DOI: 10.3901/JME.2020.24.246

考虑客户收货顺序的“货到人”分拣系统的 订单排序和客户分批优化

胡金昌 马文凯 杨 栋 吴耀华
(山东大学控制科学与工程学院 济南 250061)

摘要: 可以并行分拣多个客户订单的“货到人”分拣系统中, 每个客户包含多个订单, 客户要求按订单排序依次收货。为提高该系统的分拣效率, 以最小化料箱出入库数量为目标, 从订单排序和客户分批两方面进行优化。分别建立两个 0-1 整数规划模型解决多客户同时拣选时的订单排序优化和客户分批优化问题; 针对客户分批问题, 又提出种子算法和遗传算法来解决。设计试验检验了不同客户数量、客户订单数量、总品项数量时订单排序模型和客户分批算法的优化效果。试验结果表明, 0-1 整数规划模型优化订单排序, 可提高效率约 15%, 具有有效性; 客户分批优化方面, 0-1 整数规划模型、遗传算法和种子算法都可以不同程度地提高系统效率, 分别适合不同问题规模和时间要求的场景。

关键词: “货到人”分拣; 客户分批; 订单排序; 0-1 整数规划模型; 种子算法; 遗传算法

中图分类号: O223

“Part-to-picker” Picking System Order Scheduling and Customer Batching Considering Commodities Received by an Order's Sequence

HU Jinchang MA Wenkai YANG Dong WU Yaohua
(College of Control Science and Engineering, Shandong University, Jinan 250061)

Abstract: In the “part-to-picker” picking system which can pick multiple customer orders in parallel, each customer has multiple orders, and requires to receive commodities by the order sequence. To improve picking efficiency of this system, order scheduling and customer batching are optimized respectively, aiming at minimizing the number of “In-Out Stock” boxes. 0-1 integer programming model is established for scheduling orders of customers. Two different 0-1 integer programming models are established to solve customer batch optimization problems with multiple customers picking at the same time and the order scheduling optimization problem respectively. In addition, the seed algorithm and genetic algorithm are proposed for solving the customer batch problem. Numerical experiments are designed to evaluate the performance of proposed order scheduling model and algorithms of customer batching problem in different customer number, order number and stock keeping unit number. The experimental results show that the 0-1 integer programming model for optimizing order sequence can improve the efficiency by about 15%, which is effective. For customer batch optimization, 0-1 integer programming model, genetic algorithm and seed algorithm can improve the efficiency of the system with varying degrees, which can be applied for scenarios with different problem sizes and time requirements.

Key words: “part-to-picker” picking; customer batching; order scheduling; 0-1 integer programming; seed algorithm; genetic algorithm

0 前言

随着科技的发展和人力成本的提高, 仓储分拣领域逐渐向自动化、机械化、信息化方向发展, 分

销中心开始使用“货到人”仓储分拣系统取代传统的“人到货”分拣模式, 目前主流的“货到人”分拣系统有多层穿梭车仓储分拣系统、移动机器人仓储分拣系统、垂直升降式仓储分拣系统等。这些仓储分拣系统的一个拣选位置可以同时拣选多个拣选单位。以订单为拣选单位的“货到人”分拣系统根据订单情况, 将相应品项(Stock keeping unit, SKU)

的料箱运送到拣选位置, 拣选人员将该位置所有需要该 SKU 的订单一并拣选, 完成后系统将料箱运回仓库区域, 因此提高 SKU 料箱一次出入库满足的订单数量, 将减少料箱出入库次数, 提高系统效率。

某些分销场景以客户为拣选单位, 一个拣选位置可以同时为多个客户拣选, 同时拣选的客户为一批。一个客户有多个订单, 每个订单包含多个 SKU 的货物, 客户以某种容器(例如托盘)接收货物。为方便分销给下一级客户, 客户对收货时货物在容器中的摆放提出了要求: 同一个订单的货物集中摆放, 不同订单的货物不可混放。同一客户不同订单的摆放位置不作要求, 同一订单不同 SKU 的摆放位置不作要求。在按批拣选的前提下, 一批客户中任一客户未完成拣选前不能开始拣选下一批, 因此, 只提高一批客户中某一客户的拣选速度对拣选效率的提高影响不大。为保证各客户订单拣选速度平衡并保证收货顺序无误, 系统分拣时规定: 一批客户中不同客户同样次序的订单才能同时拣选。例如, 客户 A 的第 1 个订单没有完成之前第 2 个订单不能开始拣选, 客户 B 也是如此。客户 A 的第 1 个订单只能和客户 B 的第 1 个订单同时拣选, 客户 A 的第 2 个订单只能和客户 B 的第 2 个订单同时拣选。若此时客户 A 的第 1 个订单需要某 SKU, 而客户 B 的第 2 个订单也需要该 SKU, 则该 SKU 料箱出库只能满足客户 A 和 B 的第 1 个订单, 然后回库再出库满足客户 B 的第 2 个订单。因此, 某 SKU 的料箱出库并不一定能同时为所有需要该 SKU 的客户拣选。若能通过为每个客户的订单排序, 将尽量多相同 SKU 需求的不同客户的订单同时拣选, 则会减少料箱到达拣选位置的次数, 从而提高效率, 即订单排序优化。另外, 如何将所有的客户分批, 按批分配至拣选位置, 令其同时拣选, 也会影响料箱出入次数, 即客户分批优化。

以前的研究拣选单位多为订单级别, 即订单分批或排序。对于订单分批的研究多针对“人到货”分拣系统, 目标为最小化拣选人员行走距离, CERGIBOZAN 等^[1]对之前订单分批问题从分类、解决问题的技术和未来研究方面做了总结。MEHRJERDI 等^[2]对整合批次拣选和分配调度问题建立了混合整数规划模型, 并提出了两阶段启发式算法在合理时间内解决, 减少了任务完成总时间。VAN GILS 等^[3]运用全因素方差分析方法证明了订单分批和储存、分区、拣选路径之间的关系, 同时证明了协调这 4 个因素会大幅提高生产效益。张珺^[4]研究了订单合并(分批), 讨论了当前订单合并策略可

分为以下 5 种: 优先规则算法、种子算法、节约算法、数据挖掘算法和启发式算法。在“货到人”分拣系统方面, 吴颖颖等^[5]定义订单耦合因子表示两个订单之间可放置于拣选缓存中的共用货箱数量, 提出改进的 K-Mean 聚类算法, 对订单做排序优化。胡金昌^[6]研究了订单分配优化, 改进了 K-Means 算法, 提出了 4 种订单分配至拣选位置的策略。赵鑫亮^[7]建立了补货等待时间最短的订单排序优化模型, 设计了改进的自适应遗传算法对问题进行求解。吴思沛^[8]讨论了 K-Means 算法中动态 K 值和动态分配的情况, 提出 4 种订单聚类及分配拣选位置的策略。XIANG 等^[9]研究了移动机器人仓储分拣系统的库存安置和订单分批问题, 提出改进的变邻近搜索算法, 以最大化订单之间的相关性或者最小化订单的孤立, 实现移动机器人行走距离最短。NICOLAS 等^[10]建立了垂直升降式仓储系统的订单分批优化模型, 目的是最小化总完成时间, 并用实际数据对模型进行了验证。ZIVANIC 等^[11]研究了订单执行顺序对“pick or pass”系统的影响, 改进了原有算法, 并进行了仿真验证。这些研究都是一个订单包含多个 SKU, 针对订单的分批操作, 而对一个客户包含多个订单, 订单包含多个 SKU 的订单排序和客户分批的研究尚未出现。

本文参照已有文献, 将最小化料箱出入库次数作为优化目标, 针对客户要求收货顺序和多个客户同时拣选的场景, 通过优化客户订单拣选顺序和客户分批, 实现系统优化, 从而减少料箱的出入库次数, 提高系统效率。

1 订单排序模型

假设同时拣选的客户 $\{C_1, C_2, C_3, \dots, C_b\}$, 分别有 $n_1, n_2, n_3, \dots, n_b$ 个订单, 令 $N = \max(n_1, n_2, n_3, \dots, n_b)$, 有 s 种 SKU, 用 $Z_{ijk} = 1$ 表示第 i 个客户的第 j 个订单包含第 k 种 SKU, 否则 $Z_{ijk} = 0$, 其中 $1 \leq i \leq b$, $1 \leq j \leq N$, $1 \leq k \leq s$ 。当第 i 个客户的订单数量不足 N 个, 即 $n_i < N$ 时, 则 $Z_{ijk} = 0$, $n_i \leq j \leq N$, $1 \leq k \leq s$ 。 C_i 的第 j 个订单用数组表示为 $\{Z_{ij1}, Z_{ij2}, \dots, Z_{ijs}\}$, C_i 的所有订单可以用 $N \times s$ 的矩阵表示为

$$\begin{pmatrix} Z_{i11} & \cdots & Z_{i1s} \\ \vdots & & \vdots \\ Z_{iNs} & \cdots & Z_{iNs} \end{pmatrix}$$

在要求按任意订单顺序收货的情况下,若多个客户同时拣选,影响料箱出入库数量的是各客户的订单的组合,例如,客户 C_{i_1} 中的第 j_1 个订单和 C_{i_2} 中的第 j_2 个订单同时拣选会减少的料箱出入库数量,则可将 C_{i_2} 中的第 j_2 个订单调整为第 j_1 个拣选。设决策变量 $T_{j_1, j_2, \dots, j_b} = 1$ 表示 C_1 的第 j_1 个订单、 C_2 的第 j_2 个订单 $\dots C_b$ 的第 j_b 个订单组合拣选, $T_{j_1, j_2, \dots, j_b} = 0$ 表示 C_1 的第 j_1 个订单、 C_2 的第 j_2 个订单 $\dots C_b$ 的第 j_b 个订单没有同时拣选。

若 C_1 的第 j_1 个订单、 C_2 的第 j_2 个订单 $\dots C_b$ 的第 j_b 个订单组合拣选,那么此组合所需要的第 k 种 SKU 出库料箱次数的计算方法为: b 个订单只要有一个需要第 k 个 SKU,第 k 个 SKU 出入库次数则为 1,用逻辑“或”符号表示为 $Z_{1j_1k} \parallel Z_{2j_2k} \parallel \dots \parallel Z_{bj_bk}$,简写为 $z_{j_1, j_2, \dots, j_b|k}$,值为 1 或 0。定义 $W_{C_1, C_2, \dots, C_b}$ 为同时拣选客户 $C_1, C_2, \dots, C_b$ 的货物节省出入库料箱的数量。0-1 整数规划模型为

$$\max W_{C_1, C_2, \dots, C_b} = \sum_{j=1}^N \sum_{k=1}^s (Z_{1j,k} + Z_{2j,k} + \dots + Z_{bj,k}) - \sum_{j_1=1}^N \sum_{j_2=1}^N \dots \sum_{j_b=1}^N \sum_{k=1}^s T_{j_1, j_2, \dots, j_b} z_{j_1, j_2, \dots, j_b|k}$$

s.t.

$$\sum_{j_2=1}^N \sum_{j_3=1}^N \dots \sum_{j_b=1}^N T_{j_1, j_2, \dots, j_b} = 1, j_1 = 1, 2, \dots, N \quad (1)$$

$$\sum_{j_1=1}^N \sum_{j_3=1}^N \dots \sum_{j_b=1}^N T_{j_1, j_2, \dots, j_b} = 1, j_2 = 1, 2, \dots, N \quad (2)$$

...

$$\sum_{j_1=1}^N \sum_{j_2=1}^N \dots \sum_{j_{b-1}=1}^N T_{j_1, j_2, \dots, j_b} = 1, j_b = 1, 2, \dots, N \quad (3)$$

式(1)~(3)分别保证客户 C_1 到 C_b 中某一订单只能和其他客户的订单组合一次。

2 客户分批模型

设该“货到人”分拣系统某一时刻共有 B 个客户的订单。每个拣选位置可同时分配 b 个客户,所以每批有 b 个客户,则总批次数 $p = \lfloor B/b \rfloor$,批次 p (最后一批)的客户数量若小于 b ,可用零矩阵作为客户订单补齐,根据实际情况, $B > b$ 。系统包含多个拣选位置,此 p 个批次依次适时分配给不同拣选位置,由于各个拣选位置相互独立,所以将既定的批次分配给拣选位置的过程不影响系统效率,不作考虑。假设每个料箱只有一种 SKU 的货物,而且货物

数量足够。

根据订单排序模型计算 $W_{C_1, C_2, \dots, C_b}$,其中 $C_1, C_2, \dots, C_b \in \{1, 2, \dots, B\}$ 。因为同一批次客户的排序对目标函数没有影响,即 $W_{C_1, C_2, \dots, C_b} = W_{C_2, C_1, \dots, C_b} = \dots$,为了减少计算量,只需计算 $C_1 < C_2 < \dots < C_b$ 时的 $W_{C_1, C_2, \dots, C_b}$,否则令 $W_{C_1, C_2, \dots, C_b} = 0$ 。

设 $X_{C_1, C_2, \dots, C_b} = 1$ 表示编号为 $C_1, C_2, \dots, C_b$ 的客户作为一批同时拣选,反之 $X_{C_1, C_2, \dots, C_b} = 0$,其中 $C_1, C_2, \dots, C_b \in \{1, 2, \dots, B\}$ 。0-1 整数规划模型为

$$\max \sum_{C_1=1}^B \sum_{C_2=1}^B \dots \sum_{C_b=1}^B X_{C_1, C_2, \dots, C_b} W_{C_1, C_2, \dots, C_b}$$

s.t.

$$\sum_{j_2=1}^B \sum_{j_3=1}^B \dots \sum_{j_b=1}^B X_{j_1, j_2, \dots, j_b} + X_{j_2, j_3, \dots, j_b} + \dots + X_{j_2, j_3, \dots, j} = 1,$$

$$j = 1, 2, \dots, B \quad (4)$$

$$X_{C_1, C_2, \dots, C_b} = 0, C_1, C_2, \dots, C_b \in \{1, 2, \dots, B\},$$

$$\text{非 } C_1 < C_2 < \dots < C_b \quad (5)$$

式(4)保证一个客户在所有批次中有且只有一次。式(5)非 $C_1 < C_2 < \dots < C_b$ 条件避免一个批次中出现多个相同的客户,也避免出现因客户顺序不同而导致批次重复。

3 种子算法

本节参考文献[4]提出了基于节约矩阵的种子算法。该算法首先根据第一节的订单排序模型求出所有客户两两组合拣选的最大节约出入库料箱数量,构建出入库节约料箱矩阵(简称节约矩阵)

$$\begin{pmatrix} W_{C_1, C_1} & \dots & W_{C_1, C_B} \\ \vdots & & \vdots \\ W_{C_B, C_1} & \dots & W_{C_B, C_B} \end{pmatrix}$$

该矩阵可用来衡量客户之间的抽象距离,选出种子客户和与种子客户距离近的客户。算法分为定义阶段和分批阶段。

3.1 定义阶段

定义 β_i 表示第 i 批已经分配的客户数量,集合 A_i 表示第 i 批已经分配的客户, θ 表示应分配的批次号, η 表示未分配的客户总数。初始时刻 $\beta_i = 0$, $A_i = \phi$, $1 \leq i \leq p$; $\theta = 1$, $\eta = B$ 。计算节约矩阵。

3.2 分批阶段

(1) 如果 $\theta = p + 1$, 结束分配过程。否则找出节

约矩阵中最大值, 假设为 W_{C_i, C_j} , 其对应的客户为 C_i 和 C_j , 将这两个客户分配给批次 θ , 记 $\beta_\theta = 2$, $\eta = \eta - 2$, $A_\theta = A_\theta \cup \{C_i, C_j\}$, 转到步骤(2)。

(2) 为批次 θ 分配其他客户。若 $\eta \leq b - 2$, 则将该 η 个客户分配给批次 θ , 记 $\beta_\theta = 2 + \eta$, $\eta = 0$, 结束分配过程。若 $\eta > b - 2$, 转到步骤(3)。

(3) 若 $\beta_\theta = b$, $\theta = \theta + 1$, 然后转到步骤(1); 否则, 在节约矩阵中找出未被分批的 η 个客户和 A_θ 中客户节约出入库料箱数量最大的客户, 记为 C_i , 将 C_i 分配给批次 θ , $\beta_\theta = \beta_\theta + 1$, $\eta = \eta - 1$, $A_\theta = A_\theta \cup \{C_i\}$, 循环步骤(3)。

4 遗传算法

遗传算法(Genetic algorithm, GA)是解决组合优化问题的经典算法, 针对考虑客户收货顺序的客户分批问题, 本文重新定义了编码、初始解、交叉和变异等方法。

4.1 编码

定义客户编号为算法的基因, 即 $C_1, C_2, \dots, C_B$; 客户编号的组合(一批)为算法的染色体, 例如 $[C_1, C_2, \dots, C_b]$; 所有染色体组成一个个体, 为该问题的一个解, 例如 $\{[C_1, C_2, \dots, C_b], [C_{b+1}, C_{b+2}, \dots, C_{2b}], \dots, [C_{b(p-1)+1}, C_{b(p-1)+2}, \dots, C_B]\}$ 。

4.2 种群

设种群的大小为 Q , 则该算法的种群由 Q 个个体构成。种子算法得到的解作为初始解, 对应的个体为初始个体。由 Q 个初始个体构成了初始种群。

4.3 交叉

本文算法交叉是指个体之间的操作, 随机选取多组个体组成交叉对, 分别做交叉处理, 目的是保留适应度较强的染色体, 具体步骤如下。

(1) 确定交叉系数 μ_1 , 交叉的染色体数为 $\gamma_1 = \mu_1 P$ 。随机生成 γ_1 个不大于 P 的不重复整数作为交叉染色体的序号, 用 δ_i 表示, $i \leq \gamma_1$ 。对于交叉对中两个体分别对应的第 δ_i 个染色体, 用步骤(2)处理。

(2) 设两个个体 x 和 y 对应的第 δ_i 个染色体分别为 $[C_x^1, C_x^2, \dots, C_x^b]$ 和 $[C_y^1, C_y^2, \dots, C_y^b]$, 交换两染色体。然后在个体 x 中找到染色体 $[C_y^1, C_y^2, \dots, C_y^b]$ 中的每个基因, 分别用 $C_x^1, C_x^2, \dots, C_x^b$ 依次代替; 在个体 y 中找到染色体 $[C_x^1, C_x^2, \dots, C_x^b]$ 中的每个基因, 分别用 $C_y^1, C_y^2, \dots, C_y^b$ 依次代替。

4.4 变异

本文算法变异是指个体内部, 多个染色体的基因的变异, 目的是创造新的染色体。具体步骤如下。

(1) 确定染色体变异系数 μ_2 , 每个个体变异染色体数为 $\gamma_2 = \mu_2 P$; 确定基因变异系数 μ_3 , 每条染色体变异基因数为 $\gamma_3 = \mu_3 b$ 。随机生成 γ_2 个不大于 P 的不重复整数作为变异染色体的序号, 随机生成 γ_3 个不大于 b 的不重复整数作为变异染色体中变异基因的序号, 对于每个变异染色体的变异基因, 总数为 $\gamma_2 \times \gamma_3$, 用步骤(2)处理。

(2) 随机生成一不大于 B 的整数, 假设为 i , 用 C_i 替换变异基因; 然后用变异基因替换个体中的基因 C_i 。

4.5 适应度函数

适应度函数如下所示。

适应度 =

$$\frac{\text{当前个体目标值} - \text{当前种群中最小目标值}}{\text{当前种群中最大目标值} - \text{当前种群中最小目标值}}$$

4.6 遗传算法主要流程

(1) 令 $g = 1$, 设定迭代次数 M , 交叉系数 μ_1 , 染色体变异系数 μ_2 , 基因变异系数 μ_3 , 种群大小 Q , 初始化种群 P_g 。

(2) 若 $g \leq M$, 执行步骤(3)到(5); 否则输出目标函数值, 算法结束。

(3) 随机选取 $\lceil Q/4 \rceil \times 2$ 个个体, 共 $\lceil Q/4 \rceil$ 对作交叉处理; 其他 $Q - \lceil Q/4 \rceil \times 2$ 个个体做变异处理。

(4) 计算新生成的 Q 个个体和上一代所有个体的适应度, 用轮盘赌的方式决定是否保留该个体, 假设最后保留的个体数为 Q' 。

(5) 若 $Q' < Q - 1$, 用最优个体补足种群到个体数量为 Q ; 否则截取适应度最大的 $Q - 1$ 个个体, 然后加上最优个体构成新的种群。 $g = g + 1$, 返回步骤(2)。

5 试验分析

本文设计了衡量订单排序模型和客户分批算法性能的试验, 包括 0-1 整数规划模型求解订单顺序, 0-1 整数规划模型、种子算法和遗传算法求解客户分批试验。所有试验在 Intel(R) Core(TM) i5-3210M CPU @2.50GHz 双核处理器 8.00G 内存的计算机上试验, 0-1 整数规划模型利用 Gurobi 求解, 种子算法和遗传算法在软件 Matlab 2015A 上完成。

本文根据实际作业情况, 设定一个拣选位置可以同时拣选 4 个客户的订单, 分别选取客户数量为 32、36、40、44、48、52; 每个客户最大订单(Purchase order, 简称 PO)数量为 2、4、6; 总 SKU 种类数为

5、10、15；共 $6 \times 3 \times 3 = 54$ 种试验情景，每种试验情景设计 20 个试验案例，试验数据取平均值。

试验分为以下两个部分。

第一部分试验验证了订单排序模型的效果。对于所有的情景，模拟了订单排序随机、客户分批随机的出入库数量和运用 0-1 整数规划模型优化订单排序、客户分批随机的出入库数量，对比结果如图 1 所示，横轴 B 和 PO 代表不同客户数量和客户最大订单数量的试验情景(图 2~7 相同)，得出模型的优化效果随客户数量的增多而增加，SKU 种类数量越少优化效果越好，平均优化效果约 15%，最高约 25.4%。另外，由图 2 得，该模型的执行时间随客户数量增多而增长，但是所有试验情境该模型的执行时间均未超过 0.8 s，执行时间较短，所以该模型具有有效性和实用性。

图 1 订单排序优化效果

图 2 订单排序模型执行时间

第二部分试验衡量 3 种客户分批算法的性能，统计了 0-1 整数规划模型、种子算法、遗传算法分批相对随机分批的优化比例，3 种算法和随机分批都在分批之后对所有批次进行了订单排序优化。试验记录了 3 种算法的总执行时间(All time, AT, 单位 s)、目标函数值，0-1 整数规划模型执行时间还记录了遍历客户分批后订单排序用时(Sequence time, ST, 单位 s)，即求所有 $W_{C_1, C_2, \dots, C_b}$ 所用的时间，其中 $C_1, C_2, \dots, C_b \in \{1, 2, \dots, B\}$ 。遗传算法参数取值如下： $M=100$ ， $\mu_1=0.3$ ， $\mu_2=0.5$ ， $\mu_3=0.5$ ， $Q=100$ 。

表 1 记录了各个情景下 0-1 整数规划模型的 AT 和 ST，通过对比试验得出以下几点。

表 1 0-1 整数规划模型 AT 和 ST 对比表

订单	数量	品项(SKU) 数量	客户数量					
			32	36	40	44	48	52
ST	2	5	152.83	245.95	387.25	581.10	831.74	1 183.43
AT			245.70	395.27	616.49	962.44	1 410.31	2 128.77
ST		10	180.53	295.45	426.80	648.46	920.11	1 314.59
AT			273.68	446.21	658.49	1 003.77	1 506.32	2 181.37
ST		15	196.27	287.97	418.49	614.62	903.61	1 304.76
AT			303.64	435.41	643.75	964.23	1 427.70	2 087.35
ST	4	5	901.16	1 464.50	2 272.05	3 075.18	4 360.00	6 210.29
AT			989.23	1 608.07	2 493.80	3 418.32	4 892.65	7 069.10
ST		10	994.85	1 467.29	2 245.27	3 399.97	4 831.07	6 772.04
AT			1 084.36	1 613.43	2 465.57	3 754.72	5 360.20	7 573.29
ST		15	1 076.09	1 715.88	2 689.17	3 985.44	5 453.11	7 324.23
AT			1 168.93	1 866.06	2 919.00	4 367.29	6 028.08	7 832.22
ST	6	5	3 360.84	5 831.43	8 362.48	12 377.54	18 017.43	26 210.17
AT			3 434.23	5 940.58	8 559.09	12 641.26	18 541.34	26 980.49
ST		10	3 552.82	5 949.89	8 504.49	12 488.32	18 217.50	26 321.84
AT			3 631.27	6 079.61	8 704.09	12 776.56	18 730.08	27 092.40
ST		15	4 231.66	6 918.52	10 735.85	15 585.41	22 604.80	30 234.21
AT			4 325.73	7 072.62	10 966.76	15 934.03	23 160.92	33 212.22

(1) 总执行时间随客户数量和客户订单数量的增加而急速增长, 原因是客户数量或客户订单数量增加, 令遍历不同客户组合时订单最优排序用时急速增长, 该时间占总时间 55.59%~98.16%, 客户数量或客户订单数量越大, 所占比重越大。

(2) 总执行时间和 SKU 种类数关系不大。

(3) 若客户数量较大或者客户订单数量较大时, 该模型执行时间太长, 实用性大幅下降。

图 3 显示了 0-1 整数规划模型的优化效果, 因为该模型得到的解为最优解, 所以优化百分比即问题的优化空间, 由图可得该算法优化空间随着问题复杂度的提高而减小, 比如客户最大订单数量增加或者 SKU 种类数增多都会使问题优化空间减少, 平均优化空间为 16.39%, 客户总数量的变化对结果影响不大。

图 3 客户分批 0-1 整数规划模型优化效果图

对于种子算法, 通过图 4 可得种子算法对该问题的优化效果不稳定, 基本保持在 4%~7%, 平均为 6%, 问题的规模对优化效果影响不明显。图 5 显示了各个情景下种子算法的执行时间, 显然随客户数量增多而快速增长, 原因同 0-1 整数规划模型相同, 客户增多, 计算两两客户之间的节约矩阵用时大量增加, 不过由于两两客户的组合数量较少, 执行时间相对较短, 平均为 17.11 s。

图 5 种子算法执行时间

对于遗传算法, 通过图 6 可知, GA 的优化效果优于种子算法, 优化效果受总 SKU 数量和客户订单数量的影响, SKU 数量和客户订单数量越大优化效果越差。5 个 SKU 的情景可平均优化约 14.53%, 10 个 SKU 的情景可平均优化约 10.23%, 15 个 SKU 的情景可平均优化约 8.55%, 总平均为 11.10%。图 7 可得客户订单和客户数量越大 GA 执行时间越长, 与 SKU 种类数关系不大。

图 6 遗传算法客户分批优化效果

图7 遗传算法执行时间

总之,用0-1整数规划模型可以得到最优解,但是问题较为复杂,耗时较长;种子算法运行快速,但是效果一般;遗传算法优化效果优于种子算法,执行时间可以接受。另外,遗传算法可调节参数多,具有控制执行时间和优化效果的灵活性。

6 结论

考虑了客户按订单顺序收货的“货到人”分拣系统的订单排序和客户分批问题。建立了0-1整数规划模型求解多客户同时拣选时的各个客户的最优订单排序,提出0-1整数规划模型、种子算法、遗传算法解决客户分批优化问题,通过试验得到以下结论。

(1) 0-1整数规划模型可以有效解决多个同时拣选的客户的订单排序问题,得到最优解。

(2) 0-1整数规划模型解决小规模的客户分批问题效果较好,可以得到最优解,但是问题规模增大时模型执行时间增长迅速,时间成本较高,实用性不佳。

(3) 种子算法进行客户分批,优化效果一般,但是执行时间最短,适用于对时间要求高的情景。

(4) 遗传算法解决的客户分批问题效果优于种子算法,可以通过调节参数,控制执行时间和优化效果,具有较好的灵活性。

客户要求收货顺序的分销中心,在使用“货到人”分拣系统时,可以根据实际的应用场景选择合适的算法进行客户分批优化,然后利用0-1整数规划模型对每个批次客户的订单进行排序,

以提高系统的分拣效率,故本文对“货到人”分拣系统中分拣策略的设计和优化有较高的指导意义。

参考文献

- [1] CERGIBOZAN Ç, TASAN A S. Order batching operations: An overview of classification, solution techniques, and future research[J]. Journal of Intelligent Manufacturing, 2019, 30(1): 335-349.
- [2] MEHRJERDI Y Z, ALIPOUR M, MOSTAFAEIPOUR A. Integrated order batching and distribution scheduling in a single-block order picking warehouse considering s-shape routing policy[J]. International Journal of Engineering, 2018, 31(10): 1723-1733.
- [3] VAN GILS T, RAMAEKERS K, BRAEKERS K, et al. Increasing order picking efficiency by integrating storage, batching, zone picking, and routing policy decisions[J]. International Journal of Production Economics, 2018, 197, 243-261.
- [4] 张琨. 电子商务环境下并行分区拣选系统的订单合并优化研究[D]. 大连: 大连理工大学, 2014.
ZHANG Jun. Study on order batching strategy optimization under e-commerce synchronized zone order picking system[D]. Dalian: Dalian University of Technology, 2014.
- [5] 吴颖颖, 孟祥旭, 王艳艳, 等. “货到人”拣选系统订单排序优化[J]. 机械工程学报, 2016, 52(4): 206-212.
WU Yingying, MENG Xiangxu, WANG Yanyan, et al. Order sequence optimization for “part-to-picker” order picking system[J]. Journal of Mechanical Engineering, 2016, 52(4): 206-212.
- [6] 胡金昌. 多层穿梭车分拣系统的订单分配优化[D]. 济南: 山东大学, 2016.
HU Jinchang. Order allocating optimization for multi-shuttle picking system[D]. Jinan: Shandong University, 2016.
- [7] 赵鑫亮. 并行自动分拣系统订单排序优化研究[D]. 济南: 山东大学, 2017.
ZHAO Xinliang. Study on order sequence optimization for parallel automated picking system[D]. Jinan: Shandong University, 2017.

- [8] 吴思沛. 面向制造业 B2C 模式的多层穿梭车自动分拣系统设计与订单分配[D]. 济南: 山东大学, 2018.
- WU Sipei. Design of Automatic sorting multi-layer shuttle system and order allocating optimization for manufacturing B2C mode[D]. Jinan: Shandong University, 2018.
- [9] XIANG X, LIU C, MIAO L. Storage assignment and order batching problem in Kiva mobile fulfilment system[J]. *Engineering Optimization*, 2018, 50(11): 1941-1962.
- [10] NICOLAS L, YANNICK F, RAMZI H. Order batching in an automated warehouse with several vertical lift modules: Optimization and experiments with real data[J]. *European Journal of Operational Research*, 2018, 267(3): 958-976.
- [11] ZIVANIC D, ZELIC A, LALIC B, et al. Improving the order picking efficiency by optimising the orders' sequence[J]. *International Journal of Simulation Modelling*, 2019, 18(1): 125-137.
-
- 作者简介: 胡金昌, 男, 1992 年出生, 博士研究生。主要研究方向为生产调度和物流系统优化。
E-mail: 201620367@mail.sdu.edu.cn
- 马文凯, 男, 1991 年出生, 博士研究生。主要研究方向为自动拣选系统、物流系统规划、设计和仿真等。
E-mail: kevin.ma@mail.sdu.edu.cn
- 杨栋, 男, 1988 年出生, 博士研究生。主要研究方向为智能物流装备设计与优化、仓储拣选系统优化等。
E-mail: yangdong.lv@foxmail.com
- 吴耀华(通信作者), 男, 1963 年出生, 博士, 教授。主要研究方向为自动拣选和仓储系统设计和优化等。
E-mail: mike.wu@263.net