Software Engineering and Applications 软件工程与应用, 2024, 13(4), 532-542
Published Online August 2024 in Hans. https://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2024.134056

Hans汉斯

基于启发式算法和遗传算法的三维装箱问题混合算法研究

刘城霞，王　萌，徐　倩

北京信息科技大学计算机学院，北京

收稿日期：2024年7月15日；录用日期：2024年8月20日；发布日期：2024年8月28日

摘　要

在三维装箱问题中，启发式算法和遗传算法都能够较好地解决问题。本文在可放置点生成的启发式算法和遗传算法的基础上，将二者结合生成了新的混合算法来研究三维装箱问题，并通过真实应用场景数据对新的混合算法进行测试，混合算法的装载率和原传统算法相比稳中有增，尤其是针对货物规格种类较多的情况下，混合算法的优势更为明显。

关键词

可放置点，启发式算法，遗传算法，混合算法，装载率

Research on Hybrid Algorithms for 3D Packing Problem Based on Heuristic Algorithm and Genetic Algorithm

Chengxia Liu, Meng Wang, Qian Xu

Computer School, Beijing Information and Technology University, Beijing

Received: Jul. 15th, 2024; accepted: Aug. 20th, 2024; published: Aug. 28th, 2024

Abstract

In the three-dimensional packing problem, both heuristic algorithms and genetic algorithms can effectively solve the problem. Based on the heuristic algorithm for generating placeable points and the genetic algorithm, this paper integrated the two algorithms into a new hybrid algorithm for the

文章引用: 刘城霞, 王萌, 徐倩. 基于启发式算法和遗传算法的三维装箱问题混合算法研究[J]. 软件工程与应用, 2024, 13(4): 532-542. DOI: 10.12677/sea.2024.134056

three-dimensional packing problem. The new hybrid algorithm was tested by real data in the application, and its loading rate was stable and had increased compared to the original traditional algorithm. Especially for cases with multiple types of goods, the advantages of the hybrid algorithm were more obvious.

Keywords

Placeable Points, Heuristic Algorithm, Genetic Algorithm, Hybrid Algorithm, Loading Rate

1. 引言

三维装箱问题(3D Packing Problem)，指的是在一个三维容器(如长方体)中，将多个不同形状、不同大小的物体(如立方体、长方体等)装载到其中，使得物体占用的体积最小，且不允许重叠或悬浮。该问题在物流和运输领域中是一个常见的优化问题，其目的是通过优化装箱策略，来提高装载效率、降低运输成本和提高物流效率。三维装箱问题是一个 NP 难度问题，意味着其时间复杂度会随着问题规模的增大而急剧增加，通常需要使用启发式算法[1]、数学规划法等方法来求解。

关于三维装箱，国内外学者们进行了许多探索研究。张钧和贺可太提出了一种新的混合算法，它将遗传算法、三空间分割启发式算法以及模拟退火算法糅合起来，可以较好地解决货物数量较多的多集装箱装载问题[2]。林云鹏等人提出了基于空间矩阵的三维装箱算法，在求解电商物流中，与传统的三维装箱算法相比更精确[3]。于明正等人提出了一种基于双层启发式遗传算法的三维装箱问题，可以有效避免陷入局部最优以及提高算法的稳定性，提升了最终装载方案的质量[4]。Fan 等人改进了分组遗传算法来解决三维装箱问题，他们构造了一种新的交叉算子并且设计了一种替换插入的方法来改进变异操作，该算法具有可靠的全局收敛性[5]。Cant 等人提出了一种生成最优集装箱装载布局的熵引导蒙特卡罗树的搜索方法，该算法生成的布局达到了拟合一个约束空间的目标，又具有一致性和整洁性[6]。张德富等人研究出一种多层启发式搜索算法，它能够通过多层搜索来评价可行块，据此来装载最合适的块[7]。李孙寸等人提出了多元优化的装箱算法，得到了比较理想的效果[8]。Xiang 等人研究了以最大化三维空间利用率为目标的集装箱装载优化问题，相较于一般的启发式算法，其装载率能提高 5%左右[9]。Kilincci 和 Medinoglu 提出了一种通过成形箱尺寸来解决三维装箱问题的高效算法，是一种包含静态规则的秩序化和整数非线性规划模型的混合算法，提高了集装箱装箱作业的有效性[10]。

本文在之前的学者们研究的基础上，结合生成可放置点的启发式算法和遗传算法优化装箱算法得到一种混合算法，该混合算法得到的装载率相对传统装载方法来说有一定的提升。

2. 生成可放置点的启发式算法

启发式算法(Heuristic Algorithm)是一种基于经验或启发式知识，使用控制结构求解问题的算法。启发式算法通常用来解决复杂的优化问题、组合问题、规划问题等，其优点在于求解过程简单、时间复杂度低、适用范围广等，但一般无法保证找到全局最优解。在解决装箱问题时，使用最多的启发式算法是将问题解决方案逐步构建出来的构造性启发式算法。

2.1. 三维装箱问题模型的建立

本文研究的三维装箱问题是：给定长宽高固定的集装箱和若干种货物，货物具有的属性是长、宽、高、重量、数量等。在进行货物装载的过程中，要求在一定的约束条件下将集装箱的体积进行最大化的利用。三维装箱问题的数学模型进行如下的建立：

1) 货物装载容器为集装箱，建立集装箱三维模型：长(L)、宽(W)、高(H)为 1190 cm、235 cm、268 cm，容积大约为 75 立方米。取集装箱左后角为坐标轴中心 0 点，建立三维直角坐标系，x 轴对应集装箱的长，y 轴对应集装箱的宽，z 轴对应集装箱的高，如图 1 所示。

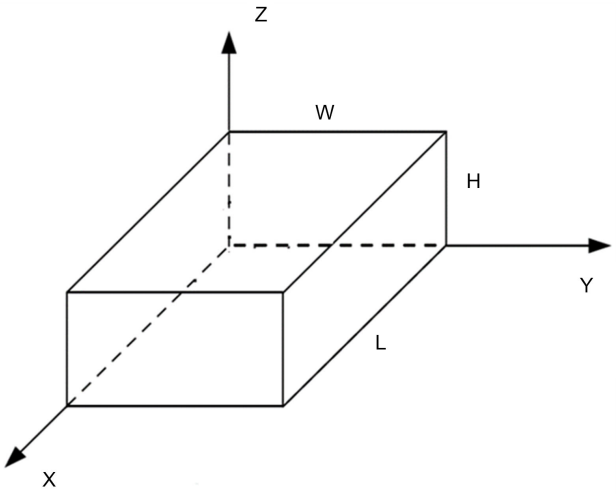

Figure 1. 3D model diagram of container
图 1. 集装箱三维模型图

2) 根据货物的具体属性建立货物三维模型，货物具有长、宽、高、重量、数量等属性。

3) 目标函数。目标函数为集装箱的装载率最优：

$$\text{total\_filling\_ratio} = \max \frac{\sum_{i=1}^{n} v_i}{V} \left(i = 1,2,3,\cdots,n\right) \tag{1}$$

其中， total_filling_ratio 为箱子的装载率，i 为货物号，v_i 为第 i 号货物的体积，V 为集装箱的体积。

4) 容积约束。装入集装箱的所有货物的体积和不能大于集装箱的体积。

$$\sum_{i=1}^{n} v_i \le V \tag{2}$$

5) 位置约束。在货物装载过程中，三个维度的坐标都需要满足以下约束：

$$\begin{cases} 0 \le x_i + l_i \le L \\ 0 \le y_i + w_i \le W \\ 0 \le z_i + h_i \le H \end{cases} \tag{3}$$

在对三维装箱问题的数学模型进行假设和建立后，进一步研究三维装箱问题的求解算法。

2.2. 基于生成可放置点的启发式算法

在装载过程中，使用四个点来表示一个货物的空间位置，如图 2 所示。其中，点 O 是货物的原点，该点存储在集装箱已放置货物的列表中，其余三个点分别是点 top(x, y, $z + h$)、点 front($x + l$, y , z)、点 right(x,

$y + w, z$)，这三个点用来判断货物之间是否发生重叠的情况。

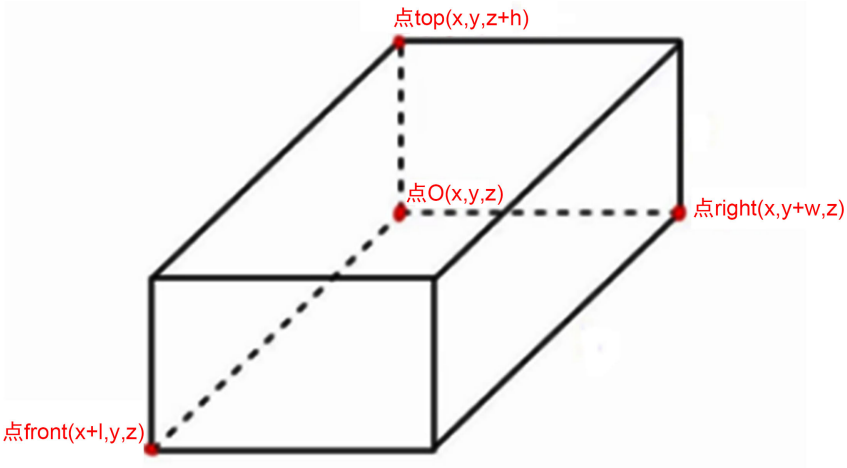

Figure 2. Schematic diagram of goods
图 2. 货物示意图

在货物装载过程中，以集装箱的长 L 作为 x 轴，以集装箱的宽 W 作为 y 轴，以集装箱的高 H 作为 z 轴，建立三维坐标系，每当装载一个货物，生成三个可放置点，即点(top, right, front)，在下一个货物进行装载时，可以进行方向的选择，然后完成装载。针对三个可放置点，可以设置三个列表，分别是 x 轴方向的可放置点列表：长度列表；y 轴方向的可放置点列表：宽度列表；z 轴方向的可放置点列表：高度列表。这三个可放置点列表用于下一个货物的装载。初始状态时，集装箱为空，此时只存在一个可放置点，即坐标原点(0, 0, 0)，装载第一个货物，然后将该货物的 front、right、top 三个点存储在三个可放置点列表中。装载完第一个货物之后，剩余的货物装载按照先沿 z 轴方向装载，后沿着 y 轴方向装载，再沿着 x 轴方向装载的顺序进行货物的装载：

1) 先从第一个货物的上方(即沿 z 轴方向)开始装载，形成一个竖排，直到该竖排的上方无法再装入下一个货物为止，z 轴方向装载如图 3 所示。

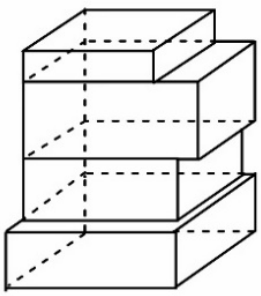

Figure 3. Schematic diagram of loading in z-axis
图 3. z 轴方向装载示意图

2) 然后从第一个箱子的右方(即沿 y 轴方向)开始装载，并且每一竖排按照 z 轴方向的装载方式构造出第二竖排、第三竖排，以此类推，直到该竖排的右方无法再装入下一个货物为止，此时构造出了一个类似 yOz 平面的货品墙，y 轴方向装载如图 4 所示。

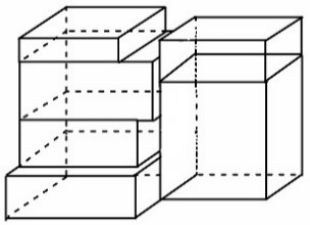

Figure 4. Schematic diagram of loading in *y*-axis
图 4. *y* 轴方向装载示意图

3) 在 *y* 轴方向的装载结束后，从图 4 得到的货品墙的前方(沿 *x* 轴方向)开始装载，即第一竖排货物的前方开始装载，装载方法和 *z* 轴与 *y* 轴方向装载相同，直至集装箱中无法再装入下一个货物为止，*x* 轴方向装载如图 5 所示。

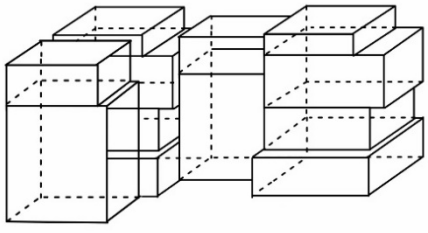

Figure 5. Schematic diagram of loading in *x*-axis
图 5. *x* 轴方向装载示意图

在货物装载过程中，要满足货物需要完全包含在集装箱中的约束、货物不能悬空的约束、货物之间不能重叠的约束等，所以需要对货物的合法性进行判断。

对于需要进行装载的货物，必须在集装箱内部，不能超出集装箱范围。假设该货物内部有一点(a, b, c)，要满足上述约束，则需要添加以下判断的条件：

$$\begin{cases} a \geq x \\ b \geq y \\ c \geq z \end{cases} \text{且} \begin{cases} a \leq x + l \\ b \leq y + w \\ c \leq z + h \end{cases} \tag{4}$$

对于之后要装载的货物，它所对应的 top、right、front 点将有可能和其他箱子发生重叠的情况，所以在进行下一步装载时，需要判断货物的这三点是否合法。以 top 点为例来判断是否重叠，如图 6 所示。

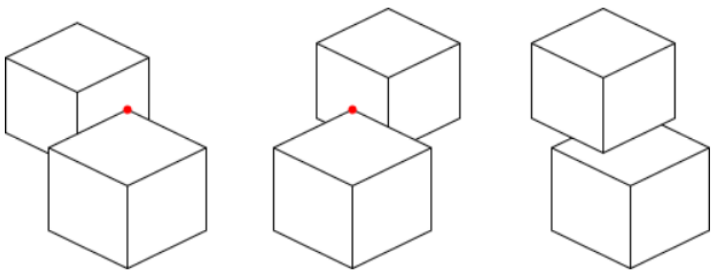

Figure 6. Legal overlap of “top” points
图 6. top 点合法重叠情况

如图 6 所示，top 点能够和箱子上的以上三个面发生重合，但是此时属于紧邻的情况，是合法的，因此在 $\begin{cases} a \geq x \\ b \geq y \\ c > z \end{cases}$ 时，top 点的重叠判断条件为 $\begin{cases} a < x+l \\ b < y+w \\ c \leq z+h \end{cases}$。在 $\begin{cases} a \geq x \\ b > y \\ c \geq z \end{cases}$ 时，right 点的重叠判断条件为 $\begin{cases} a < x+l \\ b \leq y+w \\ c < z+h \end{cases}$。在 $\begin{cases} a > x \\ b \geq y \\ c \geq z \end{cases}$ 时，front 点的重叠判断条件为 $\begin{cases} a \leq x+l \\ b < y+w \\ c < z+h \end{cases}$。

3. 遗传算法

遗传算法本质上模拟了生物进化中的自然选择和变异进化过程，生成一个计算模型，该模型在求解复杂问题时，能够较快地获得优化结果。

3.1. 遗传算法编码

设计遗传算法编码(即基因序列)由三部分组成，分别是货物的装载顺序、货物的旋转状态、该条基因序列对应的装载率。

1) 货物的装载顺序

用数字对所有的货物进行 1……n 的编号，并生成初始的货物装载的顺序。

2) 货物的旋转状态

在录入货物信息时，会对货物的放置状态如图 7 所示。

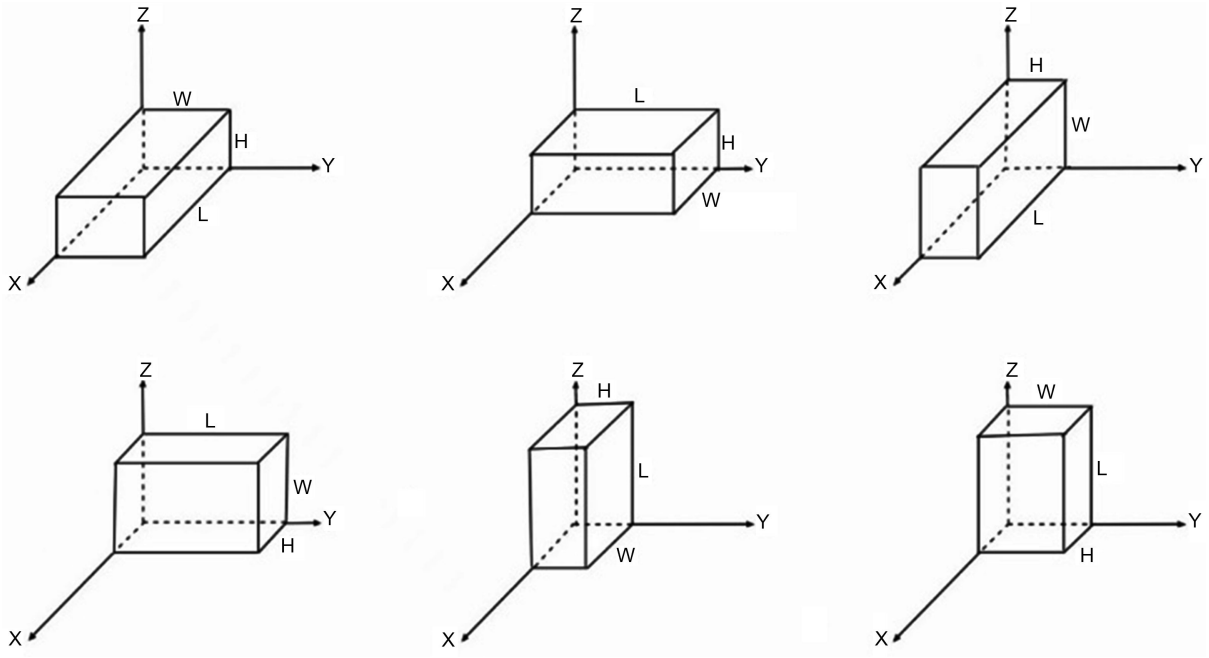

Figure 7. Schematic diagram of six rotational placement methods
图 7. 六种旋转放置方式示意图

根据不同的放置方式，对应编码为 0、1、2、3、4、5，如表 1 所示。

3) 该条基因序列对应的装载率

用编码序列的第 $2n+1$ 个数表示当前基因序列(当前装载方案)下的装载率，便于后期进行统计和比较。

本文中的编码表示如下所示：

$$X=\left\{x_1,x_2,x_3,\cdots,x_n,x_{n+1},\cdots,x_{2n},x_{2n+1}\right\} \tag{5}$$

Table 1. Encoding of six placement methods
表 1. 六种放置方式的编码

放置方式	编码方式	编码	X 轴	Y 轴	Z 轴
正放 (H 为高)	方式 0	RT_LWH = 0	L	W	H
	方式 1	RT_WLH = 1	W	L	H
侧放 (W 为高)	方式 2	RT_LHW = 2	L	H	W
	方式 3	RT_HLW = 3	H	L	W
侧放 (L 为高)	方式 4	RT_WHL = 4	W	H	L
	方式 5	RT_HWL = 5	H	W	L

3.2. 遗传算法的选择、交叉和变异

可以通过轮盘赌模型选择适应度较高的基因序列。

1) 选择算子

① 计算出群体中每条基因序列的适应度：

$$f\left(x_i\right)=\frac{\sum_{k=1}^{n}v_{ik}}{V}\left(k=1,2,3,\cdots,n;i=1,2,3,\cdots,\text{sizepop}\right) \tag{6}$$

其中，$f\left(x_i\right)$是每条基因序列的适应度，即该条基因序列对应的装载率，sizepop 为种群数量。

② 计算每个个体(即每条基因序列)被遗传到下一代群体中的概率，用 $P\left(x_i\right)$ 表示：

$$P\left(x_i\right)=\frac{f\left(x_i\right)}{\sum_{j=1}^{n}f\left(x_j\right)} \tag{7}$$

③ 计算每个个体(即每条基因序列)的累积概率：

$$q_i=\sum_{j=1}^{i}p\left(x_j\right) \tag{8}$$

④ 在[0, 1]区间内产生伪随机数 r。

⑤ 若 $r<q[1]$，则选择个体 1，否则选择个体 k，使得 $q[k-1]<r\le q[k]$成立。

⑥ 重复④、⑤操作共 sizepop 次，其中 sizepop 为种群大小。

通过多次的轮盘赌算法，对种群进行选择，选出所需要的染色体基因序列。

2) 交叉算子

对编码后的基因序列进行交叉。在交叉操作中，采取均匀交叉算子和部分映射交叉相结合的方式，所谓均匀交叉就是以同样的交叉概率对两个配对个体的基因都进行交换，这样两个新个体同时产生。均匀交叉示意图如图 8 所示。

部分映射交叉是两条亲代染色体在交叉位点之间的部分基因进行交换，交叉点是随机选取的。然后固定交换的部分不变，将染色体中未交换的部分进行一一分析，假如出现了与互换部分重叠的基因时，就把重叠的基因全部改为不互换的基因，这样进行下去，直至在各段染色体组型中，均不含有重叠的基因。部分交叉映射的例子如图 9 所示。

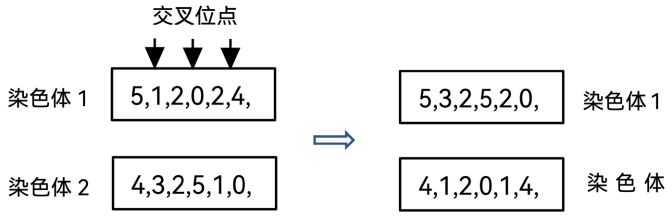

Figure 8. Schematic diagram of uniform crossing
图 8. 均匀交叉示意图图

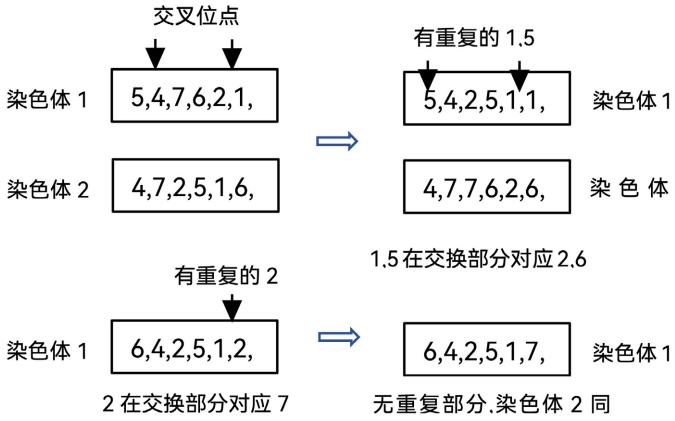

Figure 9. Schematic diagram of partial mapping cross
图 9. 部分映射交叉示意图

由上述分析可知，编码长度为 $2n+1$，对于 $[1,n]$ 之间的基因，存储的是当前货物的装载顺序，通过部分映射交叉进行，$[n+1,2n]$ 之间的基因，存储的是当前货物的旋转放置方式，通过均匀交叉进行。

3) 变异算子

为了防止求解过程中陷入局部最优解，需要对个体进行变异，其中变异概率为 Pm。对于 $[1,n]$ 之间的基因，采用顺序逆转变异的方式，即将该段基因的序列逆序赋值，对于 $[n+1,2n]$ 之间的序列，在旋转放置方式可允许的范围内进行变异操作。

4. 启发式算法和遗传算法结合混合算法

首先根据基于生成可放置点的启发式算法对需要装载的货物进行装载，得到每个基因序列对应的适应度值(即装载率)，然后进行遗传算法的操作：对种群进行选择，进行适应度评价，通过交叉、变异操作生成新的子代，继续迭代，直到达到设定的次数。当生成的种群代数大于设定的迭代次数时，遗传算法运行结束，输出最优解。

基于生成可放置点的启发式算法和遗传算法的混合算法求解装载率最大的三维装箱问题的流程图如图 10 所示。

经过启发式算法和遗传算法的混合算法进行迭代后，最终可以在有限时间内找到一个最优装载方案。

5. 测试与总结

5.1. 测试数据及结果

测试数据中货物种类分为三类：纯纸箱/纯木箱类、混装类和纯托类货物。这里以实际集装箱货物数

据为基础对混合算法进行测试，测试数据的典型实例如表 2~4 所示。不同类型的货物要求的装载率会有所不同，一般情况下装载纯木箱/纯纸箱类货物装载率要求达到 90%以上；装载托散混装类货物装载率要求达到 88%以上；装载纯托货物类装载率要求达到 73%以上。

通过 8 组货物实例数据进行测试运行，得到的结果如表 5 所示。

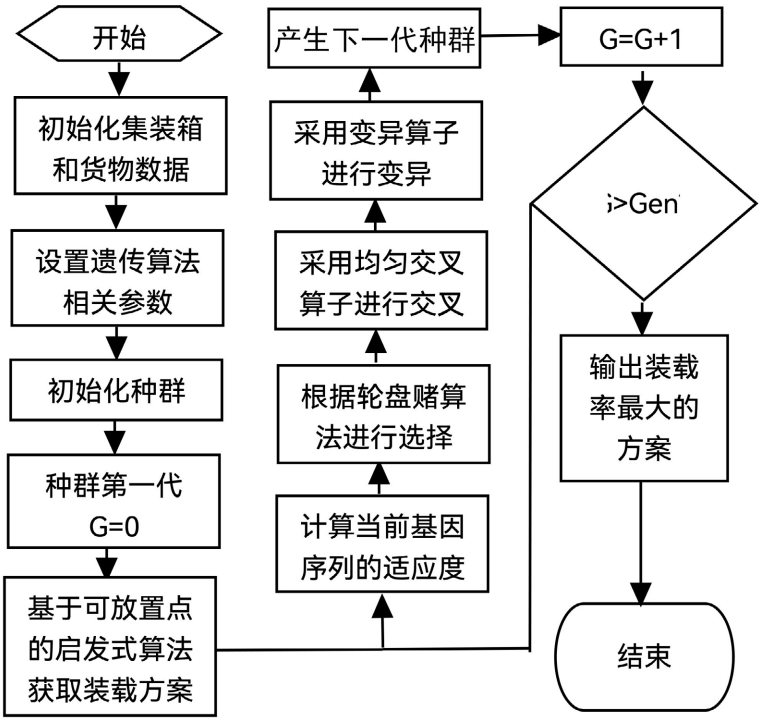

Figure 10. Flowchart of hybrid algorithm
图 10. 混合算法流程图

Table 2. Dataset interception for pure cardboard or pure wooden box goods
表 2. 纯纸箱或纯木箱类货物数据集截取

序号	品名	装箱位置	码放要求	包装类别	件数	单件毛重(KG)	长(cm)	宽(cm)	高(cm)
1	物品 1	2-无要求	正放&侧放&立放	1-纸箱	17	16.00	32.00	46.00	33.00
2	物品 2	2-无要求	正放&侧放&立放	1-纸箱	14	16.00	51.00	47.00	35.00
3	物品 3	2-无要求	正放&侧放&立放	1-纸箱	40	19.00	55.00	41.00	45.00
4	物品 4	2-无要求	正放&侧放&立放	1-纸箱	10	17.00	56.00	40.00	40.00
5	物品 5	2-无要求	正放&侧放&立放	1-纸箱	30	23.00	64.00	40.00	23.50
6	物品 6	2-无要求	正放&侧放&立放	1-纸箱	10	16.00	47.50	38.00	27.50
7	物品 7	2-无要求	正放&侧放&立放	1-纸箱	25	21.50	46.00	27.00	35.00
8	物品 8	2-无要求	正放&侧放&立放	1-纸箱	150	18.50	43.00	37.00	11.00
……	……	……	……	……	……	……	……	……	……
49	物品 1	2-无要求	正放&侧放&立放	1-纸箱	100.00	12.80	43.00	37.00	11.00

Table 3. Dataset interception for hybrid goods
表 3. 混装类货物数据集截取

序号	品名	装箱位置	码放要求	包装类别	件数	单件毛重(KG)	长(cm)	宽(cm)	高(cm)
1	物品 1	2-无要求	正放	木箱	1	808.33	170	120	147
2	物品 2	2-无要求	正放	木箱	1	808.33	203	148	136
3	物品 3	2-无要求	正放	托盘	1	460.00	100	85	114
4	物品 4	2-无要求	正放&侧放&立放	纸箱	41	30.59	36	26	16
5	物品 5	2-无要求	正放&侧放&立放	纸箱	20	30.59	85	50	20
6	物品 6	2-无要求	正放	木箱	1	377.50	108	88	71
7	物品 7	2-无要求	正放&侧放&立放	纸箱	736	16.03	49	32	23
8	物品 8	2-无要求	正放&侧放&立放	纸箱	31	13.59	46	35	17
……	……	……	……	……	……	……	……	……	……
21	物品 21	2-无要求	正放&侧放&立放	纸箱	50	13.59	40	30	21

Table 4. Dataset interception for pure pallet goods
表 4. 纯托类货物数据集截取

序号	品名	装箱位置	码放要求	包装类别	件数	单件毛重(KG)	长(cm)	宽(cm)	高(cm)
1	物品 1	2-无要求	正放	托盘	16	32.00	52.00	29.00	28.00
2	物品 2	2-无要求	正放	托盘	1	445.00	86.00	84.00	107.00
3	物品 3	2-无要求	正放	托盘	1	445.00	255.00	104.00	184.00
4	物品 4	2-无要求	正放	托盘	1	126.00	133.00	69.00	164.00
5	物品 5	2-无要求	正放	托盘	98	13.00	40.00	28.00	42.00
6	物品 6	2-无要求	正放	托盘	61	13.00	53.00	33.00	48.00
7	物品 7	2-无要求	正放	托盘	20	13.00	53.00	37.00	48.00
8	物品 8	2-无要求	正放	托盘	213	13.00	49.00	39.00	50.00
……	……	……	…….	……	……	……	……	……	……
15	物品 15	2-无要求	正放	托盘	1	705.00	188.00	103.00	122.00

Table 5. Data test result of the examples of goods
表 5. 货物实例数据测试结果

实例数据	货物数量	物品种类	所属类别	目标装载率	传统启发式算法装载率	混合算法装载率
货物实例 1	1648	1	1-纯纸箱、木箱类	90%	91%	91%
货物实例 2	1502	49	1-纯纸箱、木箱类	90%	85%	92%
货物实例 3	3588	1	1-纯纸箱、木箱类	90%	91%	93%
货物实例 4	1175	21	2-托散混装类	88%	83%	89%
货物实例 5	650	26	2-托散混装类	88%	85%	90%
货物实例 6	658	32	2-托散混装类	88%	78%	89%
货物实例 7	133	28	3-纯托类	73%	57%	78%
货物实例 8	756	15	3-纯托类	73%	84%	82%

通过上述结果分析可以看到，大多数情况下混合算法在装载率上是优于其他方法的，尤其是在货物规格种类较多的真实装箱情况下，混合算法的优势会更加明显。

5.2. 测试总结

本文中研究的混合装箱算法中，用基于生成可放置点的启发式算法来减少装载过程中集装箱的空间浪费，同时，在启发式算法的基础上用遗传算法来寻找最优值，二者相互补充，让装载方案的整个过程更加简洁、算法运行效率更高，使得装载率能够得到更进一步的提升。

基金项目

国家自然基金(62076028)支持；促进高校分类发展——“青年骨干教师”支持计划支持。

参考文献

[1] Araya, I., Guerrero, K. and Nuñez, E. (2017) VCS: A New Heuristic Function for Selecting Boxes in the Single Container Loading Problem. *Computers & Operations Research*, **82**, 27-35. https://doi.org/10.1016/j.cor.2017.01.002

[2] 张钧, 贺可太. 求解三维装箱问题的混合遗传模拟退火算法[J]. 计算机工程与应用, 2019, 55(14): 32-39+47.

[3] 林云鹏, 宋爽, 江志斌, 张大力. 电商物流背景下基于空间矩阵的三维装箱算法[J]. 工业工程, 2022, 25(5): 128-136+152.

[4] 于明正, 徐斌, 陈佳. 基于双层启发式遗传算法的三维装箱问题[J]. 科学技术与工程, 2020, 20(5): 2042-2047.

[5] Fan, Y., Chu, J. and Xu, H. (2020) Improvement Grouping Genetic Algorithm for Solving the Bin Packing Problem. *Journal of Physics*: *Conference Series*, **1550**, Article ID: 032168. https://doi.org/10.1088/1742-6596/1550/3/032168

[6] Cant, R., Remi-Omosowon, A., Langensiepen, C. and Lotfi, A. (2018) An Entropy-Guided Monte Carlo Tree Search Approach for Generating Optimal Container Loading Layouts. *Entropy*, **20**, Article 866. https://doi.org/10.3390/e20110866

[7] 张德富, 彭煜, 张丽丽. 求解三维装箱问题的多层启发式搜索算法[J]. 计算机学报, 2012, 35(12): 2553-2561.

[8] 李孙寸, 施心陵, 张松海, 等. 基于多元优化算法的三维装箱问题的研究[J]. 自动化学报, 2018, 44(1): 106-115.

[9] Xiang, X., Yu, C., Xu, H. and Zhu, S.X. (2018) Optimization of Heterogeneous Container Loading Problem with Adaptive Genetic Algorithm. *Complexity*, **2018**, 1-12. https://doi.org/10.1155/2018/2024184

[10] Kilincci, O. and Medinoglu, E. (2021) An Efficient Method for the Three-Dimensional Container Loading Problem by Forming Box Sizes. *Engineering Optimization*, **54**, 1073-1088. https://doi.org/10.1080/0305215x.2021.1913734