arXiv:2304.02498v1 [cs.DS] 5 Apr 2023

最小加权和装箱问题两种算法的改进分析

Guillaume Sagnol[0000−0001−6910−8907]

柏林工业大学，数学研究所，17. Juni 大街 136 号，10623 柏林，德国
sagnol@math.tu-berlin.de

摘要。 我们研究最小加权和装箱问题，这是经典装箱问题的一种变体：物品具有重量，每件物品的代价等于其重量乘以所装入箱子的序号。事实上，这等价于一种批次调度问题，广泛见于预约调度、仓储物流等应用领域。我们改进了该问题两种简单算法的近似比下界和上界。具体而言，我们证明，背包分批算法的近似比至多为 17/10。该算法通过迭代求解背包问题，在剩余待装物品中选取重量总和最大的物品装入当前箱子。

关键词： 装箱问题· 批次调度 · 近似算法

1 引言

装箱问题是计算机科学中的一个基础问题：给定一组物品，需将它们装入尽可能少的相同箱中。它在物流、数据存储和云计算等诸多领域都有应用。装箱问题目标的一个特点是，所有箱都被视为“同样好”，但在涉及时间因素的应用中，这一点并不总是成立。例如，考虑将一组 n 患者分配到不同日期，以便预约医生就诊。每位患者都有相应的服务时间和重量，后者表示其健康状况的严重程度。某一天所分配患者的总检查时间不得超过固定阈值。因此，各天对应于按 1、2、.....编号的箱；优先安排编号较小的箱，尤其是对重量较大的患者而言，因为这样可以缩短患者的等待时间。

在最小加权和装箱问题（MWSBP）中，该问题在 [5], 中被正式提出，输入由一组 n 大小为 $s_i \in (0, 1]$ 、重量为 $w_i > 0$[1] 的物品组成。目标是将这组物品以可行的方式分配到箱中，使代价最小；也就是说，将 $[n] := \{1, \ldots, n\}$ 划分为 B_1、 .. 、 B_p，使得对于所有 $k \in [p]$， $\sum_{i \in B_k} s_i \leq 1$ 都成立。其中，将物品 i 放入 B_k 的代价为 $k \cdot w_i$。换言之，如果我们使用以下记法

[1] 在 [5] 中，假设重量为 $w_i \geq 1$，但通过缩放可将问题约化到这种情形；这一操作不会影响我们对近似比所得的下界。

$x(S) := \sum_{i\in S} x_i$ 对于向量 $x \in \mathbb{R}^n$ 和子集 $S \subseteq [n]$，可行分配的代价为

$$\Phi(B_1, \ldots, B_p) := \sum_{k=1}^{p} k \cdot w(B_k) = \sum_{k=1}^{p}\sum_{j=k}^{p} w(B_j). \tag{1}$$

对这个问题的另一种解释是：有一组处理时间均为 1 的作业，我们要将它们安排在一台批处理机器上。该机器能够同时处理一批总大小不超过 1 的作业，目标是最小化加权完成时间和。按照 Graham 等人引入的三字段表示法，[9]，MWSBP 因此等价于 $1|p\text{-}\,batch,\ s_j,\ p_j = 1|\sum w_j C_j$；我们参考 [7]，其中提供了近期综述，并介绍了更多关于并行批次调度的背景。总的来说，MWSBP 涵盖了许多现实问题中的核心挑战：物品必须随时间进行装箱，例如上文提到的预约调度问题，或订单拣选仓库中的批次调度问题。

对于给定的 MWSBP 实例和算法 ALG，我们用 OPT 表示最优解的代价，而算法 ALG 返回的解的代价也记作 ALG。回顾一下，算法 ALG 的近似比 $\mathcal{R}$(ALG) 是满足以下条件的最小常数 $\rho \geq 1$：对于该问题的所有实例，均有 ALG $\leq \rho \cdot$ OPT。

相关工作。MWSBP 的复杂度已得到充分研究：该问题在强意义下是 NP-hard 的，并且存在一个多项式时间近似方案（PTAS）[5]。本文还给出了一种基于 Next-Fit 的简单算法，其近似比为 2。在此之前，已有若干启发式算法被提出，用于处理该问题的一种推广形式：其中，不同的批次时间对应着不同且彼此不兼容的物品族 [1], ；研究还表明，First-Fit 的一个变体同样具有 2 的近似比。

该问题的无权版本——最小和装箱（MSBP）——也受到了一些关注，其中每件物品的重量均为 $w_i = 1$。[4] 针对这一特殊情形描述了一种更简单的 PTAS。作者还分析了若干基于 First-Fit 或 Next-Fit 的算法的渐近近似比，即将注意力限制在 OPT $\to \infty$ 的实例上时，近似比的极限。具体而言，他们给出了一种渐近近似比至多为 1.5604 的算法。此外，已知 First-Fit 的一种变体按物品尺寸非递减的顺序处理物品，是 MSBP 的一个 $(1+\sqrt{2}/2)$-近似算法 [10]。

–

另一个相关问题是最小加权和集合覆盖（MWSSC）：输入给定一个由物品组成的全集 $[n]$ 的若干子集 S_1、……、S_m，以及相应的重量 w_1、……、$w_n > 0$。与 MWSBP 中一样，解是对 B_1、B_2、……、B_p 的一个划分，其批次代价由式（1）给出；但不同之处在于，每个批次 $B_k \subseteq [n]$ 必须是某个 S_j、$j \in [m]$ 的子集。换言之，与 MWSBP 的区别在于，可行批次是通过一组极大批次显式描述的，而不是隐式地由背包约束 $s(B_k) \leq 1$ 描述的。因此，任何 MWSBP 实例都可以转化为一个 MWSSC 实例，不过这需要规模为

表 1. MWSBP 简单算法近似比的新旧界汇总。本文的结果以粗体标示，并注明证明这些结果的命题（Pr.）或定理（Th.）编号。

算法	下界		上界		特殊情况下的上界
	先前	new	先前	new	MWSBP 的情形
WNFI	2 [1]		2 [1]		1.618（ $w_i = 1$, 最优值 $\to \infty$ [4]
WFFI	2 [1]		2 [1]		1.707（ $w_i = 1$ [10]
WNFD-R	2 [5]		2 [5]		
WFFI-R	1.354 [4]	**1.557**（命题 6）	2 [1]		**1.636** 对于 $w_i = s_i$（定理 7）
KB	1.354 [4]	**1.433**（命题 5）	4 [6]	**1.7**（定理 3）	

指数级规模（枚举所有总大小至多为 1 的物品极大子集）的输入。MWSSC 的无权版本由 [6] 提出。作者证明，一种自然的贪心算法是 4-近似算法；除非 $P = NP$，否则这一性能保证已是最佳可能。

贡献与结构安排。 鉴于 MWSBP 在实际应用中的重要性，我们认为，即使已经存在多项式时间近似方案，了解该问题的简单算法的性能仍然十分重要。事实上， [5] 的多项式时间近似方案运行时间极长，无法用于实际应用。本文改进了 MWSBP 两种简单算法的近似比率下界和上界；特别是，我们首次获得了近似比严格低于 2 的简单算法，此前结果与本文的新结果总结见表 1。

本文介绍了两种经过分析的算法：背包分批算法（KB）和加权重排递增首次适配（WFFI-R），并在第2节介绍这些算法，同时补充装箱问题和 MWSBP 算法的背景知识。在第3节中，我们证明 $\mathcal{R}$(KB) $\in (1.433,\ 1.7]$；在第4节中，我们证明$\mathcal{R}$(WFFI-R) > 1.557。此外，所有已知的最佳下界均在所有物品都满足 $w_i = s_i$ 的实例上达到。这种情况让人联想到最小化加权完成时间和的调度问题，其中最坏实例中的作业具有相同的史密斯比；例如参见 [11]。因此，研究这一情形是很自然的；我们证明，对于所有 $i \in I$ 均满足 $w_i = s_i$ 的实例，WFFI-R 是一个$(7+\sqrt{37})/8$-近似算法。

2 预备知识

针对装箱问题，人们提出了许多广为人知的启发式算法。例如，下一适配算法（NF）和首次适配算法（FF）按照任意给定的顺序考察物品列表，并依次将它们分配到箱中。这两种算法选择当前物品放入哪个箱的方式不同。具体而言，NF 始终跟踪一个已开启的箱；如果当前物品能放入该箱，就将其放入其中，否则开启另一个箱并将物品放入其中。相比之下，FF 会先检查当前物品能否放入迄今使用过的任一箱中。如果可以，就将物品放入能容纳它且最早开启的箱中；否则开启一个新箱。众所周知，FF 装入所有物品所需的箱数至多为 $\lfloor 17/10 \cdot OPT \rfloor$ 箱，其中 OPT表示所需箱数的最小值

这一结果由 Dósa 和 Sgall [2] 在一系列改进性能保证中加法项的论文之后得出。

在 MWSBP 中，物品不仅有大小，还有重量。因此，自然会考虑 FF 和 NF 的加权变体 WFFI、WFFD、WNFI、WNFI，其中 W 代表 加权，字母 I（分别地，D）代表 递增（分别地，递减），表示按比率 s_i/w_i 对物品进行非递减（分别地，非递增）排序。利用简单的交换论证可以看出，每种算法都可以通过将其输出的箱按重量非递增顺序重排而得到改进。我们在这些算法的缩写后添加后缀 -R，以表示所得算法。尽管 WNFD、WFFD 和 WFFD-R 的近似比没有界，即使物品的重量均为单位重量 [4], ，文献 [1] 证明 WNFI 和 WFFI 的近似比恰为 2，WNFD-R 也同样如此 [5]。本文在第4节给出了 WFFI-R 更优的界。

MWSBP 的另一种自然算法是背包分批（KB）算法，该算法由 [1] 提出，可描述如下：在第 $k = 1$、2、..... 次迭代中，求解一个背包问题，找出剩余物品 $B_k \subseteq [n] \setminus (B_1 \cup \ldots \cup B_{k-1})$ 的一个子集，使其重量最大且能装入一个箱（即，$s(B_k) \leq 1$）。事实上， [5] 认为，KB 算法是引言中提到的最小和集合覆盖问题贪心算法的直接转换，因此其近似比至多为 4。实际应用中，可以在每次迭代时使用一个全多项式时间近似方案 （FPTAS），在多项式时间内求得背包问题的近似最优解，从而得到一个 $(4+\varepsilon)$-近似算法来求解 MWSBP。在下一节中，我们将证明 KB 算法的一个改进上界 1.7（如果使用 FPTAS 在多项式时间内求解背包问题，则为 $1.7+\varepsilon$ ）。

3 背包分批算法

本节研究背包分批（KB）算法。在本节中，对于 MWSBP 的任意实例，我们用 $B_1, \ldots, B_p$ 表示 KB 返回的箱集合，用 $O_1, \ldots, O_q$ 表示最优箱集合。为方便记号，我们还对所有 $p' \geq p+1$定义 $B_{p'} = \emptyset$，并对所有 $q' \geq q+1$定义 $O_{q'} = \emptyset$。回顾一下，KB 在每次迭代中都会针对剩余物品子集求解一个背包问题，因此，对于所有 k，对所有满足 $s(B) \leq 1$ 的 $B \subseteq [n] \setminus (B_1 \cup \ldots \cup B_{k-1})$，均有 $w(B_k) \geq w(B)$。

我们首先证明以下命题：只要证明 KB 在前 αk 个箱中装入的重量至少与 OPT 在仅有 k 个箱中装入的重量一样多，就能证明其性能保证为 α 。证明的关键是将 OPT 和 KB 表示为某条曲线下方的面积。接着我们证明，将对应 KB 的曲线缩小 α 倍后，可得到 OPT 曲线的一个下估计；见图 1。 [6] 中也采用了类似思路来界定贪心算法用于最小和集合覆盖时的近似比，但其界为 4，这是因为该方法沿 x 轴和 y 轴均将曲线缩小了 2 倍。

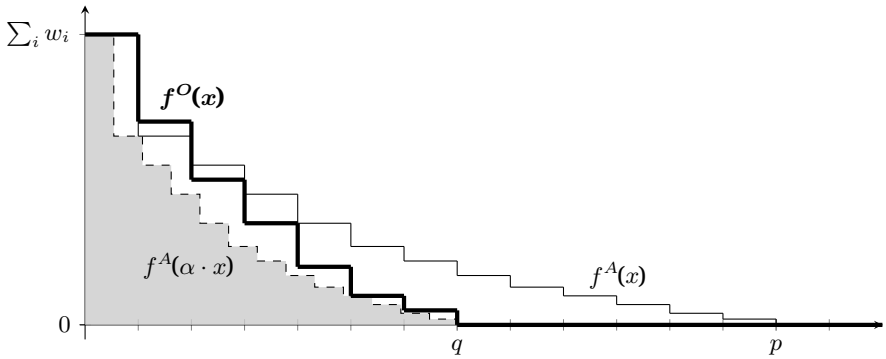

图 1. 命题 1 的证明示意图。$x \mapsto f^O(x)$ 的粗曲线下方区域为最优值，$x \mapsto f^A(x)$ 的细曲线下方区域为背包分批算法。将该区域在水平方向缩小 α 倍后，得到 $x \mapsto f^A(\alpha \cdot x)$ 下方的阴影区域，该区域必小于最优值。

命题 1 设 $\alpha \geq 1$。若对所有 $k \in [q]$ 均有

$$w(B_1) + w(B_2) + \ldots + w(B_{\lfloor \alpha k \rfloor}) \geq w(O_1) + w(O_2) + \ldots + w(O_k),$$

那么背包分批算法 $\leq \alpha$ ·最优值。

证明。对于所有 $x \geq 0$，令 $f^O(x) := \sum_{j=\lfloor x \rfloor+1}^{\infty} w(O_j)$。注意，$f^O$ 是分段常数函数，并且满足：对于所有 $x \in [k-1, k)$ $k \in [q]$，$f^O(x) = \sum_{j=k}^{q} w(O_j)$；对于所有 $x \geq q$，$f^O(x) = 0$。因此，利用式 (1) 中的第二个表达式，我们可以将最优值表示为曲线 $f^O(x)$ 下方的面积：

$$\texttt{OPT} = \sum_{k=1}^{q} \sum_{j=k}^{q} w(O_j) = \int_0^{\infty} f^O(x)\, dx.$$

类似地，我们有背包分批算法 $= \int_0^{\infty} f^A($ x $=$ $\sum_{j=\lfloor x \rfloor+1}^{\infty} w$ B_j f^A $\alpha \cdot k$ $\leq$ $)\, dx$，其中对于所有 $x \geq 0$，定义 $f^A( ) : ( )$。引理的假设可以改写为：对于所有 $k \in [q]$，$( ) f^O(k)$；并且我们注意到，当 $k = 0$ 时，这个不等式也成立，因为 $f^A(0) = f^O(0) = \sum_{i=1}^{n} w_i$。

现在，我们论证这意味着 $f^A(\alpha \cdot x) \leq f^O(x)$ 对所有 $x \geq 0$ 都成立。如果 x 落在对于某个 $k \in \{0, \ldots, q-1\}$ 而言形如 $x \in [k, k+1)$ 的区间内，那么我们有 $f^O(x) = f^O(k) \geq f^A(\alpha \cdot k) \geq f^A(\alpha \cdot x)$，其中最后一个不等式由 $k \leq x$ 以及 f^A 是非递增的这一事实得出。否则，我们有 $x \geq q$，因此有 $f^O(x) = 0 = f^O(q) \geq f^A(\alpha \cdot q) \geq f^A(\alpha \cdot x)$；有关示意图请参见图 1。现在我们可以结束该证明：

$$\texttt{KB} = \int_0^{\infty} f^A(x)\, dx = \alpha \cdot \int_0^{\infty} f^A(\alpha \cdot y)\, dy \leq \alpha \cdot \int_0^{\infty} f^O(y)\, dy = \alpha \cdot \texttt{OPT}.$$

■

接下来，我们证明，对于 $\alpha = 17/10$，KB 满足命题 1 的假设。这个结果本身也很有意义，因为它意味着，对于给定重量下求装箱所需最少箱数的问题，KB 也是一种 $17/10$-近似算法。

命题 2 对于所有满足以下条件的 *MWSBP* 实例：箱 $O_1, \ldots, O_q$ 为最优解，且 *KB* 输出箱 $B_1, \ldots, B_p$；并且对于所有 $k \in [q]$，均有

$$w(B_1) + w(B_2) + \ldots + w(B_{\lfloor \frac{17}{10} k \rfloor}) \geq w(O_1) + w(O_2) + \ldots + w(O_k).$$

证明。令 $k \in [q]$。我们定义集合 $B := B_1 \cup \ldots \cup B_{\lfloor 1.7k \rfloor}$, $O := O_1 \cup \ldots \cup O_k$。对于所有 $i \in B$，用 $\beta(i)$ 表示 $\in [1.7\,k]$ 包含 i 的背包分批算法箱的编号，即 $i \in B_{\beta(i)}$。现在，我们按照以下方式对 O 中的物品排序：首先按 $\beta(i)$ 非递减的顺序排列 $O \cap B$ 中的物品，然后将 $O \setminus B$ 中所有剩余物品按任意顺序排列。接下来，对按上述顺序排列的 O 中的物品应用首次适配算法，从而构造一个新的装箱 $H_1, \ldots, H_{q'}$，其中包含 O 中的物品。对于所有 $i \in O \cap B$，令 $\beta'(i)$ 表示满足 $i \in H_{\beta'(i)}$ 的编号。显然，我们对 O 采用的排序方式意味着，对于所有 $i \in O \cap B$，$\beta'(i) \leq \beta(i)$。

由 [2] 可得 $q' \leq \lfloor 1.7 \cdot k \rfloor$。因此，我们定义 $H_j := \emptyset$ 对于 $j = q' + 1,\ q' + 2,\ \ldots\ldots,\ \lfloor 1.7 \cdot k \rfloor$，且 $O = H_1 \cup \ldots \cup H_{\lfloor 1.7k \rfloor}$ 成立。现在，我们将证明对于所有 $j = 1,\ \ldots\ldots,\ \lfloor 1.7k \rfloor$，$w(H_j) \leq w(B_j)$ 均成立。为此，利用背包分批算法的贪心性质，只需证明在求解第 j 次迭代中的背包问题时，H_j 的所有元素都仍然保留，即 $H_j \subseteq [n] \setminus (B_1 \cup \ldots \cup B_{j-1})$。因此，设 $i \in H_j$。若 $i \notin B$，则 $i \notin (B_1 \cup \ldots \cup B_{j-1})$ 显然成立。否则，$i \in O \cap B$，因此我们有 $j = \beta'(i) \leq \beta(i)$，这意味着 i 不属于任何满足 $\ell < j$ 的 B_ℓ。现在可以得出证明：

$$w(O) = \sum_{i=1}^{\lfloor 1.7 \cdot k \rfloor} w(H_i) \leq \sum_{i=1}^{\lfloor 1.7 \cdot k \rfloor} w(B_i) = w(B).$$

■

命题 1 和命题 2 得出本节的主要结果：

定理 3 $\mathcal{R}(KB) \leq \frac{17}{10}$。—

备注 很容易调整上述证明，以说明：如果在 *KB* 算法的每次迭代中，用 *FPTAS* 求解背包问题，获得 $(1 + \frac{-\varepsilon}{n})$-近似解，以替代最优背包，那么我们就能得到一个用于 *MWSBP* 的多项式时间 $(\frac{17}{10} + \varepsilon)$-近似算法。

接下来我们给出 KB 的近似比下界。对于某些整数 s 和 k，满足 $s \geq 2^k$，令 $\epsilon = \frac{1}{2^k \cdot s}$。给定一个整数向量 $\boldsymbol{n} \in \mathbb{Z}_{\geq 0}^{k+1}$，我们构造如下实例：物品被划分为 $k+1$ 个类别，即 $[n] = C_1 \cup C_2 \cup \ldots \cup C_{k+1}$。对于所有 $j \in [k]$，类别 C_j 由 $N_j = n_1 + \ldots + n_{k+2-j}$ 个类别 j 中的物品组成，其中 $s_i = w_i = \frac{1}{2^j} + \epsilon,\ \ \forall i \in C_j$。此外，类别 C_{k+1} 包含 $N_{k+1} := n_1 \cdot (s - k)$ 个满足 $s_i = w_i = \epsilon$ 的微小物品。我们还假设 $m_1 := n_1 \cdot (2^{-k} - k\epsilon) = \frac{n_1}{(s-k)\, 2^k \cdot s}$ 是整数，并且对于所有 $j \in [k]$，$\frac{N_j}{2^j - 1}$ 是整数。然后，对于 $j = 2, \ldots, k+1$，令 $m_j := \frac{N_{k+2-j}}{2^{k+2-j} - 1} \in \mathbb{Z}$。

在这个实例中，背包分批算法可以先装入总重量为 1 的 m_1 个箱，每箱包含 $2^k \cdot s$ 个微小物品。之后，只剩下类别 C_1、……、C_k 中的物品，而背包问题的解是将类别 C_k 的 $2^k - 1$ 个物品放入一个箱中。因此，背包分批算法将 $m_2 = N_k/(2^k - 1)$ 个此类箱加入解中，每个箱的重量为 $(2^k - 1) \cdot (2^{-k} + \epsilon) = 1 - 2^{-k} + (2^k - 1) \cdot \epsilon$。继续

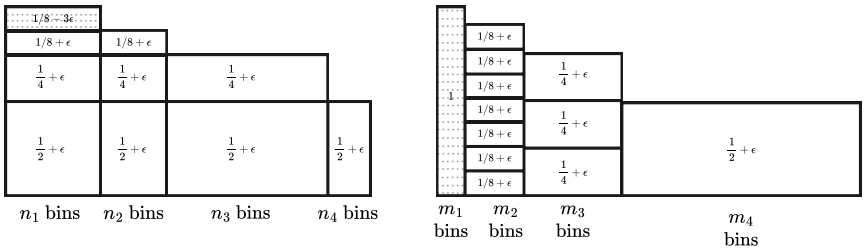

图 2. $k=3$ 对应的"良好"装箱示意图（左）以及 KB 返回的装箱（右）。该实例在引理 4 之前定义。虚线方块表示微小物品组。选择 m_i，使得两种装箱中每个类别的物品数量相同，即 $m_1 = n_1 \cdot (1/8 - 3\epsilon)$，$m_2 = (n_1+n_2)/7$，$m_3 = (n_1+n_2+n_3)/3$ 和 $m_4 = n_1+n_2+n_3+n_4$。

这一推理，我们可以看到，对于每个 $j=2$、.....、$k+1$，当只剩下类别 C_1、.....、C_{k+2-j} 中的物品时，背包分批算法会添加一组 m_j 个箱，其中包含 $(2^{k+2-j}-1)$ 个类别 C_j 的物品，每个箱的重量为 $1-2^{-(k+2-j)}+(2^{k+2-j}-1)\cdot\epsilon$。由此可得：

$$\texttt{KB} = \sum_{i=1}^{m_1} i + \sum_{j=2}^{k+1} \left(1 - 2^{-(k+2-j)} + (2^{k+2-j}-1)\cdot\epsilon\right) \cdot \sum_{i=m_1+\ldots+m_{j-1}+1}^{m_1+\ldots+m_j} i. \quad (2)$$

另一方面，我们可以为该实例构造如下装箱：前 n_1 个箱中，每个箱包含类别 $C_1,\ldots,C_k$ 的各一个物品，另加 $(s-k)$ 个微小物品；因此它们的总重量为 $\sum_{j=1}^{k}(2^{-j}+\epsilon) + (s-k)\epsilon = 1-2^{-k}+s\cdot\epsilon = 1$。然后，对于每个 $j=2,\ldots,k+1$，我们添加一组 n_j 个箱，每个箱中包含类别 $C_1, C_2, \ldots, C_{k+2-j}$ 的各一个物品。因此，第 j 组中的箱所含重量为 $\sum_{i=1}^{k+2-j}(2^{-i}+\epsilon) = 1-2^{-(k+2-j)} + (k+2-j)\cdot\epsilon$。显然，该装箱的代价是最优值的一个上界：

$$\texttt{OPT} \le \sum_{i=1}^{n_1} i + \sum_{j=2}^{k+1} \left(1 - 2^{-(k+2-j)} + (k+2-j)\cdot\epsilon\right) \cdot \sum_{i=n_1+\ldots+n_{j-1}+1}^{n_1+\ldots+n_j} i. \quad (3)$$

$k=3$ 的这两种装箱示意图如图2所示。

表达式 (2) 和 (3) 不适合用来求出使 $\mathcal{R}$（背包分批算法）的下界达到最大的 n_i 取值。下一个引理将说明，我们可以转而最大化两个二次函数之比，从而使问题更容易处理。

引理 4 设 $\boldsymbol{L}$ 为一个$(k+1)\times(k+1)$ 下三角矩阵，其下三角部分的所有元素均为 1，$\boldsymbol{u} := [\frac{1}{2^{k+1}}, \frac{1}{2^{k+1}}, \frac{1}{2^k}, \ldots, \frac{1}{2^2}] \in \mathbb{R}^{k+1}$，$\boldsymbol{v}$：$= [\frac{1}{2^k}, \frac{1}{2^k-1}, \frac{1}{2^{k-1}-1}, \cdots, \frac{1}{2^2-1}, \frac{1}{2-1}] \in \mathbb{R}^{k+1}$，并令 $\boldsymbol{U}$:=Diag($\boldsymbol{u}$) 以及 $\boldsymbol{V}$:=Diag($\boldsymbol{v}$)。那么，对所有 $\boldsymbol{x} \in \mathbb{R}^{k+1}_{>0}$，均有

$$\mathcal{R}(\mathit{KB}) \ge R(\boldsymbol{x}) := \frac{\boldsymbol{x}^T\boldsymbol{L}^T\boldsymbol{V}\boldsymbol{L}^T\boldsymbol{U}\boldsymbol{L}\boldsymbol{V}\boldsymbol{L}\boldsymbol{x}}{\boldsymbol{x}\boldsymbol{L}^T\boldsymbol{U}\boldsymbol{L}\boldsymbol{x}}.$$

证明。设 $\boldsymbol{w} \in \mathbb{R}^{k+1}$ 和 $\boldsymbol{a} \in \mathbb{N}^{k+1}$ 满足 $a_1 < a_2 < \ldots < a_{k+1}$，为方便记号，令 $a_0 := 0$，并定义

$$F(\boldsymbol{a}) := \sum_{j=1}^{k+1} w_j \sum_{i=a_{j-1}+1}^{a_j} i = \sum_{j=1}^{k+1} \frac{w_j}{2}(a_j^2 - a_{j-1}^2 + a_j - a_{j-1}) = \boldsymbol{a}^T \operatorname{Diag}(\delta(\boldsymbol{w}))\boldsymbol{a} + \boldsymbol{a}^T \delta(\boldsymbol{w}),$$

其中 $\delta(\boldsymbol{w}) := \frac{1}{2}[w_1 - w_2, w_2 - w_3, \ldots, w_k - w_{k+1}, w_{k+1}]^T \in \mathbb{R}^{k+1}$。现在，假设对于某个任意矩阵 M，$\boldsymbol{a} = M\lfloor t\boldsymbol{x} \rfloor$ $\boldsymbol{x} \in \mathbb{R}_{>0}^{k+1}$ 和 $t > 0$，即对于某个 $\boldsymbol{b} \in \mathbb{N}^{k+1}$，使得 $b_i = \lfloor t x_i \rfloor$，对所有 $i \in [k+1]$，都有 $\boldsymbol{a} = M\boldsymbol{b}$。利用 $\boldsymbol{a} = t \cdot (M\boldsymbol{x} + o_{t\to\infty}(1))$ 这一事实，以及 F 是二次函数这一事实，忽略线性项后可得到渐近近似：

$$F(M\lfloor t\boldsymbol{x} \rfloor) = t^2 \cdot \left(\boldsymbol{x}^T M^T \operatorname{Diag}\left(\delta(\boldsymbol{w})\right) M\boldsymbol{x} + o_{t\to\infty}(1)\right).$$

现在，令 $\boldsymbol{n} = \alpha\lfloor t\boldsymbol{x} \rfloor$，其中 $\boldsymbol{x} \in \mathbb{R}_{>0}^{k+1}$ 是某个任意的；α 满足 m_1，m_2，……，m_{k+1} 均为整数，例如 $\alpha := s \cdot 2^k \cdot \prod_{j=1}^{k}(2^j - 1)$。我们注意到，式 (3) 中 OPT 的上界具有 $F(\boldsymbol{a})$ 的形式，其中 $\boldsymbol{a} = [n_1, n_1 + n_2, \ldots\ldots, n_1 + \ldots + n_k]^T = L\boldsymbol{n}$ 和 $\boldsymbol{w} = [1, 1 - 2^{-k} + k\epsilon, \ldots, 1 - 2^{-1} + \epsilon]^T$，且 $\delta(\boldsymbol{w}) \to \boldsymbol{u}$ 与 $\epsilon \to 0$ 一样。因此，我们证明了 OPT $\leq t^2(\alpha^2 \boldsymbol{x}^T \boldsymbol{L}^T \boldsymbol{D}_\epsilon \boldsymbol{L}\boldsymbol{x} + o_{t\to\infty}(1))$，其中 $\boldsymbol{D}_\epsilon :=$ Diag$(\delta(\boldsymbol{w}))$ 随着 $\epsilon \to 0$ 的变化收敛到 $\boldsymbol{U}$。

类似地，(2) 也具有 $F(\boldsymbol{a})$ 的形式，但将 $\boldsymbol{a} = L\boldsymbol{m}$ 和重量 $\boldsymbol{w}$ 替换为 $\boldsymbol{w}'$：$= [1, 1 - 2^{-k} + (2^k - 1)\epsilon, \ldots\ldots, 1 - 2^{-1} + (2^{-1} - 1)\epsilon]$。使用 $\boldsymbol{m} \to \boldsymbol{V}\boldsymbol{L}\boldsymbol{n}$ 和 $\delta(\boldsymbol{w}') \to \boldsymbol{u}$ 作为 $\epsilon \to 0$，可得渐近近似 背包分批算法 $= t^2(\alpha^2 \boldsymbol{x}^T \boldsymbol{L}^T \boldsymbol{V} \boldsymbol{L}^T \boldsymbol{D}'_\epsilon \boldsymbol{L}\boldsymbol{V}\boldsymbol{L}\boldsymbol{x} + o_{t\to\infty}(1))$，其中 $\boldsymbol{D}'_\epsilon :=$ Diag$(\delta(\boldsymbol{w}'))$ 收敛到 $\boldsymbol{U}$，当 $\epsilon \to 0$ 时。这意味着存在一个实例序列（满足 $t \to \infty$ 和 $\epsilon \to 0$），使得 背包分批算法 /最优值收敛到大于 $R(\boldsymbol{x})$ 的值。■

命题 5 $\mathcal{R}$(背包分批算法) > 1.4334。

证明。这可由将引理 4 应用于向量 $\boldsymbol{x} = [0.97, 0.01, 0.01,$0。01，0。03，0。07，0。15，$0.38] \in \mathbb{R}_{>0}^8$ 得出。事实上，该向量是问题的最优解，即在 $\mathbb{R}^8$ 上最大化 $R(\boldsymbol{x})$ 的解；为便于阅读，数值保留两位小数。为求得该向量，我们将最大化 $R(\boldsymbol{x})$ 的问题化为特征值问题，并对相应的特征向量应用逆变换，恢复了 $\boldsymbol{x}$。请注意，与优化 $k = 7$ 相比，优化 $k = 50$ 仅使界提升了 $3 \cdot 10^{-5}$。■

4 重排首次适应递增算法

在本节中，我们分析 WFFI-R 算法。首先回顾一下，WFFI（不对箱进行重排）的近似比为 2。为了说明这个界是紧的，考虑这样一个包含 2 件物品的实例：$w_1 = s_1 = \epsilon$ 和 $w_2 = s_2 = 1$。显然，最优装箱会将大件物品放入第一个箱，将小件物品放入第二个箱，因此 OPT $= 1 + 2\epsilon$。另一方面，由于这两件物品的比率相同 s_i/w_i，WFFI 可能会将小件物品放入第一个箱，从而得到 WFFI $= \epsilon + 2$。因此，WFFI/OPT $\to 2$，因为 $\epsilon \to 0$。不过，在这个实例中，我们可以看到，按重量递减顺序重排箱即可解决这个问题。

不过，很容易找到一个使 WFFI-R/OPT 接近 $3/2$ 的实例：设 $w_1 = s_1 = \epsilon$ 和 $w_2 = w_3 = s_2 = s_3 = \frac{1}{2}$。此时，最优装箱会将物品 2 和 3 放入第一个箱，得到 OPT $= 1 + 2\epsilon$；而 WFFI-R 可能返回箱 $B_1 = \{1, 2\}$，$B_2 = \{3\}$，使得 WFFI-R $= \frac{1}{2} + \epsilon + 2 \cdot \frac{1}{2} = \frac{3}{2} + \epsilon$。

接下来，我们将说明如何得到更强的 $\mathcal{R}$(WFFI-R) 界。为此，我们回顾一下：对于所有足够小的 $k \in \mathbb{N}$ 和 $\epsilon > 0$，都存在一个具有以下性质的装箱问题实例 [8,2]：其中有 $n = 30k$ 个物品，可分为三类。物品 $i = 1, \ldots, 10k$ 是小物品，大小为 $s_i = 1/6 + \delta_i\epsilon$；此外，还有 $10k$ 个中等物品，大小为 $s_i = 1/3 + \delta_i\epsilon$ ($i = 10k + 1 \ldots, 20k$)，以及 $10k$ 个大物品，大小为 $s_i = 1/2 + \epsilon$ ($i = 20k + 1, \ldots, 30k$)。常数 $\delta_i \in \mathbb{R}$ 可以为正或负，并按如下方式选取：若 FF 按照 $\mathbf{1}, \ldots, n$ 的顺序处理这些物品，则会得到一种装箱方案 $B_1, \ldots B_{17k}$，其中前 $2k$ 个箱各装入 5 个小物品，接下来的 $5k$ 个箱各装入 2 个中等物品，最后的 $10k$ 个箱各装入 1 个大物品。另一方面，所有物品可以装入 $10k + 1$ 个箱中，其中 $10k - 1$ 个箱各由一个小物品、一个中等物品和一个大物品组成，大小为 $1 - O(\epsilon)$；其余两个箱的大小为 $1/2 + O(\epsilon)$（一个装有小物品和中等物品，另一个只装有一个大物品）。

我们可以将这个装箱问题实例转换为一个 MWSBP 实例：令所有 i 都满足 $w_i = s_i$。这确保所有物品具有相同的比率 s_i/w_i，因此我们可以假设 WFFI-R 按我们希望的任意顺序考虑这些物品。此外，我们考虑两个整数 u 和 v，并将中等物品的数量从 $10k$ 增加到 $10k + 2u$（因此中等物品为 $i = 10k + 1, \ldots, 20k + 2u$），并将大型物品的数量从 $10k$ 增加到 $10k + 2u + v$（因此大型物品为 $i = 20k + 2u + 1, \ldots, 30k + 4u + v$）。对于 $i = 1, \ldots 20k$，δ_i 保持不变；对于所有额外的中等物品（$i = 20k + 1, \ldots, 20k + 2u$），我们令 $\delta_i = 1$。接着，假设 WFFI-R 按 $1, 2, \ldots$ 的顺序考虑物品，该算法会装入 $2k$ 个箱，每箱有 5 个小型物品；装入 $5k + u$ 个箱，每箱有 2 个中等物品；最后 $10k + 2u + v$ 个箱各装入一个大型物品。另一方面，我们可以使用原实例的最优装箱，并将额外物品装入 $2u$ 个大小为 $5/6 + 2\epsilon$ 的箱中，每箱包含一个中等物品和一个大型物品；再装入 v 个箱，每箱只有一个大型物品。这总计为 $10k - 1$ 个大小为 $1 - O(\epsilon)$ 的箱、$2u$ 个大小为 $5/6 + O(\epsilon)$ 的箱，以及 $v + 2$ 个大小为 $1/2 + O(\epsilon)$ 的箱。当 $\epsilon \to 0$ 趋于极限时，我们得到

$$\begin{aligned}\texttt{WFFI-R} &= \frac{5}{6}\sum_{i=1}^{2k} i + \frac{2}{3}\sum_{i=2k+1}^{7k+u} i + \frac{1}{2}\sum_{i=7k+u+1}^{17k+3u+v} i \\ &= \frac{5}{6}k(2k+1) + \frac{1}{3}(5k+u)(9k+u+1) + \frac{1}{4}(10k+2u+v)(24k+4u+v+1)\end{aligned}$$

and

$$\begin{aligned}\texttt{OPT} &\geq \sum_{i=1}^{10k-1} i + \frac{5}{6}\sum_{i=10k}^{10k-1+2u} i + \frac{1}{2}\sum_{i=10k+2u}^{10k+2u+v+1} i \\ &= 5k(10k-1) + \frac{5}{6}u(20k+2u-1) + \frac{1}{4}(v+2)(20k+4u+v+1).\end{aligned}$$

命题 6 $\mathcal{R}(\textit{WFFI-R}) > 1.5576$。

证明。 将 $k = 10350, u = 11250, v = 24000$ 代入上述表达式即可得到该界。与上一节一样，这些值可以通过将求得 k, u, v 最优取值的问题化为一个特征值问题，并对所得特征向量进行放缩和舍入来获得。■

所有针对 WFFI-R（甚至针对表 1 中列出的所有算法）的糟糕示例都具有这样一个性质：所有物品的比率 s_i/w_i 都相同。从直觉上看，这很合理，因为算法不再能从对物品排序中获益，即物品可以按对算法不利的顺序提供。虽然目前已知的 WFFI-R 近似比上界只有 2（因为 WFFI-R 的表现只会优于 WFFI），但我们认为 $\mathcal{R}$(WFFI-R) 实际上更接近命题 6 中给出的下界。为支持这一观点，接下来我们将证明，对于满足 $s_i = w_i$ 的实例，上界约为 1.6354。

定理 7 对于所有满足 $s_i = w_i$ 且对所有 $i \in [n]$均成立的 *MWSBP* 实例，有

$$\textit{WFFI-R} \leq \frac{7+\sqrt{37}}{8} \cdot \textit{OPT}.$$

证明。 考虑一个满足 $w_i = s_i$ 且对所有 $i \in [n]$ 均成立的 MWSBP 实例，并将 WFFI-R 返回的箱 $B_1, \ldots, B_p$的重量（或等价地，大小）记为 $W_1 \geq \ldots \geq W_p$。我们先处理至多有一个箱的重量为 $\leq \frac{2}{3}$ 的情形，此时可得到如下界：3 2

断言 1 如果 $W_{p-1} > \frac{2}{3}$，那么 *WFFI-R* $\leq \frac{3}{2}$最优值。

该断言在附录中证明。因此，我们可以不失一般性地假设，至少有 2 个重量为 $\leq \frac{2}{3}$ 的箱。令 $r \in [p-1]$ 表示第一个满足 $W_r \leq 2/3$ 的箱的索引，并定义 $s := p - r \in [p-1]$。我们定义向量 $\boldsymbol{y} \in \mathbb{R}^r$ 和 $\boldsymbol{x} \in \mathbb{R}^s$，使得 $y_i = W_i$ $(\forall i \in [r])$ 且 $x_i = W_{r+i}$ $(\forall i \in [s])$。根据构造，有 $1 \geq y_1 \geq \ldots \geq y_{r-1} > 2/3 \geq y_r \geq x_1 \geq \ldots \geq x_s$。为简便起见，对于所有 $i > s$，我们还定义 $x_0 := y_r$ 和 $x_i := 0$。注意，对于所有 $i \neq j$，必须有 $W_i + W_j > 1$，否则首次适配算法会将箱 i 和 j 中的物品放入同一个箱中。这意味着 $x_{s-1} + x_s > 1$，因此 $x_{s-1} = \max(x_{s-1}, x_s) > 1/2$。使用上述记号，我们有：

$$\texttt{WFFI-R} = A(\boldsymbol{y}, \boldsymbol{x}) := \sum_{i=1}^{r} i \cdot y_i + \sum_{i=1}^{s} (r+i) \cdot x_i.$$

接下来，我们证明一个依赖于 $\boldsymbol{x}$ an $\boldsymbol{y}$ 的 OPT 下界。注意，在重量为$\leq 2/3$ 的 WFFI-R 箱 B_r，B_{r+1}、.....、B_{r+s} 中，至多有一个箱能装下两个或更多物品（这些箱中最先开启的那个箱；因为放入后续箱中的物品大小——因此重量也——严格大于 $1/3$）。因此，这 $s+1$ 个箱中有 s 个必须各自只装有一个物品，并且不存在一对这样的单个物品能够共同装入一个箱。因此，这 s 个单独的物品必须分别放入最优装箱中的不同箱子。这意味着

$$w(O_i) \geq w(B_{r+i}) = x_i, \quad \text{for all } i \in [s]. \tag{4}$$

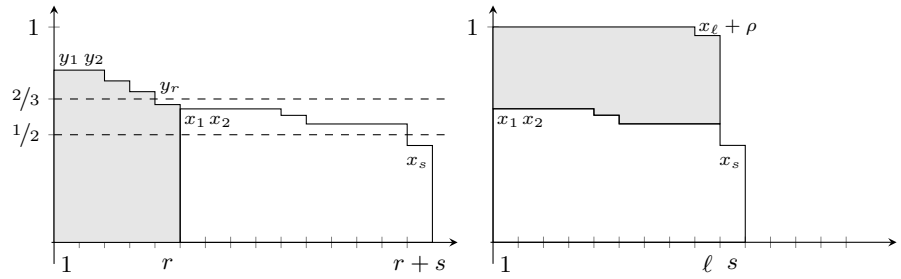

图 3. 定义 $A(\boldsymbol{y},\ \boldsymbol{x})$ 的装箱示意图（左），以及定义 $B(\boldsymbol{y},\ \boldsymbol{x})$ 的对应伪装箱（右）：ℓ 和 ρ 的取值确保两个阴影区域相等。

现在，令 $\ell \in \mathbb{N}$ 为满足 $\sum_{i=1}^{\ell-1}(1-x_i) < \sum_{i=1}^{r} y_i \le \sum_{i=1}^{\ell}(1-x_i)$ 的唯一索引，并定义 $\rho := \sum_{i=1}^{r} y_i - \sum_{i=1}^{\ell-1}(1-x_i) \in (0,\ 1-x_\ell]$。我们断言

$$\texttt{OPT} \ge B(\boldsymbol{y},\boldsymbol{x}) := \frac{1}{2}\ell(\ell-1) + (x_\ell+\rho)\ell + \sum_{i=\ell+1}^{s} i\cdot x_i,$$

其中，当 $\ell \ge s$ 时，最后一项求和为 0；这对应于一种“伪装箱”：箱 1、……、$\ell-1$ 的重量为 1，箱 $\ell+1$、……、s 的重量为 x_i，并将箱ℓ的重量调整为 $x_\ell+\rho$，使总重量等于 $\sum_i x_i + \sum_i y_i$。示意见图 3。显然，这给出了最优值的下界，因为将重量从靠后的箱转移到靠前的箱只会改善解；而定义 $B(\boldsymbol{x},\ \boldsymbol{y})$ 的伪装箱将全部重量装得比任何满足 (4) 的装箱 O_1、……、O_q 都靠前。我们将函数 $B(\cdot,\ \cdot)$ 的定义扩展到任意一对向量 $(\boldsymbol{y}',\ \boldsymbol{x}') \in \mathbb{R}^r_{>0} \times [0,1]^t$，其中 $t \le s$，并令所有 $i > t$ 均满足 $x'_i = 0$。

上述不等式意味着 WFFI-R/OPT 的上界为$R(\boldsymbol{y},\boldsymbol{x}) := A(\boldsymbol{y},\boldsymbol{x})/B(\boldsymbol{y},\boldsymbol{x})$。接下来我们给出一系列技术性断言（证明见附录），使我们能够计算当 $\boldsymbol{x}$ 和$\boldsymbol{y}$ 如上所述时 $R(\boldsymbol{y},\boldsymbol{x})$ 的最大值。第一个断言表明，对于某些形如 $\boldsymbol{y}' = [\frac{2}{3},\dots,\frac{2}{3},\alpha] \in \mathbb{R}^r$ 和 $\boldsymbol{x}' = [\alpha,\dots,\alpha] \in \mathbb{R}^t$ 的向量 $\boldsymbol{y}'$和 $\boldsymbol{x}'$（其中$\alpha \in [\frac{1}{2},\frac{2}{3}]$和 $t \le s$ 为某些值），我们得到一个上界。其证明依赖于对向量 $\boldsymbol{y}$ 和 $\boldsymbol{x}$ 的某些坐标取平均，并使用序贯舍入过程以使向量 $\boldsymbol{x}$ 中的坐标相等，并将 $y_1,\dots,y_{r-1}$ 减小到 $\frac{2}{3}$。

断言 2 $R(\boldsymbol{y}',\boldsymbol{x}')$ $\alpha \in [\frac{1}{2},\frac{2}{3}]$ $\boldsymbol{y}' = [\frac{2}{3},\dots,\frac{2}{3},\alpha] \in \mathbb{R}^r$ t $R(\boldsymbol{y},\boldsymbol{x}) \le$存在一个……和一个整数，$\boldsymbol{x}' = [\alpha,\dots,\alpha] \in \mathbb{R}^t$ 使得$(,)$ $(,)$对向量……和……成立。

接下来，当 $\boldsymbol{y}'$ 和 $\boldsymbol{x}'$ 具有前述断言中得到的形式时，我们给出 $R(\boldsymbol{y}',\boldsymbol{x}')$的一个便于使用的上界。

命题 3 令 $\boldsymbol{y}' := [\frac{2}{3},\dots,\frac{2}{3},\alpha] \in \mathbb{R}^r$ 和 $\boldsymbol{x}' := [\alpha,\dots,\alpha] \in \mathbb{R}^t$。若 $\frac{2}{3}(r-1)+\alpha \le (1-\alpha)t$，则成立

$$R(\boldsymbol{y}',\boldsymbol{x}') \le H_1(\alpha,r,t) := \frac{3(1-\alpha)(3\alpha(t+1)(2r+t)+2r(r-1))}{3\alpha+4r^2+(6\alpha-2)r+9(1-\alpha)\alpha t(t+1)-2}.$$

否则（若 $2\,3(r-1)+\alpha>(1-\alpha)t$），以下界成立：

$$R(\boldsymbol{y}',\boldsymbol{x}')\le H_2(\alpha,r,t):=\frac{6r(r-1)+9\alpha(t+1)(2r+t)}{(2r+3\alpha(t+1)-2)(2r+3\alpha(t+1)+1)}.$$

我们首先对第二个表达式求界。在这种情况下，我们得到的界为 $\frac{13}{8}=1.625$。

命题4 对于所有 $\alpha\in[\frac{1}{2},\frac{2}{3}]$、$r\ge 1$ 和 $0\le t\le \frac{2/3(r-1+\alpha)}{1-\alpha}$，$H_2(\alpha,r,t)\le\frac{13}{8}$。

接下来，我们对 H_1 求界。结果表明，$H(\alpha,r,t)$ 关于 α 的导数在 α 的定义域上非正，因此令 $\alpha=\frac{1}{2}$ 可得到一个上界。

命题5 对于所有 $\alpha\in[\frac{1}{2},\frac{2}{3}]$、$r\ge 1$ 和 $t\ge 0$，均有

$$H_1(\alpha,r,t)\le H_1(\frac{1}{2},r,t)=\frac{3(4r^2+r(6t+2)+3t(t+1))}{16r^2+4r+9t^2+9t-2}.$$

最后，我们通过在定义域 $r\ge 1,t\ge 0$ 上最大化上述表达式，得到上界。令 $u:=r-1\ge 0$。据此可将前一个上界改写为

$$H_1(\frac{1}{2},r,t)=R_1(u,t):=\frac{3(6+3t^2+10u+4u^2+9t+6ut)}{18+9t+9t^2+36u+16u^2}.$$

对于某些非负变量 u 和 t。我们不采用复杂的微分学来最大化 R_1，而是给出一个简短的证明：依据一个证书可知，某个矩阵是余正定的（例如参见 [3]）；该证书是通过求解一个半定规划得到的。注意，$R_1(u,t)=\boldsymbol{z}^TA\boldsymbol{z}/\boldsymbol{z}^TB\boldsymbol{z}$，其中

$$\boldsymbol{z}=\begin{pmatrix}u\\t\\1\end{pmatrix},\quad A=\begin{pmatrix}12&9&15\\9&9&27/2\\15&27/2&18\end{pmatrix},\quad\text{and}\quad B=\begin{pmatrix}16&0&18\\0&9&9/2\\18&9/2&18\end{pmatrix}.$$

令 $\tau:=\frac{7+\sqrt{37}}{8}$，$X:=\begin{pmatrix}0&0&3\\0&0&9/8\\3&9/8&0\end{pmatrix}$。读者可以验证，该矩阵

$$Z=\tau B-A-X=\begin{pmatrix}2(1+\sqrt{37})&-9&9/4\cdot(\sqrt{37}-1)\\-9&9/8\cdot(\sqrt{37}-1)&9/16\cdot(\sqrt{37}-19)\\9/4\cdot(\sqrt{37}-1)&9/16\cdot(\sqrt{37}-19)&9/4\cdot(\sqrt{37}-1)\end{pmatrix}$$

是半正定的。因此，对于所有 $u,\ t\in\mathbb{R}$，$\boldsymbol{z}^T(\tau B-A)\boldsymbol{z}\ge\boldsymbol{z}^TX\boldsymbol{z}=6u+9/4\cdot t$ 均成立；由于 $u\ge 0$ 和 $t\ge 0$，这个量非负。这证明了 $\tau\cdot\boldsymbol{z}^TB\boldsymbol{z}-\boldsymbol{z}^TA\boldsymbol{z}\ge 0$，因此 $R_1(u,t)\le\tau$。我们注意到，该界是紧的，因为对于满足 t_i/u_i 收敛到 $2\,9(1+\sqrt{37})$ 的整数序列 $\{(t_i,u_i)\}_{i\in\mathbb{N}}$，$R_1(u_i,t_i)\to\tau$。由断言 2–5，我们由此证明了 $R(\boldsymbol{y},\boldsymbol{x})\le\max(13\,8,\frac{7+\sqrt{37}}{8})=\frac{7+\sqrt{37}}{8}$，从而完成了证明。■

致谢。作者感谢一群同事，特别是 Rico Raber 和 Martin Knaack，感谢他们在团队静修期间进行的讨论，这些讨论为本文的完成作出了贡献。作者还感谢匿名审稿人，他们的意见有助于提升本文稿件的质量。

参考文献

1. Dobson, G., Nambimadom, R.S.：批量装载与调度问题。Oper. Res. **49**(1), 52–65 (2001) 2. Dósa, G., Sgall, J.：首次适配装箱：紧确分析。载于：第30届国际计算机科学理论方面研讨会，STACS 2013。LIPIcs，第20卷，第538–549页 (2013) 3. Dür, M.：余正规划综述。载于：优化及其工程应用的最新进展，第3–20页。施普林格 (2010) 4. Epstein, L., Johnson, D.S., Levin, A.：最小和装箱。J. Comb. Optim.**36**(2), 508–531 (2018) 5. Epstein, L., Levin, A.：最小加权和装箱。载于：近似与在线算法，第5届国际研讨会，WAOA 2007。《计算机科学讲义》，第4927卷，第218–231页 (2007) 6. Feige, U., Lovász, L., Tetali, P.：最小和集合覆盖的近似算法。Algorithmica**40**(4), 219–234 (2004) 7. Fowler, J.W., Monch, L.：并行批次（p-批处理）加工调度综述。Eur. J. Oper. Res. **298**(1), 1–24 (2022) 8. Garey, M.R., Graham, R.L., Ullman, J.D.：内存分配算法的最坏情况分析。载于：第4届年度ACM计算理论研讨会论文集，STOC 1972。第143–150页 (1972) 9. Graham, R., Lawler, E., Lenstra, J., Kan, A.R.：确定性排序与调度中的优化与近似：综述。载于：《离散优化 II》，《离散数学年刊》第5卷，第287–326页 (1979) 10. Li, R., Tan, Z., Zhu, Q.：采用最小和准则的非相同作业大小批次调度。J. Comb. Optim. **42**(3), 543–564 (2021) 11. Schwiegelshohn, U.：最大比率优先规则的 Kawaguchi–Kyan 界的另一种证明。Oper. Res. Lett. **39**(4), 255–259 (2011)

附录：省略的证明

断言 *1* 的证明。 我们假设箱子权重为 $W_1 \geq \ldots \geq W_p$，且$W_{p-1} \geq \frac{2}{3}$。记 $S := \sum_k W_k$为所有物品重量之和。进一步定义 $x = \lfloor S \rfloor$ 和 $f = S - s \in [0, 1)$。显然，最优值 $\geq \sum_{i=1}^{x} i + f \cdot (x+1)$，这对应于一种尽可能早地放入全部重量的理想装箱。于是，我们写作

$$\texttt{OPT} \geq \sum_{i=1}^{x} i + f \cdot (x+1) = (x+1) \cdot (\frac{x}{2} + f) = \frac{1}{2}(x+f)(x+f+1) + \frac{1}{2} f(1-f),$$

这意味着最优值 $\geq S(S+1)/2$，因为 $f(1-f) \geq 0$。然后，我们观察到，对 W_1、……、W_{p-1} 取平均只会得到更差的装箱（因为我们将重量从前面的箱转移到后面的箱），从而得到 WFFI-R 的一个上界：

$$\texttt{WFFI-R} \leq \frac{S - W_p}{p-1} \cdot \sum_{i=1}^{p-1} i + p \cdot W_p = \frac{p}{2}(S + W_p).$$

此外，我们知道 $S \geq \frac{2}{3}(p-1) + W_p$，这意味着

$$\texttt{WFFI-R} \leq p(S - \frac{1}{3}(p-1)) \leq \frac{(3S+1)^2}{12},$$

其中，最后一个不等式是通过对 p 取最大值得到的。于是，我们得到

$$\frac{\texttt{WFFI-R}}{\texttt{OPT}} \leq \frac{(3S+1)^2}{6S(S+1)} = \frac{3}{2} - \frac{3S-1}{6S(S+1)}.$$

显然，对于所有 $S > 1$，上述界都是 $\leq \frac{3}{2}$；而当 $S \leq 1$（所有物品都能装入同一个箱）时，WFFI-R 算法显然是最优的。

命题 *2* 的证明。设 $\ell \in \mathbb{N}$ 满足：定义 $B(\boldsymbol{x}, \boldsymbol{y})$ 的伪装箱使用了 ℓ 个箱，即存在 $\rho \in (0, 1 - x_\ell]$ 使得

$$\sum_{k=1}^{r} y_k = \sum_{j=1}^{\ell-1} (1 - x_j) + \rho.$$

证明分为两步。首先，我们给出如下形式的一个上界

$$R(\boldsymbol{y}, \boldsymbol{x}) \leq R([\beta, \ldots, \beta, \alpha], [\alpha, \ldots, \alpha]), \tag{5}$$

然后，我们将 $r-1$ 从 $\boldsymbol{y}$ 的 β 个首坐标减小至 $2\,3$。对于第一步，我们区分两种情况。若 $\ell > s$ 或 $\rho \geq \frac{1}{3}$，则属于第一种情况。接着，我们定义 $\beta = 1/(r-1) \cdot (y_1 + \ldots + y_{r-1})$、$\ell_0 = \min(\ell, s)$、$\alpha = 1/(l_0+1) \cdot (x_0 + \ldots + x_{\ell_0})$、$\bar{\boldsymbol{y}} = [\beta, \ldots, \beta, \alpha] \in \mathbb{R}^r$ 以及 $\bar{\boldsymbol{x}} = [\alpha, \ldots, \alpha, x_{\ell_0+1}, \ldots, x_s] \in \mathbb{R}^s$，即 $\bar{\boldsymbol{y}}$ 和 $\bar{\boldsymbol{x}}$分别通过对 $y_1, \ldots, y_{r-1}$ 和 $x_0, \ldots, x_{\ell_0}$取平均得到（回想一下，$x_0 = y_r$）。我们声称，对这些坐标取平均会提高界。

事实上，我们有 $A(\boldsymbol{y},\boldsymbol{x}) \le A(\bar{\boldsymbol{y}},\bar{\boldsymbol{x}})$，因为 $(\bar{\boldsymbol{y}},\bar{\boldsymbol{x}})$ 是通过将重量从较早的箱转移到较晚的箱得到的。另一方面，可以看出，对这些坐标取平均并未改变定义 B 的伪装箱，因此 $B(\boldsymbol{y},\boldsymbol{x}) = B(\bar{\boldsymbol{y}},\bar{\boldsymbol{x}})$。当 $\ell > s$ 时，这始终成立；否则需要满足 $\rho \ge \alpha - x_\ell$，而这得益于 $\alpha - x_\ell \le \frac{2}{3} - \frac{1}{3} = \frac{1}{3}$ 以及我们的假设 $\rho \ge \frac{1}{3}$。因此，$R(\boldsymbol{y},\boldsymbol{x}) \le R(\bar{\boldsymbol{y}},\bar{\boldsymbol{x}})$。

如果 $\ell > s$，我们已经得到了形式如 (5) 的界。否则，我们将采用序贯舍入过程，将 $\bar{\boldsymbol{x}}$ 的最后几个坐标舍入为 α 或 0；后一种情况等价于删除 $\bar{\boldsymbol{x}}$的一些坐标。令 j 满足 $\bar{x}_1 \ge \ldots \ge \bar{x}_{s-j} > \bar{x}_{s-j+1} = \bar{x}_{s-j+2} = \ldots = \bar{x}_s$，即 $\bar{\boldsymbol{x}}$ 末尾有 j 个相等的坐标，并将通过对 $\bar{\boldsymbol{x}}$最后 j 个坐标分别加上 $\delta \in \mathbb{R}$ 得到的向量记为 $\bar{\boldsymbol{x}}(\delta)$。不难看出，函数 $A(\bar{\boldsymbol{y}}, \bar{\boldsymbol{x}}(\delta))$ 和 $B(\bar{\boldsymbol{y}}, \bar{\boldsymbol{x}}(\delta))$ 在区间 $[-\bar{x}_s, \bar{x}_{s-j} - \bar{x}_{s-j+1}]$上都是线性的。这意味着 $R(\boldsymbol{y}, \bar{\boldsymbol{x}}(\delta))$ 在该区间上必为单调有理函数，因此，令 $\delta = -\bar{x}_s$（这意味着删除 $\bar{\boldsymbol{x}}$最后 j 个坐标），或令 $\delta = \bar{x}_{s-j} - \bar{x}_{s-j+1}$（此时 $\bar{\boldsymbol{x}}$末尾相等坐标的数量增加），都能得到更大的界。重复这一过程即可得到所需形式 (5) 的界。

在第二种情形下，$\ell \le s$ 和 $\rho < \frac{1}{3}$。接下来，我们大致沿用上述方法，但稍作调整。具体而言，现在我们需要定义 $\ell_0 = \ell - 1$，以确保对 x_0、……、x_{ℓ_0} 取平均不会改变定义 $B(\boldsymbol{y}, \boldsymbol{x})$的伪装箱。因此，在序贯舍入过程的最后一步，当我们将一组坐标（ℓ，……、s'）舍入为 0 或 α时，必须确保对于所有 $\delta \in [-x_\ell, \alpha - x_\ell]$，伪装箱所需的箱数 ℓ 都不发生变化，从而“仍处于”分段线性函数 $\delta \mapsto B(\bar{\boldsymbol{y}}, \bar{\boldsymbol{x}}(\delta))$的同一段内。只要 $x_\ell + \rho + \delta \le 1$，这一点就成立；而根据我们的假设，$\rho < \frac{1}{3}$，因此对所有 $\rho \le \alpha - x_\ell$都有 $x_\ell + \rho + \delta \le x_\ell + \rho + \alpha - x_\ell = \rho + \alpha < \frac{1}{3} + \frac{2}{3} = 1$。

尚需证明该断言的第二步，其中我们将 β减小至$\frac{2}{3}$的值。当 $r = 1$时无需证明，因此我们假设$r > 1$。记 $\boldsymbol{x}' = [\alpha, \ldots, \alpha] \in \mathbb{R}^t$ 为该证明第一步结束时得到的向量，并令 $\boldsymbol{y}' := [\frac{1}{3}, \ldots, \frac{1}{3}, \alpha] \in \mathbb{R}^r$。进一步，如下定义 $\ell' \in \mathbb{N}$：若$\frac{2}{3}(r-1) + \alpha \le (1-\alpha) \cdot t$，则 ℓ'是使得$\frac{2}{3}(r-1) + \alpha \le (1-\alpha) \cdot \ell'$成立的最小整数；否则，它是使得$\frac{2}{3}(r-1) + \alpha \le (1-\alpha) \cdot t + (\ell' - t)$成立的最小整数。可以看出，$\ell'$定义了定义 $B(\boldsymbol{y}', \boldsymbol{x}')$的伪装箱中重量为 $> x'_i$ 的箱的数量，即 ℓ' 是在尺寸为 $\boldsymbol{x}'$的箱上剩余空间中装箱总重量为 $\sum_{i=1}^{r} y'_i = \frac{2}{3}(r-1) + \alpha$ 所需的最小箱数。

接下来，我们通过区分两种情况来证明 $\ell' \ge \frac{r}{2}$。如果 $\frac{2}{3}(r-1) + \alpha \le (1-\alpha) \cdot t$，则

$$\ell' \ge \frac{\frac{2}{3}(r-1) + \alpha}{1-\alpha} \ge \frac{4}{3}(r-1) + 1 \ge \frac{r}{2},$$

其中第二个不等式由 $\alpha \ge \frac{2}{5}$ 得出，最后一个不等式对所有$r \ge \frac{2}{5}$ 都成立。否则，我们有

$$\ell' \ge \frac{2}{3}(r-1) + \alpha(1+t) \ge \frac{2}{3}(r-1) + \frac{1}{2} \ge \frac{r}{2},$$

其中第二个不等式用到了 $\alpha \geq \frac{1}{2}$ 和 $t \geq 0$，最后一个不等式对所有 $r \geq 1$都成立。

为完成此证明，注意到

$$A(\boldsymbol{y}', \boldsymbol{x}') = A(\bar{\boldsymbol{y}}, \boldsymbol{x}') - (\beta - \frac{2}{3}) \sum_{i=1}^{r-1} i = A(\bar{\boldsymbol{y}}, \boldsymbol{x}') - (\beta - \frac{2}{3})(r-1)\frac{r}{2}.$$

另一方面，在定义 $B(\boldsymbol{y}, \boldsymbol{x}')$ 的伪装箱中，我们从索引为 $\geq \ell'$ 的箱中总共移除了重量 $(\beta - \frac{2}{3}) \cdot (r-1)$，因此

$$B(\boldsymbol{y}', \boldsymbol{x}') \leq B(\bar{\boldsymbol{y}}, \boldsymbol{x}') - (\beta - \frac{2}{3})(r-1)\ell'.$$

现在，$A(\bar{\boldsymbol{y}}, \boldsymbol{x}') \geq B(\bar{\boldsymbol{y}}, \boldsymbol{x}') - \ell' \geq \frac{r}{2}$ $A(\bar{\boldsymbol{y}}, \boldsymbol{x}') \cdot \ell' \cdot (r-1)(\beta - \frac{2}{3}) \geq (\ ,\ )$ $B(\bar{\boldsymbol{y}}, \boldsymbol{x}') \cdot (r-1)$ $(\ ,\ )$ $(\ )(\ )$ $(\bar{\boldsymbol{y}}, \ )$ $\frac{r}{2}(\ )(\beta - \frac{2}{3})$。最后，从两边减去 $(\bar{\boldsymbol{y}}, \ )$ $(\bar{\boldsymbol{y}}, \ )$，再除以 $B(\bar{\boldsymbol{y}}, \boldsymbol{x}') \cdot ((\beta - \frac{2}{3})(r-1)\ell' - B(\bar{\boldsymbol{y}}, \boldsymbol{x}')) < 0$，即可得到所需结果：

$$R(\bar{\boldsymbol{y}}, \boldsymbol{x}') = \frac{A(\bar{\boldsymbol{y}}, \boldsymbol{x}')}{B(\bar{\boldsymbol{y}}, \boldsymbol{x}')} \leq \frac{A(\bar{\boldsymbol{y}}, \boldsymbol{x}') - (\beta - \frac{2}{3})(r-1)\frac{r}{2}}{B(\bar{\boldsymbol{y}}, \boldsymbol{x}') - (\beta - \frac{2}{3})(r-1)\ell'} \leq \frac{A(\boldsymbol{y}', \boldsymbol{x}')}{B(\boldsymbol{y}', \boldsymbol{x}')} = R(\boldsymbol{y}', \boldsymbol{x}').$$

命题 *3* 的证明。令 $\boldsymbol{y}' := [\frac{2}{3}, \ldots, \frac{2}{3}, \alpha] \in \mathbb{R}^r$ 和 $\boldsymbol{x}' := [\alpha, \ldots, \alpha] \in \mathbb{R}^t$。那么，我们有

$$A(\boldsymbol{y}', \boldsymbol{x}') = \frac{1}{3} r(r-1) + \frac{\alpha}{2}(t+1)(2r+t).$$

如果 $\sum_i y'_i = \frac{2}{3}(r-1) + \alpha \leq (1-\alpha)t$，那么我们有

$$B(\boldsymbol{y}', \boldsymbol{x}') = \frac{\ell(\ell-1)}{2} + (\alpha + \rho)\ell + \frac{\alpha}{2}(t-\ell)(t+\ell+1),$$

其中，$\ell \in \mathbb{N}$ 和 $\rho \in (0, 1-\alpha]$ 满足 $\frac{2}{3}(r-1) + \alpha = \ell(1-\alpha) + \rho$。令$\tilde{\ell} := \frac{2/3(r-1+\alpha)}{1-\alpha}$，使得 $\frac{2}{3}(r-1) + \alpha = \tilde{\ell}(1-\alpha)$，并定义

$$B' = \frac{\tilde{\ell}(\tilde{\ell}+1)}{2} + \frac{\alpha}{2}(t-\tilde{\ell})(t+\tilde{\ell}+1).$$

将 $\tilde{\ell}$ 代入 B' 中的 $(\ell-1) + \frac{\rho}{1-\alpha}$，经过一些化简后，得到 $B(\boldsymbol{y}', \boldsymbol{x}') - B' = \frac{\rho}{(1-(\alpha+\rho))2(1-\alpha)} \geq 0$。因此，我们有

$$\begin{aligned} R(\boldsymbol{y}', \boldsymbol{x}') \leq \frac{A(\boldsymbol{y}', \boldsymbol{x}')}{B'} &= \frac{\frac{1}{3}r(r-1) + \frac{\alpha}{2}(t+1)(2r+t)}{\frac{(2r+1)(3\alpha+2r-2)}{18(1-\alpha)} + \frac{\alpha}{2}t(t+1)} \\ &= \frac{3(1-\alpha)(3\alpha(t+1)(2r+t) + 2r(r-1))}{3\alpha + 4r^2 + (6\alpha-2)r + 9(1-\alpha)\alpha t(t+1) - 2}. \end{aligned}$$

在另一种情况下，有 $\sum_i y'_i = \frac{2}{3}(r-1) + \alpha > (1-\alpha)t$，我们以类似的方式进行：令 $\ell \in \mathbb{N}, \rho \in (0,1]$ 和 $\tilde{\ell} \in \mathbb{R}$ 满足 $\frac{2}{3}(r-1) + \alpha = (1-\alpha)t + (\ell - t - 1) + \rho = (1-\alpha)t + (\ell - t)$，于是

$$B(\boldsymbol{y}', \boldsymbol{x}') = \frac{\ell(\ell-1)}{2} + \rho\ell,$$

~ ~ ~并定义 $B'' = \ell\ \ell+1\ /2$ $\ell = \ell - \ 1-\rho$ $B\ \boldsymbol{y}'\ \boldsymbol{x}'\ - B'' =:$ $\frac{\rho}{2}\ 1-\rho\ \geq 0$ $\tilde{\ell}$ $\frac{2}{3}\ r-1\ +\alpha\ 1+t$ B'' (-)。利用 ( )，我们得到 (,) ( )。最后，将 ( ) ( ) 代入，得到

$$R(\boldsymbol{y}', \boldsymbol{x}') \leq \frac{A(\boldsymbol{y}', \boldsymbol{x}')}{B''} = \frac{\frac{1}{3}r(r-1) + \frac{\alpha}{2}(t+1)(2r+t)}{\frac{1}{18}(2r+3\alpha(t+1)-2)(2r+3\alpha(t+1)+1)}$$
$$= \frac{6r(r-1)+9\alpha(t+1)(2r+t)}{(2r+3\alpha(t+1)-2)(2r+3\alpha(t+1)+1)}.$$

命题*4*的证明。 通过直接计算可得

$$\frac{dH_2(\alpha,r,t)}{d\alpha} = \frac{-9(t+1)}{h_2(\alpha,r,t)^2}\Big(r^2\big[6(2\alpha-1)+4t(3\alpha-1)\big] + \big[t(2+9\alpha^2(1+t)^2)\big]$$
$$+ r\big[18\alpha^2t^2 + 2(18\alpha^2-6\alpha+1) + 2(9\alpha^2-6\alpha+3)\big]\Big).$$

对于所有 $t \geq 0$， $r \geq 0$和 $\alpha \geq \frac{1}{2}$，此表达式均非负，因为 $2\alpha-1 \geq 0$对于 $\alpha \geq \frac{1}{2}$ 成立，且两个二次函数 $18\alpha^2-6\alpha+1$和 $9\alpha^2-6\alpha+3$始终非负。因此， $H_2(\alpha,$ $r,\ t) \leq H_2(1/2,\ r,\ t) = {}^{6}/_{(4r^2+3t(1+t)+r(2+6t))\,(-1+4r+3t)(5+4r+3t)}$。~~接下来~~，我们计算

$$\frac{dH_2(\frac{1}{2},r,t)}{dt} = \frac{18}{h_3(r,t)^2}\Big(t^2[3+6r] + 2t[4r^2+10r-5] + [16r^2-2r-5]\Big),$$

其中， $h_3(r,\ t)$ 是 $H_2(1/2,\ r,\ t)$ 的分母。对于所有 $t \geq 0$ 和 $r \geq 1$，此表达式均非负，因为对于所有 $r \geq 1$， $4r^2+10r-5 \geq 0$ 和 $16r^2-2r-5 \geq 0$ 均成立。因此，函数 $t \mapsto H_2(1/2,\ r,\ t)$ 在定义域$t \in [0, 4/3(r-1)+1]$上是单调不减的，这意味着

$$H_2(\alpha,r,t) \leq H_2(\frac{1}{2},r,\frac{4}{3}(r-1)+1) = \frac{26r^2+2r-1}{16r^2+4r-2}$$

对于所有 $\alpha \geq \frac{1}{2}$和 $t \in [0, 4/3(r-1)+1]$。最后，容易验证该上界关于 r 对于 $r \geq 1$是非递减的，因此通过令 $r \to \infty$我们得到上界$13\ 8$ 。

命题*5*的证明。 标准代数运算表明

$$\frac{dH_1(\alpha,r,t)}{d\alpha} = \frac{-3}{h(\alpha,r,t)^2}\Big(8r^4 + (t+1)\big[24r^3(2\alpha-1) + 3t(3\alpha^2-4\alpha+2)\big]$$
$$+ 6r^2\big[(t^2+3t)(3\alpha^2-2\alpha+1) + 6\alpha^2-4\alpha+1\big]$$
$$+ 2r\big[6t^2(2\alpha-1) + 9t\alpha^2 + 9\alpha^2 - 12\alpha + 5\big]\Big),$$

其中 $h_1(\alpha,r,t)$是 $H_1(\alpha,r,t)$的分母。容易看出上述表达式对于所有 $r \geq 0, t \geq 0$ 和 $\alpha \geq \frac{1}{2}$都是非正的，因为 $2\alpha-1 \geq 0$对于$\alpha \geq \frac{1}{2}$，且二次函数表达式 $3\alpha^2-4\alpha+2$、 $3\alpha^2-2\alpha+1$、 $6\alpha^2-4\alpha+1$和$9\alpha^2-12\alpha+5$总是非负的。因此， $\alpha \mapsto H_1(\alpha,r,t)$在区间 $[\frac{1}{2}, \frac{2}{3}]$ 上是非递增的，通过将 α替换为$1\ 2$即可得到结果。