Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2023, Article ID 8892689, 16 pages
https://doi.org/10.1155/2023/8892689

Research Article

Appling the Roulette Wheel Selection Approach to Address the Issues of Premature Convergence and Stagnation in the Discrete Differential Evolution Algorithm

Asaad Shakir Hameed,[1,2] **Haiffa Muhsan B. Alrikabi**,[3] **Abeer A. Abdul–Razaq**,[4] **Zakir Hussain Ahmed**,[5] **Huda Karem Nasser**,[2] **and Modhi Lafta Mutar**[2,6]

[1]*Quality Assurance and Academic Performance Unit, Mazaya University College, Thi-Qar, Iraq*
[2]*Department of Mathematics, General Directorate of Thi-Qar Education, Ministry of Education, Thi-Qar, Iraq*
[3]*Department of Mathematics, College of Education for Pure Sciences, Thi-Qar University, Thi-Qar, Iraq*
[4]*Department of Mathematics, College of Computer Science and Mathematics, Thi-Qar University, Thi-Qar, Iraq*
[5]*Department of Mathematics and Statistics, College of Science, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia*
[6]*Computer Engineering Technology Department, Technical Engineering College, Al-Ayen University, Thi-Qar, Iraq*

Correspondence should be addressed to Asaad Shakir Hameed; asaadutem@yahoo.com

Received 13 December 2022; Revised 3 April 2023; Accepted 5 April 2023; Published 2 May 2023

Academic Editor: Jun He

Copyright © 2023 Asaad Shakir Hameed et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The discrete differential evolution (DDE) algorithm is an evolutionary algorithm (EA) that has effectively solved challenging optimization problems. However, like many other EAs, it still faces problems such as premature convergence and stagnation during the iterative process. To address these concerns in the DDE algorithm, this work aims to achieve the following objectives: (i) investigate the causes of premature convergence and stagnation in the DDE algorithm; (ii) propose techniques to prevent premature convergence and stagnation in DDE, including a quantitative measurement of premature convergence based on the level of mismatching between the population solutions and then divide the population into individual groups based on the level of mismatching between the population solutions and the best solution; and applying the roulette wheel selection (RWS) approach to determine whether a higher degree of nonmatching is more suitable for choosing a population of separate groups to be able to produce a new solution with more options to prevent the occurrence of premature convergence; (iii) evaluate the effectiveness of the proposed techniques through employing the DDE algorithm to solve the quadratic assignment problem (QAP) as a standard to evaluate our results and their effect on avoiding premature convergence and stagnation issues, which led to the enhancement of the algorithm's accuracy. Our comparative study based on the statistical analysis shows that the DDE algorithm that uses the proposed techniques is more efficient than the traditional DDE algorithm and the state-of-the-art methods.

1. Introduction

Nowadays, most studies have focused on finding optimization techniques that can obtain an optimum (or near optimum) solution to complex combinatorial optimization problems (COPs) within a moderate computational effort. In general, there are two groups of real-world problems: optimization problems and decision problems. The decision problems are the problems that can be solved by answering yes or no. While the optimization problems are the problems whose solutions involve determining the optimum (maximum or minimum) solution of the problem [1]. Furthermore, the optimization problems have been classified into two categories: discrete problems that have discrete variables and continuous problems that have continuous variables. In the same context, the COPs belong to discrete optimization problems.

One of the most important types of COPs is the quadratic assignment problem (QAP) which involves discrete variables and a set of feasible solutions [1]. It was initially defined as a mathematical model related to economic activities in [2] that aim to identify the best way to allocate locations for facilities so that every facility is mapped to only one location while every location is mapped to only one facility so as to minimize the total distance multiplied by the corresponding flows.

The problem can be stated as given n facilities and n locations (n is the problem size); a distance matrix (D) consisting of distances between every pair of locations; a flow matrix (F) consisting of traffic flows between every pair of facilities. A solution (π) is a permutation or one-to-one mapping of facilities to locations, and the objective function is defined by the sum of distances between all assigned pairs of facilities multiplied by the corresponding flows. In other words, the QAP can be stated as follows:

$$\mathrm{Min} f_{(\pi)} = \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} d_{\pi(i)\pi(j)}. \quad (1)$$

Exact and approximate algorithms are currently the most popular methods for solving the QAP. Finding an optimal solution of the large size of the QAP using exact algorithms is a great challenge. Generally, these methods require large computational time, and hence, they are used to solve only very small-sized problem instances [3]. Most large-sized problem instances remain nearly intractable using exact algorithms [4]. This reason has motivated the researchers to use approximate methods to find better solutions to the QAP instances within a moderate computational effort. On the other hand, there are two types of approximate algorithms: heuristic and metaheuristic algorithms. Generally, the study [5] gave the definitions of those algorithms as follows: a heuristic algorithm is an approach to problem-solving that uses a practical method that is not guaranteed to be optimal. In this situation, the heuristics are treated as ways offered to search and obtain better solutions, while metaheuristics are a set of intelligent strategies to enhance the efficiency of heuristic procedures.

Evolutionary algorithms (EAs) are one of the best metaheuristic algorithms and have the advantage of global exploration due to the diversity of the population. Despite those advantages of the EAs possessed, however, they still suffer from the problems of premature convergence and stagnation. Amongst EAs, the differential evolution (DE) algorithm is considered the recent algorithm in EAs, and it is very competitive approach for solving optimization problems [6]. Additionally, unlike the exact algorithms, DE algorithms are found to be very efficient in finding solutions that are optimal (or near optimal) within a moderate effort for large-sized problem instances. As these algorithms were initially developed for solving continuous optimization problems, so as to deal with the QAP, some studies have suggested improving the DE algorithm to the discrete DE (DDE) algorithm [7, 8].

The main goal of this study is to answer the following question: how to handle the premature convergence and stagnation issues in the discrete differential evolution algorithm (DDE)? In order to achieve that, the following specific objectives are designed to guide the study as well: (i) to utilize a quantitative measurement of premature convergence based on the degree of nonmatching between the population solutions, then the population is divided into individual groups based on the degree of nonmatching between the population solutions and the best solution; (ii) to apply the roulette wheel selection (RWS) method to consider whether a greater nonmatching degree is of higher fitness to select a population of individual groups to be able to generate a new solution with more opportunities to avoid the occurrence of the premature convergence; and (iii) to utilize another technique when those proposed techniques in (i) and (ii) cannot succeed in preventing the stagnation for a set of solutions through regenerating those solutions after waiting for the defined number of iterations defined in the experiment setting. This research work has employed the DDE algorithm to solve the quadratic assignment problem (QAP) as a standard to evaluate our results and their effect on avoiding premature convergence and stagnation issues, which led to the enhancement of the accuracy of the algorithm. Our comparative study based on the statistical analysis shows that the DDE algorithm that uses the proposed techniques is more efficient than the traditional DDE algorithm and the state-of-the-art methods.

The layout of this paper is presented as follows: Section 2 presents the research problem; related works are introduced in Section 3; the materials and methods are presented in Section 4; while the computational results and discussion are reported in Section 5, whilst Section 6 reports the statistical analysis, whereas theoretical analysis and critical explanation of the performance of enhancement proposed are introduced in Section 7. Finally, Section 8 presents the conclusion.

2. Research Problem

There are two types of metaheuristic algorithms, one of them called single-solution based metaheuristics, while the second type is called population-solution metaheuristics. Premature convergence is a situation where the convergence of the population to local optimum causes the algorithm to lose the diversity of the population [9]. The category of the EAs is among the most important metaheuristic algorithms under the population-solution metaheuristics category, and they have a global search capacity and are very suitable for solving complex optimization problems [10]. However, they still suffer from the premature convergence issue, where all individuals have the same learning direction during the iterative process [11], which is the main cause of the stagnation issue, thus decreasing the algorithm efficiency.

The problem statement is as follows: the set of existing solutions for this issue uses the objective function of the problem as the base to design the fitness function used in selection operations such as the RWS. Therefore, as the fitness does not directly express the diversity of the population, there is a need to design a method that directly expresses like a nonmatching degree to measure the diversity of solution. This way the probability of the occurrence of premature convergence is reduced through increased diversity of the population.

3. Related Works

Regarding the DE algorithm for the COPs, nowadays, most of the researchers are interested in either developing a better algorithm or improving an existing algorithm. Furthermore, it is found that many literatures have tried to change the algorithms for continuous space to discrete space as later contains a large number of problems such as scheduling problem. In [12], a study for solving the discrete space problem is presented by adding a truncation method for rounding the rational values of a DE population such that the parameters of objective function would become discrete.

The advantages of the DE algorithm attracted researchers to improve it to deal with discrete optimization problems, such as the QAP. In [13], an approach associated with DDE is proposed for calculating variances in the flow-shop preparation problem. As reported, the algorithm is not found to be effective because of using low mutation probability (0.20). But, the DDE algorithm operation is found to be more useful and effective if a local search method is incorporated. Initially, the DE approach was adapted to solve COPs and then applied to solve the QAP [7]. In [8], swap and insertion mutations are used along with the local search method for improving the DDE algorithm. The modified DDE algorithm using local search showed better results for two types of sparse and dense QAPLIB instances (https://coral.ise.lehigh.edu/data-sets/qaplib/).

Premature convergence and stagnation issues have been distinguished in the literature by the study [14], and the phrase "stagnation issue" describes a circumstance in which the optimum-seeking process stalls before locating a globally optimal solution. Stasis typically happens almost for no apparent cause. The population remains diversified and unconverted after stagnation, unlike premature convergence, but the optimization process no longer advances.

Investigation in the DE algorithms on the premature convergence and stagnation issues has been carried out in [9] as follows: when the population is converged to the local optimum, that means, the algorithm has lost the diversity of the population, then the convergence is premature convergence. On the other hand, when the algorithm becomes incapable of generating better new solutions (new offspring) by the evolutionary process; that is, it loses the capability to improve the solutions, then it is a stagnation problem. In this context, crossover operator plays an efficient role in the algorithm to achieve the balance between the convergence speed and population diversity [15]. Recently, an efficient crossover operator named uniform like crossover (ULX) applied by the study [16] for the DDE algorithm to solve the QAP.

Most DE algorithms developed for the scheduling problems require a transformation method for encoding the solutions as vectors, which are decoded only at the evaluation time. These kinds of schemes were used in several DE algorithms in various studies [17, 18]. Another kind of the DE is known as the DDE, in which the evolutionary operators are developed depending on discrete permutation representation. Furthermore, it can skip the transformation operation between integer and real solution representations. However, the research on the DDE algorithm is too little, and they are generally used for solving the flow-shop scheduling problems [19, 20].

There are many studies that have focused on the issue of premature convergence and how to prevent its occurrence, especially in evolutionary algorithms such as the genetic algorithm, where population diversity helps prevent this issue. The studies [21, 22] have summarized some methods that dealt with this issue as follows ("restricted selection, dynamic application of mutation, constraints for crossover and mutation probabilities, stochastic universal sampling, variable fitness assignment, population partial reinitialization, individuals grouping methods, restricted mating, zymogenesis, species conserving techniques, ranking sort based on Pareto dominance, local search based on diversity, elitist technique, and dynamic genetic clustering algorithm").

On the other hand, most studies stated that the selection of parents was based on the value of the objective function, as they would be selected if they had the best value of the objective function after applying some restriction on the selection process. A recent study [23] reported the selection operators used in the EAs as follows: RWS, elitism selection (ES), tournament selection (TOS), stochastic universal sampling (SUS), linear rank selection (LRS), exponential rank selection (ERS), and the truncation selection (TRS).

The researchers in the study [24] discussed a stagnation issue in the DE algorithm and how parents choose in the DE algorithm based on previous studies included in their study, where summarized the method proposed by them according to the fitness values of the solutions in the existing population, solutions with higher fitness values have a greater chance of being chosen as parents. The choice of parents is made depending on how far off the present population's solutions are from one another. It is more likely that the solutions with short distances will be chosen as parents. Thus, the parents that been chosen from the most recently updated solutions will maximize the likelihood of producing successful solutions.

The review [25] has proposed tracking mechanism (TM) and backtracking mechanism (BTM) when population of the DE algorithm tends to stagnate or undergo premature convergence, which prevents it from achieving the global optimum. In order to address those issues used the TM to encourage population convergence when the population entered a state of stagnation and used BTM to restore the population's diversity when it entered a state of premature convergence. The study [26] has addressed premature convergence and stagnation issues in the DE algorithm as follows: the mutation operator significantly impacts differential evolution's (DE) effectiveness. Misconfigured mutation techniques and control parameters might result in premature convergence owing to overexploitation or stagnation due to overexploration. An efficient DE algorithm must strike a balance between exploration and exploitation. The enhanced DE (EDE) for truss design presented in this work makes use of two novel strategies—integrated mutation and adaptive mutation factor strategies—to achieve a fair balance between the exploration and exploitation of DE.

On the other hand, computing now includes quantum computing (QC), and the quantum-inspired DE (QDE) fully uses the QC's rapidity and the DE's optimization capabilities. Moreover, various research studies suggested an enhanced QDE with multiple methods (MSIQDE) in the literature. However, it still has poor search accuracy and premature convergence. Hence, the study [27] has resolved these issues with the MSIQDE through a new differential mutation, and a technique of a difference vector is suggested to improve the searchability and descent ability. A new multipopulation mutation evolution method is created to assure the relative independence of each subpopulation and the population variety. The quantum chromosome is mapped from a unit space to a solution space using the viable solution space transformation approach to arrive at the best outcome.

Recently, the study [28] has provided an effective DE version, OLELS-DE, by designing orthogonal learning and elites local search algorithms, effectively addressing DE's stagnation and premature convergence issues. A population diversity estimation technique is used to empirically differentiate between these two circumstances once the stagnation or premature convergence phenomena has been found by keeping track of the best individual's update condition during the evolution.

Another recent study [29] has covered the topics of premature convergence and stagnation in the DE algorithm, which are still thought to be unresolved problems by researchers. The literature [29] aims to (1) provide a new insight into function landscape analysis with domain transform (DT); (2) alleviate the problems of premature convergence and stagnation, which frequently occur on complicated multimodal function landscape; and (3) construct a new searching paradigm based on DT. The DT from signal processing and communication fields to evolutionary computation is introduced. The domain transform-based evolutionary optimization (DTEO) technique will be presented in this part before being used to develop noiseless and noisy optimization on DE, respectively.

4. Materials and Methods

In this section, all the algorithms and techniques that were used to achieve the objectives of this research work have been presented as follows:

4.1. Discrete Differential Evaluation (DDE) Algorithm. The steps of DDE algorithm are stated as follows:

Initialization: initialize random population matrix $\pi = \{\pi_1, \pi_2, \pi_3, \ldots, \pi_{P_s}\}$ of size $P_s * N_d$ where P_s is the population size, and N_d is the dimension of problem space. Each population individual must be unique.

Evaluate fitness: obtain the best solution π_{best}^{t-1} in the population P_s by using equation (1).

Mutation: the following Equation can be used to find the mutant individual.

$$v_i^t = \begin{cases} \text{insert}\left(\pi_b^{t-1}\right), & \text{if}\left(r < P_m\right), \\ \text{swap}\left(\pi_b^{t-1}\right), & \text{otherwise}, \end{cases} \quad (2)$$

here, π_b^{t-1} represents the best solution from the previous generation in the target population; the P_m is the mutation probability; and insert and swap are merely the single insertion and swap moves, respectively, $r\in [0, 1]$ is a uniform random number.

Crossover: the crossover operation can be performed under the condition as shown in the following equation:

$$u_i^t = \begin{cases} \text{CR}\left(v_i^t, \pi_i^{t-1}\right), & \text{if}\left(r < P_c\right), \\ v_i^t, & \text{otherwise}, \end{cases} \quad (3)$$

where the CR and P_c are crossover operation and crossover probability, respectively. That means, the crossover operation is used if a randomly generated number, $r < P_c$, and then produce the individual u_i^t. Otherwise, the individual is selected as $u_i^t = v_i^t$.

Selection: selection operation that depends on fitness function can be calculated by equation (4). The selection is based on the existence of the correct amongst the test and target individuals.

$$\pi_i^t = \begin{cases} u_i^t, & \text{if}\left(f\left(u_i^t\right) \leq f\left(\pi_i^{t-1}\right)\right), \\ \pi_i^{t-1}, & \text{otherwise}. \end{cases} \quad (4)$$

4.2. Fitness Proportionate Selection Mode. Fitness proportionate selection (FPS) is a selection method in the genetic algorithm (GA) and RWS method is one example of FPS methods. According to a summary provided by [30], the first step of the FPS method is to calculate each individual's fitness value. Next, the individual's proportion of fitness within the entire group is calculated, which alludes to the probability that an individual is selected amid the process of selection. The probability that the individual i is chosen, is calculated in the following.

$$p_i = \frac{f_i}{\sum_i f_i}, \text{where } f_i \text{ is the fitness of the individual } i. \quad (5)$$

FPS is a very effective method for a parent to be selected. This gives everyone the privilege of becoming a parent with a proportional probability to their fitness value. Consequently, only the higher fitness value selections are made, which are eventually propagated to the generation that follows [31]. The RWS is one of the first methods for selection operator that has been used successfully in many applications of EAs [32]. Figure 1 shows the steps of the RWS.

In RWS, there is a circular wheel, as outlined below, along with a fixed point for choosing chromosomes arranged along the wheel's circumference. To choose the first parent, While population size < *pop_size* do

Generate *pop_size* random number r

Calculate cumulative fitness, total fitness (P_i) and sum of proportional fitness (*Sum*)

Spin the wheel *pop_size* times

If *Sum* < r then

Select the first chromosome, otherwise, select jth chromosome

End If

End While

Figure 1: Steps of roulette wheel selection [32].

the area of the wheel that comes ahead of the fixed point is chosen, and this process is also applied to choose the second parent. Notably, fitter individuals with greater wheel areas will have a higher probability of being selected whenever the wheel spins, which means that the chance of selecting an individual is directly informed by its fitness value. In the example given in Table 1, the fitness value of chromosome 1 is the highest, and so it has the greatest probability of being selected in comparison to the other chromosomes. Likewise, chromosome 5 has the lowest probability of being selected.

Figure 2 shows the implementation of the FPS by using RWS to select chromosomes.

Similarly, the concept of the FPS has been used in the transition rule on the ant system (AS) algorithm, but in a way that fits into the algorithmic components (such as the pheromone denoted by τ) in the transition rule. This is called the random proportional rule (RPR), which is an implementation of the transition rule that gives the probability that the ant k in the city r chooses to move the city s [33]. Equation (6) shows the RPR.

$$p_k(r,s) = \begin{cases} \dfrac{[\tau(r,s)].\ [\eta(r,s)]^{\beta}}{\sum_{\mu \epsilon J_k(r)} [\tau(r,\mu)].\ [\eta(r,\mu)]^{\beta}}, & \text{if } s \in J_k(r), \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

where τ represents the pheromone; $\eta = 1/\delta$ represents the inverse of the distance $\delta(r,s)$, $J_k(r)$ refers to the set of cities that have yet to be visited by ant k positioned in city r (to make the solution feasible); and the parameter ($\beta > 0$), determines the relative importance of pheromone versus distance. It is also worth noting that the probability value of the node visited previously is 0, which avoids repeated visits. On the other hand, the ant colony optimization (ACO) algorithm suffers from the premature convergence issue. For this reason, a new implementation of transition probability is presented in [34] for ACO, called global random proportional rule (GRP), to prevent this issue by enhancing a random proportional rule. The primary purpose of GRP is to enhance exploration through an increased probability of choosing solution components with a low pheromone trail in order to use the algorithm to encourage ants to choose a new shorter route to prevent premature convergence. Equation (7) shows the GRP. It shows the meaning of $x(r, s)$ as it reflects the effect in passing on edges from r to s during the trial paths.

Table 1: Fitness value and probability.

No. of individual	Fitness individual	Probability $\mathbf{p_i}$
Chromosome 1	7	0.35
Chromosome 2	4	0.20
Chromosome 3	5	0.25
Chromosome 4	3	0.15
Chromosome 5	1	0.05
Total	20	1.00

$$p_k(r,s) = \begin{cases} \dfrac{[\tau(r,s)].\ [\eta(r,s)]^{\beta}.\chi(r,s)}{\sum_{\mu \epsilon J_k(r)} [\tau(r,\mu)].\ [\eta(r,\mu)]^{\beta}.\chi(r,\mu)}, & \text{if } s \in J_k(r), \\ 0, & \text{otherwise}. \end{cases} \tag{7}$$

In our study, we use the same idea of GPR to use the FPS and implement its fitness as a nonmatching degree in order to give more opportunity for a group of solutions which can avoid failure of premature convergence. The fitness value will formulate on the no-matching between the best solution and the existing solutions in the groups after dividing the population's solutions into those groups based on the degree of difference. The probability of those groups is calculated in.

$$p_i = \frac{\text{Degree of difference in each group}}{\sum_i \text{Degree of difference in each group}}. \tag{8}$$

4.3. Proposed Enhance Discrete Differential Evolution (EDDE) Algorithm. This section includes the enhancement of the DDE algorithm to deal with the premature convergence and stagnation issues. Figure 3 shows the phases of enhancing and evaluating of proposed EDDE algorithm as follows:

The steps of the proposed enhanced DDE (EDDE) algorithm are presented as follows:

4.3.1. Population Initialization Stage. The first step is to initialize the population (solutions) randomly. As the goal of the QAP, it assigns only one location to one facility; therefore, the same facility should not be repeated in one solution. Since beginning with a better initial population gives a better solution, several researchers use heuristic algorithms to generate better population, and we apply the sequential sampling algorithm [35] for initiating the population. It is a basic form of the sequential constructive sampling algorithm which is summarized as follows:

Given a distance matrix, organize the locations in nondescending order of their distances in every row of the matrix. Beginning from allocating facility 1 to a location from first row of the matrix, a complete allocation is constructed by allocating the residual facilities to other locations probabilistically from the residual locations (the locations that are not presently allocated to any other facility in the current allocation) in each row. The process is continued until a complete allocation is constructed. The probability of allocating a residual location (in a row of the matrix) is

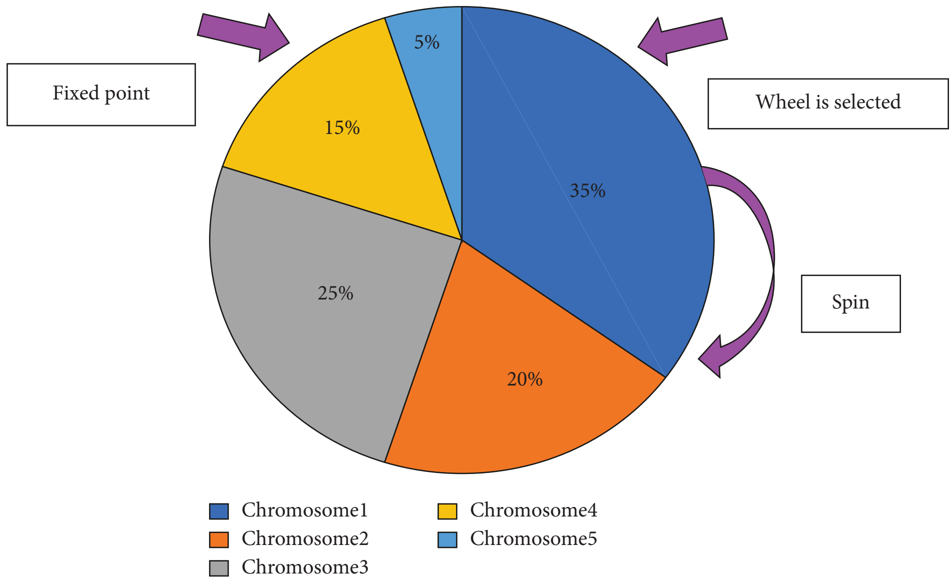

Figure 2: Selection chromosomes by using RWS.

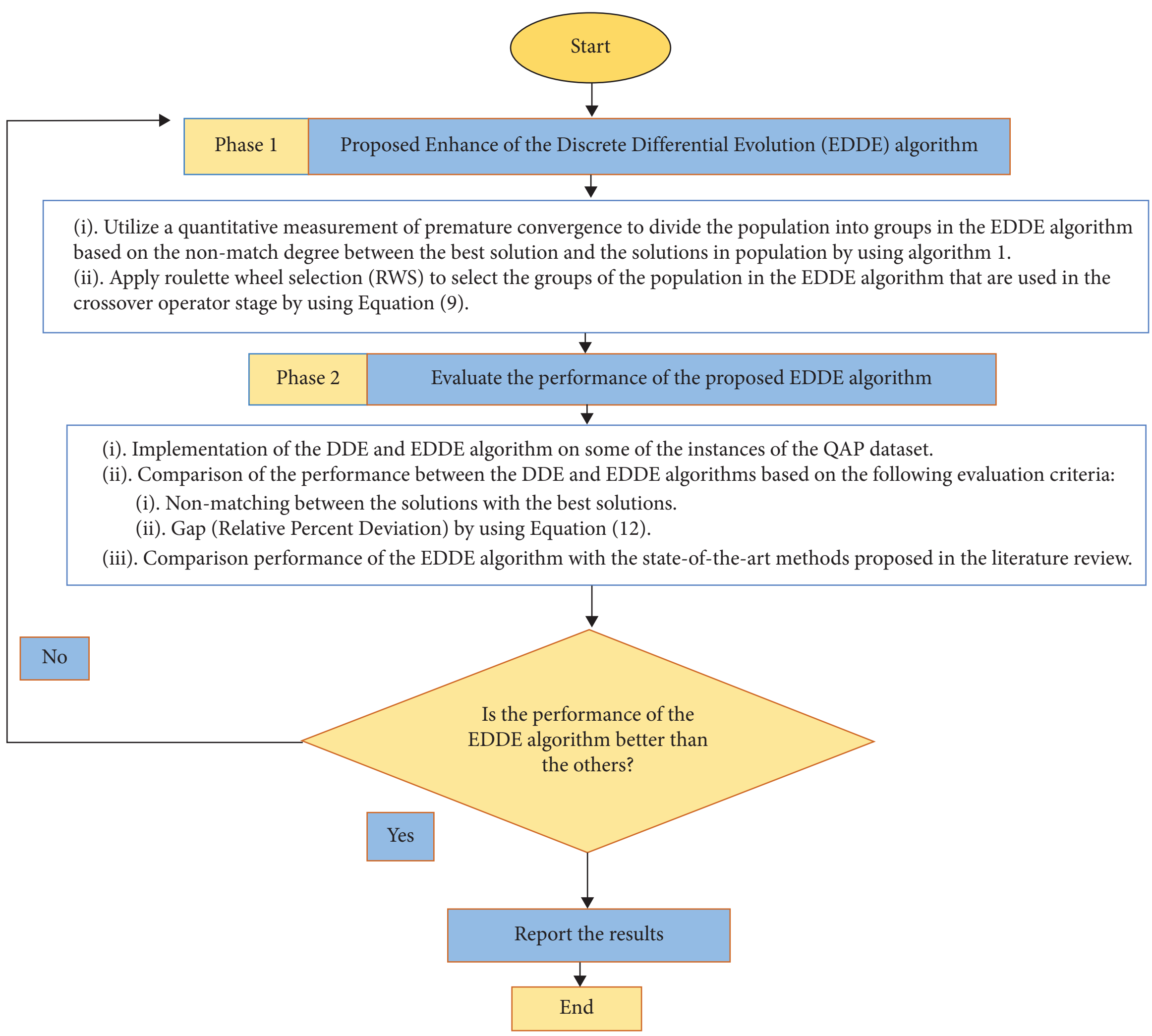

Figure 3: Phases of enhancing and evaluating of the proposed EDDE algorithm.

allocated in way so that the first residual location is assigned more probability than second one, second is more than third one, and so on. Next, for each residual location, cumulative probability is computed. Then, a random number $r\in[0,1]$ is produced, and the location that symbolizes the number in the cumulative probability range is accepted. Repeat the process until the population is full.

4.3.2. Finding Best Solution π^{t-1}_{best}. The next step is to find the best solution, π^{t-1}_{best} from the population based on the objective function by using equation (1).

4.3.3. Divide Population into Groups. In this step, the population (solutions) is divided into groups based on nonmatch degree (Degrees of Difference) with the best solution π^{t-1}_{best} by using Algorithm 1 as follows:

Suppose the best solution π_{best} is (1 2 3 4 5 6 8 7) and suppose the solutions found in the population are given in Table 2.

Based on the example given in Table 2, the degree of difference starts at 2, and their number is equal the size of the problem. The degrees of the differences can be obtained by comparing the solutions in Table 2 with the best solution as follows: solution π_1 has a degree of difference is 2 from the best solution, while solution π_2 has a degree of difference is 3 from the best solution, and solution π_3 has a degree of difference is 2 from the best solution, ..., etc. The last column in Table 2 shows the degree of differences between the best solution and the solutions in population. The population is divided into groups based on the degree of differences which is shown in Table 3.

Each group contains several solutions that are not necessarily equal to the degree of differences. For example, if the degree of difference is 2, then this group can contain one solution or more than one solution which differs from two degrees from the best solution.

4.3.4. Apply Roulette Wheel Selection (RWS) to Select the Solutions Group π^{t-1}_{gh}. In this section, the solutions group π^{t-1}_{gh} has been selected using the FPS that is given in equation (8) and RWS. Let the set of groups be $\{g_1, g_2, \ldots, g_m\}$ and $1 \le h \le m$ be the number of groups that divide the population based on the degree of the nonmatching between the best solution and the solutions found in the population, S is assigned to 1 to start with the first element of the selected group. Considering Table 3, the fitness value of group 1 is the highest that has more chance of selection than the other groups, and group 4 has very rare or no chance of selection, the implementation of the FPS is done by RWS to select the group. Table 4 presents the fitness value and probability of each group.

4.3.5. Mutation Operation Stage. This step is presented by using equation (2).

4.3.6. Crossover Operator Stage. This step has been included in finding the new solutions u^t_i by using crossover operator

```
Input: Best solution π_best^{t-1} and current solutions π_i
Output: Nonmatch degree
Nonmatch = 0
For each facility in the current solutions π_i
  If the location of the facility not-match in π_best^{t-1} and π_i
    Nonmatch = nonmatch + 1
  End if
End For
```

ALGORITHM 1: Nonmatching degree.

TABLE 2: Solutions found in the population.

Solution	Distributions of facilities to locations								Degree of differences
π_1	2	1	3	4	5	6	8	7	2
π_2	1	2	3	6	4	5	7	8	5
π_3	1	3	2	4	5	6	8	7	2
π_4	7	8	3	4	5	2	6	1	5
π_5	1	4	5	2	3	6	8	7	4
π_6	1	2	3	7	4	8	5	6	5
π_7	1	5	2	8	6	4	2	7	6
π_8	8	5	6	4	7	2	1	3	8
π_9	1	4	5	2	3	6	8	7	4
π_{10}	8	2	3	4	5	7	6	1	4

TABLE 3: Division of the population into groups.

Group	Degree of differences	Number of solutions
Group_1	8	1
Group_2	4	3
Group_3	5	3
Group_4	2	2
Group_5	6	1

TABLE 4: Fitness value and probability of each group.

Group	Degree of difference in each group	Probability $\mathbf{p_i}$
Group_1	8	8/25 = 0.32
Group_2	4	4/25 = 0.16
Group_3	5	5/25 = 0.20
Group_4	2	2/25 = 0.08
Group_5	6	6/25 = 0.24
Total	25	1

between the solutions found in the group π^{t-1}_{gh} and the solution generated from the mutation stage v^t_i. Equation (9) shows this step.

$$u^t_i = \begin{cases} \mathrm{CR}\left(v^t_i, \pi^{t-1}_{\text{gh}}\right), & \text{if } r < P_c, \\ v^t_i, & \text{otherwise,} \end{cases} \tag{9}$$

where the $P_c\in[0,1]$ is the crossover probability. If the $r < P_c$, the ULX operator [16] is used to generate the u^t_i otherwise the v^t_i is selected. The process of the ULX is as follows: first, the similar locations in both parents are check-up and then copied to the child (new solution). The second step involves selecting an item randomly and uniformly from both parents that have not yet been selected for the child after checking the unassigned locations from left to right. Finally, the rest of the items are randomly assigned to the locations.

4.3.7. Selection Operator Stage. The selection operator is the final step of the DDE algorithm which depends on the objective function to choose the best solution after the crossover step. The new solutions π^t_{gh} are the solutions either in the u^t_i or in the group π^{t-1}_{gh}. Equation (10) shows the selection step.

$$\pi^t_{\text{gh}} = \begin{cases} u^t_i, & \text{if } \left(f\left(u^t_i\right) < f\left(\pi^{t-1}_{\text{gh}}\right)\right), \\ \pi^{t-1}_{\text{gh}}, & \text{otherwise.} \end{cases} \tag{10}$$

4.3.8. Check Solution Stagnation Stage. This section included the number of trials the EDDE algorithm waits for before considering the solution is stagnated and should be regenerated randomly after 10 trials. Equation (11) shows this step.

$$W_i = \begin{cases} 0, & \pi^t_{\text{ghi}} = u^t_i, \\ W_i + 1, & \pi^t_{\text{ghi}} = \pi^{t-1}_{\text{ghi}}. \end{cases} \tag{11}$$

The steps of the proposed EDDE algorithm are illustrated through the pseudo code shown in Algorithm 2.

The flowchart of EDDE algorithm given in Figure 4 as follows:

5. Computational Results and Discussion

This section elucidates the efficiency of the improved algorithm (EDDE). In order to encode the improved algorithm, MATLAB (R2018b (9.5.0.944444), 64 bit (win64), August 28-2018, and License Number: 968398) is employed on a PC with Intel (R) Core (TM) i7-3770 CPU @3.40 GHz under MS Windows 10 and 8 GB RAM. This section comprises two parts: the first part highlights the parameters used for the proposed algorithm, whereas the second part discusses the results of the study.

5.1. Parameters Tuning. The same parameter settings that were suggested in [8] are used for the DDE algorithm for solving the QAP. These parameters are related to the population size P_s, mutation probability P_m, probability of crossover P_c, and Number of runs as given in Table 5.

Our EDDE algorithm is implemented and tested on some of the QAP instances of various sizes from the QAPLIB website and then compared with the traditional DDE algorithm. The criteria used for comparisons are based on the gap (Relative Percent Deviation) of the solutions by the algorithms and nonmatching between the solutions.

5.2. Comparison Based on the Nonmatching between the Solutions with the Best Solution. The value of the nonmatching was calculated through a comparison between the arrangement of facilities occupied by the locations in the best solution and between those facilities occupied by the locations in those solutions in the population by using the following algorithm. Then, the crossover is performed between the best solution and the selected solution in the group to produce better offspring. So, if there is a big match between these parents then there is a chance that the offspring will be identical to one of the two parents, and thus the premature convergence will occur. Certainly, the solutions having high degrees of nonmatching will increase the diversity of the population by creating new solutions (offspring). Table 6 shows this comparison as follows:

The nonmatching values are computed by Algorithm 1, if the value of the average nonmatching is small this means that the solutions in the population are identical with the best solution due to a lack of diversification. On the contrary, if the value of the average nonmatching is high, then this means that the solutions in the population have a difference from the best solution, and in this way, the diversification is preserved.

The above results were obtained by finding the degrees of nonmatching between the solutions and the best solution in all iterations after 10 runs, then sum them and dividing by the number of iterations multiplied number of runs and multiplied size of the population. The crossover stage in the DDE algorithm produced solutions with high degrees of matching and in some cases are completely identical to the best solution. For example, the mean of nonmatching of the solutions of the instances (Tai20a, Tai20b, Tai25a, Tai25b, Tai30a, Tai30b, Tai35a, Tai35b, and Tai40a) are 0, 0, 0, 0, 0, 0, 0, 0, and 0, respectively, whilst the mean of nonmatching of the solutions these instances are (4.136, 4.363, 4.412, 7.415, 4.419, 10.643, 4.919, 6.595, and 16.890, respectively) by using the EDDE algorithm.

The RWS technique that was applied to select a population of individual groups for the crossover operation to give a higher probability to selected the nonmatching solution to generate solutions through this stage with a high percentage of difference with the best solution i.e., in other words, the locations of those facilities that these solutions contain are different enough to allow the generation of new solutions so that those locations are highly different degree compared to existing locations in the best solution. In this way, it is possible to preserve the diversity of the population and thus reduce the issue of premature convergence. A graphic representation of Table 6 is shown in Figure 5.

5.3. Comparison Based on the Gap (Relative Percent Deviation). This section includes the comparison between DDE and EDDE algorithms based on the gap (relative percent deviation) that is given by

$$\text{Gap} = \left(\frac{\text{BS} - \text{BKS}}{\text{BKS}}\right) * 100, \tag{12}$$

t: Iteration
$t_{\max}$: Maximum number of iterations
S: Solution in group π_{gh}^{t-1}
$|\pi_{gh}^{t-1}|$: Maximum number of solutions in group π_{gh}^{t-1}
$W_{\max}$: Maximum number of wait for improvement of the solution
Population Initialization Stage
For $t = 1$ to $t_{\max}$
  Find the best solution π_{best}^{t-1} by using equation (1)
  Divide population to groups based on nonmatching with π_{best}^{t-1} by using Algorithm 1
  Determining solutions group π_{gh}^{t-1} by using RWS
  For $S = 1$ to $|\pi_{gh}^{t-1}|$
    Mutation stage by using equation (2)
    Crossover stage by using equation (9)
    Selection stage by using equation (10)
    Check solution stagnation stage W by using equation (11)
    If keep same the solution
      $W_i = W_i + 1$
    Else
      $W_i = 0$
    End If
    If $Wi > W_{\max}$
      Regenerate the solution at random after 10 trials
      $W_i = 0$
    End If
  End For
End For

Algorithm 2: EDDE algorithm.

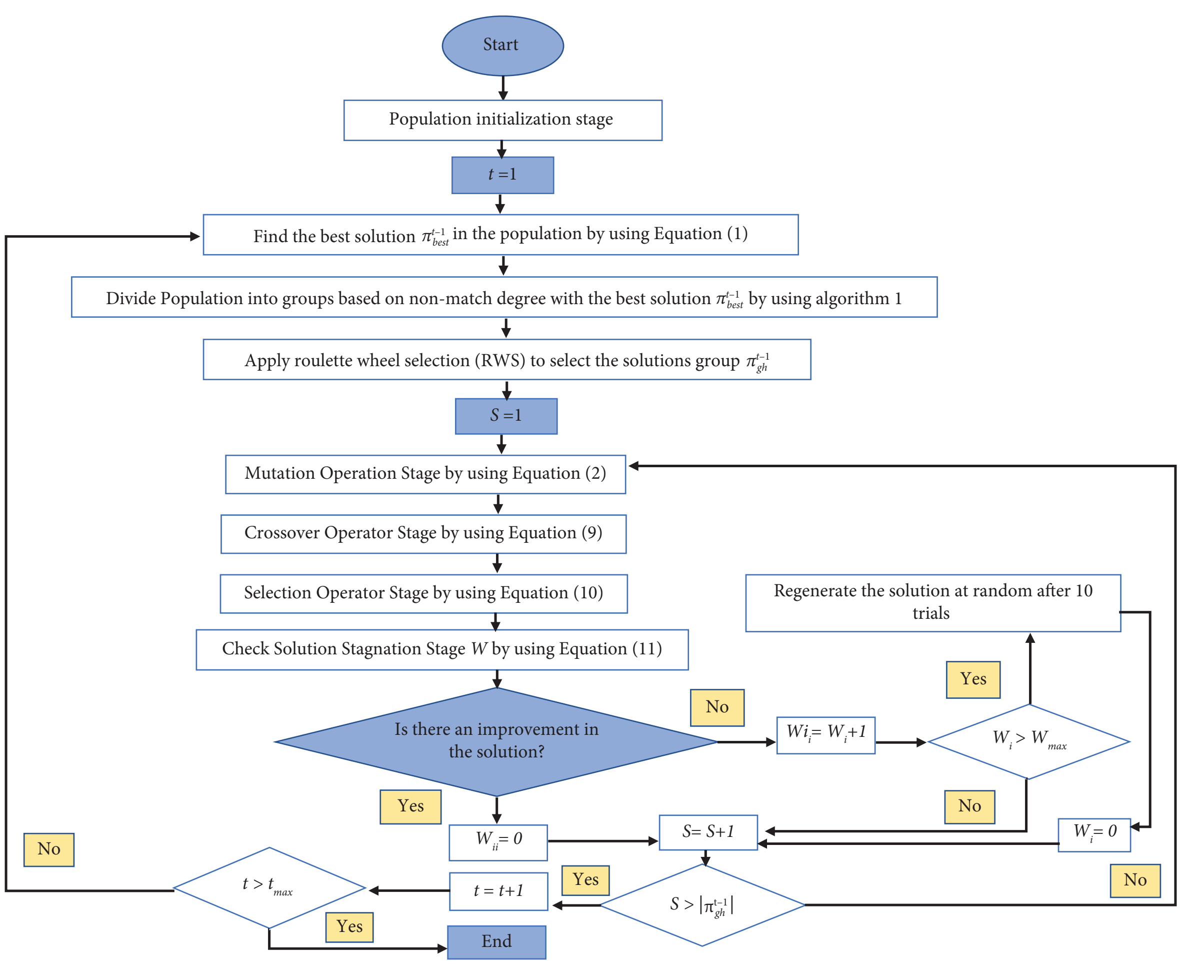

Figure 4: Flowchart of the proposed EDDE algorithm.

Table 5: Parameter setting of the DDE algorithm [8].

Parameter	Value
P_s size of population	100
P_m probability of mutation	0.9
P_c probability of crossover	0.9
Number of Run	10

Table 6: Mean of nonmatching solutions obtained by DDE and EDDE algorithms.

No.	Instances	DDE algorithm	EDDE algorithm
1	Tai20a	0	4.136
2	Tai20b	0	4.363
3	Tai25a	0	4.412
4	Tai25b	0	7.415
5	Tai30a	0	4.419
6	Tai30b	0	10.643
7	Tai35a	0	4.919
8	Tai35b	0	6.595
9	Tai40a	0	16.890
10	Tai40b	0.167	11.368
11	Tai50a	0	5.725
12	Tai50b	0	8.608
13	Tai60a	0.127	6.000
14	Tai60b	0	9.456
15	Tai80a	0	6.754
16	Tai80b	0.291	10.398
17	Tai100a	0.520	7.451
18	Tai100b	0.927	12.015
19	Sko42	0.037	6.531
20	Sko49	0.086	7.777
21	Sko56	0.087	7.797
22	Sko64	0.116	8.441
23	Sko72	0.078	9.565
24	Sko81	0.187	10.137
25	Sko90	0.176	12.133
26	Sko100a	0.420	11.768
27	Sko100b	0.233	11.297
28	Sko100c	0.515	11.328
29	Sko100d	0.276	12.405
30	Sko100e	0.090	11.759
31	Sko100f	0.128	10.174

where BS denotes the best solution found by any algorithm after 10 runs, while the BKS refers to the optimal solution or the best-known solution reported in QAPLIB. In this study, the EDDE algorithm addresses the issue of premature convergence which has responded positively by improving the gaps obtained by the DDE algorithm. Table 7 compares the results between the DDE and the EDDE algorithms based on the gap of the obtained solutions. The results of this comparison show the average of the gaps obtained by the DDE algorithm before the enhancement is 3.369, which is refined to 1.297 after enhancement. This is conducted by applying the concept of the FPS to generate new solutions with a high degree of nonmatching in order to reduce the probability of creating solutions that are matched with the best solution by the crossover operator stage. Figure 6 shows the graphic representation of Table 7.

The values of the gap have been calculated using equation (12), and the less value of the gap is better than the high value of the gap.

The graphical representation of Table 7 is shown in the following figure:

6. Statistical Analysis

This section has discussed the results of this study using the SPSS software as follows:

6.1. Normal Distribution Test. We are testing whether the data follow a normal distribution by using the following hypotheses:

Null Hypothesis: the data are normally distributed. **Alternative Hypothesis:** the data do not follow a normal distribution.

In order to check normality, we used the one-sample Kolmogorov–Smirnov Test. Table 8 presents the results of this test as follows:

In the SPSS software, the p value is labeled "Sig.," from above Table both p values are below 0.05 so we rejected the Null hypothesis in our test and accept the alternative hypothesis, Hence, this does not follow a normal distribution.

6.2. Nonparametric Test. This test includes three Tables 9–11, Table 9 includes descriptive statistics, Table 10 relates to Wilcoxon Signed Ranks Test, and Table 11 includes test statistics.

The following Table presents the Wilcoxon Signed Ranks Test as follows:

In order to use Wilcoxon Signed Ranks test, the following hypotheses must be used:

Null hypothesis: there is no difference in the mean of the two samples.

Alternative hypothesis: there is a real difference between the mean of the two samples.

From the above table, we find that the Wilcoxon Signed Ranks test value is −4.831, and p value is 0.000 which is less than 0.05; therefore, we rejected the null hypothesis and accepted the alternative hypothesis.

7. Theoretical Analysis and Critical Explanation of the Performance of Enhancement Proposed

Research indicates that metaheuristic algorithms, including EAs, are among the current state-of-the-art algorithms for addressing NP-hard problems. While EAs benefit from global exploration search, issues such as stagnation, premature convergence occur in the iterative operation and slow in the exploitation mechanisms. Although the exploitation issue in the DDE algorithm has been addressed by the recent study [36]; however, the critical reading of the literature indicates that stagnation, premature convergence issues have not been investigated in the DDE algorithm associated with EAs to solve the QAP. Therefore, further

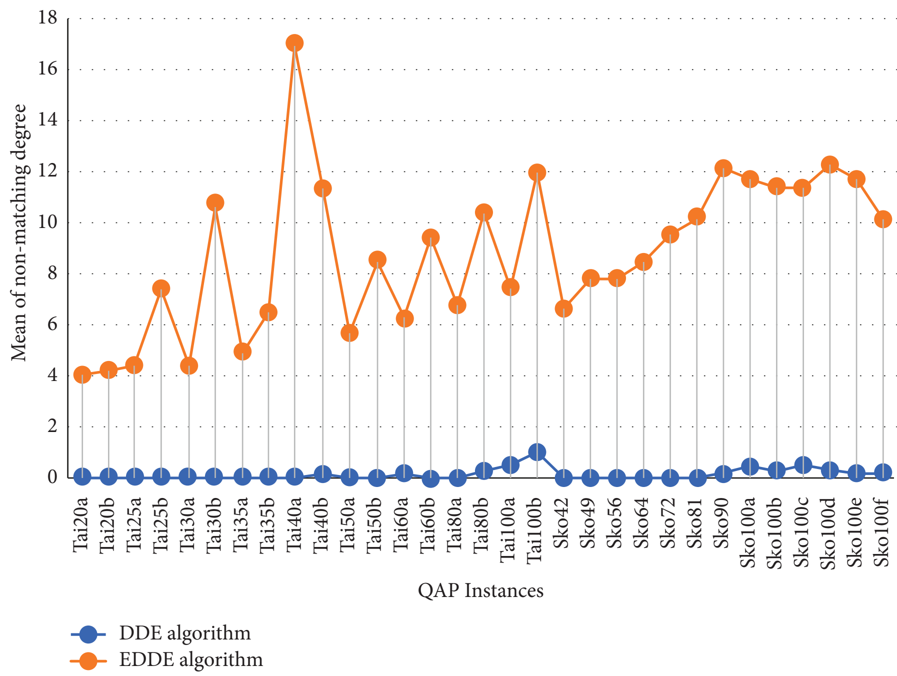

FIGURE 5: Graphic representation of Table 6.

TABLE 7: Best gaps found by DDE and EDDE algorithms.

No.	Instances	DDE algorithm	EDDE algorithm
1	Tai20a	2.690	0
2	Tai20b	0.497	0
3	Tai25a	3.433	0
4	Tai25b	2.672	0
5	Tai30a	3.729	0
6	Tai30b	0.429	0
7	Tai35a	5.213	0
8	Tai35b	1.182	0.802
9	Tai40a	3.503	0.37
10	Tai40b	5.975	0.407
11	Tai50a	4.068	1.453
12	Tai50b	2.912	1.847
13	Tai60a	4.799	1.264
14	Tai60b	1.889	1.294
15	Tai80a	5.975	2.625
16	Tai80b	4.428	2.08
17	Tai100a	3.561	2.35
18	Tai100b	6.867	2.734
19	Sko42	3.681	1.328
20	Sko49	2.155	1.046
21	Sko56	3.389	1.973
22	Sko64	1.360	1.740
23	Sko72	2.934	2.408
24	Sko81	3.586	2.000
25	Sko90	3.292	1.712
26	Sko100a	3.434	2.100
27	Sko100b	3.259	2.037
28	Sko100c	3.511	2.308
29	Sko100 d	3.004	1.708
30	Sko100e	3.709	1.821
31	Sko100f	3.317	2.803
	Average	**3.369**	**1.297**

research is required to enhance the algorithm that accounts for current challenges, which is the area of contribution of our study, specifically the avoidance of stagnation and premature convergence in the DDE algorithm. The following sections have discussed the theoretical analysis and critical explanation of the performance of enhancement proposed.

7.1. Analysis of the Stagnation Situation for DDE and EDDE Algorithms. This section included the comparison between DDE and EDDE algorithms illustrating the stagnation situation for DDE and how EDDE avoids this issue. This comparison applied an instance of the QAP data called "Tai12a" as a case study. The size of this instance is 12 facilities/locations, and the optimal solution of this instance in the QAP database is 224416. Suppose the number of solutions in a population in the DDE and EDDE algorithms is 7 solutions, the results of this comparison are discussed in Tables 12 and 13 as follows:

7.2. Critical Explanation of the Performance of Enhancement Proposed. A key point revealed by the critical analysis of the improved algorithm's performance (i.e., EDDE) is that premature convergence arises from limited population diversity. Due to this, the algorithm generates solutions that are comparable in the evolutionary process, losing its ability to enhance solutions and leading to stagnation. These issues were considered in this study by introducing quantitative measurement to premature convergence in the improved algorithm (EDDE) based on the degree of nonmatching between the population solutions. Then, the population was

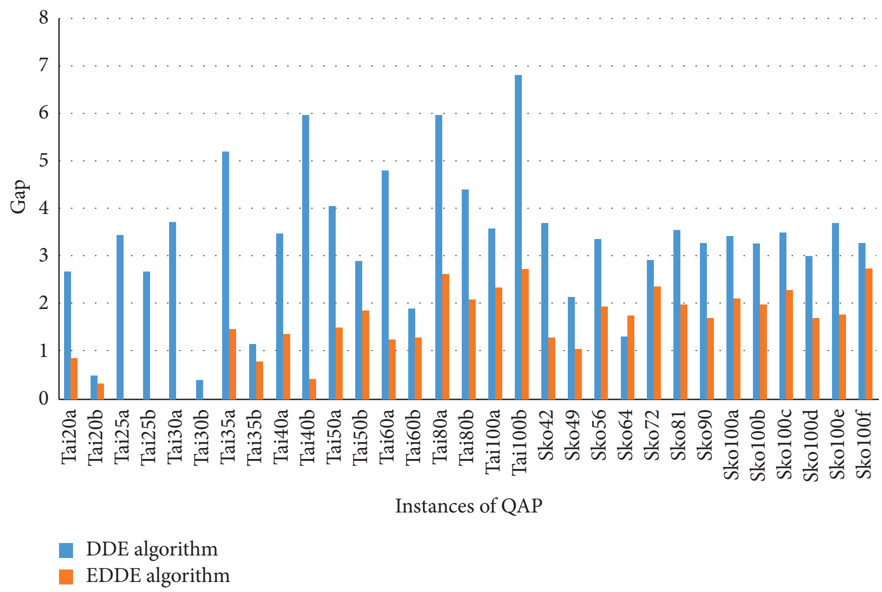

Figure 6: Graphic representation of Table 7.

Table 8: One-sample Kolmogorov–Smirnov test.

		DDE algorithm	EDDE algorithm
N		31	31
Normal parameters[a,b]	Mean	3.369	1.361
	Std. deviation	1.463	0.948
Most extreme differences	Absolute	0.177	0.159
	Positive	0.177	0.150
	Negative	−0.123	−0.159
Test statistic		0.177	0.159
Asymp. sig. (2-tailed)		0.014[c]	0.045[c]

[a]Test distribution is normal. [b]Calculated from data. [c]Lilliefors significance correction.

Table 9: Descriptive statistics.

	N	Mean	Std. deviation	Minimum	Maximum	Percentiles 25th	Percentiles 50th (median)	Percentiles 75th
Best gap by DDE algorithm	31	3.369	1.463	0.429	6.867	2.690	3.433	3.729
Best gap by EDDE algorithm	31	1.361	0.948	0.000	2.803	0.370	1.708	2.080

Table 10: Ranks.

		N	Mean rank	Sum of ranks
Best gap by EDDE algorithm—best gap by DDE algorithm	Negative ranks	30[a]	16.48	494.50
	Positive ranks	1[b]	1.50	1.50
	Ties	0[c]		
	Total	31		

[a]Best gap by EDDE algorithm < best gap by DDE algorithm. [b]Best gap by EDDE algorithm > best gap by DDE algorithm. [c]Best gap by EDDE algorithm = best gap by DDE algorithm.

Table 11: Test statistics[a].

	Best gap by EDDE algorithm—best gap by DDE algorithm
Z	−4.831[b]
Asymp. sig. (2-tailed)	0.000

[a]Wilcoxon signed ranks test. [b]Based on positive ranks.

Table 12: Comparison between the DDE and EDDE algorithms and the illustration of the stagnation situation for DDE and how EDDE avoids it.

	DDE algorithm solutions												Cost	Gap	Nonmatching degree	Solution stagnation start	Waiting for solution improvement
$\boldsymbol{\pi_{\mathbf{Best}}}$	**4**	**5**	**11**	**7**	**1**	**8**	**12**	**10**	**9**	**3**	**6**	**2**	272700	3.84	—	—	
π_1	4	5	11	7	1	8	12	10	9	3	6	2	233040	3.84	0	113 iteration	
π_2	4	5	11	7	1	8	12	10	9	3	6	2	233040	3.84	0	98 iteration	
π_3	4	5	11	7	1	8	12	10	9	3	6	2	233040	3.84	0	101 iteration	
π_4	4	5	11	7	1	8	12	10	9	3	6	2	233040	3.84	0	103 iteration	
π_5	4	5	11	7	1	8	12	10	9	3	6	2	233040	3.84	0	99 iteration	
π_6	4	5	11	7	1	8	12	10	9	3	6	2	233040	3.84	0	85 iteration	
	EDDE algorithm solutions												*Cost*	*Gap*	*Nonmatching degree*	*Group*	
$\boldsymbol{\pi_{\mathbf{Best}}}$	**4**	**5**	**11**	**3**	**7**	**10**	**12**	**9**	**8**	**6**	**1**	**2**	230704	2.80	—	—	—
π_1	4	12	11	3	2	7	5	8	9	6	1	10	232164	3.45	7	1	9 times
π_2	4	5	11	7	1	8	12	10	9	3	6	4	233040	3.84	8	2	7 times
π_3	4	5	11	3	7	10	12	9	8	6	1	2	230704	2.80	0	—	1 times
π_4	4	5	11	10	1	7	12	8	3	6	9	2	235704	5.03	7	1	4 times
π_5	4	2	11	6	12	1	7	5	9	3	10	8	243100	8.33	10	3	3 times
π_6	*6*	*5*	*1*	*7*	*2*	*4*	*8*	*12*	*3*	*11*	*10*	*9*	315722	31.95	11	4	Regenerated

$\boldsymbol{\pi_{\mathbf{Best}}}$: the π_{Best} (bold) is the best solution in the population of the DDE and EDDE algorithms. Cost: the best value of the objective function obtained by DDE and EDDE algorithms by using equation (1). Nonmatching is the number of different facility locations between two solutions and calculated by using Algorithm 1. Group: a set of individuals with the same nonmatching degree as the best individual in EDDE algorithm. Gap: the best value obtained by equation (12). Solution stagnation start: the number of iterations that the solution has not changed (not improved) in the DDE algorithm, where max iteration is 1000. Waiting for solution improvement: the number of trials the EDDE algorithm waits for before considering the solution is stagnated and should be regenerated randomly (italic) the default value is 10 trials.

Table 13: Example of the process to generate the new solution (offspring) by crossover stage in the EDDE and DDE algorithms.

Process	Solution	Nonmatching degree
EDDE algorithm		
Select best solution (π_{Best})	4 5 11 3 7 10 12 9 8 6 1 2	—
Mutation stage to select the parent 1	4 3 11 5 7 10 12 9 8 6 1 2	2
Select individual as parent 2 from the group that selection by RWS	4 2 11 6 12 1 7 5 9 3 10 8	10
Crossover stage between parent 1 and parent 2	**4 2 11 6 7 1 12 9 8 3 10 5**	6
DDE algorithm		
Select best solution (π_{Best})	4 5 11 7 1 8 12 10 9 3 6 2	—
Mutation stage to select the parent 1	4 5 1 7 11 8 12 10 9 3 6 2	2
Select individual from a population as parent 2	4 5 11 7 1 8 12 10 9 3 6 2	0
Crossover stage between parent 1 and parent 2	*4 5 11 7 1 8 12 10 9 3 6 2*	0

It can be noted that the new solution (offspring) generated by the EDDE algorithm (bold) by the crossover stage has a high degree of nonmatching with the best solution which means the diversity of the population will be increased. While the opposite is in the DDE algorithm that generated offspring (italic) by the stage of crossover that degree of nonmatching was zero compared with the best solution, which means that the new solution is matched with the best solution which leads to loss of diversity.

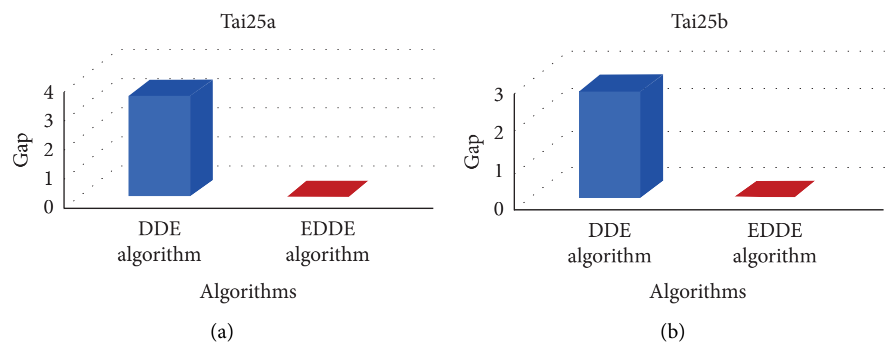

Figure 7: (a) Solution of the instance Tai25a, and (b) solution of the instance Tai25b.

Table 14: Best values of the gap found by the proposed EDDE algorithm and others algorithms.

Instances of QAP	ACO [37]	BA [37]	GA [37]	PSO [37]	Modified PSO [37]	TS [37]	Our proposed EDDE algorithm
nug8a	0.00	0.00	0.42	0.19	0.84	3.00	0.00
nug12	0.00	4.95	7.44	3.98	4.22	5.96	0.00
tai12a	0.00	9.07	11.11	10.86	9.49	7.63	0.00
esc16a	0.00	2.65	6.76	6.47	2.94	2.94	0.00
tai20a	1.02	11.90	13.36	13.00	10.58	5.07	0.00
nug28	0.74	16.10	18.11	16.21	13.16	4.26	0.36
tai30a	2.22	12.24	13.22	12.21	10.91	5.27	0.00
esc32e	0.00	0.00	17.50	0.00	0.00	0.00	0.02
Average	**0.498**	**7.114**	**10.990**	**7.865**	**6.518**	**4.266**	**0.004**

divided into individual groups based on the degree of nonmatching between the population solutions and the best solution. Moreover, the new utilization of the RWS has helped to a greater nonmatching degree to select a population of individual groups to be able to generate a new solution with more opportunities to avoid the occurrence of premature convergence in the improved algorithm EDDE.

Figure 7 illustrates a practical example of the impact of the positive contributions of this research on the performance of the DDE algorithm (before enhancement) and EDDE algorithm (after enhancement). The example included the solutions of two QAP instances, Tai25a and Tai25b; the results of solutions in those instances show the DDE algorithm was unable to achieve the optimal solution in both instances (Tai 25a and Tai25b), as it obtained the best gap of 3.433 and 2.672, respectively. While the EDDE algorithm has achieved the optimal solutions for those instances, hence it has achieved the best gap is 0 and 0, respectively. The blue color in Table 7 indicates the solution by the DDE algorithm, while the red in Table 7 indicates the solution by the EDDE algorithm.

7.3. Comparison Performance of the EDDE Algorithm with the State-of-the-Art Methods. This section has included the comparison between the EDDE algorithm and the state-of-the-art methods that solved QAP. A recent study [37] discussed a performance study of metaheuristic approaches for the QAP that includes the ACO, GA, PSO, bat algorithm (BA), tabu search (TS) algorithm, and a modified variant of the discrete PSO algorithm.

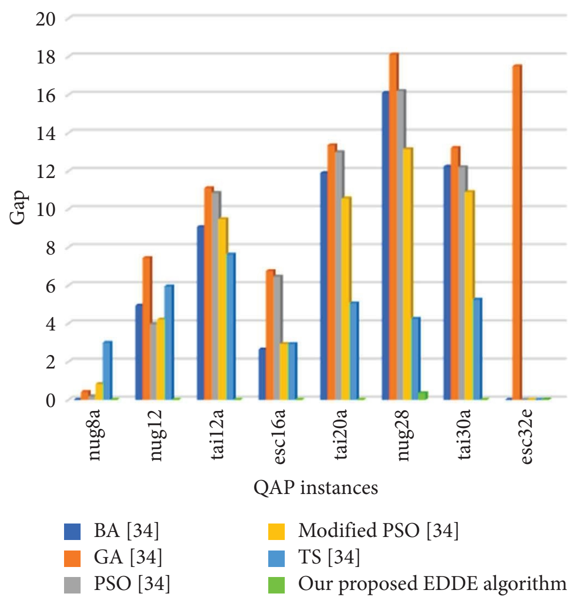

Figure 8: Graphic representation of Table 14.

Table 14 shows the results of the comparison between the proposed algorithm EDDE and among those algorithms that have been proposed in the recent literature that dealt with solutions of QAP instances. The results of that comparison showed that the proposed algorithm EDDE had a more efficient performance than others to converge to the optimal solutions. Moreover, the EDDE algorithm obtained 0.004 of the best average value of the gap whilst the other algorithms (ACO, BA, GA, PSO, Modified PSO, and TS) have obtained 0.498, 7.114, 10.990, 7.865, 6.518, and 4.266, respectively. Figure 8 shows the graphic representation of these results as follows.

7.4. Limitations of the Study. Recently, the authors in the study [16] utilized ULX for the DDE algorithm to solve the QAP. However, there are many crossover operators suggested for the QAP in other algorithms such as the genetic algorithm [38]. Furthermore, this study used only the DDE algorithm which belongs to the EAs category although there are many known algorithms that belong to the category of EAs.

8. Conclusion

This study aims to address the limitations of evolutionary algorithms (EAs), which can suffer from premature convergence and stagnation. Despite the advantages of EAs, these issues can hinder their ability to solve complex optimization problems (COPs). Therefore, to avoid premature convergence in the EAs by ensuring the diversity of the population considered to enhance the selection operations by proposing diversity measures between solutions, hence the enhancement of convergence to optimal solutions. We implemented our proposed enhancement DDE as an algorithm of the evolutionary algorithm category. To evaluate the performance of the DDE algorithm before and after the enhancement, it was tested on some benchmark instances from the QAPLIB website. Subsequently, we compared the results obtained by DDE and enhanced DDE algorithms based on the gap and nonmatching solutions. Our comparative study reveals that the EDDE algorithm is more efficient than the traditional DDE algorithm. Moreover, the proposed algorithm is better than the other algorithms, including ACO, GA, PSO, bat algorithm (BA), tabu search (TS), and a modified variant of the discrete PSO algorithm in convergence to optimal solutions of some QAP instances. We suggest applying the proposed algorithm EDDE to solve other problems of the COPs in future work, such as capacitated vehicle routing problem (CVRP) and nurse scheduling problem (NSP).

Data Availability

The data that used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] R. KumarBhati and A. Rasool, "Quadratic assignment problem and its relevance to the real world: a survey," *International Journal of Computer Application*, vol. 96, no. 9, pp. 42–47, 2014.

[2] T. C. Koopmans and M. Beckmann, "Assignment problems and the location of Activities," *Econometrica*, vol. 25, pp. 53–76, 1957.

[3] K. Hussain, M. N. Mohd Salleh, S. Cheng, and Y. Shi, "Metaheuristic research: a comprehensive survey," *Artificial Intelligence Review*, vol. 52, pp. 1–43, 2018.

[4] S. Samanta, D. Philip, and S. Chakraborty, "A quick convergent artificial bee colony algorithm for solving quadratic assignment problems," *Computers & Industrial Engineering*, vol. 137, Article ID 106070, 2019.

[5] Z. Beheshti and S. M. H. Shamsuddin, "A review of population-based meta-heuristic algorithm," *International Journal of Advances in Soft Computing and Its Applications*, vol. 5, no. 1, pp. 1–35, 2013.

[6] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[7] J. Kushida, K. Oba, A. Hara, and T. Takahama, "Solving quadratic assignment problems by differential evolution," in *Proceedings of the The 6th International Conference on Soft Computing and Intelligent Systems, and The 13th International Symposium on Advanced Intelligence Systems*, vol. 1, pp. 639–644, Kobe, Japan, November 2012.

[8] M. F. Tasgetiren, Q. K. Pan, P. N. Suganthan, and I. E. Dizbay, "Metaheuristic algorithms for the quadratic assignment problem," in *Proceedings of the 2013 IEEE Symposium on Computational Intelligence in Production and Logistics Systems (CIPLS)*, pp. 131–137, Singapore, April 2013.

[9] M. Yang, C. Li, Z. Cai, and J. Guan, "Differential evolution with auto-enhanced population diversity," *IEEE Transactions on Cybernetics*, vol. 45, no. 2, pp. 302–315, 2015.

[10] S. Prabha and R. Yadav, "Differential evolution with biological-based mutation operator," *Engineering Science and Technology, an International Journal*, vol. 23, no. 2, pp. 253–263, 2020.

[11] K. Zhang, P. Mu, Y. Zhang, Z. Jin, and Q. Huang, "Enhancing differential evolution algorithm with repulsive behavior," *Soft Computing*, vol. 24, no. 12, pp. 9279–9305, 2020.

[12] J. Lampinen and I. Zelinka, "Mixed integer-discrete-continuous optimization by differential evolution," in *Proceedings of the International Conference on Soft Computing*, pp. 71–76, Riva del Garda, Italy, September, 1997.

[13] Q. K. Pan, M. F. Tasgetiren, and Y. C. Liang, "A discrete differential evolution algorithm for the permutation flowshop scheduling problem," *Computers & Industrial Engineering*, vol. 55, no. 4, pp. 795–816, 2008.

[14] L. Jouni and Z. Ivan, "On Stagnation of the differential evolution algorithm," *Proceedings of the. MENDEL*, pp. 76–83, 2000.

[15] A. W. Mohamed, "RDEL: restart differential evolution algorithm with local search mutation for global numerical optimization," *Egyptian Informatics Journal*, vol. 15, no. 3, pp. 175–188, 2014.

[16] A. S. Hameed, B. M. Mohd, N. H. Choon, and M. L. Mutar, "Improved discrete differential evolution algorithm in solving quadratic assignment problem for best solutions," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 12, pp. 434–439, 2018.

[17] G. Onwubolu and D. Davendra, "Scheduling flow shops using differential evolution algorithm," *European Journal of Operational Research*, vol. 171, no. 2, pp. 674–692, 2006.

[18] Y. Liu, M. H. Yin, and W. X. Gu, "An effective differential evolution algorithm for permutation flow shop scheduling problem," *Applied Mathematics and Computation*, vol. 248, pp. 143–159, 2014.

[19] G. L. Deng and X. S. Gu, "A hybrid discrete differential evolution algorithm for the noidle permutation flow shop scheduling problem with makespan criterion," *Computers and Operations Research*, vol. 39, no. 9, pp. 2152–2160, 2012.

[20] Q. K. Pan, L. Wang, L. Gao, and W. D. Li, "An effective hybrid discrete differential evolution algorithm for the flow shop scheduling with intermediate buffers," *Information Sciences*, vol. 181, no. 3, pp. 668–685, 2011.

[21] E. Nicoară, "Mechanisms to avoid the premature convergence of genetic algorithms," *Bul. Univ. Pet. Gaze din Ploiesti, Mat*, vol. 61, no. 1, pp. 87–96, 2009.

[22] M. Shikha Malik and M. Sumit Wadhwa, "Preventing premature convergence in genetic algorithm using DGCA and elitist technique," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 4, no. 6, p. 2277, 2014.

[23] C. Lamini, S. Benhlima, and A. Elbekri, "Genetic algorithm based approach for autonomous mobile robot path planning," *Procedia Computer Science*, vol. 127, pp. 180–189, 2018.

[24] S. M. Guo, C. C. Yang, P. H. Hsu, and J. S. H. Tsai, "Improving differential evolution with a successful-parent-selecting framework," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 717–730, 2015.

[25] L. Cui, Q. Huang, G. Li et al., "Differential evolution algorithm with tracking mechanism and backtracking mechanism," *IEEE Access*, vol. 6, pp. 44252–44267, 2018.

[26] C. Y. Kao, S. L. Hung, and B. Setiawan, "Two strategies to improve the differential evolution algorithm for optimizing design of truss structures," *Advances in Civil Engineering*, vol. 2020, Article ID 8741862, 20 pages, 2020.

[27] W. Deng, J. Xu, X. Z. Gao, and H. Zhao, "An enhanced MSIQDE algorithm with novel multiple strategies for global optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 3, pp. 1578–1587, 2022.

[28] C. Li, L. Deng, L. Qiao, and L. Zhang, "An efficient differential evolution algorithm based on orthogonal learning and elites local search mechanisms for numerical optimization," *Knowledge-Based Systems*, vol. 235, Article ID 107636, 2022.

[29] S. X. Zhang, Y. N. Wen, Y. H. Liu, L. M. Zheng, and S. Y. Zheng, "Differential evolution with domain transform," *IEEE Transactions on Evolutionary Computation*, p. 1, 2022.

[30] F. Yu, X. Fu, H. Li, and G. Dong, "Improved fitness proportionate selection-based genetic algorithm," in *Proceedings of the 2016 3rd International Conference on Mechatronics and Information Technology*, pp. 136–140, Shenzhen, China, January 2016.

[31] M. Gangwar, M. Din, and V. K. Jha, "Comparative study of selection methods in genetic algorithm," *International Journal of Soft Computing and Artificial Intelligence*, vol. 5, no. 1, pp. 37–40, 2017.

[32] R. Abd Rahman, R. Ramli, Z. Jamari, and K. R. Ku-Mahamud, "Evolutionary algorithm with roulette-tournament selection for solving aquaculture diet formulation," *Mathematical Problems in Engineering*, vol. 2016, Article ID 3672758, 10 pages, 2016.

[33] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.

[34] S. Zheng, M. Ge, C. Li, C. Wang, and A. Xue, "New transition probablity for ant colony optimization: global random-proportional rule," in *Proceedings of the 2010 8th World Congress on Intelligent Control and Automation*, pp. 2698–2702, Jinan, China, July 2010.

[35] Z. H. Ahmed, "An improved genetic algorithm using adaptive mutation operator for the quadratic assignment problem," in *Proceedings of the 2015 38th International Conference on Telecommunications and Signal Processing (TSP)*, pp. 1–5, Prague, Czech Republic, July 2015.

[36] A. S. Hameed, M. L. Mutar, H. M. B. Alrikabi, Z. H. Ahmed, A. A. Abdul-Razaq, and H. K. Nasser, "A hybrid method integrating a discrete differential evolution algorithm with tabu search algorithm for the quadratic assignment problem: a new approach for locating hospital departments," *Mathematical Problems in Engineering*, vol. 2021, Article ID 6653056, 21 pages, 2021.

[37] T. Achary, S. Pillay, S. M. Pillai, M. Mqadi, E. Genders, and A. E. Ezugwu, "A performance study of meta-heuristic approaches for quadratic assignment problem," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 17, pp. 1–29, 2021.

[38] M. Chyży and W. Kosiński, "Comparison of crossover operators for the state assignment problem," *Intelligent Information Systems*, vol. 34, no. 2, pp. 195–200, 2002.