摘要

本文研究的是总和型装箱问题 FFI 算法最坏情况比的改进. 我们先定义了最差实例, 我们希望仅仅对于最差实例进行研究. 我们将所有的实例都转化为对应的最差实例, 然后通过最差实例得到关于实例箱子数和实例目标值之间的大小关系. 通过箱子数趋向于无穷得到一个双向不等式的逼近, 从而可以知道目标值之比等于实例箱子数之比. 之后我们就只需要研究所有目标值之比等于箱子数之比的实例的性质即可. 一方面目标值之比取到很大的时候实例箱子数之比不会很大, 另一方面当实例箱子数之比很大的时候目标值之比也不会很大, 从而可以从两方面去进行逼近. 最后我们可以得到渐近性能比上界不超过 1.5 的结论.

关键词: FFI, 总和型装箱, 最差实例, 渐近上限

Abstract

This article studies the improvement of worst situation in FFI algorithm for Min-Sum bins problem. First, we define what is ”worst case” and we hope to only study these worst cases and they will show the properties of all cases. To do this, we need to map every cases to these worst cases and then by using the definition of ”worst” to derive the relationship between the proportion of the bins and the proportion of the targets. By making the number of bins to infinity we will get the approaching of the inequality and finally derive an equality. Namely, when the number of cases tends to be infinity, these worst cases will have property like the proportion of the bins equals to the proportion of the targets. Then, we need to study all the cases which hold this property. On the one hand, when the proportion of targets is large, the proportion of bins would not be large. On the other hand, when the proportion of bins is not large, the proportion of targets would be large. So, we can approximate it from 2 ways. Finally, we will see that the proportion for limit of capacity should be no more than 1.5.

Key word: FFI, Min-sum, worst cases, limit upper bound

总和型装箱问题 FFI 算法最坏情况比的改进

May 28, 2019

1 绪论

如果一个问题是 NP 完全的问题, 那么这个问题在现有的知识结构下我们认为不存在多项式时间内的方法去解决它. 但是我们可以通过一些多项式时间内的操作性强的启发式算法去研究这个问题. 我们研究的问题就是一个 NP-Complete 问题装箱问题的特殊情况下的启发式算法的性能.

1.1 背景

1.1.1 什么是 NP 完全问题

正式的来说, 一个 NP 完全问题是一个取一些字符串为输入并要求输出为是或否的问题. 若有一个算法 (譬如图灵机, 或一个 LISP 或 Pascal 的程序并有无限的内存) 能够在最多 nk 步内对一个串长度为 n 的输入给出正确答案, 其中 k 是某个不依赖于输入串的常数, 则我们称该问题可以在多项式时间内解决, 并且将它置入类 P. 直观的讲，我们将 P 中的问题视为可以较快解决的问题.

现在假设有一个算法 A(w,C) 取两个参数, 一个串 w，也就是我们的决定问题的输入串，而另一个串 C 是“建议证明”, 并且使得 A 在最多 n^k 步之内产生”是/否”答案（其中 n 是 w 的长度而 k 不依赖于 w）. 进一步假设: w 是一个答案为“是”的例子, 当且仅当, 存在 C 使得 A(w,C) 返回“是”. 则我们称这个问题可以在非决定性多项式时间内解决, 且将它放入 NP 类. 我们把算法 A 作为一个所建议的证明的检验器, 它运行足够快.

NP 完全问题导致了这个问题只能通过近似算法来进行解决, 而我们研究的问题就是近似算法的性能问题.

1.1.2 问题介绍

将大小为 $w_1, w_2, \ldots, w_n$ 的 n 个物品放入容量为 1 的 m 个箱子中，放入每个箱子的物品大小之和不超过箱子容量，且 $|B_1| \geq |B_2| \geq \ldots \geq |B_n|$.

我们的目标为极小化 $\sum_{i=1}^{m} i\,|B_i|$.

1.1.3 FFI 近似算法介绍

FFI 算法是指将物品按照重量从小到大进行排序, 然后按照 FF 算法的方法进行装箱.

这样的算法是 FF 算法的一种, 他继承了 FF 算法的很多性质.

1.1.4 FFI 算法与 FFD 算法的比较

FF 算法是一个比较优秀的近似算法, 而其中受关注最多的是 FFD 算法. FFD 算法在研究使得箱子数最少的装箱问题的时候性能相当卓越, 但是在这个问题中并不能体现很出色的性能. 这是因为存在一个实例可以使得 FFD 算法无限差. 如图1.1即为 FFD 算法失效的实例.

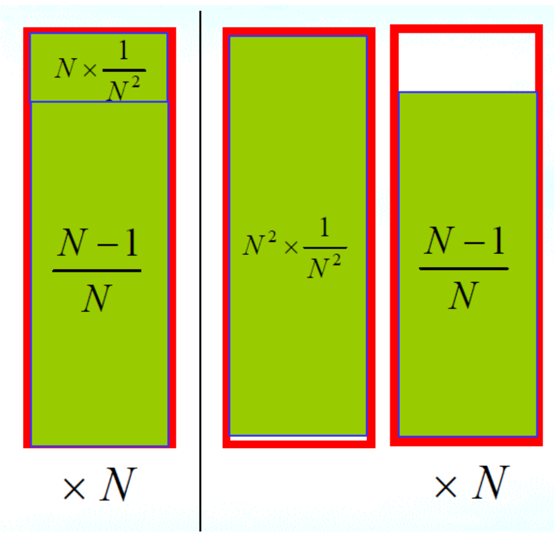

Figure 1.1: FFD fails

FFD 算法以及" 最优算法" 的目标值为1.1. 这里的" 最优算法" 不一定是真正意义上的最优算法. 只是存在的一种算法.

$$
\begin{aligned}
C^{FFD} &= (N+1)\sum_{i=1}^{N} i = \frac{N(N+1)^2}{2} \\
C^{*} &= N^2 + \sum_{i=1}^{N}(i+1) = \frac{3N(N+1)}{2}
\end{aligned} \tag{1.1}
$$

FFD 不好的原因是我们对放在后面箱子中的物体有一个惩罚, 虽然我们直观上 FFD 算法可以使得箱子数相对比较少, 但是他可能为了使得箱子数非常少使得后面的箱子中放了大量的物体, 从而受到了很严重的惩罚. FFI 算法相比 FFD 算法, 可以尽可能地将重量小的物体放在前面的箱子中, 从而可以减少我们问题对算法的惩罚情况. 因此我们可以猜测 FFI 算法可以比 FFD 算法好.

而对于使得 FFD 算法失效的那一个实例, 我们可以看到的是 FFI 算法的排列与最优排列相同, 从而可以知道性能比为 1. 这让我们猜测 FFI 算法可能是一个可行的启发式算法.

同时由于 FFI 算法实际上是 FF 算法的一种, 所以他也继承了 FF 算法的性质, 例如实际上任意实例的排列的箱子数也不会太多, 至多与最优排列数的箱子数的比例为 1.7. 这保证了箱子数不会太多, 而我们对放在前面箱子的惩罚情况也不会太过于严重, 这让我们相信 FFI 算法或许是一个比较好的启发式算法.[1]

而实际上 FFI 算法则有一个上界估计. 他保证了 FFI 算法对于任意的一个实例性能都不会太差.

1.2 技术路线

我们使用新定义的“最差实例”进行实例改进. 若是存在一个实例使得最差实例的性能比新构造的实例的性能强, 那么我们认为最差实例不再是最差实例, 这造成了矛盾，我们用这种构造方法来对实例的性质进行刻画. 我们可以用新构造的实例满足一些限制条件以方便我们使用性质.

同时注意到 FFI 算法是 FF 算法的一种特例, 所以我们可以用 FF 算法的一些性质进行实例刻画. 注意到改进的实例时候可以对实例有一定的限制, 这个时候我们可以用一些算法的限制情况来得到实例的一些限制, 利用一些论文中的引理从而得到渐近性质. 而一旦得到了渐近性质, 我们就可以通过进一步的限制来得到实例的不同的性质, 把渐近性质当作一个基础的限制条件从而加以使用.

最后可以证明得到 FFI 算法对于总和型装箱问题的渐近性能比不超过 1.5.

2 正文

2.1 FFI

我们证明的思路是:

- 定义最差实例是什么, 说明在什么情况下最差实例是存在的.
- 用某种方法将任何一个实例映射到一个最差实例上, 之后我们就只要对这些最差实例进行研究.
- 我们可以知道当我们考虑的实例的物品数趋向于无穷的时候, 我们同时可以假设他们的箱子数趋向于无穷.
- 在箱子数趋向于无穷的时候最差实例有性质 $C^F(A)/C^*(A) = r(A)$.
- 之后证明满足 $C^F(A)/C^*(A) = r(A)$ 的实例的渐近性能比不超过 1.5.

我们记物品总数为 n_t, 物品总重量为 $weigh$, 最优解法的箱子个数为 n^*, FFI 解法的箱子个数为 n^F, 那么由于 FFI 算法的性质, 在按顺序从前到后的排列中箱子中物品的个数依次是不减的.

我们记最优算法的第 i 个箱子为 B_i, 里面放了 $|B_i|$ 个物品, FFI 算法的第 i 个箱子为 C_i, 里面放了 $|C_i|$ 个物品. $\frac{n^F}{n^*} = r$, 把 d 记为实例中物品重量大于 $\frac{1}{2}$ 的物品数量, 若重量小于等于 $\frac{1}{2}$ 的物品中最重的物品的重量加大于 $\frac{1}{2}$ 的最轻的物品的重量之和大于 1, 定义 d 为实例中物品重量大于 $\frac{1}{2}$ 的物品数量 +1. 定义 t 为实例的最优算法中仅放了一个物品的箱子数. 为了方便, 我们记 n^* 为 n.

Lemma 2.1. $t \leq d$.

Proof. 若 $t \leq d$ 不成立, 实例的最优算法中仅放一个物品的箱子数大于 d, 那么这些箱子必然放在最优算法排序的最后 t 个. 我们可以将最重的 t 个物品换到这 t 个箱子中, 那么这必然也是可行排放方法, 且 C 与最优算法的相同. 1. 若重量小于等于 $\frac{1}{2}$ 的物品中最重的物品的重量加大于 $\frac{1}{2}$ 的最轻的物品的重量之和小于 1, 那么由于 $t > d$, 那么至少有一个重量小于等于 $\frac{1}{2}$ 的物体单独放在一个箱子中, 我们可以将这个箱子放到重量大于 $\frac{1}{2}$ 的物品中最轻的那个单独放的箱子中,

我们可以知道 C 会减小. 这和这是最优算法矛盾. 2. 若重量小于等于 $\frac{1}{2}$ 的物品中最重的物品的重量加大于 $\frac{1}{2}$ 的最轻的物品的重量之和大于 1, 由于 $t > d$, 在最优算法中至少有两个重量小于 $\frac{1}{2}$ 的物品被放在了一个箱子中, 我们可以将这两个物品放在同一个箱子中, 而这依然是可行放置方法, 并且 C 相比最优算法的 C 会减少. 这与该算法是最优算法相矛盾. 综上, 我们有 $t \leq d$. □

我们给出以下记号:

$$C^F = \sum_{i=1}^{n^F} i * |C_i| \qquad C^* = \sum_{i=1}^{n^*} i * |B_i| \tag{2.1}$$

下面给出本文要证明的性质:

Claim 1. 当实例总重量 weigh 趋于无穷时，我们有

$$\frac{C^F}{C^*} \leq \frac{3}{2} \tag{2.2}$$

首先，当实例重量趋于无穷, n 一定趋于无穷, 此时有 $\frac{n^F}{n} \leq 1.7$.

Lemma 2.2. 对于任一实例 A, 假设 n, r 记号是对于该实例的, 若实例不满足

$$\frac{C^F(A)}{C^*(A)} \geq \frac{n^F(A)+1}{n(A)+1} \tag{2.3}$$

那么一定存在一个实例 B, 使得实例 B 的物品数和箱子数不少于实例 A, 并且 $\frac{C^F(B)}{C^*(B)} > \frac{C^F(A)}{C^*(A)}$, 并且 $\frac{C^F(B)}{C^*(B)} \geq \frac{n^F(B)+1}{n(B)+1}$.

Proof. 假设实例 A 有性质:

$$\frac{C^F(A)}{C^*(A)} < \frac{n(A)r(A)+1}{n(A)+1} \tag{2.4}$$

下面用归纳法的方法寻找实例 B.

我们给实例 A 增添一个重量为 1 的物体, 把这个新的实例记为 A^1. 那么 A^1 满足 $C^F(A^1) = C^F(A) + n^F(A) + 1$, $C^*(A^1) \leq C^*(A) + n^*(A) + 1$. 故

$$\frac{C^F(A^1)}{C^*(A^1)} \geq \frac{C^F(A) + n^F(A) + 1}{C^*(A) + n(A) + 1} > \frac{C^F(A)}{C^*(A)}. \tag{2.5}$$

若 $\frac{C^F(A^1)}{C^*(A^1)} < \frac{n(A^1)r(A^1)+1}{n(A^1)+1}$. 我们给实例再加一个重量为 1 的物体, 把新的实例记成 A^2, 这样形成一个归纳过程.

一方面, 下面可以说明这个过程一定有一个极限. 另一方面, 若我们可以说明这个极限存在, 那么这个极限的实例一定满足上面叙述的性质.

记 A^i 是加了 i 个重量为 1 的物品之后的新构造实例, 那么 $r(A^i) \leq \frac{i+n^F(A)}{i} \to 1$. 此时 $\frac{C^F(A)}{C^*(A)} \leq \frac{C^F(A^i)}{C^*(A^i)} \leq \frac{n(A^i)r(A^i)+1}{n(A^i)+1}$. 因为 $r(A^i) \to 1$, 故不等式右边的式子趋向于 1, 此时必然有 $C^*(A) = C^F(A)$, 此时 FFI 的排列方式同样也是最优算法, 此时 $n^F(A) = n(A)$, 与 A 的性质2.3相矛盾. □

Remark 2.1. 对于任意实例 A, 假设 A 的所有物品数按照重量从轻到重进行排列为 $a_1, a_2, \ldots, a_k$, 我们考虑所有的实例集合 $A, A \cup \{1\}, A \cup 2 * \{1\}, \ldots$, 也即实例集实例 A 和实例 A 加上 j 个重量为 1 的箱子, 其中 j 为正整数.

那么我们考虑实例 A 是这样的实例集合中目标值之比最大的情况. 此时 A 一定满足 $\frac{n^F(A)+1}{n(A)+1} \leq \frac{C^F(A)}{C^*(A)}$. 这实际上也是上述证明的一个具体例子.

Definition 2.1 (最差实例). 最差实例定义为在满足若干条件限制下的所有实例中使得 $\frac{C^F(A)}{C^*(A)}$ 最大的实例. 若存在若干个使得 $\frac{C^F(A)}{C^*(A)}$ 达到这些实例中最大值的实例, 那么我们认为他们都是最差实例.

Lemma 2.3. 对任意 k 为整数, 在物品数不超过 k 个的所有实例中必然存在最差实例.

Proof. 因为所有实例的物品数不超过 k 个, 所以 $C^F(A), C^*(A)$ 的取值方法是有限的且不超过 k. 对于每一种取值方法实例只有是可能取到或者不取到. 那么一定存在最优实例.

也就是说对于数组 $(C^F(A), C^*(A)) = (m, n)$. 其中 m,n 为不大于 $\frac{k(k+1)}{2}$ 的任意两个正整数. 这是因为对于任意一个物品数不超过 k 的实例用任何一种排列方式都不可能使得 $C(A) > \frac{k(k+1)}{2}$. 并且其中参数 m, n 满足 $m > n$. 那么 (m, n) 的组合种类数是有限的. 并且对于任何一种组合都是能取到或者不能取到, 也即能够存在实例使得这种组合能够取到或者是不存在实例使得这种组合能够取到. 我们选出所有能取到的组合中目标值之比最小的那个实例定义为最差实例即可. □

Remark 2.2. 当 k 趋于无穷的时候, 可以认为所有物体数量不大于 k 个的所有实例中的最差实例会趋向于所有实例中的最差实例. 上述性质是对于最差实例定义的, 所以全局最差实例也满足上述性质.

Corollary 2.1. 对任意 k 为整数, 在物品数为 k 个的所有实例中必然存在最差实例.

Proof. 该证明与引理2.3类似. □

Lemma 2.4. 对于任意一个实例 A, $A, A\cup\{1\}, A\cup 2*\{1\}, \ldots$ 的所有实例中必然存在最差实例.

Proof. 由引理2.2可以知道若实例 A 不满足引理的条件, 那么一定会有一个实例 A^i 使得比 A 差, 那么由这个构造过程可以知道当 $s > i$, A_i 应该会不优于 A^s. 并且实际上仅可能有 A^{i+1} 的目标值之比和他的相同. □

Remark 2.3. 这样构造出来的实例的物品数若超过 $4k$, 这样构造出来的实例的目标值之比会小于 $\frac{5}{4}$, 我们不用担心这种情况. 这也就是说, 对于这样的构造出来的最差实例我们不用担心物品数会太大. 同时假设这样构造出来的实例的物品数为 k_1, 他的目标值之比应该不大于物品数为 k_1 的所有实例中的最差实例.

Lemma 2.5. 对任意 k 为正整数, 在物品数不超过 k 个的所有实例中的最差实例 A 必然有以下不等式成立:

$$\frac{C^F(A)}{C^*(A)} \le \frac{n^F(A)}{n(A)} \tag{2.6}$$

Proof. 证明的思路是若不等式不成立, 可以找到一个新的实例 B，使得 $k(B) \le k(A)$，并且 $\frac{C^F(B)}{C^*(B)} \ge \frac{C^F(A)}{C^*(A)}$. 那么若这样的 B 存在, A 也就不是最差实例, 与假设相矛盾.

若引理中的假设不成立, 那么

$$\frac{C^F(A)}{C^*(A)} > \frac{n^F(A)}{n^*(A)} \tag{2.7}$$

若该实例的物品可以同时放进一个箱子中, 那么此时 FFI 算法和最优算法的排放方式相同, $\frac{C^F(A)}{C^*(A)} = 1$, 此时 $\frac{C^F(A)}{C^*(A)}$ 达到了最小值, 这种情况是平凡的. 注

意到对于实例 A 的 FFI 算法，重量最大的物品必定放在最后一个箱子中, 设这个最重的物品为 I, 若我们构造新实例 B 为 $A/\{I\}$, 那么实例 B 的物品数不大于实例 A 的物品数, 并且 $C^F(A/\{I\}) = C^F(A) - n^F(A)$, 而对最优算法可以把最后一个箱子中任一物品与 I 对调, 之后删去 I, 可知这也是 $A/\{I\}$ 的一个可行排序. 所以我们有 $C^*(A/\{I\}) \leq C^*(A) - n(A)$.

那么对于实例 A 有

$$\frac{C^F(A/\{I\})}{C^*(A/\{I\})} \geq \frac{C^F(A) - n^F(A)}{C^*(A) - n(A)} > \frac{C^F(A)}{C^*(A)} \tag{2.8}$$

与最差实例假设矛盾，所以原假设对任意 k 都不成立。 □

类似上述结论我们有以下引理:

Lemma 2.6. 对任意 k 为正整数, 在物品数等于 k 个的所有实例中的最差实例 A 必然有以下不等式成立:

$$\frac{C^F(A)}{C^*(A)} \leq \frac{n^F(A) - 1}{n(A) - 1} \tag{2.9}$$

Proof. 证明的思路是若不等式不成立, 可以找到一个新的实例 B，使得 $k(B) = k(A)$，并且 $\frac{C^F(B)}{C^*(B)} \geq \frac{C^F(A)}{C^*(A)}$. 那么若这样的 B 存在, A 也就不是最差实例, 与假设相矛盾.

若引理中的假设不成立, 那么

$$\frac{C^F(A)}{C^*(A)} > \frac{n^F(A) - 1}{n^*(A) - 1} \tag{2.10}$$

下面先考虑对于实例 A 的最优算法和 FFI 算法第一个箱子中都没有放满物体的情况, 假设此时最优算法和 FFI 算法重第一个箱子中分别空了 ϵ_1, ϵ_2 的重量, 假设实例中最轻的物体重量为 ϵ_3. 此时如下构造实例 B, 去掉实例 A 中最重的物体, 之后加上一个重量为 $\min\{\epsilon_1, \epsilon_2, \epsilon_3\}$ 的物体. 假如最重的物体在最优算法中放在最后一个箱子中我们可以直接去掉最重的物体之后加入一个重量为 $\min\{\epsilon_1, \epsilon_2, \epsilon_3\}$ 的物体. 假如最重的物体在最优算法中不在最后一个箱子中, 我们可以先在最优算法去掉最重物体, 再把最后一个箱子中的一个物体换到最重物体原来在的箱子中, 之后将新加的物体放在第一个箱子中, 这样构成了一个可行算法. 对于 FFI 算法, 这个新加的物体会放在第一个箱子中并且不影响之后箱子中物体的排序方式, 最重的物体可以假设他按照算法排在最后一个箱

子中. 此时实例 B 的物体数量与 A 相同并且有 $C^*(B) \le C^*(A) - n(A) + 1$, $C^F(B) = C^F(A) - n^F(A) + 1$. 那么

$$\frac{C^F(B)}{C^*(B)} \ge \frac{C^F(A) - n^F(A) + 1}{C^*(A) - n(A) + 1} > \frac{C^F(A)}{C^*(A)} \tag{2.11}$$

与 A 是最差实例相矛盾.

再考虑实例 A 的最优算法或者 FFI 算法的第一个箱子中放满了物体的情况. 假设实例的 FFI 算法排序中第 i 个箱子中最轻的物体重量分别为 δ_i, FFI 算法的第 i 个箱子中有 num_i 个物体, FFI 算法的第 i 个箱子中空的重量为 $space_i$, 那么一定有对所有的 i: $space_i < \delta_{i+1}$. 我们考虑到可以将实例 A 中 FFI 算法第一个箱子中每个物体都减少 $\min\{\frac{\delta_1}{2}, \frac{\delta_2 - space_1}{4 \cdot num_1}\}$ 重量, 此时令 $space'_1 = space_1 - \min\{\frac{\delta_1}{2}, \frac{\delta_2 - space_1}{4 \cdot num_1}\} * num_1$, 第 i 个箱子中的每一个物体都减少 $\min\{\frac{\delta_i - space'_{i-1}}{2}, \frac{\delta_{i+1} - space_i}{4 \cdot num_i}\}$ 的重量, 之后 $space'_i = space_i + \min\{\frac{\delta_i - space'_{i-1}}{2}, \frac{\delta_{i+1} - space_i}{4 \cdot num_i}\} \cdot num_i$, 其中 $2 \le i \le n^F - 1$. 当 $i = n^F$ 时, 把第 i 个箱子中所有物体减少 $\frac{\delta_i - space'_{i-1}}{2}$ 重量. 这种递归可以知道每一次都使得 FFI 算法的排序方式不变 (这里的不变是指每一个物体对应第若干个箱子的顺序不变) 并且每一个物体都变小了. 此时化归到第一种情况. □

Remark 2.4. 实际上我们还可以增加一种重量为 0 的物体来证明这个性质. 因为重量为 0 的物体必然要放在第一个箱子中, 并且无论第一个箱子中有没有放满都可以放进去. 另一方面, 对于最坏的情况我们可以假设这种重量为 0 的物品并不存在于实例中. 所以虽然理论上这个使得实例可以出现的情况增加了, 但他并不影响上界的证明. 用这种方法可以直接在第一个箱子中增加一个物体, 这样在上述证明中我们并不需要担心第一个箱子中放满了物品的情况.

Lemma 2.7. 对于任一 k 为正整数, 实例 B 是恰有 k 个物体的所有实例中的最差实例, 实例 A 定义为 $B, B \cup \{1\}, B \cup 2 * \{1\}, \ldots$ 实例中的最差实例. 则实例 A 会满足

$$\frac{n^F(A) + 1}{n(A) + 1} \le \frac{C^F(A)}{C^*(A)} \le \frac{n^F(A) - 1}{n(A) - 1} \tag{2.12}$$

Proof. 若实例 B 是实例 A, 那么由引理2.6以及条件 A 是所有物品数为 k 的实例中的最差实例可以知道右边不等式一定成立. 所以若不成立必然是左边的不

等式不成立. 由引理2.2的证明, 当左边不等式不成立时, 存在一个实例 B^1 使得 $\frac{C^F(B^1)}{C^*(B^1)} > \frac{C^F(A)}{C^*(A)}$. 实际上 B^1 也即 $B \cup \{1\}$. 这和 B 是最差实例相矛盾.

假设实例 B 不是实例 A, 那么同样的我们可以假设最差实例 A 可以表为 B^i. 那么一方面由引理2.6 的证明可以知道不等式左边成立. 又因为 B^i 的目标值之比相比 B^{i-1} 的要大, 故我们有

$$\frac{C^F(B^{i-1})}{C^*(B^{i-1})} < \frac{n^F(B^{i-1})+1}{n(B^{i-1})+1} \tag{2.13}$$

由于 B^i 是 $B^{i-1} \cup \{1\}$, 所以我们只需要证明

$$\frac{C^F(B^{i-1})+n^F(B^{i-1})+1}{C^*(B^{i-1})+n(B^{i-1})+1} \leq \frac{n^F(B^i)}{n(B^i)} \tag{2.14}$$

注意到由于我们有 $n^F(B^{i-1}) > n(B^{i-1})$, 所以用上面的式子即可证明右边不等式. □

Remark 2.5. 我们不能够得到关于至少 k 个物品的所有实例的最差实例的结论, 因为这样的最差实例不一定有定义. 比如说可能当实例中的物体数趋向于无穷的时候目标值之比会趋向于一个最差值, 这个时候就不存在最差实例.

Remark 2.6. 我们没有对实例的物体的大小进行任何的限制, 所以导致了我们在讨论恰有 k 个物体中的最差实例时, 由 k 趋于 ∞ 并不能导致 $n \to \infty$, 但是实际上我们是可以对我们讨论的实例的物品大小做出一定的限制的, 而做出限制之后上述结论仍然成立, 同时有重量不小于某一个固定的数的物品会趋向于无穷, 此时就可以说明 n 会随着 k 趋向于无穷而趋于无穷.

为了说明这种情况, 我们引入一个引理

Lemma 2.8. 考虑对于任意物品重量不超过 $\frac{1}{k}$ 的实例, FF 算法的箱子数与其他任意一种排列方式的箱子数之比不超过 $\frac{k+1}{k}$.

Proof. Gyorgy Dosa 证明了这个结论.[2] □

Corollary 2.2. 对于实例 A 去掉重量最大的 d 个物体得到实例 B, 实例 B 的 FFI 算法的箱子数 $n^F(B)$ 与最优算法的箱子数 $n(B)$ 有关系:

$$n(B) \geq \frac{2}{3} n^F(B) \tag{2.15}$$

Proof. 注意到由于 d 的定义, 我们会去掉所有的重量大于 $\frac{1}{2}$ 的物体, 所以此时由引理2.8可以知道这个结论是显然的. □

Lemma 2.9. 假设存在一个单调趋向于 0 的数列 $\{1, \epsilon_1, \epsilon_2, \ldots, \}$, 其中 $\epsilon_i > 0, \forall i$. 我们考虑以下的情况: 对于所有物品数大于 k, 且其中重量不小于 ϵ_s 的物体数量不小于 $\log(k+1)$ 的所有实例中的最差实例, 当 $k \to \infty$, 对于最差实例 A 有

$$\frac{C^F(A)}{C^*(A)} = \frac{n^F(A)}{n(A)} \tag{2.16}$$

注意到这里的 ϵ_s 是独立于 k 的, 也即这个值不随着 k 的增大而增大, 是个固定值.

Proof. 注意到这个条件并不影响证明过程, 我们可以用类似于上述引理的证明方法得到类似于不等式2.12的结论. 之后注意到当 k 趋向于无穷的时候, 由于有至少 $\log(k+1)$ 的物品的重量不小于 ϵ_s, 所以 n 也会趋向于无穷. 故

$$\lim_{n(A)\to\infty} \frac{n^F(A)+1}{n(A)+1} \le \frac{C^F(A)}{C^*(A)} \le \lim_{n(A)\to\infty} \frac{n^F(A)-1}{n(A)-1} \Longrightarrow \frac{C^F(A)}{C^*(A)} = \frac{n^F(A)}{n(A)} \tag{2.17}$$

□

这个引理表达的是当实例中大于一个固定数值的物品数量以一种方式趋于无穷 (可能慢于 k 增加的速度), 此时随着总物品数的增加实例的总重量也会趋于无穷, 所以 n 也就会趋向于无穷.

Remark 2.7. 实际上对于上述的对于上面的引理, 当 ϵ_s 足够小并且 k 足够大的时候, 所有物品数大于 k 的实例中的最差实例一定不是一个只有不超过 $\log(k+1)$ 数量的重量不小于 ϵ_s 的物体的实例.

为了下面阐述简便, 我们假设 $\epsilon_s = \frac{1}{s}$.

这是因为我们可以考虑实例 B 构造为: 实例 A 去掉所有重量大于 $\frac{1}{s}$ 的物体. 那么由于引理2.8的结论有 $\frac{n^F(B)}{n(B)} \le \frac{s+1}{s}$. 我们使用反证法, 若 A 是最差实例, 那么 $\frac{C^F(A)}{C^*(A)} \le \frac{n^F(A)-1}{n(A)-1}$. 注意到 $n(A) \ge n(B)$, $n^F(A) \le n^F(B) + \log(k+1)$, 所以 $\frac{n^F(A)-1}{n(A)-1} \le \frac{n^F(B)+\log(k+1)-1}{n(B)-1} \le \frac{n(B)\cdot\frac{s+1}{s}+\log(k+1)-1}{n(B)-1} \le \frac{n(B)\cdot\frac{s+1}{s}+\log(k+1)-1}{n(B)-1}$. 若 $n(B) = O((\log(k+1))^2)$, 此时 $\frac{n(B)\cdot\frac{s+1}{s}+\log(k+1)-1}{n(B)-1} \to 1$, 这种情况是平凡的. 若 $n(B) = o((\log(k+1))^2)$, 此时 $C^F(A) \le C^F(B) + (n^F(B)+1) + \cdots + (n^F(B) + \lceil\log(k+1)\rceil) =$

$C^F(B)+n^F(B)\cdot\log(k+1)+O((\log(k+1))^2)=C^F(B)+O((\log(k+1))^3)$, 注意到 $C^F(B)$ 与 $C^*(B)$ 都是 $O(k)$ 的, 所以我们有 $\frac{C^F(A)}{C^F(A)} \le \frac{C^F(B)+O((\log(k+1))^3)}{C^*(B)} \to \frac{C^F(B)}{C^*(B)}$. 所以此时这些实例的 $\frac{C^F(A)}{C^*(A)}$ 会趋向于 B 的 $\frac{C^F(B)}{C^*(B)}$. 而对于后一种情况而言是平凡的, 当 B 的是最差实例的时候, $\frac{C^F(B)}{C^*(B)} \le \frac{s+1}{s}$.

这说明了我们只需要考虑当物体数趋向于无穷, 箱子总数同时也趋向于无穷的情况.

对于物体数足够多的实例, 我们可以只需要考虑引理2.7中的最差实例即可.

下面只要说明对于任一物品总重量趋于无穷的实例 A 满足

$$\frac{C^F(A)}{C^*(A)} = r(A) \tag{2.18}$$

的条件下成立我们要证明的 Claim1即可.

以上的引理是我们的准备工作, 下面我们开始证明 Claim 1.

2.2 主要证明

下面的 d,t 都是对于实例 A 定义的. 这是因为实例 B 去掉了所有重量大于 $\frac{1}{2}$ 的箱子, 对于实例 B 定义 d 没有意义.

注意到事实：任何两个重量大于 $\frac{1}{2}$ 的物品不能放在同一个箱子中.

Lemma 2.10. 由实例 A 可以去掉 d 个最重的物体得到实例 B, 根据 d 的定义可以知道所有重量大于 $\frac{1}{2}$ 的物体被去除了. 实例 B 的 FFI 算法的箱子数为 $n^F(A)-d$ 或者 $n^F(A)-d+1$, 这取决于这 d 个物品中最轻的那个在 FFI 算法中是否和别的物体一起放在一个箱子中. 若在一个箱子中为 $n^F(A)-d+1$, 若不在一个箱子中则为 $n^F(A)-d$. 实例 B 最优算法的箱子数不大于 $n(A)-t-\lfloor\frac{d-t}{2}\rfloor$.

Proof. 这里的结论对于实例 B 的 FFI 算法是显然的, 因为当实例 A 去掉重量大于 $\frac{1}{2}$ 的物品的时候, 这些物品必然放在最后的 d 个不同箱子中.

对于实例 B 的最优算法, 先考虑去掉实例 A 中 t 个最重的物品 (这里用到了 $t \le d$). 我们可以假设去掉的是最后 t 个箱子中的物品. 因为若不是, 我们可以将这些物品换到最后的箱子中来, 这一定是可行的方案. 之后我们再将剩下的 $(d-t)$ 个物品去掉. 由我们的定义可以知道这些物品中的任意两个不能放进同

一个箱子中, 并且任意两个和他们放在同一个箱子中的物品都可以单独拿出来放在同一个箱子中. 那么这构成了一个可行算法. 这样减少的值也是最多的. 注意两个实例之间的转变是建立在增加物体之上的. 增加物体之后的实例的最优算法的目标值应该不超过增加之前最优算法的目标值值加 n. 当取到等号的时候对应的一定是增加之前的最优算法. □

综合引理2.10以及推论2.2我们有:

Lemma 2.11.

$$n(B) \geq \frac{2}{3}(n^F(A) - d) \tag{2.19}$$

Proof. 我们只需要注意到对于实例 B 而言有

$$n^F(B) \leq n^F(A) - d \tag{2.20}$$

并且实例 B 中的所有物体重量均不超过 $\frac{1}{2}$. 所以可以用引理2.7. □

我们分情况讨论: $\frac{d-t}{2}$ 为整数或者不为整数情况.

2.2.1 $\frac{d-t}{2}$ 为整数的情况

Lemma 2.12. 对于任意实例 A 有以下关系:

$$\frac{d}{6} - \frac{t}{2} \geq (\frac{2r}{3} - 1)n(A) \tag{2.21}$$

Proof. 一方面由于引理2.19我们有:

$$\frac{2}{3}(n^F(A) - d) \leq n(B)$$

另一方面由于已知 $\frac{d-t}{2}$ 为整数, 所以有

$$n(B) \leq n(A) - t - \frac{d-t}{2}$$

故综合上面两个式子我们有

$$\frac{d}{6} - \frac{t}{2} \geq (\frac{2r}{3} - 1)n(A)$$

□

如果对于实例有

$$d \leq 3t \tag{2.22}$$

那么根据引理2.12, 我们有

$$(\frac{2r}{3} - 1)n \leq 0 \Longrightarrow r \leq \frac{3}{2} \tag{2.23}$$

那么 Claim1成立.

我们给出类似于不等式2.22的条件来帮助理解:

Claim 2. 关于 d 与 t 有以下关系:

$$r \geq \frac{d+3}{2t+2} \tag{2.24}$$

同时该关系也可以写成

$$t \geq \frac{d+3-2r}{2r} \tag{2.25}$$

若 Claim2成立, 那么:

$$\begin{aligned}
&\frac{d}{6} - \frac{t}{2} \geq (\frac{2r}{3} - 1)n \Longrightarrow \frac{n}{6} - \frac{nt}{2d} \geq (\frac{2r}{3} - 1)n \Longrightarrow \frac{7}{6} - \frac{t}{2d} \geq \frac{2r}{3} \\
&\Longrightarrow \frac{7}{6} - \frac{d+3-2r}{4dr} \geq \frac{2r}{3} \Longrightarrow 8dr^2 - (14d+6)r + (3d+9) \leq 0 \\
&\Longrightarrow r \leq \frac{3}{2}
\end{aligned}$$

第一步推导用到 $\frac{d}{6} - \frac{t}{2} \geq 0$ 与 $d \leq n$, 第三步推导用到 Claim2.

那么我们可以知道的是当 Claim2 成立的时候我们有 Claim1成立.

若 $d \leq 3t$ 与 Claim2都不成立.

Proof. 若实例 A 的 FFI 算法中的倒数第 d 个箱子中有 k 个物品, 其中 $k \geq 2$, 那么构造新实例 B 中重量不超过 $\frac{1}{2}$ 的物品中重量最大的那个提高到 1, 其他物品不变, 那么有新实例的最优算法算法的目标值不大于 $C^* + t + 1$, 而 FFI 算法的目标值为 $C^F + d + 1$. 若 $\frac{C^F(B)}{C^*(B)} \leq \frac{C^F(A)}{C^*(A)}$, 那么

$$\frac{C^F(B)}{C^*(B)} \geq \frac{C^F(A)+d+1}{C^*(A)+t+1} \Longrightarrow \frac{C^F(A)}{C^*(A)} \geq \frac{d+1}{t+1} \tag{2.26}$$

注意到 $\frac{C^F(A)}{C^*(A)} = r \le 1.7$, $d > 3t$. 所以有 $\frac{3t+1}{t+1} \le 1.7 \Longrightarrow t = 0$，那么因此有 d=0，此时实例中没有重量大于 $\frac{1}{2}$ 的物体. 显然 Claim 欲证结论成立. 那么 $\frac{C^F(B)}{C^*(B)} > \frac{C^F(A)}{C^*(A)}$. 之后我们考虑 B 而非讨论 A. 只要证明 $\frac{C^F(B)}{C^*(B)} \le 1.5$ 成立结论即可.

若实例 A 的 FFI 算法中倒数第 d 个箱子中只有一个物品.

注意到实例 A 的 FFI 算法倒数第 (d +1) 个箱子中的物品一定重量都不大于 $\frac{1}{2}$。我们分这个箱子里物品个数 k 的不同情况进行讨论。

若 k 等于 1, 那么此时在倒数第 d 个箱子中的物体的重量必然大于 $\frac{1}{2}$. 那么注意到这个重量小于 $\frac{1}{2}$ 的物体不能和重量大于 $\frac{1}{2}$ 的物体放在一个箱子中，根据 d 的定义知道矛盾. 因为这个重量小于 $\frac{1}{2}$ 应该也包含在 d 个物体的集合中.

若实例 A 的 FFI 算法中倒数第 d+1 个箱子中仅有二个物体. 我们考虑构造实例 B: 把该箱子中 2 个物品全部去掉，再去掉任意 2 个重量大于 $\frac{1}{2}$ 的物品. 此时 B 的 FFI 算法的排列顺序应当保持, 并且我们可以关于 A 的最优算法进行新实例的一个算法构造. 那么 $C^F(B) = C^F(A) - 4n(A)r(A) + d + 3$, $C^*(B) \le C^*(A) - 4n(A) + 2t + 2$. 若 $\frac{C^F(B)}{C^*(B)} \le \frac{C^F(A)}{C^*(A)}$, 那么

$$\frac{C^F(A)}{C^*(A)} \ge \frac{C^F(A) - 4n(A)r(A) + d + 3}{C^*(A) - 4n(A) + 2t + 2} \Longrightarrow \frac{C^F(A)}{C^*(A)} \le \frac{-4n(A)r(A) + d + 3}{-4n(A) + 2t + 2}$$
$$\Longleftrightarrow r(A) \le \frac{-4n(A)r(A) + d + 3}{-4n(A) + 2t + 2} \Longrightarrow r(A) \ge \frac{d+3}{2t+2}$$

此时 Claim2成立, 所以定理得证. 若 $\frac{C^F(B)}{C^*(B)} > \frac{C^F(A)}{C^*(A)}$, 那么我们可以只证明 $\frac{C^F(B)}{C^*(B)} \le 1.5$ 成立结论即可.

若实例 A 的 FFI 算法中倒数第 d+1 个箱子中有 k 个物体 ($k \ge 3$, 那么这种情况下倒数第 d+1 个箱子中最轻的那个物体的重量必定不大于 $\frac{1}{k}$。分两种情况讨论

1. 构造新的实例 B: 删掉倒数第 d+1 个箱子中重量最小的物品. 若对于实例 B 其余箱子中物品排序不变，那么 $C^F(B) = C^F(A) - n^F(A) + d$, $C^*(B) = C^*(A) - n^*(A) + t$. 若 $\frac{C^F(B)}{C^*(B)} \le \frac{C^F(A)}{C^*(A)}$, 那么有

$$\frac{C^F(A)}{C^*(A)} \ge \frac{C^F(A) - n^F(A) + d}{C^*(A) - n^*(A) + t} \Longrightarrow r(A) \ge \frac{d}{t} \tag{2.27}$$

那么由于 $r \le 1.7$, 故 $d \le 3t$ 不成立. 若 $\frac{C^F(B)}{C^*(B)} > \frac{C^F(A)}{C^*(A)}$, 那么我们可以只证明 $\frac{C^F(B)}{C^*(B)} \le 1.5$ 成立结论即可.

2. 若实例 B 中 FFI 算法其他物体的排序会改变, 注意到倒数第 (d+1) 个箱子中的物体不可能放到前面的箱子中, 故必然是实例 A 中倒数第 d 个箱子中的物体被放到倒数第 d+1 个箱子中.

我们考虑构造实例 B: 去掉实例 A 中倒数第 d+1 个箱子中全部 k 个物体, $C^F(B) = C^F(A) - k \cdot n^F(A) + (k-1)d$, $C^*(B) \leq C^*(A) - k \cdot n(A) + kt + k(k-1)/2$. 若 $\frac{C^F(B)}{C^*(B)} \leq \frac{C^F(A)}{C^*(A)}$, 那么

$$\frac{C^F(B)}{C^*(B)} \geq \frac{C^F(A) - k \cdot n^F(A) + (k-1)d}{C^*(A) - k \cdot n(A) + kt + k(k-1)/2} \Longrightarrow \frac{C^F(A)}{C^*(A)} \leq \frac{-kn^F(A) + (k-1)d}{-kn(A) + kt + k(k-1)/2}$$
$$\Longleftrightarrow r \leq \frac{-kn^F(A) + (k-1)d}{-kn(A) + kt + k(k-1)/2} \Longrightarrow d \leq \frac{ktr}{k-1} + \frac{kr}{2} \tag{2.28}$$

注意到我们由 Claim2不成立有 $d > 2tr + 2r - 3$. 结合上面的式子有 $t < k$. 那么此时 $d \leq \frac{ktr}{k-1} + \frac{kr}{2} < \frac{k^2r}{k-1} + \frac{kr}{2} \leq 4.5 * 1.5 * k$. 注意到我们考虑 $n \to \infty$, 而 d 是有限的, 所以由引理2.12可以知道 $r \leq 1.5$.

若 $\frac{C^F(B)}{C^*(B)} > \frac{C^F(A)}{C^*(A)}$, 那么我们可以只证明 $\frac{C^F(B)}{C^*(B)} \leq 1.5$ 成立结论即可.

注意到, 上面的构造过程会使得重量小于 $\frac{1}{2}$ 的物品减少, 所以一定会有一个极限. □

2.2.2 $\frac{d-t-1}{2}$ 为整数的情况

Proof. 我们分两种情况进行讨论：对于实例 A 去掉所以重量大于 $\frac{1}{2}$ 的箱子构造实例 B 的情况下, 对于实例 B 的 FFI 算法有多少个箱子, 若实例 B 的 FFI 算法的箱子数为 $n(A)r - d + 1$, 那么可以化归到第一种的情况, 因为之后的引理2.12依然满足, 其他的证明过程和前面的证明方法完全相同.

若箱子数为 $n(A)r - d$，那么分两种情况进行讨论。

1. 若实例 B 的 FFI 算法中最后一个箱子中空余的重量不超过 $\frac{1}{2}$. 我们将最后一个箱子中空余的重量设为 k. 我们构造实例 C: 在实例 B 的情况下加一个重量为 $\frac{1}{2} - \frac{k}{2}$ 的物体. 首先这个物体必然在 FFI 算法的倒数第一个箱子里或者倒数第二个箱子里, 因为在实例 B 中最后一个箱子至少有两个物体, 而总重量为 1-k, 所以最轻的物体的重量至少为 $\frac{1-k}{2}$. 并且这个物体必然会让实例 C 的 FFI 算法相比较实例 B 的 FFI 算法多一个箱子. 那么可以知道对于实例 C, 最优算法的

箱子数不超过 $n(A)-\frac{d+t-1}{2}$ 个，而 FFI 算法的箱子数为 n(A)-d+1 个，而实例中所有的物品的重量都不超过 $\frac{1}{2}$，所以引理2.12依旧成立。所以我们化归到原来的证明中。

2. 实例 B 中的 FFI 算法中最后一个箱子的空余的重量超过 $\frac{1}{2}$. 那么此时在实例 B 中最后一个箱子中的物品重量之和不超过 $\frac{1}{2}$. 我们分情况进行讨论。

(1) 若实例 B 的最后一个箱子中仅有一个物品. 此时由前面的证明可以知道有 $\frac{d}{t} \leq r \leq 1.7$. 这与 $d > 3t$ 相矛盾.

(2) 若该箱子中有 k 个物品 $(k \geq 2)$. 那么其中重量最小的那个物品的重量不超过 $\frac{1}{2k}$. 构造实例 C: 在实例 B 的基础上去掉最后一个箱子中除了最轻的物体之外的所有的物体. 此时实例 C 的 FFI 算法的箱子数减少了 d 个, 最优算法的箱子数至少减少了 $\frac{d+t+1}{2}$ 个. 那么由 HH 文章的引理：$\frac{n^F(A)-d}{n(A)-\frac{t+d-1}{2}} \leq \frac{2k+1}{2k}$, 从而可以得到 $\frac{n^F(A)-d}{n(A)-\frac{t+d-1}{2}} \leq \frac{5}{4}$. 只要说明 $\frac{5}{4}n-\frac{5}{8}(t+d-1) \leq \frac{3}{2}n-\frac{3}{4}(t+d)$ 即可得到式子2.12, 从而可以化归到第一种情况中.

而上述式子等价于 $\frac{1}{4}n \geq \frac{1}{8}(t+d)+\frac{5}{8}$. 注意到 $n \geq d$, 我们只要说明 $n-t \geq 5$ 即可证得结论. 注意到若 $n-t<5$, 容易由 $\frac{2}{3}(n-d) \leq n-[\frac{d+t}{2}]$ 得到结论. □

所以我们证明了 FFI 算法对于总和型装箱问题的渐近性能比上限应该不超过 1.5.

3 参考文献

[1] EPSTEIN L, JOHNSON D S, LEVIN A. Min-Sum Bin Packing[J]. Journal of Combinatorial Optimization, 2018, 36(4): 1–24.

[2] DOSA G. The tight absolute bound of First Fit in the parameterized case.[J]. Theoretical Computer Science, 2015, 596(C): 149–154.