mathematics

MDPI

Article

An Efficient Parallel Reptile Search Algorithm and Snake Optimizer Approach for Feature Selection

Ibrahim Al-Shourbaji [1,2], Pramod H. Kachare [3], Samah Alshathri [4,*], Salahaldeen Duraibi [1], Bushra Elnaim [5] and Mohamed Abd Elaziz [6,7,8,*]

[1] Department of Computer and Network Engineering, Jazan University, Jazan 45142, Saudi Arabia; alshourbajiibrahim@gmail.com (I.A.-S.); sduraibi@jazanu.edu.sa (S.D.)
[2] Department of Computer Science, University of Hertfordshire, Hatfield AL10 9AB, UK
[3] Department of Electronics & Telecomm, Engineering, Ramrao Adik Institute of Technology, Nerul, Navi Mumbai 400706, Maharashtra, India; kachare.pramod1991@gmail.com
[4] Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
[5] Department of Computer Science, College of Science and Humanities in Al-Sulail, Prince Sattam bin Abdulaziz University, Kharj 16278, Saudi Arabia; b.elamin@psau.edu.sa
[6] Faculty of Science & Engineering, Galala University, Suze 435611, Egypt
[7] Artificial Intelligence Research Center (AIRC), College of Engineering and Information Technology, Ajman University, Ajman 346, United Arab Emirates
[8] Department of Mathematics, Faculty of Science, Zagazig University, Zagazig 44519, Egypt
* Correspondence: sealshathry@pnu.edu.sa (S.A.); abd_el_aziz_m@yahoo.com (M.A.E.)

Abstract: Feature Selection (FS) is a major preprocessing stage which aims to improve Machine Learning (ML) models' performance by choosing salient features, while reducing the computational cost. Several approaches are presented to select the most Optimal Features Subset (OFS) in a given dataset. In this paper, we introduce an FS-based approach named Reptile Search Algorithm–Snake Optimizer (RSA-SO) that employs both RSA and SO methods in a parallel mechanism to determine OFS. This mechanism decreases the chance of the two methods to stuck in local optima and it boosts the capability of both of them to balance exploration and explication. Numerous experiments are performed on ten datasets taken from the UCI repository and two real-world engineering problems to evaluate RSA-SO. The obtained results from the RSA-SO are also compared with seven popular Meta-Heuristic (MH) methods for FS to prove its superiority. The results show that the developed RSA-SO approach has a comparative performance to the tested MH methods and it can provide practical and accurate solutions for engineering optimization problems.

Keywords: classification; feature selection; metaheuristic algorithms; reptile search algorithm; snake optimizer

MSC: 68Txx

check for updates

Citation: Al-Shourbaji, I.; Kachare, P.H.; Alshathri, S.; Duraibi, S.; Elnaim, B.; Abd Elaziz, M. An Efficient Parallel Reptile Search Algorithm and Snake Optimizer Approach for Feature Selection. *Mathematics* **2022**, *10*, 2351. https://doi.org/10.3390/math10132351

Academic Editor: Pasi Luukka

Received: 11 June 2022
Accepted: 27 June 2022
Published: 5 July 2022

1. Introduction

With the rapid development of contemporary enterprises and the digital world, the transformation process of data into useful information has become more and more difficult due to the large amount of data produced by different sources. Machine Learning (ML) can play an essential role in Knowledge Discovery, which is categorized into a number of tasks, including data preprocessing (data preparation, reduction, and transformation), pattern evaluation, and knowledge presentation [1].

FS is a major preprocessing step, which can improve the ML model's performance by eliminating the size of features and simplifying the classification problem [2,3]. The biggest concern of the FS process is to discard the irrelevant, redundant, and noisy features from the whole set of features to derive a subset of representative features. This process is used in many areas of science such as data classification [4], image processing [5], text categorization [6], data clustering [7], and signal processing [8]. The primary objective of the FS process is to find OFS from highly discriminated features that result in high classification accuracy.

Recently, several MH methods have been introduced in the literature to simulate the behaviors of natural phenomena or living organisms for various problems. These methods show potential in selecting OFS from a given dataset and solving diverse and complex optimization problems, such as scheduling engineering design, production problems, and ML [9–11]. MH methods use exploration and exploitation principles [12,13]. Exploration refers to the ability to search the entire search space; this ability is linked to avoiding local optima and resolving traps in local optima. On the other hand, exploitation is the ability to investigate nearby prospective ideas to increase local quality. A proper balance between these two properties gives an excellent algorithm performance [14].

Various MH methods such as Multi-Verse Optimizer (MVO) [15], Particle Swarm Optimization (PSO) [16], Whale Optimization Algorithm (WOA) [17], Gray Wolf Optimizer (GWO) [18], and Salp Swarm Algorithm (SSA) [19] are some of the commonly applied MH methods for FS. However, the computational cost, classification accuracy, and finding a global optimum by these methods still need more focus and efforts to improve.

One can combine two or more MH approaches to develop a new one with a higher performance that can achieve a convincing balance between the two MH principles rather than using each of them alone for the problem of FS [20–22]. In the present work, a novel combined MH-based approach named Reptile Search Algorithm–Snake Optimizer (RSA-SO) is introduced to solve FS. The RSA-SO approach utilizes the best characteristics and capabilities of both the RSA [23] and SO [24] algorithms to obtain an optimal subset of informative features, where both are collaborated in a parallel mechanism. RSA and SO methods are among the most recent MH algorithms, and they show promising capabilities to solve FS problems with efficient balance between exploration and exploitation aspects. The parallel collaboration helps to decrease the chance of the two methods becoming stuck at local optima and it boosts the capability of both of them in balancing between exploration and explication. The contributions of this paper can be summarized as follows:

- An efferent RSA-SO approach is introduced, which merges RSA and SO in a parallel mechanism to enhance the selection process of the OFS.
- The developed RSA-SO is tested on twelve datasets from different fields and it is applied to solve two well-known engineering optimization problems with constraints.
- The results show that the RSA-SO performed well when it is compared to other popular MH methods, and it can also provide a practical and accurate solution for engineering optimization problems.

The structure of the paper is as follows. The next section gives an overview of RSA and SO. The details of the proposed RSA-SO approach are described in Section 3, while Section 4 analyzes and discusses the experimental results. Finally, the conclusion and future research directions are given in the last section.

2. Materials and Methods

2.1. Reptile Search Algorithm (RSA)

RSA is a nature-inspired MH approach based on crocodiles' encircling and hunting behavior, introduced by [23] in 2022. It is a gradient-free method that begins with generating random candidate solutions. The ith candidate solution of the jth input feature $x_{i,j}$ can be calculated as follows:

$$x_{i,j} = rand_{\in U(0,1)} * \left(UB_j - LB_j\right) + LB_j \;\; i \in \{1, \cdots, N\} \text{ and } j \in \{1, \cdots, M\} \quad (1)$$

where UB_j and LB_j are upper and lower boundaries of the jth feature, $rand_{\in U(0,1)}$ stands for uniformly distributed random number in the range [0, 1], N is the total number of candidate solutions, and M is the feature dimension.

Like the other MH algorithms, RSA works in two principles: exploration and exploitation. These principles are facilitated by crocodiles' movement while searching for food. In the RSA, the total iterations are split into four stages to take advantage of the natural behavior of crocodiles. In the first two stages, RSA accomplishes the exploration based on the encircling behavior comprising the high and the belly walking movements. Crocodiles begin their encircling to search the region, facilitating a more exhaustive search of the solution space, and this can be mathematically modeled as:

$$x_{i,j}(g+1) = \begin{cases} [-n_{i,j}(g) \cdot \gamma \cdot Best_j(g)] - \left[rand_{\in[1,\ N]} \cdot R_{i,j}(g)\right], & g \leq \frac{G}{4} \\ ES(g) \cdot Best_j(g) \cdot x_{(rand_{\in[1,\ N]},j)}, & g \leq \frac{2G}{4} \ \ and\ g > \frac{G}{4} \end{cases} \quad (2)$$

where $Best_j(g)$ is the best solution for jth feature, $n_{i,j}$ refers to the hunting operator for the jth feature in the ith solution (calculated as in Equation (3)), and parameter γ controls the exploration accuracy and is set as 0.1. The reduce function $R_{i,j}$ is used to reduce the search region and is computed as in Equation (6), $rand_{\in[1,\ N]}$ is a number between 1 and N used to randomly select one of the possible candidate solutions, and Evolutionary Sense $ES(g)$ stands for the probability ratio which reduces from 2 to -2 over iterations, calculated as in Equation (7).

$$n_{i,j} = Best_j(g) \times P_{i,j}, \quad (3)$$

where $P_{i,j}$ indicates the percentage difference between the jth value of the best solution to its corresponding value in the current solution and is calculated as:

$$P_{i,j} = \theta + \frac{x_{i,j} - \mu(x_i)}{Best_j(g) \times (UB_j - LB_j) + \epsilon}, \quad (4)$$

where θ denotes a sensitive parameter that controls the exploration performance, ϵ is a small floor value, and $\mu(x_i)$ refers to the average solutions and is defined as:

$$\mu(x_i) = \frac{1}{n}\sum\nolimits_{j=1}^{n} x_{i,j}, \quad (5)$$

$$R_{i,j} = \frac{Best_j(g) - x_{(rand_{\in[1,\ N]},j)}}{Best_j(g) + \epsilon}, \quad (6)$$

$$ES(g) = 2 \times rand_{\in\{-1,\ 1\}} \times \left(1 - \frac{1}{G}\right), \quad (7)$$

where the value 2 acts as a multiplier to provide correlation values in the range [0, 2], and $rand_{\in\{-1,\ 1\}}$ is a random integer number between $\{-1, 1\}$.

In the last two stages, RSA exploits (hunting) the search space and approaches the optimal solution by using hunting coordination and cooperation. The candidate solution can update its value during the exploitation stage using the following:

$$x_{i,j}(g+1) = \begin{cases} rand_{\in[-1,1]} \cdot Best_j(g) \cdot P_{i,j}(g), & g \leq \frac{3G}{4}\ and\ g > \frac{2G}{4} \\ [\epsilon \cdot Best_j(g) \cdot n_{i,j}(g)] - \left[rand_{\in[-1,1]} \cdot R_{i,j}(g)\right], & g \leq G\ and\ g > \frac{3G}{4} \end{cases} \quad (8)$$

The quality of candidate solutions at each iteration is measured using the predefined FF. the algorithm stops after G iterations, and a candidate solution with the least FF is used as OFS.

2.2. Snake Optimizer (SO)

SO is a MH algorithm proposed by [24] in 2022 to mimic the mating behavior of snakes. Mating happens when the temperature is low and food is available. The SO, like other MH methods, initializes random candidate solutions using Equation (1). This method divides the swarm into male and female groups equally using the following:

$$\begin{gathered} N_{male} \approx \frac{N}{2} \\ N_f = N - N_{male} \end{gathered} \tag{9}$$

where N is the number of individuals, N_{male} refers to the male individuals, and N_{female} refers to the female individuals.

In each iteration, the best individual candidate solution (food position f_{food}) is found by analyzing each group for individual best male $f_{best,\ male}$ and best female $f_{best,female}$.

The Temperature (T) and the Food Quantity (FQ) can be defined as:

$$\begin{gathered} T = exp\left(\frac{-g}{T}\right) \\ FQ = c_1\ exp\left(\frac{g-T}{T}\right) \end{gathered} \tag{10}$$

where g is the current iteration, T is the total number of iterations, and c_1 is a constant equal to 0.5.

When FQ < Threshold (Threshold = 0.5), the snakes search for food by selecting a random position and then update their position. To mathematically model the exploration behavior of the male and female snakes, the following can be used:

- For male snakes:

$$\begin{gathered} x_{i,j}(g+1) = x_{(rand_{\in[1,N/2]},j)}(g) \pm c_2 \times A_{i,male}\Big((UB - LB) \times rand_{\in U(0,1)} + LB\Big), \\ \text{where } A_{i,male} = exp\left(\frac{-f_{rand,male}}{f_{i,male}}\right) \end{gathered} \tag{11}$$

where $x_{i,j}$ is ith the male snake position, $x_{(rand_{\in[1,\ N/2]},\ j)}$ refers to the position of a random male snake, $rand$ is a random number between 0 and 1, $A_{i,male}$ is the ability to find the food by the male, $f_{rand,male}$ is the fitness of the earlier selected random male snake, and $f_{i,male}$ is the fitness of ith male in the group. The flag direction operator $\pm$ (i.e., diversity factor) can be used to randomly scan all the possible directions in the given search space.

- For female snakes:

$$\begin{gathered} x_{i,j} = x_{(rand_{\in[1,\ N/2]},j)}(g+1) \pm c_2 \times A_{i,female}\Big((UB - LB) \times rand_{\in U(0,1)} + LB\Big), \\ \text{where } A_{i,female} = exp\left(\frac{-f_{rand,female}}{f_{i,female}}\right) \end{gathered} \tag{12}$$

where $x_{i,j}$ is ith the female snake position, $x_{(rand_{\in[1,\ N/2]},j)}$ is the position of a random female snake, $A_{i,female}$ refers to her ability to find food, $f_{rand,female}$ is the fitness of the earlier selected random female snake, and $f_{i,female}$ is the fitness of ith individual in the female group.

In the exploitation phase, SO uses two conditions to find the best solutions and they are:

1. If FQ < Threshold (T > 0.6), then the snakes move to find only:

$$x_{i,j}(g+1) = x_{food} \pm c_3 \times T \times rand \times \Big(x_{food} - x_{i,j}(g)\Big) \tag{13}$$

   where $x_{i,j}$ is the position of individuals, either male or female; x_{food} is the position of the best individuals; and c_3 is a constant equal to 2.

2. If FQ < Threshold (Threshold < 0.6), then the snakes will be in two modes, either fighting or mating. The fighting and mating models can be represented as the follows:

- Fighting mode

The fighting ability of the male agent F_{male} can be written as:

$$x_{i,j}(g+1) = x_{i,j}(g) \pm c_3 \times F_{i,\ male} \times rand \times \left(x_{best,female} - x_{i,male}(g)\right), \\ \text{where } F_{i,\ male} = exp\left(\frac{-f_{best,f}}{f_i}\right) \tag{14}$$

where $x_{i,j}$ refers to the ith male position and $x_{best,female}$ refers to the position of the best individual in the female group. Similarly, the fighting ability of the male agent $F_{i,male}$ can be written as:

$$x_{i,j}(g+1) = x_{i,j}(g) \pm c_3 \times F_{i,female} \times rand \times \left(x_{best,male} - x_{i,f}(g+1)\right), \\ \text{where } F_{i,\ female} = exp\left(\frac{-f_{best,male}}{f_i}\right) \tag{15}$$

where $x_{i,j}$, refers to the ith female position, $x_{best,male}$ refers to the position of the best individual in the male group, and $F_{i,female}$ is the fighting ability of the female agent.

- Mating mode

  In this mode, the male and female agents can update their positions as:

$$x_{i,male}(g+1) = x_{i,m}(g) \pm c_3 \times MM_{i,male} \times rand \times \left(Q \times x_{i,female} - x_{i,male}(g)\right), \\ \text{where } MM_{i,male} = exp\left(\frac{-f_{i,female}}{f_{i,male}}\right) \\ x_{i,female}(g+1) = x_{i,f}(g) \pm c_3 \times MM_{i,female} \times rand \times \left(Q \times x_{i,male} - x_{i,female}(g+1)\right), \\ \text{where } MM_{i,female} = exp\left(\frac{-f_{i,male}}{f_{i,female}}\right) \tag{16}$$

where $x_{i,m}$ and $x_{i,f}$ are the positions of ith of male and female agents, and $MM_{i,male}$ and $MM_{i,female}$ refer to the mating ability of males and females.

3. Proposed RSA-SO Method

FS is a multi-objective problem where the minimal number of OFS and higher classification accuracy are simultaneously achieved [25]. The literature survey on different MH algorithms explores various nature-inspired phenomena to effectively search for the best solutions in a given search space. A combination of these MH algorithms is reported to enhance the overall performance by complementing the other's exploration and exploitation processes, which in turn can decrease the probability of trapping in local optima.

RSA and SO methods are among the most recent MH algorithms, showing promising capabilities to solve several problems with an efficient balance between exploration and exploitation aspects. In this work, RSA and SO methods collaborate in a parallel strategy to solve an FS problem. The primary objective of the parallel mechanism is that if one of the algorithms cannot improve the candidate solutions or becomes stuck in local optima, the other algorithm moves the current candidate solutions into another search region where some better solutions might be found.

Figure 1 provides the procedural steps of the RSA-SO. At first, the hyper-parameters of RSA, SO, and the shared ones are initialized. A uniformly distributed random number generator in the range $[-1, 1]$ is employed to initialize N candidate solutions for M features, as described earlier in the RSA section (Equation (1)).

At the start of each iteration, the population (i.e., candidate solutions) is equally divided into two parts between the RSA and SO algorithms. For the gth iteration, candidate solutions $\{x_{i,j}(g),\ 1 \le i \le N \text{ and } 1 \le j \le M\}$ are split into two parts. The first half is passed to RSA and the second half is passed to SO. It can be mathematically seen as follows:

$$x_i^{RSA-SO} = \begin{cases} x_i^{RSA}, & 1 \le i \le N/2 \\ x_i^{SO}, & N/2 < i \le N \end{cases} \tag{17}$$

In the first iteration, both RSA and SO are executed in a parallel manner on the respective parts, and candidate solutions are updated according to Equations (2)–(8) in the RSA and Equations (9)–(16) in SO methods. At the end of the first titration, the updated candidate solutions from both algorithms are evaluated using the Fitness Function (FF). The solutions are sorted in ascending order using the Quick Sort algorithm based on their fitness values. The candidate solutions with the smaller fitness values are selected from each part of the population. The top $N/2$ solutions from the entire population are found and passed to both algorithms for the next iteration. The complete set of candidate solutions after each iteration can be generated by merging solutions from both algorithms, as in Equation (17).

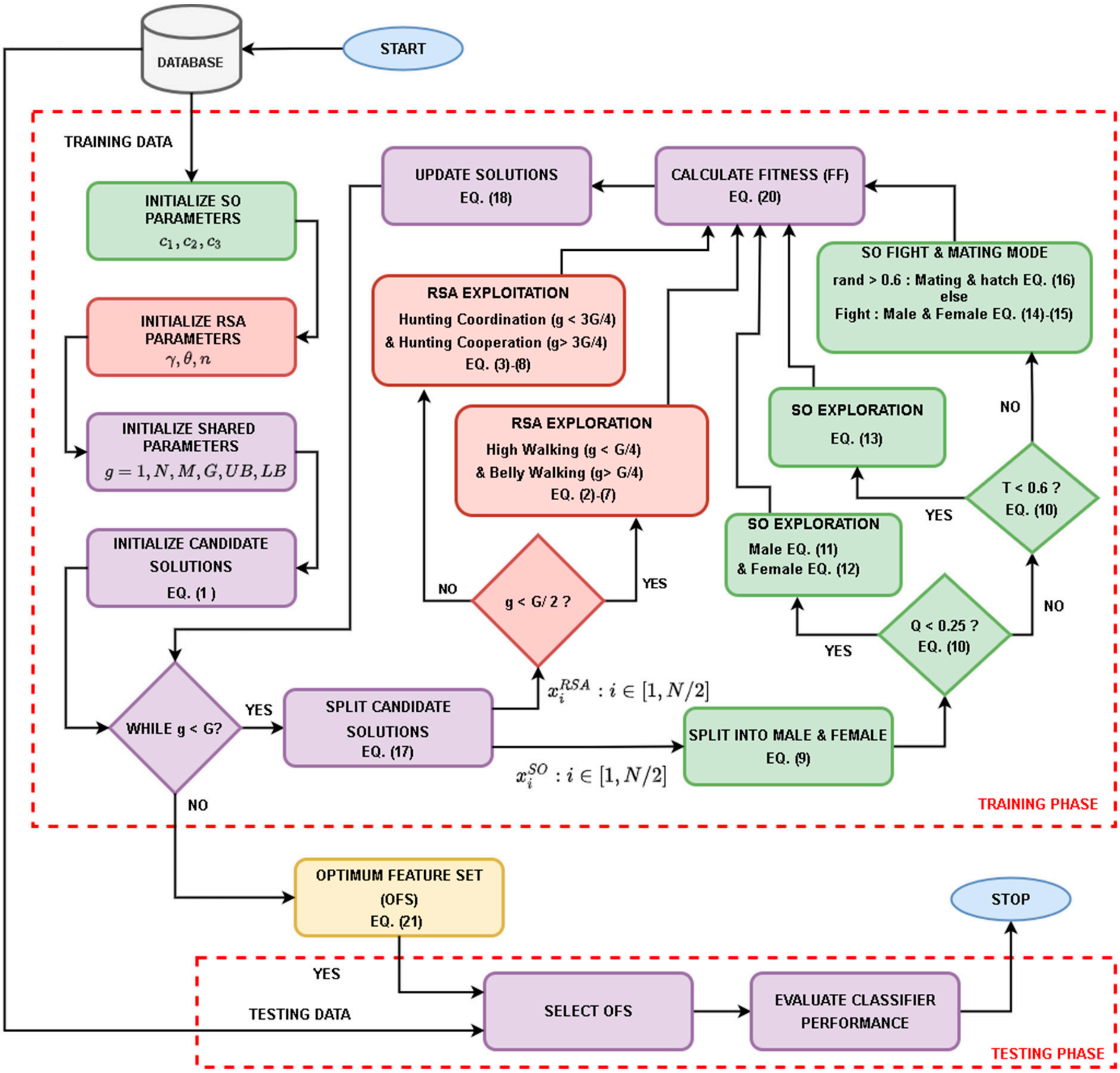

Figure 1. Structure of RSA-SO approach.

The sorting finds the best $N/2$ solutions from the entire population with fitness values smaller than any solution other than the selected ones. These found solutions may be distributed differently amongst the RSA and SO algorithms. A set of improved low-fitness candidate solutions $\hat{x}_i(g)$ is obtained by swapping high-fitness candidate solutions with the low-fitness candidate solutions found by the complementary algorithm. The candidate solutions can be updated as follows:

$$x_i^{RSA}(g+1) = x_i^{SO}(g+1) = \hat{x}_i(g), \qquad 1 \le i \le N/2 \tag{18}$$

where $\hat{x}_i(g) = x_{argmin\ (f_i(g))}(g)$.

If the found candidate solutions comprise more solutions from RSA than SO, then the high-fitness candidate solutions from SO are replaced by solutions found by RSA and vice versa. Hence, the RSA will dominate the next iteration. On the other hand, if the found candidate solutions comprise more solutions from SO than RSA, then the SO will dominate in the next iteration. Lastly, if an equal number of low-fitness candidate solutions are found by both algorithms, then the next iteration displays the codominance of both algorithms. All three cases can be summarized as,

$$\begin{array}{l} if\ \dim(argmin\ f_i \le N/2) > \dim(argmin\ f_i \le N/2)\ \ then\ \ \text{RSA dominates } (i+1) \text{ iteration} \\ if\ \dim(argmin\ f_i \le N/2) < \dim(argmin\ f_i \le N/2)\ \ then\ \ \text{SO dominates } (i+1) \text{ iteration} \\ if\ \dim(argmin\ f_i \le N/2) = \dim(argmin\ f_i \le N/2)\ \ then\ \ \text{RSA \& SO codominates } (i+1) \text{ iteration} \end{array} \tag{19}$$

An example of candidate solution optimization using $N = 8$ is shown in Figure 2. Candidate solutions from RSA (red) and SO (green) are identified using different colors of the bounding boxes. The corresponding fitness value marks each candidate solution with a maximum of 1 (darker shade fill) and a minimum of 0 (lighter shade fill). The top $N/2 = 4$ found low-fitness solutions (ligher shade fill) are marked by an additional bounding box (dotted black). In the case of Figure 2a, the gth iteration marks three solutions from RSA and only one from SO as low-fitness. In the (g+1)th iteration, a high-fitness solution from RSA is replaced with a low-fitness solution from SO, while three high-fitness solutions in SO are replaced with three low-fitness solutions from RSA. Hence, the (g+1)th iteration is dominated by RSA, as observed in Figure 2a's selected solutions. A contradictory situation is presented in Figure 2b, where three solutions from SO and only one from RSA are marked low-fitness. Hence, solutions for the (g+1)th iteration are obtained by replacing three high-fitness solutions from RSA with low-fitness solutions from SO, and vice versa. Hence, the (g+1)th iteration is dominated by SO, as observed in Figure 2b's selected solutions. Finally, Figure 2c shows the equal number of solutions found by both algorithms. Hence, even after the replacement, both algorithms have equal shares indicating the codominance of both algorithms for the (g+1)th iteration. It should be noted that after optimizing, both algorithms will continue the next iteration using the exact same set of low-fitness candidate solutions, except for the sequence of solutions, as seen in Figure 2a,c. This can effectively coordinate and improve global exploration and local exploitation in the search space.

In the next iteration, $x_i(g+1)$, the generated population is first split into two parts using Equation (19), and each part is passed as an input to the RSA and SO methods to simultaneously search other regions in the feature space. After finishing the second iteration, the obtained candidate solutions are sorted using FF. A new population that is composed of the best candidate solutions is obtained from each population part. This process continues until the termination condition is satisfied (i.e., the maximum number of iterations is reached). The pseudo-code of the RSA-SO is provided in Algorithm 1.

Algorithm 1: Pseudo-code of the interdicted RSA-SO approach.

1. Split the dataset into training and testing
Training Phase
2. Load training dataset
3. Initialize SO parameters c_1, c_2, c_3
4. Initialize RSA parameters $\gamma,\ \theta, n$
5. Initialize shared parameters $N,\ M,\ G,\ UB,\ LB$
6. Initialize candidate solutions Equation (1)
7. for g = 1 to G do
8. Split candidate solutions for RSA and SO using Equation (17)
9. Revise candidate solutions $\hat{x}_i$ using RSA Equations (2)–(8) and SO Equations (9)–(16)
10. Evaluate FF (f) using Equation (20) for revised candidate solutions
11. Update RSA and SO solutions for next iteration using Equation (18)
12. Calculate complete solution for next iteration
13. end for
14. Extract OFS for candidate solution with minimum FF using threshold of 0.5
Testing Phase
15. Load testing dataset
16. Select only optimum features as described in OFS Equation (21)
17. Evaluate performance using KNN classifier

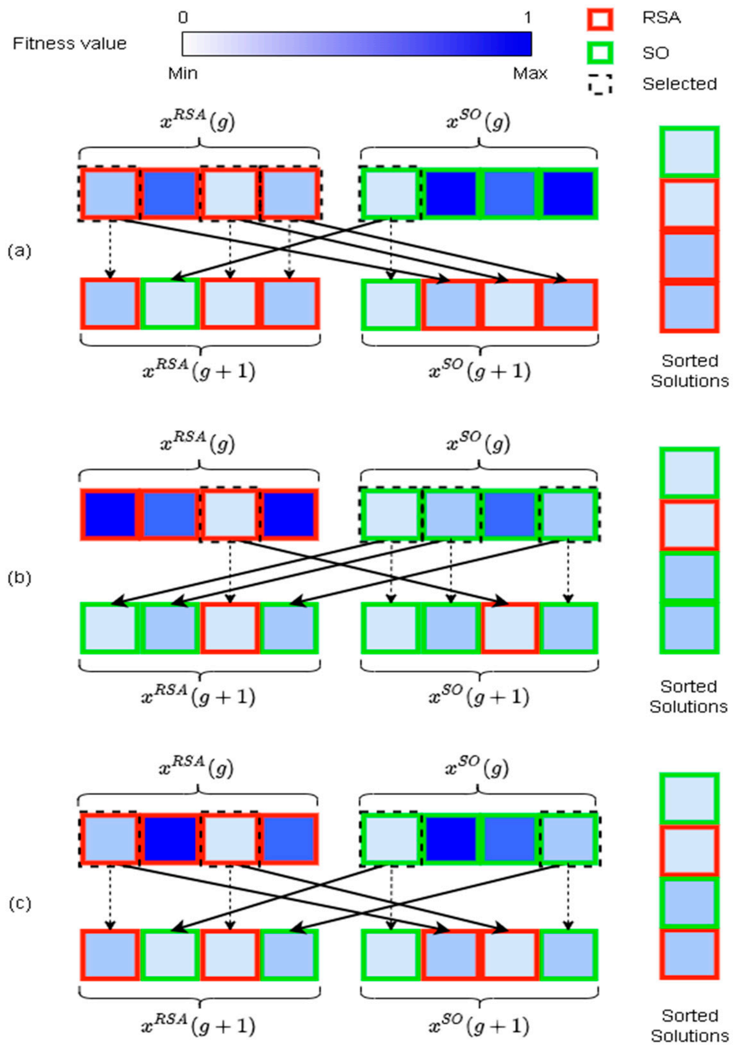

Figure 2. Example of optimizing the low-fitness candidate solutions in the proposed RSA-SO algorithm for dominance of (**a**) RSA, (**b**) SO, and (**c**) codominance of both, as shown in Equation (19).

The K-Nearest Neighbor (KNN) classifier with k = 5 is used as the FF. The threshold value is set to 0.5 to produce a small number of features, as recommended by the work of [26,27]. The solution with the smallest number of features and highest accuracy is the best one (smallest fitness f) and it is defined as:

$$f_i = \alpha \times \gamma + \beta \times \frac{SF_i}{M} \tag{20}$$

where γ is the error rate of the KNN, SF_i is the number of OFS, and M is the number of features in the original dataset. α and β are two weights that control the importance of classification quality and feature reductions; the value of α in the range of [0, 1] and the value of β is $1-\alpha$. The parameters α and β are set to 0.99 and 0.01, respectively, in this work [28,29], and each feature in the OFS follows:

$$SF_i = \begin{cases} 1 & if\ x_i > 0.5, \\ 0 & otherwise, \end{cases} \tag{21}$$

4. Experiments and Results

To assess the capability of the RSA-SO, its performance is compared with other MH methods, including PSO [16], GWO [18], MVO [15], WOA [17], SSA [19], RSA [23], and SO [24], on twelve datasets; the results are provided in this section. All the experiments are implemented using Python scikit-learn and conducted on a 3.13 GHz PC with 16 GB RAM and Windows 10 operating system.

4.1. Dataset

The RSA-SO is tested on eight datasets taken from the UCI data repository, and each of them is split into 80% of the samples used for training and the remaining used for testing. Table 1 summarizes the details of the used datasets.

Table 1. List of the datasets used in the experiments.

No.	Dataset	Instances	Features	Classes	Domain
1	Breastcancer	699	9	2	Biology
2	BreastEW	569	30	2	Biology
3	Churn	3150	16	2	Telecom
4	HeartEW	270	13	2	Biology
5	IonosphereEW	351	34	2	Electromagnetic
6	KrvskpEW	3196	36	2	Game
7	SonarEW	208	60	2	Biology
8	SpectEW	267	22	2	Biology
9	Tic-tac-toe	958	9	2	Game
10	Vote	300	16	2	Politics
11	Chemical Water	178	13	3	Chemistry
12	Zoo	101	16	6	Artificial

4.2. Parameter Settings

To compare RSA-SO with other methods, six popular methods in the field of FS are selected. The population size and the maximum number of iterations are empirically set as 20 and 100, respectively. All the methods are run 20 times independently. The parameter settings of these methods are defined according to their implementations in the original work, and they are listed in Table 2.

Table 2. Parameter settings.

Algorithm	Parameters
PSO	$c_1 = c_2 = 2$, $w_{min} = 0.1$ and $w_{max} = 0.9$
GWO	α variable decreases linearly from 2 to 0, C is a random value € [0, 2], and A linearly decreases from 1 to -1
MVO	$WEP_{max} = 1$, $WEP_{min} =$ decreases from 2 to 0 and $p = 6$
WOA	α decreases from 2 to 0 and α_2 decrease from -1 to -2
SSA	c_2 and c_2 are random values € [1, 0]
RSA	$\gamma = 0.9$, $\theta = 0.5$, UB and LB vary according to features in the dataset
SO	$c_1 = 0.5$, $c_2 = 0.05$, $c_3 = 2$, x_{max} and x_{min} vary according to features in the dataset
RSA-SO	It uses the parameters of the RSA and SO

4.3. Results and Discussion

A set of widely used performance measures is employed to assess the obtained results by the RSA-SO and other FS methods. These metrics include, classification accuracy, number of selected OFS, fitness values (best, worst, average (Avg), and standard deviation (STD)), and computational time consumed by each method. The Friedman ranking test is applied to rank each method for a fair comparison. Moreover, the convergence behavior of the introduced RSA-SO and other methods is provided in this section.

Figure 3 shows the distribution of the best-selected candidate solutions obtained by RSA (in red color) and SO (in green color) for twelve datasets. It provides the number of iterations on the x-axis and the selected solutions on the y-axis. It can be noticed in Figure 2 that the RSA and SO begin by exploring the search space, followed by exploiting the best candidate solution in the feature space. For example, in the initial 25 iterations in the KrvskpEW dataset, more candidate solutions are selected from the first half of the revised candidate solutions, indicating that high walking in the RSA is more effective than SO. Similarly, the last 25 iterations indicate that the hunting cooperation process in RSA exploits candidate solutions more effectively than SO. The dominance of SO during iterations 25 to 50 and 50 to 75 shows that exploration using belly walking and exploitation using hunting coordination in RSA are not very effective. Similar observations can be made for SpectEW, Tic-tac-toe, and Chemical Water datasets. In IonosphereEW and Votes datasets, most of the iterations are dominated by the SO. On the other hand, most iterations for the Breastcancer dataset show approximately equal candidate solutions selected from both methods, indicating the codominance of both algorithms. Similar codominance can be observed in the first 25 and the last 25 iterations for BreastEW, Churn, HeartEW, Sonar, and Zoo datasets.

Tables 3 and 4 compare all the FS approaches in terms of the average testing accuracy and the number of OFS. In MA methods, the solution with the highest classification accuracy and minimum number of features is the best one in the population that needs to be accomplished. In Table 3, the RSA-SO scored the best accuracy compared to other techniques in eight out of twelve datasets. This can be interpreted by the improved capability of the RSA-SO in broadly searching the high-performance regions in the search space. For IonosphereEW, SO is placed first, while for SonarEW and Zoo datasets, GWO performed the best. Both WOA and RSA-SO achieved similar accuracy results on the Chemical Water dataset.

As per the results in Table 4, the introduced RSA-SO had the smallest value of the selected OFS in nine out of twelve datasets. This confirms the efficiency of the proposed RSA-SO in eliminating irrelevant features in the datasets and reducing the search space. However, RSA-SO had the same results in Breastcancer, IonosphereEW, and Tic-tac-toe datasets, while the RSA method gained the best results only in the Churn dataset and the SO attained the best results in SonarEW and Vote datasets. A similar number of OFS is selected by all the methods on Breastcancer and Tic-tac-toe datasets. GWO selected the smallest number of OFS on the SonarEW dataset.

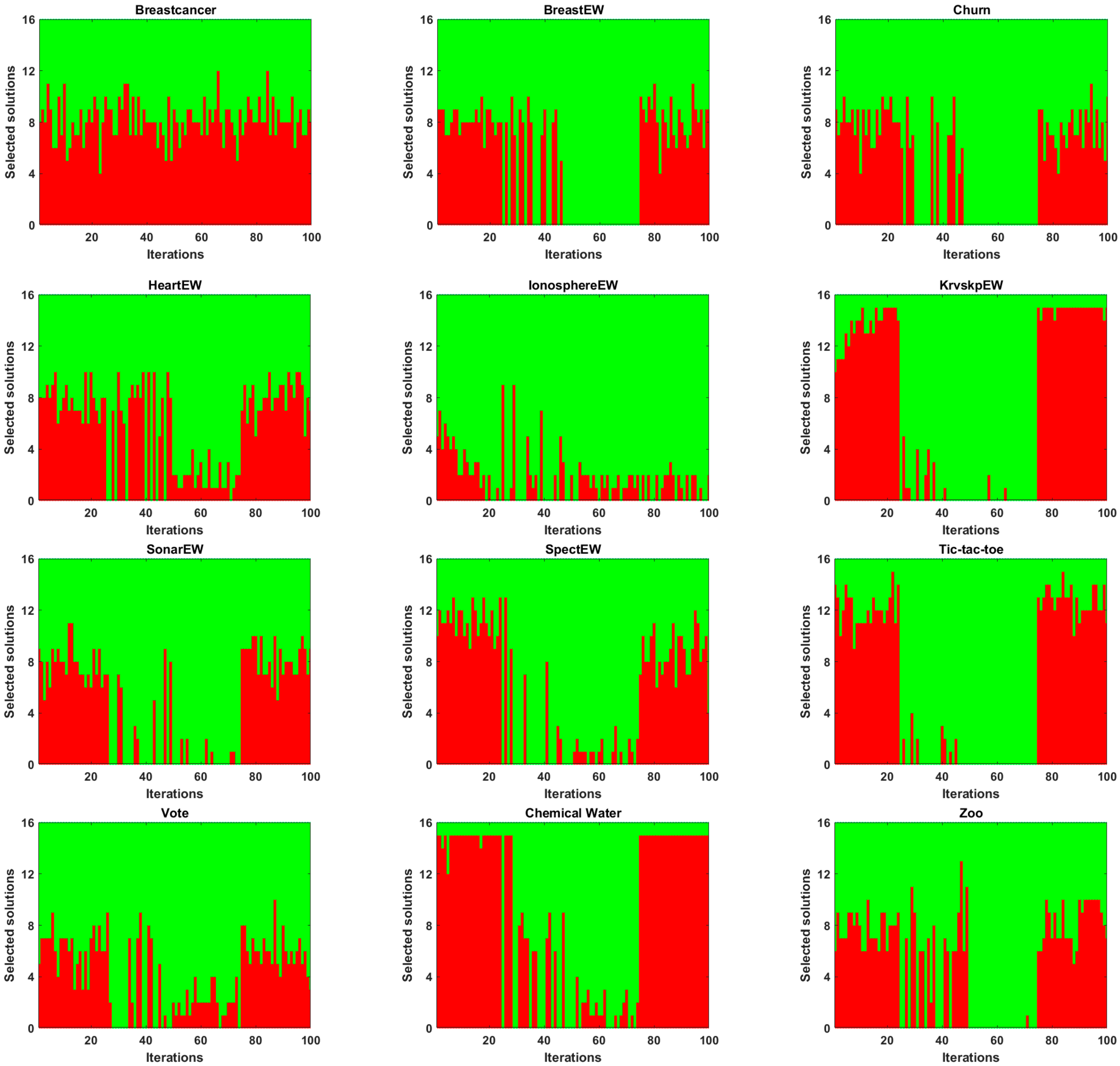

Figure 3. Distribution of best selected candidate solutions between RSA and SO for different datasets.

Table 5 records a summary of the results obtained by the RSA-SO against the other MH algorithms for different datasets. It also presents ranks of MH algorithms for each dataset depending on average, STD, best, and worst of fitness values. From Table 5, it can be observed that the RSA-SO earned the first rank in nine out of twelve datasets. For Breastcancer, IonosphereEW, and Zoo datasets, PSO, MVO, and RSA achieved first ranks while the proposed RSA-SO achieved ranks of 4, 4, and 2, respectively. The RSA-SO provides the best fitness values in eight datasets while all the methods have similar average best fitness on the Tic-tac-toe dataset. RSA-SO has the smallest worst fitness value in seven datasets, while it has similar average best fitness on the Breastcancer dataset. Moreover, the RSA-SO has better Avg and STD of fitness values in eight and six datasets, respectively. RSA-SO and SSA had the same Avg and STD of fitness values on the HeartEW dataset, while WOA, SSA, RSA, and RSA-SO had the same Avg and STD on the Chemical Water dataset. These results prove the capability of the introduced RSA-SO in sustaining a stable balance between the two main principles of MH methods.

Table 3. Results of RSA-SO and other methods in terms of classification accuracy.

Dataset	PSO	GWO	MVO	WOA	SSA	RSA	SO	RSA-SO
Breastcancer	99.1401	99.1474	99.1473	99.1255	99.1620	99.1839	99.1766	99.2132
BreastEW	95.4001	95.5495	95.3739	95.3726	95.6496	95.7337	95.3722	96.1579
Churn	89.6476	95.6078	92.9740	95.5873	96.3150	94.5260	93.2201	96.4688
HeartEW	73.7800	80.9395	76.1839	84.5661	99.2293	99.1479	99.3587	99.7400
IonosphereEW	93.1179	93.4326	93.1345	92.3234	92.6408	92.9915	93.5485	92.5147
KrvskpEW	96.3220	95.8335	96.7343	96.0780	96.4841	95.5679	97.0928	97.1065
SonarEW	89.2195	91.0967	88.8637	87.3249	88.4737	87.5277	87.7426	90.2545
SpectEW	87.0948	85.8567	87.2818	86.0475	86.8869	86.0594	87.4435	87.5323
Tic-tac-toe	82.7718	82.7874	82.6882	82.7665	82.7874	82.6150	82.6934	82.8031
Vote	64.0552	64.3368	64.3513	63.2006	63.9436	63.3290	62.4167	64.5815
Chemical Water	99.9503	99.9671	99.9607	99.9944	99.9837	99.9888	99.9713	99.9944
Zoo	96.7110	97.7109	96.9901	96.4038	97.3632	97.1310	96.8753	97.3928

Table 4. Comparison between RSA-SO and other methods in terms of average OFS.

Dataset	PSO	GWO	MVO	WOA	SSA	RSA	SO	RSA-SO
Breastcancer	9	9	9	9	9	9	9	9
BreastEW	3	3	3	9	3	7	3	2
Churn	14	10	13	9	11	8	12	11
HeartEW	13	10	11	5	3	2	5	1
IonosphereEW	4	4	5	4	4	4	4	4
KrvskpEW	31	32	29	31	29	29	27	23
SonarEW	27	17	28	26	27	28	20	23
SpectEW	11	11	11	14	11	13	10	8
Tic-tac-toe	9	9	9	9	9	9	9	9
Vote	6	5	7	5	7	6	3	6
Chemical Water	9	6	7	2	3	2	5	1
Zoo	13	8	11	8	9	6	10	5

The average computational time in seconds for the RSA-SO and the other MH methods, which is computed over 20 independent runs on all the datasets, is provided in Table 6. According to the results in Table 6, the average computational time consumed by the RSA-SO is lower than PSO, GWO, MVO, WOA, SSA, RSA, and SO in five datasets. This is because both the RSA and SO run at the same time in a parallel manner at each iteration, which decreases the running time. Taking into account the accuracy rate and running time, the introduced RSA-SO proves to be superior since it gained a high accuracy rate and competitive execution time on most of the datasets. WOA ranked first for BreastEW, while SSA placed first for HeartEW and Tic-tac-toe datasets. GWO does not need much effort on SpectEW and Chemical Water, and PSO needed lower time on the Zoo dataset.

The convergence behavior of the introduced RSA-SO is shown over 100 iterations on the x-axis in Figure 4 and the average fitness values on the y-axis. Figure 4 presents the convergence curves of the best solution obtained after executing each method 20 runs. In Figure 4, one can observe that RSA-SO has a faster and better convergence than the other methods among the used twelve datasets except three of them, namely, IonosphereEW, SpectEW, and Zoo datasets. However, RSA-SO has the fastest convergence speed on nine out of twelve datasets, which proves its suitability for the problem of FS.

Table 5. Best, worst, Avg, and STD fitness values obtained by different methods.

Dataset	Metric	PSO	GWO	MVO	WOA	SSA	RSA	SO	RSA-SO
Breastcancer	Best	0.0160	0.1605	0.0160	0.1605	0.0160	0.1605	0.0160	0.1605
	Worst	0.1895	0.1895	0.1895	0.1895	0.1895	0.1895	0.1895	0.1895
	Avg.	0.0161	0.0161	0.0161	0.0161	0.0161	0.0161	0.0161	0.0161
	STD.	0.0008	0.0008	0.0010	0.0010	0.0011	0.0010	0.0011	0.0010
	Rank	1	2	3	4	5	4	5	4
BreastEW	Best	0.0492	0.0439	0.0473	0.0436	0.0401	0.0385	0.0491	0.0382
	Worst	0.0562	0.0579	0.0579	0.0578	0.0579	0.0473	0.0578	0.0491
	Avg.	0.0492	0.0439	0.0473	0.0436	0.0421	0.0486	0.0491	0.0401
	STD.	0.0019	0.0045	0.0028	0.0044	0.0044	0.0019	0.0026	0.0018
	Rank	8	4	5	3	2	6	7	1
Churn	Best	0.0418	0.0421	0.0422	0.0403	0.0406	0.0403	0.0415	0.0393
	Worst	0.1346	0.0638	0.1345	0.0817	0.0491	0.0996	0.1346	0.0484
	Avg.	0.0418	0.0421	0.0421	0.0403	0.0406	0.0403	0.0415	0.0393
	STD.	0.0354	0.0058	0.0353	0.0119	0.0025	0.0220	0.0276	0.0018
	Rank	5	6	7	2	3	2	4	1
HeartEW	Best	0.2865	0.2828	0.2864	0.1909	0.0002	0.0001	0.0003	0.0000
	Worst	0.2692	0.1967	0.2440	0.1566	0.0001	0.0001	0.0002	0.0000
	Avg.	0.1983	0.1250	0.1323	0.1286	0.0000	0.0001	0.0001	0.0000
	STD.	0.0248	0.0748	0.0528	0.0191	0.0000	0.0000	0.0001	0.0000
	Rank	8	5	7	6	2	3	4	1
IonosphereEW	Best	0.0904	0.0734	0.0848	0.1045	0.0903	0.0819	0.0706	0.0932
	Worst	0.0692	0.0661	0.0694	0.0773	0.0742	0.0706	0.0651	0.0753
	Avg.	0.0593	0.0594	0.0566	0.0650	0.0621	0.0621	0.0621	0.0594
	STD.	0.0072	0.0039	0.0075	0.0094	0.0080	0.0069	0.0034	0.0082
	Rank	2	3	1	8	7	6	5	4
KrvskpEW	Best	0.0500	0.0577	0.0518	0.0519	0.0546	0.0596	0.0453	0.0497
	Worst	0.0451	0.0502	0.0404	0.0475	0.0428	0.0518	0.0362	0.0350
	Avg.	0.0264	0.0373	0.0236	0.0379	0.0264	0.0307	0.0230	0.0230
	STD.	0.0049	0.0047	0.0077	0.0036	0.0099	0.0068	0.0089	0.0107
	Rank	5	7	3	8	4	6	2	1
SonarEW	Best	0.0917	0.0723	0.1010	0.0957	0.0966	0.0868	0.1005	0.0775
	Worst	0.1113	0.0910	0.1148	0.1291	0.1186	0.1282	0.1247	0.1004
	Avg.	0.0917	0.0723	0.1010	0.0957	0.0966	0.0868	0.1005	0.0775
	STD.	0.0111	0.0091	0.0128	0.0135	0.0140	0.0211	0.0154	0.0113
	Rank	4	2	8	5	6	3	7	1

Table 5. *Cont.*

Dataset	Metric	PSO	GWO	MVO	WOA	SSA	RSA	SO	RSA-SO
SpectEW	Best	0.1227	0.1300	0.1390	0.1338	0.1227	0.1301	0.1227	0.1190
	Worst	0.1328	0.1450	0.1307	0.1444	0.1350	0.1438	0.1286	0.1271
	Avg.	0.1227	0.1300	0.1191	0.1338	0.1227	0.1301	0.1227	0.1190
	STD.	0.0066	0.0102	0.0099	0.0088	0.0086	0.0117	0.0046	0.0061
	Rank	5	7	2	8	6	5	3	1
Tic-tac-toe	Best	0.1832	0.1832	0.1832	0.1822	0.1832	0.1853	0.1832	0.1832
	Worst	0.1806	0.1804	0.1814	0.1806	0.1804	0.1821	0.1813	0.1802
	Avg.	0.1749	0.1775	0.1780	0.1771	0.1770	0.1780	0.1776	0.1739
	STD.	0.0022	0.0018	0.0021	0.0018	0.0025	0.0023	0.0017	0.0016
	Rank	2	5	7	4	3	8	6	1
Vote	Best	0.3756	0.3688	0.3712	0.3824	0.3734	0.3848	0.3824	0.3620
	Worst	0.3597	0.3564	0.3574	0.3674	0.3615	0.3665	0.3742	0.3546
	Avg.	0.3461	0.3483	0.3484	0.3484	0.3461	0.3484	0.3575	0.3461
	STD.	0.0074	0.0055	0.0058	0.0082	0.0074	0.0101	0.0083	0.0046
	Rank	3	4	5	6	2	7	8	1
Chemical Water	Best	0.0006	0.0005	0.0006	0.0001	0.0002	0.0002	0.0004	0.0001
	Worst	0.0005	0.0003	0.0004	0.0001	0.0002	0.0001	0.0003	0.0001
	Avg.	0.0003	0.0002	0.0002	0.0001	0.0001	0.0001	0.0001	0.0001
	STD.	0.0001	0.0001	0.0001	0.0000	0.0000	0.0000	0.0001	0.0000
	Rank	6	4	5	1	2	2	3	1
Zoo	Best	0.0731	0.0420	0.0731	0.0620	0.0421	0.0421	0.0623	0.0412
	Worst	0.0406	0.0279	0.0369	0.0406	0.0315	0.0323	0.0373	0.0291
	Avg.	0.0318	0.0209	0.0214	0.0311	0.0210	0.0209	0.0213	0.0209
	STD.	0.0090	0.0077	0.0135	0.0109	0.0066	0.0074	0.0085	0.0078
	Rank	8	3	6	7	4	1	5	2

Table 6. Comparison between RSA-SO and other methods in terms of computation time.

Dataset	PSO	GWO	MVO	WOA	SSA	RSA	SO	RSA-SO
Breastcancer	15.8043	15.7762	15.7891	16.9361	18.8809	12.2260	13.9439	11.3005
BreastEW	16.9211	17.0046	16.8496	15.4815	16.4994	17.4666	18.0835	20.7034
Churn	46.4102	65.7563	46.2434	44.2000	45.3247	45.1050	44.5310	44.1699
HeartEW	15.8490	16.1843	15.8615	16.1266	13.7198	14.8492	16.6837	14.7071
IonosphereEW	16.4906	16.3805	16.4499	16.0545	20.2799	18.3783	17.7728	12.0760
KrvskpEW	23.3375	22.7748	20.0801	20.7943	17.6028	26.9347	21.4032	15.1755
SonarEW	16.0206	15.8344	15.9396	15.6063	14.8404	13.0648	17.7407	15.8526
SpectEW	15.1340	12.7896	15.0302	14.5697	24.6677	13.6435	20.6525	15.6375
Tic-tac-toe	8.1686	8.2544	8.3290	15.0991	8.0667	8.5040	12.8882	8.4919
Vote	6.4341	6.4319	6.6094	6.9672	6.4030	6. 7186	9.1398	6.2819
Chemical Water	5.3900	4.7936	4.9758	4.8917	10.4849	16.3850	13.1670	14.0011
Zoo	11.6261	13.6194	12.1658	12.9702	11.7885	24.3258	19.8634	15.4363

4.4. Performance of RSA-SO in Engineering

In this section, the performance of the RSA-SO is tested on well-known engineering problems, which are Pressure Vessel Design (PVD) and Cantilever Beam Design (CBD).

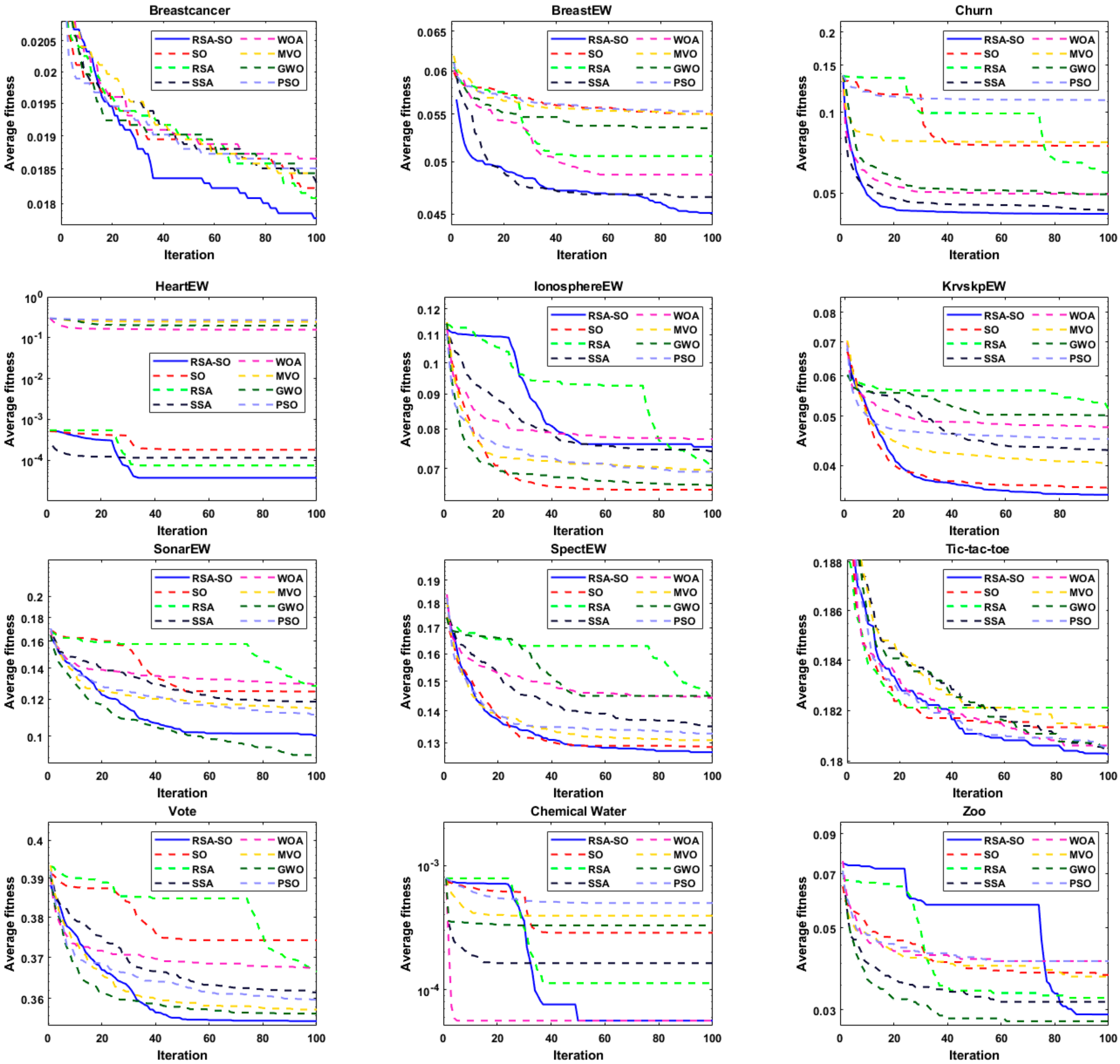

Figure 4. Convergence curves of the RSA-SO and the other methods.

4.4.1. Pressure Vessel Design (PVD)

The optimal design of a PVD aims to reduce the total of a pressure vessel constrained by material, shaping, and welding costs [30]. The PVD problem consists of four variables, as given in Figure 5: T_s denotes the thickness of the shell, T_h presents the thickness of the head, R is the inner radius, and L provides the length of the cylindrical section of the vessel. The objective function of this problem can be written as:

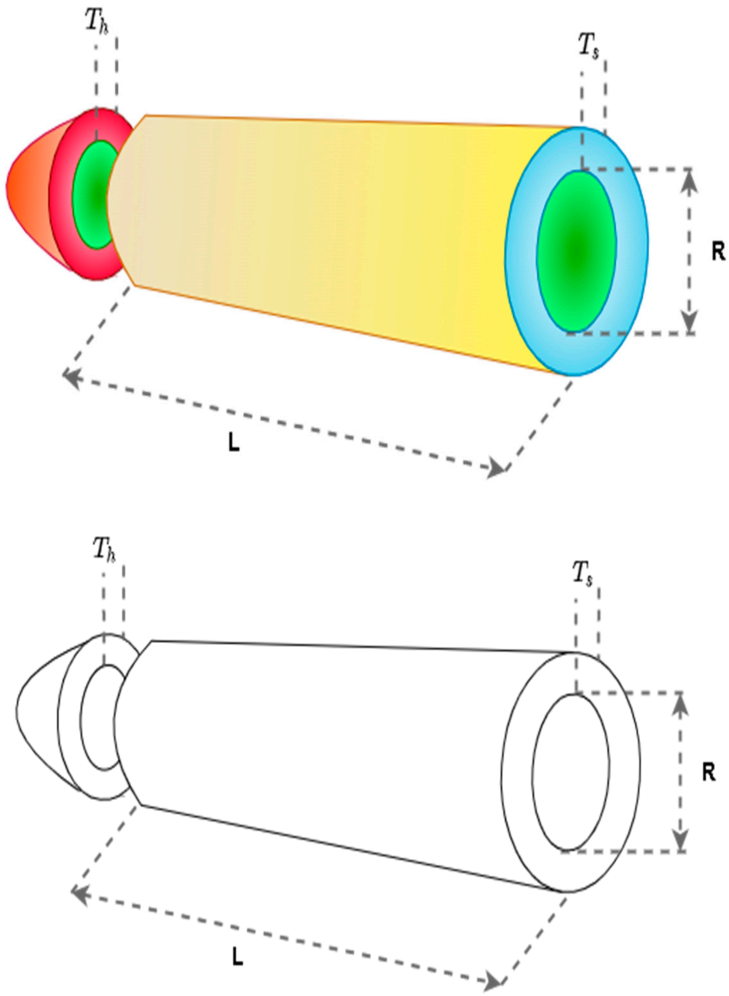

Figure 5. The PVD problem.

Minimize:

$$f(x) = 0.6224x_1x_2, x_3 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \tag{22}$$

Subject to:

$$\begin{gathered} g_1(x) = -x_1 + 0.0193x_3 \leq 0, \\ g_2(x) = -x_3 + 0.00954x_3 \leq 0, \\ g_3(x) = -\pi x_3^2x_4 - \tfrac{4}{3}\pi x_3^3 + 1,296,000 \ \leq 0, \\ g_4(x) = x_4 - 240 \leq 0, \end{gathered} \tag{23}$$

Variable range $(0 \leq x_i \leq 100,\ i = 1.2)$ and $(10 \leq x_i \leq 200,\ i = 3.4)$.

Table 7 lists the results obtained by the RSA-SO for the PVD problem and compares it with the other methods. As listed in Table 7, the suggested RSA-SO provides a lower cost than the PSO, GWO, MVO, WOA, SSA, RSA, and SO methods, and therefore, RSA-SO is suggested as a helpful method for the PVD problem. GWO placed second, MVO and SO placed third and fourth, and RSA placed last for the PVD problem.

Table 7. Results of RSA-SO and other methods for solving the PVD problem.

Method	Best Values for Variables				Best Cost
	T_s	T_h	R	L	
PSO	1.0000	0.0000	1.0000	1.0000	2758.9974
GWO	1.2591	0.0000	65.2298	10.0000	2613.1828
MVO	1.2614	0.0000	65.2280	10.1553	2630.2904
WOA	1.2679	0.0000	65.6966	13.7572	2878.7608
SSA	1.2738	0.0000	64.9012	11.4029	2734.5819
RSA	1.0000	0.0000	1.0000	1.0000	4277.1962
SO	1.2667	0.0000	65.4471	10.0000	2650.2554
RSA-SO	1.2588	0.0000	65.2252	10.0000	2611.9240

4.4.2. Cantilever Beam Design (CBD)

Figure 6 illustrates the design of the CBD problem. The problem tries to minimize the total weight, and this problem has five parameters: x_1, x_2, x_3, x_4, and x_5 [31]. The objective function of the CBD problem can be mathematically presented as follows:

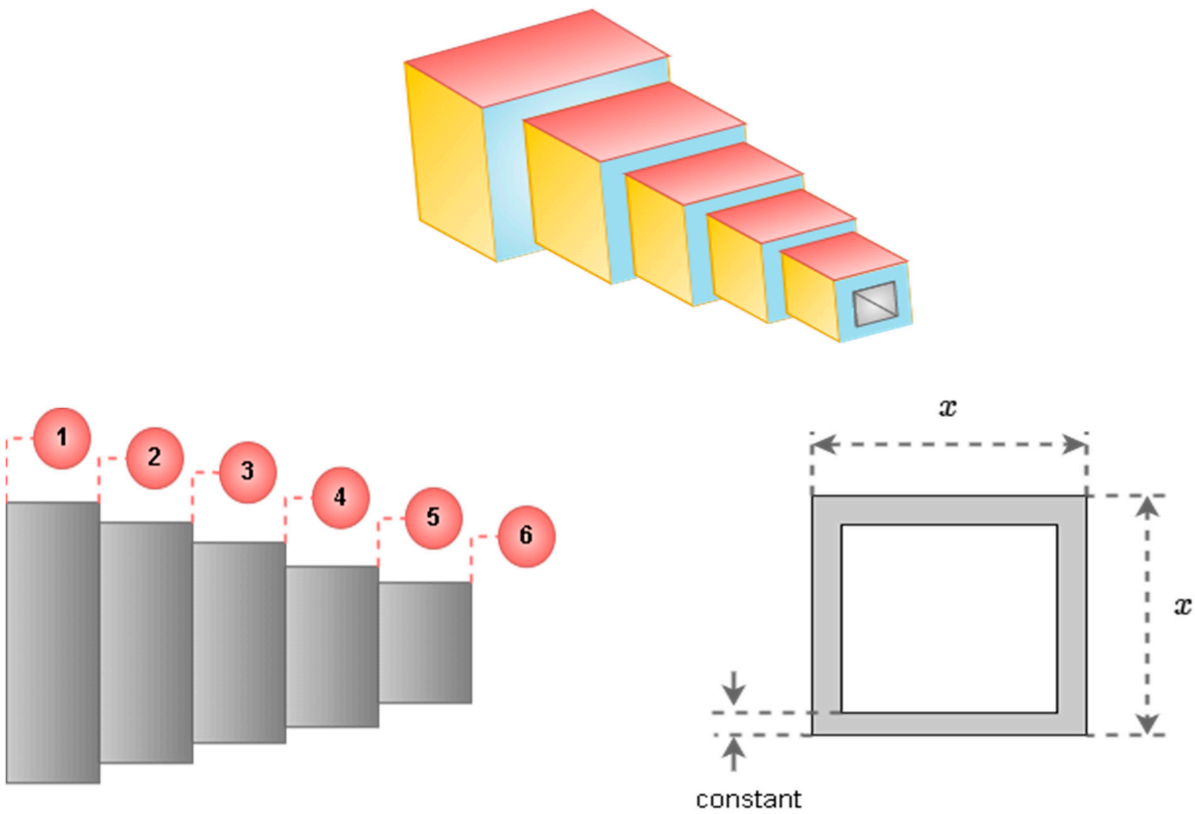

Figure 6. The CBD problem.

Minimize:

$$f(x) = 0.6224(\ x_1 x_2,\ x_3, x_4 x_5), \tag{24}$$

Subject to:

$$g(x) = \frac{60}{x_1^3} + \frac{27}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0 \tag{25}$$

In Table 8, the performance results of the RSA-SO for the CBD engineering problem are given when it is compared with other MH methods. As per Table 8, the best weight obtained by RSA-SO is the smallest compared to the other methods. MVO, WOA, and SO place second, third, and fourth, respectively, while SSA and RSA are in last place.

Table 8. Results of RSA-SO and other methods for solving the CBD problem.

Method	Best Values for Variables					Best Weight
	x_1	x_2	x_3	x_4	x_5	
PSO	1.0000	1.0000	1.0000	1.0000	1.0000	13.6384
GWO	5.5091	5.0942	4.5572	3.6607	2.2053	13.0869
MVO	5.9006	4.8694	4.4550	3.4898	2.1957	13.0146
WOA	5.9583	4.9565	4.4321	3.3923	2.1759	13.0176
SSA	6.3791	3.9871	8.6664	3.6680	1.7987	15.2484
RSA	1.0000	1.0000	1.0000	1.0000	1.0000	15.7689
SO	5.9832	4.7939	4.6247	3.4697	2.0584	13.0268
RSA-SO	5.9481	4.8974	4.4228	3.5007	2.1396	13.0135

Based on the previous results and discussion, the developed RSA-SO has a high ability to explore the feasible region which contains the optimal solution. However, the time complexity of RSA-SO still needs more improvements, especially when applied to handle high-dimensional data.

5. Conclusions and Future Works

FS is one of the key factors in improving the classifier capability in classification problems. In this paper, an FS approach based on RSA and SO, named RSA-SO, is presented. The introduced RSA-SO approach employs both RSA and SO in a parallel mechanism to tackle the problem of FS. We tested the RSA-SO approach on twelve different public datasets taken from UCI and two engineering problems. RSA-SO's capability was evaluated using a set of evaluation measures and compared with some recently reported MH methods for FS, including SO, RSA, SSA, WOA, MVO, GWO, and PSO. The results verify that RSA-SO has a comparative performance to other MH methods for FS, and it can provide practical and accurate solutions for two engineering optimization problems. For future work, RSA-SO will be applied to address other problems in different fields, such as sentiment analysis, Big Data, smart cities, and other practical engineering problems.

Author Contributions: Conceptualization, I.A.-S., P.H.K. and M.A.E.; methodology, I.A.-S. and P.H.K.; software, I.A.-S.; validation, I.A.-S. and P.H.K.; formal analysis, I.A.-S., P.H.K., S.A., S.D. and M.A.E.; resources, I.A.-S. and P.H.K.; writing—original draft preparation, I.A.-S. and P.H.K.; writing—review and editing, I.A.-S., P.H.K., S.A., S.D., B.E. and M.A.E.; funding acquisition, S.A. All authors have read and agreed to the published version of the manuscript.

Funding: Princess Nourah bint Abdulrahman Researchers Supporting Project number (PNURSP2022R197), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Data Availability Statement: The data available upon request from authors.

Acknowledgments: Princess Nourah bint Abdulrahman Researchers Supporting Project number (PNURSP2022R197), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Conflicts of Interest: All authors declare that they have no conflict of interest.

Nomenclature

Acronyms	
FS	Feature Selection
MH	Meta-Heuristic
OFS	Optimal Features Subset
RSA	Reptile Search Algorithm
SO	Snake Optimizer
Symbols	
$x_{i,j}$	ith candidate solution for jth feature dimension
N	Number of candidate solutions
M	Feature dimension
G	Total number of iterations for MH method
f_i	Fitness value of ith candidate solution
$n_{i,j}$	Hunting operator for the jth feature in the ith solution in RSA
x_i^{RSA}	ith candidate solution vector for RSA
x_i^{SO}	ith candidate solution vector for SO

References

1. Han, J.; Pei, J.; Kamber, M. *Data Mining: Concepts and Techniques*; Elsevier: Amsterdam, The Netherlands, 2011.
2. Crone, S.F.; Lessmann, S.; Stahlbock, R. The impact of preprocessing on data mining: An evaluation of classifier sensitivity in direct marketing. *Eur. J. Oper. Res.* **2006**, *173*, 781–800. [CrossRef]
3. Nguyen, B.H.; Xue, B.; Zhang, M. A survey on swarm intelligence approaches to feature selection in data mining. *Swarm Evol. Comput.* **2020**, *54*, 100663. [CrossRef]
4. Qaraad, M.; Amjad, S.; Manhrawy, I.I.; Fathi, H.; Hassan, B.A.; El Kafrawy, P. A hybrid feature selection optimization model for high dimension data classification. *IEEE Access* **2021**, *9*, 42884–42895. [CrossRef]
5. Sawalha, R.; Doush, I.A. Face recognition using harmony search-based selected features. *Int. J. Hybrid Inf. Technol.* **2012**, *5*, 1–16.
6. Shang, W.; Huang, H.; Zhu, H.; Lin, Y.; Qu, Y.; Wang, Z. A novel feature selection algorithm for text categorization. *Expert Syst. Appl.* **2007**, *33*, 1–5. [CrossRef]
7. Boutemedjet, S.; Bouguila, N.; Ziou, D. A hybrid feature extraction selection approach for high-dimensional non-Gaussian data clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *31*, 1429–1443. [CrossRef]
8. Vo, T.T.; Liu, M.K.; Tran, M.Q. Identification of Milling Stability by using Signal Analysis and Machine Learning Techniques. *Intern. J. Robot.* **2021**, *4*, 30–39.
9. Tarhan, İ.; Oğuz, C. Generalized order acceptance and scheduling problem with batch delivery: Models and metaheuristics. *Comput. Oper. Res.* **2021**, *134*, 105414. [CrossRef]
10. Ikeda, S.; Nagai, T. A novel optimization method combining metaheuristics and machine learning for daily optimal operations in building energy and storage systems. *Appl. Energy* **2021**, *289*, 116716. [CrossRef]
11. Band, S.S.; Ardabili, S.; Danesh, A.S.; Mansor, Z.; AlShourbaji, I.; Mosavi, A. Colonial competitive evolutionary Rao algorithm for optimal engineering design. *Alex. Eng. J.* **2022**, *61*, 11537–11563. [CrossRef]
12. Zelinka, I. A survey on evolutionary algorithms dynamics and its complexity–Mutual relations, past, present and future. *Swarm Evol. Comput.* **2015**, *25*, 2–14. [CrossRef]
13. Braik, M.; Sheta, A.; Al-Hiary, H. A novel meta-heuristic search algorithm for solving optimization problems: Capuchin search algorithm. *Neural Comput. Appl.* **2021**, *33*, 2515–2547. [CrossRef]
14. Alam, S.; Dobbie, G.; Rehman, S.U. Analysis of particle swarm optimization based hierarchical data clustering approaches. *Swarm Evol. Comput.* **2015**, *25*, 36–51. [CrossRef]
15. Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **2016**, *27*, 495–513. [CrossRef]
16. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November 1995.
17. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
18. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
19. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [CrossRef]
20. Heidari, A.A.; Aljarah, I.; Faris, H.; Chen, H.; Luo, J.; Mirjalili, S. An enhanced associative learning-based exploratory whale optimizer for global optimization. *Neural Comput. Appl.* **2020**, *32*, 5185–5211. [CrossRef]
21. Moradi, P.; Gholampour, M. A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy. *Appl. Soft Comput.* **2016**, *43*, 117–130. [CrossRef]
22. Al-Shourbaji, I.; Helian, N.; Sun, Y.; Alshathri, S.; Elaziz, M.A. Boosting Ant Colony Optimization with Reptile Search Algorithm for Churn Prediction. *Mathematics* **2022**, *10*, 1031. [CrossRef]

23. Abualigah, L.; Abd Elaziz, M.; Sumari, P.; Geem, Z.W.; Gandomi, A.H. Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer. *Expert Syst. Appl.* **2022**, *191*, 116158. [CrossRef]
24. Hashim, F.A.; Hussien, A.G. Snake Optimizer: A novel meta-heuristic optimization algorithm. *Knowl.-Based Syst.* **2022**, *242*, 108320. [CrossRef]
25. Wang, X.H.; Zhang, Y.; Sun, X.Y.; Wang, Y.L.; Du, C.H. Multi-objective feature selection based on artificial bee colony: An acceleration approach with variable sample size. *Appl. Soft Comput.* **2020**, *88*, 106041. [CrossRef]
26. Ewees, A.A.; El Aziz, M.A.; Hassanien, A.E. Chaotic multi-verse optimizer-based feature selection. *Neural Comput. Appl.* **2019**, *31*, 991–1006. [CrossRef]
27. Ibrahim, R.A.; Elaziz, M.A.; Ewees, A.A.; El-Abd, M.; Lu, S. New feature selection paradigm based on hyper-heuristic technique. *Appl. Math. Model.* **2021**, *98*, 14–37. [CrossRef]
28. Song, X.F.; Zhang, Y.; Gong, D.W.; Gao, X.Z. A fast hybrid feature selection based on correlation-guided clustering and particle swarm optimization for high-dimensional data. *IEEE Trans. Cybern.* **2021**, 1–14. [CrossRef]
29. Arora, S.; Anand, P. Binary butterfly optimization approaches for feature selection. *Expert Syst. Appl.* **2019**, *116*, 147–160. [CrossRef]
30. Askarzadeh, A. A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Comput. Struct.* **2016**, *169*, 1–12. [CrossRef]
31. Gafar, M.G.; El-Sehiemy, R.A.; Sarhan, S. A Hybrid Fuzzy-Crow Optimizer for Unconstrained and Constrained Engineering Design Problems. *Hum. Cent. Comput. Inf. Sci.* **2022**, *12*, 1–24.