10

Multiobjective Versions of Some $\mathcal{NP}$-Hard Problems

10.1 The Knapsack Problem and Branch and Bound

As for the assignment problem in the previous section, we consider only finding efficient solutions. And we also restrict ourselves to the bicriterion case. The bicriterion knapsack problem is the binary integer program

$$\max f_1(x) = \sum_{i=1}^{n} c_i^1 x_i \tag{10.1}$$

$$\max f_2(x) = \sum_{i=1}^{n} c_i^2 x_i \tag{10.2}$$

$$\text{subject to } \sum w_i x_i \leq W \tag{10.3}$$

$$x_i \in \{0,1\}; \quad j = 1, \ldots, n. \tag{10.4}$$

The problem is obviously $\mathcal{NP}$-hard, as a counterpart of an $\mathcal{NP}$-hard single objective problem (see Lemma 8.11). Whether the problem is $\#\mathcal{P}$-complete or intractable is yet unknown.

We will present a branch and bound algorithm. To avoid trivial solutions and to have a meaningful problem we make some basic assumptions on the parameters of the knapsack problem. We assume that all values c_i^k, all weights w_i as well as the capacity W are nonnegative. Furthermore, no single weight exceeds capacity, i.e. $w_i \leq W$ for all $i = 1, \ldots, n$, but the total weight of all items is bigger than W, $\sum_{i=1}^{n} w_i > W$.

For the solution of knapsack problems the value to weight ratios c_i^k / w_i are of essential importance. In the single objective linear knapsack problem (where $x_i \in \{0,1\}$ is replaced by $0 \leq x_i \leq 1$),

$$\begin{aligned} \max \sum_{i=1}^{n} c_i x_i & \\ \text{subject to } \sum_{i=1}^{n} w_i x_i \leq W & \\ x_i \leq 1 \quad & i = 1, \ldots, n \\ x_i \geq 0 \quad & i = 1, \ldots, n \end{aligned}$$

they are used to easily find an optimal solution.

Assume that items $1, \ldots, n$ are ordered such that

$$\frac{c_1}{w_1} \geq \frac{c_2}{w_2} \geq \ldots \geq \frac{c_n}{w_n}. \tag{10.5}$$

Let $i^* := \min\{i : \sum_{j=1}^{i} w_j > W\}$ be the smallest index such that the weight of items 1 to i exceeds the total capacity. Item i^* is called the *critical item.* The solution of the continuous knapsack problem is simply given by taking all items 1 to $i^* - 1$ and a fraction of the critical item, that is $x_i = 1$ for $i = 1, \ldots, i^* - 1$ and

$$x_{i^*} = \frac{\left(W - \sum_{i=1}^{i^*-1} w_i \right)}{w_{j^*}}.$$

Good algorithms for the single objective problem use this fact and focus on optimization of items around i^*, see e.g. Martello and Toth (1990); Pisinger (1997); Kellerer *et al.* (2004). Ideas of such algorithms have been adapted to the bicriterion case by Ulungu and Teghem (1997).

The two criteria induce two different sequences of value to weight ratios. Let $\mathcal{O}_k$ be the ordering (10.5) according to c_i^k / w_i, $k = 1, 2$. Let r_i^k be the rank or position of item i in order $\mathcal{O}_k$ and let $\mathcal{O}$ be the order according to increasing values of $(r_i^1 + r_i^2)/2$, the average rank of an item.

The branch and bound method will create partial solutions by assigning zeros and ones to subsets of variables denoted $\mathcal{B}_0$ and $\mathcal{B}_1$, respectively. These *partial solutions* constitute nodes of the search tree. Variables not assigned either zero or one are called *free variables* for a partial solution and define a set $\mathcal{F} \subseteq \{1, \ldots, n\}$ such that $\{1, \ldots, n\} = \mathcal{B}_1 \cup \mathcal{B}_0 \cup \mathcal{F}$. A solution formed by assigning all free variables a value is called *completion* of a partial solution. Variables of a partial solution will be assigned a value according to the order $\mathcal{O}$. It is convenient to number the items in that order so that we will have

$$\mathcal{B}_1 \cup \mathcal{B}_0 = \{1, \ldots, l - 1\}, \mathcal{F} = \{l, \ldots, n\}$$

for some l. Furthermore we shall denote r_k the index of the first variable in $\mathcal{F}$ according to order $\mathcal{O}_k$, for $k = 1, 2$.