

DOI:10.19322/j.cnki.issn.1006-4710.2022.04.012

# 基于拣货单排序的货物动态协同与路径联合优化研究

孙军艳, 闫春妍, 陈智瑞, 牛亚儒

(陕西科技大学 机电工程学院, 陕西 西安 710021)

**摘要:** 为提高拣货作业效率,提出了一种“路径优化-货物动态协同-拣货单排序”的循环拣货策略,设计了嵌套 GASA 算法进行求解。算例表明:在 5、13、54 张不同规模的拣货单条件下,相比于传统拣货策略,本文提出的排序协同拣货策略分别节约拣货时间 12.8%、17.9%、24.7%,拣货路径节约了 16.7%、23.8%、39.7%;相较于动态拣货策略,拣货时间分别节约 5.13%、12.05%、15.69%,拣货路径节约了 6.69%、16.02%、24.20%。随着拣货单数量的增加,节约的时间和距离越多,优化效果越明显。本文所提模型和方法为大批量拣货问题的优化提供了思路。

**关键词:** 拣货策略; 路径优化; 货物动态协同; 拣货单排序; GASA 算法

**中图分类号:** F252; TP391.9

**文献标志码:** A

**文章编号:** 1006-4710(2022)04-0558-12

## Research on dynamic cargo collaboration and path joint optimization based on pick order sorting

SUN Junyan, YAN Chunyan, CHEN Zhirui, NIU Yaru

(College of Mechanical &amp; Electrical Engineering, Shaanxi University of Science &amp; Technology, Xi'an 710021, China)

**Abstract:** In order to improve the efficiency of picking operation, a circular picking strategy of “path optimization-dynamic cargo collaboration-picking order sorting” was proposed, with the nested GASA algorithm designed to solve it. The example shows that the sorting cooperative picking strategy can save 12.8%, 17.9% and 24.7% picking time and 16.7%, 23.8% and 39.7% picking path respectively compared with that by the traditional picking strategy under the condition of 5, 13, and 54 picking orders of different sizes. Compared with the dynamic picking strategy, the picking time can be saved by 5.13%, 12.05% and 15.69% respectively, and the picking path can be saved by 6.69%, 16.02% and 24.20% respectively. With the increase of the number of picking orders, the more time and distance saved, the more obvious the optimization effect. The model and method proposed in this paper provide ideas for optimizing the problem of bulk picking.

**Key words:** picking strategy; path optimization; dynamic cargo collaboration; picking order sorting; GASA algorithm

拣选作业作为配送中心的核心工序之一,其效率水平直接影响着配送中心的效率和水平。影响拣选作业效率的主要有拣选路径、储位分配、订单分批、拣选单排序等因素,国内外学者分别在这方面做了大量的研究。

在拣货路径优化方面,国内外学者大多采用智能算法如遗传算法、混合蚁群算法、禁忌搜索算法等对拣货路径和时间进行优化,以最大限度地减少拣选行程距离和时间,从而提高仓库运营效率。闫军艳等<sup>[1]</sup>对货物拣取路径顺序进行优化,以最短拣货路径

**收稿日期:** 2022-02-08; **网络出版日期:** 2022-06-20

**网络出版地址:** <https://kns.cnki.net/kcms/detail/61.1294.N.20220618.1017.002.html>

**基金项目:** 陕西省科技厅项目(2018GY-026);陕西省社会科学基金项目(2020R043);陕西科技大学博士启动基金项目(2018BJ-12);西安市科技局项目(21RKYJ0023)

**第一作者:** 孙军艳,女,博士,副教授,硕导,研究方向为物流信息、供应链管理。E-mail: 1427069529@qq.com

**通信作者:** 闫春妍,女,硕士生,研究方向为系统优化。E-mail: 2124199113@qq.com

为目标,建立了 TSP 模型优化拣货路径,将 GASA 算法和 GA 算法进行比较,结果表明 GASA 算法优于 GA 算法,拣货路径比之前节约了 0.6%。Zhou 等<sup>[2]</sup>针对 V 型仓储布局,建立了返回型随机拣选路径模型,利用大数据技术进行仿真,验证了返回型拣货路径策略优于 S 型,提高了拣货效率。张水旺等<sup>[3]</sup>针对多区型仓库,建立了拣货路径优化模型,并采用人工鱼群算法求解,验证了算法的有效性。

科学的货物存储分配方法可以缩短步行距离,减少搜索时间,提高仓库拣选效率。在货物动态存储策略方面,Hadi 等<sup>[4]</sup>开发了一种动态存储分配模型,动态确定货物的适当存储位置,从而提高了系统性能。Li 等<sup>[5]</sup>基于贪婪遗传算法,应用数据挖掘技术计算了产品间的关联度,解决了动态货物存储位置分配问题,该算法计算效率较高,且显著改善了拣选订单的行程距离。Manzini 等<sup>[6]</sup>建立了混合整数线性规划模型,基于时间将存储区域内的货物动态分配给不同的储位,该模型可优化复杂的仓储系统拣货决策。

在订单排序或者拣选单排序方面,叶楠等<sup>[7]</sup>通过分配拣货单的顺序,使各拣货单的复核台位置首尾相连,使用遗传算法动态优化拣货路径,该动态调整算法可为大规模、复杂仓库提供更高效率的拣货路径规划。胡金昌等<sup>[8]</sup>通过建立 0-1 整数规划模型来求解同时分拣多客户订单的订单排序问题,提高了系统的拣选效率。夏德龙等<sup>[9]</sup>提出了一种适用于“货到人”的订单排序模型,并用改进  $k$ -means 聚类算法求解了拣选台内相邻订单和拣选台之间订单的共用货架数量,以减少货架的搬运次数。

显然,拣选作业过程中,路径规划、存储分配、订单分批、拣货单排序之间相辅相成、相互影响,各环节的独立优化很难做到全局的优化。有学者进行了两两联合优化,比如 Ardjmand 等<sup>[10]</sup>针对基于拣选台的分拣系统,研究订单分批和路径优化问题,提出了一种基于列表的模拟退火算法(LBSA)和一种混合遗传算法(GA-LBSA)。研究发现,对于较小规模的问题,LBSA 在解决方案的质量方面是一个更好的选择,而当考虑 CPU 时间时,根据问题的大小,GA-LBSA 可能是更好的选择。Kübler 等<sup>[11]</sup>建立了动态存储位置分配和拣货路径联合优化模型,提出了一种迭代启发式算法,结果表明,这种方法可以显著节省行程距离。Cano 等<sup>[12]</sup>对分批后的拣货单进行排序并优化拣货路径,建立了联合拣选模型,同时最小化总行程和客户订单延误,提高了订单分拣流程的效率。Cai 等<sup>[13]</sup>对存储位置分配和路径进行

协同优化,考虑了货物位置分配和路径规划之间的关系,提出了一种货架周转的位置分配策略,通过模拟研究确定了该方法的有效性。

两两联合优化明显提高了拣货作业效率,但仍存在一定的局限。在拣货流程中,订单分批、拣货单排序、货物动态协同、拣货路径优化等是一个系统工程,两两联合优化依然只是局部优化,很难达到系统全局的优化。

本文在订单分批形成拣货单的前提下,对拣货系统中最重要的环节——拣货单排序、货物动态协同、拣货路径优化三者进行联合优化,以求在系统层面实现全局最优。具体研究思路为:将拣货单排序的思想融入货物动态协同与拣货路径优化中,设计了一种拣货单路径优化-货物动态协同-拣货单排序的多阶段循环拣货策略,利用拣货单之间、最优路径之间的相似性,尽最大可能利用拣货车的剩余容量,减少后续拣货单的拣选距离,从整体上对拣货作业进行优化。

## 1 动态拣货策略设计

对于拣选作业,通常情况下,在一个拣货周期内,拣选系统会根据一定的规则和拣货车的容量,将具有相似性的订单形成拣选单,拣货员根据拣货单最优路径循环完成拣货作业,以达到减少拣选员的行走路径,提高拣货效率的目的,我们将该策略称为传统拣货策略。

为进一步提高拣货效率,一种动态拣货策略被提出<sup>[14]</sup>,其主要思想是利用相邻拣货单之间共同的拣选路径,在拣选上一张拣货单  $h$  的同时,提前将下一张拣货单  $h+1$  上的某个货品顺路拣出,以最大可能地减少下一张拣货单  $h+1$  的拣选路径,以此类推,最终实现所有拣货单总拣货时间的缩减,提高拣货效率。

本文将上述动态拣货策略与拣货单排序相结合,提出了对拣选作业进行“路径优化-货物动态协同-拣货单排序”的循环拣货策略(简称“排序协同策略”),该策略的核心思想是:首先对所有拣货单进行路径优化,然后计算在拣选拣货单  $h$  的货物时能够协同拣选的拣货单  $h+1$  的货物组合,要求使拣货单  $h+1$  的最优路径减少最多,以此类推,得到两两拣货单之间能够使对方路径减少最多的路径节约矩阵;最后以路径节约矩阵为基础进行拣货单排序,以“边排序边协同”的方式,达到缩短所有拣货单拣选时间的目的。

具体流程见图 1,具体步骤如下。

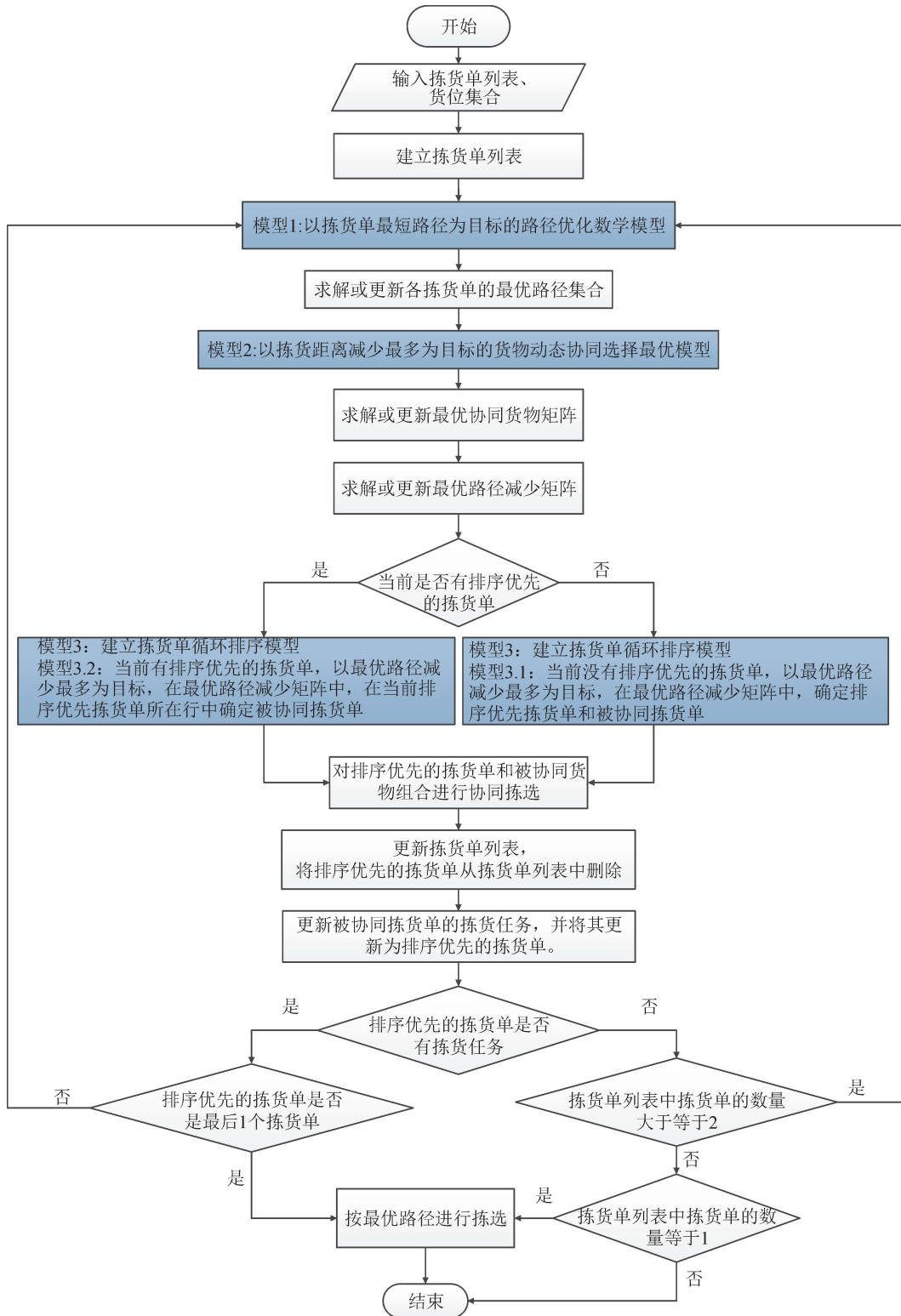


图 1 基于拣货单排序的货物动态协同与路径协同优化流程图

Fig. 1 Flow chart of dynamic cargo collaboration and path collaboration optimization based on picking orders sorting

Step1:初始条件设置

确定仓库布局,在拣货台处设置一个位置,作为被协同货物的临时存放处。明确货位信息、拣货车容量、货位之间、货位与拣货台之间的距离等信息。

Step2:拣货单最优路径模型的建立

建立以拣货时间最短为目标的数学模型,设计启发式算法,求解各拣货单的最优路径。

Step3:建立拣货单列表,求解各拣货单的最优路径集合

根据一个周期内拣货单  $h(h=1,2,\dots,M)$  的数量

$M$ ,建立拣货单列表  $H=\{\text{拣货单 } 1, \text{拣货单 } 2, \dots, \text{拣货单 } M\}$ ,基于 Step1 中的模型和算法,循环求解各拣货单  $h$  的最优路径。

Step4:建立动态协同货物组合最优模型

对于拣货单  $h$  和拣货单  $g$ ,在满足拣货车容量的前提下,根据共同路径,在拣取拣货单  $h$  的同时,协同(捎取)拣货单  $g$  的某些货物,以使拣货单  $g$  的最优路径减少最多,得到拣货单  $h$  协同拣货单  $g$  的最优货物组合,以此类推,建立最优协同货物矩阵。

Step5:建立拣货单循环排序模型

根据最优协同货物矩阵,计算当各拣货单去掉被协同货物时,最优路径的减少值,建立最优路径节约矩阵。

Step5.1:当前没有排序优先的拣货单。从最优路径节约矩阵中找出节约值最大的一对拣货单,将协同拣货单作为排序优先的拣货单,被协同拣货单作为下一个拣货单。

Step5.2:当前有排序优先的拣货单。将当前排序优先的拣货单作为协同拣货单,在最优路径节约矩阵中,定位当前排序优先拣货单所在的行,找出最大值,将其对应的拣货单作为被协同拣货单,被协同拣货单作为下一个拣货单。

Step6:对排序优先的拣货单和被协同货物组合进行协同拣选

对排序优先的拣货单根据最优路径进行拣选,同时对共同路径上的下一张拣货单的最优货物组合进行协同拣选,将其捎取到拣选台被调整处。

Step7:更新被协同拣货单的拣货任务

更新拣货单列表  $H$ 。将排序优先的拣货单从中删除,更新被协同拣货单的拣货任务,并将其更新为排序优先的拣货单。

如果排序优先的拣货单有拣选任务:更新被协同拣货单的最优路径。更新最优协同货物矩阵,删除排序优先的拣货单所在的行与列,根据更新后的拣货单列表,更新被协同拣货单所在的行。更新最优路径节约矩阵,删除排序优先的拣货单所在的行与列,根据更新后的最优协同货物矩阵,更新被协同拣货单所在的行。将被协同拣货单更新为排序优先的拣货单。进入 step8.1。

如果排序优先的拣货单没有需要拣选的货物,则将其从拣货单列表  $H$  中删除,更新拣货单列表。更新最优协同货物矩阵,删除其所在的行与列。更新最优路径节约矩阵,删除其所在的行与列。进入 step8.2。

Step8:判断拣货单列表中是否有拣货单

Step8.1:根据拣货单列表  $H$ ,判断排序优先的拣货单是否是最后一个拣货单,如果否,进入 step5.2;如果是,根据最优拣货路径完成该拣货单的拣货任务,结束。

Step8.2:根据拣货单列表  $H$ ,判断拣货单列表  $H$  中拣货单的数量。如果有 2 个及 2 个以上的拣货单,进入 step5.1;如果有 1 个拣货单,根据最优拣货路径完成该拣货单的拣货任务,结束;如果没有拣货单,结束。

## 2 数学模型的建立

### 2.1 基本假设

- 1) 采用摘果式“人到货”的拣货方式进行拣选;
- 2) 拣货员可以沿纵向和横向的通道双向行走;
- 3) 每个货位存储一种货物,即货位和品项一一对应;
- 4) 货位足够大,拣货过程中不存在货位缺货现象;
- 5) 订单中货物、各货位的数量及其存储位置已知;
- 6) 订单分批形成拣货单,订单分批时不允许进行分割,保证一个订单一次拣完,每张拣货单的大小不超过拣货车的最大容量;
- 7) 拣货员一个循环完成一个拣货单的拣选,一个工作周期内完成多个拣货单的拣选任务;
- 8) 每张拣货单在拣选过程中可以协同下一个拣货单的多个货物;
- 9) 在拣货台设置一个临时存放处,用于暂存下一个拣货单的被调整货物。

### 2.2 参数设置

基本参数与符号定义如表 1 所示。

变量定义如下:

$W_h$ : 拣货单  $h$  在储位上的货物集合,  $W_h = \{\omega_h^1, \omega_h^2, \dots, \omega_h^l\}$ ;

$W'_h$ : 拣货单  $h$  最优路径上途径的储位集合,  $W'_h = \{\omega_h^1, \omega_h^2, \dots, \omega_h^l\}$ ;

$W'_{h,g}$ : 拣货单  $h$  和  $g$  最优路径上的共有货物集合,  $W'_{h,g} = \{\omega_{h,g}^1, \omega_{h,g}^2, \dots, \omega_{h,g}^l\}$ ;

$\bar{W}^g_{h,g}$ : 拣货单  $h$  与  $g$  最优路径上,拣货单  $g$  可被协同的货物集合,  $\bar{W}^g_{h,g} = \{\omega_{h,g}^{g,1}, \omega_{h,g}^{g,2}, \dots, \omega_{h,g}^{g,l}\}$ ;

$\bar{W}^g_{h,g}$ : 拣货单  $h$  与  $g$  最优路径上,拣货单  $g$  最优协同的货物集合,  $\bar{W}^g_{h,g} = \{\bar{\omega}_{h,g}^{g,1}, \bar{\omega}_{h,g}^{g,2}, \dots, \bar{\omega}_{h,g}^{g,l}\}$ ;

$W''_{h,g}$ : 拣货单  $g$  的货物被协同后,剩余货物的集合,  $W''_{h,g} = \{\omega_{h,g}^{g,1}, \omega_{h,g}^{g,2}, \dots, \omega_{h,g}^{g,l}\}$ ;

$G_{h,k}^i$ : 排序为  $k$  的拣货单  $h$  是否包含货位  $i$ , 若包含, 为 1; 不包含, 为 0;

$G_{k+1}^i$ : 排序为  $k+1$  的原始拣货单是否包含货位  $i$ , 若包含, 为 1; 不包含, 为 0。

表 1 基本参数和符号的定义

Tab. 1 Definition of basic parameters and symbols

参数	含义	参数	含义
$M$ /个	拣货单的总数量	$n^i/m^3$	第 $i$ 个拣货位的单个货物的体积
$H$	拣货单编号列表, $H = \{1, 2, \dots, M\}$	$d_{ij}/m$	拣货位 $i, j$ 之间的距离
$h$	拣货单编号, $h \in H$	$v/(m \cdot s^{-1})$	拣货人员平均行走速度
$g$	拣货单编号, $g \in H$	$x_h^i/\text{个}$	拣货单 $h$ 在第 $i$ 个拣货位的品项的数量
$K$	拣货单排序编号列表, $K = \{1, 2, \dots, M\}$	$x_k^i/\text{个}$	排序为 $k$ 的拣货单在第 $i$ 个拣货位的品项的数量
$k$	拣货单排序编号, $k \in K$	$T/s$	拣货时间
$I$ /个	货位的总数量	$\Delta T/s$	减少的拣货时间
$Q/m^3$	拣货车的容量	$S/m$	行走距离
$Q'/m^3$	拣货台被调整处的容量	$\Delta S/m$	节约的行走距离
$t_u/s$	拣取每单位货物所需的时间	$V/m^3$	总货物体积
$t'_u/s$	协同每单位货物所需的时间	$N/\text{个}$	调整的货物数

2.3 模型的构建

由图 1 可以看出, 本文提出的排序协同策略涉及三个数学模型: 拣货单最优路径数学模型、货物动态协同选择的数学模型、拣货单循环排序模型。

三个模型中, 以模型 3 为中心, 在运行模型 3 时, 首先需要调用模型 1, 计算各拣货单最优路径, 建立拣货单列表; 接着调用模型 2, 计算最优的货物动态协同矩阵, 得到最优的路径节约矩阵; 以此为基础, 运行模型 3 得到排序优先的拣货单和被协同拣货单, 拣选完成后更新拣货单列表。以此类推, 模型 3 重复调用模型 1 和模型 2, 直到所有的拣货单被排序。

2.3.1 模型 1: 拣货单最优路径数学模型

对于拣货单  $h$ , 以拣货单最短路径为目标建立数学模型。

目标函数为:

$$S_h = \min \sum_{i=1}^I \sum_{j=1}^I d_{ij} \cdot e_h^{i,j} \quad \forall h \in H \quad (1)$$

约束条件:

$$1) \quad w_h^i = \begin{cases} 1 & \text{拣货单 } h \text{ 中包含} \\ & \text{货位 } i \text{ 上的货物} \\ 0 & \text{否则} \end{cases} \quad (2)$$

2) 拣货单  $h$  中每个货物的货位都要被访问 1 次:

$$\sum_{i=1}^I e_h^{i,j} = 1 \quad \forall i \neq j, \forall w_h^i = 1, \forall w_h^j = 1 \quad (3)$$

$$\sum_{j=1}^I e_h^{i,j} = 1 \quad \forall i \neq j, \forall w_h^i = 1, \forall w_h^j = 1 \quad (4)$$

3) 拣货单  $h$  的体积小于拣货车容量  $Q$ :

$$\sum_{i=1}^I x_h^i \cdot w_h^i \cdot n^i \leq Q \quad (5)$$

4) 拣货单  $h$  所包含的货位的数量不能大于货位的总数量:

$$\sum_{i=1}^I w_h^i \leq I \quad (6)$$

决策变量:

$$e_h^{i,j} = \begin{cases} 1 & \text{拣货单 } h \text{ 中, 货位 } i, j \text{ 之间存在路径} \\ 0 & \text{拣货单 } h \text{ 中, 货位 } i, j \text{ 之间不存在路径,} \\ & \text{或 } w_h^i = 0 \text{ 或 } w_h^j = 0 \end{cases} \quad (7)$$

2.3.2 模型 2: 货物动态协同选择最优模型

拣货单  $h$  协同拣货单  $g$  时, 以剩余拣货单  $g$  最优路径减少最多为目标, 建立拣货单  $h$  的最优协同货物选择模型。

目标函数为:

$$\Delta S_{h,g}^i = \max \{ (S_g - S'_{h,g}^i) \} \quad \forall g \in H, \forall g \neq h \quad (8)$$

$$S_g = \min \sum_{i=1}^I \sum_{j=1}^I d_{ij} \cdot e_g^{i,j} \quad (9)$$

$$S'_{h,g}^i = \min \sum_{i=1}^I \sum_{j=1}^I d_{ij} \cdot e'_{h,g}^{i,j} \quad (10)$$

约束条件为:

$$1) \quad w_g^i = \begin{cases} 1 & \text{拣货单 } g \text{ 中包含} \\ & \text{货位 } i \text{ 上的货物} \\ 0 & \text{否则} \end{cases} \quad (11)$$

2) 拣货单  $g$  中每个货物对应的货位都要被访问 1 次:

$$\sum_{i=1}^I e_g^{i,j} = 1 \quad \forall i \neq j, \forall w_g^i = 1, \forall w_g^j = 1 \quad (12)$$

$$\sum_{j=1}^I e_g^{i,j} = 1 \quad \forall i \neq j, \forall w_g^i = 1, \forall w_g^j = 1 \quad (13)$$

3) 拣货单  $g$  的体积小于拣货车容量  $Q$ :

$$\sum_{i=1}^I x_g^i \cdot w_g^i \cdot n^i \leq Q \quad (14)$$

4) 拣货单  $g$  所包含的货位的数量不能大于货位的总数量:

$$\sum_{i=1}^I w_g^i \leq I \quad (15)$$

$$w_{h,g}^{g,i} = w_h^i \cdot w_g^i \cdot w_h^i \cdot w_g^i \quad (16)$$

$$w_{h,g}^{g,i} = \begin{cases} 1 & \text{拣货单 } g \text{ 货位 } i \text{ 上的货物} \\ & \text{可被拣货单 } h \text{ 协同} \\ 0 & \text{否则} \end{cases}$$

$$w_h^i = \begin{cases} 1 & \text{货位 } i \text{ 在拣货单 } h \text{ 的最优路径上} \\ 0 & \text{否则} \end{cases}$$

6) 拣货单  $g$  中,被协同的货物与剩余的货物之和等于拣货单  $g$  上所有的货物:

$$w_g = w_{h,g}^{g,i} \cdot Y_{h,g}^{g,i} + w_{h,g}^{h,i} \quad (17)$$

7) 去除协同货物后,拣货单  $g$  剩余货物的货位都要被访问 1 次:

$$\sum_{i=1}^I e_g^{i,j} = 1 \quad \forall i \neq j, \forall w_{h,g}^{g,j} = 1, \forall w_{h,g}^{h,i} = 1 \quad (18)$$

$$\sum_{j=1}^I e_g^{i,j} = 1 \quad \forall i \neq j, \forall w_{h,g}^{g,i} = 1, \forall w_{h,g}^{h,j} = 1 \quad (19)$$

8) 被协同货物不能超过拣货车剩余容积:

$$\sum_{i=1}^I x_g^i \cdot w_{h,g}^{g,i} \cdot n^i \cdot Y_{h,g}^{g,i} \leq Q - \sum_{i=1}^I x_h^i \cdot n^i \cdot w_h^i \quad (20)$$

9) 协同货物数量不能超过共同路径上的货物

$$D = \begin{bmatrix} \{0,0,\dots,0\} & \{w_{1,2}^{2,1}, w_{1,2}^{2,2}, \dots, w_{1,2}^{2,I}\} & \dots & \{w_{1,M}^{M,1}, w_{1,M}^{M,2}, \dots, w_{1,M}^{M,I}\} \\ \{w_{2,1}^{1,1}, w_{2,1}^{1,2}, \dots, w_{2,1}^{1,I}\} & \{0,0,\dots,0\} & \dots & \{w_{2,M}^{M,1}, w_{2,M}^{M,2}, \dots, w_{2,M}^{M,I}\} \\ \vdots & \vdots & \ddots & \vdots \\ \{w_{M,1}^{1,1}, w_{M,1}^{1,2}, \dots, w_{M,1}^{1,I}\} & \{w_{M,2}^{2,1}, w_{M,2}^{2,2}, \dots, w_{M,2}^{2,I}\} & \dots & \{0,0,\dots,0\} \end{bmatrix} \quad (23)$$

$$\bar{D} = \begin{bmatrix} \{0,0,\dots,0\} & \{\bar{w}_{1,2}^{2,1}, \bar{w}_{1,2}^{2,2}, \dots, \bar{w}_{1,2}^{2,I}\} & \dots & \{\bar{w}_{1,M}^{M,1}, \bar{w}_{1,M}^{M,2}, \dots, \bar{w}_{1,M}^{M,I}\} \\ \{\bar{w}_{2,1}^{1,1}, \bar{w}_{2,1}^{1,2}, \dots, \bar{w}_{2,1}^{1,I}\} & \{0,0,\dots,0\} & \dots & \{\bar{w}_{2,M}^{M,1}, \bar{w}_{2,M}^{M,2}, \dots, \bar{w}_{2,M}^{M,I}\} \\ \vdots & \vdots & \ddots & \vdots \\ \{\bar{w}_{M,1}^{1,1}, \bar{w}_{M,1}^{1,2}, \dots, \bar{w}_{M,1}^{1,I}\} & \{\bar{w}_{M,2}^{2,1}, \bar{w}_{M,2}^{2,2}, \dots, \bar{w}_{M,2}^{2,I}\} & \dots & \{0,0,\dots,0\} \end{bmatrix} \quad (24)$$

其中,  $\bar{w}_{h,g}^{g,i} = \begin{cases} 1 & \text{拣货单 } h \text{ 和 } g \text{ 之间,拣货单 } g \text{ 的最优被协同货位 } i \text{ 上的货物} \\ 0 & \text{否则} \end{cases}$

1) 模型 3.1:当前没有排序优先的拣货单

以最优路径减少最多为目标,在最优路径节约矩阵中,寻找排序优先拣货单和被协同拣货单,目标函数为:

数量:

$$\sum_{i=1}^I Y_{h,g}^{g,i} \leq \sum_{i=1}^I w_{h,g}^{g,i} \quad (21)$$

决策变量:

$$e_g^{i,j} = \begin{cases} 1 & \text{拣货单 } g \text{ 中,货位 } i,j \text{ 之间存在路径} \\ 0 & \text{拣货单 } g \text{ 中,货位 } i,j \text{ 之间不存在路径,} \\ & \text{或 } w_g^i = 0 \text{ 或 } w_g^j = 0 \end{cases}$$

$$e_g^{i,j} = \begin{cases} 1 & \text{去掉协同货物后,拣货单 } g \text{ 中,} \\ & \text{货位 } i,j \text{ 之间存在路径} \\ 0 & \text{拣货单 } g \text{ 的货位 } i,j \text{ 之间不存在路径,} \\ & \text{或 } w_{h,g}^{g,i} = 0, \text{ 或 } w_{h,g}^{g,j} = 0 \end{cases}$$

$$Y_{h,g}^{g,i} = \begin{cases} 1 & \text{拣货单 } h \text{ 和 } g \text{ 共同路径上,货位 } i \text{ 上} \\ & \text{的货物是拣货单 } g \text{ 的被协同货物} \\ 0 & \text{否则} \end{cases}$$

### 2.3.3 模型 3:拣货单循环排序模型

首先,基于两两拣货单中最优路径上的共同货位,构建协同货物矩阵  $D$ ,基于模型 2,得到最优路径节约矩阵  $\Delta S$ 。 $\Delta S$  中,第  $h$  行表示拣货单  $h$  协同其他拣货单时,其他拣货单的最优距离节约值;第  $g$  列表示其他拣货单协同拣货单  $g$  时,拣货单  $g$  的最优距离节约值。例如  $\Delta S_{h,g}^{g,i}$  代表拣货单  $h$  协同拣货单  $g$  时,拣货单  $g$  的最优距离节约值。

$$\Delta S = \begin{bmatrix} 0 & \Delta S_{1,2}^2 & \dots & \Delta S_{1,M}^{M,I} \\ \Delta S_{2,1}^1 & 0 & \dots & \Delta S_{2,M}^{M,I} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta S_{M,1}^1 & \Delta S_{M,2}^2 & \dots & 0 \end{bmatrix} \quad (22)$$

其次,基于最优路径节约矩阵  $\Delta S$ ,得到最优协同货物矩阵:

$$\max \Delta S_{h,g}^{g,i} \quad \forall g \neq h; \forall g \in H, \forall h \in H \quad (25)$$

其对应的  $h$  即为排序优先拣货单,  $g$  为被协同拣货单。

2) 模型 3.2:当前有排序优先的拣货单  $h$

以最优路径减少最多为目标,在最优路径节约矩阵中,在第  $h$  行中寻找被协同拣货单  $g$ , 目标函数为:

$$\max \Delta S_{h,g}^s \quad \forall g \neq h; \forall g \in H \quad (26)$$

2.3.4 所有拣货单的总作业时间

在拣货单排序的协同优化策略下,对于排序为  $k$  的拣货单,除了需要根据前一个拣货单  $k-1$  已经捎取的货物,更新排序为  $k$  的拣货单的拣货任务;另一方面,需要采用循环判断找出可使下一个拣货单  $k+1$  拣货距离减少最多的货物组合,对其进行动态调整。若排序为  $k$  的拣货单为拣货单  $h$ ,则排序为  $k$  的拣货单的拣选时间包括三部分:更新后的行走时间、更新后的作业时间、协同货物时间。优化后,1 个周期内所有拣货单总的拣货时间  $T_C$  为:

$$T'_{h,k} = \frac{\sum_{i=1}^I \sum_{j=1}^I d_{ij} \cdot e_{h,k}^{i,j}}{v} + \sum_{i=1}^I x_h^i \cdot G'_{h,k} \cdot t_u + \sum_{i=1}^I x_{k+1}^i \cdot G_{k+1}^i \cdot t'_u \quad (27)$$

$$T_C = \sum_{k=1}^M \sum_{h=1}^M P_{h,k} \cdot T'_{h,k} \quad (28)$$

其中:

$$P_{h,k} = \begin{cases} 1 & \text{拣货单 } h \text{ 排序为 } k \quad k = 1, 2, \dots, M \\ 0 & \text{否则} \quad h = 1, 2, \dots, M \end{cases}$$

$$\sum_{h=1}^M P_{h,k} = 1, k = 1, 2, \dots, M$$

每个排序位置只能有一张拣货单。

3 算法设计与算例分析

3.1 算法设计

根据协同拣货策略和数学模型可知,该问题属于三阶段循环 NP-hard 决策问题。由于各阶段决策密不可分,环环相扣,循环更新,具有连贯性、协同性、多约束等特点,除了算法的可靠性以外,算法的效率至关重要。本文算法的设计思想是在确保解合理性的前提下,尽量提高各阶段算法的运算效率。

1) 模型 1,采用混合遗传模拟退火(GASA)算

法对拣货单路径进行优化。GASA 算法将模拟退火算法(SA)较强的局部搜索能力与遗传算法融合,解决了群体的多样性和收敛速度的矛盾,可避免算法陷入局部最优,同时提升了算法的收敛速度。具体步骤如下。

Step1:编码设计。对拣货单的拣货位编号采用自然数编码。例如,对于有 6 个拣货位的拣货单,该拣货单的编码串(即染色体  $C_1$ )为[2,15,22,13,53,7],其中基因值为需拣取的拣货位编号,基因位对应拣货位的排序,此编码串表示该拣货单的拣货顺序为 2-15-22-13-53-7。随机生成初始种群。

Step2:适应度函数。适应度函数  $f_1$  为拣货单最优路径  $S_h$  的倒数。计算种群中所有个体的适应度值:

$$f_1 = \frac{1}{S_h}$$

Step3:遗传操作。由于本文在设计交叉和变异算子时,新解只能在可行区域内产生,所以遗传交叉过程中只会产生可行解。

a) 选择算子,采用线性排序选择。通过排序给每个个体安排相应的选中概率。种群中的个体首先根据适应度值进行排序,然后给所有个体赋予一个序号,最好的个体序号为  $N$ ,被选中的概率为  $P_{\max}$ ,最差的个体序号为 1,被选中的概率为  $P_{\min}$ ,设  $P_i$  为第  $i$  个个体被选中的概率,则

$$P_i = P_{\min} + (P_{\max} - P_{\min}) \frac{i-1}{N-1}$$

b) 顺序交叉算子,交叉概率为  $P_c$ 。如图 2 所示,在两个父代中随机选择起始和结束位置,将父代 1 中选中的基因复制到子代 1 相同的位置,在父代 2 中将其缺少的基因按顺序填入,另一个子代以相同的方式得到。

c) 变异算子,变异概率为  $P_m$ 。随机选择一个插入位和一个基因位,把染色体中的基因位移到插入位,将其之后的基因顺移。例如,插入位是第 3 位,基因位是第 6 位,则变异后的染色体如图 3 所示。

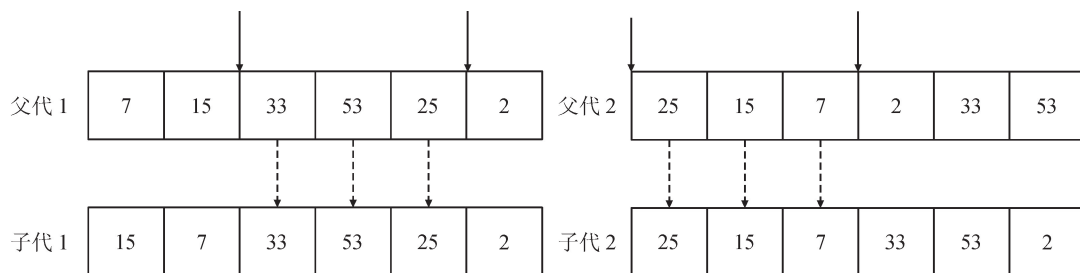


图 2 交叉操作示意图

Fig. 2 Cross operation diagram

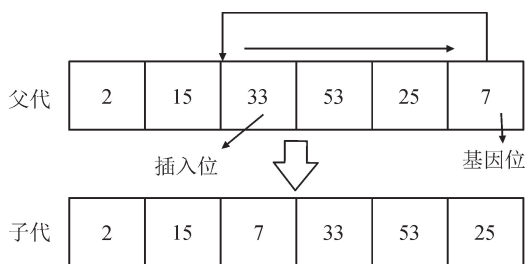


图3 变异操作示意图

Fig. 3 Mutation operation diagram

Step4:模拟退火操作。

a) 模拟退火初始化。初始化模拟退火参数,包括初始温度  $T_0$ 、降温速率  $\theta$  和温度下迭代次数  $L$ ,并以遗传操作产生的子代作为初始解。对当前解  $C_j$  作随机扰动,过程如图4所示,随机选择两个拣货位然后交换它们的位置,产生一个新解  $C_j'$ 。计算新解的适应度值  $f(C_j')$ 。

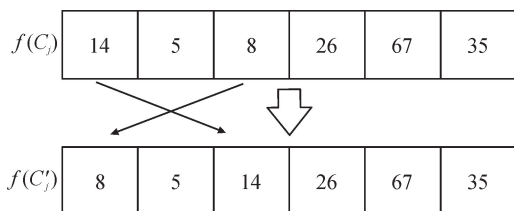


图4 产生新解的过程示意图

Fig. 4 Diagram of the process for generating a new solution

b) 降温函数。在未满足终止退火条件,同时已达到温度下迭代次数时,进行降温操作,迭代  $n+1$  次时的温度为  $T_{n+1} = \theta \times T_n$ 。

c) Metropolis 准则。遵循 Metropolis 准则给出的以一定概率接受新状态,计算适应度函数值增量  $\Delta f$ ,令  $\Delta f = f(C_j) - f(C_j')$ ,若  $\Delta f \leq 0$ ,则接受该新解为当前解;否则,若  $\exp(-\Delta f/T)$  大于  $[0, 1]$  之间的随机数,则接收当前解为新解。

Step5:判断是否达到最大迭代次数,如果是,则输出最优路径结果,否则返回 Step3。

2) 模型2,动态协同货物选择的最优。由于最优协同货物模型是一个多约束的嵌套组合问题,它是协同货物组合的可行解为基础,循环调用模型1,根据拣货单最优路径的节约值的变化,不断的在协同货物组合的可行解中寻优。本文提出一种嵌套 GASA 算法,外层采用遗传算法对拣货单协同货物组合进行优化,内层采用 GASA 算法对拣货单路径进行优化。具体步骤如下。

Step1:调用 GASA 算法计算各拣货单的最优路径。

Step2:计算两两拣货单之间共同路径上的货位

的货物,得到协同货物矩阵  $D$ 。

Step3:计算  $D$  中的各元素  $D_{h,g}$  ( $h \neq g, h \in H, g \in H$ ) 的最优协同货物组合。

Step4:编码设计。采用二进制编码,对协同货物矩阵  $D$  中的元素  $D_{h,g}$ ,即拣货单  $g$  的可能被协同货物进行编码设计。染色体  $C_2 = [i_1, i_2, i_3, \dots, i_m, \dots, i_n]$ ,基因位对应拣货位编号,基因值  $i_m$  为 0 时,代表该货物被拣货单  $h$  协同,基因值  $i_m$  为 1 时,代表该货物留在拣货单  $g$  中拣选,随机生成初始种群。

Step5:首先判断染色体中被拣货单  $h$  协同的货物的体积之和是否满足拣货车剩余容量,若不满足,则适应度值为 0,跳转至 Step6,若满足,则调用 GASA 算法,计算适应度函数  $f_2$ 。适应度函数为拣货单最优路径节约值的倒数,计算公式为:  $f_2 = 1/\Delta S_{h,g}^{\%}$ 。

Step6:产生新个体。选择操作采用轮盘赌法,交叉操作采用多点交叉算子,变异操作采用单点突变法,产生新种群。

Step7:判断是否达到最大迭代次数,若是,则生成拣货单  $g$  的最优路径节约值和最优协同货物组合;若否,则判断新种群的新个体是否满足拣货车剩余容量,若满足,则返回 Step5,若不满足,则返回 Step6 产生新个体。

Step8:循环执行 Step3,得到最优协同货物矩阵  $\bar{D}$ 。

3) 模型3,拣货单排序优化。基于模型2的最优协同货物组合,调用模型1,通过计算原始拣货单与被协同后的拣货单的最优路径,可得最优路径节约矩阵  $\Delta S$ 。不论当前有没有排序优先的拣货单,均可采用遍历的方法在  $\Delta S$  中快速准确地找到  $\max \Delta S_{h,g}^{\%}$ ,以确定排序优先的拣货单及协同拣货单。

### 3.2 实例分析

本文以双区型仓库为实例,如图5所示。设货架的长和宽均为 1m,每个巷道的宽度为 1m。纵向有 20 列、横向有 40 排货架,共 800 个货位。

#### 3.2.1 初始参数设置

模型的参数设置:拣货人员的平均行走速度  $v=1$  m/s,正常拣货时间  $t_u=2$  s,协同拣货时间  $t'_u=2.5$  s,拣货车容量  $Q=80$  m<sup>3</sup>。种群规模  $N=80$ ,  $T_0=50\ 000$ ,交叉概率  $P_c=0.4$ ,变异概率  $P_m=0.08$ 。实验数据来源于某电商的订单数据,每张拣货单包含了若干张订单,每个订单中包含了不同货物的品类与数量。



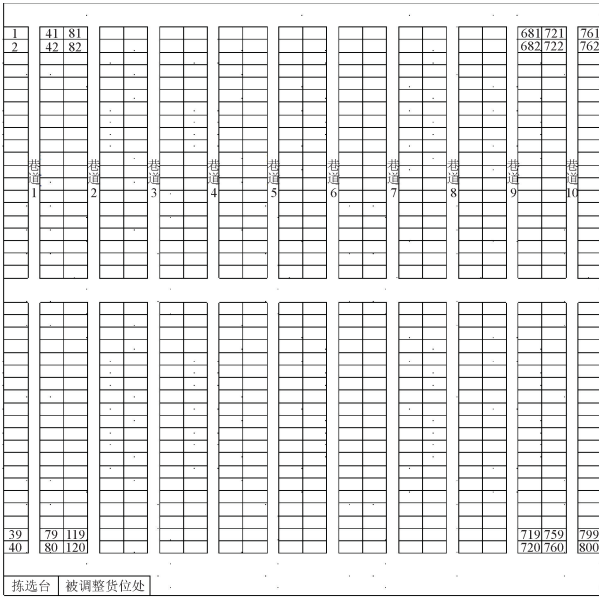


图 5 仓库布局图  
Fig. 5 Warehouse layout

### 3.2.2 策略的合理性验证

首先以 5 张拣货单为例,利用上述嵌套 GASA 算法对其进行求解,拣货单拣选顺序为 3-2-1-4-5,总拣货时间为 1 627 s,总拣货路径为 1 291 m。每张拣货单具体路径如图 6 所示。图 6(a)为排序第 1 的拣货单 3,黑点为拣货单 3 中的拣货位的货物,红点为被协同的拣货单 2 的拣货位的货物,可以看出,在进行拣货单 3 的拣选时,拣货单 2 的 4 个货物被协同拣选。图 6(b)的黑点为拣货单 2 剩余的拣货位的货物,6 个红点为被协同的拣货单 4 的拣货位的货物,绿点对应图 6(a)的红点,为被拣货单 3 已经提前拣选的 4 个货物。以此类推,可知后续拣货单的货物协同情况。对比表 2 所示的传统拣货策略,本文提出的排序协同拣货策略的拣货时间节约了 185 s,拣货距离缩短了 191 m;和动态拣货策略相比,协同货物增加了 6 个。验证了该策略的合理性。

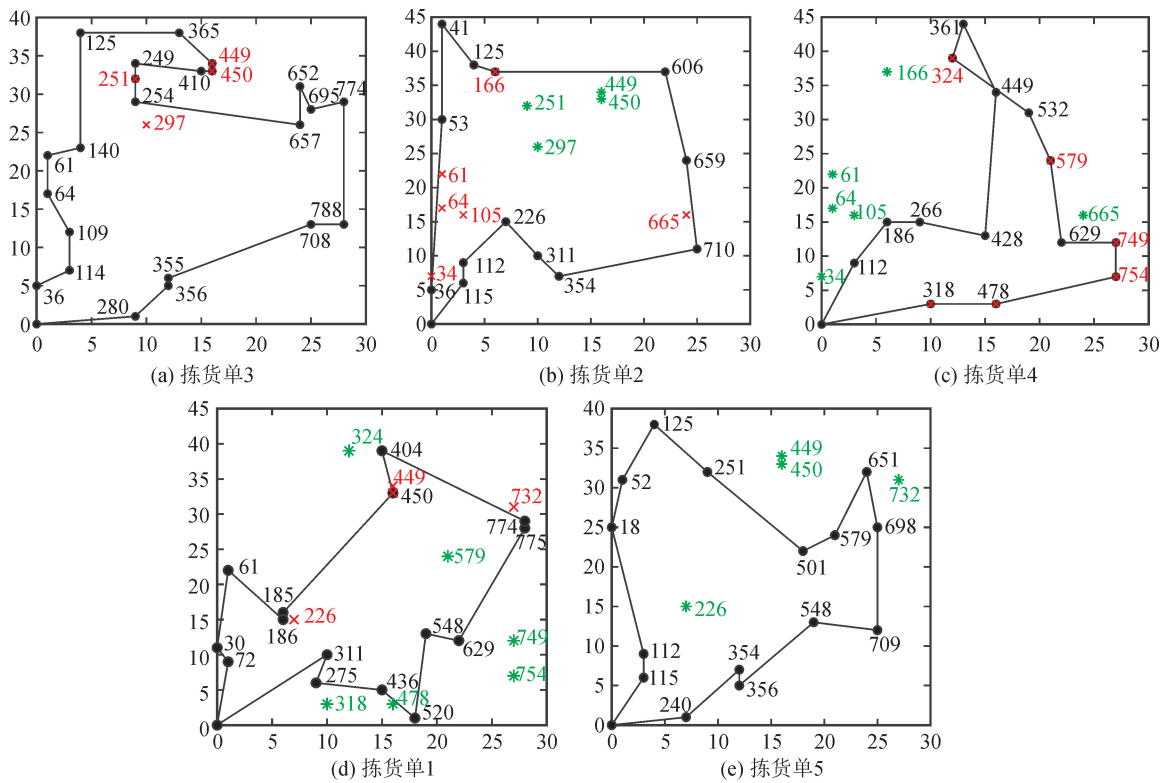


图 6 货物动态协同优化路径图  
Fig. 6 Dynamic cargo collaborative optimization path map

### 3.2.3 算法有效性分析

为验证本文提出的嵌套 GASA 算法的有效性,将其与嵌套 GA 算法进行对比,如表 3 所示。

由表 3 可知,对于不同拣货策略,在对规模大小

不同的拣货单进行求解时,嵌套 GASA 算法的求解结果均优于 GA 算法。传统策略下,GASA 比 GA 算法的拣货时间平均少 2.4%,动态策略下少 5.1%,排序协同策略下少 5.4%。可以看出,不论哪种策略,

GASA 均优于 GA 算法,并且在排序协同策略下效果更好。

3.2.4 不同规模拣货单的对比分析

分别以 5 张、13 张、54 张拣货单为例进行仿真实验,如表 4 所示。

由表 4 可以看出,与传统拣货策略相比,排序协同策略的拣货时间分别节约了 12.8%、17.9% 和

24.7%,拣货距离节约了 16.7%、23.8% 和 39.7%。

与动态策略相比,排序协同策略的拣货时间分别节约了 5.13%、12.05% 和 15.69%,拣货距离节约了 6.69%、16.02% 和 24.20%。说明采用排序协同策略可使拣货效率明显提升,且随着拣货单数量的增加,拣货效率提升越发明显。

表 2 三种拣货策略对比分析  
Tab.2 Comparative analysis of three kinds of picking strategies

h	传统拣货策略		动态拣货策略						排序协同拣货策略				
	T/s	S/m	T/s	S/m	ΔT/s	ΔS/m	N/个	k	T/s	S/m	ΔT/s	ΔS/m	N/个
1	312	242	316	242	-4	0	2	4	273	195	39	47	4
2	316	260	312	246	4	14	4	2	289	227	27	33	6
3	314	248	280	212	34	36	6	1	322	242	-8	6	4
4	346	272	282	220	64	52	2	3	304	228	42	44	6
5	339	269	326	260	13	9	—	5	254	208	85	61	
合计	1 627	1 291	1 516	1 180	111	111	14	—	1 442	1 100	185	191	20

表 3 不同算法下拣货时间的对比结果  
Tab.3 Comparison results of picking time by different pick methods

拣货策略	拣货规模/张	T/s		ΔT/T/%	ΔT/T 平均/%
		GASA 算法	GA 算法		
排序协同策略	5	1 442	1 534	6.4	
	13	3 827	4 006	4.7	5.4
	54	13 222	13 894	5.1	
动态策略	5	1 516	1 588	4.7	
	13	4 002	4 220	5.4	5.0
	54	14 296	15 011	5.0	
传统策略	5	1 627	1 680	3.3	
	13	4 511	4 622	2.5	2.4
	54	16 482	16 736	1.5	

表 4 不同规模拣货单的对比  
Tab.4 Comparison of picking lists of different sizes

拣货规模/张	传统拣货策略		动态拣货策略		排序协同策略		与传统策略对比		与动态策略对比	
	T/s	S/m	T/s	S/m	T/s	S/m	ΔT/T/%	ΔS/S/%	ΔT/T/%	ΔS/S/%
5	1 627	1 291	1 516	1 180	1 442	1 106	12.8	16.7	5.13	6.69
13	4 511	3 561	4 288	3 338	3 827	2 877	17.9	23.8	12.05	16.02
54	16 482	13 668	15 297	12 152	13 222	9 784	24.7	39.7	15.69	24.20

## 4 结 语

本文基于“边拣货边协同货物”的协同优化思想,设计了路径优化-货物动态协同-拣货单排序”的循环拣货策略,以总拣选时间最短为总目标,设计了一种多阶段循环求解算法,首先基于 GASA 算法对拣货单路径进行优化,然后使用嵌套 GASA 算法求解最优协同货物组合,最后采用遍历法对拣货单进行动态排序。

算例表明:与传统拣货策略或者动态拣货策略相比,本文提出的排序协同策略在拣货时间和拣货路径方面均具有一定的优越性,且随着拣货单数量的增加,优越性越发明显。本文的拣货策略、模型和数值仿真结果可为电商企业拣货提供决策参考。

### 参考文献:

- [1] 闫军,王璐璐,常乐. 仓配一体化仓库拣货路径优化策略应用研究[J]. 制造业自动化,2021,43(10): 1-4,14.  
YAN Jun, WANG Lulu, CHANG Le. Application research on picking path optimization strategy of warehouse distribution integration[J]. Manufacturing Automation, 2021, 43 (10): 1-4,14.
- [2] ZHOU Li, FAN Xiani, WANG Jinlong, et al. A random classified-storage picking path model for V-type storage layout[J]. Complexity, 2020: 1-12.
- [3] 张水旺,谢浩,付林萍,等. 考虑车载容量的多区型仓库拣货路径优化研究[J]. 系统科学与数学,2021,41(1): 238-253.  
ZHANG Shuiwang, XIE Hao, FU Linping, et al. Research on picking path optimization of multi-district warehouse considering vehicle capacity [J]. Journal of Systems Science and Mathematics, 2021, 41 (1): 238-253.
- [4] HADI M Z, DJATNA T. A modeling of dynamic storage assignment for order picking in beverage warehousing with drive-in rack system[C]//IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2018, 337(1): 012020.
- [5] LI Jiayi, MOGHADDAM M, NOF S Y. Dynamic storage assignment with product affinity and ABC classification-a case study[J]. The International Journal of Advanced Manufacturing Technology, 2016, 84(9): 2179-2194.
- [6] MANZINI R, ACCORSI R, GAMBERI M, et al. Modeling class-based storage assignment over life cycle picking patterns[J]. International Journal of Production Economics, 2015, 170: 790-800.
- [7] 叶楠,毕忠勤,魏恒达,等. 多区型仓库多复核台场景的拣货路径优化研究[J]. 应用科学学报,2021,39(4): 581-593.  
YE Nan, BI Zhongqin, WEI Hengda, et al. Research on picking path optimization of multi zone warehouse with multiple review stations[J]. Journal of Applied Science, 2021, 39 (4): 581-593.
- [8] 胡金昌,马文凯,杨栋,等. 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[J]. 机械工程学报,2020,56(24):246-253.  
HU Jinchang, MA Wenkai, YANG Dong, et al. Order sequencing and customer batch optimization of “goods to people” sorting system considering customer receiving order[J]. Journal of Mechanical Engineering, 2020, 56 (24): 246-253.
- [9] 夏德龙,吴耀华,王艳艳,等. 基于智能机器人的“货到人”系统订单排序优化[J]. 深圳大学学报理工版,2019,36(6): 696-701.  
XIA Delong, WU Yaohua, WANG Yanyan, et al. Order sequencing optimization of “goods to people” system based on intelligent robot[J]. Journal of Shenzhen University Science and Engineering,2019,36(6):696-701.
- [10] 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.
- [11] KÜBLER P, GLOCK C H, BAUERNHANSL T. A new iterative method for solving the joint dynamic storage location assignment, order batching and picker routing problem in manual picker-to-parts warehouses [J]. Computers & Industrial Engineering, 2020, 147: 106645.
- [12] CANO J A, CORREA-ESPINAL A A, GÓMEZ-MONTOYA R A. Mathematical programming modeling for joint order batching, sequencing and picker routing problems in manual order picking systems[J]. Journal of King Saud University-Engineering Sciences, 2020, 32(3): 219-228.
- [13] CAI Jianming, LI Xiaokang, LIANG Yue, et al. Collaborative optimization of storage location assignment and path planning in robotic mobile fulfillment systems [J]. Sustainability, 2021, 13(10): 5644.
- [14] 孙军艳,牛亚儒,苏宝,等. 双区型仓库动态拣货策略的设计及路径优化研究[J]. 包装工程,2018,39(23):

1-8.  
SUN Junyan, NIU Yaru, SU Bao, et al. Research on dynamic picking strategy design and path optimization

of double zone warehouse[J]. Packaging Engineering, 2018, 39(23): 1-8.

(责任编辑 周 蓓)

(上接第 506 页)

- [7] MOUNIRA B, YASMINA B. Design optimizatic ocriteria for windows providing low energy demand in office buildings in Algeria[J]. Environmental and Sustainability Indicators, 2020, 6(1): 33-35.
- [8] 李莉萍. 云南传统民居综合节能的考察及研究[J]. 建筑科学, 1998(6): 3-5.  
LI Liping. Preliminary investigation and research on energy saving for traditional dwellings in Yunnan Province [J]. Building Science, 1998(6): 3-5.
- [9] WANG L, GWILLIAM J, JONES P. Case study of zero energy house design in UK[J]. Energy & Buildings, 2009, 41(11): 1215-1222.
- [10] GELESZ A, CATTO L E, GOIA F, et al. Characteristics that matter in acclimate facade: a sensitivity analysis with building energy simulation tools[J]. Energy & Buildings, 2020, 229(1): 5-8.
- [11] HAMDY M, HASAN A, KAI S. Applying a multi-objective optimization approach for design of low-emission cost-effective dwellings [J]. Building & Environment, 2011, 46(1): 109-123.
- [12] 康熙, 赵士永, 白佳慧, 等. 既有居住建筑被动式超低能耗绿色化改造实践研究[J]. 建筑节能, 2020, 48(8): 18-23.  
KANG Xi, ZHAO Shiyong, BAI Jiahui, et al. Practical research on passive ultra-low energy green renovation of existing residential buildings [J]. Journal of Building Energy Efficiency, 2020, 48(8): 18-23.
- [13] 杨剑雄. 歙县历史文化名城保护利用研究[D]. 杭州: 浙江大学, 2016.  
YANG Jianxiong. Research on the protection and utilization of Shexian historical and cultural city[D]. Hangzhou: Zhejiang University, 2016.
- [14] 左传雨. 皖南传统民居的改造与再利用设计研究[D]. 苏州: 苏州大学, 2018.  
ZUO Chuanyu. Study on the protection and reuse design of traditional dwellings in southern Anhui [D]. Suzhou: Soochow University, 2018.
- [15] 司鹏飞, 南艳丽, 李博文, 等. 南方地区传统民居外墙构造及热工性能计算分析[J]. 科学技术与工程, 2014, 14(22): 264-269.  
SI Pengfei, NAN Yanli, LI Bowen, et al. The analysis of exterior wall structure and thermal performance of traditional dwelling in southern region [J]. Science Technology and Engineering, 2014, 14(22): 264-269.
- [16] 中华人民共和国住房和城乡建设部. 夏热冬冷地区居住建筑节能设计标准: JGJ 134—2010[S]. 北京: 中国建筑工业出版社, 2010.
- [17] 中华人民共和国建设部. 民用建筑设计通则: GB 50352—2005[S]. 北京: 中国建筑工业出版社, 2005.

(责任编辑 周 蓓)