Improved Barnacles Movement Optimizer (IBMO) Algorithm for Engineering Design Problems

Syed Kumayl Raza Moosavi[1][0000−0001−7064−255X], Muhammad Hamza Zafar[2][0000−0002−5025−2009], Seyedali Mirjalili[3][0000−0002−1443−9458], and Filippo Sanfilippo[2,4][0000−0002−1437−8368]

[1] National University of Sciences and Technology, Islamabad, 44000, Pakistan
[2] Dept. of Engineering Sciences, University of Agder, 4879, Grimstad, Norway
[3] Centre for Artificial Intelligence Research and Optimization, Torrens University Australia, Fortitude Valley, Brisbane, 4006, QLD, Australia
[4] Department of Software Engineering, Kaunas University of Technology, LT-44029 Kaunas, Lithuania
filippo.sanfilippo@uia.no

Abstract. A better understanding of natural behavior modeling in mathematical systems has enabled a new class of stochastic optimization algorithms that can estimate optimal solutions using reasonable computational resources for problems where exact algorithms show poor performance. The position up-dating mechanism in various optimization algorithms utilizes similar chaotic random behavior which impedes the performance of the search for a globally optimum solution in monotonic nonlinear search space. In this work, an approach is proposed that tackles these issues on an already established algorithm; Improved Barnacle Mating Optimization (IBMO) Algorithm, inspired by the movement and mating of Gooseneck Barnacles. The algorithm introduces the mimicry of the movement and mating behavior in nature to model an optimization process. Several benchmark functions and engineering case studies are employed to gauge the performance of the proposed optimization technique. Results are compared with several meta-heuristics and conventional optimization algorithms. It is observed that the IBMO algorithm per-forms generally better and provides a huge potential for solving real-world problems.

Keywords: Metaheuristic Algorithms · Artificial Intelligence · Improved Barnacles Mating Optimization · Engineering Design Problems

1 Introduction

The process of evolution in nature has made many micro and major processes integrate mutually for a common goal. In recent decades, the complexity of real-world problems has resulted in the need for highly intelligent and reliable optimization techniques [7]. During the past century computing resources have exploded and brought forth a new class of optimization and control known as meta-heuristics into both science and industry. Such optimization techniques belong to the field of Computational Intelligence with three main branches of fuzzy logic, neural network and evolutionary computation. Swarm intelligence techniques belong to the family of evolutionary computation, which is unique in the sense as it mimics the simplest tasks performed by the swarm of a less intelligent species that require collective behavior for survival. The shared distributive tasks and ability to share information among the swarm particle/agents/members enable the solution for complex problems.

These solutions of a swarm-based movement/mating technique supersede conventional algorithms due to it being free of the gradient based technique which inherently gets trapped in local optima [25, 31]. The main obstacle comes from the chaotic behavior of individual particles with nonlinear activity models. Despite the connotation of 'chaos' which suggests unpredictable and irregular systems, chaos theory suggests that seemingly random events can lead to a pattern over time [19]. This also holds authentic for modern swarm intelligence-based systems which randomly initiate problem formulation and improvise the random solutions over some time. The main role in swarm intelligence is played by random exploratory and exploitative phases. In the exploration phase, the expectation is to get a high enough variance between agents to cover a massive area of the search space while the exploitation phase focuses on a small subset of the search space that already contains the best solution at any given time. The exploration task facilitates the avoidance of local minima entrapment while the exploitation task explores nearby promising solutions [2]. Most techniques utilize pre-defined criteria for the balancing of exploration and exploitation behavior such as in [13].

The most popular swarm-based algorithm, encouraged by the motion of a swarm of birds, started this field of study is the Particle Swarm Optimization (PSO) algorithm [12]. Other similar swarm-based algorithms proposed in the literature include Grey Wolf Optimization (GWO) Algorithm [15], where the alpha wolf carries the best solution and the consequent wolf classes, Ant Colony Optimization (ACO) algorithm [8], Grasshopper Optimization Algorithm (GOA) [17], Fruit fly Optimization Algorithm (FOA) [27] and many more.

While the above-mentioned algorithms conduct solution finding mechanism using the movement and/or food foraging techniques of the swarm, other branches of meta-heuristic solutions include physics-based techniques and evolutionary algorithms. Some popular algorithms that mimic the laws of physics include Big-Bang Big Crunch (BBBC) [21], Gravitational Search Algorithm (GSA)[20], Black Hole [10], and Small World Optimization Algorithm (SWOA) [9]. Using methods such as gravitational force, electromagnetic force, inertial force and weights these algorithms adopt physical rules to incorporate movement and communication of multi agents not unlike evolutionary algorithms that employ models of evolution in nature. The set of candidate solutions is improved iteratively by offspring production that inherit genes/properties of parents. For this category, popular algorithms include Genetic algorithm (GA) [16], evolutionary Strategies (ES) [4], Biogeography-based Optimization [22], and Differential evolution (DE) [18]. A new trend in solving the optimization tasks is developed where the hybrid models are employed taking advantage of fast convergence of gradient based decision processes and the effective exploration of computation-ally expensive optimization processes. For instance, the eagle strategy in con-junction with Flower Pollination Algorithm (FPA) was used by [28] to balance the exploration and exploitation. In [1], Abdel-Raouf et al. chose Sudoku puzzles as an optimization problem and then innovated a hybrid technique for optimization using FPA and Chaotic Harmony Search (FPCHS). Using improved PSO, improved whale optimization algorithm and dual population evolutionary mechanism, [26] - conceptualized a multi-objective hybrid algorithm for Automatic train operation.

From all the optimization algorithms described in the literature, two things can be surmised; (1) one single algorithm cannot be termed as the best suited for all optimization related problems and (2) determination of the exploration and exploitation phase requirements vary with each proposed optimization problem. The former has been logically proven by the No-Free Lunch (NFL) Theorem [11]. If one algorithm works for one type of dataset problem, it cannot be guaranteed that it will be best suited for another. This continuously encourages the research field to bring forth new techniques. In case of the latter, each algorithm has a specific set of instructions for determining the exploration and exploitation phases. Both vary using tuning parameters within the algorithms. Hence, finding a proper balance of the two, especially keeping in mind the stochastic nature of meta-heuristic algorithms, is a difficult task.

Barnacles movement Optimizer (BMO) [24], a newly proposed bio-inspired algorithm, has the features of fewer parameters and mathematical manipulation to search for promising search space solutions. It can be conferred that with a smaller mathematical model, the computation time is low but the tradeoff be-tween accuracy is too high and warrants improvement in the areas of performance and parameter tuning. In addition, the random behavior impedes the performance for the search of a globally optimum solution in monotonic nonlinear search space. This paper proposes an improved version of the bio-inspired meta-heuristic algorithm in which the mimicry of the movement and mating behavior of gooseneck barnacles is modelled. The effectiveness of the Improved Barnacles Mating Algorithm (IBMO) is evaluated using 23 benchmark test functions. These functions contain unimodal, multi-modal and fixed dimension multi-model functions. Secondly it has been put into comparison with other meta-heuristic algorithms that are; PSO, GWO, BMO, Arithmetic Optimization Algorithm (AOA) [3], Flower Pollination Algorithm (FPA) [29] and Dragonfly Algorithm (DFA) [14].

2 Improved Barnacles Mating Algorithm

This section first presents the inspiration for the algorithm studied from Gooseneck Barnacles' movement and mating technique. Successively, the initialization and selection process for mating of the barnacles is discussed. Next the barnacles' reproduction probabilistic model and the resulting exploration / exploitation phases are studied. Finally, the improvement from the BMO technique is introduced i.e., the movement of the barnacles, prior to the mating process, is articulated.

2.1 Inspiration

Found in the rocky shores of the Eastern Atlantic, gooseneck barnacles (GB), a species of filter-feeding crustaceans, begin their life cycle as free moving larvae in the ocean [6]. Unlike their counterpart crustaceans such as crabs, lobsters and shrimp that have free ranging lives, these barnacles grow to eventually be attached to a hard surface. Amongst the crashing waves and direct sunlight, these barnacles whiten in the intertidal zone. Only on rare occasions can they be found stashed safely away from the rays of the sun and the high tides of the sea. Goose-neck barnacles, unlike the acorn barnacles that encrust themselves on the hard surface, bend their long necks back and forth but despite that, both types of these barnacles once attached, stay there until the end of their life cycle.

GBs settle on hard surfaces in a rather remarkable way. Once their growth is completed from free moving nauplius larvae to cyprid larvae to adults, they attach on a rocky surface using their heads with a cement like strong substance. The material has been used for many dental surgery adhesives. Once fully submerged the GB metamorphoses its feet into feathery feeding appendages known as 'cirri'. These cirri are used to clean the dead matter of the ocean by feeding on it. Because it's the feet that are faced towards the ocean, if any predators end up eating the cirri of GB, it simply regrows it again. Contrary to that, if the GB attached its feet on the ground and used its head as a feeding mechanism it would surely die from predation [5]. The complete life cycle of GB is shown in Fig. 1.

Once a GB finds its suitable location on the marine shores, other GBs also fol-low the initial GB's location and attach themselves close by for feeding and mating purposes primarily because they do not physically move from their position of attachment. Barnacles are hermaphroditic, meaning they have both male and female reproductive organs. The life cycle of these barnacles is illustrated in Fig. 1. Prior to the mating ritual, the movement and search of these barnacles during the Nauplius to Cyprid Larvae to adulthood stage is used as the inspiration for the improvement of the algorithm explained in the coming sections.

2.2 Initialization

In IBMO, the candidate solutions are termed here as GB. The initial population is initialized as:

$$GB = \begin{pmatrix} g_1^1 \cdots g_1^N \\ \vdots \ddots \vdots \\ g_t^1 \cdots g_t^N \end{pmatrix} \tag{1}$$

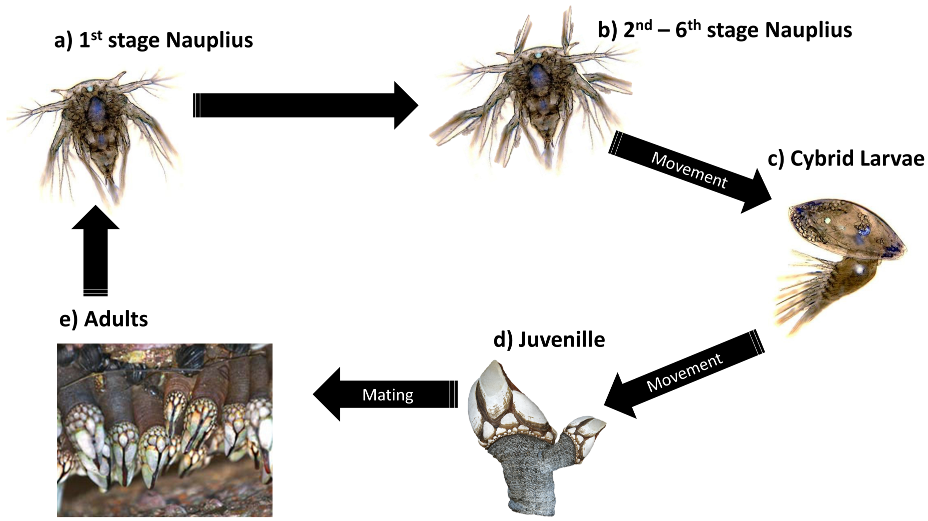

Fig. 1. Life Cycle of Gooseneck barnacles: a) Eggs develop and hatch inside the adults and are released into the water. b) Different stages of the nauplius larvae are developed drifting overall several weeks developing complex structure. c) Cyprid larvae begin searching for a suitable place to attach. d) Attached at the trunk (peduncle) of an adult initially and later moves to the surface for permanent attachment. e) Adults attached on rocks or hard surfaces mating either by the penis elongation or by the sperm cast process

where g is the generic candidate and t is the complete population of the GB's size and N here represents the size of the control variables within each GB in the population. The boundary conditions on the movement of the GB population are applied using equations (2);

$$\begin{aligned} ub &= \left[ub, \ldots, ub_i\right] \\ lb &= \left[lb, \ldots, lb_i\right] \end{aligned} \tag{2}$$

where lb and ub is the bound for lower and upper respectively.

2.3 Mating Selection Method

The mating of every two barnacles in the selection process is dependent on the length of the penis size (ps) of the male barnacle. The mating ritual is dependent on the following principle:

- Random selection of the barnacles and restriction based upon the length of the penis i.e., penis size. Since barnacles are hermaphrodites, each barnacle can provide and receiving sperm from other barnacles. If the position of the two barnacles in the mating process is larger than the penis size, sperm casting is applied. Sperm cast happens when adjacent male penis is out of range of the GB, so the male-acting barnacle ejects sperm in the ocean towards the direction of the female-acting barnacle for reproduction.

In the above-mentioned principal assumptions, the exploitation phase and exploration phase are prescribed in the IBMO. The choosing strategy male-female barnacles is expressed as:

$$\begin{aligned} GB_{male} &= rand(n) \\ GB_{female} &= rand(n) \end{aligned} \tag{3}$$

where n is the total number of barnacles.

2.4 Reproduction

The Hardy-Weinberg principle [23] states that in a random population of alleles i.e., variant form of genes, with no external factors such as selection, crossover or mutation, the offspring generation can be surmised as a simple relationship between allele frequency and genotype frequency. This law of offspring production can be seen in equation (4):

$$\sum_{i} p_i^2 A_i A_j + \sum_{i<j} 2 p_i p_j A_i A_j \tag{4}$$

where p_i is the allelic frequency of allele A_i and similarly p_j is the allelic frequency of allele A_j. When in context with the GB mating process, A_i can be expressed as the Male barnacle and A_j can be expressed as the Female barnacle during the iteration process. The expected genotype process can be expressed as such that the frequency of Male plus Male self-mating process generate offspring with the probability of p^2. Similarly, the Female plus Female process has a probability of q^2 and Male plus Female generates with the probability of $2pq$. The sum of these probability entries is seen in equation (5):

$$p^2 + 2pq + q^2 = 1 \tag{5}$$

Thus, in supposition, we can deduce that the offspring selection is dependent on these two probabilities. Self-procreation of Male or Female barnacles produces offspring of very little variance from the predecessor and as such this selection does not use the exploitation phase of multi-agent optimization algorithms to its optimal capacity as the offspring does not move in the search space therefore self-procreation is not dealt with in this algorithm.

2.5 Exploration / Exploitation Phase

In order to produce the new variables, equations (6), (7) and (8) are used for the reproduction phase:

$$GB_i^{new} = A \cdot GB_{male}^N + B \cdot GB_{female}^N \tag{6}$$

where A is the random number between [0,1] and $B = (1 - A)$. GB^N_{male} and GB^N_{female} are the male and female variables that are selected from equation (3). A and B dictate the weightage of the Male and Female's positional characteristics which are going to be passed down onto the new offspring. ps is responsible for the determination of the exploration, exploitation phases. If the barnacle's selection is within the range of penile length of the chosen male, the former exploitation phase occurs.

During the first half of the iteration process, the exploration phase needs to be widened to find better optimum solutions in the search space. Therefore, a parameter k is multiplied into the values of A and B as shown by the equation (7). Using this parameter enhances the exploration phase from the traditional Barnacles Mating Optimizer Algorithm.

$$\begin{aligned} A &= k \cdot A \\ B &= k \cdot B \end{aligned} \tag{7}$$

In IBMO, the process of sperm casting is considered the exploration phase and it happens when the selection of barnacles that are to be mated in the current iteration becomes greater than the penis size ps. The process can be expressed as:

$$GB^{new}_i = rand() \cdot GB^N_{barnacle\_male} \tag{8}$$

2.6 Movement

GB larvae move across the ocean and attach on a rocky surface once fully grown. For the purposes of mating, the GB attaches itself onto a surface with either a GB already present or follows another GB that is going to attach itself onto the rocky surface. This movement of following and attaching of the GB is represented as following the entire population of GB's best solution i.e., the global minima GB for each iteration of the exploitation and exploration phase. This consideration of the global minima is added in equation (6) and subtracted in equation (8) as shown below:

$$\begin{aligned} GB^{new}_i &= A \cdot GB^N_{male} + B \cdot GB^N_{female} + r \cdot GB_{best} \\ GB^{new}_i &= rand() \cdot GB^N_{barnacle\_male} - r \cdot GB_{best} \end{aligned} \tag{9}$$

where r is a random number within the range [0,1] and GB_{best} is the position of the gooseneck barnacle with the global best solution from all previous iterations. The pseudo code for the IBMO technique is shown in Algo 1. The detailed process flow diagram of IBMO is illustrated in Fig. 2.

The main contribution of this work is the inclusion of the parameter k. The inclusion of the k parameter and the movement algorithm enhancement from the traditional barnacles mating optimizer helps expand the search range of the algorithm and helps traverse towards the global minima solution much quickly. This makes it easier also to move away from the local minima cost solutions and increases the diversity of the search space solution for the barnacle population [24].

3 Experimental Results

In this section, numerous simulations are conducted to illustrate the efficacy of the IBMO algorithm. First and foremost, 23 point of reference test functions [25] are utilized to observe the features of the IBMO. The benchmark test functions are categorized into; unimodal, multimodal, and composite functions. As the name suggests the unimodal functions have single global minima point. Therefore, the basic criteria of finding the minima solution with IBMO is tested. Next, the multimodal and composite functions are tested which have multiple local minima solutions so while the unimodal functions test the exploitation phase of the algorithm, these two categories of functions test the exploration phase and the avoidance of getting trapped in local minima solutions.

Algorithm 1: Pseudo Code of Proposed Algorithm

```
Initialize population GB_i, i = 1, 2, ..., N
Evaluate the fitness of each barnacle
Sort the solutions and find the best solution
while iter<iter_max do
    set the value of ps
    select male and female GB using eq. (3)
    if Distance of male and female ≤ ps then
        for Each variable do
            Generate offspring using eq. (9)-A
    Else if Distance of male and female > ps then
        for Each variable do
            Generate offspring using eq. (9)-B
    Apply boundary conditions on updated solution Calculate Fitness and sort
      population to update the best solution iter = iter + 1
return Best Solution;
```

For a fair quantitative comparison of the test functions, the base parameters of the multi-agent algorithm needed to be kept the same. While the upper and lower boundaries for each test function is different, the number of barnacles chosen for the test is 30 and a maximum of 50 iterations were employed for the barnacles to search the global minima solution. Each test for performed a total of 30 times to generate statistical results. Different performance indicators were used to test out the features of the algorithm which is the average of the 30 generated tests and the standard deviation of the best solutions. Qualitative results are illustrated in the next section which includes search history, fitness average, and search history of the multi-agents.

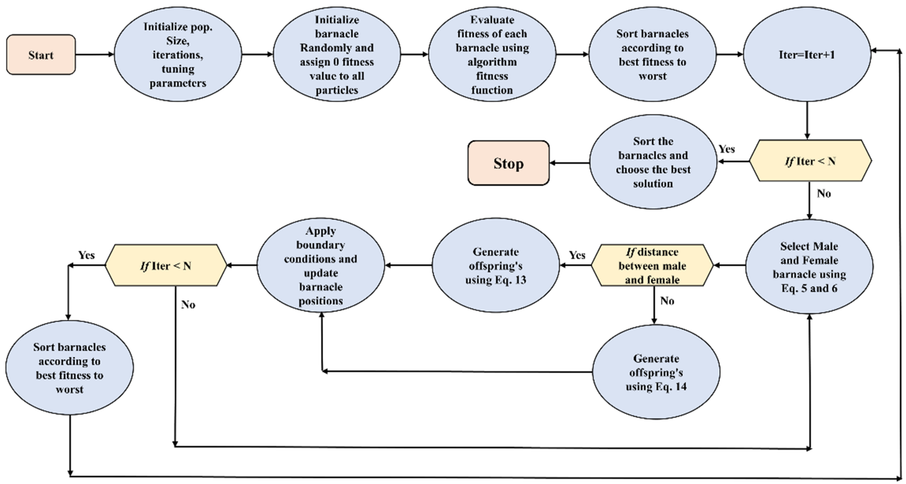

Fig. 2. Flow chart of Proposed Algorithm

For comparison, six recent and traditional algorithms were used under the same light of parameters namely, BMO, AOA, GWO, PSO, FPA, DFA. These algorithms were chosen to distinguish the different capabilities IBMO has over the characteristics of such nature of algorithms. The tuning parameters are chosen for each algorithm, as suggested by the respective original papers. GWO and FPA have been widely used in recent years, AOA and BMO are recently conceived novel algorithms and DFA has a very wide based search capability. A computational complexity analysis is also presented using the CEC-2019 test functions. Finally, to understand the effectiveness of the algorithm in real world applications, three mechanical design problems are tested and compared with other algorithms to demonstrate the superiority of IBMO.

3.1 Qualitative Analysis

The first set of simulation results is performed on all 23 test functions with IBMO with tuning parameters at $ps = 0.4$ and $k = 2.0$, 30 barnacles, and 50 iterations. The main objective of the simulation was to demonstrate the IBMO algorithms efficacy. The test functions contain unimodal, multimodal, and fixed dimension multi modal functions which would be the perfect litmus test for the given algorithm because they include single minima and multiple minima which will show the usage of the exploration and exploitation phase of the IBMO.

3.2 Quantitative Analysis

The effectiveness of the algorithm is tested against two further distinguishing parameters. One is the increase in the dimensions for each function and the other is the effect of increases in the number of barnacles associated with the IBMO algorithm. These parameters are highlighted primarily because the real-world datasets are becoming more complex and hence it is required the algorithm to be tested in complex scenarios. Secondly, the IBMO is further compared and tested on the test functions (F1 – F23) with other meta-heuristic algorithms, some recent and some traditional. The comparison draws out the superiority of the algorithm against others. All values recorded as 0 represent values below 1.0e-300.

3.3 Comparative Analysis with other Algorithms

To show the efficacy of the proposed technique, the comparison of IBMO for F1 to F23 test functions are performed with BMO, AOA, PSO, GWO, FPA and DFA. Each algorithm uses 50 particles run through 500 iterations. The tuning parameters of the IBMO are set the same as in the previous section. The convergence curve comparison for the algorithms can be seen in Fig. 3.

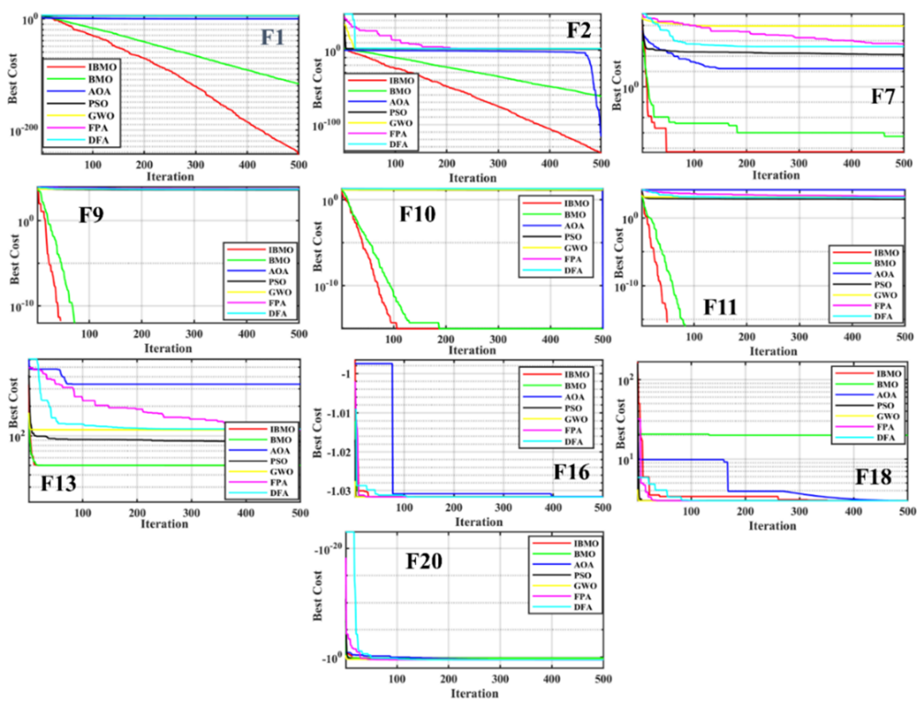

Fig. 3. Convergence Curves

Detailed analysis of each function against different dimensions is prepared. Referring to Table 1 and Table 2, IBMO has gained the far most performance by finding the global minima solution mostly better than other compared optimizers. The best results are written in bold for better visualization of each test function.

Table 1: Comparison of meta heuristic algorithms with IBMO for unimodal and multimodal functions.

Func	Dim	IBMO	BMO	AOA	PSO	GWO	FPA	DFA
F1	2	**0**	3.34E-184	**0**	1.59E-04	**0**	1.94E-39	1.17E-123
	10	**0**	3.12E-131	**0**	4.11E-01	1.94E-64	3.82E-10	4.21E-03
	30	**5.34E-272**	1.38E-123	2.39E-28	4.64E+01	4.47E-06	2.98E+00	1.22E+03
F2	2	**0**	3.41E-90	**0**	8.14E-04	**0**	9.42E-21	2.35E-62
	10	**1.25E-160**	1.36E-69	0	1.88E-01	6.64E-18	1.72E-05	2.87E-01
	30	**1.23E-143**	8.33E-66	0	3.25E+00	5.13E+00	2.47E+00	1.23E+01
F3	2	**0**	4.57E-176	**0**	3.53E-07	**0**	3.91E-33	8.96E-124
	10	**0**	1.09E-107	**0**	3.04E+00	9.07E-02	1.40E+00	1.57E+00
	30	**2.54E-212**	7.37E-91	3.68E-03	8.08E+02	6.06E+03	1.68E+04	1.17E+04
F4	2	**0**	3.06E-87	**0**	4.84E-03	1.07E-304	8.86E-17	5.89E-62
	10	1.88E-144	2.07E-58	**0**	4.03E-01	3.65E-22	9.62E-03	9.19E-01
	30	**2.55E-137**	1.55E-56	4.86E-33	6.58E+00	1.11E+01	1.36E+01	1.75E+01
F5	2	6.54E-07	1.00E+00	4.55E-02	1.16E-04	2.14E-03	6.54E-07	**6.54E-07**
	10	**2.26E+00**	8.82E+00	5.69E+00	1.14E+01	3.54E+00	6.56E+00	2.27E+00
	30	**2.70E+01**	2.90E+01	2.88E+01	2.15E+03	2.89E+01	2.46E+02	2.24E+04
F6	2	**1.03E-04**	6.45E-06	1.51E-05	2.04E-04	2.52E-04	2.42E-04	1.01E-05
	10	6.04E-01	2.50E+00	3.18E-02	8.62E-02	**6.02E-01**	7.01E-10	3.18E-02
	30	**1.56E+01**	2.39E+01	1.69E+01	1.63E+03	1.01E+04	4.19E+03	1.92E+04
F7	2	1.92E-04	7.84E-05	**3.75E-05**	3.22E-04	4.88E-03	1.17E-04	7.97E-05
	10	**4.60E-05**	3.63E-04	4.69E-05	3.61E-03	1.12E-02	7.32E-03	8.91E-03
	30	**1.79E-04**	3.86E-04	8.34E-03	7.45E-02	4.90E-01	7.08E-02	3.03E-01
F8	2	**-8.38E+02**	-6.21E+02	-7.19E+02	-7.20E+02	-8.38E+02	-8.36E+02	-8.35E+02
	10	**-3.24E+03**	-1.77E+03	-3.23E+03	-2.11E+03	-2.65E+03	-3.24E+03	-2.83E+03
	30	-4.41E+03	-3.97E+03	-4.79E+03	-5.20E+03	**-5.53E+03**	-5.37E+03	-3.42E+03
F9	2	**0**	**0**	**0**	3.31E-05	9.95E-01	**0**	**0**
	10	**0**	**0**	**0**	5.46E+00	2.31E+01	2.44E+01	1.39E+01
	30	**0**	**0**	4.16E+00	5.35E+01	1.42E+02	2.30E+02	1.77E+02
F10	2	**8.88E-16**	**8.88E-16**	**8.88E-16**	6.73E-03	**8.88E-16**	**8.88E-16**	**8.88E-16**
	10	**8.88E-16**	**8.88E-16**	1.31E-03	1.41E+00	8.41E+00	8.24E-03	6.60E+00
	30	**8.88E-16**	**8.88E-16**	1.79E+01	3.75E+00	1.10E+01	6.74E+00	2.00E+01
F11	2	**0**	**0**	**0**	1.36E-02	7.40E-03	7.12E-10	7.40E-03
	10	**0**	**0**	3.49E+00	4.60E-01	5.16E+00	2.09E-01	2.75E-01
	30	**0**	**0**	4.64E+02	1.44E+00	5.44E+01	7.97E-01	7.64E+00
F12	2	**2.36E-31**	5.82E-04	5.54E-03	2.53E-06	1.38E-04	1.38E-04	2.12E-25
	10	**1.92E-04**	1.47E-01	9.78E-02	5.59E-04	5.82E-01	4.77E-10	8.48E-02
	30	**7.37E-03**	1.29E+00	6.79E+00	2.25E-01	1.45E+00	1.58E-02	7.40E-01
F13	2	**1.35E-32**	2.23E-03	1.17E-02	8.31E-08	**1.35E-32**	1.49E-04	**1.35E-32**
	10	**9.96E-12**	7.30E-01	8.43E-02	1.06E-03	4.66E-01	9.66E-02	9.18E-03

	30	**4.31E-03**	2.85E+00	1.03E+00	3.03E-02	4.36E+00	1.98E+00	1.32E+00

Tabular results indicates that IBMO is far better at balancing between exploitation and exploration phases compared to other meta-heuristic algorithms. Each test is run a total of 30 times and the best optimization score was recorded for each algorithm.

Table 2: Comparison of meta heuristic algorithms with IBMO for fixed dimension multimodal functions.

Func	**IBMO**	**BMO**	**AOA**	**PSO**	**GWO**	**FPA**	**DFA**
F14	**9.98E-01**	3.80E+00	1.13E+01	9.98E-01	3.97E+00	**9.98E-01**	2.98E+00
F15	**5.98E-04**	5.64E-03	9.83E-04	1.63E-03	1.04E-03	6.86E-04	2.24E-03
F16	**-1.03E+00**	**-1.03E+00**	**-1.03E+00**	**-1.03E+00**	**-1.03E+00**	**-1.03E+00**	**-1.03E+00**
F17	**3.98E-01**	4.45E-01	3.98E-01	4.00E-01	**3.98E-01**	4.00E-01	**3.98E-01**
F18	**3.00E+00**	1.99E+01	**3.00E+00**	3.01E+00	3.01E+00	3.01E+00	3.01E+00
F19	**-3.86E+00**	-3.81E+00	-3.84E+00	-3.86E+00	**-3.86E+00**	-3.84E+00	-3.84E+00
F20	-3.20E+00	-1.55E+00	-2.94E+00	-3.20E+00	-2.59E+00	**-3.32E+00**	-2.84E+00
F21	**-6.03E+00**	-4.40E+00	-4.81E+00	-5.05E+00	-5.06E+00	-1.02E+01	-1.02E+01
F22	**-8.36E+00**	-3.67E+00	-3.72E+00	-1.04E+01	-1.04E+01	-1.04E+01	-3.72E+00
F23	**-7.02E+00**	-1.65E+00	-4.97E+00	-1.05E+01	-5.13E+00	-1.05E+01	-3.83E+00

3.4 Computational Cost Analysis

A time-based computational cost investigation has been conducted in this section with the IBMO algorithm. It can be determined by testing the algorithm on the benchmark test functions for CEC-2020 protocol. These functions are set as minimization problems with CEC01 to CEC03 having different dimensions while CEC04 to CEC10 functions, which are rotated and shifted, have a dimension value of 10. Four specific computational times, T_0,T_1,T_2,$\hat{T}_2$ are determined which establish the complexity of the algorithm, evaluation criteria for which is detailed in [30]. T_0 is the runtime of a specific mathematical algorithm, T_1 is the computational time of the CEC function which has been run 10,000 times, T_2 is the computational time of the technique used to solve the minimization problem of the CEC function in under 10,000 iterations and $\hat{T}_2$ is the mean value of 5 repetitions of the T_2 time analysis. It is executed 5 times to cater for the variations in the time of execution due to the probabilistic nature of the algorithms. The algorithms complexity is reflected by the following equation.

$$\overrightarrow{T} = \frac{\hat{T}_2 - T_1}{T_0} \tag{10}$$

The proposed algorithm of IBMO tested on these computational times and is compared with other similar metaheuristic algorithms on each CEC benchmark test function. The results of the experiment is shown in Table 3. The value of

T_0 was determined to be 0.1162 seconds. Since the algorithms have multi-agent solution finding technique, a population size of 10 was used for each algorithm in comparison. Rest of the parameters are same as used in the previous section. The machine used for the complete analysis is a Core i7 9750h with 16GB RAM.

Table 3: Comparison of meta heuristic algorithms with IBMO for $\overrightarrow{T}$ (sec) time complexity analysis

Func.	$\overrightarrow{T}$ **IBMO**	$\overrightarrow{T}$ **FDO**	$\overrightarrow{T}$ **BMO**	$\overrightarrow{T}$ **AOA**	$\overrightarrow{T}$ **PSO**	$\overrightarrow{T}$ **GWO**	$\overrightarrow{T}$ **FPA**	$\overrightarrow{T}$ **DFA**
CEC01	335.14	10213.71	329.9	316.47	317.4	317	464.89	594.96
CEC02	14.54	197.15	13.41	5.82	4.92	5.27	242.62	542.33
CEC03	18.96	254.5	18.83	10.6	10.33	10.58	274.56	2075.9
CEC04	14.31	342.83	12.12	3.88	3.72	3.89	153.3	303.76
CEC05	14.77	615.55	13.74	4.4	4.36	4.97	153.31	302.95
CEC06	125.59	3312.43	115.17	110.05	110.4	111.38	259.73	410.83
CEC07	15.07	646.37	11.99	4.37	4.24	4.4	149.01	294.98
CEC08	13.52	191.17	12.08	4.51	4.25	4.39	149.83	302.48
CEC09	13.78	232.48	12.05	3.9	4.29	4.07	151.19	301.13

It is of paramount importance to realize that the dimensions of the multi-agent algorithms i.e. to have well-balanced phases of exploration and exploitation phase in congruence with the convergence speed is a significant aspect in the evaluation of the novel IBMO algorithms. The exhaustive experimentation conducted in this section proves that the optimization process of the IBMO algorithm tends to localize towards a global search space solution for better results.

4 Discussion

Engineering optimization tasks involve finding many optimal solutions. With constrained involved the number of local optima solutions increase while the global optima seem to be hard to determine. The study conducted in this work provides a review in the field of multi-modal test function optimization with a comprehensive comparative analysis with other traditional and latest methods with an application on engineering design constraint problems. The improved method and other existing ones are analyzed in this paper. While most of the algorithms provide sufficiently acceptable results for low dimensional functions, IBMO is able to find a global optima solution for higher dimensional functions. The best results are written in bold for better visualization of each test function. The traditional BMO algorithm suffers from the drawback of low search accuracy and easy trapping of the solution onto a local state space. The paper strategizes two main techniques. Firstly, it introduces random movement variable during the exploration process to improve the reproduction process. Secondly, the movement of the Gooseneck barnacles for mating is added. The mathematical mimicry of this movement ensures the algorithm does not get stuck in a local optima solution.

5 Conclusion

This paper presents an efficient optimizer motivated by gooseneck barnacles to tackle the optimization problems. The qualitative investigation of the proposed algorithm consists of the following metrics, namely; search history, the path of the first dimension in the search space, fitness average, and speed of convergence curve of IBMO. Furthermore, the IBMO algorithm is assessed with 23 benchmark test functions involving uni-modal, multi-modal, fix-dimension multi-modal, and composite functions. The results conducted from the experiments demonstrate that IBMO assures the functioning of explorations while accomplishing higher exploitation within an acceptable convergence speed, thus keeping an exceptional balance between the exploitation phase and exploration phase. Statistically, the algorithm returns a higher performance average compared with the other meta-heuristic algorithms the suitability of IBMO's performance can be theoretically credited to the points as discussed below:

- The parameter penis size i.e. ps permits the IBMO to retain a constant disorder rate whilst also assuring fast convergence, therefore evading local optima solution traps.
- k guarantees the effectiveness of the early exploration and later exploitation.
- Based on historical data, suitable use of the discrete fitness values allows IBMO to find better positional variables that evidently allow better adaptability of the IBMO in different search phases.

Acknowledgements This work was supported by Top Research Centre Mechatronics (TRCM), University of Agder (UiA), Norway.

References

1. Abdel-Raouf, O., El-Henawy, I., Abdel-Baset, M., et al.: A novel hybrid flower pollination algorithm with chaotic harmony search for solving sudoku puzzles. International Journal of Modern Education and Computer Science **6**(3), 38 (2014)
2. Abualigah, L.: Group search optimizer: a nature-inspired meta-heuristic optimization algorithm with its results, variants, and applications. Neural Computing and Applications **33**(7), 2949–2972 (2021)
3. Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., Gandomi, A.H.: The arithmetic optimization algorithm. Computer methods in applied mechanics and engineering **376**, 113609 (2021)
4. Asselmeyer, T., Ebeling, W., Rosé, H.: Evolutionary strategies of optimization. Physical Review E **56**(1), 1171 (1997)
5. Buckeridge, J.S., Newman, W.A.: A revision of the iblidae and the stalked barnacles (crustacea: Cirripedia: Thoracica), including new ordinal, familial and generic taxa, and two new species from new zealand and tasmanian waters. Zootaxa **1136**(1), 1–38 (2006)
6. Chan, B.K., Dreyer, N., Gale, A.S., Glenner, H., Ewers-Saucedo, C., Pérez-Losada, M., Kolbasov, G.A., Crandall, K.A., Høeg, J.T.: The evolutionary diversity of barnacles, with an updated classification of fossil and living forms. Zoological Journal of the Linnean Society **193**(3), 789–846 (2021)

7. Dong, Y., Hou, J., Zhang, N., Zhang, M.: Research on how human intelligence, consciousness, and cognitive computing affect the development of artificial intelligence. Complexity **2020**, 1–10 (2020)
8. Dorigo, M., Stützle, T.: Ant colony optimization: overview and recent advances. Springer (2019)
9. Du, H., Wu, X., Zhuang, J.: Small-world optimization algorithm for function optimization. In: Advances in Natural Computation: Second International Conference, ICNC 2006, Xi'an, China, September 24-28, 2006. Proceedings, Part II 2. pp. 264–273. Springer (2006)
10. Hatamlou, A.: Black hole: A new heuristic optimization approach for data clustering. Information sciences **222**, 175–184 (2013)
11. Igel, C.: No free lunch theorems: Limitations and perspectives of metaheuristics. Theory and principled methods for the design of metaheuristics pp. 1–23 (2014)
12. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks. vol. 4, pp. 1942–1948. IEEE (1995)
13. Mafarja, M., Mirjalili, S.: Whale optimization approaches for wrapper feature selection. Applied Soft Computing **62**, 441–453 (2018)
14. Mirjalili, S.: Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural computing and applications **27**, 1053–1073 (2016)
15. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Advances in engineering software **69**, 46–61 (2014)
16. Mirjalili, S., Mirjalili, S.: Genetic algorithm. Evolutionary Algorithms and Neural Networks: Theory and Applications pp. 43–55 (2019)
17. Mirjalili, S.Z., Mirjalili, S., Saremi, S., Faris, H., Aljarah, I.: Grasshopper optimization algorithm for multi-objective optimization problems. Applied Intelligence **48**, 805–820 (2018)
18. Nadimi-Shahraki, M.H., Taghian, S., Mirjalili, S., Faris, H.: Mtde: An effective multi-trial vector-based differential evolution algorithm and its applications for engineering design problems. Applied Soft Computing **97**, 106761 (2020)
19. Oestreicher, C.: A history of chaos theory. Dialogues in clinical neuroscience (2022)
20. Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S.: Gsa: a gravitational search algorithm. Information sciences **179**(13), 2232–2248 (2009)
21. Rezaee Jordehi, A.: A chaotic-based big bang–big crunch algorithm for solving global optimisation problems. Neural Computing and Applications **25**, 1329–1335 (2014)
22. Simon, D.: Biogeography-based optimization. IEEE transactions on evolutionary computation **12**(6), 702–713 (2008)
23. Stark, A.E.: The hardy-weinberg principle. Genetics and molecular biology **28**, 485–485 (2005)
24. Sulaiman, M.H., Mustaffa, Z., Saari, M.M., Daniyal, H.: Barnacles mating optimizer: A new bio-inspired algorithm for solving engineering optimization problems. Engineering Applications of Artificial Intelligence **87**, 103330 (2020)
25. Wang, C., Koh, J.M., Yu, T., Xie, N.G., Cheong, K.H.: Material and shape optimization of bi-directional functionally graded plates by giga and an improved multi-objective particle swarm optimization algorithm. Computer Methods in Applied Mechanics and Engineering **366**, 113017 (2020)
26. Wang, L., Wang, X., Liu, K., Sheng, Z.: Multi-objective hybrid optimization algorithm using a comprehensive learning strategy for automatic train operation. Energies **12**(10), 1882 (2019)

27. Xing, B., Gao, W.J., Xing, B., Gao, W.J.: Fruit fly optimization algorithm. Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms pp. 167–170 (2014)
28. Yang, X.S., Karamanoglu, M., He, X.: Multi-objective flower algorithm for optimization. Procedia Computer Science **18**, 861–868 (2013)
29. Yang, X.S., Karamanoglu, M., He, X.: Flower pollination algorithm: a novel approach for multiobjective optimization. Engineering optimization **46**(9), 1222–1237 (2014)
30. Yue, C., Price, K.V., Suganthan, P.N., Liang, J., Ali, M.Z., Qu, B., Awad, N.H., Biswas, P.P.: Problem definitions and evaluation criteria for the cec 2020 special session and competition on single objective bound constrained numerical optimization. Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, Tech. Rep **201911** (2019)
31. Zhang, J., Xiao, M., Gao, L., Pan, Q.: Queuing search algorithm: A novel metaheuristic algorithm for solving engineering optimization problems. Applied Mathematical Modelling **63**, 464–490 (2018)