Available online at www.sciencedirect.com
ScienceDirect
Computers & Operations Research 35 (2008) 241 – 252
computers & operations research
www.elsevier.com/locate/cor

A cross entropy algorithm for the Knapsack problem with setups

M. Caserta*, E. Quiñonez Rico, A. Márquez Uribe

Instituto Tecnologico de Monterrey, Calle del Puente, 222, Col. Ejidos de Huipulco, Del. Tlalpan, México DF, 14380, México

Available online 2 May 2006

Abstract

In this article we propose a new metaheuristic-based algorithm for the Integer Knapsack Problem with Setups. This problem is a generalization of the standard Integer Knapsack Problem, complicated by the presence of setup costs in the objective function as well as in the constraints. We propose a cross entropy based algorithm, where the metaheuristic scheme allows to relax the original problem to a series of well chosen standard Knapsack problems, solved through a dynamic programming algorithm. To increase the computational effectiveness of the proposed algorithm, we use a turnpike theorem, which sensibly reduces the number of iterations of the dynamic algorithm. Finally, to testify the robustness of the proposed scheme, we present extensive computational results. First, we illustrate the step-by-step behavior of the algorithm on a smaller, yet difficult, problem. Subsequently, to test the solution quality of the algorithm, we compare the results obtained on very large scale instances with the output of a branch and bound scheme. We conclude that the proposed algorithm is effective in terms of solution quality as well as computational time.

Keywords: Combinatorial optimization; Metaheuristics; Knapsack Problem; Scheduling; Dynamic programming; Cross entropy

1. Introduction

The Integer Knapsack Problem with Setup is described as an integer Knapsack Problem with additional fixed costs of setup discounted both in the objective function and in the constraints. In the literature, it raises in two different contexts. First, it may be considered as a natural generalization of the pure integer knapsack problem. Secondly, it is viewed as a particular case of a well studied problem, the Multi-Item Capacitated Lot Sizing Problem with Setup Times, as presented in Miller et al. [1]. In both cases, the study of the Integer Knapsack Problem with Setups presented in this paper gives some insight on its importance as combinatorial problem. Moreover, it provides further insight of the difficulty of each one of these problems. In addition, very recently, a new application of the integer knapsack problem with setups has been proposed in Jacobson et al. [2], where one wants to optimize a baggage screening performance measure subject to a finite amount of resources.

Due to its vast industrial applicability, researchers have devoted special attention to the Multi-item Capacitated Lot sizing problem MCL with setup times (see Dzielinski and Gomory [3], Pochet and Wolsey [4], Eppen and Martin [5], and Constantino [6]), where one wants to find the optimal schedule of a set of items over a time horizon, with items competing for a shared capacity. The problem is complicated by the existence of setup costs as well as inventory

* Corresponding author. Tel.: +52 55 5483 2189; fax: +52 55 5483 2163.
E-mail addresses: marco.caserta@itesm.mx (M. Caserta), quinonez@itesm.mx (E. Quiñonez Rico), alberto.marquez@itesm.mx (A. Márquez Uribe).

0305-0548/$ - see front matter
doi:10.1016/j.cor.2006.02.028

costs. For this reason, a trade off exists between producing an item at every period, and, consequently, paying less inventory but more setup, or producing more sparingly, hence paying less setup but more inventory. Since the multi-item capacitated lot sizing problem with setups is very difficult to solve to optimality, many researchers have tried to tackle the problem by working on relaxations of the same. A good description of some well studied relaxations of MCL is provided by Miller et al. [1].

In the literature we can find two broad approaches to MCL, both based upon the definition of a relaxation of the original problem. On the one hand, some authors tackle the problem by relaxing the time dimension, which is, they define the single period relaxation, where only one period at a time is considered. On the other hand, other researchers propose a different kind of relaxation, based upon the conversion of the original, multi-item problem, to a single-item, multi-period lot sizing problem with setups.

Miller et al. [7] exploit the first kind of relaxation, based upon the reduction of the model to a single-period problem. They call this relaxation PI, for preceding inventory, since the model takes into account initial inventory levels for each item, inherited from the preceding period. The solution approach is based upon the study of the polyhedral structure of the model and the definition of cover inequalities for PI, which induce facets of the convex hull.

In a different study, Miller et al. [8] present a special case of PI, the multi-item single-period capacitated lot sizing problem with setups, called PIC, where setup times and demands are constant for all items. For this special case, the authors provide a polynomial time algorithm. They prove that the inequalities identified in Miller et al. [7] suffice to solve PIC by linear programming in polynomial time.

With respect to the second type of relaxation, the single item relaxation, some indications are provided in Pochet [9], Pochet and Wolsey [4] and Miller et al. [10]. They provide guidelines about how to derive valid inequalities for the single-item problem with the aim of applying the same inequalities to the multi-item problem.

Quite differently, Trigeiro et al. [11] tackle the problem by relaxing the capacitated lot sizing to an uncapacitated lot sizing problem, where the lagrangian relaxation of the capacity constraints is exploited. The original problem is decomposed into a set of uncapacitated single item lot sizing problem. The lagrangian dual costs are then updated by subgradient optimization and the single item problem is solved via dynamic programming.

In this paper we propose a different approach to MCL, somehow related to the single-period relaxation. However, rather than solving the single-period problem through a study of the polyhedral structure of the model, we reduce the single-period problem to a series of Integer Knapsack Problems with Setups. We effectively solve each Knapsack Problem by using a metaheuristic-based scheme, which allows to simplify the Knapsack Problem with Setups to a set of standard Knapsack Problems, easily solved via dynamic programming.

The paper has the following structure: Section 2 presents the formulation of the Integer Knapsack Problem with Setups, along with an explanation of how MCL can be reduced to a series of Knapsack Problems. Section 3 offers a brief introduction to the Cross-Entropy (CE) method and describes the proposed CE scheme for the Knapsack Problem. Section 4 presents the overall algorithm, while Section 5 offers an indication of the effectiveness of the algorithm by comparing the proposed algorithm with two well-known algorithms for the knapsack problem. Furthermore, we present computational results, both on small problems and on very large scale instances of the Knapsack Problem. Finally, Section 6 summarizes results and findings of the paper.

2. Integer Knapsack problem with setups

The Integer Knapsack Problem with Setups is concerned with the choice of the set of production items of maximum value, where a unitary value is associated to each item. Furthermore, each item takes some unitary amount of resources. In addition, a fixed amount of resources is consumed when the production line of an item is setup and, consequently, a fixed cost must be discounted from the overall production schedule value. Indeed, in the context of programming several items on a single machine with a limited amount of available time, switching among items takes some time to setup the machine, hence reducing the time available for actual production. Moreover, a fixed cost must be discounted from the total production value, since no item is produced while setting up the machine.

Let $y_j \in \mathbb{Z}_+$ indicate the number of items of type j chosen, with $j = 1, \ldots, n$. Let us set $x_j \in \mathbb{B}$ to 1 if at least one unit of item j is chosen and to 0 otherwise. A standard formulation of the Integer Knapsack Problem with Setups as a