

# “仓店一体化”模式下订单拣选与配送集成调度优化

陈伟, 张玉林

(东南大学 经济管理学院, 江苏 南京 211189)

**摘 要:** 针对“仓店一体化”模式下提供限时送达服务的新零售仓店, 研究多拣货员、多客户、配送方式为带限时送达约束的路径优化分批配送情形下的订单拣选与配送集成调度问题。以最小化订单最大履行时间和配送成本为目标, 构建了混合整数非线性规划模型, 并设计两阶段启发式算法(H-2)进行求解, 最后通过数值实验对算法进行验证与分析。结果表明, H-2 具有较高的求解质量; 相较于传统调度算法(TS)在效率提升、资源节约以及客户满意度提高方面具有更优的表现, 进而为新零售仓店管理者提供决策支持。

**关键词:** 集成调度; 订单拣选; 路径优化; 限时送达; 两阶段启发式算法

## Integrated Order Picking and Delivery Scheduling under “Warehouse and Store Integration” Mode

CHEN Wei, ZHANG Yu-lin

(School of Economics and Management, Southeast University, Nanjing, Jiangsu, 211189, China)

**Abstract:** The “warehouse and store integration” mode in new-retail context has received much attention which provides a commit-to-delivery in short time service. Consequently, immediate consumption demands are satisfied greatly and consumption experience is upgraded. However, due to the small lot-size, high frequency, time-sensitive, multi-variety customer orders, the new-retail enterprises face a big challenge that is how to fulfill order picking and delivery at a relatively low cost in the shortest promised time. To achieve the global optimal order fulfillment performance, this paper studies the integrated order picking and delivery scheduling problem with multi-pickers and due time. A mixed integer nonlinear programming model is established to minimize the maximum order fulfillment time and total cost of delivery. The model is solved by two-stage heuristic algorithm named H-2 proposed in this paper. To further verify the effectiveness and efficiency of algorithm H-2 in practice, we conduct numerical experiments to compare H-2 with traditional scheduling algorithm named TS. The experimental results indicate that the near optimal solution could be generated and algorithm H-2 performs better than TS in terms of efficiency improvement, resource conservation and customer satisfaction improvement, which provides decision-making references for new-retail enterprises managers.

**Key words:** integrated scheduling; order picking; vehicle route; commit-to-delivery; two-stage heuristic algorithm

---

收稿日期: 2019-08-29

基金项目: 国家自然科学基金(71671036, 71171046); 江苏省高校哲学社会科学研究重大项目(2018SJZDA005)

作者简介: 陈伟(1995—), 男, 福建福安人, 硕士研究生, 研究方向: 物流系统工程, E-mail: eliwchen@

foxmail.com; 通讯作者: 张玉林(1964—), 男, 江苏兴化人, 教授, 博士生导师, 研究方向: 收益管理、供应链管理  
等, E-mail: zhangyl@seu.edu.cn。

## 0 引言

近年来,物联网、大数据云计算以及人工智能等高科技的发展驱动消费习惯改变,“多品种、小批量、高频次、时间敏感”的订单碎片化特征愈加明显。单渠道购物已经无法完全满足消费者的各式需求(例如:B2C 电子商务线上渠道无法满足消费者的即时消费需求),“全渠道”消费成为一种不可阻挡的新趋势。“新零售”在此背景下应运而生,这是一种通过技术驱动提供精准极致体验的零售模式,该模式需要依托高效的物流体系打通线上与线下的渠道壁垒,因此具有高物流效率的“仓店一体化”模式备受关注。所谓的“仓店一体化”模式就是将仓库与实体门店相融合以服务全渠道客户的模式。在线下层面,实体店具备完全的门店销售功能;在线上层面,实体店可进行标准的仓储作业,从而实现人员和场地复用,提高了绩效。

以阿里巴巴旗下的“盒马”为例,其将物流配送中心与生鲜超市、饭馆、甚至是社交场所相融合,线上客户可免运费享受 30 分钟送达的配送服务,极大满足即时消费需求。2018 年“盒马”对外披露的运营业绩显示<sup>1</sup>,1.5 年以上门店,用户月购买次数达到 4.5 次,单店日均销售额超过 80 万元,单店坪效超过 5 万元,其中线上销售占比超过 60%,是传统超市的近四倍,展现了惊人的营收能力。同类业态的新零售平台还有京东“7Fresh”,永辉“超级物种”等。“盒马”等新零售平台的核心挑战在于如何在承诺的送达时间内,以较低的成本完成订单的拣选与配送。相较于以往研究的配送中心拣货,新零售仓店拣货存在诸多不同。首先,相较于自动化程度较高的配送中心,仓店配备多名拣货员进行人工拣选,订单批量小,品类复杂,采用订单分批策略。其次,仓店拣货会受到线下消费的顾客影响,高峰期拣货通道可能堵塞,造成拣货速度不稳定。此外,为进一步增大市场份额,新零售平台一般提供限时送达服务,仓店订单履行时间存在“分钟级”时效要求。因此,根据仓店拣货特点构建新调度模型有待进一步研究。

与本文相关的研究成果主要包含两类。一类是订单拣选优化,包括订单分批、指派与排序(order batching、assignment and sequencing, OBAS)。另一类是生产与配送集成调度(integrated production and delivery scheduling, IPDS)。订单拣选是指根据客户的订单信息或拣选作业计划,按一定的方式尽可能迅速、准确地将商品从其储位拣选出来以待配送的作业过程<sup>[1]</sup>。随着数据驱动思想的深入,基于订单数据分析的智能分批策略逐渐得到广泛应用。De Koster 等<sup>[2]</sup>设计节约法(C&W)进行订单分批,使订单总拣选时间最小。Azadnia 等<sup>[3]</sup>考虑订单的完成期限,利用带权重的关联规则算法计算订单间的关联度,构建订单关联度最大化的订单分批优化模型。以上的研究通常假设在订单拣选的过程中拣货员的数量为 1,不考虑后续的订单指派排序。实际上为了提高大规模订单的拣选效率,配送中心通常会雇佣多名拣货员进行拣货,因此多拣货员的订单分批、指派与排序(order

<sup>1</sup> 数据来源: <http://www.eeo.com.cn/2018/0917/337267.shtml>

batching、assignment and sequencing with multi-pickers, OBASMP) 也得到学者关注。Zhang 等<sup>[4]</sup>提出时窗分批结合通道相似度聚类的混合方法来完成在线订单分批, 并引入订单紧急度来解决紧急订单的分批问题, 最后基于紧迫度优先和拣货员工作量均衡原则完成订单指派排序。Henn 和 Sebastian<sup>[5]</sup>以最小化订单总延迟时间为目标, 建立了 OBASMP 模型, 并采用邻域搜索算法进行求解, 得到了订单分批、指派与排序调度方案。总的来看, 目前对订单同时到达且信息已知的离线订单分批研究比较成熟, 对于多拣货员的订单分批、指派与排序的集成研究相对较少。另外, 在新零售背景下, 为了更进一步与后端的配送环节衔接, 有必要在订单拣选环节中优化订单最大拣选完成时间。因此以最小化订单最大拣选完成时间为目标的 OBASMP 模型有待研究。

此外, 单独优化“订单拣选”难以有效降低订单履行时间, 订单拣选与配送集成调度 (integrated order picking and delivery scheduling, IOPDS) 一般能够优化目标值 5%~20%<sup>[6]</sup>, 获得众多学者关注。IOPDS 本质上是 IPDS 问题<sup>[7]</sup>。早期主要研究单机或平行机条件下的直接或批量生产调度问题。King 和 Love<sup>[8]</sup>针对单机/平行机、单辆配送车、多顾客等多种情形下的调度模型设计了启发式算法进行求解, 并证明了该问题是 NP-hard。马士华和吕飞<sup>[9]</sup>研究供应链环境下的生产与运输集成调度问题。李政道等<sup>[10]</sup>拓展了经典的订单指派与生产配送调度问题, 建立了多工厂布局以及带有送达时间约束的集成调度模型。以上的研究主要关注生产阶段的调度, 在配送方式上以简单的立即直接配送或成批配送为主, 近期研究进一步聚焦具有多种配送方式的集成调度问题。冯鑫等<sup>[11]</sup>讨论了订单可分批的单机生产调度, 配送方式涉及直接配送、分批配送以及带有时间窗的路径优化配送等。Saglam 和 Banerjee<sup>[12]</sup>考虑具有不同成本结构的运输方式的 IPDS 问题。Qiu 等<sup>[13]</sup>以及 Wang 等<sup>[14]</sup>学者研究配送路径更趋于复杂情形的集成调度问题。目前, IPDS 问题的相关研究成果已经非常丰富了<sup>[6]</sup>, 但应用在订单拣选中的文献却比较少, 且主要集中在 B2C 电子商务环境下配送中心的联合调度, 例如 Schubert 等<sup>[15]</sup>, Ramaekers 等<sup>[16]</sup>, Moons 等<sup>[17]</sup>, 这些学者普遍在模型中考虑了带时间窗的配送路径优化, 但通常将配送中心抽象化为一个拣货员或未考虑订单分批优化, OBASMP 问题仍存在一定探讨空间。另外, IOPDS 中对于新零售仓店的集成调度关注更少, 相似的研究有 Zhang 等<sup>[18]</sup>考虑了 O2O 社区商超环境下拣货员具有学习效应的在线订单拣选与配送联合调度问题。以上研究受限于订单拣选阶段的复杂性, 通常简化后端配送环节, 更贴近实践的调度模型仍有待研究。

综上所述, 本文聚焦新零售“仓店一体化”模式下考虑多拣货员、配送方式为带限时送达约束的路径优化分批配送的订单拣选与配送集成调度 (integrated scheduling of order picking and vehicle routing problem with multi-pickers and due time, ISOPVRPMD)。以最小化订单最大履行时间和配送成本为目标, 构建了混合整数非线性规划模型, 采用“路径优化-订单拣选优化”两阶段启发式算法进行求解, 最后通过数值实验对算法进行验证, 并与传统算法进行比较, 具有现实意义和参考价值。拟回答的具体问题是: (1) 订单如何分批; (2) 批次订单如何指派给拣货员; (3) 属于同一拣货员的批次订单如何安排拣货顺序; (4) 如何在承诺的送达时间内进行路径规划。

# 1 问题与模型

## 1.1 问题描述

本文研究“仓店一体化”背景下 IOPDS 问题，研究对象为某一时刻仓店三公里范围内  $N$  个社区生成的  $N$  张订单，每个社区生成一个社区订单。在接到订单之后，仓店需要分析订单信息结构，按照“订单分批、指派、排序”和“配送路径优化”算法生成完整的调度计划。仓店拣选区域按照储存的货品属性分为多个区域，只有右下角一个出入口，拣货员需要从出入口出发依次遍历所有分区，对每张批次订单依次完成 POS 机扫描—行走寻货—拣选—交由悬挂系统吊装与传输等操作，最后回到出入口。拣选完成的订单传入后仓进行分装打包以备配送，配送员按照规划配送路线在承诺的送达期限内完成配送。订单履行过程如图 1 所示，拣货区布局如图 2 所示。

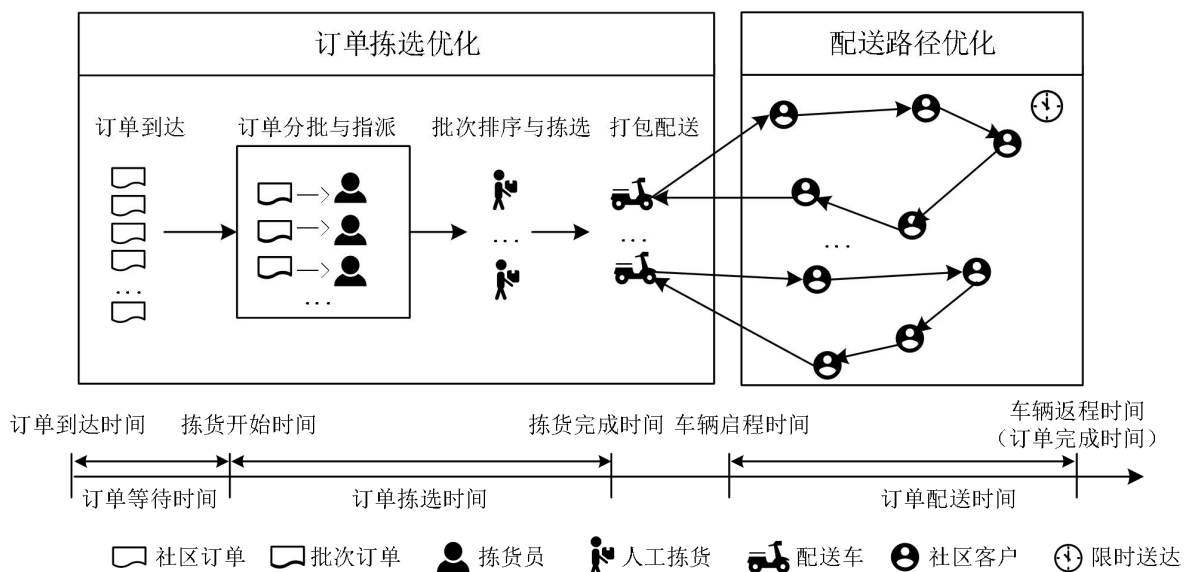


图 1 仓店订单履行过程

基于 Chen<sup>[19]</sup>提出五参数表示法  $\alpha|\beta|\pi|\delta|\gamma$ ，本文研究的集成调度问题描述参见表 1。

表 1 本文模型通用表达式解释

符号	解释
$\alpha$	$P_k$ ：拣货员数量为 $k$ ； $Manual$ ：人工拣选系统
$\beta$	$off-line$ ：离线订单； $batch$ ：订单分批； $S-shape$ ：拣货路径为 S 型
$\pi$	$V(\infty, Q)$ ：配送车辆数量无限制，最大容量为 $Q$ ； $CVRPD$ (capacitated vehicle routing problem with due time)：考虑配送时间和容量约束的路径优化问题
$\delta$	$n$ ： $n$ 个社区客户（客户以社区为单位）
$\gamma$	$\omega_1 T_{max} + \omega_2 TC$ ：目标函数； $T_{max}$ ：最大订单履行时间； $TC$ ：总配送成本； $\omega$ ：目标权重

$$P_k, Manual | off-line, batch, S-shape | V(\infty, Q), CVRPD | n | \omega_1 T_{max} + \omega_2 TC$$

## 1.2 模型构建

### 1.2.1 模型假设

(1) 不考虑到店消费的线下客户选货对拣货员按单拣选的影响，即拣货通道不存在堵塞；(2) 不考虑拣选准备时间、货物分类与打包时间；(3) 客户订单信息已知且客户订单必须归入某个批次；(4) 拣货任务不可中断（拣货员在完成当前批次拣选任务后立即执行下一个批次的拣选任务）；(5) 所有配送车辆类型相同；(6) 配送时间具有对称性；(7) 客户订单不允许分拆配送。

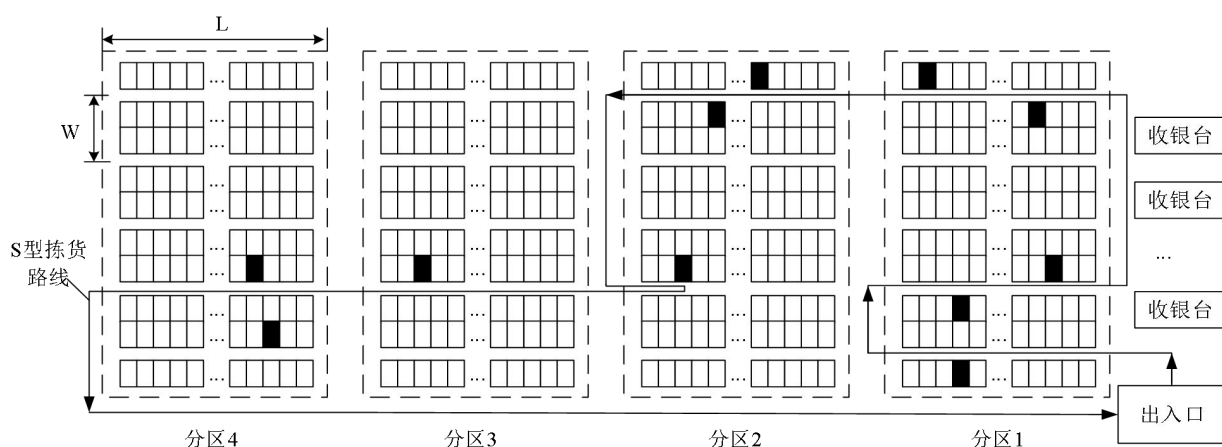


图 2 拣货区域布局图<sup>[18]</sup>

### 1.2.2 符号描述

本文所用的符号如表 2 和表 3 所示。

表 2 常量表

符号	解释	符号	解释
$N$	社区订单（配送点） $i$ 集合， $N = \{1, \dots, \overline{N}\}$	$q_i$	社区订单 $i$ 包含的子订单数
$N_0$	包含仓店的配送点 $i$ 集合， $N_0 = \{0\} \cup N$	$d$	线路总配送时间期限
$M$	批次订单 $j$ 集合， $M = \{1, \dots, \overline{M}\}$	$v^{travel}$	拣货员的行走速度
$P$	拣货员 $k$ 集合， $P = \{1, \dots, \overline{P}\}$	$v^{pick}$	拣货员的拣选速度
$V$	配送车辆 $v$ 集合， $V = \{1, \dots, \overline{V}\}$	$s$	车辆的行驶速度
$Q$	配送车所能容纳的最大订单数	$\theta$	单位配送时间成本（元/分钟）
$Q_i$	社区订单 $i$ 包含的需求量数	$\beta$	单位配送里程成本（元/公里）
$N_i$	社区订单 $i$ 包含的品项数	$\lambda$	单张子订单服务时间

表 3 变量表

变量符号	解释
$x_{ij}$	如果社区订单 $i$ 归并到批次订单 $j$ 则为 1；否则为 0
$y_{jk}$	如果批次订单 $j$ 指派给拣货员 $k$ 则为 1；否则为 0
$z_{iv}$	如果订单 $i$ 指派给配送车 $v$ 则为 1；否则为 0
$w_{igv}$	如果车辆 $v$ 从配送点 $i$ 驶向 $g$ 则为 1；否则为 0
$t_{jk}^{start}$	指派给拣货员 $k$ 的批次订单 $j$ 的开始拣货时间
$t_{jk}^{pick}$	指派给拣货员 $k$ 的批次订单 $j$ 的拣货时间
$t_{jk}^{complete}$	指派给拣货员 $k$ 的批次订单 $j$ 的拣货完成时间
$dis_j$	批次订单 $j$ 的拣选距离
$Dis_v$	配送车辆 $v$ 的配送里程
$\tau_{ig}$	配送点 $i$ 和 $g$ 之间的行驶时间, $\tau_{ii} = 0, \forall i \in N_0$
$dis_{ig}$	配送点 $i$ 和 $g$ 之间的行驶距离, $dis_{ii} = 0, \forall i \in N_0$
$t_v^{depart}$	配送车辆 $v$ 的启程时间（离开仓店的时间）
$t_v^{route}$	配送车辆 $v$ 所负责线路的总配送时间
$t_v^{delivery}$	配送车辆 $v$ 的配送时间（不包含从最后一个客户位置回到仓店的行驶时间）
$t_v^{return}$	配送车辆 $v$ 的返程时间即订单完成时间
$T_{max} (\overline{\overline{T_{max}}})$	最大订单履行时间（归一化数值）
$TC (\overline{\overline{TC}})$	配送总成本（归一化数值）

### 1.2.3 模型建立

基于以上分析，本文的 ISOPVRPMD 模型可构建如下：

$$\text{Min } \omega_1 \cdot \overline{\overline{T_{max}}} + \omega_2 \cdot \overline{\overline{TC}} \quad (1)$$

s.t.

$$\sum_{j \in M} x_{ij} = 1, \quad \forall i \in N \quad (2)$$

$$\sum_{k \in P} y_{jk} = 1, \quad \forall j \in M \quad (3)$$

$$t_{jk}^{start} \geq 0, \quad \forall j \in M, k \in P \quad (4)$$

$$t_{j+1,k}^{start} = \max_{j' \in \{1, \dots, j\}} \{t_{j'k}^{complete}\} - M_1(1 - y_{j+1,k}), \quad \forall k \in P \quad (5)$$

$$t_{jk}^{pick} = \frac{dis_j}{v^{travel}} + \sum_{i \in N} \frac{x_{ij} \cdot Q_i}{v^{pick}}, \quad \forall j \in M \quad (6)$$

$$t_{jk}^{complete} = \sum_{k \in P} y_{jk} (t_{jk}^{start} + t_{jk}^{pick}), \quad \forall j \in M \quad (7)$$

$$t_v^{depart} = \max \{x_{ij} z_{iv} t_{jk}^{complete}\}, \quad \forall v \in V \quad (8)$$

$$\sum_{v \in V} z_{iv} = \begin{cases} 1, & i \in N \\ V, & i = 0 \end{cases} \quad (9)$$

$$\sum_{g \in N} \sum_{v \in V} w_{igv} = \begin{cases} 1, & i \in N \\ V, & i = 0 \end{cases} \quad (10)$$

$$\sum_{g \in N_0} w_{igv} = z_{iv}, \quad \forall i \in N_0, v \in V \quad (11)$$

$$\sum_{g \in N} w_{0gv} = 1, \quad \forall v \in V \quad (12)$$

$$\sum_{i \in N} z_{iv} q_i \leq Q, \quad \forall v \in V \quad (13)$$

$$t_v^{delivery} = \sum_{g \in N} w_{0gv} \cdot \tau_{0g} + \sum_{i \in N} \sum_{g \in N} w_{igv} \cdot \tau_{ig} + \sum_{i \in N} \sum_{g \in N} w_{igv} q_i \lambda, \quad \forall v \in V \quad (14)$$

$$t_v^{return} = t_v^{depart} + t_v^{delivery}, \quad \forall v \in V \quad (15)$$

$$t_v^{route} = t_v^{delivery} + \sum_{i \in N} w_{i0v} \cdot \tau_{i0}, \quad \forall v \in V \quad (16)$$

$$t_v^{route} \leq d, \quad \forall v \in V \quad (17)$$

$$Dis_v = \sum_{g \in N} w_{0gv} \cdot dis_{0g} + \sum_{i \in N} \sum_{g \in N} w_{igv} \cdot dis_{ig} + \sum_{i \in N} w_{i0v} \cdot dis_{i0}, \quad \forall v \in V \quad (18)$$

$$T_{max} = \max \{t_v^{return}\}, \quad \forall v \in V \quad (19)$$

$$TC = \sum_{v \in V} (\theta \cdot t_v^{route} + \beta \cdot Dis_v) \quad (20)$$

$$x_{ij}, y_{jk}, z_{iv} \in \{0,1\}, \quad \forall i \in N; j \in M; k \in P; v \in V \quad (21)$$

$$w_{igv} \in \{0,1\}, \quad \forall i \in N_0; g \in N_0; v \in V \quad (22)$$

式（1）是“最小化最大订单履行时间”和“最小化总配送成本”两个子目标的线性加权函数。由于时间和成本是两个不同量纲的目标函数，不可直接相加，因此在加权前需要去量纲，即同度量处理。本文采用“极值法”对数值进行同度量处理，这是一种直线型无量纲化方法，主要思路是先对目标函数值确定一个标准值，然后计算各函数值  $X$  与标准值  $X_m$  之比，记为  $\bar{X}$ 。本文中的标准值设为目标函数值的上界  $X_{max}$ ，最大订单履行时间  $T_{max}$  的上界应为最后一组批次订单拣选完成时间加上最大配送时间。量纲归一化公式如（23）所示。

$$\overline{X} = \frac{X}{X_{\max}} \quad (23)$$

在约束条件方面，式（2）~（7）是拣货阶段约束。式（2）确保单张订单只能归并到一个批次；式（3）限制单批订单只能指派给一个拣货员；式（4）表示所有批次订单的开始拣货时间非负数；式（5）则表明拣货员的拣选批次之间没有空闲，下一批次的开始拣货时间要等于前一批次的拣选完成时间，其中  $M_1$  是个假定的极大值数；式（6）表示指派给拣货员  $k$  的批次订单  $j$  的拣货时间，该时间由订单拣选距离和拣货量决定；式（7）表示指派给拣货员  $k$  的批次订单  $j$  的拣选完成时间为其拣货开始时间与拣货时间之和。

式（8）~（18）是配送阶段约束。根据式（8）可计算出承运批次订单  $j$  的车辆  $v$  启程时间；式（9）~（12）对各级配送点的起运情况作了限制，其中式（9）规定每个配送点只能由一辆车访问一次（每张订单只能由一辆车承运），而式（10）表示各配送点对之间只能访问一次且所有车辆只能从 0 点出发最后回到 0 点，式（11）确保进入配送点  $i$  的车辆只能从该配送点出发，式（12）保证了所有车辆只能从 0 点出发一次；式（13）限制了单辆配送车的承运订单数不能超过上限  $Q$ ；式（14）是车辆  $v$  的配送时间表达式，由从 0 点出发到下一配送点的行驶时间、各配送点之间的行驶时间以及各配送点的服务时间组成；式（15）是车辆  $v$  的返程时间表达式，由车辆启程时间、配送时间组成，同时该时间也是车辆承运的批次订单的合计订单履行时间；式（16）是车辆  $v$  所负责线路的总配送时间，由车辆配送时间以及从最后一个配送点回到 0 点的行驶时间组成；式（17）严格规定所有线路的总配送时间需要在规定的期限内；式（18）表示车辆的行驶距离，由车辆从 0 点出发到下一配送点的行驶距离、各配送点之间的行驶距离、从最后一个配送点回到 0 点的行驶距离组成。式（19）表示所有批次订单中的最大订单履行时间。式（20）表示所有批次订单的总配送成本。式（21）和（22）表示决策变量取值范围。

## 2 算法设计

### 2.1 两阶段启发式算法

#### 2.1.1 算法描述

上述建立的 ISOPVRPMD 模型由 OBAS 和 CVRPD 组合而来，属于 NP-hard 问题，通常应用启发式算法如禁忌搜索法<sup>[20]</sup>、遗传算法<sup>[21]</sup>等进行分阶段求解，并被验证是可行有效的<sup>[10]</sup>。本文在王旭坪等<sup>[7]</sup>提出的启发式算法基础上设计了两阶段算法，记为 H-2。

集成调度目标是最小化订单的最大履行时间并尽可能优化后端配送环节以降低配送成本，因此核心思路是构建两阶段联系，降低提前完成拣选的订单等待时间，从整体上降低订单履行时间。新零售背景下订单具有高批次、小批量的特点，配送车容量与拣货员的最大拣货量可以近似相等，故可采用“逆推”思路，首先配送部门进行路径优化，确定配



送调度方案；仓店再将生成的某条线路上的订单合并为单个批次订单，然后进行批次订单的指派与排序，生成完整的调度计划。

在订单指派与排序环节，需要解决的核心问题是如何将多组批次订单交给多名拣货员进行处理，并安排订单拣选顺序使得订单拣选最大完成时间最短。这类问题可转化为经典的平行机排序问题  $P_m | n | T_{max}$ ，并采用近似算法进行求解，如：LPT 算法<sup>[22]</sup>、LS 算法<sup>[23]</sup>、Multifit 算法<sup>[24]</sup>等。本文结合订单紧迫度优先的原则，采用 LS 算法进行求解，保证了具有较高紧迫度的订单能尽早完成配送，从整体上降低订单履行时间。所谓的紧迫度指的是该订单需要进行立即处理的紧急程度，一般来讲如果该订单所需的配送时间越长，那么预留给订单拣选环节的时间就越少则紧迫度越高。LS 算法步骤如下。

---

#### LS 算法

---

算法开始：

- (1) 初始化。 $y_{jk} = 0, L_k = 0, k \in P, j \in M$ ， $L_k$  表示拣货员  $k$  当前工作所需要的时间；
- (2) 将前  $\bar{P}$  个批次订单按次序指派给  $\bar{P}$  个拣货员。令  $y_{kk} = 1, L_k = t_{jk}^{pick}, k \in P$ ；
- (3) 将剩余订单指派给最先空闲的拣货员；  
对任意属于序列  $\{\bar{P}+1, \dots, \bar{M}\}$  的订单  $j$ ，设  $L_t = \min\{L_k\} : x_{jt} = 1, L_t = L_t + t_{jk}^{pick}$
- (4) 输出  $t_{max}^{complete} = \max\{L_k\}$

算法结束。

---

根据 LS 算法求解问题  $P_m | n | T_{max}$ ，最优值下界为  $\text{Max}\left\{\left(\sum t_j^{pick}\right)/\bar{P}, \text{Max}\left\{t_j^{pick}\right\}\right\}$ 。LS 算法的时间复杂度为  $O(n)$ ，竞争比为  $2-1/\bar{P}$ <sup>[25]</sup>，在计算时间和结果上基于紧迫度优先规则和 LS 算法的订单指派排序方法接近最优解。

综上所述，算法简述如下。

#### 算法 1 订单拣选与配送集成调度的 H-2 算法

步骤 1：配送路径生成。调用 OR-Tools 中的 pywrapcp 和 routing\_enums\_pb2 算法包，用 Python 编程求解 CVRPD 模型，输入订单数据（订单位置坐标、订单量等），输出各配送线路订单的配送顺序  $\rho_v$ 、该线路的配送时间  $t_v^{delivery}$  和里程  $Dis_v$ 。

步骤 2：订单分批优化。基于生成的线路，将该线路上涉及的所有订单合并成一个批次  $j, j \in M$ 。

步骤 3：订单指派与排序。根据配送时间  $t_v^{delivery}$  进行非升序排列，得到序列  $\sigma$ ；再根据 LS 算法将序列  $\sigma$  内的批次订单  $j$  依次指派给拣货员，输出拣货员  $k$  的拣货顺序  $\sigma_k$ 。

步骤 4：返回获得的集成调度方案以及目标函数值，算法结束。

## 2.1.2 算法分析

### (1) 算法可行性分析

对于 IPDS 问题，Chen<sup>[26]</sup>给出了最优调度的性质：1) 在生产阶段，当机器负载大于 2 时不存在空闲；2) 在配送阶段，配送启程时间取决于配送批次中工件的最大加工完成时间，在该批次全部完工后立刻交付配送；3) 由同一台机器生产且在同一配送批次的所有订单在机器上需连续生产。本文研究的 ISOPVRPMD 属于 IPDS 中的特殊情形，模型约束和假设中已涵盖了上述性质要求，满足最优调度的条件。ISOPVRPMD 模型中的订单履行时间包含  $t_k^{start}$ 、 $t_k^{pick}$  以及  $t_v^{delivery}$ ，配送成本包含  $t_v^{route}$  和  $Dis_v$ 。其中， $t_k^{pick}$  取决于订单分批算法， $t_k^{start}$  取决于订单指派排序算法， $t_v^{delivery}$ 、 $t_v^{route}$  以及  $Dis_v$  由路径优化算法得到。综上所述，H-2 算法是可行的。

### (2) 算法性能分析

目标函数为  $\omega_1 \cdot T_{\max} + \omega_2 \cdot TC = \omega_1 \cdot \text{Max}\{t_v^{return}\} + \omega_2 \cdot (\sum (\theta \cdot t_v^{route} + \beta \cdot Dis_v))$ ，其上界为最后一组批次订单拣选完成时间加上最大配送时间与总配送成本加权之和，记算法上界为  $UB$ ，则  $UB = \omega_1 \cdot (\text{Max}\{t_j^{complete}\} + \text{Max}\{t_v^{delivery}\}) + \omega_2 \cdot (\sum (\theta \cdot t_v^{route} + \beta \cdot Dis_v))$ ；下界为最后一组批次订单拣选完成时间加上最小配送时间与总配送成本加权之和，记算法下界为  $LB$ ，则  $LB = \omega_1 \cdot (\text{Max}\{t_j^{complete}\} + \text{Min}\{t_v^{delivery}\}) + \omega_2 \cdot (\sum (\theta \cdot t_v^{route} + \beta \cdot Dis_v))$ ；记算法误差率为  $ER$ ，那么有  $ER = (UB - LB) / LB$ 。

## 2.2 传统算法

ISOPVRPMD 模型由“订单拣选”和“配送路径优化”两阶段组成，传统求解思路是单独处理两阶段子问题。仓店在接单后，先根据订单分批算法进行分批，然后根据 LPT 算法<sup>[22]</sup>进行订单指派与排序，等所有订单都拣选完毕再整批交付给配送部门；配送部门以最小化配送成本为目标，根据 VRP 优化算法生成各条线路，并交由配送员进行承运。传统调度 (traditional scheduling, TS) 算法简述如下。

### 算法 2 订单拣选与配送集成调度的 TS 算法

步骤 1：订单分批优化。不考虑后续配送的影响，以最小化订单拣选时间为目标，采用文献<sup>[2]</sup>中的节约法进行订单分批，生成拣选批次  $j$ 。

步骤 2：采用 LPT 算法进行订单指派与排序。先将拣选批次订单按照拣货时间  $t_j^{pick}$  进行非升序排列，依次指派给最先空闲的拣货员，得出拣货员  $k$  的拣货排序  $\sigma_k$ 。

步骤 3：配送路径优化。采用 2.1.1 所提方法进行路径规划，结合拣选批次的最迟完成时间，生成配送调度方案。

步骤 4：返回获得的订单拣选与配送调度方案以及目标函数值，算法结束。

### 3 数值实验

#### 3.1 参数设置

本文参照新零售仓店的运作环境设置参数。拣选区域采用多区型布局，不考虑品项之间的体积差异，每个品项单独占一个储位，按照商品编号依次存放。根据存储货品属性分为 4 个区域，可存储 1200 个品项。每个拣货分区一共有 5 个拣货通道，每个通道左右两边货架存储 60 个品项，通道长度  $L$  为 15 米，相邻两通道之间的距离  $W$  为 2 米，分区之间的距离忽略不计。一共有 3 名拣货员，拣货员的行走速度  $v^{travel}$  为 80 米/分钟，拣选速度  $v^{pick}$  为 30 个/分钟。配送车一次最多能容纳的订单数  $Q$  为 15，车辆的行驶速度  $s$  为 650 米/分钟，单位配送时间成本  $\theta$  为 0.8 元/分钟，单位配送里程成本  $\beta$  为 1 元/公里。线路总配送时间期限  $d$  为 25 分钟。目标函数权重系数  $\omega_1$  和  $\omega_2$  分别为 0.7 和 0.3。

假设仓店 3 公里范围内一共分布有 18 个社区，社区坐标  $(x,y)$  各自服从  $[0,10]$  上的均匀分布，每个单元格为  $300m \times 300m$ 。仓店坐标为  $(5,5)$ ，各坐标之间的距离采用曼哈顿距离  $d(i,j) = |x_1 - x_2| + |y_1 - y_2|$ 。社区订单中的子订单数  $q_i$  服从  $[1,8]$  上的均匀分布，每张子订单服务时间  $\lambda$  为 1 分钟。由于订单具有小批量等特征，令每张订单的需求量  $Q_i$  在  $[1,3]$  上服从均匀分布，品项数  $N_i$  服从  $[1,5]$  均匀分布。随机产生的订单数据如表 4 所示。CVRPD 求解程序用 Python 语言编写，开发平台为 Pycharm，规划求解器为 Solver。计算机硬件配置中 CPU 为 Intel(R)Core(TM)i5，内存 8GB，操作系统为 Win10。

表 4 社区客户订单数据

订单编号	1	2	3	4	5	6	7	8	9
坐标	(2,4)	(7,7)	(2,10)	(8,2)	(2,0)	(5,1)	(1,6)	(0,10)	(10,6)
子订单数	6	4	8	2	5	8	4	3	7
品项数	21	9	21	3	11	21	10	4	22
需求量	35	18	34	4	18	39	17	5	40
订单编号	10	11	12	13	14	15	16	17	18
坐标	(6,3)	(4,9)	(7,1)	(4,7)	(10,1)	(1,1)	(9,10)	(4,4)	(6,9)
子订单数	6	2	8	8	3	8	5	5	7
品项数	17	4	26	16	8	16	11	16	11
需求量	30	7	42	27	13	29	17	27	18

3.2 结果分析

3.2.1 H-2 算法结果

在路径优化环节，CVRPD 结果如表 5 和图 3 所示。一共生成了 8 条配送线路，每条线路包含 2~3 张社区订单，各批次的配送时间在 11~17 分钟之间，总配送时间为 119 分钟，合计线路总配送时长为 139 分钟，总配送成本为 147.2 元。

表 5 CVRPD 结果表

配送批次	配送路线	订单量	配送时间	线路配送时间	线路配送里程	配送成本
1	0 -> 3-> 8-> 0	11	14	18	6000	20.4
2	0-> 12 -> 14 -> 4-> 0	13	17	19	5400	20.6
3	0 -> 9-> 16 -> 0	12	16	20	6000	22
4	0 -> 2 -> 18-> 0	11	13	15	3600	15.6
5	0 -> 15->5-> 0	13	16	19	5400	20.6
6	0 -> 10 -> 6 -> 0	14	16	17	3000	16.6
7	0 -> 13-> 11 -> 0	10	11	13	3000	13.4
8	0 -> 17 -> 1 -> 7 -> 0	15	16	18	3600	18

表 6 和图 4 给出了 H-2 算法结果。拣货员 1 的拣货顺序 $\sigma_1 = [2,1,4]$ ，拣货员 2 的拣货顺序 $\sigma_2 = [3,8]$ ，拣货员 3 的拣货顺序 $\sigma_3 = [5,6,7]$ 。所有批次订单的平均拣货时间为 6.41 分钟，平均等待时间为 5.66 分钟，拣货完成时间（车辆启程时间）在 5~18 分钟之间，平均拣货完成时间（车辆平均等待时间）为 12.07 分钟，最大拣货完成时间为 18.09 分钟。订单履行时间在 21~32 分钟之间，最大履行时间为 32.96 分钟，平均履行时间为 26.94 分钟，总履行时间为 215.53 分钟，按时履单率为 88.89%。

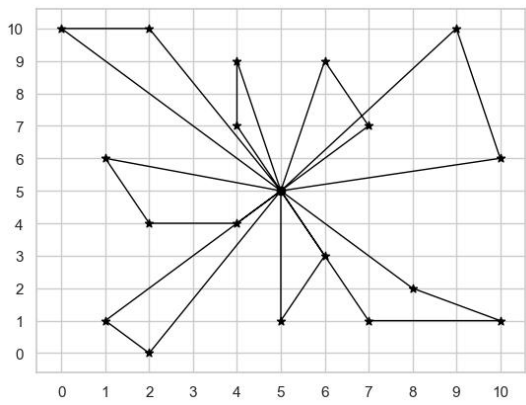


图 3 配送路径优化图

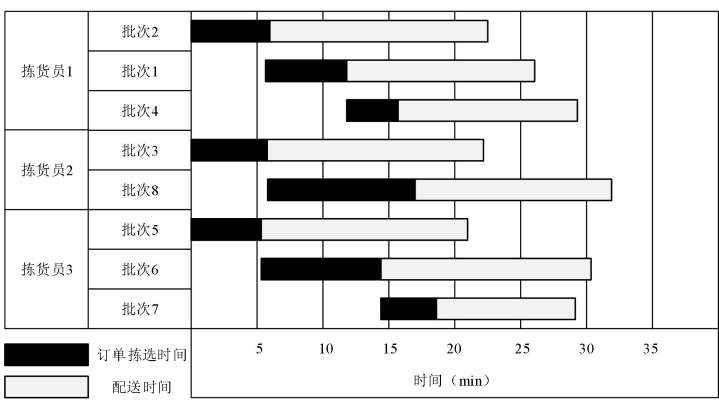


图 4 H-2 订单拣选与配送集成调度甘特图

表 6 H-2 订单拣选与配送集成调度表

拣货员	订单批次	配送时间	拣货时间	拣货开始时间	拣货完成时间	订单履行时间	订单完成数
1	2	17	6.87	0.00	6.87	23.87	3
	1	14	5.39	6.87	12.25	26.25	2
	4	13	3.98	12.25	16.23	29.23	2
2	3	16	6.23	0.00	6.23	22.23	2
	8	16	10.73	6.23	16.96	32.96	2
	5	16	5.45	0.00	5.45	21.45	2
3	6	16	9.00	5.45	14.45	30.45	1
	7	11	3.63	14.45	18.09	29.09	2
合计		119	51.28	45.25	96.53	215.53	16
最大值		17	10.73	14.45	18.09	32.96	
平均值		14.875	6.41	5.66	12.07	26.94	
按时履单率							88.89%

为检测 H-2 的性能，在订单拣选阶段以最小化订单最大拣选完成时间为目标，选取了 5~8 个批次订单进行测试，通过规划求解器求得最优解，与 H-2 算法结果进行对比，结果如表 7 所示。

表 7 H-2 性能分析

批次订单数	Solver	H-2	Gap (%)
4	9.36	9.36	0.00
5	10.84	10.84	0.00
6	12.98	12.98	0.00
7	14.39	14.48	0.62
8	17.60	18.08	2.65

对于特小规模问题（批次订单数<7），H-2 可得到最优解，当问题规模增大时，H-2 的差距率基本在 3%以内，根据算例的不同差距率存在一定波动，因此 H-2 基本可以生成接近最优的解。从整体上看，当算例的批次订单数为 8 时，由 2.1.2 节的分析可得 H-2 的上界订单履行时间为 35.09，下界订单履行时间为 29.09，模型最大订单履行时间为 32.96。根据公式（23）进行归一化，则  $X_{max}$  为 35.09，上述的各数值去量纲后分别为 1、0.83 和 0.94，上下界的误差率为 20.6%，按 0.7 和 0.3 权重对目标函数值进行加权，则模型解与下界差距率为 8.7%。H-2 所得结果边界较紧，接近最优解。

3.2.2 TS 算法结果

TS 算法结果如表 8~9 和图 5~6 所示。一共生 6 个拣选批次，拣货员 1 的拣货顺序  $\sigma_1 = [1,6]$ ，拣货员 2 的拣货顺序  $\sigma_2 = [3,4]$ ，拣货员 3 的拣货顺序  $\sigma_3 = [2,5]$ 。在订单拣选阶段，所有拣选批次的平均拣货时间为 9.37 分钟，平均等待时间为 5.96 分钟，拣货完成时间在 10~19 分钟之间，平均拣货完成时间为 15.32 分钟，最大拣货完成时间为 19.27 分钟。某一配送路线的车辆启程时间取决于该路线上所涉订单的最晚拣货完成时间，例如：配送批次 1 包含社区订单 3 和 8，分别散落在拣选批次 4 和 1 内，因此只有等拣选批次 4 和 1 都拣选完毕才能开始配送。

表 8 订单拣选调度表

拣货员	拣选批次	包含的社区订单	拣货时间	拣货开始时间	拣货完成时间
1	1	10、15、11、8、4	13.65	0.00	13.65
	6	5、7	4.49	13.65	18.14
2	3	1、13、18	11.85	0.00	11.85
	4	6、3	6.93	11.85	18.79
3	2	12、2、16	10.24	0.00	10.24
	5	9、17、14	9.03	10.24	19.27
合计			56.20	35.75	91.95
最大值			13.65	13.65	19.27
平均值			9.37	5.96	15.32

车辆启程时间在 11~19 分钟之间，车辆平均等待时间为 16.88 分钟，车辆的启程顺序为 4—（1—7）—6—（2—3—5—8），批次 4 最早开始配送，1 和 7 同时开始配送，然后是 2、3、5、8 同时开始配送。所有订单的总履行时间为 254.03 分钟，最大履行时间为 36.27 分钟，平均履行时间为 31.07 分钟，按时履单率为 61.11%。

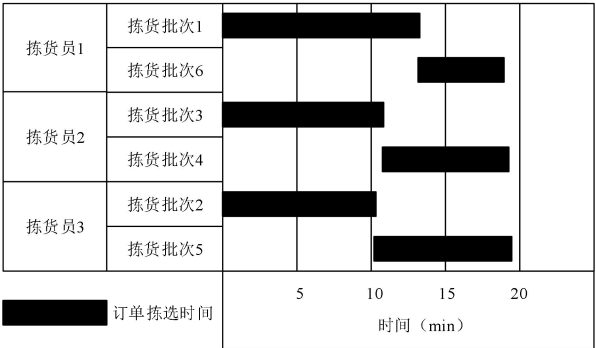


图 5 拣选调度甘特图

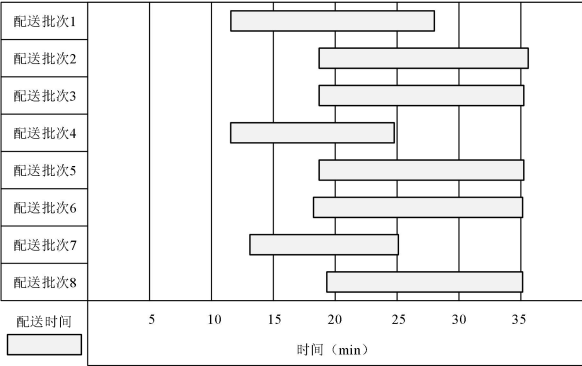


图 6 配送调度甘特图

表 9 配送调度表

配送批次	配送路线	最后拣选批次	车辆启程时间	配送时间	订单履行时间	订单完成数
1	0 -> 3-> 8-> 0	1	13.65	14	27.65	2
2	0-> 12 -> 14 -> 4-> 0	5	19.27	17	36.27	1
3	0 -> 9-> 16 -> 0	5	19.27	16	35.27	1
4	0 -> 2 -> 18-> 0	3	11.85	13	24.85	2
5	0 -> 15->5-> 0	5	19.27	16	35.27	1
6	0 -> 10 -> 6 -> 0	4	18.79	16	34.79	1
7	0 -> 13-> 11 -> 0	1	13.65	11	24.65	2
8	0 -> 17 -> 1 -> 7 -> 0	5	19.27	16	35.27	1
合计			135.03	119	254.03	
最大值			19.27	17	36.27	
平均值			16.88	14.875	31.75	
按时履单率						61.11%

3.3 结果对比分析

本小节将从拣货时间、订单等待时间、拣货完成时间、车辆等待时间、订单履行时间、车辆启程顺序、按时履单率以及目标函数值，这八个指标对 H-2 和 TS 求解结果进行对比分析，结果如表 10 所示。

表 10 订单拣选与配送集成调度结果对比

对比指标	H-2				TS			
	最大值	最小值	平均值	合计	最大值	最小值	平均值	合计
拣货时间	<u>10.73</u>	<u>3.63</u>	<u>6.41</u>	<u>51.28</u>	13.65	4.49	9.37	56.20
订单等待时间	14.45	0.00	<u>5.66</u>	45.25	<u>13.65</u>	<u>0.00</u>	5.96	<u>35.75</u>
拣货完成时间	<u>18.09</u>	<u>5.45</u>	<u>12.07</u>	96.53	19.27	10.24	15.32	<u>91.95</u>
车辆等待时间	<u>18.09</u>	<u>5.45</u>	<u>12.07</u>	<u>96.53</u>	19.27	11.85	16.88	135.03
订单履行时间	<u>32.96</u>	<u>21.45</u>	<u>26.94</u>	<u>215.53</u>	36.27	24.65	31.75	254.03
车辆启程顺序	<u>5—3—2—1—6—4—8—7</u>				4—（1—7）—6—（2—3—5—8）			
按时履单率	88.89%				61.1%			
目标函数值	<u>0.93</u>				1			

备注：数值 表示更优

从表 10 的数值对比中可得到如下结果：

（1）从整体来看，H-2 在绝大部分的评价指标上优于 TS，只有“订单等待时间”以及“合计拣货完成时间”方面不占优。相较于 H-2，TS 的合计订单等待时间下降了 21%。原因在于 TS 可以生成更大批量的拣选订单，拣选批次数更少，拣货员访问出入口的次数会随之降低，拣货员的有效工作量得以提高。因此，如果调度目标侧重改进订单等待时间，提高拣货员的有效工作量，那么对订单拣选环节进行单独优化是更好的选择。

（2）在订单拣选阶段，H-2 在“拣货时间”和“拣货完成时间”方面基本优于 TS，拣货时间的平均值以及合计值的改进程度分别为 31.6%和 8.9%，这说明基于 H-2 进行订单指派与排序更加高效。

（3）在配送阶段，H-2 在“车辆等待时间”这一指标上全面优于 TS，最大值改进程度为 6.1%而平均值和合计值的改进程度均为 28.5%，这说明 H-2 对于车辆资源的利用率更高。原因在于 H-2 将订单拣选与配送相联系，在时点衔接上更密切，相较于 TS 具有更大的灵活性和易操作性。

（4）在“车辆启程顺序”方面，H-2 下的配送车依次取货离开仓店，而 TS 下的配送车存在一定的堵塞可能性，例如配送批次（1—7）以及（2—3—5—8）需要在各自相同的启程时间内同时离开仓店，这意味着配送员需要在相同的时间期限内完成取货、装车等操作，可能会造成后仓出库区的繁忙和堵塞。

（5）在“订单履行时间”方面，H-2 基本优于 TS。H-2 的订单平均履行时间为 26.94，在承诺的 30 分钟送达时间内，而 TS 的订单平均履行时间为 31.75，超出了承诺的送达时间，其改进程度为 15.1%。另外，最大值的改进程度为 9.1%，按时履单率也从 61.11%提升至 88.89%，目标函数值的优化程度为 7%。从 TS 的执行流程中可看出，配送部门的规划与订单拣货过程完全隔绝，造成拣选批次与配送批次之间不一致。同一配送批次之间的订单可能分散在不同的拣选批次中，只有当配送批次中的所有订单都拣选完毕才能执行配送，这将导致提前完成的订单需要等待一段时间，从而增大了订单履行时间。而 H-2 能够统筹两阶段特点，协调两阶段的不匹配情况，有效改进了物流系统运作效率，大幅度提升客户满意度。

### 3.4 目标权重敏感性分析

为了进一步验证 H-2 的有效性与优越性，对目标权重进行敏感性分析。探讨不同权重组合下函数值的变化，比较 H-2 与 TS 算法的优劣，求解得出的目标函数值如表 11 和图 7 所示。



表 11 目标权重敏感性分析

目标权重( $\overline{T_{max}}$ , $\overline{TC}$ )	H-2	TS	Gap (%)
(0, 1)	1	1	0
(0.1, 0.9)	0.994	1.004	0.96
(0.2, 0.8)	0.988	1.007	1.93
(0.3, 0.7)	0.982	1.011	2.91
(0.4, 0.6)	0.976	1.014	3.9
(0.5, 0.5)	0.97	1.018	4.9
(0.6, 0.4)	0.964	1.021	5.92
(0.7, 0.3)	0.958	1.024	6.95
(0.8, 0.2)	0.952	1.028	7.99
(0.9, 0.1)	0.946	1.031	9.05
(1, 0)	0.94	1.035	10.12

从表 11 和图 7 可以看出，随着  $\overline{T_{max}}$  权重的不断增大和  $\overline{TC}$  权重的减小，TS 的目标函数值逐渐缓慢增大，而 TS 的目标函数值呈现较为明显的下降，二者之间的差距率则逐渐拉大且 H-2 目标函数值总优于 TS。当  $\overline{T_{max}}$  权重为 0， $\overline{TC}$  权重为 1 即完全转换为以  $\overline{TC}$  为目标的单目标问题，H-2 与 TS 的差距率最小；当  $\overline{T_{max}}$  权重为 1， $\overline{TC}$  权重为 0 即完全转换为以  $\overline{T_{max}}$  为目标的单目标问题，H-2 与 TS 的差距率最大。

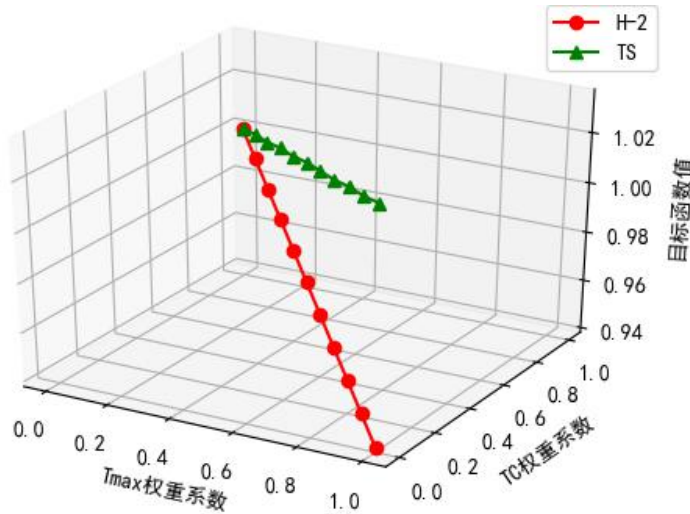


图 7 目标权重敏感性分析

由此可见，TS 算法对于目标权重并不敏感，而 H-2 对于  $\overline{T_{max}}$  的权重较为敏感，在  $\overline{T_{max}}$  权重较大时能够取得更优效果，这是因为 H-2 算法通过建立拣货批次与配送批次的一致性，降低了订单等待配送的时间，从而极大优化了最大订单履行时间。

## 4 结论

“仓店一体化”模式可以有效应对新零售背景下订单“多品种、小批量、高频次、时间敏感”等碎片化特征所带来的物流时效性要求。本文研究“仓店一体化”模式下新零售仓店多拣货员、多客户、配送方式为带限时送达约束的路径优化的成批配送的集成调度问题,并设计两阶段启发式算法进行求解,最后通过数值实验对算法的有效性进行验证分析。实验结果表明,H-2 具有较高的求解质量且对目标“最大订单履行时间”权重较为敏感,能够较大程度优化最大订单履行时间;相较于 TS 在效率提升、资源节约以及客户满意度提高方面具有更优的表现,能够大幅度降低订单履行时间,改善配送车的利用效率以及有效避免后仓出库区的堵塞情况,提高按时履单率;为新零售仓店管理者提供了建模与求解的新思路。但如果管理者更侧重降低从接单到开始拣选这一阶段的订单等待时间,提升拣货员的有效工作量,那么集成调度并不是一个更优的选择,此时需要单独对其进行优化。

本文未考虑仓店拣货可能存在拣货通道堵塞的情况以及未采用更有实际意义的分区拣选策略,后续研究可在此基础上进行拓展。此外,仓店订单通常是随着时间动态到达,在线订单集成调度问题有待进一步研究。

## 参考文献

- [1] De Koster R, Le-Duc T, Roodbergen K J. Design and control of warehouse order picking: A literature review[J]. *European Journal of Operational Research*, 2007, 182(2): 481-501.
- [2] De Koster M B M, Van der Poort E S, Wolters M. Efficient orderbatching methods in warehouses[J]. *International Journal of Production Research*, 1999, 37(7): 1479-1504.
- [3] Azadnia A H, Taheri S, Ghadimi P, Mat Saman M Z, Wong K Y. Order batching in warehouses by minimizing total tardiness: a hybrid approach of weighted association rule mining and genetic algorithms[J]. *The Scientific World Journal*, 2013, 2013:1-13.
- [4] Zhang J, Wang X, Huang K. Integrated on-line scheduling of order batching and delivery under B2C e-commerce[J]. *Computers & Industrial Engineering*, 2016, 94:280-289.
- [5] Henn S. Order batching and sequencing for the minimization of the total tardiness in picker-to-part warehouses[J]. *Flexible Services & Manufacturing Journal*, 2015, 27(1): 86-114.
- [6] Moons S, Ramaekers K, Caris A, Arda Y. Integrating production scheduling and vehicle routing decisions at the operational decision level: A review and discussion[J]. *Computers & Industrial Engineering*, 2017, 104:224-245.
- [7] 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.
- [8] King R H, Love R R. Coordinating decisions for increased profits[J]. *INFORMS Journal on Applied Analytics*, 1980, 10(6): 4-19.
- [9] 马士华, 吕飞. 基于 Supply-Hub 的生产与配送协同模式研究[J]. *中国管理科学*, 2014, 22(6): 50-60.
- [10] 李政道, 周泓. 基于限时送达服务的订单分派与生产运送调度的集成优化[J]. *计算机集成制造系统*,

- 2014, 20(7): 1643-1653.
- [11] 冯鑫, 郑斐峰, 徐寅峰. 批加工生产配送二级供应链协同调度[J]. 系统管理学报, 2015, 24(2): 275-279.
  - [12] Sağlam Ü, Banerjee A. Integrated multiproduct batch production and truck shipment scheduling under different shipping policies[J]. *Omega*, 2018, 74:70-81.
  - [13] Qiu Y, Qiao J, Pardalos P M. Optimal production, replenishment, delivery, routing and inventory management policies for products with perishable inventory[J]. *Omega*, 2019, 82:193-204.
  - [14] Wang J, Yao S, Sheng J, Yang H. Minimizing total carbon emissions in an integrated machine scheduling and vehicle routing problem[J]. *Journal of Cleaner Production*, 2019, 229:1004-1017.
  - [15] Schubert D, Scholz A, Wäscher G. Integrated order picking and vehicle routing with due dates[J]. *OR Spectrum*, 2018, 40(4): 1109-1139.
  - [16] Ramaekers K, Caris A, Moons S, van Gils T. Using an integrated order picking-vehicle routing problem to study the impact of delivery time windows in e-commerce[J]. *European Transport Research Review*, 2018, 10(2):56.
  - [17] Moons S, Braekers K, Ramaekers K, Caris A, Arda Y. The value of integrating order picking and vehicle routing decisions in a B2C e-commerce environment[J]. *International Journal of Production Research*, 2019, 57(20): 6405-6423.
  - [18] Zhang J, Liu F, Tang J, Li Y. The online integrated order picking and delivery considering Pickers' learning effects for an O2O community supermarket[J]. *Transportation Research Part E: Logistics and Transportation Review*, 2019, 123:180-199.
  - [19] Chen Z. Integrated production and outbound distribution scheduling: review and extensions[J]. *Operations Research*, 2010, 58(1): 130-148.
  - [20] Armentano V A, Shiguemoto A L, Løkketangen A. Tabu search with path relinking for an integrated production – distribution problem[J]. *Computers & Operations Research*, 2011, 38(8): 1199-1209.
  - [21] Low C, Chang C, Li R, Huang C. Coordination of production scheduling and delivery problems with heterogeneous fleet[J]. *International Journal of Production Economics*, 2014, 153:139-148.
  - [22] Woeginger K G. The exact LPT-bound for maximizing the minimum completion time [J]. *Operations Research Letters*, 1992, 11(5): 281-287.
  - [23] 唐峰, 聂劲. 平行机上订单半在线排序的 LS 算法的性能比分析[J]. *系统工程*, 2016, 6: 72-77.
  - [24] Yue M. On the exact upper bound for the multifit processor scheduling algorithm[J]. *Annals of Operations Research*, 1990, 24(1): 233-259.
  - [25] Graham R L. Bounds for certain multiprocessing anomalies[J]. *Bell System Technical Journal*, 1974, 45(9): 1563-1581.
  - [26] Chen Z, Vairaktarakis G L. Integrated scheduling of production and distribution operations[J]. *Management Science*, 2005, 51(4): 614-628.