

## Hybridize the Dwarf Mongoose Optimization (DMO) Algorithm to Obtain the Optimal Solution for Solve Optimization Problems



Omar D. Shalal\* , Ban A. Mitras 

Department of Mathematics, College of Computer Science & Mathematics, University of Mosul, Mosul 41002, Iraq

Corresponding Author Email: [omar.21csp24@student.uomosul.edu.iq](mailto:omar.21csp24@student.uomosul.edu.iq)

Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ijcmem.120207>

### ABSTRACT

**Received:** 28 November 2023

**Revised:** 17 March 2024

**Accepted:** 22 May 2024

**Available online:** 30 June 2024

#### Keywords:

*meta-heuristic algorithm, conjugate gradient algorithm, dwarf mongoose optimization (DMO) algorithm, sand cat swarm optimization (SCSO) algorithm,, hybrid algorithms*

In this paper, two distinct strategies were used to enhance problem-solving abilities. The first strategy involved developing a conjugate gradient algorithm in which several new parameters were derived and proposed. The second strategy included hybridizing the dwarf mongoose optimization (DMO) algorithm in two ways, the first using the community by taking advantage of the developed conjugate gradient algorithm that was extracted from the first strategy and obtaining the hybrid algorithm (CG-DMO) that gives better results than the results of the original algorithm. The second method is to combine the sand cat swarm optimization algorithm (SCSO) and the dwarf mongoose optimization algorithm (DMO), and a hybrid algorithm (SCSO-DMO) is obtained. The dwarf mongoose optimization (DMO) algorithm uses three mongoose social groups: the alpha group, the scout group, and the babysitter group to replicate their foraging behavior. The Alpha group underwent hybridization, using the attack method of sand cats, known for their keen hearing of low-frequency sounds and their adeptness at detecting prey by digging. This hybrid approach led to the development of an equation for identifying candidate food sites within the alpha group. The proposed algorithms (CG-DMO) and (SCSO-DMO) underwent extensive testing on standard test functions, resulting in superior results compared to the original algorithm.

## 1. INTRODUCTION

The great difference between living organisms in trying to survive in nature, searching for food, attacking, or hiding, has prompted many scientists to create mathematical models (algorithms) that simulate the social behavior of these living organisms. These models have become a major role in solving optimization problems, but due to the rapid development in various fields, individual algorithms have become. It does not always lead to optimal solutions, so it becomes necessary to combine algorithms and obtain a better mathematical model.

In the realm of solving optimization problems, methodologies are typically categorized into two primary types: deterministic algorithms and stochastic algorithms [1]. Within classical algorithmic frameworks, deterministic strategies have historically held precedence. However, in the realm of stochastic algorithms, two distinct sub-types emerge: heuristic and meta-heuristic algorithms [2]. Heuristic algorithms are grounded in a straightforward trial-and-error methodology. On the other hand, meta-heuristic algorithms commence with a diverse set of initial solutions, which are progressively refined through iterations. This approach is exemplified in various algorithms such as the particle swarm [3], gray wolf [4], bee colony [5], and genetic algorithms [6]. Nature's mechanisms have profoundly influenced the development of many meta-heuristic algorithms. Researchers

have harnessed natural processes to create algorithms capable of solving complex problems across various domains, including but not limited to the traveling salesman problem and optimal control, and medical image processing [7-10]. The efficacy of these nature-inspired algorithms stems from their capacity to replicate the most efficient characteristics observed in natural phenomena. An example of such innovation is the Dwarf Mongoose Optimization (DMO) algorithm, a meta-heuristic technique inspired by the foraging patterns observed in dwarf mongooses. This algorithm incorporates three primary social groups found within mongoose communities: the alpha group, the babysitter group, and the scout group to mimic their natural foraging strategies. Dwarf mongooses exhibit specialized behavioral adaptations for effective feeding, including strategies related to prey size, spatial utilization, group dynamics, and food distribution. The algorithm models this by having the alpha female lead the group, while a subset of the population, the babysitters, cares for the young during foraging. These babysitters are dynamically replaced as the algorithm progresses. Unlike their real-life counterparts, which don't use permanent nests and frequently change their foraging locations, the algorithm adapts these behaviors into its operational framework [11].

The work that we did was to hybridize the Dwarf Mongoose Optimization (DMO) algorithm using two algorithms, Specifically the Formulated conjugate gradient algorithm that

was derived in the first part of this paper, which is a Progress to the initial population of the (DMO) and obtained the (CG-DMO) algorithm. The second hybridization was by merging the Dwarf Mongoose Optimization (DMO) algorithm with the Sand Cat Swarm Optimization (SCSO) algorithm to develop the hybrid SCSO-DMO algorithm. The SCSO algorithm a meta-heuristic technique, draws inspiration from the survival instincts observed in sand cats, particularly their ability to detect low-frequency sounds under 2 kilohertz and their exceptional digging skills for hunting prey [12, 13]. The newly formulated SCSO-DMO algorithm has demonstrated its effectiveness in solving complex optimization problems. Empirical results have shown its ability to achieve optimal solutions, as evidenced by its performance on five global test functions, consistently reaching the optimal solution, which is zero.

The paper follows a structured framework outlined as follows: Section 2 developed conjugate gradient algorithm. Section 3 introduces the sand cat swarm optimization (SCSO) algorithm. Section 4 presents the dwarf mongoose optimization (DMO) algorithm. Section 5 presents the proposed algorithms (CG-DMO) and (SCSO-DMO). Section 6 provides conclusions drawn from the study.

## 2. DEVELOPED CONJUGATE GRADIENT ALGORITHM

It is one of the mathematical methods used to solve problems of finding the minimum or maximum of functions. This method is usually used in problems searching for solutions to systems of linear equations or improving performance in numerical research problems. The conjugate gradient method is based on the use of conjugate directives to quickly optimize the function being optimized. Instead of using traditional regression directions, the conjugate gradient method uses directions that are interconnected with each other. You start moving towards the initial downhill direction. A conjugate direction of progress is then determined based on the previous step and the interconnected trends. This process is repeated until the solutions improve simultaneously in the directions of all regressions.

**Definition 1:** The optimization [14]

It means finding the best solution to the given problem, and finding the minimum or maximum value of a function consisting of  $n$  variables, where  $n$  can be any integer greater than zero.

**Definition 2:** Global and local minimum [14]

**A-** The global minimum value: represents the lowest value of the function in the entire research field.

**B-** Local minimum value: represents the lowest value of the function in specific locations within the search field. Algorithms that trend toward a global minimum are known as globally convergent algorithms, while algorithms that trend toward a local minimum are known as locally convergent algorithms.

### 2.1 Derivation of new conjugation coefficients

In three term conjugates gradient direction Babaie-Kafaki and Ghanbari [15] proposed a new value of  $t^*$  as follow:

$$t^* = \frac{\|y_k\|}{\|s_k\|} + \frac{s_k^T y_k}{\|s_k\|^2}$$

Ibrahim and Shareef [16] proposed another value of  $t$  as follow:

$$t_k = \gamma \frac{\|y_k\|}{\|s_k\|} + (1 - \gamma) \frac{s_k^T y_k}{\|s_k\|^2}, \text{ where } \gamma \in (0,1)$$

Here, we drive a new conjugacy coefficient with a new value of  $t$  as [16, 17]:

$$\alpha_k = t_k = \frac{(s_k^T g_k/2)^2}{(f_{k+1} - f_k)^2}$$

$$d_{k+1}^{CG_3} = -g_{k+1} + \beta_k d_k - t_k \left( \frac{g_{k+1}^T d_k}{d_k^T y_k} \right) y_k$$

$$\text{Let } d_{k+1}^{CG_2} = d_{k+1}^{CG_3}$$

$$-g_{k+1} + \beta_k^{New} d_k = -g_{k+1} + \beta_k d_k - t_k \left( \frac{g_{k+1}^T d_k}{d_k^T y_k} \right) y_k$$

Multiply both sides of the above equation by  $y_k^T$  we get:

$$-y_k^T g_{k+1} + \beta_k^{New} y_k^T d_k = -y_k^T g_{k+1} + \beta_k y_k^T d_k - t_k \left( \frac{g_{k+1}^T d_k}{d_k^T y_k} \right) \|y_k\|^2$$

$$\beta_k^{New} = \beta_k - t_k \left( \frac{g_{k+1}^T d_k}{(d_k^T y_k)^2} \right) \|y_k\|^2 \quad (1)$$

$$\text{where, } t_k = \frac{(s_k^T g_k/2)^2}{(f_{k+1} - f_k)^2} > 0$$

From Ibrahim and Shareef [16],

**The first case:** If  $\beta_k = \beta^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k}$  then Eq. (1) become:

$$\beta_k^{New1} = \frac{g_{k+1}^T y_k}{d_k^T y_k} - t_k \left( \frac{g_{k+1}^T d_k}{(d_k^T y_k)^2} \right) \|y_k\|^2$$

**The second case:** If  $\beta_k = \beta^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}$  then Eq. (1) become:

$$\beta_k^{New2} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2} - t_k \left( \frac{g_{k+1}^T d_k}{(d_k^T y_k)^2} \right) \|y_k\|^2$$

**The third case:** If  $\beta_k = \beta^{PR} = \frac{g_{k+1}^T y_k}{g_k^T g_k}$  then Eq. (1) become:

$$\beta_k^{New3} = \frac{g_{k+1}^T y_k}{g_k^T g_k} - t_k \left( \frac{g_{k+1}^T d_k}{(d_k^T y_k)^2} \right) \|y_k\|^2$$

### 2.2 Convergence analysis of the new conjugate vector method

In this section, we will show that the proposed algorithm that has been identified achieves the property of sufficient

regression that satisfies the property of convergence.

**Assumption (1):** If  $f$  is bounded on the set  $s = \{x \in R^n: f(x) \leq f(x_0)\}$  and is differentiable with the gradient  $\nabla f$  and there is a Lipchitz Constant  $L > 0$ , then

$$\|\nabla f(x) - \nabla f(y)\| < L\|x - y\| \text{ for all } x, y \in s.$$

**Theorem (1):** The search direction  $d_k$  generated by the proposed algorithm of the developed CG achieves the property of sufficient steepness for each  $k$ , when the step size  $\lambda_k$  satisfies Wolfe conditions.

**Proof:** Using the principle of mathematical induction.

When  $k=0$ ,  $d_0 = -g_0 \Rightarrow d_0^T g_0 = -\|g_0\|^2 < 0$  then the theorem is true.

Now we assume that the theorem is true for all values of  $k \geq 0$ , i.e

$$g_k^T d_k < 0, g_k^T d_k \leq -c\|g_k\|^2, c > 0$$

We will now explain that the theorem is true when  $k+1$ .

$$d_{k+1} = -g_{k+1} + \beta_k d_k \quad (2)$$

By multiplying both sides of Eq. (2) above by  $g_{k+1}^T$  we get:

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + \beta_k g_{k+1}^T d_k \quad (3)$$

**The first case:** If  $\beta_k = \beta_k^{New1}$

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + \beta_k^{New1} g_{k+1}^T d_k$$

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + \left[ \frac{g_{k+1}^T y_k}{d_k^T y_k} - t_k \left( \frac{g_{k+1}^T d_k}{(d_k^T y_k)^2} \right) \|y_k\|^2 \right] g_{k+1}^T d_k \quad (4)$$

From Wolff's second strong condition, as shown in Eq. (5) below:

$$g(x_k + \alpha_k d_k)^T d_k \leq -\sigma g_k^T d_k \Rightarrow g_{k+1}^T d_k \leq -\sigma g_k^T d_k \quad (5)$$

It is known that

$$g_{k+1}^T y_k = g_{k+1}^T (g_{k+1} - g_k) \|g_{k+1}\|^2 - g_{k+1}^T g_k$$

Taking advantage of one direction of Powell's recovery condition, as shown in Eq. (6):

$$g_{k+1}^T y_k \leq \|g_{k+1}\|^2 + 0.2 \|g_{k+1}\|^2 = 1.2 \|g_{k+1}\|^2 \quad (6)$$

From Wolff's strong condition we get the following Eq. (7):

$$-(1 - \sigma) g_k^T d_k \leq y_k^T d_k \leq -(1 + \sigma) g_k^T d_k \quad (7)$$

Substituting Eqs. (5), (6) and (7) into Eq. (4) results in:

$$+ \frac{c\sigma \|g_k\|^2}{d_k^T y_k} \left[ 1.2 \|g_{k+1}\|^2 - t_k \frac{c\sigma \|g_k\|^2}{d_k^T y_k} \cdot \|y_k\|^2 \right]$$

Since  $d_k^T y_k > 0$ ,  $c > 0$ ,  $0.5 < \sigma < 1$ , therefore the part is  $W = \frac{c\sigma \|g_k\|^2}{d_k^T y_k} > 0$ . This leads to

$$g_{k+1}^T d_{k+1} \leq -\|g_{k+1}\|^2 + 1.2W \|g_{k+1}\|^2 - t_k W^2 \cdot \|y_k\|^2$$

Since  $t_k > 0$ ,  $t_k W^2 \cdot \|y_k\|^2 > 0$ , this leads to

$$g_{k+1}^T d_{k+1} \leq -(1 - 1.2W) \|g_{k+1}\|^2$$

$$g_{k+1}^T d_{k+1} \leq -\Omega_1 \|g_{k+1}\|^2$$

where,  $\Omega_1 = 1 - 1.2W$ , when  $1 - 1.2W > 0$ .

**The second case:** If  $\beta_k = \beta_k^{New2}$  in Eq. (3), we get:

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + \beta_k^{New2} g_{k+1}^T d_k + \left[ \frac{\|g_{k+1}\|^2}{\|g_k\|^2} - t_k \left( \frac{g_{k+1}^T d_k}{(d_k^T y_k)^2} \right) \|y_k\|^2 \right] g_{k+1}^T d_k \quad (8)$$

Substituting Eq. (5) into Eq. (8) we get:

$$+ \left[ \frac{\|g_{k+1}\|^2}{\|g_k\|^2} - t_k \left( \frac{c\sigma \|g_k\|^2}{(d_k^T y_k)^2} \right) \|y_k\|^2 \right] (c\sigma \|g_k\|^2) + c\sigma \|g_{k+1}\|^2 - t_k \left( \frac{c^2 \sigma^2 \|g_k\|^4}{(d_k^T y_k)^2} \right) \|y_k\|^2$$

Since  $t_k > 0$ ,  $t_k \left( \frac{c^2 \sigma^2 \|g_k\|^4}{(d_k^T y_k)^2} \right) \|y_k\|^2 > 0$ . This leads to

$$g_{k+1}^T d_{k+1} \leq -(1 - c\sigma) \|g_{k+1}\|^2$$

$$g_{k+1}^T d_{k+1} \leq -\Omega_2 \|g_{k+1}\|^2$$

where,  $\Omega_2 = 1 - c\sigma$ , when  $1 - c\sigma > 0$ ,  $c > 0$ ,  $0.5 < \sigma < 1$ .

**The third case:** If  $\beta_k = \beta_k^{New3}$  in Eq. (3) we get:

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + \beta_k^{New3} g_{k+1}^T d_k$$

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + \left[ \frac{g_{k+1}^T y_k}{\|g_k\|^2} - t_k \left( \frac{g_{k+1}^T d_k}{(d_k^T y_k)^2} \right) \|y_k\|^2 \right] g_{k+1}^T d_k \quad (9)$$

Substituting Eqs. (5) and (6) into Eq. (9) we get:

$$+ \left[ \frac{1.2 \|g_{k+1}\|^2}{\|g_k\|^2} - t_k \left( \frac{c\sigma \|g_k\|^2}{(d_k^T y_k)^2} \right) \|y_k\|^2 \right] (c\sigma \|g_k\|^2)$$

$$- t_k \left( \frac{c^2 \sigma^2 \|g_k\|^4}{(d_k^T y_k)^2} \right) \|y_k\|^2$$

Since  $t_k > 0$ ,  $t_k \left( \frac{c^2 \sigma^2 \|g_k\|^4}{(d_k^T y_k)^2} \right) \|y_k\|^2 > 0$ . This leads to

$$g_{k+1}^T d_{k+1} \leq -(1 - 1.2c\sigma) \|g_{k+1}\|^2$$

$$g_{k+1}^T d_{k+1} \leq -\Omega_3 \|g_{k+1}\|^2,$$

where,  $\Omega_3 = 1 - 1.2 c\sigma$ , when  $1 - 1.2 c\sigma > 0$ ,  $c > 0$ ,  $0.5 < \sigma < 1$

### 2.3 Comprehensive convergence analysis of the proposed algorithm

**Lemma (1):** Suppose that Assumption (1) is fulfilled and that the conjugate gradient method is fulfilled, since  $d_k$  is a slope search direction, and  $\alpha_k$  is generated from the strong Wolff conditions (SWP) if  $\sum_{k=1}^{\infty} \frac{1}{\|d_{k+1}\|^2} = \infty$  then  $\lim_{k \rightarrow \infty} \inf \|g_k\| = 0$ .

**Theorem (2):** Suppose that Assumption (1) is fulfilled and that the proposed conjugate gradient method is fulfilled in the direction of the search slope  $d_k$ , and that the step length  $\alpha_k$  is generated from the conditions (SWP) then  $\lim_{k \rightarrow \infty} \inf \|g_k\| = 0$ .

**Proof:** Using Lemma (1), and since the algorithm fulfills theorem (1), and if  $g_{k+1} \neq 0$ , then we must prove that  $\|d_{k+1}\|$  is constrained from above, we take  $\|\cdot\|$  for both sides of the Eq. (2), we get

$$\begin{aligned} \|d_{k+1}\| &= \|-g_{k+1} + \beta_k d_k\| \\ \Rightarrow \|d_{k+1}\| &\leq \|g_{k+1}\| + |\beta_k| \|d_k\| \end{aligned} \quad (10)$$

**The first case:** If  $\beta_k = \beta_k^{New1}$

$$\Rightarrow \|d_{k+1}^{New1}\| \leq \|g_{k+1}\| + |\beta_k^{New1}| \|d_k\|$$

$$|\beta_k^{New1}| = \left| \frac{1}{d_k^T y_k} \left[ g_{k+1}^T y_k - t_k \left( \frac{g_{k+1}^T d_k}{d_k^T y_k} \right) \|y_k\|^2 \right] \right|$$

Using Eqs. (5) and (6) we get the following

$$|\beta_k^{New1}| \leq \left| \frac{1}{d_k^T y_k} \left[ 1.2 \|g_{k+1}\|^2 - t_k \frac{-\sigma g_k^T d_k}{d_k^T y_k} \cdot \|y_k\|^2 \right] \right|$$

$$\Rightarrow |\beta_k^{New1}| \leq A_1$$

$$\Rightarrow \|d_{k+1}^{New1}\| \leq \|g_{k+1}\| + A_1 \|d_k\|$$

$$\Rightarrow \|d_{k+1}^{New1}\| \leq T_1 + A_1 U_1 \leq M_1$$

$$\Rightarrow \|d_{k+1}^{New1}\| \leq M_1 \Rightarrow \frac{1}{\|d_{k+1}^{New1}\|} \geq \frac{1}{M_1}$$

$$\Rightarrow \sum_{k=1}^{\infty} \frac{1}{\|d_{k+1}^{New1}\|^2} \geq \sum_{k=1}^{\infty} \frac{1}{M_1^2} = \frac{1}{M_1^2} \sum_{k=1}^{\infty} 1 = +\infty$$

According to Lemma (1), this leads to

$$\lim_{k \rightarrow \infty} \inf \|g_k\| = 0$$

**The second case:** If  $\beta_k = \beta_k^{New2}$  in Eq. (10) we get:

$$\Rightarrow \|d_{k+1}^{New2}\| \leq \|g_{k+1}\| + |\beta_k^{New2}| \|d_k\|$$

$$|\beta_k^{New2}| = \left| \frac{\|g_{k+1}\|^2}{\|g_k\|^2} - t_k \left( \frac{g_{k+1}^T d_k}{(d_k^T y_k)^2} \right) \|y_k\|^2 \right|$$

Using Eq. (5) we get the following

$$|\beta_k^{New2}| \leq \left| \frac{\|g_{k+1}\|^2}{\|g_k\|^2} - t_k \left( \frac{c\sigma \|g_k\|^2}{(d_k^T y_k)^2} \right) \|y_k\|^2 \right|$$

$$\Rightarrow |\beta_k^{New2}| \leq A_2$$

$$\Rightarrow \|d_{k+1}^{New2}\| \leq \|g_{k+1}\| + A_2 \|d_k\|$$

$$\Rightarrow \|d_{k+1}^{New2}\| \leq T_2 + A_2 U_2 \leq M_2$$

$$\Rightarrow \|d_{k+1}^{New2}\| \leq M_2 \Rightarrow \frac{1}{\|d_{k+1}^{New2}\|} \geq \frac{1}{M_2}$$

$$\Rightarrow \sum_{k=1}^{\infty} \frac{1}{\|d_{k+1}^{New2}\|^2} \geq \sum_{k=1}^{\infty} \frac{1}{M_2^2} = \frac{1}{M_2^2} \sum_{k=1}^{\infty} 1 = +\infty$$

According to Lemma (1), this leads to

$$\lim_{k \rightarrow \infty} \inf \|g_k\| = 0$$

**The third case:** If  $\beta_k = \beta_k^{New3}$  in Eq. (10) we get:

$$\Rightarrow \|d_{k+1}^{New3}\| \leq \|g_{k+1}\| + |\beta_k^{New3}| \|d_k\|$$

$$|\beta_k^{New3}| = \left| \frac{g_{k+1}^T y_k}{g_k^T g_k} - t_k \left( \frac{g_{k+1}^T d_k}{(d_k^T y_k)^2} \right) \|y_k\|^2 \right|$$

Using Eqs. (5) and (6) we get the following

$$|\beta_k^{New3}| \leq \left| \frac{1.2 \|g_{k+1}\|^2}{\|g_k\|^2} - t_k \left( \frac{c\sigma \|g_k\|^2}{(d_k^T y_k)^2} \right) \|y_k\|^2 \right|$$

$$\Rightarrow |\beta_k^{New3}| \leq A_3$$

$$\Rightarrow \|d_{k+1}^{New3}\| \leq \|g_{k+1}\| + A_3 \|d_k\|$$

$$\Rightarrow \|d_{k+1}^{New3}\| \leq T_3 + A_3 U_3 \leq M_3$$

$$\Rightarrow \|d_{k+1}^{New3}\| \leq M_3 \Rightarrow \frac{1}{\|d_{k+1}^{New3}\|} \geq \frac{1}{M_3}$$

$$\Rightarrow \sum_{k=1}^{\infty} \frac{1}{\|d_{k+1}^{New3}\|^2} \geq \sum_{k=1}^{\infty} \frac{1}{M_3^2} = \frac{1}{M_3^2} \sum_{k=1}^{\infty} 1 = +\infty$$

According to Lemma (1), this leads to

$$\lim_{k \rightarrow \infty} \inf \|g_k\| = 0$$

### 3. SAND CAT SWARM OPTIMIZATION (SCSO) ALGORITHM

Sand cats, belonging to the family of mammals, thrive in harsh desert environments such as the Arabian Peninsula, Central Asia, and the African Sahara. Their ability to endure high temperatures is facilitated by dense fur covering the soles of their feet, providing insulation against extreme desert conditions. Moreover, the unique properties of their fur make detection and tracking processes challenging.

Sand cats typically have a body length ranging from 45 to 57 cm, with a tail length of approximately 28 to 35 cm, and an

adult weight ranging from 1 to 3.5 kg. The ears of the sand cat play a crucial role in prey detection and tracking processes. The nocturnal, subterranean, and speedy nature of these cats makes them distinctive, which is why sand cats reveal that their prey (insects and rodents) moves under the ground, Figure 1 shows sand cats in their natural habitat [18, 19].



**Figure 1.** Sand cats in their natural habitat, depicted in various activities including living, searching, and hunting

### 3.1 Mathematical model for SCSO

The Sand Cat Swarm Optimization (SCSO) algorithm draws its inspiration from the unique hunting technique of sand cats, which relies on detecting low-frequency sounds. This aspect is integrated into the SCSO algorithm by assigning a sensitivity range to each virtual 'cat', enabling them to detect frequencies below 2 kilohertz. The algorithm is designed to decrease this detection threshold,  $r_G^{\rightarrow}$ , linearly from 2 kilohertz to zero as the algorithm progresses through its iterations.

To simulate the exploratory behavior of sand cats, the SCSO algorithm begins with a randomly initialized search space. Each virtual cat in the algorithm is assigned a random initial position, emulating the way sand cats explore new territories in their natural habitat. This approach allows the algorithm to cover a wide and varied search area, enhancing its ability to discover optimal solutions [10]. In this way, cats can explore new areas as described in the equations below [13]:

$$r_G^{\rightarrow} = S_m - \left( \frac{2 * S_m * iter_c}{iter_{max} * iter_{max}} \right) \quad (11)$$

$$R^{\rightarrow} = 2 * r_G^{\rightarrow} * rand(0,1) - r_G^{\rightarrow} \quad (12)$$

$$r^{\rightarrow} = r_G^{\rightarrow} * rand(0,1) \quad (13)$$

$R^{\rightarrow}$  is the carrier responsible for the conversions and depends on the general sensitivity range  $r_G^{\rightarrow}$ . The locations of the sand cats are updated based on the best candidate location ( $pos_{bc}^{\rightarrow}$ ) obtained so far and its sensitivity range  $r^{\rightarrow}$ . Eq. (14) represents the exploration phase.

$$pos^{\rightarrow}(i+1) = r^{\rightarrow} * (pos_{bc}^{\rightarrow}(i) - rand(0,1) * pos_c^{\rightarrow}(i)) \quad (14)$$

$$pos_{rad}^{\rightarrow} = |rand(0,1) * pos_b^{\rightarrow}(i) - pos_c^{\rightarrow}(i)| \quad (15)$$

$$pos^{\rightarrow}(i+1) = pos_b^{\rightarrow}(i) - r^{\rightarrow} * pos_{rad}^{\rightarrow} * \cos(\theta) \quad (16)$$

During the exploitation phase of the Sand Cat Swarm Optimization (SCSO) algorithm, the methodology for updating the position of each agent 'cat' is based on calculating the distance between the optimal position ( $pos_b^{\rightarrow}$ ) and its current position ( $pos_c^{\rightarrow}$ ). This process is visualized as navigating the periphery of a circle. Key to this phase is the implementation of a random angle  $\theta$ , which dictates the direction of each cat's movement. The angle is randomly

chosen within a full circular range, translating to values between 0 and 360 degrees. This is mathematically represented as a range between -1 and 1, ensuring comprehensive coverage of all possible movement directions on the circle.

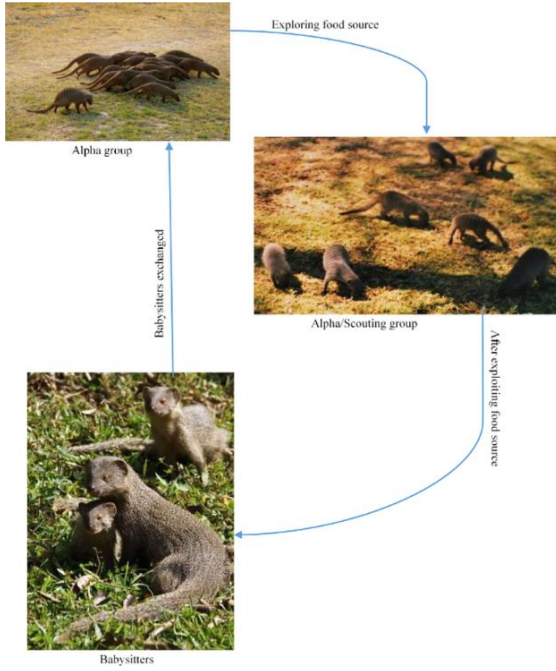
To enhance the algorithm's efficiency and avoid getting local solution, the SCSO utilizes the roulette wheel selection algorithm. This approach randomly selects an angle  $\theta$ , thereby diversifying the search and exploration paths of each agent. The incorporation of this random angle in Eq. (15) is crucial. It significantly influences the approach and direction of each individual in the population towards the target (hunting). Eq. (16) then provides the formula for updating the position of each cat in this phase, reflecting their movement towards the optimal point while considering the randomized directional input. This approach effectively balances the exploration and exploitation capabilities of the algorithm, thereby enhancing its overall effectiveness in discovering optimal solutions.

### 3.2 Algorithm for SCSO

- 1-Start and Set maximum iteration count.
- 2-Evaluate the fitness function of the objective function.
- 3-Initialization of variables:  $r^{\rightarrow}, r_G^{\rightarrow}, R$ .
- 4-While ( $i \leq$  maximum iteration)
- 5-For each search agent
- 6-Generate a random angle based on the roulette wheel selection.
- 7-IF ( $|R| \leq 1$ ) then
- 8-Update the search agent's position using Eq. (16):  
 $pos_b^{\rightarrow}(i) - r^{\rightarrow} * pos_{rad}^{\rightarrow} * \cos(\theta)$
- 9-Else
- 10-Update the search agent's position using Eq. (14):  
 $r^{\rightarrow} * (pos_{bc}^{\rightarrow}(i) - rand(0,1) * pos_c^{\rightarrow}(i))$
- 11-End
- 12-End
- 13-Increment iteration count.
- 14-End

### 4. DWARF MONGOOSE OPTIMIZATION (DMO) ALGORITHM

The habitat of the dwarf mongoose typically includes regions abundant in termite mounds, rocks, and hollow trees, providing ample shelter, particularly in semi-desert areas and savannah shrublands across Africa. Remarkably, the dwarf mongoose ranks among the smallest carnivorous animals known. Its body length is approximately 47 cm. The weight of an adult ferret is about 400 grams. Females outperform males and children outperform their older brothers. Division of labor and altruism are the highest recorded in mammals. The dwarf mongoose relies on a semi-nomadic lifestyle, covering a relatively long distance. Therefore, it detects hunting areas better without returning to its previous sleeping mound, so it does not exploit it well and does not exhaust all its prey. The group of dwarf mongooses operates cohesively during feeding, guided by the chirps emitted by the alpha female. The distance covered by a dwarf mongoose group is contingent upon factors such as the presence of offspring and the overall size of the group. Figure 2 shows The optimization procedures of the DMO.



**Figure 2.** The optimization procedures of the proposed DMO

#### 4.1 Mathematical model for DMO

The initial population is randomly initialized within the search space defined by the upper bound (UB) and lower bound (LB) for the given optimization problem. This process is represented by Eq. (17).

$$X = \begin{bmatrix} X_{1,1} & X_{1,2} & \dots & X_{1,d-1} & X_{1,d} \\ X_{2,1} & X_{2,2} & \dots & X_{2,d-1} & X_{2,d} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ X_{n,1} & X_{n,2} & \dots & X_{n,d-1} & X_{n,d} \end{bmatrix} \quad (17)$$

The population of current candidates, denoted as  $X$ , is randomly generated using Eq. (18), where,  $x_{i,j}$  represents the position of dimension ( $j$ ) of the population ( $i$ ),  $n$  refers to the size of the population, and  $d$  refers to the dimension of the problem.

$$x_{i,j} = \text{unifrnd}(\text{Varmin}, \text{Varmax}, \text{Varsize}) \quad (18)$$

where,  $\text{unifrnd}$  is a random number uniformly distributed between the lower and upper limits of the problem ( $\text{Varmin}$ ,  $\text{Varmax}$ ).  $\text{Varsize}$  is the dimension size of the problem. The social structure of the dwarf mongoose is segmented into three groups: the alpha group, the scouts, and the babysitters. Each group plays a role in contributing to compensatory behavioral conditioning, as outlined below.

##### 4.1.1 Alpha group

After preparing the population, the fitness of each solution is evaluated by calculating the population fitness using Eq. (19). Subsequently, the alpha female ( $\alpha$ ) is selected based on this probability.

$$\alpha = \frac{f_i t_i}{\sum_{i=1}^n f_i t_i} \quad (19)$$

To determine the position of the food candidate ( $x_{i+1}$ ),

DMO uses Eq. (20).

$$x_{i+1} = x_i + ph_i * \text{peep} \quad (20)$$

So that  $ph_i$  is a random number uniformly distributed between  $[-1,1]$ . The alpha female's vocalization that keeps the family within a path is denoted by  $\text{peep}$ . The initial sleep stack is set to  $\emptyset$  in which all the dwarf mongoose sleep. After each iteration, the sleep hill is calculated as shown in Eq. (21).

$$sm_i = \frac{f_i t_{i+1} - f_i t_i}{\max\{f_i t_{i+1}, f_i t_i\}} \quad (21)$$

The average value of the sleep hill is obtained using Eq. (22).

$$\phi_i = \frac{\sum_{i=1}^n sm_i}{n} \quad (22)$$

##### 4.1.2 Scouting group

The dwarf mongoose optimization algorithm starts with the exploration phase, so that the scout group searches for the new sleeping pile, so that the mongoose does not visit the previous sleeping pile, which ensures the exploration process. This stage is to evaluate the success or failure in knowing where the next sleeping pile [20]. Eq. (23) simulates the scouting mongoose.

$$x_{i+1} = \begin{cases} x_i - cf * ph_i * \text{rand}(0,1) * [x_i - m_i] & \text{if } \phi_{i+1} > \phi_i \\ x_i + cf * ph_i * \text{rand}(0,1) * [x_i - m_i] & \text{else} \end{cases} \quad (23)$$

The variable  $\text{rand}$  represents a random number between 0 and 1.  $cf = (1 - \frac{\text{iter}}{\max\_iter})^{(\frac{2 \cdot \text{iter}}{\max\_iter})}$  It denotes the parameter responsible for the collective voluntary movement of the ferret group, and it decreases linearly with iterations. The vector  $m$  determines the direction of movement of the mongoose towards the new sleeping pile.

##### 4.1.3 Babysitters group

The care of the young within the group is typically assigned to lower-ranking members who take turns in this role. This arrangement permits the alpha female, often the mother, to guide the others in daily food-gathering activities. She typically revisits the young midday or during the evening to nurse them. The quantity of caregivers is influenced by the overall number of the group. This dynamic is factored into the population algorithm, adjusting the total count in accordance with a predetermined percentage.

#### 4.2 Algorithm for DMO

- 1-Start
- 2-Define algorithm parameters: [ $\text{peep}$ ].
- 3-Create initial mongoose population (search agents):  $n$ .
- 4-Set the initial number of caregivers:  $bs$ .
- 5-Adjust the population:  $n = n - bs$ .
- 6-Define the caregiver rotation threshold  $L$ .
- 7-For  $i = 1$ : maximum iteration.
- 8-Assess mongoose fitness and start the time counter.
- 9-Identify the alpha mongoose using Eq. (19).
- 10-Generate a potential food location according to Eq. (20).
- 11-Evaluate new fitness of  $x_{i+1}$ .
- 12-Determine the sleeping mound location with Eq. (21).

- 13-Calculate the average of the sleeping mound from Eq. (22).
- 14-Formulate the movement vector,  $m = \sum_{i=1}^n \frac{x_i sm_i}{x_i}$ .
- 15-Exchange babysitters if  $C > L$  and set.
- 16-Initialize bs position and calculate fitness,  $f_i t_i \leq \alpha$ .
- 17-Simulate the scout mongoose next position using Eq. (23).
- 18-Update best solution so far
- 19-End of Iteration Loop
- 20-Output Best Solution
- 21-End of Algorithm

## 5. HYBRID THE DWARF MONGOOSE OPTIMIZATION (DMO)

Metaheuristic algorithms are a modern and innovative approach to artificial intelligence. Swarm-based algorithms are an effective tool for solving complex problems and achieving common goals through continuous interaction and collaboration among individual and members.

In this section, the two proposed algorithms will be presented (CG-DMO) and (SCSO-DMO).

### 5.1 Hybrid the dwarf mongoose optimization (DMO) by developed conjugate gradient algorithm

Conjugate gradient is an iterative method that starts from a specific point in a direction and generates a sequence of iterations until the minimum value of the function is reached. The hybridization process occurs by linking two methods, such as if, for example, one of the two methods has good computational properties and the second method has strong comprehensive convergence properties. Figure 3 shows the steps of hybridization using developed conjugate gradient algorithm.

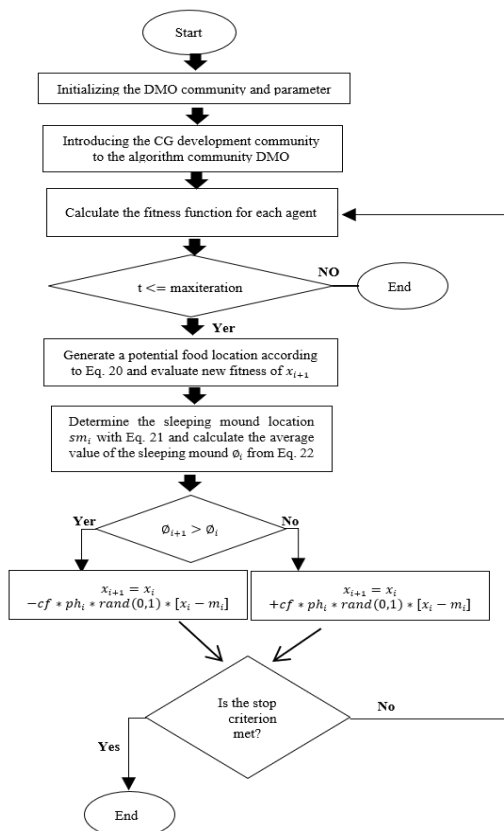


Figure 3. The steps of the CG-DMO

## 5.2 Hybridization of dwarf mongoose optimization (DMO) algorithm by sand cats swarm optimization (SCSO)

In this section, the dwarf mongoose optimization (DMO) algorithm tool was improved using some equations of the sand cat swarm optimization (SCSO) algorithm, that is, the desired features were used that contributed to improving the performance of the dwarf mongoose optimization algorithm. Hybridization was the use of equations to obtain the optimal solution.

### 5.2.1 Mathematical model for SCSO-DMO

The hybridization that we conducted differs from the hybridization used previously, which depends only on the initial population, that is, introducing the best solutions for the first algorithm as an initial population for the second algorithm, and it does not always reach the global optimal solution, while the hybridization that we performed is the use of mathematical models (equations), that is, entering a step by step, the Dwarf Mongoose Optimization (DMO) algorithm was improved in the Alpha group and an equation was developed to determine the location of food candidate in this group using the sand cat attack method, which is characterized by its ability to hear low-frequency sounds less than 2 kilohertz and an amazing ability to dig in search of prey. In addition to developing a Scouting Group using the vector  $R^{\rightarrow}$  is the carrier responsible for the conversions and depends on the general sensitivity range  $r_G^{\rightarrow}$ .

**Alpha group.** After preparing the population using Eq. (7), the suitability of each solution is calculated by calculating the population fitness using Eq. (9), so that an alpha female ( $\alpha$ ) is chosen. To know the location of the candidate food, the distance between the current position and any location ( $k$ ) for any element of group Alpha ( $\alpha$ ) is calculated as shown in Eq. (24).

$$Rand - position = abs(rand(0,1) * (x_i - x_k)) \quad (24)$$

So that the position of the food candidate, is determined as in Eq. (25) using Eqs. (12), (13).  $R^{\rightarrow}$  refers to the carrier, which is responsible for the conversions and depends on the general sensitivity range  $r_G^{\rightarrow}$  Each search agent updates its position and its sensitivity range  $r^{\rightarrow}$ .

$$x_{i+1} = r^{\rightarrow} * R^{\rightarrow} * rand(0,1) * Rand - position + phi_i * peep \quad (25)$$

$$peep = x_i - r^{\rightarrow} * c_1 * x_k \quad (26)$$

$$c_1 = 2 * rand(0,1) * \left( \frac{i}{iter_{max}} \right) \quad (27)$$

The initial sleep stack is set to  $\emptyset$  in which each search agent sleeps. After each iteration, the sleep hill is computed using Eq. (21), and the average value of the sleep hill is also determined using Eq. (22). The evaluation of the next food source or sleeping pile occurs once the babysitter exchange criterion is satisfied.

**Scouting group.** The proposed algorithm (SCSO-DMO) moves to the exploration stage if  $\phi_{i+1} > \phi_i$  as in Eq. (28).

$$x_{i+1} = x_i - cf * phi_i * rand * [x_i - R^{\rightarrow}] \quad (28)$$

If  $\emptyset_{i+1} \leq \emptyset_i$ , then the proposed algorithm (SCSO-DMO) begins with the exploitation phase, and this is done using Eq. (29).

$$x_{i+1} = x_i + cf * ph_i * rand * [x_i - R^{\rightarrow}] \quad (29)$$

The vector  $R^{\rightarrow}$  determines the direction of movement of thesearch agent towards the new sleeping pile, which depends on the general sensitivity range  $r_G^{\rightarrow}$  of the search agents.

### 5.2.2 Algorithm of SCSC-DMO

- 1-Initialize the population (search agent(n)).
- 2-Initialize the  $r, r_G, R, ns, \emptyset$ .
- 3-set  $n=ns$ .
- 4-**While** ( $i \leq$  maximum iteration)
- 5-Calculate the fitness function of the objective function.
- 6-Use Eq. (19) to identify the alpha, the best performing agent.
- 7-Apply Eq. (25) to produce a candidate food position.
- 8-Evaluate the fitness of the new position  $x_{i+1}$ .
- 9-Use Eq. (21) to determine the sleeping mound's position.
- 10-Use Eq. (22) to find the average value of  $\emptyset$  for the sleeping mound.
- 11-**For** each search agent
- 12-**IF** ( $\emptyset_{i+1} > \emptyset_i$ ) then
- 13-Update the search agent position based on the **Eq. (28)**:  
 $x_{i+1} = x_i - cf * ph_i * rand * [x_i - R^{\rightarrow}]$

14-**Else**

15-Update the search agent position based on the **Eq. (29)**:

$$x_{i+1} = x_i + cf * ph_i * rand * [x_i - R^{\rightarrow}]$$

16-**End**

17-**End**

18- $i = i + 1$

19-**End**

In general, the above tables showed the success of the two proposed algorithms (SCSO-DMO) and (CG-DMO) in obtaining the optimal solution. This result puts the two proposed algorithms (SCSO-DMO) and (CG-DMO) at the forefront for solving complex optimization problems. In addition to comparison with some contemporary algorithms (MFO, GWO, ZOA, MGO) refer to Moth-Flame Optimization, Zebra Optimization, Mountain Gazelle Optimizer, and Gray Wolf Optimizer respectively. The success rate of the hybrid algorithm reached 100%. The results were verified by applying the program to five standard test functions shown in Table 1. The results were for two possibilities, shown in Tables 2 and 3, for 20 and 40 search items respectively, and 500 iterations. In our work, we relied on the MATLAB R2021a program. The hybridization efficiency (SCSO - DMO) and (SCSO - CG) can also be observed in Figures 4-8, where the dark red color indicates (SCSO - DMO). The light red color indicates (SCSO - CG). Blue color indicates (DMO), while green color indicates (SCSO). In addition to the colors that indicate contemporary algorithms (MFO, GWO, ZOA, MGO).

**Table 1.** Introducing standard benchmark test function (unimodal, multimodal) that is used to assess the efficiency of algorithms

$F_n$	Function Name	Function	D	Range	$f_{min}$
$F_1$	Sphere	$\sum_{i=1}^n x_i^2$	30	[-100,100]	0
$F_2$	Schwefe 12.22	$\sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	30	[-10,10]	0
$F_3$	Schwefel 1.2	$\sum_{i=1}^n \left( \sum_{j=1}^i x_j \right)^2$	30	[-100,100]	0
$F_4$	Schwefel 2.21	$\max_i \{ x_i , 1 \leq i \leq n\}$	30	[-100,100]	0
$F_5$	Rastrigin	$\sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	[-5 · 12,5 · 12]	0

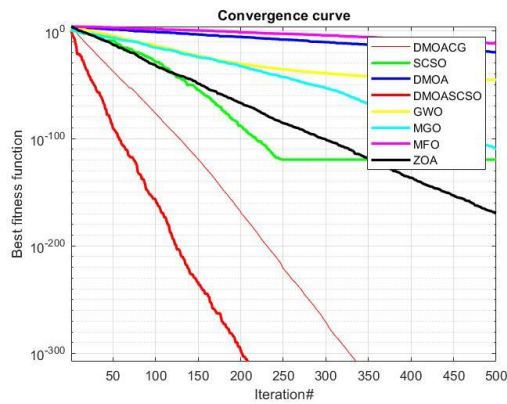
**Table 2.** Comparison outcomes of SCSO, DMO,CG-DMO and SCSO-DMO using the number of elements along with 20 elements and the number of Iterations 500. In addition to a comparison with some contemporary algorithms (MFO, GWO, ZOA, MGO)

Function Symbol	MFO [21]	GWO [22]	ZOA [23]	MGO [24]	SCSO	DMO	CG-DMO	SCSO-DMO
$F_1$	4.9375e-13	2.1672e-23	2.7909e-248	1.3478e-64	2.2387e-118	3.3605e-19	0	0
$F_2$	2.367e-09	5.5197e-14	1.2354e-134	1.0463e-39	2.9216e-59	9.0712e-14	1.6552e-247	0
$F_3$	0.02506	6.1997e-06	7.841e-155	6.6492e-07	3.5931e-104	0.74749	0	0
$F_4$	0.2942	4.3426e-06	4.6546e-115	2.499e-23	1.7584e-53	0.0084269	8.227e-201	0
$F_5$	17.9092	7.7541	0	0	0	13.7479	0	0

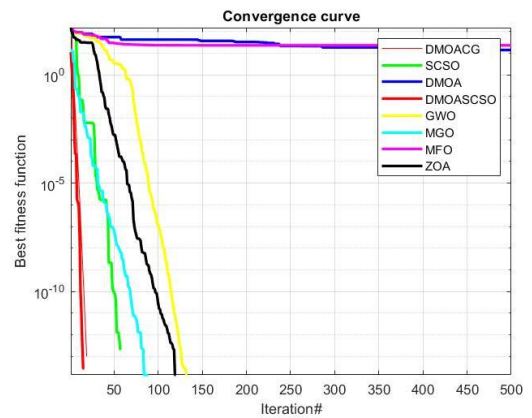
**Table 3.** Comparison outcomes of SCSO, DMO, CG-DMO and SCSO-DMO using the number of elements along with 40 elements and the number of iterations 500. In addition to a comparison with some contemporary algorithms (MFO, GWO, ZOA, MGO)

Function Symbol	MFO [18]	GWO [19]	ZOA [20]	MGO [21]	SCSO	DMO	CG-DMO	SCSO-DMO
$F_1$	7.8551e-15	5.5123e-30	1.8334e-259	1.6377e-80	6.6202e-125	1.7882e-22	0	0
$F_2$	2.4263e-09	1.8844e-18	1.3374e-137	1.0374e-45	6.0859e-64	2.4793e-15	4.7298e-252	0
$F_3$	0.00050369	7.7464e-10	1.7967e-168	2.5048e-14	9.9133e-110	0.1238	0	0
$F_4$	0.015684	4.5393e-08	1.562e-116	3.0226e-28	7.6933e-54	0.0021931	3.2201e-205	0
$F_5$	13.9294	1.7053e-13	0	0	0	10.7216	0	0

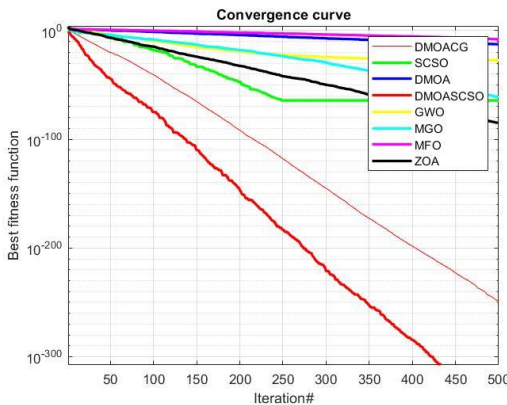




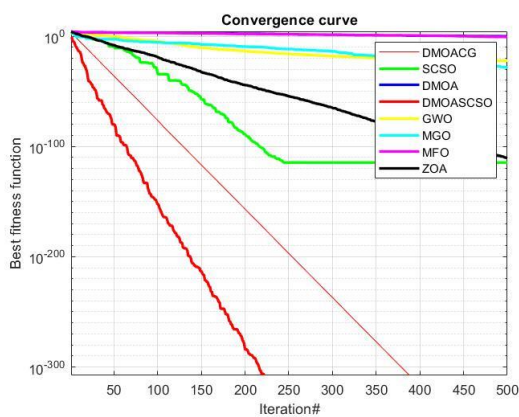
**Figure 4.** Function graph for  $F_1$



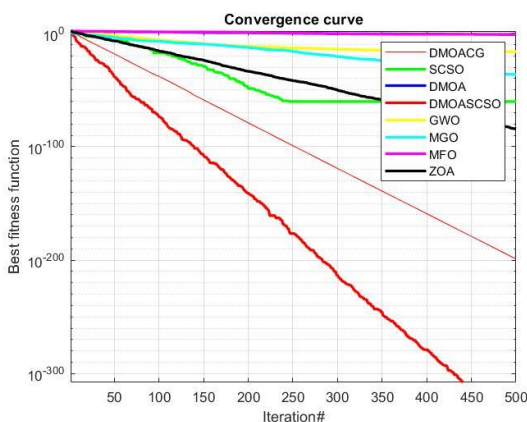
**Figure 8.** Function graph for  $F_5$



**Figure 5.** Function graph for  $F_2$



**Figure 6.** function graph for  $F_3$



**Figure 7.** function graph for  $F_4$

## 6. CONCLUSIONS

The hybridization of the (DMO) Algorithm has given rise to two new hybrid algorithms (SCSO-DMO) and (CG-DMO). These two algorithms have several properties, including their ability to work with complex, multi-dimensional problems. And the efficiency of their speed compared to the speed of the two original algorithms, so that the two proposed algorithms (SCSO-DMO) and (CG-DMO) gave better numerical results than the results of the two original algorithms with less time and effort, as can be seen from Tables 2 and 3 above for five of the standard test jobs. Which shows comparative results with 500 iterations and 30 and 60 search items, respectively.

Through these features, the two proposed algorithms (SCSO-DMO) and (CG-DMO) can be exploited to solve many problems, including improving performance, reducing cost, and improving engineering and mathematical design.

Thanks to these advantages, the two proposed algorithms (SCSO-DMO) and (CG-DMO) can be used to solve a wide range of problems in different fields.

## REFERENCES

- [1] Dehghani, M., Trojovská, E., Trojovský, P. (2022). A new human-based metaheuristic algorithm for solving optimization problems on the base of simulation of driving training process. *Scientific Reports*, 12(1): 9924. <https://doi.org/10.1038/s41598-022-14225-7>
- [2] Hopper, E., Turton, B.C.H. (2001). An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem. *European Journal of Operational Research*, 128(1): 34-57. [https://doi.org/10.1016/S0377-2217\(99\)00357-4](https://doi.org/10.1016/S0377-2217(99)00357-4)
- [3] Peng, X., Chen, H., Guan, C. (2023). Energy management optimization of fuel cell hybrid ship based on particle swarm optimization algorithm. *Energies*, 16(3): 1373. <https://doi.org/10.3390/en16031373>
- [4] Xie, Q.Y., Guo, Z.Q., Liu, D.F., Chen, Z.S., Shen, Z.L., Wang, X.L. (2021). Optimization of heliostat field distribution based on improved gray wolf optimization algorithm. *Renewable Energy*, 176: 447-458. <https://doi.org/10.1016/j.renene.2021.05.058>
- [5] Hu, Q., Tian, Y.Q., Qi, H.Q., Wu, P., Liu, Q.X. (2023). Optimization method for cloud manufacturing service composition based on the improved artificial bee colony

- algorithm. *Journal on Communication/Tongxin Xuebao*, 44(1): 200-210. <https://doi.org/10.11959/j.issn.1000-436x.2023024>
- [6] Yang, J., Cui, X.R., Li, J., Li, S.B., Liu, J.H., Chen, H.H. (2021). Particle filter algorithm optimized by genetic algorithm combined with particle swarm optimization. *Procedia Computer Science*, 187: 206-211. <https://doi.org/10.1016/j.procs.2021.04.052>
- [7] Ezugwu, A.E., Adeleke, O.J., Akinyelu, A.A., Viriri, S. (2020). A conceptual comparison of several metaheuristic algorithms on continuous optimisation problems. *Neural Computing and Applications*, 32: 6207-6251. <https://doi.org/10.1007/s00521-019-04132-w>
- [8] Oyelade, O.N., Ezugwu, A.E. (2021). Ebola Optimization Search Algorithm (EOSA): A new metaheuristic algorithm based on the propagation model of Ebola virus disease. *ArXiv*, 2106. 01416. <https://doi.org/10.48550/arXiv.2106.01416>
- [9] Zheng, R., Jia, H.M., Abualigah, L., Liu, Q.X., Wang, S. (2022). An improved arithmetic optimization algorithm with forced switching mechanism for global optimization problems. *Mathematical Biosciences and Engineering*, 19(1): 473-512. <https://doi.org/10.3934/mbe.2022023>
- [10] Nadimi-Shahraki, M.H., Fatahi, A., Zamani, H., Mirjalili, S., Abualigah, L. (2021). An improved moth-flame optimization algorithm with adaptation mechanism to solve numerical and mechanical engineering problems. *Entropy*, 23(12): 1637. <https://doi.org/10.3390/e23121637>
- [11] Agushaka, J.O., Ezugwu, A.E., Abualigah, L. (2022). Dwarf mongoose optimization algorithm. *Computer Methods in Applied Mechanics and Engineering*, 391: 114570. <https://doi.org/10.1016/j.cma.2022.114570>
- [12] Talbi, E.G. (2009). *Metaheuristic: From Design to Implementation*. John Wiley & Sons. <https://doi.org/10.1002/9780470496916>
- [13] Seyyedabbasi, A., Kiani, F. (2023). Sand cat swarm optimization: A nature-inspired algorithm to solve global optimization problems. *Engineering with Computers*, 39: 2627-2651. <https://doi.org/10.1007/s00366-022-01604-x>
- [14] Beale, E.M.L. (1988). *Introduction to Optimization*. Wiley-Interscience.
- [15] Babaie-Kafaki, S., Ghanbari, R. (2014). The Dai–Liao nonlinear conjugate gradient method with optimal parameter choices. *European Journal of Operational Research*, 234(3): 625-630. <https://doi.org/10.1016/j.ejor.2013.11.012>
- [16] Ibrahim, A.L., Shareef, S.G. (2020). A new class of three-term conjugate gradient methods for solving unconstrained minimization problems. *General Letters in Mathematics*, 7(2): 79-86. <https://doi.org/10.31559/glm2019.7.2.4>
- [17] Al-Bayati, A.Y., Hassan, S.K. (2011). New extended Polak-Ribiere CG-methods for nonlinear unconstrained optimization. *Canadian Journal on Science and Engineering Mathematics*, 2(1): 9-19.
- [18] Cole, F.R., Wilson, D.E. (2015). *Felis margarita* (Carnivora: Felidae). *Mammalian Species*, 47(924): 63-77. <https://doi.org/10.1093/mspecies/sev007>
- [19] Huang, G., Rosowski, J., Ravicz, M., Peak, W. (2002). Mammalian ear specializations in arid habitats: structural and functional evidence from sand cat (*Felis margarita*). *Journal of Comparative Physiology A*, 188: 663-681. <https://doi.org/10.1007/s00359-002-0332-8>
- [20] Chou, J.S., Truong, D.N. (2021). A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. *Applied Mathematics and Computation*, 389: 125535. <https://doi.org/10.1016/j.amc.2020.125535>
- [21] Ceylan, O., Paudyal, S. (2017). Optimal capacitor placement and sizing considering load profile variations using moth-flame optimization algorithm. In *2017 International Conference on Modern Power Systems (MPS)*, Cluj-Napoca, Romania, pp. 1-6. <https://doi.org/10.1109/MPS.2017.7974468>
- [22] Nadimi-Shahraki, M.H., Taghian, S., Mirjalili, S. (2021). An improved grey wolf optimizer for solving engineering problems. *Expert Systems with Applications*, 166: 113917. <https://doi.org/10.1016/j.eswa.2020.113917>
- [23] Trojovská, E., Dehghani, M., Trojovský, P. (2022). Zebra optimization algorithm: A new bio-inspired optimization algorithm for solving optimization algorithm. *IEEE Access*, 10: 49445-49473. <https://doi.org/10.1109/ACCESS.2022.3172789>
- [24] Seini, T., Yussif, A.F.S., Katali, I.M. (2023). Enhancing mountain gazelle optimizer (MGO) with an improved f-parameter for global optimization. *International Research Journal of Engineering and Technology*, 10(6): 921-930.