

考虑商品订购数量的“货到人”仓储系统 订单分批问题研究^{*}

李珍萍 韩倩倩

(北京物资学院信息学院, 北京 101149)

摘要 结合“货到人”仓储系统的订单拣选场景, 在考虑订单中各种商品订购数量和货箱中商品存储量的情况下, 研究了自动小车存储及取货系统的订单分批拣选问题. 对于给定的待拣选订单, 以货箱出库次数极小化为目标, 建立了订单分批问题的整数规划模型, 并利用聚类思想设计了两阶段启发式算法. 利用不同规模的算例进行仿真实验, 验证了模型和算法的有效性. 通过对比按照本文模型和算法得到的分批结果与按照先到先服务策略得到的分批结果, 可以发现, 按照文章模型和算法进行订单分批, 拣选效率大约提升了 25%–45%. 进一步分析了拣选台容量和相似度加权系数等参数变化对订单分批结果的影响. 最后利用两个具体算例, 对比分析了考虑商品订购数量和不考虑商品订购数量的订单分批模型之间的关系, 验证了考虑商品订购数量的订单分批模型的优越性.

关键词 货到人, 自动小车存储及取货系统, 商品订购数量, 订单分批, 两阶段启发式算法.

MR(2000) 主题分类号 90B08

Study on the Order Batching Problem of “Parts-to-Picker” Warehouse System Considering the Quantity of Items in Orders

LI Zhenping HAN Qianqian

(School of Information, Beijing Wuzi University, Beijing 101149)

Abstract Based on the scene of “parts-to-picker” warehouse system, the order batching problem of autonomous vehicle storage and retrieval system is studied by considering the quantity of every item in orders and SKU bins. Given a set of orders to be picked, the order batching problem is formulated into an integer programming model, the goal of the model is to minimize the number of SKU bins to be transported. A two-stage heuristic algorithm is designed based on the clustering idea. Simulation

^{*} 国家自然科学基金项目 (71771028), 北京市自然科学基金项目 (Z180005), 2018 年北京市属高校高水平创新团队支持计划项目 (IDHT20180510), 北京市智能物流协同创新中心开放课题 (BILSCIC-2019KF-18), 科技创新服务能力建设 - 高精尖学科建设 (市级), 北京物资学院校级重大课题 (2019XJZD09) 资助.

收稿日期: 2019-11-21, 收到修改稿日期: 2020-01-16.

编委: 房勇.

experiments are carried out on different size of examples to verify the effectiveness of the model and algorithm. By comparing with the order batching results obtained by the first-come-first-served strategy, we find that the picking efficiency can be improved by 25%–45% based on the batching results obtained by the model and algorithm of this paper. Furthermore, the sensitivity analysis is done on parameters of the picking station's capacity and the weighted coefficient of similarity. Finally, the relationship between the order batching model considering the quantity of items in orders and the order batching model without considering the quantity of items in orders is compared and analyzed by using two examples. And the superiority of the order batching model considering the quantity of items in orders is verified.

Keywords Parts-to-picker, autonomous vehicle storage and retrieval system, quantity of items, order batching, two-stage heuristic algorithm.

1 引言

“货到人”订单拣选系统利用先进的自动化技术代替传统的人工操作,提高了配送中心的工作效率,减少了作业成本。同时,该系统具有拣选效率高,工作准确性高与智能化程度高等特点。因此“货到人”订单拣选系统在物流行业具有很好的应用前景,目前已经在大型电商,医药物流,冷链物流等领域得以应用^[1]。

在电商和零售等企业,小件拆零货物的订单量激增,对“货到人”系统拆零拣选的需求急速提升。自动小车存储及取货系统是“货到人”订单拣选系统的一种主要形式,与应用较早的自动存储与取货系统相比,能更好地满足电商或零售企业的拆零拣选需求。对订单进行分批拣选可以有效提高拣选效率,将适当的订单分批策略应用于自动小车存储及取货系统,可以有效减少货箱的出库次数,降低拣选成本,提高拣选效率^[1]。

目前,国内外对订单分批问题的研究主要有两个方向:“人到货”拣选系统的订单分批问题和“货到人”拣选系统的订单分批问题。“人到货”系统的订单分批问题研究较为成熟,在该系统中,优化目标多为行走路径最短,以使拣选时间最短^[2]。李诗珍和杜文宏^[3]研究订单分批问题时,在聚类分析的基础上提出了三种相似系数的计算公式。罗晓萌等^[4]分别运用通道重合数和通道重合率作为相似系数,对订单分批问题进行静态与动态求解。王转和裴泽平^[5]研究了整箱拣选模式下的订单分批问题,以里程节约量最大化为目标建立模型,并用启发式算法求解。王旭坪等^[6]研究了基于订单完成期限的固定时间窗订单分批问题,并考虑了订单紧急程度与订单相似度等因素,设计启发式算法进行求解分析。Hwang等^[7]研究了自动存储及取货系统的订单分批问题,先使用拣选机器进行批次总量拣选,再手工二次分拣。以拣选机器行驶路程最短为目标建立模型,基于聚类算法思想进行求解。Menéndez等^[8]提出了几种基于变邻域搜索的策略来解决订单分批问题,通过两阶段变邻域搜索较好地实现模型目标。Muter和Öncan^[9]认为订单分批是一种集合划分问题,并针对该问题提出了一种基于列生成的算法。Scholz等^[10]研究了同时解决订单分批,批次分配,排序与拣选路径问题的方法,提出了变邻域下降算法并验证其可行性。Zhang等^[11]研究了在线订单分批与批次分配的集成问题,提出了基于混合规则的算法解决该问题。

由于“货到人”系统是近年来快速发展的一种新型仓储作业模式,对于“货到人”系统的订单分批问题研究相对较少。Füßler和Boysen^[12]研究了自动存储及取货系统订单分批

后的批次内部订单排序问题,建立了以最小化拣选员拣选次数为目标的混合整数规划模型. Boysen 等^[13]研究了基于移动机器人的订单拣选系统的批次内部订单排序问题,建立了以最小化移动货架到访拣选台的次数为目标的混合整数规划模型. Nicolas 等^[14]研究了带有多个垂直升降机模块的自动化立体仓库拣选系统的订单分批优化问题时,以最小化订单完成时间为目标建立了0-1整数规划模型. Xiang 等^[15]研究了KIVA系统中的订单分批问题,建立了以货架到访次数最小化为目标的整数规划模型,并用启发式算法求解.

以上针对“货到人”系统的订单分批问题研究成果很好地解决了待拣选订单中每种物品订购数量较少,货箱中存放的同种物品数量充足情况下的订单分批问题^[13, 15]. 当每个货箱中存放的每种物品数量均可以满足待拣选订单对该物品需求的时候,拣选每批次订单中的一种物品,只需要出库一个货箱即可,此时不需要考虑订单中各种商品的订购数量^[15]. 实际中,由于每个货箱中包含的同一种物品数量有限,如果一批订单中对某种商品的订购数量超过一个货箱中存放的该商品总量,则一个货箱中的物品无法满足该批次订单的拣选需求,需要从多个货箱中拣选同一种物品,现有研究成果均未考虑这种情况. 本文将在考虑订单中各种商品订购数量的前提下,研究自动小车存储及取货系统订单分批拣选问题,建立数学模型,并设计求解模型的算法,推广现有文献的研究成果.

2 自动小车存储及取货系统

本文研究的“货到人”订单拣选系统主要由箱式自动化立体仓库,多层穿梭小车,提升机,出入库缓存区,输送设备,拣选员与拣选台组成,如图1所示. 该系统以存放在货架上的标准化货箱作为基本存储单元,每个货箱可放置一种或多种货物. 同一个巷道两边的货架共用多层穿梭小车,出入库缓存区与提升机. 在货架的每层都配置穿梭小车与出入库缓存区,穿梭小车沿着巷道轨道行驶,在货架与出入库缓存区之间输送货箱. 每组货架利用一台提升机垂直输送货箱,完成货箱出入库的操作. 传送带将不同巷道的货箱输送至指定拣选区域,并将拣选完成的货箱回库. 拣选区域的多个拣选台可并行进行订单拣选工作,每个拣选台可以容纳有限个订单箱同时进行拣选.

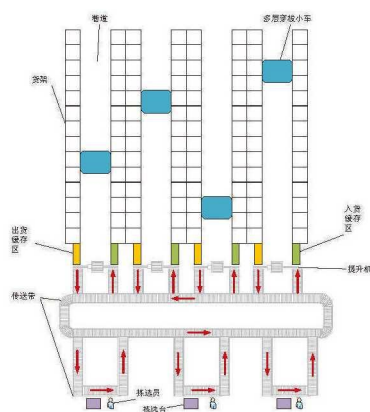


图1 拣选系统布局图

(Figure 1 Schematic layout of picking system)

由于每个拣选台的面积有限,可以摆放的订单箱数量有限;通常每个订单箱只能存放一

个订单中的商品, 因此拣选员可以同时处理的订单数量有限. 系统在进行订单分批拣选时, 每批次订单的数量不能超过拣选台容量. 系统每隔一段时间会对订单池中的订单进行分批, 并把分批后的拣选任务分配给各个拣选台. 然后根据每个批次订单中包含的商品信息确定需要输送的货箱, 指令传输系统 (穿梭小车, 提升机, 传送带等共同协作) 将货箱输送到拣选台供拣选员拣取商品, 完成拣选任务之后的货箱仍然由输送系统送回仓库. 图 2 描述了单拣选台的订单分批拣选过程.

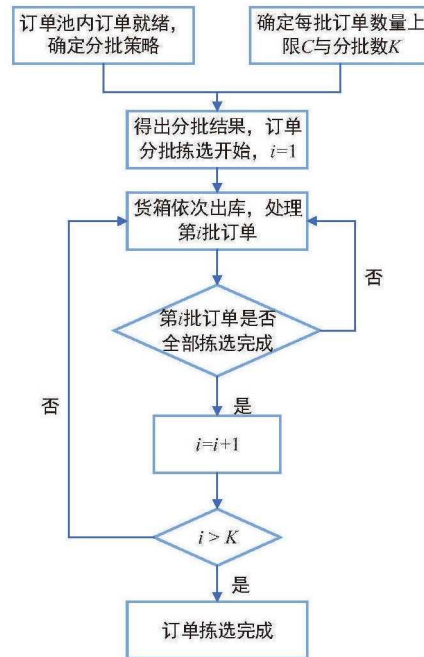


图 2 单拣选台分批拣选工作流程图

(Figure 2 Flowchart of single picking station picking in batches)

3 “货到人”拣选系统订单分批问题的数学模型

3.1 问题描述与分析

本文研究的基于自动小车存储及取货订单拣选系统包含一个拣选台. 已知仓库中 M 个货箱共存储了 T 种货物, 每个货箱的最大存储量为 Q , 一个货箱可存储一种或多种货物, 已知每个货箱中存储的各种货物数量. 在一段时间内, 该配送中心共接到 N 个订单 $O = \{O_1, O_2, \dots, O_N\}$, 已知每个订单 O_i ($i = 1, 2, \dots, N$) 中包含的货物种类及数量 $O_i = (d_{i1}, d_{i2}, \dots, d_{iT})$ (d_{it} 表示订单 i 中包含货物 t 的数量). 由于拣选台容量有限, 工作人员每次最多可以同时处理 C 个订单. 问如何将订单分批并确定拣选每个批次订单需要输送的货箱, 才能使拣选效率最高.

货箱的出入库次数是影响拣选效率的重要因素, 在“货到人”订单拣选系统中假设系统存取速度稳定, 即每个货箱出入库的时间大致相同. 拣选员拣取每件货物的速度相同, 拣取

货物时的移动距离可忽略不计,即拣取订单中所有货物的时间由订单中货物总数量决定.基于以上假设,在订单拣选过程中,减少货箱出入库次数,可以缩短订单拣选时间,提高拣选效率.因此订单分批问题的目标可以归结为减少货箱出入库次数.为了简化问题,本文假设仓库中所有货箱中存放的同种商品总数量均可以满足所有订单对该商品的总需求,但一个货箱中存放的每种物品数量未必能满足所有订单的拣选需求.

假设本文研究的“货到人”系统使用标准化规格的订单箱,每个订单箱的体积相同,为了方便对同一个订单中订购的所有商品进行打包,拣选过程中每个订单箱中只能收集一个订单中的商品.由于拣选台的面积有限,可以同时容纳的订单箱数量有限,因此每个批次的订单数量不能超过拣选台的容量.假设订单分批之后,拣选过程严格按照批次进行,即当一个批次的订单全部拣选完之后,才能拣选下一个批次的订单.基于以上分析,拣选台的容量限制等同于每个批次最多可以同时拣选的订单数量.

3.2 模型建立

索引

i : 订单索引, $i = 1, 2, \dots, N$;

k : 批次索引, $k = 1, 2, \dots, K$;

m : 货箱号索引, $m = 1, 2, \dots, M$;

t : 货物种类索引, $t = 1, 2, \dots, T$;

参数

C : 拣选台容量;

K : 最大批次数量, 且 $K = \lceil N/C \rceil$;

Q : 货箱最大存储量;

c_{mt} : 货箱 m 存放货物 t 的数量, 且 $\sum_{t=1}^T c_{mt} \leq Q$;

d_{it} : 订单 i 需要货物 t 的数量;

决策变量

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

$$y_{mk} = \begin{cases} 1, & \text{拣选批次 } k \text{ 需要运送货箱 } m, \\ 0, & \text{否则;} \end{cases}$$

s_{mkt} : 拣选批次 k 时从货箱 m 中拣取货物 t 的数量;

则订单分批问题可以表示成如下整数规划模型

$$\min \quad z = \sum_{m=1}^M \sum_{k=1}^K y_{mk} \quad (3.1)$$

$$\sum_{k=1}^K x_{ik} = 1, \quad \forall i, \quad (3.2)$$

$$\sum_{i=1}^N x_{ik} \leq C, \quad \forall k, \quad (3.3)$$

$$\sum_{i=1}^N d_{it} x_{ik} = \sum_{m=1}^M s_{mkt}, \quad \forall k, t, \quad (3.4)$$

$$\sum_{t=1}^T s_{mkt} \leq Q y_{mk}, \quad \forall k, m, \quad (3.5)$$

$$\sum_{k=1}^K s_{mkt} \leq c_{mt}, \quad \forall m, t, \quad (3.6)$$

$$x_{ik}, y_{mk} \in \{0, 1\}, \quad \forall i, m, k, \quad (3.7)$$

$$s_{mkt} \in Z^+, \quad \forall m, k, t. \quad (3.8)$$

目标函数 (3.1) 表示最小化货箱到访拣选台的总次数. 约束条件 (3.2) 表示每个订单只能被分到一个批次中. 约束条件 (3.3) 表示每批次包含的订单数量不超过拣选台容量. 约束条件 (3.4) 表示任一批次的每种货物需求都要满足. 约束条件 (3.5) 表示如果拣选批次 k 时从货箱 m 中拣取某种货物, 则需要将货箱 m 输送至拣选台. 约束条件 (3.6) 表示从货箱 m 中拣取的货物 t 的总数量不超过货箱中的存储量. 约束条件 (3.7) 和 (3.8) 为决策变量取值约束.

通过求解以上整数规划, 不仅可以得到订单分批结果, 而且可以得到拣选每批订单需要输送的货箱序号.

示例 已知仓库中有 6 个货箱, 存储着 8 种物品, 每个货箱的最大存储量为 10, 每个货箱存储各种货物的数量如图 3 所示, 其中字母表示货物代码, 括号里的数字表示货物的存储数量. 假设某段时间内收到 9 个订单, 每个订单中包含的各种货物代码及数量分别为

$$O_1 = \{A(2), E(1)\}, O_2 = \{C(1), F(2), H(4)\}, O_3 = \{B(3), C(2)\}, O_4 = \{A(4)\},$$

$$O_5 = \{B(2), C(1), F(1)\}, O_6 = \{A(1), B(2), D(2)\}, O_7 = \{F(1), H(4)\},$$

$$O_8 = \{A(5)\}, O_9 = \{D(2), F(2), G(1)\}.$$

假设拣选台容量 $C = 3$, 直接利用商业软件求解订单分批问题的整数规划模型, 得到如下分批结果: O_4, O_7, O_8 为第一批, O_1, O_6, O_9 为第二批, O_2, O_3, O_5 为第三批. 按照这种分批结果进行拣选, 总共需要输送货箱出库 9 次, 具体拣选过程如图 4 所示. 其中, 拣选第一批订单需要输送货箱 1 与货箱 4. 拣选第二批订单需要货箱 3, 货箱 5 与货箱 6. 拣选第三批订单需要货箱 2, 货箱 3, 货箱 4 与货箱 6. 拣选完每批次的订单以后, 回库时货箱中货物存储情况见每个小图的右上角, 如拣选完批次 1 之后, 货箱 1 中只剩 1 件 A 货物. 拣选完批次 2 之后, 货箱 6 中剩余物品为 $\{G(4), F(3)\}$, 拣选完批次 3 之后, 货箱 6 中剩余物品为 $\{G(4), F(1)\}$.

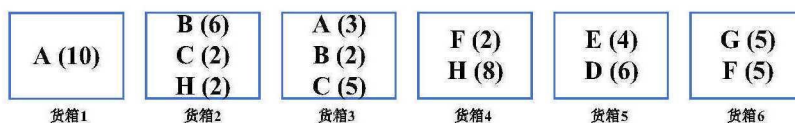


图 3 货物存储情况

(Figure 3 Illustration of storage assignment)

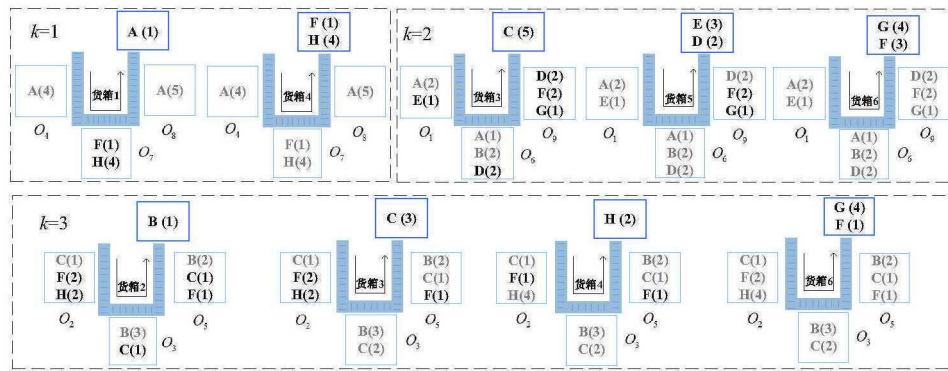


图4 分批拣选过程

(Figure 4 Process of picking orders in batches)

4 求解订单分批问题的启发式算法

订单分批拣选问题的本质是将订单池中的所有订单划分为若干个批次, 并确定拣选每个批次订单需要出库的最少货箱数量, 使得拣选所有订单的总出库货箱数量之和最少. 其中将订单划分为若干批次可以归结为对订单池中的订单进行聚类的问题^[16], 确定拣选每个批次订单需要出库的最少货箱数量可以归结为集合覆盖问题. 由于聚类问题和集合覆盖问题均为复杂的组合优化问题, 当问题规模较大时, 无法在短时间内利用商业软件求解整数规划模型得到最优解. 为此, 需要设计快速求解模型的有效算法.

4.1 启发式算法的思想

为了减少同一批次订单拣选过程中需要出库的货箱次数, 在对订单进行分批的时候, 应该使同一批次的订单中包含的货物种类尽可能少; 或者使同一批次订单中包含的货物尽量存放在相同的货箱中. 基于以上分析, 本文设计启发式聚类算法求解订单分批问题.

首先定义任意两个订单之间的相似度系数, 该系数反映了两个订单包含的货物种类或货物存放的货箱之间的相似程度. 然后利用相似度系数进行聚类, 将相似度较大的订单尽可能分到同一批次中, 以减少货箱的总出入库次数, 缩短订单拣选时间, 提高拣选作业效率.

4.2 订单之间的相似度

本文参考文献[3]的巷道相似系数定义, 提出品项相似系数, 品项相似系数指两订单中订购的相同货物种类数与两订单中的总货物种类数之间的比值. 设 r_i 表示订单 i 中的货物种类集合, 则订单 i 和订单 j 的品项相似系数可以表示为

$$s_{ij} = \frac{|r_i \cap r_j|}{|r_i \cup r_j|}.$$

由于“货到人”系统中, 一个货箱中可以存放多种货物, 存放在同一个货箱中的货物可以同时拣选, 因此可以根据订单中的货物所在的货箱是否相同定义储位相似系数^[7]. 令 $VA_i = (a_{1i}, a_{2i}, \dots, a_{Mi})$ 表示订单 O_i 中的货物在货箱中的存储情况, 其中

$$a_{mi} = \begin{cases} 1, & \text{订单 } O_i \text{ 中存在某种货物存储于货箱 } m \text{ 中,} \\ 0, & \text{否则.} \end{cases}$$

据此定义储位相似系数 s'_{ij} 表示订单 i 和订单 j 中货物的共同储位数与总储位数的比值, 储位相似系数的计算方法为

$$s'_{ij} = \frac{(VA_i)(VA_j)^T}{(VA_i)(VA_i)^T + (VA_j)(VA_j)^T - (VA_i)(VA_j)^T}.$$

将品项相似系数与储位相似系数加权求和, 求得两订单之间的加权相似度. 加权相似度 h_{ij} 的计算公式为

$$h_{ij} = \omega s_{ij} + (1 - \omega) s'_{ij},$$

其中 $0 \leq \omega \leq 1$.

对任意两个订单分别计算加权相似度, 得到订单之间的加权相似度矩阵:

$$H = \begin{bmatrix} 0 & h_{12} & \cdots & h_{1N} \\ h_{21} & 0 & \cdots & h_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N1} & h_{N2} & \cdots & 0 \end{bmatrix}.$$

矩阵中的 h_{ij} 为订单 i 和订单 j 的加权相似度. 为了简化问题, 当 $i = j$ 时, 令 $h_{ij} = 0$.

4.3 两阶段启发式聚类算法

本文设计两阶段启发式聚类算法求解订单分批问题, 第一阶段采用贪婪算法生成初始解, 第二阶段采用大邻域搜索算法改进初始解, 得到近似最优解.

第一阶段: 贪婪算法生成初始解

利用加权相似度矩阵, 设计贪婪算法生成初始解, 算法步骤如下

第 1 步 根据订单加权相似度矩阵对订单进行分批

步骤 1.1 设置加权系数 ω , 计算订单加权相似度矩阵 H .

步骤 1.2 在加权相似度矩阵 H 中找出最大的相似度系数 h_{ij} 及其对应的订单 i 与订单 j . 若 $h_{ij} < 0$, 则结束; 否则, 转步骤 1.3.

步骤 1.3 判断订单 i 与订单 j 是否已经被分配到某个批次中, 并按照以下 4 种情况为订单分配批次.

情况 1 若两个订单都已被分配批次, 直接进入步骤 1.4.

情况 2 若只有一个订单 i (或 j) 已被分配到批次 k , 则将另一个订单 j (或 i) 分配到批次 k , 转步骤 1.4.

情况 3 若两个订单都未被分配批次, 且已形成的拣选批次数小于 K , 则生成一个新的批次 k , 并将订单 i 与订单 j 分配到新批次 k 中, 转步骤 1.4.

情况 4 若两个订单都未被分配批次, 且已形成的拣选批次数等于 K . 如果存在一个批次 k 包含的订单数量不超过 $C-2$, 将订单 i 与订单 j 分配到该批次中; 否则将订单 i 与订单 j 随机分配到两个未饱和批次中, 转步骤 1.4.

步骤 1.4 将订单 i 与订单 j 之间的相似度 h_{ij} 修改为 -1 .

步骤 1.5 判断各个批次包含的订单数量 q 是否等于拣选台容量 C .

若 $q = C$, 则批次 k 中的订单数量已经饱和, 将批次 k 中所有订单在相似度矩阵 H 中对应行和列的相似度全部修改为 -1 , 转步骤 1.2.

若 $q < C$, 直接转步骤 1.2.

第 2 步 利用启发式算法确定拣选每批次订单需要出库的货箱 (计算可行解对应的目标函数值)

步骤 2.1 将每个批次 k 中所有订单包含的各种货物数量用向量 $P_k = (p_{k1}, p_{k2}, \dots, p_{kT})$ 表示, 其中 p_{kt} 表示批次 k 中包含货物 t 的数量. 令 $k = 1$, 转步骤 2.2.

步骤 2.2 选择可以满足批次 k 中最多货物品项需求的货箱 m^* 出库. 其中

$$m^* = \arg \max_m \sum_{t=1}^T (\min\{c_{mt}, p_{kt}\}).$$

更新货箱 m^* 中剩余的货物数量和批次 k 中尚未满足需求的货物数量

$$c_{m^*t} = c_{m^*t} - \min\{c_{m^*t}, p_{kt}\}, \quad \forall t,$$

$$p_{kt} = p_{kt} - \min\{c_{m^*t}, p_{kt}\}, \quad \forall t.$$

步骤 2.3 如果 $\sum_{t=1}^T p_{kt} = 0$, 即批次 k 中包含的所有货物均已拣选完毕, 则令 $k = k + 1$, 转步骤 2.4; 否则, 转步骤 2.2, 继续拣选批次 k 中的货物.

步骤 2.4 如果 $k \leq K$, 转步骤 2.2.

步骤 2.5 结束, 输出拣选每个批次需要出库的货箱顺序, 并计算拣选所有批次订单需要的货箱数量, 即整数规划模型的目标函数值 z .

第二阶段: 利用大邻域搜索算法改进初始解

输入 贪婪算法求得的初始解 (初始订单分批结果) x^0 , 初始目标函数值 (初始货箱出库次数) z^0 , 最大迭代次数 L_{\max} , 移除操作 D , 插入操作 R .

初始化 初始化当前解与当前目标函数值: $x_c = x^0, z_c = z^0$, 初始化迭代次数 $l = 0$.

第 3 步 当 $l \leq L_{\max}$, 循环执行步骤 3.1–3.3

步骤 3.1 任意选择 20% 的批次, 从每个批次中随机选择一个订单进行移除, 得到破坏解 $x_{destroy}$. 将移除的订单放到集合 U 中, U 表示待分批订单集合.

步骤 3.2 对集合 U 中的每一个订单 i , 执行贪婪插入操作, 得到邻域解 x_{repair} . 贪婪插入操作步骤为: 对于一个待插入订单, 计算其与破坏解 $x_{destroy}$ 中各未饱和批次中的所有订单 (包括新插入的订单) 的平均相似度, 选择平均相似度最大的未饱和批次, 将订单 i 插入该批次.

步骤 3.3 计算邻域解 x_{repair} 的目标函数值 z_{repair} . 如果 $z_{repair} < z_c$, 则更新当前解和当前目标函数值, 即令 $x_c = x_{repair}, z_c = z_{repair}$; 否则保持当前解不变.

令 $l = l + 1$.

输出 近似最优解 $x^* = x_c$, 最优目标函数值 $z^* = z_c$.

5 模拟计算与结果分析

为验证本文的整数规划模型与两阶段启发式算法的有效性, 现对多个拣选场景进行仿真计算. 本文采用小, 中, 大三种规模的算例进行模拟计算, 分析求解结果 (货箱出库次数) 和计算时间.

5.1 算例参数设置与模拟计算规则

设置参数货物种类数 T , 货箱数 M , 订单数 N , 货箱最大存储量 Q 与货箱最大存储品项数 P 的不同取值, 分别生成小规模, 中规模和大规模的算例. 各种规模算例对应的基本参数设置如下: 小规模算例 $T/M/N \in \{10/10/30, 10/20/50\}$, 中规模算例 $T/M/N \in \{20/30/50, 20/30/100\}$, 大规模算例 $T/M/N \in \{30/50/100, 30/80/100\}$. 基于以上基本参数, 货箱最大存储量分别取 $Q \in \{15, 20\}$, 货箱最大存储品项数分别取 $P \in \{3, 4\}$, 拣选台容量设置为 $C = 4$, 相似度加权系数 $\omega = 0.5$. 由此形成 24 种参数组合, 利用每种参数组合, 随机生成一个算例, 其中的订单到达顺序为随机产生的.

对于每个算例分别运用 Gurobi 求解器和两阶段启发式聚类算法进行求解, 根据求解结果比较两种方法的求解质量与计算速度. 首先, 使用 Gurobi8.1.1 版本在 Inter Core i5-8265U 的电脑上用 Matlab 环境编程运行, 设定程序最大运行时间为 600 秒, 记录求解结果. 再使用两阶段启发式聚类算法运行 10 次求得平均解, 设置第二阶段的最大迭代次数为 1000. 最后计算按照先到先服务策略 (即按照订单到达先后顺序) 进行分批时对应的货箱出库次数, 并与启发式算法求的平均解对应的货箱出库次数进行对比.

5.2 小规模算例求解结果分析

小规模算例的求解结果如表 1 所示, 第一列表示算例参数设置; 第二列表示用 Gurobi 软件求解整数规划模型, 程序运行 600 秒得到的近似最优解对应的目标函数值 (货箱出库次数); 第三列表示两阶段启发式聚类算法运算 10 次得到的平均解; 第四列表示两阶段启发式聚类算法的平均求解时间; 第五列表示先到先服务策略下订单分批问题的目标函数值; 第六列表示启发式算法的平均解与 Gurobi 求解结果的误差百分比; 第七列表示与先到先服务策略下的订单分批结果对比, 本文分批策略对应的拣选效率提升的百分比.

由表 1 可知, 对于小规模算例, 由本文的两阶段启发式聚类算法求得平均解与 Gurobi 求解误差不超过 10%, 总体平均误差约为 6.7%. 平均求解时间不超过 13 秒, 启发式算法的求解时间明显优于求解器的计算时间. 与先到先服务策略相比, 运用本文的分批策略拣选效率提升约 42.2%.

表 1 小规模算例求解结果
(Table 1 Results of small examples)

$T/M/N/Q/P$	Gurobi 求解结果	启发式算法 平均结果	运算时间	先到先服务 策略结果	算法对比 误差率	效率提升
10/10/30/15/3	19	20	10 s	36	5.3%	44.4%
10/10/30/15/4	17	18	9.8 s	30	5.9%	40%
10/10/30/20/3	16	17	10.9 s	31	6.3%	45.2%
10/10/30/20/4	15	16	10.7 s	26	6.7%	38.5%
10/20/50/15/3	28	30	16.3 s	51	7.1%	41.2%
10/20/50/15/4	25	26	16 s	42	4%	38.1%
10/20/50/20/3	24	26	15.4 s	46	8.3%	43.5%
10/20/50/20/4	20	22	14.4 s	41	10%	46.3%
平均			12.9 s	/	6.7%	42.2%

5.3 中、大规模算例结果分析

对于中大规模算例的求解结果如表 2 所示, 各列含义与表 1 相同. 由于 Gurobi 软件无法在短时间内得到整体最优解, 本文设置 Gurobi 软件的运行时间为 600 秒. 将启发式算法求得平均解与 Gurobi 程序运行 600 秒的结果进行相比, 总体平均误差约为 5.4%. 随着算例规模的增大, 启发式算法的求解时间变长, 总体平均求解时间约为 46.8 秒. 与先到先服务策略相比, 运用本文的分批策略使中大规模算例的拣选效率提升约 33.2%. 由于该算法的求解时间与第二阶段迭代次数有关, 在问题求解中, 可调整迭代次数以获得较满意的求解时间与求解精度.

表 2 中大规模算例求解结果
(Table 2 Results of medium and large examples)

$T/M/N/Q/P$	Gurobi 求解结果	启发式算法 平均结果	运算时间	先到先服务 策略结果	算法对比 误差率	效率提升
20/30/50/15/3	38	39	20 s	65	2.6%	40.0%
20/30/50/15/4	37	38	20.3 s	57	2.7%	33.3%
20/30/50/20/3	36	38	20.6 s	66	5.6%	42.4%
20/30/50/20/4	32	33	19.2 s	50	3.1%	34.0%
20/30/100/15/3	79	84	44.4 s	133	6.3%	36.8%
20/30/100/15/4	76	81	45.3 s	109	6.6%	25.7%
20/30/100/20/3	69	78	43.2 s	116	13%	32.8%
20/30/100/20/4	65	68	41 s	100	4.6%	32.0%
30/50/100/15/3	88	98	60.6 s	145	11.3%	32.4%
30/50/100/15/4	90	95	58.4 s	131	5.6%	27.5%
30/50/100/20/3	81	90	57.9 s	141	11.1%	36.2%

续表 2 中大规模算例求解结果
(Table 2 Results of medium and large examples (Continued))

$T/M/N/Q/P$	Gurobi	启发式算法		先到先服务	算法对比	
	求解结果	平均结果	运算时间	策略结果	误差率	效率提升
30/50/100/20/4	80	83	54.3 s	123	3.8%	32.5%
30/80/100/15/3	97	93	64.5 s	131	4.1%	29.0%
30/80/100/15/4	84	84	67.2 s	119	0	29.4%
30/80/100/20/3	84	89	68.8 s	136	5.9%	34.6%
30/80/100/20/4	76	76	63.2 s	113	0	32.7%
平均			46.8 s	/	5.4%	33.2%

5.4 灵敏度分析

拣选台容量限制了可同时拣选的订单数量, 相似度加权系数的取值直接影响订单的相似度测量, 因此这两个参数的变化会影响订单分批结果. 本节分别从小, 中, 大规模算例中选取一个算例进行灵敏度分析.

图 5 描述了拣选台容量变化时, 货箱出入库次数的变化情况. 当相似度加权系数固定, C 从 3 增加到 6, 三个算例中货箱出入库的次数都逐渐减少. 这是因为随着拣选台容量的增加, 每个批次可以同时拣选的订单数量增大, 拣选批次数量相应减少. 此时同一个货箱需要出库的次数减少, 因此拣选所有订单需要出库的货箱总次数必然减少.

图 6 描述了拣选台容量固定, 相似度加权系数变化时, 货箱出入库次数的变化情况. 从图 6 可以看出, 当拣选台容量固定时, 相似度加权系数的变化对货箱出库次数的影响没有明显的规律. 换句话说, 品项相似度与储位相似度两个聚类指标对结果的影响依赖于具体的数据, 没有统一的变化规律. 实际中应该根据具体情况确定相似度系数的取值.

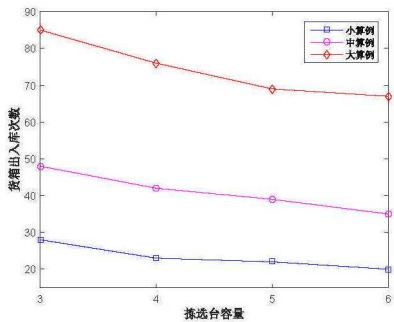


图 5 货箱出入库次数与拣选台容量的变化关系
(Figure 5 The relationship between the number of SKU bins to be transported and the capacity of the picking station)

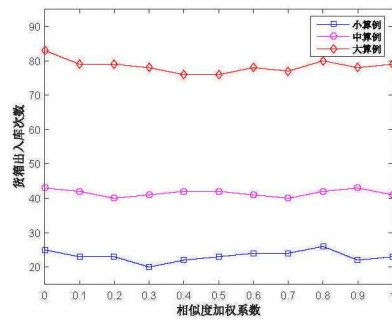


图6 货箱出入库次数与相似度加权系数的变化关系

(Figure 6 The relationship between the number of SKU bins to be transported and the weighted coefficient of similarity)

5.5 与不考虑商品订购数量的订单分批模型对比

文献 [15] 研究“货到人”系统订单分批问题时, 假设一个货箱中存放的物品数量可以满足所有订单对该物品的需求, 此时只需要考虑订单中订购的各种商品种类, 不需要考虑订单中各种商品订购数量. 为了便于对比分析, 本文将文献 [15] 中的模型用 3.2 节中的符号表达如下

$$\min z = \sum_{m=1}^M \sum_{k=1}^K y_{mk}, \quad (5.9)$$

$$\sum_{k=1}^K x_{ik} = 1, \quad \forall i, \quad (5.10)$$

$$\sum_{i=1}^N x_{ik} \leq C, \quad \forall k, \quad (5.11)$$

$$\sum_{i=1}^N d_{it} x_{ik} \leq T u_{tk}, \quad \forall k, t, \quad (5.12)$$

$$\sum_{m=1}^M c_{mt} y_{mk} \geq u_{tk}, \quad \forall k, t, \quad (5.13)$$

$$x_{ik}, y_{mk}, u_{tk} \in \{0, 1\}, \quad \forall i, m, k, t, \quad (5.14)$$

其中参数 d_{it} 取值 1 或 0 分别表示订单 i 中是否订购货物 t , 参数 c_{mt} 取值 1 或 0 分别表示货物 t 是否存储在货箱 m 中, 0-1 决策变量 u_{tk} 表示批次 k 中是否订购货物 t .

本文模型 (简称模型一) 和文献 [15] 中的模型 (简称模型二) 的目标函数完全相同, 表示极小化货箱或货架的出库次数; 模型一的约束条件 (3.2), (3.3) 与模型二的约束条件 (5.10), (5.11) 相同, 分别表示订单分配批次限制与每个批次中的订单数量限制. 两个模型的不同点在于: 模型二的约束条件 (5.12), (5.13) 表示如果批次 k 中包含商品 t , 则只需要搬运一个包含商品 t 的货架出库, 即可满足该批次中商品 t 的拣选需求; 模型一中同时考虑了订单中包含的各种商品数量和货箱中商品存储商品数量, 引入了一个组新的决策变量 s_{mkt} 表示拣选

批次 k 时从货箱 m 中拣取商品 t 的数量, 并通过约束条件 (3.4)–(3.6), 表示从出库货箱中拣取的商品数量必须满足每个批次中商品数量的需求. 显然, 考虑商品数量的模型比文献 [15] 中不考虑商品数量的模型更加复杂.

下面通过具体算例对两个模型的求解结果进行对比分析.

利用 3.2 节示例的数据, 直接利用商业软件求解模型二, 得到分批结果与货箱出库结果, 并将其与利用本文模型得到的求解结果进行对比, 具体信息如表 3 所示.

表 3 两种模型的计算结果对比 (1)
(Table 3 Computational compare of two models (1))

		模型一	模型二
		考虑商品订购数量	不考虑商品订购数量
订单分批 结果	第 1 批	O_4, O_7, O_8	O_3, O_4, O_8
	第 2 批	O_1, O_6, O_9	O_1, O_6, O_9
	第 3 批	O_2, O_3, O_5	O_2, O_5, O_7
货箱出库 结果	第 1 批	货箱 1, 4	货箱 3
	第 2 批	货箱 3, 5, 6	货箱 3, 5, 6
	第 3 批	货箱 2, 3, 4, 6	货箱 2, 4

从表 3 可以看出, 本例中考虑商品订购数量和不考虑商品订购数量的模型得到的订单分批结果均对应 3 个批次, 每个批次包含 3 个订单, 但两种模型得到的货箱出库次数不同. 模型一得到的分批结果对应的货箱出库次数为 9 次, 模型二对应的为 6 次, 前者比后者的货箱出库次数多 3, 通过仔细分析发现, 由于模型二未考虑订单中订购的商品数量, 如果按照模型二的货箱出库策略进行拣选, 由于出库货箱中的商品储量无法满足订单中的商品数量需求, 因此实际拣选过程中必然需要调用更多的货箱出库, 才能满足订单拣选需求.

按照模型二的分批及货箱出库策略进行拣选时, 各个批次中出现缺货的商品信息以及需要增加的出库货箱信息如表 4 所示, 其中第 2 列表示每个批次订单中订购的各种商品总量, 第 3 列表示按照模型二的计算结果对应的出库货箱信息, 第 4 列表示按照模型二得到的出库货箱进行拣选时, 由于货箱中的商品数量小于订单中的商品订购量而无法满足的商品数量, 第 5 列表示为了满足各个批次订单中的需求, 至少需要增加的出库货箱信息. 第 5 列需要增加的出库货箱是按照 4.3 节的第 1 阶段第 2 步的启发式算法计算出来的. 容易验证, 本例中利用启发式算法得到的需要增加的货箱出库次数已经达到最小. 综合表 3 和表 4 的结果可以看出, 按照模型二的求解结果进行订单分批拣选时, 如果要满足订单中的商品数量需求, 实际需要出库的货箱次数为 11 次, 比直接利用模型一得到的结果多两次. 可见考虑商品订购数量的模型得到的订单分批拣选策略明显优于不考虑商品订购数量的模型.

表 4 模型二计算结果的进一步分析
(Table 4 Further analysis of results obtained by model 2)

批次	订购商品信息	模型二得到的 出库货箱	缺货商品信息	需要增加的 出库货箱
第 1 批	A(9), B(3), C(2)	货箱 3	A(6), B(1)	货箱 1, 2
第 2 批	A(3), B(2), D(4) E(1), F(2), G(1)	货箱 3, 5, 6	A(3), B(2)	货箱 1, 2
第 3 批	B(2), C(2), F(4), H(8)	货箱 2, 4	F(2)	货箱 6

示例 (续) 已知仓库中的货箱存储信息与拣选台容量与 3.2 节示例中相同. 假设仓库中新到达了 9 个订单, 各个订单的信息如下

$$O_1 = \{A(1), C(1)\}, O_2 = \{B(2)\}, O_3 = \{A(1), C(1)\}, O_4 = \{D(1), E(1)\},$$

$$O_5 = \{D(1), F(1)\}, O_6 = \{E(1)\}, O_7 = \{B(1), F(1), H(1)\},$$

$$O_8 = \{B(1), G(1), H(1)\}, O_9 = \{D(2)\}.$$

分别运用模型一与模型二对 9 个订单进行分批, 求解结果如表 5 所示.

表 5 两种模型的计算结果对比 (2)
(Table 5 Computational compare of two models (2))

		模型一 考虑商品订购数量	模型二 不考虑商品订购数量
订单分批 结果	第 1 批	O_1, O_2, O_3	O_1, O_2, O_3
	第 2 批	O_4, O_6, O_9	O_4, O_6, O_9
	第 3 批	O_5, O_7, O_8	O_5, O_7, O_8
货箱出库 结果	第 1 批	货箱 3	货箱 3
	第 2 批	货箱 5	货箱 5
	第 3 批	货箱 2, 5, 6	货箱 2, 5, 6

从表 5 可以看出, 两个模型的求解结果完全相同. 仔细分析可以发现, 本例中, 每个订单中包含的各种商品数量均很少, 且每个批次中包含的各种商品数量均不超过任何一个货箱中该商品的存储量, 此时拣选每种商品只需要搬运一个货箱或货架即可满足需求, 此时的问题可以简化为不考虑商品订购数量的情况, 即可以直接利用模型二求得最优解.

由以上分析可以看出, 本文的模型是对文献 [15] 中模型的一般化推广, 当每个货箱中存放的每种商品数量均可以满足所有订单拣选需求时, 问题可以简化为不考虑商品订购数量的情况, 直接按照文献 [15] 中的方法求解, 得到与本文模型相同的结果; 但是, 如果每个货箱中存放的同一种商品总量无法满足一个批次的订单拣选需求时, 按照文献 [15] 中的方法得到的分批结果进行拣选, 会出现商品缺货的情况, 此时, 需要按照本文模型求解.

近年来, 节日促销已经成为电商企业经常采取的手段 (如双 11, 双 12 等). 在促销期间, 商家会推出“多买多优惠”等营销策略, 客户为了获得优惠, 往往会多选购一些同类商品, 由此极易导致一个订单 (一个批次) 中的同种商品数量超过一个货箱的存储量, 这种情况下, 如果将问题简化为不考虑商品订购数量的情况, 就很难得到最优解或可行解. 相反, 按照考虑

商品订购数量的模型求解订单分批问题, 则可以有效避免缺货的情况. 因此本文研究的考虑商品订购数量订单分批模型与算法为解决“货到人”系统的订单分批问题提供了决策依据.

6 结 论

本文研究了考虑商品订购数量的自动小车存储及取货系统的订单分批拣选问题, 建立了以系统货箱出库次数极小化为目标的整数规划模型, 设计了快速求解模型的启发式算法. 利用不同规模算例进行仿真实验, 并与传统的先到先服务分批策略对比, 发现利用本文的模型和算法得到的订单分批结果, 拣选效率大约提升了 25%–45%. 进一步分析发现, 当拣选台容量增大时, 拣选订单需要出库的货箱次数呈现明显的递减趋势.

通过对比分析考虑商品订购数量和不考虑商品订购数量的订单分批模型之间的关系, 并利用两个具体算例进行计算, 证明了考虑商品订购数量的订单分批模型是不考虑商品订购数量的订单分批模型的一般化推广.

当所有订单中包含的同种商品数量不超过一个货箱中该商品的存储量时, 拣取每个批次订单中的同一种商品只需要出库一个货箱即可, 这种情况下可以不考虑订单中包含的商品数量, 直接利用文献 [15] 中的模型进行求解.

当同一批次订单中包含某种商品数量超过一个货箱中该商品的存储量时, 需要从多个货箱中拣取同一种商品, 此时问题无法简化为不考虑商品订购数量的情况. 需要使用本文的模型和算法进行求解. 在实际中, 由于每个货箱容量有限, 再加上电商促销等情况时有发生, 因此一个货箱中的商品存储量很难满足所有订单的需求, 因此本文的模型具有更广泛的适用性.

与文献 [15] 中不考虑商品订购数量的订单分批模型相比, 由于本文考虑了订单中的各种商品数量和货箱中的商品存储量, 因此本文的订单分批模型中决策变量和约束条件均比文献 [15] 中的模型多, 这使得本文模型比文献 [15] 中的模型更加复杂, 求解时间也更长. 实际中, 可以根据货箱中存储的商品数量和订单中包含的商品数量之间的关系判断能否将问题简化为不考虑商品订购数量的情况, 当问题无法简化时, 需要使用本文模型进行求解, 此时可以直接利用两阶段启发式算法在较短的时间内得到近似最优解.

由于本文研究的“货到人”拣选系统中只有一个拣选台, 并假设各个批次的订单单独拣选, 未考虑各个批次订单的拣选顺序以及拣选各个批次时货箱的出库顺序. 实际中, 当订单分批之后, 拣选不同批次订单可能需要调用同一个货箱, 如果不考虑拣选批次顺序和货箱出库顺序, 有可能出现同一个货箱多次出库的情况. 为了避免或减少同一个货箱多次出库的情况, 需要对分批以后的订单拣选顺序以及货箱出库顺序进行优化; 另外, 当系统中有多个拣选台的时候, 还需要协调各个拣选台的任务分配以及出库货箱的调度顺序, 避免货箱出库时设备忙碌需要等待以及货箱在拣选台前排队等待等问题. 后续研究中我们将考虑这些因素, 建立更加符合实际的模型.

参 考 文 献

- [1] 吴颖颖, 孟祥旭, 王艳艳, 等. “货到人”拣选系统订单排序优化. 机械工程学报, 2016, **52**(4): 206–212.
(Wu Y Y, Meng X X, Wang Y Y, et al. Order sequence optimization for “part-to-picker” order picking system. *Journal of Mechanical Engineering*, 2016, **52**(4): 206–212.)
- [2] 周丽, 朱杰, 郭健. 分类存储返回型与 S 型拣选路径随机模型的比较研究. 系统科学与数学, 2011, **31**(8): 921–931.
(Zhou L, Zhu J, Guo J. A comparative study of stochastic models for classified storage return type and S type picking path. *Journal of Systems Science and Mathematical Sciences*, 2011, **31**(8): 921–931.)
- [3] 李诗珍, 杜文宏. 基于聚类分析的订单分批拣货模型及启发式算法. 统计与决策, 2008, (12): 53–56.
(Li S Z, Du W H. Order batch picking model and heuristic algorithm based on cluster analysis. *Statistics & Decision*, 2008, (12): 53–56.)
- [4] 罗晓萌, 夏雪垠, 李建斌. 快消品电子商务仓储订单批次问题研究. 系统科学与数学, 2016, **36**(6): 847–859.
(Luo X M, Xia X Y, Li J B. Picking wave generation optimization strategies in B2C warehouse of fast moving consumer goods. *Journal of Systems Science and Mathematical Sciences*, 2016, **36**(6): 847–859.)
- [5] 王转, 裴泽平. 启发式路径下节约里程的订单分批算法. 计算机工程与应用, 2018, **54**(23): 203–209, 222.
(Wang Z, Pei Z P. Order batching algorithm based on heuristic picking route for saving mileage. *Computer Engineering and Applications*, 2018, **54**(23): 203–209, 222.)
- [6] 王旭坪, 张珩, 马骏. 考虑完成期限的电子商务在线订单分批模型及算法. 管理科学, 2014, **27**(6): 103–113.
(Wang X P, Zhang J, Ma J. E-commerce on-line order batching model and algorithm: Considering due time. *Journal of Management Science*, 2014, **27**(6): 103–113.)
- [7] Hwang H, Baek W J, Lee M K. Clustering algorithms for order picking in an automated storage and retrieval system. *International Journal of Production Research*, 1988, **26**(2): 189–201.
- [8] Menéndez B, Pardo E G, Alonso-Ayuso A, et al. Variable neighborhood search strategies for the order batching problem. *Computers & Operations Research*, 2017, **78**: 500–512.
- [9] Muter I, Öncan T. An exact solution approach for the order batching problem. *IIE Transactions*, 2015, **47**(7): 728–738.
- [10] Scholz A, Schubert D, Wäscher G. Order picking with multiple pickers and due dates — Simultaneous solution of order batching, batch assignment and sequencing, and picker routing problems. *European Journal of Operational Research*, 2017, **263**(2): 461–478.
- [11] Zhang J, Wang X, Chan F T S, et al. On-line order batching and sequencing problem with multiple pickers: A hybrid rule-based algorithm. *Applied Mathematical Modelling*, 2017, **45**: 271–284.
- [12] Füllner D, Boysen N. Efficient order processing in an inverse order picking system. *Computers & Operations Research*, 2017, **88**: 150–160.
- [13] Boysen N, Briskorn D, Emde S. Parts-to-picker based order processing in a rack-moving mobile robots environment. *European Journal of Operational Research*, 2017, **262**(2): 550–562.
- [14] Nicolas L, Yannick F, Ramzi H. Order batching in an automated warehouse with several vertical lift modules: Optimization and experiments with real data. *European Journal of Operational Research*, 2018, **267**(3): 958–976.
- [15] Xiang X, Liu C, Miao L. Storage assignment and order batching problem in Kiva mobile fulfilment system. *Engineering Optimization*, 2018, (4): 1–22.
- [16] 王艳艳, 周以齐, 沈长鹏, 等. 一种两类“货到人”订单拣选系统的适用性选择方法. 机械工程学报, 2015, **51**(4): 206–212.
(Wang Y Y, Zhou Y Q, Shen C P, et al. Applicability selection method of two parts-to-picker order picking systems. *Journal of Mechanical Engineering*, 2015, **51**(4): 206–212.)