European Journal of Operational Research 237 (2014) 440–447

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Upper and lower bounding procedures for the multiple knapsack assignment problem

Seiji Kataoka *, Takeo Yamada

Department of Computer Science, National Defense Academy, Yokosuka, Kanagawa 239-8686, Japan

ARTICLE INFO

Article history:
Received 19 November 2012
Accepted 4 February 2014
Available online 12 February 2014

Keywords:
Combinatorial optimization
Heuristics
Multiple knapsack problem
Assignment problem
Lagrangian relaxation

ABSTRACT

We formulate the multiple knapsack assignment problem (MKAP) as an extension of the multiple knapsack problem (MKP), as well as of the assignment problem. Except for small instances, MKAP is hard to solve to optimality. We present a heuristic algorithm to solve this problem approximately but very quickly. We first discuss three approaches to evaluate its upper bound, and prove that these methods compute an identical upper bound. In this process, *reference capacities* are derived, which enables us to decompose the problem into mutually independent MKPs. These MKPs are solved euristically, and in total give an approximate solution to MKAP. Through numerical experiments, we evaluate the performance of our algorithm. Although the algorithm is weak for small instances, we find it prospective for large instances. Indeed, for instances with more than a few thousand items we usually obtain solutions with relative errors less than 0.1% within one CPU second.

1. Introduction

This article is concerned with the *multiple knapsack assignment problem* (MKAP), as an extension of the *multiple knapsack problem* (MKP, Kellerer, Pferschy, & Pisinger, 2004; Martello & Toth, 1990; Pisinger, 1999), as well as of the *assignment problem* (Burkard, Dell'Amico, & Martello, 2009; Kuhn, 2005; Pentico, 2007), where we are given a set of n items $N = \{1, 2, \ldots, n\}$ to be packed into m possible knapsacks $M = \{1, 2, \ldots, m\}$. As in ordinary MKP, by w_j and p_j we denote the weight and profit of item $j \in N$ respectively, and the capacity of knapsack $i \in M$ is c_i. However, items are divided into K mutually disjoint subsets of items N_k $(k = 1, \ldots, K)$, thus we have $N = \cup_{k=1}^{K} N_k, n_k := |N_k|$, and $n = \sum_{k=1}^{K} n_k$. The problem is to determine the assignment of knapsacks to each subset, and fill knapsacks with items in that subset, so as to maximize the total profit of accepted items. To formulate this mathematically, we introduce binary decision variables x_{ij} and y_{ik} such that $x_{ij} = 1$ if item j is included in knapsack i, and $x_{ij} = 0$ otherwise. Also, $y_{ik} = 1$ if we assign knapsack i to subset N_k, and $y_{ik} = 0$ otherwise. Then, we have the following.

MKAP:

$$\text{maximize} \quad z(\boldsymbol{x}, \boldsymbol{y}) := \sum_{i=1}^{m} \sum_{k=1}^{K} \sum_{j \in N_k} p_j x_{ij}, \tag{1}$$

$$\text{subject to} \quad \sum_{j \in N_k} w_j x_{ij} \leqslant c_i y_{ik}, \quad i = 1, \ldots, m, \quad k = 1, \ldots, K, \tag{2}$$

$$\sum_{i=1}^{m} x_{ij} \leqslant 1, \quad j = 1, \ldots, n, \tag{3}$$

$$\sum_{k=1}^{K} y_{ik} \leqslant 1, \quad i = 1, \ldots, m, \tag{4}$$

$$x_{ij}, y_{ik} \in \{0, 1\}, \quad \forall i, j, k. \tag{5}$$

Here, (1) gives the total profit of items accepted, and (2) and (3) represent the same conditions as in MKP with respect to each N_k and the set of knapsacks assigned to this subset of items. Constraint (4) means that each knapsack can be assigned to at most one subset.

Such a problem may be encountered by a marine shipping company in drawing up a cargo plan. Here items are to be shipped to respective destinations, and we have m ships for this transportation. Let N_k represent the set of items destined to the kth destination, and c_i is the capacity of ship i. Cargo planning is to allocate ships to destinations, and for each k load the items in N_k to the allocated ships.

* Corresponding author.
E-mail addresses: seiji@nda.ac.jp (S. Kataoka), yamada.144b@gmail.com (T. Yamada).

http://dx.doi.org/10.1016/j.ejor.2014.02.014
0377-2217/

MKAP is $\mathcal{NP}$-hard, since the special case of $K = 1$ is simply an MKP, which is already $\mathcal{NP}$-hard. For recent works on MKP, readers are referred to Chekuri and Khanna (2006), Dawande, Kalagnanam, Keskinocak, Ravi, and Salman (2000), and Lalami, Elkihel, Baz, and Boyer (2012). Since MKAP described as above is a linear 0–1 programming problem, small instance of this problem may be solved using free or commercial MIP (mixed integer program) solvers such as Gurobi (2012). However, as we shall see later, solvers can solve only small instances within a reasonable CPU time.

Instead of solving MKAP exactly, we present an approach to solve larger instances approximately, but very quickly. More specifically, we first apply the Lagrangian relaxation to (2), and obtain an upper bound quickly. Here, we show that only one multiplier suffices to eliminate these mK inequalities, and the obtained upper bound is shown to be identical to the upper bound derived by the continuous (LP) relaxation of MKAP. In addition, we present an efficient way to solve this LP problem by decomposing it into K independent continuous knapsack problems.

We exploit the result of this computation to derive a heuristic solution, which gives a lower bound to MKAP. Through numerical experiments on a series of randomly generated instances, we evaluate the quality (CPU time and relative errors) of the obtained solutions.

2. Upper bound

To discuss upper bounds, without much loss of generality, we assume the following.

- A_1: Problem data p_j, w_j $(j = 1, 2, \ldots, n)$, and c_i $(i = 1, 2, \ldots, m)$ are all positive integers.
- A_2: Within each subset, items are arranged in non-increasing order of profit per weight, i.e., for all $k = 1, \ldots, K$ the following is satisfied:

$$\forall j, \quad j' \in N_k, \quad j < j' \Rightarrow p_j/w_j \geqslant p_{j'}/w_{j'}.$$

2.1. Lagrangian relaxation

With non-negative multipliers λ_{ik} associated with (2), the Lagrangian relaxation (Fisher, 1981) of MKAP is as follows.

LMKAP($\boldsymbol{\lambda}$):

$$\text{maximize} \quad L(\boldsymbol{\lambda}, \boldsymbol{x}, \boldsymbol{y}) := \sum_i \sum_k \sum_{j \in N_k} (p_j - \lambda_{ik} w_j) x_{ij} + \sum_i \sum_k c_i \lambda_{ik} y_{ik},$$

$$\text{subject to} \quad (3)\text{–}(5).$$

With $\boldsymbol{\lambda} \geqslant \mathbf{0}$ fixed, this problem is easily solved, and the optimal objective value is

$$\bar{z}(\boldsymbol{\lambda}) = \sum_k \sum_{j \in N_k} \max_i \{(p_j - \lambda_{ik} w_j)^+\} + \sum_i \max_k \{\lambda_{ik}\} c_i, \tag{6}$$

where $(\cdot)^+ := \max\{\cdot, 0\}$. Then, $\bar{z}(\boldsymbol{\lambda})$ is a piecewise linear and convex function of $\boldsymbol{\lambda}$. Moreover, if we consider

Lagrangian DUAL:

$$\text{minimize } \bar{z}(\boldsymbol{\lambda}) \quad \text{subject to } \boldsymbol{\lambda} \geqslant \mathbf{0},$$

we have the following.

Theorem 1. *There exists an optimal solution $\boldsymbol{\lambda}^\dagger = (\lambda^\dagger_{ik})$ to Lagrangian DUAL such that $\lambda^\dagger_{ik}$ is constant over i and k, i.e., $\lambda^\dagger_{ik} \equiv \lambda^\dagger$.*

Proof. Let $\boldsymbol{\lambda} = (\lambda_{ik})$ be a feasible solution to the above problem, and put $k^\dagger(i) := \arg\max_k\{\lambda_{ik}\}$. Then, for all k we have $\lambda_{ik} \leqslant \lambda_{ik^\dagger(i)}$. Since $(p_j - \lambda_{ik} w_j)^+$ is a non-increasing function of λ_{ik}, this is minimized at $\lambda_{ik} = \lambda_{ik^\dagger(i)}$, for all k. Thus, in the Lagrangian dual we can assume that $\lambda_{ik} \equiv \lambda_i$, i.e., constant over k.

Next, let $\lambda^\dagger := \min_i\{\lambda_i\}$. Then, we have $\max_i\{(p_j - \lambda_i w_j)^+\} = (p_j - \lambda^\dagger w_j)^+$, and thus

$$\bar{z}(\boldsymbol{\lambda}) = \sum_k \sum_{j \in N_k} (p_j - \lambda^\dagger w_j)^+ + \sum_i \lambda_i c_i,$$

which is minimized at $\lambda_i \equiv \lambda^\dagger$. □

Remark 1. Due to the fact that the coefficients of (2) are identical for all i, this is obtained as an extension of the known result for MKP (i.e., $K = 1$, Martello & Toth, 1990, pp. 164–165). See also (Yamada & Takeoka, 2009).

From this theorem, to obtain $\lambda^\dagger$ it suffices to minimize the one-dimensional

$$\bar{z}(\lambda) = \sum_k \sum_{j \in N_k} (p_j - \lambda w_j)^+ + \lambda C \tag{7}$$

over $\lambda \geqslant 0$, where C is the total knapsack capacity, i.e.,

$$C := \sum_i c_i. \tag{8}$$

At differentiable $\lambda \geqslant 0$, we have

$$d\bar{z}(\lambda)/d\lambda = C - \sum_k \sum_{j \in N_k(\lambda)} w_j, \tag{9}$$

with $N_k(\lambda) := \{j \in N_k | p_j - \lambda w_j > 0\}$. Thus, $\bar{z}(\lambda)$ is a piecewise-linear, convex function of λ, and the optimal solution $\lambda^\dagger$ to the Lagrangian dual is characterized by

$$\lambda \gtrless \lambda^\dagger \Rightarrow C - \sum_{k=1}^{K} \sum_{j \in N_k(\lambda)} w_j \gtrless 0. \tag{10}$$

Such a $\lambda^\dagger$ can be found by the standard *binary search* method, and we obtain the corresponding *Lagrangian upper bound* $\bar{z}_{\mathcal{L}} := \bar{z}(\lambda^\dagger)$.

2.2. Continuous relaxation

By replacing the 0–1 condition (5) with non-negativity requirements, we obtain the continuous relaxation of MKAP as follows.

CMKAP:

$$\begin{aligned} &\text{maximize} && (1), \\ &\text{subject to} && (2)\text{–}(4) \quad \text{and} \\ &&& x_{ij} \geqslant 0, \quad y_{ik} \geqslant 0, \quad \forall i, j, k. \end{aligned}$$

Let $\bar{z}_{\mathcal{C}}$ be the optimal objective value to this problem. This gives the upper bound by the continuous relaxation of MKAP. Then, the following states the relation between the Lagrangian and continuous relaxations.

Theorem 2. *Upper bounds derived from the Lagrangian and continuous relaxations are identical, i.e., $\bar{z}_{\mathcal{L}} = \bar{z}_{\mathcal{C}}$.*

We note that the coefficient matrix of constraints (3) and (4) in LMKAP($\boldsymbol{\lambda}$) is totally unimodular. Then, this theorem follows immediately from Theorem 10.3 (p. 172) of Wolsey (1998).

2.3. Continuous relaxation: an alternative approach

Instead of applying LP algorithms such as the *simplex method* directly, CMKAP may be solved efficiently as follows. Let $u_k := \sum_{i=1}^m c_i y_{ik}$ and $x_j := \sum_{i=1}^m x_{ij}$. Here, u_k is the knapsack capacity allocated (from the total knapsack capacity C) to subset N_k. Then, adding (2) for $i = 1, \ldots, m$, the problem is decomposed into K independent subproblems for each subset as follows.