Narwhal Optimizer: A Novel Nature-Inspired Metaheuristic Algorithm

Seyyid Ahmed Medjahed
Department of Computer Science
University of Relizane, Algeria
seyyidahmed.medjahed@univ-relizane.dz

Fatima Boukhatem
Faculty of Science and Technology
University of Djilali Liabes, Algeria
fatima.boukhatem@univ-sba.dz

Abstract: *Recently, metaheuristic algorithms have become a very interesting research field due to their ability to address complex and diverse problems. This paper presents a novel metaheuristic called Narwhal Optimizer (NO) inspired by narwhals behaviors. The NO algorithm mimics the hunting mechanism of narwhals. The narwhals are marine mammals known for their sophisticated communication based on clicks sound to locate their prey. The algorithm is based on three main steps: signal emission, signal propagation, and position updating of the narwhals. The hunting process, which is based on signal emission and propagation, is formulated as an optimization algorithm. The strategies observed in narwhal pods are emulated to enhance exploration and exploitation in the search space. The NO algorithm is benchmarked on 13 well-known functions, including unimodal, multimodal, and fixed-dimension multimodal functions. The experimental results showed that NO provides satisfactory and reasonable solutions in terms of avoiding local minima and achieving global optimality.*

Keywords: *Narwhal optimizer, meta-heuristic, optimization, nature-inspired algorithms, swarm intelligence.*

Received December 18, 2023; accepted April 21, 2024
https://doi.org/10.34028/iajit/21/3/6

1. Introduction

Optimization can be defined as the process of selecting a set of variables, called decision variables, that represent the best possible combination to solve a specific optimization problem. The final goal is to reach the maximum or minimum of an objective function or a multi-objective function. The process of finding a solution to an optimization problem is very crucial in any science or engineering field. That's why the demand for robust and efficient optimization algorithms is increasing [1].

In the last decade, many metaheuristic optimization algorithms have been developed. The most common ones include Genetic Algorithm (GA) [4], Particle Swarm Optimization (PSO) [15] and Ant Colony Optimization (ACO) [5]. Metaheuristics offer many advantages. The first one is that all these methods are very simple and the concept is inspired from physical phenomena, evolutionary concepts or behaviors of animals. The second advantage is their flexibility. Meta-heuristics can be applied to many real-world applications [6, 7, 12].

Metaheuristic algorithms can be categorized based on the type of the approach which can be nature-inspired or non-nature-inspired. It can be also categorized by the type of the objective function, such as mono-objective, multi-objective, dynamic objective, multi objective, dynamic objective or static objective [9, 24]. Generally, the nature-inspired algorithms are divided into four categories: Swarm Intelligence-based algorithms: these algorithms are inspired by the collective behavior of social organisms in the nature. For example, a group of agents interacts with them and their environments to reach an objective. We can cite: PSO [15], Ant Colony Optimization (ACO) [5], Bee Algorithms (BA), Walrus optimizer [3].

Evolutionary-based algorithms: This type of algorithm is based on the principles of natural evolution. They are inspired by the process of natural selection, reproduction, and genetic variation. An example of evolutionary based algorithms is: GA [4] and Differential Evolution (DE) [8].

Human-based algorithms: this type of algorithms is based on the human behavior. It is based specifically in the human intelligence, intuition, decision-making. Interactive Genetic Algorithm and Crowdsourcing for Optimization, are the most common algorithms.

Physics-based (PB) algorithms this type of algorithms is a class of optimization methods inspired by principles and phenomena from the field of physics. These algorithms often model physical processes or concepts to solve complex optimization problems. We cite: Simulated Annealing (SA) [10], Gravitational Search Algorithm (GSA) [23] and Harmony Search (HS) [13].

Optimization is used in many fields such as computer vision, machine learning, artificial intelligence, etc., [2 , 17].

In this paper, a new nature-inspired metaheuristic algorithm called Narwhal Optimizer (NO) is proposed. This algorithm is based on the behavior of narwhals in nature when they hunt. Narwhals use signals to locate their prey, which is a click sound. This sound is echolocation. The signal emission and signal propagation are mathematically modeled. Following these two processes, the position of narwhals is updated over iterations until they reach the prey which is the optimal solution.

The rest of the paper is organized as follows: Section 2 presents the model of the proposed NO. Section 3 presents the experimental results obtained by NO which are evaluated using 13 benchmark functions. Section 4, draw the conclusion.

2. Narwhal Optimizer (NO)

This section describes the NO, its inspiration and mathematical model.

2.1. Biological Fundamentals

The narwhal (Monodon Monoceros) is a fascinating marine mammal and is a medium sized toothed whale known for its long and spiral tusk. Narwhals are medium-sized compared to other whales (about 4 to 6 meters excluding their tusks). They are characterized by a mottled gray or brownish speckled skin that helps them blend in with the Arctic ice, as illustrated in Figure 1. They live year-round in the Arctic waters around Greenland in Canada, Norway and Russia. They have a bit ability to navigate through sea ice.

Figure 1. Narwhals in ocean.

2.1.1. Social Structure

The narwhals are composed of groups called “Pods”. The pods can range from a dozen individuals to hundred individuals. Groups can be “nurseries” composed of only females and young or they can be composed of juveniles and males. Also, a mixed group composed of male, female and young can occur at any time of year.

2.1.2. Migration

In summer season, the narwhals compose a Pods of range 10 to 100 and move closer to coasts. In winter season, they move to deeper waters under think pack ice.

2.1.3. Diet

The narwhals use their long tusks to stun fish before capturing and eating them. Their prey is composed of Arctic cod, polar, Greenland halibut.

2.1.4. Communication and Coordination

The narwhals use sound to hunt for food and to navigate. their vocalization which is composed of clicks, knocks and whistles which is an important wat for communication within the pod. The clicks are used for prey detection and for locating obstacles at short distances.

The communication with each Narwhal has a very important role to coordinate the group activities that includes hunting for prey. The narwhals use many kinds of vocalizations such as the clicks and whistles which are used of echolocation for navigation and finding prey. Also, the clicks which are a short plus of sound are used to identify the objects in the water including prey.

Narwhals travel and hunt in pods, using the communication, the narwhals can share information about the location of prey or other relevant details.
These vocalizations help the narwhals to coordinate their movements and update their position by ensuring that they stay as a group while searching and capturing prey. When narwhals locate prey, communication within the pod may intensify. Vocalizations could be used to signal the presence of prey, coordinate the attack, or share information about the location of potential threats.

Inspired from the behaviors of narwhals in communicating and hunting the prey, we propose a new meta-heuristic algorithm NO.

2.2. Mathematical Model

In NO, a solution is the location of a narwhal in the search space of the prey, which can be a potential solution of the optimization problem. When narwhals locate a potential prey, they communicate between them by sending a signal and the communication within the pod may intensify. The vocalization will be used to signal the presence of the prey and they share information about the location of the prey, also, they coordinate to prepare for the attack.

The process of locating their prey is based on echolocation to locate the prey. Echolocation is a technique in which narwhals emit a clicks in the water and listen to determinate the position of the prey.

2.2.1. Initialization

Initially, the optimization algorithm starts with a set of random solution which represents the locations of narwhals (X). In each iteration of the algorithm, these locations are updated continuously and is defined by the following matrix:

$$X = \begin{bmatrix} X_{1,1} & \cdots & X_{1,d} \\ \vdots & \ddots & \vdots \\ X_{n,1} & \cdots & X_{n,d} \end{bmatrix} \tag{1}$$

where n represents the population size of narwhals and d is the dimension of decision variables.

2.2.2. Signal Emission

When the narwhals navigate, they share their positions and try to locate the prey by sending a signal. Firstly, we assume that the intensity of the signal is very low which represents the exploration phase of the narwhals in the search space. The signal emitted by the narwhal I depends on this own position and its perception of the environment. The function of emission is described as follows:

$$S_E(X_i) = \frac{0.1}{1+\alpha.\|X_i - X_{prey}\|} \quad (2)$$

where X_i is the position of i^{th} narwhal and X_{prey} is the position of the prey which can also detects the signal and maybe changes its position (Narwhal prey, such as fish, can detect sounds narwhals).

$\|X_i - X_{prey}\|$ is the Euclidean distance between the position of the ith narwhal and the potential prey.

α is a control factor that controls the signal intensity.

The function $S_E(Xi)$ represents the signal emission at a particular location X_i. X_i represents the locations in the water or on the surface where the signal is measure. In addition, X_{prey} is the location of potential prey that the narwhal is targeting. α could modulate how the emission intensity changes with distance from the prey, possibly representing factors like the narwhal's vocalization behavior or the transmission properties of sound in water. The constant 0.1 represents a scaling factor determining the baseline emission rate which can be the characteristic value for narwhal vocalizations. The value 0.1 is chosen to normalize the emission function to a certain range or magnitude that is suitable for the model. It ensures that the emission values are not too large or too small, making them easier to interpret or work with.

This function produces a stronger signal when the narwhal is closer to the prey, and the signal decreases as the distance increases.

2.2.3. Signal Propagation

The emitted signal will be propagated through the water which can be modeled as a function based on the distance between the narwhals. The propagation function is defined as follows:

$$S_p(X_i) = S_E(X_i) \times P_R(X_i, X_{prey}) \quad (3)$$

Where $P_R(X_i, X_{prey})$ is the propagation function which is used to propagate the signal. This function is defined as follows:

$$P_R(X_i, X_{prey}) = \exp\left(-\frac{\|X_i - X_{prey}\|^2}{2\times(\sigma^t)^2}\right) \quad (4)$$

where σ^t is the standard deviation of the Gaussian at iteration t, which controls the decay of influence with distance.

Note that if σ^t has a small value, the communication will be more local, while a large value of σ^t the communication will be more global and adapted to large distance.

The value of σ^t decreases linearly over the iterations. It starts with σ_0.

$$\sigma^t = \sigma_0 - \left(\frac{t}{T}\right) \times \sigma_0 \quad (5)$$

2.2.4. Position Update

The location of narwhals is updated continuously in each iteration. It is updated following the emitted signal and its propagation. We can model this iteratively over time by updating positions at each step using the following function:

$$X_i^{t+1} = X_i^t + \Delta^t \quad (6)$$

Δ^t is the step at the t iteration and it is given by the following equation:

$$\Delta^t = \beta \times \left|S_p(i) \times X_{prey} - X_i\right| \quad (7)$$

$$\beta = r_1 - \frac{1}{\sigma^t + 1} \quad (8)$$

β is a parameter which is related to the σt that controls the decay of the propagation.

As mentioned previously, the prey can detect the signal emitted by the narwhals. In other terms, the prey can be affected by the emitted signal, that's whay we proposed $S_p(i) \times X_{prey}$. This process is described in Figure 2.

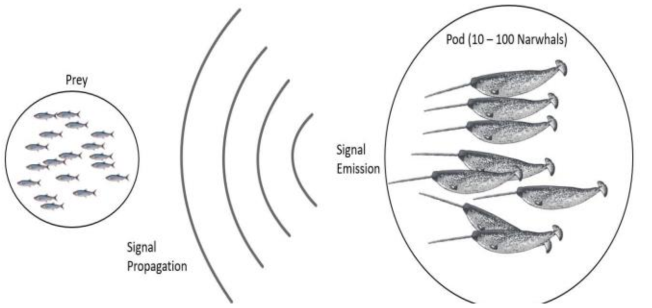


Figure 2. Signal emission and signal propagation to locate the prey.

2.3. Algorithm of Narwhal Optimizer

In NO, the σ^t parameter is used to determine if the algorithm performs the exploration phase or exploitation phase. Initially, the algorithm starts with an initial value of σ which decreases over the iterations.

The parameter σt in signal propagation can be associated with the range of influence of the signal emitted by narwhals. A larger value of σt will increase the range of influence, potentially allowing more extensive exploration of the environment. On the other hand, a smaller value of σt limits the range of influence of the signal, favoring more local exploitation of the information.

Another very important parameter in the algorithm is the parameter β. A small value of β leads to slower position adjustments which promote the exploration of the search space. Conversely, a high value of β could result in faster and more pronounced position adjustments in response to the received signal. This could encourage more exploitation of search space.

The pseudo code of NO algorithm is described in Algorithm (1).

Algorithm 1: Pseudo code of the Narwhal Optimizer

Initialize randomly the position of narwhals X_i
Calculate the value of the objective function $f(X)$
Initialize the parameters α, β and σ_0
while t < Max number of iterations do
  Update the value of σ using equation 5
  for each search agent narwhal i do
    Calculate the signal emission $S_E(X_i)$ using equation 2
    Calculate the signal propagation $S_P(X_i)$ using equation 3
    Update the value of β using equation 8
    Update the value of Δ^t using equation 7
    Update the position of narwhals X_i^{t+1} using equation 6
  end for
  Calculate the new value of the objective function
  t = t + 1
end while

In the Algorithm (1), the process of hunting prey by narwhals is be modeled as an optimization algorithm.

The algorithm starts with an initial random solution. At each iteration, it calculates the signal emission and signal propagation. The value of signal propagation is used to calculate the value of the step Δ. We assume that the prey can detect the signal and try to avoid it by changing its position. In each iteration, the algorithm updates the position of narwhals related to the position of the prey.

The calculation of the objective function depends on the context of the problem begin solved. It represents the quantity that the algorithm is attempting to maximize or minimize. We assume that *f*(*X*) is the objective function and *X* is the vector which the dimension represents the problem dimension. The objective functions used for testing the No algorithm are described in Table 1.

The complexity is very powerful to evaluate the performance and the computational time of the algorithm. The number of search agents (narwhals) is *N* and the dimension of the optimization problem is *D*. the computational complexity of the proposed algorithm is O(N × D).

3. Experimental Results

In this section, we present the experimental results of the NO algorithm. The proposed optimization algorithm NO is tested under 13 benchmark functions largely used in the literature to demonstrate the performance of a new metaheuristic [6, 16, 21].

Table 1 describes the 13 benchmark functions.

Table 1. Benchmark functions.

Functions	D	$[lb, ub]$	min
$f_1(x)=\sum_{i=1}^{n} x_i^2$	30	$[-100,100]$	0
$f_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	30	$[-10,10]$	0
$f_3(x)=\sum_{i=1}^{n}\left(\sum_{j-1}^{i} x_j\right)$	30	$[-100,100]$	0
$f_4(x)=\mathrm{MAX}_i\{\lvert x_i, 1\le i\le n\rvert\}$	30	$[-100,100]$	0
$f_5(x)=\sum_{i=1}^{n-1}\left[100\left(x_{i+1}-x_i^2\right)^2+(x_i-1)^2\right]$	30	$[-30,30]$	0
$f_6(x)=\sum_{i=1}^{n}([x_i+0.5])^2$	30	$[-100,100]$	0
$f_7(x)=\sum_{i=1}^{n} i x_i^4+rand[0,1)$	30	$[-1.28,1.28]$	0
$f_8(x)=\sum_{i=1}^{8}-x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$	30	$[-500,500]$	-418.9829×5
$f_9(x)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+20+e$	30	$[-32,32]$	0
$f_{10}(x)=\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}\left(x_i-a_{ij}\right)^6}\right)^{-1}$	2	$[-65,65]$	1
$f_{11}(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_1\left(b_i^2+b_i x_2\right)}{b_i^2+b_i x_3+x_4}\right]^2$	4	$[-5,5]$	0.00030
$f_{12}(x)=-\sum_{i=1}^{4} c_i\exp\left(-\sum_{j=1}^{3} a_{ij}\left(x_j-p_{ij}\right)^2\right)$	3	[1,3]	-3.86
$f_{13}(x)=-\sum_{i=1}^{10}\left[(X-a_i)(X-a_i)^T+c_i\right]^{-1}$	4	[0,10]	-10.5363

We propose to run the NO algorithm 100 times and record the optimal value of the objective function in each time. The average and the standard deviation are calculated. The maximum number of iterations is set to 500 with 30 search agents. α and σ0 are set to 2. The results obtained by NO algorithm are presented in Tables 2 and 3.

Table 2. Result of NO under benchmark functions (Part 1).

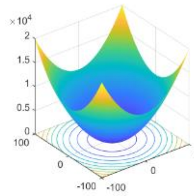
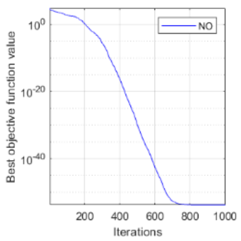
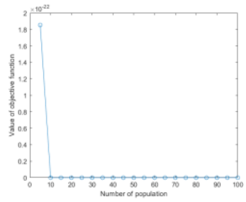
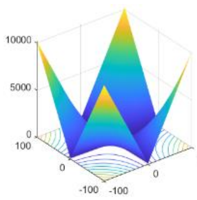
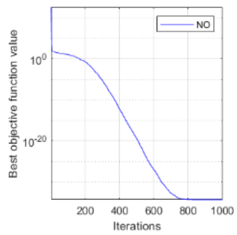
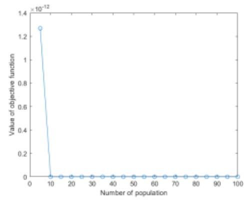
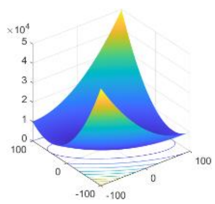
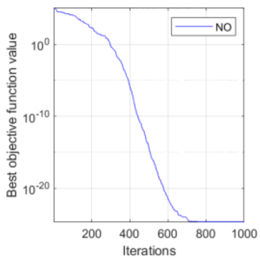
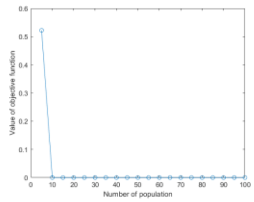
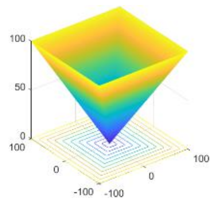
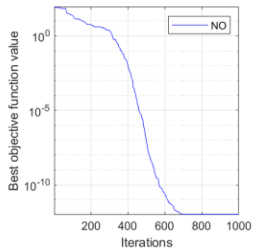
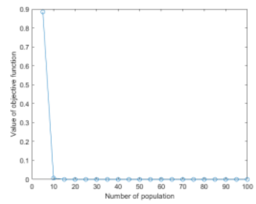
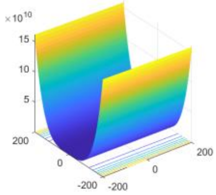
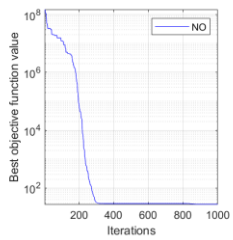
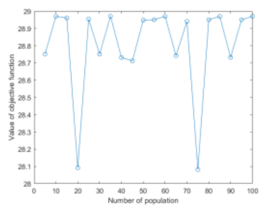
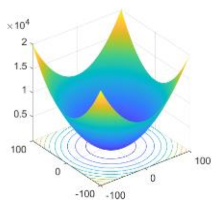
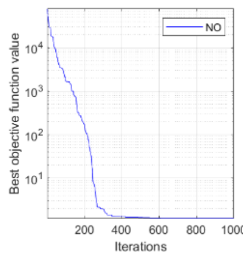
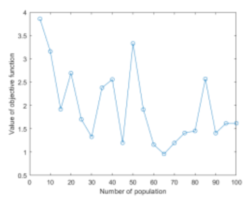
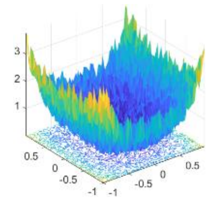
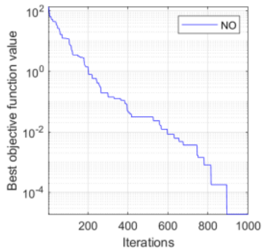
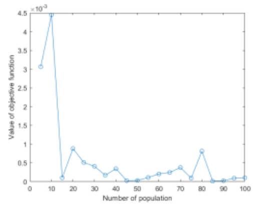
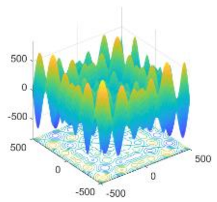
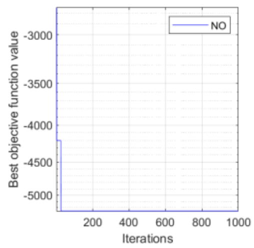
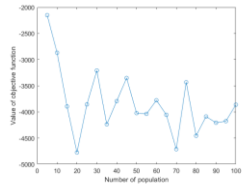
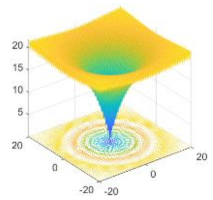
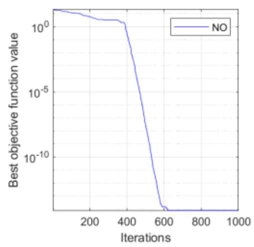
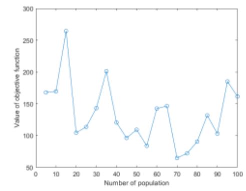
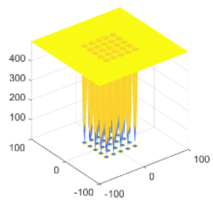
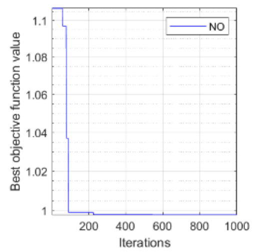
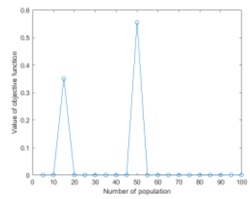
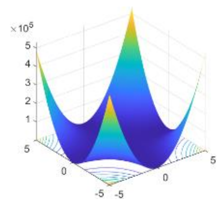
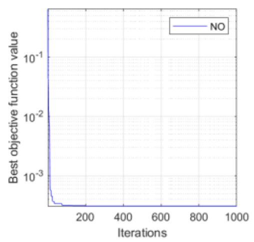
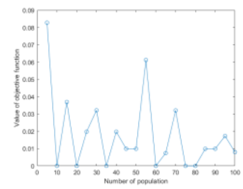
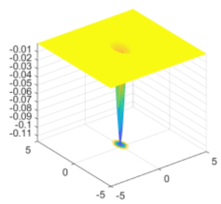
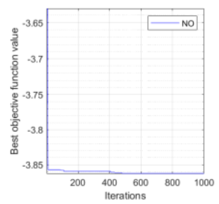
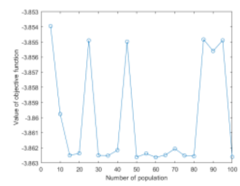
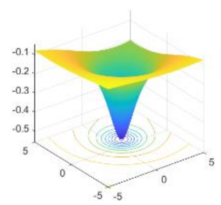
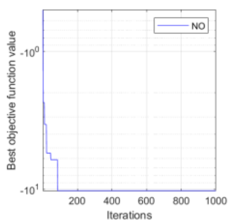
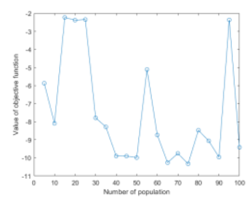
f	**NO**			**Convergence**		**Sensitivity of No**
	ave	**std**	**best**	**2D**	**Convergence curve**	**Population size**
$f_1(x)$	3.3673e-41	2.4278e-40	1.422e-53			
$f_2(x)$	1.0816e-29	5.4739e-29	2.3026e-34			
$f_3(x)$	7.3071e-11	6.0787e-10	2.2079e-25			
$f_4(x)$	7.7578e-06	3.6722e-05	1.0494e-12			
$f_5(x)$	28.8509	0.22997	27.5005			
$f_6(x)$	2.2185	0.47302	1.2022			

Table 3. Result of NO under benchmark functions (Part 2).

f	**NO**			**Convergence**		**Sensitivity of No**
	ave	**std**	**best**	**2D**	**Convergence curve**	**Population size**
$f_7(x)$	0.00038157	0.00025773	1.8817e-05			
$f_8(x)$	-3730.6946	361.465	-5259.8157			
$f_9(x)$	0.00038984	0.0021302	7.9936e-15			
$f_{10}(x)$	8.7363	6.119	0.998			
$f_{11}(x)$	0.01023	0.017371	0.00030814			
$f_{12}(x)$	-3.8607	0.0027875	-3.8628			
$f_{13}(x)$	-7.1484	2.9547	-10.2614			

Tables 2 and 3 present the results obtained by the NO algorithm. The first column represents the name of the function. The second column is numerical results which are the average (ave), standard deviation (std) and the best value of the objective function (best). The third column illustrates the visual results, which are the convergence curve and the last column is the sensitivity of NO related to the variation of the population size.

The analysis of the results described in Tables 2 and 3 demonstrates the highly competitive outcomes. The algorithm produces a satisfactory result. The unimodal functions allow us to test the exploitation phase of the algorithm. With the multimodal functions, we can test the proposed algorithm in terms of the exploration phase. Multimodal functions are characterized by the largen number of local optima. According to the results presented in Tables 2 and 3, we observe that the proposed algorithm is able to provide a very competitive result. The experimental results obtained through the application of the narwhal optimizer reveal its noteworthy performance in solving optimization problems.

In the conducted experiments, we observed variations in the performance of the optimization algorithms across different test cases. However, the performance of f_5 was less satisfactory in this particular test case; the algorithm did not perform very well. The algorithm achieved commendable results in optimizing f_8, it is crucial to note a less desirable outcome in terms of standard deviation. Similarly, for f_{10}, where the optimal value is 1, the algorithms provided positive results, achieving 0.99. On a positive note, the optimization process for f_{11} proved highly successful, with the algorithm effectively reaching the optimum value. The algorithm demonstrated impressive efficacy in achieving optimal results for f_{12}, successfully reaching the optimum solution.

The third column of the Tables 2 and 3 is the values of objective function obtained by changing the population size. The experiment aims to assess the performance sensitivity of the NO algorithm by systematically varying the population size parameter from 5 to 100 in steps of 5. By analyzing the results, optimal population sizes can be identified to enhance the algorithm's effectiveness for specific applications or problem domains. The parameters of the algorithm have been chosen experimentally. We tried many values and we figure out that 30 search agents and α and σ0 is set to 2, provide a good result. The parameters are adjust following the optimization problem.

We propose to compare the NO Algorithm with three optimization algorithms widely used in the literate which are, Particle Swarm Optimization (PSO), Gravitational Search Algorithm (GSA) and Differential Evolution (DE) [19]. Table 5 describes the obtained results. The comparison of the NO algorithm with PSO, GSA, and DE was conducted based on the average objective function value and standard deviation.

As can be seen from Table 4, the NO method consistently performed better than PSO, GSA, and DE in terms of average objective function values across all benchmark functions. This shows that, compared to the other algorithms, the results produced by the NO algorithm were, on average, closer to the ideal or desired solution. This indicates that the NO method exhibits superior convergence and more efficient exploration of the search space. The NO algorithm's durability and generalizability across diverse optimization tasks are suggested by the continuous performance advantage obtained across a range of benchmark functions.

In terms of standard deviation, the NO algorithm also showcased favorable values. The standard deviation measures the variability or spread of the objective function values obtained across multiple runs of the algorithm.

Table 4. Comparison of results obtained by the proposed approach and PSO, GSA and DE [18, 19, 20].

	NO		PSO		GSA		DE	
	ave	**std**	**ave**	**std**	**ave**	**std**	**ave**	**std**
$f_1(x)$	3.3673e-41	2.4278e-40	0.000136	0.000202	2.53E-16	9.67E-17	8.2E-14	5.9E-14
$f_2(x)$	1.0816e-29	5.4739e-29	0.042144	0.045421	0.055655	0.194074	1.5E-09	9.9E-10
$f_3(x)$	7.3071e-11	6.0787e-10	70.12562	22.11924	896.5347	318.9559	6.8E-11	7.4E-11
$f_4(x)$	7.7578e-06	3.6722e-05	1.086481	0.317039	7.35487	1.741452	0	0
$f_5(x)$	28.8509	0.22997	96.71832	60.11559	67.54309	62.22534	0	0
$f_6(x)$	2.2185	0.47302	0.000102	8.28E-05	2.5E-16	1.74E-16	0	0
$f_7(x)$	0.00038157	0.00025773	0.122854	0.044957	0.089441	0.04339	0.00463	0.0012
$f_8(x)$	-3730.6946	361.465	-4841.29	1152.814	-2821.07	493.0375	-11080.1	574.7
$f_9(x)$	0.00038984	0.0021302	46.70423	11.62938	25.96841	7.470068	69.2	38.8
$f_{10}(x)$	8.7363	6.119	3.627168	2.560828	5.859838	3.831299	0.998004	3.3E-16
$f_{11}(x)$	0.01023	0.017371	0.000577	0.000222	0.003673	0.001647	4.5E-14	0.00033
$f_{12}(x)$	-3.8607	0.0027875	-3.86278	2.58E-15	-3.86278	2.29E-15	N/A	N/A
$f_{13}(x)$	-71487	2.9547	-9.95291	1.782786	-10.5364	2.6E-15	-10.5364	1.9E-07

In the following, we applied the NO algorithm under real-world optimization problem known as “Three-bar truss design” [11, 22]. Structural design problem is one the popular real-world optimization problem which can be resolved using stochastic method. In this section, we propose to test the proposed approach NO on the problem of "Three-bar truss design problem" which is widely used in the literature. The problem is formulated as follows:

$$Consider\ \vec{x} = [x_1 x_2] = [A_1 A_2]$$

$$Minimise \qquad f(\vec{x}) = (2\sqrt{2x_1} + x_2) * l$$

$$Subject\ to \quad g_1(\vec{x}) = \frac{\sqrt{2x_1} + x_2}{\sqrt{2x_1^2 + 2x_1 x_2}} P - \sigma \leq 0$$

$$g_2(\vec{x}) = \frac{x_2}{\sqrt{2x_1^2 + 2x_1 x_2}} P - \sigma \leq 0$$

$$g_3(\vec{x}) = \frac{1}{\sqrt{2x_2 + x_1}} P - \sigma \leq 0$$

Variable range $0 \leq x_1, x_2 \leq 1$

where $l = 100\ cm, P = 2\ KN/cm^2, \sigma = 2\ KN/cm^2$

To solve the problem of Three-bar truss design, we used 30 populations and 800 iterations. The results are described in the following Table 5 and compared to SALP Swarm Algorithm (SSA) [20], hybrid Particle Swarm Optimization with Differential Evolution (PSO-DE) [14] and Cuckoo Search (CS) [11].

Table 5. Results obtained for three-bar truss design problem.

Methods	Optimal values for variables		Optimal weight
	x_1	x_2	
NO	0.788	0.408	263.68
SSA [22]	0.78866541425806	0.40827578444454	263.8958434
PSO-DE [14]	0.7886751	0.4082482	263.8958433
CS [11]	0.78867	0.40902	263.9716

Table 5 presents the experimental results of applying the NO algorithm to the real-world optimization problem of the "three-bar truss." The performance of NO is compared against three well-established metaheuristic algorithms: SSA, PSO-DE, and CS. The aim is to evaluate the effectiveness of NO in solving this specific engineering optimization problem and to assess its performance against existing state-of-the-art algorithms.

We clearly observe that NO consistently achieves satisfactory results comparable to algorithms, demonstrating its effectiveness in finding high-quality solutions for the three-bar truss optimization problem.

4. Conclusions

In this paper, we present a new metaheuristic algorithm called the NO to solve global optimization problems. The proposed algorithm NO mimicked the process of hunting prey by narwhals. The main key and inspiration of NO stem from the behavior of sending click sound to locate prey, which can be represented as signal emission and propagation. Following this mechanism, the position of narwhals is updated over iterations until they reach the prey. The performance evaluation of the proposed algorithm was conducted on a set of 13 benchmark functions with varying dimensions and ranges. The analysis of the results showed the higher performance of the proposed approach in terms of optimality and scalability. As a perspective, we are going to develop a new version of the NO algorithm specifically designed for binary and multi-objective problems. Future research will focus on enhancing the NO algorithm. The convergence speed and solution quality can be improved. In addition, NO algorithm can be combined with other optimization techniques by developing hybrid approaches. Hybrid algorithms that combine the strengths of various algorithms, such as genetic algorithms, simulated annealing, or machine learning-based approaches, may lead to enhanced performance and versatility in tackling complex optimization tasks. Also, introducing adaptive strategies into the NO algorithm can dynamically adjust its parameters. Research in this direction could lead to more robust and flexible optimization algorithms capable of efficiently solving a wide range of problems.

References

[1] Abdel-Basset M., Mohamed R., Jameel M., and Abouhawwash M., "Nutcracker Optimizer: A Novel Nature-Inspired Metaheuristic Algorithm for Global Optimization and Engineering Design Problems," *Knowledge-Based Systems*, vol. 262, pp. 110248, 2023. https://doi.org/10.1016/j.knosys.2022.110248

[2] Ali A., Yaseen M., Aljanabi M., and Abed S., "Transfer Learning: A New Promising Techniques," *Mesopotamian Journal of Big Data*, vol. 2023, 2023. DOI:10.58496/MJBD/2023/004

[3] Braik M., Hammouri A., Atwan J., Al-Betar M., and Awadallah M., "White Shark Optimizer: A Novel Bio-Inspired Meta-Heuristic Algorithm for Global Optimization Problems," *Knowledge-Based Systems*, vol. 243, pp. 108457, 2022. https://doi.org/10.1016/j.knosys.2022.108457

[4] Bosire A. and Maingi D., "Using Deep Analysis of Driver Behavior for Vehicle Theft Detection and Recovery," *in Proceedings of International Arab Conference on Information Technology*, Oman, pp. 1-6, 2021. DOI:10.1109/acit53391.2021.9677433

[5] Bonabeau E., Dorigo M., and Theraulaz G., *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, 1999. https://doi.org/10.1093/oso/9780195131581.001.0001

[6] Digalakis J. and Margaritis K., "On Benchmarking Functions for Genetic Algorithms," *International Journal of Computer Mathematics*, vol. 77, no. 4, pp. 481-506, 2021. https://doi.org/10.1080/00207160108805080

[7] Dorigo M., Birattari M., and Stutzle T., "Ant Colony Optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28-39, 2006. DOI: 10.1109/MCI.2006.329691

[8] El-kenawy E., Khodadadi N., Mirjalili S., Abdelhamid A., Eid M., and Ibrahim A., "Grey

Lag Goose Optimization: Nature-Inspired Optimization Algorithm," *Expert Systems with Applications*, vol. 238, pp. 122147, 2024. https://doi.org/10.1016/j.eswa.2023.122147

[9] Eslami N., Yazdani S., Mirzaei M., and Hadavandi E., "Aphid Ant Mutualism: A Novel Nature-Inspired Metaheuristic Algorithm for Solving Optimization Problems," *Mathematics and Computers in Simulation*, vol. 201, pp. 362-395, 2022. https://doi.org/10.1016/j.matcom.2022.05.015

[10] Han M., Du Z., Yuen F., Zhu H., Li Y., and Yuan Q., "Walrus Optimizer: A Novel Nature-Inspired Metaheuristic Algorithm," *Expert Systems with Applications*, vol. 239, pp. 2023. https://doi.org/10.1016/j.eswa.2023.122413

[11] Gandomi A., Yang X., and Alavi H., "Cuckoo Search Algorithm: A Metaheuristic Approach to Solve Structural Optimization Problems," *Engineering with Computers*, vol. 29, pp. 17-35, 2013.

[12] Kennedy J. and Eberhart R., "Particle Swarm Optimization," *in Proceedings of ICNN'95-International Conference on Neural Networks*, Perth, pp. 1942-1948, 1995. DOI: 10.1109/ICNN.1995.488968

[13] Kirkpatrick S., Gelatt C., and Vecchi M., "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671-680, 1983. DOI: 10.1126/science.220.4598.67

[14] Liu H., Cai Z., and Wang Y., "Hybridizing Particle Swarm Optimization with Differential Evolution for Constrained Numerical and Engineering Optimization," *Applied Soft Computing*, vol. 10, no. 2, pp. 629-640, 2012. https://doi.org/10.1016/j.asoc.2009.08.031

[15] Maxwell J., *A Treatise on Electricity and Magnetism*, Oxford: Clarendon, 1892.

[16] Mahdavi M., Fesanghary M., and Damangir E., "An Improved Harmony Search Algorithm for Solving Optimization Problems," *Applied Mathematics and Computation*, vol. 188, no. 2, pp.1567-1579, 2007. https://doi.org/10.1016/j.amc.2006.11.033

[17] Maree M., Eleyat M., and Mesqali E., "Optimizing Machine Learning-based Sentiment Analysis Accuracy in Bilingual Sentences via Preprocessing Techniques," *The International Arab Journal of Information Technology*, vol. 21, no. 2, pp. 257-270, 2024. https://doi.org/10.34028/iajit/21/2/8

[18] Mirjalili S. and Lewis A., "S-Shaped Versus V-Shaped Transfer Functions for Binary Particle Swarm Optimization," *Swarm and Evolutionary Computation*, vol. 9, pp. 1-14, 2013. https://doi.org/10.1016/j.swevo.2012.09.002

[19] Mirjalili S., Mirjalili S., and Lewis A., "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46-61, 2014. https://doi.org/10.1016/j.advengsoft.2013.12.007

[20] Mirjalili S., Gandomi A., Mirjalili S., Saremi S., Faris H., and Mirjalili S., "Salp Swarm Algorithm: A Bio-Inspired Optimizer for Engineering Design Problems," *Advances in Engineering Software*, vol. 114, pp. 163-191, 2017. https://doi.org/10.1016/j.advengsoft.2017.07.002

[21] Rashedi E., Nezamabadi H., and Saryazdi S., "GSA: A Gravitational Search Algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232-2248, 2009.

[22] Sadollah A., Bahreininejad A., Eskandar H., and Hamdi M., "Mine Blast Algorithm: A New Population-Based Algorithm for Solving Constrained Engineering Optimization Problems," *Applied Soft Computing*, vol. 13, no. 5, pp. 2592-2612, 2013. https://doi.org/10.1016/j.ins.2009.03.004

[23] Storn R. and Price K., "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. DOI:10.1023/A:1008202821328

[24] Zhong C., Li C., and Meng Z., "Beluga Whale Optimization: A Novel Nature-Inspired Metaheuristic Algorithm," *Knowledge Based Systems*, vol. 251, pp. 109215, 2022. https://doi.org/10.1016/j.knosys.2022.109215

Seyyid Medjahed received a Doctor of Informatics degree from University of Science and Technology Mohamed-Boudiaf Oran, Algeria in 2017. He is currently an associate professor at Computer Science Department in Reliane University, Algeria since 2012. His research includes meta-heuristics, global optimization, machine learning, data mining, bioinformatics.

Fatima Boukhatem holds a Doctor degree from University of Djilali Liabes, Sidi Belabes, Algeria. Her research areas of interest include physical, Bioinformatics, Artificial Intelligent.