Contribution by a hybrid algorithm to solve the multi-dimensional multiple-choice knapsack problem MMKP

FAWZIA HAMDAOUI
Computer Sciences Department University of sciences & Technology of Oran MOHAMED BOUDIAF
Oran, Algeria
fawzia.hamdaoui@univ-usto.dz
MAHMOUD ZENNAKI
Computer Sciences Department University of sciences & Technology of Oran MOHAMED BOUDIAF
Oran, Algeria
Dr . KADDOUR SADOUNI
Computer Sciences Department University of sciences & Technology of Oran MOHAMED BOUDIAF
Oran, Algeria
kaddour.sadouni@univ-usto.dz
mzennaki@univ-usto.dz

Abstract— In this paper, we approximately solve the multiple-choice multi-dimensional knapsack problem. We propose a mixed algorithm based on branch and bound method and Pareto-algebraic operations. The algorithm starts by an initial solution and then combines one-by-one groups of the problem instance to generate partial solutions in each iteration. Most of these partial solutions are discarded by Pareto dominance and bounding process leading at the end to optimality or near optimality in the case when only a subset of partial solutions is maintained at each step. Furthermore, a rounding procedure is introduced to improve the bounding process by generating high quality feasible solutions during algorithm execution. The performance of the proposed heuristic has been evaluated on several problem instances. Encouraging results have been obtained.
Keywords— combinatorial optimization, heuristics, knapsacks, branch and bound.

I. Introduction

In this article, we try to propose a resolution to approximate the problem of multidimensional knapsack multiple choice (MMKP).

The problem of multi bag back choice multidimensional MMKP is a special case of the general problem of the backpack, considered one of the combinatorial optimization problems most studied in recent years, because this problem has many practical applications [1].

The fundamental problem of the bag back to binary 0-1 considers element n, where each element has a value of v profit and cost of the proposed weight w resource.

The goal is to put the items in a backpack so that the capacity of the backpack resources is not

exceeded and the value added benefit of packaged items is maximized [2].

The MMKP is a variant of the complex problem of binary bag back 0-1, it is classified as a combinatorial optimization problem NP-hard [1]; where the items are listed in the classroom, the selected item will eliminate the choices of other objects belonging to the same class.

It is applied in many forms; in industrial or economic real world applications [1], such as space management or cutting.

It is a sub-problem to solve a more general problem. So it's resolution contributes to solving it.

Formally MMKP is to maximize the objective function (gain) at a number of capacity constraints and choice constraints[3].

The idea of the MMKP is to choose exactly one item from each class to maximize the value of total profit for this choice subject to resource constraints. Considering the decision variables x_{ij}

when j element of the i class is taken then it is equal to 1 or 0, the MMKP can be formulated in a linear program [4].

2 II. DEFINITION OF MMKP PROBLEM

The MMKP problem is characterized by: - A vector of size m said capacity or resources $R = (R_1, R_2, .., R_m)$
- A set $S = (S_1, ..., S_i, ..., S_n)$ to be divided into n disjoint classes such that for every pair (p, q) objects; such that: $p \neq q$; $p \leq n$ and $q \leq n$, we have $S_p \cap S_q = \emptyset$; and $S_1 U ... U S_{n-1} = S$.
Each class i; i = 1, .., n is number of objects of class i. we must seek to maximize an objective function that is a profit where ach object j of class i associated vij a positive profit and a weight vector $W_{ij} = (w_{ij}^1, w_{ij}^2 .., w_{ij}^m)$.

The goal is to assign the knapsack, exactly one and only one object per class with a maximum benefit without violating the capacity constraints [2].

The MMKP can be formulated in an Integer Linear Program (ILP) as follows: [7]

$$(MMKP)\begin{cases} Z(x) = \max \sum_{i=1}^{n}\sum_{j=1}^{r_i} v_{ij}x_{ij} & \\ \text{s.c.} \sum_{i=1}^{n}\sum_{j=1}^{r_i} w_{ij}^k x_{ij} \leq R^k, & k \in \{1,\ldots,m\} \\ \sum_{j=1}^{r_i} x_{ij} = 1, & i \in \{1,\ldots,n\} \\ x_{ij} \in \{0,1\}, & i \in \{1,\ldots,n\},\ j \in \{1,.. \end{cases}$$

Note that the variable x_{ij} is 1 if the object j of the class i is taken from the bag and is 0 otherwise. The constraints of type (1) are the capacity constraints. The constraints of type (2), called selection constraints, assure that each class of a single object must be selected. Authors are considered a variant that generalizes two other problems generalizing also the problem of bag-to-back: the problem of MDKP and the problem of MCKP, and recently MMMKP. The problem MMKP becomes a problem when there MCKP one capacity constraint, whereas if there is only one class of choice and constraints will no longer have reason to be then it becomes a MDKP problem [7].

Solving methods of MMKP differ from that of other variants of KP problem because the wording is different, and the process of resolution can be derived from the methods used for the other variants. [8]

This paper presents BPH, for Branch and bound Pareto-algebraic Heuristic. BPH is a heuristic based on Branch and Bound (B&B) and uses the principle of Pareto algebra.

III A branch and bound based heuristic

Discuss in this section, largely specific to the resolution of the problem of MMKP existing methods, they represent two broad approaches to resolution: heuristics or accurate, their derivatives are based on the characteristics of each that prove to be complementary. [9]

An exact method is characterized by the near certainty of achieving the optimal solution is theoretical but given the time of exponential calculation (2k) such that k is the number of objects) in the space of solutions of the problem. An approximate method known as heuristic consists in solving an optimization problem to reduce the search space resulting in reduced time to implementation; while not guaranteeing the optimality of the solution found [8] [13] [14].

The exact methods tell complete because it lists all the solutions; and approximate methods are called incomplete because it explores a subset of solutions. In the literature, there is very little accurate treating MMKP algorithms. These algorithms are based on branch and bound methods and differ in the valuation method used and the method of separation function. The first such algorithm was proposed by Khan et al. [10] It is based on the upper bound produced by the simplex method and then uses the method of Branch & Bound exploring the search tree by selecting the Best first (best first). The algorithm produces an optimal solution for instances of small and medium size [11].

As for the heuristic approaches, one of the first proposed heuristic is the one set by Moser [5] based on Lagrangian relaxation; Then comes the heuristic-based greedy algorithm proposed by Dantzig [8]; the generation of the column described in [1], it was enhanced by the concept of hybridizing with the connection to improve the quality of solutions Their concept converges to the same idea: first find a feasible solution for MMKP instance and iterate this calculation method by removing elements to improve the final solution[1].

Motivated by the success of branch and bound algorithm in exact methods and Pareto-algebra in approximate methods, we propose a heuristic based on a combination of the two aforementioned approaches enhanced by a rounding procedure which can generate high quality feasible solutions during the search process.

IV HYBRID ALGORITHM

We propose a procedure based on the branch-and-bound (B&B) incorporating a modified method of B&B combined operations Pareto-algebraic version of hybrid algorithm; but does not guarantee the optimality of the solutions obtained. So the proposed approach we combine classical exact branch of B&B with the heuristic of Pareto algebra. The Branch and Bound use the separation algorithm and evaluation (Branch and Bound), so it guarantees an intelligent exploration of the field of solutions [2].

However the efficiency depends on how to choose to carry out the separation and evaluation. The principle of separation: The separation principle is to divide the problem into a number of sub-problems each with its set of feasible solutions. Resolving all sub-problems and taking the best solution found, it is guaranteed to have solved the original problem.
The separation principle is applied recursively to each of subsets as it contains several solutions.

Note: The process of separating a set stops when the following conditions are satisfied:

-knows the best solution of all;
-knows better than any of the solution set;
-knows all there are no feasible solution.
The Strategy applied:
The strategy is the rule for choosing the next summit to be separated from the set of vertices of the tree. Among the best known strategies course include:

[3] The depth-first: The exploration focuses on sub-problems obtained by the largest number of separations applied to the initial problem, that is to say the most distant peaks of the root (the highest depth). Rapidly obtaining a feasible solution (for problems where it is difficult to get a good heuristic) and the little space required memory are the benefits. The downside is the exploration of subsets which may be inauspicious to obtain an optimal solution.

[4] The breadth-first: This strategy facilitates the sub-problems obtained by the least separation problem of starting, that is to say the closest to the root apexes (depth the lowest).

[5] The best first: This strategy encourages the exploration of sub-problems with the smaller lower bounds.

The strategy directs research where the probability of finding a better solution is the largest. We use the following strategies: DFS, BFS and The Best first, the strategy will determine the next steps in terms of quality and optimal execution time. [2] In our algorithm, we choose the method of Best First reduce for the execution time.

The higher maximum Zsup is initialized, for each iteration of a branch of the tree, resulting Zbest maximum is compared to Zsup if above the Zsup retrieves the value of Zbest. [8]

The algebra of Pareto Using algebra Pareto in [11] was combined with another heuristic; but it is based on the basic concept of algebraic operations Pareto namely configurations dominated and infeasible are removed from the search space of the solution remains the only dominant. It overcomes the explosion of the space of possible combinations of the search tree of B & B [12].

The concept of the hybrid algorithm The algorithm begins by generating an initial solution and then combines the groups one by one instance of the problem to generate partial solutions to each iteration.
Most of these solutions are eliminated by the Pareto dominance; but also by the process of evaluation at the end leading to the optimal solution or a solution close to the optimum in the case where only part of the partial solutions is maintained at each step. In addition, a rounding procedure is introduced to improve the evaluation process by generating feasible solutions of high quality while running the algorithm. The performance of the heuristics should be evaluated on two types of bodies; namely regular and non-regular instances (Table 1 & 2).

Regular Instances	Number of CLASS	Number of CONTRAINTES	Number of objects in class	Total objects
I01	5	5	5	25
I02	10	5	5	50
I03	15	10	10	150
I04	20	10	10	200
I05	25	10	10	250
I06	30	10	10	300
I07	100	10	10	1000
I08	150	10	10	
I09	200	10	10	
I10	250	10	10	
I11	300	10	10	
I12	350	10	10	
I13	400	10	10	

Tab1 A regular Instances used

Irregular Instances	Number of CLASS	Number of CONTRAINTES	Total obje
RTI07	10	5	23
RTI08	20	10	109
RTI09	30	10	158
RTI10	30	10	235
RTI11	30	20	208
RTI12	40	10	241
RTI13	50	10	295
INST21	100	10	565
INST22	100	20	538
INST23	100	30	541
INST24	100	40	584
INST25	100	10	871
INST26	100	20	842
INST27	200	10	1076
INST28	300	10	1643
INST29	400	10	2223
INST30	500	10	2704

Tab2 Irregular Instances used

For regular instances, they are among 13 instances. [10] The first six bodies are small and medium in which the optimal solutions are known their size. The remaining seven bodies are characterized by their large size, the number of class is of the order of 100 to 400 with the same number of constraints 10, note that their optimal solutions have not been proven. [1] As for non-regular instances, the number of classes is in the range 10 {discrete; 20, 30, 40, 50, 100; 200; 300, 400; 500}; the number of objects varies between 23 and more than 2,500 objects.

V THE PSEUDO-ALGORITHM OF BRANCH & BOUND – PARETO – HEURISTIC (BPH)

The algorithm is based on combining one-by-one groups of the MMKP instance using Pareto-algebra product operation as explained in figure 1. It is obvious that exact solution based on Pareto-algebra product cannot be considered for large instances.

As a first step, the algorithm tries to find Pareto points in each configuration set. Dominated configurations cannot contribute to an optimal solution of the MMKP instance.

Input : MMKP instance with a vector of configurtions S and a vector of capacity F
C_i, C_{i+1} configurations sets
Output : Z_{best} solution

- for all A_i Є S do min(A_i)
- Z_{debut} = initial sol() // -∞
- Z_{best} = Z_{debut}
- Sort vector S in order to put groups with first items
- Initialize a set of partial solutions $A_{partial}$=S(1) and S= S – S(1)

6. for all A_i Є S do
 combine $A_{partial}$ with configurations A_i
 eliminate discard any infeasible
 or dominated configuration from $A_{partial}$
 $A_{partial}$ = product sum min(C_i , C_{i+1} , F)
 for all configuration $\bar{a}_j$ Є $A_{partial}$ do
 calculate bound value Z_{sup}
 if Z_{sup} + profit($\bar{a}_j$) ≤ Z_{best} then
 discard partial solution
 else
 if Z_{sup} + profit($\bar{a}_j$) >Z_{best} then
 Z_{best} = Z_{sup} + profit($\bar{a}_j$)
 apply best first search strategy to select
 A_{best} =best ($A_{partial}$)
7. Z= best configuration ($A_{partial}$)
 if Z > Z_{best} then Z_{best} =Z
8. return Z_{best}

Fig1 The Pseudo-algorithm BPH proposed

VI Computational results

The purpose of this section is to experimentally investigate the various aspects of BPH on standard benchmarks. We evaluate the performance of BPH compared to the state-of-the-art best results. The obtained results are also compared to those obtained when running one hour Cplex Solver v12.2 on the same set of instances. Our algorithms were coded in C++ and all experiments were done on a PC with a 2.13GHz Intel Pentium Dual Core CPU and 2GB of memory.

Regular instances					
#Inst	n	r_i	m	$\sum_{i=1}^{n} r_i$	Opt
I01	5	5	5	25	173
I02	10	5	5	50	364
I03	15	10	10	150	1602
I04	20	10	10	200	3597
I05	25	10	10	250	3905,7
I06	30	10	10	300	4799.3
Irregular instances					
#Inst	n	r_{max}	m	$\sum_{i=1}^{n} r_i$	Opt
RTI07	10	5	5	23	564
RTI08	20	10	10	109	6576
RTI09	30	10	10	158	7806.2
RTI10	30	20	10	235	7032
RTI11	30	20	20	208	6880
RTI12	40	10	10	241	11564
RTI13	50	10	10	295	10561

Table 1. Small to medium size test problem details

Regular instances					
#Inst	n	r_i	m	$\sum_{i=1}^{n} r_i$	Upper b.
I07	100	10	10	1000	24607.95
I08	150	10	10	1500	36904.41
I09	200	10	10	2000	49193.87
I10	250	10	10	2500	61486.30
I11	300	10	10	3000	73797.74
I12	350	10	10	3500	86100.45
I13	400	10	10	4000	98448.64
Irregular instances					
		r_{max}			
INST21	100	10	10	565	44315
INST22	100	10	20	538	42076
INST23	100	10	30	541	42763
INST24	**100**	**10**	**40**	**584**	**42252**
INST25	**100**	**20**	**10**	**871**	**44201**
INST26	**100**	**20**	**20**	**842**	**45011**
INST27	**200**	**10**	**10**	**1076**	**87650**
INST28	**300**	**10**	**10**	**1643**	**134672**
INST29	**400**	**10**	**10**	**2223**	**179245**
INST30	**500**	**10**	**10**	**2704**	**214257**

Table 2. Large size test problem details

The problems we considered are summarized in Tables 1 and 2. We tested a total of 30 instances corresponding to two groups: (i) regular instances with groups containing the same number of items.

We can draw several conclusions from these results.

First, the BPH results are competitive in terms of quality and running time especially those given in bold.

Second, in columns reporting the pure Cplex results with a time budget of one hour, we may conclude that hybrid heuristics outperform pure Cplex when given equal time budgets.

Third, the results we obtained with genetic algorithms are disappointing despite the use of several repair operators to deal with infeasible solutions generated by crossover operator.

In fact, we are persuaded that in the case of MMKP, any purely heuristic approach is doomed to fail. Hybridization with exact methods is better suited.

Finally, note that with BPH heuristic, we have only one parameter to adjust; the parameter L representing the number of partial solutions maintained at each iteration, while in all the state-of-the-art heuristics, there are several parameters to consider, and it is well known that when using approximate algorithms to solve optimization problems, different parameter settings lead to results of variable quality and the configuration of these parameters is a difficult task.

VII CONCLUSION

We started to implement the proposed algorithm to solve the problem of bodies own multidimensional multiple choice knapsack using a hybrid algorithm.

The algorithm is based primarily on the use of Pareto - a product that combines an all sectors of the MMKP instance at hand. Second, much of the generated partial solutions is rejected either by Pareto dominance or better by the exact method of Branch & Bound. A rounding procedure is used to generate realistic high quality solutions in the execution of the algorithm, and the improvement of the selection process.

Encouraging results are possible because we think it will provide high-quality solutions in a reasonable computation time, and can generate good solutions in a reduce due to rounding performed at each iteration of the algorithm execution time .

References

[1] N. Cherfi, M. Hifi, ‘A column generation method for the multiple-choice multi-dimensional knapsack problem’, Comput. Optim. App. online first, Springer Netherlands, 2008.

[2] M. SAKAVOVITCH, “Optimisation combinatoire”, HERMAN. (1984), pp. 1-4 & 19-136

[3] Daniel C. Vanderster, Nikitas J. Dimopoulos, Randall J. Sobie “Metascheduling Multiple Resource Types using the MMKP” Grid Computing Conference , 2006, pp. 231-237.

[4] H.Shojaei, A.Ghamarian, T.Basten, M.Geilen, S.Stuijk, R.Hoes “A Parameterized Compositional Multi-dimensional Multiple-choice Knapsack Heuristic for CMP Run-time

Management", DAC'09, July 26-31, 2009, San Francisco, California, USA 917-922

[5] M. Moser, "An algorithm for the multidimensional multiplechoice knapsack problem," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. 80, no. 3, pp. 582–589, 1997.

[6] Bing Han, Jimmy Leblet, Gwendal Simon "Hard Multidimensional Multiple Choice Knapsack Problems, an Empirical Study" Computers & Operations ResearchVolume 37, Issue 1, January 2010, Pages 172–181

[7] A. Sbihi, "A best first search exact algorithm for the multiple-choice multidimensional knapsack problem", Journal of Combinatorial Optimization 13 (4) (2007) 337-351

[8] A. Sbihi, "Les methodes hybrides en optimisation combinatoire: algorithmes exacts et heuristiques", Thèse de doctorat, CERMSEM - LaRIA ; 2006

[9] M.R. Razzazi, T.Ghasemi "An Exact Algorithm for the Multiple choice Multidimensional Knapsack Based on the core", CSICC 2008, CCIS 6 pp275-282, Springer.

[10] S. Khan, K. F. Li, E. G. Manning and MD. M. Akbar. "Solving the knapsack problem for adaptive multimedia systems". Studia Informatica, an International Journal, Special Issue on Cutting, Packing and Knapsacking problems, 2/1 :154_174_589, 2006.

[11] H.Shojaei, T.Basten, M.Geilen, A.Davoodi "A Fast and Scalable Multi-dimensional Multiple-choice Knapsack Heuristic"; ACM Transactions on Design Automation of Electronic Systems, **2013**

[12] H.Shojaei, T.Basten, M.Geilen, P.Stanley-Marbell, "SPaC: A Symbolic Pareto Calculator" , 2008

[13] R.Picot-Clemente, F.Mendes, C.Cruz, C.Nicolle, 'TOURISM-KM A variant of MMKP applied to the tourism domain', Author manuscript, published in "ICORES 2012, Portugal (**2012**)"

[14] W. Wang, W.Lu, L.Wang, W.Xing, Z.Li 'A ranking-based Approach for service composition with multiple QoS Constraints', Proceedings of the 2012 International Conference on Information Technology and Software Engineering; pp185-196; 2012