Приближенные схемы

Задачи упаковки

Задача об упаковке в контейнеры

- *Дано*: n предметов и их размеры $a_1,\ldots,a_n \in (0,1]$.
- *Найти* упаковку всех предметов в контейнеры размера 1, так чтобы минимизировать число использованных контейнеров.

Правило «Первый подходящий»

- Рассмотрим предметы в произвольном порядке. Пусть мы получили лист частично заполненных контейнеров, скажем $B_1,\ldots, B_k$.
- Положим очередной предмет в первый контейнер, в который он войдет. Если такого нет, то откроем новый контейнер B_{k+1} и положим предмет в него.

Алгоритм «Первый подходящий»

Input $(a_1,\ldots,a_n)$

1) $i \leftarrow 1,\ k \leftarrow 1,\ \mathrm{bin}(1) \leftarrow 1,\ \mathrm{bin}(2) \leftarrow 1.$
2) **While** $i \le n$ **do:**

   $r \leftarrow \min\{j \le k+1 \mid a_i \le \mathrm{bin}(j)\}$

   if $(r \le k)$ **then** $\mathrm{bin}(r) \leftarrow \mathrm{bin}(r) - a_i$

   otherwise $k \leftarrow k + 1$

   $\mathrm{bin}(k) \leftarrow 1 - a_i$

   $\mathrm{bin}(k+1) \leftarrow 1$

Output (k)

Оценка качества алгоритма «Первый подходящий»

Теорема 6.1

Алгоритм «Первый подходящий» 2-приближенным алгоритмом для задачи об упаковке.

Доказательство: Пусть использовано k контейнеров.

$$OPT \geq \sum_{i=1}^{n} a_i > \frac{k-1}{2} \Rightarrow 2OPT > k-1 \Rightarrow 2OPT \geq k$$

Неаппроксимируемость

Теорема 6.2

Для любого $\varepsilon > 0$, не существует ρ-приближенного алгоритма для задачи об упаковке в контейнеры с $\rho = 3/2 - \varepsilon$, если **P ≠ NP**.

Идея доказательства.

Сведем NP-трудную задачу «Разбиение» к проверке можно ли разместить все предметы в два контейнера.

Асимптотическая приближенная схема

- **Теорема 6.3**

  Для любого ε, $0 < \varepsilon \leq 0.5$, существует полиномиальный от n алгоритм A_ε, который находит упаковку, использующую не более $(1+2\varepsilon)OPT + 1$ контейнер.

Упаковка ограниченного числа больших предметов

- **Лемма 6.4**

  Пусть $\varepsilon > 0$ фиксированная константа и $K > 0$ фиксированное целое. Рассмотрим примеры задачи об упаковке в контейнеры, в которых размеры предметов не меньше ε, и число различных размеров равно K. Тогда существует полиномиальный алгоритм, который точно решает такие примеры.

Доказательство

- Число предметов в одном контейнере $\le \lfloor 1/\varepsilon \rfloor := M$.
- Пусть x_i – число предметов i-го размера в контейнере.
- Вектор $(x_1, \ldots, x_K)$ определяет тип контейнера.
- Число различных типов контейнеров $\le \binom{M+K-1}{M} := R$.
- Общее число контейнеров $\le n$.
- Число возможных допустимых упаковок $\le \binom{n+R-1}{R} := P$ и ограничено полиномом от n.

Упаковка больших предметов

- **Лемма 6.5**

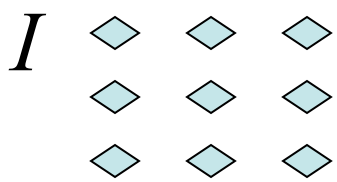
  Пусть $\varepsilon > 0$ фиксированная константа. Рассмотрим примеры задачи об упаковке в контейнеры, в которых размеры предметов не меньше ε. Тогда существует полиномиальный алгоритм, который находит упаковку в не более чем $(1+\varepsilon)OPT$ контейнеров, где OPT – число контейнеров в оптимальной упаковке.

Доказательство (преобразование примера *I*)

Упорядочим все предметы по не убыванию размеров.

Разобьем их в $K=\lceil 1/\varepsilon^2 \rceil$ групп по не более чем $Q=\lfloor n\varepsilon^2 \rfloor$ предметов.

Округлим размер каждого предмета к размеру наибольшего в группе.



Доказательство (преобразование примера *I*)

Упорядочим все предметы по не убыванию размеров.

Разобьем их в $K=\lceil 1/\varepsilon^2 \rceil$ групп по не более чем $Q=\lfloor n\varepsilon^2 \rfloor$ предметов.

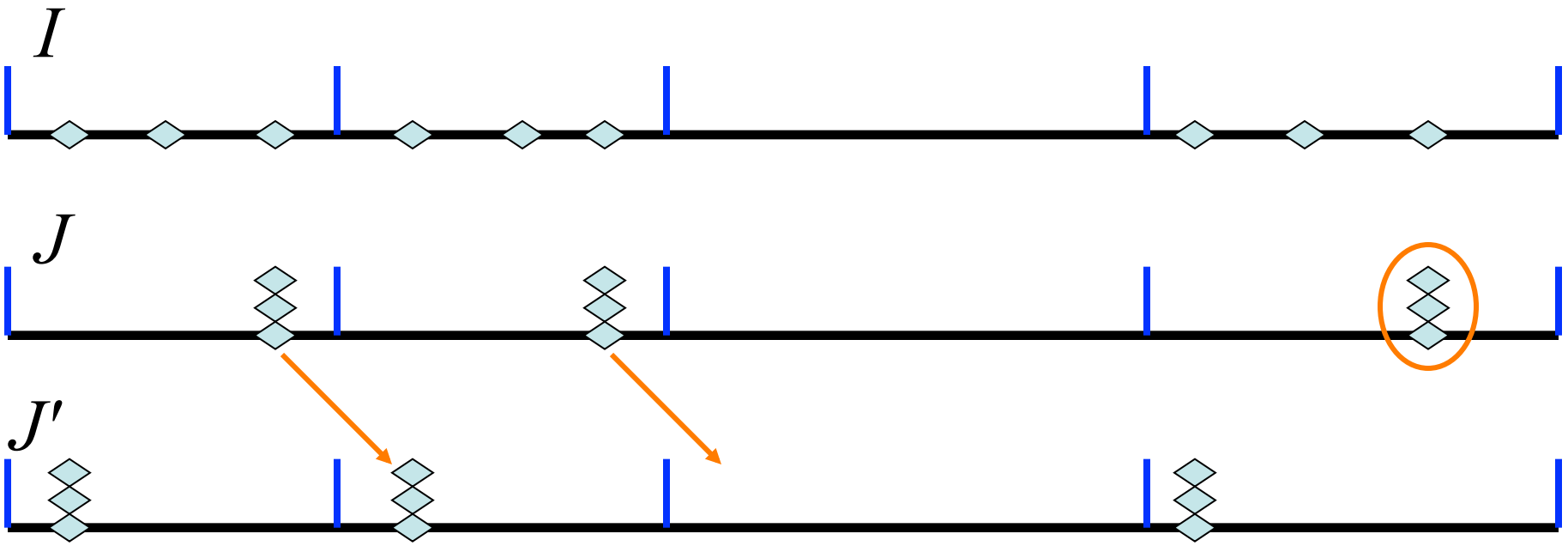
Округлим размер каждого предмета к размеру наибольшего в группе.

В *J* не более *K* различных размеров предметов.

По лемме 6.5 можно найти оптимальную упаковку для примера *J*.

J

Доказательство
OPT(J) ≤(1+ε) OPT(I)



$OPT(J') \le OPT(I)$

$OPT(J) \le OPT(J') + Q \le OPT(I) + Q$

$OPT(I) \ge n\varepsilon$ ⇒ $Q = n\varepsilon^2 \le \varepsilon OPT(I)$ ⇒ $OPT(J) \le (1+\varepsilon) OPT(I)$

Алгоритм Фернандес де ла Вега-Луекера

Input $(a_1,\ldots,a_n)$

1) Определить **Big** = $\{ j \mid a_j \geq \varepsilon\}$ и **Small** = $\{ j \mid a_j < \varepsilon\}$
2) Разбить большие предметы в $K=\lceil 1/\varepsilon^2 \rceil$ групп по не более чем $Q=\lfloor n\varepsilon^2 \rfloor$ предметов.
3) Округлить размер каждого большого предмета к размеру наибольшего в группе.
4) Упаковать большие предметы оптимальным образом относительно новых размеров.
5) Упаковать маленькие предметы используя правило «Первый подходящий». Пусть k – число использованных контейнеров в полученной упаковке.

Output (k)

Асимптотическая приближенная схема

- **Теорема 6.3**

  Для любого фиксированного ε, $0 < \varepsilon \leq 0.5$, алгоритм Фернандес де ла Вега-Луекера находит упаковку, использующую не более $(1+2\varepsilon)OPT + 1$ контейнера и время его работы ограничено полиномом от n.

Доказательство Теоремы 6.3

- k – число использованных контейнеров в полученной упаковке.
- Пусть I' пример, полученный из исходного примера I, удалением множества **Small.**
- На шаге 4 алгоритм получит упаковку больших предметов в не более чем $(1+\varepsilon)\mathrm{OPT}(I')$ контейнеров.
- Если на шаге 5 вставляя маленькие работы не потребуется ни одного нового контейнера, то $k = (1+\varepsilon)\mathrm{OPT}(I') \leq (1+\varepsilon)\mathrm{OPT}(I)$.
- В противном случае, по крайней мере k–1 контейнер должен быть заполнен не меньше чем на 1–ε.
- Следовательно, сумма размеров предметов в I, по крайней мере $(k–1)(1–\varepsilon) \leq \mathrm{OPT}(I)$.
- $k \leq \mathrm{OPT}(I)/(1–\varepsilon) + 1 \leq (1+2\varepsilon)\mathrm{OPT}(I)+1$, $(0 < \varepsilon < 1/2)$.

Задача $P||C_{max}$

- *Дано*: m машин, n работ и их длительности $p_1,\ldots,p_n$.
- *Найти* назначение всех работ на m машин, так чтобы минимизировать длину расписания (время завершения всех работ, максимальную загрузку машины).

Жадный алгоритм с произвольным списком (GL)

- Задан список работ в произвольном порядке.
- Когда какая-нибудь машина освобождается, первая работа из списка назначается на эту машину и удаляется из списка.

Первый приближенный алгоритм

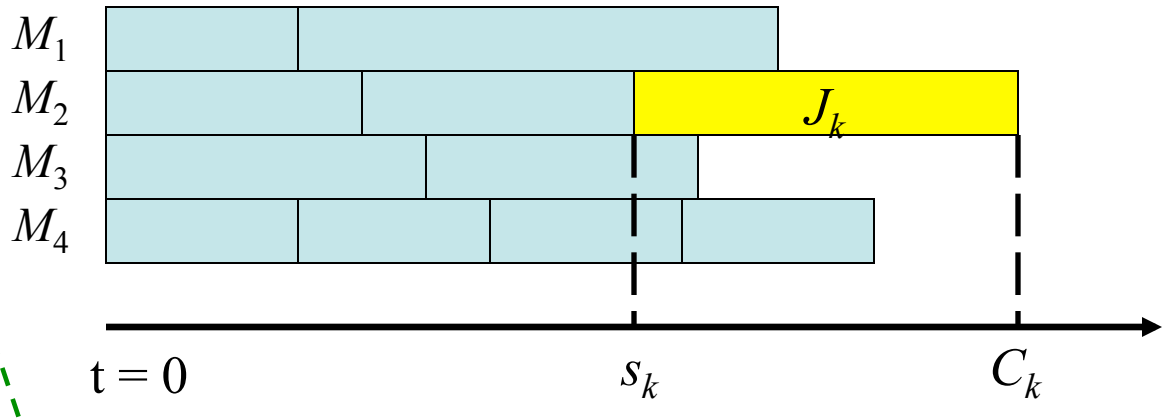
Теорема 6.6 (Грэхэм [1966])

Алгоритм GL *является* $(2 - 1/m)$*-приближенным алгоритмом для* $P||C_{max}$.

Доказательство

$$C^*_{\max} \geq (1/m)\sum_{j=1}^{n} p_j,$$

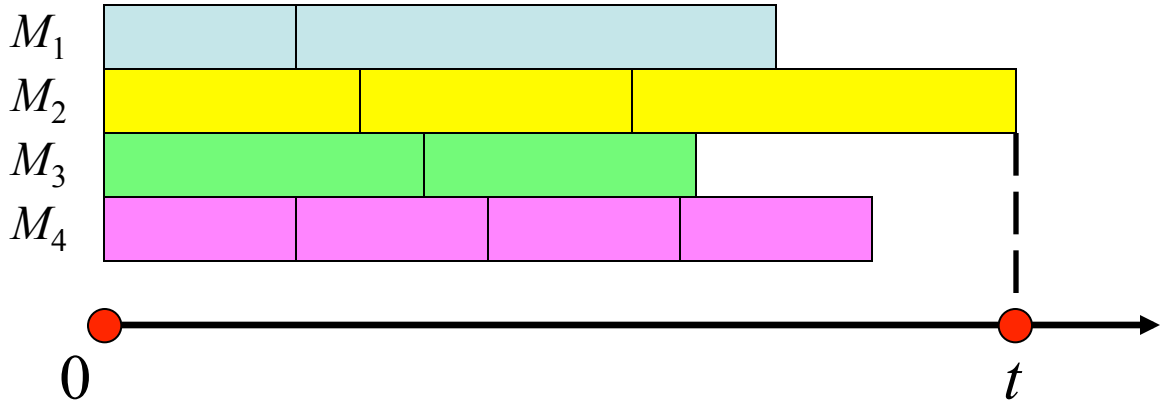
$$C^*_{\max} \geq p_k,$$



$$C^{LS}_{\max} = C_k = s_k + p_k \leq (1/m)\sum_{j \neq k} p_j + p_k =$$

$$(1/m)\sum_{j=1}^{n} p_j + (1-1/m)p_k \leq C^*_{\max} + (1-1/m)C^*_{\max} = (2-1/m)C^*_{\max}.$$

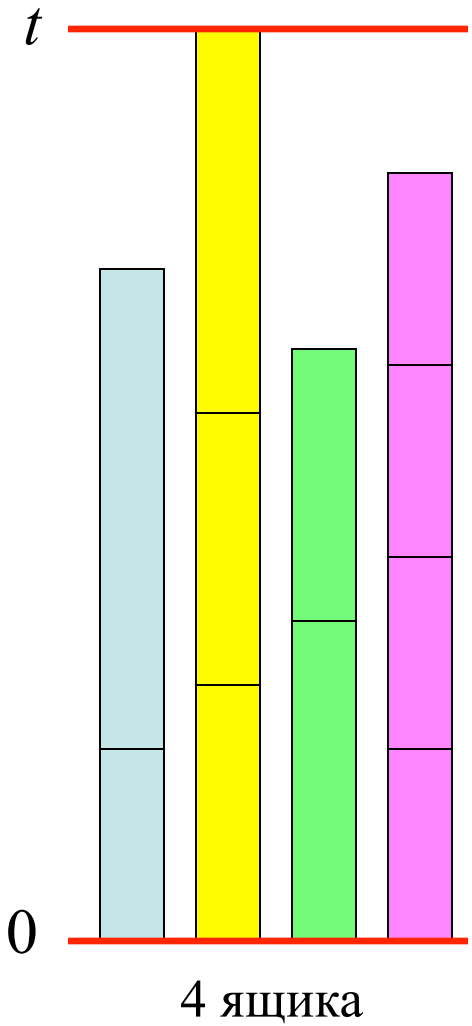
P||$C_{\max}$ и задача об упаковке



Пусть I набор предметов с размерами $p_1,\ldots,p_n$.

bins(I,t) – минимальное число контейнеров размера t, требуемое упаковать все предметы из множества I.

$$C^*_{\max} = \min\{t : \text{bins}(I,t) \le m\}$$



Бинарный поиск

- Чтобы осуществить сведение нам нужно знать или угадать значение $C_{\max}$ в оптимальном решении.

$$LB = \max\left\{(1/m)\sum_{j=1}^{n} p_j, \max_j\{p_j\}\right\}$$ – нижняя оценка.

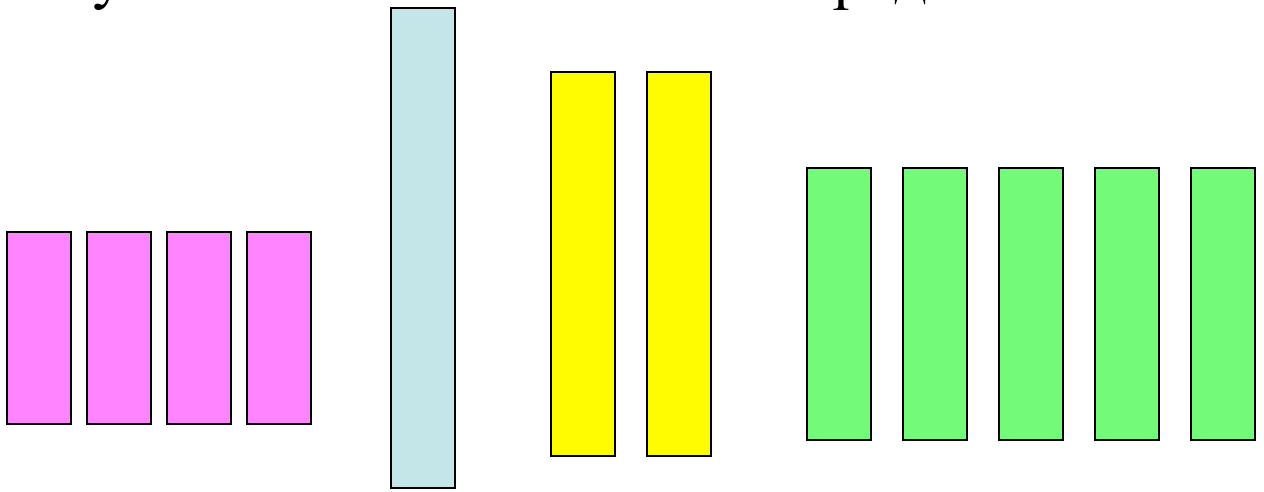
$$LB \le C^*_{\max} \le 2LB$$

- Можно угадать значение оптимального решения устроив бинарный поиск и используя в качества проверки решение задачи об упаковке в контейнеры.

Упаковка предметов с ограниченным числом размеров

- Пусть k – фиксированное число размеров предметов.
- Зафиксируем порядок размеров.
- Для каждого примера вход может быть представлен как вектор из k компонент, $(i_1, i_2,\ldots, i_k)$ указывающий число предметов каждого размера.
- BINS$(i_1, i_2,\ldots, i_k)$ – минимальное число контейнеров требуемое упаковать это множество предметов.

(4,1,2,5)

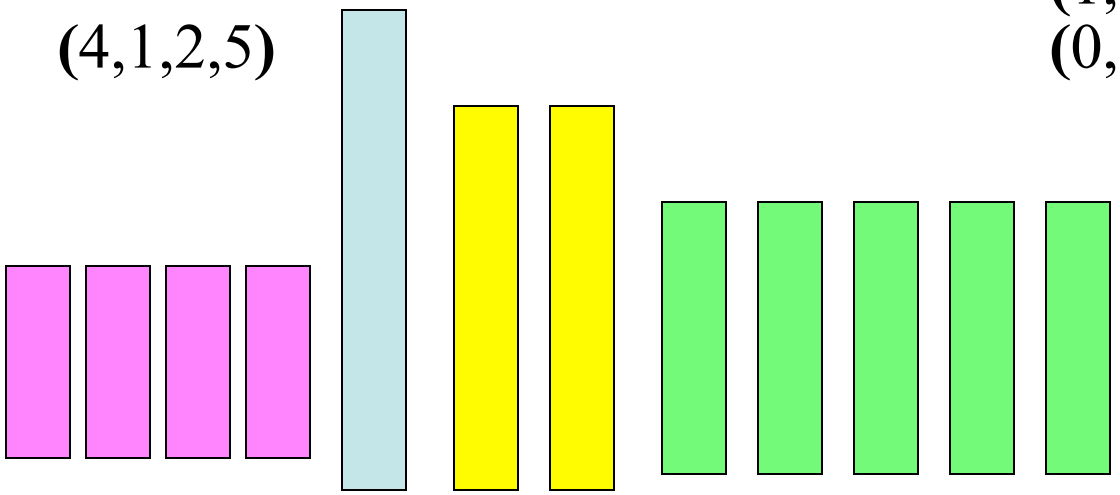


Один контейнер

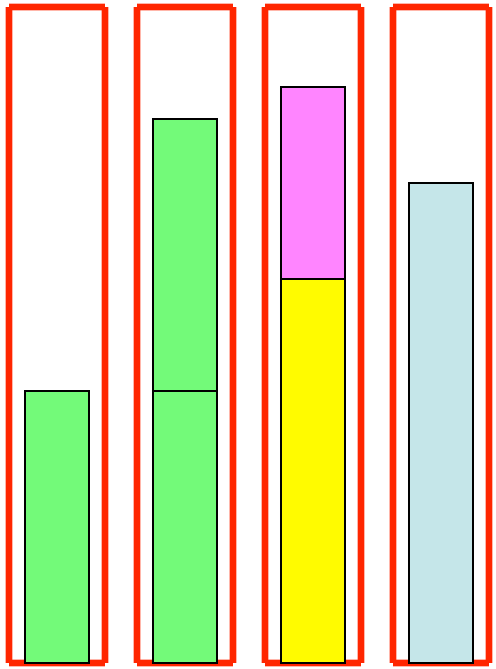
Для данного примера $(n_1, n_2, \ldots, n_k)$, $\Sigma n_i = n$, вычислим множество Q всех векторов $(q_1, q_2, \ldots, q_k)$, таких что $\mathrm{BINS}(q_1, q_2, \ldots, q_k) = 1$ и $0 \le q_i \le n_i$, $1 \le i \le k$.

$$|Q| \le \prod_{i=1}^{k} n_i \le n^k$$

(4,1,2,5)



(0,0,0,1)
(0,0,0,2)
(1,0,1,0)
(0,1,0,0)



Динамическое программирование

- Вычислим все значения k-размерной таблицы BINS(i_1, i_2,…, i_k) для каждого (i_1, i_2,…, i_k)$\in\{0,\ldots,n_1\}\times\{0,\ldots,n_2\}\times\ldots\times\{0,\ldots,n_k\}$.
- Сначала заполним таблицу для элементов из Q.
- Вычислим остальные значения по формуле
$$\mathrm{BINS}(i_1,i_2,\ldots,i_k)=1+\min_{q\in Q}\{\mathrm{BINS}(i_1-q_1,i_2-q_2,\ldots,i_k-q_k)\}.$$
- Все значения в таблице могут быть найдены за $O(n^{2k})$.

Идея схемы

- Поскольку в PTAS мы можем позволить некоторую неточность в вычислении длины расписания, то мы можем свести задачу $P||C_{max}$ к задача об упаковке в контейнеры ограниченного числа предметов.
- При сведении появятся два источника ошибок.
  - при округлении длительностей работ, для того чтобы ограничить число различных размеров предметов.
  - при остановке бинарного поиска.
- Покажем, что можно получить относительную погрешность сколь угодно близкой к 1. При этом трудоемкость алгоритма экспоненциально зависит от $1/\varepsilon$, и полиномиально от n.

Основная процедура для фиксированного t ($LB \le t \le 2LB$)

Input ($p_1,\ldots,p_n,\varepsilon,t$)

1) Определить **Big** = $\{ j \mid p_j \ge t\varepsilon\}$ и **Small** = $\{ j \mid p_j < t\varepsilon\}$.
2) Округлить размер каждого большого предмета:
   - **if** $p_j \in [t\varepsilon(1+\varepsilon)^i, t\varepsilon(1+\varepsilon)^{i+1})$ **then** $p_j' \leftarrow t\varepsilon(1+\varepsilon)^i$**.**
3) Используя динамическое программирование, найти оптимальную упаковку $\boldsymbol{U}$ больших предметов (p_j') в контейнеры размера t.
4) Рассмотрим $\boldsymbol{U}$, как упаковку предметов с исходными предметами. Тогда $\boldsymbol{U}$ допустимая упаковка в контейнеры размера $t(1+\varepsilon)$.
5) Упаковать маленькие предметы, используя правило «Первый подходящий». Пусть $\alpha(I,\varepsilon,t)$ – число использованных контейнеров в полученной упаковке.

Output ($\alpha(I,\varepsilon,t)$)

Трудоемкость основной процедуры

- Число различных значений p_j' -х $k = \lceil \log_{1+\varepsilon} 1/\varepsilon \rceil$.
- Трудоемкость основной процедуры определяется трудоемкостью динамического программирования и равна $O(n^{2k})$.
- При фиксированном ε трудоемкость процедуры полиномиально зависит от n.

Нижняя оценка на число ящиков

- **Лемма 6.7**

$$\alpha(I,\varepsilon,t) \leq \mathrm{bins}(I,t).$$

Доказательство

- Пусть на шаге 5 процедуры (до упаковки маленьких предметов) не был открыт ни один новый контейнер. Утверждение леммы верно, так как даже для округленного примера в найденном основной процедурой оптимальном решении требуется $\alpha(I,\varepsilon,t)$ контейнеров.
- В противном случае, все контейнеры кроме одного должны быть заполнены не меньше чем на t. Следовательно сумма размеров предметов $> t\,(\alpha(I,\varepsilon,t) - 1)$ и нужно как минимум $\alpha(I,\varepsilon,t)$ контейнеров, чтобы упаковать все предметы.

Нижняя оценка на длину расписания

- **Следствие 6.8**

$$\min\{t: \alpha(I,\varepsilon,t) \le m\} \le OPT.$$

Доказательство.

$OPT = \min\{t: \text{bins}(I,t) \le m\}$.

По лемме 6.7 для любого t: $\alpha(I,\varepsilon,t) \le \text{bins}(I,t)$.

Отсюда, $\min\{t: \alpha(I,\varepsilon,t) \le m\} \le OPT$.

Как найти t ?

- Бинарный поиск в интервале $[LB, 2LB]$.
- На каждой итерации длина интервала уменьшается вдвое.
- Остановим процедуру бинарного поиска, когда длина интервала станет εLB и обозначим его $[T - \varepsilon LB, T]$.
- Это потребует $\lceil \log_2 1/\varepsilon \rceil$ итераций.

Нижняя оценка

- **Лемма 6.9**

$$T \le (1+\varepsilon)OPT.$$

Доказательство.

$$\min\{t: \alpha(I,\varepsilon,t) \le m\} \in [T - \varepsilon LB, T]$$

$$T \le \min\{t: \alpha(I,\varepsilon,t) \le m\} + \varepsilon LB \le (1+\varepsilon)OPT.$$

Приближенная схема

- **Теорема 6.10**

  Для любого $\varepsilon > 0$, существует алгоритм A_ε, который находит расписание длины не больше $(1+\varepsilon)^2 OPT \leq (1+3\varepsilon)OPT$ за $O(n^{2k} \lceil \log_2 1/\varepsilon \rceil)$ операций, где $k = \lceil \log_{1+\varepsilon} 1/\varepsilon \rceil$.

Упражнения

- Рассмотрим вариант алгоритма «Первый подходящий», называемый алгоритм «Следующий подходящий». Если очередной предмет не помещается в последний открытый контейнер, то открываем новый пустой контейнер. Показать, что этот алгоритм является 2-приближенным, и оценка 2 является точной.