electronics

MDPI

Article

A Novel Metaheuristic Moss-Rose-Inspired Algorithm with Engineering Applications

Hussein M. Hathal [1], Ramzy S. Ali [2,*] and Abdulkareem S. Abdullah [2]

[1] Electrical Engineering Department, College of Engineering, Mustansiriyah University, Baghdad 10047, Iraq; hussein.m.hathal@uomustansiriyah.edu.iq

[2] Electrical Engineering Department, College of Engineering, University of Basrah, Basra 61004, Iraq; abdulkareem.abdullah@uobasrah.edu.iq

* Correspondence: ramzy.ali@uobasrah.edu.iq

Abstract: Every day, a moss rose generates new flowers with variable diameters. Two flowering mechanisms are controlled by exposure to sunlight, namely, a variable concentration of florigen based on photoreceptors called phytochromes, and the biological clock, which is responsible for the changing diameters of the plant flowers at night and some hours during the day. By explaining and idealizing the flowering mechanisms of the moss rose in nature, a new sort of nature-inspired optimization algorithm called the moss rose optimization algorithm (MROA) was proposed in this study. The MROA was benchmarked using three methods. First, 18 benchmark functions were utilized to evaluate the effectiveness of the MROA. Second, the MROA was used for planning a smart antenna system (SA) as an online solution to find unknown weights. Third, the MROA was used to find the optimal dimensions for a microstrip antenna for the frequency (2.4 GHz) as an offline solution. The MROA was compared with other algorithms. The results show the capacities and proficiencies of the proposed algorithm regarding finding the ideal solutions. The promising arrangements for smart antenna identification and microstrip antenna design highlight the importance of this algorithm for resolving current issues with unknown fields of investigation.

Keywords: moss rose optimization algorithm; metaheuristic algorithms; benchmark functions; smart antenna system identification; microstrip patch antenna

check for updates

Citation: Hathal, H.M.; Ali, R.S.; Abdullah, A.S. A Novel Metaheuristic Moss-Rose-Inspired Algorithm with Engineering Applications. *Electronics* **2021**, *10*, 1877. https://doi.org/10.3390/electronics10161877

Academic Editor: Dimitra I. Kaklamani

Received: 13 July 2021
Accepted: 30 July 2021
Published: 4 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

In many applications, such as in engineering, businesses and industrial designs, optimization is extremely important. Many researchers ask a common question: there are so many optimization algorithms, so what is the best?

It is a simple question, but unfortunately, there is no simple answer. We cannot answer this question simply for several reasons. One reason is that the complexity and diversity of problems in the real world often make it easier to solve some problems, whereas others can be extremely difficult. Consequently, a single method is unlikely to solve all types of problems. Another reason is due to the so-called no free lunch (NFL) theorem, which reads that no universal algorithm exists for all problems [1].

This theorem states that: in the search for an extremity of an objective function, if any algorithm A surpasses another algorithm B, then algorithm B surpasses other objective functions. In general, the NFL theorem applies to the scenario of either deterministic or stochastic parameters, where the objective or cost function can be defined using a set of continuous (or discrete or mixed) parameters [1].

The aim of every optimization solution involves at least some of the following: to reduce energy and costs and to maximize profit, output, performance and efficiency. Metaheuristic algorithms are currently turning out to be incredible techniques for solving numerous complex issues using streamlining [2–13]. By far, most heuristic and metaheuristic algorithms were inspired by the behavior of biological system frameworks or potential frameworks in nature. There are several different types of algorithms; some of the popular and modern algorithms are discussed below.

The beginning of the study of metaheuristic algorithms was developed in the thesis of [2] in 1992, where he proposed an algorithm that was inspired by an ant colony (AC). This search technique was inspired by the swarm intelligence of social ants using a pheromone as a chemical messenger.

In 1995, ref. [3] proposed an algorithm called particle swarm optimization (PSO), which was created based on the swarm behavior of birds and fish. The multiple-particle swarm moves in the search space, starting from some initial random guess. The swarm gives information about the current best solution. Then, to focus on the high-quality solutions, they share the global best solution.

In 2006, ref. [4] presented a novel numerical algorithm for stochastic optimization that is based on weeds. In a simple but successful optimization algorithm known as invasive weed optimization (IWO), the robustness, adaptation and randomness of the colonizing weeds is tested.

In 2009, ref. [5] proposed an algorithm called the firefly algorithm (FA), which is based on the behavior of firefly flashlights in the summer night's sky. Then, they provided a comparison study between the FA and PSO algorithms.

In 2010, ref. [6] proposed an algorithm called the cuckoo search algorithm (CS), which is based on some cuckoo species that engage in brood parasitism.

In 2010, ref. [7] proposed a novel nature-inspired metaheuristic algorithm called the bat algorithm (BA). This algorithm is based on microbats' behavior, which involves using echolocation for finding certain directions.

In 2012, ref. [8] proposed an algorithm called the flower pollination algorithm (FPA). Using another domain for metaheuristic inspiration, this algorithm is based on the pollination behavior of plants and types of pollination spreads in fields.

In 2013, ref. [9] presented research on complex system modeling and calculations using a novel biologically inspired approach known as the root growth algorithm (RG). This general model of optimization gleaned ideas from the behaviors of root growth in soil.

In 2016, ref. [10] proposed a novel algorithm for nature-based metaheuristic optimization named the whale optimization algorithm (WOA), which mimics humpback whales' social behavior. The bubble-net hunting strategy was the inspiration of the algorithm.

The bio-inspired computational technique for geometrically optimized joints of a compact coplanar waveguide (CPW)-fueled microstrip antenna with a defective ground structure was introduced in 2017 by [11] and is known as the adaptive bacterial foraging optimization (ABFO). The ABFO was compared with the original (BFO) technique, PSO, the invasive weed optimization technique, and the artificial bee colony (ABC) to check its adequacy.

In 2018, ref. [12] proposed a modern model for optimization that was inspired by nature called the squirrel search algorithm (SSA). This optimization algorithm imitates the southern flying squirrel's complex foraging behavior and its effective method of transport known as gliding. The proposed algorithm mathematically modeled this behavior to realize the optimization process. In terms of the efficiency of the proposed SSA, the statistical analysis, convergence rate analysis, Wilcoxon test and ANOVA were evaluated with respect to the classic and modern CEC 2014 benchmarks. The performance of the SSA over other popular optimization techniques with regard to the optimization accuracy and convergence rate was demonstrated in an exhaustive comparative analysis.

In 2019, ref. [13] proposed a new modified algorithm with a high calculation speed and simplified camel-based structural optimization (modified CA). The results showed that the modified camel algorithm is preferable when compared with particle swarm optimization (PSO) and the crow searching algorithm (CSA).

In 2020, the smart flower optimization algorithm (SFOA) was proposed in [14] to provide a new kind of nature-inspired optimization algorithm. There were two modes for the proposed algorithm: sunny and cloudy or snowy, depending on weather conditions.

For the testing of SFOA's efficiency using statistical analysis and Wilcoxon's test, a collection of 15 benchmarking features in the CEC 2015 was used. For the design of a system of adaptive IIR to adapt to an unknown system, SFOA was used.

Each of these algorithms has specific weaknesses and focal points. For example, simulations annealing can almost guarantee that the ideal arrangement is found if the cooling process is moderate and the simulation is sufficiently long [15].

In this study, another metaheuristic technique was proposed, known as the moss rose optimization algorithm (MROA), which is based on the flowering of this type of plant. In addition to other factors, such as day length and daytime temperature, the ability of the moss rose to flower at some time of day depends mainly on light factors. In the rest of the paper, the moss rose optimization algorithm for optimizing the plant's floral diameters is proposed. A brief overview of the algorithm is then given, and a comparison with other algorithms is provided to show that the proposed algorithm works correctly.

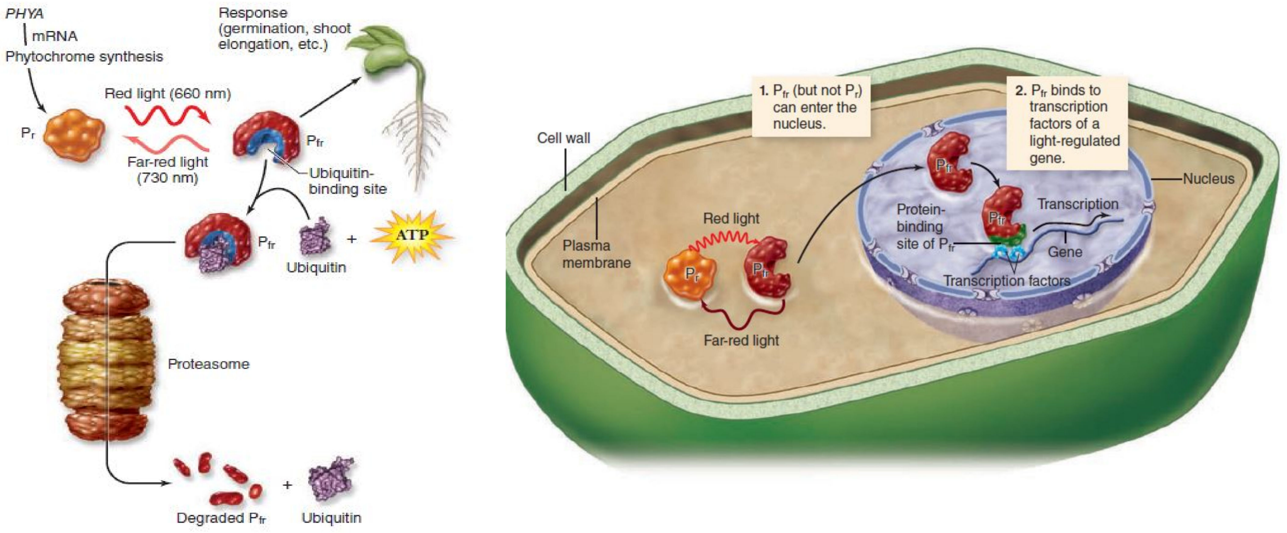
2. Moss Rose Optimization Algorithm

In this part, the proposed plant algorithm's inspiration is considered. Then, the mathematical model of the algorithm is presented.

2.1. Inspiration

The inspiration for the proposed approach was the flowering behavior of the moss rose. The scientific name of this plant is Portulaca grandiflora Hook. This plant is classified as a long-day plant. The most important factor that activates the flowering is exposure to red light. The light interacts with photoreceptors (phytochromes) to make a special protein called florigen.

A phytochrome is a multiform pigment that is capable of absorbing red light. These pigments absorb light in a very narrow and specific spectral range. The phytochromes are the first step in providing information about the red phytochrome (Pr) and far-red phytochrome (Pfr) levels in a signaling system, leading to developmental changes in gene expression. There are two convertible forms of the phytochrome molecule: first, Pr absorbs red light at a 660 nm wavelength, and second, Pfr absorbs 730 nm long-red light. The light from the Sun contains more red light than far-red light. Pr is biologically inactive; when red photons are present, it is transformed into the active form, Pfr. When far-red photons are available, Pfr is transformed back into Pr. In other words, if Pfr is present, there will be biological reactions affecting the phytochromes. The reactions cannot occur if most of the Pfr has been replaced by Pr [16], as shown in Figure 1a.



(**a**) (**b**)

Figure 1. Moss rose flowering mechanism: (**a**) phytochrome generation and **(b)** florigen generation in the cell.

The phytochromes are located in the cytoplasm but enter the core to allow light-response genes to be transcribed. It can pass into the nucleus when Pr is converted to Pfr. Once inside the nucleus, Pfr binds to other proteins to form a transcription complex, which leads to the production of light-controlled genes. This operation is shown in Figure 1b [16].

From the information above regarding the biological mechanism, we see that light is the flowering signal of moss roses. Light activates photoreceptors and triggers signal cascades in plant cells of apical or lateral meristems [17].

2.2. Mathematical Model

The moss rose optimization is a population-based optimization algorithm. For the proposed algorithm, the search space can be modified based on the mechanisms of flowering in the plant. In this algorithm, it is assumed that each moss rose has the ability to produce flowers in a dimensional search space.

Since the moss rose algorithm depends on the population, a population of a flowering cluster can be represented in a matrix as follows:

$$Fd = \begin{bmatrix} fd_{1,1} & fd_{1,2} & \cdots & \cdots & fd_{1,Dim} \\ fd_{2,1} & fd_{2,2} & \cdots & \cdots & fd_{2,Dim} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ fd_{M,1} & fd_{M,2} & \cdots & \cdots & fd_{M,Dim} \end{bmatrix} \quad (1)$$

where M denotes the number of flowers in a plant and Dim denotes the number of variables (dimension). In the proposed algorithm, the diameter of each flower of a moss rose leads to a random solution to the optimization problem. Each flower has a fitness value that depends on the fitness function value of the optimization problem that represents the diameter of its flower. A better fitness value represents the diameter of a larger flower. New flower diameters (solutions) allow the algorithm to predict and prepare to complete their flowering during the same day in the decision space based on internal mechanisms. The definitions of the properties of the MROA are represented in Table 1.

Table 1. The properties of the MROA.

Decision Variable	→	Moss Rose's Flowering in a Day
Initial solution	→	Randomly generated lighting of a moss rose
Old solution	→	Old flower diameter of a moss rose
New solution	→	New flowering of a moss rose
Best solution	→	Best flower diameter of a moss rose
Objective function	→	Florigen amount, which depends on the light (phytochrome tincture) and biological clock (red photon spectrum)
Process of generating a new solution	→	Flowering mechanisms of a moss rose

The mathematical model that was used to simulate the flowering mechanisms of a moss rose is presented in the following equations:

1. Create random variables that represent the flowering diameters:

$$fd(i) = [max - min] * rand\ fd + min \quad (2)$$

where fd—the diameters of roses between max and min values.

2. Generate the flowering age parameter, which depends on the current diameter and the maximum diameter that the flower will reach. The equation for the *age* is

$$f\_age = (fd_{max} * TH) + fd \quad (3)$$

where fd_{max}—maximum flower diameter and TH—total hours in a day.

3. Generate the phytochrome parameter, which depends on the max flowering diameter, the number of hours the flowers are open, a random time in the morning and the minimum flowering diameter:

$$phytochrome = fd_{max} * e^{-(SF*clock-BF)^{oh}} + fd_{min} \tag{4}$$

where *BF*—biological opening factor (controls the clock time of flowering; the value is 3), *SF*—scaling factor (to ensure the maximum reaction of the rose to light effect; the value is 2.7), *oh*—maximum number of hours the flowers are open and fd_{min}—minimum flower diameter.

4. Calculate the new *fd* according to the following equation:

$$fd_{new} = fd_{old} + phytochrome * f_{age} * \mathrm{Pr} * (fd_{best} - fd_{old}) \tag{5}$$

where Pr—red photon wavelength (nm) and fd_{best}—best flowering diameter, which is calculated based on the fitness function.

Any flower on the plant can update its diameter according to the random sunlight receptors in response to the composition of the 'phytochrome' in the apical meristem without the other. Therefore, the same concept can be extended with a dimensional search space by changing the exponential morning clock function 'clock', which is chosen randomly between 7 a.m. and 3 p.m.

In order to have a clear visualization of the work of the algorithm, the basic work of the two equations mentioned in steps 2 and 3 involves adding the values that control the improvement of the random variable. As for step 4, this is the basic equation for updating the randomly imposed value and it contains the values of the variables that were generated in the previous two steps.

The moss rose will close its flowers during the night and for several hours of the day, and its flowers will bloom during the rest of the day. This cyclic prototype permits a moss rose to be repositioned around another solution. This can guarantee that the intensification of the space is characterized by two arrangements. To diversify the inquiry space, the flowers should have the option to look outside the local space of the best arrangements they are comparing. This can be achieved by changing the scope of the phytochrome during a 24 h day/night cycle, as shown in Figure 2.

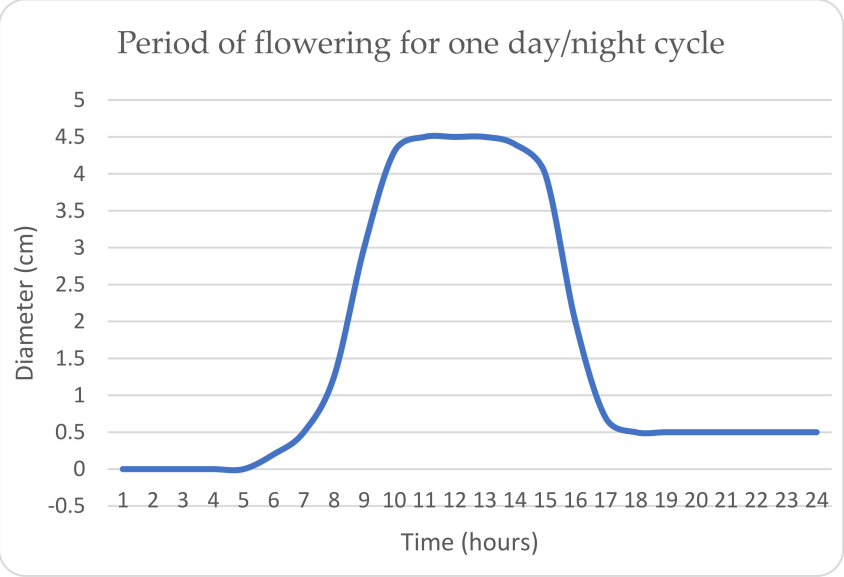
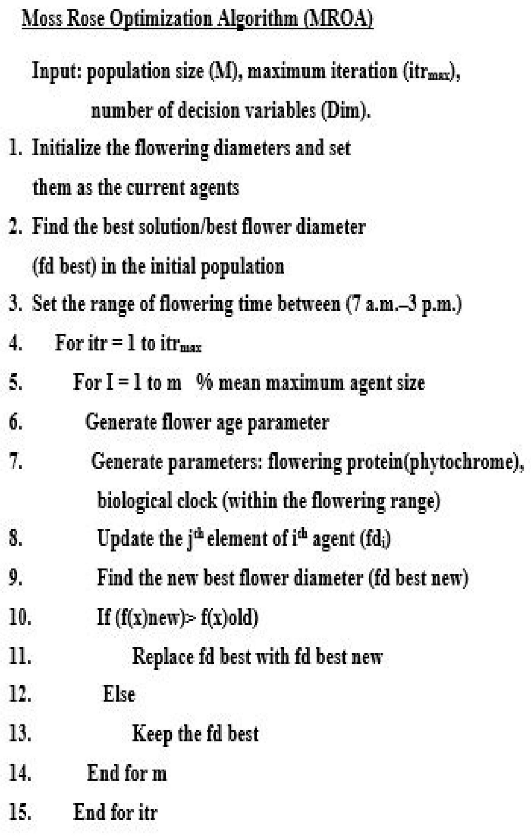


Figure 2. Period of flowering for one day/night cycle.

In stochastic algorithms, a calculation should have the option to adjust between intensification and diversification stages to compute the promising regions of the pursuit space and eventually join to produce the global optimum. For adjusting the intensification and diversification, the flowering diameter of the roses was decreased during successive iterations. It is shown in Figures 1 and 2 that the MROA investigates the pursuit space when the phytochrome is enacted and the organic clock of the moss rose works regularly.

5. The MROA finalizes the optimization process when evaluating the maximum number of fitness functions or obtaining the accuracy of the global optimum. The pseudocode of the MROA is illustrated in Figure 3. The flowchart of the MROA is shown in Figure 4



```
Moss Rose Optimization Algorithm (MROA)

Input: population size (M), maximum iteration ($itr_{max}$),
       number of decision variables (Dim).
Initialize the flowering diameters and set
them as the current agents
Find the best solution/best flower diameter
(fd best) in the initial population
Set the range of flowering time between (7 a.m.–3 p.m.)
  For itr = 1 to $itr_{max}$
    For I = 1 to m   % mean maximum agent size
      Generate flower age parameter
      Generate parameters: flowering protein(phytochrome),
       biological clock (within the flowering range)
       Update the $j^{th}$ element of $i^{th}$ agent ($fd_i$)
       Find the new best flower diameter (fd best new)
       If (f(x)new)> f(x)old)
           Replace fd best with fd best new
       Else
           Keep the fd best
      End for m
    End for itr
```

Figure 3. General pseudocode of the MROA.

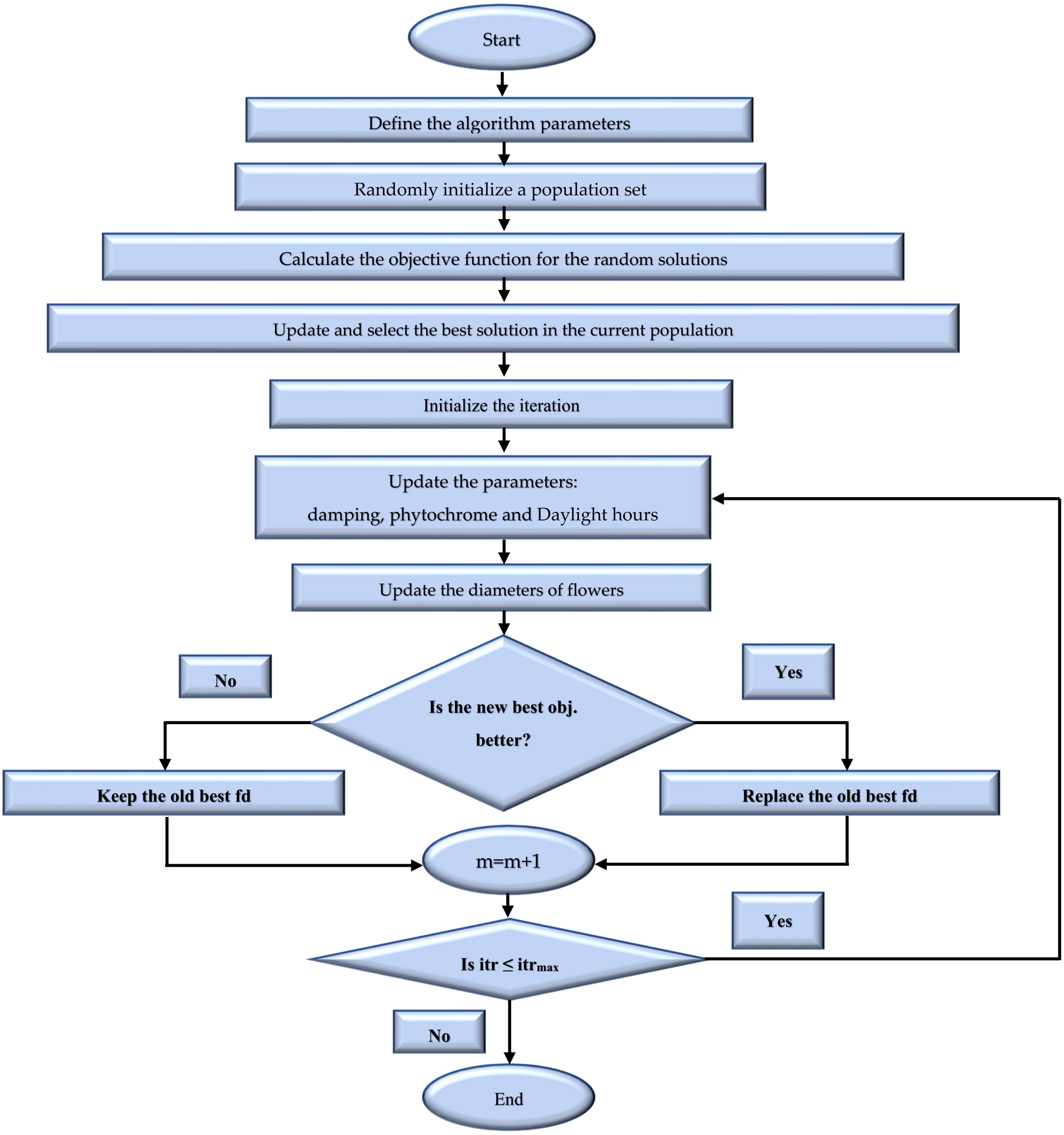

Start
Define the algorithm parameters
Randomly initialize a population set
Calculate the objective function for the random solutions
Update and select the best solution in the current population
Initialize the iteration
Update the parameters:
damping, phytochrome and Daylight hours
Update the diameters of flowers
Is the new best obj. better?
No
Yes
Keep the old best fd
Replace the old best fd
m=m+1
Yes
Is itr ≤ itrmax
No
End

Figure 4. Flowchart of the proposed MROA.

3. Computational Results of Benchmark Functions

To check the efficiency of the proposed MROA, a total of 18 benchmark functions were implemented. Two types of functions composed these benchmarks:

(a) Unimodal functions for benchmarking;
(b) Functions for multimodal benchmarks.

The experiments were carried out using the R2019a version of MATLAB software and Windows 10 Pro N on a laptop with an i7 Core 2.4 GHz processor, a 256 Gb SSD and 8 Gb RAM.

Table 2 lists the global/local benchmarking functions' names, measurements, attributes and the associated values. Details of the 18 benchmark functions are given in [18,19].

Table 2. Characteristics of benchmark functions.

Type	Name	Specifications	Search Range	Global Minimum at
Unimodal	Acklay2 Function	Continuous, differentiable, non-separable, non-scalable	$-32 \leq xi \leq 32.$	$f(x*) = -200$ at $x* = (0, 0)$
	Beale Function	Continuous, differentiable, non-separable, non-scalable	$-4.5 \leq xi \leq 4.5$	$f(x*) = 0$ at $x* = (3, 0.5)$
	Dixon and Price Function	Continuous, differentiable, non-separable, scalable	$-10 \leq xi \leq 10$	$f(x*) = 0$ at $x* = (2\ (2i–2/2i))$
	Leon Function	Continuous, differentiable, non-separable, non-scalable	$-1.2 \leq xi \leq 1.2$	$f(x*) = 0$ at $x* = (1, 1)$,
	Matyas Function	Continuous, differentiable, non-separable, non-scalable	$-10 \leq xi \leq 10$	$f(x*) = 0$ at $x* = (0, 0)$
	Powell Sum Function	Continuous, differentiable, separable, scalable	$-1 \leq xi \leq 1$	$f(x*) = 0$ at $x* = (0, 0)$
	Elliptic Function	Continuous, differentiable, non-separable, non-scalable	$-100 \leq xi \leq 100$	$f(x*) = 0$ at $x* = (0, 0)$
	Rosenbrock Function	Continuous, differentiable, non-separable, scalable	$-30 \leq xi \leq 30$	$f(x*) = 0$ at $x* = (1, \ldots, 1)$
	Schwefel 2.22 Function	Continuous, differentiable, non-separable, scalable	$-100 \leq xi \leq 100$	$f(x*) = 0$ at $x* = (0, 0)$
	Step 2 Function	Discontinuous, non-differentiable, separable, scalable	$-100 \leq xi \leq 100$	$f(x*) = 0$ at $x* = (0.5, \ldots, 0.5)$
	Sum Square Function	Continuous, differentiable, non-separable, non-scalable	$-10 \leq xi \leq 10$	$f(x*) = 0$ at $x* = (0, 0)$
Multimodal	Colville Function	Continuous, differentiable, non-separable, non-scalable	$-10 \leq xi \leq 10$	$f(x*) = 0$ at $x* = (1, \ldots, 1)$
	Easom Function	Continuous, differentiable, separable, non-scalable	$-100 \leq xi \leq 100$	$f(x*) = -1$ at $x^* = (\pi, \ldots, \pi)$
	Quartic Function	Continuous, differentiable, separable, scalable	$-1.28 \leq xi \leq 1.28$	$f(x*) = 0$ at $x* = (0, \ldots, 0)$
	Quartic with Noise Function	Continuous, differentiable, separable, scalable	$-1.28 \leq xi \leq 1.28$	$f(x*) = 0$ at $x* = (0, \ldots, 0)$
	Schwefel 2.36 Function	Continuous, differentiable, separable, scalable	$0 \leq xi \leq 500$	$f(x*) = -3456$ at $x* = (12, \ldots, 12)$
	Griewank Function	Continuous, differentiable, non-separable, scalable	$-100 \leq xi \leq 100$	$f(x*) = 0$ at $x* = (0, \ldots, 0)$
	Schaffer 6 Function	Continuous, differentiable, non-separable, scalable	$-100 \leq xi \leq 100$	$f(x*) = 0$ at $x* = (0, \ldots, 0)$

Three well-known search algorithms, namely, the crow searching algorithm (CSA) [20], the modified camel algorithm (MCA) [17] and particle swarm optimization (PSO) [3], were used as tests of MROA's benchmarking functions. The maximum population size and the iteration values were set to 50 and 5000, respectively, for all algorithms to ensure that the comparison was fair. All algorithms in Table 3 had the default parameters. For the proposed MROA, the default parameter values of maximum diameter, minimum diameter, Pr and daylight hours were set to 4, 0.5, 660 nm and [7,15], respectively. Each algorithm was run 20 times for every function to evaluate the algorithms using the benchmark functions. The results of the benchmark functions are shown in Table 4. For each algorithm, the minimum value of the functions is called 'Min. Value', the average solution for a given function after 20 implementations is called 'Mean', the standard function deviation for each algorithm is 'Std. Dev.' and a bold number represents the best mean for each benchmark function for every single algorithm. The algorithms were ranked according to the smallest mean solution. It is easy to detect that MROA produced much better results than the compared algorithms based on the results of the mean values illustrated in Table 4.

It can be concluded from Table 4 that the MROA algorithm found the best solution (global optimum) for all benchmark functions. The multimodal benchmark functions illustrated the second part of the functions testing, which contained seven functions.

Given the difficulty of solving these functions, this means that the global searchability test is hard for any algorithm [19]. It is shown in Table 4 that the MROA ranked first for all functions (Colville Function, Easom Function, Quartic Function, Quartic with Noise Function, Schwefel 2.36 Function, Griewank Function and Schaffer 6 Function) based on the mean of each solutions. These results show that the MROA provided better results than the other algorithms.

As displayed above, the benchmark functions were classified into two parts. The first part (unimodal) contained eleven functions, where MROA had the best mean values for eight of the eleven unimodal functions, (bold numbers in the Table 4), the eight unimodal functions are: Acklay2 Function, Dixon and Price Function, Matyas Function, Powell Sum Function, Elliptic Function, Rosenbrock Function, Step 2 Function and Sum Square Function, but had the second-best mean values for three functions (Beale Function, Leon Function and Schwefel 2.22 Function).

Table 3. The parameters of compared algorithms.

Algorithms	Parameters	Value
MROA	Number of rose flowers for each plant	30
	Plant population	50
	Maximum number of iterations	5000
	Red photon wavelength	660 nm
	Maximum flowering diameter	4
	Minimum flowering diameter	0.5
CSA	Number of steps for each flight	30
	Flock population	50
	Awareness probability	0.1
	Flight length (fl)	2
	Maximum number of iterations	5000
MCA	Number of decision variables	30
	Camel caravan population	50
	Maximum number of iterations	5000
	Visibility	0.1
	Minimum temperature	30°
	Maximum temperature	60°
PSO	Number of decision variables	30
	Population	50
	Maximum number of iterations	5000
	Cognitive constant (*c*1)	1
	Social constant (*c*2)	1
	Inertia weight	0.4–0.9

Table 4. Compared algorithms' results for the benchmark functions.

Function	Results	MROA	PSO	MCA	CSA
Acklay2	**Min. Value**	−200	−199.42	−198.34	−200.00
	Mean	**−200**	−194.2915	−194.8375	−199.9995
	St. Dev.	0	2.991608043	2.797878661	0.002236068
	Rank	1	3	4	2
Beale	**Min. Value**	3.2471×10^{-15}	0	0.00012672	2.548×10^{-17}
	Mean	6.0773×10^{-5}	0.049739907	0.097696486	$\mathbf{6.37883} \times 10^{-13}$
	St. Dev.	0.000177395	0.100935338	0.086289362	1.89474×10^{-12}
	Rank	2	3	4	1
Dixon & Price	**Min. Value**	0.018996	0.022127	0.00049427	0.019022
	Mean	**0.1036305**	0.20409175	0.148765864	0.1661664
	St. Dev.	0.08940493	0.170401316	0.136410879	0.13953285
	Rank	1	4	2	3

Table 4. *Cont.*

Function	Results	MROA	PSO	MCA	CSA
Leon	**Min. Value**	2.3882×10^{-17}	2.1952×10^{-16}	2.4476×10^{-11}	7.8064×10^{-11}
	Mean	1.55727×10^{-9}	0.008226761	**1.30748** $\times 10^{-9}$	1.89626×10^{-8}
	St. Dev.	3.75185×10^{-9}	0.021144602	1.60812×10^{-9}	3.18738×10^{-8}
	Rank	2	4	1	3
Matyas	**Min. Value**	3.6864×10^{-17}	5.5254×10^{-18}	0.00032757	1.2634×10^{-10}
	Mean	**3.50209** $\times 10^{-10}$	0.000349969	0.005781294	1.5075×10^{-9}
	St. Dev.	1.05425×10^{-9}	0.001343955	0.004856965	1.85197×10^{-9}
	Rank	1	3	4	2
Powell Sum	**Min. Value**	2.6923×10^{-116}	4.0765×10^{-43}	5.1255×10^{-82}	4.0934×10^{-20}
	Mean	**3.38865** $\times 10^{-31}$	9.01383×10^{-7}	0.012883587	4.14734×10^{-16}
	St. Dev.	1.51545×10^{-30}	3.95512×10^{-6}	0.057616743	1.82849×10^{-15}
	Rank	1	3	4	2
Elliptic	**Min. Value**	1.01×10^{-8}	0.000024538	0.000096338	3.26000×10^{-6}
	Mean	**1.11779** $\times 10^{-6}$	0.001778024	0.045285091	3.43628×10^{-6}
	St. Dev.	1.61468×10^{-6}	0.004665153	0.078707248	3.52554×10^{-5}
	Rank	1	3	4	2
Rosenbrock	**Min. Value**	8.6376×10^{-14}	0	−79367000	−29.999
	Mean	**5.12256** $\times 10^{-6}$	0.000824868	−80249600	3
	St. Dev.	1.60557×10^{-5}	0.003229872	697280.5371	30.62496402
	Rank	1	2	4	3
Schwefel 2.22	**Min. Value**	2.46×10^{-8}	0	0.00084192	5.6813×10^{-7}
	Mean	8.25162×10^{-6}	0.0740245	0.007106103	**−3.32486** $\times 10^{-7}$
	St. Dev.	1.01709×10^{-5}	0.186976099	0.008900305	6.07854×10^{-6}
	Rank	2	4	3	1
Step 2	**Min. Value**	1.9125×10^{-14}	0	0.086092	0
	Mean	**0.00191227**	0.053905812	0.7723646	0.182425969
	St. Dev.	0.008507456	0.241070298	0.592366856	0.292870893
	Rank	1	2	4	3
Sum Square	**Min. Value**	9.6547×10^{-21}	0	4.9192×10^{-6}	2.8783×10^{-9}
	Mean	**3.75465** $\times 10^{-10}$	0.0072807	6.51515×10^{-5}	0.014581727
	St. Dev.	1.18746×10^{-9}	0.022974547	0.000121492	0.030715494
	Rank	1	3	2	4
Colville	**Min. Value**	1.9816×10^{-7}	0.000019144	10.403	0.000060625
	Mean	**0.00990933**	0.112682992	54.9052	0.064946773
	St. Dev.	0.013830272	0.191159812	31.80264295	0.164085831
	Rank	1	3	4	2
Easom	**Min. Value**	−1	−1	−0.34376	−1
	Mean	**−0.9999945**	−0.937501	−0.01718949	−0.99984
	St. Dev.	1.70062×10^{-05}	0.130961483	0.076866722	0.000493335
	Rank	1	3	4	2
Quartic	**Min. Value**	0	0.00094364	5.0625×10^{-24}	$1.7000000 \times 10^{-20}$
	Mean	**1.0326** $\times 10^{-141}$	0.006156197	5.59873×10^{-10}	1.81534×10^{-19}
	St. Dev.	4.6177×10^{-141}	0.006189817	2.00435×10^{-9}	7.70156×10^{-19}
	Rank	1	4	3	2
Quartic with Noise	**Min. Value**	6.6502×10^{-6}	0.000086509	0.00024593	0.0068233
	Mean	**0.000151943**	0.002846577	0.007795547	0.010313575
	St. Dev.	0.000129066	0.011295313	0.006333955	0.026795825
	Rank	1	2	3	4
Schwefel 2.36	**Min. Value**	−3456	−3456	−3435.2	−3452.9
	Mean	**−3208.137**	$-1.3532 \times 10^{+11}$	−3051.015	−3854443.57
	St. Dev.	863.9120062	$3.54737 \times 10^{+11}$	424.5019339	15039861.74
	Rank	1	4	2	3

Table 4. *Cont.*

Function	Results	MROA	PSO	MCA	CSA
Griewank	**Min. Value**	0	0	−0.080845	-1.9164×10^{-9}
	Mean	$\mathbf{7.00056} \times 10^{-10}$	0.003452655	−0.09689335	1.17946×10^{-9}
	St. Dev.	1.66136×10^{-09}	0.009206785	0.19265963	4.96471×10^{-9}
	Rank	1	3	4	2
Schaffer 6	**Min. Value**	1.0991×10^{-14}	0	0.010537	-3.9162×10^{-10}
	Mean	$\mathbf{5.68756} \times 10^{-10}$	0.015566735	0.04689305	-6.08983×10^{-10}
	St. Dev.	2.05703×10^{-9}	0.050868454	0.02463849	2.36199×10^{-9}
	Rank	1	3	4	2

4. Engineering Optimization Applications

This section demonstrates MROA's benefits regarding optimizing application parameters in two fields (an online control system application and an offline application). A smart anti-jamming antenna system was chosen as an online system to demonstrate its ability to solve problems of real-time optimization. The dimensions of narrowband microstrip patch antenna are estimated as a problem of offline optimization.

4.1. Smart Antennas with Anti-Jamming

Smart antennas with anti-jamming are extremely helpful in military applications, where they are presented with intentional jamming attempts sent by an adversary. It is a real-time system, as it refreshes its activity boundaries continuously. Figure 5 represents the operation idea and the boundaries of an M-component smart antenna system. The ideal transmission is fixed in the line of sight (LOS) of the receiving station. The other jamming sources are appropriated in various ways, either to communicate fake messages or just to meddle with the target system. The smart antenna system with anti-jamming attempts to arrange the receiving antenna's fundamental beam toward the ideal transmission and finds nulls in the directions of the jamming signals, as shown in Figure 5. In fact, if there is any correlation between the ideal signal and the jamming signal, the beam-forming calculations cannot drop the jamming signal completely [21]. Under this condition, the regular beam-forming calculations can only weaken the jamming signals. Since the jamming signals are sometimes communicated with a higher power than the ideal signal power, attenuation will be a bad solution for this situation. The moss rose algorithm does not depend on the correlation of signals; it completely nullifies the jamming signals without taking into consideration the transmission power.

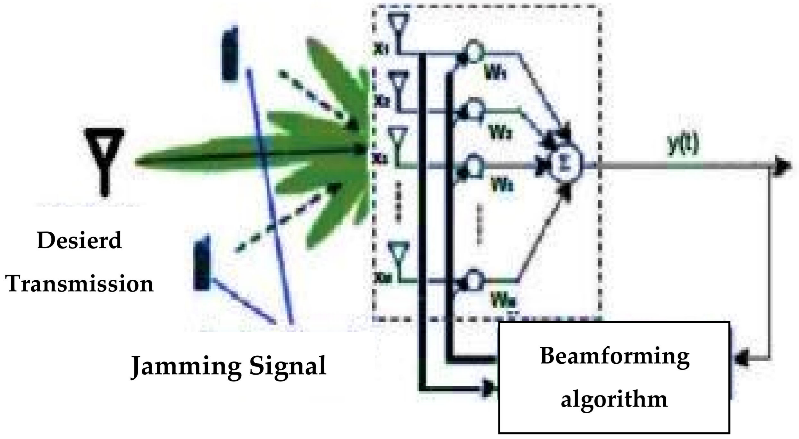


Figure 5. Smart antenna array with M-components.

Let there be $K + 1$ signals that are received by an M-component smart antenna from various directions, where these signals are given by their strengths S (the power signal square root) as $[S_o, S_1, S_2, \ldots, S_K]$. The arrived signal S_o is thought to be the ideal signal, which is received by the smart antenna from the broadside direction ($\Phi = 90°$). Various incoming signals are viewed as jamming signals. The signals that are received by all components can be represented by the vector $x = [x_1, x_2, \ldots, x_m]$ and the weights vector of the system array is represented as $w = [w_1, w_2, \ldots, w_m]$, The smart antenna system output is given by:

$$y = w^H x \tag{6}$$

where H represents the Hermitian transpose. The vector of the received signal that is selected by the smart antenna system is created as follows [21]:

$$x = \sum_{k=0}^{K} S_k a_k \tag{7}$$

where the steering vector's kth signal is referred to as a_k, which has an arrival angle of ϕ_k. The formula for the steering vector signal is as follows [21]:

$$a_k = \begin{bmatrix} 1 \\ exp[j\beta dcos(\phi_k)] \\ exp[j2\beta dcos(\phi_k)] \\ \vdots \\ exp[j(M-1)\beta dcos(\phi_k)] \end{bmatrix} \tag{8}$$

where d represents the space between two adjoining elements in terms of the wavelength λ and β denotes the propagation phase constant:

$$\beta = \frac{2\pi}{\lambda} \tag{9}$$

For an array antenna, the array factor (AF) is equal to the radiation pattern whenever the antenna elements are omnidirectional [22]. Thus, the array factor equation can be used to determine the radiation pattern as follows [21]:

$$AF = w^H a(\phi) \tag{10}$$

where $a(\phi)$ denotes the common steering vector for any angle ϕ:

$$a(\phi) = \begin{bmatrix} 1 \\ exp[j\beta dcos(\phi)] \\ exp[j2\beta dcos(\phi)] \\ \vdots \\ exp[j(M-1)\beta dcos(\phi)] \end{bmatrix} \tag{11}$$

The phase angle δ between the elements of the weights vector can be used to find the angle of the main beam of the antenna, which should be guided toward the desired signal direction ϕ. The phase angle between the elements of the weights vector can be found using the following criterion [22]:

$$\beta d \cos(\phi) + \delta = 0 \tag{12}$$

If the desired signal angle is $\left(\phi = 90^{\circ}\right)$, then δ has a value equal to zero. The null positions of the antenna array factor magnitude are determined using the weights vector.

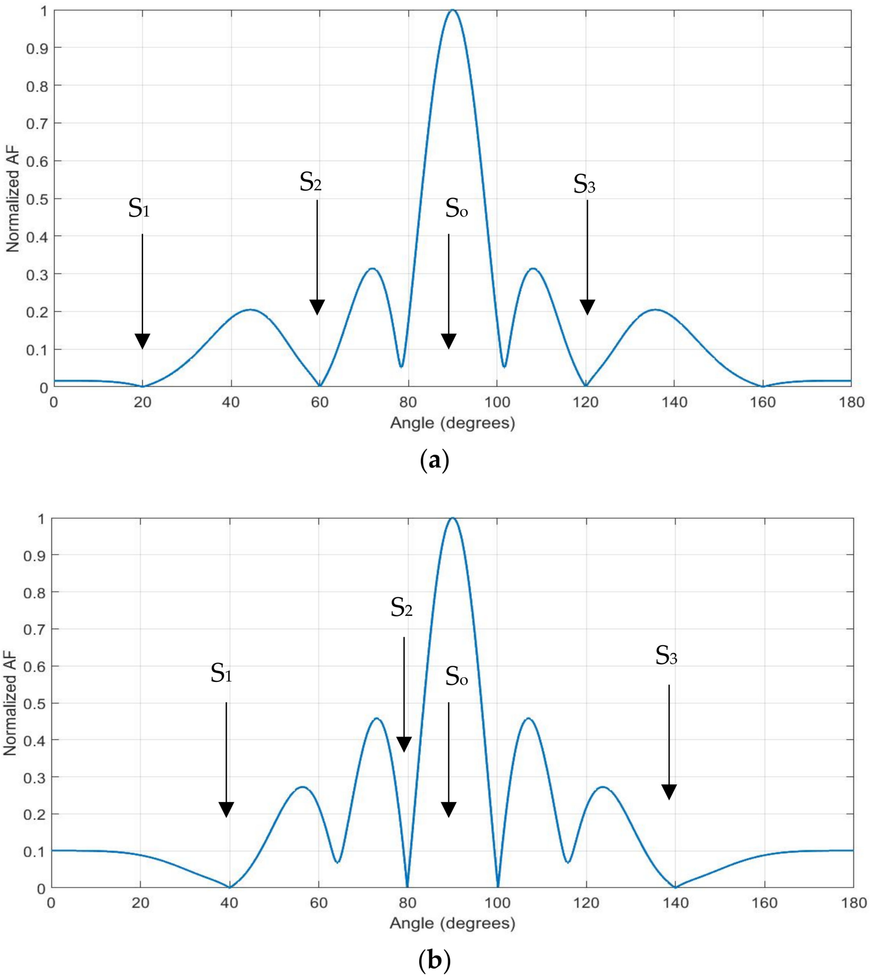
As a result, the weights vector magnitude $|w|$ is the optimization variable of these systems. The proposed objective function F for the system is

$$F = \min[|S - |w^H x||] \tag{13}$$

Numerical example: Consider M = 10 smart antenna equivalents to S_o = 1 with a desired signal power. The assumption of serious jamming involves three strong jamming signals with S_1 = 3, S_2 = 2 and S_3 = 3 powers. For the four jamming signals, the following cases indicate different arrival angles:

Case 1: $\phi 1 = 20^\circ$, $\phi 2 = 60^\circ$, $\phi 3 = 120^\circ$
Case 2: $\phi 1 = 40^\circ$, $\phi 2 = 80^\circ$, $\phi 3 = 140^\circ$
Case 3: $\phi 1 = 50^\circ$, $\phi 2 = 100^\circ$, $\phi 3 = 130^\circ$

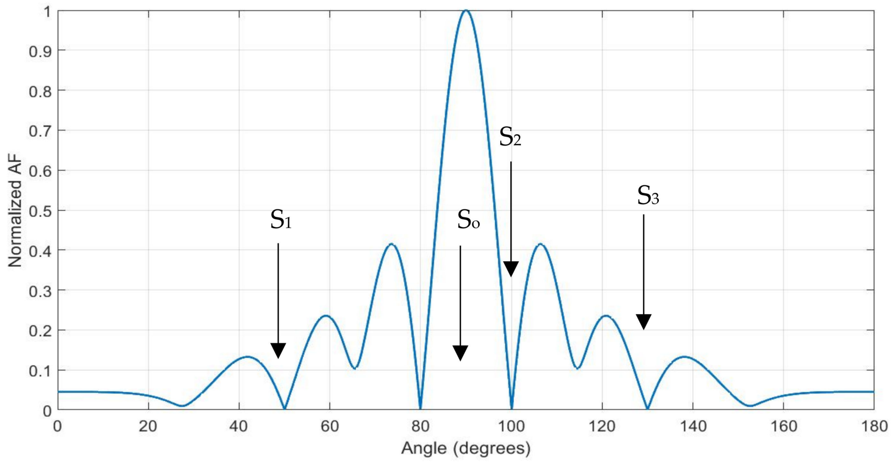
The optimized magnitude of the weight vector is shown in Table 5 after the moss rose flower that was produced for each event. The normalized size of the array factor that resulted from the vector weights for each case is shown in Figure 6. The same numerical example was applied using the other compared algorithms (MCA, CSA and PSO). It is apparent that the antenna system perfectly removed the jamming signal by pointing nulls toward its arrival angle, irrespective of the similarity between the signals for all algorithms, but with different average elapsed times for 20 runs. The smallest elapsed time for finding the optimal weights in the numerical example was found when using the moss rose algorithm. The elapsed time for each algorithm is shown in Table 6.



(**a**)

(**b**)

Figure 6. *Cont.*



(**c**)

Figure 6. Standardized smart antenna array factor using the moss rose algorithm: (**a**) case 1, (**b**) case 2 and (**c**) case 3.

Table 5. The optimized sizes of the vector weights for the smart antenna system proposed.

$\mid \mathbf{w_{opt}} \mid$ **Case 1**	$\mid \mathbf{w_{opt}} \mid$ **Case 2**	$\mid \mathbf{w_{opt}} \mid$ **Case 3**
0.3398	0.2494	0.2824
0.4334	0.6775	0.5368
0.2699	0.4279	0.3438
0.5270	0.1571	0.3622
0.3145	0.1670	0.3168
0.2633	0.3353	0.1845
0.6227	0.1745	0.1952
0.3644	0.4673	0.3178
0.2383	0.5486	0.5901
0.1947	0.2801	0.4896

Table 6. Elapsed times (s) for finding the optimal weights.

Algorithm	**MROA**	**CSA**	**MCA**	**PSO**
Ave. time	0.4039	0.8815	0.4452	1.0504
Rank	1	3	2	4

4.2. Narrowband Microstrip Patch Antenna Design

The 1970s were especially popular regarding space-borne applications for microstrip antennas. They are currently used for public and commercial purposes. A microstrip patch antenna involves a metal patch on a grounded substratum. There are several variations for the metallic patch. However, because of its easy analysis and development and its attractive radiation characteristics, especially the low cross-polarization radiation, the rectangular and circular patches are the most common [23].

A thin metal band (patch) with a substrate height that is a small fraction of a wavelength ($h \ll \lambda$, typically $0.003\lambda < h < 0.05\lambda$) is made up of microstrip antennas that are mounted above the ground plane. The patch is designed in such a way that its pattern is natural for the patch (broadside radiator). This is done by correctly selecting the mode (field configuration) for the patch. Careful mode selection also allows for end-of-fire radiation. The L length of an element is typically $\lambda/3 < L < \lambda/2$ for a rectangular patch. A dielectric sheet separates the strip (patch) from the ground plane (referred to as the substrate), as shown in Figure 7 [22].

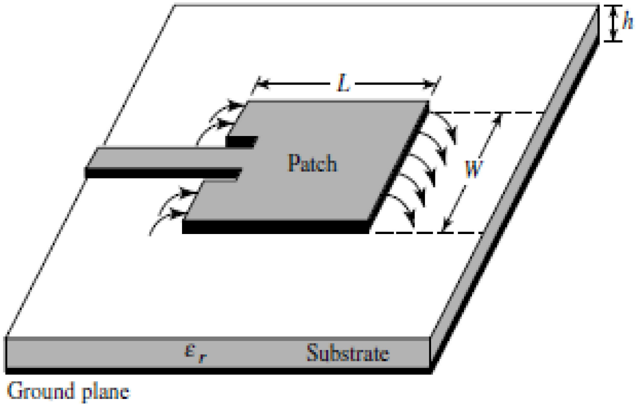


Figure 7. Microstrip patch antenna.

To achieve the desired frequency, this antenna must be equipped with certain parameters.

The design procedure assumed that the specified information contains the dielectric substrate constant (ε_r), as well as the substrate height (h), and is as follows: [22,23]

1. Specify the center frequency and choose a permittivity (ε_r) and the thickness of the substrate (h):

$$h \geq 0.06 \frac{\lambda_{air}}{\sqrt{\varepsilon_r}}$$

2. Find the patch width (W_p) using

$$W_p = \frac{v_o}{2f_r}\sqrt{\frac{2}{\varepsilon_r + 1}} \tag{14}$$

where v_o is the light velocity, and f_r is the resonant frequency.

3. Calculate the effective dielectric constant $\varepsilon_{\text{reff}}$ using the following equation:

$$\varepsilon_{\text{reff}} = \frac{\varepsilon_r + 1}{2} + \frac{\varepsilon_r - 1}{2}\left[1 + 12\frac{h}{W_p}\right]^{-1/2} \tag{15}$$

4. Calculate the extended length (ΔL):

$$\frac{\Delta L}{h} = 0.412 \frac{(\varepsilon_{\text{reff}} + 0.3)(\frac{W_p}{h} + 0.264)}{(\varepsilon_{\text{reff}} - 0.258)(\frac{W_p}{h} + 0.8)} \tag{16}$$

5. Calculate the patch length (L_p) using the following equation:

$$L_p = \frac{v_o}{2f_r\sqrt{\varepsilon_{\text{reff}}}} - 2\Delta L \tag{17}$$

6. Find the notch width using the following equation:

$$f_r = \frac{v_0}{\sqrt{2 \times \varepsilon_{etj}}} \frac{4.6 \times 10^{-14}}{g} + \frac{f}{1.01} \tag{18}$$

7. Calculate the matching impedance Z_o as follows:

$$Z_o = R_{\text{in}} \cos^2(\frac{\pi}{L_p}d) \tag{19}$$

where the input impedance (R_{in}) is obtained using

$$R_{\text{in}} = \frac{1}{2(G_1 + G_{12})} \tag{20}$$

The input admittance G_1 can be calculated using the following equation:

$$G_1 = \frac{1}{120\pi^2} \int_0^{\pi} [\frac{\sin(\frac{k_0 W_p}{2} \cos\theta)}{\cos\theta}]^2 \sin^3\theta d\theta \tag{21}$$

The mutual admittance G_{12} can be calculated using the following equation:

$$G_{12} = \frac{1}{120\pi^2} \int_0^{\pi} [\frac{\sin(\frac{k_0 W_p}{2} \cos\theta)}{\cos\theta}]^2 * J_0(k_0 L_p \sin\theta) \sin^3\theta d\theta \tag{22}$$

Numerical example where the resonant frequency was chosen to be 2.4 GHz: By using the above technique, the width and length of the patch are considered random input variables. The permittivity was 4.3 and the substrate thickness was h = 1.6 mm. The microstrip patch antenna dimensions were found as shown in Table 7.

Table 7. Dimensions (mm) of the microstrip patch antenna.

Dimension	MROA	CSA	MCA	PSO
W_p	**38.3935**	10.1	26.0	40.1
L_p	**29.4**	10.0	32.6	30.0
F_i	**9.0442**	3.0	9.9	9.4
W_f	**3.3353**	1.2	3.4	3.2

It is concluded that the proposed algorithm produces exact dimensions that has minimum reflection coefficient of (S11 = −34.041775 dB) and minimum voltage standing wave ratio of (VSWR = 1.0405184) at the target frequency (2.4 GHz) when compared with the other algorithms. The comparison algorithms have not significant results at the target frequency. The results of S11 and VSWR are as shown in Figures 8 and 9, and Table 8 compares S11 and VSWR values between different algorithms.

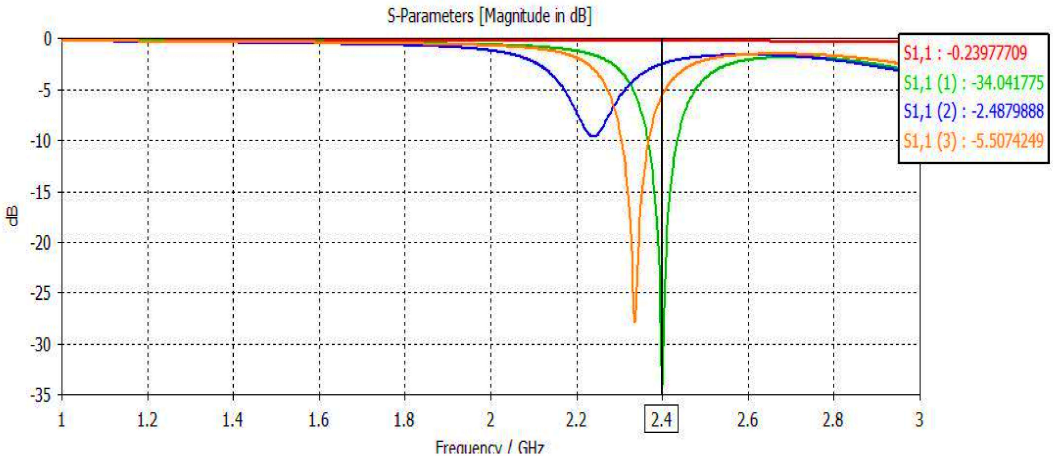


Figure 8. Sr value for target frequency.

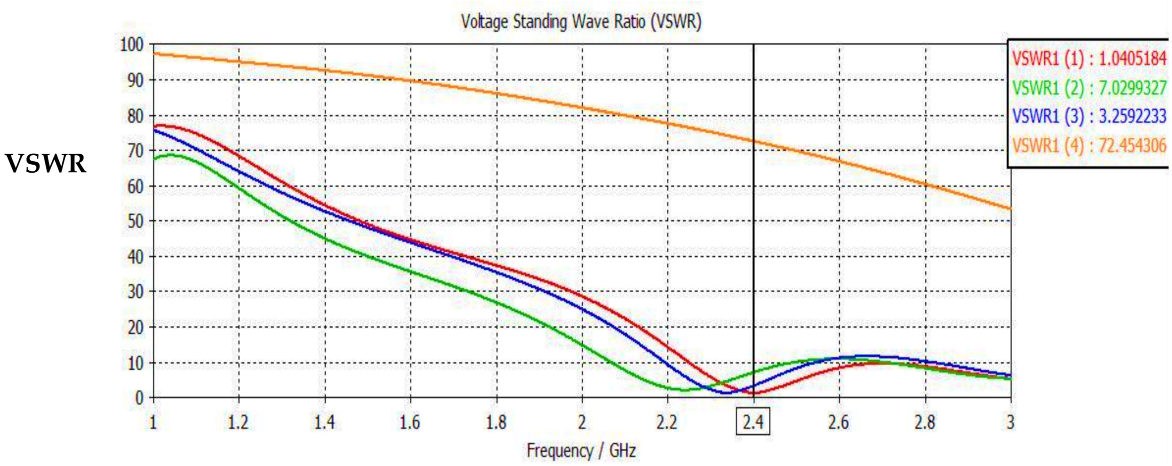


Figure 9. VSWR value for the target frequency.

Table 8. Dimensions (in mm) of microstrip patch antenna.

Parameter	MROA	CSA	MCA	PSO
S_Parameter	**−34.041775**	−0.23977709	−2.4879888	−5.5074249
VSWR	**1.0405184**	72.454306	3.2592233	7.0299327

5. Conclusions

In this study, a new algorithm known as the moss rose optimization algorithm (MROA), which was inspired by the flowering process of the moss rose, was suggested. During the day, a moss rose opens toward the sunlight. For the flowering of plants, proteins, such as florigen, play a major role. Florigen is released to allow for plant flowering with various diameters at different points of the light cycle. The flowers are opened several hours a day and are then closed in preparation for the next day to re-adjust the diameters. The main structure of the algorithm is as follows:

1. Generate random variables.
2. Identify the values of the variables that produced the best solution to the calculated fitness function.
3. Generate a new variable representing the lifetime of the variable.
4. Generating a second variable that represents the extent of the influence of natural factors upon updating the variable data.
5. Update the variables and compare the results with the previous results to obtain convergence toward a better outcome.

The proposed MROA was tested for extremely demanding modern requirements. The results of the benchmark function tests showed that the proposed algorithm was better than other algorithms, especially for multimodal functions, as illustrated in Table 4. In addition, these tests showed that the MROA balanced the exploitation and exploration stages with the results well (especially for multimodal functions) in comparison to other algorithms, namely, the crow search algorithm (CSA), the modified camel algorithm (MCA) and particle swarm optimization (PSO).

MROA's success was also confirmed by optimizing restricted selected engineering applications, namely, an anti-jamming smart antenna as an online problem and the computation of microstrip antenna dimensions for a 2.4 GHz frequency as an offline problem.

For the online problem, the algorithm perfectly removed jamming signals by nullifying them. The advantage of this algorithm in this application came from the algorithm speed when finding the results.

For the offline problem, the algorithm found the optimal dimensions that operated at a target resonant frequency despite the complexity of the system; the other algorithms did not give accurate results for this application.

Author Contributions: Conceptualization, H.M.H., R.S.A. and A.S.A.; methodology, H.M.H.; software, H.M.H.; validation, R.S.A. and A.S.A.; formal analysis, H.M.H.; investigation, R.S.A.; resources, H.M.H., R.S.A. and A.S.A.; data curation, H.M.H.; writing—original draft preparation, H.M.H.; writing—review and editing, R.S.A. and A.S.A.; visualization, A.S.A.; supervision, A.S.A. and R.S.A.; project administration, A.S.A. and R.S.A.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors would like to thank Mustansiriyah university (www.uomustansiriyah.edu.iq) Baghdad-Iraq for its support in present work, also, the authors would like to thank university of Basrah (Basrah-Iraq) for its support in present work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yang, X.-S. *Nature-Inspired Optimization Algorithms*, 1st ed.; Elsevier Inc.: Amsterdam, The Netherlands, 2014.
2. Dorigo, M. Optimization, Learning and Natural Algorithms. Ph.D. Thesis, Politecnico di Milano, Milan, Italy, 1992.
3. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the IEEE CNN'95 International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
4. Mehrabian, A.; Lucas, C. A novel numerical optimization algorithm inspired from weed colonization. In *Ecological Informatics*; Elsevier: Amsterdam, The Netherlands, 2006; pp. 355–366.
5. Yang, X.-S. *Firefly Algorithms for Multimodal Optimization*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 169–178.
6. Yang, X.-S. Engineering optimization by cuckoo search. *Int. J. Math. Model. Numer. Optim.* **2010**, *1*, 330–343.
7. Yang, X.-S. *A New Metaheuristic Bat-Inspired Algorithm, Nature-Inspired Cooperative Strategies for Optimization (NICSO)*; Springer: Cham, Switzerland, 2010; Volume 284, pp. 65–74.
8. Yang, X.-S. *Flower Pollination Algorithm for Global Optimization Lecture Notes in Computer Science*; Springer: Cham, Switzerland, 2012; pp. 240–249.
9. Hao, Z.; Yunlong, Z.; Hanning, C. *Root Growth Model: A Novel Approach to Numerical Function Optimization and Simulation of Plant Root System*; Springer: Cham, Switzerland, 2013.
10. Seyedali, M.; Andrew, L. *The Whale Optimization Algorithm Advances in Engineering Software*; Elsevier Ltd.: Amsterdam, The Netherlands, 2016.
11. Nancy, G.; Jyoti, S.; Kamaljit, S. *Bhatia, Design Optimization of CPW-Fed Microstrip Patch Antenna Using Constrained ABFO Algorithm*; Springer: Cham, Switzerland, 2017.
12. Mohit, J.; Vijander, S.; Asha, R. A novel nature-inspired algorithm for optimization: Squirrel search algorithm. In *ScienceDirect. Swarm and Evolutionary Computation*; Elsevier Ltd.: Amsterdam, The Netherlands, 2018.
13. Ali, R.S.; Alnahwi, F.M.; Abdulkareem, S.A. A modified camel travelling behaviour algorithm for engineering applications. *Aust. J. Electr. Electron. Eng.* **2019**, *16*, 176–186. [CrossRef]
14. Sattar, D.; Salim, R. A smart metaheuristic algorithm for solving engineering problems. *Eng. Comput.* **2020**, 1–29. [CrossRef]
15. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [CrossRef] [PubMed]
16. Raven, P.H.; Johnson, G.B.; Mason, K.A.; Losos, J.B.; Susan, R. *Singer Biology*, 11th ed.; McGraw-Hill Education: New York, NY, USA, 2017.
17. Mizoguchi, T. August Distinct Roles of GIGANTEA in Promoting Flowering and Regulating Circadian Rhythms in *Arabidopsis*. *Plant Cell* **2005**, *17*, 2255–2270. [CrossRef] [PubMed]
18. Jamil, M.; Yang, X.S. A literature survey of benchmark functions for global Optimization problems. *Int. J. Math. Model. Numer. Optim.* **2013**, *4*, 150–194.
19. Kashif, H.; Mohd, N.; Mohd, S.; Shi, C.; Rashid, N. Common Benchmark Functions for Metaheuristic Evaluation: A Review. *Int. J. Inform. Vis.* **2017**, *1*, 218–223.
20. Almoataz, Y.A.; Fathy, A. A novel approach based on crow search algorithm for optimal selection of conductor size in radial distribution networks. *Int. J. Eng. Sci. Technol.* **2017**, *20*, 391–402.
21. Godara, L.C. *Smart Antennas*; CRC Press: Boca Raton, FL, USA, 2004.
22. Constantine, A. *Balanis Antenna Theory Analysis and Design*; Wiley-Interscience: New York, NY, USA, 2005.
23. Muhammad, A.A.A.; Norhudah, S.; Tien, H. Chua Microstrip antenna design with partial ground at frequencies above 20 GHz for 5G telecommunication systems. *Indones. J. Electr. Eng. Comput. Sci.* **2019**, *15*, 1466–1473.