

基于AGV的智能仓库系统订单分批问题研究

李珍萍¹, 付红叶¹, 卜晓奇¹, 张国维^{2,3}, 吴凌云^{2,3}

(1. 北京物资学院 信息学院, 北京 101149; 2. 中国科学院 数学与系统科学研究院, 应用数学研究所, 管理、决策与信息系统重点实验室, 国家数学与交叉科学中心, 北京 100190; 3. 中国科学院大学 数学科学学院, 北京 100190)

摘要:研究了基于自动引导小车(AGV)的“货到人”智能仓库订单分批拣选问题,在同时考虑工作人员拣选商品成本和AGV搬运货架成本的前提下,建立了以总成本极小化为目标函数的订单分批问题整数规划模型。根据订单中包含的商品信息和商品所在的货架信息构建了描述订单之间关系的加权相似度指标,分析了加权相似度与总拣选成本之间的正相关关系。基于订单之间的加权相似度设计了求解模型的贪婪算法。利用具体算例进行模拟计算,分析了加权系数的变化对订单分批结果的影响,以及加权系数 λ 的取值与工作人员拣取一件商品的成本 c_1 和AGV搬运一次货架的成本 c_2 之间的关系,得到了贪婪算法中加权系数 λ 的确定方法。进一步分析了贪婪算法的计算时间和计算效果,结果显示,通过适当选取加权系数,利用贪婪算法可以在短时间内得到订单分批问题的近似最优解;对于小规模算例,贪婪算法在最坏情况下近似比不超过1.35。利用本文的模型和算法进行订单分批,兼顾了工作人员拣取商品的成本和AGV搬运货架的成本,可以有效提高订单拣选效率,降低订单拣选总成本。

关键词:智能仓库;AGV;订单分批;整数规划模型;加权相似度;贪婪算法

中图分类号:O221.4;F253.4 文章标识码:A 文章编号:1007-3221(2020)09-0001-09 doi:10.12005/orms.2020.0221

Research on Order Batching Problem of Intelligent Warehouse System Based on AGV

LI Zhen-ping¹, FU Hong-ye¹, BU Xiao-qi¹, ZHANG Guo-wei², WU Ling-yun^{2,3}

(1. School of Information, Beijing Wuzi University, Beijing 101149, China; 2. Academy of Mathematics and Systems Science, Institute of Applied Mathematics, Chinese Academy of Sciences, Key Laboratory of Management, Decision and Information Systems, National Center for Mathematics and Interdisciplinary Sciences, Beijing 100190, China; 3. School of Mathematics Sciences, University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: The problem of order batching in parts-to-picker intelligent warehouse based on automatic guided vehicle (AGV) is studied. The costs of workers picking items from shelves and the costs of AGV transporting shelves are considered simultaneously. An integer programming model of order batching problem is established to minimize the total costs. Based on the items contained in each order and the shelves where the items of an order are stored, the weighted similarity is formulated to describe the relationship between orders. The positive correlation between weighted similarity and total picking cost is analyzed. Based on the weighted similarity between orders, a greedy algorithm for solving the model is designed. The simulation is done on a specific example. The sensitivity analysis of weighted coefficient is done. The influence of weighting coefficient change on batching results, and the relationship among the value of and the cost of λ worker picking one item c_1 and the cost of AGV moving one shelf c_2 are analyzed. The results show that the approximate optimal solution of the order batching problem can be obtained in a short time by using the greedy algorithm with appropriate weighting coefficient. For small scale examples, the approximate ratio is less than 1.35 in the worst case. Using the model and algorithm in this paper for order batching, two types of costs are considered at the same time, which can effectively improve the efficiency and reduce the total costs of order picking.

Key words: intelligent warehouse; AGV; order batching; integer programming model; weighted similarity; greedy algorithm

收稿日期:2018-10-17

基金项目:国家自然科学基金资助项目(71771028);北京市自然科学基金项目(Z180005);北京市属高校高水平科研创新团队建设项目(IDHT20180510);北京高等学校高水平人才交叉培养“实培计划”项目毕业设计(科研类);北京市智能物流协同创新中心开放课题(BILSCIC-2019KF-18)**作者简介:**李珍萍(1966-),女,博士,教授,研究方向:物流优化,智能算法;吴凌云(1975-),男,博士,研究员,研究方向:组合优化,智能算法,生物信息学。

0 引言

随着电子商务的迅速发展,电商每天要处理的订单数量日益增长,在订单处理过程中,超过 50% 的时间耗费在订单拣选作业上^[1]。因此,缩短订单拣选时间是提高物流效率、降低物流成本的有效途径。传统配送中心采取拣选人员在货架中来回穿梭的人到货拣选模式,劳动强度大、耗时长,且拣选过程容易出现差错。近年来出现的基于自动引导小车(AGV)的智能仓库中,拣选人员站立在固定的工作台前,由仓储机器人把货架搬运到拣选人员面前,完成订单拣选工作。这种新型的货到人拣选作业模式,将拣选人员从繁重的工作中解放出来,大大提高了拣选工作效率^[2]。无论是在传统的人到货拣选仓储系统还是最新的人到智能仓储系统中,单个订单中包含的物品种类和数量都不多,为了提高订单拣选效率,配送中心通常将多个订单合并成一个拣选单进行拣选。订单分批问题就是将订单池中的众多订单按照一定的规则划分成若干个批次,每一批次的订单构成一张拣货单并在一次作业中完成拣选^[3]。在基于 AGV 的智能仓库中,采取合理的订单分批策略可以大大减少工作人员的拣选操作次数和仓库机器人搬运货架的次数,降低拣选成本、提高拣选效率。

针对传统人到货拣选模式下的订单分批拣选问题的研究成果已经非常丰富^[4~11]。由于基于 AGV 的货到人拣选仓库系统的工作流程与传统的人到货拣选仓库系统的工作流程不同,已有的针对人到货拣选系统中订单分批问题的研究成果不能直接用于解决货到人智能仓库系统中的订单分批拣选问题。近年来,部分学者针对货到人智能仓库系统中的订单分批拣选相关问题开展了研究工作。张彩霞等综合考虑订单分批、路径、任务分配等三个过程,研究了基于 AGV 的“货到人”拣选模式的优化方法和模型^[1]。王艳艳等以并行自动分拣系统为研究对象,运用迭代优化、聚类分析等工具,研究了智能仓库中订单拣选拆分优化、系统调度优化及补货缓存优化问题,建立优化模型并设计了启发式自适应遗传算法^[12]。王善超以订单是否拆分为准则,建立基于通道相似度的订单不可拆分模型和基于车辆路径问题的订单可拆分模型,分别设计了求解两种模型的算法,并比较了在两种独立分拣模式下不同订单分批方法的工作效率^[13]。Lenoble Nicolas 等人在具有多个垂直升降模块的自动化仓

库中进行实验,以完成订单拣选时间最短为目标建立优化模型,并设计了求解模型的元启发式方法^[14]。王旭坪等人研究了基于相似度聚类的订单分批策略,利用相同通道数系数作为衡量订单间相似度的指标^[15]。Xi Xiang 等人研究了 Kiva 系统中的储位分配和订单分批问题,以最小化货架的访问次数作为订单分批的目标。通过最大化顺序关联或最小化顺序异化获得初始可行解,并采用变邻域搜索算法对可行解进行改进^[16]。

针对订单分批问题的研究大部分都是根据订单相似度进行聚类。在传统人到货仓储系统中,拣选人员穿越通道的数量是影响拣选效率的主要因素,因此订单相似度主要根据订单中物品所在的通道是否相同进行定义。基于 AGV 的智能仓库系统中,机器人搬运货架的次数和拣选人员从货架上拣取物品的次数是影响订单拣选效率的两个主要因素,现有文献大都基于其中一个因素定义订单相似度,尚未发现同时考虑两种因素的研究成果。

本文将研究基于 AGV 的智能仓库系统订单分批拣选问题,在给定仓库布局、货位分配信息的前提下,同时考虑订单中包含的商品信息和订单中商品所在的货架信息,构建加权订单相似度指标,并基于该指标研究订单分批策略。建立订单分批问题的数学模型并设计求解模型的算法,同时兼顾工作人员从货架上拣取商品成本和 AGV 搬运货架成本,使智能仓库系统完成订单拣选的总成本达到最小。

1 问题描述与分析

基于 AGV 的智能仓库订单分批问题可以描述为:已知仓库中有 S 个货架,每一个货架上有 h 个货位,每个货位最多存放一种商品,仓储中共存放着 M 种商品,且每种商品的货位已知。假设该仓库某时刻的订单池中有 N 张订单需要拣选,已知每张订单上包含的商品品项信息、工作人员从货架上拣取一种商品的成本、AGV 搬运一次货架的成本。问如何将 N 张订单进行分批才能使订单拣选的总成本最低?

为了简化问题,假设每张订单中的商品均不缺货;每种商品在货架上的货位固定,每个货架在仓库中的位置固定;拣选每个批次的订单时都需要 AGV 把包含该批次订单商品的货架搬运至拣选台,等待工作人员从货架上取下待拣选商品后,再将货架搬回仓库中原来的位置;拣选下一个批次订单时需要重新搬运货架,即不同批次订单中的同种

商品不能合并拣选。

在以上假设下,影响每个批次订单拣选效率和拣选成本的主要因素包括两个方面,一个是工作人员从货架上拣选商品的次数;另一个是AGV将货架搬运至拣选工作台的次数和来回行走距离。由于每个货架在仓库中的位置是确定的,为了简化问题,计算订单拣选成本时可以不考虑货架搬运距离,只考虑货架搬运次数。

基于以上假设,为了提高订单拣选效率、降低拣选总成本,在对订单进行分批的时候,应该重点考虑减少工作人员从货架上拣取商品的次数、和减少AGV搬运货架的次数。如果两个订单中包含的商品品项相同,若将它们合并拣选可以使工作人员从货架上拣取商品的次数减少一半;如果两个订单中包含的商品存放在同一个货架上,若将它们合并拣选,可以使AGV搬运货架的次数减少一半。因此在进行订单分批的时候,应该综合考虑订单中包含的商品品项信息和订单中包含的商品所在货架信息。

2 订单分批问题的整数规划模型

为了建立订单分批问题的数学模型,定义如下符号。

索引:

i, j : 订单索引, $i, j = 1, 2, \dots, N$;

k : 批次索引, $k = 1, 2, \dots, K$;

s : 货架索引, $s = 1, 2, \dots, S$;

t : 商品索引, $t = 1, 2, \dots, M$ 。

参数:

q : 每个批次允许的最大订单数量;

$a_{it} = \begin{cases} 1, & \text{订单 } i \text{ 中包含商品 } t \\ 0, & \text{否则} \end{cases}$

$b_{ts} = \begin{cases} 1, & \text{商品 } t \text{ 存放在货架 } s \text{ 上} \\ 0, & \text{否则} \end{cases}$

c_1 : 从货架上拣取一种商品的成本;

c_2 : 搬运一次货架的成本。

决策变量:

$x_{ik} = \begin{cases} 1, & \text{订单 } i \text{ 被分配到批次 } k \\ 0, & \text{否则} \end{cases}$

$y_{ks} = \begin{cases} 1, & \text{拣选批次 } k \text{ 需要搬运货架 } s \\ 0, & \text{否则} \end{cases}$

$z_{kt} = \begin{cases} 1, & \text{批次 } k \text{ 中包含商品 } t \\ 0, & \text{否则} \end{cases}$

基于以上符号,订单分批问题可以表示成如下0-1规划模型:

$$\min f = c_1 \sum_{k=1}^K \sum_{i=1}^M z_{kt} + c_2 \sum_{k=1}^K \sum_{s=1}^S y_{ks} \quad (1)$$

$$\text{s. t. } \sum_{k=1}^K x_{ik} = 1, \forall i \quad (2)$$

$$\sum_{i=1}^N x_{ik} \leq q, \forall k \quad (3)$$

$$\sum_{i=1}^N a_{it} x_{ik} \leq N \cdot z_{kt}, \forall k, t \quad (4)$$

$$z_{kt} \leq \sum_{s=1}^S b_{ts} y_{ks}, \forall k, t \quad (5)$$

$$x_{ik} \in \{0, 1\}, \forall i, k \quad (6)$$

$$y_{ks} \in \{0, 1\}, \forall k, s \quad (7)$$

$$z_{kt} \in \{0, 1\}, \forall k, t \quad (8)$$

目标函数(1)表示极小化订单拣选的总成本,其中第一项表示从货架上拣取商品的成本,第二项表示货架搬运的成本;约束条件(2)表示每个订单恰好被分配到一个批次中;约束条件(3)表示分配到每个批次的订单数量不超过规定的最大数量;约束条件(4)表示如果批次 k 中有任意一个订单包含商品 t , 则该批次就包含商品 t ; 约束条件(5)表示如果批次 k 包含商品 t , 则拣选该批次时需要搬运包含商品 t 的货架; 约束条件(6)~(8)表示决策变量取值约束。

定理 1 智能仓库系统订单分批问题属于 NP-hard 问题。

由于订单分批问题属于 NP-hard 问题,对于小规模问题,可以直接利用商业求解器如 Lingo、Cplex 等求解整数规划模型得到最优解,对于大规模问题,需要设计快速有效的近似求解算法。

3 订单加权相似度

从订单分批问题数学模型的目标函数和约束条件可以看出,影响订单拣选成本的主要因素是各个批次中包含的商品品项数量和满足各个批次拣选需求的货架数量。因此,合并包含相同商品品项或需要相同货架的订单,就可以有效降低订单拣选的总成本。基于以上思想,本节先分别按照订单中包含的商品品项是否相同、订单中包含的商品所在的货架是否相同定义两种描述订单相似度的指标:基于商品品项的订单相似度和基于货架的订单相似度,进一步定义加权相似度作为订单分批的依据。

3.1 基于商品品项的订单相似度

基于商品品项的订单相似度是根据两个订单中包含的相同商品品项数量与两个订单中的总商品品项数量之比定义的。

假设订单 i 和订单 j 中包含的商品品项集合分别为 I_i, I_j , 则基于商品品项的订单相似度 r_{ij} 可以定义为:

$$r_{ij} = \frac{|I_i \cap I_j|}{|I_i \cup I_j|} \quad (9)$$

例如, 订单 i 包含的商品品项集合为 $I_1 = \{A, D, E\}$, 订单 j 包含的商品品项集合为 $I_2 = \{B, D, E\}$ 。则订单 i 和订单 j 基于商品品项的相似度为:

$$r_{ij} = \frac{2}{4} = 0.5。$$

3.2 基于货架的订单相似度

基于货架的订单相似度定义为两个订单中的商品对应的相同货架数量与两个订单中商品对应的总货架数量之比。

表1 仓库中3个货架的各个货位存放的商品信息

货架序号	货位1	货位2	货位3	货位4	货位5
货架1	A	G	H	I	J
货架2	B	C	D	H	F
货架3	E	F	I	G	C

3.3 加权相似度

基于商品品项的订单相似度反映了订单中包含的相同商品的比例, 将商品品项相似度大的订单合并拣选, 可以减少从货架上拣取商品的次数; 基于货架的订单相似度反映了订单中商品所在的相同货架所占的比例, 将货架相似度较大的订单合并拣选, 可以减少货架搬运的次数。由于从货架上拣取商品的次数和货架搬运的次数均是影响拣选效率和拣选成本的主要因素, 因此, 本文综合考虑两方面因素, 分别赋予两种相似度权系数 λ 和 $1 - \lambda$ (其中 $0 \leq \lambda \leq 1$), 定义订单 i 和订单 j 之间的加权相似度如下:

$$T_{ij} = \begin{cases} \lambda r_{ij} + (1 - \lambda) l_{ij} & i \neq j \\ 0 & i = j \end{cases} \quad (11)$$

对于某个批次 k , 若其中包含 p 个订单 $p \geq 2$, 不妨设批次 k 对应的订单集合为: $Batch(k) = \{O_{k1}, O_{k2}, \dots, O_{kp}\}$, 则批次 $Batch(k)$ 中包含的订单之间的平均加权相似度 $Z(Batch(k))$ 为

$$Z(Batch(k)) = \frac{\sum_{i=1}^p \sum_{j>i}^p T_{ij}}{p} \quad (12)$$

加权相似度综合考虑了商品拣选的成本(次数)和货架搬运的成本(次数)。根据加权相似度进行分批, 使同一批次内订单的平均加权相似度极大化等价于使商品拣选的次数和货架搬运次数的加权和极小化, 因此根据加权相似度进行分批可以使订单拣选的总成本达到最小。

假设订单 i 和订单 j 中商品所在的货架集合分别为 S_i 和 S_j , 则基于货架的订单相似度 l_{ij} 定义为:

$$l_{ij} = \frac{|S_i \cap S_j|}{|S_i \cup S_j|} \quad (10)$$

例如, 仓库中3个货架存放的商品品项如表1所示。订单 i 包含的商品集合为 $I_i = \{A, D, E\}$, 订单 j 包含的商品集合为 $I_j = \{B, D, E\}$ 。则订单 i 中的商品对应的货架集合为 $S_i = \{1, 2, 3\}$, 订单 j 中的商品对应的货架集合为 $S_j = \{2, 3\}$, 订单 i 和订单 j 中的商品对应的相同货架集合为 $\{2, 3\}$, 总货架集合为 $\{1, 2, 3\}$, 因此订单 i 和订单 j 基于货架的相似度为 $l_{ij} = \frac{2}{3}$ 。

4 求解订单分批问题的贪婪算法

由于实际问题中涉及到的订单数量、商品品项及货架数量都很多, 其对应的订单分批问题规模很大, 无法在短时间内通过直接求解整数规划模型得到精确解, 因此需要设计快速有效的近似求解算法来得到近似最优解。本节基于订单之间的加权相似度设计求解订单分批问题的贪婪算法。

算法的基本思想: 首先根据待拣选订单中包含的商品品项及各种商品所在货架信息计算订单之间的加权相似度; 先将每个订单单独作为一个批次; 再分别计算任意两个批次合并以后, 平均加权相似度的增加量, 依次选择合并后满足批次内订单总数容量约束并且平均加权相似度增量达到最大的批次进行合并, 直到平均加权相似度不再增加或者不存在合并后满足容量约束的订单批次为止。

基于加权相似度的订单分批问题贪婪算法基本步骤如下:

输入: 待拣选订单中包含的商品品项信息及各个货架上存放的商品品项信息, 加权系数 λ 。

Step 1 计算任意两个订单之间的加权相似度 T_{ij} , 得到加权相似度矩阵

Step 2 将每个订单单独作为一个批次, 记录各个批次中的订单数量, 初始化每个批次内部订单之间的平均加权相似度为 0 (此时每个批次包含的订单数量均为 1)

Step 3 对于任意两个批次 $u, v (u \neq v)$, 按照公式 (13) 计算将批次 u, v 合并为一个批次 $Batch(u) \cup Batch(v)$ 以后, 平均加权相似度之和的增加量 Δ_{uv} 以及两个批次合并以后总订单数量 N_{uv}

$$\Delta_{uv} = Z(Batch(u) \cup Batch(v)) - Z(Batch(u)) - Z(Batch(v)) \quad (13)$$

Step 4 在满足容量约束 $N_{uv} \leq q$ 的批次中找出 Δ_{uv} 的最大值 $\max \Delta$ 及对应的批次 u^*, v^* 。

Step 5 若 $\max \Delta > 0$, 将批次 $Batch(u^*)$, $Batch(v^*)$ 合并为一个批次, 转 Step 3; 若 $\max \Delta \leq 0$, 或者不存在满足 $N_{uv} \leq q$ 的订单批次, 则结束计算。

输出: 各个批次对应的订单集合。

定理 2 对于 N 个订单, S 个货架, M 种商品, 需要将订单划分为 K 个批次的问题, 用贪婪算法求解的时间复杂度为 $O(N^2M + N^3)$ 。

5 模拟计算与分析

5.1 算例描述

某电商企业采用基于 AGV 的智能仓库系统,

该企业经销的 100 种商品 (商品编码为 1 ~ 100) 存放在 20 个货架上 (货架编码为 1 ~ 20), 每一个货架上有 5 个货位, 每个货架上的 5 个货位摆放的商品品项信息如表 2 所示。

该公司在某段时间内收到 100 个订单, 每个订单中包含的商品品项信息如表 3 所示。假设工作人员从货架上拣取一件商品的成本为 0.6 元, AGV 搬运一次货架的成本为 0.4 元, 问如何对 100 个订单进行分批才能使总拣选成本最低?

表 2 货架-商品关系表

货架序号	存放的商品序号	货架序号	存放的商品序号
1	1,2,3,4,5	11	51,52,53,54,55
2	6,7,8,9,10	12	56,57,58,59,60
3	11,12,13,14,15	13	61,62,63,64,65
4	16,17,18,19,20	14	66,67,68,69,70
5	21,22,23,24,25	15	71,72,73,74,75
6	26,27,28,29,30	16	76,77,78,79,80
7	31,32,33,34,35	17	81,82,83,84,85
8	36,37,38,39,40	18	86,87,88,89,90
9	41,42,43,44,45	19	91,92,93,94,95
10	46,47,48,49,50	20	96,97,98,99,100

表 3 订单-商品关系表

订单序号	订单中的商品序号	订单序号	订单中的商品序号	订单序号	订单中的商品序号	订单序号	订单中的商品序号
1	31,33,82,83	26	13,37,38,39,40	51	41,42,72,73,75	76	82,83,84,85
2	48,50,96,98,99	27	41,44,45,56,57,58	52	1,2,5,46,49	77	54,55,76,78,79,80
3	6,7,81,84	28	37,83,84,85	53	8,9,10,86,87	78	22,24,44
4	16,19,72,73,74,75	29	1,86,89,90	54	61,63,65,92,94	79	51,52,53,54
5	21,23,25,31,33	30	56,59,60,93,95	55	21,25,51,52,54	80	17,18,20,87,89,90
6	2,3,4,47	31	49,50,51,55	56	6,10,38,40	81	26,30,51,53
7	22,23,25,67	32	42,43,46,48,49	57	53,54,93,95	82	6,8,9,39
8	92,93,95	33	51,52,53,56,57	58	11,13,15,57,58	83	37,38,39,78,80
9	9,10,27,28,30	34	21,22,25,41,42	59	14,15,47,48,49	84	7,8,96,97,98
10	21,24,25,36,39,40	35	16,17,47,50	60	21,23,24,25	85	73,75,91,93,94
11	46,48,49,50	36	46,48,49,87,89	61	24,25,26,29,30	86	69,70,82,83,85
12	81,84,85,97,100	37	68,69,91,92,94,95	62	26,27,29,97,98,99	87	71,75,81,83,84,85
13	16,17,20,46,47,49	38	26,28,29,69,70	63	18,20,96,99,100	88	7,10,51,52,53
14	27,28,96,97,99	39	52,54,55,81,83	64	31,32,34,43,44	89	60,81,83,84,85
15	22,23,81,84	40	12,91,92,93,95	65	3,5,83	90	31,33,34,98,99,100
16	7,10,46,49	41	28,29,46,48,49,50	66	56,57,59,98,99,100	91	51,53,55,83,84,85
17	61,63,64,65,88,89	42	19,91,92,93,95	67	11,86,87	92	37,38,39,40
18	41,43,44,53,54	43	61,62,64,81,85	68	87,88,89,91,92	93	39,40,56,57
19	25,41,43,44,45	44	23,24,83,84	69	76,77,79,80	94	23,24,25,91,94
20	71,74,86,89	45	3,4,67,68,69,70	70	11,12,13,15,94	95	6,8,41,43
21	21,22,23,25	46	46,47,49,81	71	61,62,64,76,80	96	48,49,87,90
22	41,42,44,86,89	47	51,53,72,73,74	72	57,59,63,64	97	16,19,38,40
23	10,11,12,13	48	6,9,91,92,94,95	73	5,26,27,28,29	98	32,33,34,59,60
24	66,67,69,70,88,89	49	13,14,36,37,40	74	18,56,59,60	99	8,9,31,33,34,35
25	68,69,70	50	9,10,68,69,70	75	1,3,4,5,50	100	17,20,93

5.2 订单分批结果与分析

本例中工作人员拣选一种商品的成本 $c_1 = 0.6$ 元, AGV 搬运一个货架的成本为 $c_2 = 0.4$ 元, 为了利用贪婪算法求解整数规划模型, 首先要分析贪婪算法中参数 λ 的取值与整数规划模型中目标函数系数 c_1, c_2 取值的关系。不妨让参数 λ 分别取 $[0, 1]$ 之间的不同值, 计算加权相似度。利用 MATLAB 编写的贪婪算法实现程序, 按照不同参数下的加权相似度运行贪婪算法程序, 得到各种参数对应的订单分批结果, 并计算分批结果对应的各个批次中包含的不同商品品项 n_1 (即拣选各个批次订单时, 工作人员从货架上拣取商品的总次数)、各个批次中的商品对应的不同货架数量 n_2 (即拣选各个批次订单时, AGV 需要搬运货架的总次数), 以及拣选订单的总成本 f (即整数规划模型的目标函数值), 其中拣选订单总成本计算公式为 $f = c_1 \cdot n_1 + c_2 \cdot n_2$, 具体计算结果见表 4。

表 4 不同参数下的订单分批结果及对应的拣选成本指标

λ	分批 批次数	工作人员 拣取物品 的总次数	AGV 搬运货 架总次数	拣选订单 总成本(整数规划 目标函数)
0	26	316	104	231.2
0.1	27	296	105	219.6
0.2	27	294	104	218
0.3	28	295	106	219.4
0.4	28	293	106	218.2
0.5	28	293	106	218.2
0.6	25	281	105	210.6
0.7	25	283	105	211.8
0.8	28	285	109	214.6
0.9	31	292	115	221.2
1.0	31	292	117	222

由表 4 可以看出, 参数 λ 的变化直接影响订单分批结果。参数 λ 表示在加权相似度指标中基于商品品项的订单相似度所占的比重, 当 λ 取最小值 0 时, 基于商品品项的相似度所占比重为 0, 此时订单分批主要根据基于货架的相似度, 分批结果对应的工作人员拣选商品的总次数最多, AGV 搬运货架的总次数最少; 当 λ 由小到大变化时, 基于商品品项的相似度所占比重越来越大, 基于货架的相似度所占比重越来越小, 因此分批结果对应的工作人员拣选商品的总次数呈现递减趋势, AGV 搬运货架的总次数呈现递增趋势; 当 λ 取最大值 1 时, 基于商品品项的相似度所占比重达到最大值 1, 基于货架的相似度所占比重为 0, 此时的分批结果对应的工作人员拣选商品的总次数达到最少, AGV 搬运货架的总次数达到最多。因此, λ 的取值很好的体现了订单拣选过程中工作人员的拣选

成本和 AGV 搬运货架的成本所占的比重。由于订单分批问题整数规划模型的目标函数是极小化订单拣选总成本, 为了分析最优的订单分批结果与参数 λ 取值的关系, 本文直接取 $c_1 = 0.6, c_2 = 0.4$ 计算各种分批结果对应的订单拣选总成本 (见表 4 最后一列), 从总成本的变化情况可以看出, 当 $\lambda = 0.6$ 时的分批结果对应的拣选总成本达到最小。此时的参数 λ 与 c_1, c_2 的取值恰好满足下面的关系式 (14)。

$$\frac{c_1}{c_2} = \frac{\lambda}{1 - \lambda} \quad (14)$$

这说明, 在实际进行订单分批时, 可以直接根据工作人员从货架上拣取一种商品的成本 c_1 和 AGV 搬运一个货架的成本 c_2 , 利用公式 (14) 确定 λ 的取值, 即取 $\lambda^* = \frac{c_1}{c_1 + c_2}$, 然后根据 λ^* 计算加权相似度, 进一步利用贪婪算法求解订单分批问题的近似最优解。

本例中当 $\lambda^* = 0.6$ 时, 利用贪婪算法得到的订单分批结果为: 将 100 个订单分为 25 个批次, 各个批次中包含的订单序号见表 5, 其对应的订单拣选总成本 (整数规划目标函数值) 210.6 达到最小。

表 5 $\lambda = 0.6$ 时的订单分批结果

批次 序号	各个批次中包含 的订单序号	批次 序号	各个批次中 包含的订单序号
1	1, 65, 86	14	18, 33, 55, 57, 79, 88
2	2, 63, 66, 84, 90	15	19, 27, 34, 78
3	3, 12, 28, 76, 87, 89	16	20, 22, 24, 29, 53, 67, 68
4	4, 47, 51	17	23, 58, 70
5	5, 7, 21, 60	18	25, 45, 50
6	8, 37, 40, 42, 48, 85, 94	19	30, 74
7	9, 38, 61, 73, 81	20	36, 96
8	10, 26, 49, 83, 93, 92, 97	21	39, 91
9	11, 16, 31, 32, 41, 46, 59	22	56, 82, 95
10	13, 35, 80, 100	23	64, 98, 99
11	14, 62	24	69, 77
12	15, 44	25	6, 52, 75
13	17, 43, 54, 71, 72		

5.3 贪婪算法的计算时间和计算效果分析

为了分析贪婪算法的运算时间和计算效果, 本节模拟产生了一批不同参数下的小规模算例, 分别利用 Lingo 软件编程求解整数规划模型得到精确最优解, 然后利用贪婪算法求出近似最优解, 并将精确最优解和近似最优解对比, 分析各种参数发生变化时贪婪算法的计算时间和近似比的变化规律。

(1) 参数 λ 发生变化时的求解效果分析

为了分析参数 λ 发生变化对贪婪算法求解效

果的影响,模拟生成 20 个订单、20 种商品、5 个货架的小规模算例。首先取不同参数 c_1, c_2 ($c_1 + c_2 = 1$) 利用 Lingo 软件编程直接求解本文的整数规划模型得到精确最优解,然后令参数 $\lambda = c_1$ 利用贪

婪算法求出近似最优解,并将各种参数下贪婪算法得到的近似最优解与精确最优解进行对比。分析两种算法的运算时间和得到的最优解对应的总拣选成本。具体计算结果见表 6。

表 6 参数 λ 发生变化时贪婪算法和 Lingo 精确求解的计算时间和计算效果对比

参数取值		运行时间(秒)		总拣选成本		
$C_1(\lambda)$	$C_2(1-\lambda)$	贪婪算法	精确求解	贪婪算法	精确最优解	贪婪算法的近似比
0	1	0.0583	37	13	13	1.00
0.3	0.7	0.0626	2680	24.1	17.8	1.35
0.5	0.5	0.0391	2786	24.5	19.5	1.26
0.7	0.3	0.0615	5902	30.9	23.8	1.29
1	0	0.0372	769	31	28	1.11

通过分析发现,对于包含 20 个订单的小规模算例(5 个货架,20 种商品),在不同参数下,用 Lingo 直接求解整数规划模型的计算时间差别较大,当 c_1 取 0 或 1 的时候计算时间较短, c_1 取其它值时,计算时间均超过了 20 分钟,最长的时间超过了 1 小时。而贪婪算法在各种参数下的最长计算时间不足 0.1 秒。比较两种方法得到的目标函数值可以看出,对于包含 20 个订单的小规模算例,贪婪算法得到的近似最优解在最坏情况下对应的近似比不超过 1.35。

(2) 订单总数发生变化时的求解效果分析

表 7 订单数发生变化时贪婪算法和 Lingo 精确求解的计算时间和计算效果对比

参数取值	运行时间(秒)		总拣选成本		
订单总数	贪婪算法	精确求解	贪婪算法	精确最优解	贪婪算法的近似比
5	0.134	0.010	5.5	5.5	1
10	0.197	0.100	17	14	1.21
15	0.217	2.000	20	19	1.05
20	0.356	3600	24.5	23.5	1.04
25	0.416	3600	34	26	1.31

(3) 物品总数发生变化时的求解效果分析

为了分析物品总数变化对求解效果的影响,固定 20 个订单、5 个货架,固定 $c_1 = c_2 = \lambda = 0.5$,选

为了分析订单总数变化对求解效果的影响,对于 20 种商品、5 个货架的算例,固定 $c_1 = c_2 = \lambda = 0.5$,选取不同的订单数目生成算例,分别进行求解,结果见表 7。

从表 7 可以看出,当订单数从 5 增加到 25 时,贪婪算法的求解时间变化不大,均未超过 1 秒钟,但 Lingo 软件的求解时间增长迅速,当订单总数为 20 和 25 时,Lingo 软件求解整数规划模型需要的时间均达到 1 小时。随着订单数量的增加,贪婪算法的近似比会有小幅度的波动,最坏情况下的近似比不超过 1.31。

取不同的物品总数并按照每个货架上存放的商品数量相同的原则生成算例,分别进行求解,结果见表 8。

表 8 物品数发生变化时贪婪算法求解和 Lingo 软件精确求解的计算时间和计算效果对比

参数取值	运行时间(秒)		总拣选成本		
物品总数	贪婪算法	精确求解	贪婪算法	精确最优解	贪婪算法的近似比
10	0.326	760	20.5	20	1.03
15	0.360	322	22	22	1
20	0.351	7200	24.5	23.5	1.04
25	0.371	7200	26	27	0.96
30	0.402	7200	27	28	0.96

从表 8 可以看出,随着物品总数的增加精确求解的计算时间增长迅速,当物品总数为 20 种时,精确求解的计算时间已经超过 2 小时,当物品总数为 25 种时,精确求解需要的时间超过 5 个小时,由于运算时间太长,我们记录了 Lingo 软件运行 2 小时得到的局部最优解与贪婪算法的求解结果进行对比,发现贪婪算法得到的近似最优解优于 Lingo 软件运行 2 小时得到的局部最优解。

(4) 货架总数发生变化时的求解效果分析

为了分析货架总数变化对求解效果的影响,固

表 9 货架数量发生变化时贪婪算法和 Lingo 精确求解的计算时间和计算效果对比

参数取值	运行时间(秒)		总拣选成本		
	贪婪算法	精确求解	贪婪算法	精确最优解	贪婪算法的近似比
2	0.379	41	17	17	1
4	0.373	762	21.5	21	1.02
5	0.370	7200	24.5	23.5	1.04
8	0.380	7200	27.5	28	0.98
10	0.383	7200	28.5	29	0.98

由表 7 ~ 9 中的计算结果可以看出, Lingo 软件求解整数规划模型需要的运算时间随着物品总数、订单总数、货架总数的增加呈指数增长,而贪婪算法在求解小规模问题时运算时间均未超过 1 秒钟,且贪婪算法得到的近似比均未超过 1.31。

对于前一节中的 100 个订单、100 种商品的算例,我们也尝试用 Lingo 软件编程直接求解整数规划模型,结果发现 Lingo 程序运行 24 小时仍然没有得到最终结果,而用贪婪算法运行时间不足 5 秒就可以得到近似最优解,这说明贪婪算法适合求解大规模订单分批问题。由于实际中的订单分批问题都是大规模的在线问题,必须在短时间内得到近似最优结果,因此本文设计的贪婪算法是求解这类问题的有效方法。

6 结论

本文研究了基于 AGV 的智能仓库系统订单分批拣选问题,分析了智能仓库中影响订单拣选效率和拣选成本的两种主要因素分别是工作人员从货架上拣取商品的次数和 AGV 搬运货架次数,以订单拣选总成本极小化为目标建立了订单分批问题的整数规划模型,进一步根据订单中包含的商品信息和商品对应的货架信息构建了订单相似度指标,并设计了基于订单之间的加权相似度贪婪算法。最后利用具体算例进行了模拟计算,分析了加权系

数 λ 的变化对订单分批结果的影响,以及加权系数 λ 的取值与工作人员拣取一件商品的成本 c_1 和 AGV 搬运一次货架的成本 c_2 之间的关系,得到了贪婪算法中加权系数 λ 的确定方法。进一步分析了物品总数、订单总数、货架总数等发生变化时,贪婪算法的计算时间和计算效果。结果显示,本文设计的贪婪算法具有较高的求解效率,由于本文的模型和算法同时考虑了商品拣取成本和货架搬运成本,通过适当选取加权系数,利用贪婪算法可以得到总拣选成本最低的订单分批结果。

由表 9 可以看出,当货架总数为 5 时,用 Lingo 软件精确求解的运算时间已经超过 2 小时,对于货架总数超过 5 的算例,表 9 中记录了 Lingo 软件运行 2 小时得到的局部最优解,可以看出,贪婪算法在不足 1 秒的时间内得到的近似最优解优于 Lingo 软件运行 2 小时得到的局部最优解。

本文结合我国电商企业智能物流系统的实际业务场景开展研究,目前,京东、菜鸟等大型电商平台已经实现了分拣作业中心无人化,物流系统根据订单发布任务,就近调配机器人;机器人搬运货架至拣选台,机器人可以自由旋转保证货架上的商品直接调配到拣选员面前;完成拣选任务后,系统可以根据实时订单信息变化动态调整货架的存放位置。本文给出的订单分批方法为设计智能物流系统后台控制系统提供了理论依据,本文的方法不仅适用于解决基于 AGV 的智能仓库系统订单分批问题,对于解决其他类型的无人仓系统(如基于旋转货架和传送带的无人仓系统)中的订单分批问题,同样具有参考价值。

本文在研究订单分批问题时做了一些必要的简化假设,如没有考虑商品缺货、大订单拆分、紧急订单插单等实际情况,实际中订单拣选过程中可能

遇到某些畅销商品缺货的情况;由于每个货位上的商品数量有限,有些订购商品数量较多的订单可能需要拆分多个拣选单;而且不同订单的优先级不同,遇到优先级高的订单往往需要加急处理等情况。以上情况在本文研究的问题中均未考虑,后续研究中可以进一步考虑这些因素,扩展本文的模型和算法。另外,本文设计的贪婪算法还有很大的改进空间,后续研究中可以针对模型的特点,改进贪婪算法或设计近似度更高的智能算法。

参考文献:

- [1] 张彩霞,吴永前.“货到人”模式下的电商订单拣选优化研究[J]. 经营与管理,2015(9):124-127.
- [2] Peter Wurman R, Raffaello D' Andrea, and Mick Mountz. Coordinating Hundreds of Cooperative, autonomous vehicles in warehouses[J]. AI Magazine, 2008, 29(1): 9-20.
- [3] Sebastian Henn. Algorithms for on-line order batching in an order picking warehouse[J]. Computers & Operations Research, 2012, 39(11): 2549-2563.
- [4] Sören Koch, Gerhard Wäscher. A grouping genetic algorithm for the order batching problem in distribution warehouses[J]. Journal of Business Economics, 2016, 86(1-2): 131-153.
- [5] Jason Chao-Hsien Pa, Po-Hsun Shih, Ming-Hung Wu. Order batching in a pick-and-pass warehousing system with group genetic algorithm[J]. Omega, 2015, 57: 238-248.
- [6] 马廷伟,雷全胜,李军,等. 基于启发式算法的订单分批问题研究[J]. 物流工程与管理, 2015(11): 105-108.
- [7] 马超,雷斌,陈洪满. 一种配送中心订单分拣优化问题的研究[J]. 吉林师范大学学报(自然科学版), 2014(3):118-121.
- [8] Tzu-Li Chen, Chen-Yang Cheng, Yin-Yann Chen, et al. An efficient hybrid algorithm for integrated order batching, sequencing and routing problem[J]. International Journal of Production Economics, 2015, 159: 158-167.
- [9] 胡小建,韦超豪. 基于 Canopy 和 k-means 算法的订单分批优化[J]. 合肥工业大学学报(自然科学版), 2017,40(3):414-419.
- [10] 魏伟. 订单分批及并行化拣选路径的研究[J]. 电子技术与软件工程,2014(9):181-182.
- [11] 卢烨彬,刘少轩. 随机存储机制下基于引力模型的订单批次划分方法的研究[J]. 管理现代化,2016,36(4):101-105.
- [12] 王艳艳. 并行自动分拣系统分拣任务及补货缓存优化研究[D]. 山东大学,2012.
- [13] 王善超. A 公司配送中心订单拣选优化与仿真[D]. 东华大学,2017.
- [14] Lenoble Nicolas, Frein Yannick, Hammami Ramzi. Order batching in an automated warehouse with several vertical lift modules: optimization and experiments with real data [J]. European Journal of Operational Research, 2018, 267(3): 958-976.
- [15] 王旭坪,张珺,易彩玉. B2C 电子商务环境下订单拣选与配送联合调度优化[J]. 中国管理科学,2016,24(7):101-109.
- [16] Xiang X, Liu C, Miao L. Storage assignment and order batching problem in Kiva mobile fulfillment system[J]. Engineering Optimization, 2018(4): 1-22.