Algorithm 4.1 DYNAMIC PROGRAMMING KNAPSACK

Input. Integers W, C, vectors $w, c \in \mathbb{N}^n$.

Output. Vector $x \in \{0,1\}^n$ such that weight$(x) \leq W$.

Step 1. Set $m_{0,0} = 0$, $m_{0,k} = \infty$ for $k = 1, \ldots, C$, and $x(0,0) = 0$.

Step 2. For $j = 1, \ldots n$ and $k = 0, \ldots, C$ do the following:

If $c_j \leq k$ and $m_{j-1,k-c_j} + w_j \leq \min\{W, m_{j-1,k}\}$ then set

$$m_{j,k} = m_{j-1,k-c_j} + w_j$$

and set $x(j,k)_i = x(j-1, k-c_j)_i$ for $i \neq j$ and $x(j,k)_j = 1$.

Otherwise set

$$m_{j,k} = m_{j-1,k}$$

and $x(j,k) = x(j-1,k)$.

Step 3. Determine the largest $k \in \{0, \ldots, C\}$ such that $m_{n,k} < \infty$. Return $x(n,k)$.