化策略和方法。虽然该算法具有计算效率高、易于实现与调试的优点,但是存在找不到全局最优以及对子问题划分敏感的问题。Silver^[8]、Merschformann 等^[9]提出合作 A* 算法(Cooperative A*, CA*)、层次化合作 A* 算法(Hierarchical Cooperative A*, HCA*)以及带窗口的层次化协作 A* 算法(Windowed Hierarchical Cooperative A*, WHCA*)等系列算法将问题分解为一系列的单智能体搜索问题,以解决在狭窄通道和高智能体密度的复杂地图中搜索爆炸、僵局和循环重复的问题,但该类算法在每个时间步长内只考虑了一个窗口内的冲突,无法避免窗口交错时的冲突,从而影响了规划路径的质量。李珣等^[10]运用博弈论中的 Nash 均衡理论实现多搬运机器人的任务分配,但该方法存在计算复杂、对初始条件敏感以及难以适应动态环境变化的问题。郑延斌等^[11]通过博弈论构建多智能体动态避障模型,较好地解决了冲突位点的博弈问题,但仍存在容易陷入局部最优、算法复杂度高、同步通信困难以及难于扩展的问题。

针对以上问题,本文通过减少邻域搜索数量提高搜索效率;同时针对机器人失联的问题,引入决策树策略来诊断机器人的状态,运用谈判博弈策略为受影响的机器人重新分配路径节点,使用以决策博弈为核心的改进 WHCA* (Decision Game-Window Hierarchy Cooperative A*, DG-WHCA*) 算法重新进行路径规划。通过 RAWSim-0 仿真平台部署 32 个机器人进行 24 h 的仿真实验。结果表明,使用减少邻域扩展策略后,改进算法的搬运效率相较原始 WHCA* 算法提高了 3.14%,行驶距离降低了 2.17%。当系统中出现机器人失联时,DG-WHCA* 算法使其余机器人采取绕行措施以提高搬运效率,验证了其可行性和高效性。

1 理论基础

1.1 相关算法

CA* 算法是一种多智能体路径规划算法,其利用三维空间(包括时间维度)中的预约表来避免机器人冲突。如图 1 所示,CA* 算法根据优先级顺序为每个机器人进行路

径规划,并将已规划的路径信息作为约束条件加入到后续机器人的路径规划中^[12]。

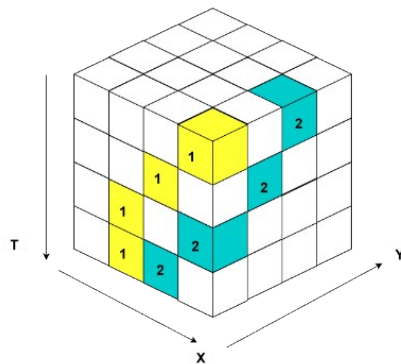


Fig. 1 Route planning with a time dimension

图 1 包含时间维度的路径规划

HCA* 算法通过在计算层使用 RRA* (Reverse Resumable A*) 算法执行反向搜索以获得更准确的启发值。表示为:

$$h^{HCA*}(w_s) = g^{RRA*}(w_s) \quad (1)$$

该算法在单机规划的同时考虑到其他机器人的路径信息,因此具有更高的求解效率,但其无法保证较好的求解结果,且求解成功率和路径质量对机器人的优先级顺序较为敏感^[13-14]。

WHCA* 算法则是通过限制协作搜索深度来减少计算量,随着搜索窗口的移动,代理计算新的部分路径,并且该算法仅在 w 步长内考虑其他代理的路径信息,以提高多代理协作的效率。此外,WHCA* 算法通过边缘代价函数来管理动态窗口,该函数通过错开窗口的方式允许在所有智能体之间的计算分散,从而有效平衡计算并优化路径规划过程。即:

$$\text{cost}(P, Q) = \begin{cases} 0 & \text{if } P = Q = G, t < w \\ \text{Dist}(P, G) & \text{if } t = w \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

总体来说,CA* 为多智能体路径规划提供了基本框架,通过三维预约表来避免冲突。HCA* 在 CA* 的基础上引入层次化抽象,提高了算法效率,适用于大规模或高密度的环境。WHCA* 在 HCA* 的基础上引入窗口机制,使得算法能够更好地适应动态环境,同时进一步提高了计算效率。

1.2 决策树

决策树是一种机器学习方法,具有树形结构,其中每个节点代表一个属性的判断,每个分支代表一个判断结果的输出,每个叶节点代表一种分类结果^[15]。决策树是常用的分类方法,其利用已知分类结果进行学习,从而对新数据进行准确分类。决策树的构建是一个递归的过程,从根节点开始通过选择最佳属性来分裂成子节点。该过程会一直持续,直到满足预设的停止条件^[16]。

1.3 谈判博弈论

谈判博弈构建了一个平台,使每个参与者通过策略性

谈判来争取分配有限的资源或利益。这一过程包含多种策略和决策,由参与者依次提出要求,目标是形成所有参与者均能接受的协议^[17]。根据博弈理论,采用 G 表示一个博弈, $G = \{B_1, \dots, B_n; U_1, \dots, U_n\}$ 。 G 有 n 个博弈方,博弈方集合表示为 $R = \{R_1, R_2, \dots, R_n\}$,每个博弈方 R_i 的全部可选谈判组合称为谈判空间,用 B_i 表示;博弈方的得益用 U_i 表示^[18]。

1.4 地图环境描述

本文采用的布局基于文献[19]中描述的生成器构建。如图2所示,该布局由3个不同区域构成,分别为:①库存区域。该区域包含所有托盘存储位置区域,用蓝色方块表示,下方路径点通过双向边与其他路径点相连,中间通道路径点通过单向边连接;②大厅区域。其是一个位于库存区域周围的区域,充当机器人从存储位置到车站的通道,有助于减少车站前的拥堵情况;③进站缓冲区域。该区域位于大厅区域内,主要用于缓冲机器人进入车站的过程,还包括所有补给站(黄色方块)和拾取站(红色方块)。

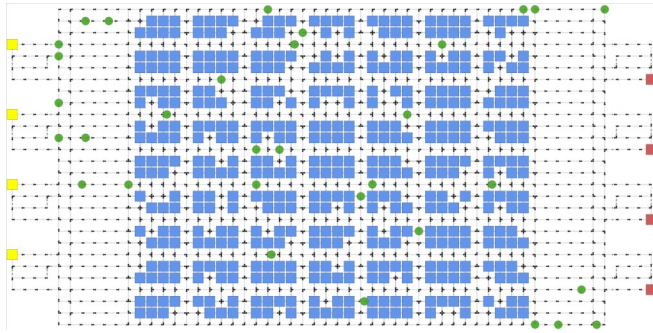


Fig. 2 Map layout of the mobile robot system

图2 移动机器人系统地图布局

2 算法构建

2.1 减少邻域扩展

在处理高密度场景中的多机器人问题时,WHCA*算法对机器人到达节点的时间有着严格要求。为提高算法效率,本文提出一种将4邻域扩展改为3邻域扩展的方法。利用图2地图布局,会始终至少留有一个机器人出口应对死锁现象,尽管有可能某条路径不是最优解,但对于提高总体效率影响非常小。根据目标点相对于当前节点所处的象限来选择要扩展的搜索邻域,即预先判断目标点相对于当前节点的位置,与目标节点相背的节点不计入计算。具体步骤为:根据函数 $M(1)$ 和 $M(2)$ 确定目标点相对于当前节点的位置,与目标节点相背的节点不计入计算。即:

$$M(1) = \{g_i.x - cur_i.x\} \quad (3)$$

$$M(2) = \{g_i.y - cur_i.y\} \quad (4)$$

式中: g_i 表示目标节点; cur_i 表示当前节点。

将 $M(1)$ 、 $M(2)$ 的值与0进行比较,将搜索邻域分为4个象限。表示为:

$$\text{第一象限: } M(1) \geq 0, M(2) \geq 0 \quad (5)$$

$$\text{第二象限: } M(1) \leq 0, M(2) \geq 0 \quad (6)$$

$$\text{第三象限: } M(1) \leq 0, M(2) \leq 0 \quad (7)$$

$$\text{第四象限: } M(1) \geq 0, M(2) \leq 0 \quad (8)$$

根据不同象限将传统的4邻域改为3邻域,第一、四象限只搜索向右和垂直方向的3个相邻节点,第二、三象限只搜索向左和垂直方向的3个相邻节点。如图3所示,目标节点 g 相对于当前节点 cur 的位置是在第一象限,因此只搜索向右和垂直方向的3个相邻节点。

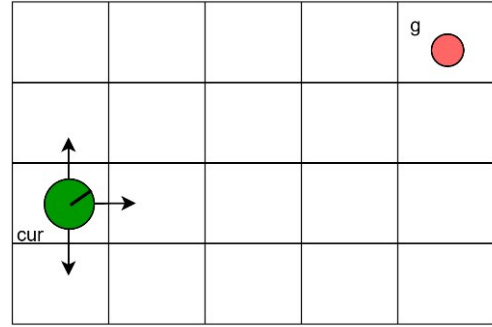


Fig. 3 Three-neighborhood expansion

图3 三邻域扩展

分别对三邻域WHCA*(LWHCA*)算法与WHCA*算法进行24 h的模拟,取10次实验的平均结果。表1为两种算法各参数的算术平均值。图4和图5分别为WHCA*算法和LWHCA*算法经过24 h模拟所呈现的热力图。可以看出,采用减少邻域策略不仅有效提升了机器人搬运托盘的数量,而且确保了机器人行驶路径不变。

Table 1 Parameter comparison of WHCA* algorithm and WHCA* algorithm

表1 LWHCA*算法与WHCA*算法参数比较

算法	处理托盘数/个	行驶总距离/m	拣补站休息时间占比/%
LWHCA*	4 701	657 272.010 8	44.5
WHCA*	4 558	671 548.073 3	45.8

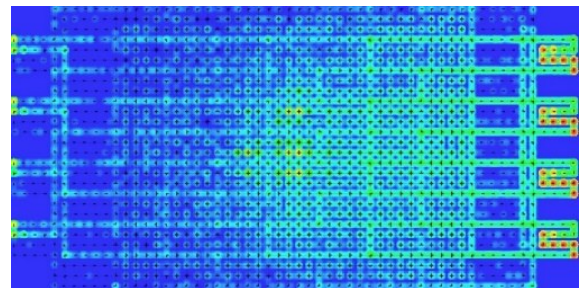


Fig. 4 Experimental heat map of the WHCA* algorithm

图4 WHCA*算法实验热力图

2.2 决策博弈

多机器人系统是一个高度集成的体系,由物理和虚拟部件构成,故障可发生在这一体系的任何组件中,从而对机器人运行产生负面影响^[20]。本文根据机器人实时信息,采用决策树策略诊断机器人是否失联。为了确保失联的机器人不会对其他正常运行的机器人造成影响,需立刻将失联的机器人停止运动,等待后续处理。同时,采用谈判

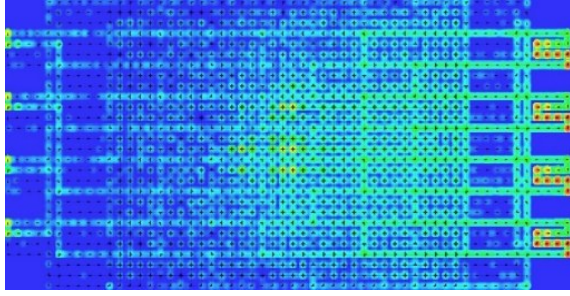


Fig. 5 Experimental heat map of the LWHCA* algorithm

图5 LWHCA*算法实验热力图

博弈为受影响的机器人分配利益,以更好地规划其路径。决策谈判博弈模型如图6所示。

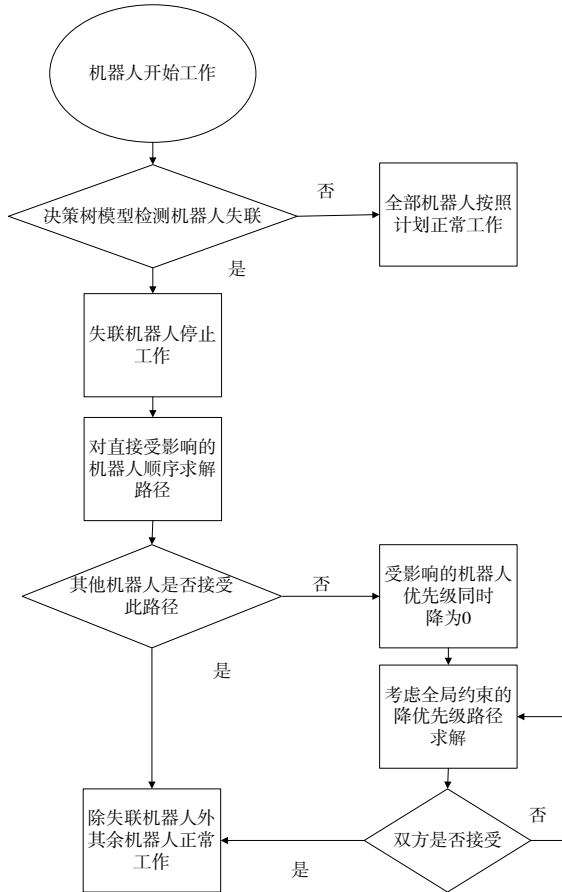


Fig. 6 Decision-making negotiation game model

图6 决策谈判博弈模型

2.2.1 决策树策略判定机器人状态

信息熵和信息增益是决策树算法中的两个基本概念,是用于构建决策树的重要指标。信息熵 $H(Y)$ 为用来衡量数据集不确定性的指标。计算公式为:

$$H(Y) = - \sum_{i=1}^N p(y_i) \log_2 p(y_i) \quad (9)$$

式中: $p(y_i)$ 为事件 y_i 发生的概率。信息熵越高,数据集的不确定性越大。

信息增益则是决策树算法中用来选择属性的一个标准,衡量的是在已知某个属性的信息后,数据集不确定性减少的程度。计算公式为:

$$\text{Gain}(Y, X) = H(Y) - H(Y|X) \quad (10)$$

式中: $H(Y|X)$ 为在已知随机变量 X 的条件下随机变量 Y 的条件熵。计算公式为:

$$H(Y|X) = - \sum_{j=1}^M p(x_j) \sum_{i=1}^N p(y_i|x_j) \log_2 p(y_i|x_j) \quad (11)$$

式中: $p(x_j)$ 为随机变量 X 取值 x_j 的概率; $p(y_i|x_j)$ 为在已知 $X=x_j$ 的条件下,随机变量 Y 取值 y_i 的条件概率。

在决策树构建过程中,属性选择对于机器人运行状态的准确分类与故障识别至关重要。本研究以信息增益作为核心评估指标,即原始信息熵与条件熵之差,优先选取信息增益高的属性作为决策树的划分节点,以降低系统状态判断的不确定性。

具体而言,研究设定五大判定准则作为决策节点划分依据:①将系统允许停止或故障停止的机器人定义为固定障碍物,用于触发路径规划中的避障策略;②以3 s为阈值,判断机器人命令执行响应能力,超时则视为处理异常;③监测机器人在节点的停留时间,超过5 s即判定为异常停留;④机器人之间每2 s进行一次信息交互,若连续3次超时未响应,则识别为通信链路异常;⑤以机器人与服务器每2 s一次的指令同步为基准,超时未恢复即判定为主从通信故障。根据以上信息训练出的决策树模型如图7所示。

2.2.2 谈判博弈为机器人分配利益

如图8所示,机器人 R_1 从 S_1 直行经节点 W_4 转弯向上前往 G_1 ,机器人 R_2 从 S_2 直行经节点 W_5 转弯向左前往 G_2 ,机器人 R_3 经决策树判定失联需立即停机。为防止机器人碰撞,本文采用谈判博弈论来解决冲突。在WHCA*算法顺序求解中,如果机器人遇到高优先级机器人占用节点时,将移除优先级约束,通过谈判博弈为受影响的机器人分配合理利益,然后采用LWHCA*算法重新规划路径。

由于机器人 R_1 的优先级高于 R_2 ,在顺序求解过程中机器人 R_1 无需考虑 R_2 的路径信息。首先,利用WHCA*算法对受影响的机器人进行顺序求解,得到 R_1 的路径 p_{10} ,并将其加入预约表。由于 R_2 的路径已经确定,导致其在该条件下与 R_1 在节点 W_n 处发生碰撞。因此, R_2 不接受 R_1 的 p_{10} 路径方案,转而选择删除 R_1 路径信息的方式进行求解,得到路径 p_{20} , R_2 以 p_{20} 对 R_1 进行还价。保留上述过程获得路径信息,进入谈判环节。

在谈判环节中,待求机器人集合 $R = \{R_1, R_2, \dots, R_n\}$ 不再仅考虑高优先级的路径预约信息,而是需要考虑除自身以外的包含其他所有机器人信息的全预约表。在所得全预约表的基础上, R_1 放弃原先的 p_{10} 路径方案,转而在考虑 R_2 的 p_{20} 基础上重新规划路径得到 p_{11} 。在该过程中, p_{20} 和 p_{11} 路径规划所用时间、计划路径长度和机器人是否载有托盘构成谈判空间 B ,由谈判立场组成。其中, $B = \{B_1, B_2, \dots, B_n\}$ 表示所有参与者的谈判立场。即:

$$B_i = (Faf, T_i, D_i, p_i) \quad (12)$$

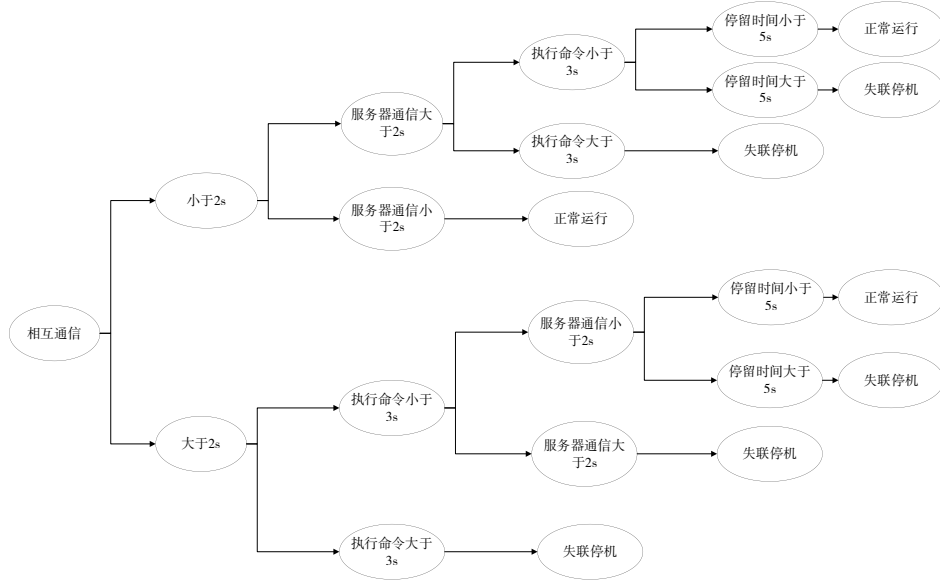


Fig. 7 Decision tree model

图7 决策树模型

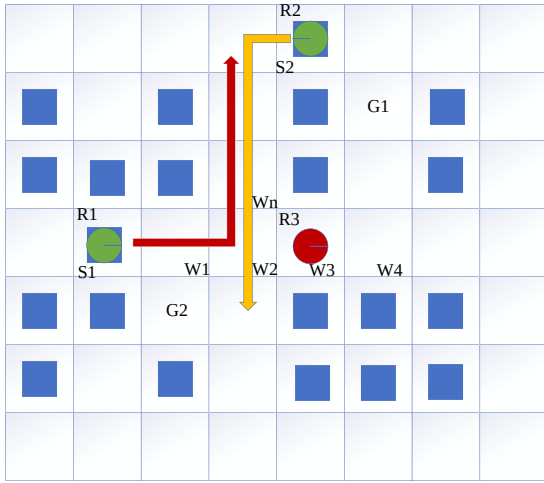


Fig. 8 Lost contact with robots causes crowding

图8 失联机器人造成拥挤

式中: Faf 为除第 i 个机器人以外的全预约表; T_i 、 D_i 、 p_i 分别为第 i 个机器人在 Faf 下的计划时长方案、路径长度方案、是否载有托盘方案。

效用函数 $U_i(B_i)$ 的公式为:

$$U_i(B_i) = \alpha * T_i(B_i) + \beta * D_i(B_i) + \delta p_i \quad (13)$$

计算出 R_1 机器人的效用,并计算协议中所有参与者的效用之和 $V(A)$ 。即:

$$V(A) = \sum_{B_i \in A} U_i(B_i) \quad (14)$$

然后, R_1 以 p_{11} 再次向 R_2 出价,根据纳什解 $\phi_i(A)$ 来公平分配协议效用。即:

$$\phi_i(A) = \prod_{b_i \in A} U_i(b_i) \quad (15)$$

更新每个参与者的谈判立场以达成协议,重复以上环节,选择效用价值最大的路径组合,得到最终结果 (p_{11}, p_{21}) 。新路径方案 p_{11} 考虑了 R_2 的行进路径,如图9所示。

示,机器人 R_1 会在路径点 W_1 等待机器人 R_2 离开后再经过节点 W_2 转弯,然后前往目的地。

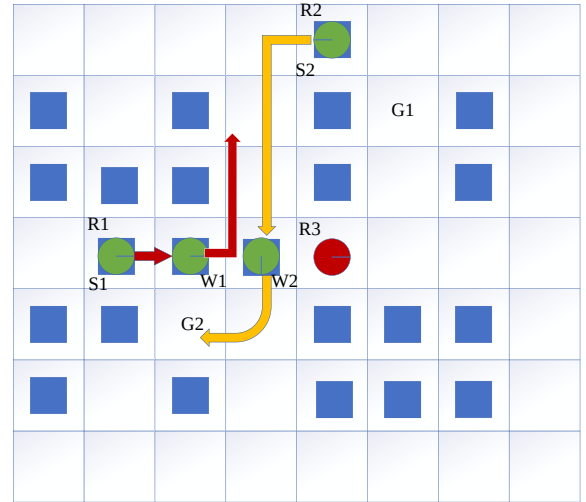


Fig. 9 Robot collision avoidance path diagram

图9 机器人避碰后路径图

3 实验方法与结果分析

3.1 实验方法

所有算法均在 Intel(R) Core(TM) i5-10210U CPU 处理器,内存为 12 GB 的 PC 机上以 C# 程序运行。

所有实验数据均为采用如图 2 所示的地图布局仿真运行 24 h 后的平均值。在使用 DG-WHCA* 算法进行路径规划时,比较 3 种不同场景下使用 32 台机器人仿真的效率,分别为机器人正常运行场景、某台机器人失联并停止运行 10 min 的场景、某台机器人失联并停止运行 20 min 的场景。同时,本文在使用 DG-WHCA* 算法进行路径规划

时,对系统在遭遇某台机器人失联并停止运行 10 min 的状态与正常运行状态下的实时内存消耗情况进行了比较。此外,使用 32 台机器人比较不同算法应对某台机器人失联并停止运行 10 min 和 20 min 等特殊场景时的实验结果。需要特别注意的是,在其他算法的实验中,当遇到类似情况时,所有机器人均会在相应的时间段内停止运行。

3.2 实验结果与分析

图 10 和图 11 分别为 DG-WHCA* 算法在 3 种场景下完成托盘数和行驶距离比较。可以看出,当一个机器人失联时,尽管 DG-WHCA* 算法能有效消除碰撞风险,但与正常情况相比,失联会迫使其他机器人绕路,从而增加行驶距离,降低订单处理量。

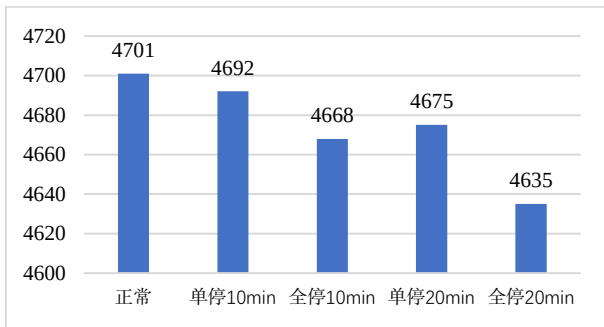


Fig. 10 Number of completed pallets

图 10 完成托盘数

图 12 为 DG-WHCA* 和 WHCA* 算法的实时内存消耗比较。可以看出,采用谈判博弈模块为受影响机器人分配利益时实时内存消耗激增,但经过 10 min 对失联机器人的处理后,实时内存逐渐回到 WHCA* 算法水平。

表 2 和表 3 分别为某台机器人失联 10 min 和 20 min 时

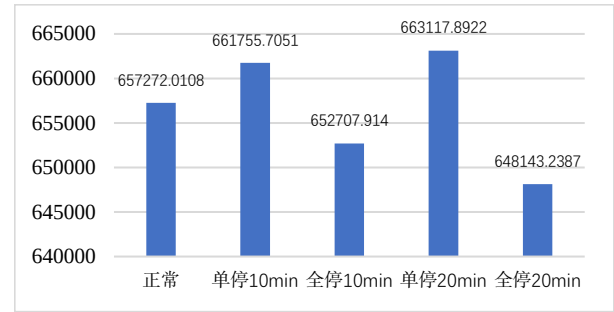


Fig. 11 Driving distance

图 11 行驶距离

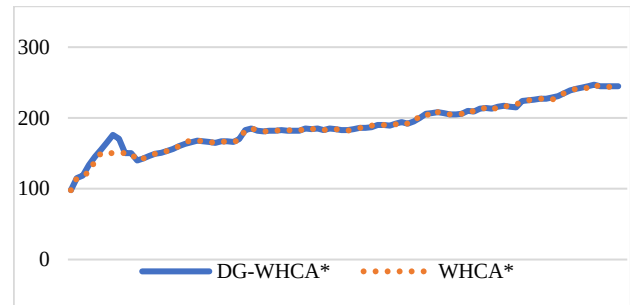


Fig. 12 Real time memory consumption

图 12 实时内存消耗

不同算法的仿真结果,与本文算法进行比较的对照算法为 CBS^[4]、FAR^[6]、OD&ID^[7]。可以看出,DG-WHCA* 算法在处理托盘个数、拣补站休息时间占比和处理命令数等方面均表现出明显优势,这主要得益于其分布式冲突消解策略和启发式搜索方法。相比之下,CBS 算法的集中式策略在面对机器人失联故障时表现较差,而 FAR 和 OD&ID 算法在路径规划的实时性和任务调度的灵活性方面也不如 DG-WHCA*。

Table 2 Simulation results of different algorithms when a certain robot goes missing for 10 minutes

表 2 某台机器人失联 10 min 时不同算法的仿真结果

算法	处理托盘数/个	行驶总距离/m	拣补站休息时间占比/%	处理命令数/个
DG-WHCA*	4 692	661 755.705 1	45.2	20 923
WHCA*	4 526	666 887.529 7	45.8	20 720
CBS	4 575	657 844.638 7	56.5	20 718
FAR	4 625	658 029.248 0	47.2	20 801
OD&ID	4 546	656 893.910 8	49.3	20 689

Table 3 Simulation results of different algorithms when a certain robot goes missing for 20 minutes

表 3 某台机器人失联 20 min 时不同算法的仿真结果

算法	处理托盘数/个	行驶总距离/m	拣补站休息时间占比/%	处理命令数/个
DG-WHCA*	4 675	658 117.892 2	44.5	20 853
WHCA*	4 532	648 143.238 7	45.1	20 611
CBS	4 543	653 241.329 4	54.2	20 573
FAR	4 593	653 424.646 8	43.8	20 656
OD&ID	4 514	652 297.254 2	45.3	20 553

4 结语

DG-WHCA* 算法显著提升了高密度场景中多机器人的处理效能,通过缩减搜索邻域降低了时间复杂度,同时

提高了托盘搬运效率。通过结合决策树与谈判博弈机制,该算法能精确判断机器人状态并有效分配资源,解决了机器人之间的冲突问题。相较于 WHCA* 和 CBS 等算法,DG-WHCA* 算法在托盘处理数、命令处理数、行驶总距离等指标方面展现出一定优势,验证了其有效性。然而,该

算法在极端高密度环境以及大量机器人协作时仍面临挑战,会出现复杂度高、资源消耗大和鲁棒性不足等问题。未来研究将聚焦于降低算法复杂度、开发高效资源管理策略、综合利用多种传感器数据和分布式计算方法,以提升算法的实时性、效率和可靠性,进一步优化多机器人系统在复杂环境下的协作性能。

参考文献:

- [1] XU X B, MA Z Q. Research progress of picking system based on mobile robot[J]. *Acta Automatica Sinica*, 2022, 48(1): 1-20.
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*, 2022, 48(1): 1-20.
- [2] WAGNER G, CHOSSET H. Subdimensional expansion for multirobot path planning[J]. *Artificial Intelligence*, 2015, 219: 1-24.
- [3] SHARON G, STERN R, FELNER A, et al. Meta-agent conflict-based search for optimal multi-agent path finding[C]//*Proceedings of the International Symposium on Combinatorial Search*, 2012: 97-104.
- [4] SHARON G, STERN R, FELNER A, et al. Conflict-based search for optimal multi-agent pathfinding[J]. *Artificial Intelligence*, 2015, 219: 40-66.
- [5] CHAN S H, LI J, GANGE G, et al. ECBS with flex distribution for bounded-suboptimal multi-agent path finding[C]//*Proceedings of the International Symposium on Combinatorial Search*, 2021: 159-161.
- [6] LIKHACHEV M, GORDON G J, THRUN S. ARA*: anytime A* with provable bounds on sub-optimality[C]//*Proceedings of the 17th International Conference on Neural Information Processing Systems*, 2003: 767-774.
- [7] LAM E, LE BODIC P, HARABOR D, et al. Branch-and-cut-and-price for multi-agent path finding[J]. *Computers & Operations Research*, 2022, 144: 105809.
- [8] SILVER D. Cooperative pathfinding[C]//*Proceedings of the 1st Artificial Intelligence and Interactive Digital Entertainment Conference*, 2005: 117-122.
- [9] MERSCHFORMANN M, XIE L, ERDMANN D. Multi-agent path finding with kinematic constraints for robotic mobile fulfillment systems[DB/OL]. <https://arxiv.org/pdf/1706.09347v2>.
- [10] LI X, NAN K K, ZHAO Z F, et al. Task allocation of handling robot in textile workshop based on multi-agent game theory[J]. *Journal of Textile Research*, 2020, 41(7): 78-87.
李珣, 南恺恺, 赵征凡, 等. 多智能体博弈的纺织车间搬运机器人任务分配[J]. *纺织学报*, 2020, 41(7): 78-87.
- [11] ZHENG Y B, WANG L L, XI P X, et al. Multi-agent path planning algorithm based on ant colony algorithm and game theory[J]. *Journal of Computer Applications*, 2019, 39(3): 681-687.
郑延斌, 王林林, 席鹏雪, 等. 基于蚁群算法及博弈论的多Agent路径规划算法[J]. *计算机应用*, 2019, 39(3): 681-687.
- [12] ERDMANN M, LOZANO-PEREZ T. On multiple moving objects[J]. *Algorithmica*, 1987, 2: 477-521.
- [13] SHARON G. Novel search techniques for path finding in complex environment[D]. Beersheba: Ben-Gurion University of the Negev, 2015.
- [14] 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.
- [15] XIE X, ZHANG X Y, WANG X Y, et al. Neighborhood decision tree construction algorithm for variable precision neighborhood equivalent grains[J]. *Journal of Computer Applications*, 2022, 42(2): 382-388.
谢鑫, 张贤勇, 王旋晔, 等. 变精度邻域等价粒的邻域决策树构造算法[J]. *计算机应用*, 2022, 42(2): 382-388.
- [16] KOTSIANTIS S B. Decision trees: a recent overview[J]. *Artificial Intelligence Review*, 2013, 39: 261-283.
- [17] ZHANG K X, MAO J L, XIANG F H, et al. B-IHCA* multi-robot path planning algorithm based on bargaining game mechanism[J]. *Acta Automatica Sinica*, 2023, 49(7): 1483-1497.
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.
- [18] WEI T, LONG C. Path planning of mobile robot based on improved genetic algorithm[J]. *Journal of Beijing University of Aeronautics and Astronautics*, 2020, 46(4): 703-711.
魏彤, 龙琛. 基于改进遗传算法的移动机器人路径规划[J]. *北京航空航天大学学报*, 2020, 46(4): 703-711.
- [19] LAMBALLAIS T, ROY D, DE KOSTER M B M. Estimating performance in a robotic mobile fulfillment system[J]. *European Journal of Operational Research*, 2017, 256(3): 976-990.
- [20] KHALASTCHI E, KALECH M. Fault detection and diagnosis in multi-robot systems: a survey[J]. *Sensors*, 2019, 19(18): 4019.

(责任编辑:尹晨茹)