Applied Soft Computing 144 (2023) 110479

Contents lists available at ScienceDirect

Applied Soft Computing

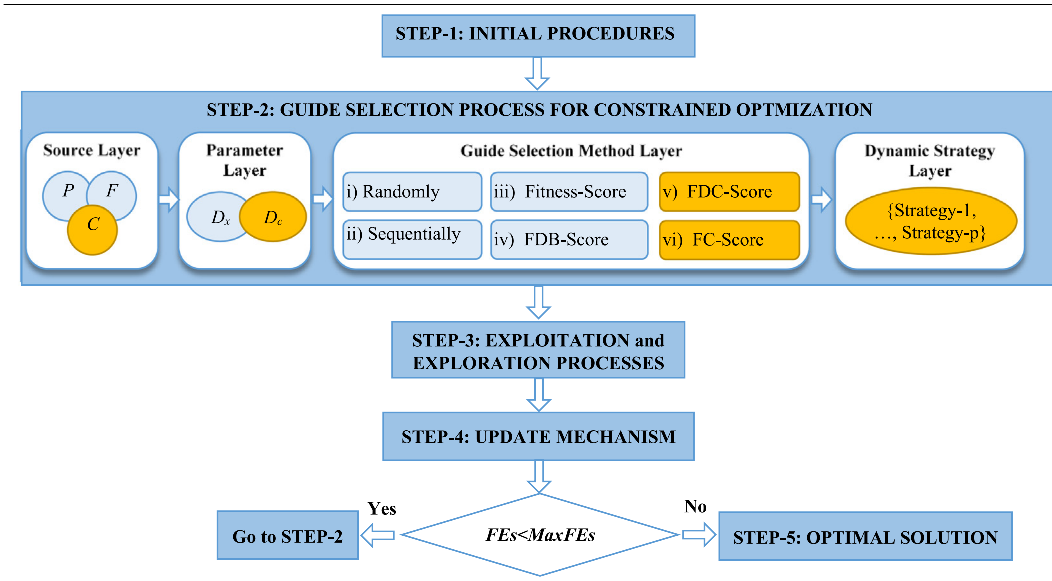
journal homepage: www.elsevier.com/locate/asoc

Fitness-Distance-Constraint (FDC) based guide selection method for constrained optimization problems

Burcin Ozkaya [a], Hamdi Tolga Kahraman [b,*], Serhat Duman [a], Ugur Guvenc [c]

[a] Electrical Engineering, Engineering and Natural Sciences Faculty, Bandirma Onyedi Eylul University, Bandirma, Turkey
[b] Software Engineering Of Technology Faculty, Karadeniz Technical University, Trabzon, Turkey
[c] Electrical and Electronics Engineering, Engineering Faculty, Duzce University, Duzce, Turkey

GRAPHICAL ABSTRACT



ARTICLE INFO

Article history:
Received 1 April 2023
Received in revised form 22 May 2023
Accepted 25 May 2023
Available online 14 June 2023

Dataset link: https://www.mathworks.com/matlabcentral/fileexchange/130244-fitness-distance-constraint-in-constrained-optimization?s_tid=srchtitle_FDC_1

Keywords:
Fitness-Distance-Constraint (FDC)
Meta-heuristic search algorithm design

ABSTRACT

In the optimization of constrained type problems, the main difficulty is the elimination of the constraint violations in the evolutionary search process. Evolutionary algorithms are designed by default according to the requirements of unconstrained and continuous global optimization problems. Since there are no constraint functions in these type of problems, the constraint violations are not considered in the design of the guiding mechanism of evolutionary algorithms. In this study, two new methods were introduced to redesign the evolutionary algorithms in accordance with the requirements of constrained optimization problems. These were (i) constraint space-based, called Fitness-Distance-Constraint (FDC), selection method and (ii) dynamic guiding mechanism. Firstly, thanks to the FDC guide selection method, the constraint violation values of the individuals in the population were converted into score values and the individuals who increase the diversity in the search process were selected as guide. On the other hand, in dynamic guiding mechanism, the FDC method was applied in case of constraint violation, otherwise the default guide selection method was used The proposed methods were used to redesign the guiding mechanism of adaptive guided differential evolution

* Corresponding author.
E-mail addresses: bozkaya@bandirma.edu.tr (B. Ozkaya), htolgakahraman@ktu.edu.tr (H.T. Kahraman), sduman@bandirma.edu.tr (S. Duman), ugurguvenc@duzce.edu.tr (U. Guvenc).

https://doi.org/10.1016/j.asoc.2023.110479
1568-4946/

Optimization
Constrained optimization problem

(AGDE), a current evolutionary algorithm, and the FDC-AGDE algorithm was designed. The performance of the FDC-AGDE was tested on eleven different constrained real-world optimization problems. The results of the FDC-AGDE and AGDE were evaluated using the Friedman and Wilcoxon test methods. According to Wilcoxon pairwise results, the FDC-AGDE showed better performance than the AGDE in nine of the eleven problems and equal performance in two of the eleven problems. Moreover, the proposed algorithm was compared with the competitive and up-to-date MHS algorithms in terms of the results of Friedman test, Wilcoxon test, feasibility rate, and success rate. According to Friedman test results, the first three algorithms were the FDC-AGDE, LSHADE-SPACMA, and AGDE algorithms with the score of 2.69, 4.05, and 4.34, respectively. According to the mean values of the success rates obtained from the eleven problems, the FDC-AGDE, LSHADE-SPACMA, and AGDE algorithms ranked in the first three with the success rates of 67%, 48% and 28%, respectively. Consequently, the FDC-AGDE algorithm showed a superior performance comparing with the competing MHS algorithms. According to the results, it is expected that the proposed methods will be widely used in the constrained optimization problems in the future.

1. Introduction

Many of the real world optimization problems have different linear or nonlinear constraints, which can be considered as constrained optimization problems (COPs). While solving these problems, the feasible solution region decreases and the search process may become more complicated. In the COPs, the solution candidates that meet all the constraints of the problem are called feasible solutions, while the solution candidates that do not meet any of the constraints are called infeasible solutions. The most important problem in solving the COPs is how to deal with infeasible solutions throughout the search-process life cycle. For this reason, it is very difficult to obtain the global optimal solution for the COPs [1–3]. In addition, the optimization algorithms are designed for the necessities of the unconstrained global optimization problems and these algorithms show premature search performance in solving the COPs. For these reasons, the constraint handling (CH) techniques are widely used together with optimization algorithms in the COPs [4,5].

The CH methods are very important, since the performance and success of the optimization algorithms directly depends on them. Many CH methods were presented in the literature such as penalty function method [1,6–9], feasibility rules [10], superiority of the feasible solutions [1,9,11,12], ε-constraint handling method [1,5,9,13,14], repair algorithms [8,15], and stochastic ranking [1,9,16]. Besides of these methods, hybrid CH methods were presented [17–20]. The most important point in the use of optimization algorithms with the CH methods is where the CH method used is involved during the search process. Therefore, in the literature, the studies on the CH methods were examined and a research was conducted on this subject. According to it, we can classify these CH methods presented in the literature in two categories according to where they were used.

(i) *Methods based on constraint violations incorporating into the fitness function*: In the studies included in this classification, the constraint violations are combined with the objective function value depending on which CH method is used. The most common CH method used for constraints handling is the penalty function method, where the penalty coefficient is used in order to control the constraint violation and fitness function. In this method, the infeasible solutions are penalized to obtain the feasible solutions [1]. Different penalty function approaches have been proposed in the literature such as static penalty [7], dynamic penalty [21], adaptive penalty [21], self-adaptive penalty [1,16], and death penalty [22,23]. In these methods, how the constraint violation values are merged to the objective function differs from each other.

(ii) *Methods used in the selection process of the optimization algorithms*: The CH methods are used in the selection process of the optimization algorithm to provide that the better individual is identified in the next iteration generation. Here, the feasibility rules which are specific to the constraint violation method are consider in determining the next generation. A few of the CH methods with their own feasibility rules were the superiority of feasible solution [1], Deb's feasibility rules [9], ε-constraint handling method [12]. Generally, in these methods, the constraint violation and objective function are handled separately [1,14,16]. These methods do not require any parameters in order to handle the constraints and are particularly good at overcoming the inequality constraints. However, it is very difficult to continue the population diversity in these methods, which may cause the premature convergence [24,25]. In stochastic ranking method, the solution candidates in the population are ranked within a probability factor according to the constraint violation value or the objective function value. Thus, this method provides a chance for selecting the infeasible solution that have better objective function value [9].

According to the authors' knowledge, there was no study in the literature about the design of the guiding mechanism of optimization algorithms for COPs. Among the optimization algorithms, Evolutionary Algorithms (EAs) have been commonly applied to solve the COPs with CH methods and have been received much attention in recent years [1,3,24]. However, the guiding mechanisms of EAs are designed based on the requirements of the unconstrained and continuous type global optimization problems. Since there are no constraint functions in this type of problems, the states/distribution of the individuals in the population and the constraint violation of the individuals are not considered in the design of the guiding mechanism of EAs. In this study, two researches were conducted to design the EAs according to the requirements of the COPs: (i) designing a new guide selection method to solve the COPs and (ii) dynamically deciding the guide selection method according to the constraint violation value. In the proposed Fitness-Constrained (FC) and Fitness-Distance Constrained (FDC) methods, the constraint violation values of the individuals in the population are converted into score values. Thus, the individual who increase the diversity throughout the search process are selected as guide. On the other hand, in the dynamic guiding mechanism, as long as the constraint violations continue, the constraint space-based selection method is used for the guide selection. If the constraint violation is eliminated, the other selection methods presented in the literature are used for the guide selection.

A comprehensive experimental study was carried out and the contributions of these paper to the literature are summarized below:

(i) *Fitness-Distance-Constrained (FDC) and Fitness-Constrained (FC) selection methods were introduced*: In order to solve the

COPs, a novel guide selection methods were presented. In these methods, a constraint matrix is defined for the first time in this study by using the constraint violation values of the individuals in the population.

(ii) *A dynamic guide selection mechanism was proposed*: In this mechanism, when the individuals in the population are exposed to the constraint violation, FDC/FC methods are used to select the guide. Otherwise, if the constraint violations are eliminated, any of the other selection methods can be used.

(iii) *FDC-AGDE algorithm was proposed for optimization of constrained problems by applying the methods introduced in (i, ii)*: The proposed methods were applied to the AGDE algorithm and the FDC-AGDE algorithm was introduced to the literature. The performance of the proposed algorithm was tested on 11 constrained optimization problems of the benchmark introduced from CEC 2020 real-world single-objective constrained optimization competition [26]. The data obtained from the experimental studies were evaluated using Wilcoxon and Friedman test methods. The results clearly showed that the performance of the AGDE algorithm has increased significantly with using the proposed methods introduced in (i, ii).

(iv) *The performance of the FDC-AGDE algorithm was compared with the powerful and up-to-date 10 MHS algorithms*: To test and verify the performance of the proposed FDC-AGDE algorithm, 10 MHS algorithms were used to compare it. These MHS algorithms were coati optimization algorithm (COA) [27], fire hawk optimizer (FHO) [28], beluga whale optimization (BWO) [29], dandelion optimizer (DO) [30], golden jackal optimization (GJO) [31], mountain gazelle optimizer (MGO) [32], cooperation search algorithm (CSA) [33], MadDE [34], AGDE [35], and LSHADE-SPACMA [36]. To show the superior performance of the proposed algorithm, the data obtained from the proposed and competing MHS algorithms were analyzed using the most known non-parametric statistical tests methods such as Friedman and Wilcoxon. Accordingly, the proposed FDC-AGDE algorithm yielded a superior performance for solving the COPs compared to other MHS algorithms.

(v) *The contribution and importance of the methods given (i, ii)*: The proposed FDC/FC can be used to enhance the performance of the other MHS algorithms in solving the COPs. Besides, the proposed dynamic guide selection mechanism enables the selection of the guide mechanism according to the constraint violation. Due to this feature, it is a user-independent method and can be applied smoothly during the guide selection process of the optimization algorithms.

After the introduction section, the remaining of the article is planned as follows:

- Section 2 presents the related studies of the constrained optimization problems and constraint handling methods.
- Section 3 consists of four sub-section. The first sub-section explains the mathematical model of COP and the CH method used in this study. The second presents the design of FDC/FC method. Then, the AGDE algorithm was briefly explained in third sub-section. The last sub-section introduces the application of the FDC/FC methods to the AGDE algorithm.
- Section 4 includes three sub-section. The first presents the simulation environment and the parameters of the competing algorithms. The second explains the performance evaluation criteria of the COPs. The third presents the analyzes of the results obtained from the proposed method.
- Section 5 presents the summary and evaluation of the conclusions.

2. Related studies

In the COPs, as well as the objective function of the them, the constraints of the each individual in the population are taken into consideration. Generally, the individuals of the population are divided into two group in terms of constraint violation values: feasible solution and infeasible solution. The individuals satisfying all constraints are called feasible solution, while the individuals which do not satisfy any of the constraints are called infeasible solutions. Therefore, the most important point in the solving COPs is how to deal with infeasible solutions during the search-process life cycle. In other words, the constraint handling is one of the biggest challenges for the solving COPs. The constraints may lead the search-process to only seek a feasible solution. Thus, this may cause in an inability to focus on finding the optimal solution. Another important point is that the MHS algorithms presented in the literature are designed according to the requirements of unconstrained and continuous global optimization problems. For these reasons, in solving the COPs, since it is necessary to direct the search-process to feasible solution region, a CH method must be used with the MHS algorithms [1,2,24].

In the literature, the most widely used CH method is the penalty function method, where a penalty coefficient is applied to the constraint violation values of the individuals, and this value is combined with the objective function [6,11]. The disadvantage of this method is that when multiple constraints are violated, different parameters need to be described by the user in order to control the amount of penalty. Therefore, different penalty method approaches have been proposed, in which the penalty coefficients are user independent [8,24]. These approaches presented in the literature are static penalty method [21], self-adaptive penalty method [1,16], adaptive penalty method [21], and dynamic penalty method [7,9].

The other most used CH method is the superiority of feasible solution. In [37], the authors used the superiority of feasible solution to handle various constraints in the problem. In this method, the aim was to direct the search process towards the global optimal solution using the information of the infeasible solution. There was three rules and the individuals were updated based on these rules in the selection process of the algorithm. The other frequently used CH method was the ε-constraint handling method where a threshold ε value was determined to control the relaxation of the constraints [13]. This method was similar to superiority of the feasible solution method in terms of the selection of the individuals; however, whether an individual was accepted as a feasible solution determined by the ε value. Similar to the superiority of the feasible solution method, it was used to provide that the individuals in the population direct to the feasible solution in the search space. The authors in [1,16] presented an ensemble of CH methods including the self-adaptive penalty, ε-constraint handling method, the superiority of feasible solutions, and stochastic ranking. The authors in [21] used the ensemble CH methods including the repair method, the static penalty method, the rejection method, the dynamic penalty method, and adaptive penalty method to handle the constraints, and compared the performance of them. The authors in [38] introduced a heuristic strategy in order to solve the various constraints of the problem. In order to cope with the complex constraints, an enhanced CH strategy based on the feasibility based rules directing the optimization to the feasible region was proposed. While creating the initial position of each individual, the limit of equality and inequality constraints was determined by the proposed CH method. In addition, the feasibility-based rules were applied to update the positions of the solution candidates.

When the studies related to the CH methods given in the literature were examined, new or hybrid methods were suggested as well as the classical and frequently used methods. In

Ref. [5], the TS-ε and Z-ε constrained methods based on the ε-constraint method were proposed, where the aim was to direct the infeasible solutions to the feasible solution region. In [20], the authors presented a hybrid CH method, where the population was divided into three groups: infeasible, feasible, and semi-feasible. For each group, different CH method was applied and the results of them were used to guide the evolution of the population. In another study, a hyperspace dynamic constraint handling region and collaborative constraint-handling technology (CCHT) methods were proposed and used to create the new population after the selection operator [19]. In [39], the authors proposed the conditional selection strategies performed based on three conditional selection. The proposed CH method was used in the selection process to decide the better individual where the aim was to create a new population as the next generation. In Ref. [40], $\sum$-constrained handling method was proposed. Here, the sum of the constraint violation and objective function used as a combined function are optimized, simultaneously, and is used to choose the better population vector using feasibility rules.

In Table 1, the studies carried out in the literature in recent years about CH methods were given. While creating the table, the CH method proposed/used in the studies and whether or not the constraint violation values were used in the guiding mechanism of the MHS were examined.

When analyzed Table 1 in terms of where the CH methods were used, these methods were generally incorporated to the objective function and used in the selection process of the optimization algorithms. However, there were no study based on the design of guiding mechanism of the optimization algorithms. Since MHS algorithms have been designed for unconstrained optimization problems, the constraint violations are not considered in the design of the guiding mechanism. Therefore, if the guide candidates violate the constraints, they should direct the search process to the infeasible solution region. For these reasons, the constraint violations of the individuals should be considered in the design of guiding mechanism of the MHS algorithms. In addition, various methods were proposed for the design of the guiding mechanism of the MHS algorithms to increase the diversity in the search space for unconstrained optimization problems. In this case, for solving COPs, diversity must be provided not only in the decision space but also in the constraint space. Based on the FDB selection method proposed in Ref. [49], it should be provided that the individuals who were least similar to each other in the decision space, that is individuals who complement each other, were determined and selected as a guide. To summarize, for the solving of the COPs, a generalized method that can be adapted to the guiding mechanism of all MHS algorithms and is taken into account the constraint space is required.

3. Method

In order to make easier the understanding of the FDC selection method and proposed FDC-AGDE algorithm, the subsections were prepared. Thus, the researchers can learn how to apply the FDC method to MHS algorithms to improve their search performance in solving the COPs. This section includes four subsections. In the first sub-section, the mathematical model of COP and the constrained handling method used in the study were presented. An overview of the AGDE algorithm was presented in the second sub-section. Then, the design of FDC based guiding mechanism was explained in detail in the third sub-section. Lastly, the application of the proposed FDC method to the AGDE algorithm was presented in the fourth sub-section.

3.1. The constrained optimization problem and constraint handling method

The majority of real-world optimization problems consist of the nonlinear constraints in the continuous domain. The constrained optimization problems (COPs) aims to maximize or minimize the objective functions. In general, they include the decision variables, the search space, the constrained functions, and the objective functions [50–52]. The generalized mathematical model for the COPs is defined as:

$$\begin{aligned} &\text{Minimize/Maximize} &&: f(X),\ X = (x_1, x_2, \ldots, x_d) \in S, \\ & && lb_i \le x_i \le ub_i \\ &\text{subject to} &&: \begin{cases} g_j(X) \le 0, & j = 1, \ldots, p \\ h_k(X) = 0, & k = p+1, \ldots, m \end{cases} \end{aligned} \tag{1}$$

where S represents the decision space; X represents the d-dimensional solution vector; $f(X)$ is the objective function; $h_k(X)$ and $g_j(X)$ are the kth equality constraint and jth inequality constraint, respectively; p and $(m\text{-}p)$ represent the number of inequality and inequality constraints; lb_i and ub_i are the lower and upper bounds of the x_i, respectively.

In COPs, the equality constraints are converted into inequality constraints through the Eq. (2). Here, ε represents a positive and small tolerance parameter and generally set 0.0001.

$$|h_k(X)| - \varepsilon \le 0, k = p+1, \ldots, m \tag{2}$$

The aim of the optimization problems is to get a global solution for the problem in search space. Since the global solution of the optimization problems in real world problems is not known, design variables that solve the objective function and provide constraints among the solution candidates are considered as the solution to the problem. To find the best solution for the COPs given in Eq. (1), the MHS algorithms can be considered. Assume that the population is represented by P, n is the number of solution candidates in P, d is the number of design variable, and F denotes the fitness values of the solution candidates. In a P-population, ith solution candidate is represented by $X_i = \left[x_{i[1]}, x_{i[2]}, \ldots, x_{i[d]}\right]$. The P-population and the fitness values (F) matrices of the individuals are given in the Eqs. (3) and (4), respectively.

$$P \equiv \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} \equiv \begin{bmatrix} x_{11} & \cdots & x_{1d} \\ \vdots & \ddots & \vdots \\ x_{11} & \cdots & x_{nd} \end{bmatrix}_{nxd} \tag{3}$$

$$F \equiv \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix}_{nx1} \tag{4}$$

A constraint handling strategy is necessary for the optimization algorithms since they are designed for solving the unconstrained optimization problems to solve the COPs. The most commonly used technique for solving the COPs is the penalty method, where main idea is to convert a constrained optimization problem into an unconstrained optimization problem by adding a penalty value to the constraint violation [3,53]. The general form of objective function with penalty method is given as below:

$$\min F(X, \alpha, \beta) = f(X) + \alpha \sum_{j=1}^{p} \max\left\{g_j(X), 0\right\} + \beta \sum_{k=p+1}^{m} |h_k(X)| \tag{5}$$

where α and β are the penalty factors.

3.2. Proposed method: Design of fitness-distance-constraint based guide selection mechanism

Hypothesis: In evolutionary search algorithms, the guiding mechanism should be designed specifically for the requirements of

Table 1
The studies given in the literature related to constraint handling methods.

Year, Ref	Proposed or Used constraint handling method	Method used for guide selection
2023, [5]	*Proposed method*: A combined constraint handling method which consists of TS-ε and Z-ε constrained methods.	• No studies were conducted for the guide selection of the algorithm. • By evaluating the fitness values of possible solution candidates according to the proposed CH method, the combined CH method enabled the solution candidates to direct the optimal solution.
2023, [20]	*Proposed method*: Hybrid constraint handling method.	• No studies were conducted for the guide selection of the algorithm. • In hybrid CH method, this information was used to guide the evolution of the population.
2022, [19]	*Proposed method*: Hyperspace dynamic constraint handling region and collaborative constraint-handling technology (CCHT) including the penalty function method and stochastic ranking.	• No studies were conducted for the guide selection of the algorithm. • The proposed CCHT and HDR were used to create the new population after the selection operator.
2022, [22]	*Used method*: Death penalty function method	• No studies were conducted for the guide selection of the algorithm. • The death penalty function method was used in the objective function.
2022, [14]	*Used method*: ε-constraint handling method	• No studies were conducted for the guide selection of the algorithm. • The ε-constraint handling method was used in the selection process to provide that the solution candidates in the population direct to the feasible solution in the search space.
2021, [3]	*Used method*: Penalty function method	• No studies were conducted for the guide selection of the algorithm. • The constraint violation function was merged with the objective function as a penalty function.
2021, [9]	*Used method*: Static penalty function, stochastic ranking, feasibility rules, ε-constrained handling method, and gradient-based repair.	• No studies were conducted for the guide selection of the algorithm. • The CH methods (i.e. stochastic ranking, feasibility rules, static penalty function, gradient-based repair method, and ε-constrained method) were compared according to the results of the problems.
2020, [13]	*Used method*: ε-constraint handling method	• No studies were conducted for the guide selection of the algorithm. • The ε-constraint handling method was used to update the population in local and global search phases of the algorithm.
2020, [41]	*Proposed method*: Heuristic constraint handling method	• No studies were conducted for the guide selection of the algorithm. • In the study, the equality and inequality constraint violations were evaluated separately by using the proposed heuristic CH method. To handle the constraints, the total constraint violation were combined with the objective value.
2020, [42]	*Used method:* Cumulative constraint handling method.	• No studies were conducted for the guide selection of the algorithm. • The fitness value and constraint value of the individuals in the population were calculated. Then, the population was sorted according to the superiority of the feasible solution method.
2020, [43]	*Used method:* Feasibility rules.	• No studies were conducted for the guide selection of the algorithm. • New mutation strategy was proposed. • The feasibility rules were used in the selection process of the algorithm.
2019, [20]	*Used method*: Penalty function method and repair technique.	• No studies were conducted for the guide selection of the algorithm. • While the inequality constraints were evaluated by the repair method, the equality constraint was evaluated by the penalty function method. Therefore, the total constraint violation was evaluated only in terms of equality constraints and this value merged with the objective function.
2019, [44]	*Proposed method*: A new constraint handling strategy.	• No studies were conducted for the guide selection of the algorithm. • The strategy was developed to satisfy the all constraints which had two parts: Firstly, every solution obtained by the MHS algorithm were repaired. Then, a penalty function was construct based on the objective function and constraint violations values.

(*continued on next page*)

Table 1 (*continued*).

Year, Ref	Proposed or Used constraint handling method	Method used for guide selection
2019, [45]	*Proposed method*: Modified equality constraints handling method.	• No studies were conducted for the guide selection of the algorithm. • The proposed modified equality CH method was used in the objective function.
2019, [46]	*Used method*: ε-constraint handling method.	• No studies were conducted for the guide selection of the algorithm. • When evaluating the objective function for each individual, the constraint violations of them were evaluated. The ε-constraint handling method was used in the selection process of the algorithm.
2019, [21]	*Used method*: Rejection method, repair method, static penalization method, dynamic penalization method, and adaptive penalization method.	• No studies were conducted for the guide selection of the algorithm. • The CH methods (i.e. static penalization method, dynamic penalization method, adaptive penalization method, rejection method, and repair method) were combined with the CSA algorithm and the optimal CH method was determined according to the results.
2018, [47]	*Proposed method*: New cumulative constraints handling method.	• No studies were conducted for the guide selection of the algorithm. • After the initial population was generated, the fitness value and the constraint violation value of the each individual in the population were calculated separately. Then, the population was sorted based on the constraint violation value according to the superiority of the feasible solution method.
2018, [7]	*Used method*: Static penalty approach.	• No studies were conducted for the guide selection of the algorithm. • In static penalty approach method, a penalty function was multiplied with the sum of the constraint violations and was added to the objective function.
2018, [48]	*Used method*: Constraint handling method.	• No studies were conducted for the guide selection of the algorithm. • The constraint handling method, which controls equality and inequality constraint violations according to the problem, was used after the initial population was generated.

constrained optimization problems. For this purpose, the constraint space should also be considered as a parameter in the design of the guiding mechanism. The information that forms the basis of this hypothesis is explained below:
(i) The common requirement for both unconstrained and constrained optimization problems is to find the individual with the best fitness value. On the other hand, the main requirement in the optimization of constrained problems is that the solution does not have any constraint violations. Although this requirement makes the constrained problems different from the unconstrained problems, it is not considered in the design of the guiding mechanism of evolutionary algorithms. However, the main issue in the optimization of constrained problems is the constraint violations rather than the individuals' fitness values. Therefore, the constraint violations of the individuals should be taken into consideration in the guide selection.
(ii) The main challenge for the algorithms in the evolutionary search process is to avoid local solution traps. Especially, the algorithms often exhibit premature convergence in multi-modal type problems. The basis of the premature convergence problem is the greedy guide selection method designed on the fitness value. This elitist-based method of guide selection causes population members to be overly similar to P_{best} (individual with the best fitness in the population) and to each other. This leads to loss of diversity in the population and to premature convergence. Recently, to overcome this problem, the Fitness-Distance Balance (FDB) selection method was proposed. In the FDB selection method, the fitness values of the individuals and the distances of them to P_{best} are taken as reference. The distances of individuals to P_{best} are calculated in the design parameters space. In the algorithms in which the FDB was applied, a sustainable diversity was achieved and the premature convergence problem was significantly eliminated. Although the FDB selection method was successful, it provided the diversity in the design parameters space. This indicates the importance and necessity of the providing diversity in the constraint space for the optimization of constrained problems. For this, the constraint space must be used in the guide selection process.
(iii) Converting the constraint violations of the individuals into penalties and using the fitness values with penalties in the guide selection process is not an appropriate method to provide diversity in the constraint space, since the penalty value is the result of the sum of the violation values that the individual has the constraint functions. The penalty value does not indicate the status of the individual for each of the constraint functions. That is, since the penalty value is a *one*-dimensional vector, it does not represent the *m*-dimensional constraint space defined in Eq. (1).

According to the hypothesis introduced above, the constraint space-based guide selection method proposed in this study is explained below.

The main elements used in the optimization of a constrained problem are the design variables, the constraint functions, and the objective function as introduced in Section 3.1. The constraint functions are not considered in the design of constrained optimization problems, whereas they are the most important elements for constrained optimization problems. In COPs, while the individuals who do not violate the constraint are considered as feasible solutions, the individuals who violate the constraints are considered as infeasible solutions. Since these infeasible solutions may divert the search process to the infeasible region, they need to be directed towards the feasible solution region. In addition to this drawback, the optimization algorithms proposed in the literature do not consider the constraint functions, because they are designed for unconstrained optimization problems. Therefore, in this study, a constraint space-based guide selection method was proposed, where the constraint functions values of the individuals was taken into account. The aim of the constraint space-based guide selection mechanism is to determine the individual that will contribute the most to the search process in a population, taking into account the decision space and the constraint space. In this selection mechanism, the contributions of the individuals to each other in the constraint space are taken into consideration. When considering the constraint violation values of individuals in a population, the individuals with similar constraint violation values cannot contribute to each other to eliminate the constraint violations, because these individuals are genetically similar to each other and cause a local solution trap. As a result, these cause the termination of the search process as premature convergence.

In order to eliminate these drawbacks, two different methods have been proposed based on the constraint space-based guide selection mechanism. These are the Fitness-Distance-Constrained (FDC) and the Fitness Constrained (FC) based selection methods. In the FC method, the fitness values of the individuals and the distance of the constraint values of the individuals are taken into account. On the other hand, in the FDC method, the fitness value of the individuals, the distance of the constraint violations of the individuals, and the distance of the design variables of the individual in the population are considered. FC/FDC score calculation is explained below step by step.

(i) The equality and inequality constraints of each individual are calculated. The constraint matrix (C) representing the constraint values of the individuals in a P-population with p-inequality constraints and (m-p)-equality constraints and consisting of n-individuals is given in Eq. (6). The rows of the C-constraint matrix represent the constraint values of the each individual.

$$C = \begin{bmatrix} C_1 \\ \vdots \\ C_n \end{bmatrix} \equiv \begin{bmatrix} g_{11} & \cdots & g_{1p} & h_{1(p+1)} & \cdots & h_{1m} \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ g_{n1} & \cdots & g_{np} & h_{n(p+1)} & \cdots & h_{nm} \end{bmatrix}_{nxm} \tag{6}$$

In a P-population, the constraint vector of the individual with the best fitness function value is accepted as the best and is denoted by C_{best}.

(ii) The distance of the each individual from the best solution (X_{best}) is calculated. The distance between the ith individual in the P-population and X_{best} is calculated using the Euclidean metric as given in Eq. (7).

$$^{n}_{i=1}\forall x_i, D_{x_i} = \sqrt{\left(x_{i[1]} - x_{best[1]}\right)^2 + \left(x_{i[2]} - x_{best[2]}\right)^2 + \cdots + \left(x_{i[d]} - x_{best[d]}\right)^2} \tag{7}$$

(iii) The distance of the individuals to the best constraint vector (C_{best}) is calculated. The distance between the constraint vector of the ith individual in the P-population and C_{best} is calculated using Euclidean metric by Eq. (8).

$$^{n}_{i=1}\forall C_i, D_{C_i} = \sqrt{\left(C_{i[1]} - C_{best[1]}\right)^2 + \left(C_{i[2]} - C_{best[2]}\right)^2 + \cdots + \left(C_{i[m]} - C_{best[m]}\right)^2} \tag{8}$$

(iv) The D_x-matrix representing the distance of each individual in the P-population from X_{best} and D_c-matrix representing the distance of the constraint value of each individual from C_{best} are given in Eqs. (9) and (10), respectively.

$$D_x \equiv \begin{bmatrix} D_{x,1} \\ \vdots \\ D_{x,n} \end{bmatrix}_{nx1} \tag{9}$$

$$D_c \equiv \begin{bmatrix} D_{c,1} \\ \vdots \\ D_{c,n} \end{bmatrix}_{nx1} \tag{10}$$

(v) The FDC/FC score are calculated. The score calculation for the FDC and FC methods are different from each other as explained below:

Calculation of FC score of individuals: When the FC score is calculated, the F-matrix given in Eq. (4) and the D_c-matrix given in Eq. (10) are taken into consideration. In the score calculation, these matrices are normalized in the range of [0,1] to prevent them dominating each other. *normF* and *normC* represent the normalized fitness values and distance of the constraint vectors of the individuals from the C_{best}, respectively. The FC scores of the individuals are calculated using *normF* and *normC* by using Eq. (11), where the effect of *normF* and *normC* in the score calculation is taken as equal.

$$^{n}_{i=1}\forall x_i, S_{x_i} = normF_i + normC_i \tag{11}$$

Calculation of FDC score of individuals: In the calculation of the FDC score, F-matrix given in Eq. (4), the D_x-matrix given in Eq. (9), and the D_c-matrix given in Eq. (10) are used. Similar to the FC method, the normalization is performed to prevent these three matrix from dominating each other. *normF*, *normD*, and *normC* represent the normalized fitness values, distance of the individuals from the X_{best}, distance of the constraint vectors of the individuals from the C_{best}, respectively. By using *normF*, *normD*, and *normC*, the FDC scores are computed using Eq. (12), where the effect of them in the score calculation is taken as equal.

$$^{n}_{i=1}\forall x_i, S_{x_i} = normF_i + normD_i + normC_i \tag{12}$$

(*vi*) The n-dimensional S-matrix representing the FC/ FDC score values of the individuals in the population can be defined with Eq. (13).

$$S \equiv \begin{bmatrix} s_1 \\ \vdots \\ s_n \end{bmatrix}_{nx1} \tag{13}$$

The steps of the constraint space based guiding mechanism are given in Algorithm-1. In Algorithm-1, while the inputs are the population (P), the fitness values (F) of the individuals in the P-population, and the constraint matrix (C), the output is the FC/FDC score (S). Accordingly, the FC or FDC method was selected in line 10 of Algorithm-1 and the score values of the individuals in the population are calculated according to selected method.

The application of the proposed guide selection method in MHS algorithms was explained in Fig. 1.

In Fig. 1, the background of the elements proposed for the first time in this study was colored with orange. When Fig. 1. was examined, it was seen that the search process in MHS algorithms includes five steps. These steps were described as below, respectively:

***Step-1*:** In this step, P-population and F-fitness matrices were created using Eqs. (3) and (4). After the initial processes, the search-process lifecycle, which consists of three steps in MHS algorithms, was initialized. In these iterative steps, the guides were selected (Step-2), the exploitation and exploration processes were performed (Step-3), and the population was updated (Step-4), respectively.

***Step-2*:** The proposed method for designing the guide selection mechanism was designed as four layers. Thanks to the flexibility provided by the layered architecture, it was possible that the proposed guide selection mechanism can be easily integrated into different MHS algorithms. The layers of the guide selection mechanism were defined as below.

Source Layer: In the proposed method, the FDB/FDC/FC scores of individuals in P-population (guide candidates) were calculated.

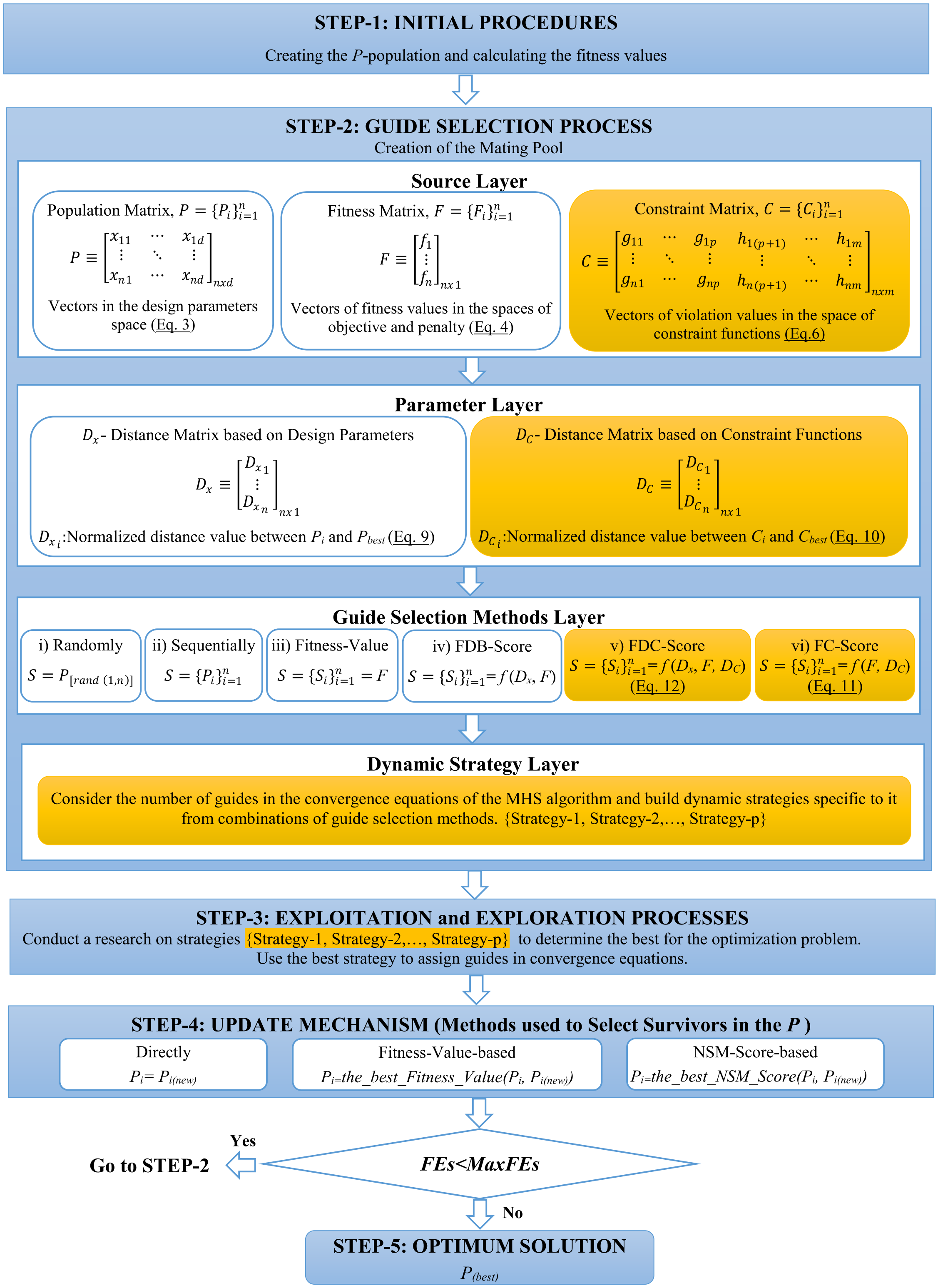


Fig. 1. Integrating the proposed method into the meta-heuristic search process.

Algorithm-1 Steps of Constraint-Space based Guiding Mechanism

Inputs: P (Eq. (3)), F (Eq. (4)), C (Eq. (6))
Output: S (Eq. (13))

```
Begin
Find the X_best and C_best.
    for i =1 : n
        Calculate the D_xi between X_best and X_i using Eq. (7).
        Calculate the D_ci between C_best and C_i using Eq. (8).
    end for
    for i =1 : n
        Normalize the F (Eq. (4)), D_x (Eq. (9), and D_c (Eq. (10)) matrices.
        Select the FC or FDC method
        switch method
            case 'FC'
                if sum (C_i) > ε
                    Calculate the FC score value using Eq. (11).
                else
                    Calculate the score value without constraint.
                end if
            case 'FDC'
                if sum (C_i) > ε
                    Calculate the FDC score value using Eq. (12).
                else
                    Calculate the score value without constraint.
                end if
        end switch
    end for
Select the solution candidates in the search process based on their scores (S) given in Eq. (13).
end
```

The input parameters required to calculate these three scores of the guide candidates were defined in the source layer. For example, F-matrix is needed to implement a fitness-based guiding mechanism (which can be based on greedy or probabilistic choice) in MHS algorithms. Unlike the selection methods in the literature, C-constraint matrix was defined in the source layer for the first time in this study. The C-matrix was used to calculate the FDC and FC scores of the guide candidates. The purpose of using the C-matrix in the design of the guiding mechanism was to convert the distances of the individuals in the constraint space into a score that can be used in the guide selection. The rows of the C-matrix belongs to the constraint values of the each individual.

Parameter Layer: In this layer, D_x and D_c distance matrices were defined. As introduced in the FDB [49], the D_x-distance matrix was calculated based on the design variables of the individual in the population. Thanks to D_x-matrix, the individual who complement each other in the decision space can be selected as guides. D_c-matrix was proposed for the first time in this study for the design of the guiding mechanism. The D_c-matrix was calculated based on the constraint violation values of the individuals in the population. Thanks to D_c-matrix, it was provided that the individuals who complement each other in the constraint space can be found and were selected as guide.

Guide Selection Methods Layer: The guide selection methods were given in this layer. The well known guide selection methods given in the literature were (i, ii, and iii). The (iv)-FDB score based guide selection method was introduced in 2020 and has been used in the design of more than forty algorithms in three years [49]. In the FDB method, two input parameters, D_x and F, are used to calculate the scores of the individuals in P-population, and the individual with the best FDB-score was selected as *Pbest*'s mate. The value of the F-parameter indicates the quality of the individual (mate candidate), while the value of the D_x-parameter represents the individual's contribution to genetic diversity. In the FDB method, the states of the population members in the constraint space are not taken into account. To sum up, the FDB method is a selection method proposed to design of the guide selection mechanism of the MHS algorithms to be used for solving unconstrained optimization problems. However, in this method, the constraint violation values are not considered in the guide selection. So, it is a handicap in terms of applying the FDB method to COPs. This handicap can be eliminated thanks to the proposed selection methods designed based on (v)-FDC and (vi)-FC scores. In these two methods, unlike the FDB method, the distance information in the constraint space was also used in the calculating the scores of the guide candidates. Thus, the individuals with better fitness values compared to their competitors and who contribute more to the diversity of the population in decision and constraint spaces are selected as guides.

Dynamic Strategy Layer: The strategies designed for guide selection were defined in this layer. The strategies were designed based on two situations. These were the number of the guides in the convergence equations of the MHS algorithm and the methods used for guide selection. The most important condition to be considered in the design of strategies was that only one of the FDC or FC methods can be used in the selection of the guide in case of the constraint violation. This particular condition makes the strategies used in the design of the guide selection mechanism dynamic. Accordingly, different method can be used for the selection of each guide in the convergence equations. Different strategies can be designed depending on the number of guides and the methods used in the guide selection. The pseudocode of this dynamic guide selection strategy based on constraint violations was presented in Algorithm-2.

As defined in line 3 of Algorithm-2, if the guide candidate had a constraint violation, only one of the FDC or FC methods can be used for guide selection. Thus, the C-constraint matrix was taken into account in calculating the scores of the candidates who violate the constraint, and the constraint space mate of X_{best} was selected.

Step-3: In this step, the new positions in the search space were searched in MHS algorithms. For this purpose, the guides and

Algorithm-2. Pseudocode for selecting guides using a dynamic strategy

Inputs: *P, GSM (guide selection method)*
Output: *S*

```
while (search-process lifecycle)
    for i=1: n (the number of solution candidates)
        if (P[i] has constraint violation)
            S[i]= f_FDC/FC (P[i]) % use the (v or vi) defined in the guide selection methods layer
        else
            S[i]= f_GSM (P[i]) % use the (i, ii, iii, or iv) defined in the guide selection methods layer
        end if
    end for
end while
```

the convergence equations were used in this step. The guides selected according to the procedure defined in Algorithm-2 in Step-2 were used as a parameter of the convergence equation in the MHS algorithm and the exploitation–exploration operations were performed in the search space.

Step-4: In MHS algorithms, the updating process of the population was performed by the survival mechanism. Although three methods were used in the design of the update mechanism in the literature, the most up-to-date among these three methods is the Natural Survivor Method (NSM). For detailed information on NSM, you can review the Ref. [54].

Step-5: After the search process life cycle was completed in MHS algorithms, the individual with the best fitness value in P was accepted as the optimum solution by using the greedy selection method.

In summary, the proposed guide selection method and the application of the dynamic guiding mechanism to MHS algorithms were explained in this sub-section. In the following sections, the application of the proposed guide selection method and dynamic guiding mechanism to the AGDE algorithm was explained.

3.3. Overview of the AGDE algorithm

The AGDE algorithm is a variation of DE algorithm introduced by Wagdy and Khater to enhance the search capabilities of the DE algorithm [35]. The original DE algorithm includes three main steps, during search-process lifecycle: selection, mutation, and crossover [55,56]. These three basic steps are also performed in the AGDE algorithm; however, the application of the mutation scheme and parameter adaptation scheme are different in the AGDE algorithm.

Mutation strategy in AGDE: In AGDE, a new mutation technique was proposed to enhance the exploration ability of the algorithm. Before examining this new mutation strategy, the mutation strategy used in the original DE algorithm should be examined. In the DE algorithm, the DE/rand/1 mutation strategy is used and the mutant vector v_i^G is generated by using Eq. (14) for the each target vector x_i^G at the generation G [55]. Here, x_{r_1}, x_{r_2}, and x_{r_3} are the randomly selected solution candidates from the population where r_1, r_2, and r_3 are randomly chosen indices.

$$v_i^G = x_{r_1}^G + F * \left(x_{r_2}^G - x_{r_3}^G\right), r_1 \neq r_2 \neq r_3 \neq i \tag{14}$$

The AGDE initializes with a random P-population and each solution candidate is represented by the target vector x_i. In the application of the proposed mutation strategy, the population consists of three groups where the first two mutant vectors are chosen randomly from the top and bottom 100p% individuals from the population. The third vector is chosen randomly from the [*number of population-2*(100p%)]. Here, p is set as 0.1. The proposed mutant vector v_i^{G+1} is generated as:

$$v_i^{G+1} = x_r^G + F * \left(x_{p_{best}}^G - x_{p_{worst}}^G\right) \tag{15}$$

where $x_{p_{best}}$, $x_{p_{worst}}$, and x_r represent the solution candidates randomly chosen from the population the 100p% top, 100p% bottom, and middle [*number of population-2*(100p%)], respectively. Moreover, F is the scaling factor produced independently according to the uniform distribution in the range (0.1, 1). F is an important the parameter that sets the exploration–exploitation balance in the AGDE algorithm.

3.4. Application of the proposed FDC method to the AGDE algorithm

In this sub-section, the implementation of constraint space-based guide mechanisms FDC/FC to the AGDE algorithm was presented. To better understand this section, Sections 3.2 and 3.3 should be examined in detail.

The AGDE algorithm is an evolutionary algorithm, which is an improved version of the DE algorithm. When the AGDE algorithm was applied to solve the COPs, the results demonstrated that the AGDE was insufficient for eliminating the constraint violations of the individuals in the population. Therefore, this situation caused the AGDE to get caught in local solution traps and showed premature convergence performance. In order to overcome these problems of the AGDE, the mutation strategy of it was redesigned by using FDC/FC methods. Since there are no user-defined control parameters in the AGDE, different strategies were developed in the process of adapting the FDC/FC methods to the AGDE. The developed strategies are described below.

Strategy-1*:* This strategy was a classical implementation strategy and the guide candidate obtained from the FDC/FC method was chosen as the mate of X_{best}.

Strategy-2: In this strategy, first of all, the individuals in the population were ranked according to their fitness values. The k-number of elites with the best fitness value in the population were selected. In this study, the k-value was taken as 3. Accordingly, the first three solution candidates with the best fitness value were represented by $X_{\text{best}[1]}$, $X_{\text{best}[2]}$, $X_{\text{best}[3]}$. The mates of these three individuals were the guide individuals which were determined using FDC/FC methods.

Strategy-3: This strategy was a combined strategy, where the Strategy-1 and Strategy-2 were used together. The pseudocode of the variations to be created using Strategy-3 was given in Algorithm-3. In Algorithm-3, *current_iteration*, *number_of_case*, *frequency_epoch* are denoted as the current iteration number, the number of variations to be used when applying Strategy-3, and the frequency of applying the variations. The output *switch_index* represents which variant will be selected. To explain how Strategy-3 was implemented with an example; the

Table 2
Mathematical model of the FDC-AGDE algorithm.

Method	Explanation	Selection of the vectors $x_{r_1}^G$, $x_{r_2}^G$, and $x_{r_3}^G$	Mutation strategy of the cases	
Case-1	The proposed improvement in Eq. (7) was achieved using Strategy-1. Instead of the $x_{r_1}^G$, x_{FDC}^G candidate selected by FDC method was used.	$\{x_{r1}^G \rightarrow x_{FDC}^G$	$v_i^{G+1} = x_{r_3}^G + F * \left(x_{FDC}^G - x_{r_2}^G\right)$	(16)
Case-2	The proposed improvement in Eq. (7) was achieved using Strategy-1. Instead of the $x_{r_3}^G$, x_{FC}^G candidate selected by FC method was used.	$\{x_{r_3}^G \rightarrow x_{FC}^G$	$v_i^{G+1} = x_{FC}^G + F * \left(x_{r_2}^G - x_{r_3}^G\right)$	(17)
Case-3	The proposed improvement in Eq. (7) was achieved using Strategy-2. In Strategy-2, the mating of the $x_{r_1}^G$ was $x_{r_2}^G$ selected by FDC method x_{FDC}^G.	$\{x_{r_2}^G \rightarrow x_{FDC}^G$	$v_i^{G+1} = x_{r_3}^G + F * \left(x_{r_1}^G - x_{FDC}^G\right)$	(18)
Case-4	The proposed improvement in Eq. (7) was achieved using Strategy-2. In Strategy-2, the mating of the $x_{r_1}^G$ was $x_{r_3}^G$ selected by FDC method x_{FDC}^G.	$\{x_{r_3}^G \rightarrow x_{FDC}^G$	$v_i^{G+1} = x_{FDC}^G + F * \left(x_{r_1}^G - x_{r_2}^G\right)$	(19)
Case-5	In this case, the Strategy-3 was used. Two different cases were used and the frequency_epoch was set as 1. The first case was created using Strategy-1 and the second case was created using Strategy-2.	$\begin{cases} (1)\ x_{r_1}^G \rightarrow x_{FC}^G \\ (2)\ x_{r_2}^G \rightarrow x_{FC}^G \end{cases}$	$\begin{cases} (1)\ v_i^{G+1} = x_{r_3}^G + F * \left(x_{FC}^G - x_{r_2}^G\right) \\ (2)\ v_i^{G+1} = x_{r_3}^G + F * \left(x_{r_1}^G - x_{FC}^G\right) \end{cases}$	(20)
Case-6	In this case, the Strategy-3 was used. Two different cases were used and the frequency_epoch was set as 10. The first case was created using Strategy-1 and the second case was created using Strategy-2.	$\begin{cases} (1)\ x_{r_1}^G \rightarrow x_{FC}^G \\ (2)\ x_{r_3}^G \rightarrow x_{FDC}^G \end{cases}$	$\begin{cases} (1)\ v_i^{G+1} = x_{r_3}^G + F * \left(x_{FC}^G - x_{r_2}^G\right) \\ (2)\ v_i^{G+1} = x_{FDC}^G + F * \left(x_{r_1}^G - x_{r_2}^G\right) \end{cases}$	(21)
Case-7	In this case, the Strategy-3 was used. Two different cases were used and the frequency_epoch was set as 1. Both cases were created using Strategy-1.	$\begin{cases} (1)\ x_{r_1}^G \rightarrow x_{FDC}^G \\ (2)\ x_{r_2}^G \rightarrow x_{FC}^G \end{cases}$	$\begin{cases} (1)\ v_i^{G+1} = x_{r_3}^G + F * \left(x_{FDC}^G - x_{r_2}^G\right) \\ (2)\ v_i^{G+1} = x_{r_3}^G + F * \left(x_{r_1}^G - x_{FC}^G\right) \end{cases}$	(22)

Algorithm-3. The pseudocode of the Strategy-3

Inputs : *current_iteration, number_of_case, frequency_epoch, counter*
Output : *switch_index*

```
if mod (current_iteration, frequency_epoch)
    counter = counter + 1;
end if
switch_index = mod (counter, number_of_case)
```

frequency_epoch and *number_of_case* was chosen as 10 and 2, respectively. Accordingly, in the search-process lifecycle, the first case was applied during the first 10th iteration. The other case was applied between 11th and 20th iteration. This cycle continued until the termination criterion was satisfied. In this study, the *number_of_case* value was set as 2 and the *frequency_epoch* value was chosen 1 or 10 according to the cases.

According to the strategies described above, FDC/FC methods were applied to select the $x_{p_{best}}$, $x_{p_{worst}}$, and x_r in the search-process life cycle of the AGDE algorithm. Different variations have been created using FDC/FC methods and strategies (Strategy-1, Strategy-2 and Strategy-3). The top seven methods with the best search performance among the variations were described in this section. The mathematical models of these methods were given in Table 2. It was note that the mutation strategy given in Eq. (15), $x_{p_{best}}$, $x_{p_{worst}}$, and x_r will be called as x_{r_1}, x_{r_2}, and x_{r_3} in the continuation of the study. The pseudocode of the FDC-AGDE algorithm was given in Algorithm-4.

4. Experimental study

4.1. Simulation environment and competing algorithms

In this study, a comprehensive experimental study was carried out to validate the performance of the AGDE variations created using the FDC/FC method for solving constrained optimization problems. The aim of this study was to show the performance of the both the developed FDC/FC method and proposed FDC-AGDE algorithm. Experimental studies were performed for this purpose and the settings are given below:

- The benchmark suite used in this study was selected from Ref. [26]. It has different engineering design problems and the details of these problems are given in Table 3. In Table 3, the name of the problems, the dimension of the problems, the number of equality and inequality constraints, and the feasible objective function value of the problem were given.

Algorithm-4. The pseudocode of the proposed FDC-AGDE algorithm

Inputs: *Problem number*
Output: *Optimal solution* (*global_best*)

for $k = 1$: length (*case_number*) // Case-1, Case-2, Case-3, Case-4, Case-5, Case-6, Case-7 //
  Create a random *P*-population of *n* solution candidates.
  for $i = 1 : n$
    Compute the objective functions and constraint functions using Eq. (1) and (2).
    Create the constraint matrix given as Eq. (8).
  end for
  while FEs < maxFEs
    for $i = 1 : n$
      Generate $F = \text{rand}\ (0.1,1)$
      Compute the crossover rate.
      Select the vectors x_r^G, $x_{p_{best}}^G$, $x_{p_{worst}}^G$ according to *case_number* (Case-1, Case-2, Case-3, Case-4, Case-5, Case-6, Case-7) given in Table 2.
      Specify the number of mutation points ($j_{rand} = \text{randint}\ (1, D)$)
      for $j = 1 : D$
        if $\text{rand}_{j,i}\ [0, 1] < Cr_i$ or $j = j_{rand}$
          According to *case_number*, apply the mutation strategies using Eqns. (16) - (22) (Case-1, Case-2, Case-3, Case-4, Case-5, Case-6, Case-7) given in Table 2.
        else
          $u_{j,i}^G = x_{j,i}^G$
        end if
      end for
      if $\text{fit}\ (u_i^G) \leq \text{fit}\ (x_i^G)$
        $x_i^{G+1} = u_i^G$
        if $\text{fit}\ (u_i^G) \leq \text{fit}\ (x_{best}^G)$
        $x_i^{G+1} = u_i^G$
        end if
      else
        $x_i^{G+1} = x_i^G$
      end if
    end for
  end while
Save the $x_{best,k}^G$.
end for
$x_{best,k}^G = arg\ min_{i=1}^n\ \text{fit}\left(x_i^G\right)$

Table 3
The details of the constrained optimization problems [6].

Problem	Name	*Dimension*	*g*	*h*	$f(x*)$
	Industrial chemical processes				
P1	Heat exchanger network design (case 1)	9	0	8	1.8931162966E+02
P2	Optimal operation of alkylation unit	7	14	0	−4.5291197395E+03
P3	Blending-Pooling-Separation problem	38	0	32	1.8638304088E+00
	Process synthesis and design problems				
P4	Process flow sheeting problem	3	3	0	1.0765430833E+00
P5	Process synthesis problem	7	9	0	2.9248305537E+00
P6	Process design problem	5	3	0	2.6887000000E+04
	Mechanical engineering problems				
P7	Tension/compression spring design (case 1)	3	4	0	1.2665232788E−02
P8	Welded beam design	4	5	0	1.6702177263E+00
P9	Multiple disk clutch brake design problem	5	8	0	2.3524245790E−01
P10	Planetary gear train design optimization problem	9	10	1	5.2576870748E−01
P11	Step-cone pulley problem	5	8	3	1.6069868725E+01

$f(x*)$: Feasible objective function value

Table 4
Parameter settings for all algorithms.

Algorithm	Year	Parameters	Values
COA [27]	2023	Population size (N)	30
FHO [28]	2023	Number of initial candidates	25
BWO [29]	2022	Population size (n) Probability of whale fall (W_f)	50 Decreasing value at interval [0.1 0.05]
DO [30]	2022	Population size Adaptive step size (α) Control parameter (k)	30 $\alpha \in [0, 1]$ $k \in [0, 1]$
GJO [31]	2022	Number of search agents Decreasing energy of the prey (E_1) r : random number c_1 : constant value	30 Linearly reduction from 1.5 to 0 $r \in [0, 1]$ 1.5
MGO [32]	2022	Population size (N)	30
CSA [33]	2021	Population size Learning coefficients α and β Number of elite solutions (M)	50 0.1 and 0.15 3
MadDE [34]	2021	Maximum population size (NP^{max}) Minimum population size (NP^{min}) Archive rate Percentage of population in p-best mutation (p) Probability of qBX crossover (p_{qBX}) Memory size multiplier (H_m) Memory size	2*(dimension)^2 4 2.3 0.18 0.01 10 H_m *dimension
AGDE [35]	2019	Population size (NP) p	50 0.1
LSHADE-SPACMA [36]	2017	Population size (N) Minimum population size (N^{min}) Archive rate Memory size (H) Pbest individuals rate (p) Probability variable (FCP) Learning rate (c)	18*dimension 4 1.4 5 0.11 0.5 0.8

- The termination criteria of the algorithms was set as maximum number of fitness evaluations (*maxFEs*). For each problem, *maxFEs* value was different and it was determined by Eq. (23).

$$maxFEs = \begin{cases} 1 \times 10^5, & if\ D \leq 10 \\ 2 \times 10^5, & if\ 10 < D \leq 30 \\ 4 \times 10^5, & if\ 30 < D \leq 50 \\ 8 \times 10^5, & if\ 50 < D \leq 150 \\ 1 \times 10^6, & if\ 150 < D \end{cases} \tag{23}$$

- In the study, the penalty function method was used as the constraint handling method. In the penalty function method, a penalty coefficient is used and it varies for different COPs. The penalty coefficients of the problems (P1 to P11) given in Table 3 were set as {0.01, 100, 5000, 100, 100, 10 000, 100, 1000, 1000, 1000, 1000}, respectively. To provide the fairness, these penalty coefficients were used in all experimental studies.
- In the experimental studies, 10 competing MHS algorithms were used to validate the performance of the proposed FDC-AGDE algorithm. The parameter settings of all algorithms are given in Table 4.
- The FDC-AGDE and competing MHS algorithms implemented 25 independent runs on each problem of the benchmark suite.

4.2. Performance evaluation criteria

The benchmark suite used in this study consists of different engineering applications. The characteristics and complexity of the problems are different from each other. For this reason, different evaluation criteria were used to determine the degree of difficulty of each problem in the benchmark suite. In this study, the evaluation criteria of the problems were the mean constraint violation, the success rate, the feasibility rate. The definitions of them were given below:

- *Mean Constraint Violation* (MV): The mean value of the constraint violation is calculated as:

$$v(x) = \frac{\sum_{i=1}^{p} G_i(x) + \sum_{j=p+1}^{m} H_j(x)}{m} \tag{24}$$

where $G_i(x)$ and $H_j(x)$ are defined as:

$$G_i(x) = \begin{cases} g_i(x), & if\ g_i(x) > 0 \\ 0, & if\ g_i(x) \leq 0 \end{cases} \tag{25}$$

$$H_j(x) = \begin{cases} \left|h_j(x)\right|, & if\ \left|h_j(x)\right| - 0.0001 > 0 \\ 0, & if\ \left|h_j(x)\right| - 0.0001 \leq 0 \end{cases} \tag{26}$$

- *Feasibility Rate* (FR): It is defined as the ratio of the number of runs within the *maxFEs* and the total runs for which at least one suitable solution is reached.
- *Success Rate* (SR): It is defined as the ratio of the total number of runs an algorithm achieves a feasible solution

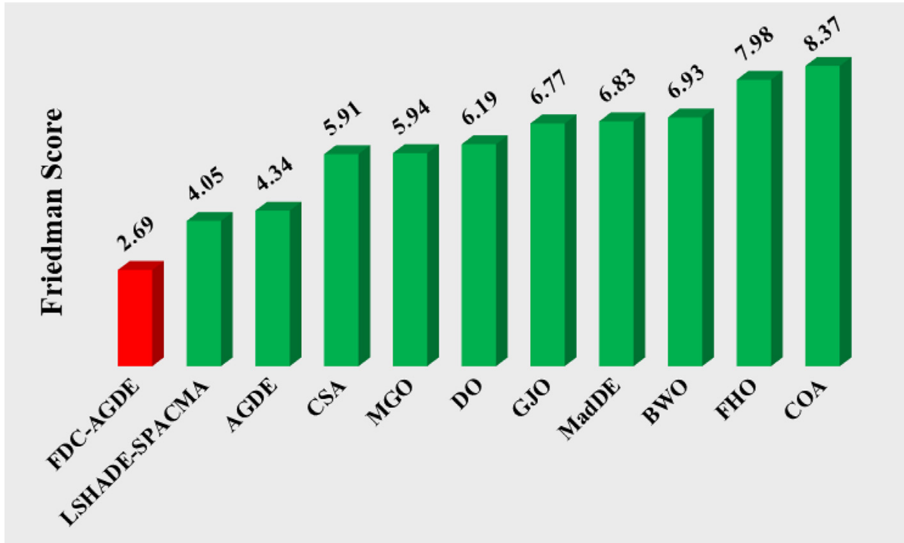


Fig. 2. Friedman scores of the proposed FDC-AGDE and competing MHS algorithms.

that satisfies $f(x) - f(x*) \leq 10^{-8}$ within the *maxFEs* and the total runs.

According to above mentioned criteria, the following criteria are used to assess the difficulty level of problems: (*i*) Firstly, the problems are evaluated based on the SR value, (*ii*) Secondly, the problems are evaluated based on the FR value, (*iii*) the problems are evaluated based on the MV value.

4.3. Simulation and analysis results

4.3.1. Comparison of FDC-AGDE and competing MHS algorithms

In order to show the superiority of the proposed FDC-AGDE algorithm, a comprehensive comparison between proposed FDC-AGDE and competing MHS algorithms were carried out. This section consists of two sub-section. In the first sub-section, to compare the performance of the proposed FDC-AGDE algorithm with competitors, Friedman and Wilcoxon tests were used to analyze the data obtained from the experimental studies. Moreover, the minimum, median, mean, maximum, standard deviation, mean violation, FR and SR values of the algorithms for each problem were given. In the second sub-section, the convergence characteristics of the proposed FDC-AGDE and competing algorithms were examined.

4.3.1.1. Statistical analysis. In the experimental studies, to demonstrate the superiority of the proposed FDC-AGDE algorithm, 11 constrained optimization problems and 10 competing MHS algorithms were used. The algorithms were run 25 times for each problem. According to these, the total number of data used in the statistical analysis method was 3025 (11*11*25). In order to analyze the data, Friedman and Wilcoxon test were applied using error values of the problems. The Friedman test results of the proposed FDC-AGDE and competing MHS algorithms were given in Fig. 2. As it can be seen in Fig. 2, the Friedman score of the FDC-AGDE was 2.69 and it was ranked first among all MHS algorithms. The LSHADE-SPACMA and AGDE algorithms had the second and third best score with 4.05 and 4.34, respectively. Among all algorithms, the COA algorithm had the worst score with 8.37.

The Wilcoxon pairwise comparison results of the proposed FDC-AGDE and competing MHS algorithms were given in Table 5. In Table 5, "+", "=", "−" symbols represent that the proposed FDC-AGDE algorithm was better than, similar to, and worst than corresponding MHS algorithm, respectively. The sum of the scores of the each row was equal to the total number of problem used in the study. According to the results of pairwise comparison between the FDC-AGDE and LSHADE-SPACMA, while the FDC-AGDE was defeated in 2 of 11 problem, it performed better than 5 of 11 problem and the two algorithms achieved similar results in 4 problems. On the other hand, the pairwise comparison between the FDC-AGDE and AGDE show that the FDC-AGDE performed had better performance in 9 of the 11 problem and equal performance in 2 of the 11 problem. Consequently, the proposed FDC-AGDE showed superior performance against the AGDE by not losing in any problem.

Table 5
Wilcoxon pairwise comparison results of the proposed FDC-AGDE and competing MHS algorithms.

FDC-AGDE vs.	+	=	−
LSHADE-SPACMA	5	4	2
AGDE	9	2	0
CSA	7	2	2
MGO	8	2	1
DO	10	1	0
GJO	10	1	0
BWO	9	1	1
MadDE	9	1	1
FHO	10	1	0
COA	10	0	1

The results obtained from the proposed and competing MHS algorithms were evaluated using the performance evaluation criteria given in Section 4.2 and the minimum (min), median, mean, maximum (max), standard deviation (std), MV, FR, and SR values of them for each problem were given in Table 6. The results of the algorithm that was the best among its competitors were marked in bold. The assessment of the results were explained below:

- For P1, the FDC-AGDE algorithm had the best SR value with 84% compared to its competitors. The MadDE algorithm had the second best SR value with 32%.
- For P2, SR value was 0 in all algorithms. For this reason, FR value was taken into account in the comparison. Accordingly, while the LSHADE-SPACMA algorithm had the best FR value of 80%, the second algorithm was CSA with 28%. The proposed FDC-AGDE algorithm was ranked third with 20% FR value.

Table 6
Results of the FDC-AGDE and competing MHS algorithms.

Problem	Algorithm	Min	Median	Mean	Max	Std	MV	FR	SR
	FDC-AGDE	**1.8931E+02**	**1.8931E+02**	**1.8931E+02**	**1.8931E+02**	**2.5369E−06**	**0.0000E+00**	**100**	**84**
	AGDE	1.8931E+02	2.0042E+02	2.0806E+02	2.6765E+02	2.2656E+01	4.3517E+00	28	0
	LSHADE-SPACMA	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
	CSA	1.2639E+02	4.1486E+02	3.9563E+02	7.9750E+02	1.4633E+02	2.3459E+04	0	0
	MGO	2.5440E+00	2.6452E+02	3.1743E+02	8.1786E+02	1.9293E+02	2.0471E+04	40	0
P1	DO	1.9030E+02	3.0439E+02	3.8591E+02	8.4680E+02	2.0910E+02	9.7139E+01	20	0
	GJO	0.0000E+00	3.4667E−12	1.9806E+02	7.7549E+02	2.4984E+02	1.3008E+05	0	0
	BWO	1.8933E+02	2.7934E+02	3.4781E+02	8.4079E+02	2.0713E+02	1.8337E+03	0	0
	MadDE	1.7542E+01	1.2942E+02	1.2778E+02	1.8931E+02	5.7748E+01	0.0000E+00	100	32
	FHO	0.0000E+00	3.1246E+02	3.8666E+02	9.5262E+02	2.8763E+02	1.8294E+05	0	0
	COA	0.0000E+00	2.9445E+02	3.0706E+02	9.8002E+02	3.0491E+02	2.1742E+05	0	0
	FDC-AGDE	−1.8550E+04	−1.7464E+04	−1.4010E+04	−8.8916E+02	6.7898E+03	8.7959E+00	20	0
	AGDE	−1.9531E+04	−1.8538E+04	−1.7759E+04	−2.3762E+03	3.2952E+03	1.0884E+01	0	0
	LSHADE-SPACMA	**−1.9550E+04**	**−1.9550E+04**	**−1.9430E+04**	**−1.8550E+04**	**3.3166E+02**	**8.1162E−01**	**80**	**0**
	CSA	−1.8879E+04	−1.8550E+04	−1.2899E+04	7.2257E+02	8.4908E+03	6.2995E+00	28	0
	MGO	−1.8550E+04	−1.8550E+04	−1.8550E+04	−1.8550E+04	5.8942E−12	7.7263E+00	0	0
P2	DO	−1.8550E+04	−1.8550E+04	−1.8548E+04	−1.8539E+04	2.4608E+00	7.7256E+00	0	0
	GJO	−1.8549E+04	−1.8548E+04	−1.8538E+04	−1.8347E+04	4.0228E+01	7.7204E+00	0	0
	BWO	−1.8461E+04	−1.7354E+04	−1.3810E+04	−6.6752E+03	5.2220E+03	5.1235E+00	4	0
	MadDE	−1.8550E+04	−1.8550E+04	−1.7615E+04	−6.9042E+03	3.1547E+03	7.1934E+00	0	0
	FHO	−1.8539E+04	−1.8458E+04	−1.4582E+04	−4.5535E+03	6.3691E+03	9.6944E+00	0	0
	COA	−1.2112E+04	−9.8277E−02	−5.4367E+02	−9.8276E−02	2.4281E+03	7.2255E+00	0	0
	FDC-AGDE	**1.7152E+00**	**2.2125E+00**	**2.1678E+00**	**2.5684E+00**	**2.5349E−01**	**0.0000E+00**	**100**	**24**
	AGDE	1.5533E+00	2.1853E+00	2.1371E+00	2.4463E+00	1.8903E−01	0.0000E+00	100	4
	LSHADE-SPACMA	1.4624E+00	2.0353E+00	1.9775E+00	2.3542E+00	2.1396E−01	0.0000E+00	100	0
	CSA	1.2513E+00	2.2335E+00	2.1824E+00	2.5560E+00	3.0021E−01	0.0000E+00	100	12
	MGO	9.9790E−01	1.6044E+00	1.3504E+00	1.6766E+00	3.0117E−01	1.1717E+00	48	0
P3	DO	9.9831E−01	1.2672E+00	1.4346E+00	2.3594E+00	4.2718E−01	1.7335E+00	16	0
	GJO	9.9790E−01	9.9790E−01	1.0895E+00	2.3632E+00	2.9347E−01	2.5906E+00	0	0
	BWO	9.9790E−01	9.9790E−01	9.9928E−01	1.0194E+00	4.5985E−03	4.0848E+00	0	0
	MadDE	1.2542E+00	1.7429E+00	1.6830E+00	1.9705E+00	2.0256E−01	1.0028E+00	44	0
	FHO	9.9790E−01	1.4564E+00	1.8257E+00	2.7520E+00	7.0777E−01	3.5750E+00	0	0
	COA	9.9790E−01	1.0088E+00	1.2311E+00	1.6976E+00	2.6811E−01	1.2259E+01	0	0
	FDC-AGDE	1.0765E+00	1.0765E+00	1.0768E+00	1.0783E+00	6.0304E−04	0.0000E+00	100	80
	AGDE	1.0765E+00	1.0765E+00	1.0846E+00	1.2500E+00	3.4526E−02	0.0000E+00	100	60
	LSHADE-SPACMA	1.0000E−01	1.0765E+00	1.0079E+00	1.2500E+00	3.5082E−01	0.0000E+00	100	60
	CSA	**1.0765E+00**	**1.0765E+00**	**1.0765E+00**	**1.0765E+00**	**4.5325E−16**	**0.0000E+00**	**100**	**100**
	MGO	1.0765E+00	1.2500E+00	1.1945E+00	1.2500E+00	8.2582E−02	0.0000E+00	100	32
P4	DO	1.0765E+00	1.2500E+00	1.2084E+00	1.2500E+00	7.5608E−02	0.0000E+00	100	12
	GJO	1.0767E+00	1.2500E+00	1.1980E+00	1.2500E+00	7.8838E−02	0.0000E+00	100	0
	BWO	1.2500E+00	1.2500E+00	1.2500E+00	1.2500E+00	0.0000E+00	0.0000E+00	100	0
	MadDE	1.0007E−01	4.6353E−01	5.3094E−01	1.2500E+00	3.7946E−01	0.0000E+00	100	0
	FHO	1.0880E+00	1.2500E+00	1.2392E+00	1.2500E+00	3.6763E−02	0.0000E+00	100	0
	COA	1.1394E+00	1.2500E+00	1.2456E+00	1.2500E+00	2.2121E−02	0.0000E+00	100	0
	FDC-AGDE	**2.9248E+00**	**2.9248E+00**	**2.9248E+00**	**2.9248E+00**	**4.5325E−16**	**0.0000E+00**	**100**	**100**
	AGDE	2.9248E+00	2.9249E+00	2.9250E+00	2.9256E+00	1.6980E−04	0.0000E+00	100	0
	LSHADE-SPACMA	2.9248E+00	2.9248E+00	2.9257E+00	2.9470E+00	4.4261E−03	0.0000E+00	100	96
	CSA	2.9254E+00	2.9470E+00	3.0546E+00	4.2094E+00	3.2258E−01	0.0000E+00	100	0
	MGO	2.9249E+00	2.9470E+00	2.9795E+00	3.0817E+00	6.5659E−02	0.0000E+00	100	0
P5	DO	2.9248E+00	2.9470E+00	3.5069E+00	4.6328E+00	7.3041E−01	0.0000E+00	100	0
	GJO	2.9250E+00	4.5894E+00	4.6736E+00	1.1518E+01	1.7847E+00	0.0000E+00	100	0
	BWO	2.9258E+00	2.9305E+00	2.9342E+00	2.9661E+00	1.1276E−02	0.0000E+00	100	0
	MadDE	9.2483E−01	1.9248E+00	2.0952E+00	2.9248E+00	6.1493E−01	0.0000E+00	100	24
	FHO	2.9263E+00	2.9331E+00	3.1402E+00	6.0617E+00	7.0850E−01	0.0000E+00	100	0
	COA	4.6506E+00	6.4914E+00	6.1682E+00	6.8307E+00	5.9293E−01	0.0000E+00	100	0
	FDC-AGDE	2.6887E+04	2.6887E+04	2.6887E+04	2.6887E+04	1.1138E−11	0.0000E+00	100	12
	AGDE	2.6882E+04	2.6891E+04	2.6895E+04	2.6954E+04	1.5013E+01	0.0000E+00	100	0
	LSHADE-SPACMA	**2.2303E+04**	**2.6887E+04**	**2.6226E+04**	**2.6887E+04**	**1.5796E+03**	**0.0000E+00**	**100**	**16**
	CSA	2.6887E+04	2.6887E+04	2.6887E+04	2.6887E+04	1.1139E−11	0.0000E+00	100	0
	MGO	2.6887E+04	2.6887E+04	2.6887E+04	2.6887E+04	1.1139E−11	0.0000E+00	100	0
P6	DO	2.6887E+04	2.6887E+04	2.6887E+04	2.6887E+04	2.3643E−05	0.0000E+00	100	0
	GJO	2.6887E+04	2.6888E+04	2.6888E+04	2.6888E+04	2.1838E−01	0.0000E+00	100	0
	BWO	2.6886E+04	2.6887E+04	2.6888E+04	2.6891E+04	9.2772E−01	0.0000E+00	100	12
	MadDE	2.6813E+04	2.6813E+04	2.6828E+04	2.6887E+04	3.0578E+01	0.0000E+00	100	0
	FHO	2.7138E+04	2.9253E+04	2.9317E+04	3.1305E+04	1.2381E+03	0.0000E+00	100	0
	COA	2.6887E+04	2.6888E+04	2.7740E+04	3.1238E+04	1.4128E+03	0.0000E+00	100	0

(*continued on next page*)

Table 6 (*continued*).

Problem	Algorithm	Min	Median	Mean	Max	Std	MV	FR	SR
P7	FDC-AGDE	**1.2665E−02**	**1.2665E−02**	**1.2665E−02**	**1.2665E−02**	**8.6494E−09**	**0.0000E+00**	**100**	**100**
	AGDE	1.2665E−02	1.2676E−02	1.2679E−02	1.2754E−02	1.8771E−05	0.0000E+00	100	4
	LSHADE-SPACMA	1.2665E−02	1.2666E−02	1.2668E−02	1.2680E−02	4.2837E−06	0.0000E+00	100	20
	CSA	1.2718E−02	1.3916E−02	1.4287E−02	1.7706E−02	1.4763E−03	0.0000E+00	100	0
	MGO	1.2670E−02	1.2719E−02	1.3018E−02	1.6007E−02	8.3600E−04	0.0000E+00	100	0
	DO	1.2685E−02	1.2719E−02	1.2893E−02	1.3674E−02	2.6680E−04	0.0000E+00	100	0
	GJO	1.2668E−02	1.2722E−02	1.2712E−02	1.2730E−02	1.9560E−05	0.0000E+00	100	0
	BWO	1.2830E−02	1.2982E−02	1.3019E−02	1.3506E−02	1.5416E−04	0.0000E+00	100	0
	MadDE	4.1530E−03	1.2665E−02	9.7185E−03	1.2672E−02	3.2984E−03	0.0000E+00	100	52
	FHO	1.2684E−02	1.2999E−02	1.2988E−02	1.3237E−02	1.0933E−04	0.0000E+00	100	0
	COA	6.5316E−03	1.3187E−02	1.3108E−02	1.8763E−02	2.3986E−03	0.0000E+00	100	16
P8	FDC-AGDE	**1.6702E+00**	**1.6702E+00**	**1.6702E+00**	**1.6708E+00**	**1.1244E−04**	**0.0000E+00**	**100**	**100**
	AGDE	1.6702E+00	1.6702E+00	1.6702E+00	1.6702E+00	2.3179E−07	0.0000E+00	100	84
	LSHADE-SPACMA	**1.6702E+00**	**1.6702E+00**	**1.6702E+00**	**1.6702E+00**	**0.0000E+00**	**0.0000E+00**	**100**	**100**
	CSA	1.6703E+00	1.6759E+00	1.7624E+00	3.0534E+00	2.9745E−01	0.0000E+00	100	0
	MGO	1.6714E+00	1.7004E+00	1.7052E+00	1.8134E+00	3.4274E−02	0.0000E+00	100	0
	DO	1.6702E+00	1.6706E+00	1.6714E+00	1.6805E+00	2.2752E−03	0.0000E+00	100	0
	GJO	1.6706E+00	1.6715E+00	1.6721E+00	1.6762E+00	1.3783E−03	0.0000E+00	100	0
	BWO	1.7233E+00	1.8503E+00	1.8425E+00	1.9979E+00	5.3949E−02	0.0000E+00	100	0
	MadDE	3.9067E−02	2.7497E−01	2.6960E−01	5.1487E−01	1.5942E−01	0.0000E+00	100	0
	FHO	2.1712E+00	2.2190E+00	2.2220E+00	2.2664E+00	2.0369E−02	0.0000E+00	100	0
	COA	1.9759E+00	2.2196E+00	2.2038E+00	2.3848E+00	1.1304E−01	0.0000E+00	100	0
P9	FDC-AGDE	**2.3524E−01**	**2.3524E−01**	**2.3524E−01**	**2.3524E−01**	**1.1563E−08**	**0.0000E+00**	**100**	**100**
	AGDE	2.3524E−01	2.3524E−01	2.3524E−01	2.3524E−01	3.2073E−07	0.0000E+00	100	72
	LSHADE-SPACMA	**2.3524E−01**	**2.3524E−01**	**2.3524E−01**	**2.3524E−01**	**1.1331E−16**	**0.0000E+00**	**100**	**100**
	CSA	2.3524E−01	2.3524E−01	2.3524E−01	2.3524E−01	7.4258E−08	0.0000E+00	100	92
	MGO	**2.3524E−01**	**2.3524E−01**	**2.3524E−01**	**2.3524E−01**	**1.1331E−16**	**0.0000E+00**	**100**	**100**
	DO	**2.3524E−01**	**2.3524E−01**	**2.3524E−01**	**2.3524E−01**	**8.1166E−09**	**0.0000E+00**	**100**	**100**
	GJO	2.3524E−01	2.3526E−01	2.3526E−01	2.3528E−01	1.0560E−05	0.0000E+00	100	0
	BWO	2.3525E−01	2.3533E−01	2.3537E−01	2.3574E−01	1.2355E−04	0.0000E+00	100	0
	MadDE	**2.3524E−01**	**2.3524E−01**	**2.3524E−01**	**2.3524E−01**	**4.1061E−14**	**0.0000E+00**	**100**	**100**
	FHO	2.3651E−01	2.4904E−01	2.4993E−01	2.6038E−01	6.3117E−03	0.0000E+00	100	0
	COA	2.3525E−01	2.6712E−01	2.8267E−01	3.3081E−01	4.5276E−02	0.0000E+00	100	0
P10	FDC-AGDE	**5.2559E−01**	**5.2636E−01**	**5.2648E−01**	**5.2766E−01**	**7.2873E−04**	**0.0000E+00**	**100**	**40**
	AGDE	5.2577E−01	5.2735E−01	5.2812E−01	5.3000E−01	1.5923E−03	0.0000E+00	100	4
	LSHADE-SPACMA	5.2577E−01	5.2628E−01	5.2700E−01	5.3706E−01	3.0472E−03	0.0000E+00	100	36
	CSA	5.2700E−01	5.4385E−01	6.1686E−01	9.8368E−01	1.3477E−01	0.0000E+00	100	0
	MGO	5.2325E−01	5.3000E−01	5.3085E−01	5.3706E−01	4.6289E−03	0.0000E+00	100	0
	DO	5.2325E−01	5.3706E−01	5.3514E−01	5.4848E−01	6.0690E−03	0.0000E+00	100	4
	GJO	5.2577E−01	5.3706E−01	5.3504E−01	5.4971E−01	6.1277E−03	0.0000E+00	100	4
	BWO	5.3706E−01	5.4159E−01	5.6577E−01	6.7522E−01	4.2068E−02	0.0000E+00	100	0
	MadDE	5.2577E−01	5.3319E−01	5.3270E−01	5.3706E−01	3.6329E−03	0.0000E+00	100	4
	FHO	5.2577E−01	5.2735E−01	7.0935E−01	1.1344E+00	2.3381E−01	0.0000E+00	100	4
	COA	5.3706E−01	9.2958E−01	8.8141E−01	1.1276E+00	1.4886E−01	0.0000E+00	100	0
P11	FDC-AGDE	**1.6070E+01**	**1.6070E+01**	**1.6070E+01**	**1.6070E+01**	**2.3387E−14**	**0.0000E+00**	**100**	**100**
	AGDE	1.6070E+01	1.6070E+01	1.6070E+01	1.6070E+01	2.4834E−07	0.0000E+00	100	76
	LSHADE-SPACMA	**1.6070E+01**	**1.6070E+01**	**1.6070E+01**	**1.6070E+01**	**3.6260E−15**	**0.0000E+00**	**100**	**100**
	CSA	1.5137E+01	1.6525E+01	1.6427E+01	1.7115E+01	5.3700E−01	0.0000E+00	100	20
	MGO	1.6252E+01	1.6892E+01	1.6862E+01	1.7083E+01	2.0838E−01	0.0000E+00	100	0
	DO	1.6070E+01	1.6822E+01	1.6716E+01	1.7071E+01	2.7492E−01	0.0000E+00	100	0
	GJO	1.6160E+01	1.6638E+01	1.6674E+01	1.7964E+01	4.0266E−01	0.0000E+00	100	0
	BWO	1.6059E+01	1.6496E+01	1.6621E+01	1.7377E+01	3.1650E−01	0.0000E+00	100	4
	MadDE	6.2689E−04	1.0677E−01	1.6165E−01	6.4129E−01	1.8151E−01	0.0000E+00	100	0
	FHO	1.6725E+01	2.0902E+01	2.0127E+01	2.3288E+01	2.0357E+00	0.0000E+00	100	0
	COA	1.5714E+01	1.9096E+01	1.8957E+01	2.3820E+01	1.7651E+00	0.0000E+00	100	4

- For P3, the FDC-AGDE algorithm was ranked first with 24% SR value among all algorithms.
- For P4, the CSA algorithm had the best SR value with 100%. The proposed algorithm achieved the second best SR value after the CSA algorithm with 80%.
- For P5, the FDC-AGDE algorithm achieved 100% success by obtaining the feasible solution. The LSHADE-SPACMA algorithm was ranked second with 96% SR value.
- For P6, while the LSHADE-SPACMA algorithm was ranked first with 16% SR value, the proposed FDC-AGDE and BWO algorithms had the same SR value with 12% and were ranked second.
- For P7, the FDC-AGDE algorithm had gained a serious advantage over its competitors with its 100% SR value. Its closest competitor was the MadDE algorithm achieved 52% success.
- For P8, both the FDC-AGDE and LSHADE-SPACMA algorithms yielded 100% success. After these algorithms, the AGDE algorithm achieved 84% success and took the second place.
- For P9, the FDC-AGDE, LSHADE-SPACMA, MGO, DO, and MadDE algorithms had equal success rates with 100%.
- For P10, the proposed algorithm achieved 40% success and ranked first among its competitors.
- For P11, the FDC-AGDE and LSHADE-SPACMA algorithms had equal SR value with 100%. After these algorithms, the AGDE algorithm achieved 76% success.

To summarize the results presented in Table 6, it was shown that the FDC-AGDE algorithm was the first by itself in 5 of 11 problems. It was shared the first with the competitor algorithm in 3 of 11 problems and ranked second or third after its competitor in 3 of 11 problems. These results clearly demonstrated

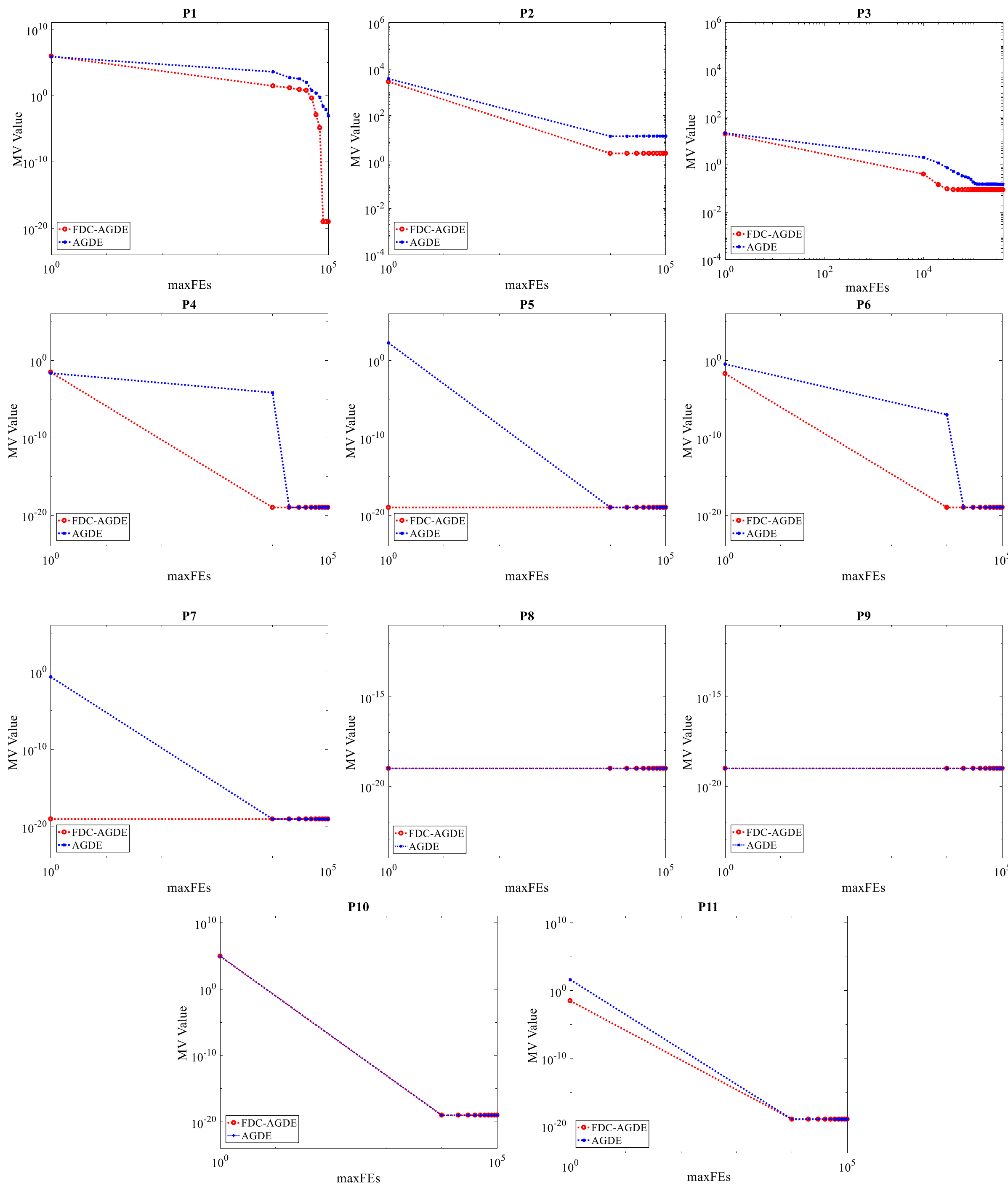


Fig. 3. Convergence curves of the FDC-AGDE and AGDE algorithms.

that the performance of the proposed algorithm outperformed its competitors.

4.3.1.2. Convergence analysis. The convergence performance of the proposed FDC-AGDE and original AGDE algorithm was evaluated for all problems in this section. The convergence curves presented in Fig. 3 were drawn based on the MV value corresponding to the best fitness value of the algorithm for all problems. For P1, the FDC-AGDE algorithm succeeded in converging up to 10^{-19} and it had better convergence performance compared to the AGDE algorithm. For P2, it was seen that the FDC-AGDE algorithm converged faster in eliminating the constraint violation than the AGDE algorithm. For P3, the FDC-AGDE algorithm converged stably with lower error value. For P4 and P6, while both FDC-AGDE and AGDE algorithm converged to 10^{-19}, the proposed FDC-AGDE eliminated the constraint violation in

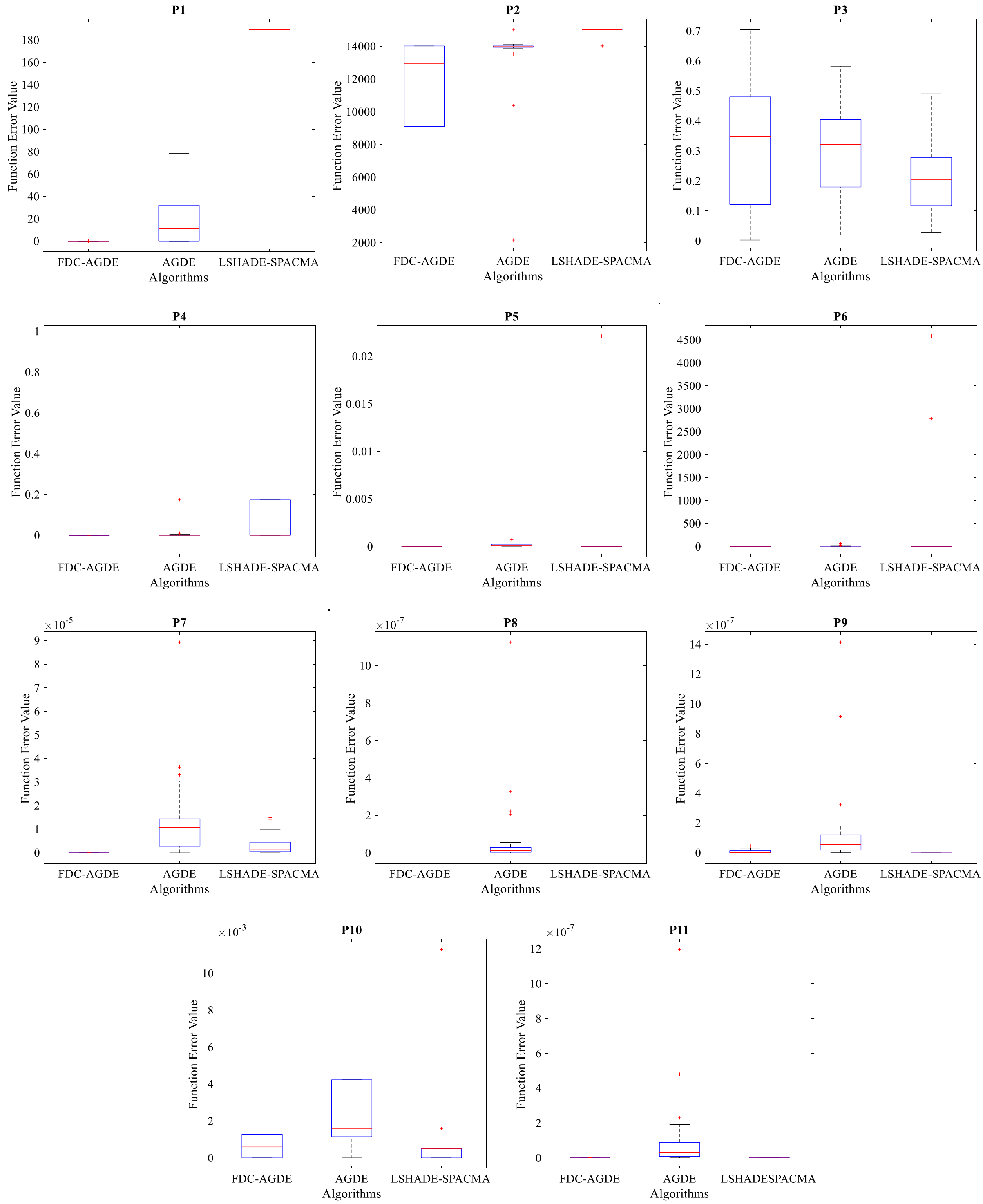


Fig. 4. The box-plots of the FDC-AGDE, LSHADE-SPACMA, and AGDE algorithms.

a shorter time and more stable. For P5 and P7, the FDC-AGDE algorithm achieved approximately 0 during the search process life cycle. Besides, the AGDE converged from 10^2 to approximately 0. For P8 and P9, the both FDC-AGDE and AGDE algorithms converged to 10^{-19} throughout the search process life cycle. For P10, both algorithms exhibited close convergence performance and converged to 10^{-19}. For P11, the FDC-AGDE algorithm converged faster than the AGDE algorithm. In summary, when the convergence graphs given in Fig. 3 were examined, it was seen that the proposed FDC-AGDE algorithm was superior in eliminating the constraint violations compared to the AGDE algorithm.

In order to examine the convergence performance of the algorithm in more detail, Fig. 4 was prepared. The box-plot charts were drawn using the error value of the problems in 25 independent runs of the top three algorithms according to the Friedman score given in Fig. 2. The problems used in this study have different dimensions and different number of equality and inequality constraints. Therefore, the performance of the proposed FDC-AGDE algorithm and competing algorithms (AGDE and LSHADE-SPACMA) may vary for each problem.

As can be seen from Fig. 4, the proposed FDC-AGDE algorithm exhibited stable search performance in the problems P1, P4, P5, P7, and P10 in converging steadily to the optimal solution. However, for P2, none of the algorithms could converge to the optimal solution. For P3 and P6, all algorithms performed similar performance in order to obtain the minimum error values. In the other problems P8, P9, and P11, the FDC-AGDE and LSHADE-SPACMA algorithm showed similar search performance and converged successfully to optimal solution with lower error value.

To sum up, when the performance of the FDC-AGDE algorithm was compared with the AGDE algorithm, it was shown that it converged steadily to optimum solutions with lower error values. That is, the exploration capability of the AGDE algorithm was improved using the FDC method.

5. Conclusion

In this article, a comprehensive research was conducted on the design of MHS algorithms used in the solving COPs, and the following conclusions have been reached as a result of these researches:

(i) *A new guide selection method based on the distances in the constraint space was proposed:* The Fitness-Distance-Constraint (FDC/FC) method, introduced in this article, has been successfully presented to the literature as the first guide selection method developed based on the distances in the constraint space.

(ii) *Dynamic guiding mechanism was proposed*: The dynamic guiding mechanism introduced in this paper was the first proposed method for establishing the mating pool in EAs. Thanks to this mechanism, the method used in the guide selection in EAs can be dynamically determined depending on the constraint violation. Thus, when the constraint was violated, the FDC/FC guide selection method was used. Otherwise, the default guide selection method was used automatically.

(iii) *Feasibility Rate (FR) and Success Rate (SR) values have been improved and the constraint violations can be handled in shorter time:* Using the methods mentioned in (i) and (ii), the guiding mechanism of the AGDE, a current evolutionary algorithm, has been redesigned. The data obtained from experimental studies were analyzed using the statistical test methods. The analysis results clearly showed that the FDC-AGDE algorithm yielded superior and competitive performance over the classic AGDE algorithm. According to Wilcoxon test results, the FDC-AGDE was able to obtain better solutions than its competitor in nine of eleven problems. For the other two problems, both algorithms had equal performance for obtaining the optimal solutions. As can be clearly seen from the convergence curves of the mean constraint violations in Fig. 3, the FDC-AGDE algorithm eliminates the constraint violations in a shorter time thanks to the proposed guiding mechanism compared to the AGDE algorithm

(iv) *A competitive evolutionary algorithm was proposed that can be used in the constrained optimization problems:* According to the Friedman test results given in the Section 4.3.1.1 among the eleven competing algorithms, when the FDC-AGDE algorithm was ranked first with the score of 2.69, the LSHADE-SPACMA algorithm was ranked second with the score of 4.05. According to the Wilcoxon pairwise comparison results between these two algorithms, while the FDC-AGDE found better solutions than its competitor in 5 of 11 problems, the LSHADE-SPACMA outperformed its competitor in 2 of 11 problems, and the competitors performed similar performance in 4 of 11 problems. When the performance of the algorithms were compared according to the mean values of the SR obtained from the eleven problems, the FDC-AGDE, LSHADESPACMA, and AGDE algorithms were ranked in the first three with the SR rates of 67%, 48% and 28%, respectively.

In summary, FDC-based dynamic guiding mechanism has been successfully introduced so that the EAs can be designed specifically for the COPs. The main difficulty in applying the proposed method is the search for the most appropriate guide selection strategy for the convergence equations of the algorithm. Depending on the number of convergence equations and the number of guides in these equations, this research phase may take longer. In future studies, it is planned to apply the FDC-based dynamic guiding mechanism in different EAs and to search for the most effective algorithm.

CRediT authorship contribution statement

Burcin Ozkaya: Methodology, Validation, Writing – original draft, Writing – review & editing, Software, Investigation, Resources. **Hamdi Tolga Kahraman:** Conceptualization, Methodology, Validation, Writing – original draft, Writing – review & editing, Software, Supervision. **Serhat Duman:** Software, Validation, Investigation, Resources, Data curation, Visualization. **Ugur Guvenc:** Software, Validation, Investigation, Resources, Data curation, Visualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The source code available at: https://www.mathworks.com/matlabcentral/fileexchange/130244-fitness-distance-constraint-in-constrained-optimization?s_tid=srchtitle_FDC_1.

References

[1] R. Mallipeddi, P.N. Suganthan, Ensemble of constraint handling techniques, IEEE Trans. Evol. Comput. 14 (4) (2021) 561–579.

[2] K. Yu, X. Wang, Z. Wang, Constrained optimization based on improved teaching–learning-based optimization algorithm, Inform. Sci. 352 (2016) 61–78.

[3] F. Wang, H. Zhang, A. Zhou, A particle swarm optimization algorithm for mixed-variable optimization problem, Swarm Evol. Comput. 60 (2021) 100808.

[4] A. Yadav, K. Deep, J.H. Kim, A.K. Nagar, Gravitational swarm optimizer for global optimization, Swarm Evol. Comput. 31 (2016) 64–89.

[5] W.C. Wang, L. Xu, K.W. Chau, C.J. Liu, Q. Ma, D.M. Xu, Cε-LDE: A lightweight variant of differential evolution algorithm with combined ε constrained method and Lévy flight for constrained optimization problems, Expert Syst. Appl. 211 (2023) 118644.

[6] N. Amjady, H. Sharifzadeh, Solution of non-convex economic dispatch problem considering valve loading effect by a new modified differential evolution algorithm, Int. J. Electr. Power Energy Syst. 32 (8) (2010) 893–903.

[7] S. Khalilpourazari, S. Khalilpourazary, A Robust Stochastic Fractal Search approach for optimization of the surface grinding process, Swarm Evol. Comput. 38 (2018) 173–186.

[8] Z. Li, D. Zou, Z. Kong, A harmony search variant and a useful constraint handling method for the dynamic economic emission dispatch problems considering transmission loss, Eng. Appl. Artif. Intell. 84 (2019) 18–40.

[9] V.H. Cantú, C. Azzaro-Pantel, A. Ponsich, Constraint-handling techniques within differential evolution for solving process engineering problems, Appl. Soft Comput. 108 (2021) 107442.

[10] K. Deb, An efficient constraint handling method for genetic algorithms, Comput. Methods Appl. Mech. Engrg. 186 (2–4) (2000) 311–338.

[11] X. Yuan, L. Wang, Y. Yuan, Y. Zhang, B. Cao, B. Yang, A modified differential evolution approach for dynamic economic dispatch with valve-point effects, Energy Convers. Manage. 49 (12) (2008) 3447–3453.
[12] X. Yuan, A. Su, Y. Yuan, H. Nie, L. Wang, An improved PSO for dynamic load dispatch of generators with valve-point effects, Energy 34 (1) (2009) 67–74.
[13] S. Chu, Z. Yang, M. Xiao, H. Qiu, K. Gao, L. Gao, Explicit topology optimization of novel polyline-based core sandwich structures using surrogate-assisted evolutionary algorithm, Comput. Methods Appl. Mech. Eng. 369 (2020) 113215.
[14] H. Li, B. Song, X. Tang, Y. Xie, X. Zhou, Controller optimization using data-driven constrained bat algorithm with gradient-based depth-first search strategy, ISA Trans. 125 (2022) 212–236.
[15] E. Zahara, Y.T. Kao, Hybrid Nelder–Mead simplex search and particle swarm optimization for constrained engineering design problems, Expert Syst. Appl. 36 (2) (2009) 3880–3886.
[16] R. Mallipeddi, S. Jeyadevi, P.N. Suganthan, S. Baskar, Efficient constraint handling for optimal reactive power dispatch problems, Swarm Evol. Comput. 5 (2012) 28–36.
[17] N. Amjady, H. Sharifzadeh, Security constrained optimal power flow considering detailed generator model by a new robust differential evolution algorithm, Electr. Power Syst. Res. 81 (2) (2011) 740–749.
[18] M. Hajinassiry, N. Amjady, H. Sharifzadeh, Hydrothermal coordination by bi-level optimization and composite constraint handling method, Int. J. Electr. Power Energy Syst. 62 (2014) 476–489.
[19] L. Fu, H. Ouyang, C. Zhang, S. Li, A.W. Mohamed, A constrained cooperative adaptive multi-population differential evolutionary algorithm for economic load dispatch problems, Appl. Soft Comput. 121 (2022) 108719.
[20] H. Peng, Z. Xu, J. Qian, X. Dong, W. Li, Z. Wu, Evolutionary constrained optimization with hybrid constraint-handling technique, Expert Syst. Appl. 211 (2023) 118660.
[21] C. Hongwei, F. Qihong, Z. Xianmin, W. Sen, Z. Wensheng, L. Fan, Well placement optimization with cat swarm optimization algorithm under oilfield development constraints, J. Energy Resour. Technol. 141 (1) (2019).
[22] G. Miao, K. Zhuo, G. Li, J. Xiao, An advanced optimization strategy for enhancing the performance of a hybrid pressure-swing distillation process in effective binary-azeotrope separation, Sep. Purif. Technol. 282 (2022) 120130.
[23] Y. Lu, J. Zhou, H. Qin, Y. Li, Y. Zhang, An adaptive hybrid differential evolution algorithm for dynamic economic dispatch with valve-point effects, Expert Syst. Appl. 37 (7) (2010) 4842–4849.
[24] A.O. Kusakci, M. Can, An adaptive penalty based covariance matrix adaptation–evolution strategy, Comput. Oper. Res. 40 (10) (2013) 2398–2417.
[25] H. Zhang, G.P. Rangaiah, An efficient constraint handling method with integrated differential evolution for numerical and engineering optimization, Comput. Chem. Eng. 37 (2012) 74–88.
[26] A. Kumar, G. Wu, M.Z. Ali, R. Mallipeddi, P.N. Suganthan, S. Das, A test-suite of non-convex constrained optimization problems from the real-world and some baseline results, Swarm Evol. Comput. 56 (2020) 100693.
[27] M. Dehghani, Z. Montazeri, E. Trojovská, P. Trojovský, Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems, Knowl.-Based Syst. 259 (2023) 110011.
[28] M. Azizi, S. Talatahari, A.H. Gandomi, Fire Hawk Optimizer: A novel metaheuristic algorithm, Artif. Intell. Rev. 56 (1) (2023) 287–363.
[29] C. Zhong, G. Li, Z. Meng, Beluga whale optimization: A novel nature-inspired metaheuristic algorithm, Knowl.-Based Syst. 251 (2022) 109215.
[30] S. Zhao, T. Zhang, S. Ma, M. Chen, Dandelion Optimizer: A nature-inspired metaheuristic algorithm for engineering applications, Eng. Appl. Artif. Intell. 114 (2022) 105075.
[31] N. Chopra, M.M. Ansari, Golden jackal optimization: A novel nature-inspired optimizer for engineering applications, Expert Syst. Appl. 198 (2022) 116924.
[32] B. Abdollahzadeh, F.S. Gharehchopogh, N. Khodadadi, S. Mirjalili, Mountain gazelle optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems, Adv. Eng. Softw. 174 (2022) 103282.
[33] Z.K. Feng, W.J. Niu, S. Liu, Cooperation search algorithm: A novel metaheuristic evolutionary intelligence algorithm for numerical optimization and engineering optimization problems, Appl. Soft Comput. 98 (2021) 106734.
[34] S. Biswas, D. Saha, S. De, A.D. Cobb, S. Das, B.A. Jalaian, Improving differential evolution through Bayesian hyperparameter optimization, in: 2021 IEEE Congress on Evolutionary Computation, CEC, 2021, pp. 832–840.
[35] A.W. Mohamed, A.K. Mohamed, Adaptive guided differential evolution algorithm with novel mutation for numerical optimization, Int. J. Mach. Learn. Cybern. 10 (2019) 253–277.
[36] A.W. Mohamed, A.A. Hadi, A.M. Fattouh, K.M. Jambi, LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problem, in: 2017 IEEE Congress on Evolutionary Computation, CEC, 2017, pp. 145–152.
[37] P.P. Biswas, P.N. Suganthan, G.A. Amaratunga, Optimal power flow solutions incorporating stochastic wind and solar power, Energy Convers. Manage. 148 (2017) 1194–1207.
[38] Y. Wang, J. Zhou, H. Qin, Y. Lu, Improved chaotic particle swarm optimization algorithm for dynamic economic dispatch problem with valve-point effects, Energy Convers. Manage. 51 (12) (2010) 2893–2900.
[39] G. Chen, L. Liu, Z. Zhang, S. Huang, Optimal reactive power dispatch by improved GSA-based algorithm with the novel strategies to handle constraints, Appl. Soft Comput. 50 (2017) 58–70.
[40] P.B. Rana, D.I. Lalwani, Optimization of turning process using amended differential evolution algorithm, Eng. Sci. Technol. Int. J 20 (4) (2017) 1285–1301.
[41] W.J. Niu, Z.K. Feng, Y.B. Chen, Y.W. Min, S. Liu, B.J. Li, Multireservoir system operation optimization by hybrid quantum-behaved particle swarm optimization and heuristic constraint handling technique, J. Hydrol. 590 (2020) 125477.
[42] K.M. Sallam, S.M. Elsayed, R.K. Chakrabortty, M.J. Ryan, Multi-operator differential evolution algorithm for solving real-world constrained optimization problems, in: 2020 IEEE Congress on Evolutionary Computation, CEC, 2020, pp. 1–8.
[43] J. Gurrola-Ramos, A. Hernàndez-Aguirre, O. Dalmau-Cedeño, COLSHADE for real-world single-objective constrained optimization problems, in: 2020 IEEE Congress on Evolutionary Computation, CEC, 2020, pp. 1–8.
[44] D. Zou, S. Li, X. Kong, H. Ouyang, Z. Li, Solving the combined heat and power economic dispatch problems by an improved genetic algorithm and a new constraint handling strategy, Appl. Energy 237 (2019) 646–670.
[45] Q. Zhang, D. Zou, N. Duan, X. Shen, An adaptive differential evolutionary algorithm incorporating multiple mutation strategies for the economic load dispatch problem, Appl. Soft Comput. 78 (2019) 641–669.
[46] P.P. Biswas, P.N. Suganthan, R. Mallipeddi, G.A. Amaratunga, Optimal reactive power dispatch with uncertainties in load demand and renewable energy sources adopting scenario-based approach, Appl. Soft Comput. 75 (2019) 616–632.
[47] S. Elsayed, R. Sarker, C.C. Coello, T. Ray, Adaptation of operators and continuous control parameters in differential evolution for constrained optimization, Soft Comput. 22 (2018) 6595–6616.
[48] G. Xiong, D. Shi, Hybrid biogeography-based optimization with brain storm optimization for non-convex dynamic economic dispatch with valve-point effects, Energy 157 (2018) 424–435.
[49] H.T. Kahraman, S. Aras, E. Gedikli, Fitness-distance balance (FDB): a new selection method for meta-heuristic search algorithms, Knowl.-Based Syst. 190 (2020) 105169.
[50] E. Mezura-Montes, C.A.C. Coello, Constraint-handling in nature-inspired numerical optimization: past, present and future, Swarm Evol. Comput. 1 (4) (2011) 173–194.
[51] P.P. Biswas, P.N. Suganthan, R. Mallipeddi, G.A. Amaratunga, Optimal power flow solutions using differential evolution algorithm integrated with effective constraint handling techniques, Eng. Appl. Artif. Intell. 68 (2018) 81–100.
[52] K.M. Ang, W.H. Lim, N.A.M. Isa, S.S. Tiang, C.H. Wong, A constrained multi-swarm particle swarm optimization without velocity for constrained optimization problems, Expert Syst. Appl. 140 (2020) 112882.
[53] Z. Liu, Z. Qin, P. Zhu, H. Li, An adaptive switchover hybrid particle swarm optimization algorithm with local search strategy for constrained optimization problems, Eng. Appl. Artif. Intell. 95 (2020) 103771.
[54] H.T. Kahraman, M. Katı, S. Aras, D.A. Taşçi, Development of the Natural Survivor Method (NSM) for designing an updating mechanism in metaheuristic search algorithms, Eng. Appl. Artif. Intell. 122 (2023) 106121.
[55] M. Pant, H. Zaheer, L. Garcia-Hernandez, A. Abraham, Differential evolution: A review of more than two decades of research, Eng. Appl. Artif. Intell. 90 (2020) 103479.
[56] M.F. Ahmad, N.A.M. Isa, W.H. Lim, K.M. Ang, Differential evolution: A recent review based on state-of-the-art works, Alexandria Eng. J. 61 (5) (2022) 3831–3872.