报告提纲

背景介绍

主要研究问题：两类问题：单箱型装箱可行性问题，多箱型组合装箱方案问题

特殊约束（平转，顶部，完全支撑）

目标函数：目前为整体满载率，即使用集装箱的总体积最少，此处存疑？（使用1个体积100的集装箱VS使用3个体积30的集装箱）

已有算例解读

集装箱和货物的体型特点（预先存在不可装入货物）

堆叠算法可行性与准确性

在少影响底面利用率的情况下，充分利用箱体高度进行堆叠生成。（高度求余）

算法思想：

* 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（生成堆叠时，不涉及货物的平转决策）

1. 长宽最接近的优先；
2. 厚货物优先（最多只有2个异类的能堆叠）；

* 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，允许堆叠整体平转，找到使得**满载率最大**（已装入空间/总空间）最小的方案，即，使得装入堆叠的总体积尽可能大的二维装箱问题（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；
* 集装箱列表与货物列表同时更新，再对剩下的集装箱和货物重复执行上述步骤；

代码实现展示

主程序输入输出逻辑

实体类定义

算法展示