European Journal of Operational Research 252 (2016) 761–774

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

An exact decomposition algorithm for the generalized knapsack sharing problem

Isma Dahmani, Mhand Hifi, Lei Wu*

EPROAD-EA 4669, Université de Picardie Jules Verne, 7 rue du Moulin Neuf, 80000 Amiens, France

ARTICLE INFO

Article history:
Received 29 April 2015
Accepted 6 February 2016
Available online 12 February 2016

Keywords:
Exact decomposition
Knapsack
Sharing
Reduction
Upper bound

ABSTRACT

This paper presents an exact algorithm for solving the knapsack sharing problem with common items. In literature, this problem is also denominated the Generalized Knapsack Sharing Problem (GKSP). The GKSP is NP-hard because it lays on the 0–1 knapsack problem and the knapsack sharing problem. The proposed exact method is based on a rigorous decomposition technique which leads to an intense simplification of the solution procedure for the GKSP. Furthermore, in order to accelerate the procedure for finding the optimum solution, an upper bound and several reduction strategies are considered. Computational results on two sets of benchmark instances from literature show that the proposed method outperforms the other approaches in most instances.

1. Introduction

This paper presents an exact algorithm for solving the knapsack sharing problem with common items (see e.g., Fujimoto & Yamada, 2006). In literature, this problem is also denominated the Generalized Knapsack Sharing Problem (GKSP). As a generalization of the knapsack sharing problem, which has attracted wide attention in the context of the fair distribution of resources, the GKSP appears to be more valuable in real-world applications. This is due to the main consideration of the GKSP, which assumes that all agents have common interests while each of them searches for his own benefits. Such a system could be helpful for developing the incentive mechanism design for mobile phone sensing (see e.g., Zhang et al., 2014).

An instance of the GKSP is characterized by a fixed knapsack capacity C_{gksp} and a set $\mathcal{N}$ of $n+1$ disjoint subsets $\mathcal{N}_0,\ \mathcal{N}_1, \ldots,\ \mathcal{N}_n$, such that $\mathcal{N} = \cup_{i=0}^{n}\mathcal{N}_i$ and $\mathcal{N}_i \cap \mathcal{N}_j = \emptyset$ for $(i \neq j)$. The first subset $\mathcal{N}_0$ denotes the set of common items whereas all other subsets $\mathcal{N}_i,\ i = 1, \ldots, n$, denotes the set of uncommon (individual) items. Each item i belonging to the set $\mathcal{N}_i,\ i = 0, \ldots, n$, is characterized by its profit and weight. In this paper, we assume that the profits and weights are positive integers. The objective of the problem is to determine a set of items that maximizes the minimal value of a set of linear functions under the capacity.

Let P_{kp} be a 0–1 Knapsack Problem (KP) defined on the set of common items (i.e., $\mathcal{N}_0$) and P_{ksp} be the Knapsack Sharing Problem (KSP) defined on the set of uncommon items (i.e., $\cup_{i=1}^{n}\mathcal{N}_i$). Assume that $\mathcal{N}_0$ contains m_0 items and $\mathcal{N}_i$ contains m_i items, for $i = 1, \ldots, n$. Therefore, P_{kp} and P_{ksp} can be formally defined as follows:

$$(P_{kp}) \quad Z_{kp} = \max\ \mathbf{p_0} \cdot \mathbf{x_0} \quad \text{s.t.}\ \ \mathbf{w_0} \cdot \mathbf{x_0} \le C_{kp}, \quad \mathbf{x_0} \in \{0, 1\}^{m_0}$$

and

$$(P_{ksp}) \quad Z_{ksp} = \max \min_{1 \le i \le n} \{\mathbf{p_i} \cdot \mathbf{x_i}\} \quad \text{s.t.}\ \ \sum_{i=1}^{n} \mathbf{w_i} \cdot \mathbf{x_i} \le C_{ksp}, \quad \mathbf{x_i} \in \{0, 1\}^{m_i},\ \forall\ i = 1, \ldots, n,$$

where C_{kp} (resp. C_{ksp}) is the capacity, $\mathbf{p_0}$ (resp. $\mathbf{p_i}$, $\forall\ i = 1, \ldots, n$) is the profit vector related to the common (resp. uncommon) items, $\mathbf{w_0}$ (resp. $\mathbf{w_i}$, $\forall\ i = 1, \ldots, n$) is the weight vector related to the common (resp. uncommon) items, and $\mathbf{x_0}$ is the variable vector (resp. $\mathbf{x_i}$, $\forall\ i = 1, \ldots, n$) of P_{kp} (resp. P_{ksp}). Based on mathematical models of P_{kp} and P_{ksp}, P_{gksp} can be written as follows:

$$(P_{gksp}) \quad Z_{gksp} = \max \min_{1 \le i \le n} \{\mathbf{p_i} \cdot \mathbf{x_i}\} + \mathbf{p_0} \cdot \mathbf{x_0} \quad \text{s.t.}\ \ \sum_{i=1}^{n} \mathbf{w_i} \cdot \mathbf{x_i} + \mathbf{w_0} \cdot \mathbf{x_0} \le C_{gksp}, \quad \mathbf{x_0} \in \{0, 1\}^{m_0},\ \ \mathbf{x_i} \in \{0, 1\}^{m_i},\ \forall\ i = 1, \ldots, n,$$

where $C_{gksp} = C_{kp} + C_{ksp}$. Indeed, P_{gksp} can be recognized as a combination of P_{kp} and P_{ksp}. From this property, the rest of this

* Corresponding author. Tel.: +33 322827543.
E-mail address: lei.wu@u-picardie.fr, wulei.wenlan@gmail.com (L. Wu).

http://dx.doi.org/10.1016/j.ejor.2016.02.009
0377-2217/

paper is devoted to propose some decomposition and reduction techniques for the GKSP. It implies that, instead of directly addressing P_{gksp}, we try to find the optimum solution of the GKSP by solving a series of P_{kp} and P_{ksp}.

2. Related work

GKSP can be viewed as a blend of two well-known combinatorial optimization problems: the 0–1 Knapsack Problem (KP) and the Knapsack Sharing Problem (KSP). As introduced in Karp (1972), the KP is one of the classical NP-hard problems, which has been widely studied in literature. For further details, the reader can refer to Martello and Toth (1990b), Kellerer, Pferschy, and Pisinger (2004). Recently, in Rooderkerk and van Heerde (2016), authors showed that the KP can be also investigated in the retail assortment optimization problem in order to balance the risk and the return of assortments. In literature, two well known exact algorithms have been proposed for the KP: (i) the dynamical programming algorithm (see e.g., Horowitz & Sahni, 1974) and the branch-and-bound algorithm (see e.g., Pisinger, 1997). Based on these two approaches, some hybrid methods have been developed for solving complex optimization problems belonging to the knapsack family (see e.g., Martello & Toth, 1984; Pisinger, Martello, & Toth, 1999).

KSP has been first studied by Brown (1979), where its binary version becomes NP-hard since it represents an intuitive generalization of the KP or the maximum independent set problem (Hifi & M'Hallah, 2012; Kellerer et al., 2004). Yamada, Futakawa, and Kataoka (1998) proposed different exact algorithms based on the branch-and-bound algorithm and the dichotomous search for solving the KSP. Hifi and Sadfi (2002) and Hifi, M'Halla, and Sadfi (2005) designed several dynamic programming algorithms to accelerate the resolution process. In order to enhance the performance of the dynamic programing, Boyer, Baz, and Elkihel (2011) proposed several improved dominance notions based on solving a series of instances of the KP. Meanwhile, Hifi and M'Halla (2010) proposed a tree search-based approach to improve the performance of the branch-and-bound algorithm. Recently, Hifi and Wu (2014) developed an upper bound and an efficient dichotomous exact method for the KSP. The proposed method applies an exact decomposition strategy, where the original problem is decomposed into a series of minimization and maximization knapsack problems. For the large complex KSP, we cite a hybrid metaheuristic proposed in Haddar, Khemakhem, Rhimi, and Chabchoub (2014), which applies a quantum particle swarm optimization approach to find approximate solutions for the KSP. The results showed that the proposed method was able to provide high-quality solutions within a reasonable time.

To our knowledge, few papers are available in literature about the exact solution of the GKSP. Among these papers, we cite the paper of Fujimoto and Yamada (2006) in which an exact enumeration method was designed. The method consists of determining the optimum solution by enumerating all possible values related to the capacity of the KP and the KSP. For solving the KP, Horowitz and Sahni's (1974) exact algorithm is used whereas Yamada et al.'s (1998) exact method is applied for solving the relevant KSP. In Fujimoto and Yamada (2006), the authors underlined the effectiveness of the method on uncorrelated and weakly correlated instances. Nevertheless, it failed to find optimum solutions for large strongly correlated instances. In Haddar, Khemakhem, Hanafi, and Wilbaut (2015), a quantum particle swarm optimization was elaborated to approximately solve the GKSP. The provided results showed that the proposed approach was able to efficiently find the high-quality solutions in most cases, especially in the strongly correlated case.

As shown in Conejo, Castillo, Minguez, and Garcia-Bertrand (2006) and Raidl (2015), decomposition based techniques achieved great success in reducing computational effort for solving complex combinatorial optimization problems. Therefore, this paper addresses an exact decomposition technique for finding the optimum solution of the GKSP. The provided subproblems are solved by using the most recently available strategies. Furthermore, a new upper bound and several reduction strategies are introduced for accelerating the convergence of optimum solutions of the GKSP.

The remainder of the paper is organized as follows. Section 3 begins by summarizing the principle of the proposed exact method. Section 3.1 introduces a new upper bound used to curtail the search process. Section 3.2 discusses some strategies used to determine and improve lower bounds for the GKSP. Section 3.3 gives an overview of the proposed exact method. In Section 4, the performance of the proposed exact method is evaluated and analyzed on a variety of instances from literature. The obtained results are compared with those of the most recent approach (Haddar et al., 2014), the Cplex solver (version 12.6) and the best exact method available in literature (Fujimoto & Yamada, 2006).

3. A decomposition method for the GKSP

In this section, we present an algorithm for optimally solving the GKSP. The principle of the algorithm is based on enumerating all possible combinations between the capacity setting of the KP and that of the KSP. Let $Z_{kp}(C_{kp})$ (resp. $Z_{ksp}(C_{ksp})$) be the optimum objective value of the KP (resp. the KSP) for a given capacity C_{kp} (resp. C_{ksp}). According to P_{gksp}, P_{ksp} and P_{kp} given in Section 1, an available objective value (a valid lower bound) for the GKSP, namely L_{gksp}, can be written as follows:

$$L_{gksp}(C_{kp}) = Z_{kp}(C_{kp}) + Z_{ksp}(C_{gksp} - C_{kp}) \tag{1}$$

where C_{kp} is a positive integer which denotes the capacity of the relevant 0–1 knapsack problem P_{kp}. Note that, such a value has been already defined as a discontinuity point in Fujimoto and Yamada (2006). In the rest of this section, our study focuses on the use of evaluation and reduction strategies for finding a target value $C^{\star}_{kp}$ such that $L_{gksp}(C^{\star}_{kp})$ is equal to Z_{gksp}, where Z_{gksp} represents the optimum objective value of P_{gksp}.

3.1. Upper bound for the GKSP

The development of upper bounds plays a central role in improving the performance of exact methods used to solving maximization problems, such as branch-and-bound based algorithms. The effectiveness of an upper bound is measured by two criteria: its objective value and the required computational effort. A tighter upper bound can usually induce faster convergence toward the optimum solution. Meanwhile, the runtime required to compute the upper bound must be acceptable. Therefore, in this section, we introduce an upper bound of P_{gksp} based on using certain special properties of the GKSP.

An integer linear program for P_{gksp}, denoted by LP_{gksp}, can be formally written as follows:

$$(LP_{gksp}) \qquad Z_{gksp} = \max \ \gamma + \mathbf{p_0} \cdot \mathbf{x_0}$$

$$\text{s.t.} \qquad \mathbf{w_0} \cdot \mathbf{x_0} \le C_{kp} \tag{2}$$

$$\sum_{i=1}^{n} \mathbf{w_i} \cdot \mathbf{x_i} \le C_{gksp} - C_{kp} \tag{3}$$

$$\mathbf{p_i} \cdot \mathbf{x_i} \ge \gamma, \qquad \forall\ i = 1, \ldots, n \tag{4}$$

$$C_{kp} \in \mathbb{N} \tag{5}$$

$$\gamma \in \mathbb{N},\ \ \mathbf{x_0} \in \{0,1\}^{m_0},\ \ \mathbf{x_i} \in \{0,1\}^{m_i},\ \ \forall\ i = 1, \ldots, n. \tag{6}$$