BALKAN JOURNAL OF ELECTRICAL & COMPUTER ENGINEERING, Vol. 12, No. 2, June 2024
Research Article

Optimal Solution of the Dynamic Economic Dispatch by Improved Teaching-Learning-Based Artificial Bee Colony Algorithm

Burçin Ozkaya

Abstract—**Dynamic economic dispatch is one of the most handled problem in modern power system operations. It aims to optimize the output power from thermal generating units over a specified time period to minimize the total fuel cost, while satisfying the several constraints such as generation limits, ramp rate limits, and power balance. In addition to these constraints, the prohibited operating zones and the valve-point loading effect are included the DED problem. In this case, the complexity, nonlinearity, and non-convexity of the DED problem are increases. Therefore, in order to solve the DED problem, a powerful meta-heuristic search (MHS) algorithm are proposed. In this study, an improved teaching-learning-based artificial bee colony (TLABC) algorithm, where the fitness-distance balance based TLABC (FDB-TLABC) and natural-survivor method based TLABC (NSM-TLABC) algorithms were hybridized. To prove the performance of the proposed algorithm, it was applied to solve the DED problem and benchmark problem suites. In the simulation study carried out on benchmark problems, the results of the proposed algorithm and five MHS algorithms were evaluated statistically. According to Friedman test results, the proposed algorithm ranked first with 2.2836 values among them. On the other hand, the proposed algorithm and its rival algorithms were applied to solve the two DED cases. The results of them show that the proposed algorithm achieved superior performance to find the best objective values for both case studies.**

Index Terms—**Dynamic economic dispatch, Natural-Survivor method, Fitness-Distance Balance, NSM-FDB-TLABC algorithm.**

I. Introduction

IN RECENT years, the dynamic economic dispatch (DED) problem has become a crucial optimization challenge in modern power system operations. It focuses on optimizing the output power from generators over time to minimize fuel costs while adhering to various constraints [1, 2]. When compared to the static ED problem, DED is a very complex problem as it must comply with the generation unit ramp rate limits and plan the output powers of all generation units within a specific time period [3]. On the other hand, the cost function for each generator has been generally simplified as a quadratic function, disregarding the valve-point loading effect (VPLE) arising from the multiple steam admitting valves. In order to accurately model the DED problem as a real world operation, it is crucial to incorporate the impact of VPLE on the cost of power generators and the constraints such as the generator capacity limits, the ramp-rate limits, the power balance constraints, and the prohibited operation zones (POZs). Considering all of these constraints, the DED problem exhibits non-convex and non-linear properties that make it more complex [3, 4]. To solve this complex DED problem, meta-heuristic search (MHS) algorithms have been applied.

In the literature, several studies have been carried out for the solution of the DED problem using MSH algorithms. Mohammadi-Ivatloo et al. [1] used the imperialist competitive algorithm for solving the DED problem, incorporating the VPLE, transmission losses, and POZs. The performance of the algorithm was proven on a 5-, 10-, and 54-unit test system. Ivatloo et al. [2] proposed a time-varying acceleration coefficients iteration particle swarm optimization for the solution of the DED problem. In the study, transmission losses of the system, VPLE, and POZs were considered. Sonmez et al. [3] used the symbiotic organisms search algorithm for the solution of the DED problem, where it was applied to five case studies created using three scale of test systems and operational constraints. Dai et al. [4] proposed an adaptive hybrid backtracking search optimization algorithm in order to solve the DED problem with VPLE. In the study, six case studies were performed, which were created using three scales of test systems, transmission loss, and POZs. Mohammadi-Ivatloo et al. [5] proposed a hybrid algorithm by combining the immune and genetic algorithms to solve the DED problem, including VPLE and POZs, where the three scales of test systems were taken into account. Mohammadi- Azizipanah-Abarghooee [6] introduced hybrid bacterial foraging and simplified swarm optimization algorithms for solving the DED problem. In the study, the proposed algorithm was applied on four different test systems. Moreover, the cases with and without the inclusion of operational constraints were analyzed. Zhang et al. [7] presented a hybrid bare-bones particle swarm optimization including directionally chaotic search to solve the DED problem with VPLE, where three case studies were considered. Xiong and Shi [8] proposed a hybrid method by combining biogeography-based optimization with brain storm optimization for solving the DED problem, where the transmission loss and VPLE were considered. Here, three scales

Burçin Özkaya, is with Department of Electrical Engineering, Bandirma Onyedi Eylul University, Balıkesir, Turkey, (e-mail: bozkaya@bandirma.edu.tr).
https://orcid.org/0000-0002-9858-3982
Manuscript received May 17, 2024; accepted Jun 27, 2024.
DOI: 10.17694/bajece.1486015

of test systems including 5-, 10-, and 30-units were used. Zou et al. [9] introduced a memory-based global differential evolution algorithm for the solution of five case studies of the DED problem considering the VPLE, transmission loss, and POZs. To handle the constraints of the DED problem, a repair technique was proposed. Ghasemi et al. [10] presented a novel version of the particle swarm optimization algorithm to solve the DED problem in 10- and 30-unit test systems, where the four case studies were created using test systems and operational constraints. Zheng et al. [11] proposed an improved version of the invasive weed optimization algorithm to solve the DED problem, including VPLE, POZs, and transmission losses. Here, to prove the performance of the proposed algorithm, six case studies of the DED problem on three different-scales were performed. Santra et al. [12] presented a hybrid method incorporating termite colony optimization and particle swarm optimization algorithms for the solution of the four DED cases, where 5-, 10-, and 30-unit test systems were used. Yang et al. proposed an enhanced exploratory whale optimization algorithm for the solution of the DED problem, where both VPLE and transmission losses were considered. Hu et al. [14] introduced an adaptive backtracking search optimization algorithm including the dual-learning strategy to solve the DED problem with VPLE and transmission losses, where 5-, 10-, and 30-units test systems were considered. Basak et al. [15] introduced a hybrid algorithm based on crow search algorithm and JAYA to solve the DED problem including wind energy sources, where 10- and 15-unit test systems were studied. Yang et al. [16] presented an improved chaos moth flame optimization algorithm for solving the DED problem where plug-in electric vehicles (PEVs) were connected to the grid. Yang et al. [17] proposed an improved grey wolf optimization algorithm to solve the DED problem, where 5-, 10-, and 15-unit test systems were considered. Nagarajan et al. [18] presented an enhanced cheetah optimizer algorithm for the solution of the DED problem incorporating wind and solar energy sources.

The DED problem is still a very complex and constrained optimization problem as it tries to optimize the output power from generators over time to minimize fuel costs while satisfying the constraints. In particular, the level of complexity increases with the inclusion of the operational constraints such as VPLE and POZ. When the studies summarized above were evaluated, various algorithms have been proposed by researchers to solve the DED problem. However, it was observed that the proposed algorithms were insufficient to find the optimal solution because they were not designed in accordance with the structure and constraints of the DED problem. Therefore, in this study, an improved teaching-learning-based artificial bee colony (TLABC) algorithm was proposed for the solution of the DED problem, where the fitness-distance balance based TLABC (FDB-TLABC) [19] and the natural-survivor method based TLABC (NSM-TLABC) [20] algorithms were hybridized. In the NSM-FDB-TLABC algorithm, while the selection of surviving individuals in the teaching-based employed bee stage was carried out using the NSM method, the guide individual in the learning-based onlooker bee stage was selected using the FDB method. Thus, with the use of NSM and FDB methods in the proposed algorithm, it was aimed to enhance the ability of the TLABC algorithm to imitate nature and to enhance its exploitation and exploration capabilities. To prove the performance of the NSM-FDB-TLABC algorithm, it was applied to solve both DED and benchmark problems.

The contributions of this study were explained as below:

- The NSM-FDB-TLABC algorithm was proposed in the literature as a competitive MHS algorithm.
- The proposed algorithm was implemented for solving both DED and benchmark problems.
- The best optimal solutions were obtained for solving the DED problem by the proposed algorithm.

The outline of the rest of the study is explained as follows: Section 2 presents the formulation of the DED problem. In section 3, the proposed NSM-FDB-TLABC algorithm is introduced. In section 4, the simulation study and results are given. Section 5 presents the conclusion of the study.

II. Formulation of the Dynamic Economic Dispatch Problem

In this study, the DED problem is considered. Here, the goal is to minimize the total fuel cost of the system during the dispatch period. Traditionally, the fuel cost of the thermal generating units can be defined as the quadratic cost function. However, in multi-valve stream turbine-based generators, the valve-point loading effect (VPLE) is widely considered. For the DED problem, it is required to model the VPLE on the cost function of the thermal generating units [4]. The cost function including VPLE can be mathematically expressed as in Eq. (1).

$$F_{k,t}\left(P_{k,t}\right)=\delta_k+\beta_k P_{k,t}+\alpha_k P_{k,t}^2+\left|\mu_k \sin\left(\xi_k\left(P_{k,t}^{\min}-P_{k,t}\right)\right)\right| \tag{1}$$

Here, $F_{k,t}$ and $P_{k,t}$ are the total fuel cost and the output power of the kth thermal generating unit at the time interval t, respectively. $\{\delta_k, \beta_k, \alpha_k\}$ denote the cost coefficients of the kth thermal generating unit, μ_k and ξ_k are the coefficients of the VPLE. The objective function of the DED problem can be expressed as in Eq. (2) [4]:

$$Minimize\ OF=\sum_{t=1}^{T}\sum_{k=1}^{N}F_{k,t}\left(P_{k,t}\right) \tag{2}$$

Here, OF represents the objective function, which is the fuel cost of the system. T and N denote the number of dispatch time periods and the number of generating units, respectively.

A. Constraints

In the DED problem, both equality and inequality constraints are taken into account.

Equality constraints: The power balance equations with transmission losses for each hour are expressed as [4]:

$$P_{D,t} + P_{L,t} = \sum_{k=1}^{N} P_{k,t},\ t = 1,2,\ldots,T \tag{3}$$

where $P_{D,t}$ is the sum of the power demand at the time interval t and $P_{L,t}$ is the total transmission loss of the system at the time interval t, calculated using the Eq. (4). Here, B_{ji}, B_{0j}, and B_{00} denote the loss coefficients.

$$P_{L,t} = \sum_{j=1}^{N}\sum_{i=1}^{N} P_{j,t} B_{ji} P_{i,t} + \sum_{j=1}^{N} B_{0j} P_{j,t} + B_{00} \tag{4}$$

Inequality constraints:

(*i*) Generator constraints: Each generator's output power is constrained by its upper ($P_k^{\max}$) and lower ($P_k^{\min}$) limits [4], which are determined by:

$$P_k^{\min} < P_k < P_k^{\max} \tag{5}$$

(*ii*) Ramp-rate limits: In reality, the operating conditions affect the way in which the active output power of each generator is adjusted. It should fall within an acceptable range at each interval and can be modeled by the following ramp rate limits [4]:

$$\begin{cases} P_{k,t} - P_{k,t-1} \le UR_k \\ P_{k,t-1} - P_{k,t} \le DR_k \end{cases},\ t = 1,2,\ldots,T \tag{6}$$

where DR_k and UR_k are the down- and up-rate limits of the kth generating unit.

(*iii*) Prohibited operating zones (POZs): The POZs limits of the thermal generating limits can be described as in Eq. (7). Here, $P_{k,z}^{l}$ and $P_{k,z}^{u}$ are the lower and upper limits of the zth POZ, m is the number of the POZs of kth unit [4].

$$P_{k,t} \begin{cases} P_k^{\min} \le P_{k,t} \le P_{k,1}^{l} \\ P_{k,z-1}^{u} \le P_{k,t} \le P_{k,z}^{l},\ z = 2,3,\ldots,m \\ P_{k,m}^{u} \le P_{k,t} \le P_k^{\max} \end{cases} \tag{7}$$

B. Constraint Handling Method

The DED problem consists of equality and inequality constraints. To handle these constraints, the most commonly used constraint handling method, called as the penalty function method, is used. In this method, the constraint violations are multiplied by a penalty coefficient and added to the objective function [21]. All the constraint violation degrees must be determined before using the penalty function approach. In the DED problem considered in this study, three constraint violations must be taken into account. Accordingly, the fitness function (*fitness*) of the DED problem can be expressed as:

$$fitness = OF + \xi_1 \cdot V_{CL} + \xi_2 \cdot V_{RR} + \xi_3 \cdot V_{POZ} \tag{8}$$

where V_{CL} and V_{RR} represent the violation degrees for the capacity limits, the ramp-rate limits of the thermal generating units, respectively. V_{POZ} is the violation degrees for the POZs. If POZs is not included in the DED problem, it should not be included in the fitness function. ξ_1, ξ_2, and ξ_3 denote the penalty coefficient for the V_{CL}, V_{RR}, and V_{POZ},

III. Proposed Method: Hybrid NSM and FDB based TLABC (NSM-FDB-TLABC) Algorithm

A. Overview of the TLABC algorithm

TLABC is an optimization algorithm that combines the exploration process of the ABC algorithm and the exploitation process of the TLBO algorithm. The teaching-based employed bee stage, the generalized oppositional scout bee stage, and the learning-based looker bee stage are the search phases used in TLABC to find solutions. TLABC begins with the randomly generated *NP* food sources, and then three stages are applied [22].

Teaching-based employed bee stage: Each employed bee looks for a new food source performed using Eq. (9).

$$v_{i,d} = \begin{cases} x_{i,d}^{old} + r_2\left(x_{t,d} - T_F x_{m,d}\right), \text{ if } r_1 < 0.5 \\ x_{n1,d} + F\left(x_{n2,d} - x_{n3,d}\right), \text{ otherwise} \end{cases} \tag{9}$$

In Eq. (9), r_1 and r_2 are the uniformly distributed random numbers within [0,1], $n1$, $n2$, and $n3$ are selected randomly integers within [1, *NP*], where $n1 \neq n2 \neq n3 \neq i$. The scale factor is denoted by F, whose value is between [0, 1].

Learning-based on looker bee stage: Here, an onlooker bee uses the selection probability (p) to determine which food source x_s to seek out. Then, the new food sources are searched by Eq. (10). Here, *rand* is a uniformly distributed random vector within [0, 1], j is the number in the range of [1, *NP*], and $j \neq s$.

$$v_s = \begin{cases} rand\left(x_s - x_j\right) + x_s, \text{ if } f\left(x_s\right) \le f\left(x_j\right) \\ rand\left(x_j - x_s\right) + x_s, \text{ otherwise} \end{cases} \tag{10}$$

Generalized oppositional scout bee stage: After the learning-based on looker bee stage, the algorithm enters this stage. In this stage, a food source is deemed exhausted and would be abandoned if it could not be improved any further for at least a limited amount of time. A new solution candidate is generated by Eq. (11) and its oppositional solution (x_i^{op}) is generated using Eq. (12). ρ and m are randomly generated numbers between 0 and 1.

$$x_{ij} = \rho\left(x_{max,j} - x_{min,j}\right) + x_{min,j} \tag{11}$$

$$x_{ij}^{op} = m\left(\max\left(x_{ij}\right) + \min\left(x_{ij}\right)\right) - x_{ij} \tag{12}$$

Finally, the better solution of x_i and x_i^{op} are used to replace the old depleted food source according to Eq. (13).

$$x_i = \begin{cases} x_i, & \text{if } f(x_{ij}^{op}) \ge f(x_i) \\ x_{ij}^{op}, & \text{if } f(x_{ij}^{op}) \ge f(x_i) \end{cases} \tag{13}$$

B. Overview of the FDB method

The FDB selection method, presented in the literature by Kahraman et al. in 2020 [23], is a selection method preferred by researchers for the last three years to enhance the search performance of MHS algorithms. The aim of this method is to effectively discover the guiding solution candidate that will contribute the most to the search process in MHS algorithms. In this method, the selection process is performed according to the score value calculated by considering the fitness values of the solution candidates and their distance to the best solution candidate (x_{best}) in the population [23].

In a population (P), the fitness value (fv) of each solution candidate is computed. The vector representing the population and fitness values is given in Eq. (14). Here, m and n represent the number of design variables and solution candidates, respectively.

$$P \equiv \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix}_{n \times m}, \quad fv \equiv \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix}_{n \times 1} \tag{14}$$

The calculation of the FDB score is explained step by step as follows:

(i) The distance value of the ith solution candidate (x_i) from the x_{best} is calculated using Eq. (15). The distance vector can be defined in Eq. (16).

$$\forall_{i=1}^{m}, P_i \neq P_{best}, D_{P_i} = \sqrt{\left(x_{1[i]} - x_{1[best]}\right)^2 + \ldots + \left(x_{m[i]} - x_{m[best]}\right)^2} \tag{15}$$

$$D_P \equiv \begin{bmatrix} d_1 \\ \vdots \\ d_n \end{bmatrix}_{n \times 1} \tag{16}$$

(ii) The FDB score is calculated with Eq. (17) using the distance values (D_P) given in Eq. (16) and the fitness values (fv) given in Eq. (14), where ς is the weighting factor that is taken value in the range of [0, 1]. In this study, it is set as 0.5. The vector of score vector is expressed as in Eq. (18).

$$\forall_{i=1}^{m} P_i, S_{P_i} = \varsigma * normfv_i + (1 - \varsigma) * normD_{P_i} \tag{17}$$

$$S_P \equiv \begin{bmatrix} s_1 \\ \vdots \\ s_n \end{bmatrix}_{n \times 1} \tag{18}$$

To learn more about the FDB selection process in detail, you can review Ref [23].

C. Overview of the NSM method

The Natural Survivor Method (NSM) is a new method introduced to the literature by Kahraman et al. in 2023 [20], that can be preferred in the update mechanism to identify the survivors in MHS algorithms. It is determined which individuals will survive and which will die by using NSM scores instead of fitness values in the update mechanism in MHS algorithms. In this method, the success of each individual in fulfilling their duties, that is, their NSM score, is calculated and the individual's survival depends on this score. There are three criteria to determine the NSM score of an individual: (i) the individual's contribution to the mating pool; (ii) the individual's contribution to the population; and (iii) the individual's contribution to its fitness value. The criteria for calculating the NSM score are explained below.

Contribution of the guides: Within the NSM, it is considered that an individual has a better chance of surviving if it makes a greater genetic diversity contribution to the guide solution candidate than its rivals. The distance information was used to determine the similarity between two individuals, and the chances of survival of individuals with different characteristics than the guide solution candidates were increased.

Contribution of the population: In this contribution, the distance information between an individual and others is used, and this information shows the difference between the candidates who will survive in the population and the others. In summary, an individual has a better chance of survival if they contribute more to the population's diversity.

Contribution to the objective function: The most successful people are those who have the highest fitness values for the objective function. The fitness value of the individual is used to represent individual strength.

According to these contributions, the NSM score of x_i is computed by using Eq. (19). Here, $x_{i,NSM_{MPS}}$, $x_{i,NSM_{PS}}$, and $x_{i,NSM_{OFS}}$ correspond to the mating pool source value of x_i, contribution of solution candidate x_i to the P-population, and individual strength of x_i, respectively. w_1, w_2, and w_3 are the weighting coefficients.

$$x_{i,NSM_{score}} = w_1 * x_{i,NSM_{MPS}} + w_2 * x_{i,NSM_{PS}} + w_3 * x_{i,NSM_{OFS}} \tag{19}$$

To obtain more detailed information about the NSM method, you can review [20].

D. Proposed NSM-FDB-TLABC algorithm

In the literature, two powerful versions of the TLABC algorithm were presented. One of these is the FDB-TLABC algorithm [19], while the other is the NSM-TLABC algorithm [20]. In this study, by combining FDB-TLABC and NSM-TLABC, an extremely powerful and competitive hybrid TLABC version was proposed, called as the NSM-FDB-TLABC. In the proposed algorithm, the FDB method was applied to the TLABC algorithm to maximize the ability of the base TLABC algorithm to mimic nature. In other words, the goal was to enhance the TLABC algorithm's exploration, exploitation, and balanced search capabilities [19]. The improvements were made in the learning-based onlooker bee of the TLABC algorithm and the solution candidate given in Eq. (10) chosen by the FDB method was used instead of x_s given in Eq. (20).

$$v_s = \begin{cases} rand\left(x_{FDB} - x_j\right) + x_s, & \text{if } f\left(x_j\right) \geq f\left(x_s\right) \\ rand\left(x_j - x_{FDB}\right) x_s, & \text{otherwise} \end{cases} \tag{20}$$

In the proposed algorithm, the NSM-based update mechanism was implemented on the teaching-based employed bee stage of the TLABC algorithm. In general, there are three steps common to all MHS algorithms. These include choosing the population's guiding solution candidate, developing a new solution candidate from the selected individual, and figuring out the fitness function values of the newly formed and original individuals. After these three stages are completed, the traditional update mechanism in MHS algorithms is performed, where the fitness function values of the newly created individual and the previous individual are compared, and then the individual with the better fitness function value survives. However, in the NSM-based update mechanism, the NSM score is calculated to determine the surviving individual by using Eq. (17), where the individual with the better score value survives.

The pseudocode of the NSM-FDB-TLABC algorithm is given in Algorithm-1.

Algorithm-1. Pseudocode of the NSM-FDB-TLABC algorithm

1. Initialize the position of individual x_i (i=1,…, NP)
2. Calculate the fitness function value of the population $fv(x_i)$
3. Set $trial$=0 for each individual.
4. ***while*** $FE < maxFEs$ ***do***

    //Teaching-based employed bee stage//

5. ***for*** $i = 1 : NP$
6. Create a new solution candidate v_i by using Eq. (10)
7. Compute the fitness function value of each individual $fv(v_i)$

    /NSM-based update mechanism/

8. Calculate the NSM score of x_i and v_i by using Eq. (19)
9. ***if*** ($v_i > x_i$) ***then*** $x_i = v_i$, ***else*** use x_i;
10. ***if*** x_i doesn't improve then $trial(i) = trial(i)+1$, ***else*** $trial(i)$=0;
11. ***end for***

    //Learning-based employed bee stage//

12. Compute the probability p
13. ***for*** $i = 1 : NP$
14. Create an individual x_i

    /FDB-based selection method/

15. Calculate the FDB score of the each individual by using Eq. (17)
16. Select an individual based on the FDB-score of individuals.
17. Calculate the fitness function value of selected individual v_s
18. ***if*** ($v_s > x_i$) ***then*** $x_i = v_s$, ***else*** use x_i;
19. ***if*** x_i doesn't improve ***then*** $trial(i)= trial(i)+1$, ***else*** $trial(i)$=0;
20. ***end for***

    //Generalized oppositional bee stage//

21. ***if*** $limit \leq \max(trial(i))$ ***then***
22. Create a new solution candidate solution x_i and its oppositional solution x_i^{op}
23. Specify a better solution candidate between x_i and x_i^{op} by Eq. (12)
24. ***end if***
25. ***end while***

According to Algorithm-1, the NSM-FDB-TLABC algorithm initializes with a randomly generated population in line 1, and then the fitness values of them are calculated in line 2. After that, the search-process lifecycle begins. Between lines 5 and 11, the teaching-based employed bee stage performs. The NSM-based method is applied to the update mechanism of the individual between lines 8 and 10. Then, the learning-based employed bee stage performs from lines 12 to 20. The FDB method is used to select the guide solution candidate between lines 15 and 17, and then the update process is performed. The generalized oppositional bee stage performs in lines 21 to 23. All of these processes continue until the termination requirement is fulfilled, which is specified as the maximum number of fitness function evaluations (maxFEs).

IV. Simulation Results and Discussions

In this section, in order to validate the performance of the NSM-FDB-TLABC algorithm, an extensive simulation study was conducted.

- In the first sub-section, the performance of the NSM-FDB-TLABC algorithm on benchmark problems was tested. Accordingly, the CEC2017 [24] and CEC2020 [25] benchmark suites were used. To show the improvement of the proposed algorithm against the its rival algorithms, including FDB-TLABC, NSM-TLABC, FLA [26], EMA [27], and EO [28], their results were analyzed and examined using the statistical analysis methods such as Friedman and Wilcoxon.
- In the second sub-section, the NSM-FDB-TLABC algorithm was implemented to solve the DED problem. To show the superiority of the proposed algorithm in solving the DED problem, their results were compared with the results of the NSM-TLABC, FDB-TLABC, EMA, FLA, and EO algorithms. Moreover, the results of the proposed algorithm were compared with the results reported in the literature.

A. Application of the NSM-FDB-TLABC algorithm on benchmark problems

In this sub-section, the proposed NSM-FDB-TLABC and the 5 MHS algorithms, including FDB-TLABC, NSM-TLABC, FLA, EMA, and EO, were applied to solve the CEC2017 and CEC2020 benchmark problems. All algorithms were run 51 trials and three dimensional search spaces (30, 50, and 100). The maximum number of fitness function evaluations (maxFEs), which was set at 10000**Dimension* (D), served as the termination criterion to provide fairness across the methods. Moreover, the parameters of the algorithms were set as given in their original articles. The parameters of the proposed NSM-FDB-TLABC were the same as those of NSM-TLABC and FDB-TLABC algorithms.

To compare the algorithms statistically, Wilcoxon-signed rank and Friedman tests were performed on the results of the benchmark problems for all algorithms. The Friedman test results are presented in Table 1. While performing the Friedman test, the error values of the algorithms for benchmark problems were used. In Table 1, the results of the six experiments were given and the best score value of each experiment was highlighted in bold. According to Table 1, the proposed algorithm yielded the best score value in 5 of 6 experiments. In the experiment conducted in the CEC2017 benchmark suite and 100 dimensional search space, the FLA algorithm ranked first and the EMA algorithm ranked second, while the proposed algorithm ranked third. On the other hand, when evaluating the mean rank value of all algorithms displayed in the final row of Table 1, the proposed algorithm achieved the highest ranking of all algorithms, with a score value of 2.2836.

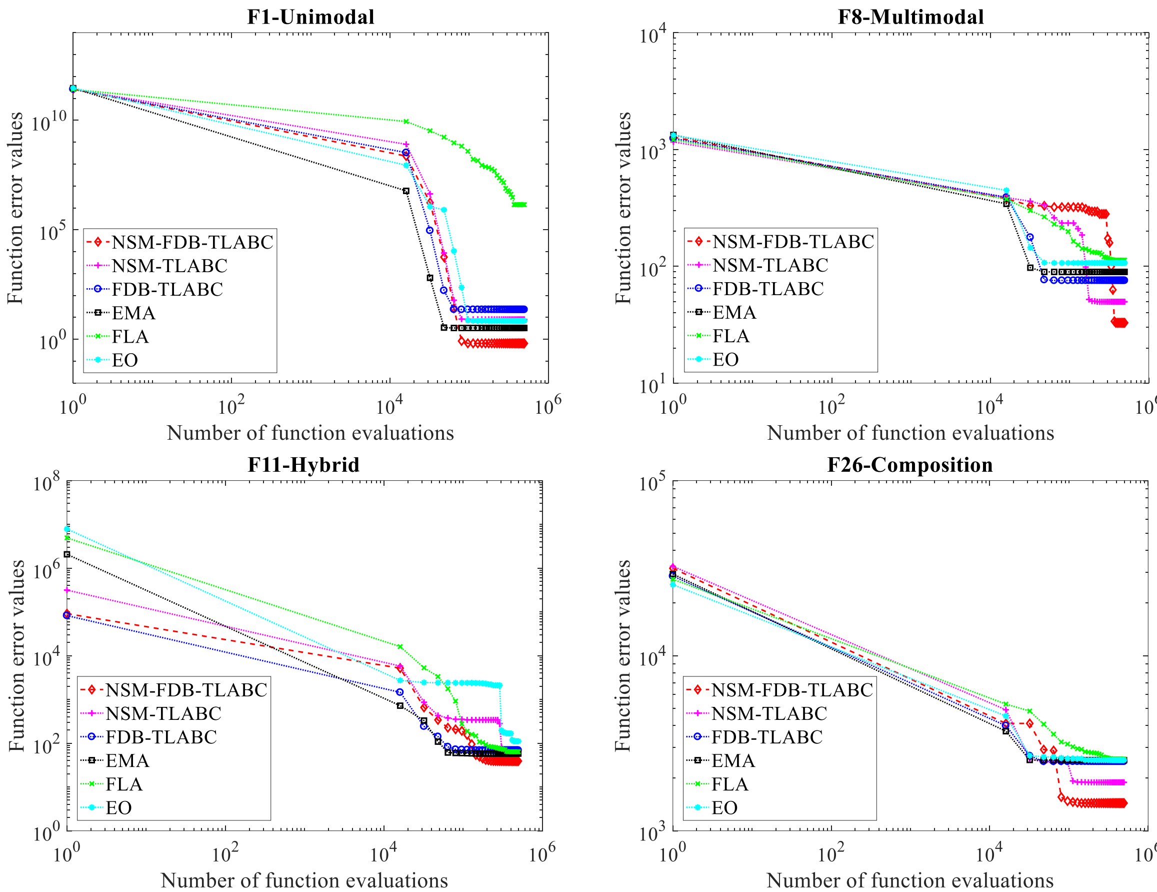


Fig. 1. Convergence graphs of the selected problems for 100-dimensional search space from CEC2017 benchmark suite.

TABLE I
FRIEDMAN TEST RESULTS OF THE ALGORITHMS

	NSM-FDB-TLABC	NSM-TLABC	FDB-TLABC	EMA	FLA	EO
CEC2017 (D=30)	**2.2593**	3.0794	2.5057	4.7417	4.6748	3.7390
CEC2017 (D=50)	**2.0085**	2.9878	2.7028	4.9675	4.4293	3.9040
CEC2017 (D=100)	2.9601	4.0798	3.8661	2.8573	**2.5747**	4.6619
CEC2020 (D=30)	**2.2314**	3.1020	2.5167	4.2814	5.0912	3.7775
CEC2020 (D=50)	**2.0873**	3.1235	2.7137	4.4216	4.7931	3.8608
CEC2020 (D=100)	**2.1549**	2.8608	2.6235	4.8843	4.7725	3.7039
Mean Rank	**2.2836**	3.2056	2.8214	4.3590	4.3893	3.9412

The Wilcoxon test is frequently used to compare MHS algorithms statistically. The NSM-FDB-TLABC and other algorithms were compared using the Wilcoxon test in this study, where the error values obtained from the algorithms were considered. Table 2 presents the results of the Wilcoxon test conducted between the NSM-FDB-TLABC algorithm and its rival algorithms. The number of problems where the NSM-FDB-TLABC won, the number of problems where the NSM-FDB-TLABC and its opponent drew, and the number of problems where the opponent won are represented by " + ", " = ", and " - " signs. According to Table 2, the proposed algorithm lost only to the EMA and FLA algorithms in the experiment conducted in the CEC2017 benchmark suite and 100 dimensional search space. It outperformed its rivals in all other pairwise comparisons.

Besides the Wilcoxon and Friedman tests, the convergence graphs are used to evaluate the search performance of the NSM-FDB-TLABC algorithm and its competitors. Accordingly, four types of problems, which were F1 (unimodal), F8 (multimodal), F11 (hybrid), and F26 (composition) type problems, were selected from the CEC2017 benchmark problem suite. The convergence curves of these problems were drawn based on the function error values in 100 dimensions. The convergence curves of all algorithms are presented in Fig. 1.

TABLE II
WILCOXON TEST RESULTS OF THE ALGORITHMS

NSM-FDB-TLABC vs. (+/=/-)	NSM-TLABC	FDB-TLABC	EMA	FLA	EO
CEC2017 (D=30)	21/5/3	14/6/9	24/3/2	26/3/0	20/4/5
CEC2017 (D=50)	22/6/1	16/7/6	26/3/0	26/2/1	23/4/2
CEC2017 (D=100)	21/8/0	15/8/6	8/7/14	8/8/13	23/4/2
CEC2020 (D=30)	7/2/1	4/3/3	7/2/1	10/0/0	7/2/1
CEC2020 (D=50)	8/2/0	4/4/2	8/2/0	10/0/0	8/2/0
CEC2020 (D=100)	5/5/0	4/4/2	8/2/0	9/1/0	7/3/0

According to Fig. 1, for the F1-unimodal type problem, only the proposed algorithm converged to an error value of 10^0, while its closest competitor, the EO algorithm, converged to an error value of 10^1. The worst performance was performed by the FLA algorithm, whose error value was over 10^5. For the F8-multimodal type problem, the lowest error value was obtained from the proposed NSM-FDB-TLABC algorithm. Hybrid problems are employed to examine the balance between exploitation and exploration of the algorithms. For the F11-hybrid type problem, the convergence curve of all algorithms demonstrated that the NSM-FDB-TLABC converged to an error value below 10^2. It showed better convergence performance compared than the rivals. Composition type problems, known for their computational complexity, are utilized to assess the search performance of algorithms. For the F26-composition type problem, the proposed algorithm converged with a lower minimum error value than others.

In summary, the evaluation of convergence analysis and statistical analysis results demonstrated the effectiveness of the NSMFDB-TLABC algorithm in solving the CEC2017 and CEC2020 benchmark suites in comparison to its rivals.

B. Solving the DED problem using the NSM-FDB-TLABC algorithm

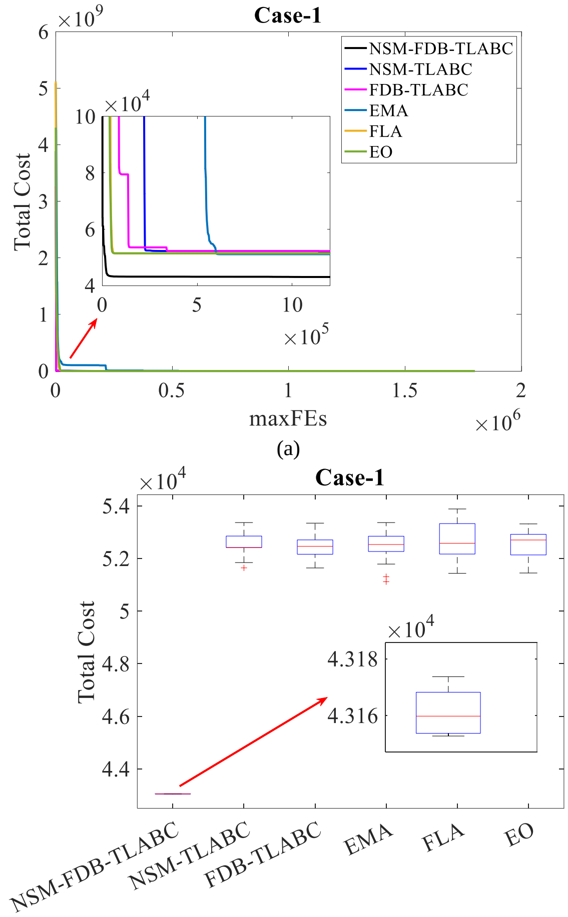
In this section, to show the performance of the proposed NSM-FDB-TLABC algorithm, it was applied to solve the DED problem. Here, the 5-unit test system was considered. The data for the 5-unit test system and the *B*-coefficients are taken from [4]. Two DED cases on the 5-unit test system are considered, and they are explained as follows:

- Case-1: The 5-unit test system considering transmission losses.
- Case-2: The 5-unit test system considering transmission losses and POZs.

These case studies were solved by the proposed NSM-FDB-TLABC, NSM-TLABC, FDB-TLABC, EMA, EO, and FLA algorithms.

1) Case-1: The 5-unit test system considering transmission losses

In this case, the proposed algorithm was implemented on the 5-unit test system with transmission losses. The optimal solutions obtained from the NSM-FDB-TLABC algorithm are presented in Table 3. The statistical results of the Case-1 including the minimum (min), mean, standard deviation (std), and maximum (max) of the results for NSM-FDB-TLABC, NSM-TLABC, TLABC, EMA, FLA, EO, and the results reported in the literature are presented in Table 4. Accordingly, the proposed algorithm obtained 43044.0111$/h, which was lower 16.6515%, 16.6464%, 15.7955%, 16.3128%, 16.3354%, 0.1694%, 0.2146%, 0.1081%, 0.1886%, and 0.0093% than the NSM-TLABC, FDB-TLABC, EMA, FLA, EO, ICA [1], TVAC-IPSO [2], SOS [3], HIGA [5], and MBF-SSO [6], respectively. However, only the MGDE [9] algorithm achieved 0.0835% better results than the proposed algorithm. On the other hand, according to the mean values of the all algorithms, the proposed algorithm achieved the best mean value among them.



(a)

(b)

Fig. 2. For Case-1: (a) Convergence curves and (b) box-plots of the algorithms

TABLE III
OPTIMAL SOLUTIONS OF THE CASE-1 OBTAINED FROM NSM-FDB-TLABC ALGORITHM

Hour	P_1 (MW)	P_2 (MW)	P_3 (MW)	P_4 (MW)	P_5 (MW)	Cost ($/h)
1	20.6717	98.5580	30.0000	124.9642	139.6215	1251.2715
2	10.0029	97.6376	66.9647	124.6689	139.8493	1427.1847
3	10.0041	98.9542	106.9202	124.0000	139.9037	1400.2080
4	10.4024	98.8530	112.9200	174.0000	139.8349	1665.7307
5	10.0052	92.6691	112.7733	209.8889	139.4195	1590.4788
6	10.0088	98.7799	112.7703	209.9633	184.4699	1875.0228
7	10.0017	72.3517	112.6499	209.8913	229.5653	1841.8776
8	12.7471	98.3541	112.6477	209.9465	229.5623	1799.7490
9	42.6628	105.3829	112.7129	209.8933	229.5483	2012.7303
10	64.0264	98.3526	112.6859	209.9241	229.5703	1998.7705
11	75.0000	103.7549	112.8779	209.8840	229.5263	2038.1223
12	75.0000	124.5498	112.7418	209.8887	229.5391	2180.5800
13	64.0517	98.2066	112.8542	209.8696	229.5759	2000.2273
14	49.0088	98.7477	112.9922	209.8925	229.5272	1981.2869
15	35.8566	98.7871	112.8989	186.0012	229.5801	2015.8971
16	10.0069	98.5840	112.6975	136.3794	229.5655	1682.8253
17	10.0066	87.6042	112.6405	124.8909	229.5406	1615.6752
18	10.0028	98.5468	112.6697	165.1964	229.5353	1853.6463
19	12.6961	98.3982	112.7147	209.9050	229.5435	1798.9801
20	42.5314	120.0242	112.6369	209.8875	229.5784	2116.7549
21	39.1224	98.5533	112.7461	209.9103	229.5701	1945.7246
22	10.0035	98.4763	112.7261	209.8855	181.8219	1865.0050
23	10.0049	98.8733	112.8114	171.9523	139.2987	1661.9658
24	10.0047	80.3089	112.8588	124.6164	139.6983	1424.2963
					Total Cost ($/h)	**43044.0111**

TABLE IV
THE STATISTICAL RESULTS OF THE CASE STUDIES

	Min.	Mean	Max.	Std.
NSM-FDB-TLABC	43044.0111	**43051.6125**	43060.914	4.6128
NSM-TLABC	51643.4121	52576.7265	53365.3516	401.7314
FDB-TLABC	51640.266	52447.641	53342.9554	396.7994
EMA	51118.4465	52467.1522	53364.3619	503.6893
FLA	51434.3969	52651.9171	53882.4227	651.6330
EO	51448.3000	52573.5138	53314.6128	511.5934
ICA [1]	43117.055	43144.472	43209.533	*NR*
TVAC-IPSO [2]	43136.561	43185.664	43302.233	*NR*
SOS [3]	43090.5925	43103.0828	43162.2146	*NR*
HIGA [5]	43125.365	43162.243	43259.352	*NR*
MBF-SSO [6]	43048	43068	43093	*NR*
MGDE [9]	**43008.1049**	43084.9049	43403.2808	98.5234

NR : Not reported

TABLE V
OPTIMAL SOLUTIONS OF THE CASE-2 OBTAINED FROM NSM-FDB-TLABC ALGORITHM

Hour	P_1 (MW)	P_2 (MW)	P_3 (MW)	P_4 (MW)	P_5 (MW)	Cost ($/h)
1	16.3763	98.5766	30.0003	40.0007	229.1750	1252.5526
2	10.1468	98.7272	30.0005	71.1344	229.5246	1442.7911
3	10.0217	90.1248	30.0006	120.6099	229.5155	1439.2974
4	13.2421	98.5321	70.0003	124.9104	229.5461	1663.5090
5	10.0006	90.2485	110.0000	124.9309	229.5169	1617.8055
6	40.0001	99.9153	112.6257	133.7977	229.5203	1838.3339
7	10.0006	98.5453	112.6350	183.7394	229.5281	1865.7661
8	12.5632	98.6481	112.7112	209.8150	229.5208	1797.4928
9	42.5218	105.5390	112.7582	209.8615	229.5201	2013.2808
10	64.0313	98.5236	112.6322	209.8530	229.5196	1997.1436
11	75.0000	102.4267	114.2132	209.8632	229.5298	2038.7198
12	74.9480	124.7350	112.6653	209.8494	229.5224	2180.5455
13	64.1408	98.4593	112.6663	209.7677	229.5249	1997.1759
14	49.6293	98.5494	112.6301	209.8431	229.5167	1978.1822
15	35.4353	98.5362	112.6669	186.9641	229.5247	2007.7227
16	10.0006	98.0600	112.6616	136.9909	229.5188	1688.4631
17	10.0008	92.5674	107.6671	124.9470	229.5278	1619.5258
18	15.0380	98.5555	112.8514	159.9568	229.5246	1868.8543
19	12.7430	98.5366	112.6403	209.8214	229.5164	1797.6383
20	42.7171	118.0164	114.5427	209.8440	229.5204	2119.3944
21	39.3619	98.5553	112.6340	209.8284	229.5223	1945.0432
22	12.0809	98.5823	112.6513	160.0007	229.5458	1849.2003
23	10.0005	96.0897	72.6795	124.8716	229.5120	1656.6863
24	10.0004	70.8416	32.6953	124.9179	229.5222	1477.6292
					Total Cost ($/h)	**43152.7537**

The convergence curves and the box-plot of all algorithms for Case-1 are presented in Fig. 2. According to Fig. 2 (a), it was clearly seen that the proposed algorithm converged to the best fitness value compared to the rivals. From Fig. 2 (b), the proposed algorithm obtained the best minimum and median values against the others and showed stable search performance among all algorithms.

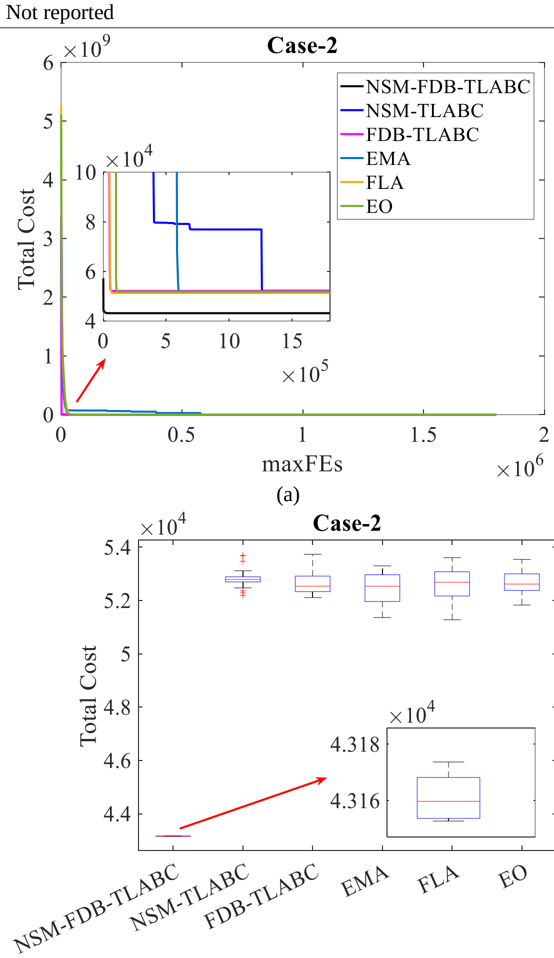
2) Case-2: The 5-unit test system considering transmission losses and POZs

Here, the proposed algorithm was implemented on the 5-unit test system with transmission losses and POZs of the generating units. The optimal solutions obtained from the NSM-FDB-TLABC algorithm are presented in Table 5. In Table 6, the statistical results of the case studies for all algorithms are presented. From Table 3, the total cost value obtained from the NSM-FDB-TLABC was 43152.7537$/h, which was lower 17.3067%, 17.1919%, 15.9840%, 15.8453%, 16.7436%, 0.0538%, and 0.0734% than the NSM-TLABC, FDB-TLABC, EMA, FLA, EO, MBF-SSO [6], and MGDE [9] algorithms, respectively. Besides, the proposed algorithm achieved the best mean value among them.

TABLE VI
THE STATISTICAL RESULTS OF THE CASE STUDIES

	Min.	Mean	Max.	Std.
NSM-FDB-TLABC	**43152.7537**	**43161.3386**	43173.6983	8.1836
NSM-TLABC	52184.1304	52819.029	53688.8109	339.971
FDB-TLABC	52111.7365	52665.5418	53737.1936	436.3525
EMA	51362.5670	52421.6731	53301.1135	531.7966
FLA	51277.8689	52605.8738	53610.1305	606.9064
EO	51831.1520	52710.1195	53545.2565	480.5690
MBF-SSO [6]	43176	*NR*	*NR*	*NR*
MGDE [9]	43184.4654	43280.8562	43461.7934	90.8574

NR : Not reported



(a)

(b)

Fig. 3. For Case-2: (a) Convergence curves and (b) box-plots of the algorithms

The convergence curves and the box-plot of all algorithms for Case-2 are presented in Fig. 3. From Fig. 3 (a), it can be seen that the proposed algorithm converged to the best objective function value among all algorithms. On the other hand, when the box-plots of the algorithms given in Fig. 3 (b) were analyzed, the proposed algorithm achieved a successful search performance against its rivals.

V. CONCLUSION

In this study, the NSM-FDB-TLABC algorithm was presented to the literature as a competitive and powerful algorithm by combining the strengths of the NSM-TLABC and FDB-TLABC algorithms. In the NSM-FDB-TLABC algorithm, the updating mechanism of the teaching-based employed bee stage was redesigned using the NSM-based method as in the NSM-TLABC, and the guide individual in the learning-based employed bee stage was chosen by the FDB-based method as in the FDB-TLABC. One of the most important points of the study was that a comprehensive simulation study was carried out to verify the performance of the NSM-FDB-TLABC algorithm. In the first simulation study, the proposed algorithm and its five rivals were applied to solve the CEC2017 and CEC2020 benchmark problems. The results were analyzed using the Friedman test, Wilcoxon test, and convergence analysis. According to Friedman test results, the proposed algorithm ranked first in terms of mean rank value with 2.2836. In second simulation study, the proposed algorithm and its five rivals were applied to solve two DED case studies. According to their results, the proposed algorithm achieved the best optimal solutions among them. On the other hand, the results of the proposed algorithm were compared with the results reported in the literature. Based on the results of them, the proposed algorithm obtained the best optimal solutions for both case studies. The other important point of the study was that the best optimal solutions for two case studies were presented to the literature. To sum up, the supremacy of the proposed algorithm was proven on both the DED problem and benchmark suites.

In future studies, hybrid uses of the FDB and NSM methods used in this study will be tested on different meta-heuristic algorithms. Additionally, the NSM-FDB-TLABC algorithm presented in the study will be applied to solve different real-world engineering problems.

REFERENCES

[1] B. Mohammadi-Ivatloo, A. Rabiee, A. Soroudi, M. Ehsan. "Imperialist competitive algorithm for solving non-convex dynamic economic power dispatch." Energy, vol. 44. 1, 2012, pp 228-240.

[2] B. Mohammadi-Ivatloo, A. Rabiee, M. Ehsan. "Time-varying acceleration coefficients IPSO for solving dynamic economic dispatch with non-smooth cost function." Energy conversion and management, vol. 56, 2012, pp 175-183.

[3] Y. Sonmez, H. T. Kahraman, M. K. Dosoglu, U. Guvenc, S. Duman. "Symbiotic organisms search algorithm for dynamic economic dispatch with valve-point effects." Journal of Experimental & Theoretical Artificial Intelligence, vol. 29. 3, 2017, pp 495-515.

[4] C. Dai, Z. Hu, Q. Su. "An adaptive hybrid backtracking search optimization algorithm for dynamic economic dispatch with valve-point effects." Energy, vol. 239, 2022, 122461.

[5] B. Mohammadi-Ivatloo, A. Rabiee, A. Soroudi. "Nonconvex dynamic economic power dispatch problems solution using hybrid immune-genetic algorithm." IEEE Systems Journal, vol. 7.4, 2013, pp 777-785.

[6] R. Azizipanah-Abarghooee. "A new hybrid bacterial foraging and simplified swarm optimization algorithm for practical optimal dynamic load dispatch." International Journal of Electrical Power & Energy Systems, vol. 49, 2013, pp 414-429.

[7] Y. Zhang, D. W. Gong, N. Geng, X. Y. Sun."Hybrid bare-bones PSO for dynamic economic dispatch with valve-point effects." Applied Soft Computing, vol. 18, 2014, pp 248-260.

[8] G. Xiong, D. Shi, "Hybrid biogeography-based optimization with brain storm optimization for non-convex dynamic economic dispatch with valve-point effects." Energy, vol. 157, 2018, pp 424-435.

[9] D. Zou, S. Li, X. Kong, H. Ouyang, Z. Li, "Solving the dynamic economic dispatch by a memory-based global differential evolution and a repair technique of constraint handling." Energy, vol. 147, 2018, pp 59-80.

[10] M. Ghasemi, E. Akbari, M. Zand, M. Hadipour, S. Ghavidel, L. Li. "An efficient modified HPSO-TVAC-based dynamic economic dispatch of generating units." Electric Power Components and Systems, vol. 47. 19-20, 2019, pp 1826-1840.

[11] Z. Zheng, J. Li, Y. Han. "An improved invasive weed optimization algorithm for solving dynamic economic dispatch problems with valve-point effects." Journal of Experimental & Theoretical Artificial Intelligence, vol. 32. 5, 2020, pp 805-829.

[12] D. Santra, A. Mukherjee, K. Sarker, S. Mondal, "Dynamic economic dispatch using hybrid metaheuristics." Journal of Electrical Systems and Information Technology, vol. 7, 2020, pp 1-30.

[13] W. Yang, Z. Peng, Z. Yang, Y. Guo, X. Chen, "An enhanced exploratory whale optimization algorithm for dynamic economic dispatch." Energy Reports, vol. 7, 2021, pp 7015-7029.

[14] Z. Hu, C. Dai, Q. Su, "Adaptive backtracking search optimization algorithm with a dual-learning strategy for dynamic economic dispatch with valve-point effects." Energy, vol. 248, 2022, 123558.

[15] S. Basak, B. Bhattacharyya, B. Dey, "Dynamic economic dispatch using hybrid CSAJAYA algorithm considering ramp rates and diverse wind profiles." Intelligent Systems with Applications, vol. 16, 2022, 200116.

[16] W. Yang, X. Zhu, F. Nie, H. Jiao, Q. Xiao, Z. Yang, "Chaos Moth Flame Algorithm for Multi-Objective Dynamic Economic Dispatch Integrating with Plug-In Electric Vehicles." Electronics, vol. 12. 12, 2023, 2742.

[17] W. Yang, Y. Zhang, X. Zhu, K. Li, Z. Yang, "Research on Dynamic Economic Dispatch Optimization Problem Based on Improved Grey Wolf Algorithm." Energies, vol. 17. 6, 2024, 1491.

[18] K. Nagarajan, A. Rajagopalan, M. Bajaj, R. Sitharthan, S. A. Dost Mohammadi, V. Blazek, Optimizing dynamic economic dispatch through an enhanced Cheetah-inspired algorithm for integrated renewable energy and demand-side management. Scientific Reports, vol. 14. 1, 2024, 3091.

[19] S. Duman, H. T. Kahraman, Y. Sonmez, U. Guvenc, M. Kati, S. Aras. "A powerful meta-heuristic search algorithm for solving global optimization and real-world solar photovoltaic parameter estimation problems." Engineering Applications of Artificial Intelligence, vol. 111, 2022, 104763.

[20] H. T. Kahraman, M. Katı, S. Aras, D. A. Taşci. "Development of the Natural Survivor Method (NSM) for designing an updating mechanism in metaheuristic search algorithms." Engineering Applications of Artificial Intelligence, vol. 122, 2023, 106121.

[21] X. Chen, B. Xu, C. Mei, Y. Ding, K. Li. "Teaching–learning–based artificial bee colony for solar photovoltaic parameter estimation." Applied Energy, vol. 212, 2018, pp 1578-1588.

[22] H. T. Kahraman, S. Aras, E. Gedikli. "Fitness-distance balance (FDB): a new selection method for meta-heuristic search algorithms." Knowledge-Based Systems, vol. 190, 2020, 105169.

[23] N. H. Awad, M. Z. Ali, J. J. Liang, B. Y. Qu, P. N. Suganthan."Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective bound constrained real-parameter numerical optimization." Technical Report, 2016.

[24] C. T. Yue, K. V. Price, P. N. Suganthan, J. J. Liang, M. Z. Ali, B. Y. Qu,... , P. P. Biswas. "Problem definitions and evaluation criteria for the CEC 2020 special session and competition on single objective bound constrained numerical optimization." Technical Report, 2019.

[25] F. A. Hashim, R. R. Mostafa, A. G. Hussien, S. Mirjalili, K. M. Sallam. "Fick's Law Algorithm: A physical law-based algorithm for numerical optimization." Knowledge-Based Systems, vol. 260, 2023, 110146.

[26] M. H. Sulaiman, Z. Mustaffa, M. M. Saari, H. Daniyal, S. Mirjalili. "Evolutionary mating algorithm. "Neural Computing and Applications, vol. 35. 1, 2023, pp 487-516.

[27] A. Faramarzi, M. Heidarinejad, B. Stephens, S. Mirjalili. "Equilibrium optimizer: A novel optimization algorithm." Knowledge-based systems, vol. 191, 2020, 105190.

BIOGRAPHY

Burçin Özkaya obtained his BSc degree in Electrical and Electronics Engineering from Eskişehir Osmangazi University (ESOGU) in 2014. She received MSc. in Electrical and Electronics Engineering from the Süleyman Demirel University in 2018, and PhD degree in Electrical and Electronics Engineering from Düzce University in 2022. She worked as a Research Assistant with the Department of Electrical and Electronics Engineering at Suleyman Demirel University from 2015 to 2018, and with Department of Electrical and Electronics Engineering at Isparta University of Applied Sciences from 2018 to April 2023. She has been working as an Assistant Professor since April 2023 at Department of Electrical Engineering at Bandırma Onyedi Eylül University. Her research interests include power systems applications, optimization, meta-heuristic algorithms, and artificial intelligence.