云南大学学报 (自然科学版), 2004, 26 (5) : 392～396　　CN 53 - 1045/ N　ISSN 0258 - 7971
Journal of Yunnan University

装箱问题的一种新的近似算法*

孙春玲, 陈智斌, 李建平
(云南大学 数学系,云南 昆明　650091)

摘要:研究了一维装箱问题(Bin Packing Problem),给出了一个新的近似算法:交叉装填算法(简称 CF 算法). 证明了 CF 算法达到装箱问题的最好的近似值 $\frac{3}{2}$;并且当这些物件的大小按非增性质预先排序后,CF 算法的时间复杂度是线性的.

关键词:一维装箱问题;NP - 完备;近似算法
中图分类号:O 157.6;TP 301.5　　**文献标识码**:A　　**文章编号**:0258 - 7971(2004)05 - 0392 - 05

在 70 年代初期 NP - 完备性刚建立时,一维的装箱问题(Bin Packing Problem)就得到了广泛研究,取得了许多好的成果. 同时因为 $P \neq NP$ 的著名猜想,解决一维装箱问题的一些好的算法一直以来都被作为近似算法的经典范例,其原因不仅因为对一维装箱问题的研究能提供人们一个学习、熟悉、甚至应用近似算法的基础,而且还因为一维装箱问题有着极其广泛的应用背景. 例如有装载量限制的装载卡车问题及库存削减问题等.

下面给出一维装箱问题的数学描述:

任意给定的一个序列 $L = (a_1, a_2, \cdots, a_n)$,表示 n 个物件, 每个物件的大小 $s(a_i) \in (0,1]$, 其中 $i = 1,2, \cdots, n$, 此处有一些容量为 1 的箱子(bins). 一维的装箱问题要求给出一种方案, 把物件 $a_1, a_2, \cdots, a_n$ 全部放入上述容量为 1 的某些箱子中,使得所用箱子数目最少.

由于一维装箱问题是 NP - 完备的[1]. 在历史上产生了很多装箱问题的近似算法,较为经典的算法有 Next - Fit(NF) 和 First - Fit(FF). NF 算法能获得 $NF(L) \leq 2OPT(L) - 1$ 的结果, 这里 $OPT(L)$ 是最优的箱子数目, 而 FF 算法能取得 2 的近似值[2].

对一维装箱问题有下面的经典结果来说明其困难程度.

定理 1[3]　如果 $P \neq NP$,那么对于任意给定的 $\varepsilon > 0$,不存在近似值为 $\frac{3}{2} - \varepsilon$ 的近似算法来解决一维装箱问题.

因此,这个 $\frac{3}{2}$ 近似值是我们对装箱问题的近似算法的最好期待. 事实上, 人们得到的很多近似算法的近似值都是朝这个目标迈进[2]. 本文给出的算法已经解决了这个问题. 在下面的证明过程中可以看到算法能达到最优的近似值 $\frac{3}{2}$,同时算法的复杂度也能达到非线性最优 $O(n\log n)$. 特别, 当物件的大小(size) 按非增性质预先排序后, CF 算法能够在线性时间内找到近似解, 其近似值为 $\frac{3}{2}$. 我们给出的算法的主要思想是:先把物件按从大到小的顺序排列, 然后首先把左端最大的一个物件放入一个箱子, 再从物件序列的最右端开始从右往左依次把物件放入箱子中, 直到下一个物件不能放入该箱子, 那么开启新的箱子, 重复上述装填步骤, 直到所有的物件都装入箱子中.

1　主要结果

为了叙述方便, 首先给出一些概念与符号. 令 $L = (a_1, a_2, \cdots, a_n)$ 为任意给定的 n 个物件的一

---

* **收稿日期**:2003 - 12 - 24
基金项目:国家自然科学研究基金资助项目(10271103);云南省自然科学研究基金资助项目(2003F0015M).
作者简介:孙春玲(1980 -　),女,云南人,硕士生,主要从事组合优化方面的研究.
李建平(1965 -　),男,云南人,教授,主要从事组合优化方面的研究.

个序列. 每个物件的大小 $s(a_i) \in (0,1]$,其中 $i \in \{1,2,\cdots,n\}$, B_j 表示所用的第 j 个箱子,箱子 B_j 的容量(也称长度) 均为 1, $j \in \{1,2,\cdots,m\}$,这里 m 表示一种给定的装箱方案所需的箱子数目. 易知,不同算法产生的箱子数目不必相同. 令 $\sigma(B_j)$ 表示箱子 B_j 中所有物件的数目. 令 $f(B_j)$ 表示装入箱子 B_j 中的所有物件的大小之和. a_{j_k} 表示按照算法放入第 j 个箱子的第 k 个物件,其中 $j \in \{1,2,\cdots,m\}$.

因此我们有

$$f(B_j) = \sum_{k=1}^{\sigma(B_j)} s(a_{j_k}),\ j = 1,2,\cdots,m,$$

$$\sum_{j=1}^{m} f(B_j) = \sum_{i=1}^{n} s(a_i).$$

下面给出装箱问题的交叉装填算法:

算法 1　交叉装填算法(CF 算法)

输入: $L = (a_1, a_2, \cdots, a_n)$ 及每个物件的大小 $s(a_i) \in (0,1]$, $i = 1,2,\cdots,n$.

输出:需要的箱子数目 m.

step 1:把物件 $a_1, a_2, \cdots, a_n$ 按其大小进行非增序排列,不妨设为 $s(a_1) \geq s(a_2) \geq \cdots \geq s(a_n)$.

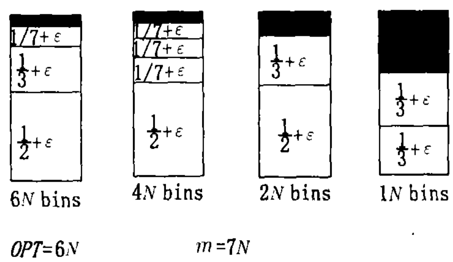
step 2:首先把 a_1 放入箱子 B_1 中,然后从最右端开始,依次把物件 $a_n, a_{n-1}, \cdots$ 放入 B_1,直到下一个物件不能再放入箱子为止,开启新的箱子 B_2.

step 3:设在第 i 步循环时,打开第 i 个箱子,此时把物件 a_i 放入 B_i 中,再从最右端开始从右往左把物件依次放入 B_i 中. 假设第 $i-1$ 个箱子 B_{i-1} 中最后一个放入的物件为 a_k,则在第 i 步循环时最右端的物件为 a_{k-1},那么当 $s(a_i) + s(a_{k-1}) + \cdots + s(a_l) \leq 1$,且 $s(a_i) + s(a_{k-1}) + \cdots + s(a_l) + s(a_{l-1}) > 1$ 时,把 $a_{k-1}, a_{k-2}, \cdots, a_l$ 放入 B_i 中,开启新的箱子 B_{i+1}. 直到把物件 $a_1, a_2, \cdots, a_n$ 放入 m 个箱子.

step 4:输出 m.

为了便于更好的理解算法,我们利用 CF 算法来计算下面一个例子所需要的箱子数目.

例 1　$$L = \Big(\overbrace{\tfrac{1}{7}+\varepsilon,\cdots,\tfrac{1}{7}+\varepsilon}^{6N},\overbrace{\tfrac{1}{3}+\varepsilon,\cdots,\tfrac{1}{3}+\varepsilon}^{6N},\overbrace{\tfrac{1}{2}+\varepsilon,\cdots,\tfrac{1}{2}+\varepsilon}^{6N}\Big)$$



2　算法分析

下面讨论 CF 算法的近似值及时间复杂性. 为了证明 CF 算法是一维装箱问题的一个 $\frac{3}{2}$ - 近似算法,我们首先给出引理 1:

引理 1　由 CF 算法得到的 m 个箱子中,若至少存在 $m-1$ 个箱子,每个箱子中的物件数目不少于 2,即 $\sigma(B_j) \geq 2$, $j \in \{1,2,\cdots,m\}$ 则这 $m-1$ 个箱子中的任何一个箱子所装的物件容量之和大于 $\frac{2}{3}$.

证明　若不然,不妨设存在 2 个箱子 B_i 和 B_j 使得, $f(B_i) \leq \frac{2}{3}$, $f(B_j) \leq \frac{2}{3}$, $1 \leq i < j \leq m$. 则 B_i 中所装物件为 $a_{i_1}, a_{i_2}, \cdots, a_{i_{\sigma(B_i)}}$, B_j 中所装物件为 $a_{j_1}, a_{j_2}, \cdots, a_{j_{\sigma(B_j)}}$,这里 a_{i_k} 和 $a_{j_{k'}}$, $k \in \{1,2,\cdots,\sigma(B_i)\}$, $k' \in \{1,2,\cdots,\sigma(B_j)\}$ 分别为第 k 个和第 k' 个装入 B_i 和 B_j 中的物件. 按算法顺序 $a_{i_{\sigma(B_i)}}$, $a_{j_{\sigma(B_j)}}$ 分别为最后一个装入 B_i 和 B_j 和中的物件.

由假设知

$$f(B_i) = s(a_{i_1}) + s(a_{i_2}) + \cdots + s(a_{i_{\sigma(B_i)}}) \leq \frac{2}{3}, \tag{1}$$

$$f(B_j) = s(a_{j_1}) + s(a_{j_2}) + \cdots + s(a_{j_{\sigma(B_j)}}) \leq \frac{2}{3}, \tag{2}$$

且由算法知以下性质成立

$$s(a_{i_2}) \leq s(a_{i_3}) \leq \cdots \leq s(a_{i_{\sigma(B_i)}}) \leq s(a_{i_1}) = s(a_i), \tag{3}$$

$$s(a_{j_2}) \leq s(a_{j_3}) \leq \cdots \leq s(a_{j_{\sigma(B_j)}}) \leq s(a_{j_1}) = s(a_j), \tag{4}$$

又因为 $i < j$,所以

$$s(a_{i_1}) \geq s(a_{j_1}), \tag{5}$$

并且

$$s(a_{i_2}) \leq \cdots \leq s(a_{i_{\sigma(B_i)}}) \leq s(a_{j_2}) \leq \cdots \leq s(a_{j_{\sigma(B_j)}}) \leq s(a_{j_1}) \leq s(a_{i_1}). \tag{6}$$

现在考虑第 $i+1$ 个箱子 B_{i+1},令 $a_{(i+1)_2}$ 为按照算法CF第2个放入 B_{i+1} 的物件,此处 $i+1 \leq j$. 根据算法 CF, 此时必有: $f(B_i) + s(a_{(i+1)_2}) > 1$(否则 $a_{(i+1)_2}$ 可以放入 B_i 中). 又因为 $s(a_{(i+1)_2}) \leq s(a_{j_2})$,所以只需证明: $f(B_i) + s(a_{j_2}) \leq 1$ 就可以推出矛盾,从而引理成立.

下面分情况讨论:

①当 $s(a_{j_1}) \geq \frac{1}{3}$,此时必有 $\frac{1}{3} \leq s(a_{j_1}) \leq s(a_{i_1}) \leq \frac{2}{3}$,由(1),(2)知 $f(B_i) + f(B_j) \leq \frac{4}{3}$ 即

$$s(a_{i_1}) + \cdots + s(a_{i_{\sigma(B_i)}}) + s(a_{j_1}) + \cdots + s(a_{j_{\sigma(B_j)}}) \leq \frac{4}{3},$$

有

$$s(a_{i_1}) + \cdots + s(a_{i_{\sigma(B_i)}}) + s(a_{j_2}) + \cdots + s(a_{j_{\sigma(B_j)}}) \leq \frac{4}{3} - s(a_{j_1}) \leq 1,$$

所以

$$s(a_{i_1}) + \cdots + s(a_{i_{\sigma(B_i)}}) + s(a_{j_2}) \leq 1,$$

即

$$f(B_i) + s(a_{j_2}) \leq 1;$$

②当 $s(a_{j_1}) < \frac{1}{3}$,由(4)知

$$s(a_{j_2}) \leq \cdots \leq s(a_{j_{\sigma(B_j)}}) \leq s(a_{j_1}) < \frac{1}{3},$$

又因为

$$f(B_i) \leq \frac{2}{3},$$

所以

$$f(B_i) + s(a_{j_2}) < \frac{2}{3} + \frac{1}{3} = 1.$$

故如果存在2个箱子 $B_i, B_j, 1 \leq i < j \leq m$ 使得 $f(B_i) \leq \frac{2}{3}, f(B_j) \leq \frac{2}{3}$,则第 $i+1$ 个箱子中第2个放入的物件 $a_{(i+1)_2}$ 仍可放入箱子 B_i 中,这与算法 CF 的选法矛盾,故引理得证.

引理2 由算法CF得到的输出解 m 中,若至少有 $m-1$ 个箱子,每个箱子中的物件数目不少于2,那么CF算法是一维装箱问题的一个 $\frac{3}{2}$ - 近似算法.

证明 设CF算法的输出值为 m,其问题本身的最优值为 OPT,由假设及引理1知若至少有 $m-1$ 个箱子,每个箱子中的物件数目不少于2,那么其所装的物件容量之和大于 $\frac{3}{2}$,则存在 $m^* \in \{1,2,\cdots,m\}$, 使得对任意的 $j \in \{1,2,\cdots,m\} \setminus \{m^*\}$ 有 $f(B_j) > \frac{2}{3}$,则

$$\sum_{j=1}^{m} f(B_j) = \sum_{\substack{j=1 \\ j \neq m^*}}^{m} f(B_j) + f(B_{m^*}).$$

此时我们有如下论断:

论断 由算法知 B_{m^*} 中的所有物件不能放入其余 $m-1$ 个箱子中的任何一个箱子内.

若不然,则存在一个箱子 B_i,使得 B_{m^*} 中的物件全部放入箱子 B_i 中,即 $f(B_i) + f(B_{m^*}) \leq 1$,当 $i < m^*$ 时,必然有箱子 $B_{i+1}(i+1 \leq m^*)$ 中的一个物件 $a_{(i+1)_2}$,使得 $s(a_{(i+1)_2}) \leq f(B_{m^*})$,所以 $f(B_i) + s(a_{(i+1)_2}) \leq 1$,这表明 $a_{(i+1)_2}$ 仍可放入 B_i 中,与算法矛盾. 当 $i > m^*$ 时,由于 $f(B_i) + f(B_{m^*}) \leq 1$,说明 B_i 中的物件也可全部放入 B_{m^*} 中,所以必然有箱子 B_{m^*+1} 中的一个物件 $a_{(m^*+1)_2}$ 使得 $s(a_{(m^*+1)_2}) \leq f(B_{m^*})$, 即 $f(B_{m^*}) + s(a_{(m^*+1)_2}) \leq 1$,这也表明 $a_{(m^*+1)_2}$ 仍可放入 B_{m^*} 中,与算法矛盾,故论断成立.

利用该论断,我们得到下面事实

$$f(B_{m^*}) > \max_{i=1,2,\cdots,m,\, i \neq m^*} \{1 - f(B_i)\} \geq \frac{1}{m-1} \sum_{\substack{i=1 \\ i \neq m^*}}^{m} (1 - f(B_i)) = 1 - \frac{1}{m-1} \sum_{\substack{i=1 \\ i \neq m^*}}^{m} f(B_i),$$

$$f(B_{m^*}) - \frac{1}{m-1} f(B_{m^*}) > 1 - \frac{1}{m-1} \sum_{i=1}^{m} f(B_i) > 1 - \frac{1}{m-1} OPT,$$

即

$$\frac{m-2}{m-1} f(B_{m^*}) > 1 - \frac{OPT}{m-1},$$

$$f(B_{m^*}) > \frac{m-1-OPT}{m-2},$$

又因为

$$OPT \geq \left[\sum_{i=1}^{n} s(a_i)\right] \geq$$

$$\sum_{i=1}^{n} s(a_i) = \sum_{j=1}^{m} f(B_j),$$

所以

$$OPT \geq \sum_{\substack{j=1 \\ j \neq m^*}}^{m} f(B_j) + f(B_{m^*}) >$$

$$\frac{2}{3}(m-1) + f(B_{m^*}) >$$

$$\frac{2}{3}(m-1) + \frac{m-1}{m-2} - \frac{OPT}{m-2},$$

即

$$OPT > \frac{2(m-2)}{3} + 1,$$

$$3OPT > 2m - 1,$$

由 m 和 OPT 的整数性知

$$3OPT \geq 2m,$$

即

$$\frac{m}{OPT} \leq \frac{3}{2}.$$

引理 3　CF 算法得到的 m 个箱子中若存在 m_0 个箱子($2 \leq m_0 \leq m$),每个箱子中只有一个物件,则必存在一种最优方案使得最优解 OPT 中包含 m_0 个箱子,每个箱子中只装有一个物件且与由算法 CF 得到的 m_0 个箱子中所装物件相同.

证明　当 $m_0 = 1$ 时,知另外 $m-1$ 个箱子中每个箱子所装物件数目至少为 2,此时由引理 2 即知 CF 算法可获得装箱问题的$\frac{3}{2}$ 的近似值.

当 $2 \leq m_0 \leq m$ 时,不妨设这 m_0 个只装有一个物件的箱子,按开启箱子的顺序依次为 $B_1^1, B_2^1, \cdots, B_{m_0}^1$. 设存在一种最优方案 I^*,其最优值为 OPT,由算法得到的一个方案为 I.

若 $B_1^1, B_2^1, \cdots, B_l^1, 1 \leq l \leq m_0$,为按算法开启的前 l 个箱子 $B_1, B_2, \cdots, B_l$,易知在 I^* 中一定存在 l 个箱子,里面只装有一个物件且与 $B_1^1, B_2^1, \cdots, B_l^1$ 中所装的物件相同,当 $l = m_0$ 时引理得证.

若 $B_1^1 \neq B_1$,设 B_k^1 为这 m_0 个只装有一个物件的箱子中的任意一个箱子,且 B_k^1 是 I 中按算法开启的所有箱子中的第 j 个箱子,$1 < j \leq m$,即 $B_k^1 = B_j$,由假设及算法知 B_k^1 内只装有一个物件 a_j.

假设在 I^* 中,a_j 所在的箱子 B_j^* 中还存在其它物件,不妨设这些物件的集合为 $L(a_j)$,那么易知 $L(a_j)$ 中的物件只能是 I 中已经装入 B_i, $i < j$, $L(a_j) \subset \{a_{h+1}, \cdots, a_n\}$ 中且不是 a_i 的那些物件,即如果 a_h 为按算法第 1 个不能放入 B_k^1 的物件,则 $L(a_j) \subset \{a_{h+1}, \cdots, a_n\}$. 于是 $L(a_j)$ 中的物件必然能够全部放入 I^* 中装有 a_i 的那些箱子 B_i^* 的某些箱子中. 于是总可以把 I^* 转化为另一个最优方案 I', I' 中有一个箱子 B_k^* 里面只装有 a_j,且 OPT 的数目不改变.

由 B_k^1 的任意性知引理成立.

引理 4　由 CF 算法得到的 m 个箱子中若存在 m_0 个箱子($2 \leq m_0 \leq m$),每个箱子中只装有一个物件,则 CF 算法是一维装箱问题的一个$\frac{3}{2}$ - 近似算法.

证明　由引理 3 知问题本身的最优值 OPT 中也存在 m_0 个箱子,这 m_0 个箱子中所装的物件与 CF 算法得到的 m_0 个箱子中所装物件相同,则 $OPT = m_0 + OPT'$, $m = m_0 + m'$,由于 m' 个箱子中所装物件数目至少为 2,由引理 2 知

$$\frac{m'}{OPT'} \leq \frac{3}{2},$$

即 $3OPT' \geq 2m'$,则

$$\frac{m}{OPT} = \frac{2(m_0 + m')}{2(m_0 + OPT')} \leq \frac{2m_0 + 3OPT'}{2(m_0 + OPT')} \leq \frac{3(m_0 + OPT')}{2(m_0 + OPT')} = \frac{3}{2},$$

所以

$$\frac{m}{OPT} \leq \frac{3}{2}.$$

定理 2　CF 算法是一维装箱问题的一个$\frac{3}{2}$ - 近似算法. 并且 CF 算法的时间复杂性为 $O(n\log n)$.

证明　由引理 1～4 知 CF 算法是一维装箱问题的一个$\frac{3}{2}$ - 近似算法.

下面证明 CF 算法的时间复杂性为 $O(n\log n)$.

由于在初始化步骤(step 1)中,由文献[4]得到排序所用时间为 $O(n\log n)$. 在迭代步骤中,第 1 次循环所用时间为 $O(\sigma(B_1))$,第 2 次循环所用时间为 $O(\sigma(B_2))$, $\cdots$, 第 m 次循环所用时间为 $O(\sigma(B_m))$,而 $\sigma(B_1) + \sigma(B_2) + \cdots + \sigma(B_m) = n$.

从而迭代步骤所需时间为 $O(n)$. 故 CF 算法的时间复杂性为 $O(n\log n)$. 如果物件按非增性质先排好序,CF 算法的时间复杂性为 $O(n)$.

3 结 论

本文对经典的一维装箱问题给出了一个新的近似算法,利用交叉装填的思想,获得了问题的最优近似值 $\frac{3}{2}$. 与目前已知的求解装箱问题的其它近似算法相比,近似值和复杂性从总体上而言都有了较大的提高.

此外,现在人们更多的关注于由一维经典装箱问题所衍生出来的一些装箱问题,如最大基数装箱问题[5],箱子覆盖问题[6],有可变容量的装箱问题[7]等. 这些问题一方面与一维装箱问题有着密切联系,另一方面又有着更广泛的实际应用背景. 我们未来的工作将继续关注并研究一维装箱问题以及装箱问题的各种衍生问题,以期能对实际生活工作提供更大的帮助.

参考文献:

[1] ARP R M. Reducibility among combinatorial problems [A]. Complexity of Computations [C], New York: Plenum, 1972. 85—103.

[2] COFFMAN JR E G, GARAY M R, JOHNSON D S. Approximation algorithms for bin packing: A survery [A]. Approximation Algorithms For NP-hard Problems [C]. Boston: PWS Publishers, 1996. 46—93.

[3] VAZIRANI Vijay V. Approximation algorithms [M]. Hong Kong: Springer, 2001.

[4] KNUTH D E. The art of computer programming: sorting and search [M]. Boston: Addison-Wesley, 1998.

[5] LABBé M, LAPORTE G, MARTELLO S. Upper bounds algorithms for themaximum cardinality bin packing problem [J]. European Journal of Operational Research, 2003, 149: 489—490.

[6] JANSEN K, SOLIS-OBA R. An asymptotic fully polynomial time approximation schemefor bin covering [J]. Theoretical Computer Science, 2003, 306: 543—551.

[7] KANG J, PARK S. Algorithms for the variable sized bin packing problem [J]. European Journal of Operational Research, 2003, 147: 365—372.

A new approximation algorithm for Bin-Packing problem

SUN Chun-ling, CHEN Zhi-bin, LI Jian-ping
(Department of Mathematics, Yunnan University, Kunming 650091, China)

Abstract: The Bin-Packing problem is studied again and derived a new $\frac{3}{2}$ - approximation algorithm in $O(n\log n)$ steps; In addition, if the sizes of all objects decreasing according to their sizes, our algorithm runs in $O(n)$ steps.

Key words: Bin-Packing problem; NP-completeness; approximation algorithm