Algorithm 3.1 GREEDY

Input. Integer c, vectors $p, w \in \mathbb{N}^n$ with $w_j \leq c$, $\sum_j w_j > c$, and $p_1/w_1 \geq \cdots \geq p_n/w_n$.

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

(1) Define $k = \min\{j \in \{1,\ldots,n\} : \sum_{i=1}^{j} w_i > c\}$.

(2) Let $x = (1^{k-1}, 0^{n-k+1})$ and $y = (0^{k-1}, 1, 0^{n-k})$.

(3) Return x if $\text{val}(x) \geq \text{val}(y)$; otherwise y.

Algorithm 3.2 Nemhauser-Ullmann

Input. Integer c, vectors $p, w \in \mathbb{N}^n$ with $w_j \leq c$, and $\sum_j w_j > c$.

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

(1) Let $S_0 = \{0\}$.

(2) For $j = 1, \ldots, n$ let $T_j = \{x \oplus 1_j : x \in S_{j-1}\}$ and

$$S_j = S_{j-1} \cup T_j - \{y \in S_{j-1} \cup T_j : \; y \text{ is dominated by some } x \in S_{j-1} \cup T_j\}.$$

(3) Return the most profitable solution $x^* \in S_0 \cup \cdots \cup S_n$ with weight$(x^*) \leq c$.

Algorithm 3.3 RGREEDY

Input. Integer r, vectors v, μ, real $p \in [0, 1]$

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

(1) Choose index k with probability v_k.

(2) If $k < r$ insert item k, i.e., $x_k = 1$. If $k = r$, flip another independent coin and insert item r, i.e., $x_r = 1$ with probability p, otherwise discard it.

(3) Then insert the items $1, 2, \ldots, k-1, k+1, \ldots, r$ in the greedy order, i.e., for $j = 1, 2, \ldots, k-1, k+1, \ldots, r$ set $x_j = 1$ if the item j still fits.