



**HAL**  
open science

# An Iterative Algorithm for Solving the Multiple Knapsack Assignment Problem

Dahmani Isma, Meriem Ferroum, Mhand Hifi

► **To cite this version:**

Dahmani Isma, Meriem Ferroum, Mhand Hifi. An Iterative Algorithm for Solving the Multiple Knapsack Assignment Problem. 23ème congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision, INSA Lyon, Feb 2022, Villeurbanne - Lyon, France. hal-03595242

**HAL Id: hal-03595242**

**<https://hal.science/hal-03595242>**

Submitted on 3 Mar 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# An Iterative Algorithm for Solving the Multiple Knapsack Assignment Problem

Isma DAHMANI<sup>1</sup>, Meriem FERROUM<sup>1</sup>, Mhand HIFI<sup>2</sup>

<sup>1</sup> AMCD-RO, Département de RO, USTHB, Algérie  
{dahmani.isma, meriem1.mf}@gmail.com

<sup>2</sup> EPROAD, UPJV, 7 rue du Moulin Neuf, 80000 Amiens, France  
hifi@u-picardie.fr

**Mots-clés :** *Greedy, knapsack, local branching, optimization.*

## 1 Introduction

In this paper, the Multiple Knapsack Assignment Problem ( $P_{MKAP}$ ) is tackled. An instance of  $P_{MKAP}$  is characterized by a set  $S$  of  $n$  items, and a second set  $M$  of  $m$  knapsacks. On the one hand, each item  $j$ ,  $j \in S$ , is characterized with its nonnegative profit  $p : S \rightarrow R^+$  and nonnegative weight  $w : S \rightarrow R^+$ . On the other hand, each knapsack  $i$  is characterized by a nonnegative weight  $c : M \rightarrow R^+$ . However, items are divided into  $r$  mutually disjoint subsets of items  $S_k, k \in K = \{1, \dots, r\}$ , where  $S = \cup_{k=1}^r S_k$ ,  $n_k = |N_k|$  and  $n = \sum_{k=1}^r n_k$ . The goal of the problem is to determine the assignment of all knapsacks to each subset, and to fill knapsacks with items in that subset, so as to maximize the total profit related to the selected items.

Several solution procedures have been designed for approximately solving  $P_{MKAP}$ , like Kataoka and Yamada [3] who designed a quick solution procedure for solving small and medium-sized instances. Lalla-Ruiz et Voβ [4] proposed a biased random key genetic algorithm, where its principle is based upon combining some random strategies for creating diversification of the population. More recently, Martelo and Monaci [5] tackled the problem by using a constructive heuristic, with its enhancement. Their algorithm uses the bounding strategy derived from either Lagrangian relaxation or surrogate relaxation.

## 2 The principle of the iterative algorithm

$P_{MKAP}$  is tackled by using an iterative search, where a descent procedure is combined with an intensification operator coupled with a non-cycling strategy. The following steps are applied :

1. A starting feasible solution is built by using a tailored greedy bin-packing procedure (cf. Martelo and Monaci [5]).
2. For each current solution, several neighboring strategies are applied. As used in Hifi and Michrafy [2], a reactive operator is called for enhancing the quality of the successive solutions, where two complementary strategies are considered :
  - (a) To build a partial solution by randomly dropping some items from the current solution. It means that a subset of variables is temporary assigned.
  - (b) To solve the rest of the problem by applying a repair operator to complete the current partial solution.

In order to enhance the quality of the solutions, two procedures are used :

3. Both truncated 2-opt and 3-opt operators are applied around the current solution for highlighting the quality of the solutions.
4. The local branching (cf. Fischetti and Lodi [1]) is called for highlighting the solution at hand, especially when restricting the search space.

Such a process is iterated till matching the stopping condition. Hence, the final solution achieved by the method is returned as the best solution for  $P_{MKAP}$ .

### 3 Preliminary results

The proposed algorithm was analyzed on some benchmark instances extracted from Kataoka and Yamada [3], where its achieved results are compared to those achieved by the best available methods of the literature (cf. Martelo and Monaci [5]).

#Inst			Results from [5, 3]		This work		
fam_r	m	n	Err%	Err%	Av.	Best	
UNC_2	10	20	7,70	0,10	<b>7446,00</b>	<b>0,00</b>	
		40	3,50	1,03	<b>16270,20</b>	<b>0,00</b>	
		60	0,74	0,01	<b>24688,70</b>	<b>0,00</b>	
	20	20	6,00	0,00	<i>4685,80</i>	0,00	
		40	4,49	1,14	<b>15440,80</b>	<b>0,06</b>	
		60	3,33	0,66	<b>24574,40</b>	<b>0,11</b>	
		Av	4,29	0,49	<b>15517,65</b>	<b>0,04</b>	
	WEA_2	10	20	9,81	0,47	<b>5547,50</b>	<b>0,00</b>
			40	1,59	0,98	<b>12314,70</b>	<b>-0,01</b>
			60	1,61	<b>-0,06</b>	18683,50	-0,02
20		20	12,10	0,00	<i>2957,00</i>	0,00	
		40	5,76	1,40	<b>11628,70</b>	<b>0,00</b>	
		60	6,26	1,72	<b>18593,70</b>	<b>0,00</b>	
Av		6,19	0,59	<b>11620,85</b>	<b>-0,01</b>		
STR_2		10	20	5,45	0,02	<b>7159,60</b>	<b>0,00</b>
	40		2,00	0,08	<b>15375,30</b>	<b>-0,01</b>	
	60		1,39	<b>-0,07</b>	23145,80	-0,01	
	20	20	7,98	0,00	<i>4150,30</i>	0,00	
		40	5,03	1,16	<b>14989,10</b>	<b>0,00</b>	
		60	4,18	0,46	<b>23107,30</b>	<b>0,00</b>	
	Av	4,34	0,27	<b>14654,57</b>	<b>0,00</b>		

TAB. 1 – Behavior of the iterative method vs two methods of the literature.

Table 1 reports some preliminary results provided by the proposed method (under "This work") and those achieved by two available methods of the literature. From this table, we can observe that the proposed method remains very competitive since, on several occasions, it is able to succeed (values in bold-space) to the results reached by two other compared methods.

### 4 Conclusion

In this paper, an iterative algorithm was proposed for approximately solving a variant of the knapsack problem : the multiple knapsack assignment problem. From a current solution, the method combines several strategies for enhancing the quality of the solutions. Finally, the preliminary experimentation showed that the proposed approach remains competitive, where new bounds have been reached.

### Références

- [1] Fischetti, M., and Lodi, A. (2003). *Local branching*. Mathematical Programming, 98(1-3), 23–47. <https://doi.org/10.1007/s10107-003-0395-5>
- [2] Hifi, M., Michrafy, M. (2006). *A reactive local search-based algorithm for the disjunctively constrained knapsack problem*. Journal of the Operational Research Society, 57, 718-726.
- [3] Kataoka S , Yamada T. (2014). *Upper and lower bounding procedures for the multiple knapsack assignment problem*. European Journal of Operational Research, 237, 440-7.
- [4] Lalla-Ruiz E , Voβ S. (2015). *A biased random-key genetic algorithm for the multiple knapsack assignment problem*. In : Dhaenens C, Jourdan L, Marmion M, editors. Learning and intelligent optimization : 9th international conference, LION 9, Lille, France, January 12-15, 2015. Revised selected papers. Cham : Springer International Publishing, p. 218-222
- [5] Martelo S., Monaci M. (2020). *Algorithmic approaches to the multiple knapsack assignment problem*. Omega, 90, 1-11, 102004.