

# Задача упаковки в контейнеры

**Дано:** множество предметов  $L = \{1, \dots, n\}$  и их веса  $w_i \in (0, 1)$ ,  $i \in L$ .

**Найти:** разбиение множества  $L$  на минимальное число  $m$  подмножеств

$B_1, B_2, \dots, B_m$  такое, что

$$\sum_{i \in B_j} w_i \leq 1, \text{ для всех } 1 \leq j \leq m.$$

Множества  $B_j$  называют контейнерами.

Требуется упаковать предметы в минимальное число контейнеров.

Задача NP-трудна и часто возникает в приложениях.

## Алгоритм «Следующий подходящий» (NF)

В произвольном порядке упаковываем предметы по следующему правилу. Первый предмет помещаем в первый контейнер.

На  $k$ -м шаге пытаемся поместить  $k$ -й предмет в текущий контейнер.

Если предмет входит, то помещаем его и переходим к следующему шагу, иначе помещаем предмет в новый контейнер.

$T = O(n)$ ,  $P = O(1)$ , если не считать место для исходных данных.

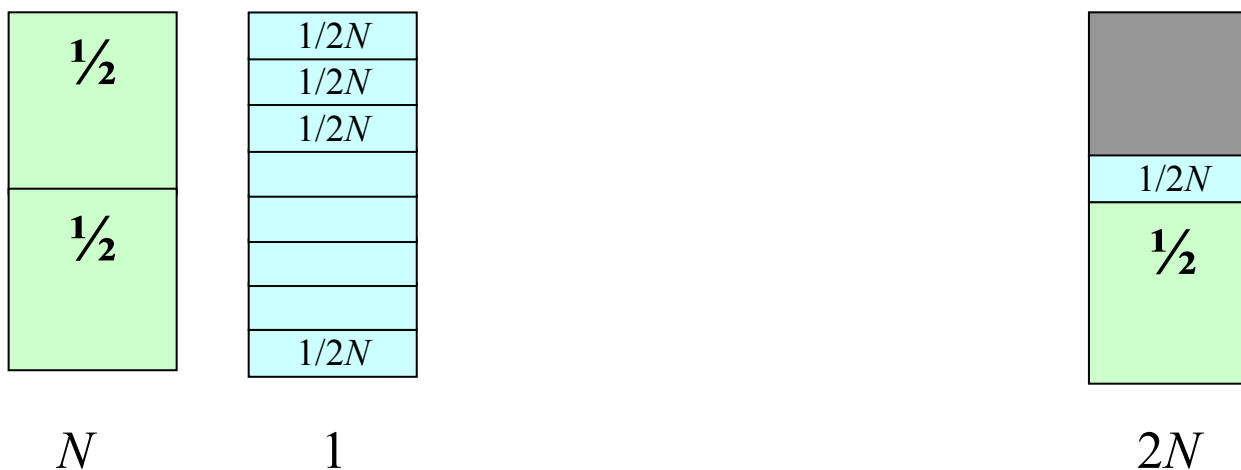
**Теорема.**  $NF(L) \leq 2OPT(L)$ .

**Доказательство.** Пусть  $W = \sum_{i \in L} w_i$ . Так как любые два последовательных кон-

тейнера содержат предметы суммарным весом не меньше единицы, то  $NF(L) < 2\lceil W \rceil$ . Кроме того,  $OPT(L) \geq \lceil W \rceil$ , откуда и следует требуемое. ■

## Пример.

$L = \{\frac{1}{2}, \frac{1}{2N}, \frac{1}{2}, \frac{1}{2N}, \dots, \frac{1}{2}, \frac{1}{2N}\}$ . Всего  $4N$  предметов.



$$OPT(L) = N + 1$$

$$NF(L) = 2N$$

**Замечание.**  $NF(L) \leq 2OPT(L) - 1$  для всех  $L$ .

Пусть алгоритм  $A$  для множества  $L$  порождает  $A(L)$  контейнеров и

$$R_A(L) \equiv A(L) / OPT(L).$$

Для задачи на минимум гарантированная относительная точность  $R_A$  для алгоритма  $A$  определяется как

$$R_A \equiv \inf \{r \geq 1 \mid R_A(L) \leq r \text{ для всех } L\}.$$

**Определение.** Асимптотическая гарантированная относительная точность  $R_A^\infty$  для алгоритма  $A$  определяется как

$$R_A^\infty \equiv \inf \{r \geq 1 \mid \exists N > 0 \text{ такое, что } R_A(L) \leq r \text{ для всех } L \text{ с } OPT(L) \geq N\}.$$

## Алгоритм «Первый подходящий» (FF)

В произвольном порядке упаковываем предметы по следующему правилу. Первый предмет помещаем в первый контейнер.

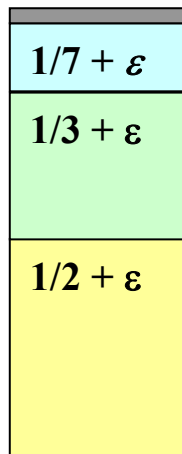
На  $k$ -м шаге находим контейнер с наименьшим номером, куда помещается  $k$ -й предмет, и помещаем его туда. Если такого контейнера нет, то берем новый пустой контейнер и помещаем предмет в него.

$$T = O(n^2), \quad \Pi = O(n).$$

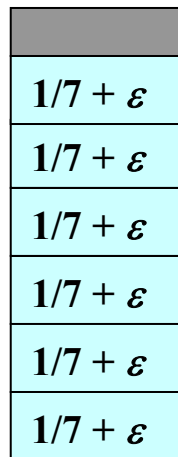
**Теорема.**  $FF(L) \leq \lceil \frac{17}{10} OPT(L) + 1 \rceil$  для всех  $L$  и существуют примеры со сколь угодно большими значениями  $OPT$ , для которых  $FF(L) \geq \lceil \frac{17}{10} OPT(L) - 1 \rceil$ .  
(Без доказательства)

Пример.

$$L = \{1, \dots, 18m\} \quad w_i = \begin{cases} \frac{1}{7} + \varepsilon, & 1 \leq i \leq 6m \\ \frac{1}{3} + \varepsilon, & 6m < i \leq 12m \\ \frac{1}{2} + \varepsilon, & 12m < i \leq 18m \end{cases}$$



$$OPT(L) = 6m$$



$m$

$3m$

$6m$

$$FF(L) = 10m$$

$$\frac{FF(L)}{OPT(L)} = \frac{5}{3}$$

## Алгоритм «Наилучший подходящий» (BF)

В произвольном порядке упаковываем предметы по следующему правилу. Первый предмет помещаем в первый контейнер.

На  $k$ -м шаге размещаем  $k$ -й предмет. Находим частично заполненные контейнеры, где достаточно для него свободного места и выбираем среди них наиболее заполненный. Если таких нет, то берем новый пустой контейнер и помещаем  $k$ -й предмет в него.

$$T = O(n^2), \quad \Pi = O(n).$$

**Теорема.**  $R_{BF} = R_{FF}$ ,  $R_{BF}^{\infty} = R_{FF}^{\infty}$  и существуют примеры со сколь угодно большими значениями  $OPT(L)$ , для которых  $BF(L) = 4/3 FF(L)$  и  $FF(L) = 3/2 BF(L)$ .

(Без доказательства)

## Алгоритмы типа On-line

Предметы поступают в непредсказуемом порядке. Требуется упаковать их в минимальное число контейнеров. Упакованный предмет нельзя перемещать в другой контейнер. Место для предварительного хранения предметов отсутствует.

Алгоритмы NF, FF, BF являются **On-line алгоритмами**.

**Теорема.** Для любого On-line алгоритма  $A$  справедливо неравенство  $R_A^\infty > 1.5$   
(Без доказательства)



# Алгоритмы с ограниченным доступом к контейнерам

On-line алгоритм называют **алгоритмом с ограниченным доступом к контейнерам**, если на каждом шаге алгоритм имеет возможность помещать предметы только в один из  $K$  контейнеров ( $K$  — const). Эти контейнеры называются **открытыми**. Если контейнер закрыли, то он уже не открывается (например, отправляется потребителю). Прежде чем добавить пустой открытый контейнер, нужно закрыть один из  $K$  открытых контейнеров.

Алгоритм NF — пример для  $K = 1$ .

## Правила для выбора контейнера

1. Закрыть контейнер с наименьшим номером
2. Закрыть самый заполненный контейнер.

## Примеры алгоритмов с ограниченным доступом

$FF_1$  — алгоритм FF с правилом 1.

$FF_2$  — алгоритм FF с правилом 2.

$BF_1$  — алгоритм BF с правилом 1.

$BF_2$  — алгоритм BF с правилом 2.

**Теорема.** Для любого  $K \geq 2$

$$1) R_{FF_1}^{\infty} = R_{FF_2}^{\infty} = 1,7 + \frac{3}{10(K-1)}.$$

$$2) R_{BF_1}^{\infty} = 1,7 + \frac{3}{10K}.$$

$$3) R_{BF_2}^{\infty} = 1,7.$$

4) Для любого алгоритма  $A$  с ограниченным доступом к контейнерам

$$R_A^{\infty} \geq 1,69103...$$

## Алгоритм «Первый подходящий с упорядочиванием» (FFD)

- Сортируем предметы по невозрастанию весов

$$w_1 \geq w_2 \geq \dots \geq w_n$$

- Применяем алгоритм FF (BF).

**Теорема.**  $FFD(L) \leq \frac{11}{9} OPT(L) + 4$  для всех  $L$  и существуют примеры со сколь угодно большими значениями  $OPT(L)$ , для которых

$$FFD(L) \geq \frac{11}{9} OPT(L).$$

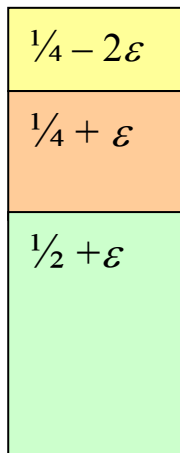
Кроме того  $R_{FFD}^{\infty} = R_{BFD}^{\infty} = \frac{11}{9} \approx 1.22$ .

(Без доказательства)

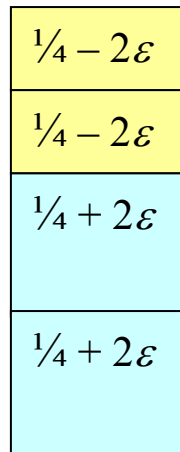
Пример.

$$L = \{1, \dots, 30m\}$$

$$w_i = \begin{cases} \frac{1}{2} + \varepsilon, & 1 \leq i \leq 6m \\ \frac{1}{4} + 2\varepsilon, & 6m < i \leq 12m \\ \frac{1}{4} + \varepsilon, & 12m < i \leq 18m \\ \frac{1}{4} - 2\varepsilon, & 18m < i \leq 30m \end{cases}$$

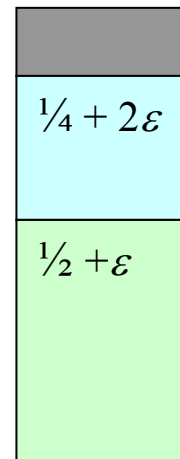


$6m$

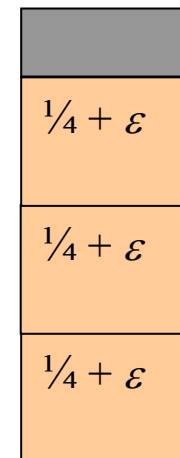


$3m$

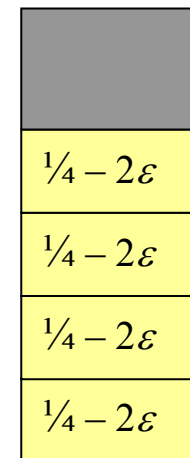
$$OPT(L) = 9m$$



$6m$



$2m$



$3m$

$$FFD(L) = 11m$$

## Асимптотические гарантированные оценки точности

Алгоритм	$T_A^*$	$R_A^\infty$	$R_A^\infty(1/2)$	$R_A^\infty(1/3)$	$R_A^\infty(1/4)$
<b>NF</b>	$O(n)$	2.000	2.000	1.500	1.333
<b>FF</b>	$O(n \log n)$	1.700	1.500	1.333	1.250
<b>BF</b>	$O(n \log n)$	1.700	1.500	1.333	1.250
<b>NFD</b>	$O(n \log n)$	1.691	1.424	1.302	1.234
<b>FFD</b>	$O(n \log n)$	1.222	1.183	1.183	1.150
<b>BFD</b>	$O(n \log n)$	1.222	1.183	1.183	1.150

$R_A^\infty(\alpha)$  — асимптотическая точность для примеров с весами предметов  $w_i \leq \alpha$ , для всех  $i \in L$ .

**Теорема.** Для любого  $\varepsilon \in (0,1]$  существует алгоритм  $A_\varepsilon$ , который находит упаковку с числом контейнеров не более  $(1 + 2\varepsilon) OPT + 1$ . Трудоемкость  $A_\varepsilon$  полиномиально зависит от  $n$ .

*(Без доказательства)*

### **Алгоритм $A_\varepsilon$**

1. Удалить предметы с весом менее  $\varepsilon$ .
2. Упорядочить оставшиеся предметы и разбить их на  $K = \lceil 1/\varepsilon^2 \rceil$  групп.
3. В каждой группе увеличить веса предметов до максимального веса в группе.
4. Найти оптимальную упаковку предметов, имеющих только  $K$  различных весов каждый из которых не менее  $\varepsilon$ .
5. Вернуть исходные веса предметов и применить алгоритм FF для предметов с весом менее  $\varepsilon$ .

## Негативный результат

**Теорема.** Для любого  $\varepsilon > 0$  существование приближенного полиномиального алгоритма  $A$  с гарантированной точностью  $R_A = \frac{3}{2} - \varepsilon$  влечет  $P = NP$ .

**Доказательство.** Пусть такой алгоритм  $A$  существует. Покажем, как с его помощью можно решить точно одну из NP-полных задач, а именно задачу о разбиении. Дано  $n$  неотрицательных чисел  $a_1, \dots, a_n$ . Можно ли разбить их на два подмножества так, чтобы сумма чисел в каждом подмножестве равнялась

$C = \frac{1}{2} \sum_{i=1}^n a_i$ ? Рассмотрим задачу упаковки в контейнеры с весами предметов

$w_i = a_i / C$ ,  $i = 1, \dots, n$ . Если их можно упаковать в два контейнера, ответ в задаче о разбиении — «ДА». Применим алгоритм  $A$  к задаче о контейнерах. Если  $OPT = 2$ , то алгоритм  $A$  тоже дает 2, иначе  $R_A \geq \frac{3}{2}$ , то есть алгоритм  $A$

точный. ■

## Нижние оценки

**Переменные задачи**

$$y_j = \begin{cases} 1, & \text{если используется контейнер } j \\ 0, & \text{в противном случае} \end{cases}$$
$$x_{ij} = \begin{cases} 1, & \text{если предмет } i \text{ помещен в контейнер } j \\ 0, & \text{в противном случае} \end{cases}$$

## Математическая модель

$$\min \sum_{j=1}^n y_j$$

при ограничениях

$$\sum_{i=1}^m w_i x_{ij} \leq y_j, \quad j = 1, \dots, n,$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n,$$

$$y_j, x_{ij} \in \{0, 1\} \quad i, j = 1, \dots, n.$$



# Релаксация линейного программирования

Заменим условие булевости переменных на условия:

$$0 \leq y_j \leq 1, \quad j = 1, \dots, n$$

$$0 \leq x_{ij} \leq 1, \quad i, j = 1, \dots, n.$$

Тогда одно из оптимальных решений имеет вид

$$x_{ij}^* = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}, \quad y_j^* = w_j,$$

что дает нижнюю оценку

$$H_0 = \left[ \sum_{i=1}^n w_i \right]$$

(предметы можно резать произвольным образом).

## Оценки Martello & Toth

Для примера  $L = \{1, \dots, n\}$ ,  $0 \leq w_i < 1$  и произвольного  $0 \leq \alpha \leq \frac{1}{2}$  положим

$L_1 = \{ i \in L \mid w_i > 1 - \alpha \}$  — крупные предметы

$L_2 = \{ i \in L \mid 1 - \alpha \geq w_i > \frac{1}{2} \}$  — средние предметы

$L_3 = \{ i \in L \mid \frac{1}{2} \geq w_i \geq \alpha \}$  — мелкие предметы

**Теорема.** Для любого  $0 \leq \alpha \leq \frac{1}{2}$  величина

$$H_1(\alpha) = |L_1| + |L_2| + \max \left( 0, \left\lceil \sum_{i \in L_3} w_i - (|L_2| - \sum_{i \in L_2} w_i) \right\rceil \right).$$

является нижней оценкой для  $OPT(L)$ .

**Доказательство.** Каждый предмет из множества  $L_1 \cup L_2$  требует отдельный контейнер. Поэтому в любом допустимом решении не менее  $|L_1| + |L_2|$  контейнеров. Предметы из множества  $L_3$  не лежат вместе с предметами из  $L_1$ . Значит, они лежат либо вместе с предметами из  $L_2$ , либо в отдельных контей-

нерах. В контейнерах для  $L_2$  осталось  $S = \left( |L_2| - \sum_{i \in L_2} w_i \right)$  свободного места.

Следовательно, для предметов из множества  $L_3$  требуется как минимум

$\left\lceil \sum_{i \in L_3} w_i - S \right\rceil$  отдельных контейнеров. ■

**Теорема.** Для любого  $0 \leq \alpha \leq \frac{1}{2}$  величина

$$H_2(\alpha) = |L_1| + |L_2| + \max \left\{ 0, \left\lceil \frac{|L_3| - \sum_{i \in L_2} \left\lfloor \frac{1-w_i}{\alpha} \right\rfloor}{\left\lfloor \frac{1}{\alpha} \right\rfloor} \right\rceil \right\}$$

является нижней оценкой для  $OPT(L)$ .

**Доказательство.** Заменим вес каждого предмета из множества  $L_3$  на  $\alpha$ . Тогда в один контейнер войдет  $\left\lfloor \frac{1}{\alpha} \right\rfloor$  предметов, и для множества  $L_3$  потребовалось бы  $\left\lceil \frac{|L_3|}{\left\lfloor \frac{1}{\alpha} \right\rfloor} \right\rceil$  дополнительных контейнеров. Но часть предметов из  $L_3$  можно уложить в контейнеры для  $L_2$ . Каждый из них имеет  $1 - w_i$ ,  $i \in L_2$  свободного места, где поместится  $\left\lfloor \frac{1 - w_i}{\alpha} \right\rfloor$  предметов из  $L_3$ . ■

**Следствие 1.** Величина  $H = \max\{H_1(\alpha), H_2(\alpha), 0 \leq \alpha \leq 0,5\}$

является нижней оценкой для  $OPT(L)$ .

**Следствие 2.**  $H \geq H_0 \equiv \left\lceil \sum_{i \in L} w_i \right\rceil$ .

**Доказательство.** При  $\alpha = 0$  получаем  $H \geq H_1(0) = \max\{|L_2|, H_0\} \geq H_0$ .

Как найти  $H$ , не перебирая все значения  $\alpha$ ?

**Следствие 3.** Пусть  $V$  — множество всех различных значений  $w_i \leq 0,5$ .

Тогда

$$H = \begin{cases} n, & \text{если } V = \emptyset, \\ \max\{H_1(\alpha), H_2(\alpha), \text{ для } \alpha \in V\}, & \text{если } V \neq \emptyset. \end{cases}$$

т. е. после сортировки предметов получаем  $T_H = O(n + n \log n)$ .

## Дополнительная литература

1. E.G. Coffman, M.R. Garey, D.S. Johnson. Approximation algorithms for bin packing: A survey. <http://www.math.nsc.ru/LBRT/k5/bp-chapter.pdf>
2. Э.Х. Гимади. О некоторых математических моделях и методах планирования крупномасштабных проектов //Модели и методы оптимизации. Труды Института математики. Новосибирск. Наука. Сиб. Отд–ние. 1988. с. 89–115.
3. М. Гэри, Д. Джонсон. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982. с. 154–191.