装箱问题（Bin packing problem）

问题的定义

装箱问题（Bin packing problem），又称集装优化，是一个利用运筹学去解决实际生活的的经典问题。在维基百科的定义如下：

In the bin packing problem, items of different volumes must be packed into a finite number of bins or containers each of a fixed given volume in a way that minimizes the number of bins used. In computational complexity theory, it is a combinatorial NP-hard problem. The decision problem (deciding if items will fit into a specified number of bins) is NP-complete.

中文大概意思就是：装箱问题中，不同体积的物品必须被打包到有限数量的箱子（bins）或容器中，每个箱子的体积都是固定的，并且要使所使用的箱子数量最小化。在计算复杂度理论中，它是一个 NP-难的问题。

从20世纪70年代初开始,装箱问题就引起了广泛的探讨和研究。然而装箱问题可以追溯到1831年高斯(Gauss)开始研究布局问题，因为装箱问题和布局问题本质上是一样的,到现在已有百余年的历史。虽然经过几代人的努力，但迄今尚无成熟的理论和有效的数值计算方法。（摘自百度百科）

装箱问题是NP难解问题，这意味着该问题不存在在多项式时间内求得精确解的算法。因此对装箱问题算法的研究指的是对其近似算法的研究，所谓近似算法即该算法可以求得与精确解接近的结果但不一定得到精确解。目前，已经提出了大量的近似算法，其中适应近似算法是目前时间复杂性比较低的一种近似算法。如下次适应（NF）算法、首次适应（FF）算法、最佳适应（BF）算法、降序首次适应(FFD)算法、降序最佳适应(BFD)算法等。

装箱问题中最早被研究的是一维装箱问题。随着研究的深入，人们发现实际生活中更多存在的是一些带约束的装箱问题，因此也就抽象化出了，如二维装箱问题（条形装箱问题、剪裁问题）、三维装箱问题、变容装箱问题、有色装箱问题、对偶装箱问题等等一系列的带约束的装箱问题。一维装箱问题是指要求把一些物品放入到具有固定容量的箱子中，并最小化所使用的箱子数目。本文介绍的就是一维装箱问题。

数学模型

假设我们有一堆物品长度分别为 $w_{i} $，若干长度为 $C (C>w_i)$ 的箱子；问：最少需要几个箱子才能把所有物品全部装进箱子？我们首先对问题进行一个简单的分析，回答两个简单的问题。

- 最坏的情况是什么？

  因为物品的长度小于箱子的长度，所以不存在一件物品一个箱子装不下的情况。因此，最坏的情况就是一个物品放一个箱子，所以可行解的上界是物品的个数 n。

- 最理想的情况是什么？

  最理想的情况就是每个箱子都填满，我们假设物品可以分割，那么每个箱子都可以被填满，在这种情况下所需要的箱子数应该为 $₩lceil ₩frac{₩sum w_i}{C} ₩rceil$ 。

综上所述，OPT 的取值范围应当为 $[₩lceil ₩frac{₩sum w_i}{C} ₩rceil, n]$。所以我们编写的算法得出来的箱子个数应该在这个范围内，而且越接近 $₩lceil ₩frac{₩sum w_i}{C} ₩rceil $，算法越好。

MathJax.Hub.Config({tex2jax: {inlineMath: [['$','$'], ['₩₩(','₩₩)']]}});

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（过往记忆）所有，未经许可不得转载。
本文链接: 【】（）