mathematics

MDPI

Article

Hybrid Sine Cosine Algorithm for Solving Engineering Optimization Problems

Ivona Brajević [1,*], Predrag S. Stanimirović [2,3], Shuai Li [4], Xinwei Cao [5], Ameer Tamoor Khan [6] and Lev A. Kazakovtsev [3]

[1] Faculty of Applied Management, Economics and Finance, University Business Academy in Novi Sad, Jevrejska 24, 11000 Belgrade, Serbia
[2] Faculty of Sciences and Mathematics, University of Niš, Višegradska 33, 18000 Niš, Serbia
[3] Laboratory "Hybrid Methods of Modelling and Optimization in Complex Systems", Siberian Federal University, Prosp. Svobodny 79, 660041 Krasnoyarsk, Russia
[4] Department of Electronic and Electrical Engineering, Swansea University, Fabian Way, Swansea SA1 8EN, UK
[5] School of Business, Jiangnan University, Lihu Blvd, Wuxi 214122, China
[6] Department of Computing, The Hong Kong Polytechnic University, 11 Yuk Choi Rd, Hung Hom 999077, Hong Kong
* Correspondence: ivona.brajevic@mef.edu.rs

Abstract: Engineering design optimization problems are difficult to solve because the objective function is often complex, with a mix of continuous and discrete design variables and various design constraints. Our research presents a novel hybrid algorithm that integrates the benefits of the sine cosine algorithm (SCA) and artificial bee colony (ABC) to address engineering design optimization problems. The SCA is a recently developed metaheuristic algorithm with many advantages, such as good search ability and reasonable execution time, but it may suffer from premature convergence. The enhanced SCA search equation is proposed to avoid this drawback and reach a preferable balance between exploitation and exploration abilities. In the proposed hybrid method, named HSCA, the SCA with improved search strategy and the ABC algorithm with two distinct search equations are run alternately during working on the same population. The ABC with multiple search equations can provide proper diversity in the population so that both algorithms complement each other to create beneficial cooperation from their merger. Certain feasibility rules are incorporated in the HSCA to steer the search towards feasible areas of the search space. The HSCA is applied to fifteen demanding engineering design problems to investigate its performance. The presented experimental results indicate that the developed method performs better than the basic SCA and ABC. The HSCA accomplishes pretty competitive results compared to other recent state-of-the-art methods.

Keywords: sine cosine algorithm; artificial bee colony; hybrid algorithm; constrained design optimization

MSC: 68T20

Citation: Brajević, I.; Stanimirović, P.S.; Li, S.; Cao, X.; Khan, A.T.; Kazakovtsev, L.A. Hybrid Sine Cosine Algorithm for Solving Engineering Optimization Problems. *Mathematics* **2022**, *10*, 4555. https://doi.org/10.3390/math10234555

Academic Editor: Jakub Nalepa

Received: 6 November 2022
Accepted: 27 November 2022
Published: 1 December 2022

1. Introduction

Design optimization problems thrive in diverse engineering areas. Some examples are structural design problems in civil engineering, synthesis problems in chemical engineering, and mechanical problems in mechanical engineering [1]. Most engineering design optimization problems are hard to solve due to the complicated objective function and different types of constraints. An additional difficulty is that these problems might have objective functions with continuous and discrete variables.

An engineering design optimization problem tries to minimize or maximize an objective function with respect to design variables while satisfying the constraints established on the search space. This problem can be formulated as follows:

$$\text{minimize } f(x),\ x = (x_1, x_2, \ldots, x_D) \tag{1}$$

with respect to

$$l_i \leq x_i \leq u_i,\ i = 1, 2, ..., D$$

subject to

$$g_j(x) \leq 0,\ \ j = 1, \ldots, l$$
$$h_j(x) = 0,\ \ j = l+1, \ldots, m.$$

In Equation (1), a problem solution is denoted as x_i, $f(x)$ is the objective function, the number of design variables is denoted as D, inequality constraints are indicated as $g_j(x)$, while $h_j(x)$ denote equality constraints. In this problem, formulation design variables are discrete or continuous.

Diverse accurate and approximate algorithms have been proposed to solve constrained engineering design problems [2]. The efficacy of accurate optimization algorithms is solid for some design problems. However, these methods have to make certain presumptions that can not be justified in some cases. For instance, gradient-based algorithms belong to the group of exact algorithms, and these methods require the functions to be smooth enough. Exact methods are often inefficient when solving non-linear optimization problems which possess multiple local optima, since these problems are demanding to solve [3].

Since features of objective and constraint functions, such as modality, convexity, and smoothness, have limited the applicability of exact methods, numerous approximate algorithms have been developed in the last few decades [4–6]. Two groups of approximate algorithms are specific heuristics and metaheuristics. Specific heuristics are developed to solve a particular problem. Metaheuristics are applicable to a wide range of optimization problems. These algorithms use randomness in their search equations to help them avoid local optima and effectively explore the search space. Even though it is not guaranteed that the global optimum solution could be reached, metaheuristics can tackle optimization problems by finding solutions of sufficient quality in a reasonable time.

Many metaheuristic algorithms mimic natural metaphors to solve diverse challenging optimization problems. An essential class of these methods includes population-based metaheuristic algorithms, which reach appropriate solutions by iteratively choosing and combining solutions from the population. These algorithms are separated into five categories: evolutionary-based, swarm-based, physic-based, human-based, and maths-based metaheuristics [5]. Evolutionary algorithms, such as genetic and differential evolution algorithms, draw inspiration from nature and employ selection, recombination, and mutation search operators. Swarm intelligence methods are inspired by the collective behavior of animal societies. There are numerous prominent metaheuristic algorithms that belong to this category and some notable representatives include artificial bee colony (ABC) algorithm [7], particle swarm optimization (PSO) [8], cuckoo search (CS) [9], bat algorithm (BA) [10], firefly algorithm (FA) [11]. Human-based techniques, such as teaching learning-based algorithms, are motivated by distinct human-made happenings. Metaheuristics that mimic rules in physics, such as gravitational search algorithm (GSA) [12], fall into the category of physics-based algorithms. Math-based metaheuristics emulate mathematical rules. One of the popular recently proposed maths-based algorithms is the sine cosine algorithm (SCA) [13]. Additionally, in recent years applying machine learning to develop efficient metaheuristic algorithms has been studied [14,15]. Basic metaheuristic algorithms, after their creation, are often modified or hybridized with some other methods to enhance their performances for some problem classes.

The Sine Cosine Algorithm (SCA), proposed by Mirjalili, is inspired by the properties of trigonometric cosine and sine functions. This technique has several modified and hybridized variants that are used to solve various optimization problems. In [5] it was noticed that the SCA has to be combined with some other metaheuristic algorithm to solve specific real-world optimization problems. Even though this algorithm has many advantages, such as simplicity, adaptability, robustness, and reasonable execution time, it has been pointed out that the SCA may suffer from premature convergence. To escape from a local optimum point, the SCA should maintain a suitable diversity in the population. Hybridization has been a broadly established approach to produce differences among individuals during the search process [16].

Motivated by the above observations, a hybrid method (HSCA) is developed to solve complex engineering optimization problems. The foundations of this method are modified SCA and ABC metaheuristics. The proposed approach is collaborative hybrid algorithms with a sequential structure. In the HSCA, both methods, the modified SCA and ABC with two distinct search equations, execute alternatively until the convergence criterion is met. To arrive at a preferable equilibrium in the middle of exploitation and exploration abilities, a modified SCA search equation is proposed. In the developed SCA strategy, the base vector is selected randomly from the population and another difference vector is added to the base vector. Since the ABC metaheuristic has superior exploration ability, the ABC method with two search strategies was combined with the modified SCA. In the employed ABC variant, both ABC search equations are engaged in generating candidate solutions to provide suitable diversity in the population. Additionally, a mechanism to deal with the constraints based on the three feasibility rules is incorporated in the proposed HSCA. The proposed HSCA is tested on 15 challenging engineering design optimization problems. The achieved experimental results are compared to the basic ABC, SCA and recent prominent metaheuristics developed to tackle design problems.

The structure of this paper is as follows. A brief review of the SCA and ABC is presented in Section 2. Section 3 describes a novel hybrid sine cosine algorithm. Section 4 presents fifteen engineering design optimization problems. The experimental design and the reached results are given in Section 5. Concluding remarks are provided in Section 6.

2. A brief Literature Review

2.1. Sine Cosine Algorithm

The search strategy of the SCA is based on the cyclic pattern of cosine and sine functions [13]. The search equation employed to update the positions of solutions in the population is given as follows:

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^{t} + r_1 \sin(r_2) \left| r_3 y_j^t - x_{i,j}^t \right|, & r_4 < 0.5 \\ x_{i,j}^{t} + r_1 \cos(r_2) \left| r_3 y_j^t - x_{i,j}^t \right|, & r_4 \geq 0.5 \end{cases} \tag{2}$$

where $i = 1, \ldots, SP$, $j = 1, \ldots, D$, such that SP is the size of population and D is the number of design variables. In the Equation (2), $x_{i,j}$ is jth component of solution x_i at iteration t, y_j is jth component of the best achieved solution at iteration t. Specific control parameters, r_1, r_2, r_3 and r_4, are randomly chosen.

The parameter r_1 regulates the location of the novel solution. If $r_1 < 1$, the new solution will be located in the space between the current solution and the best solution reached so far. On the other hand, if $r_1 > 1$, the novel solution will be positioned outside that space. The distance towards or outwards the best-reached solution is determined by the parameter r_2, which has to be chosen from the range $[0, 2\pi]$. The value of r_3 represents a randomly chosen weight for the best-reached solution. The random parameter r_4 takes values from $[0, 1]$ and changes among the sine and cosine components in the Equation (2).

In the SCA, the optimization process starts with randomly created solutions. The method updates other solutions from the population using the Equation (2). The optimiza-

tion process of the SCA is finished when the iteration counter exceeds the maximal number of iterations (*MNI*).

Exploration, the process of generating solutions with a great deal of diversity, and exploitation, the process of searching a local area for good solutions found, are essential components of any metaheuristic algorithm. These two processes have to be well-balanced to achieve quality results. To balance the impact of the exploration and exploitation in the SCA, the r_1 value is updated during the search by the equation

$$r_1 = a - t\,\frac{a}{MNI}, \tag{3}$$

where t is the present cycle and a is the constant. The pseudocode of the SCA is given as Algorithm 1.

Algorithm 1 The pseudocode of the SCA

```
Initialize parameters SP, MNI, r1, r2, r3, r4
Generate a population of individuals xi, i = 1, 2, ..., SP randomly
t = 1
while t <= MNI do
    Record the best solution y reached so far
    for i = 1 to SP do
        Generate a new solution by employing the Equation (2) and evaluate it
    end for
    Update control parameters r1, r2, r3, r4
    t = t + 1
end while
```

The basic SCA is proposed to solve continuous optimization problems. After its development, several modified variants have been developed to enhance its performance [5]. These modified versions are divided into nine categories: binary, chaotic, opposition-based learning, orthogonal-based learning, Lévy flight, mutation, fuzzy, adaptive, and improved. For instance, binary SCA is proposed to discover the beneficial features [17]. The S-shaped and V-shaped transfer functions were employed to convert the continuous form of SCA into the respective binary algorithm. The chaotic oppositional SCA that uses an opposition-based learning mechanism was proposed to improve the performance of the SCA for solving global optimization in [18]. In [19], an original band selection method based on the SCA and Lévy flight was proposed.

The SCA algorithm is also combined with other metaheuristic methods to improve its performance in general or for a specific group of problems. For example, a hybrid metaheuristic method that combines the Lévy flight distribution, PSO, and SCA, for unconstrained numerical function optimization and constrained engineering design problems was proposed in [20]. This algorithm incorporates the death penalty technique to handle constraints. A joint procedure arising from SCA and ABC for solving thresholds at several levels has recently been developed [21]. In the proposed approach, the SCA was employed to achieve the global optimal solution, while ABC was used to shorten the search area. In general, the SCA method is used to solve optimization models from various fields, for instance, electrical engineering, image processing, computer engineering, classification, and control engineering [5].

2.2. Artificial Bee Colony Algorithm

The behavior of a honey bee swarm in search of food influenced the emergence of the ABC metaheuristic [7]. Three types of individuals exist in the swarm of ABC: employed, onlooker, and scout bees. Employed bees try to find sources of food. Onlookers search around the previously found sources of food. The scout bees are created from certainly employed bees that leave unfavorable sources to search for novel ones. The framework of the ABC is described as Algorithm 2.

Algorithm 2 The framework of the ABC algorithm

```
Initialization stage
t = 1
while t <= MNI do
    Employed bee stage
    Onlooker bee stage
    Scout bee stage
    t = t + 1
end while
```

In the initialization part of the ABC, the control parameters of the algorithm are initialized as well as a population of solutions x_i, $i = 1, 2, \ldots, SP$ is randomly generated in the search space. The ABC employed two standard control parameters, the maximum number of iterations (MNI) and the size of the population (SP), and the specific parameter *limit* which defines the number of attempts to abandon the food source. There are three main phases in the ABC algorithm. In both the employed and onlooker stages, the same search strategy is used to generate a candidate solution v_i from the current one x_i. This strategy is described by the equation

$$v_{ij} = x_{ij} + \varphi\,(x_{ij} - x_{kj}), \tag{4}$$

where j is a randomly selected index, $x_{i,j}$ is jth component of current solution x_i, x_k is a randomly picked solution that is distinct from x_i and φ is a random number between $(-1, 1)$. After a candidate solution is created, the boundary constraint-handling mechanism is used to v_i, and the greedy selection process is applied among x_i and v_i. In the employed stage, each solution from the population is subjected to the update process described by the Equation (4).

The same search strategy is used in the onlooker stage as in the employed stage. Also, the proportionate fitness selection is used to determine solutions that will be subjected to the update process. The greedy selection among candidates and the current solution decides whether the current solution will be updated. In the scouting phase, a solution that does not change over a predetermined number of trials is replaced with a randomly generated solution.

The ABC algorithm is one of the most notable swarm intelligence techniques. Many modified and hybridized ABC algorithms for continuous optimization problems have been developed since its invention. For instance, in [22], the ABC algorithm incorporated the constraint handling method which is based on three feasibility rules into its selection step with the intention to solve engineering optimization problems. A modified variant of the ABC algorithm for solving constrained design optimization problems, which included the crossover operator in the scout bee stage and the modified ABC search equation in the onlooker stage, was presented in [23]. A variant of the ABC algorithm combined with a dual-search mechanism and differential self-perturbation for solving engineering optimization problems was presented in [24]. A novel ABC variant, named I-ABC greedy, has recently been developed for solving mechanical engineering design problems. The I-ABC greedy algorithm integrates three modifications in the basic ABC: the use of an opposition-based learning concept, a modified ABC search equation, and a constraint-handling technique based on Deb's rules [25].

Despite the fact that the standard ABC algorithm was developed to tackle optimization problems in the continuous domain, nowadays, there are a lot of ABC variants for combinatorial, binary, and integer programming problems [26–28]. To be suitable for solving these types of problems, diverse encoding types, search equations, and selection operators are incorporated into the ABC algorithm.

3. Proposed Approach: HSCA

The performance of the SCA relies on its solution-seeking strategy which has characteristics that ensure exploration and exploitation abilities. The use of cosine and sine functions allows an agent to be created near another agent. This strategy ensures the exploitation of the space bounded among two agents. According to Equation (2), exploration is guaranteed since the algorithm can search outside the space bounded by the current solution and the best solution reached so far. Even though the SCA has good search ability, simple implementation, a small number of control parameters, and good execution time, it lacks the accuracy of the final solution in solving specific real-world optimization problems.

The problem of premature convergence is related to maintaining the solution variety during the search [16]. The diversity between the agents of a population is a prerequisite for exploration, and it is often high in the initial part of a search. It is expected that these differences among agents decrease as specific individuals find favorable search areas and the population progress towards the global optimum [29].

Since the search strategy of the ABC algorithm has a strong exploration ability, an approach that combines the SCA and ABC could efficiently sustain diversity in the population. Hence, a collaborative hybrid method is developed with a sequential structure based on the SCA and ABC. In the proposed HSCA, the SCA with an enhanced search strategy and the ABC algorithm with two distinct search equations are run alternately by working on the same population until the MNI is reached. The SCA with a modified search equation and the multi-strategy ABC algorithms employ a selection mechanism based on three feasibility rules for handling constraints.

A novel mutation operator is proposed and used in the SCA to achieve a better ratio between exploitation and exploration and thus a more productive search. This search strategy generates the candidate solution v_i from the current solution x_i according to the following equation:

$$v_{i,j}^{t+1} = \begin{cases} x_{n1,j}^t + r_1 \sin(r_2) \cdot |y_j^t - x_{i,j}^t| + rand_i \cdot (y_j^t - x_{n2,j}^t), & R_{i,j} < 0.5 \\ x_{n1,j}^t + r_1 \cos(r_2) \cdot |y_j^t - x_{i,j}^t| + rand_i \cdot (y_j^t - x_{n2,j}^t), & R_{i,j} \geq 0.5 \end{cases} \tag{5}$$

where x_{n1} and x_{n2} are two neighbourhood solutions chosen randomly from the population, y is the best solution achieved so far, $rand_i$ is a random number from $(0, 1]$ and $j = 1, \ldots, D$. As in the Equation (2), r_1 regulates the location of the candidate solution and is updated by the Equation (3), the parameter r_2 determines the distance towards or outwards to the best-reached solution and takes value from the range $[0, 2\pi]$ and parameter $R_{i,j}$ takes the value from range $[0, 1]$ and changes among the sine and cosine components in the Equation (5). In the proposed mutation strategy, the two vectors, x_{n1} and x_{n2} are picked randomly from the population, and the base vector is then chosen at random between these two solutions. In Equation (5), two difference vectors are added to the base vector. The second term of the Equation (5) is the same as the corresponding term of the Equation (2), except for the use of a random weight for the best-found solution. In the Equation (5) this parameter is not used since it may provide too much exploration and inefficient search. The best solution found and the randomly picked neighborhood solution forms the third term of the Equation (5). The use of the third term and the randomly picked base vector might lead to better perturbation.

The selection process based on three feasibility rules is incorporated in the proposed approach to steer the search toward feasible areas. Hence, after the candidate solution v_i is created by Equation (5) these rules are applied to determine the acceptability of the new solution v_i in the population.

Deb developed the feasibility rules using the following criteria to compare two solutions [30]: (1) if both solutions are feasible, the one which has a better objective function value is chosen, (2) any feasible solution is favored over any infeasible solution, (3) between

two infeasible solutions, the one with the lowest sum of constraint violations (CV) is chosen. This sum is calculated by

$$CV(x) = \sum_{j=1}^{l} \max\{0, g_j(x)\} + \sum_{j=l+1}^{m} |h_j(x)| \tag{6}$$

In the proposed ABC algorithm, two search strategies with a superior mixing ability coexist through the search process. The proposed ABC algorithm does not include three distinct phases, i.e., only the employed bee stage is performed. Each agent randomly chooses a search strategy for creating a candidate solution in this stage. After generating a potential new solution, Deb's rules are conducted to determine whether a novel solution will join the population.

Both search operators used in the developed ABC approach were previously used in the ABC variants for constrained optimization [22,23]. The first search strategy creates a candidate solution v_i from the current solution x_i according to the following equation:

$$v_{i,j}^{t+1} = \begin{cases} x_{i,j}^t + \varphi_i \left(x_{i,j}^t - x_{k,j}^t\right), & \text{if } R_j < MR \\ x_{i,j}^t, & \text{otherwise} \end{cases} \tag{7}$$

where φ_i is a uniform random number which takes value from $[-1, 1]$, t is current iteration number, x_k is another agent picked randomly from the population, R_j is a real number picked in a random way from $[0, 1]$, such that $j = 1, 2, \ldots, D$.

The quantity MR in Equation (7) is a modification rate control parameter that controls the potential changes of design parameters. The MR value is increased from 0.1 to the user-specified value $MRmax$ in the first $P \cdot MNI$ cycles to reduce the chances of skipping to exploit current agents from the population. This parameter is set to $MRmax$ in the lasting cycles. Hence the MR value is updated by

$$MR^{t+1} = \begin{cases} MR^t + \frac{(MR_{max} - 0.1)}{P \cdot MNI}, & \text{if } MR^t < MR_{max} \\ MR_{max}, & \text{otherwise} \end{cases} \tag{8}$$

where the control parameter P takes values from $(0, 1)$ and it regulates the number of starting cycles in which lower values of MR are employed.

The second search strategy used in the proposed multi-strategy ABC algorithm is described by

$$v_{i,j}^{t+1} = x_{i,j}^t + \varphi_i \left(x_{v,j}^t - x_{k,j}^t\right), \tag{9}$$

where x_v and x_k are different neighbourhood solutions chosen randomly from the population, φ_i is a uniform random number from the range $[-1, 1]$ and $j = 1, 2, \ldots, D$.

Let us denote the Equation (7) as S_1 and the Equation (9) as S_2. The pseudocode of the HSCA is given as Algorithm 3.

In each iteration of the HSCA, one of the three search strategies described by Equation (5), Equation (7), or Equation (9) is applied to each solution from the population with the intention to create a potential new solution. Each time when a candidate solution oversteps the boundaries of design variables, various values are created by the well-known boundary constraint handling technique described in [31]. Discrete design variables are addressed as though they were continuous. Each time when a novel solution is generated, these values are then rounded to the closest accessible discrete values.

Algorithm 3 The pseudocode of the HSCA

Initialize parameters SP,MNI,a, r_2, MR_{max}, P
Generate a random population of individuals x_i, $i = 1, 2, \ldots, SP$
$t = 1$
$r_1 = a$
$MR = 0.1$
while $t <= MNI$ **do**
  Record the best solution y reached so far
  Randomly assign strategy S_i, $i = 1, 2$ to solution x_i, $i = 1, 2, \ldots, SP$
  if (t mod 2 == 0) **then**
    for $i = 1$ to SP **do**
      Produce new solution v_i by employing the Equation (5) and evaluate it
      Apply selection process based on Deb's method among x_i and v_i
    end for
  else
    for $i = 1$ to SP **do**
      Produce new solution v_i by assigned strategy S_i and evaluate it
      Apply selection process based on Deb's method among x_i and v_i
    end for
  end if
  Update r_1 by using Equation (3)
  Update MR by using Equation (8)
  $t = t + 1$
end while

The HSCA employs four specific control parameters: the parameter a which represent the initial value of parameter r_1, the parameter r_2, the MR_{max} and P. Control parameters a and r_2 are used in the search strategy of the modified SCA algorithm, while parameters MR_{max} and P are employed in the Equation (7) of the proposed multi-strategy ABC algorithm. Also, two standard control parameters for algorithms with population, size of population SP, and the maximum number of iterations MNI are used in the HSCA. It can be seen from Algorithm 3 that in each cycle of the HSCA each agent is updated at most once. Hence, the computational time complexity of the proposed HSCA is $O(MNI \cdot SP \cdot f)$, where $O(f)$ is the computational time complexity of evaluating the objective function value for a specific problem f.

The performance of the HSCA relies on the employed three search strategies and the incorporated constrained handling technique. The good exploitation capacity of the modified SCA strategy is reached by using the current best solution in both difference vectors. In the second term of the Equation (5) the value of parameter r_1 is in the range [0, a], where a is a non-negative constant. Since the lower values of this parameter encourage the exploitation ability of this search equation, the value of parameter a is set to 0.75. The exploitation is further increased by gradually reducing the parameter r_1 during the search by using Equation (3). The control parameter r_2 takes value from the range $[0, 2\pi]$, as in the basic SCA [13]. The exploration capacity of the modified SCA search equation is guided by selecting the base vector randomly from the population and adding another difference vector to the base vector. To better maintain diversity in the population, the modified SCA optimizer is combined with the multi-strategy ABC algorithm. In the ABC search operator given by the Equation (7), the higher value of MR_{max} and lower value of P parameters are needed during the search, to produce excessive diversity in a solution. The ABC search strategy given by the Equation (9) does not use the MR_{max} parameter, i.e., by using this equation each component of an old solution is updated to create a potential new solution.

4. Design Problems

The performance of the proposed HSCA is evaluated on fifteen challenging structural and engineering design problems. These problems are: speed reducer, tension/compression

spring, pressure vessel, welded beam, gear train, stepped cantilever beam, multiple disk clutch brake, cantilever beam, helical compression spring, hollow shaft, hydro-static thrust bearing, Belleville spring, car side impact, gas transmission compressor, and reinforced concrete beam. A brief description of these problems is given as follows.

The speed reducer problem aims to optimize the weight. This problem has six continuous and one integer optimization variable and eleven inequality constraints. The most precise recorded rounded value to achieve (VTA) for this problem is 2994.470166, while its mathematical description can be found in [32].

Tension/compression spring problem tries to optimize the weight. This problem has three continuous optimization variables and four inequality constraints. The best recorded rounded VTA is 0.012665 [33]. The mathematical definition of this problem is presented in [22]. Optimization of materials, forming and welding costs are the goal of the pressure vessel problem. This problem has two design variables which have to be integer multiples of 0.0625, two continuous variables, and three inequality constraints. The global optimum for this problem is 6059.714335048436 [34]. The mathematical description of this problem is given in [22].

The target of the welded beam problem is to minimize the fabrication cost of the beam. This problem involves a non-linear objective function of four continuous variables and seven inequality restrictions. The best recorded rounded VTA for this problem is 1.724852 [33]. The mathematical description of this problem is given in [22].

The target of the gear train problem is to optimize the cost of the gear ratio of a gear train. This problem has four integer optimization variables. The optimal solution for this problem is 2.700857×10^{-12}. The model of this problem is presented in [22].

The target of the stepped cantilever beam problem is to minimize the volume of a stepped cantilever beam. This problem has four continuous and six discrete design variables and eleven inequality constraints. Value 63,893.43 is the best reported rounded VTA for this problem [35]. The mathematical description of this problem is given in [36].

Multiple disk clutch brake problem aims to optimize the mass. This problem has five integer variables and eight inequalities. The best achieved VTA for this problem is $2.3524245790 \times 10^{-1}$, while its mathematical description can be found in [37].

The cantilever beam problem tries to optimize the overall weight of a cantilever beam. This problem has five continuous variables and one inequality constraint. The global optimal solution is 1.339956367 [34]. The mathematical description of this problem is presented in [34].

The helical compression spring design problem tries to optimize the volume of spring steel. This problem has one continuous variable and two discrete variables, and eight inequality restrictions. The best reported rounded VTA for this problem is 2.658559, while its mathematical description can be found in [4].

The hollow shaft problem tries to minimize the weight of a hollow shaft. This problem has two continuous variables and two non-linear constraints. The best recorded rounded VTA is 2.370398, while its mathematical description can be found in [4].

The hydro-static thrust bearing problem tries to optimize bearing power loss. This problem has four continuous decision variables and seven inequality restrictions. The best-known VTA for this problem is $1.6254428092 \times 10^{3}$ [37]. A mathematical description of this problem is presented in [38].

The Belleville spring design problem aims to reach the minimum volume. It includes four continuous variables and seven non-linear constraints. The most precise recorded rounded VTA is 1.979675 [35]. In [38] the mathematical definition of this problem can be found.

The goal of the car side impact problem is to minimize the weight. This problem has nine continuous and two discrete variables and ten inequality constraints. The best calculated VTA is 22.84297, while its mathematical description is presented in [38].

The mathematical description of the gas transmission compressor problem is presented in [37]. The best known VTA for this problem is 2.9648954173×10^6. This problem has four design variables and one inequality constraint.

In the problem reinforced concrete beam the target is to design the beam for minimum cost. There are three discrete decision variables and two inequality constraints. Value 359.2080 is the best reported rounded VTA for this problem [35]. The mathematical description of this problem is presented in [36].

5. Experimental Study

This section presents the computational results achieved by the proposed HSCA for the considered fifteen structural and mechanical engineering design problems. Since the HSCA is based on the combination of SCA and ABC, these optimizers are compared with our developed method. A direct comparison, where the results achieved by the HSCA are compared with the results obtained by our implementation of the SCA and ABC is conducted. These algorithms have been implemented in the Java programming language. To promote a fair comparison, the same changes as those described in the original paper [22] are employed in our implementation of the ABC algorithm. Also, in our implementation of the SCA, Deb's rules are employed to favor feasible areas of total search space. Additionally, results obtained by the HSCA are compared with the results recently achieved for the same design problems employing other optimizers in preceding research studies. The results obtained by these metaheuristic optimizers are taken from the specialized literature.

The fixed control parameter values utilized by HSCA are the following: $SP = 30$, $a = 0.75$, $P = 0.3$, and $MR_{max} = 0.8$. Also, r_2 is a random number from $[0, 2\pi]$ [13].

The ABC algorithm used identical parameter values as those reported in the respective paper [22] ($SP = 30$, $MR = 0.9$, $SPP = 400$, $limit = SP \cdot D \cdot 5$).

In the SCA, the parameter SP was also set to 30, while this algorithm used identical specific parameter values as those reported in the original paper [13]. The number of function evaluations (NFEs) used by ABC, SCA, and HSCA was specified in each test case. For each design optimization problem, 50 runs were carried out.

5.1. Computational Results

In Tables 1–15, "Best", "Mean", "Worst", "SD", and "NFEs" symbolize, respectively, the best-found value, the average result, the worst-reached result, standard deviation values, and the number of function evaluations. The best results among the compared optimizers are shown in bold in these tables. The capacity of an optimizer to accomplish the best-known solution is decided by the best results. Algorithmic robustness or algorithmic stability is the capability of an algorithm to constantly converge to low values of objective functions without depending on the search space and initial solutions. Average and standard deviation results decide the stability or robustness of an optimizer. The NFEs represent the measure of the convergence rate. In Tables 1–4, NA denotes that results are not available in the respective paper.

Table 1. Performance of HSCA and other optimizers for the speed reducer design problem.

Algo.	Best	Mean	Worst	SD	NFEs
MBA	2994.482453	2996.769019	2999.652444	1.56	6300
EJAYA	**2994.471066**	2994.471070	2994.471097	7.1926×10^{-6}	17,000
IAPSO	**2994.47106614598**	**2994.47106614777**	**2994.47106615489**	$\mathbf{2.65 \times 10^{-9}}$	**6000**
I-ABC greedy	**2994.4710**	2994.6631	2994.902	1.87×10^{-12}	6500
CB-ABC	**2994.471066**	**2994.471066**	NA	2.48×10^{-7}	15,000
ABC	2994.47109	2994.471237	2994.471467	9.36×10^{-5}	**6000**
SCA	3051.055341	3182.229260	3360.733265	8.25×10^{1}	**6000**
HSCA	**2994.471066**	**2994.471066**	**2994.471066**	2.14×10^{-8}	**6000**

The speed reducer design problem was lately solved by the mine blast algorithm (MBA) [39], enhanced Jaya (EJAYA) [40], improved accelerated PSO (IAPSO) [32], crossover-based ABC (CB-ABC) [23] and I-ABC greedy [25]. The experimental results achieved by the HSCA, SCA, and ABC, and these optimizers are presented in Table 1.

The results given in Table 1 indicate that the HSCA is capable of greatly enhancing the capacity of the SCA and ABC to reach the optimal result, as well as the stability of these algorithms. In addition, exclusively, the IAPSO and HSCA could accomplish the best-rounded VTA in all runs. The HSCA converges faster to the optimal solution than all the compared optimizers, except for the IAPSO algorithm. The IAPSO and HSCA used the same NFEs to achieve these results.

The tension/compression spring design problem was lately solved by using the following optimizers: the MBA [39], elephant clan optimization (ECO) [41] atomic orbital search (AOS) [42], queuing search algorithm (QSA) [43], IAPSO [32], improved SCA (ISCA) [44], CB-ABC [23] and I-ABC greedy [25]. The statistical results achieved by the HSCA, SCA, ABC, and other metaheuristic algorithms are given in Table 2.

From the results presented in Table 2 it can be observed that only the HSCA reached the best-rounded VTA in all runs. The number of function evaluations needed by the HSCA to solve the tension/compression spring design problem is less or equal to that of the six optimizers (AOS, QSA, EJAYA, CB-ABC, ABC, and SCA), and higher compared to the MBA, ECO, IAPSO, and IABC-greedy. For the ISCA optimizer, the number of function evaluations is not reported in the respective paper.

Table 2. Performance of HSCA and other optimizers for the tension/compression spring design problem.

Algo.	Best	Mean	Worst	SD	NFEs
MBA	**0.012665**	0.012713	0.012900	6.30×10^{-5}	7650
ECO	**0.012665**	0.012709	0.01278	4.36×10^{-5}	11,484
AOS	**0.012665233**	0.012737649	0.013596859	0.000121146	200,000
QSA	**0.01266523279**	0.01266666921	0.01267436276	2.5895×10^{-6}	18,000
EJAYA	**0.012665**	0.012668	0.012687	4.6331×10^{-6}	15,000
IAPSO	**0.01266523**	0.013676527	0.01782864	1.573×10^{-3}	**2000**
ISCA	0.012667	NA	NA	NA	NA
I-ABC greedy	**0.012665**	0.013731147	0.012665	1.12×10^{-6}	**2000**
CB-ABC	**0.012665**	0.012671	NA	1.42×10^{-5}	15,000
ABC	**0.012665**	0.0129644	0.01344	2.09×10^{-4}	30,000
SCA	0.012724	0.012855	0.0130922	7.6976×10^{-5}	15,000
HSCA	**0.012665**	**0.012665**	**0.012665**	$\mathbf{3.49 \times 10^{-7}}$	15,000

Table 3. Performance of HSCA and other optimizers for the pressure vessel design problem.

Algo.	Best	Mean	Worst	SD	NFEs
ECO	6059.71448	6325.50262	7466.4068	398.99839	21,750
QSA	**6059.7143350484**	6078.3193234831	6370.7797127298	61.5629	25,000
IAPSO	**6059.7143**	6068.7539	6090.5314	14.0057	**7500**
ISCA	6059.7489	NA	NA	NA	NA
I-ABC greedy	**6059.7142**	6067.816	6086.982	19.044	8000
CB-ABC	**6059.714335**	6126.623676	NA	1.14×10^{2}	15,000
ABC	6060.974518	6282.680193	7017.127406	2.43×10^{2}	30,000
SCA	6119.1385	6379.0104	7101.9145	2.22×10^{2}	**7500**
HSCA	**6059.7143**	**6060.2669**	**6068.3329**	**1.61**	**7500**

Table 4. Performance of HSCA and other optimizers for the welded beam design problem.

Algo.	Best	Mean	Worst	SD	NFEs
FA	1.7312065	1.8786560	2.3455793	0.2677989	50,000
MBA	**1.724853**	**1.724853**	**1.724853**	$\mathbf{6.94 \times 10^{-19}}$	47,340
ECO	1.724856	1.72509	1.7256	4.0663×10^{-4}	21,924
AOS	**1.724852309**	1.725673538	1.732516334	0.024786984	200,000
QSA	**1.7248523085973**	**1.7248523085973**	**1.7248523085973**	1.1215×10^{-15}	18,000
EJAYA	**1.7248523086**	**1.7248523093**	**1.7248523105**	5.5631×10^{-10}	24,000
IAPSO	**1.7248523**	1.7248528	1.7248624	2.02×10^{-6}	12,500
I-ABC greedy	**1.724852**	1.724865	1.724910	1.92×10^{-5}	14,500
CB-ABC	**1.724852**	**1.724852**	NA	2.38×10^{-11}	15,000
ABC	**1.724852**	1.757544	1.896621	4.1×10^{-2}	30,000
SCA	1.7371620826	1.7571107369	1.7904976725	1.24×10^{-2}	**12,000**
HSCA	**1.7248523086**	**1.7248523086**	**1.7248523086**	1.46×10^{-12}	**12,000**

The pressure vessel design problem was lately solved by the following optimizers: ECO [41], QSA [43], IAPSO [32], ISCA [44] CB-ABC [23] and I-ABC greedy [25]. The statistical results reached by the HSCA, SCA, ABC, and other optimizers are presented in Table 3. As it can be noticed from Table 3, the best-rounded VTA was obtained by all compared optimizers, except for the ECO, ISCA, ABC, and SCA. It can also be observed that the HSCA achieved the smallest average and standard deviation results among all compared optimizers with the smallest number of NFEs.

The welded beam design problem was lately examined by employing diverse meta-heuristic algorithms, including the use of the FA [36], MBA [39], ECO [41], AOS [42], QSA [43], EJAYA [40], IAPSO [32], CB-ABC [23] and I-ABC greedy [25]. The statistical results achieved by the HSCA, SCA, ABC, and other optimizers are given in Table 4. In Table 4, NA denotes that results are not available in the respective paper.

Table 5. Performance of HSCA and other optimizers for the gear train design problem.

Algo.	Best	Mean	Worst	SD	NFEs
MBA	$\mathbf{2.700857 \times 10^{-12}}$	2.471635×10^{-9}	2.062904×10^{-8}	3.94×10^{-9}	1120
IAPSO	$\mathbf{2.700857 \times 10^{-12}}$	5.492477×10^{-9}	1.827380×10^{-8}	6.36×10^{-9}	800
I-ABC greedy	2.702×10^{-12}	6.452×10^{-9}	1.68×10^{-8}	5.29×10^{-10}	**60**
ABC	$\mathbf{2.700857 \times 10^{-12}}$	1.402023×10^{-8}	1.239641×10^{-7}	2.07×10^{-8}	750
SCA	$\mathbf{2.700857 \times 10^{-12}}$	3.771823×10^{-8}	5.041463×10^{-7}	7.97×10^{-8}	750
HSCA	$\mathbf{2.700857 \times 10^{-12}}$	$\mathbf{1.50506 \times 10^{-9}}$	$\mathbf{1.312515 \times 10^{-8}}$	2.30×10^{-9}	750

Table 6. Performance of HSCA and other optimizers for the stepped cantilever beam problem design problem.

Algo.	Best	Mean	Worst	SD	NFEs
FA	63,893.52578	64,144.75312	64,262.99420	175.91879	50,000
diversity-guided PSO	**63,893.43**	**63,893.43080**	**63,893.43080**	0.00000	50,000
I-ABC greedy	64,599.65	68,263.83	69,877.91	5.97×10^3	**9700**
ABC	**63,893.430796**	64,362.230828	68,333.430795	1.15×10^3	45,000
SCA	64,143.539011	64,640.851769	65,275.647478	2.57×10^2	45,000
HSCA	**63,893.430796**	**63,893.430796**	**63,893.430796**	$\mathbf{2.18 \times 10^{-11}}$	45,000

Table 7. Performance of HSCA and other optimizers for the multiple disc clutch brake design problem.

Algo.	Best	Mean	Worst	SD	NFEs
Rao-1	0.313657	0.321780	0.392071	0.009985	600
Rao-2	0.313657	0.315413	0.339999	0.00668	600
Rao-3	0.313657	0.319783	0.392071	0.016399	600
AOS	0.235242480	2.35×10^{-1}	2.35×10^{-1}	$\mathbf{6.45 \times 10^{-25}}$	200,000
IAPSO	0.313656	0.313656	0.313656	1.13×10^{-16}	**400**
I-ABC greedy	0.313656	0.313656	0.313656	1.27×10^{-16}	750
ABC	**0.2352424579**	0.2362422383	0.2440640500	2.16×10^{-3}	600
SCA	**0.2352424579**	0.2403031112	0.2547234739	4.79×10^{-3}	600
HSCA	**0.2352424579**	**0.2352424579**	**0.2352424579**	1.38×10^{-16}	600

Table 8. Performance of HSCA and other optimizers for the cantilever beam design problem.

Algo.	Best	Mean	Worst	SD	NFEs
AOS	1.339957	1.351954	1.491711	0.02499743	100,000
ABC	1.3399770	1.3400402	1.3403453	6.25×10^{-5}	**12,000**
SCA	1.3406087	1.3431266	1.3467008	1.57×10^{-3}	**12,000**
HSCA	**1.3399564**	**1.3399565**	**1.3399568**	$\mathbf{9.21 \times 10^{-8}}$	**12,000**

From Table 4 it can be seen that all considered optimizers obtain the best-rounded VTA, except the FA, ECO, and SCA. The HSCA, EJAYA, QSA, and MBA reached the mean results, equal to the best-rounded VTA among all compared techniques. Even though the HSCA obtained a worse standard deviation value than the QSA and MBS, the HSCA obtained these statistical results with the minimal of NFEs between considered optimizers.

The gear train design problem was solved by applying the MBA [39], IAPSO [35] and I-ABC greedy [25]. The comparative results obtained by the HSCA, SCA, ABC, and other approaches are presented in Table 5. From Table 5 it can be observed that all considered optimizers obtain the optimal solution, except the I-ABC greedy algorithm. The SCA, ABC, and HSCA reached the optimal solution with fewer NFEs than the MBA and IAPSO. Additionally, the proposed HSCA obtained the smallest average and standard deviation results of the remaining five considered optimizers.

The stepped cantilever beam problem was lately solved by employing the FA [36], diversity-guided PSO [35] and I-ABC greedy [25] approaches. The results reached by the HSCA, SCA, ABC, and other optimizers are presented in Table 6. The presented results show that only the HSCA, ABC, and diversity-guided PSO could obtain the best-rounded VTA. The ABC and HSCA reached the best-rounded VTA with fewer NFEs than the diversity-guided PSO. Also, only the HSCA and diversity-guided PSO accomplished the best-rounded VTA in all runs.

Table 9. Performance of HSCA and other optimizers for the helical compression spring design problem.

Algo.	Best	Mean	Worst	SD	NFEs
FA	2.658575665	4.3835958	7.8162919	4.6076313	75,000
diversity-guided PSO	**2.658559**	2.658890	2.660784	0.000611	**50,000**
Rao-1	**2.658559**	2.658675	2.659211	1.41×10^{-4}	75,000
Rao-2	**2.658559**	2.666750	2.699494	1.67×10^{-2}	75,000
Rao-3	**2.658559**	2.665386	2.699494	1.55×10^{-2}	75,000
ABC	**2.658559**	2.658910	2.673552	2.12×10^{-3}	75,000
SCA	2.658560	2.658636	2.659086	1.09×10^{-4}	75,000
HSCA	**2.658559**	**2.658559**	**2.658559**	$\mathbf{2.66 \times 10^{-15}}$	75,000

Table 10. Performance of HSCA and other optimizers for the hollow shaft design problem.

Algo.	Best	Mean	Worst	SD	NFEs
Rao-1	**2.370398**	**2.370398**	**2.370398**	**0**	5000
Rao-2	**2.370398**	**2.370398**	**2.370398**	**0**	5000
Rao-3	**2.370398**	**2.370398**	**2.370398**	**0**	5000
ABC	**2.370398244**	2.370443611	2.370652102	6.17×10^{-5}	**3000**
SCA	2.382173022	2.429340456	2.510303871	2.89×10^{-2}	**3000**
HSCA	**2.37039813**	**2.37039813**	**2.37039813**	1.59×10^{-10}	**3000**

Table 11. Performance of HSCA and other optimizers for the hydrostatic thrust bearing design problem.

Algo.	Best	Mean	Worst	SD	NFEs
EJAYA	1625.442764498248	1631.509586823626	1767.660483606390	26.27208859624	150,000
ABC	1723.9870064	1967.02318543	2351.79562005	1.34×10^{2}	**75,000**
SCA	1833.12880218	2085.518614380	2268.47587046	1.05×10^{2}	**75,000**
HSCA	**1625.44275908**	**1625.44275908**	**1625.44275908**	$\mathbf{1.99 \times 10^{-11}}$	**75,000**

The multiple-disc clutch brake design problem was lately solved by the Rao-1, Rao-1, Rao-3 [4], AOS [42], IAPSO [32], I-ABC greedy [25] algorithms. The comparative results obtained by the HSCA, SCA, ABC, and other methods are presented in Table 7. From Table 7 it can be seen that the best-known rounded VTA were obtained only by the ABC, SCA, and HSCA with the same number of NFEs. Moreover, the HSCA reached smaller average and standard deviation values concerning the other eight optimizers.

For the cantilever beam design problem, the global optimum result 1.339956367 was found in [34]. This design problem was lately solved by the AOS algorithm [45]. The statistical results of the AOS, ABC, SCA, and HSCA are given in Table 8. Results arranged in Table 8 confirm that only the proposed HSCA can find the best-rounded solution. Additionally, the HSCA obtained smaller average and standard deviation values than the other compared optimizers with the smallest NFEs.

Table 12. Performance of HSCA and other optimizers for the Belleville spring design problem.

Algo.	Best	Mean	Worst	SD	NFEs
MBA	**1.9796747**	1.984698	2.005431	7.7800×10^{-3}	**10,600**
diversity-guided PSO	**1.979675**	**1.979675**	**1.979675**	**0.000000**	50,000
ABC	1.9904011	2.0234035	2.1194233	2.77×10^{-2}	15,000
SCA	2.0130731	2.1238541	2.2075343	3.63×10^{-2}	15,000
HSCA	**1.9796747**	**1.9796747**	**1.9796748**	$\mathbf{3.84 \times 10^{-9}}$	15,000

Table 13. Performance of HSCA and other optimizers for the car side impact design problem.

Algo.	Best	Mean	Worst	SD	NFEs
FA	22.84298	22.89376	24.06623	0.16667	**20,000**
EJAYA	22.8429707	22.9439823	23.2619126	1.7098×10^{-1}	27,000
ABC	22.8437697	22.86236564	23.0056316	2.38×10^{-2}	24,000
SCA	23.0621090	23.3790495	23.7173399	1.44×10^{-1}	24,000
HSCA	**22.8429702**	**22.8429744**	**22.8429845**	$\mathbf{3.12 \times 10^{-6}}$	24,000

Table 14. Performance of HSCA and other optimizers for the gas transmission compressor design problem.

Algo.	Best	Mean	Worst	SD	NFEs
AOS	2,964,895.417	2,965,102.327	2,966,483.832	251.8360974	200,000
ABC	2,964,954.101	2,965,486.819	2,966,916.960	4.79×10^{2}	**8000**
SCA	2,965,653.755	2,972,099.03	2,983,981.646	3.52×10^{3}	**8000**
HSCA	**2,964,895.4173**	**2,964,895.4173**	**2964895.4173**	$\mathbf{2.46 \times 10^{-7}}$	**8000**

Table 15. Performance of HSCA and other optimizers for the reinforced concrete beam design problem.

Algo.	Best	Mean	Worst	SD	NFEs
FA	**359.2080**	460.706	669.150	80.73870	25,000
diversity-guided PSO	**359.2080**	**359.2080**	**359.2080**	**0.0000**	20,000
QSA	**359.20800**	**359.20800**	**359.20800**	5.74205×10^{-14}	10,000
ABC	**359.2080**	359.34379	362.6340	5.22×10^{-1}	**4500**
SCA	359.2116	360.5925	363.0331	1.13	**4500**
HSCA	**359.2080**	**359.2080**	**359.2080**	5.82×10^{-12}	**4500**

The helical compression spring design problem was lately solved by FA method [36], diversity-guided PSO [35], Rao-1, Rao-1, and Rao-3 [4]. Table 9 presents the comparative results obtained by the FA, diversity-guided PSO, Rao-1, Rao-1, Rao-3, ABC, SCA, and HSCA. From these statistical results, it can be noticed that most optimizers reached the best-rounded VTA. The only exceptions are the FA and SCA approaches which found slightly worse best results. The diversity-guided PSO converges to the best VTA faster than other compared optimizers. On the other hand, the proposed HSCA produced minimal mean and standard deviation results concerning the other seven metaheuristic algorithms.

The comparative results reached by the Rao-1, Rao-1, Rao-3 [4], ABC, SCA, and HSCA for the hollow shaft design problem are given in Table 10. It can be observed that each metaheuristic approach, except the SCA, was capable to obtain the best-rounded VTA. Between these optimizers, only Rao-1, Rao-1, Rao-3, and HSCA accomplished the best-rounded VTA in each run. The HSCA used the smallest NFEs to achieve these results.

Results reached by the EJAYA [40], ABC, SCA, and HSCA for the hydrostatic thrust bearing design problem are presented in Table 11. From Table 11 it can be observed that HSCA found the best-rounded VTA and the best average and standard deviation results among all compared optimizers. Also, the HSCA reached these results with the smallest NFEs.

The Belleville spring design problem was lately solved by the MBA [39] and diversity-guided PSO [35]. Table 12 presents the comparative results obtained by the MBA, diversity-guided PSO, ABC, SCA, and HSCA. The best VTA was obtained by the MBA, diversity-guided PSO, and HSCA optimizers. The MBA obtained the best VTA with the smallest NFEs, but only the diversity-guided PSO and HSCA reached the best-rounded solution in each run among all compared optimizers.

The car side impact design problem was lately solved by the FA [36] and EJAYA [40]. The statistical results achieved by the FA, EJAYA, SCA, ABC, and HSCA are arranged in Table 13. From Table 13 it is observable that only the HSCA can find the best-rounded solution. Also, the proposed HCSA obtained smaller average and standard deviation values than the other compared optimizers. The HSCA outperforms the EJAYA, ABC, and SCA in computational efficiency.

The gas transmission compressor design problem was lately solved by the AOS algorithm [45]. The comparative results of the AOS, ABC, SCA, and HSCA are presented in Table 14. From Table 14 it can be observed that the HSCA and AOS can find the best-rounded solution. Also, the HSCA performs more stable and converges faster to the optimal solution than the other compared optimizers.

The comparative results obtained by the FA method [36], diversity-guided PSO [35], QSA [43], ABC, SCA, and HSCA for the reinforced concrete beam design problem are presented in Table 15. From Table 15, it can be noticed that just the proposed HSCA, diversity-guided PSO, and QSA obtained the best-rounded solution in each run. The HSCA achieved the best VTA with the smallest NFEs in comparison with the other five optimizers.

Figure 1 demonstrates six convergence curves of the average solutions achieved by the ABC, SCA, and HSCA through 50 runs for specific design problems. This figure indicates that the proposed HSCA converges faster than the ABC and SCA on the chosen design problems. The statistical analysis derived by Wilcoxon's test among HSCA and SCA and among HSCA and ABC for these fifteen design problems at the 0.05 significance level obtained the p value 0.001. A summary of the results obtained by HSCA shows that our

proposed approach can significantly enhance the performance of SCA and ABC concerning the quality of the reached results, stability, and computational efficiency for all fifteen design problems.

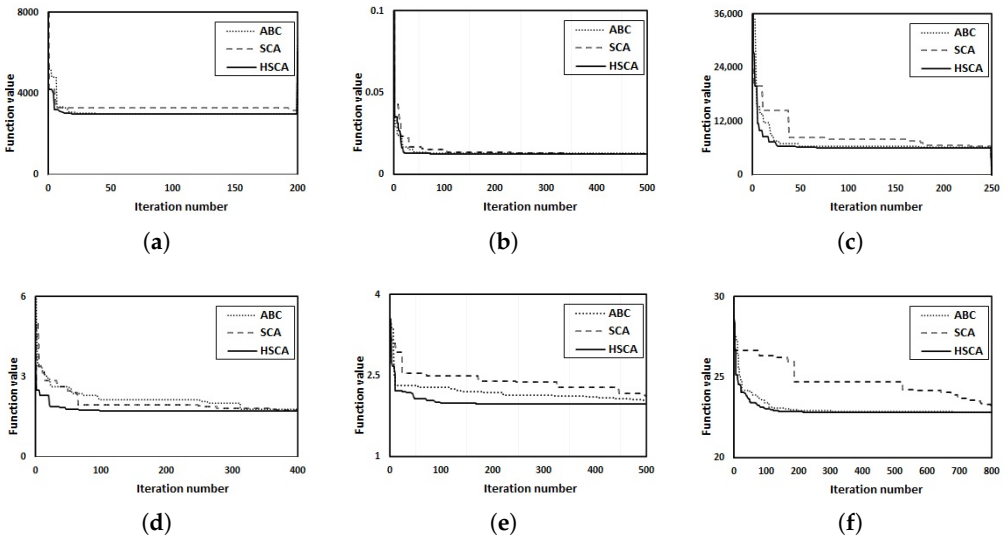

Figure 1. Convergence curves of the average solutions achieved by the ABC, SCA, and HSCA through 50 runs for some design problems: (**a**) Speed reducer, (**b**) Tension/compression spring, (**c**) Pressure vessel, (**d**) Welded beam, (**e**) Belleville spring, (**f**) Car side impact.

Table 16 presents results based on the performance of metaheuristic optimizers for fifteen design problems. In Table 16 "Best", "Mean" and "Worst" symbolize, respectively, optimizers which achieved the "best" best, the "best" mean and the "best" worst solutions, while "NFEs" represent optimizers that reached the "best" best solution for the smallest NFEs. From Table 16 it can be observed that the HSCA optimizer reached the "best" best, mean and worst solutions for twelve problems (speed reducer, pressure vessel, welded beam, gear train, stepped cantilever beam, multiple disc clutch brake, cantilever beam, hollow shaft, hydrostatic thrust bearing, car side impact, gas transmission compressor, and reinforced concrete beam) with the smallest NFEs. For the tension/compression spring problem, the IAPSO and I-ABC greedy obtained the "best" best solution with the lower NFEs than the HSCA. For the helical compression spring problem, the diversity-guided PSO reached the "best" best solution with the lower NFEs than the HSCA, while for the Belleville spring problem, the MBA obtained the "best" best solution with the lower NFEs than the HSCA. However, the HSCA showed good performance in solving these three design problems, since it reached the best-rounded VTA in each run with acceptable NFEs. It can be concluded that the comparative results with the other metaheuristic optimizers showed that the HSCA has efficient performance regarding the quality and stability of the results, and convergence speed.

5.2. Analyses

In this section, the behavior of the HSCA, ABC, and SCA when solving design problems is analyzed with regard to the diversity of a population. Diversity indicates differences between agents and preserving the proper amount of diversity during the search is an essential factor in avoiding local optimums and balancing exploration and exploitation. The exploration and exploitation are intertwined in the three mutation operators employed in the HSCA, and it is expected that these strategies can produce a suitable amount of diversity when it is needed.

To measure the diversity of the entire population, we employ the dimension-wise diversity metric, which is described by the next equations [46]:

$$Div_j = \frac{1}{SP}\sum_{i=1}^{SP} median_j - x_{i,j} \tag{10}$$

$$Div(t) = \frac{1}{D}\sum_{i=1}^{D} Div_j \tag{11}$$

Table 16. Results based on the performance of metaheuristic optimizers for fifteen design problems.

Problem	Best	Mean	Worst	NFEs
Speed reducer	EJAYA/IAPSO/ I-ABC greedy/ CB-ABC/HSCA	IAPSO/ CB-ABC/ HSCA	IAPSO/ HSCA	IAPSO/ HSCA
Tension/ compress spring	MBA/ECO/AOS/ QSA/EJAYA/IAPSO/ I-ABC greedy/ CB-ABC/ABC/HSCA	HSCA	HSCA	IAPSO/ I-ABC greedy/ MBA/ECO
Pressure vessel	QSA/IAPSO/ I-ABC greedy/ CB-ABC/HSCA	HSCA	HSCA	IAPSO/ HSCA
Welded beam	MBA/AOS/QSA/ EJAYA/IAPSO/ I-ABC greedy/ CB-ABC/ ABC/HSCA	MBA/QSA/ EJAYA/ CB-ABC/ HSCA	MBA/QSA/ EJAYA/ HSCA/	HSCA
Gear train	MBA/IAPSO/ABC/ SCA/HSCA/	HSCA	HSCA	ABC/SCA/ HSCA
Stepped cantilever beam	divers-guided PSO/ ABC/HSCA	divers-guided PSO/ HSCA	divers-guided PSO/ HSCA	ABC/ HSCA
Multi. disc clutch brake	ABC/SCA/ HSCA	HSCA	HSCA	ABC/SCA/ HSCA
Cantilever beam	HSCA	HSCA	HSCA	HSCA
Helical compress spring	divers-guided PSO/ Rao-1/Rao-2/Rao-3/ ABC/ HSCA	HSCA	HSCA	divers-guided PSO
Hollow shaft	Rao-1/Rao-2/Rao-3/ ABC/HSCA	Rao-1/Rao-2/ Rao-3/HSCA	Rao-1/Rao-2/ Rao-3/HSCA	ABC/HSCA
Hydr. thrust bearing	HSCA	HSCA	HSCA	HSCA
Belleville spring	divers-guided PSO/ MBA/HSCA	divers-guided PSO/HSCA	divers-guided PSO/HSCA	MBA
Car side impact	HSCA	HSCA	HSCA	HSCA
Gas trans. compressor	AOS/ HSCA	HSCA	HSCA	HSCA
Reinforced concrete beam	FA/QSA/ divers-guided PSO/ ABC/HSCA	divers-guided PSO/ QSA/HSCA	divers-guided PSO/ QSA/HSCA	ABC/HSCA

In the Equation (10), $median_j$ is the median of jth element in the entire population, $x_{i,j}$ is jth component of ith agent, SP is the total number of agents in the population. The calculated value Div_j represents the diversity of jth dimension. The Equation (11) is used to calculate the population diversity in tth iteration.

The diversity behavior of the ABC, SCA, and HSCA for the six design problems, speed reducer, tension/compression spring, pressure vessel, welded beam, Belleville spring, and car side-impact, is presented in Figure 2.

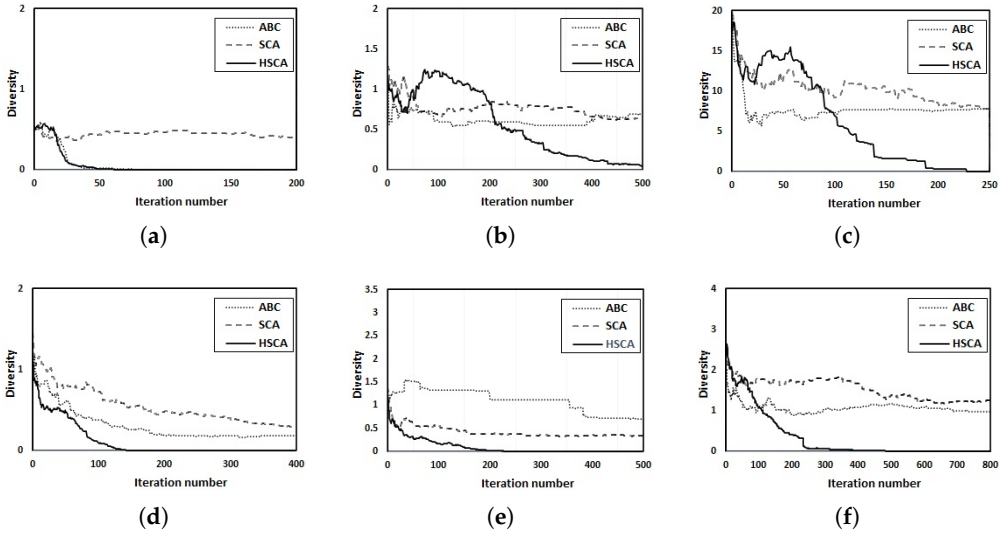

Figure 2. Diversity behavior of the ABC, SCA, and HSCA for some design problems: (**a**) Speed reducer, (**b**) Tension/compression spring, (**c**) Pressure vessel, (**d**) Welded beam, (**e**) Belleville spring, (**f**) Car side impact

From Figure 2 it can be seen that the diversity of the HSCA is high in the first part of the search process, and then it gradually decreases for all selected problems. High diversity in the initial iterations indicates that the HSCA may be capable of searching for a global optimum solution with greater accuracy. Further, low diversity in the final iterations may signify that promising areas are found and that fine adjustment of the reached results is provided. The population diversity of ABC and SCA is also higher in the initial generations, but decreases very slowly or remains on a similar level in later iterations for most of the chosen problems. Since encouraging high diversity in each stage of the search process might be inefficient, this diversity behavior may point to slow convergence and deficiency of precision of the ultimate solution.

The fact that HSCA outperformed ABC and SCA in tackling design problems and the presented diversity behavior of these algorithms indicates that the HSCA can maintain appropriate diversity through the search. Hence it can be concluded that the use of three mutation operators and Deb's rules in the HSCA enables a preferable balance between exploration and exploitation.

6. Conclusions

This paper proposes a hybridization of the sine cosine algorithm, called HSCA, for solving structural and mechanical engineering design optimization problems. The HSCA is a collaborative hybrid algorithm with a sequential structure. The modified SCA and ABC with two different search strategies execute alternately until the convergence criterium is met in the proposed method. The modified SCA search strategy is developed to reach a suitable balance between local exploitation and global exploration abilities. In the proposed equation, the global exploration is guided by choosing the base vector randomly from the population and adding another difference vector to the base vector. Also, the local exploitation is enhanced by using the current best solution in both difference vectors of this search operator. Further, modified SCA is combined with the multi-strategy ABC algorithm to better maintain diversity in the population. In the employed ABC variant, two ABC search equations with good global exploration abilities are engaged in creating candidate

solutions. The modified SCA and ABC employ a constraint handling technique based on Deb's rules to steer the search toward feasible areas.

The performance of the developed hybrid optimizer was explored on fifteen complex engineering problems. The experimental results of direct comparison of the HSCA with the standard ABC and SCA revealed that the HSCA performs significantly better than these two metaheuristics. The proposed approach has also established a highly competitive performance than the other optimizers studied by the preceding researchers in terms of the quality and stability of the results. Additionally, the HSCA showed an effective convergence speed in reaching these results. Therefore, the developed HSCA has a great capacity to deal with mixed-type design variables concurrently while satisfying various complicated design constraints.

The proposed hybrid method is easy to implement. The HSCA has a simple architecture that does not use extra components and is computationally inexpensive. This algorithm uses three search operators with distinct benefits which coexist during the search. The analyses indicated that the combination of the modified SCA mutation operator and the two ABC search equations with the use of three feasibility rules properly sustain the balance between exploration and exploitation for these design problems. The HSCA does not increase the number of control parameters in the SCA and the ABC optimizers. But a common complexity of finding the proper settings of algorithm-dependent parameters is present in the HSCA, as in lots of other metaheuristic optimizers. Additionally, the issue of more sophisticated control of the balance between exploitation and exploration during diverse stages of the search process exists in the proposed approach.

The considered fifteen design problems have different numbers of inequality constraints. To solve problems with equality constraints, the appropriate equality constraint handling technique needs to be employed. It is also important to note that the performance of the HSCA may be different depending on the constraint-handling technique employed. The efficiency of some other constraint-handling techniques for complex constraints will be investigated in future work. Extending the HSCA for solving multi-objective problems will also be studied.

Author Contributions: Conceptualization, I.B. and P.S.S.; methodology, I.B., P.S.S., S.L. and X.C.; validation, A.T.K. and L.A.K.; investigation, I.B., A.T.K. and L.A.K.; writing—original draft preparation, I.B., P.S.S., S.L., X.C, A.T.K. and L.A.K.; writing—review and editing, I.B., P.S., S.L., X.C, A.T.K. and L.A.K.; supervision, I.B. and P.S.S. 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: The source code used in this work is available from the corresponding author on request.

Acknowledgments: Predrag Stanimirović is supported by the Science Fund of the Republic of Serbia, (No. 7750185, Quantitative Automata Models: Fundamental Problems and Applications—QUAM). This work was supported by the Ministry of Science and Higher Education of the Russian Federation (Grant No. 075-15-2022-1121).

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

References

1. Martins, J.R.R.A.; Ning, A. *Engineering Design Optimization*; Cambridge University Press: Cambridge, UK, 2021.
2. Talibi, E.G. *Metaheuristics: From Design to Implementation*; John Wiley & Sons: Hoboken, NJ, USA, 2009.
3. Zhang, L.; Liu, L.; Yang, X.S.; Dai, Y. A Novel Hybrid Firefly Algorithm for Global Optimization. *PLoS ONE* **2016**, *11*, e0163230. [CrossRef] [PubMed]
4. Rao, R.V.; Pawar, R.B. Constrained design optimization of selected mechanical system components using Rao algorithms. *Appl. Soft Comput.* **2020**, *89*, 106141. [CrossRef]

5. Gabis, A.B.; Meraihi, Y.; Mirjalili, S.; Ramdane-Cherif, A. A comprehensive survey of sine cosine algorithm: Variants and applications. *Artif. Intell. Rev.* **2021**, *54*, 5469–5540. [CrossRef] [PubMed]
6. Khan, A.T.; Cao, X.; Li, S.; Katsikis, V.N.; Brajevic, I.; Stanimirovic, P.S. Fraud detection in publicly traded U.S firms using Beetle Antennae Search: A machine learning approach. *Expert Syst. Appl.* **2022**, *191*, 116148. [CrossRef]
7. Karaboga, D. *An Idea Based on Honey Bee Swarm for Numerical Optimization*; Technical Report-TR06; Erciyes University, Engineering Faculty, Computer Engineering Department: Kayseri, Turkey, 2005.
8. Kennedy, J.; Eberhart, R.C. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; pp. 1942–1948.
9. Yang, X.S.; Deb, S. Cuckoo Search via Lévy flights. In Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, 9–11 December 2009; pp. 210–214.
10. Yang, X.S. A New Metaheuristic Bat-Inspired Algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*; González, J.R., Pelta, D.A., Cruz, C., Terrazas, G., Krasnogor, N., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 65–74.
11. Yang, X.S. Firefly Algorithms for Multimodal Optimization. In *Stochastic Algorithms: Foundations and Applications, Proceedings of the 5th International Symposium, SAGA 2009, Sapporo, Japan, 26–28 October 2009*; Watanabe, O., Zeugmann, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 169–178.
12. Rashedi, E.; Nezamabadi-pour, H.; Saryazdi, S. GSA: A Gravitational Search Algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [CrossRef]
13. Mirjalili, S. SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowl. Based Syst.* **2016**, *96*, 120–133. [CrossRef]
14. Karimi-Mamaghan, M.; Mohammadi M.; Meyer P.; Karimi-Mamaghan A.M.; Talbi E.G. Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art. *Eur. J. Oper. Res.* **2022**, *296*, 393–422. [CrossRef]
15. Baldo, A.; Boffa, M.; Cascioli, L.; Fadda, E.; Lanza, C.; Ravera, A. The polynomial robust knapsack problem. *Eur. J. Oper. Res.* **2022**, *305*, 1424–1434. [CrossRef]
16. Ting, T.O.; Yang, X.S.; Cheng, S. Hybrid Metaheuristic Algorithms: Past, Present, and Future. In *Recent Advances in Swarm Intelligence and Evolutionary Computation*; Yang, X.S., Ed.; Recent Advances in Swarm Intelligence and Evolutionary Computation; Springer: Cham, Switzerland, 2015; pp. 71–83.
17. Taghian, S.; Nadimi-Shahraki, M.H. Binary sine cosine algorithms for feature selection from medical data. *Adv. Comput. Int. J.* **2019**, *10*, 1–10. [CrossRef]
18. Liang, H.; Cai, Z.; Wang, M.; Zhao, X.; Chen, H.; Li, C. Chaotic oppositional sine–cosine method for solving global optimization problems. *Eng. Comput.* **2022**, *38*, 1223–1239. [CrossRef]
19. Wang, M.; Wu, C.; Chen, M.; Chen, B.; Jiang, Y. A band selection approach based on Lévy sine cosine algorithm and alternative distribution for hyperspectral image. *Int. J. Remote Sens.* **2020**, *41*, 3429–3445. [CrossRef]
20. Chegini, S.N.; Bagheri, A.; Najafi, F. PSOSCALF: A new hybrid PSO based on Sine Cosine Algorithm and Levy flight for solving optimization problems. *Appl. Soft Comput.* **2018**, *73*, 697–726. [CrossRef]
21. Ewees, A.A.; Elaziz, M.A.; Al-Qaness, M.A.A.; Khalil, H.A.; Kim, S. Improved Artificial Bee Colony Using Sine-Cosine Algorithm for Multi-Level Thresholding Image Segmentation. *IEEE Access* **2020**, *8*, 26304–26315. [CrossRef]
22. Akay, B.; Karaboga, D. Artifcial bee colony algorithm for large–scale problems and engineering design optimization. *J. Intell. Manuf.* **2012**, *23*, 1001–1014. [CrossRef]
23. Brajevic, I. Crossover-based artificial bee colony algorithm for constrained optimization problems. *Neural. Comput. Appl.* **2015**, *26*, 1587–1601. [CrossRef]
24. Dong, C.; Xiong, Z.; Liu, X.; Ye, Y.; Yang, Y.; Guo, W. Dual-search artificial bee colony algorithm for engineering optimization. *IEEE Access* **2019**, *7*, 24571–24584. [CrossRef]
25. Sharma, T.K.; Abraham, A. Artifcial bee colony with enhanced food locations for solving mechanical engineering design problems. *J. Ambient Intell. Humaniz. Comput.* **2020**, *11*, 267–290. [CrossRef]
26. Akay, B.; Karaboga, D.; Gorkemli, B.; Kaya, E. A survey on the Artificial Bee Colony algorithm variants for binary, integer and mixed integer programming problems. *Appl. Soft Comput.* **2021**, *106*, 107351. [CrossRef]
27. Brajević, I. A Shuffle-Based Artificial Bee Colony Algorithm for Solving Integer Programming and Minimax Problems. *Mathematics* **2021**, *9*, 1211. [CrossRef]
28. Jooda, J.O.; Makinde, B.O.; Odeniyi, O.A.; Okandeji, M.A. A review on hybrid artificial bee colony for feature selection. *Glob. J. Adv. Res.* **2021**, *8*, 170–177.
29. Črepinšek, M.; Liu, S.H.; Mernik, M. Exploration and exploitation in evolutionary algorithms: A survey. *ACM Comput. Surv.* **2013**, *45*, 1–33. [CrossRef]
30. Deb, K. An Efficient Constraint-handling Method for Genetic Algorithms. *Comput. Methods Appl. Mech. Eng.* **2000**, *186*, 311–338. [CrossRef]
31. Brajević, I.; Stanimirović, P.S.; Li, S.; Cao, X. A Hybrid Firefly and Multi-Strategy Artificial Bee Colony Algorithm. *Int. J. Comput. Intell.* **2020**, *13*, 810–821. [CrossRef]
32. Guedria, N.B. Improved accelerated PSO algorithm for mechanical engineering optimization problems. *Appl. Soft Comput.* **2016**, *40*, 455–467. [CrossRef]
33. Brajević, I.; Ignjatović, J. An upgraded firefly algorithm with feasibility-based rules for constrained engineering optimization problems. *J. Intell. Manuf.* **2019**, *30*, 2545–2574. [CrossRef]

34. Yang, X.S.; Huyck, C.; Karamanoglu, M.; Khan, N. True Global Optimality of the Pressure Vessel Design Problem: A Benchmark for Bio-Inspired Optimisation Algorithms. *Int. J. Bio-Inspired Comput.* **2013**, *5*, 329–335. [CrossRef]
35. Kim, T.-H.; Cho, M.; Shin, S. Constrained Mixed-Variable Design Optimization Based on Particle Swarm Optimizer with a Diversity Classifier for Cyclically Neighboring Subpopulations. *Mathematics* **2020**, *8*, 2016. [CrossRef]
36. Gandomi, A.H.; Yang, X. S.; Alavi, A. H. Mixed variable structural optimization using Firefly Algorithm. *Comput. Struct.* **2011**, *89*, 2325–2336. [CrossRef]
37. Kumar, A.; Wu, G.; Ali, M.Z.; Mallipeddi, R.; Suganthan, P.N.; Das, S. A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. *Swarm Evol. Comput.* **2020**, *56*, 100693. [CrossRef]
38. Abderazek, H.; Yildiz, A.R.; Sait, S.M. Mechanical engineering design optimisation using novel adaptive differential evolution algorithm. *Int. J. Veh. Des.* **2019**, *80*, 285–329. [CrossRef]
39. Sadollah, A.; Bahreininejad, A.; Eskandar, H.; Hamdi, M. Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Appl. Soft Comput.* **2013**, *13*, 2592–2612. [CrossRef]
40. Zhang, Y.; Chi, A.; Mirjalili, S. Enhanced Jaya algorithm: A simple but efficient optimization method for constrained engineering design problems. *Knowl. Based Syst.* **2021**, *233*, 107555. [CrossRef]
41. Jafari, M.; Salajegheh, E.; Salajegheh, J. Elephant clan optimization: A nature-inspired metaheuristic algorithm for the optimal design of structures. *Appl. Soft Comput.* **2021**, *113*, 107892. [CrossRef]
42. Azizi, M. Atomic orbital search: A novel metaheuristic algorithm. *Appl. Math. Model.* **2021**, *93*, 657–683. [CrossRef]
43. Gupta, S.; Abderazek, H.; Yildiz, B.S.; Yildiz, A.R.; Mirjalili, S.; Sait, S.M. Comparison of metaheuristic optimization algorithms for solving constrained mechanical design optimization problems. *Expert Syst. Appl.* **2021**, *183*, 115351. [CrossRef]
44. Long, W.; Wu, T.; Liang, X.; Xu, S. Solving high-dimensional global optimization problems using an improved sine cosine algorithm. *Expert Syst. Appl.* **2019**, *123*, 108–126. [CrossRef]
45. Azizi, M.; Talatahari, S.; Giaralis, A. Optimization of engineering design problems using atomic orbital search algorithm. *IEEE Access* **2021**, *9*, 102497–102519. [CrossRef]
46. 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]