International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Intelligent Assignment Problem Using Novel Heuristic: The Dhouib-Matrix-AP1 (DM-AP1)

Souhail Dhouib[1]

Submitted: 15/11/2021 **Accepted : 16/02/2022**

Abstract: The assignment problem is one of the most popular optimization problems where the main objective is to find the total minimal cost for assigning *n* objects to *n* other objects. This paper presents an original heuristic, named Dhouib-Matrix-AP1, to generate an initial basic feasible solution for the classical assignment problem in very easy and fast steps. The proposed method Dhouib-Matrix-AP1 finds the optimal or a near optimal solution for the assignment problem after just n iterations with only three easy steps in each one. The first step consists in computing the total cost by rows and columns using an original formula (*Sum* – (*Min* * *n* /2)). The second step looks for selecting the greatest value (*Z*) from these total costs. Finally, the third step is based on choosing the minimal row or column matrix element which corresponds to the value *Z*. For this purpose, we generate a detailed step by step process application based on 4x4 dimensional sample. Moreover, a stepwise application of the Dhouib-Matrix-AP1 method is presented with details for three examples. Besides, the results of a set of fifteen literature examples with different dimensions are discussed. The outcomes of the study show that the proposed method provides the optimal or near optimal solutions in easier and faster manner.

Keywords: *Artificial Intelligence, Operational Research, Heuristic, Intelligent Assignment Problem, Dhouib-Matrix, Decision Support Systems.*

1. Introduction

The Assignment Problem (AP) is one of the standard combinatorial optimization problems. It is an NP-complete problem [1]. It consists of a set of n objects (jobs) allocated to a set of n objects (machines) to effectively minimize the cost (or maximize the profit), such that each job will be assigned to a unique machine and each machine is used by one and only one job. The AP is formulated as:

Minimize:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \quad (1)$$

Subject to:

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, \dots, n$$

$$\sum_{i=1}^{n} x_{ij} = 1, \quad j = 1, \dots, n \quad (2)$$

$$x_{ij} = 0 \; or \; 1, \quad i = 1, \dots, n, j = 1, \dots, n$$

Where $i = \{1,2, \dots, n\}$ is the index set of n jobs, $j = \{1,2, \dots, n\}$ is the index set of n machines, x_{ij} is a binary variable {0,1} that represents the affectation of job i to machine j and c_{ij} is the cost attributed in assigning the jth job to the ith machine. The generated solution for the AP is more explicit when it is represented as a network diagram flow as in Figure 1 which depicts the assignment of 8 jobs to 8 machines.

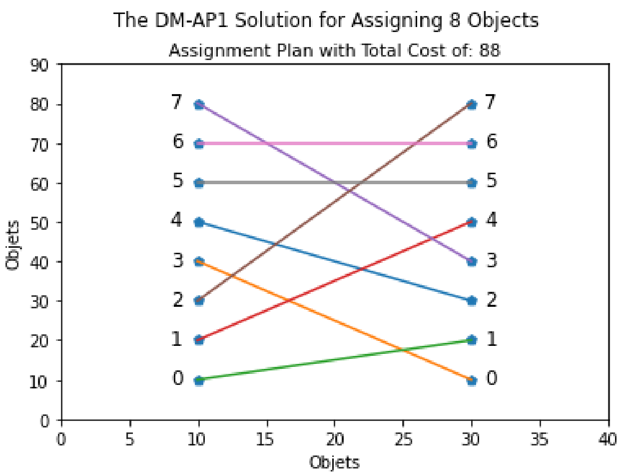

Fig 1. The assignment network diagram plan for 8 objects

Therefore, many different heuristics have been developed to solve the AP with different computational complexity in the order of: O(n3) for the Hungarian algorithm and O(n!) for the Brute Force method (where n represents the number of objects). Our aim in this paper is to design an easy and quick heuristic, in which there is only one iterative structure, to generate a good initial basic feasible solution for the AP: The Dhouib-Matrix-AP1 (DM-AP1) method. In fact, the proposed DM-AP1 method will be useful to rapidly generate a good starting point (initial feasible solution) for any standard metaheuristic such as the Variable Neighborhood Search algorithm in [13], the Simulated Annealing method [14], the

[1] *Laboratory of OLID, Higher Institute of Industrial Management, University of Sfax, Tunisia*
ORCID ID : 0000-0003-2487-0062
* *Corresponding Author Email: dhouib.matrix@gmail.com*

Threshold Accepting procedure [15] or the Record to Record Travel metaheuristic [16].
The proposed method DM-AP1 is easy to apply and it needs only three steps repeated in n iterations. There is only one iterative structure in the proposed program for finding a good initial basic feasible solution for the AP. In fact, the DM-AP1 method presents a computational complexity time in the order of O(n). Figure 2 depicts a comparison of the computational complexity between the DM-AP1, the Hungarian and the Brute Force methods for the matrix size up to 10. Obviously, the gap will increase speedily when the matrix size increases. However, that the Hungarian method generates the optimal assignment but with a greater computational time. For an example of 10 objects, the Brute Force method requests (10! = 3,628,800) iterations to find the optimal solution; whereas the Hungarian method needs (103 = 1,000) iterations; while our proposed method, the DM-AP1 method, needs just 10 iterations to generate the optimal solution or a near optimal solution. So, the decision maker has the ability to choose between the two criteria: time or quality.

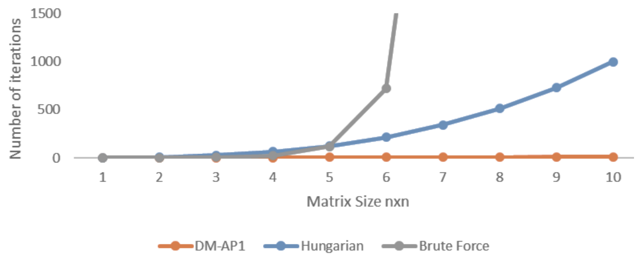

Fig 2. Comparing the computational complexity

The remainder of this paper is structured as follows. In the next section, the methodology of the DM-AP1 heuristic will be described with a stepwise based on 4x4 dimensional example. In section 3, a numerical application is illustrated on 4x4, 5x5 and 6x6 examples. In section 4, computational results on a set of fifteen numerical examples with different dimensions are discussed. In section 5, conclusions and suggestions for further research are presented.

2. Related work

In recent years, there are several papers studying the variants of AP. The Branch and Bound algorithm is designed in order to minimize the total walking distance travelled by all passengers in the airport gate assignment problem in [2]. The best traveling routes and simultaneously assigned consultants to clients using two algorithms named RMIP and MIP-based on the neighborhood search algorithm are determined in [3]. Three Mixed Integer Programming models with integrated decisions for different facility layout problems are designed in [4]. Several methods (the Hungarian algorithm, the Brute Force algorithm, the Linear Programming Algorithm and the Greedy Algorithm) are proposed to solve the AP in [5]. An iterative algorithm is used to assign the jobs to the manufacturing units in a polynomial time in [6]. The Artificial Neural Network model named the Sparse Clustered Neural Networks method is applied to solve the AP in [7].
Moreover, a generalization of the Hungarian method and the Hall's theorem is developed to solve the classical problem of determining a perfect matching in bipartite graphs in [8]. The tail AP using a mathematical programming approach for realistic cases drawn from a Spanish airline is solved in [9]. An efficient approximation algorithm to solve the partial AP where the objective is to assign items to buyers such that the total values are maximized is proposed in [10]. Moreover, the unbalanced AP using a Modified Hungarian Method is solved in [11]. The Branch-and-Bound algorithm and a Russian Doll Search algorithm to optimize the AP with additional conflict constraints in conjunction with the assignment constraints and binary restrictions is used in [12].

3. The Proposed Method: Dhouib-Matrix-AP1 (DM-AP1)

Recently, we designed and developed a new heuristic, entitled Dhouib-Matrix-TSP1 (DM-TSP1), to solve the Traveling Salesmen Problem (TSP) with crisp parameters in [17-18-19]. Furthermore, we adapted the DM-TSP1 to solve the TSP under uncertain environment in [20-21-22]; also, under neutrosophic environment in [23-24]. Moreover, we invented a new method, named Dhouib-Matrix-TP1 (DM-TP1), to solve the Transportation Problem in [25-26-27]. Besides, two metaheuristics are designed the Dhouib-Matrix-3 (DM3) and the Dhouib-Matrix-4 (DM4): DM3 is based on an iterative structure [28]; where the DM4 is founded on a Multi-Start structure based on several descriptive statistical metrics [29].
Whereas in this paper, we introduce a new column-row method, entitled the Dhouib-Matrix-AP1 (DM-AP1), to generate a good initial basic feasible solution for the AP using an original formula *(Sum – (Min * n /2))* which drives the construction process. The DM-AP1 method is composed of three simple steps (see Figure 3) repeated in just *n* iterations (where *n* represents the number of objects). The DM-AP1 is developed using Python programming language.

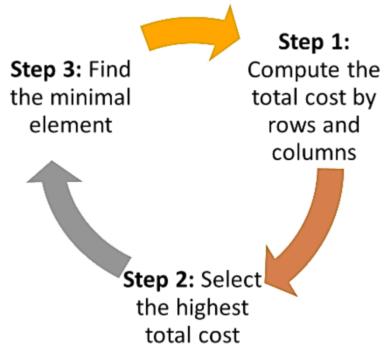

Fig 3. Flow chart of the proposed DM-AP1 heuristic

Before starting the DM-AP1 heuristic, we need to generate a balanced cost matrix: the *n*-by-*n* square matrix with non-negative elements. Then, the DM-AP1 heuristic is executed with only one iterated structure (*n* iterations) gathering three sequential steps:

- Step 1: Compute the total cost by rows and columns: Compute the total cost for each row and each column using the proposed original formula: *Sum – (Min * n /2).*

- Step 2: Select the highest total cost: Search the biggest total cost value. This value gives us the index of row or column, let us define it by *Z.*

- Step 3: Choose the minimal element: Select the smallest element in the corresponding *Z* index and discard its column and row (the smallest element represents the assignment of a job to a machine).

This example (4x4 cost matrix) may be helpful to clarify the three steps of the proposed method (see Figure 4), which needs only four (*n*=4) iterations:

14	2	7	2
5	12	8	4
8	6	3	6
7	5	9	10

Fig 4. The original 4x4 cost matrix

3.1. Iteration number 1

The proposed DM-AP1 heuristic starts by computing the total cost for each row and each column (using this formula: *Sum – (Min * n /2)*). Then, the biggest value which is 24 in column 1 is selected. Thus, the smallest element in this column which is 5 is found at position d_{21}. Next, column 1 and row 2 are discarded (See Figure 5).

14	2	7	2	21
5	12	8	4	21
8	6	3	6	17
7	5	9	10	21
24	21	21	18	

Fig 5. After iteration number 1

3.2. Iteration number 2

Identically, the same treatments are repeated. the total cost for each row and each column (using this formula: *Sum – (Min * n /2)*) is computed. Then, the biggest value which is 14 in column 4 is selected. Thus, the smallest element in this row which is 2 is found at position d_{14}. Next, column 4 and row 1 are discarded (See Figure 6).

Iteration 2: Select the element d_{14}

14	2	7	2	7
5	12	8	4	
8	6	3	6	9
7	5	9	10	14
	9	13	14	

Fig 6. After iteration number 2

3.3. Iteration number 3

Similarly, in this iteration. The total cost for each row and each column is computed and the biggest value which is 6 in column 3 is selected. Thus, the smallest element in this column which is 3 is found at position d_{33}. Next, column 3 and row 3 are discarded (See Figure 7).

Iteration 3: Select the element d_{33}

14	2	7	2	
5	12	8	4	
8	6	3	6	3
7	5	9	10	4
	1	6		

Fig 7. After iteration number 3

3.4. Iteration number 4

And finally, we add the last non assigned job for the non-assigned machine: job 4 is affected to machine 1 (See Figure 8).

Iteration 4: Select the element d_{42}

14	2	7	2	
5	12	8	4	
8	6	3	6	
7	5	9	10	-5
	-5			

Fig 8. After iteration number 4

Therefore, the DM-AP1 method finds the optimal solution 15 with a lower computational complexity (in only 4 iterations, *n*=4): assign job 2 to machine 1 with the cost of (5), assign job 4 to machine 2 with the cost of (5), affect job 3 to machine 3 with cost of (3) and finally affect job 1 to machine 4 with the cost of (2). Whereas, the Greedy method found the cost 16 and the Hungarian, the Brute Force and the Linear Programming found also the optimal solution but in more complicated and greater number of iterations. Figure 9 depicts the generated assignment network diagram plan solution using DM-AP1 heuristic which is developed under Python programming language.

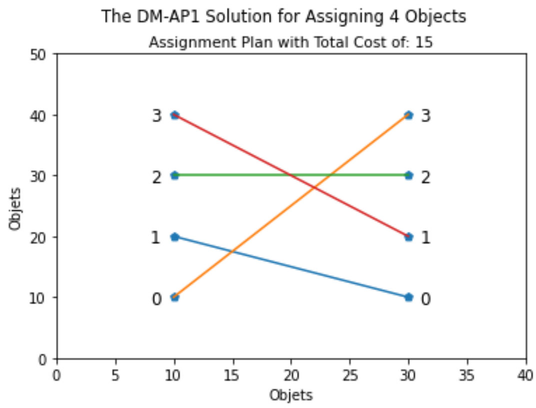

Fig 9. The generated solution by DM-AP1 heuristic for 4 objects

4. Numerical Applications

More examples will be given in this section to explain and to prove the simplicity and the efficiency of the proposed DM-AP1 method to solve the AP occurring in real life situations.

4.1. Numerical Example 1

An example of 4x4 matrix from [5] is illustrated in Figure 10. The DM-AP1 method finds the optimal solution (15) in only 4 (*n*=4) simple iterations:

14	5	8	7
2	12	6	5
7	8	3	9
2	4	6	10

Fig 10. The original distance matrix

Iteration number 1:

- Step 1: Compute the total cost for each row and each column.
- Step 2: Select the biggest cost which is 24 in row 1.
- Step 3: Choose the smallest element in this row which is 5 at position d_{12}. Thus, job 1 will be assigned to machine 2 (with

cost = 5). Then, row 1 and column 2 are discard (See Figure 11).

Iteration number 2:

- Step 1: Compute the total cost for each row and column.
- Step 2: Select the biggest cost which is 14 in column 4.
- Step 3: Choose the smallest element in this column which is 5 at position d_{24}. Thus, job 2 will be assigned to machine 4 (with cost = 5). Then, row 2 and column 4 are discard.

Iteration number 3:

- Step 1: Compute the total cost for each row and column.
- Step 2: Select the biggest which is 5 in column 1.
- Step 3: Choose the smallest element in this row which is 2 at position d_{41}. Thus, job 4 will be assigned to machine 1 (with cost = 2). Then, row 4 and column 1 are discard.

Iteration number 4:

- There is only one element in position d_{33}, so assign job 3 to machine 3 (with cost = 3).

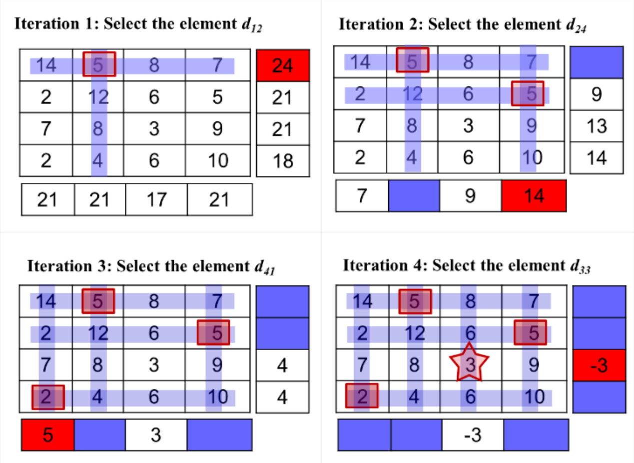

Fig 11. Just four iterations to solve 4x4 matrix

Figure 12 depicts the generated assignment network diagram plan for the initial basic feasible solution using the DM-AP1 heuristic (developed under Python programming language).

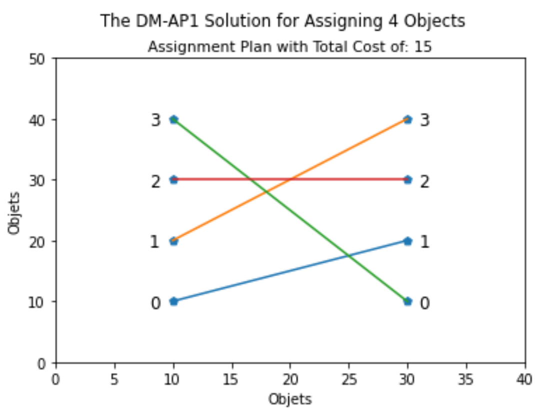

Fig 12. The generated solution using the DM-AP1 heuristic

4.2. Numerical Example 1

In Figure 13, the original cost 5x5 matrix from [30] is exposed where the optimal assignment cost is 9.

8	4	2	6	1
0	9	5	5	4
3	8	9	2	6
4	3	1	0	3
9	5	8	9	5

Fig 13. The original 5x5 matrix

Well, the DM-AP1 heuristic starts by computing the total cost for each row and each column then selecting the biggest value which is 26 in row 4. Thus, the smallest element in this column which is 5 is found at position d_{52}. Next, row 5 and column 2 are discarded etc. Figure 14 presents the stepwise of the algorithm.

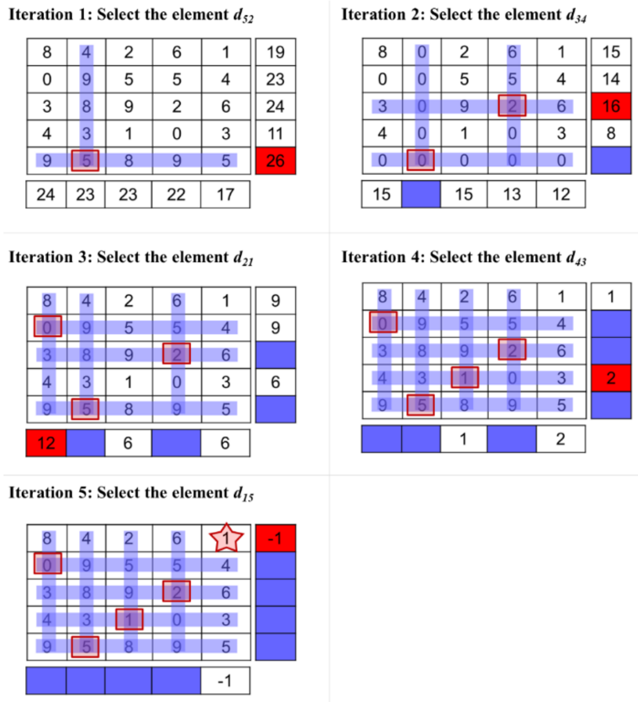

Fig 14. Just five iterations to solve 5x5 matrix

Our method found the optimal result 9 precisely in 5 iterations (n=5): assign job 1 to machine 5 (with cost = 1), assign job 2 to machine 1 (with cost = 0), assign job 3 to machine 4 with (cost = 2), affect job 4 to machine 3 with (cost = 1) and finally affect job 5 to machine 2 (with cost = 5).

The generated assignment network diagram plan solution using the DM-AP1 heuristic (developed under Python programming language) is presented in Figure 15.

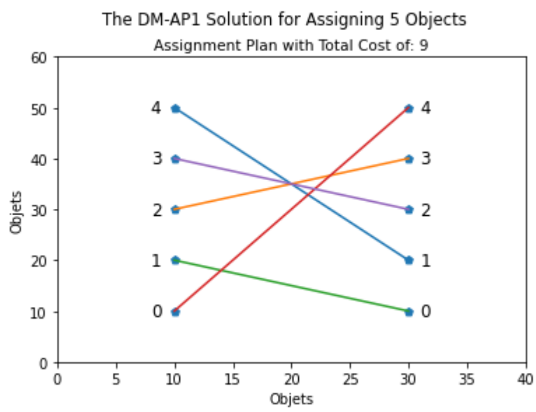

Fig 15. The generated solution using DM-AP1 heuristic for 5 objects

4.3. Numerical Example 3

Figure 16 presents a case study of a 6x6 processing time matrix from [31], where a manager needs to assign six different jobs (with different processing times) to six different machines in order to minimize the total processing time.

10	15	12	18	14	13
17	14	22	16	19	20
12	15	13	8	12	9
11	16	15	22	21	18
13	10	17	19	15	10
15	8	14	25	16	18

Fig 16. The original processing time matrix

[31] solved this problem using the Hungarian method (with a complexity of n^3=63=216) and found the optimal solution (68). Whereas, our method DM-AP1 found easily (in just n=6 iterations) the near optimal solution (71), with a deviation of 00.04%, by assigning 3-4 ; 2-2 ; 4-1 ; 5-6 ; 1-3 ; 6-5 (see Figure 17).

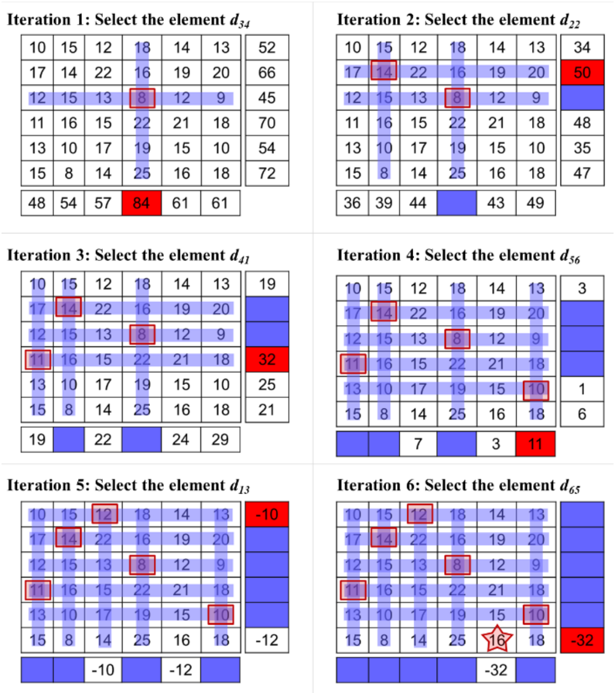

Fig 17. Just six iterations to solve 6x6 matrix

The generated assignment network diagram plan for the found solution using the DM-AP1 heuristic is presented in Figure 18.

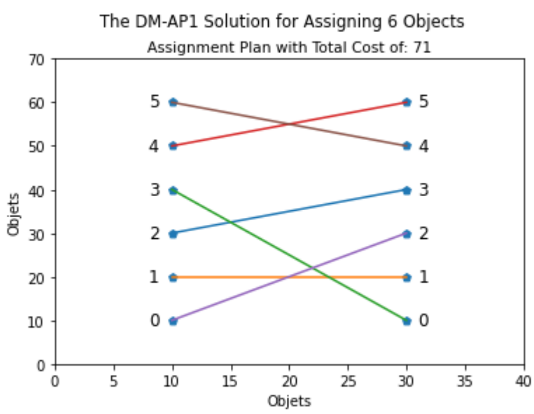

Fig 18. The generated solution using the DM-AP1 heuristic for 6 objects

In this section, the performance of the proposed method DM-AP1 is verified by carrying out a comparative study. Fifteen samples cost minimizing assignment problems of different sizes, selected from different journals mentioned in Table 1 are exploited to illustrate the performance of the proposed DM-AP1 method compared to the Hungarian method. The deviation is computed by comparing the solution found by the DM-AP1 (Solution) to the optimal solution (Optimal) using this formula: *Dev = (Solution - Optimal)/Optimal*.

Table 1. A set of fifteen examples from the literature

Problem Number	Source	Hungarian	DM-AP1	Dev %
Exp. 1	[32]	15	15	00.00
Exp. 2	[32]	127	127	00.00
Exp. 3	[33]	13	13	00.00
Exp. 4	[34]	24	24	00.00
Exp. 5	[34]	23	23	00.00
Exp. 6	[35]	84	85	00.01
Exp. 7	[35]	369	373	00.01
Exp. 8	[36]	14	14	00.00
Exp. 9	[36]	21	26	00.24
Exp. 10	[36]	14	14	00.00
Exp. 11	[37]	64	69	00.08
Exp. 12	[37]	202	212	00.05
Exp. 13	[31]	266	272	00.02
Exp. 14	[38]	88	88	00.00
Exp. 15	[39]	1422	1865	00.31

From table 1, we find a very low total average deviation equal to 00.05% with a small standard deviation equal to 10%. Thus, our DM-AP1 heuristic is concurrent to the Hungarian method. However, for larger instances, the DM-AP1 method generates very rapidly (computational complexity time in the order of *O(n)*) a good initial basic feasible solution while the Hungarian method generates the optimal solution but with more complicated iterations and more computational complexity (in the order of $O(n^3)$). For the example number 15 (Exp. 15), with a matrix of 12x12 cost matrix, the DM-AP1 finds the solution 1865 after just 12 iterations with a deviation of 00.31% from the optimal solution. Whereas, the Hungarian method finds the optimal solution but after 123=1728 iterations. The comparison between the DM-AP1 and the Hungarian methods is shown in Figure 19.

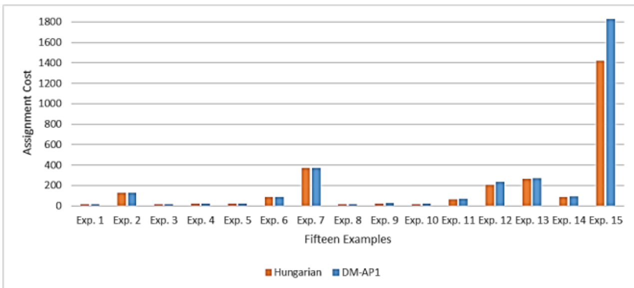

Fig 19. Comparing DM-AP1 to Hungarian method

According to the simulation results, the Hungarian method yields a better solution than the DM-AP1 in 46% of cases and for the rest (54%) the performance is similar. Nevertheless, the deviation is extremely reduced where the largest deviation is for 00.31%. Besides, DM-AP1 is the fast heuristic in the literature: it quickly generates (just after *n* iterations) a good initial basic feasible solution (in a very low computational time, see Figure 1). Thus, for small instances DM-AP1 yields a very closed deviation to the optimal with an average deviation of 00.05% and for large instances the heuristic DM-AP1 is more suitable than the Hungarian method viewing its rapidity that will empower it to be used as a good starting point for any metaheuristic.

5. Conclusion

The assignment problem plays an important role in the real-life industry where the decision maker is in the situation to find the optimal assignment of *n* objects (tasks, workers or machines) to *m* other objects.

In this study, a new heuristic entitled Dhouib-Matrix-AP1 is introduced in order to rapidly construct a good initial basic feasible solution for the assignment problem. It is composed of three simple steps and repeated only in n iterations (where n is the number of objects) using an original formula (*Sum* – (*Min* * *n* /2) to drive the construction process.

Then, several numerical examples are depicted to prove the simplicity and the performance of the proposed Dhouib-Matrix-AP1 method. For further research works, we intend to enrich the Dhouib-Matrix-AP1 with a stochastic process in an iterative structure in order to solve the Multi-objective Assignment Problem and also to prove the performance of this method to solve the assignment problem in uncertain environments: fuzzy, intuitionistic and neutrosophic.

References

[1] E. O. Kundakcioglu, and S. Alizamir, "Generalized Assignment Problem," *Encyclopedia of Optimization, Springer*. pp. 1153-1162, 2008.

[2] Ö. Karsu, M. Azizoğlu, and K. Alanl, "Exact and Heuristic Solution Approaches for the Airport Gate Assignment Problem," *Omega*, Paper ID 102422, 2021.

[3] Z. Lyu, and J.Y. Andrew, "Consultant Assignment and Routing Problem With Priority Matching," *Computers & Industrial Engineering*, vol. 151, paper ID: 106921, 2021.

[4] A. Erik, and Y. Kuvvetli, "Integration of Material Handling Devices Assignment and Facility Layout Problems," *Journal of Manufacturing Systems*, vol. 58, no. A, pp. 59-74, 2021.

[5] Ö. İ. Güneri, B. Durmuş, and D. Aydın, "Different Approaches to Solution of The Assignment Problem Using R Program," *Journal of Mathematics and Statistical Science*, vol. 5, pp. 129-145, 2019.

[6] P. Kaur, A. Sharmac, V. Vermab, and K. Dahiya, "A Priority Based Assignment Problem," *Applied Mathematical Modelling*, vol. 40, no. 17, pp. 7784-7795, 2016.

[7] S. Medjkouh, B. Xue, and G. B. Hacene, "Sparse Clustered Neural Networks for the Assignment Problem," *The Ninth International Conference on Advanced Cognitive Technologies and Applications*, pp. 69-75, 2017.

[8] D. Bokal, B. Brešar, and J. Jerebic, "A Generalization of Hungarian Method and Hall's Theorem with Applications in Wireless Sensor Networks," *Discrete Applied Mathematics*, vol. 160, no. 4, pp. 460-470, March 2012.

[9] M. Fuentes, L. Cadarso, V. Vaze, and C. Barnhart, "The Tail Assignment Problem: A Case Study at Vueling Airlines," *Transportation Research Procedia*, vol. 52, pp.445-452, 2021.

[10] G. Gao, L. Ning, H. Ting, Y. Xu, Y. Zhang, and Y. Zou, "Approximation Algorithms for the Partial Assignment Problem," *Theoretical Computer Science*, vol. 838, pp. 231-237, 2020.

[11] Q. Rabbani, A. Khan, and A. Quddoos, "Modified Hungarian Method for Unbalanced Assignment Problem with Multiple Jobs," *Applied Mathematics and Computation*, vol. 361, pp. 493-498, 2019.

[12] T. Öncan, M. Z. Şuvak, H. Akyüz, and K. İ. Altınel, "Assignment Problem with Conflicts," *Computers & Operations Research*, vol. 111, pp. 214-229, 2019.

[13] M. Miledi, and S. Dhouib, "VNS Metaheuristic Based on Thresholding Functions for Brain MRI Segmentation," *International Journal of Applied Metaheuristic Computing*, vol. 12, no. 1, pp. 94-110, 2021.

[14] S. Dhouib, S. Kouraichi, T. Loukil, and H. Chabchoub, "An Interactive Simulated Annealing multi-agents platform to solve Hierarchical Scheduling Problems with Goals," *Studies in Computational Intelligence: Springer-Verlag*, vol. 236, pp. 177-187, 2009.

[15] S. Dhouib, "A threshold accepting framework to optimize discrete and continuous search space," *International Journal of Innovative Computing and Applications*, vol. 2, no. 3, pp.178 – 187, 2010.

[16] S. Dhouib, A. Kharrat, and H. Chabchoub, "Goal Programming using Multiple Objective Hybrid Metaheuristic Algorithm," *Journal of the Operational Research Society*, vol. 62, no. 4, pp. 677-689, 2010.

[17] S. Dhouib, "A New Column-Row Method for Traveling Salesman Problem: The Dhouib-Matrix-TSP1," *International Journal of Recent Engineering Science*. vol. 8, no. 1, pp. 6-10, 2021.

[18] S. Dhouib, "Stochastic Column-Row Method for Travelling Salesman Problem: The Dhouib-Matrix-TSP2," *International Journal of Engineering Research & Technology*, vol. 10, no. 3, pp. 524-527, 2021.

[19] S. Dhouib, "Minimizing the Total Distance for the Supply Chain Problem Using Dhouib-Matrix-TSP2 Method," *International Journal of Advanced Research in Engineering and Technology*, vol. 12, no. 5, pp. 1-12, 2021.

[20] Sa. Dhouib, and S. Dhouib, "Optimizing the Trapezoidal Fuzzy Travelling Salesman Problem Through Dhouib-Matrix-TSP1 Method Based on Magnitude Technique," *International Journal of Scientific Research in Mathematical and Statistical Sciences*, vol. 8, no. 2, pp. 1-4, 2021.

[21] M. Miledi, S. Dhouib, and T. Loukil, "Dhouib-Matrix-TSP1 Method to Optimize Octagonal Fuzzy Travelling Salesman Problem Using α-Cut Technique," *International Journal of Computer and Information Technology*, vol. 10, no. 3, pp. 130-133, 2021.

[22] S. Dhouib, "Haar Dhouib-Matrix-TSP1 Method to Solve Triangular Fuzzy Travelling Salesman Problem," *Research Journal of Recent Sciences*, vol. 10, no.3, pp.1-3, 2021.

[23] S. Dhouib, "Neutrosophic Triangular Fuzzy Travelling Salesman Problem Based on Dhouib-Matrix-TSP1 Heuristic," *International Journal of Computer and Information Technology*, vol. 10, no. 5, pp 180-183, 2021.

[24] S. Dhouib, "Optimization of Travelling Salesman Problem on Single Valued Triangular Neutrosophic Number using Dhouib-Matrix-TSP1 Heuristic," *International Journal of Engineering*, vol. 34, no. 12, pp. 2642-2647, 2021.

[25] S. Dhouib, "A Novel Heuristic for the Transportation Problem: Dhouib-Matrix-TP1," *International Journal of Recent Engineering Science*, vol. 8, no. 4, pp. 1-5, 2021.

[26] S. Dhouib, "Solving the Trapezoidal Fuzzy Transportation Problems Via New Heuristic the Dhouib-Matrix-TP1," *International Journal of Operations Research and Information Systems*, vol. 12, no. 4, pp. 1-16, DOI: 10.4018/IJORIS.294119.

[27] S. Dhouib, "Solving the Single-Valued Trapezoidal Neutrosophic Transportation Problems through the Novel Dhouib-Matrix-TP1 Heuristic," *Mathematical Problems in Engineering*, Article ID 3945808, DOI: 10.1155/2021/3945808, 2021.

[28] S. Dhouib, "Novel Metaheuristic based on Iterated Constructive Stochastic Heuristic: Dhouib-Matrix-3 (DM3)," *Applied Computational Intelligence and Soft Computing*, Article ID

7761993, DOI: 10.1155/2021/7761993, 2021.

[29] S. Dhouib, "Multi-Start Constructive Heuristic Through Descriptive Statistical Metrics: The Dhouib-Matrix-4 (DM4) Metaheuristic," *International Journal of Operational Research*, DOI: 10.1504/IJOR.2021.10045069, In Press.

[30] H. A. Ahmad, "The Best Candidates Method for Solving Optimization Problems," *Journal of Computer Science*. vol. 8, no. 5, pp. 711-715, 2012.

[31] S. L. Mary, and V. Mahadevan, "A New Algorithm to Find the Optimal Feasible Assignment Problem for an Assignment Problem," *Journal of Emerging Technologies and Innovative Research*, vol. 5, no. 6, pp. 649-652, 2018.

[32] S. Rao, and M. Srinivas, "An Effective Algorithm to Solve Assignment Problems: Opportunity Cost Approach," *International Journal of Mathematics and Scientific Computing*, vol 6, no. 1, 2016.

[33] M. M. Vinchoo and R. V. Deolekar "Comparative Analysis of Different Approaches to Solve the Job Assignment Problem," *International Conference on Trends in Electronics and Informatics ICEI 2017*, pp. 129-134, 2017.

[34] H. Basirzadeh, "Ones Assignment Method for Solving Assignment Problems," *Applied Mathematical Sciences*, vol. 6, no. 47, pp. 2345-2355, 2012.

[35] S. Kabiru, B. M. Saidu, A. Z. Abdul, and U. A. Ali, "An Optimal Assignment Schedule of Staff-Subject Allocation," *Journal of Mathematical Finance*, vol. 7, pp. 805-820, 2017.

[36] M. D. H. Gamal, "A Note on Ones Assignment Method," *Applied Mathematical Sciences*, vol. 8, no. 40, pp. 1979-1986, 2014.

[37] E. Munapo, "Development of an accelerating Hungarian method for assignment problems," *Mathematics and Cybernetics - applied aspects*, vol. 4, no. 4, pp. 6-13, 2020.

[38] M. Sharma, and R. Tiwari, "Route Optimization Using Hungarian Method Combined with Dijkstra's Method in Home Health Care Services," *Research Journal of Computer and Information Technology Sciences*, vol. 5, no. 3, pp. 7-15, 2017.