

A Multidimensional Multiple-Choice Knapsack Model for Resource Allocation in a Construction Equipment Manufacturer Setting Using an Evolutionary Algorithm

Alejandra Duenas, Christine Martinelly, G. Tütüncü

► **To cite this version:**

Alejandra Duenas, Christine Martinelly, G. Tütüncü. A Multidimensional Multiple-Choice Knapsack Model for Resource Allocation in a Construction Equipment Manufacturer Setting Using an Evolutionary Algorithm. IFIP International Conference on Advances in Production Management Systems (APMS), Sep 2014, Ajaccio, France. pp.539-546, 10.1007/978-3-662-44739-0_66 . hal-01388592

HAL Id: hal-01388592

<https://hal.inria.fr/hal-01388592>

Submitted on 27 Oct 2016

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.



A multidimensional multiple-choice knapsack model for resource allocation in a construction equipment manufacturer setting using an evolutionary algorithm

Alejandra Duenas¹, Christine Di Martinelly², G.Yazgi Tütüncü^{2,3}

¹IESEG, School of Management (LEM-CNRS), Socle de la Grande Arche,
1 Parvis de la Défense, 92044 Paris France

²IESEG School of Management (LEM-CNRS), 3 rue de la Digue, 59000 Lille, France

³ İzmir University of Economics, Department of Mathematics, Sakarya cad. No:156, Balçova,
İzmir, Turkey

{a.duenas, c.dimartinelly, y.tutuncu}@ieseg.fr
yazgi.tutuncu@ieu.edu.tr

Abstract. This paper presents an approach to production resource allocation. The approach is applied to a real-world problem within the construction equipment manufacturing industry. A multidimensional knapsack problem formulated; was the proposed model being based on an evolutionary algorithm using a three-dimensional binary-coded chromosome. Various tests were carried out to show the appropriateness of the solution. The experiment results suggest to be satisfactory from the manufacturing company perspective.

Keywords: Multidimensional multiple-choice knapsack problem, Evolutionary Algorithm, Combinatorial Optimization, Resource Allocation, Heuristics

1 Introduction

This paper presents a real-world existing problem, which occurred in a construction equipment manufacturer. A variety of complex products is produced sequentially on the high-variety assembly line; the components are added as the semi-finished products move from station to station. Component specifications are dependent upon the products being assembled. Component assembly is made on-site on dedicated machines and requires specific manpower competencies. While the demand for components in number is stable, the workload of the component assembly department is subject to high variations; this causes dissatisfaction among the personnel, increased overtime work and delays on the main production line.

In the context described above, the optimization of the assignment of personnel to the different components, so as to level the workload and minimize operational disturbances on the main assembly lines, is addressed using an evolutionary algorithm.

The described assignment problem is known as the generalized assignment problem (GAP). The problem has been and is of acute interest for researchers as it has various applications in a wide range of areas from vehicle routing, to assigning jobs to computers in a network and it is known to be NP-hard [1, 2].

Researchers are thus interested in finding, in an efficient way, feasible and near optimal solutions. A wide range of techniques have been developed to solve the problem. Cattrysse et al. [3] provided a survey of branch-and-bound techniques and LP relaxation used to solve the GAP. However, the size of this combinatorial optimization problem makes it well suited for meta-heuristics solution methods. Various techniques have been applied such as meta-heuristics based on a single solution approach (tabu search [4], greedy heuristics [5]) or meta-heuristics based on a search within a population of solutions (genetic algorithms [6], bees algorithms [7]). Hybrid approaches combining exact and heuristics methods have also been developed to solve this combinatorial optimization problem [8].

Multiple variants of the GAP [9] exist, which differ from objective functions, constraints and dimensions (or indices). The efficiency of the pre-listed methods is dependent upon the characteristics of the problem.

In this paper, the problem described is characterized as a three-dimensional (3D) assignment problem [10, 11] to which we apply an evolutionary algorithm. Section 2 contains the definition of the (real-world) problem, based on the operators' allocation of jobs. The evolutionary algorithm proposed to solve the resource allocation problem is presented in Section 3. The results analysis is presented and discussed in Section 4. Finally, conclusions and directions for future work are outlined in Section 5.

2 Multidimensional multiple-choice knapsack problem

The objective of the proposed model, elaborated in collaboration with a construction equipment manufacturer, is to minimize the longest completion time.

A final product is composed of m/c types of components; the component assembly department is composed of m machines ($m = 1, \dots, M$); each machine is dedicated to the assembly of one type of component. Every day, the main assembly line produces E construction equipment (excavators). As a result, the demand for the component assembly department on the previous day is $E * M = C$ components. For instance, if 6 products are being produced on day $d+1$ on the main assembly line and the component assembly department is composed of 7 machines, the component assembly department must have assembled $6*7$ components on day d . The number of components constitutes the items to schedule. The assembly time of component c of product e on machine m is denoted as a_{cem} ; which can then be further simplified as the processing time of component c on machine m , a_{cm} (the component m of product e can only be produced on machine m).

The number of employees available in the component assembly department is fixed to O . Each operator o ($o = 1, \dots, O$) possesses different competencies on each machine, which depends on his level of training, k_{om} . The processing time of component c of product e on machine m (or component c) by operator o is denoted p_{ocm}

and is equal to $k_{om} * a_{cm}$. Each operator o can be reallocated and transferred from machine to machine in order to fulfil assembly activities according to his competencies, as long as it does not exceed a maximum number of working hours, W_o .

The generalized assignment problem can be described using the terminology of knapsack problems [12]. It has therefore been decided to model this allocation problem as a multidimensional multiple-choice knapsack problem (MMKP). The MMKP is a variant of the 0-1knapsack problem which is an NP-Hard problem [13]. Table 1 presents the MMKP notation used in this model.

Table 1. MMKP Notation

Sets		Indices	
O	Set of operators in the component assembly department	o	A operator
M	Set of machines	m	A machine
C	Set of components	c	A component
Parameters			
a_{cm}	Assembly time of component c on machine m		
k_{om}	Competencies of operator o to work on machine m		
p_{ocm}	Processing time for component c being produced on machine m by operator o		
W_o	Maximum number of working hours of operator o		
Variable			
	$x_{ocm} \begin{cases} 1 & \text{if operator } o \text{ is selected to produce on machine } m \text{ component} \\ 0 & \text{otherwise} \end{cases}$		

The mathematical formulation of the MMKP with the objective of minimizing the longest completion time is as follows:

$$\min \max_m \sum_{o=1}^O \sum_{c=1}^C p_{ocm} * x_{ocm} \quad (1)$$

Subject to

$$\sum_{m=1}^M \sum_{c=1}^C p_{ocm} * x_{ocm} \leq W_o, \forall o = 1, \dots, O \quad (2)$$

$$\sum_{o=1}^O \sum_{m=1}^M x_{ocm} = 1, \forall c = 1, \dots, C \quad (3)$$

$$x_{ocm} \in \{0, 1\}$$

Where equation (2) represents the constraint that operators cannot work longer than W_o hours per day. Equation (3) represents the constraint that only one operator can be assigned to one machine at a time.

Figure 1 presents a schematic representation of the binary-coded chromosome, considering o different operators, m different machine and c different components. This representation is similar to the representation proposed in [14].

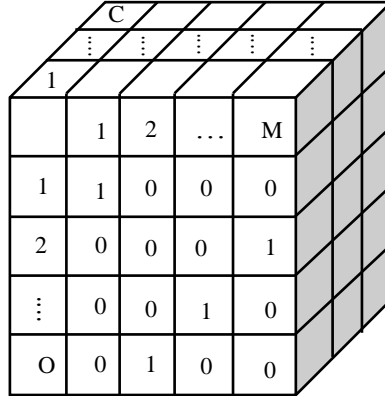


Fig. 1. Binary-coded chromosome for the MMKP

3 Evolutionary Algorithm

Evolutionary algorithms (EAs) are instances of algorithms that work with evolutionary principles. An EA is a search algorithm, inspired by natural selection and genetics that uses a population of possible solutions (candidate solutions) instead of a single solution. The candidate solutions are usually represented as strings (chromosomes) and they are evaluated by an objective (fitness) function. The search is iterative, where better solutions are generated in each iteration after applying certain genetic operators (selection, recombination, mutation etc.) [15].

Due to the problem formulation and especially due to constraint (3), it was decided to propose an EA where only a mutation operator is applied without using recombination in order to keep constraint violations to a minimum. As demonstrated by Hesser and Manner [16], the mutation operator can be considered as a search operator in itself. Hence, Figure 2 shows the EA's pseudocode with selection and mutation operators:

```

begin
     $t \leftarrow 0$ 
    initialise  $P(t)$ 
    evaluate  $P(t)$ 
    while (not termination-condition) do
        begin
             $t \leftarrow t + 1$ 
            select  $P(t)$  from  $P(t - 1)$ 
            mutate  $P(t)$ 
            evaluate  $P(t)$ 
        end
    end

```

Fig. 2. EA's pseudocode

The EA's initial population is generated making sure that all individuals are feasible (i.e., meet constraints (2) and (3)).

To determine the position of undergoing mutation a uniform random choice is used, so each position has the same probability of mutation p_m ; where, p_m is defined as the probability of independently inverting the value assigned to operator o from 0 to 1 or from 1 to 0. Once the inversion is performed, a repair algorithm is applied to make sure that only one operator is allocated a value of 1 as shown in Figure 3.

In this way, the mutated offspring meets the constraint that only one operator can be assigned to one machine at a time eq. (3) but might generate infeasible solutions with respect to the constraint that operators cannot work longer than W_o hours per day eq. (2).

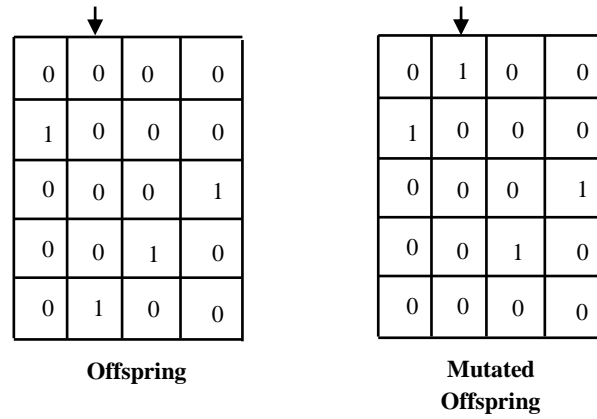


Fig. 3. Mutation operator for the MMKP

In most applications of population-based search methods (such as EA) to constrain optimization problems the penalty function approach of handling constraints is used. In this way comparisons between two feasible solutions, one feasible and one infeasible solution, and two infeasible solutions are possible. In this paper, it was decided to use the constraint handling approach presented by Deb [17]. Therefore, a tournament selection operator is used where two solutions (mutated offsprings) are compared enforcing the following criteria: any feasible solution is preferred to any infeasible solution; from two feasible solutions the one with the better objective function value is preferred; and from two infeasible solutions the one with the smaller constraint violation is preferred. Consequently, in the comparison of two infeasible solutions only the constraint violation is used without having to compute the objective function value [17].

$$F(\bar{x}) = \begin{cases} f(\bar{x}) & \text{if } \bar{x} \text{ is feasible} \\ f_{max} + g(\bar{x}) & \text{otherwise} \end{cases} \quad (4)$$

where $f(\bar{x})$ represents the objective function value given by eq. (1), f_{max} is the objective function value of the worst feasible solution in the population and $g(\bar{x})$ represents the constraint violation defined as follows:

$$g(\bar{x}) = \sum_{o=1}^O (W_o - \sum_{m=1}^M \sum_{c=1}^C p_{ocm} * x_{ocm}) \quad (5)$$

4 Results Analysis

The problem under consideration is a real-world production resource allocation problem defined in collaboration with a construction equipment manufacturer. In this problem, 18 operators with different skills are to be allocated to 7 machines in order to produce 7 different excavators per day.

Table 2 presents the results obtained when running the EA using the approach presented. The initial parameter specifications of the EA are as follows:

Population size (N_p): 50,

Tournament size (T): 2,

W_o = 8 hours a day,

Table 2. Results found by the EA

Number of simulation runs A	Operator number with constraint violations (Constraint (3))	Violation constraint value
100	1,2,4,5,8,9,12,14,15	8.4714
300	6,7,8,12,13,18	7.4206
500	1,4,5,16,17,18	4.296
800	2,6,11,12	3.1923
1000	2,16,18	1.63205
1500	3,15,18	1.50405

From Table 2, it can be observed (as expected) that the violation constraint value decreases as the number of simulations increases. It is also shown that the number of operators which have a constraint violation may decrease, because the constraint violation is related to the fact that operators cannot work longer than W_o hours per day. This, in economic terms, means that the company has to pay overtime to those operators with a constraint violation, having a direct impact on the company's profit.

Furthermore, when comparing the algorithm run 300 and 500 simulations, it is possible to see that the number of operators with a constraint violation remains the same. However, the violation constraint value for 500 simulations is almost 50% lower than the value found for 300 simulations. This shows that the EA has not been defined to decrease the number of operators with a constraint violation but to decrease the violation constraint value. In future work, the minimization of number of operator with a constraint violation can be considered as a second objective resulting in a multiple objective optimization problem.

Finally, when comparing the violation constraint values between 1000 and 1500, it can be seen that the difference between them is not as big as in other instances. This suggests that it may not be advisable to run the EA for a bigger number of simulations since the results might not improve after a certain number of iterations.

5 Conclusions

In this paper, an approach to solve the MMKP is presented. The approach is applied to a real-world problem of a construction equipment manufacturer with the aim of allocating different operators to different components to be processed on different machines. The main characteristic of the problem is that the operators' different skills and competences need to be considered in the operators' processing times.

The MMKP is an NP-complete problem therefore an EA was developed to solve the problem. The MMKP coding considered is a 3D binary-coded chromosome which is then transformed to a 2D representation in order to apply 2D crossover and mutation operators. The EA's fitness function is represented by the sum of the objective function and a penalty term. The proposed EA seems to have good performance.

Future work should include the variant where two or more operators can be allocated to the same component and the problem solution when more than one objective is simultaneously considered. Another aspect to analyze is the uncertainty related to the processing time definition since no historical data is available. The use of fuzzy sets for modeling this uncertainty could also be interesting for future research.

References

1. Ramanan, P., J.S. Deogun, and C.L. Liu, *A personnel assignment problem*. Journal of Algorithms, 1984. **5**(1): p. 132-144.
2. Fisher, M.L., R. Jaikumar, and L.N.V. Wassenhove, *A Multiplier Adjustment Method for the Generalized Assignment Problem*. Management Science, 1986. **32**(9): p. 1095-1103.
3. Cattrysse, D.G. and L.N. Van Wassenhove, *A survey of algorithms for the generalized assignment problem*. European Journal of Operational Research, 1992. **60**(3): p. 260-272.
4. Díaz, J.A. and E. Fernández, *A Tabu search heuristic for the generalized assignment problem*. European Journal of Operational Research, 2001. **132**(1): p. 22-38.
5. Romeijn, H.E. and D.R. Morales, *A class of greedy algorithms for the generalized assignment problem*. Discrete Applied Mathematics, 2000. **103**(1-3): p. 209-235.
6. Chu, P.C. and J.E. Beasley, *A genetic algorithm for the generalised assignment problem*. Computers & Operations Research, 1997. **24**(1): p. 17-23.

7. Özbakir, L., A. Baykasoğlu, and P. Tapkan, *Bees algorithm for generalized assignment problem*. Applied Mathematics and Computation, 2010. **215**(11): p. 3782-3795.
8. Woodcock, A.J. and J.M. Wilson, *A hybrid tabu search/branch & bound approach to solving the generalized assignment problem*. European Journal of Operational Research, 2010. **207**(2): p. 566-578.
9. Pentico, D.W., *Assignment problems: A golden anniversary survey*. European Journal of Operational Research, 2007. **176**(2): p. 774-793.
10. Geetha, S. and M.N. Vartak, *The three-dimensional bottleneck assignment problem with capacity constraints*. European Journal of Operational Research, 1994. **73**(3): p. 562-568.
11. Gilbert, K.C. and R.B. Hofstra, *multidimensional assignment problems*. Decision Sciences, 1988. **19**(2): p. 306-321.
12. Martello, S. and P. Toth, *Knapsack problems* 1990: Wiley New York.
13. Hifi, M., M. Michrafy, and A. Sbihi, *Heuristic algorithms for the multiple-choice multidimensional knapsack problem*. Journal of the Operational Research Society, 2004. **55**(12): p. 1323-1332.
14. Tavana, M., K. Khalili-Damghani, and A.-R. Abtahi, *A fuzzy multidimensional multiple-choice knapsack model for project portfolio selection using an evolutionary algorithm*. Annals of Operations Research, 2013. **206**(1): p. 449-483.
15. Bäck, T., ed. *Introduction to evolutionary algorithms, Chapter 7*. Evolutionary Computation 1, ed. T. Bäck, Fogel, D.B. and Michalewicz, Z. 2000, Institute of Physics Publishing.
16. Hesser, J. and R. Manner, *TOWARDS AN OPTIMAL MUTATION PROBABILITY FOR GENETIC ALGORITHMS*. Lecture Notes in Computer Science, 1991. **496**: p. 23-32.
17. Deb, K., *An efficient constraint handling method for genetic algorithms*. Computer Methods in Applied Mechanics and Engineering, 2000. **186**(2-4): p. 311-338.