Applied Intelligence
https://doi.org/10.1007/s10489-021-02629-3

Dynamic FDB selection method and its application: modeling and optimizing of directional overcurrent relays coordination

Hamdi Tolga Kahraman[1] · Huseyin Bakir[2] · Serhat Duman[3] · Mehmet Katı[4] · Sefa ARAS[1] · Ugur Guvenc[5]

Accepted: 17 June 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract
This article has four main objectives. These are: to develop the *dynamic* fitness-distance balance (*d*FDB) selection method for meta-heuristic search algorithms, to develop a strong optimization algorithm using the *d*FDB method, to create an optimization model of the coordination of directional overcurrent relays (DOCRs) problem, and to optimize the DOCRs problem using the developed algorithm, respectively. A comprehensive experimental study was conducted to analyze the performance of the developed *d*FDB selection method and to evaluate the optimization results of the DOCRs problem. Experimental studies were carried out in two steps. In the first step, to test the performance of the developed *d*FDB method and optimization algorithm, studies were conducted on three different benchmark test suites consisting of different problem types and dimensions. The data obtained from the experimental studies were analyzed using non-parametric statistical methods and the most effective among the developed optimization algorithms was determined. In the second step, the DOCRs problem was optimized using the developed algorithm. The performance of the proposed method for the solution to the DOCRs coordination problem was evaluated on five test systems including the IEEE 3-bus, the IEEE 4-bus, the 8-bus, the 9-bus, and the IEEE 30-bus test systems. The numerical results of the developed algorithm were compared with previously proposed algorithms available in the literature. Simulation results showed the effectiveness of the proposed method in minimizing the relay operating time for the optimal coordination of DOCRs.

Keywords Meta-heuristic search (MHS) · Dynamic fitness distance balance (*d*FDB) · Manta ray foraging optimization (MRFO) · Directional overcurrent relays (DOCRs) coordination · Power system optimization · Power system protection

1 Introduction

Optimizing non-smooth, non-convex, and high-dimensional type real-world problems is a major challenge [1, 2]. Meta-heuristic search (MHS) algorithms are frequently used in solving such problems. However, the performance of the MHS algorithms varies depending on the type and complexity of the optimization problem. This makes it necessary to develop the most suitable MHS algorithm for each real-world engineering problem and this requires extensive research. In this article, a comprehensive research and development study was carried out for optimization of the directional overcurrent relays (DOCRs) coordination problem and includes studies on MHS algorithms and the DOCRs coordination problem, respectively.

✉ Hamdi Tolga Kahraman
htolgakahraman@ktu.edu.tr

Huseyin Bakir
hbakir@dogus.edu.tr

Serhat Duman
sduman@bandirma.edu.tr

Mehmet Katı
mkati@havelsan.com.tr

Sefa ARAS
sefaaras@ktu.edu.tr

Ugur Guvenc
ugurguvenc@duzce.edu.tr

[1] Software Engineering, OF Technology Faculty, Karadeniz Technical University, 61080 Trabzon, Turkey

[2] Department of Mechatronics, Dogus Vocational School of Higher Education, Dogus University, 34775 Istanbul, Turkey

[3] Electrical Engineering, Engineering and Natural Sciences Faculty, Bandirma Onyedi Eylul University, 10200 Bandirma, Turkey

[4] HAVELSAN, Ankara, Turkey

[5] Electrical and Electronics Engineering, Technology Faculty, Duzce University, 81620 Duzce, Turkey

Published online: 29 July 2021

In this article, two studies were conducted on MHS algorithms. The subject of the first study was the development of the selection method used in MHS algorithm design, which has a significant impact on performance. For this purpose, research was conducted to improve the selection method called fitness-distance balance (FDB) [1]. As a result, a powerful selection method called *dynamic* fitness-distance balance (*d*FDB) was developed that can be used in the design of MHS algorithms. The most important feature of this *d*FDB is that it has a dynamically changing weight parameter. In this way, it has gained the capability of adapting effectively to the search spaces of different types of problems. The subject of the second study on MHS algorithms was the application of the *d*FDB method in the up-to-date MHS algorithm known as the manta ray foraging optimization (MRFO) algorithm [3], and the improving of its performance. As a result of the studies carried out for this purpose, an improved MRFO algorithm called *d*FDB-MRFO was very successfully developed. Thus, the studies on MHS algorithms were completed.

In this article, two studies were conducted on the DOCRs coordination problem. The subject of the first study was the modeling of the DOCRs coordination problem in order to solve it using optimization algorithms. The aim of the model was to find the optimal values of the control parameters for relays subject to various constraints, such as coordination and boundary limits. The DOCRs have two settings, called control variables: the time dial setting (TDS) and the pickup current setting (PCS) [4]. These parameters directly affect the operating time of the relay [5]. In the DOCRs coordination problem, the coordination time interval (CTI), which ensures the reliability of the protection system, must be considered [6]. In other words, the backup relay should operate in cases when the primary relay fails to take the appropriate action [7]. The subject of the second study was the optimization of the model developed for the DOCRs coordination problem. In the literature, many studies have been carried out to solve the DOCRs coordination problem. In these studies, the DOCRs coordination problem was designed as an optimization problem whose objective function was to minimize the operating time of all the primary relays [8]. Nature-inspired optimization techniques have been used to find the optimal solution values of the DOCRs coordination problem in modern power systems. These include: the particle swarm optimization (PSO) [9], genetic algorithm (GA) [10], ant colony algorithm [11], non-dominated sorting genetic algorithm-II (NSGA-II) [12], seeker optimization algorithm (SOA) [13], teaching learning-based optimization algorithm (TLBO) [14], adaptive differential evolution (ADE) algorithm [15], opposition-based chaotic differential evolution (OCDE) algorithm [16], biogeography-based optimization (BBO) [17], gray wolf optimizer (GWO [18], harmony search algorithm [19], modified water cycle algorithm (MWCA) [6], stochastic fractal search algorithm (SFSA) [20], firefly algorithm (FA) [21], artificial bee colony (ABC) [22], flower pollination algorithm (FPA) [23], symbiotic organisms search (SOS) optimization [24], improved invasive weed optimization (IIWO) [25], and Hyper-Sphere Search (HSS) algorithm [26]. In addition, the problem has been solved with hybrid algorithms such as the hybrid whale optimization algorithm and gray wolf optimizer (HWGO) [27], hybridized whale optimization algorithm (HWOA) [28], hybrid gravitational search algorithm-sequential quadratic programming (GSA-SQP) [29], and hybrid PSO-DE algorithm [30].

In this article, there are reasons why the MRFO algorithm was chosen to test the dynamic FDB method and validate its performance. Why was the MRFO method chosen, although there are hundreds of meta-heuristic search methods in the literature? Because extensive research has been done and it has been realized that MRFO has strong foundations. MRFO is one of the most up-to-date MHS algorithms whose design has not been improved yet. Moreover, it was realized that by using the dynamic FDB method, the foraging strategies of the MRFO algorithm could be redesigned, thereby improving the algorithm's ability to mimic nature.

Why was the DOCRs coordination problem chosen, although there are dozens of engineering optimization problems in the literature? There are many studies in the literature on the optimization of DOCRs. Because DOCRs coordination is one of the most popular constrained real-world engineering problems, which is difficult to find the optimum solution. DOCRs coordination problem is one of the most appropriate problems to verify the performance of the *d*FDB-MRFO algorithm developed in this article because there are many competing studies.

The main contributions of this paper can be summarized as follows:

- A new selection method, called *dynamic* fitness-distance balance, has been developed that can be used in the design of meta-heuristic search algorithms.
- Thanks to the *dynamic* fitness-distance balance method, the capability of meta-heuristic search algorithms to adapt to the optimization process has been improved. This improvement has been proven by studies on manta ray foraging optimization, a recently developed meta-heuristic search algorithm.
- A new algorithm, the *dynamic* fitness-distance balance-manta ray foraging optimization, was developed, which gives a superior performance in the optimization of the directional overcurrent relays coordination problem compared to competing algorithms.
- The superiority of the proposed *dynamic* fitness-distance balance-manta ray foraging optimization algorithm over its competitors has been proved by using different standard test systems (the IEEE 3-bus, IEEE 4-bus, 8-bus, 9-bus, and IEEE 30-bus test systems).
- By using the proposed method, the total operation times of all primary relays are minimized by providing coordination criteria and control variable limits.

- Because of its dynamic adaptation capability, the proposed *dynamic* fitness-distance balance-manta ray foraging optimization is a powerful method that can be used by researchers to optimize real-world applications and continuous and constrained engineering design problems.

The article consists of six main sections with subsections. Section 2 presents the coordination problem of DOCRs and the objective function. In Section 3, topics related to the meta-heuristic optimization method are introduced. The first three of these subsections describe the basics of the meta-heuristic optimization process, the MRFO algorithm, and the selection methods used in the design of the MHS algorithms, respectively. The other two subsections describe the *d*FDB selection method and the *d*FDB-MRFO optimization algorithm developed in this article and recommended to the literature, respectively. In Section 4, the standards and settings taken into consideration in conducting experimental studies are introduced. Information is given about the benchmark test suites used in the performance analysis of the *d*FDB selection method and the *d*FDB-MRFO algorithm developed in the article. In Section 5, experimental study results are analyzed. The results of two different experimental studies are presented in two separate subsections. These two subsections should be read carefully. In the first, the results obtained from the experimental studies of the *d*FDB selection method and the *d*FDB-MRFO algorithm developed in the article are analyzed. In the second, the results obtained from the experimental studies carried out for the optimization of the DOCRs coordination problem are analyzed. In this subsection, the performances of the *d*FDB-MRFO algorithm and its competitors are compared. In both subsections, nonparametric test and analysis methods are used to evaluate the experimental data. Finally, Section 6 provides information about the conclusions and future work.

2 Mathematical model of directional overcurrent relays coordination problem

In the DOCRs coordination problem, each relay must guarantee primary protection in its area (line) and backup protection for all adjacent lines. The primary relay should only trigger when there is a fault in the protection zone (line). When the primary relay fails to clear the fault, the backup relay should be activated after the time delay. In this problem, the fault currents for the primary and backup relay pairs are predetermined [29]. Near-end, far-end and mid-point fault currents are used in test systems.

A simple network with primary/backup relay pairs for a near-end fault (F1) and far-end fault (F2) is shown in Fig. 1 [31]. For the system in Fig. 1, the primary/backup relay pairs for F1 and F2 faults are R1/R5, R1/R4, and R2/R8.

The DOCRs coordination problem is to determine the sequences of relay operations for each predetermined fault location so that the faulted line is isolated with sufficient coordination margins [31]. The coordination of DOCRs is a non-smooth, non-convex, and high-dimension optimization problem and it should be solved subject to various constraints, such as coordination and boundary limits.

2.1 Objective function

In this paper, the objective function (OF) constitutes the summation of operating times of all primary relays [32]:

$$\min_{TDS_i, PCS_i} \text{OF} = \sum_{i=1}^{m} \sum_{k} T_{ik} \tag{1}$$

where TDS_i and PCS_i control variables represent the time dial setting and the pickup current setting of the ith relay, respectively. m is the number of relays in the network and T_{ik} is the operating time (OT) of the ith relay for a fault at k. The OT of the relay is a function of TDS_i, PCS_i and fault current (I_f) and is expressed by the nonlinear equation shown below:

$$T_{ik} = \frac{a.TDS_i}{\left(\frac{I_{ik}}{PCS_i}\right)^{b} - c} \tag{2}$$

where I_{ik} is the fault current seen by the ith relay for a fault at k and a, b, and c are constant values that indicate the relay characteristic given as 0.14, 0.02, and 1.0, respectively, according to the International Electrotechnical Commission (IEC) 60,255–151 standards. It is assumed that inverse-definite minimum time (IDMT) OCRs are being used.

Each relay is connected to the line through a current transformer (CT) to reduce the current level. The primary rating of the CT is known in the problem. Therefore, the fault current seen by the relay (I_{ik}) is found as a ratio of the fault current at the CT primary terminals ($I_{f,\,ik}$) and the primary rating of the CT, which is as follows:

$$I_{ik} = \frac{I_{f,ik}}{CT_{rating}} \tag{3}$$

2.2 Constraints of problem

The bounds of the control variables are as follows:

$$TDS_i^{\,l} \le TDS_i \le TDS_i^{\,u} \qquad i = 1, \ldots\ldots, m \tag{4}$$

$$PCS_i^{\,l} \le PCS_i \le PCS_i^{\,u} \qquad i = 1, \ldots\ldots, m \tag{5}$$

where $TDS_i^{\,l}$and $TDS_i^{\,u}$ represent the lower and upper bounds of the time dial setting for ith relay. $PCS_i^{\,l}$ and $PCS_i^{\,u}$are the lower and upper limits of pickup current setting for the ith relay, respectively. The operation time of ith relay at fault k

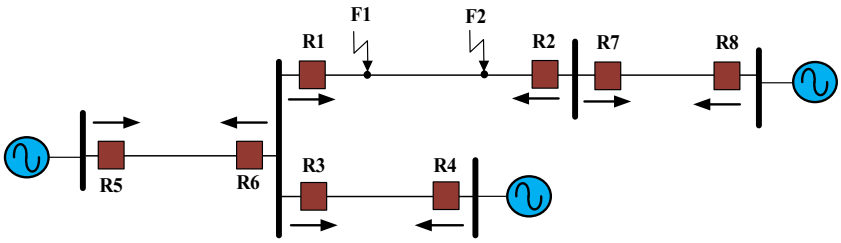


Fig. 1 Simple network with primary/backup relay pairs

is defined as follows, and should be set between 0.05 s and 1.0 s:

$$T_{ik}{}^{l} \le T_{ik} \le T_{ik}{}^{u} \qquad i = 1, \ldots\ldots, m \tag{6}$$

The backup relay operates only when the corresponding primary relay fails. To ensure relay coordination, the operation time of the backup relay must be greater than the corresponding primary relay for all the faults. The operation of the backup relay was restricted to operate after a time interval called CTI to avoid miss-coordination between the primary and backup relay and can be formulated as follows: [31]:

$$CTI = T_{jk} - T_{ik} \qquad i = 1, \ldots\ldots, m \tag{7}$$

where T_{jk} is the operation time of the jth backup relay for a fault at k inside the zone protected by the ith primary relay. Therefore, the CTI coordination criteria can be stated as follows:

$$CTI \ge CTI_{min} \tag{8}$$

In Eq. (8) CTI_{min} is the minimum coordination time interval. The typical value of $CTI_{min}$is 0.2–0.3 s [15]. Figure 2 depicts an example of the CTI constraints for primary and backup relays in case of F1 (near-end) and F2 (far-end) faults.

2.3 Modification of objective function

In spite of the fact that the CTI constraint between the primary and backup relay pairs is fulfilled for suitable selectivity, much-delayed operation of backup relays is not desired for effective relay coordination. For this reason, the objective function in Eq. (1) has been modified as follows to optimize the CTI [31]:

$$\min_{TDS_i, PCS_i} \mathrm{MOF} = \alpha_1 \sum_{i=1}^{m} \sum_{k} T_{ik}{}^{2} + \alpha_2 \sum_{p=1}^{m_p} \left[\Delta T_{mbp} - \beta \left(\Delta T_{mbp} - \left|\Delta T_{mbp}\right|\right)\right]^2 \tag{9}$$

$$\Delta T_{mbp} = T_{jk} - T_{ik} - CTI_{min} \tag{10}$$

where ΔT_{mbp} is the operation time difference with pth relay pair between CTI_{min}, m_p is the number of primary/backup relay pairs, α_1, and α_2 are weight factors, and β is used to consider the miscoordination.

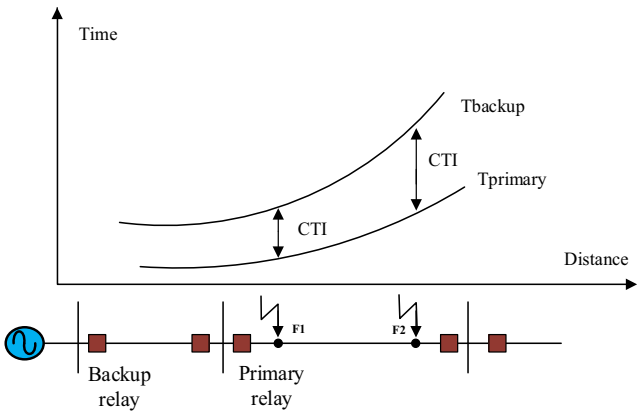


Fig. 2 Coordination between primary and backup relays

3 Method

The preparation of this section included two main purposes. The first was to introduce the developed *d*FDB method. The second was to develop a powerful MHS algorithm for optimization of the DOCRs coordination problem. For this, a recently developed MHS method, the MRFO [3] was used. Improved variants of the MRFO algorithm were developed using the proposed *d*FDB method. In order to facilitate the understanding of the *d*FDB selection method, subsections were prepared describing the basics of the meta-heuristic optimization process. In this way, researchers can learn how to apply the *d*FDB method to MHS algorithms to improve their search performance and develop stronger optimization algorithms for their own studies.

This section consists of five subsections. The first outlines the basics of the meta-heuristic search process and the key elements of constrained optimization problems. The second subsection presents the search process life cycle and an overview of the MRFO algorithm mathematical model. The third subsection describes the selection methods used in the MHS algorithms. In the fourth subsection, the *d*FDB selection method developed in this article is introduced step-by-step. In the fifth subsection, improved versions of the MRFO algorithm are developed using the proposed *d*FDB method and the

improved MRFO with *d*FDB is proposed for optimization of the DOCRs coordination problem.

3.1 Overview of meta-heuristic optimization process

An m-dimensional search space can be represented by the $X=[x_1, x_2, \cdots, x_m]$, $m \in N^+$, $\forall_{i=1}^{m} x_i \in X$. Where $x_1, x_2, \cdots, x_m$ are the design variables of the optimization problem. Their data type can be continuous, discrete or a mixture of the two. In the continuous search space, the bounds of design variables are represented by $[a, b]$, $a, b \in R$ and $-\infty < (a, b) < +\infty$. The objective function for this search problem can be represented by G, as given in Eq. (11) [2].

$$\underset{x \in R^n}{minimize/maximize}\ G = f(x_1, x_2, \ldots, x_m)$$
$$\text{Subject to} \quad \begin{aligned} &\varnothing_j(x) = 0, (j = 1, 2, \ldots, J), \\ &\varphi_k(x) \leq 0, (k = 1, 2, \ldots, K) \end{aligned} \tag{11}$$

Equation (11) represents the general form of constrained optimization problems; $\varnothing$ and φ represent equality and inequality constraints, respectively, of the optimization problem. The MHSs are algorithms that make parallel searches in the search space with n-solution candidates. Depending on the above definitions and the objective and the constraint functions given in Eq. (11), in a MHS algorithm, the community of solution candidates, i.e., the population, is represented as given in Eq. (12):

$$P \equiv \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix}_{nxm} \tag{12}$$

The fitness values of the solution candidates in the P are represented by the F-vector as given in Eq. (13):

$$F \equiv \begin{bmatrix} f_1 \\ . \\ . \\ . \\ f_n \end{bmatrix}_{nx1} \tag{13}$$

Algorithm 1. General steps of the search process in MHS algorithms [1]

1. **Begin (initialization)**
2. *P*: Create the population vector as given in Eq. (12)
3. **for** *i*=1: *n* (number of solution candidates)
4. *F*: Use the Eq. (11) and create the fitness vector as given in Eq. (13)
5. **end**
6. **while** (search process lifecycle)
7. Step 1: Selection:
   Selection of guides (reference positions) from the *P* that will determine the direction of the search process
8. Step 2: Search:
9. Exploitation (neighborhood search around guides)
10. Exploration (diversification in *P* using guides)
11. Step 3: Update:
    Update the *P*-population depending on the fitness values of solution candidates
12. **next generation until termination criterion**
13. **End**

The general steps for solving optimization problems using MHS algorithms are given in Algorithm 1. In the initial phase, the optimization model of a problem is created as given in Eq. (11). The number of design variables and their bounds, the settings of the MHS algorithm, and the population size are defined at this stage. The search process life cycle in an MHS algorithm basically consists of three steps, as given in Algorithm 1: selection, search, and update processes, respectively. The performance of a MHS algorithm in the optimization process mainly depends on its success in these three steps. Therefore, researchers are trying to develop more effective methods, stronger strategies, and innovative approaches, especially for the selection (Step 2) and search (Step 3) processes. In the next subsection, the MRFO algorithm is reviewed.

3.2 Overview of manta ray foraging optimization

First of all, MRFO is an effective bio-inspired algorithm that has steps similar to other MHSs [3]. Manta rays are among the largest sea creatures and can reach 7 m in width. They have a lifespan of about 20 years and feed on plankton (microscopic organisms). Manta rays are known to have been around for 5 million years and owe their survival skills to three powerful nutritional strategies they have developed: chain foraging, cyclone foraging, and somersault foraging. The manta rays that apply the chain search strategy are lined up one after another. Thus, plankton that the front manta ray cannot collect can be collected by those behind. The cyclone feeding strategy is used when the plankton concentration is very high. The manta rays gather in a spiral and carry the water to the surface with their tails. They float to the surface and collect the plankton by filtering the water through their mouths. In the somersault foraging strategy, the manta rays circle around the plankton and with this cyclical process they optimize their food intake.

3.2.1 Mathematical model of manta ray foraging optimization

The manta ray foraging optimization algorithm is different from other MHS algorithms because of the nutritional strategies that are essential in defining the search process life cycle. The other steps of the algorithm are the same as the general steps of the search process in the MHS algorithms given in Algorithm 1. After giving brief information about the MRFO, this section describes the mathematical models of the MRFO nutritional strategies that provide its exploitation and exploration abilities.

The first of the strategies used for the nutritional process in manta rays is chain foraging. The mathematical model of this strategy is given in Eqs. (14) and (15):

$$x_i^d(t+1) = \begin{Bmatrix} x_i^d(t) + r \cdot \left(x_{best}^d - x_i^d(t)\right) + \alpha \cdot \left(x_{best}^d - x_i^d(t)\right), i = 1 \\ x_i^d(t) + r \cdot \left(x_{i-1}^d(t) - x_i^d(t)\right) + \alpha \cdot \left(x_{best}^d - x_i^d(t)\right), i = 2, \ldots, N \end{Bmatrix} \tag{14}$$

$$\alpha = 2 \cdot r \cdot \sqrt{|\log(r)|} \tag{15}$$

As with many other swarm-based search algorithms, in the MRFO algorithm, candidates for solutions are represented by the position of the food. Suppose that the population of manta rays is represented by X. In this case, the *i*th individual in the population is also represented by x_i. If the size of the optimization problem (number of design parameters) is D, the position vector of a manta ray for the *d*th dimension at time *t* is represented by$X_i^d(t)$. The new position of a manta ray that moves according to the foraging chain strategy at time (t + 1) is calculated as given in Eq. (14). In this equation, *r* is a randomly generated vector in the range [0, 1], while α is the weight coefficient vector, and X_{best}^d is the position vector with the highest density of plankton in the manta ray population. The value of the weight coefficient is calculated as given in Eq. (15). Thus, three methods providing the exploitation and exploration capabilities of the MRFO algorithm are explained.

The second method used for nutrition in the MRFO algorithm is the cyclone foraging strategy. The characteristic feature of this strategy is that the manta rays form a spiral shape and swim towards the plankton in deep water. Equation (16) is used to model spiral motion in an *N*-dimensional search space:

$$x_i^d(t+1) = \begin{cases} x_{best}^d + r \cdot \left(x_{best}^d - x_i^d(t)\right) + \beta \cdot \left(x_{best}^d - x_i^d(t)\right), i = 1 \\ x_{best}^d + r \cdot \left(x_{i-1}^d(t) - x_i^d(t)\right) + \beta \cdot \left(x_{best}^d - x_i^d(t)\right), i = 2, \ldots, N \end{cases} \tag{16}$$

$$\beta = 2e^{r_1 \frac{T-t+1}{T}} \cdot \sin(2\pi r_1) \tag{17}$$

$$x_{rand}^d = Lb^d + r \cdot \left(Ub^d - Lb^d\right) \tag{18}$$

$$x_i^d(t+1) = \begin{cases} x_{rand}^d + r \cdot \left(x_{rand}^d - x_i^d(t)\right) + \beta \cdot \left(x_{rand}^d - x_i^d(t)\right), i = 1 \\ x_{rand}^d + r \cdot \left(x_{i-1}^d(t) - x_i^d(t)\right) + \beta \cdot \left(x_{rand}^d - x_i^d(t)\right), i = 2, \ldots, N \end{cases} \tag{19}$$

where β is the weight coefficient, *T* is the maximum number of iterations, and r_1 is a random number between [0, 1]. In Eq. (18), a random vector is created between the bound values for the *d*-dimension of the problem. Random vectors are used to explore in the search space and to increase diversity. In the MRFO algorithm, exploration capability is provided by Eq. (19). The *d*-dimension of the *i*th solution candidate is generated randomly as given in Eq. (19).

The third method used for nutrition in the MRFO algorithm is the somersault foraging strategy. In this strategy, the position of the food source is considered the pivot. Manta rays turn around this pivot to try to achieve a better position.

$$x_i^d(t+1) = x_i^d(t) + S \cdot (r_2 . x_{best} - r_3 \cdot x_i(t)), i = 1, 2, \ldots, \mathrm{N} \tag{20}$$

In Eq. (20), *S* is the somersault factor *S* = 2, r_2 and r_3 are random numbers generated between [0, 1]. Accordingly, manta rays search between their position and the best manta ray position to find better plankton positions. For this reason, the somersault foraging strategy is used to create a fine convergence effect (fine-tune) in the MRFO algorithm.

3.3 Overview of selection methods

Selection methods play an important role in determining the reference positions that guide the search process of MHS algorithms in establishing the balance between exploitation-exploration capabilities, and in determining the performance of the algorithms [1]. They are used in the first step of the MHS process life cycle (see Algorithm 1). It is possible to group these methods in three categories: the greedy, the probabilistic (such as the roulette wheel and tournament methods), and the random selection methods. In the greedy selection method, the solution candidates in the population are ranked according to their fitness values and the candidate with the highest fitness value is selected. In the probabilistic selection method, the probability of candidates to be selected from the population depends on their fitness values. In the random selection method, individuals are selected completely randomly from within the population. Information about the methods used in the selection of solution candidates guiding the search process in MHS algorithms is given in Table 1.

Thirty-four MHS algorithms were reviewed in the preparation of Table 1. In the first, second, and third columns of Table 1, respectively, the selection method abbreviation, selection method description, and MHS algorithm references using this selection method are given. According to this, selection methods used in MHS algorithms can be classified into four categories. The ordinal-based selection method was used in only one of the thirty-four MHS algorithms, whereas the probabilistic selection method was applied in only three algorithms. It is clear that the ordinal-based method was used to make a regular search within the population. The greedy selection method was used in 28 of the 34 studies to increase the impact of successful solution candidates in the search process, while the random selection method was used in 24 of these studies in order to increase diversity in the search process.

3.4 Proposed selection method: *dynamic* fitness-distance balance

The *d*FDB method is an improved version of the FDB. The FDB is a selection method developed to provide more effective guidance to the search process in population-based MHS algorithms [1]. The FDB is also a greedy method in which solution candidates are selected from the population according to their scores. Fitness and distance values are used in calculating the score value of a solution candidate. Accordingly, fitness values of solution candidates in a *P*-population are represented by the *F*-vector, as given in Eq. (13). Suppose that

Table 1 Methods used in the selection of solution candidates in MHS algorithms

Selection method	Definition	Algorithms using the method
Random	Solution candidates are randomly selected from the *P*-population.	24 of 34: [3, 33–55]
Ordinal-based	All solution candidates in *P*-population are selected sequentially throughout the search process life cycle.	33 of 34: [3, 34–65]
Greedy	It is also known as the elitist method. Solution candidates are ranked first according to their fitness values. The best solution candidate or the top ranked candidates are selected.	28 of 34: [3, 33–35, 37–44, 46, 47, 50–53, 55, 57–65]
Probabilistic	It is a method that includes both randomness and greed. Roulette wheel and tournament methods are the two best known examples of probabilistic selection. The probability of candidates being elected varies depending on their fitness values. For example, in the binary tournament method, two solution candidates are randomly selected from the population, and whichever of these two has a higher fitness value, it wins the tournament.	3 of 34: [33, 40, 55]

the *i*th solution candidate is represented by $P_i \equiv <x_{1[i]}, x_{2[i]}, x_{3[i]}, \ldots x_{n[i]}>$ and the best solution candidate is represented by $P_{best} \equiv <x_{1[best]}, x_{2[best]}, x_{3[best]}, \ldots x_{n[best]}>$ in *P*. The distance value (D_{P_i}) required to calculate the score of *i*th solution candidate (P_i) can be calculated using the Euclidean metric as given in Eq. (21):

$$_{i=1}^{m} \forall\, P_i, D_{P_i} = \sqrt{\left(x_{1[i]} - x_{1[best]}\right)^2 + \left(x_{2[i]} - x_{2[best]}\right)^2 + \ldots + \left(x_{n[i]} - x_{n[best]}\right)^2} \tag{21}$$

The D_P distance vector for the *P*-population is represented as given in Eq. (22)

$$D_P \equiv \begin{bmatrix} d_1 \\ . \\ . \\ . \\ d_m \end{bmatrix}_{mx1} \tag{22}$$

Before calculating the FDB scores of the solution candidates, the fitness values vector (*F*) given in Eq. (13) and the distance values vector (D_p) given in Eq. (22) are normalized. Using the normalized values ($normF_i$ and $normD_{P_i}$) the FDB score of the *i*th solution candidate (S_{P_i}) is calculated as given in Eq. (23):

$$_{i=1}^{m} \forall\, P_i,\ S_{P_i} = w*normF_i + (1-w)*normD_{P_i} \tag{23}$$

where *w* is the weighting coefficient that determines the effects of the candidate's fitness and distance values ($normF_i$ and $normD_{P_i}$) on the score calculation. In the FDB method, *w* was accepted as fixed at 0.5. The FDB scores of solution candidates in the population are represented by the S_p vector as given in Eq. (24). More detailed information about the FDB method can be obtained from the reference study [1].

$$S_P \equiv \begin{bmatrix} s_1 \\ . \\ . \\ . \\ s_m \end{bmatrix}_{mx1} \tag{24}$$

Summary of the FDB The main motivation in the development of MHS algorithms is the successful balance of exploitation and exploration. As explained in Subsection 3.3. "*Overview of selection methods*", candidates who are selected by the greedy and random methods are frequently used in these two tasks. In order for an MHS algorithm to converge or perform its exploitation task, X_{best} (the solution candidate with the best fitness value in a population) is used. In addition, it is necessary to provide diversity in the population in order not to experience premature convergence problems. In order to ensure diversity, i.e., to perform the exploration task, random individuals are selected from within the population and mutation-like operations are performed. In the FDB method, the aim is to identify the candidate with the feature that offers the highest contribution to X_{best}. If a solution candidate's distance from X_{best} is too close to zero, these two solution candidates represent very similar positions in the search space. It is not expected that two candidates who are in the same position in the search space will contribute to the search process. Therefore, it is necessary to determine whether the candidates for solution are in the same or very similar position with X_{best}. For this purpose, both the fitness values of the solution candidates and the distance values to X_{best} are taken into consideration in the FDB selection method. Although fitness information is an important indicator for the exploitation task, distance information is also an important indicator for the exploration task. For this reason, this selection method is called the "fitness-

distance balance". The *w*-coefficient determines the balance between the two. In the original FDB study [1], the effects of fitness and distance on the score are considered as equal.

The FDB method can be summarized in a simple example as described below. Assume that design parameters x_1 and x_2 that minimize the $G = (x_1 - 10)^2 + (x_2 - 5)^2$ equation will be optimized. To optimize this problem, a population($P \equiv [P_1, P_2, \ldots, P_{10}]$) of 10 candidate solutions has been created. The positions of the solution candidates in the search space can be seen in Fig. 3 (a). The normalized fitness values and normalized distance values of the solution candidates are also shown in Fig. 3 (b and c), respectively. When the fitness values are compared, the best solution candidate in the population is P_5. The FDB scores of the solution candidates were calculated using Eq. (23). As shown in Fig. 3 (d), the candidate with the highest score was P_6.

Motivation for *d*FDB The search process life cycle in MHS algorithms is a long process consisting of hundreds of thousands or even millions of cycles. The needs are changing at each stage of the search process. Although algorithms want to exhibit a robust exploitation ability, they often experience premature convergence in the early stages of the search process. The way to avoid local solution traps is to gradually increase the exploration capability. Therefore, in some periods of the search process, more diversity (a stronger exploration operation) is needed to prevent the problem of premature convergence, while in some periods, exploitation is needed. It is essential to make a sensitive search especially in the last periods of the search process. In order to overcome these difficulties, the *w*-coefficient, which provides the balance between exploitation and exploration, must be converted into a dynamically changing parameter. To discuss the effect of the change in the value of *w* on exploitation and exploration, the FDB scores of the solution candidates given in Fig. 3 (a) were recalculated for $w = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]$. The change of solution candidates selected from the population, depending on the FDB scores calculated for different *w*-coefficient values, is given in Fig. 4.

To understand the effect of the change in the value of *w* on exploitation and exploration, Eq. (23), as shown in Fig. 4 (a), should be examined. Accordingly, the *w*-coefficient is the multiplier of the solution candidate's fitness value ($normF_i$). Therefore, the effect of the fitness parameter on the FDB score

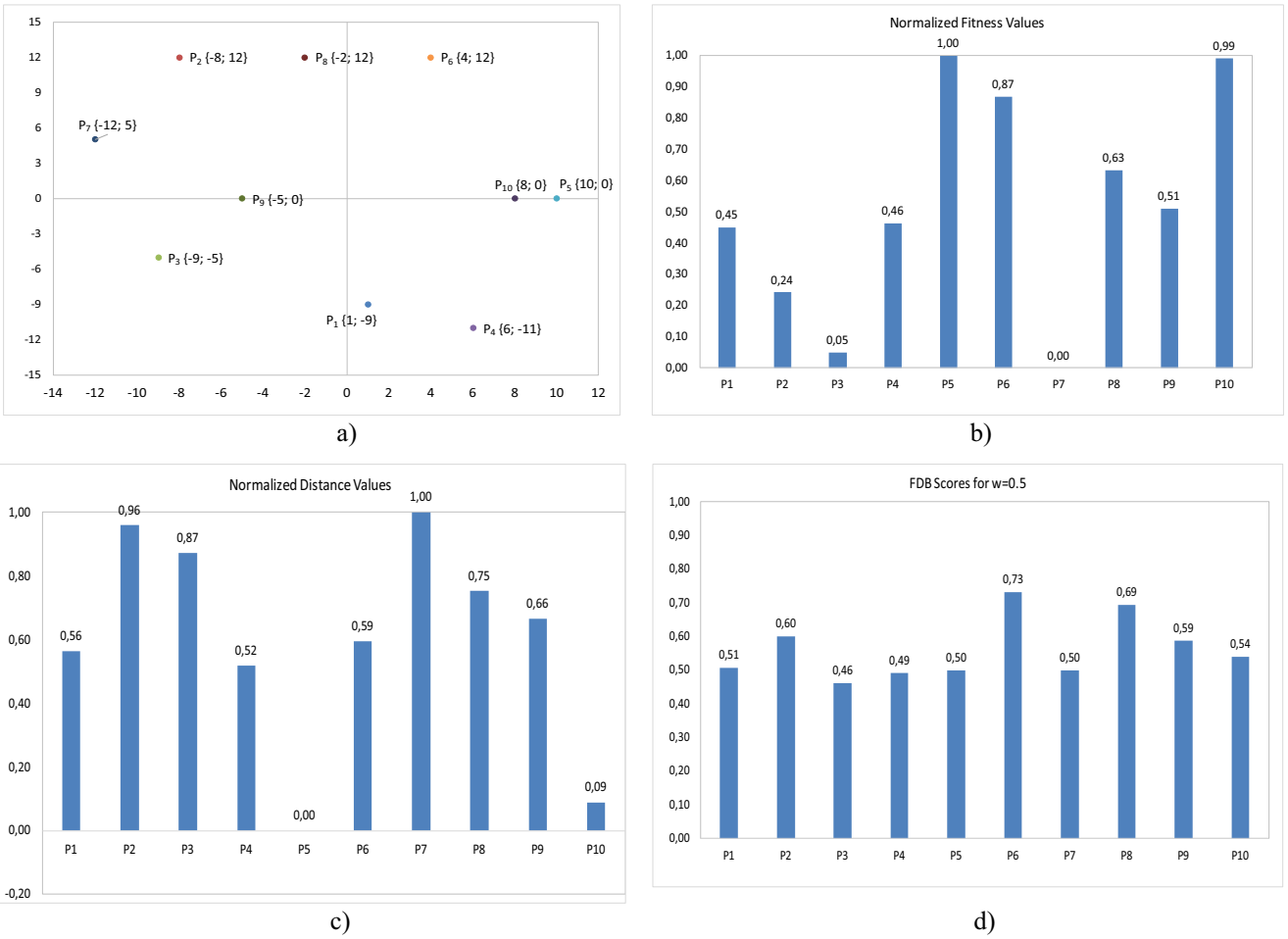


Fig. 3 A simple practical example of FDB explained with graphics

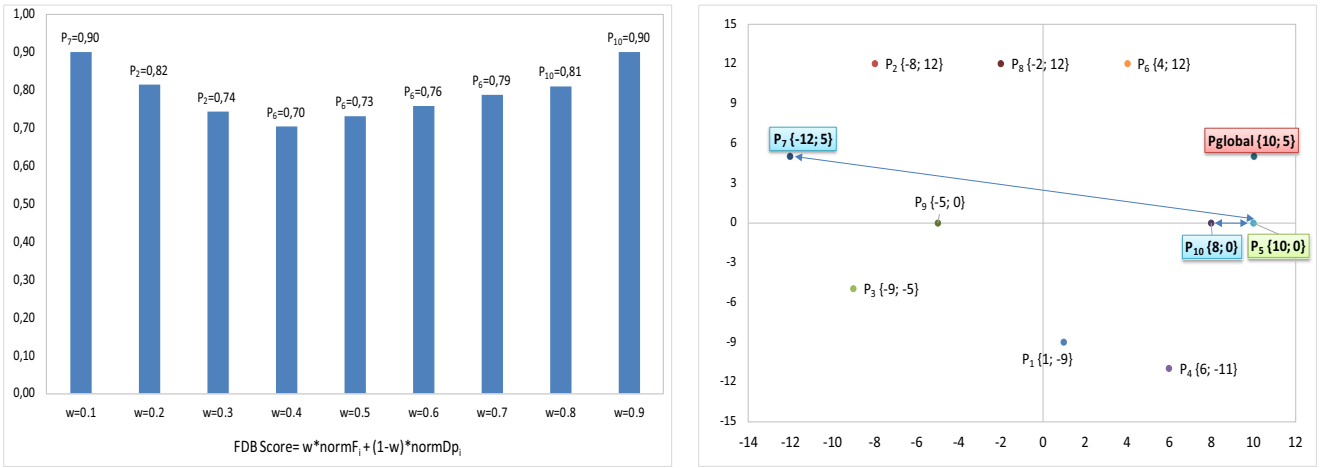


Fig. 4 The effect of the change of *w*-coefficient on exploitation and exploration

is directly proportional to the change in the value of *w*. As the value of *w* increases, the effect of the fitness value on the score increases. This makes it possible to choose a solution candidate that further strengthens and makes the exploitation process more sensitive. The change in the value of *w* and the effect of the distance parameter ($normD_{P_i}$) on the FDB score are inversely proportional. As the value of *w* decreases, the effect of the distance value on the score increases. This makes it possible to choose a candidate for the solution that further strengthens the exploration process. According to the information shown in Fig. 4 (a), the candidates with the highest value score for $w = 0.1$ and 0.9 were P_7 and P_{10}, respectively. Figure 4 (b) was prepared to show more clearly the contribution of the P_7 and P_{10} solution candidates to the exploitation and exploration process. Figure 4 (b) shows information for four different positions. These are P_{global}, representing the global solution of the optimization problem and P_{best} (P_5), the best fitness value in the population; P_7 was selected for $w = 0.1$, and P_{10} s for $w = 0.9$.

The purpose of developing the *d*FDB method was to set the *w*-parameter to make the P_{best} (the best solution candidate in the population in the current iteration) solution candidate as close as possible to the P_{global} position. When the positions of solution candidates in Fig. 4 (b) are examined, it is seen that P_{10} is the candidate closest to P_{best} (P_5). Therefore, the P_{10} solution candidate, which is very similar to P_{best}, was selected for the high values of the *w*-parameter and the exploitation task was performed. When the positions of the candidates in Fig. 4 (b) are examined, it is understood that P_7 is the solution candidate that can make the highest contribution, even though it is far from P_{best}. A design parameter swap (crossover process) to be performed between P_7 and P_{best} can create a very powerful exploration effect.

In the *d*FDB method, the w_{dFDB}-coefficient was developed to become dynamic depending on the steps of the search process life cycle. Each step of the search process life cycle is represented by the current iteration number (*h*). The maximum number of fitness evaluations in a MHS process is indicated by *maxFE*. Accordingly, depending on the value of *h*, the value of w_{dFDB} was calculated as given in Eq. (25):

$$w_{dFDB} = \frac{h}{maxFE} * (1 - lb) + lb \tag{25}$$

The *lb* parameter ($0 < lb < 0.5$, $lb \in R$) shows the lower bound value of $w_{d\mathrm{FDB}}$. It is defined to ensure that the effect of the fitness value is not less than a certain threshold value in the search process. Thus, it is ensured that the search process does not turn into a random search. Accordingly, the pseudo code of the algorithm required to generate a frequency-adjustable $w_{d\mathrm{FDB}}$ signal is given in Algorithm 2.

Algorithm 2. Pseudo code to generate the frequency adjustable $w_{d\mathrm{FDB}}$ signal

```
Begin
Assign the values of maxFE and f
if (maxFE/f >=1)
        mod_ref=round(maxFE / f)
            for ii=1: maxFE
                h=( ii mod mod_ref )
                w_dFDB = (h / maxFE) * (1 - lb) + lb
            end
end
End
```

Two different $w_{d\mathrm{FDB}}$ signals with frequency values 1 and 10 were generated using the pseudo code given in Algorithm 2. The change of the w_{dFDB} produced for both frequency values depending on *h* can be seen in Fig. 5a, b.

Using the *d*FDB method, the score of solution candidates can be calculated as given in Eq. (26):

$$_{i=1}^{m} \forall\ P_i,\ \ S_{P_i} = w_{dFDB} * normF_i + (1 - w_{dFDB}) * normD_{P_i} \tag{26}$$

When Fig. 5 and Eq. (26) are taken into consideration, the value of the $w_{d\mathrm{FDB}}$ coefficient increases linearly starting from 0.2 and becomes 1 when it reaches $h = maxFE$. Thus, the exploration process is dominant in the early stages of the

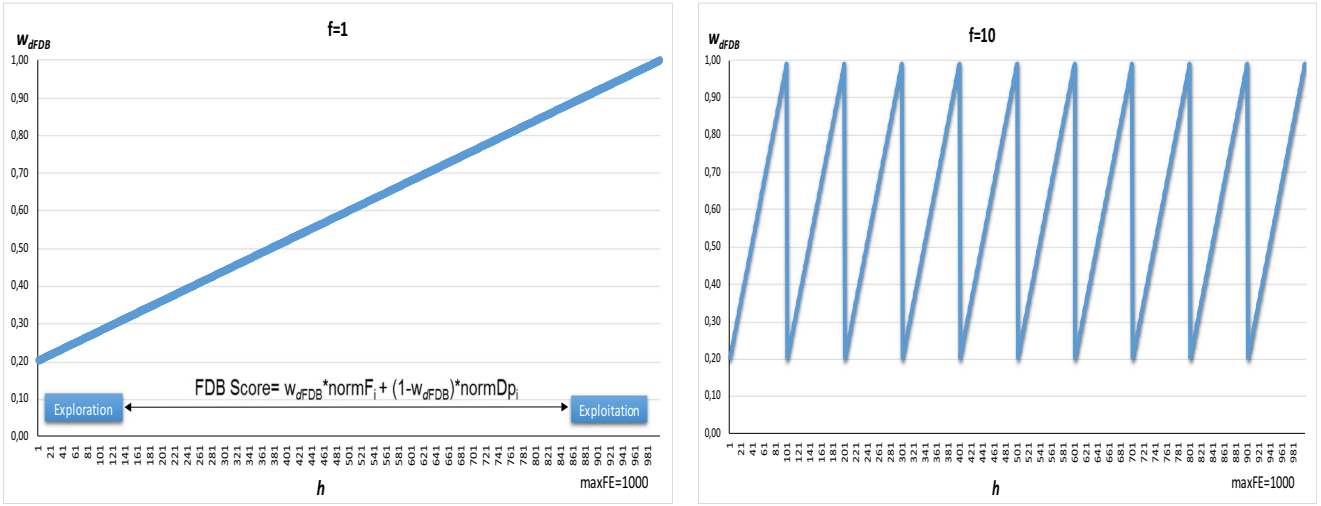


Fig. 5 Generating of w_{dFDB} signal with different frequencies ($f = 1$ (**a**) and $f = 10$ (**b**))

search process and the problem of premature convergence is prevented. In the later stages of the search process, the dominance between exploration and exploitation disappears and a balance is established. In the last stages, the exploitation process becomes dominant, and a sensitive convergence capability is provided.

3.5 Proposed algorithm: development of manta ray foraging optimization versions with dynamic fitness distance balance method

This section presents research conducted to improve the search performance of the MRFO. For this purpose, the *d*FDB method was applied in the search process life cycle of the MRFO and the effects of the method on the performance of the algorithm were examined. When the mathematical model of the MRFO algorithm is examined, there are many cases where the *d*FDB method can be applied. To create these cases, it is necessary to examine Eqs. (14, 16, 19, and 20), where the feeding strategies of the manta rays are modeled. In each equation there are solution candidates that can be selected with the *d*FDB method. Information about the equations used in the MRFO algorithm and the selection of the solution candidates defined in these equations using the *d*FDB method is given in Table 2.

Depending on the information provided in Table 2, dozens of MRFO variations can be created with *d*FDB. However, this section only provides information about the MRFO variations whose performances are analyzed and presented in the experimental study section. The variations of the proposed algorithm with FDB and *d*FDB are introduced in Table 3.

According to the information given in Table 3, the *w*-coefficient is fixed in the FDB method. In the *d*FDB method, four different cases were created using different bounds and different frequencies for *w*. Accordingly, the pseudo code of the proposed MRFO algorithm improved by the *d*FDB method is given in Algorithm 3.

Algorithm 3. The pseudo code of the MRFO with *d*FDB (proposed algorithm)

Begin
  P: initialize a population of n-points (solution candidates/manta rays) as given in Eq. (12).
  F= Use the Eq. (11) and create the fitness vector as given in Eq. (13)
  while (search process lifecycle: up to *maxFE*)
  {
    for i=1:N
      If (rand<0.5) // Cyclone foraging strategy
        If (t/T_{max}<rand)
          $x_{rand}=x_r+\text{rand}\,(x_u-x_l)$
        $$x_i^d(t+1)=\begin{cases} x_{rand}^d + r.\left(x_{rand}^d - x_i^d(t)\right) + \beta.\left(x_{rand}^d - x_i^d(t)\right), i=1 \\ x_{rand}^d + r.\left(x_{i-1}^d(t) - x_i^d(t)\right) + \beta.\left(x_{rand}^d - x_i^d(t)\right), i=2,\ldots,N \end{cases}$$
      Else
        $$x_i^d(t+1)=\begin{cases} x_{best}^d + r.\left(x_{best}^d - x_i^d(t)\right) + \beta.\left(x_{best}^d - x_i^d(t)\right), i=1 \\ x_{best}^d + r.\left(x_{i-1}^d(t) - x_i^d(t)\right) + \beta.\left(x_{best}^d - x_i^d(t)\right), i=2,\ldots,N \end{cases}$$
      End if
    Else // Chain foraging strategy (proposed *d*FDB selection method is used)
      $$x_i^d(t+1)=\begin{cases} x_i^d(t) + r.\left(x_{best}^d - x_i^d(t)\right) + \alpha.\left(x_{dFDB}^d - x_i^d(t)\right), i=1 \\ x_i^d(t) + r.\left(x_{i-1}^d(t) - x_i^d(t)\right) + \alpha.\left(x_{dFDB}^d - x_i^d(t)\right), i=2,\ldots,N \end{cases}$$
    End if
    // Updating Process
    If (fitness of $x_i(t+1) > x_i(t)$ **then** $x_i(t) = x_i(t+1)$ and
    If (fitness of $x_i(t) > x_{best}(t)$ **then** $x_{best} = x_i(t)$
    End if
  End for
  // Somersault foraging strategy
  for i=1:N
    $x_i^d(t+1) = x_i^d(t) + S.(r_2.x_{best} - r_3.x_i(t))$
    // Updating Process (as given in line-13)
  End for
End while

4 Experimental settings

A comprehensive experimental study was conducted to test and verify the search performance of the *d*FDB selection method and MRFO variations with the *d*FDB proposed in the article. The aim was to explicitly reveal the performance of both the developed *d*FDB and the proposed algorithm. The

Table 2 Solution candidate information used in the MRFO algorithm and implementation of *d*FDB strategy

The nutrition strategy	Equation number	Solution candidates defined in equation	Implementing the *d*FDB strategy
chain foraging	14	X_{best}, X_i, X_{i-1}	Solution candidates defined in equations can be selected using the *d*FDB method. $X_{best}=X_{dFDB}$; $Xi=X_{dFDB}$; $X_{rand}=X_{dFDB}$; $X_{i-1}=X_{dFDB}$;
cyclone foraging	16, 19	X_{best}, X_{rand}, X_i, X_{i-1}	
somersault foraging	20	X_{best}, X_i	

activities carried out for this purpose and the experimental study settings are given below:

- Three different benchmark test suites consisting of four different types of problems were used. There were 90 benchmark test functions in total in these sets.
- In the adjustment of the algorithm parameters, the settings given in the base MRFO model [3] were taken as reference for population sizes and all other settings.
- The *maxFE* value was set as the 10.000 * dimension in order to ensure equality of opportunity among the algorithms and to comply with the standards defined in the literature [66, 67].
- All experimental studies were conducted as 51 independent runs for each of the test functions.
- To test the stability and performance of the proposed methods in different problem sizes, the dimensions of the test functions were dynamically selected as 30, 50 and 100.
- Two well-known and strong nonparametric statistical tests including the Wilcoxon and the Friedman tests were selected to further analyze the performance of the proposed methods. The Wilcoxon statistical test was performed at 5% significance level.
- Experimental studies were implemented in MATLAB®R2018a and performed on the Intel (R) Core ™ i7-4770KCPU @ 3.50GHz and 16 GB RAM and ×64-based processor.

4.1 Benchmark test suites used in experimental studies

For a detailed knowledge of the search performance of MHS algorithms, it is necessary to test them in different problem types and in low/middle/high-dimensional search spaces. In the literature, four types of continuous-valued benchmark problems were identified [66, 67]. These are the unimodal, multimodal, hybrid, and composition type problems. Unimodal problems have simple search spaces that do not have local solution traps compared to the other types of problems. Test problems of this type are used to analyze the exploitation performance of MHS algorithms. There are many local solution traps in the search spaces of multimodal-type problems. Problems of this type are used to analyze the exploration performance of algorithms. Hybrid and composition-type problems are more complex than the others. Such problems are used to test the balanced search capabilities of algorithms. Composition-type problems in particular are used to analyze the success of algorithms in achieving an exploitation-exploration balance. A comprehensive benchmark pool was created to effectively measure the performance of the methods proposed in this article. Table 4 provides information about the three benchmark test suites used in the experimental studies.

5 Results and analysis

In this section, the analysis results of the two studies are given. Two subsections were prepared for this. In the first subsection, the performance of the algorithms developed by applying the *d*FDB method is analyzed. For this purpose, performance comparisons between variations of MRFO with FDB and *d*FDB were made and the strongest was proposed. The proposed algorithm was applied to optimize the DOCRs

Table 3 Variations of MRFO with FDB and *d*FDB

Variations	Bounds of *w:* [ub, lb]	*f (frequency)*	Implementation stage	Explanation
Case1: MRFO with FDB	0.5 (fixed)	–	It has been applied to the chain foraging strategy. (See Table 2)	X_{best} candidate given in Eq. (14) is not used in the proposed method. X_{FDB} or X_{dFDB} is used instead of X_{best}. In other words, the solution candidate chosen from the population by FDB and *d*FDB methods is used instead of X_{best}.
Case2: MRFO with *d*FDB	[0; 1]	1		
Case3: MRFO with *d*FDB	[0.4; 1]	1		
Case4: MRFO with *d*FDB	[0; 1]	100		
Case5: MRFO with *d*FDB	[0.4; 1]	100		

Table 4 Benchmark test suites used in experimental study

Benchmark suite	Number of test functions	Function types	Search range	Dimension
Classic Test Functions [1]	30	Unimodal Multimodal	Please see reference study [1]	30, 50 and 100
CEC 2014 [66]	30	Unimodal Multimodal	[−80, 80]	
CEC 2017 [67]	30	Hybrid Composition	[−100, 100]	

coordination problem, which is a constrained engineering design problem, and the analysis results are presented in the second subsection. In addition, powerful up-to-date competitor MHS algorithms were used to optimize the DOCRs coordination problem and comparisons were made.

5.1 Determining the best manta ray foraging optimization method with dynamic fitness distance balance on benchmark test suites

In order to make this subsection easier to understand, the section titled "3.5. Proposed algorithm: development of manta ray foraging optimization versions with dynamic fitness distance balance method" should be read again. Five variations of the MRFO with FDB and *d*FDB were created. In this section, performance comparisons are made between the MRFO, MRFO with FDB, and the variations of the MRFO with *d*FDB. The MRFO with FDB is referred to as Case-1 in this subsection. The variations of the MRFO with *d*FDB are given as Case-2, Case-3, Case-4, and Case-5. For information about these variations, please see Table 3 in Section 3.5. This section consists of three subsections. In the following subsection, the performances of the algorithms are analyzed statistically and the most successful one is determined. In the second subsection, the convergence performances of the algorithms are analyzed. In the third subsection, the computational complexity of the algorithms is tested and compared.

5.1.1 Statistical analysis

In this subsection, the performances of six competing algorithms are examined. In the study which yielded the experimental data, 90 test problems, 3 different problem dimensions, 6 competing algorithms, and 51 independent runs were conducted. Data numbering a total of 82.620 were obtained in the experimental studies for statistical analysis. Based on these data, the rankings of the competing algorithms according to the Friedman test results are presented in Table 5.

According to Table 5, Case-3 demonstrated the most successful performance among the six competitors, while Case-5 was ranked second. Both algorithms are variations of the MRFO with *d*FDB. In both algorithms, the lower bound values of the *w*-coefficient are equal, but the frequency values are different. Although the frequency values are different, Case −1 and Case-3 exhibited similar performances. In addition, despite the difference in the problem dimensions, both algorithms had stable performances. Case-1 ranked third among the six algorithms. This indicates that the MRFO with FDB showed a superior performance compared to the MRFO with *d*FDB defined in Case-4 and Case-2. When Case-2 and Case-4 are examined, the lower bound value of the *w*-coefficient is defined as zero in both algorithms. In fact, the only difference between Case-2 and Case-3 is that the lower bound values of the *w*-coefficient are 0 and 0.4, respectively. Due to this difference, Case-3 ranked first, while Case-2 ranked fifth. The original version of the MRFO algorithm, i.e., the base model, ranked last among the six algorithms. This clearly demonstrates the strength and utility of the FDB and *d*FDB selection methods. Finally, according to Table 5, if the value of the *w*-coefficient is determined dynamically, significant improvements are provided in the performance of the FDB selection method. The fact that Case-3 and Case-5, i.e., variations of the MRFO with *d*FDB, outperformed their competitors in all of the experiments is clear evidence of the success of the *d*FDB method developed in this article.

The Wilcoxon signed-rank test was applied to make pairwise comparisons between the MRFO and its variations using the data obtained in the experimental study. The sum of the numbers in each cell in Table 6 is 30. This shows the number of test problems in the benchmark suite used in the experiment. Three different scores are given in each cell. For example, in the first cell in Table 6, a result of 8/21/1 is given. This experiment was carried out between the MRFO and Case-3 and in the suite of classic benchmark problems for D = 30. According to the scores in the cell, the MRFO algorithm was defeated against Case-3 in 8 of 30 test functions. The two algorithms achieved similar results in 21 problems. In just one of the 30 test functions, the MRFO algorithm performed better than its competitor. According to the results of the analysis, there was no benchmark suite in which the MRFO algorithm excelled over its competitors. The pairwise comparison results clearly revealed the affirmative effect of the *d*FDB method on the performance of the MRFO algorithm.

Table 5 The ranks obtained by the Friedman test

Algorithms	Dimension=30			Dimension=50			Dimension=100			Mean Rank
	Classic	Cec2014	Cec2017	Classic	Cec2014	Cec2017	Classic	Cec2014	Cec2017	
Case-3	3.3232	3.3990	3.2752	3.3895	3.3536	3.2444	3.4121	3.3141	3.2480	3.3288
Case-5	3.4353	3.3794	3.2075	3.3876	3.3676	3.3131	3.4206	3.3647	3.3209	3.3552
Case-1	3.3938	3.4905	3.4578	3.4703	3.3670	3.3444	3.3350	3.3529	3.2585	3.3856
Case-4	3.4340	3.4431	3.5206	3.4271	3.4938	3.4428	3.4614	3.5016	3.4814	3.4673
Case-2	3.5033	3.4719	3.5291	3.5585	3.5520	3.5373	3.5618	3.5954	3.6389	3.5498
MRFO	3.9105	3.8160	4.0098	3.7670	3.8660	4.1180	3.8092	3.8712	4.0523	3.9133

5.1.2 Convergence analysis

In this subsection, the convergence curves of the six competing algorithms are given. To show the algorithm convergence speed on 30-, 50-, and 100-dimensional search spaces, four different types of problems were selected from the CEC2017 [67] benchmark suite to examine the convergence capabilities of the algorithms. These were the F1 (unimodal), F16 (multimodal), F19 (hybrid), and F26 (composition) test functions. The convergence curves of the algorithms are shown in Fig. 6.

Analysis for unimodal-type problems Case-3 for D = 30, 50, and 100 shows a consistently superior convergence performance over its competitors. The convergence curves show that Case-3 effectively accomplished the exploitation function without being affected by the dimension of the search space.

Analysis for multimodal-type problems The F6 test function, which has many local solution traps, seems to be distinctive in terms of algorithm search performance. Despite this, Case-3 consistently outperformed its competitors for all problem dimensions. Convergence curves show that Case-3 had been performing its exploration task steadily. However, in the high-dimensional search space (for D = 100), it is understood that the algorithms had difficulties in providing diversity, overcoming local solution traps, and converging to the global solution.

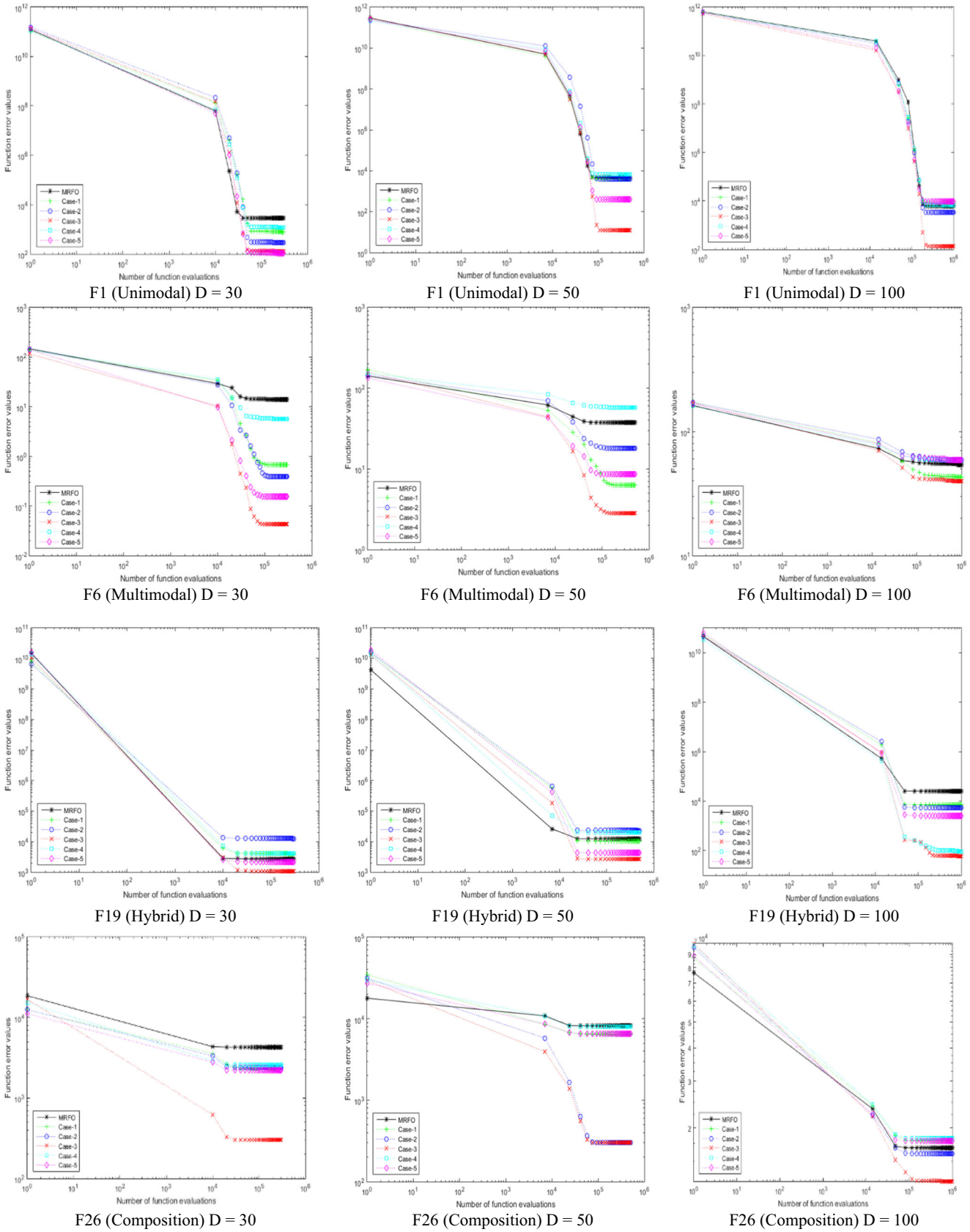
Analysis for hybrid-type problems Examination of the convergence graphs of the algorithms for the F19 test function showed that they exhibited a successful performance in general. In addition, it is understood that Case-3, Case-4, and Case-5 algorithms provided better exploitation-exploration balance. Case-3, in particular, outperformed its competitors with its balanced search capability.

Analysis for composition-type problems Compared to the other problem types, F26 is the one with the highest level of complexity. Case-3 had a clear advantage over its competitors in the high difficulty search space. When other curves were analyzed, the *d*FDB-based methods performed successfully in composition-type search spaces.

When the convergence curves in Fig. 6 are examined, it is seen that the change in error value for $FEs > 10{,}000$ is zero, that is, the improvement in the search process has stopped. At this stage, the algorithms have completed the search process. This indicates that the value assigned for *maxFEs*, which is the search process termination criterion, is sufficient. Therefore, the curves accurately present the necessary information to interpret the convergence performance of the algorithms. Accordingly, when the black curve showing the convergence performance of the base version of MRFO is examined, it is seen that the error value for D = 30, 50 and 100 is large in all four different problem types. This indicates that MRFO has a premature convergence problem and therefore has difficulty

Table 6 Wilcoxon test comparison results

vs. MRFO +/=/−	Dimension=30			Dimension=50			Dimension=100		
	Classic	Cec2014	Cec2017	Classic	Cec2014	Cec2017	Classic	Cec2014	Cec2017
Case-3	8/21/1	9/20/1	16/13/1	7/22/1	8/22/0	15/14/1	7/23/0	4/25/1	12/18/0
Case-5	7/22/1	12/17/1	17/12/1	8/21/1	6/23/1	16/14/0	8/22/0	5/24/1	13/17/0
Case-1	8/21/1	8/19/3	15/14/1	7/22/1	5/23/2	13/15/2	7/22/1	5/24/1	14/14/2
Case-4	7/22/1	10/18/2	16/13/1	8/21/1	7/22/1	12/17/1	8/21/1	6/23/1	12/16/2
Case-2	8/21/1	8/19/3	16/13/1	7/22/1	8/19/3	10/19/1	7/22/1	5/23/2	11/18/1



F1 (Unimodal) D = 30 | F1 (Unimodal) D = 50 | F1 (Unimodal) D = 100

F6 (Multimodal) D = 30 | F6 (Multimodal) D = 50 | F6 (Multimodal) D = 100

F19 (Hybrid) D = 30 | F19 (Hybrid) D = 50 | F19 (Hybrid) D = 100

F26 (Composition) D = 30 | F26 (Composition) D = 50 | F26 (Composition) D = 100

Fig. 6 Convergence curves of algorithms for unimodal / multimodal / hybrid / composition problem types in CEC2017 (D = 30, 50, 100)

converging to the global solution. On the other hand, it is understood that Case-3 is the algorithm that best converges to the global solution in all problem types without exception. When the mathematical definition of Case-3 is examined, it is seen that the chain feeding strategy of MRFO is redesigned using the *d*FDB method. The guide vector (x_{best}) leading the chain foraging strategy was selected using the *d*FDB method. The convergence curves show that the change of leader in chain feeding strategy has improved the MRFO's ability to mimic nature. Guides selected using the *d*FDB method represent promising locations where food is concentrated.

In summary, thanks to the *d*FDB selection method, MRFO was able to imitate nature more effectively. Thus, improvements have been observed in MRFO's exploitation, exploration, and balanced search abilities.

5.1.3 Algorithm complexity

An important criterion in evaluating the MHS search performance is algorithm complexity information. This information is critical for researchers in order to demonstrate the availability and functionality of a proposed MHS algorithm. An algorithm with high computational complexity has little usability, and thus MHS algorithms with this feature are generally not preferred by researchers for application in multi-dimensional and complex search spaces. Therefore, it is necessary to analyze the calculation complexity as well as the convergence performance of an algorithm. The computational complexities of the six competing algorithms are given in Table 7. In addition, the effect of the *d*FDB method on the complexity of the MRFO algorithm is discussed. The standards defined in IEEE CEC 2014 [66] were taken into account in calculating the complexity of the algorithms. T0 and T1 are parameters defined in document CEC14 and used to calculate algorithm complexity. T0 represents the time the algorithm uses to evaluate the F18 problem once. T1, on the other hand, represents the calculation time of the algorithm for a test program defined in CEC14.

According to the data given in Table 7, the *d*FDB method had no significant effect on the computational complexity of the MRFO algorithm. In fact, the computing complexity of Case-3, which gave the most successful performance in the experimental studies, was slightly better than that of its competitors. The reason for this success is that Case-3 was more successful in eliminating local solution traps than its competitors and in this way was able to find the global solution in a shorter time.

As a result of the analysis, it is understood that the MRFO algorithm, modified by the *d*FDB method, had achieved significant improvements in search performance. Tested in different problem types and in low/middle/high dimensional search spaces, the MRFO algorithm with the *d*FDB method demonstrated a stable and successful search performance against all of its competitors in all the experimental studies. This success was achieved because of the *d*FDB method proposed in this study. According to the experimental study results, Case-3 was the most successful among the MRFO with FDB variations. In the following section, Case-3 is used to optimize the DOCRs coordination problem and is compared with several up-to-date and powerful MHS algorithms in the literature.

5.2 Determining the best method on directional overcurrent relays coordination problem

In this paper, the proposed MRFO with *d*FDB, atom search optimization (ASO) [57], artificial electric field algorithm (AEFA) [58], salp swarm algorithm (SSA) [60], and MRFO [3] optimization algorithms were used to investigate solving the problem of coordination of directional overcurrent relays on different test systems (IEEE 3-bus, IEEE 4-bus, 8-bus, 9-bus, and IEEE 30-bus). These test systems are identified as Model I, Model II, Model III, Model IV, and Model V, respectively. As a rule of thumb in this study, the population size was considered to be 50 for all the algorithms. Furthermore, the algorithms stop when the maximum specified number of generations is reached (assumed 10^4 generations). The algorithms were run with the same initial population and the best solution obtained was reported. The configurations of the test systems and variables are given in Tables 8 and 9.

5.2.1 Test models

The performance of the proposed *d*FDB-MRFO and other optimization algorithms for the solution to the DOCRs coordination problem was evaluated on five different test systems including the IEEE 3-bus, the IEEE 4-bus, the 8-bus, the 9-

Table 7 Algorithm complexity

Dimension	T0	T1	MRFO	Case-1	Case-2	Case-3	Case-4	Case-5
D=30	0.02	0.86	84.77	93.25	87.36	83.34	84.20	83.86
D=50		1.53	97.22	99.32	102.39	100.93	97.68	99.81
D=100		5.28	145.84	140.66	136.04	133.05	134.89	138.73

bus, and the IEEE 30-bus test systems. These test systems are described in detail as follow.

Model-I: IEEE 3-bus test system The IEEE 3-bus test system consists of 1 generator, 3 lines, 6 DOCRs, and 8 primary/backup relay pairs as given in Fig. 7. The aim was to find the optimum settings of the six relays that respond to eliminate all near-end and far-end faults. Here, optimal settings for 12 control variables (TDS1–TDS6 and PCS1–PCS6) are to be found. The lower and upper bounds of TDS were 0.05 and 1.1, respectively. The limits of PCS are range from 1.25 to 1.50. The CTI_{min} was set to 0.3 s. The fault current values at the CT primary terminals (I_f), CT_{rating} and fault current values for the primary/backup relays belonging to Model-I are given in Tables 10 and 11 [68].

The optimal values of control variables as well as OF are given in Table 12. This table demonstrates that the optimal value of the control variables obtained by the considered algorithms was within the acceptable range. The minimum total OT of the primary relays was 4.78057 (s) obtained by the conventional MRFO and the *d*FDB-MRFO. In other words, the MRFO and *d*FDB-MRFO results were 6.88029%, 1.01418%, and 0.73258% lower than the simulation results obtained for ASO, AEFA, and SSA, respectively.

Table 13 presents coordination time interval values for R5/R1, R6/R3, R4/R5, and R2/R6 backup/primary relay pairs in case of near-end and far-end faults. CTI refers to the difference between the operating time of the backup relay and the primary relay. For example, the value of CTI1 is the difference between the operating time of R5 and R1, respectively. To coordinate the two overcurrent relays, the operating time difference between the backup relay and primary relay for all relay pairs must be greater than the CTI_{min}. The value of CTI_{min} considered for Model 1 is 0.3 s. It is clear from the numerical results in Table 13 that the coordination margins obtained for all algorithms are within acceptable limits. That is, the CTI1-CTI8 values are greater than or equal to the CTI_{min}. The operating times of the relays should be within certain bounds to ensure system stability and protect equipment damage. In this paper, OT limits of primary relays (T_{ik}^{l}, T_{ik}^{u}) are accepted in the range of 0.1–1 s. The operating times of the primary relays for near-end and far-end faults are tabulated in Table 14. Accordingly, the operating time constraints have been successfully provided for all primary relays.

Table 8 Configuration of test systems

Model	Number of relays	Control variables	Selectivity constraints
Model-I	6	12	8
Model-II	8	16	9
Model-III	14	28	20
Model-IV	24	48	32
Model-V	38	76	62

Table 9 Variable limits for the test systems

Variables	Model-I	Model-II	Model-III	Model-IV	Model-V
[TDS^{l},TDS^{u}]	[0.05, 1.1]	[0.05, 1.1]	[0.1, 1.1]	[0.01, 1]	[0.1, 1.1]
[PCS^{l},PCS^{u}]	[1.25, 1.5]	[1.25, 1.5]	[0.5, 2.5]	[0.5, 2.5]	[1.5, 6]
[T_{ik}^{l},T_{ik}^{u}]	[0.1, 1.1]	[0.1, 1.1]	[0.1, 1.1]	[0.1, 1.1]	[0.1, 1.1]
CTI_{min}	0.3	0.3	0.3	0.2	0.3

Model-II: IEEE 4-bus test system The IEEE 4-bus system contains 2 generators, 4 lines, 8 relays, and 9 relay pairs. In the IEEE 4-bus system shown in Fig. 8, there are 16 control variables (TDS1–TDS8 and PCS1–PCS8) that need to be optimized. The lower and upper bounds of TDS were 0.05 and 1.1, the range of PCS was from 1.25 to 1.50, and the CTI_{min} was set as 0.3 s. Tables 15 and 16 show the fault current values at the CT primary terminals (I_f),CT_{rating} and fault current values for the primary/backup relay pairs for Model-II [68].

The optimal setting values of the TDS and PCS control variables computed by the *d*FDB-MRFO and the results of the techniques noted in this study are provided in Table 17. This table shows that the OF values of the base MRFO, *d*FDB-MRFO, ASO, AEFA, and SSA algorithms were 3.66934 s, 3.66934 s, 3.92797 s, 3.71191 s, and 3.72290 s, respectively. When the results of the algorithms for Model-II are compared, it is understood that the MRFO and *d*FDB-MRFO algorithms offer the lowest OF values. The simulation results of the MRFO and *d*FDB-MRFO algorithms reduced the total primary relay operation time by 0.25863 s, 0.04257 s, and 0.05356 s according to the results of the other methods.

The values of the coordination time interval and relay operating time for the corresponding algorithms are given in Tables 18 and 19, respectively. Upon examination of Table 18, it is understood that the directional overcurrent relays operate without the miscoordination pairs. This

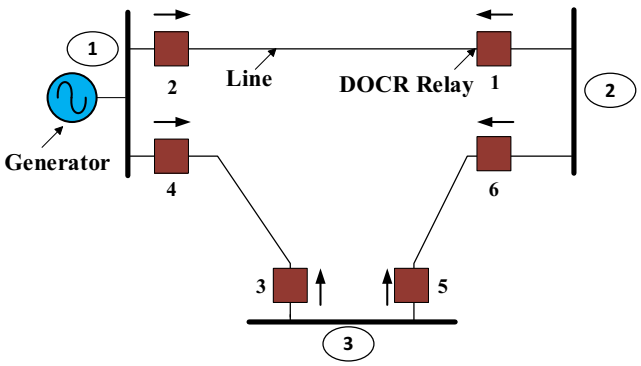


Fig. 7 IEEE 3-bus test system (Model-I)

Table 10 Values of I_f and CT_{rating} for Model-I

Relay	Fault current (p.u.)		CT_{rating}	Line	Relay	Fault current (p.u.)		CT_{rating}	Line
	Near-end	Far-end				Near-end	Far-end		
R1	9.46	14.08	2.06	1–2	R4	37.68	136.23	2.23	1–3
R2	26.91	100.63	2.06	1–2	R5	17.93	25.90	0.80	2–3
R3	8.81	12.07	2.23	1–3	R6	14.35	19.20	0.80	2–3

Table 11 Fault current values for primary/backup relays in Model-I

Backup relay	Fault current (p.u.)		Primary relay	Fault current(p.u.)	
	Near-end	Far-end		Near-end	Far-end
R5	14.08	9.46	R1	14.08	9.46
R6	12.07	8.81	R3	12.07	8.81
R4	25.90	17.93	R5	25.90	17.93
R2	14.35	19.20	R6	14.35	19.20

conclusion is based on the fact that CTI1-CTI9 values calculated for the backup/primary relay pairs meet the coordination margin constraint. The numeric results in Table 19 clearly show that the primary relay operation times were within the acceptable range.

Model-III: 8-bus test system The 8-bus test system (Fig. 9) has 2 generators, 2 transformers, 7 lines, 14 relays, and 20 primary/backup relay pairs, with each relay having two decision variables (TDS1-TDS14 and PCS1-PCS14). The lower and upper TDS bounds were 0.1 and 1.1, and the PCS values ranged from 0.5 to 2.5. The CTI_{min} was set as 0.3 s and the CT

Table 12 Results of control variables for Model-I

Relay	ASO		AEFA		SSA		MRFO		*d*FDB-MRFO	
	TDS	PCS	TDS	PCS	TDS	PCS	TDS	PCS	TDS	PCS
R1	0.0500	1.2990	0.0500	1.2500	0.0500	1.2520	0.0500	1.2500	0.0500	1.2500
R2	0.2127	1.3840	0.2090	1.3640	0.2105	1.3390	0.1976	1.5000	0.1976	1.5000
R3	0.0500	1.3530	0.0500	1.2500	0.0500	1.2570	0.0500	1.2500	0.0500	1.2500
R4	0.2356	1.3970	0.2284	1.2510	0.2147	1.4280	0.2090	1.5000	0.2090	1.5000
R5	0.2147	1.3860	0.1845	1.4360	0.1853	1.4220	0.1812	1.5000	0.1812	1.5000
R6	0.1980	1.3210	0.1888	1.3550	0.1822	1.4780	0.1807	1.5000	0.1807	1.5000
OF(s)		5.13379		4.82955		4.81585		**4.78037**		**4.78037**
MOF		53.25686		46.88991		46.64826		46.00785		46.00785

Table 13 Comparison of the CTI values for Model-I (T_{backup}-$T_{primary}$)

Fault	Relay		Coordination Time	ASO	AEFA	SSA	MRFO	*d*FDB-MRFO
	Backup	Primary						
Near-end	R5	R1	CTI1	0.369	0.300	0.300	0.300	0.300
	R6	R3	CTI2	0.306	0.300	0.300	0.300	0.300
	R4	R5	CTI3	0.300	0.300	0.300	0.300	0.300
	R2	R6	CTI4	0.389	0.384	0.381	0.390	0.390
Far-end	R5	R1	CTI5	0.412	0.334	0.334	0.336	0.336
	R6	R3	CTI6	0.317	0.317	0.320	0.321	0.321
	R4	R5	CTI7	0.401	0.386	0.397	0.400	0.400
	R2	R6	CTI8	0.302	0.300	0.300	0.300	0.300

Table 14 Operation time (OT) values of all optimization algorithms for Model-I

Relay	Operation time (s)	ASO		AEFA		SSA		MRFO		*d*FDB-MRFO	
		OT_{near}	OT_{far}	OT_{near}	OT_{far}	OT_{near}	OT_{far}	OT_{near}	OT_{far}	OT_{near}	OT_{far}
R1	OT1	0.274	0.207	0.265	0.203	0.266	0.203	0.265	0.203	0.265	0.203
R2	OT2	0.648	0.403	0.633	0.394	0.632	0.395	0.625	0.383	0.625	0.384
R3	OT3	0.323	0.249	0.301	0.235	0.302	0.236	0.301	0.235	0.301	0.235
R4	OT4	0.645	0.420	0.598	0.395	0.593	0.385	0.590	0.380	0.590	0.380
R5	OT5	0.525	0.462	0.457	0.402	0.458	0.402	0.456	0.400	0.456	0.400
R6	OT6	0.518	0.464	0.499	0.447	0.498	0.445	0.497	0.444	0.497	0.444

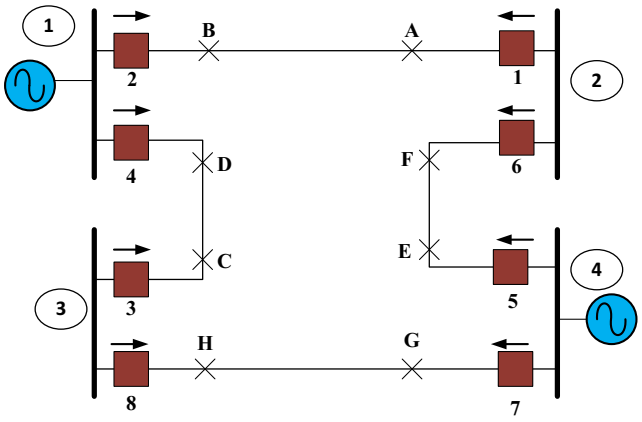


Fig. 8 IEEE 4-bus test system (Model-II)

ratio was 800/5 for R3, R7, R9, and R14, and 1200/5 for the other relays. Table 20 presents the primary/backup relay pairs and fault current values in the 8-bus test system [13]. ASO, AEFA, SSA, MRFO, and the proposed *d*FDB-MRFO algorithm were applied to the optimization of the DOCRs coordination problem in 8-bus test system. The optimal values of the TDS and PCS control variables and the comparative analysis of the objective function of the proposed algorithm with other algorithms are presented in Table 21.

It can be noted from Table 21 that the *d*FDB-MRFO algorithm reached an OF value of 8.54785 s, while a value of 10.75494 s was attained using the original MRFO algorithm. The simulation result of the proposed *d*FDB-MRFO algorithm represents a 20.5216% reduction in terms of the OF value, compared to the simulation result of the MRFO algorithm. The MRFO with the *d*FDB algorithm was lower by 19.9051%, 29.5346%, and 24.9218% compared to the best reported results of 10.67216 s, 12.13057 s, and 11.26083 s of the other algorithms, respectively. Accordingly, the proposed *d*FDB-MRFO algorithm is eliminated local solution traps, explore promising regions of the search space, and converge towards to global solution effectively compared to other algorithms. The results show that the total OT of the primary relays had been significantly reduced using the proposed method. The CTI and OT values for Model-III are shown in Tables 22 and 23, respectively. It is understood from Table 22 that the selectivity constraints (CT1-CT20) for twenty primary/backup relay pairs are within acceptable limits. This shows that DOCRs are operating without the miscoordination pair.

Model-IV: 9-bus test system Figure 10 presents a single-line diagram of the 9-bus test system. This system comprises 12 lines, 24 DOCRs, and 32 primary/backup relay pairs. The aim was to arrange the settings of 24 relays and the primary target was to minimize the total OT of all the primary relays. The CT ratio for each relay was 500:1 and the CTI_{min} was set as 0.2 s. In addition, details of this system such as the P/B relationship between relay pairs, and the fault currents are shown in Table 24 [69].

Table 15 Values of I_f and CT_{rating} for Model-II

Relay	Fault current (p.u.)		CT_{rating}	Line	Relay	Fault current (p.u.)		CT_{rating}	Line
	Near-end	Far-end				Near-end	Far-end		
R1	20.32	12.48	0.4800	1–2	R5	116.70	31.92	1.5259	2–4
R2	88.85	23.75	0.4800	1–2	R6	16.67	12.07	1.5259	2–4
R3	13.61	10.38	1.1789	1–3	R7	71.70	18.91	1.2018	3–4
R4	116.81	31.92	1.1789	1–3	R8	19.27	11.00	1.2018	3–4

Table 16 Fault current values for primary/backup relays in Model-II

Backup relay	Fault current (p.u.)		Primary relay	Fault current (p.u.)	
	Near-end	Far-end		Near-end	Far-end
R5	20.32	12.48	R1	20.32	12.48
R7	13.61	10.38	R3	13.61	10.38
R1	1.16	–	R4	116.81	–
R2	12.07	16.67	R6	12.07	16.67
R4	11.00	19.27	R8	11.00	19.27

A comparison of the optimization results of Model-IV including TDS and PCS is reported in Table 25, as well as the objective function value. Considering the OF values, it is seen that *d*FDB-MRFO is the only pioneer establishing superiority over the others in finding a more optimal objective function value, which is followed by MRFO, ASO, SSA, and AEFA. Accordingly, the *d*FDB-MRFO algorithm had the minimum OF value of 9.92824 s, which was lower by 0.44646 s, 1.36726 s, 1.51164 s, and 2.49322 s than the MRFO, ASO, SSA, and AEFA algorithm simulation results, respectively. The *d*FDB-MRFO objective function result was 12.1044%, 20.0718%, 13.2137%, and 4.3033% lower than the simulation results obtained for ASO, AEFA, SSA, and MRFO, respectively. Simulation results showed that the *d*FDB-MRFO, as in other test cases, was able to converge to a much better global solution for the DOCRs coordination problem in Model-IV compared to other algorithms.

The comparison of CTI values for the corresponding algorithms is reported in Table 26. For effective coordination of DOCRs in Model-IV, thirty-two selectivity criteria (CT1-CT32) for primary/backup relay pairs should be set within limits. Accordingly, when the numerical values in Table 26 are analyzed in-depth, it is understood that the CTI values of the algorithms except SSA are within acceptable limits. SSA failed to provide coordination (CT11) for the R10/R8 relay pair. The operating time (OT) of the primary relays obtained by all optimization algorithms for Model-IV is presented in Table 27. From the results in Table 27, it is understood that the

Table 17 Results of control variables for Model-II

	ASO		AEFA		SSA		MRFO		*d*FDB-MRFO	
Relay	TDS	PCS	TDS	PCS	TDS	PCS	TDS	PCS	TDS	PCS
R1	0.0500	1.284	0.0500	1.354	0.0500	1.341	0.0500	1.273	0.0500	1.273
R2	0.2671	1.313	0.2185	1.371	0.2187	1.380	0.2122	1.500	0.2122	1.500
R3	0.0500	1.301	0.0500	1.255	0.0500	1.250	0.0500	1.250	0.0500	1.250
R4	0.1588	1.368	0.1634	1.250	0.1572	1.403	0.1516	1.500	0.1516	1.500
R5	0.1318	1.456	0.1352	1.307	0.1374	1.258	0.1264	1.500	0.1264	1.500
R6	0.0540	1.300	0.0500	1.252	0.0500	1.274	0.0500	1.250	0.0500	1.250
R7	0.1441	1.375	0.1415	1.340	0.1406	1.355	0.1338	1.500	0.1338	1.500
R8	0.0500	1.320	0.0500	1.252	0.0500	1.333	0.0500	1.250	0.0500	1.250
OF(s)	3.92797		3.71191		3.72290		**3.66934**		**3.66934**	
MOF	214,952.84		214,949.22		214,949.38		214,948.61		214,948.61	

Table 18 Comparison of the CTI values for Model-II (T_{backup}-$T_{primary}$)

Fault	Relay		Coordination Time	ASO	AEFA	SSA	MRFO	*d*FDB-MRFO
	Backup	Primary						
Near-end	R5	R1	CTI1	0.311	0.300	0.300	0.300	0.300
	R7	R3	CTI2	0.312	0.300	0.300	0.300	0.300
	R2	R6	CTI3	0.409	0.324	0.324	0.326	0.326
	R4	R8	CTI4	0.391	0.385	0.392	0.398	0.398
	R1	R4	CTI5	0.301	0.350	0.343	0.300	0.300
Far-end	R5	R1	CTI6	0.412	0.392	0.389	0.400	0.400
	R7	R3	CTI7	0.359	0.346	0.346	0.350	0.350
	R2	R6	CTI8	0.378	0.300	0.300	0.300	0.300
	R4	R8	CTI9	0.300	0.300	0.300	0.300	0.300

Table 19 Operation time (OT) values of all optimization algorithms for Model-II

Relay	Operation time (s)	ASO		AEFA		SSA		MRFO		*d*FDB-MRFO	
		OT_{near}	OT_{far}	OT_{near}	OT_{far}	OT_{near}	OT_{far}	OT_{near}	OT_{far}	OT_{near}	OT_{far}
R1	OT1	0.097	0.113	0.098	0.115	0.098	0.115	0.096	0.113	0.096	0.113
R2	OT2	0.359	0.497	0.297	0.411	0.297	0.412	0.294	0.410	0.294	0.410
R3	OT3	0.157	0.180	0.154	0.176	0.154	0.176	0.154	0.176	0.154	0.176
R4	OT4	0.249	0.361	0.250	0.361	0.248	0.361	0.243	0.356	0.243	0.356
R5	OT5	0.224	0.337	0.223	0.332	0.225	0.332	0.216	0.327	0.216	0.327
R6	OT6	0.174	0.206	0.158	0.186	0.159	0.188	0.158	0.186	0.158	0.186
R7	OT7	0.258	0.404	0.251	0.392	0.250	0.392	0.245	0.389	0.245	0.389
R8	OT8	0.137	0.177	0.134	0.172	0.137	0.178	0.134	0.172	0.134	0.172

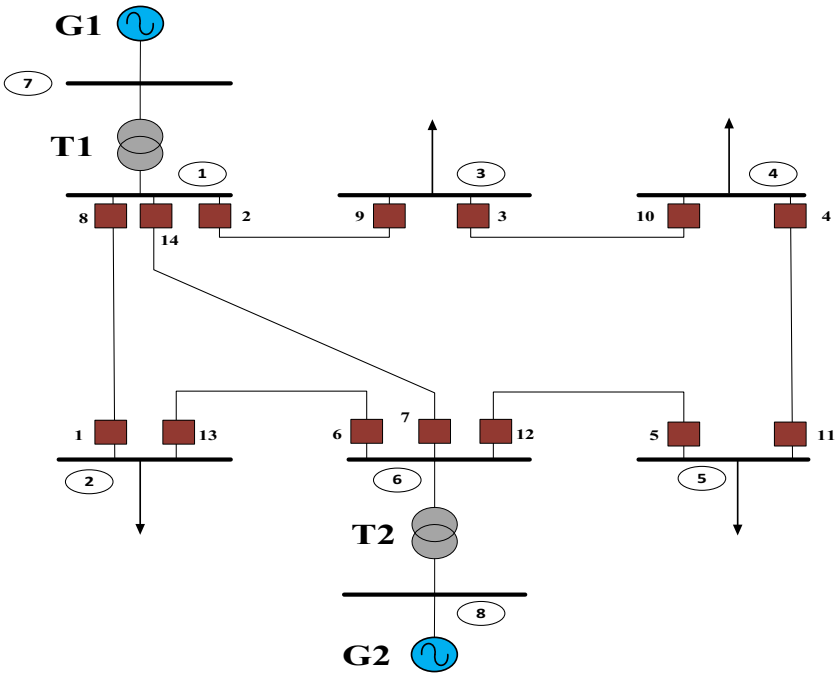


Fig. 9 8-bus test system (Model III)

Table 20 Primary/backup relays and fault currents for the 8-bus test system (Model-III)

Primary relay	Fault current (I_f) (A)	Backup relay	Fault current (I_f) (A)	Primary relay	Fault current (I_f) (A)	Backup relay	Fault current (I_f) (A)
R1	3232	R6	3232	R8	6093	R7	1890
R2	5924	R1	996	R8	6093	R9	1165
R2	5924	R7	1890	R9	2484	R10	2484
R3	3556	R2	3556	R10	3883	R11	2344
R4	3783	R3	2244	R11	3707	R12	3707
R5	2401	R4	2401	R12	5899	R13	987
R6	6109	R5	1197	R12	5899	R14	1874
R6	6109	R14	1874	R13	2991	R8	2991
R7	5223	R5	1197	R14	5199	R1	996
R7	5223	R13	987	R14	5199	R9	1165

Table 21 Simulation results of control variables for Model-III

Relay	ASO		AEFA		SSA		MRFO		*d*FDB-MRFO	
	TDS	PCS	TDS	PCS	TDS	PCS	TDS	PCS	TDS	PCS
R1	0.2238	1.1510	0.2929	0.8450	0.1645	1.7820	0.1000	2.4910	0.1000	2.390
R2	0.3787	1.4350	0.3642	1.8950	0.4235	1.5630	0.4389	1.4920	0.2594	2.500
R3	0.2762	2.1540	0.3294	1.8990	0.3956	1.4560	0.4587	1.0130	0.2241	2.499
R4	0.2614	1.2900	0.2700	1.7010	0.4468	0.5380	0.3270	1.2850	0.1592	2.499
R5	0.1218	2.3220	0.2365	1.1470	0.1977	1.5920	0.2825	0.8760	0.1000	2.463
R6	0.3493	1.0140	0.3152	1.6210	0.4314	0.9780	0.2702	0.9970	0.3188	0.607
R7	0.3739	1.4110	0.3422	1.8490	0.4145	1.4290	0.3085	2.4880	0.2878	1.876
R8	0.2696	1.7140	0.3242	1.7300	0.2873	1.4950	0.1702	2.4440	0.1628	2.500
R9	0.2501	1.5400	0.4594	0.5030	0.1765	2.4820	0.1907	2.3910	0.1499	2.499
R10	0.2413	2.0940	0.2914	1.9600	0.2291	2.0060	0.4047	0.6620	0.1800	2.458
R11	0.2773	1.7480	0.2903	1.9630	0.3221	1.1990	0.2349	2.4090	0.2163	2.055
R12	0.3747	1.7290	0.3711	2.0640	0.3742	1.6960	0.4295	1.2570	0.2746	2.496
R13	0.2076	1.3170	0.3668	0.6120	0.1296	2.0050	0.1000	2.3960	0.1000	2.211
R14	0.3450	1.7920	0.3915	1.5380	0.3847	1.4290	0.4421	1.1090	0.2535	2.466
OF(s)	10.67216		12.13057		11.26083		10.75494		**8.54785**	
MOF	230.1577		296.4023		257.9658		242.9288		**151.1337**	

AEFA and SSA algorithms violate the primary relay operating time constraint. Accordingly, the primary relay operating times obtained by the AEFA (OT22) and SSA (OT20) algorithms are not within the acceptable range of 0.1–1.1 s. The results showed that AEFA and SSA algorithms were insufficient to provide coordination of DOCRs for Model-IV.

Table 22 Comparison of the CTI values for Model-III (T_{backup}-$T_{primary}$)

Relay		Coordination time	ASO	AEFA	SSA	MRFO	*d*FDB-MRFO
Backup	Primary						
R6	R1	CTI1	0.3000	0.3000	0.5630	0.3000	0.3000
R1	R2	CTI2	0.3000	0.3000	0.3050	0.3000	0.4860
R7	R2	CTI3	0.3000	0.3000	0.3000	0.3000	0.3000
R2	R3	CTI4	0.3000	0.3000	0.3000	0.3000	0.3000
R3	R4	CTI5	0.3000	0.3000	0.3000	0.3000	0.3000
R4	R5	CTI6	0.3000	0.3000	0.3000	0.3000	0.3000
R5	R6	CTI7	0.3720	0.3300	0.3010	0.5510	0.4090
R1	R6	CTI8	0.5270	0.5430	0.3560	0.7170	0.5460
R5	R7	CTI9	0.3000	0.3000	0.3000	0.3000	0.3000
R13	R7	CTI10	0.4550	0.5120	0.3550	0.4720	0.4360
R7	R8	CTI11	0.5240	0.4450	0.6540	0.8670	0.5940
R9	R8	CTI12	0.4280	0.3480	0.4450	0.6880	0.4900
R10	R9	CTI13	0.3000	0.3000	0.3000	0.3020	0.3000
R11	R10	CTI14	0.3000	0.3000	0.3000	0.3000	0.3000
R12	R11	CTI15	0.3000	0.3000	0.3000	0.3000	0.3000
R13	R12	CTI16	0.3000	0.3000	0.3000	0.3070	0.3000
R14	R12	CTI17	0.3000	0.3000	0.3000	0.3000	0.3000
R8	R13	CTI18	0.3000	0.3000	0.4400	0.3020	0.3000
R1	R14	CTI19	0.3960	0.3960	0.5150	0.4790	0.5900
R9	R14	CTI20	0.3000	0.3000	0.3000	0.3000	0.3000

Table 23 Operation time (OT) values of all optimization algorithms for Model-III

Relay	Operation time (s)	ASO	AEFA	SSA	MRFO	*d*FDB-MRFO	Relay	Operation time (s)	ASO	AEFA	SSA	MRFO	*d*FDB-MRFO
R1	OT1	0.621	0.720	0.558	0.408	0.398	R8	OT8	0.681	0.822	0.690	0.497	0.480
R2	OT2	0.906	0.968	1.045	1.065	0.775	R9	OT9	0.740	0.906	0.662	0.701	0.564
R3	OT3	0.809	0.914	0.989	1.008	0.702	R10	OT10	0.809	0.946	0.752	0.858	0.656
R4	OT4	0.713	0.830	0.895	0.890	0.594	R11	OT11	0.872	0.965	0.860	0.869	0.736
R5	OT5	0.575	0.748	0.739	0.792	0.492	R12	OT12	0.962	1.023	0.954	0.982	0.821
R6	OT6	0.735	0.779	0.897	0.565	0.575	R13	OT13	0.632	0.827	0.487	0.418	0.398
R7	OT7	0.807	0.810	0.898	0.817	0.685	R14	OT14	0.809	0.871	0.835	0.886	0.671

Model-V: IEEE 30-bus test system The IEEE 30-bus test system [70] which has a more complex structure than the other test systems was used to demonstrate the effectiveness of the proposed algorithm in large-scale power systems. This system consists of 38 relays and 62 primary/backup relay pairs. The test model has 76 control variables (TDS1–TDS38 and PCS1–PCS38). The TDS range was 0.1 to 1.1 and for PCS, 1.5 to 6. The CT ratio for each relay was assumed equal to 1000/5 (Fig. 11). The CTI_{min} was considered to be 0.3 s. The fault current values of the test system are presented in Table 28 [70].

Table 29 presents the optimal values of the control variables obtained using the optimization algorithms for Model-V. The ASO, AEFA, SSA, MRFO, and *d*FDB based MRFO algorithms were used to solve the DOCRs coordination problem on the IEEE 30-bus test system. The OF results obtained from the *d*FDB-MRFO, ASO, AEFA, SSA, and MRFO optimization algorithms were **25.94583 s**, 86.88706 s, 48.72887 s, 32.94089 s, and 26.45235 s, respectively. Accordingly, the *d*FDB-MRFO objective function result was **70.1384%**, 46.7547%, 21.2351%, and 1.9148% lower than the simulation results obtained for ASO, AEFA, SSA, and MRFO, respectively. The simulation results clearly showed that the proposed algorithm is an effective method to solve the DOCRs coordination problem in large-scale power systems.

The comparison of CTI parameters set for backup/primary relay pairs is represented in Table 30. Upon examination, it is understood that the algorithms, except AEFA, successfully meet the selectivity constraints. From the results in Table 31, it is understood that the OT values obtained by the ASO, AEFA, and SSA algorithms are not within acceptable limits. Accordingly, ASO, AEFA, and SSA algorithms are insufficient in solving the DOCRs coordination problem for Model-V.

In summary, depending on the results obtained from experimental studies for the IEEE 3-bus, IEEE 4-bus, 8-bus, 9-bus, and IEEE 30-bus test systems, it can be said that the *d*FDB-MRFO is superior compared to its competitors for the solution

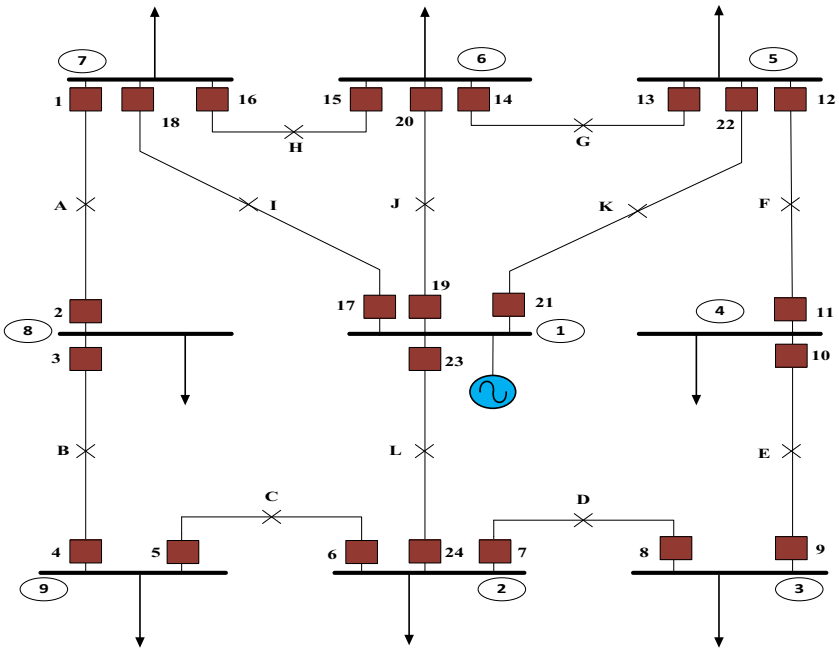


Fig. 10 9-bus test system (Model-IV)

Table 24 Fault current values for 9-bus test system (Model-IV)

Location	Primary relay	Backup relay	$I_{f,primary}$ (A)	$I_{f,backup}$ (A)	Location	Primary relay	Backup relay	$I_{f,primary}$ (A)	$I_{f,backup}$ (A)
A	R1	R15	24,779	9150	G	R13	R11	16,087	3088
	R1	R17	24,779	15,632		R13	R21	16,087	13,000
	R2	R4	8327	8327		R14	R16	18,213	6285
B	R3	R1	16,390	16,390		R14	R19	18,213	11,934
	R4	R6	14,671	14,671	H	R15	R13	18,213	6285
C	R5	R3	9454	9454		R15	R19	18,213	11,935
	R6	R8	23,280	4777		R16	R2	16,087	3088
	R6	R23	23,280	18,507		R16	R17	16,087	13,000
D	R7	R5	23,280	4777	I	R18	R2	8161	2426
	R7	R23	23,280	18,507		R18	R15	8161	5736
	R8	R10	9454	9454	J	R20	R13	9286	4644
E	R9	R7	15,304	15,304		R20	R16	9286	4644
	R10	R12	16,490	16,490	K	R22	R11	8161	2426
F	R11	R9	8326	8327		R22	R14	8161	5736
	R12	R14	24,779	9150	L	R24	R5	6149	3075
	R12	R21	24,779	15,631		R24	R8	6149	3075

Table 25 Simulation results of control variables for Model-IV

Relay	ASO		AEFA		SSA		MRFO		*d*FDB-MRFO	
	TDS	PCS	TDS	PCS	TDS	PCS	TDS	PCS	TDS	PCS
R1	0.5276	0.565	0.3825	2.326	0.3639	1.760	0.3303	2.158	0.3294	1.940
R2	0.1370	2.092	0.2295	1.204	0.1285	2.177	0.2136	1.004	0.1147	2.236
R3	0.3151	1.325	0.3515	1.499	0.2450	2.353	0.2337	2.474	0.2278	2.286
R4	0.2325	1.462	0.2448	2.023	0.2716	0.901	0.2467	1.588	0.2062	1.541
R5	0.2014	1.959	0.3178	1.045	0.2132	1.708	0.2425	1.143	0.1698	2.183
R6	0.2945	1.857	0.3396	1.774	0.3577	1.050	0.2892	2.300	0.3046	1.375
R7	0.3715	1.144	0.3656	1.963	0.3202	1.675	0.2748	1.905	0.3089	1.383
R8	0.2674	1.024	0.2797	1.146	0.2434	1.295	0.1758	2.305	0.1732	2.085
R9	0.3356	0.565	0.3179	1.443	0.2347	1.655	0.1924	1.927	0.2412	0.998
R10	0.4211	0.595	0.3216	1.556	0.2785	1.846	0.2447	2.160	0.2362	2.111
R11	0.1444	2.037	0.3565	0.501	0.1506	1.842	0.1249	2.074	0.1049	2.470
R12	0.4441	1.191	0.3649	2.180	0.4777	0.775	0.3164	2.311	0.3502	1.596
R13	0.3626	0.977	0.4224	0.792	0.3413	0.981	0.2309	2.258	0.2624	1.602
R14	0.5193	0.553	0.4783	0.699	0.3760	1.307	0.3071	1.779	0.2848	2.042
R15	0.3592	1.533	0.4431	1.034	0.3081	1.933	0.3655	1.205	0.2868	2.010
R16	0.2758	2.005	0.4536	0.603	0.2861	1.645	0.2709	1.562	0.2470	1.800
R17	0.4222	1.716	0.4525	1.665	0.7979	1.140	0.3627	2.100	0.4278	1.204
R18	0.0868	1.094	0.0795	0.874	0.2057	2.205	0.0562	0.933	0.0661	2.201
R19	0.4623	1.142	0.3686	2.357	0.5013	1.812	0.3731	1.538	0.3275	1.936
R20	0.0991	0.822	0.0426	1.947	0.0124	2.246	0.0773	1.312	0.0638	1.608
R21	0.5022	1.057	0.4089	1.893	0.4434	2.285	0.5166	0.658	0.4103	1.368
R22	0.0868	0.990	0.0307	1.684	0.1562	1.864	0.0756	0.918	0.0659	2.318
R23	0.3648	2.176	0.4051	2.307	0.4995	2.008	0.3820	1.773	0.3686	1.602
R24	0.0489	1.631	0.0548	1.359	0.0438	1.040	0.0563	2.281	0.0987	0.814
OF(s)	11.29550		12.42146		11.43988		10.37470		**9.92824**	
MOF	324.9641		390.2518		339.0981		274.1420		**251.0414**	

Table 26 Comparison of the CTI values for Model-IV (T_{backup}-$T_{primary}$)

Relay		Coordination Time	ASO	AEFA	SSA	MRFO	*d*FDB-MRFO	Relay		Coordination Time	ASO	AEFA	SSA	MRFO	*d*FDB-MRFO
Backup	Primary							Backup	Primary						
R15	R1	CTI1	0.200	0.200	0.200	0.200	0.200	R11	R13	CTI17	0.200	0.200	0.200	0.200	0.200
R7	R1	CTI2	0.200	0.200	0.894	0.200	0.201	R21	R13	CTI18	0.361	0.295	0.585	0.355	0.353
R4	R2	CTI3	0.200	0.200	0.200	0.200	0.200	R16	R14	CTI19	0.200	0.200	0.200	0.200	0.200
R1	R3	CTI4	0.200	0.200	0.202	0.200	0.200	R19	R14	CTI20	0.200	0.275	0.561	0.236	0.218
R6	R4	CTI5	0.200	0.200	0.200	0.200	0.200	R13	R15	CTI21	0.200	0.200	0.200	0.200	0.200
R3	R5	CTI6	0.200	0.200	0.200	0.200	0.200	R19	R15	CTI22	0.264	0.249	0.613	0.201	0.217
R8	R6	CTI7	0.200	0.200	0.200	0.200	0.200	R2	R16	CTI23	0.200	0.200	0.200	0.200	0.200
R23	R6	CTI8	0.256	0.290	0.530	0.201	0.212	R17	R16	CTI24	0.382	0.354	1.077	0.376	0.363
R5	R7	CTI9	0.200	0.200	0.200	0.200	0.200	R2	R18	CTI25	0.912	0.953	0.409	0.801	0.802
R23	R7	CTI10	0.200	0.211	0.513	0.271	0.203	R15	R18	CTI26	1.005	1.073	0.485	0.976	0.906
R10	R8	CTI11	0.200	0.200	0.199	0.200	0.200	R13	R20	CTI27	0.887	1.042	0.999	0.928	0.849
R7	R9	CTI12	0.200	0.200	0.202	0.200	0.200	R16	R20	CTI28	1.024	1.000	1.097	0.846	0.858
R12	R10	CTI13	0.200	0.200	0.202	0.200	0.200	R11	R22	CTI29	0.944	0.982	0.585	0.841	0.849
R9	R11	CTI14	0.200	0.200	0.227	0.200	0.206	R14	R22	CTI30	0.952	1.071	0.692	0.953	0.904
R14	R12	CTI15	0.200	0.200	0.200	0.200	0.200	R5	R24	CTI31	1.052	1.062	1.029	0.762	0.888
R21	R12	CTI16	0.200	0.200	0.384	0.200	0.200	R8	R24	CTI32	0.859	0.976	0.956	1.011	0.861

of DOCRs coordination problem. Especially, the results obtained for the 8-bus, 9-bus, and IEEE 30-bus test systems were important indicators in terms of distinguishing the performance of the proposed algorithm. AEFA and SSA algorithms violated the relay operating time constraint in the 9-bus test system.

In the IEEE 30-bus test system, ASO, AEFA, and SSA algorithms were insufficient in solving the DOCRs coordination problem due to the violation of the constraints. For all test cases, there were only two algorithms that could successfully overcome the DOCRs coordination problem. These are MRFO and *d*FDB-MRFO. The simulation results of the 8-bus, 9-bus, and IEEE 30-bus test systems showed that the MRFO algorithm was insufficient to convergence the optimum solution due to the premature convergence problem. The main reason for the premature convergence problem was the poor exploration ability of the MRFO algorithm. On the other hand, the *d*FDB-MRFO algorithm has exhibited superior performance in convergence to the global solution with its powerful exploration capability.

Table 27 Operation time (OT) values of all optimization algorithms for Model-IV

Relay	Operation Time (s)	ASO	AEFA	SSA	MRFO	*d*FDB-MRFO	Relay	Operation Time (s)	ASO	AEFA	SSA	MRFO	*d*FDB-MRFO
R1	OT1	0.789	0.849	0.738	0.715	0.689	R13	OT13	0.701	0.769	0.661	0.592	0.594
R2	OT2	0.453	0.596	0.433	0.518	0.392	R14	OT14	0.832	0.814	0.765	0.691	0.672
R3	OT3	0.666	0.773	0.634	0.617	0.583	R15	OT15	0.769	0.840	0.713	0.725	0.673
R4	OT4	0.527	0.624	0.527	0.575	0.476	R16	OT16	0.676	0.767	0.654	0.608	0.582
R5	OT5	0.608	0.746	0.606	0.588	0.539	R17	OT17	–	–	–	–	–
R6	OT6	0.620	0.704	0.636	0.653	0.584	R18	OT18	0.219	0.185	0.705	0.133	0.226
R7	OT7	0.676	0.783	0.652	0.583	0.593	R19	OT19	–	–	–	–	–
R8	OT8	0.623	0.679	0.619	0.572	0.538	R20	OT20	0.216	0.129	0.040	0.199	0.178
R9	OT9	0.565	0.706	0.547	0.474	0.477	R21	OT21	–	–	–	–	–
R10	OT10	0.705	0.715	0.657	0.611	0.585	R22	OT22	0.211	0.092	0.493	0.179	0.232
R11	OT11	0.471	0.688	0.468	0.411	0.378	R23	OT23	–	–	–	–	–
R12	OT12	0.803	0.792	0.771	0.701	0.689	R24	OT24	0.166	0.170	0.121	0.230	0.248

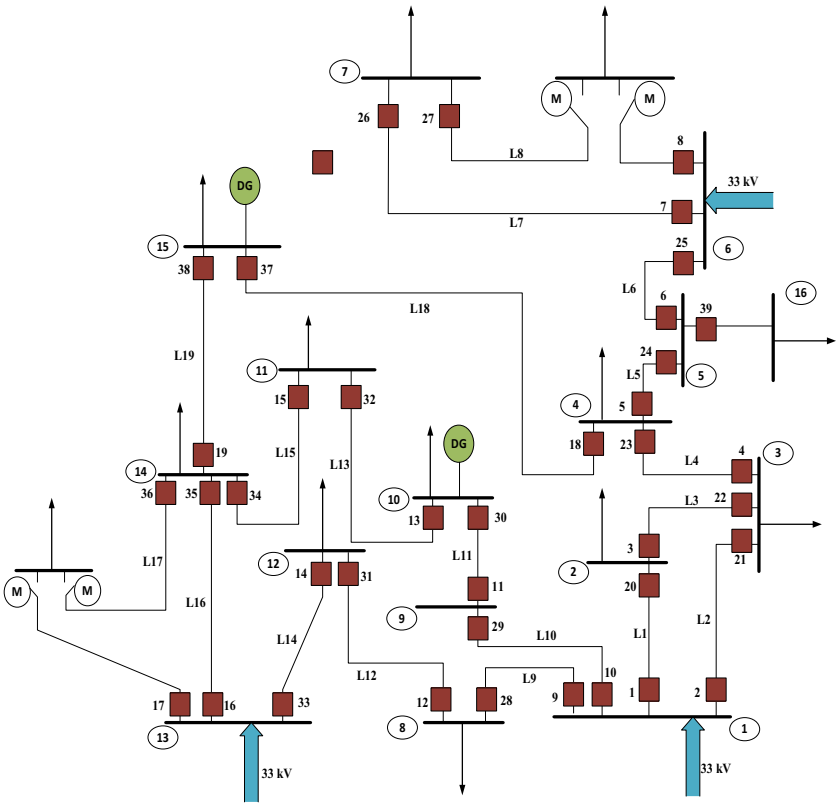


Fig. 11 IEEE 30-bus test system (Model-V)

5.2.2 Literature comparison and statistical analysis

The results of the ASO, AEFA, SSA, MRFO, and the proposed *d*FDB-MRFO methods and the results of other methods in the literature are given for all test systems in Table 32. According to comparison with results in the literature, the minimum relay operation times obtained with the proposed algorithm were 4.78037 s, 3.66934 s, 8.54785 s, 9.92824 s, and 25.94583 s for all cases, respectively. From the comparison results in Table 32, it is obvious that the results of the proposed *d*FDB-MRFO method were better than those of the other heuristic algorithms in the literature.

When the infeasible solutions for Model-III are investigated according to the simualtion results in Table 32, the SOA [13]*, GA [77]*, and the hybrid GA-LP [77]* algorithms have an infeasible solutions, because they do not meet the CTI constraints. The obtained results can be defined as the solution with miscoordination between backup/primary relays: R1/R2, R2/R3, R4/R5, R12/R11, and R9/R14 for the SOA [13]*algorithm, the solution with miscoordination between backup/primary relays: R7/R2, R5/R6, R5/R7, R9/R8, R14/R12, R8/R13, and R9/R14 for GA [76]* algorithm, and the solution with miscoordination between backup/primary relays: R7/R2, R5/R6, R5/R7, R14/R12, and R9/R14 for the hybrid GA-LP [76]* algorithm.

For Model-V, the MCPA [79]* is has an infeasible solution, because it does not meet the CTI constraint. Solution with miscoordination between backup/primary relays: R5/R6, R7/R27, R8/R26, R12/R14, R13/R15, R18/R38, R22/R20, R24/R23, R30/R29, R31/R28, R33/R31, R34/R32, and R37/R23.

The minimum, mean, maximum, and standard deviation values obtained from 30 independent runs of all optimization algorithms used for each test models are given in Table 33. Also, Wilcoxon signed-rank test was applied to make pairwise comparisons between proposed *d*FDB-MRFO and competitive optimization algorithms using the obtained results from algorithms. Wilcoxon signed-rank test results for each test model are presented in Table 34. Accordingly, "R+" represents the score obtained by summing the relevant run number where the competitor algorithm is better than *d*FDB-MRFO. Similarly, "R -" denotes the score obtained by summing the relevant run number where the proposed method is better than the competitor algorithm. Eventually, in order to facilitate the understanding and interpretation of the search performance of the algorithms for five different test models, box-plot graphs have been prepared. Accordingly, box-plot graphs containing the minimum, maximum, and mean/standard deviation margins of the algorithms are shown in Fig. 12.

Upon examination of Table 33, it is clearly seen that MHS algorithms except for ASO in the Model-I test system are successful in finding solutions near the global optimum. On the other hand, the ASO was insufficient in achieving a balance of exploration and exploitation. Based on the mean and

Table 28 Fault current values for IEEE 30-bus test system (Model-V)

Primary relay	Backup relay	$I_{f,primary}$ (A)	$I_{f,backup}$ (A)	Primary relay	Backup relay	$I_{f,primary}$ (A)	$I_{f,backup}$ (A)
R3	R1	4086.7	4086.7	R9	R20	7212.6	1103.5
R4	R2	5411.2	2138.8	R10	R20	7339.3	1095.8
R22	R2	4333	2147	R1	R21	7665.3	698.8
R4	R3	5411.2	3272.5	R9	R21	7212.6	721.2
R21	R3	5411.8	3243.6	R10	R21	7339.3	716.1
R5	R4	4960.8	3001.3	R20	R22	3481.5	3481.5
R18	R4	4719.4	3002.1	R21	R23	5411.8	2193.5
R6	R5	2416	2416	R22	R23	4333	2204.6
R7	R6	5669	1790.9	R18	R24	4719.4	1717.7
R8	R6	5607.7	1774.8	R23	R24	3689.7	1724.2
R27	R7	1472.3	1472.3	R24	R25	2695	2695
R26	R8	1026.8	1026.8	R1	R28	7665.3	1552
R12	R9	5034.9	5034.9	R2	R28	7985.7	1545.8
R11	R10	3457.1	3457.1	R10	R28	7339.3	1538
R13	R11	3727.3	2875	R1	R29	7665.3	1380.6
R14	R12	2906.5	2906.5	R2	R29	7985.7	1375.2
R15	R13	2660.5	2660.5	R9	R29	7212.6	1379
R16	R14	6185.6	1668.1	R29	R30	2518.9	2518.9
R17	R14	7492.9	1641.1	R28	R31	2036.8	2036.8
R19	R15	5445.2	1527.3	R30	R32	2998.8	2149
R35	R15	4222	1533.2	R31	R33	3263.6	3263.6
R36	R15	6420.2	1509.7	R32	R34	2930.4	2930.4
R19	R16	5445.2	1527.3	R17	R35	7492.9	1885.4
R34	R16	5796.6	3123.9	R33	R35	6456.2	1954.5
R36	R16	6420.2	3052.4	R16	R36	6185.6	490.9
R19	R17	5445.2	801.3	R33	R36	6456.2	500.6
R34	R17	5796.6	800.1	R5	R37	4960.8	1961
R35	R17	4222	794	R23	R37	3689.7	1968.5
R38	R18	3133.2	2292.2	R34	R38	5796.6	1886.8
R37	R19	3788.9	2940.9	R35	R38	4222	1896.7
R2	R20	7958.7	1053.9	R36	R38	6420.2	1867.7

standard deviation values, it is understood that the proposed *d*FDB-MRFO method exhibits a more stable and robust search performance. Moreover, Wilcoxon signed-rank test confirmed that for Model-I the proposed method is better than its competitors in converging to the global optimum for all runs compared to the ASO, AEFA, and SSA methods. According to the score given as R+(40)/R-(113) in MRFO vs *d*FDB-MRFO Wilcoxon signed-rank test results, there is the superiority of a dFDB-based MRFO on thirteen runs, similar results on four runs, and superiority of the base MRFO algorithm on thirteen runs. The box-plot charts given in Fig. 12 show that the minimum, maximum, and mean/standard deviation margins are reasonable for optimization algorithms except for the ASO.

According to the results given in Table 33, it is seen that MRFO and *d*FDB-MRFO algorithms converge better than other methods in the Model-II test system. This conclusion is based on the mean and standard deviation values of these algorithms. In addition, Wilcoxon test results applied to demonstrate the statistical validity of the results obtained from the algorithms are given in Table 34. Accordingly, the proposed method outperformed all competitors, except MRFO, in 30 independent runs. Further, there is the superiority of proposed method on twenty runs, similar results on nine runs, and superiority of the base MRFO algorithm on one run. Figure 12 (b) is prepared to visually monitor the search performance of the algorithms for the Model-II test system. Accordingly, the ASO method was insufficient to establish the exploration-exploitation balance, on the contrary, the abilities of other methods to explore promising solution candidates of the search space and avoid local optima were superior.

Table 29 Simulation results of control variables for Model-V

Relay	ASO		AEFA		SSA		MRFO		*d*FDB-MRFO	
	TDS	PCS	TDS	PCS	TDS	PCS	TDS	PCS	TDS	PCS
R1	0.6255	4.697	0.3971	3.115	0.3082	5.555	0.3669	2.430	0.3594	2.563
R2	0.7545	3.422	0.3333	4.446	0.5548	1.500	0.2040	3.018	0.2397	2.467
R3	0.6515	3.490	0.3725	2.360	0.5003	1.500	0.2529	2.846	0.2493	2.965
R4	0.6482	3.997	0.2409	5.505	0.3141	3.632	0.2123	3.329	0.2454	2.500
R5	0.7333	3.533	0.2876	3.715	0.4751	1.557	0.2059	2.220	0.1277	3.961
R6	0.5646	3.730	0.3922	1.799	0.2432	1.500	0.1630	1.505	0.1237	2.149
R7	0.6387	4.695	0.4001	3.926	0.1192	2.389	0.1184	2.406	0.1000	2.859
R8	0.6450	3.625	0.5083	2.377	0.1000	2.479	0.1039	2.224	0.1000	2.294
R9	0.7690	3.671	0.3962	3.711	0.3213	4.881	0.2671	5.008	0.3071	3.771
R10	0.6764	3.283	0.3321	4.470	0.3477	5.047	0.2693	4.232	0.2472	4.793
R11	0.5992	3.293	0.2667	4.719	0.5915	1.500	0.3057	2.217	0.2670	2.917
R12	0.6551	3.976	0.3538	2.964	0.2270	3.313	0.2694	2.796	0.2772	2.432
R13	0.4989	2.970	0.2865	4.430	0.1628	5.337	0.1928	3.716	0.2479	2.539
R14	0.5596	3.447	0.2032	4.676	0.1514	3.499	0.2275	2.186	0.2216	1.997
R15	0.4799	2.497	0.4830	1.500	0.2734	1.526	0.2214	1.731	0.1981	2.065
R16	0.6803	3.495	0.1000	5.835	0.3584	2.111	0.3313	2.328	0.2432	3.552
R17	0.4498	2.776	0.8579	2.683	0.1605	1.655	0.1417	1.757	0.1346	1.753
R18	0.6106	3.861	0.3026	5.179	0.4525	1.572	0.1588	4.635	0.1803	3.424
R19	0.6845	4.667	0.1345	5.949	0.3318	1.669	0.1855	4.158	0.2815	2.078
R20	0.5921	3.166	0.2487	2.231	0.2892	1.500	0.1972	1.816	0.2304	1.513
R21	0.1881	2.272	0.4077	1.652	0.1696	1.636	0.1553	1.500	0.1552	1.503
R22	0.6621	4.721	0.2767	3.202	0.3227	2.461	0.2764	2.106	0.2094	3.776
R23	0.7057	4.046	0.3730	1.500	0.2138	3.580	0.2057	3.066	0.1987	3.247
R24	0.5819	3.972	0.9537	3.931	0.3719	1.509	0.1131	4.188	0.1289	3.784
R25	0.8080	3.768	0.8396	5.189	0.6223	2.098	0.2423	2.414	0.1655	4.297
R26	0.4814	2.203	0.5395	2.144	0.1000	1.500	0.1000	1.500	0.1000	1.500
R27	0.4726	2.753	0.3177	2.022	0.1000	1.500	0.1000	1.500	0.1000	1.500
R28	0.4650	3.059	0.3021	1.624	0.1722	3.471	0.1441	3.385	0.1846	2.658
R29	0.3024	3.525	0.1000	4.300	0.2212	2.382	0.1628	2.751	0.2129	2.084
R30	0.6641	3.478	0.2616	1.582	0.1389	5.691	0.2569	2.292	0.2215	3.190
R31	0.5097	3.169	0.1000	6.000	0.3846	1.583	0.2872	1.974	0.1805	3.764
R32	0.4975	3.939	0.2539	1.899	0.3726	1.500	0.2458	2.733	0.2525	2.770
R33	0.5173	3.853	0.1000	4.032	0.3841	2.586	0.3565	2.240	0.2852	2.957
R34	0.7011	2.838	0.1348	6.000	0.2707	3.922	0.2872	3.223	0.2195	4.729
R35	0.6693	3.607	0.4749	3.248	0.2806	2.300	0.2059	3.013	0.1793	3.213
R36	0.3322	1.652	0.1000	6.000	0.1000	1.501	0.1000	1.500	0.1000	1.500
R37	0.4740	4.235	0.2433	2.665	0.1318	5.251	0.2014	2.739	0.1803	3.124
R38	0.4530	3.853	0.4910	3.706	0.3418	1.502	0.2964	1.712	0.1537	3.680
OF(s)	86.88706		48.72887		32.94089		26.45235		**25.94583**	
MOF	36,178.6895		942,123,230.2518		2470.8590		1553.2687		**1492.0680**	

The minimum and mean value indices given in Table 33 confirmed that the *d*FDB-MRFO method found more stable and better solutions than its competitors for Model-III. According to ASO vs *d*FDB-MRFO and SSA vs *d*FDB-MRFO wilcoxon pairwise comparison results, the proposed algorithm outperformed its competitors in 29 of 30 runs in converging to the minimum value. In addition, it is understood from the Wilcoxon statistical results of AEFA vs *d*FDB-MRFO that the proposed method exhibits better search performance than AEFA in all runs. The fact that the *d*FDB based

Table 30 Comparison of the CTI values for Model-V (T_{backup}-$T_{primary}$)

Relay		Coordination time	ASO	AEFA	SSA	MRFO	*d*FDB-MRFO	Relay		Coordination time	ASO	AEFA	SSA	MRFO	*d*FDB-MRFO
Backup	Primary							Backup	Primary						
R1	R3	CTI1	0.399	0.268	0.329	0.300	0.300	R20	R9	CTI32	5.117	0.713	0.432	0.300	0.300
R2	R4	CTI2	2.254	1.593	0.866	0.419	0.423	R20	R10	CTI33	5.600	0.839	0.340	0.382	0.404
R2	R22	CTI3	1.571	1.630	0.919	0.300	0.300	R21	R1	CTI34	1.003	2.702	0.457	0.369	0.372
R3	R4	CTI4	0.579	0.278	0.358	0.300	0.300	R21	R9	CTI35	0.534	2.436	0.388	0.300	0.300
R3	R21	CTI5	2.405	0.334	1.025	0.634	0.644	R21	R10	CTI36	0.966	2.581	0.301	0.384	0.407
R4	R5	CTI6	0.802	0.624	0.359	0.389	0.462	R22	R20	CTI37	1.115	0.295	0.327	0.300	0.300
R4	R18	CTI7	1.066	0.289	0.390	0.300	0.300	R23	R21	CTI38	4.386	0.294	0.911	0.750	0.763
R5	R6	CTI8	0.800	0.273	0.791	0.300	0.300	R23	R22	CTI39	1.884	0.289	0.300	0.300	0.300
R6	R7	CTI9	2.031	0.294	0.607	0.300	0.300	R24	R18	CTI40	2.922	7.101	0.334	0.421	0.450
R6	R8	CTI10	2.358	0.287	0.659	0.351	0.329	R24	R23	CTI41	2.008	7.421	0.570	0.300	0.300
R7	R27	CTI11	6.563	2.728	0.300	0.300	0.300	R25	R24	CTI42	1.089	0.747	1.136	0.300	0.300
R8	R26	CTI12	8.982	0.298	0.393	0.300	0.300	R28	R1	CTI43	1.422	0.251	0.390	0.300	0.300
R9	R12	CTI13	0.303	0.288	0.581	0.300	0.300	R28	R2	CTI44	1.382	0.295	0.348	0.672	0.623
R10	R11	CTI14	0.315	0.276	0.300	0.300	0.300	R28	R10	CTI45	1.584	0.258	0.300	0.365	0.382
R11	R13	CTI15	0.938	0.282	0.891	0.300	0.300	R29	R1	CTI46	1.086	0.392	0.344	0.321	0.324
R12	R14	CTI16	0.809	0.293	0.325	0.300	0.300	R29	R2	CTI47	1.049	0.444	0.300	0.693	0.647
R13	R15	CTI17	0.320	0.288	0.372	0.300	0.300	R29	R9	CTI48	0.831	0.283	0.339	0.300	0.300
R14	R16	CTI18	2.258	2.031	0.300	0.300	0.300	R30	R29	CTI49	1.925	0.220	0.300	0.300	0.300
R14	R17	CTI19	3.299	0.297	0.884	0.873	0.783	R31	R28	CTI50	0.347	0.185	0.312	0.300	0.300
R15	R19	CTI20	0.303	1.434	0.361	0.350	0.300	R32	R30	CTI51	0.300	0.212	0.305	0.300	0.300
R15	R35	CTI21	0.356	0.296	0.300	0.300	0.389	R33	R31	CTI52	0.331	−0.199	0.306	0.300	0.300
R15	R36	CTI22	2.243	1.648	0.957	0.815	0.834	R34	R32	CTI53	0.325	0.195	0.300	0.300	0.300
R16	R19	CTI23	0.461	0.093	0.419	0.516	0.384	R35	R17	CTI54	3.652	0.868	1.024	0.935	0.854
R16	R34	CTI24	1.070	0.114	0.300	0.300	0.300	R35	R33	CTI55	2.987	2.655	0.300	0.300	0.300
R16	R36	CTI25	2.423	0.311	1.021	0.988	0.929	R36	R16	CTI56	3.710	−1.203	0.507	0.540	0.644
R17	R19	CTI26	3.959	5.443	0.344	0.494	0.665	R36	R33	CTI57	3.903	−1.137	0.324	0.449	0.544
R17	R34	CTI27	6.521	14.378	0.333	0.300	0.300	R37	R5	CTI58	1.337	0.250	0.300	0.508	0.612
R17	R35	CTI28	6.163	13.519	0.406	0.480	0.488	R37	R23	CTI59	0.695	0.272	0.561	0.300	0.300
R18	R38	CTI29	1.656	0.295	0.566	0.300	0.300	R38	R34	CTI60	1.446	3.054	0.350	0.300	0.300
R19	R37	CTI30	1.945	0.179	0.335	0.300	0.300	R38	R35	CTI61	0.883	1.880	0.408	0.465	0.471
R20	R2	CTI31	5.995	0.969	0.446	0.743	0.690	R38	R36	CTI62	2.789	3.274	1.064	0.981	0.923

MRFO algorithm outperforms the MRFO algorithm in 27 of 30 runs is confirmed that the exploration and exploitation capabilities of the proposed method are superior to the base algorithm. The box-plot charts for Model-III (Fig. 12-c) show that the minimum, maximum, and mean/standard deviation margins of the dFDB-MRFO method were reasonable.However, the ASO, AEFA, SSA, and MRFO algorithms were caught in local solution traps.

The simulation results for Model-IV showed that the proposed *d*FDB-MRFO method has a strong exploration ability to avoid premature convergence problems. Other optimization algorithms converged prematurely due to local solution traps and failed to reach the global optimum. Wilcoxon pairwise comparison results confirmed that the proposed method outperforms its competitors in exploration and balanced search capabilities. Accordingly, the proposed method was superior to AEFA in converging to the global optimum value in all independent runs. Based on Wilcoxon test results between ASO vs *d*FDB-MRFO, the proposed method had better results in 24 runs, but worse in 6 runs. Further, the *d*FDB-MRFO method outperformed SSA in 28 runs. Wilcoxon pairwise comparison results between the base MRFO and the proposed method showed that the number of runs in which the algorithms were superior was equal. Although the comparison between the results of the reciprocal runs is similar, it is understood from the simulation results in Table 33 that the proposed method converges to the minimum value in some of these runs. When the box-plot charts shown in Fig. 12(d) are

Table 31 Operation time (OT) values of all optimization algorithms for Model-V

Relay	Operation time (s)	ASO	AEFA	SSA	MRFO	*d*FDB-MRFO	Relay	Operation time (s)	ASO	AEFA	SSA	MRFO	*d*FDB-MRFO
R1	OT1	2.042	1.080	1.096	0.906	0.905	R20	OT20	2.390	0.830	0.806	0.597	0.644
R2	OT2	2.097	1.040	1.145	0.539	0.586	R21	OT21	0.518	0.993	0.411	0.365	0.365
R3	OT3	2.535	1.182	1.306	0.881	0.887	R22	OT22	2.996	0.994	1.016	0.811	0.824
R4	OT4	2.328	1.042	1.073	0.695	0.704	R23	OT23	3.207	1.015	0.898	0.788	0.787
R5	OT5	2.583	1.040	1.168	0.583	0.479	R24	OT24	3.293	5.353	1.163	0.669	0.702
R6	OT6	3.324	1.415	0.799	0.536	0.493	R25	OT25	–	–	–	–	–
R7	OT7	2.442	1.389	0.329	0.328	0.298	R26	OT26	3.949	4.287	0.562	0.562	0.562
R8	OT8	2.162	1.406	0.282	0.280	0.273	R27	OT27	3.331	1.699	0.433	0.433	0.433
R9	OT9	2.302	1.192	1.102	0.928	0.931	R28	OT28	2.674	1.131	1.107	0.906	0.957
R10	OT10	1.914	1.081	1.203	0.854	0.833	R29	OT29	1.641	0.644	0.914	0.738	0.814
R11	OT11	2.488	1.420	1.653	1.021	1.032	R30	OT30	3.135	0.796	0.994	0.940	0.986
R12	OT12	2.439	1.133	0.768	0.839	0.811	R31	OT31	2.141	0.693	1.127	0.932	0.849
R13	OT13	1.867	1.376	0.900	0.823	0.853	R32	OT32	2.616	0.852	1.119	1.007	1.044
R14	OT14	2.684	1.240	0.734	0.825	0.766	R33	OT33	1.668	0.330	1.039	0.911	0.816
R15	OT15	1.975	1.516	0.865	0.745	0.731	R34	OT34	2.063	0.590	0.928	0.895	0.832
R16	OT16	2.137	0.413	0.910	0.874	0.770	R35	OT35	2.605	1.743	0.867	0.726	0.654
R17	OT17	1.179	2.218	0.349	0.314	0.298	R36	OT36	0.761	0.410	0.222	0.222	0.222
R18	OT18	2.319	1.375	1.138	0.672	0.641	R37	OT37	2.182	0.852	0.710	0.715	0.688
R19	OT19	2.669	0.610	0.809	0.678	0.746	R38	OT38	2.229	2.350	0.997	0.917	0.732

examined, it is seen that the proposed method achieves superior minimum and average values compared to its competitors in the optimization of the Model-IV.

From the simulation results obtained as a result of 30 independent runs of the optimization algorithms, it is understood that the ASO, AEFA, and SSA algorithms fail to converge to the global optimum in Model-V test system due to the premature convergence problems. The key reason for the premature convergence problem was the poor exploration ability of these algorithms. On the other hand, MRFO and *d*FDB-MRFO methods exhibited stable and robust search performance. Moreover, the proposed algorithm offered superior mean and standard deviation value. Wilcoxon statistical analysis results are given in the last line of Table 34 clearly revealed the overwhelming superiority of the *d*FDB-MRFO over its competitors in the optimization of the Model-V. Accordingly, the proposed method was superior to the ASO, AEFA, and SSA methods in all runs. Also, the MRFO algorithm was defeated against proposed algorithm in 20 of 30 runs. The box-plot graphs given in Fig. 12e, f showed that the *d*FDB-MRFO is a powerful algorithm for optimizing large-scale power systems.

In summary, the search performance of the optimization algorithms used for five different test models was analyzed in depth. In this context, firstly, the simulation results obtained as a result of 30 independent runs of optimization algorithms were evaluated. Then, using the data obtained from 30 independent runs, the Wilcoxon signed-rank test was applied to control the significant difference between *d*FDB-MRFO and competing algorithms. Finally, box-plot charts were drawn for easy interpretation of the search performance of the algorithms. The results confirmed that the recommended *d*FDB-MRFO for all test models exhibited a more stable and robust search performance than its competitors.

5.2.3 Convergence analysis of the results of directional overcurrent relays coordination problem

In this subsection, the convergence curves of four different optimization algorithms and the proposed *d*FDB-MRFO are presented. Figure 13 shows the convergence curves of *d*FDB-MRFO and other competing algorithms for Model I-V.

When Fig. 13a is examined in detail, it is seen that algorithms other than ASO can converge near the global optimum without difficulty. This search performance is an indication that the algorithms have successfully fulfilled their exploitation task. The ASO was insufficient of balancing exploration and exploitation. Further, the *d*FDB-MRFO algorithm provided high-speed convergence compared to its competitors. From the convergence curves given in Fig. 13(b), it is seen that the algorithms for Model II exhibit similar search performance. This result is based on the best fitness values given on the vertical axis of the graph. Accordingly, the test model search space has a limited number of local solution traps.

Table 32 Comparison results in the literature (Model I-V)

Model-I				Model-II			
Methods	Objective function value (s)	Methods	Objective function values(s)	Methods	Objective function value (s)	Methods	Objective function values(s)
SOMA [71]	8.0101	MDE3 [68]	4.7822	TLBO-MOF [14]	8.7088	MDE1 [68]	3.6694
TLBO-MOF [14]	6.9720	PSO-C [72]	4.7821	TLBO [14]	5.5890	MDE5 [68]	3.6694
TLBO [14]	5.3349	LXPSO-W [72]	4.7807	GA [71]	3.8587	LXPSO-W [72]*	3.6693
GA [71]	5.0761	LXPSO-C [72]	4.7806	SOMA [71]	3.7892	LXPSO-C [72]*	3.6693
Basic DE [68]	4.8422	MDE4 [68]	4.7806	RST [73]	3.7050	MDE3 [68]*	3.6692
LX-PM [74]	4.8368	MDE5 [68]	4.7806	LX-POL [74]	3.7034	MDE4 [68]*	3.6674
LX-POL [74]	4.8368	OCDE1 [16]	4.7806	LX-PM [74]	3.7029	OCDE1 [16]*	3.6674
RST [73]	4.8354	OCDE2 [16]	4.7806	BEX-PM [75]	3.6957	OCDE2 [16]*	3.6674
MDE1 [68]	4.8070	ASO	5.13379	PSO-C [72]	3.6779	ASO	3.92797
BEX-PM [75]	4.7899	AEFA	4.82955	Basic DE [68]	3.6774	AEFA	3.71191
SOMGA [71]	4.7898	SSA	4.81585	SOMGA [71]	3.6745	SSA	3.72290
MDE2 [68]	4.7873	MRFO	4.78037	MDE2 [68]	3.6734	MRFO	3.66934
PSO-W [72]	4.7838	***d*FDB-MRFO**	**4.78037**	PSO-W [72]	3.6702	***d*FDB-MRFO**	**3.66934**
Model-III				Model-IV			
Methods	Objective function value (s)	Methods	Objective function values(s)	Methods	Objective function value (s)	Methods	Objective function values(s)
GA [76]	11.4462	CSA [76]	9.7620	GSA [29]	14.7384	Hybrid IIWO [25]	8.870
GA [77]*	11.0010	Hybrid BBO-LP [17]	8.7550	IWO [25]	14.6810	ASO	11.29550
Hybrid GA-LP [77] *	10.9490	SOA [13]*	8.4270	IDE [29]	14.0448	AEFA	12.42146
BBO [17]	10.5490	ASO	10.67216	IIWO [25]	12.2070	SSA	11.43988
Jaya [78]	10.2325	AEFA	12.13057	SQP [29]	10.2030	MRFO	10.37470
DJaya [78]	9.9661	SSA	11.26083	Hybrid GSA-SQP [29]	8.8923	***d*FDB-MRFO**	**9.92824**
OJaya [78]	9.8520	MRFO	10.75494				
Hybrid GA-NLP [76]	9.7631	***d*FDB-MRFO**	**8.54785**				

Model-V			
Methods	Objective function value (s)	Methods	Objective function values(s)
GSA [29]	51.7747	MCPA [79]*	23.85000
SQP [29]	41.0390	ASO	86.88706
PSO [79]	39.1800	AEFA	48.72887
SOA [79]	33.7700	SSA	32.94089
GA [79]	28.0190	MRFO	26.45235
Hybrid GSA–SQP [29]	26.8258	***d*FDB-MRFO**	**25.94583**

*Infeasible solution

Convergence curves are given in Fig. 13(c) illustrates that the *d*FDB-MRFO method is better than other methods in converging to the optimal solution for the Model-III test system. Accordingly, the proposed algorithm effectively fulfilled the exploration task to converge to the global solution. On the other hand, it is seen that the MRFO algorithm converges prematurely and is worse than the ASO and *d*FDB-MRFO methods in terms of convergence accuracy. Comparison between *d*FDB-MRFO and MRFO in terms of best fitness value for Model-III showed that the FDB selection method improved the exploration ability of the MRFO algorithm. Upon examination, the convergence curves of the algorithms for Model-IV demonstrate that the proposed *d*FDB-MRFO strongly exhibited the exploitation-exploration balance.

Table 33 Minimum, mean, maximum, and standard deviation of the optimization algorithm simulation results for all test models

Methods	Model-I	Model-II	Model-III	Model-IV	Model-V
ASO					
Min	53.25686	214,952.84	230.1577	324.9641	36,178.689
Mean	64.23509	214,961.40	284.7799	380.4825	2,230,946
Max	85.69570	214,971.90	351.3214	424.1936	45,393,906
Std	7.47791	5.26935	30.7673	23.05435	8,198,807
AEFA					
Min	46.88991	214,949.22	296.4023	390.2518	9.42E+08
Mean	47.25651	214,949.50	379.3708	445.8023	2.93E+09
Max	47.82490	214,949.90	436.4400	486.5847	6.26E+09
Std	0.19723	0.17931	34.5272	22.7017	1.17E+09
SSA					
Min	46.64826	214,949.38	257.9658	339.0981	2470.859
Mean	47.96288	214,950	333.3128	454.198	4650.662
Max	50.434	214,950.7	387.3683	563.1821	7553.426
Std	0.86854	0.36621	35.50345	54.75835	1392.751
MRFO					
Min	46.00785	214,948.61	242.9288	274.1420	1553.268
Mean	46.01146	214,948.60	316.4743	346.8121	1994.982
Max	46.04740	214,948.64	471.2668	471.2668	2296.825
Std	0.00890	0.00082	36.9091	48.65744	191.6367
*d*FDB-MRFO					
Min	**46.00785**	**214,948.61**	**151.1337**	**251.0414**	**1492.068**
Mean	46.00829	214,948.60	212.9988	341.6082	1896.711
Max	46.01340	214,948.64	293.0195	428.0358	2189.408
Std	0.00139	0.00082	34.75019	42.02937	154.5468

Also, the proposed method reduces the number of iterations required to obtain the solution compared to other methods. The curves in Fig. 13(e) present the convergence performance of the algorithms in a non-convex, high-dimensional and large-scale test system (Model-V). When the convergence graphs are examined in detail, it is seen that the *d*FDB-MRFO provides an overwhelming advantage over its competitors in terms of convergence accuracy and speed. This points out that the proposed algorithm ensures a powerful balance between exploration and exploitation that results in high local optima avoidance. In summary, the convergence curves revealed that the proposed method outperforms other algorithms in terms of convergence speed, best fitness value, and robustness for all test models. These results are strong evidence that the *d*FDB-MRFO algorithm is a powerful method for solving real-world engineering problems.

6 Conclusions

As a result of the studies reported in this article, unique information and important contributions have been provided for

Table 34 Wilcoxon signed-rank test results for different test models

Test Model	ASO vs *d*FDB-MRFO			AEFA vs *d*FDB-MRFO			SSA vs *d*FDB-MRFO			MRFO vs *d*FDB-MRFO		
	R+	R-	*p* value	R+	R-	*p* value	R+	R-	*p* value	R+	R-	*p* value
Model-I	0	465	2×10^{-6}	0	465	2×10^{-6}	0	465	2×10^{-6}	40	113	83.843×10^{-3}
Model-II	0	465	2×10^{-6}	0	465	2×10^{-6}	0	465	2×10^{-6}	14	217	41.6×10^{-5}
Model-III	4	461	3×10^{-6}	0	465	2×10^{-6}	1	464	2×10^{-6}	9	456	4×10^{-6}
Model-IV	72	393	9.63×10^{-4}	0	465	2×10^{-6}	4	461	3×10^{-6}	217	248	0.749871
Model-V	0	465	2×10^{-6}	0	465	2×10^{-6}	0	465	2×10^{-6}	134	331	4.2767×10^{-2}

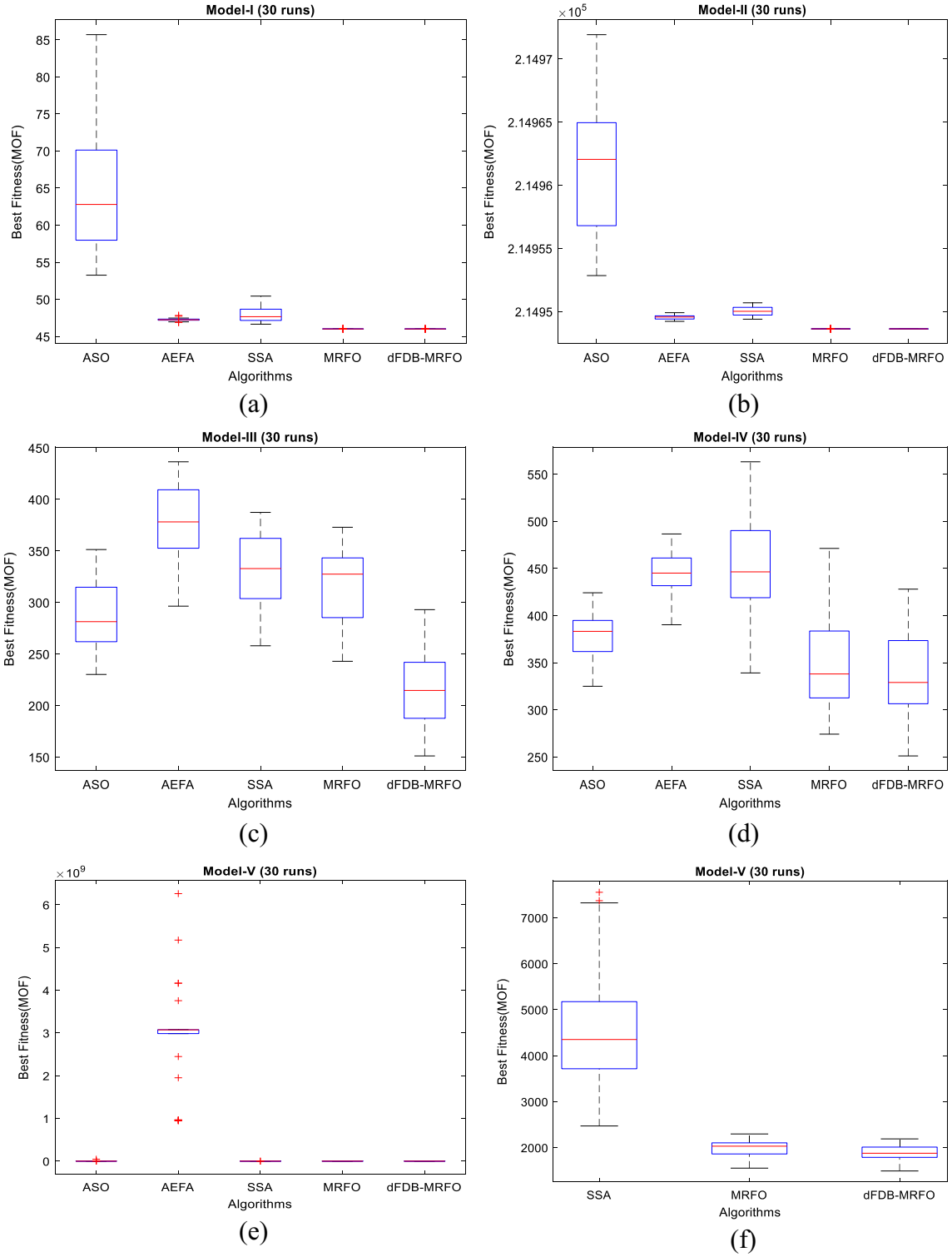


Fig. 12 Box-plot characteristics of all algorithms for 30 runs: **a** Model-I, **b** Model-II, **c** Model-III, **d** Model-IV, **e** Model-V and **f** Model-V (Zoom version)

researchers on meta-heuristic search algorithms, optimization studies, and coordination of overcurrent relays in power systems. In this research study, notable targets were achieved in two areas. The first concerns the methods and algorithms developed as a result of studies on meta-heuristic search algorithms. The selection method *d*FDB was successfully developed for use in the design of meta-heuristic search algorithms. Experimental data and statistical analysis results showed that dynamizing of the *w*-coefficient improved the performance of the selection method. Because of its dynamic

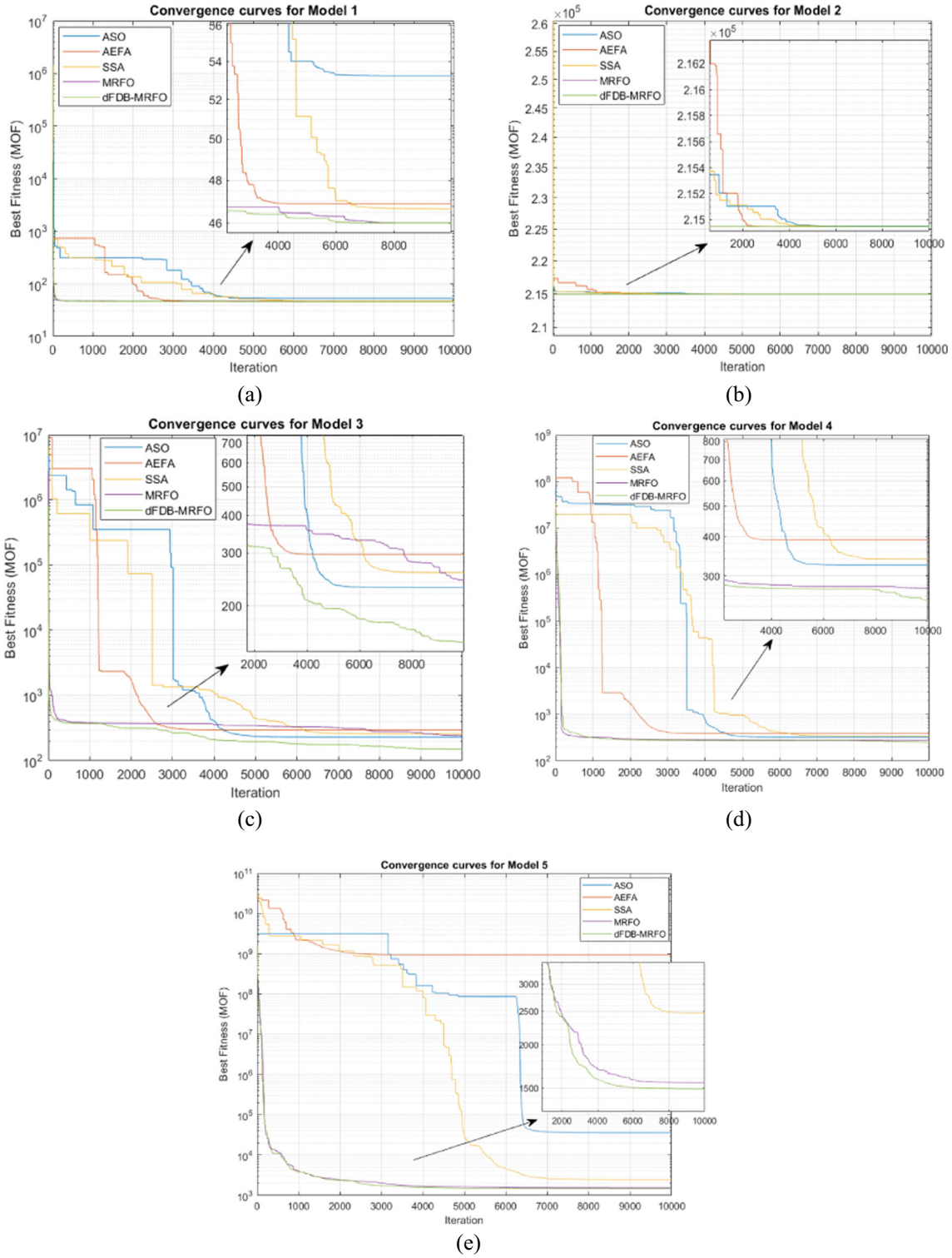


Fig. 13 Convergence curves of the fitness function of all optimization algorithms for Model I-V

adaptability, the *d*FDB selection method enabled the algorithms to perform more effectively in solving different types of optimization problems.

This success of the *d*FDB selection method will enable it to be widely used in MHS algorithm design studies. In addition, as in this study, the *d*FDB selection method can be used to improve the performance of existing MHS algorithms. Moreover, researchers can carry out development work to improve the performance of the *d*FDB method. When evaluated from these aspects, the *d*FDB

selection method will be an important research topic for MHS algorithm studies. The *d*FDB-based MRFO algorithm proposed in this paper is also important for researchers working on optimization problems. A remarkable improvement was achieved in the performance of the MRFO algorithm designed with the *d*FDB method. Researchers will be able to use the *d*FDB-MRFO algorithm to find optimal solutions for real-world optimization problems. The second issue in the evaluation of the results obtained from the research is that of the coordination of the overcurrent relays. The results of all of the studies conducted in five different test systems for the optimization of the DOCRs coordination problem were more successful than those in the literature studies. The *d*FDB selection method and the *d*FDB-MRFO algorithm developed in this article demonstrated a superior performance in more than 90 test problems of different types and dimensions, and in the optimization of the DOCRs coordination problem. In the future, these two methods can be used to design many MHS algorithms and to solve many optimization problems.

In summary, by using the *d*FDB method developed in the article, the guide selection process of meta-heuristic search algorithms can be redesigned and thus their search performance can be improved. The *d*FDB-MRFO algorithm proposed in the article can be used for the optimization of single-objective global optimization problems. Two of the input parameters of the *d*FDB selection method suggested in the article are *FEs* (the number of function evaluations) and *maxFEs* (the maximum number of function evaluations) parameters. In MHS algorithms where *FEs* and *maxFEs* parameters are not used, the current iteration number can be used instead of *FEs* and the maximum iteration number can be used instead of *maxFEs* in order to apply the *d*FDB method. The analysis results presented in this article indicate that *d*FDB-MRFO is a competitive algorithm. However, as claimed in the No Free Lunch theorem [80], there is no single optimization technique that can solve all optimization problems. Therefore, researchers should determine the most appropriate MHS method to optimize their problems. For this, they must conduct extensive research using competitive MHS algorithms.

Owing to the superior outcomes of the current study, the proposed method can be utilized for a number of power system applications like optimal reactive power flow, automatic generation control, dynamic combined economic emission dispatch, energy hub management, and short-term forecasting of electric energy consumption.

Source codes of the *d*FDB selection method and *d*FDB-MRFO algorithm can be accessed at this link: https://se.mathworks.com/matlabcentral/fileexchange/96113-dfdb-mrfo-a-powerful-meta-heuristic-optimization-algorithm

References

1. Kahraman HT, Aras S, Gedikli E (2020) Fitness-distance balance (FDB): a new selection method for meta-heuristic search algorithms. Knowl-Based Syst 190:105169
2. Kahraman HT, Aras S (2019) Investigation of the Most effective meta-heuristic optimization technique for constrained engineering problems. In: The International Conference on Artificial Intelligence and Applied Mathematics in Engineering, Springer, Cham, pp 484–501
3. Zhao W, Zhang Z, Wang L (2020) Manta ray foraging optimization: an effective bio-inspired optimizer for engineering applications. Eng Appl Artif Intell 87:103300
4. Zeineldin HH, El-Saadany EF, Salama MMA (2006) Optimal coordination of overcurrent relays using a modified particle swarm optimization. Electr Power Syst Res 76(11):988–995
5. Mahari A, Seyedi H (2013) An analytic approach for optimal coordination of overcurrent relays. IET Gener Transm Distrib 7(7): 674–680
6. Korashy A, Kamel S, Youssef AR, Jurado F (2019) Modified water cycle algorithm for optimal direction overcurrent relays coordination. Appl Soft Comput 74:10–25
7. Farzinfar M, Jazaeri M, Razavi F (2014) A new approach for optimal coordination of distance and directional over-current relays using multiple embedded crossover PSO. Int J Electr Power Energy Syst 61:620–628
8. So CW, Li KK, Lai KT, Fung KY (1997) Application of genetic algorithm for overcurrent relay coordination. In: International Conf. of Developments in Power Syst. Protection, pp 66–69
9. Mansour MM, Mekhamer SF, El-Kharbawe N (2007) A modified particle swarm optimizer for the coordination of directional overcurrent relays. IEEE Trans Power Deliv 22(3):1400–1410
10. Razavi F, Abyaneh HA, Al-Dabbagh M, Mohammadi R, Torkaman H (2008) A new comprehensive genetic algorithm method for optimal overcurrent relays coordination. Electr Power Syst Res 78(4):713–720
11. Shih MY, Enríquez AC, Trevino LMT (2014) On-line coordination of directional overcurrent relays: performance evaluation among optimization algorithms. Electr Power Syst Res 110:122–132
12. Moravej Z, Adelnia F, Abbasi F (2015) Optimal coordination of directional overcurrent relays using NSGA-II. Electr Power Syst Res 119:228–236
13. Amraee T (2012) Coordination of directional overcurrent relays using seeker algorithm. IEEE Trans Power Deliv 27(3):1415–1422
14. Singh M, Panigrahi BK, Abhyankar AR (2013) Optimal coordination of directional over-current relays using teaching learning-based optimization (TLBO) algorithm. Int J Electr Power Energy Syst 50: 33–41
15. Moirangthem J, Krishnanand KR, Dash SS, Ramaswami R (2013) Adaptive differential evolution algorithm for solving non-linear coordination problem of directional overcurrent relays. IET Gener Transm Distrib 7(4):329–336
16. Chelliah TR, Thangaraj R, Allamsetty S, Pant M (2014) Coordination of directional overcurrent relays using opposition based chaotic differential evolution algorithm. Int J Electr Power Energy Syst 55:341–350
17. Albasri FA, Alroomi AR, Talaq JH (2015) Optimal coordination of directional overcurrent relays using biogeography-based optimization algorithms. IEEE Trans Power Deliv 30(4):1810–1820
18. Kim CH, Khurshaid T, Wadood A, Farkoush SG, Rhee SB (2018) Gray wolf optimizer for the optimal coordination of directional overcurrent relay. J Electr Eng Technol 13(3):1043–1051
19. Rajput VN, Pandya KS (2017) Coordination of directional overcurrent relays in the interconnected power systems using

effective tuning of harmony search algorithm. Sustain Comput Inform Syst 15:1–15
20. El-Fergany AA, Hasanien HM (2017) Optimized settings of directional overcurrent relays in meshed power networks using stochastic fractal search algorithm. Int Trans Electr Energy Syst 27(11): e2395
21. Zellagui M, Benabid R, Boudour M, Chaghi A (2014) Application of firefly algorithm for optimal coordination of directional overcurrent protection relays in presence of series compensation. J Autom Syst Eng:92–107
22. Hussain MH, Musirin I, Abidin AF, Rahim SRA (2014) Solving directional overcurrent relay coordination problem using artificial bees colony. Int J Electr Electron Sci Eng 8(5):705–710
23. El-Fergany A (2016) Optimal directional digital overcurrent relays coordination and arc-flash hazard assessments in meshed networks. Int Trans Electr Energy Syst 26(1):134–154
24. Saha D, Datta A, Das P (2016) Optimal coordination of directional overcurrent relays in power systems using symbiotic organism search optimisation technique. IET Gener Transm Distrib 10(11): 2681–2688
25. Srinivas STP (2019) Application of improved invasive weed optimization technique for optimally setting directional overcurrent relays in power systems. Appl Soft Comput 79:1–13
26. Ahmadi SA, Karami H, Sanjari MJ, Tarimoradi H, Gharehpetian GB (2016) Application of hyper-spherical search algorithm for optimal coordination of overcurrent relays considering different relay characteristics. Int J Electr Power Energy Syst 83:443–449
27. Korashy A, Kamel S, Jurado F, Youssef AR (2019) Hybrid whale optimization algorithm and Grey wolf optimizer algorithm for optimal coordination of direction overcurrent relays. Electr Power Compon Syst 47(6–7):644–658
28. Khurshaid T, Wadood A, Farkoush SG, Kim CH, Yu J, Rhee SB (2019) Improved firefly algorithm for the optimal coordination of directional overcurrent relays. IEEE Access 7:78503–78514
29. Radosavljević J, Jevtić M (2016) Hybrid GSA-SQP algorithm for optimal coordination of directional overcurrent relays. IET Gener Transm Distrib 10(8):1928–1937
30. Zellagui M, Abdelaziz AY (2015) Optimal coordination of directional over-current relays using hybrid PSO-DE algorithm. International Electrical Engineering Journal (IEEJ) 6(4):1841–1849
31. Radosavljević J (2018) Metaheuristic optimization in power engineering. Institution of Engineering and Technology
32. Corrêa R, Cardoso G Jr, de Araújo OC, Mariotto L (2015) Online coordination of directional overcurrent relays using binary integer programming. Electr Power Syst Res 127:118–125
33. Sulaiman MH, Mustaffa Z, Saari MM, Daniyal H (2020) Barnacles mating optimizer: a new bio-inspired algorithm for solving engineering optimization problems. Eng Appl Artif Intell 87:103330
34. Kamboj VK, Nandi A, Bhadoria A, Sehgal S (2020) An intensify Harris hawks optimizer for numerical and engineering optimization problems. Appl Soft Comput 89:106018
35. Zhang Y, Jin Z (2020) Group teaching optimization algorithm: a novel metaheuristic method for solving global optimization problems. Expert Syst Appl 148:113246
36. Faramarzi A, Heidarinejad M, Stephens B, Mirjalili S (2019) Equilibrium optimizer: a novel optimization algorithm. Knowl-Based Syst 191:105190
37. Mohamed AW, Mohamed AK (2019) Adaptive guided differential evolution algorithm with novel mutation for numerical optimization. Int J Mach Learn Cybern 10(2):253–277
38. Arora S, Singh S (2019) Butterfly optimization algorithm: a novel approach for global optimization. Soft Comput 23(3):715–734
39. Tang D, Liu Z, Yang J, Zhao J (2018) Memetic frog leaping algorithm for global optimization. Soft Comput 1-29
40. Chen X, Xu B (2018) Teaching-learning-based artificial bee colony. In: International Conference on Swarm Intelligence. Springer, Cham, pp 166–178
41. Pierezan J, Coelho LDS (2018) Coyote optimization algorithm: a new metaheuristic for global optimization problems. In: 2018 IEEE congress on evolutionary computation (CEC), pp 1–8
42. Civicioglu P, Besdok E, Gunen MA, Atasever UH (2018) Weighted differential evolution algorithm for numerical function optimization: a comparative study with cuckoo search, artificial bee colony, adaptive differential evolution, and backtracking search optimization algorithms. Neural Comput & Applic:1–15
43. Abedinpourshotorban H, Shamsuddin SM, Beheshti Z, Jawawi DN (2016) Electromagnetic field optimization: a physics-inspired metaheuristic optimization algorithm. Swarm Evol Comput 26:8–22
44. Punnathanam V, Kotecha P (2016) Yin-Yang-pair optimization: a novel lightweight optimization algorithm. Eng Appl Artif Intell 54: 62–79
45. Askarzadeh A (2016) A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. Comput Struct 169:1–12
46. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
47. Salimi H (2015) Stochastic fractal search: a powerful metaheuristic algorithm. Knowl-Based Syst 75:1–18
48. Shareef H, Ibrahim AA, Mutlag AH (2015) Lightning search algorithm. Appl Soft Comput 36:315–333
49. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl-Based Syst 89:228–249
50. Cheng MY, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. Comput Struct 139:98–112
51. Gandomi AH (2014) Interior search algorithm (ISA): a novel approach for global optimization. ISA Trans 53(4):1168–1183
52. Civicioglu P (2013) Backtracking search optimization algorithm for numerical optimization problems. Appl Math Comput 219:8121–8144
53. Civicioglu P (2012) Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm. Comput Geosci 46:229–247
54. Yang XS, Deb S (2009) Cuckoo search via Lévy flights. In: 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), pp 210–214
55. Karaboga D, Akay B (2009) A comparative study of artificial bee colony algorithm. Appl Math Comput 214(1):108–132
56. Baloochian H, Ghaffary HR, Balochian S (2020) Metaheuristic anopheles search algorithm. Evolutionary Intelligence. https://doi.org/10.1007/s12065-019-00348-w
57. Zhao W, Wang L, Zhang Z (2019) Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. Knowl-Based Syst 163:283–304
58. Anita, Yadav A (2019) AEFA: Artificial electric field algorithm for global optimization. Swarm Evol Comput 48:93–108
59. Wang GG (2018) Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems. Memetic Computing:1–14
60. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw 114:163–191
61. Mirjalili S, Gandomi AH (2017) Chaotic gravitational constants for the gravitational search algorithm. Appl Soft Comput 53:407–419
62. Mittal H, Pal R, Kulhari A, Saraswat M (2016) Chaotic kbest gravitational search algorithm (ckgsa). In: 2016 Ninth International Conference on Contemporary Computing (IC3), pp 1–6
63. Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. Knowl-Based Syst 96:120–133

64. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
65. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179(13):2232–2248
66. Liang JJ, Qu BY, Suganthan PN (2013) Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore
67. Awad NH, Ali MZ, Liang JJ, Qu BY, Suganthan PN (2016) Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization. Technical Report
68. Thangaraj R, Pant M, Deep K (2010) Optimal coordination of overcurrent relays using modified differential evolution algorithms. Eng Appl Artif Intell 23(5):820–829
69. Singh M, Panigrahi BK, Abhyankar AR, Das S (2014) Optimal coordination of directional over-current relays using informative differential evolution algorithm. J Comput Sci 5(2):269–276
70. Mohammadi R, Abyaneh HA, Rudsari HM, Fathi SH, Rastegar H (2011) Overcurrent relays coordination considering the priority of constraints. IEEE Trans Power Deliv 26(3):1927–1938
71. Dipti (2007) Hybrid genetic algorithms and their applications. PhD thesis, Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee, India
72. Deep K, Bansal JC (2009) Optimization of directional overcurrent relay times using laplace crossover particle swarm optimization (LXPSO). In: Proc nature & biologically inspired computing, 2009. NaBIC 2009. World Congress, pp 288–293
73. Deep K, Birla D, Maheshwari R, Gupta H, Takur M (2006) A population based heuristic algorithm for optimal relay operating time. World Journal of Modelling and Simulation 3:167–176
74. Thakur M (2007) New real coded genetic algorithms for global optimization. Ph.D. Thesis, India: Department of Mathematics, Indian Institute of Technology Roorkee
75. Thakur M, Kumar A (2016) Optimal coordination of directional over current relays using a modified real coded genetic algorithm: a comparative study. Int J Electr Power Energy Syst 82:484–495
76. Darji GU, Patel MJ, Rajput VN, Pandya KS (2015) A tuned cuckoo search algorithm for optimal coordination of Directional Overcurrent Relays. In 2015 International Conference on Power and Advanced Control Engineering (ICPACE), pp 162–167
77. Noghabi A, Sadeh J, Mashhadi H (2009) Considering different network topologies in optimal overcurrent relay coordination using a hybrid GA. IEEE Trans Power Deliv 24(4):1857–1863
78. Yu J, Kim CH, Rhee SB (2019) Oppositional Jaya algorithm with distance-adaptive coefficient in solving directional over current relays coordination problem. IEEE Access 7:150729–150742
79. Korashy A, Kamel S, Youssef A. R, Jurado F (2019) Most valuable player algorithm for solving direction overcurrent relays coordination problem. In 2019 International conference on innovative trends in computer engineering (ITCE), pp 466–471
80. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.