

2020

**A hybrid modified method of the sine cosine algorithm using latin hypercube sampling with the cuckoo search algorithm for optimization problems**

Siti Rosli

Hasliza Rahim

Khairul Najmy Abdul Rani

Ruzelita Ngadiran

R Ahmad

*See next page for additional authors*

Follow this and additional works at: <https://ro.uow.edu.au/dubaipapers>

---

**Recommended Citation**

Rosli, Siti; Rahim, Hasliza; Abdul Rani, Khairul Najmy; Ngadiran, Ruzelita; Ahmad, R; Yahaya, Nor; Abdul Malek, Mohd Fareq; Jusoh, Muzammil; Yasin, Mohd; Sabapathy, Thennarasan; and Andrew, Allan: A hybrid modified method of the sine cosine algorithm using latin hypercube sampling with the cuckoo search algorithm for optimization problems 2020, 1-23.  
<https://ro.uow.edu.au/dubaipapers/1189>







---

**Authors**

Siti Rosli, Hasliza Rahim, Khairul Najmy Abdul Rani, Ruzelita Ngadiran, R Ahmad, Nor Yahaya, Mohd Fareq Abdul Malek, Muzammil Jusoh, Mohd Yasin, Thennarasan Sabapathy, and Allan Andrew

Article

# A Hybrid Modified Method of the Sine Cosine Algorithm Using Latin Hypercube Sampling with the Cuckoo Search Algorithm for Optimization Problems

Siti Julia Rosli <sup>1,2</sup>, Hasliza A Rahim <sup>1,2,\*</sup>, Khairul Najmy Abdul Rani <sup>1,2</sup>,  
Ruzelita Ngadiran <sup>1,2</sup>, R. Badlishah Ahmad <sup>2</sup>, Nor Zakiah Yahaya <sup>3</sup>,  
Mohamedfareq Abdulmalek <sup>4</sup>, Muzammil Jusoh <sup>1,2</sup>, Mohd Najib Mohd Yasin <sup>1,2</sup>,  
Thennarasan Sabapathy <sup>1,2</sup> and Allan Melvin Andrew <sup>1,5</sup>

<sup>1</sup> Advanced Communication Engineering, Centre of Excellence (CoE), Universiti Malaysia Perlis (UniMAP), 01000 Kangar, Perlis, Malaysia; sitijulia@unimap.edu.my (S.J.R.); khairulnajmy@unimap.edu.my (K.N.A.R.); ruzelita@unimap.edu.my (R.N.); muzammil@unimap.edu.my (M.J.); najibyasin@unimap.edu.my (M.N.M.Y.); thennarasan@unimap.edu.my (T.S.); allanmelvin@unimap.edu.my (A.M.A.)

<sup>2</sup> Faculty of Electronic Engineering Technology, Universiti Malaysia Perlis (UniMAP), 02600 Arau, Perlis, Malaysia; badli@unimap.edu.my

<sup>3</sup> Physics Section, School of Distance Education, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia; norzakiah@usm.my

<sup>4</sup> Faculty of Engineering and Information Sciences, University of Wollongong in Dubai, Dubai 20183, UAE; MohamedFareqMalek@uowdubai.ac.ae

<sup>5</sup> Faculty of Electrical Engineering Technology, Universiti Malaysia Perlis (UniMAP), 02600 Arau, Perlis, Malaysia

\* Correspondence: haslizarahim@unimap.edu.my

Received: 12 August 2020; Accepted: 16 October 2020; Published: 27 October 2020



**Abstract:** The metaheuristic algorithm is a popular research area for solving various optimization problems. In this study, we proposed two approaches based on the Sine Cosine Algorithm (SCA), namely, modification and hybridization. First, we attempted to solve the constraints of the original SCA by developing a modified SCA (MSCA) version with an improved identification capability of a random population using the Latin Hypercube Sampling (LHS) technique. MSCA serves to guide SCA in obtaining a better local optimum in the exploitation phase with fast convergence based on an optimum value of the solution. Second, hybridization of the MSCA (HMSCA) and the Cuckoo Search Algorithm (CSA) led to the development of the Hybrid Modified Sine Cosine Algorithm Cuckoo Search Algorithm (HMSCACSA) optimizer, which could search better optimal host nest locations in the global domain. Moreover, the HMSCACSA optimizer was validated over six classical test functions, the IEEE CEC 2017, and the IEEE CEC 2014 benchmark functions. The effectiveness of HMSCACSA was also compared with other hybrid metaheuristics such as the Particle Swarm Optimization–Grey Wolf Optimization (PSOGWO), Particle Swarm Optimization–Artificial Bee Colony (PSOABC), and Particle Swarm Optimization–Gravitational Search Algorithm (PSOGSA). In summary, the proposed HMSCACSA converged 63.89% faster and achieved a shorter Central Processing Unit (CPU) duration by a maximum of up to 43.6% compared to the other hybrid counterparts.

**Keywords:** computational intelligence; sine cosine; cuckoo search; metaheuristic; optimization algorithm; hybrid algorithm

## 1. Introduction

Metaheuristics is a process of designing heuristic procedures to identify the ideal solution to complex issues in the optimization algorithm and provides quality results. The existing literature shows that various techniques produce effective components for the algorithm and generate a high volume of information during the iteration process. The capabilities of the techniques assist designers to obtain high performance of value parameters and are easily involved in an arrangement strategy to facilitate the decision-making process of a metaheuristic algorithm [1,2].

Various Evolutionary Algorithms (EAs) [3] have been introduced due to their ability to solve problems using a population-based stochastic search technique dependent on design parameters, such as the Genetic Algorithm (GA) in 1960 [4], Ant Colony Optimization (ACO) in 2000 [5], Differential Evolution (DE) [6], Artificial Bee Colony (ABC) by Karaboga in 2005 [7], Particle Swarm Optimization (PSO) in 2010 [8], Ageist Spider Monkey Optimization (ASMO) in 2016 [9], Grey Wolf Optimizer (GWO) in 2014 [10], Squirrel Search (SS) in 2018 [11], and Polar Bear Optimization (PBO) in 2017 [12]. Primarily, nature's behavioral intelligence has been continuously used in recent years and has gradually gained popularity. As suggested by Mirjalili in 2016, SCA has become a recent population-based metaheuristic optimization algorithm [13], which has also received significant attention from many researchers to solve optimization problems. For issues related to SCA, the selection features of classical SCA have been used in various attempts to increase exploration and exploitation efficiency. Specifically, SCA has been used as the main algorithm to enhance circuit design requirements in order to reduce the area occupied by circuit transistors [14].

The Sine Cosine Algorithm's (SCA) main function is to improve population diversity, and it has been combined with specific equations through hybridization with the Enhanced Brain Storm (EBS) to develop the Enhanced Brain Storm–Sine Cosine Algorithm (EBSSCA) optimizer [15]. It was also used to perform feature selection for dimensionality minimization to shorten the computational time and enhance both the classification and the investigation in the search space [16]. Generally, some random and flexible parameters are present in the utilization and investigation function of SCA to facilitate search procedures at both the local and global levels. In addition to understanding the random traversing of the search space on a global scale, a balance is present in the choice of either exploration or exploitation for a good solution at the local scale. Furthermore, implementing a constant switching possibility and the bound functions of magnitude functions (which ranged from negative one (−1) to positive one (+1)) of the SCA [17] led to the susceptibility of the minima or maxima to the search process.

Hybridization and modification of the SCA are becoming popular among researchers who want to utilize the benefits of SCA. However, the original SCA often exhibits low-accuracy enhancement and impact reduction of the local minima due to constraints in the exploration and exploitation mechanisms. According to Rizk-Allah [18], the Multi-Orthogonal Sine Cosine Algorithm (MOSCA) is capable of minimizing unbalanced exploitation and local optima SCA trapping. Notably, this study enhanced the exploration capability and solution quality of SCA by developing a local search algorithm and boosting its exploitation tendencies. However, its limitations include premature convergence that curved towards several function inaccuracies of the SCA phase. Suid et al. [19] synthesized the original SCA to improve its exploration and exploitation based on a nonlinear strategy to determine the algorithm strength. The Improved SCA (ISCA) algorithm not only exhibited significant investigation ability, but it was also capable of preventing local optima stagnation issues. Specifically, the ISCA outperformed some other compatible functions at minimization, which included the functions of fixed-dimension multimodal benchmark, fixed-dimension multimodal benchmark, and unimodal benchmark. However, the performance of ISCA was not higher than the selected functions due to its unstable optimization operations.

Modification tasks were performed using several techniques, including the modification of a set of candidate solutions in SCA for Opposition Based Learning (OBL) [20]. The concept of this modification was the development of more efficient approaches in different fields such as intelligent and expert

systems. Although the modified method was able to enhance the exploration of the search space, the range value of the population led to a drawback in the number of function optimization processes. Sindhu et al. [21] proposed a Modified SCA (MSCA) using the elitism strategy and upgraded the new solution by selecting the ideal features to improve SCA accuracy. The aim of these modifications was to optimize the algorithm's quality with higher precision and reduce the number of features using wrapper-based feature selection. Based on the ten tested benchmark datasets in this experiment, it was proven that MSCA achieved higher efficiency with a lower number of features. This method was also compared with other metaheuristic algorithms, such as GA, PSO, and basic SCA.

Long et al. [22] presented a new technique using SCA by modifying the position-updating equation and the conversion parameter research. This improved the SCA in terms of the position-updating weight to stabilize investigation and utilization. The modification of SCA was made on the Gaussian function to reduce the non-linear effect. Following that, the enhanced SCA was tested using 24 benchmark functions and five-dimensional values. The mean values and CPU frequency of the ISCA were small, and its improved performance was exhibited through faster convergence compared to the basic SCA. However, the ISCA showed the lowest minimization in the F6 function in comparison to the Whale Optimization Algorithm (WOA) and Teaching-Learning-Based Optimization (TLBO) algorithm.

The improvement in SCA with PSO [23] led to advantages over the original SCA and other metaheuristics in terms of its precision and duration of calculation. This work improved the convergence and reduced the possibility of being confined to a local optimum solution or optimization issues, which contributed to the maximum length of continuous substrings. Not all of the test functions provided a minimum standard deviation, while exploitation in finding the substrings was poor. Qu et al. [24] improved the SCA by decreasing the conversion parameter and inertia weight, and including the use of random optimal individuals with greedy Levy mutation techniques. Chegini et al. [25] developed a hybrid PSOSCA and Levy flight, which allowed the optimizer to identify optimal solutions. Notably, PSOCALF was more effective than the basic PSO and other algorithms in preventing the confinement of the local minimum. However, not all optimization stages in the convergence behavior were improved, besides the early convergence curves trapped in the local minimum. Nenavath et al. [26] proposed a combination of two algorithms—SCA and PSO. The proposed algorithm was evaluated using 23 notable benchmark issues within IEEE CEC 2017 standards [27] and IEEE CEC 2014 [28]. The proposed method had a better performance in utilization and investigation and required less time to find local minima and maxima. However, a limitation remained in this method due to insufficient internal memory, not guaranteeing an accurate convergence detection.

Singh et al. [29] hybridized GWO and SCA, in which the grey wolves' natural movements were used to update the location of SCA. The proposed GWOSCA performance was compared with the SCA, GWO, Mean GWO (MGWO), Hybrid Approach GWO (HAGWO), WOA, PSO, and Ant Lion Optimizer (ALO). Following this conception, the highest capability quality avoided the local optima and performed better than other metaheuristics. Slow convergence was still an issue in the hybridization, which could become trapped in the partitioning procedure. Moreover, Nenavath et al. [30] adopted DE into SCA (DESCA) to solve the optimization problem and object identification; DESCAs proved that the hybridization of the two algorithms exhibited an ability to prevent the local optima through matured convergence compared to the basic DE and SCA. However, provided that the SCA did not lead to an optimal solution to more complex problems, it had an impact on the mean, standard deviation, and declined value to every independent run (low probability) to determine the implication of the statistical test outcomes compared to other hybrid algorithms. Table 1 presents all of the aforementioned limitations.

One of the most recent advanced techniques of Swarm Intelligence (SI) developed is the CSA, which was used by Chi [31] and Fister et al. [32] as a metaheuristic optimization method, while Yang and Deb [33] led its evolution. Primarily, the results of the original CSA were more efficient than PSO and GA. Additionally, the CSA was further enhanced by adding series information through ideal global solutions to achieve a higher probability of finding a cuckoo egg in the host's nest.

The design of the optimum structure based on the CSA and its continuous objective function was recently developed by Umenai et al. [34], while its limitations were presented in the methods proposed by [35–37]. The proposed method was not able to guarantee the detection of precise convergence for the suggested objective function and had an immature convergence and inaccuracy phase for some benchmark test functions. This method was also trapped in the local optima caused by the number of search agent problems.

**Table 1.** Comparisons of state-of-the-art works.

Ref	Proposed Method	Enhancement	Limitation
[20]	Modified SCA	(a) improved version of SCA with consideration of OBL. (b) increased accuracy of the optimal solution and ability to explore the search space in optimization process.	(a) unable to solve the dimension problem.
[21]	ISCA by modified position-updating equation	(a) modified the position updating weight to balance the exploration and exploitation of SCA. (b) modified the conversion parameter for decreasing the non-linear strategy based on Gaussian function.	(a) means of CPU times of ISCA were small values.
[29]	Hybrid GWO-SCA	(a) improved by using position update equations of SCA. (b) highest capability to avoid the local optima and considerably superior to other metaheuristics.	(a) the high number of dimensions is worse than in the WOA and TLBO algorithm.
[18]	Hybrid SCA—Multi-Orthogonal Search Strategy	(a) enhanced exploration capability. (b) boosts exploitation tendencies. (c) robust, statistically sound and quick convergence.	(a) immature convergence inaccuracy phase for some benchmark functions.
[19]	ISCA with new updating mechanism	(a) improved exploration and exploitation based on nonlinear strategy to find the algorithm strength. (b) proposed algorithm able to get away from local optima stagnation problem.	(a) not all selected functions are minimized, and ISCA shows instability in percentage performance compared to other optimization methods.
[23]	Hybrid SCA-PSO	(a) good performance in accuracy and computational time. (b) improves the convergence and the tendency.	(a) poor exploitation in finding the longest consecutive substrings. (b) not all the test function provided minimum standard deviation.
[24]	Modified SCA—Neighborhood Search and Greedy Levy Mutation	(a) improved SCA through three optimization techniques by decreasing the conversion parameter and inertia weight, using random optimal individuals, and greedy Levy mutation techniques. (b) effectively avoids becoming trapped in a local optimum; has faster convergence; has higher optimization accuracy. Twenty benchmark test functions were applied to verify the performance.	(a) still in infancy stage, and the complexity was greatly increased.
[30]	Hybrid SCA-DE	(a) solved the optimization problem and object tracking.	(a) SCA does not lead to optimal solutions to complex problems when used on some benchmark functions. (b) does not compare each of the independent runs (chance of low probability).
[26]	Hybrid SCA-PSO	(a) good performance method in exploiting the optimum and has advantages in exploration.	(a) did not guarantee an accurate convergence detection.
[25]	Hybrid PSO-SCA and Levy flight	(a) allows the design to be searched further to find optimal solution.	(a) not all stages of the optimization in convergence behavior improve. (b) early convergence and trapped in local minimum.
[16]	Improved SCA with elitism strategy and new updating mechanism	(a) to improve SCA accuracy by selecting the best features using the elitism strategy and new solution.	(a) immature convergence curve for minimization fitness features.
[22]	Modified SCA	(a) local optima with faster convergence. (b) solving three constrained real engineering design problems.	(a) immature convergence. (b) does not affect the configuration of the original SCA.

The MSCA using the LHS approach and HMSCACSA was not used by any of the past studies. Hybridization was possibly the key feature to improve the effectiveness of the traditional SCA. Furthermore, HMSCACSA was used to modify the conventional SCA with one new operation LHS to facilitate a local search method and enhance the algorithm intensification. In this study, LHS with a control method was combined with the local search method to improve the capability of the global search. Provided that LHS exhibited a strong global search ability, a symmetric Latin hypercube design, which is a variant of LHS, was used for population initialization [38]. The setting method (length of each dimension), which generated the hypercube size, could be determined by the location for the search agent. This was followed by the transformation of SCA into MSCA to minimize the fitness of the current best nest selected by random walk solutions. The hybrid and modified SCA also led to some improvements in terms of local and global search precision to exploit the sine cosine functions. The convergence curved during faster runs. Meanwhile, the stability of optimization problems was

related to the phases during the determination of the optimum values for the parameters of a system from all probable values, including minimum profitability. This stability was compared with the traditional SCA.

The simple application and cost-efficient computational overhead of the MSCA and the HMSCACSA algorithms may provide a solution to compound optimization tasks. The main contributions of this study are as follows:

- (1) We include the modified LHS (originating from the SCA) operation within the MSCA.
- (2) We develop an emerging sine-cosine method based on the hybrid CSA, known as HMSCACSA. This gives the fastest convergence curve, improves the capabilities in a global search domain with sensitive parametric analysis, and minimizes CPU time.
- (3) We provide a performance comparison of the HMSCACSA with recent state-of-the-art hybrid metaheuristic algorithms including PSOGWO [39], PSOABC [40], and PSOGSA [41].

The literature review indicated that the modification and hybridization of the SCA comprises information on the exploration and exploitation of global optimal solutions. This proposed method was tested using six benchmark test functions from the IEEE CEC 2017 standards [27] and IEEE CEC 2014 [28] to validate the performance of feature optimization. The mean and standard deviations were tabulated to evaluate the performance of the suggested method for the benchmark of the optimization of the mathematical function. Few feature optimizations were present with an improved CPU time and faster convergence speed, while the minimum fitness optimization was tested to illustrate the improvement that the proposed method represented. The diverse sizes of parameter sensitivity for the population, dimension, and diversity plot were fine-tuned to show the improved capabilities of the global domain search of the SCA. Three hybrid algorithms—hybrid PSOABC, PSOGWO and PSOGSA—were benchmarked against the proposed MSCA and HMSCACSA.

The rest of this paper is organized as follows: The basic methodology used to design the proposed algorithm is outlined in Section 2. In Section 3, the proposed MSCA using LHS is described in detail, including how it is implemented and working principles. The second method, which is the HMSCACSA, is explained in Section 4. Section 5 discusses the performance of the proposed algorithms using six benchmark mathematical functions tested within the IEEE CEC 2014 [28] and a recent benchmark test function within the IEEE CEC 2017 [27]. The analysis, simulation, and statistical results are presented in this section to report the efficiency of both methods. Section 6 presents the conclusions and the planned future work.

## 2. Related Work

Since the hybrid and modified SCA are becoming increasingly popular, the CSA and SCA are discussed here based on the working principles and the basic parameters for easy understanding.

### 2.1. Original Sine Cosine Algorithm (SCA)

A recent development of the SCA metaheuristic algorithm was made according to the mathematical sine cosine functions, followed by its implementation for the utilization and investigation of an optimized use on the global level. This algorithm was created by Mirjalili in 2016 [13]. The SCA also corresponds to other population-based metaheuristic algorithms, in which it begins with a random distribution of a category of solutions. The placement improvement in every search agent in this algorithm is conducted through Equations (1) and (2).

$$X_i^t(j+1) = \left\{ X_i^t(j) + r_1 \times \sin(r_2) \times |r_3 P_i^t - X_i^t(j)| \right\}, r_4 < 0.5 \quad (1)$$

$$X_i^t(j+1) = \left\{ X_i^t(j) + r_1 \times \cos(r_2) \times |r_3 P_i^t - X_i^t(j)| \right\}, r_4 \geq 0.5 \quad (2)$$

where, based on (1) and (2),  $X_i^t(j)$  refers to the placement of  $j^{\text{th}}$  search agent along the  $i^{\text{th}}$  dimension after the  $t^{\text{th}}$  iteration, while  $P_i$  refers to the placement of the destination point along the  $i^{\text{th}}$  dimension.

Furthermore,  $r_1, r_2, r_3$  and  $r_4$  refer to the major four parameters in the SCA, with parameter  $r_1$  indicating the direction of movement inside or outside the area between the destination and resolution, as shown in Figure 1. Parameter  $r_2$  represents the random number in  $[0, 2\pi]$  as the outward movement represents positive sine and cosine of the investigation, while the ingoing movement represents negative sine and cosine. Parameter  $r_3$  denotes the weight to highlight ( $r_3 > 1$ ) or reduce the highlight  $r_3 < 1$  on a random impact of location on the definition of distance. As a random value in  $[0, 1]$ , parameter  $r_4$  identifies Equation (3), signaling the subsequent location. The calculation of the  $r_1$  value is based on Equation (3), in which the existing iteration is represented by  $t$ , while  $T$  refers to the highest iterations, and the constant is indicated by  $a$ , given as

$$r_1 = a - ta/T \tag{3}$$

To utilize the search space in the SCA, the search agent relocation can be made at another resolution. In the search space investigation, the resolution can also be found outside of the equivalent locations.

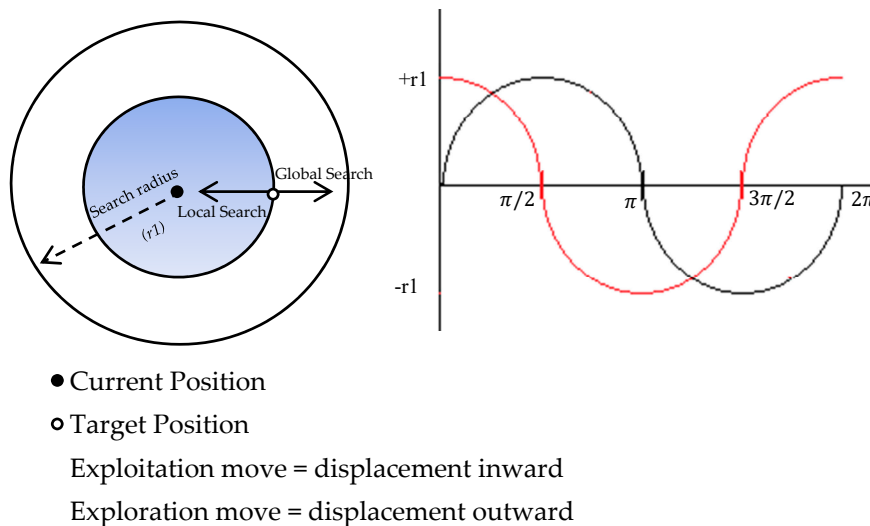


Figure 1. Effects of sine and cosine on search radius [17].

### 2.2. Original Cuckoo Search Algorithm (CSA)

The CSA is a metaheuristic algorithm based on nature, in which the position of a bird’s nest is randomly initialized according to the brood parasitism exhibited by several species of cuckoo in the region available in the mathematical function. Notably, CSA has exhibited high efficiency in various implementations by numerous studies [37,42].

The optimal values of a bird’s nest fitness are initially placed in the next generation of the process evolution. Next, the Levy flight mechanism is used to update the new location and position of the bird’s nest. Finally, the distribution of the random number  $R \in [0, 1]$  with the foreign egg,  $p_a$  if  $R > p_a$  are compared to obtain a new set of bird numbers and ensure that both the location and solution are optimal. In the CSA, the finding of the bird’s nest path is updated based on the “get best nest” and “get cuckoo” according to Equation (4) for each iteration [43], given as

$$x_i^{t+1} = x_i^t + a \oplus Levy(\lambda) \tag{4}$$

where  $x_i^t$  and  $x_i^{t+1} \lim_{x \rightarrow \infty} \sqrt{b^2 - 4ac}$  are the placement vectors of the bird’s nest in the development of  $t + 1$  and  $t$ , respectively. Apart from representing the step size of length adaptation,  $a$  is a higher constant than 0, which may incorporate different values in other circumstances. In general, the step size of the length adjustment factor is specified as  $\alpha = 0.01$ , while  $\oplus$  denotes a multiplication by point-to-point and  $Levy(\lambda)$  refers to a random search direction.



### 2.3. Latin Hypercube Sampling (LHS)

LHS is among the most remarkable sampling methods suggested by Stein et al. [44], which is also efficient for obtaining sample points. This method has significant strength, space-filling impact, and convergence features in comparison with other random or stratified sampling algorithms. In this study, the new sample size generated by the LHS method exhibits higher stability and wider implementation in the SCA adjustment. The LHS is presented by the  $n \times d$  matrix (e.g., a matrix with  $d$  columns and  $n$  rows). Every column,  $L$ , comprises a permutation of the integer 1 to  $n$ , while each row of  $L$  is represented by the (discrete) sample point and the expression of LHS is given as

$$L = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \left( \begin{matrix} x_{11} & \cdots & x_{1p} \end{matrix} \right) \\ \vdots \\ \left( \begin{matrix} x_{N1} & \cdots & x_1 \end{matrix} \right) \end{bmatrix} \quad (5)$$

where  $x_i$  refers to the  $j^{\text{th}}$  sample point, while samples  $x$  are organized through the random classification of the terms of  $x$ . generated vector elements. The procedures,  $x_{nm}$ , are presented in detail in [45], while the development of a large sample is based on diverse ideas by several researchers in [46–48].

## 3. Proposed Modified Sine Cosine Algorithm (MSCA)

### 3.1. The Theory of Sine Cosine Algorithm Adjustment

Here, the formation of the proposed MSCA algorithm, its implementation, and the working principles are explained in detail. For any optimization algorithm to achieve a global optimum, a proper modification should show a high convergence rate to the true global minimum even at a high number of dimensions [49]. The SCA has been tested using LHS as an efficient sampling method that is widely used in computer experiments. LHS sampling was proven in [45], in which this method improved various optimization algorithms efficiency.

The modification involves an additional information exchange to affect the performance of an algorithm, as in the case of LHS and SCA, which resulted in a stagnation effect and convergence complexity. The iterative modification concept is used to generate the set of random solutions. In [48], LHS was generally described as a spectrum of stratified sampling designs known as a partially stratified sample design. LHS represents the extremes of the partially stratified sample spectrum. The variance of partially stratified sample estimates is derived along with some asymptotic properties. The design of the partially stratified sample demonstrates its ability to reduce variance associated with variable interactions, whereas LHS reduces variance associated with the main effects. The SCA can be enhanced through the implementation of LHS in its optimizer. Firstly, the input of the SCA in an  $n$ -dimensional hypercube with lower and upper bounds of  $X$  is initialized for a random set solution. To obtain a set of random solutions for  $H$  sampling scale as an output, there are three steps involved, which are generated by partitioning the sampling matrix and sampling point from each selected hypercube. This output is then used as the replacement for a new set of SCA solutions.

Given enough computation, the SCA will always find the optimum, but a fast convergence cannot be guaranteed since the search relies entirely on random walks. Presented here for the first time, one modification to the method is made with the aim of increasing the convergence rate and the minimum optimal solution, thus ensuring the method becomes more practical for a wide range of applications without losing the attractive features of its original method.

### 3.2. Iterative Adjustment Using Latin Hypercube Sampling (LHS)

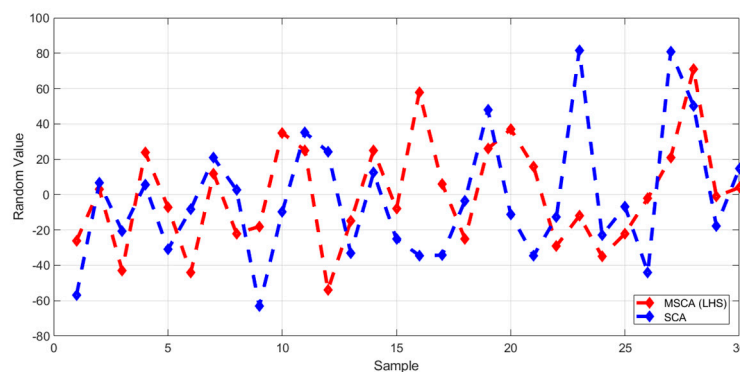
An initial population of 30 individuals was randomly chosen using LHS. This method was used in order to test the full design space with the minimum number of samples. As the random nature of LHS did not guarantee optimal space filling sampling, we iteratively generated 500 LHS and sampling with the maximum distance criteria between sample points was selected. LHS was actually intended

to improve the unnecessary and redundant features of the entire feature set. The value of the feature set (30 populations and 500 maximum iterations) was updated to control the optimization process based on the improvement made.

### 3.3. Adjustment of Sine Cosine Algorithm (SCA) Using Latin Hypercube Sampling (LHS)

The SCA is a recently developed optimization strategy. Researchers have developed an interest in solving optimization problems because of their flexibility and simplicity. LHS has proven its validity results with large deviations in the context of built estimators and is a well-known sampling technique for variance reduction, as discussed in [44–46]. This method has been extensively applied due to its simple implementation and beneficial features. Specifically, only one sample in LHS is chosen in every column or row for every sub-hypercube [45], with  $n$  (initial sample size) points and  $d$  random variables denoted as  $X_{n \times d} = [x_1^T, x_2^T, \dots, x_n^T]^T$ , where each row  $X_i = [x_{i1}, x_{i2}, \dots, x_{id}]$  represents a sample point as generated in [45]. To improve the SCA using LHS, it can be treated as space filling criteria to search the best combination of the sample matrix  $X_{n \times d}$ . This sample provides a maximum and minimum number of search agents (population),  $N$ .

Different sampling algorithms using LHS in the SCA are slightly better than the original sample in terms of efficiency. This is because the proposed algorithm was optimized each time in the sample points. As an extension algorithm of LHS, the performance of MSCA is slightly superior compared to traditional SCA sampling. Figure 2 shows the random sample value of the SCA with and without LHS, and this example had a sample size of  $n = 30$ . The use of random points in the hypercube intervals (random LHS) relied upon the selection of the initial design in LHS. LHS increases the efficiency of these sampling strategies through the development of new algorithms based on the combination of innovative space-filling criteria and specialized optimization schemes.



**Figure 2.** The random sample value of the Sine Cosine Algorithm (SCA) with and without Latin Hypercube Sampling (LHS).

Algorithm 1 illustrates the pseudocode of the MSCA algorithm. The identification of the MSCA process initially led to a new category of random resolutions. Following the random initialization is the measurement of a category of random resolutions through the fitness function, which is the aim of the placement of location. When the fitness function of the initial population is evaluated, the most ideal resolutions would be regarded as those to be applied to identify the next resolutions. The targeted resolution was reached by MSCA through some iterations (generations). The update of the cosine and sine functions was made when the iteration counter increased. When the optimal solution (destination) for the particular domain of issue was achieved and the termination requirements were fulfilled, the algorithm ended.

In this case, it was assumed that the set of SCA resolutions from LHS sampling in Equations (1) and (2) are the replacement of the new solution of  $X$ . Following that, the calculation was updated in a certain range ( $r$  value). The efficiency of LHS samples into the SCA resulted in the fastest convergence of MSCA, which reflects the decreasing number of iterations (Figure 3). The aim of the proposed

algorithm is to generate an additional sample matrix and optimize the matrix by exchanging the elements within the column. It is also demonstrated that the proposed algorithm exhibited good performance in efficiency and convergence compared to the traditional extension algorithms.

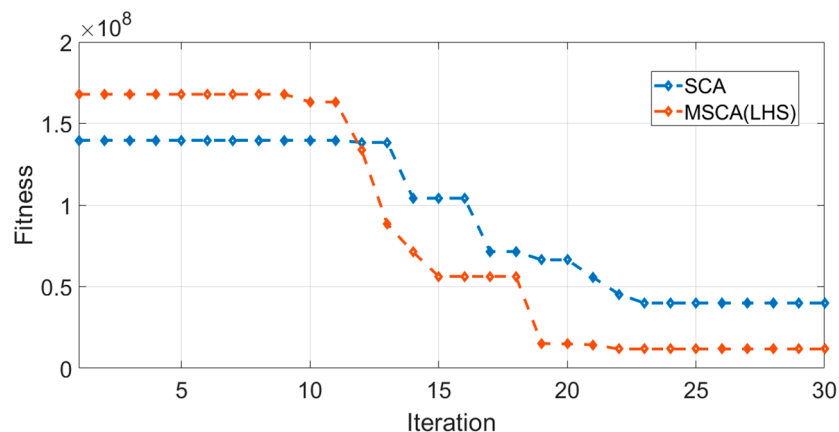
---

**Algorithm 1.** Pseudo-code of MSCA Algorithm.

---

01: Input: The population  $X = \{X_1, X_2, \dots, X_D\}$ , the constant magnitude  $a$   
 02: Output:  $X_{best}$  and the updated population  $X' = \{X'_1, X'_2, \dots, X'_N\}$   
 03: Generate a random set of population  $X$   
 04: Evaluate the fitness of each and every population  
 05: **while do**  
 06: Set initial  $r_1$  using Equation (2)  
 07: Update the best solution  
 08: **for** ( $t < \text{maximum number of generation}$ )  
 09: Perform Latin Hypercube Sampling  
 10: Update  $r_1, r_2, r_3$  and  $r_4$   
 11: Update the position of the search agents (population) using (1), (2) and (3)  
 12: Evaluate the fitness of each and every search agent  
 13: **If** fitness (current) < fitness (previous) then  
 14: Update the best solution  
 15: **Else if** fitness (current) = fitness (previous) and Number of features selected (current) < Number of features selected (previous) then  
 16: Update the best solution and its position,  $P_i^t = X_{best}$   
 17: **End**  
 18: Replace the least fit search agent with LHS and update the new population  
 19: **End**  
 20: **End**  
 21: Return the updated solution,  $X$  and the best result  $X_{best}$

---



**Figure 3.** Convergence curves of SCA and modified Sine Cosine Algorithm (MSCA).

#### 4. Proposed Hybrid Modified Sine Cosine Algorithm Cuckoo Search Algorithm (HMSCACSA)

In this section, we discuss in detail the scheme of the proposed hybridization of the adjusted SCA through LHS with CSA, its application, and its working principles.

##### 4.1. Concept of Hybridization

SCA uses the characteristics of sine and cosine trigonometric functions to update the solutions. However, similar to other population-based optimization algorithms, the SCA suffers from low diversity, stagnation in local optima and the skipping of true solutions [13]. Therefore, we attempted to eradicate these issues by proposing a hybrid version of the SCA. The proposed algorithm is named

the HMSCACSA. In this study, the CSA was integrated with MSCA to find the optimum convergence rate rapidly, thus making the method more practical for a wide range of applications without losing the attractive features of the original method.

#### 4.2. Iterative Hybridization of the Modified Sine Cosine Algorithm (MSCA) with the Cuckoo Search Algorithm (CSA)

The initial motivation to develop a hybrid metaheuristic approach was that obtaining an efficient solution for an optimization problem is a very challenging task which depends on the correct selection of optimization techniques. The optimal solution will satisfy six main requirements: easy implementation; balance of exploration and exploitation values; every iteration must be found in a true global optimum; fast convergence; minimum parameter tuned; and minimum computational power complexity.

With the HMSCACSA technique, we aimed to achieve the above six requirements. Similar to the SCA and the CSA, it is based on population algorithms and uses a population to pursue the global solution [50]. The initial solution is transferred for distribution through a solution space generated by the initial population. There is no change in the number of populations for the entire algorithm iteration. Towards reaching the optimum minima, the population is fully guided through the iteration process of reproduction (to obtain the best number of search agents with the lowest cost). The pseudocodes of the proposed HMSCACSA are presented in Algorithm 2. In this case, the new CSA is assumed from the LHS in the SCA, generated by the fitness minimum value in the current best solution that replaces the new population.

---

#### Algorithm 2. Pseudo-code of the HMSCACSA.

---

```

01: Input: The population  $X = \{X_1, X_2, \dots, X_D\}$ , the constant magnitude  $a$ 
02: Output:  $X_{best}$  and the updated population  $X' = \{X'_1, X'_2, \dots, X'_N\}$ 
03: Generate a random set of population  $X$ .
04: Evaluate the fitness of each and every population
05: while do
06: Set initial  $r_1$  using Equation (2)
07: Update the best solution
08: for ( $t <$  maximum number of generation)
09: Perform Latin Hypercube Sampling
10: Update  $r_1, r_2, r_3$  and  $r_4$ 
11: Update the position of the search agents (population) using (1), (2) and (3)
12: Evaluate the fitness of each and every search agent
13: If fitness (current)  $<$  fitness (previous) then
14: Update the best solution
15: Else if fitness (current) = fitness (previous) and Number of features selected (current)  $<$  Number of features selected (previous) then
16: Update the best solution and its position,  $P_i^t = X_{best}$ 
17: End
18: Get a cuckoo (4) by the minimum value of sampling LHS (5) position and fitness destination.
19: Keep the best solution (or nests with quality solutions), rank the solutions and find the current best
20: Replace the least fit search agents and update the new population
21: End
22: End
23: Return the updated solution,  $X$ . and the best result ( $X_{best}$ )

```

---

The MSCA begins the search by applying the standard sine cosine functions for the number of iterations. The best-obtained solution is then passed to the CSA to accelerate the search and overcome the slow convergence of the standard SCA. To determine the success of this HMSCACSA algorithm, three elements of the parameter are gradually examined in an attempt to change the value of five

samples to obtain the best iterative algorithm solution. Further enhancements can be made by gradually examining the HMSCACSA solution through the fine-tuning of internal parameters such as population size (number of search agents), dimension size, and random parameters chosen (range value) using a maximum of 1500 iterations. The flowchart of the proposed method is shown in Figure 4. In this algorithm, the advantages of both the SCA and the CSA are combined. The major limitations of the SCA are resolved through modifications using LHS and hybridization with the CSA.

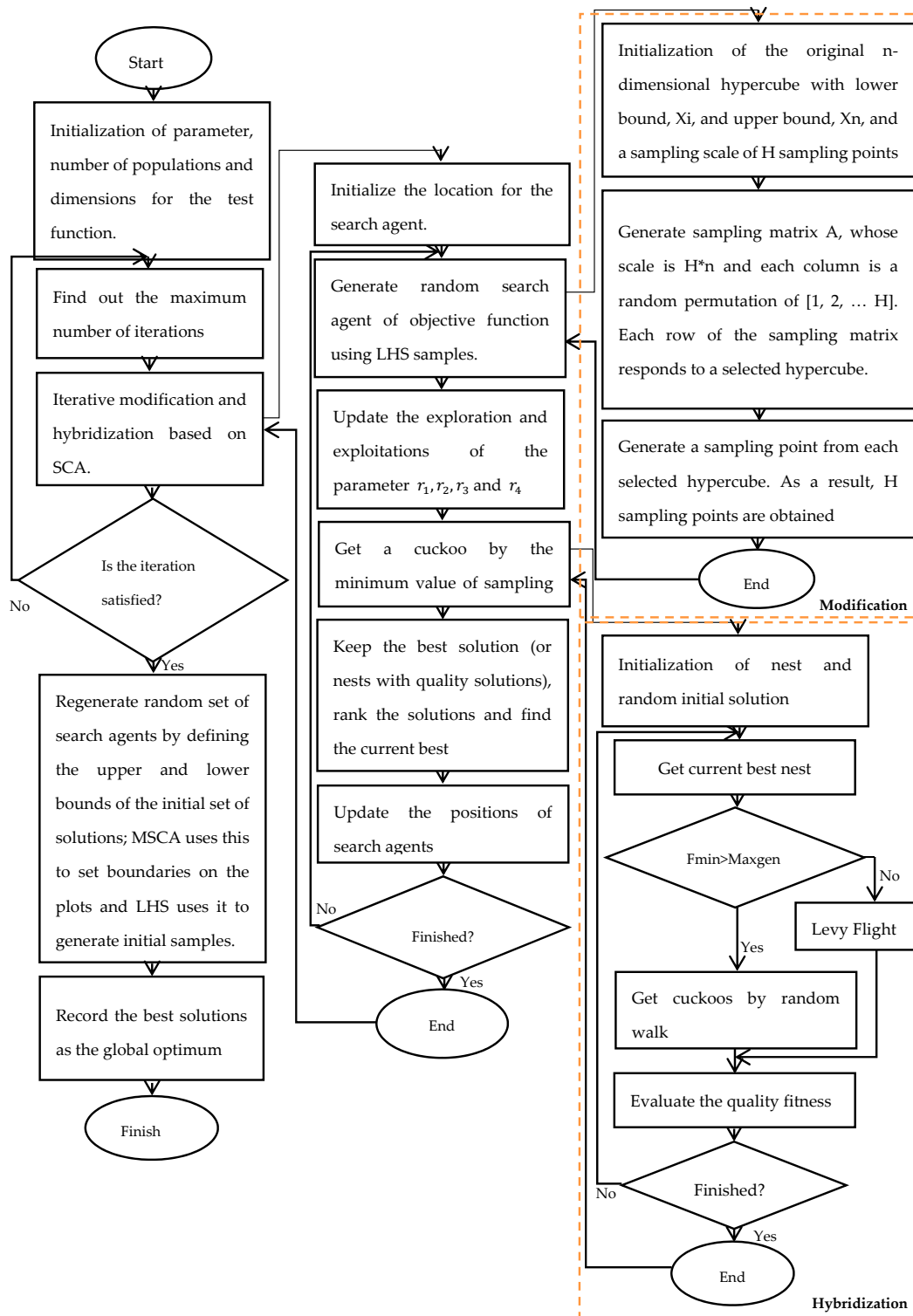


Figure 4. Flowchart of the hybridization and modification for the proposed method based on the SCA.

### 5. Development of the Execution of the Proposed Algorithm

To test the effectiveness of the proposed algorithm, analysis and tests should be carried out to ensure that real-life performance has been improved. Thus, a single objective function was used to test the performance of HMSCACSA. Table 2 illustrates the classical set of six benchmark mathematical functions [49] from 23 notable benchmark mathematical functions within IEEE CEC 2014 [28] for each category to describe the selection functions.

**Table 2.** Description of classical benchmark mathematical functions.

Function	Dim	Range
$F_1(x) = \sum_i^n x_i^2$	20	(-100,100)
$F_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	20	(-10,10)
$F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	20	(-100,100)
$F_4(x) = \max\{ x_i , 1 \leq i \leq n\}$	20	(-100,100)
$F_5(x) = \sum_{i=1}^{n-1} \left[ \begin{matrix} 100(x_{i+1} - x_i^2)^2 \\ + (x_i - 1)^2 \end{matrix} \right]$	20	(-30,30)
$F_6(x) = \sum_{i=1}^n ( x_i + 0.5 )^2$	20	(-100,100)

#### 5.1. Simulation Findings

To validate the performance of the proposed algorithm, an experimental setting was implemented. Six categories of benchmark mathematical functions were deployed to test the algorithm’s efficiency (Table 2). The variants were coded in MATLAB 2018a on a Core i5, 3.1 GHz system, and the configuration was performed using the same personal computer (PC) for all of the simulation experiments. To ensure a fair comparison between the metaheuristics, a comparison was also made between the hybrid-to-hybrid metaheuristics, namely, hybrid PSOABC, PSOAGWO and PSOAGSA.

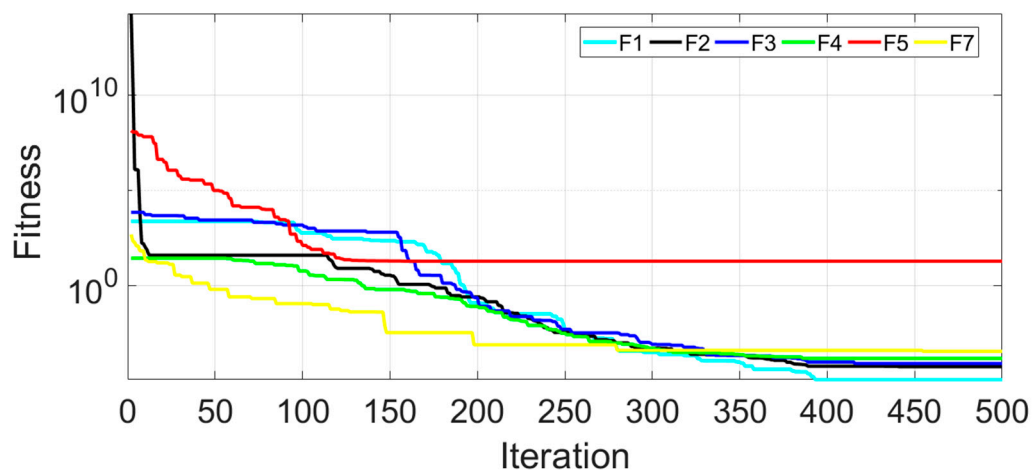
#### 5.2. Analysis of the Hybrid Modified Sine Cosine Algorithm Cuckoo Search Algorithm (HMSCACSA) Results

The analysis of the HMSCACSA is presented in this section. The general concept is the utilization of the CSA and SCA, followed by the development of the emerging hybrid variant to create a new transformation as an alternative to the negative result through a one-to-one idea. This innovation indicates an effective enhancement of the SCA to create a stable investigation and utilization of the overall iterative search procedure. Furthermore, it also leads to a prompt gathering among diverse individuals until they develop a similarity to the global optimal individual at a fast rate. When the best global individual is confined in the local optima, other individuals would be drawn to the local optima, leading to premature convergence of the SCA.

Through the LHS modification to the SCA, a set of random solutions from the function SCA is created by a random population using boundary numbers. Besides, the numbered boundary of all variables is based on the upper and lower number entered by the users to obtain the random population of X global optimum. The initialization function was replaced by the LHS procedure to develop a Latin hypercube sample of size N and a new value of X, which was equally administered on the unit square. During this process, the exploitation phases of SCA were gradually changed in the random solutions, while lower random variations were found compared to the variations in the investigation process. Through this process, the placement of solutions was updated to enable the search of space and to determine the optimal fitness values of the best index cuckoo nest using the solutions. These processes were reserved to update the ideal nest location and gain a new position.

In the IEEE CEC 2014 [28] benchmark set, the F1 to F3 test functions were unimodal, while the F4 to F16 test functions were multimodal. Figure 5 shows that the proposed HMSCACSA exhibited the fastest convergence in two out of six test functions, namely, the F5 and F7 test functions. In the majority of the evaluations, the convergence exhibited a higher performance through the proposed HMSCACSA in comparison with the adjusted and original SCA. This enhancement was led by the characteristics of

MSCA hybridization with CSA, as the SCA has internal memory to retain the potential solutions and converge on a global optimization [26].



**Figure 5.** Convergence curves of the proposed Hybrid Modified Sine Cosine Algorithm Cuckoo Search Algorithm (HMSCACSA) using six mathematical functions. The MSCA, HMSCACSA, Particle Swarm Optimization–Artificial Bee Colony (PSOABC), Particle Swarm Optimization–Artificial Bee Colony (PSOGWO), and Particle Swarm Optimization–Gravitational Search Algorithm (PSOGSA) are presented as blue, green, cyan, black, and magenta smoothed lines, respectively.

The MSCA comprises exploration capabilities to facilitate the access of optimum resolutions using LHS, while HMSCACSA consists of operators (sine, cosine, and Levy flight) to enhance the diversity of the population and its resistance to the local optima. Figure 6 presents the convergence curves obtained using the MSCA, HMSCACSA, PSOABC, PSOGWO and PSOGSA, which are presented in blue, green, cyan, black, and magenta smoothed lines, respectively. The performance graph approach is illustrated in a curvilinear way through an optimum focus point.

The HMSCACSA was further compared with three existing metaheuristic hybrid algorithms—PSOGSA, PSOGWO, and PSOABC—using the F5 test function with a population of 30 and 20 dimensions and 1500 iterations. In this case, five cycles of CPU time for 1500 iterations were used to avoid the immature convergence inaccuracy phase for several benchmark functions. Despite the challenging competition, the HMSCACSA had higher performance than the hybrid metaheuristics. Figure 6e shows that the proposed HMSCACSA achieved the fastest convergence by up to 63.89% and 60.83% after about 1200 iterations, with a minimization value up to 82.49% and 3.83% compared to other benchmarked hybrid metaheuristics and the MSCA for F5 (Table 3). These comparisons list two different percentages between HMSCACSA and another metaheuristics, that is, the iteration and minimization values, using Equation (6):

$$\text{Percentage\_Difference} = \frac{|\text{highest}(\text{another\_meta}) - \text{least}(\text{another\_meta})|}{\text{highest}(\text{another\_meta})} \times 100 \quad (6)$$

Based on the boxplots shown in Figure 7, several salient characteristics of the PSOGSA, PSOGWO, PSOABC, MSCA, and HMSCACSA search processes are presented. Taking the F5 function into consideration through this analysis, we attempted to highlight the overall performance of the HMSCACSA, which had a lower median compared to the other algorithms (Figure 7b). Although the MSCA exhibited a closed quartile bias range, it also comprised a larger interquartile range and a lower mean compared to the PSOGSA, PSOABC, and PSOGWO. However, the PSOGWO had a lower median than the other algorithms. It was also observed that all four algorithms exhibited similar results, except for PSOGWO, which had a bias towards the upper quartile. The statistical analysis indicated that the HMSCACSA exhibited the fastest convergence rate; this was primarily due to

the advantages of the modification and hybridization processes in the proposed HMSCACSA that improved the investigation and utilization in global searching. To validate the processing time of the proposed variant methods, all algorithms were verified using the CPU time [29] deployed to reach the convergence.

The mean and standard deviation were measured to analyze the performance of the proposed method for benchmarking of the mathematical function optimization. The results for both the mean and standard deviation are presented in Table 4, which also comprised 1500 iterations, a population of 30, and 20 dimensions. Furthermore, the mean values of the HMSCACSA optimization through the fourteen test functions indicated that the majority of the functions had low values (162.1748, 3,690,500,000, 666.0412, 2.6749, 14,274,000, and 224,550). Meanwhile, the standard deviations of the proposed HMSCACSA, which ranged from  $10^3$  to  $10^{10}$  for the different functions, were also low. These results indicated that the proposed HMSCACSA exhibited a higher performance than the other hybrid algorithms in the minimization for six test functions. Therefore, the proposed HMSCACSA enhanced the original SCA in terms of reaching optimal solutions and the ability to perform a local–global search.

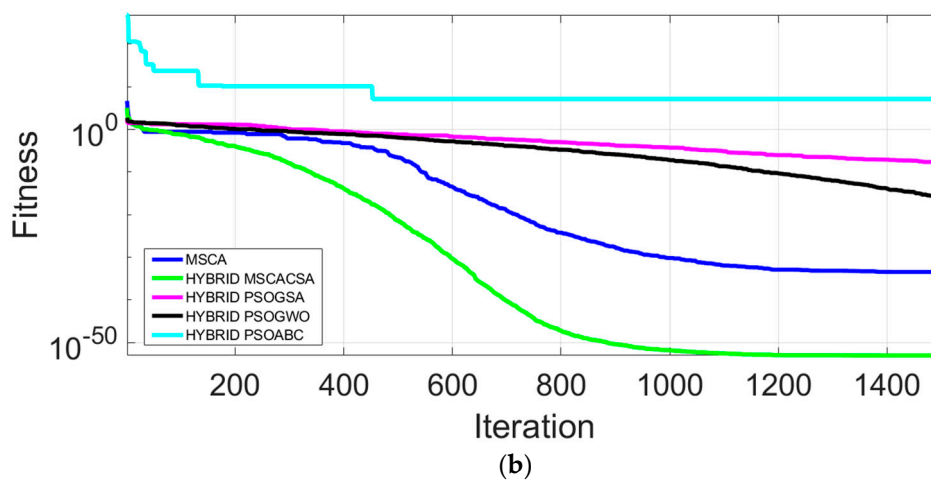
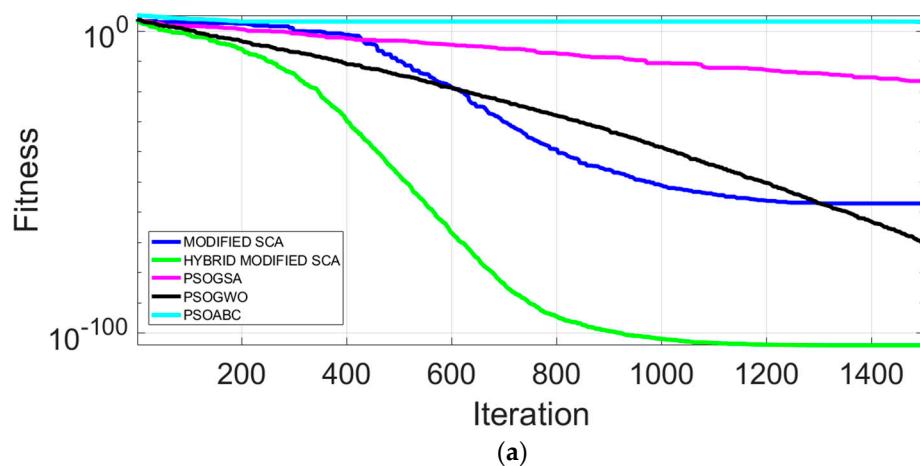


Figure 6. Cont.



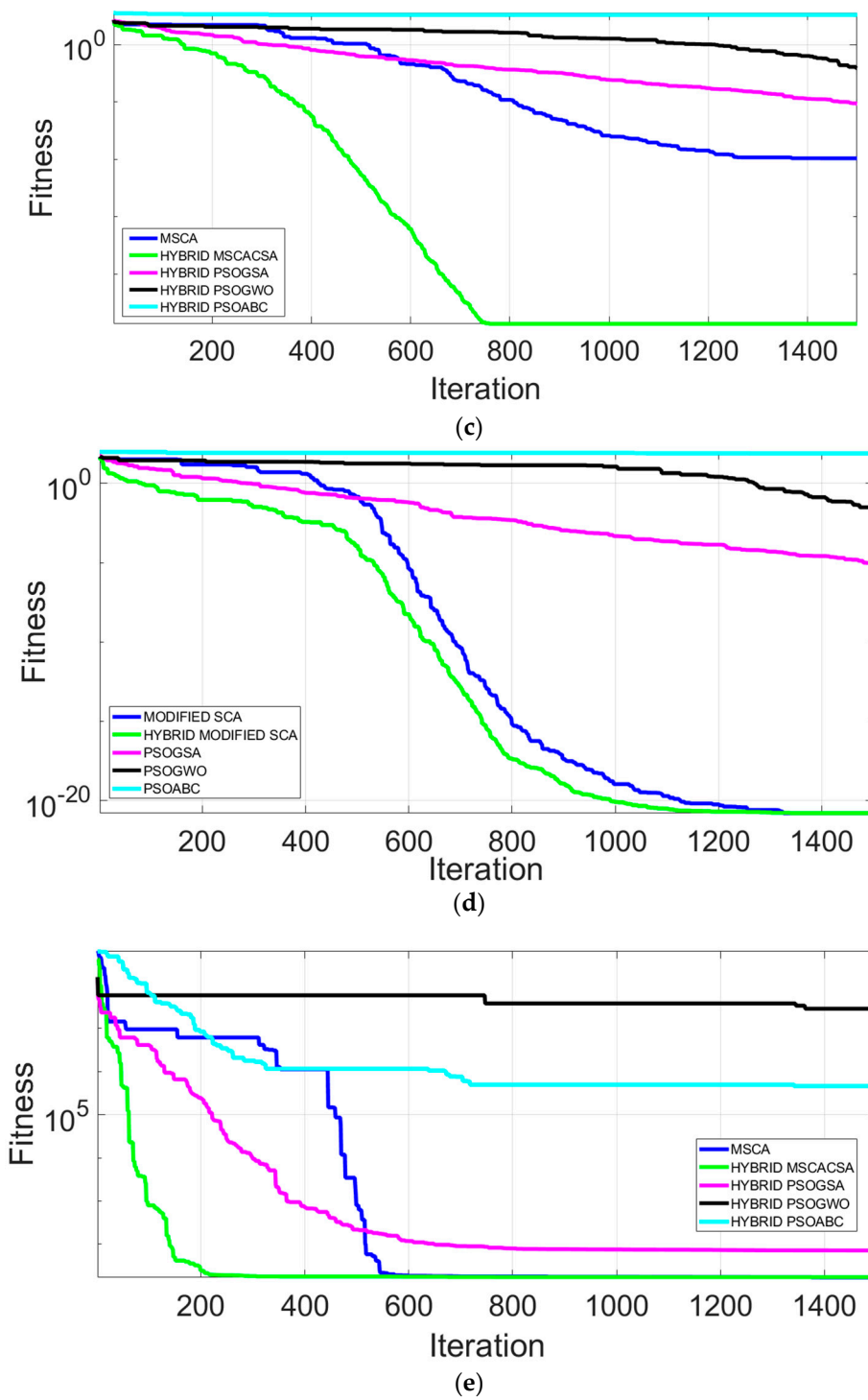
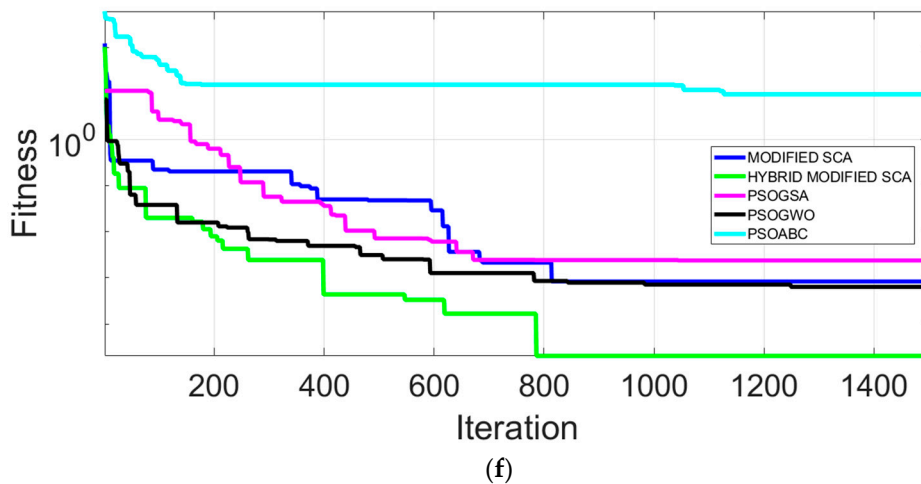


Figure 6. Cont.



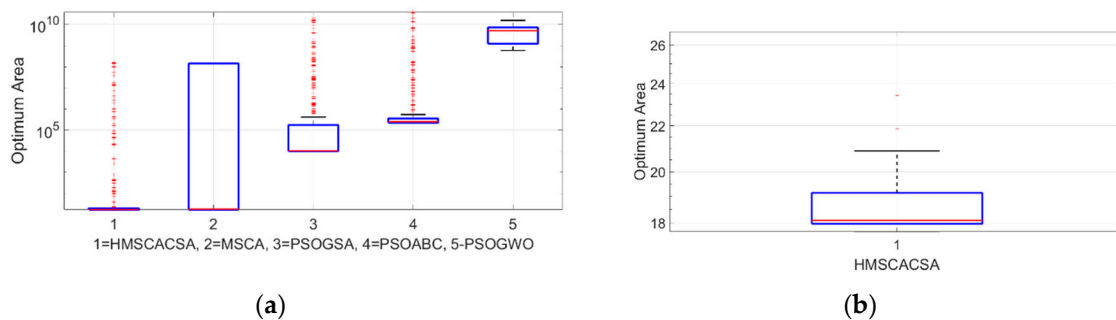
**Figure 6.** Convergence curves of mathematical functions for (a) F1, (b) F2, (c) F3, (d) F4, (e) F5 and (f) F7 using the MSCA, HMSCACSA, PSO GSA, PSO GWO, and PSO ABC.

**Table 3.** Percentage differences between our proposed method (HMSCACSA) and other metaheuristics (MSCA, PSO GSA, PSO ABC, and PSO GWO; results also presented in Figure 6).

Method	Iterations	Minimization Value	% Difference (Iteration)	% Difference (Minimization)
HMSCACSA	282	17.22	-	-
MSCA	720	16.58	60.83	3.829
PSO GSA	781	98.33	63.89	82.49
PSO ABC	1332	945,855.73	78.83	99.9982
PSO GWO	1358	1,247,463.24	79.23	99.9986

**Table 4.** Comparative statistical results for state-of-the-art algorithms using six benchmark functions for a population = 30, with 20 dimensions and a maximum of 1500 iterations.

Function	MSCA		HMSCACSA		PSO ABC		PSO GSA		PSO GWO	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
F1	$1.2243 \times 10^3$	$1.8078 \times 10^3$	162.1748	510.20913	509.4438	$3.1509 \times 10^3$	$1.1840 \times 10^3$	$4.1671 \times 10^3$	$3.5320 \times 10^4$	865.9147
F2	$1.0471 \times 10^{11}$	$4.0395 \times 10^{12}$	$3.6905 \times 10^9$	$6.1621 \times 10^{10}$	$3.2873 \times 10^{34}$	$1.2579 \times 10^{36}$	$2.5138 \times 10^{24}$	$5.2057 \times 10^{25}$	$1.2011 \times 10^{23}$	$4.3896 \times 10^{24}$
F3	$3.3982 \times 10^3$	$1.6561 \times 10^3$	666.0412	$1.4958 \times 10^3$	$2.0912 \times 10^4$	$1.0609 \times 10^4$	$7.5861 \times 10^3$	$4.7666 \times 10^3$	$2.2595 \times 10^3$	$5.8552 \times 10^3$
F4	18.5994	9.6382	2.6749	6.8591	33.9277	6.1326	74.2110	0.7134	63.4489	7.0077
F5	$4.8489 \times 10^7$	$7.0512 \times 10^7$	$1.4274 \times 10^7$	$3.7084 \times 10^7$	$2.7276 \times 10^8$	$2.2989 \times 10^9$	$2.7344 \times 10^8$	$1.8786 \times 10^9$	$7.5571 \times 10^8$	$1.5812 \times 10^9$
F7	$4.6469 \times 10^6$	$5.7564 \times 10^6$	$2.2455 \times 10^5$	$1.0826 \times 10^6$	$2.4465 \times 10^7$	$2.2364 \times 10^8$	$4.7658 \times 10^7$	$2.0160 \times 10^8$	$1.2024 \times 10^7$	$6.1536 \times 10^7$



**Figure 7.** Convergence boxplot for (a) solving function F5 with 1500 independent runs (between hybrid-to-hybrid metaheuristic) and (b) the HMSCACSA for function F5.

Table 5 shows that the HMSCACSA had the best optimal solutions in most cases using the F1, F2, F3, F4, and F5 standard functions, reducing the minimum processing time by 23.5%, 31.32%, 29.1%, 29.0%, and 43.60%, respectively. Meanwhile, the PSOAGSCA reached convergence faster than the MSCA by 27.69%, 29.47%, 26.44%, 29.49%, and 32.2% for functions F1, F2, F3, F4, and F5, respectively. Overall, it was indicated from the simulation outcomes that the proposed HMSCACSA had faster convergence than the MSCA.

**Table 5.** Central Processing Unit (CPU) time for tested algorithms using standard benchmark functions.

Function	PSOABC	PSOGWO	PSOGSA	MSCA	HMSCACSA
F1	0.0001039	0.0001096	0.0000915	0.0000968	0.0000700
F2	0.0001039	0.0001197	0.0000993	0.0000967	0.0000682
F3	0.0000939	0.0001126	0.0000938	0.0000904	0.0000665
F4	0.0000974	0.0001140	0.0000947	0.0000953	0.0000672
F5	0.0001004	0.0001197	0.0000922	0.0000767	0.0000520

The selection of specific internal parameters should be emphasized due to the significant effect of the optimization algorithm’s performance. Notably, the SCA exhibits sensitive parameter values, which could be fine-tuned to improve the capability of exploring optimal solutions in the global search domain. Three main parameters can be considered for tuning to obtain significant performance enhancements: (1) the reasonable value for a search agent (population) should be considered; (2) the convergence curve and problem dimensions must be taken into account in the selection of population values; and (3) the computational time might be compromised by high population values, which might also be redundant in the search process due to the increase in the number of iterations. Finally, four small fraction parameters ( $r_1, r_2, r_3$  and  $r_4$ ) in the exploration and exploitation SCA algorithm were used to determine the placements of a new resolution, which may be towards or outwards from the destination points.

To successfully solve the optimization using the HMSCACSA technique, five specific samples exhibited promising values to develop a strong potential selection. However, the adjustment of the selection parameters was a challenging task. Furthermore, constant values (nearer to the basic code) would display better results in the optimization of benchmark issues, which had a higher possibility of affecting the identification of a resolution, including the population (number of search agents) and dimension sizes, and the randomly selected parameter (value of investigation or utilization range). The details of the design specifications are presented in Table 6, along with the comparative results simulated by the F5 benchmark test function.

**Table 6.** Parametric specification of population, dimension and range value for five samples with varied population sizes.

Parameter	Value
Population	20, 30, 40, 50, 60
Dimension	20
Range	2
Maximum Iterations	500
Dimension	20, 30, 40, 50, 60
Population	60
Range	2
Maximum Iterations	500
Range	1.4, 1.8, 2, 2.2, 2.4
Dimension	20
Population	60
Maximum Iterations	500
Dimension	20

Figure 8 displays the convergence curves using different numbers of search agents (population), which were specified as 20, 30, 40, 50, and 60, with a dimension value of 20. It was found that the increase in population from 20 to 60 led to a higher convergence rate by a maximum of 7.4%. Based on the tuning of the experimental dimensional value, it was found that the low dimension value had the lowest optimal solutions compared to other values shown in Figure 9. The dimension values of the HMSCACSA were set at 20, 30, 40, 50, and 60, with the number of search agents set to 30. As a result, the convergence rate was reduced by a maximum of 37.86%. Four variables were used in the SCA for the tuning of  $r$  variables. In this case,  $r_1$  determined the action to be performed by the search agent between investigation and utilization. Although all stochastic algorithms consist of investigation and utilization, the balance of these algorithms is important.  $r_2$  determined the distance of the movement of the resolution, while  $r_3$  distributed a random weight, and  $r_4$  determined the formula to be applied between sine or cosine [51]. The resolutions in every iteration were assessed by the fitness function, while the algorithm distributed the ideal resolution acquired at the point of location. This was followed by the upgrade of the  $r$  variables. Moreover, the sine cosine values were set from 1.4, 1.8, 2.0, 2.2, to 2.4 for a diversity of findings. The values significantly affected parameters  $r_1, r_2, r_3,$  and  $r_4$ . Figure 10 illustrates the diversity plot of the HMSCACSA to indicate the difference between investigation and utilization  $r$  in the converged iterations. Notably, the fixed value of range 1.4 had higher stability by up to 68.75% compared to the other ranges in terms of finding the global optimal solution optimization.

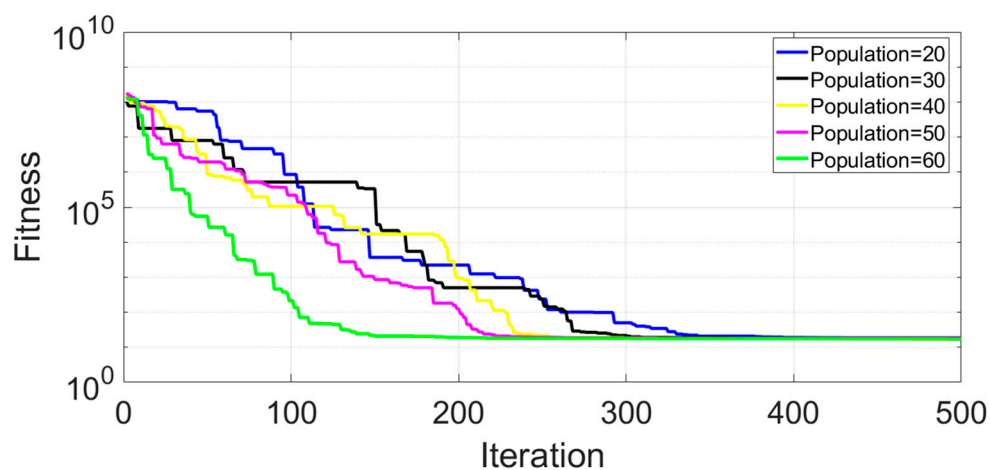


Figure 8. Comparison of parametric results with different numbers of search agents in the HMSCACSA.

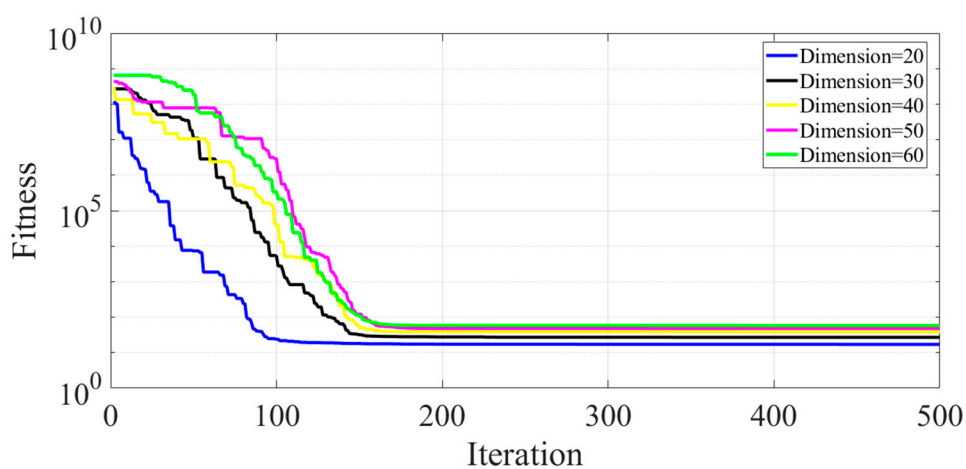
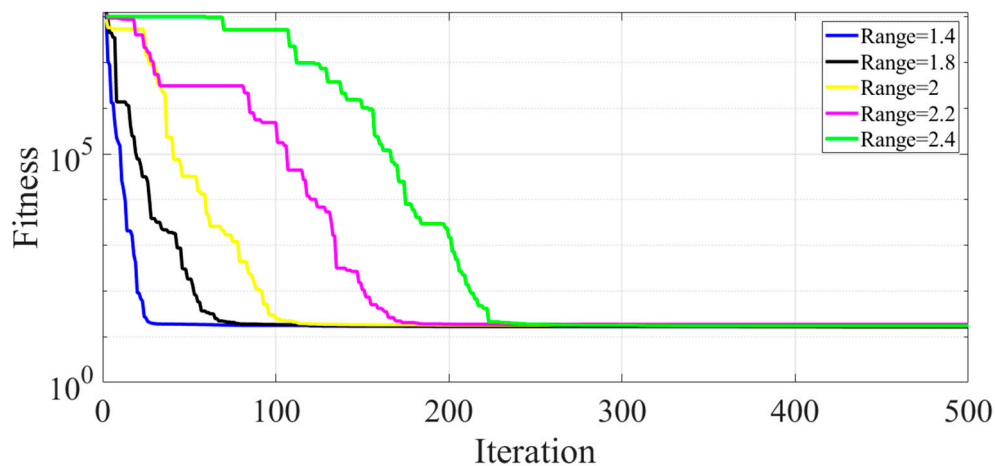


Figure 9. Comparison of parametric results with different numbers of dimensions in the HMSCACSA.



**Figure 10.** Diversity plot of the HMSCACSA for variants in the range of  $r$  exploration and exploitation.

Based on the three aforementioned hypotheses, it was observed that the HMSCACSA successfully achieved the fastest convergence, shortest CPU duration, and lowest mean and standard deviation values of fitness. As shown in Figures 6 and 7, a performance comparison was made between the HMSCACSA and other hybrid metaheuristic algorithms. Table 5 presents the 1500 mean CPU execution time results (in seconds) for the HMSCACSA using five benchmark test functions and with a population size of 60, 20 dimensions, and a range of 1.4. The hybrid variant was found to resolve the majority of the standard functions (F5) within the minimum duration and 29.5% faster when compared to function F1. All simulation findings indicate that the proposed HMSCA improved the CSA's effectiveness in terms of result quality and computational attempts.

## 6. Conclusions

In this study, the LHS method was applied to develop a MSCA. Moreover, to enhance the MSCA, a hybridization between the MSCA and traditional CSA was introduced. The proposed HMSCACSA—comprised of the sine function, cosine function, and Levy flight—was validated using six chosen IEEE CEC 2014 [28] and IEE CEC 2017 [27] benchmark test functions. It was found that the overall performance of the HMSCASCA outperformed other algorithms (hybrid PSOABC, PSOGWO, and PSOGSA) by up to 63.89% in terms of achieving better optimal solutions and ability to perform a global search. Additionally, the means and standard deviation values of fitness were low, ranging from  $10^3$  to  $10^{10}$  for the HMSCASCA. Moreover, the HMSCASCA also had a lower median fitness value compared to the other algorithms.

The proposed HMSCACSA also demonstrated higher stability and reduced computational time when the population size was high and the number of dimensions was low, respectively. The proposed HMSCACSA consumed the minimum CPU time by up to 43.6% compared with the other benchmarked hybrid metaheuristics. However, a limitation occurred whenever the proposed HMSCACSA was compared with its counterparts in minimizing the F5 test function. In this case, the HMSCACSA was outperformed by the MSCA in the F5 fitness minimization by 3.49%. In the future, a study could be done to apply the proposed HMSCACSA in solving Low Autocorrelation Binary Sequences (LABS) for radar communication systems [52–54], with the aim of obtaining optimized high Energy Levels (E), low peak Sidelobe Levels (SL), and a high merit factor (MF). The proposed algorithm can also be applied as a detection algorithm in Massive Multi-Input Multi-Output (MIMO) by searching for the optimum solution vector in the modulation alphabet with linear detection. Such an optimization will result in the minimization of the Bit Error Rate (BER) for large-scale antenna [55]. In addition, the proposed optimization algorithm could also be run in a more complex antenna array synthesis to optimize locations, excitation amplitudes, and the excitation phase of array elements, achieving a

high antenna directivity, small half-power beamwidth, low average side lobe level suppression, and predefined nulls mitigation [56].

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

**Funding:** This research was funded by a Universiti Sains Malaysia RUI grant, grant number 1001/PJJAUH/8011058 and The APC was funded by Universiti Sains Malaysia RUI grant, grant number 1001/PJJAUH/8011058 and UniMAP.

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

## Abbreviations

The following abbreviations are used in this manuscript:

ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
ALO	Ant Lion Optimizer
ASMO	Ageist Spider Monkey Optimization
BER	Bit Error Rate
CEC	Congress on Evolutionary Computation
CPU	Central Processing Unit
CSA	Cuckoo Search Algorithm
DE	Differential Evolution
DESCA	Differential Evolution Sine Cosine Algorithm
E	Energy Level
EA	Evolutionary Algorithms
EBS	Enhanced Brain Storm
EBSSCA	Enhanced Brain Storm–Sine Cosine Algorithm
GA	Genetic Algorithm
GSA	Gravitational Search Algorithm
GWO	Grey Wolf Optimizer
HAGWO	Hybrid Approach Grey Wolf Optimizer
HMSCACSA	Hybrid Modified Sine Cosine Algorithm Cuckoo Search Algorithm
ISCA	Improved Sine Cosine Algorithm
LABS	Lows Autocorrelation Binary Sequence
LHS	Latin Hypercube Sampling
MF	Merit Factor
MGWO	Mean Grey Wolf Optimizer
MIMO	Massive Multi-Input Multi-Output
MOSCA	Multi-Orthogonal Sine Cosine Algorithm
MSCA	Modified Sine Cosine Algorithm
OBL	Opposition Based Learning
PBO	Polar Bear Optimization
PC	Personal Computer
PSO	Particle Swarm Optimization
PSOSCALF	Hybrid Particle Swarm Optimization–Sine Cosine Algorithm and Levy flight
RADAR	Radio Detection and Ranging
SCA	Sine Cosine Algorithm
SI	Swarm Intelligence
SL	Sidelobe Levels
SS	Squirrel Search

TLBO	Teaching–Learning–Based Optimization
WOA	Whale Optimization Algorithm

## References

1. Yang, X.-S. Metaheuristic optimization: Algorithm analysis and open. In *Experimental Algorithms. SEA 2011. Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6630, pp. 21–32.
2. Yang, X.-S. *Engineering Optimization: An Introduction with Metaheuristic Applications*, 1st ed.; Wiley A John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2010.
3. Coello, C.A.C.; Lamont, G.B.; Van Veldhuizen, D.A. *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed.; Springer: New York, NY, USA, 2007.
4. Sivanandam, S.N.; Deepa, S.N. Genetic algorithm optimization problems. In *Introduction to Genetic Algorithms*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 165–209.
5. Ping, G.; Chunbo, X.; Yi, C.; Jing, L. Adaptive ant colony optimization algorithm. In Proceedings of the 2014 International Conference on Mechatronics and Control (ICMC), Jinzhou, China, 3–5 July 2014; pp. 95–98.
6. Das, S.; Mullick, S.S.; Suganthan, P.N. Recent advances in differential evolution—an updated survey. *Swarm Evol. Comput.* **2016**, *27*, 1–30. [[CrossRef](#)]
7. Gao, W.; Liu, S. Improved artificial bee colony algorithm for global optimization. *Inf. Process. Lett.* **2011**, *111*, 871–882. [[CrossRef](#)]
8. Basu, B.; Mahanti, G.K. A comparative study of modified particle swarm optimization, differential evolution and artificial bees colony optimization in synthesis of circular array. In Proceedings of the 2010 International Conference on Power, Control and Embedded Systems, Allahabad, India, 29 November–1 December 2010.
9. Sharma, A.; Sharma, A.; Panigrahi, B.K.; Kiran, D.; Kumar, R. Ageist spider monkey optimization algorithm. *Swarm Evol. Comput.* **2016**, *28*, 58–77. [[CrossRef](#)]
10. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
11. Jain, M.; Singh, V.; Rani, A. A novel nature-inspired algorithm for optimization: Squirrel search algorithm. *Swarm Evol. Comput.* **2019**, *44*, 148–175. [[CrossRef](#)]
12. Połap, D.; Woźniak, M. Polar bear optimization algorithm: Meta-heuristic with fast population movement and dynamic birth and death mechanism. *Symmetry* **2017**, *9*, 203. [[CrossRef](#)]
13. Mirjalili, S. SCA: A sine cosine algorithm for solving optimization problems. *Knowl. Based Syst.* **2016**, *96*, 120–133. [[CrossRef](#)]
14. Majeed, M.A.M.; Rao, P.S. Optimization of CMOS analog circuits using sine cosine algorithm. In Proceedings of the 8th International Conference on Computing, Communications and Networking Technologies (ICCCNT), Delhi, India, 3–5 July 2017.
15. Li, C. An enhanced brain storm sine cosine algorithm for global optimization problems. *IEEE Access* **2019**, *7*, 28211–28229. [[CrossRef](#)]
16. Belazzoug, M.; Touahria, M.; Nouioua, F.; Brahimi, M. An improved sine cosine algorithm to select features for text categorization. *J. King Saud Univ. Comput. Inf. Sci.* **2019**, *32*, 454–464. [[CrossRef](#)]
17. Zamli, K.Z.; Din, F.; Ahmed, B.S.; Bures, M. A hybrid Q-learning sine-cosine-based strategy for addressing the combinatorial test suite minimization problem. *PLoS ONE* **2018**, *13*, 1–29. [[CrossRef](#)]
18. Rizk-Allah, R.M. Hybridizing sine cosine algorithm with multi-orthogonal search strategy for engineering design problems. *J. Comput. Des. Eng.* **2018**, *5*, 249–273. [[CrossRef](#)]
19. Suid, M.Z.T.; Ahmad, M.H.; Ismail, M.R.T.R.; Ghazali, M.R.; Irawan, A. An improved sine cosine algorithm for solving optimization problems. In Proceedings of the 2018 IEEE Conference on Systems, Process and Control (ICSPC), Melaka, Malaysia, 14–15 December 2018.
20. Elaziz, M.A.; Oliva, D.; Xiong, S. An improved opposition-based sine cosine algorithm for global optimization. *Expert Syst. Appl.* **2017**, *90*, 484–500. [[CrossRef](#)]
21. Sindhu, R.; Ngadiran, R.; Yacob, Y.M.; Zahri, N.A.H.; Hariharan, M. Sine–cosine algorithm for feature selection with elitism strategy and new updating mechanism. *Neural Comput. Appl.* **2017**, *28*, 2947–2958. [[CrossRef](#)]
22. 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](#)]

23. Issa, M. ASCA-PSO: Adaptive sine cosine optimization algorithm integrated with particle swarm for pairwise local sequence alignment. *Expert Syst. Appl.* **2018**, *99*, 56–70. [[CrossRef](#)]
24. Qu, C.; Zeng, Z.; Dai, J.; Yi, Z.; He, W. A modified sine-cosine algorithm based on neighborhood search and greedy levy mutation. *Comput. Intell. Neurosci.* **2018**, *2018*, 1–19. [[CrossRef](#)] [[PubMed](#)]
25. 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](#)]
26. Nenavath, H.; Jatoth, D.R.K.; Das, D.S. A synergy of the sine-cosine algorithm and particle swarm optimizer for improved global optimization and object tracking. *Swarm Evol. Comput.* **2018**, *43*, 1–30. [[CrossRef](#)]
27. Wu, G.; Mallipeddi, R.; Suganthan, P.N. *Problem Definitions and Evaluation Criteria for the CEC 2017 Competition and Special Session on Constrained Single Objective Real-Parameter Optimization*; Technical Report; Nanyang Technological University: Singapore, 2017.
28. Liang, J.J.; Qu, B.Y.; Suganthan, P.N. *Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization*; Technical Report; Nanyang Technological University: Singapore, 2013.
29. Singh, N.; Singh, S.B. A novel hybrid GWO-SCA approach for optimization problems. *Eng. Sci. Technol. Int. J.* **2017**, *20*, 1586–1601. [[CrossRef](#)]
30. Nenavath, H.; Jatoth, R.K. Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking. *Appl. Soft Comput.* **2018**, *62*, 1019–1043. [[CrossRef](#)]
31. Chi, R.; Su, Y.X.; Zhang, D.H.; Chi, X.X. Adaptive cuckoo search algorithm for continuous function optimization problems. In Proceedings of the World Congress on Intelligent Control and Automation (WCICA), Guilin, China, 12–15 June 2016.
32. Fister, I.; Yang, X.S.; Fister, D.; Fister, I. Cuckoo search: A brief literature review. In *Cuckoo Search and Firefly Algorithm*; Yang, X.-S., Ed.; Springer: Cham, Switzerland, 2014; Volume 516, pp. 49–62.
33. Yang, X.; Deb, S. Engineering optimisation by cuckoo search. *Int. J. Math. Model. Numer. Optim.* **2010**, *1*, 330–343. [[CrossRef](#)]
34. Umenai, Y. A modified cuckoo search algorithm for dynamic optimization problems. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2016), Vancouver, BC, Canada, 24–29 July 2016.
35. Walton, S.; Hassan, O.; Morgan, K.; Brown, M.R. Modified cuckoo search: A new gradient free optimisation algorithm. *Chaos Solitons Fractals* **2011**, *44*, 710–718. [[CrossRef](#)]
36. Jiang, Y.; Liu, X.; Yan, G.; Xiao, J. Modified binary cuckoo search for feature selection: A hybrid filter-wrapper approach. In Proceedings of the 2017 13th International Conference on Computational Intelligence and Security (CIS), Hong Kong, China, 15–18 December 2017.
37. Abdul Rani, K.N.; Ali, A. Modified cuckoo search algorithm in weighted sum optimization for linear antenna array synthesis. In Proceedings of the IEEE Symposium on Wireless Technology and Applications (ISWTA), Bandung, Indonesia, 23–26 September 2012.
38. Wu, Q.; Zhang, C.; Zhang, M.; Yang, F.; Gao, L. A modified comprehensive learning particle swarm optimizer and its application in cylindricity error evaluation problem. *Math. Biosci. Eng.* **2019**, *16*, 1190–1209. [[CrossRef](#)]
39. Senel, F.A.; Gokce, F.; Yuksel, A.S.; Yigit, T. A novel hybrid PSO-GWO algorithm for optimization problems. *Eng. Comput.* **2019**, *35*, 1359–1373. [[CrossRef](#)]
40. Chun-Feng, W.; Kui, L.; Pei-Ping, S. Hybrid artificial bee colony algorithm and particle swarm search for global optimization. *Math. Probl. Eng.* **2014**, *2014*. [[CrossRef](#)]
41. Mirjalili, S.; Hashim, S.Z.M. A new hybrid PSOGSA Algorithm for function optimization. In Proceedings of the 2010 International Conference on Computer and Information Application, Tianjin, China, 3–5 December 2010.
42. Shehab, M.; Khader, A.T.; Al-Betar, M.A. A survey on applications and variants of the cuckoo search algorithm. *Appl. Soft Comput.* **2017**, *61*, 1041–1059. [[CrossRef](#)]
43. Duan, Y. A hybrid optimization algorithm based on bat and cuckoo search. *Adv. Mater. Res. Vols.* **2014**, *930*, 2889–2892. [[CrossRef](#)]
44. Stein, M. Large sample properties of simulations using latin hypercube sampling. *Technometrics* **1987**, *29*, 143–151. [[CrossRef](#)]
45. Li, W.; Lu, L.; Xie, X.; Yang, M. A novel extension algorithm for optimized latin hypercube sampling. *J. Stat. Comput.* **2017**, *87*, 2549–2559. [[CrossRef](#)]



46. Rajabi, M.M.; Ataie-Ashtiani, B.; Janssen, H. Efficiency enhancement of optimized latin hypercube sampling strategies: Application to Monte Carlo uncertainty analysis and meta-modeling. *Adv. Water Resour.* **2015**, *76*, 127–139. [[CrossRef](#)]
47. Liu, Z.; Li, W.; Yang, M. Two general extension algorithms of latin hypercube sampling. *Math. Probl. Eng.* **2015**. [[CrossRef](#)]
48. Shields, M.D.; Zhang, J. The generalization of latin hypercube sampling. *Reliab. Eng. Syst. Saf.* **2016**, *148*, 96–108. [[CrossRef](#)]
49. Laskar, N.M.; Guha, K.; Chatterjee, I.; Chanda, S.; Baishnab, K.L.; Paul, P.K. HWPSO: A new hybrid whale-particle swarm optimization algorithm and its application in electronic design optimization problems. *Appl Intell.* **2019**, *49*, 265–291. [[CrossRef](#)]
50. Kanagaraj, G.; Ponnambalam, S.G.; Jawahar, N. A hybrid cuckoo search and genetic algorithm for reliability—Redundancy allocation problems. *Comput. Ind. Eng.* **2013**, *66*, 1115–1124. [[CrossRef](#)]
51. Ekiz, S.; Erdoğan, P.; Özgür, B. Solving constrained optimization problems with sine-cosine algorithm. *Period. Eng. Nat. Sci.* **2017**, *5*, 378–386. [[CrossRef](#)]
52. Brest, J.; Boskovic, B. A heuristic algorithm for a low autocorrelation binary sequence problem with odd length and high merit factor. *IEEE Access* **2018**, *6*, 4127–4134. [[CrossRef](#)]
53. Rosli, S.J.; Rahim, H.A.; Abdul Rani, K.N. Design of amplitude and phase modulated pulse trains with good autocorrelation properties for radar communications. *Indones. J. Electr. Eng. Comput. Sci.* **2019**, *13*, 990–998. [[CrossRef](#)]
54. Rosli, S.J. Design of binary coded pulse trains with good autocorrelation properties for radar communications. *MATEC Web Conf.* **2018**, *150*. [[CrossRef](#)]
55. Li, L.; Meng, W.; Ju, S. A novel artificial bee colony detection algorithm for massive mimo system. *Wirel. Commun. Mob. Comput.* **2016**, *16*, 3139–3152. [[CrossRef](#)]
56. Abdul Rani, K.N.; Abdulmalek, M.; Rahim, H.A.; Chin, N.S.; Abd Wahab, A. Hybridization of strength pareto multiobjective optimization with modified cuckoo search algorithm for rectangular array. *Sci. Rep.* **2017**, *7*, 1–19. [[CrossRef](#)]

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



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).