Les Annales RECITS (http://www.lrecits.usthb.dz)
Vol. 05, 2018, pages 31-36

An exact method for the multi-objective assignment problem

SATLA Hamou and CHERGUI Mohamed El-Amine

RECITS laboratory-USTHB-Bab Ezzouar-Alger

hsatla@usthb.dz, mchergui@usthb.dz

Abstract: In this paper, we consider the multi-objective assignment problem with at least two criteria. First, we present some works which has treated the problem. Second, we call back a mathematic formulation of the problem and describe a branch and bound based method to generate the set of all efficient solutions. The present work is in progress, an example is illustrated in detail and the computational experiment is on track to be able to implement the algorithm.

Keywords: Assignment problem; multi-objective linear integer programming; branch and bound; non dominated vector; efficient solution.

Résumé : Dans cet article, nous considérons le problème de l'affectation multi-objectif avec au moins deux critères. Premièrement, nous présentons quelques travaux qui ont traité le problème. Ensuite, nous rappelons une formulation mathématique du problème et décrivons une méthode basée sur le principe de séparation et évaluation pour générer l'ensemble de toutes les solutions efficaces. Le présent travail est en cours, un exemple est illustré en détail et l'expérimentation numérique est en bonne voie pour pouvoir implémenter l'algorithme.

Mots clés : Problème d'affectation, programmation linéaire multi objective en nombres entiers , séparation et évaluation, vecteur non dominé, solution efficace

---

1 Introduction

The assignment problem (AP) is a classical combinatorial optimization problem, which deals with the allocation of the various resources to the various activities on one to one basis. It belongs to the class of combinatorial optimization problems that can be solved in polynomial time [2].

However, the multi-objective version (MOAP) is NP-hard [6] when the number of considered criteria in (AP) is greater than or equal to 2. To our knowledge, this problem has not received enough attention from researchers and very little work exists. In [5], a generalization of the two phase method is applied to the assignment problem with three objectives. In [2], only the bi-objective case of MOAP is treated. Approximate methods have been proposed to solve the MOAP problem ([3], [8]). In the present paper, we describe a new exact method to find all efficient solutions to the MOAP problem. To do this, the associated bipartite graph of MOAP problem is used in the branch and bound based method. The branching process is done on an edge of the obtained graph up to the current step. This has the effect to break down the initial problem into a reduced size sub problems MOAPs, each can be solved by the general method dedicated to (MOILP) described in [1].

2 Principle of method

First of all, the mathematical model of the MOAP problem can be written as follow: Given the complete bipartite graph $G(V_1 \cup V_2, E)$ where sets $V_1$and V_2 design respectively resources and activities, $n =\mid V_1 \mid=\mid V_2 \mid$

$$(\mathrm{P}) \begin{cases} Max\ Z(x) = \ (Z_k(x) = \sum_{i=1}^n \sum_{j=1}^n c_{ij}^k x_{ij})_{k=\overline{1,p}} \\ \sum_{i=1}^n x_{ij} = 1 \quad j = \overline{1,n} \\ \sum_{j=1}^n x_{ij} = 1 \quad i = \overline{1,n} \\ x_{i,j} \in \{0,1\} \qquad i,j = \overline{1,n} \end{cases}$$

$c_{ij}^k \in R : k^{th}$ weight of the edge $(ij) \in E$, $k = \overline{1,p},\ p \geq 2$.

$$x_{ij} = \begin{cases} 1 \text{ if the } i^{th} \text{ resource is assigned to the } j^{th} \text{ activity} \\ 0 \text{ otherwise, } \ i,j = \overline{1,n} \end{cases}$$

A vector $Z(x) \in R^p$ dominates another vector $Z(y) \in R^p$ if $Z(x) \geq Z(y)$ and $Z(x) \neq Z(y)$. In this case, the solution y is not efficient for (P). The ideal point I for (P) is a point from the criteria space and has $(Z_1^*, ..., Z_p^*)$ for coordinates, where Z_k^* is the weight of a maximum assignment problem for criterion k, $k = \overline{1,p}$. In general, the ideal point I is not a feasible solution [7]. However, for criterion k, $k = \overline{1,p}$, the computation of Z_k^* is an easy problem [4].

The use of the principle of branch and bound is mentioned in the following steps:

A. Branching step

The branching process is made with respect to the edges such that; $x_{ij} = 1$ or $x_{ij} = 0$. Maintaining the edge (ij) in a solution$(x_{ij} = 1)$ has a constraint propagation effect on adjacent edges that is all adjacent edges are automatically removed from the current graph and many constraints of the program (P) become saturated. All this leads to a significant decrease in the size of (P).

In the other hand, rejecting the edge (ij) $(x_{ij} = 0)$ has the effect to delete it from the current graph and in this case too, the size of (P) decreases.

An heuristic can be adapted to select an edge for the branching step.

B. Vector bounds step

At each node l of the search tree, two bounds are calculated. The upper bound corresponds to the vector ideal point I_l in the sub domain of (P) of node l, which is easy to calculate. Indeed, for each criterion k, $k = \overline{1,p}$, the polynomial algorithm described in [4] for an assignment problem is used to calculate the optimal value Z_k^{*l}.

The lower bound consists of the set ND_l of all the non-dominated vectors of the solution previously generated up to node l.

Thus, the node l is fathomed if the vector ideal point I_l is dominated by at least one vector of the set ND_l.

C. Principle

The process will continue with the branching and vector bounding steps at each node of the tree search following the principle depth first strategy, with the assumption that the number of levels measuring the depth of the tree search is fixed beforehand to avoid the combinatorial explosion.

In the worst case, to reduce the matrix of constraints to a null matrix using the branching steps, we will have $2^{((n(n-1))/2)}$ leaves in the tree search when the density of the graph becomes zero (in this case the graph is a set of isolated vertices). Therefore, fixing the depth of the tree search has the effect of avoiding the exploration of $2^{((n(n-1))/2)}$ leaves.

Thus, each leaf of the tree search is associated to a sub graph of G of non-zero density, corresponding to a program of reduced size compared to the initial program (P). In this case we can use one of the methods known in the literature to solve the small size corresponding MOILP program.

Example 1 *Consider the complete bipartite graph $G = (V_1 \cup V_2, E)$ where $V_1 = V_2 = 1, 2, 3, 4$. With each edge is associated a weighting vector of dimension three, as indicated in the following matrix:*

$$\begin{bmatrix} (2,1,3) & (3,2,5) & (9,4,2) & (1,1,3) \\ (4,3,2) & (2,1,6) & (5,1,1) & (4,3,2) \\ (2,1,1) & (1,1,4) & (2,1,3) & (2,2,2) \\ (1,3,5) & (2,2,10) & (5,2,3) & (3,1,3) \end{bmatrix}$$

We put $h = 2$.
$l = 0$; $Eff_0 = \emptyset$; $ND_0 = \emptyset$; $ND = \emptyset$;
Branching step: Select an edge $(ij) \in E$ such that, for example: $c_{ij} = max\{\sum_{k=1}^{3} c_{yz}^{k} \backslash yz \in E\}$.
$(ij) = (13)$;
At level one, the branching is done with respect to the edge 13 creating two nodes: node 1 and node 2. At level two, we continue the branching from node 1 and node 2 with respect to the new edge founded 42 for the two nodes, creating four leafs: nodes 3 and 4, and nodes 5 and 6 respectively to the nodes 1 and 2. At node 3, we obtain a graph of density non zero. By applying The method described in [1], we find one solution:
$S_1 = \{13, 42, 21, 34\}$; $Z(S_1) = (17, 11, 16)$; $Eff_3 = \{S_1\}$; $ND_3 = \{Z(S_1)\}$
$Eff = \{S_1\}$ *and* $ND = \{Z(S_1)\}$. *The node is fathomed .*
At the node 4, the ideal point $I_4 = (17, 11, 15)$ corresponding is dominated by $Z(S_1)$. Therefore, the node is fathomed. At the node 5, the method described in [1] is used to solve the sub problem founded, it returns the following solutions:
$S_2 = \{11, 23, 34, 42\}; Z(S_2) = (11, 6, 16)$;
$S_3 = \{11, 24, 33, 42\}; Z(S_3) = (10, 7, 18)$;
$Eff_5 = \{S_2, S_3\}; ND_5 = \{Z(S_2), Z(S_3)\}$;
Note that $Z(S_2)$ is dominated by $Z(S_1)$, then:
$Eff = \{S_1, S_3\}; ND = \{Z(S_1), Z(S_3)\}$. *The node is fathomed.*
At the node 6, by applying the method described in [1], the solutions founded are presented as follow:
$S_4 = \{12, 24, 31, 43\}; Z(S_4) = (14, 8, 9)$;
$S_5 = \{12, 24, 33, 41\}; Z(S_5) = (10, 9, 15)$;

$S_6 = \{14, 22, 33, 41\}; Z(S_6) = (6, 6, 17)$;
$Eff_6 = \{S_4, S_5, S_6\}; ND_6 = \{Z(S_4), Z(S_5), Z(S_6)\}$; *the node is fathomed.*
Note that both vectors $Z(S_4)$ *and* $Z(S_5)$ *are dominated by the vector* $Z(S_1)$. *Hence, sets* Eff *and* ND *are updated:*
$Eff = \{S_1, S_3, S_6\}$;
$ND = \{Z(S_1), Z(S_3), Z(S_6)\}$.

3 Conclusion

A branch and bound based method is developed for the multi-objective assignment problem. It makes it possible to generate all the efficient solutions by adopting a decomposition strategy of the initial problem into disjoint problems of reduced sizes, each of them being solved by a general method reported in the literature and dedicated to the problem MOILP. This strategy based on the principle of divide and conquer seems better suited compared to a general method MOILP in view of the first results obtained from the experiment computation.

References

[1] Mohamed El-Amine Chergui, Mustapha Moulaï et Fatma Zohra Ouaïl, "Solving the Multiple Objective Integer Linear Programming Problem", Modelling, Computation and Optimization in Information Systems and Management Sciences, Communications in Computer and Information Science, 2008, Volume 14, Part 1, Part 1, 69-76, Springer.

[2] Matthias Ehrgott, Xavier Gandibleux, and Anthony Przybylski, "Exact Methods for Multi-Objective Combinatorial Optimization". In: Greco S., Ehrgott M., Figueira J. (eds) Multiple Criteria Decision Analysis. International Series in Operations Research & Management Science, vol 233, 2016. Springer

[3] Abdul Munaam Kadhem Hammadi, "Solving multi objective Assignment problem using Tabu search algorithm", Global Journal of Pure and Applied Mathematics 13.9 (2017): 4747-4764.

[4] H. W. Kuhn, "The Hungarian Method for the Assignment Problem", Naval Research Logistic Quarterly, vol. 2, pp. 83-97, 1955.

[5] Anthony Przybylski, Xavier Gandibleux, and Matthias Ehrgott, "A two phase method for multi-objective integer programming and its application to the assignment problem with three objectives". Discrete Optimization 7.3 (2010): 149-165.

[6] P. Serafini, "Some considerations about computational complexity for multiobjective combinatorial problems". In J. Jahn and W. Krabs, eds., Recent Advances and Historical Development of Vector Optimization, volume 294 of Lecture notes in Economics and Mathematical Systems. Springer Verlag, Berlin, 1986

[7] Ralph E. Steuer, "Multiple Criteria Optimization: Theory, Computation and Application", New York : John Wiley and Sons, 1986.

[8] Anita Ravi, Tailor, and Jayesh M. Dhodiya, "A Genetic Algorithm based Hybrid Approach to Solve Multi-objective Interval Assignment Problem by Estimation Theory". Indian Journal of Science and Technology 9.35 (2016).