International Journal of Electrical and Computer Engineering (IJECE)
Vol. 13, No. 4, August 2023, pp. 4317∼4338
ISSN: 2088-8708, DOI: 10.11591/ijece.v13i4.pp4317-4338

Rhizostoma optimization algorithm and its application in different real-world optimization problems

Esraa A. Mahareek[1], Mehmet Akif Cifci[2,3], Habiba A. El-Zohni[1], Abeer S. Desuky[1]
[1]Department of Mathematics, Faculty of Science, Al-Azhar University, Cairo, Egypt
[2]Department of Computer Engineering, Bandirma Onyedi Eylul University, Balikesir, Turkey
[3]Informatics, Klaipeda State University of Applied Sciences, Klaipeda, Lithuania

Article Info

Article history:

Received Jun 5, 2022
Revised Dec 7, 2022
Accepted Dec 21, 2022

Keywords:

Meta-heuristic algorithms
Rhizostoma optimization algorithm
Data reduction
Student performance prediction
Classical engineering problems

ABSTRACT

In last decade, numerous meta-heuristic algorithms have been proposed for dealing the complexity and difficulty of numerical optimization problems in the real-world which is growing continuously recently, but only a few algorithms have caught researchers' attention. In this study, a new swarm-based meta-heuristic algorithm called Rhizostoma optimization algorithm (ROA) is proposed for solving the optimization problems based on simulating the social movement of Rhizostoma octopus (barrel jellyfish) in the ocean. ROA is intended to mitigate the two optimization problems of trapping in local optima and slow convergence. ROA is proposed with three different movement strategies (simulated annealing (SA), fast simulated annealing (FSA), and Levy walk (LW)) and tested with 23 standard mathematical benchmark functions, two classical engineering problems, and various real-world datasets including three widely used datasets to predict the students' performance. Comparing the ROA algorithm with the latest meta-heuristic optimization algorithms and a recent published research proves that ROA is a very competitive algorithm with a high ability in optimization performance with respect to local optima avoidance, the speed of convergence and the exploration/exploitation balance rate, as it is effectively applicable for performing optimization tasks.

Corresponding Author:

Esraa A. Mahareek
Department of Mathematics, Faculty of Science, Al-Azhar University
Cairo, Egypt
Email: esraa.mahareek@azhar.edu.eg

1. INTRODUCTION

Over the last few decades meta-heuristic optimization techniques became very popular. Algorithms such as the grey wolf optimizer (GWO) [1], the whale optimization algorithm (WOA) [2], the Jellyfish search (JS) [3], the ant lion optimizer (ALO) [4], the archimedes optimization algorithm (AOA) [5], and particle swarm optimization (PSO) [6] are successful examples for these techniques which used efficiently in scientific research. Five main reasons make the meta-heuristics remarkably common [7]: simplicity because of the very simple concepts their inspiration based on, flexibility because of the stability in their structure with the applicability in different issues, derivation-free mechanism because of the random solution(s) their process starts with [8], and finding the optimum do not need any calculations to derive the search spaces, local optima avoidance due to their stochastic nature which allows them to avoid falling into local solutions and getting to the global solution, beside the ease of their implementation [3]. These make meta-heuristics considering as highly suitable and a good option for real problems.

Any meta-heuristic algorithm based on two main processes: exploration and exploitation. Exploration is the process that performed by an algorithm to evaluate the selected solutions. This process aims at falling into the local optimum trap. Exploitation is performing a search in the space close to the current solution(s). It can be figured as a local search. The way in which the meta-heuristics balance these two main processes is the challenge that differentiate a meta-heuristic from another [7]. The main aim of this study is to develop a novel meta-heuristic by inspiring the social behavior of Rhizostoma octopus in the ocean, including their food searching, and their moves inside the swarm. This algorithm provides an improvement for exploration/exploitation balance, local optima avoidance, and and the speed rate of convergence, compared to the latest algorithms.

This research paper is arranged as follows. A literature review on meta-heuristic optimization algorithms is presented in section 2. Section 3 describes the proposed artificial Rhizostoma optimizer and explains how to implement it in detail. Section 4 verifies the efficiency of the proposed algorithm by solving various benchmark functions. Section 5 describes Rhizostoma optimization algorithm (ROA) in solving two classical engineering problems. Section 6 describes the ROA in real problems. The last section expresses conclusions.

2. LITERATURE REVIEW

There are about more than 1,000 publications on meta-heuristics in the last 33 years, and the proposal of swarm intelligence (SI) concepts was firstly appeared in 1993 [2]. All meta-heuristics are categorized into two groups. The first group consists of the algorithms that mimic physical or biological phenomena which can be classified into three subgroups, The first is evolution-based which is inspired by the laws of natural evolution. The most popular evolution-inspired method is genetic algorithms (GA) which based on the Darwinian evolution simulation. Other popular algorithms are evolution strategy (ES), probability-based incremental learning (PBIL) and others [2]. The second subgroup is physics-based that imitates the physical rules in the universe. The most popular algorithms in this subgroup are simulated annealing (SA) [9], big-bang big-crunch (BBBC) [10], black hole (BH) algorithm and others. The third subgroup is swarm-based methods [11] that mimic the social behavior of the groups of animals. The most popular algorithm of this subgroup is PSO [6] that is inspired by the social behavior of bird flocking, monkey search (MS), firefly algorithm (FA) [12], grasshopper optimization algorithm (GOA) [13], bat-inspired algorithm (BA) [14], ALO [4], ant colony optimization (ACO), GWO, AOA, giza pyramids construction (GPC) [15], The WOA [2], heap based optimizer (HBO) [16] and political optimizer (PO) [17]. The second category [2] contains those are inspired by human phenomena, social-based algorithm (SBA), harmony search (HS), group counselling optimization (GCO) algorithm, and the exchange market algorithm (EMA) are some of the most popular algorithms of this category.

The search process in meta-heuristic algorithms is divided into two bands [18]: exploration and exploitation. Operators are vital for any optimizer to globally explore the search space: in this band (exploration), movements should be randomized. The second band is exploitation, it can be defined as the process in which the promising area (s) of the search space which is (are) investigated in details. Exploitation hence related to the capability of searching locally in the design space' promising regions which exists in the exploration phase. Searching to find the appropriate balance ratio between the exploration and the exploitation is the most challenging problem in the development stage of any meta-heuristic algorithm because of the stochastic nature of the optimization' process [19].

Many SI techniques are proposed so far, which are inspired by behaviors of hunting and search. For our information so far, however, there is no SI technique in this literature review that mimics the movement and food search for Rhizostoma Pulmo. This motivated us to propose a new SI artificial algorithm inspired by Rhizostoma octopus called ROA and explore its abilities to solve the benchmark equations, both exploration and exploitation are considered in our algorithm. In the beginning, we describe the Rhizostoma dive pattern in the exploration for the maximum concentration of prey in the water column. Over time, Rhizostoma Pulmo's movements switch to forming a swarm and searching food moving inside swarms for exploitation.

3. ARTIFICIAL RHIZOSTOMA OPTIMIZER

This section describes the Rhizostoma octopus optimization algorithm inspiration. The mathematical model which describes his movement in the ocean is provided. Also, his way of choosing the best place to find his food, especially because he eats a lot of food.

3.1. Inspiration

Barrel jellyfish found in the southern and western shores of the British Isles in summer months, they are solid with a thick, mushroom-shaped bell which diameter can grow up to 1 m with 8 frilly arms beneath its surface, hence they have the former name: Rhizostoma octopus, giants of the jellyfish world, they are referred to as “dustbin-lid jellyfish” or “frilly-mouthed jellyfish” and so large that small crabs and young fish are looking for a protective shelter in their tentacles. Despite their large size, humans have little to worry about from Rhizostoma pulmo. Their sting causes no ill effects because it is so mild even some people do not feel it.

Jellyfish methods of feeding vary, some of them bring food to their mouths using their tentacles whereas filter-feeding is used by others [20], so they eat whichever the current brings. Rhizostoma pulmo vigorously hunts prey and freezes them via the sting by their tentacles. R. Octopus have features that enable them to control their movements, whereas other species of jellyfish which mostly drift in the water depending on currents and tides. Rhizostoma octopus (R. octopus) has the capability of active swim against localized currents because its unusually large weighing 27 kg or more Figure 1 shows its size. A rising sea temperature and amount of food help the formation of their swarms which are called blooms because they survive better than the other ocean animals under such circumstances as high salinity and low oxygen concentrations. Through studying the R. octopus, Reynolds [21] found that it uses different strategies for searching to find huge quantities of plankton in large levels of concentration. Three those unique strategies are proposed in this work.

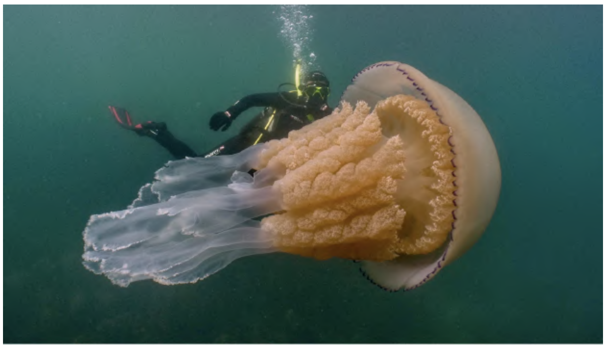

Figure 1. Effects of selecting different switching under dynamic condition

The first is based on SA which is a local search method invented to avoid local minima and converges to global optimum. the second is based on a Levy walk (LW) where an agent searches for something in the nearby areas then it makes random jumps to more faraway places if there is nothing locally, another random jump will make a farther distance away. Scientists have found that this strategy is used by other animals, such as honeybees and sharks, to gain big effect because it results in rapidly finding one of several possible suitable prey species. But because R. octopus is more discerning about finding prey it makes a sudden up or down movements in the water if there is a food present and if not, returns to its past position. Reynolds pointing out that as a form of fast SA-a kind of LW search pattern and it is very effective when used for quickly finding a specific target in a large search space in a block of noise. R. octopus consumes huge quantities of plankton regularly enough to meet its needs and Its uniqueness comes from its techniques which have never been observed in any other animal [21].

The quality of food varies in the ocean; thus, R. octopus is using more than one powerful stochastic search algorithm so, it has the ability to locate a global maximum (best resource) that is hidden through multiple poorer local maxima (poorer resources) in the search space. Therefore, we developed a new algorithm that is inspired by the search behavior and movement of R. octopus in the ocean model, the algorithm iteration represented in Figure 2. In the next subsection, the behavior and movements of R. octopus in the ocean are simulated and introduced as a mathematical model and the optimization algorithm that is driven from this mathematical simulation is also introduced.

3.2. Mathematical model for ROA

The proposed algorithm is based on three basic rules: i) exploration and exploitation bands: Rhizostoma pulmo moves in the ocean searching for the best food locations (plankton in vast amounts) using more than one random searching algorithm then form a swarm with forming swarm and feeding motion. In the artificial ROA the two main bands of a meta-heuristic algorithm are considered. The first type of motion is exploration, and the second type of motion inside the swarm is exploitation; ii) Rhizostoma searches for food or moves inside the swarm, and a "motion control factor" governs the switching between these types of movements; and iii) the quality of food found (current solution) is measured by the current location and its associated objective function.

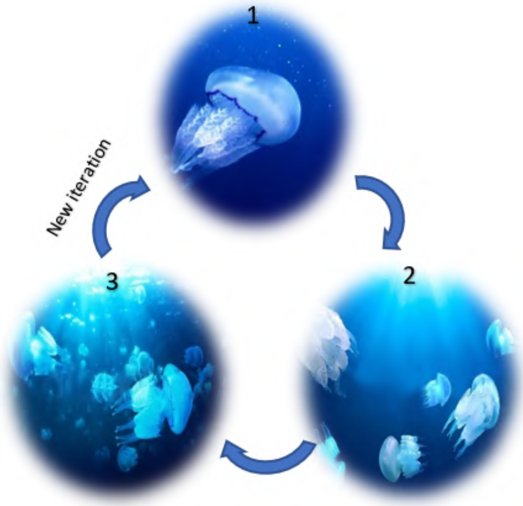

Figure 2. ROA iteration cycle

3.2.1. Population initialization and boundary conditions

The population of R. octopus is randomly initialized.

$$x_i^* = \lambda x_i(1 - x_i), 0 \leq x_0 \leq 1 \tag{1}$$

The location value of the i-th R. octopus is x_i; x_0 is the generated initial population of R. octopus, x_0 belongs to 0 and 1, $x_0 \notin \{0, 0.25, 0.5, 0.75, 1\}$ and parameter λ is set to 4.0 when a R. octopus exceeds the boundaries of the search area, it will re-enter the opposite bound. In (2) illustrates this returning process.

$$X_{i,d}^* = \begin{cases} (x_{i,d} - Ub_d) + Lb_d, if x_{i,d} > Ub_d \\ (x_{i,d} - Lb_d) + Ub_d, if x_{i,d} < Lb_d \end{cases} \tag{2}$$

$x_{i,d}$ is the R. octopus' location in the d-th dimension; $x_{i,d}^*$ is the location updated after checking the constraints of boundary. respectively, Ub_d and Lb_d are the upper and lower bounds of the d-th dimension in the search spaces.

3.2.2. Food searching strategies

According to [21] the movement pattern of R. octopus changes over time, he proposed fast simulated annealing (FSA) for its pattern while scientists thought he was moving using LW like sharks and honeybees, in all types of this food searching patterns the generative function exists and directs the way of updating the variables in each search attempt. In our proposal we introduce ROA based on three different stochastic search algorithms: LW, fast simulated annealing (food searching strategies), and SA then compare results to find the best strategy for its motion. The first strategy is levy flight which is named after the French mathematician Paul Levy, [14] it is a random walk in which the lengths of steps have a fibrous distribution. When defined as a walk in a space of greater than one dimension, the taken steps are in random, rival directions. Where x_i is the R. octopus currently with the best location, x_j is the new location of R. octopus. If $f(x_j) > f(x_i)$ The R. octopus at x_i will take a step toward the point x_j, This step satisfies a heavy-tailed Lévy distribution, which can be represented by a clear power-law as (3):

$$L(s) \approx |s|^{1-\beta}, where\ 0 < \beta \leq 2 \quad (3)$$

where s is random Lévy step length which can be calculated by $s = u/|v|^{1/\beta}$ where u and v are drawn from normal distributions. The position-updating with maximum consideration of food can be formulated as (4).

$$x_{next} = x_i + \mu \times rand(0, 1) \quad (4)$$

where μ is the step length, rand is a random number distributed uniformly between (0, 1) and $\mu \times rand(0, 1)$ is the actual random walks or flights drawn from Levy distribution. Second motion strategy is fast simulated annealing algorithm (FSA) [22] it is kind of LW where a cost function, e.g., the prey density $f(x)$ e.g., the density of prey, at x (the current position), is compared with $f(y)$, which is the cost function at another position y, while from a Cauchy distribution, the step length, $s = |y - x|$, is randomly drawn as (5).

$$p(s) = 1/\pi \times T/(s^2 + T^2) \quad (5)$$

where T is the temperature which measures of the size of the fluctuations in step-length. The accepted probability value required to accept the new position is calculated as (6),

$$P = min\{1, exp(\Delta f/T)\} \quad (6)$$

where $\Delta f = f(y) - f(x)$ is the change in cost between the two positions current and previous. The acceptance condition the new position is that it has to be better than the previous position, i.e., if $\Delta f > 0$ will be accepted only if the new position is the worse, then R. octopus returns to its previous position. In [17] showed that FSA is converging to the global optima (maximum) if the annealing is scheduled in terms of $T(k) = T_0/k$, where the initial temperature is T_0 and the step counter is k.

Third R. octopus motion strategy is SA different kind of LW search pattern where the length of each step is generated randomly from a Gaussian distribution.

$$g(s) = (2\pi T)^{-D/2} e^{(\Delta X^2)/2T} \quad (7)$$

where D is dimension of the search space (number of variables in the cost function) and Δx shows the rate of change of X (variables' vector). So $x_{next} = x_i + \Delta x$ where x_i denotes the current state and x_{next} shows the next state of variables.

3.2.3. R. Octopus swarm

In Rhizostoma swarm, there are two types of motions when the swarm is just formed the first type is feeding motion and the second type is forming swarm [21]. R. octopus is moving with the first type when the swarm has just been formed. Over time, they change to the second type of motion [23], [24].

The first type of motion is the motion of R. octopus around their own locations and the corresponding updated location of each one of them is given by (8).

$$x_i(t + 1) = x_i(t) + \beta \times rand(0, 1) \times (ub - lb) \quad (8)$$

where ub and lb are the upper bound and lower bound of search spaces, respectively; and $\beta > 0$ is a motion coefficient, according to the analysis results of the evaluation experiments we found that the best result of ROA is obtained when $\beta = 0.1$, it is depends on the motion' length around R. octopus locations.

To simulate the second type of motion,R_j, R_i random and for determining the direction of movement, a vector from R_i to R_j is used. When the food' quantity at the R_j location exceeds the Ri location, R_i moves toward R_j; if the available food' quantity to the selected R_j is lower than the quantity available to R_i it moves away from it directly. So, each one moves toward the best direction to forming a swarm. The updated location of a R. octopus and its direction of motion is simulated in (9):

$$\vec{S} = x_i^{(t+1)} - x_i^t \quad (9)$$

$$Where \vec{S} = rand(0, 1) \times \vec{D} \quad (10)$$

$$\vec{S} = \begin{cases} x_j^t - x_i^t \; if \; f(x_i) \geq f(x_j) \\ x_i^t - x_j^t \; if \; f(x_i) < f(x_j) \end{cases} \tag{11}$$

where f is an objective function of location x, $\vec{S}$ is the step, and $\vec{D}$ is the direction. Hence,

$$x_i^{(t+1)} = x_i^t + \vec{S} \tag{12}$$

3.2.4. Motion control factor

A motion control factor is used to determine the type of motion. It controls not only the two types of swarm motions but also the searching food strategies of R. octopus. Rhizostoma pulmo is attracted to the locations that have a huge quantity of planktons, over time, more than one closed together and forming a swarm with the two types of motions that we introduced before.

The motion control factor is introduced to regulate this situation that includes a motion control function $M(f)$ is a constant that varies randomly from 0 to 1. m_0 is a constant equal to 0.5 which is the mean of zero and one when the value of $M(f)$ is less than m_0 the R. octopus follow one food search strategy. Else, they move inside the swarm.

$$M(f) = |1 - exp((t-1)/t_{max})(2 \times rand(0,1) - 1)| \tag{13}$$

where t is the iteration number and t_{max} is the maximum number of iterations, which is an initialized parameter. The pseudocode of ROA represented in Figure 3. Exploration/exploitation balance can be obtained by the adaptive values of $M(f)$ that allows ROA to transit smoothly between the exploration process and the exploitation process and equally divide the iterations between the two processes.

Pseudocode of ROA Algorithm

```
Begin
    Initialize the parameters, population size n, maximum no. of iteration t_max,
    and population of R. octopuses.
    x_i = (1, 2, ..., n),
    Calculate the fitness of each search agent f(x_i)
    X* is the best current location.
    Initialize iteration t = 1
    Repeat
        For i = 1: n_Pop do
        Calculate the motion control factor M(f) using Eq. (13)
        If M(f) ≥ 0 · 5 R. octopus follows one of the searching food strategies
        Else: R. octopus moves inside a swarm
            If rand (0,1) > (1 − M(f)): R. octopus exhibits first type of motion.
              • Update the position of the current search agent using Eq. (8)
            Else: R. octopus exhibits second type of motion.
              • Determine R. octopus direction using Eq. (11)
              • Update the position of the current search agent using Eq. (12)
           End if
        End if
        Check if any search agent goes beyond the search space and ament it.
        Calculate the fitness of each search agent.
        Update X* if there is a better solution.
        End for i
        Update the iteration t = t + 1
    Until stop criterion is met (t > t_max)
    Output the best result and visualization.
End
```

Figure 3. Pseudocode of ROA

4. RESULTS AND PERFORMANCE EVALUATION

A strong optimization algorithm should be able to (1) explore the search space, the (2) exploit the promising areas, and (3) presents a good balance between (1) and (2), (4) converge to the best solution quickly. The optimization efficiency of the proposed ROA algorithm was tested in this study by using 23 classical mathematics benchmark functions [25], [12] which are usually classified to three groups uni, multi, and fixed-dimension multi-modal. The summarizes of the test problems reporting the cost function number, variation range of the variables of optimization and the optimal value mentioned in literature f_{min}, note that "Dim" represents the counted design variables can be found in detailed in [2]. For each algorithm in our experiments, 30 is the population size used and 500 is the maximum iteration that has been utilized.

The proposed ROA and the other algorithms were run 30 times per each benchmark function, starting from different random populations then, the average statistical results (cost function and the associated standard deviation) are recorded in Tables 1 to 7 ROA was compared with latest optimization algorithms PSO, SA, JS [3], GWO [1], WOA [2], AOA [5], HBO [16], GPC [15] and PO [17]. note that PSO, GWO, and WOA results are taken from [1], [2]. For any pair of compared algorithms, the better results are highlighted in bold.

Table 1. Set internal parameter values for comparison with metaheuristic algorithms

Algorithm	Parameters
ROA	n =30; $t_{max} = 500$; M(f) from corresponding equations
GWO	n =30; $t_{max} = 500$; problem dimension; search domain
SA	n =30; $t_{max} = 500$; temperature (t) decreasing with time
JS	n=30; $t_{max} = 500$
WOA	n =30; $t_{max} = 500$;
AOA	C1 = 2; C2 = 6;C3 = 2 and C4 = 0.5 (CEC and engineering problems)
GPC	n=30;$t_{max} = 500$;Gravity=9.8; Angle of ramp=10; Initial velocity=rand(0,1); Minimum Friction=1; Maximum Friction=10; Substitution Probability=0.5
PSO	n =30; $t_{max} = 500$; Inertia weight = 0.5; Personal learning coefficient = 2.05; Inertia weight damping ratio= 0.99; Global learning coefficient = 2.05
HBO	n=30; [C, p1, p2] from corresponding equations.
PO	n=30

Table 2. The compared optimization results for F1:F4 mathematical functions

Algorithm	F1		F2		F3		F4	
	Ave	Std	Ave	Std	Ave	Std	Ave	Std
ROA-SA	0	0	0	0	0	0	0	0
ROA-FSA	1.259e-90	3.9276e-90	0.021236	5.4714e-05	0.0001136	4.238e-07	0.0094079	0.0014109
ROA-LW	0.0011344	0.0026719	0.087454	0.099995	3.3713	10.0688	0.28624	0.5239898e-4
GWO	6.59E-28	6.34E-05	7.18E-17	0.029014	3.29E-06	79.14958	5.61E-07	1.315088
SA	7.366e-7	4.0345e-6	2.216e-7	1.2139e-6	2.576e-7	1.4113e-6	3.541e-5	1.9398e-4
JS	3.8532e-30	6.3139e-30	4.1282e-13	9.9390e-13	1.0924	2.1761	2.9258e-18	3.2299e-18
WOA	1.41e-30	4.91e-30	1.06E21	2.39E21	5.39E07	2.93E06	0.072581	0.39747
AOA	5.8259e-85	1.8423e-84	2.4433e-48	7.4123e-48	7.4123e-48	8.7804e-71	2.7766e-70	1.0139e-40
GPC	2.4615e-24	1.3482e-23	5.8742e-13	3.2174e-12	4.1289e-23	2.2615e-22	2.9608e-13	1.6217e-12
PSO	0.000136	0.000202	0.042144	0.045421	70.12562	22.11924	1.086481	0.317039
HBO	3.9645e-06	1.1644e-05	4.0886e-06	2.9734e-06	25435.9324	8466.5845	13.7876	4.1238
PO	2.0212e-32	3.8288e-32	1.6587e-17	3.1293e-17	8.099e-23	2.4476e-22	1.1124e-14	2.8893e-14

Table 3. The compared optimization results for F5:F8 mathematical functions

Algorithm	F5		F6		F7		F8	
	Ave	Std	Ave	Std	Ave	Std	Ave	Std
ROA-SA	3.2831e-24	8.0155e-24	.5309e-07	1.1422e-06	6.7809e-05	5.812e-05	-12569.4853	0.0023895
ROA-FSA	0	0	0.26078	6.8948e-05	N/A	N/A	N/A	N/A
ROA-LW	5.5138	7.9698	6.2351e-07	6.9182e-7	0.0063961	0.0050955	-12565.538	2.5028
GWO	26.81258	69.90499	0.816579	0.000126	0.002213	0.100286	-6123.1	-4087.44
SA	1.719e-4	9.4183e-4	4.0111e-5	2.1970e-04	0.1322	0.7242	-395.9531	2.1687e+03
JS	0.1629	0.3146	5.8572e-07	6.8134e-07	4.7949e-04	3.4541e-04	-9.3256e+3	1.2873e+03
WOA	27.86558	0.763626	3.116266	0.532429	0.001425	0.001149	5080.76	695.7968
AOA	28.8790	0.0591	42.14912	10.32415	47.01694	7.427119	46.28484	9.35206
GPC	0.96104	5.2638	0.21498	1.1775	2.0868e-06	1.143e-05	-1.333	7.3013
PSO	96.71832	60.11559	0.000102	8.28E05	0.122854	0.044957	4841.29	1152.814
HBO	102.29	107.6586	5.8736e-07	1.0638e-06	0.033578	0.008506	-11709.1053	345.0109
PO	1.4333e-27	2.8463e-27	308.2064	720.8046	0.0013103	0.0010812	-12214.1716	1123.6047

Unimodal functions F1 to F7 have one global minimum. These functions are used sufficiently for testing the rate of convergence and the exploitation ability of optimization algorithms. Results in Tables 2 and 3 shows that the proposed ROA has a high competitiveness with other method used in the comparison. In particular, for F1 to F5, only ROA is able to provide the exact optimum value. Moreover, the ROA algorithm is the most efficient optimizer for functions F6 and F7 in terms of average and standard deviation. As a result, the ROA algorithm has a superior exploitation capability.

Table 4. The compared optimization results for F9:F12 mathematical functions

Algorithm	F9		F10		F11		F12	
	Ave	Std	Ave	Std	Ave	Std	Ave	Std
ROA-SA	0	0	8.8818e-16	0	0	0	1.5705e-32	2.885e-48
ROA-FSA	N/A	N/A	0.014692	0.0040639	1.117e-05	4.5231e-06	0.007692	0.004922
ROA-LW	17.8287	19.7396	0.049404	0.11383	0.090238	0.22451	1.7396e-07	3.3397e-7
GWO	0.310521	47.35612	1.06E-13	0.077835	0.004485	0.006659	0.053438	0.020734
SA	1.3598	7.4478	1.5507e-05	8.4936e-05	3.2927e-04	0.0018	4.1038e-10	2.2477e-09
JS	0.0031	0.0034	1.723e-14	2.285e-14	0	0	4.1615e-9	4.7539e-9
WOA	0	0	7.4043	9.897572	0.000289	0.001586	0.339676	0.214864
AOA	37.71428	7.8072005	45.94354	12.735777	0	0	0.832024	0.1679651
GPC	0	0	5.6755e-14	3.1086e-13	0	0	0.040501	0.22183
PSO	46.70423	11.62938	0.276015	0.5090	0.009215	0.007724	0.006917	0.026301
HBO	13.2669	6.5803	0.00011683	4.3068e-05	0.0027127	0.0045679	0.010367	0.032783
PO	5.6843e-15	1.7975e-14	1.2434e-15	1.1235e-15	0	0	0.634	1.3369

The multimodal functions F8 to F13 differ from the unimodal functions, in that they have a large number of local minima. As a result, these kinds of benchmark functions are better for testing the exploration capability and avoidance of local optima of algorithms. Fixed-dimensional multimodal functions F14 to F23 have a pre-defined number of design variables and provide different search areas compared to the multimodal functions. Tables 4 to 7 show that ROA provides the exact optimum value for multimodal F9, F11 and F18. Moreover, the ROA algorithm is the most efficient optimizer for functions F8, F10, F12 and F15 in terms of average and standard deviation. For the rest functions ROA is competitive with algorithms that achieved the better performance. Thus, ROA has also a high exploration capability which leads this algorithm to explore the promising regions without any disruption. The convergence curves of the ROA with GWO, WOA, PO, HBO, JS and GPC algorithms for some of the functions by considering the maximum number of $iterations$=500 are shown in Figures 4, 5, and 6, and Table 1. represents each other internal parameters used in these examples. The iterations are shown on the horizontal axis while the average function values are shown on the vertical axis. As can be observed, our proposal ROA algorithm can escape from the local optima and converges faster to the optimal solution with the best balance between exploration and exploitation as compared to other optimization algorithms.

Table 5. The compared optimization results for F13:F16 mathematical functions

Algorithm	F13		F14		F15		F16	
	Ave	Std	Ave	Std	Ave	Std	Ave	Std
ROA-SA	1.3498e-32	2.885e-48	0.998	1.282e-16	0.0003075	2.8303e-08	-1.0316	1.282e-16
ROA-FSA	5.0621e-06	6.327e-06	12.6705	2.0512e-15	N/A	N/A	N/A	N/A
ROA-LW	0.0083598	0.01481	0.998	1.855e-11	0.00031724	1.9206e-5	-1.0316	1.480e-16
GWO	0.654464	0.004474	4.042493	4.252799	0.000337	0.000625	-1.0316	-1.03163
SA	8.0455e-09	4.4067e-08	0.0333	0.1822	1.0250e-05	5.6139e-05	-0.0344	0.1883
JS	3.0149e-8	3.0230e-8	0.998	0	3.075e-04	8.5140e-8	-1.0316	1.046e-16
WOA	1.889015	0.266088	2.111973	2.498594	0.000572	0.000324	-1.0316	4.2E07
AOA	2.925065	0.0332898	1.25662	0.445446	8.0972e-04	2.5216e-04	-1.0314	0.0004611
GPC	0.087356	0.47847	0.42235	2.3133	1.8111e-05	9.9197e-05	-0.034341	0.18809
PSO	0.006675	0.008907	3.627168	2.560828	0.000577	0.000222	-1.0316	6.25E16
HBO	3.2871e-07	4.8869e-07	0.998	0	0.00070161	0.00011168	-1.0316	7.4015e-17
PO	1.3498e-32	2.885e-48	0.998	7.4015e-17	0.00031082	4.7609e-06	-1.0316	2.2676e-12

Table 6. The compared optimization results for F17:F20 mathematical functions

Algorithm	F17		F18		F19		F20	
	Ave	Std	Ave	Std	Ave	Std	Ave	Std
ROA-SA	0.39789	0	3	1.819e-15	-0.30048	7.4908e-08	-3.322	5.1672e-07
ROA-FSA	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
ROA-LW	0.39789	0	3	1.594e-15	-0.30048	2.7102e-8	-3.322	7.4907e-8
GWO	0.397889	0.397887	3.000028	3	-3.86263	-3.86278	-3.28654	-3.25056
SA	N/A	N/A	0.1000	0.5477	-0.1288	0.7052	-0.1107	0.6065
WOA	0.397914	2.7E05	3	4.22E15	3.85616	0.002706	2.98105	0.376653
JS	0.3979	0	3	1.247e-15	-0.3005	0	-3.3220	3.1727e-5
AOA	N/A	N/A	3.779	1.9483	-3.841	0.030059	-2.977	0.20285
GPC	0.019086	0.10454	0.1	0.54772	-0.11137	0.60998	-0.03138	0.17188
PSO	0.397887	0	3	1.33E15	3.86278	2.58E15	3.26634	0.060516
HBO	0.39789	0	3	4.9096e-16	-3.8628	9.3622e-16	-3.322	4.6811e-16
PO	0.39789	2.3231e-13	7.6612	8.7635	-3.8628	3.7695e-08	-3.2982	0.050127

Table 7. The compared optimization results for F21:F23 mathematical functions

Algorithm	F21		F22		F23	
	Ave	Std	Ave	Std	Ave	Std
ROA-SA	**-10.1532**	**1.0447e-07**	**-10.4029**	**1.0431e-06**	**-10.5364**	**4.3607e-07**
ROA-FSA	-9.6872	0.63502	-10.1141	0.34647	-10.2505	0.36433
ROA-LW	-10.1501	0.0054771	-10.3873	0.042422	-10.5328	0.0091774
GWO	-10.1514	-9.14015	-10.4015	-8.58441	-10.5343	-8.55899
SA	-0.3384	1.8537	-0.3468	1.8993	-0.3512	1.9237
JS	**-10.1532**	3.0903e-6	**-10.4029**	**1.963e-15**	**-10.5364**	**9.030e-12**
WOA	7.04918	3.629551	8.18178	3.829202	9.34238	2.414737
AOA	-5.6074	2.2714	-6.5904	2.5396	-6.8493	2.36
GPC	-0.033799	0.18513	-0.039591	0.21685	-0.094438	0.51726
PSO	6.8651	3.019644	-8.45653	3.087094	-9.95291	1.782786
HBO	-9.2346	1.8725	**-10.4029**	**2.1349e-15**	**-10.5364**	**2.581e-15**
PO	**-10.1532**	**1.2713e-13**	**-10.4029**	2.1289e-11	**-10.5364**	1.224e-11

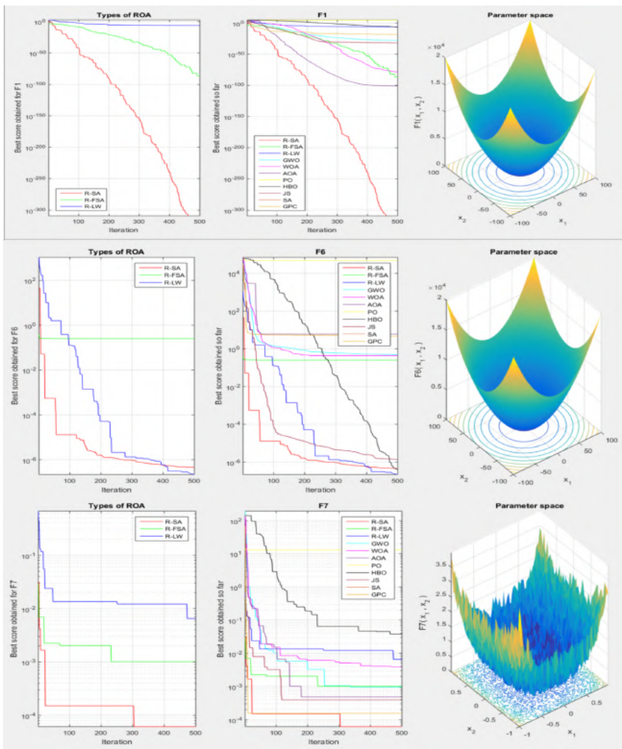

Figure 4. 3D and 2D representation of the unimodel benchmark functions

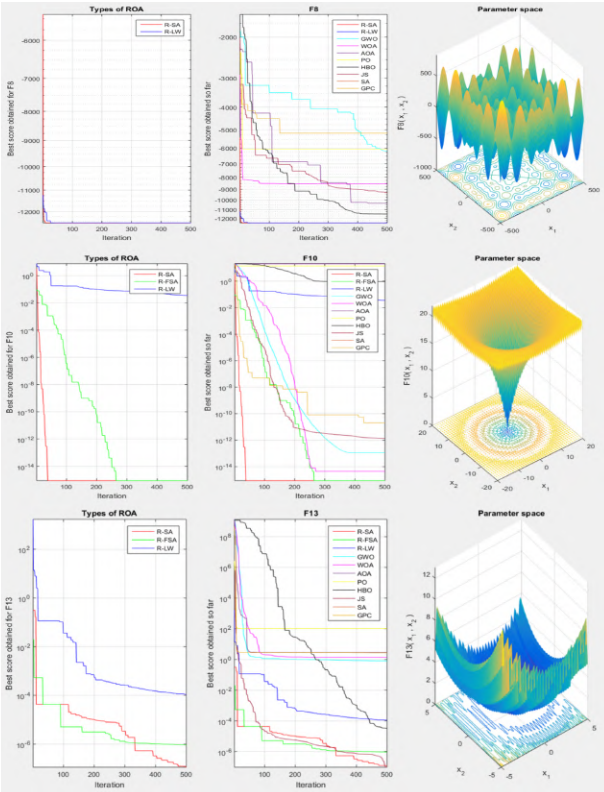

Figure 5. 3D and 2D representation of the multimodal benchmark functions

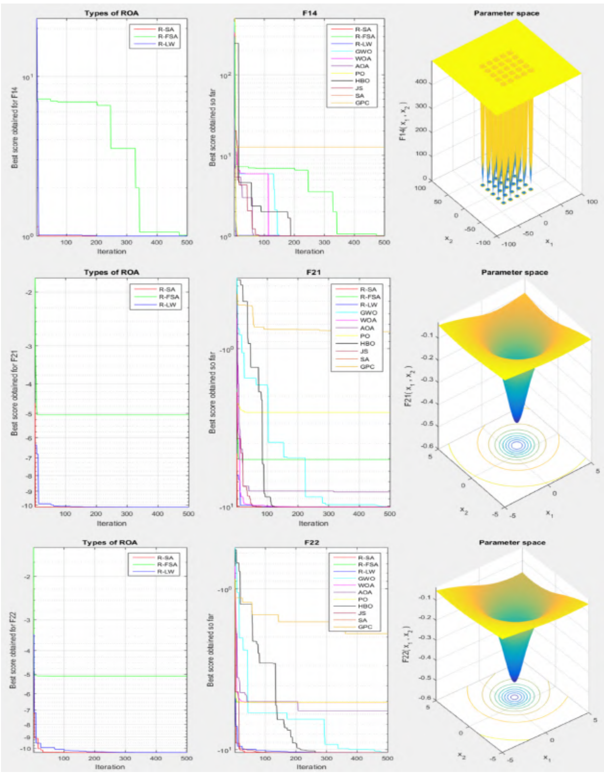

Figure 6. 2D and 3D representation of the fixed-dimension multimodal benchmark functions

5. EXPERIMENTS

The effectiveness of the proposed ROA algorithm is tested using two classical engineering problems then it applied on data classification by performing three experiments. The experiments were conducted regarding the data reduction. The first experiment proves the superiority of ROA against three other latest optimization algorithms AOA, GWO and WOA by applying data reduction for different 12 datasets obtained from KEEL repository [26]. In the second experiment a recent published paper [27], [28] were used in a comparison with our proposed algorithm. The last experiment was conducted on three student performance prediction data as a real-life application.

5.1. ROA for classical engineering problems

Pressure vessel designs and speed reducer designs are two restricted engineering design issues that are used in this part. Because these problems involve several equality and inequality constraints, the ROA code needs be updated (without changing the algorithm's mechanism) to deal with such constrained optimization problems more accurately. it can be seen that the different actions in this code are based on boolean algebra. in order to solve the above problems and to get the most efficient design for each case, it is vital to apply the ROA algorithm to be able to better deal with the inequality constraints.

5.1.1. Speed reducer design problem

Speed reducer design is a generalized geometric programming problem in which the main goal is to find the optimal design by minimizing the total weight of a speed reducer under constraints [29]. The speed reducer is a part of the gear box of mechanical system. Seven variables are involved in the design of the speed reducer. As presented in Figure 7, the speed reducer is considered with x_1 the width of the face, x_2 the teeth module, x_3 the teeth number on pinion, x_4, x_5 are the length of first and second shaft between bearings respectively, and x_6, x_7 are the first and second shaft diameters, respectively. Another schematic of the speed reducer is presented in Figure 8. This problem is associated with 11 constraints. These formulated as follows:

$$
\begin{aligned}
F(x) = 0.7854 * x_1 * x_2^2 * (3.3333 * x_3^2 + 14.9334 * x_3 - 43.0934) \\
- 1.508 * x_1 * (x_6^2 + x_7^2) + 7.4777 * (x_6^3 + x_7^3) \\
+ 0.7854 * (x_4 * x_6^2 + x_5 * x_7^2) \\
subjectto: \\
g(1) = 27/(x_1 * x_2^2 * x_3) - 1 \leq 0 \\
g(2) = 397.5/(x_1 * x_2^2 * x_3^2) - 1 \leq 0 \\
g(3) = (1.93 * x_4^3)/(x_2 * x_3 * x_6^4) - 1 \leq 0 \\
g(4) = (1.93 * x_5^3)/(x_2 * x_3 * x_7^4) - 1 \leq 0 \\
g(5) = ((sqrt(((745 * x_4)/(x_2 * x_3))^2 + 16.9e6))/(110 * x_6^3)) \\
- 1 \leq 0 \\
g(6) = ((sqrt(((745 * x_5)/(x_2 * x_3))^2 + 157.5 \exp 6))/(85 * x_7^3)) \\
- 1 \leq 0 \\
g(7) = ((x_2 * x_3)/40) - 1 \leq 0 \\
g(8) = (5 * x_2/x_1) - 1 \leq 0 \\
g(9) = (x_1/12 * x_2) - 1 \leq 0 \\
g(10) = ((1.5 * x_6 + 1.9)/x_4) - 1 \leq 0 \\
g(11) = ((1.1 * x_7 + 1.9)/x_5) - 1 \leq 0 \\
where \\
2.6 \leq x_1 \leq 3.6, 0.7 \leq x_2 \leq 0.8, 17 \leq x_3 \leq 28 \\
7.3 \leq x_4 \leq 8.3, 7.3 \leq x_5 \leq 8.3, 2.9 \leq x_6 \leq 3.9 \\
5.0 \leq x_7 \leq 5.5
\end{aligned}
$$

The statistical results and the best solution obtained by four algorithms water cycle algorithm (WCA) [30], GWO, JS, and AOA were compared with the proposed ROA-SA in Table 8 while ROA-FSA and ROA-LW solutions are NA. The JS is the worst while others performed well as shown. The results show that ROA performs well than the competitive algorithms. ROA presents a high performance in solving this kind of problems and shows a good rate in avoiding local optima and can quickly converge towards the optimum while satisfying all constraints.

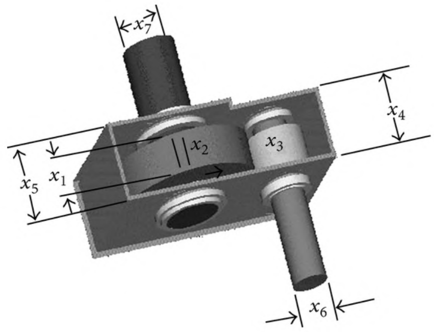

Figure 7. The speed reducer design

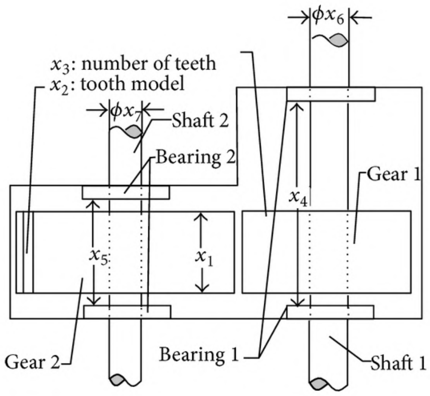

Figure 8. The speed reducer (schematic view)

Table 8. Comparison of solutions obtained for the speed reducer problem

Variables	ROA-SA	WCA	GWO	JS	AOA
x_1	3.50065	3.500000	3.50892	3.5764	3.4976
x_2	0.700031	0.700000	0.7	0.715613	0.7
x_3	17	17	17	18.4192	17
x_4	7.33066	7.300000	7.37782	7.31221	7.3
x_5	7.72448	7.715319	7.90736	8.29388	7.8
x_6	3.35038	3.350214	3.35432	3.78292	3.3501
x_7	5.28697	5.286654	5.28679	5.49218	5.2857
F(x)	2994.5882	2994.471066	3004.0097	1.5213e+12	3.00E+03
Mean	2994.6126	2994.474392	3.0085e+03	1.5213e+12	3.00E+03
SD	0.38847	7.4E-03	2.9074	1.9620e+12	1.22E-12

5.1.2. Pressure vessel design problem

This design problem aims to find the minimum whole cost including welding, forming, and the material of a cylindrical vessel as in Figure 9 [31].This problem has four constraints where thickness of the shell

(T_s), thickness of the head (T_h), inner radius (R), and length of the cylindrical section without considering the head (L). This problem and constraints are formulated as [31]:

$$F(x) = 0.6224 * x_1 * x_3 * x_4 + 1.7781 * x_2 * x_3^2 + 3.1661$$
$$x_1^2 * x_4 + 19.84 * x_1^2 * x_3;$$
$$subjectto:$$
$$g(1) = -x_1 + 0.0193 * x_3 \leq 0$$
$$g(2) = -x_2 + 0.00954 * x_3 \leq 0$$
$$g(3) - pi * x_3^2 * x_4 - (4/3) * pi * x_3^3 + 1296000 \leq 0$$
$$g(4) = x_4 - 240 \leq 0$$
$$where$$
$$0 \leq x_1 \leq 99, 0 \leq x_2 \leq 99$$
$$10 \leq x_3 \leq 200, 10 \leq x_4 \leq 200$$

The experimental results obtained for this problem are listed in Table 9 which clearly illustrates that the ROA has the ability to reach the optimal design that has the best optimum cost with the best mean and stranded deviation and minimum *F(x)*, while WCA comes in second place. In summary, ROA proves that it can performed highly in solving these challenging problems. ROA avoids local optima successfully and converges quickly to the optimum with high convergence rate satisfying all constraints.

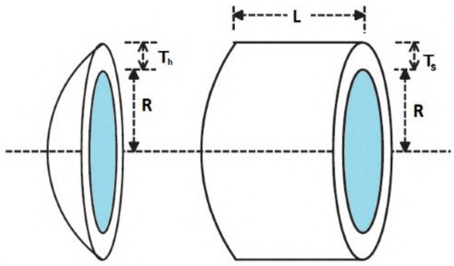

Figure 9. The pressure vessel (schematic view)

Table 9. Comparison of solutions obtained for the pressure vessel problem

Variables	ROA-SA	WCA	GWO	JS	AOA
x_1	0.7877859	0.7781	0.812500	14.04208	0.7900
x_2	0.3900009	0.3846	0.434500	10.58337	0.3899
x_3	40.81198	40.3196	42.089181	71.07832	41.0226
x_4	193.3671	-200.0000	176.758731	107.3518	190.4405
F(x)	5906.9382	5885.3327	6051.5639	506841.0511	5.90E+03
Mean	5909.4435	6230.4247	6038.01	5.7317e+05	6.52E+03
SD	9.0077	338.7300	249.6246	5.7024e+05	4.31E+02

5.2. ROA for improving data classification

5.2.1. ROA for data reduction

The operation on collecting data from different data-warehouses this causes difficulty in the analysis because of the huge amount of data. This is why data reducion is very important to decrease the volume of data and save memory resources and reduce time consumption without any effect of the result obtained from data mining that means the result obtained from data mining before data reduction and after data reduction is almost the same (or even better) which means that there will be no loss of information and the same quality of information can be obtained without data reduction. Analyzing a large amount of data without data reduction will result in very poor quality analysis duo to slow processing time and reduced menory availability, as a result the quality ana quantity of data that is obtained will be limited. Data reduction will also improve the quality of data analysis since only the necessary data will be analyzed, resulting in improved analysis with fewer errors.

The ROA algorithm is applied for data reduction on different (balanced and imbalanced) real-world datasets. The application of ROA for data reduction issue performed by searching for the optimal subset of examples in the training set using the capability of the ROA algorithm to optimally reduce the data. The algorithm starts the search with a search variables generated randomly. The binary encoding type is used in the proposed algorithm as its representation scheme to select the optimal reduced training set. In binary encoding, each search variable is represented by a binary vector, and the examples of training data are considered as either existed "1" or not existed "0". The ones represent the remained examples while zeros represent the removed examples. The search variables in ROA are evaluated by the accuracy and F-measure, as fitness value.

Figure 10 presents the ROA flowchart that shows its implication in the problem of the data reduction for the real-world datasets. As the figure shows, the original used data set is divided into three subsets: testing, training and validation. Our proposal focuses on finding the best abbreviated subset from the training set first and then, is tested via the validation set. If one of the criteria for termination is met, the best search variable is considered as converged to the global optimum and this search variable which has the best value of F-measure is decoded and recorded as the solution that consists the reduced training set.

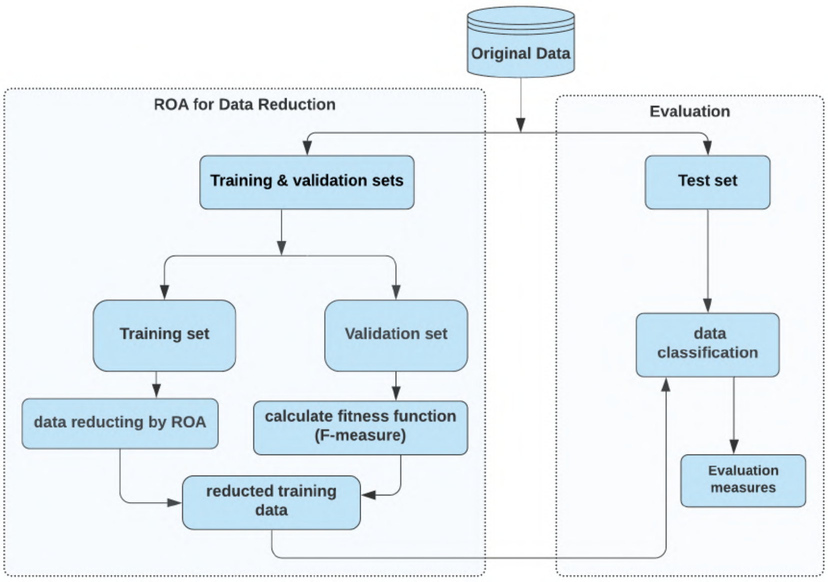

Figure 10. Flowchart for ROA in data reduction

The first experiment on data reduction is conducted by comparing our algorithm ROA with other latest optimization algorithms AOA, WOA and GWO. The compared algorithms are applied on 12 datasets which their statistics are listed in Table 10 and evaluate the performance with both accuracy and F-measure, the results listed in Table 11. This table generally views the performance of each algorithm in term of accuracy and F-measure measures. Additionally, the original dataset results before reduction are provided for references.

To get the best fair comparison, the non-parametric statistical hypothesis Wilcoxon signed-rank test [32] was used to analyze the results statistically and draw fairly effective conclusions. All used algorithms were compared with ROA for each dataset. For each two compared optimization algorithms,the differences were ranked in ascending order from 1 to 12 after their calculation and the negative differences of the ranks were assigned a sign. After summing up all the positive and negative ranks separately, they assigned as *R+* and *R-* respectively. A significance level *alpha*=0.1 was used in the comparison of the *T* value, with 17 used as a critical value for 12 datasets where *T* equals *minR+,R-*. The null hypothesis which was only rejected if the critical value *T* ¡ or = 17 was assumed to be that all differences in performance between any couple of compared algorithms may occur by chance.

Table 10. Statistics of 12 experimental datasets

Datasets	Features	Instances	Positive	Negative
breastEW	30	569	212	357
glass5	9	214	70	144
pima	8	768	268	500
Tic-tac-toe	9	958	332	626
Transfusion	4	748	174	574
vehicle1	18	846	217	629
Vowel(3-others)	10	528	48	480
Wine(1-others)	13	178	59	119
Wine (2-to other)	13	178	71	107
Wisconsin	9	683	239	444
yeast3	8	1484	163	1321
yeast6	8	1484	35	1449

Table 11. Performance of original data, ROA, AOA, GWO and WOA of the 12 datasets

Datasets	Matrices	Original	ROA_SA	ROA_FSA	ROA_LW	AOA	GWO	WOA
breastEW	Acc	96.48	96.48	95.77	**97.89**	97.18	94.37	95.07
	F-measure	95.41	95.41	94.55	**97.2**	96.23	92.86	93.58
glass5	Acc	96.23	**98.11**	**98.11**	**98.11**	**98.11**	96.23	**98.11**
	F-measure	50	**66.67**	**66.67**	**66.67**	**66.67**	NaN	**66.67**
pima	Acc	69.27	**70.31**	68.75	69.27	69.27	**70.31**	67.71
	F-measure	46.85	**52.89**	45.45	47.37	46.96	51.28	52.24
Tic-tac-toe	Acc	**98.74**	**98.74**	**98.74**	**98.74**	98.33	**98.74**	98.33
	F-measure	**98.16**	**98.16**	**98.16**	**98.16**	97.53	**98.16**	97.53
Transfusion	Acc	**75.4**	**75.4**	**75.4**	74.33	74.33	73.26	**75.4**
	F-measure	**23.33**	**23.33**	**23.33**	22.58	22.58	19.35	**23.33**
vehicle1	Acc	76.3	**76.78**	76.3	75.83	76.3	76.3	74.88
	F-measure	35.9	**39.51**	35.9	35.44	35.9	35.9	27.4
Vowel(3-others)	Acc	**99.24**	98.48	**99.24**	**99.24**	98.48	**99.24**	97.73
	F-measure	**95.65**	90.91	**95.65**	**95.65**	90.91	**95.65**	85.71
Wine(1-others)	Acc	84.09	**93.18**	81.82	84.09	84.09	86.36	79.55
	F-measure	69.57	**90.32**	63.64	69.57	69.57	75	57.14
Wine(2-to other)	Acc	**81.82**	**81.82**	**81.82**	**81.82**	77.27	70.45	**81.82**
	F-measure	**73.33**	**73.33**	**73.33**	**73.33**	66.67	58.06	**73.33**
wisconsin	Acc	94.71	**95.88**	95.29	94.71	95.29	**95.88**	94.71
	F-measure	92.44	**94.21**	93.33	92.44	93.33	**94.21**	92.44
yeast3	Acc	94.34	94.07	**94.88**	94.34	94.07	93.53	93.26
	F-measure	69.57	68.57	**72.46**	70.42	68.57	65.71	63.77
yeast6	Acc	98.11	98.11	98.11	**98.38**	97.84	98.11	98.11
	F-measure	36.36	36.36	46.15	**50**	20	36.36	36.36

Comparison of accuracy: Table 12 shows the average accuracy as a result of the significance test for ROA_SA vs. WOA, GWO and AOA in case of the support vector machines (SVM) with and radial basis function (RBF) kernel. For AOA and GWO with the SVM classifier, ROA_SA is better - +ve difference- for 11 datasets, while AOA and GWO is better (-ve difference) than ROA_SA in only one dataset. While in case of ROA_SA vs. WOA our proposal is better in all datasets. After calculating R+ = 67 (all +ve ranks summation) and R- = 8 (all -ve ranks summation) in ROA_SA vs. AOA, R+ = 65 (all +ve ranks) and R- = 7 in ROA_SA vs. GWO, and R+ = 72 (all +ve ranks summation) and R- = 0 (all -ve ranks summation) in ROA_SA vs. WOA. It can be concluded that ROA_SA can outperform AOA statistically as $T = min\{R+, R\} = min\{67, 8\} = 8 < 17$. ROA_SA can outperform the GWO statistically as $T = min\{65, 7\} = 7 < 17$. ROA_SA can outperform the WOA statistically as $T = min\{72, 0\} = 0 < 17$.

The same procedures are followed to perform the test for ROA_FSA and ROA_LW and the results are as follows: the average accuracy as a result of the significance test for ROA_FSA vs. WOA, GWO and AOA in case of SVM with rbf kernel. For AOA and GWO with the SVM classifier, ROA_FSA is better (positive difference) for 9 datasets, while AOA and GWO is better - -ve difference- than ROA_FSA for 3 datasets. In case of ROA_FSA vs. WOA our proposal is better in all datasets. After calculating R+ = 48 (all +ve ranks summation) and R- = 27 (all -ve ranks summation) in ROA_FSA vs. AOA, R+ = 48 and R- = 24 in ROA_FSA vs. GWO, and R+ = 72 and R- = 0 in ROA_FSA vs. WOA.As 12 datasets were used, the T value at the level of 0.1 should be¡ or = 17 to reject our null hypothesis, returning to the critical table. It can be concluded that as $T = min\{72, 0\} = 0 < 17$.

Regarding the average accuracy as a result of the significance test for ROA LW vs. WOA ,GWO and AOA in case of SVM with rbf kernel. For AOA with the SVM classifier, ROA LW is better, +ve difference, for 10 datasets, while AOA is better, -ve difference, than ROA LW in 2 datasets, whereas in case of GWO with SVM, ROA LW is better, +ve difference, for 8 datasets, while GWO is better, -ve difference, than ROA LW in 4 datasets, Whereas in case of ROA LW vs. WOA our proposal is better in 11 datasets, while WOA is better, -ve difference, than ROA LW for only one dataset. After calculating R+ = 56 (all +ve ranks summation) and R- = 16 (all -ve ranks summation) in ROA_LW vs. AOA, R+ = 49 and R- = 28 in ROA_LW vs. GWO, and R+ = 68 and R- = 7 in ROA_LW vs. WOA. It can be concluded that ROA_LW cannot outperform GWO statistically but, ROA_LW can outperform the AOA statistically as $T = min\{56, 16\} = 16 < 17$, ROA_LW can statistically outperform the WOA as $T = min\{68, 7\} = 7 < 17$.

Table 12. Results of average Accuracy for ROA_SA vs. WOA, GWO and AOA

Dataset	ROA_SA	AOA	Difference	Rank	GWO	Difference	Rank	WOA	Difference	Rank
breastEW	96.48	97.18	-0.7	-8	94.37	2.11	9	95.07	1.41	9
glass5	98.11	98.11	0	1	96.23	1.88	8	98.11	0	1
pima	70.31	69.27	1.04	9	70.31	0	1	67.71	2.6	11
Tic-tac-toe	98.74	98.33	0.41	5	98.74	0	1	98.33	0.41	5
Transfusion	75.4	74.33	1.07	10	73.26	2.14	10	75.4	0	1
vehicle1	76.78	76.3	0.48	6	76.3	0.48	5	74.88	1.9	10
Vowel(3-others)	98.48	98.48	0	1	99.24	-0.76	-7	97.73	0.75	6
Wine(1-others)	93.18	84.09	9.09	12	86.36	6.82	11	79.55	13.63	12
Wine (2-to other)	81.82	77.27	4.55	11	70.45	11.37	12	81.82	0	1
Wisconsin	95.88	95.29	0.59	7	95.88	0	1	94.71	1.17	8
yeast3	94.07	94.07	0	1	93.53	0.54	6	93.26	0.81	7
yeast6	98.11	97.84	0.27	4	98.11	0	1	98.11	0	1
		$T = min\{67, 8\} = 8$				$T = min\{65, 7\} = 7$			$T = min\{72, 0\} = 0$	

Comparison of f-measure: Table 13 shows the average f-measure as a result of the significance test for ROA SA vs. WOA, GWO, and AOA in the case of SVM with an RBF kernel. For AOA and GWO with the SVM classifier, ROA SA is better, +ve difference, for 11 datasets, while AOA and GWO is a better, -ve difference, than ROA SA for only one dataset. While in the case of ROA SA vs. WOA our proposal is better in all datasets. After calculating R+ = 69 (all +ve ranks summation) and R- = 6 (all -ve ranks summation) in ROA_SA vs. AOA, R+ = 54 and R- = 9 in ROA_SA vs. GWO, and R+ = 72 and R- = 0 in ROA_SA vs. WOA. It can be concluded that ROA_SA can outperform AOA statistically as $T = min\{R+, R\} = min\{69, 6\} = 6 < 17$. ROA_SA can statistically outperform the GWO as $T = min\{54, 9\} = 9 < 17$. ROA_SA can statistically outperform the WOA as $T = min\{72, 0\} = 0 < 17$.

Table 13. Results of average F-measure for ROA_SA vs. WOA, GWO and AOA

Dataset	ROA_SA	AOA	Difference	Rank	GWO	Difference	Rank	WOA	Difference	Rank
breastEW	95.41	96.23	-0.82	-6	92.86	2.55	5	93.58	1.83	8
glass5	66.67	66.67	0	1	NaN	67.67	0	66.67	0	1
pima	52.89	46.96	5.93	9	51.28	1.61	4	52.24	0.65	6
Tic-tac-toe	98.16	97.53	0.63	4	98.16	0	1	97.53	0.63	5
Transfusion	23.33	22.58	0.75	5	19.35	3.98	8	23.33	0	1
vehicle1	39.51	35.9	3.61	8	35.9	3.61	7	27.4	12.11	11
Vowel(3-others)	90.91	90.91	0	1	95.65	-4.74	-9	85.71	5.2	10
Wine(1-others)	90.32	69.57	20.75	12	75	15.32	11	57.14	33.18	12
Wine (2-to other)	73.33	66.67	6.66	10	58.06	15.27	10	73.33	0	1
Wisconsin	94.21	93.33	0.88	7	94.21	0	1	92.44	1.77	7
yeast3	68.57	68.57	0	1	65.71	2.86	6	63.77	4.8	9
yeast6	36.36	20	16.36	11	36.36	0	1	36.36	0	1
		$T = min\{69, 6\} = 6$				$T = min\{54, 9\} = 9$			$T = min\{72, 0\} = 0$	

Same way the average f-measure as a result of the significance test for ROA FSA vs. WOA, GWO, and AOA in case of SVM with RBF kernel. For AOA and GWO with the SVM classifier, ROA FSA is better, +ve difference, for 9 datasets, AOA and GWO is a better, -ve difference, than ROA FSA for 3 datasets. In the case of ROA FSA vs. WOA, ROA FSA is better, +ve difference, for 11 datasets, while WOA is better, -ve difference, than ROA FSA for one dataset. In case of ROA_FSA vs. WOA , ROA_FSA is better - +ve difference- for 11 datasets, while WOA is better - -ve difference- than ROA_FSA for one dataset. After calculating R+ = 52 (all +ve ranks summation) and R- = 23 (all -ve ranks summation) in ROA_FSA vs. AOA, R+ = 42 and R- = 21 in ROA_FSA vs. GWO, and R+ = 67 and R- = 8 in ROA_FSA vs. WOA. It can be concluded that ROA_FSA can outperform WOA statistically as $T = min\{R+, R\} = min\{67, 8\} = 8 < 17$. ROA_FSA cannot statistically outperform in case of AOA and GWO.

Same way the average f-measure as a result of the significance test for ROA LW vs. WOA, GWO and AOA In case of SVM with RBF kernel. For AOA and WOA with the SVM classifier, ROA LW is better, +ve difference, for 10 datasets, while AOA and WOA is better, -ve difference, than ROA LW for 2 datasets. While in case of ROA LW vs. GWO, ROA LW is better, +ve difference, for 8 datasets, while GWO is better, -ve difference, than ROA LW for 4 datasets. After calculating R+ = 63 (all +ve ranks summation) and R- = 12 (all -ve ranks summation) in ROA_LW vs. AOA, R+ = 44 and R- = 22 in ROA_LW vs. GWO, and R+ = 63 and R- = 12 in ROA_LW vs. WOA. It can be concluded that ROA_LW can outperform AOA statistically as $T = min\{R+, R\} = min\{63, 12\} = 12 < 17$. ROA_LW can statistically outperform the WOA as $T = min\{63, 12\} = 12 < 17$. ROA_LW cannot statistically outperform in case of the GWO.

5.2.2. ROA for class imbalance problem

Class imbalance problem cases is a difficulty in classification learning algorithms in pattern. The article [27] proposed a new oversampling technique to increase the efficiency of the learning classification algorithms based on a fuzzy representativeness difference-based oversampling technique, using affinity propagation and the chromosome theory of inheritance (FRDOAC). We used our meta-heuristic optimization technique ROA for data reduction on 16 imbalanced datasets which were used in [27] as a comparison with recent published research in the second experiment. For more details for the 16 benchmark datasets and for performance evaluation matrices see [27].

We use F-measure (F_M), G-mean (G_M) and area under curve (AUC) to evaluate the performance of the proposal to be fair since this is the metrics that have been used in [27] and more details about these measures can be found in the same reference because the experiment is on imbalanced datasets and the metrics accuracy is not fair enough. Random forest (RF) classifier used for this comparation with cross validation for 5-fold like the previous paper, 5 run times and take the averages for the results of (F-measure, G-mean, and AUC). The results of the performance shown in Table 14.

The results showed that after applied our meta-heuristic optimization algorithm on the 16 benchmark imbalanced datasets the performance of evaluation matrices improved compared with the FRDOAC method which mention in [27] article in F-measure and G-mean matrices and for AUC metric our proposal not better than FRDOAC method but it is still better than original data. Figure 11 represents box plot that is summarizing the set of data and it gives a clear and qualitative description of the performance of the used classifier. From this figure, the box plot of FRDOAC is visibly lower than ROA types in both F-measure and G-mean so that they more directly demonstrate the advantages of ROA algorithm. We got the better results on imbalance data without increasing the number of instances in datasets with any pre-processing oversampling techniques which produced the results of reducing the complexity and processing time, so our proposal is outperformed on oversampling algorithm that mentioned in the compared article [27] for dealing with imbalanced data problem.

5.2.3. ROA in real life problem

For more validation for our proposal, last comparison has been conducted as an application on real life problem; where three popular students performance prediction datasets used to compare between ROA and latest optimization algorithms SA, AOA, GWO, and WOA with respect to data reduction. The first dataset contains 649 instances and 33 attributes, this dataset which addresses the students performance is collected from two secondary schools of Portuguese (Mousinho da Silveira (MS) and Gabriel Pereira (GP)). The dataset includes student's attributes like academic grades, demographic attributes, social attributes, and school-related attributes. School reports and questionnaires are used for collecting data from the student [33]. The second dataset was collected from three different colleges, Doomdooma College, Duliajan College, and Digboi College of Assam, India are those three colleges. Initially, this data composed of twenty two attributes [34].

The third dataset was collected in Dibrugarh University from the common entrance conducted by it in 2013. This data has 12 attributes for students who came for counseling cum admission into Assam medical colleges [35]. The three datasets suffer from the imbalance problem due to the decline in repetition rate among students in the compilation places of the database. the descriptions of the used datasets are listed in Table 15 each dataset split into three parts in which training validation and testing datasets (ratio of 50%: 25%: 25%). We employ ROA and other optimization algorithms (SA, AOA, GWO, and WOA) to choose the optimal subset for training data and extract utilizable information for student performance to help in understanding university students' performance and identifying factors that affect it to help decision-makers. Representation for the comparison of the results for SVM classifier with data reduction is shown in Table 16 and Figures 12 to 14 respectively. From the previous results we can conclude that ROA is outperformed in both accuracy and F-measures in the three datasets except for CEE-dataset where SA has the best F-measure value.

Table 14. Averages of performance by different methods on 16 datasets using RF classifier

Datasets	Metrics	NONE	ROA_SA	ROA_FSA	ROA_LW	FRDOAC
Sonar	F-M	0.681	0.706	0.766	**0.793**	0.732
	G-m	0.708	0.712	0.783	**0.789**	0.742
	AUC	0.735	0.712	0.786	**0.790**	0.78
Bupa	F-M	0.618	**0.727**	0.701	**0.727**	0.658
	G-m	0.676	**0.705**	0.688	**0.705**	0.693
	AUC	0.704	0.717	0.696	0.717	**0.734**
Pima	F-M	0.625	**0.723**	**0.723**	**0.723**	0.666
	G-m	0.703	0.705	0.705	0.705	**0.742**
	AUC	0.727	0.716	0.716	0.716	**0.764**
Glass Identification (GlaId)	F-M	0.749	**0.815**	**0.815**	**0.815**	0.782
	G-m	0.805	0.812	0.812	0.812	**0.852**
	AUC	0.819	0.815	0.815	0.815	**0.862**
Verterbral(Verb)	F-M	0.726	**0.828**	0.799	0.820	0.774
	G-m	0.793	0.820	0.798	0.812	**0.841**
	AUC	0.805	0.824	0.800	0.816	**0.855**
Haberman (Hab)	F-M	0.377	0.646	0.640	**0.690**	0.458
	G-m	0.512	0.597	0.577	**0.664**	0.618
	AUC	0.592	0.636	0.627	**0.685**	0.665
Glass	F-M	0.886	**0.934**	0.915	**0.934**	0.888
	G-m	0.927	0.929	0.922	0.929	**0.938**
	AUC	0.93	0.930	0.923	0.930	**0.941**
Yeast	F-M	0.547	0.867	**0.869**	0.838	0.61
	G-m	0.649	0.794	**0.832**	0.776	0.747
	AUC	0.703	0.815	**0.845**	0.800	0.764
Glass6 (Gla6)	F-M	0.876	**0.958**	**0.958**	**0.958**	0.893
	G-m	0.906	**0.948**	**0.948**	**0.948**	0.917
	AUC	0.911	**0.949**	**0.949**	**0.949**	0.922
Ecoil0675(Ec75)	F-M	0.79	**0.923**	**0.923**	**0.923**	0.732
	G-m	0.829	**0.891**	**0.891**	**0.891**	0.789
	AUC	0.848	**0.897**	**0.897**	**0.897**	0.871
Movement (Move)	F-M	0.355	**0.905**	**0.905**	**0.905**	0.863
	G-m	0.451	**0.839**	**0.839**	**0.839**	0.828
	AUC	0.618	**0.852**	**0.852**	**0.852**	**0.852**
Ecoil4 (Ec4)	F-M	0.704	0.862	0.870	**0.905**	0.84
	G-m	0.743	0.832	0.745	0.855	**0.879**
	AUC	0.784	0.844	0.778	0.865	**0.897**
Glass5 (Gla5)	F-M	0.584	**0.794**	NaN	0.733	0.739
	G-m	0.627	0.577	0.000	0.575	**0.788**
	AUC	0.772	0.667	0.500	0.663	**0.877**
kddcup-land vs portsweep(Klvp)	F-M	0.98	**1**	**1**	**1**	**1**
	G-m	0.909	**1**	**1**	**1**	0.997
	AUC	0.99	**1**	**1**	**1**	0.99
Poker-8vs6(Po86)	F-M	0	NaN	NaN	NaN	**0.841**
	G-m	0	0.000	0.000	0.000	**0.185**
	AUC	0.5	0.500	0.500	0.490	**0.567**
Abalone19(Ab19)	F-M	0	**0.120**	NaN	**0.120**	0.02
	G-m	0	**0.350**	0.000	**0.350**	0.045
	AUC	0.5	**0.560**	0.500	**0.560**	0.508

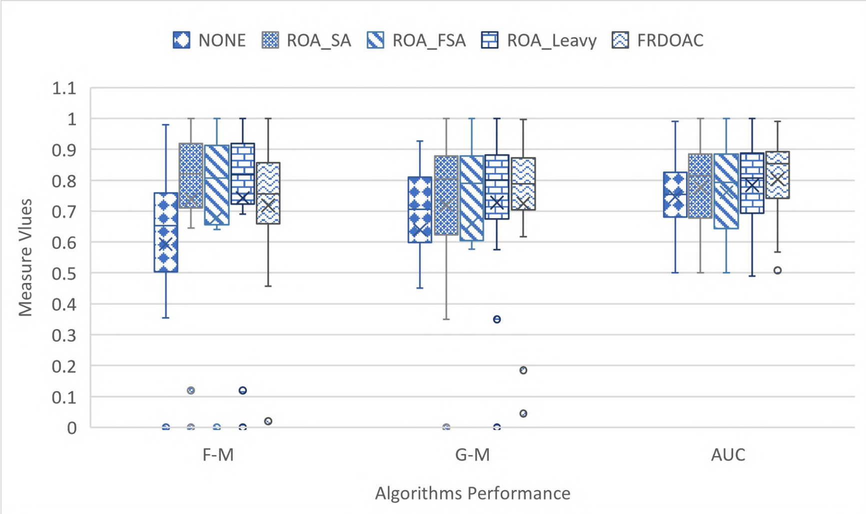

Figure 11. Overall performances of all methods obtained by RF classifier

Table 15. Information of student performance datasets

Dataset	Instances	No. of attributes	No. of classes	
			Pass	Fail
Portuguese course (student-port)	649	33	452	197
Sapfile	300	22	224	76
CEE	666	12	509	157

Table 16. Comparison for CEE-dataset, Sapfile, and Student-port datasets

Technique	CEE-dataset		Sapfile-dataset		Student-port-dataset	
	ACC	F-measure	ACC	F-measure	ACC	F-measure
Original	66.27	60	73.33	84.62	94.44	96.7
ROA_SA	64.46	58.74	**76**	**86.15**	**96.91**	**98.2**
ROA_FSA	**66.87**	60.43	73.33	84.62	89.51	94.0
ROA_LW	66.27	60	73.33	84.62	92.59	95.7
SA	64.66	**71.52**	75.56	85.79	89.09	93.85
AOA	66.27	60	73.33	84.62	89.51	94.12
GWO	64.46	58.74	73.33	84.62	94.44	96.8
WOA	65.66	58.99	74.67	85.5	94.44	96.7

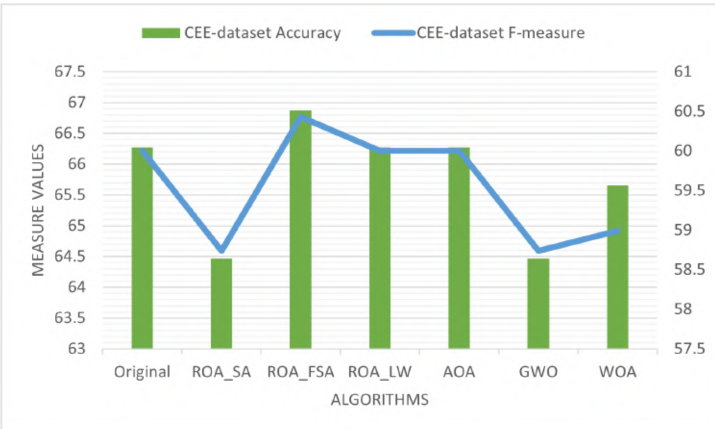

Figure 12. Comparison for CEE-dataset

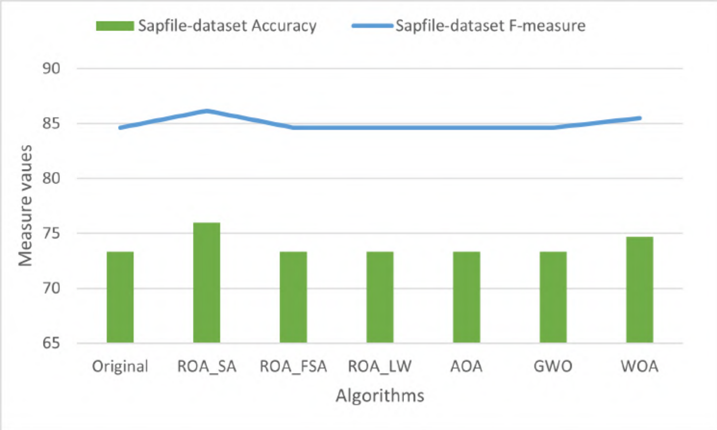

Figure 13. Comparison for Sapfile-dataset

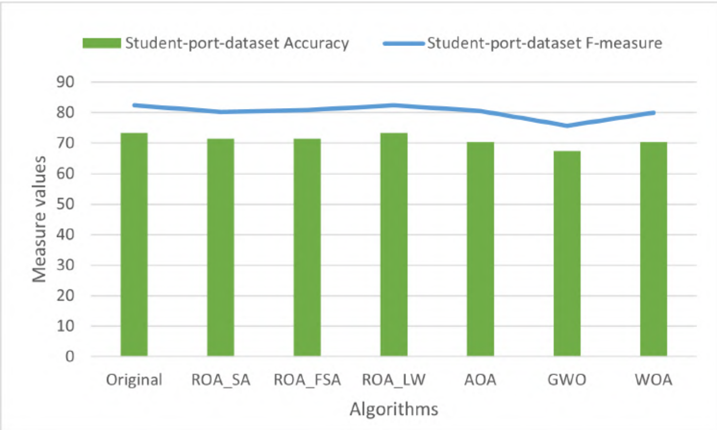

Figure 14. Comparison for student-port-dataset

6. CONCLUSION

The main common considerations regarding the performance and design of any optimization algorithm are the simplicity, the flexibility, and the robustness. Dealing these features in appropriate manner, makes this optimization algorithm widely acceptable in the researchers' community. From this perspective, metaheuristic algorithms that have been introduced recently, especially the swarm-based inspired algorithms, have produced very interesting results. This paper proposed a novel swarm-based optimization algorithm inspired by the simulation of R. octopus social behavior. The proposed method (named as ROA, Rhizostoma optimization algorithm) with only three operators, to simulate the food search, forming a swarm, and searching for food in the swarm, to keep the simplicity, and to effectively ensure a good flexibility, three kinds of movement strategies (SA, FSA, and LW) were proposed and modelled mathematically. ROA not only has the simplicity and the flexibility, as it holds a few parameters for control and three different movement strategies, but it also has the robustness. The proposed ROA can generate an objective function with minimum error values to solve optimization problems and also maintains the ability to avoid the trap of sub optimal solutions. A comprehensive experiments was performed on 23 mathematical benchmark functions to analyze the balance between exploration/exploitation processes, the avoidance of local optima rate, and the speed of convergence rate of the proposed ROA. ROA proved it is enough competitive with other latest meta-heuristic methods when we apply it on engineering problems and comparing its results with the results of recently published paper dealing with imbalanced datasets, also after applying ROA for real-world problems, the significance of Wilcoxon signed-rank test. Also proved the ROA ability to reach the best optimal solutions compared with other latest optimizers.

ACKNOWLEDGEMENT

We would like to thank the BANU for their assistance with the project number BAP-22-1004-003.

REFERENCES

[1] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.
[2] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.
[3] J.-S. Chou and D.-N. Truong, "A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean," *Applied Mathematics and Computation*, vol. 389, Jan. 2021, doi: 10.1016/j.amc.2020.125535.
[4] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80–98, May 2015, doi: 10.1016/j.advengsoft.2015.01.010.
[5] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. Al-Atabany, "Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems," *Applied Intelligence*, vol. 51, no. 3, pp. 1531–1551, Mar. 2021, doi: 10.1007/s10489-020-01893-z.
[6] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft Computing*, vol. 22, no. 2, pp. 387–408, Jan. 2018, doi: 10.1007/s00500-016-2474-6.
[7] K. A. ElDahshan, A. A. AlHabshy, and B. I. Hameed, "Meta-heuristic optimization algorithm-based hierarchical intrusion detection system," *Computers*, vol. 11, no. 12, Nov. 2022, doi: 10.3390/computers11120170.
[8] M. Khishe and M. R. Mosavi, "Chimp optimization algorithm," *Expert Systems with Applications*, vol. 149, Jul. 2020, doi: 10.1016/j.eswa.2020.113338.
[9] M.-T. Vakil-Baghmisheh and A. Navarbaf, "A modified very fast simulated annealing algorithm," in *2008 International Symposium on Telecommunications*, Aug. 2008, pp. 61–66, doi: 10.1109/ISTEL.2008.4651272.
[10] O. K. Erol and I. Eksin, "A new optimization method: big bang–big crunch," *Advances in Engineering Software*, vol. 37, no. 2, pp. 106–111, Feb. 2006, doi: 10.1016/j.advengsoft.2005.04.005.
[11] W. Zhao, L. Wang, and Z. Zhang, "Atom search optimization and its application to solve a hydrogeologic parameter estimation problem," *Knowledge-Based Systems*, vol. 163, pp. 283–304, Jan. 2019, doi: 10.1016/j.knosys.2018.08.030.
[12] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *International Journal of Bio-Inspired Computation*, vol. 2, no. 2, pp. 78–84, 2010.
[13] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: theory and application," *Advances in Engineering Software*, vol. 105, pp. 30–47, Mar. 2017, doi: 10.1016/j.advengsoft.2017.01.004.
[14] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Studies in Computational Intelligence*, vol. 284, 2010, pp. 65–74.
[15] S. Harifi, J. Mohammadzadeh, M. Khalilian, and S. Ebrahimnejad, "Giza pyramids construction: an ancient-inspired metaheuristic algorithm for optimization," *Evolutionary Intelligence*, vol. 14, no. 4, pp. 1743–1761, Dec. 2021, doi: 10.1007/s12065-020-00451-3.
[16] Q. Askari, M. Saeed, and I. Younas, "Heap-based optimizer inspired by corporate rank hierarchy for global optimization," *Expert Systems with Applications*, vol. 161, Dec. 2020, doi: 10.1016/j.eswa.2020.113702.
[17] Q. Askari, I. Younas, and M. Saeed, "Political optimizer: a novel socio-inspired meta-heuristic for global optimization," *Knowledge-Based Systems*, vol. 195, May 2020, doi: 10.1016/j.knosys.2020.105709.
[18] E. Alba and B. Dorronsoro, "The exploration/exploitation tradeoff in dynamic cellular genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 2, pp. 126–142, Apr. 2005, doi: 10.1109/TEVC.2005.843751.
[19] L. Lin and M. Gen, "Auto-tuning strategy for evolutionary algorithms: balancing between exploration and exploitation," *Soft Computing*, vol. 13, no. 2, pp. 157–168, Jan. 2009, doi: 10.1007/s00500-008-0303-2.
[20] F. Aznar, M. Pujol, and R. Rizo, "A swarm behaviour for Jellyfish bloom detection," *Ocean Engineering*, vol. 134, pp. 24–34, Apr. 2017, doi: 10.1016/j.oceaneng.2017.02.009.
[21] A. M. Reynolds, "Signatures of active and passive optimized Lévy searching in jellyfish," *Journal of The Royal Society Interface*, vol. 11, no. 99, Oct. 2014, doi: 10.1098/rsif.2014.0665.
[22] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983, doi: 10.1126/science.220.4598.671.
[23] G. L. Mariottini and L. Pane, "Mediterranean Jellyfish venoms: a review on scyphomedusae," *Marine Drugs*, vol. 8, no. 4, pp. 1122–1152, Apr. 2010, doi: 10.3390/md8041122.
[24] S. Fossette *et al.*, "Current-oriented swimming by jellyfish and its role in bloom maintenance," *Current Biology*, vol. 25, no. 3, pp. 342–347, Feb. 2015, doi: 10.1016/j.cub.2014.11.050.
[25] J. G. Digalakis and K. G. Margaritis, "On benchmarking functions for genetic algorithms," *International Journal of Computer Mathematics*, vol. 77, no. 4, pp. 481–506, Jan. 2001, doi: 10.1080/00207160108805080.
[26] J. Alcalá-Fdez *et al.*, "KEEL data-mining software tool: data set repository, integration of algorithms and experimen-

tal analysis framework," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, no. 2, pp. 255–287.
[27] R. Ren, Y. Yang, and L. Sun, "Oversampling technique based on fuzzy representativeness difference for classifying imbalanced data," *Applied Intelligence*, vol. 50, no. 8, pp. 2465–2487, Aug. 2020, doi: 10.1007/s10489-020-01644-0.
[28] R. P. Espíndola and N. F. F. Ebecken, "On extending F-measure and G-mean metrics to multi-class problems," *Data Mining VI*, vol. 1, pp. 25–34, 2005, doi: 10.2495/data050031.
[29] E. Mezura-Montes and C. A. C. Coello, "Useful infeasible solutions in engineering optimization with evolutionary algorithms," in *Mexican International Conference on Artificial Intelligence*, 2005, pp. 652–662, doi: 10.1007/11579427_66.
[30] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, "Water cycle algorithm-a novel metaheuristic optimization method for solving constrained engineering optimization problems," *Computers and Structures*, vol. 110–111, pp. 151–166, Nov. 2012, doi: 10.1016/j.compstruc.2012.07.010.
[31] B. K. Kannan and S. N. Kramer, "An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design," in *19th Design Automation Conference: Volume 2 — Design Optimization; Geometric Modeling and Tolerance Analysis; Mechanism Synthesis and Analysis; Decomposition and Design Optimization*, Sep. 1993, pp. 103–112, doi: 10.1115/DETC1993-0382.
[32] D. J. Sheskin, *Handbook of parametric and nonparametric statistical procedures*. Chapman and Hall/CRC, 2020.
[33] Y. K. Salal, S. M. Abdullaev, and M. Kumar, "Educational data mining: Student performance prediction in academic," *International Journal of Engineering and Advanced Technology*, vol. 8, no. 4C, pp. 54–59, 2019.
[34] S. Hussain, N. Abdulaziz Dahan, F. M. Ba-Alwi, and N. Ribata, "Educational data mining and analysis of students' academic performance using WEKA," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 9, no. 2, pp. 447–459, Feb. 2018, doi: 10.11591/ijeecs.v9.i2.pp447-459.
[35] S. Hussain, R. Atallah, A. Kamsin, and J. Hazarika, *Classification, clustering and association rule mining in educational datasets using data mining tools: a case study*. Springer International Publishing AG, 2019.

BIOGRAPHIES OF AUTHORS

Esraa A. Mahareek received the B.Sc. degree in computer science, in 2012, the M.Sc. degree in computer science, in 2021. She is currently an Teaching assistant in computer science at the Mathematics Department, Faculty of Science, Al-Azhar University, Cairo, Egypt. She has published research paper in the field of AI, machine learning, metaheuristic optimization. She can be contacted at email: esraa.mahareek@azhar.edu.eg.

Mehmet Akif Cifci (Member, IEEE) received the B.A. degree from the Department of ELT, Dokuz Eylul University, the B.C.S. degree from the Department of Computer Engineering, UoP University, and the M.C.S. and Ph.D. degrees in Computer Science from Istanbul Aydin University. He was an educator with over ten-year experience and an active researcher. He has published several research articles, presented at conferences and has provided training sessions on image processing, NLP, food safety, and data analysis. His research interests include AI, ML, and DL. He is a member of CMC and a Founder Member of ABV. He can be contacted at email: meh.cifci@kvk.lt.

Habiba A. El-Zohni received the M.Sc. and Ph.D degree in mathimatics, at the Mathematics Department, Faculty of Science, Al-Azhar University, Cairo, Egypt. She has published a number of research papers in the field of algebra. She is the supervisor of some master's and doctoral thesis. She is a reviewer in many Scopus-indexed journals. She can be contacted at email: elzohny_7@azhar.edu.eg.

Abeer S. Desuky received the B.Sc. degree in science, in 2003, the M.Sc. degree in computer science, in 2008, and the Ph.D. degree in computer science, in 2012. She was an Associate Professor, in 2017. She is currently an Associate Professor in computer science at the Mathematics Department, Faculty of Science, Al-Azhar University, Cairo, Egypt. She has published a number of research papers in the field of AI, machine learning, metaheuristic optimization, and data mining and analysis. She is the supervisor of some master's and doctoral thesis. She is a reviewer in many Scopus-indexed journals such as IEEE ACCESS, Egyptian Informatics Journal, and PeerJ. She can be contacted at email: abeerdesuky@azhar.edu.eg.