DOI: 10.24996/ijs.2024.65.7.24

ISSN: 0067-2904

A hybrid The Dwarf Mongoose Optimization Algorithm with Nelder- Mead method and its application for allocation reliability

Roaa aziz Fadhil*, Zahir Abdul Haddi Hassan

Department of Mathematics,College of Education for Pure Sciences University of Babylon

Received: 5/12/2022 Accepted: 2/3/2023 Published: 30/7/2024

Abstract

The Dwarf Mongoose Optimization Algorithm (DMO) is inspired by the behaviour of Dwarf Mongoose which can strike the ideal balance throughout research between exploration and exploitation. In this article, we combine algorithms of the Dwarf Mongoose Optimization Algorithm and the Nelder-Mead Algorithm (DMONM). In addition, the statistically evaluated functions is utilized by calculating the average and the standard deviation values that are used to validate the suggested algorithm's performance. The experimental results are on high-efficiency optimization functions with various dimensions. The hybrid algorithm produces good, encouraging, and better outcomes than the original algorithms. The results show that the proposed algorithm could enhance the effects of DMO when it used to solve the optimization issues of the multi-objective reliability system

Keywords: dwarf mongoose optimization algorithm, Nelder-Mead Algorithm.

هجين خوارزمية تحسين النمس القزم مع طريقة نيلدر– ميد وتطبيقها في تخصيص الموثوقية

رؤى عزيز فاضل *, زاهر عبد الهادي

قسم الرياضيات, كلية التربية للعلوم الصرفة, جامعة بابل , العراق

الخلاصة

خوارزمية تحسين النمس القزم (DMO) ، المستوحاة من سلوك Dwarf Mongoose ، تحقيق التوازن المثالي خلال البحث بين الاستكشاف والاستغلال. في هذه المقالة ، نقوم بدمج خوارزميات خوارزمية Dwarf Mongoose Optimization مع خوارزمية Nelder-Mead (DMONM). تم استخدام الدوال المقيمة إحصائياً من خلال حساب المتوسط وقيم الانحراف المعياري للتحقق من أداء الخوارزمية المقترحة. كانت النتائج التجريبية على وظائف التحسين عالية الكفاءة ذات الأبعاد المختلفة. أنتجت الخوارزمية الهجينة نتائج جيدة ومشجعة وأفضل من الخوارزميات الأصلية. أظهرت النتائج أن الخوارزمية المقترحة يمكن أن تعزز تأثيرات DMOعند استخدامها لحل مشكلات التحسين لنظام الموثوقية متعدد الأهداف .

1. Introduction

Recently, many authors have used many meta-heuristic algorithms in various applications to handle different optimization problems [1] [2] [3]. The straightforward research technique offered by Nelder and Mead (1965) [4]. It is a derivative-free technique to find local search. This technique involves to applying four fundamental operators to remeasure the single

*Email: roaaazizfadhil@gmail.com

information on the local behaviour [5], and it broadly uses the Nelder-Mead algorithm (NM) for optimizing meta-heuristics. Condensation is a technique for speeding up the search and overcoming the algorithm's slow convergence, which is similar to a hybrid Nelder-Mead method and cuckoo search algorithm (HCSNM). The experimental findings demonstrate the effectiveness of the (HCSNM) algorithm and its superior capacity to resolve integer programming and mini(max) problems that are more quickly than other algorithms [6] to utilize two methods to enhance the bat algorithm (BA) performance for solving electrical engineering optimization issues. The first is based on applying the crossover technique to a conventional BA that is similar to the genetic algorithm method. The Nelder-Mead (NM) simplex method and the BA are combined in a second approach to produce the NM-BA algorithm. Improvement is therefore based on fusing traditional BA with NM. This combination seeks to speed up the optimization process using standard BA, and it improves the NM algorithm's exploitation stages to avoid trapping in a local extremum [7]. The Nelder-Mead algorithm is employed to solve an optimization problem for a structural design. The hybrid marine predators and Nelder-Mead algorithm (HMPANM) are used to enhance the local exploitation capabilities of the marine predator's algorithm (MPA). The outcomes unequivocally demonstrate the HMPANM's capacity for the best component design in the automotive sector, where the hybrid marine predator optimization algorithm is applied for structural optimization of the vehicle component. The outcomes demonstrate that the hybrid marine predator's optimization algorithm produces superior effects versus other techniques [8]. A hybrid algorithm for power system optimization is a reactive power dispatch (ORPD) problem that combines the Firefly Algorithm (FA) and Nelder Mead (NM) simplex approach. A hybrid algorithm is used to find the generator voltage method's ideal settings instead of the original FA and other existing techniques. This algorithm has improved convergence characteristics and resilience. It is demonstrated that the hybrid approach can deliver more effective solutions [9]. The multi-objective system reliability optimization is a significant in the industry which becomes more than ever [10] [11]. The purpose of this paper is to develop the Dwarf Mongoose Optimization Algorithm (DMO) [12] with the Nelder-Mead algorithm [4] [5] that proposed hybrid Dwarf Mongoose Optimization Nelder Mead algorithm (DMONM). The effectiveness of (DMONM) is verified by the experimental results of statistical analysis for optimization problems. Reveals (DMONM) is superior to DMO, Multi-objective system reliability optimization, due to its importance in the industry. Optimization has multiple objectives, such as maximizing reliability and minimizing cost. Optimization is presented for multi-objective system reliability optimization and ensuring diversity in exploring the search space [11] [13]. In this research, we emphasize on improving the numerical results obtained for the DMONM algorithm compared to the original algorithm. This paper is organized as follows. In section 2, we provide basic facts for the Dwarf Mongoose optimization algorithm and the Nelder-Mead method. In Section 3, we present the proposed algorithm. In Section 3, we mention the types of test functions. In Section 4, we show the statistical methods results for test functions. In Section 4, we apply the algorithms to improve network reliability.

1.1 Preliminaries

1.1.1. The Dwarf Mongoose Optimization Algorithm

The initial design of the Dwarf Mongoose Optimization Algorithm (DMO) can be found in [12] . The proposed DMO imitates DMO compensating behaviour. We apply the next formula and we start to determine a starting value for the set of solutions:

$$x_{i.j} = l_j + rand \times (u_j - l_j) . \tag{1}$$

Where the $rand$ is a random number in [0, 1] . The search domain's boundaries are u_j and l_j .The DMO is made up of three groups: the Alpha Group, Scouts, and Babysitters. To catch the food, each group employs a distinctive method. These specific groups are modelled in the next manner as follows:

1.1.1.1. Alpha Group

After the population is established, each solution's fitness is calculated. According to Eq. (2), each population's fitness probability value is established, and this likelihood is used to determine the alpha female (α).

$$\alpha = \frac{f_i\, t_i}{\sum_{i=1}^{n} f_i\, t_i} \quad . \qquad ... (2)$$

The $n - b_s$ is as many as the alpha group of mongooses has members. where b_s represents the number of nannies. Peep is the dominant female vocalisation that keeps the family on track. The first sleeping mound, which is located at, is every mongoose sleeps in the initial sleeping mound, which is set at ∅. The DMO selects a candidate for a food role using Eq (3).

$$\chi_{i+1} = \chi_i + ph_i * peep \, . \qquad \dots (3)$$

The value ph_ihas a uniform distribution and falls between [-1,1]. Eq. (4) provides the sleeping mound that follows each repeat.

$$sm_i = \frac{f_i t_{i+1} + f_i t_i}{\max\{|f_i t_{i+1} . f_i t_i|\}} \, . \qquad ... (4)$$

Eq. (5) provides the average number of the discovered sleeping mounds.

$$\varphi = \frac{\sum_{i=1}^{n} sm_i}{n} \quad . \qquad ... (5)$$

1.1.1.2. Scout Group

The algorithm is advanced to the scouting phase if the prerequisite for a babysitting exchange is satisfied as well as once the condition for a childcare swap is met, it analyses the next food source or sleeping mound. Mongooses are known to avoid old sleeping mounds, thus scouts search for the next one to ensure exploration. The manner of moving depends on whether he successfully locates a new sleeping mound in our model, which combines foraging and reconnaissance. If they are wander far enough, the family will find a new sleeping mound. Equation and also serve as representations of the scout mongoose (6).

$$\chi_{i+1} = \begin{cases} \chi_i - cf * ph_i * rand\, [\chi_i - \vec{\mu}]. if\ \varphi_{i+1} > \varphi_i \\ \chi_i + cf * ph_i * rand\, [\chi_i - \vec{\mu}]. otherewise \end{cases} \qquad \dots (6)$$

where $rand \in [0.1]$, Eq. (7) is used to compute cf value while Eq. (8) is used to calculate $\vec{\mu}$ value.

$$cf = (1 - \frac{iter}{\max iter})^{(2 * \frac{iter}{\max iter})} \quad , \qquad ... (7)$$

$$\vec{\mu} = \sum_{i=1}^{n} \frac{\chi_i * sm_i}{\chi_i} \quad . \qquad ... (8)$$

Babysitters are often lesser group members who look after the children and they are frequently rotated, so the alpha female can oversee the daily hunting excursions of the group.

Algorithm: Pseudo-code of the DMO

Step 1 : Input: Set the requirements and solutions of the algorithm.
Step 2 : Initialize the algorithmic parameters settings and solution.
Step 3 : For iter=1: max_iter
Step 4 : Determine the Mongoose Fitness Function..
Step 5 : Establish a timer (C).
Step 6 : Using Eq.(2) to determine the alpha value.
Step 7 : Using Eq.(3) to locate a potential food position.
Step 8 : Estimate the new fitness χ_{i+1}.
Step 9 : Calculate the average value for the sleeping mound as it is determined by Eq. (4).
Step 10 : Eq.(5)can be used to calculate the average mound sleeping.
Step 11 : Eq. (8) can be utilized to determine the movement vector.
Step 12 : Based on the Equation, simulate the next location of the scout mongoose (6).
Step 13 : end for.
Step 14 : $\tau = \tau + 1$.
Step 15 : end while.
Step16 : Output: Return the best solution (χ).

1.1.2. Nelder-Mead method

The Nelder-Mead simplex method is frequently employed to identify local minimum solutions if the derivative is unknown for well-defined problems. The fundamental building block algorithm is the possibility of transformation reflection, expansion, contraction, and shrinkage. These are the steps that make up the NM simplex algorithm [4] [5] .

Step1: Compute trial steps. In all iterations. First, all the vertices Order $n + 1$ depending on the objective function value to satisfy

$$f(x_1) \leq f(x_2) \leq \cdots \leq f(x_{n+1}).$$

Step2: Reflection Calculate the reflection point x_r from

$$x_r = x_c + \lambda(x_c - x_{n+1}), \quad \ldots(9)$$

Where

$$x_c = \frac{1}{n}\sum_{\substack{i=1 \\ i \neq worst}}^{n+1} x_i \quad \ldots(10)$$

the centroid of every point with exception of x_{n+1} , if $f(x_1) \leq f(x_r) < f(x_n)$, accept the reflected point x_r and end the iteration

Step3: Expansion If $f(x_r) < f(x_1)$, then it is calculated that the expanded point x_e is

$$x_e = x_c + \beta(x_r - x_c) \quad \ldots(11)$$

and If $f(x_e) < f(x_r)$, then accept x_e and end the iteration otherwise $f(x_r) < f(x_e)$, accept x_r and end the iteration

Step 4: Contract. If $(x_r) \geq f(x_n)$. A contraction takes place. Two contractions are conceivable.

a. Outside. $f(x_n) \leq f(x_r) < f(x_{n+1})$, contraction by the formulae (12)

$$x_{con} = x_c + \sigma(x_r - x_c) \;.\; 0 \leq \sigma \leq 1, \quad \ldots(12)$$

and If $f(x_{con}) < f(x_r)$, accept x_{con} and end the iteration , otherwise go to calculate a shrink step.

b. Inside. If $f(x_r) \geq f(x_{n+1})$ calculate inside contraction

$$x_{cont} = x_c - \sigma(x_c - x_{n+1}) \;\; 0 \leq \sigma \leq 1, \quad \ldots(13)$$

and If $f(x_{cont}) < f(x_{n+1})$, then accept x_{cont} and end the iteration , otherwise go to calculate a shrink step.
Step5: A shrink step calculates the shrink by the formulae

$$v_i = x_1 + \delta(x_i - x_1).\ 0 < \delta < 1 \quad , \quad i = 2. \dots .n + 1 \qquad \dots(14)$$

2. The proposed DMONM algorithm

The proposed Dwarf Mongoose Optimization Nelder Mead algorithm (DMONM) follows the same steps as the traditional Dwarf Mongoose Optimization Algorithm (DMO). Then it is used to improve the best result from the previous step of the DMO algorithm, the obtained solution from the Dwarf Mongoose Optimization Algorithm is applied to the Nelder-Mead algorithm for the same iteration.

3. Benchmark Functions

To evaluate the effectiveness of the suggested Dwarf Mongoose Optimization-Mead algorithm (DMONM) using different test functions. Thirteen benchmark functions have been employed for unimodal and multimodal. The purpose of the unimodal test functions $\{f_1 - f_7\}$ in Table (1) tests the exploitation capacity of the algorithm because they have one optimum limit. Multimodal functions $\{f_8 - f_{13}\}$ are shown in Table (2). There are many locally optimal solutions for multimodal functions. So the optimization algorithms need to have a lot of exploring power 30 and 50 dimensions that are used to test these two classes of functions. The analysis has been performed on MATLAB 2019, and it describes the parameters settings that are employed in the experimentation$\lambda = 2. \beta = 3. \gamma = 0.01. \delta = 0.5$. The iterations number is 500 iterations.

Table 1: Unimodal test functions.

Objective function	Dimensions	Range
$f_1(\chi) = \sum_{i=1}^{m} \chi_i^2$	30,50	[-10,10]
$f_2(\chi) = \sum_{i=1}^{m} \lvert\chi_i\rvert + \prod_{i=1}^{m} \lvert\chi_i\rvert$	30,50	[-10,10]
$f_3(\chi) = \sum_{i=1}^{m} \left(\sum_{j=1}^{i} \chi_i \right)^2$	30,50	[-100,100]
$f_4(\chi) = \max \{ \lvert\chi_i\rvert, I \le i \le m \}$	30,50	[-12,12]
$f_5(\chi) \sum_{i=1}^{m-1} [100(\chi_{i+1} - \chi_i^2)^2 + (\chi_i - 1)^2]$	30,50	[-30,30]
$f_6(\chi) = \sum_{i=1}^{m} (\lvert\chi_i + 0.5\rvert)^2$	30,50	[-100,100]
$f_7(\chi) = \sum_{i=1}^{m} i\chi_i^4 + random(0.1)$	30,50	[-1,1]

Table 2: Multimodal test functions

Objective function	Dimensions	Range
$f_8(\chi) = \sum_{i=1}^{m} -\chi_i \sin(\sqrt{\lvert\chi_i\rvert})$	30,50	[−100.100]
$f_9(\chi) = \sum_{i=1}^{m} [\chi_i^2 - 10\cos(2\pi\chi_i) + 10]$	30,50	[−5.2]
$f_{10}(\chi) = -20\exp\left(-0.2\sqrt{\frac{1}{m}\sum_{i=1}^{m}\chi_i^2}\right) - \exp(\frac{1}{m}\sum_{i=1}^{m}\cos(2\pi\chi_i) + 20 + e$	30,50	[−10.10]
$f_{11}(\chi) = \frac{1}{4000}\sum_{i=1}^{m}\chi_i^2 - \prod_{i=1}^{m}\cos(\frac{\chi_i}{\sqrt{i}}) + 1$	30,50	[−17.17]
$f_{12}(\chi) = \frac{\pi}{m}\{10\sin(\pi y_i) + \sum_{i=1}^{m-1}(y_i - 1)^2 1 + 10 sin^2(\pi y_{i+1})$ $+(y_m - 1)^2\} + \sum_{i=1}^{n} u(\chi_i.10.100.4)$ $u(\chi_i .a.i.n) = \begin{cases} k(\chi_i - a)^n.\chi_i > -a \\ 0 \quad . \quad -a < \chi_i < a \\ k(-\chi_i - a)^n.\chi_i < -a \end{cases}$	30,50	[−13.13]
$f_{13}(\chi) = 0.1\{ sin^2(3\pi\chi_m) + \sum_{i-1}^{m}(\chi_i - 1)^2 [1]$ $+sin^2(3\pi\chi_i + 1) + (\chi_n - 1)^2$ $[1 + sin^2(2\pi\chi_m)] + \sum_{i=1}^{m} u(\chi_i.5.100.4)$	30,50	[−50.50]

4. Result and discussion

The performance of the Dwarf Mongoose Optimization Nelder Mead algorithm (DMONM) in several benchmark function classes using statistical methods is measured. The average (avg) and standard deviation (std) and performance comparison with algorithm Dwarf Mongoose Optimization Algorithm (DMO) from the experimental results are presented in Table (3), we find that the Dwarf Mongoose Optimization Nelder Mead algorithm (DMONM) is able with its best performance for the test functions in 30 dimensions. Table (4) demonstrates that the DMONM algorithm's results are superior to those obtained when DMO is evaluated in 50 dimensions. With the exception of the function (f_8) of performance, the DMO algorithm is superior to that of the suggested approach in both dimensions.

Table 3: Comparison Statistical results of DMONM and DMO algorithmson test functions with dim=50

function	DMO		DMONM	
	avg	Std	avg	Std
f_1	118.33848	21.37447	**0.15308**	0.15879
f_2	83.98244	13.33750	**0.96055**	0.73752
f_3	118908.55	12750.62	**1445.3617**	912.477
f_4	10.79053	0.37106	**1.59495**	0.28245
f_5	101255595.4	29520605.99	**6471.109**	5991.942
f_6	11716.42	2069.31149	**15.19601**	16.64612
f_7	10.77554	2.88337	**0.02407**	0.00724
f_8	**-1461.1541**	91.01792	-1487.955	109.278
f_9	529.69589	23.57921	**125.0746**	73.37372
f_{10}	7.15218	0.36453	**0.34426**	0.34302
f_{11}	1.08658	0.01409	**0.02129**	0.02141
f_{12}	1063.95	1166.061	**0.01954**	0.05176
f_{13}	210469911.7	66974355.38115	**18.33956**	21.37227

5. Application algorithms in allocation reliability

In order to create a highly reliable system by allocating greater component reliability and lower cost, it is crucial to raise the dependability of a multi-objective system. In this research, we obtain a system from the shutdown simplified modular Petri net system that is described in [14]. Conversion Petri nets in Figure (1)

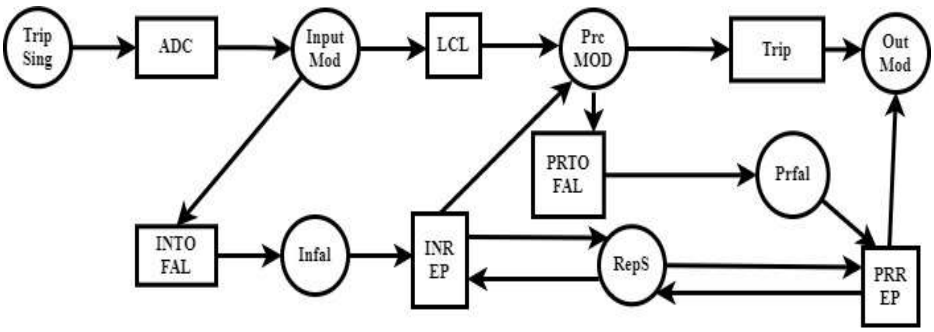

Figure 1: Simplifies modular Petri net [14]

Conversion Petri Nets as the network is turned into a graph in this instance places are replaced with nodes, and the transitions and their connecting arcs are replaced with a single edge [15]. We get the network that is shown in Figure (2)

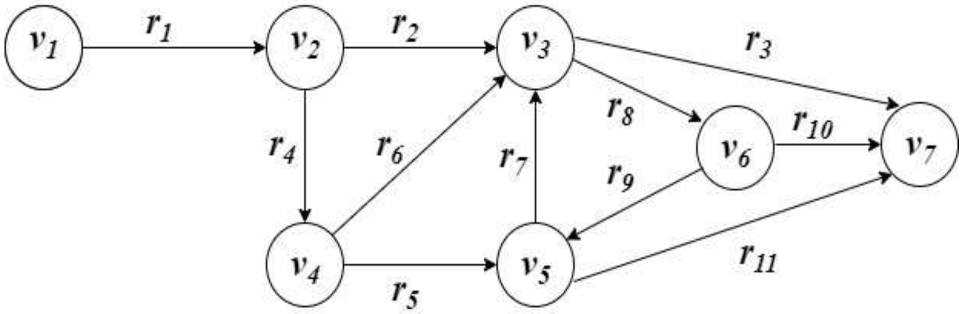

Figure 2: The network system.

We use the Sum-of-Disjoint Product [16] to calculate the reliability structure system in Eq.(15)

$$
\begin{aligned}
\boldsymbol{R_s} = {} & r_1r_2r_3 + r_1r_3r_4r_6 + r_1r_2r_8r_{10} + r_1r_4r_5r_{11} - r_1r_2r_3r_4r_6 + r_1r_3r_4r_5r_7 \\
& -r_1r_2r_3r_8r_{10} + r_1r_4r_6r_8r_{10} + r_1r_2r_8r_9r_{11} - r_1r_2r_3r_4r_5r_7 - r_1r_2r_3r_4r_5r_{11} \\
& -r_1r_3r_4r_5r_6r_7 - r_1r_3r_4r_5r_6r_{11} - r_1r_2r_4r_6r_8r_{10} - r_1r_3r_4r_5r_7r_{11} - r_1r_3r_4 \\
& r_6r_8r_{10} - r_1r_2r_3r_8r_9r_{11} + r_1r_4r_5r_7r_8r_{10} + r_1r_4r_6r_8r_9r_{11} - r_1r_2r_8r_9r_{10}r_{11} \\
& +r_1r_2r_3r_4r_5r_6r_7 + r_1r_2r_3r_4r_5r_6r_{11} + r_1r_2r_3r_4r_5r_7r_{11} + r_1r_2r_3r_4r_6r_8r_{10} - \\
& r_1r_2r_4r_5r_7r_8r_{10} + r_1r_3r_4r_5r_6r_7r_{11} - r_1r_3r_4r_5r_7r_8r_{10} - r_1r_2r_4r_5r_8r_9r_{11} - r_1 \\
& r_2r_4r_5r_8r_{10}r_{11} - r_1r_2r_4r_6r_8r_9r_{11} - r_1r_4r_5r_6r_7r_8r_{10} - r_1r_3r_4r_6r_8r_9r_{11} + r_1 \\
& r_2r_3r_8r_9r_{10}r_{11} + r_1r_4r_5r_6r_8r_9r_{11} - r_1r_4r_5r_7r_8r_{10}r_{11} - r_1r_4r_5r_6r_8r_{10}r_{11} - r_1 \\
& r_4r_6r_8r_9r_{10}r_{11} - r_1r_2r_3r_4r_5r_6r_7r_{11} + r_1r_2r_3r_4r_5r_7r_8r_{10} + r_1r_2r_3r_4r_5r_8r_9r_{11} \\
& +r_1r_2r_4r_5r_6r_7r_8r_{10} + r_1r_2r_3r_4r_5r_8r_{10}r_{11} + r_1r_2r_3r_4r_6r_8r_9r_{11} + r_1r_3r_4r_5r_6r_7 \\
& r_8r_{10} + r_1r_2r_4r_5r_6r_8r_9r_{11} + r_1r_2r_4r_5r_6r_8r_{10}r_{11} + r_1r_3r_4r_5r_6r_8r_9r_{10}r_{11} + r_1r_2 \\
& r_4r_5r_7r_8r_{10}r_{11} + r_1r_3r_4r_5r_6r_8r_{10}r_{11} + r_1r_3r_4r_5r_7r_8r_{10}r_{11} + r_1r_2r_4r_6r_8r_9r_{10}r_{11} \\
& +r_1r_2r_4r_6r_7r_8r_9r_{10}r_{11} + r_1r_3r_4r_6r_8r_9r_{10}r_{11} + r_1r_4r_5r_6r_7r_8r_{10}r_{11} + r_1r_4r_5r_6r_8 \\
& r_9r_{10}r_{11} - r_1r_2r_3r_4r_5r_6r_7r_8r_{10} - r_1r_2r_3r_4r_5r_6r_8r_9r_{11} - r_1r_2r_3r_4r_5r_6r_8r_{10}r_{11} - \\
& r_1r_2r_3r_4r_5r_7r_8r_{10}r_{11} - r_1r_2r_3r_4r_5r_8r_9r_{10}r_{11} - r_1r_2r_3r_4r_6r_8r_9r_{10}r_{11} - r_1r_2r_4r_5r_6 \\
& r_7r_8r_{10}r_{11} - r_1r_3r_4r_5r_6r_7r_8r_{10}r_{11} - r_1r_2r_4r_5r_6r_8r_9r_{10}r_{11} - r_1r_3r_4r_5r_6r_8r_9r_{10}r_{11} \\
& +r_1r_2r_3r_4r_5r_6r_7r_8r_{10}r_{11} + \\
& r_1r_2r_3r_4r_5r_6r_8r_9r_{10}r_{11} \qquad \ldots(15)
\end{aligned}
$$

5.1. Mathematical model for multi-objective system reliability optimization

The following can be used to express the multi-objective problem [11] [13]

$$\max R_s(r_i) \qquad \text{for } i = 1.2.\ldots.11$$

$$\min C_s(r_i) = \sum_{i=1}^{11} a_i \exp\left(\frac{b}{1-x_i}\right)$$

subject to $0.95 \le R_s \le 0.9999$

$0.6 \le r_i \le 0.9999$ for $i = 1.2.\ldots.11$

Where $a = 0.01. b = 0.03 for\ i = 1.2.\ldots.11$

R_s is represented by Eq.(15)

5.2. Numerical case study

Multiple techniques can be used to solve problems involving multi-objective optimization. The weighted-sum approach reduces a multi-objective problem to a single-objective problem by giving weights to each function. The constraint handling is done via a penalty function [17] [14] [18].

$$\min f(r_i) = \mu_1 C_s - \mu_2 R_s + \alpha\,(r_i)$$

Where $\mu_1 = \mu_2 = 0.5$ and $\alpha\ (r_i)$ is the penalty function

$$\alpha\ (r_i) = \alpha_1 \max(0.9999 - R_s) + \alpha_2 \max\ (0. R_s - 0.95)$$

Where $\alpha_1 . \alpha_2$ is the penalty factor, we find out the best reliability of the system by using DMONM and DMO algorithms. The number of iterations is 500 which is used to describe the parameters settings that are employed $\lambda = 2. \beta = 3. \gamma = 0.01. \delta = 0.005$, The results values of components reliability are r_i and the cost components are $C_i$with the best value of reliability system is R_s and the total cost is C_s in Table (5).

Table 5: Comparison of DMONM and DMO algorithms for results values components

Components	DMO	DMONM
	Value of r_i	Value of r_i
r_1	0.9915	**0.9917**
r_2	0.9799	**0.9805**
r_3	0.9696	**0.9724**
r_4	0.9795	**0.9804**
r_5	**0.9586**	0.9583
r_6	**0.9397**	0.9379
r_7	**0.7696**	0.7516
r_8	**0.9595**	0.9570
r_9	0.8246	**0.8336**
r_{10}	0.9484	**0.9496**
r_{11}	**0.9607**	0.9590
Total R_s	**0.9908**	**0.9911**
Total C_s	**0.5773**	**0.6093**

Through Table(5), we can make the following observations**:** At values using a DMO algorithm are $0.7696 \leq r_i \leq 0.9915$, for all *i* = 1, 2,…,11 and the values using the HBANM algorithm are $0.7516 \leq r_i \leq 0.9917$, for all *i* = 1,2,…,11, we notice an improvement in the total value $\boldsymbol{R_s}$ (0.9911) in DMO**NM** algorithm compared to the total value $\boldsymbol{R_s}$of the DMO algorithm (0.9908) and the suggested algorithm have been improved six components are $(\boldsymbol{r_1 . r_2 . r_3 . r_4 . r_9 . r_{10}})$ compared to the value of the DMO algorithm.

6. Conclusions

The Dwarf Mongoose Optimization Algorithm (DMO) and Nelder-Mead algorithm (DMONM) and Nelder-Mead algorithm are combined in this study to form a hybrid Dwarf Mongoose Optimization Nelder Mead algorithm (DMONM), the suggested algorithm improves the best result from the DMO to confirm the robustness and efficiency of the average and standard deviation that are used to examine its effectiveness. We also discover that the DMONM significantly improves the majority of the functions. The DMONM algorithm was used to improve a multi-objective reliability system. In addition, the results show that they are better than the DMO algorithm.

References

[1] I. a. F. K. Naruei, "A new optimization method based on COOT bird natural life model," *Expert Systems with Applications*, p. 115352, 2021.

[2] A. S. a. R. D. A.-D. Khalil, "A Comparative Study on Meta-Heuristic Algorithms For Solving the RNP Problem," *Iraqi Journal of Science*, pp. 1639-1648, 2019.

[3] H. A. Chachan, "Using Non-dominated Sorting Particle Swarm Optimization Algorithm II for Bi-objective Flow Shop Scheduling Problems," *Iraqi Journal of Science,* pp. 275-288, 2021.

[4] J. A. N. a. R. A. Mead, "A simplex method for function minimization," *The computer journal,* pp. 308-313, 1965.

[5] R. R. a. J. S. I. J. Barton, "Modifications of the Nelder-Mead simplex method for stochastic simulation response optimization," *Institute of Electrical and Electronics Engineers*, 1991.

[6] A. F. a. M. A. T. Ali, "A hybrid cuckoo search algorithm with Nelder Mead method for solving global optimization problems," *SpringerPlus,* pp. 1-22, 2016.

[7] B. B. Z. a. H. T. Kheireddine, "Improvements of bat algorithm using crossover technique and hybridization with Nelder-Mead simplex method," *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering,* 2019.

[8] N. e. a. Panagant, "A novel hybrid marine predators-Nelder-Mead optimization algorithm for the optimal design of engineering problems," *Materials Testing,* vol. 63, no. 5, pp. 453-457, 2021.

[9] A. a. T. M. Rajan, "Optimal reactive power dispatch using hybrid Nelder–Mead simplex based firefly algorithm," *International Journal of Electrical Power & EnergySystems,* pp. 9-24, 2015.

[10] Abed, Saad Abbas, Hatem Kareem Sulaiman, and Zahir AbdulHaddi Hassan, "Reliability Allocation and Optimization for (ROSS) of a Spacecraft by using Genetic Algorithm," *Journal of Physics: Conference Series,* vol. No3, 2019.

[11] M. A. a. E. Z. Mellal, "An adaptive particle swarm optimization method for multi-objective system reliability optimization.," *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability,* vol. 233, no. 6, pp. 990-1001, 2019.

[12] J. O. A. E. E. a. L. A. Agushaka, "Dwarf mongoose optimization algorithm," *Computer methods in applied mechanics and engineering,* p. 114570, 2022.

[13] V. B. S. N. M. a. J. R. Ravi, "Nonequilibrium simulated-annealing algorithm applied to reliability optimization of complex systems," *EEE Transactions on Reliability,* vol. 46, no. 2, pp. 233-239, 1997.

[14] D. e. a. Zhang, "Reliability Modeling and Analysis of Reactor Protect System Based on Petri Net," *Journal of Physics: Conference Series,* vol. Vol. 1754, no. No. 1, 2021.

[15] T. Spiteri Staines, "Rewriting petri nets as directed graphs," 2011.

[16] S. K. Chaturvedi, Network reliability: measures and evaluation, John Wiley & Sons, 2016.

[17] M. A. a. E. Z. Mellal, "A penalty guided stochastic fractal search approach for system reliability optimization," *Reliability Engineering & System Safety,* pp. 213-227, 2016.

[18] R. A. a. C. S. N. Sarker, Optimization modelling: a practical approach, CRC press, 2007.

[19] R. T. a. J. S. A. Marler, "Survey of multi-objective optimization methods for engineering," *Structural and multidisciplinary optimization,* vol. 6, pp. 369-395, 2004.