Dynamic Programming Approaches to the Multiple Criteria Knapsack Problem

Kathrin Klamroth* Margaret M. Wiecek

Clemson University
Clemson, SC
USA

Abstract

In this paper we study the integer multiple criteria knapsack problem and propose dynamic-programming-based approaches to finding all the nondominated solutions.

Different and more complex models are discussed including the binary multiple criteria knapsack problem, problems with more than one constraint, and multiperiod as well as time-dependent models.

1 Introduction

The single criterion knapsack problem is a well known combinatorial optimization problem with a wide range of applications (for an overview see e.g. [20, 21]).

Since in many real world applications the preferences of multiple decision makers have to be incorporated into the model, it is a natural extension of the classical knapsack model to consider more than one criterion. Examples

*On leave from the Department of Mathematics, University of Kaiserslautern, Kaiserslautern, Germany.

This work was partially supported by ONR Grant N00014-97-1-0784

may be found in affordability analysis where projects have to be chosen with respect to more then a single criterion or in Capital Budgeting (see e.g., [2, 28, 19]). Multiple criteria knapsack problems were used by Kostreva et al. [18] to deal with relocation issues arising in conservation biology. Teng and Tzeng [24] applied the multiple criteria multiple constraint knapsack problem to transportation investment planning.

In this paper we consider the integer multiple criteria knapsack problem (MCKP) formulated as

$$\begin{array}{ll} \text{vmax} & f(x) = Cx \\ \text{s.t.} & ax \leq b \\ & x_j \geq 0, \text{ integer}, \quad j = 1, \ldots, n \end{array} \tag{1}$$

where C is an $m \times n$ matrix with nonnegative entries c_j^i, $i = 1, \ldots, m$, $j = 1, \ldots, n$. We denote the ith row of C by c^i and the jth column of C by c_j. Thus $f_i(x) = c^i x$, $i = 1, \ldots, m$, represent the m conflicting objective functions. The constraint $ax \leq b$ is interpreted as a *capacity constraint* (*budget constraint*). The set of feasible solutions of (1) is given by $X = \{x \in \mathbb{N}_0^n \, : \, ax \leq b\}$.

Throughout the paper we additionally assume that the weight coefficients a_j, $j = 1, \ldots, n$ and the right-hand-side of the capacity constraint b are positive integers. In order to avoid trivial solutions let $0 < a_j \leq b$, $j = 1, \ldots, n$ and $\sum_{j=1}^n a_j > b$. We also find it convenient to consider the MCKP with the right-hand-side $k = 0, 1, \ldots, b$ of the capacity constraint and denote this problem by k-MCKP.

A special case of the above formulation is the case that $m = 2$, i.e. the bicriteria case. In this case problem (1) becomes

$$\begin{array}{ll} \text{vmax} & f(x) = [c^1 x \, , \, c^2 x] \\ \text{s.t.} & ax \leq b \\ & x_j \geq 0, \text{ integer}, \quad j = 1, \ldots, n. \end{array} \tag{2}$$

Solving (1) is understood as generating its efficient (Pareto) solutions. A feasible solution $\hat{x} \in X$ is said to be an efficient solution of (1) if there is no

other feasible solution $x \in X$ such that $f(x) \geq f(\hat{x})$, i.e.:

$$\begin{array}{lll} & \forall i \in \{1,\ldots,m\} & f^i(x) \geq f^i(\hat{x}) \\ \text{and} & \exists i \in \{1,\ldots,m\} \quad s.t. & f^i(x) > f^i(\hat{x}). \end{array} \tag{3}$$

Let $\mathcal{X}_e$ denote the set of efficient solutions of (1) and let $\mathcal{Y}_e$ denote the image of $\mathcal{X}_e$ in the objective space, that is $\mathcal{Y}_e = f(\mathcal{X}_e)$, where $f = [f_1, \ldots, f_m]$. $\mathcal{Y}_e$ is referred to as the set of nondominated solutions of (1).

Despite the wide range of applications, the literature on the MCKP is very limited. Many authors focus on the determination of the set of supported nondominated solutions, i.e., those solutions whose pre-images are optimal solutions of the weighted-sum of the objective functions

$$\begin{array}{ll} \max & \sum_{i=1}^{m} \lambda_i c^i x \\ \text{s.t.} & ax \leq b \\ & x_j \geq 0, \text{ integer}, \; j = 1,\ldots,n \end{array} \tag{4}$$

for some $\lambda_i \geq 0, i = 1,\ldots,m$ and $\sum_{i=1}^{m} \lambda_i = 1$.

In [22], Rosenblatt and Sinuany-Stern suggested a branch and bound algorithm to determine the set of all supported nondominated solutions of problem (1) with binary variables. Additionally they presented a heuristic procedure that finds an approximation of the supported nondominated solutions using significantly less time than the exact procedure. Eben-Chaime [9] continued the work of Rosenblatt and Sinuany-Stern and proposed a network-based approach to solve the individual parameterized binary single-criterion knapsack problems.

In several recent papers, Ulungu and Teghem [26, 27] and Visée, Teghem, Pirlot and Ulungu [31] pointed out the importance of considering also non-supported nondominated solutions. In [31] they presented a table showing that for a large set of randomly generated examples with 10 to 500 variables, the supported nondominated solutions make up only a small percentage of all nondominated solutions. By considering only the supported nondominated solutions a decision maker will be restricted to a small subset of the nondominated solutions and may be forced to make an unfavorable decision.

Therefore a two phase method was suggested in [26] that in the first phase constructs all supported nondominated solutions and in the second phase determines the nonsupported nondominated solutions applying a branch-and-bound-based algorithm. That work was continued in [27] and [31], where especially the second phase of the algorithm was improved.

As the MCKP falls into the category of multiple criteria integer programs, algorithms proposed for finding all efficient solutions of the latter could be also applied to solve the former. Bitran [3, 4] developed theory and algorithms for multiple criteria linear programs with binary variables. The algorithms were based on enumerative schemes and solving some auxiliary multiple objective programs. Multiple criteria integer linear programs were studied by several authors. Klein and Hannan [15] developed an algorithm for generating the complete efficient set of such problems. This is a sequential procedure in which one of the criterion functions is optimized subject to progressively more constrained feasible sets determined by the other criteria and previously found efficient solutions. A variation of the weighted-sum method was studied by Chalmet et al. [6] for the same class of problems. As this method can only find the supported nondominated solutions, the authors introduced an additional constraint to ensure access to the nonsupported nondominated solutions.

Villarreal and Karwan [29] were perhaps the only ones who proposed dynamic programming (DP) approaches to the integer multiple criteria multiple constraint knapsack problem. They proposed four approaches: two basic ones, an embedded state approach, and a hybrid approach. The first basic approach was very similar to Nemhauser and Garfinkel's [10] recursive equations (I) developed for the single objective single constraint knapsack problem while the second basic approach was a generalization of the recursive equations (III) of Nemhauser and Garfinkel [10] for the multiple constraint knapsack problem. The two other approaches aimed at reducing the computational complexity of the basic approaches. Villarreal and Karwan [30] extended dynamic programming recursive equations to the general multiple criteria integer framework and presented them on the binary MCKP with multiple constraints.

In this paper, we follow upon DP approaches and study them in the context of different variations of the multiple criteria knapsack model. In

contrast to the work by Villarreal and Karwan [29] who concentrated on computational issues, we argue that DP, being a very flexible technique, is a unifying umbrella under which all those variations can be solved. Therefore the purpose of the paper is to provide evidence for the claim that DP is a versatile modeling tool enabling the user to solve different variations of the knapsack problem in the same environment.

In Section 2 we present and discuss several DP formulations of the MCKP. We have chosen those formulations that turned out to be specially useful while dealing with more complex models. Our presentation is based on three recursive equations provided by Nemhauser and Garfinkel [10], and four representations collected by Ibaraki [13] for the single criterion single constraint integer knapsack problem as well as on two basic approaches proposed by Villarreal and Karwan [29] for the MCKP with multiple constraints. While adapting DP formulations to the multiple objective case, we used multiple objective dynamic programming that has been studied by many authors. Brown and Strauch [5] were probably the first to show that Bellman's principle of optimality [1] can be extended to models with multiple criteria. Results directly related to this paper were obtained by Klötzler [17] and Yu and Seiford [32]. Among others, Henig [12] developed a general theory of dynamic programming with multiple objective functions. Corley and Moon [7] and Hartley [11] showed how to compute the set of nondominated paths in a network with vector costs on links while algorithmic procedures for finding all nondominated solutions to a multi-stage discrete decision process were proposed by Trzaskalik [25].

In Section 3, several variations of the original problem are examined in the context of the DP framework. In particular, we study the binary MCKP, the MCKP with multiple constraints, a multiple period model and a time-dependent model, and discuss the applicability of the DP approaches of Section 2 to handle all these problems.

Throughout the paper we use a didactic bi-criteria example and its extensions to illustrate our findings. Section 4 concludes the paper and highlights the directions of further research.

2 Dynamic programming approaches

Throughout this section we will use the following didactic example of a bi-criteria knapsack problem (2) to illustrate the discussed DP-approaches:

$$
\begin{array}{ll}
\text{vmax} & f(x) = [x_1 + x_2 + 2x_3 + 3x_4\ ,\ 4x_1 + 7x_2 + 2x_3 + x_4] \\
\text{s.t.} & x_1 + 2x_2 + x_3 + x_4 \leq 3 \\
& x_j \geq 0,\ \text{integer},\ \ j = 1, \ldots, 4.
\end{array} \tag{5}
$$

In this example, the vector a of weights is given by $[1, 2, 1, 1]$, the capacity b is 3, and the two objective vectors c^1 and c^2 are equal to $[1, 1, 2, 3]$ and $[4, 7, 2, 1]$, respectively. The set of efficient solutions $\mathcal{X}_e$ and the set of nondominated solutions $\mathcal{Y}_e$ of this example problem are:

$$
\mathcal{X}_e = \left\{ \begin{bmatrix} 3 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 3 \end{bmatrix} \right\},
$$

$$
\mathcal{Y}_e = \left\{ \begin{bmatrix} 3 \\ 12 \end{bmatrix}, \begin{bmatrix} 4 \\ 10 \end{bmatrix}, \begin{bmatrix} 5 \\ 9 \end{bmatrix}, \begin{bmatrix} 6 \\ 7 \end{bmatrix}, \begin{bmatrix} 7 \\ 6 \end{bmatrix}, \begin{bmatrix} 8 \\ 4 \end{bmatrix}, \begin{bmatrix} 9 \\ 3 \end{bmatrix} \right\}.
$$

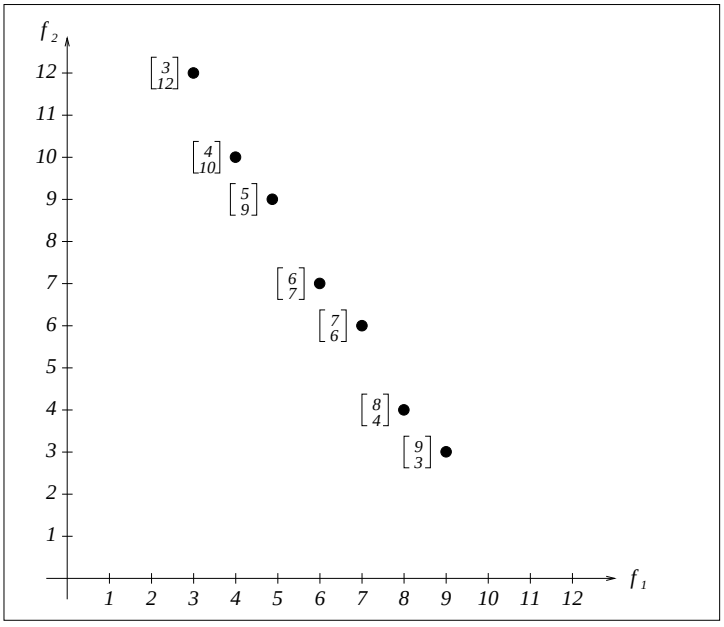

Figure 1: The nondominated solutions of problem (5) in the objective space.

In the following subsections we present five distinct DP formulations of the MCKP resulting from different definitions of states and decisions to be taken at a given state. At the beginning of each subsection we inform the reader in detail about related results available in the literature. In general, whether single or multiple criteria, the knapsack problem can be treated as a sequential decision making process whose states and transitions between them are determined by the feasibility constraint of the original problem. The multiple criteria are treated as a vector cost function defined on the states and decisions of the process. For each model, we develop recursive equations that yield the set of all nondominated solutions of the MCKP. As these results are based upon two theoretical well known foundations, we present them without proofs. First, analogous recursive equations are available for the single criterion knapsack problem. Second, general recursive equations for sequential decision making problems with multiple criteria were proved, see Yu and Seiford [32].

The sequential decision making process of the MCKP can be represented by a network whose nodes and arcs come from the states and decisions to be taken in the process. Equivalently, solving the MCKP can be viewed as finding the set of all nondominated paths from the initial state(s) to the final state(s) in the network. Therefore, specialized algorithms designed to handle the shortest path problem in networks with vector costs on links could also be used to find the nondominated solutions of the MCKP.

2.1 Model I

The following DP-approach is an adaptation of the recursive equations given for the single criterion case in Garfinkel and Nemhauser [10] (equations III), Ibaraki [13] (representation 1) and for the MCMCKP in Villarreal and Karwan [29] (approach 2).

Let a set of states Q be defined as

$$Q := \{q(0), q(1), \ldots, q(b)\}.$$

In this model the state $q(k)$, $k = 0, \ldots, b$, represents all nonnegative integer solutions of the k-MCKP, i.e.

$$q(k) := \{x \in I\!N_0^n \ : \ ax = k\}.$$

Note that for the initial state $q(0)$ we get $q(0) = \{\underline{0}\}$ independently from the weight vector a and the objective coefficients.

Since there may occur nondominated solutions in all states the set of final states Q_F is given by

$$Q_F := \{q(0), q(1), \ldots, q(b)\}.$$

The decision of increasing a variable x_j by 1 for a solution $x \in q(k)$ results in an increase of the right-hand side k by a_j and thus corresponds to a transition of x from the state $q(k)$ to the state $q(k + a_j)$.

The possible transitions between states for the example problem (5) are represented by the arcs in the network N_1 given in Figure 2. The objective value (c_j^1, c_j^2) of each transition and the corresponding variable x_j are identified for each arc and denoted by the vector $[j, (c_j^1, c_j^2)]$.

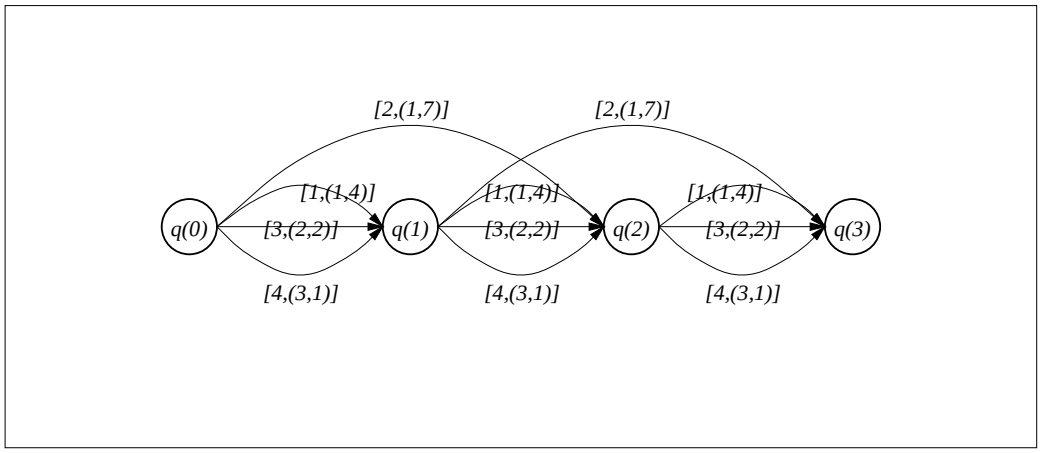

Figure 2: The vertices of network N_1 represent the states of the corresponding DP-formulation for the example problem (5).

Let $G(q(k))$, $k = 0, 1, \ldots, b$, be the set of all nondominated solutions of the k-MCKP. Then the original MCKP can be solved applying the following recursive equations:

$$\begin{aligned} G(q(0)) &= \{\underline{0}\} \\ G(q(k)) &= \text{vmax}\{G(q(k - a_j)) + c_j \,:\, j \in S,\ k - a_j \geq 0\}, \qquad k = 1, \ldots, b, \end{aligned}$$

where $S := \{1, \ldots, n\}$ denotes the index set of the variables x_j, $j = 1, \ldots, n$, and operation $vmax$ computes the nondominated solutions in the set being

the algebraic sum of the cost vector c_j and the set of all the nondominated solutions of the $(k-a_j)$-MCKP.

Since all the states are final, the set of vector costs of all nondominated solutions $\mathcal{Y}_e$ is obtained as the vector-maximum of the union of the sets $G(q(k))$, $k=1,\ldots,b$, i.e.

$$\mathcal{Y}_e = \text{vmax} \bigcup_{k=0,1,..,b} G(q(k)). \tag{6}$$

Note that in this formulation multiple nondominated solutions may occur in each step of the recursion since the variables can be selected in a different order. To avoid multiple solutions a reduction has to be applied in each stage additionally to the reduction due to dominated solutions.

In the following we illustrate the application of the recursion using the example introduced in (5).

$$\begin{aligned}
G(q(0)) &= \left\{\begin{bmatrix}0\\0\end{bmatrix}\right\} \\
G(q(1)) &= \text{vmax}\ \{G(q(0))+c_1; G(q(0))+c_3; G(q(0))+c_4\} \\
&= \text{vmax}\ \left\{\left\{\begin{bmatrix}0\\0\end{bmatrix}\right\}+\begin{bmatrix}1\\4\end{bmatrix}; \left\{\begin{bmatrix}0\\0\end{bmatrix}\right\}+\begin{bmatrix}2\\2\end{bmatrix}; \left\{\begin{bmatrix}0\\0\end{bmatrix}\right\}+\begin{bmatrix}3\\1\end{bmatrix}\right\} \\
&= \text{vmax}\ \left\{\begin{bmatrix}1\\4\end{bmatrix}; \begin{bmatrix}2\\2\end{bmatrix}; \begin{bmatrix}3\\1\end{bmatrix}\right\} \\
&= \left\{\begin{bmatrix}1\\4\end{bmatrix}, \begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}3\\1\end{bmatrix}\right\} \\
G(q(2)) &= \text{vmax}\ \{G(q(1))+c_1; G(q(0))+c_2; G(q(1))+c_3; G(q(1))+c_4\} \\
&= \text{vmax}\ \left\{\left\{\begin{bmatrix}1\\4\end{bmatrix}, \begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}3\\1\end{bmatrix}\right\}+\begin{bmatrix}1\\4\end{bmatrix}; \left\{\begin{bmatrix}0\\0\end{bmatrix}\right\}+\begin{bmatrix}1\\7\end{bmatrix}; \right. \\
&\qquad \left.\left\{\begin{bmatrix}1\\4\end{bmatrix}, \begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}3\\1\end{bmatrix}\right\}+\begin{bmatrix}2\\2\end{bmatrix}; \left\{\begin{bmatrix}1\\4\end{bmatrix}, \begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}3\\1\end{bmatrix}\right\}+\begin{bmatrix}3\\1\end{bmatrix}\right\} \\
&= \text{vmax}\ \left\{\begin{bmatrix}2\\8\end{bmatrix}, \begin{bmatrix}3\\6\end{bmatrix}, \begin{bmatrix}4\\5\end{bmatrix}; \begin{bmatrix}1\\7\end{bmatrix}; \begin{bmatrix}3\\6\end{bmatrix}, \begin{bmatrix}4\\4\end{bmatrix}, \begin{bmatrix}5\\3\end{bmatrix}; \right. \\
&\qquad \left.\begin{bmatrix}4\\5\end{bmatrix}, \begin{bmatrix}5\\3\end{bmatrix}, \begin{bmatrix}6\\2\end{bmatrix}\right\}
\end{aligned}$$

$$
\begin{aligned}
&= \left\{ \begin{bmatrix} 2 \\ 8 \end{bmatrix}, \begin{bmatrix} 3 \\ 6 \end{bmatrix}, \begin{bmatrix} 4 \\ 5 \end{bmatrix}, \begin{bmatrix} 5 \\ 3 \end{bmatrix}, \begin{bmatrix} 6 \\ 2 \end{bmatrix} \right\} \\
G(q(3)) &= \text{vmax } \{G(q(2)) + c_1; G(q(1)) + c_2; G(q(2)) + c_3; G(q(2)) + c_4\} \\
&= \cdots = \left\{ \begin{bmatrix} 3 \\ 12 \end{bmatrix}, \begin{bmatrix} 4 \\ 10 \end{bmatrix}, \begin{bmatrix} 5 \\ 9 \end{bmatrix}, \begin{bmatrix} 6 \\ 7 \end{bmatrix}, \begin{bmatrix} 7 \\ 6 \end{bmatrix}, \begin{bmatrix} 8 \\ 4 \end{bmatrix}, \begin{bmatrix} 9 \\ 3 \end{bmatrix} \right\} \\
\mathcal{Y}_e &= \text{vmax } \{G(q(0)), G(q(1)), G(q(2)), G(q(3))\} \\
&= \text{vmax } \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}; \begin{bmatrix} 1 \\ 4 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 3 \\ 1 \end{bmatrix}; \begin{bmatrix} 2 \\ 8 \end{bmatrix}, \begin{bmatrix} 3 \\ 6 \end{bmatrix}, \begin{bmatrix} 4 \\ 5 \end{bmatrix}, \begin{bmatrix} 5 \\ 3 \end{bmatrix}, \begin{bmatrix} 6 \\ 2 \end{bmatrix}; \right. \\
&\qquad \left. \begin{bmatrix} 3 \\ 12 \end{bmatrix}, \begin{bmatrix} 4 \\ 10 \end{bmatrix}, \begin{bmatrix} 5 \\ 9 \end{bmatrix}, \begin{bmatrix} 6 \\ 7 \end{bmatrix}, \begin{bmatrix} 7 \\ 6 \end{bmatrix}, \begin{bmatrix} 8 \\ 4 \end{bmatrix}, \begin{bmatrix} 9 \\ 3 \end{bmatrix} \right\} \\
&= \left\{ \begin{bmatrix} 3 \\ 12 \end{bmatrix}, \begin{bmatrix} 4 \\ 10 \end{bmatrix}, \begin{bmatrix} 5 \\ 9 \end{bmatrix}, \begin{bmatrix} 6 \\ 7 \end{bmatrix}, \begin{bmatrix} 7 \\ 6 \end{bmatrix}, \begin{bmatrix} 8 \\ 4 \end{bmatrix}, \begin{bmatrix} 9 \\ 3 \end{bmatrix} \right\}.
\end{aligned}
$$

The set of efficient solutions $\mathcal{X}_e$ can be found by keeping track of the variables that contribute to the elements of the sets $G(q(k)), k = 0, 1, \ldots, b$.

2.2 Model II

The recursive equations given in this section were developed by Ibaraki [13] (representation 2) for the single criterion case.

In Model II, a set of states Q is defined as

$$Q := \{q(k, j) \,:\, k = 1, \ldots, b,\ j = 1, \ldots, n\}$$

where the state $q(k, j)$, $k = 1, \ldots, b$, $j = 1, \ldots, n$, is defined as the set of all feasible solutions of the k-MCKP satisfying $x_j > 0$ and $x_{j+1}, \ldots, x_n = 0$, i.e.

$$q(k, j) := \{x \in I\!N_0^n \,:\, \sum_{p=1}^{j} a_p x_p = k, x_j > 0, x_{j+1}, \ldots, x_n = 0\}.$$

The initial state $q(0, 0)$ is given by $q(0, 0) = \{\underline{0}\}$.

As in the previous model, all states in this model are final, i.e. the set of final states Q_F is given by

$$Q_F := \{q(0, 0)\} \cup \{q(k, j) \,:\, k = 1, \ldots, b,\ j = 1, \ldots, n\}.$$

The decision of increasing a variable x_j by 1 in a state $q(k,p)$ with $0 \leq p \leq j$ corresponds to a transition from the state $q(k,p)$ to the state $q(k+a_j, j)$. Once a variable x_j is increased, a variable x_p with $p < j$ cannot be changed. Model II applied to the example problem (5) yields the network in Figure 3.

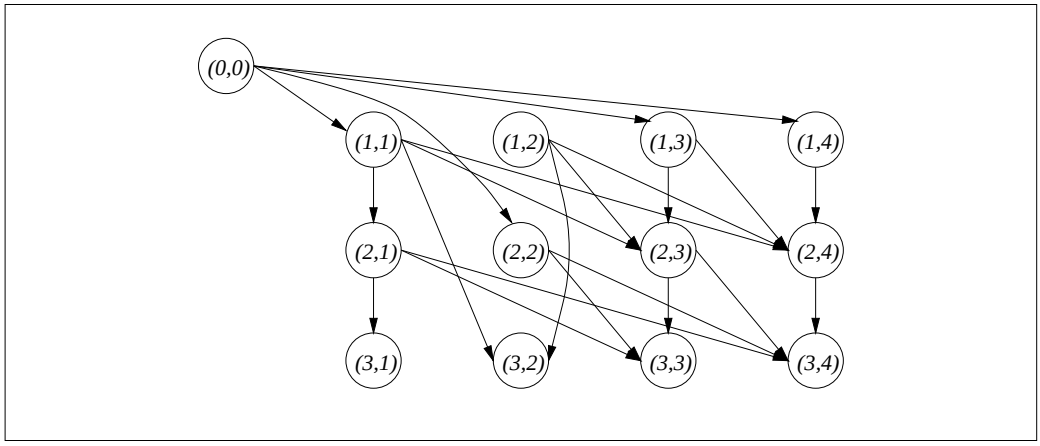

Figure 3: The vertices of network N_2 represent the states of Model II for the example problem (5). The edge weights of all edges pointing to a vertex (k,j), $k = 1,\ldots,b$, in this network are equal to the cost vector c_j. Note that some states in this model may be empty, e.g., $q(1,2) = \emptyset$.

Let $G(q(k,j))$ denote the set of all nondominated solutions in the state $q(k,j)$.

For convenience we assume in the following that the sets of nondominated solutions are initialized by $G(q(k,j)) = \{\underline{0}\}$ for all $k = 1,\ldots,b$, $j = 1,\ldots,n$.

Then the MCKP can be solved by applying the following recursive equations:

$$\begin{aligned} G(q(0,0)) &= \{\underline{0}\}, \\ G(q(k,j)) &= \text{vmax}\{G(q(k-a_j,p)) + c_j \,:\, k - a_j \geq 0,\ p \leq j\} \\ &\qquad\qquad k = 1,\ldots,b,\ j = 1,\ldots,n, \end{aligned}$$

where operation *vmax* computes the nondominated solutions of the set being the algebraic sum of the cost vector c_j and the set of all the nondominated solutions in the state $q(k-a_j,p)$.

Since all states in this model are finite states, the set of nondominated solutions $\mathcal{Y}_e$ is obtained as the vector-maximum of $G(q(0,0))$ and the union

of the sets $G(q(k,j))$, $k = 1,\ldots,b$, $j = 1,\ldots,n$; i.e.

$$\mathcal{Y}_e = \text{vmax} \bigcup_{\substack{k=1,\ldots,b \\ j=1,\ldots,n}} G(q(k,j)) \quad \cup \quad G(q(0,0)). \tag{7}$$

Model II contains considerably more states than Model I. On the other hand, multiple nondominated solutions are avoided due the structure of the recursion.

2.3 Model III

The model described in this section, developed by Garfinkel and Nemhauser [10] (equations I) and Ibaraki [13] (representation 3) for the single criterion case, and by Villarreal and Karwan ([29], approach 1), [30] for the MCKP with multiple constraints, can be viewed as an extension of Model II. A very similar model was also used by Eben-Chaime [9] to determine the supported nondominated solutions of the MCKP.

The set of states Q is defined as

$$Q := \{q(k,j) \,:\, k = 0,1,\ldots,b,\ j = 0,1,\ldots,n\}$$

where the state $q(k,j)$, $k = 0,1,\ldots,b$, $j = 0,1,\ldots,n$, represents all nonnegative integer solutions satisfying $\sum_{p=1}^{j} a_p x_p = k$, i.e.

$$q(k,j) := \{x \in I\!N_0^n \,:\, \sum_{p=1}^{j} a_p x_p = k,\ x_{j+1},\ldots,x_n = 0\}.$$

Note that in this model and similarly in Model IV (as it will be seen in the following section) the assumption that $x_{j+1},\ldots,x_n = 0$ for all $x \in q(k,j)$ could be omitted since each variable will be necessarily fixed during the decision process.

Each stage j, $j = 1,\ldots,n$, of the corresponding DP-procedure consists of the states $q(k,j)$, $k = 0,1,\ldots,b$. Thus stage j may be interpreted as consisting of those feasible solutions for which the first j variables are fixed.

In this model we define $q(k,0) = \{\underline{0}\}$ for all states $q(k,0)$, $k = 0,1,\ldots,b$ in the initial stage of the procedure. The set of final states Q_F is given by

$$Q_F := \{q(k,n) \,:\, k = 0,1,\ldots,b\}.$$

The decision of fixing a variable x_j to a value $\alpha \in \mathbb{N}_0$ in the state $q(k, j-1)$ corresponds to a transition from the state $q(k, j-1)$ to the state $q(k + a_j x_j, j)$ since this variable is fixed and the current right-hand side increases accordingly by $a_j x_j$.

Network N_3 (as shown in Figure 4 for the example problem (5)) depicts all possible transitions between the different states.

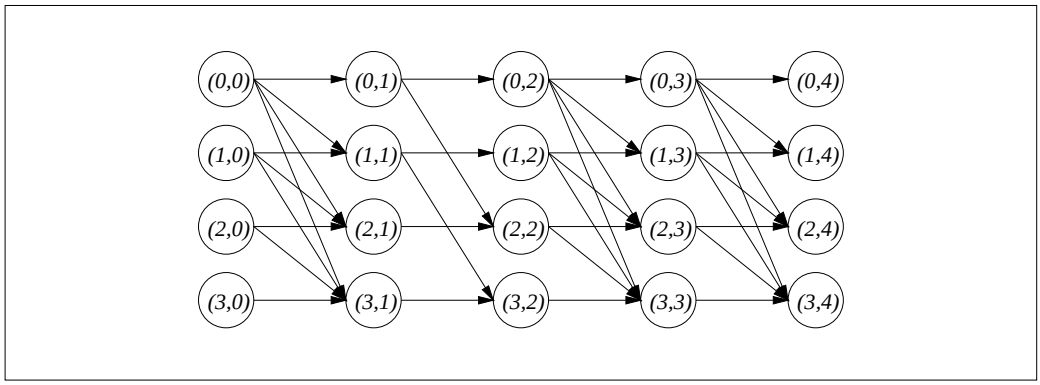

Figure 4: The vertices of network N_3 represent the states of Model III for the example problem (5). The weights of all horizontal arcs are equal to $(0, 0)$.

Let $G(q(k, j))$ denote the set of all nondominated solutions of the k-MCKP whose first j variables have been fixed.

With the states defined above, the MCKP can be solved by applying the following recursive equations:

$$\begin{aligned} G(q(k,0)) &= \{\underline{0}\}, \qquad k = 0, 1, \ldots, b \\ G(q(k,j)) &= \text{vmax}\{G(q(k-a_j x_j, j-1)) + x_j c_j \ : \ x_j \in \mathbb{N}_0, \ k - a_j x_j \geq 0\} \\ & \qquad\qquad k = 0, 1, \ldots, b, \ j = 1, \ldots, n, \end{aligned}$$

where operation $vmax$ computes the nondominated solutions of the set being the algebraic sum of the cost vector c_j and the set of all the nondominated solutions of the $(k - a_j x_j)$-MCKP whose first $j-1$ variables have been fixed.

The set of nondominated solutions $\mathcal{Y}_e$ is obtained as the vector-maximum of the union of the sets $G(q(k, n))$, $k = 0, 1, \ldots, b$; i.e.

$$\mathcal{Y}_e = \text{vmax} \bigcup_{k=0,1,\ldots,b} G(q(k,n)). \tag{8}$$

In this formulation multiple nondominated solutions are avoided as in the previous model since in each stage of the recursion exactly one additional variable is fixed.

2.4 Model IV

This model was developed by Garfinkel and Nemhauser [10] (equations II) for the single criterion case. It is very similar to Model II since variables are always increased by 1, and to Model III as decisions are consecutively made for the individual variables. A specific feature of this model is that in the recursion each state has exactly two predecessors.

Define the set of states Q as

$$Q := \{q(k,j) \,:\, k = 0,1,\ldots,b,\ j = 0,1,\ldots,n\}.$$

The state $q(k,j)$, $k = 0,1,\ldots,b$, $j = 0,1,\ldots,n$, represents all nonnegative integer solutions satisfying $\sum_{p=1}^{j} a_p x_p = k$, i.e.

$$q(k,j) := \{x \in I\!N_0^n \,:\, \sum_{p=1}^{j} a_p x_p = k,\ x_{j+1},\ldots,x_n = 0\}.$$

As in Model III, $q(k,0) = \{\underline{0}\}$ for all states $q(k,0)$, $k = 0,1,\ldots,b$ in the initial stage of the procedure. The set of final states Q_F is given by

$$Q_F := \{q(k,n) \,:\, k = 0,1,\ldots,b\}.$$

Here the decision of increasing the variable x_j by 1 in the state $q(k,j)$ corresponds to a transition from the state $q(k,j)$ to the state $q(k+a_j,j)$.

Network N_4 (see Figure 5 for the example problem (5)) shows the possible transitions between the different states.

Let $G(q(k,j))$ denote the set of all nondominated solutions of the k-MCKP in the state $q(k,j)$.

Then the following recursive equations yield a solution to the MCKP:

$$\begin{aligned} G(q(k,0)) &= \{\underline{0}\}, \qquad k = 0,1,\ldots,b \\ G(q(k,j)) &= \text{vmax}\{G(q(k,j-1)); G(q(k-a_j,j)) + c_j \,:\, k-a_j \geq 0\} \\ &\qquad\qquad k = 0,1,\ldots,b,\ j = 1,\ldots,n, \end{aligned}$$

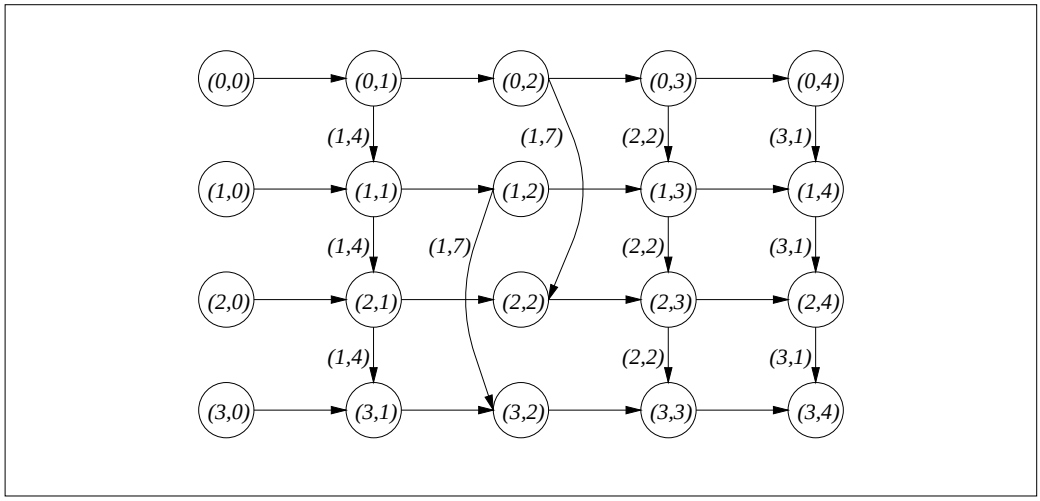

Figure 5: The states of Model IV for the example problem (5). The weights of all horizontal arcs are equal to $(0, 0)$.

where operation $vmax$ computes the nondominated solutions of the union of the set of all the nondominated solutions in the state $q(k, j-1)$ and the set being the algebraic sum of the cost vector c_j and the set of all the nondominated solutions in the state $q(k - a_j, j)$.

The set of nondominated solutions $\mathcal{Y}_e$ is obtained as the vector-maximum of the union of the sets $G(q(k, n))$, $k = 0, 1, \ldots, b$; i.e.

$$\mathcal{Y}_e = \text{vmax} \bigcup_{k=0,1,\ldots,b} G(q(k, n)). \tag{9}$$

2.5 Model V

Model V, given by Ibaraki [13] (representation 4) for the single criterion case, describes a completely different approach to solve the MCKP. Namely, the multiple-criteria and single constraint model is solved as a single criterion and multiple-constraint model.

Consider the following problem, where the roles of the objective functions and the constraint are interchanged with respect to problem (1):

$$\begin{array}{ll} \min & ax \\ \text{s.t.} & Cx \leq y \\ & x_j \geq 0, \text{ integer}, j = 1, \ldots, n. \end{array} \tag{10}$$

The right-hand side $y = (y_1, \ldots, y_m)^T \in \mathbb{R}^m_+$ in this model can be interpreted as a nonnegative vector value of problem (1).

For different right-hand sides y, let $z(y) \in \mathbb{R}_+$ denote the optimal objective value of (10). Then any solution that satisfies $z(y) \leq b$ is a feasible solution to the original problem (1). Thus the set of all nondominated solutions of (1) can be found as the set of nondominated right-hand side vectors y for which the optimal solution $z(y)$ of (10) satisfies $z(y) \leq b$.

Let

$$Y := \{y \in \mathbb{N}^m \, : \, y_i \in \{0, 1, \ldots, y_{i,\max}\}, i = 1, \ldots m\}$$

where $y_{i,\max}$ is an upper bound on the right-hand side component y_i. An example for the choice of $y_{i,\max}$ is

$$y_{i,\max} := b \cdot \max \left\{ \frac{c^i_j}{a_j} \, : \, j = 1, \ldots, n \right\}, \qquad i = 1, \ldots, m.$$

Using this derivation, the set Y for the example problem introduced in (5) would be given by

$$Y = \left\{ y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \in \mathbb{N}^2 \, : \, y_1 \in \{0, 1, \ldots, 9\}, \, y_2 \in \{0, 1, \ldots, 12\} \right\}.$$

Then a set of states Q can be defined as

$$Q := \{q(y, j) \, : \, y \in Y, \, j = 0, 1, \ldots, n\},$$

yielding

$$Q = \{q(y_1, y_2, j) \, : \, y_1 = 0, 1, \ldots, 9, \, y_2 = 0, 1, \ldots, 12, \, j = 0, 1, \ldots, 4\}$$

for the example problem and thus a total of 650 states!

Analogously to Model III, the state $q(y,j)$, $y \in Y$, $j = 0,1,\ldots,n$, represents all the nonnegative integer solutions satisfying $\sum_{p=1}^{j} x_p c^p = y$, i.e.

$$q(y,j) := \{x \in I\!N_0^n \ : \ \sum_{p=1}^{j} x_p c_p = y, \ x_{j+1},\ldots,x_n = 0\}.$$

Each stage j, $j = 1,\ldots,n$, consists of the states $q(y,j)$, $y \in Y$.

As in Model III, we have $q(y,0) = \{\underline{0}\}$ for all states $q(y,0)$, $y \in Y$ in the initial stage of the procedure. The set of final states Q_F is defined as

$$Q_F := \{q(y,n) \ : \ y \in Y\}.$$

Let $G(q(y,j))$ denote the optimal solution $z(y)$ in the state $q(y,j)$.

Then the MCKP can be solved by applying the following recursive equations:

$$\begin{aligned} G(q(y,0)) &= \{0\}, \qquad y \in Y \\ G(q(y,j)) &= \min\{G(q(y - x_j c_j, j-1)) + a_j x_j \ : \ x_j \in I\!N_0, \ y - x_j c_j \geq \underline{0}\} \\ &\qquad\qquad y \in Y, \ j = 1,\ldots,n. \end{aligned}$$

Note that in this recursion no vector-minimization is needed and that each set $G(q(y,j))$ contains exactly one scalar. Namely $G(q(y,j))$ is the solution $z(y,j)$ of problem (10) with right-hand side y, where only the first j variables $x_1,\ldots,x_j$ are considered.

The set of nondominated solutions $\mathcal{Y}_e$ of the original problem (1) is obtained as the vector-maximum of all the feasible solutions of (1) determined by the above recursion, i.e.

$$\mathcal{Y}_e = \text{vmax}\{y \in Y \ : \ G(q(y,n)) \leq b\}. \tag{11}$$

2.6 Comparison of the models

We now compare and discuss the five models representing the MCKP. Although the states of all the models are defined differently, each model represents a loop-free sequential decision making process, i.e., a process whose states can be indexed in an increasing order so that the transition from one state always takes place to a state with a higher index.

As the definition of the state is closely connected to the definition of the decision taken at every state, almost every model employs another decision concept. Observe that the recursions given in Models I, II, and IV are based on sequential decisions to increase a variable by 1 while in Models III and V they are based on decisions to fix a variable to some integer value α. Fixing a variable implies that once a value is assigned to the variable, it is never changed throughout the decision process. On the other hand, in Model I a variable x_j can be increased at any time during the process while Models II - V use a different approach where the variables x_j of a feasible solution x are determined in the order $j = 1, \ldots, n$.

The definition of the decision to be taken in every state determines whether a model is a multi-stage decision process or not, i.e. a process in which each transition from a state always takes place to a state at the next stage. We conclude that only Models III and V represent a multi-stage sequential decision process.

The structure of each model, including the number of states, the number of final states, and the number of transition edges, affects the computational complexity of each formulation. In this regard, Model I is superior to the others while Model V is the most complex. Simplicity of Model I, however, produces multiple nondominated solutions, as mentioned in Section 2.1, which is an additional although computationally insignificant obstacle not featured by the other models.

A summary of all the features of the five models is presented in Table 1.

3 Extensions

In this section we propose several extensions of the basic MCKP and relate them to real-life applications. We also discuss the extensions in the context of the five models of Section 2 and recommend the most efficient DP approach to each of them.

3.1 The binary multiple criteria knapsack problem

The binary MCKP as well as the integer MCKP with bounded variables are of special interest since they are needed to model many real-life situations. As an example consider a set of different projects that a decision maker may

	Model I	Model II	Model III	Model IV	Model V
decision taken at the state	$q(k)$: any variable can be increased by 1	$q(k,j)$: the variable x_j or any other variable x_i, $i > j$ can be increased by 1	$q(k,j)$: the variable x_{j+1} is fixed	$q(k,j)$: the variable x_j can be increased by 1 or the next variable x_{j+1} is considered	analogous to Model III
loop-free	yes	yes	yes	yes	yes
multi-stage	no	no	yes	no	yes
final states	$q(k)$, k=0, 1, ..., b	$q(k,j)$, k=0, 1, ..., b, j=1, ..., n	$q(k,n)$, k=0, 1, ..., b	$q(k,n)$, k=0, 1, ..., b	$q(y,n)$, $y \in Y$
# of states	$b+1$	$bn+1$	$(b+1)(n+1)$	$(b+1)(n+1)$	$(n+1)\|Y\|$
# of transition edges	nb	$O(n^2b)$	$O(nb^2)$	$n(2b+1)$	$O(n\|Y\|^2)$

Table 1: Comparison of Models I-V

either choose or not. Since a partial engagement may not be possible, binary variables are needed to model this type of problem.

In this section we concentrate on the binary MCKP (BMCKP) formulated as

$$\begin{aligned} \text{vmax} \quad & f(x) = Cx \\ \text{s.t.} \quad & ax \leq b \\ & x \in \{0,1\}^n. \end{aligned} \tag{12}$$

All five DP approaches described in the previous section can be adapted to the case of binary variables since the BMCKP can be formulated as the (integer) MCKP with multiple constraints (see Section 3.2). Nevertheless, we mainly focus on Models II and III since binary variables can be incorporated into these models without increasing the overall number of states. This is not possible in the case of Models I and IV where the additional information about the value of each variable has to be incorporated into the different states. Model V will be disregarded in this discussion since it does not seem to be favorable due to a large number of states.

We first modify Model II to handle binary variables. Each state $q(k,j)$

is redefined as

$$q(k,j) := \{x \in \{0,1\}^n \,:\, \sum_{p=1}^{j} a_p x_p = k,\; x_j = 1,\; x_{j+1}, \ldots, x_n = 0\}.$$

The decision of fixing a variable x_j to 1 in a state $q(k,p)$ with $0 \leq p < j$ (compare with Model II) corresponds to a transition from the state $q(k,p)$ to the state $q(k+a_j, j)$. The corresponding decision process is represented by the network given for the example problem (5) in Figure 6.

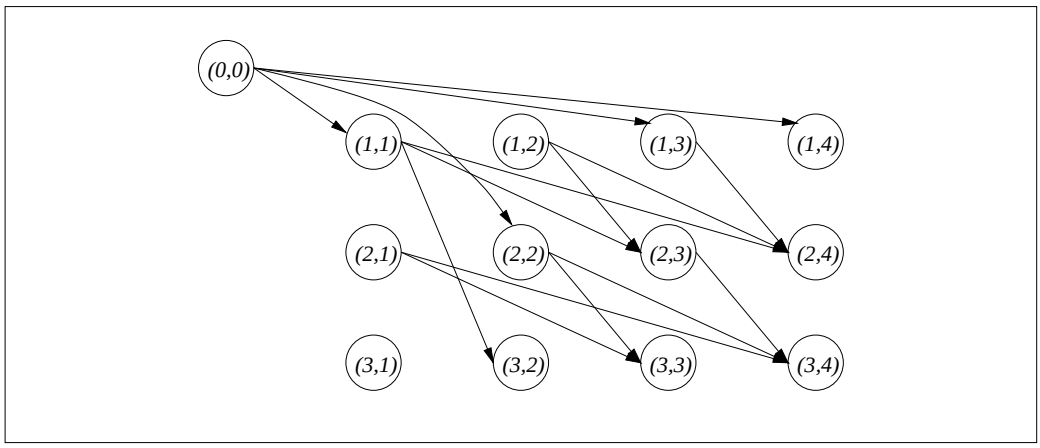

Figure 6: The edge weights of all edges pointing to a vertex (k,j), $k = 1, \ldots, b$, in this network are equal to the cost vector c_j.

Similar to the recursion defined in Model II, the following recursive equations can be used to solve the BMCKP:

$$\begin{aligned} G(q(0,0)) &= \{\underline{0}\}, \\ G(q(k,j)) &= \text{vmax}\{G(q(k-a_j,p)) + c_j \,:\, k - a_j \geq 0,\; p < j\} \\ &\qquad\qquad k = 1, \ldots, b,\; j = 1, \ldots, n. \end{aligned}$$

The set of nondominated solutions $\mathcal{Y}_e$ of the BMCKP can be then found applying formula (7).

A similar adaptation to the BMCKP can be given for Model III. In this model, a state $q(k,j)$ is defined as

$$q(k,j) := \{x \in \{0,1\}^n \,:\, \sum_{p=1}^{j} a_p x_p = k,\; x_{j+1}, \ldots, x_n = 0\}.$$

Similar to the original model, the decision of fixing a variable x_j to 0 or 1 in the state $q(k, j-1)$ corresponds to a transition from the state $q(k, j-1)$ to the state $q(k + a_j x_j, j)$.

The network for the example problem (5) given in Figure 7 depicts all possible transitions between the different states as well as the corresponding objective values (c_j^1, c_j^2).

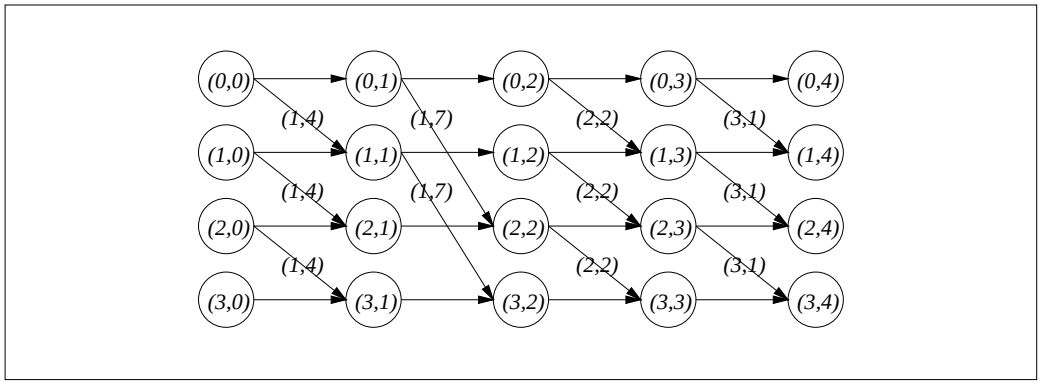

Figure 7: The weights of all horizontal arcs in this network are equal to $(0, 0)$.

With the states defined above, the BMCKP can be solved by applying the following recursive equations:

$$
\begin{aligned}
G(q(k,0)) &= \{\underline{0}\}, \qquad k = 0, 1, \ldots, b \\
G(q(k,j)) &= \operatorname{vmax}\{G(q(k - a_j x_j, j-1)) + x_j c_j \ : \ x_j \in \{0,1\},\ k - a_j x_j \geq 0\} \\
&\qquad\qquad k = 0, 1, \ldots, b,\ j = 1, \ldots, n.
\end{aligned}
$$

The set of nondominated solutions $\mathcal{Y}_e$ of the BMCKP can be then found applying formula (8).

Both models solve the BMCKP utilizing the same number of states as was needed to solve the integer MCKP. Model II leads again to a loop-free procedure whereas Model III is a multi-stage procedure as was discussed in Section 2. Note that the methods developed in this section can also be easily adapted to handle integer models with bounded variables.

3.2 Models with multiple constraints

Knapsack models with more than a single constraint have many applications especially in affordability analysis. Often more than one budget category has

to be taken into account before deciding whether a project should be undertaken or not. Each budget category may have its own budget constraints. Furthermore, the multiple constraint knapsack problem can be viewed as the most general formulation of the knapsack problem, including e.g. the binary problem as well as problems with bounded variables as special cases.

The resulting problem is the multiple constraint multiple criteria knapsack problem (MCMCKP):

$$\begin{array}{ll} \text{vmax} & f(x) = Cx \\ \text{s.t.} & Ax \leq B \\ & x_j \geq 0, \text{ integer}, j = 1, \ldots, n \end{array} \tag{13}$$

with s linearly independent constraints, where A is an $s \times n$ matrix and B is an s-dimensional vector. We denote the ith row of A by a^i and the jth column of A by a_j.

Analogously to the original problem formulation, we assume that the weights a^i_j, $i = 1, \ldots, s$, $j = 1, \ldots, n$ are nonnegative integers and the capacities b_i, $i = 1, \ldots, s$, are positive integers. In order to avoid trivial solutions let $0 \leq a^i_j \leq b_i$, $i = 1, \ldots, s$, $j = 1, \ldots, n$, and $\sum_{j=1}^{n} a^i_j > b_i$, $i = 1, \ldots, s$.

As in the case of the BMCKP, Models I through IV given in Section 2 can be adapted so that the MCMCKP can be solved by the corresponding DP procedures. One way of achieving this is obviously the transformation of the problem with multiple constraints into a problem with a single constraint so that both problems have the same set of feasible solutions. The integration of several constraints into a single constraint is possible, on the other hand the tradeoff of this approach is a considerably increased value of the right-hand-side b [23].

A more efficient way of incorporating multiple constraints into Models I through IV is the introduction of an s-dimensional right-hand-side vector $\underline{0} \leq K \leq B$, replacing the right-hand-side $k \in \mathbb{N}_0$ in the single constraint model. The corresponding K-MCMCKP is accordingly defined as the MCMCKP with the right-hand-side vector K, $\underline{0} \leq K \leq B$. The notation $K \leq B$ is defined as $k_i \leq b_i$, $i = 1, \ldots, s$ where $K = [k_1, \ldots, k_s]$.

Using this notation, the recursions given in Models I through IV in Section 2 can be immediately applied to solve the MCMCKP, where the overall

number of states needed in each model can be calculated as follows:

model	number of states
I	$\prod_{i=1}^{s}(b_i+1)$
II	$1+n\cdot\prod_{i=1}^{s}b_i$
III & IV	$(n+1)\cdot\prod_{i=1}^{s}(b_i+1)$.

As Model I uses the smallest number of states, we illustrate our approach with this model. Villarreal and Karwan [29] used the same model for their analysis of the MCMCKP.

A state $q(K)$ is defined as

$$q(K) := \{x \in I\!N_0^n \,:\, Ax = K\}.$$

Figure 8 depicts the decision process of Model I for problem (5) extended by the second constraint $2x_2 + x_3 + 2x_4 \leq 2$.

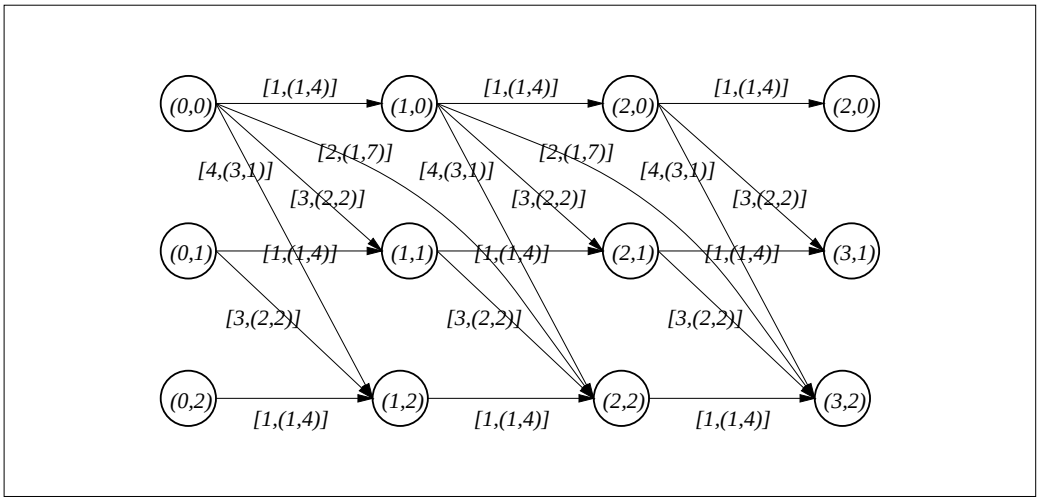

Figure 8: States and possible transitions between states with respect to Model I for the example problem (5) extended by the constraint $2x_2 + x_3 + 2x_4 \leq 2$.

Let $G(q(K))$, $\underline{0} \leq K \leq B$, be the set of all nondominated solutions of the K-MCKP. Then the MCMCKP can be solved applying the following recursive equations:

$$\begin{aligned} G(q(\underline{0})) &= \{\underline{0}\} \\ G(q(K)) &= \text{vmax}\{G(q(K-a_j)) + c_j \,:\, j \in S,\ K - a_j \geq \underline{0}\}, \quad \underline{0} \leq K \leq B. \end{aligned}$$

The set of nondominated solutions $\mathcal{Y}_e$ of the MCMCKP can be then found applying formula (6) and taking the union over all $\underline{0} \leq K \leq B$.

3.3 Multiperiod models

Problems with multiple time periods are frequently encountered in practice. In many real-life situations the available budget depends on the fiscal year or other time periods during which certain expenses have to be made.

In [8], Dudziński and Walukiewicz propose a multiple period model of the binary single criterion knapsack problem and solve it using LP-relaxation. Following their model, we formulate the multiple period multiple constraint knapsack problem (MPMCKP) as:

$$\begin{array}{ll} \text{vmax} & f(x) = Cx \\ \text{s.t.} & \sum_{i=1}^{r} \sum_{j \in S_i} a_j x_j \leq b_r, \quad r = 1, \ldots, s \\ & x_j \geq 0, \text{ integer}, \; j = 1, \ldots, n \end{array} \tag{14}$$

where the index set of the variables $S = \{1, \ldots, n\}$ is partitioned into s pairwise disjoint subsets $S_1, \ldots, S_s$ so that $S = \bigcup_{r=1}^{s} S_r$. The constraint coefficients a_j, $j = 1, \ldots, n$ and b_r, $r = 1, \ldots, s$ are positive integers satisfying $0 < b_1 \leq \cdots \leq b_s$. Each of the subsets S_r, $r = 1, \ldots, s$ can be interpreted as corresponding to the time period r. A variable x_j may be changed from 0 to a value $\alpha \in I\!N_0$ in the r-th time period only if $j \in S_r$. Thus the set S_r can be interpreted as a set of projects which may be selected only in one given time period. Each time period has its individual budget constraint. If the budget is not entirely used in one time period, the remaining budget is still available in the following time period. On the other hand no part of the budget can be used before it gets available. The total available budget in time period r thus is given by b_r, $r = 1, \ldots, s$.

First notice that the problem (14) is a special case of the knapsack model with multiple constraints discussed in Section 3.2 and therefore can be solved using the corresponding recursive equations.

A different approach based on Model III that exploits the special structure of the staircase constraints and thus uses remarkably less states is given in this section.

Assume that the variables are sorted so that $S_1 = \{j_0, \ldots, j_1\}, S_2 = \{j_1 + 1, \ldots, j_2\}, \ldots, S_s = \{j_{s-1} + 1, \ldots, j_s\}$ where $j_0 = 1$ and $j_s = n$. Let $S_r = \{j_{r-1} + 1, \ldots, j_r\}, r = 1, \ldots, s$. The set Q of states is defined as

$$Q := \bigcup_{r=1}^{s} \{q(k,j) \,:\, k = 0, 1, \ldots, b_r,\ j \in S_r\} \quad \cup \quad \{q(k,0) \,:\, k = 0, 1, \ldots, b_1\},$$

where the state $q(k,j)$, $k = 0, 1, \ldots, b_r$, $j \in S_r$, is defined as

$$q(k,j) := \{x \in I\!N_0^n \,:\, \sum_{p=1}^{j} a_p x_p = k,\ x_{j+1}, \ldots, x_n = 0\}.$$

Similar to the original Model III, the decision of fixing a variable x_j to $\alpha \in I\!N_0$ in the state $q(k, j-1)$ corresponds to a transition from the state $q(k, j-1)$ to the state $q(k + a_j x_j, j)$.

Figure 9 shows all possible transitions between states for the example problem (5) with two time periods. Time period 1 is given by $S_1 = \{1, 2\}$ and time period 2 is given by $S_2 = \{3, 4\}$. The corresponding right-hand-sides of the capacity constraints are $b_1 = 2$ and $b_2 = 3$.

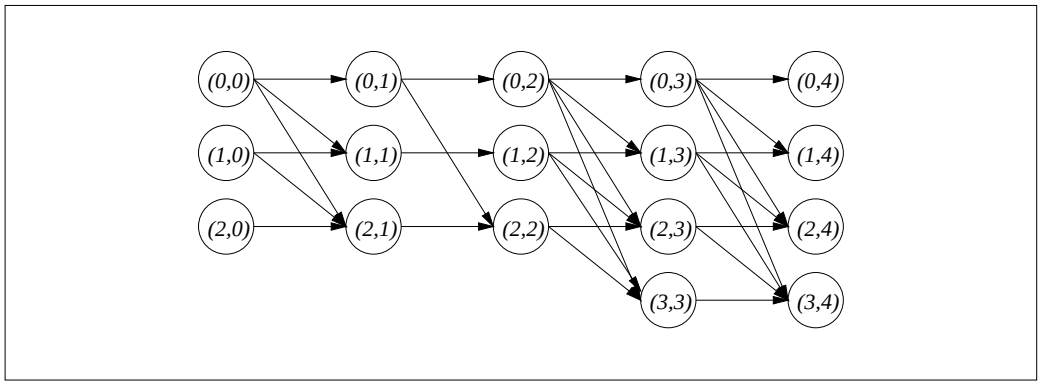

Figure 9: The weights of all horizontal arcs are equal to $(0, 0)$.

For convenience, we assume that the sets of nondominated solutions are initialized by $G(q(k,j)) = \{\underline{0}\}$ for all $k = 0, 1, \ldots, b_s$, $j = 0, 1, \ldots, n$. Then the MPMCKP can be solved applying the following recursive equations:

$$\begin{aligned} G(q(k,0)) &= \{\underline{0}\}, \qquad k = 0, 1, \ldots, b_1 \\ G(q(k,j)) &= \text{vmax}\{G(q(k - a_j x_j, j-1)) + x_j c_j \,:\, x_j \in I\!N_0,\ k - a_j x_j \geq 0\} \\ &\qquad\qquad r = 1, \ldots, s,\ k = 0, 1, \ldots, b_r,\ j \in S_r. \end{aligned}$$

The set of nondominated solutions $\mathcal{Y}_e$ of the MPMCKP can be then found applying formula (8) and taking the union over all $k = 0, 1, \ldots, b_s$.

Using these recursive equations, the MPMCKP can be solved with the same number of states as was needed to solve the MCKP in Model III. The total number of states is bounded by $(n + 1)(b_s + 1)$ and the obtained DP procedure is again a multi-stage process as was already discussed in Section 2.

Further extensions such as the incorporation of binary or bounded variables (see Section 3.1) are also possible.

3.4 Time-dependent models

Introducing time-dependency to the knapsack model is a challenging task however dictated by real-life applications that feature time-varying data. Furthermore, in many applications the parameters of the knapsack problem may change over time (e.g., variable earnings or revenues to be maximized), which has been recently recognized by Kleywegt and Papastavrou [16].

A time dependent version of the MCKP in which the vector of objective functions is composed of time-dependent functions, i.e. the time-dependent multiple criteria knapsack problem (TDMCKP) can be solved using Model I. Under the assumption that time is to be minimized while other monotonous, time dependent functions are to be maximized, recursive equations based on those developed in Model I can be derived in order to obtain all the nondominated solutions of the TDMCKP. A distinct feature of this model is that its feasible solutions are defined as sequences of elements to be chosen at different times in a decision process. For a detailed discussion of this model the reader is referred to [14].

4 Conclusions

In this paper we study the basic MCKP and its more complex extensions including binary variables, multiple constraints, multiple periods, and time-dependent criterion functions.

We first reviewed the DP approaches available in the literature (almost all to the single criterion problem) and then generalized them to handle the basic problem and the extensions. We hence proposed a comprehensive DP-framework able to solve a broad class of knapsack problems.

We believe that this class could be enlarged with knapsack models featuring uncertainty or time-dependent constraints. Another direction of further research could aim at efficient implementations of the proposed DP approaches.

References

[1] R. Bellman. *Dynamic Programming.* Princeton University Press, Princeton, New Jersey, 1957.

[2] K. Bhaskar. A multiple objective approach to capital budgeting. *Accounting and Business Research*, 9:25–46, 1979.

[3] G. R. Bitran. Linear multiple objective programs with zero-one variables. *Mathematical Programming*, 13:121–139, 1977.

[4] G. R. Bitran. Theory and algorithms for linear multiple objective programs with zero-one variables. *Mathematical Programming*, 17:362–390, 1979.

[5] T.A. Brown and R. E. Strauch. Dynamic programming in multiplicative laticces. *Journal of Mathematical Analysis and Applications*, 12:364–370, 1965.

[6] L. G. Chalmet, L. Lemonidis, and D. J. Elzinga. An algorithm for the bi-criterion integer programming problem. *European Journal of Operational Research*, 25:292–300, 1986.

[7] H.W. Corley and I.D. Moon. Shortest paths in networks with vector weights. *Journal of Optimization Theory and Applications*, 46:79–86, 1985.

[8] K. Dudzinski and S. Walukiewicz. On the multiperiod binary knapsack problem. *Methods of Operations Research*, 50:223–232, 1985.

[9] M. Eben-Chaime. Parametric solution for linear bicriteria knapsack models. *Management Science*, 42:1565–1575, 1996.

[10] R. S. Garfinkel and G. L. Nemhauser. *Integer Pogramming.* John Wiley & Sons, New York, 1972.

[11] R. Hartley. Vector optimal routing by dynamic programming. In P. Serafini, editor, *Mathematics of Multiobjective Optimization*, pages 215–224. Springer-Verlag, Vienna, 1985.

[12] M. Henig. Vector-valued dynamic programming. *SIAM J. Control and Optimization*, 21:490–499, 1983.

[13] T. Ibaraki. Enumerative approaches to combinatorial optimization, part ii. In P. L. Hammer, editor, *Annals of Operations Research*, volume 11, pages 343–602. Baltzer, Basel, 1987.

[14] K. Klamroth and M.M. Wiecek. A time-dependent multiple criteria knapsack problem. Technical Report 664, Dept. of Math. Sc., Clemson University, Clemson, SC, 1998.

[15] D. Klein and E. Hannan. An algorithm for the multiple objective integer linear programming problem. *European Journal Of Operational Research*, 93:378–385, 1982.

[16] A. J. Kleywegt and J. D. Papastavrou. The dynamic and stochastic knapsack problem. *Operations Research*, 46:17–35, 1998.

[17] R. Klötzler. Multiobjective dynamic programming. *Math. Operationsforsch. Statist., Ser. Optimization*, 9:423–426, 1978.

[18] M. M. Kostreva, W. Ogryczak, and D. W. Tonkyn. Relocation problems arising in conservation biology. *Computers and Mathematics with Applications.* to appear.

[19] W. Kwak, Y. Shi, H. Lee, and C.F. Lee. Capital budgeting with multiple criteria and multiple decision makers. *Review of Quantitative Finance and Accounting*, 7:97–112, 1996.

[20] S. Martello, D. Pisinger, and P. Toth. New trends in exact algorithms for the 0-1 knapsack problem. In J. Barceló, editor, *Proceedings of EURO/INFORMS-97, Barcelona*, pages 151–160, 1997.

[21] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations.* John Wiley & Sons, New York, 1990.

[22] M.J. Rosenblatt and Z. Sinuany-Stern. Generating the discrete efficient frontier to the capital budgeting problem. *Operations Research*, 37:384–394, 1989.

[23] H.M. Salkin. *Integer Programming.* Addison-Wesley, Reading, Mass., 1975.

[24] J.-Y. Teng and G.-H. Tzeng. A multiobjective programming approach for selecting non-independent transportation investment alternatives. *Transportation Research - B*, 30:291–307, 1996.

[25] T. Trzaskalik. Multiple criteria discrete dynamic programming. *Mathematics Today*, XII-A:173–199, 1994.

[26] E.L. Ulungu and J. Teghem. Application of the two phases method to solve the bi-objective knapsack problem. Technical report, Department of Mathematics & Operational Research, Faculté Polytechnique de Mons, Belgium, 1994.

[27] E.L. Ulungu and J. Teghem. Solving multi-objective knapsack problems by a branch-and-bound procedure. In J.N. Climaco, editor, *Multicriteria Analysis*, pages 269–278. Springer-Verlag, 1997. to appear.

[28] R. Vetschera. Time preferences in capital budgeting - an application of interactive multiobjective optimization. *Methods of Operations Research*, 50:649–660, 1985.

[29] B. Villarreal and M. H. Karwan. Multicriteria integer programming: A (hybrid) dynamic programming recursive approach. *Mathematical Programming*, 21:204–223, 1981.

[30] B. Villarreal and M. H. Karwan. Multicriteria dynamic programming with an application to the integer case. *Journal of Mathematical Analysis and Applications*, 38:43–69, 1982.

[31] M. Visée, J. Teghem, M. Pirlot, and E.L. Ulungu. Two-phases method and branch and bound procedures to solve the bi-objective knapsack problem. Technical report, Department of Mathematics & Operational Research, Faculté Polytechnique de Mons, Belgium, 1996.

[32] P.L. Yu and L. Seiford. Multistage decision problems with multiple criteria. In P. Nijkamp and J. Spronk, editors, *Multiple Criteria Analysis*, pages 235–244. Gower, Aldershot, England, 1981.