mathematics

Article

Propagation Search Algorithm: A Physics-Based Optimizer for Engineering Applications

Mohammed H. Qais [1,*], Hany M. Hasanien [2], Saad Alghuwainem [3] and Ka Hong Loo [4]

1 Centre for Advances in Reliability and Safety, Hong Kong, China
2 Electrical Power and Machines Department, Faculty of Engineering, Ain Shams University, Cairo 11517, Egypt; hanyhasanien@ieee.org
3 Electrical Engineering Department, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia; saadalgh@ksu.edu.sa
4 Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong, China; kh.loo@polyu.edu.hk
* Correspondence: mohqais@cairs.hk

Abstract: For process control in engineering applications, the fewer the coding lines of optimization algorithms, the more applications there are. Therefore, this work develops a new straightforward metaheuristic optimization algorithm named the propagation search algorithm (PSA), stirred by the wave propagation of the voltage and current along long transmission lines. The mathematical models of the voltage and current are utilized in modeling the PSA, where the voltage and current are the search agents. The propagation constant of the transmission line is the control parameter for the exploitation and exploration of the PSA. After that, the robustness of the PSA is verified using 23 famous testing functions. The statistical tests, comprising mean, standard deviation, and *p*-values, for 20 independent optimization experiments are utilized to confirm the robustness of the PSA to find the best result and the significant difference between the outcomes of the PSA and those of the compared algorithms. Finally, the proposed PSA is applied to find the optimum design parameters of four engineering design problems, including a three-bar truss, compression spring, pressure vessel, and welded beam. The outcomes show that the PSA converges to the best solutions very quickly, which can be applied to those applications that require a fast response.

Keywords: algorithms; engineering optimization; metaheuristics; propagation search algorithm

MSC: 68W20

Citation: Qais, M.H.; Hasanien, H.M.; Alghuwainem, S.; Loo, K.H. Propagation Search Algorithm: A Physics-Based Optimizer for Engineering Applications. *Mathematics* **2023**, *11*, 4224. https://doi.org/10.3390/math11204224

Academic Editors: Petr Stodola and Janez Žerovnik

Received: 15 September 2023
Revised: 29 September 2023
Accepted: 3 October 2023
Published: 10 October 2023

1. Introduction

Metaheuristic optimization algorithms have been extensively utilized in several engineering applications to reduce costs. They have used either an optimal design or online process control. For optimal design problems, it does not pose a significant problem if the optimization algorithm is complex and takes a long time to find the best design parameters. However, for online process control, it is required that optimization algorithms have a fast response and less computation complexity. For example, energy harvesting from solar and wind resources or the optimal power flow in large electrical networks require fast online tracking of power variations, which increases system efficiency. Therefore, particle swarm optimization (PSO) has been utilized in most engineering applications because of its simplicity of coding and capability to find an optimum local or global solution. However, many metaheuristic algorithms with high computational complexity and long codes have recently emerged to achieve optimum solutions, which can be applied to optimal design problems but not in online control processes. Therefore, proposing new metaheuristic algorithms with less computational complexity is very much welcome for engineering applications.

In the literature review, many metaheuristic optimization algorithms have been stimulated by nature, biological behavior, or physical actions. For biological behavior, algorithms are inspired by different behaviors, either social communities, reproduction, food finding, or survival instinct. In 1975, Holland invented the genetic algorithm (GA), the first metaheuristic algorithm version that uses random searches to produce a new set of offspring [1]. After that, in 1995, Kennedy and Eberhart presented a new simple algorithm, the PSO algorithm, stimulated by the swarming performance of birds and fish [2]. Then, in 1995, Dorigo and Caro proposed the ant colony optimization (ACO) algorithm, which is motivated by the manners of ants to find a straight path between the colony and food position [3]. After that, many optimization algorithms emerged, such as the artificial bee colony (ABC) [4], which is stimulated by the swarming deeds of honeybees; firefly algorithm (FA) [5], which is stimulated by the blinking light of fireflies for communicating and attracting prey; and cuckoo search (CS) algorithm [6], which is stimulated by the levy walk and intrusions on the nests of other birds.

Furthermore, there are many recent metaheuristic algorithms that have been stimulated by alive creatures' behaviors, such as the grey wolf optimizer (GWO), which was stimulated by the hierarchy of guidance and hunting [7]; the whale optimization algorithm (WOA), which was stimulated by producing spiral bubbles around a school of fish [8]; the salp swarm algorithm (SSA), which was stimulated by the teeming of salps to track food [9]; Harris hawk optimization (HHO), which was stimulated by the teeming work of many hawks to attack prey [10]; the mantis search algorithm (MSA), which was inspired by the foraging process of mantises [11]; the nutcracker optimization algorithm (NOA), which was stimulated by the seasonal deeds of nutcrackers in finding, storing, and memorizing food [12]; the Aquila optimizer (AO), which was stimulated by the hunting style of Aquila [13]; the black widow optimizer (BWO), which was stimulated by the mating and flesh-eating of black widow spiders [14]; and the Tunicate swarm algorithm (TSA), which was stimulated by the swarming manners of tunicates in tracking food [15]. Consequently, many algorithms are stimulated by the conduct of living creatures, for example, dolphins [16], white sharks [17], vultures [18], orcas [19], starlings [20], rabbits [21], frogs [22], butterflies [23], hyenas [24], reptiles [25], coati [26], leopards [27], and eagles [28].

Physicists such as Newton, Einstein, etc., were attracted by the physical phenomena in the universe, which led them to find mathematical laws and paradigms after a long study. Therefore, many algorithms have been proposed based on their physical models, such as the annealing process of metals [29]; the law of gravity by Newton [30]; the law of gas by William Henry [31]; the heat transfer between materials and ambient [32]; the collision of bodies [33]; the pull and repulsion forces between atoms [34]; the laws of electrostatic and dynamic charges by Coulomb and Newton [35]; the migrant light between mediums with different intensities [36]; the transient behavior of first- and second-order electrical circuits [37]; the motion and speed of planets by Kepler [38]; the electrical trees and figures of lightning by Lichtenberg [39]; the electromagnetic field [40]; circles geometry [41]; and the electrical field [42]. Moreover, many metaheuristic algorithms were inspired by the gravity effect on the motion of planets and stars [43]; the centrifugal forces and gravity relations [44]; ion motions [45]; and the orbits of electrons around the nucleus inside atoms [46].

The research gap is defined by the no-free lunch theorem, which states that there is no single algorithm that can succeed in solving all optimization problems. In addition, the mathematical modeling and program coding of some recent metaheuristic algorithms are complex and cannot be easily applied to the online process control of engineering applications. Moreover, some models of algorithms are not investigated by related scientists. Therefore, using a studied mathematical model, simple software code, and fast convergence motivated us to propose a new metaheuristic optimizer called the propagation search algorithm (PSA), stimulated by the propagation of the waveforms of the electrical voltage and current along long transmission lines. Scientists previously offered mathematical voltage and current models at any transmission line section. Then, we adapted these models to be randomized models for random transmission lines with random impedances and admittances. The voltage and current are considered the search agents of the PSA, where they propagate based on their previous values and the propagation constant of the transmission line. These search agents rely on each other's values, which helps them to encircle the best solution.

The main contributions of this paper are summarized below:

1. Developing a new physics-based metaheuristic algorithm called the propagate search algorithm (PSA), inspired by voltage and current waveform propagation along long transmission lines.
2. Testing the PSA using the 23 famous testing functions and comparing the outcomes with eight famous metaheuristic algorithms.
3. Applying the PSA to find the optimum design of four famous engineering problems and comparing it with other metaheuristic algorithms.

The remainder of this paper is designed as follows: Section 2 describes the background, the mathematical modeling, and the flowchart of the proposed PSA; Section 3 describes the testing results; Section 4 shows the application of the proposed PSA to different engineering applications; and Section 5 presents a brief conclusion about the contribution and the results of this paper.

2. Propagation Search Algorithm

The PSA is a physics-based metaheuristic optimization algorithm that is stimulated by the propagation of electrical voltage (V) and current (I) waveforms along long transmission lines, as shown in Figure 1. The parameters of the transmission line, series impedance (z), and shunt admittance (y) play essential roles in the propagation of the voltage and current. For $y > z$, $V(x + \Delta x)$ is higher than $I(x + \Delta x)$; however, for y < z, $V(x + \Delta x)$ is lower than $I(x + \Delta x)$, and for $y = z$, $V(x + \Delta x)$ is equal to $I(x + \Delta x)$, as shown in Figure 2. In this work, we will introduce the mathematical modeling of the presented PSA based on the mathematical modeling of the propagation of the voltage and current along the transmission line.

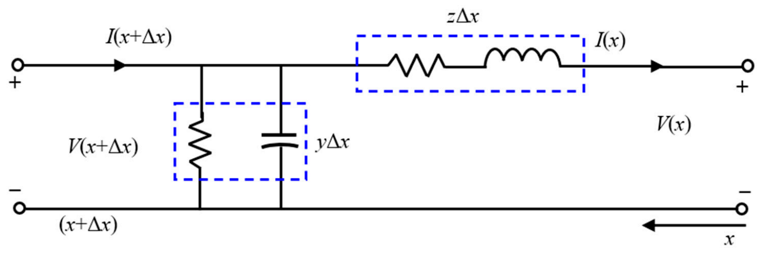

Figure 1. Line section of a long transmission line.

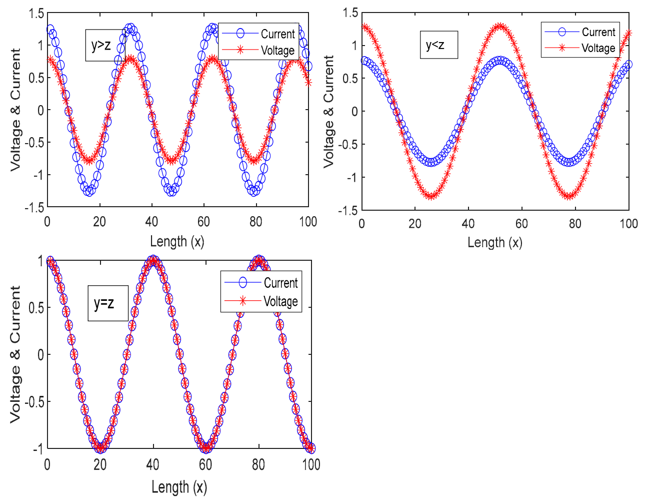

Figure 2. Effect of z and y on the trajectories of voltage and current.

2.1. Background

For a long transmission line, its impedance (z) and admittance (y) are uniformly distributed along its length and cannot be lumped. Therefore, the voltage (V) and current (I) can be calculated at each line section (Δx), as shown in Figure 1. The voltage at any distance can be calculated using Kirchhoff's voltage law as follows [47]:

$$V(x+\Delta x) = (z\Delta x)I(x) + V(x) \Rightarrow \frac{V(x+\Delta x)-V(x)}{\Delta x} = zI(x) \Rightarrow \lim_{\Delta x \to 0}\frac{V(x+\Delta x)-V(x)}{\Delta x} = \frac{dV(x)}{dx} = zI(x) \tag{1}$$

For electrical current calculation, Kirchhoff's current law can be applied as follows:

$$I(x+\Delta x) = (y\Delta x)V(x+\Delta x) + I(x) \Rightarrow \frac{I(x+\Delta x)-I(x)}{\Delta x} = yV(x+\Delta x) \Rightarrow \lim_{\Delta x \to 0}\frac{I(x+\Delta x)-I(x)}{\Delta x} = \frac{dI(x)}{dx} \approx yV(x) \tag{2}$$

where $V(x + \Delta x) \approx V(x)$, by taking the differential of (1) and substituting it in (2), we can obtain the following:

$$\frac{d^2V(x)}{dx^2} = z\frac{dI(x)}{dx} = zyV(x) \Rightarrow \frac{d^2V(x)}{dx^2} - zyV(x) = 0 \tag{3}$$

Then, the second-order linear differential equation in (3) can be solved to find $V(x)$, as follows [47]:

$$V(x) = A_1 e^{\sqrt{zy}x} + A_2 e^{-\sqrt{zy}x} \tag{4}$$

Then, to find the current expression, we will differentiate (4) and compare it with (1), as follows:

$$\frac{dV(x)}{dx} = A_1\sqrt{zy}e^{\sqrt{zy}x} - A_2\sqrt{zy}e^{-\sqrt{zy}x} = zI(x) \tag{5}$$

$$I(x) = \frac{\sqrt{zy}\left(A_1 e^{\sqrt{zy}x} - A_2 e^{-\sqrt{zy}x}\right)}{z} = \frac{A_1 e^{\gamma x} - A_2 e^{-\gamma x}}{Z_c} \tag{6}$$

where Z_c is the characteristic impedance of the transmission line, $Z_c = \frac{z}{\sqrt{zy}}$, γ is the propagation constant, $\gamma = \sqrt{zy}$, and A_1 and A_2 are constants that can be obtained by solving (6) and (4) for $V(0) = V_R$ and $I(0) = I_R$, where V_R and I_R are the receiving end voltage and current, respectively. Then, we achieve A_1 and A_2 as follows:

$$A_1 = \frac{V_R + Z_c I_R}{2} \;\&\; A_2 = \frac{V_R - Z_c I_R}{2} \tag{7}$$

Then, substitute (7) into (4) and (6) to obtain the following:

$$V(x) = \left(\frac{V_R + Z_c I_R}{2}\right)e^{\gamma x} + \left(\frac{V_R - Z_c I_R}{2}\right)e^{-\gamma x} = V_R\left(\frac{e^{\gamma x} + e^{-\gamma x}}{2}\right) + Z_c I_R\left(\frac{e^{\gamma x} - e^{-\gamma x}}{2}\right) \tag{8}$$

$$I(x) = \left(\frac{V_R + Z_c I_R}{2Z_c}\right)e^{\gamma x} - \left(\frac{V_R - Z_c I_R}{2Z_c}\right)e^{-\gamma x} = \frac{V_R}{Z_c}\left(\frac{e^{\gamma x} - e^{-\gamma x}}{2}\right) + \left(\frac{e^{\gamma x} + e^{-\gamma x}}{2}\right)I_R \tag{9}$$

Then, the hyperbolic functions can be used to represent the exponential expressions as follows:

$$V(x) = V_R\cosh(\gamma x) + Z_c I_R \sinh(\gamma x) \tag{10}$$

$$I(x) = \frac{V_R}{Z_c}\sinh(\gamma x) + I_R\cosh(\gamma x) \tag{11}$$

For long transmission lines, the series and shunt resistances can be neglected, and only inductance (L) and capacitor (C) are considered. Therefore, $z = j\omega L$ and $y = j\omega C$, so $Zc = \sqrt{L/C}$ and $\gamma = j\omega\sqrt{LC}$.

2.2. PSA Inspiration

The PSA is a physics-based metaheuristic optimization algorithm stimulated by the propagation performance of the voltage and current waveforms through a long transmission line. The search agents of the PSA imitate the behavior of the voltage and current. The mathematical model of the PSA is developed based on Equations (10) and (11) with some modifications, as in (12) and (13). The modifications assumed that $L = C$ then $Z_c = 1$ and $\gamma = C$, the propagation constant γ is expressed as a random number as in (14), the voltage V_R is expressed as X_v, and the current I_R is expressed as X_i. The length of the transmission line (l) is the shrinking variable that decreases from 1 to 0, as described in (14).

$$X_v(t+1) = \overbrace{X_v^*(t) - X_v(t)}^{V_R}\cosh(\gamma l) + \overbrace{(X_i^*(t) - X_i(t))}^{I_R}\sinh(\gamma l) \tag{12}$$

$$X_i(t+1) = \overbrace{X_i^*(t) - X_i(t)}^{I_R}\cosh(\gamma l) + \overbrace{(X_v^*(t) - X_v(t))}^{V_R}\sinh(\gamma l) \tag{13}$$

$$l = \begin{cases} 1 - \sqrt{t/T} & t < 0.5T \\ \sqrt{1 - t/T} & t \geq 0.5T \end{cases} \; \& \; \gamma = \begin{cases} r_1\sqrt{\pi r_2} & t < 0.5T \\ r_n\sqrt{\frac{\pi}{2}r_3} & t \geq 0.5T \end{cases} \tag{14}$$

where t is the current iteration; T is the total iterations; r_1, r_2, and r_3 are random numbers uniformly distributed between 0 and 1; and r_n is a random number generally distributed with a mean of zero. The search agents of the PSA algorithm are X_v and X_i, and the best agents are X_v^* and X_i^*.

2.2.1. Search Agents Initialization

The PSA first initializes the vector of the search agent randomly between the lower and upper bounds (**LB** and **UB**) with the dimension (1 × D) of a cost function. Initialization plays a vital part in successfully helping the proposed algorithm find the global optimum solution. There are two types of initialization: symmetrical and nonsymmetrical methods. For symmetrical initialization, all elements of the search agent vector are multiplied by the same random number. For the nonsymmetrical initialization, every element is multiplied by a different random number. For the PSA, there are two search agent vectors, $\mathbf{X}_v$ and $\mathbf{X}_i$, which can be initialized by any initialization method or by both initialization methods, as in (15).

$$\mathbf{X}_v = \mathbf{LB} + r \times (\mathbf{UB} - \mathbf{LB}) \tag{15}$$

$$\mathbf{X}_i = \mathbf{LB} + \mathbf{R} \times (\mathbf{UB} - \mathbf{LB}) \tag{16}$$

where $\mathbf{X}_v = [x_{v1}, x_{v2}, x_{v3}, \ldots x_{vd}]^{\mathrm{T}}$, $\mathbf{X}_i = [x_{i1}, x_{i2}, x_{i3}, \ldots x_{id}]^{\mathrm{T}}$, $\mathbf{UB} = [ub_1, ub_2, ub_3, \ldots ub_d]^{\mathrm{T}}$, LB = $[lb_1, lb_2, lb_3, \ldots lb_d]^{\mathrm{T}}$, and $\mathbf{R} = [r_1, r_2, r_3, \ldots r_d]^{\mathrm{T}}$. The initialization can be conducted for a number n of search agents, so the dimension of all search agents is ($N \times D$).

2.2.2. Exploration and Exploitation of PSA

For better exploration, firstly, the search agents X_v and X_i are initialized differently, as in (15) and (16), which will widen the search areas of the PSA. Secondly, these agents are updated using (12) and (13), where they search in opposite directions for $V_R \neq I_R$, as shown in Figure 3. For exploitation of the PSA, the changes of both search agents converge to zero for $V_R = I_R$, as shown in Figure 3. Moreover, both search agents are designed to encircle the best solution due to the presence of $V_R = -X_v$ in (12) and $I_R = -X_i$ in (13). The balance between exploration and exploitation is managed by (14), where the first half of total iterations is used for exploitation, and the second half of iterations is utilized for exploration, where the search agents can be diverted using normally distributed random numbers.

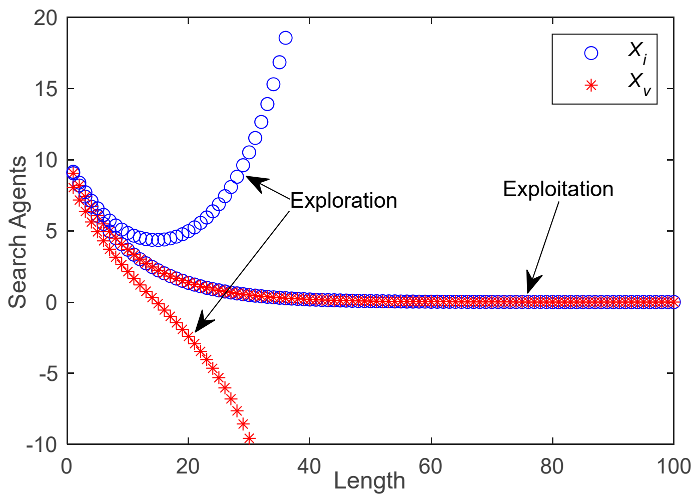

Figure 3. Exploration and exploitation of PSA.

For more explanation about exploitation and exploration, two common functions, the Rosenbrock and Rastrigin functions, are used to test the exploration and exploitation of the PSA. Figure 4 shows how the search agents found the best voltage immediately after 10 iterations for the Rastrigin function; however, they were stuck in a local solution for the Rosenbrock function. After $t \geq 0.5\ T$ (at t = 300), the exploration capability of the PSA was activated, and the search agents moved to the best voltage.

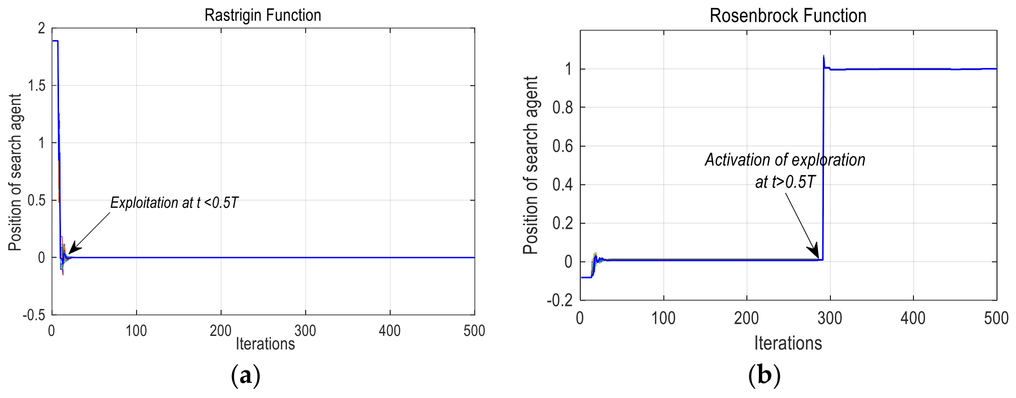

Figure 4. Examples of exploitation and exploration of PSA as in (**a**,**b**).

2.2.3. Encircling Behavior of PSA

Even though the search agents X_v and X_i of the PSA go in opposite directions, as shown in Figure 3, in the end, they encircle the best solution (X^*) because each agent relies on the update step of the other best agent, as shown in (12) and (13). We can notice that the term V_R exists in the search agent X_i and the term I_R exists in the search agent X_v. Therefore, both search agents encircle the best position, X^*. The flowchart of the proposed PSA is depicted in Figure 5.

2.2.4. Computational Complexity of PSA

The complexity of the algorithms varies with the number of loops, the number of search agents (N), and the number of fitness function evaluations. Big O Notation is utilized to measure the computational complexity of PSA. As shown in Figure 4, it starts with the loop of initialization for two kinds of search agents, X_v and X_i, and then the PSA has $O(2N)$ complexity. For the update process and function evaluations, the PSA has $O(T \times 2N)$. Therefore, the total complexity of the PSA is $O(2N + 2T \times N)$, which can be reduced to the term with the largest order, such as $O(2T \times N)$.

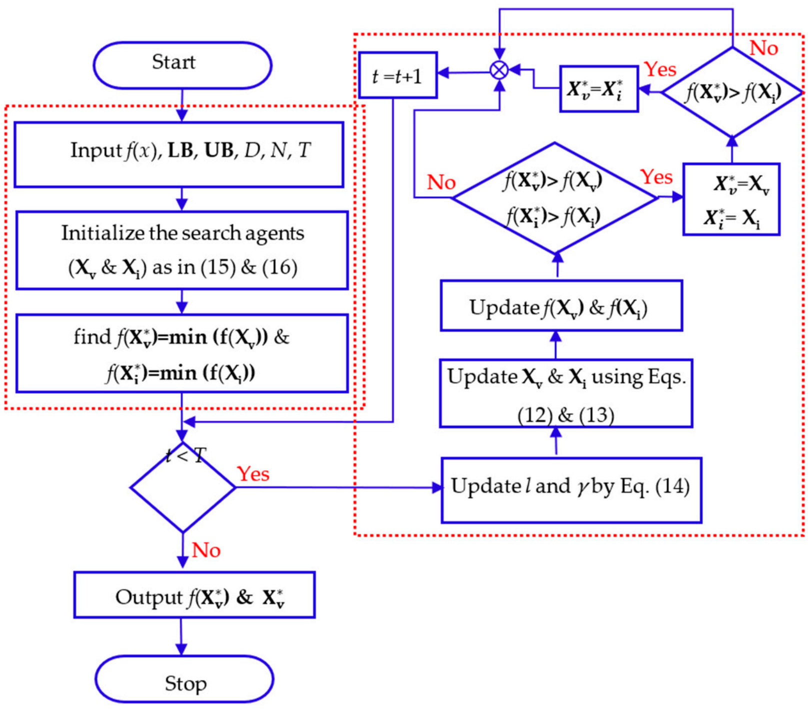
Start
Input $f(x)$, **LB**, **UB**, D, N, T
Initialize the search agents ($\mathbf{X}_v$ & $\mathbf{X}_i$) as in (15) & (16)
find $f(\mathbf{X}_v^*)$=min (f($\mathbf{X}_v$)) & $f(\mathbf{X}_i^*)$=min (f($\mathbf{X}_i$))
$t < T$
Yes
No
Output $f(\mathbf{X}_v^*)$ & $\mathbf{X}_v^*$
Stop
$t = t+1$
Update l and γ by Eq. (14)
Update $\mathbf{X}_v$ & $\mathbf{X}_i$ using Eqs. (12) & (13)
Update $f(\mathbf{X}_v)$ & $f(\mathbf{X}_i)$
$f(\mathbf{X}_v^*)> f(\mathbf{X}_v)$ $f(\mathbf{X}_i^*)> f(\mathbf{X}_i)$
No
Yes
$X_v^*=\mathbf{X}_v$ $X_i^*=\mathbf{X}_i$
$f(\mathbf{X}_v^*)> f(\mathbf{X}_i)$
Yes
No
$X_v^*=X_i^*$

Figure 5. The flowchart of the PSA.

3. Optimization Results of Testing Functions

First of all, the robustness of the proposed PSA should be tested using the 23 well-known testing functions. These functions are utilized to verify the exploration and exploitation behaviors of all previously published algorithms. Some of these functions are utilized to test the exploitation ability of the algorithms because they are convex and have one lowest solution, such as the unimodal functions (F1–F7) displayed in Table 1. Other functions are utilized to verify the exploration capability of algorithms to escape being trapped in a local minimum solution because they have different local minimum solutions rather than the global minimum solution, such as the multimodal functions (F8–F23) displayed in Tables 2 and 3. Then, the minimum values of these functions found by the PSA are compared with those of other renowned algorithms.

Table 1. Unimodal testing functions with unfixed dimensions (d).

No.	Name	Formula	Dimension	[LB, UB]	Minimum
F1	Sphere	$\sum_{i=1}^{d} y_i^2$	d = 30	$[-100, 100]^d$	0
F2	Shwefel 2.22	$\sum_{i=1}^{d} \lvert y_i\rvert + \prod_{i=1}^{d} \lvert y_i\rvert$	d = 30	$[-10, 10]^d$	0
F3	Shwefel 1.2	$\sum_{i=1}^{d} \left(\sum_{j=1}^{i} y_j\right)^2$	d = 30	$[-100, 100]^d$	0
F4	Schwefel 2.21	$\max_i\{\lvert y_i\rvert, 1 \le i \le d\}$	d = 30	$[-100, 100]^d$	0
F5	Rosenbrock	$\sum_{i=1}^{d} [100(y_{i+1} - y_i^2)^2 + (y_i - 1)^2]$	d = 30	$[-30, 30]^d$	0
F6	Step	$\sum_{i=1}^{d} (y_i + 0.5)^2$	d = 30	$[-100, 100]^d$	0
F7	Quartic Noise	$\sum_{i=1}^{d} i.y_i^4 + random[0, 1)$	d = 30	$[-1.28, 1.28]^d$	0

Table 2. Multimodal testing functions with unfixed dimensions (d).

No.	Name	Formula	Dimension	[LB, UB]	Minimum
F8	Schwefel 2.26	$\sum_{i=1}^{d} y_i \sin(\sqrt{\lvert y_i \rvert})$	d = 30	$[-500, 500]^{d}$	$-418.98 \times d$
F9	Rastrigin	$\sum_{i=1}^{n} [{y_i}^2 - 10\cos(2\pi y_i) + 10]$	d = 30	$[-5.12, 5.12]^{d}$	0
F10	Ackley	$-20\exp(-0.2\sqrt{\frac{1}{d}\sum_{j=1}^{d} y_j}) - \exp(\frac{1}{n}\cos(2\pi y_j)) + 20 + e$	d = 30	$[-32, 32]^{d}$	0
F11	Griewank	$\frac{1}{4000}\sum_{i=1}^{d} {y_i}^2 - \prod_{i=1}^{d}\cos(\frac{y_i}{\sqrt{i}}) + 1$	d = 30	$[-600, 600]^{d}$	0
F12	Penalized	$\frac{\pi}{d}\{10\sin(\pi y_1) + \sum_{i=1}^{d}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})]$ $+(y_d - 1)^2\} + \sum_{i=1}^{d} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i+1}{4}$ & $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	d = 30	$[-50, 50]^{d}$	0
F13	Generalized Penalized	$0.1 \times \{10\sin^2(3\pi y_1) + \sum_{i=1}^{d}(y_i - 1)^2[1 + \sin^2(3\pi y_i + 1)]$ $+(y_d - 1)^2[1 + \sin^2(2\pi y_d)]\} + \sum_{i=1}^{d} u(y_i, 5, 100, 4)$	d = 30	$[-50, 50]^{d}$	0

Table 3. Fixed-dimension (d) multimodal testing functions.

No.	Name	Formula	Dimension	[LB, UB]	Best Solution
F14	De Jong Fifth	$\left(\frac{1}{500}+\sum_{i=1}^{25}\frac{1}{i+\sum_{j=1}^{2}(y_i-a_{ij})^6}\right)^{-1}$	d = 2	$[-65, 65]^d$	1
F15		$\sum_{i=1}^{11}\left[a_i-\frac{y_i(b_i^2+b_iy_2)}{b_i^2+b_iy_3+y_4}\right]^2$	d = 4	$[-5, 5]^d$	0.00030
F16	Six-Hump Camel	$4{y_1}^2-2.1{y_1}^4+\frac{1}{3}{y_1}^6+y_1y_2-4{y_2}^2+4{y_2}^4$	d = 2	$[-5, 5]^d$	−1.0316
F17	Branins	$\left(y_2-\frac{5.1}{4\pi^2}{y_1}^2+\frac{5}{\pi}y_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos(y_1)+10$	d = 2	$[-5, 5]^d$	0.398
F18	Goldstein–Price	$[1+(y_1+y_2+1)^2(19-14y_1+3{y_1}^2-14y_2+6y_1y_2+3{y_2}^2)]\times$ $[30+(2y_1-3y_2)^2\times(18-32y_1+12{y_1}^2$ $+48y_2-36y_1y_2+27{y_2}^2)]$	d = 2	$[-2, 2]^d$	3
F19	Hartmann 3D	$-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{3}a_{ij}(y_j-p_{ij})^2\right)$	d = 3	$[1, 3]^d$	−3.86
F20	Hartmann 6D	$-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{6}a_{ij}(y_j-p_{ij})^2\right)$	d = 6	$[0, 1]^d$	−3.32
F21	Shekel 5	$-\sum_{i=1}^{5}\left[(Y-a_i)(Y-a_i)^T+c_i\right]^{-1}$	d = 4	$[0, 10]^d$	−10.1532
F22	Shekel 7	$-\sum_{i=1}^{7}\left[(Y-a_i)(Y-a_i)^T+c_i\right]^{-1}$	d = 4	$[0, 10]^d$	−10.4028
F23	Shekel 10	$-\sum_{i=1}^{10}\left[(Y-a_i)(Y-a_i)^T+c_i\right]^{-1}$	d = 4	$[0, 10]^d$	−10.5363

In this section, we used eight metaheuristic optimization algorithms besides the PSA. These are the particle swarm optimization (PSO), grey wolf optimizer (GWO), whale optimization algorithm (WOA), sine-cosine algorithm (SCA), transient search optimization (TSO), salp swarm algorithm (SSA), cuckoo search algorithm (CS), and artificial electric field algorithm (AEFA). These algorithms are carefully selected based on the most popular algorithms applied in different engineering applications, such as PSO, GWO, and the WOA. Some of these algorithms are selected because they are inspired by physical behaviors, such as TSO and the AEFA. The remaining algorithms are selected randomly for more comparisons. The values of the algorithms' parameters are displayed in Table 4.

Table 4. The setting of the compared algorithms.

Algorithm	**Parameters**
Proposed PSA	l decrease from 1 to 0
PSO	weight ω varies between [0.5, 0.3] $c_1 = c_2 = 2$
GWO	The variable a changes between [2, 0]
SSA	The probability is 0.5
SCA	a = 2 and $p = 0.5$
CS	$P_a = 0.25$, tolerance is 1×10^{-5}, beta = 1.5,
WOA	$l = 1$, a decrease from 2 to 0, a_2 decrease from -1 to -2
TSO	t decreases from 2 to 0, K = 1, probability update is 0.5
AFEA	Alfa = 30, $K_0 = 500$;

In this work, the testing experiments are implemented using MATLAB R2023a on Windows 10 64 bits on a Core i7, 16 GB RAM laptop. For an unbiased evaluation, all compared algorithms have a similar number of search agents, $N = 30$, and the same total number of iterations, $T = 500$, and all algorithms have the same initialized search agents.

3.1. Statistical Analysis

Due to the random operation of all metaheuristic algorithms, we need to test these algorithms in many independent optimization experiments. Therefore, all optimization algorithms are executed 20 times for each testing function in this paper. Then, the robustness of an algorithm is confirmed by applying different statistical methods, for example, mean (m) and standard deviation (σ) and nonparametric testing methods. Table 5 displays the m and σ of 20 independent runs for all 23 testing functions using the offered PSA and other compared algorithms. The outcomes proved that the offered PSA is competitive with different algorithms and obtained the optimum outcomes for 16 functions out of 23. However, the second competitive algorithm is the CS algorithm, which achieves 9 best functions out of 23 functions.

The exploitation of the algorithms is measured using unimodal functions (F1–F8), where the proposed PSA achieved the best outcomes for seven out of eight functions, which proved the exploitation capability of the PSA among other algorithms. On the other side, the exploration capability of the PSA is measured using the multimodal functions (F9–F23). The results show that the PSA obtained the 10 best results out of 15, whereas the CS algorithm achieved second place with the 9 best results out of 15. To check the equilibrium between exploration and exploitation, Rosenbrock (F5) and Rastrigin (F9) are used because they are contrasted, and no algorithm can solve both of them successfully. The results verified that the PSA succeeded in obtaining better results than other algorithms in both types of functions.

Table 5. Mean and standard deviations of 20 independent trials.

No.		GWO	SCA	SSA	CS	WOA	PSO	TSO	AEFA	PSA
F1	m	2.954E−38	4.401E−11	1.204E−07	6.780E−01	1.845E−75	3.772E−07	2.037E−77	1.818E+00	**3.507E−81**
	σ	4.217E−38	7.184E−11	7.488E−08	3.424E−01	8.119E−75	5.262E−07	9.038E−77	4.526E+00	1.568E−80
F2	m	1.798E−22	4.088E−08	1.919E−01	4.904E−01	**4.471E−52**	9.866E−04	7.933E−40	9.613E+00	2.814E−42
	σ	2.141E−22	9.436E−08	1.990E−01	1.785E−01	1.355E−51	3.135E−03	3.353E−39	5.891E+00	8.774E−42
F3	m	1.465E−08	1.942E−08	1.891E+02	3.220E+02	2.092E+00	9.432E+02	2.050E−72	8.880E+02	**5.964E−81**
	σ	5.082E−08	4.062E−08	3.092E+02	2.957E+02	5.104E+00	1.118E+03	9.161E−72	4.508E+02	2.536E−80
F4	m	1.022E−01	8.313E−01	7.620E−01	1.083E+00	1.833E−02	2.120E+00	9.245E−40	1.486E+00	**6.223E−43**
	σ	2.591E−01	1.103E+00	6.087E−01	6.512E−01	3.710E−02	2.301E+00	4.122E−39	1.199E+00	1.474E−42
F5	m	2.161E+01	5.604E+01	1.991E+01	5.693E+01	7.043E+00	2.968E+01	4.282E+01	6.921E+02	**2.507E−02**
	σ	1.056E+01	8.190E+01	1.286E+01	4.234E+01	1.228E+01	2.689E+01	1.308E+02	1.143E+03	4.152E−02
F6	m	5.181E−01	5.225E+00	1.006E−06	5.647E−01	6.694E−03	**6.793E−07**	5.687E−01	1.379E+00	4.303E−02
	σ	3.616E−01	1.234E+00	3.178E−06	3.298E−01	1.232E−02	1.157E−06	6.711E−01	3.051E+00	1.902E−01
F7	m	1.701E−03	5.580E−04	2.901E−02	1.043E−02	7.335E−04	3.340E−02	6.114E−03	2.270E−01	**6.207E−05**
	σ	1.196E−03	7.296E−04	2.039E−02	1.271E−02	6.587E−04	2.274E−02	1.298E−02	1.791E−01	6.558E−05
F8	m	−1.176E+04	−1.173E+04	−1.239E+04	−1.243E+04	−1.239E+04	−1.234E+04	−1.247E+04	−1.226E+04	**−1.257E+04**
	σ	1.174E+03	1.161E+03	7.945E+02	2.125E+02	7.951E+02	8.072E+02	2.035E+02	4.015E+02	2.810E−02
F9	m	5.013E+01	2.966E+01	5.970E+00	1.652E+01	**0.000E+00**	1.323E+01	4.601E−05	3.100E+01	**0.000E+00**
	σ	4.214E+01	4.088E+01	1.225E+01	1.156E+01	0.000E+00	1.463E+01	1.111E−04	1.977E+01	0.000E+00
F10	m	1.910E−14	2.723E−05	8.452E−01	1.482E+00	3.997E−15	1.493E+00	9.446E−04	1.379E+00	**4.441E−16**
	σ	3.972E−15	7.773E−05	1.101E+00	8.255E−01	2.577E−15	1.916E+00	1.101E−03	1.058E+00	0.000E+00
F11	m	1.215E−03	6.158E−09	1.418E−02	6.696E−01	**0.000E+00**	1.746E−02	8.731E−02	6.641E−01	**0.000E+00**
	σ	3.741E−03	1.751E−08	1.305E−02	1.297E−01	0.000E+00	1.371E−03	1.968E−02	4.874E−01	0.000E+00
F12	m	4.925E−01	1.218E+00	4.070E−02	8.174E−02	1.105E−04	1.371E+00	1.219E−02	2.001E+00	**1.433E−05**
	σ	1.417E+00	1.462E+00	1.183E−01	7.730E−02	2.800E−04	2.738E+00	2.050E−02	1.659E+00	2.472E−05

Table 5. *Cont.*

F13	m	3.058E−01	2.085E+00	1.750E−01	3.605E−01	1.707E−02	**4.397E−03**	4.041E−01	1.017E+01	1.076E−02
	σ	2.599E−01	8.328E−01	6.583E−01	4.287E−01	5.871E−02	5.525E−03	9.038E−01	1.078E+01	4.679E−02
F14	m	1.978E+00	1.652E+00	1.048E+00	**9.980E−01**	1.486E+00	**9.980E−01**	**9.980E−01**	3.074E+00	1.147E+00
	σ	2.643E+00	9.196E−01	2.223E−01	7.675E−16	2.184E+00	0.000E+00	4.434E−08	2.259E+00	4.857E−01
F15	m	1.354E−03	5.678E−04	9.093E−04	**3.141E−04**	**3.131E−04**	4.319E−04	3.889E−04	1.045E−02	4.438E−04
	σ	4.475E−03	3.540E−04	6.895E−04	6.6833E−06	6.186E−06	1.139E−04	1.244E−04	6.900E−03	2.274E−04
F16	m	**−1.032E+00**	**−1.032E+00**	**−1.032E+00**	**−1.032E+00**	**−1.032E+00**	**−1.032E+00**	**−1.032E+00**	**−1.032E+00**	**−1.032E+00**
	σ	8.071E−08	9.936E−05	1.929E−14	8.823E−17	8.258E−10	2.161E−16	7.439E−06	2.038E−16	2.783E−06
F17	m	**3.979E−01**	4.020E−01	**3.979E−01**	**3.979E−01**	**3.979E−01**	**3.979E−01**	**3.979E−01**	**3.979E−01**	**3.979E−01**
	σ	2.293E−06	4.196E−03	9.744E−15	0.000E+00	1.453E−05	0.000E+00	4.011E−05	0.000E+00	3.393E−05
F18	m	**3.000E+00**	**3.000E+00**	**3.000E+00**	**3.000E+00**	**3.000E+00**	**3.000E+00**	3.107E+01	3.056E+00	**3.000E+00**
	σ	1.935E−05	6.407E−05	1.253E−13	1.585E−15	8.710E−05	9.557E−16	1.209E+00	2.518E−01	2.115E−04
F19	m	−3.860E+00	−3.850E+00	**−3.863E+00**	**−3.863E+00**	−3.857E+00	−3.862E+00	−3.731E+00	−3.862E+00	−3.862E+00
	σ	3.312E−03	4.917E−03	9.363E−11	2.040E−15	1.049E−02	1.762E−03	3.073E−01	1.102E−03	1.400E−03
F20	m	**−3.298E+00**	−2.696E+00	−3.255E+00	−3.322E+00	−3.294E+00	−3.286E+00	−3.275E+00	−3.322E+00	−3.223E+00
	σ	7.242E−02	3.335E−01	1.025E−01	4.843E−09	7.383E−02	6.652E−02	1.799E−01	4.556E−16	8.357E−02
F21	m	−8.318E+00	−7.132E+00	**−1.015E+01**	**−1.015E+01**	−9.898E+00	−8.373E+00	−1.013E+01	−7.285E+00	**−1.015E+01**
	σ	2.229E+00	2.255E+00	3.153E−11	1.500E−08	1.140E+00	2.488E+00	3.212E−02	2.158E+00	6.068E−04
F22	m	−8.548E+00	−7.536E+00	−9.612E+00	**−1.040E+01**	−1.014E+01	−8.289E+00	−1.038E+01	−9.712E+00	**−1.040E+01**
	σ	2.445E+00	2.342E+00	1.932E+00	1.765E−08	1.188E+00	2.656E+00	5.051E−02	1.703E+00	8.270E−04
F23	m	−8.912E+00	−7.562E+00	−1.027E+01	**−1.054E+01**	−9.995E+00	−8.923E+00	−1.051E+01	**−1.054E+01**	**−1.054E+01**
	σ	2.200E+00	2.216E+00	1.199E+00	1.223E−07	1.664E+00	2.528E+00	2.716E−02	1.578E−15	2.631E−03

Additionally, the Wilcoxon signed-rank test with a 5% significance level is utilized to verify the significance of the PSA among other compared algorithms. This test is a null hypothesis test, which hypothesizes that the results of the PSA and other algorithms are the same if the *p*-value is higher than 0.05. Table 6 shows the *p*-values and the denial of the null hypothesis (*h* = true) or acceptance of it (*h* = false). We can notice that the most dominant decision is *h* = true, meaning a significant difference exists between the PSA and other algorithms.

Finally, a boxplot is used for further statistical analysis to verify the performance of the PSA during 20 independent experiments. Figure 6a–f shows the boxplot for six benchmark functions (F2, F4, F8, F10, F12, and F21). The boxplot analysis shows that the PSA has obtained a very small deviation between the maximum and minimum values of 20 experiments. Moreover, it is noticeable that there are outlier points, which are labeled by (+), for all compared algorithm except PSA. Therefore, the boxplot proved the robustness of the PSA over other algorithms.

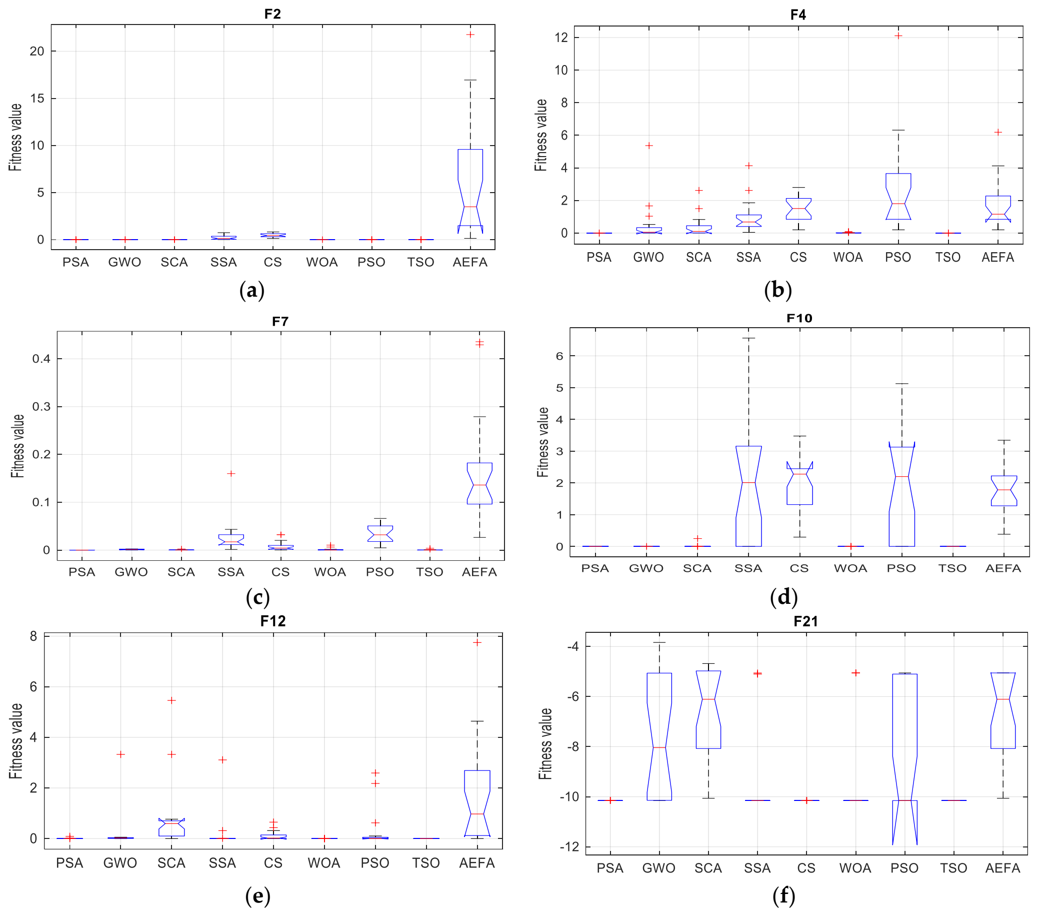

Figure 6. Statistical analysis using boxplot for all compared algorithms.

Table 6. Wilcoxon signed-rank test.

No.		GWO	SCA	SSA	CS	WOA	PSO	TSO	AEFA
F1	*p*-value	6.683E−05	6.683E−05	6.683E−05	6.683E−05	6.683E−05	6.683E−05	6.683E−05	6.683E−05
	h	true	true	true	true	true	true	true	true
F2	*p*-value	6.683E−05	6.683E−05	6.683E−05	6.683E−05	1.03E−04	6.683E−05	6.683E−05	6.683E−05
	h	true	true	true	true	true	true	true	true
F3	*p*-value	6.683E−05	6.683E−05	6.683E−05	6.683E−05	6.683E−05	6.683E−05	6.683E−05	6.683E−05
	h	true	true	true	true	true	true	true	true
F4	*p*-value	6.683E−05	6.683E−05	6.683E−05	6.683E−05	6.683E−05	6.683E−05	6.683E−05	6.683E−05
	h	true	true	true	true	true	true	true	true
F5	*p*-value	6.683E−05	6.683E−05	6.683E−05	6.683E−05	1.32E−03	1.03E−04	6.683E−05	6.683E−05
	h	true	true	true	true	true	true	true	true
F6	*p*-value	6.81E−04	6.683E−05	6.683E−05	1.20E−04	3.04E−02	1.03E−04	7.80E−04	2.82E−03
	h	true	true	true	true	true	true	true	true
F7	*p*-value	6.683E−05	2.54E−04	6.683E−05	6.683E−05	2.19E−04	6.683E−05	1.20E−04	6.683E−05
	h	true	true	true	true	true	true	true	true
F8	*p*-value	6.683E−05	6.683E−05	1.51E−03	6.683E−05	9.11E−01	7.31E−02	1.03E−04	6.683E−05
	h	true	true	true	true	true	true	true	true
F9	*p*-value	1.32E−04	4.38E−04	6.683E−05	6.683E−05	trueE+00	6.683E−05	6.683E−05	6.683E−05
	h	true	true	true	true	false	true	true	true
F10	*p*-value	6.81E−05	6.683E−05	6.683E−05	6.683E−05	6.10E−05	6.683E−05	6.683E−05	6.683E−05
	h	true	true	true	true	true	true	true	true
F11	*p*-value	5.00E−01	6.683E−05	6.683E−05	6.683E−05	trueE+00	6.683E−05	6.683E−05	6.683E−05
	h	false	true	true	true	false	true	true	true
F12	*p*-value	6.683E−05	6.683E−05	2.19E−04	6.683E−05	5.75E−01	6.01E−01	6.683E−05	6.683E−05
	h	true	true	true	true	false	false	true	true

Table 6. *Cont.*

No.		GWO	SCA	SSA	CS	WOA	PSO	TSO	AEFA
F13	*p*-value	2.19E−04	6.683E−05	1.32E−03	6.683E−05	1.52E−02	4.78E−01	3.90E−04	6.683E−05
	h	true	true	true	true	true	false	true	true
F14	*p*-value	4.38E−02	7.80E−04	1.16E−03	6.683E−05	3.33E−02	6.683E−05	5.02E−01	3.38E−04
	h	true	true	true	true	true	true	false	true
F15	*p*-value	1.79E−01	9.30E−02	1.11E−02	2.19E−04	7.80E−04	7.94E−01	5.02E−01	6.683E−05
	h	false	false	true	true	true	false	false	true
F16	*p*-value	2.54E−04	6.683E−05	6.683E−05	6.683E−05	1.03E−04	6.683E−05	1.87E−02	6.683E−05
	h	true	true	true	true	true	true	true	true
F17	*p*-value	3.04E−02	6.683E−05	6.683E−05	6.683E−05	1.56E−01	6.683E−05	2.28E−02	6.683E−05
	h	true	true	true	true	false	true	true	true
F18	*p*-value	1.52E−02	1.79E−01	6.683E−05	6.683E−05	3.33E−02	6.683E−05	6.683E−05	1.51E−03
	h	true	true	true	true	false	true	true	true
F19	*p*-value	3.32E−01	6.683E−05	6.683E−05	6.683E−05	2.28E−02	1.32E−03	2.19E−04	2.04E−01
	h	false	true	true	true	true	true	true	false
F20	*p*-value	6.42E−03	6.683E−05	2.63E−01	6.683E−05	2.20E−03	1.24E−02	4.55E−03	6.683E−05
	h	true	true	false	true	true	true	true	true
F21	*p*-value	6.683E−05	6.683E−05	6.683E−05	6.683E−05	8.52E−01	6.01E−01	6.683E−05	1.03E−04
	h	true	true	true	true	false	false	true	true
F22	*p*-value	6.683E−05	6.683E−05	7.31E−02	6.683E−05	2.63E−01	3.13E−01	6.683E−05	trueE+00
	h	true	true	false	true	false	false	true	false
F23	*p*-value	6.683E−05	6.683E−05	1.51E−03	6.683E−05	4.33E−01	trueE+00	1.63E−04	6.683E−05
	h	true	true	true	true	false	false	true	true

3.2. Optimization Convergence

The speed of finding the best solution plays a critical role in control systems, so we need to check the behavior of algorithms with each increment of iterations. Table 7 displays the total elapsed time for all compared algorithms during each independent experiment. However, Figure 7 shows the convergence behavior for all compared algorithms toward the best minimum solution. It is noticeable that all algorithms start from the same initial point for unbiased comparison. The proposed PSA shows fast convergence for the testing functions F1–F8, and then at the middle of total iterations (t = 250) it becomes slower because of changes in the values of the propagation constant (γ). For the remaining functions, the PSA competes with the other algorithms in searching for global solutions; particularly, at t = 250 the PSA escapes being trapped in a local solution.

Table 7. Elapsed time in seconds for algorithm execution.

	GWO	**SCA**	**SSA**	**CS**	**WOA**	**PSO**	**TSO**	**AEFA**	**PSA**
F1	0.19396	0.12807	0.12076	0.32297	0.07844	0.10489	0.07901	0.70572	0.14529
F3	0.28908	0.25333	0.21823	0.59093	0.21227	0.24697	0.20247	0.5662	0.21162
F5	0.22024	0.14419	0.11669	0.37283	0.09557	0.12428	0.09445	0.74423	0.18258
F7	0.25270	0.27462	0.17785	0.52765	0.18240	0.20720	0.15772	0.58886	0.23347
F9	0.21667	0.14238	0.10190	0.35608	0.09275	0.12364	0.07937	0.70294	0.15709
F11	0.21608	0.15083	0.11007	0.39285	0.09911	0.13238	0.09883	0.75162	0.18998
F13	0.43091	0.37946	0.34172	0.89211	0.32739	0.34480	0.31621	0.71861	0.45095
F15	0.08482	0.06125	0.04754	0.24100	0.04562	0.04603	0.04342	0.19139	0.07792
F17	0.04381	0.05114	0.03805	0.20785	0.03501	0.02629	0.03271	0.14607	0.05805
F19	0.05045	0.05739	0.05281	0.22891	0.05190	0.04391	0.04880	0.17714	0.12969
F21	0.06674	0.06795	0.06347	0.25504	0.05845	0.05484	0.06027	0.21141	0.10314
F23	0.08750	0.08496	0.08194	0.29582	0.07910	0.07160	0.07826	0.22082	0.14134
Sum	2.15295	1.79558	1.47103	4.68403	1.35800	1.52684	1.29151	5.72502	2.08111

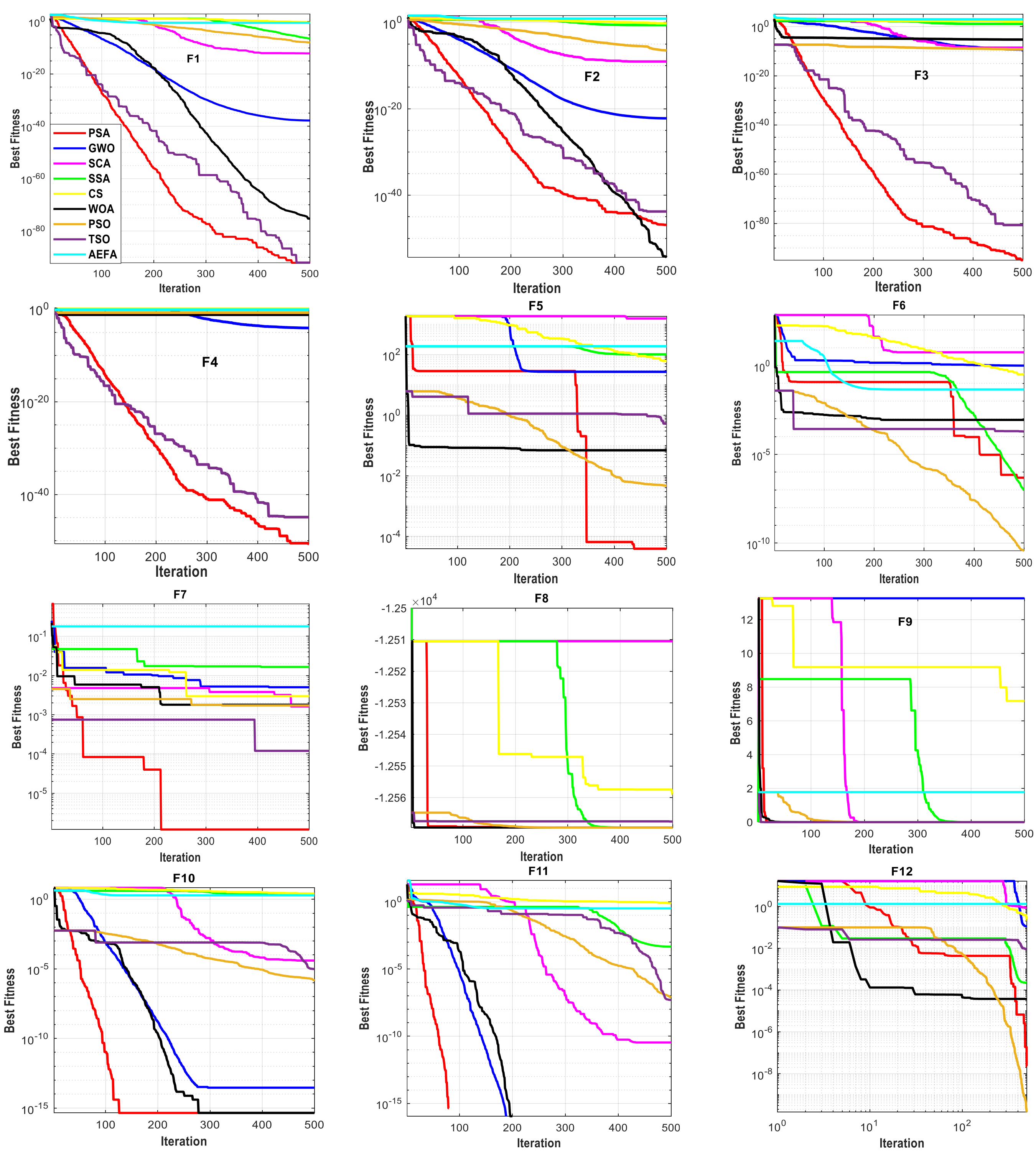

Figure 7. *Cont.*

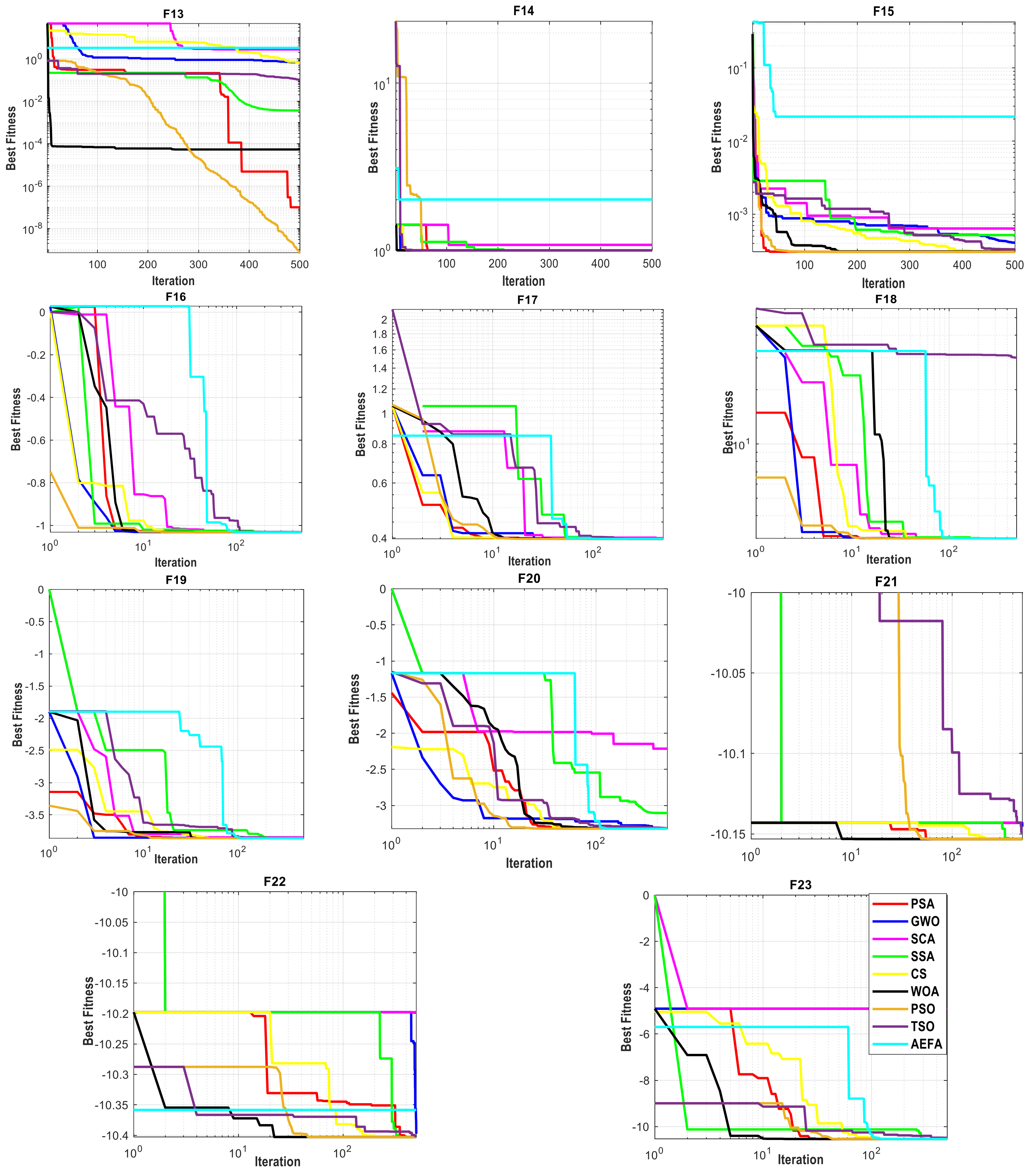

Figure 7. The optimization convergence of testing functions using different algorithms.

4. Renowned Engineering Problems

In this section, the offered PSA and other algorithms are utilized to obtain the optimum design of four famous engineering problems, including a three-bar truss structure, a compression spring, a pressure vessel, and a welded beam. The optimization experiments are conducted on MATLAB 2023a on a Windows 10 Core-i7 laptop. The number of search agents is 200, and the total number of iterations is 1000. The mean and standard deviation are calculated for 20 independent code runs.

4.1. Three-Bar Truss Design

Figure 8 depicts the structure of this truss, which is constructed of three bars with different cross-sectional areas (A_1, A_2, and A_3), where $A_1 = A_3$. These bars meet at a common node, where a P load is connected to this node. The dimensions between the supporting points of the three bars are equal, which is l, and the vertical dimension between the supporting points and the common node is l too. Reducing the weight of the truss structure is the chief objective, and the design parameters are A_1 and A_2. The objective function is shown in (17), and the subjected constraints are shown in (18). The PSA is applied to obtain the optimal cross-sectional areas that result in the minimum weight of the three-bar truss. Table 8 displays the outcomes of the optimal areas and the minimum weights obtained using eight algorithms. The statistical results (m and σ) show that the offered PSA has a performance that is comparable with other algorithms, and even better. Moreover, Figure 9 indicates that the PSA converges to the minimum result after only 85 iterations, which is quicker than most of the compared algorithms.

$$\min f(x = [A_1, A_2]) = \left(2\sqrt{2}A_1 + A_2\right) \times l \tag{17}$$

$$\begin{gathered} g_1(x) = \frac{2\sqrt{2}A_1 + A_2}{\sqrt{2}{A_1}^2 + 2A_1A_2}P - \sigma \leq 0 \;\&\; g_2(x) = \frac{A_2}{\sqrt{2}{A_1}^2 + 2A_1A_2}P - \sigma \leq 0 \\ g_3(x) = \frac{1}{A_1 + \sqrt{2}A_2}P - \sigma \leq 0\,,\; P = 2\ \text{kN},\, \sigma = 2\ \text{kN/cm}^2,\; l = 100\ \text{cm} \end{gathered} \tag{18}$$

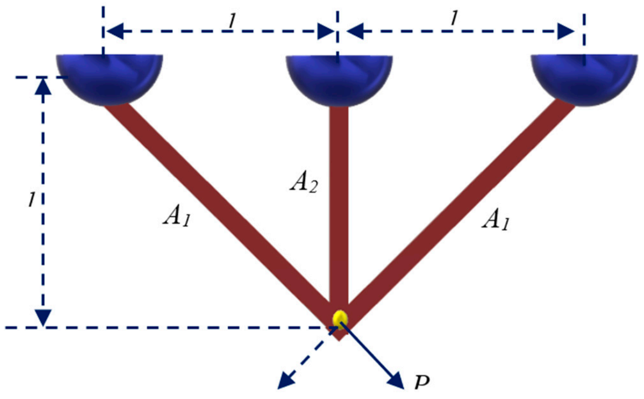

Figure 8. Three-bar truss structure.

Table 8. Optimal weight and parameters of three-bar truss design.

	GWO	SCA	SSA	CS	WOA	PSO	TSO	AEFA	PSA
Best	263.8959846	263.9090923	263.9195433	263.8958434	263.8959030	263.8958434	263.8969047	263.9195433	263.8958824
m	263.8964313	263.9553596	264.0115258	263.8958434	263.9431231	263.8958439	265.5472387	264.0115258	263.8984017
σ	4.11448E−04	4.21400E−02	9.00459E−02	0.00000E+00	6.82155E−02	7.34918E−07	1.49798E+00	9.00459E−02	2.21516E−03
x1	0.789005319	0.789674131	0.788484498	0.788675136	0.788960013	0.788681652	0.787476956	0.788484498	0.789351215
x2	0.407315799	0.405555191	0.408787881	0.408248286	0.407443127	0.408229858	0.411647866	0.408787881	0.406573046

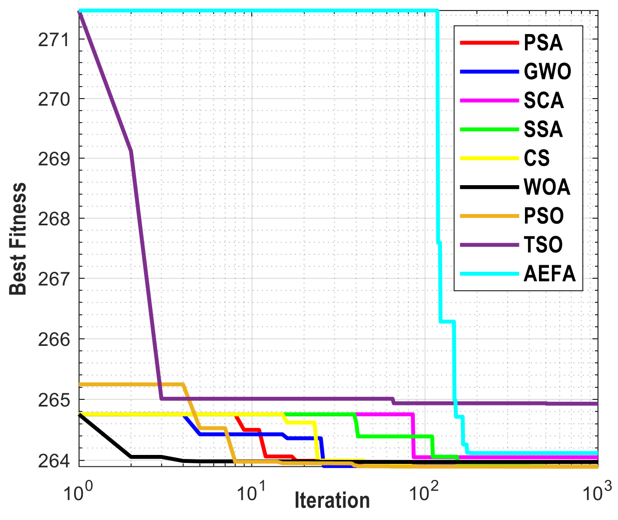

Figure 9. Convergence speed of algorithms to the minimum weight of the three-bar truss design.

4.2. Compression Spring Design

The structure of the compression spring, which is made of steel wire in a spiral form with several coils (N), a material diameter (d), and outer diameter of coil (D), is depicted in Figure 10. Reducing the weight of the compressional spring is the chief objective, where the controlling design parameters are D, d, and N. However, there are constraints to minimizing the weight of the spring, such as the torsional stress, deflection, traveling waves, and dimension. The objective function and the constraint functions of the compression spring design are expressed in (19) and (20). The PSA and other algorithms are applied to find the minimum weight of the compression spring, and the optimal outcomes are presented in Table 9. Furthermore, Figure 11 compares the PSA's convergence speed with other algorithms, showing that the PSA offers a quicker convergence than most of the compared algorithms, and the PSA converges to the minimum solution after 10 iterations only.

$$\min f(x = [D, d, N]) = D^2 d(N + 2) \tag{19}$$

$$\begin{aligned}
&g_1(x) = 1 - \frac{d^3 N}{71785 D^4} \leq 0 \text{ and } g_2(x) = \frac{4d^2 - Dd}{12566 D^3 (d - D)} + \frac{1}{5108 D^2} - 1 \leq 0 \\
&g_3(x) = 1 - \frac{140.45 D}{N d^2} \leq 0 \text{ and } g_4(x) = \frac{D + d}{1.5} - 1 \leq 0 \\
&0.05 \leq D \leq 2 \;\&\; 0.25 \leq d \leq 1.3 \;\&\; 2.00 \leq N \leq 15
\end{aligned} \tag{20}$$

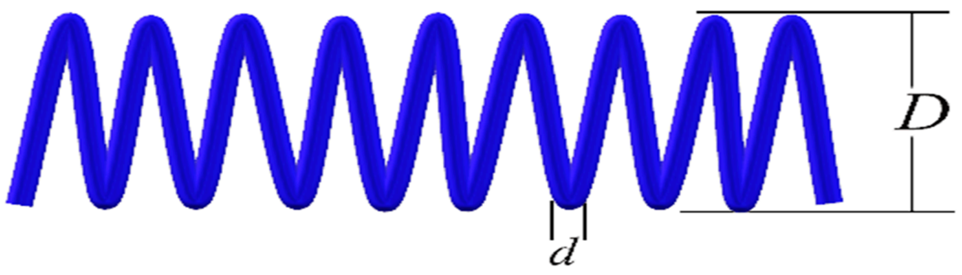

Figure 10. Compression spring structure.

Table 9. Optimal weight and parameters of compression spring design.

	GWO	SCA	SSA	CS	WOA	PSO	TSO	AEFA	PSA
Best	0.0177757	0.0127331	0.0178723	0.0126721	0.0127984	0.0178487	0.0126668	0.0196575	0.0127226
m	0.0177873	0.0128085	0.0219118	0.0126977	0.0136459	0.0179014	0.0139309	0.0336801	0.0132851
σ	0.0000080	0.0000546	0.0082340	0.0000195	0.0011371	0.0000373	0.0015793	0.0137893	0.0006535
D	0.0689962	0.0500000	0.0691542	0.0515176	0.0544458	0.0690812	0.0519697	0.0572892	0.0500000
d	0.9335070	0.3172711	0.9404314	0.3525262	0.4267345	0.9370252	0.3635050	0.5068545	0.3174241
N	2.0000000	14.0532354	1.9739055	11.5439263	8.1174138	1.9914889	10.9019563	9.8167597	14.0323211

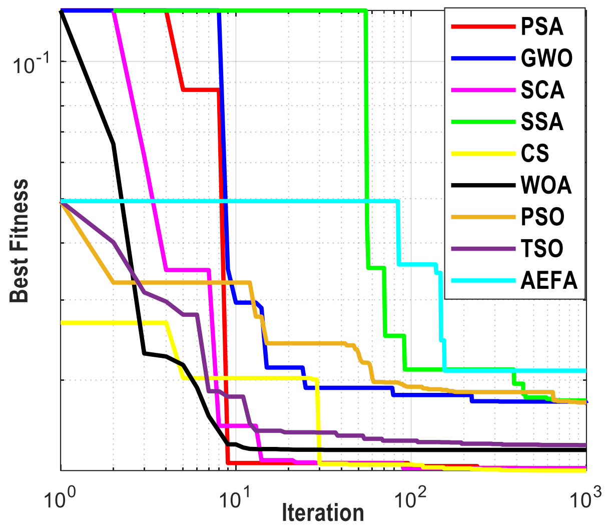

Figure 11. Convergence speed of algorithms to the minimum weight of the compression spring design.

4.3. Welded Beam Design

The structure of the welded beam, which is made by welding a beam to a rigid support member, is depicted in Figure 12. Therefore, reducing the fabrication cost of the welded beam, including labor and material costs, is the main objective, where the controlling design parameters are the dimensions of welding (h) and (l) and the dimensions of the member bar (t) and (b), as shown in Figure 10. However, there are constraints to lessening the cost of the welded beam, such as the weld stress (τ), bending stress (σ), and bar buckling load (P_c). The objective function and the constraint functions of welded beam design are expressed in (21) and (22). The PSA and other algorithms are utilized to find the minimum cost of the welded beam design, and the optimal outcomes are shown in Table 10. Furthermore, Figure 13 compares the PSA algorithm's convergence speed with other algorithms, showing that the PSA offers a faster convergence speed than most different algorithms, and the PSA converges to the best minimum after only 40 iterations.

$$\min f(x = [h\ l\ t\ b]) = 1.10471 \times h^2 \times l + 0.04811t \times b(14.0 + l) \tag{21}$$

$$\begin{aligned} &g_1(x) = \tau - \tau_{\max} \le 0 \text{ and } g_2(x) = \sigma - \sigma_{\max} \le 0 \\ &g_3(x) = \delta - \delta_{\max} \le 0 \text{ and } g_4(x) = h - b \le 0 \\ &g_5(x) = P - P_c \le 0 \text{ and } g_6(x) = 0.125 - h \le 0 \\ &g_7(x) = 0.10471 \times h^2 + 0.04811 \times l \times b \times (14.0 + l) - 5.0 \le 0 \\ &0.1 \le h \le 2\ \&\ 0.1 \le l \le 10\ \&\ 0.1 \le t \le 10\ \&\ 0.1 \le b \le 2 \end{aligned} \tag{22}$$

where

$$P = 6,000\ lb; L = 14\ in; E = 30 \times 10^6\ psi; G = 12 \times 10^6\ psi$$
$$\tau_{\max} = 13,600\ psi; \sigma_{\max} = 30,000\ psi; \delta_{\max} = 0.25\ in$$
$$M = P(L + \tfrac{l}{2}); R = \sqrt{\tfrac{l^2}{4} + (\tfrac{h+t}{2})^2}; \tau' = \tfrac{P}{\sqrt{2h \times l}}; \tau'' = \tfrac{MR}{J}$$
$$J = 2\sqrt{2}h \times l(\tfrac{l^2}{12} + (\tfrac{h+t}{2})^2); \tau = \sqrt{\tau'^2 + \tau'\tau''\tfrac{l}{R} + \tau''^2}$$
$$P_c = 4.013\tfrac{E}{L^2}\sqrt{\tfrac{t^2 \times b^6}{36}}(1 - \tfrac{t}{2L}\sqrt{\tfrac{E}{4G}})$$
$$\sigma = \tfrac{6PL}{b \times t^2}; \delta = \tfrac{4PL^3}{E \times b \times t^2}$$

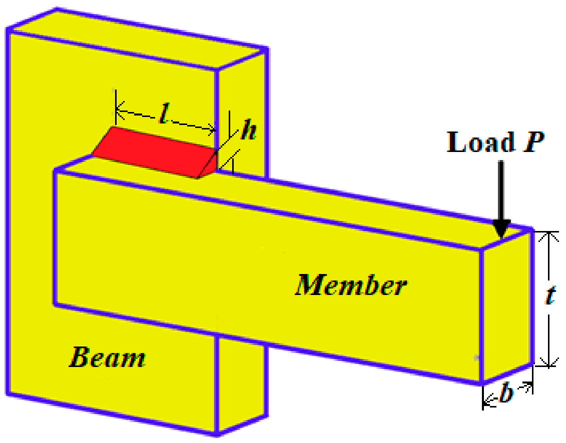

Figure 12. Welded beam structure.

Table 10. Optimal cost and parameters of welded beam design.

	GWO	SCA	SSA	CS	WOA	PSO	TSO	AEFA	PSA
Best	1.7261166637	1.7789887672	2.4719864210	1.7318832346	1.7332745737	1.7248523086	1.7338377644	2.4719864210	1.7250306573
m	1.7267581063	1.8287256087	3.0407655957	1.7378652824	1.7947045480	1.7248523088	1.7835502505	3.0407655957	1.7327229652
σ	5.19265E−04	2.34891E−02	3.96367E−01	4.99296E−03	3.61117E−02	5.07366E−10	4.43964E−02	3.96367E−01	5.14617E−03
h	0.2057226591	0.2074198677	0.1534791096	0.2060105624	0.2002029004	0.2057296398	0.1824037798	0.1534791096	0.2056551204
l	3.4718445411	3.6040616975	6.8582665831	3.4837895853	3.5923940636	3.4704886656	4.5358535044	6.8582665831	3.4718699729
t	9.0382895241	9.0678166725	9.5571430874	9.0294913594	9.0419370726	9.0366239104	8.8276280063	9.5571430874	9.0373416776
b	0.2058352914	0.2093402452	0.2391445616	0.2065207396	0.2057031460	0.2057296398	0.2155863505	0.2391445616	0.2057274905

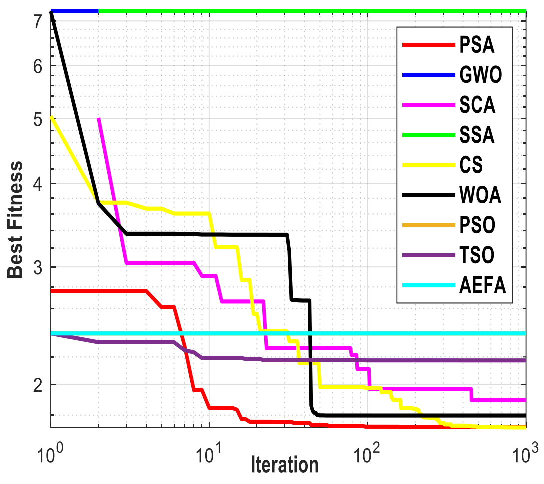

Figure 13. Convergence speed of algorithms to the minimum cost of the welded beam design.

4.4. Pressure Vessel Design

The structure of the pressure vessel, which is made of stainless steel plates that are welded to form a cylindrical shape, is depicted in Figure 14. Therefore, reducing the weight of the pressure vessel is the chief objective, where the controlling design parameters are the dimensions of the vessel, comprising the thickness of the shell (t_s), thickness of the head (t_h), inner radius (r), and length of the vessel (l). However, there are constraints to minimizing the cost of the pressure vessel design, such as the inner volume and the overall size of the vessel. The cost function and the constraint functions of pressure vessel design are expressed in (23) and (24). The PSA and other algorithms are utilized to find the lowest weight of the pressure vessel, and the optimal outcomes are shown in Table 11. Additionally, Figure 15 compares the convergence curve of the PSA with other algorithms, where the PSA offers a quicker convergence than the most of compared algorithms, and the PSA converges to the best minimum after only 50 iterations.

$$\min f(x = [T_s, T_h, R, L]) = 0.6224 T_s T_h L + 1.7781 T_h R^2 + 3.1661 {T_s}^2 L + 19.84 {T_s}^2 R \tag{23}$$

$$\begin{array}{l} g_1(x) = -T_s + 0.0193R \leq 0 \ \& \ g_2(x) = -T_h + 0.00954R \leq 0 \\ g_3(x) = -\pi R^2 L - \frac{4}{3}\pi R^3 + 1{,}296{,}000 \leq 0 \ \& \ g_4(x) = L - 240 \leq 0 \\ 0 \leq T_s \leq 99 \ \& \ 0 \leq T_h \leq 99 \ \& \ 10 \leq R \leq 200 \ \& \ 10 \leq L \leq 200 \end{array} \tag{24}$$

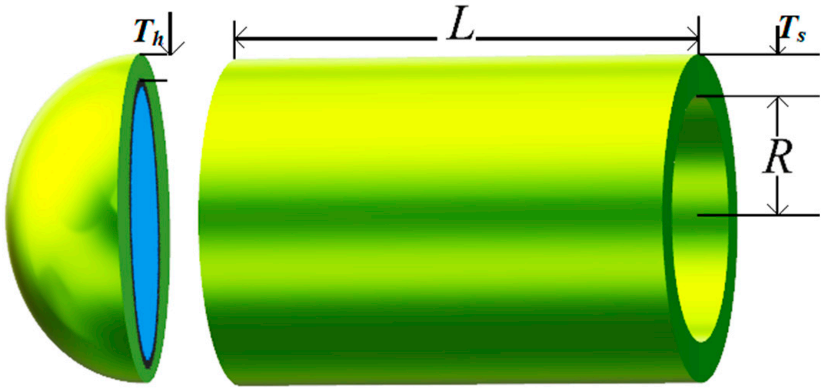

Figure 14. Cylindrical pressure vessel structure.

Table 11. Optimal weight and parameters of pressure vessel design.

	GWO	SCA	SSA	CS	WOA	PSO	TSO	AEFA	PSA
Best	5889.3161636	6084.9664550	6288.2580937	5886.5458106	6033.9780276	5942.1267062	6713.1969272	26533.6320741	5886.9897087
m	5894.1446757	6307.2406394	7009.5719313	5893.2120452	6631.8600064	6101.1999200	7102.4268313	96354.3822355	6077.8295567
σ	6.1000446	139.9687665	874.9974107	3.8547414	493.3610049	140.4149521	893.3723107	44235.2712807	308.1664002
t_s	0.7784311	0.7895409	0.9652823	0.7782347	0.7910326	0.8100689	1.0831728	2.8963402	0.7783762
t_h	0.3848615	0.3983170	0.4771401	0.3847895	0.4320066	0.4004175	0.5158489	1.1004971	0.3846826
r	40.3285572	40.7806669	50.0145885	40.3216202	40.9557154	41.9724803	54.0592595	64.9921818	40.3199574
l	200.0000000	200.0000000	98.2297744	200.0000000	191.3306399	178.2037836	69.0821101	51.8458794	200.0000000

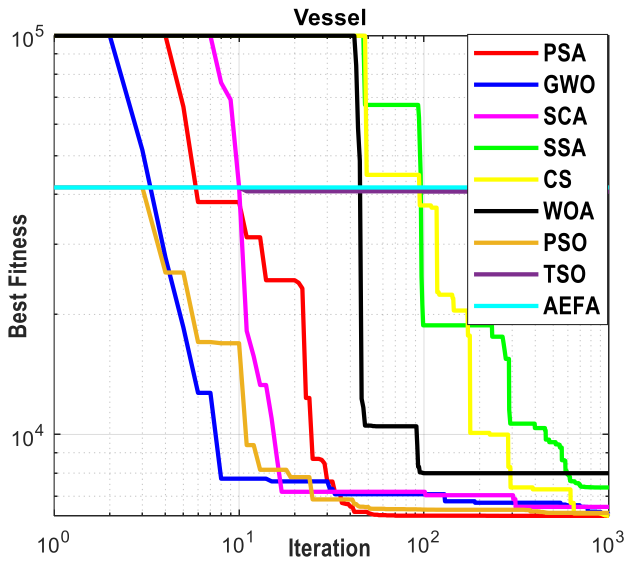

Figure 15. Convergence speed of algorithms to the minimum weight of the pressure vessel design.

5. Conclusions

In this work, a new physics-based metaheuristic optimization algorithm named the propagation search algorithm (PSA) is offered and utilized to optimize different engineering design problems. First, the exploration and exploitation capabilities of the PSA are confirmed using 23 famous testing functions. The obtained outcomes for the Rosenbrock and Rastrigin functions have confirmed that the PSA has a balanced capability between exploration and exploitation in the optimization procedure. The propagation constant of the PSA controls this balancing capability. The statistical tests are applied to verify the superiority of the PSA among other compared algorithms. The PSA has succeeded in finding the optimal solution for 17 functions out of 23; however, the second-ranked algorithm solved only 9 functions. The Wilcoxon signed-ranked test is applied to confirm the significance level of the outcomes of the PSA compared with those of other algorithms, where 90% of the computed p-values are less than 5%. The PSA and the compared algorithms are utilized to obtain the optimum design of four famous engineering problems, including the three-bar truss, compression spring, welded beam, and pressure vessel. The PSA has succeeded in finding competitive results and fast convergence to the minimum fitness values. The optimization results have shown that the proposed PSA is a promising optimizer that can be easily applied in engineering fields. Finally, the simple form of the PSA limits its performance in some engineering applications. Therefore, the future work will present different enhancements to the PSA and will apply it in power systems.

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

Funding: This research was funded by the Deputyship for Research and Innovation, "Ministry of Education" in Saudi Arabia (IFKSUOR3-328-1).

Data Availability Statement: Not applicable.

Acknowledgments: The authors extend their appreciation to the Deputyship for Research and Innovation, "Ministry of Education", in Saudi Arabia for funding this research (IFKSUOR3-328-1).

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

References

1. Holland, J.H. Genetic Algorithms. *Sci. Am.* **1992**, *267*, 66–73. [CrossRef]
2. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
3. Dorigo, M.; Di Caro, G. Ant colony optimization: A new meta-heuristic. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 6–9 July 1999; Volume 2, pp. 1470–1477.
4. Karaboga, D.; Basturk, B. On the performance of artificial bee colony (ABC) algorithm. *Appl. Soft Comput.* **2008**, *8*, 687–697. [CrossRef]
5. Yang, X.-S. *Firefly Algorithms for Multimodal Optimization BT—Stochastic Algorithms: Foundations and Applications*; Watanabe, O., Zeugmann, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 169–178.
6. Yang, X.S.; Deb, S. Cuckoo search via Lévy flights. In Proceedings of the 2009 World Congress on Nature and Biologically Inspired Computing, NABIC 2009, Coimbatore, India, 9–11 December 2009; pp. 210–214.
7. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
8. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
9. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [CrossRef]
10. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [CrossRef]
11. Abdel-Basset, M.; Mohamed, R.; Zidan, M.; Jameel, M.; Abouhawwash, M. Mantis Search Algorithm: A novel bio-inspired algorithm for global optimization and engineering design problems. *Comput. Methods Appl. Mech. Eng.* **2023**, *415*, 116200. [CrossRef]
12. Abdel-Basset, M.; Mohamed, R.; Jameel, M.; Abouhawwash, M. Nutcracker optimizer: A novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems. *Knowl.-Based Syst.* **2023**, *262*, 110248. [CrossRef]
13. Abualigah, L.; Yousri, D.; Abd Elaziz, M.; Ewees, A.A.; Al-qaness, M.A.A.; Gandomi, A.H. Aquila Optimizer: A novel meta-heuristic optimization algorithm. *Comput. Ind. Eng.* **2021**, *157*, 107250. [CrossRef]
14. Hayyolalam, V.; Pourhaji Kazem, A.A. Black Widow Optimization Algorithm: A novel meta-heuristic approach for solving engineering optimization problems. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103249. [CrossRef]
15. Kaur, S.; Awasthi, L.K.; Sangal, A.L.; Dhiman, G. Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. *Eng. Appl. Artif. Intell.* **2020**, *90*, 103541. [CrossRef]
16. Kaveh, A.; Farhoudi, N. A new optimization method: Dolphin echolocation. *Adv. Eng. Softw.* **2013**, *59*, 53–70. [CrossRef]
17. Braik, M.; Hammouri, A.; Atwan, J.; Al-Betar, M.A.; Awadallah, M.A. White Shark Optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems. *Knowl.-Based Syst.* **2022**, *243*, 108457. [CrossRef]
18. Abdollahzadeh, B.; Gharehchopogh, F.S.; Mirjalili, S. African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. *Comput. Ind. Eng.* **2021**, *158*, 107408. [CrossRef]
19. Jiang, Y.; Wu, Q.; Zhu, S.; Zhang, L. Orca predation algorithm: A novel bio-inspired algorithm for global optimization problems. *Expert Syst. Appl.* **2022**, *188*, 116026. [CrossRef]
20. Zamani, H.; Nadimi-Shahraki, M.H.; Gandomi, A.H. Starling murmuration optimizer: A novel bio-inspired algorithm for global and engineering optimization. *Comput. Methods Appl. Mech. Eng.* **2022**, *392*, 114616. [CrossRef]
21. Wang, L.; Cao, Q.; Zhang, Z.; Mirjalili, S.; Zhao, W. Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. *Eng. Appl. Artif. Intell.* **2022**, *114*, 105082. [CrossRef]
22. Eusuff, M.; Lansey, K.; Pasha, F. Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization. *Eng. Optim.* **2006**, *38*, 129–154. [CrossRef]
23. Qi, X.; Zhu, Y.; Zhang, H. A new meta-heuristic butterfly-inspired algorithm. *J. Comput. Sci.* **2017**, *23*, 226–239. [CrossRef]
24. Dhiman, G.; Kumar, V. Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications. *Adv. Eng. Softw.* **2017**, *114*, 48–70. [CrossRef]
25. Abualigah, L.; Elaziz, M.A.; Sumari, P.; Geem, Z.W.; Gandomi, A.H. Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer. *Expert Syst. Appl.* **2022**, *191*, 116158. [CrossRef]
26. Dehghani, M.; Montazeri, Z.; Trojovská, E.; Trojovský, P. Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems. *Knowl.-Based Syst.* **2023**, *259*, 110011. [CrossRef]
27. Rabie, A.H.; Mansour, N.A.; Saleh, A.I. Leopard seal optimization (LSO): A natural inspired meta-heuristic algorithm. *Commun. Nonlinear Sci. Numer. Simul.* **2023**, *125*, 107338. [CrossRef]
28. Mohammadi-Balani, A.; Dehghan Nayeri, M.; Azar, A.; Taghizadeh-Yazdi, M. Golden eagle optimizer: A nature-inspired metaheuristic algorithm. *Comput. Ind. Eng.* **2021**, *152*, 107050. [CrossRef]
29. van Laarhoven, P.J.M.; Aarts, E.H.L. *Simulated Annealing: Theory and Applications*; Springer: Dordrecht, The Netherlands, 1987; pp. 7–15. ISBN 978-94-015-7744-1.
30. Rashedi, E.; Nezamabadi-pour, H.; Saryazdi, S. GSA: A Gravitational Search Algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [CrossRef]
31. Hashim, F.A.; Houssein, E.H.; Mabrouk, M.S.; Al-Atabany, W.; Mirjalili, S. Henry gas solubility optimization: A novel physics-based algorithm. *Future Gener. Comput. Syst.* **2019**, *101*, 646–667. [CrossRef]

32. Kaveh, A.; Dadras, A. A novel meta-heuristic optimization algorithm: Thermal exchange optimization. *Adv. Eng. Softw.* **2017**, *110*, 69–84. [CrossRef]
33. Kaveh, A.; Mahdavi, V.R. Colliding bodies optimization: A novel meta-heuristic method. *Comput. Struct.* **2014**, *139*, 18–27. [CrossRef]
34. Zhao, W.; Wang, L.; Zhang, Z. Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. *Knowl.-Based Syst.* **2019**, *163*, 283–304. [CrossRef]
35. Kaveh, A.; Talatahari, S. A novel heuristic optimization method: Charged system search. *Acta Mech.* **2010**, *213*, 267–289. [CrossRef]
36. Kaveh, A.; Khayatazad, M. A new meta-heuristic method: Ray Optimization. *Comput. Struct.* **2012**, *112–113*, 283–294. [CrossRef]
37. Qais, M.H.; Hasanien, H.M.; Alghuwainem, S. Transient search optimization: A new meta-heuristic optimization algorithm. *Appl. Intell.* **2020**, *50*, 3926–3941. [CrossRef]
38. Abdel-Basset, M.; Mohamed, R.; Azeem, S.A.A.; Jameel, M.; Abouhawwash, M. Kepler optimization algorithm: A new meta-heuristic algorithm inspired by Kepler's laws of planetary motion. *Knowl.-Based Syst.* **2023**, *268*, 110454. [CrossRef]
39. Pereira, J.L.J.; Francisco, M.B.; Diniz, C.A.; Antônio Oliver, G.; Cunha, S.S.; Gomes, G.F. Lichtenberg algorithm: A novel hybrid physics-based meta-heuristic for global optimization. *Expert Syst. Appl.* **2021**, *170*, 114522. [CrossRef]
40. Abedinpourshotorban, H.; Mariyam Shamsuddin, S.; Beheshti, Z.; Jawawi, D.N.A. Electromagnetic field optimization: A physics-inspired metaheuristic optimization algorithm. *Swarm Evol. Comput.* **2016**, *26*, 8–22. [CrossRef]
41. Qais, M.H.; Hasanien, H.M.; Turky, R.A.; Alghuwainem, S.; Tostado-Véliz, M.; Jurado, F. Circle Search Algorithm: A Geometry-Based Metaheuristic Optimization Algorithm. *Mathematics* **2022**, *10*, 1626. [CrossRef]
42. Anita; Yadav, A. AEFA: Artificial electric field algorithm for global optimization. *Swarm Evol. Comput.* **2019**, *48*, 93–108. [CrossRef]
43. Muthiah-Nakarajan, V.; Noel, M.M. Galactic Swarm Optimization: A new global optimization metaheuristic inspired by galactic motion. *Appl. Soft Comput. J.* **2016**, *38*, 771–787. [CrossRef]
44. Formato, R.A. Central force optimization: A new nature inspired computational framework for multidimensional search and optimization. *Stud. Comput. Intell.* **2008**, *129*, 221–238. [CrossRef]
45. Javidy, B.; Hatamlou, A.; Mirjalili, S. Ions motion algorithm for solving optimization problems. *Appl. Soft Comput. J.* **2015**, *32*, 72–79. [CrossRef]
46. Azizi, M. Atomic orbital search: A novel metaheuristic algorithm. *Appl. Math. Model.* **2021**, *93*, 657–683. [CrossRef]
47. Glover, J.D.; Overbye, T.; Sarma, M.S. *Power System Analysis and Design*, 6th ed.; Cengage Learning: Boston, MA, USA, 2016.