Advances in Applied Mathematics 应用数学进展, 2022, 11(1), 434-441
Published Online January 2022 in Hans. http://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2022.111051

Hans汉斯

具有稳定性约束的一维箱子覆盖问题

吴彬彬[1]，苗睿卿[1]，张同全[2]

[1]云南民族大学数学与计算机科学学院，云南 昆明
[2]云南民族大学预科教育学院，云南 昆明

收稿日期：2021年12月26日；录用日期：2022年1月16日；发布日期：2022年1月28日

摘 要

给定一个物品序列和若干箱子，如何合理地放置物品以使得最多的箱子被覆盖就称为箱子覆盖问题。作为一维箱子覆盖问题的推广，提出了具有稳定性约束的一维箱子覆盖问题，即要求同一箱子中小的物品置于大的物品上方。分析了问题的NP-完备性，给出了一种弱渐进近似算法，并分析了算法的时间复杂度。

关键词

箱子覆盖问题，NP-完备性，稳定性约束，弱渐进近似算法

One-Dimensional Bin-Covering Problem with Stability Constraints

Binbin Wu[1], Ruiqing Miao[1], Tongquan Zhang[2]

[1]School of Mathematics and Computer Sciences, Yunnan Minzu University, Kunming Yunnan
[2]School of Preparatory Education, Yunnan Minzu University, Kunming Yunnan

Received: Dec. 26th, 2021; accepted: Jan. 16th, 2022; published: Jan. 28th, 2022

Abstract

The bin-covering problem is to reasonably place the items under a given sequence of items and a number of bins, so that most bins are covered. For promoting the one-dimensional bin-covering problem, the paper proposes the one-dimensional bin-covering problem with stability constraints, which requires the small items in the same bin to be placed above the large items. The paper analyzes the NP-completeness of the problem and provides a weakly asymptotic approximation algorithm for the problem by analyzing the time complexity of the algorithm.

文章引用: 吴彬彬, 苗睿卿, 张同全. 具有稳定性约束的一维箱子覆盖问题[J]. 应用数学进展, 2022, 11(1): 434-441.
DOI: 10.12677/aam.2022.111051

Keywords

Bin-Covering Problem, NP-Completeness, Stability Constraints, Weakly Asymptotic Approximation Algorithm

1. 引言

本文将稳定性约束引入到一维箱子覆盖问题中，从而形成了一个新的问题——具有稳定性约束的一维箱子覆盖问题，该问题是一个与实际生活相关的带约束的箱子覆盖问题。比如在装修房屋时，由于装修材料有限，所以只能装修部分房屋，目标是在给定有限材料且规定用料顺序(对每个房屋，先用较大材料再用较小材料)的情况下使得被装修的房屋尽可能多。类似地，生活中还有很多相关例子。

一维箱子覆盖问题[1] [2]的描述为：给定含有 n 个物品的序列及物品尺寸，提供若干容量为l的箱子，要求用给定的物品序列中的所有物品来覆盖所给箱子，这里，箱子被覆盖指的是装入该箱子的所有物品的尺寸之和不小于 l，目标是使得被覆盖的箱子数目达到最大。由于一维箱子覆盖问题是 NP-完备的[2]，即不存在一个可求得一维箱子覆盖问题最优解的多项式时间算法(如果 P ≠ NP)，而一维箱子覆盖问题在日常生活中具有广泛应用背景及现实意义，人们不断尝试着设计(渐进)近似算法以求得该问题的近似解，并证明(渐进)近似算法的(渐进)近似值及时间复杂性，用这两个指标来判断所设计算法的好坏。

装箱问题的研究主要起源于 20 世纪 70 年代，近几十年来越来越多的研究者对装箱问题展开广泛而深入的研究[3]，并取得一些较好的成果[4] [5] [6] [7]。

1983 年，Assmann 等人[1]对一维箱子覆盖问题进行研究，设计出了解决该问题的一些渐进近似算法，如 NF (Next Fit)算法和 FFD (First Fit Decreasing)算法，并证明以上两个算法的渐进近似值分别为 2 和 $\frac{3}{2}$，以及它们的时间复杂度分别为 $O(n)$ 和 $O(n\log n)$，另外，Assmann 等人还给出了一个求解一维箱子覆盖问题的离线近似算法，其渐进近似值为 $\frac{4}{3}$，时间复杂度为 $O\left(n\log^2 n\right)$。1999 年，Woeginger 等人[8]对变尺寸一维箱子覆盖问题进行研究，并给出了一个求解该问题的渐进近似算法。孙春玲和李建平[9]两人讨论了最小基数箱子覆盖问题，并针对该问题给出了一个时间复杂度为 $O(n)$ 的启发式算法。2016 年，唐浩龙[10]讨论了一维捆绑式箱子覆盖问题，并设计了三个算法来解决该问题，即 K-NF 算法、K-FFD 算法和 K-T 算法，这三种算法的渐进近似值分别为 2、$\frac{3}{2}$ 和 2，时间复杂度分别为 $O(n)$、$O(n\log n)$ 和 $O(n)$。

本文提出一种带约束的箱子覆盖问题——具有稳定性约束的一维箱子覆盖问题，分析了问题的 NP-完备性，给出了一种求解该问题的弱渐进近似算法，并分析了该算法的时间复杂度。

2. 问题描述

经典的一维箱子覆盖问题为：给定含有 n 个物品的序列 $I=(a_1,a_2,\cdots,a_n)$，每个物品具有相应尺寸 $s(a_i)\in(0,l]$，这里，$i=1,2,\cdots,n$，提供若干个容量为 l 的箱子，要求用给定物品序列中的所有物品来覆盖所给箱子，使得被覆盖箱子的数目达到最大。这里，箱子被覆盖指的是装入该箱子中所有物品的尺寸

之和不小于箱子的容量。

现在给出具有稳定性约束的一维箱子覆盖问题的定义：给定含有 n 个物品的序列 $I=(a_1,a_2,\cdots,a_n)$，每个物品具有相应尺寸 $s(a_i)\in(0,l]$，这里，$i=1,2,\cdots,n$，提供若干个容量为 l 的箱子，要求将所有物品放入箱中，且小的物品放在大的物品上方，目标是使得被覆盖的箱子数目达到最大。这里，箱子被覆盖指的是装入该箱子中所有物品的尺寸之和不小于箱子的容量。

3. 研究内容

这一部分是本文研究的主要内容，包括对于具有稳定性约束的一维箱子覆盖问题的模型建立、算法设计思路、算法设计和算法分析。

3.1. 模型建立

$$Z=\max\sum_{j=1}^{\left\lfloor\frac{\sum_{i=1}^{n}s(a_i)}{l}\right\rfloor}y_j$$

s.t.

$$\sum_{i=1}^{n}s(a_i)x_{ij}\geq y_j\cdot l,\ \ j=1,2,\cdots,\left\lfloor\frac{\sum_{i=1}^{n}s(a_i)}{l}\right\rfloor$$

$$\sum_{j=1}^{\left\lfloor\frac{\sum_{i=1}^{n}s(a_i)}{l}\right\rfloor}x_{ij}=1,\ \ i=1,2,\cdots,n$$

若 $x_{kj}=x_{rj}=1$ 且 $s(a_k)\geq s(a_r)$，则物品 a_r 应置于物品 a_k 上方

$$y_j=\begin{cases}1, & \text{箱子}B_j\text{被使用}\\ 0, & \text{箱子}B_j\text{未被使用}\end{cases},\ \ j=1,2,\cdots,\left\lfloor\frac{\sum_{i=1}^{n}s(a_i)}{l}\right\rfloor$$

$$x_{ij}=\begin{cases}1, & \text{物品}a_i\text{被装入箱子}B_j\text{中}\\ 0, & \text{物品}a_i\text{未被装入箱子}B_j\text{中}\end{cases},\ \ i=1,2,\cdots,n;\ j=1,2,\cdots,\left\lfloor\frac{\sum_{i=1}^{n}s(a_i)}{l}\right\rfloor$$

3.2. 算法设计思路

具有稳定性约束的一维箱子覆盖问题是在经典的一维箱子覆盖问题上加上稳定性约束，目标是使得被覆盖的箱子数目达到最大。本文设计的算法是受求解一维捆绑式箱子覆盖问题的 K-FFD 算法以及求解经典一维箱子覆盖问题的 CF 算法启发。

第一，在 K-FFD 算法的理论基础上，我们考虑当前研究的问题的目标是使被覆盖箱子数目达到最大，故考虑在算法的执行过程中尽量减少对物品的浪费。

第二，在 CF 算法的理论基础上，我们考虑当前研究的问题的约束条件，故在将物品按尺寸大小非增序排列的情况下，考虑在装了当前物品序列中最大的一个物品后，改变 CF 算法中从物品序列最右端自右至左依次装物品的策略。本文对物品按尺寸大小非增序排列并分类，分为大物品、小物品，在当前箱子中放一个大物品后，从小物品序列最左端自左至右依次装物品，这一策略与稳定性约束这一约束条件是相符的。

综合以上两点考虑以及对研究问题本身的讨论，得到求解具有稳定性约束的一维箱子覆盖问题的一

个弱渐进近似算法。

3.3. 算法设计

为了解决具有稳定性约束的一维箱子覆盖问题，本节设计了一个弱渐进近似算法。为了叙述方便，这里规定：如果 $s(a_i)>\frac{1}{2}l$，则 a_i 为大物品；如果 $s(a_i)\le\frac{1}{2}l$，则 a_i 为小物品。该算法策略为：1) 将物品按尺寸进行非增序排列，并划分为大、小物品。2) 将最大的大物品装入第一个箱子，然后按排列好的物品顺序将小物品依次装入第一个箱子中，直至第一个箱子被覆盖为止，关闭该箱子。3) 将次大的大物品装入第二个箱子，然后按排列好的物品顺序将余下小物品依次装入第二个箱子中，直至第二个箱子被覆盖为止，关闭该箱子。4) 将余下的大、小物品按以上方法进行装箱。直至大(小)物品装完，则对小(大)物品使用 NF 算法进行装箱。直至装完所有物品。5) 若存在非空未被覆盖的箱子，则将该箱子中的所有物品装入已被覆盖的箱子中。

本节所设计的算法的具体描述如下：

算法 1.1

Input：物品序列 $I=(a_1,a_2,\cdots,a_n)$ 以及每个物品的尺寸 $s(a_i)\in(0,l]$，这里，$i=1,2,\cdots,n$，容量为 l 的箱子若干。

Output：被覆盖的箱子数目 h 以及装箱方案 $B_1,B_2,\cdots,B_h$。

Begin

step1：对物品进行非增序排列，不妨设

$l\ge s(a_1)\ge s(a_2)\ge\cdots\ge s(a_{k-1})>\frac{1}{2}l\ge s(a_k)\ge s(a_{k+1})\ge\cdots\ge s(a_n)>0$。

step2：置 $j:=1,i:=1,B_1:=\{a_1\},s(B_1):=s(a_1)$。

step3：For $m=k$ to n do。

If ($s(B_j)+s(a_m)<l$) then

置 $B_j:=B_j\cup\{a_m\},s(B_j):=s(B_j)+s(a_m)$。

If ($m=n$) then

令 $i:=i+1$，置 $B_j:=B_j\cup\{a_i\}$，关闭 B_j，记 $h=j$，令 $j:=j+1$。

For $c=i$ to $k-1$ do。

If ($s(B_j)+s(a_c)<l$) then

置 $B_j:=B_j\cup\{a_c\},s(B_j):=s(B_j)+s(a_c)$。

Else

置 $B_j:=B_j\cup\{a_c\},s(B_j):=s(B_j)+s(a_c)$，关闭 B_j，记 $h=j$，令 $j:=j+1$。

Else

Else

置 $B_j:=B_j\cup\{a_m\},s(B_j):=s(B_j)+s(a_m)$，关闭 B_j，记 $h=j$。

If ($i<k-1$ and $m=n$) then

关闭 B_j，记 $h=j$，令 $j:=j+1,B_j:=\varnothing,s(B_j):=0$。

For $c=i$ to $k-1$ do。

If ($s(B_j)+s(a_c)<l$) then

置 $B_j:=B_j\cup\{a_c\},s(B_j):=s(B_j)+s(a_c)$。

Else

置 $B_j := B_j \cup \{a_c\}, s(B_j) := s(B_j) + s(a_c)$，关闭 B_j，记 $h = j$，令 $j := j+1$。

Else if ($i < k-1$ and $m < n$) then

关闭 B_j，记 $h = j$，令 $j := j+1, i := i+1$，置 $B_j := B_j \cup \{a_i\}, s(B_j) := s(B_j) + s(a_i)$。

Else if ($i = k-1$) then

关闭 B_j，记 $h = j$，令 $j := j+1$，置 $B_j := \varnothing, s(B_j) := 0$。

Step4：若存在非空未被覆盖的箱子，则将该箱子中的所有物品装入已被覆盖的箱子中。

If ($j > h$) then

置 $B_h := B_h \cup B_j$。

Step5：输出被覆盖的箱子数目 h 以及装箱方案 $B_1, B_2, \cdots, B_h$。

End

3.4. 算法分析

定理 1.1：具有稳定性约束的一维箱子覆盖问题是 NP-完备的。

证明：由于一维箱子覆盖问题是 NP-完备的，具有稳定性约束的一维箱子覆盖问题是一维箱子覆盖问题的推广，故具有稳定性约束的一维箱子覆盖问题是 NP-完备的，不存在一个多项式时间算法可以求得其最优解。

定理 1.2：对于具有稳定性约束的一维箱子覆盖问题的任意实例 I，算法 1.1 是一个弱渐进近似值为 2 的算法。

证明：这里我们用 $OUT(I)$ 表示用算法 1.1 装 I 中物品被覆盖的箱子个数，$OPT(I)$ 表示用最优算法装 I 中物品被覆盖的箱子个数。

根据算法 1.1 中 step4，下面将分为 2 种情形进行分析：

情形 1：不存在非空未覆盖的箱子

不妨设所使用的箱子序列为 $\{B_1, B_2, \cdots, B_{OUT(I)}\}$，由于小物品的尺寸不超过 $\frac{1}{2}l$，则每个被覆盖的箱子都满足

$$l \le s(B_j) < \frac{3}{2}l，\text{这里 } j = 1, 2, \cdots, OUT(I)$$

情形 1.1：若执行 step4 时大物品先装完或大、小物品在同一箱中装完，则

$$OPT(I) \cdot l \le \sum_{i=1}^{n} s(a_i) < OUT(I) \cdot \frac{3}{2}l$$

则

$$OPT(I) < \frac{3}{2}OUT(I)$$

$$OPT(I) \le \frac{3}{2}OUT(I) - 1$$

$$\frac{3}{2}OUT(I) \ge OPT(I) + 1$$

即

$$OUT(I) \ge \frac{2}{3}OPT(I) + \frac{2}{3}$$

情形 1.2：若执行 step4 时小物品先装完，假设装完小物品后仅装大物品的箱子是 $B_t, B_{t+1}, \cdots, B_{OUT(I)}, \left(OUT(I) \geq t\right)$。

1) 若 $OUT(I)=t$，则

$$OPT(I)\cdot l \leq \sum_{i=1}^{n} s(a_i) < \left[OUT(I)-1\right]\cdot \frac{3}{2}l + 2l$$

$$OPT(I) < \frac{3}{2}OUT(I) - \frac{3}{2} + 2$$

$$OPT(I) < \frac{3}{2}OUT(I) + \frac{1}{2}$$

$$OPT(I) \leq \frac{3}{2}OUT(I) - \frac{1}{2}$$

即

$$OUT(I) \geq \frac{2}{3}OPT(I) + \frac{1}{3}$$

2) 若 $OUT(I) > t$，则

$$OPT(I)\cdot l \leq \sum_{i=1}^{n} s(a_i) < (t-1)\cdot \frac{3}{2}l + \left[OUT(I)-t+1\right]\cdot 2l$$

$$OPT(I) < \frac{3}{2}t - \frac{3}{2} + 2OUT(I) - 2t + 2$$

$$OPT(I) < 2OUT(I) - \frac{1}{2}t + \frac{1}{2}$$

$$OPT(I) < 2OUT(I) + \frac{1}{2}$$

$$OPT(I) \leq 2OUT(I) - \frac{1}{2}$$

$$2OUT(I) \geq OPT(I) + \frac{1}{2}$$

即

$$OUT(I) \geq \frac{1}{2}OPT(I) + \frac{1}{4}$$

情形 2：存在非空未覆盖的箱子

由算法 1.1 的执行过程易知，至多存在 1 个非空未被覆盖的箱子。不妨设所使用的箱子序列为 $\left\{B_1, B_2, \cdots, B_{OUT(I)}, B_{OUT(I)+1}\right\}$，则前 $OUT(I)$ 个箱子都被覆盖，第 $OUT(I)+1$ 个箱子未被覆盖。

情形 2.1：若执行 step4 时大物品先装完或大、小物品在同一箱中装完，则每个被覆盖的箱子都满足

$l \leq s\left(B_j\right) < \frac{3}{2}l$，这里 $j = 1, 2, \cdots, OUT(I)$

则

$$OPT(I)\cdot l \leq \sum_{i=1}^{n} s(a_i) < OUT(I)\cdot \frac{3}{2}l + l$$

$$OPT(I)<\frac{3}{2}OUT(I)+1$$

$$OPT(I)\le\frac{3}{2}OUT(I)$$

即

$$OUT(I)\ge\frac{2}{3}OPT(I)$$

情形 2.2：若执行 step4 时小物品先装完，假设装完小物品后仅装大物品的箱子是 $B_t,B_{t+1},\cdots,B_{OUT(I)+1},(OUT(I)\ge t)$。

1) 若 $OUT(I)=t$，则

$$OPT(I)\cdot l\le\sum_{i=1}^{n}s(a_i)<OUT(I)\cdot\frac{3}{2}l+l$$

$$OPT(I)<\frac{3}{2}OUT(I)+1$$

$$OPT(I)\le\frac{3}{2}OUT(I)$$

即

$$OUT(I)\ge\frac{2}{3}OPT(I)$$

2) 若 $OUT(I)>t$，由于大物品的尺寸不超过 l，小物品的尺寸不超过 $\frac{1}{2}l$，则每个被覆盖的箱子都满足

$$l\le s(B_j)<\frac{3}{2}l，\text{这里 } j=1,2,\cdots,t-1$$

$$l\le s(B_j)<2l，\text{这里 } j=t,t+1,\cdots,OUT(I)$$

则

$$OPT(I)\cdot l\le\sum_{i=1}^{n}s(a_i)<(t-1)\cdot\frac{3}{2}l+\left[OUT(I)-t+1\right]\cdot 2l+l$$

$$OPT(I)<\frac{3}{2}t-\frac{3}{2}+2OUT(I)-2t+2+1$$

$$OPT(I)<2OUT(I)-\frac{1}{2}t+\frac{3}{2}$$

$$OPT(I)<2OUT(I)+\frac{3}{2}$$

$$OPT(I)\le 2OUT(I)+\frac{1}{2}$$

$$2OUT(I)\ge OPT(I)-\frac{1}{2}$$

即

$$OUT(I) \geq \frac{1}{2} OPT(I) - \frac{1}{4}$$

不难看出，情形 1~2 涵盖了所有可能情况。综上可知，对于具有稳定性约束的一维箱子覆盖问题的任意实例 I，算法 1.1 是一个弱渐进近似值为 2 的算法。

定理 1.3： 算法 1.1 的时间复杂度为 $O(n\log n)$。

证明：算法 1.1 执行过程中，step1 的步数最多不会超过 $O(n\log n)$；step2 的步数最多为 4；step3 中的 For 步数最多为 $O(n)$，step3 中步数最多为 $O(n)$，step5 步数为 2。综上，算法 1.1 的时间复杂度为 $O(n\log n)$。

4. 结束语

本文将稳定性约束考虑到经典的一维箱子覆盖问题中，定义一种新的箱子覆盖问题的约束类型，设计用于求解该问题的弱渐进近似算法，并分析算法的时间复杂度。对这一新问题的研究和讨论可以丰富组合优化领域的研究内容，另外，针对其提出的算法可以较有效地提高生活中类似问题中资源的利用率。

致　谢

非常感谢对本稿提出意见的审稿人及导师。

基金项目

云南民族大学数学与计算机科学学院研究生项目(SJXY-2021-026)。

参考文献

[1] Assmann, S.F., Johnson, D.S., Kleitman, D.J. and Leung, J.Y.-T. (1984) On a Dual Version of the One-Dimensional Bin Packing Problem. *Journal of Algorithms*, **5**, 502-525. https://doi.org/10.1016/0196-6774(84)90004-X

[2] Han, S., Hong, D. and Leung, J.Y.-T. (1996) Leung. Probabilistic Analysis of a Bin Covering Algorithm. *Operations Research Letters*, **18**, 193-199. https://doi.org/10.1016/0167-6377(95)00053-4

[3] Gehring, H., Menschner, K. and Meyer, M. (1990) A Computer-Based Heuristic for Packing Pooled Shipment Containers. *European Journal of Operational Research*, **44**, 277-289. https://doi.org/10.1016/0377-2217(90)90363-G

[4] Maruyama, K., Chang, S.K. and Tang, D.T. (1977) A General Packing Algorithm for Multidimensional Resource Requirements. *International Journal of Computer & Information Sciences*, **6**, 131-149. https://doi.org/10.1007/BF00999302

[5] Bartal, Y., Leonardi, S., Marchetti-Spaccamela, A., *et al*. (2000) Multiprocessor Scheduling with Rejection. *SIAM Journal on Discrete Mathematics*, **13**, 64-78. https://doi.org/10.1137/S0895480196300522

[6] Gnanavelbabu, A., Caldeira Rylan, H. and Vaidyanathan, T. (2021) A Simulation-Based Modified Backtracking Search Algorithm for Multi-Objective Stochastic Flexible Job Shop Scheduling Problem with Worker Flexibility. *Applied Soft Computing Journal*, **113**, Article ID: 107960. https://doi.org/10.1016/j.asoc.2021.107960

[7] 程会兵. 考虑时间窗约束的装箱问题研究[[D]: [硕士学位论文]. 南昌: 江西财经大学, 2019.

[8] Woeginger, G.J. and Zhang, G. (1999) Optimal On-Line Algorithms for Variable-Sized Bin Covering. *Operations Research Letters*, **25**, 47-50. https://doi.org/10.1016/S0167-6377(99)00023-1

[9] 孙春玲, 李建平. 最小基数箱子覆盖问题及其启发式算法[J]. 云南大学学报(自然科学版), 2004(S1): 8-11.

[10] 唐浩龙. 一维捆绑式箱子覆盖问题[D]: [硕士学位论文]. 昆明: 云南大学, 2016.