

Research Article

Dingo Optimizer: A Nature-Inspired Metaheuristic Approach for Engineering Problems

Amit Kumar Bairwa ¹, Sandeep Joshi ¹ and Dilbag Singh ²

¹Manipal University Jaipur, Rajasthan, Jaipur, India

²Bennett University, Uttar Pradesh, Noida, India

Correspondence should be addressed to Sandeep Joshi; sjoshinew@yahoo.com

Received 23 July 2020; Revised 8 October 2020; Accepted 16 December 2020; Published 10 June 2021

Academic Editor: Yann Favennec

Copyright © 2021 Amit Kumar Bairwa et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Optimization is a buzzword, whenever researchers think of engineering problems. This paper presents a new metaheuristic named dingo optimizer (DOX) which is motivated by the behavior of dingo (*Canis familiaris dingo*). The overall concept is to develop this method involving the collaborative and social behavior of dingoes. The developed algorithm is based on the hunting behavior of dingoes that includes exploration, encircling, and exploitation. All the above prey hunting steps are modeled mathematically and are implemented in the simulator to test the performance of the proposed algorithm. Comparative analyses are drawn among the proposed approach and grey wolf optimizer (GWO) and particle swarm optimizer (PSO). Some of the well-known test functions are used for the comparative study of this work. The results reveal that the dingo optimizer performed significantly better than other nature-inspired algorithms.

1. Introduction

The challenges of the modern world are composed of various goals that must be optimized at the same time. Optimization is a process that seeks one or more solutions to the problem that leads to the extreme values of one or more objective [1]. The optimization can, therefore, be done based on single or multiple objective functions [2–4].

Keeping this in the mind, there is a requirement of new metaheuristic-based solution to reduce the burden of any of the model designing. The objective of this paper is to develop a nature-based algorithm called dingo optimizer, which can be abbreviated as DOX. It is based on dingo's social hierarchy and prey hunting behavior.

Metaheuristic algorithms are remarkably common due to its nature of flexibility, simplicity, less mathematical complexity, and avoidance of local optima. If we talk about flexibility, then it means we can use such algorithms in a wide variety of engineering problems. Such algorithms provide satisfactory results for many of the complex problems [5]. It is simple because it is inspired by nature like

animal behavior to accomplish a particular task, physical phenomena, and other evolutionary behavior.

One of the main reasons to use the metaheuristics in real-life problems is that almost all the optimization solutions start with the random processes, and for such solutions, there is no need to find out the optimum. Metaheuristic algorithms are very powerful in terms of finding local optima compared with the traditional optimization algorithms. Finding the real search space in the real world problem is very much complicated because of finding with lots of local optima in the search. That is the reason metaheuristic algorithms are most suitable to find out such challenging issues.

There are so many metaheuristic algorithms proposed every year, and they show the promising result with respect to the engineering problem. However, day by day the nature and complexity of new applications are introducing with new challenges. And, it might not be possible to solve the particular problem with the guarantee. This motivates us to develop a new metaheuristics algorithm as dingo optimizer (DOX). Also, the method which is mathematically modeled

and inspired by the social hierarchy of dingoes is also motivated to solve a real-time engineering problem.

This paper is arranged as follows. Section 2 provides a literature survey that explains the brief principles of DOX focused on dog pack hunting. The proposed multiobjective DOX is presented in Section 3. In section 4, the performance of DOX is tested with various benchmark functions and experimental results are compared with other state-of-the-art algorithms. In Section 5, the conclusion and future research directions discussed.

2. Literature Review

In the past few years, the problems related to real-life have increased, and it is motivating researchers to develop a better metaheuristic technique with the concept of randomization and local search. Such approaches have been used to determine the best solutions to real-life engineering problems that are feasible. These methods are more widely accepted due to their difficulty and reliability relative to other existing methods.

Metaheuristic algorithms can be categorized into evolutionary-based [13], physical algorithms [54], bio-inspired algorithms, swarm-based [25], and others methods. The evolutionary algorithm gives approximately close solutions to all types of optimization models since these approaches are not dependent on basic fitness and assumptions [55]. Bio-inspired algorithms are also popular to solve various complicated problems, which are motivated by biological evolution such as selection, replication, mutation, and recombination. Physical algorithms are motivated by evolutionary algorithms by the principle of natural selection in which a species attempts to live in different environments based on the fitness test. The most common parameters such as inertia force, electromagnetic force, and gravitational force help the algorithm to search the agent's coordinate and search around the space. Swarm-based algorithms are inspired by the self-organized nature of the social creatures, which shows the collective behavior of decentralization. Corporate wisdom is influenced by the contact of swarms with each other and their surroundings. Some of the popular swarm intelligence technique is quite similar to the nature-inspired algorithm. Generally, swarm-based algorithms are more popular as they have fewer operators (i.e., discovery, encircling, and exploitation). Table 1 lists all these algorithms which are divided into single objective and multi-objectives depending on the number of objective functions.

3. Dingo Optimizer (DOX)

3.1. Motivation. Nature is always the most powerful teacher from the beginning. Ever species surviving on the Earth have its way of unique mechanism for survival. Social relationships are one of them, which is dynamic. Based on the general study of the social behavior of the animal, it can be segmented into some of the categories. The first category is depending on the environmental factors, i.e., nearby resource availability and challenges created by other species. Another category depends on individual behavior or quality.

Keeping this in the mind, dingo is motivation to our work which follows strictly the social relationship. Dingo is the dog's sort. The scientific name of dingo is *Canis lupus* (wolf) *dingo*, changed recently from *Canis familiaris* (dog). Dingoes are complicated, intelligent, and highly social animals. Dingoes are skillful hunters living in a pack of the average size 12–15.

Social hierarchy is highly structured, alpha is on the top of the hierarchy, and they might be male or female. They can be identified based on the responsibility like making decisions, sleeping places, and hunting. The most dominant and strongest member in the pack is called alpha, considered as the leader of the pack of dingoes. It reflects that the discipline and organization are more important than power. The decision taken by the alpha is dictated to the pack. In general, all the members of a pack acknowledge the alpha by holding their tails down.

Beta dingoes are at the second level in the hierarchy, which played a role of intermediate between alpha and another pack for the related tasks. It plays the important role as an adviser of alpha and maintains the discipline for the whole pack. The beta confirms the orders of the alpha in the group and communicates to the alpha. The beta dingo is second in the hierarchy after the alpha. If alpha does not survive due to any of the reasons, all the commands will be handed over by the beta to control the other lower-level dingoes.

If a dingo does not belong to an alpha or beta, they are considered as subordinate. These subordinates follow alphas and betas. Scouts shall be liable for observing the area of the territories and shall alert the group in case of any threat circumstance. Hunters shall support the alphas and betas to catch the prey and provide food for the group.

Based on the studies, dingoes have an accurate sense of communication. They communicate with each other through sensing different sound intensities in the air. In DOX, dingo creates sound feedback in such a way that dingoes exchange their knowledge with others to create common community details. The amplitude of the vibration is modified by the strength of the person as the dingo enters a new location from the previous one.

Group hunting is an interesting social behavior of dingoes, which makes its more extension to the social behavior of dingoes. Hunting strategy is categorized in their phases as follows:

- Chasing and approaching
- Encircling and harassing
- Attack

The above steps are properly shown in Figure 1. Also, hunting behavior and the social arrangements of dingo are modeled mathematically, to develop DOX to perform nature-inspired optimization.

Exploration and exploitation are the two main components of DOX. In the exploration part, the algorithm reaches several expected solutions in the search space but exploitation allows searching for optimal solutions within the given space. To find out the best solution for any real-life problem, both the components are required with fine-

TABLE 1: Classification of different metaheuristic approaches.

| Types of algorithms | Single objective | Multiobjective |
|--|--|-----------------------------|
| Evolutionary algorithms (EA) [6] | Genetic algorithm (GA) [7] | NSGA II [2, 8] |
| | Genetic programming (GP) [9] | Multiobjective GP [10] |
| | Differential evolution (DE) [11] | Multiobjective DE [12] |
| | Evolutionary strategy (ES) [13] | Multiobjective ES [14] |
| Bio-inspired algorithms (BIA) | Artificial immune system (AIS) algorithm [15] | Multiobjective AIS [16] |
| | Bacterial foraging algorithm (BFA) [17] | Multiobjective BFA [18] |
| | Dendritic cell algorithm (DCA) [19] | |
| | Krill herd algorithm (KHA) [20] | |
| Physical algorithms | Simulated annealing (SA) [21] | Multiobjective SA |
| | Memetic algorithm (MA) [22] | Multiobjective MA |
| | Shuffled frog-leaping algorithm [23] | Multiobjective SFA |
| Swarm intelligence (SI) | Ant colony optimization (ACO) [24] | Multiobjective ACO |
| | Particle swarm optimization (PSO) [25] | Multiobjective PSO [1] |
| | Artificial bee colony (ABC) [26] | Multiobjective ABC |
| | Fish swarm algorithm (FSA) [27] | Multiobjective FSA |
| | Grey wolf optimizer (GWO) [28] | |
| | Dragonfly algorithm (DA) [29] | |
| Other nature-inspired algorithms | Firefly algorithm [30] | Multiobjective firefly [31] |
| | Whale optimization algorithm (WOA) [32] | |
| | Gravitational search algorithm [33] | Multiobjective GSA |
| | Bat algorithm (BA) [34] | Multiobjective Bat |
| | Cuckoo search algorithm (CSA) [35] | Multiobjective cuckoo |
| Human behavior-inspired algorithms | Cat swarm algorithm (CSA) [36] | Multiobjective CSO |
| | Harmony search (HS) [37] | Multiobjective HS [38] |
| | Tabu search (TS) [39] | Multiobjective TS [40] |
| | Parameter adaptive harmony search (PAHS) [41] | Multiobjective PAHS [41] |
| | Group search optimizer (GSO) [42] | Multiobjective GSO |
| | Exchange market algorithm (EMA) [43] | Multiobjective EMA |
| | Imperialist competitive algorithm (ICA) [44] | Multiobjective ICA |
| | Soccer league competition algorithm (SLCA) [45] | Multiobjective SLCA |
| | League championship algorithm (LCA) [46] | Multiobjective LCA |
| | Social-based algorithm (SBA) [47] | Multiobjective SBA |
| | Firework algorithm (FA) [48] | Multiobjective FA |
| | Colliding bodies optimization (CBO) [49] | Multiobjective CBO |
| | Soccer league competition algorithm (SLCA) [45] | Multiobjective SLCA |
| | Interior search algorithm (ISA) [50] | Multiobjective ISA |
| | Artificial ecosystem-based optimization (AEO) [51] | |
| Spiral optimization algorithm (SOA) [52] | | |
| Adolescent identity search algorithm (AISA) [53] | | |

tuning. However, it is a challenge to make a balance between the components of the proposed algorithm due to its stochastic nature. To solve a real-life engineering problem, this inspiring fact motivates a hybridize metaheuristics algorithm design.

3.2. Mathematical Models. The representation of the searching, encircling, and attacking prey is designed mathematically to perform the dingo optimization in this section.

3.2.1. Encircling. Dingoes are enough capable to find the location of the prey. After tracing the location, the pack followed by alpha encircles the prey. To model dingo's social hierarchy, it is assumed that the existing best agent approach is the goal or aim prey, which is similar to the optimal since the quest area is not known a priori. In the meantime, other quest agencies are still seeking to refresh their strategies on the next possible approach. This behavior of the dingoes is

modeled by the following mathematical equations (1)–(5). Also, a detailed description of the nomenclatures used in the equation is provided in Table 2

$$\vec{D}_d = |\vec{A} \cdot \vec{P}_p(x) - \vec{P}(i)|, \quad (1)$$

$$\vec{P}(i+1) = \vec{P}_p(i) - \vec{B} \cdot \vec{D}(d), \quad (2)$$

$$\vec{A} = 2 \cdot \vec{a}_1, \quad (3)$$

$$\vec{B} = 2\vec{b} \cdot \vec{a}_2 - \vec{b}, \quad (4)$$

$$\vec{b} = 3 - \left(I * \left(\frac{3}{I_{\max}} \right) \right). \quad (5)$$

Positions of the neighborhood dingoes can be illustrated using Figure 2, which is represented using a two-dimensional

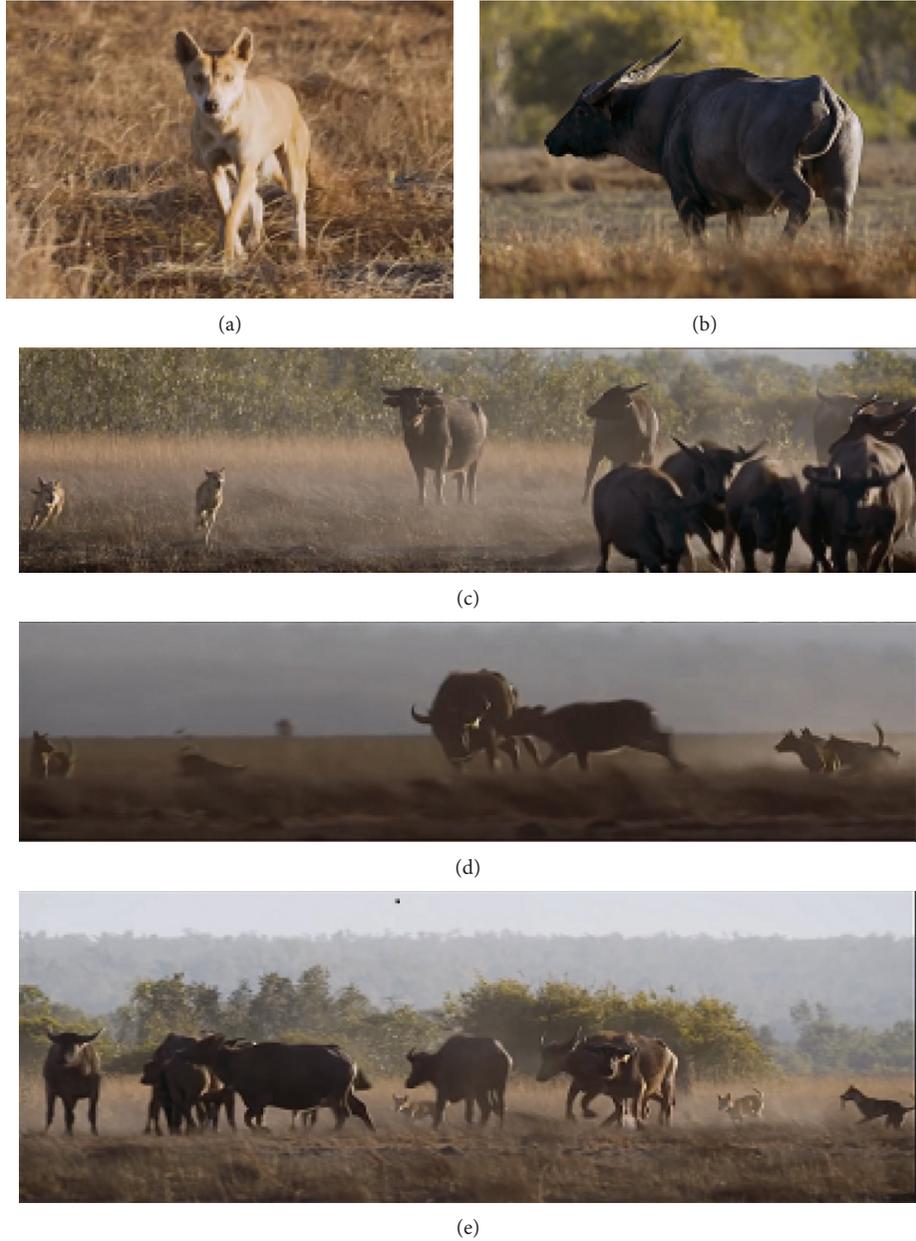


FIGURE 1: Hunting behavior of dingoes: (a) dingo; (b) prey; (c) chasing and approaching; (b–d) harassing and encircling; (e) attack.

TABLE 2: Nomenclature of equations (1)–(5).

| Elements | Description |
|------------------|---|
| \overline{D}^d | Distance between the dingo and prey |
| \overline{P}^p | Position vector (prey) |
| \overline{P}^d | Position vector (dingo) |
| \overline{A} | Coefficient vector |
| \overline{B} | Coefficient vector |
| \overline{a}_1 | Random vector in [0,1] |
| \overline{a}_2 | Random vector in [0,1] |
| b | Linearly decrement from 3 to 0 at every iteration |
| $\ $ | Absolute value and multiplication with vectors |
| I | 1, 2, 3, ..., J_{\max} |
| J_{\max} | Maximum no. of iteration |

position vector. According to the position of the prey (P^* , Q^*), a dingo can update its position at the position of (P , Q). All the possible locations are marked in the diagram around the best agent, concerning the current location by changing the value of \overline{A} and \overline{B} vectors. For example, by setting $\overline{A} = (1, 0)$ and $\overline{B} = (1, 1)$, dingo can be reached at $(P^* - P, Q^*)$. It can also be represented using 3-dimensional space as in Figure 3. It is clearly illustrated how random vectors a_1 and a_2 enable dingoes to enter any place between the points. Equations (1) and (2) help dingoes to change their locations inside the quest area around the prey in any random location. To reach a search space with N dimensions, the same equations can be used and the dingo will move in hypercubes around the best result got so far.

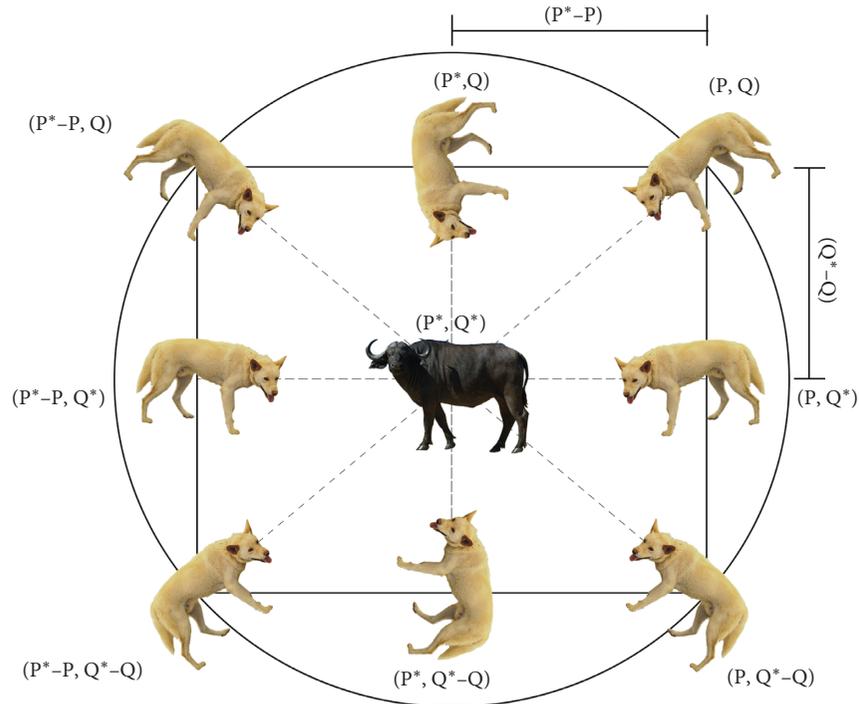


FIGURE 2: 2D position vectors of dingoes.

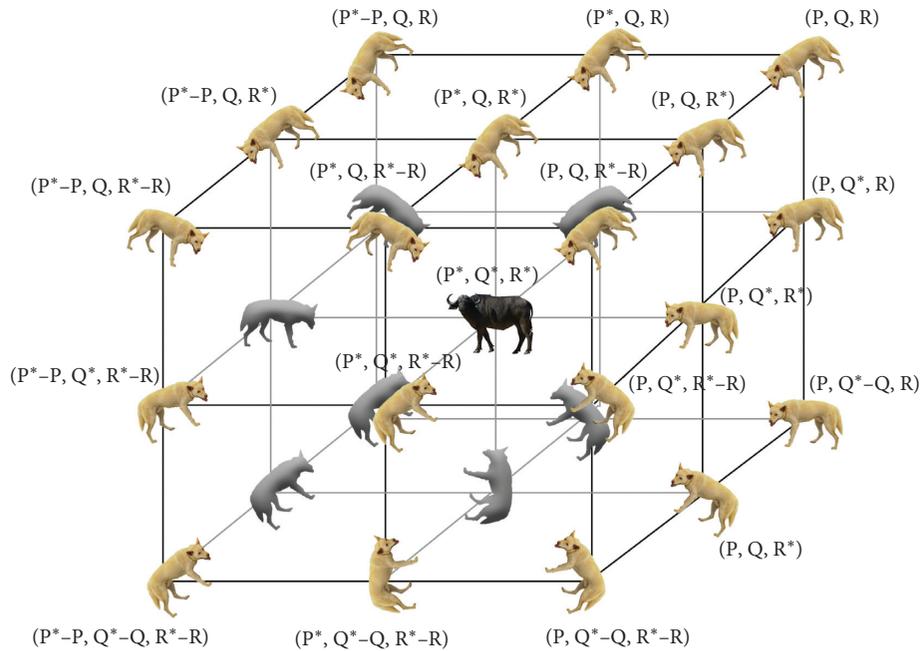


FIGURE 3: 3D position vectors of dingoes.

3.2.2. *Hunting*. However, in the search space according to the concept, agents do not normally have a calculation of the position of the prey (optimum). Designing the dingoes hunting plan mathematically, we assume that all the pack members including alpha, beta, and others have good knowledge about the potential location of prey. The alpha dingo always commands the hunting. However, sometimes

beta and other dingoes might also participate in hunting. Hence, we consider the first two best values achieved so far. As per the location of the best search agent, other dingoes also need to update their position. According to the discussion, equations (6)–(14) are modeled in this concern. Also, a detailed description of the nomenclatures used in the equation is provided in Table 3.

$$\vec{D}_\alpha = |\vec{A}_1 \cdot \vec{P}_\alpha - \vec{P}|, \quad (6)$$

$$\vec{D}_\beta = |\vec{A}_2 \cdot \vec{P}_\beta - \vec{P}|, \quad (7)$$

$$\vec{D}_o = |\vec{A}_3 \cdot \vec{P}_o - \vec{P}|, \quad (8)$$

$$\vec{P}_1 = |\vec{P}_\alpha - \vec{B} \cdot \vec{D}_\alpha|, \quad (9)$$

$$\vec{P}_2 = |\vec{P}_\beta - \vec{B} \cdot \vec{D}_\beta|, \quad (10)$$

$$\vec{P}_3 = |\vec{P}_o - \vec{B} \cdot \vec{D}_o|. \quad (11)$$

To calculate the intensity of each dingo, following equations are being used:

$$\vec{T}_\alpha = \log\left(\frac{1}{F_\alpha - (1E - 100)} + 1\right), \quad (12)$$

$$\vec{T}_\beta = \log\left(\frac{1}{F_\beta - (1E - 100)} + 1\right), \quad (13)$$

$$\vec{T}_o = \log\left(\frac{1}{F_o - (1E - 100)} + 1\right). \quad (14)$$

The position update in the 2D search space is described in Figure 4. In this, we can easily visualize the position updated of alpha, beta, and other dingoes. It can also be understood that dingoes (alpha, beta, and others) update their positions randomly and calculate the position of the prey in the search space.

3.2.3. Attacking Prey. If there is no position update, it means dingo finished the hunt by attacking the prey. To mathematically formulate the strategy, the value of b is decreased linearly. Point to be noted is that the alteration range of \vec{D}_α is also decreased by \vec{b} . This may also be known as $\vec{D}\alpha$ which is a random value in the $[-3b, 3b]$ interval where b is reduced from 3 to 0 during iterations. When random values of $\vec{D}\alpha$ are in $[1, 1]$, a search agent's next position may be in any position between its current and the prey's position.

The proposed encircling method does indeed reveal exploration to some extent; however, to accentuate exploration, DOX requires more operators. Figure 4 is the illustration that shows that <1 drives the dingo to strike the prey. The DOX assists its quest agents in changing their location based on the positioning of α , β , others, and the targeted prey. Even, with these operators, the DOX can inactivate local solutions.

3.2.4. Searching. Dingoes hunt for the prey mostly according to the pack's location. They always travel forward to hunt for and strike predators. Accordingly, \vec{B} is used for random values where, if the value is less than -1 , it means prey is moving away from the search agent, but if the value is greater than 1 , it means pack approaches the prey. This

TABLE 3: Description of equations (1)–(4).

| Elements and description | |
|--------------------------|---|
| \vec{D}_d | Distance between the dingo and prey |
| \vec{P}^p | Positioning of a prey vector |
| \vec{P} | Positioning of a dingo vector |
| \vec{A} | Coefficient vector |
| \vec{B} | Coefficient vector |
| \vec{a}_1 | Random vector in $[0, 1]$ |
| \vec{a}_2 | Random vector in $[0, 1]$ |
| \vec{b} | Linear decrease over the course of iterations |
| $\ $ | Absolute value |
| I | $1, 2, 3, \dots, I_{\max}$ |
| I_{\max} | Maximum no. of iteration |
| F_α | Fitness value of alpha (α) dingo |
| F_β | Fitness value of alpha (β) dingo |
| F_o | Fitness value of other dingoes |

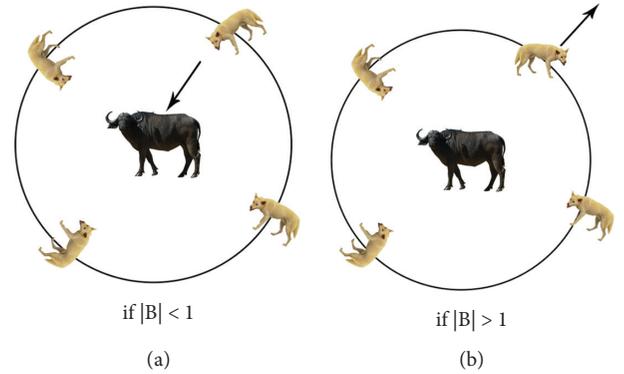


FIGURE 4: 2D position vectors of dingoes.

intervention helps the DOX to scan the targets globally. To find out which prey is better suited, Figure 4 reflects that 1 lets dingoes avoid the predators. Another component of DOX that makes exploration likely is \vec{A} . In equation (3), the vector \vec{A} can produce any random number between $[0, 3]$ for arbitrary prey weights. DOX represents a stochastic function, regarded as vector ≤ 1 precedes than ≥ 1 to explain the impact of the gap formulated in equation (1).

This would be good for searching and avoidance of nearby optima. Depending on a dingo's location, it will arbitrarily agree on the prey's value and make it necessary to meet dingo rigidly or beyond. Intentionally, we used \vec{A} to provide stochastic exploration values from the initial to the final iterations. This method is effective in protecting the solution from local optima. Eventually, the DOX terminates itself whenever it meets the termination criteria.

3.3. Optimization Algorithm. The DOX pseudo code demonstrates how it can solve optimization problems, and several points can be mentioned in Algorithm 1. Here, stopping criteria belong to the maximum number of iterations. The dingo optimization algorithm process is discussed in the following steps.

Input: The population of dingoes D_n ($n = 1, 2, \dots, n$)
Output: The best dingo. (Here, the best values is minimum)
(1) Generate initial search agents \vec{D}_{in}
(2) Initialize the value of b , \vec{A} and \vec{B} .
(3) **While** Termination condition not reached do
(4) Evaluate each dingo's fitness and intensity cost.
(5) D_α = Dingo with the best search
(6) D_β = Dingo with the second best search
(7) D_o = Dingoes search results afterwards
(8) Iteration1
(9) repeat
(10) **for** $i = 1: D_{in}$ do
(11) Renew the latest search agent status.
(12) **endfor**
(13) Estimate the fitness and intensity cost of dingoes.
(14) Record the value of $S_\alpha, S_\beta, S_\delta$
(15) Record the value of b, \vec{A} , and \vec{B} .
(16) Iteration = Iteration +1
(17) check if, Iteration \geq Stopping criteria
(18) output
(19) **endwhile**

ALGORITHM 1: Dingo optimizer (DOX).

4. Results and Discussion

4.1. Experimental Setup. The overall simulation is done in MATLAB, taking into account the various parameters which will be explained in the setup of the simulation. The proposed DOX is implemented in Windows 10 with memory 8 GB RAM and processor Intel CPU 2.50 GHz. To generate the solutions for each predefined benchmark function, the DOX uses 25 individual runs and each run applies 500 times of iterations.

4.2. Results. The DOX is the algorithm that has been tested on 23 well-known test functions [56]. These test functions are the classical functions used by various research groups. The results of the model being suggested using the dingo algorithm are shown as follows. Such testing functions were chosen to align our experiments with the current meta-heuristics despite the convenience. These benchmark functions are defined in appendix Tables 4–6 where Dim indicates the function size, Range is the search space boundary, and the maximum is f_{max} . Figure 5 represents comparison of convergence curves of DOX obtained in some of the benchmark problems. The benchmark functions are typical functions of minimization and can be segmented into four different categories.

5. DOX for Engineering Problems

Here, DOX was tested on a small engineering design problem called a pressure vessel. Such kind of problem is having different design constraints to handle the optimization.

5.1. Pressure Vessel Design. This is the problem which is used by many researchers to validate the solution that was proposed by Kannan and Kramer [57] to minimize the total

cost, including cost of material, forming, and welding of cylindrical vessel which are capped at both ends by hemispherical heads.

- (1) p1: thickness of the shell
- (2) p2: thickness of the head
- (3) p3: inner radius
- (4) p4: length of the cylindrical section without considering the head

The mathematical formulation of this problem is formulated as follows.

Consider $p = [p_1 p_2 p_3 p_4]$.

Minimize the following function:

$$f(p) = 0.6224p_1p_3p_4 + 1.7781p_2p_3^2 + 3.1661p_1^2p_4 + 19.84p_1^2p_3, \quad (15)$$

Subject to

$$\begin{aligned} f_1(z) &= -f_1 + 0.0193p_3 \leq 0, \\ f_2(z) &= -f_3 + 0.00954f_3 \leq 0, \\ f_3(z) &= -\Pi f_3^2 f_4 - \frac{4}{3}\Pi f_3^3 + 1296000 \leq 0, \\ f_4(z) &= f_4 - 240 \leq 0. \end{aligned} \quad (16)$$

Variable range is as follows:

$$\begin{aligned} 0 &\leq f_1 \leq 99, \\ 0 &\leq f_2 \leq 99, \\ 0 &\leq f_3 \leq 200, \\ 0 &\leq f_4 \leq 200. \end{aligned} \quad (17)$$

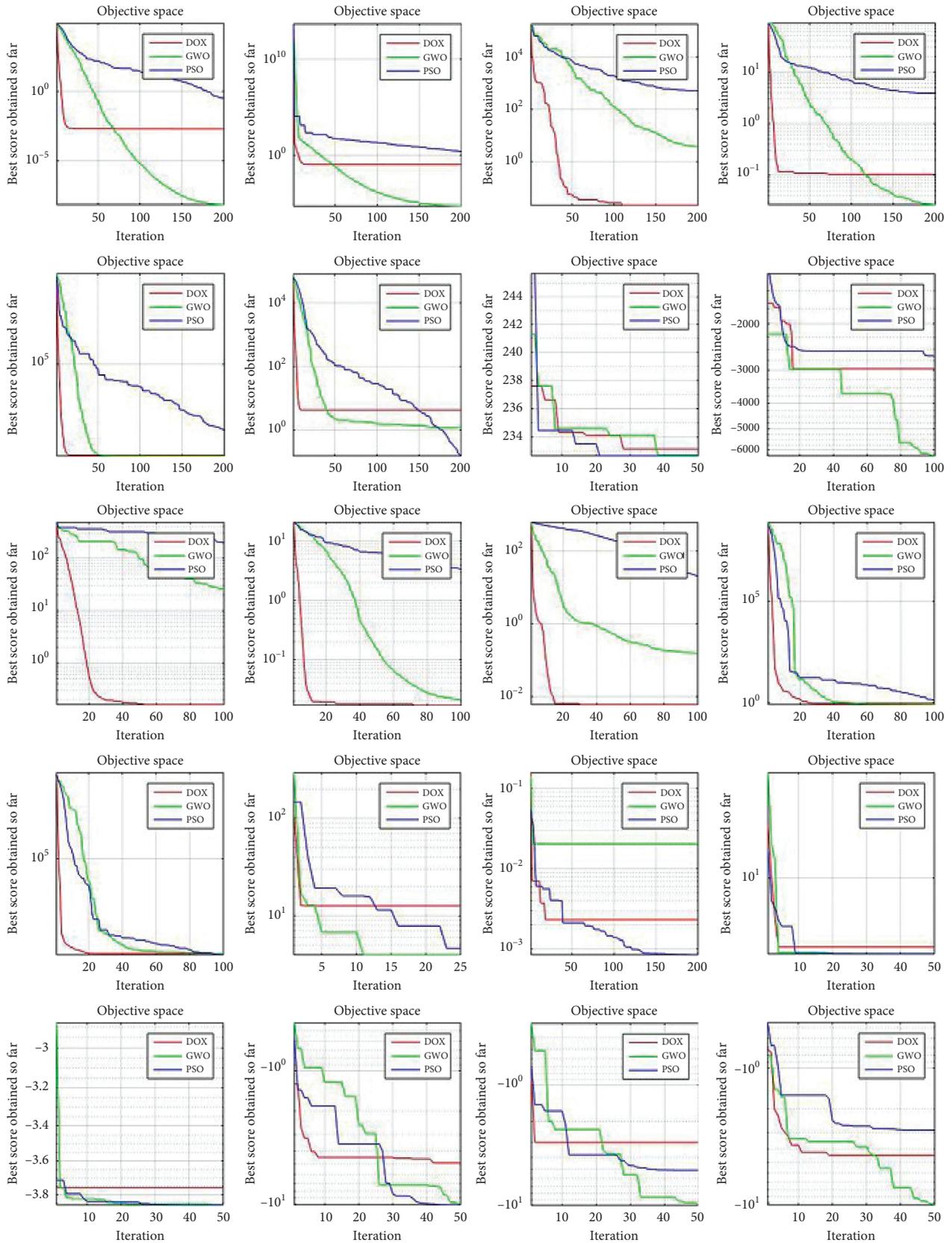


FIGURE 5: Comparison of convergence curves of DOX obtained in some of the benchmark problems.

TABLE 4: Unimodal functions.

| Function | Dim | Range | f_{\min} |
|--|-----|---------------|------------|
| $F_1(k) = \sum_{j=1}^m k_j^2$ | 30 | [-100, 100] | 0 |
| $F_2(k) = \sum_{j=1}^m k_j + \prod_{j=1}^m k_j $ | 30 | [-10, 10] | 0 |
| $F_3(k) = \sum_{j=1}^m (\sum_{j=1}^m k_j^2)^2$ | 30 | [-100, 100] | 0 |
| $F_4(k) = \max_j k_j , 1 \leq j \leq m$ | 30 | [-100, 100] | 0 |
| $F_5(k) = \sum_{j=1}^{m-1} [100(k_{j+1} - k_j^2)^2 + (k_j - 1)^2]$ | 30 | [-30, 30] | 0 |
| $F_6(k) = \sum_{j=1}^m (k_j + 0.5)^2$ | 30 | [-100, 100] | 0 |
| $F_7(k) = \sum_{j=1}^m ik_j^4 + \text{random}[0, 1]$ | 30 | [-1.28, 1.28] | 0 |

TABLE 5: Multimodal benchmark functions.

| Function | Dim | Range | f_{\min} |
|---|-----|--------------|---------------------|
| $F_8(k) = \sum_{j=1}^m -k_j \sin(\sqrt{ k_j })$ | 30 | [-500, 500] | -418.982×5 |
| $F_9(k) = \sum_{j=1}^m [k_j^2 - 10 \cos(2\pi k_j) + 10]$ | 30 | [-32, 32] | 0 |
| $F_{10}(k) = -20 \exp(-0.2\sqrt{(1/m)\sum_{j=1}^m k_j^2}) - \exp((1/m)\sum_{j=1}^m \cos(2\pi k_j)) + 20 + e$ | 30 | [-100, 100] | 0 |
| $F_{11}(k) = (1/4000)\sum_{j=1}^m k_j^2 - \prod_{j=1}^m \cos(k_j/\sqrt{j}) + 1$ | 30 | [-600, 600] | 0 |
| $F_{12}(k) = (\pi/m) \left\{ 10 \sin(\pi y_1) \sum_{j=1}^m (y_1 - 1)^2 [1 + 10 \sin^2(\pi y_{j+1})] + (y_m - 1)^2 \right\} + \sum_{j=1}^m u(k_j, 10, 100, 4) y_j = 1 + (k_j + 1/4)$ | 30 | [-50, 50] | 0 |
| $u(k_j, a, l, n) = \begin{cases} l(k_j - a)^n, & k_j > a. \\ 0, & -a < k_j < a. \\ l(-k_j - a)^n, & k_j < a \end{cases}$ | | | |
| $F_{13}(k) = 0.1 \left\{ \begin{array}{l} \sin^2(3\pi k_j) + \sum_{j=1}^m (k_j - 1)^2 [1 + \sin^2(3\pi k_j + 1)] \\ + (k_j - 1)^2 [1 + \sin^2(2\pi k_j)] \end{array} \right\} + \sum_{j=1}^m u(k_j, 5, 100, 4)$ | 30 | [-50, 50] | 0 |
| $F_{14}(k) = -\sum_{j=1}^m (\sin(k_j) \cdot \sin(j \cdot k_j^2/\pi))^{2n}, n = 10$ | 30 | [-0, π] | -4.687 |
| $F_{15}(k) = [e^{-\sum_{j=1}^m (k_j/\beta)^{2n}} - 2e^{-\sum_{j=1}^m k_j^2}] \cdot \prod_{j=1}^m \cos^2 k_j, n = 5$ | 30 | [-20, 20] | 0 |
| $F_{16}(k) = \{[\sum_{j=1}^m \sin^2(k_j)] - \exp(-\sum_{j=1}^m k_j^2)\} \cdot \exp[-\sum_{j=1}^m \sin^2 \sqrt{ k_j }]$ | 30 | [-10, 10] | -1 |

This problem has been popular among researchers in various studies.

Table 4 is a comparison of the best optimal solution for DOX and other documented approaches, such as GWO and PSO. According to this table, DOX will find an optimum design at a minimal rate. Table 4 shows DOX comparison of the historical effects of the issue of construction of pressure vessels. The DOX results work better than any other algorithm in terms of the best optimal solution.

The DOX algorithm obtained the near optimal solution in the initial steps of iterations and achieved better results than other optimization methods for pressure vessel problem.

The comparison of best optimal solution among several algorithms is given in Table 4. This problem has been tested

with different optimization methods such as GWO and PSO. The comparison for the best solution obtained by such algorithms is presented.

6. Statistical Testing

The ANOVA test was performed to test whether the outcomes obtained from the proposed algorithms vary statistically substantially from the findings of other algorithms. We took 30 as the sample size for the ANOVA test. We used 95% confidence for the ANOVA test. The results of the ANOVA test for the benchmark functions are shown in Table 8. The findings demonstrate that the DOX is statistically important relative to other rival algorithms.

TABLE 6: Fixed-dimension multimodal benchmark functions.

| Function | Dim | Range | f_{\min} |
|--|-----|-----------|------------|
| $F_{14}(k) = ((1/500) + \sum_1^{25} (1/j + \sum_{i=1}^2 (x_i - a_{ij})^6))^{-1}$ | 2 | [-65, 65] | 1 |
| $F_{15}(k) = \sum_{j=1}^{11} [a_j - (x_1(b_j^2 + b_j x_2)/b_j^2 + b_j x_2 + x_4)]^2$ | 4 | [-5, 5] | 0.00030 |
| $F_{16}(k) = 4x_i^2 - 2.1x_1^4 + (1/3)z_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$ | 2 | [-5, 5] | -1.0316 |
| $F_{17}(k) = (x_2 - (5.1/4\pi^2)x_1^2 + (5/\pi)x_1 - 6)^2 + 10(1 - (1/8\pi))\cos x_1 + 10$ | 2 | [-5, 5] | 0.398 |
| $F_{18}(k) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2) + 3x_2^2] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)]$ | 2 | [-2, 2] | 3 |
| $F_{19}(k) = -\sum_{j=1}^4 c_j \exp(\sum_{l=1}^3 a_{jl} (x_l - p_{jl})^2)$ | 3 | [1, 3] | -3.86 |
| $F_{20}(k) = -\sum_{j=1}^4 c_j \exp(\sum_{l=1}^6 a_{jl} (x_l - p_{jl})^2)$ | 6 | [0, 1] | -3.32 |
| $F_{21}(k) = -\sum_{j=1}^5 [(X - a_j)(X - a_j)^T + c_l]^{-1}$ | 4 | [0, 10] | -10.1532 |
| $F_{22}(k) = -\sum_{j=1}^7 [(X - a_j)(X - a_j)^T + c_l]^{-1}$ | 4 | [0, 10] | -10.1532 |
| $F_{23}(k) = -\sum_{j=1}^{10} [(X - a_j)(X - a_j)^T + c_l]^{-1}$ | 4 | [0, 10] | -10.1532 |

TABLE 7: Results comparison for pressure vessel design problem.

| Algo. | P_1 | P_2 | P_3 | P_4 | Optimum |
|-------|--------|--------|---------|----------|-----------|
| DOX | 0.7782 | 0.3848 | 40.3150 | 200.0000 | 5885.5773 |
| GWO | 0.7790 | 0.3846 | 40.3277 | 199.6502 | 5889.3689 |
| PSO | 0.7789 | 0.3847 | 40.3209 | 200.0000 | 5891.3879 |

TABLE 8: Unimodal functions.

| F | P value | DOX | GWO | PSO |
|-----|----------|---------|---------|---------|
| F1 | 1.91E-64 | GWO,PSO | DOX,PSO | DOX,GWO |
| F2 | 3.62E-65 | GWO,PSO | DOX,PSO | DOX,GWO |
| F3 | 3.57E-36 | GWO,PSO | DOX,PSO | DOX,GWO |
| F4 | 2.87E-22 | GWO,PSO | DOX,PSO | DOX,GWO |
| F5 | 2.15E-24 | GWO,PSO | DOX,PSO | DOX,GWO |
| F6 | 1.94E-54 | GWO,PSO | DOX,PSO | DOX,GWO |
| F7 | 1.56E-13 | GWO,PSO | DOX,PSO | DOX,GWO |
| F8 | 1.59E-43 | GWO,PSO | DOX,PSO | DOX,GWO |
| F9 | 1.88E-87 | GWO,PSO | DOX,PSO | DOX,GWO |
| F10 | 1.29E-34 | GWO,PSO | DOX,PSO | DOX,GWO |
| F11 | 1.87E-35 | GWO,PSO | DOX,PSO | DOX,GWO |
| F12 | 2.36E-23 | GWO,PSO | DOX,PSO | DOX,GWO |
| F13 | 1.91E-98 | GWO,PSO | DOX,PSO | DOX,GWO |
| F14 | 6.35E-36 | GWO,PSO | DOX,PSO | DOX,GWO |
| F15 | 1.14E-07 | GWO,PSO | DOX,PSO | DOX,GWO |
| F16 | 2.63E-35 | GWO,PSO | DOX,PSO | DOX,GWO |
| F17 | 1.83E-58 | GWO,PSO | DOX,PSO | DOX,GWO |
| F18 | 4.61E-36 | GWO,PSO | DOX,PSO | DOX,GWO |
| F19 | 9.54E-15 | GWO,PSO | DOX,PSO | DOX,GWO |
| F20 | 1.99E-73 | GWO,PSO | DOX,PSO | DOX,GWO |
| F21 | 2.54E-62 | GWO,PSO | DOX,PSO | DOX,GWO |
| F22 | 8.11E-06 | GWO,PSO | DOX,PSO | DOX,GWO |
| F23 | 2.70E-18 | GWO,PSO | DOX,PSO | DOX,GWO |
| F24 | 5.55E-36 | GWO,PSO | DOX,PSO | DOX,GWO |
| F25 | 1.22E-87 | GWO,PSO | DOX,PSO | DOX,GWO |
| F26 | 7.66E-45 | GWO,PSO | DOX,PSO | DOX,GWO |
| F27 | 2.48E-69 | GWO,PSO | DOX,PSO | DOX,GWO |
| F28 | 2.31E-43 | GWO,PSO | DOX,PSO | DOX,GWO |
| F29 | 1.14E-47 | GWO,PSO | DOX,PSO | DOX,GWO |

7. Conclusion and Future Scope

As per the comparison of DOX with other popular meta-heuristic algorithms such as PSO and DSO, DOX provides well competitive outcomes as presented in the results. The DOX is analyzed for the exploration and exploitation activity of agents using twenty-three test functions. The concise results, which are based on comparative analysis between the proposed DOX and other optimization algorithms, demonstrate that the approach suggested will cope with different kinds of constraints and provide stronger alternatives than any other optimizer. The suggested methodology is inspired by the real-life problems, which required less computational or mathematical efforts to find the best available optima.

Some other major findings may be preferred for future studies. DOX may be used to address various technological problems. Multiobjective problems can be solved as another future contribution as MODOX. Binary DOX might also be other benchmarks to expand this algorithm.

Appendix

A. Benchmark Functions

A.1. Unimodal Functions. The list of the unimodal test functions (F1–F7) is given in Table 4.

A.2. Multimodal Functions. The list of the multimodal test functions (F8–F16) is given in Table 5.

A.3. Fixed-Dimension Multimodal Functions. The list of the fixed-dimension multimodal test functions (F14–F23) is given in Table 6.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] H. Ali, W. Shahzad, and F. A. Khan, "Energy-efficient clustering in mobile ad-hoc networks using multi-objective particle swarm optimization," *Applied Soft Computing*, vol. 12, no. 7, pp. 1913–1928, 2012.
- [2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [3] G. Qi, H. Wang, M. Haner, C. Weng, S. Chen, and Z. Zhu, "Convolutional neural network based detection and judgement of environmental obstacle in vehicle operation," *CAAI Transactions on Intelligence Technology*, vol. 4, no. 2, pp. 80–91, 2019.
- [4] C. Zhu, W. Yan, X. Cai, S. Liu, T. H. Li, and G. Li, "Neural saliency algorithm guide bi-directional visual perception style transfer," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 1, pp. 1–8, 2020.
- [5] T. Wiens, "Engine speed reduction for hydraulic machinery using predictive algorithms," *International Journal of Hydromechatronics*, vol. 2, no. 1, pp. 16–31, 2019.
- [6] X. Xin Yao, Y. Yong Liu, and G. Guangming Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.
- [7] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co., Inc., Boston, Massachusetts, USA, 1989.
- [8] X. Xue, J. Lu, and J. Chen, "Using NSGA-III for optimising biomedical ontology alignment," *CAAI Transactions on Intelligence Technology*, vol. 4, no. 3, pp. 135–141, 2019.
- [9] W. Banzhaf, F. D. Francone, R. E. Keller, and P. Nordin, *Genetic Programming: An Introduction: On the Automatic Evolution of Computer Programs and its Applications*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.
- [10] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: formulation discussion and generalization," in *Proceedings of the 5th International Conference on Genetic Algorithms*, pp. 416–423, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, June 1993.
- [11] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [12] F. Xue, A. C. Sanderson, and R. J. Graves, "Multi-objective differential evolution - algorithm, convergence analysis, and applications," *IEEE Congress on Evolutionary Computation*, vol. 1, pp. 743–750, 2005.
- [13] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies – a comprehensive introduction," *Natural Computing*, vol. 1, no. 1, pp. 3–52, 2002.
- [14] X. Hu, C. A. C. Coello, and Z. Huang, "A new multi-objective evolutionary algorithm: neighbourhood exploring evolution strategy," *Engineering Optimization*, vol. 37, no. 4, pp. 351–379, 2005.
- [15] T. Jansen and C. Zarges, "Artificial immune systems for optimisation," in *Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference Companion - GECCO Companion '12*, pp. 1059–1078, Association for Computing Machinery, New York, NY, USA, July 2012.
- [16] F. Campelo, F. G. Guimarães, and H. Igarashi, "Overview of artificial immune systems for multi-objective optimization," in *Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization, EMO'07*, pp. 937–951, Springer-Verlag, Berlin, Heidelberg, March 2007.
- [17] K. M. Passino, "Bacterial foraging optimization," *International Journal of Swarm Intelligence Research*, vol. 1, no. 1, pp. 1–16, 2010.
- [18] "Multi-swarm cooperative multi-objective bacterial foraging optimisation," *International Journal of Bio-Inspired Computation*, vol. 13, no. 1, pp. 21–31, 2019.
- [19] Z. Chelly and Z. Elouedi, "A survey of the dendritic cell algorithm," *Knowledge and Information Systems*, vol. 48, no. 3, pp. 505–535, 2016.
- [20] A. Gandomi and A. Alavi: Krill Herd Algorithm.
- [21] P. J. M. Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*, Kluwer Academic Publishers, New York, NY, USA, 1987.
- [22] N. Krasnogor, *Memetic Algorithms*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 905–935, 2012.

- [23] M. Eusuff, K. Lansey, and F. Pasha, "Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization," *Engineering Optimization*, vol. 38, no. 2, pp. 129–154, 2006.
- [24] M. Dorigo, M. Birattari, and T. Stützle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.
- [25] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948, 1995.
- [26] D. Karaboga and B. Basturk, "Artificial bee colony (abc) optimization algorithm for solving constrained optimization problems," in *Foundations of Fuzzy Logic and Soft Computing*, P. Melin, O. Castillo, L. T. Aguilar, J. Kacprzyk, and W. Pedrycz, Eds., pp. 789–798, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
- [27] M. Neshat, A. Adeli, G. Sepidnam, M. Sargolzaei, and A. Najaran Toosi, "A review of artificial fish swarm optimization methods and applications," *International Journal on Smart Sensing and Intelligent Systems*, vol. 5, no. 1, pp. 108–148, 2012.
- [28] G. wolf optimizer, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014, <https://doi.org/10.1016/j.advengsoft.2013.12.007>.
- [29] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Computing and Applications*, vol. 27, no. 4, pp. 1053–1073, 2016.
- [30] J. Ma, H. Y. Chen, R. Su, Y. Wang, S. Zhang, and S. Shan, "Improved firefly algorithm and its application," *Proceedings of the 4th International Conference on Crowd Science and Engineering*, Association for Computing Machinery, in *Proceedings of the 4th International Conference on Crowd Science and Engineering, ICCSE'19*, pp. 180–185, October 2019.
- [31] X.-S. Yang, "Firefly algorithms for multimodal optimization," 2010, <https://arxiv.org/abs/1003.1466>.
- [32] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, no. C, pp. 51–67, 2016.
- [33] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "Gsa: a gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.
- [34] X.-S. Y. Seyedali Mirjalili and S. M. Mirjalili, "Binary bat algorithm," *Neural Computing and Applications*, vol. 25, no. 3–4, pp. 663–681, 2014.
- [35] Y. C. Shih, "A cuckoo search algorithm: effects of coevolution and application in the development of distributed layouts," *Journal of Algorithms & Computational Technology*, vol. 13, Article ID 1748302619889523, 2019.
- [36] S.-C. Chu, P.-W. Tsai, and J.-S. Pan, "Cat swarm optimization," *Lecture Notes in Computer Science*, in *Proceedings of the 9th Pacific Rim International Conference on Artificial Intelligence, PRICAI'06*, Springer-Verlag, Berlin, Heidelberg, pp. 854–858, August 2006.
- [37] V. Kumar, J. K. Chhabra, and D. Kumar, "Parameter adaptive harmony search algorithm for unimodal and multimodal optimization problems," *Journal of Computational Science*, vol. 5, no. 2, pp. 144–155, 2014.
- [38] L. Wang, Y. Mao, Q. Niu, and M. Fei, "A multi-objective binary harmony search algorithm," in *Lecture Notes in Computer Science Advances in Swarm Intelligence*, Y. Tan, Y. Shi, Y. Chai, and G. Wang, Eds., Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 74–81, 2011.
- [39] Y. Fang, G. Liu, He Yi, and Y. Qiu, "Tabu search algorithm based on insertion method," in *Proceedings of the international conference on neural networks and signal processing*, pp. 420–423, Nanjing, China, December 2003.
- [40] S. Carcangiu, A. Fanni, and A. Montisci, "Multiobjective tabu search algorithms for optimal design of electromagnetic devices," *IEEE Transactions on Magnetics*, vol. 44, no. 6, pp. 970–973, 2008.
- [41] V. Kumar, J. K. Chhabra, and D. Kumar, "Parameter adaptive harmony search algorithm for unimodal and multimodal optimization problems," *Journal of Computational Science*, vol. 5, no. 2, pp. 144–155, 2014.
- [42] S. He, Q. H. Wu, and J. R. Saunders, "Group search optimizer: an optimization algorithm inspired by animal searching behavior," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 973–990, 2009.
- [43] N. Ghorbani and E. Babaei, "Exchange market algorithm," *Applied Soft Computing*, vol. 19, pp. 177–187, 2014.
- [44] E. Atashpaz-Gargari and C. Lucas, "Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition," in *Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, pp. 4661–4667, Singapore, September 2007.
- [45] N. Moosavian, "Soccer league competition algorithm for solving knapsack problems," *Swarm and Evolutionary Computation*, vol. 20, pp. 14–22, 2015.
- [46] A. Husseinzadeh Kashan, "League championship algorithm (lca): an algorithm for global optimization inspired by sport championships," *Applied Soft Computing*, vol. 16, pp. 171–200, 2014.
- [47] F. Ramezani and S. Lotfi, "Social-based algorithm (sba)," *Applied Soft Computing*, vol. 13, no. 5, pp. 2837–2856, 2013.
- [48] Y. Tan, C. Yu, S. Zheng, and K. Ding, "Introduction to fireworks algorithm," *International Journal of Swarm Intelligence Research*, vol. 4, no. 4, pp. 39–70, 2013.
- [49] A. Kaveh and V. Mahdavi Dahoei, *Colliding Bodies Optimization*, 2015.
- [50] A. Gandomi, *Interior Search Algorithm*, 2014.
- [51] W. Zhao, L. Wang, and Z. Zhang, "Artificial ecosystem-based optimization: a novel nature-inspired meta-heuristic algorithm," *Neural Computing and Applications*, vol. 32, pp. 1–43, 2020.
- [52] L. Benasla, A. Belmadani, and R. Mostefa, "Spiral optimization algorithm for solving combined economic and emission dispatch," *International Journal of Electrical Power and Energy Systems*, vol. 62, pp. 163–174, 2014.
- [53] E. Bogar and S. Beyhan, "Adolescent identity search algorithm (aisa): a novel metaheuristic approach for solving optimization problems," *Applied Soft Computing*, vol. 95, Article ID 106503, 2020.
- [54] R. Wang, H. Yu, G. Wang, G. Zhang, and W. Wang, "Study on the dynamic and static characteristics of gas static thrust bearing with micro-hole restrictors," *International Journal of Hydromechanics*, vol. 2, p. 189, 2019.
- [55] S. Osterland and J. Weber, "Analytical analysis of single-stage pressure relief valves," *International Journal of Hydromechanics*, vol. 2, p. 32, 2019.
- [56] C. S. Marcin Molga, *Test Functions for Optimization Needs*, 2005.
- [57] B. Kannan and S. Kramer, "An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design," *Journal of Mechanical Design*, vol. 116, pp. 405–411, 1994.