

基于改进贪婪式算法的 AMR 任务分配

谢 进¹, 向 勇^{2,3}, 杨秀清^{2,3}, 周新志¹

(1. 四川大学电子信息学院, 成都 610065; 2. 中国民航局第二研究所, 成都 610065;
3. 民航成都物流技术有限公司, 成都 610065)

摘 要: 采用自主移动机器人 AMR(Autonomous Mobile Robot)集群智能、高效处理机场行李时,为了解决机场环境中 AMR 集群的分配调度问题,提出一种改进贪婪式算法的任务调度策略. 根据随机行李数量,分配合适的 AMR 数量执行处理任务. 该算法综合考虑在机场环境下行李任务的到达规律和 AMR 特性,据此改进贪婪选择策略,使其较其他算法更好体现行李任务与 AMR 之间的调度分配关系. 首先,采用 A * 算法计算代价,能够获得更加符合实际环境的代价值;其次,对 AMR 进行类型划分和使用预先出发的策略,减小了任务分配时间和系统运行时间. 仿真结果表明,该算法与相关文献算法相比,能够获得更小的任务分配时间和系统运行时间.

关键词: 任务分配; AMR; 贪婪算法; 机场行李

中图分类号: TP242 **文献标识码:** A **DOI:** 10.19907/j.0490-6756.2021.042003

AMR task allocation based on improved greedy algorithm

XIE Jin^{1,2}, XIANG Yong^{2,3}, YANG Xiu-Qing^{2,3}, ZHOU Xin-Zhi¹

(1. College of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China;
2. The Second Research Institute of Civil Aviation Administration of China, Chengdu 610065, China;
3. Civil Aviation Chengdu Logistics Technology Company Limited, Chengdu 610065, China)

Abstract: When using Autonomous Mobile Robot (AMR) cluster to process airport baggage intelligently and efficiently, in order to solve the allocation and scheduling problem of AMR cluster in the airport environment, a task scheduling strategy with an improved greedy algorithm is proposed. According to the random baggage quantity, the appropriate AMR quantity is allocated to perform the processing task. The proposed algorithm comprehensively considers the arrival rules and AMR characteristics of baggage tasks in an airport environment, and accordingly improves the greedy selection strategy, which making it better than other algorithms to reflect the scheduling and allocation relationship between baggage tasks and AMR. Firstly, the A * algorithm is used to calculate the cost, which can obtain a substitute value that is more in line with the actual environment. Secondly, the type division of AMR and the use of advance departure strategies reduce the task allocation time and system runtime. Simulation results show that the algorithm can obtain at least 8.9% improvement in system runtime compared with the greedy algorithm.

Keywords: Task allocation; AMR; Greedy algorithm; Airport baggage

收稿日期: 2020-09-05
基金项目: 国家自然科学基金委员会-中国民用航空局民航联合研究基金(U1933123)
作者简介: 谢进(1995—), 男, 贵州正安人, 硕士研究生, 主要研究方向为模式识别与智能系统. E-mail:1639329432@qq.com
通讯作者: 周新志. E-mail: xz.zhou@scu.edu.cn

1 引言

随着全球民航业旅客流量的日益增长,机场行李处理量随之日渐攀升,据国际航空电讯集团(SITA)2020 年行李 IT 洞察报告^[1]可知,2019 年全球航空业共运输了 45.4 亿人次旅客及行李,行李处理量在 20 亿件以上,2018 年和 2019 年中国民航业分别完成旅客运输量 6.1 亿人次和 6.6 亿人次,旅客托运行李量均超过了 3 亿件.对机场行李进行安全、智能、高效、稳定的处理不仅对航空安全、航班准点率和旅客满意度具有举足轻重的意义,而且对我国的民航运输服务品质的提升和民航建设具有重大的促进作用.

民用机场自建成以来,国内外众多科研院所和学者针对机场行李处理技术的研究从未停止过.截至目前,虽然我国机场行李处理系统已历经了三代技术与更迭,历经了近 40 年、但是为满足未来“四型机场”建设需求、进一步提升民航运输服务品质以及旅客日益增长的个性化需求,机场行李处理技术的智能化水平应得到进一步提升,紧跟未来智能化、智慧化的发展步伐,而现有的行李处理技术的智能化程度已很难满足如上所述需求.目前,虽有与机场行李处理领域相似的快递物流和仓储领域已开研究并应用于自动导航小车 AGV(Automatic Guided Vehicle)技术的包裹或仓库货运处理系统,但是 AGV 通常采用二维码、电磁和磁带等导航技术,按照预设规划的路径完成各个任务点之间分货物运送,这需要在行驶路线上分别预布二维码标签、金属导线和磁带,这就使得其灵活性差,智能化程度低,还会造成大量资源浪费,而且很难满足如上所述多方面需求.

因此,需要研究具备自主能力更强,智能化程度更高,面向未来机场的行李处理技术.基于 AMR 集群的行李处理技术具备较高的智能性、扩展性和灵活性,因为 AMR 通常采用激光与其他传统技术相结合的导航方式,不再需要在地面预布二维码,金属导线和磁带等地标,其自身具备强大的计算能力且能够感知周围环境并做出相应决策.所以可根据机场实时变化的行李量而实时调度不同集群规模的 AMR 小车,高效、智能、灵活地完成行李处理.在机场行李处理技术中,关于 AMR 小车的任务分配与调度是其核心问题之一,其含义主要是指在行李动态抵达和 AMR 小车位置持续变化的环境中,为 AMR 集群分配合适任务,使得行李

任务处理系统的运行时间达到最小.而如何实现行李的安全、稳定、高效的处理对航空安全、航班准点率和旅客满意度具有举足轻重的重要意义.本文针对 AMR 集群的任务分配与调度问题提出了一种适用于机场环境的任务分配与调度策略,以缩短任务分配时间和系统运行时间.

2 相关工作

目前,针对多任务分配与调度系统的研究备受国内外学者的关注. Banziger 等^[2]提出了一种将仿真作为遗传算法中的适应度函数来优化机器人团队任务分配的新方法,该方法不仅可以评估任务的不同分布,也可以考虑工人和机器人在共享工作场所的交互动态. Xia 等^[3]提出了一种解决地面运动目标多无人机协同任务分配和航迹规划问题的系统框架.该方案不仅能有效地规划出合理的航迹而且能解决不确定性问题,得到最优的任务分配方案. El-Ashmawi 等^[4]基于 SSA 提出了一种 MSL-RH 算法用于解决任务分配问题,在树状结构数据集中,该算法的最小平均分配成本比遗传算法降低了 62%,比 PSO 和 JAYA 降低了 42%. Li 等^[5]提出了一种基于改进的灰太狼优化算法(IGWO)的分布式协同任务分配策略.先将 MRTA 问题转化为多个旅行商问题,然后利用 IGWO 求解多个 TSP 问题的最优解.最后,对最优解空间进行积分,得到 MTSP 的最优解.其实验结果表明,IGWO 算法具有较快的收敛速度和较高的精度. Zhang 等^[6]针对多 AGV 系统存在着资源分配、冲突和死锁等一系列问题,采用时间窗法建立了多 AGV 任务分配系统,能有效地解决 AGVs 系统的冲突问题,具有较高的稳定性和实时性. Chen 等^[7]对移动吊车与 AGV 集成调度问题作为一个多机器人协调调度问题进行研究,提出了一种具有两组流量平衡约束的起重机和 AGV 多商品网络流模型,有效地实现了 AGV 与起重机在自动化终端中的精确协调. Dang 等^[8]提出了一种将遗传算法与禁忌搜索相结合的混合启发式算法来解决考虑完工期最小化的问题. Mousavi 等^[9]通过建立数学模型并和进化算法相结合,在考虑 AGV 电池电量的情况下,以最小化 AGV 的操作时间和数量为目标,对 AGV 的任务调度进行了优化. Tang 等^[10]在传统的先来先服务调度算法上利用帕累托空间搜索,解决可预测内存层次的芯片上实时流应用程序的调度问题. Alworafi 等^[11]在短作业调度算法的

基础上进行改进, 极大的缩短了云计算环境中最后一个任务的完成时间与平均响应时间, 实现虚拟机之间的负载均衡。王鑫等^[12]在考虑云环境下任务与虚拟机资源的特征, 改进了贪婪选择策略获得了更好的执行效率和负载均衡能力。Tong 等^[13]提出了离线预测结合在线分配的任务分配算法 POLAR-OP 提升了在空间纵包领域下的任务匹配对数; 桑泽磊等^[14]针对传统合同网协议 CNP (Contract net protocol) 存在协商频繁和投标并发操作问题, 提出了一种基于节拍的改进合同网协议, 缩短了 AGV 完工时间并提升了机床、AGV 利用率。严飞等^[15]针对多无人机协同搜索和攻击作战中, 考虑存在相互耦合的任务分配的问题, 提出了一种获得最大系统效能的任务分配算法。张梦颖等^[16]针对无人机群协同实时任务分配问题, 在 CNP 的基础上, 将招标者参与投标和引入并发机制对 CNP 做出了改进, 减少了通信量、运算量和拍卖回合。

尽管目前还未有将 AMR 小车应用于机场行李处理领域相关的研究报道, 但是在上述文献中针对多任务分配与调度系统的研究对 AMR 小车的任务分配与调度研究具有重要借鉴意义。

3 问题描述

3.1 场景模型

机场的行李处理系统中, AMR 集群可应用于机场值机岛、开包间和分拣等局部场景, 也可应用于机场行李处理整体场景。本文以值机岛场景应用 AMR 集群为例, 在值机岛场景中, 旅客行李从值机柜台进入, 由 AMR 小车托运至集中安检区进行安检, 然后再将安全行李运输至行李分拣口, 为了方便描述, 可将 AMR 值机岛场景描述为图 1 所示模型。

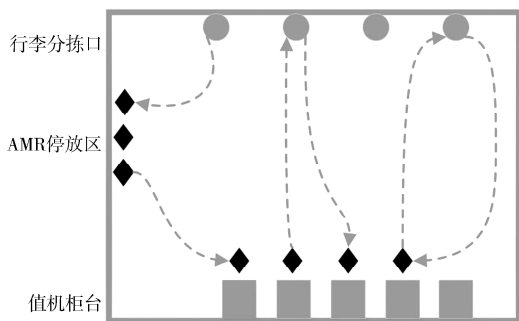


图 1 场景模型

Fig. 1 The scene model

图 1 所示模型主要由值机柜台区, AMR 停放

区, 行李分拣口区及值机柜台区和行李分拣口区之间的运行区组成, 图 1 中菱形代表 AMR 小车, 方形代表值机柜台, 圆形代表行李分拣口区。当行李到达值机柜台后, 在停放区的 AMR 运行到达值机柜台接上行李之后, 需要根据行李的目标位置将其安全运送至行李分拣口之一, 然后该 AMR 再根据值机柜台的行李任务情况选择返回停放区还是直接返回值机柜台继续接取新的行李。

3.2 相关定义

本节对 3.1 节的行李任务和 AMR 小车做出相关定义, 设 t_i 为任务集合 T 中的任务, $t_i \in T$, 对单个任务 t_i 的定义为

$$t_i = \langle taskId, t_s, l_s, l_e \rangle$$

其中, $taskId$ 代表该任务的编号; t_s 代表任务到达值机柜台的时间; l_s 代表任务开始的位置, 即任务在某个值机柜台的位置; l_e 代表任务结束的位置, 即任务在最终到达行李分拣口的某个位置。

设 w_i 为 AMR 集合 W 中一台 AMR, $w_i \in W$, 对单个 AMR 的定义为

$$w_i = \langle workId, l_c, C_f, C_a \rangle$$

其中, $workId$ 代表该小车的编号; $l_c = \langle x, y \rangle$, 代表该 AMR 当前的位置; x 和 y 分别代表横纵坐标; C_f 代表该 AMR 接取新任务的距离代价; C_a 代表该 AMR 执行完自身当前任务的代价, 如果 AMR 没有在执行行李运输任务, 则 $C_a = 0$ 。每轮分配任务时需要更新计算 C_f 和 C_a 的值。

3.3 任务匹配

在机场环境中, 任务集合 T 中的任务并不是所有任务一起到达值机柜台, 而是在不同的时间点到达, 所以任务分配需要在每轮任务到达后执行, 这是一个多轮的任务分配问题, 直到任务集合中的最后的任务都到达值机柜台, 完成任务分配。用 WT 表示总任务分配集合, 假设任务集合 T_i 是 T 的子集合, 即表示第 i 轮任务分配的任务数量, 设 WT_i 表示第 i 轮的任务分配集合, 那么每个分配可表示为 $WT_i = \langle w_j, t_k \rangle$, 其中, $w_j \in T$, $t_k \in T_i$; w_j 代表 j 个 AMR; t_k 代表第 k 个任务, 在每轮任务分配中, 每个 AMR 允许分配得到 T_i 中的多个任务, 任务分配一旦完成, 就不会发生改变, 由 AMR 完成自身任务。

4 任务分配

AMR 集群系统可视为一个随机服务系统, 行李处理任务的到达流可用泊松流^[17]描述, 其到达

时间间隔服从参数为 λ 的负指数分布,而服务时间服从参数为 μ 的负指数分布. 本文提出一种基于改进贪婪式的 AMR 小车任务分配与调度策略. 当任务到达值机柜台之后,首先考虑到机场环境中存在障碍物和 AMR 小车位置动态发生变化的问题,采用 A^* 算法计算 AMR 小车接取任务的代价并且根据任务动态更新. 其次考虑到任务到达规律和 AMR 小车规模对任务分配和调度的影响,对 AMR 小车进行类型划分和使用预先出发的策略.

4.1 代价计算

本文中 AMR 小车的接取任务的代价 C_i 使用 A^* 算法计算,这是考虑到在场景中存在障碍物的问题,如果采用常规的距离算法如曼哈顿距离,欧氏距离计算,将会对任务分配的准确性有着较大的影响. 如图 2 所示,图中黑色条形为障碍物,不可通过. 当任务出现在位置 $A(0,10)$ 的时候,在位置 $B(22,16)$ 和位置 $C(17,3)$ 分别有一辆 AMR 小车参与该任务的分配,此时如果采用欧氏距离公式计算如下式.

$$d = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$
 (1)

根据式(1)可计算位置 $B(22,16)$ 与 A 位置 $(0,10)$ 的欧式距离 $d_{AB} = 22.8$,位置 C 与 A 的欧式距离 $d_{AC} = 18.4$. 显然如果按照欧氏距离来计算代价,那么将由位置 C 处的小车获得该任务,但是由于图中存在障碍物,小车是不可通过路径 AC 的. 而采用 A^* 算法估算路径时,路径长度计算如下式.

$$d = \sqrt{2} * n_1 + n_2$$
 (2)

式中, n_1 代表 A^* 算法斜向步数; n_2 代表横向或纵向步数. 根据式(2), AB 路径长度为 $d_{AB} = 24.5$, AC 的路径长度 $d_{AC} = 29.9$,显然应当由 B 处的小车获取该任务. 因此在实际情况中,采用 A^* 算法估计代价更加符合实际需求.

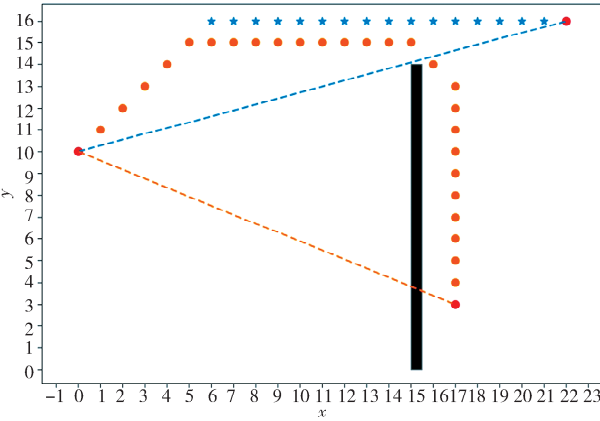


图 2 代价计算
Fig. 2 The cost calculation

4.2 任务分配算法

贪婪算法在任务分配与调度问题中是一种常见的策略. 贪婪算法总是做出在当前看来最好的选择,算法的关键是贪婪策略的选择. 算法应用在本文中,即每当任务到来时,贪婪算法总是从所有 AMR 中选择接取任务代价最小的 AMR,直至所有的任务分配完毕. 但每次任务到来时,总是从所有 AMR 中选择会使得计算量较大. 因此在改进策略中,根据 AMR 不同运行状态的特点,对 AMR 进行类型划分,第一种是空闲的 AMR 小车即停放在 AMR 停放区的小车,用 $flag=0$ 来标识;第二种是执行完当前任务正在返回 AMR 停放区的小车,用 $flag=1$ 来标识;第三种是正在执行任务的 AMR 小车用 $flag=2$ 来标识. 在计算代价 C_i 可以直观地知道在三种类型的小车中,第一种小车的代价 C_i 小于另外两种类型的小车的 C_i ,第二种小车的代价 C_i 小于第三种代价的小车. 当任务到来的时候,不需要所有小车全部参与分配而是分批次考虑参与该轮任务的分配,优先考虑 $flag=0$ 的小车参与分配,接着考虑 $flag=1$ 的小车参与分配,最后才考虑 $flag=2$ 的参与分配. 采取这样分批次参与分配的方案可以有效减少无用小车的代价估算,降低计算量. 此外,由于 AMR 集群系统中行李处理任务的到达流为泊松流,到达时间间隔服从参数为 λ 的负指数分布,而服务时间服从参数 μ 为负指数分布. 因此任务的到达是可预估的. 由于可以预估到下一轮任务的到达,如果系统存在空闲的小车就可以提前出发到值机柜台接取任务,对于 $flag=1$ 的小车可以不用返回停放区直接前往服务区接取任务,这种模式甚至有可能做到车等任务的效果,能够有效减少系统运行时间.

因此,本文在基于任务泊松流到达的基础上,对 AMR 小车的代价估算使用更加符合实际应用场景的 A^* 算法计算,并且采用分批次 AMR 小车的任务分配和 AMR 小车提前出发的策略,能够有效缩短任务分配时间和系统运行时间. 算法如图 3 所示.

图 3 中 $list0$, $list1$, $list2$ 分别对应存放 $flag=0$, $flag=1$, $flag=2$ 类型的小车,当新任务到来. 先判断 $list0$ 中是否为空,如果不为空就直接为 $list0$ 中的 AMR 小车分配任务并执行任务; $list0$ 如果为空就判断 $list1$ 是否为空, $list1$ 不为空就为 $list1$ 中的 AMR 小车分配任务并执行;如果 $list1$ 仍然为空,那么就从 $list2$ 中分配任务并执行. 新一轮任

务到来之前,空闲的 AMR 小车可预先出发前往值机柜台区等待,没有空闲的 AMR 小车那就不做处理.当没有新任务到来后,算法结束.

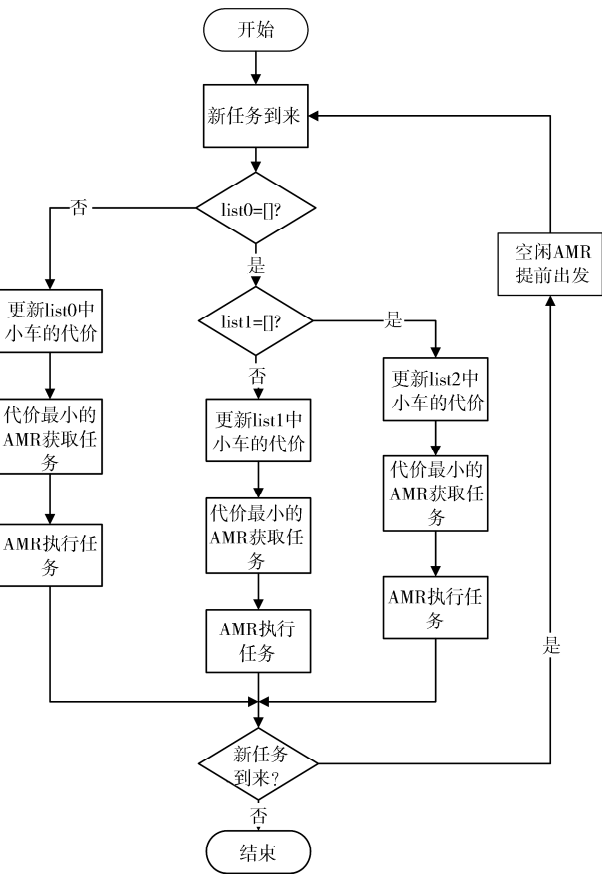


图 3 算法流程
Fig. 3 The algorithm flow

5 仿真实验

为了验证本文所提算法的有效性,使用任务分配时间和系统运行时间作为评价指标,通过与 SimpleGreedy,节拍合同网协议^[14]和 GR 分配算法^[18]进行对比实验研究. SimpleGreedy 采取的策略是当任务到达值机柜台之后,所有 AMR 小车参与该轮的任务分配. GR 采取的策略是任务到达之后不会立即分配,而是设定一个时间片,收集该时间片的任务数量之后再进行任务分配.本文进行了两组实验,第一组仿真实验对比分析不同算法获得的任务分配时间;第二组仿真实验对比分析不同算法获取的系统运行时间.本文仿真环境基于 Python3.7.4,在处理器为 3.2 GHz Inter(R) Core (TM) i5-4460,内存为 16 G 的计算机上运行.

实验中假设场景大小为 40 × 30 的网格,任务按照泊松流到达,任务的开始位置随机指定为某一

值机柜台的坐标,任务的终点位置随机指定为行李分拣口之一终点坐标. AMR 小车速度设置为 2 m/s,初始位置坐标为(20,0).分析 AMR 小车数量变化,任务数量变化及任务到达时间长度变化对任务分配时间的影响.分析 AMR 数量变化时,AMR 数量从 8~16 个变化,任务数量设置为 103 个,任务到达时间长度设置为 5 min,按照泊松流方法 103 个任务在 5 min 内的不同时刻到达;分析任务数量变化时,任务数量从 34~122 个变化,AMR 数量设置为 10 个,任务到达时间长度为 6 min,任务按照泊松流到达;分析任务到达时间长度时,时间长度从 6.7~20.2 min 变化,任务数量设置为 100 个,AMR 数量设置为 10 个,任务按照泊松流到达.

5.1 任务分配时间实验

本节对比分析 4 种算法获得的任务分配时间.图 4 表示 AMR 数量变化时,4 种算法的所耗去的任务分配时间.从图 4 中可以看出,随着 AMR 数量的增加,SimpleGreedy,GR 和节拍合同网协议(CNP)的任务分配时间都持续增大,这是由于每轮分配任务过程中,所有 AMR 都参与了分配,而随着任务数量增加,每轮任务分配时有更多的 AMR 需要计算接取代价,从而增加了分配时间.值得注意的是,当 AMR 数量增加时,本文所提的算法任务分配时间变化不大,任务分配时间较另外 3 种算法都更小.这是由于在算法中将 AMR 类型进行分类,每次进行任务分配时是根据 AMR 的自身情况决定是否参与分配任务的,如果有空闲的 AMR,空闲的 AMR 会参与任务分配,而其他的 AMR 不会,此时由参与任务分配的 AMR 决定分配时间.

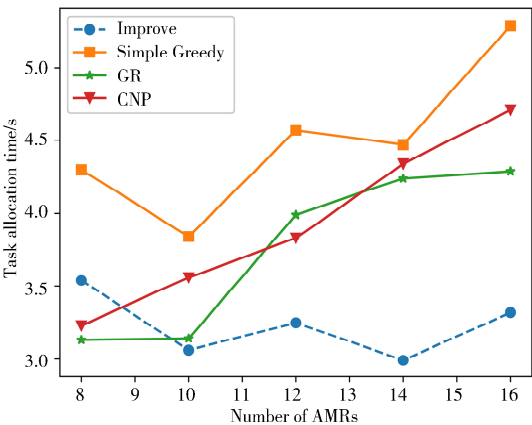


图 4 AMR 数量变化时的任务分配时间对比
Fig. 4 The comparison of task allocation time with the change of AMRs number

图 5 表示当任务数量变化时,4 种不同算法的分配时间对比情况。从图 5 中可以看出,随着任务数量增加,4 种算法的任务分配时间都继续增加,这是显而易见的。但是可以观察到本文所提的算法与另外 3 种算法的分配时间在逐渐接近。这是由于任务到达时间长度固定为 6 min,随着任务数量增加,任务到达的密集程度必然增加,AMR 处理任务的速度无法跟上,那么累积在 flag=2 类型的 AMR 较多,在分配任务时则会导致参与任务分配的 AMR 数量增加,从而任务分配时间增大。

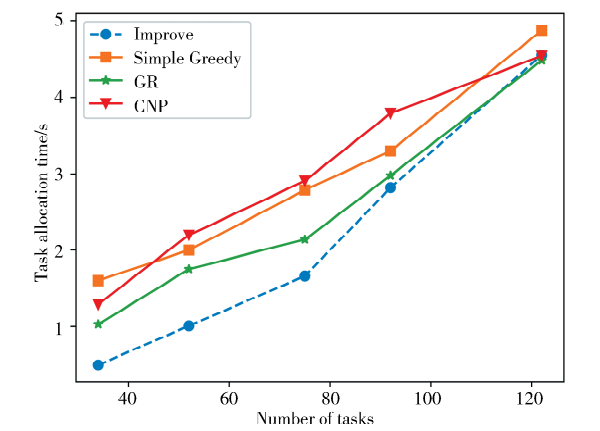


图 5 任务数量变化的任务分配时间对比
Fig. 5 The comparison of task allocation time with the change of tasks number

图 6 表示时间变化时,任务分配时间的对比。类似的情况,由于 AMR 数量和任务数量固定,对于 SimpleGreedy 和 GR 和节拍合同网协议(CNP)来说,任务分配时间变化不大,但是本文所提的算法分配时间先减小,然后又增大。刚开始任务到达密集程度高,累积在 flag=2 类型的 AMR 较多,而最后任务到达密集程度低,累积在 flag=0 类型的 AMR 较多,所以参与任务分配的 AMR 数量是一个由大变小,再由小变大的过程。

5.2 系统运行时间实验

本节对比分析 4 种算法的获取的系统运行时间。图 7 表示 AMR 数量变化时 4 种算法的系统运行时间。从图 7 中可以看出,随着 AMR 数量增加,4 种算法的系统运行时间都持续减小。当 AMR 数量大于等于 10 个后,本文所提出的算法的运行时间明显要小得多,如表 1 所示较之于节拍合同网协议最差有 7.6%的提升。这是由于当 AMR 数量小于 10 个时,任务到来比较密集,较多 AMR 会处于一个比较繁忙的状态,在值机柜台与分拣口之间来回运输行李。这点从系统运行时间上也可以看出,

行李任务是在 5 min 之内到达的,但 AMR 数量为 8 个时,系统处理完任务最快的 Improve 差不多也要 750 s。

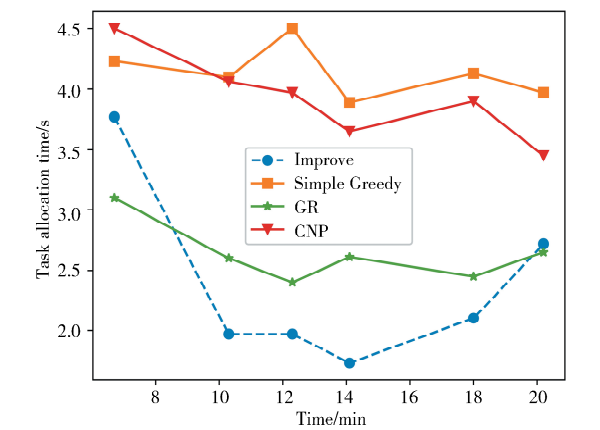


图 6 时间变化的任务分配时间对比
Fig. 6 The comparison of task allocation time with time change

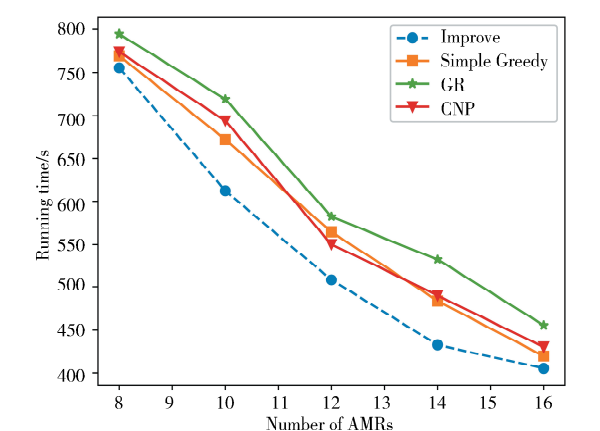


图 7 AMR 数量变化时的运行时间对比
Fig. 7 The comparison of system running time with the change of AMRs number

表 1 AMR 数量变化的运行时间提升量			
Tab. 1 The running time increase of AMRs number change			
AMRNum	Improve/s	CNP/s	Increase/%
10	612.5	694.2	11.8
12	508.3	550.0	7.6
14	436.8	490.0	10.9

图 8 表示任务数量变化时 4 种不同算法的系统运行时间。从图 8 中可以看出,随着任务数量增加,任务到达的密集程度必然增加,系统运行时间必然增加,但本文提出的算法任务数量大致在 55~100 之间系统运行时间能够获得较大提升,如表 2 所示,本文所提算法较之于节拍合同网协议(CNP)运行时间最差有 8.4%的提升。

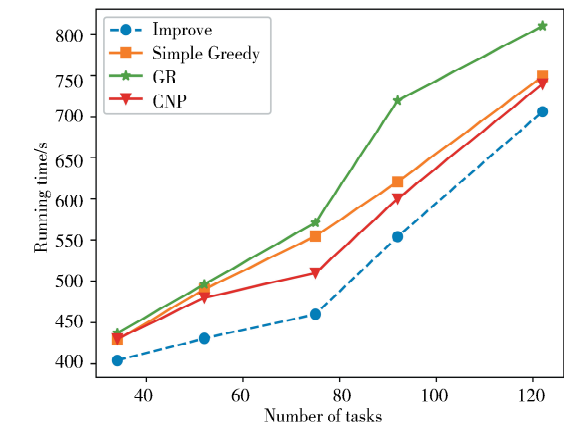


图 8 任务数量变化时的运行时间对比
Fig. 8 The comparison of system running time with the change of tasks number

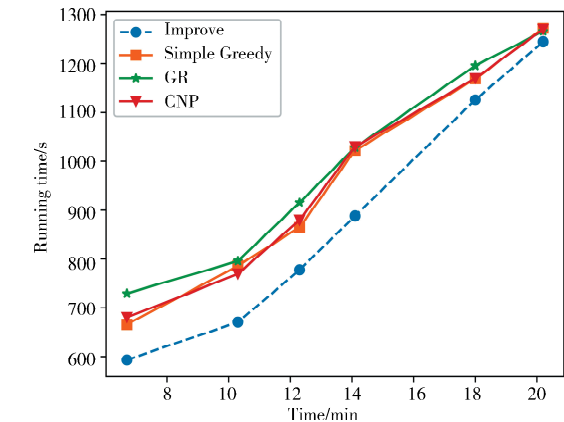


图 9 时间变化时的运行时间对比
Fig. 9 The running time comparison with time change

表 2 任务数量变化的运行时间提升量
Tab. 2 The running time increase of tasks number change

TaskNum	Improve/s	CNP/s	Increase/%
52	430.6	480.3	10.3
75	459.9	510.2	9.9
92	554.0	600.5	8.4

图 9 表示当任务到达时间长度变化时,4 种不同算法的系统运行时间对比。从图 9 中可以看出,随着任务到达时间长度增加,4 种算法的系统运行时间持续增大,但最后系统运行时间会呈现接近趋势。这是由于总体任务个数为 100,刚开始 Simple-Greedy,GR 算法和节拍合同网协议(CNP)处理任务速度没有本文所提算法快,但随着任务到达时间长度增大,任务到达的密集程度降低,另外三种算法也能较快的处理任务。如表 3 所示,在 10.3~14.1 min 之间,本文所提算法与 CNP 相比,系统运行时间至少能有 11.6%的提升。

表 3 时间变化的运行时间提升量			
Tab. 3 The running time increase of time change			
Time/min	Improve/s	CNP/s	Increase/%
10.3	670.2	770.4	13.0
12.3	778.2	880.2	11.6
14.1	888.4	1030.0	13.7

6 结 论

本文研究了机场环境行李动态抵达情况下实时分配 AMR 小车的任务分配问题,在贪婪式分配算法的基础上,根据实时行李的规模和 AMR 集群状态,提出一种基于改进贪婪式的 AMR 任务分配与调度策略。本文算法与 SimpleGreedy、GR 算法和节拍合同网协议相比,行李量大致在 55~100 之间系统运行时间至少能够获得 8.4%的提升;在本文所设定的值机岛场景下 AMR 数量在 10~14 之间系统运行时间至少能有 7.6%的提升;任务在 10.3~14.1 min 之间抵达时系统运行时间至少能有 11.6%的提升。

参考文献:

[1] SITA. Air transport industry insights: the baggage report of 2020 [EB/OL]. [2020-08-10]. <https://www.sita.aero/resources/type/surveys-reports/baggage-it-insights-2020>.

[2] Banziger T, Kunz A, Wegener K. Optimizing human-robot task allocation using a simulation tool based on standardized work descriptions [J]. J Intell Manuf, 2018, 31: 1635.

[3] Xia C, Yongtai L, Liyuan Y, et al. Cooperative task assignment and track planning for multi-UAV attack mobile targets [J]. J IIntell Robot Syst, 2020, 100: 1383.

[4] El-Ashmawi W H, Ali A F. A modified salp swarm algorithm for task assignment problem [J]. Appl Soft Comp, 2020, 94: 135.

[5] Li J, Yang F. Task assignment strategy for multi-robot based on improved Grey Wolf Optimizer [J]. J Amb IIntell Hum Comp, 2020, 11: 6319.

[6] Zhang W, Peng Y, Wei W, et al. Real-Time Conflict-free task assignment and path planning of Multi-AGV system in intelligent warehousing [C]// Proceeding of the 37th Chinese Control Conference (CCC). Wuhan, China: IEEE, 2018.

[7] Chen X, He S, Zhang Y, et al. Yard crane and AGV scheduling in automated container terminal;

A multi-robot task allocation framework [J]. Transport Res Cmer, 2020, 114: 241.

[8] Dang Q, Nguyen C T, Rudov H. Scheduling of mobile robots for transportation and manufacturing tasks [J]. J Heuristics, 2019, 25: 175.

[9] Mousavi M, Yap H J, Musa S N, *et al.* Multi-objective AGV scheduling in an FMS using a hybrid of genetic algorithm and particle swarm optimization [J]. Plos One, 2017, 12: 131.

[10] Tang Q, Baste T, Geilen M, *et al.* Task-FIFO Coshcheduling of streaming applications on MPSoCs with predictable memory hierarchy [J]. ACM T Embed Compuut S, 2017, 16: 1.

[11] Alworafi M A, Dhari A, Al-Hashmi A A, *et al.* An improved SJF scheduling algorithm in cloud computing environment [C]//Proceeding of the International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT). Mysuru, India: IEEE, 2016.

[12] 王鑫, 王人福, 蒋华. 改进贪婪算法的云任务调度研究[J]. 微电子学与计算机, 2018, 35: 109.

[13] Tong Y, Wang L, Zhou Z, *et al.* Flexible online task assignment in real-time spatial data [C]//Proceeding of the VLDB Endowment. [S. l.: s. n.], 2017.

[14] 桑泽磊, 唐敦兵, 郑堃, 等. 节拍合同网协议及其在作业车间 AGV 调度的应用[J]. 南京航空航天大学学报, 2016, 48: 895.

[15] 严飞, 祝小平, 周洲, 等. 考虑同时攻击约束的多异构无人机实时任务分配[J]. 中国科学: 信息科学, 2019, 49: 555.

[16] 张梦颖, 王蒙一, 王晓东, 等. 基于改进合同网的无人机群协同实时任务分配问题[J]. 航空兵器, 2019, 26: 38.

[17] 徐春芳. 泊松分布与泊松流[J]. 硅谷, 2010, 4: 11.

[18] To H, Shahabi C, Kazemi L. A Server-assigned spatial crowdsourcing framework [J]. ACM TSAS, 2015, 1: 1.

引用本文格式:

中文: 谢进, 向勇, 杨秀清, 等. 基于改进贪婪式算法的 AMR 任务分配[J]. 四川大学学报: 自然科学版, 2021, 58: 042003.

英文: Xie J, Xiang Y, Yang X Q, *et al.* AMR task allocation based on improved greedy algorithm [J]. J Sichuan Univ; Nat Sci Ed, 2021, 58: 042003.