Expert Systems with Applications 42 (2015) 4653–4666

Contents lists available at ScienceDirect

Expert Systems with Applications

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

A hybrid heuristic for the 0–1 Knapsack Sharing Problem

Boukthir Haddar [a], Mahdi Khemakhem [a,*], Saïd Hanafi [b,c,d], Christophe Wilbaut [b,c,d]

[a] LOGIQ ISGI, University of Sfax, Sfax, Tunisia
[b] UVHC, LAMIH, F-59313 Valenciennes, France
[c] CNRS, UMR 8201, F-59313 Valenciennes, France
[d] Univ Lille Nord de France, F-59000 Lille, France

ARTICLE INFO

Article history:
Available online 7 February 2015

Keywords:
Hybrid method
Relaxation
Knapsack problem
Particle swarm optimization

ABSTRACT

The Knapsack Sharing Problem (KSP) is a variant of the well-known NP-hard knapsack problem that has received a lot of attention from the researches as it appears into several real-world problems such as allocating resources, reliability engineering, cloud computing, etc. In this paper, we propose a hybrid approach that combines an Iterative Linear Programming-based Heuristic (ILPH) and an improved Quantum Particle Swarm Optimization (QPSO) to solve the KSP. The ILPH is an algorithm conceived to solve 0–1 mixed integer programming. It solves a series of reduced problems generated by exploiting information obtained through a series of linear programming relaxations and tries to improve lower and upper bounds on the optimal value. We proposed several enhancements to strengthen the performance of the ILPH: (i) New valid constraints are introduced to speed up the resolution of reduced problems; (ii) A local search is incorporated as an intensification process to reduce the gap between the upper and the lower bounds. Finally, QPSO is launched by using the k best solutions encountered in the ILPH process as an initial population. The proposed QPSO explores feasible and infeasible solutions. Experimental results obtained on a set of problem instances of the literature and other new harder ones clearly demonstrate the good performance of the proposed hybrid approach in solving the KSP.

1. Introduction

In this paper we consider the 0–1 Knapsack Sharing Problem (KSP) introduced by Brown (1979), which is a max–min optimization problem with a knapsack constraint. This problem has a wide range of commercial or industrial applications and occurs when resources have to be shared or distributed fairly to several entities (see, for instance, Brown (1979) and Tang (1988)). In the 0–1 KSP, we deal with a set $N = \{1, \ldots, n\}$ of n items. Each item $j \in N$ yields v_j units of profit and consumes a given amount of resource w_j as for the standard 0–1 knapsack problem. The set N is decomposed into m disjoint classes of items: i.e. for each couple (i,j), $i \neq j$, $i \leqslant m$ and $j \leqslant m$, $N_i \cap N_j = \emptyset$ and $N = \bigcup_{i=1}^{m} N_i$. A linear function is associated with each class of items. The objective of the 0–1 KSP is to determine the subset of items to put in a knapsack of capacity c in order to maximize the minimal value of the set of m linear functions subject to a single linear knapsack constraint. In the following, we assume that all the data are non-negative. Let x_j be a binary variable, with $x_j = 1$ if the item j is added in the knapsack, and $x_j = 0$ otherwise. The 0–1 KSP can be formulated as a max–min problem (KSP^{mm}):

$$(\boldsymbol{KSP^{mm}})\begin{cases} \max & \min\limits_{1\leqslant i\leqslant m}\left\{\sum\limits_{j\in N_i} v_j x_j\right\} \\ \text{subject to} & \sum\limits_{j\in N} w_j x_j \leqslant c \\ & x_j \in \{0,1\}, \quad \forall\, j \in N \end{cases}$$

A possible way to reformulate the problem KSP^{mm} as a Mixed Integer Program (MIP) consists in introducing an auxiliary continuous variable z representing the objective of the problem. Specifically, the problem KSP^{mm} can be equivalently reformulated as the following 0–1 MIP:

$$(\boldsymbol{KSP})\begin{cases} \max & z \\ \text{s.t.} & \sum\limits_{j\in N_i} v_j x_j \geqslant z, \ \forall\, i = 1,\ldots,m \\ & \sum\limits_{j\in N} w_j x_j \leqslant c \\ & z \geqslant 0 \\ & x_j \in \{0,1\}, \ \forall\, j \in N \end{cases}$$

* Corresponding author.
E-mail addresses: boukthir.haddar@gmail.com (B. Haddar), mahdi.khemakhem@gmail.com (M. Khemakhem), said.hanafi@univ-valenciennes.fr (S. Hanafi), christophe.wilbaut@univ-valenciennes.fr (C. Wilbaut).

http://dx.doi.org/10.1016/j.eswa.2015.01.049
0957-4174/

The KSP is known to be NP-hard since it is a generalization of the standard 0–1 knapsack problem (KP) when we $m = 1$ (see, e.g., Martello and Toth (1990, 1997) or Kellerer, Pferschy, and Pisinger (2004)). An impressive number of references can be found for solving hard variants of knapsack problems. A significant number of papers are devoted to recent population-based algorithms. For instance, Wang, Wang, and Xu (2012) proposed a hybrid algorithm based on estimation of distribution algorithm to solve the multidimensional knapsack problem (MKP). More recently, Baykasoğlu and Ozsoydan (2014) proposed a firefly algorithm for solving the MKP in both static and dynamic environments. Alijla, Wong, Lim, Khader, and Al-Betar (2014) proposed an intelligent water drop algorithm for solving combinatorial optimization problems including the MKP. Finally, Changdar, Mahapatra, and Pal (2015) proposed an improved genetic algorithm to solve constrained knapsack problem in fuzzy environment.

Two variants of the KSP exist in the literature: the continuous KSP and the binary KSP. The continuous KSP has been extensively studied by Tang (1988), Pang and Yu (1989), Brown (1991), Kuno, Konno, and Zemel (1991), Luss (1992) or Yamada and Futakawa (1997). A large variety of exact and approximate methods have been developed and adapted specifically for this problem. On the contrary, the literature on the binary KSP is quite poor. Few exact algorithms have been developed for solving optimally this variant of the problem. Yamada, Futakawa, and Kataoka (1998) proposed a Branch-and-Bound (B&B) algorithm and a binary search algorithm to solve the binary KSP. The upper bounds required for the B&B algorithm were obtained by decomposing the problem into a series of single KP, and the lower bounds were calculated with a greedy heuristic. The B&B algorithm was tested on a collection of randomly generated instances divided into three different sets: the uncorrelated instances, the weakly correlated instances and the strongly correlated instances.

For the uncorrelated and the weakly correlated instances, the number of variables varied from 200 to 80,000 while it varies from 200 to 400 for the strongly correlated instances, whereas the number of classes was limited to 2. The B&B algorithm was able to solve most of the uncorrelated and weakly correlated instances with 5,000 variables, and only a few strongly correlated instances. The binary search algorithm was able to solve all the uncorrelated and weakly correlated instances with two classes, and the strongly correlated instances with 200 variables. Then, the authors tested this method when the number of classes was set to 3, 5 and 10, respectively. The approach was able to provide an optimal solution for all the uncorrelated and weakly correlated instances with 200 to 10,000 variables.

Hifi and Sadfi (2002) proposed an exact approach in which the original KSP is divided into a series of single knapsack problems. Then, each problem was solved via a dynamic programming procedure slightly modified from that proposed by Gilmore and Gomory (1966) and Martello and Toth (1990) for solving the 0–1 KP, to ensure the optimality and to reduce the computation time. Computational results showed the competitiveness of the proposed method compared to the alternative approaches in the literature. Indeed, the algorithm can solve some larger instances of the problem in a reasonable running time.

Hifi, M'Halla, and Sadfi (2005) presented an accelerated version of the previous algorithm. This new version was mainly composed of two phases. The first phase builds a set of *critical elements* using a greedy heuristic introduced in Hifi, Sadfi, and Sbihi (2002). A critical element is associated with a given class N_i of items and it corresponds to an estimation of the sub-capacity devoted to this class of items. Based on a dynamic programming technique, the knapsack problems associated with the current critical elements are solved and an alternate critical element is chosen to guide the search to a better solution. The process is repeated until the convergence of the method to an optimal solution. The computational results showed a significant advantage of the proposed approach over the previous version of the algorithm.

Hifi and M'Halla (2010) proposed another exact approach in which the main idea is to replace the dynamic programming phase in Hifi et al. (2002) with a specialized tree-search algorithm. This method is particularly efficient when the number of classes is small. The proposed algorithm realized an important average acceleration, in particular for the strongly correlated instances.

Recently, Boyer, Baz, and Elkihel (2011) proposed an algorithm based on a dynamic programming algorithm with a dominance technique in order to reduce the memory occupancy and, thus, to improve the performance of the dynamic programming methods proposed in Hifi and Sadfi (2002) and Hifi, M'Halla, and Sadfi (2005).

The proposed algorithm was tested on a set of uncorrelated and strongly correlated instances of the problem. The results showed that this approach was able to find an optimal solution for large instances in a reasonable running time and that it consumed less memory than the previous exact methods in the literature. However, the best results were obtained when the number of classes is relatively small.

Recently, Hifi and Wu (2014) proposed a new exact algorithm based on a dichotomous search method. The original KSP is decomposed into a series of minimization and maximization knapsack problems to generate lower and upper bounds, respectively. They proposed two versions of their dichotomous search algorithm to reduce iteratively the gap between lower and upper bounds.

A very limited number of heuristics have been proposed for obtaining optimal or near-optimal solutions for the binary KSP. Yamada and Futakawa (1997) extended an approach originally developed for the continuous KSP to the binary KSP. It is based on a greedy heuristic using the classical *efficiency value* v_j/w_j to order the items. The algorithm identifies the class such that the sum of its fixed-items profits in the solution is the smallest one. Then, the first compatible item of this class is added to the current solution, and the index of the class realizing the minimum over all the classes is updated. The process is repeated while it is possible to find an item to add in the knapsack. The computational experiments showed that the algorithm can provide solutions of good quality, especially for the uncorrelated instances.

Finally, Hifi et al. (2002) proposed a tabu search based algorithm to solve the binary KSP. A simple and a more advanced version of the method have been developed. In the first version, a depth parameter and a tabu list are used. In the second version, some intensification and diversification strategies are introduced. In the paper, the authors emphasized the speed of the first version and the effectiveness of the second version for the correlated and uncorrelated instances. Logically, the experimental study showed that the second version produced a larger number of optimal solutions but required important computational effort.

This short survey on the relevant literature reveals that a few exact algorithms have been proposed in the literature to solve the KSP. However, a major drawback of these approaches remains the temporal complexity when dealing with large instances. Furthermore, it discloses that heuristics and metaheuristics have received relatively less attention. Indeed, only two approximate algorithms have been proposed for solving the KSP. Finally and to the best of our knowledge, it can be observed that hybrid (or cooperative) algorithms which combines the desirable properties of different approaches to minimize their individual weaknesses have not been used up to now to deal with this problem.

In this paper we propose a new hybrid approach that combines an Iterative Linear Programming-based Heuristic (ILPH) with a relatively recent evolutionary computation technique, the Quantum