Example – Multiobjective Binary Knapsack Problem

Same problem as last time, but with the caveat that we will have multiple objectives and solve it by the DP algorithm. First, let us consider the single objective variant:

N : number of items,

w_k : weight of the item k (assumed integer),

c_k : value of the item k,

x_k : remaining capacity before deciding on taking item k,

u_k : decision about item k (take or not, $u_k \in \{0, 1\}$),

W : capacity of the bag (knapsack).

The idea of the DP formulation is the following – we go through every item sequentially (the order does not matter), the stages will be equivalent to the choices about different items. The state x_k will have every possible remaining volume of the knapsack ($x_k \in \{0, 1, \ldots, W\}$). The system equation:

$$x_{k+1} = x_k - w_k u_k,$$

the criteria functions:

$$g_k(x_k, u_k) = -c_k u_k, \quad g_N(x_N) = 0,$$

with the condition that u_k can be always 0, but can be 1 only if there is enough space left in the bag. The cost-to-go functions then have the following format:

$$\begin{aligned} J_N(x_N) &= g_N(x_N), \\ J_k(x_k) &= \min_{u_k \in U_k} \{g_k(x_k, u_k) + J_{k+1}(x_k - w_k u_k)\}. \end{aligned}$$

For the multiobjective formulation, we are going with the following scenario. Imagine we are a thief, trying to steal some of the N items. The first objective is the same as in the single objective case (maximize profit), the second objective will consider the probability that we get caught (which we want to minimize). For the each item, let us denote the probability that we will not get caught if we take it as p_k. Moreover, depending on how heavy our bag is, we might get spotted, searched and caught (this will be reflected in the criterion for the final state). We have:

$$g_k^1(x_k, u_k) = -c_k u_k, \quad g_N^1(x_N) = 0,$$

$$g_k^2(x_k, u_k) = \begin{cases} 1 \text{ if } u_k = 0 \\ p_k \text{ otherwise} \end{cases} = u_k p_k + (1 - u_k), \quad g_N^2(x_N) = p_N(x_N).$$

The probability of not getting caught (assuming independence) is

$$g_N^2(x_N) \prod_{k=0}^{N-1} g_k^2(x_k, u_k),$$

which can be approached by using log (all values are positive):

$$\ln\left(g_N^2(x_N) \prod_{k=0}^{N-1} g_k^2(x_k, u_k)\right) = \ln\left(g_N^2(x_N)\right) + \sum_{k=0}^{N-1} \ln\left(g_k^2(x_k, u_k)\right),$$

which is in the format we can work with. Since we want to maximize this quantity (probability of not getting caught), we transform it to the minimization form:

$$-\ln\left(g_N^2(x_N)\right) - \sum_{k=0}^{N-1} \ln\left(g_k^2(x_k, u_k)\right).$$

Assignment: For the following instance of the problem, find the set of noninferior solutions (the pareto front):

- $N = 10$
- $W = 12$
- $w = [2; 3; 7; 1; 5; 4; 3; 5; 6; 2]$
- $c = [5; 6; 11; 3; 7; 7; 6; 7; 8; 5]$
- $p = [0.95; 0.93; 0.7; 0.96; 0.94; 0.96; 0.95; 0.91; 0.9; 0.95]$
- $g_N^2(x_N) = 1 - 0.15\frac{W - x_N}{W}$