Computers and Mathematics with Applications 62 (2011) 2913–2930

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

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

A two state reduction based dynamic programming algorithm for the bi-objective 0–1 knapsack problem

Aiying Rong [a,*], José Rui Figueira [b,1], Margarida Vaz Pato [c]

[a] Cemapre (Center of Applied Mathematics and Economics), ISEG -Technical University of Lisbon, Rua do Quelhas 6, 1200-781 Lisboa, Portugal

[b] INPL, Ecole des Mines de Nancy, Laboratoire LORIA, Parc de Saurupt- CS 14 234, 54 052 Nancy Cedex, France

[c] Centro de Investigação Operacional, FC - University of Lisbon, ISEG - Technical University of Lisbon, Rua do Quelhas 6, 1200-781 Lisboa, Portugal

ARTICLE INFO

Article history:
Received 14 March 2011
Received in revised form 26 July 2011
Accepted 27 July 2011

Keywords:
Bi-objective knapsack instances
Multi-objective optimization
Dynamic programming
State reduction

ABSTRACT

In this paper, we present a dynamic programming (DP) algorithm for the multi-objective 0–1 knapsack problem (MKP) by combining two state reduction techniques. One generates a backward reduced-state DP space (BRDS) by discarding some states systematically and the other reduces further the number of states to be calculated in the BRDS using a property governing the objective relations between states. We derive the condition under which the BRDS is effective to the MKP based on the analysis of solution time and memory requirements. To the authors' knowledge, the BRDS is applied for the first time for developing a DP algorithm. The numerical results obtained with different types of bi-objective instances show that the algorithm can find the Pareto frontier faster than the benchmark algorithm for the large size instances, for three of the four types of instances conducted in the computational experiments. The larger the size of the problem, the larger improvement over the benchmark algorithm. Also, the algorithm is more efficient for the harder types of bi-objective instances as compared with the benchmark algorithm.

1. Introduction

The 0–1 knapsack problem (KP) is one of the most intensively studied NP-hard combinatorial optimization problems [1]. The multi-objective 0–1 KP (MKP) is a generalization and a natural extension of the single objective 0–1 KP by considering two or more objectives. The MKPs are frequently encountered in practice since multiple conflicting objectives are more appropriate to model real-world situations. Examples can be found in capital budgeting [2], selection of transportation investment alternatives [3], relocation issues arising in nature conservation, biology [4], selection of building renovation methods [5], environmental investments [6], and facility location [7].

The MKP is described below. Given n items and r profit objectives for each item, with the kth profit objective $c_j^k (k = 1, \ldots, r)$ and weight w_j for item j $(j = 1, \ldots, n)$ and a knapsack of capacity W, the problem is to select a subset of items whose total weight does not exceed W and whose total profit objectives are maximized in the Pareto sense. The MKP can be formulated as the following multi-objective integer linear programming model:

$$\text{“max”} \left(\sum_{j=1}^{n} c_j^1 x_j, \; \ldots, \; \sum_{j=1}^{n} c_j^r x_j \right) \tag{1}$$

* Corresponding author. Tel.: +351 213922747; fax: +351 213922782.
E-mail addresses: arong@iseg.utl.pt (A. Rong), Jose.Figueira@mines.inpl-nancy.fr (J.R. Figueira), mpato@iseg.utl.pt (M.V. Pato).

1 Associate member at CEG-IST, Lisboa, Portugal.

0898-1221/$ – see front matter
doi:10.1016/j.camwa.2011.07.067

subject to

$$\sum_{j=1}^{n} w_j x_j \leq W, \tag{2}$$

$$x_j \in \{0, 1\}, \quad j = 1, \ldots, n \tag{3}$$

where x_j are decision variables indicating whether the jth item is selected to place in the knapsack or not. Here, we follow the common assumptions in most literature: $W, w_j, c_j^k (j = 1, \ldots, n; k = 1, \ldots, r)$ are positive integers. To avoid trivial solutions, it is assumed that $w_j \leq W (j = 1, \ldots, n)$ and $\sum_{j=1}^{n} w_j > W$.

For the single objective 0–1 KP, decades of algorithmic improvements have made it possible to solve in a reasonable time limit nearly all standard instances from the literature. But some types of instances are still hard to solve because of its NP-hard nature. Ref. [8] pointed out that the strongly correlated instances (weight coefficients and profit coefficients are strongly correlated) and some other types of instances challenged the existing algorithms. Ref. [9] compared the solution times of all recent algorithms using classical and new benchmark test instances. The new benchmark set includes instances with large ($\geq 10^5$) or moderate (10^3) weight coefficients and all the algorithms based on currently used upper bound techniques showed bad performance on these instances. This study pointed out that dynamic programming (DP) is one of the best approaches for solving the hard types of the 0–1 KP.

In the multi-objective optimization context, the solution process consists of finding the Pareto frontier (PF) with a number of non-dominated objective vectors in the objective space which corresponds to efficient solutions in the decision space. Hence, compared with the single objective KP, the MKP poses more challenges. On the one hand, there are intractable instances of multi-objective combinatorial optimization problems, for which the number of efficient solutions is not polynomial in the size of their instances [10]. On the other hand, for most multi-objective combinatorial optimization problems, deciding whether a given objective vector is dominated or not is an NP-hard problem [11], even if the underlying single objective version can be solved in polynomial time.

However, these difficulties do not prevent the research effort from developing efficient algorithms able to find the PF quickly from the practical viewpoints. In the following, we review the main accurate approaches for solving the MKP. Ref. [12] presented the theoretical DP framework for the multi-objective integer KP. Ref. [13] implemented an exact algorithm for the bi-objective 0–1 KP (BKP) by exploring developments for the multi-objective linear programming problem. The above two research contributions can be considered as theoretical developments because the authors did not present extensive experimental results. Other researchers have developed specific algorithms based on extensive numerical tests. Most of this research work focused on calculating the PF with the exception of [14]. Ref. [14] presented a generic labeling algorithm, which calculated both the PF in the objective space and the corresponding efficient solutions in the decision space, for the multi-objective integer KP. Ref. [15] presented a two-phase branch and bound algorithm for the BKP. Ref. [16] presented a labeling algorithm for the BKP by transforming a KP into a shortest path problem. It is in essence a DP algorithm. Ref. [17] studied a dominance based on the DP (DDP) algorithm for the MKP and numerical tests were conducted for the BKP and tri-objective KP. Ref. [18] applied bound sets in the DP algorithm for the MKP and numerical tests were conducted for the BKP. Both [17,18] are hybrid DP algorithms.

Similar to the single-objective 0–1 KP, DP algorithms [16–18] are among the best approaches to solve the MKP. Ref. [16] attempted to reduce the number of states to be calculated by generating a forward reduced-state DP space (FRDS) but the computational effort for the final stage is very heavy. The DP space consists of all of the states and related transitions between states in the DP process. Ref. [17] relied on several dominance relations to discard partial solutions that cannot lead to new non-dominated objective vectors and Ref. [18] applied elaborate bounding techniques to reduce the number of states to handle in the DP process. This can significantly reduce the computational effort for the algorithm. However, applying dominance relations and bounding techniques still needs a heavy computational effort.

In this paper, we focus our attention on finding the PF for the MKP, i.e., the algorithm is designed to address the multi-objective case. But our implementation and numerical tests focus on the bi-objective case. We follow DP approaches for dealing with the MKP and use a backward reduced-state DP space (BRDS) to avoid heavy computational effort for the FRDS in the final stage. The major contributions of the paper are summarized as follows. First, we identify a BRDS by exploring the network of the basic sequential DP (BDP) process. Second, we derive the condition under which the BRDS is effective to the MKP based on the analysis of its impact on the solution time and memory requirements. Finally, we develop a new DP algorithm by applying the BRDS in conjunction with a property governing the objective relations between states, which can help to reduce further the number of states to be calculated in the BRDS. To our knowledge, it is the first time that the BRDS is used in the DP algorithm.

Some states, which have been discarded in our approach, may coincide with those discarded by the dominance relations in [17] or by bounding techniques [18]. However, the techniques used for discarding the states are completely different. Moreover, our techniques are especially efficient for the hard types of KP instances as compared with [17,18]. The hard types of instances include conflicting instances where the profit objectives are negatively correlated. The hardest instances are those where conflicting objectives are positively correlated with the weight coefficients. Usually the cardinality of the PF for these instances increases rapidly as the problem size increases. Very few techniques can handle these instances efficiently.

The paper is broken down as follows. In Section 2, we give the network representation of the multi-objective BDP process to provide the foundation for generating the BRDS. In Section 3, we outline the main components of a two state reduction