Computers and Mathematics with Applications 63 (2012) 1462–1480

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

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

# Computational performance of basic state reduction based dynamic programming algorithms for bi-objective 0–1 knapsack problems[☆]

Aiying Rong[a,*], José Rui Figueira[b,1]

[a] Cemapre (Center of Applied Mathematics and Economics), ISEG -Technical University of Lisbon, Rua do Quelhas 6, 1200-781 Lisboa, Portugal
[b] CEG-IST, Instituto Superior Técnico, Av. Rovisco Pais, 1049-001 Lisboa, Portugal

ARTICLE INFO

*Article history:*
Received 28 July 2011
Received in revised form 14 December 2011
Accepted 16 March 2012

*Keywords:*
Multi-objective optimization
Bi-objective knapsack problem
Dynamic programming
Basic state reduction techniques

ABSTRACT

This paper studies a group of basic state reduction based dynamic programming (DP) algorithms for the multi-objective 0–1 knapsack problem (MKP), which are related to the backward reduced-state DP space (BRDS) and forward reduced-state DP space (FRDS). The BRDS is widely ignored in the literature because it imposes disadvantage for the single objective knapsack problem (KP) in terms of memory requirements. The FRDS based DP algorithm in a general sense is related to state dominance checking, which can be time consuming for the MKP while it can be done efficiently for the KP. Consequently, no algorithm purely based on the FRDS with state dominance checking has ever been developed for the MKP. In this paper, we attempt to get some insights into the state reduction techniques efficient to the MKP. We first propose an FRDS based algorithm with a local state dominance checking for the MKP. Then we evaluate the relative advantage of the BRDS and FRDS based algorithms by analyzing their computational time and memory requirements for the MKP. Finally different combinations of the BRDS and FRDS based algorithms are developed on this basis. Numerical experiments based on the bi-objective KP instances are conducted to compare systematically between these algorithms and the recently developed BRDS based DP algorithm as well as the existing FRDS based DP algorithm without state dominance checking.



## 1. Introduction

The multi-objective 0–1 knapsack problem (MKP) can be viewed as an extension of the single objective 0–1 knapsack problem (KP) [1] to accommodate more than one objective [2]. The MKP can be defined as follows: given a knapsack of capacity $W$ and a set of $n$ items, each item associated with $r$ integer profits, denoted by an $r$-dimensional vector $(c_j^1, \ldots, c_j^r)$, and an integer weight $w_j, j = 1, \ldots, n$. The problem consists of selecting a subset of the items whose total weight does not exceed $W$ and whose profit objectives are "maximized" in the Pareto sense. The problem can be formulated as the following multi-objective linear integer programming model:

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

☆ The paper has been evaluated according to old Aims and Scope of the journal.

* Corresponding author. Tel.: +351 213922747; fax: +351 213922782.
*E-mail addresses:* arong@iseg.utl.pt (A. Rong), figueira@ist.utl.pt (J.R. Figueira).
1 Associate researcher at LORIA Laboratory, Nancy, France.

0898-1221/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2012.03.057

subject to:

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

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

Here binary variables $x_j$ are used to indicate whether item $j$ is included in the knapsack or not. In addition, 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 $\sum_{j=1}^{n} w_j > W$.

Several approaches have been proposed to solve the MKP, involving both approximate methods and exact algorithms. The approximate approaches mainly include the fptas (fully polynomial approximation scheme) [3] and metaheuristics [4–8]. The exact approaches include branch and bound (BB) algorithm [9], dynamic programming (DP) based algorithms [10–13], and other exact algorithms, for example, an exact algorithm by exploiting the development for the multi-objective linear programming [14].

In this paper, we revisit the DP based exact algorithms and attempt to identify the state reduction techniques efficient to the MKP convincingly based on studying the computational time and memory requirements of a group of DP based algorithms using different reduction techniques for the bi-objective KP (BKP) instances. The study is motivated by the following facts.

First, 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. Usually, the computational effort increases as the number of objectives increases [2,15]. It is a hard task to generate the exact PF. However, there is a growing need for efficient exact algorithms from both theoretical and practical viewpoints though the approximate approaches can generate a good approximation of the PF.

Second, based on numerical experiments, it is known that DP based algorithms are among the best exact approaches to solve the MKP. Especially, different DP algorithms show different solution time efficiency against different types of the BKP instances. The DP algorithm using labeling (LDP) approaches [10] outperforms BB based algorithm [9] and the dominance based DP (DDP) algorithm [11] outperforms the LDP algorithm. In practice, multiple conflicting objective instances, where the profit objectives are negatively correlated, are more appropriate to model the real-world situations and these instances are called hard instances defined in [11]. The bounding technique based DP (BDP-II) algorithm [12] is more efficient than the DDP algorithm [11] for the non-hard type of instances while less efficient for the hard type of instances. A recently developed two state reduction based DP (TDP) algorithm [13] performs better than the DDP algorithm especially for the hardest type of instances, where the two conflicting objectives are correlated with weight coefficients, while worse for the easy (quasi single objective) instances with two "unconflicting" profit objectives.

Finally, the TDP algorithm [13] differs from all the other DP based algorithms in that the former is based on a backward reduced-state DP space (BRDS) while the latter on the forward reduced-state DP space (FRDS). The DP space consists of all of the states and related transitions between the states in the DP process. The LDP algorithm [10] is based on a special type of FRDS without using state dominance relations. The DDP [11] and BDP-II [12] algorithms are hybrid FRDS based DP algorithms. Before the TDP algorithm, the BRDS has never been applied in developing a DP algorithm because it imposes disadvantage for the KP in terms of the memory requirements. The current algorithm development indicates that the BRDS and FRDS respond differently to the MKP.

The FRDS in a general sense is related to the state dominance relation. The state dominance relations can be defined as follows: assume that two states *s* and *s*' in the DP space represent the solutions to the same sub-problem, then state *s* dominates state *s*' if the total weight of items at state *s*' is not smaller than that at state *s* and total profit values at state *s*' are not larger than those at state *s*. The FRDS is generated by discarding the dominated states in the DP space and the remaining non-dominated states (nodes) are called sparse nodes. The FRDS based sparse node DP (SDP) algorithm has not be applied to solve the MKP though it was used to solve the 0–1 KP for more than three decades ago [16] and extended to solve the integer KP later [17,18] because the state dominance checking can be done efficiently for the KP. A standard SDP algorithm which discards all the dominated states can be inefficient for the MKP because the state dominance checking can be time-consuming when the number of objective vectors of a state is large and it is an NP-hard problem to determine whether a given objective vector is dominated or not [19]. Up to now, the usage of the state dominance relation for the MKP appeared only in a hybrid DP algorithm [11], where several dominance relations were applied together.

All the above mentioned DP algorithms make use of some techniques to reduce explicitly or implicitly the number of the states in the DP space. In this paper, we restrict our attention to the basic state reduction techniques used to generate BRDS and FRDS, including state dominance relations.

The major contributions of the paper can be summarized as follows. First, we propose a variant of multi-objective SDP algorithm with a local state dominance checking to fill the gap for the FRDS based algorithm using state dominance relations for the MKP. Second, we evaluate the relative computational advantage of the BRDS and FRDS based algorithms by conducting time and space analysis. Third, we propose new DP based algorithms by combining the FRDS and BRDS based algorithms in an innovative manner on this basis. Finally, we compare systematically the computational time and memory requirements of different algorithms by constructing the confidence intervals (using statistical inference [20]) so that the conclusions can be made convincingly in a general sense.