Information Sciences 369 (2016) 634–647

Contents lists available at ScienceDirect

Information Sciences

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

Exact and approximate algorithms for discounted {0-1} knapsack problem

Yi-Chao He[a], Xi-Zhao Wang[b,*], Yu-Lin He[b], Shu-Liang Zhao[c], Wen-Bin Li[a]

[a] College of Information Engineering, Hebei GEO University, Shijiazhuang 050031, Hebei, China
[b] Big Data Institute, College of Computer Science & Software Engineering, Shenzhen University, Shenzhen 518060, Guangdong, China
[c] College of Mathematics and Information Science, Hebei Normal University, Shijiazhuang 050024, Hebei, China

ARTICLE INFO

Article history:
Received 30 March 2016
Revised 14 June 2016
Accepted 15 July 2016
Available online 16 July 2016

Keywords:
Discounted {0-1} knapsack problem
Exact algorithm
Approximate algorithm
Dynamic programming
Particle swarm optimization

ABSTRACT

The **D**iscounted {0-1} **K**napsack **P**roblem (D{0-1}KP) is an extension of the classical 0-1 knapsack problem (0-1 KP) that consists of selecting a set of item groups where each group includes three items and at most one of the three items can be selected. The D{0-1}KP is more challenging than the 0-1 KP because four choices of items in an item group diversify the selection of the items. In this paper, we systematically studied the exact and approximate algorithms for solving D{0-1}KP. Firstly, a new exact algorithm based on the dynamic programming and its corresponding fully polynomial time approximation scheme were designed. Secondly, a 2-approximation algorithm for D{0-1}KP was developed. Thirdly, a greedy repair algorithm for handling the infeasible solutions of D{0-1}KP was proposed and we further studied how to use binary particle swarm optimization and greedy repair algorithm to solve the D{0-1}KP. Finally, we used four different kinds of instances to compare the approximate rate and solving time of the exact and approximate algorithms. The experimental results and theoretical analysis showed that the approximate algorithms worked well for D{0-1}KP instances with large value, weight, and size coefficients, while the exact algorithm was good at solving D{0-1}KP instances with small value, weight, and size coefficients.

1. Introduction

Knapsack **P**roblem (KP) [1,6,10] is a classical NP-hard problem in computer science, which has found many applications in various areas, such as business management, computational complexity, cryptology, and applied mathematics, and so on. KP has some extended versions, e.g., the unbounded KP, multiple-choice KP, and quadratic KP, etc. These KP variants [22,27,29,30,33] have been well studied and successfully solved with different techniques and methods in the past few decades.

Discounted {0-1} **K**napsack **P**roblem (D{0-1}KP) a latest variant of classical 0-1 KP, which was firstly proposed by Guldan [12] in 2007 and used the concept of *discount* to reflect the sales promotion in real business activities. D{0-1}KP is a 0-1 integer programming problem with $n+1$ inequality constraints including the *discount* constraints and knapsack capacity constraints, while there is only one inequality constraint in the classical 0-1 KP, where n is the number of item groups. Due to the better expressive capability to real business sales, D{0-1}KP has obtained a large number of applications in investment

* Corresponding author.
E-mail addresses: heyichao119@163.com (Y.-C. He), xzwang@szu.edu.cn, xizhaowang@ieee.org (X.-Z. Wang), dryulinhe@yahoo.com (Y.-L. He).

http://dx.doi.org/10.1016/j.ins.2016.07.037

decision and resource allocation. However, the algorithmic study on how to solve D{0-1}KP in a more effective way is rare. In [12], Guldan presented an exact algorithm to solve D{0-1}KP based on dynamic programming [3] and discussed how to use the heuristic algorithm to solve D{0-1}KP. Rong et al. [32] in 2012 solved D{0-1}KP by combining the special kernel of D{0-1}KP with exact algorithm proposed in [12]. Currently, the study on how to solve D{0-1}KP mainly focuses on the exact algorithm. As far as we know, there is no work which uses the approximate and evolutionary algorithms to solve D{0-1}KP.

Motivated by designing the high-performance and low-complexity algorithms for solving D{0-1}KP, we systematically studied the exact and approximate algorithms for D{0-1}KP in this article. The main contributions of this article included the following four aspects: (1) proposing a **N**ew **E**xact algorithm for D{0-1}KP (NE-DKP) with lower complexity than algorithm studied in [12] when the sum of value coefficients is less than knapsack capacity; (2) developing a fully **Poly**nomial-time approximate scheme (Poly-DKP) to simplify the aforementioned exact algorithm NE-DKP; (3) presenting a **2**-**App**roximation algorithm (App-2-DKP) for D{0-1}KP based on greedy strategy; and (4) providing a **P**article **S**warm **O**ptimization based **G**reedy **R**epair algorithm for D{0-1}KP (PSO-GRDKP). On four kinds of well-known instances from real applications, we tested the practical performances of proposed exact/approximate algorithms and analyzed their computation complexities and approximation rates. The experimental results and theoretical analysis showed that the approximate algorithms, i.e., Poly-DKP, App-2-DKP , and PSO-GRDKP, work well for the large scale D{0-1}KPs, while the exact algorithm, i.e., NE-DKP, is good at solving the small scale D{0-1}KPs.

The remainder of this article is organized as follows. In Section 2, we provide a preliminary of D{0-1}KP. In Section 3, we describe the new exact algorithm for D{0-1}KP. Sections 4 depicts three approximate algorithms for D{0-1}KP, respectively. In Section 5, we report experimental comparisons that demonstrate the feasibility and effectiveness of proposed exact and approximate algorithms. Finally, we give our conclusions and suggestions for further research in Section 6.

2. Preliminary

In this section, the definition, mathematical model, and existing exact algorithm of discounted {0-1} knapsack problem (D{0-1}KP) are described.

Definition 1 ((Discounted {0-1} knapsack problem) [12,32]**.** Given n item groups having 3 items and one knapsack with capacity C, where the items in the i-th ($i = 0, 1, \cdots, n-1$) item group are denoted as $3i$, $3i+1$, and $3i+2$. The value coefficients of $3i$, $3i+1$, and $3i+2$ are p_{3i}, p_{3i+1}, and $p_{3i+2} = p_{3i} + p_{3i+1}$, respectively. The weight coefficients of $3i$, $3i+1$, and $3i+2$ are w_{3i}, w_{3i+1}, and w_{3i+2}, where w_{3i+2} is the discounted weight, $w_{3i} + w_{3i+1} > w_{3i+2}$, $w_{3i+2} > w_{3i}$, and $w_{3i+2} > w_{3i+1}$. D{0-1}KP is to maximize the total value of items which can be put into the knapsack, where at most one item is selected from each item group and the sum of weight coefficients is less than knapsack capacity C.

Without loss of generality, we assume that the value coefficient p_k, weight coefficient w_k ($k = 0, 1, \cdots, 3n-1$), and knapsack capacity C are the positive integers, and $w_{3i+2} \le C$ ($i = 0, 1, \cdots, n-1$), $\sum_{i=0}^{n-1} w_{3i+2} > C$, then the mathematical model of D{0-1}KP is defined as [12]:

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

$$\text{s.t. } x_{3i} + x_{3i+1} + x_{3i+2} \le 1, i = 0, 1, \cdots, n-1, \tag{2}$$

$$\sum_{i=0}^{n-1} (x_{3i}w_{3i} + x_{3i+1}w_{3i+1} + x_{3i+2}w_{3i+2}) \le C, \tag{3}$$

$$x_{3i}, x_{3i+1}, x_{3i+2} \in \{0, 1\}, i = 0, 1, \cdots, n-1, \tag{4}$$

where, x_{3i}, x_{3i+1}, and x_{3i+2} represent whether the items $3i$, $3i+1$, and $3i+2$ are put into the knapsack: $x_k = 0$ indicates the item k ($k = 0, 1, \cdots, 3n-1$) is not in knapsack, while $x_k = 1$ indicates the item k is in knapsack. It is worth noting that a binary vector $X = (x_0, x_1, \cdots, x_{3n-1}) \in \{0, 1\}^{3n}$ is a potential solution of D{0-1}KP. Only if X meets both Eqs. (2) and (3), it is a feasible solution of D{0-1}KP.

D{0-1}KP obviously has the properties of optimal substructure and overlapping subproblem, thus the dynamic programming is an appropriate method to solve D{0-1}KP. Guldan in [12] provided the first dynamic programming based exact algorithm to solve D{0-1}KP. In order to distinguish Guldan's exact algorithm with our new exact algorithm, we abbreviate it as OE-DKP (**O**ld **E**xact algorithm for D{0-1}KP). The recursion formula and computational complexity analysis of OE-DKP are as follows.

Let $\mathrm{P} = \{p_k | k = 0, 1, \cdots, 3n-1\}$ and $\mathrm{W} = \{w_k | k = 0, 1, \cdots, 3n-1\}$ be the value set and weight set, respectively. For each item group $i' \in \{0, 1, \cdots, i\}$, at most one item is selected to put into knapsack. For all items in knapsack, $G[i, j]$ is the