

Задачи о рюкзаке

Модель размещения капитала

Дано $P_1, P_2, \dots, P_N$ — проекты;

T — горизонт планирования (длина наиболее продолжительного проекта);

s_{tk} — доход от проекта P_k к концу года t ;

y_{tk} — инвестиции в проект P_k в начале года t ; $s_{0k} = y_{T+1k} = 0$;

r — коэффициент дисконтирования затрат

$b_k = \sum_{t=0}^T (s_{tk} - y_{t+1,k}) / (1+r)^t$ — суммарная прибыль от проекта P_k ;

$C = (c_1, \dots, c_T)$ — доступный капитал для развития проектов

$A_k = (a_{1k}, \dots, a_{Tk})$ — вектор затрат на реализацию проекта P_k (целые);

Если доход нельзя реинвестировать, то $a_{tk} = y_{tk}$, иначе $a_{tk} = y_{tk} - s_{t-1k}$.

Найти подмножество проектов, которые можно реализовать на капитал C и которые в сумме дают максимальную прибыль, то есть

$$\max \sum_{k=1}^N b_k x_k$$

при ограничениях

$$\sum_{k=1}^N a_{tk} x_k \leq c_t, \quad t = 1, \dots, T$$

$$x_k \in \{0, 1\}, \quad k = 1, \dots, N$$

Замечание 1. При $T = 1$ получаем линейную распределительную задачу с 0-1 переменными — задачу о рюкзаке.

Замечание 2. Без ограничения общности можно считать, что $\sum_{k=1}^N a_{tk} \geq c_t$, $t = 1, \dots, T$, (можно получить задачу о рюкзаке даже при $T > 1$).

Алгоритм динамического программирования

Обозначим через $f_k(Y)$ максимальную прибыль от первых k проектов при доступном капитале $Y = (y_1, \dots, y_T)$.

Тогда $f_0(Y) = 0$

$$f_{k+1}(Y) = \max [f_k(Y), b_{k+1} + f_k(Y - A_{k+1})], \quad k = 0, \dots, N-1, \quad 0 \leq Y \leq C,$$

где $f_k(Y - A_{k+1}) = -\infty$, если вектор $Y - A_{k+1}$ имеет хотя бы одну отрицательную компоненту.

$$T_{ДП} = O(N \cdot c_1 \cdot \dots \cdot c_T);$$

$$П_{ДП} = O(N \cdot c_1 \cdot \dots \cdot c_T).$$

Полный перебор — 2^N вариантов.

Верхняя оценка

Релаксация линейного программирования

$$\max \sum_{k=1}^N b_k x_k \quad (1)$$

при ограничениях

$$\sum_{k=1}^N a_{tk} x_k \leq c_t, \quad t = 1, \dots, T \quad (2)$$

$$0 \leq x_k \leq 1, \quad k = 1, \dots, N. \quad (3)$$

Теорема 2.1. Существует оптимальное решение x^{LP} с не более чем $\min(T, N)$ дробными компонентами

Доказательство. Пусть $T < N$ (иначе утверждение очевидно). Приведем задачу к канонической форме. Получим $2N + T$ переменных и $N + T$ ограничений:

$$\min \sum_{k=1}^N -b_k x_k, \quad (4)$$

$$\sum_{k=1}^N a_{tk} x_k + \lambda_t = c_t, \quad t = 1, \dots, T, \quad (5)$$

$$x_j + \mu_j = 1, \quad j = 1, \dots, N, \quad (6)$$

$$x_j \geq 0, \quad \mu_j \geq 0, \quad \lambda_t \geq 0. \quad (7)$$

Любое базисное допустимое решение имеет не менее N нулей. Предположим, что T из них соответствуют переменным λ_t . Тогда $N - T$ нулей останется для x_j и μ_j . Если для некоторого j имеем $\mu_j = 0$, то $x_j = 1$ — целое. Если $x_j = 0$ — тоже целое. Таким образом, получаем $N - T$ целых компонент для x_j , то есть T дробных. ■

Округление дробного решения

Пусть x^{LP} — оптимальное решение задачи (4)–(7). Для $\gamma \in [0,1]$ положим

$$x_j = \begin{cases} 1, & \text{если } x_j^{LP} = 1 \\ 0, & \text{если } x_j^{LP} < \gamma \end{cases}$$

Для оставшихся дробных значений переменных сформируем подзадачу вида (4)–(7), пересчитав правые части ограничений. Найдем оптимальное решение x^{LP} для этой подзадачи и положим

$$x_j = \begin{cases} 1, & \text{если } x_j^{LP} = 1, \\ 0, & \text{если } x_j^{LP} = 0, \\ 0 & \text{для } j = \arg \min \{x_j^{LP} \mid 0 < x_j^{LP} < 1\}. \end{cases}$$

На этом шаге значение как минимум одной переменной будет зафиксировано. Повторяя процедуру, найдем допустимое решение исходной задачи.

Вопросы

- Пусть T — трудоемкость решения задачи линейного программирования. Оцените трудоемкость алгоритма округления дробного решения.
- Пусть P — объем памяти для решения задачи линейного программирования. Оцените объем памяти алгоритма округления дробного решения.
- Пусть x^{LP} — оптимальное решение задачи линейного программирования. Если все компоненты вектора x^{LP} целые, то x^{LP} — оптимальное решение исходной целочисленной задачи (*Да или Нет?*)
- Если некоторые компоненты вектора x^{LP} целые, то эти целые значения сохранятся и в оптимальном решении исходной целочисленной задачи (*Да или Нет?*)

Классическая задача о рюкзаке

Найти:

$$\max \sum_{j \in J} p_j x_j$$

при ограничениях

$$\sum_{j \in J} w_j x_j \leq C,$$

Все коэффициенты p_j , w_j , C — целые числа. $x_j \in \{0, 1\}$, $j \in J$.

Определение Алгоритм A называется *приближенным алгоритмом с гарантированной абсолютной точностью K* , если для любого примера I алгоритм находит значение $z^A(I)$ с отклонением от оптимума $z^*(I)$ не более K , то есть

$$z^*(I) - z^A(I) \leq K, \quad \text{для всех } I.$$

Обозначим через $T_A(n, C)$ трудоемкость алгоритма A для задачи с n предметами и вместимостью рюкзака C .

Теорема 2.2. Пусть A — приближенный алгоритм с гарантированной абсолютной точностью K и трудоемкостью $T_A(n, C)$. Тогда алгоритм A для любого примера позволяет найти точное решение задачи о рюкзаке с той же трудоемкостью.

Доказательство. Пример I задается числами $p_1, \dots, p_n, w_1, \dots, w_n, C$. Построим новый пример I' , положив $C' = C, p'_j = (K + 1)p_j, w'_j = w_j, j \in J$. Оба примера имеют одно и то же множество допустимых решений. Так как целевая функция для I' в $(K + 1)$ раз больше, чем для I , то оптимальные наборы x_j^* совпадают.

Для примера I' имеем $z^*(I') - z^A(I') \leq K$, но $z^A(I') = (K + 1) z^A(I)$ и $z^*(I') = (K + 1) z^*(I)$, то есть $z^*(I) - z^A(I) \leq \frac{K}{K + 1}$

Так как p_j — целые, то $z^*(I) - z^A(I) \leq 0$, то есть $z^*(I) = z^A(I)$, что и требовалось доказать. ■

Вопросы

- Правда ли, что для задачи размещения капитала трудоемкость точного решения задачи совпадает с трудоемкостью приближенного решения с гарантированной абсолютной точностью?

Алгоритм A называют *приближенным алгоритмом с гарантированной относительной точностью* K , если для любого примера I он находит такое значение $z^A(I)$, что $z^A(I) \geq K \cdot z^*(I)$ для всех I .

- Верно ли аналогичное утверждение для приближенных решений с гарантированной *относительной* точностью?

Жадные алгоритмы

Упорядочим предметы по плотности p_j / w_j и будем считать, что

$$p_1 / w_1 \geq p_2 / w_2 \geq \dots \geq p_n / w_n$$

Жадный алгоритм

1. $\bar{w} := 0; \quad z^G := 0;$

2. for $j := 1$ to n do

 if $\bar{w} + w_j \leq C$ then

$x_j := 1; \quad \bar{w} := \bar{w} + w_j; \quad z^G := z^G + p_j;$

 else $x_j := 0;$

$$T_G = O(n \log n + n), \quad \Pi_G = O(n)$$

Упражнение Если последнюю строку заменить на else {for $k := j$ to n do $x_k := 0$; break}, то такое решение можно найти с $T = O(n)$.

Релаксация линейного программирования

LP–релаксация

$$\begin{aligned} z^{LP} &= \max \sum_{j \in J} p_j x_j \\ \sum_{j \in J} w_j x_j &\leq C, \\ 0 \leq x_j &\leq 1, \quad j \in J. \end{aligned}$$

Так как область допустимых решений увеличилась, то $z^{LP} \geq z^*$. Пусть предметы упорядочены по плотностям и для некоторого $s \in J$ верно:

$$\sum_{j=1}^{s-1} w_j \leq C \quad \text{и} \quad \sum_{j=1}^s w_j > C.$$

Положим

$$x_j^{LP} = \begin{cases} 1, & j = 1, \dots, s-1, \\ \frac{1}{w_s} (C - \sum_{j=1}^{s-1} w_j), & j = s, \\ 0, & j = s+1, \dots, |J|. \end{cases}$$

Теорема 2.3. Решение x^{LP} является оптимальным решением LP –релаксации и

$$z^{LP} = \sum_{j=1}^{s-1} p_j + \frac{p_s}{w_s} (C - \sum_{j=1}^{s-1} w_j).$$

Доказательство. Будем считать, что предметы с одинаковой плотностью слиты в один и

$$\frac{p_1}{w_1} > \frac{p_2}{w_2} > \dots > \frac{p_n}{w_n}.$$

Пусть $\bar{x} = (\bar{x}_1, \dots, \bar{x}_n)$ — оптимальное решение, не равное x^{LP} . Так как $\sum_{j \in J} w_j \bar{x}_j = C$, то найдутся как минимум два номера $k > s$ и $i \leq s$ такие, что

$\bar{x}_k > 0$ и $\bar{x}_i < x_i^{LP}$. Положим $d = \min\{w_k \bar{x}_k, w_i(x_i^{LP} - \bar{x}_i)\} > 0$. Построим новое решение x' , которое будет отличаться от $\bar{x}$ только в координатах i, k :

$$x'_i = \bar{x}_i + \frac{d}{w_i}, \quad x'_k = \bar{x}_k - \frac{d}{w_k}.$$

Решение x' является допустимым, так как $x'_j \geq 0$ по выбору d и

$$\sum_{j \in J} w_j x'_j = \sum_{j \in J} w_j \bar{x}_j + \frac{w_i d}{w_i} - \frac{w_k d}{w_k} = C.$$

Кроме того

$$\sum_{j \in J} p_j x'_j = \sum_{j \in J} p_j \bar{x}_j + d \left(\frac{p_i}{w_i} - \frac{p_k}{w_k} \right) > \sum_{j \in J} p_j \bar{x}_j$$

так как $\frac{p_i}{w_i} > \frac{p_k}{w_k}$, что противоречит оптимальности $\bar{x}$. ■

Свойства LP -релаксации

Верхняя оценка $U^{LP} = \lfloor z^{LP} \rfloor$, $\hat{p} = \sum_{j=1}^{s-1} p_j$

Свойство 1. $\hat{p} \leq z^* \leq U^{LP} \leq z^{LP} \leq \sum_{j=1}^s p_j \leq \hat{p} + p_s \leq z^G + p_s$.

Свойство 2. $z^* - z^G \leq z^* - \hat{p} \leq p_{\max}$, где $p_{\max} = \max_{j \in J} p_j$.

Свойство 3. $z^{LP} \leq 2z^*$ и для любого $\varepsilon > 0$ найдется пример задачи о рюкзаке такой, что $z^{LP} \geq 2z^* - \varepsilon$.

Доказательство. 1. Так как $z^* \geq \sum_{j=1}^{s-1} p_j$ и $z^* \geq p_s$ то $2z^* \geq z^{LP}$.

2. Рассмотрим пример $n = 2$, $C = 2M$ и $w_j = M + 1$, $p_j = 1$, $j = 1, 2$. Тогда $z^* = 1$, но $z^{LP} = \frac{2M}{M+1}$ и с ростом M получаем $z^{LP} / z^* \rightarrow 2$. ■

Определение Алгоритм A называется *приближенным алгоритмом с гарантированной относительной точностью* K , если для любого примера I

алгоритм находит значение $z^A(I)$ такое, что $\frac{z^A(I)}{z^*(I)} \geq K$ для всех I .

Если $\varepsilon = 1 - K$, то $\frac{z^*(I) - z^A(I)}{z^*(I)} \leq \varepsilon$ — относительная погрешность алгоритма.

Пример. Положим $n = 2$, $C = M$ и $p_1 = 2$, $w_1 = 1$, $p_2 = M$, $w_2 = M$. Тогда жадный алгоритм получит $x_1 = 1$, $x_2 = 0$, $z^A = 2$,

но $x_1^* = 0$, $x_2^* = 1$, $z^* = M$, то есть для жадного алгоритма

$$\frac{z^A}{z^*} \rightarrow 0 \text{ при } M \rightarrow \infty.$$

Модифицированный жадный алгоритм

Используем предыдущий жадный алгоритм, получаем z^G .

Затем полагаем $z^{MG} = \max \{z^G, \max \{p_j \mid j \in J\}\}$.

Теорема 2.4. Модифицированный жадный алгоритм A^{MG} имеет гарантированную относительную точность $K = 1/2$.

Доказательство. Из свойства 2 для LP -релаксации имеем

$$z^* \leq z^G + p_{\max} \leq z^{MG} + z^{MG}. \quad \blacksquare$$

Пример. Положим $n = 3, C = 2M$,

$$p_1 = 2, \quad p_2 = M, \quad p_3 = M,$$

$$w_1 = 1, \quad w_2 = M, \quad w_3 = M.$$

Получаем $z^* = 2M$, $z^{MG} = 2 + M$, то есть оценку $K = 1/2$ нельзя улучшить.

Алгоритм $G^{3/4}$

Сокращаем погрешность за счет трудоемкости

Алгоритм $G^{3/4}$

1. $z^A := \max \{p_j \mid j \in J\};$
2. Для всех пар $(i, k) \in J \times J$
if $w_i + w_k \leq C$ then
 - 2.1 применяем алгоритм A^{MG} к задаче с множеством
 $\{j \mid p_j \leq \min\{p_i, p_k\}\} \setminus \{i, k\}$
и вместимостью рюкзака $C - w_i - w_k$
 - 2.2 if $p_i + p_k + z^{MG} > z^A$ then $z^A := p_i + p_k + z^{MG}.$

Теорема 2.5. Алгоритм $G^{3/4}$ имеет гарантированную относительную точность $K = 3/4$.

Доказательство. Если оптимальное решение x_j^* содержит только один предмет, то $z^A = z^*$ и утверждение верно. Предположим, что в оптимальном решении не меньше двух предметов. Выберем среди них два (i_*, k_*) с наибольшими p_j . На некотором шаге алгоритм $G^{3/4}$ выберет эту пару (i_*, k_*) и применит алгоритм A^{MG} к задаче с множеством предметов $\{j \mid p_j \leq \min\{p_{i_*}, p_{k_*}\}\} \setminus \{i_*, k_*\}$ и вместимостью рюкзака $C - w_{i_*} - w_{k_*}$.

Обозначим через z_s^* оптимальное решение этой подзадачи. Тогда $z^* = p_{i_*} + p_{k_*} + z_s^*$. Алгоритм A^{MG} для этой подзадачи найдет значение z_s^{MG} . Так как z^A — лучшее из решений, рассмотренных алгоритмом $G^{3/4}$, то $z^A \geq p_{i_*} + p_{k_*} + z_s^{MG}$. По теореме 2.4 имеем $z_s^{MG} \geq \frac{1}{2} z_s^*$.

Рассмотрим два случая.

Случай 1. $p_{i_*} + p_{k_*} \geq \frac{1}{2}z^*$. Тогда $z^A \geq p_{i_*} + p_{k_*} + z_s^{MG} \geq p_{i_*} + p_{k_*} + \frac{1}{2}z_s^* =$
 $= p_{i_*} + p_{k_*} + \frac{1}{2}(z^* - p_{i_*} - p_{k_*}) = \frac{1}{2}(z^* + p_{i_*} + p_{k_*}) \geq \frac{3}{4}z^*$.

Случай 2. $p_{i_*} + p_{k_*} < \frac{1}{2}z^*$. Тогда $\min(p_{i_*}, p_{k_*}) < \frac{1}{4}z^*$. По определению z_s^* содержит предметы с $p_j \leq \frac{1}{4}z^*$, значит $z_s^* \leq z_s^{LP} \leq z_s^{MG} + \frac{1}{4}z^*$.

Теперь $z^* = p_{i_*} + p_{k_*} + z_s^* \leq p_{i_*} + p_{k_*} + z_s^{MG} + \frac{1}{4}z^* \leq z^A + \frac{1}{4}z^*$. ■

Пример. Положим $n = 5, C = 4M$,

$$p_1 = 2, \quad p_2 = p_3 = p_4 = p_5 = M,$$

$$w_1 = 1, \quad w_2 = w_3 = w_4 = w_5 = M.$$

Очевидно, что $z^* = 4M$, $z^A = 3M + 2$, то есть оценку $K = \frac{3}{4}$ нельзя улучшить.

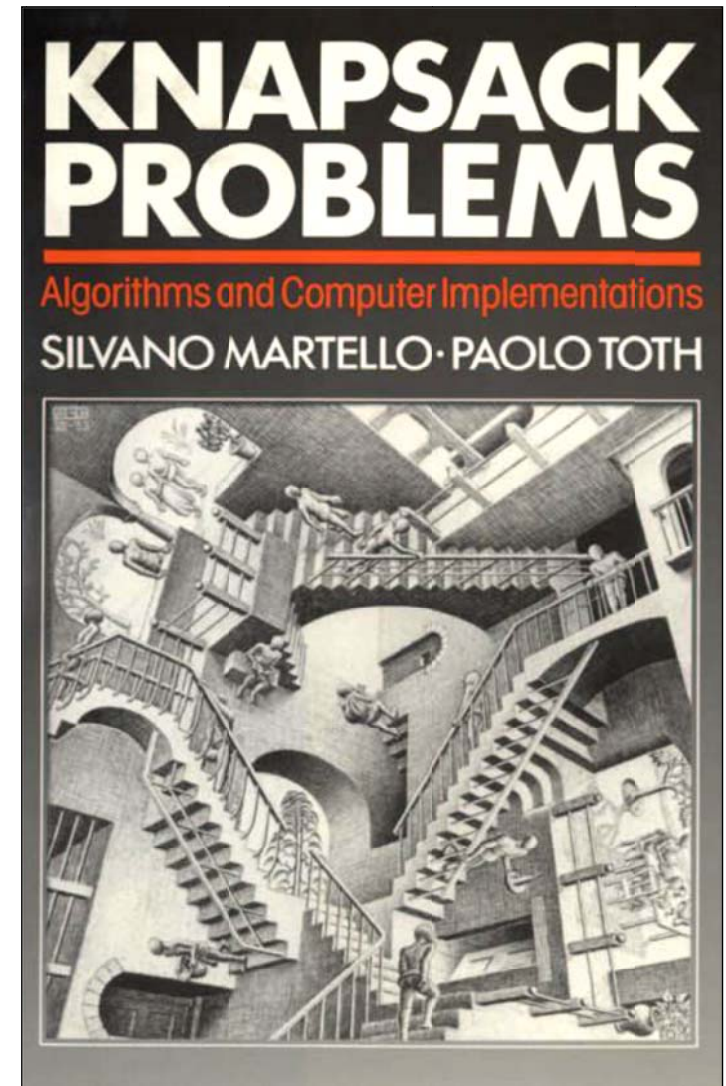
Silvano Martello, Paolo Toth
Knapsack Problem
Algorithms and Computer Implementations

University of Bologna

John Wiley & Sons 1990

296 p

http://www.math.nsc.ru/LBRT/k5/knapsack_problem.pdf



Вопросы

- Алгоритм $G^{3/4}$ для задачи о рюкзаке является полиномиальным (*Да или Нет?*)
- Алгоритм $G^{3/4}$ требует $O(n)$ затрат памяти (*Да или Нет?*)
- Алгоритм $G^{3/4}$ становится точным, если вместо алгоритма A^{MG} использовать на шаге 2.1 точный алгоритм для оставшегося места в рюкзаке и подмножества предметов $\{j \mid p_j \leq \min\{p_i, p_k\}\} \setminus \{i, k\}$ (*Да или Нет?*)
- Алгоритм $G^{3/4}$ становится точным, если вместо алгоритма A^{MG} использовать на шаге 2.1 точный алгоритм для оставшегося места в рюкзаке и подмножества предметов без пары $\{i, k\}$ (*Да или Нет?*)
- Можно ли за полиномиальное время получить точность 0.9 (*Да или Нет?*)