Applied Soft Computing 41 (2016) 94–103

Contents lists available at ScienceDirect

Applied Soft Computing

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

Solving 0-1 knapsack problem by greedy degree and expectation efficiency

Jianhui Lv[a], Xingwei Wang[a,*], Min Huang[b], Hui Cheng[c], Fuliang Li[a]

[a] College of Computer Science and Engineering, Northeastern University, Shenyang 110819, China
[b] College of Information Science and Engineering, Northeastern University, Shenyang 110819, China
[c] School of Computing and Mathematical Sciences, Liverpool John Moores University, Liverpool L3 5UX, UK

ARTICLE INFO

Article history:
Received 26 December 2014
Received in revised form 2 November 2015
Accepted 29 November 2015
Available online 31 December 2015

Keywords:
Economics
Region partition
Greedy degree
Expectation efficiency
Parallel computing

ABSTRACT

It is well known that 0-1 knapsack problem (KP01) plays an important role in both computing theory and real life application. Due to its NP-hardness, lots of impressive research work has been performed on many variants of the problem. Inspired by region partition of items, an effective hybrid algorithm based on greedy degree and expectation efficiency (GDEE) is presented in this paper. In the proposed algorithm, initially determinate items region, candidate items region and unknown items region are generated to direct the selection of items. A greedy degree model inspired by greedy strategy is devised to select some items as initially determinate region. Dynamic expectation efficiency strategy is designed and used to select some other items as candidate region, and the remaining items are regarded as unknown region. To obtain the final items to which the best profit corresponds, static expectation efficiency strategy is proposed whilst the parallel computing method is adopted to update the objective function value. Extensive numerical investigations based on a large number of instances are conducted. The proposed GDEE algorithm is evaluated against chemical reaction optimization algorithm and modified discrete shuffled frog leaping algorithm. The comparative results show that GDEE is much more effective in solving KP01 than other algorithms and that it is a promising tool for solving combinatorial optimization problems such as resource allocation and production scheduling.

1. Introduction

Knapsack problem (KP) is known as a well-studied combinatorial optimization problem and it has been thoroughly studied in the past decades. Generally speaking, KP is classified into separable KP (SKP) in which the items can be split arbitrarily and 0-1 KP (KP01) in which the items cannot be split. KP01 is an important type of KP due to its NP-hardness [1] and it offers many practical applications such as capital budgeting, project selection, resource allocation, cutting stock, investment decision-making, etc. [2]. Therefore, more and more researchers have paid attention to the problem of KP01 optimization. Especially, Martello gave a comprehensive review with further discussions on techniques commonly used in solving KP01 [3]. Since KP01 has been proven to be NP-hard, the methods employed to solve KP01 have been divided into three categories, i.e., exact methods with the exact solutions, meta-heuristic methods and heuristic methods with the approximate solutions.

About exact methods, some related research has been carried out. In [4], dynamic programming algorithm was proposed to solve KP01. And then, Rong [5] developed dynamic programming algorithm according to state reduction and Figueira [6] used many complementary dominance relations to improve dynamic programming algorithm. Kolesar [7] proposed branch and bound algorithm to solve KP01 in 1967. Later, Horowitz [8], Martello [9], and Pisinger [10] respectively expanded branch and bound algorithm in 1974, 1981, and 1993. In [11], the procedure based on linear mathematical programming formulation was proposed. In [12], a Lagrangian decomposition based algorithm was proposed. In [13], an algorithm based on surrogate, Lagrangian, and continuous relaxations was proposed. In [14], an algorithm with a single continuous variable for KP01 was proposed. Although they can produce the optimal solutions in solving small-scale problems, these exact algorithms perform poorly when the scale comes to be large.

In recent decades, many computational intelligence methods [15] have been developed and regarded as meta-heuristic algorithms to solve KP01. In [16], a global harmony search algorithm

* Corresponding author. Tel.: +86 18512451179.
E-mail addresses: lvjianhui2012@163.com (J. Lv), wangxw@mail.neu.edu.cn (X. Wang), mhuang@mail.neu.edu.cn (M. Huang), H.Cheng@ljmu.ac.uk (H. Cheng), lifuliang207@126.com (F. Li).

http://dx.doi.org/10.1016/j.asoc.2015.11.045
1568-4946/

was proposed. In [17], Zhang proposed a harmony search algorithm while it adopted the parallel updating strategy rather than serial updating strategy. In [18], Kong proposed a simplified binary harmony search algorithm for large scale KP01. In [19], an algorithm based on amoeboid organism was proposed by transforming the longest path into the shortest path. In [20], a quantum-inspired artificial immune system for KP01 was proposed. In [21], an improved hybrid encoding cuckoo search algorithm was proposed. In [22], a shuffled frog leaping algorithm was proposed. In [23], a particle swarm-based algorithm was proposed. In [24], a human learning optimization algorithm was proposed. In [25], a DNA-based computing method was proposed with fast parallel molecular. Although meta-heuristics can effectively solve KP01, they have to undergo the complex iteration process and set different number of populations with different instances. With that said, it will increase the difficulty of solving KP01, for example, the number of populations for a certain KP01 instance is difficult to set. Furthermore, the complex iteration process is not adapted to the optimization of engineering problem.

It becomes increasingly popular for the researchers worldwide to apply heuristic techniques in the optimization problems [26]. Given this, there are a number of effective heuristics for solving KP01. In [27], a hybrid differential evolution approach was studied. In [28], a polyhedral study with disjoint cardinality constraints was proposed. In [29], a population-based incremental learning algorithm based on greedy strategy was presented. In [30], a chemical reaction optimization algorithm with greedy strategy was proposed. In [31], an efficient fully polynomial time approximate scheme for KP01 was proposed. In [32], an iterative rounding search based algorithm was proposed to solve KP01. In [33], a soccer league competition algorithm was proposed. In [34], a thermodynamical selection based discrete differential evolution method was presented. However, they still cost the high time complexity especially in [27–29] and [31]. Thus, an expectation efficiency based on economical model was studied [35] and the time complexity was $O(n)$, which suggested that the expectation efficiency model for solving KP01 is more effective compared to other existing heuristics. In this paper, the expectation efficiency will be studied by combining with greedy degree.

Although a large number of KP01s have been resolved successfully by these existing algorithms, some new and more difficult KP01s are always hidden in the real world. Thus, the research on KP01 should be further improved and developed. Furthermore, especially in the industry, it usually focuses on finding the good approximate solution rather than spending a lot of time for the exact solution. Under this context, heuristic methods for KP01 will play more important role than exact methods and meta-heuristic methods. Given this, the heuristic methods should be encouraged so that the optimization and application of KP01 can be enhanced.

Given the above consideration, a hybrid algorithm based on greedy degree and expectation efficiency, called GDEE, is proposed which is inspired by region partition of items to solve KP01. The main contributions of this paper are summarized as follows. (a) A greedy degree model inspired by greedy strategy is designed to select the first some items to put into knapsack early and these items are never removed from knapsack in the following operations. (b) A dynamic expectation efficiency strategy is proposed to select some remaining items to put into knapsack, meanwhile, one candidate objective function value is obtained. (c) A static expectation efficiency strategy is also presented as the benchmark to update the candidate objective function value, as a result, a number of new objective function values are generated. (d) To accelerate the update speed of objective function value, the parallel computing method is adopted. (e) Experimental results based on a large number of instances reveal that GDEE is correct, feasible, effective, and stable.

The rest of this paper is organized as follows. Section 2 presents the design of GDEE in detail. In Section 3, experimental results based on a large number of instances are reported. Finally, Section 4 concludes this paper and suggests potential future work.

2. Design GDEE for KP01

2.1. KP01 description

Given n items, where item i owns weight w_i and profit p_i, and a knapsack that holds a fixed capacity *cap*, the goal of KP01 is to load some possible items into knapsack so that the total profit of the selected items has the maximal value while the total weight of the items is not larger than *cap*. Mathematically, KP01 can be described as follows.

$$\begin{cases} Maximize \quad optp = \sum_{i=1}^{n} p_i x_i \\ s.t. \quad \sum_{i=1}^{n} w_i x_i \le cap \end{cases} \tag{1}$$

where *optp* means the objective function value and $x_i \in \{0, 1\}$. If x_i is 1, item i is in knapsack; otherwise, it is not in knapsack. Let $X = (x_1, x_2, \ldots, x_n)$ means the solution of KP01. Thus, the goal of KP01 is to find a possible X which maximizes *optp*.

2.2. Greedy degree

Let r_i denote the ratio of p_i and w_i. In this paper, n items are rearranged by r_i in descending order in the first place, which is also done under greedy strategy, backtracking algorithm, and dynamic programming algorithm. Thus, the arrangement will be executed before performing the following GDEE operations. It is well known that greedy algorithm is applied to solve KP01 at the beginning and the obtained solution is local optimal rather than the optimal. As a matter of fact, some items can be still loaded into knapsack from the perspective of greedy algorithm, in other words, the items are determinate while the remaining items are uncertain. Based on this consideration, the determinate items should be loaded into knapsack in advance and never be removed from knapsack. However, how many and which items are determinate? To answer this question, the concept of greedy degree is proposed as follows.

Definition 1. n items are rearranged, if the first m items can be always loaded into knapsack by greedy strategy, and then m is regarded as the size of greedy degree.

The objective function value, the total weight of items, and the number of items can be obtained by greedy algorithm. Let *Goptp*, *GW*, and *Q* represent them respectively, and the constraint conditions are shown as follows.

$$\begin{cases} \dfrac{\sum_{i=1}^{j} w_i}{GW} \wedge \dfrac{\sum_{i=1}^{j} p_i}{Goptp} \le \lambda \\ \dfrac{\sum_{i=1}^{j+1} w_i}{GW} \wedge \dfrac{\sum_{i=1}^{j+1} p_i}{Goptp} > \lambda \\ \dfrac{\sum_{i=1}^{n/2} w_i + \sum_{i=n/2+1}^{j} w_i}{GW} \wedge \dfrac{\sum_{i=1}^{n/2} p_i + \sum_{i=n/2+1}^{j} p_i}{Goptp} < \xi \\ \dfrac{\sum_{i=1}^{n/2} w_i + \sum_{i=n/2+1}^{j+1} w_i}{GW} \wedge \dfrac{\sum_{i=1}^{n/2} p_i + \sum_{i=n/2+1}^{j+1} p_i}{Goptp} \ge \xi \end{cases} \tag{2}$$

where λ and ξ are parameters, $\lambda > \xi$ and $\lambda, \xi \in (0, 1)$; j is the serial number of item; $X \wedge Y \le Z$ means that $X \le Z$ and $Y \le Z$. Right after this, the greedy degree algorithm is designed and described in Algorithm 1.