

doi:10.19920/j.cnki.jmsc.2025.01.004

面向双层分拣系统的成组分拣优化研究^①

镇 璐, 谭哲一, 高佳静

(上海大学管理学院, 上海 200444)

摘要:近年来随着电子商务的快速发展,包裹分拣订单呈现批量小、品种多、配送周期短、精度要求高等特征,这对分拣作业提出了更高的要求。为了缩短分拣时间、降低分拣成本、提高分拣效率,物流企业开始逐步使用“双层自动分拣系统”。为了进一步提升分拣效率,“成组分拣”策略近来被一些企业采用,用以提升交叉带托盘利用率,该策略是将分散的单件包裹按一定的规则组合在一起,成为一个规格化、标准化的大分拣单元再进行分拣。本文研究了考虑成组分拣的双层自动分拣系统中包裹、交叉带托盘、分拣格口、供包台等资源的分配优化问题,建立了整数规划模型,然后设计了一种高效的基于列生成的求解算法,并且提出了一些子问题加速策略以进一步缩短求解时间。实验结果证实了采用成组分拣方法与采用双层自动分拣系统可有效提高分拣效率;另外,基于所提模型和算法的敏感性分析实验,本研究也提出了针对自动分拣系统选型与设置的一些管理启示。

关键词: 双层分拣系统; 成组分拣; 分拣优化; 列生成算法; 整数规划

中图分类号: F213 **文献标识码:** A **文章编号:** 1007-9807(2025)01-0042-19

0 引言

我国是一个物流大国,2021年全社会物流总额接近335万亿;从与居民感受最密切的快递业来看,全年业务量达1 083亿件。面对如此巨大的物流服务需求量和市场,进一步提升我国物流效率对于实现全社会物流行业降本增效目标至关重要。物流配送中心作为物流网络中重要一环,它的运营效率直接影响整个物流系统乃至供应链系统的整体运营效率。配送中心的作业包括仓储、流通加工、分拣、运输等,其中分拣作业是配送中心中瓶颈作业环节之一^[1]。分拣作业是指从指定的存储位置取出特定数量、品项的货物以满足顾客订单的一系列流程^[2]。近年来,订单呈现小批量、多品种、频繁订货、配送周期短、精度要求高等特征,这对分拣作业提出了更高要求。因此,一些大型物

流企业开始逐步使用自动分拣系统用于提升分拣效率,缩短分拣时间,从而提升配送中心的运营效率。

面向物流配送中心的分拣系统研究近年来得到了物流领域学者的研究关注。从管理科学、运筹优化角度针对分拣系统开展的定量化研究还不是很多,一些已有研究都是针对一些特定的分拣系统。针对“单层分拣系统”Briskorn 和 Emde^[3]提出了考虑总分拣距离最小化的分拣调度问题,建立了数学模型,决策每个包裹分拣的交叉带托盘、供包台、分拣格口还有分拣顺序。在此基础上,Boysen 等^[4]对订单聚类进行了考虑,以最小化总分拣距离为目标,并且对同一客户订单的聚类进行了考虑。其后,Boysen 等^[5]又对分区集成仓储系统因素进行考虑,以分拣完成时间总和最小化为目标,对于订单的分拣顺序进行了决策,并使用精确

① 收稿日期: 2020-01-31; 修订日期: 2022-06-25。

基金项目: 国家自然科学基金资助项目(72025103; 72401179; 72394360; 72394362; 72361137001)。

作者简介: 镇 璐(1981—),男,湖北宜都人,教授,博士生导师。Email: lzhen@shu.edu.cn

式和启发式算法进行求解。“多层级分拣系统”适用于物流集散中心的超大批量分拣任务,该系统使用多台设备对包裹由城市到地区进行层层分拣,Novoa 等^[6]研究了该系统的目的地分配问题,以平衡各个分拣机的操作量为目标建立数学模型。“双层分拣系统”被开发用于提升“单层分拣系统”的分拣效率,区别于“单层分拣系统”,“双层分拣系统”需对于平衡上下层的分拣负荷进行考虑,Fedtke 和 Boysen^[7]以最小化上下层最大分拣距离为目标建立了求解包裹与供包台、传送带还有分拣格口分配关系的数学模型,并设计了模拟退火算法求解模型。Chen 和 Song^[8]研究了一种具有多个入站口和多个出站口的“并行分拣系统”,以最小化完工时间为目
标,构建了数学模型,并使用四种基于约翰逊规则的启发式算法进行求解。Gaast^[9]等研究了“基于混合设备的分拣系统”,并分析分拣设备混载对于自动分拣系统的影响,并通过使用聚类分析法以及矩阵几何法来研究系统的吞吐能力。基于以上关于各类不同类型的自动分拣系统的相关文献调研,发现有部分研究针对当前比较流行的“交叉带分拣系统”的调度问题进行研究;本研究针对交叉带分拣系统中最先进的模式——“双层交叉带自动分拣系统”的效率提升问题开展了定量化研究。

“分拣策略”对分拣系统的运营效率同样有重要的影响作用。因此,一些学者针对分拣策略进行研究。Hou^[10]等针对基于传送带的分拣系统研究了“订单到达均衡”策略,并提出了数学模型。王帅安^[11]针对 A 字分拣系统研究了“上货策略”,通过建立基于马尔可夫链的数学模型,提出了确定最优上货策略的方法。卢少平^[12]等研究了“并行分区拣选策略”,并分析了设备各闸门的动作时序对设备整体运行效率的影响,建立了品项分配聚类的数学模型,应用启发式聚类算法对模型求解。Zenker 和 Boysen^[13]对单层分拣设备研究了“暂存区策略”,并以每批次暂存区内的包裹数量最小化为目标,提出了一个两阶段算法。以上关于各类“分拣策略”研究的相关文献中,基本都是考虑一个托盘分拣一件包裹的情况,较少考虑“成组分拣”的情况。张源凯等^[14]对于订单拆分以及订单合并打包策略进行研究,并得出了订单

合并打包策略的经济适用范围。Gaast 等^[15]针对传送带和分拣区的拥堵情况进行研究,构建了一种多拥堵循环排队网络,并提出了一种基于区域跳跃的近似策略,利用该策略可有效避免拥堵发生。Zou 等^[16]建立了封闭的排队网络来评估系统的分拣性能,对于机器人在单层、双层布局中的拥堵情况进行研究,并利用德邦公司的数据进行实验,实验结果表示双层布局可有效缓解拥堵情况,同时证明了随机分配策略和最短队列分配策略对于具有大量机器人的仓库来说是有效的。Chen 等^[17]等对于“单层倾斜托盘分拣系统”的“捷径”策略进行研究,并对各种“捷径”的布局进行实验,从而得到能在最大程度上提升系统分拣效率的“捷径”策略。“成组分拣”是指一种将分散的单件包裹按一定的规则组合在一起,成为一个规格化、标准化的大分拣单元进行分拣的方法。本研究针对上述“成组分拣”策略开展了研究,以期提升交叉带托盘利用率,进而提升整个分拣系统的运营效率。

综上所述,本文研究了一个考虑成组分拣的双层自动分拣系统中包裹、交叉带托盘、分拣格口、供包台等资源的分配优化问题,提出相应的整数规划模型用于求解最优分拣调度方案。由于该问题为 NP-hard 问题,在短时间内 Cplex 较难解得问题的最优解,故针对该问题开发出一种基于列生成的算法用于求解该问题,并将该算法的结果与 Cplex、原问题模型的下界模型进行比较,从而对本研究算法的有效性进行了验证。

1 问题与模型

1.1 问题描述

本研究将当前主流的大型配送中心(比如 EMS)普遍使用的“双层交叉带自动分拣系统”作为研究对象,并引入“成组分拣”概念提升分拣效率,对该系统中的包裹、交叉带托盘、分拣格口、供包台等资源分配问题进行了研究。通过提出的整数规划模型优化面向“成组分拣”的分组决策、“成组分拣”后的包裹组与交叉带托盘的分配决策、供包台(入口)的选择决策和分拣格口(出口)的选择决策。

1.1.1 流程描述

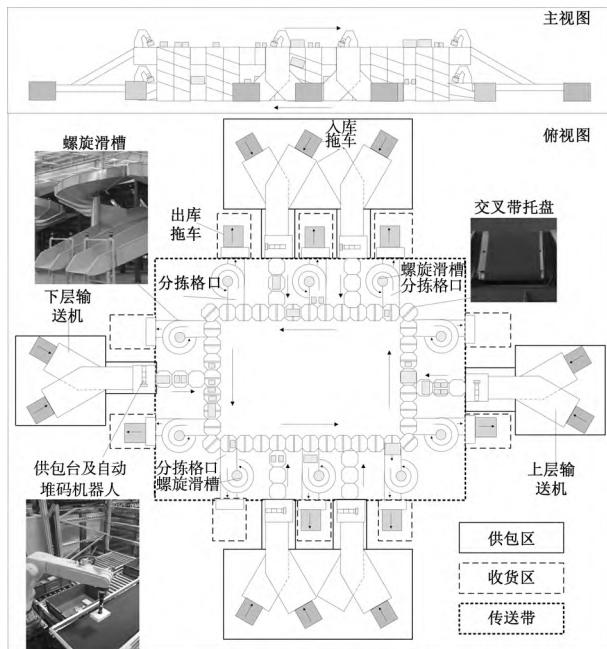


图 1 双层自动分拣系统中双层交叉带分拣系统主与俯视图

Fig. 1 Main and top view of double-layer-crossbelt sorting system in double-layer automatic sorting system

双层自动分拣系统一般分为三部分：供包区、传送带、收货区。图 1 为本文研究的双层自动分拣系统的主视图与俯视图。相较于单层自动分拣系统，该双层分拣系统具有效率更高、占地面积更小的特点。该分拣系统的上层传送带沿逆时针方向运行，下层传送带沿顺时针方向运行。包裹进入配送中心后，由入库拖车送至上层/下层输送机并传输至供包台，由自动堆码机器人装载至交叉带托盘上。当包裹由交叉带托盘输送至指定分拣格口时，交叉带托盘将包裹卸下并送入分拣格口，同一目的地的包裹必须送入同一分拣格口，进入分拣格口后包裹从螺旋滑槽进入物料箱，结束该件包裹的分拣。当物料箱内的包裹到达一定数量时，出库拖车将包裹装入出库货车进行配送。在下一个批次的包裹到达前，该分拣系统已完成本批次包裹的分拣。

1.1.2 “成组分拣”与分组模板

由于双层自动分拣系统的交叉带托盘面积固定，且一条传送带上的交叉带托盘数量固定，因此，本研究引入了“成组分拣”的概念，即：根据包裹的目的地信息、尺寸、重量等信息，将满足条件

的包裹由同一交叉带托盘进行分拣的方式。通过“成组分拣”能够提升交叉带托盘利用率，减少分拣时间，缩短分拣距离，进而提高分拣效率。采用“成组分拣”的包裹须满足以下条件：1)由同一交叉带托盘进行“成组分拣”的包裹目的地相同；2)这些包裹的总面积不能大于交叉带托盘的面积。由于包裹尺寸大多有所不同，这也提升了包裹“成组分拣”的决策难度。

为了简化问题并将其与实际情况相结合，本研究提出了分组模板的概念，即每个交叉带托盘上装载的每个规格的包裹数量必须等于预先设定的数值。当包裹信息被采集后，自动堆码机器人根据分组模板对包裹进行分组，分组后的包裹被送至供包台。关于单个交叉带托盘所能使用的分组模板举例如图 2 所示。为了方便获取包裹信息以用于“成组分拣”，需要将包裹的二维码朝上放置。因此，包裹间不能堆叠。在现实作业环境中，包裹通常根据规格分成以下四类：超大规格包裹，大规格包裹，中规格包裹和小规格包裹。其中超大规格包裹单独分拣，因此，本研究对大、中、小三种规格包裹进行研究。其中，三种规格包裹的最大尺寸分别为 $0.7 \text{ m} \times 0.7 \text{ m}$ 、 $0.6 \text{ m} \times 0.4 \text{ m}$ 和 $0.4 \text{ m} \times 0.2 \text{ m}$ 。交叉带托盘所能分拣的包裹的最大尺寸为 $0.8 \text{ m} \times 0.8 \text{ m}$ 。在每个分组模板中，可以放置的每个规格包裹的数量是固定的，此外，每个分拣模板可被多个托盘选择，每个托盘也必定会选择一个分拣模板。

虽然采用“成组分拣”可有效提升交叉带托盘利用率，但相较于单件分拣方式，“成组分拣”会在供包台处产生额外的包裹装载距离。由于自动堆码机器人一次只能将处于供包台的一件包裹放置在交叉带托盘上，且一个分组模板可能有多个包裹，若依照分组模板进行“成组分拣”则会产生大量包裹装载距离。若只对分拣距离进行考虑，则会过分采用“成组分拣”而忽视是否不采用“成组分拣”的可能性，因此，如何平衡装载距离与分拣距离的关系也是本研究的难点之一。为了尽可能提升双层自动分拣系统的分拣效率，本研究以双层分拣系统中的总分拣距离最小化为目标进行

优化。总分拣距离包括交叉带托盘分拣所有包裹产生的分拣距离以及包裹“成组分拣”时产生的装载距离。其中，包裹的装载距离为按照分组模板将处于供包台的包裹放置在交叉带托盘的距离总和。

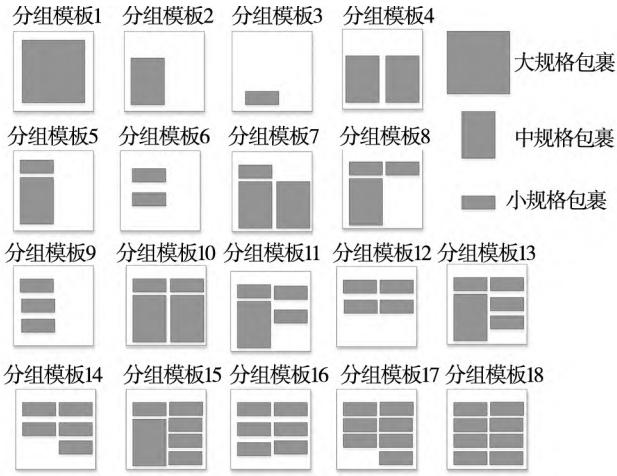


图2 单个交叉带托盘能使用的分组模板举例

Fig. 2 Examples of grouping templates with a single crossbelt tray

1.2 模型构建与符号说明

为了便于理解符号，本研究采用拉丁字母和希腊字母分别表示参数(索引，集合)和变量，模型中使用的参数和变量的符号如下所示。

1.2.1 索引和集合

D —目的地的集合,下标为 d ;

K —供包台的集合,下标为 k , 1 至 $|K|/2$ 为下层供包台, $|K|/2+1$ 至 $|K|$ 为上层供包台;

J —交叉带托盘的集合,下标为 j ;

I_d —目的地为 d 的包裹集合,下标为 i ;

I —包裹的集合, $I = \bigcup_{d \in D} I_d$;

S —包裹尺寸的集合,下标为 s ;

E —分拣格口的集合,下标为 e , 当 $e=0$ 时表示交叉带托盘不启用;

G —分组模板的集合,下标为 g , 当 $g=0$ 时表示交叉带托盘不启用,未对包裹进行分拣。

1.2.2 参数

$t_{e,k}$ —包裹由供包台 k 供包并由分拣格口 e 出库所产生的分拣距离,分拣距离由包裹进入传送带前的距离、传送带传送距离和包裹离开传送带后的距离组成;

h_g —使用分组模板 g 所产生的装载距离;

$f_{i,s}$ —若包裹 i 的尺寸为 s , 则为 1; 否则为 0;

$n_{s,d}$ —目的地为 d 尺寸为 s 的包裹数量;

$l_{g,s}$ —在分组模板 g 中, 尺寸为 s 的包裹数量。

1.2.3 变量

$\alpha_{j,g}^d$ —0-1 决策变量, 若交叉带托盘 j 采用分组模板 g 分拣目的地为 d 的包裹, 则为 1; 否则为 0;

$\beta_{j,k,e}^d$ —0-1 决策变量, 若交叉带托盘 j 由供包台 k 供包并由分拣格口 e 对目的地为 d 的包裹进行分拣则为 1; 否则为 0;

$\gamma_{i,j}^d$ —0-1 决策变量, 若目的地为 d 的包裹 i 由交叉带托盘 j 进行分拣, 则为 1; 否则为 0;

$\delta_{d,e}$ —0-1 决策变量, 目的地为 d 的包裹由分拣格口 e 出库, 则为 1; 否则为 0;

$\theta_{j,d}$ —0-1 决策变量, 目的地为 d 的包裹由交叉带托盘 j 进行分拣则为 1; 否则为 0。

1.2.4 数学模型

$$\min \sum_{d \in D} \sum_{j \in J} \left(\sum_{k \in K} \sum_{e \in E} t_{e,k} \beta_{j,k,e}^d + \sum_{g \in G} h_g \alpha_{j,g}^d \right) \quad (1)$$

$$\text{s. t. } \sum_{d \in D} \sum_{g \in G \cup \{0\}} \alpha_{j,g}^d = 1 \quad \forall j \in J \quad (2)$$

$$\sum_{d \in D} \sum_{e \in E \cup \{0\}} \sum_{k \in K} \beta_{j,k,e}^d = 1 \quad \forall j \in J \quad (3)$$

$$\sum_{j \in J} \gamma_{i,j}^d = 1 \quad \forall d \in D, i \in I_d \quad (4)$$

$$\sum_{e \in E} \delta_{d,e} = 1 \quad \forall d \in D \quad (5)$$

$$\sum_{d \in D} \delta_{d,e} \leq 1 \quad \forall e \in E \quad (6)$$

$$\theta_{j,d} = \sum_{e \in E} \sum_{k \in K} \beta_{j,k,e}^d \quad \forall d \in D, j \in J \quad (7)$$

$$\sum_{j \in J} \sum_{g \in G \cup \{0\}} l_{g,s} \alpha_{j,g}^d = n_{s,d} \quad \forall d \in D, s \in S \quad (8)$$

$$\sum_{j \in J} \sum_{k \in K} \beta_{j,k,e}^d \leq |J| \delta_{d,e} \quad \forall d \in D, e \in E \quad (9)$$

$$\gamma_{i,j}^d \leq \theta_{j,d} \quad \forall i \in I_d, j \in J, d \in D \quad (10)$$

$$\alpha_{j,g}^d \leq \theta_{j,d} \quad \forall d \in D, j \in J, g \in G \quad (11)$$

$$\sum_{i \in I_d} f_{i,s} \gamma_{i,j}^d \leq l_{g,s} \alpha_{j,g}^d + |I|(1 - \alpha_{j,g}^d) \quad (12)$$

$$\forall d \in D, j \in J, s \in S, g \in G \cup \{0\}$$

$$\sum_{i \in I_d} f_{i,s} \gamma_{i,j}^d + |I|(1 - \alpha_{j,g}^d) \geq l_{g,s} \alpha_{j,g}^d \quad (13)$$

$$1 - \sum_{k \in K} \beta_{j,k,0}^d \geq \gamma_{i,j}^d \quad \forall d \in D, i \in I_d, j \in J \quad (14)$$

$$\sum_{k \in K} \beta_{j,k,0}^d = \alpha_{j,0}^d \quad \forall d \in D, j \in J \quad (15)$$

$$\delta_{d,e} \in \{0,1\} \quad \forall d \in D, e \in E \quad (16)$$

$$\alpha_{j,g}^d \in \{0,1\} \quad \forall d \in D, j \in J, g \in G \cup \{0\} \quad (17)$$

$$\beta_{j,k,e}^d \in \{0,1\} \quad \forall j \in J, e \in E \cup \{0\}, \\ k \in K, d \in D \quad (18)$$

$$\gamma_{i,j}^d \in \{0,1\} \quad \forall d \in D, i \in I, j \in J \quad (19)$$

$$\theta_{j,d} \in \{0,1\} \quad \forall j \in J, d \in D \quad (20)$$

目标函数(1)为最小化双层分拣系统中的总分拣距离,由两部分构成:在分拣环节,每个交叉带托盘分拣包裹所产生的分拣距离;采用分组模板进行“成组分拣”产生的包裹装载距离。约束(2)确保每个交叉带托盘只能选择一个分组模板,且只能选择一个目的地的包裹进行分拣。约束(3)确保交叉带托盘只能由一个供包台进行供包,并且只能选择一个分拣格口来卸载其分拣的所有包裹。约束(4)保证每件包裹只能由负责运输与该包裹目的地相同包裹的交叉带托盘进行分拣。约束(5)确保每个目的地的包裹只能由一个分拣格口分拣。约束(6)表示一个分拣格口至多只能分拣一个目的地的包裹。约束(7)表示决策变量 $\theta_{j,d}$ 与 $\beta_{j,k,e}^d$ 之间的关系。约束(8)表示每件包裹都会被一个交叉带托盘进行分拣。约束(9)表示若分拣格口 e 未被分配给目的地 d ,则所有与目的地 d 对应的交叉带托盘不能将包裹卸在分拣格口 e 。约束(10)表示若交叉带托盘 j 分拣包裹 i ,包裹 i 的目的地与交叉带托盘 j 所对应的目的地必须相同;否则,无需相同。约束(11)表示决策变量 $\alpha_{j,g}^d$ 与 $\theta_{j,d}$ 之间的关系。约束(12)与约束(13)表示决策变量 $\alpha_{j,g}^d$ 与 $\gamma_{i,j}^d$ 之间的关系。约束(14)和约束(15)表示只有处于可用状态的交叉带托盘才能分拣包裹。约束(16)至约束(20)定义了决策变量的取值范围。

1.2.5 建模思路

本文所研究的双层分拣系统优化问题的建模思路图如图 3 所示。图 3 中有编号分别为 $i_1 \dots i_n$ 的 n 件包裹需进行分拣,本研究模型的目标函数为最小化总分拣距离,由分拣包裹所产生的分拣距离以及“成组分拣”产生的包裹装载距离组成。

1) 装载距离:影响包裹装载距离的主要决策变量为 $\alpha_{j,g}^d$ 。分组模板与托盘的匹配关系 $\alpha_{j,g}^d$ 由交叉带托盘与包裹的匹配关系 $\gamma_{i_1,j}^d \dots \gamma_{i_n,j}^d$ 以及约束(12)至约束(13)进行决策。 $\gamma_{i_1,j}^d \dots \gamma_{i_n,j}^d$ 则由包裹 $i_1 \dots i_n$ 的尺寸信息 $f_{i_1,s} \dots f_{i_n,s}$ 、分组模板中所能

存放的各个尺寸包裹数量 $l_{g,s}$ 进行决策。在确定分组模板与托盘进行匹配关系后即可根据参数 h_g 与决策变量 $\alpha_{j,g}^d$ 得到由于“成组分拣”产生的包裹装载距离。

2) 分拣距离:影响包裹分拣距离的主要决策变量为 $\beta_{j,k,e}^d$ 。交叉带托盘与供包台的匹配关系 $\beta_{j,k,e}^d$ 由交叉带托盘与目的地的对应关系 $\theta_{j,d}$ 、分组模板与托盘进行匹配关系 $\alpha_{j,g}^d$ 、分拣格口与目的地的匹配关系 $\delta_{d,e}$ 、交叉带托盘与包裹的匹配关系 $\gamma_{i,j}^d$ 以及约束(7)、约束(9)、约束(14)、约束(15)进行决策。 $\delta_{d,e}$ 由约束(5)与约束(6)进行决策。 $\theta_{j,d}$ 由 $\alpha_{j,g}^d$ 以及约束(10)至约束(11)对进行决策。根据参数 $t_{e,k}$ 和决策变量 $\beta_{j,k,e}^d$ 确定包裹分拣距离,当 $1 \leq k \leq |K| / 2$ 时, k 表示为下层供包台, $|K| / 2 + 1 \leq k \leq |K|$ 时, k 表示为上层供包台。

本研究所提出的决策模型通过最小化双层分拣系统的总分拣距离,优化了包裹、交叉带托盘、分拣格口、供包台等资源的分配决策,从而提升分拣效率。本研究设计了一种基于列生成的求解算法对于模型进行高效求解。

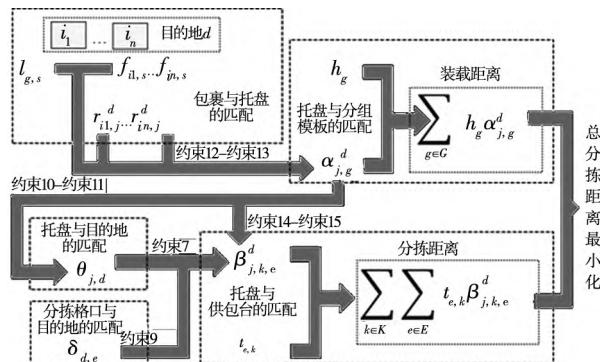


图 3 建模思路

Fig. 3 Modeling ideas

2 基于列生成的算法设计

2.1 列生成算法设计

对于求解规模较大的 0-1 规划问题,一些商业求解器(如 Cplex)很难对其进行求解。而列生成算法可有效地对 0-1 规划问题进行求解^[18],不仅可以提高求解速度,而且求解结果准确性较高,本章设计了列生成算法对模型进行求解。本研究使用的列生成算法的流程如图 4 所示。

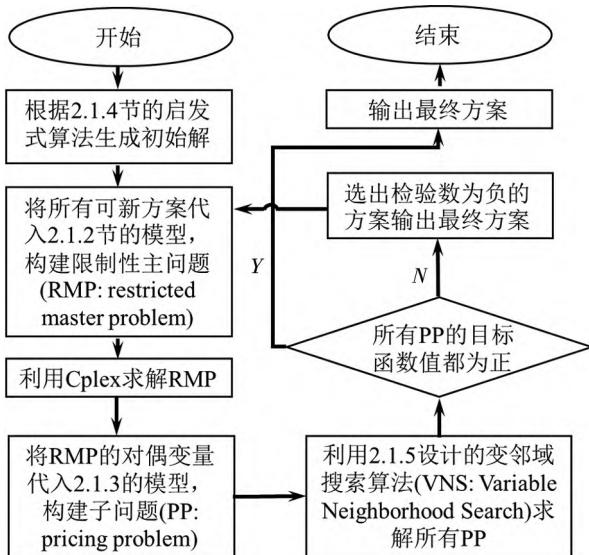


图4 列生成算法流程图

Fig. 4 Flow chart of column generation algorithm

2.1.1 集合覆盖模型

在本节中, 使用 Dantzig-Wolfe 分解将该问题构建成一个集合覆盖模型^[19, 20]. 令 p_d 为给定范围内目的地 d 的所有可能的分拣调度方案. 目的地 d 的每个分拣调度方案 p_d 都会包含以下信息: 1) 该目的地的包裹由哪几个交叉带托盘进行分拣; 2) 运送该目的地包裹的交叉带托盘由哪个供包台供包, 并将包裹送往哪个分拣格口. 令所有分拣调度方案的集合为 $P = \cup_{d \in D} p_d$. 在目的地 d 的每个分拣调度方案 p_d 中, 存在以下已知参数

x_{j,g,p_d} — 在分拣调度方案 p_d 中, 若交叉带托盘 j 使用分组模板 g 进行分拣, 则为 1, 否则为 0;

y_{j,k,e,p_d} — 在分拣调度方案 p_d 中, 若交叉带托盘 j 由供包台 k 进行供包将所运所有包裹送往分拣格口 e 进行分拣, 则为 1, 否则 0;

w_{i,j,p_d} — 在分拣调度方案 p_d 中, 该方案对应的包裹 i 由交叉带托盘 j 进行分拣, 则为 1, 否则 0;

z_{e,p_d} — 在分拣调度方案 p_d 中, 该方案对应的所有包裹送往分拣格口 e 进行分拣, 则为 1, 否则 0.

令 c_{p_d} 为启用分拣调度方案 p_d 所产生的总分拣距离, c_{p_d} 的计算公式如约束(28)所示. 定义决策变量 λ_{p_d} , 当目的地 d 采用分拣调度方案 $p_d \in P$ 时则为 1, 否则为 0. 根据以上参数以及决策变量建立集合覆盖模型

$$\min \sum_{d \in D} \sum_{p_d \in P_d} c_{p_d} \lambda_{p_d} \quad (21)$$

$$\text{s. t. } \sum_{d \in D} \sum_{p_d \in P_d} x_{j,g,p_d} \lambda_{p_d} \leq 1 \quad \forall j \in J, g \in G \quad (22)$$

$$\sum_{d \in D} \sum_{p_d \in P_d} y_{j,k,e,p_d} \lambda_{p_d} \leq 1, \quad \forall j \in J, k \in K, e \in E \quad (23)$$

$$\sum_{d \in D} \sum_{p_d \in P_d} w_{i,j,p_d} \lambda_{p_d} \leq 1 \quad \forall j \in J, i \in I \quad (24)$$

$$\sum_{d \in D} \sum_{p_d \in P_d} z_{e,p_d} \lambda_{p_d} \leq 1 \quad \forall e \in E \quad (25)$$

$$\sum_{p_d \in P_d} \lambda_{p_d} = 1 \quad \forall d \in D \quad (26)$$

$$\lambda_{p_d} \in \{0, 1\} \quad \forall p_d \in P_d, d \in D \quad (27)$$

$$c_{p_d} = \sum_{j \in J} \left(\sum_{k \in K} \sum_{e \in E} t_{e,k} y_{j,k,e,p_d} + \sum_{g \in G} h_g x_{j,g,p_d} \right)$$

$$\forall p_d \in P_d, d \in D \quad (28)$$

在该模型中, 目标函数(21)为最小化总分拣距离. 约束(22)表示在交叉带托盘 j 使用分组模板 g 进行分拣的情况至多只能在一个计划中出现. 约束(23)表示交叉带托盘 j 由供包台 k 进行供包将所运所有包裹送往分拣格口 e 进行分拣的情况至多只能发生一次. 约束(24)表示包裹 i 由交叉带托盘 j 进行分拣的情况至多只能出现一次. 约束(25)表示一个分拣格口至多只能分配给一个目的地. 约束(26)表示一个目的地只能分配一个分拣调度方案. 约束(27)定义了决策变量的取值范围. 约束(28)计算了启用分拣调度方案 p_d 所产生的总分拣距离.

2.1.2 限制性主问题的构建

由于在主问题(master problem, MP)中包含了目的地的所有可能的分拣调度方案, 而求解 MP 所需的计算时间会随着分拣调度方案的数量的增大而呈指数级增长, 因此无法直接对 MP 进行求解. 为了便于求解, 首先, 通过对 MP 的决策变量 λ_{p_d} 进行线性松弛, 构造线性主问题(linear master problem, LMP), 即将约束(27)替换为

$$\lambda_{p_d} \geq 0, \quad \forall p_d \in P_d, d \in D \quad (29)$$

在线性化后, LMP 中分拣调度方案的数量仍然非常庞大. 因此, 本研究在求解 LMP 时仅考虑部分分拣调度方案 $P' = \cup_{d \in D} p'_d \subseteq P$, 构造更加易于求解的限制性主问题(restricted master problem, RMP). RMP 的目标函数为

$$\min \sum_{p_d \in P'} c_{p_d} \lambda_{p_d} \quad (30)$$

RMP 的约束为约束(22)至约束(26)及约束(29).

在算法的每次迭代过程中, RMP 的对偶变量被当作输入参数加入到子问题中去, 用于生成新的可行分拣调度. 这些对偶变量定义如下

$\dot{\pi}_{j,g}$ —与约束(22)对应的对偶变量, $\forall j \in J, g \in G$;

$\ddot{\pi}_{j,k,e}$ —与约束(23)对应的对偶变量, $\forall j \in J, k \in K, e \in E$;

$\ddot{\pi}_{i,j}$ —与约束(24)对应的对偶变量, $\forall j \in J, i \in I$;

μ_e —与约束(25)对应的对偶变量, $\forall e \in E$;

ρ_d —与约束(26)对应的对偶变量, $\forall d \in D$.

2.1.3 子问题的构建

构建子问题(pricing problem, PP)的目的是获得一个检验数(reduced cost)为负的可行分拣调度方案. 在算法的每次迭代过程中, 需要对 $|D|$ 个PP进行求解, 每个PP对应于一个目的地 d , 并且为每个目的地生成一个可行的分拣调度方案. 对于通过求解子问题生成的 $|D|$ 个可行分配方案, 只有具有检验数为负的可行分拣调度方案才可以添加到RMP中进行下一轮的循环过程, 这意味着在算法的每次迭代过程中, 最多有 $|D|$ 个可行分配方案被加入RMP中. 以下为每个目的地对应的PP的数学模型.

1) 参数

n_s —尺寸为 s 的包裹数量;

$l_{g,s}$ —在分组模板 g 中, 尺寸为 s 的包裹数量;

$f_{i,s}$ —若包裹 i 的尺寸为 s , 则为1; 否则为0;

$\dot{\pi}_{j,g}, \ddot{\pi}_{j,k,e}, \ddot{\pi}_{i,j}, \mu_e, \rho_d$ 从RMP中获得的对偶变量.

2) 变量

$x_{j,g}$ —0-1决策变量, 在分拣计划 p_d 中, 若交叉带托盘 j 采用分组模板 g 分拣, 则为1; 否则为0;

$y_{j,k,e}$ —0-1决策变量, 在分拣计划 p_d 中, 若交叉带托盘 j 由供包台 k 供包从分拣格口 e 出库则为1; 否则为0;

$w_{i,j}$ —0-1决策变量, 在分拣计划 p_d 中, 若包裹 i 由交叉带托盘 j 进行分拣, 则为1; 否则为0;

z_e —0-1决策变量, 在分拣计划 p_d 中, 目的地为 d 的包裹从分拣格口 e 出库, 则为1; 否则为0.

3) 子问题模型

$$\min c_{p_d} - \sum_{g \in G} \sum_{j \in J} \dot{\pi}_{j,g} x_{j,g} - \sum_{j \in J} \sum_{k \in K} \sum_{e \in E} \ddot{\pi}_{j,k,e} y_{j,k,e} - \sum_{i \in I} \sum_{j \in J} \ddot{\pi}_{i,j} w_{i,j} - \sum_{e \in E} \mu_e z_e - \rho_d \quad (31)$$

$$\text{s.t. } \sum_{g \in G \cup \{0\}} x_{j,g} = 1 \quad \forall j \in J \quad (32)$$

$$\sum_{e \in E \cup \{0\}} \sum_{k \in K} y_{j,k,e} = 1 \quad \forall j \in J \quad (33)$$

$$\sum_{j \in J} w_{i,j} = 1 \quad \forall i \in I_d \quad (34)$$

$$\sum_{e \in E} z_e = 1 \quad (35)$$

$$w_{i,j} \leq \sum_{e \in E} \sum_{k \in K} y_{j,k,e} \quad \forall i \in I_d, j \in J \quad (36)$$

$$x_{j,g} \leq \sum_{e \in E} \sum_{k \in K} y_{j,k,e} \quad \forall j \in J, g \in G \quad (37)$$

$$\sum_{j \in J} \sum_{k \in K} y_{j,k,e} \leq |J| z_e \quad \forall e \in E \quad (38)$$

$$\sum_{i \in I_d} f_{i,s} w_{i,j} \leq l_{g,s} x_{j,g} + |I|(1 - x_{j,g}) \quad \forall j \in J, s \in S, g \in G \cup \{0\} \quad (39)$$

$$\sum_{i \in I_d} f_{i,s} w_{i,j} + |I|(1 - x_{j,g}) \geq l_{g,s} x_{j,g} \quad \forall j \in J, s \in S, g \in G \cup \{0\} \quad (40)$$

$$1 - \sum_{k \in K} y_{j,k,0} \geq w_{i,j} \quad \forall i \in I_d, j \in J \quad (41)$$

$$\sum_{k \in K} y_{j,k,0} = x_{j,0} \quad \forall j \in J \quad (42)$$

$$\sum_{d \in D} \sum_{j \in J} \sum_{p \in P \cup \{0\}} l_{g,s} x_{j,g} = n_s \quad \forall s \in S \quad (43)$$

$$x_{j,g} \in \{0,1\} \quad \forall j \in J, g \in G \quad (44)$$

$$y_{j,k,e} \in \{0,1\} \quad \forall j \in J, e \in E, k \in K \quad (45)$$

$$w_{i,j} \in \{0,1\} \quad \forall i \in I_d, j \in J \quad (46)$$

$$z_e \in \{0,1\} \quad \forall e \in E \quad (47)$$

$$c_{p_d} = \sum_{j \in J} \left(\sum_{k \in K} \sum_{e \in E} t_{e,k} y_{j,k,e} + \sum_{g \in G} h_g x_{j,g} \right) \quad \forall p_d \in p_d, d \in D \quad (48)$$

一旦分拣计划 p_d 的检验数为负, 则该分拣计划就会作为新增分拣计划代入RMP进行下一轮的求解. 其中包括决策变量 $x_{j,g}, y_{j,k,e}, w_{i,j}, z_e$ 以及采用分拣计划 p_d 所产生的分拣距离 c_{p_d} 都将作为已知参数代入RMP.

目标函数(31)为最小化检验数. 约束(32)确保每个交叉带托盘只能选择一个分组模板. 约束(33)表示若交叉带托盘可用, 它只能由一个供

包台进行供包,并且只能选择一个分拣格口来卸载其所有运输的包裹。约束(34)保证每件包裹只能由一个交叉带托盘进行分拣。约束(35)确保在分拣计划 p_d 中一个目的地的包裹由一个分拣格口进行分拣。约束(36)表示若交叉带托盘 j 负责对包裹 i 进行分拣,则交叉带托盘 j 必须启用。约束(37)表示决策变量 $x_{j,g}$ 与 $y_{j,k,e}$ 之间的关系。约束(38)表示分拣计划 p_d 中若分拣格口 e 未被分配给目的地 d ,则所有与目的地 d 对应的交叉带托盘不能将包裹卸在分拣格口 e 。约束(39)与约束(40)表示决策变量 $w_{i,j}$ 与 $x_{j,g}$ 之间的关系。约束(41)和约束(42)表示只有处于可用状态的交叉带托盘才能分拣包裹。约束(43)表示每件包裹都会被一个交叉带托盘进行分拣。约束(44)至约束(47)定义了决策变量的取值范围。约束(48)计算了启用分拣调度方案 p_d 所产生的总分拣距离。

2.1.4 初始解的生成

列生成的 RMP 的启动需要一组初始可行的分拣调度方案作为输入参数,因此本研究设计了一种启发式算法用于初始解的生成。

步骤1 将所有变量设置为 0,根据每个目的地包裹的数量对分拣格口进行分配,包裹数量越多,则将分拣距离越短的供包台与分拣格口的组合分配给该目的地,再根据分配情况对 z_{e,p_d} 进行赋值,确保每个目的地分配一个分拣格口。

步骤2 将每个目的地的包裹按照尺寸进行划分,同一规格的包裹尽可能由同一交叉带托盘进行分拣。大规格包裹只能单件进行分拣,故每件大规格包裹都由分组模板 1 进行分拣,中规格和小规格包裹尽可能按照分组模板 4 以及分组模板 18 进行分拣。若所有包裹都可由上述两个分组模板分配交叉带托盘则对 x_{j,g,p_d}, w_{i,j,p_d} 进行赋值;若存在不满足上述两个分组模板的包裹,则启用其他分组模板进行分拣再对 x_{j,g,p_d}, w_{i,j,p_d} 进行赋值。通过该步骤保证每件包裹都由一个交叉带托盘进行分拣。

步骤3 根据交叉带托盘是否启用情况以及供包台与分拣格口的组合的分配情况对 y_{j,k,e,p_d} 进行赋值。

2.1.5 子问题的求解

PP 的目标是找到需要添加到 RMP 中的检验

数(reduced cost)为负的列。由 2.1.3 可知,在每次 CG 迭代过程中都会生成 $|D|$ 个 PP,并对这些 PP 进行求解。若采用一些商业求解器对 PP 进行求解,这一过程耗费相当长的一段时间。为了进一步提高 CG 的求解效率,本研究提出了一些加速策略。

1) 采用变邻域搜索算法(variable neighborhood search, VNS)对于 PP 进行求解。该算法是一种全局逐步寻优算法,算法的邻域结构不依照固定模式进行搜索,能更有效地跳出局部最优,提升解的质量^[21, 22]。该算法主要思路如下:取得一个初始解,通过变邻域搜索策略进行优化,解得局部最优解并储存,然后通过对于已求得的最优解进行振动,并将振动后所得的解作为初始解,以避免在进一步搜索时再次搜索到该局部最优解。在完成所有邻域搜索策略后,需对算法迭代次数以及停止准则进行判断,若满足停止准则或超出最大迭代次数则算法停止并输出求得的局部最优解;否则继续执行邻域搜索策略。其中,停止准则为:若局部最优解在 n 次迭代中未发生变化则停止;振动过程表示为:若对局部最优解进行更新,则随机取得一个浮点值,若该浮点值大于振动过程的阈值,则将该局部最优解作为初始解,否则不更新初始解。

2) 每次迭代生成多个可行的计划,将目标函数值小于 0 的所有计划加入到 RMP 中。此操作可以快速进行优化。

3) 设计了 Cplex 可行化策略和规则可行化策略。实验结果显示规则可行化策略和 Cplex 可行化策略得到的目标值是一致的,但是规则可行化的运行时间明显要短。

① 初始解的确定

在每次迭代流程中,RMP 会对每个计划对应的决策变量 λ_{p_d} 进行计算。在本研究所设计的算法中,会根据决策变量 λ_{p_d} 的取值大小对于以往生成的计划进行选择,针对于某一目的地的子问题则会选取隶属于该目的地中 λ_{p_d} 取值最大的计划,将该计划设置为本次迭代流程中的初始解。

② 邻域的选择

插入:主要用于对 $y_{j,k,e}$ 、 $w_{i,j}$ 进行插入操作。例如:当对 $w_{i,j}$ 进行插入时,选择包裹 i 与交叉带

托盘 j , 执行③的可行化策略, 若存在分组模板 g 可使包裹 i 由交叉带托盘 j 进行运输, 若可行则对检验数进行计算, 若无法满足约束则继续选择其他包裹. 若该计划所得的检验数为负, 且小于已求得的最优检验数时, 将最优计划替换为该计划.

交换:主要用于对 $w_{i,j}$ 、 z_e 进行交换操作. 例如: 当对 $w_{i,j}$ 进行交换时, 选择包裹 i_1 与 i_2 , 原先两件包裹分别由交叉带托盘 j_1 与 j_2 进行运输, 若存在分组模板 g_1 与 g_2 可使两件包裹由对方所属的交叉带托盘进行运输, 执行③的可行化策略, 若可行则对检验数进行计算, 若无法满足约束则继续选择其他包裹. 若该计划所得的检验数为负, 且小于已求得的最优检验数时, 将最优计划替换为该计划.

重置:若在 n 次迭代流程后, 所得的最优解未进行变化, 则重新对包裹与交叉带托盘间的关系进行匹配. 匹配规则为优先将为正的对偶变量 $\pi_{j,g}^d, \pi_{j,k,e}^d, \pi_{i,j}^d, \mu_e, \rho_d$ 所对应的决策变量进行赋值, 即将 $y_{j,k,e}$ 、 $w_{i,j}$ 、 z_e 由 0 变为 1, 执行③的可行化策略, 若可行对检验数进行计算, 若出现不满足约束的情况则优先满足所对应的对偶变量较大的决策变量. 再根据 $w_{i,j}$ 对于 $x_{j,g}$ 进行赋值.

③可行化策略

在通过 VNS 求得候选解后, 需对候选解进行可行化再进行判优, 本研究的可行化方式共两种, 使用 Cplex 进行可行化以及通过规则进行可行化.

Cplex 可行化策略: 将由 VNS 所得候选解中的决策变量 $\alpha_{j,g}^d, \gamma_{i,j}^d, \beta_{j,k,e}^d, \theta_{j,d}, \delta_{d,e}$ Cplex 中进行求解, 若满足 Cplex 的全部约束则输出所的候选解的检验数; 若不满足约束则代入部分变量 $\theta_{j,d}, \delta_{d,e}$, 并求解剩余变量 $\alpha_{j,g}^d, \gamma_{i,j}^d, \beta_{j,k,e}^d$ 以及该可行解的检验数.

规则可行化策略: 将决策变量 $\theta_{j,d}, \gamma_{i,j}^d$ 代入约束(4)、约束(10)~约束(13)判断是否一个包裹只由一个交叉带托盘进行运输以及该交叉带托盘是否存在一个可行的分组模板, 若可行则可行令可行分组模板为 g_a , 令 $\alpha_{j,g_a}^d = 1$ 且 $\alpha_{j,g_a}^d = 0$, $\forall g \in G, g \neq g_a$, 若不可行则判断不可行原因, 若一件包裹由多个交叉带托盘运输, 则将该件包裹

分配给当前邻域对应的交叉带托盘 j 进行运输; 若交叉带托盘不存在一个可行的分组模板, 则将当前邻域中的包裹放回对应的交叉带托盘. 此时, 满足约束(2)、约束(4)、约束(8)、约束(10)~约束(13). 再将决策变量 $\delta_{d,e}$ 代入约束(5)和约束(6), 若不可行则将目的地 d 分配给当前邻域对应的分拣格口 e_a , 令 $\delta_{d,e_a} = 1$ 且 $\delta_{d,e} = 0, \forall e \in E, e \neq e_a$, 此时满足约束(5)与(6). 再将决策变量 $\theta_{j,d}, \delta_{d,e}$ 代入约束(7)、约束(9)、约束(14)、约束(15)求得 $\beta_{j,k,e}^d$ 的取值, 再将 $\beta_{j,k,e}^d$ 代入约束(3), 若不满足以上约束则令当前邻域对应的供包台为 k_a , 令 $\beta_{j,k_a,e}^d = 1$ 且 $\beta_{j,k,e}^d = 0, \forall k \in K, k \neq k_a$. 此时满足约束(3)、约束(7)、约束(9)、约束(14)、约束(15).

2.2 基于列生成解的启发式算法设计

本研究第 2.1 节中所提出的列生成算法仅能求解 RMP, 并不能保证找到可行的整数解. 因此, 本研究提出了一种基于列生成解的启发式算法, 使用不同的策略对分拣调度方案进行选择, 从而得出一个可行的近似较优解. 该启发式算法的流程如图 5 所示.

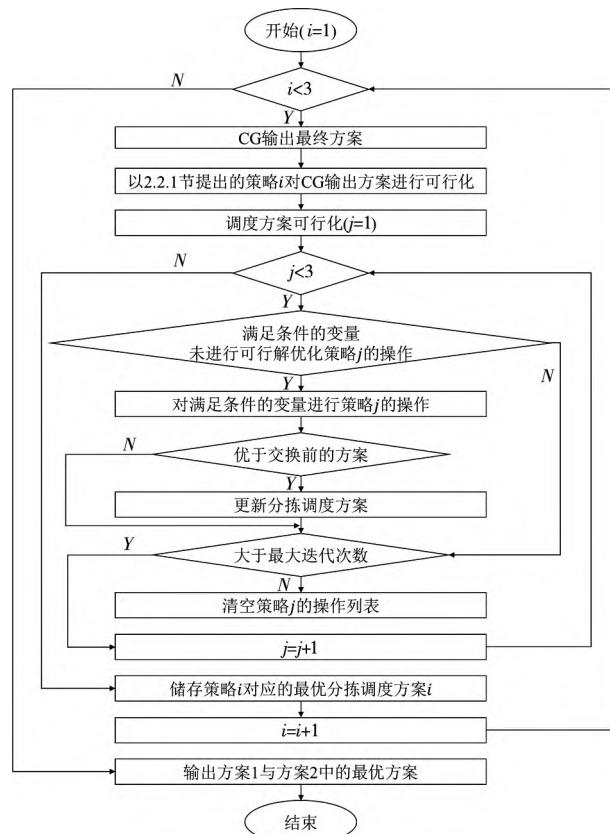


图 5 基于列生成解的启发式算法流程

Fig. 5 Heuristic algorithm based on column generation's solution

2.2.1 可行解选择策略规划

在该启发式算法的第二步,得到了许多分拣调度方案.本研究提出两种启发式策略来从众多分拣调度方案中选择最合适的方法.

策略1 由于分拣调度方案对应的决策变量 λ_{p_d} 的值越大,则该方案对于某一目的地而言是全局最优解一部分的可能性越大.因此,该策略从每个目的地对应的所有可行分拣调度方案中选择决策变量值 λ_{p_d} 最大的那个分拣调度方案,将该方案作为相应目的地的分拣调度方案.若存在两个决策变量值 λ_{p_d} 相同的分拣调度方案,则选择分拣距离 c_{p_d} 较短的方案.

策略2 由于目标函数为分拣距离最小化.因此,该策略从候选的可行分拣调度方案中选择产生的分拣距离 c_{p_d} 较短的方案,将该方案作为相应目的地的分拣调度方案.如果有两个分拣调度方案产生的分拣距离 c_{p_d} 相同,则选择决策变量 λ_{p_d} 较大的分拣调度方案.

2.2.2 可行解的优化策略规划

在上节中,根据约束对所得解进行了可行化.在确保所得解为可行的情况下,该可行解并非一个较优的局部最优解,因此在本节中设计了两个可行解的优化策略.

1) 变异优化策略

该策略主要侧重于对决策变量: $\beta_{j,k,e}^d, \gamma_{i,j}^d, \delta_{d,e}$ 进行优化.该策略主要思想为对一组变量中的某个变量进行变异操作.以决策变量 $\gamma_{i,j}^d$ 为例,在一个可行解中,包裹*i*由交叉带托盘*j₁*进行分拣,即 $\gamma_{i,j_1}^d = 1, \gamma_{i,j_2}^d = 0$.此时,将包裹*i*的分拣任务分配给交叉带托盘*j₂*,即 $\gamma_{i,j_1}^d = 0, \gamma_{i,j_2}^d = 1$.该策略称之为变异优化策略.该流程如图6所示.

2) 交换优化策略

该策略主要侧重于对决策变量: $\gamma_{i,j}^d, \delta_{d,e}$ 进行优化,该策略主要思想为对两组变量进行交换操作.以决策变量 $\gamma_{i,j}^d$ 为例,在一个可行解中,包裹*i₁*由交叉带托盘*j₁*进行分拣,包裹*i₂*由交叉带托盘*j₂*进行分拣,即 $\gamma_{i_1,j_1}^d = 1, \gamma_{i_1,j_2}^d = 0, \gamma_{i_2,j_2}^d = 1, \gamma_{i_2,j_1}^d = 0$.此时,将两个包裹对应的交叉带托盘进

行交换,即 $\gamma_{i_1,j_1}^d = 0, \gamma_{i_1,j_2}^d = 1, \gamma_{i_2,j_2}^d = 0, \gamma_{i_2,j_1}^d = 1$.

1. 该策略称之为交换优化策略.

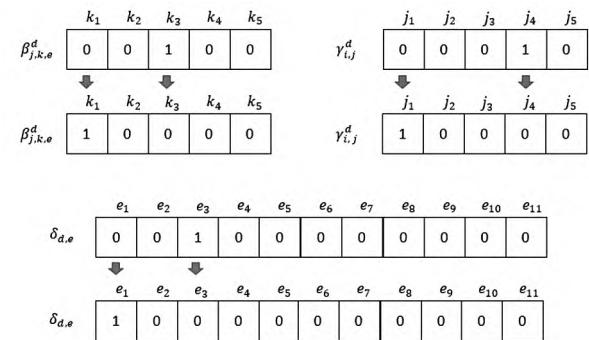


图6 变异优化策略

Fig. 6 Mutation optimization strategy

对于决策变量 $\gamma_{i,j}^d$ 则是对分拣包裹*i*与*i₁*对应的交叉带托盘进行交换,如图7所示.

对于决策变量 $\delta_{d,e}$ 的调整策略则是交换两个目的地*d, d₁*所对应的分拣格口,如图8所示.

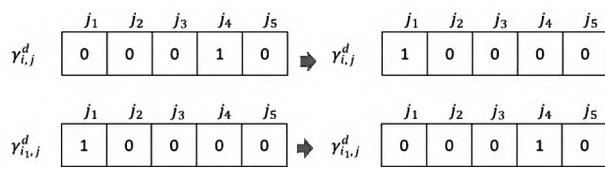


图7 $\gamma_{i,j}^d$ 的交换优化策略

Fig. 7 Exchange optimization strategy for $\gamma_{i,j}^d$

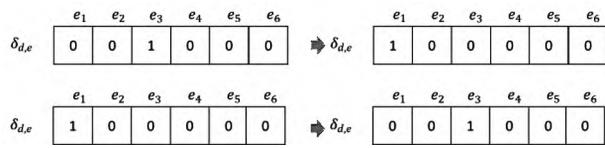


图8 $\delta_{d,e}$ 交换优化策略

Fig. 8 Exchange optimization strategy for $\delta_{d,e}$

3 数值实验

3.1 数据说明

本研究所用实验平台的CPU为Intel Xeon E5-2643 v4 3.4Ghz,内存为256GB,六核处理器,采用Windows10 64位处理系统.代码在C#中实现,C#版本为Visual Studio 2015.使用版本号为12.6.1的Cplex用于求解原问题模型、RMP与PP.

本研究实验中所出现的包裹规格共三种,每种规格包裹的数量均为总数的三分之一,三种规

格包裹的最大尺寸分别为 $0.7 \text{ m} \times 0.7 \text{ m}$ 、 $0.6 \text{ m} \times 0.4 \text{ m}$ 和 $0.4 \text{ m} \times 0.2 \text{ m}$, 交叉带托盘所能分拣的包裹的最大尺寸为 $0.8 \text{ m} \times 0.8 \text{ m}$. 每个自动分拣系统由上下两条传送带组成, 每条传送带对应 50 个分拣格口以及 4 个供包台. 当目的地数量 $|D|$ 大于分拣格口总数 $|E|$ 时, 将目的地分成 $|E|$ 组, 每组目的地的货量尽可能均匀分配, $|E|$ 个目的地组合由 $|E|$ 个分拣格口进行分拣. 例如: 分拣格口有 10 个, 当目的地数量为 30 时, 根据目的地 d 的总包裹数量 $\sum_{s \in S} n_{s,d}$ 由大到小进行排序, 将货量顺位为 20, 19, 18, 17, 16, 15, 14, 13, 12, 11 的目的地依次分配给货量顺位为 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 的目的地组成目的地组合, 再将货量顺位为 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 的目的地依次分配给货量顺位为 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 的目的地组成目的地组合. 最后, 通过执行算法为 10 个目的地组合制定合理的分拣格口分配方案. 表 1 列出了七个算例组 (ISG) 的参数设置. 小规模算例为 ISG1 至 ISG3, 其余算例为大规模算例. 表 2 及后续表格中所有结果都为一次计算的结果.

表 1 算例说明

Table 1 Instance illustration

算例规模	包裹数量	目的地数量	供包台数量
ISG1	60	9	8
ISG2	90	10	8
ISG3	120	11	8
ISG4	150	12	8
ISG5	180	13	8
ISG6	210	14	8
ISG7	240	15	8

3.2 算法参数测试

本研究所设计算法的参数如下: VNS 的最大迭代次数、VNS 中振动过程的阈值、VNS 的停止准则所涉及的迭代次数以及可行解的优化策略中的最大迭代次数. 为确定本研究所设计算法中各个参数的取值范围, 故对各项参数进行实验.

表 2 VNS 停止准则涉及的迭代次数对比

Table 2 Comparison of the iteration numbers involved in the VNS stopping criterion

算例		VNS 不同停止准则的求解时间			
规模	编号	F_{CGHA}	t_1 (s)	t_3 (s)	t_5 (s)
ISG1	1	52	15	11	11
	2	54	11	11	11
ISG2	1	83	37	25	24
	2	82	28	27	26
ISG3	1	120	50	49	48
	2	114	78	77	76
ISG4	1	148	138	124	122
	2	147	125	90	84
ISG5	1	179	207	205	134
	2	184	321	138	135
ISG6	1	209	599	233	221
	2	203	339	331	320
ISG7	1	251	766	340	315
	2	253	619	438	437

注: F_{CGHA} 表示由 VNS 算法与规则可行化加速后基于列生成的算法得到的最优解; t_1 、 t_3 、 t_5 、 t_7 分别表示 VNS 停止准则涉及的迭代次数为 1、3、5 和 7 计算的耗时.

表 2 对 VNS 的停止准则所涉及的迭代次数进行测试. 由于 VNS 中的最大迭代次数和停止准则所涉及的迭代次数本质上无差别, 因此本研究未对 VNS 的最大迭代次数进行实验, 本研究 VNS 中最大迭代次数为 50 次. 由实验结果可知, 在 VNS 的停止准则设置为: 若局部最优解在 5 次迭代中未发生变化则停止. 当迭代次数为 1 或 3 时, 由于 PP 生成的计划未能完全求得较优解, 会使 RMP 的求解次数增多导致求解时间偏长; 当迭代次数为 7 时, 不必要的迭代时间内会延长算法的整体求解时间.

表 3 对 VNS 中振动过程的阈值进行测试. 在 VNS 迭代过程中, 若对局部最优解进行更新, 则随机取得一个浮点值, 若该浮点值大于振动过程的阈值, 则将该局部最优解作为初始解, 否则不更新初始解. 因此, 振动过程阈值的大小不仅影响到求解速度的快慢, 也对求解质量的高低有影响. 由表 3 可知当阈值为 0.2 时求解质量最佳. 当阈值为 0 时, 表示振动过程未进行, 此时极易导致陷入局部最优解, 从而增加 RMP 的求解次数; 当阈值为 0.5 或 0.8 时, 会限制局部最优解的更新, 导致 VNS 求解时间变长, 也会影响求解时间.

表3 VNS 振动过程的阈值对比

Table 3 Comparison of threshold values for VNS vibration processes

算例		VNS 振动过程不同阈值的求解时间			
规模	编号	F_{CGHA}	t_0 (s)	$t_{0.2}$ (s)	$t_{0.5}$ (s)
ISG1	1	52	11	11	11
	2	54	11	11	11
ISG2	1	83	37	24	24
	2	82	25	24	25
ISG3	1	120	49	48	49
	2	114	75	73	76
ISG4	1	148	131	122	132
	2	147	89	84	83
ISG5	1	179	168	134	139
	2	184	135	135	136
ISG6	1	209	221	214	224
	2	203	329	315	328
ISG7	1	251	344	315	339
	2	253	484	437	437

注： F_{CGHA} 表示由 VNS 算法与规则可行化加速后基于列生成的算法得到的最优解； t_1 、 t_3 、 t_5 、 t_7 分别表示 VNS 振动过程阈值为 0.0.2.0.5 和 0.8 计算的耗时。

表4 对于可行解的优化策略中的最大迭代次数进行测试。由实验数据可知，当优化策略中的最大迭代次数为 3 时，效果最佳。若为 1 时，则可能无法达到优化效果；当为 5 时则会产生不必要的额外迭代此时，导致求解时间过长。因此，优化策略中的最大迭代次数为 3 次。

表4 可行解的优化策略中最大迭代次数对比

Table 4 Comparison of the maximum iteration numbers in the optimization strategies of feasible solutions

算例		优化策略中不同最大迭代次数的求解时间			
规模	编号	t_1 (s)	F_1	t_3 (s)	F_3
ISG1	1	11	54	11	54
	2	11	52	11	52
ISG2	1	26	83	24	82
	2	24	82	26	82
ISG3	1	48	120	48	120
	2	70	114	73	114
ISG4	1	121	148	122	148
	2	80	148	84	147
ISG5	1	124	179	134	179
	2	125	184	135	184
ISG6	1	203	210	221	209
	2	313	203	315	203
ISG7	1	290	251	315	251
	2	421	253	437	253

注： F_1 、 F_3 、 F_5 分别表示可行解的优化策略中最大迭代次数为 1、3 和 5 时，算法得到的最优解； t_1 、 t_3 、 t_5 分别表示 VNS 振动过程阈值为 0.0.2.0.5 和 0.8 计算的耗时。

根据以上结果可知，当迭代次数为 50 次，停止条件涉及的迭代次数为 5 次，振动过程涉及的阈值为 0.2，可行解的优化策略中的最大迭代次数为 3 次时算法性能最佳。

3.3 有效性测试

为了测试本研究提出的算法是否能够在合理的计算时间内求得近似最优解，本研究使用 Cplex 对所提的原问题模型进行求解。为方便比较，本研究提出了一个下界模型 (LB)。在原问题模型中，交叉带托盘与目的地的服务关系以及包裹与交叉带托盘的归属关系决定每个交叉带托盘的分组模板，分组模板决定“成组分拣”距离；目的地与分拣格口归属关系以及每个交叉带托盘运送包裹的目的地类型决定了交叉带托盘分拣包裹所产生的分拣距离。通过对于模型的分析可知，在原问题模型中 $\theta_{j,d}, \delta_{d,e}$ 对于决策的重要程度高于 $\alpha_{j,g}^d, \gamma_{i,j}^d, \beta_{j,k,e}^d$ ，因此若需对决策变量进行松弛，则只能在 $\alpha_{j,g}^d, \gamma_{i,j}^d, \beta_{j,k,e}^d$ 中进行选择，否则会导致下界模型与原问题模型的 gap 过大。且在对于变量松弛后，去除部分与该类决策变量相关的约束可进一步加快求解，因此该下界模型为去除原问题模型中的约束(3)与约束(4)后，并对决策变量 $\gamma_{i,j}^d, \beta_{j,k,e}^d$ 进行松弛，从而得到原问题模型的 LB，LB 也通过 Cplex 进行求解。求解速度相较原问题模型更为迅速。在该实验中使用的小规模算例组 ISG1，ISG2 和 ISG3，Cplex、LB 与本研究算法之间的比较结果如表 5 所示。

在小规模情况下，Cplex 只能对部分算例进行求解，当规模增大时，Cplex 无法在两小时内求得原问题模型最优解，本研究模型的下界模型 (LB) 在求解速度上要快于原问题模型且目标值小于原问题模型。本研究所提出的基于列生成的启发式算法在算例规模为 ISG1 和 ISG2 的情况下与 Cplex 的 gap 保持在 1.20% 以下，并且求解速度快于 Cplex，与下界模型 (LB) 的 gap 保持在 1.48% 以下，验证了该算法的有效性和高效性。在对子问题的求解流程加速之后，该算法的求解效率进一步提升，随着算例规模的变大，规则可行化加速的优势越来越明显。在算例规模为 ISG2 和 ISG3 的情况下，规则可行化加速后的求解时间均明显短于 Cplex 可行化的求解时间，并且求解质量一致。

大规模实验已经无法用 Cplex 求解原问题模型解决,故采用原问题模型的下界模型(LB)和基于列生成的启发式算法进行求解,并对得到的解进行评价。从表 6 中可以看出,在大规模实验中,基于列生成的启发式算法的求解效果较好,与下界模型(LB)的差距保持在 4.93% 以内。规则可行化加速后的算法求解时间均明显短于 Cplex 可行化加速以及未加速并用 Cplex 可行化的算法求解时间,并且求解质量也一致。由此可见本研究所使用的算法在求解质量上还是较为出色的,该算法对于原问题模型的求解是有效且快速的,对于子问题求解的加速同样是非常有效的。并且在求解子问题或进行可行化时,在保证求解质量的前提下不采用 Cplex,则可有效提升求解速度。

本系统共 8 个供包台,在 ISG7 中每个供包台分拣的包裹数量为 30 个,一台自动堆码机器人完

成一个包裹的信息收集流程即:拿起包裹、扫码、称重、测量尺寸、放下包裹、确认完成以上流程共需要 7 s 完成,则每个供包台信息收集所需时间为 210 s。本研究所研究的分拣系统每层传送带长 630 m,速度为 2.7 m/s,若每个包裹的平均运输时间为 117 s,则完成所有包裹的分拣任务的时间为 327 s,即 5.5 min;若每个包裹的平均运输时间为 234 s 时,则完成所有分拣任务的时间为 444 s,即 7.4 min,即每个批次包裹的分拣完成时间为 6 min~8 min。在实际运行环境中,在本批次的包裹到达供包台时,本批次包裹的调度方案计算完成,本批次包裹依照该调度方案进行分拣,同时系统计算下一批次包裹的调度方案,在下一批次包裹到达时,本批次的包裹分拣完成且下一批次的分拣调度方案计算完成,并循环往复直至所有批次任务完成。因此,该算法可有效应用于实际的包裹分拣运作中。

表 5 小规模下 Cplex、下界模型和基于列生成的启发式算法对比

Table 5 Comparison of Cplex, LB and column generation-based heuristics in small-scale instances

算例		Cplex		LB	CGH + Cplex		CGHA + Cplex		CGHA + 规则			
规模	编号	F_{Cplex}	t_{Cplex} (s)	F_{LB}	F_{CGH}	t_{CGH} (s)	F_{CGHM}	t_{CGHM} (s)	F_{CGHA}	t_{CGHA} (s)	Δ_{Cplex}	Δ_{LB}
ISG1	5-1	55	207	54.20	55	32	55	10	55	13	0.00%	1.48%
	5-2	57	161	57.00	57	27	57	8	57	13	0.00%	0.00%
	5-3	52	153	52.00	52	22	52	7	52	12	0.00%	0.00%
	5-4	53	190	52.26	53	51	53	10	53	13	0.00%	1.42%
	5-5	54	182	54.00	54	38	54	10	54	13	0.00%	0.00%
ISG2	5-6	83	3 612	82.91	84	93	84	32	84	40	1.20%	1.31%
	5-7	83	4 103	83.00	83	151	83	40	83	28	0.00%	0.00%
	5-8	85	3 220	85.00	85	128	85	31	85	28	0.00%	0.00%
	5-9	89	3 466	87.98	89	111	89	38	89	29	0.00%	1.16%
	5-10	88	3 558	88.00	88	121	88	45	88	40	0.00%	0.00%
ISG3	5-11	—	>7 200	119.00	120	269	120	100	120	107	—	0.84%
	5-12	—	>7 200	122.00	122	375	122	103	122	53	—	0.00%
	5-13	—	>7 200	125.38	126	304	126	94	126	54	—	0.49%
	5-14	—	>7 200	123.00	123	372	123	116	123	58	—	0.00%
	5-15	—	>7 200	125.33	127	313	127	131	127	79	—	1.33%

注: F_{Cplex} 、 F_{LB} 、 F_{CGH} 、 F_{CGHM} 、 F_{CGHA} 分别表示由 Cplex、下界模型、基于列生成的启发式算法、VNS 算法与 Cplex 可行化加速后基于列生成的算法和 VNS 算法与规则可行化加速后基于列生成的算法得到的最优解; t_{Cplex} 、 t_{CGH} 、 t_{CGHM} 、 t_{CGHA} 分别表示 Cplex、基于列生成的启发式算法、VNS 算法与 Cplex 可行化加速后基于列生成的算法和 VNS 算法与规则可行化加速后基于列生成的算法计算的耗时;

Δ_{Cplex} 表示算法所得结果与 Cplex 所得结果的相对误差, $\Delta_{Cplex} = \frac{F_{CGHA} - F_{Cplex}}{F_{Cplex}}$; Δ_{LB} 表示算法所得结果与下界模型所得结果的相对

$$\text{误差}, \Delta_{LB} = \frac{F_{CGHA} - F_{LB}}{F_{LB}}.$$

表6 在大规模情况下,下界模型和基于列生成的启发式算法对比

Table 6 Comparison of LB and column generation-based heuristics in large-scale instances

算例		LB	CGH + Cplex		CGHA + Cplex		CGHA + 规则		
规模	编号	F_{LB}	F_{CGH}	$t_{CGH}(s)$	F_{CGHM}	$t_{CGHM}(s)$	F_{CGHA}	$t_{CGHA}(s)$	Δ_{LB}
ISG4	6-1	152.29	155	792	155	259	155	132	1.78%
	6-2	149.00	150	866	150	182	150	93	0.67%
	6-3	142.00	149	902	149	198	149	91	4.93%
ISG5	6-4	178.00	178	1124	178	399	178	215	0.00%
	6-5	173.00	173	1395	173	368	173	149	0.00%
	6-6	176.85	177	1389	177	377	177	153	0.08%
ISG6	6-7	198.00	201	2923	201	777	201	239	1.52%
	6-8	189.00	193	2734	193	785	193	242	2.12%
	6-9	190.00	192	3063	192	878	192	247	1.05%
ISG7	6-10	242.00	246	3512	246	1277	246	337	1.65%
	6-11	246.00	251	3768	251	1234	251	371	2.03%
	6-12	252.00	257	3795	257	1467	257	482	1.98%

注： F_{Cplex} 、 F_{LB} 、 F_{CGH} 、 F_{CGHM} 、 F_{CGHA} 分别表示由 Cplex、下界模型、基于列生成的启发式算法、VNS 算法与 Cplex 可行化加速后基于列生成的算法和 VNS 算法与规则可行化加速后基于列生成的算法得到的最优解； t_{Cplex} 、 t_{CGH} 、 t_{CGHM} 、 t_{CGHA} 分别表示 Cplex、基于列生成的启发式算法、VNS 算法与 Cplex 可行化加速后基于列生成的算法和 VNS 算法与规则可行化加速后基于列生成的算法计算的耗时； Δ_{LB} 表示第四种算法 (CGHA + 规则, 本研究所提算法中最好的) 所得结果与下界模型所得结果的相对误差 $\Delta_{LB} = \frac{F_{CGHA} - F_{LB}}{F_{LB}}$ 。

3.4 敏感度分析

对于标准化分拣系统而言分拣格口、供包台数量是固定的,但对于定制化系统而言供应商可根据客户的需求对于设备做出布局上的调整. 本节对于分拣系统中各类设备的各项参数进行敏感度分析,意在通过实验得知分拣系统在哪些布局上的调整能对分拣效率产生实质性影响.

通过 1.2 节的数学模型可知,在众多对分拣效率的影响因素中,本研究模型中主要涉及到了供包台数量、分拣格口数量和目的地数量等,其中供包台数量和分拣格口数量会影响系统设计从而对分拣效率产生实质影响;当包裹数量不变,目的地数量变化直接影响散货数量,从而对分拣系统的效率也会产生影响. 在对分拣效率进行优化时,供包台数量、分拣格口数量和目的地数量是不容忽视的,因此本研究的敏感性分析从供包台数量、分拣格口数量和目的地数量入手进行试验分析.

3.4.1 供包台数量的敏感度分析

本研究的分拣系统共 8 个供包台,在一般的双层分拣系统中,供包台数量一般控制在 4 个至 20 个. 选取 4 个规模进行敏感度测试,其中算例

规模 ISG4 至 ISG7 对应的目的地数量分别为 12、13、14 和 15. 由图 9 可知,当供包台数量与目的地数量的比值小于 100%,供包台数量越多,每小时的包裹吞吐量越大;当供包台数量与目的地数量的比值超过 100% 后吞吐量开始停止增长. 在制定调度方案时,越接近供包台的分间隔口,越有可能被分配货运量较大的目的地,即该分拣格口的“热度”较高. 因此,供包台的数量越多,“热度”较高的分拣格口数量越多,分拣效率越高.

3.4.2 分拣格口数量的敏感度分析

本研究的分拣系统的分拣格口数量为 50 个,为了验证分拣格口数量对于吞吐量的影响. 对分拣格口数量为 50 至 100 的情况分别进行了实验. 实验结果由图 10 可知,分拣系统的吞吐量不随分拣格口的增加而产生变化. 由于供包台数量并未增加,而分拣格口间的距离又相对固定. 若增加分拣格口,传送带越长,已有的分拣格口到供包台的距离也不会发生改变,“热度”较高的分拣格口数量也不会出现变化,目的地与分拣格口间的匹配方案也不会发生任何改变. 因此在分拣格口的数量与吞吐量的增长无关.

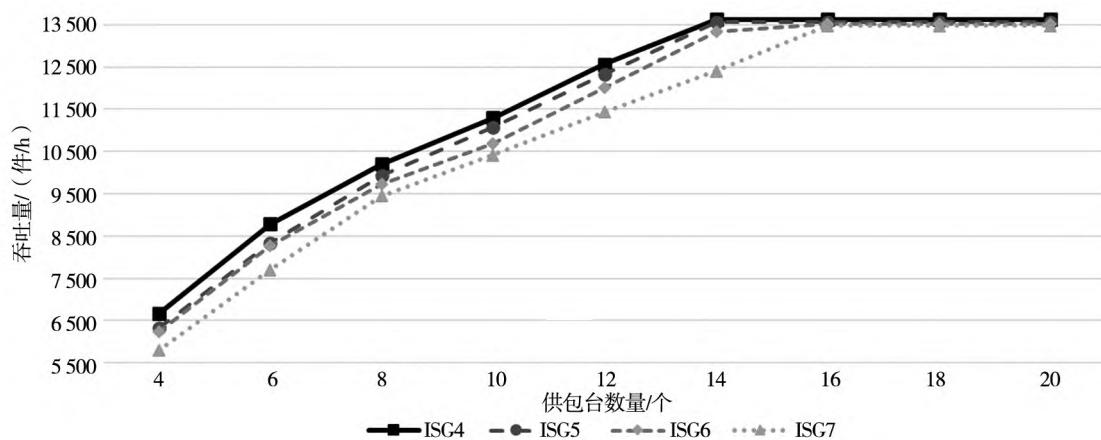


图 9 供包台数量与分拣系统吞吐量的关系

Fig. 9 Relationship between the numbers of supply stations and sorting system throughput

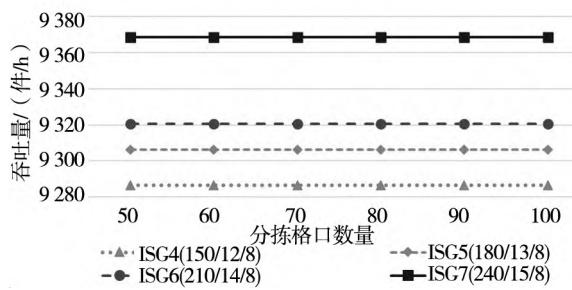


图 10 分拣格口数量与分拣系统吞吐量的关系

Fig. 10 Relationship between the numbers of sorting boxes and sorting system throughput

3.4.3 关于目的地数量的敏感度分析

本节选取 ISG5 规模的算例对于目的地的数量进行敏感度测试。由图 11 可知,在同一规模算例中,由于包裹数量不变,目的地数量越多,散货数量越多,六种分拣系统的效率也随之降低。

在算例规模为 ISG5 时,当目的地数量由 10 个增加至 50 个,双层反向系统与双层同向系统的吞吐量的比值随目的地数量的增加而递增,由 137% 增长至 181%,随着散货数量的增加双层反向系统的优越性变得更为明显。在双层反向系统中由于上下层传送带的运输方向相反,包裹可选的路线相较双层同向系统来的更多。因此,在分拣格口利用率较高的情况下,双层反向系统相较双层同向系统的劣势越明显。并且,由图 11 可知“成组分拣”可进一步提升分拣系统的分拣效率。由于成组分拣系统考虑一次运输多件包裹,该系统的分拣效率远高于相较其他四类分拣系统。而在同时采用双层反向系统与成组分拣系统时,分

拣效率最高。

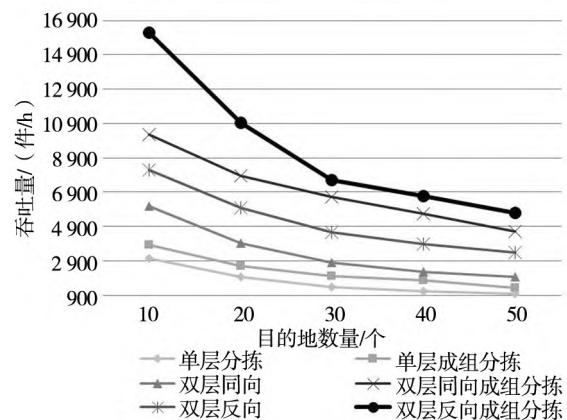


图 11 目的地数量与分拣系统吞吐量的关系

Fig. 11 Relationship between the numbers of destinations and sorting system throughput

3.4.4 关于大规格包裹比例的敏感度分析

本节进行了大规格包裹比例对分拣系统吞吐量的影响的敏感度分析,并且对于本研究所提出的“成组分拣”的适用范围进行了测试,进行了大规格包裹比例对分拣系统吞吐量的影响的敏感度分析,并对本研究所提出的“成组分拣”的适用范围进行了测试。本节将大规格包裹的数量比例在 10%–100% 的范围内进行变化,在大规格包裹数量比例固定的情况下,中小规格包裹的比例相同,若出现无法进行均分的情况时,规定中规格包裹可比小规格包裹多一件,例如:ISG4 中,此时包裹数量为 150 件,当大规格包裹比例为 10% 时,中小规格包裹的比例均为 45%,此时大规格包裹为 15 件,中规格包裹为 67 件,小规格包裹为 68 件。

图12为小规模与大规模情况下大规格包裹比例与分拣系统吞吐量的关系图的关系图,随着大规格包裹的比例变大,采用“成组分拣”的次数减少,分拣系统的吞吐量会随之降低;并且由图12可知,当大规格包裹的比例超过90%时,大部分情况下进行“成组分拣”和不进行“成组分拣”的吞吐量一样,说明此时“成组分拣”不再适用。此时,每件包裹都独自由一个托盘进行分拣,不会出现“成组分拣”的情况。并且,当目的地的

包裹数量较少时也会导致“成组分拣”失效,此时每件包裹无法进行“成组分拣”,ISG1-ISG5 每个目的地的平均包裹数量为7件、9件、11件、13件、14件。ISG1 平均每个目的地包裹数量为7件,在大规格包裹比例为80%时,“成组分拣”失效;ISG2-ISG4 平均每个目的地包裹数量为9件、11件、13件,在大规格包裹比例为90%时,“成组分拣”失效;ISG5 平均每个目的地包裹数量为14件,“成组分拣”未失效。

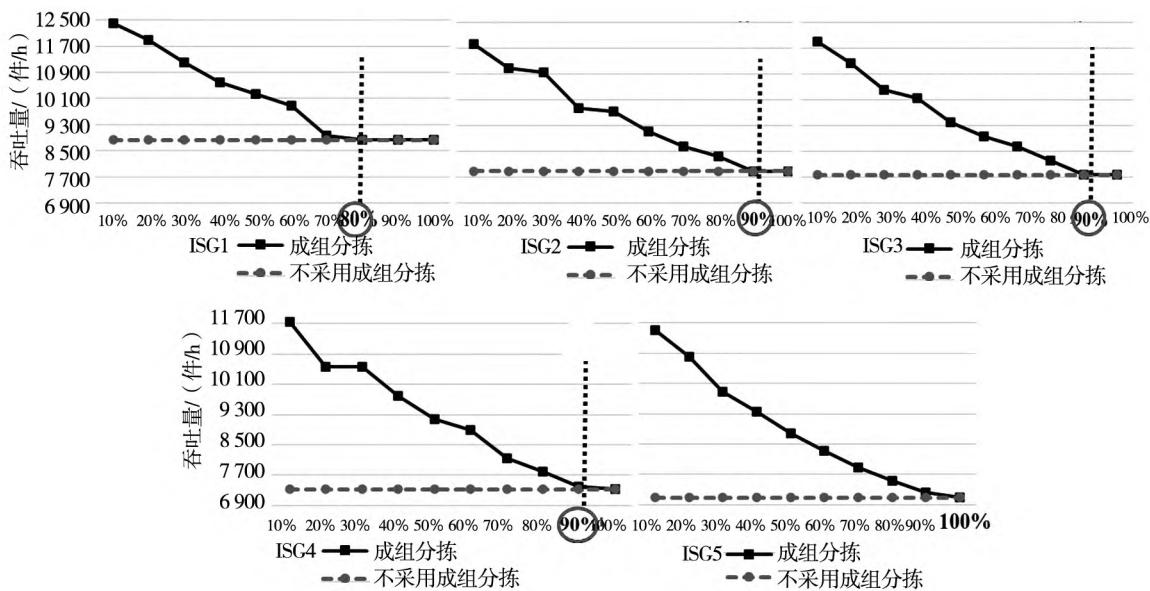


图12 成组分拣模式采用临界点关系 (ISG1 – ISG5)

Fig. 12 Critical point for adopting group sorting mode (ISG1 – ISG5)

因此,“成组分拣”在以下三种情况下失效:
1) 包裹数量小于9件,大规格包裹比例超过80%;2) 包裹数量不小于9件且不大于13件,且大规格包裹比例超过90%;3) 所有包裹均为大规格包裹。若采用可“成组分拣”的双层分拣系统对包裹进行分拣时,应使每个目的地的包裹数量尽量超过14件,以最大化分拣系统的分拣效果。

4 结束语

本研究针对文献中较少研究但当前普遍流行的新系统——“双层自动化分拣系统”,考虑了一种包裹分拣的新策略——“成组分拣”,采用了一种仓库系统优化研究中较少使用但在大规模整数

规划求解中当前被普遍推崇的新算法——“列生成算法”,研究了一个考虑成组分拣的双层分拣系统中包裹、交叉带托盘、分拣格口、供包台等资源的分配优化问题,建立了整数规划模型。该模型结合“成组分拣”以及双层分拣系统的特点,以双层分拣系统完成所有包裹的总分拣距离,即“成组分拣”与分拣所产生的距离之和最小化为目标,有效地提高了分拣系统的分拣效率。本研究所设计的基于列生成的算法,通过将原问题拆解为限制性主问题与子问题,再将限制性主问题中的对偶变量代入子问题中进行求解,能更加有效地生成可行的分拣调度方案,达到提升求解质量的目的。为了进一步提升求解速度,本研究还专门针对子问题求解设计了变邻域搜索算法。为验证本研究所设计算法的高效性和稳定性,本研究开展

了大量实验将该算法与 Cplex 和原问题模型的下界模型进行比较。相较于 Cplex, 该算法的求解时间更短, 且能求解大规模算例。与下界模型的比较也验证了该算法的有效性, 无论大规模还是小规模, 该算法与下界的差距不会超过 3%。

基于上述模型和算法, 本研究还通过敏感性分析实验探究了哪些参数会对分拣系统的吞吐量起到关键因素。基于本研究实验结果, 可以为配送中心运营商在自动化分拣系统选型与配置决策提供如下管理启示与建议:

1) 供包台数量的大小在极大程度上影响着设备的分拣效率, 供包台数量越多, “热度”较高的分拣格口数量也就越多, 分拣效率越高。因此, 在选择分拣系统时, 建议选择供包台数量较大的分拣系统。

2) 分拣格口的数量对于分拣效率没有明显的提升。并且分拣格口数量越多, 传送带越长, 分拣系统的造价也就越高。因此, 在选择分拣系统时, 不建议选择分拣格口数量较大的分拣系统。

3) 在选择分拣系统时, 建议采用双层分拣系统进行分拣作业。相较于单层分拣系统, 双层分拣系统具有占地面积小、分拣效率高的优点。此外, 在使用双层分拣系统时, 建议使上下层传送带沿相反的方向运行, 用于增加“热度”较高的分拣格

口数量, 从而提升双层分拣系统的分拣效率。

4) 在采用双层分拣系统对包裹进行分拣时, 应该考虑将包裹进行“成组分拣”。考虑“成组分拣”可以有效地提升交叉带托盘的利用率, 这种分拣方式能有效减少分拣作业的分拣距离, 从而缩短分拣时间, 进而提高分拣效率。

5) 在采用可“成组分拣”的双层分拣系统对包裹进行分拣时, 应使每批次分拣每个目的地的包裹数量尽量超过 14 件, 以最大化分拣系统的分拣效果。

未来的研究可以从如下三个方面进一步扩展:

1) 本研究未考虑传送带拥堵的影响, 今后的研究可将拥堵因素考虑在内, 通过最小化传送带的传送距离, 用于减少由于拥堵而产生的延误, 进而提升分拣效率。

2) 本研究引入了分组模板的概念, 今后的研究可以考虑将包裹的实际尺寸引入进行建模。从而将本研究所研究的问题扩展为一个考虑二维装包问题的目的地指派问题。

3) 本研究所使用的算法为基于列生成的启发式算法, 未来可尝试将本研究所用的列生成算法与分支定界法进行结合, 探索基于分支定价精确解算法的可行性。

参 考 文 献:

- [1] 朱 杰, 周 丽, 郭 键. 分类存储人工拣选随机服务系统效率研究[J]. 管理科学学报, 2012, 15(2): 59–71.
Zhu Jie, Zhou Li, Guo Jian. Research on model and efficiency of manual order picking service system[J]. Journal of Management Sciences in China, 2012, 15(2): 59–71. (in Chinese)
- [2] Khachatryan M. Small Parts High Volume Order Picking Systems[D]. Georgia: Georgia Institute of Technology, 2006.
- [3] Briskorn D, Emde S, Boysen N. Scheduling shipments in closed-loop sortation conveyors[J]. Journal of Scheduling, 2017, 20(1): 25–42.
- [4] Boysen N, Fedtke S, Weidinger F. Optimizing automated sorting in warehouses: The minimum order spread sequencing problem[J]. European Journal of Operational Research, 2018, 270(1): 386–400.
- [5] Boysen N, Stephan K, Weidinger F. Efficient order consolidation in warehouses: The product-to-order-assignment problem in warehouses with sortation systems[J]. IIE Transactions, 2022, 54(10): 963–975.
- [6] Novoa L J, Jarrah A I, Morton D P. Flow balancing with uncertain demand for automated package sorting centers[J]. Transportation Science, 2018, 52(1): 210–227.
- [7] Fedtke S, Boysen N. Layout planning of sortation conveyors in parcel distribution centers[J]. Transportation Science, 2017,

- 51(1) : 3 – 18.
- [8] Chen F, Song K. Minimizing makespan in two-stage hybrid cross docking scheduling problem[J]. Computers & Operations Research, 2009, 36(6) : 2066 – 2073.
- [9] van der Gaast J, de Koster R, Adan I. Conveyor merges in zone picking systems: A tractable and accurate approximate model[J]. Transportation Science, 2018, 52(6) : 1428 – 1445.
- [10] Hou J L, Wu N, Wu Y J. A job assignment model for conveyor-aided picking system[J]. Computers & Industrial Engineering, 2009, 56(4) : 1254 – 1264.
- [11] 王帅安. 自动分拣系统订单处理策略研究[D]. 北京: 清华大学, 2009.
- Wang Shuaian. Research on Order Processing Strategy in Automated Sorting Systems [D]. Beijing: Tsinghua University, 2009. (in Chinese)
- [12] 卢少平, 张贻弓, 吴耀华, 等. 自动分拣系统并行分区拣选优化策略[J]. 深圳大学学报(理工版), 2010, 27(1) : 120 – 126.
- Lu Shaoping, Zhang Yigong, Wu Yaohua, et al. Synchronized zoning strategy optimization of automated picking system [J]. Journal of Shenzhen University Science and Engineering, 2010, 27(1) : 120 – 126. (in Chinese)
- [13] Zenker M, Boysen N. Dock sharing in cross-docking facilities of the postal service industry[J]. The Journal of the Operational Research Society, 2018, 69(7) : 1061 – 1076.
- [14] 张源凯, 胡祥培, 黄敏芳, 等. 网上超市拆分订单合并打包策略经济决策模型[J]. 管理科学学报, 2019, 22(10) : 24 – 36 + 100.
- Zhang Yuankai, Hu Xiangpei, Huang Minfang, et al. Economic decision model for package consolidation in fulfilling split orders of online supermarkets[J]. Journal of Management Sciences in China, 2019, 22(10) : 24 – 36 + 100. (in Chinese)
- [15] vander Gaast J, de Koster R, Adan I, et al. Capacity analysis of sequential zone picking systems[J]. Operations Research, 2020, 68(1) : 161 – 179.
- [16] Zou B, de Koster R, Gong Y, et al. Robotic sorting systems: Performance estimation and operating policies analysis[J]. Transportation Science, 2021, 55(6) : 1430 – 1455.
- [17] Chen T L, Chen J C, Huang C F, et al. Solving the layout design problem by simulation-optimization approach-A case study on a sortation conveyor system[J]. Simulation Modelling Practice and Theory, 2021, 106 : 102192.
- [18] 田倩南, 李昆鹏, 李文莉, 等. 基于改进列生成算法的受扰航班优化调度[J]. 系统工程理论与实践, 2019, 39(11) : 2815 – 2827.
- Tian Qiannan, Li Kunpeng, Li Wenli, et al. Optimization operation of disrupted flights by improving column generation algorithm[J]. Systems Engineering: Theory & Practice, 2019, 39(11) : 2815 – 2827. (in Chinese)
- [19] 罗治洪, 唐立新. 炼钢热轧一体化生产与物流计划模型及求解[J]. 管理科学学报, 2011, 14(6) : 16 – 23.
- Luo Zihong, Tang Lixin. Modeling and solution to integrated production and logistics planning of steel making and hot rolling[J]. Journal of Management Sciences in China, 2011, 14(6) : 16 – 23. (in Chinese)
- [20] 李兆进, 刘雅, 杨臻. 考虑订单合并和货物转运的多式联运路径优化研究[J]. 运筹与管理, 2022, 31(4) : 28 – 34.
- Li Zhaojin, Liu Ya, Yang Zhen. Research on the optimization of multimodal transportation route considering order consolidation and freight transshipment[J]. Operations Research and Management Science, 2022, 31(4) : 28 – 34. (in Chinese)
- [21] 李珍萍, 贾顺顺, 卜晓奇, 等. 无人仓系统储位分配问题的优化模型与算法[J]. 中国管理科学, 2022, 30(1) : 124 – 135.
- Li Zhenping, Jia Shunshun, Bu Xiaoqi, et al. The optimization model and algorithm for storage location assignment problem in unmanned warehouse system[J]. Chinese Journal of Management Science, 2022, 30(1) : 124 – 135. (in Chinese)
- [22] 李阳, 范厚明, 张晓楠, 等. 求解模糊需求车辆路径问题的两阶段变邻域禁忌搜索算法[J]. 系统工程理论与实

践, 2018, 38(2): 522–531.

Li Yang, Fan Houming, Zhang Xiaonan, et al. Two-phase variable neighborhood tabu search for the capacitated vehicle routing problem with fuzzy demand[J]. Systems Engineering: Theory & Practice, 2018, 38(2): 522–531. (in Chinese)

Parcel sorting optimization for double-layer automatic sorting systems

ZHEN Lu, TAN Zhe-yi, GAO Jia-jing

School of Management, Shanghai University, Shanghai 200444, China

Abstract: With the rapid development of e-commerce in recent years, the delivery orders of parcels are characterized by small batches, multiple varieties, frequent orders, short delivery cycles, and high precision requirements, which requires more efficient sorting operations. Logistics companies have gradually begun to use “double-layer automatic sorting systems” to improve the sorting efficiency and shorten the sorting time. In order to further improve the sorting efficiency, the “group sorting” strategy is recently adopted by some companies to improve the pallet utilization. This strategy combines multiple parcels with the same destination according to certain rules, and then sorts the normalized and standardized large units. This paper studies an optimization problem related to locating packages, pallets, sorting boxes, and supply stations in a double-layer sorting system considering the group sorting strategy. An integer programming model is established, an efficient solution algorithm based on column generation is developed, and acceleration techniques for pricing problem are implemented to further shorten the solution time. Experimental results validate the effectiveness of applying the group sorting strategy in a double-layer automatic sorting system. In addition, based on some sensitivity analysis experiments on the proposed model and algorithm, this paper proposes managerial implications for technology adoption and equipment setting in automatic sorting systems.

Key words: double-layer sorting system; group sorting; sorting optimization; column generation; integer programming