**需求设计说明与模型建立**

**问题描述**

**背景**

货运集装箱问题，当前久章支持适用于快递包装的**小规模装载可行性问题**的精确求解，对于大规模**可行性问题以及 3D-BPP 问题无法求解，需要设计启发式算法**；

3D-BPP，考虑**几种**集装箱，每种集装箱**数量；**每种集装箱都有其**使用成本；给定一批货物，**其中所有货物**均可以平转**（交换LW），除**某个种类**的货物外，其他货物均可堆叠，找出使得**总成本最小**的货物装载方案；特别的，用户要求回答以下问题：

1. 一批货物，**是否**能够以及如何装**一个**集装箱；
2. 一批货物，**是否**能够以及如何装**多个不同类**集装箱，并且总费用最低；（或装载率最高）

**问题假设**

离线装载，即装载时货物种类及数量均已知，且有限；

货物可以平转，即可交换长宽边，但不能翻倒，即不能交换长边与高边、宽边与高边；

货物摆放与集装箱边始终保持平行，不存在斜放；

**符号定义**

集合类符号

|  |  |  |
| --- | --- | --- |
| 符号 | 说明 | 范围 |
|  | 所有货物集合 |  |
|  | 所有集装箱集合 |  |

参数类符号

|  |  |  |
| --- | --- | --- |
| 符号 | 说明 | 单位 |
|  | 货物种类整型参数 | - |
|  | 各货物的初始长宽高 |  |
|  | 各货物的体积 |  |
|  | 表示货物只能放置于上层 | - |
|  | 集装箱种类整型参数 | - |
|  | 各集装箱的长宽高 |  |
|  | 各集装箱的容积 |  |

决策变量

|  |  |  |
| --- | --- | --- |
| 符号 | 说明 | 范围 |
|  | 表示第i个货物装入第j个集装箱内 |  |
|  | 第j个集装箱是否被使用 |  |
|  | 货物i在集装箱内的左后下角坐标 |  |
|  | 货物i在集装箱内的右前上角坐标 |  |
|  | 表示第i个货物需要平转 |  |
|  | 货物i严格位于货物k的右侧，即时取1 |  |
|  | 货物i严格位于货物k的前侧，即时取1 |  |
|  | 货物i严格位于货物k的上侧，即时取1 |  |
|  | 与货物i直接接触的、货物i下方货物集合 |  |
|  | 与货物i直接接触的、货物i上方货物集合 |  |
|  | 货物i与货物k均在集装箱j内，即  后续对其进行线性化 |  |
|  | 表示货物i在集装箱内堆叠，即 |  |

**目标函数**

最大化集装箱装载率。当货物量固定时，即最小化使用集装箱成本（以体积代替）

|  |  |
| --- | --- |
|  | （1） |

**约束条件**

|  |  |  |
| --- | --- | --- |
|  |  | （2） |
|  |  | （3） |
|  |  | （4） |
|  |  | （5） |
|  |  | （6） |
|  |  | （7） |
|  |  | （8） |
|  |  | （9） |
|  |  | （10） |
|  |  | （11） |
|  |  | （12） |
|  |  | （13） |
|  |  | （14） |
|  |  | （15） |
|  |  | （16） |
|  |  | （17） |
|  |  | （18） |
|  |  | （19） |
|  |  | （20） |
|  |  | （21） |
|  |  | （22） |
|  |  | （23） |
|  |  | （24） |
|  |  | （25） |
|  |  | （26） |

式2为总体积约束，表示集装箱装入货物不超过其容积。

式3表示全部货物均需装载。

式4-6表示货物的装入长宽高位置均要位于箱子以内。

式7-8表示允许货物发生平转。

式9-11表示放入同一个集装箱的货物才具备前后、左右和上下的位置关系。

式12-14表示同一集装箱内货物的严格右、前、上关系。

式15表示同一集装箱内货物不可重叠，即不存在2个以上的坐标投影面相交。

式16表示堆积货物的底下面积需要大于货物的底面积，才可进行堆叠。

式17是堆积货物的变量表达。

式18表示堆积货物的底下货物集合。

式19表示如果货物是顶层货物，那么其严格上方的货物均不能与其有底面投影相交。

式20是两个货物同处一个集装箱的变量表达。

式21-26是变量的定义域约束。

式20的线性化

|  |  |  |
| --- | --- | --- |
|  |  | （27） |
|  |  | （28） |
|  |  | （29） |

仍存在面积计算、集合计算、最值表达等约束，即约束非线性，难以直接调用线性求解器求解。

**输入**

* **给定数量的**集装箱 ，其内径三维 ，使用成本 ；其中使用成本可先用集装箱体积代替；
* 需要装载的货物 及其尺寸 ；其中每个货物属于 10 种预先给定的**货物种类**之一；
* 不可堆叠的货物集合 ；某个货物不可堆叠**等价于**这些货物属于 10 种货物中特定的几种；

**决策**

* 货物被分配的集装箱 ;
* 每个货物的装载位置，由其装载左下角坐标表示 ；
* 货物是否旋转，只支持平转

**约束**

* 某些种类的货物不可堆叠；这些货物只能放在最上层；
* 货物不可交错堆叠，上层货物的底部必须包含在下层货物的顶部之内；
* 货物必须全部装载；
* 其它合法性约束，货物必须装在容器内部，货物的空间立方体不可有重叠；

**目标**

* 使用的集装箱总成本最小；

**说明**

* 集装箱属于 3 种预先给定的种类之一，每种的使用成本，长宽高一致；
* 货物属于 10 种预先给定的种类之一，实际业务场景中，这 10 个种类不太可能发生变化，可以利用；
* 不可堆叠的货物种类固定，意味着这些货物尺寸一致；
* Excel 文档里给出了三个集装箱，以及三组问题的示例，可以通过**排列组合**，并**补充集装箱的数量**等条件做测试：
* 简要思路：
* 先对集装箱，按照其种类进行分组 （3 组）；然后对每组集装箱，求解一个集装箱和（剩余）所有货物组成的子问题；
* 子问题为找出使得浪费空间最小的 3 维装载问题：
* 首先尝试将所有货物生成堆叠；
* 将 货物作为初始堆叠**栈**；
* 按照约束条件，将长宽不小于初始堆叠底部的货物入栈，按照以下原则生成堆叠

1. 长宽最接近的优先；
2. 厚货物优先；
3. **薄**货物最后插空的；

* 堆叠生成后，使用启发式搜索（或建模MIP），求解对应的二维装箱问题
* 对 3 组集装箱生成的 3 个子问题，找到使得 （浪费空间/总空间 \* 使用成本）最小的方案（可以思考下更好的目标函数，从用户角度较合理），执行该方案；
* 对剩下的集装箱和货物重复执行上述步骤；

**需求**

* 调研场景类似的 3D-BPP 装箱问题，输出简要综述报告，PPT或文档 （3人天）；
* 3D-BPP 问题启发式算法编程（7人天）；
* 算法测试用例报告（3人天）：
* 求解 excel 文档给出的三组问题，其中每个假设每个集装箱只有一个，成本用体积近似；
* 上述问题三组问题的货物混合后的问题，假设每个集装箱给两个；
* 装箱问题用户接口设计，flask 调用开发测试 （7 人天）；

最小化使用集装箱成本（以体积代替）

树搜索

一个节点：当前使用箱型的分支

节点输入：

父节点的所有输出

节点输出：

累计至当前消耗箱型总成本的上界（LB）

当发现可行解，即有上界（UB），若上界更低，更新（降低）全局上界

已放置货物的layout方案

已放置货物的列表、字典

未放置货物的列表、字典

已使用的集装箱列表、字典

未使用的集装箱列表、字典

节点fathomed：

最优解：当前节点已放入所有货物

无解：当前节点处已用完所有集装箱，但仍存在货物

不理想：当前节点解LB高于全局UB

分枝策略：

当节点不fathomed，则选择需要进行分支

依据节点未使用的集装箱种类的数量，决定分支数量（如3类未使用数量>0，就分3支）

树搜索策略：DFS？

剪枝策略：

每求解完一个节点后，对所有活着的节点进行的操作。

令不理想的节点fathomed掉（叉掉）