European Journal of Operational Research 000 (2016) 1–14

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

A new exact approach for the 0–1 Collapsing Knapsack Problem

Federico Della Croce[a,*], Fabio Salassa[b], Rosario Scatamacchia[a]

[a] Dipartimento di Automatica e Informatica, Politecnico di Torino, Corso Duca degli Abruzzi 24, Torino 10129, Italy
[b] Dipartimento di Ingegneria Gestionale e della Produzione, Politecnico di Torino, Corso Duca degli Abruzzi 24, Torino 10129, Italy

ARTICLE INFO

Article history:
Received 15 January 2016
Accepted 1 December 2016
Available online xxx

Keywords:
Branch and Bound
Collapsing Knapsack Problem
0–1 Programming
Exact algorithm

ABSTRACT

We consider the 0/1 Collapsing Knapsack Problem (CKP) and a generalization involving more than a capacity constraint (M-CKP). We propose a novel ILP formulation and a problem reduction procedure together with an exact approach. The proposed approach compares favorably to the methods available in the literature and manages to solve to optimality very large size instances particularly for CKP and 2-CKP.

1. Introduction

The 0/1 Collapsing Knapsack Problem (CKP) can be seen as a generalization of the standard 0/1 Knapsack Problem (KP) where the capacity of the constraint is not a scalar but a non-increasing function of the number of included items, namely, it is inversely related to the number of items placed inside the knapsack. While KP has been widely studied, CKP has gained less attention. According to Posner and Guignard (1978), CKP has wide applications such as in satellite communication and time-sharing computer systems, namely in problems where a structural overhead is induced by the number of items or users considered. In a satellite communication, a correct transmission on the band requires that the parts of the band dedicated to each user must be separated by proper gaps. In time–sharing computer systems, just the adding of a process while other processes are running causes an overhead of the processing capabilities. Another application of interest is the transportation of fragile items, which may require additional coverings if they are transported with other items. These and similar real-life applications can be modeled as a Collapsing Knapsack Problems where the non-increasing function of the capacity represents the overhead of the resources produced by the number of items included in a solution.

CKP was first introduced in Posner and Guignard (1978), where an implicit enumeration algorithm was proposed. An exact algorithm making use of new upper and lower bounds was presented in Fayard and Plateau (1994). In Pferschy, Pisinger, and Woeginger (1997), a pseudo-polynomial time dynamic programming approach was proposed together with a reduction scheme to the standard knapsack problem. An improved reduction scheme is proposed in Iida and Uno (2002). Finally, an exact algorithm based on the partitioning of the CKP into sub-problems was presented in Wu and Srikanthan (2006).

The contribution of the paper is twofold. On the one hand, we present a novel ILP formulation of CKP and an effective reduction procedure for restricting the solution space of the problem. We remark that our novel ILP formulation, despite its simplicity, provides a significant contribution for tackling the CKP since it makes possible to exploit the potentials of the modern IP solvers. The previous formulation of the problem is not linear and the reduction schemes to a standard KP illustrated in Pferschy et al. (1997) and Iida and Uno (2002) induce very large size coefficients that make the KP very difficult to solve in practice.

On the other hand, we introduce an exact approach for CKP which is also extended to the multidimensional variant of CKP denoted hereafter M-CKP (with $M > 1$ indicating the number of capacity constraints). To the best of the authors' knowledge, no work has been developed to tackle collapsing knapsack problems with more than one capacity constraint. We propose a new exact approach that relies on the ILP formulation of CKP and on an original branching scheme that induces the solution of several KPs (with the additional constraint that the number of items in the knapsack is fixed) by exploiting the particular structure of CKP. The proposed approach is capable of solving to optimality all instances with up to 100,000 items within a time limit of 600 seconds, while instances tackled in the literature until now were limited to 1000 items in size. The exact approach is capable of solving to optimality

* Corresponding author.
E-mail address: federico.dellacroce@polito.it (F. Della Croce).

http://dx.doi.org/10.1016/j.ejor.2016.12.009
0377-2217/

also 2-CKP in instances up to 100,000 items in reasonable time. For M-CKP with M = 3,4,5, the proposed approach is capable of solving to optimality all instances with up to 12000, 1500 and 1000 respectively within a time limit of 3600 seconds.

The paper is organized as follows. In Section 2, the new integer linear programming formulation of the problem is provided together with its generalization to the multidimensional CKP. In Section 3 the reduction procedure is introduced. Section 4 is devoted to the description of the proposed exact solution algorithm. In Section 5, computational results for CKP and M-CKP are presented. Section 6 concludes the paper with final remarks.

2. ILP modeling of CKP and M-CKP

In this section, we briefly recall the original formulation of CKP and we investigate the effectiveness of two schemes laid out in the literature for reducing CKP to the classical knapsack problem. After that, we present our novel ILP formulation and its extension to the multidimensional variant of the problem.

2.1. A previous formulation of CKP

According to Wu and Srikanthan (2006), the 0/1 Collapsing Knapsack Problem (CKP) can be expressed as follows:

$$\text{maximize} \quad \sum_{i=1}^{n} p_i x_i \tag{1}$$

$$\text{subject to} \quad \sum_{i=1}^{n} w_i x_i \le B\left(\sum_{i=1}^{n} x_i\right) \tag{2}$$

$$x_i \in \{0,1\} \quad \forall i = 1,\ldots,n \tag{3}$$

where p_i and w_i, positive integers, denote *profit* and *weight* of each item i. Function $B(\cdot)$ is non-increasing over $\{1, 2, \ldots, n\}$, indicating the capacity of the knapsack. This implies that the capacity will decrease while the number of the items inserted into the knapsack increases. Each binary variable x_i indicates if item i is selected or not. CKP is known to be weakly NP-hard as it is pseudo-polynomially solvable (Pferschy et al., 1997) and contains KP as special case (when function $B(\cdot)$ is a constant). Naively, according to model (1)–(3), one can enumerate all possible sub-problems by iteratively fixing the total number of items and taking the corresponding capacity value from the function $B(\cdot)$. Then, one can solve all the sub-problems and consider the best solution among them. This task corresponds for each sub-problem to solve a KP problem with the additional constraint that the number of items in the knapsack is fixed, which is generally handled well by simply using a standard ILP solver. However, this approach is not expected to be effective as soon as the number of variables, and correspondingly the number of possible capacity values, increases.

2.2. Reduction schemes of CKP to a standard KP

In Pferschy et al. (1997) a reduction scheme of CKP to a pure KP is proposed. That reduction relies on doubling the number of variables and introducing coefficients of very large size. The authors of Pferschy et al. (1997), however, indicated that the presence of very large coefficients made those KP intractable in practice. Indeed, in Martello, Pisinger, and Toth (1999), COMBO, the current state-of-the-art algorithm for KP, was limited to instances with up to 200 items only, as the generation of larger instances was not possible on the machine used by the authors. Preliminary computational tests on that reformulation confirmed this fact. We considered CKP instances with 1000 items generated as the largest instances in Pferschy et al. (1997). Then, we launched CPLEX 12.5 for solving the standard knapsack problem produced by the reduction scheme: a CPU time limit of 1200 seconds was not sufficient to reach the optimal solution. An improved reduction scheme is proposed in Iida and Uno (2002). The scheme produces coefficients smaller than those in Pferschy et al. (1997). We tested this scheme by CPLEX 12.5 as well. Such a scheme was able to solve to optimality the 1000-item instances in few seconds but was not able to reach the optimum on instances with 2000 items, within a CPU time limit of 1200 seconds.

In the light of these considerations, the reduction schemes in the literature are not appealing in practice. We propose instead, in the following, a new ILP formulation of CKP and M-CKP that constitutes the basic element of the proposed solution approach.

2.3. A novel ILP formulation

It is possible to formulate explicitly function $B(\cdot)$ in CKP, by adding a new set of 0–1 variables and two constraints in order to deal with the non-increasing property of function $B(\cdot)$. Let us denote by $b_j = B(j)$ $(j = 1,\ldots,h)$ the h (with $h \le n$) possible *capacity values* associated with the sum of selected items x_i. We introduce h 0–1 variables y_j indicating whether a specific *capacity value j* is selected or not. Then, we have the following ILP model for CKP:

$$\text{maximize} \quad \sum_{i=1}^{n} p_i x_i \tag{4}$$

$$\text{subject to} \quad \sum_{i=1}^{n} w_i x_i \le \sum_{j=1}^{h} b_j y_j \tag{5}$$

$$\sum_{j=1}^{h} y_j = 1 \tag{6}$$

$$\sum_{i=1}^{n} x_i = \sum_{j=1}^{h} j y_j \tag{7}$$

$$x_i \in \{0,1\} \quad \forall i = 1,\ldots,n \tag{8}$$

$$y_j \in \{0,1\} \quad \forall j = 1,\ldots,h. \tag{9}$$

Constraint (5) links the weighted sum of the items to all possible capacity values. Constraint (6) ensures that exactly one *capacity value* is selected. Constraint (7) relates an index j to the total number of items selected in the solution. In other words if a variable y_j is selected, then we guarantee that the capacity value b_j corresponds to the associated number of selected items. Finally in (8) and (9) variables x_i and y_j are defined as binary.

We remark that a simple upper bound on the total number of the items present in the optimal solution can be straightforwardly computed. Let us denote by means of square brackets the set of weights sorted in nondecreasing order so that $w_{[1]} \le w_{[2]} \le \cdots \le w_{[n]}$. The following Property which is a reformulation of Proposition 5 in Pferschy et al. (1997) holds.

Property 1. *(Proposition 5 in Pferschy et al. (1997)) Let* $k^* = \min\{k \mid \sum_{j=1}^{k} w_{[j]} > b_k\}$. *Then, without loss of optimality,* $y_l = 0 \;\; \forall l \in [k^*, \ldots, h]$.

From Property 1, the following constraint is added without loss of generality to (namely the relevant variables are deleted from) model (4)–(9).

$$y_l = 0 \;\; \forall l \in [k^*, \ldots, h] \tag{10}$$