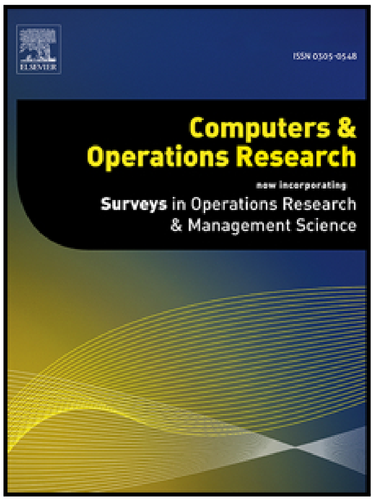
Accepted Manuscript



Compressed data structures for bi-objective {0,1}-knapsack problems

Pedro Correia, Luís Paquete, José Rui Figueira

PII: S0305-0548(17)30215-0
DOI: 10.1016/j.cor.2017.08.008
Reference: CAOR 4304

To appear in: *Computers and Operations Research*

Received date: 20 June 2016
Revised date: 11 August 2017
Accepted date: 13 August 2017

Please cite this article as: Pedro Correia, Luís Paquete, José Rui Figueira, Compressed data structures for bi-objective {0,1}-knapsack problems, *Computers and Operations Research* (2017), doi: 10.1016/j.cor.2017.08.008

Compressed data structures for bi-objective {0,1}-knapsack problems

Pedro Correia[a,b], Luís Paquete[a], José Rui Figueira[b]

[a] CISUC, Department of Informatics Engineering, University of Coimbra, Portugal
[b] CEG-IST, Instituto Superior Técnico, Universidade de Lisboa, Portugal

Abstract

Solving multi-objective combinatorial optimization problems to optimality is a computationally expensive task. The development of implicit enumeration approaches that efficiently explore certain properties of these problems has been the main focus of recent research. This article proposes algorithmic techniques that extend and empirically improve the memory usage of a dynamic programming algorithm for computing the set of efficient solutions both in the objective space and in the decision space for the bi-objective knapsack problem. An in-depth experimental analysis provides further information about the performance of these techniques with respect to the trade-off between CPU time and memory usage.

Keywords: Multi-objective optimization, implicit enumeration techniques

1. Introduction

Dealing with a large amount of solutions for further processing is a key concern in the field of multi-objective combinatorial optimization. Such processing include, for example, gathering or producing a collection of data sets within a limited memory (internal or external), extraction of important pieces of information from the whole data, manage the data, which deals with several operations, and process these operations in a reasonable amount of CPU-time. These aspects require the use of efficient data structures.

Solution methods for multiobjective combinatorial optimization (MOCO) problems typically require a large usage of memory resources. Parametric and recursive programming [1, 2], approximation methods [3], metaheuristics [4], or exact methods [5, 6, 7] are example of approaches that require more memory usage due to the large number of potential solutions that need to be kept during the search process. For instance, the experimental analysis reported in Figueira et al. [7] shows that more than five million solutions need to be kept in memory in order to solve bi-objective knapsack problem instances with less than one thousand items; see similar results reported in Paquete et al. [8] for a related

*Corresponding author
Email addresses: pamc@dei.uc.pt (Pedro Correia), paquete@dei.uc.pt (Luís Paquete), figueira@tecnico.ulisboa.pt (José Rui Figueira)