Smart agents for the Multidimensional Multi-choice Knapsack Problem

Skander Htiouech
University of Jeddah, Kingdom of Saudi Arabia (KSA)

Ameur Alzaidi
University of Jeddah, Kingdom of Saudi Arabia (KSA)

ABSTRACT

In this paper, we propose a multi-agent approach for solving the multidimensional multi-choice knapsack problem (called MMKP). The MMKP is an NP-Hard optimization problem in strong sense. It is considered as a combination of two other variants such as: the multi-choice knapsack problem (MCKP) and the multidimensional knapsack problem (MDKP). The MMKP can be applied in many problems in real world. It can model many industrial situations, such as capital budgeting, model of allocation resources and finance. The particular properties of the MMKP favor its decomposition into many MMKP sub-problems with small sizes. The assignment of sub-problems and the sharing of available resources are allocated to a first agent. Each subproblem is then solved by an agent. To work collaboratively, a strategic negotiation between agents has been defined. A coordinator agent (CA) will evaluate and merge the generated solutions to build a feasible solution to the initial problem. The choice rules of the CA is modeled as a multidimensional knapsack problem (MKP). The proposed method is able to solve several instances of literature effectively, in particular for large size instances.

General Terms

Metaheuristic, knapsack problem

Keywords

Combinatorial optimization, Agents, multiple choice, knapsack problem

1. INTRODUCTION

The multidimensional multi-choice knapsack problem (MMKP) is an extended variant of the classical knapsack problem (KP) [6] that is classified into the most complex combinatorial optimization problem. In MMKP, we are given a set of item groups. Each item is characterized by a profit and requires certain resources represented by a weight vector. To solve the MMKP, we must to pick only one item from each group such that all the resource constraints are satisfied and to find the subset of items whose the sum of profits values are maximized. Formally, the MMKP can be written as follows :

$$\textit{Maximize} \quad \sum_{i=1}^{n}\sum_{j=1}^{n_i} c_{ij}x_{ij} \tag{1}$$

$$\textit{Subject to} \quad \sum_{i=1}^{n}\sum_{j=1}^{n_i} a_{ij}^{k}x_{ij} \leq b^{k} \ , k=1,\ldots,m \tag{2}$$

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

$$x_{ij} \in \{0,1\}, i=1,\ldots,n, j=1,\ldots,n_i \tag{4}$$

the vector b=$(b^1,b^2,\ldots,b^m)$ represent the capacity of the all knapsack resources, and a set of n item groups N=$\{N_1,\ldots,N_i,\ldots,N_n\}$ where every group i, i=$1,\ldots,n$ has n_i items. Every item j, j=$1,\ldots n_i$, of the ith group has a non-negative profit value c_{ij}, and requires an amount of resources a_{ij}=$(a_{ij}^1,a_{ij}^2,\ldots,a_{ij}^k)$. The weight terms a_{ij}^k (with $1 \leq k \leq m, 1 \leq i \leq n, 1 \leq j \leq n_i$) must be positive. It is worthy to note that x_{ij} takes either 1 or 0, which means that item j of the i^{th} group is picked or not, respectively. To eliminate trivial solutions, we assume that for all $1 \leq j \leq n_i$ we have :

$$\sum_{i=1}^{n} min\{a_{ij}^k\} \leq b^k \leq \sum_{i=1}^{n} max\{a_{ij}^k\} \quad k=1,\ldots,m \tag{5}$$

Constraints (2) represent the capacities knapsacks constraints. Constraints (3) ensure to choose exactly one item of each group, and constraints (4) for the integrality $(0or1)$.The maximization of the total profit is modeled by the linear objective function (1).

The remainder of the paper is organized as follows. Section 2 presents a brief reference of some relevant related work for the MMKP. We describe our contribution in section 3. We explain how an MMKP can be divided into many subproblems an how can be solved using agents. We report the computational results in Section 4. We discuss the essential and the futures works of our contribution in Section 5.

2. RELATED WORKS

The MMKP is the result of merging the Multi-choice knapsack problem (MCKP) [6][7] and the Multidimensional knapsack problem (MMDKP) [2][3][15][16]. It is obviously NP-hard[8][9]. The MCKP is a special case of the MMKP if $(m = 1)$ [13] [27]. The MMKP becomes an MDKP by removing choice constraints (3) and considering only one group of items ($n = 1$) (all objects are in the same group) [26].

The first algorithms for the MMKP are due to Moser *et al.* [4] in 1996. The authors have proposed heuristics using Lagrangian relaxation that starts by finding a first feasible solution, and switches repeatedly objects to improve its quality. In 1998, Khan *et al.*[1] have designed a method using the aggregate resources introduced by Toyoda [17] for the MDKP. More detailed information for the MDKP can be found in [12][15][16] and [22]. In [18], Parra-Hernandez and Dimopoulos transform an algorithm due to Pirkul [29] for MDKP to solve the MMKP. First, they relax the choice constraint (3) and transform it into $\sum_{j=1}^{n_i} x_{ij} \leq 1$. This means that at most one object can be selected in each group. They improve the solutions obtained by Akbar *et al.* [30] but penalized with a significant increase in computation time. Hifi *et al.* [17] propose several approaches for the MMKP. They investigate two algorithms that are later included in a main algorithm. The first is a constructive procedure to product an initial feasible solution. The second aimes to improve the quality of the solution found. The main algorithm is a guided local search that uses a penalization strategy. In Mansi *et al.* [31] and Shojaei *et al.* [32], the authors summarize the state of the art of the best approaches proposed for the MMKP. Mansi *et al.* [31] investigate an iterative algorithm to find an upper and lower bounds for the optimal solution of the MMKP using different family of cuts. Mansi *et al.* [25] describe a new linear programming relaxations based on solution which reduces the problem by fixing some particular variables. These solutions are used to update the global lower and upper bounds. Akbar *et al.* [30] obtained good results especially for the uncorrelated instances when compared to the algorithms of Moser *et al.* [4]. In doing so, they use a generation of convex hulls and the aggregation of MDKP constraints into a single one using a penalty vector.

In [20], Chaitr *et al.* provide an empirical study on particular test instances to show the performance of their greedy heuristics for the MMKP.

In [24], Cherfi and Hifi describe an exact approach for the MMKP that uses a variant of the column generation algorithm and a rounding heuristic to solve some of the problems at the branching nodes. The authors compare their approach with the heuristic described in [18] using a set of benchmark instances. In 13 of 30 instances, the value of the best known lower bound was improved.

Cherfi *et al.* [24] describes three new approaches for the MMKP. The patterns are, respectively, based on a local branching algorithm, on a hybrid algorithm combining local branching with column generation, and on a truncated branch-and-bound algorithm that embeds the previous hybrid method.

Htiouech *et al.* [13] describe an oscillation heuristic which explores both sides of the feasibility border; it uses surrogate constraint information for building the choice rules. In order to strengthen their constraint information, the authors enhance the method with constraints normalization. In [27], Htiouech *et al.* propose a tabu search algorithm using a Lagrangian relaxation informations to generate others choice rules. Numerical results show that the performance of his approach is better than previously published results.

In [36], Chen and Hao propose a 'reduce and solve' heuristic that combines problem reduction techniques with integer linear programming. Their method recognizes variables which are highly likely to be part of the optimal solution and fixes them to one (group fixing).

More recently, Stefan Vo and Eduardo Lalla-Ruiz [33] gives a reformulation of the MMKP. A set of partitioning problems are proposed to allow for new insights into modeling the MMKP. The computational experimentation provides new insights into the problem itself and shows that the new model is able to improve the best of the known results for some of the most common benchmark instances.

Multi-agent systems can be defined as computational systems in which several agents interact or work together in order to achieve goals. In recent years, agent-based computation methods have gained great importance from the artificial intelligence research community [37] [38]. To our knowledge, very few papers addressing the solving of knapsacks problem family by multi-agents systems are available. In [34], a distributed algorithm is proposed for the $0-1$ knapsack problem based on the mobile agent, and it is feasible and effective in theoretical analysis. The authors combine the mobile agent technology with a traditional parallel algorithm of the knapsack problems. The theoretical analysis demonstrates that the algorithm has smaller communication spending and good load balance for certain nodes. In [35], Brent A. Smolinski introduces a market model based on agent decomposition and market auctions for approximating the 0-1 multiple knapsack problem, and an algorithm that implements the proposed model.

More recently, Ben rejeb, Htiouech and Bouamama [28] propose a new approach to solve the MMKP using CPU-GPU architecture using CUDA. To do this, the authors apply an efficient parallel implementation of the branch and bound algorithm developed by Hifi and Sbihi *et al.* [17] on a CPU-GPU.

In this paper, a multi-agent system is proposed for the MMKP. We describe how agents should cooperate and negotiate extra resources to solve their subproblems and how to assign sub-problems efficiently to each agent. We explain how to define each sub-problem by a set of groups starting from the initial problem, redistribute the resources to the sub-problems an solve them using autonomous agents.

3. METHOD IN DETAILS

Different variants of knapsack problems are found in the literature [19][9], including, the multidimensional (MDKP)[2][3], multiple-choice (MCKP) knapsack problems[10][11], and bounded problems [12], and others. The MMKP is a variant of the MDKP where items are divided into groups, and exactly one item per group must be selected. In this section, we use a decomposition of the initial problem into a set of sub-problems which will be executed independently and concurrently. This decomposition is possible thanks to the properties of the MMKP problem where all groups of objects are disjoint. If we assign to each sub-problem a part of the available resources, it becomes a new MMKP problem.

Finding a feasible solution to each sub-problem will necessarily lead to a feasible solution of the initial MMKP. The merging of all feasible solutions of sub-problems provides feasible solution of the initial MMKP with a profit equal to the sum of all profits of sub-problems (Fig. 1). The division phase of the initial MMKP into several smaller MMKP amounts to choosing the right parameters in order to distribute the available resources proportionally to the resource vectors of each object. The quality of our multi agent system should specify :

(1) How the assigning agent (AA) should assign the part of resources for each agent from the initial resource vector b.

(2) The performance of the used algorithm by agents to solve its problems and the result of the negotiations between CA and executive Agents for extra resources.

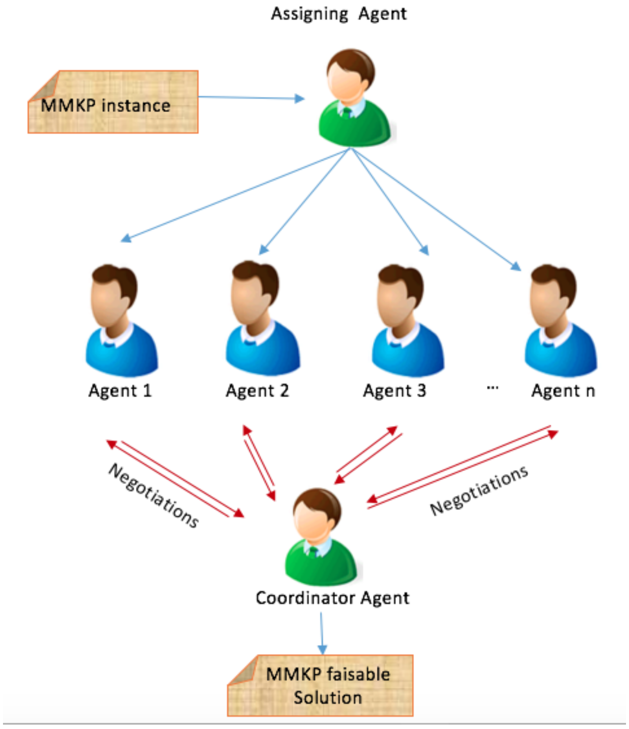

Fig. 1. The principle of AMMKP

3.1 Resources assigning

The task of the assignment officer (AA) is the most delicate. Indeed, AA must take into account several parameters, such as the profits of the objects, the resource vectors of each group of objects. AA will classify groups of objects with the closest characteristics, and choose them for assigning to the same sub-problem. This strategy allows us to minimize the resolution efforts to the agents, since the sub-problems to be solved will have very close correlations.

3.2 Behavior of agents

In this work, each agent is responsible for solving a part of the initial problem. First, a Coordinator Agent (called CA) divides the initial problem into several sub-problems using rules defined in subsection 3.1. Then, each sub-problem will be assigned to an agent with its part of resources. Each agent is responsible of the resolution of exactly one sub-problem. During the resolution, agents could negotiate CA for additional resources to improve the quality of solutions. This negotiation is done by sending a message to CA containing the missing resources vector and the estimated improvement value.

The negotiation and decision to distribute extra resources are made through CA, which decide to respond positively to an agent's request or not.

Probably, CA could receive many agents messages for additional resources.

The CA tries to satisfy agents which will offer the best (in terms of objective function value) and the most economical (in terms of resource consumption) improvements. This situation can be modeled formally as a multidimensional knapsack problem as follows :

$$\textit{Maximize} \quad \sum_{i=1}^{na} A_i x_i \tag{6}$$

$$\textit{Subject to} \quad \sum_{i=1}^{n} \delta_i^j x_i \leq B^j \ , j = 1, \ldots, m \tag{7}$$

$$x_i \in \{0, 1\}, i = 1, \ldots, na \tag{8}$$

Where A_i represents the improvement value proposed by the agent i, and δ_i^j is the extra resource required by the agent i in the dimension j. The solution of this problem represents the list of agents that CA can satisfy. The agent having completed its job returns a list of best solutions found, and must imperatively return the remaining resources to the CA to satisfy more requests for other agents.

A collector agent (Called COA) is responsible for gathering and merging solutions to build the final solution of the initial MMKP problem. It can decide in some cases to refuse the quality of solutions provided by some agents and require to improve it by providing more resources.

Each agent has a list of algorithms. The choice of the algorithm to use is primordial. Each agent uses rules to choose the appropriate algorithm for the resolution of its sub-problem. We propose, in our case, an exact and an approximate approach from the literature. The exact algorithm used is proposed by Sbihi in [23]. The author uses a branch and bound algorithm with the best-first search strategy.The algorithm becomes very greedy when the number of groups is greater than 10. When the sub-problem is still relatively large (more than 10 groups) the agent could use the heuristic due to Htiouech *et al.* [27] which explore both sides of the feasibility border to find a hight quality solution in a reduced time. This heuristic involves tabu search techniques and starts by either feasible or not feasible solution.

To improve the quality of the solution S by CA. The procedure proposed by Htiouech *et al.* [13] use surrogate constraint information as choice rules to intensify the research. The authors use a normalization phase to strengthen the surrogate constraint information in order to improve the computational results.

4. EXPERIMENTATIONS AND TESTS

4.1 Experimental design

In this section, we detail the computational results provided by our approach (called AMMKP). We compare our heuristic to other methods from the literature.

The set of 13 regular instances (named $I01 - I13$) due to Khan *et al.* [6] are correlated instances and frequently used to test recent MMKP algorithms [40] [13][41]. For each instance, we present in Table 1 the number of groups n, the number of items n_i in each group i, the number of constraints m and the total number N of variables $N = \sum_{i=1}^{n} n_i$.

4.2 Experimental results

Table 2 shows the performance results of different approaches including ours. To evaluate our algorithm, we compare the results (Table 2) with existing literature approaches for the MMKP, namely respectively, Moser [4](Moser in 1997), Iqbal *et al.* [21](Ant in 2010), Htiouech *et al.* [13](OSC in 2013) and Xia *et al.* [41](SLS in 2015). The values in bold indicate that our results are greater than or equal to the best results. The available resources vector is decreased by a factor $f = 5\%$ and given to CA for more flexibility to satisfy agents requests. We conduct experiments on a intel i5 2.3

Table 2. Solution Quality Comparison

$Inst$	$Moser$	Ant	Osc	SLS	$AMMKP$	$\#agent$
I01	151	173	173	173	**173**	1
I02	291	364	364	364	**364**	1
I03	1464	1602	1594	1602	1594	2
I04	3375	3569	3514	3592	**3592**	2
I05	3905.7	3905.7	3905.7	3905.7	**3905.7**	3
I06	4115.2	4799.3	4799.3	4799.3	**4799.3**	3
I07	23556	24159	24162	24311	24310	5
I08	35373	36240	36405	36463	**36530**	8
I09	47205	48367	48567	48580	**48711**	10
I10	58648	60475	60858	60661	**60911**	12
I11	70532	72558	73022	72778	**73200**	15
I12	82377	84707	85284	84889	**85338**	18
I13	94166	96834	97545	97082	**97744**	20

Table 1. Instances details

$Inst$	n	n_i	m	N
I01	5	5	5	25
I02	5	10	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	1500
I09	200	10	10	2000
I10	250	10	10	2500
I11	300	10	10	3000
I12	350	10	10	3500
I13	400	10	10	4000

GHz machine. All procedures were implemented in Java, we use the Java Agent DEvelopment Framework (JADE) platform version 4.4.0 to implement agents.

The values in bold indicate that our results are greater than or equal to the best published results. We initially noted that our approaches could generate a feasible solution for all instances. We come to this success thanks to the strategic distribution of resources (subsection 3.1). A top view also shows that the quality of solutions is interesting : greater than or equal to the best solution among those Moser, Ant, Osc and SLS are found in $11/13$ cases. The exceptions being for instances $I03$ and $I07$ that achieve an objective function value 1.08% close to optimum are due to the parameters choice. This shows that our approach based on the decomposition of the initial problem in several sub-problems provides a good start for the agents which guide the search process towards the elite solution areas.

We benefit from the fairly important amount of resources that is left to select promising items not selected in their sub-problems due to lack of available resources. For large instances ($I07 - I13$), our approaches give the best results in a very short time (< 10s).

AMMKP provides solutions with total value on average equal to 0.81% close to the optimum and better performance than those obtained by Moser [4](3.62%), Iqbal *et al.* [21](1.58%), Htiouech *et al.* [13](1.03%) and Xia *et al.* (1.26%). For instance I13 (which is a very large instance of MMKP; 4000 variables), note that our approach takes more lower computational time to reach its result which is 0.69% of the optimal solution (given by CPLEX).

5. CONCLUSION AND FURTHER WORKS

In this work, an agent-based approach to solve the multidimensional multi choice knapsack problem has been proposed. Our solution seems to confirm that agent-base computing can provide important advantages in term of computational times, thanks to their ability to divide the initial MMKP instance in several subproblems.

Experimental results show that AMMKP produces a very competitive solutions quality. We believe first, that it is possible to use a multi-agent system effectively to implement a decomposition based optimization approach, second, that our approach can be adapted to solve other variants of knapsack problems successfully. Future work includes using other efficient algorithms by agents could give more flexibility and intelligence for agents.

6. REFERENCES

[1] Glover F, Laguna M. Tabu search. Boston: Kluwer Academic Publishers; (1998).

[2] Glover F, GA.Kochenberger. ”Critical event tabu search for multidimensional knapsack problems”. In: Osman IH, Kelly JP, editors. Metaheuristics: theory and applications. p. 407-27. (1996)

[3] Hanafi S, Freville A. An efficient tabu search approach for the 0-1 multidimensional knapsack problem. European Journal of Operational Research (1998).

[4] Moser M, Declarative scheduling for optimally graceful QoS degradation, in Proc. IEEE Int. Conf. Multimedia Computing Systems (ICMCS), pp. 86 93(1996).

[5] L. Chen, The utility model applied to layer-coded sources, Dept. of Computer Science, University of Victoria, Victoria, BC, Canada, (1998).

[6] Khan S, Quality adaptation in a multisession multimedia system : Model, algorithms and architecture, Ph.D. dissertation, Department of Electrical and Computer Engineering, University of Victoria, Canada 1998.

[7] Khan S, K. F. Li and E.G. Manning. The Utility Model for Adaptive Multimedia System. In International Workshop on Multimedia Modeling, pages 111-126 (1997).

[8] Garey MR, Johnson DS. Computers and intractability: a guide to the theory of NP completeness. San Francisco: Freeman; (1979).

[9] Martello S, Toth P. Knapsack problems. New York: Wiley; (1990).

[10] Healy Jr. WC. Multiple choice programming. Operations Research;12:122. (1964)

[11] Lin EYH. Multiple choice programming: a state-of-the-art review. International Transactions in Operational Research;1:409-421. (1994)

[12] Lin EYH. A bibliographical survey on some well-known non-standard knapsack problems.;36(4): 274 317 INFOR (1998)

[13] Htiouech. S, Sadok Bouamama and Rabeh Attia: Using surrogate information to solve the multidimensional multi-choice knapsack problem. IEEE Congress on Evolutionary Computation 2013: 2102-2107

[14] Peterson CC. Computational experience with variants of the Balas algorithm applied to the selection of research and development projects. Management Science 13:736 50.(1967)

[15] Chu PC, Beasley JE. A genetic algorithm for the multidimensional knapsack problem. Journal of Heuristics;4: 686. (1998)

[16] Bertsimas D, Demir R. An approximate dynamic programming approach to multidimensional knapsack problems. Management Science;48(4):550-65 (2002)

[17] Mhand Hifi, Slim Sadfi, Abdelkader Sbihi. ”An exact algorithm for the multiple-choice multidimensional knapsack problem”. Cahiers de la MSE b04024, Maison des Sciences Economiques, University paris pantheon-Sorbonne. (2004)

[18] Parra-Hemendez R, N.Dimopoulos. ”A New Heuristic for Solving the Multi-choice Multidimensional knapsack problem”. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions. Volume: 35 , Issue: 5 Page(s): 708-717 (2005)

[19] Kellerer H, Pferschy U, Pisinger D. Knapsack problems. Berlin: Springer; 2004.

[20] Chaitr S. Hiremath, Raymond R. Hill2. New greedy heuristics for the Multiple-choice Multi-dimensional Knapsack Problem. International Journal of Operational Research Volume 2, 2007 Pages 495-512.

[21] Shahrear Iqbal, Md. Faizul Bari, M. Sohel Rahman: Solving the Multi-dimensional Multi-choice Knapsack Problem with the Help of Ants. ANTS Conference 2010: 312-323

[22] Alaya I, C. Solnon et K. Ghedira. Des fourmis pour le sac--dos multidimensionnel. Dans : 4mes Journes Francophones de Recherche Oprationnelle (Francoro 2004), Fribourg, Suisse, pp. 159-160.

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

[24] N. Cherfi and M. Hifi. Hybrid algorithms for the multiple-choice multi-dimensional knapsack problem. International Journal of Operational Research, 5(1):89-109, 2009.

[25] Raid Mansi, Claudio Alves, J. M. Valerio de Carvalho and Said Hanafi. A hybrid heuristic for the multiple choice multidimensional knapsack problem. Engineering Optimization, 2012. DOI:10.1080/0305215X.2012.717072 .

[26] Htiouech. S, S. Bouamama, and R. Attia. 2013. ”OSC: Solving the Multidimensional Multi-Choice Knapsack Problem with Tight Strategic Oscillation Using Surrogate Constraints”. International Journal of Computer Applications 73. (13): 1-22.

[27] S.Htiouech, S.Bouamama, A Lagrangian and Surrogate information enhanced tabu search for the MMKP. 2014 IEEE World Congress on Computational Intelligence. IEEE WCCI. July 6-11, 1461-1468, Beijing, China.

[28] M. Ben Rejeb, S. Htiouech, and S. Bouamama, ” A CPU-GPU Parallel Approach to Solve Multiple Choice Multidimensional Knapsack Problem” Proceedings of META-2016 : 6th International Conference on Metaheuristics and Nature Inspired computing. P154-156

[29] H. Pirkul. Efficient algorithms for the capacitated concentrator location problem. Computers and Operations Research, 14(3) :197-208, 1987.

[30] M. Akbar, O. Ergun, and A. Punnen. Heuristic solutions for the multiplechoice multi-dimensional knapsack problem. In International Conference on Computer Science, San Francisco, USA, 2001.

[31] Mansi, R., C. Alves, J. M. Valerio de Carvalho, and S. Hanafi. 2013. A Hybrid Heuristic for the Multiple Choice Multidimensional Knapsack Problem. Engineering Optimization 45 (8): 983-1004.

[32] H Shojaei, TH Wu, A Davoodi, and T Basten, ”A pareto-algebraic framework for signal power optimization in global routing,” in Proceedings of the 16th ACM/IEEE international symposium on Low power electronics and design, Austin, Texas, USA, 2010, pp. 407 - 412.

[33] Stefan Vo, Eduardo Lalla-Ruiz (2016) A set partitioning reformulation for the multiple-choice multidimensional knapsack problem, Engineering Optimization, 48:5, 831-850, DOI: 10.1080/0305215X.2015.1062094

[34] Shanping Qiao, Ling Zhao, Yongzheng Lin . Shixian Wang. A Distributed Algorithm for 0-1 Knapsack Problem Based on Mobile Agent. Intelligent Systems Design and Applications, 2008. ISDA ’08. Eighth International Conference on. DOI: 10.1109/ISDA.2008.110. Nov 2008.

[35] Brent A. Smolinski. Approximating the 0-1 Multiple Knapsack Problem with Agent Decomposition and Market Negotiation. International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems IEA/AIE 2000: Intelligent Problem Solving. Methodologies and Approaches p 296-306.

[36] Chen, Y., and J.-K. Hao. 2014. A Reduce and Solve Approach for the Multiple-Choice Multidimensional Knapsack Problem. European Journal of Operational Research 239 (2): 313-322.

[37] J. ferber, Multi-agent systems : an Introduction to distributed artificial intelligence, new York: addison-wesley, 1999.

[38] J.Liu, H. jing, Y.Y . Tang. Multi-agent oriented constraint satisfaction, Artificial Intelligence, Vol. 136, No.1, 101-144 2002.

[39] J Han, J Pei, and M Kamber, Data mining : concepts and techniques.: Burlington : Elsevier Science, 2011.

[40] CS Hiremath and RR Hill, ”First-level tabu search approach for solving the multiple-choice multidimensional knapsack problem,” International Journal of Metaheuristics, vol. 2, pp. 174-199, 2013.

[41] Y Xia, C Gao, and JL Li, ”A Stochastic Local Search Heuristic for the Multidimensional Multiple-choice Knapsack Problem,” Communications in Computer and Information Science, vol. 562, pp. 513-522, December 2015.