Лекция 12

Приближённые схемы

План лекции

- Асимптотическая приближённая схема
  - Определение. Класс APTAS
  - АППС для задачи упаковки ящиков
    - Динамическое программирование (общий принцип)
- Полностью полиномиальная приближённая схема
  - Определение. Класс FPTAS
  - ПППС для задачи «Рюкзак»
- Задание 6

Приближённые схемы

Полиномиальная приближённая схема (polynomial-time approximation scheme, PTAS) — приближённый алгоритм, который для заданного экземпляра задачи и числа $\varepsilon>0$ за полиномиальное (от размера задачи) время находит $(1+\varepsilon)$-приближённое решение.

PTAS — класс задач, для которых существуют ППС.

Очевидно, что PTAS $\subseteq$ APX.

В силу доказанной на предыдущей лекции теоремы,

BP $\notin$ PTAS, поэтому PTAS $\subset$ APX.

Асимтотические ППС

Для задачи «Упаковка ящиков» не существует ППС.
Но существует асимптотическая ППС (APTAS)!
Абсолютный коэффициент приближения

$$R(A) = \sup \left\{ \frac{c(A(x))}{c(opt(x))} : x \in X \right\}$$

Асимптотический коэффициент приближения

$$R^{\infty}(A) = \lim_{|x| \to \infty} \sup \left\{ \frac{c(A(x))}{c(opt(x))} : x \in X \right\}$$

Асимптотическая ППС для BP

Алгоритм Fernandez de la Vega — Lueker (FVL)

Основные идеи

- Свести исходную задачу к упрощённой
  - Исключить маленькие предметы
  - Ограничить количество возможных весов, сгруппировав предметы и округлив их веса
- Найти *точное* решение упрощённой задачи
- Получить решение исходной задачи, доразместив маленькие предметы

Асимптотическая ППС для ВР

Нормализация данных. Для упрощения выкладок примем предположение: b=1, т. е. вместимость ящика всегда равна 1. Экземпляр задачи с произвольной вместимостью можно свести к нормализованному варианту, разделив все веса на b.

Асимптотическая ППС для BP

Выберем $\varepsilon>0$.

Всё множество предметов разделим на:

- Small = $\{i: w_i < \varepsilon\}$ - пока откладываем в сторону
- Big = $\{i: w_i \geq \varepsilon\}$ - распределяем по ящикам

Рассмотрим ограниченный вариант задачи:

- веса всех предметов не меньше ε;
- количество различных весов предметов равно (или не превышает) заданному натуральному числу k.

Асимптотическая ППС для ВР

Лемма 1. Существует алгоритм, за полиномиальное время находящий точное решение ограниченного варианта задачи.

Доказательство

Рассмотрим некоторое размещение предметов по ящикам. Выберем произвольный ящик. Количество предметов в нём не превосходит $m = \lfloor 1/\varepsilon \rfloor$.

Пусть n_i — количество предметов i-го размера, помещённых в данный ящик. Тогда *тип загрузки* ящика задаётся вектором $(n_1, n_2, \ldots, n_k)$.

Асимптотическая ППС для ВР

Количество различных типов не превышает $r = C_{m+k}^{m}$

Общее число ящиков не превосходит *n*.

// Это сильно завышенная оценка и на практике

// лучше брать не *n*, а стоимость приближённого

// решения, найденного FF или BF.

Поэтому количество возможных упаковок не превышает

$$p = C_{n+r-1}^{r}$$

Это величина, полиномиально зависящая от *n*. Перебрав все варианты и выбрав наилучший, мы найдём точное решение за полиномиальное время. ЧТД

Асимптотическая ППС для ВР

Каким же способом можно найти точное решение?

- Полный перебор вариантов.

$$p = C^{r}_{n+r-1}$$, где $$r = C^{m}_{m+k}$$

Это действительно полином от *n* (при фиксированном ε), но...

Асимптотическая ППС для ВР

Каким же способом можно найти точное решение?

- Полный перебор вариантов.

$p = C_{n+r-1}^{r}$, где $r = C_{m+k}^{m}$

Это действительно полином от *n* (при фиксированном ε), но...

Для n = 20

Асимптотическая ППС для ВР

Каким же способом можно найти точное решение?

- Полный перебор вариантов.

$p = C_{n+r-1}^{r}$, где $r = C_{m+k}^{m}$

Это действительно полином от *n* (при фиксированном ε), но...

Для n = 20

при ε=0.3:

Асимптотическая ППС для ВР

Каким же способом можно найти точное решение?

- Полный перебор вариантов.

$p = C^r_{n+r-1}$, где $r = C^m_{m+k}$

Это действительно полином от *n* (при фиксированном ε), но...

Для n = 20

при ε=0.3: p = 6288988015652093750613680284237 5

Асимптотическая ППС для ВР

Каким же способом можно найти точное решение?

- Полный перебор вариантов.

$p = C_{n+r-1}^{r}$, где $r = C_{m+k}^{m}$

Это действительно полином от *n* (при фиксированном ε), но...

Для n = 20

при ε=0.3: p = 62889880156520937506136802842375

при ε=0.2:

Асимптотическая ППС для ВР

Каким же способом можно найти точное решение?

- Полный перебор вариантов.

$p = C^{r}_{n+r-1}$, где $r = C^{m}_{m+k}$

Это действительно полином от *n* (при фиксированном ε), но...

Для n = 20

при ε=0.3: p = 62889880156520937506136802842375

при ε=0.2: p = 215767663434879982271362684862329942665970184139098118545344624819826592385004

Асимптотическая ППС для ВР

- Динамическое программирование.

  Сложность: $O(n^{2k})$

- Линейное программирование.

  Сложность: $O(C_{\varepsilon} + n \log(1/\varepsilon))$

Асимптотическая ППС для ВР

Динамическое программирование

1. Разработать рекурсивную процедуру, вычисляющую решение для большой задачи на основе решений подзадач меньшего размера.
2. Для самых маленьких задач значения находятся с помощью другого (нерекурсивного) алгоритма.
3. В цикле рассчитать решения подзадач от маленьких к большим.
4. Все построенные решения сохранять.

Асимптотическая ППС для BP

Пусть bins($q_1,q_2,\ldots,q_k$) = количество ящиков, необходимых для упаковки множества объектов, содержащего для каждого i q_i предметов i-го типа.

1. Построить и запомнить множество Q наборов таких ($q_1,q_2,\ldots,q_k$), что bins($q_1,q_2,\ldots,q_k$)=1. Количество таких наборов не превышает O(n^k) и может быть найдено за такое же время.
2. Создать «таблицу» для всех возможных наборов ($i_1,i_2,\ldots,i_k$), для наборов из Q проинициализировать 1.
3. Последовательно заполнить по принципу:

$$\text{bins}(i_1,i_2,\ldots,i_k) = 1 + \min \{ \text{bins}(i_1-q_1,i_2-q_2,\ldots,i_k-q_k) : q \in Q\}$$

Сложность: O(n^{2k})

Асимптотическая ППС для ВР

Таким образом, ограниченную задачу можно точно решить за полиномиальное время. Теперь необходимо свести исходную задачу к неограниченной.

Вспомним, что такое ограниченный вариант задачи:

1. веса всех предметов не меньше ε;
2. количество различных весов предметов равно (или не превышает) заданному натуральному числу k.

Рассмотрим задачу, в которой сохраняется только 1-е ограничение.

Асимптотическая ППС для BP

<u>Лемма 2</u>. Существует полиномиальный алгоритм, который для любого набора, в котором веса всех предметов не меньше ε, находит упаковку с использованием не более $(1+\varepsilon)$opt ящиков.

Доказательство

Пусть x — исходный набор предметов.

Обозначим: $k = \lceil 1/\varepsilon^2 \rceil$, $p = \lfloor n\varepsilon^2 \rfloor$.

Упорядочим все предметы по неубыванию весов и разобъём на k групп, в каждой (кроме, может быть, последней) по p предметов. (Может оказаться так, что в двух группах будут предметы одинакового веса.)

Асимптотическая ППС для ВР

Построим новый набор x', заменяя вес каждого предмета на максимальный вес предмета из той же группы. Тогда в x' количество различных значений веса не превышает *k*, вес каждого предмета $\geq \varepsilon$. По лемме 1, решение для x' можно найти за полиномиальное время.

Покажем, что $opt(x') \leq (1+\varepsilon)\ opt(x)$.

Для этого построим ещё один набор — x'', аналогично x', но вес предмета будем заменять на *минимальный* вес в группе.

Очевидно, что $opt(x'') \leq opt(x)$.

Асимптотическая ППС для BP

Каждому предмету i из q-й группы набора x' соответствует (взаимно однозначно) некоторый предмет $f(i)$ из группы (q+1) набора x'', таким образом, что вес i не превышает вес $f(i)$. Это справедливо для $q < k$.

Найдём оптимальную упаковку для x''.

Построим теперь решение для x'.

- Предметы i из всех групп $q < k$ разместим в тех же ящиках,что и соответствующие им $f(i)$.
- Предметы из последней (k-й) группы разместим в отдельных дополнительных ящиках.

Получаем: $\text{opt}(x') \leq \text{opt}(x'') + p \leq \text{opt}(x) + p.$

Асимптотическая ППС для BP

Получаем: opt(x') $\leq$ opt(x'') + $p \leq$ opt(x) + p.

Поскольку в x вес каждого предмета не меньше ε, справедлива оценка: opt(x) $\geq n\varepsilon$. Поэтому $p = \lfloor n\varepsilon^2 \rfloor \leq \varepsilon$ opt(x) .

Получаем: opt(x') $\leq (1 + \varepsilon)$ opt(x).

ЧТД.

Леммы 1 и 2 касались больших предметов (вес $\geq \varepsilon$) - множество Big. Их можно упаковать за полиномиальное время.

А что делать с маленькими предметами? Нам-то надо упаковать x = Big $\cup$ Small.

Асимптотическая ППС для BP

Лемма 3. Пусть известна упаковка Big в $(1 + \varepsilon)$ opt(x) ящиков. Тогда за полиномиальное время можно найти упаковку x = Big $\cup$ Small в не более чем $(1+2\varepsilon)$ opt(x) + 1 ящиков.

Доказательство

К известной упаковке Big добавим предметы из Small, используя алгоритм «Первый подходящий» (FF) или «Лучший подходящий» (BF). Получим упаковку в *B* ящиков.

Если не пришлось создать ни одного нового ящика, то условие выполнено, в силу $B \leq$ opt(Big) $\leq$ opt(x).

Асимптотическая ППС для BP

Допустим, пришлось пришлось создать новые ящики.

В худшем случае (таком, при котором значение *B* максимально) все маленькие предметы размещены в новых ящиках. Это значит, что каждый старый ящик был заполнен не менее чем на $1-\varepsilon$. Старых ящиков — не менее *B*-1 штук.

Поэтому $(B-1)(1-\varepsilon) \le$ сумма весов предметов в Big $\le$ сумма весов всех предметов $\le opt(x)$.

Отсюда: $B \le opt(x) / (1-\varepsilon) + 1 \le (1+2\varepsilon)\, opt(x) + 1$.

Асимптотическая ППС для BP

Итого, алгоритм Fernandez de la Vega — Lueker (FVL):

1. Разделить весь набор предметов на два:
   - Small = $\{i: w_i < \varepsilon\}$
   - Big = $\{i: w_i \geq \varepsilon\}$
2. Упаковать предметы из Big, используя подход из доказательства леммы 2.
3. К полученной упаковке добавить предметы из Small, применяя алгоритм «Первый подходящий» (FF) или «Лучший подходящий» (BF).

Сложность: $O(n^{2/\varepsilon^2})$

Приближённые схемы

Т.е. недостаток ППС — произвольная зависимость от ε, в т.ч. такая как $O(n^{1/\varepsilon})$.

Более практичный вариант:

Полностью полиномиальная приближённая схема (fully polynomial-time approximation scheme, FPTAS) — приближённый алгоритм, который для заданного экземпляра задачи и числа $\varepsilon>0$ находит $(1+\varepsilon)$-приближённое решение за время, полиномиально зависящее от n и $1/\varepsilon$.

FPTAS — класс задач, для которых существуют ПППС.

Известно, что FPTAS $\subset$ PTAS $\subset$ APX.

Приближённые схемы

Рассмотрим задачи с числовыми параметрами, т. е. задачи, для которых входные данные содержат набор чисел, который может варьироваться для одного и того же входа.

Примеры: «Упаковка ящиков», «Рюкзак», «Коммивояжёр».

Определение. NP-трудная задача называется *NP-трудной в сильном смысле*, если она либо не содержит числовых параметров, либо является NP-трудной даже при ограниченных (полиномом от длины входа) значениях параметров.

Теорема. При P $\neq$ NP ни для какой задачи, NP-трудной в сильном смысле, не может существовать полностью полиномиальная приближённая схема.

NP-трудными в сильном смысле являются: «Упаковка ящиков», «Коммивояжёр» :(

ПППС для «Рюкзака»

Снова рассмотрим задачу о рюкзаке.

Вспомним рассмотренный ранее метод динамического программирования.

Без потери общности можно считать веса и стоимости предметов целыми числами.

Пусть b — вместимость рюкзака, n — количество предметов.

Алгоритм:

- Создадим двумерную таблицу A. Значение A[i, r] = минимальный суммарный вес предметов из $\{1, \ldots, i\}$, имеющих суммарную стоимость ровно r (и суммарный вес, не превышающий b).
- Инициализация:
  - $A[0, r] = +\infty$
  - $A[i, 0] = 0$ для всех $i=0, \ldots, n$

ПППС для «Рюкзака»

- Заполняем ячейки таблицы, в цикле по i=1,..., *n* и по *r*.
  - Если $c_i \le r$, то A[*i*, *r*] =min { A[*i* - 1, *r* - c_i] + w_i , A[*i* - 1, *r*] }.

    Первый вариант соответствует ситуации «кладём i-й предмет в рюкзак»; второй вариант - «не кладём».
  - Иначе A[*i*, *r*] =A[*i* - 1, *r*].
  - Во вспомогательной таблице *P* запоминаем, какой из вариантов выбора значения использован.
- Стоимость оптимального решения = значение в самой правой ячейке нижней строки, не превышающее *b*.
- Для реконструкции решения проходим по P от этой ячейки до верхней строки.

ПППС для «Рюкзака»

Каковы должны быть размеры таблицы?

Строки индексируются от 0 до *n*.

Колонки соотвествуют возможным суммарным стоимостям. В идеале должны индексироваться от 0 до c^*, где $c^* = \mathrm{opt}(x)$.

Проблема: мы не знаем значение c^*. Поэтому вместо c^* используем оценку сверху: $\bar{c}$. Простая, но грубая оценка: $\bar{c} = n\, c_{max}$.

Временная сложность: $O(n^2\, c_{max})$.

Это *псевдополиномиальный* алгоритм — экспоненциальный в общем случае (почему?), но полиномиальный при ограниченных числовых параметрах.

ПППС для «Рюкзака»

Т.е. проблема возникает при больших значениях $\{c_i\}$.

Что, если уменьшить (*масштабировать*) эти параметры?

- Выберем число t.
- Построим масштабированный экземпляр задачи x': столько же предметов, с теми же весами, но $c'_i = \lfloor c_i / 2^t \rfloor$.
- Решим масштабированную задачу с помощью дин. программирования — это можно сделать в 2^t раз быстрее, чем исходную!
- Решение (набор предметов) масштабированной задачи используем как приближённое решение исходной задачи. Это допустимое решение, т. к. веса мы не изменяли.

ПППС для «Рюкзака»

Пусть apx(x) — стоимость такого приближённого решения.

Во сколько раз apx(x) меньше, чем opt(x)?

Это определяется коэффициентом масштабирования *t*.

Очевидно, что (opt - apx) $\leq$ n 2^t.

Выбрем максимальное *t*, такое, что n $2^t \leq \varepsilon /2$ * opt. (#)

Тогда:

$$apx(x) \geq opt - n\, 2^t \geq opt - \varepsilon /2 * opt = (1- \varepsilon /2)\, opt \geq opt \,/\, (1 + \varepsilon)$$

Получаем, что коэффициент приближения R = $(1 + \varepsilon)$.

ПППС для «Рюкзака»

Заметим, что условие

Выберем максимальное t, такое, что $n\, 2^t \leq \varepsilon /2 * opt$. (#)

некорректно, потому что мы не знаем величину opt. Поэтому вместо opt снова используем оценку, но уже снизу: $opt \geq c_{max}$. Итого, получим условие:

$$n\, 2^t \leq \varepsilon /2 * c_{max}. \qquad (\#\#)$$

Осталось оценить временную сложность.

Сложность точного решения масштабированной задачи:

$O(n^2 c'_{max}) = O(n^2 c_{max} / 2^t)$.

В силу выбора t имеем: $n\, 2^{t+1} \geq \varepsilon /2 * opt \geq \varepsilon /2 * c_{max}$ (###)

Поэтому временная сложность оценивается как $O(n^3 / \varepsilon)$ - полином от n и от $1/\varepsilon$.

Задание 6

Реализовать ПППС для задачи «Рюкзак».

Входные данные — как для задания 1, но добавляется параметр ε.