Knapsack Problems involving dimensions, demands and multiple choice constraints: generalization and transformations between formulations

Amine Lamine, Mahdi Khemakhem and Habib Chabchoub
Higher Institute of Industrial Management
University of Sfax, 3018 Sfax , Tunisia
amine.lamine@yahoo.fr, mahdi.khemakhem@isecs.rnu.tn, habib.chabchoub@fsegs.rnu.tn

Abstract

In this paper we studied a set of knapsack problems involving the notion of dimensions, demands and multiple choice constraints. Specifically, we defined a new problem called the multiple demand multidimensional multiple choice knapsack problem and we showed it as a generalization of other related problems. Moreover, we presented a set of transformations between the different integer linear programs of the studied problems. Using these transformations, we showed that any algorithm able to solve the generalized problem can definitely solve its related problems. Then, we tested the new integer linear programs on different sets of benchmarks using the commercial software Cplex 9.0 . Computational results highlighted the ability of the generated formulations to produce a reasonable CPU time value compared with the original ones.

Keywords: Modeling, generalization of knapsack problem, multiple demand multidimensional multiple choice constraints, integer linear programs transformations

1 Introduction

Extensions of knapsack problem [7, 12] play a significant role in the study of discrete mathematics. These extensions concern many practical problems in the real life as the service level agreement, the model of allocation resources, computer systems design, project selection, cutting stock and cargo-loading.

In the literature, large specific algorithms are developed to solve extensions of the knapsack problem. Although there is a relation between many knapsack problems, many existing algorithms must be redeveloped to solve other extensions. To be appropriate to the problem, the redevelopment of these algorithms is very difficult in general and can lose its specifications.

In this context, we define a new problem called the multiple demand multidimensional multiple choice knapsack problem (MDMMKP) which is considered as a generalization of its related problems. Using a set of transformations, we show that any algorithm able to solve the generalized problem can definitely solve its related problems. Therefore the redevelopment of algorithms in this case is not considered.

The outline of this paper is as follows. In Section 2, we define the concepts and the preliminaries of our work. In section 3, a definition of knapsack problems is given. In section 4, we present a set of transformations between integer linear programs of the knapsack problems. In Section 5, computational results are given to validate these transformations. Finally, in Section 6, we conclude and outline our future directions.

2 Preliminaries

2.1 Integer linear program

Integer linear programming refers to mathematical programming with discrete variables and linearities in the objective function and constraints.

The general form of an integer linear program (ILP) is (see Figure 1)

Figure 1. A general form of an integer linear program

$$\boxed{\begin{array}{ll} \text{maximize} & c^T x \\ & \\ \text{subject to} & \\ & Ax \leq b \\ & x \in X \text{ Integer} \end{array}}$$

where c is a n-vector, A is an $m \times n$ matrix and b is an m-vector. x is the decision variables required to be integer valued. X is bounding-box-type restrictions on the variable. We refer to [3] for more information about ILP fundamentals.

2.2 Knapsack problem constraints

The knapsack problem (KP) [7, 12] is an integer linear program comprising binary variables, a single constraint with positive coefficients and binary restrictions on the variables. It can be defined by a set of items, each having a profit and a weight. The problem consists in choosing a subset of the items such that their overall profit is maximized so that the overall weight does not exceed a given limit (knapsack capacity).

The KP has been used to model various decision making processes and finds a variety of real world applications: resource allocation problems, cutting stock, capital budgeting, project selection and processor allocation in distributed computing systems. Industrial applications find the need for satisfying additional constraints such as multidimensional knapsack constraints, demand constraints and multiple choice constraints. These constraints can be defined as follows:

- ***Knapsack constraint***: The knapsack constraint is to choose a subset of items set such that their overall weight does not exceed a knapsack capacity. In case when the knapsack has a set of dimensions, the constraint is called multidimensional knapsack constraint in which each dimension is called a knapsack constraint.
- ***Demand constraint***: The demand constraint is to choose a subset of items set such that their overall weight must exceed a demand capacity. As the previous constraint, the demand constraint can be multidimensional.
- ***Multiple choice constraint***: When items are distributed on a set of disjointed sets then the multiple choice constraint is to choose an item of each set.

These necessities (additional constraints) lead to many extensions and variants of knapsack problems such as the multidimensional knapsack problem (MKP) [5], the multiple demand knapsack problem (MDMKP) [2], the multiple choice knapsack problem (MCKP) [14], the multidimensional multiple choice knapsack problem (MMKP) [13, 8] and multidimensional knapsack problems with generalized upper bound constraints (GUBMKP) [10, 11]. In Table 1, we characterize the problems that will be defined in the following according to the type of constraints.

Table 1. Knapsack problems according to the type of constraints

	Type of constraint			
Problem	knapsack	multidimensional knapsack	multiple demand	multiple choice
KP	X			
MKP		X		
MDMKP		X	X	
MCKP	X			X
MMKP		X		X
GUBMKP		X		X
MDMMKP		X	X	X

MDMMKP is an abbreviation of the multiple demand multidimensional multiple choice knapsack problem that we define in the next section and mentioned as the most general problem of all these problems.

In order to give a more intuitive presentation, let us define and denote the following terms (see Table 2).

Table 2. Notation of knapsack terms

N	the number of items
p_j	the profit of item j
w_j	the weight of item j
c	the capacity of a single knapsack
m	the number of knapsack constraints
w_j^k	the weight of item j in knapsack k
c^k	the capacity of knapsack k
q	the number of demand constraints
n	the number of groups
$G = \{G_1, ..., G_n\}$	the set of groups
$\|G_i\|$	the number of items of group G_i
p_{ij}	the profit of item j of group G_i
w_{ij}^k	the weight of item j of group G_i in knapsack k

2.3 Reduction, generalization and problems transformation

- *Reduction and generalization*
  Of any two related problems, such as A and B, we say that the problem A is a generalization of the problem B (marked B$\rightarrow$ A) if and only if:
  - every solution to problem A is also a solution to problem B; and
  - there are solutions to problem B which are not solutions to problem A.

  Note that the reduction is the symmetric relationship of the generalization then the problem B is a reduction of problem A.
  Also it is obvious that the generalization and the reduction are a transitive relation. For example if B$\rightarrow$ A and C$\rightarrow$ B then C$\rightarrow$ A.

- *Problem transformation*
  We define a problem transformation as the operation that takes a problem and generates it into another problem in accordance with a set of rules without losing specifications.

3 The Knapsack Problem Family involving the notion of dimensions, demands and sets

3.1 The knapsack problem

The knapsack problem (KP) can be defined by a set of N items; each item j has a profit p_j and a weight w_j. The problem is to choose a subset of the items such that their overall profit is maximized, while the overall weight does not exceed the knapsack capacity c. It may be formulated as the following integer linear program (see Figure 2)

Figure 2. An integer linear program of the knapsack problem

$$\begin{array}{lllr} \text{maximize} & \sum_{j=1}^{N} p_j x_j & & (1) \\ & & & \\ \text{subject to} & & & \\ & \sum_{j=1}^{N} w_j x_j \leq c & & (2) \\ & x_j \in \{0,1\} & (j = 1, \dots, N) & (3) \end{array}$$

where Equation (1) provides the total profit of selecting items and Equation (2) ensures that the knapsack constraint is satisfied. The binary decision variables x_j are used to indicate whether item j is included in the knapsack or not.

There follows (Figure 3) a small illustrative problem which will be used throughout the paper. For this problem the number of items N is equal to 8.

3.2 The multidimensional knapsack problem

The multidimensional knapsack problem (MKP) [5] is considered as an extension of the classical knapsack problem in which knapsack has a set of dimensions. Each dimension is called a knapsack constraint.

The MKP can be defined by a set of N items and a knapsack with m dimensions. The knapsack has a limited capacity in each dimension k denoted by c^k. Each item j has a profit p_j and a weight in each dimension, denoted by w_j^k. The goal is to select a subset of items with maximum total profit, see Equation (4). Chosen items must, however, not exceed knapsack constraints, see Equation (5). The 0-1 decision variables x_j indicate which items are selected.

Figure 5 represents an extended example of the knapsack problem to present the multidimensional knapsack problem where the number of dimensions m is equal to 2.

3.3 The multiple demand multidimensional knapsack problem

The multiple demand multidimensional knapsack problem (MDMKP) [2] is considered as an extension of the multidimensional knapsack problem in which there are *greater-than-or-equal-to*

Figure 3. An instance example of the knapsack problem

Input:

$N \leftarrow 8$

$$\begin{array}{ccccccccc}
 & & & & \rightarrow \mathcal{P}rofits \rightarrow & & & & \\
p_j & 10 & 20 & 30 & 40 & 50 & 60 & 70 & 80 \\
 & & & & \rightarrow \mathcal{W}eights \rightarrow & & & & \\
w_j & 5 & 20 & 25 & 35 & 40 & 45 & 55 & 60 \\
 & & & & \rightarrow \mathcal{C}apacity \rightarrow & & & & \\
c & & & & 150 & & & &
\end{array}$$

Constraints:

$$5x_1 + 20x_2 + 25x_3 + 35x_4 + 40x_5 + 45x_6 + 55x_7 + 60x_8 \leq 150$$
$$x_j \in \{0, 1\} \ \ (j = 1, \ldots, 8)$$

Objective:

$$maximize \ \ 10x_1 + 20x_2 + 30x_3 + 40x_4 + 50x_5 + 60x_6 + 70x_7 + 80x_8$$

Optimal solution:

$$\begin{array}{ccccccccccc}
x_j & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & \text{value} & 200
\end{array}$$

Figure 4. An integer linear program of the multidimensional knapsack problem

$$\begin{array}{lllr}
\text{maximize} & \sum_{j=1}^{N} p_j x_j & & (4) \\
 & & & \\
\text{subject to} & & & \\
 & \sum_{j=1}^{N} w_j^k x_j \leq c^k & (k = 1, \ldots, m) & (5) \\
 & x_j \in \{0, 1\} & (j = 1, \ldots, N) & (6)
\end{array}$$

inequalities called *demand constraints*, in addition to the standard *less-than-or-equal-to* inequalities. Formally the problem can be stated as integer linear program as shown in Figure 6.

Each of the m constraints of family Equation (8) represents a knapsack constraint, while each of the q constraints of family Equation (9) represents a demand constraint.

Figure 7 represents an example of the MDMKP which can be considered as an extension of the MKP example with two demand constraints ($q = 2$).

3.4 The multiple choice knapsack problem

The multiple choice knapsack problem (MCKP) [14] is considered as a an extension of the classical knapsack problem in which items are distributed on n disjointed groups $G = (G_1 \cup G_2 \cup \ldots \cup G_n)$ (see Equation 11).

The MCKP consists in selecting one and only one item of each group without violating the knapsack capacity c in order to maximize the total profit of the selected items. The MCKP can be modeled as an integer linear program as shown in Figure 8.

The variable x_{ij} is equal to 1 when the item j of the group G_i is selected, 0 otherwise. The objective function Equation (12) represents the total profit to be maximized. The knapsack constraint is presented in Equation (13) and the n multiple choice constraints are presented in Equation (14).

Figure 5. An instance example of the multidimensional knapsack problem

Input:

$N \leftarrow 8 \;\; m \leftarrow 2$

$\rightarrow \mathcal{P}rofits \rightarrow$

p_j	10	20	30	40	50	60	70	80

$\rightarrow \mathcal{W}eights \rightarrow$

w_j^k	5	20	25	35	40	45	55	60
	90	120	70	110	90	65	80	150

$\rightarrow \mathcal{C}apacity \rightarrow$

c^k 150 300

Constraints:

$$5x_1 + 20x_2 + 25x_3 + 35x_4 + 40x_5 + 45x_6 + 55x_7 + 60x_8 \leq 150$$
$$90x_1 + 120x_2 + 70x_3 + 110x_4 + 90x_5 + 65x_6 + 80x_7 + 150x_8 \leq 300$$
$$x_j \in \{0,1\} \;\; (j = 1, \ldots, 8)$$

Objective:

$$maximize \;\; 10x_1 + 20x_2 + 30x_3 + 40x_4 + 50x_5 + 60x_6 + 70x_7 + 80x_8$$

Optimal solution:

x_j 0 0 1 0 0 0 1 1 value 180

Figure 6. An integer linear program of the multiple demand multidimensional knapsack problem

$$\begin{array}{lllr} \text{maximize} & \sum_{j=1}^{N} p_j x_j & & (7) \\ & & & \\ \text{subject to} & & & \\ & \sum_{j=1}^{N} w_j^k x_j \leq c^k & (k = 1, \ldots, m) & (8) \\ & \sum_{j=1}^{N} w_j^k x_j \geq c^k & (k = 1 + m, \ldots, (m+q)) & (9) \\ & x_j \in \{0,1\} & (j = 1, \ldots, N) & (10) \end{array}$$

To avoid unsolvable situations we assume that the sum of the minimum weight of items in each group is smaller than the knapsack capacity c.

Figure 9 represents an example of MCKP where items are subdivided into $n = 3$ groups, the cardinality of $G_1 = 3$, $G_2 = 2$ and $G_3 = 3$.

3.5 The multidimensional multiple choice knapsack problem

The multidimensional multiple choice knapsack problem (MMKP) [13, 8] is a particular variant of the knapsack problem. It can be viewed as a combination of aspects of the multidimensional knapsack problem (MKP) and the multiple choice knapsack problem (MCKP). The MMKP is a an extension of the MCKP in which one item is selected from each group. However, in the MMKP, the knapsack is multidimensional, i.e., the knapsack consists of multiple resource constraints simultaneously satisfied. The MMKP problem can be stated as an integer linear program as shown

Figure 7. An instance example of the multiple demand multidimensional knapsack problem

Input:

$N \leftarrow 8 \;\; m \leftarrow 2 \;\; q \leftarrow 2$

$\rightarrow \mathcal{P}rofits \rightarrow$

p_j	10	20	30	40	50	60	70	80

$\rightarrow \mathcal{W}eights \rightarrow$

w_j^k	5	20	25	35	40	45	55	60
	90	120	70	110	90	65	80	150
	5	20	100	35	60	45	50	60
	90	60	70	110	90	45	20	10

$\rightarrow \mathcal{C}apacity \rightarrow$

c^k 150 300 80 200

Constraints:

$$\begin{aligned}
&5x_1 + 20x_2 + 25x_3 + 35x_4 + 40x_5 + 45x_6 + 55x_7 + 60x_8 \leq 150\\
&90x_1 + 120x_2 + 70x_3 + 110x_4 + 90x_5 + 65x_6 + 80x_7 + 150x_8 \leq 300\\
&5x_1 + 20x_2 + 100x_3 + 35x_4 + 60x_5 + 45x_6 + 50x_7 + 60x_8 \geq 80\\
&90x_1 + 60x_2 + 70x_3 + 110x_4 + 90x_5 + 45x_6 + 20x_7 + 10x_8 \geq 200\\
&x_j \in \{0,1\} \;\; (j = 1,\ldots,8)
\end{aligned}$$

Objective:

$$maximize \;\; 10x_1 + 20x_2 + 30x_3 + 40x_4 + 50x_5 + 60x_6 + 70x_7 + 80x_8$$

Optimal solution:

x_j	0	0	0	1	1	0	1	0	value	160

$$\forall\,(p,q)\,,\; p \neq q,\; p \leq n\;, q \leq n,\; G_p \cap G_q = \emptyset \text{ and } \bigcup_{i=1}^{n} G_i = G \qquad (11)$$

in Figure 10

Equation (16) provides the profit of selecting items, a value to be maximized. Equation (17) ensures the resource capacity of knapsack k is not exceeded while Equation (18) ensures selecting a single item from each of the G_i groups. Equation (19) is the binary selection requirement on decision variable x_{ij} such that x_{ij} is equal to 1 if the item j of the group G_i is selected, 0 otherwise.

Figure 11 represents an extended example of the multiple choice knapsack problem to present the multidimensional multiple choice knapsack problem where the number of dimensions m is equal to 2.

3.6 The multidimensional knapsack problems with generalized upper bound constraints

The multidimensional knapsack problems with generalized upper bound constraints (GUBMKP) [10, 11] is defined as a multidimensional knapsack problem (MKP) with mutually exclusive generalized upper-bound(GUB) constraints where all GUBs are fixed at 1. It can be viewed as a reduction of the MMKP, in which it is required that at most one item per group can be chosen. The GUBMKP problem is formulated as an integer linear program as shown in Figure 12.

Figure 8. An integer linear program of the multiple choice knapsack problem

$$
\begin{array}{lllr}
\text{maximize} & \sum_{i=1}^{n} \sum_{j=1}^{|G_i|} p_{ij} x_{ij} & & (12) \\
\\
\text{subject to} & & & \\
 & \sum_{i=1}^{n} \sum_{j=1}^{|G_i|} w_{ij} x_{ij} \leq c & & (13) \\
 & \sum_{j=1}^{|G_i|} x_{ij} = 1 & (i = 1, \ldots, n) & (14) \\
 & x_{ij} \in \{0, 1\} & (i = 1, \ldots, n), (j = 1, \ldots, |G_i|) & (15)
\end{array}
$$

Figure 9. An instance example of the multiple choice knapsack problem

Input:

$N \leftarrow 8 \;\; n \leftarrow 3$

$$\rightarrow \mathcal{P}rofits \rightarrow$$

$$p_{ij} \quad \underbrace{10\ 20\ 30}_{G_1} \quad \underbrace{40\ 50}_{G_2} \quad \underbrace{60\ 70\ 80}_{G_3}$$

$$\rightarrow \mathcal{W}eights \rightarrow$$

$$w_{ij} \quad \underbrace{5\ 20\ 25}_{G_1} \quad \underbrace{35\ 40}_{G_2} \quad \underbrace{45\ 55\ 60}_{G_3}$$

$$\rightarrow \mathcal{C}apacities \rightarrow$$

$$c \qquad 130$$

Constraints:

$$5x_{11} + 20x_{12} + 25x_{13} + 35x_{21} + 40x_{22} + 45x_{31} + 55x_{32} + 60x_{33} \leq 130$$
$$x_{11} + x_{12} + x_{13} = 1$$
$$x_{21} + x_{22} = 1$$
$$x_{31} + x_{32} + x_{33} = 1$$
$$x_{ij} \in \{0, 1\} \;\; (i = 1, 2, 3) \;\; (j = 1, \ldots, |G_i|)$$

Objective:

$$maximize \;\; 10x_{11} + 20x_{12} + 30x_{13} + 40x_{21} + 50x_{22} + 60x_{31} + 70x_{32} + 80x_{33}$$

Optimal solution:

$$\underbrace{0\ 1\ 0}_{G_1} \quad \underbrace{0\ 1}_{G_2} \quad \underbrace{0\ 0\ 1}_{G_3} \qquad value \;\; 150$$

Equation (20) represents the total profit to be maximized. Equation (21) ensures the knapsack capacities are not exceeded while Equation (22) ensures selecting at most one item from each of the n disjoint groups. Equation (23) is the binary selection requirement on decision variable x_{ij} such that x_{ij} is equal to 1 when the item j of the group G_i is selected, 0 otherwise.

Since the structure and the specification of the GUBMKP is similar to the MMKP, we use the same input of the latter to present an example of the GUBMKP (see Figure 13)

3.7 The multiple demand multidimensional multiple choice knapsack problem

We define the multiple demand multidimensional multiple choice knapsack problem (MDMMKP) as a combination of aspects of the multidimensional knapsack constraint, multiple demand con-

Figure 10. An integer linear program of the multidimensional multiple choice knapsack problem

$$
\begin{array}{llll}
\text{maximize} & \sum_{i=1}^{n} \sum_{j=1}^{|G_i|} p_{ij} x_{ij} & & (16) \\
 & & & \\
\text{subject to} & & & \\
 & \sum_{i=1}^{n} \sum_{j=1}^{|G_i|} w_{ij}^{k} x_{ij} \leq c^{k} & (k = 1, \ldots, m) & (17) \\
 & \sum_{j=1}^{|G_i|} x_{ij} = 1 & (i = 1, \ldots, n) & (18) \\
 & x_{ij} \in \{0, 1\} & (i = 1, \ldots, n), (j = 1, \ldots, |G_i|) & (19)
\end{array}
$$

Figure 11. An instance example of the multidimensional multiple choice knapsack problem

Input:

$N \leftarrow 8 \;\; n \leftarrow 3 \;\; m \leftarrow 2$

$$\rightarrow \mathcal{P}rofits \rightarrow$$

$$p_{ij} \quad \underbrace{10\ 20\ 30}_{G_1} \quad \underbrace{40\ 50}_{G_2} \quad \underbrace{60\ 70\ 80}_{G_3}$$

$$\rightarrow \mathcal{W}eights \rightarrow$$

$$
\begin{array}{lcccccccc}
w_{ij} & 5 & 20 & 25 & 35 & 40 & 45 & 55 & 60
\end{array}
$$

$$\underbrace{90\ 120\ 70}_{G_1} \quad \underbrace{110\ 90}_{G_2} \quad \underbrace{65\ 80\ 150}_{G_3}$$

$$\rightarrow \mathcal{C}apacities \rightarrow$$

$$c^{k} \qquad 130\ 230$$

Constraints:

$$
\begin{array}{l}
5x_{11} + 20x_{12} + 25x_{13} + 35x_{21} + 40x_{22} + 45x_{31} + 55x_{32} + 60x_{33} \leq 130 \\
90x_{11} + 120x_{12} + 70x_{13} + 110x_{21} + 90x_{22} + 65x_{31} + 80x_{32} + 150x_{33} \leq 230 \\
x_{11} + x_{12} + x_{13} = 1 \\
x_{21} + x_{22} = 1 \\
x_{31} + x_{32} + x_{33} = 1 \\
x_{ij} \in \{0, 1\} \;\; (i = 1, 2, 3) \;\; (j = 1, \ldots, |G_i|)
\end{array}
$$

Objective:

$$maximize \;\; 10x_{11} + 20x_{12} + 30x_{13} + 40x_{21} + 50x_{22} + 60x_{31} + 70x_{32} + 80x_{33}$$

Optimal solution:

$$\underbrace{0\ 0\ 1}_{G_1} \quad \underbrace{0\ 1}_{G_2} \quad \underbrace{1\ 0\ 0}_{G_3} \qquad value \;\; 140$$

straint and multiple choice constraint. It is considered as an extension of the multidimensional multiple choice knapsack problem (MMKP) in which there are *greater-than-or-equal-to* inequalities, in addition to the standard *less-than-or-equal-to* inequalities. The integer linear program of the MDMMKP can be stated in Figure 14.

The variable x_{ij} is equal to 1 when the item j of the group G_i is selected, 0 otherwise. The ob-

Figure 12. An integer linear program of the multidimensional knapsack problems with generalized upper bound constraints

$$
\begin{array}{llll}
\text{maximize} & \sum_{i=1}^{n}\sum_{j=1}^{|G_i|} p_{ij}x_{ij} & & (20) \\
\text{subject to} & & & \\
 & \sum_{i=1}^{n}\sum_{j=1}^{|G_i|} w_{ij}^{k}x_{ij} \leq c^{k} & (k = 1, \ldots, m) & (21) \\
 & \sum_{j=1}^{|G_i|} x_{ij} \leq 1 & (i = 1, \ldots, n) & (22) \\
 & x_{ij} \in \{0,1\} & (i = 1, \ldots, n), (j = 1, \ldots, |G_i|) & (23)
\end{array}
$$

Figure 13. An instance example of the multidimensional knapsack problems with generalized upper bound constraints

Constraints:

$$
\begin{array}{l}
5x_{11} + 20x_{12} + 25x_{13} + 35x_{21} + 40x_{22} + 45x_{31} + 55x_{32} + 60x_{33} \leq 130 \\
90x_{11} + 120x_{12} + 70x_{13} + 110x_{21} + 90x_{22} + 65x_{31} + 80x_{32} + 150x_{33} \leq 230 \\
x_{11} + x_{12} + x_{13} \leq 1 \\
x_{21} + x_{22} \leq 1 \\
x_{31} + x_{32} + x_{33} \leq 1 \\
x_{ij} \in \{0,1\} \;\; (i = 1,2,3) \;\; (j = 1, \ldots, |G_i|)
\end{array}
$$

Objective:

$$maximize \;\; 10x_{11} + 20x_{12} + 30x_{13} + 40x_{21} + 50x_{22} + 60x_{31} + 70x_{32} + 80x_{33}$$

Optimal solution:

$$\underbrace{0\ 0\ 1}_{G_1} \quad \underbrace{0\ 1}_{G_2} \quad \underbrace{1\ 0\ 0}_{G_3} \qquad value \;\; 140$$

Figure 14. An integer linear program of the multidimensional multiple demand multiple choice knapsack problem

$$
\begin{array}{llll}
\text{maximize} & \sum_{i=1}^{n}\sum_{j=1}^{|G_i|} p_{ij}x_{ij} & & (24) \\
\text{subject to} & & & \\
 & \sum_{i=1}^{n}\sum_{j=1}^{|G_i|} w_{ij}^{k}x_{ij} \leq c^{k} & (k = 1, \ldots, m) & (25) \\
 & \sum_{i=1}^{n}\sum_{j=1}^{|G_i|} w_{ij}^{k}x_{j} \geq c^{k} & (k = 1 + m, \ldots, (m + q)) & (26) \\
 & \sum_{j=1}^{|G_i|} x_{ij} = 1 & (i = 1, \ldots, n) & (27) \\
 & x_{ij} \in \{0,1\} & (i = 1, \ldots, n), (j = 1, \ldots, |G_i|) & (28)
\end{array}
$$

jective function equation (24) represents the total profit to be maximized. The knapsack constraints are presented in equation (25) and the demand constraints are presented in equation (26). Equation (27) represents the n multiple choice constraints.

There follows an illustrative example that represents an extension of the MMKP example with $q = 2$ demand constraints.

Figure 15. An instance example of the multidimensional multiple demand multiple choice knapsack problem

Input:

$N \leftarrow 8 \;\; n \leftarrow 3 \;\; m \leftarrow 2$

$$\rightarrow \mathcal{P}rofits \rightarrow$$

$$p_{ij} \quad \underbrace{10\;20\;30}_{G_1} \quad \underbrace{40\;50}_{G_2} \quad \underbrace{60\;70\;80}_{G_3}$$

$$\rightarrow \mathcal{W}eights \rightarrow$$

$$\begin{array}{cccccccccc}
w_{ij} & 5 & 20 & 25 & 35 & 40 & 45 & 55 & 60 \\
 & 90 & 120 & 70 & 110 & 90 & 65 & 80 & 150 \\
 & 5 & 20 & 100 & 35 & 60 & 45 & 50 & 60
\end{array}$$

$$\underbrace{90\;60\;70}_{G_1} \quad \underbrace{110\;90}_{G_2} \quad \underbrace{45\;20\;10}_{G_3}$$

$$\rightarrow \mathcal{C}apacities \rightarrow$$

$$c^k \quad 130\;230\;80\;200$$

Constraints:

$$5x_{11} + 20x_{12} + 25x_{13} + 35x_{21} + 40x_{22} + 45x_{31} + 55x_{32} + 60x_{33} \leq 130$$
$$90x_{11} + 120x_{12} + 70x_{13} + 110x_{21} + 90x_{22} + 65x_{31} + 80x_{32} + 150x_{33} \leq 230$$
$$5x_{11} + 20x_{12} + 25x_{13} + 35x_{21} + 40x_{22} + 45x_{31} + 55x_{32} + 60x_{33} \geq 80$$
$$90x_{11} + 60x_{12} + 70x_{13} + 110x_{21} + 90x_{22} + 45x_{31} + 20x_{32} + 10x_{33} \geq 200$$
$$x_{11} + x_{12} + x_{13} = 1$$
$$x_{21} + x_{22} = 1$$
$$x_{31} + x_{32} + x_{33} = 1$$
$$x_{ij} \in \{0, 1\} \;\; (i = 1, 2, 3) \;\; (j = 1, \ldots, |G_i|)$$

Objective:

$$maximize \;\; 10x_{11} + 20x_{12} + 30x_{13} + 40x_{21} + 50x_{22} + 60x_{31} + 70x_{32} + 80x_{33}$$

Optimal solution:

$$\underbrace{0\;\;0\;\;1}_{G_1} \quad \underbrace{0\;\;1}_{G_2} \quad \underbrace{1\;\;0\;\;0}_{G_3} \qquad value \;\; 140$$

3.8 Relation schema between problems

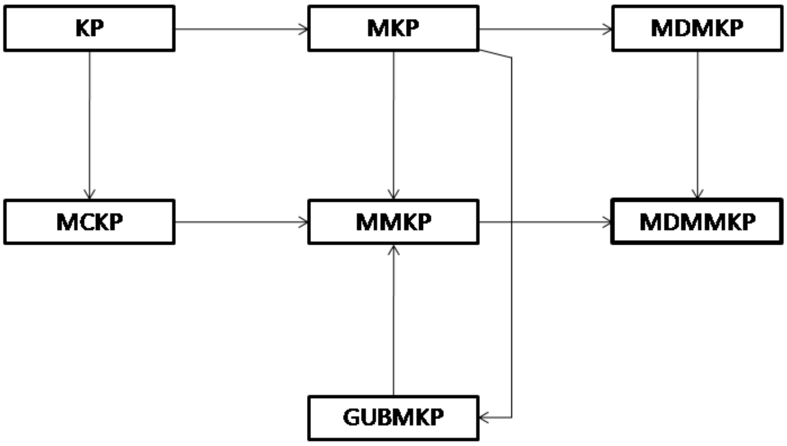
Figure 16 shows a set of possible generalizations between the problems mentioned above where each arrow represents a generalization between a target problem and a destination problem. In fact:

- The problems in which knapsack constraint is multidimensional are a generalization of problems with one dimension. Among these problems the MKP and the MMKP that are considered as a generalization of KP and MCKP respectively.
- The problems with demand constraints are a generalization of the problems without demand constraints. It is apparent that the MDMKP and the MDMMKP are considered as a general-

ization of the MKP MMKP respectively.

- The problems that have a notion of groups such as MCKP, MMKP and MDMMKP are a generalization of problems without groups such as KP, MKP and MDMKP respectively.
- The GUBMKP is a generalization of MKP and a reduction of MMKP.

Figure 16. Relation between knapsack problems

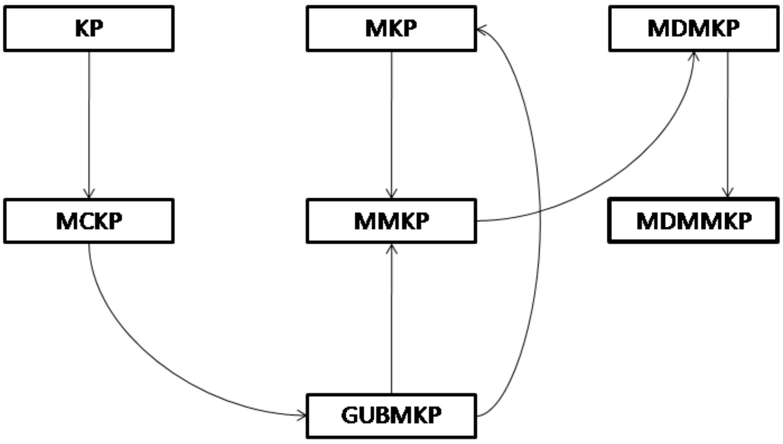


According to the transitivity characteristics of the generalization between problems we consider the MDMMKP as the most generalized problem.

4 Transformations between Integer Linear Programs

In this section, we present a set of transformations between the different integer linear programs of the knapsack problems mentioned above. These transformations are summarized in Figure 17. Each arrow indicates that the transformation between the linked problems is proved.

Figure 17. Transformation between ILPs of KPs



4.1 Transformation of the GUBMKP into the MMKP

The GUBMKP can be easily transformed into the MMKP. The formulation of the MMKP can be built by substituting the inequality ($\leq$) by the strict equality ($=$) for equation (22) in the GUBMKP formulation. Variables in each of the multiple choice constraints sum to 1 exactly. This modification is obtained by adding a dummy item into each group (GUB constraint) in which its consumption and its profit are zero. Indeed, selecting any item in the original GUBMKP formulation is similar to selecting the dummy item in the generated MMKP formulation. Therefore, the GUBMKP formulation mentioned above can be transformed into the MMKP formulation as shown in Figure 18.

Figure 18. An integer linear program of GUBMKP based on the MMKP formulation

$$\begin{array}{lll}
\text{maximize} & \sum_{i=1}^{n}\sum_{j=1}^{|G_i|+1} p_{ij}x_{ij} & & (29) \\
\text{subject to} & & & \\
& \sum_{i=1}^{n}\sum_{j=1}^{|G_i|+1} w_{ij}^{k}x_{ij} \leq c^{k} & (k=1,\ldots,m) & (30) \\
& \sum_{j=1}^{|G_i|+1} x_{ij} = 1 & (i=1,\ldots,n) & (31) \\
& x_{ij} \in \{0,1\} & (i=1,\ldots,n),(j=1,\ldots,|G_i|+1) & (32)
\end{array}$$

where $p_{i(|G_i|+1)} = 0$ and $w_{i(|G_i|+1)} = 0$ $(i = 1, \ldots, n)$.

The instance example of the GUBMKP (Figure 13) is moved to an MMKP instance (see Figure 19)

4.2 Transformation of the MKP into the MMKP

To explain this part let us take the ILP of the MKP mentioned above to show the transformation of the MKP into the MMKP. We can transform an MKP instance into an MMKP instance by the creation of a set of n groups each of which each contains two items. The first item represents an item of the items set of the MKP and the second item presents a dummy item whose weight and profit are zero. Then, each group G_j $(j = 1, \ldots, n)$ contain two items where:

$$p_{j1} = p_j,\ p_{j2} = 0,\ w_{j1}^{k} = w_j^{k},\ w_{j2}^{k} = 0\ (j = 1, \ldots, n)\ (k = 1, \ldots, m)$$

For example in case when the item is not selected in the original formulation of MKP, it is similar to select the dummy item belonging to the same set in the MMKP formulation.

The integer linear program of the MKP can be transformed into an integer linear program of the MMKP as shown in Figure 20.

The instance example of the MKP (Figure 5) is moved to an MMKP instance (see Figure 21)

Note that we will use the same principle to move the MDMKP into the MDMMKP and the KP into the MCKP.

4.3 Transformation of the GUBMKP into the MKP

In this part we are going to transform the GUBMKP into the MKP. Let us define the following terms $N = \sum_{i=1}^{n} |G_i|$, $N_h = \sum_{i=1}^{h} |G_i|$, $\forall h \in (1, \ldots, n)$ and $N_0 = 0$, and rename the following terms $x_{ij} = y_l$ and $w_{ij}^{k} = w_l^{k}$ where $l = N_{i-1} + j$.

Figure 19. An instance example of the GUBMKP based on the MMKP formulation

Input:

$N \leftarrow 11 \;\; n \leftarrow 3 \;\; m \leftarrow 2$

$$\rightarrow \mathcal{P}rofits \rightarrow$$

$$p_{ij} \quad \underbrace{10\;20\;30\;0}_{G_1} \quad \underbrace{40\;50\;0}_{G_2} \quad \underbrace{60\;70\;80\;0}_{G_3}$$

$$\rightarrow \mathcal{W}eights \rightarrow$$

$$w_{ij} \quad 5 \quad 20 \quad 25 \quad 0 \quad 35 \quad 40 \quad 0 \quad 45 \quad 55 \quad 60 \quad 0$$

$$\underbrace{90\;120\;70\;0}_{G_1} \quad \underbrace{110\;90\,0}_{G_2} \quad \underbrace{65\;80\;150\,0}_{G_3}$$

$$\rightarrow \mathcal{C}apacities \rightarrow$$

$$c^k \qquad 130\;230$$

Constraints:

$$5x_{11} + 20x_{12} + 25x_{13} + 35x_{21} + 40x_{22} + 45x_{31} + 55x_{32} + 60x_{33} \leq 130$$
$$90x_{11} + 120x_{12} + 70x_{13} + 110x_{21} + 90x_{22} + 65x_{31} + 80x_{32} + 150x_{33} \leq 230$$
$$x_{11} + x_{12} + x_{13} + x_{14} = 1$$
$$x_{21} + x_{22} + x_{23} = 1$$
$$x_{31} + x_{32} + x_{33} + x_{34} = 1$$
$$x_{ij} \in \{0,1\} \;\; (i = 1,2,3) \;\; (j = 1,\dots,|G_i|)$$

Objective:

$$maximize \;\; 10x_{11} + 20x_{12} + 30x_{13} + 40x_{21} + 50x_{22} + 60x_{31} + 70x_{32} + 80x_{33}$$

Optimal solution:

$$\underbrace{0\;0\;1\;0}_{G_1} \;\; \underbrace{0\;1\;0}_{G_2} \;\; \underbrace{1\;0\;0\;0}_{G_3} \quad value \quad 140$$

Figure 20. An integer linear program of MKP based on the MMKP formulation

$$\text{maximize} \quad \sum_{i=1}^{N} \sum_{j=1}^{2} p_{ij} x_{ij} \qquad (33)$$

subject to

$$\sum_{i=1}^{N} \sum_{j=1}^{2} w_{ij}^{k} x_{ij} \leq c^k \qquad ((k = 1,\dots,m) \qquad (34)$$
$$\sum_{j=1}^{2} x_{ij} = 1 \qquad (i = 1,\dots,N) \qquad (35)$$
$$x_{ij} \in \{0,1\} \qquad (i = 1,\dots,N), (j = 1,2) \qquad (36)$$

The idea in this part is to view each GUB constraint as a knapsack constraint. So we add to the classical m knapsack constraints a new n constraints by substituting the n GUB constraints to n equivalent knapsack constraints. In fact the ILP of the GUBMKP can be reformulated to an ILP of the MKP in this manner:

The n GUBMKP constraints $\sum_{j=1}^{|G_i|} x_{ij} \leq 1,\ \forall i \in (1,\dots,n)$ are transformed into $\sum_{l=1}^{N} w_l^k y_l \leq c^k, \forall k \in (m+1,\dots,m+n)$ where $c^k = 1\ \forall k \in (m+1,\dots,m+n)$ and for each $k+m$ knapsack constraint $\forall k \in (1,\dots,n)$

Figure 21. An instance example of the MKP based on the MMKP formulation

Input:

$N \leftarrow 16 \;\; m \leftarrow 2$

$$\rightarrow \mathcal{P}rofits \rightarrow$$

$$p_j \quad \underbrace{10\;0}_{G_1} \quad \underbrace{20\;0}_{G_2} \quad \underbrace{30\;0}_{G_3} \quad \underbrace{40\;0}_{G_4} \quad \underbrace{50\;0}_{G_5} \quad \underbrace{60\;0}_{G_6} \quad \underbrace{70\;0}_{G_7} \quad \underbrace{80\;0}_{G_8}$$

$$\rightarrow \mathcal{W}eights \rightarrow$$

$$w_j^k \quad 5 \;\; 0 \quad 20 \;\; 0 \quad 25 \;\; 0 \quad 35 \;\; 0 \quad 40 \;\; 0 \quad 45 \;\; 0 \quad 55 \;\; 0 \quad 60 \;\; 0$$

$$\underbrace{90\;0}_{G_1} \quad \underbrace{120\;0}_{G_2} \quad \underbrace{70\;0}_{G_3} \quad \underbrace{110\;0}_{G_4} \quad \underbrace{90\;0}_{G_5} \quad \underbrace{65\;0}_{G_6} \quad \underbrace{80\;0}_{G_7} \quad \underbrace{150\;0}_{G_8}$$

$$\rightarrow \mathcal{C}apacity \rightarrow$$

$$c^k \qquad 150\;300$$

Constraints:

$$5x_{11} + 20x_{21} + 25x_{31} + 35x_{41} + 40x_{51} + 45x_{61} + 55x_{71} + 60x_{81} \leq 150$$
$$90x_{11} + 120x_{21} + 70x_{31} + 110x_{41} + 90x_{51} + 65x_{61} + 80x_{71} + 150x_{81} \leq 300$$
$$x_{11} + x_{12} = 1$$
$$x_{21} + x_{22} = 1$$
$$x_{31} + x_{32} = 1$$
$$x_{41} + x_{42} = 1$$
$$x_{51} + x_{52} = 1$$
$$x_{61} + x_{62} = 1$$
$$x_{71} + x_{72} = 1$$
$$x_{81} + x_{82} = 1$$
$$x_{ij} \in \{0,1\} \;\; (i = 1, \ldots, 8) \;\; (j = \{1,2\})$$

Objective:

$$maximize \;\; 10x_{11} + 20x_{21} + 30x_{31} + 40x_{41} + 50x_{51} + 60x_{61} + 70x_{71} + 80x_{81}$$

Optimal solution:

$$\underbrace{0\;1}_{G_1} \quad \underbrace{0\;1}_{G_2} \quad \underbrace{1\;0}_{G_3} \quad \underbrace{0\;1}_{G_4} \quad \underbrace{0\;1}_{G_5} \quad \underbrace{0\;1}_{G_6} \quad \underbrace{1\;0}_{G_7} \quad \underbrace{1\;0}_{G_8} \quad value \quad 180$$

Figure 22. An integer linear program of MKP based on the MMKP formulation

$$\begin{array}{llll}
\text{maximize} & \sum_{l=1}^{N} p_l y_l & & (37) \\
 & & & \\
\text{subject to} & & & \\
 & \sum_{l=1}^{N} w_l^k y_l \leq c^k & (k = 1, \ldots, m+n) & (38) \\
 & y_l \in \{0,1\} & (\,(l = 1, \ldots, N) & (39)
\end{array}$$

$$w_l^{k+m} = \left\{ \begin{array}{l} 1\,, \forall l \in (1 + N_{k-1}, \ldots, N_k) \\ 0\text{, otherwise.} \end{array} \right\}$$

The instance example of the GUBMKP (Figure 13) is moved to an MKP instance (see Figure

23)

Figure 23. An instance example of the GUBMKP based on the MKP formulation

Input:

$N \leftarrow 8 \;\; m \leftarrow 5$

$\rightarrow \mathcal{P}rofits \rightarrow$

p_j	10	20	30	40	50	60	70	80

$\rightarrow \mathcal{W}eights \rightarrow$

w_j^k	5	20	25	35	40	45	55	60
	90	120	70	110	90	65	80	150
	1	1	1	0	0	0	0	0
	0	0	0	1	1	0	0	0
	0	0	0	0	0	1	1	1

$\rightarrow \mathcal{C}apacity \rightarrow$

c^k 150 300 1 1 1

Constraints:

$$5x_1 + 20x_2 + 25x_3 + 35x_4 + 40x_5 + 45x_6 + 55x_7 + 60x_8 \leq 150$$
$$90x_1 + 120x_2 + 70x_3 + 110x_4 + 90x_5 + 65x_6 + 80x_7 + 150x_8 \leq 300$$
$$x_1 + x_2 + x_3 \leq 1$$
$$x_4 + x_5 \leq 1$$
$$x_6 + x_7 + x_8 \leq 1$$
$$x_j \in \{0, 1\} \;\; (j = 1, \ldots, 8)$$

Objective:

$$maximize \;\; 10x_1 + 20x_2 + 30x_3 + 40x_4 + 50x_5 + 60x_6 + 70x_7 + 80x_8$$

Optimal solution:

x_j	0	0	1	0	1	1	0	0	value	140

4.4 Transformation of the MMKP into the MDMKP

The n multiple choice constraints of the MMKP can be subdivided into two kinds of constraints n *less-than-or-equal-to* and n *greater-than-or-equal-to inequalities*. Indeed, the MMKP is modeled as shown in Figure 24.

And using the same principle to transform the GUBMKP into the MKP mentioned above, we can view the MMKP program as an MDMKP program as shown in Figure 25.

where $c^k = 1 \forall k \in (m+1, \ldots, m+n, m+n+1, \ldots, m+2n)$.

The instance example of the MMKP (Figure 11) is moved to an MDMKP instance (see Figure 26)

4.5 Transformation of the MCKP into the GUBMKP

The MCKP can be transformed into an equivalent GUBMKP. The transformation is done by eliminating one of the items from each group. We mark this item j^* where x_{ij^*} is the variable with $w_{ij^*} = min\{w_{ij}^1 \; \forall j \in (1, \ldots, |G_i|)\}$, $\forall i \; \in (1, \ldots, n)$. To facilitate the removal of items,

Figure 24. A second integer linear program of MMKP

$$\begin{array}{lllr}
\text{maximize} & \sum_{i=1}^{n}\sum_{j=1}^{|G_i|} p_{ij}x_{ij} & & (40) \\
 & & & \\
\text{subject to} & & & \\
 & \sum_{i=1}^{n}\sum_{j=1}^{|G_i|} w_{ij}^{k}x_{ij} \leq c^{k} & ((k = 1, \ldots, m) & (41) \\
 & \sum_{j=1}^{|G_i|} x_{ij} \leq 1 & (i = 1, \ldots, n) & (42) \\
 & \sum_{j=1}^{|G_i|} x_{ij} \geq 1 & (i = 1, \ldots, n) & (43) \\
 & x_{ij} \in \{0, 1\} & (i = 1, \ldots, n), (j = 1, \ldots, |G_i|) & (44)
\end{array}$$

Figure 25. An integer linear program of MMKP based on the MDMKP formulation

$$\begin{array}{lllr}
\text{maximize} & \sum_{l=1}^{N} p_l y_l & & (45) \\
 & & & \\
\text{subject to} & & & \\
 & \sum_{l=1}^{N} w_l^{k} y_l \leq c^{k} & (k = 1, \ldots, m+n) & (46) \\
 & \sum_{l=1}^{N} w_l^{k} y_l \geq c^{k}, & (k = m+n+1, \ldots, m+2n) & (47) \\
 & y_l \in \{0, 1\} & (l = 1, \ldots, N) & (48)
\end{array}$$

we modify the index of items in order that the index j^* of each group G_i is equal to $= |G_i|$ $\forall i \in (1, \ldots, n)$. Then the MCKP can be modeled as shown in Figure 27.

where $\bar{v} = \sum_{i=1}^{n} p_{ij^*}$. This value presents a lower bound of the MCKP, $c' = c - \sum_{i=1}^{n} w_{ij^*}^{1}$, $p'_{ij} = p_{ij} - p_{ij^*}$ and $w'_{ij} = w_{ij}^{1} - w_{ij^*}^{1}$.

With this kind of modeling, we are sure that at most one item is selected by group. This item represents the item which has the smallest weight for each group marked by j^*. So if any item of group G_i is selected on the GUBMKP formulation, we are sure that the item j^* is selected.

The instance example of the MCKP (Figure 9) is moved to a GUBMKP instance (see Figure 28)

4.6 Algorithms of MDMMKP are able to solve the other problems

Given that any algorithm can solve the MDMMKP, based on the transitivity characteristics of the generalization and the set of transformations, this algorithm is able to solve all the other problems. In fact, we can easily transform their instances into the MDMMKP ones which can be solved directly by the considered algorithm.

5 Experimental results

This section is aimed at:

- validating experimentally the transformations between problems.
- answering the following questions: Are the generated problems using the transformations able to produce a reasonable CPU time value compared with the original formulations?

Figure 26. An instance example of the MMKP based on the MDMKP formulation

Input:

$N \leftarrow 8 \;\; m \leftarrow 5 \;\; q \leftarrow 3$

$$
\begin{array}{lcccccccc}
 & & & \multicolumn{4}{c}{\rightarrow \mathcal{P}rofits \rightarrow} & & \\
p_j & 10 & 20 & 30 & 40 & 50 & 60 & 70 & 80 \\
 & & & \multicolumn{4}{c}{\rightarrow \mathcal{W}eights \rightarrow} & & \\
w_j^k & 5 & 20 & 25 & 35 & 40 & 45 & 55 & 60 \\
 & 90 & 120 & 70 & 110 & 90 & 65 & 80 & 150 \\
 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
 & & & \multicolumn{4}{c}{\rightarrow \mathcal{C}apacity \rightarrow} & & \\
c^k & & & \multicolumn{6}{l}{150\ 300\ 1\ 1\ 1\ 1\ 1\ 1}
\end{array}
$$

Constraints:

$$
\begin{aligned}
&5x_1 + 20x_2 + 25x_3 + 35x_4 + 40x_5 + 45x_6 + 55x_7 + 60x_8 \leq 150 \\
&90x_1 + 120x_2 + 70x_3 + 110x_4 + 90x_5 + 65x_6 + 80x_7 + 150x_8 \leq 300 \\
&x_1 + x_2 + x_3 \leq 1 \\
&x_4 + x_5 \leq 1 \\
&x_6 + x_7 + x_8 \leq 1 \\
&x_1 + x_2 + x_3 \geq 1 \\
&x_4 + x_5 \geq 1 \\
&x_6 + x_7 + x_8 \geq 1 \\
&x_j \in \{0, 1\} \;\; (j = 1, \ldots, 8)
\end{aligned}
$$

Objective:

$maximize \;\; 10x_1 + 20x_2 + 30x_3 + 40x_4 + 50x_5 + 60x_6 + 70x_7 + 80x_8$

Optimal solution:

x_j 0 0 1 0 1 1 0 0 value 140

Figure 27. An integer linear program of MCKP based on the GUBMKP formulation

$$\text{maximize} \quad \bar{v} + \sum_{i=1}^{n} \sum_{j=1}^{|G_i|-1} p'_{ij} x_{ij} \tag{49}$$

subject to

$$\sum_{i=1}^{n} \sum_{j=1}^{|G_i|-1} w'_{ij} x_{ij} \leq c' \tag{50}$$

$$\sum_{j=1}^{|G_i|-1} x_{ij} \leq 1 \qquad (i = 1, \ldots, n) \tag{51}$$

$$x_{ij} \in \{0, 1\} \; (i = 1, \ldots, n) \qquad (j = 1, \ldots, |G_i| - 1) \tag{52}$$

Note that the computational platform used to solve the test problems is consisted as the CPLEX Solver version 9.0 on a Windows XP with 2.50 GHz and 2 GB of shared memory. We modeled the

Figure 28. An instance example of the MCKP based on the GUBMKP formulation

Input:

$N \leftarrow 8 \;\; n \leftarrow 3$

$$\rightarrow \mathcal{P}rofits \rightarrow$$

$$p_{ij} \quad \underbrace{10\ 20}_{G_1} \quad \underbrace{10}_{G_2} \qquad \underbrace{10\ 20}_{G_3}$$

$$\rightarrow \mathcal{W}eights \rightarrow$$

$$w_{ij} \quad \underbrace{15\ 20}_{G_1} \quad \underbrace{5}_{G_2} \qquad \underbrace{10\ 15}_{G_3}$$

$$\rightarrow \mathcal{C}apacities \rightarrow$$

$$c \quad 45 = 130 - (5 + 35 + 45)$$

Constraints:

$$15x_{11} + 20x_{12} + 5x_{21} + 10x_{31} + 15x_{32} \leq 45$$
$$x_{11} + x_{12} \leq 1$$
$$x_{21} \leq 1$$
$$x_{31} + x_{32} \leq 1$$
$$x_{ij} \in \{0,1\} \;\; (i = 1,2,3) \;\; (j = 1,\ldots,|G_i|)$$

Objective:

$$maximize \;\; 110 = (10 + 40 + 60) + 10x_{11} + 20x_{12} + 10x_{21} + 10x_{31} + 20x_{32}$$

Optimal solution:

$$\underbrace{1\ 0}_{G_1} \;\; \underbrace{1}_{G_2} \;\; \underbrace{0\ 1}_{G_3} \quad value \quad 150$$

MKP, MDKP, MMKP, GUBMKP and MDMMKP into CPLEX.

5.1 Instances details

To test the transformation mentioned in the last section we use a set of benchmarks available in OR-LIBRARY [1] maintained by Beasley. The proposed transformation of the MKP into the MMKP is tested on sets of MKP instances. Indeed, these instances are available at OR-LIBRARY and results have been published by [4]. The problem instances that we considered are summarized in Table 3, where each instance set *mknapcb* for $i = (1,\ldots,6)$ contains 30 instances. The headers Set, N and m_1 indicates respectively the name of the set, the number of items in each instance and the number of knapsack constraints.

Table 3. *Test MKP instances details*

Set	N	m_1
mknapcb1	100	5
mknapcb2	250	5
mknapcb3	500	5
mknapcb4	100	10
mknapcb5	250	10
mknapcb6	500	10

For the MDMKP, we use the benchmarks proposed by [2]. These instances are generated by properly modifying the MKP instances solved in [4]. Given an MKP instance with m_1 knapsack constraints, 6 MDMKP instances are generated, one for each combination of profits type (either positive or mixed) and number of constraints ($m_2 = 1$, $m_2 = m_1/2$ and $m_2 = m$ respectively).

We test the first six instances sets where each set *mdmkp_ct* for $i = (1, \dots, 6)$ contains 90 instances. The MDKP instances are reported in Table 4 where the headers Set, N, m_1 and m_2 indicates respectively the name of the set, the number of items in each set, the number of knapsack constraints, and the number of demand constraints.

Table 4. *Test MDMKP instances details*

Set	N	m_1	m_2
mdmkp_ct1	100	5	1 2 5
mdmkp_ct2	250	5	1 2 5
mdmkp_ct3	500	5	1 2 10
mdmkp_ct4	100	10	1 5 10
mdmkp_ct5	250	10	1 5 10
mdmkp_ct6	500	10	1 5 10

For the MMKP, the instances set is summarized in Table 5. The header Instance, n, n_i $\sum n_i$ and m_1 indicates respectively the name of the instance, the number of the groups, the number of the items of each group, the number of the total items, and the number of knapsack constraints. This instances set contains 13 instances (marked $I01, \dots, I13$) varying from small to large-scale size ones. These instances are given by [9].

Table 5. *Test MMKP instances details*

Instance	n	n_i	$\sum n_i$	m_1
I01	5	5	25	5
I02	10	5	50	5
I03	15	10	150	10
I04	20	10	200	10
I05	25	10	250	10
I06	30	10	300	10
I07	100	10	1000	10
I08	150	10	1500	10
I09	200	10	2000	10
I10	250	10	2500	10
I11	300	10	3000	10
I12	350	10	3500	10
I13	400	10	4000	10

Because the unavailability of the instances of MCKP and GUBMKP, we used the weakly correlated procedure proposed by [6] to generate instances.

For the GUBMKP, we range the instances into four sets and we vary the number of items between 2500 and 15000. For each set, we generate 30 instances and on total we generate 120 instances. The instances sets are reported in Table 6 in which the headers Set, n, n_i, m_1 and $\sum n_i$ respectively indicates the name of the sets, the number of the groups, the number of the items of each group, the number of the total items, and the number of knapsack constraints.

Table 7 shows the MCKP instances in which the headers Set, n, n_i and $\sum n_i$ respectively indicates the name of the set, the number of the groups, the number of the items of each group, and the number of the total items. Note that each set *mckp* for $i = (1, \dots, 3)$ contains 30 instances.

Table 6. *Test GUBMKP problem details*

Set	n	n_i	$\sum n_i$	m_1
gubmkp1	10	250	2500	5
gubmkp2	10	500	5000	5
gubmkp3	10	1000	10000	5
gubmkp4	10	1500	15000	5

Table 7. *Test MCKP instances details*

Set	n	n_i	$\sum n_i$
mckp1	1000	1000	1000000
mckp2	1500	1000	1500000
mckp3	2000	1000	2000000

5.2 Evaluation of the transformation

In this section, we give the results obtained by applying the different transformations between problems mentioned above. To compare between the generated problems with the original ones, we use the percentage of deviation marked *Dev* which is calculated by the following way $Dev = \frac{T2}{T1} \times 100$. *T1* and *T2* represent respectively the CPU time value of the solution of the classical formulation marked *model1* and the CPU time value of the solution of the generated problem based on the transformation marked *model2*. First of all, all the results obtained validate the transformation between problems and reach the same values of optimality. Also the generated problems are able to give a reasonable computing time. We report, in Table 8, the overall results obtained by both formulations over the all set problems. The columns marked m_{T1} and m_{T2} represent the average of computation time of *model1* and *model2* respectively, and *Dev* represents the percentage of deviation.

For the transformation of the GUBMKP into the MMKP, we notice that the generated formulations (*model2*) use less time than the original ones (*model1*) for all sets and the average of the *Dev* is equal to $84,1\%$.

Also, we notice the same thing for the transformation of the MDKP into the MDMMKP and the transformation of the MCKP into the GUBMKP which values of the average of the *Dev* are equal to $96,4\%$ and 89% respectively.

For the transformation of the MKP into the MMKP, we notice that all the values reached by the two formulations are very close and the average of the deviation of CPU time value *Dev* is equal to $99,5\%$.

For the transformation of the MMKP into the MDMKP, we remark that the generated formulations (*model2*) use less time than the original ones *model1* except the instance *I11*. We note that the average of the *Dev* for the whole instances is equal to $85,9\%$.

We do not test the transformation of the GUBMKP into the MKP because the similarity of the *lp file* of CPLEX for both formulations because CPLEX does not show the zero multipliers in the *lp file*.

6 Conclusion and future research direction

This paper presented a set of knapsack problems involving dimensions, demands and multiple choice constraints among which are the MKP, the MDMKP, the MCKP, the MMKP and the

Table 8. *Performances comparison of the transformation between different ILPs*

Set	m_{T1}	m_{T2}	Dev
GUBMKP→ MMKP			
gubmkp1	1,8	1,3	71,2%
gubmkp2	2,8	2,3	82,6%
gubmkp3	5,0	4,3	84,7%
gubmkp4	8,4	8,2	98,0%
MKP→ MMKP			
mknapcb1	3,0	2,9	98,7%
mknapcb2	129,0	126,1	97,7%
mknapcb3	1743,8	1753,2	100,5%
mknapcb4	23,9	24,0	100,1%
mknapcb5	3088,2	3078,6	99,7%
mknapcb6	1952,1	1958,8	100,3%
MDMKP→ MDMMKP			
mdmkp_ct1	48,3	48,2	99,9%
mdmkp_ct2	2437,4	2431,9	99,8%
mdmkp_ct3	2394,0	1939,1	81,0%
mdmkp_ct4	1414,5	1408,3	99,6%
mdmkp_ct5	2509,8	2501,5	99,7%
mdmkp_ct6	3282,7	3238,4	98,6%
MMKP→ MDMKP			
I1	0	0	–
I2	0	0	–
I3	2	2	100,0%
I4	54	26	48,1%
I5	0	0	–
I6	0	0	–
I7	4771	4739	99,3%
I8	6052	4933	81,5%
I9	4735	4287	90,5%
I10	5411	4736	87,5%
I11	4517	4544	100,6%
I12	5494	4904	89,3%
I13	6700	5096	76,1%
MCKP→ GUBMKP			
mckp1	21,8	21,0	96,0%
mckp2	43,5	32,2	73,9%
mckp3	44,8	43,5	97,1%

GUBMKP. Specifically, we defined the multiple demand multidimensional multiple choice knapsack problem (MDMMKP) as a generalization of these problems. Moreover, we applied a set of transformations between the different integer linear programs of knapsack extensions..

Using these transformations, we showed that any algorithm able to solve the generalized problem can definitely solve its related problems mentioned above. Computational results indicate that solving the new formulations using the transformations is able to generate reasonable computing time compared with the original ones.

Another interesting aspect of the current work is to show the importance of the transformations between formulations using heuristics. In fact, the transformations may very well prove to be useful in using heuristics already developed. For example the MKP is extensively studied in the literature comparing it with the GUBMKP, so it is very interesting to use methods of the MKP to solve the GUBMKP.

References

[1] J. Beasley. Or-library: distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41:1069–1072, 1990.

[2] P. Cappanera and M. Trubian. A local-search-based heuristic for the demand-constrained multidimensional knapsack problem. *INFORMS Journal on Computing*, 17(1):82–98, 2005.

[3] D.S. Chen, R.G. Batson, and Y. Dang. *Applied Integer Programming: Modeling and Solution*. Wiley, 2011.

[4] P.C. Chu and J.E. Beasley. A genetic algorithm for the multidimensional knapsack problem. *Journal of Heuristics*, 4:63–86, 1998.

[5] A. Freville. The multidimensional 0-1 knapsack problem: An overview. *European Journal of Operational Research*, 155(1):1–21, 2004.

[6] B. Han, J. Leblet, and G. Simon. Hard multidimensional multiple choice knapsack problems, an empirical study. *Computer and Operations Research*, 37:172–181, 2010.

[7] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, Berlin, Germany, 2004.

[8] S. Khan. *Quality adaptation in multi-session adaptative multimedia system: Model and architecture*. PhD thesis, Departement of Electronical and computer Engineering, University of Victoria, 1998.

[9] S. Khan, F. Kin, E. Manning, and M. Akabr. Solving the knapsack problem for adaptative multimedia system. *Studia Informatica - Special Issue on Combinatorial Problems*, 2:154–174, 2002.

[10] V.C. Li. Tight oscillation tabu search for multidimnesional knapsack problems with generalized upper bound constraints. *Computer and Operations Research*, 32:2843–2852, 2005.

[11] V.C. Li and G.L. Curry. Solving multidimensional knapsack problems with generalized upper bound constraints using critical event tabu search. *Computer and Operations Research*, 32:825–848, 2005.

[12] S. Martello and P. Toth. *Knapsack problems: Algorithms and computer implementations*. DEIS, University of Bologna, John Wiley & Sons Ltd, Baffins Lane, Chichester West Sussex - PO19 1UD, England, 1990.

[13] M. Moser, D.P. Jokanovic, and N. Shiratori. An algorithm for the multidimensional multiple-choice knapsack problem. *IEICE Transactions on Fundamentals of Electronics*, 80:582–589, 1997.

[14] D. Pisinger. A minimal algorithm for the multiple-choice knapsack problem. *European Journal of Operational Research*, 83(2):394–410, 1995.