

BoxFiller 软件开发报告

1 背景

1.1 装箱问题

装箱问题可以描述为在空间约束下对元素进行布局并达到某种效率的问题，其中空间约束是指不重叠、不覆盖、符合物理稳定性的约束（顾名思义，如同在一个容器里装箱子）。空间可以是一维、二维或是三维的，其中由于一维装箱问题约束最为简单，研究已经较为成熟。二维装箱是在一维装箱上进行的拓展，对一维装箱的研究可以经过转化运用到二维装箱问题上从而起到“降维”的效果。三维装箱最为复杂而又最为实际常见，但是目前研究相对较少。

对装箱问题进行分类，最常见的分类指标是空间的维度以及物体的规则性，物体的规则性是指物体所具有的几何形状、物理量分布等是否均匀协调，研究中一般以均匀矩形体最为常见。

1.2 装箱算法

对装箱问题进行求解的算法称为装箱算法。无论是一维、二维还是三维装箱问题都已经被确认为 **NP-Hard**，常采用近似算法、启发式算法以及智能优化算法等进行求解。

目前一维装箱算法最为成熟，经典的近似算法包括其次适配（Next Fit, NF）算法、首次适配（First Fit, FF）算法、最佳适配（Best Fit, BF）算法以及最坏适配（Worst Fit, WF）算法。

二维装箱较为复杂，其算法以启发式为主，并且常常通过某种“降维”的方式转化为一维装箱，利用其算法求解。

三维装箱问题的困难性使得其求解算法研究较少，一般采用启发式与智能优化的混合算法，并结合“降维操作”转化为二维装箱求解。但是一般而言随着箱子数量的增加，其搜索时间也将大大增加，求解效率并不高。

2 内容

本文的研究对象为规则矩形体的三维装箱，并根据文献【2】所提出的算法进行软件开发，实现报告文档生成、数据统计以及三维可视化。

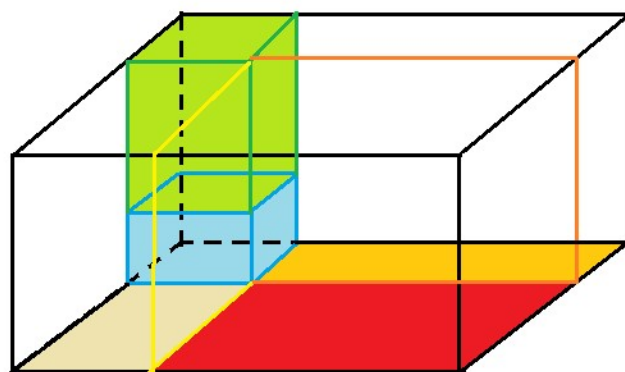
随着三维装箱问题的不断深入研究，其解的空间填充率已经十分令人满意。然而，在目前的研究和应用中存在以下问题：一方面当箱子的种类和数量特别巨大时，传统算法难于甚至于不能够在合适的时间内进行求解，其计算复杂度较高；另一方面，在以三维装箱为模型的实际应用中，经常出现“箱子”的删除、增加或者手动调整等情况，需要相关算法能够迅速相应。所以无论是学术研究还是工业实际，这种基于直接求解的快速算法都十分必要。

为实现三维装箱问题的高效求解，在综合考虑剩余空间分割和箱子布置的前提下，文献【2】借鉴二维 PH 算法的直接求解方式，提出了启发式的剩余空间最优化算法（Three-Dimensional Residual-Space-Optimized Algorithm，简称 3D-RSO）。该算法在求解过程中不需要额外的预处理和最优解搜索操作，能够在极少的计算消耗下得到一个较优的解，相对于现阶段其他算法在求解效率上具有明显优势。

3 算法详解

3.1 剩余空间

3D-RSO 算法的核心在于分解装箱后的剩余空间。一般而言，在一个矩形体三维空间内装入一个矩形体箱子之后的剩余空间可分为 4 块，如下图所示



上图中黑色边框代表容器，蓝色代表装入的箱子，剩余空间分为 4 块，箱体上方的绿色区域、底面积为淡色、黄橙色、红色的三块矩形空间区域。为了充分利用容器空间，考虑将红色区域并入黄橙色或淡色区域，并且选择其中产生空间最大的进行合并。该原则称为“剩余空间最大原则”。

3.2 装载方式

在容器中进行装箱时，除了基本的三维空间不可重叠的约束之外，3D-RSO 算法还考虑以下三个约束：

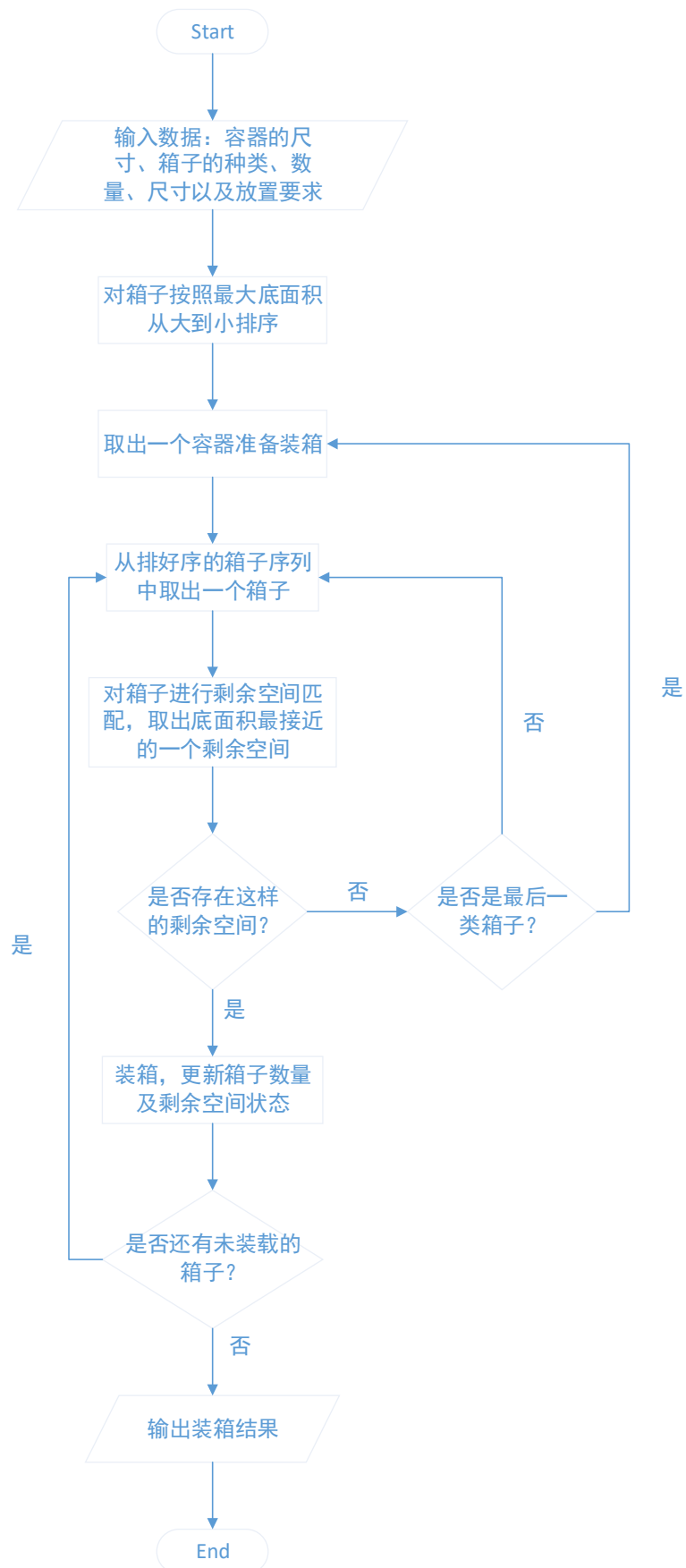
C1：方向性约束，箱子的装载具有方向性约束，也就是说，每个箱子只能按照其给定的放置姿态进行放置（考虑是否允许倒置；箱子边缘需与容器边缘平行）。

C2：稳定性约束，每个被装载箱子必须得到容器底部或者其他已装载箱子的支撑。本文主要考虑箱子的完全支撑约束，即被装载箱子的底部不允许有任何悬空。

C3：完全切割约束，此约束是二维装箱中一刀切约束的扩展。它要求在最终的装填状态中，已放入箱子的集合可以由多个竖直的平面分割成多个子空间，并且每个子空间又可以被多个水平的平面分割成更小的子空间。以此方式，每个箱子都可以通过竖直或者水平的平面被完整的分割出来。

虽然存在 C1 约束，但由于可以倒置或旋转箱子的方位，箱子的装载方式存在多种选择，该选择由“剩余空间最大化”原则确定。C2 和 C3 约束具有一定的实际意义，例如在集装箱的装卸过程中，当集装箱的放置状态同时满足以上两个约束时，叉车就可以从集装箱的底部直接叉取货物，大大减少其装卸的难度和时间消耗。

4 算法框图



5 算法效能分析

3D-RSO 算法的复杂度为 $O(n^2)$ 。文献【2】通过实际算例的演示，将 3D-RSO 算法与其他算法相比较，发现 3D-RSO 算法虽然牺牲了些许存储效率，但是将求解时间大大缩短，尤其是强异构体装箱，存储效率也接近其他优化算法，这说明 3D-RSO 算法非常适用于强异构体快速装箱问题。

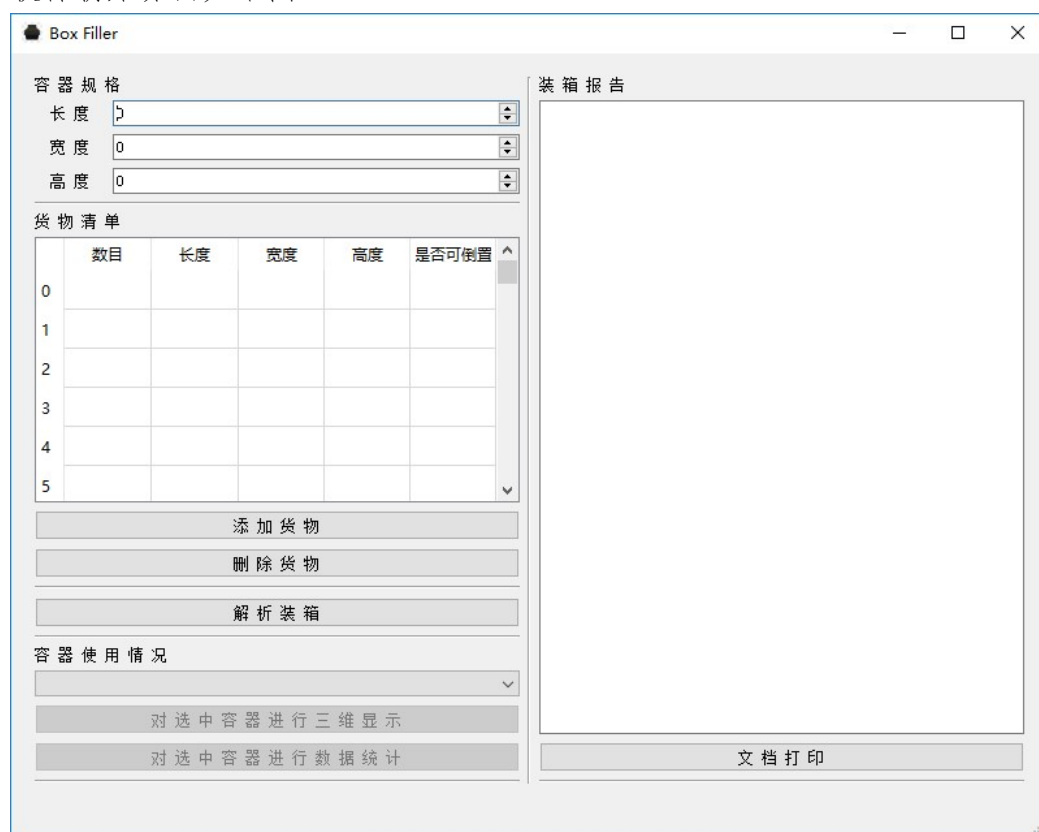
6 软件开发

本软件使用 PyQt5 进行开发。

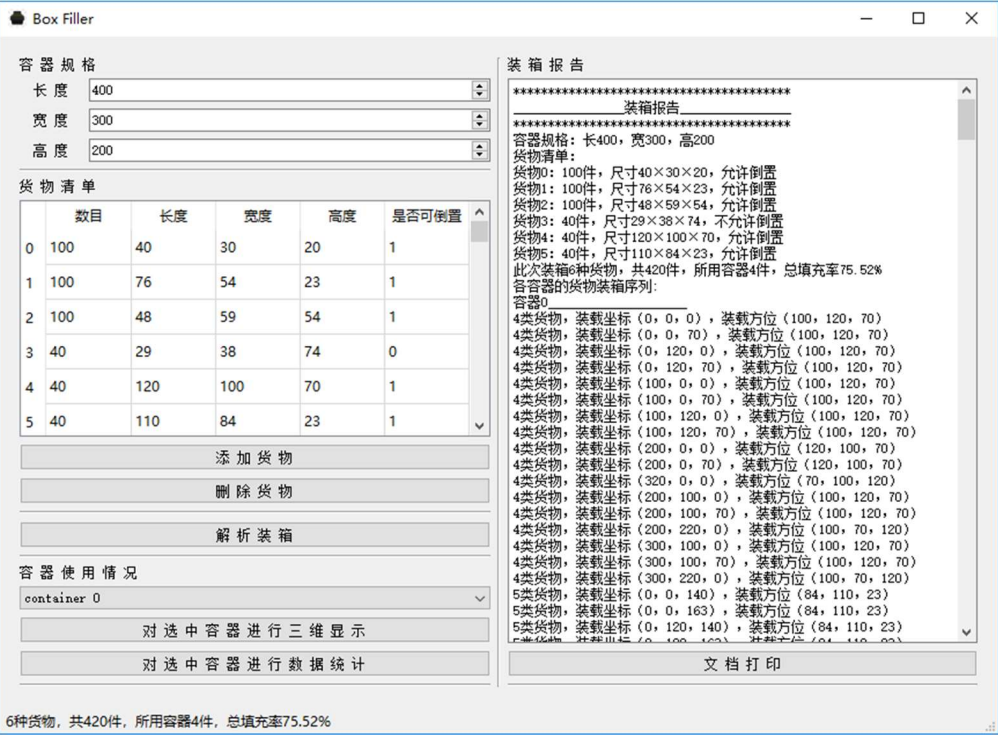
PyQt5 是一套来自 Digia 的 Qt5 应用框架和 Python 的粘合剂，由 Riverbank Computing 开发。支持 Python2.x 和 Python3.x 版本。本软件使用 Python3。Qt 库是最强大的 GUI 支持库的一种，采用 C++ 语言。

PyQt5 以一套 Python 模块的形式来实现功能，包含了超过 620 个类，600 个方法和函数，是一个多平台的工具套件，可以运行在所有的主流操作系统中，包含 Unix，Windows 和 Mac OS。

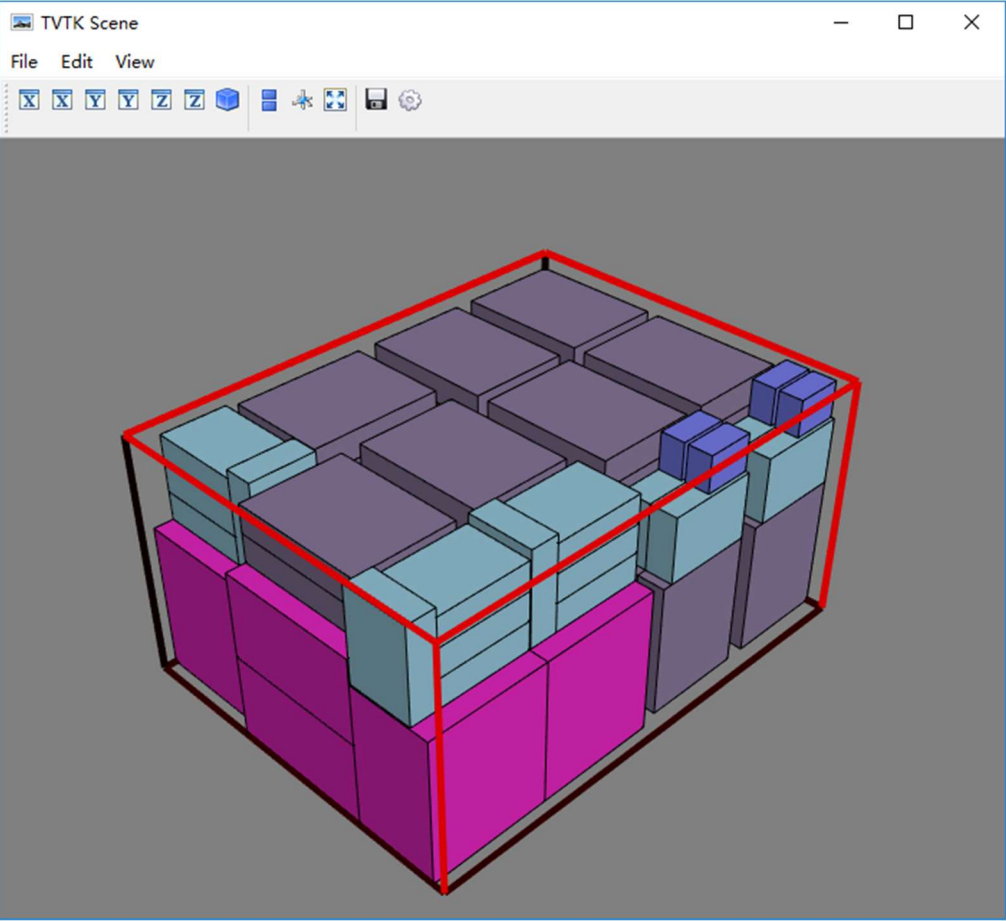
软件初始界面如下图



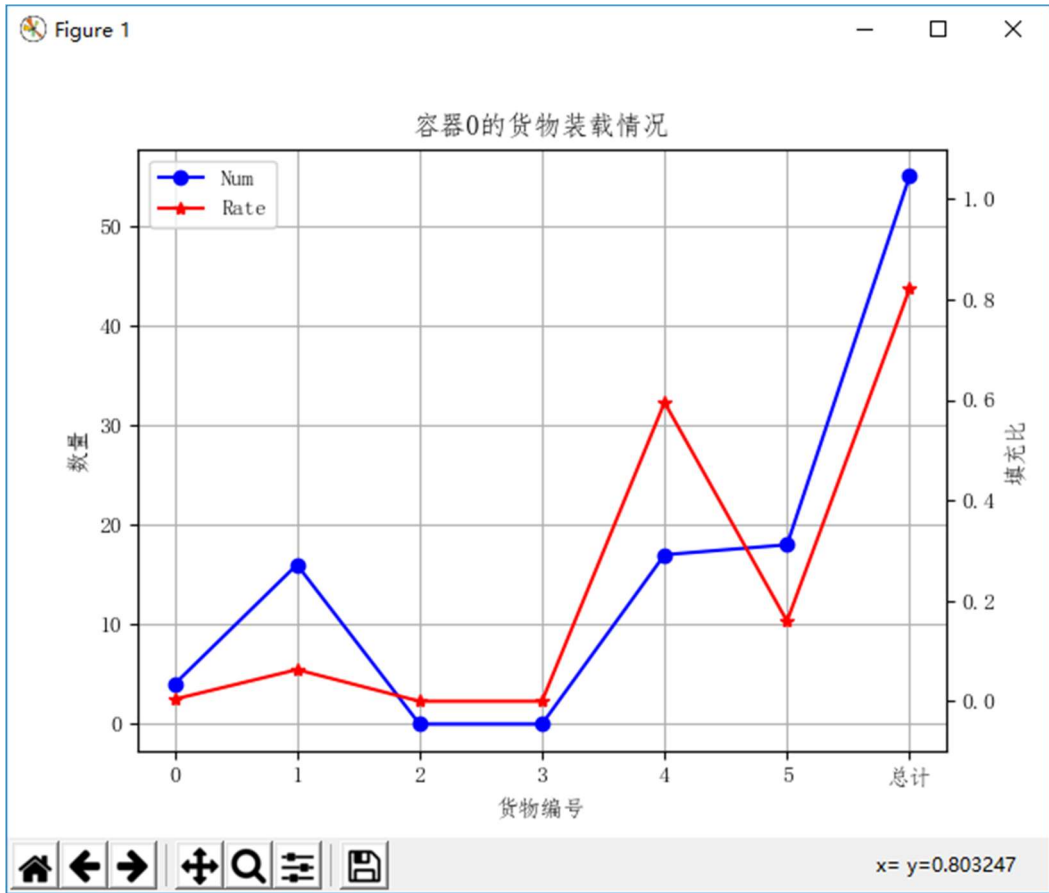
设置好容器规格及货物参数之后点击解析装箱将在右侧报告区生成装箱报告，并在左下角出现装箱的整体情况信息，三维显示及数据统计按钮激活，如下



点击三维显示按钮将出现三维可视化图形，如下



点击数据统计按钮将会出现数据统计图，如下



7 软件评述

目前软件在数据录入、报告生成、数据统计及图形可视化上基本可以满足一般的需要，但是由于算法本身的特点，装箱效率有待提升，下一步将结合多种情境，载入多种优化算法，择优选用，扩大软件适用范围。

参考文献

- 【1】 装箱问题的算法及最新进展，刘明明
- 【2】 高效求解三维装箱问题的剩余空间最优化算法，尚正阳