Lecture 9
Knapsack Problem

MATH3220 Operations Research and Logistics
Mar. 4, 2015

Pan Li
The Chinese University of Hong Kong

Agenda

1. **Knapsack Problem**
2. **Dynamic Programming formulation**
3. **2D Knapsack Problem**

Knapsack problem

Consider a hiker who is going to carry a knapsack with him on his trip. Items to be put in the knapsack can be chosen among many items, each of which has a weight and a value to him. Certainly, he would like to carry with him the maximum amount of value with total weight less than a prescribed amount.

Let there be n types of items and let

$$\begin{aligned}
w_j &= \text{weight of each item of type } j, \\
v_j &= \text{value of each item of type } j, \\
x_j &= \text{number of items of type } j \text{ that the hiker carries with him}, \\
b &= \text{total weight limitation.}
\end{aligned}$$

Then the problem becomes

$$\begin{aligned}
\text{max} \quad & \sum_{j=1}^{n} v_j x_j \\
\text{subject to} \quad & \sum_{j=1}^{n} w_j x_j \leq b \\
& x_j \geq 0, \ \text{integers.}
\end{aligned}$$

Example

Consider the knapsack problem with $b = 8$

item	1	2	3
v_j	4	6	5
w_j	3	8	5

$$\frac{v_1}{w_1} = \frac{4}{3}, \quad \frac{v_2}{w_2} = \frac{6}{8}, \quad \frac{v_3}{w_3} = \frac{5}{5},$$

$\Rightarrow$ The first type has the greatest value per unit of weight.
$\Rightarrow$ It seems natural to attempt to load as many type-1 items as possible into the knapsack.
Since the capacity of the knapsack is 8, such an attempt will then result in the loading combination $x_1 = 2$, $x_2 = x_3 = 0$, which achieves a total value of 8.
Is this loading combination optimal?

Dynamic programming formulation

(i) OPTIMAL VALUE FUNCTION:

$S(k, y)$ = maximum value obtained by using only the items of types $1, 2, \cdots, k$, when the total weight limitation is y, for $1 \leq k \leq n$ and $0 \leq y \leq b$.

(ii) RECURRENCE RELATION:

Relation I:

$$S(k, y) = \max_{j=0,1,\ldots,\lfloor y/w_k \rfloor} \{v_k \times j + S(k-1, y - w_k \times j)\},$$

Relation II:

$$S(k, y) = \max\{S(k-1, y); S(k, y - w_k) + v_k\}.$$

Dynamic Programming formulation

(iii) OPTIMAL POLICY FUNCTION:

$P(k, y) =$ the maximum index of the types of items used in $S(k, y)$, i.e. if $P(k, y) = j$, then $x_j \geq 1$, or items of type j are used in $S(k, y)$ and $x_q = 0$ for all $q > j$.

The values of $P(k, y)$ can be determined as follows:

$$\begin{aligned} P(1, y) &= 0 \quad \text{if} \quad S(1, y) = 0; \\ P(1, y) &= 1 \quad \text{if} \quad S(1, y) \neq 0; \end{aligned}$$

and

$$P(k, y) = \begin{cases} P(k-1, y) & \text{if } S(k-1, y) > S(k, y - w_k) + v_k \\ k & \text{if } S(k-1, y) \leq S(k, y - w_k) + v_k. \end{cases}$$

Dynamic Programming formulation

(iv) BOUNDARY CONDITIONS:

$$\begin{aligned} S(0, y) &= 0 \text{ for all } y \ (0 \leq y \leq b); \\ S(k, 0) &= 0 \text{ for all } k \ (0 \leq k \leq n), \text{ and} \end{aligned}$$

(v) ANSWER SOUGHT: $S(n, b)$ will be the maximum value.

Example

Consider the knapsack problem with $b = 10$, $v_1 = 1$, $v_2 = 3$, $v_3 = 5$, $v_4 = 9$, $w_1 = 2$, $w_2 = 3$, $w_3 = 4$ and $w_4 = 7$.

We summarize the DP computation in the following table.

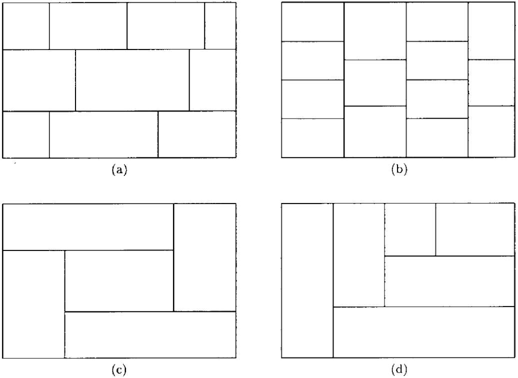
	y										
k	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1_1	1_1	2_1	2_1	3_1	3_1	4_1	4_1	5_1
2	0	0	1_1	3_2	3_2	4_2	6_2	6_2	7_2	9_2	9_2
3	0	0	1_1	3_2	5_3	5_3	$6_{2,3}$	8_3	10_3	10_3	11_3
4	0	0	1_1	$\boxed{3_2}$	5_3	5_3	$6_{2,3}$	9_4	10_3	$10_{3,4}$	$\boxed{12_4}$

Computational efficiency

The computational requirements of this approach are nb additions and comparisons.

Two Dimensional Knapsack Problem

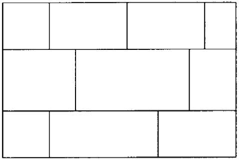
- Assume that we are given a large rectangular board and that we would like to cut the large board into small rectangles that will be sold at the market, how should we cut the large board so that we get the maximum profit?
- The problem is called a *stock-cutting problem* which has wide applications in various industries such as garment, steel, lumber, transportation, etc.

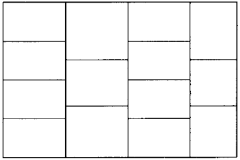


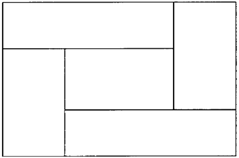
We now consider only a more restricted way of cutting. Two-stage cuttings:

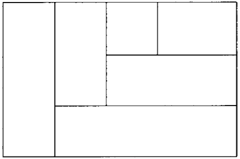
(i) by horizontal lines all the way across the board followed by vertical cuttings on each of the horizontal strips as shown in Figure (a).

(ii) by vertical lines all the way across the board followed by horizontal cuttings on each of the vertical strips as shown in Figure (b).


(a)


(b)


(c)


(d)

- Suppose that we are given the values v_i of n types of rectangles, where ℓ_i is the (horizontal) length of a type i rectangle and w_i the (vertical) width.
- We shall cut the large board in two stages such that the total value of the resulting rectangles is a maximum.
- Further assume that the rectangles can not be rotated.
- If a resulting rectangle is not exactly $\ell_i \times w_i$ for any i, we shall assume that the rectangle has value equal to the maximum of the values of all rectangles that can fit inside it. More specifically, let us consider the following numerical example.

Example

Let the large board be of size $L = 14,\ W = 11$ and the small rectangles be

$$\begin{aligned} v_1 &= \$6,\ \ell_1 = 7,\ w_1 = 2 \\ v_2 &= \$7,\ \ell_2 = 5,\ w_2 = 3 \\ v_3 &= \$9,\ \ell_3 = 4,\ w_3 = 5. \end{aligned}$$

One way of cutting by horizontal lines and then vertical lines is shown in Figure (a) with a total value of \$63, while another way of cutting by vertical lines and then horizontal lines is shown in Figure (b) with a total value of \$64.

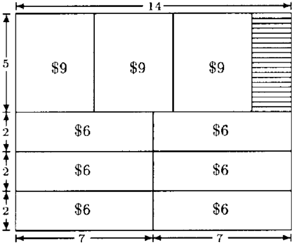


Figure (a)

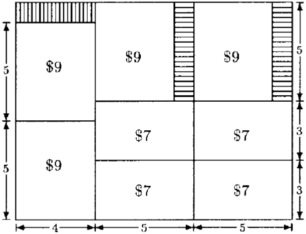


Figure (b)

Example

Let the large board be of size $L = 14$, $W = 11$ and the small rectangles be

$$\begin{aligned} v_1 &= \$6,\ \ell_1 = 7,\ w_1 = 2 \\ v_2 &= \$7,\ \ell_2 = 5,\ w_2 = 3 \\ v_3 &= \$9,\ \ell_3 = 4,\ w_3 = 5. \end{aligned}$$

To get the optimum pattern of cutting horizontally and then vertically, we consider the problem of cutting vertically a board of length 14 and width w_k. If $F_k(x)$ denotes the optimum value obtained when the width is w_k and the length is x (note that the rectangles are so ordered that the indices are increasing with the widths of the rectangles), then clearly

$$\begin{aligned} F_k(x) \quad & = \max \textstyle\sum_{j=1}^{k} v_j x_j \\ \text{subject to} \quad & \textstyle\sum_{i=1}^{k} \ell_i x_i \le x, \\ & x_i \ge 0 \quad \text{integers.} \end{aligned}$$

This problem can be solved by the knapsack algorithm of the previous section. In fact, we can show that $F_1(14) = 12$, $F_2(14) = 14$, and $F_3(14) = 27$).

Example

Let the large board be of size $L = 14$, $W = 11$ and the small rectangles be

$$\begin{aligned} v_1 &= \$6,\ \ell_1 = 7,\ w_1 = 2 \\ v_2 &= \$7,\ \ell_2 = 5,\ w_2 = 3 \\ v_3 &= \$9,\ \ell_3 = 4,\ w_3 = 5. \end{aligned}$$

The next question is "How many of these strips worth \$12, \$14, and \$27 should be produced?" This is again a one-dimensional knapsack problem, namely:

$$\begin{array}{rl} \max & 12y_1 + 14y_2 + 27y_3 \\ \text{subject to} & 2y_1 + 3y_2 + 5y_3 \leq 11 \\ & y_i \geq 0 \quad \text{integers.} \end{array}$$

Show that the optimum value is \$63.

Exercise

Consider the optimum pattern of cutting vertically and then horizontally. It can be obtained in a similar manner. Note that we have to reorder the rectangles so that the indices are increasing with the lengths of the rectangles.