Article

A Hybrid Whale Optimization Algorithm for Global Optimization

Chun-Yao Lee * and Guang-Lin Zhuo

Department of Electrical Engineering, Chung Yuan Christian University, No. 200, Zhongbei Road, Zhongli District, Taoyuan City 320, Taiwan; s10528245@cycu.org.tw

* Correspondence: cyl@cycu.edu.tw; Tel.: +886-3-2654827

Abstract: This paper proposes a hybrid whale optimization algorithm (WOA) that is derived from the genetic and thermal exchange optimization-based whale optimization algorithm (GWOA-TEO) to enhance global optimization capability. First, the high-quality initial population is generated to improve the performance of GWOA-TEO. Then, thermal exchange optimization (TEO) is applied to improve exploitation performance. Next, a memory is considered that can store historical best-so-far solutions, achieving higher performance without adding additional computational costs. Finally, a crossover operator based on the memory and a position update mechanism of the leading solution based on the memory are proposed to improve the exploration performance. The GWOA-TEO algorithm is then compared with five state-of-the-art optimization algorithms on CEC 2017 benchmark test functions and 8 UCI repository datasets. The statistical results of the CEC 2017 benchmark test functions show that the GWOA-TEO algorithm has good accuracy for global optimization. The classification results of 8 UCI repository datasets also show that the GWOA-TEO algorithm has competitive results with regard to comparison algorithms in recognition rate. Thus, the proposed algorithm is proven to execute excellent performance in solving optimization problems.

Citation: Lee, C.-Y.; Zhuo, G.-L. A Hybrid Whale Optimization Algorithm for Global Optimization. *Mathematics* **2021**, *9*, 1477. <https://doi.org/10.3390/math9131477>

Academic Editor: Inmaculada Rodríguez Martín

Received: 28 May 2021
Accepted: 22 June 2021
Published: 24 June 2021

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

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: whale optimization algorithm; genetic algorithm; thermal exchange optimization; optimization problems

1. Introduction

With the rapid development of technology and higher requirements for product quality, a large number of practical optimization problems has arisen in various fields. However, most optimization problems are NP-hard problems and cannot be solved by polynomials in a reasonable time [1]. Therefore, researchers have proposed many meta-heuristics, which can find a good enough solution in a reasonable time, effectively applying it to the optimization problem. Meta-heuristic algorithms are group-heuristic search strategies and have the advantages of easy implementation, simple conception, and wide application to different problems in different fields. Meta-heuristic algorithms are inspired by biological or physical phenomena in nature and solve optimization problems by mimicking their behavior. In recent years, numerous meta-heuristic algorithms have been applied to solve complex optimization problems in various fields, and these algorithms can be divided into three categories, evolutionary algorithm, physical algorithm, and swarm intelligence algorithm. The commonly used evolutionary algorithms are differential evolution (DE) [2], evolutionary programming (EP) [3], genetic algorithm (GA) [4], evolutionary strategy (ES) [5] and state transition algorithm (STA) [6]. The commonly used physical algorithms are simulated annealing (SA) [7], gravitational search algorithm (GSA) [8], charged system search (CSS) [9], water evaporation optimization (WEO) [10], and thermal exchange optimization (TEO) [11]. The commonly used swarm intelligence algorithms are particle swarm optimization (PSO) [12], bat algorithm (BA) [13], ant lion optimizer (ALO) [14], chicken swarm optimization (CSO) [15], seagull optimization algorithm (SOA) [16] and whale optimization algorithm (WOA) [17].

WOA is a new swarm intelligence algorithm proposed by Mirjalili and Lewis and has the advantages of low internal parameters and easy implementation. In the literature [17], it has been demonstrated that the WOA is better than many classical algorithms, such as GSA, PSO, DE, due to its evolution strategy with covariance matrix adaptation (CMA-ES) [18], improved harmony search (HS) [19], ray optimization (RO) [20] and fast evolutionary programming (FEP) [21]. The WOA has achieved good results in many optimization problems, such as the DNA fragment assembly problem [22], resource allocation problems in wireless networks [23], forecasting gold price fluctuations [24], connection weights in neural networks [25], optimal reactive power dispatch problems [26], maximum power point tracking (MPPT) [27], hyperparameters of convolutional neural networks (CNN) [28], photovoltaic model identification [29], fuzzy programming [30], design of the proportional-integral (PI), and controller [31] and feature selection [32].

However, swarm intelligence algorithms, including WOA, generally have the following shortcomings: falling easily into local optima, low accuracy, and slow convergence speed. Hybrid algorithms deal with the above-mentioned shortcomings by combining the advantages of each algorithm, and further improve the performance of the algorithm [33,34]. Patwal et al. [35] proposed a hybrid PSO, in which the advantage of mutation strategy (MS) is embedded in the time varying acceleration coefficient (TVAC)-PSO, named TVAC-PSO-MS. The algorithm uses a three-stage mutation strategy, including Cauchy, Gaussian, and opposition, so that the algorithm has a higher probability of escaping the local optimum, a higher accuracy, and maintains the balance between exploration and exploitation capabilities. In recent years, scholars have proposed several hybrid WOA to further enhance its performance in solving optimization problems. Ling et al. [36] proposed a WOA based on the Lévy flight trajectory. Lévy flight helps to increase the diversity of the population, prevent premature convergence, and enhance the ability to escape the local optimum. Xiong et al. [37] proposed an improved WOA (IWOA). IWOA developed two prey search strategies to effectively balance local exploitation and global exploration. Sun et al. [38] proposed an adaptive weight WOA, which enhanced local optimization capabilities by introducing adaptive weights and improved the convergence accuracy of WOA. Elaziz et al. [39] proposed opposition-based learning WOA to enhance the exploration of the search space. Although these WOA variants have successfully improved performance, they still have shortcomings, such as low convergence speed and convergence accuracy [36,38,39], low exploitation capability [39] and falling into local optimum [37,38]. This motivates the development of models that better solve optimization problems in various fields.

In this paper, a genetic and thermal exchange optimization-based whale optimization algorithm (GWOA-TEO) is proposed. The WOA can better balance exploitation and exploration; the GA algorithm has good exploration ability, while the TEO algorithm has a strong exploitation ability. The proposed algorithm can combine the advantages of each algorithm to further improve the exploration and exploitation capabilities of the WOA algorithm and escape from the local optimum. The main contributions of this article are as follows. First of all, the initial population generation mechanism combined with the meta-heuristic algorithm can improve the convergence accuracy of the algorithm. Second, combining the position update mechanism of GA and TEO can solve the shortcomings of low exploitation and exploration capabilities in WOA. Finally, a memory is considered that can store some historical best solutions and perform a memory check mechanism in each iteration. If this is the same as the historical best solution, the mechanism is implemented to escape the local optimum. Through this memory mechanism, the local optimal avoidance can be improved further.

The remainder of this paper is summarized as follows. Section 2 introduces the original WOA, TEO, and GA procedures to better understand the advantages of the proposed algorithm. Section 3 presents the procedure of the GWOA-TEO algorithm. Section 4 discusses two experimental studies of CEC 2017 benchmark functions and UCI

repository datasets. Section 5 defines the advantages and limitations of the proposed algorithm. Finally, Section 6 explains the conclusions.

2. Methods

2.1. Whale Optimization Algorithm

The WOA algorithm mimics the hunting behavior of humpback whales. The algorithm has two phases to mimic these behaviors, the exploitation phase and the exploration phase. The exploitation phases are encircling prey and bubble-net attacking method, and the exploration phase is search for prey.

2.1.1. Exploitation Phases

Equations (1) and (2) mimic the behavior of humpback whales encircling prey. In the algorithm, the leading solution is supposed as the target prey, and other solutions will try to close on the target prey.

$$\vec{U} = \left| \vec{\beta} \cdot \vec{X}_b(t) - \vec{X}(t) \right| \quad (1)$$

$$\vec{X}(t+1) = \left| \vec{X}_b(t) - \vec{\alpha} \cdot \vec{U} \right| \quad (2)$$

where t is the current iteration, $\vec{X}_b$ is the best-so-far solution, $\vec{\alpha}$ and $\vec{\beta}$ are coefficient vectors that are shown as follows:

$$\vec{\alpha} = 2\vec{a} \cdot \vec{r} - \vec{a} \quad (3)$$

$$\vec{\beta} = 2 \cdot \vec{r} \quad (4)$$

where $\vec{a}$ is a linear decrease coefficient, which will decrease from 2 to 0 in the iteration process, and $\vec{r}$ is a random vector in $(0, 1)$. The different positions of the current position relative to the best-so-far solution are controlled by adjusting the values of the $\vec{\alpha}$ and $\vec{\beta}$ vectors.

Based on Equation (2) the algorithm assumes that the best-so-far solution is the prey, updates the current position of the humpback whale to the position near the prey and simulates the situation of surrounding the prey.

To mimic the bubble-net attacking of humpback whales, two mathematical models are proposed as follows:

1. Shrinking encircling mechanism: This model is implemented by linearly decreasing the value of vector $\vec{a}$. Based on the vector $\vec{a}$ and the random vector $\vec{r}$, the fluctuation range of the coefficient vector $\vec{\alpha}$ is between $(-\vec{\alpha}, \vec{\alpha})$, where $\vec{a}$ is reduced from 2 to 0 in the iteration process.
2. Spiral updating position: The model first calculates the distance between itself and the prey, and then the humpback whale surrounds the prey in a logarithmic spiral motion, the mathematical model is shown as follows:

$$\vec{X}(t+1) = \vec{U}' \times e^{\gamma\psi} \times \cos(2\pi\psi) + \vec{X}_b(t) \quad (5)$$

where $\vec{U}' = \left| \vec{X}_b(t) - \vec{X}(t) \right|$ represents the distance between the prey and the humpback whale, γ is a constant for defining the shape of the logarithmic spiral, and ψ is a random value in $(-1, 1)$.

In the exploitation phase, when the position of the prey is determined, the humpback whale will dive deep, and then begin to form spiral bubbles around the prey and move upstream toward the surface. The humpback whale gradually shrinks into the circle while simultaneously hunting prey along a spiral path. The hunting behavior assumes that the shrinking circle and the spiral-shaped path have the same implementation probability to update the position of the humpback whale in the iteration process.

2.1.2. Exploration Phase

In the exploration phase, the algorithm forces the solution to be far from the current best solution and randomly explores the search space. For this reason, the WOA algorithm uses a random value of the $\vec{\alpha}$ vector greater than +1 or less than −1, and randomly selects the reference solution. This mechanism and $|\vec{\alpha}| > 1$ allow the algorithm to perform global exploration. The mathematical model is as follows:

$$\vec{U} = \left| \vec{\beta} \cdot X_{rand} - \vec{X} \right| \quad (6)$$

$$\vec{X}(t+1) = X_{rand} - \vec{\alpha} \cdot \vec{U} \quad (7)$$

where X_{rand} is the position vector of the solution randomly chosen from the current population.

2.1.3. Overview of WOA

In the WOA algorithm, the adaptive change of the coefficient vector $\vec{\alpha}$ allows the WOA algorithm to smoothly transition between exploration and exploitation: exploration is performed when $|\vec{\alpha}| > 1$, and exploitation is performed when $|\vec{\alpha}| < 1$. During the exploitation phase, WOA can switch between shrinking circle and spiral-shaped path. The WOA has only two main internal parameters, $\vec{\alpha}$ and $\vec{\beta}$, that need to be adjusted.

2.2. Thermal Exchange Optimization

The TEO is inspired by Newton's law of cooling, i.e., the rate of heat loss of an object is proportional to the difference in temperatures between the object and its surroundings.

In the TEO algorithm, some solutions are defined as cooling objects, and others represent the environment. The environmental temperature modified by the previous temperature of the solution is defined as follows:

$$T_i^{env.} = (1 - (c_1 + c_2 \times (1 - t)) \times Rand) \times T_i^{env.} \quad (8)$$

$$t = \frac{Iter}{MaxIter} \quad (9)$$

where c_1 and c_2 are the controlling variables, $T_i^{env.}$ is the previous temperature of the solution, $Iter$ is the current iteration, and $MaxIter$ is the maximum number of iterations.

According to $T_i^{env.}$, the temperature update equation of each solution is defined as follows:

$$T_i^{new} = T_i^{env.} + (T_i^{old} - T_i^{env.}) \exp(-\varepsilon t) \quad (10)$$

$$\varepsilon = \frac{cost(object)}{cost(worst\ object)} \quad (11)$$

where when an object has lower ε , it changes the temperature slightly.

To improve the capability of escaping from local optimum traps, the parameter Pro , is introduced, and it is specified whether a component of each cooling object must be changed or not. If $Ran(i)$ ($i = 1, 2, \dots, n$) $< Pro$, one dimension of the i th object is selected randomly, and its value is regenerated as follows:

$$T_{i,j} = T_{j,min} + Rand(T_{j,max} - T_{j,min}) \quad (12)$$

where $T_{i,j}$ is the j th variable of the i th solution. $T_{j,max}$ and $T_{j,min}$ are the upper and lower bounds of the j th variable, respectively.

2.3. Crossover Operator

GA is a well-known evolutionary algorithm used to solve the optimization problem. The GA algorithm applies the following genetic operators: selection, crossover, mutation, and replacement. First, two chromosomes are selected based on their fitness value, and then the crossover operator is applied to combine the two parents to create offspring chromosomes. The mutation operator is applied to create localized change in offspring. Finally, the replacement operator eliminates insufficient fitness value chromosomes from the population. Therefore, the fittest chromosomes will create more offspring and converge towards the global optimum.

3. Genetic Whale Optimization Algorithm–Thermal Exchange Optimization for Global Optimization Problem

This paper proposes five mechanisms to enhance the WOA algorithm. Based on GA, solutions with high potential are selected using crossover operation to generate more effective solutions in order to improve the global exploration capability of the algorithm. Based on TEO, the position update equation of the TEO algorithm will be applied and replaces the original equation. The TEO algorithm's thermal memory (*TM*) is also combined with the proposed algorithm to save some historical-best-so-far solutions. Based on these two mechanisms, the TEO algorithm can improve the local exploitation capability of the algorithm.

3.1. Generate a High-Quality Initial Population

In the global optimization problem, if there are some solutions near the region that contains the global optimum, the performance of global optimization algorithms can be improved. This paper proposes a mechanism which combines the local-search algorithm to generate high-quality initial populations. First, a larger initial population is generated to cover the search space completely, and then the best part is evaluated and selected as the population of the algorithm. This step can find one or more potential areas that contain the global optimum. Second, the position update equation of the TEO algorithm is used as a local-search algorithm to explore the region near the population. This step can increase the probability of the population approach to the global optimum. Figures 1 and 2 show 100 solutions for solving the Shifted and Rotated Expanded Scaffer's F6 function in the CEC 2017 benchmark test function [38], which are generated by a random initial population and a high-quality initial population.

Figure 1. Random initial population.

Figure 2. High-quality initial population.

3.2. Improvement of the Exploitation Phase

Basic WOA proposes two mathematical formulas in the local exploitation phase, the encircling prey formula and the spiral updating position. However, in the encircling prey formula, the solution only follows the best solution in the population. When this falls into the local optimum, the other solutions will also do so. Therefore, it is necessary to further improve the algorithm exploitation capabilities. The proposed algorithm combines a local search algorithm (TEO) which has powerful exploitation capabilities. In this article, the position update equation of the TEO algorithm will replace the prey-around formula.

3.3. The Thermal Memory of the TEO Algorithm

Inspired by TEO, a memory that saves some historical-best-so-far solutions can improve the algorithm's performance without adding additional computational costs. In this paper, the *TM* is adopted and combined with the proposed algorithm. During the iterations, the new position of solutions will be compared with the *TM*. When the new position matches the *TM*, this may indicate that the overall evolution is sluggish, and the position update mechanism will be implemented to search the new area in the search space.

3.4. The Crossover Operator Mechanism

In the GA, the parent of the offspring with a better fitness value is selected to generate an offspring of solutions through evolutionary operators, including crossover, mutation, and elitism. The crossover has a powerful global search capability, which can explore new areas in the search space. This study uses a crossover operator with three random cutting points to create a new solution. First, a pair of solutions are randomly selected from the historical best solution stored in the *TM*. Next, the crossover operator creates two new solutions by cutting the pair of solutions at three random points and swapping segments. Then, one of the new solutions is randomly selected as the current solution. Figure 3 illustrates the crossover operator with three cutting points.

Figure 3. The crossover operator with three cutting points.

3.5. Update Leading Solution Based on Thermal Memory

The solutions in the basic WOA algorithm always follow the leading solution to update the current position. However, the best-so-far solution only represents the best solution of all solutions, and it cannot indicate that the overall evolution is correct. In the problem with multi-local optimum, if the leading solution is the local optimum, the other solutions whose positions are followed by the leading solution will also fall into the local optimum. To enhance the local exploitation ability, *TM* is used in this paper. The proposed algorithm will randomly select one solution from *TM* instead of just following a current best solution.

3.6. The Proposed Algorithm GWOA-TEO

The proposed algorithm is combined with *TM* and four improvement mechanisms: the high-quality initial population, improvement for the exploitation phase, crossover operation, and update best-so-far solution. The pseudocode and procedure of the proposed algorithm are shown in Algorithm 1 and Figure 4. The implementation time *IT* is used to prevent falling into an infinite loop and the increase of a large number of computation costs. When solving low-dimensional problems, because population diversity is not high, it will take a long time to implement the *TM* mechanism. The size of the thermal memory *TM* will also affect the calculation cost because, when implementing the mechanism for escaping the local optimum, it will check whether the solution matches the memory one by one. Therefore, the appropriate *IT* value and thermal memory *TM* must be selected to optimize the calculation cost. In this article, *IT* = 20 and the size of thermal memory *TM* = 10 are the settings for all case studies.

Algorithm 1: The procedure of the proposed GWOA-TEO**Input:** Initial population X_i ($i = 1, 2, \dots, n$) // n is population size**Output:** Optimal solution X^* Initialize high-quality population X_i ($i = 1, 2, \dots, n$)

Evaluate the fitness value of each solution

Construct thermal memory TM_i ($i = 1, 2, \dots, L$) // L is size of thermal memorySet the best solution X^* **while** ($t < T$) // T is maximum number of iterationsSet the leading solution LX^* based on TM **for** each solutionUpdate a , α and β // Equations (3) and (4) $p \rightarrow$ probability of implement shrinking encircling or spiral updating**if** ($p < 0.5$)**if** ($|\alpha| < 1$)

Use Equation (11) to update the current solution

else if ($|\alpha| \geq 1$)

Use Equation (8) to update the current solution

end if**else if** ($p \geq 0.5$)

Use Equation (6) to update the current solution

end if**for** $i < L$ **if** the current solution same with TM_i

Implement the crossover operator mechanism

 $times = times + 1$ **end if****if** $times > IT$ $times = 0$; // avoid infinite loop**end if****end for****end for**

Evaluate the fitness value of each solution

Update TM if obtain the better solutionUpdate LX^* based on TM Update X^* if X^* is better solution $t = t + 1$ **end while**return X^*

Figure 4. The flowchart of the proposed GWOA-TEO.

4. Experimental Results

In this section, two case studies are discussed, and the experimental results are simulated by Intel Core i7-3930 K 3.2 Ghz, 24 GB RAM, and MATLAB 2017a.

4.1. Case Study 1: CEC 2017 Benchmark Test Functions

4.1.1. Experimental Setup

In this experimental study, CEC 2017 benchmark functions [40], including 30 test functions, are applied to verify the optimization capability of GWOA-TEO; however, the second function has been excluded due to the unstable behavior. The descriptions of the 29 benchmark functions are shown in Table 1.

Table 1. Descriptions of CEC 2017 benchmark functions.

Function		Name	$F_i^* = F_i(x^*)$
UFs	F1	Shifted and Rotated Bent Cigar function	100
	F3	Shifted and Rotated Rosenbrock's function	300
SMFs	F4	Shifted and Rotated Rastrigin's function	400
	F5	Shifted and Rotated Expanded Scaffer's F6 function	500
	F6	Shifted and Rotated Lunacek Bi_Rastrigin function	600
	F7	Shifted and Rotated Non-Continuous Rastrigin's function	700
	F8	Shifted and Rotated Levy function	800
	F9	Shifted and Rotated Schwefel's function	900
HFs	F10	HF 1 ($n = 3$)	1000
	F11	HF 2 ($n = 3$)	1100
	F12	HF 3 ($n = 3$)	1200
	F13	HF 4 ($n = 4$)	1300
	F14	HF 5 ($n = 4$)	1400
	F15	HF 6 ($n = 4$)	1500
	F16	HF 6 ($n = 5$)	1600
	F17	HF 6 ($n = 5$)	1700
	F18	HF 6 ($n = 5$)	1800
	F19	HF ($n = 6$)	1900
CFs	F20	CF 1 ($n = 3$)	2000
	F21	CF 2 ($n = 3$)	2100
	F22	CF 3 ($n = 4$)	2200
	F23	CF 4 ($n = 4$)	2300
	F24	CF 5 ($n = 5$)	2400
	F25	CF 6 ($n = 5$)	2500
	F26	CF 7 ($n = 6$)	2600
	F27	CF 8 ($n = 6$)	2700
	F28	CF 9 ($n = 3$)	2800
	F29	CF 10 ($n = 3$)	2900
	F30	CF 11 ($n = 3$)	3000

Note: UF is unimodal function, SMF is simple multimodal function, HF is hybrid function and CF is composition function.

4.1.2. Parameter Setting

The performance of GWOA-TEO is compared with other optimization algorithms such as the heap-based optimizer (HBO) [41], complex path-perceptual disturbance WOA (CP-PDWOA) [42], and the basic whale optimization algorithm (WOA). The codes of all the stated algorithms are published by their original authors. The parameter settings of the above algorithms are shown in Table 2, where the maximum number of iterations is considered as the convergence criteria.

Table 2. Parameter setting of the algorithms for case study 1.

HBO [41]	CP-PDWOA [42]	WOA [17]	GWOA-TEO
Number of solutions: 40 Number of iterations: 1282 $degree = 3$ $C = T/25$ $p_1 = 1 - t/T$ $p_2 = p_1 + [(1 - p_1)/2]$	Number of solutions: 40 Number of iterations: 1282 equal-pitch Archimedean spiral curve $a: [0, 2]$ $\gamma: 1$ $\psi: [-1, 1]$	Number of solutions: 40 Number of iterations: 1282 $a: [0, 2]$ $\gamma: 1$ $\psi: [-1, 1]$	Number of solutions: 40 Number of thermal objects memory: 10 Number of iterations: 1282 $pro = 0.3$ $a: [0, 2]$ $\gamma: 1$ $\psi: [-1, 1]$

4.1.3. Statistical Results

In this subsection, three comparative algorithms, including HBO, CP-PDWOA, and WOA are adopted to evaluate the performance of GWOA-TEO. All the functions are tested in 10 dimensions. Each algorithm evaluates all benchmark functions, including average fitness value (avg) and standard deviation (std) for 30 independent runs, and the results are shown in Table 3. The results show that GWOA-TEO has good performance, and its results are better than the other algorithms. Although GWOA-TEO better HBO, GWOA-TEO has comparable results to HBO in many benchmark functions. For instance, GWOA-TEO achieved the best results for F3, F4, F10, F14, F15, F21, F22, F23, F24, F27, F29, and F30. Especially in the optimization of composition functions, GWOA-TEO achieved the best results in 7 of 11 test functions. Moreover, for F4, F10, F14, F15, F21, F22, F24, F27, and F30, the performance of GWOA-TEO is better than HBO. Furthermore, for F5, F6, F8, F9, F11, F17, F20, F25, and F26, the results of GWOA-TEO do not show a significant difference to the results of HBO.

Table 3. Statistical results of the optimization algorithms.

Function	HBO [41]		CP-PDWOA [42]		WOA [17]		GWOA-TEO	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
F1	5.36×10^2	5.43×10^2	7.29×10^9	4.88×10^9	5.75×10^6	1.33×10^7	5.44×10^2	4.74×10^2
F3	3.00×10^2	1.16×10^{-1}	3.19×10^4	1.80×10^4	2.22×10^3	1.79×10^3	3.00×10^2	3.48×10^{-1}
F4	4.05×10^2	4.28×10^{-1}	8.41×10^2	2.97×10^2	4.33×10^2	4.97×10^1	4.04×10^2	2.00
F5	5.11×10^2	3.63	5.85×10^2	2.57×10^1	5.50×10^2	1.63×10^1	5.13×10^2	4.82
F6	6.00×10^2	4.8×10^{-14}	6.51×10^2	1.84×10^1	6.37×10^2	1.26×10^1	6.01×10^2	3.26×10^{-1}
F7	7.13×10^2	4.03	8.29×10^2	2.57×10^1	7.73×10^2	2.10×10^1	7.23×10^2	4.56
F8	8.09×10^2	3.93	8.76×10^2	2.23×10^1	8.40×10^2	1.56×10^1	8.12×10^2	4.82
F9	9.00×10^2	0.00	2.22×10^3	7.63×10^2	1.44×10^3	4.65×10^2	9.01×10^2	3.41
F10	1.55×10^3	1.57×10^2	2.76×10^3	4.24×10^2	1.94×10^3	2.59×10^2	1.47×10^3	1.95×10^2
F11	1.10×10^3	1.23	3.06×10^3	2.36×10^3	1.19×10^3	4.49×10^1	1.11×10^3	4.05
F12	8.17×10^4	9.11×10^4	1.13×10^8	1.43×10^8	4.53×10^6	4.80×10^6	8.29×10^4	1.53×10^5
F13	2.96×10^3	2.56×10^3	1.28×10^6	4.48×10^6	1.76×10^4	1.83×10^4	6.55×10^3	3.97×10^3
F14	1.45×10^3	7.49×10^1	5.11×10^3	4.11×10^3	1.59×10^3	1.32×10^2	1.42×10^3	1.03×10^1
F15	1.60×10^3	2.02×10^2	3.73×10^4	6.17×10^4	6.26×10^3	5.38×10^3	1.52×10^3	2.85×10^1
F16	1.60×10^3	2.74	2.19×10^3	3.14×10^2	1.82×10^3	1.41×10^2	1.69×10^3	1.12×10^2
F17	1.70×10^3	4.30	1.89×10^3	1.05×10^2	1.78×10^3	3.50×10^1	1.72×10^3	1.37×10^1
F18	3.77×10^3	1.15×10^3	6.41×10^6	1.54×10^7	2.31×10^4	1.48×10^4	8.32×10^3	6.23×10^3
F19	1.98×10^3	1.58×10^2	4.79×10^6	1.45×10^7	4.95×10^4	9.70×10^4	3.71×10^3	1.91×10^3
F20	2.00×10^3	7.92×10^{-2}	2.32×10^3	9.56×10^1	2.17×10^3	7.97×10^1	2.03×10^3	9.80
F21	2.25×10^3	4.96×10^1	2.37×10^3	4.84×10^1	2.33×10^3	5.60×10^1	2.20×10^3	1.83
F22	2.30×10^3	4.92	3.56×10^3	7.22×10^2	2.31×10^3	1.61×10^1	2.29×10^3	2.86×10^1
F23	2.61×10^3	4.34	2.68×10^3	2.94×10^1	2.65×10^3	2.61×10^1	2.61×10^3	5.98×10^1
F24	2.63×10^3	9.50×10^1	2.82×10^3	3.15×10^1	2.77×10^3	5.02×10^1	2.55×10^3	1.04×10^2
F25	2.91×10^3	1.98×10^1	3.35×10^3	2.72×10^2	2.96×10^3	2.44×10^1	2.93×10^3	2.18×10^1
F26	2.89×10^3	6.26×10^1	4.30×10^3	5.29×10^2	3.42×10^3	5.13×10^2	2.91×10^3	8.24×10^1
F27	3.09×10^3	1.62	3.20×10^3	6.49×10^1	3.16×10^3	5.23×10^1	3.08×10^3	2.22×10^1
F28	3.17×10^3	7.37×10^1	3.67×10^3	1.19×10^2	3.28×10^3	9.42	3.23×10^3	5.23
F29	3.18×10^3	1.37×10^1	3.55×10^3	1.99×10^2	3.34×10^3	6.32×10^1	3.18×10^3	2.53×10^1
F30	2.57×10^4	1.46×10^4	8.52×10^6	7.07×10^6	2.91×10^5	6.11×10^5	2.15×10^4	2.22×10^4

Note: the algorithm achieves better results are bold in each function.

4.2. Case Study 2: UCI Repository Datasets

4.2.1. Experimental Setup

Eight standard datasets from the UCI data repository are performed in this case study to evaluate the optimization performance of the proposed algorithm [43]. The details of the selected datasets involving the number of features, instances, and classes in each dataset are shown in Table 4.

Table 4. Details of 8 repository datasets.

Datasets	Features	Instances	Classes
Vowel	10	990	11
Wine	13	178	3
CongressEW	16	435	2
Vehicle	18	94	4
WDBC	30	569	2
Ionosphere	34	351	2
krvskp	36	3196	2
Sonar	60	208	2

4.2.2. Parameter Setting

In this case study, GWOA-TEO is considered as a wrapper approach based on the k-nearest neighbors (k-NN) classifier. The k-NN classifier with the number of nearest neighbor $k = 1$ and 10-fold cross-validation is used in this experiment to generate the best results. The proposed algorithm is compared to optimization methods including WOA, GA, and TEO. In this case study, the fitness function is used to evaluate each solution, as shown in the following equation:

$$Fitness = \sigma \gamma_R(D) + \omega \frac{|R|}{|N|} \quad (13)$$

where $\gamma_R(D)$ is the classification error rate of a used classifier. $|R|$ is the number of selected features of each solution and $|N|$ is the total number of features. σ and ω are two parameters to balance between the classification accuracy and the number of selected features.

The parameter setting of the above algorithms is shown in Table 5. In Table 2, the maximum number of iterations is considered as the convergence criteria. Each algorithm evaluates all standard datasets, including the average classification accuracy, the average number of selected features and the average computational time for 30 independent runs.

Table 5. Parameter setting of the algorithms of case study 2.

WOA [17]	GA [4]	TEO [11]	GWOA-TEO
Number of solutions: 10 Number of iterations: 100 a : [0, 2] γ : 1 ψ : [−1, 1]	Number of chromosomes: 10 Maximum number of iterations: 100 Crossover rate: 0.85 Mutation rate: 0.01 Elitism rate: 0.05	Number of thermal objects: 10 Number of thermal objects memory: 2 Number of iterations: 100 $pro = 0.3$ c_1 : {0 or 1} c_2 : {0 or 1}	Number of solutions: 10 Number of thermal objects memory: 10 Number of iterations: 100 $pro = 0.3$ a : [0, 2] γ : 1 ψ : [−1, 1]

4.2.3. Comparison with GWOA-TEO, WOA, TEO, and GA

The performances of the WOA, GA, TEO, and GWOA-TEO for average classification accuracy, average number of selected features, and average computational time for 30 independent runs are compared in this subsection. The convergence curves from the best-so-far solution for each dataset from the enumerated algorithms are also compared in this subsection. The convergence curve of eight repository datasets is shown in Figure 5. From the convergence curve of Figure 5, GWOA-TEO has a better initial population and achieves higher classification accuracy, especially for WDBC, Ionosphere, krvsdp and Sonar. For Vehicle, Ionosphere, krvsdp and Sonar, the GWOA-TEO shows strong exploitation capability so that the best-so-far solution value keeps decreasing in the iteration.

Figure 5. Convergence curves for the algorithms in case study 2.

From Table 6, the results show that the GWOA-TEO is valid in the feature selection field. The GWOA-TEO has the best average classification accuracy for eight adopted

datasets. The WOA has better average classification accuracy than the TEO on six of eight datasets, including Vowel, Vehicle, WDBC, Ionosphere, krvsdp, and Sonar. However, in the less-feature-dataset case, such as Vowel, Wine, and CongressEW, the average classification accuracies of the TEO are better or the same as those of the WOA. These results show that TEO has better local search capability than WOA, while in the multi-feature-dataset cases, such as Vehicle, Ionosphere, and Sonar, the GA has better average classification accuracy than the WOA. These results also show that GA has a better global search capability than WOA. The proposed algorithm GWOA-TEO combines the advantages of TEO and GA, which achieves the best average classification accuracy in this study. As shown from the average computational time in Figure 6, GA is the best computing costs algorithm. Besides, the average calculation time of GWOA-TEO is very close to that of WOA and TEO. GWOA-TEO can maintain computing costs, and classification accuracy is better than other algorithms.

Table 6. Comparison between GWOA-TEO and the other algorithms.

Datasets	WOA		GA		TEO		GWOA-TEO	
	Avg Acc (%)	Avg No.F	Avg Acc (%)	Avg No.F	Avg Acc (%)	Avg No.F	Avg Acc (%)	Avg No.F
Vowel	98.35	9.63	98.17	7.33	98.04	7.36	99.47	9.07
Wine	97.69	9.37	97.52	7.63	97.86	7.8	99.77	8
CongressEW	94.73	2.93	95.24	6.97	94.94	8.63	96.12	3.07
Vehicle	74.33	9.47	75.74	8.9	70.95	8.5	79.15	6.1
WDBC	97.1	20.43	97.56	15.36	96.89	16.1	98.13	14.27
Ionosphere	92.64	11.47	93.47	14.17	91.53	16.43	95.37	8.6
krvsdp	95.88	26.07	94.95	22.03	89.36	19	98.04	22.6
Sonar	91.35	40.8	93.27	31.4	90.3	30.67	95.63	29.5

Figure 6. Convergence curves for the algorithms in case study 2.

4.2.4. Comparison with Published Algorithms

In this subsection, to further evaluate the GWOA-TEO algorithm's performance, a comparison is performed with other state-of-the-art feature selection algorithms. Some brief descriptions of state-of-the-art feature selection algorithms are as follows: the hybrid seagull optimization algorithm (SOA) and thermal exchange optimization (TEO) are based on improved local searchability, named SOA-TEO3 [44]; the heap-based optimizer is based on the corporate rank hierarchy (CRH) principle, named HBO [41]; the binary version of the ant lion optimizer is based on crossover operation and mutation operation to improve the

local searchability, named BALO-1 [45]; the high-level relay hybrid (HRH) model for whale optimization algorithm and simulated annealing (SA) is based on tournament selection to maintain the diversity of the population, named WOASAT-2 [46].

Table 7 compares the results of the eight standard data sets of the above-mentioned state-of-the-art feature selection algorithms. SOA-TEO3 achieves the best performance on the fewest number of selected features because the algorithm effectively strengthens the exploitation capability through TEO; however, too much exploitation may lead to local optimum. WOASAT-2 performs well on all datasets, achieving the best classification accuracy on the CongressEW dataset and the best performance on the Sonar dataset. This is related to the fact that the algorithm uses a tournament selection mechanism to reduce the probability of falling into a local optimum, extensively explores regions in the space, and uses the local search algorithm (SA) to strengthen these regions.

Table 7. Comparison with published feature selection algorithms.

Datasets	Algorithms	Avg Acc (%)	Avg No.F
Vowel	HBO [41]	99.35	9.27
	GWOA-TEO	99.47	9.07
Wine	SOA-TEO3 [44]	90.94	3.81
	HBO [41]	99.38	8.77
	BALO-1 [45]	98.90	5.77
	WOASAT-2 [46]	99.00	6.4
	GWOA-TEO	99.77	8
CongressEW	HBO [41]	96.05	4.8
	BALO-1 [45]	97.00	5.01
	WOASAT-2 [46]	98.00	6.4
	GWOA-TEO	96.12	3.07
Vehicle	HBO [41]	75.99	7.8
	GWOA-TEO	79.15	6.1
WDBC	SOA-TEO3 [44]	94.55	9.11
	HBO [41]	97.76	15.87
	BALO-1 [45]	97.90	14.82
	WOASAT-2 [46]	98.00	11.6
	GWOA-TEO	98.13	14.27
Ionosphere	SOA-TEO3 [44]	90.53	7.58
	HBO [41]	93.96	12.47
	BALO-1 [45]	88.9	14.65
	WOASAT-2 [46]	96.00	12.8
	GWOA-TEO	95.37	8.6
krvskp	HBO [41]	98.30	21.1
	BALO-1 [45]	96.70	16.45
	WOASAT-2 [46]	98.00	18.4
	GWOA-TEO	98.04	22.6
Sonar	SOA-TEO3 [44]	93.97	27.17
	HBO [41]	93.85	31.93
	BALO-1 [45]	86.8	26.58
	WOASAT-2 [46]	97.00	26.4
	GWOA-TEO	95.63	29.5

Based on the classification accuracy criterion, GWOA-TEO performs best in the comparison algorithm. On the vowel, wine, vehicle and WDBC data sets, GWOA-TEO has better classification accuracy than other published algorithms. GWOA-TEO ranks second in classification accuracy on the Ionosphere, krvsdp and Sonar datasets. The proposed algorithm strengthens exploration through GA, strengthens exploitation through TEO, and combines a thermal memory mechanism to further improve local optimal avoidance.

Finally, based on the classification accuracy criterion, GWOA-TEO has better capabilities in the field of feature selection.

5. Discussion

According to the above experimental results, the superiority of the proposed algorithm can be summarized as follows.

- (1) Improvement in the initial population: in general, the traditional meta-heuristic algorithms randomly generate the initial population. However, this randomness may lead to non-uniform and low-quality population distribution and slow convergence. Therefore, an initial population generation mechanism combined with a local exploitation algorithm is proposed in this paper. This mechanism can increase the probability of the population approach being the global optimum. Figure 5 shows the convergence curve of GWOA-TEO and three comparison algorithms. The experimental results show that in seven of the eight repository data sets, including Vowel, Wine, CongressEW, WDBC, Ionosphere, krvsdp and Sonar, the improved initial population has a better fitness value than the randomly generated initial population.
- (2) Improvement in the exploration and exploitation phase: the proposed algorithm combines GA to enhance exploration capabilities and TEO to enhance exploitation capabilities. Figure 5 shows the convergence curve of GWOA-TEO and the three comparison algorithms. In the convergence curve of the WDBC dataset, GWOA-TEO shows the advantage of fast convergence. In the convergence curves of the Ionosphere and Sonar datasets, GWOA-TEO shows strong exploration and exploitation capabilities. Compared with the original WOA, WOA converges at the 58th iteration, while GWOA-TEO converges at the 70th and 76th iterations, and has higher accuracy.
- (3) Improvement in escape from local optimum: in this paper, a memory that saves some historical-best-so-far solutions is proposed to further improve the avoidance of local optimum. Composition functions are challenging problems because the functions contain randomly located global optimum and several randomly located deep local optimums. Table 3 shows the statistical results of 11 composition functions in the CEC 2017 benchmark function. The proposed algorithm is better than CP-PDWOA and has competitive performance with HBO. HBO achieved the best fitness value in six composition functions, including F20, F23, F25, F26, F28 and F29, and GWOA-TEO achieved the best fitness value in seven composition functions, including F21, F22, F23, F24, F27, F29 and F30.

Note that, despite these advantages, the proposed algorithm still has some limitations.

- (1) The robustness of the proposed algorithm: in this paper, the standard deviation is a parameter to evaluate the robustness of the model. The smaller the value of standard deviation, the more robust the model. In Table 3, although the algorithm achieves a higher accuracy rate, it only ranks second in standard deviation, better than WOA variants (CP-PDWOA) and WOA, but not as good as HBO. The proposed algorithm only achieved the best standard deviation in F14, F15, F21 and F28 among the 30 CEC 2017 benchmark functions.
- (2) Computational time and complexity: in Figure 6, the computational time of the proposed algorithm is lower than GA but higher than WOA and TEO. Although GA has powerful exploration capabilities, the implementation of the crossover operator mechanism in the proposed algorithm is the main reason for increasing the computational time. Therefore, it is necessary to study further the global exploration algorithm to reduce the complexity.
- (3) The generalization capacity of the proposed algorithm: the proposed algorithm is tested only on CEC 2017 benchmark functions and eight UCI repository datasets. The performance of the proposed algorithm on the actual optimization problems was not mentioned in this study. Therefore, the effectiveness of the GWOA-TEO algorithm still needs to be studied further.

6. Conclusions

This paper proposes an effective global optimization algorithm to solve complex optimization problems. The proposed algorithm can combine the advantages of each algorithm to further improve the exploration and exploitation capabilities of the WOA algorithm, and escape from the local optimum. The main contributions of this paper are as follows. First, an initial population generation mechanism based on the local search algorithm is proposed, which can improve the convergence accuracy of the algorithm. Secondly, combining GA (strong exploration capability) and TEO (strong exploitation capability) can solve the shortcomings of slow convergence speed and low exploitation and exploration capability of WOA. Finally, the concept of memory is introduced, and a memory check mechanism is performed in each iteration. If it is the same as the historical best solution, a mechanism is implemented to escape from the local optimum. The concept of memory is introduced, which can further improve local optimal avoidance. The experimental results based on the CEC 2017 benchmark function show that, compared with the three optimization algorithms HBO, CP-PDWOA, and WOA, the GWOA-TEO algorithm has good global optimization capabilities. Especially in the 11 combinatorial functions of the challenging optimization task, the proposed algorithm shows almost the same optimization results as HBO. Experimental results based on eight UCI knowledge base data sets show that, compared with the traditional three optimization algorithms, GWOA-TEO achieves the best classification accuracy and maintains a reasonable calculation time. Compared with some of the most advanced feature selection algorithms (including SOA-TEO3, HBO, BALO-1 and WOASAT-2), the proposed algorithm exhibits competitive results in classification accuracy and achieves the best classification accuracy in four datasets. However, the proposed algorithm still has some limitations that have been mentioned in the discussion chapter: (1) the robustness, computational time and complexity of the proposed algorithm still need to be further improved; (2) the generalization ability of the proposed algorithm has not been tested. Therefore, it is necessary to further study the global exploration algorithm to verify its effectiveness.

Author Contributions: Resources, C.-Y.L.; visualization, C.-Y.L. and G.-L.Z.; validation, G.-L.Z.; supervision, C.-Y.L.; methodology, C.-Y.L. and G.-L.Z.; software, C.-Y.L.; data curation, G.-L.Z.; writing—original draft preparation, G.-L.Z.; writing—review and editing, C.-Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Abdel-Basset, M.; Abdel-Fatah, L.; Sangaiah, A.K. Metaheuristic Algorithms: A Comprehensive Review. In *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*; Elsevier: Amsterdam, The Netherlands, 2018; pp. 185–231.
2. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [\[CrossRef\]](#)
3. Lee, K.Y.; Yang, F.F. Optimal reactive power planning using evolutionary algorithms: A comparative study for evolutionary programming, evolutionary strategy, genetic algorithm, and linear programming. *IEEE Trans. Power Syst.* **1998**, *13*, 101–108. [\[CrossRef\]](#)
4. Holland, J.H. Genetic algorithms. *Sci. Am.* **1992**, *267*, 66–72. [\[CrossRef\]](#)
5. Rechenberg, I. *Evolutionsstrategien*; Springer: Berlin/Heidelberg, Germany, 1978; Volume 8, pp. 83–114.
6. Zhou, X.; Yang, C.; Gui, W. State transition algorithm. *J. Ind. Manag. Optim.* **2012**, *8*, 1039–1056. [\[CrossRef\]](#)
7. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [\[CrossRef\]](#) [\[PubMed\]](#)
8. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. GSA: A gravitational search algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [\[CrossRef\]](#)
9. Kaveh, A.; Talatahari, S. A novel heuristic optimization method: Charged system search. *Acta Mech.* **2010**, *213*, 267–289. [\[CrossRef\]](#)

10. Kaveh, A.; Bakhshpoori, T. Water evaporation optimization: A novel physically inspired optimization algorithm. *Comput. Struct.* **2016**, *167*, 69–85. [\[CrossRef\]](#)
11. Kaveh, A.; Dadras, A. A novel meta-heuristic optimization algorithm: Thermal exchange optimization. *Adv. Eng. Softw.* **2017**, *110*, 69–84. [\[CrossRef\]](#)
12. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
13. Yang, X.-S. A new metaheuristic bat-inspired algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*; Springer: New York, NY, USA, 2010; pp. 65–74.
14. Mirjalili, S. The ant lion optimizer. *Adv. Eng. Softw.* **2015**, *83*, 80–98. [\[CrossRef\]](#)
15. Meng, X.; Liu, Y.; Gao, X.; Zhang, H. A new bio-inspired algorithm: Chicken swarm optimization. In *Advances in Swarm Intelligence (Lecture Notes in Computer Science)*; Springer: Cham, Switzerland, 2014; Volume 8794, pp. 86–94.
16. Dhiman, G.; Kumar, V. Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. *Knowl.-Based Syst.* **2018**, *165*, 169–196. [\[CrossRef\]](#)
17. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [\[CrossRef\]](#)
18. Hansen, N.; Müller, S.D.; Koumoutsakos, P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *E Comput.* **2003**, *11*, 1–18. [\[CrossRef\]](#)
19. Mahdavi, M.; Fesanghary, M.; Damangir, E. An improved harmony search algorithm for solving optimization problems. *Appl. Math. Comput.* **2007**, *188*, 1567–1579. [\[CrossRef\]](#)
20. Kaveh, A.; Khayatizad, M. A new meta-heuristic method: Ray Optimization. *Comput. Struct.* **2012**, *112*, 283–294. [\[CrossRef\]](#)
21. Yao, X.; Liu, Y.; Lin, G.M. Evolutionary programming made faster. *IEEE Trans. E Comput.* **1999**, *3*, 82–102.
22. Abdel-Basset, M.; Mohamed, R.; Sallam, K.M.; Chakraborty, R.K.; Ryan, M.J. An Efficient-Assembler Whale Optimization Algorithm for DNA Fragment Assembly Problem: Analysis and Validations. *IEEE Access* **2020**, *8*, 222144–222167. [\[CrossRef\]](#)
23. Pham, Q.-V.; Mirjalili, S.; Kumar, N.; Alazab, M.; Hwang, W.-J. Whale optimization algorithm with applications to resource allocation in wireless networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 4285–4297. [\[CrossRef\]](#)
24. Alameer, Z.; Elaziz, M.A.; Ewees, A.; Ye, H.; Jianhua, Z. Forecasting gold price fluctuations using improved multilayer perceptron neural network and whale optimization algorithm. *Resour. Policy* **2019**, *61*, 250–260. [\[CrossRef\]](#)
25. Aljarah, I.; Faris, H.; Mirjalili, S. Optimizing connection weights in neural networks using the whale optimization algorithm. *Soft Comput.* **2016**, 1–15. [\[CrossRef\]](#)
26. Medani, K.B.O.; Sayah, S.; Bekrar, A. Whale optimization algorithm based optimal reactive power dispatch: A case study of the Algerian power system. *Electr. Power Syst. Res.* **2018**, *163*, 696–705. [\[CrossRef\]](#)
27. Cherukuri, S.K.; Rayapudi, S.R. A novel global MPP tracking of photovoltaic system based on whale optimization algorithm. *Int. J. Renew. Energy Dev.* **2016**, *5*, 225–232.
28. Dixit, U.; Mishra, A.; Shukla, A.; Tiwari, R. Texture classification using convolutional neural network optimized with whale optimization algorithm. *SN Appl. Sci.* **2019**, *1*, 655. [\[CrossRef\]](#)
29. Elazab, O.S.; Hasanien, H.M.; Elgendy, M.A.; Abdeen, A.M. Whale optimisation algorithm for photovoltaic model identification. *J. Eng.* **2017**, *2017*, 1906–1911. [\[CrossRef\]](#)
30. Ghahremani-Nahr, J.; Kian, R.; Sabet, E. A robust fuzzy mathematical programming model for the closed-loop supply chain network design and a whale optimization solution algorithm. *Expert Syst. Appl.* **2019**, *116*, 454–471. [\[CrossRef\]](#)
31. Hasanien, H.M. Performance improvement of photovoltaic power systems using an optimal control strategy based on whale optimization algorithm. *Electr. Power Syst. Res.* **2018**, *157*, 168–176. [\[CrossRef\]](#)
32. Hussien, A.G.; Hassanien, A.E.; Houssein, E.H.; Bhattacharyya, S.; Amin, M. S-shaped binary whale optimization algorithm for feature selection. In *Recent Trends in Signal and Image Processing*; Springer: Singapore, 2018; Volume 727, pp. 79–87.
33. Garg, H. A Hybrid GA-GSA Algorithm for Optimizing the Performance of an Industrial System by Utilizing Uncertain Data. In *Handbook of Research on Artificial Intelligence Techniques and Algorithms*; Pandian, V., Ed.; IGI Global: Hershey, PA, USA, 2015; pp. 620–654.
34. Garg, H. A hybrid PSO-GA algorithm for constrained optimization problems. *Appl. Math. Comput.* **2016**, *274*, 292–305. [\[CrossRef\]](#)
35. Patwal, R.S.; Narang, N.; Garg, H. A novel TVAC-PSO based mutation strategies algorithm for generation scheduling of pumped storage hydrothermal system incorporating solar units. *Energy* **2018**, *142*, 822–837. [\[CrossRef\]](#)
36. Ling, Y.; Zhou, Y.; Luo, Q. Lévy flight trajectory-based whale optimization algorithm for global optimization. *IEEE. Access* **2017**, *5*, 6168–6186. [\[CrossRef\]](#)
37. Xiong, G.; Zhang, J.; Yuan, X.; Shi, D.; He, Y.; Yao, G. Parameter extraction of solar photovoltaic models by means of a hybrid differential evolution with whale optimization algorithm. *Sol. Energy* **2018**, *176*, 742–761. [\[CrossRef\]](#)
38. Sun, W.; Zhang, C.C. Analysis and forecasting of the carbon price using multi resolution singular value decomposition and extreme learning machine optimized by adaptive whale optimization algorithm. *Appl. Energy* **2018**, *231*, 1354–1371. [\[CrossRef\]](#)
39. Abd Elaziz, M.; Oliva, D. Parameter estimation of solar cells diode models by an improved opposition-based whale optimization algorithm. *Energy Convers. Manag.* **2018**, *171*, 1843–1859. [\[CrossRef\]](#)

-
40. Awad, N.H.; Ali, M.Z.; Liang, J.J.; Qu, B.Y.; Suganthan, P.N. Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization. *Tech. Rep.* **2016**, 201611. Available online: https://www.researchgate.net/publication/317228117_Problem_Definitions_and_Evaluation_Criteria_for_the_CEC_2017_Competition_and_Special_Session_on_Constrained_Single_Objective_Real-Parameter_Optimization (accessed on 14 March 2021).
 41. Askari, Q.; Saeed, M.; Younas, I. Heap-based optimizer inspired by corporate rank hierarchy for global optimization. *Expert Syst. Appl.* **2020**, *161*, 113702. [[CrossRef](#)]
 42. Sun, W.-Z.; Wang, J.-S.; Wei, X. An improved whale optimization algorithm based on different searching paths and perceptual disturbance. *Symmetry* **2018**, *10*, 210. [[CrossRef](#)]
 43. UCI Machine Learning Repository. Available online: <http://archive.ics.uci.edu/ml> (accessed on 14 March 2021).
 44. Jia, H.; Xing, Z.; Song, W. A new hybrid seagull optimization algorithm for feature selection. *IEEE Access* **2019**, *7*, 49614–49631. [[CrossRef](#)]
 45. Emary, E.; Zawbaa, H.M.; Hassanien, A.E. Binary ant lion approaches for feature selection. *Neurocomputing* **2016**, *213*, 54–65. [[CrossRef](#)]
 46. Mafarja, M.M.; Mirjalili, S. Hybrid whale optimization algorithm with simulated annealing for feature selection. *Neurocomputing* **2017**, *260*, 302–312. [[CrossRef](#)]