4

The Knapsack Problem and Its Variants: Formulations and Solution Methods

Christophe Wilbaut, Saïd Hanafi, Igor Machado Coelho, and Abilio Lucena

4.1 Introduction

The knapsack problem asks for the selection of items from a set N with n elements, every item $j \in N$ with a non-negative integer *weight* w_j associated with it. A subset of items $S \subseteq N$ is called *feasible* when $\sum_{j \in S} w_j \leq c$

C. Wilbaut (✉) · S. Hanafi
LAMIH, CNRS, UMR 8201, University Polytechnique Hauts-de-France,
Valenciennes, France
e-mail: christophe.wilbaut@uphf.fr

S. Hanafi
e-mail: said.hanafi@uphf.fr

INSA Hauts-de-France, Valenciennes, France

I. M. Coelho
Programa de Pós-Graduação em Ciências Computacionais, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, Brazil
e-mail: igor.machado@ime.uerj.br; imcoelho@ic.uff.br

Instituto de Computação, Universidade Federal Fluminense, Niterói, Brazil

A. Lucena
Programa de Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil
e-mail: abiliolucena@cos.ufrj.br

S. Salhi and J. Boylan (eds.), *The Palgrave Handbook of Operations Research*,
https://doi.org/10.1007/978-3-030-96935-6_4

applies, where c is the *capacity* of a knapsack for carrying the items. Additionally, a profit is incurred by choosing a feasible subset of items and is measured through a properly defined function of N. The aim is to select a feasible $S \subseteq N$ which maximizes profit. The problem appears in distinct versions, each of them characterized by a different type of objective function. Variants to these versions, in turn, typically impose further restrictions on item selection. The linear and the quadratic objective function versions of the problem are those that attract the most interest, with the latter version lagging well behind the former.

The linear 0-1 Knapsack Problem (KP) maximizes a linear objective function. It associates a non-negative integer profit p_j with the selection of every $j \in N$ and relies on binary variables, $x \in \{0, 1\}^n$, for its standard formulation. Accordingly, $x_j = 1$ holds if j is selected and $x_j = 0$ applies otherwise. The Quadratic Knapsack Problem (QKP) is similarly formulated but its objective function involves cross profits $q_{ij} \in \mathbb{Z}$, $i, j \in N$, $i \leq j$, to account for eventual fringe benefits attained by simultaneously selecting items i and j (for simplicity, we assume that $q_{ii} = p_i$ holds for every $i \in N$). Additional versions of the problem will not be treated here since they only attract very marginal interest. The standard formulation of KP is given by:

$$\max \quad \sum_{j=1}^{n} p_j x_j \tag{4.1}$$

$$\text{subject to:} \quad \sum_{j=1}^{n} w_j x_j \leq c \tag{4.2}$$

$$x_j \in \{0, 1\} \qquad\qquad j \in N = \{1, \ldots, n\} \tag{4.3}$$

Furthermore, to avoid trivial solutions, conditions

$$\max\{w_j : j \in N\} < c < \sum_{j=1}^{n} w_j \tag{4.4}$$

are generally assumed. A formulation for QKP is obtained by noting that $x_i = x_i x_i$ holds for every $i \in N$ and replacing objective function (4.1) by

$$\max \sum_{i=1}^{n-1} \sum_{j=i}^{n} q_{ij} x_i x_j. \tag{4.5}$$

Given that the overwhelming majority of Knapsack Problem references are associated with KP, we will assume here that QKP and all its variants simply correspond to KP variants.

Historically, the first mention of KP dates back to 1896 [145], with references to the subject appearing on early studies in [41] (in folklore, the problem was probably studied under different names hundreds of years before). In the 1950s, the first dynamic programming (DP) approach for solving KP was introduced by Bellman in [13], and in the 1960s the branch-and-bound (B&B) paradigm brought additional solution alternatives to the problem. From that point on, KP has been intensively investigated, starting especially with the works in [73, 129] or [150]. Due to the number of knapsack variants and the associated contributions some books were dedicated to knapsack problems, such as [111, 143].

KP has a wide range of practical applications, such as: cargo loading, cutting stock, capital budgeting or project selection (see, e.g., [129, 150]). The problem can be solved by DP, which means it is not *strongly* $\mathcal{NP}$-Hard [70], only being limited by the integral magnitude of capacity c (the complexity of such a DP algorithm is $O(n \cdot c)$). A first survey on solution approaches for KP appears in [166].

A few variants of KP directly follow from the standard formulation of the problem. Among these we consider first the fractional or the continuous knapsack problem, FKP, where (0, 1) fractions of the items are also allowed to be selected. A formulation for FKP is obtained by simply replacing (4.3) in the standard formulation of KP by

$$x_j \in [0, 1] \quad \forall j \in N. \tag{4.6}$$

This allows greedy techniques to work, as suggested by Dantzig in [40]. Indeed it is well-known that FKP can be solved in $O(n \log n)$ time, by greedily selecting the items in *descending order* of the ratios $\frac{p_j}{w_j}$, $j \in N$.

It is also possible to allow an integral selection of multiple items, leading to the unbounded knapsack problem. From the standard KP formulation we just need to replace (4.3) by

$$x_j \in \mathbb{Z}_{\geq 0} \quad \forall j \in N, \tag{4.7}$$

in order to formulate the problem.

Finally, the mixed (linear) knapsack problem consists in a collection composed by $n_{\mathcal{D}}$ discrete and $n_{\mathcal{F}}$ fractional items (see, e.g., when $n_{\mathcal{F}} = 1$ [136]). This problem has several practical applications (see, e.g., [200]) and can be modeled in a generalized format, by allowing distinct lower and upper bounds for each item, namely l_j and u_j:

$$\max \sum_{j=1}^{n_{\mathcal{D}}+n_{\mathcal{F}}} p_j x_j \tag{4.8}$$

$$\text{s.t.}: \sum_{j=1}^{n_{\mathcal{D}}+n_{\mathcal{F}}} w_j x_j \leq c \tag{4.9}$$

$$l_j \leq x_j \leq u_j \qquad \forall j \in \{1, \ldots, n_{\mathcal{D}} + n_{\mathcal{F}}\} \tag{4.10}$$

$$x_j \text{ integer} \qquad \forall j \in \{1, \ldots, n_{\mathcal{D}}\} \tag{4.11}$$

$$x_j \in \mathbb{R} \qquad \forall j \in \{n_{\mathcal{D}} + 1, \ldots, n_{\mathcal{D}} + n_{\mathcal{F}}\} \tag{4.12}$$

In practice, instances of the KP variants we have just described involving hundreds of thousands of items are routinely solved by most of mixed-integer programming solvers. Among others, these solvers benefit from preprocessing techniques and the use of facet defining inequalities for the knapsack polytope, i.e., the convex hull of the feasible (integral) solutions to KP (see, e.g., [128] for details).

For the remaining of this chapter we consider several additional KP variants, classical or new. All of them can be formulated by changing the objective function or adding constraints to the standard KP formulation. In practice, these variants are generally much harder to solve than KP and while selecting them we focused on those associated with new interesting applications. First, in Sect. 4.2, we consider variants where a single knapsack constraint must be satisfied. Next, in Sect. 4.3, we discuss additional variants involving two or more knapsack constraints. We conclude the chapter in Sect. 4.4 by considering some future research directions on the topic.

4.2 Variants with a Single Knapsack Constraint

Several extensions of KP where only one knapsack constraint must be enforced can be identified in the literature. In this section we consider some of these variants where additional constraints conduct to some challenging problems.

4.2.1 The Multiple-Choice Knapsack Problem

The Multiple-Choice Knapsack Problem (MCKP) is a well-known variant of the KP in which the set of items is partitioned into classes (or groups). In this case the binary choice of taking an item or not in the knapsack is replaced by the selection of exactly one item out of each class of items. Let us denote by n^g the number of disjoint groups $N_1, \dots, N_{n^g}$ of items and by n_k the number of items in group N_k. Then, $N = \bigcup_{k=1}^{n^g} N_k$ and the total number of items is $n = \sum_{k=1}^{n^g} n_k$. As for KP, every item $j \in N_k$ is associated with a profit p_{kj} and a weight w_{kj} and MCKP aims at choosing exactly one item from each class such that the profit sum is maximized and the knapsack capacity c is respected. By defining variable $x_{kj} = 1$ if and only item j in class N_k is chosen, MCKP can be mathematically formulated as follows:

$$\max \quad \sum_{k=1}^{n^g} \sum_{j \in N_k} p_{kj} x_{kj} \tag{4.13}$$

$$\text{s.t.:} \quad \sum_{k=1}^{n^g} \sum_{j \in N_k} w_{kj} x_{kj} \leq c \tag{4.14}$$

$$\sum_{j \in N_k} x_{kj} = 1 \qquad\qquad k = 1, \dots, n^g \tag{4.15}$$

$$x_{kj} \in \{0, 1\}, \qquad\qquad k = 1, \dots, n^g,\ j \in N_k \tag{4.16}$$

Like KP it is always assumed that all coefficients p_{kj}, w_{kj} and c are non-negative integers. It is interesting to observe that some additional variants of KP may be derived directly from MCKP. For instance when $p_{kj} = w_{kj}$ for all $k = 1, \dots, n^g$ and for all $j \in N_k$ MCKP is called the multiple-choice subset-sum problem. In addition, the Multidimensional Multiple-Choice Knapsack Problem (MMCKP) can also be obtained by allowing into MCKP one or more properly defined additional knapsacks. Finally, the discounted 0-1 knapsack problem (see Sect. 4.2.2) can be viewed as a particular case of MCKP.

Several very efficient approaches were proposed to solve MCKP as those described in [51, 155, 173]. In particular, Pisinger developed in [155] an efficient dynamic programming-based algorithm using a list representation and a core problem. Classes are consecutively added to the core and reduction rules are also used to fix some of the variables. Kellerer et al. provided in their book [, Chap. 11] experiments to compare the performance of the approaches described in the last three articles cited above. The results demonstrated the efficiency of all the three approaches for all types of instances, except for strongly correlated ones for which only Pisinger's approach was able to provide results in the fixed time limit of one hour to solve 100 instances. However, these works showed that MCKP can be solved in very reasonable time for large instances.111

The situation is quite different for the multidimensional case. Some of the current best approaches dealing with MMCKP are based on hybrid approaches combining mathematical programming and heuristic strategies like in [32, 33] where authors considered the use of column generation techniques, whereas authors in [38] extended an iterative relaxation-based procedure proposed in [83]. The proposed method was more general and introduced what authors called the semi-continuous relaxation, where variables are constrained to take values close to zero or one. Mansi et al. proposed some variants of an hybrid strategy consisting of new cuts and a reformulation procedure used at each iteration to improve the iterative relaxation-based heuristic framework in [132]. Chen et Hao developed in [31] a "*reduce and solve*" heuristic combining problem reduction techniques (based on group-fixing and variable-fixing rules) with an optimal solution of the resulting integer linear problem. Another interesting work is presented in [183] where authors introduced a reformulation of MMCKP as a set partitioning problem allowing to reduce the decision variable dimension regarding the classical model to one dimension, even if the total number of variables and constraints remain the same. Experiments were conducted to compare this new model with the classical one when using CPLEX and GUROBI solvers. The results showed that the new model improved on average the objective function values and the required computational time to converge to the best solutions. In addition the use of this new formulation was competitive when compared to some state-of-the-art approaches for solving MMCKP. We also noted the work in [69] where authors proposed an iterative pseudo-gap enumeration approach which is a two-step procedure. In the first step, a family of pseudo-cuts is derived from the reduced cost constraints with regarding what authors called a "*pseudo-gap*". In the second step the original problem enriched by these pseudo cuts is solved with CPLEX. The pseudo-gap is used as a hypothesized gap between the upper bound and the lower bound of the original problem and authors proposed a strategy to enumerate it. When the pseudo-gap become valid the proposed iterative approach converges to an optimal solution. This approach obtained very good results compared with other state-of-the art methods for solving MMCKP. Mansini and Zanotti proposed very recently in [135] a new and very effective core-based exact approach. It solves subproblems of increasing size by means of a recursive variable-fixing process and stops when an optimality condition is satisfied. The proposed method obtained 10 new optimal solutions on open benchmark instances in the literature and improved several other best-known solutions.

MCKP and MMCKP have a wide range of classical applications including resources allocation [71, 173], capital budgeting [157], telecommunication networks [93], the management of resources in multimedia systems [113], the VLSI design [146] or the reliability of complex systems [177]. They also appear as a subproblem in several other diverse applications: in nurse scheduling [78], in generalized assignment problems [10] or in the context of warehouse optimization [11]. Fisher showed that MCKP appears when using Lagrangian relaxation of several integer programming problems [62], whereas Kellerer [110] proposed an approximation algorithm for a scheduling problem based on a relaxation formulated as MCKP.

Several additional applications dealing with MCKP or MMCKP can be found in the literature as in [179] where authors introduced what they called the "*selective* MCKP" in the context of QoS-aware service evaluation and selection. The formulation corresponds to the MCKP in which the choice constraints are relaxed as $\sum_{j \in N_k} x_{kj} \leq 1$. Zhong and Young used the MCKP to obtain optimal solutions of transportation programming problems when alternative versions of projects are under consideration in [202]. Authors provided six steps to build a MCKP and solved it with the method from [51]. Bae et al. considered in [9] the MCKP as a subproblem when solving a problem of finding a reconnaissance route of an unmanned combat vehicle in a terrain modeled as a grid. They proposed a two-phases method in which all feasible partial routes are generated in a first step. They showed that the problem of selecting partial routes from the set of candidates corresponds to a MCKP. Caserta and Voß solved the reliability Redundancy Allocation Problem (RAP) via the MCKP in [25]. They introduced the fact that the RAP corresponds to look for an optimal distribution of resources among subsystems. Then, they showed that RAP can be optimally solved in two steps: in the first one a multidimensional knapsack problem (see Sect. 4.3.1) associated to each subsystem is solved, whereas in the second a MCKP is solved to select the specific amount of resources to be assigned to each subsystem. Ykman-Couvreur et al. showed in [199] how the MPSoC runtime management can be modeled as MMCKP, whereas Fischer et al. considered MCKP in a classification problem in [61]. Zhao et al. considered in [201] the optimization of the quality of experience of multiple clients for video streaming over wireless network. They showed that the problem of finding an optimal allocation of bitrate level for multiple wireless clients can be modeled via a MCKP. Tisch et al. dealt with the configuration of learning factories as a MMCKP combined with a two-dimensional bin-packing problem in [178]. Al-Dulaimy et al. showed in [3] that the placement of virtual machines selected for migration in the context of energy efficient cloud data centers management can be modeled as a MCKP, whereas authors in [148] considered an inventory problem under uncertain demand in which a seller stocks an item in anticipation of a single selling season. The corresponding problem is non-convex and they showed that it can be solved via the solution of a polynomial number of particular MCKPs called "*two-sided*" MCKP in which both lower and upper bounds exist on the knapsack capacity. They proposed a DP-based algorithm and a heuristic and showed their methods obtained better results than a global optimizer for non-convex mixed-integer nonlinear problems. Diallo et al. introduced in [48] a model for a selective maintenance problem for serial systems and proposed a two-phase approach in which the problem is transformed into a MMCKP. Finally, Rogeau et al. introduced very recently what they called a "*coupling constraint*" into MMCKP to model a retrofit planning problem at territory scale, with a building-level resolution [161].

4.2.2 The Discounted Knapsack Problem

An interesting variant of the KP is the discounted 0-1 knapsack problem (DKP), where items are selected from 3-sized item groups [89, 163]. Although much harder than KP, fully polynomial approximation schemes and DP-based solution approaches have been developed for it.

Given n^g item groups having three items each, and one knapsack with capacity c, where the items in the i-th ($i = 0, 1, \cdots, n^g - 1$) group are denoted as $3i, 3i + 1$ and $3i + 2$, with value coefficients p_{3i}, p_{3i+1} and $p_{3i+2} = p_{3i} + p_{3i+1}$, and weights w_{3i}, w_{3i+1} and w_{3i+2} (discounted weight), $w_{3i} + w_{3i+1} > w_{3i+2}$, $w_{3i+2} > w_{3i}$, and $w_{3i+2} > w_{3i+1}$, the goal is to maximize the profits by selecting at most one item in a group. We assume profits and weights to be positive integers, so as the capacity c. DKP can be formulated by (4.17)–(4.20):

$$\max \quad \sum_{i=0}^{n^g-1} (p_{3i}x_{3i} + p_{3i+1}x_{3i+1} + p_{3i+2}x_{3i+2}) \tag{4.17}$$

$$\text{s.t.}: \quad \sum_{i=0}^{n^g-1} (w_{3i}x_{3i} + w_{3i+1}x_{3i+1} + w_{3i+2}x_{3i+2}) \leq c \tag{4.18}$$

$$x_{3i} + x_{3i+1} + x_{3i+2} \leq 1 \qquad \forall i \in \{0, \ldots, n^g - 1\} \tag{4.19}$$

$$x_{3i}, x_{3i+1}, x_{3i+2} \in \{0, 1\} \qquad \forall i \in \{0, l \ldots, n^g - 1\} \tag{4.20}$$

DKP has been introduced in the Master's Thesis by Guldan [80] and it can be viewed as a particular case of MCKP where the number of items in each group is set to three. Specific relations between items in a group are defined in DKP, which is not the case in MCKP.

DP-based algorithms were proposed to deal with DKP in [80, 89, 163]. Experiments reported in these works show that DP can be used to solve instances with up to $n^g = 1,000$ groups in reasonable time under the condition of having sufficient memory resources. A fully polynomial-time approximation scheme, a 2-approximation algorithm and a particle swarm optimization heuristic were also proposed in [89]. Several other population-based metaheuristics have then been proposed to deal with DKP: genetic algorithm [87], differential evolution algorithms [203], multi-strategy monarch butterfly optimization heuristic [58], moth search [56]. Recently, He et al. proposed a new kind of method to design an evolutionary algorithm by using algebraic theory in [86]. The approach used two evolution operators using the addition, multiplication and inverse operation of the direct product of rings. Wu et al. proposed in [195] a discrete hybrid teaching-learning-based optimization algorithm. Very recently, Wilbaut et al. proposed to combine Variable Neighborhood Search (VNS) with DP and showed that all the instances from the literature can be solved in less than 2 seconds thanks to reduction procedures [192].

4.2.3 The Knapsack Problem with Setup

Another feature which is common to several distinct KP variants is the presence of setup costs [27]. The setup knapsack problem (SKP) originates from (and is named after) a particular machine scheduling problem which involves setup costs [49]. SKP asks for the selection of items from families $\mathcal{F}$, where each item family needs to be set up before usage. Such a condition is modeled through additional variables z_f, which indicate whether or not family f corresponding to item $j \in \mathcal{N}_f$ has been set up [27]. A formulation for SKP is given by:

$$\max \quad \sum_{f\in\mathcal{F}} (u_f z_f + \sum_{j\in\mathcal{N}_f} p_j x_j) \tag{4.21}$$

$$\text{s.t.} \quad \sum_{f\in\mathcal{F}} (s_f z_f + \sum_{j\in\mathcal{N}_f} w_j x_j) \leq c \tag{4.22}$$

$$z_f - x_j \geq 0 \qquad \forall f \in \mathcal{F}, j \in \mathcal{N}_f \tag{4.23}$$

$$z_f - \sum_{j\in\mathcal{N}_f} x_j \leq 0 \qquad \forall f \in \mathcal{F} \tag{4.24}$$

$$z_f, x_j \in \{0, 1\} \qquad \forall f \in \mathcal{F}, j \in \mathcal{N}_f \tag{4.25}$$

Note that, in this case, we have $u_f, p_j \in \mathbb{R}$, $s_f, w_j \in \mathbb{Z}^+$ and $N = \bigcup_{f\in\mathcal{F}} \mathcal{N}_f$, with $\mathcal{N}_{f_1} \cap \mathcal{N}_{f_2} = \emptyset$, $\forall f_1, f_2 \in \mathcal{F}$, $f_1 \neq f_2$. Since KP can be reduced to SKP, it is also $\mathcal{NP}$-Hard [70].

A similar SKP variant is called the Multiple-class Integer Knapsack problem with Setups (MIKS), where items are organized in distinct classes with setups and multiplicity constraints [147]. From this perspective, the weight of item at a given class is a multiple of the class weight, while also allowing multiple items to be selected within lower and upper class limits. A more constrained version of MIKS eliminates multiplicity constraints and allows whole integer interval, being called Integer Knapsack Problem with Setups (IKS), that allows the development of more efficient algorithms. In this sense, class weights are restricted to negative factors. A further relaxation of IKS yields the Continuous Knapsack Problem with Setups (CKS), and further adaptations can generate the classic binary knapsack problem. CKS appears as subproblem for capacitated multi-item lot sizing problems, and IKS as subproblem for cutting stock problems. An interesting point is that the relaxation of the binary knapsack with multiple choices is equivalent to CKS [147]. When dealing with numerical experimentation, most success is achieved by hybrid approaches that involve B&B and DP.

Regarding the SKP Khemakhem and Chebil proposed in [114] a tree-search-based combination heuristic which is based on the Filter-and-Fan method. The approach is an iterative local search in which moves are generated according to a tree-search strategy. The main particularity of the method lies in a new technique that makes a bijection between a solution of SKP and an integer index. The algorithm obtains better average results as CPLEX. The same authors proposed in [28] two DP-based algorithms for SKP. The first one considers a stage associated with every item in the problem and computes the optimal values of two auxiliary problems. In the second one authors introduces a space reduction technique to exploit the bijection mentioned above. The DP algorithm considers only the setup variables and a complete solution is obtained by solving a particular KP. This approach is able to solve large correlated instances with up to 10,000 items in reasonable CPU time.

4.2.4 The Knapsack Problem with Neighbor Constraints

For the 1-Neighbor Knapsack Problem (1-NKP), an item can be selected only if at least one of its neighbours is also selected while items with no neighbors can always be selected. Contrary to that, for the all-neighbours knapsack problem (All-NKP), an item can be selected only if all its neighbors are also selected. Approximation algorithms and hardness results for 1-NKP and All-NKP are provided in [16, 17]. These two NKPs can be generalized as described by the following 0-1 linear program NKP$_k$:

$$\max \quad \sum_{i \in N} p_i x_i \tag{4.26}$$

$$\text{s.t.:} \quad \sum_{i \in N} w_i x_i \leq c \tag{4.27}$$

$$\min(k, n_i) x_i \leq \sum_{j \in N_i} x_j \qquad \forall i \in N, N_i \neq \emptyset \tag{4.28}$$

$$x_i \in \{0, 1\} \qquad \forall i \in N \tag{4.29}$$

where N_i is the set of neighbors of vertex i in the graph G and k is an integer less or equal to the maximum degree of vertices in G, i.e. $k \leq \Delta = \max\{|N_i| : i \in N\}$. Note that NKP$_1$ is equivalent to 1-NKP and NKP$_\Delta$ is equivalent to All-NKP, while NKP$_0$ corresponds to KP.

The Subset-Union Knapsack Problem (SUKP) is a special case of All-NKP in which the input graph is directed and bipartite and such that the vertex set is partitioned into vertices associated with *items* and vertices associated with *elements* and every arc points from an *item vertex* to an *element vertex*. The precedence-constrained knapsack problem [15] and the partially ordered knapsack problem [118] are also special cases of All-NKP on directed graph. The general, undirected 1-neighbor knapsack problem generalizes several maximum coverage problems including the budgeted variant considered in [116].

4.2.5 The Knapsack Constrained Maximum Spanning Tree Problem

The Knapsack Constrained Maximum Spanning Tree problem (KCMST) is a combination of the knapsack problem and the maximum spanning tree problem, introduced in [197]. The weight-constrained Minimum Spanning Tree (MST) problem is to find a spanning tree of total edge weights at most a given capacity and minimum total edge costs. A spanning tree of a graph G is a subgraph connecting all vertices of G. For a spanning tree T of G, we define its profit $p(T)$ and weight $w(T)$, respectively, as the sum of the profits and weights of its constituent edges. A spanning tree T is feasible for KCMST if it satisfies $w(T) \leq c$. The problem is to fill the knapsack with a feasible spanning tree such that the tree profit is maximized, which can be formulated as follows:

$$\max \quad p(T) \tag{4.30}$$

$$\text{s.t.:} \quad w(T) \leq c \tag{4.31}$$

$$T \text{ is a spanning tree of } G \tag{4.32}$$

Based on the connectivity requirements of a tree, four distinct IP formulations for MST are discussed in [55], namely subtour elimination, cut set, single-commodity flow and Martin's formulation. A subtour elimination formulation is based on the fact that a tree T has no cycles and contains $n-1$ edges [53] and is described as follows:

$$\max \quad \sum_{e\in E} p_e x_e \tag{4.33}$$

$$\text{s.t.:} \quad \sum_{e\in E} w_e x_e \leq c \tag{4.34}$$

$$\sum_{e\in E} x_e = n - 1 \tag{4.35}$$

$$\sum_{e\in E(S)} x_e \leq |S| - 1 \qquad \forall S \subset N \tag{4.36}$$

$$x_e \in \{0, 1\} \qquad e \in E \tag{4.37}$$

Martin's formulation (see [35, 144]) uses binary variables y^k_{ji} if j is the father of i in the tree structure obtained by rooting T at k, 0 otherwise:

$$\max \quad \sum_{e \in E} p_e x_e \tag{4.38}$$

$$\text{s.t.:} \quad \sum_{e \in E} w_e x_e \le c \tag{4.39}$$

$$\sum_{e \in E} x_e = n - 1 \tag{4.40}$$

$$x_{ij} = y^k_{ij} + y^k_{ji} \qquad \forall (i, j) \in E, k \in N \tag{4.41}$$

$$\sum_{j \in N} y^k_{ij} = 1 \qquad \forall i \neq k \in N \tag{4.42}$$

$$y^k_{kj} = y^k_{ii} = 0 \qquad \forall (i, j) \in E, k \in N \tag{4.43}$$

$$x_e \in \{0, 1\} \qquad e \in E \tag{4.44}$$

The KCMST problem can be extended directly to generalized variants of KP where the MST is replaced by Generalized Maximum Spanning Tree. Pop presented in [158] a survey on different integer programming formulations of the generalized MST problem.

4.2.6 The Set Union Knapsack Problem

In the Set-Union knapsack Problem (SUKP), introduced in [77], we are given a set N of n elements and a set M of m items. Each item $i \in M$ is characterized by its profit $p_i > 0$ and by its set of associated elements $N_i \subseteq N$. For any subset $I \subseteq M$, the profit of I is given by $\sum\limits_{i \in I} p_i$, while the weight of I is $\sum\limits_{j \in \cup_{i \in I} N_i} w_j$, where w_j is the weight of element j. The objective of the SUKP is to select a maximum profit subset of M, while respecting the capacity $c > 0$ of the knapsack. A generic set based formulation of SUKP can be expressed as follows:

$$\max \quad p(I) \tag{4.45}$$

$$\text{s.t.:} \quad w(\cup_{i \in I} N_i) \leq c \tag{4.46}$$

$$I \subseteq N \tag{4.47}$$

The graph $G = (M \cup N, E)$, where an edge $(i, j) \in E$ only for $i \in M$ and $j \in N_i$, is a bipartite graph (also known as a bigraph) and sets M and N form a partition of the vertex set. It may be observed that SUKP includes KP as a special case when $N_i \cap N_{i}^{'} = \emptyset \ \forall i, i^{'} \in M$ and $|N_i| = 1 \ \forall i \in M$. Since KP is $\mathcal{NP}$-hard, SUKP is $\mathcal{NP}$-hard as well. SUKP has various domain-specific applications including information security systems, financial decision-making [7, 111], flexible manufacturing machine [77, 116], database partitioning [76], smart city [180] and data stream compression [198]. A mixed-integer programming model can be derived from the previous set-based formulation. It is based on two sets of binary variables. The first one is associated with items whereas the second one is associated with elements. More precisely, we define binary variables x_i and y_j for each item $i \in M$ and for each element $j \in N$, respectively, as:

$$x_i = \begin{cases} 1 \text{ if item } i \text{ is selected} \\ 0 \text{ otherwise} \end{cases}$$

$$y_j = \begin{cases} 1 \text{ if element } j \text{ is selected} \\ 0 \text{ otherwise.} \end{cases}$$

Using these variables SUKP can be formulated as:

$$\max \quad \sum_{i \in M} p_i x_i \tag{4.48}$$

$$\text{s.t.:} \quad \sum_{j \in N} w_j y_j \leq c \tag{4.49}$$

$$x_i \leq y_j \qquad \forall i \in M, \forall j \in N \tag{4.50}$$

$$x_i, y_j \in \{0, 1\} \qquad \forall i \in M, \forall j \in N \tag{4.51}$$

The objective (4.48) maximizes the total profit associated with the selected items whereas constraint (4.49) is the knapsack constraint. Constraints (4.50) imply that the selection of item i forces the selection of all elements in N_i.

Several algorithms have been proposed to solve SUKP. Goldschmidt et al. developed in [77] an exact DP algorithm for SUKP and derived sufficient conditions for having a polynomial run time. Arulsevan showed in [7] that a greedy algorithm for the budgeted maximum coverage problem approximates SUKP within a constant factor. Following the same research direction, Taylor proposed in [175] an approximation algorithm relying on an algorithm for the densest k-subhypergraph problem. Other recent papers dealing with SUKP can be found in the literature. He et al. developed a binary artificial bee colony algorithm and used greedy repairing and optimization to handle infeasibility of a solution in [88], whereas authors in [57] proposed several versions of discrete moth search, while a hybrid approach that combines a genetic algorithm and particle swarm optimization was developed in [151]. Wei and Hao proposed in [184] an iterated two-phase local search (I2PLS) approach for solving SUKP, whereas the same authors proposed in [186] a multistart solution-based tabu search and in [185] a kernel-based tabu search. The last two algorithms conduct to the current best-known solutions for several benchmark instances used in the literature.

4.2.7 The Precedence Constrained Knapsack Problem

The Precedence Constraint Knapsack Problem (PCKP), also called partially ordered knapsack problem, is a generalization of KP which includes a partial order on items described by an acyclic directed graph $G = (N, A)$ with weights and profits associated to its vertices. An arc $(i, j) \in A$ means that "*item i precedes item j*", and implies that if item j is selected then item i must be also selected.

Formally, PCKP can be described by the following 0-1 integer linear program:

$$\max \quad \sum_{i \in N} p_i x_i \tag{4.52}$$

$$\text{s.t.:} \quad \sum_{i \in N} w_i x_i \le c \tag{4.53}$$

$$x_i \ge x_j \qquad\qquad (i, j) \in A \tag{4.54}$$

$$x_i \in \{0, 1\} \qquad i \in N \tag{4.55}$$

PCKP appears in many applications as network design, investment, management problems, scheduling, production planning, see, e.g., [50, 103, 171, 174].

Johnson and Niemi were the first to propose DP algorithms when G is out-tree, called left-right approach in [106]. Then, Samphaiboon and Yamada presented heuristics and DP algorithms for general acyclic graph G in [167].

Beyond the wide range of applications available for PCKP, precedence constraints are of theoretical and computational interest because by themselves they define a polyhedron with integer vertices. More precisely, the convex hull of feasible integer points is the same as the region obtained by relaxing the integrality restrictions. As a consequence, this property allows to solve the Lagrangian relaxation of PCKP associated with the knapsack constraint in polynomial time. Several researchers investigated the polyhedral structure of PCKP by exploiting this property and the rich and large existing results on the knapsack polytope, see e.g., [19, 152].

4.2.8 The Disjunctively Constrained Knapsack Problem

The knapsack problem with conflicts, also known as the Disjunctively Constrained Knapsack Problem (DCKP), was introduced in [196]. Given a conflict graph $G = (N, E)$ describing incompatibilities between items, DCKP consists in determining a maximum profit set of compatible items to be packed into the knapsack. A compact integer linear programming formulation for DCKP makes use of a set of binary variables x_i associated with every item $i \in N$ and taking value 1 if item i is packed in the knapsack, 0 otherwise, and can be stated as:

$$\max \quad \sum_{i \in N} p_i x_i \tag{4.56}$$

$$\text{s.t.:} \quad \sum_{i \in N} w_i x_i \leq c \tag{4.57}$$

$$x_i + x_j \leq 1 \qquad (i, j) \in E \tag{4.58}$$

$$x_i \in \{0, 1\} \qquad i \in N \tag{4.59}$$

Additional DCKP formulations are found in [92] where an aggregated formulation is proposed, and in [14] where the authors suggest an equivalent MIP based on the determination of a family of cliques in the conflict graph G. Pferschy and Schauer showed in [153] that DCKP is strongly $\mathcal{NP}$-hard and presented pseudo-polynomial algorithms for graphs of bounded tree width and chordal graphs. DCKP is also known as KP with conflicts and is a combination of the Maximum Weighted Independent Set Problem (MWISP) and the KP. When no conflicts are considered, i.e., $E = \emptyset$, the problem reduces to KP whereas when the knapsack constraint is omitted the problem becomes the MWISP.

Several exact and heuristic approaches are considered in the literature. We note in particular the B&B algorithms proposed in [14, 36, 165, 168, 196]. From a heuristic perspective Yamada et al. proposed a local search algorithm [196], whereas more sophisticated approaches based on Tabu Search (TS) were developed in [91] and [168]. The local branching framework was also considered in [2]. Other metaheuristics were considered as scatter search in [94] or the guided neighborhood search in [95], probabilistic tabu search in [164]. Finally we note the memetic algorithms proposed in [164] and very recently in [187].

4.2.9 The Product Knapsack Problem

The Product Knapsack Problem (PKP) is one of the most recent KP variants. It was introduced in [52] and is defined as we describe next. It aims at determining a subset of items $S \subseteq N$ that maximizes $\prod_{j\in S} p_j$ and such that $\sum\limits_{j\in S} w_j \le c$ holds. With the usual binary variable x_j equal to 1 iff item j is selected, a mixed-integer nonlinear formulation of PKP is described as follows:

$$\max \prod_{j\in N} \max\{1, p_j x_j\} = \prod_{j\in N} (1 + (p_j - 1)x_j) \tag{4.60}$$

$$\text{s.t.:} \sum_{j\in N} w_j x_j \le c \tag{4.61}$$

$$x \in \{0, 1\}^n \tag{4.62}$$

Denoting by $N^- = \{j \in N : p_j < 0\}$ the sets of items with negative profits and based on the observation that an optimal solution to PKP must include an even number of items $j \in N^-$, D'Ambrosio et al. proposed the following mixed-integer linear formulation for the problem:

$$\max \quad \sum_{j\in N} \log(|p_j|x_j) \tag{4.63}$$

$$\text{s.t.:} \quad \sum_{j\in N} w_j x_j \le c \tag{4.64}$$

$$\sum_{j\in N^-} w_j x_j = 2y \tag{4.65}$$

$$(x, y) \in \{0,1\}^n \times \mathbb{Z}_+ \tag{4.66}$$

As for solution approaches to PKP, D'Ambrosio et al. proposed a DP algorithm and a mixed-integer linear programming one. Pferschy et al. showed in [154] that PKP is weakly $\mathcal{NP}$-Hard and proposed a fully polynomial-time approximation scheme for the problem.

4.3 Variants with Multiple Knapsack Constraints

Some KP variants involve more than one knapsack constraints, a requirement that is straightforward to implement directly over the KP formulation. Additionally, they may eventually combine that feature with, among others, different types of objective functions. In this section we consider a few KP variants that fall in such a group, starting with the simplest (from the formulation point of view) of them, the multidimensional knapsack problem.

4.3.1 The Multidimensional Knapsack Problem

The 0-1 Multidimensional Knapsack Problem (MKP) is the most natural generalization of the KP in which several capacity constraints have to be satisfied. MKP is defined for a set $M = \{1, \ldots, m\}$ of $m > 1$ capacity constraints. Denoting by w_{ij} the weight of item $j \in N$ in constraint $i \in M$ and by c_i the capacity of knapsack $i \in M$, MKP can be formulated as follows:

$$\max \sum_{j=1}^{n} p_j x_j \tag{4.67}$$

$$\text{s.t.} : \sum_{j=1}^{n} w_{ij} x_j \leq c_i \qquad i \in M \tag{4.68}$$

$$x_j \in \{0, 1\} \qquad j \in N \tag{4.69}$$

As for KP it is assumed that all coefficients p_j, w_{ij} and c_i are non-negative integers values. Some weights can be equal to 0, i.e. $w_{ij} = 0$, as long as $\sum_{i=1}^{m} w_{ij} \geq 1$ holds for all items $j \in N$. In addition, it is generally assumed that $w_{ij} \leq c_i, \forall j \in N, \forall i \in M$ to guarantee that every item can be selected. Finally, in the same way as (4.4) for KP it is assumed that $\sum_{j=1}^{n} w_{ij} \geq c_i, \forall i \in M$. It can be observed that MKP is also known under other names such as m-dimensional KP or multi-constraint KP.

As mentioned in previous sections, many applications of KP and its variants can be listed in the selection and packing area. In most of the cases, real-world applications require more than one constraint. Thus MKP can be considered to deal with these practical problems. In addition, MKP is a particular case of the general 0-1 integer programming problem. The first references dedicated to the MKP include [129, 137, 188]. Cutting stock [73] and loading problems [13] are other classical applications. The MKP was also used in [182] to model the daily management of a satellite like SPOT. In the computer science context, MKP enables to model the resource allocation in distributed data processing [71] and the planning of data-processing programs [176]. Other recent applications can be listed. For instance Mansini and Speranza showed in [133] how the problem of optimally selecting the assets for leasing can be modeled as a MKP. Holte pointed out an equivalence between a multi-unit combinatorial auction and a MKP in [97]. Kelly considered in [112] a variety of auction winner determination problems and their corresponding family of KPs. He also highlighted the fact that existing algorithms dedicated for these problems can be applied effectively to solve the corresponding auction winner determination problems. Finally, Chen et al. considered a stochastic variant of MKP as an allocation resource problem in which the demand may change over time in [30].

Theoretical works demonstrated that MKP is $\mathcal{NP}$-hard and that it has no fully polynomial-time approximation scheme unless $\mathcal{P} = \mathcal{NP}$ (see, e.g., [72]). In fact, despite its simple formulation, the problem remains very difficult to solve at optimality, especially when the number of constraints increases. This is undoubtedly what explains the success of MKP, which is often used to calibrate and evaluate new approaches, in particular metaheuristics and hybrid methods. However, one can observe that the number of papers dedicated to exact methods is comparatively small. It is not possible to present a summary of all interesting publications dealing with the MKP in a few pages. We restrict our presentation to some of the recent approaches leading to the current best-known solutions over the instances of the literature. Interested readers can find surveys dedicated specifically to MKP, see, e.g., [64, 119, 159, 190]. We do not consider either in this section the particular case when $m = 2$ for which many references can also be found in the literature.

Efficient recent approaches dealing with the MKP concern mainly metaheuristics and hybrid methods combining mathematical programming with specific components dedicated to the problem. One of the current best approach to solve hard MKP instances is described in [18]. Authors proposed a multi-level search strategy based on the reduced costs of the non-basic variables of the LP-relaxation solution. In their approach the top-level branches of the search tree were enumerated following resolution search [34], the middle-level branches were enumerated using a B&B and the lower-level branches were enumerated with a depth first search enumeration strategy. Mansini and Speranza proposed in [134] another approach using similar concepts: the exact solving of a sequence of sub-problems, each with a different cardinality constraint, and the use of a reduced-cost constraint based on the objective function of the continuous relaxation. Each sub-problem is built in a first phase through a recursive variable-fixing process up to a predefined size, and is optimally solved in a second phase. Promising results, especially in terms of running time were reported. Defining small-sized sub-problems and considering different branching strategies to solve 0-1 integer programming and mixed-integer programming models have also been considered in other works, as in [85, 189, 191] where authors proposed several iterative schemes where pseudo-cuts are added to strengthen the problem and direct the search. Angelelli et al. applied the *kernel search* to identify a restricted set of promising items [5]. Starting from an optimal solution of the LP-relaxation, new items are then identified and added in the kernel according to solutions of small to moderate size sub-problems. Della Croce and Grosso proposed in [46] another heuristic closely related to these works and using an LP-based core problem. This heuristic is embedded in a partial enumeration where the branching scheme is based on reduced costs of the corresponding LP-relaxation solution value to fix variables. Very recently, an interesting approach was proposed in [169, 170] in which the aim is to reduce the number of constraints. The method reduces the number of dimensions to be constrained by using new basis vectors to represent the optimal item set with less coordinates. Promising results are reported on the largest instances available in the literature, i.e., when $m = 30$ in the OR-Library instances[1] and the instances generated by Glover and Kochenberger in [74]. In particular, the proposed approach provides the optimal or best-known objective values for most instances. It also improves the currently best-known value of one of the Glover and Kochenberger MKP instances and proves the optimality of one other best-known solution.

Concerning metaheuristics we note in particular the scatter search-based algorithm in [84] and the TS-based approach in [115] where authors proposed an hybrid tree-search algorithm that combined TS with a dynamic and adaptive neighborhood search procedure. A two-phase tabu-evolutionary algorithm was also proposed recently in [124]. Authors introduced two solution-based TS methods into the evolutionary framework. They also used a diversity-based population updating rule to keep an efficient population. Computational experiments demonstrated the robustness of the approach since it converged to many best-known solutions and it provided few new best lower bounds. Metaheuristics field is constantly evolving and new approaches are proposed regularly. A very large share of these new methods are tested on MKP. Fingler et al. considered in [60] a parallel implementation of Ant Colony Optimization (ACO) using CUDA and under the GPGPU. They compared the results of this approach with a hybrid DP method, a kernel search and a nested partition and showed that parallel ACO can be an interesting alternative to deal with MKP. Very recently, Lai et al. introduced in [123] a new quantum Particle Swarm Optimization (PSO) that integrates a distance-based diversity preserving strategy and a variable neighborhood descent-based local search method. The results are encouraging and authors showed that the method performs well for instances with a limited number of constraints. It provides better solutions in average than another quantum PSO approach proposed in [81] in very reasonable running time for larger instances. Several other nature-inspired or bio-inspired methods can be identified such as the binary grey wolf optimizer proposed in [130] with some specific components to deal with MKP: an initial elite population generator, a pseudo-utility based repair operator. The results showed that the approach obtained average better results than a few other nature-inspired and bio-inspired methods. Abdel-Basset et al. introduced a modified multi-verse optimization algorithm to solve binary problems [1]. They evaluated this new approach on KP, MKP and MMKP. The results show an interesting behavior of this metaheuristic to tackle different variants of KPs.

To conclude this section it should be observed that in many works dedicated to population-based approaches some of the large instances in the

[1] http://people.brunel.ac.uk/~mastjjb/jeb/info.html.

literature are not considered and that authors do not necessarily compare their results with the state-of-the-art, making it difficult objective comparison.

4.3.2 The Multidimensional Knapsack Problem with Demand Constraints

The 0-1 Multidimensional Knapsack Problem with Demand Constraints (MKPD) is generalization of the MKP in which a set of q *greater than* constraints, $q \geq 1$, complements its multiple knapsack constraints. Accordingly, MKPD extends the MKP formulation as follows:

$$\max \quad \sum_{j=1}^{n} p_j x_j \tag{4.70}$$

$$\text{s.t.}: \quad \sum_{j=1}^{n} w_{ij} x_j \leq c_i \qquad i \in \{1, \ldots, m\} \tag{4.71}$$

$$\sum_{j=1}^{n} w_{ij} x_j \geq c_i \qquad i \in \{m+1, \ldots, m+q\} \tag{4.72}$$

$$x_j \in \{0, 1\} \qquad j \in N = \{1, \ldots, n\} \tag{4.73}$$

In this formulation we distinguish the first m classical knapsack constraints from the subsequent q which are the *demand* constraints. Comparing with MKP and other extensions of KP the MKPD can be viewed as a recent variant if we consider the references to this problem in the literature. Indeed, to the best of our knowledge the MKPD has been formally introduced at the beginning of the twentieth century by Cappanera and Trubian, firstly in [21] and then in the associated paper [22]. Like MKP the MKPD is embedded in the models of many practical applications, in particular in obnoxious and semiobnoxious facility location problems (see, e.g., [20, 162]) and portfolio-selection problems [12]. More recently a possible application in the context of the ideal product mix for a mobile retailer where a resource constraint represents the space of the retailer whereas a demand constraint models the fact the product mix has to meet or exceed a given revenue threshold was considered in [193]. We also noted the work presented in [96] where author considers the constrained problem of multi-robot tasks allocation and compares dedicated algorithms with approaches presented in [6, 102] for solving the more general MKPD. The main conclusion is that a direct application of MKPD dedicated methods does not allow to obtain efficient solutions even for medium-sized instances.

It is interesting to observe that finding a feasible solution of the MKPD is an $\mathcal{NP}$-complete problem [22] due to the presence of conflicting binary constraints. Like for the MKP and other variants a large set of 810 instances is available in the literature to compare the proposed approaches. Most of the proposed approaches dealing with MKPD are based on heuristics and metaheuristics. This can probably be explained by its practical difficulty and the fact that exact methods cannot solve instances with more than 100 items in reasonable running and memory limits. Concerning theoretical works we noted the promising work in [45] in which the author introduced the use of equality rather than inequalities in cutting planes (called equality cuts) in binary programming in general to reduce the linear relaxation space, and the concept of *anticover* inequality. He applied it on the MKPD and showed that using these equality cuts with CPLEX improved the computational effort by about 7% when considering instances with up to 250 items and $m = 10$ and $q = 10$. Wishon and Villalobos introduced in [194] new efficiency measure calculations and some properties and showed that these measures are applicable to all KP variants with a single linear objective function and linear constraints of any quantity. They validated these measures with three solution algorithms (a fixed-core algorithm, a kernel search algorithm and a Genetic Algorithm (GA)) on instances randomly generated with up to 1,000 items and $m + q \leq 25 + 25$.

Among the heuristic and metaheuristic approaches, Cappanera and Trubian proposed in [22] a sophisticated local search algorithm into two phases, which combines a standard attribute-based TS with an oscillation method based on the one proposed for the MKP in [74], but is restricted to the feasible space. The approach was not able to provide a feasible solution to every instance but provided better solutions in general than CPLEX. Arntzen et al. proposed in [6] an adaptive memory search procedure which uses a dynamic TS mechanism and a weighting scheme to manage infeasible solutions. The computational results demonstrated the efficiency of the approach which was able to provide a feasible solution for every instance, except one. Then, Hvattum and L**o**kketangen conducted in [102] several implementations of Scatter Search (SS) for the MKPD based on some proposals about SS in general [75]. They performed extensive experiments and obtained good results compared with CPLEX and Cappanera and Trubian's algorithm. Gortázar et al. introduced in [79] a black box SS method for general classes of binary optimization problems. According to its general description three of the five methods used in SS are problem-dependent. Authors considered some diversification generators and combination methods and evaluated the approach on four optimization problems among which is the MKPD. The results showed that the approach was dominated by the one in [6] but converged quickly to good solutions compared with commercial solvers. Hvattum et al. proposed in [101] an alternating control tree-search framework related to the relaxation-based heuristics proposed in [189] for the MKP. The approach is based on three components: the first one solves an LP-relaxation of the current problem; the second one partitions the set of variables and build sub-problems that can be solved efficiently; the third one updates the problem by adding inequalities and ensuring the convergence of the approach. The overall method can be also used as a heuristic. The algorithm is evaluated on the MKPD and the MKP. The results revealed that the method was very effective for the MKPD compared to the TS algorithm proposed in [6] and the SS algorithm proposed in [102]. It was also competitive for the MKP when compared with some of the best solution approaches. Recently, Lai et al. introduced in [125] a two-stage TS-based algorithm for solving MKPD. The TS uses one-flip and swap neighborhoods and a hash-based mechanism to determine the tabu status of neighbor solutions. In the approach the fist stage aimed to identify a promising hyperplane within the whole search space. The second stage is dedicated to the search for an improved solution by examining both feasible and infeasible solutions on the identified hyperplane. The computational results reported on a subset of the available instances show that this approach performs well and provides efficient solutions in reasonable running times. Very recently, Al-Shihabi proposed a core-based optimization framework and evaluate it over the MDMKP [4]. The main idea is to define a new core problem iteratively using a kind of local search where the aim is to find the *best* neighbouring core problem. The initial core problem is built from the reduced costs of variables and a probing technique is used to control the number of changes in two consecutive core problems. Author evaluates this framework when using CPLEX and the approach proposed in [125] to solve the core problems. Promising results are presented and the method leads to several new best-known solutions.

4.3.3 The Multiple Knapsack Problem

The Multiple Knapsack Problem (M-KP) is a generalization of KP in which a set M of $m \geq 2$ knapsacks are available for carrying the items. This problem, one should note, is different from the multidimensional knapsack problem (see Sect. 4.3.1) and the multidimensional multiple-choice knapsack problem (see Sect. 4.2.1) since in the M-KP every item has to be assigned to one knapsack (at most). In the formulation of the M-KP a binary variable x_{ij} is associated with knapsack i and item j and is set to 1 if item $j \in N$ is assigned to knapsack i (0 otherwise). The problem aims at maximizing total profit while respecting the capacity of every individual knapsack and can be formulated as follows:

$$\max \quad \sum_{i=1}^{m}\sum_{j=1}^{n} p_j x_{ij} \tag{4.74}$$

$$\text{s.t.}: \quad \sum_{j=1}^{n} w_{ij} x_{ij} \leq c_i \qquad i \in M \tag{4.75}$$

$$\sum_{i=1}^{m} x_{ij} \leq 1 \qquad j \in N \tag{4.76}$$

$$x_{ij} \in \{0, 1\} \qquad i \in M, j \in N \tag{4.77}$$

Constraints (4.76) guarantee that every item is placed in at most one knapsack. A special case of the M-KP is obtained if all the knapsacks have an identical capacity c. This variant is generally referred to as the multiple knapsack problem with identical capacities. Additionally, one also finds subset-sum variants for both models, with $p_j = w_j, \forall j \in N$, thus applying. The M-KP is also related to the classical bin-packing problem (BPP). Three main differences can be listed: (i) items have both a weight and a profit in the M-KP, whereas items only have a weight in BPP; (ii) bins (knapsacks) in the M-KP have varying sizes, while bin capacities are uniform in BPP; (iii) the aim of the M-KP is to maximize the total profits of the items packed in a set of bins (knapsacks), and some items are not assigned, whereas all items must be packed in the BPP. Finally one can observe that the M-KP is a special case of the generalized assignment problem in which a profit p_{ij} is defined if item j is assigned to knapsack i. The M-KP and its variants are strongly $\mathcal{NP}$-Hard problems and it was proved that even when $m = 2$ the subset-sum variants do not admit a fully polynomial-time approximation scheme unless $\mathcal{P} = \mathcal{NP}$ (see, e.g., [111]). Polynomial-time approximation schemes were developed in [29] for the MKP and in [24] for the subset-sum variant.

Eilon and Christofides [54] were among the first to consider the M-KP as a cargo loading problem. Ferreira et al. used in [59] a generalization of the M-KP to solve real-world problems in the design of processors for mainframe computers, in the layout of electronic circuits, and in sugar cane alcohol production in Brazil. Another application can be found in [107] where authors considered the M-KP for task allocation among autonomous agents, continuous double-call auctions. The M-KP was also used to model multiprocessor scheduling and the assignment of files to storage devices in order to maximize the number of files stored in the fastest storage devices in [122]. More recently Laalaoui and M'Hallah [121] addressed a single machine scheduling problem with machine unavailability as a M-KP, whereas Simon et al. modeled problems of maintaining operational capability without external support through M-KPs in [172]. Additional constraints occur among which are demand constraints and some others related to the application.

Some B&B methods were proposed to solve the M-KP as in [140] where Martello and Toth solved at each node a relaxed version of the M-KP where the assignment constraint is omitted. The branching item is chosen among those which had been assigned to $\bar{m} > 1$ knapsacks and the branching operation generates $\bar{m}$ nodes by assigning the item to one of the $\bar{m} - 1$ knapsacks or excluding it from all of these. Next year, Martello and Toth [141] proposed one of the most well-known B&B for the M-KP known as `MTM`. In this algorithm a lower bound is derived at each node of the tree by solving m individual KPs in the following way: the first KP is solved optimally. Then, the chosen items are removed and the next KP can be solved and so on until all the m knapsacks are filled. Upper bounds are computed from a surrogate relaxation. The branching scheme is based on the obtained solution by generating two nodes: the first one assigning the next item of a greedy solution to the chosen knapsack, while the other excluding this item from the knapsack. Pisinger [156] proposed a more efficient procedure starting from `MTM`. In this approach called `Mulknap` lower bounds are derived by solving a series of subset-sum problems which are also used for tightening the capacity constraints of the knapsacks. He introduced efficient reduction rules to determine which items cannot be packed into any of the knapsacks and he derived upper bounds through surrogate relaxation. He compared the results of `MTM` and `Mulknap` and showed that the latter approach was able to solve very efficiently and optimally large-size instances of the M-KP with up to $n = 100,000$ items and $m = 10$ knapsacks. However, Fukunaga showed in [66] that `Mulknap` performs less effectively when the ratio of items to knapsacks (n/m) falls between 2 and 5 or when the data are strongly correlated. Fukunaga and Korf [67] introduced a B&B algorithm called *bin-completion* and based on a bin-oriented branching structure and a powerful dominance criterion. Authors showed that this new method is particularly competitive for solving MKP instances when the ratio of items to knapsacks is small (i.e., $n/m < 4$). This work was extended in [66] by integrating additional techniques from constraint programming like path-symmetry and path-dominance criteria for pruning nodes in the M-KP B&B search space. Experimental results confirmed the competitiveness of the approach for instances with a small ration n/m and the integration of `Mulknap` algorithm significantly improved the results for higher ratios. Very recently, Dell'Amico et al. presented in [47] two pseudo-polynomial formulations for the M-KP based on arc-flow and reflect models. Authors proposed a new effective hybrid method combining B&B, Master/Slave decomposition, Benders's cuts and constraint programming. This approach outperforms other exact methods and is able to solve optimally 1,988 instances over the 2,100 instances considered in the paper in reasonable running time.

Papers related to heuristics for the M-KP can also be found in the literature, especially the well-known `MTHM` polynomial-time heuristic based on greedy algorithm and local search procedure proposed in [142], which was tested on instances with up to $n = 10,000$ and $m = 40$ in [143]. More recently, Lalami et al. proposed a heuristic based on the core concept [126]. The approach recursively solves the core of different KPs through DP. The method is tested on randomly generated instances with up to 100,000 items and $m = 100$ and compared with `MTHM`. The results showed that the approach provided good solutions in very short time. Laalaoui [120] proposed two simple extensions of the `MTHM`. It consists in applying a local search starting from the solution of `MTHM` and considering two swap moves: the replacement of one item currently assigned to a knapsack by one or two others currently not assigned; or the replacement of two items currently assigned by one item currently not assigned. The results demonstrates that this method improves by around 15% the solution value provided by `MTHM` with a reasonably increased execution time. Final solutions were at less than 1% in average of the solutions obtained by GAs proposed in [68]. Laalaoui and M'Hallah considered in [121] a single machine scheduling problem with machine unavailability periods corresponding to maintenance periods and modeled it as a M-KP. They proposed a VNS that helps to yield better results than `MTHM` and other heuristics on scheduling instances with up to 4,800 items and 2,400 knapsacks.

To conclude this section we would like to underline another variant of the M-KP in the literature introduced in [109] and named the multiple knapsack assignment problem. In this variant items are partitioned into subsets and additional constraints impose that a knapsack can only contain items of the same class. This interesting problem which has managerial applications in transportation logistics was considered very recently in [139] where authors proposed some upper bounds derived from Lagrangian and surrogate relaxation and a heuristic procedure.

4.3.4 The Compartmentalized Knapsack Problem

The Compartmentalized Knapsack Problem (CKP) is a variant of KP where items are classified under some criteria or affinities and divided into several classes, like for instance in the MCKP. However, in the CKP items of different classes cannot be mixed in the knapsack. To select items some compartments have to be built. The compartments are not predetermined and have to be added inside the knapsack such that each compartment contains items of the same class. Moreover, building a compartment incurs a fixed cost and a fixed loss of the capacity in the original knapsack, and the compartments are lower and upper bounded. The objective is to find the compartments while maximizing the total value of the items loaded in the overall knapsack minus the cost of the compartments. Like in KP each item has a weight and an utility value.

CKP takes its origin from cutting application where it models a particular two-stage cutting process where items are grouped into classes and only items of the same class can be cut from an intermediate roll. It arises for instance in the case of cutting steel coils in two phases in the metallurgical industry [160]. Items are associated to ribbon coils (corresponding to the classes of items) that are grouped by their thickness. The ribbons are obtained by cutting the steel coils available in stock, which correspond to the knapsack to be filled. Another example in the literature is steel roll cutting where items are grouped according to their thickness [127]. In that case the first cut on the original roll generates different intermediate rolls (corresponding to the compartments in the knapsack), which have the same thicknesses as the original roll. In these applications items are modeled by integer variables. Hoto was among the first to introduce and to define the CKP in his Ph.D. thesis [98]. He distinguished the *restricted* (or *constrained*) version where the number of copies for an item is bounded and the *unrestricted* (or *unconstrained*) case where this bound does not exist. He provided an integer nonlinear optimization model for the unrestricted CKP and he proposed a B&B algorithm to solve it. Then, Marques and Arenales proposed in [138] three new heuristics and a model for the restricted case. Hoto et al. considered the cutting stock problem where each cutting pattern is restricted to be compartmentalized [99]. They presented a concise formulation of CKP and proposed heuristic and exact methods for the unrestricted case. They also presented an integer nonlinear programming formulation of the restricted CKP and proposed a two-phase heuristic which obtained better results than the heuristic proposed in [138]. Later, Hoto et al. [100] introduced techniques based on column generation for the restricted CKP, whereas Leão et al. proposed a linear treatment for the restricted CKP and a hybrid heuristic to solve it in [127]. The natural formulation of CKP is nonlinear and some works in the literature were devoted to propose a linear formulation like in [104] where author extended previous works in [39, 100] by proposing new linearity studies of CKP together with a new exact algorithm, called the exhaustive decomposition algorithm, to solve it. More recently, Inarejos et al. [105] proposed an integer linear optimization model for the restricted CKP and proved its linearity. This model was strengthened in [160]. It can be observed from the literature that several variants of the restricted CKP exist. The main difference among these variants is the objective function which is considered. Very recently, Baazaoui et al. proposed in [8] a new variant of CKP with a formulation involving only binary variables. This formulation is related to the one presented in [105] and is introduced below to illustrate CKP. Let c be the capacity of the knapsack and N the set of items which is partitioned into a set M of n^g classes: $N = \cup_{k \in M} N_k$ with $N_k \cap N_{k'} = \emptyset$, $k \neq k' \in M$. Each item j in class k is characterized by a utility and a weight denoted by p_j^k and w_j^k, $j \in N_k, k \in M$, respectively. In addition, for each class $k \in M$ there is a maximum and a minimum capacity to be consumed by a compartment, namely c_{max}^k and c_{min}^k, respectively, whereas building a compartment involves a fixed loss cost f_k. In the formulation a set P_k of all the possible compartments associated with group k is used to define the set of variables (one can observe that the size of P_k can be simply bounded by the number of items in N_k). Binary variable x_j^k is introduced to know if item j in set N_k is selected, whereas binary variable y_{ij}^k is used to know if item $j \in N_k$ is assigned to compartment $i \in P_k$. Finally, binary variable z_i^k is equaled to 1 if compartment $i \in P_k$ is built. Then, a binary variant of CKP can be stated as follows:

$$\max \quad \sum_{k \in M} \left(\sum_{j \in N_k} p_j^k x_j^k - \sum_{i \in P_k} f_k z_i^k \right) \tag{4.78}$$

$$\text{s.t.}: \quad \sum_{k \in M} \sum_{j \in N_k} w_j^k x_j^k \leq c \tag{4.79}$$

$$c_{min}^k \leq \sum_{j \in N_k} w_j^k y_{ij}^k \leq c_{max}^k \qquad \forall k \in M, j \in P_k \tag{4.80}$$

$$\sum_{j \in P_k} y_{ij}^k = x_i^k \qquad k \in M, i \in N_k \tag{4.81}$$

$$x_j^k, y_{ij}^k, z_i^k \in \{0, 1\} \qquad k \in M, j \in N_k, i \in P_k \tag{4.82}$$

Interested readers are invited to consult [105, 160] for other linear formulations with integer variables.

4.3.5 The Multiple Knapsack Problem with Color Constraints

The Multiple Knapsack Problem with Color constraints (MKPC) is another generalization of KP in which: (*i*) a new attribute, called *color*, is associated with every item ; (*ii*) assignment restrictions are introduced regarding the compatibility of items in a given knapsack. MKPC can also be formulated in different ways. The problem was first introduced with this name in [42] motivated by a real application in the steel industry called the surplus inventory matching problem. Here the orders correspond to items and the slabs correspond to knapsacks. The goal of the problem is to maximize the total weight of the surplus used to fulfill orders in the order book. Other objectives can be considered as the minimization of the leftover weight of each used slab. To introduce more formally the problem we have to provide some details about the correspondence between MKPC and the related application [42]. For each item (order) $j \in N$ characterized by a weight w_j, a set M_j of applicable knapsacks (slabs) from the surplus inventory can be identified. These assignment restrictions are based on quality and physical dimension considerations. In addition only knapsacks that are of the same quality or better can be applied for any given item. Then, for each slab (knapsack) $i \in M$ with a weight b_i a set N_i of applicable orders (items) is defined. The thickness and width requirements for each item need to be compatible with those of the applicable knapsacks. Another important notion in MKPC is the color associated with each item. Color constraints restrict the sets of items that can be matched to the same knapsack in the surplus inventory. In fact not all items assignable to a given knapsack can be packed together on that knapsack, due to processing considerations in the finishing line of a steel mill. A route associated with each item that specifies the set of process operations that need to be applied is given. A unique color is associated with each route and the number of colors on a knapsack is restricted (in practice to two). The color can be introduced with parameter k_j to denote the color of order $j \in N$ and with $\mathcal{K}_i$ to denote the set of colors incident on slab $i \in M$.

Forrest et al. proved in [63] that the problem of applying orders against an existing surplus inventory is the surplus inventory matching problem and can be formulated as a multiple knapsack problem with color and assignment constraints considering only one objective. This is achieved by introducing the following binary variables: $x_{ij}, \forall i \in M, j \in N$ which is set to 1 if item j is assigned to knapsack i; $y_{ki}, \forall i \in M, k \in \mathcal{K}_i$ which has value 1 if items of color k obtain material from knapsack i; and $z_i, \forall i \in M$ which is set to 1 if any item is incident to knapsack i. Then, MKPC can be modeled as follows [63]:

$$\max \quad \sum_{j \in N} \sum_{i \in M_j} w_j x_{ij} - \sum_{i \in M} (b_i - \sum_{j \in N_i} w_j x_{ij}) z_i \tag{4.83}$$

$$\text{s.t.}: \quad \sum_{j \in N_i} w_j x_{ij} \leq b_i z_i \qquad i \in M \tag{4.84}$$

$$\sum_{i \in M_j} x_{ij} \leq 1 \qquad j \in N \tag{4.85}$$

$$\sum_{c \in \mathcal{K}_i} y_{ci} \leq 2 \qquad i \in M \tag{4.86}$$

$$x_{ij} \leq y_{c_j,i} \qquad i \in M, j \in N \tag{4.87}$$

$$x_{ij} \in \{0, 1\} \qquad i \in M, j \in N \tag{4.88}$$

$$y_{ci} \in \{0, 1\} \qquad c \in \mathcal{K}_i, i \in M \tag{4.89}$$

$$z_i \in \{0, 1\} \qquad i \in M \tag{4.90}$$

As mentioned by [63] nonlinear objective function (4.83) can be rewritten as

$$\sum_{j \in N} \sum_{i \in M_j} 2 w_j x_{ij} - \sum_{j \in N} b_i z_i \tag{4.91}$$

since $z_i = 0$ implies $x_{ij} = 0, \forall i \in N_j$, whereas $z_i = 1$ implies $x_{ij} z_i = x_{ij}$ for all feasible solutions.

Some variants and formulations of MKPC can be found in the literature. Dawande and Kalagnanam [42] proposed two integer linear programming formulations, one by separating the set of variables for modeling the color constraints and another where these color constraints are implicitly modeled without introducing new variables. They presented a polyhedral study of the problem and showed that the color constraints significantly change the classical multiple knapsack integer polytope. In addition they demonstrated that under certain sufficient conditions color constraints define the facets of the integer polytope. Kalagnanam et al. considered a bi-objective version in [108] where the second objective minimizes the unused weight of the slabs that are matched with orders. Authors proposed a mathematical formulation of the problem as well as a network flow-based heuristic. Experiments performed on real instances proved the efficiency of the heuristic approach to provide near-optimal solutions within a reasonable amount of time. Forrest et al. proposed in [63] a branch-and-price algorithm to deal with the previous formulation that helps in solving a real-life instance of MKPC, called mkc. However, the method could not solve a larger real-life instance, called mkc7 (mkc and mkc7 are two well-known instances in the MIPLIB[2]).

MKPC is also related to the variable-sized bin packing problem with color constraints which consists in assigning to each used bin a capacity from a set of available capacities. Each item is also characterized by a weight and a color among a set of colors. The objective is to minimize the residual capacity in the used bins such that: (i) each item is assigned to exactly one type of the bins, (ii) the total weights of items assigned to each bin do not exceed its capacity and (iii) no more than two colors appear in each used bin. This variant was introduced in [42] where authors presented an asymptotic polynomial approximation scheme that serves to classify the complexity of the problem. Later, Dawande et al. [44] proposed several approximation algorithms. They first considered the case where the maximum number of colors is set to two and then the general case. Then, Dawande et al. [43] developed an efficient heuristic algorithm which is used in a real context and the steel plant savings achieved $2.5 million per year. In a recent work, Kochetov and Kondakov [117] applied a VNS-based matheuristic to a special case where each item has zero weight and arbitrary number of colors with the objective of minimizing the number of bin used. VNS is used for the pricing problem as well as for approaching near-optimal solutions for randomly generated instances. Crévits et al. [37] considered another special case where each color is assigned to only one item. They considered several different linear formulations for that special case of the problem, including some standard ones. Out of the computational experiments reported by the authors, the one that

[2] http://miplib.zib.de.

appears to be the most promising is called the *matching formulation*. Experiments provided promising results for this formulation when it is compared to other classical ones. This work was extended in [82] where authors showed that for that special case instances with up to 20 bin sizes can be solved in less than 2 seconds.

Finally the constraint programming community also considered MKPC as a bin-packing problem with color constraints. The corresponding problem is problem number 38 in the `CSPLIB`[3] and consists of 380 benchmark instances. Frisch et al. proposed in [65] a constraint-based model for the problem and added symmetry-breaking techniques as well as efficient implied constraints that help reducing significantly the search space. Heinz et al. solved all the 380 instances of CSPLIB using a column generation-based algorithm in [90]. The approach also solved the `mkc` instance but was not able to solve the `mkc7` one. The problem was also efficiently solved in [181] where authors used constraint programming with an improved symmetry breaking scheme.

4.4 Conclusion and Suggestions

In this chapter we review a few variants of KP which are associated with recently introduced interesting applications. Our choice of variants is obviously personal and is restricted by writing space limitations. Accordingly, numerous existing KP variants are not addressed here. For each KP variant we consider, at least one formulation is provided, together with a list of existing applications for it. Then, we focused on the most recent and the most effective approaches for solving the problem. Quite clearly, we could have considered many other KP variants including quadratic knapsack problems or multi-objective knapsack problems for instance. Likewise, stochastic and robust versions of KPs have also been considered in the literature. For instance Caserta and Voß introduced in [26] a robust version of the MMCKP and studied the relation between the deterministic and the robust problems. They showed that a given nonlinear model arising from the use of robust optimization can be transformed into an equivalent linear model and thus that the complexity of the deterministic version can be preserved. Online or bilevel knapsack problems are other examples of interesting variants (see., e.g., [23, 149]).

[3] http://www.csplib.org/Problems/prob038/.

In our opinion the number of publications dealing with KP variants is not about to decrease, for the following reasons: (i) many of KP variants remain extremely challenging problems, even when considering very simple versions of them; (ii) as shown in this chapter and in the literature several recent works identify KP variants as sub-problems within even more difficult problems. Thus, the need of very effective approaches to solve KPs is still relevant; (iii) we can imagine that other new variants will emerge in the next few years, in particular by combining the characteristics of existing KPs. For instance Mansini et al. introduced recently the multiple multidimensional knapsack with family-split penalties which combines the M-KP and the MKP in [131]. In this problem the profit is not associated with each single item but with the family as a whole and to earn this profit it is necessary to select all the items of the family. A splitting penalty incurs when items from the same family are assigned to different knapsacks. The problem arises in distributed computing resource management items that are grouped into families. As noted by the authors efficiently handling large instances can be an interesting challenge. Thus, we encourage interested readers to consider both theoretical and more practicable works to help resolve the more emerging and difficult knapsack problems.

Acknowledgments Saïd Hanafi wishes to thank the support of the Chaire Française at Rio of Janeiro State University, 2019 that initiated this project.

References

1. Abdel-Basset, M., D. El-Shahat, H. Faris, and S. Mirjalili (2019). A binary multi-verse optimizer for 0-1 multidimensional knapsack problems with application in interactive multimedia systems. *Computers & Industrial Engineering 132*, 187–206.
2. Akeb, H., M. Hifi, and M. E. O. A. Mounir (2011). Local branching-based algorithms for the disjunctively constrained knapsack problem. *Computers & Industrial Engineering 60*(4), 811–820.
3. Al-Dulaimy, A., W. Itani, R. Zantout, and A. Zekri (2018). Type-aware virtual machine management for energy efficient cloud data centers. *Sustainable Computing: Informatics and Systems 19*, 185–203.
4. Al-Shihabi, S. (2021). A novel core-based optimization framework for binary integer programs—The multidemand multidimensional knapsack problem as a test problem. *Operations Research Perspectives 8*, 100182.

5. Angelelli, E., R. Mansini, and M. Grazia Speranza (2010). Kernel search: A general heuristic for the multi-dimensional knapsack problem. *Computers & Operations Research 37*(11), 2017–2026.
6. Arntzen, H., L. M. Hvattum, and A. Løkketangen (2006). Adaptive memory search for multidemand multidimensional knapsack problems. *Computers & Operations Research 33*, 2508–2525.
7. Arulselvan, A. (2014). A note on the set union knapsack problem. *Discrete Applied Mathematics 169*, 214–218.
8. Baazaoui, M., S. Hanafi, R. Todosijevic, M. Ratli, and C. Wilbaut (2021). A note on the compartmentalized knapsack problem. Technical Report, Université Polytechnique Hauts-de-France.
9. Bae, K.-Y., Y.-D. Kim, and J.-H. Han (2015). Finding a risk-constrained shortest path for an unmanned combat vehicle. *Computers & Industrial Engineering 80*, 245–353.
10. Barcia, P. and K. Jörnsten (1990). Improved Lagrangean decomposition: An application to the generalized assignment problem. *European Journal of Operational Research 46*, 84–92.
11. Basnet, C. and J. Wilson (2005). Heuristics for determining the number of warehouses for storing non-compatible products. *International Transactions in Operational Research 12*(5), 527–538.
12. Beaujon, G. J., S. P. Marin, and G. C. McDonald (2001). Balancing and optimizing a portfolio of R&D projects. *Naval Research Logistics 48*, 18–40.
13. Bellman, R. (1957). *Dynamic Programming*. Princeton, NJ, USA: Princeton University Press.
14. Bettinelli, A., V. Cacchiani, and E. Malaguti (2014). Bounds and algorithms for the knapsack problem with conflict graph. Technical report, OR-14-16, DEIS—University of Bologna, Bologna, Italy.
15. Boland, N., G. Fricke, C.and Froyland, and R. Sotirov (2005). Clique-based facets for the precedence constrained knapsack problem. Technical Report, Tilburg University Repository, The Netherlands.
16. Borradaile, G., B. Heeringa, and G. Wilfong (2011). The 1-neighbour knapsack problem. In *International Workshop on Combinatorial Algorithms*, pp. 71–84. Berlin and Heidelberg: Springer.
17. Borradaile, G., B. Heeringa, and G. Wilfong (2012). The knapsack problem with neighbour constraints. *Journal of Discrete Algorithms 16*, 224–235.
18. Boussier, S., M. Vasquez, Y. Vimont, S. Hanafi, and P. Michelon (2010). A multi-level search strategy for the 0-1 Multidimensional Knapsack Problem. *Discrete Applied Mathematics 158*(2), 97–109.
19. Boyd, E. A. (1993). Polyhedral results for the precedence-constrained knapsack problem. *Discrete Applied Mathematics 41*(3), 185–201.
20. Cappanera, P., G. Gallo, and F. Maffioli (2003). Discrete facility location and routing of obnoxious activities. *Discrete Applied Mathematics 133*, 3–28.

21. Cappanera, P. and M. Trubian (2001). A local search based heuristic for the demand constrained multidimensional knapsack problem. Technical Report TR-01-10, Dipartimento di Informatica, Università di Pisa.
22. Cappanera, P. and M. Trubian (2005). A local-search-based heuristic for the demand-constrained multidimensional knapsack problem. *INFORMS Journal on Computing 17*(1), 82–98.
23. Caprara, A., M. Carvalho, A. Lodi, and G. J. Woeginger (2016). Bilevel knapsack with interdiction constraints. *INFORMS Journal on Computing 28*(2), 319–333.
24. Caprara, A., H. Kellerer, and U. Pferschy (2000). A PTAS for the multiple subset-sum problem with different knapsack capacities. *Information Processing Letters 73*(3), 111–118.
25. Caserta, M. and S. Voß (2015). An exact algorithm for the reliability redundancy allocation problem. *European Journal of Operational Research 244*, 110–116.
26. Caserta, M. and S. Voß (2019). The robust multiple-choice multidimensional knapsack problem. *Omega 86*, 16–27.
27. Chajakis, E. D. and M. Guignard (1994). Exact algorithms for the setup knapsack problem. *INFOR: Information Systems and Operational Research 32*(3), 124–142.
28. Chebil, K. and M. Khemakhem (2015). A dynamic programming algorithm for the knapsack problem with setup. *Computers & Operations Research 64*, 40–50.
29. Chekuri, C. and S. Khanna (2000). A PTAS for the multiple knapsack problem. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 213–222.
30. Chen, F., T. La Porta, and M. B. Srivastava (2012). Resource allocation with stochastic demands. In *2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems*, Hangzhou, Zhejiang, China, pp. 257–264.
31. Chen, Y. and J. K. Hao (2014). A "reduce and solve" approach for the multiple-choice multidimensional knapsack problem. *European Journal of Operational Research 239*, 313–322.
32. Cherfi, N. and M. Hifi (2009). Hybrid algorithms for the multiple-choice multi-dimensional knapsack problem. *International Journal of Operational Research 5*(1), 89–109.
33. Cherfi, N. and M. Hifi (2010). A column generation method for the multiple-choice multi-dimensional knapsack problem. *Computational Optimization and Applications 46*, 51–73.
34. Chvátal, V. (1997). Resolution search. *Discrete Applied Mathematics 73*, 81–99.
35. Conforti, M., G. Cornuéjols, and G. Zambelli (2010). Extended formulations in combinatorial optimization. *4OR 8*(1), 1–48.

36. Coniglio, S., F. Furini, and P. San Segundo (2020). A new combinatorial branch-and-bound algorithm for the knapsack problem with conflicts. *European Journal of Operational Research 289*(2), 435–455.
37. Crévits, I., S. Hanafi, A. R. Mahjoub, R. Taktak, and C. Wilbaut (2019). A special case of variable-sized bin packing problem with color constraints. In *6th International Conference on Control, Decision and Information Technologies (CoDIT'19)*, Paris, France, pp. 1150–1154.
38. Crévits, I., S. Hanafi, R. Mansi, and C. Wilbaut (2012). Iterative semi-continuous relaxation heuristics for the multiple-choice multidimensional knapsack problem. *Computers & Operations Research 39*(1), 32–41.
39. Cruz, E. P. (2010). *A linear heuristic approach to constrained compartmentalized knapsacks*. Master thesis, Londrina State University, Londrina, Brazil.
40. Dantzig, G. B. (1957). Discrete-variable extremum problems. *Operations Research 5*(2), 266–288.
41. Dantzig, T. (1930). *Number the Language of Science*. Free Press.
42. Dawande, M. and J. Kalagnanam (1998). The multiple knapsack problem with color constraints. Technical Report RC21138, IBM Research Division.
43. Dawande, M., J. Kalagnanam, H. S. Lee, C. Reddy, S. Siegel, and M. Trumbo (2004). The slab-design problem in the steel industry. *Interfaces 34*(3), 215–225.
44. Dawande, M., J. Kalagnanam, and J. Sethuraman (2001). Variable sized bin packing with color constraints. *Electronic Notes in Discrete Mathematics 7*, 154–157.
45. Delissa, L. (2014). *The existence and usefulness of equality cuts in the multidemand multidensional knapsack problem*. Master thesis, Kansas State University.
46. Della Croce, F. and A. Grosso (2012). Improved core problem based heuristics for the 0/1 multi-dimensional knapsack problem. *Computers & Operations Research 39*(1), 27–31.
47. Dell'Amico, M., M. Delorme, M. Iori, and S. Martello (2019). Mathematical models and decomposition methods for the multiple knapsack problem. *European Journal of Operational Research 274*, 886–899.
48. Diallo, C., U. Venkatadri, A. Khatab, and Z. Liu (2018). Optimal selective maintenance decisions for large serial k-out-of-n: G systems under imperfect maintenance. *Reliability Engineering and System Safety 175*, 234–245.
49. Dietrich, B. L. (1988). *Scheduling on parallel unrelated machines with set-ups*. IBM TJ Watson Research Center.
50. Dietrich, B. L. and L. F. Escudero (1991). New procedures for preprocessing 0-1 models with knapsack-like constraints and conjunctive and/or disjunctive variable upper bounds. *INFOR: Information Systems and Operational Research 29*(4), 305–317.
51. Dyer, M. E., N. Kayal, and J. Walker (1984). A branch and bound algorithm for solving the multiple choice knapsack problem. *Journal of Computational and Applied Mathematics 11*(2), 231–249.

52. D'Ambrosio, C., F. Furini, M. Monaci, and E. Traversi (2018). On the product knapsack problem. *Optimization Letters 12*(4), 691–712.
53. Edmonds, J. (1971). Matroids and the greedy algorithm. *Mathematical Programming 1*(1), 127–136.
54. Eilon, S. and N. Christofides (1971). The loading problem. *Management Science 17*, 259–268.
55. Fan, N. and M. Golari (2014). Integer programming formulations for minimum spanning forests and connected components in sparse graphs. In *International Conference on Combinatorial Optimization and Applications*, Cham, Springer, pp. 613–622.
56. Feng, Y.-H. and G.-G. Wang (2018). Binary moth search algorithm for discounted 0-1 knapsack problem. *IEEE Access 6*, 10708–10719.
57. Feng, Y., H. An, and X. Gao (2019). The importance of transfer function in solving set-union knapsack problem based on discrete moth search algorithm. *Mathematics 7*(1).
58. Feng, Y., G.-G. Wang, W. Li, and N. Li (2018). Multi-strategy monarch butterfly optimization algorithm for discounted 0-1 knapsack problem. *Neural Computing and Applications 30*(10), 3019–3036.
59. Ferreira, C., A. Martins, and R. Weismantel (1996). Solving multiple knapsack problems by cutting planes. *SIAM Journal on Optimization 6*(3), 858–877.
60. Fingler, H., E. N. Caceres, H. Mongelli, and S. W. Song (2014). A CUDA based solution to the multidimensional knapsack problem using the Ant Colony Optimization. *Procedia Computer Science, ICCS 2014. 14th International Conference on Computational Science 29*, 84–94.
61. Fischer, L., B. Hammer, and H. Wersing (2016). Optimal local rejection for classifiers. *Neurocomputing 214*, 445–457.
62. Fisher, M. L. (1981). The Lagrangian relaxation method for solving integer programming problems. *Management Science 27*, 1–18.
63. Forrest, J. J. H., J. Kalagnanam, and L. Ladanyi (2006). A column generation approach to the multiple knapsack problem with color constraints. *INFORMS Journal on Computing 18*(1), 129–134.
64. Fréville, A. and S. Hanafi (2005). The multidimensional 0-1 knapsack problem—Bounds and computational aspects. *Annals of Operations Research 139*(1), 195–227.
65. Frisch, A. M., I. Miguel, and T. Walsh (2001). Symmetry and implied constraints in the steel mill slab design problem. In *CP'01 Workshop on Modelling and Problem Formulation*, pp. 8–15.
66. Fukunaga, A. S. (2011). A branch-and-bound algorithm for hard multiple knapsack problems. *Annals of Operations Research 184*(1), 97–119.
67. Fukunaga, A. S. and R. Korf (2007). Bin completion algorithms for multicontainer packing, knapsack, and covering problems. *Journal of Artificial Intelligence Research 28*, 393–429.

68. Fukunaga, A. S. and S. Tazoe (2009). Combining multiple representations in a genetic algorithm for the multiple knapsack problem. In *2009 IEEE Congress on Evolutionary Computation*, pp. 2423–2430.
69. Gao, C., G. Lu, and J. Li (2017). An iterative pseudo-gap enumeration approach for the multidimensional multiple-choice knapsack problem. *European Journal of Operational Research 260*, 1–11.
70. Garey, M. R. and D. S. Johnson (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: W. H. Freeman and Company.
71. Gavish, B. and H. Pirkul (1982). Allocation of databases and processors in a distributed computing system. In J. Akoka (Ed.), *Management of Distributed Data Processing*, pp. 215–231.
72. Gens, G. V. and E. V. Levner (1979). Computational complexity of approximation algorithms for combinatorial problems. In G. Goos, J. Hartmanis, P. Brinch Hansen, D. Gries, C. Moler, G. Seegmüller, J. Stoer, N. Wirth, and J. Bečvář (Eds.), *Mathematical Foundations of Computer Science 1979*, Volume 74, pp. 292–300. Berlin and Heidelberg: Springer.
73. Gilmore, P. C. and R. E. Gomory (1966). The theory and computation of knapsack functions. *Operations Research 14*(6), 1045–1074.
74. Glover, F. and G. A. Kochenberger (1996). Critical event tabu search for multidimensional knapsack problems. In I. H. Osman and J. P. Kelly (Eds.), *Meta-Heuristics*, pp. 407–427. Boston, MA: Springer.
75. Glover, F., M. Laguna, and R. Martí (2003). Scatter search. In A. Ghosh and S. Tsutsui (Eds.), *Advances in Evolutionary Computation: Theory And Applications*, pp. 519–537. Springer.
76. Goldschmidt, O., D. Nehme, and G. Yu (1993). On a generalization of the knapsack problem with applications to flexible manufacturing systems and database partitioning. Technical report, Department of Management Science and Information Systems, Working Paper No. 92193-3-7, University of Texas at Austin.
77. Goldschmidt, O., D. Nehme, and G. Yu (1994). Note: On the set-union knapsack problem. *Naval Research Logistics 41*(6), 833–842.
78. Goodman, M. D., K. Dowsland, and J. M. Thompson (2009). A GRASP-knapsack hybrid for a nurse-scheduling problem. *Journal of Heuristics 15*, 351–379.
79. Gortázar, F., A. Duarte, M. Laguna, and R. Martí (2010). Black box scatter search for general classes of binary optimization problems. *Computers & Operations Research 37*, 1977–1986.
80. Guldan, B. (2007). *Heuristic and exact algorithms for discounted knapsack problems*. Master thesis, University of Erlangen-Nürnberg, Germany.
81. Haddar, B., M. Khemakhem, S. Hanafi, and C. Wilbaut (2016). A hybrid quantum particle swarm optimization for the multidimensional knapsack problem. *Engineering Applications of Artificial Intelligence 55*, 1–13.

82. Hanafi, S., A. R. Mahjoub, R. Taktak, and C. Wilbaut (2021). Variable-sized bin packing problem with color constraints. In *JPOC 12—Journées Polèdres et Optimisation Combinatoire, 22-25 juin 2021.*

83. Hanafi, S., R. Mansi, and C. Wilbaut (2009). Iterative relaxation-based heuristics for the multiple-choice multidimensional knapsack problem. In M. J. Blesa, C. Blum, L. Di Gaspero, A. Roli, M. Sampels, and A. Schaerf (Eds.), *Hybrid Metaheuristics, HM 2009*, Volume 5818 of Lecture Notes in Computer Science, pp. 73–83. Berlin and Heidelberg: Springer.

84. Hanafi, S. and C. Wilbaut (2008). Scatter search for the 0-1 multidimensional knapsack problem. *Journal of Mathematical Modelling and Algorithms 7*(2), 143–159.

85. Hanafi, S. and C. Wilbaut (2011). Improved convergent heuristics for the 0-1 multidimensional knapsack problem. *Annals of Operations Research 183*(1), 125–142.

86. He, Y., X. Wang, and S. Gao (2019). Ring Theory-Based Evolutionary Algorithm and its application to D{0-1} KP. *Applied Soft Computing 77*, 714–722.

87. He, Y., X. Wang, W. Li, X. Zhang, and Y. Chen (2016). Research on genetic algorithm for discounted 0-1 knapsack problem. *Chinese Journal of Computers 39*(12), 2614–2630.

88. He, Y., H. Xie, T. L. Wong, and X. Wang (2018). A novel binary artificial bee colony algorithm for the set-union knapsack problem. *Future Generation Computer Systems 78*, 77–86.

89. He, Y.-C., X.-Z. Wang, Y.-L. He, S.-L. Zhao, and W.-B. Li (2016). Exact and approximate algorithms for discounted {0–1} knapsack problem. *Information Sciences 369*, 634–647.

90. Heinz, S., T. Schlechte, R. Stephan, and M. Winkler (2012). Solving steel mill slab design problems. *Constraints 17*(1), 39–50.

91. Hifi, M. and M. Michrafy (2006). A reactive local search-based algorithm for the disjunctively constrained knapsack problem. *Journal of the Operational Research Society 57*(6), 718–726.

92. Hifi, M. and M. Michrafy (2007). Reduction strategies and exact algorithms for the disjunctively constrained knapsack problem. *Computers & Operations research 34*(9), 2657–2673.

93. Hifi, M., M. Michrafy, and A. Sbihi (2004). Heuristic algorithms for the multiple-choice multidimensional knapsack problem. *Journal of the Operational Research Society 55*(12), 1323–1332.

94. Hifi, M. and N. Otmani (2012). An algorithm for the disjunctively constrained knapsack problem. *International Journal of Operational Research 13*(1), 22–42.

95. Hifi, M., S. Saleh, and L. Wu (2015). A hybrid guided neighborhood search for the disjunctively constrained knapsack problem. *Cogent Engineering 2*(1), 2–24.

96. Hojda, M. (2017). Task allocation for multi-robot teams in dynamic environments. In W. Mitkowski, J. Kacprzyk, K. Oprzedkiewicz, and P. Skruch (Eds.), *Trends in Advanced Intelligent Control, Optimization and Automation*, Volume 577 of Advances in Intelligent Systems and Computing, pp. 483–492. Cham: Springer.
97. Holte, R. C. (2001). Combinatorial auctions, knapsack problems, and hill-climbing search. In G. Goos, J. Hartmanis, J. van Leeuwen, E. Stroulia, and S. Matwin (Eds.), *Advances in Artificial Intelligence*, Volume 2056, pp. 57–66. Berlin and Heidelberg: Springer.
98. Hoto, R. (2001). *The compartmentalized knapsack problem applied to the steel roll cutting*. Ph.D. thesis, Rio de Janeiro Federal University, Brazil.
99. Hoto, R., M. N. Arenales, and N. Maculan (2007). The one-dimensional compartmentalised knapsack problem: A case study. *European Journal of Operational Research 183*(3), 1183–1195.
100. Hoto, R., N. Maculan, and A. Borssoi (2010). A study of the compartmentalized knapsack problem with additional restrictions. *IEE Latin America Transactions 8*(3), 269–274.
101. Hvattum, L. M., H. Arntzen, A. Løkketangen, and F. Glover (2010). Alternating control tree search for knapsack/covering problems. *Journal of Heuristics 16*, 239–258.
102. Hvattum, L. M. and A. Løkketangen (2007). Experiments using scatter search for the multidemand multidimensional knapsack problem. In K. F. Doerner, M. Gendreau, P. Greistorfer, W. Gutjahr, R. F. Hartl, and M. Reimann (Eds.), *Metaheuristics*, Volume 39 of *Operations Research/Computer Science Interfaces Series*, pp. 3–24. Boston, MA: Springer US.
103. Hwan, S. S. and A. W. Shogan (1989). Modelling and solving an FMS part selection problem. *International Journal of Production Research 27*(8), 1349–1366.
104. Inarejos, O. (2015). *On the non-linearity of the compartmentalized backpack problem*. Master thesis, Londrina State University, Londrina, Brazil.
105. Inarejos, O., R. Hoto, and N. Maculan (2019). An integer linear optimization model to the compartmentalized knapsack problem. *International Transactions in Operational Research 26*(5), 1698–1717.
106. Johnson, D. S. and K. Niemi (1983). On knapsacks, partitions, and a new dynamic programming technique for trees. *Mathematics of Operations Research 8*(1), 1–14.
107. Kalagnanam, J., A. Davenport, and H. Lee (2001). Computational aspects of clearing continuous call double auctions with assignment constraints and indivisible demand. *Electronic Commerce Research 1*, 221–238.
108. Kalagnanam, J., M. Dawande, M. Trumbo, and H. S. Lee (2000). The surplus inventory matching problem in the process industry. *Operations Research 48*(4), 505–516.

109. Kataoka, S. and T. Yamada (2014). Upper and lower bounding procedures for the multiple knapsack assignment problem. *European Journal of Operational Research 237*, 440–447.
110. Kellerer, H. (2008). An approximation algorithm for identical parallel machine scheduling with resource dependent processing times. *Operations Research Letters 36*, 157–159.
111. Kellerer, H., U. Pferschy, and D. Pisinger (2004). *Knapsack Problems*. Berlin and Heidelberg: Springer.
112. Kelly, T. (2006). Generalized knapsack solvers for multi-unit combinatorial auctions: Analysis and application to computational resource allocation. In P. Faratin and J. A. Rodríguez-Aguilar (Eds.), *Agent-Mediated Electronic Commerce VI. Theories for and Engineering of Distributed Mechanisms and Systems*, Volume 3435 of Lecture Notes in Computer Science pp. 73–86. Berlin and Heidelberg: Springer.
113. Khan, S., K. F. Li, and E. G. Manning (1997). The utility model for adaptive multimedia system. In *International Workshop on Multimedia Modeling*, pp. 111–126.
114. Khemakhem, M. and K. Chebil (2016). A tree search based combination heuristic for the knapsack problem with setup. *Computers & Industrial Engineering 99*, 280–286.
115. Khemakhem, M., B. Haddar, K. Chebil, and S. Hanafi (2012). A filter-and-fan metaheuristic for the 0-1 multidimensional knapsack problem. *International Journal of Applied Metaheuristic Computing 3*(4), 43–63.
116. Khuller, S., A. Moss, and J. Naro (1999). The budgeted maximum coverage problem. *Information Processing Letters 70*(1), 39–45.
117. Kochetov, Y. and A. Kondakov (2017). VNS matheuristic for a bin packing problem with a color constraint. *Electronic Notes in Discrete Mathematics 58*, 39–46.
118. Kolliopoulos, S. G. and G. Steiner (2007). Partially ordered knapsack and applications to scheduling. *Discrete Applied Mathematics 155*(8), 889–897.
119. Laabadi, S., M. Naimi, H. Amri, and B. Achchab (2018). The 0/1 multidimensional knapsack problem and its variants: A survey of practical models and heuristic approaches. *American Journal of Operations Research 8*(5), 395–439.
120. Laalaoui, Y. (2013). Improved swap heuristic for the multiple knapsack problem. In I. Rojas, G. Joya, and J. Gabestany (Eds.), *Advances in Computational Intelligence: 12th International Work-Conference on Artificial Neural Networks—IWANN 2013*, pp. 547–555. Heidelberg: Springer and Berlin.
121. Laalaoui, Y. and R. M'Hallah (2016). A binary multiple knapsack model for single machine scheduling with machine unavailability. *Computers & Operations Research 72*, 71–82.
122. Labbé, L., G. Laporte, and S. Martello (2003). Upper bounds and algorithms for the maximum cardinality bin packing problem. *European Journal of Operational Research 149*, 490–498.

123. Lai, X., J.-K. Hao, Z.-H. Fu, and D. Yue (2020). Diversity-preserving quantum particle swarm optimization for the multidimensional knapsack problem. *Expert Systems with Applications 149*, 113310.
124. Lai, X., J.-K. Hao, F. Glover, and Z. Lü (2018). A two-phase tabu-evolutionary algorithm for the 0-1 multidimensional knapsack problem. *Information Sciences 436–437*, 282–301.
125. Lai, X., J.-K. Hao, and D. Yue (2019). Two-stage solution-based tabu search for the multidemand multidimensional knapsack problem. *European Journal of Operational Research 274*, 35–48.
126. Lalami, M., M. Elkihel, D. El Baz, and V. Boyer (2012). A procedure-based heuristic for 0-1 multiple knapsack problems. *International Journal of Mathematics in Operational Research 4*(3), 214–224.
127. Leão, A. A. S., M. O. Santos, R. Hoto, and M. N. Arenales (2011). The constrained compartmentalized knapsack problem: Mathematical models and solution methods. *European Journal of Operational Research 212*, 455–463.
128. Letchford, A. N. and G. Souli (2020). Lifting the knapsack cover inequalities for the knapsack polytope. *Operations Research Letters 48*(5), 607–611.
129. Lorie, J. H. and L. J. Savage (1955). Three problems in rationing capital. *The Journal of Business 28*(4), 229–239.
130. Luo, K. and Q. Zhao (2019). A binary grey wolf optimizer for the multidimensional knapsack problem. *Applied Soft Computing 83*, 105645.
131. Mancini, S., M. Ciavotta, and C. Meloni (2021). The multiple multidimensional knapsack with family-split penalties. *European Journal of Operational Research 289*(3), 987–998.
132. Mansi, R., C. Alves, J. M. V. de Carvalho, and S. Hanafi (2013). A hybrid heuristic for the multiple choice multidimensional knapsack problem. *Engineering Optimization 45*(8), 983–1004.
133. Mansini, R. and M. G. Speranza (2002). A multidimensional knapsack model for asset-backed securitization. *Journal of the Operational Research Society 53*(8), 822–832.
134. Mansini, R. and M. G. Speranza (2012). CORAL: An exact algorithm for the multidimensional knapsack problem. *INFORMS Journal on Computing 24*(3), 399–415.
135. Mansini, R. and R. Zanotti (2020). A core-based exact algorithm for the multidimensional multiple choice knapsack problem. *INFORMS Journal on Computing 32*(4), 1061–1079.
136. Marchand, H. and L. A. Wolsey (1999). The 0-1 knapsack problem with a single continuous variable. *Mathematical Programming 85*(1), 15–33.
137. Markowitz, H. M. and A. S. Manne (1957). On the solution of discrete programming problems. *Econometrica 25*(1), 84–110.
138. Marques, F. P. and M. N. Arenales (2002). The compartmentalized knapsack problem and applications. *Pesquisa Operacional 22*(3), 285–304.
139. Martello, S. and M. Monaci (2020). Algorithmic approaches to the multiple knapsack assignment problem. *Omega 90*, 102004.

140. Martello, S. and P. Toth (1980). Solution of the zero-one multiple knapsack problem. *European Journal of Operational Research 4*, 276–283.
141. Martello, S. and P. Toth (1981a). A bound and bound algorithm for the zero-one multiple knapsack problem. *Discrete Applied Mathematics 3*, 275–288.
142. Martello, S. and P. Toth (1981b). Heuristic algorithms for the multiple knapsack problem. *Computing 27*(2), 93–112.
143. Martello, S. and P. Toth (1990). *Knapsack Problems: Algorithms and Computer Implementations*. Wiley-Interscience Series in Discrete Mathematics and Optimization. Chichester and New York: Wiley.
144. Martin, R. K. (1991). Using separation algorithms to generate mixed integer model reformulations. *Operations Research Letters 10*(3), 119–128.
145. Mathews, G. B. (1896, 11). On the partition of numbers. *Proceedings of the London Mathematical Society s1-28*(1), 486–490.
146. Mejia-Alvarez, P., E. V. Levner, and D. Mosse (2002). An integrated heuristic approach to power-aware real-time scheduling. In *International Workshop on Power Aware Computer Systems (PACS'02)*, Volume 2325 of Lecture Notes in Computer Science. Springer.
147. Michel, S., N. Perrot, and F. Vanderbeck (2009). Knapsack problems with setups. *European Journal of Operational Research 196*(3), 909–918.
148. Mohammadivojdan, R. and J. Geunes (2018). The newsvendor problem with capacitated suppliers and quantity discounts. *European Journal of Operational Research 271*, 109–119.
149. Navarra, A. and C. M. Pinotti (2017). Online knapsack of unknown capacity: How to optimize energy consumption in smartphones. *Theoretical Computer Science 697*, 98–109.
150. Nemhauser, G. L. and Z. Ullmann (1969). Discrete dynamic programming and capital allocation. *Management Science 15*(9), 494–505.
151. Ozsoydan, F. B. and A. Baykasoglu (2019). A swarm intelligence-based algorithm for the set-union knapsack problem. *Future Generation Computer Systems 93*, 560–569.
152. Park, K. and S. Park (1997). Lifting cover inequalities for the precedence-constrained knapsack problem. *Discrete Applied Mathematics 72*(3), 219–241.
153. Pferschy, U. and J. Schauer (2009). The knapsack problem with conflict graphs. *Journal of Graph Algorithms and Applications 13*(2), 233–249.
154. Pferschy, U., J. Schauer, and C. Thielen (2020). The product knapsack problem: Approximation and complexity.
155. Pisinger, D. (1995). A minimal algorithm for the multiple-choice knapsack problem. *European Journal of Operational Research 83*, 394–410.
156. Pisinger, D. (1999). An exact algorithm for large multiple knapsack problems. *European Journal of Operational Research 114*, 528–541.
157. Pisinger, D. (2001). Budgeting with bounded multiple-choice constraints. *European Journal of Operational Research 129*, 471–480.

158. Pop, P. C. (2009). A survey of different integer programming formulations of the generalized minimum spanning tree problem. *Carpathian Journal of Mathematics 25*(1), 104–118.

159. Puchinger, J., G. R. Raidl, and U. Pferschy (2010). The multidimensional knapsack problem: Structure and algorithms. *INFORMS Journal on Computing 22*(2), 250–265.

160. Quiroga-Orozco, J. J., J. M. V. de Carvalho, and R. S. V. Hoto (2019). A strong integer linear optimization model to the compartmentalized knapsack problem. *International Transactions in Operational Research 26*(5), 1633–1654.

161. Rogeau, A., R. Girard, Y. Abdelouadoud, M. Thorel, and G. Kariniotakis (2020). Joint optimization of building-envelope and heating-system retrofits at territory scale to enhance decision-aiding. *Applied Energy 264*, 114639.

162. Romero-Morales, D., E. Carrizosa, and E. Conde (1997). Semi-obnoxious location models: A global optimization approach. *European Journal of Operational Research 102*(2), 295–301.

163. Rong, A., J. R. Figueira, and K. Klamroth (2012). Dynamic programming based algorithms for the discounted 0-1 knapsack problem. *Applied Mathematics and Computation 218*(12), 6921–6933.

164. Salem, M. B., S. Hanafi, R. Taktak, and H. Ben-Abdallah (2017). Probabilistic tabu search with multiple neighborhoods for the disjunctively constrained knapsack problem. *RAIRO-Operations Research 51*(3), 627–637.

165. Salem, M. B., R. Taktak, A. R. Mahjoub, and H. Ben-Abdallah (2018). Optimization algorithms for the disjunctively constrained knapsack problem. *Soft Computing 22*(6), 2025–2043.

166. Salkin, H. M. and C. A. D. Kluyver (1975). The knapsack problem: A survey. *Naval Research Logistics Quarterly 22*(1), 127–144.

167. Samphaiboon, N. and Y. Yamada (2000). Heuristic and exact algorithms for the precedence-constrained knapsack problem. *Journal of optimization theory and applications 105*(3), 659–676.

168. Senisuka, A., B. You, and T. Yamada (2005). Reduction and exact algorithms for the disjunctively constrained knapsack problem. In *International Symposium on OR and Its Applications*, pp. 241–254.

169. Setzer, T. and S. M. Blanc (2020a). Corrigendum to “empirical orthogonal constraint generation for multidimensional 0/1 knapsack problems”. *European Journal of Operational Research 286*(2), 791–795.

170. Setzer, T. and S. M. Blanc (2020b). Empirical orthogonal constraint generation for multidimensional 0/1 knapsack problems. *European Journal of Operational Research 282*(1), 58–70.

171. Shaw, D. X., G. Cho, and H. Chang (1997). A depth-first dynamic programming procedure for the extended tree knapsack problem in local access network design. *Telecommunication Systems 7*(1), 29–43.

172. Simon, J., A. Apte, and E. Regnier (2017). An application of the multiple knapsack problem: The self-sufficient marine. *European Journal of Operational Research 256*(3), 868–876.

173. Sinha, A. and A. Zoltners (1979). The multiple-choice knapsack problem. *Operations Research 27*, 503–515.
174. Stecke, K. E. and I. Kim (1988). A study of FMS part type selection approaches for short-term production planning. *International Journal of Flexible Manufacturing Systems 1*(1), 7–29.
175. Taylor, R. (2016). Approximations of the densest k-subhypergraph and set union knapsack problems.
176. Thesen, A. (1972). *Scheduling of computer programs in a multiprogramming environment*. Ph.D. thesis, University of Illinois at Urbana-Champaign.
177. Tillman, I. A., C. L. Hwang, and W. Kuo (1980). *Optimization of System Reliability*. New York: Marcel Dekker.
178. Tisch, M., H. Laudemann, A. Kreß, and J. Metternich (2017). Utility-based configuration of learning factories using a multidimensional, multiple-choice knapsack problem. In *Procedia Manufacturing*, Volume 9, pp. 25–32.
179. Tsesmetzis, D., I. Roussaki, and E. Sykas (2008). QoS-aware service evaluation and selection. *European Journal of Operational Research 191*, 1101–1112.
180. Tu, M. and L. Xiao (2016). System resilience enhancement through modularization for large scale cyber systems. In *2016 IEEE/CIC International Conference on Communications in China (ICCC Workshops)*, pp. 1–6.
181. Van Hentenryck, P. and L. Michel (2008). The steel mill slab design problem revisited. In *International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming*, pp. 377–381.
182. Vasquez, M. and J.-K. Hao (2001). A "logic-constrained" knapsack formulation and a tabu algorithm for the daily photograph scheduling of an earth observation satellite. *Computational Optimization and Applications 20*(2), 137–157.
183. Voß, S. and E. Lalla-Ruiz (2016). A set partitioning reformulation for the multiple-choice multidimensional knapsack problem. *Engineering Optimization 48*(5), 832–850.
184. Wei, Z. and J. K. Hao (2019). Iterated two-phase local search for the set-union knapsack problem. *Future Generation Computer Systems 101*, 1005–1017.
185. Wei, Z. and J. K. Hao (2021a). Kernel based tabu search for the set-union knapsack problem. *Expert Systems with Applications 165*, 113802.
186. Wei, Z. and J. K. Hao (2021b). Multistart solution-based tabu search for the set-union knapsack problem. *Applied Soft Computing Journal 105*, 107260.
187. Wei, Z. and J. K. Hao (2021c). A threshold search based memetic algorithm for the disjunctively constrained knapsack problem.
188. Weingartner, H. M. (1966). Capital budgeting of interrelated projects: Survey and synthesis. *Management Science 12*(7), 485–516.
189. Wilbaut, C. and S. Hanafi (2009). New convergent heuristics for 0-1 mixed integer programming. *European Journal of Operational Research 195*(1), 62–74.

190. Wilbaut, C., S. Hanafi, and S. Salhi (2007). A survey of effective heuristics and their application to a variety of knapsack problems. *IMA Journal of Management Mathematics 19*(3), 227–244.

191. Wilbaut, C., S. Salhi, and S. Hanafi (2009). An iterative variable-based fixation heuristic for the 0-1 multidimensional knapsack problem. *European Journal of Operational Research 199*(2), 339–348.

192. Wilbaut, C., R. Todosijevic, S. Hanafi, and A. Fréville (2021). Heuristic and exact fixation-based approaches for the discounted 0-1 knapsack problem.

193. Wishon, C. and J. R. Villalobos (2016a). Alleviating food disparities with mobile retailers: Dissecting the problem from an OR perspective. *Computers & Industrial Engineering 91*, 154–164.

194. Wishon, C. and J. R. Villalobos (2016b). Robust efficiency measures for linear knapsack problem variants. *European Journal of Operational Research 254*, 398–409.

195. Wu, C., J. Zhao, Y. Feng, and M. Lee (2020). Solving discounted 0-1 knapsack problems by a discrete hybrid teaching-learning-based optimization algorithm. *Applied Intelligence 50*, 1872–1888.

196. Yamada, T., S. Kataoka, and K. Watanabe (2002). Heuristic and exact algorithms for the disjunctively constrained knapsack problem. *Information Processing Society of Japan Journal 43*(9), 2864–2870.

197. Yamada, T., K. Watanabe, and S. Kataoka (2005). Algorithms to solve the knapsack constrained maximum spanning tree problem. *International Journal of Computer Mathematics 82*(1), 23–34.

198. Yang, X., A. Vernitski, and L. Carrea (2016). An approximate dynamic programming approach for improving accuracy of lossy data compression by bloom filters. *European Journal of Operational Research 252*(3), 985–994.

199. Ykman-Couvreur, C., V. Nollet, F. Catthoor, and H. Corporaal (2016). Fast multi-dimension multi-choice knapsack heuristic for MP-SoC run-time management. *ACM Transactions on Embedded Computing Systems (TECS) 10*, 1–4.

200. Zhang, B. and Z. Hua (2012). Simple solution methods for separable mixed linear and quadratic knapsack problem. *Applied Mathematical Modelling 36*(7), 3245–3256.

201. Zhao, M., X. Gong, J. Liang, W. Wang, X. Que, Y. Guo, and S. Cheng (2017). QoE-driven optimization for cloud-assisted DASH-based scalable interactive multiview video streaming over wireless network. *Signal Processing: Image Communication 57*, 157–172.

202. Zhong, T. and R. Young (2010). Multiple choice knapsack problem: Example of planning choice in transportation. *Evaluation and Program Planning 33*, 128–137.

203. Zhu, H., Y. He, X. Wang, and E. C. C. Tsang (2017). Discrete differential evolutions for the discounted 0-1 knapsack problem. *International Journal of Bio-Inspired Computation 10*(4), 219–238.