

doi: 10.3969/j.issn.1007-7375.240086

# 考虑工人疲劳的电商订单分批拣货与排序优化

刘进平, 臧美娜, 张燕, 郑建凤

(大连海事大学 交通运输工程学院, 辽宁 大连 116026)

**摘要:** 随着各大电商购物平台在订单履行时效性方面的竞争日益加剧, 如何提高拣货效率已经成为电商仓库亟待解决的关键问题。以往研究认为分批有利于提高拣货效率, 但没有考虑工人工作效率会因疲劳而下降的因素, 导致订单分批与排序优化结果难以付诸实践。本文考虑人员的能量消耗, 将工人的休息批次和订单分批与排序决策联合优化, 以最小化订单总完成时间为目标, 建立混合整数规划模型。设计两阶段法和基于控制思想的种子算法, 并对算法有效性进行对比和验证。研究表明, 基于控制思想的种子算法要优于两阶段法。当考虑工人疲劳因素安排工人进行休息时, 既能增加员工满意度, 又可以减少订单总完成时间。敏感度分析揭示了当拣选员与合并员的工作强度差距增大或缓冲区容量不确定时, 基于控制思想的种子算法优化效果更好且具有稳定性。

**关键词:** 仓库拣货; 订单分批; 排序与调度; 疲劳; 种子算法

中图分类号: C934; F406

文献标志码: A

文章编号: 1007-7375(2025)01-0061-12

## Optimization of E-commerce Order Batching and Sequencing Considering Worker Fatigue

LIU Jinping, ZANG Meina, ZHANG Yan, ZHENG Jianfeng

(Transportation Engineering College, Dalian Maritime University, Dalian 116026, China)

**Abstract:** As competition in terms of order fulfillment timeliness among major e-commerce shopping platforms intensifies, improving order picking efficiency has become a key problem for e-commerce warehouses. Previous studies suggest that order batching strategies can enhance picking efficiency but overlook the fact that worker efficiency tends to decline with fatigue, leading to challenges in implementing the optimization results of batching and sequencing. A mixed integer programming model is developed to minimize the total order completion time. This model considers worker energy consumption while jointly optimizing worker rest batching, order batching and sequencing decisions. A two-stage algorithm and a control-based seed algorithm are developed, while their performance is compared and verified. Results show that the control-based seed algorithm is superior to the two-stage algorithm; the rest arrangement with consideration of worker fatigue can not only enhance employee satisfaction but also decrease the total order completion time. Sensitivity analysis reveals that when the work intensity difference between pickers and sorters increases or the buffer capacity becomes uncertain, the control-based seed algorithm exhibits superior effectiveness and stability.

**Key words:** warehouse picking; order batching; sequencing and scheduling; fatigue; seed algorithm

中国电商订单履行模式不断革新升级, 为提高订单履行的时效性, 很多电商企业采用自动化拣货技术。然而考虑到商品规格多样性和复杂性, 传统的人工拣货方式在电商领域仍然具有广泛的应用<sup>[1]</sup>, 例如京东“211”限时达、天猫的次日达等。鉴于

B2C 电子商务订单数量庞大且单个订单规模较小的特点, 电商仓库经常采用一种先拣选再合并的订单分批策略。这种“拣选-合并”的分批处理方式, 能够有效应对大规模订单流, 确保订单拣货过程快速且准确, 从而提升整个订单履行流程的效率。在

收稿日期: 2024-03-12

基金项目: 国家自然科学基金资助项目(72371046)

作者简介: 刘进平(1976—), 辽宁省人, 讲师, 博士, 主要研究方向为仓储物流优化。Email: liujp@dlnu.edu.cn

拣选-合并的人工拣货作业中, 货品拣选和订单合并作为两道紧密相连的工序, 若调度不当, 易出现工序间的相互等待现象。同时, 由于作业过程中需要工人反复进行举重、搬运和放置等动作, 工人容易感到疲劳。这种疲劳状态不仅会影响工人的身体健康, 还会直接降低作业效率, 导致订单的总完工时间延长。因此, 如何在充分考虑工人疲劳度增加的前提下, 合理决策订单的分批与排序, 同时妥善安排工人的休息时间, 成为电商仓库管理者亟需解决的难题之一。

订单分批问题的优化模型源于 Gademann 等<sup>[2]</sup>的研究, 该研究还通过复杂度分析证明了订单分批问题属于 NP-hard 问题。有关订单分批问题的研究已经十分丰富, 可以参考较新的综述文章<sup>[3]</sup>。订单分批与排序问题是订单分批问题的一个变体, 很多学者从不同角度针对人工拣货系统的订单分批与排序问题进行研究。Menendez 等<sup>[4]</sup>基于人工拣货系统以最小化订单延迟时间为目标构建订单分批和排序模型, 并采用变邻域搜索算法求解, 但没有考虑订单合并阶段。近年来有多篇文献关注基于分拣墙的人工拣货系统。Boysen 等<sup>[5]</sup>针对类似的拣货系统联合考虑分区拣货, 构建拣货箱顺序优化模型, 旨在降低合并员的空闲时间, 但该研究没有考虑订单批次的优化。Ardjmand 等<sup>[6]</sup>基于分拣墙的拣选-合并系统, 采用超启发式算法解决订单分批与路径的联合优化难题。然而, 该研究并未充分考虑到批排序对拣选和合并两个阶段平衡性的潜在影响。Jiang 等<sup>[7]</sup>针对分拣墙场景下的拣选-合并系统进行深入研究, 探讨缓冲区有容量限制的订单分批与排序问题, 构建以最小化总完工时间为目标的混合整数规划模型, 创新性地考虑中间有缓冲区的订单分批与排序问题并设计了基于控制思想的种子算法, 通过模型的两个性质分析和下界验证了算法的有效性。该研究充分体现了种子算法在协调两阶段分批问题上的巧妙和高效。

上述研究在探讨订单分批与排序问题时, 均基于一个共同的假设, 即员工的工作效率恒定不变。然而, 由于仓库工人需要执行大量手动搬运活动, 因此员工疲劳在人工拣货系统中很常见, 这会导致工人的工作效率降低<sup>[8]</sup>。在实际工作环境中, 员工的疲劳程度往往随着工作时间的延长而增加, 将会严重影响拣货效率。目前已经有学者<sup>[9-10]</sup>关注工人疲

劳问题。疲劳可以从生理和身体两方面来定义, 一般来讲心理性疲劳比较主观<sup>[11]</sup>, 本文的疲劳是指身体上的疲劳。

近年来仓储领域关于工人疲劳的研究也在增加。赵小松等<sup>[12]</sup>针对仓储配送系统中的拣选货品操作问题, 建立人工拣选货品的排程模型, 综合考虑工人的疲劳程度、工作负荷以及货品的最大拣选时间, 降低人因因素导致的拣货失误率。Battini 等<sup>[13]</sup>建立以工人能量消耗为基础的货品储位模型, 对储位优化问题进行研究。通过上述文献回顾发现近年来学者在仓储方面的研究开始关注工人疲劳因素, 但在订单分批方面还相对匮乏。然而, 忽视工人疲劳因素会导致订单分批与排序的优化结果在实际应用中难以有效实施, 从而限制了理论研究成果向实践应用的转化。

综上, 已有关于订单分批与排序问题的文献大都假定人的工作速度是恒定不变且持续工作, 没有考虑工人疲劳对工作效率的影响。本文在考虑工人疲劳与休息的基础上, 针对 B2C 电商人工拣货系统中订单分批与排序问题进行研究, 以总完工时间最小化为目标, 构建混合整数规划模型, 并设计基于控制思想的种子算法对不同规模问题进行求解, 最后通过算例分析来验证算法的有效性, 并进行相关因素的敏感性分析。

## 1 模型构建

### 1.1 问题描述

本研究聚焦于 B2C 电商仓库在某时段内接收到多个订单, 旨在探讨这些订单的分批拣货与排序优化问题, 以提高订单分拣效率。基于分拣墙的拣选-合并两阶段的人工拣货系统布局如图 1 所示。

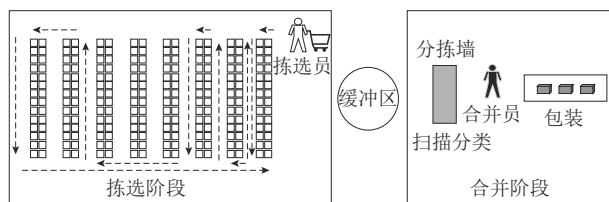


图 1 拣选-合并人工拣货系统布局

Figure 1 Layout of the manual pick-and-sort system

在货品拣选区和合并区各有一位工作人员, 拣选员按照 S 路径原则从货架上拣选完每个批次的所有货品之后, 将该批次货品送到缓冲区, 之后再返回拣选区进行下一批次货品的拣选。在合并阶段,

合并员需要将缓冲区的每一批货品按照客户订单进行分类合并, 然后送至发货区。本研究考虑缓冲区容量有限的情况, 即缓冲区可以容纳的批数目是有限的。当缓冲区有空余位置时, 拣选员将批次货品放置在缓冲区后可返回拣货区; 否则, 拣选员需要在缓冲区等待。

问题可描述为以下情况。假设有相同发货期的一组订单  $j \in J$ , 需要将订单  $j$  分配到批次  $b \in B$ , 批次  $b$  依次经过拣选区和订单合并区, 每个批次的货物不能超过拣选车的容量  $s$ 。拣选员按照 S 型路线在拣选区拣选货物。拣选区和合并区中间的缓冲区容量为  $q$ , 当缓冲容量达到  $q$  时, 拣选员需要等待; 当缓冲区存量为 0 时, 合并员需要等待。

本研究旨在通过优化订单分批与排序, 实现所有订单总完成时间的最小化。同时, 充分考虑工人的疲劳因素, 在拣选和合并环节中根据能量消耗情况, 设置最优的工人休息位置。通过实施休息策略, 可以有效避免工人过度疲劳, 从而在提升员工满意度的同时提高工作效率。

本文引入疲劳系数来描述工人的疲劳程度。疲劳系数被定义为工人因疲劳而导致工作效率下降的系数, 它直接反映了工作时长增加对工作效率的负面影响。通过这一量化指标, 能够更加准确地评估工人在不同工作时长下的实际工作效率, 并为后续的订单分批与排序联合优化提供重要依据。

本文问题在本质上可归结为缓冲区有限的两阶段流水车间调度问题, Papadimitriou 等<sup>[14]</sup>不仅证明该问题为 NP 完全问题, 而且还分析了当缓冲区容量为 1 时, 相比较无缓冲区流水车间可以节约 33% 的完工时间。关于缓冲区容量设置的相关研究<sup>[15]</sup>表明, 生产线绩效会随着缓冲区规模的增加而迅速下降。关于缓冲区容量问题, 本文将在参数设置和敏感性实验中做进一步分析。

基于上文问题描述, 考虑工人疲劳与休息的订单分批与排序优化问题的假设条件如下。

1) 一个订单只能划分到一个批次, 订单不允许被分割成多个批次。

2) 不允许某个订单批次在拣选或者合并过程中断, 工人疲劳度是逐渐累积的过程, 当工人休息后疲劳度从零开始累积。

3) 当工人工作负荷水平达到高水平的时候, 工

人需要休息, 否则不需要休息。

4) 在同一批次中, 工人的工作效率相同。

## 1.2 工人休息判断及考虑疲劳与休息的工作时间计算

鉴于工人疲劳与休息在拣货调度问题中的重要性, 本研究聚焦于确定工人何时需要休息。为了实现这一目标, 本文通过计算工人能量消耗值、最大可接受工作时间以及工人能量消耗阈值来评估工人的工作负荷水平, 从而判断工人休息批次, 为后续模型提供重要依据。能量消耗值  $Q$  由相对代谢率<sup>[16]</sup>、基础代谢率、工人表面积和工作时间决定。工人的能量消耗阈值  $\bar{Q}$  由最大可接受工作时间  $D_{MAW}$ <sup>[17]</sup> 和工人单位时间内的能量阈值  $\bar{E}$ <sup>[18]</sup> 决定。工人休息与否取决于其能量消耗值  $Q$  与设定的能量阈值  $\bar{Q}$  之间的关系。当工人的能量消耗值达到或超过该阈值时, 工人将会休息以恢复体力; 若能量消耗值低于阈值, 则工人将继续工作。

工人能量消耗值的相关中间参数 (工人表面积  $S_{BSA}$ 、相对代谢率  $R_{RM}$ 、基础代谢率  $R_{BM}$ 、相对心率  $H_R$ )、最大可接受工作时间  $D_{MAW}$ 、能量消耗阈值  $\bar{Q}$  和工人操作第  $n$  项作业时所需的能量消耗值  $Q_n$  的具体计算如下。

$$S_{BSA} = 0.0061B + 0.0128W - 0.1529; \quad (1)$$

$$R_{RM} = 0.072H_{work} - 5.608; \quad (2)$$

$$R_{BM} = \frac{[(13.7W + 5.0B) - (6.8A) + 66]S_{BSA}}{24}; \quad (3)$$

$$H_R = [(H_{work} - H_{rest}) / (H_{max} - H_{rest})] \times 100\%; \quad (4)$$

$$D_{MAW} = -2.67 + e^{7.02 - (5.72H_R)}; \quad (5)$$

$$\bar{Q} = D_{MAW}\bar{E}; \quad (6)$$

$$Q_n = (R_{RM} + 1.2)(R_{BM}S_{BSA}/60)T_n. \quad (7)$$

其中,  $B$  表示身高, 单位为 cm;  $W$  表示体重, 单位为 kg;  $H_R$  为相对心率, 单位为 beat/min;  $A$  为年龄;  $H_{work}$  为工人工作时心率,  $H_{rest}$  为工人休息时心率,  $H_{max}$  为工人最大心率;  $\bar{E}$  表示工人单位时间内的能量阈值;  $T_n$  表示第  $n$  项作业所需的操作时长。

工人疲劳会对工作时长产生影响, 导致工人拣选与合并时间的计算更加复杂, 为了便于构建模型, 以下将对拣选与合并时间的具体计算思路进行

说明, 以确保模型能够反映工人疲劳对拣货效率的影响。

工人在执行第  $n$  项作业时, 其疲劳度会随着工作量的累积而逐渐上升<sup>[12]</sup>。这种疲劳度的增加直接导致工人在进行后续作业时所需时间延长。值得注意的是, 每当工人进行一次休息后, 其疲劳度会重置至初始状态, 即疲劳度归零。当充分考虑工人的疲劳状态和休息时间时, 作业时长会依据工人的休息位置及其特有的疲劳系数而变化, 这种变化具体表现为一种指数函数  $(1+\alpha)^{n-k_n}$  的形式。因此, 当综合考虑疲劳和休息的影响后, 工人的作业时长  $L_n$  表示为

$$L_n = T_n(1+\alpha)^{n-k_n} + t_r Y_n. \quad (8)$$

其中,  $T_n$  表示不考虑疲劳时执行第  $n$  项作业的工作时长;  $\alpha$  表示工人疲劳系数;  $t_r$  表示工人休息时间;  $Y_n$  表示工人是否需要在第  $n$  项作业时休息;  $k_n$  表示工人在操作第  $n$  项作业之前的最近一次休息位置。

此外, 针对拣选员与合并员在两个阶段操作订单批次的时间逻辑约束设置<sup>[7]</sup>, 即第  $n$  个批次的拣选完成时间不仅要大于第  $n-q-2$  个批次的合并完成时间与拣选时长之和, 而且还要大于第  $n-1$  个批次的拣选完成时间与拣选时长之和; 第  $n$  个批次的合并完成时间不仅要大于第  $n$  个订单批次的拣选完成时间与合并时长之和, 还要大于第  $n-1$  个批次的合并完成时间与合并时长之和。当考虑了疲劳因素和工人休息之后, 工人的工作时长则变为  $L_n$ , 时间计算逻辑约束也需要作出相应的调整。

### 1.3 模型

模型中的符号说明及决策变量定义如表 1 所示。

表 1 符号说明

Table 1 Notations and descriptions

符号	符号说明
$I$	SKU 集合, $i \in I$
$J$	订单集合, $j \in J$
$K$	通道集合, $k \in K$
$B$	订单批次集合, $b \in B$
$N$	订单批次的排序集合, $n \in N$
$S_j$	订单 $j$ 中的 SKU 集合
$R_b$	订单批次 $b$ 中的订单集合
$U_b$	订单批次 $b$ 中的 SKU 集合
$G_b$	订单批次 $b$ 中的 SKU 所在的通道集合

续表

符号	符号说明
$c_i$	SKU $_i$ 占据拣选车的容量
$s$	拣选车最大容量
$l_{bi}$	订单批次 $b$ 中 SKU $_i$ 的数量
$f_{ij}$	订单 $j$ 中 SKU $_i$ 的数量
$h_i$	SKU $_i$ 所在的通道
$l$	通道长度
$w$	通道间宽度
$v$	拣选员的行走速度
$t^{\text{scan}}$	单个商品平均扫描时间
$t^{\text{pack}}$	每个订单平均打包时间
$q$	缓存区域容量
$t_r^p$	拣选员固定休息时长
$t_r^s$	合并员固定休息时长
$\alpha_1$	拣选员疲劳系数
$\alpha_2$	合并员疲劳系数
$D_n^p$	第 $n$ 批次顺序的拣选员工作负荷水平
$D_n^s$	第 $n$ 批次顺序的合并员工作负荷水平
$Q_n^p$	第 $n$ 批次顺序的拣选员能量消耗值
$Q_n^s$	第 $n$ 批次顺序的合并员能量消耗值
$\bar{Q}$	工人能量消耗阈值
$R_{BM}^p$	拣选员的基础代谢率
$R_{BM}^s$	合并员的基础代谢率
$S_{BSA}^p$	拣选员的身体表面积
$S_{BSA}^s$	合并员的身体表面积
$R_{RM}^p$	拣选员的相对代谢率
$R_{RM}^s$	合并员的相对代谢率
$M$	非常大的数
$X_{jb}$	0-1 变量; 如果订单 $j$ 分到订单批次 $b$ , $X_{jb} = 1$ ; 否则, 为 0
$Z_{bn}$	0-1 变量; 如果批次 $b$ 排在第 $n$ 个位置, $Z_{bn} = 1$ ; 否则, 为 0
$Y_n^p$	0-1 变量; 如果在拣选第 $n$ 个订单批次之前拣选员有休息, $Y_n^p = 1$ ; 否则, 为 0
$Y_n^s$	0-1 变量; 如果在合并第 $n$ 个订单批次之前合并员有休息, $Y_n^s = 1$ ; 否则, 为 0
$k_n^p$	拣选员在拣选第 $n$ 个订单批次之前的最晚休息位置 ( $k_n^p = 1, 2, \dots, n$ 且 $k_1^p = 1$ )
$k_n^s$	合并员在合并第 $n$ 个订单批次之前的最晚休息位置 ( $k_n^s = 1, 2, \dots, n$ 且 $k_1^s = 1$ )
$t_b^p$	订单批次 $b$ 的拣选时长
$t_b^s$	订单批次 $b$ 的合并时长
$T_n^p$	不考虑疲劳与休息的第 $n$ 个订单批次的拣选时长
$T_n^s$	不考虑疲劳与休息的第 $n$ 个订单批次的合并时长
$L_n^p$	考虑疲劳与休息的第 $n$ 个订单批次的拣选时长
$L_n^s$	考虑疲劳与休息的第 $n$ 个订单批次的合并时长
$C_n^p$	第 $n$ 个订单批次的拣选完成时间
$C_n^s$	第 $n$ 个订单批次的合并完成时间

基于上述工人负荷水平计算、拣货与合并时间的计算分析及参数设置, 建立数学模型如式 (9)~(34) 所示。

$$\min C_{\max} \circ \quad (9)$$

s.t.

$$\sum_{b \in B} X_{jb} = 1, \quad \forall j \in J; \quad (10)$$

$$\sum_{b \in B} Z_{bn} = 1, \quad \forall n \in N; \quad (11)$$

$$\sum_{n \in N} Z_{bn} = 1, \quad \forall b \in B; \quad (12)$$

$$\sum_{i \in T_b} l_{bi} c_i \leq s, \quad b \in B; \quad (13)$$

$$t_b^p = \frac{1}{v} \left[ 2[\max_{i \in U_b} h_i - 1]w + |G_b|l \right], \quad \forall i \in I, j \in J, b \in B; \quad (14)$$

$$t_b^s = \left( \sum_{i \in U_b} l_{bi} \right) t^{\text{scan}} + |R_b| t^{\text{pack}}, \quad j \in J, b \in B; \quad (15)$$

$$T_n^p \geq t_b^p + M(Z_{bn} - 1), \quad \forall b \in B, n \in N; \quad (16)$$

$$T_n^s \geq t_b^s + M(Z_{bn} - 1), \quad \forall b \in B, n \in N; \quad (17)$$

$$L_n^p = T_n^p(1 + \alpha_1)^{n-k_n^p} + t_r^p Y_n^p, \quad \forall n \in N; \quad (18)$$

$$L_n^s = T_n^s(1 + \alpha_2)^{n-k_n^s} + t_r^s Y_n^s, \quad \forall n \in N; \quad (19)$$

$$C_n^p \geq C_{n-q-2}^s + L_n^p, \quad \forall n \in N; \quad (20)$$

$$C_n^p \geq C_{n-1}^p + L_n^p, \quad \forall n \in N; \quad (21)$$

$$C_n^s \geq C_n^p + L_n^s, \quad \forall n \in N; \quad (22)$$

$$C_n^s \geq C_{n-1}^s + L_n^s, \quad \forall n \in N; \quad (23)$$

$$C_n^p \leq C_{n-1}^s, \quad \forall n \in N; \quad (24)$$

$$C_{n-q-1}^s \leq C_n^p, \quad \forall n \in N; \quad (25)$$

$$k_n^p = k_{n-1}^p(1 - Y_n^p) + nY_n^p, \quad \forall n \in N; \quad (26)$$

$$k_n^s = k_{n-1}^s(1 - Y_n^s) + nY_n^s, \quad \forall n \in N; \quad (27)$$

$$Q_n^p = (R_{\text{BM}}^p + 1.2)(R_{\text{BM}}^p S_{\text{BSA}}^p / 60) \sum_{k_n^p} T_n^p(1 + \alpha_1)^{n-k_n^p}, \quad \forall n \in N; \quad (28)$$

$$Q_n^s = (R_{\text{BM}}^s + 1.2)(R_{\text{BM}}^s S_{\text{BSA}}^s / 60) \sum_{k_n^s} T_n^s(1 + \alpha_2)^{n-k_n^s}, \quad \forall n \in N; \quad (29)$$

$$D_n^p = Q_n^p / \bar{Q}, \quad \forall n \in N; \quad (30)$$

$$D_n^s = Q_n^s / \bar{Q}, \quad \forall n \in N; \quad (31)$$

$$Y_n^p = 1 - I(D_n^p), \quad \forall n \in N; \quad (32)$$

$$Y_n^s = 1 - I(D_n^s), \quad \forall n \in N; \quad (33)$$

$$C_{\max} \geq C_n^s, \quad \forall n \in N. \quad (34)$$

其中,  $I(x)$  是一个示性函数,  $I(x) = \begin{cases} 0, & x \geq 1; \\ 1, & x < 1. \end{cases}$

式 (9) 目标函数表示最小化最大总完工时间。约束 (10) 表示每个订单只能分配到一个订单批次中; 约束 (11) 表示每个订单批次只能安排在一个顺序位置上; 约束 (12) 表示每个排序的位置只能有一个订单批次; 约束 (13) 确保一个订单批次中的商品数量不能超过拣选车的最大拣选商品数量, 其中  $R_b = \{j | X_{jb} = 1\}$ , 表示订单批次  $b$  中的订单集合, 相应的可以得到  $l_{bi} = \sum_{j \in R_b} f_{ij}$ , 表示订单批次  $b$  中的 SKU  $i$  的数量; 约束 (14) 表示拣选员拣选订单批次  $b$  所需要的拣选时长, 其中,  $U_b = \cup_{j \in R_b} S_j$ , 表示订单批次  $b$  中 SKU 集合,  $G_b = \{h_i | i \in U_b\}$  表示订单批次  $b$  中商品所在通道集合; 约束 (15) 表示合并员合并订单批次  $b$  所需要的合并时长; 约束 (16) 和 (17) 分别界定了不考虑疲劳和休息的第  $n$  个订单批次的拣选时长和合并时长; 约束 (18) 和 (19) 为考虑疲劳和休息的第  $n$  个订单批次的拣选时长和合并时长。

约束 (20)~(25) 为两阶段工人处理订单批次的时间逻辑约束。约束 (20) 和 (21) 界定了拣选员拣选第  $n$  个订单批次的拣选完成时间; 约束 (22) 和 (23) 界定了合并员合并第  $n$  个订单批次的合并完成时间; 约束 (24) 与 (25) 保证了前后订单批次拣选与合并流程之间的协调。

约束 (26) 表示拣选员直到第  $n$  个订单批次的最晚休息位置; 约束 (27) 表示合并员直到第  $n$  个订单批次的最晚休息位置; 约束 (28) 表示拣选员拣选第  $n$  个订单批次的能量消耗值; 约束 (29) 表示合并员合并第  $n$  个订单批次的能量消耗值; 其中, 工人基础代谢率、相对代谢率和身体表面积的计算在前

文有详细解释；约束 (30) 表示拣选员拣选第  $n$  个订单批次的劳动工作负荷水平；约束 (31) 表示合并员合并第  $n$  个订单批次的劳动工作负荷水平；约束 (32) 表示拣选员是否在拣选第  $n$  个订单批次前休息；约束 (33) 表示合并员是否在合并第  $n$  个订单批次前休息；约束 (34) 表示第  $n$  个订单批次在仓库内的总完工时间不大于总完工时间。

## 2 算法设计

### 2.1 基于控制思想的种子算法

根据文献[3]，精确算法可以求解小规模订单分批问题，但当订单数量升至 100 以上时大多采用启发式算法，具体包括优先规则算法、节约里程法、种子算法和元启发式算法。Jiang 等<sup>[7]</sup>采用基于控制思想的种子算法解决此类问题，其算法逻辑是通过经典的种子算法生成初始订单批次，利用控制思想更改种子策略，进而调控每一个批次的生成规则，以使得拣选和合并两个过程阶段达到平衡。上述算法求解质量和求解效率均优于经典种子算法。本文将改进上述算法，通过加入工作负荷水平判断、工人休息决策来解决考虑工人疲劳与休息的订单分批与排序问题。整个算法的核心设计包括订单批次生成、工人工作负荷水平与休息位置判断、基于控制思想、种子订单调整和订单批次排序 4 个部分。

#### 1) 订单批次生成。

在订单批次生成阶段，由种子算法生成一个订单批次。本文采用的种子算法主要思想解释如下。在当前未完成分批的订单中选择覆盖通道数最小的订单作为该批次的种子订单；在该批次没有达到容量限制的时候，将当前待分批订单中与种子订单相似度最高的订单作为附加订单加入当前批次。这里的相似度通过  $S_{ij} = |A_i \cap A_j| / |A_i \cup A_j|$  来计算，其中， $A_i$  表示订单  $i$  覆盖的通道集合； $A_j$  表示当前批次  $j$  中订单所覆盖的通道集合； $|A_i \cap A_j|$  表示订单  $i$  和批次  $j$  同时覆盖的通道数量； $|A_i \cup A_j|$  表示把订单  $i$  和批次  $j$  所覆盖的所有通道数量。

#### 2) 工人工作负荷水平与休息位置判断。

根据约束 (30) 和 (31) 计算上一个休息位置到当前批次 (第  $n$  个订单批次) 的工人工作负荷水平  $D_n$ 。文中的工作负荷水平  $D_n$  可以定义为工人工作时能量消耗值与能量消耗阈值之比。如果  $D_n \geq 1$ ，视为高负荷，则工人需要在这一批次前进行休息，此时，

第  $n$  个订单批次的拣选完成时间或合并完成时间需要加上相应的休息时长；如果  $D_n < 1$ ，则工人不需要休息，相应的拣选完成时间或合并完成时间则不需要加入休息时长。

#### 3) 基于控制思想的种子订单调整。

根据约束 (20) 和 (21) 计算当前批次 (第  $n$  个订单批次) 的拣选完成时间，根据约束 (22) 和 (23) 计算第  $n$  个订单批次的合并完成时间。根据约束 (25) 判断是否发生拣选员等待现象，如果第  $(n-z-1)$  个订单批次的合并完成时间大于等于第  $n$  个订单批次的拣选完成时间，即  $C_n^p \geq C_{n-q-1}^s$ ，说明拣选员等待未发生，此时需要使用种子原则 1，即无需更改种子订单选择规则，继续使用最小通道数规则选择种子订单。

如果第  $n$  个批次的拣选完成时间小于第  $(n-z-1)$  个订单批次的合并完成时间，即  $C_n^p < C_{n-q-1}^s$ ，说明拣选员等待现象发生，此时需要使用种子原则 2，即选择比第  $n$  批次种子订单所覆盖通道数多 1 的订单作为种子订单。如果不存在覆盖通道数多 1 的订单，则需要选择数量多 2 的订单作为种子订单，以此类推。

#### 4) 订单批次排序。

通过判断是否发生拣选员等待现象，进而决定当前批次是否排序。如果拣选员等待发生，需要将当前分好批次的订单重新放入未分批的订单中，更换种子订单原则，调整种子订单，直到拣选员等待未发生；但是，如果目前所有待分批的订单已经不存在满足要求的种子订单了，那么将当前批次订单直接放入排序的最后一位，开始下一批的分批工作，直到没有待分批订单为止。

本文设计的基于控制思想的种子算法具体流程如图 2 所示。

### 2.2 两阶段法

为了深入探究工人疲劳与休息对订单分批与排序的具体影响，将两阶段求解方法与本文的联合优化求解结果进行对比分析。在两阶段方法中，第 1 阶段是借鉴文献[7]的研究成果，对订单进行分批优化与排序；第 2 阶段则基于第 1 阶段的输出结果，进一步引入工人疲劳和休息参数，对工人的休息批次位置进行优化。下面将采用伪代码的方式描述两阶段法的求解过程。第 1 阶段中，种子订单选择规则 AilseA 是指订单池中覆盖通道数最少的订单；种子订单选择规则 AilseB 是指从订单池中选择比 AilseA 的种子订单覆盖通道数增加 1 个的订单，如

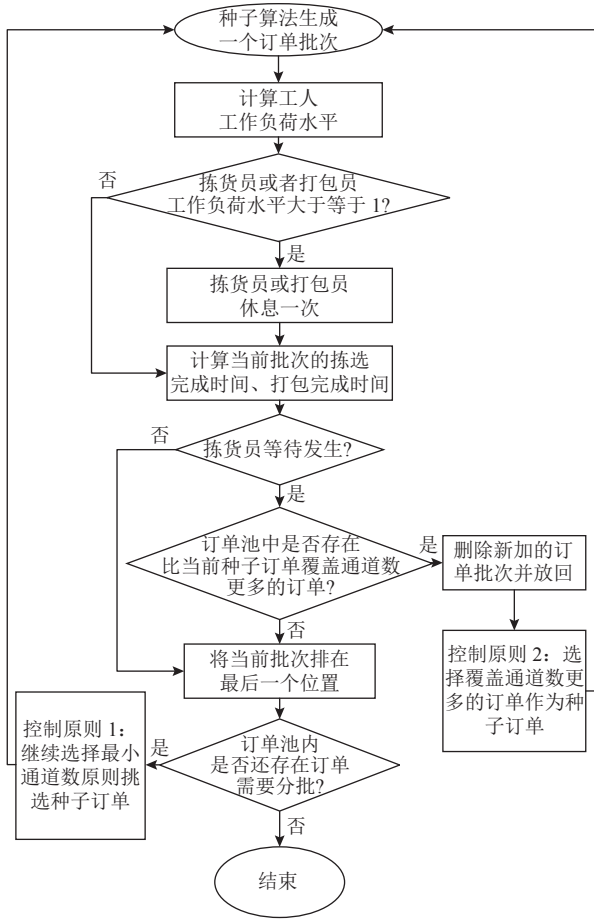


图2 考虑工人疲劳与休息的基于控制思想的种子算法基本流程

Figure 2 Flowchart of the control-based seed algorithm considering worker fatigue and rest

果没有就增加2个, 以此类推; 附加订单选择规则 Accompany 是指选择与种子订单所覆盖的通道相似度最大的订单; orders、C、batch 是指订单池(所有待分批的订单)、拣选车容量、虚拟拣货车; Batch、 $C_{max1}$  是指分批结果、总完工时间。第1阶段伪代码如下。

两阶段法第1阶段

输入 AilseA、AilseB、Accompany、orders、C、batch

输出 Batch、 $C_{max1}$

```

for each  $i$  in length (orders) do
    根据 AilseA 选择 orders[ $i$ ]、batch.append (orders[ $i$ ])、del orders[ $i$ ]
do//新批次生成
while length (orders) > 0 do
    根据 AilseB 选择 orders[ $i$ ]、batch.append (orders[ $i$ ])、del orders[ $i$ ]
if length (batch) < C then
if length (orders) < 0 then
    update length (orders) (重新开始 while 循环)
if length (orders)  $\geq$  0 then
    Batch.append (batch)

```

```
end for //结束整个基于控制思想的种子算法
```

```
return Batch、 $C_{max1}$ 
```

```
if length (batch)  $\geq$  C then
```

```
Batch.append (batch)
```

```
del batch
```

```
计算 Batch 的  $C_{max1}$ 
```

```
if 停滞状态发生 then
```

```
if 订单池中存在比新加入订单组的种子订单所覆盖的通道数多的订单 then
```

```
del Batch 中新加入的 batch, 并将此 batch 中的所有订单放回到 orders 中, 然后根据 AilseA 选择 orders[ $i$ ]、batch.append (orders[ $i$ ])、del orders[ $i$ ] repeat//返回最开始, 进行新批次的生成
```

```
if 订单池中不存在比新加入订单组的种子订单所覆盖的通道数多的订单 then
```

```
Batch.append (batch), 根据 AilseA 选择 orders[ $i$ ]、batch.append (orders[ $i$ ])、del orders[ $i$ ] repeat//返回最开始, 重新进行新批次的生成
```

```
if 停滞状态未发生 then
```

```
Batch.append (batch), 根据 AilseA 选择 orders[ $i$ ]、batch.append (orders[ $i$ ])、del orders[ $i$ ] repeat//返回最开始, 重新进行新批次的生成
```

第1阶段得出订单分批和排序的结果, 第2阶段将以这个结果作为输入, 在此基础上计算工人休息位置以及总完工时间  $C_{max2}$ 。第2阶段伪代码如下。

两阶段法第2阶段

输入 Batch

输出 rest1、rest2、 $C_{max2}$

记第1批次开始前为休息1次

```
for each  $n$  in length (Batch) do
```

根据约束 (33) 和 (34) 计算 Batch[ $n$ ]的工人工作负荷水平  $D_n$

```
if  $D_n \geq 1$  then
```

工人需要在这一批次前进行休息, rest1.append ( $n$ ) 或者 rest2.append ( $n$ ),  $Y_n=1$

根据约束 (20)~(23) 计算 Batch[ $n$ ]加入疲劳系数以后的拣选完成时间、合并完成时间、 $C_{max2}$

```
if  $D_n < 1$  then
```

工人不需要在这一批次前进行休息,  $Y_n=0$

根据约束 (20)~(23) 计算 Batch[ $n$ ]加入疲劳系数以后的拣选完成时间、合并完成时间、 $C_{max2}$

```
end if
```

```
end for
```

```
return rest1、rest2、 $C_{max2}$ 
```

## 3 算例分析

### 3.1 算例设置

在已有标准算例<sup>[19]</sup>基础上生成多个算例集  $S$ , 每个  $S$  包含5个算例, 算例集  $S_1 \sim S_5$  的订单量分别为300、350、400、450、500; 每个订单中SKU的

个数服从 1~9 的均匀分布。根据文献[7]和实际调研设置相关拣货系统参数,其中,拣货通道长为 45 m,通道宽为 5 m,拣选员行走速度为 45 m/min,合并员每扫描一个商品需要 0.1 min 且每合并一件商品需要 0.3 min,拣选车的容量为 50 件商品。已有研究<sup>[7]</sup>结果表明,缓冲区容量大小在一定条件下对最后的总完工时间结果影响不大,因此为了简化问题,本文算例设置缓冲区容量为 1,后续会针对缓冲区容量进行相应的敏感性分析。

根据文献[20],拣选工作属于重劳动,合并工作属于中劳动,因此本文设定拣选员的疲劳系数大于合并员。本文假设拣选员的疲劳系数为 0.09,合并员的疲劳系数为 0.02;工人身高和体重服从 168~180 cm、65~85 kg 的均匀分布;单位时间内的拣选员能量阈值为 10.056,合并员能量阈值 4.933;根据已有研究<sup>[10,12]</sup>以及两种工作的强度差异<sup>[20]</sup>,设置拣选员与合并员的休息时间分别为 5 min 与 3 min。基于文献[12],拣选员和合并员的其他相关参数如表 2 所示。

表 2 工人基本信息

Table 2 Basic information of workers

工人类型	身高/cm	体重/kg	年龄/岁	$H_{rest}/(\text{beat}\cdot\text{min}^{-1})$	$H_{max}/(\text{beat}\cdot\text{min}^{-1})$	$H_{work}/(\text{beat}\cdot\text{min}^{-1})$
拣选员	178	75	35	60	170	135
合并员	178	75	35	60	160	110

表 3 基于控制思想的种子算法与遗传算法对比<sup>1)</sup>

Table 3 Comparison between the control-based seed algorithm and the genetic algorithm

算例集	基于控制思想的种子算法(CBSA)		混合遗传算法(SGA)		总完工时间差距(GAP <sub>1</sub> )/%
	总完工时间/min	求解时间/s	总完工时间/min	求解时间/s	
$S_1$	273.55	1.46	294.37	1.78	-7.07
$S_2$	318.71	1.54	347.03	2.01	-8.16
$S_3$	365.25	1.89	389.72	2.40	-6.28
$S_4$	410.26	1.68	442.52	2.90	-7.29
$S_5$	449.45	1.94	478.55	3.04	-6.08

1)  $GAP_1 = [(CBSA \text{ 的总完工时间} - SGA \text{ 总完工时间}) / SGA \text{ 总完工时间}] \times 100\%$ 。

从表 3 可知,随着问题规模的逐步扩大,SGA 运行时间呈现出明显的增长趋势。相比之下,CBSA 不仅运行时间更短,而且其增长幅度也相对较小,显示出较高的求解效率。在求解质量方面,CBSA 同样表现出色,其求解的总完工时间始终优

### 3.2 算法验证

根据综述文献[3]的梳理,在订单分批问题的研究中,有超过 50% 的研究者选择遗传算法作为解决方案。本文将以种子算法与遗传算法相结合的混合算法(SGA)作为对比来验证基于控制思想的种子算法(CBSA)的有效性。本文设计的混合遗传算法通过采用种子算法生成初始种群,在此基础上以  $obj_v = 1/C_{max}$  作为适应度函数,其中,  $C_{max}$  代表总完工时间,适应度函数大,代表总完工时间越小;设计轮盘赌选择策略,将批次与批次之间的订单进行交叉,交叉后的两个订单组容量不超过拣选车容量;将不同的订单批次顺序进行交换,达到变异的目的。与基于控制思想的种子算法不同,遗传算法是通过交叉、变异思想不断更替种群,休息位置也会随种群而变化,进而根据工人工作负荷水平判断工人需要在哪一批次前进行休息。

在算法运行方面,采用 Python3.7.0 编程,运行环境为 Inter (R) Core (TM) i5-8250U CPU @1.60 GHz, 8 GB 内存, Windows10 操作系统。在求解结果处理方面,CBSA 在同一算例上每次求解结果相同,所以对每个算例集的 5 个算例分别求解一次,并将其均值用于结果分析;而 SGA 在同一算例上每次求解结果不同,因此针对 SGA 每个算例集中的 5 个算例分别求解 10 次,并将其均值用于结果分析。两种算法针对不同算例集下的具体求解结果见表 3。

于 SGA。具体来说,对于所有问题集,CBSA 求解结果均比 SGA 优化了 6.08%~8.16%。因此,可以得出,针对不同问题集,CBSA 都可以在较短的时间内找到比 SGA 更优的解,这充分证明 CBSA 在求解效率和质量上的优越性。

### 3.3 对比分析

本文的研究问题是在考虑工人疲劳与休息因素

的基础上实现订单分批与排序的联合优化。表 4 显示了不同问题集下两种优化方法的实验结果。

表 4 两阶段求解与本研究联合优化结果对比<sup>1)</sup>

Table 4 Result comparison of the two-stage method and the proposed joint optimization method

算例集	优化方法	拣选员平均等待次数	合并员平均等待次数	拣选员平均休息次数	合并员平均休息次数	总完工时间/min	GAP <sub>2</sub> /%
S <sub>1</sub>	两阶段求解	4.8	8.0	5.0	3.0	289.97	5.66
	本研究	0.8	1.8	5.1	3.0	273.55	
S <sub>2</sub>	两阶段求解	6.2	12.0	6.0	3.0	341.38	6.64
	本研究	3.4	0.4	6.2	3.2	318.71	
S <sub>3</sub>	两阶段求解	9.2	12.8	7.0	4.0	388.51	5.99
	本研究	5.8	0.6	7.0	4.0	365.25	
S <sub>4</sub>	两阶段求解	10.2	12.2	7.6	4.0	436.3	5.97
	本研究	9.2	0.8	7.8	4.2	410.26	
S <sub>5</sub>	两阶段求解	11.6	11.6	8.0	5.0	457.21	1.70
	本研究	11	0	8.2	5.0	449.45	

1) GAP<sub>2</sub> = [(两阶段求解目标函数值 - 本研究联合优化目标函数值) / 两阶段求解目标函数值] × 100%。

实验结果表明, 在两种不同的优化方法下, 虽然大部分算例的分批数量保持一致, 但拣选员和合并员的平均等待次数却呈现出显著差异。针对各种规模的算例集, 拣选员在两阶段求解方法下的平均等待次数分别为 4.8、6.2、9.2、10.2 和 11.6, 而在联合优化方法下, 这一数字则降低至 0.8、3.4、5.6、10.2 和 11.6。对于合并员而言, 在联合优化方法下的平均等待次数均保持在 0~2 次的低范围内, 而两阶段求解方法求得的等待次数则分布在 8~12 次之间。这一发现表明, 联合优化求解方法在减少工人空闲时间方面表现优越, 尤其是在处理规模为 300~450 的算例集时, 其优化效果更为显著。

此外, 还观察到在所有算例集中, 两种优化方法下的拣选员和合并员的休息次数存在细微差异。具体来说, 联合优化方法下的休息次数略多于两阶段优化方法。这种安排不仅有助于提高拣货效率, 更重要的是能够提升员工满意度, 因为适当休息有助于员工保持更好的工作状态。实验还发现两种优化方法下的休息批次位置以及订单的具体分批结果都存在不同。

总体来看, 相较于两阶段优化方法, 同时考虑疲劳、休息、分批和排序的联合优化方法在缩短总完工时间方面表现更优。不同规模的算例集减少的比例不尽相同, 当订单数量在 300~450 之间时, 总完工时间最多可降低 6.64%, 效果显著。然而,

当订单数量达到 500 时, 两种优化方法之间的差距则相对较小。这表明, 在中小规模的算例集中, 采用联合优化方法显得尤为必要, 有助于提升订单履行效率。

### 3.4 拣选员与合并员的劳动强度差异对总完工时间的影响

基于文献[20], 拣选员相较于合并员更容易感到疲劳, 在上述实验中设定拣选员与合并员的疲劳系数之比为 4.5:1。具体来说, 合并员的疲劳系数  $\alpha_2$  为 0.02, 而拣选员的疲劳系数  $\alpha_1$  为 0.09。然而, 考虑到现代配送中心普遍采用拣选辅助设备来减轻员工的工作强度, 从而降低拣选员的疲劳程度。这里针对拣选员与合并员的疲劳系数之比进行敏感性实验, 以探究两种员工的劳动强度差异变化对整体拣货效率的影响。通过调整拣选员与合并员的疲劳系数之比, 可以更准确地模拟实际工作环境, 并为配送中心的优化管理提供更有价值的参考。

图 3 显示了疲劳系数比值变化对两种优化方法订单总完工时间差距所产生的影响。实验结果展示了一个明显趋势: 对于所有规模的算例集而言, 随着拣选员与合并员疲劳系数比值的不断增大, 联合优化方法相较于两阶段方法在订单总完工时间上的优势差距也在逐步增加。具体来说, 对于小规模算例 (包含 300 个订单), 订单总完工时间的差距从最初的 1.55% 逐步扩大至 5.66%; 而对于大规模的

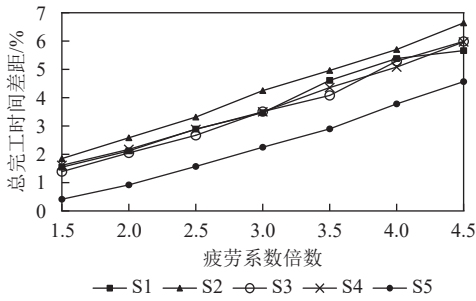


图 3 拣选员与合并员疲劳系数倍数对两种优化方法订单总完工时间差距的影响

Figure 3 Impacts of the fatigue coefficient ratios between pickers and sorters on the total order completion time gap using the two optimization methods

算例 (包含 500 个订单), 这一差距则从 0.41% 增加到了 4.57%。实验结果有力地证明了当拣选员与合

并员的工作强度比值增大时, 考虑工人疲劳与休息的联合分批优化方法显得更为重要。这种优化方法不仅有助于缩短订单的总完工时间, 同时也能够更好地保障工人的健康和工作满意度。

图 4 进一步揭示了两种优化方法在不同疲劳系数比值下的表现。对于两阶段优化方法而言, 在所有规模的问题集中, 随着拣选员与合并员疲劳系数比值的增加, 订单的总完工时间也呈现出明显的增长趋势。然而, 与之形成对比的是, 本文提出的联合优化方法在面对疲劳系数比值变化时, 表现出了卓越的稳定性。无论疲劳系数比值如何波动, 该方法下的订单总完工时间都基本保持不变。以  $S_1$  为例, 其订单总完工时间始终稳定在 271 min 左右。

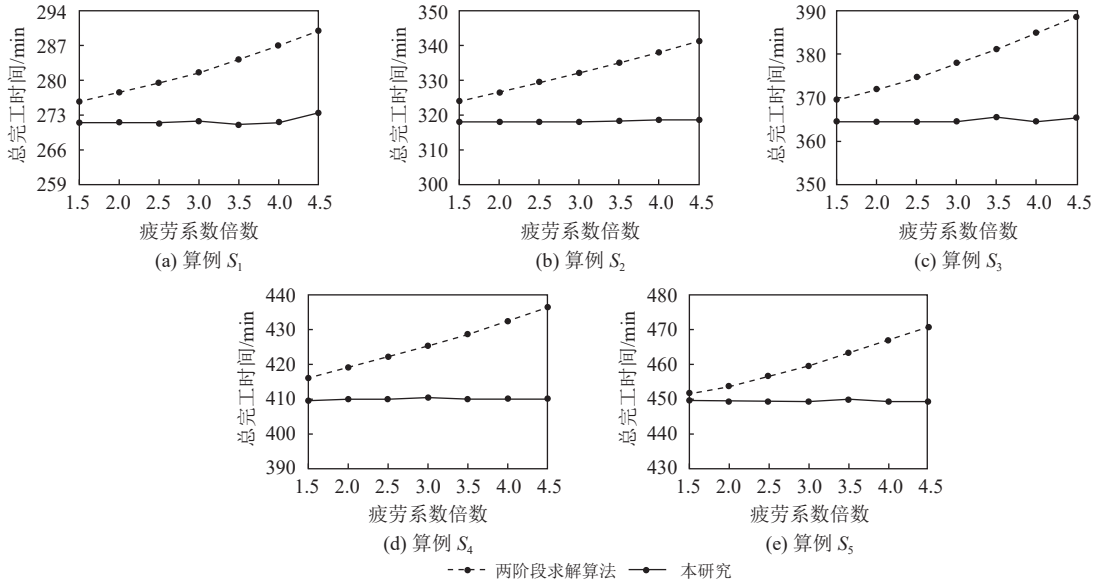


图 4 不同算例集下疲劳系数倍数对订单总完工时间的影响

Figure 4 Impacts of fatigue coefficient ratios on the total order completion time with various numerical examples

这一发现充分证明了在考虑工人休息与疲劳的情境下, 本文提出的联合优化方法不仅能够有效降低订单的总完工时间, 而且在实际应用中具有更高的稳定性。这种稳定性对于配送中心而言至关重要, 因为它意味着在面对各种不确定因素时, 联合优化方法仍然能够保证持续、高效的订单履行流程。

### 3.5 缓冲容量变化对总完工时间的影响

上述实验中设定拣选区与合并区之间的缓冲容量为 1, 但这一设定在现实场景中可能会有很大的变化。如图 5 所示, 当拣货区与合并区之间的缓冲容量为 1 时, 本研究提出的联合优化与两阶段优

化之间的订单完工时间差距最为显著, 前者相较于后者可以节约 6% 的订单总完工时间。然而, 随着拣选区和合并区之间缓冲容量的逐步增加, 联合优化与两阶段优化在订单完工时间上的差距逐渐缩小。当缓冲容量增至 6 时, 两种优化算法在订单完工时间上的差异已经变得微乎其微, 几乎可以忽略不计。但针对联合优化算法而言, 在 5 组不同的算例集中, 无论缓冲区容量如何变化, 总完工时间始终保持在相对稳定的水平。这一发现揭示了缓冲容量大小在两阶段法中的显著影响, 而本文提出的联合优化算法在面对这一变量时展现出了较强的稳健

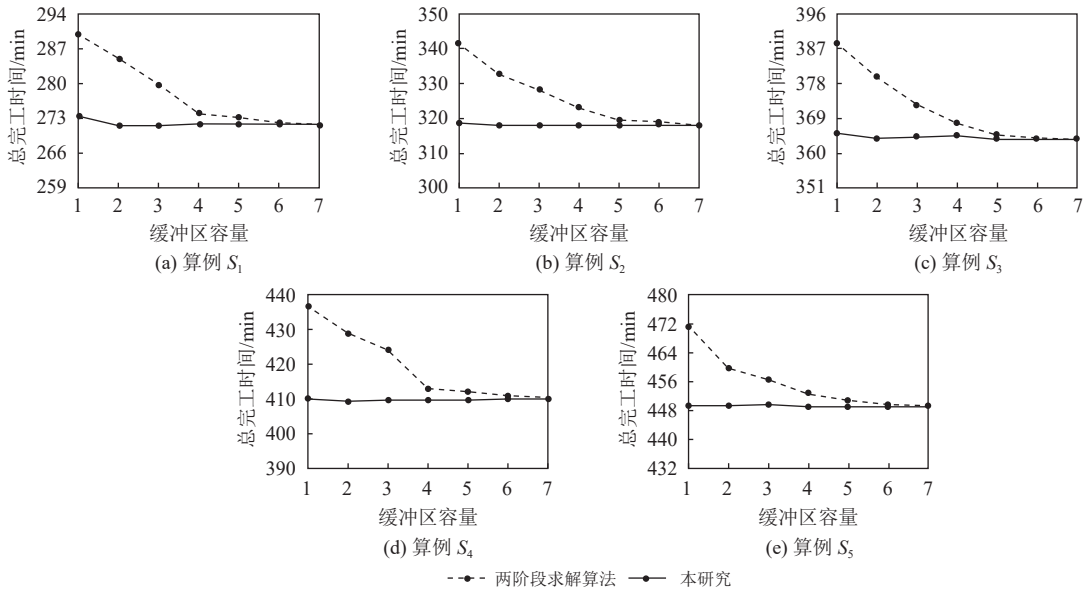


图 5 不同算例集下缓冲容量对总完工时间的影响

Figure 5 Impacts of buffer capacity on the total order completion time with various numerical examples

性, 其结果几乎不受缓冲容量大小的影响。这一发现也验证了文献[7]的研究结果。

## 4 结论

本文深入探讨了在 B2C 电子商务环境中, 在工人疲劳与休息因素影响下的订单分批拣货与排序问题。基于此, 构建了目标为最小化最大总完工时间的混合整数规划数学模型, 并采用基于控制思想的种子算法进行求解, 通过多种规模算例集验证了所提方法和模型的有效性与实用性。本研究不仅为电子商务订单处理提供新视角, 也为考虑工人福祉的订单管理系统设计提供了理论支持。

实验结果表明基于控制思想的种子算法针对所有算例集在求解质量、求解效率方面均优于遗传算法。对比联合优化和两阶段优化方法, 发现在中小规模算例实验中, 联合优化获得的订单完成总时间明显优于两阶段优化; 虽然两种优化方案的分批数量相同, 但每个批次的订单组成和工人休息位置却存在明显差异。此外, 在联合优化中, 拣选员停滞次数、合并员等待次数明显降低。这些发现不仅揭示了联合优化方法在订单分批和工人休息安排上的灵活性和优越性, 还证实了联合优化方法在提升工作效率和员工满意度方面的有效性。

通过两阶段疲劳系数比值的敏感性分析, 发现两阶段优化方法对疲劳系数比值变化比较敏感, 而

在联合优化方法下, 无论疲劳系数比值如何变化, 订单总完工时间保持基本不变。这表明在考虑工人疲劳与休息的分批与排序联合优化方法的框架下, 求解过程更加稳健, 能够有效应对疲劳系数变化带来的不确定性。缓冲区容量敏感性分析的结果表明, 尽管缓冲容量大小在两阶段法中起着显著作用, 但本文提出的联合优化算法在处理该变量时表现出了高度的稳定性, 使得订单总完工时间受缓冲容量大小的影响相对有限。上述发现为考虑工人疲劳与休息的订单分批提供了坚实的理论基础。在实际应用中, 当决策者面临拣选区与合并区之间的缓冲区容量不确定, 或拣选和合并工人的疲劳系数不确定时, 更应该采用考虑疲劳与休息的联合优化方法。这样的优化方法不仅有助于提高拣货效率, 还能在保障工人健康和工作效率的同时, 实现整体订单履行流程的优化。

本文仅针对已知的订单进行分批拣货与排序决策的研究, 结合动态订单进行实时分批是未来有价值的研究课题。此外, 本文假设工人在作业时的心率是固定不变的, 但是在很多情况下, 工人的心率是动态变化的, 这也是未来研究需要改进和深化的方向。

### 参考文献:

[1] BOYSEN N, DE KOSTER R, WEIDINGER F. Warehousing in the e-commerce era: a survey[J]. *European Journal of Opera-*

- tional Research, 2019, 277(2): 396-411.
- [2] GADEMANN N, van de VELDE S. Order batching to minimize total travel time in a parallel-aisle warehouse[J]. IIE Transactions, 2006, 37(1): 63-75.
- [3] PINTO A R F, NAGANO M S. A comprehensive review of batching problems in low-level picker-to-parts systems with order due dates: main gaps, trade-offs, and prospects for future research[J]. Journal of Manufacturing Systems, 2022, 65: 1-18.
- [4] MENENDEZ B, BUSTILLO M, PARDO E, et al. General variable neighbourhood search for the order batching and sequencing problem[J]. European Journal of Operational Research, 2017, 263(1): 82-93.
- [5] BOYSEN N, STEPHAN K, WEIDINGER F. Manual order consolidation with put walls: The batched order bin sequencing problem[J]. EURO Journal on Transportation and Logistics, 2019, 8(2): 169-193.
- [6] ARDJMAND E, BAJGIRAN O S, YOUSSEF E. Using list-based simulated annealing and genetic algorithm for order batching and picker routing in put wall based picking systems[J]. Applied Soft Computing, 2019, 75: 106-119.
- [7] JIANG X, SUN L, ZHANG Y, et al. Order batching and sequencing for minimising the total order completion time in pick-and-sort warehouses[J]. Expert Systems with Applications, 2022, 187: 115943.
- [8] DARIA B, MARTINA C, ALESSANDRO P, et al. Fatigue and recovery: research opportunities in order picking systems[J]. IFAC-Papers on Line, 2017, 50(1): 6882-6887.
- [9] 张树柱, 邱兵兵, 山家骏, 等. 考虑驾驶员疲劳的车辆路径优化及算法研究[J]. 工业工程, 2023, 26(2): 132-140.  
ZHANG Shuzhu, QIU Bingbing, SHAN Jiajun, et al. Vehicle routing optimization and its algorithm considering driver fatigue[J]. Industrial Engineering Journal, 2023, 26(2): 132-140.
- [10] 赵小松, 赵舒萌, 刘娜, 等. 考虑疲劳和恢复的混流装配线平衡问题[J]. 系统工程学报, 2020, 35(2): 265-275.  
ZHAO Xiaosong, ZHAO Shumeng, LIU Na, et al. Balance problem of mixed flow assembly line considering fatigue and recovery[J]. Journal of Systems Engineering, 2020, 35(2): 265-275.
- [11] GAWRON V J, FRENCH J, FUNKE D. An overview of fatigue[M]//HANCOCK P A, DESMOND P A. Stress, workload, and fatigue. Boca Raton: CRC Press, 2001: 581-595.
- [12] 赵小松, 武锦慧, 何桢. 考虑疲劳和工作负荷的人工拣选货品排程研究[J]. 系统工程学报, 2015, 30(5): 703-710.  
ZHAO Xiaosong, WU Jinhui, HE Zhen. Research on the work-rest scheduling of the manual picking with workers' fatigue and workload[J]. Journal of Systems Engineering, 2015, 30(5): 703-710.
- [13] BATTINI D, GLOCK C H, GROSSE E H, et al. Human energy expenditure in order picking storage assignment: a bi-objective method[J]. Computers & Industrial Engineering, 2016, 94: 147-157.
- [14] PAPADIMITRIOU C H, KANELLAKIS P C. Flowshop scheduling with limited temporary storage[J]. Journal of Association for Computing Machinery, 1980, 27: 533-549.
- [15] VOUIROS G A, PAPADOPOULOS H T. Buffer allocation in unreliable production lines using a knowledge based system[J]. Computers and Operations Research, 1998, 25(12): 1055-1067.
- [16] 郭伏, 钱省三. 人因工程学[M]. 北京: 机械工业出版社, 2018.
- [17] WU H, WANG M. Determining the maximum acceptable work duration for high-intensity work[J]. European Journal of Applied Physiology, 2001, 85(3-4): 339-344.
- [18] FITZPATRICK P. The Effects of acute and chronic sodium bicarbonate supplementation on high-intensity intermittent performance, recovery and subsequent performance in rugby union players[D]. Dublin: School of Health and Human Performance Dublin City University, 2012: 33-34.
- [19] HENN S, WASCHER G. Tabu search heuristics for the order batching problem in manual order picking systems[J]. European Journal of Operational Research, 2012, 222(3): 484-494.
- [20] 石英, 王秀红, 祁丽霞, 等. 人因工程学[M]. 北京: 清华大学出版社, 2011: 103-104.

(责任编辑: 孟晓燕)