

自动化立体仓库货位分配与作业调度集成优化



汤洪涛 闫伟杰 陈青丰 鲁建厦 詹燕

浙江工业大学特种装备制造与先进加工技术教育部浙江重点实验室 杭州 310023

(tanght@zjut.edu.cn)

摘要 针对动态提高单载具堆垛机式自动化立体仓库拣选效率的问题,文中提出了一种基于共享货位存储与动态订单拣选策略下的货位分配与作业调度集成优化方法。将动态移库优化扩展到仓库的整个拣选生命周期,建立以双指令循环下堆垛机拣选任务所需的总作业时间最短为评价目标的数学模型,提出了一种基于 K-Medoids 聚类的粒子群优化(Particle Swarm Optimization, PSO)算法,用 K-Medoids 算法通过产品与订单的相关性进行初始货位的聚类分析,筛除劣质解的货位范围,并在 K-Medoids 聚类算法生成的解类簇基础上获得精确解。实验结果表明,考虑动态移库可以使仓库拣选效率提高 20%,且该算法与传统 PSO 算法相比求解时间下降 66%左右。

关键词: 动态拣选;双指令循环;货位共享;集成调度;粒子群算法

中图法分类号 TP391

Integrated Optimization of Location Assignment and Job Scheduling in Automated Storage and Retrieval System

TANG Hong-tao, YAN Wei-jie, CHEN Qing-feng, LU Jian-sha and ZHAN Yan

Key Laboratory of E&M, Ministry of Education & Zhejiang Province, Zhejiang University of Technology, Hangzhou 310023, China

Abstract To improve the dynamic operation efficiency of single shuttle stacker Automated Storage and Retrieval System (AS/RS), the integrated optimization method of location assignment and job scheduling based on shared location storage and dynamic order picking strategy is proposed. The dynamic shift library optimization is extended to the entire picking life cycle of the warehouse, the mathematical model with minimized total working time required for the stacker to do tasks under single shuttle dual-command cycle is established. The PSO algorithm based on K-Medoids clustering algorithm is designed, K-Medoids algorithm is used to analyze the initial location of the product through the correlation between the product and the order, screen out the range of inferior quality solutions, and the PSO algorithm is used to find the exact solution to the problem based on the class cluster of the solution generated by the K-Medoids class algorithm. The experiments show that considering the transfer case under special circumstances could really improve 20% of the operation efficiency of the warehouse and the solution time of the algorithm could reduce about 66% compare with the traditional PSO algorithm.

Keywords Dynamic picking, Dual command circle, Location sharing, Integrated scheduling, Particle swarm

1 引言

自动化仓库系统(Automated Storage/Retrieval Systems, AS/RS)具有高吞吐量、高空间利用率、高周转率和安全性等特点,已受到国内外的广泛关注^[1-3]。典型的 AS/RS 由一个

或多个自动化存储/检索机器(多称为堆垛机)及其访问的货架组成,货架沿直通道成对排列,执行仓库系统发出指令,堆垛机沿通道行走完成出入库作业(存取、拣选或重定位)。堆垛机一般是单载具或者双载具(很少超过双载具),其中每个载具仅运送一个单元货物(通常为托盘或料箱,下文直接称之

到稿日期:2019-04-07 返修日期:2019-08-05 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:国家重点研发计划(2018YFB1308100);特种装备制造与先进加工技术教育部/浙江省重点实验室开放基金(EM2017120104);浙江省科技厅重点研发计划(2018C01003);浙江省教育厅科研资助项目(Y201839558);浙江省自然科学基金(LY19G020010);浙江工业大学科研启动基金(3827102007T)

This work was supported by the National Key Research and Development Plan (2018YFB1308100), Special Equipment Manufacturing and Advanced Processing Technology Ministry of Education/Zhejiang Key Laboratory Open Fund (EM2017120104), Zhejiang Provincial Science and Technology Department Key Research and Development Plan (2018C01003), Zhejiang Provincial Education Department Scientific Research Project (Y201839558), Zhejiang Natural Science Foundation (LY19G020010), and Zhejiang University of Science and Technology Startup Fund (3827102007T).

通信作者:陈青丰(qfchen@zjut.edu.cn)

为托盘)。AS/RS的两个关键问题分别是货位分配和堆垛机存取作业路径优化,良好的货位分配策略与堆垛机存取作业路径优化可以降低仓储的成本,提高堆垛机的拣选效率,进而提高AS/RS的利用率。

国内外学者对立体仓库的货位分配与堆垛机存取作业路径优化问题^[4-7]进行了大量研究。在实际应用中,货位分配与堆垛机存取作业路径优化相互影响,单独优化货位分配或堆垛机作业调度只能起到部分优化的作用,因此同时考虑货位分配和堆垛机存取作业路径优化,可以更好地提高AS/RS的利用效率。Chen等^[8]以最小化堆垛机存/取作业总时间为目标,研究共享货位存取策略下单载具AS/RS的货位分配和作业调度联合优化问题,提出先通过两阶段优化算法求得初始解,然后利用禁忌搜索算法对求得的解进行改进。Zhou等^[9]提出了以各堆垛机作业时间总和最小与作业时间均衡为目标的出入库作业优化模型,用出入库优化作业的规则替代原有作业规则,分析了堆垛机在全速与减速两种运行状态下联合作业和单指令作业两种模式的作业时间,解决了多台堆垛机作业不均衡问题。Cai等^[10]以堆垛机执行任务所需总时间为评价标准,针对多台立体化仓库堆垛机调度问题,建立了一种基于混合命令序列的堆垛机调度模型,运用遗传算法进行优化研究。Hachemi等^[11]以优化双指令循环周期下单载具堆垛机的最短行程时间为目标,同时考虑货位分配和拣选路径的集成,提出逐步求解的优化方法,根据存储和检索请求确定存储产品的位置以及要拣选产品的位置。Yang等^[12]以最小化行程时间为目标,提出货位分配与作业调度集成优化方法,从指令分组、货位分配及子行程作业模式与顺序3方面进行分析,并设计了双层遗传搜索算法进行求解。Lu等^[13]研究了仓库中两种移库策略下的动态移库优化问题,提出了两阶段启发式方法来生成初始解决方案,然后用禁忌搜索算法改进初始解决方案。Koh等^[14]以堆垛机作业时间为评价目标,研究基于周转与随机存储分配规则下单指令循环和双指令循环的作业时间。Yang等^[15]以堆垛机总作业时间最短为评价目标,研究了共享存储下AS/RS中货位分配和作业调度的联合优化模型,提出可变邻域搜索算法进行求解,考虑允许重用由检索操作产生的空位置,验证了共享存储下操作模式的优势。Tanaka等^[16]以最小化总行程时间为评价目标,研究共享存储策略下考虑具有单独输入点和输出点的单元负载AS/RS的作业拣选问题,提出基于MILP(混合整数线性规划)求解器的求解算法。

综上所述,Zhou等^[9]、Cai等^[10]、Hachemi等^[11]、Yang等^[12]、Koh等^[14]、Tanaka等^[16]考虑了AS/RS货位分配与作业调度的集成优化来提高仓库效率,采用了传统的静态货位分配与订单拣选方法,但该方法需要批量生成静态拣选清单,耗时且仓库利用率低^[17]。Chen等^[8]、Yang等^[15]、Tanaka等^[16]考虑在AS/RS货位分配与作业调度的集成优化的基础上,增加货位共享的存储模式,采用动态的货位分配方式,允许重用由拣选操作产生的空货位。为了进一步缩短响应时间,Lu等^[18]考虑了动态订单拣选策略,允许在拣选周期内更改拣选清单,提高拣选效率。虽然已有的AS/RS货位分配与

作业调度集成优化的研究较多,但是同时考虑共享货位存储与动态订单拣选策略的研究很少。Lu^[13]研究了仓库不进行作业拣选时的动态移库优化问题,并证明了移库有助于提高仓库的拣选效率。因此,本文在前人的研究基础上,提出研究基于共享货位存储与动态订单拣选策略的AS/RS货位分配与作业调度的集成优化问题,并将动态移库优化扩展到仓库的整个拣选生命周期,提出了一种基于K-Medoids聚类的粒子群优化(Particle Swarm Optimization,PSO)算法,用于AS/RS货位分配与作业调度的集成优化,从而提高仓库的拣选效率。

2 问题描述及符号意义

本文研究的立体仓库布局如图1所示,由总入库口、总出库口、巷道、堆垛机、货架、输送机 and 传送带组成,其中每个巷道分别有一台堆垛机、一个入库口和一个出库口,且入库口和出库口在同一个位置。货物托盘到达仓库,从总入库口进入,由传送带输送到目标货位所在的巷道出入库口,再由堆垛机搬运至指定的货位存储后,堆垛机拣取其他出库订单货物托盘至巷道出入库口,由输送机运到总出库口,完成出库。在单载具堆垛机调度研究中,根据存储和拣选的处理方式不同,可以分为单指令周期和双指令循环周期两类。单指令周期将存储和拣选单独处理,在一个周期里只完成一个存货任务或一个取货任务;双指令循环周期将存储和拣选配对成批处理,在一个周期里完成一个存货任务和一个取货任务。本文研究单载具堆垛机双指令循环周期下的调度问题,即在立体仓库动态仓储环境中,堆垛机每次运行时先后执行存/取两个操作的最优货位分配与作业调度问题。堆垛机存取作业的时间和距离采用切比雪夫距离来衡量,这里切比雪夫距离指垂直行驶时间和水平行驶时间之间的最大行程时间。每个堆垛机在巷道出入库口有足够的缓冲空间用于装载/卸载托盘。

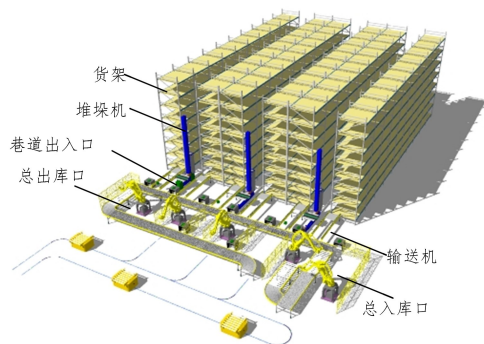


图1 堆垛机式AS/RS

Fig. 1 AS/RS with stacker crane

AS/RS集成调度优化研究多以堆垛机总作业距离最小或堆垛机总作业时间最短为评价目标,本文考虑堆垛机总作业时间包括堆垛机行走的时间、挂起与放下托盘的时间,因此采用堆垛机总作业时间最短为评价目标;考虑仓库的货位共享,即存储托盘可以存储在任何货架中,拣选过程中产生的空货位可以重用。按照订单进行动态拣选,即依据订单要求,把物品从其存储位置拣出,允许在拣选周期内更改拣选清单,加

入新到达的出库订单和新到达的入库货物,并使用“截止日期”以确保某个货位的出库请求不会被过度延迟,关联性货物就近存储,高频收发产品靠近出口,缩短货位检索时间。在移库中堆垛机的运行情况如图2所示。其中,图2(a)表示堆垛机执行拣货进库作业;图2(b)表示堆垛机执行拣货出库作业;图2(c)表示堆垛机空闲时执行移库作业,无拣选任务时堆垛机将远离进出口的托盘,移动到靠近进出口的位置;图2(d)表示堆垛机执行双指令作业,从仓库出入口拣选托盘到目标的货位以后,拣选订单托盘出库;图2(e)表示堆垛机在特殊情况下执行移库作业,即符合移库条件的移动距离小于拣货出库的移动距离。堆垛机有订单拣选任务时,只有当天出库的托盘才可以考虑移库。以每个巷道入库口堆垛机的位置为原点,以 (X, Y, Z) 来标识堆垛机正在服务的库位,其中 X 代表货架列数, Y 代表货架层数, Z 代表货架排数;脚标 $move$ 表示要移库的货位,脚标 $stop$ 表示堆垛机的停驻点,脚标 out 表示拣货出库。具体移库条件为:

$$x_{move} < x_{stop}, y_{move} < y_{stop} \text{ 且 } x_{stop} < x_{out}, y_{stop} < y_{out} \quad (1)$$

即符合库内移动条件的托盘完全在停驻点回程路线上,即在同一货架,且需要拣货出库的托盘在远离出入口的方向,远离停驻点,移动距离大于 D_2 :

$$(x, y, z)_{out} - (x, y, z)_{stop} > D_2, \text{ 且 } (x, y, z)_{move} - (x, y, z)_{stop} \leq D \quad (2)$$

即当停驻点到拣货出库的托盘的距离大于 D_3 ,停驻点到符合移库条件的托盘的距离小于 D_1 时,选择库内移动。其中, $D = \{D_1, D_2, D_3\}$ 为常数,且 $D_1 < D_2 < D_3$, D 单位为仓库单位货格,例如移库限制为64,表示在货位上下左右8个货位的范围, D 的选取与仓库的大小及存取货物相关。

空闲时的移库原则是无拣选任务时堆垛机将远离进出口的托盘移动到靠近进出口的位置。考虑仓库的拣选能力,限定拣选订单个数不超过堆垛机的拣选能力,每个堆垛机同时拣选订单数不超过2。

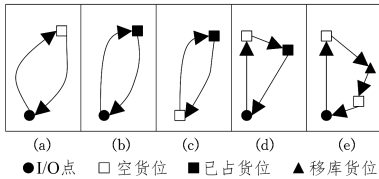


图2 堆垛机运行情况示意

Fig. 2 Schematic of stacker crane operation

本文符号的意义如下: I 表示入库产品托盘集合; i 表示入库产品托盘, $i \in I$; B 表示订单集合; b 表示入库订单, $b \in B$; F 表示仓库中的产品信息集合; s 表示堆垛机,堆垛机总数为 S ; m 表示拣选作业,拣选作业总数为 M ; ω 表示移库判断系数,不符合移库条件时 $\omega = 1$,否则 $\omega = 0$; o 表示堆垛机的进出口; L 表示仓库产品对应的位置集合; LR 表示检索对应的位置集合; LE 表示初始的空位置集合; AL 表示仓库全部位置集合, $AL = LR \cup LE$; p, q 表示位置 p 和位置 q , $p \in AL$, $q \in AL$; $d_{s,pq}$ 表示堆垛机 s 在位置 p 和位置 q 之间的作业时间, $s = 1, \dots, S$; d_s 表示堆垛机 s 的总作业时间; C 表示产品聚簇集合; c, c_l 表示类 c , $c \in C$, $c_l \in C$; $k_{i,c,m}$ 表示第 m 次拣选托

盘 i 属于类 c 值为1,否则为0; $y_{i,c,cl,m}$ 表示第 m 次拣选托盘 i 从类 c 重新分配到类 cl 值为1,否则为0; $z_{p,i,b,m}$ 表示第 m 次拣选 b 订单时,托盘 i 在位置 p 时的值为1,否则为0。

3 货位分配与作业调度模型构建

3.1 模型假设

结合实际情况,对构建的模型做如下假设:

(1)货架上每个货位的尺寸都是相同的,并且每个货位上都只存储一个货箱或者托盘;

(2)AS/RS仓库中没有库存短缺,且仓库容量够用。

3.2 模型的建立

本文的目标是通过对AS/RS货位分配与作业调度的集成优化,来使堆垛机拣选所有作业所需的总作业时间最短,作业效率最高,各堆垛机间的效率平衡,进而提高仓库的作业效率。基于Yang等^[15]的数学模型,增加考虑移库需求,建立数学模型如下:

$$\min \sum_{m=1}^M \sum_{s=1}^S \sum_{b=1}^B \sum_{i=1}^I \sum_{c \in C} \sum_{p \in LE} (d_{s,op} k_{i,c,m} z_{p,i,b,m} + \sum_{q \in AL} \sum_{p \in LE} [\omega d_{s,pq} k_{i,c,m} z_{p,i,b,m} z_{q,i,b,m} + (1-\omega) \sum_{cl \in C} d_{s,pq} k_{i,c,m} y_{i,c,cl,m} z_{p,i,b,m} z_{q,i,b,m}]) + d_{s,op} k_{i,c,m} z_{p,i,b,m} \quad (3)$$

约束条件为:

$$\sum_{m=1}^M z_{q,i,b,m} \leq 1, i \in I, q \in LE, b \in B \quad (4)$$

$$\sum_{q \in LR} z_{q,i,b,m} + \sum_{q \in LM} z_{q,i,b,m} = 1, i \in I, b \in B, m = 1, \dots, M \quad (5)$$

$$k_{i,c,m} + k_{i,c,m+1} - 1 \leq y_{i,c,cl,m}, i \in I, c \in C, cl \in C, m = 1, \dots, M \quad (6)$$

$$y_{i,c,cl,m} \leq k_{i,c,m}, i \in I, c \in C, cl \in C, m = 1, \dots, M \quad (7)$$

$$y_{i,c,cl,m} \leq k_{i,c,m+1}, i \in I, c \in C, cl \in C, m = 1, \dots, M \quad (8)$$

$$\omega = \begin{cases} 1, & d_{s,pq} < D, s = 1, \dots, S, p \in LE, q \in LE \\ 0, & \text{else} \end{cases} \quad (9)$$

$$z_{q,i,b,m} \in \{0, 1\}, i \in I, q \in AL, b \in B, m = 1, \dots, M \quad (10)$$

其中,式(3)为目标函数,第一部分为堆垛机执行拣货进库作业的时间;第二部分有两个求和公式,第一个为堆垛机从拣货入库作业的位置到拣货出库作业的位置的执行时间,第二个为堆垛机进行移库作业时库内移动的时间;第三部分为堆垛机执行拣货出库作业的时间。约束(4)保证每个空位置在每次拣选时最多访问一次;约束(5)保证堆垛机执行拣货出库作业与执行库内移动作业时最多只发生一次,且在每个检索位置每次拣选最多访问一次;约束(6)~约束(8)保证类之间的运动符合 $y_{i,c,cl,m}$ 的定义;约束(9)表示移库操作的约束条件;约束(10)指决策变量的取值范围。

4 算法描述

随着物流业的快速发展,产品的出入库更加频繁,需要更加快速、高效地完成作业,对自动化立体仓库调度提出了更高要求。PSO是一种基于群体的自适应搜索的启发算法^[19],具有较快的收敛速度、较少的参数设置、高效的搜索能力等优点,但其缺点是无法较好地处理离散的优化问题,容易陷入局部最优。对此,本文引入K-Medoids聚类算法^[20-22],该算法可以很好地控制离群数据,具有较强的鲁棒性与较高的准确

性。针对 K-Medoids 聚类算法的聚类结果过于依赖于初始聚类中心的缺点,本文提出利用产品特性确定初始中心点的方法,将进库托盘作为初始化中心点,以消除中心点选取的影响,避免了多次循环,大大减少了计算的工作量。本文提出了一种基于 K-Medoids 聚类的 PSO 算法进行求解,通过 K-Medoids 聚类算法加快 PSO 粒子的迭代,缩短求解时间。

本文算法的基本思想是:首先在产品托盘到达时判断其是否属于出库订单,若属于则继续判断是否满足出库条件,若满足则出库,否则由堆垛机负载均衡规则进行堆垛机任务分配;其次对 PSO 算法进行初始化,基于 K-Medoids 聚类算法,通过产品与订单的相关性进行初始货位的聚类分析,筛选劣质解的货位范围,提高算法的计算效率;然后在 K-Medoids 聚类算法生成的解的类簇的基础上,用 PSO 算法进行求解,算法流程如图 3 所示。

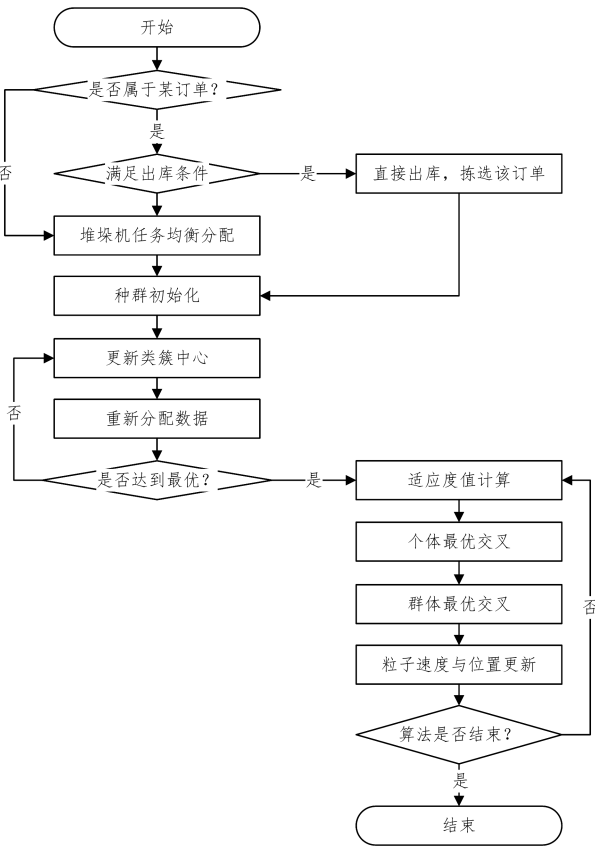


图 3 算法求解流程图

Fig. 3 Algorithm solution flow chart

(1) 基于作业均衡的堆垛机任务分配

本文从最小化作业拣选时间以及均衡存储的角度考虑堆垛机的作业均衡分配,选择最小作业时间调度方案,系统优先选择作业时间较短的堆垛机,入库时首选存货较少的货架,而出库时首选存货较多的货架。基于作业均衡^[23]的堆垛机任务分配的实现步骤如下。

1) 判断到达托盘是否属于聚簇,若属于则分配给该聚簇所在巷道的堆垛机,执行拣货入库作业,并转步骤 3), 否则转步骤 2)。

2) 为使堆垛机作业的时间不均衡最小化,对堆垛机进行如下分配:

$$\min \sum_{i=1}^I \left(\alpha \sum_{s=1}^S d_s + \beta \sqrt{\frac{\sum_{s=1}^S d_s - d}{S-1}} \right) \quad (11)$$

其中, $d_s = d_{s,op} + d_{s,pq} + d_{s,qp}$ 表示堆垛机 S 满足当前托盘上架的作业时间, $\bar{d} = \bar{d}_s / s$; α 和 β ($0 \leq \alpha \leq 1, 0 \leq \beta \leq 1$) 为权重系数,且 $\alpha + \beta = 1$, 由层次分析法的判断矩阵^[24]得出 $\alpha = 0.75, \beta = 0.25$ 。

3) 直到堆垛机均衡任务分配优化解不再变化,输出当前结果,本次求解结束,否则返回步骤 1)。

(2) PSO 算法速度与位置的计算

PSO 算法速度与位置的计算如式(12)所示:

$$v_{id}^{k+1} = \omega v_{id}^k + c_1 r_1 (p_{id} - x_{id}^k) + c_2 r_2 (p_{gd} - x_{id}^k) \quad (12)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad (13)$$

其中, c_1 和 c_2 为学习因子,也称为加速常数; k 是迭代次数; r_1 和 r_2 为 $[0, 1]$ 范围内的均匀随机数; ωv_{id} 表示惯性权重或动量,反映粒子的运动习惯,代表粒子有维持自己先前运动的趋势; $c_1 r_1 (p_{id} - x_{id}^k)$ 表示认知部分,反映了粒子对自身历史经验的记忆或回忆,代表粒子有向自身历史最佳位置逼近的趋势; $c_2 r_2 (p_{gd} - x_{id}^k)$ 表示社会部分,反映了粒子间协同合作与知识共享的群体历史经验,代表粒子有向群体或邻域历史最佳位置逼近的趋势。

(3) 货位编码

本文采用对粒子群进行实数编码的方法。货架上的货位由水平、垂直、竖直 3 个方向的距离来决定,给每个货位采取实数编号,从 1 开始,则每个货位的编号表示为:

$$NUM = K \times C \times R \quad (14)$$

其中, K 为仓库中货架的总排数, C 为该仓库中每排货架的总列数, R 为每排货架的总层数, (k, c, r) 表示当前货位对应的排、列、层位置坐标。

(4) 粒子解码

根据实际情况对仓库中的货位采用二进制解码,粒子位置只有两种状态值 0 与 1, 1 表示该货位被选中, 0 表示该货位没有被选中, 货位具有唯一性。问题的求解可行域由求解范围的货位及其状态来决定, 粒子解码的维度必须与可行域匹配, 解码索引必须与可行域求解范围中的货位号一一对应。粒子解码实例如表 1 所列, 由表 1 可知, 选中的是 38, 46, 66 号的货位。

表 1 PSO 算法的解码实例

Table 1 Algorithm decoding example

粒子位置编码	0	0	1	1	0	1	0	0
索引	1	2	3	4	5	6	7	1
对应货位号	8	32	38	46	58	66	79	8

(5) K-Medoids 聚类

采用相关性系数表示样本之间的距离,通过对历史订单数据进行采样分析求得相关性系数。定义不同种类产品 a 和 d 之间的相似系数:

$$S_{ad} = S_{da} = \sum_{i \in B} u_{ia} u_{id}, 1 \leq a \leq m, 1 \leq d \leq m, a \neq d \quad (15)$$

其中, S_{ad} 为 a 和 d 之间的相似系数, a 和 d 两种产品出现在同一订单的次数之和越大,两种产品之间的相关性就越强(为

避免不相关的产品之间因为偶然原因表示出来的相关性,当 $S_{ad} \leq 0.05$ 时令其取 0, 同种产品的相似性最强为 1)。

如果两个订单所包含的产品位于同一个巷道,那么这两个订单具有相似性,两个订单所在的共同货架数越多则相似性越大,应用以上的订单信息来计算相似性系数,将所有订单两两组对,定义产品 a 与订单 b 之间的相似系数:

$$S_{ab} = \sum_{i \in B} \max(u_{ai} u_{bi}), a \in B, b \in B \quad (16)$$

K-Medoids 聚类算法的求解过程如下:1)初始化聚类中心,将进库托盘产品作为初始化中心点,即初始化聚类中心点集 $T=I$;2)构造初始分类簇 k_j ,根据式(16)计算数据集中每个数据对象 j 的距离 $d(j)=S_{ab}$;依据 $d(j)$ 的值将数据集 F 中的样本升序排列,得到样本集 F' ,然后根据距离最近原则,将 F 中的样本分配给 T 的每一个聚类中心,得到初始的类簇 k_j ,并计算初始划分的聚类绝对值差之和;3)更新类簇的中心点,计算每一类簇的新中心点,使新中心点到其所在类簇的其他数据的距离总和最小,更新所有的类簇的中心点为新中心点;4)重新分配数据,依据距离最近原则,将数据进行重新分配,使每个样本分配到与其距离最近的类簇中心,计算当前划分的聚类绝对值差之和,若其没有变化,则算法结束,否则转步骤 3)继续执行。

(6) 适应度函数

适应度函数用于评价各粒子的性能优劣,根据适应度值的大小来寻找粒子的最优值,从而更新种群中其他粒子的状态。所有粒子由函数 $f(x)$ 评价其适应度函数值, $f(x_i^k)$ 表示 x_i^k 对应的适应度函数值,个体极值点 p_{best} 根据 $f(p_{best})$ 的大小按照下述规则更新:

$$p_{best}^{(k+1)} = \begin{cases} x_i^{k+1}, & \text{if } f(x_i^{k+1}) < f(p_{best}^{(k)}) \\ p_{best}^{(k)}, & \text{otherwise} \end{cases} \quad (17)$$

全局极值点 g_{best} 取所有个体极值点中的最好点进行更新。

(7) 惯性权重因子

算法中,惯性权重因子 ω 影响粒子的全局和局部搜索能力, ω 较大时有利于全局搜索,反之更利于局部搜索。本文采用线性递减惯性权重,即:

$$\omega(k) = \frac{\omega_{start}(\omega_{start} - \omega_{end})(T_{max} - k)}{T_{max}} \quad (18)$$

其中, ω_{start} 为初始惯性权重; ω_{end} 为迭代至最大次数时的惯性权重; k 为当前迭代次数; T_{max} 为最大迭代次数。一般来说,惯性权重取值 $\omega_{start} = 0.9$, $\omega_{end} = 0.4$ 时,算法的性能最好。这样随着迭代的进行,权重由大递减,使得算法在初期有一个较大的搜索区域进行探查,保持了较强的全局搜索能力,后期较小的惯性权重有利于更精确的局部搜索。本例根据实际规模,选择 $\omega = 0.9$ [25-26]。

(8) 学习因子

学习因子用来控制粒子转向个体极值 p_{best} 和全局极值 g_{best} 的加速权重,在算法搜索初期, c_1 取较大的值,能 c_2 取较小的值,可以增强粒子的全局搜索能力;反之,在搜索后期, c_1 取较小的值, c_2 取较大的值,能增强粒子局部搜索能力,采用能够自适应地随着迭代次数的增加而调整取值的学习因子,以提高算法的收敛精度和速度,保证算法的稳定性。因此,本

文采用粒子的适应度值来动态调节学习因子。

$$c_1 = 2 + \frac{f(x_i^{k+1}) - f_{ave}}{f_{ave} - f_{min}} \quad (19)$$

$$c_2 = 2 - \frac{f(x_i^{k+1}) - f_{ave}}{f_{ave} - f_{min}} \quad (20)$$

其中, $f(x_i^{k+1})$ 的含义同上; f_{ave} 和 f_{min} 分别表示整个粒子群的粒子适应度值的平均值和最小值。

本文算法具体步骤如下:

步骤 1 若托盘/订单到达,则转步骤 2;若无托盘/订单到达,则转到步骤 4。

步骤 2 若是属于当天出库订单的产品,则直接出库,转到步骤 4(此操作最多可执行 2 次),否则转到步骤 3。

步骤 3 基于作业均衡,分配堆垛机任务。

步骤 4 初始化粒子群,设置聚类数目 K 和粒子数目 N 。

步骤 5 按照 K-Medoids 聚类算法进行优化,生成粒子初始位置类簇,并将进库托盘产品作为初始化中心点,即初始化聚类中心点集 $T=I$,构造初始的类簇 k_j 。

步骤 6 更新类簇的中心点。

步骤 7 重新分配数据,至达到聚类优化解,转步骤 8,否则转步骤 6。

步骤 8 进行适应度函数计算,根据式(17)计算粒子的适应度值;并将其从小到大进行排序,记录个体极值 p_{best} 与全局极值 g_{best} 。

步骤 9 更新粒子的 p_{best} 与 g_{best} ,重新计算粒子的适应度值,个体通过与个体最优粒子进行交叉来更新,从而得到较好的个体;个体通过与群体最优粒子进行交叉来更新,得到较好的群体最优粒子,从而更新粒子的 p_{best} 与 g_{best} 。

步骤 10 更新粒子速度和位置,按照式(18)一式(20)更新权重系数和学习因子,按照式(12)一式(13)更新粒子的速度和位置。

步骤 11 判断、选择当前全局最优位置的粒子,并依照粒子最优位置的更新过程,来更新粒子的局部最优位置;如果达到最大迭代次数,则输出 EA,得到 Pareto 最优解,否则转步骤 8。

5 案例验证

5.1 实例介绍

为了对所提算法进行评价,以某工厂已经存储一定数量货物的自动化仓库为例,进行实例验证。该立体仓库有 6 排货架,货架有 15 层 100 列,共有 $3 \times 2 \times 15 \times 100 = 9000$ 个货位,每个货位的尺寸为 $2.98\text{m} \times 1.12\text{m}$,堆垛机在水平和垂直方向上的最大速度为分别 2.1m/s 和 0.75m/s ,加速度分别为 0.4m/s^2 和 0.6m/s^2 。堆垛机的最大运输能力是一个托盘,每天堆垛机存取作业的数量随机产生。仓库共有 300 种产品,仓库中已有的每种产品的数量为 10~15 的随机整数;订单数为 100,每个订单包含产品种类数量为 1~5 的随机整数,每种产品的数量为 5~10 的随机整数;进库产品 100 次,每次包含产品种类数量为 1~5 的随机整数,每种产品数量为 5~10 的随机整数。算法由 MATLAB 编程实现,所用计算机

配置为 Intel®Core(7M)i7-7700CPU@3.6 HZ,16.0GRAM.

5.2 仿真结果分析

为了验证所提算法的性能,针对上文随机生成的订单与出入库产品,本文采用动态拣选的方式。为了测试特定条件下的移库对堆垛机作业拣选的影响,分析了不考虑移库与考

虑移库两种情况,设置了5种移库条件。堆垛机的运行情况如图4所示,图4(a)表示不考虑特殊情况下的移库,图4(b)表示移库限制为64,图4(c)表示移库限制为81,图4(d)表示移库限制为100,图4(e)表示移库限制为169,图4(f)表示移库限制为225。

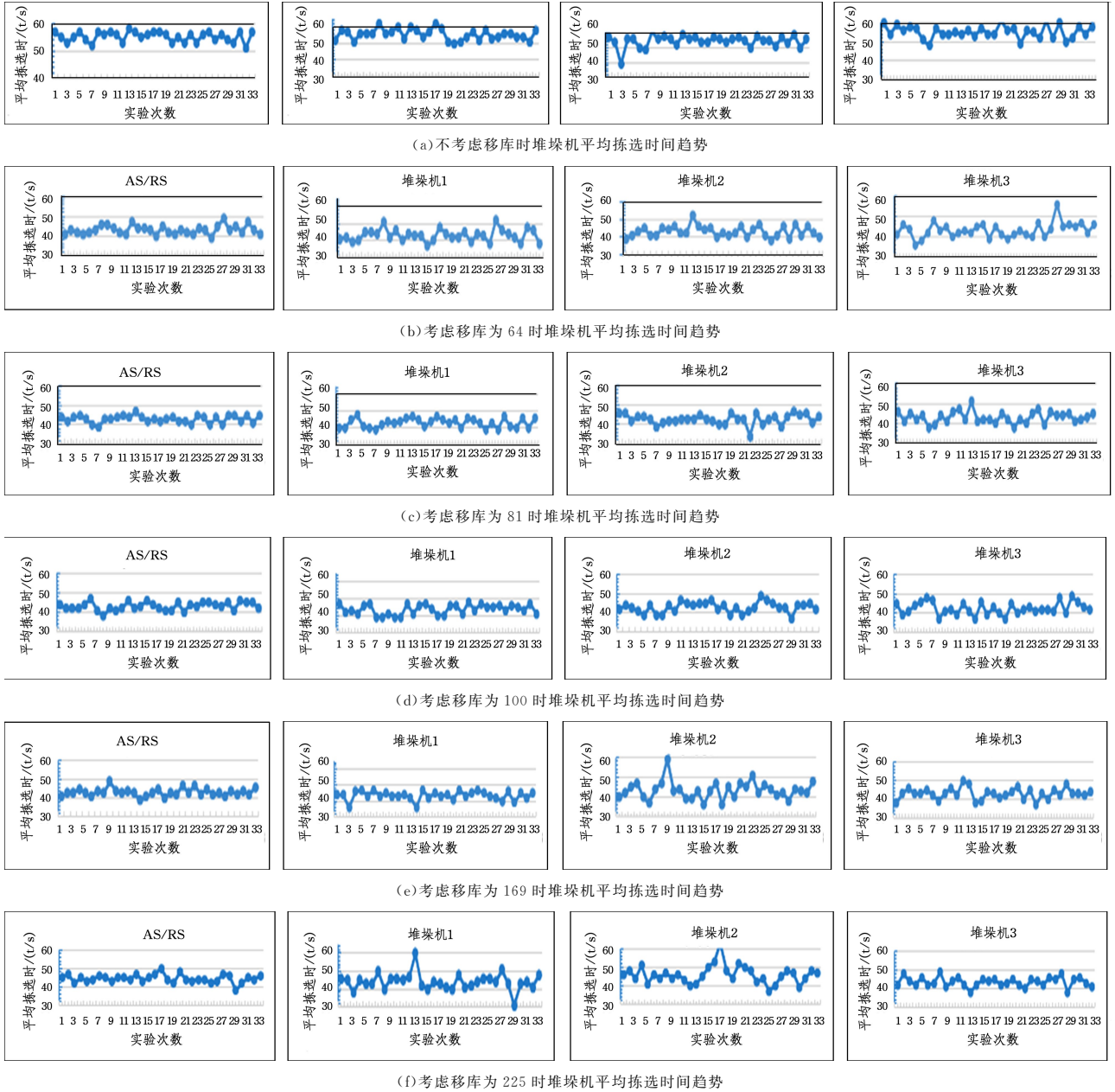


图4 平均拣选时间趋势

Fig. 4 Average picking time trend

5.2.1 移库对拣选产生的影响的分析

从图4(a)可以看出,不考虑移库时的平均拣选时间为50~60s;从图4(b)~图4(f)可以看出,在考虑了移库之后,平均拣选时间为40~50s,表明在考虑移库之后,平均拣选时间降低了20%左右,整体仓库或是单个堆垛机的拣选效率都有显著的提高。对具体的移库限制条件进行分析,从表2可以看出,平均拣选时间与移库限制的大小有关,限制条件为64~100时的平均拣选时间最短,相差不大;限制条件继续增加为169~225时平均拣选时间增长。

表2 平均拣选与移库限制的关系

Table 2 Relationship between average picking and shifting restrictions

平均拣选时间(t/s)	堆垛机1	堆垛机2	堆垛机3	整个仓库
移库限制64	43.06	43.00	43.27	43.15
移库限制81	43.15	42.82	43.00	43.06
移库限制100	43.64	43.00	42.45	43.12
移库限制169	43.45	43.30	42.97	43.24
移库限制225	44.12	45.91	43.58	44.48

5.2.2 堆垛机负载均衡分析

表 3 列出了堆垛机在不同条件下的平均拣选循环次数,表 4 列出了堆垛机平均拣选循环次数的方差。从表 3

表 3 堆垛机平均拣选循环次数
Table 3 Stacker average picking cycle

堆垛机	平均循环次数	移库条件						
		不考虑移库	移库限制为 64	移库限制为 81	移库限制为 100	移库限制为 125	移库限制为 169	移库限制为 225
堆垛机 1		106.48	112.61	111.76	114.76	109.48	110.39	110.91
堆垛机 2		109.79	115.00	115.76	112.64	107.45	108.69	111.56
堆垛机 3		111.97	111.10	110.12	110.12	111.00	110.00	110.72

表 4 堆垛机拣选循环次数的方差
Table 4 Stacker picking cycle variance

堆垛机	平均循环次数	移库条件						
		不考虑移库	移库限制为 64	移库限制为 81	移库限制为 100	移库限制为 125	移库限制为 169	移库限制为 225
堆垛机 1		11.80	9.59	11.87	15.54	11.33	13.52	12.28
堆垛机 2		14.23	12.38	11.42	13.54	11.64	13.52	12.79
堆垛机 3		8.98	12.72	9.43	12.31	13.13	14.92	11.92

5.2.3 算法性能分析

通过从图 5 所示的算法寻优时间比较可以看出,本文所改进的算法在寻优方面较原来的算法有相当显著的效果,平均求解时间由原来的 150 s 减少到 50 s 左右,K-SPO 算法(基于 K-Medoids 聚类的 PSO 算法)与 PSO 算法相比计算时间更短,全局收敛效果更好,收敛精度更高。

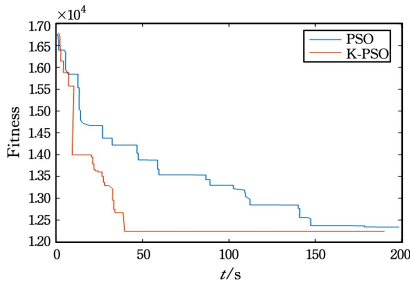


图 5 算法寻优时间比较

Fig. 5 Algorithm optimization time comparison

结束语 本文研究了自动化立体仓库货位分配与作业调度的集成优化,在共享货位存储与动态订单拣选策略条件下,建立以单载具双指令循环下堆垛机执行任务所需总作业时间最短为评价目标的数学模型,并设计了一种基于 K-Medoids 聚类的 PSO 算法求解。针对不同移库限制条件进行计算,证明了满足需求的移库可以提高仓库的拣选效率;同时基于 K-Medoids 聚类的 PSO 算法体现出了较好的鲁棒性,与标准 PSO 算法相比提高了求解速度与解的质量,满足订单的动态拣选要求,也保证了堆垛机的作业均衡分配。

本文假设在目前研究对象堆垛机的日常运行过程中不考虑该设备维修与碳排放等因素,然而在有些实际企业生产中,需要同时考虑这些因素。因此,在今后的研究中,需要进一步考虑这些因素,建立相应的优化模型,提出优化算法和求解方案。

和表 4 可以看出,在不考虑移库及 5 种移库限制条件时,3 台堆垛机基本可以实现负载均衡,并且负载均衡也较为稳定。

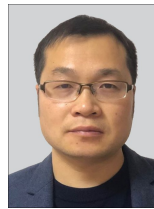
参考文献

- [1] YANG P, MIAO L X. Review of Control Optimization for Automated Storage and Retrieval Systems[J]. Industrial Engineering Journal, 2011, 14(1): 123-127.
- [2] GAGLIARDI J P, RENAUD J, RUIZ A. Models for automated storage and retrieval systems; a literature review[J]. International Journal of Production Research, 2012, 50(24): 7110-7125.
- [3] NILS B, KONRAD S. A survey on single crane scheduling in automated storage/retrieval systems[J]. European Journal of Operational Research, 2016, 254(3): 691-704.
- [4] BIENKOWSKI M, BYRKA J, CHROBAK M, et al. Approximation algorithms for the joint replenishment problem with deadlines[J]. Journal of Scheduling, 2015, 18(6): 545-560.
- [5] CAI A J, CAI Y, GUO S H, et al. Storage location assignment strategy of double-crane in automated warehouse[J]. Computer Integrated Manufacturing Systems, 2018, 24(12): 3165-3177.
- [6] LERHER T, POTRC I, SRAML M, et al. Travel time models for automated warehouses with aisle transferring storage and retrieval machine[J]. European Journal of Operational Research, 2010, 205(3): 571-583.
- [7] NILS B, DIRK B, FRANK M. A generalized classification scheme for crane scheduling with interference[J]. European Journal of Operational Research, 2017, 258(1): 343-357.
- [8] CHEN L, LANGEVIN A, RIOPEL D. The storage location assignment and interleaving problem in an automated storage/retrieval system with shared storage[J]. International Journal of Production Research, 2010, 48(4): 991-1011.
- [9] ZHOU J, ZHAO C Y, LIU Z Q, et al. Operation optimization of storage and retrieval for stackers in AS/RS of raw tobacco material[J]. Computer Integrated Manufacturing Systems, 2009, 15(4): 772-776.

- [10] CAI A J, YING J Q, WANG J, et al. Scheduling model of crane in distributed automated warehouse[J]. *Computer Integrated Manufacturing Systems*, 2016, 22(03): 793-799.
- [11] HACHEMI K, SARI Z, GHOUALI N. A step-by-step dualcycle sequencing method for unit-load automated storage and retrieval systems[J]. *Computers & Industrial Engineering*, 2012, 63(4): 980-984.
- [12] YANG W, LIU J, YUE T, et al. Integrated optimization of location assignment and job scheduling in multi-carrier automated storage and retrieval system[J]. *Computer Integrated Manufacturing Systems*, 2019, 25(1): 247-255.
- [13] LU C, ANDRE L, DIANE R. A tabu search algorithm for the relocation problem in a warehousing system[J]. *International Journal of Production Economics*, 2011, 129(1): 147-156.
- [14] KOH S G, KIM B S, KIM B N. Travel time model for the warehousing system with a tower crane S/R machine[J]. *Computers & Industrial Engineering*, 2002, 43(3): 495-507.
- [15] YANG P, MIAO L X, XUE Z J, et al. Variable neighborhood search heuristic for storage location assignment and storage/retrieval scheduling under shared storage in multi-shuttle automated storage/retrieval systems[J]. *Transportation Research Part E - Logistics and Transportation Review*, 2015, 79: 164-177.
- [16] TANAKA S, ARAKI M. Routing problem under the shared storage policy for unit-load automated storage and retrieval systems with separate input and output points[J]. *International Journal of Production Research*, 2009, 47(9): 2391-2408.
- [17] GONG Y M, RENE D K. A polling-based dynamic order picking system for online retailer[J]. *IIE Transactions*, 2008, 40(11): 1070-1082.
- [18] LU W R, FARLANE M D, GIANNIKAS V, et al. An algorithm for dynamic order-picking in warehouse operations[J]. *European Journal of Operational Research*, 2016, 248(1): 107-122.
- [19] HUANG Y, LI H Y, XU K B, et al. S-shaped Function Based Adaptive S-shaped Function Based Adaptive[J]. *Computer Science*, 2019, 46(01): 245-250.
- [20] PRANAV N, ARCHANA S, MADHAV C, et al. Empirical Analysis of Data Clustering Algorithms[J]. *Procedia Computer Science*, 2018, 125(1): 770-779.
- [21] AMIT B, ISSAM A M. Evolutionary Clustering Algorithms for Relational Data[J]. *Procedia Computer Science*, 2018, 140(1): 276-283.
- [22] CHEN J J, CHE J. IK-medoids Based Aircraft Fuel Consumption Clustering Algorithm[J]. *Computer Science*, 2018, 45(8): 306-309, 314.
- [23] DING Y, ZHANG Q, LIN G L. Quay Crane Scheduling with Yard Operation Balance at Automated Container Terminal[J]. *Journal of Chongqing Jiaotong university (Natural Science)*, 2018, 37(7): 106-112.
- [24] WOLLIAM H, XIN M. The state-of-the-art integrations and applications of the analytic hierarchy process[J]. *European Journal of Operational Research*, 2018, 267(2): 399-414.
- [25] LIU X J, WEI Y C, YUAN B X, et al. Study on Adaptive Hierarchical Clustering De-noising Algorithm of Laser Ranging in Storage of Dangerous Chemicals[J]. *Computer Science*, 2018, 45(S2): 208-211, 217.
- [26] ZOU P, LI B Z, YANG J G, et al. Hierarchical ant-Genetic algorithm-based multi-objective intelligent approach for flexible job shop scheduling [J]. *China Mechanical Engineering*, 2015, 26(21): 2873-2879, 2884.



TANG Hong-tao, born in 1976, associate professor master's tutor. His main research interests include the production of process management, manufacturing execution systems, production planning and scheduling, production and logistics system modeling and simulation.



CHEN Qing-feng, born in 1980, lecturer. His main research interests include corporate logistics and third party logistics.