Examples of Solving Knapsack Problem Using Dynamic Programming

AdvOL @McMaster, http://optlab.mcmaster.ca

February 22, 2009.

1. Consider the following knapsack problem:

$$\begin{array}{lll} \max & x_1 + 4x_2 + 3x_3 & \\ & x_1 + 3x_2 + 2x_3 & \leq 4 \end{array}$$

Solve the problem for $x_i \in \{0, 1\}$ using dynamic programming.

Solution. Let $V = [1, 4, 3]$ and $W = [1, 3, 2]$ be the array of weights and values of the 3 items respectively. Make a table representing a 2-dimensional array A of size 3×4. Element $A[i, j]$ ($i = 1, \ldots, 3, j = 1, \ldots, 4$) stores the maximal value of items from the set {item 1, item 2, ..., item i} that can be put into a knapsack with capacity j. $A[1, i]$ for all i can be easily filled in. The remaining elements in the table can be calculated in the following way:

$$A[i, j] = \begin{cases} A[i-1, j] & \text{if } W[i] > j, \\ \max\{A[i-1, j],\ V[i] + A[i-1, j - W[i]]\} & \text{otherwise.} \end{cases}$$

The table is shown below:

	$j = 1$	2	3	4
$i = 1$	1	1	1	1
2	1	1	4	5
3	1	3	4	5

The final solution is stored in $A[3, 4]$, i.e., the maximum value obtained is 5 (by choosing item 1 and 2).

2. Consider the following knapsack problem:

$$\begin{array}{lll} \max & 0.5x_1 + 4x_2 + 3x_3 & \\ & x_1 + 3x_2 + 2x_3 & \leq 5 \end{array}$$

Solve the problem for $x_i \in Z_+$ (non-negative integers: 0, 1, 2, 3,...) using dynamic programming.

Solution. The solution is very similar to the previous one except the way the elements of A are updated:

$$A[i, j] = \begin{cases} A[i-1, j] & \text{if } W[i] > j, \\ \max\{A[i-1, j],\ kV[i] + A[i-1, j - kW[i]]\} & \\ \quad \text{where } k = 1, \ldots, \lfloor \frac{j}{W[i]} \rfloor & \text{otherwise.} \end{cases}$$

The table is shown below:

	$j=1$	2	3	4	5
$i=1$	0.5	1	1.5	2	2.5
2	0.5	1	4	4.5	5
3	0.5	3	4	6	7

So the maximum value obtained is 7 (by choosing one item 2 and one item 3).