

# 基于混合遗传算法的多箱型三维装箱问题研究

周丽<sup>1</sup>, 杨江龙<sup>1</sup>, 赵俊辉<sup>1</sup>, 柳虎威<sup>1,2</sup>, 王繁<sup>1</sup>

(1.北京物资学院 信息学院, 北京 101149; 2.首都经济贸易大学 管理工程学院, 北京 100070)

**摘要:** **目的** 提高电商仓储领域打包环节包装箱的利用率。**方法** 针对电商仓储领域多箱型多种物品类型的三维装箱问题, 建立混合整数规划的数学模型, 设计基于启发式经验规则和多种算子组合的装箱过程模块算法。分别从装箱顺序和带有改进型算子这两方面设计多箱型三维装箱问题混合遗传算法, 对装箱方案进行优化。**结果** 经实验证明, 在装箱顺序优化环节 PSO-HGA 算法系列中, PSO-HGA-S1 算法最优。在带有改进算子的混合遗传算法中, IPO-HGA-S1 算法最优。**结论** 文中设计的混合遗传算法能很好地提高电商仓储领域打包环节包装箱的利用率。

**关键词:** 多箱型三维装箱; 混合遗传算法; 混合整数规划; 算法优化

**中图分类号:** TB485.3 **文献标识码:** A **文章编号:** 1001-3563(2022)21-0213-11

**DOI:** 10.19554/j.cnki.1001-3563.2022.21.028

## Research on Multi Box 3D Packing Problem Based on Hybrid Genetic Algorithm

ZHOU Li<sup>1</sup>, YANG Jiang-long<sup>1</sup>, ZHAO Jun-hui<sup>1</sup>, LIU Hu-wei<sup>1,2</sup>, WANG Fan<sup>1</sup>

(1. School of Information, Beijing Wuzi University, Beijing 101149, China; 2. School of Management Engineering, Capital University of Economics and Business, Beijing 100070, China)

**ABSTRACT:** The works aims to improve the utilization rate of packing cases in the packaging link of e-commerce warehousing. Aiming at the 3D packing problem of multi box and multi-item types in e-commerce warehousing, a mathematical model of mixed integer programming was established, and a packing process module algorithm based on heuristic empirical rules and the combination of multiple operators was designed. The hybrid genetic algorithm for multi box 3D packing problem was designed from packing sequence and improved operator to optimize the packing scheme. Experiments showed that PSO-HGA-S1 algorithm was the best among the PSO-HGA algorithm series in the optimization of packing sequence. Among the hybrid genetic algorithms with improved operators, IPO-HGA-S1 algorithm was the best. The hybrid genetic algorithm designed in the work can improve the utilization rate of packing cases in the packaging link of e-commerce warehousing.

**KEY WORDS:** multi box 3D packing; hybrid genetic algorithm; mixed integer programming; algorithm optimization

在电商仓储领域中, 物品装箱操作速度高度依赖工作人员的经验 and 熟练程度, 新手装箱的速度非常缓慢, 且装箱方案的空间利用率往往较低, 导致成本增加。同时, 电商仓储领域也面临“无人仓”与“智能化”的转型趋势, 有待突破“人工速度慢, 装箱效率低”的瓶颈。因此研究自动化装箱场景下所需的装箱方案,

设计电商仓储领域实际有效的智能装箱算法, 提升装箱工作效率就迫在眉睫。

以维度划分装箱问题是非常常见的一种研究分类方式, 主要包括一维装箱问题<sup>[1-2]</sup>、二维装箱问题<sup>[3-4]</sup>和三维装箱问题 (3D Bin Packing Problems, 3D-BPP)。

收稿日期: 2022-06-21

基金项目: 北京社科基金重点项目 (18GLA009)

作者简介: 周丽 (1978—), 女, 教授, 主要研究方向为物流与供应链管理。

通信作者: 柳虎威 (1994—), 男, 博士生, 主要研究方向为物流与供应链管理。

关于二维装箱问题(又称“二维下料问题”)的研究非常丰富,不论是二维侧刀式背包包装问题<sup>[5]</sup>,还是带切刀约束的二维带材包装问题<sup>[6]</sup>,都要求一组物品必须在不重叠的情况下被打包到一个二维的箱子中<sup>[7-8]</sup>。根据物品形状不同,二维装箱问题又可以具体分为规则物品二维装箱<sup>[5,9]</sup>和不规则物品二维装箱。关于二维不规则物品装箱问题(Two-Dimensional Irregular Packing Problem, 2DIBPP)的研究,有的涉及到下料问题,要切割的小块具有不规则形状<sup>[10]</sup>;有的涉及到条形包装问题,需要在一个矩形物体内放置一组多边形<sup>[11]</sup>;还有的研究涉及开放维度的不规则物品二维装箱问题<sup>[12]</sup>,非凸不规则物品的二维装箱问题<sup>[13-14]</sup>,以及均匀多箱不规则物品的二维装箱问题<sup>[15]</sup>。

三维装箱问题是经典的一维和二维装箱问题的自然推广,很多研究将三维装箱问题转化为二维装箱问题进行求解<sup>[16]</sup>。近年来,在这一问题上的研究和出版物迅速增长<sup>[17]</sup>。其中,3D-BPP进一步又可以分为规则物品的装箱问题、不规则物品的装箱问题。规则物品的装箱问题根据物品和箱子类型多少以及形状大小的差异,又可分为同构装箱问题(Identical 3D-BPP)和异构装箱问题(Heterogeneous 3D-BPP),与物品分类已经引入的问题类别相似,将箱子划分为相同箱子(identical large objects)、弱异构箱子(weakly heterogeneous large objects)和强异构箱子(strongly heterogeneous large objects)。其中,“同构”是指箱子或物品类型单一且尺寸相同,“异构”是指箱子或物品类型多样且尺寸不同。有学者研究了单一箱型的三维装箱问题<sup>[18]</sup>,也有学者研究了弱异构物品同构箱子的三维装箱问题<sup>[19]</sup>和考虑了物品强弱异构的不同情况<sup>[20]</sup>。

在结合实际情况进行三维装箱的研究方面,有学者研究了三维切割和包装问题,强制执行非重叠约束<sup>[21]</sup>,有学者研究了物流平台的集装箱装载问题<sup>[22-23]</sup>。关于集装箱装载问题(Container Loading Problem, CLP),又可以细分为多集装箱装载<sup>[24]</sup>和零担装载(拼箱)<sup>[25]</sup>。集装箱装载问题<sup>[17,20]</sup>,在物流规划和调度中扮演重要角色<sup>[26]</sup>。货物被包装成标准集装箱,以便于用轮船、卡车或铁路车辆运输<sup>[27]</sup>,以及航空货物的三维装箱问题<sup>[28-29]</sup>。另外,多集装箱装载问题也成为研究的关注点<sup>[30-32]</sup>。

由于电商仓储领域包装箱型号与物品种类繁多,远非现有研究中的个位数,且现有研究主要优化目标为箱子的空间利用率,在装箱方案计算时间上并没有很苛刻的要求,而这并不能满足电商仓储领域装箱实践环节的时间要求。综上,文中基于电商仓储领域装箱实践的具体情境,以提高订单平均装载率为主要优化目标,在兼顾计算耗时的现实条件下,构建了混合整数规划模型,并设计了启发式装箱规则,然后通过混合遗传算法来优化物品装箱方案。

# 1 多箱型三维装箱混合整数规划模型

## 1.1 模型变量假设

设一个订单中有  $m$  件物品,有  $n$  种型号的箱子可供选择。物品和箱子的形状均为长方体。第  $i$  件物品的长、宽、高分别为  $l_i^I$ 、 $w_i^I$ 、 $h_i^I$ ,第  $j$  种箱子的长、宽、高分别为  $l_j^B$ 、 $w_j^B$ 、 $h_j^B$ 。其中,  $i=1,2,\dots,m$ ;  $j=1,2,\dots,n$ 。

### 1.1.1 箱子型号选择

设  $X_j^B$  为 0—1 变量,  $X_j^B = \begin{cases} 1 & \text{选中第 } j \text{ 种箱子} \\ 0 & \text{不选第 } j \text{ 种箱子} \end{cases}$ ,

并且  $\sum_{j=1}^n X_j^B = 1$ 。

设  $l^{B^*}$ 、 $w^{B^*}$ 、 $h^{B^*}$  分别为选中箱子的长、宽、高,其中

$$l^{B^*} = \sum_{j=1}^n X_j^B \cdot l_j^B, w^{B^*} = \sum_{j=1}^n X_j^B \cdot w_j^B, h^{B^*} = \sum_{j=1}^n X_j^B \cdot h_j^B。$$

假设物品装箱时箱子长、宽、高的方向不变。以选中箱子的长、宽、高为基本参照,建立三维直角坐标系,见图 1。箱子的长( $l^{B^*}$ )宽( $w^{B^*}$ )高( $h^{B^*}$ )分别对应  $x$  轴、 $y$  轴、和  $z$  轴。

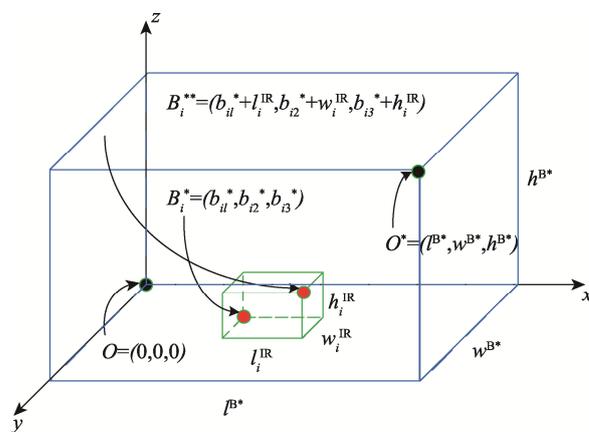


图 1 箱子及物品装箱空间位置示意图  
Fig.1 Schematic diagram of space location of boxes and articles

### 1.1.2 物品装箱位置选择

物品在箱子中的装载位置以物品装入箱子后的物品左后下角的位置坐标表示,如图 1 中  $B_i^*$  所示。设第  $i$  件物品在三维空间中初始左后下角位置坐标为  $B_i = (b_{i1}, b_{i2}, b_{i3})$ 。由于一个订单中的物品总数为  $m$  件,则物品在箱子中的装载位置共有  $m$  个,物品装箱的第  $i'$  个位置的坐标为  $B_{i'} = (b_{i'1}, b_{i'2}, b_{i'3})$ ,  $i'=1,2,\dots,m$ 。

物品装载位置选择的 0—1 矩阵  $P$  如下:

$$P = \begin{matrix} B_1' \\ B_2' \\ \vdots \\ B_m' \end{matrix} \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1m} \\ P_{21} & P_{22} & \cdots & P_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ P_{m1} & P_{m2} & \cdots & P_{mm} \end{bmatrix} \quad (1)$$

其中,

$$P_{i'i} = \begin{cases} 1 & \text{第 } i \text{ 件物品装入第 } i' \text{ 个位置} \\ 0 & \text{第 } i \text{ 件物品不装入第 } i' \text{ 个位置} \end{cases} \quad (2)$$

由于每件物品只能装入 1 个位置, 每个位置只能有 1 个位置装入, 则有:

$$\sum_{i'=1}^m P_{i'i} = 1, \sum_{i=1}^m P_{i'i} = 1 \quad (3)$$

第  $i$  物品在箱子中的最终装载位置坐标为

$B_i^* = (b_{i1}^*, b_{i2}^*, b_{i3}^*)$ , 则有  $B_i^* = \sum_{i'=1}^m P_{i'i} \cdot B_{i'}'$ , 即:

$$b_{i1}^* = \sum_{i'=1}^m P_{i'i} \cdot b_{i'1}', b_{i2}^* = \sum_{i'=1}^m P_{i'i} \cdot b_{i'2}', b_{i3}^* = \sum_{i'=1}^m P_{i'i} \cdot b_{i'3}' \quad (4)$$

### 1.1.3 物品装箱方向选择

一件物品在装箱时共有 6 种方向可供选择, 物品装箱的不同方向可以用物品在不同维度上长尺寸表示, 见图 2。

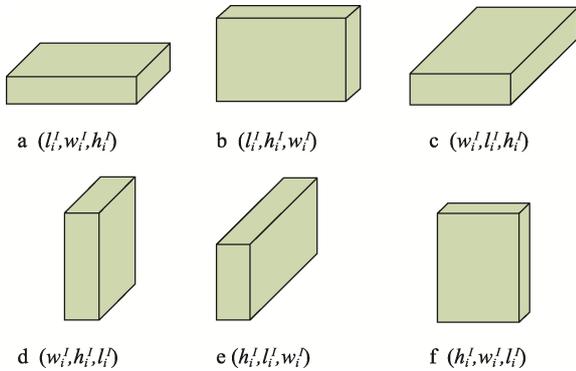


图 2 物品装箱旋转方向示意图

Fig.2 Schematic diagram of rotation direction of article packing

第  $i$  件物品装箱方向 0—1 选择矩阵  $R$  如下。

$$R = \begin{matrix} l_i^l & w_i^l & h_i^l \\ w_i^{iR} & l_i^{iR} & h_i^{iR} \\ h_i^{iR} & w_i^{iR} & l_i^{iR} \end{matrix} \begin{bmatrix} R_{11}^i & R_{12}^i & R_{13}^i \\ R_{21}^i & R_{22}^i & R_{23}^i \\ R_{31}^i & R_{32}^i & R_{33}^i \end{bmatrix} \quad (5)$$

式中:  $R_{r'r}^i = 0$  或  $1$ ,  $r', r = 1, 2, 3$ 。  $l_i^{iR}, w_i^{iR}, h_i^{iR}$ , 分别为第  $i$  件物品装箱方向下的长、宽、高尺寸。由于每件物品最终只能选择 1 种方向装入箱子, 每个装入箱子的物品必须确定 1 种方向, 则有:

$$\sum_{k=1}^3 R_{k'k}^i = 1, \sum_{k=1}^3 R_{kk}^i = 1 \quad (6)$$

确定物品装载方向后, 最终装入箱子时第  $i$  件物品在各个方向上的尺寸大小分别为:

$$l_i^{iR} = R_{11}^i \cdot l_i^l + R_{12}^i \cdot w_i^l + R_{13}^i \cdot h_i^l \quad (7)$$

$$w_i^{iR} = R_{21}^i \cdot l_i^l + R_{22}^i \cdot w_i^l + R_{23}^i \cdot h_i^l \quad (8)$$

$$h_i^{iR} = R_{31}^i \cdot l_i^l + R_{32}^i \cdot w_i^l + R_{33}^i \cdot h_i^l \quad (9)$$

## 1.2 装箱模型构建

### 1.2.1 模型优化目标及约束条件

从电商仓储领域的实际出发, 通过分析装箱数据发现, 每个订单都能被装入一个箱子中, 因此, 文中的问题转变为, 选择一个能够装入单个订单所有物品的箱子, 并使箱子的空间利用率最高。其中, 物品不能超出选中箱子的空间范围, 物品与物品之间不能出现空间位置的重叠。暂不考虑物品支撑、物品挤压、物品区隔等其他约束条件。

关于非越界约束, 如图 1 所示, 只要物品的左后下角和右前上角空间坐标, 即在  $x$  轴、 $y$  轴、 $z$  轴 3 个维度上的值, 均在箱子所围成的三维空间范围内, 则物品没有越界。其中,  $O$  为箱子左后下角的坐标, 即原点;  $O^*$  为箱子右前上角的坐标,  $B_i^{**}$  为第  $i$  件物品在箱子三维坐标系空间中右前上角的坐标。

关于物品之间的非重叠约束, 如图 3 所示, 第  $i$  件物品在  $x$  轴、 $y$  轴、 $z$  轴上占据的区间分别为  $[b_{i1}^*, b_{i1}^* + l_i^{iR}]$ ,  $[b_{i2}^*, b_{i2}^* + w_i^{iR}]$ ,  $[b_{i3}^*, b_{i3}^* + h_i^{iR}]$ , 其占据的空间范围是由  $x$  轴、 $y$  轴和  $z$  轴占据的区间交叉形成的。同理可得, 第  $i'$  件物品占据的三维空间范围是由区间  $[b_{i'1}^*, b_{i'1}^* + l_{i'}^{i'R}]$ 、 $[b_{i'2}^*, b_{i'2}^* + w_{i'}^{i'R}]$  和  $[b_{i'3}^*, b_{i'3}^* + h_{i'}^{i'R}]$  交叉得到的, 其中  $i' \neq i$ 。如果 2 件物品在三维空间中位置不存在重叠, 在以下 3 个条件中至少满足其中 1 个。

条件 1:  $b_{i1}^* + l_i^{iR} \leq b_{i'1}^*$  或  $b_{i'1}^* + l_{i'}^{i'R} \leq b_{i1}^*$ ;

条件 2:  $b_{i2}^* + w_i^{iR} \leq b_{i'2}^*$  或  $b_{i'2}^* + w_{i'}^{i'R} \leq b_{i2}^*$ ;

条件 3:  $b_{i3}^* + h_i^{iR} \leq b_{i'3}^*$  或  $b_{i'3}^* + h_{i'}^{i'R} \leq b_{i3}^*$ 。

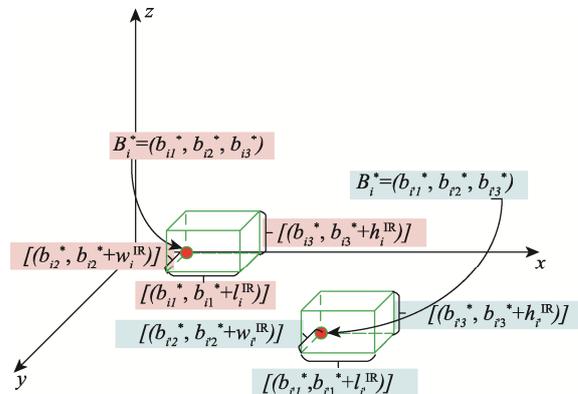


图 3 物品非重叠情况示意图

Fig.3 Schematic diagram of non-overlapping items

### 1.2.2 装箱模型构建

以订单物品装箱的剩余空间最小为优化目标, 构

建混合整数规划模型如下。

$$\min z = l^B \cdot w^B \cdot h^B - \sum_{i=1}^m l_i^{IR} \cdot w_i^{IR} \cdot h_i^{IR} \quad (10)$$

$$0 \leq b_{i1}^* \leq l^B \quad (11)$$

$$0 \leq b_{i2}^* \leq w^B \quad (12)$$

$$0 \leq b_{i3}^* \leq h^B \quad (13)$$

$$0 \leq b_{i1}^* + l_i^{IR} \leq l^B \quad (14)$$

$$0 \leq b_{i2}^* + w_i^{IR} \leq w^B \quad (15)$$

$$0 \leq b_{i3}^* + h_i^{IR} \leq h^B \quad (16)$$

$$b_{i1}^* + l_i^{IR} \leq b_{i1}^* + (1 - y_1) \cdot M \quad (17)$$

$$b_{i1}^* + l_i^{IR} \leq b_{i1}^* + (1 - y_2) \cdot M \quad (18)$$

$$b_{i2}^* + w_i^{IR} \leq b_{i2}^* + (1 - y_3) \cdot M \quad (19)$$

$$b_{i2}^* + w_i^{IR} \leq b_{i2}^* + (1 - y_4) \cdot M \quad (20)$$

$$b_{i3}^* + h_i^{IR} \leq b_{i3}^* + (1 - y_5) \cdot M \quad (21)$$

$$b_{i3}^* + h_i^{IR} \leq b_{i3}^* + (1 - y_6) \cdot M \quad (22)$$

$$y_1 + y_2 \leq 1 \quad (23)$$

$$y_3 + y_4 \leq 1 \quad (24)$$

$$y_5 + y_6 \leq 1 \quad (25)$$

$$y_1 + y_2 + y_3 + y_4 + y_5 + y_6 \geq 1 \quad (26)$$

$$M \text{ 为足够大的常数} \quad (27)$$

$$y_1, y_2, y_3, y_4, y_5, y_6 = 0 \text{ 或 } 1 \quad (28)$$

$$i \neq i'; i, i' = 1, 2, \dots, m \quad (29)$$

其中,式(10)为模型的目标函数,优化目标为订单装入箱子后,使剩余空间最小;式(11)至(13)为装入箱子中每件物品的左后下角坐标在箱子尺寸所允许的范围内,式(14)至(16)为箱子中每件物品的右前上角坐标在箱子尺寸所允许的空间范围内;式(17)至(25)分别为2件物品在x轴、y轴、z轴维度上所占据区间之间的关系,式(26)保证2件物品之间不存在空间重叠;式(27)至(29)规定了各个变量的取值范围。

## 2 多箱型三维装箱启发式算法设计

### 2.1 装箱过程模块启发式算子设计

本文设计了经验模拟算法(Empirical Simulation Algorithm,简称ESA),并基于物品装箱顺序选择模块和物品装箱位置选择模块对装箱方案进行优化。

物品装箱顺序选择是指每次选择即将装入箱子的SKU及物品。物品装箱顺序选择算子集合主要包括物品体积降序选择算子 $\alpha_1^u$ 和物品装箱顺序函数选择算子 $\alpha_2^u(\varphi^u)$ 。 $\alpha_1^u$ 的作用是将订单中的物品按照体积降序依次装入箱子。 $\alpha_2^u(\varphi^u)$ 是指,物品的装箱顺序由函数 $\varphi^u$ 所确定,其目的是通过调整物品装箱顺序,

优化物品装箱策略,提升订单装载率。

物品装箱位置的选择会随着装入箱子的物品增加而变得复杂,很多文献采用空间三分的方法进行空间划分与组合,再加上一些启发式规则用以确定物品的装箱位置。空间划分与组合算子集合主要包括空间利用率最高算子 $\alpha_1^e$ 、“柱-墙-块”生成算子 $\alpha_2^e$ 和“原点距离最近”算子 $\alpha_3^e$ 。 $\alpha_1^e$ 的作用是,计算当前物品体积与空间列表中可以容纳该物品的空间体积之比,选择比值小于等于1的空间中最大(即物品装入后空间利用率最高)并且可以容纳当前物品的空间,作为本次物品装箱的目标空间。 $\alpha_2^e$ 的作用是,筛选空间列表中能够装下当前物品的空间信息,优先选择其中空间左后下角三维坐标中y轴方向坐标最小的空间,其次选择x轴,最后选择z轴。 $\alpha_3^e$ 的作用是,在筛选空间列表中能够装下当前物品的空间信息时,优先选择其中空间左后下角三维坐标与原点之间欧式距离最短的坐标点所对应的空间。

### 2.2 多箱型三维装箱问题启发式算法设计

#### 2.2.1 启发式算法规则设定

电商仓储三维装箱问题启发式算法主要包括4个环节:箱子型号选择,物品装箱顺序选择,物品装箱位置选择与物品装箱方向选择。

1)箱子型号选择的启发式规则。计算每种箱子类型可容纳的体积,根据箱子容积大小从小到大排列。计算当前订单中所有物品体积总和。从容积大于等于物品体积总和的箱子中容积最小的箱子开始尝试,如果当前选中的箱子能够装入所有物品,则停止计算。否则,继续尝试超过订单物品总体积的次小箱子继续尝试。以此类推,直到找到合适的箱子为止。

2)物品装箱顺序的启发式规则。选择优先装入体积较大的物品,然后再依次装入体积较小的物品,如果在物品装箱过程中,出现有物品不能装入当前选中的箱子,则更换更大的箱子重新尝试。

3)物品装箱位置的启发式规则。物品装箱位置的选择与非越界约束和非重叠约束紧密相关。文中选择“左后下角”的启发式策略,即将在箱子空间中物品的左后下角位置坐标作为物品装载位置的参考,并从箱子本身三维空间的左后下角(即原点)开始进行装箱。

4)物品装箱方向的启发式规则。物品装载方向的确是根据图2中展示的6种物品方向,依次进行物品装箱尝试。得到可行的装箱方向则停止计算,否则继续尝试剩余物品方向。如果所有的装箱方向全部进行尝试并且没有找到可行的情况,则说明当前装载位置不可行。继续尝试其他装载位置,直到找到可行的装载位置和装箱方向,停止计算。

根据上述装箱问题的启发式规则,设计多种箱型三维装箱启发式算法的整体结构框架,见图4。

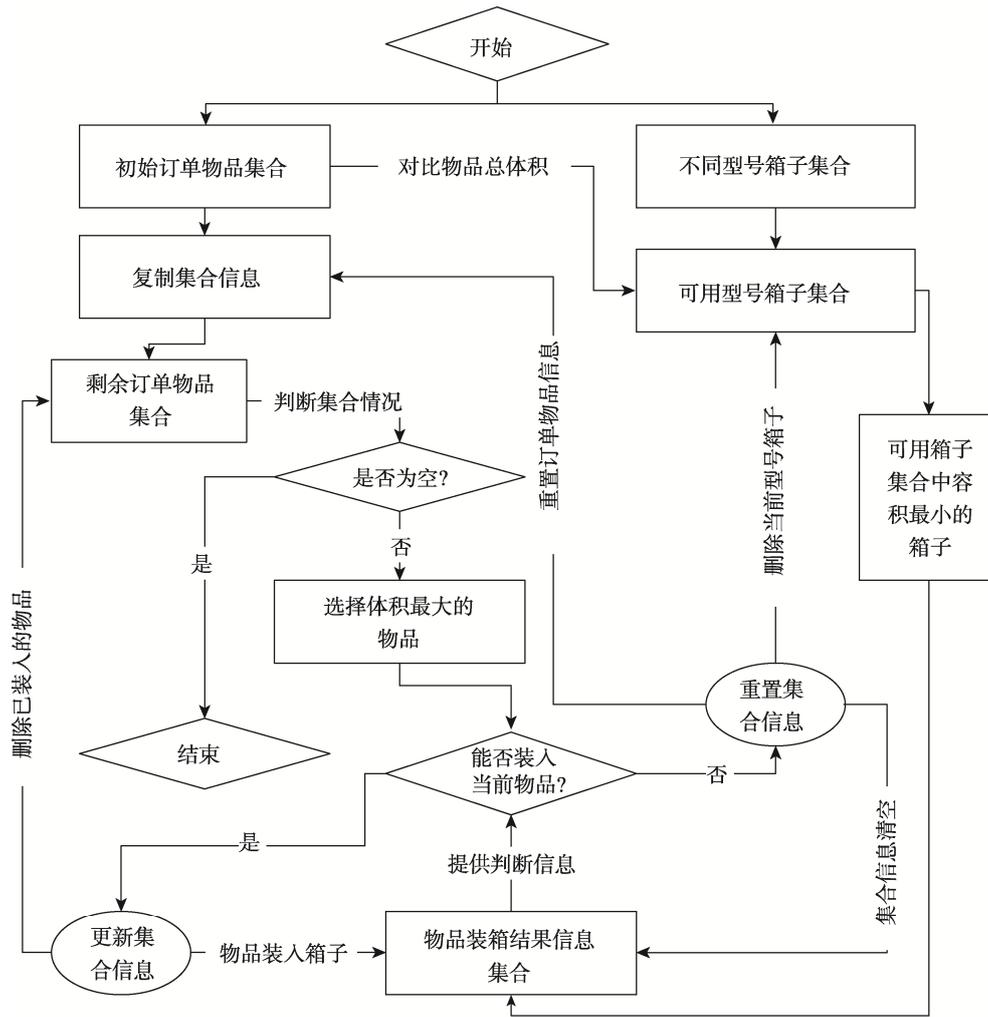


图 4 多种箱型三维装箱启发式算法框架图

Fig.4 Framework of 3D packing heuristic algorithm for various box types.

### 2.2.2 装箱问题的启发式算法集合

在装箱问题过程模块化的设计思路指导下, 针对电商仓储领域的多箱型三维装箱问题, 结合装箱过程中各个环节的设计以及过程模块中的算子集合, 构建了动态复合块装箱的 S 类算法。根据物品装箱位置选择方式的不同, 将动态复合块装箱的 S 类算法分为 ESA-SHA-S1 算法, ESA-SHA-S2 算法和 ESA-SHA-S3 算法。其中, 每种算法包含的基本算子集合如表 1 所示。

表 1 动态复合块装箱的 S 类算法算子集合  
Tab.1 Set of S-class algorithm operators for dynamic composite block packing

算法名称	算子集合
ESA-SHA-S1	$\{o_1^u, o_1^e\}$
ESA-SHA-S2	$\{o_1^u, o_2^e\}$
ESA-SHA-S3	$\{o_1^u, o_3^e\}$

## 3 多箱型三维装箱问题混合遗传算法

### 3.1 装箱顺序优化的混合遗传算法

#### 3.1.1 装箱顺序优化的混合遗传算法基本框架设计

在动态复合块装箱的 S 类算法基础上, 针对物品装箱顺序选择环节进行优化, 设计装箱顺序优化的混合遗传算法 (Hybrid Genetic Algorithm For Packing Sequence Optimization, PSO-HGA)。该算法设计的具体内容如下。

1) 设计染色体编码。本算法染色体采用实数编码, 以订单中 SKU 的序列作为编码对象, 对物品装箱顺序进行排序。通过对 SKU 序列中的元素顺序重新排列, 可以得到不同的订单物品装箱顺序, 以此作为本算法的单个染色体编码。

2) 生成初始种群。所设计的混合遗传算法采用最常见的随机排序方法。设种群中包含的染色体的数量为  $g$ , 将 SKU 序列中的元素随机排序  $g$  次, 得到  $g$

个元素顺序不同的染色体，从而形成初始群体。

3) 确定适应度函数。该算法将订单装载率作为个体适应度，则订单装载率的计算过程为适应度函数。根据模型，适应度函数可以具体表示为  $f$ ，则有：

$$f = \frac{\sum_{i=1}^m l_i^{IR} \cdot w_i^{IR} \cdot h_i^{IR}}{l^B \cdot w^B \cdot h^B} \quad (30)$$

4) 进行个体评价。根据种群中染色体信息所确定的物品装箱顺序，在动态复合块装箱的 S 类算法框架下，计算订单装载率，可以得到每个染色体的适应度函数值  $f$ 。 $f$  越大，说明订单装载率越高，反之则订单装载率越低。

5) 设计选择算子。文中设计了放大优势型轮盘赌 (Enlarge Advantage Roulette, EAR) 算子，即在对群体适应度的值进行调整的基础上，然后再使用轮盘赌的方式筛选优势个体。调整群体适应度的规则是，在较小的适应度值的基础上增加一个较小的数值，在较大适应度值的基础上增加一个较大的数值，使得较大适应度值对应的个体有更大的概率被选中。

6) 设计交叉算子。文中设计了“单点交叉，双位交换” (Single Point Crossing and Double Bit Switching, SPCDBS) 算子。首先，将群体中的染色体随机分成数量相同的两组；其次，然后从每组中随机取出 1 条染色体组成一个染色体对；最后，以一定的交叉概率，对取出的染色体对进行交叉操作，得到 2 条新的染色体，见图 5。

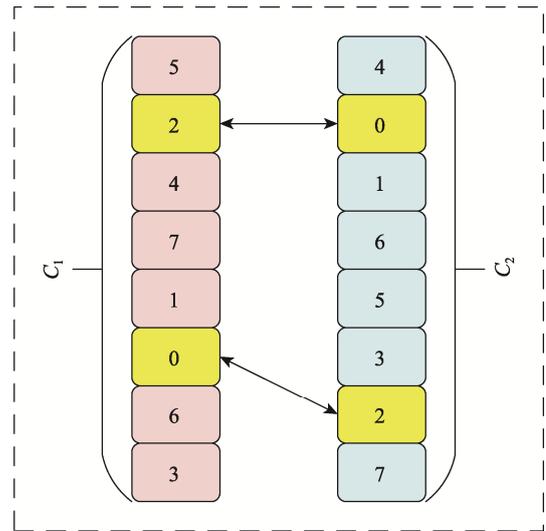
7) 设计变异算子。文中设计了“单点变异，随机顺序” (Single point variation and sequential random, SPVSR) 算子，即从种群中依次提取每个染色体，以一定的变异概率对取出的染色体进行变异操作。染色体变异机制见图 6。

8) 迭代终止条件。该模型求解的终止条件综合考虑最大迭代次数 ( $T$ ) 与订单装箱方案平均计算耗时 2 个因素，以优化订单装载率为主要目标，适当放宽计算耗时方面的要求。在订单物品装箱方案计算耗时为分钟级的时间内，确定最大迭代次数作为终止条件。

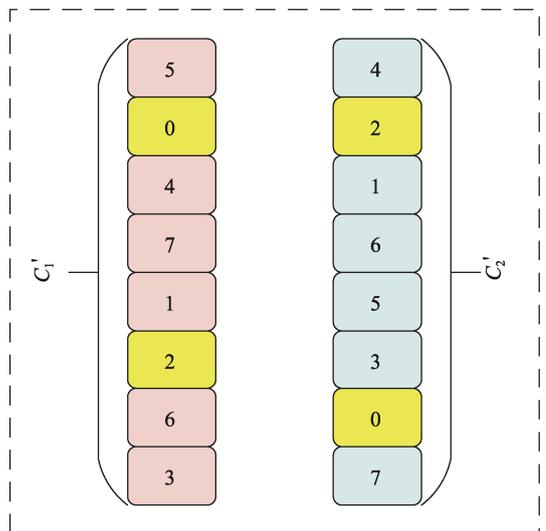
### 3.1.2 动态复合块装箱的 PSO-HGA 算法

结合动态复合块装箱的 S 类算法中各个算法的基本算子，可以进一步构建 PSO-HGA 算法系列，具体包括：PSO-HGA-S1 算法、PSO-HGA-S2 算法和 PSO-HGA-S3 算法。

其中，PSO-HGA-S1 算法是以 ESA-SHA-S1 算法为基础，用  $\alpha_2^u(\varphi^u)$  代替  $\alpha_1^u$ ，将混合遗传算法作为  $\varphi^u$  函数，根据搜索得到的物品装箱顺序，求得更优的订单装载率。PSO-HGA-S2 算法与 PSO-HGA-S3 算法同理。因此，PSO-HGA 算法系列基本上继承了动态复合块装箱 S 类算法的很多优点，例如计算耗时短，订单装载率高，并在适当放宽计算耗时的条件下，进一步优化提升装载率。



子图 (1)



子图 (2)

图 5 SPCDBS 算子染色体交叉机制示意图  
Fig.5 SPCDBS schematic diagram of operator chromosome crossover mechanism

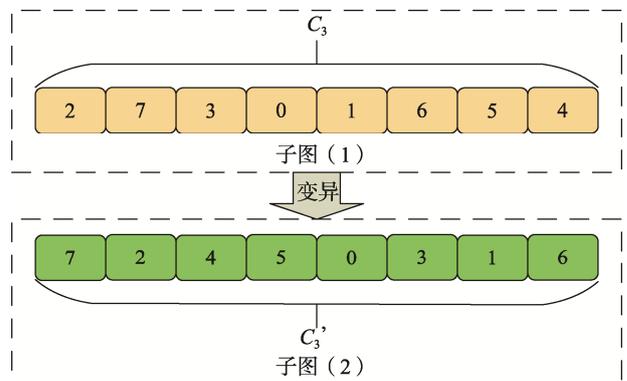


图 6 SPVSR 算子染色体变异机制  
Fig.6 SPVSR schematic diagram of operator chromosome variation mechanism

### 3.2 带有改进算子的混合遗传算法

#### 3.2.1 初始群体优选的混合遗传算法

初始群体优选的混合遗传算法 (Hybrid genetic algorithm for initial population optimization, 简称 IPO-HGA), 主要是基于 PSO-HGA 算法框架, 针对其中“生成初始种群”的环节, 不再是采用随机策略生成初始种群, 而是根据  $\alpha_i^a$  的规则, 按照物品 SKU 的体积进行从大到小排序, 从而得到染色体信息。将该条由启发式规则得到的染色体信息复制  $g$  次, 生成初始群体。

#### 3.2.2 带邻域搜索算子的混合遗传算法

带邻域搜索算子的混合遗传算法 (Hybrid genetic algorithm with neighborhood search operator, 简称 NSO-HGA), 是在 PSO-HGA 算法中加入邻域搜索算子 NSO, 以弥补其局部搜索能力的不足。如图 7 所示, 针对当前最优染色体的实数编码序列 ( $C_o$ ), 依次仅交换相邻 2 个位置的元素所得到的新染色体集合 ( $C_n$ ), 记为原染色体的邻域范围。计算新得到的邻域集合中所有染色体的适应度函数值, 并与原有群体中的染色体合并, 然后采用 EAR 算子选择得到  $g$  个染色体, 从而形成新的群体并应用于后续计算过程。

#### 3.2.3 综合多种算子的混合遗传算法

综合多种算子的混合遗传算法 (Hybrid genetic algorithm integrating multiple operators, 简称 IMO-HGA), 还综合了 PSO-HGA 算法, IPO-HGA 算法的 IPO 算子和 NSO-HGA 算法的 NSO 算子, 将改进算子作为  $\alpha_2^a(\varphi^a)$  中的  $\varphi^a$  函数, 其他装箱方案的计算环节不变。算法之间两两组合, 得到带有改进算子的混合遗传算法系列, 如表 2 所示。

## 4 实例验证与分析

### 4.1 混合遗传算法超参数确定

为了确定混合遗传算法系列的超参数设置, 选择 PSO-HGA-S1 算法作为典型代表, 将京东企业装箱实践数据集作为测试数据, 其中各种箱型包装箱尺寸信息见表 3。文中的研究主要针对电商的 B2B 业务领域。与个人订单相比, 订单中物品的种类和数量较多, 属于异构性较强的装箱问题。文中所采用的基本数据量为企业脱敏后的 1 000 个订单, 并在此基础上以该业务场景的订单统计特征为依据, 可以生成任意数量的测试订单数据集。尝试不同的超参数组合, 通过计算实验最终确定各个超参数的取值, 在设定一般取值范围<sup>[33]</sup>的基础上, 结果见表 4。

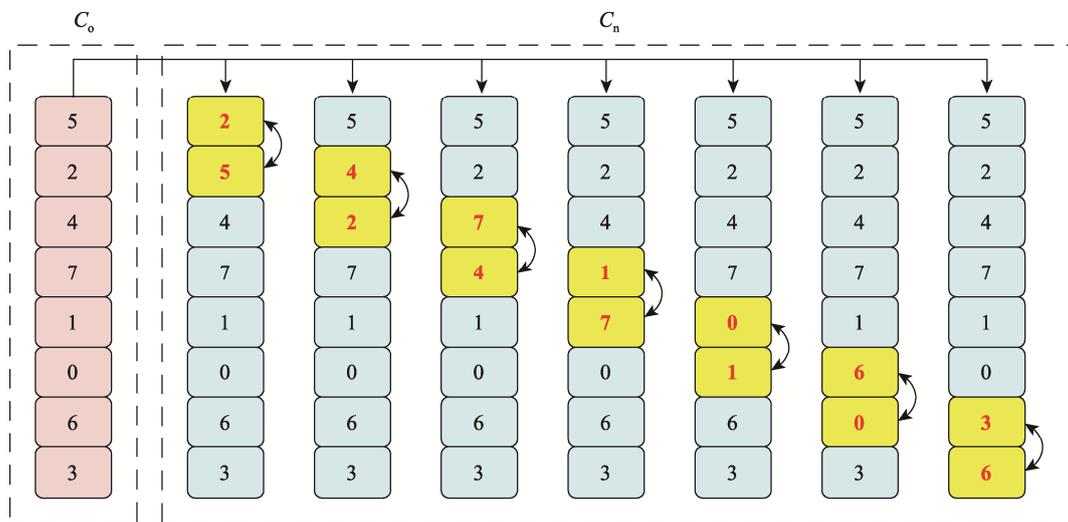


图 7 原染色体与邻域染色体编码信息变换情况

Fig.7 Transformation of coding information between proto chromosome and neighborhood chromosome.

表 2 带有改进算子的混合遗传算法系列  
Tab.2 Series of hybrid genetic algorithms with improved operator

算法	ESA-SHA-S1	ESA-SHA-S2	ESA-SHA-S3
PSO-HGA	IPO-HGA-S1	IPO-HGA-S2	IPO-HGA-S3
NSO-HGA	NSO-HGA-S1	NSO-HGA-S2	NSO-HGA-S3
IMO-HGA	IMO-HGA-S1	IMO-HGA-S2	IMO-HGA-S3

表3 各种箱型包装箱尺寸信息  
Tab.3 Size information of various box types

编号	长	宽	高
1	120	100	55
2	140	100	85
3	165	120	55
4	165	150	110
5	200	100	90
6	200	140	70
7	200	150	150
8	245	120	90
9	260	175	130
10	270	200	90
11	290	170	190
12	300	200	170
13	300	240	220
14	330	170	260
15	330	330	260
16	380	275	150
17	400	180	140
18	400	330	220
19	410	350	360
20	430	360	280
21	550	400	400
22	650	550	400

表4 混合遗传算法超参数设置情况  
Tab.4 Super parameter setting of hybrid genetic algorithm.

超参数变量	超参数含义	取值范围	变化步长	最终参数取值
$M$	种群个体数量	10 ~ 30	10	10
$T$	最大迭代次数	10 ~ 50	20	30
$P_c$	个体交叉概率	0.4 ~ 0.99	0.25	0.65
$P_m$	个体变异概率	0.01 ~ 0.15	0.045	0.10

由计算实验结果可知,遗传算法参数确定测试集的平均订单装载率为 82.61%, 订单装箱方案平均计算耗时为 39.54 s。经过 108 组参数组合计算实验,最终确定参数取值,如表 5 所示,群体数量 ( $M$ ) 为 10, 迭代次数 ( $T$ ) 为 30, 交叉概率 ( $P_c$ ) 为 0.65, 变异概率 ( $P_m$ ) 为 0.1。

## 4.2 混合遗传算法计算实验结果

### 4.2.1 装箱顺序优化的混合遗传算法实验结果

根据表 4 得到的混合遗传算法超参数设置数值,

在 JD 数据集上采用 PSO-HGA 算法系列进行计算实验,得到计算结果见表 5。其中,PSO-HGA-S1 算法在订单平均装载率指标上表现最佳,达到了 70.92%,其他算法虽然略有差距,但整体上都达到了 70%以上的装载率;另外,在订单平均计算耗时方面,PSO-HGA-S1 的计算耗时最短(装载率也最低)为 39.22 s,其他 2 个算法的计算耗时都超过了 40 s,虽然将订单装载方案计算耗时控制在了 1 分钟以内,但对于计算效率要求较高的场景还是存在一定劣势。

表5 JD数据集 PSO-HGA 算法系列计算结果  
Tab.5 Calculation results of PSO-HGA algorithm series of JD dataset

算法	订单平均装载率/%	订单平均计算耗时/s
PSO-HGA-S1	<b>70.92</b>	43.09
PSO-HGA-S2	70.67	48.07
PSO-HGA-S3	70.04	<b>39.22</b>

在 JD 数据集上采用 PSO-HGA 算法系列进行迭代,订单平均装载率的变化结果见图 8。其中,在相同迭代次数下,PSO-HGA-S1 算法的表现明显优于其他算法,而 PSO-HGA-S3 算法在整个迭代寻优过程中的表现都低于其他 2 个算法。究其原因,主要是装箱位置选择算子的不同,PSO-HGA-S1 算法采用的是  $o_1^o$ ,而 PSO-HGA-S2 算法和 PSO-HGA-S3 算法采用的分别是  $o_2^o$  和  $o_3^o$ 。由此可知,在 PSO-HGA 算法框架下, $o_1^o$  算子要比  $o_2^o$  算子和  $o_3^o$  算子更能有效提高订单平均装载率。

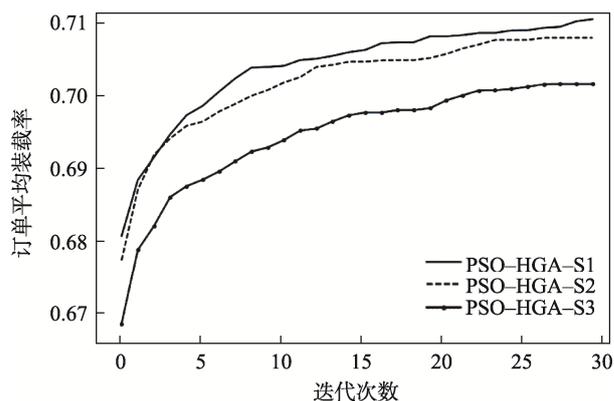


图8 JD数据集 PSO-HGA 算法系列迭代过程  
Fig.8 Iterative process of PSO-HGA algorithm series of JD dataset

### 4.2.2 带有改进算子的混合遗传算法实验结果

在 JD 数据集上采用带有改进算子的混合遗传算法计算订单实例的装箱方案,计算结果如表 6 所示。其中,IPO-HGA-S1 算法的表现最佳,订单平均装载率达到了 70.93%,订单平均计算耗时为 40.57 s,与其他带有改进算子的混合遗传算法相比,计算耗时仅

略高于 IPO-HGA-S3 算法, 比其他算法的计算耗时均短, 并且低于 PSO-HGA-S1 算法的订单平均计算耗时。在所有带改进算子的混合遗传算法中, NSO-HGA 类算法整体表现最差, 订单平均装载率低于 IMO-HGA 类算法, 且订单平均计算耗时高于 IPO-HGA 类算法; IPO-HGA 算法虽然有较高的订单平均装载率, 但是计算耗时也是最长的; IPO-HGA 算法的订单平均计算耗时远低于 NSO-HGA 类算法和 IMO-HGA 类算法, 并且 IPO-HGA-S1 算法与 PSO-HGA 算法系列相比, 以更短的计算耗时得到了更高的订单平均装载率。

表 6 JD 数据集带有改进算子的 HGA 算法系列计算结果  
Tab.6 Calculation results of HGA algorithm series with improved operator in JD dataset

算法	平均装载率/%	平均计算耗时/s
IPO-HGA-S1	<b>70.93</b>	<b>40.57</b>
IPO-HGA-S2	70.77	45.01
IPO-HGA-S3	70.17	36.18
NSO-HGA-S1	70.72	108.86
NSO-HGA-S2	70.77	123.08
NSO-HGA-S3	70.09	100.83
IMO-HGA-S1	70.91	122.52
IMO-HGA-S2	70.81	138.96
IMO-HGA-S3	70.51	102.37

在 JD 数据集上采用带有改进算子的混合遗传算法计算订单装箱实例, 以订单平均装载率为适应度函数值, 迭代过程如图 10 所示。其中, IPO-HGA-S3 算法、NSO-HGA-S3 算法与 IMO-HGA-S3 算法的表现较差, 而这 3 种算法的共同点是在物品装箱位置选择时都采用了  $\alpha_3^s$ , 说明在带有改进算子的混合遗传算法框架下,  $\alpha_3^s$  算子依然表现较差, 这一点与其在 PSO-HGA 算法系列中的情况相同。IPO-HGA-S1 算法与 IMO-HGA-S1 算法是带有改进算子的混合遗传算法中取得最高订单平均装载率的 2 个算法, 其共同点是在物品装箱位置选择环节都采用了  $\alpha_1^s$ , 并且初始种群生成环节都采用了 IPO 算子, 这说明这 2 种算子的作用相较于同类型的其他算子在 JD 数据集上的订单平均装载率指标表现更好。

另外, NSO-HGA 类算法与 IMO-HGA 算法的订单平均计算耗时均超过了 1 min, 最长的 IMO-HGA-S2 算法更是达到 2 min 以上, 主要原因是这两类算法都采用了 NSO 算子对每次迭代得到的代际最优个体进行领域搜索, 根据染色体邻域搜索编码信息, NSO 算子的加入相当于使每个迭代步骤增加了 70% 的个体数量, 加上与 EAR、SPCDBS 以及

SPVSR 之间的相互作用, 因此导致计算耗时大幅增加, 但是, 从迭代过程与最终结果来看, 相较于 IPO 算子, NSO 算子在提升订单平均装载率方面的作用较弱。

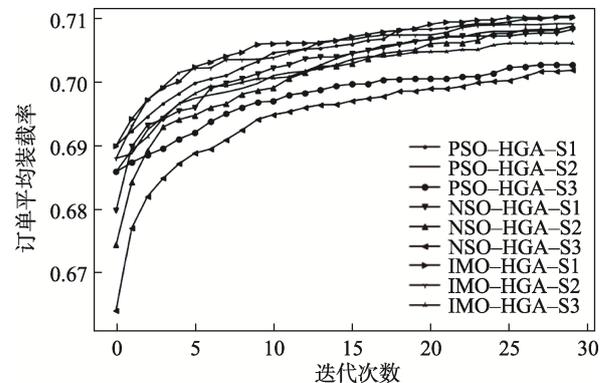


图 9 JD 数据集带有改进算子的 HGA 算法系列迭代过程

Fig.9 Iterative process of HGA algorithm series with improved operator in JD dataset

## 5 结语

文中基于装箱问题领域已有的研究成果, 结合电商仓储领域的装箱实践问题, 提出了多箱型三维装箱混合整数规划模型。基于空间划分的思想与物品非重叠非越界约束设计启发式规则, 针对多种箱型的三维装箱问题, 设计了多种启发式算子集合与组合设计, 提出装箱过程模块算法体系, 保证了装箱方案的装载率与计算耗时能够满足现实需要, 达到了秒级的响应速度。

文中研究还有更多可以改进的地方, 例如, 在算法优化方面, 文中仅对过程模块算法体系中的装箱顺序算法环节进行了智能化改造, 对于多个算法模块共同联合优化由于求解搜索空间过于庞大尚未涉及。后续研究可以对不同环节的算子进行算法优化, 如物品装箱位置选择与物品装箱方向选择等, 也可以对于多个算法模块共同联合优化。

## 参考文献:

- [1] GILMORE P C, GOMORY R E. A Linear Programming Approach to the Cutting-Stock Problem[J]. Operations Research, 1961, 9(6): 849-859.
- [2] 巩梨, 王文璨, 刘林忠. 多目标一维装箱问题模型算法研究[J]. 计算机应用研究, 2020, 37(S2): 144-146.  
GONG Li, WANG Wen-can, LIU Lin-zhong. Research on Algorithm of Multi-Objective One-Dimensional Packing Problem Model[J]. Computer Application Research, 2020, 37(S2): 144-146

- [3] WEI Li-jun, HU Qian, LIM A, et al. A Best-Fit Branch-and-Bound Heuristic for the Unconstrained Two-Dimensional Non-Guillotine Cutting Problem[J]. *European Journal of Operational Research*, 2018, 270(2): 448-474.
- [4] 尚正阳, 黄秋妍, 康正阳, 等. 一刀切约束下的二维装箱问题高效求解算法[J]. *包装工程*, 2021, 42(7): 231-238.  
SHANG Zheng-yang, HUANG Qiu-yan, KANG Zheng-yang, et al. Efficient Heuristic Algorithm for the Two-Dimensional Guillotine Rectangular Bin Packing Problem[J]. *Packaging Engineering*, 2021, 42(7): 231-238.
- [5] WEI L, LIM A. A Bidirectional Building Approach for the 2D Constrained Guillotine Knapsack Packing Problem[J]. *European Journal of Operational Research*, 2015, 242(1): 63-71.
- [6] WEI L, TIAN T, ZHU W, et al. A Block-based Layer Building Approach for the 2D Guillotine Strip Packing Problem[J]. *European Journal of Operational Research*, 2014, 239(1): 58-69.
- [7] TOLEDO F M, CARRAVILLA M A, RIBEIRO C, et al. The Dotted-Board Model: A New MIP Model for Nesting Irregular Shapes[J]. *International Journal of Production Economics*, 2013, 145(2): 478-487.
- [8] MUNDIM L R, ANDRETTA M, DE QUEIROZ T A. A Biased Random Key Genetic Algorithm for Open Dimension Nesting Problems Using No-Fit Raster[J]. *Expert Systems With Applications*, 2017, 81: 358-371.
- [9] ZHANG D, SHI L, LEUNG S, et al. A Priority Heuristic for The Guillotine Rectangular Packing Problem[J]. *Information Processing Letters*, 2016, 116(1): 15-21.
- [10] COSTA M T, GOMES A M, OLIVEIRA J F. Heuristic Approaches to Large-Scale Periodic Packing of Irregular Shapes on a Rectangular Sheet[J]. *European Journal of Operational Research*, 2007, 192(1): 29-40.
- [11] JUNIOR B A, PINHEIRO P R, SARAIVA R D. A Hybrid Methodology for Tackling the Irregular Strip Packing Problem[J]. *IFAC Proceedings Volumes*, 2013, 46(7): 396-401.
- [12] PINHEIRO P R, AMARO JÚNIOR B, SARAIVA R D. A Random-Key Genetic Algorithm for Solving the Nesting Problem[J]. *International Journal of Computer Integrated Manufacturing*, 2016, 29(11): 1159-1165.
- [13] ALVAREZ-VALDES R, MARTINEZ A, TAMARIT, J M. A Branch & Bound Algorithm for Cutting and Packing Irregularly Shaped Pieces[J]. *International Journal of Production Economics*, 2013, 145(2): 463-477.
- [14] MARTINEZ-SYKORA A, ALVAREZ-VALDES R, BENNELL J, et al. Matheuristics for the Irregular Bin Packing Problem with Free Rotations[J]. *European Journal of Operational Research*, 2017, 258(2): 440-455.
- [15] ABEYSOORIYA R P, BENNELL J A, MARTINEZ-SYKORA A. Jostle Heuristics for the 2D-Irregular Shapes Bin Packing Problems with Free Rotation[J]. *International Journal of Production Economics*, 2018, 195: 12-26.
- [16] SAMIR E, FATMA G, BURAK Y. Three-Dimensional Bin Packing and Mixed-Case Palletization[J]. *INFORMS Journal on Optimization*, 2019, 1(4): 323-352.
- [17] ZHAO X, BENNELLA J A, BEKTAS T, et al. A Comparative Review of 3D Container Loading Algorithms[J]. *International Transactions in Operational Research*, 2016, 23: 287-320.
- [18] ZHANG D F, PENG Y, LEUNG S.C.H. A Heuristic Block-loading Algorithm Based on Multi-layer Search for the Container Loading Problem[J]. *Computers & Operations Research*, 2012, 39(10): 2267-2276.
- [19] ZHAO H, SHE Q, ZHU C, et al. Online 3D Bin Packing with Constrained Deep Reinforcement Learning[C]// *Proceedings of the AAAI Conference on Artificial Intelligence*. 2021, 35(1): 741-749.
- [20] FANSLAU T, BORTFELDT A. A Tree Search Algorithm for Solving the Container Loading Problem[J]. *INFORMS Journal on Computing*, 2010, 22(2): 222-235.
- [21] SILVA E F, TOFFOLO T A M, WAUTERS T. Exact Methods for Three-Dimensional Cutting and Packing: A Comparative Study Concerning Single Container Problems[J]. *Computers and Operations Research*, 2019, 109: 12-27.
- [22] IWASAWA H, HU Yan-nan, HASHIMOTO H, et al. A Heuristic Algorithm for the Container Loading Problem with Complex Loading Constraints[J]. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, 2016, 10(3): 1-5.
- [23] RAMOS A G, SILVA E, OLIVEIRA J F. A New Load Balance Methodology for Container Loading Problem in Road Transportation[J]. *European Journal of Operational Research*, 2018, 266(3): 1140-1152.
- [24] PAQUAY C, LIMBOURG S, SCHYNS M. A Tailored Two-Phase Constructive Heuristic for the Three-Dimensional Multiple Bin Size Bin Packing Problem with Transportation Constraints[J]. *European*

- Journal of Operational Research, 2018, 267(1): 52–64.
- [25] JAMRUS T, CHIEN C F. Extended Priority-based Hybrid Genetic Algorithm for the Less-than-container Loading Problem[J]. Computers & Industrial Engineering, 2016, 96: 227–236.
- [26] HUANG Y H, HWANG F J, LU H C. An Effective Placement Method for the Single Container Loading Problem[J]. Computers & Industrial Engineering, 2016, 97: 212–221.
- [27] NGOI B K A, TAY M L, CHUA E S. Applying Spatial Representation Techniques to the Container Packing Problem[J]. Int. J. Prod. Res. 1994, 32 (1): 111–123.
- [28] PAQUAY C, SCHYNS M, LIMBOURG S. A Mixed Integer Programming Formulation for the Three-Dimensional Bin Packing Problem Deriving from an Air Cargo Application[J]. International Transactions in Operational Research, 2016, 23(1/2): 187–213.
- [29] 陈丙成, 李艳华. 基于启发式算法的单航空集装器的装箱优化设计[J]. 包装工程, 2020, 42(17): 244–251.  
CHEN Bing-cheng, LI Yan-hua. Optimal Design of Packing for Single Unit Load Device Based on Heuristic Algorithm[J]. Packaging Engineering, 2020, 42(17): 244–251.
- [30] CORRECHER J, ALONSO M, PARREÑO F, et al. Solving a Large Multicontainer Loading Problem in the Car Manufacturing Industry[J]. Computers and Operations Research, 2017, 82: 139–152.
- [31] ALONSO M, ALVAREZ-VALDES R, IORI M, et al. Mathematical Models for Multi Container Loading Problems with Practical Constraints[J]. Computers & Industrial Engineering, 2018, 127: 722–733.
- [32] 张长勇, 刘佳瑜. 基于混合遗传算法的多箱型集装箱装载问题分析[J]. 北京航空航天大学学报, 2022, 48(5): 747–755.  
ZHANG Chang-yong, LIU Jia-yu. Multi-Box Container Loading Problem Based on Hybrid Genetic Algorithm[J]. Journal of Beijing University of Aeronautics and Astronautics, 2022, 48(5): 747–755.
- [33] YANG J, ZHOU L, LIU H. Hybrid Genetic Algorithm-Based Optimisation of the Batch Order Picking in a Dense Mobile Rack Warehouse[J]. PLoS One, 2021, 16(4): 0249543.

责任编辑: 曾钰婵