Иванов А.Б.

Алгоритмы и структуры данных

NP-полные задачи

Решение NP-сложных задач

Алгоритм дает:

- решение NP-полной задачи
- точное решение
- быстрое решение

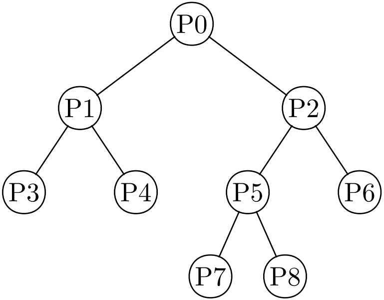
выбирайте любые два пункта

Что делать с NP-полной задачей?

- Частный случай NP-полной задачи?
  Может существовать эффективный алгоритм:
  - 2-SAT
  - максимальное независимое множество для деревьев
  - вершинное покрытие в двудольном графе (Теорема Кёнига)
  - задача о рюкзаке с $W = O(n^k)$
- Точные методы
  - задача о рюкзаке: $O(nW)$ вместо 2^n
  - задача коммивояжёра: $O(n^2 2^n)$ вместо $n!$
  - метод ветвей и границ
- Приближенные методы
  - задача о рюкзаке: $\frac{1}{2}$-приближение
  - вершинное покрытие: 2-приближение
  - задача о K-центрах: 2-приближение

Метод ветвей и границ

Ветвление строит дерево задач



- Обязательно ли надо обойти все дерево?
- Как выбирать очередной узел для ветвления?

Пример: рюкзак

$v_1 = 40,\ v_2 = 42,\ v_3 = 25,\ v_4 = 12$

$w_1 = 4,\ w_2 = 7,\ w_3 = 5,\ w_4 = 3$

$\frac{v_1}{w_1} = 10,\ \frac{v_2}{w_2} = 6,\ \frac{v_3}{w_3} = 5,\ \frac{v_4}{w_4} = 4$

$W = 10$

Аппроксимационный алгоритм

Полиномиальный алгоритм для оптимизационной задачи дает $\rho(n)$-приближенное решение $\rho(n)$, если

$$C \leq (1 + \rho(n))C_{OPT}$$

для задачи минимизации или

$$C \geq (1 - \rho(n))C_{OPT}$$

для задачи максимизации.

Пример: задача о рюкзаке

Жадный алгоритм.

1. Упорядочим предметы
   по убыванию относительной ценности v_i/w_i.

$$\frac{v_1}{w_1} \geq \frac{v_2}{w_2} \geq \frac{v_3}{w_3} \geq \dots$$

2. Кладем в рюкзак предметы $1, 2, \dots k$, пока они помещаются.
   Получили некоторое решение с ценностью
   $v_{greedy} = v_1 + \ldots + v_k$.
3. Если есть предмет (w_{max}, v_{max}): $w_{max} \leq W$, $v_{greedy} < v_{max}$, то вместо решения, полученного на шаге 2 берем решение, состоящее из одного этого предмета.

Утв. Ценность решения по «хорошему» жадному алгоритму не меньше $\frac{1}{2}$ ценности оптимального решения ($\frac{1}{2}$-аппроксимация).

Пример: Вершинное покрытие

Жадный алгоритм? Какие вершины брать?
Идея: брать вершины с наибольшей степенью.
Идея так себе...
Двудольный граф:

$$S_L, |S_L| = t$$

$$S_{R,i}, |S_{R,i}| = \frac{t}{i}$$

У каждой вершины из $S_{R,i}$ i соседей в S_R, у всех разные.

Пример: Вершинное покрытие

Алгоритм, дающий 2-приближение:

1. $C = \varnothing$
2. берем *произвольно ребро* (u, v), добавляем u и v в C
3. удаляем все ребра инцидентные u и v
4. повторяем пока еще есть ребра

Пример: Вершинное покрытие

Гипотеза: не существует лучшего приближения (если верна UGC)

Пример: Задача K-центра

Мы видели (жадный) алгоритм дающий 2-приближение.
Утв 2. Это лучший приближенный алгоритм.

Вспомним сведение задачи о вершинном покрытии к задаче K-центра:
Граф (G, V) имеет вершинное покрыие размера $k \Leftrightarrow$ у задачи k-центра есть решение с бюджетом 1.

Приближенные методы

- задача о рюкзаке: $\frac{1}{2}$-приближение
- покрытие множествами: $\ln n$-приближение
- вершинное покрытие: 2-приближение

- задача о рюкзаке: $(1-\varepsilon)$-приближение
- метрический коммивояжер: $\frac{1}{2}$-приближение (есть и $\frac{3}{2}$)

Аппроксимационная схема

Аппроксимационная схема - семейство алгоритмов такое, что для любого ε есть полиномиальный алгоритм дающий ε-приближение. Сложность, конечно, зависит от ε. Например:

$$O(n^{\frac{2}{\varepsilon}}) \qquad (PTAS)$$

или

$$O(\frac{n}{\varepsilon^2}) \qquad (FPTAS)$$

Пример: задача о рюкзаке

Пусть стоимости v_i – целые числа.

Подзадачи: для $i = 0, 1, 2 \ldots n$ и $x = 0, 1, 2, \ldots n \cdot v_{max}$
$A[i, x] =$ минимально нужный размер для того, чтобы достичь стоимости $\geq x$, используя только первые i предметов.

Рекурентное соотношение:

$$A[i, x] = \min\Big(A[i-1, x], w_i + A[i-1, x - v_i]\Big)$$

(считаем $A[i-1, x - v_i] = 0$, если $x \leq v_i$).

Сложность: $O(n^2 v_{max})$

Пример: задача о рюкзаке

Идея: уменьшим v_{max}.

$$\theta = \frac{\varepsilon v_{max}}{n} \qquad \bar{v_i} = \left\lceil \frac{v_i}{\theta} \right\rceil \theta \qquad \hat{v_i} = \left\lceil \frac{v_i}{\theta} \right\rceil$$

Используем алгоритм динамического программирования и решим задачу для $\{\hat{v_i}\}$, получим решение S. Тогда сложность его будет:

$$O(n^2 \hat{v}_{max}) = O(n^2 \left\lceil \frac{v_{max}}{\theta} \right\rceil) = O(n^2 \left\lceil \frac{n}{\varepsilon} \right\rceil) = O(\frac{n^2}{\varepsilon})$$

Почему это ε-приближение:

$$\sum_{i \in S^*} v_i \le \sum_{i \in S^*} \bar{v}_i \le \sum_{i \in S} \bar{v}_i \le \sum_{i \in S} (v_i + \theta) \le \varepsilon v_{max} + \sum_{i \in S} v_i$$

Приближение задачи коммивояжера