

基于NSGA-II算法的分区拣选优化

魏庆琦^{1,2}, 陈金迪¹

(1. 重庆交通大学 经济与管理学院, 重庆 400074; 2. 电子科技大学 经济与管理学院, 四川 成都 610054)

摘要: 电商背景下的客户订单呈现出多品种、小批量、高频次等特点, 给仓库拣选工作带来很大的挑战。为提高拣选效率, 在订单完全拆分的分批策略和组合优化的行走策略下, 设计了以总服务时间最小、分区工作量平衡度最优和二次分拣效率最高的多目标分区拣选模型。由于3个目标函数之间存在矛盾, 设计了NSGA-II算法对多目标优化模型进行求解。通过数值实验, 与传统的不拆分订单的分区拣选系统对比, 发现在订单批量环境为[1, 4]时, 分别使总服务时间减少了43.88%, 平衡度改善了84.61%, 并分析了区域个数、订单总数和订单批量环境对系统效率的影响。

关键词: 电商物流; 拣选系统效率; 订单完全拆分; 拣选分批; 分区拣选

中图分类号: C939; U16

文献标志码: A

文章编号: 1007-7375(2020)03-0001-09

A Research on Zone Picking Optimization Problem Based on NSGA-II Algorithm

WEI Qingqi^{1,2}, CHEN Jindi¹

(1. School of Economics and Management, Chongqing Jiaotong University, Chongqing 400074, China;

2. School of Management and Economics, University of Electronic Science and Technology of China, Chengdu 610054, China)

Abstract: Under the background of e-commerce, customer orders show the characteristics of multiple varieties, small batch, high frequency and so on, which brings great challenges to the warehouse picking work. In order to improve the efficiency of picking, a multi-objective zone picking model that minimizes the total service time, optimal zone workload balance and achieve the highest secondary sorting efficiency is designed under the batching strategy of the complete splitting of the order and walking strategy of the combinatorial optimization. Due to the contradiction among the three objective functions, the nondominated sorting genetic algorithm II (NSGA-II) is designed to solve the multi-objective optimization model. Through numerical experiments, it is found that when the order batch environment is [1,4], the total service time is reduced by 43.88%, the balance is improved by 84.61%, respectively, compared with the traditional partition picking system without splitting orders. The influence of the number of zones, the total number of orders and the order batch environment on the system efficiency is analyzed.

Key words: e-commerce logistics; picking system efficiency; complete order splitting; picking and batching; zone picking

订单拣选是指以最短的时间在仓库中收集一组商品的问题。由于高额的时间成本和劳动力成本, 订单拣选已成为当前电商供应链的主要瓶颈^[1]。在电子商务环境下, 由于商品种类繁多, 采用分区拣选策略进行仓库拣选操作管理可以有效提高订单拣选效率^[2]。分区拣选是指将整个拣选区域划分为几个部分, 每个拣选人员在其所负责的区域进行商品

拣选操作^[3]。根据订单处理的流程, 分区拣选分为串行分区拣选和并行分区拣选: 串行分区拣选是指一个订单同时只能在一个区域拣选; 并行分区拣选是指一个订单同时可在多个区域拣选。采用分区拣选策略时, 每个拣选人员负责一个区域, 不存在通道拥堵, 且员工对这个区域的商品熟悉, 从而减少了寻找商品的时间, 提高拣选效率。

收稿日期: 2019-06-03

基金项目: 国家自然科学基金资助项目(71471024; 71702016); 教育部人文社科研究青年基金资助项目(20YJC630156); 2020年重庆市教育委员会人文社会科学研究规划资助项目(20SKGH080)

作者简介: 魏庆琦(1982-), 女, 重庆市人, 教授, 博士, 主要研究方向为物流系统优化与仿真等。

目前,对拣选系统优化研究主要集中于拣选分批(order batching)^[4]、拣选路径优化(picker routing)^[5]、货位分配(storage assignment)^[6]和分区拣选(zone picking)^[2]4个方面。在并行分区系统中,订单分批和分区拣选是影响并行分区拣选效率的2个重要因素^[7]。有部分学者单独针对订单合并策略和求解算法进行研究。例如,Žulj等^[8]针对较大的客户订单数量和拣选设备容量的实例,提出了一种自适应大领域搜索和禁忌搜索的启发式算法(adaptive large neighborhood search/tabu search, ALNS/TS)来解决订单分批问题。王旭坪等^[9]为了减少延迟订单数量,构建考虑订单完成期限的在线订单分批混合整数规划模型,采用改进的固定时间窗订单分批启发式规则求解模型。也有部分学者单独对分区拣选进行研究。Ho等^[10]提出了一种不同于传统线性分区拣选的网络分区拣选结构,并建立了从简单到复杂的路由调度规则。同时,也有学者将订单分批与区域拣选合并研究。Jane等^[3]针对并行分区拣选系统的订单分批问题,为了平衡各区拣选人员的工作量,减少工作人员的闲置时间,提出了根据订单关联度的聚类算法。Yu等^[7]提出了一种基于排队网络理论的近似模型,用于分析人到货的拣选系统中,订单分批和分区拣选对订单平均处理时间的影响,但只适用于区域数较少的仓库。

在采用订单分批和分区拣选策略时,需要增加订单二次分拣操作才能进行打包发货,针对拣选完成后的分拣包装的研究较少。De Koster等^[11]认为拣选区域的数量增加,会减少拣选时间,但同时会增加分拣包装的时间,通过确定拣选区域的个数,使总时间(拣选和分拣)最小。Jiang等^[12]针对订单拣选与分拣包装2个进程中间的有限缓冲区域的拥堵问题,提出改进的种子算法来解决订单分批和拣选批次排序问题。

上述文献针对订单分批、分区拣选和二次分拣作了一定研究,但还存在以下问题。1) 关于拣选分批的研究都集中于订单之间合并分批,未将订单完全拆分。订单拆分是一个有效提高拣选效率的拣选策略,但需要二次分拣。而在分区拣选系统中,同样需要进行二次分拣。因此本文考虑订单完全拆分的拣选分批策略。2) 在研究分区拣选系统时,考虑总拣选时间和各区拣选任务的均衡度来进行订单分批,但忽略了分区拣选系统中订单分批对分拣包装

的影响。3) 优化分区拣选的目标之间相互不协调,甚至相互矛盾,但很少文献采用解决相互冲突的多目标求解算法。针对这些问题,本文在订单完全拆分基础上,构建考虑以总服务时间最小、分区工作量平衡度最优和二次分拣效率最高的多目标模型,并运用NSGA-II(non dominated sorting genetic algorithm-II)算法求解。

1 考虑完全拆分的并行分区合并优化模型

1.1 问题描述与模型构建

本文研究人工并行分区拣选系统,通过将订单完全拆分后合并拣选单的策略来提高分区拣选效率。系统的流程如图1所示。针对某一时间段到达的订单集合 J ,将其拆分为商品集合 I ,在考虑拣选设备容量限制的前提下,通过考虑总服务时间最小、分区工作量平衡度最优和二次分拣效率最高生成 Z 组拣选单,并将其分配给各区的拣选人员进行拣选操作,从存储区域拣选出来的商品需要通过传送带送达分拣区域进行二次分拣操作,最后进行打包发货。由此可见,整个系统的订单处理时间由订单拣选时间和订单二次分拣时间构成,因此总服务时间最小是指订单拣选时间与分拣时间之和最小。

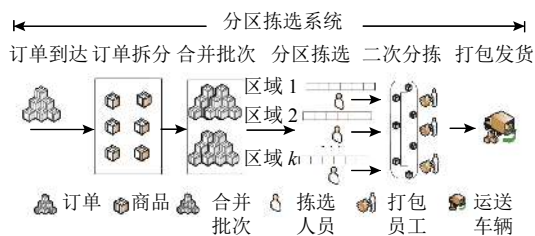


图1 分区拣选系统流程图

Figure 1 Flow chart of zone picking system

本文假设每个拣选人员负责一个区域,且一个批次同时在多个区域拣选。由于并行分区拣选要求所有区域都完成该批次的拣选工作后,才能进行下一批次的拣选作业,因此完成每个拣选批次的服务时间与各拣选区域的拣选时间关系分析如图2所示。每个批次的服务时间指最晚完成该批次拣选任务的区域的拣选时间。先完成拣选任务的区域需要等待,存在空闲时间。由此可见,为了提高分区拣选效率,减少拣选人员的空闲时间,提高拣选人员工

作量的公平性, 需要平衡各区工作量。虽然采用分区拣选策略和订单完全拆分的分批策略会减少订单拣选的时间, 但同时会增加订单二次分拣的处理时间, 从而降低二次分拣的效率。

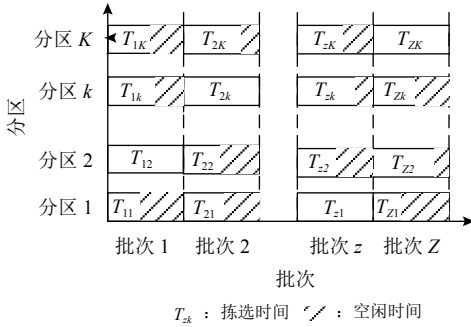


图2 并行分区拣选时间序列图

Figure 2 Parallel zone picking time series diagram

通过以上分析, 模型的假设条件如下。

1) 仓库布局如图3所示。拣选区域分为 k 个区域, 每个区域的商品拣选难度相同, 因为电子商务销售的商品种类繁多, 有部分商家的商品包装后的体积重量差别不大, 例如零食、化妆品等。这类商品在不同区域的拣选作业差别很小, 可以忽略不计。且各区域布局与面积相同, 入口在各区域的左下端, 二次分拣区域在拣选区域右侧。

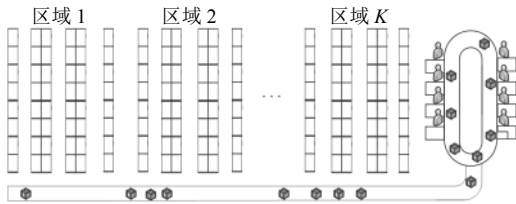


图3 仓库布局图

Figure 3 Warehouse layout

2) 每个拣选区域由一个拣选人员负责, 且拣选人员熟悉拣选任务单中商品位置。

3) 每件商品的重量体积相同, 每个货位存储一种商品, 不缺货。

4) 拣选人员行走在通道中央, 同时从左右两边取货, 因此员工服务时间只考虑行走时间和取货时间。

5) 各分区的拣选员工完成拣选任务后返回入口, 并将商品全部放到传送带上, 等待新的任务。

6) 每个区域的拣选任务单的商品数量不得超过拣选设备的容量限制。

7) 采用组合型的拣选路径策略, 即综合遍历型

与返回型策略, 拣选员根据手持指示设备, 按规定路径完成拣选。

8) 二次分拣设备上的分拣台只能完成当前订单分拣任务后, 才能进行下一个订单的二次分拣。

1.2 模型参数及变量

I 为商品集合, $i \in I$; J 为订单集合, $j \in J$; Z 为拣选批次集合, $z \in Z$; K 为拣选区域的集合, $k \in K$; w 为分拣台的集合, $w \in W$; Q 为拣选车最大容量; a 为货架格子长度; b 为货架格子宽度; c 为通道的宽度; m 为每排货架的格子数; H_i 、 L_i 和 P_i 分别为商品 i 所在货架的行号、列号, 以及左边第一个通道编号; V_{travel} 为拣选员行走速度; V_{pick} 为拣选员取货速度; V_1 为拣选区域到分拣区域的传送带的速度; V_2 为二次分拣设备的传送带的速度; N_j 为订单 j 的商品数量; T_{zk} 为批次 z 在区域 k 的拣选时间; T_{max}^z 和 T_{min}^z 分别为批次 z 中所有拣选区域完成拣选任务的最大时间和最小时间; T_{ij} 为订单 j 中商品 i 的拣选完成时间; T'_{ij} 为订单 j 中商品 i 到达二次分拣设备的时间; T_j^{max} 和 T_j^{min} 分别为订单 j 的所有商品最晚到达二次分拣设备的时间和最早到达的时间; T''_{ij} 为订单 j 中商品 i 完成二次分拣的时间, T_{end}^j 为订单 j 完成二次分拣时间; T_{end} 为所有商品完成二次分拣的时间。

X_{ik} 为商品 i 是否分配给批次 z , 1表示分配, 0表示未分配; Y_{ij}^p 为订单 j 中商品 i 是否进入分拣台 p , 1表示进入了分拣台, 0表示未进入。

1.3 模型构建

在分区拣选系统中, 将订单完全拆分, 通过考虑总服务时间最小、分区工作量平衡度最优和二次分拣效率最高3个目标建立拣选分批模型。其中, 考虑总服务时间最小是指从订单需求出现到全部订单完成二次分拣的历时最小化。为了减少拣选人员的等待时间, 提高员工的公平性, 需要平衡各区工作量, 分区工作量平衡度最优是指所有批次的每个区域完成拣选的时间差总和的倒数最大。在二次分拣设备和分拣策略确定的情况下, 二次分拣效率由商品到达情况决定。商品到达情况是指订单中所有商品到达分拣设备的时间, 订单的等待时间是指该订单中的商品最早到达分拣设备的时间与最晚到达分拣设备的时间之差, 因此定义二次分拣效率最高为所有订单的等待时间之和的倒数最大。通过分析, 发现商品在整个仓库中分为3个重要的时间节点: 商品完成拣选的时间、商品到达二次分拣设备

的时间、商品完成二次分拣的时间。

1.3.1 商品完成拣选的时间

商品完成拣选的时间是指每个区域的拣选人员根据拣选单, 行走在其负责的区域, 并将拣选单中全部商品拣选出来所用的时间。因此批次中的每个区域所包含商品的完成拣选时间是相同的。拣选时间包括拣选人员的行走时间和取货时间。取货时间与货物的数量成正比关系, 行走时间与货物所在位置、行走速度和行走策略有关。常用的遍历型策略和返回型策略容易被员工接受, 出错较少, 但随着电商仓库的信息技术的发展, 拣选人员可以使用手持 RF (radio frequency) 终端或者车载 VC (virtual circuit) 终端来指导拣选人员的作业路径, 提高工作人员的拣选效率和准确性。因此, 本文采用组合型作为拣选人员的行走策略。返回型策略^[13]是指员工从通道的一端进入, 拣选完通道所有商品后返回进入通道的端口, 依次进行其余通道的拣选。遍历型策略^[13]是指拣选人员从通道的一端进入, 拣选完所有商品后从另一端离开, 依次进行通道的拣选。本文设计的组合型策略综合前面2种拣选路径策略, 根据采用何种行走策略去往下一通道的距离更短, 来决定此拣选通道的行走策略(图4)。

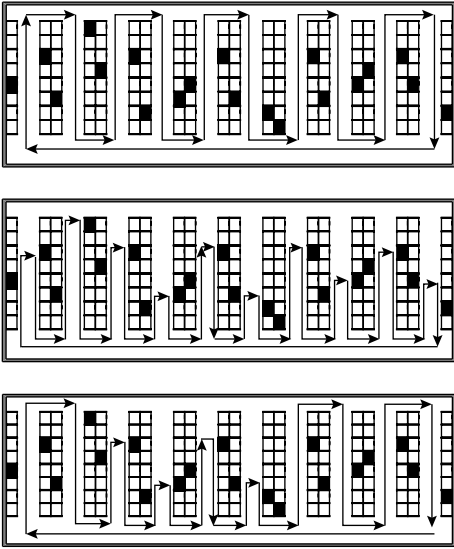


图 4 返回型、遍历型、组合型策略示意图

Figure 4 Strategy diagram of return-type, ergodic-type, and combination-type

根据以上说明, 拣选人员采用组合型策略作为拣选行走策略。因此, 拣选批次 z 的第 k 区域的行走距离 D_{zk} 的具体计算步骤如下。

Step 1 寻找批次 z 的第 k 区域中的最小货架、最小货位的商品, 记为 A , 计算起点到 A 的 D_{begin} 距离, 计算公式如下:

$$D_{\text{begin}} = cP_A + bH_A + aL_A, \quad (1)$$

并将距离记到总距离 D 中, 转到下一步。

Step 2 寻找此通道中货位号最大的商品记为 B , 计算 D_{AB} , 计算公式如下:

$$D_{AB} = a|L_A - L_B|. \quad (2)$$

把 D_{AB} 记入总距离 D 中, 并将 B 点标记为特殊点 S , 转到下一步。

Step 3 判断是否还有商品未拣选出, 若有, 找到所有未拣选出商品所在的最小通道, 记为下一通道; 若无, 转到Step6。

Step 4 分别采用返回型和遍历型策略计算 S 点到下一通道的距离, 寻找下一通道的最小货位号的商品记为 A , 最大货位号的商品记为 B , 分别计算 D_{SA} 和 D_{SB} 。

采用返回型策略计算 D_{SA} :

$$D_{SA} = c|P_A - P_S| + b|H_A - H_S| + a(L_A + L_S - 1). \quad (3)$$

采用遍历型策略计算 D_{SB} :

$$D_{SB} = c|P_B - P_S| + b|H_B - H_S| + a(2m - L_B - L_S + 1). \quad (4)$$

Step 5 判断 $D_{SA} \leq D_{SB}$, 若是, 将 B 记为 S 点, 并把 D_{SA} 、 D_{AB} 记入 D 中; 若否, 将 A 记为 S 点, 并把 D_{SB} 、 D_{AB} 记入 D 中, 返回Step3。

Step 6 计算回程距离 D_{end} 。

$$D_{\text{end}} = cP_S + bH_S + aL_S. \quad (5)$$

将 D_{end} 记入 D 中, 并将 D 值返回到 D_{zk} 中, 算法终止。

根据上面说明, 拣选批次 z 的第 k 个区域的拣选时间 T_{zk} 和其包含商品的拣出时间 T_{ij} 分别定义为

$$T_{zk} = D_{zk} / V_{\text{travel}} + I_{zk} V_{\text{pick}}, \quad (6)$$

$$T_{ij} = T_{zk}. \quad (7)$$

1.3.2 商品到二次分拣设备的时间

如仓库布局图3所示, 商品从拣选区域到分拣区域用传送带送达, 输送的距离与分拣区域的位置成正比, 且比例系数为每个拣选区域的宽度 M , 订单 j 的第 i 件商品到分拣设备的时间为

$$T'_{ij} = T_{ij} + \frac{(K - k_{ij})M}{V_l}. \quad (8)$$

其中, k_{ij} 为订单 j 的商品 i 所在的拣选区域。

1.3.3 商品完成二次分拣时间

影响订单二次分拣效率的因素主要有3个: 1) 分拣设备; 2) 分拣策略; 3) 商品到达情况。本文假设分拣设备为简单的环形分拣机, 商品随分拣机的传送带移动, 当到达指定的分拣台时, 将商品分拣出来。如何安排分拣台的分拣策略对订单分拣效率影响很大。常用的分拣策略有: 1) 下一个优先策略(next priority strategy), 即每一个空闲的拣选台会优先处理第一个到达该分拣台的未分拣的订单; 2) 固定优先级策略(fixed priority policy), 即根据事先定义好的规则赋予订单优先级, 订单根据优先级分配分拣台。本文采用第2种策略, 根据该时刻订单中到达分拣设备的商品的占比分配分拣台, 分拣策略的具体实现步骤如下。

Step 1 开始, 记time = 0。

Step 2 在time时刻记录到达分拣设备的商品和商品所在的订单, 选出未安排分拣台的订单, 并计算这些订单中到达分拣设备的个数在总商品数量中的占比。

Step 3 根据占比大的订单j优先安排分拣台, 并将订单j记为已安排。

Step 4 记录订单j的商品i到达分拣台的时间为

$$T''_{ij} = T'_{ij} + \frac{D_w^i}{V_2} \quad (9)$$

其中 D_w^i 表示从入口到订单j分配的分拣台w的流水线行走距离。

Step 5 等待某一拣选台完成其订单的分拣工作, 将结束时间赋值给time。

Step 6 判断是否还有未安排的订单, 若有, 返回Step2; 若无, 继续。

Step 7 等待未完成分拣工作的分拣台结束工作, 并记录订单中每件商品到达分拣台的时间, 算法终止。

1.3.4 考虑完全拆分的并行分区拣选模型

构建以总服务时间最小、分区工作量平衡度最优和二次分拣效率最高的多目标分区拣选模型如下。

目标函数为

$$\min \sum_{j \in J} T_j; \quad (10)$$

$$\max 1 / \sum_{z \in Z} (T_{\max}^z - T_{\min}^z); \quad (11)$$

$$\max 1 / \sum_{j \in J} (T_{ij}^{\max} - T_{ij}^{\min}), i \in I_j \circ \quad (12)$$

约束条件为

$$T_j = \max \{T''_{ij}\}, i \in I_j, j \in J; \quad (13)$$

$$T_{\max}^z = \max \{T_{z1}, T_{z2}, \dots, T_{zK}\}, z \in Z; \quad (14)$$

$$T_{\min}^z = \min \{T_{z1}, T_{z2}, \dots, T_{zK}\}, z \in Z; \quad (15)$$

$$T_{ij}^{\max} = \max \{T'_{ij}\}, i \in I_j, j \in J; \quad (16)$$

$$T_{ij}^{\min} = \min \{T'_{ij}\}, i \in I_j, j \in J; \quad (17)$$

$$\sum_{z \in Z} X_{iz} = 1; \quad (18)$$

$$\sum_{i \in I_{zk}} X_{iz} \leq Q, z \in Z, k \in K; \quad (19)$$

$$\sum_{p \in P} Y_{ij}^p = 1, i \in I_j, j \in J; \quad (20)$$

$$\sum_{i \in I_j} Y_{ij}^w = N_j, j \in J, w \in W. \quad (21)$$

其中, 式(10)为总服务时间最小; 式(11)为分区工作量平衡度最优; 式(12)为二次分拣效率最高; 式(13)为每个订单完成二次分拣的时间; 式(14)和式(15)分别为各批次中所有区域完成拣选任务的最慢时间和最快时间; 式(16)和式(17)分别为每个订单商品的最晚到达二次分拣设备的时间、最早到达分拣设备的时间; 式(18)保证每件商品只能分配给一个批次; 式(19)保证每个批次中每个区域的商品数量不会超过拣选设备的最大容量; 式(20)和式(21)保证二次分拣中, 同一订单的商品被同一分拣台拣出。

2 NSGA-II 算法应用

本文建立了以总服务时间最小、分区工作量平衡度最优和二次分拣效率最高的多目标分区拣选模型, 但3个目标之间存在冲突, 无法在改进其中一个目标的同时不消弱至少一个其他目标。这样的多目标问题通常存在一个解集。这些解之间无法比较优劣, 通常被称为Pareto最优解或非支配解。本文采用基于Pareto解的NSGA-II算法进行求解, 此算法在NSGA的基础上提出了快速非支配排序方法、拥挤距离比较和精英保留策略, 能降低计算次数, 提高解集质量^[14]。

2.1 编码方式

并行分区拣选系统中, 订单情况是随机未知的, 商品随机分布在拣选区域中。但每个区域的拣选设备有容量限制, 生成拣选批次时, 需要确定批次中属于每个拣选区域的商品数量是否有超过拣选车的容量限制。为解决这个问题, 本文设计了动态随机并行分区拣选问题的编码方式, 其染色体的构成如图5所示, 具体染色体编码做法如下。

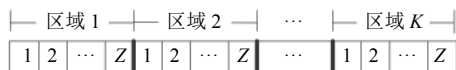


图 5 染色体编码示意图

Figure 5 Chromosome coding diagram

1) 采用自然数编码的方式, 将订单完全拆分, 每件商品代表每个染色体的一个基因。根据商品所在的区域, 将基因分为 K 类, 每类的基因进行全排列, 形成染色体的片段, 最后将所有染色体片段根据区域顺序组合为一个完整的染色体。

2) 由于拣选设备有容量限制 Q , 将每个拣选区域的染色体片段分为 $[L_k/Q]$ 批次部分, 除最后一个批次段的基因个数不为 Q 外, 其他每个批次段的基因个数都为 Q 。解码时, 每个批次的商品由各区段中属于该批次的基因段所对应的商品组成。若某个批次不能全部找到对应的批次段, 则该批次由能够找得到对应批次段组成。

2.2 遗传算子

由于本文的染色体分为 K 个区域段, 每个区域段中的基因不能相互交叉, 因此遗传算子中交叉、变异操作在染色体的每个区域段中进行, 即在父代染色体的每个区域段上都要进行一次交叉、变异操作。

1) 选择运算。本算法采用锦标赛方法选择交叉变异的种群。每次随机抽取2个个体, 如果非支配等级不同, 选择等级高的个体; 若等级相同, 选择拥挤距离大的个体。

2) 交叉运算。在交叉概率控制下进行交叉操作。由于个体的染色体分为 $[L_k/Q]$ 部分, 且每个部分之间不能相互交叉, 因此在每个部分进行一次交叉操作, 具体操作如下。

Step 1 随机选择2个交叉点。

Step 2 将2个交叉点中间的基因段互换。

Step 3 求2个染色体片段相互之间的补集, 找

到补集在该染色体片段的位置, 按序将补集放入另外一条染色体片段中。如图6所示。

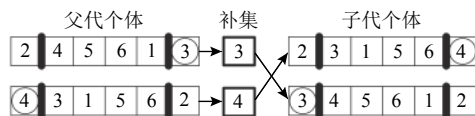


图 6 交叉操作示意图

Figure 6 Cross operation diagram

3) 变异操作。与交叉运算相同, 变异操作也在每个染色体片段内进行一次。在此染色体片段中随机寻找2个基因并交换。

3 数值实验与结果分析

3.1 实验参数

不同电商企业的仓库环境和订单量差别较大, 缺乏统一的数据标准。为说明模型和算法有效性, 采用文献[2]参数进行试算。整个仓库分为3个区域, 每个区域有6个货架, 每个货架有10个货位, 货架长度10 m、宽1 m; 有3个长10 m、宽1 m的通道; 拣选人员的行走速度为1 m/s, 取货速度为5 s/个, 拣选区域到分拣区域的传送带速度为1 m/s, 分拣机的传送带速度为0.5 m/s; 拣选设备的最大容量为10件。为与文献[2]对比, 设置了订单数量为30的3种订单批量环境, 分别为订单中商品数量满足[1,2]、[1,3]、[1,4]的随机均匀整数分布, 商品出现在订单中的概率相等, 这是考虑电商客户的订单小批量、多品种、高频次等特点, 客户的需求随机性大, 且不同商家的客户订单的差异巨大, 因此假设每件商品出现在订单的概率相同, 每次实验订单的相似度值差别大。NSGA-II算法的参数设置: 种群大小为500, 迭代500次, 交叉概率为0.9, 变异概率为0.15。

3.2 结果分析

3.2.1 对比分析

文献[2]构建了以总服务时间最小的单目标模型, 并用单目标遗传算法(single genetic algorithm, S-GA)求解; 同时构建了以总服务时间最小和各区工作量平衡度最优的双目标模型。并用双目标遗传算法(double genetic algorithm, D-GA)求解。将其得到的结果与本文的实验结果进行对比分析。设定拣选区域为3, 订单总数为30, 每种算法的求解结果如表1所示。

表 1 不同订单批量环境下3种系统实验结果表

Table 1 Experimental results of three systems in different order batch environments

订单环境	合并算法	完成时间	均衡度	二次分拣效率
批量[1, 2]	D-GA	145	0.20	0.013 7
	S-GA	139	0.08	0.004 0
	NSGA-II	134	0.25	0.005 6
批量[1, 3]	D-GA	204	0.20	0.019 6
	S-GA	202	0.10	0.009 7
	NSGA-II	198	0.20	0.002 4
批量[1, 4]	D-GA	376	0.04	0.005 9
	S-GA	353	0.02	0.003 6
	NSGA-II	211	0.25	0.002 1

由表1可知如下结果。1) 随着订单批量的增大, 每种算法的拣选完成时间明显增多。在订单批量环境为[1, 2]和[1, 3]中, NSGA-II 的拣选完成时间有小幅减少, 而在批量环境[1, 4]中, NSGA-II 明显优于D-GA和S-GA, 使拣选完成时间分别减少了43.88%和40.23%。2) 在均衡度方面, 随着批量的增大, 均衡度表现变差。其中, S-GA表现最差, 在

批量[1, 4]中, NSGA-II 明显改善了各区工作量的平衡度问题, 相对于D-GA与S-GA, 平衡度改善了84.61%和91.67%。3) 关于订单二次分拣效率, 本文算法所得结果低于另外2种算法的结果。

综上所述, 对比D-GA和S-GA, NSGA-II 在所有批量环境中, 虽然会增加二次分拣的难度, 但减少了系统总的服务时间, 同时也大大提高了分区工作量平衡度。因此, 本文设计的分区拣选系统优于传统不拆分订单的分区拣选系统。

3.2.2 区域个数对系统效率的影响

在订单总数为30、批量环境为[1, 4]的情况下, 改变拣选区域个数对系统效率的影响如图7所示。从拣选区域个数为3开始, 随着区域个数的增加, 分拣效率小幅度波动。拣选区域个数从2增加到4时, 拣选完成时间缩短了51.03%, 均衡度改善了55.56%。区域个数从4增加到6时, 拣选完成时间只缩短了0.60%, 均衡度却变差了6.72倍。每增加一个拣选区域, 就需要增加一个拣选人员, 会增加雇佣成本。因此, 综合考虑, 在订单总数为30和批量环境为[1, 4]的情况下, 4个拣选区域最合理。

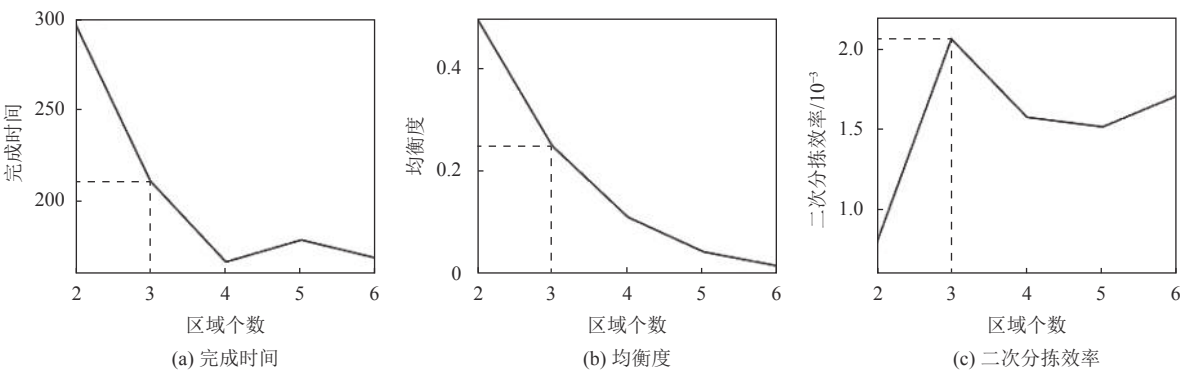


图 7 区域个数对系统效率的影响

Figure 7 The effect of number of zones on system efficiency

3.2.3 订单总数对系统效率的影响

在拣选区域个数为3, 订单批量为[1, 4]不变的前提下, 随着订单总数在10~80之间逐渐增加, 系统效率发生了显著变化, 结果如图8所示。

1) 完成时间与订单总数成正比, 除订单总数为10和15的特殊情况外, 平均订单完成时间呈现小幅度波动。

2) 均衡度先增加后减小, 订单数量为[30, 60]时呈现较稳定状态。

3) 订单分拣效率增加, 平均订单等待时间增加。

由此可见, 订单量的增加会增大订单分拣的难度。

通过以上分析可以看出, 当订单批量为[1, 4]和区域个数为3的情况下, 本文设计模型更适用于订单数量为[30, 40]的情况。订单数量过少, 均衡度表现不好; 订单数量过多, 不但均衡度表现差, 订单分拣难度也会大大增加。

3.2.4 订单批量对系统效率的影响

在订单总数为30, 区域个数为3的前提下, 订单的批量环境从[1, 2]到[1, 12]之间逐渐增加, 系统效率的变化如图9所示。随着订单批量的增加,

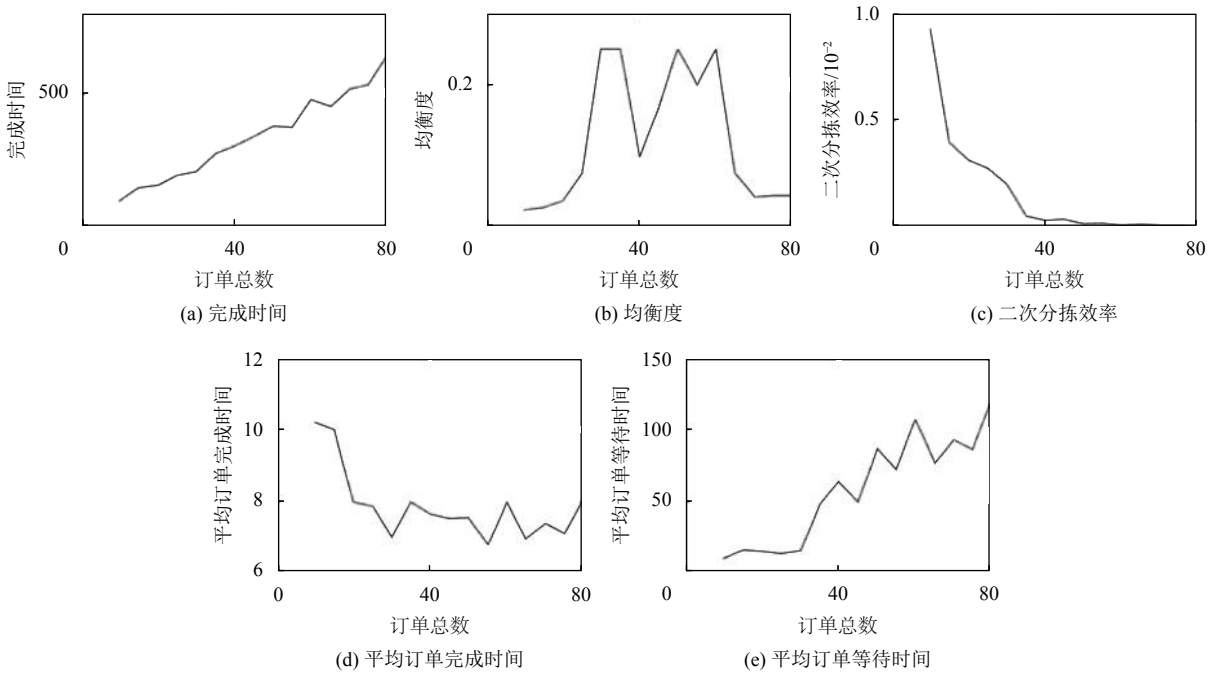


图 8 订单总数对系统效率的影响

Figure 8 The effect of total orders on system efficiency

完成时间增加, 平均订单完成时间增加, 平衡度变差, 分拣效率降低, 平均订单等待时间增加。订单批量环境从[1, 2]增加到[1, 5]时, 均衡度变化不大, 在0.23附近小幅度波动, 分拣效率降低幅度不

大, 但随着订单批量的继续增加, 均衡度和分拣效率会急剧降低。因此在此订单总数为30、拣选区域个数为3的情况下, 本文设计的系统适用于批量小于[1, 5]的订单环境中。

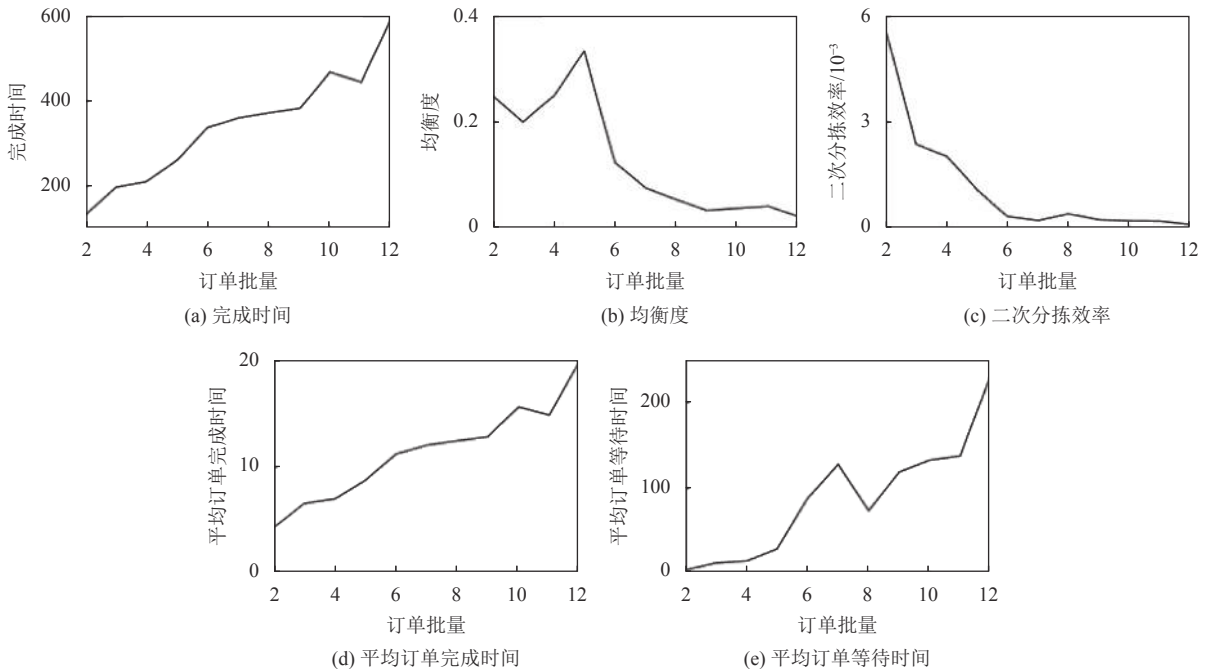


图 9 订单批量对系统效率的影响

Figure 9 Impact of order batch on system efficiency

4 结论

针对电商分区拣选问题, 本文将订单完全拆分, 并采用组合型策略作为拣选行走策略, 构建了考虑总服务时间最小、分区工作量平衡度最优和二次分拣效率最高的多目标拣选分批模型。由于目标之间存在冲突, 本文设计了NSGA-II算法对多目标优化模型进行求解, 并采用文献[2]的算例证明系统的有效性。该算法虽然会增加订单二次分拣的难度, 但能有效地减少总服务时间和提高各区工作量平衡度。同时, 本文分析了不同拣选区域个数、订单总数和订单批量对系统效率的影响, 发现随着拣选区域的个数增加, 拣选完成时间减少, 二次分拣效率提高, 但均衡度表现变差; 随着订单总数的增加, 拣选完成时间减少, 二次分拣效率降低, 过少或过多的订单数量, 都会导致均衡度表现差; 随着订单批量逐渐增加, 完成时间增加, 但均衡度变差, 二次分拣效率降低。未来的研究可将拣选与配送作业综合优化, 在拣选分批时加入客户地址和时间窗等信息, 提高算法的实用性。

参考文献:

- [1] PANSART L, CATUSSE N, CAMBAZARD H. Exact algorithms for the order picking problem[J]. *Computers & Operations Research*, 2018, 100: 117-127.
- [2] 王旭坪, 张珺, 易彩玉. 电子商务人工并行分区拣选系统服务效率优化研究[J]. *管理工程学报*, 2017, 31(2): 209-215.
WANG Xuping, ZHANG Jun, YI Caiyu. Optimizing operation efficiency of e-commerce manual synchronized zone order picking system[J]. *Journal of Industrial Engineering and Engineering Management*, 2017, 31(2): 209-215.
- [3] JANE C C, LAIH Y W. A clustering algorithm for item assignment in a synchronized zone order picking system[J]. *European Journal of Operational Research*, 2005, 166(2): 489-496.
- [4] 王转, 裴泽平. 启发式路径下节约里程的订单分批算法[J]. *计算机工程与应用*, 2018, 54(23): 203-209, 222.
WANG Zhuan, PEI Zeping. Order batching algorithm based on heuristic picking route for saving mileage[J]. *Computer Engineering and Applications*, 2018, 54(23): 203-209, 222.
- [5] VALLE C A, BEASLEY J E, DA CUNHA A S. Optimally solving the joint order batching and picker routing problem[J]. *European Journal of Operational Research*, 2017, 262(3): 817-843.
- [6] 李明, 吴耀华, 张健, 等. 基于品项相关性的阵列式自动拣选机货位优化[J]. *计算机集成制造系统*, 2015, 21(7): 1896-1905.
LI Ming, WU Yaohua, ZHANG Jian, et al. Slotting optimization of array automated picking device based on item correlations[J]. *Computer Integrated Manufacturing Systems*, 2015, 21(7): 1896-1905.
- [7] YU M, DE KOSTER R B M. The impact of order batching and picking area zoning on order picking system performance[J]. *European Journal of Operational Research*, 2009, 198(2): 480-490.
- [8] ŽULJ I, KRAMER S, SCHNEIDER M. A hybrid of adaptive large neighborhood search and tabu search for the order-batching problem[J]. *European Journal of Operational Research*, 2018, 264(2): 653-664.
- [9] 王旭坪, 张珺, 马骏. 考虑完成期限的电子商务在线订单分批模型及算法[J]. *管理科学*, 2014, 27(6): 103-113.
WANG Xuping, ZHANG Jun, MA Jun. E-commerce on-line order batching model and algorithm: considering due time[J]. *Journal of Management Science*, 2014, 27(6): 103-113.
- [10] HO Y C, LIN J W. Improving order-picking performance by converting a sequential zone-picking line into a zone-picking network[J]. *Computers & Industrial Engineering*, 2017, 113: 241-255.
- [11] DE KOSTER R B M, LE-DUC T, ZAPERPOUR N. Determining the number of zones in a pick-and-sort order picking system[J]. *International Journal of Production Research*, 2012, 50(3): 757-771.
- [12] JIANG X W, ZHOU Y X, ZHANG Y K, et al. Order batching and sequencing problem under the pick-and-sort strategy in on-line supermarkets[J]. *Procedia Computer Science*, 2018, 126: 1895-1993.
- [13] 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 random service system efficiency of sorted storage manual order picking manners[J]. *Journal of Management Sciences in China*, 2012, 15(2): 59-71.
- [14] 赖文星, 邓忠民. 基于支配强度的NSGA2改进算法[J]. *计算机科学*, 2018, 45(6): 187-192.
LAI Wenxing, DENG Zhongmin. Improved NSGA2 algorithm based on dominant strength[J]. *Computer Science*, 2018, 45(6): 187-192.

(责任编辑: 张广珍)