

МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ

факультет инноваций и высоких технологий

Кафедра дискретной математики

Задача упаковки по корзинам

Индивидуальный проект

Поляков Григорий 799

Contents

1	Введение	3
2	Задача о $o(\frac{3}{2} - \varepsilon)$ - приближении	4
3	Алгоритм, дающий $(1 + \varepsilon)OPT + 1$ корзин	5
	Список литературы	8

1 Введение

Постановка задачи

Рассмотрим некоторое множество элементов под номерами $I = \{1, \dots, n\}$. Каждый элемент $i \in I$ имеет "размер" $a_i \in (0, 1]$. Пусть также дано множество так называемых корзин $B = 1, \dots, n$.

Требуется распределить все элементы по наименьшему числу корзин так, чтобы сумма размеров всех элементов в каждой корзине не превосходила 1.

Более формально, построить такую функцию $\xi : I \rightarrow B$, что:

$$\forall b \in B \quad \sum_{i \in I : \xi(i)=b} s_i \leq 1$$

2 Задача о $o(\frac{3}{2} - \varepsilon)$ - приближении

Утверждение: Задача **NP**-трудна

Доказательство: Сведение задачи PARTITION.

Для начала сформулируем задачу PARTITION. Дан конечный набор целых чисел A . Определить существует ли подмножество $B \subset A$, что $\sum_{a \in B} a = \sum_{a \in A \setminus B} a$

Из курса знаем, что PARTITION является **NP** -полной.

Построим полиномиально-вычислимую функцию сведения. Пусть задан набор чисел a_1, \dots, a_n , для которых нужно решить задачу PARTITION. Пусть $S_a = \sum_{i=1}^n a_i$. Тогда $s_i = f(a_i) = \frac{2a_i}{S_a}$. Покажем, что существует подмножество для PARTITION \Leftrightarrow минимальное количество корзин для разбиения i_1, \dots, i_n с весами s_1, \dots, s_n равно 2.

Обозначим OPT - решение задачи разбиения на корзины.

- \Rightarrow Заметим, что $OPT > 1$, т.к. $\sum_{i=1}^n s_i = 2$. Если существуют подмножества B и $B \setminus A$ - решения задачи PARTITION, то существует разбиение набора s_1, \dots, s_n на две корзины, сумма в каждой равна единице.
- \Leftarrow Пусть $OPT = 2$. Тогда сумма весов в каждой корзине равна 1. Следовательно, подмножества исходного набора a_1, \dots, a_n , соответствующие разбиению на корзины, и будут решением задачи PARTITION.

В данном сведении приближение не играет роли, т.к. если решение задачи разбиения на корзины равно 2 то алгоритм, решающий задачу о $o(\frac{3}{2} - \varepsilon)$ - приближении всегда его найдёт. Ведь $2 \cdot \frac{3}{2} = 3$, следовательно ответ 3 - уже будет хуже, чем $(\frac{3}{2} - \varepsilon)$ - приближение.

3 Алгоритм, дающий $(1 + \varepsilon)OPT + 1$ корзину

Теорема: Пусть OPT оптимальное число корзин. Тогда, для любого $\varepsilon \in (0, 1]$ существует полиномиальный по n алгоритм, дающий разбиение на $(1 + \varepsilon) \cdot OPT + 1$ корзин в худшем случае.

Лемма 1: Пусть $\varepsilon > 0$ и $d > 0$ - некоторые константы. Пусть также $\forall i \in I : s_i \geq \varepsilon$ и количество различных среди всех s_i не превосходит d . Тогда, существует полиномиальный по n алгоритм, дающий в точности оптимальное значение для задачи упаковки по корзинам

Доказательство: Проведём полный перебор всех возможных упаковок. Максимальное число элементов в одной корзине равно $\lfloor \frac{1}{\varepsilon} \rfloor$. Обозначим это число за m .

Тогда, максимальное количество способов заполнить одну корзину C_{m+d-1}^m (в соответствии с формулой количества сочетаний с повторениями). Тогда, количество способов заполнить все n корзин не превосходит $n \cdot C_{n+m-1}^m$. Последнее выражение полиномиально зависит от n .

Таким образом, алгоритм просто будет перебирать все возможные варианты расстановки.

□

Лемма 2: Пусть $\forall i \in I : s_i \geq \varepsilon$. Тогда, существует полиномиальный по n алгоритм, дающий разбиение на $(1 + \varepsilon)OPT$ корзин, где OPT - оптимальное число корзин

Доказательство: (Во всех случаях мы сравниваем элементы по их размеру)

Отсортируем элементы для разбиения по увеличению размера s_i . Полученный набор назовём Q . Затем разделим их на $P+1$ группу. Тогда в каждой группе будет не более $\lfloor n/P \rfloor$ элементов. Проведём разбиение так, чтобы во всех группах, кроме последней элементов было поровну. Из полученного отсортированного набора элементов построим два новых набора H и J следующим образом:

- L : набор Q , в котором размер $s_{(i)}$ каждого элемента заменён на размер наименьшего по размеру элемента в соответствующей ему группе.
- G : набор Q , в котором размер $s_{(i)}$ каждого элемента заменён на размер наибольшего по размеру элемента в соответствующей ему группе.

Тогда обозначим $OPT(L)$, $OPT(Q)$, $OPT(G)$ - оптимальные решения задачи упаковки по корзинам для наборов Q , H и P соответственно.

Очевидно, что:

$$OPT(L) \leq OPT(Q) \leq OPT(G)$$

Покажем также, что $OPT(G) \leq OPT(L) + \lfloor n/P \rfloor$. Пусть мы решили задачу упаковки для L , получив некоторую функцию соответствия ξ . Построим новую функцию ψ . Заметим, что все элементы в группе $i + 1$ набора L не меньше элементов из группы i набора G (по построению). Тогда $\psi((i, l)) = \xi((i+1, l))$, где i - номер группы, l - номер элемента в группе. Элементы последней группы положим в ещё не заполненные корзины (каждый элемент в отдельную корзину). Всего элементов в последней группе $\leq \lfloor n/P \rfloor$, откуда и следует, что $OPT(G) \leq OPT(L) + \lfloor n/P \rfloor$.

Тогда:

$$OPT(G) \leq OPT(L) + \lfloor n/P \rfloor \leq OPT(Q) + \lfloor n/P \rfloor$$

Подберём такое P , что $\lfloor n/P \rfloor \leq \varepsilon \cdot OPT(Q)$. Тогда, $OPT(G) \leq (1 + \varepsilon) \cdot OPT(Q)$.

Из *леммы 1* следует, что существует алгоритм для точного решения задачи упаковки для набора G . Заметим, что по построению набора G (все элементы в нём не меньше элементов из исходного набора Q), полученное данным алгоритмом соответствие подойдёт и для набора Q . \square

Доказательство Теоремы: Разобьём исходный набор элементов I на два непересекающихся поднабора:

- набор I' , в котором для любого $i \in I'$ размер $s_i \geq \varepsilon$
- набор I'' , в котором для любого $i \in I''$ размер $s_i < \varepsilon$

Рассмотрим $\delta = \varepsilon/2 < 1/2$

Для набора I' и δ применима *Лемма 2*. Запустим алгоритм, полученный в *Лемме 2* на наборе элементов I' . Получим некоторую упаковку (соответствие).

Затем, для оставшихся элементов из I'' и полученной упаковки применим жадный алгоритм. Отсортируем их по убыванию размера s_i , затем для каждого элемента в порядке сортировки будем класть его в первую корзину, для которой сумма размеров уже лежащих элементов вместе с его размером не превзойдёт 1.

Обозначим полученное количество корзин за B .

Если после применения жадного алгоритма новых корзин не добавилось, то из *Леммы 2* получаем, что

$$B \leq (1 + \delta)OPT \leq (1 + \varepsilon)OPT + 1$$

Иначе, получим некоторую упаковку по корзинам, где для каждой (кроме может быть одной) заполненной нами корзины её остаточная вместимость $(1 - \sum_{i: \xi(i)=b_j} s_i)$ не превосходит δ .

Тогда,

$$OPT \geq \sum_{i \in I} s_i \geq (B - 1)(1 - \delta)$$

Используя неравенство $\frac{1}{1-x} \leq 1 + 2x$ ($\forall x \in [0, 1/2]$), получаем:

$$B \leq \frac{OPT}{1 - \delta} + 1 \leq (1 + 2\delta) \cdot OPT + 1 = (1 + \varepsilon) \cdot OPT + 1$$

□

Список литературы

- [1] Fernandez de la Vega, W., Lueker, G. (2007). Bin packing can be solved within $1 + \varepsilon$ in linear time. *Combinatorica*, 1(4), 349-355.
- [2] <http://u.cs.biu.ac.il/~amir/bin-packing.pdf>
- [3] http://ac.informatik.uni-freiburg.de/lak_teaching/ws11_12/combopt/notes/bin_packing.pdf