Адрианов Н.М.
Иванов А.Б.

Алгоритмы и структуры данных

Методы решения NP-полных задач

Что делать с NP-полной задачей?

- Частный случай NP-полной задачи?
  Может существовать эффективный алгоритм:
  - 2-SAT
  - максимальное независимое множество для деревьев
  - 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
- Эвристики
  - жадный алгоритм для рюкзака

Точные методы

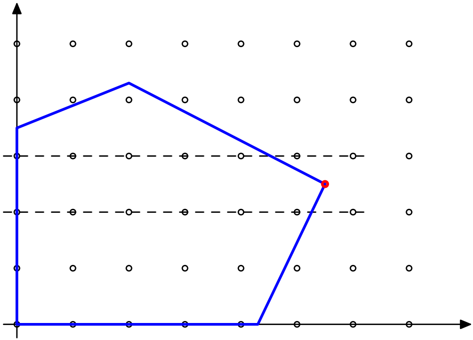
Работают за экспоненциальное время, но сокращают перебор.

- задача о рюкзаке: $O(nW)$ вместо 2^n
- задача коммивояжёра: $O(n^2 2^n)$ вместо $n!$
- метод ветвей и границ
- DPLL и CDCL алгоритмы в задаче SAT

Имеет смысл использовать солверы стандартных задач.

Метод ветвей и границ: ЦЛП

Пусть x^* – решение задачи ЛП, выберем $x_i^* \notin \mathbb{Z}$



Рассмотрим две новые задачи, добавив ограничения

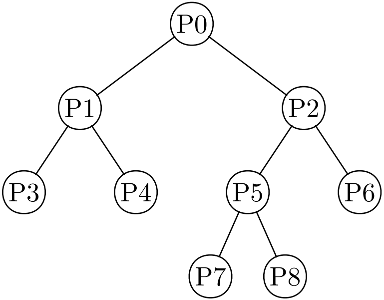
1. $x_i \leq [x_i^*]$ 2. $x_i \geq [x_i^*] + 1$

Решение исходной задачи = лучшее из решений новых задач

Метод ветвей и границ

Ветвление строит дерево задач

1. В начале: $best = -\infty$
2. При нахождении листа с целевым значением z $best = \max(best, z)$



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

Метод ветвей и границ: коммивояжёр

Задача коммивояжёра: найти гамильтонов путь минимального веса.

Идея: путь является частным случаем дерева. Гамильтонов путь – частный случай покрывающего дерева. У нас есть эффективные алгоритмы для нахождения минимального покрывающего дерева.

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: $\frac{3}{2}$-приближение

Задача о рюкзаке: еще одна динамика

Пусть стоимости v_i – целые числа.

Подзадачи: для $i = 0, 1, 2 \dots n$ и $x = 0, 1, 2, \dots n \cdot v_{max}$
$A[i, x]$ = 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 $x \leq v_i$).

Задача о рюкзаке: еще одна динамика

```
procedure KnapsackDynamo( W, v, w )
  A = 2-мерный массив
  A[0, x] = 0, если x = 0
          +∞, иначе

  for i = 1,2,...n
    for x = 0,1,... n*v_max
      A[i,x] = min ( A[i-1,x], w[i] + A[i-1, x - v[i]] )

  return максимальное x такое, что A[n,x] <= W.
```

$O(n^2 v_{max})$

$(1-\varepsilon)$-ПРИБЛИЖЕНИЕ ДЛЯ ЗАДАЧИ О РЮКЗАКЕ

Приближенный алгоритм с использованием динамики

1. выбираем m, составляем новую задачу о рюкзаке:
   $v_i' = [v_i/m]$, $\quad w_i$ и W не меняются.
2. решаем новую задачу за $O(n^2 \max(v_i'))$

Из определения v_i' имеем: $\qquad v_i - m \le m \cdot v_i' \le v_i$.
Пусть S – наше решение, S^* – оптимальное решение.

$$\sum_{i \in S} v_i \ge m \sum_{i \in S} v_i' \ge m \sum_{i \in S^*} v_i' \ge \sum_{i \in S^*} (v_i - m)$$

$$\sum_{i \in S} v_i \ge \sum_{i \in S^*} v_i - mn$$

Нам нужно получить

$$\sum_{i \in S} v_i \ge (1-\varepsilon) \sum_{i \in S^*} v_i$$

т.е. $mn \le \varepsilon \cdot \sum_{i \in S^*} v_i$. Достаточно взять $m = \varepsilon \cdot v_{max}/n$.