Improved Discrete Differential Evolution Algorithm in Solving Quadratic Assignment Problem for best Solutions

Asaad Shakir Hameed[1], Burhanuddin Mohd Aboobaider[2], Ngo Hea Choon[3], Modhi Lafta Mutar[4]
Faculty of Information and Communication Technology
Universiti Teknikal Malaysia Melaka Hang Tuah Jaya
76100, Durian Tunggal, Melaka, Malaysia

***Abstract*—The combinatorial optimization problems are very important in the branch of optimization or in the field of operation research in mathematics. The quadratic assignment problem (QAP) is in the category of facilities location problems and is considered as one of the significant complex's combinatorial optimization problems since it has many applications in the real world. The QAP is involved in allocating *N* facilities to *N* locations with specified distances amid the locations and the flows between the facilities. The modified discrete differential evolution algorithm has been presented in this study based on the crossover called uniform like a crossover (ULX). The proposed algorithm used to enhance the QAP solutions through finding the best distribution of the *N* facilities to *N* locations with the minimized total cost. The employed criteria in this study for the evaluation of the algorithm were dependent on the accuracy of the algorithm by using the relative percent deviation (PRD). The proposed algorithm was applied to 41 different sets of the benchmark QAPLIB, while the obtained results indicated that the proposed algorithm was more efficient and accurate compared with Tabu Search, Differential Evolution, and Genetic algorithm.**

Keywords*—*Quadratic assignment problem; combinatorial optimization problems; differential evolution algorithm

I. Introduction

There are several specific problems for COPs, such as the quadratic assignment problem (QAP), routing problem (RP), etc. The QAP was introduced by [1] and the model of this problem has been applied in many aspects of life and is famous on campus and in hospital layout QAP is a complex problem that has attracted the attention of researchers since its first formulation [2], [3], [4], [5], and [6]. There are many challenges facing the installation of facilities to location, such as a lack of layout in the buildings, which leads to an increase in computational complexity [7]. The methods that found solutions to the QAP problem were classified into two categories as follows: the category that obtains the exact solution to QAP was called the exact methods, including the bounded dynamic branches and processes, Lagrangian-based relaxation methods, linear and quantitative programming methods. However, in these methods, the size of the problem requires a long calculation period if there are more than 30 methods [8], [9], [10], and [11]. The second category obtains the approximate solution or near the optimal solution with reasonable calculation time and are known as the approximate methods. The approximate methods have been divided into three categories [12]:

- Local Search Algorithm, such as Tabu Search;
- Swarm Intelligence, such as Ant Colony Optimization;
- Evolutionary Algorithm, such as the differential evolution algorithm.

The differential evolution (DE) method is one of the latest evolutionary optimization methods reported by [13]. DE is a global optimizer that relies on population and random space continuously [14]. Due to its efficiency and strength, DE has increasingly become common and has been utilized in numerous fields such as the function of continuous real value and the problems of combinatorial optimization with a discrete decision. In a study, [15] proposed an algorithm regarding the discrete differential evolution (DDE) for computation of the variation of the flow-shop preparation problem. The overall operation of this method was not as efficient as other methods, which could be due to the employment of low mutation probability (0.2). In contrast, the operation of the DDE algorithm was observed to be competitive when using the local search. An earlier work [16] modified the DE to a discrete optimization problem and was applied to solve the QAP.

However, the proposed method, which utilized the property Tabu List, was not able to use the crossover. Hence, the mutant vector directly became a trial vector and can solve nearly all the instances from Nug15 to Nug30 in QAPLIB. Nevertheless, the obtained results were not superior to the max-min ant system hybrid with the random selection and the local search. In another study [17], modified DDE with the local search-based modification using insertion and swap was used. Employment of DDE with local search further improved the results of two types of sparse and dense examples of QAPLIB.

The study aimed to modify the DDE for management of the complex problems while being capable of the exact search space with minimum cost. Moreover, the execution of the proposed algorithm led to enhancement of solving instances of QAP from the benchmark QAPLIB. The remaining of the study has been organized as follows. Section II provides a description of QAP, while section III presents the materials and methods. The computational results have been discussed in section IV, while the conclusion has been provided in section V.

II. Quadratic Assignment Problem (QAP)

QAP is considered as one of the site problems which reduce the momentum within the places of high mobility such as hospitals, campuses, and several facilities to be allocated to these sites by calculating the matrix of distances between location and the flows between facilities. Solving the QAP indicates examining the assignment that reduces the cost of transportation among the facilities. In order to have a QAP instance, visibility and a list of distances of accessible locations and material flow among facilities *(F_{ij})* must be available. Each *N* facility is interchangeable, and there are *N* locations that can only provide for one facility. Moreover, there are *N* facilities set and *N* locations set and for each location pair the specification of distance *(D_{ij})* and for each facility pair, a flow (Fij) is itemized. The difficulty in assigning entire facilities to alternative positions is aimed at minimizing the sum of distances increased by conforming flows [5]. Formally, let and be two *N*N* matrices and let *P* be the set of permutation of {1, 2, …, *n*}. Then, the mathematical model of QAP can be written as

$$Min\ f(\pi) = \sum_{i=1}^{n}\sum_{j=1}^{n} F_{ij}\, D_{\pi(i)\pi(j)} \qquad (1)$$

Overall permutations $\pi \in P_n$

A. Mathematical Model Assumptions

- *N* is the dimension of the problem case
- The objective function is Mini Sum
- π signifies a potential permutation over (1, 2,...,n) and π(*i*) relates to the index of the location to which facility i is allocated
- π is an ideal way of representing a solution to a QAP problem
- Each facility is allocated to precisely one location and vice-versa
- The solution space is discrete and finite
- The number of location and facilities are known
- All decision variables of the model are binary (0–1) variables

B. Mathematical Model Outputs (Decision Variables)

$$X_{ij}=\begin{cases}1 & if\ facility\ i\ assigned\ in\ location\ j\\ 0 & Otherwise\end{cases} \qquad (2)$$

III. Materials and Methods

A. Materials

Since the first formulation of the QAP model, numerous researchers have performed studies in this are to generate algorithms with a capacity of locating practical solutions. Several algorithms were created along with numerous problem instances. Several researchers from the Graz University of Technology created the QAPLIB (http://anjos.mgi. Polymtl. ca/qaplib/) in order to deliver these data and explanations to the scientific community. Then and there, the QAPLIB was an up-to-date source which possessed all the available QAP instances. Majority of the available algorithms for the purpose of solving the QAP were examined on these benchmark instances. In excess of over 100 instances were obtained either from real life applications or randomly produced problem instances. In this study, five categories of instances were solved by the QAPLIB with problem sizes fluctuating from 12 to 80 locations as follows

- Randomly generated instances such as (Tai25a, Tai30a, Tai40a, Tai50a, Tai60b, Tai64c, Lipa70a, Lipa80a).
- Real-life instances such as (Chr12c, Chr15a, Bur26a, Kra30a, Kra30b, Ste36a).
- With grid-based distance matrix such as (Nug12, Nug14, Nug15, Nug16a, Nug16b, Nug17, Nug18, Nug20, Nug25, Sko49).
- The entries in flow matrices of the rectangular distanced problems are pseudorandom numbers (Sko49, Wil50).
- Manhattan distances of rectangular grids such as (Had12, Had14, Had20).

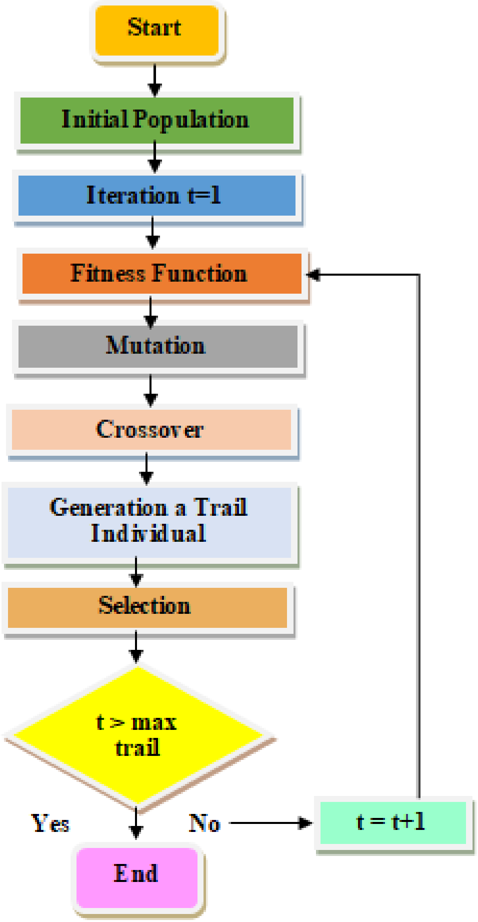

Fig. 1. Flowchart of DDE Algorithm.

B. *Methods*

Various metaheuristics have been used and developed for finding the best solution of QAP and the likelihood of receiving a satisfying solution value within an acceptable time span. In this study, the modified of discrete differential evolution (DDE) has been proposed, which includes the type of the crossover (ULX) [18] to get the diversity of the search space. Moreover, the algorithm can solve the permutation of the QAP. The DDE algorithm is simple in nature and by mutating the target population it produces the mutant population. Then, a crossover operator is required to incorporate the mutated solution with the target solution in order to produce a trail solution. Finally, the selection was based on the survival of the fittest among the trial and target solutions. Fig. 1. shows the main steps of the proposed approach to solve QAP.

1) Initial population: The DDE algorithm begins with initializing of the primary target population $\pi_i = [\pi_1, \pi_2, .., \pi_{NP}]$ with the dimension of NP. Every individual contains an n-dimensional vector with parameter values randomly and equally established among pre-defined search range. The initial population of DDE algorithm is shown in Table 1.

TABLE I. INITIAL POPULATION

π_1	6	3	1	4	2	5
π_2	2	1	6	5	4	3
. . .	. . .	. . .	. . .	. . .	. . .	. . .
π_n	4	6	2	3	5	1

a) A permutation was directly encoded as an individual vector;

b) The dimension of the individual was equivalent to the size of the QAP problem;

c) In the initialization stage, the population was generated as a random permutation.

2) Mutation individual: The differential deviation was attained in the form of perturbations of the optimum solution from the former generation in the objective population. Since perturbations are stochastically achieved, every individual in the mutant population was anticipated to be characteristic. In order to attain the mutant individual, the subsequent equation could be employed

$$v_i^t = \begin{cases} insert(\pi_b^{t-1}) & if(r < P_m) \\ swap(\pi_b^{t-1}) & otherwise \end{cases} \tag{3}$$

where π_b^{t-1} is the optimum solution from the former generation in the objective population; P_m is the perturbation probability; the insert ( ) and swap ( ) are merely the solo additions and swap shifts. A constant arbitrary number r was produced among [0,1]. If r was lower than P_m, then a sole insertion shift was employed in order to produce the mutant individual v_i^t; else, a sole swap shift was employed.

3) Crossover

The crossover operator with the crossover was proposed by [18] and is known as the uniform like crossover (ULX). The crossover was obtained as the follows First, all items were allocated to a similar location and both parents were copied to this position in the child.

a) Second, the unassigned positions of a permutation were scanned from left to right: for the unassigned position, an item was selected randomly, consistently from those in the parents if they were not yet incorporated in the child.

b) Third, remaining items were randomly allocated.

4) Generation a trail individual: After the perturbation phase, the trial individual was attained as follows:

$$u_i^t = \begin{cases} CR & if(r < P_c) \\ v_i^t & otherwise \end{cases} \tag{4}$$

where CR is the crossover operator; and P_c is the crossover probability. When a uniform random number r was less than the Pc the crossover operator was utilized to produce the trial individual u_i^t . Otherwise, the trial individual was selected as $u_i^t = v_i^t$. Hence, the trial individual was made up of the outcome of the perturbation operators or from the crossover operator.

5) Selection: The selection was based on fitness function and the following equation can be used

$$\pi_i^t = \begin{cases} u_i^t & if\left(f(u_i^t) \leq f(\pi_i^{t-1})\right) \\ \pi_i^{t-1} & otherwise \end{cases} \tag{5}$$

The selection was grounded on the existence of the rightest amongst the trial and target individuals.

6) Verification of the stopping criterion: The stopping criterion is dependent on the finish of the specified number of repetitions. The algorithm could be stopped if the solution was not improved.

IV. COMPUTATIONAL RESULTS

The algorithm which was proposed was encoded in MATLAB on a PC with Intel(R) Core (TM) i7-3770 CPU @ 3.40 GHz and 4.00 GB RAM under MS Windows 10. This section has been presented two stages, the first stage included the parameters which used by the proposed algorithm. Then, the second stage has been presented the discussion of results which obtained by using the proposed algorithm.

TABLE II. PARAMETER SETTING OF DDE ALGORITHM FOR QAP

Parameters	**Value**
Population Size	100
Number of Particles	30
Maximal iterative Number	1000
Probability of Mutation Pm	[0,1]
Probability of Crossover Pc	[0,1]

A. *Parameter setting*

The parameters to be determined in the DDE has been shown in table 2 as follows:

B. *Results and Discussions*

The operations of the DDE are tabulated in Table 3,4, and 5. The RPD field denotes the Relative Percent Deviation between the best-found solution S by proposed algorithm and the optimal (or the Best-Known Solution BKS) as a formula:

$$RPD = \frac{(S - S_{optimal(or\ BKS)})}{S_{optimal(or\ BKS)}} * 100\% \qquad (6)$$

The precision of the algorithm was estimated by utilizing the rate of RPD. Smaller values of the average RPD was more robust for the evaluated algorithm. In the cases, (Nug12, Nug14, Nug15, Nug16a, Nug16b, Nug17, Nug18, Nug20, Nug25) the proposed DDE selects the optimal solution with gap 0%. Then, the performance of DDE finds the optimal solution in an instance (Bur26a, Bur26b, Bur26c, Bur26d, Bur26e, Bur26f, Bur26g, Bur26h) with gap 0%. On the other hand, the performance of DDE has been applied in 12 instances in this work (Tai25a, Tai30a, Tai40a, Tai50a, Tai60b, Sko49, Wil50, Lipa70a, Lipa80a, Chr15a, Esc128, Kra30b).

The attained results of these instances were suitable for finding the optimum solution and excellent accuracy. Finally, the different instances of QAP (Tai64c, Lipa40b, Chr12c, Esc16i, Had12, Had14, Had20) were solved by DDE. The results of the DDE were compared with the Tabu Search Algorithm (TS) which belongs to a local search. Other algorithms were compared with DDE such as Genetic Algorithm (GA), and Differential Evolution Algorithm (DE), which belong to evolutionary methods. The Tabu Search which belongs to local search algorithms, was applied in [19] to solve some of instances of QAP from benchmark (QAPLIB) such as (Nug12, Nug14, Nug15, Nug16a, Nug16b, Nug17, Nug18, Nug20, Nug25, Bur26a, Bur26b, Bur26c, Bur26d, Bur26e, Bur26f, Bur26g, Bur26h, Tai25a, Tai30a, Tai40a, Tai50a), while optimum results were obtained in the cases (Nug14, Nug17) with gap 0%. Next, in the size of problem less than 30 (Nug12, Nug15, Nug16a, Nug16b, Nug18, Nug20, Nug25, Bur26a, Bur26b, Bur26c, Bur26d, Bur26e, Bur26f, Bur26g, Bur26h, Tai25a) the TS was unable to access the optimal solution, and the gap of the results between these cases and the results in QAPLIB was between 0% to 5%.

On the other hand, the performance of TS was inferior to the cases (Tai30a, Tai40a, Tai50a) and the gap between these cases were between 4% to 8%. Moreover, the average of relative percent deviation between the solution by TS and the optimal solution or best-known solution in benchmark QAPLIB was 1.41%. Based on the dataset above, the performance of the proposed algorithm was better than the algorithm TS and it located the best solution for 17 out of 21 cases (Nug12, Nug14, Nug15, Nug16a, Nug16b, Nug17, Nug18, Nug20, Nug25, Bur26a, Bur26b, Bur26c, Bur26d, Bur26e, Bur26f, Bur26g, Bur26h) at the gap 0%. The obtained results were better than the algorithm TS in the cases (Tai25a, Tai30a, Tai40a, Tai50a). Finally, the average of relative percent deviation between the solution by DDE and the optimal solution or best-known solution in benchmark QAPLIB was 0.69%. Table 3 illustrates the comparison between the TS and DDE as follows

Fig. 2 displays the relative percentage deviation (relative difference) of the solution quality for various problem sizes for TS and DDE algorithms. The obtained results indicated that the DDE algorithm possessed a suitable solution quality, which was higher than the TS algorithm for solving QAP instances.

TABLE III. SUMMARY OF COMPARISON OF DDE VERSUS TS

Instances	**QAPLIB**		**RPD of TS [19]**	**RPD of DDE**
	Optimal	**BKS**		
Nug12	578		0.39	0
Nug14	1014		0	0
Nug15	1150		0.39	0
Nug16a	1610		0.87	0
Nug16b	1240		1.37	0
Nug17	1732		0	0
Nug18	1930		0.69	0
Nug20	2570		1.04	0
Nug25	3744		1.55	0
Bur26a	5426670		0.09	0
Bur26b	3817852		0.19	0
Bur26c	5426795		0.26	0
Bur26d	3821225		0.02	0
Bur26e	5386879		0.03	0
Bur26f	3782044		0.05	0
Bur26g	10117172		0.01	0
Bur26h	7098658		0.01	0
Tai25a	1167256		4.26	2.05
Tai30a		1818146	4.75	3.55
Tai40a		3139370	6.12	3.88
Tai50a		4938796	6.94	5.04
Average RPDs			**1.41**	**0.69**

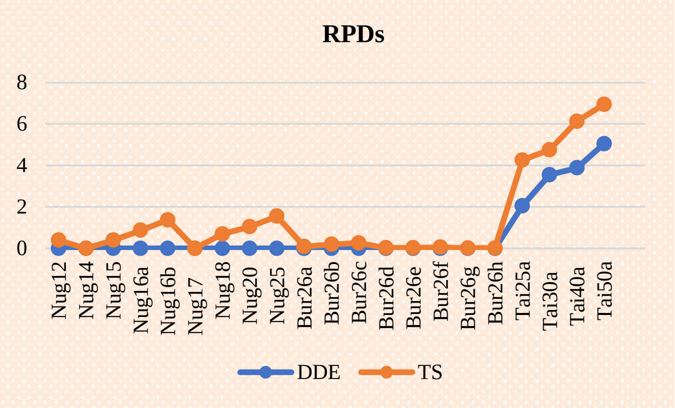

Fig. 2. Comparison of RPDs for DDE Versus TS.

TABLE IV. SUMMARY OF COMPARISON OF DDE VERSUS DE

Instances	QAPLIB		RPD of DE [17]	RPD of DDE
	Optimal	BKS		
Esc32e	2		0	0
Esc32g	6		0	0
Esc32h	438		15.07	0
Esc64a	116		24.14	0
Tai64c		1855928	0.19	0
Lipa40b	476581		24.73	0
Sko49		23558	12.5	3.45
Wil50		49482	7.58	1.36
Tai60b	630211362	630211362	26.45	3.61
Lipa70a	171605		1.76	1.08
Lipa80a	253195		1.59	1.04
Average RPDs			**10.36**	**0.95**

In a study by [17], DE was applied to solve 11 sets of benchmark instances as follows: Esc32e, Esc32g, Esc32h, Esc64a, Tai64c, Lipa40b, Sko49, Wil50, Tai60b, Lipa70a, and Lipa80a. The obtained results in the cases of Esc32e and Esc32g were an optimal solution with gap of 0%. Then, in the cases Esc32h, Esc64a, Lipa40b, Tai60b, Sko49, and Wil50, the results were inferior to the results in benchmark with the gap between 7.58 to 26.45%. Finally, the performance of DE was satisfactory in the cases of Tai64c, Lipa70a, and Lipa80a, while the gap of these results was between 0.19 to 1.76%.

The obtained results showed that the average of relative percent deviation between the solution by DE and the optimal solution or best-known solution in the benchmark was 10.36 %. By means of applying the proposed algorithm to the same cases that were solved by the algorithm DE, the execution of the proposed algorithm was superior to the DE algorithm, where the optimal solution was found for 6 cases out of 11 cases (Esc32e, Esc32g, Esc32h, Esc64a, Tai64c, Lipa40b) and the gap was 0%. On the other hand, the results for the five remaining cases (Sko49, Wil50, Tai60b, Lipa70a, Lipa80a) were noble and were superior to those obtained by DE, with the gap between 1.04 to 3.45%. The obtained results showed that the average of the relative percent deviation between the solution by DDE and the best solution in the benchmark was 0.95 %. Table 5 presents the comparison between the DE and DDE.

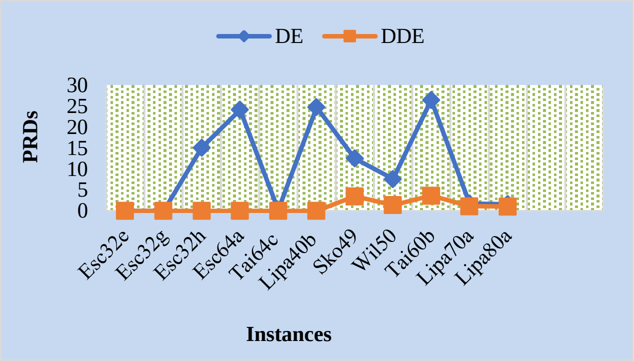

Fig. 3. Comparison of RPDs for DE Versus DDE.

TABLE V. SUMMARY OF COMPARISON OF DDE VERSUS GA

Instances	QAPLIB		RPD of GA [20]	RPD of DDE
	Optimal	BKS		
Bur26h	7098658		0.39	0
Chr12c	11156		0	0
Chr15a	9896		0.39	0
Esc128	64		0.87	0
Esc16i	14		1.37	0
Esc32h	438		0	0
Esc64a	116		0.69	0
Had12	1652		1.04	0
Had14	2724		1.55	0
Had20	6922		0.09	0
Kra30b	91420		0.19	0
Average RPDs			**13.14**	**4.38**

Fig. 3 displays the relative percentage deviation (relative difference) of the solution quality for various problems sizes for DE and DDE algorithms. The obtained results indicated that the DDE algorithm possessed a suitable solution quality, which was higher than the DE algorithm for solving QAP instances.

In a study by [20], GA was applied to solve 11 sets of benchmark instances, as follows: (Bur26h, Chr12c, Chr15a, Esc128, Esc16i, Esc32h, Esc64a, Had12, Had14, Had20, Kra30b). The obtained results showed for the three cases (Chr12c, Esc32h, Kra30b) the gap was between 6.09 to 9%, while the average of relative percent deviation between the solution by GA and the optimal solution in benchmark QAPLIB was 13.14%. The results of the proposed algorithm in this study are shown in Table 6. It was observed that DDE (4.38% grand average RPD) outperformed the GA (13.14% grand average RPD) on all 11 instances. The proposed algorithm was effective in selecting the optimal solution for eight out of 11 cases (Bur26h, Chr12c, Esc128, Esc16i, Esc32h, Esc64a, Had12, Had14, Had20).

Fig. 4 displays the relative percentage deviation (relative difference) of the solution quality for various problems sizes for GA and DDE algorithms. The obtained results indicated that the DDE algorithm possessed a suitable solution quality, which was higher than the GA algorithm for solving QAP instances.

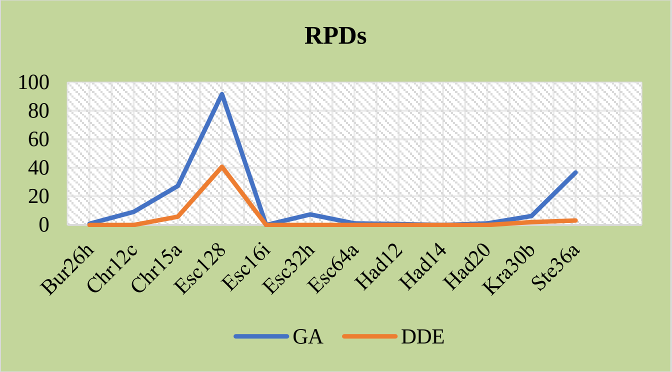

Fig. 4. Comparison of RPDs for DDE Versus GA.

The limitations of this study involve the absence of assurance in obtaining the optimum solutions in a predetermined amount of time. Nevertheless, asymptotic convergence proofs are accessible for problems, which are computationally expensive and require a massive amount of computational resources.

V. Conclusion

In this study, a modified discrete differential evolution (DDE) algorithm has been proposed for obtaining an operational solution to the QAP. Through beginning with an initial population of DDE, the single insertions and swap shift were applied to generate the mutation individually. However, uniform-like crossover (ULX) was employed as a crossover operator in this algorithm in order to obtain diversity in the search space and find the best solution of QAP. The obtained results of five classes of benchmark QAPLIB instances indicated the efficiency of the proposed algorithm. From the 41 instances which were investigated, 31 instances were solved optimally. Then a comparative study between DDE and three algorithms (TS, DE, and GA) were presented for similar instances.

The found results indicated that DDE outperformed TS, DE, and GA on all category instances. Moreover, DDE was found to be better than other algorithms in terms of the solution quality. Further studies regarding the utilization of DDE along with other algorithms could be carried out in order to provide better results.

Acknowledgment

This research has been supported by Universiti Teknikal Malaysia Melaka (UTeM) under UTeM Zamalah Scheme.

References

[1] T. C. Koopmans and M. J. Beckmann, 1957. “Assignment Problems and the Location of Economic Activities Author (s): Tjalling C. Koopmans and Martin Beckmann,” Econometrica, vol. 25, no. 1. pp.53–76.

[2] S. Sahni and T. Gonzalez, 1976. “P-Complete Approximation Problems,” J. ACM, vol. 23, no. 3, pp. 555–565.

[3] E. Duman, M. Uysal, and A. F. Alkaya, 2012. “Migrating Birds Optimization: A new metaheuristic approach and its performance on quadratic assignment problem,” Inf. Sci. (Ny)., vol. 217, pp. 65–77.

[4] U. Benlic and J. K. Hao, 2013. “Breakout local search for the quadratic assignment problem,” Appl. Math. Comput., vol. 219, no. 9, pp. 4800–4815.

[5] B. R. M. M. Y. Mohamad Amin Kaviani, Mehdi Abbasi, 2014. “A hybrid Tabu search-simulated annealing method to solve quadratic assignment problem,” Decis. Sci. Lett., vol. 3, no. 3, pp. 391–396.

[6] Shawky, L.A.E.F., Metwally, M.A.E.B. and Zaied, A.E.N.H., 2015. Quadratic Assignment Problem: A survey and Applications. International Journal of Digital Content Technology and its Applications, 9(2), p.90.

[7] R. K. Ahuja, K. C. Jha, J. B. Orlin and D. Sharma, 2002. “Very large-scale neighborhood search for the quadratic assignment problem”, Working Paper, MIT Sloan School of Management.

[8] Lim, W. L., Wibowo, A., Desa, M. I., and Haron, H. 2016. A biogeography-based optimization algorithm hybridized with tabu search for the quadratic assignment problem. Computational intelligence and neuroscience, 2016, 27.

[9] M. Abdel-Baset, H. Wu, Y. Zhou, and L. Abdel-Fatah, “Elite opposition-flower pollination algorithm for quadratic assignment problem,” J. Intell. Fuzzy Syst., vol. 33, no. 2. 2017, pp. 901–911.

[10] Shylo, P.V., 2017. Solving the quadratic assignment problem by the repeated iterated tabu search method. Cybernetics and Systems Analysis, 53(2), pp.308-311.

[11] Pradeepmon, T., Sridharan, R. and Panicker, V., 2018. Development of modified discrete particle swarm optimization algorithm for quadratic assignment problems. International Journal of Industrial Engineering Computations, 9(4), pp.491-508.

[12] Hameed, Asaad Shakir, Aboobaider, Burhanuddin Mohd, Choon, Ngo Hea, and Mutar, Modhi Lafta, 2018. Review on the Methods to Solve Combinatorial Optimization Problems Particularly: Quadratic Assignment Model. International Journal of Engineering & Technology, 7(3.20), pp.15-20.

[13] R. Storn, K. Price, 1997. “Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces”, Journal of Global Optimization, vol. 11, pp. 341- 359.

[14] Price, K.V., Storn, R. and Lampinen, J. 2005. “Differential Evolution: A Practical Approach to Global Optimization”. Springer-Verlag, London, UK.

[15] Pan, Quan-Ke, Mehmet Fatih Tasgetiren, and Yun-Chia Liang., 2008. "A discrete differential evolution algorithm for the permutation flow shop scheduling problem." Computers & Industrial Engineering 55.4, pp. 795-816.

[16] Kushida, Jun-ichi, et al., 2012. "Solving quadratic assignment problems by differential evolution." Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), Joint 6th International Conference on IEEE.

[17] M. Fatih Tasgetiren, Quan-Ke Pan, P. N. Suganthan, and Ikbal Ece Dizbay, 2013. “Metaheuristic Algorithms for the Quadratic Assignment Problem”. IEEE Symposium on Computational Intelligence in Production and Logistics Systems (CIPLS). pp. 131-137.

[18] D.M. Tate, A.E. Smith, 1995. A genetic approach to the quadratic assignment problem. Computers & Operations Research, Vol.1, pp. 73–83.

[19] Mohamad Amin Kaviania, Mehdi Abbasib, Bentolhoda Rahpeymab and Mohamad Mehdi Yusefib, 2014. A hybrid Tabu search-simulated annealing method to solve quadratic assignment problem. Decision Science Letters 3, pp. 391–396.

[20] Gamal Abd El-Nasser A. Said, Abeer M. Mahmoud, and El-Sayed M. El-Horbaty,2014. “A Comparative Study of Meta-Heuristic Algorithms for Solving Quadratic Assignment Problem”. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 5, No. 1