Int. J. Nonlinear Anal. Appl. In Press, 15–34
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.27519.3635

Novel nature-inspired meta-heuristic optimization algorithm based on hybrid dolphin and sparrow optimization

Shahab Wahhab Kareem[a,b], Amin Salih Mohammed[c], Farah Sami Khoshaba[a]

[a]*Department of Information Systems Engineering, Erbil Polytechnic University, Erbil, Iraq*

[b]*Department of Information Technology, Lebanese French University, Erbil, Iraq*

[c]*Department of Software and Informatics, Salahaddin University-Erbil, Iraq*

(Communicated by Madjid Eshaghi Gordji)

Abstract

The increasing difficulty of actual-world optimization problems has prompted computer researchers to produce process improvement techniques regularly. Metaheuristic and evolutionary computation are popular in nature-inspired optimization methods. This paper introduces hybrid dolphin and sparrow optimization (DSO), which is a modification of a new metaphorical algorithm based on the natural behavior of sparrows and dolphins. Various adaptive and arbitrary variables are combined within this algorithm to indicate the exploitation and investigation of the exploration area in various discoveries of optimization. Multiple test strategies are used to calculate DSO performance. Initially, a collection of experiment events, including unimodal, multimodal, and composite functions, is applied to examine the exploitation, exploration, local optima avoidance, and convergence of DSO. Furthermore, unique metrics, such as the most suitable solution through optimization and search history, are applied to qualitatively and quantitatively examine and verify the achievement of DSO on turned 2D inspection functions. The effects of analysis functions and achievement metrics show that the proposed method can search various regions of a search space, provide local optima avoidance, converge toward the global optimum, and utilize encouraging areas of a search range while optimization proceeds efficiently. The DSO algorithm achieves a regular frame for an airfoil with a low drag, which explains that the methods are efficient in improving physical difficulties, including restrained plus unknown search spaces.

Keywords: Meta-heuristic, Swarm intelligent, Global optimum, Dolphin, Sparrow optimization
2020 MSC: 68T05

1 Introduction

Population-based swarm intelligence algorithms have been widely used in various optimization difficulties. Unlike standard single-point-based methods, such as hill-climbing algorithms, a population-based swarm intelligence algorithm is based on a set of objects (group) that improve difficulties by providing knowledge to support and/or compete between themselves [19]. In previous decades, the procedures of meta-heuristic optimization generally relied on elementary ideas [15]. Moreover, previous studies reported several statements on optimization methods within various areas. In the present study, the author discusses meta-heuristics as a common strategy [15, 7]. Four primary reasons can explain

Email addresses: shahab.kareem@epu.edu.iq (Shahab Wahhab Kareem), amin.mohammed@su.edu.krd (Amin Salih Mohammed), farah.xoshibi@epu.edu.iq (Farah Sami Khoshaba)

Received: February 2022 *Accepted:* May 2022

this topic; simplicity, flexibility, derivation-free tool, and avoiding local optimal concepts can be implemented on computers. Moreover, integrity permits proposing beyond new checks, hybridizing two or more meta-heuristics, or improving recent meta-heuristics. Integrity also allows a quick and straightforward ephemeral inference on different specialists and can be used for their difficulties [12]. Exploitation and exploration are a significant concern to specific meta-heuristic swarm intelligence methods. In recent years, several population-based swarm intelligence methods have survived, such as particle swarm optimization (PSO), ant colony optimization [1], artificial bee colony algorithm [11, 16], imperialist competitive algorithm [8], and brainstorm optimization [20, 10].

Meanwhile, meta-heuristic algorithms intend to obtain a particular extremum from each optimized problem, in which the benefit of each objective function is consistently developed until that extremum duration is obtained [14]. Whether the method can be global or local, it is separated into global and local search algorithms. Local extremum search algorithms are provided to prepare one of the extremums on the collection about possible solutions while the actual function is the minimum or maximum value [11]. To obtain the optimum while each type concerning a specific optimized function is not entirely comprehended, or its arrangement is exceptionally complicated, the swarm intelligence algorithm is applied [18, 12]. The performance concerning a search method during the optimum implies the probability of obtaining the convergence and extremum solution regarding a problem [9]. The standard analytical difficulty in each field of engineering and scientific development is an improvement, obtaining the most appropriate solutions. Optimization techniques are stochastic or deterministic. Survival techniques for solving optimization difficulties need enormous computational expenses, thus solving difficulties such as random optimization [7]. A reliable approach to solving the optimization problem is applying meta-heuristics depending on a community of solutions or iterative development (either in the swarm or evolutionary technology) [2]. Dolphin echolocation is a new optimization method that is presented in this paper. This strategy impersonates techniques utilized by dolphins. Dolphins produce a sonar voice to meet their objective and change the sonar to adjust their area. Dolphin echolocation is depicted in Fig. 1. This phenomenon is mimicked as the main feature of the new optimization method.

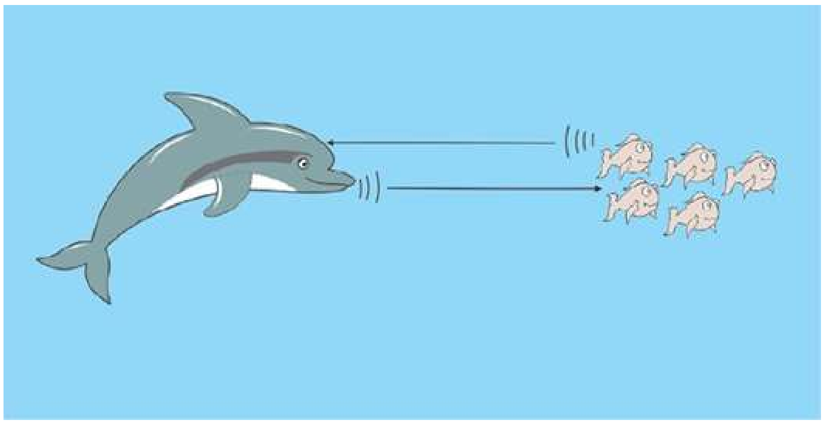


Figure 1: Dolphin hunting action

Regarding the above arguments, this paper proposes a novel optimization strategy called dolphin sparrow optimization (DSO) for hybrid swarm intelligence. The key contributions are summarized as follows: (1) DSO is presented inspired by the foraging and anti-predation activities of the dolphin and sparrow population; (2) instead of using the presented DSO, in both discovery and utilization of the optimization search space are enhanced to some extent; and (3) the presented DSO is achieved in practical engineering problems. Finally, several comparative studies are carried out to assess the efficacy and efficiency of the proposed algorithm. Simulation experiments demonstrate that the proposed DSO is better in search accuracy, premature convergence, consistency, and avoiding local optimal value than other traditional approaches. This article is arranged as follows. Following the introduction, Section 2 presents a literature review on general optimization. The dolphin optimization algorithm is introduced in Section 3. In Section 4, sparrow search optimization algorithms are presented. Section 5 provides the detailed methodology and performance evaluation of the proposed method. The last section lists the conclusions.

2 Dolphin swarm behavior

As one of the most intelligent species, dolphin features attractive biological characteristics and dynamic behaviors worthy of consideration.

1. Echolocation: Dolphins have decent visual acuity; however, perfect vision at low visibility levels opens this species to exploitation. In general, dolphins use echolocation to hunt for prey. According to echo strength, dolphins can calculate their prey's distance, position, and shape. Utilizing the echo can develop an understanding of the environment. 2. Cooperation and labor division: A few examples show that dolphins do not work in their interest; instead, they take advantage of teamwork and collaborative effort. A dolphin is likely to be able to catch prey while the others observe. A dolphin calls upon other dolphins to help hunt while he is injured. Furthermore, dolphins have a defined role distribution. For instance, dolphins closest to the prey monitor the prey's movements, whereas dolphins farther away are responsible for circling the prey [11]. 3. Exchanges of information: Recent experiments have proven that dolphins can transmit information. They may send different messages and have their language system with tones at various frequencies. Information sharing plays a vital role in the predatory process. In particular, it allows dolphins to contact each other and informs other dolphins of the position of their prey. Dolphins put into practice methods to increase predation by gaining information from other predators. Dolphin predation has three phases. Most dolphins begin their hunt for prey by creating noises and then following it up with echolocation to discern their surroundings. Dolphins at this protocol's second level dole out their info. When dolphins encounter a large catch, their friends join in to help. Dolphins that have learned how to hunt form a hunting circle around their prey. Finally, dolphins complete the hunt by simply turning to eat the meal.

Algorithm for the dolphin swarm

The dolphin swarm algorithm (DSA) is primarily implemented through modeling the biological features and lifestyle habits as observed in the real predatory system of dolphins. The predatory process simulated is comparable to that previously discussed. In this part, significant definitions and pivotal stages are introduced.

Principal definitions

2.1 dolphin

To simulate the predatory phase of dolphins, they require a certain number of dolphins depending on swarm intelligence. Every dolphin in the painting offers a different answer to the optimization issue. According to this study, dolphins (Doli) are described as $[x_1, x_2, ..., x_D]$. A D-dimensional solution is achievable by optimizing the data in the following form: $x_j(j = 1, 2, ..., D)$ with N being the size of dolphins and $x_j(j = 1, 2, ..., D)$ being the portion of each dimension to be improved.

2.2 Independent optimization technique and neighborhood optimization method

The parameters of L and neighborhood optimum solution are linked to dolphins, with L being an individual solution and the other being a local solution (denoted as K). For each Doli $(i = 1, 2, ..., N)$, two variables are associated: $L_i(i = 1, 2, ..., N)$ denotes the best possible solution found at a single instant, and $K_i(i = 1, 2, ..., N)$ represents the best possible answer obtained from others or found over time.

2.3 Fitness

Fitness E helps us decide whether a specific solution is superior. In DSA, a fitness function is used to quantify E, thus the better it is at minimizing E. To better explain the details of the experiment, this research defines the fitness function as Fitness(X).

2.4 Distance

Distances used in DSA have three types. The first is the distance $DD_{i,j}$ between Dol_i and Dol_j, which can be defined as follows:

$$DD_{i,j} = \parallel Dol_i - Dol_j \parallel, i, j = 1, 2, ..., N i \neq j \tag{1}$$

The second is the distance DK_i between Dol_i and K_i, which can be formulated as follows:

$$DK_i = \parallel Doli - Ki \parallel \; i = 1, 2, \ldots, N \tag{2}$$

A third way to represent distances is between L_i and K_i, which has a name (DKL_i) and a formula to describe it:

$$DKL = \parallel L_i - K_i \parallel \; i = 1, 2, \ldots, N \tag{3}$$

$DD_{i,j}$ inhibits the transfer of knowledge between Dol_i and Dol_i; therefore, DKi and DKLi affect Doli's action in the predation process.

i- Search phase

Every Dolphin explores its surrounding area in the search process to make noises in arbitrary directions toward M. Comparably, in this analysis, the noise is defined as $V_i = [v_1, v_2, \ldots, v_D]T(i = 1, 2, \ldots, M)$, where N is the total number of noises and $V_j(j = 1, 2, \ldots, D)$ is the element of every dimension, including the component of the noise direction. Moreover, sound accepts $\parallel V_i \parallel = speed(i = 1, 2, \ldots, M)$, where "speed" is a factor reflecting the noise speed attribute. It justifies a maximum search time of T_1 to avoid dolphins from being lost in the search stage. The sound Vj that Doli $(i = 1, 2, ..., N)$ renders at t should scan for a new X_{ijt} resolution inside the peak search time T_1, which can be defined by

$$X_{ijt} = Dol_i - V_{jt} \tag{4}$$

For the latest X_{ijt} solution that Dol_i is having, E_{ijt}'s fitness is determined on the following basis:

$$E_{ijt} = Fitness(X_{ijt}) \tag{5}$$

If

$$E_{iab} = minj = 1, 2, \ldots, M, \;\; i = 1, 2, \ldots, T_i$$

$$E_{ijt} = minj = 1, 2, \ldots, M; i = 1, 2, \ldots, T_i \; Fitness(X_{ijt}) \tag{6}$$

Then, Dol_i's independent optimum L_i solution is calculated as

$$L_i = X_{iab} \tag{7}$$

If

$$Fitness(L_i) < Fitness(K_i), \tag{8}$$

L_i then substitutes K_i; therefore, K_i is not adjusted. Dol_i are updating their $L_i(i = 1, 2, ..., N)$. DSA joins the call process as well as K_i (if they are being revised).

ii- Reception phase

Even though the reception phase occurs only after the call phase, this phase must still be elaborated first. For DSA, an N-N-order matrix called the "transmission time matrix" (TS) preserves the interchange (as well as the call and reception phases), where $TS_{i,j}$ denotes the time remaining for the noise to pass through Dol_j to Dol_i. All the words $TS_{i,j}(i = 1, 2, ..., N; j = 1, 2, ..., N)$ in the transmission time matrix reduce by one until DSA reaches the reception process to demonstrate that the noises scatter over one unit of time. Then, any term $TS_{i,j}$ in the matrix needs to be confirmed by DSA, and if

$$TS_{i,j} = 0 \tag{9}$$

Then Dol_i is hearing the noise sent to Doli from Dol_j. To mean that the corresponding sound has been received, we must replace $TS_{i,j}$ with a new time term called "full transmission time" (T_2). In comparing K_i and K_j, if

$$Fitness(K_i) > Fitness(K_j), \tag{10}$$

Then K_j will substitute K_i; otherwise, K_i will not adapt because Eq reaches all the definitions in the TS matrix and DSA reaches the stage of predation.

iii- Call phase

Every Dol_i makes noise in the call process to warn others. The transmission time matrix TS must be changed as follows: For K_i, K_j, and $TS_{i,j}$, if

$$Fitness(K_i) > Fitness(K_j), \tag{11}$$

and

$$TS_{i,j} > \left[\frac{DD_{ij}}{A.speed}\right] \tag{12}$$

where speed is a fixed value expressing the noise's velocity, while it is the constant describing the acceleration that will make noises propagate faster if the speed is prolonged. $TS_{i,j}$ is then modified as follows:

$$TS_{i,j} > \left[\frac{DD_{ij}}{A.speed}\right] \tag{13}$$

Even after the terms $TS_{i,j}(i = 1, 2, \dots, N; j = 1, 2, \dots, N)$ have been modified (if they are being modified), DSA reaches the reception stage.

iv- Predation phase

Within the predation phase, each Dol_i should calculate the encompassing sweep R_2. For each Doli, the look span R_1, which speaks to the greatest extend within the look stage, can be calculated as

$$R_1 = T_1 \times speed \tag{14}$$

For the most part, the calculation of encompassing sweep R_2 and the overhaul of the Dol_i's position ought to be talked about in three cases. We take $Dol_i(i = 1, 2, \dots, N)$ as an illustration to demonstrate these three cases. (a) For the currently known information of $Dol_i(i = 1, 2, \dots, N)$, if

$$DK_i \leq R_1 \times speed \tag{15}$$

Then the neighborhood ideal K_i of Dol_i is inside the look run. For simplicity, in this case, DSA respects the ideal arrangement L_i as K_i (fig. 2(a)). In this case, the encompassing span R_2 can be calculated as follows:

$$R_2 = (1 - \frac{2}{e})DK, e > 2 \tag{16}$$

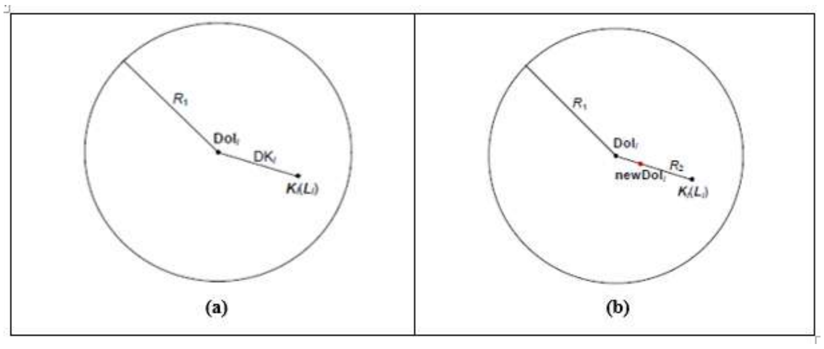


Figure 2: (a) Predation of Case (a), (b). Result of case (a)

Where e is the radius diminishment coefficient, which is more prominent than 2 and more often than not set as 3 or 4. R_2 continuously merges to zero. After obtaining the encompassing sweep R_2, we will obtain Doli's modern position newDoli:

$$NewDol_i = K_i + ((Dol_i - K_i)/Dk_i).R_2) \tag{17}$$

Dol_i moves toward K_i and stops at the position that's R_2 removed absent from K_i (fig. 2(b)). For the right now known data of $Dol_i(i = 1, 2, \dots, N)$, if

$$DK_i > R_l, \tag{18}$$

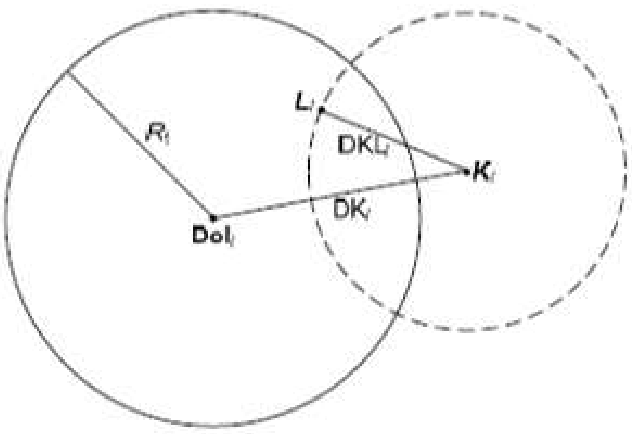


Figure 3: Prediction of Case (b)

and

$$DK \geq DKL \tag{19}$$

Then Dol_i overhauls K_i by accepting data from others, and Li is closer to Dol_i than K_i is (Fig. 3).

In this case, the encompassing span R_2 can be calculated as follows:

$$R_2 = \left(1 - \frac{\frac{DK_i}{Fitness(K_i)} + \frac{DK_i - DKL_i}{Fitness(L_i)}}{e.DK_i \frac{1}{Fitness(K_i)}}\right) DK_i, e > 2 \tag{20}$$

After obtaining the encompassing sweep R_2, Doli's unused position newDol_i can be calculated as

$$newDol_i = K_i + (Random \parallel Random \parallel) R_2 \tag{21}$$

Precisely, Dol_i moves to an arbitrary position with R_2 removed absent from Ki (Fig. 4). (c) For the presently known data of $Dol_i (i = 1, 2, \ldots, N)$, if it fulfills imbalance (18) and

$$DK_i > DKL_i, \tag{22}$$

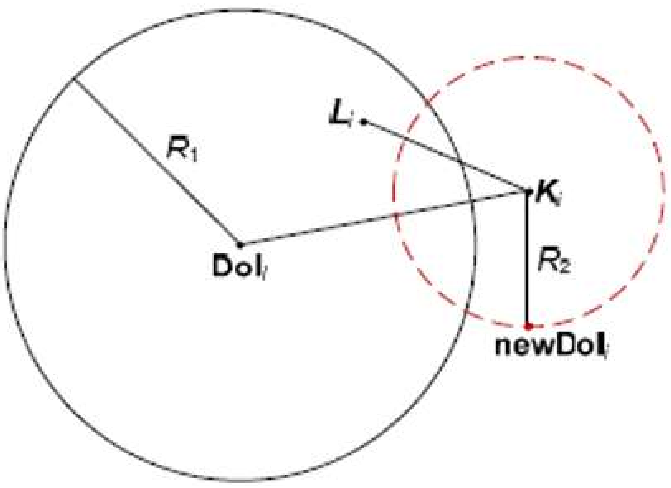


Figure 4: Result of case (b)

Then Dol_i overhauls K_i by obtaining data from others, and K_i is closer to Dol_i than Li is (Fig. 5). In this case, the encompassing sweep R_2 can be calculated as

$$R_2 = \left(1 - \frac{\frac{DK_i}{Fitness(K_i)} + \frac{DK_i - DKL_i}{Fitness(L_i)}}{e.DK_i \frac{1}{Fitness(K_i)}}\right) DK_i, e > 2 \tag{23}$$

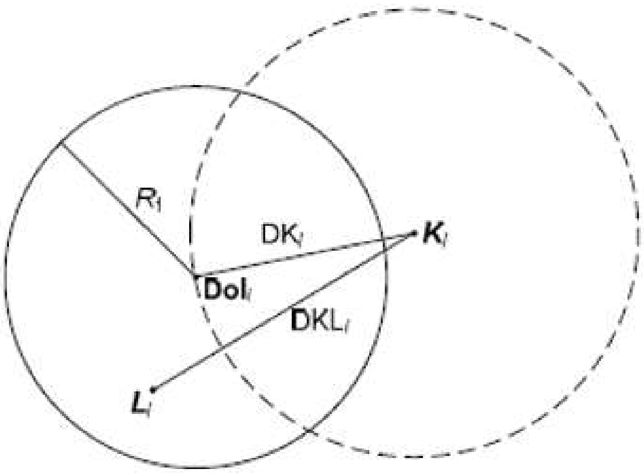


Figure 5: Prediction of Case (b)

After obtaining the encompassing sweep R2, Dol_is modern position newDol_i can be calculated using Eq. (21). In specific, Dol_i moves to an irregular position that's R_2 separate absent from Ki (Fig. 6). After Dol_i moves to the position newDol_i, comparing newDoli with Ki in terms of wellness, on the off chance that $Fitness(newDol) < Fitness(K_i)$, at that point, K_i is supplanted by newDol_i; otherwise, K_i does not change. After all the $Dol_i(i = 1, 2, \dots, N)$ overhaul their positions and K_i (in case it can be upgraded), whether DSA meets the conclusion condition is decided. In case the conclusion condition is fulfilled, DSA enters the end-stage; otherwise, DSA enters the look stage again.

3 Sparrow search algorithm (SSA)

3.1 Biological characteristics

Similar to other little birds, the sparrow is emphatically keen and has a solid memory. Hostage house sparrows have two types: makers and scroungers [7, 11, 12, 13, 14, 18, 22, 21, 6]. Similarly, the energy stores of the people may assume a significant part when the sparrow picks diverse scavenging methodologies, and sparrows with low energy hold search more [21]. Birds situated on the fringe of the populace are bound to be assaulted by hunters and continually attempt to improve their position [3, 4]. Creatures situated in the middle may draw nearer to their neighbors to limit their area of peril [17, 5]. In addition, sparrows show the common intuition of interest in all things while being consistently careful. For instance, when a bird distinguishes a hunter, the whole gathering takes off [17].

3.2 Mathematical model and algorithm

Concurring to the past depiction of the sparrows, the following rules were used.

(1) Makers ordinarily have elevated vitality levels and give scrounging zones or bearings for all borrowers.

(2) Once the sparrow distinguishes the hunter, individuals begin to chirp as troubling signs. If the security risk outweighs the alarm esteem, the producer must bring all employees to a safe zone.

(3) As long as the sparrows look for food, they may all become producers. The proportion of creators and scroungers remains constant among the total population.

(4) Manufacturers take care of the most energetic sparrows first. A small number of people, desperate for food, would try to reach places to replenish their energy to stay alive.

(5) Scroungers follow the best feed giver they know has the most incredible food and search for it. Meanwhile, scavengers look for food to increase their population, while a few people target the food source creators. Using virtual sparrows in the recreation attempt, we should be looking for food.

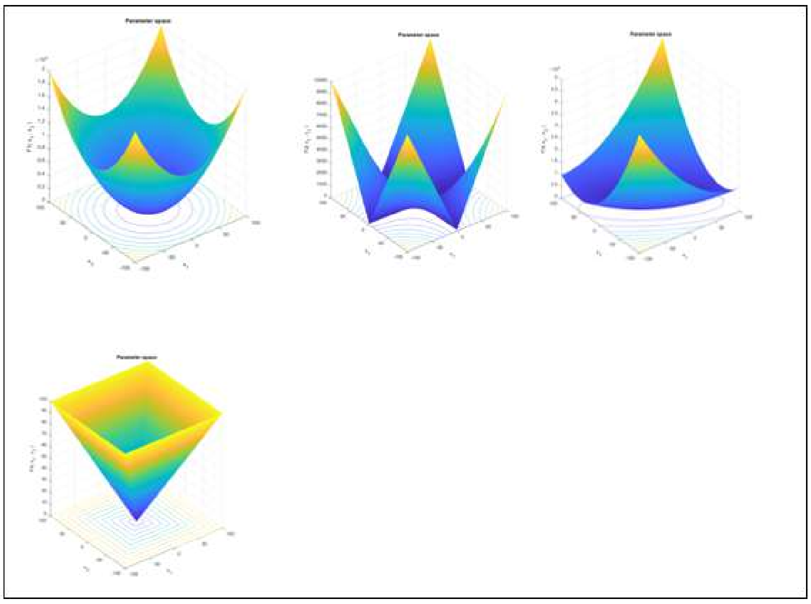


Figure 6: Unimodal functions

(6) When aware of the danger, sparrows close to the outside of the crowd congregate to find shelter, whereas sparrows in the middle go in any direction to be with others. Sparrow habitat is included in the following matrix:

$$\begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,d} \\ x_{2,1} & x_{2,2} & \dots & x_{2,d} \\ & & \dots & \\ x_{n,1} & x_{n,2} & \dots & x_{n,d} \end{bmatrix} \tag{24}$$

where n is the number of sparrows and d is the measurement of the factors to be optimized. At that point, the wellness esteem of all sparrows can be communicated by the taking after vector:

$$\begin{bmatrix} f[x_{1,1}, x_{1,2}, \dots, x_{1,d}] \\ f[x_{2,1}, x_{2,2}, \dots, x_{2,d}] \\ \dots \\ f[x_{n,1}, x_{n,2}, \dots, x_{n,d}] \end{bmatrix} \tag{25}$$

The consideration of each individual's wellbeing in FX is equivalent to the number of sparrows regarding sparrows, whereas the push of FX addresses the number of sparrows. On the inside, healthy operators with high self-esteem may include their preferred food group in their appearance regimen. Then, the developers have tended to the project's practicality, both economically and empowering the population as a whole. Therefore, the creative types find resources in much more locations than the naysayers do. During each cycle, according to rules (1) and (2), the producer's area is updated:

$$X_{ij}^{t+1} = \left\{ \begin{array}{cc} X_{ij}^{t}.exp(\frac{-i}{\alpha iter_{max}}) & if R_2 < ST \\ X_{ij}^{t} + Q.L & if R_2 \geq ST \end{array} \right\}. \tag{26}$$

The estimate for the ith sparrow at the jth cycle in the t-th round is given by X_{ij}^{t}. With max cycles, you may expect high performance. α may be a sporadic number because it belongs to the open interval (0, 1). Alarm regard and security edge are addressed separately, with each of these ranges giving equal weight to each value in its span. Q is a rare number that follows the usual scattering path. L shows up with a matrix of $1 \times d$, with every square within being precisely 1. The creator enters the broad look mode when $R_2 < ST$ (R_2being less than ST, no hunters are present nearby). Two sparrows have detected the hunter, and all birds need to fly away to other safe locations immediately. The poor souls who do not abide by the new procedures should be exterminated (4) (5). Scavengers who screen producers regularly are found more often. When they learn that the author has come upon great food, they immediately take drastic action to compete for it. On a good day, they may receive help with urgent necessities and other things from the maker, regardless of whether they win (5) is as follows:

$$X_{ij}^{t+1} = \left\{ \begin{array}{cc} X_{ij}^{t}.exp(\frac{x_{worst}^{t} - X_{ij}^{t}}{\alpha iter_{max}}) & ifi > n/2 \\ X_{p}^{t+1} + |X_{ij}^{t} - X_{p}^{t+1}|.A^{+}.L & otherwise \end{array} \right\}. \quad (27)$$

The ideal role is that of the creator, which XP serves. Xworst has the most noticeably lousy region at present. A addresses a grid of dimensions 1 × d in which every segment is assigned 1 or -1. The result of this assignment is that $A+ = AT(AAT) - 1whenI > n/2$, which shows that when a participant is randomly selected from among those who have the worse health, they are doomed to starve. In our estimation, these sparrows know of the danger. Thus, 10%–20% of the population will be represented by them. Sparrows begin their lives within the broader population without knowing where they belong. The mathematical game may be described as follows if you agree to the rules (6):

$$X_{ij}^{t+1} = \left\{ \begin{array}{cc} X_{best}^{t} + \beta.|X_{ij}^{t} - X_{t}^{best}| & iff_{i} > f_{g} \\ X_{ij}^{t} + K.(\frac{|X_{ij}^{t} - x_{worst}^{t}|}{(f_{i} - f_{w}) + \varepsilon}) & iff_{i} = f_{g} \end{array} \right\}. \quad (28)$$

The target we are aiming for is X best, the perfect locale now. As the progression measures control border, B may be a range of arbitrarily positioned, uncontrolled integers that jump by a constant of 1. $K \in [-1, 1]$ could be a value that is not fully explained or justified. The subject of our program sparrow's wellbeing is one's physical condition. Fg and fw are, by a considerable extent, the most capable and recognizable overall wellbeing measurements today. A minuscule option is the most successful in avoiding zero-division-mistake.

4 Overall implementation

Calculation 1 is found inside the DSO. The essential phases in the process are the beginning, middle, and concluding phases. DS begins with dolphin initialization, along with setting the value of the sparrow parameter. Different optimization problems provide parameters to be optimized according to what they need, but the initialization of dolphins is best achieved via egalitarian randomness. The end may be set for each exercise as necessary, such as a designated amount of time, running out of stamina, and completing specific tasks. When the end condition is fulfilled, the most excellent one of $K_i(i = 1, 2, \ldots, N)$ will be the yield. For a worldwide look, we connected the worldwide sparrow optimization and connected condition (28), whereas we utilized dolphin swarm optimization as the neighborhood look utilizing conditions (1 & 2) for overhauling the position. The details of the procedure are as follows:

- Select the position haphazardly and the esteem of the parameters of the whale optimization (initialize the whale's populace $X_i(i = 1, 2, ..., n)$) and the parameters of sparrow optimization algorithms.
- If $p > 0.5$, connect conditions 2.1 and 2.2 on the off chance that $(A) > 1$ and condition 2.8 something else for upgrading the current position of the current in nearby optimization.
- Apply the sparrow caution detecting for following strategy for overhauling the current position utilizing condition (2.9); once they discover that the maker has found great nourishment, they promptly take off their current position to compete for food.
- If $p < 0.5$, connect condition 2.5 for upgrading position utilizing the winding overhauling position.
- Check in case any look specialist goes past the look space and revises it. Calculate the wellness of each look operator Overhaul in case of a distant better;

Step 1: Initialize: Arbitrarily and equally produce the beginning dolphin swarm $Dol = \{Dol1, Dol2, \ldots, DolN\}$ within the D-dimensional space, initialized temperature T0. Calculate the wellness for each dolphin, and get $FitK = FitK, 1, FitK, 2, \ldots, FitK, N$. Step 2: Begin circle: while the conclusion condition is not fulfilled do

Step 2.1: Look phase

$E_{ijt} = Fitness(Dol_i + V_{jt})$

$FitL = \{minE_{1jt}, minE_{2jt}, \ldots, minEN_{jt}\}$

$$f(x) = \left\{ \begin{array}{ll} Fit_{L,i} & ifFit_{L,i} < Fit_{K,i} \\ Fit_{L,i}, & exp[-(fitness(K_i) - (fitness(L_i)/T] > random[0,1] \\ Fit_{K,i} & otherwise \end{array} \right\}.$$

Step 2.2: call phase

$$TS_{ij} = \left\{ \begin{array}{c} [\frac{DD_{ij}}{A.speed}], ifFit_{K,i} < Fit_{K,j} andTS_{ij} > [\frac{DD_{ij}}{A.speed}] \\ TS_{ij}, \quad otherwise \\ TS_{ij}, \quad otherwise \end{array} \right\}.$$

Step 2.3: Reception phase TS_{ij} decrease one unit time ($Temp = Temp - \Delta t$)

$$Fit_{K,i} = \left\{ \begin{array}{c} Fit_{K,i} \quad if\ Fit_{K,j} < Fit_{K,i}\ and\ TS_{ij} = 0 \\ Fit_{K,i}, \quad otherwise \\ TS_{ij}, \quad otherwise \end{array} \right\}.$$

Step 2.4: Predation phase

```
    determine $DK_i$ and $DKL_i$
    if $DK_i \leq Rl$ or $exp[-(fitness(K_i) - (fitness(L_i)/T] > random[0,1]$
    $R_2 = (1 - \frac{2}{e})DK,$
    Else
    $DK_i \geq DKL_i$ , $X_{ij}^{t+1} = \{X_{best}^t + \beta.|X_{ij}^t - X_{best}^t|\}$
    Else
    $X_{ij}^{t+1} = X_{ij}^t + K.(\frac{|X_{ij}^t - x_{worst}^t|}{(f_i - f_w) + \varepsilon})$
    end if
    $Dol_i$ gets a new position, calculates its fitness, and updates FitK, i
    end while
    output the best one of $K_i (i = 1, 2, \ldots, N)$
```

5 Experimental evaluation

The proposed algorithm is benchmarked on 23 benchmark functions. The 23 benchmark functions are the conventional functions utilized by a few analysts [11, 18]. Despite the effortlessness, the creators have chosen certain test functions to compare the comes about to present-day meta-heuristics. These benchmark functions are listed in Tables 1–3, where Dim alludes to the measurement of the work, Run alludes to the sides of the function's investigation extends, and f min alludes to the ideal [7, 10, 12, 13]. These standard capacities are exchanged, overseen, expanded, and combined with the classic capacities that speak to the foremost complex with the existing standard capacities [18]. Figures6, 7 and 8 display 2D models of the performance measurement functions applied to the test system. In general, the standard functions utilized are minimization functions in addition to its ability to split within four collections: multimodal, unimodal, fixed-dimensional multimodal, multimedia, and complex functions. Specific descriptions of standardized functions are provided by the CEC 2005 technical report [14]. The DSO algorithm is carried out 30 times in each performance measure function. Tables 5 and 6 list the statistical results (standard deviation and average). The DSO algorithm is contrasted with the PSO [1] as an SI-based technique and GSA [11] as a physics-based algorithm to validate the results. In addition, five EAs are compared to the DSO algorithm: DE [7, 10, 12, 13], FEP-Fast Evolutionary Programming [18], SO- Supernova Optimizer [2], WOA-Whale Optimization Algorithm [22], and GOA-Grey Wolf Optimizer [7].

Reference functions can be divided into four classes in general: unimodal, multimodal, multimedia, fixed-dimensions multimedia, and composite. Functions F1 to F7 are unimodal because they have only one global rule. These functions enable us to approximate the optimization potential of the meta-heuristic algorithms being studied. Tables 5 and 6 show that DSO is aggressive among several meta-heuristic indicative algorithms. The current algorithm will therefore ensure very successful exploitation.

Table 1: Unimodal benchmark functions

Function	Dim	Range	F_{min}	Function Name
F1(X)=$\sum_{i=1}^{n} x_i^2$	30	[-100,100]	0	Sphere
F2(x)=$\sum_{i=1}^{n} \lvert X_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	30	[-10,10]	0	Schwefel2.22
F3(x)= $\sum_{i=1}^{n}(\sum_{j=1}^{i} x_j)^2$	30	[-100,100]	0	Schwefel 1.2
F4(x)=$max_i\{\lvert x_i\rvert, 1 \le i \le n\}$	30	[-100,100]	0	Schwefel 1.2

Table 2: Multimodal Basic Functions

Function	Dim	Range	F_{min}	Function Name
F5(x)=$\sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[-30,30]	0	Rosenbrock's
F6(x)=$\sum_{i=1}^{n}([x_i + 0.5])^2$	30	[-100,100]	0	Shifted Rosenbrocks Function
F7(x)=$\sum_{i=1}^{n} ix_i^4 + random[0,1)$	30	[-128,128]	0	Quartic

Table 3: Multimodal benchmark functions

Function	Dim	Range	F_{min}	Function Name
F8(x)=$\sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i\rvert})$	30	[-500,500]	-418.9829*5	Schwefe
F9(x)=$\sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[-5.12,5.12]	0	Shifted Rastrigins
F10(x)=-20exp(-0.2$\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$) - exp $(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))$+20+e	30	[-32,32]	0	Ackley
F11(x)=$\frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600,600]	0	Griewangk's
F12(x)=$\frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n=1}(y_i - 1)^2[1 + 10sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10,100,4)$ $y_i = 1 + \frac{x_i+1}{4}$, $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	[-50,50]	0	Schwefels Problem
F13(x)=0.1$\{sin^2(3\pi x_i) + \sum_{i=1}^{n}(x_i - 1)^2[1 + sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + sin^2(2\pi x_n + 1)]\} + \sum_{i=1}^{n} u(x_i, 10,100,4)$	30	[-50,50]	0	Expanded Extended Griewank's

Table 4: Description of fixed-dimension multimodal benchmark functions

Function	Dim	Range	F_{min}	Function Name
F14(x)=$(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6})^{-1}$	2	[-65,65]	1	Hybrid Composition Functions
F15(x)=$\sum_{i=1}^{11}[a_i-\frac{x_i(d_i^2+b_ix_2)}{h_i^2+b_ix_3+x_4}]^2$	4	[-5.5]	0.00030	
F16(x)=$4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	[-5,5]	-1.0316	Camel
F17(x)= $(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6)^2+10\left(1-\frac{1}{8\pi}\right)cosx_1+10$	2	[-5,5]	0.398	Branin
F18(x)= $(1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)x[30+(2x_1-3x_2)^2*(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$	2	[-2,2]	3	Goldstein
F19(x)=$-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2)$	3	[1,3]	3.86	Hartmann 3-D
F20(x)=$-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2)$	6	[0,1]	-3.32	Hartmann 6-D
F21(x)= $-\sum_{i=1}^{5}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	- 10.1532	Rotated Hybrid Composition
F22(x)= $-\sum_{i=1}^{7}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	- 10.4028	Rotated Hybrid Composition Function with High Condition Number Matrix
F23(x)= $-\sum_{i=1}^{10}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	- 10.5363	F23: Non-Continuous Rotated Hybrid Composition Function

The multimedia features involve many local updates that increase significantly with the severity of the problem (number of design parameters). Consequently, if the goal is to estimate the exploration potential of the optimization technique, this research problem becomes helpful. The findings in F8–F23 (multimedia and fixed multimedia functions) in Tables 5 and 6 show that DSO also has strong exploration potential. In some test problems, the current algorithm is always the most efficient or second-best algorithm. This finding is expected in the combined procedure to search mechanisms in the DSO algorithm that drive this algorithm toward global optimization. The first set of test features has no local optima, and only one optimum global is present. This characteristic makes the algorithm a good match for convergence speed calculation and algorithm exploitation. A sample of 30 search operators can set a global limit of more than 500 iterations to solve the above test functions. The algorithm is carried out 30 times for past results that may be unreliable because of the subjective nature of the meta-heuristics, and statistical results (standard deviation and average) are accumulated and reported in Tables 5 and 6.

Despite the outcome as mentioned earlier, several other tests need to be performed to ensure confidence in the efficiency of this algorithm in solving real problems and to prove and check that the DSO algorithm is highly efficient. In specific terms, the efficiency of exploration operators must be observed by optimization: how they drive near the exploration field if they experience sudden shifts to the early stages of improvement to investigate the exploration

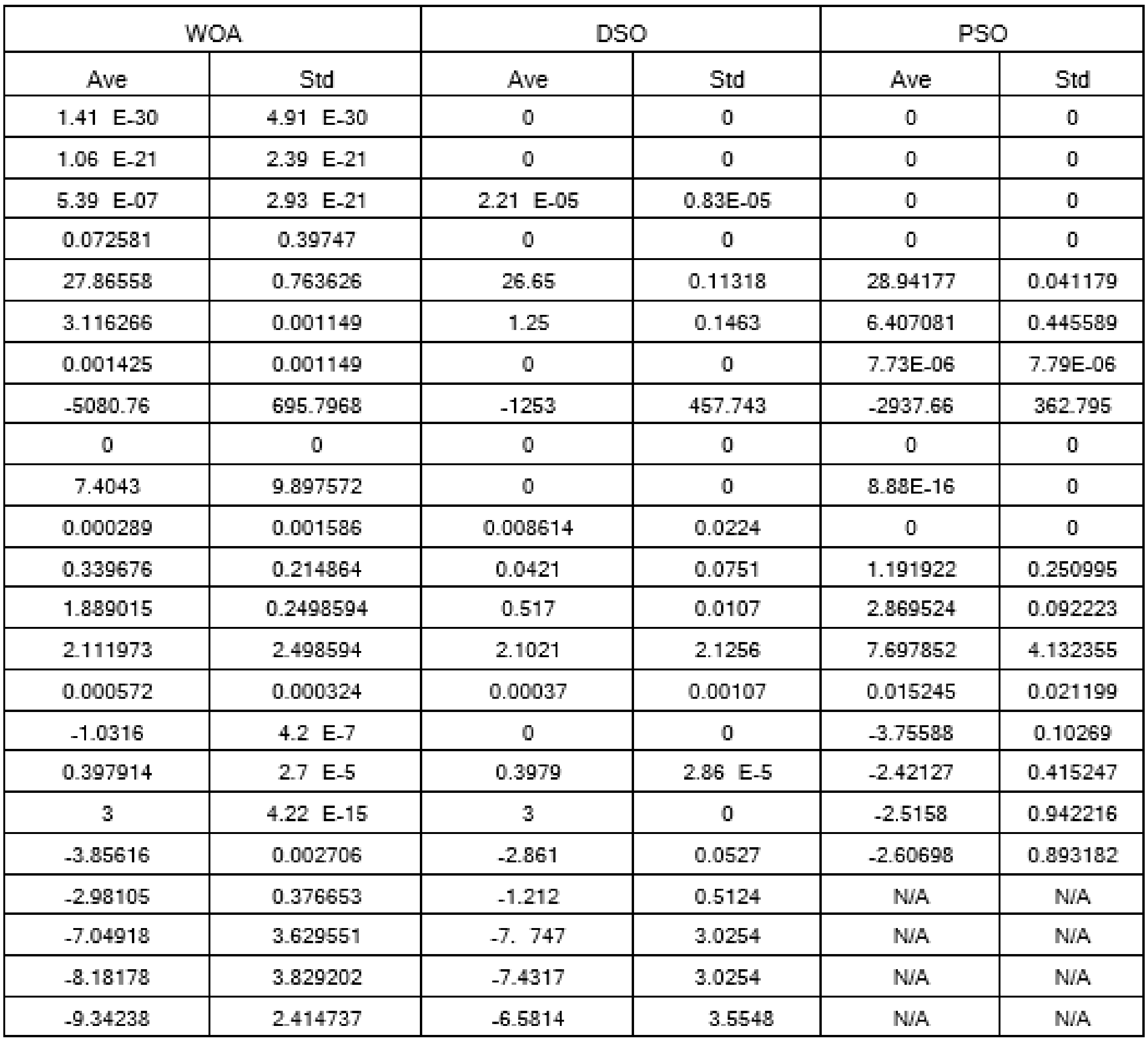
Table 5: Results of multimodal benchmark functions

GWO		PSO		GSA		DE		FEP		DSO	
Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std
6.59 E-28	6.34 E-05	1.36E-04	0.000202	2.53 E-16	9.67 E-17	8.2 E-14	5.9 E-14	5.70E-04	0.00013	2.18 E-71	2.28 E-71
7.18 E-17	0.029014	0.042144	0.045421	0.055655	0.194074	1.5 E-09	9.9 E-10	0.0081	0.00077	1.32 E-46	1.8 E-45
3.29 E-06	79.14958	70.12562	22.11924	896.5347	318.9559	6.8 E-11	7.4 E-11	0.016	0.014	4.21 E-05	1.31E-05
5.61 E-07	1.315088	1.086481	0.317039	7.35487	1.741452	0	0	0.3	0.5	0.500	1.24
26.8158	69.90499	96.71832	60.11559	67.54309	62.22534	0	0	5.06	5.87	5.65	0.28
0.816579	0.000126	0.000102	8.28 E-05	2.5 E16	1.74 E-16	0	0	0	0	0.25	0.13
0.002213	0.100286	0.122854	0.044957	0.089441	0.04339	0.00463	0.0012	0.1415	0.3522	0.02	0.01
-6123.1	-4087.44	-4841.29	1152.814	-2821.07	493.0375	-11080.1	574.7	-12554.5	52.6	-543.3	1577.743
0.310521	47.35612	46.70423	11.62938	25.96841	7.470068	69.2	38.8	0.046	0.012	0	0
1.06 E-13	0.077835	0.276015	0.50901	0.062087	0.23628	9.7 E-8	4.2 E-8	0.018	0.002	3.25 E-15	1.25 E-15
0.004485	0.006659	0.009215	0.007724	27.70154	5.040343	0	0	0.016	0.022	0.0824	0.0451
0.053438	0.020734	0.006917	0.026301	1.799617	0.95114	7.9 E-14	8 E-15	9.2 E-6	3.6 E6	0.026371	0.0264
0.654464	0.004474	0.006675	0.008907	8.899084	7.126241	5.1 E-14	4.8 E-14	0.00016	0.000073	0.241	0.097
4.042493	4.252799	3.627168	2.560828	5.859838	3.831299	0.998004	3.3 E-16	1.22	0.56	1.50	1.56
0.000337	0.000625	0.000577	0.000222	0.003673	0.001647	4.5 E-14	0.00033	0.0005	0.00032	0	0.007
-1.03163	-1.03163	-1.03163	6.25 E-16	-1.03163	4.88 E-16	-1.03163	3.1 E-13	-1.03	4.9 E-7	-.0.46	6.77 E-16
0.397889	0.397889	0.397889	0	0.397889	0	0.397889	9.9 E-9	0.398	1.5 E-7	0.39	1.56 E-5
3.000028	3	3	1.33 E-15	3	4.17 E-15	3	2 E-15	3.02	0.1	3	0
-3.86263	-3.86278	-3.86278	2.58 E-15	-3.86278	2.29 E-15	N/A	N/A	-3.86	0.000014	-3.2561	0.002
-3.28654	-3.25056	-3.26634	0.060516	-3.31778	0.023081	N/A	N/A	-3.27	0.059	-3.11	0.12
-10.1514	-9.14015	-6.8651	3.019644	-5.95512	3.737079	-10.1532	2.5E-06	-5.52	1.59	-4.60	3.02
-10.4015	-8.58441	-8.45653	3.087094	-9.68447	2.014088	-10.4029	3.9 E-7	-5.53	2.12	-2.36	3.06
-10.5343	-8.55899	-9.95291	1.782786	-10.5364	2.6 E-15	-10.5364	1.9 E-7	-6.57	3.14	-5.33	3.47

space if they undergo minor adjustments in the last iteration steps to leverage the quest space and how convergence occurs in most areas. The most promising research areas deal with improving their initial random solutions and improving their fitness values on iterations. Simulation tests are implemented using MATLAB on a computer with the following specifications: Core i5 8th Generation,1.6 GHz Processor Speed, 8GB RAM, Windows 10. In FOA, the following values are employed for the experiments: Cr= (10,0) at the beginning is 10 and decreases by 0.002 at each iteration, R1 and R1 are random numbers (0,1), CC is selected randomly at the first iteration, and then calculated using Equation (3), and B is a random number between (0,1).

To evaluate the actions of the candidate solutions, 30 search agents can solve the 2D variant of evaluation metrics. Figure 9 shows the search history of search agents. This figure demonstrates that the DSO algorithm searches around the search space's promising areas. The distribution of the sampled points around the global optima is significantly

Table 6: Results of multimodal benchmark functions



WOA		DSO		PSO	
Ave	Std	Ave	Std	Ave	Std
1.41 E-30	4.91 E-30	0	0	0	0
1.06 E-21	2.39 E-21	0	0	0	0
5.39 E-07	2.93 E-21	2.21 E-05	0.83E-05	0	0
0.072581	0.39747	0	0	0	0
27.86558	0.763626	26.65	0.11318	28.94177	0.041179
3.116266	0.001149	1.25	0.1463	6.407081	0.445589
0.001425	0.001149	0	0	7.73E-06	7.79E-06
-5080.76	695.7968	-1253	457.743	-2937.66	362.795
0	0	0	0	0	0
7.4043	9.897572	0	0	8.88E-16	0
0.000289	0.001586	0.008614	0.0224	0	0
0.339676	0.214864	0.0421	0.0751	1.191922	0.250995
1.889015	0.2498594	0.517	0.0107	2.869524	0.092223
2.111973	2.498594	2.1021	2.1256	7.697852	4.132355
0.000572	0.000324	0.00037	0.00107	0.015245	0.021199
-1.0316	4.2 E-7	0	0	-3.75588	0.10269
0.397914	2.7 E-5	0.3979	2.86 E-5	-2.42127	0.415247
3	4.22 E-15	3	0	-2.5158	0.942216
-3.85616	0.002706	-2.861	0.0527	-2.60698	0.893182
-2.98105	0.376653	-1.212	0.5124	N/A	N/A
-7.04918	3.629551	-7. 747	3.0254	N/A	N/A
-8.18178	3.829202	-7.4317	3.0254	N/A	N/A
-9.34238	2.414737	-6.5814	3.5548	N/A	N/A

high, which shows that the DSO algorithm exploits the most promising region of the search space in addition to the discovery. The optimum of the test functions is transferred to a position other than the root to provide more complex tests. According to the results in Tables 5 and 6, the DSO technique reaches the optimum value in most test functions, converging to the optimum values in most test functions. Those values are better than the values of the other chosen algorithms that use the same test functions. DSO can be similar to the best values of the algorithms selected.

The results of DSO show that this algorithm is more competitive and effective than the other algorithms listed above. In most samples, DSO is superior to all other algorithms. Those findings demonstrate the efficacy, versatility, and accuracy of the presented algorithm. The results obtained show that the DSO algorithm can find the optimal solution value in the multi-model function better than the unimodal results. This result reflects DSO's supremacy in the quest for discovery. The search agents of DSO can find promising design space regions extensively and exploit the best one. In the early stages of the optimization process, search agents alter suddenly and then correlate positively. According to Hudaib and Fakhouri [6] such actions ensure that a community algorithm ultimately converges to a point in a search space. The authors compare the results of the convergence with previous results [7, 22]. The DSO, WOA, PSO, and GSA dilemmas are correlated and plotted in Figure **??** (A, B, C) for some of the problems and in Figure 5 b (GWO, PSO, and GSA) for other problems (D, E, F). Results show that DSO is successful with other state-of-the-art meta-heuristic techniques. The DSO, PSO, WOA, and GSA dilemmas are given in Figure 8 to determine the convergence rate of the algorithms. The most significant average shows the combination of the right approach for each execution with 30 runs.

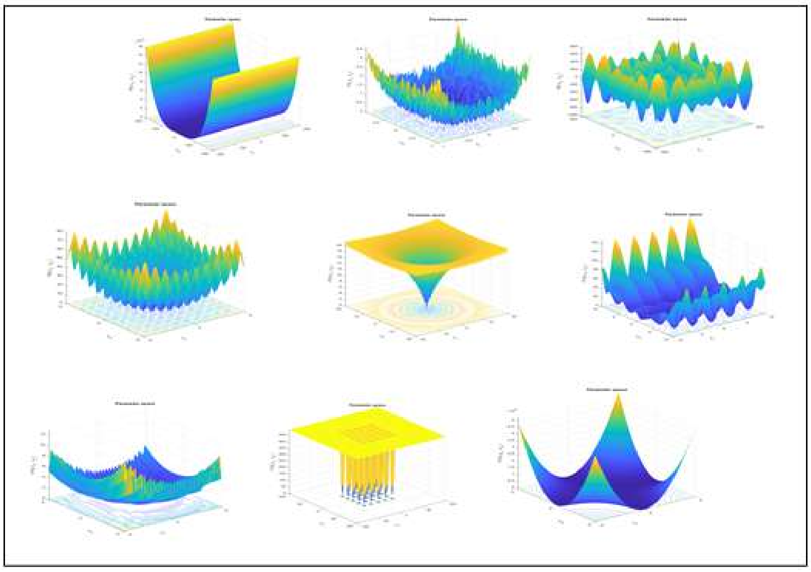


Figure 7: Multimodal Function

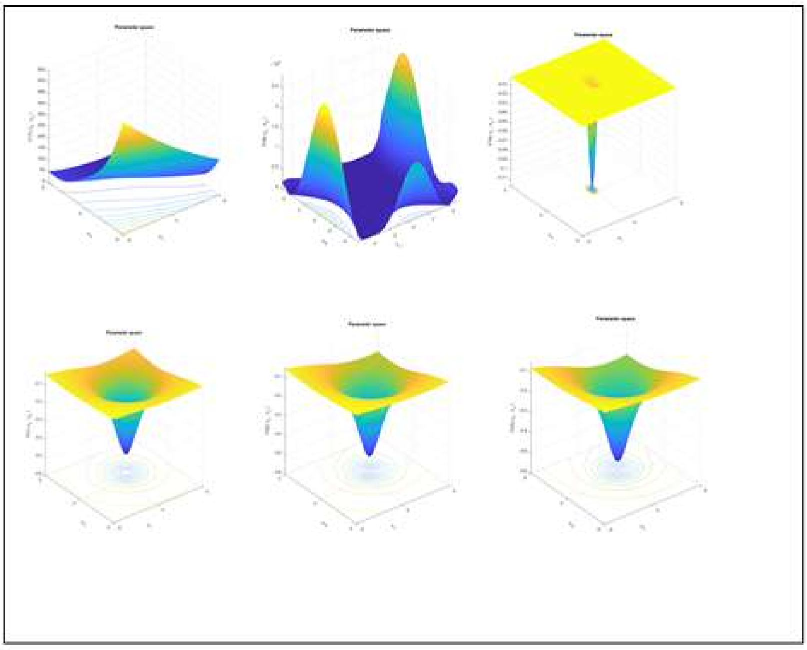


Figure 8: Fixed-dimension multimodal functions

Figure 10 shows that when optimizing the test functions, the DSO algorithm demonstrates some distinct convergence habits. First, the convergence of the DSO algorithm tends to be enhanced as iteration rises. It can be attributable to the suggested adaptive strategy for DSO that helps in the primary stages of the iterative process to search for prosperous areas of the search and accumulate quickly toward the optimum after passing almost half of the iterations. In some of the tasks, this conduct is transparent. The second action is convergence toward the optimum, as observed in F2 only in the final iterations.

Figures 6, 7, and 8 display 2D models of the performance measurement functions applied to the test system. In general, the standard functions utilized are minimization functions in addition to their ability to split within four collections: multimodal, unimodal, fixed-dimensional multimodal, multimedia, and complex functions. For the functions in Tables 1, 2, and 3, the size of each dimension is 30 (DIM = 30), and the dimension of the fixed-dimension test function is shown in Table 4. The above functions enable the researcher to estimate the potential of the meta-heuristic algorithms studied to be exploited, as shown in Figure 6. It represents the algorithm's strong aggregation

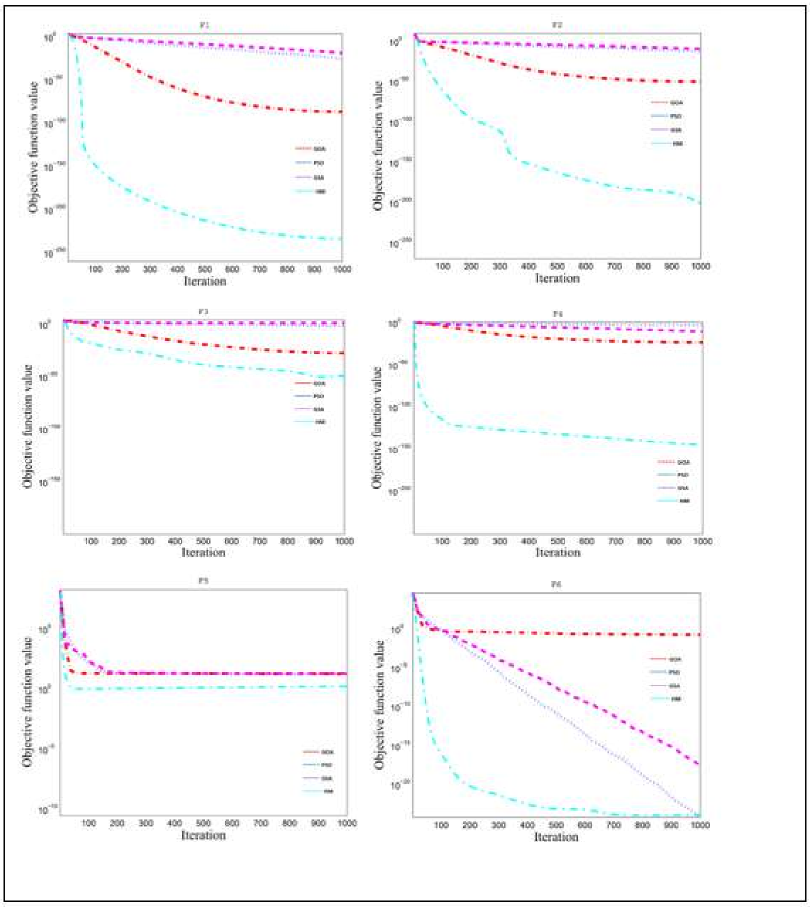


Figure 9: A - Convergence properties of GOA, PSO, GSA and DSO on test

ability and optimization power on the unimodal evaluation metrics primarily. Certain test functions are intended to concentrate on the algorithm after the optimization process to achieve the global optimum. All of the functions are described by many local best solutions for multimodal test functions, allowing the algorithm, as shown in Figure 7, to slip toward better convergence points. It can also be used to test the algorithm's local search and global search skills. The results in Tables 5 and 6 of F8–F23 (multimedia and fixed multimedia functions) indicate that DSO also has good exploration ability. Different fixed-dimensional functions are reviewed to check the algorithm's computational efficiency, stability, and refinement consistency to thoroughly evaluate the performance of DSO.

The performance ranges of a unimodal fitness function are shown in Figure 9 A to assess the computing efficiency of the four methods theoretically. DSO has a more advantageous position over the F1, F3, and F4 test functions than PSO, GSA, and WOA. DSO should have a higher fitness value in the beginning stages. It takes 500 repetitions to achieve a decent F5 test function quality. In addition, the convergence graphs of the F8 and F12 test functions show that the suggested DSO not only decreases premature convergence but maintains a favorable reputation among others. The author believes that DSO has the best advantage and performance when handling unimodal test operations. The accuracy of DSO is helpful for comparison techniques in every respect. More value is found with WOA and PSO, whereas worthless results are obtained with GSA. The DSO results are similar on F15 and F16, and the other methods are easy to tweak locally. Each algorithm explores many local optima as the test functions in their experiments. The GSA's performance is better than the other algorithms while attempting to solve F19. After approximately 500 iterations, DSO rapidly converges to a near-optimal solution. Therefore, the convergence rate of DSO is very impressive.

Furthermore, four tests may be conducted, including F17 and F19, which have a higher convergence rate than GSA, GWO, and PSO because they begin to settle to a fixed value after the first step. Simulation results reveal a significant capacity for DSO by improving the unimodal, multimodal, and fixed dimension test functions. In addition DSO has a rivalry with other contemporary algorithms. In other words, DSO has a balance between discovering globally and exploiting locally. Results show that DSO provides a convenient and effective solution similar to the

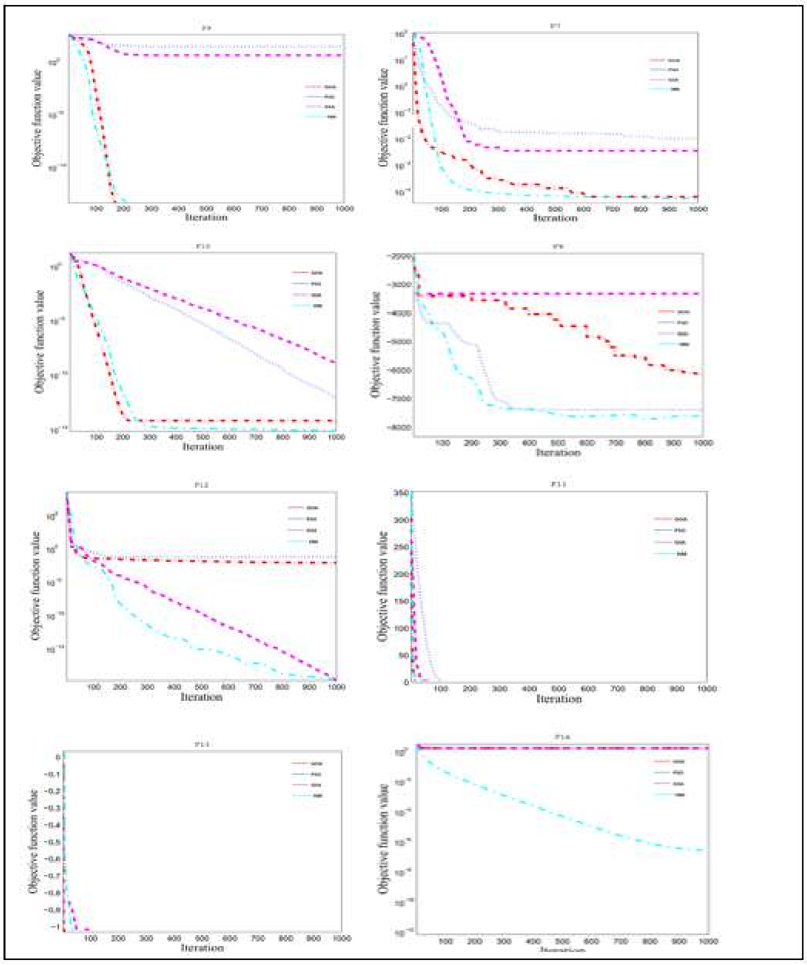


Figure9: B- Convergence properties of GOA, PSO, GSA and DSO on test

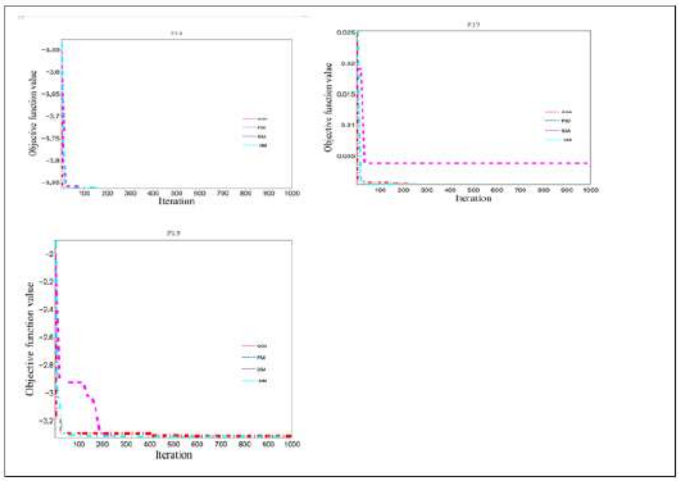


Figure9: C- Convergence properties of GOA, PSO, GSA and DSO on test

comparative algorithms of the F20, F21, F22, and F23 functions, which tackle complex matters from the multimodal extended hybrid composition functions.

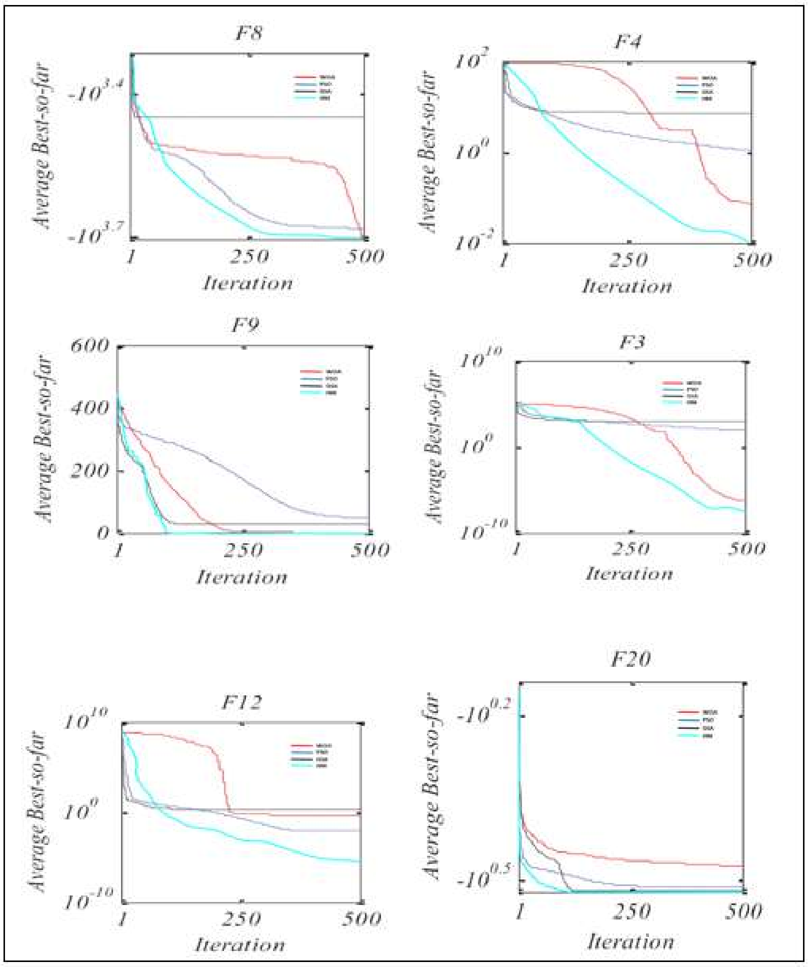


Figure 10: A- Convergence properties of WOA, PSO, GSA and DSO on test

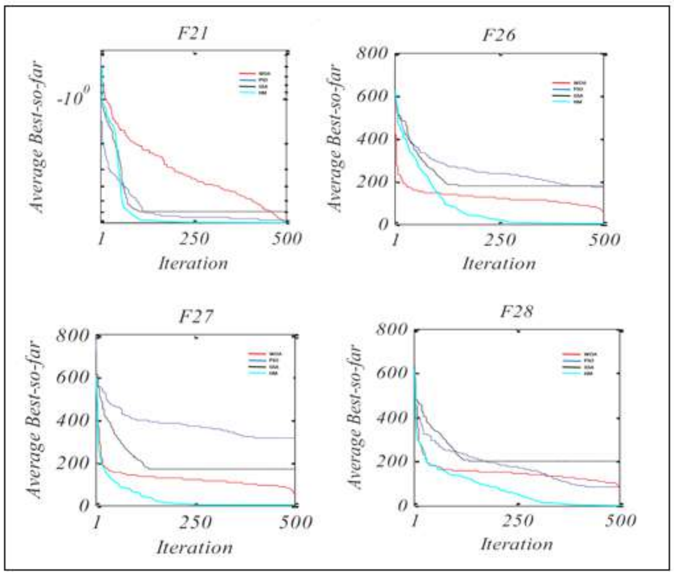


Figure10: B- Convergence properties of WOA, PSO, GSA and DSO on test

6 Conclusion

A new algorithm is implemented to strengthen the algorithm based on swarm optimization on the social behavior of hunting dolphins. DSO has been proposed as an alternative technique to solve optimization problems. Solutions have been critical in the proposed DSO algorithm to improve their sites based on the location of the optimal solution. Updating the site allows the solutions to travel to or to the destination point to ensure that the search space is used and explored. A total of 23 test functions are used to measure the intensity and efficiency of DSO in terms of exploration and exploitation. Results show that DSO could suppress FEP, DE, GSA, PSO, and GWO. In using the DSO algorithm, results obtained from unimodal test functionality exhibit dominance. Later, the ability to discover DSO with outcomes for multimodal functions is shown.

References

[1] K. Ahmed, B. Al-Khateeb and M. Mahmood, *Application of chaos discrete particle swarm optimization algorithm on pavement maintenance scheduling problem*, Cluster Comput. **22** (2019), no. 2, 4647–4657.

[2] E. Atashpaz-Gargari and C. Lucas, *Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition*, IEEE Cong. Evol. Comput., 2007, pp. 4661–4667.

[3] C.J. Barnard and R.M. Sibly, *Producers and scroungers: a general model and its application to captive flocks of house sparrows*, Animal behaviour. 29(2) (1981) 543–550.

[4] Z. Barta, A. Liker and F. Mónus, *The effects of predation risk on the use of social foraging tactics*, Animal Behav. **67** (2004), no. 2, 301–308.

[5] I. Coolen, L.A. Giraldeau and M. Lavoie, *Head position as an indicator of producer and scrounger tactics in a ground-feeding bird*, Animal Behav. **61** (2001), no. 5, 895–903.

[6] A.A. Hudaib and H.N. Fakhouri, *Supernova optimizer: a novel natural inspired meta-heuristic*, Modern Appl. Sci. **12** (2018), no. 1, 32–50.

[7] S.H. Ismael, S.W. Kareem and F.H. Almukhtar, *Medical Image Classification Using Different Machine Learning Algorithms*, AL-Rafidain J. Comput. Sci. Math. **14** (2020), no. 1, 135–147.

[8] D. Karaboga, *An idea based on honey bee swarm for numerical optimization*, Technical Report-TR06, Erciyes Univ. **200** (2005), 1–10.

[9] D. Karaboga and B. Basturk, *A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm*, J. Glob. Optim. **39** (2007), no. 3, 459–471.

[10] S.W. Kareem, *Novel swarm intelligence algorithms for structure learning of bayesian networks and a comparative evaluation*, Learning Structure of Bayesian Network, 2020.

[11] S.W. Kareem and M.C. Okur, *Pigeon inspired optimization of bayesian network structure learning and a comparative evaluation*, J. Cognitive Sci. **20** (2019), no. 4, 535–552.

[12] S.W. Kareem and M.C. Okur, *Evaluation of Bayesian Network Structure Learning Using Elephant Swarm Water Search Algorithm*, Handbook of Research on Advancements of Swarm Intelligence Algorithms for Solving Real-World Problems, IGI Global, 2020.

[13] S.W. Kareem and M.C. Okur, *Structure learning of Bayesian networks using elephant swarm water search algorithm*, Int. J. Swarm Intell. Res. **11** (2020), no. 2, 19–30.

[14] S. Kareem and M.C. Okur, *Bayesian Network Structure Learning Using Hybrid Bee Optimization and Greedy Search*, Adana, Turkey, Çukurova University. 2018.

[15] A. Kaveh and M. Khayatazad, *A new meta-heuristic method: ray optimization*, Comput. Struc. **112** (2012), 283–294.

[16] J. Kennedy and R. Eberhart, *Particle swarm optimization*, Proc. ICNN'95-Int. Conf. Neural Networks **4** (1995), 1942–1948.

[17] M.A. Koops and L.A. Giraldeau, *Producer–scrounger foraging games in starlings: a test of rate-maximizing and risk-sensitive models*, Animal Behav. **51** (1996), no. 4, 773–783.

[18] A.S. Mohammed, S.W. Kareem, A.K. Al Azzawi and M. Sivaram, *Time series prediction using SRE-NAR and SRE-ADALINE*, J. Adv. Res. Dyn. Control Syst. **10** (2018), 1716–1726.

[19] Y. Shi, *Brain storm optimization algorithm*, Int. Conf. Swarm Intell. Springer, Berlin, Heidelberg, 2011, pp. 303–309.

[20] Y. Shi, *An optimization algorithm based on brainstorming process*, Emerging Research on Swarm Intelligence and Algorithm Optimization, IGI Global, 2015 1–35.

[21] T.Q. Wu, M. Yao and J.H. Yang, *Dolphin swarm algorithm*, Front. Inf. Technol. Electron. Engin. **17** (2016), no. 8, 717–729.

[22] Y. Zhang, S. Wang and G. Ji, *A comprehensive survey on particle swarm optimization algorithm and its applications*, Math. Prob. Engin. **2015** (2015).