A Knapsack problem is defined as follows: Given A, a set of n elements, and a constant c, where each element j has a weight w_j and a value v_j, find a subset of the elements, denoted by B, such that $\sum_{j\in B} w_j \le c$ and $\sum_{j\in B} v_j$ is maximized.

Assume all w_j, v_j are positive integers, and $w_1 \le w_2 \le \ldots \le w_n$. Let $\sum_{j\in A} w_j = W$ and $\sum_{j\in A} v_j = V$.

1) Develop a dynamic programming of which the time complexity is $O(nW)$ to solve the knapsack problem.

Solution
Let $D(j, w)$ represent the maximum value using only the first j items and not exceeding the capacity w.

Recurrence relation: $D(j, w) = \max\{D(j-1, w-w_j) + v_j, D(j-1, w)\}$.

Boundary conditions: $D(0, w) = 0$ and $D(j, 0) = 0$ for all feasible w, j.

Our goal is to obtain $D(n, c)$.

2) Develop a dynamic programming of which the time complexity is $O(nV)$ to solve the knapsack problem.

Solution
Let $T(j, v)$ represent the minimum size required to obtain at least value v using a subset of the items $\{1, \ldots, j\}$.

Recurrence relation: $T(j, v) = \min\{T(j-1, v-v_j) + w_j, T(j-1, v)\}$.

Boundary conditions: $T(0, v) = \infty, v > 0$ and $T(j, 0) = 0, j \ge 0$.

Our goal is to obtain $\max\{v : T(n, v) \le c\}$.

3) There is a requirement of B such that for any three consecutive elements, k, $k+1$, and $k+2$, at most one of them is in B.

a. Formulate this problem by integer linear programming.

b. Develop a dynamic programming to solve this problem.

Solution
a. Let x_i be the binary variable being 1 if element i is selected, 0 otherwise.

$$\begin{aligned} \max \quad & \sum_{j\in A} v_j x_j \\ s.t. \quad & \sum_{j\in A} w_j x_j \le c \\ & x_j + x_{j+1} + x_{j+2} \le 1, j = 1, \ldots, n-2 \\ & x_j \in \{0, 1\}, j = 1, \ldots, n \end{aligned}$$

b. Method 1: Let $D(j, w)$ represent the maximum value using only the first j items and not exceeding the capacity w.

Recurrence relation: $D(j, w) = \max\{D(j-3, w-w_j) + v_j, D(j-1, w)\}$.

Boundary conditions: $D(-2, w) = D(-1, w) = D(0, w) = 0, w \geq 0$ and $D(j, 0) = 0, j \geq 0$.

Our goal is to obtain $D(n, c)$.

Method 2: The new constraint requires that the index of the adjacent selected elements should be greater than or equal to 3. Let $f(k, w)$ be the maximum value from element k to n given that element k is selected and the capacity is w.

Recurrence relation: $f(k, w) = \max_{k'}\{f(k', w-w_k)+v_k | k' = k+3, \ldots, n\}, k = 1, \ldots, n-3$.

Initial condition: $f(n+1, w) = f(n+2, w) = f(n+3, w) = 0$.

The optimal is given by $\max\{f(i, c) | i = 1, \ldots, n\}$.

4) There is a requirement of B such that if element k is in B, at least one of $\{k-1, k+1\}$ is in B.

a. Formulate this problem by integer linear programming.

b. Develop a dynamic programming to solve this problem.

Solution

a. Let x_i be the binary variable being 1 if element i is selected, 0 otherwise. Set $x_0 = x_{n+1} = 0$.

$$\begin{aligned} \max \quad & \sum_{j \in A} v_j x_j \\ s.t. \quad & \sum_{j \in A} w_j x_j \leq c \\ & x_{j-1} + x_{j+1} \geq x_j, j = 1, \ldots, n \\ & x_j \in \{0, 1\}, j = 1, \ldots, n \end{aligned}$$

b. Method 1: The new constraint demonstrates that the selected elements are several consecutive segments. Let $v_{ij} = v_i + \ldots + v_j$ and $w_{ij} = w_i + \ldots + w_j$ for the segment $[i, j], i < j$. Let $f(i, w)$ be the maximum value from element i to n given that element i is the first selected in some segment and the capacity is w.

Recurrence relation:

$$f(i,w) = \max\{f(k, w - w_{ij}) + v_{ij} | j > i, k > j+1\}.$$

Initial conditions:

$$f(n+1, w) = f(n, w) = 0$$

$$f(n-1, w) = \begin{cases} v_{n-1} + v_n, & \text{if } w \geq w_{n-1} + w_n \\ 0, & \text{if } w < w_{n-1} + w_n \end{cases}$$

The optimal is given by $\max\{f(i,c) | i = 1, \ldots, n-1\}$.

Method 2: Let $D(j, w, d_{j-1}, d_j)$ represent the maximum value using only the first j items and not exceeding the capacity w under the condition that the status of elements j and $j-1$ is d_j, d_{j-1}. If the status of element is 1, this element is selected, if the status of element is 0, this element is not selected.

Notice that when $d_j = 1$, $d_{j+1} = 0$, d_{j-1} can only be 1.

Recurrence relation:

$D(j, w, 0, 0) = \max\{D(j-1, w, 0, 0), D(j-1, w, 1, 0)\}$

$D(j, w, 0, 1) = \max\{D(j-1, w - w_j, 0, 0) + v_j, D(j-1, w - w_j, 1, 0) + v_j\}$

$D(j, w, 1, 0) = D(j-1, w, 1, 1)$

$D(j, w, 1, 1) = \max\{D(j-1, w - w_j, 0, 1) + v_j, D(j-1, w - w_j, 1, 1) + v_j\}$

Boundary conditions: $D(j, 0, d_{j-1}, d_j) = 0, j > 1$, $D(1, w, 0, 0) = 0, w \geq 0$, $D(1, w, 1, 0) = D(1, w, 1, 1) = -\infty, w \geq 0$, and

$$D(1, w, 0, 1) = \begin{cases} v_1, & \text{if } w \geq w_1 \\ -\infty, & \text{if } w < w_1 \end{cases}$$

Our goal is to obtain $\max\{D(n, c, 1, 1), D(n, c, 1, 0), D(n, c, 0, 0)\}$.