Communications in Mathematics and Applications
Vol. 15, No. 1, pp. 43–72, 2024
ISSN 0975-8607 (online); 0976-5905 (print)
Published by RGN Publications
DOI: 10.26713/cma.v15i1.2466

http://www.rgnpublications.com

Research Article

Niching Sparrow Search Algorithm for Solving Benchmark Problems, Speed Reducer Design, and Himmelblau's Nonlinear Optimization Problem

Gagandeep Kaur Sidhu* and Jatinder Kaur

Department of Mathematics, Chandigarh University, Gharuan, Chandigarh, India
***Corresponding author:** gagan.kw777@gmail.com

Received: October 14, 2023 **Accepted:** January 16, 2024

Abstract. Metaheuristic algorithms are invented or modified in order to solve complex optimization problems at the global level. With the development of technology, almost every domain such as engineering, industrial, medical etcetera is facing the problem of optimization. In order to solve these problems, a number of algorithms have been discovered. One of the most recent optimization algorithms is *Sparrow Search Algorithm* (SSA) which is famous for its good optimal ability along with fast convergence, Although, the SSA has a lot of merits, it is still facing numerous drawbacks namely falling into the local optima, steady convergence, etc. Therefore, we have proposed *Niching SSA* (NSSA) by introducing the Niching technique in SSA for updating the position of followers and scouters. This NSSA has been tested on 18 benchmark functions, speed reducer design, and also on Himmelblau's nonlinear optimization problem. In this work, we have examined NSSA from various aspects like optimal value, average mean for convergence accuracy, and the standard deviation for stability, and also have drawn the convergence curves through Matlab to check the convergence rate. Moreover, we have applied the Wilcoxon Signed rank test on NSSA. In all these aspects, computational results reveal that the performance of NSSA is superior with respect to SSA, GWO, PSO, and GSA.

Keywords. Sparrow Search Algorithm (SSA), Niching Sparrow Search Algorithm (NSSA), Niching technique, Optimization problems, Metaheuristic algorithms

Mathematics Subject Classification (2020). 90C47, 90C90, 90B25

1. Introduction

Optimization is gaining the best in the company of accessible resources, although fascinating the constraints. There is no limitation to optimization problems. If we take our sight on all over the world then the maximum number of things need to be optimized. We also optimize our day-to-day life such as we need the best job with more salary, the best house, etc. Similarly, in every field such as doctoral, engineering, agriculture, stock marketing, data science, scientists, mathematics, etc., we can see optimization problems. Mathematically there are various types of optimization problems that depend upon the number of objective functions a problem have, types of the objective function as well constraints, decision variable's types, etc. We can understand the types of optimization problems properly from Figure 1.

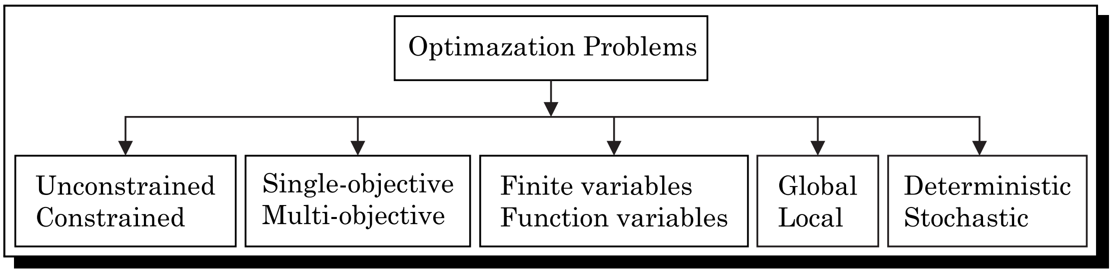


Figure 1. Types of optimization problems

In this paper, we are dealing with single-objective optimization problems. Consequently, the standard non-linear optimization problem (Venter [29]) presented below in equation (1.1):

Minimize: $f(\boldsymbol{x})$

$$\text{Subject to: } \begin{cases} h_i(\boldsymbol{x}) \le 0 & i = 1,\dots,n \\ g_j(\boldsymbol{x}) = 0 & j = 1,\dots,m \\ x_k^L \le x_k \le x_k^U & k = 1,\dots,p \end{cases} \tag{1.1}$$

In equation (1.1), $f(\boldsymbol{x})$ displays the objective function which is going to be minimized but sometimes it is going to be maximized, $h_i(\boldsymbol{x})$ are known as inequality constraints, $g_j(\boldsymbol{x})$ represents equality constraints whereas the vector $\boldsymbol{x} = (x_1, x_2, \dots, x_p)$ depicts the p design variables. x_k^L and x_k^U are the lower and upper bounds of the design variables. After the discussion on optimization problems, we can say there are various optimization problems in different fields. So, to solve these different optimization problems there are many techniques exist in literature namely exact algorithms, approximate algorithms, heuristic algorithms, and metaheuristic algorithms (Desale *et al.* [4]). For each given optimization problem, exact algorithms always offer the optimum answer. While a close optimum solution can be found using approximate techniques. Heuristic algorithms are problem-specific and problem-dependent. A metaheuristic, on the other hand, is a high-level, problem-independent algorithmic structure that provides a set of recommendations or tactics. We can see the classification of optimization algorithms from Figure 2.

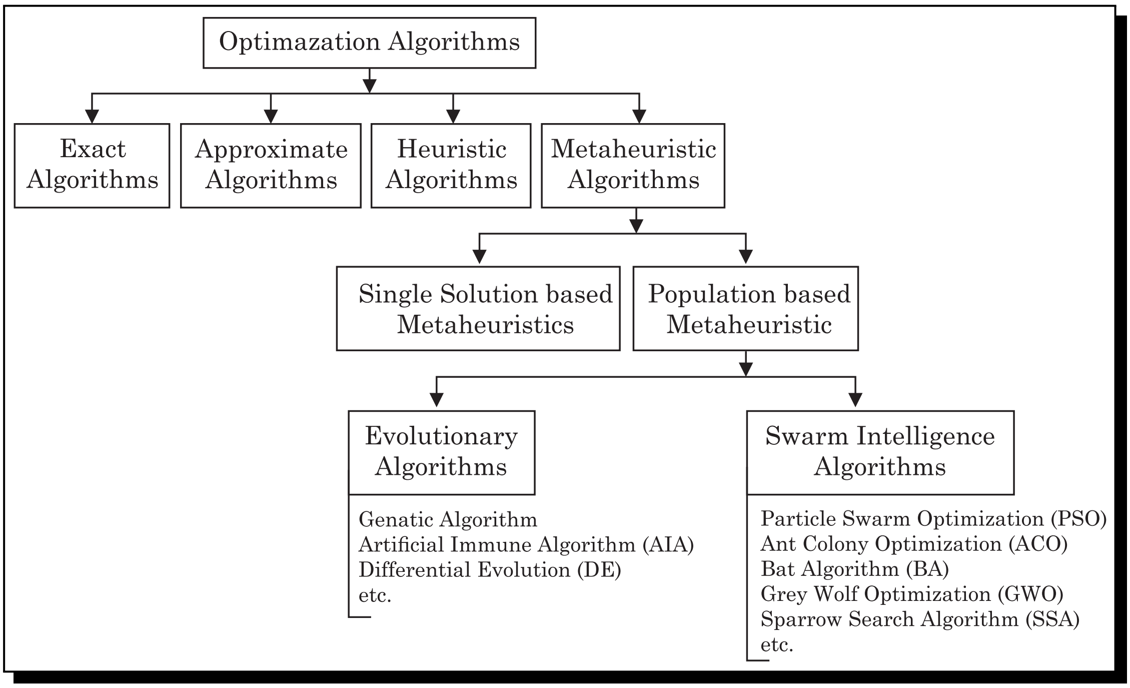


Figure 2. Classification of optimization algorithms

Even though there are many algorithms in the literature, as shown in Figure 2, researchers continue to work on developing new algorithms and modifying or combining the ones that already exist to find the best answer in terms of stability, convergence accuracy, and optimality. There is a significant number of SI algorithms, like ABC (Karaboga [9]), GWO (Mirjalili *et al*. [15]), ACO [1], PSO (Boeringer and Werner [3], and Sun *et al*. [27]), BBA (Nakamura *et al*. [17]), etc. If we focus on the most current proposed SI algorithm then by Xue and Shen in 2020 [36], the *Sparrow Search Algorithm* (SSA) is the most highlighted SI Algorithm. In the optimization algorithm SSA, the goal is to find the best solution by exploring the search space and adjusting the solutions iteratively. However, there are two common problems that can arise in this algorithm:

Sliding into Local Optima: Optimization algorithms can sometimes get stuck in local optima, which are suboptimal solutions that are better than their immediate neighbors but not the global optimum.

Persistent Convergence: Some algorithms may converge too quickly or slowly, making it difficult to find an optimal solution within a reasonable time frame.

Therefore, we presented a modified *Niching Sparrow Search Algorithm* (NSSA) to address these drawbacks. In this NSSA, we present the SSA by using the Niching strategy for updating follower and scouter positions. Niching strategy is incorporated into NSSA, is designed to address these issues by promoting diversity in the solutions and encouraging exploration of different regions of the search space of SSA. Here's how NSSA can overcome these drawbacks:

[1] A. Akhtar, Evolution of ant colony optimization algorithm – a brief literature review, *arXiv preprint* arXiv:1908.08007 (2019), DOI: 10.48550/arXiv.1908.08007.

Sliding into Local Optima: NSSA incorporates Niching techniques to maintain multiple candidate solutions in different niches (subregions of the search space). By doing so, it prevents premature convergence toward a single local optimum. The Niching mechanism encourages the algorithm to explore various niches, increasing the chances of discovering the global optimum.

Persistent Convergence: NSSA balances exploration and exploitation more effectively. The Niching approach maintains a diverse set of solutions, preventing the algorithm from converging too quickly. This allows NSSA to continue exploring the search space, especially if a better solution exists outside the current niche.

Niching strategies in NSSA involves technique like fitness sharing. This technique ensure that the algorithm does not focus exclusively on a single region of the search space but rather spreads its search effort across multiple niches. This diversity helps NSSA to find high-quality solutions efficiently and overcome the limitations associated with local optima and convergence.

In summary, NSSA addresses the drawbacks of SSA by incorporating Niching strategie that promote diversity in the solutions, prevent premature convergence, and encourage exploration of various niches within the search space. This allows NSSA to search for and potentially find better solutions to complex optimization problems. The remaining paper is organized in the manner shown below:

A summary of the literature on metaheuristic algorithms and also on SSA's latest research is shown in Section 2. Sparrow Search Algorithm is explained in Section 3 while Niching Technique is covered in Section 4. Sections 5, 6, and 7 respectively discuss the Niching Sparrow Search Algorithm, preparation of the experiment, and results of experiment and analysis. Section 8 provides an application of NSSA on engineering design problems. Section 9 ends the entire effort at the end, and this paper presents the findings.

2. Literature Review

Researchers have developed, modified, and hybridized a variety of heuristic and metaheuristic *Swarm Intelligence* (SI) optimization strategies to get the desired outcomes. As a result, we can focus on the literature below for this. Rao *et al*. [22] presented the TLBO for solving optimization problems of constrained mechanical design. By using a multiagent adaptive system known as the SPARROW algorithm, Roopa [24] has conducted an inquiry to identify the component cluster in parallel. The experiment's findings were essential since they would be used to produce effective component-based software architecture. By stabilizing the exploration techniques in *Cuckoo Search* (CS), Mlakar [16] has given the hybrid cuckoo search algorithm as a solution to engineering design optimization problems that include constraints. In order to address the issue of low positioning accuracy relying on DV-Hop (positioning technique) in the *Wireless Sensor Network* (WSN), Lei *et al*. [11] enhanced SSA. In order to strengthen SSA's global search capabilities, Peng *et al*. [19] applied it to the sensor network convergence problem for bridge monitoring and achieved successful results. An *Improved Sparrow Search Algorithm* (ISSA) was presented by Song *et al*. [26] to address the SSA's shortcomings in beginning population quality, population variety, and searchability. In order to construct the initial population for a greater quality of convergence, the ISSA used a chaotic method based on skew tent maps. Learning SSA was recommended by Ouyang *et al*. [18] as a solution to the robot path planning issue and the CEC 2017 test function. In order to study optimization problems and avoid the issue of local minima dropping in the original SSA, Yang *et al*. [38] employed T-distribution mutation coupled with chaotic mapping in adaptive SSA. An innovative sparrow search method was employed by Wu *et al*. [34] to solve the problem of a TSP. Inadequate stagnation while applying to TSP is another issue SSA faces in addition to falling into local optima. A cosine and sign search approach based on a new greedy genetic SSA was employed to solve this issue. For the purpose of forecasting the end-point phosphorus content of BOF, Quan *et al*.[2] combined the SSA model with the DELM model to create the ESSA-DELM. To avoid slipping into the local optimum and to upgrade SSA's ability to explore the world broadly, the Cauchy mutation and trigonometric substitution mechanism were included. For managing issues such as dropping local minima and a constant rate of convergence, Tang *et al*. [28] suggested a chaotic SSA. In order to overcome the engineering difficulties, adaptive step and logarithmic spiral techniques were added to chaotic SSA. The hybrid PSO and SSA were presented by Yang *et al*. [39] in order to anticipate software problems. PSO and SSA were combined because SSA has great resilience, good stability, quick convergence, and high search accuracy while PSO has a slow convergence but a high solution accuracy. The best site for the wind turbine on the wind farm was discovered by SSA, according to Kumar and Reddy [10]. Robot route planning issues have become a popular topic in the field of research, according to Li [12]. To address the drawback of the traditional raster method for path planning, SSA was developed. By employing a hybrid reverse learning technique and iterative chaotic mapping that is indefinitely folded to address engineering optimization problems with constraints, Wang *et al*. [31] updated SSA into the IHSSA algorithm. To address issues with unmanned aerial vehicle path planning, Wang [30] has combined ESSA and PSO. By incorporating a random walk method into SSA, Xie *et al*. [35] developed an enhanced SSA with regard to local and global optimization problems. The *Tent Lévy Flying Sparrow Search Method* (TFSSA), developed by Yan *et al*. [37], is used to choose the best subset of features in the packing pattern for the reason of classification.Experimental findings support the benefits of the given technique on other wrapper-based algorithms in relation to classification accuracy improvement and feature selection reduction. A recent and reliable approach for tackling optimization issues, the SSA, is reviewed by Gharehchopogh *et al*. [5]. They cover every article on variations, enhancement, optimization, and hybridization in the SSA literature. Studies show that the use of SSA has been equal to 32%, 36%, 28% and 4%, respectively, in the aforementioned areas. Ren *et al*. [23] suggest the *Multi-Strategy-Sparrow Search Algorithm* (MSSSA) as an enhanced optimization technique to address situations with extremely non-linear optimization. An *improved multi-strategies sparrow search algorithm* (EMSSA) was given

[2] L. Quan, A. Li, G. Cui, and S. Xie, Using enhanced sparrow search algorithm-deep extreme learning machine model to forecast end-point phosphorus content of BOF, *Preprints* **2021** (2021), 2021120192, DOI: 10.20944/preprints202112.0192.v1.

by Ma *et al*. [14] and is based on three strategies: the uniformity-diversification orientation strategy, the dynamic evolutionary strategy, and the hazard-aware transfer strategy. These three strategies particularly address the shortcomings of SSA. To overcome the problems that SSA has, such as a propensity for zero locations and a tendency to slip into local optima, Huang *et al*. [8] created a *Non-uniform Mutation Sparrow Search Algorithm* (NMSSA).

According to the literature mentioned above, *Sparrow Search Algorithm* (SSA)'s limitations are continually being discovered by researchers. This inspired us to change SSA. For that reason to address the drawbacks of SSA, we suggested a *Niching Sparrow Search Algorithm* (NSSA). We use the Niching technique in traditional SSA. Then it was examined using 18 benchmark test functions, the design of a speed reducer, and Himmelblau's nonlinear optimization problem.

3. Sparrow Search Algorithm

The SSA algorithm primarily relies on the sparrows' collective intelligence, ability to find food, and anti-predator behaviour. There are numerous bird sparrow species, however, in this experiment virtual sparrows were taken into account. They were divided into three groups: leaders (producers), followers (scroungers), and scouters. Only 20% of the sparrows with high levels of strength were employed as leaders, and the other 80% were followers. In order to get food, the followers follow the leader's directions. Leaders occasionally take on the role of followers during this food-searching process, and vice versa, but the proportion of followers to leaders does not change. In the meantime, when some sparrows sense danger nearby named scouters, they direct the entire population to a safe region.

In this experiment, firstly initialization of the position of sparrows is needed. The population (position of sparrows) can be expressed by the subsequent matrix:

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d} \end{bmatrix}. \tag{3.1}$$

In equation (3.1), n is the number of sparrows and d depicts the dimension of the variables to be optimized. The position of leaders is updated by equation (3.2):

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^{t} \cdot \exp\left(\frac{-i}{\alpha \cdot iter_{\max}}\right) & \text{if } R_2 < ST, \\ X_{i,j}^{t} + Q \cdot L & \text{if } R_2 \geq ST. \end{cases} \tag{3.2}$$

In equation (3.2), the index C_j ranges from 1 to d, and the current iteration is denoted as t. The value of α, which falls within the range of $(0,1]$, is considered as a random number. The notation $X_{i,j}$ represents the spatial position of the ith sparrow in the jth dimension. The constant $iter_{\max}$ signifies the maximum number of iterations. The term ST ($ST \in [0.5, 1.0]$) serves as a safety threshold, while R_2 ($R_2 \in [0,1]$) functions as an alarm value. The symbol L corresponds to a matrix of size $1 \times d$, where each element is 1. The variable Q is generated as a random number following a normal distribution.

When $R_2 < ST$, it indicates the absence of threats near the sparrows, enabling the locators to search on a larger scale. On the other hand, when $R_2 \geq ST$, it signifies the presence of threats around the sparrows. In such cases, all sparrows immediately adjust their positions toward a safe area.The position of followers is updated by equation (3.3):

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{\text{worst}}^t - X_{i,j}^t}{i^2}\right) & \text{if } i > \frac{n}{2}, \\ X_p^{t+1} + |X_{i,j}^t - X_p^{t+1}| \cdot A^+ \cdot L & \text{if } i \leq \frac{n}{2}. \end{cases} \tag{3.3}$$

In equation (3.3), X_{worst} represents the current global worst position, while X_p signifies the optimal location chosen by the locator. The total count of sparrows is denoted as n. The matrix A has a dimension of $1 \times d$, where each element takes either the value 1 or -1. Furthermore, it holds that $A^+ = A^T(A \cdot A^T)^{-1}$. In cases, where $i > \frac{n}{2}$, this indicates that the ith asker possesses an unfavorable fitness value and consequently does not acquire any sustenance, potentially leading to starvation. The update for the positions of scouters is governed by the subsequent equation (3.4):

$$X_{i,j}^{t+1} = \begin{cases} X_{\text{best}}^t + \beta \cdot |X_{i,j}^t - X_{\text{best}}^t| & \text{if } f_i > f_g, \\ X_{i,j}^t + K \cdot \left(\frac{|X_{i,j}^t - X_{\text{worst}}^t|}{(f_i - f_w) + \varepsilon}\right) & \text{if } f_i = f_g. \end{cases} \tag{3.4}$$

In the provided equation (3.4), X_{best} corresponds to the current global optimal location. The parameter β, governing the step size, is drawn from a standard normal distribution of random numbers, denoted as $N(0,1)$. Additionally, K which lies within the range of $[-1,1]$, is a randomly generated number. The fitness value of the current sparrow is denoted as f_i, while f_w and f_g represent the present worst and global best fitness values, respectively. The constant ε serves as the smallest value to prevent errors arising from division by zero. When $f_i > f_g$, it indicates that the sparrows are situated at the periphery of the group. Conversely, when $f_i = f_g$, this signifies that the sparrows located in the population's center are fully aware of potential threats and should adjust their positions to be closer to others.

4. Niching Technique

Niching technique is a method used in evolutionary algorithms and optimization to address the problem of premature convergence. It is also known as speciation, island model or multi-modal optimization. Premature convergence occurs when the population in the evolutionary algorithm converges too early to a suboptimal solution, and the algorithm is unable to explore the search space further to find better solutions. Niching technique involves dividing the search space into smaller subspaces or niches, where each niche represents a region that contains a potential optimal solution. The goal is to maintain diversity in the population by assigning individuals to different niches, where they can evolve and improve independently from each other. This helps to avoid premature convergence and allows the algorithm to explore multiple regions of the search space simultaneously. The Niching technique can be implemented using various methods such as fitness sharing, crowding, and sharing trees. Fitness sharing is a method that penalizes the fitness of individuals that are close to each other in the search space, to encourage diversity. Crowding is a method that selects individuals in the population based on their distance from each other, where the individuals with the highest fitness and the largest distance are selected. Sharing trees is a method that creates a hierarchical structure of subspaces, where each subspace represents a niche, and the individuals are assigned to the niches based on their position in the tree. Niching technique has been applied successfully in various optimization problems, such as function optimization, clustering, and classification. It has been shown to improve the performance of evolutionary algorithms, by maintaining diversity and exploring multiple regions of the search space simultaneously. The advantage of Niching techniques is that they have numerous solutions, allowing them to handle multi-modal functions and successfully arrive at the global optimal basin.

In this research paper, the Niching technique is implemented by using fitness sharing method. In fitness sharing method, the *Fitness Euclidean-distance Ratio SSA* (FER-SSA), created in accordance with the FDR-PSO concept (Ahmed *et al*. [1], Liang *et al*. [13], Peram *et al*. [20], and Qu *et al*. [21]), is used in the suggested technique to find the best neighborhood solutions rather than the global best solution. The subsequent equation has been applied to determine the FER between two sparrows, i as well as j:

$$\text{FER}_{ij} = \alpha \cdot \frac{f(p_i) - f(p_j)}{\|p_i - p_j\|}. \tag{4.1}$$

In equation (4.1), p_i and p_j are the personal best positions of the ith and jth sparrows respectively, and $f(p_i)$ and $f(p_j)$ illustrate their objective function values. The scaling factor α is defined by the given equation (4.2):

$$\alpha = \frac{\|S\|}{f(p_b) - f(p_w)}. \tag{4.2}$$

In equation (4.2), $f(p_b)$ and $f(p_w)$ are the objective function values of best and worst sparrows from the whole population, where $\|S\|$ is the size of search space, which is evaluated by diagonal distance as mentioned in equation (4.3), in which x_r^u depicts the upper bound whereas x_r^l represents the lower bound of the search space's rth dimension.

$$\|S\| = \sqrt{\sum_{r=1}^{D} (x_r^u - x_r^l)^2}. \tag{4.3}$$

Niching is an effective method that might satisfy the requirements of huge exploration capability for solving optimization problems. To keep track of the best solutions found thus far during the search process, we included the personal best features in the suggested SSA algorithm. At each iteration, the sparrows will be directed towards more of the fittest neighborhood values that can be determined by calculating the FER values. The advantage of the FER value is that no parameter specification is required.

5. Niching Sparrow Search Algorithm

In each iteration time we calculate $nbest_i$ and the steps of determining the *nbest* for the ith sparrow for FER-SSA are depicted below in Algorithm 1.

Algorithm 1: The pseudocode of finding $nbest$ for sparrow i for FER-SSA

Input: All sparrows from the whole the population

Output: Neighborhood best $nbest$ depends upon the ith sparrow's FER value

```
FER ← 0, tmp ← 0, nbest_i ← pbest_i
For i = 1 to n (total number of sparrows)
  For j = 1 to n (total number of sparrows)
  Calculate the FER using equation (4.1)
    If j = 1
      tmp = FER
    Endif
    If FER > tmp
      tmp = FER
      nbest_i = pbest_j
    Endif
  Endfor
Endfor
```

The $nbest_i$ is further used in order to modify the equations of followers as well as scouters. Modification of the followers and scouters equations can be seen in equations 5.1 and (5.2), respectively:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{\text{worst}}^t - X_{i,j}^t}{i^2}\right) & \text{if } i > \frac{n}{2}, \\ nbest_i + |X_{i,j}^t - nbest_i| \cdot A^+ \cdot L & \text{if } i \le \frac{n}{2}, \end{cases} \tag{5.1}$$

$$X_{i,j}^{t+1} = \begin{cases} nbest_i + \beta \cdot |X_{i,j}^t - nbest_i| & \text{if } f_i > f_g, \\ X_{i,j}^t + K \cdot \left(\frac{|X_{i,j}^t - X_{\text{worst}}^t|}{(f_i - f_w) + \varepsilon}\right) & \text{if } f_i = f_g. \end{cases} \tag{5.2}$$

Then, we define a *Niching Constant* (NC) as 0.5 and also create a random number (r_i) during every iteration period. If (r_i) >NC, then followers and scouters are guided by $nbest_i$ applying equations (5.1) and (5.2) respectively, if not, then followers and scouters are guided by real equations (3.3) as well as (3.4) subsequently. We should keep one thing in our mind that leaders always presented by real position update equation namely equation (3.2). The proper working of the Niching Sparrow Search Algorithm is shown in Figure 3:

6. Preparation of the Experiment

6.1 Experimental Environment and Fixing Up the Parameters

Each of the five algorithms was executed on a Windows 10 platform, utilizing a 64-bit operating system with 16.0 GB RAM, and operated on an Intel(R) Core(TM) i5-7300U CPU @ 2.60 GHz processor. The implementation of all algorithms was carried out using MATLAB R2014a.

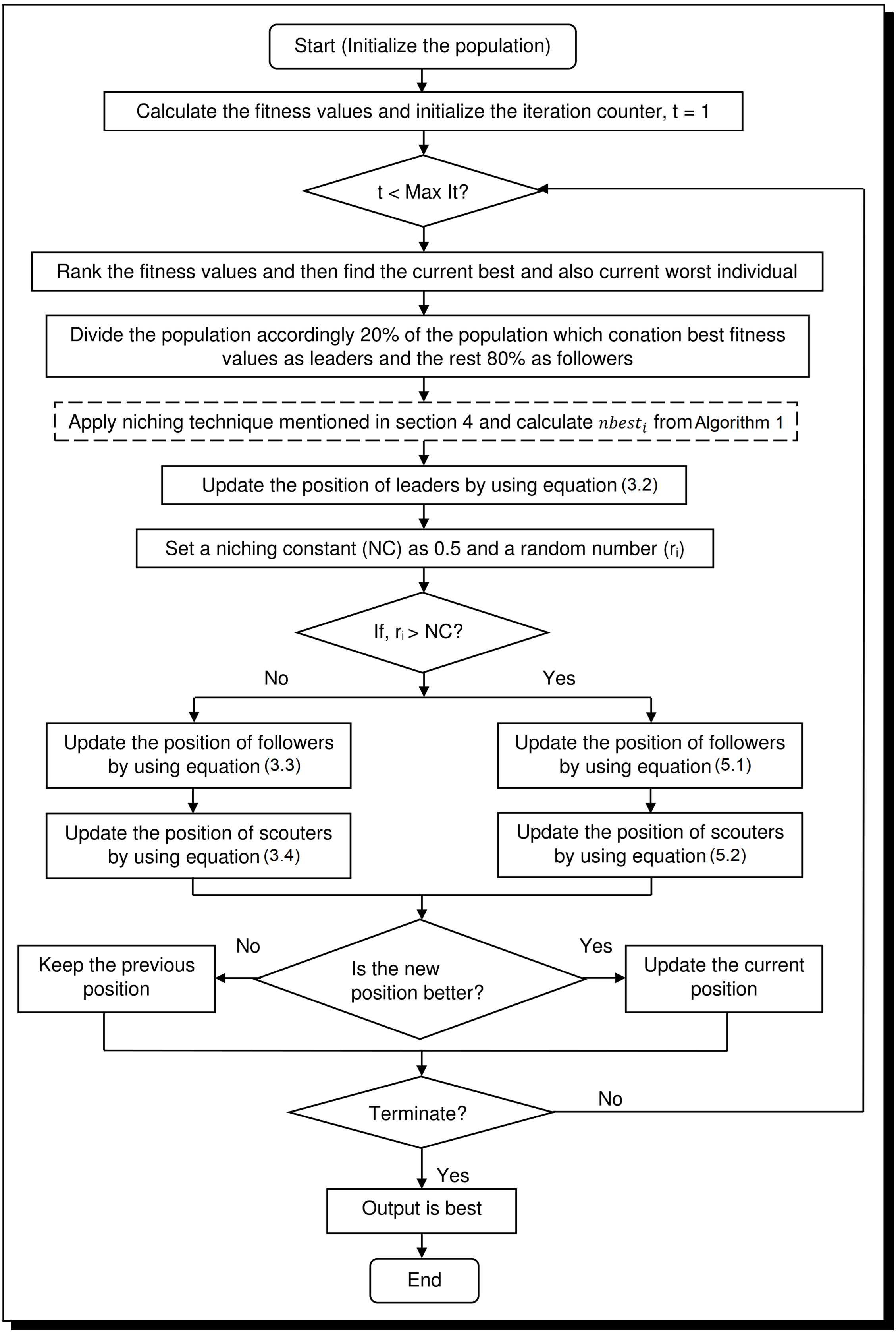


Figure 3. Flow chart of NSSA

The assessment of the *Niching Sparrow Search Algorithm's* (NSSA) optimality, effectiveness, and convergence was conducted using eighteen standard benchmark functions. To validate its performance, NSSA's execution outcomes were compared against those of various other metaheuristic techniques, including SSA, PSO, GWO, and GSA. Each technique employed an identical number of search agents, totaling 100, and the maximum iterations were set at 1000 for all methods. The specific parameters for each algorithm are detailed in Table 1.

Table 1. Parameters of all five algorithms

Algorithm	Parameter	Value
	Personal Learning Coefficient: c1	1.49445
PSO	Global Learning Coefficient: c2	1.49445
	Inertial weight: ω	$\omega = 0.729$
GWO	Random numbers: r_1, r_2	$[0,1]$, $[0,1]$
	$\vec{a}$	Decrease linearly from 2 to 0
GSA	G_0	100
	α	20
SSA	Safety threshold: ST	0.8
NSSA	Safety threshold: ST	0.8

6.2 Standard Test Functions

To check the algorithm appropriately, in every case then we run each algorithm 30 times separately on every benchmark test function. Eighteen benchmark test functions (Song *et al*. [25]) are taken out of which five are unimodal test functions, four multimodal test functions, and nine fixed-dimension multimodal benchmark test functions corresponding to Tables 2-4, respectively.

6.2.1 Unimodal Benchmark Test Functions

The unimodal test functions primarily represent the good exploitation and convergence properties of an algorithm. In general, these test functions are intended to focus the algorithm on exploiting while the optimization process is used to determine the global best.

6.2.2 Multimodal Benchmark Test Functions

Multimodal test functions personify multiple local optimal solutions. Through this, an algorithm easily falls into the local optimal points. So, it can be used for evaluating the global and local search abilities of the algorithm.

6.2.3 Fixed-dimension Multimodal Benchmark Test Functions

Fixed-dimension test functions are used to evaluate the full performance of the algorithm such as convergence accuracy, convergence speed, and stability.

Table 2. Unimodal benchmark test functions

FUNCTION	DIM	RANGE	MIN
$F_1(z)=\sum_{i=1}^{n} z_i^2$	30	$[-100,100]$	0
$F_2(z)=\sum_{i=1}^{n} \lvert z_i\rvert+\prod_{i=1}^{n} \lvert z_i\rvert$	30	$[-10,10]$	0
$F_3(z)=\max\{\lvert z_i\rvert, 1\le i\le n\}$	30	$[-100,100]$	0
$F_4(z)=\sum_{i=1}^{n-1}[100(z_{i+1}-z_i^2)^2+(z_i-1)^2]$	30	$[-30,30]$	0
$F_5(z)=\sum_{i=1}^{n} iz_i^4+\text{random}[0,1)$	30	$[-1.28,1.28]$	0

Table 3. Multimodal benchmark test functions

FUNCTION	DIM	RANGE	MIN
$F_6(z)=\sum_{i=1}^{n} -z_i\sin(\sqrt{\lvert z_i\rvert})$	30	$[-500,500]$	$-418.9829\times\text{DIM}$
$F_7(z)=\sum_{i=1}^{n}[z_i^2-10\cos(2\pi z_i)+10]$	30	$[-5.12,5.12]$	0
$F_8(z)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} z_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi z_i)\right)+20+e$	30	$[-32,32]$	0
$F_9(z)=\frac{1}{4000}\sum_{i=1}^{n} z_i^2-\prod_{i=1}^{n}\cos\left(\frac{z_i}{\sqrt{i}}\right)+1$	30	$[-600,600]$	0

Table 4. Fixed-dimension multimodal benchmark test functions

FUNCTION	DIM	RANGE	MIN
$F_{10}(z)=\sum_{i=1}^{11}\left[\alpha_i-\frac{z_1(\beta_i^2+\beta_i z_2)}{\beta_i^2+\beta_i z_3+z_4}\right]^2$	04	$[-5,5]$	0.00030
$F_{11}(z)=4z_1^2-2.1z_1^4+\frac{1}{3}z_1^6+z_1z_2-4z_2^2+4z_2^4$	02	$[-5,5]$	-1.0316
$F_{12}(z)=\left(z_2-\frac{5.1}{4\pi^2}z_1^2+\frac{5}{\pi}z_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos z_1+10$	02	$[-5,5]$	0.398
$F_{13}(z)=[1+(z_1+z_2+1)^2(19-14z_1+3z_1^2-14z_2+6z_1z_2+3z_2^2)]$ $\times[30+(2z_1-3z_2)^2(18-32z_1+12z_1^2+48z_2-36z_1z_2+27z_2^2)]$	02	$[-2,2]$	3
$F_{14}(z)=-\sum_{i=1}^{4} c_i\exp\left(-\sum_{j=1}^{3}\alpha_{ij}(z_j-p_{ij})^2\right)$	03	[0, 1]	-3.86
$F_{15}(z)=-\sum_{i=1}^{4} c_i\exp\left(-\sum_{j=1}^{6}\alpha_{ij}(z_j-p_{ij})^2\right)$	06	[0, 1]	-3.32
$F_{16}(z)=-\sum_{i=1}^{5}\left[(z-\alpha_i)(z-\alpha_i)^T+c_i\right]^{-1}$	04	[0, 10]	-10.1532
$F_{17}(z)=-\sum_{i=1}^{7}[(z-\alpha_i)(z-\alpha_i)^T+c_i]^{-1}$	04	[0, 10]	-10.4028
$F_{18}(z)=-\sum_{i=1}^{10}[(z-\alpha_i)(z-\alpha_i)^T+c_i]^{-1}$	04	[0, 10]	-10.5363

7. Experimental Results and Its Analysis

In this paper, we performed 30 independent runs of all five algorithms namely NSSA, SSA, PSO, GSA, and GWO on 18 standard benchmark test functions given in Tables 2-4 and calculated the optimal value, worse value, average value (mean), and standard deviation also. The optimal value and worse value test the optimality whereas the mean value resents convergence accuracy and the standard deviation represents the stability of every algorithm. Implementation of all algorithms on the benchmark problems has been done through MATLAB. The results of every algorithm are represented in Tables 5-6.

Table 5. Comparative output analysis of test functions (F1-F9)

Fun	Algorithm	Best	Worse	Ave	Std
F1	NSSA	0.0	0.0	0.0	0.0
	SSA	0.0	0.0	0.0	0.0
	GWO	1.0991e-88	5.3324e-84	2.38075432e-85	9.68740481e-85
	PSO	3.529e-24	1.0586e-20	5.80739203e-22	1.94032827e-21
	GSA	2.4014e-18	5.8305e-18	3.9482867e-18	8.5399790e-19
F2	NSSA	0.0	0.0	0.0	0.0
	SSA	0.0	2.4363e-265	7.6134375e-267	0.0
	GWO	3.1362e-50	1.5025e-48	3.10268267e-49	4.41283557e-49
	PSO	6.699e-14	4.8535e-08	3.074544767e-09	1.117369736e-08
	GSA	8.246e-09	1.2888e-08	1.051743e-08	1.26887657e-09
F3	NSSA	0.0	0.0	0.0	0.0
	SSA	0.0	4.6668e-292	1.5557e-293	0.0
	GWO	3.5385e-23	7.1014e-21	1.16876584e-21	1.42176540e-21
	PSO	0.020903	0.25476	0.08997477	0.06528445
	GSA	7.9489e-10	1.3805e-09	1.014846e-09	1.4645425296e-10
F4	NSSA	1.0957e-10	1.6682e-06	1.25957e-07	3.10374e-07
	SSA	1.5554e-10	1.5809e-05	1.467534258e-06	3.93947986e-06
	GWO	24.7159	27.1108	26.13573	0.603258538
	PSO	0.68972	85.6878	33.557804	25.56565453
	GSA	25.6993	26.1444	25.9632600	0.118978678
F5	NSSA	2.1848e-07	0.00013039	2.4344642e-05	3.7290395e-05
	SSA	4.6705e-06	0.00019127	9.392409e-05	5.31149469e-05
	GWO	7.033e-05	0.00052936	0.0002425677	0.00011887
	PSO	0.0026019	0.021313	0.008172467	0.003795079
	GSA	0.0020925	0.0095959	0.005225633	0.0019031837
F6	NSSA	−12569.4866	−8522.3346	−9428.2056767	1092.9226710
	SSA	−9370.0224	−7770.875	−8520.775323	406.14073504
	GWO	−7469.0088	−3879.5123	−6153.069257	812.77481648
	PSO	−9628.1914	−8009.4179	−8849.01495	379.53600727
	GSA	−4421.6216	−2458.2552	−3084.6380033	470.23092778
F7	NSSA	0.0	0.0	0.0	0.0
	SSA	0.0	0.0	0.0	0.0
	GWO	0.0	0.0	0.0	0.0
	PSO	30.8437	72.6319	46.1992033	10.262628085
	GSA	0.99496	11.9395	6.76573067	2.5251881
F8	NSSA	8.8818e-16	8.8818e-16	8.8818e-16	0.0
	SSA	8.8818e-16	8.8818e-16	8.8818e-16	0.0
	GWO	7.9936e-15	1.5099e-14	1.02435933e-14	3.02079341e-15
	PSO	9.992e-13	2.0119	0.60989333	0.69234175
	GSA	1.2624e-09	1.9214e-09	1.5908967e-09	1.754857720e-10
F9	NSSA	0.0	0.0	0.0	0.0
	SSA	0.0	0.0	0.0	0.0
	GWO	0.0	0.02255	0.0012708367	0.00447732370
	PSO	0.0	0.087839	0.017435953	0.01831456557
	GSA	1.0671	4.0935	1.76911667	0.7651032719

Table 6. Comparative output analysis of test functions (F10-F18)

Fun	Algorithm	Best	Worse	Ave	Std
	NSSA	0.00030749	0.00030749	0.00030749	0.0
F10	SSA	0.00030749	0.00030749	0.00030749	0.0
	GWO	0.00030749	0.020363	0.00106757834	0.0036549778
	PSO	0.00030749	0.020595	0.001109698	0.003694602
	GSA	0.00089935	0.002361	0.00189657	0.00035837667
	NSSA	−1.0316	−1.0316	−1.0316	0.0
F11	SSA	−1.0316	−1.0316	−1.0316	0.0
	GWO	−1.0316	−1.0316	−1.0316	0.0
	PSO	−1.0316	−1.0316	−1.0316	0.0
	GSA	−1.0316	−1.0316	−1.0316	0.0
	NSSA	0.39789	0.39789	0.39789	0.0
F12	SSA	0.39789	0.39789	0.39789	0.0
	GWO	0.39789	0.39789	0.39789	0.0
	PSO	0.39789	0.39789	0.39789	0.0
	GSA	0.39789	0.39789	0.39789	0.0
	NSSA	3	3	3	0.0
F13	SSA	3	3	3	0.0
	GWO	3	3	3	0.0
	PSO	3	3	3	0.0
	GSA	3	3	3	0.0
	NSSA	−3.8628	−3.8628	−3.8628	0.0
F14	SSA	−3.8628	−3.8628	−3.8628	0.0
	GWO	−3.8572	−3.8549	−3.86184	0.00251404
	PSO	−3.8628	−3.8628	−3.8628	0.0
	GSA	−3.8628	−3.8628	−3.8628	0.0
	NSSA	−3.322	−3.2031	−3.29425667	0.05114877
F15	SSA	−3.322	−3.2031	−3.25858667	0.0603318
	GWO	−3.322	−3.0867	−3.23738	0.08154989
	PSO	−3.322	−3.2031	−3.26255	0.06046631
	GSA	−3.322	−3.322	−3.322	0.0
	NSSA	−10.1532	−10.1532	−10.1532	0.0
F16	SSA	−10.1532	−10.1532	−10.1532	0.0
	GWO	−10.1532	−5.0552	−8.9710667	2.17910880
	PSO	−10.1532	−2.63	−7.46191	3.023418997
	GSA	−10.1532	−2.6829	−7.0141	3.445013967
	NSSA	−10.4029	−10.4029	−10.4029	0.0
F17	SSA	−10.4029	−10.4029	−10.4029	0.0
	GWO	−10.4028	−5.1286	−10.05109	1.3380811599
	PSO	−10.4029	−2.7659	−8.07438	3.146336541
	GSA	−10.4029	−10.4029	−10.4029	0.0
	NSSA	−10.5364	−10.5364	−10.5364	0.0
F18	SSA	−10.5364	−10.5364	−10.5364	0.0
	GWO	−10.5363	−10.536	−10.5362167	0.00011167
	PSO	−10.5364	−2.4217	−9.54799	2.293219724
	GSA	−10.5364	−10.5364	−10.5364	0.0

7.1 Analysis of Optimality and Convergence Accuracy

From Table 5, it could be visible that although NSSA gives the same fitness values as SSA gives for the functions F1-F3 but in terms of convergence accuracy the proposed NSSA is excellent. For the functions F4 and F5, NSSA delivers the best optimal results as well as convergence accuracy as opposed to SSA, GWO, and GSA. So, the performance of NSSA for solving unimodal test functions is superior in relation to exploitation.

Secondly, to test local search and global search abilities of our proposed NSSA we tested it on four multimodal benchmark probelms (F6-F9). In Table 5, NSSA shows good results than the other algorithms. For the F6 function, NSSA finds excellently global optimal solution by exploring the whole search space whereas other algorithms can not find such type of solution for this test function. Convergence accuracy of this F6 test function is also better in comparison with SSA, PSO, and GSA. Optimality and convergence accuracy of NSSA and SSA is same for solving F7-F9 test functions but better from PSO, GSA, and GWO. By concluding the performance of NSSA for solving multimodal benchmark functions, we can say that NSSA has robust expoloration ability.

Thirdly, the obtained solution of fixed-dimension multimodal benchmark functions are delinated in Table 6. All the five algorithms represents better performance among the exploration and exploitation on F11, F12, F13. For F10 test function all the algorithms gives optimal value except GSA but best convergence accuracy for this test function only represent by NSSA and SSA. NSSA, SSA, PSO, and GSA not only displays the same optimal value but also the same convergence accuracy for the F14 function, whereas GWO has worse value for it. For F15 benchmark function all five algorithms depicts similar optimal value but best average value only expressed by NSSA. In order to test F16 test function, the best optimal solution is found by all the five algorithms but the good convergence accuracy only presented by NSSA and SSA. On testing F17 test function, we are getting optimal value by all the algorithms such as NSSA, SSA, PSO, GWO, and GSA but by analysing the convergence accuracy we can find that PSO is worse in some extent. In F18 test function, every tested algorithm shows the good optimal value but convergence accuracy only displayed by NSSA, SSA, and GSA.

7.2 Analysis of Stability

In order to test the stability of all the algorithms for all the test functions, firstly we have discuss about unimodal test function. From Table 5, it can be seen that standard deviation of NSSA and SSA is zreo for the test functions F1-F3 that means NSSA and SSA are more stable as compare with other three algorithms such as GWO, PSO, and GSA. While dealing with F4 and F5 test functions, NSSA have better stability in relation with other four algorithms like SSA, PSO, GWO, and GSA. Thus, for all the five unimodal test function NSSA depicts better stability as opposed to other algorithms.

In the second place, stability of all techniques for four (F6-F9) multimodal test functions is tested. In Table 5 for F6 test function, we can see that the NSSA has good soluition accuracy but poor stability. For F7 test function, stability of NSSA, SSA, and GWO is better. For F8 and F9 test functions, both NSSA and SSA represents the good stability in relation with other techniques.

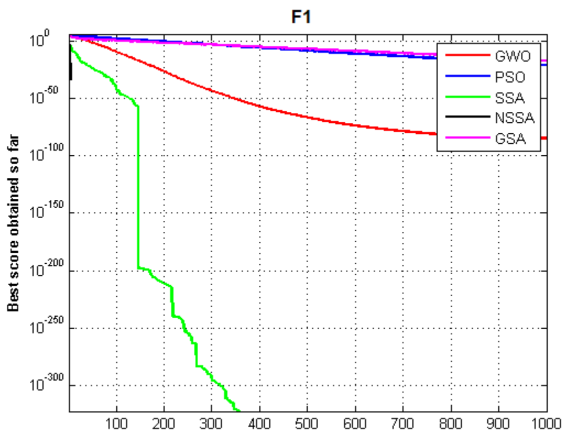
Finally, fixed-dimension multimodal test functions are taken to test the stability of all the algorithms. From Table 6, it is crystel clear that sability of NSSA is superior in comparison with other algorithms for the functions F10-F18.

Hence our proposed NSSA plays an extremely good role in terms of stability.

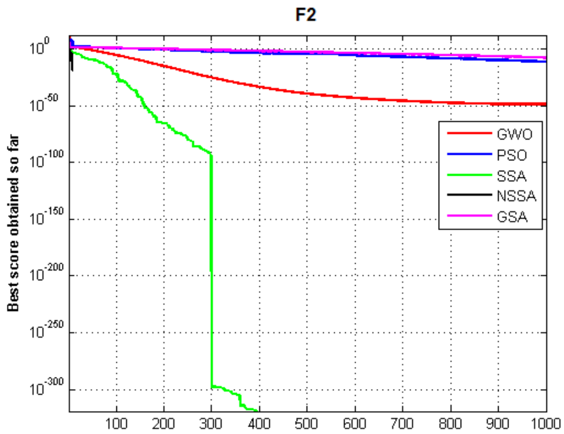
7.3 Analysis of Convergence Speed

If we move in the direction of convergence speed then we can see the fitness curves of unimodal test functions, multimodal test functions, and fixed-dimension multimodal test functions from Figures 4, 5, and 6, respectively.

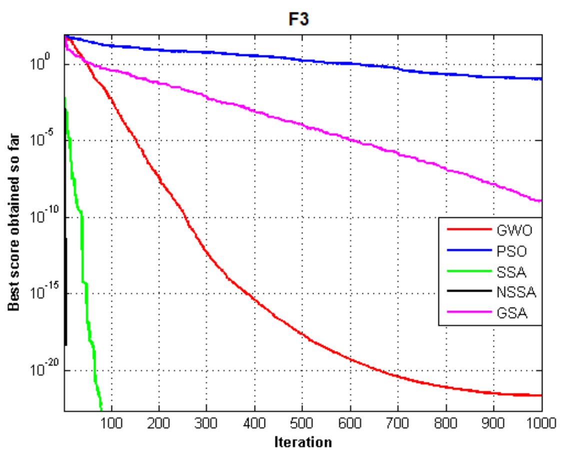
From Figures 4, 5, and 6 we conclude that convergence speed of our proposed NSSA is faster in comparison with SSA, GWO, PSO, and GSA. So, NSSA performs best in respect of convergence and efficiency while dealing with unimodal, multimodal, and fixed-dimension multimodal test functions in contrast with SSA, PSO, GWO, and GSA.



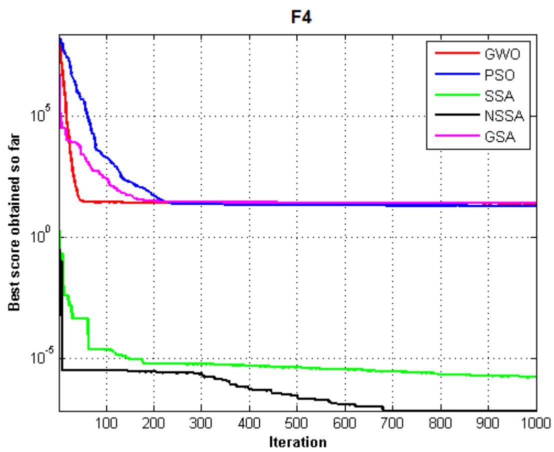
(a) F1



(b) F2

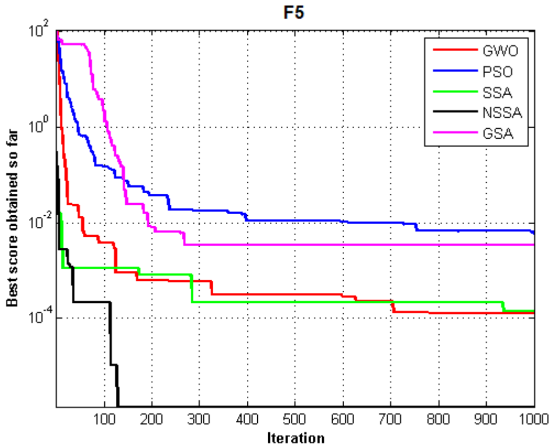


(c) F3



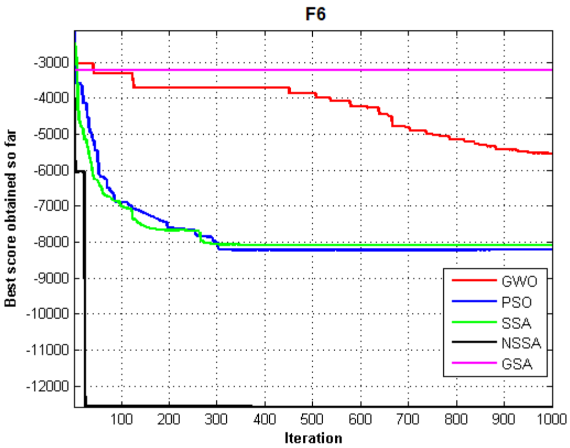
(d) F4

Contd. Figure

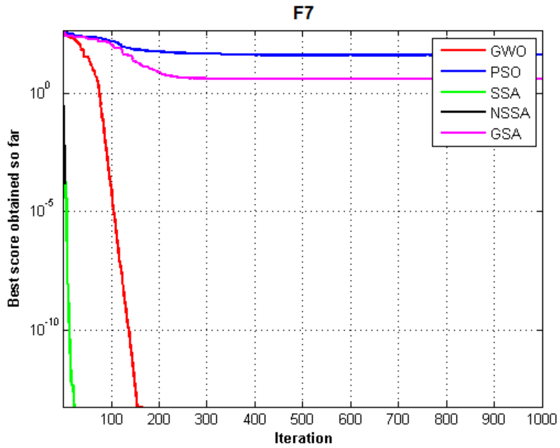


(e) F5

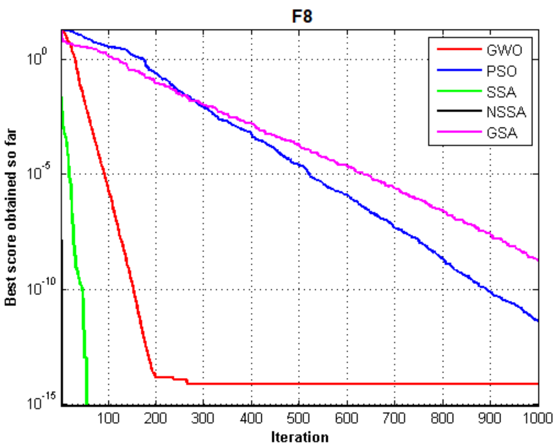
Figure 4. Performance comparison of five algorithms on unimodal test functions



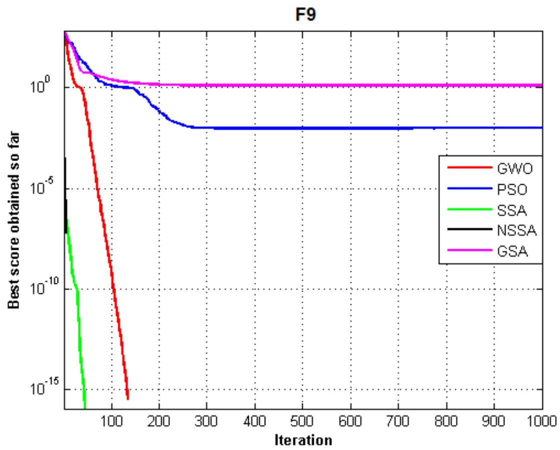
(a) F6



(b) F7

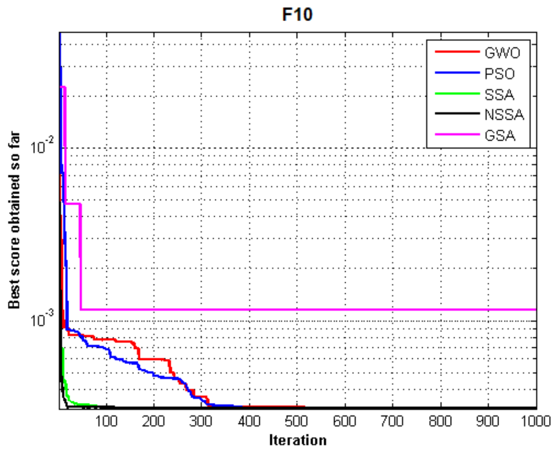


(c) F8

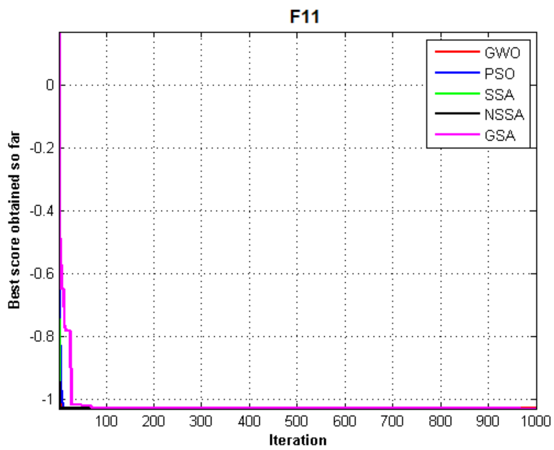


(d) F9

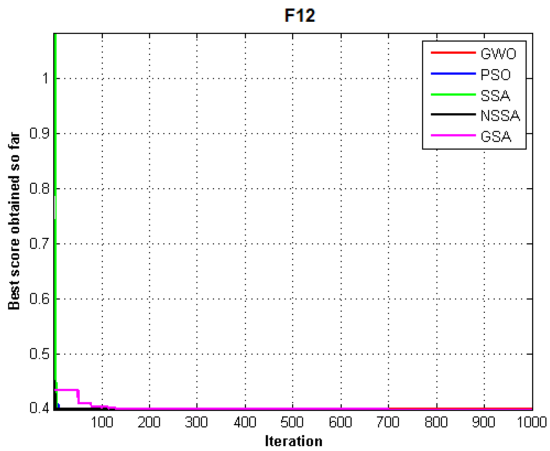
Figure 5. Performance comparison of five algorithms on multimodal test functions



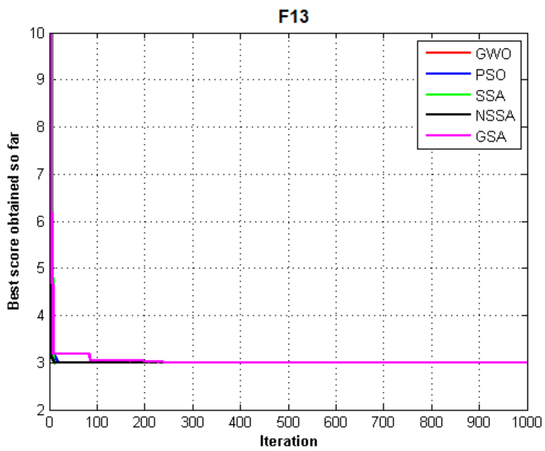
(a) F10



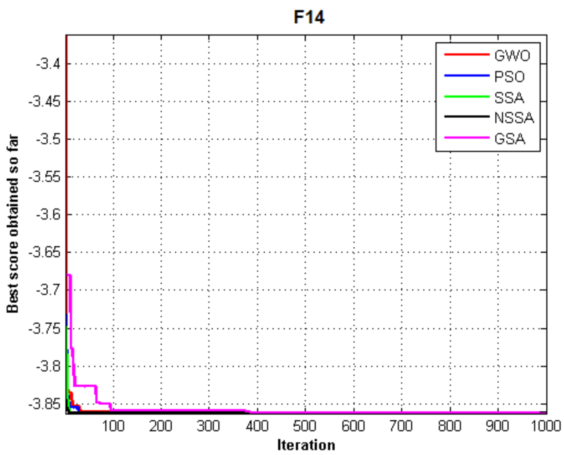
(b) F11



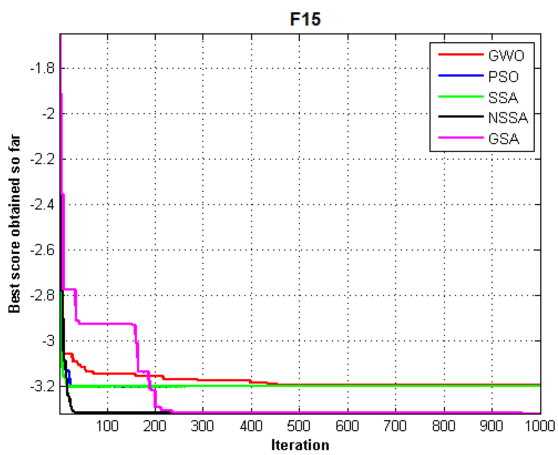
(c) F12



(d) F13

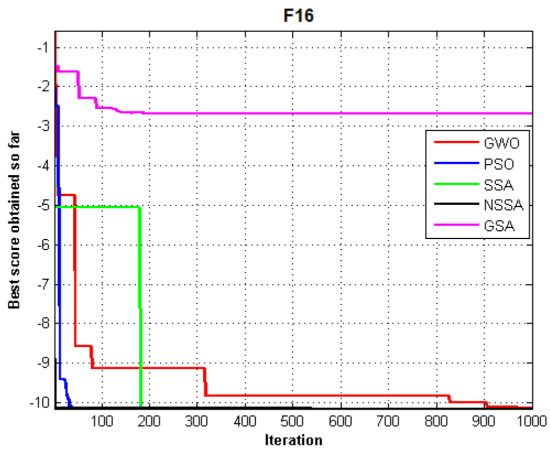


(e) F14

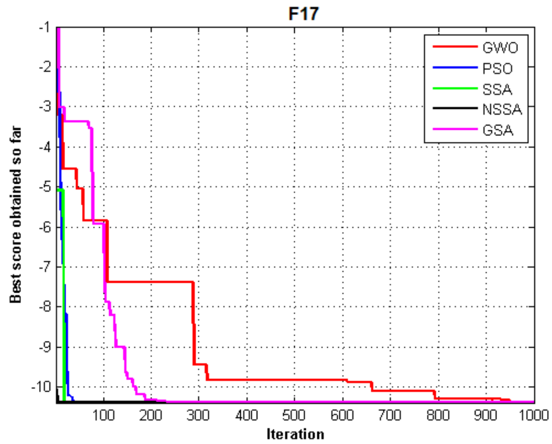


(f) F15

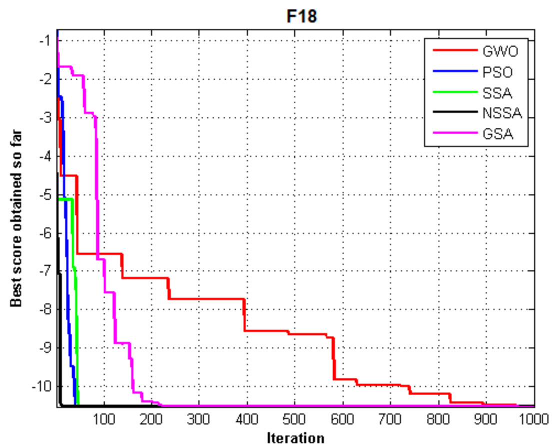
Contd. Figure



(g) F16



(h) F17



(i) F18

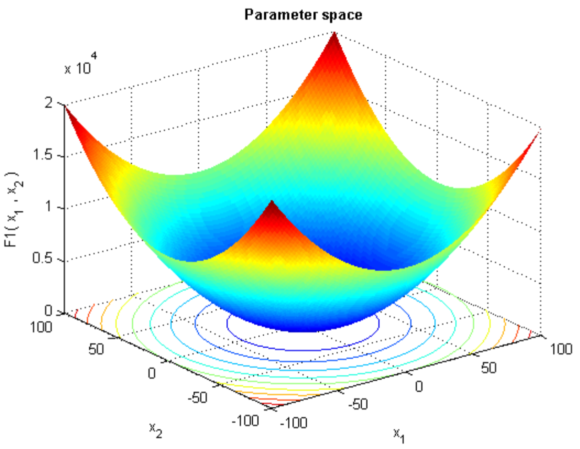
Figure 6. Performance comparison of five algorithms on fixed-dimension test functions

7.4 The Trajectories of Sparrows in Different Test Functions

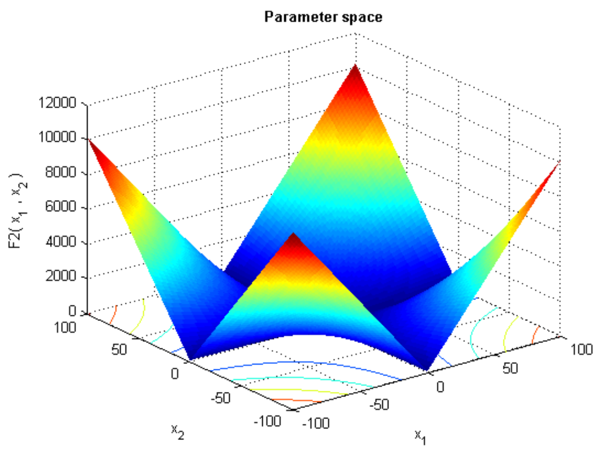
Figures 7, 8, and 9 represent the trajectories of sparrows in unimodal, multimodal, and fixed-dimension multimodal test functions respectively. That means the path of the NSSA on the 3-D version of the test functions can be seen from these figures. We can clearly see most of the sparrows combine towards the global optimum. Some sparrows are clustered at the local minima in which most of the sparrows can avoid local minima for moving towards the global best.

7.5 Wilcoxon Signed Rank-Test

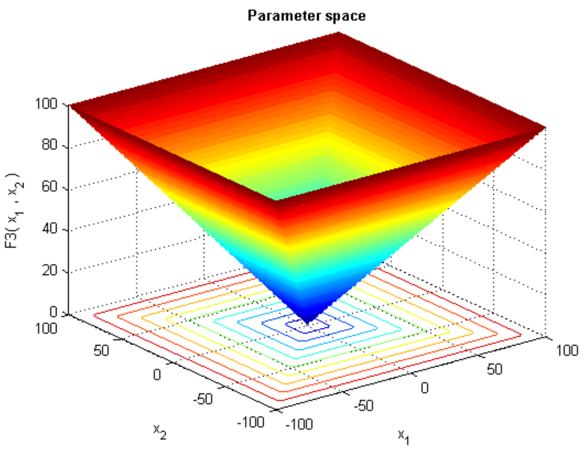
The Wilcoxon signed-rank test operates as a statistical method centered around the arrangement of observations within a sample (Wilcoxon *et al*. [32]). The algorithm with the smallest assigned rank is identified as the top performer, while the opposite holds true as well. The outcomes of this statistical ranking evaluation for all algorithms are detailed in Table 7, while Table 8 presents a summary of these ranks.



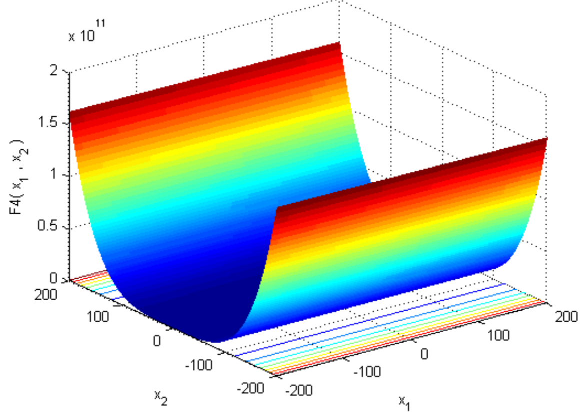
(a) F1



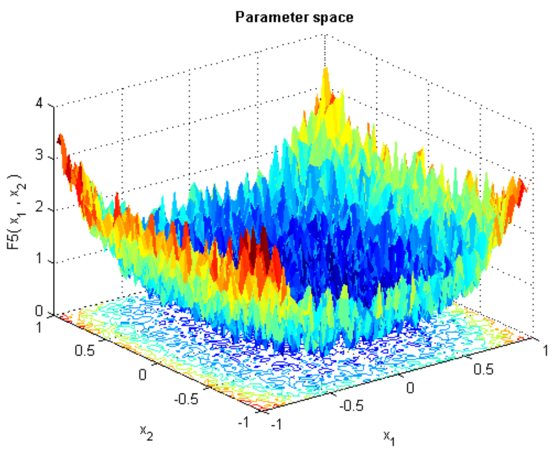
(b) F2



(c) F3

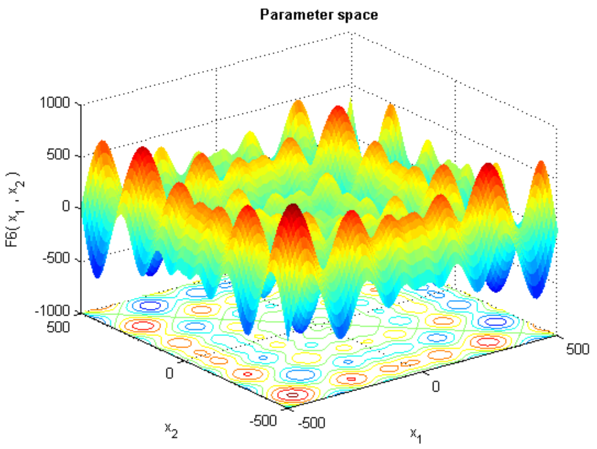


(d) F4

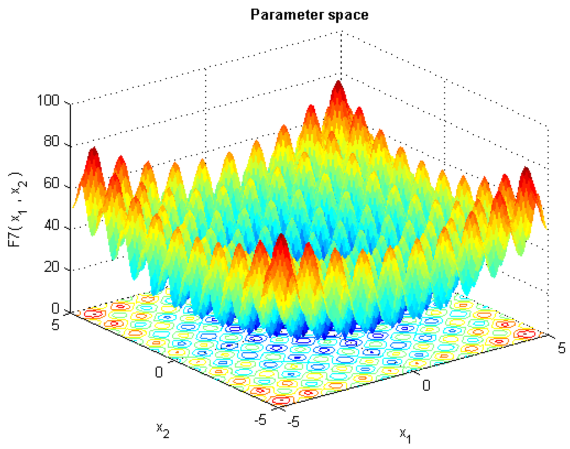


(e) F5

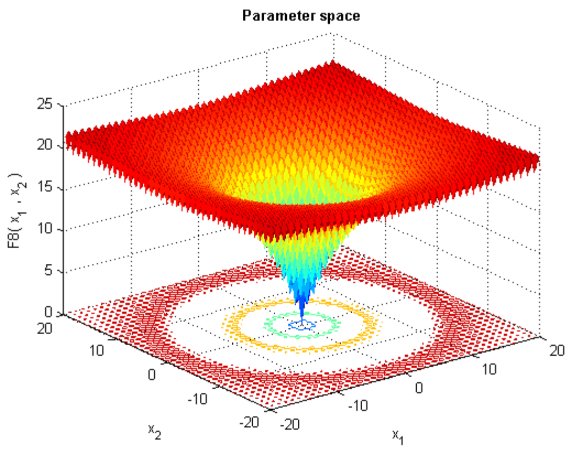
Figure 7. NSSA trajectory in 3-dimensional unimodal test functions



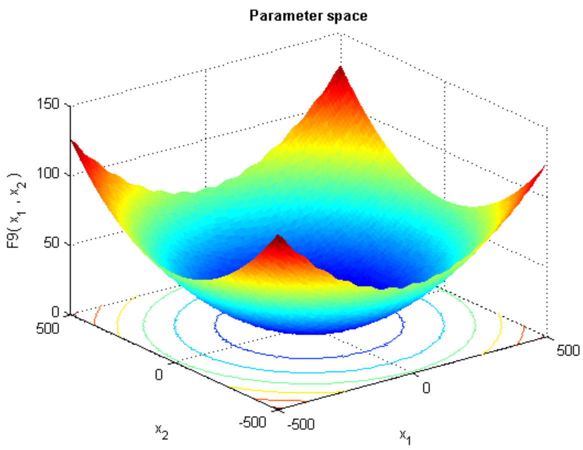
(a) F6



(b) F7

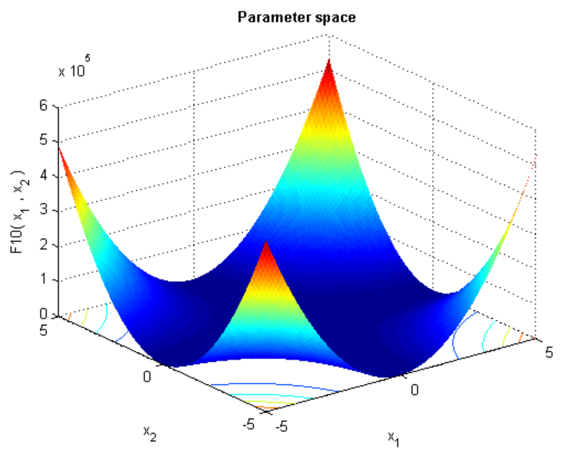


(c) F8

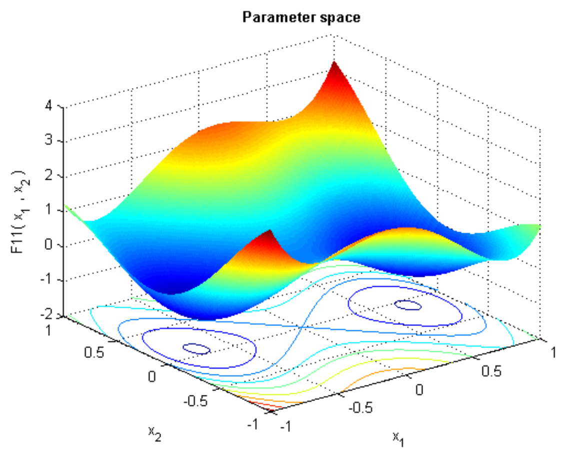


(d) F9

Figure 8. NSSA trajectory in 3-dimensional multimodal test functions

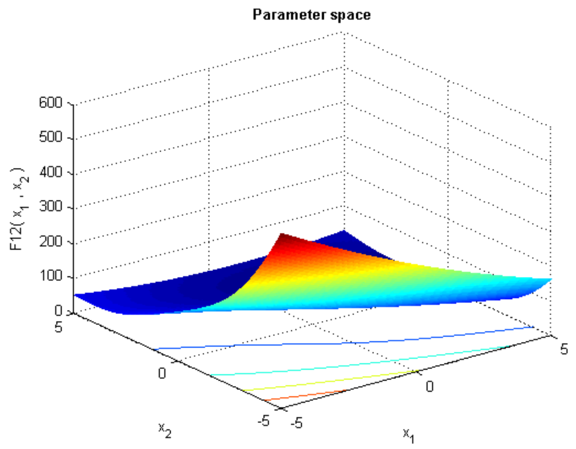


(a) F10

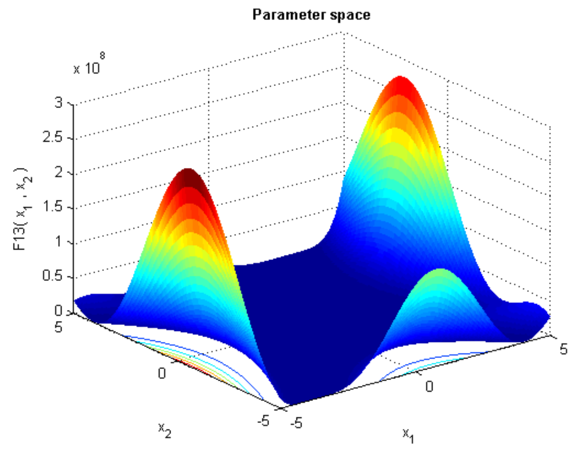


(b) F11

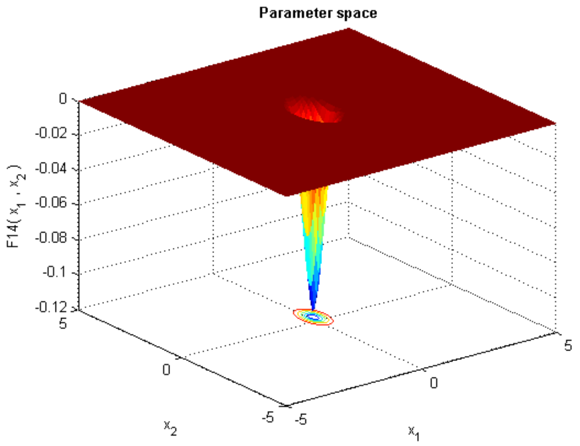
Contd. Figure



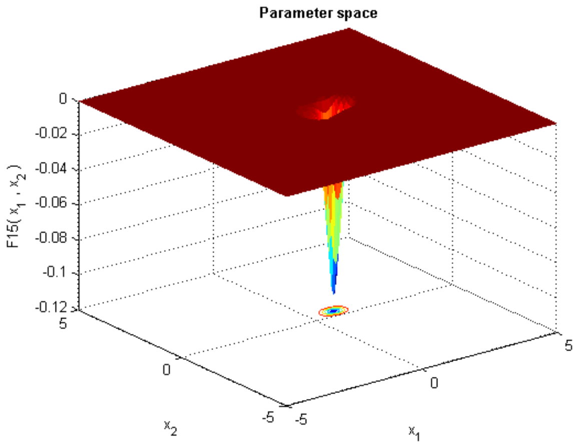
(c) F12



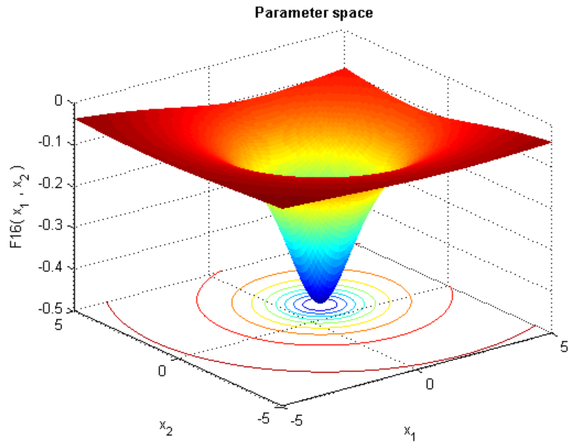
(d) F13



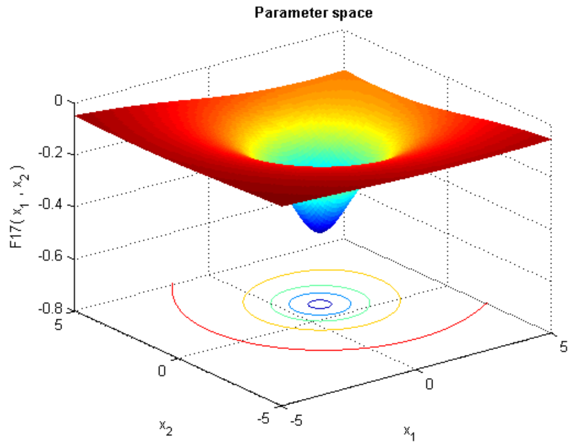
(e) F14



(f) F15

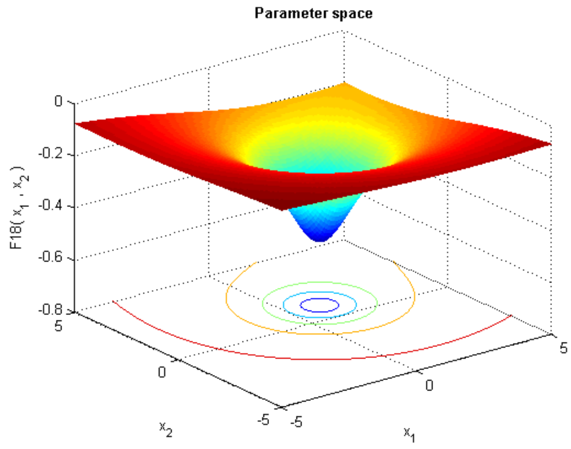


(g) F16



(h) F17

Contd. Figure



(i) F18

Figure 9. NSSA trajectory in 3-dimensional fixed-dimension test functions

The results indicate that NSSA, when compared to alternative optimization algorithms, consistently secured the lowest rank across the majority of benchmark functions. This underscores NSSA's remarkable efficacy in contrast to its counterparts. Notably, SSA, PSO, and GWO closely followed NSSA, occupying the second, third, and fourth positions, respectively. It's important to note that NSSA's widespread success shouldn't be misconstrued as a universal superiority over all existing optimization techniques within the literature, as doing so might infringe upon the principles of the *free lunch theorem* (Wolpert and Macready [33]). Rather, its exceptional performance indicates its prominence solely in the context of the algorithms considered within this study.

Table 7. Pair-wise wilcoxon signed rank test results

Function	Wilcoxon signed rank test order
F1	**NSSA** = SSA < GWO < PSO < GSA
F2	**NSSA** = SSA < GWO < PSO < GSA
F3	**NSSA** = SSA < GWO< GSA < PSO
F4	**NSSA** < SSA < PSO < GWO < GSA
F5	**NSSA** < SSA < GWO < GSA < PSO
F6	**NSSA** < PSO < SSA < GWO < GSA
F7	**NSSA** = SSA = GWO < GSA < PSO
F8	**NSSA** = SSA < GWO < PSO < GSA
F9	**NSSA** = SSA = GWO = PSO < GSA
F10	**NSSA** = SSA = GWO = PSO < GSA
F11	**NSSA** = SSA = GWO = PSO = GSA
F12	**NSSA** = SSA = GWO = PSO = GSA
F13	**NSSA** = SSA = GWO = PSO = GSA
F14	**NSSA** = SSA = PSO = GSA < GWO
F15	**NSSA** = SSA = GWO = PSO = GSA
F16	**NSSA** = SSA = GWO = PSO = GSA
F17	**NSSA** = SSA = PSO = GSA < GWO
F18	**NSSA** = SSA = PSO = GSA < GWO

Table 8. Rank summary of statistical assessment results

Function	NSSA	SSA	GWO	PSO	GSA
F1	1.5	1.5	3	4	5
F2	1.5	1.5	3	4	5
F3	1.5	1.5	3	5	4
F4	1	2	4	3	5
F5	1	2	3	5	4
F6	1	3	4	2	5
F7	2	2	2	5	4
F8	1.5	1.5	3	4	5
F9	2.5	2.5	2.5	2.5	5
F10	2.5	2.5	2.5	2.5	5
F11	3	3	3	3	3
F12	3	3	3	3	3
F13	3	3	3	3	3
F14	2.5	2.5	5	2.5	2.5
F15	3	3	3	3	3
F16	3	3	3	3	3
F17	2.5	2.5	5	2.5	2.5
F18	2.5	2.5	5	2.5	2.5
Total	**38.5**	42.5	60	59.5	69.5

8. Application of NSSA on Engineering Design Problems

8.1 Himmelblau's Nonlinear Optimization Problem

A well-known optimization problem for evaluating the effectiveness of optimization algorithms is Himmelblau's nonlinear optimization ([7]) problem. It bears David Himmelblau's name, who raised the issue in 1972. The formal explanation for the problem comes next.

$$\text{Minimize: } f(\mathbf{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.29329x_1 - 40792.141$$

$$\text{subject to: } g_1(\mathbf{x}) = 85.334407 + 0.0056858x_2x_5 + 0.00026x_1x_4 - 0.0022053x_3x_5$$

$$g_2(\mathbf{x}) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2$$

$$g_3(\mathbf{x}) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4$$

$$0 \le g_1(\mathbf{x}) \le 92, 90 \le g_2(\mathbf{x}) \le 110, 20 \le g_3(\mathbf{x}) \le 25$$

$$78 \le x_1 \le 102,\ 33 \le x_2 \le 45,\ 27 \le x_3,\ x_4, x_5 \le 45$$

For this Himmelblau's nonlinear optimization problem the number of maximum iterations are 100. For generating statistical results, we independently run 30 times. Table 9 shows the optimization results for this problem by using NSSA. On the other hand, Table 10 depicts the value of decision variables and constraints. From Tables 9 and 10, it can be clearly seen that our solution is best with the better objective function value but constraint g_3 is violated.

Table 9. NSSA results for Himmelblau's nonlinear optimization problem

Best	Mean	Std	Number of sparrows
−32217.431	−32217.431	0.0	50

Table 10. The value of decision variables and constraints

Variables	Value	Constraints	Value
x_1	78	g_1	90.1116
x_2	33	g_2	96.1674
x_3, x_4, x_5	27, 27, 27	g_3	16.7629

8.2 Speed Reducer Design Optimization Problem

The design of the speed reducer (Golinski [6]) displayed in Figure 10, is opted with the face width x_1, module of teeth x_2, number of teeth on pinion x_3, length of the first shaft between bearings x_4, length of the second shaft between bearings x_5, diameter of the first shaft x_6, and diameter of the first shaft x_7 (all variables continuous except x_3 that is integer).

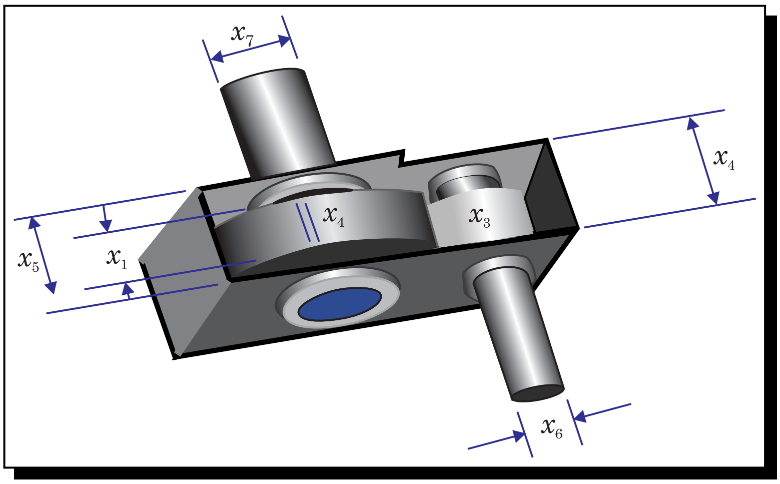


Figure 10. Speed reducer design optimization problem (Aktemur and Gusseinov [2])

The weight of the speed reducer is to be minimized subject to constraints on bending stress of the gear teeth, surface stress, transverse deflections of the shafts and stresses in the shaft. The problem is:

$$\text{Minimize: } f(\vec{x}) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934)$$
$$- 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2)$$

$$\text{subject to: } g_1(\vec{x}) = \frac{27}{x_1x_2^2x_3} - 1 \le 0, \quad g_2(\vec{x}) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \le 0$$

$$g_3(\vec{x}) = \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \le 0, \quad g_4(\vec{x}) = \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \le 0$$

$$g_5(\vec{x}) = \frac{1.0}{110x_6^3}\sqrt{\left(\frac{745.0x_4}{x_2x_3}\right)^2 + 16.9\times 10^6} - 1 \le 0$$

$$g_6(\vec{x}) = \frac{1.0}{85x_7^3}\sqrt{\left(\frac{745.0x_5}{x_2x_3}\right)^2 + 157.5\times 10^6} - 1 \le 0$$

$$g_7(\vec{x}) = \frac{x_2x_3}{40} - 1 \le 0, \quad g_8(\vec{x}) = \frac{5x_2}{x_1} - 1 \le 0$$

$$g_9(\vec{x}) = \frac{x_1}{12x_2} - 1 \le 0, \quad g_{10}(\vec{x}) = \frac{1.5x_6 + 1.9}{x_4} - 1 \le 0$$

$$g_{11}(\vec{x}) = \frac{1.1x_7 + 1.9}{x_5} - 1 \le 0, \quad \text{with } 2.6 \le x_1 \le 3.6,\ 0.7 \le x_2 \le 0.8,$$

$$17 \le x_3 \le 28,\ 7.3 \le x_4,\ x_5 \le 8.3,\ 2.9 \le x_6 \le 3.9 \text{ and } 5.0 \le x_7 \le 5.5.$$

For this, speed reducer optimization problem the number of maximum iterations are 100. For evaluating statistical results, we independently run 30 times. Table 11 shows the optimization results for this problem by using NSSA. On the other hand, Table 12 represents the value of decision variables and constraints. From Tables 11 and 12, it is crystal clearly that our solution is best with optimal value of the objective function but g_3 constraint is violated.

Table 11. NSSA results for Speed reducer design optimization problem

Best	Mean	Std	Number of sparrows
2362.267	2362.267	0.0	50

Table 12. The value of decision variables and constraints

Variables	Value	Constraints	Value
x_1	2.60	g_1	0.24665
x_2	0.70	g_2	0.079617
x_3	17.00	g_3	5.5119
x_4	7.30	g_4	−0.87686
x_5	7.80	g_5	0.54179
x_6	2.90	g_6	0.18206
x_7	5.00	g_7	−0.7025
–	–	g_8	0.34615
–	–	g_9	−0.69048
–	–	g_{10}	−0.14384
–	–	g_{11}	−0.051282

9. Conclusion

SSA is a metaheuristic swarm optimization method that was developed to find the best possible solution in every possible way. However, some researchers have discovered a few flaws in it, such as slow convergence and lower optimality. We have suggested a *Niching Sparrow Search*

Algorithm (NSSA) in this article. And using 18 various benchmark test functions on NSSA, SSA, GWO, PSO, and GSA we conducted experiments using MATLAB. According to the findings, NSSA performs remarkably well in all areas, including optimality, convergence precision, stability, and convergence rate. We applied the Wilcoxon signed rank test to NSSA, SSA, GWO, GSA, and PSO. NSSA received the smallest ranking, indicating that its performance is better to that of the other four algorithms. Moreover, NSSA applied on two engineering design problems namely Himmelblau's nonlinear optimization problem and Speed reducer design optimization problem. For these two problems NSSA gave the best performance but in each of these two problems g_3 constraint is violated.

So, we would continue to conduct in-depth analysis and research on the NSSA in our ongoing research to deal with the problem of voilated constraints. Additionally, we would attempt to apply NSSA algorithm to more challenging real-world engineering issues, such as the *Travelling Salesman Problem* (TSP), the challenge of Welded beam design optimization problem, etc. In addition, we would expand the NSSA to address the multiobjective optimization issue.

Competing Interests

The authors declare that they have no competing interests.

Authors' Contributions

All the authors contributed significantly in writing this article. The authors read and approved the final manuscript.

References

[1] R. Ahmed, A. Nazir, S. Mahadzir, M. Shorfuzzaman and J. Islam, Niching grey wolf optimizer for multimodal optimization problems, *Applied Sciences* **11**(11) (2021), 4795, DOI: 10.3390/app11114795.

[2] C. Aktemur and I. Gusseinov, A comparison of sequential quadratic programming, genetic algorithm, simulated annealing, particle swarm optimization and hybrid algorithm for the design and optimization of Golinski's speed reducer, *International Journal of Energy Applications and Technologies* **4**(2) (2017), 34 – 52.

[3] D. W. Boeringer and D. H. Werner, Efficiency-constrained particle swarm optimization of a modified bernstein polynomial for conformal array excitation amplitude synthesis, *IEEE Transactions on Antennas and Propagation* **53**(8) (2005), 2662 – 2673, DOI: 10.1109/TAP.2005.851783.

[4] S. Desale, A. Rasool, S. Andhale and P. Rane, Heuristic and metaheuristic algorithms and their relevance to the real world: a survey, *International Journal of Computer Engineering in Research Trends* **2**(5) (2015), 296 – 304.

[5] F. S. Gharehchopogh, M. Namazi, L. Ebrahimi and B. Abdollahzadeh, Advances in sparrow search algorithm: a comprehensive survey, *Archives of Computational Methods in Engineering* **30** (2023), 427 – 455, DOI: 10.1007/s11831-022-09804-w.

[6] J. Golinski, An adaptive optimization system applied to machine synthesis, *Mechanism and Machine Theory* **8**(4) (1973), 419 – 436, DOI: 10.1016/0094-114X(73)90018-9.

[7] D. M. Himmelblau, *Applied Nonlinear Programming*, 1st edition, McGraw-Hill, 498 pages (1972).

[8] Z. Huang, D. Zhu, Y. Liu and X. Wang, Multi-strategy sparrow search algorithm with non-uniform mutation, *Systems Science & Control Engineering* **10**(1) (2022), 936 – 954, DOI: 10.1080/21642583.2022.2140723.

[9] D. Karaboga, *An Idea Based on Honey Bee Swarm for Numerical Optimization*, Technical Report-TR06, Department of Computer Engineering, Engineering Faculty, Erciyes University, Turkey (2005).

[10] K. K. Kumar and G. N. Reddy, The sparrow search algorithm for optimum position of wind turbine on a wind farm, *International Journal of Renewable Energy Research* **11**(4) (2021), 1939 — 1948, DOI: 10.20508/ijrer.v11i4.12345.g8346.

[11] Y. Lei, G. De and L. Fei, Improved sparrow search algorithm based DV-Hop localization in WSN, *2020 Chinese Automation Congress* (CAC) (2020), pp. 2240 – 2244, IEEE (2020), DOI: 10.1109/CAC51589.2020.9327429.

[12] J. Li, Robot path planning based on improved sparrow algorithm, *Journal of Physics: Conference Series* **1861** (2021), 012017, DOI: 10.1088/1742-6596/1861/1/012017.

[13] J. J. Liang, B. Y. Qu, S. T. Ma and P. N. Suganthan, Memetic fitness euclidean-distance particle swarm optimization for multi-modal optimization, in: *Bio-Inspired Computing and Applications*, Lecture Notes in Computer Science Vol. **6840**, Springer, Berlin — Heidelberg (2012), DOI: 10.1007/978-3-642-24553-4_50.

[14] J. Ma, Z. Hao and W. Sun, Enhancing sparrow search algorithm via multi-strategies for continuous optimization problems, *Information Processing & Management* **59**(2) (2022), 102854, DOI: 10.1016/j.ipm.2021.102854.

[15] S. Mirjalili, S. M. Mirjalili and A. Lewis, Grey wolf optimizer, *Advances in Engineering Software* **69** (2014), 46 – 61, DOI: 10.1016/j.advengsoft.2013.12.007.

[16] U. Mlakar, Hybrid cuckoo search for constraint engineering design optimization problems, in: *Proceedings of the 3rd Student Computer Science Research Conference*, Ljubljana, Slovenia, 2016, pp. 57 – 60, (2016).,

[17] R. Y. M. Nakamura, L. A. M. Pereira, K. A. Costa, D. Rodrigues, J. P. Papa and X. S. Yang, BBA: a binary bat algorithm for feature selection, in: *Proceedings of 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images*, Ouro Preto, Brazil, 2012, pp. 291 – 297, IEEE (2012), DOI: 10.1109/SIBGRAPI.2012.47.

[18] C. Ouyang, D. Zhu and F. Wang, A learning sparrow search algorithm, *Computational Intelligence and Neuroscience* **2021** (2021), Article ID 3946958, 23 pages, DOI: 10.1155/2021/3946958.

[19] Y. Peng, Y. Liu and Q. Li, The application of improved sparrow search algorithm in sensor networks coverage optimization of bridge monitoring, in: *Machine Learning and Artificial Intelligence* (Ebook), A. J. Tallón-Ballesteros and C.-H. Chen (editors), Frontiers in Artificial Intelligence and Applications series, pp. 416 – 423, IOS Press (2020), DOI: 10.3233/FAIA200808.

[20] T. Peram, K. Veeramachaneni and C. K. Mohan, Fitness-distance-ratio based particle swarm optimization, in: *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No.03EX706), Indianapolis, IN, USA, 2003*, 174 – 181, IEEE, (2003), DOI: 10.1109/SIS.2003.1202264.

[21] B. Y. Qu, J. J. Liang and P. N. Suganthan, Niching particle swarm optimization with local search for multi-modal optimization, *Information Sciences* **197** (2012), 131 – 143, DOI: 10.1016/j.ins.2012.02.011.

[22] R. V. Rao, V. J. Savsani and D. P. Vakharia, Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems, *Computer-Aided Design* **43**(3) (2011), 303 – 315, DOI: 10.1016/j.cad.2010.12.015.

[23] J. Ren, H. Wei, Y. Yuan, X. Li, F. Luo and Z. Wu, Boosting sparrow search algorithm for multi-strategy-assist engineering optimization problems, *AIP Advances* **12** (2022), 095201, DOI: 10.1063/5.0108340.

[24] Y. M. Roopa, SPARROW algorithm for clustering software components, *International Journal of Engineering Research and Development* **10**(6) (2014), 20 – 24.

[25] N. Singh and S. B. Singh, Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance, *Journal of Applied Mathematics* **2017** (2017), Article ID 2030489, 15 pages, DOI: 10.1155/2017/2030489.

[26] W. Song, S. Liu, X. Wang and W. Wu, An improved sparrow search algorithm, in: *2020 IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA / BDCloud / SocialCom / SustainCom)*, Exeter, United Kingdom, pp. 537 – 543, IEEE (2020), DOI: 10.1109/ISPA-BDCloud-SocialCom-SustainCom51426.2020.00093.

[27] W. Sun, M. Tang, L. Zhang, Z. Huo and L. Shu, A survey of using swarm intelligence algorithms in IoT, *Sensors* **20**(5) (2020), 1420, DOI: 10.3390/s20051420.

[28] A. Tang, H. Zhou, T. Han and L. Xie, A chaos sparrow search algorithm with logarithmic spiral and adaptive step for engineering problems, *Computer Modeling in Engineering & Sciences* **130**(1) (2022), 331 – 364, DOI: 10.32604/cmes.2021.017310.

[29] G. Venter, *Review of Optimization Techniques*, John Wiley & Sons Ltd., (2010), DOI: 10.1002/9780470686652.eae495.

[30] Z. Wang, G. Sun, K. Zhou and L. Zhu, A parallel particle swarm optimization and enhanced sparrow search algorithm for unmanned aerial vehicle path planning, *Heliyon* **9**(1) (2023), E14784, DOI: 10.1016/j.heliyon.2023.e14784.

[31] Z. Wang, X. Huang and D. Zhu, A multistrategy-integrated learning sparrow search algorithm and optimization of engineering problems, *Computational Intelligence and Neuroscience* **2022** (2022), Article ID 2475460, 21 pages, DOI: 10.1155/2022/2475460.

[32] F. Wilcoxon, S. K. Katti and R. A. Wilcox, *Critical Values and Probability Levels for the Wilcoxon Rank Sum Test and the Wilcoxon Signed Rank Test*, American Cyanamid, 128 pages (1963).

[33] D. H. Wolpert and W. G. Macready, No free lunch theorems for optimization, *IEEE Transactions on Evolutionary Computation* **1**(1) (1997), 67 – 82, DOI: 10.1109/4235.585893.

[34] C. Wu, X. Fu, J. Pei and Z. Dong, A novel sparrow search algorithm for the traveling salesman problem, *IEEE Access* **9** (2021), 153456 – 153471, DOI: 10.1109/ACCESS.2021.3128433.

[35] S. Xie, S. He and J. Cheng, Research on improved sparrow algorithm based on random walk, *Journal of Physics: Conference Series* **2254** (2022), 012051, DOI: 10.1088/1742-6596/2254/1/012051.

[36] J. Xue and B. Shen, A novel swarm intelligence optimization approach: sparrow search algorithm, *Systems Science & Control Engineering* **8**(1) (2020), 22 – 34, DOI: 10.1080/21642583.2019.1708830.

[37] Q. Yang, Y. Gao and Y. Song, A tent Lévy flying sparrow search algorithm for wrapper-based feature selection: a COVID-19 case study, *Symmetry* **15**(2) (2022), 316, DOI: 10.3390/sym15020316.

[38] X. Yang, J. Liu, Y. Liu, P. Xu, L. Yu, L. Zhu, H. Chen and W. Deng, A novel adaptive sparrow search algorithm based on chaotic mapping and t-distribution mutation, *Applied Sciences* **11**(23) (2021), 11192, DOI: 10.3390/app112311192.

[39] L. Yang, Z. Li, D. Wang, H. Miao and Z. Wang, Software defects prediction based on hybrid particle swarm optimization and sparrow search algorithm, *IEEE Access* **9** (2021), 60865 – 60879, DOI: 10.1109/ACCESS.2021.3072993.