IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 July 2022, accepted 2 August 2022, date of publication 8 August 2022, date of current version 15 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3197290

RESEARCH ARTICLE

Modified Fitness Dependent Optimizer for Solving Numerical Optimization Functions

JUMAA FATIH SALIH, HARDI M. MOHAMMED, AND ZRAR KH. ABDUL

Applied Computer Department, Charmo University, KRG, Sulaymaniyah, Chamchamal 46023, Iraq

Corresponding author: Zrar Kh. Abdul (eng.zrar1394@gmail.com)

ABSTRACT The Fitness Dependent Optimizer (FDO) is a recent metaheuristic algorithm that was developed in 2019. It is one of the metaheuristic algorithms that has been used by researchers to solve various applications especially for engineering design problem. In this paper, a comprehensive survey conducted about FDO and its applications. Consequently, despite of having competitive performance of FDO, it has two major problems including low exploitation and slow convergence. Therefore, a modification of FDO (MFDO) is proposed for solving FDO issues. MFDO used two methods to enhance the performance of FDO: firstly, optimizing the range of weight factor between 0 and 0.2 which is used for finding fitness weight. Secondly, using sine cardinal mathematical function to update fitness weight and pace which is referred to the speed of the bees. To evaluate the performance of MFDO, 19 classical benchmark functions and CEC2019 benchmark functions are used. MFDO compared against all the enhancement of FDO and also it is compared with Grey Wolf Optimization (GWO), Chimp Optimization Algorithm (ChOA), Genetic Algorithm (GA), and Butterfly Optimization Algorithm (BOA). Statistical results proved that MFDO achieved significant performance compared to other algorithms. Finally, MFDO is used to solve three applications: Welded Beam Design (WDB), Pressure Vessel Design (PVD), and Spring Design Problem. Results proved that MFDO outperformed well in solving these applications against FDO, Gravitational Search Algorithm (GSA), GA, and Grasshopper Optimization Algorithm (GOA).

INDEX TERMS Modified fitness dependent optimizer, fitness dependent optimizer, metaheuristic algorithm, optimization algorithm.

I. INTRODUCTION

Metaheuristic algorithms are optimization methods that aim to find the best solution to a problem. In other words, metaheuristics are a set of intelligent strategies to improve the efficiency of heuristic procedures [1].They can be divided into four main categories; swarm intelligence based, evolution-based, physics-based and human-related algorithms [2].

Swarm Intelligence Algorithms (SIA) are used by many researchers in a variety of fields to solve different problems. SIAs are part of the nature- inspired based algorithms. The ability of natural swarm-based systems is inspired by the behavior of some social living beings, such as bees, birds, and ants. Because of having several agents to form a population, SIAs have high flexibility and, high efficiency to produce good results in terms of speed, cost, and robustness [3]. In SI, simple rules are performed by agents, and there is no centralized control structure in order to predict individual agent's behavior [4]. SI's have two main concepts which are labor division and self-organization. Self-organization refers to the ability to create procedures for developing agents without depending on external sources. On the other hand, the labor division represents the work process into spited number of tasks [5].

Metaheuristic algorithms have been used to solve different optimization problems. Finding the maximum or minimum value of some function are the aim of using these algorithms: the minimum time to make a specific trip, the minimum cost for doing a mission, and so on [6]. Nonetheless, these algorithms also have some inadequacies in terms of finding global optima, making a tradeoff between its two main essential aspects exploration and exploitation.

The associate editor coordinating the review of this manuscript and approving it for publication was Huaqing Li.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

Metaheuristic algorithms due to having high performances are utilized to solve real-world problems apparently electromagnetics problem [7], engineering design problems [8], constrained optimization problems [9], economic problem [10], medical problem [11] and task planning problems [12]. They are applied successfully to various engineering and sciences problems, e.g. electrical engineering (to find the optimal solution for power generation), in civil engineering (to design the bridges, buildings), data mining (classification, prediction, clustering, system modeling), communication (radar design, networking) [13].

The nature-inspired metaheuristic algorithms can be based on intelligence swarm [14]. Genetic Algorithm(GA) proposed by J Holland [15]. It is very common and have been used to solve many application problems. After GA, different algorithms had been proposed such as Particle Swarm Optimization (PSO) [16], Ant Colony Optimization (ACO) [17], Artificial Bee Colony (ABC) [18], Bat Algorithm (BA) [19], Cuckoo Search (CS) [20], Firefly Algorithm (FA) [21], Krill Herd (KH) [22]. none the less Recent metaheuristic algorithms, namely whale optimization algorithm (WOA) [23], Rain Water Algorithm (RWA) [24], Grasshopper Optimization Algorithm (GOA) [25], Farmland Fertility (FF) [26], Emperor Penguins Colony (EPC) [27],Harris hawks optimization (HHO) [28], Learner Performance based Behavior algorithm (LPB) [29], FDO.

Initially, the FDO was presented by Jaza and Tarik [30]. To investigate the reproductive technique, and their mass decision making behaviors of bee swarm(mimicking the collective behavior of bee swarm in finding new hives). The proposed algorithm somewhat based on PSO. Thus, in agent position updating mechanism mimics PSO algorithm but in various way. FDO due to the use of fitness weight (*fw*) technique are its power of exploration and exploitation, easily restricted into local optima, quicker convergence speed and solve some real-world problems better than other metaheuristic algorithms. (FDO was able to provide very competitive results compared with some metaheuristic algorithms and it has also applied to a several real problems showed a substantial betterment).

FDO has a more competitive result compared with other metaheuristic algorithms, and it has also been applied to solve several real-world problems namely, aperiodic antenna array designs (AAAD), Frequency modulated sound waves (FM) [30], Rainfall data [31], One-Dimensional Bin Packing Problem [32], Pressure Vessel Design(PVD), task assignment problem [33].

Metaheuristic algorithms have issues in terms of finding optimum solution. Therefore, metaheuristic algorithms have abilities to resolve intricate optimization problems [34]. Nevertheless, metaheuristic algorithms based on the stochastic mechanism have ability to gain global optima and overcome local optima [35]. Metaheuristic algorithms depend on the including the interaction among swarm could ensure the efficient of exploration [36] and balancing between exploration and exploitation [37]. Based on the information collected in this survey, there are several papers that has been done about FDO, which can be seen in Figure (1).

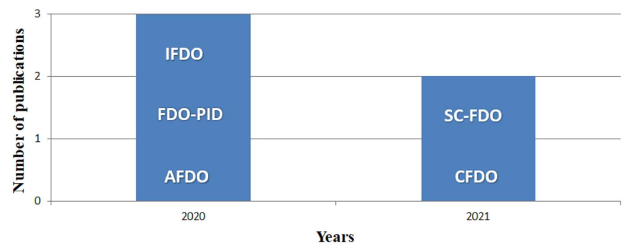

FIGURE 1. FDO and its variations.

The main contribution of this paper is to collect all the researches in a survey in order to clarify all the methods and techniques for researchers. This survey focused on FDO in different aspects: FDO detail, limitations, modifications, hybridization, applications and its results. The second contribution is proposing modified FDO (MFDO) in order to solve the issues of FDO which are poor exploitability and having low convergence speed. Finally, FDO is used to solve real world application:

The residue of the paper outline begins with describing FDO, its characteristics, and limitations followed by providing different FDO modifications, which have been used to solve various problems. Following, different applications of FDO are presented. After that, the proposed MFDO is presented. The results of the MFDO are evaluated against the original FDO and other common algorithms. Finally, conclusion is presented with future works on FDO and MFDO.

II. FDO ALGORITHM

FDO is a recent designed swarm intelligent algorithm which was proposed by Abdullah and Rashid [30]. It is based on the bee swarming characteristics during proliferation process for finding better hives. The position updating mechanism of FDO slightly mimics of the PSO algorithm. After initializing population of scout bee randomly in the search space, finding best hives is the main purpose of the search agents which are the scout bees. The scout bees ignore the previous solution if it is not better than current solution [33]. Despite of ignoring the low solution, scout bees change their position based on the previous best position if the solution does not improve. Scout bee population is represented as Xi (i=1, 2, ..., n) [31], [38].

Two methods are used to search inside the search space by scout bees: *fw* and scout bee movement process. In this algorithm, the scout bee updates its current position based on the *pace* to obtain better solution. The scout bee movement is calculated as follows:

$$X_{i+1} = X_{i,t} + pace \tag{1}$$

where x denotes the artificial search agent (scout bee), i represents the current search agent, t is the current iteration. The movement rate and direction of the artificial search agent can be discovered by the *pace*. *pace* is also depends on the

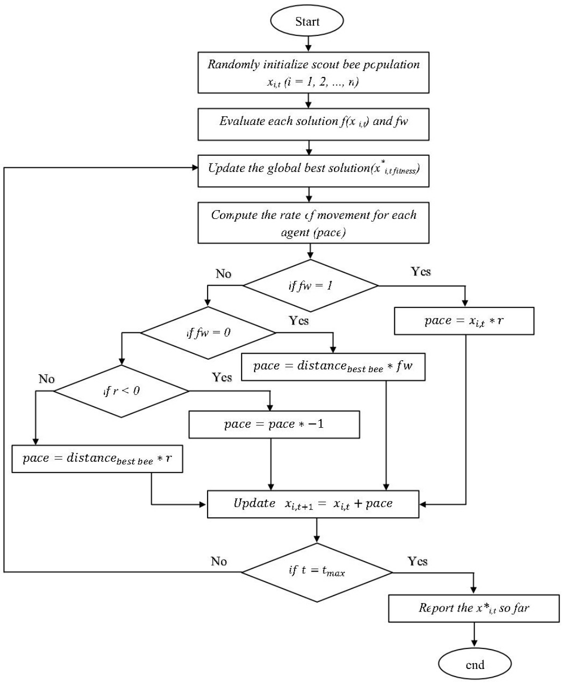

FIGURE 2. Flowchart of the FDO [32].

value of *fw* and randomization technics. Equation (2) presents the calculation of *fw* based on best fitness and current fitness:

$$fw = \left[\frac{X^*_{i,t\ fitness}}{X_{i,t\ fitness}}\right] - wf \tag{2}$$

where $X^*_{i,t\ fitness}$ denotes the fitness function of the global best solution, $X_{i,t\ fitness}$ is the current best solutions of the scout bee, and *wf* denotes the weight factor and it has the rang of [0, 1]. Furthermore, FDO has a random number *r* which is based on the *Levy flight* method. This random variable is between −1 and 1. Furthermore, FDO has three different conditions for calculating *pace* value based on the *fw* and *r*. Equation (3) represents the conditions of *pace* value. However, *wf* does not have effect on the Equation (2) when *wf* is equal to zero. The *pace* value is calculated by multiplying current position with r. As can be seen in Equation (3), if $wf = 1$. Then, if *wf* equal to zero *pace* can be calculated by multiplying $\text{distance}_{\text{bestbee}}$ with *r* Thereafter, if (1>*fw*>0) and *r*<0 pace equal $pace^*$ −1. Also, in otherwise condition Equation (3) showed how can calculate the *pace*.

$$pace = \begin{cases} X^*_{i,t}r & if\ fw = 1 \\ distance^*_{best\ bee}r & if\ fw = 0 \\ pace^* - 1 & if\ fw > 0\ and\ fw < 1\ and\ r < 0 \\ distance^*_{best\ bee}fw & if\ fw > 0\ and\ fw < 1\ and\ r \geq 0 \end{cases} \tag{3}$$

where $\text{distance}_{\text{best bee}}$ denotes the variation in the current agent from the best agent. Therefore, it can be calculated by

Algorithm 1 Pseudo Code of FDO

Initialize scout bee population $X_{i,t}$ $(i = 1, 2, \ldots, n)$
  while *iteration* (t) *limit not reached*
    setting wf value by 0
    for *each artificial scout bee* $X_{i,t}$
      find best artificial scout bee $X^*_{i,t}$
      generate random-walk r in [-1, 1] range
      if($X_{i,t\ fitness}$ *== 0) (avoid divide by zero)*
        fitness weight = 0
          else
            calculate fitness weight. Equation (2)
      end if
      calculate distance $_{best\ bee}$ *used in Equation (4)*
      if *(fitness weight = 1)*
        calculate pace using Equation (3)
        else if *(fitness weight = 0)*
          calculate pace using Equation (3)
        else *calculate pace using Equation (3)*
            if *(random number < 0)*
              calculate pace using Equation (3)
            end if
        end if
      end if
      calculate $X_{t+1,i}$ *Equation (1)*
      if($X_{t+1,i}$ *fitness <* $X_{t,i}$ *fitness)*
        move accepted and pace saved
      else
        calculate $X_{t+1,i}$ *Equation (1) with previous pace*
          if ($X_{t+1,i}$ *fitness <* $X_{t,i}$ *fitness)*
            move accepted and pace saved
          else
            maintain current position (don't move)
          end if
      end if
    end for
  end while

Equation (4):

$$distance_{best\ bee} = X^* - X_{i,t} \tag{4}$$

Algorithm (1) and Figure (2) presents the detail of FDO algorithm.

A. LIMITATIONS OF FDO

Metaheuristic algorithms have both efficiency and limitation for obtaining optimal solution and having better convergence speed. Therefore, FDO has limitations because of presence multiple randomized parameter such as *fw*, *wf*, *pace,* updating Equation, best updating solution, and *Levy* flight. Consequently, according to paper [38] FDO has issue in terms of convergence so it has low convergence because of setting *wf* to zero. Having improper balance between exploration and exploitation is another problem related to FDO since FDO uses fitness of best agent, current fitness, and *wf*. Using *pace* based on randomization is also making unbalanced between

TABLE 1. Limitations of FDO against five algorithms.

	FDO	CJADE	IPOP-CMA-ES	PSO	SHADE	GA
F	**Avg.**	**Avg.**	**Avg.**	**Avg.**	**Avg.**	**Avg.**
TF1	7.47E-21	**-1400**	1.00E-08	4.2E-18	0	748.5972
TF2	9.39E-06	3045.822	**1.00E-08**	0.003154	9000	5.971358
TF3	8.55E-07	2944296	**1.00E-08**	0.001891	40.2	1949.003
TF4	6.69E-04	3379.984	**1.00E-08**	0.001748	0.000192	21.16304
TF5	23.501	**-1000**	1.00E-08	63.45331	0	133307.1
TF6	1.42E-18	**-895.466**	1.00E-08	4.36E-17	0.596	563.8889
TF7	0.544401	**-779.176**	7.01E-02	**0.005973**	4.6	0.166872
TF8	-2285207	-679.064	20.9	**-7.10E+11**	20.7	-3407.25
TF9	14.56544	**-571.738**	4.34	10.44724	27.5	25.51886
TF10	4.00E-15	**-499.953**	1.00E-08	0.280137	0.0769	9.498785
TF11	0.568776	**-400**	2.25	0.083463	0	7.719959
TF12	19.83835	**-274.825**	1.72	8.57E-11	23	1858.502
TF13	10.2783	**-99.8952**	2.16	0.002197	50.3	68047.23
TF14	1.05E+02	3192.957	708	150	**0.0318**	130.0991
TF15	1.73E+02	201.3478	259	188.1951	3220	**116.0554**
TF16	2.34E+02	330.4337	**3.75E-01**	263.0948	0.913	383.9184
TF17	4.31E+02	464.117	**34.3**	466.5429	30.4	503.0485
TF18	222.9682	501.1961	**40.01**	136.1759	72.5	118.438
TF19	22.7801	610.7729	**2.24**	741.6341	1.36	544.1018

both phases. FDO has poor performance due to refining its solution. This problem derived from using fixed value for *wf* and achieving low results by using *pace* Equations [30]. Results conducted from literature shows that FDO has poor performance against different algorithms such as PSO, GA, SHADE, CJADE, IPOP-CMA-ES [38]. Table (1) proves that FDO has low exploitation capability in unimodal test functions. While FDO shows its weakness in exploration in solving test function 8. Regarding the balancing issues between exploration and exploitation phase, FDO also has limitations against GA and SHADE as it can be seen in Table (1).

III. MODIFICATIONS

Different modifications of FDO are presented in this section. Based on our knowledge FDO has three modifications which are Chaotic FDO (CFDO) [33], Improved FDO, and Adaptive FDO (AFDO).

A. IFDO

In optimization algorithms, randomizations have great impact on exploration and exploitation. Thus, there are various techniques can be used to generate random numbers. Therefore, FDO have several areas based on randomization such as *Levy flight* mechanism, *fw* and *wf*. FDO original use 0 for *wf*. However, paper [38] used a range value for *wf* which is [0, 1]. Also, the same researchers used alignment and cohesion method in order to find new position. Equation (5) shows the technic that was added to Equation (1) in original FDO.

$$X_{i,t+1} = X_{i,t} + pace + (alignment^{*}\frac{1}{cohesion}) \tag{5}$$

where alignment is the Scouts' *pace* that is matched to the other scouts in the neighborhood. Cohesion is a scout's behavior that causes agents to steer towards the center of mass of the neighborhood. Thus, FDO algorithm uses the *wf* to control *fw* value that is zero in the original FDO. Paper [30] mentioned that *fw* is either 0 or 1. It can be seen from Equation (7); this mechanism in the FDO tends the result to slow convergence. Therefore, IFDO improved the *fw* through calculating a weight factor value in the range [0, 1] by using random mechanism. Also, *fw* is calculated in Equation (6) if one of these conditions are true $X_{i,t\ fitness} = 0$, $fw = 0$, thus, $fw > wf$:

$$fw = fw - wf \tag{6}$$

Otherwise, *fw* is calculated based on Equation (7):

$$\text{fw} = \left[\frac{X^{*}_{i,t\ fitness}}{X_{i,t\ fitness}}\right] \tag{7}$$

As results, IFDO is better than FDO in exploring solutions in the search space. Consequently, IFDO evaluated using two different test functions (classical and IEEE CEC2019 benchmark functions). Furthermore, it is compared against several algorithms, namely, FDO, PSO, WOA SSA, DA, and GA. Therefore, IFDO is better than FDO in terms of selecting scout bee position, avoiding local optima, and it has faster convergence against FDO [38].

B. CHAOTIC FDO

CFDO used chaotic map that is a nonlinear mechanism [39].Therefore, chaotic map have dynamic and nonlinear behavior that is the reason to be used by many researchers with in the optimization algorithms [40]. Thus, 10 chaotic maps applied inside FDO in two various ways. First, it was used to initialize the scout bee population, but FDO use stochastic mechanism for this process. Second, instead of using *Levy flight* mechanism, a chaotic map was used to generate the random variable r. Furthermore, this mechanism was used to create better population inside boundary of the search space. Singer map had significant results compared to other 9 chaotic maps so singer map was used to initialize population and generate random number in order to improve FDO. As a result, CFDO performed well compared to original FDO with regard to the speed of convergence and avoiding local optima [33].

C. HYBRIDIZATION

Hybridizing the powerful properties of various algorithms is the one of the most popular mechanism that is used to improve the performance of metaheuristic algorithms [41]. Since FDO proposed, one hybridization was proposed by researchers. The following section presents the hybridization of FDO:

1) HYBRID SINE COSINE AND FITNESS DEPENDENT OPTIMIZER (SC-FDO)

Paper [31] presented hybridization FDO with Sine Cosine Algorithm (SCA). Thus, FDO is slow in terms of convergence speed and it does not have proper balance between exploitation and exploration. SCA was embedded in FDO to refine the optimal solution. SCA is an optimization algorithm that was used for solving optimization problems [42]. In addition, *pace* equation is modified in SC-FDO algorithm according to following Equation(8), as shown at the bottom of the next page, where, r is a *Levy* random number. Also, *r1, r2* and *r3* are random variables, $X^*_{i,t}$ represents the global best solution, $X_{i,t}$ is the current solution of the scout bee. And, *fw* is the fitness weight in the range of [0, 1]. Equation (8) represents all the conditions as follow: If $X^*_{i,t} = X_{i,t}$ the *pace* is calculated. Therefore, if *fw* is equal to 1, the *pace* can be calculated. Thus, if $fw > 0$, $fw < 1$, and r is less than zero, the *pace* is calculated. Subsequently, if $fw > 0$, $fw<1$, and r is greater than or equal zero, *pace* is found. By modifying *pace* equation mechanism, scout bees can obtain proper balance between exploration and exploitation phase.

In addition, *r1, r2* and *r3* variables transform searching mechanism to exploitation. Thus, *r1* is calculated according to Equation (9). Where t is represents the current iteration, t_{max} denotes the maximum iteration and it is a constant variable. Moreover, *r2* shows the direction of the movement, and *r3* parameter is the random weight. Furthermore, the scout bee's movement is defined by Equation (10)

$$r_1(t) = a^* \left(1 - \frac{t}{t_{max}}\right) \tag{9}$$

$$distance_{best\ bee} = r_3^* X^*_{i,t} - X_{i,t} \tag{10}$$

The proposed algorithm was used to improve FDO in terms of having slow convergence and trapping into local optima. So, Equation (11) describes *fw* calculation:

$$fw = \left[\frac{X^*_{i,t\ fitness}}{X_{i,t\ fitness}}\right] \tag{11}$$

After adding the three various conditions to *fw*. Then, this parameter is calculated based on the following conditions:

$$nfw_t = fw_t - wf_t \qquad \text{if } fw_t > wf_t \tag{12}$$

$$nfw_t = fw_t \qquad \text{if } fw_t \le wf_t \tag{13}$$

In Equation (12) and (13), fw_t represents the current fitness weight and new *fw* at the t_{th} iteration represented by nfw_t. wf_t represents the current weight factor in the range [0, 1]. During increasing iterations, *wf* value decreased from *wf* to zero. Therefore, to tune random *fw*, another type of *fw* parameter was proposed. This type of *fw* was used for the best solution that was found so far by any scout bee over all the iterations. Global fitness weight denoted by fw^* used to refine *wf*.

Results proved that the exploitation capability of the FDO was improved by merging SCA features. SC-FDO was obtained the better neighboring search and achieved promising solutions by having better exploration [31].

IV. APPLICATIONS OF FDO

Based on the literature, FDO has been used to solve several real-world problems. In this section, the conducted applications present that FDO and its modifications were adapted to solve them.

A. APERIODIC ANTENNA ARRAY DESIGN

This application is used in wireless communication systems and developed in1960s. Generally, it can be classified into two types for initial array antennas design: thin antenna arrays and non-uniform antenna arrays. The optimal array pattern design is achieved by increasing channel capacity of wireless communication systems [43]. Antenna arrays are utilized in a variety of applications such as radios [44], radar [45], GPS [46]. Additionally, Designers of aperiodic arrays have to rely on different of techniques, such as randomly putting components using function of probability density. Therefore, real-number vectors represent a position in non-uniform arrays. To avoid discordant lobes most of antenna array use the boundary element technic [47].

Therefore, this problem can be improved in terms of a vector of real numbers by optimized element position to get the peak side lobe level (SLL) in non-uniform arrays.

Furthermore, to avoid grating lobes, the space of particular element should be limited for traditional periodic arrays.

The 10 elements of a non-uniform isotropic array are used to design the problem. On each side, just four element positions require to be optimized. Thus, the four-dimensional of this optimization problem shows in Equation (14).

$$X_i \in (0, 2.25) \left|X_i - X_j\right| > 0.25\lambda_0 > \min\{X_i\} > 0.125\lambda_0. \quad i = 1, 2, 3, 4. i \neq j \tag{14}$$

Nonetheless, elements should be greater than 0.125λ0 and smaller than 2.0λ0. Thus, lambda (λ) is the Greek alphabet indicates the wavelength [48]..Due to of these causes, every element boundary can be larger than 0 and smaller than 2.25.

FDO was used to optimize this problem over 200 iterations. So, twenty artificial scout search agents were utilized. FDO achieved optimal solution in iteration 78 and obtained these results: {0.713, 1.595, 0.433, 0.130}.

B. FREQUENCY MODULATED (FM) SOUND WAVES

FM is a modulation in which the encoding of information in a carrier wave is altered in the instantaneous frequency for the signal [49]. FDO was applied on FM sound waves to improve the FM parameters which have important role in a variety of modern music systems. Thus, this problem has six vectors can be optimized: $X = \{a1, w1, a2, w2, a3, w3\}$. The objective of this problem is to generate a sound that shows in Equation (15). The target sound represented in Equation (16) which is very likely to Equation [50].FDO can achieve the global best value converges result with 30 scout bees and for 200 iterations from iteration 64.

$$y(t) = a_1.sin(w_1.t. + a_2.sin(w_2.t.\theta + a_3.sin(w_3.t.\theta))) \tag{15}$$

$$y(t) = 1.0^{*}sin((5.0).t + 1.5^{*}sin(4.8.t.\theta + 2^{*}sin(4.9.t.\theta))) \tag{16}$$

where $\theta\pi = 2/100$, and the parameter t should be in the range [−6.4, 6.35]. The aim fitness function minimizes the summation of square root between both the target wave with the minimum value while t equals100 turns. The estimated wave is calculated according to Equation (17) [30].

$$f(\vec{X}) = \sum_{t=0}^{100} (y(t) - y_0(t))^2 \tag{17}$$

C. ONE-DIMENSIONAL BIN PACKING

One-dimensional bin packing problem can be defined as a packing set of items with various specifications into identical bins [51]. AFDO was used to obtain better solution for this problem. Paper [32] proposed AFDO used first fit (FF) heuristic mechanism embedded inside FDO. FF it is a greedy algorithm that tries to set every new item into the first bin in which it fits [52]. Thus, through FF technic the scout bee population was initialized. AFDO algorithm achieved the final optimal solution compared to the original FDO. The experimental results of AFDO reached the minimum number of used bins and achieved a better item packing of bins. It also obtained better efficiency in gaining solutions for instances with the smallest minimum. Average fitness values from three standard datasets compared with the PSO, Jaya, and crow search(CS) algorithms, results presented that AFDO achieved higher performance [32].

D. PROPORTIONAL INTEGRAL DERIVATIVE (PID) CONTROLLER

PID controller known as Integral Proportional Derivative (I-PD) and it is the most common controllers that were utilized in industry [53]. FDO technique was applied for various constraints including Governor Dead Band (GDB), Generation Rate Constraint (GRC), Time Delay (TD), and Boiler Dynamics (BD) in PID/I-PD controllers. Optimizing this problem among thirty runs obtained significant improvement in respect of overshoot, undershoot, and settling time while it was compared to other techniques such as Teaching Learning Based Optimization (TLBO), PSO and FA [54].

E. PEDESTRIAN EVACUATION MODELING

Evacuation of pedestrian's model used for decreasing negative aspects in case of emergency like, fire, deaths and earthquake. Primarily, evacuation models can be classified into two types: continuous model and discrete model [55]. FDO was applied on a cellular automata model which is a discrete type model. Additionally, the pedestrian's model required speed and stretch from the exit door. Also, these two parameters used to obtain the evacuation time as it can be seen in Equation (18). Where pedestrian's stretch from the exit door as denoted by dist, and the pedestrian's speed denoted by desired speed.

$$evecTime = \left(\frac{dist}{2}\right) * desiredSpeed \tag{18}$$

$$Pace = \begin{cases} X_{i,t}{}^{*}r & if\ fw = 1 \\ X_{i,t} + r_1^{*}\cos(r_2)^{*}\left(r_3^{*}X_{i,t}^{*} - X_{i,t}\right)^{*}r & if\ fw = 0 \\ \left(X_{i,t} + r_1{}^{*}sin(r_2)(r_3^{*}X_{i,t}^{*} - X_{i,t}\right)^{*}fw)^{*} - 1 & if\ fw > 0\ and\ fw < 1\ and\ r < 0 \\ \left(X_{i,t} + r_1{}^{*}sin(r_2)(r_3^{*}X_{i,t}^{*} - X_{i,t}\right)^{*}fw) & if\ fw > 0\ and\ fw < 1\ and\ r \geq 0 \end{cases} \tag{8}$$

The pedestrian's stretch from the exit door dist can be calculated as follow Equation:

$$DIST = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \tag{19}$$

where the coordinate of pedestrian's location denoted by X_1 and Y_1, and coordinate of the locations of exit door is represented by X_2 and Y_2.

Finally, FDO reached the global best solution through searching the best location of the exit door during the evacuation operation, and it obtined optimum solution with 57 iterations. Additionally, IFDO was appllied to the same problm and achieved better result against FDO since IFDO gained optimum solution with 38 iterations [38].

F. PRESSURE VESSEL DESIGN (PVD) PROBLEM

PVD is a class in classical engineering problem. It is used for designing tank to hold gases or liquids. Thus, PVD depends on various parameter such as size, contents, pressure, mass and materials [56]. Therefore, PVD have four variables can be optimized such as: shell thickness T_s, inner radius R, thickness of the head T_h, and barring head L. Therefore, constraints of the problem can be calculated according to following Equations:

$$\begin{aligned} n &= 1, 2, 3, 4 \\ \vec{X} &= [X_1 X_2 X_3 X_4] = [T_s T_h R L] \\ f(\vec{X}) &= 0.6224 X_2 X_3 X_4 + 1.7781 X_2 X_3^2 \\ &\quad + 3.1661 X_1^2 X_2 + 19.84 X_1^2 X_3 \end{aligned} \tag{20}$$

Variable range $0 \leq X_1, X_2 \leq 99$ and $10 \leq X_3, X_4 \leq 200$

Therefore, the main aim of optimizing this problem is to minimize the total cost. Thus, optimizing PVD problem has to be done based on the following constraints which are represented in Equation (21), (22), (23), and (24).

$$g_1(\vec{X}) = X_1 + 0.0193 X_3 \leq 0 \tag{21}$$
$$g_2(\vec{X}) = -X_3 + 0.00954 X_3 \leq 0 \tag{22}$$
$$g_3(\vec{X}) = -\pi X_3^2 X_4 - \frac{4}{3}\pi X_3^3 + 1.296.000 \leq 0 \tag{23}$$
$$g_4(\vec{X}) = X_4 + 240 \leq 0 \tag{24}$$

Therefore, FDO applied to solve PVD and obtained best solution compared to GA, WOA and PSO. However, CFDO was also used to solve PVD so it acquired higher performance against FDO while CFDO achieved these results: $T_s = 1.54$, $T_h = 6.10$, R 30.58, and $L = 73.29$ [33].

G. TASK ASSIGNMENT PROBLEM

Task assignment is one of the main steps to ensure the capabilities of efficient exploitation of the distributed or parallel computing systems. It is a nondeterministic polynomial (NP) problem. It is working on assigning tasks among various processors of a distributed computer system [57]. The objective of optimizing task assignment is to reduce the execution cost for every task in order to gain high throughput. Equation (25) presents the objective function of the problem:

$$f(x) = \sum_{j=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij} \tag{25}$$

where C_{ij} denotes the cost of every task, X_{ij} refer to the task whether the task is allocated to a specific processor or not. Consequently, if the task is designated to specific processor $X_{ij} = 1$, Otherwise $X_{ij} = 0$.

Thus, FDO applied to solve task assignment problem and it outperformed well. FDO achieved best performance against other algorithms. CFDO also used to solve this problem and it obtained better achievement. Thus, researchers used task assignment regarding to a department which have five employees with five tasks. The execution time for whole the tasks limited between 10 and 99 [33].

Algorithm 2 Pseudo Code of MFDO

```
Initialize scout bee population X_{i,t} (i = 1, 2, ..., n)
  while iteration (t) limit not reached
    setting wf value by [0,0.2]range
    for each artificial scout bee X_{i,t}
      find best artificial scout bee X*_{i,t}
      generate random-walk r in [-1, 1] range
      if(X_{i,t fitness} == 0) (avoid divide by zero)
        fitness weight = 0
          else
            calculate fitness weight. Equation (27)
      end if
      calculate distance_{best bee} used Equation (4)
      if (fitness weight = 0)
        calculate pace using Equation (28)
        else if (fitness weight = 1)
          calculate pace using Equation (28)
        else calculate pace using Equation (3)
          if (random number < 0)
            calculate pace using Equation (3)
          end if
        end if
      end if
      calculate X_{t+1,i} Equation (1)
      if(X_{t+1,i} fitness < X_{t,i} fitness)
        move accepted and pace saved
      else
        calculate X_{t+1,i} Equation (1) with previous pace
          if (X_{t+1,i} fitness < X_{t,i} fitness)
            move accepted and pace saved
          else
            maintain current position (don't move)
          end if
      end if
    end for
  end while
```

V. PROPOSED MFDO

Despite of having better performance against several common algorithms, FDO still requires further improvement due to having low convergence, not refining solutions, and not avoiding local optima.

Considering, the shortcomings of the basic FDO, a Modified FDO (MFDO) is proposed in this paper to enhance the performance of original FDO. In this paper, FDO was modified based on two great methods. These two methods improve the performance of FDO to solve its limitations. The following section describes both methods.

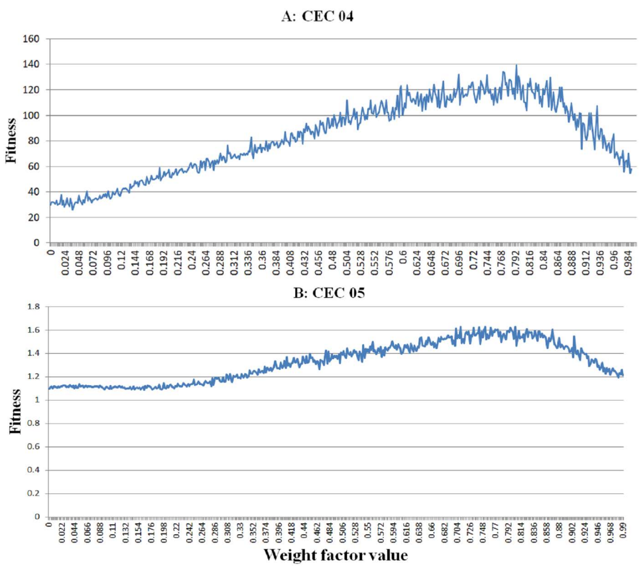

FIGURE 3. Affecting ***wf*** on benchmark functions (A) for CEC04 and (B) for CEC05.

A. OPTIMAL WF RANGE

Original FDO utilizes *pace* as the factor of movement and the agent trend. Thus, *fw* used for controlling *pace*. One way to refine the *pace* value, *fw* should be improved while this improvement cannot be done without enhancing global fitness, current fitness, and *wf*. *wf* has been focused on in this paper in order to use optimal value for it. Reference to [30] if *wf* is equal to one, FDO has better convergence and the exploration is weak. However, it has low convergence, if the *wf* is equal to zero because the FDO tries to do more exploration. Therefore, the range value of *wf* is [0, 1]. Paper [38] used this range to choose *wf* in order improve FDO.

Despite of using value of *wf* in its range, *wf* still requires further improvement. Therefore, experimental test has been done in order to shrink the range between 0 and 1. So, in this experiment, 496 numbers were chosen between 0 and 1. Then, MFDO was tested using each number over 500 iterations. Thirty agents were used in the population and results conducted using 30 runs in order to obtain average value. Figure (3) illustrates that how different numbers affect the performance of MFDO for different benchmark functions. Based on the experiment, it can be said that the best range to achieve optimal results by MFDO is [0, 0.2]. CEC2019 benchmark functions were used in this experiment and only 5 functions are presented as follows:

Overall, this experiment has shown that MFDO has no effect on the CEC06, CEC07 in CEC2019 benchmark functions because results fluctuated illogically. Therefore, it can be said that *wf* does not have impact to solve both of the functions. Figure (4) illustrates the experimental results of *wf*.

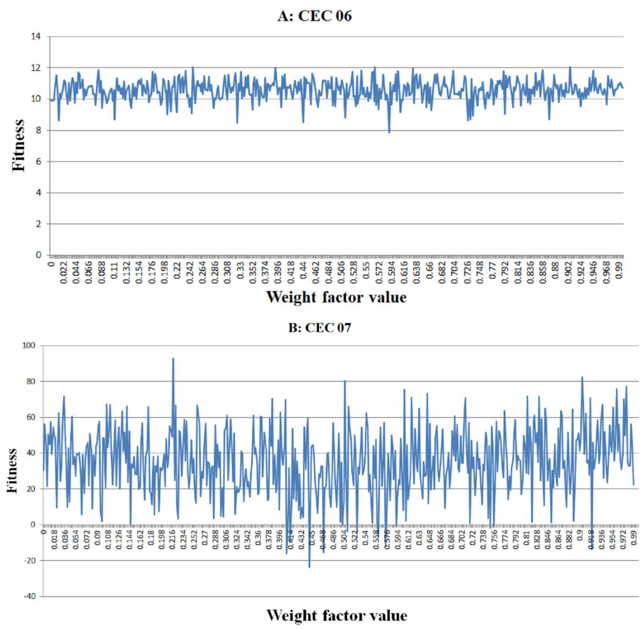

FIGURE 4. Affecting ***wf*** on benchmark functions (A) for CEC06 and (B) for CEC07.

B. ADDING SINC FUNCTION PARAMETER STRATEGY

The *sinc* function is a mathematics function which is the abbreviation "sine cardinal", denoted by *sinc(x)*. There are two conditions on the real line can be represented as Equation (26) [58].

$$sinc(x) = \begin{cases} 1 & if\ x = 0 \\ \frac{sin(x)}{x} & otherwise \end{cases} \tag{26}$$

Consequently, previous section presented importance of *wf* and its effects on the *fw*, and agent movement. Therefore, *sinc* method is used to improve *fw* parameter from Equation (2) in basic FDO and the new equation implemented which can be seen in Equation (27).

$$fw = \left[\frac{X^*_{i,t\ fitness}}{X_{i,t\ fitness}}\right] {}^* sinc(\pi^* wf) \tag{27}$$

where $X^*_{i,t\ fitness}$ represents the fitness function of the global best solution, $X_{i,t\ fitness}$ is the current best solutions of the scout bee, and *wf* indicates the weight factor and it has the rang of [0, 0.2].

TABLE 2. Comparing MFDO against FDO, IFDO, SC-FDO and CFDO using classical benchmark functions.

	MFDO		FDO		IFDO		SC-FDO		CFDO	
F	Avg.	STD	Avg.	STD	Avg.	STD	Avg.	STD	Avg.	STD
TF1	2.62E-59	1.41E-58	2.32E-82	8.62E-82	5.38E-24	2.74E-23	0.00E+00	0.00E+00	6.44E-51	3.46543E-50
TF2	2.52E-28	9.98E-28	1.33E-29	2.71E-29	0.534345844	1.620259633	0.00E+00	0.00E+00	4.02E-34	2.02891E-33
TF3	1.29E-13	2.40E-13	6.76E-15	8.70E-15	2.88E-07	6.90E-07	0.00E+00	0.00E+00	1.81E-06	5.60366E-06
TF4	3.61E-13	7.54E-13	1.95635E-13	1.31E-13	2.60E-04	9.11E-04	0.00E+00	0.00E+00	7.95E-06	4.27333E-05
TF5	1.06E+00	1.57E+00	1.24E+00	1.49E+00	1.94E+01	3.31E+01	2.69E-01	2.69E-01	1.68E+01	31.4405605
TF6	1.92E-32	2.23E-32	8.73E-33	9.28E-33	4.22E+06	8.15E-09	2.02E-02	5.53E+02	2.01E-01	0.502499011
TF7	5.09E-01	2.95E-01	6.81E-01	3.22E-01	5.68E-01	3.14E-01	4.49E-02	5.98E+02	5.75E-01	3.54E-01
TF8	-3.76E+03	4.18E+02	-3.61E+03	2.10E+02	-2.92E+06	2.24E+05	-1.02E+04	2.90E+04	-1.04E+04	1.29E+04
TF9	1.95E+00	9.91E-01	2.52E+00	1.18E+00	1.35E+01	6.66E+00	0.00E+00	0.00E+00	5.64E-01	3.04E+00
TF10	5.15E-15	1.42E-15	6.10E-15	1.77E-15	5.18E-15	1.67E-15	3.26E-15	3.66E-15	1.48E-15	1.32E-15
TF11	6.04E-02	3.45E-02	7.18E-02	4.55E-02	0.525690405	8.90E-02	0.00E+00	0.00E+00	9.26E-02	1.25E-01
TF12	**7.05E-08**	3.62E-07	2.07E-02	7.76E-02	1.81E+01	2.57E+01	4.54E-02	5.61E-02	1.19E+00	7.97E-01
TF13	3.66E-04	1.97E-03	3.66E-04	1.97E-03	4.10E+09	1.50E-05	1.73E-01	1.90E-01	6.79E-01	2.87E-01
TF14	1.63E+00	7.46E-01	1.76E+00	1.01E+00	2.68E-07	4.68E-07	6.36E+01	9.01E+01	5.50E+00	3.92E+00
TF15	3.07E-04	4.87E-19	3.19E-04	5.94E-05	4.03E-16	9.25E-16	2.28E+02	2.37E+02	3.10E-03	1.12E-02
TF16	-1.03E+00	0.00E+00	-1.03E+00	0.00E+00	9.14E-16	3.61E-16	3.55E+02	3.61E+02	-1.03E+00	2.80E-03
TF17	-1.01E+00	3.07E+01	23.82013	2.15E-01	2.38E+01	1.24E-01	5.17E+02	5.25E+02	-1.03E+00	3.01E+01
TF18	**3.00E+00**	4.44E-16	3.00E+00	4.58E-11	2.24E+02	2.68E-05	1.56E+02	1.68E+02	1.93E+01	3.51E-03
TF19	-3.86E+00	2.66E-15	-3.86E+00	2.66E-15	3.15E+01	1.32E-03	7.09+02	7.31E+02	-2.64E+00	5.17E-01

Moreover, to enhance the FDO from the perspective of convergence speed, the *pace* updating mechanism of the MFDO, guides the search agents to obtained proper balance between exploration and exploitation, improved *pace* updating equation is calculated based on the following:

$$\text{pace} = \begin{cases} \mathrm{X_{i,t}}^{*}\mathrm{r}^{*}\mathrm{sinc}(\pi * \mathrm{wf}) & \text{if fw} = 0 \\ \text{distance}_{\text{best bee}}{}^{*}\mathrm{r}^{*}\mathrm{sinc}(\pi^{*}\mathrm{wf}) & \text{if fw} = 1 \end{cases} \tag{28}$$

where $X_{i,t}$ is the current solution, r denotes a random number in the [1, −1] range, the weight factor of the agent represented by $wf \in [0, 0.2]$. Also, distance best bee is the result of subtracting the value of $X^{*}_{i,t}$ from $X_{i,t}$ and π is a mathematical constant. If fw=0, the *pace* calculated according to case one in Equation (28). However, if the current solution is equal to global best solution, *pace* is calculated by multiply distance best bee, r, and $sinc(\pi$ (approximately equal to 3.14159)$*wf)$.

C. EXPERIMENTAL RESULTS

In order to verify the performance of MFDO, two different types of benchmark functions such as CEC2005 and CEC2019 are used. Thus, the average (Ave), probability value (P-value) and standard deviation (Std) of the obtained results are used as evaluation of the performance testing. Therefore, all algorithms are run in MATLAB programming tool of personal computer, an Intel (R) Core (TM) i7-7600 CPU, 2.80 GHz processor with 8 GB RAM under windows 10 operating system, and also 500 iterations used for each of the run of the algorithms.

1) CLASSICAL BENCHMARK TEST FUNCTIONS

Classical benchmark test functions are selected to test the performance of MFDO. They have various characteristics, such as, unimodal, multimodal and composite test functions. Every set of these test functions is utilized to evaluate different aspects of the algorithm. Thus, unimodal functions have a single optimum that is used for exploitation, and convergence speed of the algorithm. Nevertheless, multimodal is used for testing the local optima, exploration levels in which they have multi optimal solutions. Moreover, the composite benchmark functions are generally combined of versions of other test functions [30]. Table (2) shows classical benchmark functions.

Table (2) presents comparison of five algorithms that were run 30 times by using 30 search agents with ten dimensions. In each test, calculating the average and standard deviation calculated by using 500 iterations. Also, it shows the comparison results of MFDO against IFDO, FDO, SCFDO and CFDO. Therefore, TF12 and TF18 results showed that MFDO generally provided best results than the other algorithms. MFDO also achieves better results for solving TF5, TF7, and TF11 against FDO, IFDO and CFDO. Thus, MFDO achieved better results in solving TF6, TF13, TF16 and TF19 against the compared algorithms. Furthermore, MFDO in all TF at least have one better result than other presented algorithm. The results show that the MFDO in eleven TF best than original FDO, Also MFDO have sixteen preferable results compared with IFDO and MFDO surpasses in

TABLE 3. Comparing MFDO With GWO, ChOA, GA, And BOA using CEC2019 benchmark functions.

	MFDO		GWO		ChOA		GA		BOA	
F	**Avg.**	**STD**	**Avg.**	**STD**	**Avg.**	**STD**	**Avg.**	**STD**	**Avg.**	**STD**
CEC01	4.92E+07	4.68E+07	2.13 E+08	3.07 E+08	4.24E+09	9.67E+09	5.32 E+04	7.04 E+04	5.89E+04	1.14E+04
CEC02	**1.73E+01**	0.00E+00	1.83E+01	3.04E-04	1.84E+01	1.86E-02	1.74E+01	1.73E+01	1.89E+01	2.91E-01
CEC03	1.27E+01	8.88E-15	1.37E+01	7.23 E-15	1.37E+01	7.11E-06	1.27E+01	1.37E+01	1.37E+01	6.17E-04
CEC04	**2.82E+01**	1.52E+01	3.01E+02	6.87E+02	5.93E+03	2.86E+03	6.23 E+04	6.20E+04	2.09E+04	7.71E+03
CEC05	**1.09E+00**	4.92E-02	2.43E+00	2.52E-01	4.21E+00	8.87E-01	7.54E+00	7.28E+00	6.18E+00	7.08E-01
CEC06	9.28E+00	6.16E-01	1.19E+01	7.31E-01	1.22E+01	6.83E-01	7.40E+00	6.69E+00	1.18E+01	7.71E-01
CEC07	**6.00E+01**	8.95E+01	5.35E+02	2.92E+02	1.01E+03	1.79E+02	7.92E+02	6.98E+02	1.04E+03	2.15E+02
CEC08	**4.13E+00**	4.68E-01	5.40E+00	9.94E-01	6.78E+00	1.56E-01	6.10E+00	5.82E+00	6.34E+00	3.59E-01
CEC09	**2.39E+00**	3.09E-02	1.47E+01	5.00E+01	4.49E+02	2.45E+02	5.31 E+03	5.29 E+03	2.27E+03	8.11E+02
CEC10	**1.63E+01**	6.66E+00	2.15E+01	6.85E-02	2.15E+01	7.20E-02	2.01E+01	2.00E+01	2.15E+01	7.95E-02

TABLE 4. Comparing MFDO with modifications Of FDO Using CEC2019 benchmark functions.

	MFDO		FDO		IFDO		SC-FDO		CFDO	
F	**Avg.**	**STD**	**Avg.**	**STD**	**Avg.**	**STD**	**Avg.**	**STD**	**Avg.**	**STD**
CEC01	4.92E+07	4.68E+07	6.59E+07	7.13E+07	**2.65E+03**	1.39E+04	4.25E+04	4.26E+04	8.93E+08	2.02E+09
CEC02	**1.73E+01**	0.00E+00	1.73E+01	1.03E-09	4.00E+00	1.00E-05	1.73E+01	1.73E+01	1.75E+01	3.16E-01
CEC03	**1.27E+01**	8.88E-15	**1.27E+01**	1.07E-11	1.37E+01	4.82E-09	**1.27E+01**	1.27E+01	1.27E+01	4.13E-11
CEC04	**2.82E+01**	1.52E+01	2.84E+01	9.74E+00	3.12E+01	1.29E+01	1.59E+03	2.05E+03	5.20E+01	1.89E+01
CEC05	**1.10E+00**	4.92E-02	1.13E+00	4.65E-02	1.13E+00	7.06E-02	1.69E+00	1.71E+00	1.17E+00	1.57E-01
CEC06	9.28E+00	6.16E-01	**8.86E+00**	7.30E-01	1.21E+01	5.21E-01	8.27E+00	8.33E+00	1.10E+01	1.21E+00
CEC07	6.00E+01	8.95E+01	**5.02E+01**	8.57E+01	1.16E+02	1.03E+01	5.93E+01	8.99E+01	8.02E+02	2.17E+02
CEC08	**4.13E+00**	4.68E-01	4.45E+00	4.25E-01	4.94E+00	8.91E-01	4.50E+00	4.53E+00	4.95E+00	7.35E-01
CEC09	**2.38E+00**	3.09E-02	2.40E+00	3.73E-02	2.47E+01	3.10E-15	4.85E+00	4.92E+00	2.88E+00	4.93E-01
CEC10	**1.63E+01**	6.66E+00	1.87E+01	5.01E+00	2.07E+01	4.44E-16	1.81E+01	1.84E+01	2.00E+01	1.03E-02

fourteen TF than CFDO. However, the MFDO was worse than the SC-FDO in ten TF. Moreover, the results show that the MFDO in TF18 and TF12 better overall in comparison with the selected comparator algorithms.

2) CEC2019 BENCHMARK TEST FUNCTIONS

To measure the algorithm's performance, CEC2019 is used to evaluate MFDO. Thus, it consists of ten multimodal benchmark functions and they are enhanced test functions for optimization [59]. The proposed MFDO is compared with five common nature inspired algorithms, namely FDO [30], Grey Wolf Optimization (GWO) [60], Chimp Optimization Algorithm (ChOA) [61], GA [15], and Butterfly Optimization Algorithm (BOA) [62]. Each algorithm runs thirty times using 30 search agents and a maximum number of 500 iterations was used to find the optimum solution. Then, the average and standard deviation of the results were obtained and presented in this section.

Table (3) shows the proposed MFDO has the first rank as it outperformed well in seven test functions compared to the other algorithms in CEC02, CEC04, CEC05, CEC07, CEC08, CEC09 and CEC010. GWO, GA, and BOA have the second, third, and fourth ranks, respectively in average results. However, the ChOA recorded the least rank in the performance comparison.

The reason behind this achievement is that range of *wf* is [0, 0.2] and *sinc* function is used for finding *fw*. Also, this *sinc* function used to find *pace* for two equations from four conditions.

Moreover, when comparing MFDO with FDO [30], and its three different modifications such as IFDO [38], SCFDO [31], and CFDO [31] for CEC2019 benchmark test functions it can be seen that the average of MFDO is well competitive with other algorithms. Table (4) presents MFDO algorithm that produce the smallest average fitness for whole instances. It also shows comparison of MFDO with three different modifications of FDO and original FDO. The modifications are IFDO [38], SCFDO [31], and CFDO [31]. Results proved that MFDO achieved optimum results against the compared modifications. The reason behind these improvements shows that MFDO improvement has great impact and it is better than the previous modifications.

TABLE 5. The WILCOXON rank-sum test for CEC2019 benchmark functions.

F	MFDO vs. FDO (*P* value)
CEC01	0.62040
CEC02	**0.04177**
CEC03	1.45600
CEC04	0.42896
CEC05	**0.01441**
CEC06	0.01441
CEC07	0.01441
CEC08	**0.01695**
CEC09	**0.03514**
CEC10	**0.01837**

However, Wilcoxon rank-sum test was used to calculate p value on achieved results in order to show the statistical results. Wilcoxon rank-sum test also called Mann–Whitney U test which is a nonparametric test used to compare two samples from populations with the same distribution [63]. Thus, the p value is determined by using the Wilcoxon rank-sum test and it should be less than 0.05 in order to be significant. Table (5) presents the p value of MFDO algorithm against the original FDO because FDO algorithm was already tested against common algorithm such as PSO, GA, DA, WOA and SSA [30].

As it can be seen from Table (5), MFDO statistically achieved higher performance against FDO in 7 benchmark functions while it was not improved well in the other three benchmark functions. Overall, it can be said that MFDO statistically improved the capacity of FDO algorithm.

D. CONVERGENCE EVALUATION

In order to evaluate the convergence of MFDO, CEC2019 benchmark functions are used. MFDO and FDO has been run it separately and observed the best solution in each iteration. Both algorithms have run over 500 and 100 iterations using 30 agents. Using both of the iterations gave the same convergence rate. Therefore, Figure (5) illustrates the convergence rate between FDO and MFDO. As it can be seen that MFDO enhanced the convergence against FDO. Overall, based on the results that have shown in previous sections, MFDO improved the performance of FDO in terms of exploitation and improving the convergence speed. The reason behind the improvement of MFDO is that MFDO used two great methods: shrinking the range of *wf* and using *sinc* method to update *fw* which affects *pace* value.

E. MFDO REAL WORLD APPLICATION

The MFDO is used as a trainer to solve real-world application problems. In this section, the proposed algorithm is used

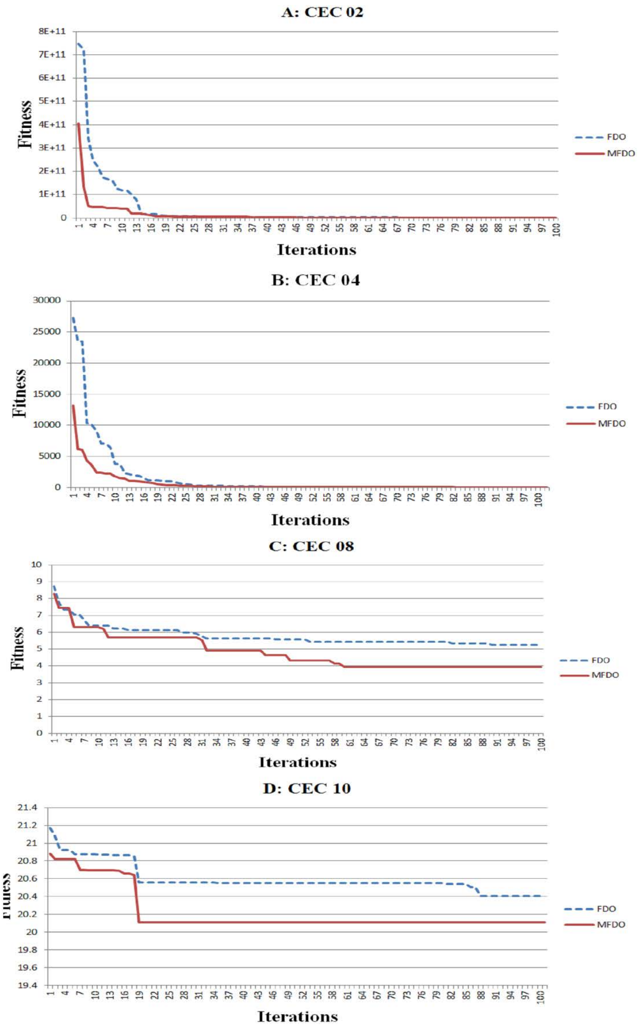

FIGURE 5. Convergence curves of the MFDO and FDO algorithms on four representative test functions; (A) for CEC02, (B) for CEC04, (C) for CEC08 and (D) for CEC10.

to solve three problems which are Welded Beam Design (WDB), PVD and Spring Design Problem.

1) WELDED BEAM DESIGN (WDB)

Welding is a method that is used for joining metal pieces using heat, pressure, or both, and with or without the additional material. It depends on various design parameters, including design for minimum cost $f(x)$ based on bar buckling load, weld stress, bending stress and end deflection. WDB considered a feasible set of dimensions $h(x_1)$, $l(x_2)$, $t(x_3)$ and $b(x_4)$. The WDB problem can be represented mathematically as follow:

$$\mathrm{Minf}(\vec{X}) = 1.10471 X_1^2 X_2 + 0.04811 X_3 X_4 \,(14.0 - X_2) \tag{29}$$

Constraints are:

$$g_1(\vec{X}) = \tau_{max} - \tau(x) \geq 0 \tag{30}$$

$$g_2(\vec{X}) = \sigma_{max} - \sigma(x) \geq 0 \tag{31}$$
$$g_3(\vec{X}) = X_4 - X_1 \geq 0 \tag{32}$$
$$g_4(\vec{X}) = 5 - 0.1047X_1^2X_2 + 0.04811X_3X_4(14.0 + X_2) \geq 0 \tag{33}$$
$$g_5(\vec{X}) = X_1 - 0.125 \geq 0 \tag{34}$$
$$g_6(\vec{X}) = \delta_{max} - \delta(x) \geq 0 \tag{35}$$
$$g_7(\vec{X}) = P_c(x) - P \geq 0 \tag{36}$$

where the values of loads and stresses are given as $P = 6000$ *lb*, $\tau_{max} = 13{,}600\ psi$, and $\sigma_{max} = 30{,}000\ psi$ and $\delta_{max} = 0.25$. Also, the variables limitations are given as: $0.125 > x_1 > 5$, $0.1 > x_2 > 10$, $0.1 > x_3 > 10$, $0.125 > x_4 > 5$.

Weld stress $\boldsymbol{\tau}(x)$ has two components: $\boldsymbol{\tau}'$ is the primary stress and $\boldsymbol{\tau}''$ is the secondary torsional stress. $\boldsymbol{\tau}(x)$ is found by Equation (37). $\boldsymbol{\sigma}(x)$ bar bending stress is computed using the Equation (38). Equation (39), (40) are used to calculated bar buckling load (Pc), $\delta(x)$ bar end deflection sequential [64], [65], [66], [67]. Figure (6) illustrates the parameters of welded beam design problem.

$$\boldsymbol{\tau}(X) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{X_2}{2R} + (\tau'')^2} \tag{37}$$
$$\boldsymbol{\sigma}(X) = \frac{6PL}{X_4X_3^2} \tag{38}$$
$$\boldsymbol{Pc}(X) = \frac{\sqrt{4.013E\frac{X_3^2X_4^6}{2R} + (\tau'')^2}}{L^2}\left(1 - \frac{X^3}{2L}\sqrt{\frac{E}{4G}}\right) \tag{39}$$
$$\boldsymbol{\delta}((X) = \frac{4PL^3}{X_3^3X_4} \tag{40}$$

Finally, Different algorithms have been used to solve this problem, such as Gravitational Search Algorithm (GSA), GOA, and GA. Thus, MFDO and FDO have been applied to solve this problem in order to obtain the global best solution by finding the least cost. Thirty agents are used over 100 iterations. Results show that the proposed algorithm was more efficient than FDO to achieve optimum solution. Table (6) shows that MFDO achieved higher performance against FDO, GSA, GA, GOA [68], [69].

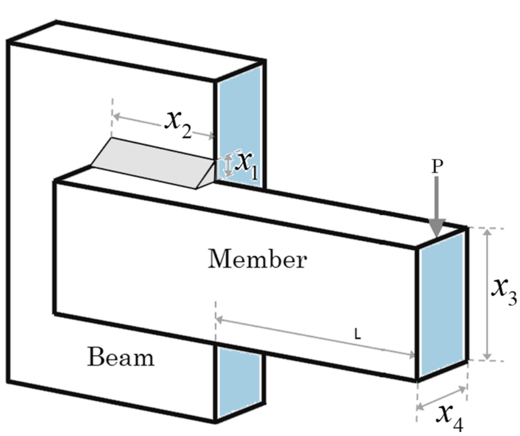

FIGURE 6. Welded beam design.

TABLE 6. Comparison of MFDO against common algorithms for solving welded beam design.

Algorithms	Mean
MFDO	**2.45E+00**
FDO	2.49E+00
GSA	3.58E+00
GA	3.65E+13
GOA	2.72E+00

TABLE 7. Comparison of MFDO with common algorithms for solving the pressure vessel design problem.

Algorithms	Mean
MFDO	**6.54E+03**
FDO	5.33E+ 04
GSA	8.93E+03
GA	2.18E+05
GOA	1.40E+05

2) PRESSURE VESSEL DESIGN (PVD)

PVD is a classical engineering problem. The aim of optimizing PVD function is to minimize the cost. Thus, it can be optimized through minimize Forming, material, and welding. PVD involves four decision variables to be optimized: X_1 denoted thickness of the pressure vessel *Ts*, X_2 is defined for thickness of the head Th, X_3 stands for inner radius of the vessel *R*, and X_4 is on behalf of length of the vessel without counting the head *L*, as presented in Figure (7) [56]. Section 4.6 described PVD in detail.

PVD had been solved by different algorithms so Table (7) shows results [69]. Therefore, FDO and MFDO are applied to optimize this problem. Thirty agents and 100 iterations have been used to solve this problem. The proposed algorithm outperforms well by obtaining result (6.54E+03) compared to other algorithms [69].

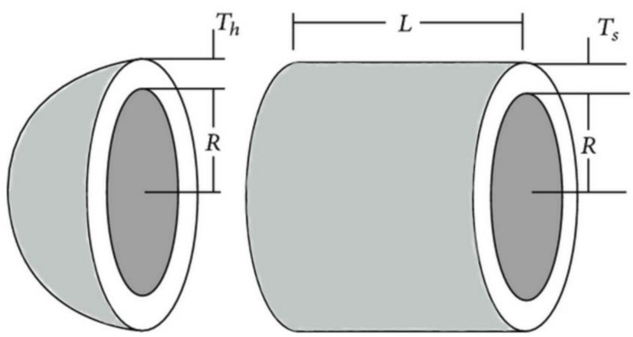

FIGURE 7. Pressure vessel design problem.

3) SPRING DESIGN PROBLEM

Spring design is the type of mechanical design problems that was presented in Siddall [72]. The aim of this engineering design problem is achieving a helical compression spring

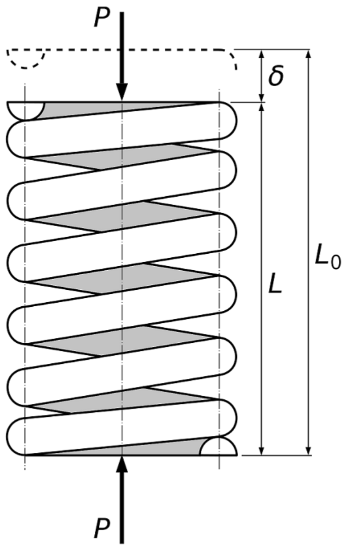

FIGURE 8. Spring design problem.

TABLE 8. Comparison of MFDO with common algorithms for solving spring design Problem.

Algorithms	Mean
MFDO	**1.38E-02**
FDO	1.39E-02
GSA	2.08E-02
GA	2.93E+14
GOA	1.65E−02

with the minimum volume. This problem has three variables that can be optimized as they are shown in Figure (8) : X_1 the number of spring coils(N), X_2 wire diameter(d), and X_3 mean coil diameter(D) [68]. Furthermore, N is integer, d is a discrete variable, and D is continuous. Thus, the bounds on the variables are: $0.5 \leq X_1 \leq 2.0$, $0.25 \leq X_2 \leq 1.30$, and $2.0 \leq X_3 \leq 15.0$ [70]. Spring design problem can be calculated mathematically as follows:

$$\text{Min} f(\vec{X}) = (X_3+2)X_2X_1^2 \tag{41}$$

Subject to:

$$g_1(\vec{X}) = 1 - \frac{X_2^3 X_3}{71785 X_1^4} \leq 0 \tag{42}$$

$$g_2(\vec{X}) = \frac{4X_2^2 - X_1X_2}{12566(X_2X_1^3 - X_1^4)} + \frac{1}{5108X_1^2} \leq 0 \tag{43}$$

$$g_3(\vec{X}) = 1 - \frac{140.45X_1}{X_2^2X_3} \leq 0 \tag{44}$$

$$g_4(\vec{X}) = 1 - \frac{X_1 + X_2}{1.5} - 1 \leq 0 \tag{45}$$

MFDO algorithm has been applied to solve this problem with 30 agents over 100 iterations. The results in table (8) show that MFDO superior to other algorithms for optimizing this problem.

VI. CONCLUSION AND FUTURE WORKS

In this study, a comprehensive survey of FDO and its applications presented. Then, based on the limitations of FDO, MFDO proposed in order to enhance the performance of FDO.

By experimental simulations on using *wf* between 0 and 1, MFDO used the range of *wf* between 0 and 0.2. Then, *wf* was used to find *fw*. Also, *sinc* method was used in three places inside MFDO to find *fw* and *pace*.

MFDO applied to solve classical benchmark functions and CEC2019 benchmark functions. MFDO achieved best results compared to original FDO. Overall, MFDO improved the performance of FDO in terms of exploitation and balancing between exploration and exploitation phases. It outperforms well against FDO, IFDO, SCFDO, CFDO, GA, GWO, BOA and ChOA. Furthermore, statistical results proved that MFDO achieved significant results compared to FDO.

As a result, MFDO performance is better than FDO in terms of avoiding local optima and refining the optimum results based on the improvements that were added to MFDO.

Therefore, MFDO also has the capability to address real world applications; three real-life problems were selected: welded beam design, pressure vessel design, and spring design problem. In all applications, the MFDO outperformed the FDO, GSA, GA, and GOA.

This study can be used by researchers in order to modify or hybridize FDO with new recent algorithms. MFDO can also be used to solve real world problems in the field on medicine and business planning.

ACKNOWLEDGMENT

The authors would like to thank Charmo University for their support during the preparation of this work.

REFERENCES

[1] Z. Beheshti and S. M. H. Shamsuddin, "A review of population-based meta-heuristic algorithms," *Int. J. Adv. Soft Comput. Appl.*, vol. 5, no. 1, pp. 1–35, 2013.
[2] A. W. Mohamed, A. A. Hadi, and A. K. Mohamed, "Gaining-sharing knowledge based algorithm for solving optimization problems: A novel nature-inspired algorithm," *Int. J. Mach. Learn. Cybern.*, vol. 11, pp. 1501–1529, Jul. 2020, doi: 10.1007/s13042-019-01053-x.
[3] A. Colorni, M. Dorigo, and V. Maniezzo, *A Genetic Algorithm to Solve the Timetable Problem*. Milan, Italy: Politecnico di Milano, 1992, pp. 60–90.
[4] D. Shrivastava, S. Sanyal, A. K. Maji, and D. Kandar, "Bone cancer detection using machine learning techniques," in *Smart Healthcare for Disease Diagnosis and Prevention*. Amsterdam, The Netherlands: Elsevier, 2020, pp. 175–183, doi: 10.1016/B978-0-12-817913-0.00017-1.
[5] E. Bonabeau, M. Dorigo, and G. Thèraulaz, *From Natural to Artificial Swarm Intelligence*. London, U.K.: Oxford Univ. Press, 1999.
[6] V. Granadeiro, J. P. Duarte, J. R. Correia, and V. M. S. Leal, "Building envelope shape design in early stages of the design process: Integrating architectural design systems and energy simulation," *Autom. Construct.*, vol. 32, pp. 196–209, Jul. 2013, doi: 10.1016/j.autcon.2012.12.003.
[7] J. Simkin and C. W. Trowbridge, "Optimization problems in electromagnetics," *IEEE Trans. Magn.*, vol. 27, no. 5, pp. 4016–4019, Sep. 1991, doi: 10.1109/20.104982.
[8] A. Kaveh and S. Talatahari, "An improved ant colony optimization for constrained engineering design problems," *Eng. Comput.*, vol. 27, no. 1, pp. 155–182, Jan. 2010, doi: 10.1108/02644401011008577.

[9] D. Karaboga and B. Basturk, "Artificial Bee Colony (ABC) optimization algorithm for solving constrained optimization problems," in *Foundations of Fuzzy Logic and Soft Computing* (Lecture Notes in Computer Science), vol. 4529, P. Melin, O. Castillo, L. T. Aguilar, J. Kacprzyk, and W. Pedrycz, Eds. Berlin, Germany: Springer, 2007, doi: 10.1007/978-3-540-72950-1_77.

[10] M. Fesanghary and M. M. Ardehali, "A novel meta-heuristic optimization methodology for solving various types of economic dispatch problem," *Energy*, vol. 34, no. 6, pp. 757–766, 2009, doi: 10.1016/j.energy.2009.02.007.

[11] S. Alagarsamy, R. R. Subramanian, T. Shree, S. Kannan, M. Balasubramanian, and V. Govindaraj, "Prediction of lung cancer using meta-heuristic based optimization technique: Crow search technique," in *Proc. Int. Conf. Comput., Commun., Intell. Syst. (ICCCIS)*, Feb. 2021, pp. 186–191, doi: 10.1109/ICCCIS51004.2021.9397199.

[12] A. Kaveh and Y. Vazirinia, "Construction site layout planning problem using Metaheuristic algorithms: A comparative study," *Iranian J. Sci. Technol., Trans. Civil Eng.*, vol. 43, no. 2, pp. 105–115, Jun. 2019, doi: 10.1007/s40996-018-0148-6.

[13] H. Salimi, "Stochastic fractal search: A powerful metaheuristic algorithm," *Knowl.-Based Syst.*, vol. 75, pp. 1–18, Feb. 2015, doi: 10.1016/j.knosys.2014.07.025.

[14] I. Fister, Jr., X.-S. Yang, I. Fister, J. Brest, and D. Fister, "A brief review of nature-inspired algorithms for optimization," 2013, *arXiv:1307.4186*.

[15] J. H. Holland, "Genetic algorithms," *Sci. Amer.*, vol. 267, no. 1, pp. 66–73, 1992.

[16] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. ICNN Int. Conf. Neural Netw.*, vol. 4, 1995, pp. 1942–1948, doi: 10.1007/s11721-007-0002-0.

[17] M. Dorigo and G. Di Caro, "Ant colony optimization: A new meta-heuristic," in *Proc. Congr. Evol. Comput. (CEC)*, Jul. 1999, pp. 1470–1477, doi: 10.1109/CEC.1999.782657.

[18] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Fac. Eng., Dept. Comput. Eng., Erciyes Univ., Kayseri, Turkey, Tech. Rep. tr06, 2005, pp. 1–10, vol. 200.

[19] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO)*. Cham, Switzerland: Springer, 2010, pp. 65–74, doi: 10.1007/978-3-642-12538-6_6.

[20] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proc. World Congr. Nature Biologically Inspired Comput. (NaBIC)*, 2009, pp. 210–214, doi: 10.1109/NABIC.2009.5393690.

[21] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Proc. Int. Symp. Stochastic Algorithms*, 2009, pp. 169–178, doi: 10.1007/978-3-642-04944-6_14.

[22] A. H. Gandomi and A. H. Alavi, "Krill herd: A new bio-inspired optimization algorithm," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 17, pp. 4831–4845, May 2012, doi: 10.1016/j.cnsns.2012.05.010.

[23] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2019, doi: 10.1016/j.advengsoft.2016.01.008.

[24] T. R. Biyanto, M. N. Syamsi, H. Y. Fibrianto, N. Afdanny, A. H. Rahman, K. S. Gunawan, and J. A. D. Pratama, "Optimization of energy efficiency and conservation in green building design using Duelist, Killer–Whale and rain-water algorithms," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 267, no. 1, 2017, Art. no. 12036, doi: 10.1088/1757-899X/267/1/012036.

[25] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: Theory and application," *Adv. Eng. Softw.*, vol. 105, pp. 30–47, Mar. 2017, doi: 10.1016/j.advengsoft.2017.01.004.

[26] H. Shayanfar and F. S. Gharehchopogh, "Farmland fertility: A new metaheuristic algorithm for solving continuous optimization problems," *Appl. Soft Comput.*, vol. 71, pp. 728–746, Oct. 2018, doi: 10.1016/j.asoc.2018.07.033.

[27] S. Harifi, M. Khalilian, J. Mohammadzadeh, and S. Ebrahimnejad, "Emperor penguins colony: A new Metaheuristic algorithm for optimization," *Evol. Intell.*, vol. 12, no. 2, pp. 211–226, Jun. 2019, doi: 10.1007/s12065-019-00212-x.

[28] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019, doi: 10.1016/j.future.2019.02.028.

[29] C. M. Rahman and T. A. Rashid, "A new evolutionary algorithm: Learner performance based behavior algorithm," *Egyptian Informat. J.*, vol. 22, no. 2, pp. 213–223, Jul. 2021, doi: 10.1016/j.eij.2020.08.003.

[30] J. M. Abdullah and T. A. Rashid, "Fitness dependent optimizer: Inspired by the bee swarming reproductive process," *IEEE Access*, vol. 7, pp. 43473–43486, 2019, doi: 10.1109/ACCESS.2019.2907012.

[31] P. C. Chiu, A. Selamat, O. Krejcar, and K. K. Kuok, "Hybrid sine cosine and fitness dependent optimizer for global optimization," *IEEE Access*, vol. 9, pp. 128601–128622, 2021, doi: 10.1109/ACCESS.2021.3111033.

[32] D. S. Abdul-Minaam, W. M. E. S. Al-Mutairi, M. A. Awad, and W. H. El-Ashmawi, "An adaptive fitness-dependent optimizer for the one-dimensional bin packing problem," *IEEE Access*, vol. 8, pp. 97959–97974, 2020, doi: 10.1109/ACCESS.2020.2985752.

[33] H. M. Mohammed and T. A. Rashid, "Chaotic fitness-dependent optimizer for planning and engineering design," *Soft Comput.*, vol. 25, no. 22, pp. 14281–14295, Nov. 2021, doi: 10.1007/s00500-021-06135-z.

[34] A. Seyed-Alireza, "Human behavior-based optimization: A novel metaheuristic approach to solve complex optimization problems," *Neural Comput. Appl.*, vol. 28, no. 1, pp. 233–244, Dec. 2017, doi: 10.1007/s00521-016-2334-4.

[35] C. Fuchs, S. Spolaor, M. S. Nobile, and U. Kaymak, "A swarm intelligence approach to avoid local optima in fuzzy C-means clustering," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jun. 2019, pp. 1–6, doi: 10.1109/FUZZ-IEEE2019.8858940.

[36] J. C. Bansal and S. Singh, "A better exploration strategy in grey wolf optimizer," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 1, pp. 1099–1118, Jan. 2021, doi: 10.1007/s12652-020-02153-1.

[37] J. Xu and J. Zhang, "Exploration-exploitation tradeoffs in metaheuristics: Survey and analysis," in *Proc. 33rd Chin. Control Conf.*, Jul. 2014, pp. 8633–8638, doi: 10.1109/ChiCC.2014.6896450.

[38] D. A. Muhammed, S. A. M. Saeed, and T. A. Rashid, "Improved fitness-dependent optimizer algorithm," *IEEE Access*, vol. 8, pp. 19074–19088, 2020, doi: 10.1109/ACCESS.2020.2968064.

[39] G. I. Sayed, G. Khoriba, and M. H. Haggag, "A novel chaotic salp swarm algorithm for global optimization and feature selection," *Appl. Intell.*, vol. 48, no. 10, pp. 3462–3481, 2018, doi: 10.1007/s10489-018-1158-6.

[40] S. Arora and P. Anand, "Chaotic grasshopper optimization algorithm for global optimization," *Neural Comput. Appl.*, vol. 31, no. 8, pp. 4385–4405, Aug. 2019, doi: 10.1007/s00521-018-3343-2.

[41] C. M. Rahman, T. A. Rashid, A. Alsadoon, N. Bacanin, P. Fattah, and S. Mirjalili, "A survey on dragonfly algorithm and its applications in engineering," *Evol. Intell.*, vol. 14, pp. 1–21, Aug. 2021, doi: 10.1007/s12065-021-00659-x.

[42] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016, doi: 10.1016/j.knosys.2015.12.022.

[43] A. Basak, S. Pal, S. Das, and A. Abraham, "Circular antenna array synthesis with a differential invasive weed optimization algorithm," in *Proc. 10th Int. Conf. Hybrid Intell. Syst.*, Aug. 2010, pp. 153–158, doi: 10.1109/HIS.2010.5600021.

[44] S. M. S. Hasan and S. W. Ellingson, "Integration of simple antennas to multiband receivers using a novel multiplexer design methodology," *IEEE Trans. Antennas Propag.*, vol. 60, no. 3, pp. 1550–1556, Mar. 2012, doi: 10.1109/TAP.2011.2180309.

[45] M. Akcakaya and A. Nehorai, "Adaptive MIMO radar design and detection in compound-Gaussian clutter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 3, pp. 2200–2207, Jul. 2011, doi: 10.1109/TAES.2011.5937292.

[46] A. A. Heidari, M. Heyrani, and M. Nakhkash, "A dual-band circularly polarized stub loaded microstrip patch antenna for GPS applications," *Prog. Electromagn. Res.*, vol. 92, pp. 195–208, 2009.

[47] P. J. Gorman, M. D. Gregory, and D. H. Werner, "Design of ultra-wideband, aperiodic antenna arrays with the CMA evolutionary strategy," *IEEE Trans. Antennas Propag.*, vol. 62, no. 4, pp. 1663–1672, Apr. 2014, doi: 10.1109/TAP.2013.2287904.

[48] B. A. Wilking, M. J. Lebofsky, and G. H. Rieke, "The wavelength dependence of interstellar linear polarization-stars with extreme values of lambda/max," *Astron. J.*, vol. 87, pp. 695–697, Apr. 1982.

[49] L. Der, *Frequency Modulation (FM) Tutorial*. Austin, TX, USA: Silicon Lab., 2008.

[50] Y. Sun and Z. Wang, "Asynchronous and stochastic dimension updating PSO and its application to parameter estimation for frequency modulated (FM) sound waves," in *Proc. IEEE Int. Conf. Prog. Informat. Comput. (PIC)*, Dec. 2015, pp. 583–587, doi: 10.1109/PIC.2015.7489915.

[51] J. N. D. Gupta and J. C. Ho, "A new heuristic algorithm for the one-dimensional bin-packing problem," *Prod. Planning Control*, vol. 10, no. 6, pp. 598–603, Jan. 1999, doi: 10.1080/095372899232894.

[52] G. N. Rouskas and C. Bandikatla, "Recursive first fit: A highly parallel optimal solution to spectrum allocation," *J. Opt. Commun. Netw.*, vol. 14, no. 4, pp. 165–176, 2022, doi: 10.1364/JOCN.445492.

[53] S. Bassi, M. Mishra, and E. Omizegba, "Automatic tuning of proportional-integral-derivative (PID) controller using particle swarm optimization (PSO) algorithm," *Int. J. Artif. Intell. Appl.*, vol. 2, no. 4, p. 25, 2011, doi: 10.5121/ijaia.2011.2403.

[54] A. Daraz, S. Abdullah, H. Mokhlis, I. U. Haq, G. Fareed, and N. N. Mansor, "Fitness dependent optimizer-based automatic generation control of multi-source interconnected power system with non-linearities," *IEEE Access*, vol. 8, pp. 100989–101003, 2020, doi: 10.1109/ACCESS.2020.2998127.

[55] S. Cao, W. Song, and W. Lv, "Modeling pedestrian evacuation with guiders based on a multi-grid model," *Phys. Lett. A*, vol. 380, no. 4, pp. 540–547, Feb. 2016, doi: 10.1016/j.physleta.2015.11.028.

[56] G. E. J. Du, "Solving design of pressure vessel engineering problem using a fruit fly optimization algorithm," *Int. J. Simul., Syst., Sci. Technol.*, vol. 17, no. 43, pp. 1–7, Jan. 2016, doi: 10.5013/IJSSST.a.17.43.05.

[57] A. Salman, I. Ahmad, and S. Al-Madani, "Particle swarm optimization for task assignment problem," *Microprocessors Microsyst.*, vol. 26, no. 8, pp. 363–371, 2002.

[58] A. S. Shamloo, S. Shahkar, and A. Madadi, "Numerical solution of the Fredholme–Volterra integral equation by the Sinc function," *Amer. J. Comput. Math.*, vol. 2, no. 2, pp. 136–142, 2012, doi: 10.4236/ajcm.2012.22019.

[59] K. V. Price, N. H. Awad, M. Z. Ali, and P. N. Suganthan, "Problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization," Nanyang Technol. Univ. Singapore, Singapore, Tech. Rep., 2018.

[60] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[61] M. Khishe and M. R. Mosavi, "Chimp optimization algorithm," *Expert Syst. Appl.*, vol. 149, Jul. 2020, Art. no. 113338, doi: 10.1016/j.eswa.2020.113338.

[62] S. Arora and S. Singh, "Butterfly optimization algorithm: A novel approach for global optimization," *Soft Comput.*, vol. 23, no. 3, pp. 715–734, 2018, doi: 10.1007/s00500-018-3102-4.

[63] R. S. M. D. Barros, J. I. G. Hidalgo, and D. R. D. L. Cabral, "Wilcoxon rank sum test drift detector," *Neurocomputing*, vol. 275, pp. 1954–1963, Jan. 2018, doi: 10.1016/j.neucom.2017.10.051.

[64] K. Deb, "Optimal design of a welded beam via genetic algorithms," *AIAA J.*, vol. 29, no. 11, pp. 2013–2015, Nov. 1991, doi: 10.2514/3.10834.

[65] M. Sarkar and T. K. Roy, "Optimization of welded beam structure using neutrosophic optimization technique: A comparative study," *Int. J. Fuzzy Syst.*, vol. 20, no. 3, pp. 847–860, Mar. 2018, doi: 10.1007/s40815-017-0362-6.

[66] A. T. Kamil, H. M. Saleh, and I. H. Abd-Alla, "A multi-swarm structure for particle swarm optimization: Solving the welded beam design problem," *J. Phys., Conf. Ser.*, vol. 1804, no. 1, 2021, Art. no. 12012, doi: 10.1088/1742-6596/1804/1/012012.

[67] A. R. Yildiz, "A new hybrid particle swarm optimization approach for structural design optimization in the automotive industry," *Proc. Inst. Mech. Eng., D, J. Automobile Eng.*, vol. 226, no. 10, pp. 1340–1351, Oct. 2012, doi: 10.1177/0954407012443636.

[68] E. H. Houssein, M. R. Saad, F. A. Hashim, H. Shaban, and M. Hassaballah, "Lévy flight distribution: A new Metaheuristic algorithm for solving engineering optimization problems," *Eng. Appl. Artif. Intell.*, vol. 94, Sep. 2020, Art. no. 103731, doi: 10.1016/j.engappai.2020.103731.

[69] E. H. Houssein, M. A. Mahdy, M. J. Blondin, D. Shebl, and W. M. Mohamed, "Hybrid slime mould algorithm with adaptive guided differential evolution algorithm for combinatorial and global optimization problems," *Expert Syst. Appl.*, vol. 174, Jul. 2021, Art. no. 114689, doi: 10.1016/j.eswa.2021.114689.

[70] K. Deb and M. Goyal, "A flexible optimization procedure for mechanical component design based on genetic adaptive search," *ASME. J. Mech. Des.*, vol. 120, no. 2, pp. 162–164, Jun. 1998, doi: 10.1115/1.2826954.

[71] S. Mirjalili, "Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Comput. Appl.*, vol. 27, no. 4, pp. 1053–1073, 2016, doi: 10.1007/s00521-015-1920-1.

[72] J. N. Siddall, *Optimal Engineering Design: Principles and Applications*. Boca Raton, FL, USA: CRC Press, 1982.