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

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

Множества B_j называют контейнерами.

Требуется упаковать предметы в минимальное число контейнеров.

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.

$\mathrm{T} = \mathrm{O}(n)$, $\Pi = \mathrm{O}(1)$, если не считать место для исходных данных.

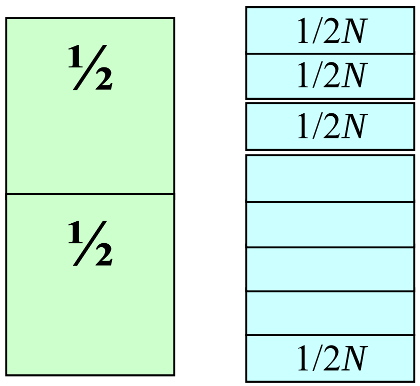
Теорема. $NF(L) \le 2OPT(L)$.

Доказательство. Пусть $W = \sum_{i \in L} w_i$. Так как любые два последовательных контейнера содержат предметы суммарным весом не меньше единицы, то $NF(L) < 2\lceil W \rceil$. Кроме того, $OPT(L) \ge \lceil W \rceil$, откуда и следует требуемое.

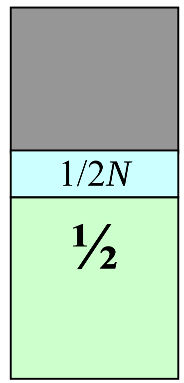
■

Пример

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



N $\quad$ 1



$2N$

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

$$NF(L) = 2N$$

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

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

$$R_A(L) \equiv {A(L)}\Big/{OPT(L)}.$$

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

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

Определение Асимптотическая гарантированная относительная точность R_A^∞ для алгоритма A определяется как

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

Алгоритм «Первый подходящий» (FF)

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

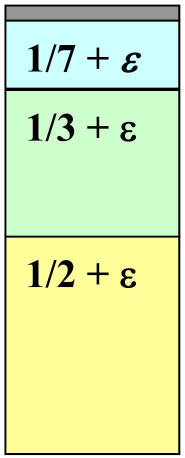
На k-м шаге находим контейнер с наименьшим номером, куда помещается k-й предмет, и помещаем его туда. Если такого контейнера нет, то берем новый пустой контейнер и помещаем предмет в него.

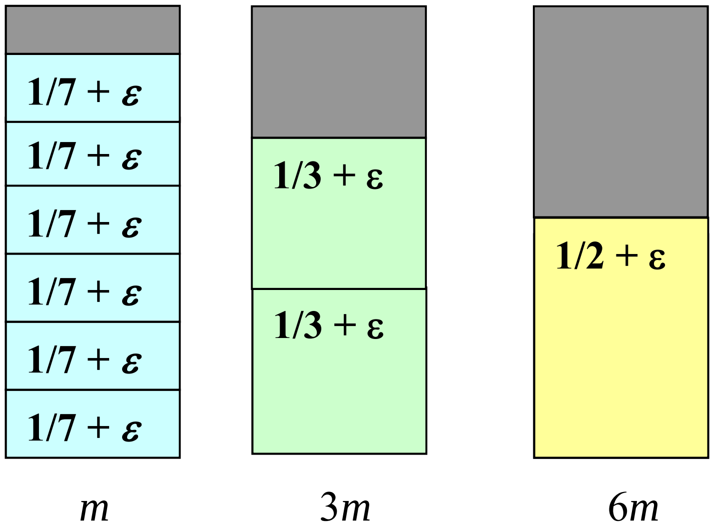
$\mathrm{T} = \mathrm{O}(n^2)$, $\Pi = \mathrm{O}(n)$.

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

Пример

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





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

$$OPT(L) = 6m$$

$$FF(L) = 10m$$

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 предмет в него.

$\mathrm{T} = \mathrm{O}(n^2)$, $\Pi = \mathrm{O}(n)$.

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

(*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*)

Алгоритм «Первый подходящий с упорядочиванием» (FFD)

- Сортируем предметы по невозрастанию весов
  $w_1 \geq w_2 \geq \cdots \geq w_n$
- 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

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

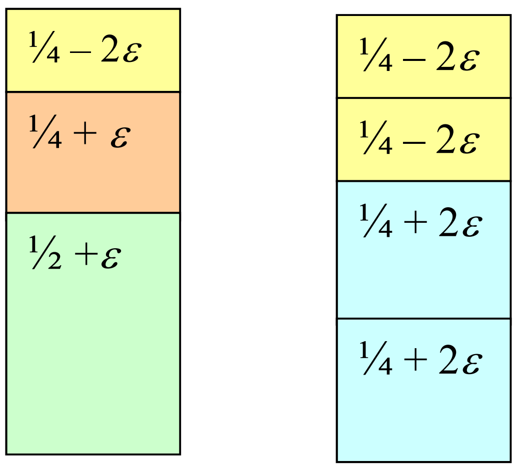
Кроме того $R_{FFD}^{\infty} = R_{BFD}^{\infty} = \frac{11}{9} \approx 1.22.$

(*Без доказательства*)

Пример

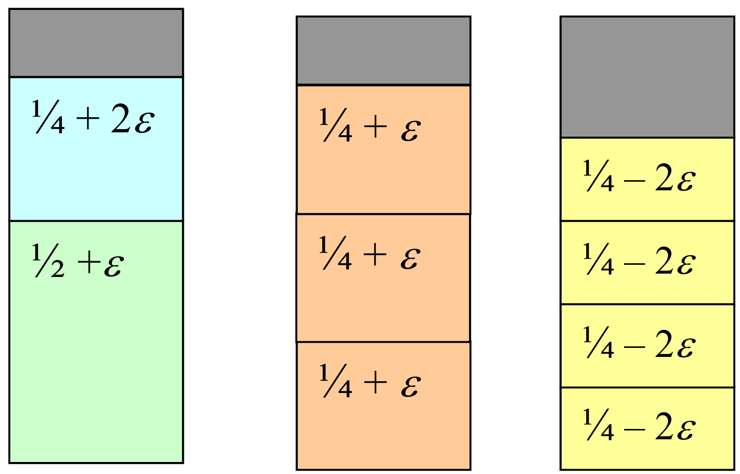
$$L = \{1, \dots, 30\,m\}$$

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



$6m$ $3m$

$$OPT(L) = 9m$$



$6m$ $2m$ $3m$

$$FFD(L) = 11m$$

Классы P и NP

Задачи распознавания — задачи с ответом *«да»* или *«нет»*.

Пример: Дан граф, является ли он связным?

Класс P — класс задач распознавания, которые можно решить с полиномиальной трудоемкостью.

Пример: Дан граф, существует ли в нем эйлеров цикл?

Класс NP — класс задач распознавания, в которых можно проверить решение с ответом «да» за полиномиальное время, то есть класс задач, решаемых за полиномиальное время на недетерминированной машине Тьюринга.

Пример: Дан граф, существует ли в нем гамильтонов цикл?

NP-полные задачи — самые трудные задачи в NP, то есть если существует точный полиномиальный алгоритм для решения одной из них, то существует точный полиномиальный алгоритм для решения всех задач из класса NP.

Вопросы

- Даны веса n предметов и число K. Можно ли упаковать эти предметы в K контейнеров? Эта задача из класса NP *(Да или Нет?)*

- Даны веса n предметов. Можно ли так разбить их на два подмножества, чтобы сумма весов предметов в каждом подмножестве была бы ровно в половину от общего веса всех предметов? Эта задача из класса NP *(Да или Нет?)*

- Если вторая задача NP-полна, то и первая задача NP-полна? *(Да или Нет?)*

- Если первая задача NP-полна, то и вторая задача NP-полна? *(Да или Нет?)*

- Правда ли, что P $\subseteq$ NP ? *(Да или Нет?)*

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 $a_1, \ldots, a_n$. 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 $w_i = \frac{a_i}{C}, i = 1, \ldots, n$. 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

$$\sum_{i=1}^{n} w_i x_{ij} \le y_j, \qquad j = 1, \dots, n,$$

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

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

Релаксация линейного программирования

Заменим условие булевости переменных на условия:

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

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

Тогда одно из оптимальных решений имеет вид

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

что дает нижнюю оценку

$$H_0 = \left\lceil \sum_{i=1}^{n} w_i \right\rceil$$

(предметы можно резать произвольным образом).

Оценки Martello & Toth

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

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

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

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

Теорема Для любого $0 \le \alpha \le \frac{1}{2}$ 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 контейнерах. 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 $0 \le \alpha \le \frac{1}{2}$ 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 α. Тогда в один контейнер войдет $\left\lfloor \frac{1}{\alpha} \right\rfloor$ предметов, и для множества L_3 потребовалось бы $\left\lceil {|L_3|} \middle/ {\left\lfloor \frac{1}{\alpha} \right\rfloor} \right\rceil$ 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 $H = \max\{H_1(\alpha), H_2(\alpha),\ 0 \le \alpha \le 0{,}5\}$

является нижней оценкой для $OPT(L)$.

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

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

Как найти H, не перебирая все значения α ?

Следствие 3 Пусть V — множество всех различных значений $w_i \le 0{,}5$. Тогда

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

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

Вопросы

- При вычислении нижних оценок Martello &Toth предметы весом менее α выбрасываются *(Да или Нет?)*

- Алгоритмы вычисления нижних оценок Martello &Toth являются полиномиальными (Да или Нет?)

- Выбор оптимального значения параметра α осуществляется полиномиальным алгоритмом *(Да или Нет?)*

- Если не выбрасывать предметы весом менее α, то можно улучшить нижнюю оценку *(Да или Нет?)*

- Почему на три группы, а не на пять, семь, …? Можно ли так улучшить нижнюю оценку *(Да или Нет?)*

Нижние оценки Гилмора и Гомори

Пусть $L = \{1, \ldots, m\}$ — множество типов предметов. Для каждого типа $i \in L$ задан вес предмета $0 < w_i < 1$ и их количество $n_i \geq 1$. Требуется упаковать все предметы в минимальное число контейнеров единичной вместимости.

Рассмотрим множество J всех вариантов упаковки одного контейнера. Пусть матрица (a_{ij}) задает число предметов -го типа в j-м варианте упаковки.

Переменные задачи:

$x_j \geq 0$, целые — число контейнеров, упакованных по -му варианту

$$\min \sum_{j \in J} x_j$$

$$\text{при условиях:} \quad \sum_{j \in J} a_{ij} x_j \geq n_i, \quad i \in L;$$

$$x_j \geq 0, \text{ целые}, \ j \in J.$$

Множество J может иметь экспоненциальную мощность.

Нижняя оценка

$$H = \min \sum_{j \in J} x_j$$

$$\sum_{j \in J} a_{ij} x_j \geq n_i, \quad i \in L;$$

$$x_j \geq 0, \qquad j \in J.$$

Решая задачу линейного программирования, получаем нижнюю оценку H.

Но задача имеет гигантскую размерность!

Метод генерации столбцов

Выберем подмножество $J' \subset J$ так, чтобы следующая подзадача ЛП(J') имела решение:

$$\min \sum_{j \in J'} x_j$$

$$\sum_{j \in J'} a_{ij} x_j \geq n_i, \quad i \in L;$$

$$x_j \geq 0, \qquad j \in J'.$$

Это легко сделать с помощью любой жадной эвристики: *NF*, *FF*, *BF*,…

Пусть x_j^* — оптимальное решение для J' и $\lambda_j^* \geq 0$ — оптимальное решение соответствующей J' двойственной задачи. Рассмотрим двойственную задачу для множества J':

$$\max \sum_{i \in L} n_i \lambda_i$$

при ограничениях
$$\sum_{i \in L} a_{ij} \lambda_i \leq 1, \quad j \in J';$$
$$\lambda_i \geq 0, \qquad i \in L.$$

Если для всех $j \in J$ справедливо

$$\sum_{i \in L} a_{ij} \lambda_i^* \leq 1, \qquad (*)$$

то вектор $\bar{x}_j = \begin{cases} x_j^*, & j \in J' \\ 0, & j \in J \setminus J' \end{cases}$ — оптимальное решение задачи линейного программирования для всего множества J и

$$H = \sum_{j \in J'} x_j^*.$$

Проблема: как проверить $(*)$, не просматривая все множество J, и что делать, если условие не выполняется.

Рассмотрим задачу о рюкзаке:

$$\alpha = \max \sum_{i \in L} \lambda_i^* y_i$$

при ограничениях:
$$\sum_{i \in L} w_i y_i \le 1$$
(вместимость контейнеров)

$$y_i \ge 0, \text{ целые, } i \in L.$$

Оптимальное решение этой задачи дает нам новый вариант упаковки контейнера.

Если $\alpha \le 1$, то $(*)$ выполнено!

Если $\alpha > 1$, то получили вариант упаковки, который следует добавить в множество J'(нашли ведущий столбец в симплекс–таблице).

Общая схема метода:

1. Выбрать подмножество $J' \subset J$
2. Решить задачу $\text{ЛП}(J')$, получить x_j^*, λ_j^*.
3. Решить задачу о рюкзаке, получить α.
4. Если $\alpha \leq 1$, то $H = \sum_{j \in J'} x_j^*$, STOP.
5. Добавить в J' новый вариант упаковки и вернуться на шаг 2

Оценка $H = \sum_{j \in J'} x_j^*$ является трудоемкой, но доминирует другие по точности. Решение $x_j = \lceil x_j^* \rceil, j \in J,$ дает верхнюю оценку и часто оказывается оптимальным.

Вопросы

- Метод генерации столбцов дает точное решение задачи упаковки в контейнеры *(Да или Нет?)*
- Если на шаге 2 придется решать полиномиальное число задач линейного программирования, то метод будет полиномиальным?
- Правда ли, что размерность задачи о рюкзаке на шаге 3 растет с ростом числа итераций метода?
- Правда ли, что метод требует решения полиномиального числа задач о рюкзаке?
- Если на шаге 4 после срабатывания STOP решить точно целочисленную задачу для финального подмножества J', то получим точное решение исходной задачи?
- В чем смысл решать одну NP-трудную задачу (упаковки) путем многократного решения другой NP-трудной задачи (о рюкзаке)?