computation

MDPI

Article

Performance and Application Analysis of a New Optimization Algorithm

Junlong Zheng [1,2,†], Chaiyan Jettanasen [1,†] and Pathomthat Chiradeja [3,*,†]

[1] Department of Electrical Engineering, School of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand; 63601014@kmitl.ac.th (J.Z.); chaiyan.je@kmitl.ac.th (C.J.)
[2] Guangxi Electrical Polytechnic Institute, Nanning 530007, China
[3] Faculty of Engineering, Srinakharinwirot University, Bangkok 10110, Thailand
* Correspondence: chiradeja@hotmail.com; Tel.: +66-2329-8330
† These authors contributed equally to this work.

Abstract: Our research focused on an optimization algorithm. Our work makes three contributions. First, a new optimization algorithm, the Maritime Search and Rescue Algorithm (MSRA), is creatively proposed. The algorithm not only has better optimization performance, but also has the ability to plan the path to the best site. For other existing intelligent optimization algorithms, it has never been found that they have both of these performances. Second, the mathematical model of the MSRA was established, and the computer program pseudo-code was created. Third, the MSRA was verified by experiments.

Keywords: maritime search and rescue; mutual inductance; optimization algorithm; test function

Citation: Zheng, J.; Jettanasen, C.; Chiradeja, P. Performance and Application Analysis of a New Optimization Algorithm. *Computation* **2024**, *12*, 1. https://doi.org/10.3390/computation12010001

Academic Editor: Demos T. Tsahalis

Received: 20 November 2023
Revised: 11 December 2023
Accepted: 12 December 2023
Published: 20 December 2023

1. Introduction

Among the actual social production projects, some have a common trait: they not only need to solve the location of the optimal point, but also need to plan the optimal route, make the optimal decision, and so on. Examples of these projects include searching for the location of the maximum magnetic mutual inductance in a three-dimensional magnetic field [1] (Project 1), searching for the location of radioactive sources in three-dimensional space [2], searching for the location of the leakage of hazardous chemicals and toxic volatile substances [3], sea surface search and rescue [4], etc. The equipment in the optimization project is what is explained here, such as the search and rescue helicopter [4] in the sea search and rescue project and the handle-held detector [2] in searching for the location of radioactive sources. To increase labor productivity, this kind of optimization project needs the right optimization method. Numerous optimization techniques are currently in use, including the SMA [5], the sparrow search algorithm [6], and others.

As seen in Figure 1, this was a simulation experiment in which we applied the SMA to Project 1 on a computer. The chaotic curves in the picture are the moving paths taken by the receiving coil. The number of individuals in the population was defined as 20 and the number of cycles as 100. The calculation process created a total of 2020 population members, meaning that there were a total of 2020 iterations. The error value of the computation result was 8.33×10^{-175}. The graphic shows that the route design was disorganized. There were a total of 2020 probe position points. Every two adjacent points in time formed a section of the route, and the total route formed many circuitous paths. The majority of intelligent algorithms, like the SMA, are to blame for this. A population made up of dozens of people is defined at the beginning of each algorithm, such as the Mayfly Algorithm [7], the Butterfly Optimization Algorithm [8], the Monarch Butterfly Optimization Algorithm [9], the sparrow search algorithm [6], the Black Widow Optimization Algorithm [10], and so on. Each person in the population represents a random search location point. A batch of the population is created during each iteration. The requirements can be met even though these

intelligent algorithms frequently need to define thousands of iterations. The definition of the population individual is a random number generated in the calculation range because these algorithms do not take path planning into account and only take optimization calculations into account. Therefore, projects that need to find the best points and perform route planning cannot use the current intelligent optimization algorithms.

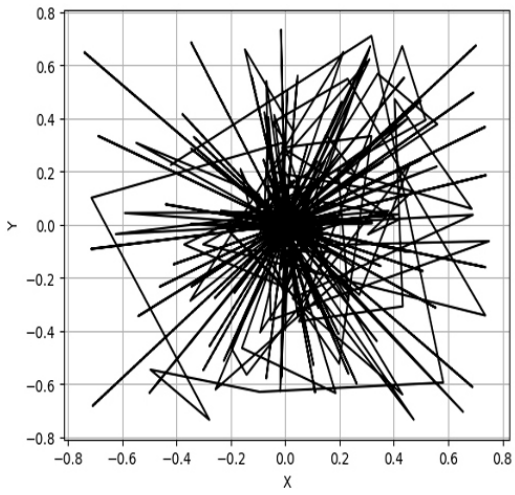

Figure 1. SMA trajectory route of optimization operation.

Shown in Figure 2 is the robot route map based on the path planning of the Gray Wolf Optimization Algorithm (GWA) [11]. Each wolf is a potential solution to the problem, according to the GWA's guiding principle. Each wolf in the path optimization of the mobile robot represents a different path that the robot will take as it moves, and the GWA will use optimization calculations to select the best path from a variety of paths. The path-planning algorithm, obviously, plans the best route between two points while also avoiding obstacles for a given starting point and end point. The existing path-planning methods, such as the GWA, the particle swarm optimization algorithm (PSO) [12,13], etc., are not relevant if they do not know the end point and need to discover the optimal location point as the end point and, also, plan the way to the best point.

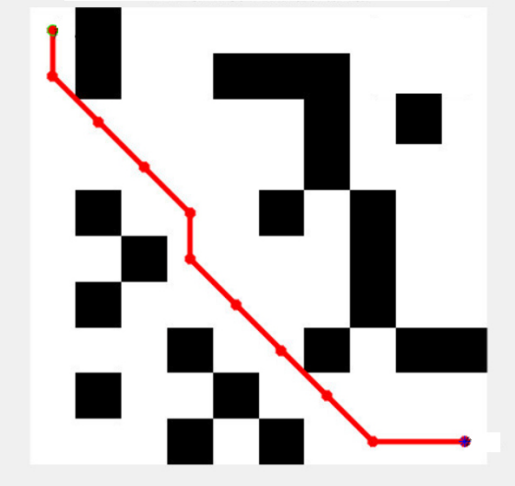

Figure 2. The path planning based on the GWA.

Aiming at the problem of not only searching for the best point, but also route planning and route movement, we propose a new algorithm: the Maritime Search and Rescue Algorithm (MSRA).

2. Studied System

The MSRA was inspired by marine search and rescue missions. For example, in the event of a maritime accident, the search rescue organization is given the task, sends search and rescue ships to the designated sea areas [LB, UB], and then, sends shipborne helicopter1, helicopter2, and so forth, to search, as shown in Figure 3. The algorithm's fundamental idea is then thoroughly explained using the behavior of a single helicopter as an example. As shown in Figure 4, the search starting point for a single helicopter's search and rescue operations is chosen at random within the delimited area [LB, UB]. The helicopter's arbitrary starting point is indicated by the red triangle in the illustration. Formula (1) can be used to describe this behavior:

$$\vec{X}_0 = rand(UB - LB) + LB \tag{1}$$

where $\vec{X}_0$ is the starting position vector, UB is the upper boundary, LB is the lower boundary, and "rand" is the random number between (0, 1). [LB, UB] is the definition area. For Project 1, [UB, LB] is the value range of the XYZ coordinates in the magnetic field space to be searched. For the Ackley test function (see Figure 5), the value range of XY is [−5, +5].

Figure 3. Scenario diagram of maritime search rescue [14].

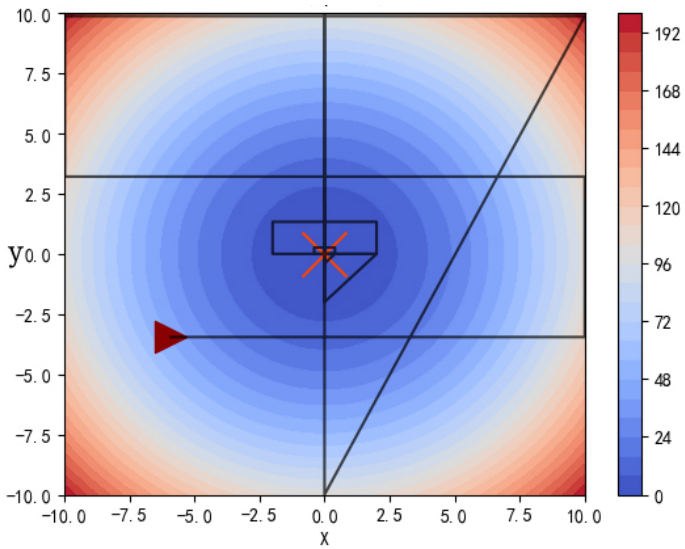

Figure 4. MSRA trajectory to find the optimal value (Sphere function).

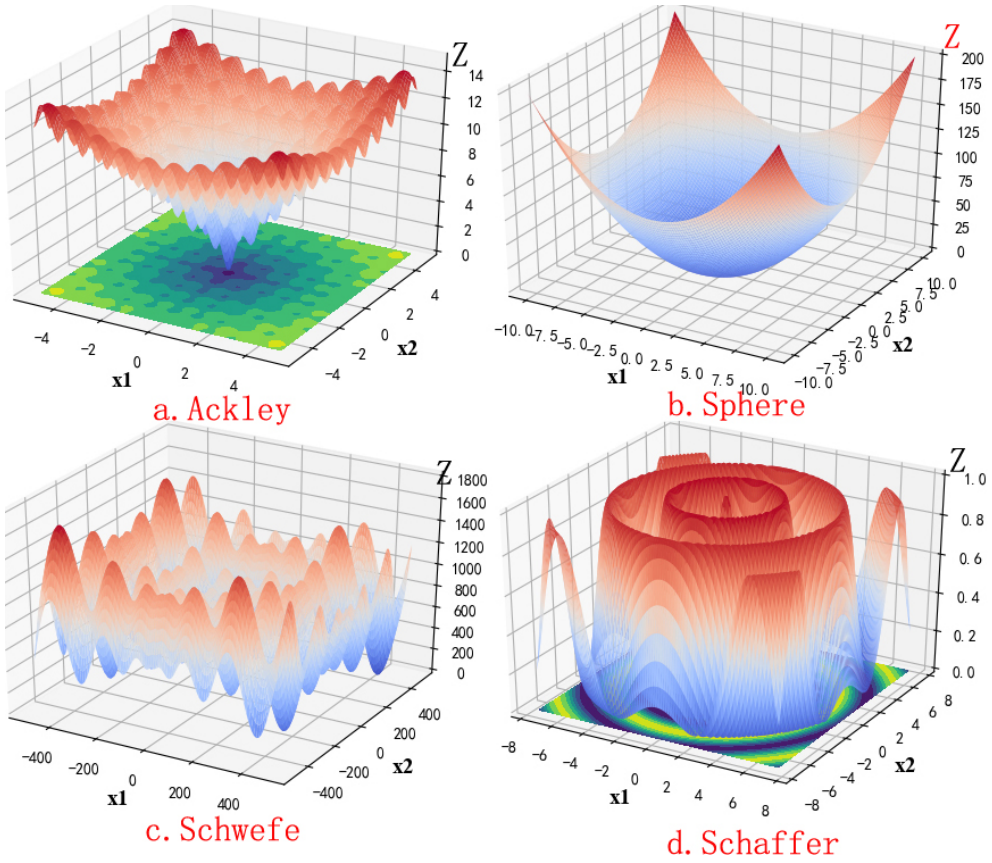

Figure 5. The figures of four test functions.

Linearly scan the sea surface in the specified X direction (or Y direction) starting from the starting point. Formula (2) can be used to describe this behavior:

$$\vec{X}_{t+1} = \vec{X}_t + \vec{Y}_{sep} + direction * X_{step} \tag{2}$$

where $\vec{X}_t$ is the position of the helicopter at time "t" from $\vec{X}_0$; $\vec{X}_{t+1}$ is the position update; $\vec{Y}_{sep}$ is the line spacing between each line of scanning; "direction" is the search direction; "Xstep" is the search step; these parameters above can be expressed by Formulas (3)–(5):

$$direction = (-1)^i \tag{3}$$

$$X_{step} = \frac{UB - LB}{step} \tag{4}$$

$$\vec{Y}_{sep} = \frac{UB - LB}{rows} \tag{5}$$

where "i" is the scanning order (0, 1, 2, 3 . . .). It scans in the positive direction at the start of i = 0. Return to scanning with i = 1 once you have reached the upper boundary. When scanning in the positive direction, "i" is even, and when scanning in the negative direction, it is odd; this cycle is followed. "Step" refers to the quantity of visits made while scanning in a positive (or negative) direction and can be interpreted as an integer of the form of 50, 100, 1000, etc. The term "rows" refers to the total number of scanning lines that are anticipated during the entire defined interval and can be represented as an integer such as 5, 6, 7, etc. The distance between planned scanning lines in maritime search and rescue should be as short as possible when encountering complex water conditions and low visibility, which increases the number of scanning lines. The two parameters "step"

and "rows" control the sensitivity and error of the MSRA. The sensitivity increases with the value, and this has an impact on the MSRA's computation time as well.

After finishing the x-direction scanning, the aircraft repeatedly scans the suspected area in the y direction. Formula (6) can be used to describe this behavior.

$$\vec{Y}_{t+1} = \vec{Y}_t + \vec{X}_{best\,fitness} + direction * Y_{step} \tag{6}$$

where "direction" and "Ystep" have the same significance as their equivalent counterparts in the x-direction scanning. This is not stated again here. $\vec{X}_{best\,fitness}$ represents the position corresponding to the optimal fitness determined in the previous X-direction scanning. For the minimum problem, the position vector corresponding to the minimum value was obtained. For the maximum problem, the position vector corresponding to the maximum value was discovered.

After completing the previously mentioned all-around rough scanning, the search region was shrunk to concentrate on the ideal suspicious place. With the exception of the smaller search region, the scanning process was the same as the rough scanning method described above. The mathematical Equations (7)–(9) can be used to describe this behavior.

$$O_{new} = \vec{Y}_{best\,fitness} \tag{7}$$

$$UB_{new} = O_{new} + \frac{UB - LB}{2max[5, 8 * (t-2)]} \tag{8}$$

$$LB_{new} = O_{new} - \frac{UB - LB}{2max[5, 8 * (t-2)]} \tag{9}$$

where [UBnew, LBnew] is a newly defined narrowing boundary that is allocated to [UB, LB], after which the x-direction and y-direction scanning described above is repeated. "t" is the sequence number of this repetition. It is clear from the formula that the convergence speed of the MSRA is governed by max(5, 8 * (t − 2)) and that this speed has an impact on the calculation inaccuracy of the MSRA. To decrease the convergence speed and increase the calculation accuracy for complicated calculation targets, the convergence parameters can be appropriately changed.

In order to facilitate computer programming, we wrote the pseudo-code of the MSRA.

In Algorithm 1, line 2 defines several necessary parameters. These parameters were explained in detail in the previous mathematical derivation.

Line 3 writes Formula (1) as a computer program;
Line 4 sets a loop body. The number of loops is determined by MSRALoop.
Line 5 writes Formula (2) as a computer program.
In line 6, the coordinate points scanned in the X direction are substituted into the test function to calculate and take the optimal value. For Project 1, this step is to collect the mutual inductance data scanned in the X direction and obtain the optimal value.
In line 7, the Y coordinate corresponding to the best fitness is assigned to Yt as the initial value of the Y-direction scan.
Line 8 writes Formula (6) as a computer program.
In line 9, the coordinate points scanned in the Y direction are substituted into the test function to calculate and take the optimal value. For Project 1, this step collects the mutual inductance data scanned in the Y direction and obtains the optimal value.
Line 10 writes Formulas (7)–(9) as a computer program.
In line 11, judge whether the loop calculation is completed or not.
In line 12, take the optimal value and the corresponding XY coordinates.

Algorithm 1 The pseudo-code of the MSRA.

```
Begin
Define the parameters: UB, LB, step, rows, Ysep, Ystep, Mter
Initialize the random position of search and rescue aircraft: X0.
While t in [1, MSRALoop]  {
    For i in range (0, step):{ X_{t+1} = X_t + Y_sep + direction * X_step } End for
    Calculate the fitness of all X_t,  and then,  obtain the best fitness of X
    Assign the Y coordinate corresponding to the best fitness to Yt
    For i in range (0, step):{ Y_{t+1} = Y_t + X_bestFitness + direction * Y_step } End for
    Calculate the fitness of all Y_t,  and then,  obtain the best fitness of Y
    Calculate the new UB, LB,  and then,  update UB, LB
    t = t + 1}  End While
Return bestFitness and the corresponding XY coordinates
End
```

3. Performance of the Optimization Algorithm

To verify the effectiveness of the MSRA, this study introduced several promising optimization algorithm test functions. Ackley, Sphere, Schaffer, Schwefe [15], etc., were some of the test functions used for verification. The following is an introduction to the functions.

3.1. Ackley Function

The Ackley function comes first. Its two-dimensional shape is characterized by an almost flat outer region, as shown in Figure 5a. The variables' range of limitations is $[-5, 5]$. Many valleys or peaks modulated by cosine waves are superimposed in this nearly flat area, creating an uneven surface and a sizable hole in the center. For optimization algorithms, especially hill-climbing algorithms, this function poses the risk of becoming stuck in one of its numerous local minima. Formula (10) provides the formulation for this function [15].

$$f_1(x) = -20exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}(x_i^2)}) - exp(\frac{1}{D}\sum_{i=1}^{D}(\cos(2\pi x_i))) + 20 + exp(1) \quad (10)$$

3.2. Sphere Function

The second is the Sphere function. It is a unimodal function with minimization as its goal and is used to check the algorithms. As shown in Figure 5b, this function's two-dimensional variable value range is $[-10, 10]$, its global minimum value is 0, located at (0, 0), and its formulation is Formula (11) [15].

$$f_2(x) = \sum_{i=1}^{n} x_i^2 \quad (11)$$

3.3. Schwefe Function

The Schwefe function has many local small peaks and valleys, and the image of its two-dimensional variable is like a large series of rolling mountains, as shown in Figure 5c. This function's variable value range is $[-500, 500]$. The global minimum value of the function is 0, located at coordinates (420.9687, 420.9687), and the global maximum value is 1675.9316, which can be found at $(-420.9687, -420.9687)$. Its formulation is Formula (12) [15].

$$f_3(x) = 418.9829d - \sum_{i=1}^{d} x_i \sin(\sqrt{|x_i|}) \quad (12)$$

3.4. Schaffer Function

The Schaffer function is often used to test the performance of optimization algorithms. The function exhibits strong fluctuations and numerous extreme points. The global maxi-

mum value is 1, located at (0, 0), and there are local extreme values surrounded by infinite circles at the periphery of the global optimum. The two-dimensional variable value range of this function is [−7.5, 7.5], and the image is shown in Figure 5d. Its formulation is Formula (13) [15]:

$$f_4(x) = 0.5 - \frac{\left[\sin\left(\sqrt{x_1^2 + x_2^2}\right)\right]^2 - 0.5}{\left[1 + 0.001 \times (x_1^2 + x_2^2)\right]^2} \quad (13)$$

In addition, there are many optimization algorithm test functions, such as the Griewank function [16], the Rastrigin function, the Levy function, the Langermann function, and so on, that are common functions and datasets used for testing optimization algorithms [17], which were not described in detail here because of space constraints.

3.5. Performance Comparison between MSRA and Two Other Algorithms

To verify the effectiveness of the MSRA, this study introduced several promising optimization algorithm test functions above and compared the test results with the classical Genetic Algorithm (GA) [18] and the SMA.

In the 1970s, the GA was first advocated by John Holland in the United States of America [19]. The algorithm was created and put forth in accordance with the natural rules of evolution that apply to creatures. It is a computer simulation of the biological evolution process that replicates the genetic and natural selection mechanisms described in Darwin's theory of biological evolution [20]. This technique mimics the natural evolution process in order to find the best answer. The method changes the process of problem-solving into one that is analogous to the crossing and mutation of chromosomes and genes in biological evolution through mathematics and computer simulation [19]. When compared to other traditional optimization techniques, it typically produces better optimization outcomes more quickly when handling complex combinatorial optimization issues. The GA has been made extensive use of in combinatorial optimization, machine learning, signal processing, adaptive control, and artificial life [21].

The SMA was proposed by Li et al. in 2020 [5]. It was inspired by the diffusion and foraging behavior of slime molds. The SMA mainly simulates the behavior and morphological changes of slime molds during the foraging process without modeling the complete life cycle. The weight index was used to simulate the three correlations between the morphological changes of the myxomycete venous duct and the contraction mode. The algorithm is a meta-heuristic algorithm, which has the characteristics of fast convergence and strong optimization ability.

To unify the assumptions, the number of iterations was defined as one operation by substituting independent variables into the objective function each time, which was counted as one iteration.

The following describes the performance comparison analysis of the MSRA, SMA, and GA applied to the four aforementioned test functions.

For the three algorithms, the maximum number of iterations was set to 5000. For the three functions, Ackley, Sphere, and Schwefe, the three algorithms all performed the operation of finding the minimum value. For the Schaffer function, the three algorithms all performed the operation of finding the maximum value. After calculation, the iterative curves are shown in Figure 6, and the error values of the calculation results are shown in Table 1. Among them, Figure 6d shows finding the maximum value and Figure 6a–c all show finding the minimum value.

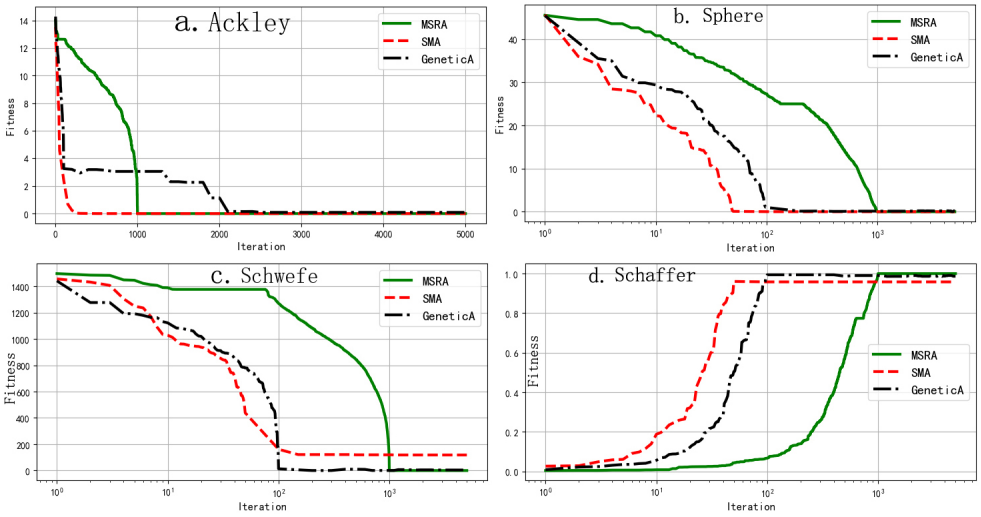

Figure 6. The iteration curves of three algorithms applying the four test functions.

Table 1. The 5000 iterations' error table—4 functions.

Test Function	Ackley	Sphere	Schaffer	Schwefe
Iteration	**5000**	**5000**	**5000**	**5000**
MSRA Error	3.55×10^{-15}	2.59×10^{-31}	6.49×10^{-15}	2.55×10^{-5}
SMA Error	2.31×10^{-86}	2.39×10^{-176}	4.19×10^{-2}	118.44
GA Error	8.71×10^{-2}	1.45×10^{-1}	1.34	4.343

Analyzing the iteration curves and error table, it was found that the SMA had the fastest convergence rate, followed by the GA and MSRA. For Ackley and Sphere, the two test functions, the three algorithms all showed better optimization performance. The SMA had the lowest error value and the highest accuracy, followed by the MSRA. Also, the MSRA had an error value of 10×10^{-15}, which is suitable for most applications. However, the GA had the largest error value, reaching a level of 10×10^{-2}, which makes it difficult to meet most applications. For relatively complex function, such as Schaffer and Schwefe, the MSRA still showed excellent optimization performance; the SMA and GA performed much worse, and it was estimated that both algorithms fell into local extremes in such a complex test function.

In the classic GA, the natural selection in the calculation process is the roulette method, that is individuals with high fitness have a high probability of being selected, inheriting their genes and phenotypes, and individuals with low fitness are easily eliminated. Due to the random roulette method, the individuals with the best fitness may also be eliminated, resulting in different results obtained from multiple runs of the classical GA, which are only close to the optimal solution, which is a limitation of the classical GA.

Try to increase the three algorithms' iteration times to 20,000. The adjusting procedure is as follows: for the MSRA, the number of scanning lines and visitation density were increased for the functions with many local extrema, and the number of cyclic operations correspondingly decreased. For functions similar to the Sphere function, the MSRA decreased the number of scanning lines while correspondingly increasing the visitation density and cyclic operations. For the SMA and GA, for functions with many local extrema, the number of populations can be raised and the number of cyclic operations can be correspondingly lowered for functions with several local extrema. The number of populations can be decreased and the number of cyclic operations can be correspondingly raised for similar functions like the Sphere function. The three algorithms were also applied to the four aforementioned test functions after the iteration times were determined. In Table 2, the operational outcomes are displayed.

Table 2. The 20,000 iterations' error table—4 functions.

Test Function	**Ackley**	**Sphere**	**Schaffer**	**Schwefe**
Iteration	**20,000**	**20,000**	**20,000**	**20,000**
MSRA Error	3.55×10^{-15}	4.40×10^{-39}	0	3.32×10^{-5}
SMA Error	0	0	9.73×10^{-3}	4.12×10^{-5}
GA Error	6.30×10^{-2}	3.43×10^{-2}	9.72×10^{-3}	2.90×10^{-1}

According to the analysis of the data in Table 2, for functions similar to spherical, conical, or disk terrain, such as the Ackley and Sphere functions, on the premise that the number of iterations was controlled at 20,000, it can be seen that the performance of the SMA was slightly better than the MSRA, while the GA was the worst; for mountain-shaped functions, such as the Schaffer and Schwefe functions, the MSRA had the best performance, and the SMA's estimation fell into local extreme points, while the GA had the worst performance.

By comparing and analyzing the data in Tables 2 and 3, regardless of the number of iterations, the performance of the GA was the worst among the three algorithms. For functions with a simple shape, when the number of iterations was 5000, the performance of the SMA was obviously better than the MSRA, but when the number of iterations was increased to 20,000, the performance of the SMA and MSRA was very similar. For functions with complex shapes, regardless of the number of iterations, the performance of the MSRA was obviously better than the SMA.

Table 3. Error and optimization target point location—Schwefe function.

Running Sequence	**1st**		**2nd**		**3rd**	
Iteration	**10,000**		**10,000**		**10,000**	
	Error	Location	Error	Location	Error	Location
MSRA	3.64×10^{-12}	(−420.9688, −420.9688)	5.06×10^{-8}	(−420.9692, −420.9692)	9.40×10^{-8}	(−420.9680, −420.9692)
SMA	368	(−453.7, 500)	392.6	(−432.5, −241.1)	137.7	(−409.3, 306.9)

The following focuses on the performance of the MSRA and SMA on complex test functions.

In order to more clearly see the performance of the MSRA and SMA applied to complex test functions (such as the Schwefe function), through computer programming, for the MSRA, it is marked on the contour map of the test function during the calculation process, which included the aircraft's exploration point, the search and rescue starting point ">", and the optimal fitness position point "X". Simultaneously, according to the search sequence, these points are connected with line segments that represent the flight route of the search and rescue aircraft, as shown in Figure 7. For the SMA, the starting point ">", search location point, and optimal fitness position "X" of the slime mold in the operation process are marked on the contour map of the test function and connected by line segments according to the order of search. For the MSRA, the contour map of the test function can actually be considered as the search and rescue sea surface in the specified area, and the highest point or the lowest point is the target point of the search and rescue. For the SMA, the contour map of the test function can be considered as the area where the slime mold forages in the specified area, and the highest or lowest point is the location where the food abundance is optimal. For the Schwefe function, in order to clearly see that the number of iterations of the iterative optimization trajectory cannot be set too large, both the MSRA and SMA only set the number of iterations to 10,000 times, all of which were operations

to obtain the maximum value, which were run three times, respectively. The statistics of the optimal position and error value are shown in Table 3; Figure 7 is the optimization calculation trajectory of the two algorithms.

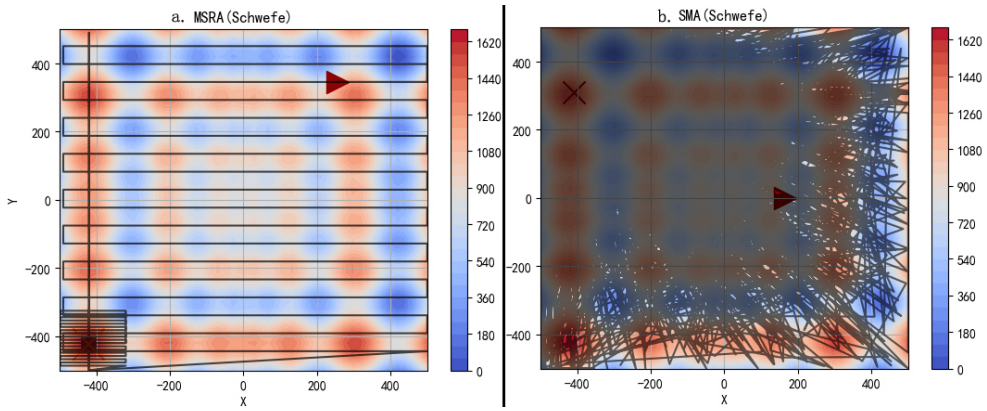

Figure 7. MSRA and SMA search track (Schwefe function).

Table 3's data analysis reveals that the MSRA consistently calculated the optimal position with a very low error value. In contrast, for the SMA, the optimal location points obtained by each operation were very different and tended to randomly fall into the local extreme points. As a result of the optimization operation, the fitness position "X" in the trajectory diagram for the MSRA is right, and the SMA's fitness position is wrong.

The MSRA trajectory is evident from an analysis of the trajectory diagram, and as a result, it is suitable for engineering projects like maritime search and rescue, locating the maximum magnetic induction strength in a three-dimensional magnetic field, locating the source of an electromagnetic wave in three dimensions, and other tasks that require actual mobile optimization equipment. On the other hand, the SMA's trajectory is chaotic, erratic, and disorderly. The SMA is unsuitable for engineering tasks requiring genuine mobile optimization equipment. It only functions when an objective function is being optimized on a computer.

The Schaffer function's maximum value is located on a small cusp in the middle. Try increasing MSRA's scanning line density and visiting point density, increasing SMA's population number and cycle operation times, and setting both algorithms' iterations to 100,000. The two algorithms were applied to the Schaffer function, and each algorithm was then run ten times. The trajectory diagram for one of the operations is shown in Figure 8. For the SMA, only the iterative position points are shown on the diagram because the trajectory diagram is a chaotic line segment and useless. It is obvious that, as a result of the optimization operation, for the fitness position "X" in the trajectory diagram, the MSRA's fitness position is right, and the SMA's fitness position is wrong. The error value and optimization target point position for the operation outcome are shown in Table 4. The data in the table were retained only three times.

Table 4. Error and optimization target point location—Schaffer function.

Running Sequence	**1st**		**2nd**		**3rd**	
Iteration	**100,000**		**100,000**		**100,000**	
	Error	Location	Error	Location	Error	Location
MSRA	0	$(-1.2129 \times 10^{-14}, 1.2129 \times 10^{-14})$	0	$(-1.21291 \times 10^{-14}, 1.21291 \times 10^{-14})$	0	$(-1.1852 \times 10^{-14}, 1.2129 \times 10^{-14})$
SMA	9.717×10^{-3}	(3.13559095 0.11067923)	9.716×10^{-3}	(2.31053014, −2.12505417)	9.721×10^{-3}	(−2.20396153, 2.23757538)

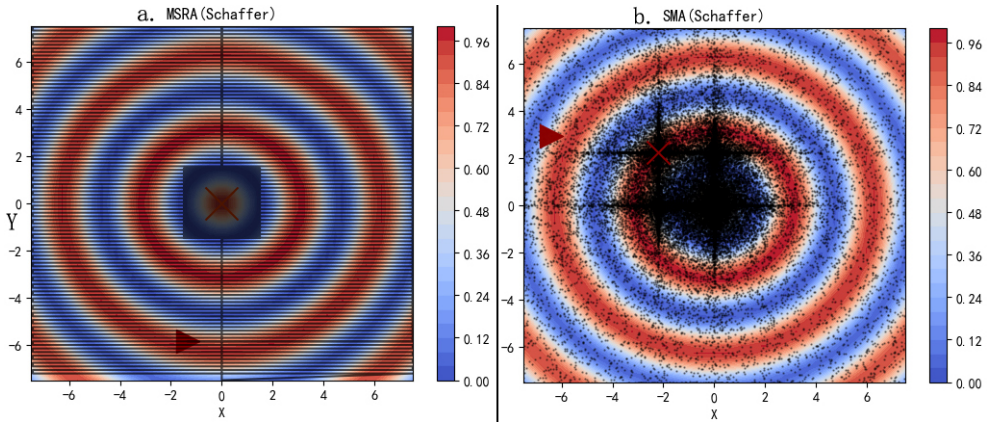

Figure 8. MSRA and SMA search track—Schaffer function.

The values in Table 4 show that the MSRA was applied to the Schaffer function to find the maximum value, and the error value of the operation result was 0. Such excellent results can be obtained after many operations. However, the SMA still randomly fell into the local extreme points, as confirmed by its iterative trajectory and the calculated best position coordinate value, and the calculation result error was relatively large.

For the complex function optimization problem, the SMA will arbitrarily land in the local extreme value.s Investigate the cause. According to the SMA principle, the slime mold population is globally distributed at random at the beginning of the first cycle. At this point, it is inevitable that some areas would have much slime mold while other areas would be empty. The portrayal of its repetitive journey lends credence to this. Beginning with the second series of procedures, the search region is then swiftly constrained around the best point identified during the previous search. Reaching local extreme points for functions as complex as the Schwefe and Schaffer functions is made simple by this algorithmic concept. Likewise, any algorithm that utilizes the idea of a random population distribution, such as the sparrow search algorithm [6], has these shortcomings by default.

The MSRA principle is that the exploration points are uniformly distributed globally during the first cycle operation; then, starting with the second cycle operation, the search range is reduced at a certain convergence speed to improve the output accuracy; for problem optimization calculations, unlike swarm intelligence optimization algorithms like the SMA, which use the principle of a randomly distributed population, the MSRA does not have this limitation. According to this principle, the MSRA can be used to solve the majority of optimization calculation issues.

Another notable benefit of the MSRA is that it can use a straightforward search route trajectory to find the ideal position point for functions like the Sphere function, which are similar to spherical or conical surfaces, but lack local extrema. According to Figure 4, which uses the Sphere function as an example, "$>$" represents the search's randomly chosen starting point; the black line denotes the search route's trajectory, and "X" denotes the best fitness position that was looked for. Three scanning lines, one column, one thousand visits per line, five cyclic operations, and twenty thousand iterations make up the MSRA's parameter settings. The error value of the calculation result was 2.87×10^{-26}. The figure shows how the MSRA can accomplish such accuracy with such a straightforward and unambiguous track route.

In addition, we also tested and compared all the other test functions mentioned in Reference [17], a total of 47, and came to the same conclusion as above, which were not described in detail here because of space constraints.

The research mentioned above makes it clear that the MSRA performs optimization operations well and that its optimization process is simple and structured. It is expected that the MSRA will have a bright future in terms of applications.

4. Experiment

The wireless power transfer automatic alignment device for electric vehicles (WPT-AAD) experimental benchmark can be used to test the performance of the MSRA. The experimental benchmark in this study was the same as that used in Reference [1]. As shown in Figure 9, the WPT-AAD's automatic alignment control technology is based on finding the position with the largest mutual inductance coefficient and simultaneously planning the route to move the transmitting coil L1 to this optimal position (Project 1). Figure 9's notes are list in Table 5.

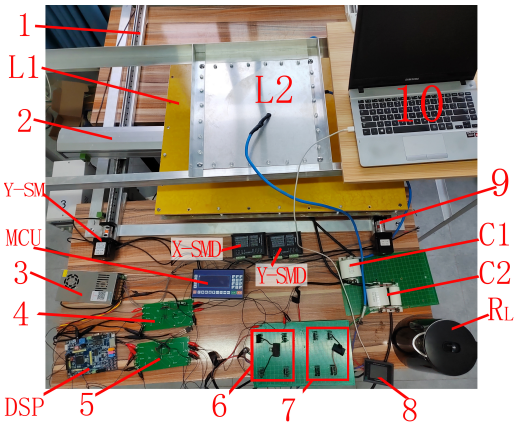

Figure 9. Photos of the WTP-ADD experimental platform [1].

Table 5. Figure 9's notes.

1	Y-axis screw-rod-1	C1	Resonant capacitance at the transmitter
2	X-axis screw-rod	C2	Resonant capacitance at the receiver
3	15V DC regulated power supply	DSP	Digital signal processing
4	Gate-drive-circuit-1	L1	Transmitting coil
5	Gate-drive-circuit-2	L2	Receiving coil
6	Full-bridge inverter circuit module	MCU	Microcontroller unit
7	Full-bridge rectifier filter circuit in output module	RL	Load resistance in the output module
8	Output voltage current sampling circuit	X-SMD	Stepper motor driver in X-axis
9	Y-axis screw-rod-2	Y-SM	Stepper motor in Y-axis
10	Upper computer;	Y-SMD	Stepper motor driver in Y-axis

5. Result

Experiment 1: Input DC voltage Ui = 400 V to the MCR-WPT system, Distance D = 20 cm between transmitting coil L1 and receiving coil L2, system load resistance RL = 70 Ω, entered drive frequency f = 83.0–87.0 kHz. This experiment was completed in Reference [1]. It can be seen that the resonant frequency of the MCR-WPT of this experimental platform was 85.3 kHz, and the corresponding maximum system efficiency was 80.41%.

Experiment 2: Carry out a mutual induction distribution experiment with a driving frequency f = 85.3 kHz, an input voltage of 400 V, distance D = 20 cm between transmitting coil L1 and receiving coil L2, and system load resistance RL = 70 Ω. The main purpose of this experiment was to accurately measure the mutual inductance (M) when the relative position of L1 and L2 changes. This experiment was also completed in Reference [1]. With the help of the Python software, the experimental data were drawn into images, as shown in Figure 10.

Looking up the experimental data, the maximum value of mutual induction M was Mmax = 7.76 × 10^{-6} (H), and there were several positions corresponding to the maximum

value: (−0.096, −0.0127), (−0.0960, 0.0127), (−0.041, 0.0127), (0.096, 0.0127); these positions were all near the coordinate origin by observing Figure 10 [1].

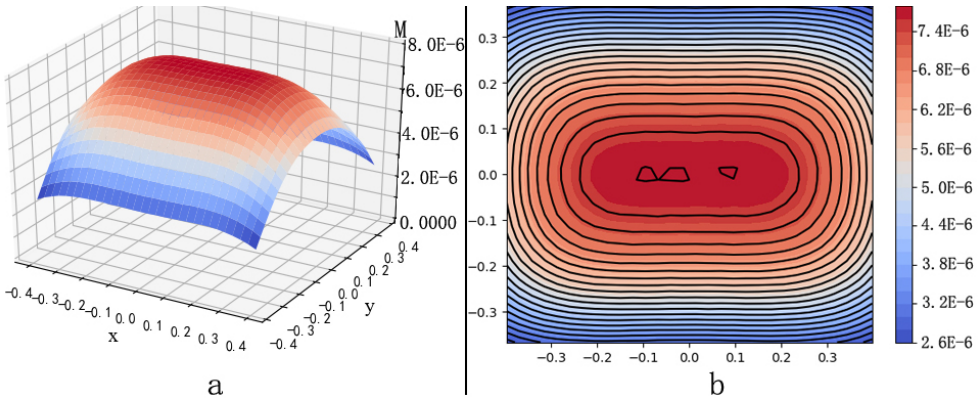

Figure 10. (**a**) M distribution map and (**b**) contour map [1].

Experiment 3: A WPT-AAD experiment was performed with drive frequency f = 85.3 kHz, input voltage Ui = 400 V, distance D = 20 cm between L1 and L2, and system load resistance RL = 70 Ω. The MSRA described in this study was applied to the WPT-AAD experiment to look for the location of the largest M. Only 600 iterations were set during the experiment to decrease the amount of data; hence, only 600 sets of data were acquired. Due to space constraints, only the first 8 rows of data and the last 8 rows of data are presented in Table 6.

Table 6. Experiment 3: MSRA applied to WPT-AAD for finding the position of the maximum mutual inductance.

X Coordinate, m	Y Coordinate, m	Input Voltage, Ui, V	Input Current, Ii, A	Output Voltage, Uo, V	Output Current, Io, A	WPT Efficiency η, %	Mutual Inductance, M, H
−0.312	−0.139	400	7.94	405.97	5.8	74.13	6.48×10^{-6}
−0.292	−0.139	400	7.94	408.53	5.84	75.07	6.65×10^{-6}
−0.272	−0.139	400	7.94	410.51	5.86	75.8	6.78×10^{-6}
−0.252	−0.139	400	7.95	412.22	5.89	76.37	6.89×10^{-6}
−0.232	−0.139	400	7.95	413.49	5.91	76.82	6.97×10^{-6}
−0.212	−0.139	400	7.93	414.05	5.91	77.18	7.04×10^{-6}
−0.192	−0.139	400	7.94	414.9	5.93	77.39	7.09×10^{-6}
−0.172	−0.139	400	7.94	415.11	5.93	77.51	7.11×10^{-6}
There are 600 lines of data in total; the middle part has been omitted; the following is the last 10 lines of the data							
−0.057	0.017	400	7.94	422.53	6.04	80.30	7.74×10^{-6}
−0.057	0.018	400	7.94	422.69	6.04	80.36	7.75×10^{-6}
−0.057	0.019	400	7.94	422.60	6.04	80.33	7.74×10^{-6}
−0.057	0.020	400	7.94	422.54	6.04	80.31	7.74×10^{-6}
−0.057	0.021	400	7.94	422.60	6.04	80.33	7.74×10^{-6}
−0.057	0.022	400	7.94	422.67	6.04	80.36	7.74×10^{-6}
−0.057	0.023	400	7.94	422.53	6.04	80.33	7.74×10^{-6}
−0.057	0.002	400	7.94	422.81	6.04	80.41	7.76×10^{-6}

Except for η and M, the data in Table 6 were obtained directly in the experiment. The calculation method of η and M was the same as that of Reference [1].

Because Table 6 does not contain all of the data from the experiment, the XY coordinates of all of the data were plotted on a two-dimensional curve using the Microsoft Excel software, which can intuitively express the track route of the MSRA looking for the location point of the maximum output power, that is the track route of the MSRA looking for the location point of maximum mutual induction M, as shown in Figure 11.

The yellow ">" in Figure 11 represents the starting place of the MSRA search. The "X" denotes the target location that the MSRA looks for and also serves as the search's endpoint. According to Table 6, the MSRA target point position is (−0.057, 0.002), the related η is

80.41%, and the corresponding M is 7.76×10^{-6}. This is the greatest value of η or M within the prescribed interval in Experiment 2's experimental data.

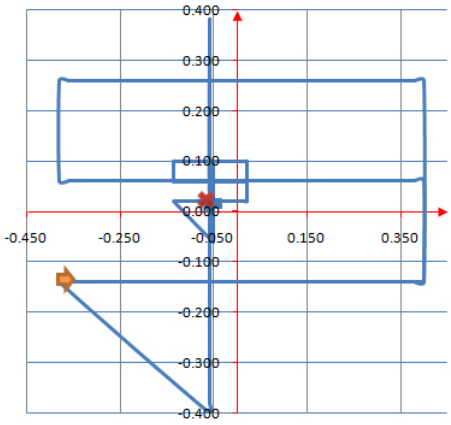

Figure 11. Track roadmap of MSRA being applied to WPT-AAD for finding the position of the maximum M.

When Figure 11 is compared to Figure 10 from Experiment 2, it is discovered that Figure 11 displays the location of the target point with the MSRA searching, which falls exactly where Figure 10 has the deepest red color, indicating that M has the highest value.

The experimental results showed that the MSRA can be used to find the largest M location and plan the route of the search process. Its route track is clean, neat, and orderly. It can be estimated that the MSRA can be expanded and applied to engineering projects that call for real-world mobile optimization equipment, such as maritime search and rescue, searching for the location of radioactive sources in three-dimensional space, searching for the location of the leakage of hazardous chemicals and toxic volatile substances, etc.

6. Conclusions

The MSRA has a significant optimization ability, based on the aforementioned study of the three optimization algorithms and the experimental confirmation. Despite the slower convergence speed, the MSRA can handle most optimization problems with a modest total number of iterations, especially for complex functions like the Schwefe and Schaffer functions. Second, the MSRA's optimization trajectory is clear and clean, making it suited to practical engineering optimization problems like locating the highest magnetic induction intensity in the three-dimensional magnetic field space. Third, based on the characteristics and complexity of the computational object, the convergence rate, visitation density, and number of cycles of the MSRA can be simply tweaked to improve the accuracy and application breadth. The MSRA is expected to have a wide range of applications.

Several problems remain unresolved: First, for the actual social production projects mentioned in the previous part of the article, we applied the MSRA to Project 1 and performed experiments to verify it. For the other three projects (searching for the location of the leakage of hazardous chemicals and toxic volatile substances, sea surface search and rescue), we had no conditions for the experimental verification. Second, it is necessary to optimize the MSRA's mathematical model and to study the mathematical models and test analyses that the MSRA applies to higher-dimensional optimization issues. Third, while planning the route, the MSRA does not have the function of avoiding obstacles. This function needs to be further developed

Author Contributions: J.Z.: conceptualization, data curation, programming, writing—original draft, writing—editing. C.J.: overall guidance, reviews, suggestions for modifications. P.C.: overall guidance, reviews, suggestions for modifications. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Srinakharinwirot University Research Fund.

Acknowledgments: The authors wish to gratefully acknowledge the financial support for this research from the Srinakharinwirot University Research fund. At the same time, we also thank the technical support given by the 2022 Guangxi university scientific research project, China. The project name is: "Research on Automatic Positioning System for Wireless Charging of Automobile Based on Maritime Search and Rescue Algorithm" (No. GuiJiaoKeYan[2022]2-2022KY1335).

Conflicts of Interest: The authors declare that they have no known competing financial interest or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Zheng, J.; Jettanasen, C. Analysis Of Mutual Inductance Between Transmitter And Receiver Coils In Wireless Power Transfer System Of Electric Vehicle. *Adv. Electr. Electron. Eng.* **2023**, *21*. [CrossRef]
2. Xu, P.; Fu, C.; Li, J.-J.; Dong, L.; Zhao, S.-P. The research of estimating the location of radioactive sources using the Bayesian estimation. *Nucl. Instruments Methods Phys. Res. Sect. Accel. Spectrometers Detect. Assoc. Equip.* **2021**, *1006*, 165405. [CrossRef]
3. Ma, D.; Mao, W.; Tan, W.; Gao, J.; Zhang, Z.; Xie, Y. Emission source tracing based on bionic algorithm mobile sensors with artificial olfactory system. *Robotica* **2022**, *40*, 976–996. [CrossRef]
4. Serra, M.; Sathe, P.; Rypina, I.; Kirincich, A.; Ross, S.D.; Lermusiaux, P. Search and rescue at sea aided by hidden flow structures. *Nat. Commun.* **2020**, *11*, 2525. [CrossRef] [PubMed]
5. Li, S.; Chen, H.; Wang, M.; Heidari, A.A.; Mirjalili, S. Slime Mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* **2020**, *111*, 300–323. [CrossRef]
6. Xue, J.; Shen, B. A novel swarm intelligence optimization approach: Sparrow search algorithm. *Syst. Sci. Control. Eng.* **2020**, *8*, 22–34. [CrossRef]
7. Zervoudakis, K.; Tsafarakis, S. A mayfly optimization algorithm. *Comput. Ind. Eng.* **2020**, *145*, 106559. [CrossRef]
8. Arora, S.; Singh, S. Butterfly optimization algorithm: A novel approach for global optimization. *Soft Comput.* **2019**, *23*, 715–734. [CrossRef]
9. Feng, Y.; Deb, S.; Wang, G.G.; Alavi, A.H. Monarch butterfly optimization: A comprehensive review. *Expert Syst. Appl.* **2021**, *168*, 114418. [CrossRef]
10. Peña-Delgado, A.F.; Peraza-Vázquez, H.; Almazán-Covarrubias, J.H.; Torres Cruz, N.; García-Vite, P.M.; Morales-Cepeda, A.B.; Ramirez-Arredondo, J.M. A Novel Bio-Inspired Algorithm Applied to Selective Harmonic Elimination in a Three-Phase Eleven-Level Inverter. *Math. Probl. Eng.* **2020**, *2020*, 8856040. [CrossRef]
11. Lin, Z.; Yue, M.; Chen, G.; Sun, J. Path planning of mobile robot with PSO-based APF and fuzzy-based DWA subject to moving obstacles. *Trans. Inst. Meas. Control.* **2022**, *44*, 121–132. [CrossRef]
12. Li, X.; Wu, D. An Improved Method of Particle Swarm Optimization for Path Planning of Mobile Robot. *J. Control. Sci. Eng.* **2020**, *2020*, 3857894. [CrossRef]
13. Song, B.; Wang, Z.; Zou, L. An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve. *Appl. Soft Comput.* **2021**, *100*, 106960. [CrossRef]
14. Liu, H.; Chen, Z.; Tian, Y.; Wang, B.; Yang, H.; Wu, G. Evaluation method for helicopter maritime search and rescue response plan with uncertainty. *Chin. J. Aeronaut.* **2021**, *34*, 493–507. [CrossRef]
15. Jamil, M.; Yang, X.S.; Zepernick, H.J. 8—Test Functions for Global Optimization: A Comprehensive Survey. In *Swarm Intelligence and Bio-Inspired Computation*; Elsevier: Amsterdam, The Netherlands, 2013; pp. 193–222. [CrossRef]
16. Locatelli, M. A Note on the Griewank Test Function. *J. Glob. Optim.* **2003**, *25*, 169–174. [CrossRef]
17. Surjanovic, S.; Bingham, D. Virtual Library of Simulation Experiments: Test Functions and Datasets. Available online: http://www.sfu.ca/~ssurjano (accessed on 15 November 2023).
18. Acampora, G.; Schiattarella, R.; Vitiello, A. Using quantum amplitude amplification in genetic algorithms. *Expert Syst. Appl.* **2022**, *209*, 118203. [CrossRef]
19. Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*; Addison-Wesley Professional: Boston, MA, USA, 1989; ISBN 978-0-201-15767-3.
20. Darwin, C. *The Origin of Species*, 1st ed.; Prakash Books India Pvt Ltd.: New Delhi, India, 2013; ISBN 13 978-8172344887.
21. Zhong, R.Y.; Xu, X. Intelligent Manufacturing in the Context of Industry 4.0: A Review. *Engineering* **2017**, *3*, 616–630. [CrossRef]