Review Article

Social Spider Optimization Algorithm: Modifications, Applications, and Perspectives

**Alberto Luque-Chang, Erik Cuevas ⓘ, Fernando Fausto,
Daniel Zaldívar ⓘ, and Marco Pérez ⓘ**

Departamento de Electrónica, Universidad de Guadalajara, CUCEI, Av. Revolución 1500, C.P. 44430, Guadalajara, Jal, Mexico

Correspondence should be addressed to Erik Cuevas; erik.cuevas@cucei.udg.mx

Swarm intelligence (SI) is a research field which has recently attracted the attention of several scientific communities. An SI approach tries to characterize the collective behavior of animal or insect groups to build a search strategy. These methods consider biological systems, which can be modeled as optimization processes to a certain extent. The Social Spider Optimization (SSO) is a novel swarm algorithm that is based on the cooperative characteristics of the social spider. In SSO, search agents represent a set of spiders which collectively move according to the biological behavior of the colony. In most of SI algorithms, all individuals are modeled considering the same properties and behavior. In contrast, SSO defines two different search agents: male and female. Therefore, according to the gender, each individual is conducted by using a different evolutionary operation which emulates its biological role in the colony. This individual categorization allows reducing critical flaws present in several SI approaches such as incorrect exploration-exploitation balance and premature convergence. After its introduction, SSO has been modified and applied in several engineering domains. In this paper, the state of the art, improvements, and applications of the SSO are reviewed.

1. Introduction

Swarm intelligence (SI) algorithms have attracted the attention of researchers due to their powerful and efficient performance for solving optimization problems. These algorithms are developed by the combination between randomness and deterministic rules, emulating animal groups in the nature swarms [1].

Insects and animal groups provide a rich set of metaphors to develop new swarm algorithms; these entities are complex systems with individuals that reproduce different behaviors depending on its biological role such as gender, type, or size [2]. In spite of this, most of SI algorithms model generic individuals with the same simple rule of behavior. Under such conditions, it is not possible to include new operators that reproduce complex biological behaviors such as task-division and gender responsibility.

The Social Spider Optimization (SSO) [3] proposed by Erik Cuevas et al., in 2013, is a population-based algorithm that simulates the cooperative behavior of the social spider.

SSO considers two search agents (spiders): male and female. Each individual is conducted with a different set of evolutionary operators depending on it gender which mimics different cooperative behaviors typically found in the colony. This individual categorization allows reducing critical flaws present in several SI approaches such as incorrect exploration-exploitation balance and premature convergence. Unlike other popular algorithms as Particle Swarm Optimization (PSO) [4], Genetic Algorithm (GA) [5], Cuckoo Search (CS) [6], Artificial Bee Colony (ABC) [7], Harmony Search (HS) [8], and Social Network Optimization (SNO) [9], it evades the concentration of particles in the best positions, preventing critical faults as premature convergence to suboptimal solutions or a limited balance between exploration and exploitation. These characteristics have motivated the use of the SSO algorithm to solve a wide variety of engineering applications in diverse areas including machine learning: Artificial Neural Networks Training [10, 11] and Support Vector Machine Parameter Tuning [12, 13]; control: Fractional Controller Design [14–16] and frequency controllers [17];
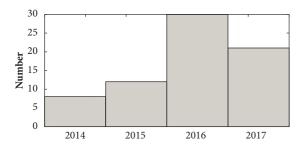
FIGURE 1: Number of JCR indexed publications about SSO since 2014.

image processing: Image Multilevel Thresholding [18], Image Contrast Enhancement [19], and Image Template Matching [20], and energy: Distribution of Renewable Energy [21], Congestion Management [22], and Anti-Islanding Protection [23].

After its introduction, SSO has been extensively modified and applied in several engineering domains. This paper presents a review of the SSO algorithm, its improvements, adaptations, and applications. The rest of the document is organized as follows: Section 2 presents an explanation of the original SSO algorithm and its operators, Section 3 presents a brief list of modifications of the algorithm doing enfaces in the principal modification in each case, Section 4 presents a list of applications and problems solved using the SSO or some of its modifications, Section 5 presents future work to realize with the algorithm and finally Section 6 presents the conclusions.

2. Social Spider Optimization (SSO)

2.1. Statistics and Publications of the SSO. Since the introduction of the Social Spider Optimization (SSO) method in 2013, it has attracted the attention of the metaheuristic community. Currently, there exist more than 120 documents that consider modification, improvement, or application of SSO in any domain.

To the best of our knowledge, in total, there are 120 documents which consider the modification and applications of the SSO method. From them, there are 71 papers which were published on journals with JCR index, such as Expert Systems with applications, Neural Computing and Applications, IEEE Transactions on Magnetics, Memetic Computing, Applied Soft Computing, Soft Computing, Swarm and Evolutionary Computation, and Engineering Applications of Artificial Intelligence. Figure 1 shows the number of Journal Citations Reports (JCR) indexed documents since 2014. In addition, there are 49 articles published on conferences, such as IEEE Symposium on Swarm Intelligence (SIS), IEEE Congress on Evolutionary Computation (CEC), the Conference Companion on Genetic and Evolutionary Computation (GECCO), International Conference on Advanced Computational Intelligence (ICACI), International Conference on Swarm Intelligence (ICSI), just to name a few.

From Figure 1, it is clear that the SSO method has attracted the attention of more and more researchers in

studies and applications. Under such conditions, the SSO algorithm has been modified and applied in several domains.

2.2. Original SSO Algorithm. The SSO assumes the search space as a communal spider web and each candidate solution in the population represents a spider. Each spider receives a weight according to the fitness value of the solution that it symbolizes. The approach models two different search set of evolutionary operators that simulate the different cooperative behaviors assumed in the colony.

The algorithm was designed to solve a nonlinear global optimization problem with box constraint in the form:

$$
\begin{aligned}
&minimize: \quad f(x) \quad x = \left(x^1, x^2, \ldots, x^d\right) \in \mathbb{R}^d; \\
&subject\ to \quad x \in X
\end{aligned}
\tag{1}
$$

where $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ is a nonlinear function and $X = \{x \in \mathbb{R}^d \mid l_h \leq x \leq u_h, h = 1, \ldots, d\}$ is a reduced feasible space limited by the lower (l_h) and upper (u_h) limits.

The SSO uses a population S of N candidate solutions to solve the optimization problem. Each solution represents a spider position whereas the general web represents the search space X. In the approach, the population S is divided into two search agents: Male (Ms) and Female (Fs). With the aim of simulating a real spider colony, the number N_f is of females, and Nf is randomly selected in a range between the 65–95% of the entire population S, whereas the rest Nm is considered as the male individuals ($Nm = S - Nf$). Under this conditions, the group Fs forms a set of female individuals ($Fs = \{fs_1, fs_{2,\ldots,} fs_{Nf}\}$), hence the Ms group, the male individuals, ($Ms = \{ms_1, ms_{2,\ldots,} ms_{nm}\}$), where $S = Fs \cup Ms$ ($S = \{s_1, s_2, \ldots, s_N\}$).

In the algorithm each spider i has a weight we_i according to its solution fitness, and that weight is calculated as follows:

$$
we_i = \frac{fit_i - worst}{best - worst}
\tag{2}
$$

where fit_i is the fitness of the i-th spider position, $i \in 1,..,N$, and $best$ and $worst$ are, respectively, the best fitness value and the worst fitness value of whole population S. The information exchange is the principal mechanism of the SSO in the optimization process. It is simulated through vibrations produced in the web. The vibration that a spider i perceives from a spider j is modeled as follows:

$$
V_{i,j} = we_j e^{d_{i,j}^2}
\tag{3}
$$

where we_j is the weight of the j-th spider and d is the distance between the two spiders. Each spider i can perceive only three types of vibration, $v_{i,n}$, $v_{i,b}$, and $v_{i,f}$. $v_{i,n}$ is the vibration produced by the nearest spider n with a higher weight with regard to ($w_n > w_i$). $v_{i,f}$ is produced by the nearest female spider and it is only applicable if i is a male spider and finally $v_{i,b}$ is produced by the best spider in the population S.

In the algorithm, a population N of spiders is operated since an initial stage k=0 to a determinate number it of iterations ($k=it$). Depending on its gender each individual is

conducted by a different set of evolutionary operators. In the case of female spiders, the new position fs_i^{k+1} is obtained by modifying the current spider position fs_i^k. The modification

$$fs_i^{k+1} = \begin{cases} fs_i^k + \alpha \cdot V_{i,n} \cdot \left(s_n - fs_i^k\right) + \beta \cdot V_{i,b} \cdot \left(s_b - fs_i^k\right) + \delta \cdot \left(\text{rand} - \dfrac{1}{2}\right) & \text{with probability } Pf \\ fs_i^k - \alpha \cdot V_{i,n} \cdot \left(s_n - fs_i^k\right) - \beta \cdot V_{i,b} \cdot \left(s_b - fs_i^k\right) + \delta \cdot \left(\text{rand} - \dfrac{1}{2}\right) & \text{with probability } 1 - Pf \end{cases} \tag{4}$$

where α, β, δ, and rand are random numbers between $[0, 1]$ and k is the iteration number and the individuals s_n and s_b symbolize the nearest spider with a higher weight than fs_i^k and the best spider in the communal web, respectively.

On the other hand, male spiders are classified into two types: dominant (D) and nondominant (ND). The D group is integrated with the male spider whose fitness values are better with regard to the complete male set. Thus, the ND group is formed by the rest of the male spiders. In the optimization process the male spiders ms_i^k are operated by the following model:

$$ms_i^{k+1}$$

$$= \begin{cases} ms_i^k + \alpha \cdot V_{i,f} \cdot \left(s_f - ms_i^k\right) + \delta \cdot \left(\text{rand} - \dfrac{1}{2}\right) & \text{if } ms_i^k \epsilon\ D \\ ms_i^k + \alpha \cdot \left(\dfrac{\sum_{h\epsilon ND} ms_h^k \cdot w_h}{\sum_{h\epsilon ND} w_h} - ms_i^k\right) & \text{if } ms_i^k \epsilon\ ND \end{cases} \tag{5}$$

where α, δ, and rand are random numbers between $[0, 1]$ and s_f is the nearest female spider to de male individual i.

In the SSO mating operator is performed between dominant male m_d and female individuals with a specific range r; this generates a new individual s_{new}. The weight of each spider defines the probabilities of the influence of each spider on s_{new}; the heavier element has more probability to influence the new individual s_{new}. Once a new spider is generated, it is compared with the rest of the population, and if the new spider has a better fitness than the worst spider member of the population, the worst spider is replaced with s_{new}; otherwise s_{new} is discarded. Figure 2 illustrates the complete evolutionary process in the form of flow chart.

2.3. Performance Comparison. To prove the performance of the SSO algorithm, it was applied to a set of benchmark functions. It was compared to the results given by two popular algorithms in the literature: the PSO and the ABC algorithms. In the analysis, a representative set of functions have been considered. They are shown in Table 2. The population for all the algorithms was set in 50, and while the stop criteria were fixed in 1000 iterations, the parameters calibration for each algorithm in the comparison was set as follows:

(1) ABC the parameter limit was set in 100 according to its own reference guideline [7].

(2) PSO the weight factor decreases linearly from 0.9 to 0.2, and the parameters $c_1=2$ and $c_2=2$ [4].

(3) SSO the probability factor Pf is set in 0.7.

is randomly controlled by a probability factor Pf and the movement is produced in relation to other spiders and its vibrations transmitted through the search space:

The performance experiment considers the results of 30 runs and takes as performance measure the following indexes: the Average Best Fitness (AB), the Median Best fitness (Med), and the Standard Deviation (STD) of the best solution. The results of the experiment presented in Table 1 show that the SSO outperforms the other algorithms in all the test functions. That is due to a good balance between exploration and exploitation.

The analysis of the final fitness values cannot absolutely characterize the capacities of an optimization algorithm. Consequently, a convergence experiment on the six functions from Table 2 has been conducted. The objective of this test is to evaluate the velocity with which each optimization scheme reaches the optimum. In the experiment, the performance of every algorithm PSO, ABC, and SSO is analyzed. In the test, all functions are operated considering 30 dimensions. In order to build the convergence graphs, we employ the raw data generated in the simulations. As each function is executed 30 times for each algorithm, we select the convergence data of the run which represents the median test from the 30 executions. Figure 3 presents the convergence tables for PSO, ABC, and SSO where it proved that the SSO convergence rate is faster than the other two algorithms finding the best solution in less than 200 iterations.

3. Modifications and Improvements

In this section, the modifications and improvements of the SSO algorithm are reviewed. In some cases, they are grouped when the methods maintain similar characteristics.

3.1. Social Spider Optimization Based on Rough Sets (SSORS). In 2016, Mohamed Abd El Aziz et al. [24] proposed a modification of SSO by using the rough set theory for the evaluation of the fitness function. The approach is applied to solve the minimum attribute reduction problem. In the algorithm, the fitness function depends on the rough sets dependency degree and takes into consideration the number of selected features.

The algorithm starts by randomly initializing a population of spiders. Next, each individual is converted into a binary vector of length N by using the following equations:

$$FP = \frac{1}{1 + e^{-x_i^j(t)}} \tag{6}$$

$$x_i^j(t+1) = \begin{cases} 0 & \text{if } FP\left(x_i^j(t)\right) > \epsilon \\ 1 & \text{for other cases} \end{cases} \tag{7}$$
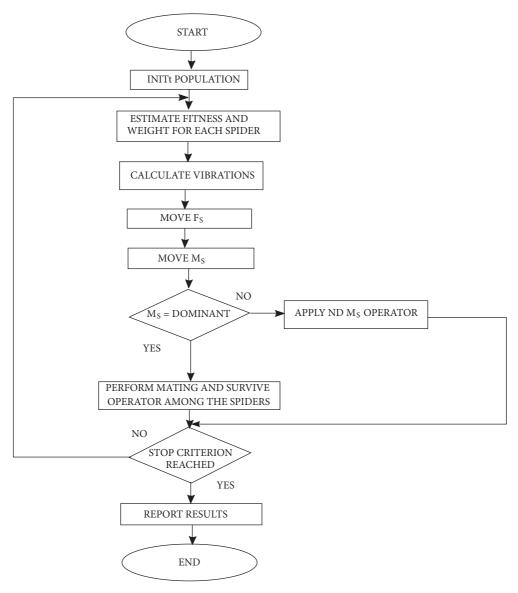
FIGURE 2: SSO flowchart.

where $x_i{}^j(t)$ is the spider value at the iteration t and $\epsilon \in [0, 1]$. The algorithm uses the dependency degree given by

$$\gamma_c(D) = \frac{|POS_C(D)|}{|U|} \quad A = C \cup D \qquad (8)$$

where C and D are called condition and decision features and $POS_C(D)$ is the positive region that contains all the objects of U that can be classified in classes of U/D using information of C, which is used to evaluate the performance of the solutions. Furthermore, the fitness function is defined by

$$F(R) = \rho\gamma_R(D) + (1 - \rho)\left(1 - \frac{|R|}{|C|}\right) \qquad (9)$$

where $\rho \in [0, 1]$ is a parameter that gives balance between number of selected features and the classification quality and $|\cdot|$ is the length of the feature set. The fitness of each spider

is compared with the global best (F_{best}), and if it has a better fitness value, then F_{best} is replaced with the current spider and its position becomes the reduct set R. The process is repeated until a stop criterion is satisfied. Figure 4 shows the *SSORS* computational procedure in form of a flowchart.

The SSORS was compared with other similar schemes to solve a feature selection problem known as minimum attribute selection. They include approaches such as Particle Swarm Optimization with rough sets (PSORS), Harmony Search with rough sets (HSRS), Artificial Bee Colony with rough sets (ABCRS), FireFly Algorithm with rough sets (FARS), the rough set method itself (RS), and Algorithm Fish Swarm with rough sets (AFSRS) and test them in eight classifiers including Random Forest (RF), Support Vector Machines (SVM), Logistic Model Tree (LMT), Logistic Regression (Log), J48 decision tree (J48), and Naïve Bayes (NB). In the analysis, the algorithms were tested in six medical data sets available in the UCI machine learning

TABLE 1: Minimization results of benchmark functions of Table 2 with n = 30 maximum.

		SSO	ABC	PSO
$f_1(x)$	**AB**	**1.96E-03**	2.90E-03	1.00E+03
	Med	2.81E-03	1.50E-03	**2.08E-09**
	STD	**9.96E-04**	1.44E-03	3.05E+03
$f_2(x)$	**AB**	**4.27E-02**	1.13E+00	8.63E+04
	Med	**3.49E-02**	6.11E-01	8.00E+04
	STD	**3.11E-02**	1.57E+00	5.56E+04
$f_3(x)$	**AB**	**1.14E+02**	1.38E+02	3.34E+04
	Med	**5.86E+01**	1.32E+02	4.03E+02
	STD	**3.90E+01**	1.55E+02	4.38E+04
$f_4(x)$	**AB**	**4.44E-04**	1.10E-01	6.93E+02
	Med	**4.05E-04**	4.97E-02	5.50E+02
	STD	**2.90E-04**	1.98E-01	6.48E+02
$f_5(x)$	**AB**	**8.59E+00**	2.64E+01	1.35E+02
	Med	**8.78E+00**	2.24E+01	1.36E+02
	STD	**1.1E+00**	1.06E+01	3.73E+01
$f_6(x)$	**AB**	**2.74E-01**	4.14E+00	1.53E+00
	Med	**3.00E-01**	4.10E+00	5.50E-01
	STD	**5.17E-02**	4.69E-01	2.94E+00

TABLE 2: Test functions used in the experimental study.

Name	Function	S	Dim	Minimum
Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	$[-100, 100]^n$	30	$X^* = (0, \ldots 0), f(X^*) = 0$
Schwefel 1.2	$f_2(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	$[-100, 100]^n$	30	$X^* = (0, \ldots 0), f(X^*) = 0$
Rosenbrock	$f_3(x) = \sum_{i=1}^{n-1} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + \left(x_i - 1 \right)^2 \right]$	$[-30, 30]^n$	30	$X^* = (1, \ldots 1), f(X^*) = 0$
Sum of Squares	$f_4(x) = \sum_{i=1}^{n} i x_i^2$	$[-10, 10]^n$	30	$X^* = (0, \ldots 0), f(X^*) = 0$
Rastrigin	$f_5(x) = \sum_{i=1}^{n} x_i^2 - 10 \cos(2\pi x_i) + 10$	$[-5.12, 5.12]^n$	30	$X^* = (0, \ldots 0), f(X^*) = 0$
Salomon	$f_6(x) = -\cos\left(2\pi \sqrt{\sum_{i=1}^{n} x_i^2} \right) + 0.1 \sqrt{\sum_{i=1}^{n} x_i^2} + 1$	$[-100, 100]^n$	30	$X^* = (0, \ldots 0), f(X^*) = 0$

repository, including Pima Indians Diabetes, Bupa Liver Disorder, Wisconsin Breast Cancer, Heart-Statlog, Breast Tissue, and Dermatology. Figures 5(a) and 5(b) present the classification performance of each approach for every data set. From the results, the authors demonstrate that the SSORS outperforms its competitors in terms of classification accuracy and speed.

3.2. Modified SSO Approach Based on Beta Distribution and Natural Gradient Local Search (MSSO). Metaheuristic algorithms need a right balance between exploration and exploitation to give successful results in optimization problems. The classical SSO requires the random selection of parameters α, β, and δ *((4) and (5))* to control the movement of the spiders, which can affect the mentioned balance leading the algorithm to a premature convergence. With the aim of improving this balance, Carlos E. Klein

et al. proposed a modification for the SSO (MSSO) in 2016 [25], where the mentioned parameters are selected from a Beta distribution in the range [0-1] [26] instead of the use of random numbers. The use of this distribution helps to preserve diversity and avoid premature convergence, improving the algorithm exploration. Furthermore, to improve the exploitation the author proposed the use of natural gradient (NG) with a rank one covariance matrix approximation to local search in each generation. In this method, the best spider realizes local search using NG after performing the operator. The NG parameters are the learning rate for the mean value and for the scale factor. The complete computational process is summarized in Figure 6 as a flow chart.

To prove the performance of the MSSO algorithm, authors considered the solution of two engineering problems such as the solenoid and brushless motor design. The
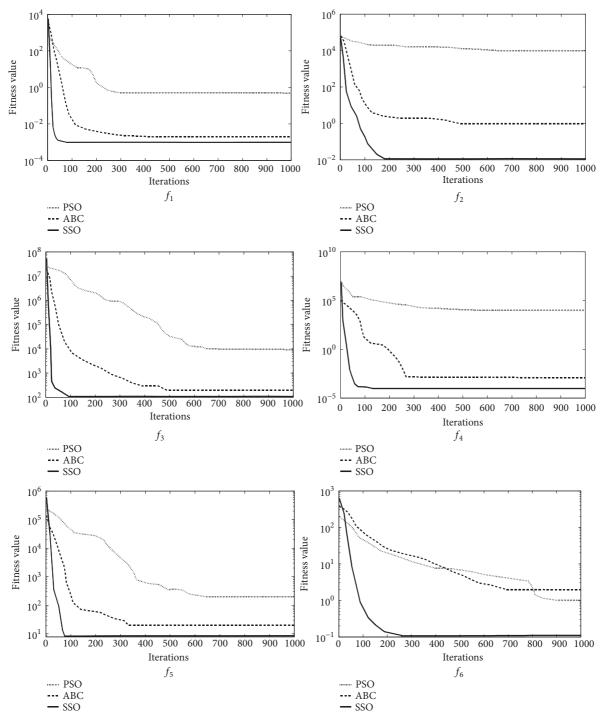
f_1

f_2

f_3

f_4

f_5

f_6

Figure 3: Evolution curves for PSO, ABC, and the proposed algorithm.

algorithm was compared with the SSO algorithm, a version of the SSO algorithm with the NG method for local search and with itself without applying the NG local search method. The MSSO outperforms its competitors in terms of the objective function minimization. Table 3 exposes the performance result of each approach. The results consider 30 executions, a population size of 20 and a stop criterion of 3000 evaluations of the objective function.

3.3. SSO for Constrained Optimization SSO-C.

In 2013, Erik Cuevas et al. [1] introduced a modification of SSO to solve constrained optimization by adding a set of penalty elements to the objective function. With this inclusion, a constrained optimization problem is converted into an unconstrained one. Therefore, the set of penalties gives a feasibility criterion to bias the generation of new individuals toward feasible regions to get better results.

Table 3: Results in terms of the objective function in 30 runs. Highlighted values represent the best responses.

Optimizer	Best	Worst	Mean	std
SSO	2.3285E-08	7.5835E-08	4.2129E-08	0.8993
SSO with NG	2.0891E-08	6.1534E-08	3.9824E-08	0.6823
MSSO without NG	2.0607E-08	6.1751E-08	3.7452E-08	0.6752
MSSO	**2.0566E-08**	**6.1520E-08**	**3.5835E-08**	**0.6522**
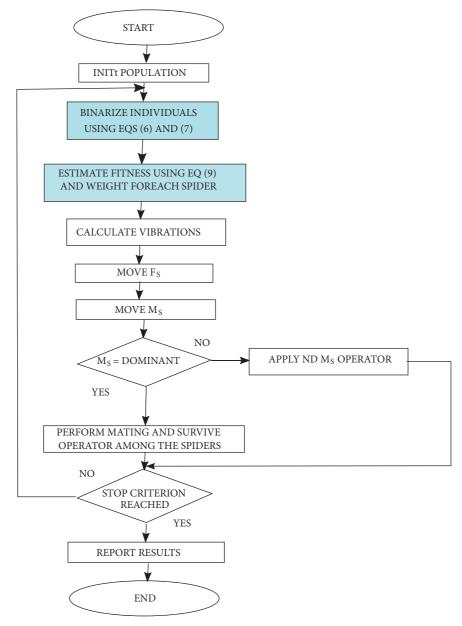
Figure 4: SSORS flowchart.

Most of the optimization problems are subject to different types of constraints. These problems are known as constrained optimization problems. A constrained optimization problem is defined by finding a parameter vector x that minimizes or maximizes an objective function $J(x)$ subject to equality or inequality constraints:

$$\text{minimize or maximize} \quad J(x),$$

$$x(x_1, \ldots, x_n) \in \mathbf{R}^n$$

$$\text{subject to:} \quad l_i \leq x_i \leq u_i, \quad i = 1, \ldots, n$$

$$g_j(x) \leq 0,$$

(a)

(b)

Figure 5: Classification results of the SSORS. (a) Positive matches and (b) negative matches.
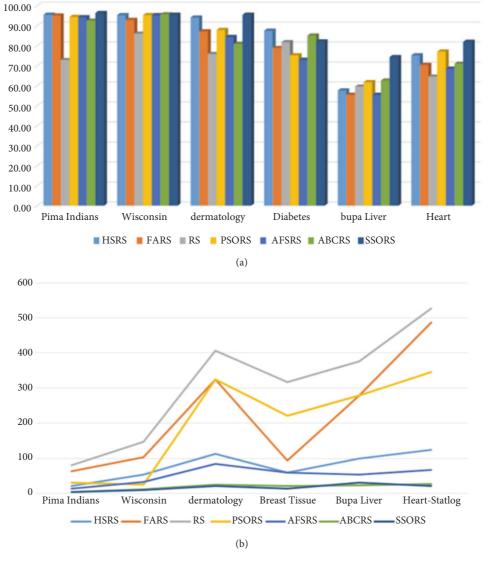
$$\text{for } j = 1, \ldots, q$$

$$h_j(x) = 0,$$

$$\text{for } j = q+1, \ldots, m \tag{10}$$

The objective function is defined on the search space S, which is characterized as an n-dimensional rectangle in $R(S \subseteq R)$. The domain of each variable is defined within the lower and upper limits (l_i and u_i). A feasible region $F \subseteq S$ is delimited by a set of additional constraints ($m \geq 0$) and x is defined on the feasible space ($x \in F \in S$). $x \in F$ at any point, and constraints g_j that satisfies $g_j(x) = 0$ are called active constraints. By extension, equality constraints h_j are so-called active at all points of S.

3.3.1. The Penalty Function. Since the SSO-C aims to solve constrained optimization problems, it replaces the original

objective function $J(x)$ with other function $C(x)$ which considers the original objective function $J(x)$ minus a penalty function $P(x)$ that introduces a tendency term that penalizes constrain violations produced by x. Furthermore, considering the constrained optimization problem defined in equation 10, the substituted and penalty function is defined as follows:

$$P(x) = \mu \cdot \sum_{i=1}^{q} g_i^2(x) + v \cdot \sum_{j=q+1}^{m} h_j^2(x)$$

$$C(x) = J(x) + P(x), \tag{11}$$

where μ and v are the penalty coefficients that weight the importance of each kind of constraint and are set to 1×10^3 and 10, respectively.

The biological metaphor considers the spider size as a characteristic that evaluates the capacity of the individual to perform better a specific task. In this approach every
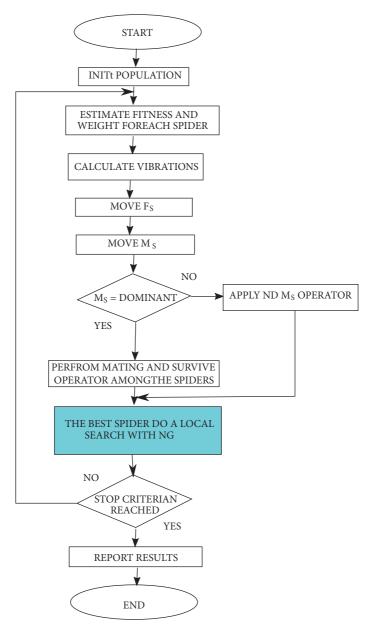
FIGURE 6: MSSO flowchart.

spider receives a weight w_i which represents the quality of the solution of each spider of the population S independently of its gender; this weight is calculated as follows:

$$w_i = \frac{worst_s - C(s)}{worst_s - best_s},$$ (12)

where $C(s)$ is the fitness obtained by the evaluation of the spider position s_i with regard to the substitute function $C(\cdot)$

3.4. Parallel Social Spider Clustering Algorithm for High Dimensional Data sets (P-SSO). As the dimensionality in specific problems increases, it is desirable that the execution time of SI algorithms would be reduced. With this aim, Urvashi Prakash Shukla et al. suggested a parallel version of

the SSO (P-SSO) in 2016 [27]. In this version, the position of each spider (female, dominant and nondominant male) is updated simultaneously; this increases the computational speed, giving the algorithm the ability to work in high dimensional problems. The computational procedure of P-SSO is illustrated in Figure 7 as a flowchart.

This algorithm was applied to solve clustering problems with high dimensional data. According to the authors, the approach performs ten times faster than the original SSO algorithm. The P-SSO outperforms several clustering schemes even in other real-life applications as multispectral image segmentation as a clustering problem. In terms of accuracy, the P-SSO provides two times better precision than one of its competitors.
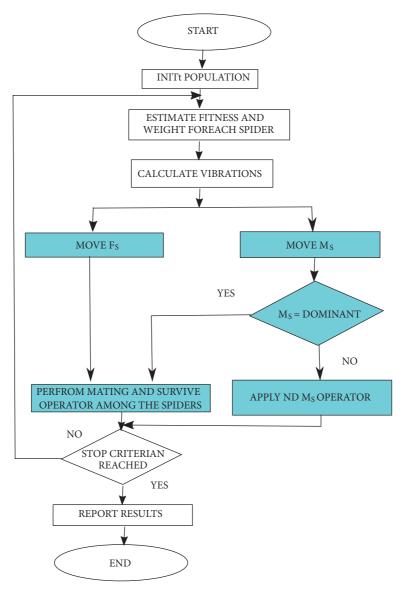
Figure 7: P-SSO flowchart.

3.5. Elite Opposition-Based Social Spider Optimization (EOSSO).

The SSO emulates the cooperative behavior of a spider colony by stochastic position changes; with this method the probability of getting a good solution is relatively low. With the purpose to increase the probability of getting a better solution Ruxin Zhao et al. proposes the use of Elite Opposition-Based Learning Strategy EOLS in the SSO and creates the EOSSO in 2017 [28].

Opposition-Based Learning (OBL) is a machine intelligence strategy that considers the current individual and its opposite particle at the same time to get a better approximation for the current candidate solution. It has been proved that an opposite candidate has a greater chance to be closer to the global optimal than a random candidate solution. Some of the concepts of OBL are defined as follows.

The EOSSO main idea is to calculate and evaluate the opposite solution of each particle at the same time and select the better one as an individual for the next generation. The best fitness valued individual is seen as an elite individual.

At the global optimization process, the strategy expands the search space of the algorithm and strengths the diversity of the population. Thus the global searching ability can be enhanced and help to get a better optimal solution. The complete computational process is summarized in Figure 8.

To test the performance of the ESSO algorithm, it was tested on a set of benchmark functions and compared with other popular algorithms in the literature such as BAT algorithm (BA), ABC, and the original SSO. Table 4 presents the results of each performance considering a population size of 50 particles, a stop criterion of 1000 iterations, and 30 executions. The results indicate that the ESSO method outperforms its competitors in terms of fitness and convergence speed.
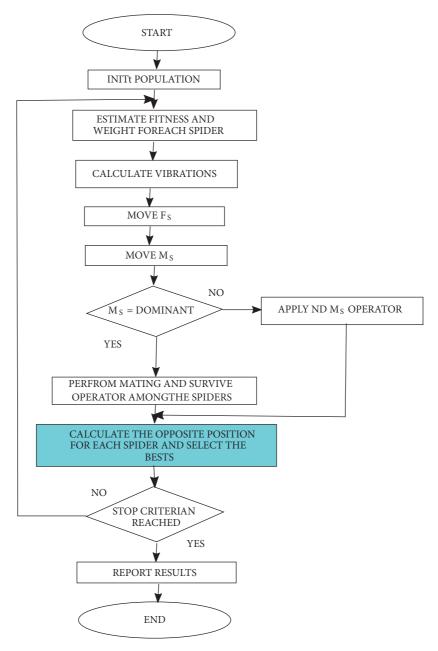
FIGURE 8: EOSSO flowchart.

3.6. Hybrid Social Spider Optimization and Genetic Algorithm. A desirable property for any population-based optimization technique is to avoid premature convergence and entrapment into local optima; with this aim, Tawhid and Ali proposed a hybrid optimization approach based on both SSO and GA algorithms in 2016 [29]. In that approach, the authors combine the properties of exploration and exploitation of the SSO. Then, it applies the arithmetical crossover and mutation operators of the GA, calling the algorithm Hybrid Social Spider Optimization and Genetic Algorithm (HSSOGA). With this combination, the search process is accelerated and helps to find a near optimal solution in a reasonable time.

The HSSOGA algorithm consists of mainly three steps:

(1) Apply the social spider cooperative operators to take advantage of its balance between exploration and exploitation.

(2) Subdivide the population and apply the arithmetical crossover operation on the populations with the aim of improving the search diversity of the proposed algorithm.

(3) Apply the GA mutation operator in order to avoid premature convergence.

Figure 9 presents the computational procedure of HSSOGA as a flowchart.

In order to demonstrate the performance of the HSSOGA algorithm, it was tested on different large-scale unconstrained

TABLE 4: EOSSO performance comparison. Highlighted values represent the best responses.

		EOSSO	SSO	ABC	BA
Sphere	AB	**3.53E-67**	8.63E-02	3.69E-08	4.58E+02
	Med	**3.53E-67**	8.63E-02	5.34E-907	3.13E+03
	STD	2.01E-82	**0E+00**	5.4E-07	1.39E+03
Schwefel 1.2	AB	**7.61E-76**	1.98E+00	1.14E+04	4.33E+03
	Med	**7.61E-76**	19.8E+00	1.97E+05	9.12E+03
	STD	**6.24E-91**	1.13E-15	3.08E+03	3.17E+03
Rosenbrock	AB	**26.8E+00**	35.4E+00	767E+00	243E+00
	Med	**26.8E+00**	35.4E+00	37.9E+00	59.4E+00
	STD	1.08E-14	**0E+00**	20.7E+00	92.4E+00
Rastrigin	AB	**0E+00**	63.3E+00	1.02E+00	1.48E+02
	Med	**0E+00**	63.3E+00	5.47E+00	1.99E+02
	STD	**0E+00**	**0E+00**	2.9E+00	26.3E+02

TABLE 5: General performance of the HSSOGA through different benchmark functions.

dims		Sphere	Schwefel 1.2	Rosenbrock	Rastrigin
30	Med	1200.35	1875.25	28.23	1275.15
	STD	60.14	335.26	0.65	251.17
100	Med	1525.23	2450.36	112.45	1656.23
	STD	45.21	385.16	1.19	35.15
400	Med	1725.16	3355.36	529.05	2215.47
	STD	51.13	391.16	27.16	31.18
1000	Med	2115.39	4715.49	1411.79	3175.47
	STD	61.25	344.59	35.17	45.16

optimization benchmark functions at dimensions 30, 100, and 400. In the comparison, the HSSOGA has been configured with a population size s=25. The Pf was set with 0.7, the crossover rate $P_c = 0.6$, the mutation rate with $Pm = 0.001$, the number of variables in each partition $v = 5$, and the number of solutions in each partition $\eta = 5$. Table 5 shows the performance of the HSSOGA algorithm considering 4 typical functions used in the literature. The authors report the average fitness and the standard deviations given by the HSSOGA through different dimensions and functions.

3.7. Improved Social Spider Optimization (ISSO). With the objective of accelerating the convergence efficiency and search ability of the SSO, Shuang-Cheng Sun et al. [30] proposed five improved SSO algorithms in 2017. The first algorithm ISSO1 takes advantage of the historical process adding the historical best position to operate the movement of the spiders in the search area through a vibration perceived by the i-th spider, leading to an increment of the convergence speed. The second improvement ISSO2 tries to overcome the chaotic search on the earlier stages of the search process due to the low influence of the spider's vibrations on a high distance between them, adding acceleration coefficients to the half of the female and dominant spiders. Furthermore, a random search term that decreases with each iteration and is controlled by an inertia weight is added. A small step size causes a careful search among the spiders but decreases the convergence speed.

On the other hand, a big step size increases the convergence speed but decreases the search accuracy, which may lead to losing the optimal position. To solve these troubles in the ISSO3 algorithm add a linearly decreased step size leading this to a reasonable tradeoff between computational time and search accuracy. In the ISSO4 algorithm, a mutation operator for the nondominant spiders is added with the objective to improve the solutions diversity. Finally, the ISSO5 algorithm makes use of the improvements of the other four algorithms at the same time. Figure 10 presents the computational process of each version ISSO1, ISSO2, ISSO3, and ISSO4 in form of a flowchart.

All algorithms have been applied to solve inverse radiation-conduction heat transfer problems, where each one outperforms the original algorithm in terms of computational speed and accuracy. Figure 11 shows a comparative analysis of all methods.

3.8. Simplex Method Social Spider Optimization. One obstacle for the SSO to be applied in complex problems is its high computational cost. Therefore, as the dimensionality of a search space and the data amount increases, a problem of local optima entrapment and poor convergence rates is present. To overcome these problems, Yongquan Zhou et al. proposed to apply the simplex method to the original SSO (SMSSO) in 2017 [31], enhancing the algorithm global and local search abilities and avoiding local optima entrapment while increasing the convergence rate.
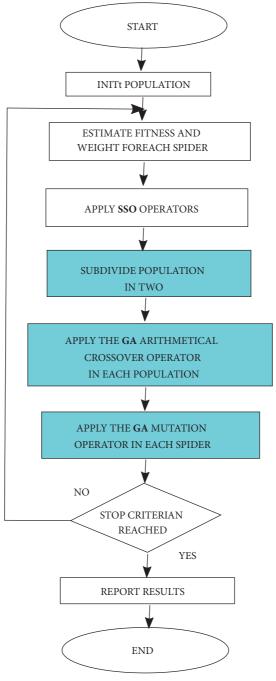
FIGURE 9: HSSOGA flowchart.

The simplex method was introduced by Spendley et al. [32] and is defined by some points equal to the number of dimensions in the search space plus one. The process of the simplex method is described as follows:

(1) Evaluate the solutions of the entire population and select the global best s_{best1} and the second best s_{best2}, assuming that s_r is the spider that must be replaced and that $f(s_{best1})$, $f(s_{best2})$, and $f(s_r)$ are the corresponding fitness values.

(2) Calculate the middle position (s_m) of s_{best1} and s_{best2} defined by

$$s_m = \frac{s_{best1} + s_{best2}}{2} \tag{13}$$

(3) Determine the reflection point s_{ref} given by

$$s_{ref} = s_m + \alpha \left(s_m - s_r \right) \tag{14}$$

The reflection coefficient α is typically set to one.
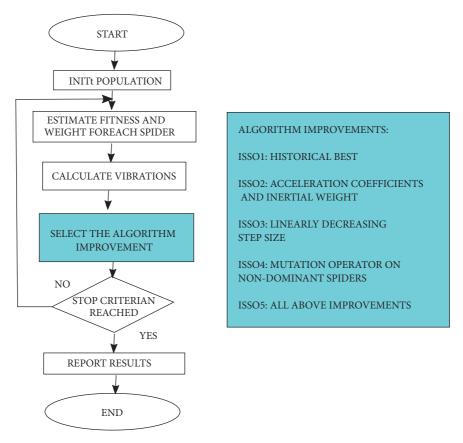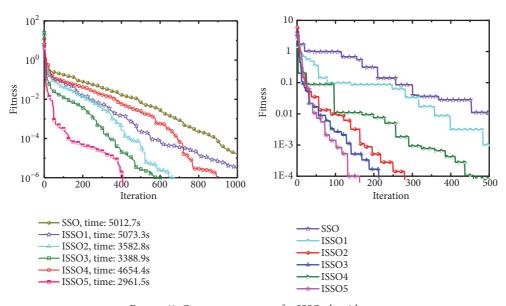
Figure 10: ISSO algorithms flowchart.

Figure 11: Convergence curves for ISSO algorithms.

(4) Compare the fitness value between s_{ref} and s_{best1}. If $f(s_{ref}) < f(s_{best1})$, the extension operation was performed based on the next equation:

$$s_e = s_m + \gamma \left(s_{ref} - S_m \right) \qquad (15)$$

where γ is the extension coefficient, and this parameter is usually set in two. Then compare the fitness value of the extension point and the global best. If $s_e < s_{best1}$, s_r should be replaced by s_e and, in another case, s_{ref} will be substituted for s_r.
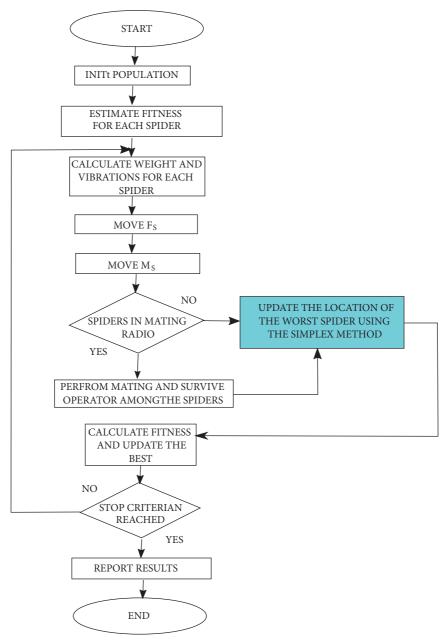
FIGURE 12: SMSSO flowchart.

(5) Compare the fitness values of s_{ref} and s_r. If $f(s_{ref}) < f(s_r)$, the compression operation should be performed using the following equation:

$$s_{comp} = s_m - \beta (s_r - s_m) \qquad (16)$$

β is the condense coefficient; it is typically set to 0.5. Then, it is compared to the fitness values between s_{comp} and s_r, and if $f(s_{comp}) < f(s_r)$, s_r should be replaced with s_{comp}; otherwise, s_{ref} will be substituted with s_r.

(6) If $f(s_{best1}) < f(s_{ref}) < f(s_r)$, the condense point s_{cond} must identify performing shrink operations where the shrink coefficient is described by σ:

$$s_{cond} = s_m - \sigma (s_r - s_m) \qquad (17)$$

If $f(s_{cond}) < f(s_r)$, s_r will be replaced with s_{cond}. In other case, s_{ref} will be replaced with s_r.

The proposed SMSSO algorithm is shown in Figure 12 as a flowchart.

The SMSSO was applied to solve data-clustering problems using different data sets selected from the UCI machine learning repository as artificial data set one (AD1), artificial data set two (AD2), iris data, and Teaching Assistant Evaluation, among others, which confirms its potential and effectiveness. The algorithm was compared to other algorithms in the state of the art such as ABC, DE, SSO, FPA, CLSPSO, and FPA. Each algorithm was applied twenty times, and the quality measures are best, mean, and the standard deviation (Std). Table 6 presents the comparative results of each method for

TABLE 6: Comparison results among four different clustering data sets.

		k-means	ABC	CLSPSO	DE	FPA	SSO	SMSSO
AD1	**Best**	1747.3859	1755.2637	1878.5902	1718.4562	1938.5000	1718.2848	**1718.2537**
	Med	1991.9351	1882.2013	2870.8968	2007.6000	2161.0000	1418.2848	**1718.2537**
	STD	3.4229E+02	1.0475E+02	5.5367E+02	3.3982E+02	1.4448E+02	2.3328E-13	**0E+00**
AD2	**Best**	525.5957	515.2971	513.9034	513.9036	520.4449	514.0804	**513.9034**
	Med	694.4421	517.9773	658.2596	514.3841	536.4305	514.0804	**513.9034**
	STD	1.9148E+02	1.8902E+00	1.8287E+02	1.9550E+01	1.2070E+01	**0E+00**	1.1664E-13
Iris	**Best**	97.1901	97.0001	96.6581	96.6561	97.9097	96.9848	**96.6555**
	Med	100.8866	97.9254	108.4708	97.1050	100.0673	96.9848	**96.6555**
	STD	8.7805E+00	9.7925E+01	1.4706E+01	3.6310E-01	1.7105E+00	2.4580E-14	**1.9160E-14**
TAE	**Best**	1504.9535	1495.7590	1493.9386	1492.1000	1493.6000	1525.7349	**1490.9258**
	Med	1529.6406	1500.1544	1527.6483	1518.8000	1498.8000	1525.7349	**1490.9258**
	STD	2.9749E+01	4.7872E+00	2.8105E+01	1.9777E+01	3.1373E+00	4.6656E-13	**3.6656E-13**

FIGURE 13: Convergence curves among different clustering data sets.

every data set. Figure 13 shows a graphical perspective of all comparisons.

4. Applications

Since its introduction, SSO has been used in a wide variety of applications in several engineering areas such as control, computer vision, image processing, and energy. This section presents a review of some of the most important problems.

4.1. Clustering for High Dimensional Data Sets. In 2016, Urvashi Prakash Shukla et al. [27] proposed a parallel version of the SSO method to segment spectral images. In order to solve the problem, the segmentation task is converted into a clustering problem. Then, the data grouping is approached as an optimization process.

Under this approach, the clustering process is conducted as a minimization of the intraclass distance (ICD) among the data. ICD is a metric of distance between the cluster center

TABLE 7: Performance analysis of SSO and PSSO.

		SSO(CEP)	SSO(time)	PSSO(CEP)	PSSO(time)
Stat	**Med**	13.22	520.87	12.44	48.40
	STD	19.06	20.41	17.17	9.76
Hand	**Med**	9.86	441.69	9.05	41.68
	STD	60.10	26.08	21.57	22.91
Der	**Med**	16.53	15.97	14.23	1.73
	STD	10.99	19.21	19.64	0.23
Mice	**Med**	12.87	60.00	10.98	6.20
	STD	15.73	68.20	21.95	4.28

and the elements present in the cluster. According to the authors, ICD produces compact clusters in comparison with another distance criterion.

Therefore, the proposed objective function f was formulated considering several elements such as the given data set R_{NxD}, the samples N, the dimensions D, and the cluster number K. The objective function was modeled as follows:

$$f = minimizes \sum_{i=1}^{N} \sum_{m=1}^{k} [\delta(r_{i,D}, c_{m,D})]^2 \qquad (18)$$

where the elements $r_{i,D} \in R_{NxD}$ and the cluster head $c_{m,D} \in C_{KxD}$ and $\delta(r_{i,D}, c_{m,D})$ is the Euclidian distance that represents the distance metric.

Each spider represents a random observation of the data set and serves as a cluster center in order to build the different groups. The number of clusters is known in prior as

$$S_{ixT} = [(\vec{r}_1, \vec{r}_2 \dots \vec{r}_K)] \qquad (19)$$

where i-th spider has dimensionality $T = [D \ x \ K]$, thus the swarm is represented by

$$S_{PxT} = [S_{MxT} \cup S_{FxT}] \qquad (20)$$

where P is the population size and M and F are the number of male and female spiders, respectively.

Each observation of R is assigned to a cluster based on the distance from the cluster center; this assignation is given by

$$C_k = \min \sum_{i=1}^{K} \delta \left[\left(R_{j,D}, S_{1+(Dx(i-1)),Dxi} \right) \right]^2;$$

$$\qquad (21)$$

$$\forall j \in [1, N]$$

After the data is assigned to the corresponding clusters using (21), the objective function of each spider is computed by

$$f(i) = ICD[S(i)]; \quad \forall i \in P \qquad (22)$$

The rest of the algorithm continues as usual. In order to test the algorithm performance, it was applied in low dimensional data sets taken from the UCI repository and other high dimensional data sets such as the real-life Statlog (stat). This data set contains multispectral satellite image, handwritten digit (hand), dermatology (der), and Mice, a study based

on protein behavior among others. The results are shown in Table 7, taking as quality measure the classification error percentage (CEP) and the execution time required reporting the median (med) value and the standard deviation (std).
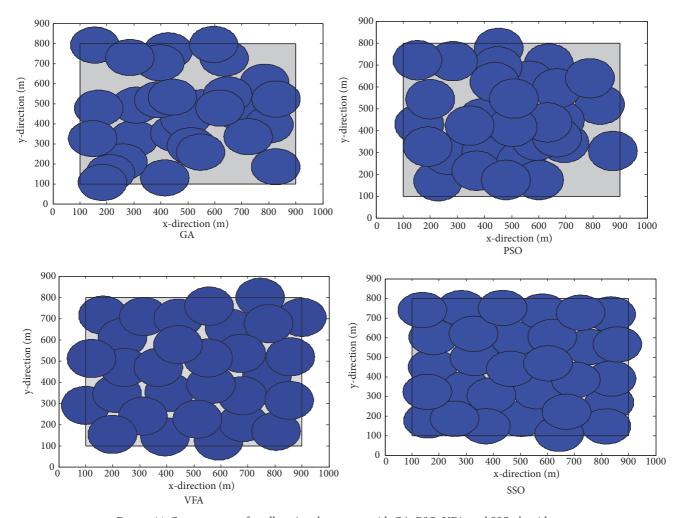
The approach demonstrated a superior performance on small data sets in terms of "average classification error percentage" in comparison with other evolutionary algorithms. For big data sets, the experimental results reveal that the algorithm is ten times computationally faster than the original SSO.

4.2. Simplex Method-Based Social Spider Optimization Algorithm for Clustering Analysis. One of the most popular clustering algorithms due to its simplicity is the K-means algorithm, but its performance strongly depends on the initial cluster centers used and it can converge to a local minimum. Many researchers have attempted to solve the clustering problem with metaheuristic algorithms to deal with these troubles. In 2017 Yongquan Zhou et al. [31] proposed the use of the SMSSO on a data-clustering problem using eleven benchmark data sets to confirm the potential of the proposed algorithm. The experimental results show that the SMSSO algorithm outperforms the K-means technique and other state of the art techniques in terms of accuracy, robustness, and convergence speed.

In the study, the SMSSO algorithm was proposed to solve data-clustering problems. The aim is to divide the data set D, where each datum d_i has m attributes, into k clusters. Furthermore, each attribute of the data set is m. To obtain the optimal coordinates of the centers of K clusters, the dimension of the solution is $k * m$. Each spider in the SMSSO represents a clustering center vector in the clustering problem. It can be described as $C = \{c_1, c_2, \dots, ck\}$. The classification minimizes the sum of the distances required. Hence, the distance between each datum d_i and the center c_j of the cluster to which it belongs is minimized. Finally, the objective function is defined as follows:

$$f(D, C)] = \sum_{i=1}^{n} \min\{\|d_i - c_k\|\} \quad | \ k = 1, 2, \dots, K \qquad (23)$$

The performance of the SMSSO was tested on different sets mentioned in Section 3.7 and some of its results are shown in Table 6 and Figure 14. According to the authors, the approach is better than other similar algorithms with regard to several metrics.

Figure 14: Coverage area after allocating the sensors with GA, PSO, VFA, and SSO algorithms.

4.3. A Hybrid SSO for Text Clustering Quality Improvement. One of the most widely studied data mining problems is text clustering. It organizes text documents into groups with the objective that each group has similar text documents. The main issues in this problem are efficiency and accuracy. As a clustering problem can be mapped to an optimization problem, the researchers have been used evolutionary optimization techniques to improve accuracy and efficiency.

In 2017 T. Ravi Chandran et al. [33] proposed the use of the SSO for textual document clustering. They compared it versus the K-means clustering and another state of the art clustering algorithms such as PSO, ACO, and improved ABC. The results proved to give a better accuracy. Then they proposed the use of two hybrid clustering techniques, the SSO + K-means (SSOK) and K-means + SSO (KSSO), and found that the SSOK algorithm outperforms KSSO, KPSO (K-means + PSO), KGA(K-means + GA), KABC (K-means + ABC), and Interleaved K-means + IBCO clustering techniques.

In this problem, each spider represents a collection of clusters. The algorithm starts initializing each spider with K randomly chosen text documents where K is the number of

clusters to be formed. These K text documents in each spider are the K initial centroids. Each text document in the data set is associated with one of these K centroids based on the distance measure.

The fitness function of each spider is the average distance between text documents and cluster centroid and is defined as follows:

$$fit = \frac{\sum_{i=1}^{K} \left(\sum_{j=1}^{n_i} dist\left(cent_i, doc_j\right)/n_i \right)}{K}, \qquad (24)$$

where $cent_i$ is the centroid of the cluster C_i, doc_j is the text document present in the cluster C_i, n_i is the number of text documents in the cluster C_i, K is the number of clusters in each spider, and $dist$ is the distance measure function that takes two document vectors.

Since there is a smaller average distance between documents and the cluster centroid, then more compact is the clustering solution; thus, the text document clustering problem can be considered a minimization problem.

Each spider is updated in relation to its cooperative behavior. The mating operation is performed for all the dominant male spiders and a set of female spiders in the

TABLE 8: Comparison of mean classification accuracy percentage for sigmoid Kernel in different data sets.

Data set	PSO	NGHS	SSO
Heart	55.55±0.03	55.55±0.01	**60.00±9.04**
Diabetes	65.30±1.42	65.30±0.1.42	65.30±0.1.42
Australian	44.31±0.77	44.31±0.77	**46.93±5.84**
Sonar	47.82±0.00	45.21±12.92	**50.86±7.02**

mating radio. The process is repeated until stop criteria are reached. SSO-based text document clustering is reported in [33].

SSO clustering algorithms produce more compact clusters than K-means due to its globalized searching ability. However, K-means is efficient for large data sets in terms of execution time. In hybridized SSO-based text document clustering, the global search ability of the SSO and the speed of convergence of the K-means algorithm are combined in order to produce an optimal clustering solution.

SSOK algorithm consists of two modules: the SSO module and K-means module. In the initial stage, the SSO module is used to discover the vicinity of the optimal solution by global search. This produces K cluster centroids. These centroids are the input data to the K-means module to refine and generate the optimal clustering solution. The entire process is reported in [33].

KSSO algorithm makes use of two modules: the K-means module and SSO module; in the initial stage the K-means module produces K optimal centroids, and these centroids are passed to SSO module to initialize and start with the SSO clustering process. The entire process is reported in [33].

4.4. Approach for Support Vector Machines Parameters Tuning. Machine learning technics are usually applied to perform pattern recognition tasks, using the data set to train the algorithms in a learning process. The performance and accuracy of these methods are strongly influenced by the selection of their hyperparameters and the choice of the features used in the classification process. Danillo R. Pereira et al. proposed the use of the SSO to perform these parameters choice, specifically focusing in the hyperparameters of support vector machines (SVM) for the Radial Basis Function kernel (RBF) in 2014 [12].

The SSO algorithm was tested in three different SVM kernels: RBF, Sigmoid, and Polynomial, on different problems, and was compared with other optimization approaches, Novel Global Harmony Search (NGHS) and PSO, through different public benchmarking data sets, proving to outperform its competitors in different kernels and data sets in terms of classification accuracy.

In order to compare the optimization methods, the algorithms were executed ten times and the mean execution time and execution times of each of them were calculated. The calibration parameters for each algorithm was set as follows:

 (i) PSO: $c1 = c2 = 2$ and $w = .9$

 (ii) NGHS: $pm=0.1$

 (iii) SSO: $Pf = 0.5$

Each algorithm works with a population of ten individuals and the stop criteria was set in 200 iterations. Table 8 presents the performance results of each algorithm for every data set. According to the authors, the SSO method reaches the best performance indexes.

4.5. Approach to Design Evolutionary Feedforward Neural Networks. Due to its nonlinear nature, the training of feedforward neural networks (FNN) is a complex task. Considering the presence of a large number of local solutions, metaheuristic algorithms are suitable to solve this problem. S. Z. Mirjalili et al. proposed the use of the SSO as a feedforward neural network (FNN) trainer on five standard classification data sets: XOR, balloon, iris, breast cancer, and heart in 2015 [11].

In this approach, the SSO algorithm is proposed to find the weights that characterize the FNN. Therefore, the set of weights is selected to minimize the following formulation:

$$MSE = \sum_{I=1}^{m} \left(O_i^k - d_i^k \right)^{2,} \qquad (25)$$

where O is the actual output of the i-th input, d is the desired output of the i-th input, and m is the number of outputs. Equation (26) is known in the literature as the mean squared error (MSE).

In order to test the performance of the proposed approach, the algorithm was applied over five benchmark data sets from the UCI Machine Learning Repository. Their results have been compared with other well-known algorithms in the literature such as PSO, GA, and ACO. Each algorithm was executed ten times and the population sizes were set in 50 for the XOR data set and 200 for the rest of the sets, and the stop criteria was set in 250 iterations. The performance of each method is presented in Table 9. The results have been exhibited in terms of the average fitness value (AVE), the standard deviation (STD), and the classification rate percentage (%).

The results show that the SSO is very efficient in learning FNN, due to its exploration and high local optima avoidance. In terms of classification, the results of the SSO were better in terms of classification accuracy. It was also observed that the algorithm outperforms the others in iris, breast cancer, and heart data sets.

4.6. Parameter Improvement in the Lukasiewicz Structure. In feature selection, the removal of unnecessary features from the original data sets is of crucial importance to reach accuracy. Furthermore, dimensionality represents an important obstacle which affects the accuracy in data classification.

TABLE 9: Performance comparison of FNN training.

		SSO	PSO	ACO	GA
XOR	**AVE**	**2.807E-05**	0.84050	0.180328	0.000181
	STD	**0E+00**	0.035945	0.025268	0.000413
	%	**100**	37.5	62.5	**100**
Iris	**AVE**	**0.0210**	0.228680	0.4705979	0.089912
	STD	**3.6571E-18**	0.057235	0.053775	0.170260
	%	**91.33**	37.33	32.66	89.33
Heart	**AVE**	**0.0627**	0.188568	0.228430	0.093047
	STD	**1.4628E-17**	0.008939	0.004979	0.022460
	%	67.5	68.75	0	**71.25**
Breast Cancer	**AVE**	**0.001600**	0.034881	0.013510	0.003026
	STD	**2.28E-19**	0.002472	0.002137	0.001500
	%	**98**	11	40	**98**
Balloon	**AVE**	1.7395E-15	0.000585	0.004854	**5.08E-24**
	STD	**0**	0.000749	0.007760	1.06E-23
	%	**100**	**100**	**100**	**100**

TABLE 10: Mean accuracy comparison for SSO parameter tuning.

		SSO	PSO	QPSO
Dermatology	**MCA**	**97.85**	96.05	96.31
	STD	**0.00252**	0.00488	0.02078
Parkinson's	**MCA**	**85.33**	80.15	85.09
	STD	**0.00977**	0.02883	0.02644
Wine	**MCA**	**96.76**	90.04	96.19
	STD	**0.00275**	0.00928	0.0282
Echocardiogram	**MCA**	**89.98**	89.31	89.43
	STD	**0.00679**	0.00678	0.01420

Under such conditions, the membership of an object can be defined by the generalized Lukasiewicz structure [34]. The membership accuracy depends on the parameters that characterize the structure. Jintara Dollaor et al. proposed the use of the SSO to determine suitable values of similarity in conjunction with a fuzzy-entropy evaluation to select data features for the optimal parameters in the Lukasiewicz structure in 2016 [35].

In the approach, the SSO was proposed to find the optimal parameter p for generalized Lukasiewicz structure. Therefore, the objective function is modeled as follows:

$$fit(S) = \max_{i=1,\dots N} S(\boldsymbol{x}, \boldsymbol{v}), \tag{26}$$

where

$$S(\boldsymbol{x}, \boldsymbol{v}) = \left(\frac{1}{t}\sum_{r=1}^{t} w_r \left(1 - \left|x(f_r)^P - v(f_r)^P\right|\right)^{m/P}\right)^{1/m}, \tag{27}$$

where $\boldsymbol{x}, \boldsymbol{v} \in [0,1]^t$ is a similarity value of sample $\boldsymbol{x}$ and $\boldsymbol{v}$ is the ideal vector represented by $\boldsymbol{x} = (x(f_1), x(f_2, \dots, x(f_t))$, $\boldsymbol{v}_i = (v_i(f_1), v_i(f_2), \dots, v_i(f_t))$. t represents the number of features $f_1, f_2, \dots, f_t$, w_r is a weight parameter, and p is a parameter in the adjustment similarity value of data class i;

a variation in this parameter can decrease or increase the classification performance.

In order to demonstrate the performance of the approach, it was tested in a benchmark data set of the UCI machine learning repository and was compared with the PSO and QPSO algorithms. Each algorithm was executed 30 times, the population was set in 50, and the stop criteria were set in 1000 iterations. Table 10 presents the computational results of each approach. In the table, the mean classification accuracy (MCA) and the standard deviation (STD) were used as quality measures.

The results demonstrated that the SSO generates a higher mean classification than PSO and QPSO in terms of average accuracy and average rank. It was also found that the SSO has been capable of seeking optimal parameter within cooperative neural network achieving the best classification results with a reduced number of features.

4.7. Image Fusion Approach for Contrast Enhancement and Brightness Preservation. Image enhancement is one image processing technique with the objective to generate an image with superior information contents in terms of quality and quantity. This processing typically improves the appearance of the output image by increasing its contrast and sharpness

TABLE 11: Performance comparison for SSO image enhancement.

		HE	LCS	PSO	SSO
Lena	**Fitness**	35.1391	150.1672	170.4274	**260.8717**
	PSNR	67.2983	70.2172	70.2376	**73.4711**
	AMBE	0.0136	0.0003	0.0422	**0.0223**
Cameraman	**Fitness**	31.7623	90.5409	154.0495	**209.0478**
	PSNR	67.2707	66.7157	64.3244	**69.0729**
	AMBE	0.0341	0.0932	0.1196	**0.0373**
Clock	**Fitness**	14.4639	77.5237	4.9646	**163.3543**
	PSNR	59.5811	71.1069	58.8907	**70.7312**
	AMBE	0.2290	0.0129	0.2545	**0.0187**
Aeroplane	**Fitness**	14.4011	34.9344	1.8144	**64.9838**
	PSNR	58.6971	70.1181	58.5824	**76.1722**
	AMBE	0.2577	0.0288	0.2859	**0.0016**

features. Image enhancement based on spatial domain uses a transform function that takes the intensity value of each pixel from the original picture and generates a new intensity value of the corresponding pixel to produce the enhanced image. Lalit Maurya et al. proposed the use of the SSO to enhance two images, one having high peak signal to noise ratio (PNRS) and the other having high contrast with maximum entropy in 2016 [19]. Both images were fused to obtain an output image that has an optimal tradeoff between contrast and noise.

In the proposed approach the SSO gives two sets of parameters for image enhancement; one has high fitness given by maximizing the objective function defined as follows:

$$F\left(I_e\right) = \log\left(\log\left(E\left(I_e\right)\right)\right) x \frac{edgels\left(I_e\right)}{MxN} xe^{H(I_e)} \qquad (28)$$

where I_e is the enhanced image generated by applying the transformation function on the original image, edgels($\cdot$) is the number of edge pixels of the enhanced image, and M x N is the size of the image. H($\cdot$) is the entropy of the enhanced image and another image of high PSNR value given by

$$PSNR\left(r, e\right) = 10\log_{10}\left(\frac{(L-1)^2}{MSE\left(r, e\right)}\right) db, \qquad (29)$$

where r, e are the reference and the enhanced image, respectively. L is the dynamic range of the pixel values.

The main idea is to approach the two populations in the SSO algorithm to give the two optimal set of parameters, while the male spiders use the PSNR as evolutionary criteria. Meanwhile, the female spiders are updated using the fitness value given by (28), and finally the two best global positions are used as the enhanced images and fused to give the output image.

In order to probe the performance of the approach, the algorithm was tested in different benchmarking images and compared with both classic techniques as Histogram Equalization (HE) and Linear Contrast Stretching (LCS) and swarm algorithms as PSO.

In order to compare between both swarm algorithms, their population was set in 30 with a stop criteria of 40

iterations. Their calibration parameters were configured as follows:

(i) PSO: *c1 = c2 = 1.8, w =0.6*

(ii) SSO: *Pf = was set as a fraction of the maximum iteration number*

Table 11 presents the computational results of each algorithm for every image of the data set. In the table, the authors report the fitness values, the PSNR metric, and the Absolute Mean Brightness Error (AMBE). According to the authors, the SSO presented the best possible performance in comparison with its competitors.

4.8. Performance Study in Multilevel Image Thresholding Approach. In image thresholding, an image is divided into one or more regions based on one or several threshold values. The approach evaluates the quality of the segmentation through the use of an objective function. In the segmentation problem, the objective is to find the optimal vector of intensity levels $t* = (t_1{}^*, t_2{}^*, \ldots t_k{}^*)$ that maximizes a given evaluation function. Salima Ouadfel et al. presented a performance study of the SSO and flower pollination algorithm (FP) to solve image segmentation problems via Multilevel Thresholding in 2016 [18].

For the study two objective functions were considered. The between-class variance is defined as follows:

$$fit = \sum_{i=0}^{k} \sigma_i, \qquad (30)$$

where

$$\sigma_i = \omega_i\left(\mu_i - \mu_t\right)^2 \qquad (31)$$

$$\mu_i = \sum_{j=t_i}^{t_{i+1}-1} \frac{iP_j}{\omega_j},$$

$$\qquad (32)$$

$$\omega_i = \sum_{j=t_i}^{t_{i+1}-1} P_j,$$

TABLE 12: Fitness values comparison for Kapur's entropy for test images.

	Thresholds	SSO	FP	BAT	PSO
Wherry	2	**12.177692**	**12.177692**	**12.177692**	**12.177692**
	3	**15.141712**	**15.141712**	15.141113	15.141689
	4	**18.004651**	18.176303	17.994920	18.003399
	5	**20.696338**	20.687249	20.686953	20.671377
Zebra	2	**12.258460**	**12.258460**	**12.258460**	**12.258460**
	3	**15.338314**	**15.338314**	**15.338314**	**15.338314**
	4	**18.179248**	18.176303	18.171796	18.158891
	5	**20.996743**	20.981965	20.950633	20.950453
Butterfly	2	**12.683959**	**12.683959**	**12.683959**	**12.683959**
	3	**15.801767**	**15.801767**	15.801176	15.801159
	4	**18.804220**	**18.804220**	18.798495	18.790101
	5	**21.482543**	21.474797	21.470672	21.470206
Bird	2	**12.088437**	**12.088437**	**12.088437**	**12.088437**
	3	**15.198964**	**15.198964**	**15.198964**	**15.198964**
	4	**17.982624**	17.982427	17.981291	17.979551
	5	**20.604057**	20.602413	20.598996	20.572345

where μ_T is the mean intensity of the original image with $t_0 = 0$ and $t_{k+1} = L$.

And the Kapur's entropy is as follows:

$$fit = \sum_{i=0}^{k} H_i \qquad (33)$$

$$H_i = -\sum_{i=t_i}^{t_{i+1}-1} \frac{P_j}{\omega_j} \ln \frac{P_j}{\omega_j},$$

$$\omega_i = \sum_{j=t_i}^{t_{i+1}-1} P_j \qquad (34)$$

for $t_0 = 0$ and $t_{k+1} = L$.

The performance of the algorithms has been evaluated using benchmark images and compared against other two algorithms in the literature: PSO and BAT; all algorithms used a population size of 50, and the stop criteria was set in 10000 function calls, and the calibration parameters for each algorithm was set as follows:

(i) PSO: $c1 = c2 = 1.429$, $w = [0.9, 0.4]$. $V = (1, -1)$

(ii) SSO: $Pf = 0.7$

(iii) FP: 0.8

(iv) BAT: $A = 0.25$, $R = 0.5$, $Q = (0.2, 0)$

Table 12 presents the computational results of each approach over every image from the data set. The experimental results obtained from several benchmark images show that the algorithms have the same results for bi-level thresholding due to the relatively low complexity of the problem. For three-level thresholding, the SSO and FP outperform the other algorithms. Finally, for a higher number of thresholds, the SSO showed its stability and accuracy due to its balance between exploration and exploitation in the search space. In all cases, for both objective functions, the SSO outperforms all the other algorithms with which it has been compared in terms of fitness values and stability.

4.9. SSO-Based Template Matching. Template matching (TM) is a key procedure for many different image processing applications. The main purpose of a TM approach is to find the best possible resemblance between a subimage (image template) and some coincident region within a source image. Typically, TM involves an exhaustive search process across every single pixel position within the source image where a similarity measurement (such as the Normalized Cross-Correlation) between a given image template of size and a corresponding subregion within the said source image is computed for each of said positions.

In 2017, Cuevas et al. [20] proposed an SSO-based TM approach in which a reduction of the computational cost related to the exhaustive evaluation of NCC values is achieved by applying said measurement to selected pixel positions provided by the SSO's search agents. For this purpose, the algorithm considers that each spider (regardless of their gender) represents a 2-D position (u, v) inside a delimited search region S within the given source image I of size $M \times N$. Furthermore, for each of said positions, a similarity value with regard to a given image template R of size $m \times n$ and a corresponding subregion from the source image is evaluated and used as fitness values. With that being said, each spider may be defined by

$$\mathbf{s}_i = \left\{ (u_i, v_j) \mid 1 \le u_i \le M - m, \ 1 \le v_i \le N - n \right\} \qquad (35)$$

Furthermore, the similarity value (fitness) related to each spider position $\mathbf{s}_i$ and the applied image template R is given by the NCC value, as given by

TABLE 13: Performance comparison of EC, ICA, PSO, and SSO algorithms.

	Algorithm	AET	SR	ANCL	AFE
Dogs	EC	10.45	**100**	219065	219065
	ICA	1.71	80	30100	30100
	PSO	1.55	3.33	29217	29217
	SSO	**1.35**	**100**	**7856**	**16521**
Faces	EC	280.39	**100**	1277963	1277963
	ICA	7.10	**100**	30100	30100
	PSO	11.02	26.66	29256	29256
	SSO	**4.19**	**100**	**2654**	**4842**
Disaster	EC	289.67	**100**	588826	588826
	ICA	7.01	50	30100	30100
	PSO	10.27	20	29342	29342
	SSO	**6.84**	99.66	**7896**	**18677**
Map	EC	68.31	**100**	616499	616499
	ICA	7.17	36.66	30100	30100
	PSO	9.82	36.66	29369	29369
	SSO	**7.14**	96.66	**10561**	**18933**

$$\text{NCC}\left(u_i, v_j\right) = \frac{\sum_{x=1}^{m} \sum_{y=1}^{n} \left[I\left(u_i + x, v_j + y\right) - \overline{I}\left(u_i, v_i\right)\right] \bullet \left[R\left(x, y\right) - \overline{R}\right]}{\sqrt{\left[\sum_{x=1}^{m} \sum_{y=1}^{n} \left[I\left(u_i + x, v_i + y\right) - \overline{I}\left(u_i, v_i\right)\right]\right]^2 \bullet \left[\sum_{x=1}^{m} \sum_{y=1}^{n} \left[R\left(x, x\right) - \overline{R}\right]\right]^2}} \tag{36}$$

where $\overline{I}(u_i, v_i)$ stand for the gray-scale average intensity of the source image I for the coincident region defined by the image template R at the pixel position (u_i, v_j), while $\overline{R}$ stand for the gray-scale average intensity of R. These values are given by

$$\overline{I}\left(u_i, v_i\right) = \frac{1}{m \bullet n} \sum_{x=1}^{m} \sum_{y=1}^{n} I\left(u_i + x, v_i + y\right)$$

$$\overline{R} = \frac{1}{m \bullet n} \sum_{x=1}^{m} \sum_{y=1}^{n} R\left(x, y\right) \tag{37}$$

Furthermore, this SSO-based TM approach also includes a memory mechanism applied to reduce computational cost further. Essentially, this mechanism is used to store previously evaluated positions (along with their respective NCC values) which are constantly compared with every new search position provided by each spider in order to avoid reevaluating the same solutions.

In order to test the performance of the algorithm, a series of comparative experiments was developed. The algorithm was applied over a set of benchmark images and compared against exhaustive computation (EC), the imperialist competitive algorithm (ICA), and PSO; each algorithm was run 30 times, their populations were set in 100, and the stop criteria were set as follows:

(i) PSO: *c1 = c2 = 15, V= (1, -1)*

(ii) SSO: *Pf =0.7*

(iii) *ICA: NumOfImper = 90, NumOfColony = 90 ξ = 0.1, ε_1 = 0.15 and ε_2 = 0.9*

Table 13 shows the computational results of each algorithm for every image of the data set. The comparisons consider four quality indexes: average elapsed time (AET), success rate (SR), average number of checked locations (ANCL), and the average number of function evaluations (AFE).

Experimental results probe a higher performance of the SSO algorithm in terms of elapsed time and number of NCC evaluations. According to the authors, the SSO version obtains the best performance results in comparison with its competitors.

4.10. Unscented Transform for Wind Turbine Uncertainty for Economic Dispatch Approach. In power systems analysis, the impact of wind power generation can be considered as an inseparable part of power grids. In 2015, Reza Khorramnia et al. [36] proposed a modification of the SSO as a probabilistic method to meet large nonconvex economic dispatch (ED) problems under uncertainties caused by wind turbines. The unscented transform was utilized to model the uncertainty in wind speed and to solve the ED problem.

In the approach, the SSO allocates wind and thermal plants with the objective to minimize the total generation cost during a certain interval of time, while satisfying the restrictions mentioned in [19]. Therefore, the optimization problem can be formulated as follows:

TABLE 14: Comparison of the production cost of 15 units test system (ED) through different algorithms.

	Best	Average	Worst
GA	33113.0	33228.0	33337.0
PSO	32858.0	33039.0	33331.0
SSO	**32680.4536**	**32687.23221**	**32693.1135**

$$minimize \ F_T = \sum_{i=1}^{n} F_i(P_i), \tag{38}$$

where $F_i(\cdot)$ is the fuel cost of a thermal unit that is modeled as a second-order polynomial function of its generation P_i. The model also includes several valve effects. In general, the model can be defined as follows:

$$F_i(P_i) = a_{ij} + b_{ij}P_i + c_i jP_i^2$$
$$+ \left| e_{ij} \sin \left(f_{ij} \left(P_{ij}^{min} - P_i \right) \right) \right| \tag{39}$$

$$if \ fuelj, P_{ij}^{min} \leq P_{ij} \leq P_{ij}^{max}, \tag{40}$$
$$\forall i = 1, 2, \ldots, n \ and \ \forall j = 1, 2, \ldots, N_{fuel}$$

where a_{ij}, b_{ij}, c_{ij}, e_{ij}, and f_{ij} are the cost coefficients for the ij-th generating unit.

In order to prove the approach, the algorithm was compared with other similar algorithms in the literature such as GA and PSO, in different ED system units reporting the best, worst, and average results. Table 14 presents the experimental result of each scheme.

Experimental results demonstrated that SSO delivers better solutions by providing a more efficient search. The approach saves time and requires a lower computation overload in comparison with other stochastic techniques.

4.11. Optimal Congestion Management Approach.

Congestion management is an essential topic in the emerging electric power market. It can impose a barrier to electricity trading. In a no regulated environment, it can be said that a system is congested when some operating restrictions are violated. In an optimal power flow method, the minimization of the congestion cost and service is used to regulate the power injection. Zakaria Hejrati et al. proposed the use of the SSO to determine the minimum redispatch cost for an optimal congestion management in a no regulated electricity market in 2014 [22].

In the approach, the SSO method is used to minimize the redispatch cost given by

$$\min \sum_{g}^{N_g} IDC_g \left(\Delta P_g \right) \Delta P_g \tag{41}$$

subject to the following:

Power balance constraint:

$$\sum_{g=1}^{N_g} \Delta P_g = 0 \tag{42}$$

TABLE 15: Redispatch cost in 24 lines obtained through different algorithms.

	Best	Average	Worst
GA	540.15	596.4	613.3
EP	550.9	596.4	642.1
SSO	**527.8**	**551.5**	**613.3**

Operating limit constraint:

$$\Delta P_g^{min} \leq \Delta P_g \leq \Delta P_g^{max}, \quad g = 1, 2, \ldots, N_g \tag{43}$$

Line flow constraint:

$$\sum_{g=1}^{N_g} \left(GS_g^{ij} \cdot \Delta P_g \right) + F_l^0 \leq F_l^{max}, \quad l = 1, 2, \ldots, N \tag{44}$$

The experiments have been conducted considering the IEEE 24-bus and 118-bus systems. In the study, they compared the SSO results with those produced by Genetic Algorithms (GA) and Evolutionary Programming (EP). The algorithms were run 20 times and, as a quality measure, the best average and worst cost were reported in Table 15.

The experimental results showed that the SSO presents better redispatch costs in comparison with other approaches such as GA and EP.

4.12. Efficient Frequency Controllers for Autonomous Two-Area Hybrid Microgrid System.

Hybrid microgrid system (HμGS) which comprises different types of renewable sources has attracted the attention of researchers and industry. Due to the production variability of renewable energies, the system output experiments voltage fluctuations. The main role of the controller is to ensure that the system finally eliminates the fluctuations. Attia A. El-Fergany et al. proposed the use of the SSO for tuning the gains of a PID controller in 2016 [17]. These gains and the absolute deviations are used to build the fitness function and formulate the optimization problem.

With the aim of finding the optimal tuning parameters for the PID controller, it is necessary to take into consideration the deviations on the line power, which are evaluated by using the absolute error (ITAE). Under such conditions, the objective function is defined as follows:

$$minimize \left\{ \int_0^{t_{sim}} t \cdot \left(|\Delta F_1| + |\Delta F_2| + |\Delta P_{tie}| \right) \cdot dt \right\} \tag{45}$$

subject to

$$k_{p,i,d \ min}^k \leq k_{p,i,d}^k \leq k_{p,i,d \ max}^k \quad \forall k \in N_{pid} \tag{46}$$

TABLE 16: Performance results in terms of ITAE, MNUS, and MPOS, in Δ f1, reported in [15].

	Best fitness (ITAE)	MNUS	MPOS
GA	6.5975E-03	1.4237E-03	5.1905E-04
SSO	**6.5476E-03**	**1.4024E-03**	**4.8399E-04**

TABLE 17: Comparison of the values of different error indexes.

	FOPID (SSO)	PID (TLBO)
ISE	**5.2803E-05**	11.1210E-02
ITSE	**2.4243E-05**	15.4289E-04
IAE	**1.2500E-02**	11.3410E-02
ITAE	1.1500E-02	**30.1024E-04**

In order to test the performance of the algorithm, the approach was validated considering the control of an isolated hybrid microgrid system (HμGS) under several energy storage scenarios. It was compared with the GA. Each algorithm was run 50 times with a stop criterion of 100 iterations. Table 16 presents the experimental results exhibiting the best fitness value, the maximum negative undershoots (MNUS), and the maximum positive overshoot (MPOS).

The results demonstrated that the proposed approach eliminates the signal deviations getting a fast response and robustness. Such results show a better performance of the SSO method in comparison with GA.

4.13. Optimal Design of Fractional Controller for LFC in an Interconnected Multisource Power System. Fractional order elements are a generalization of integration and differentiation operators that do not follow the conventional integer characteristics of the traditional calculus. This integral differential operator, denoted by $_aD_t^\alpha$, takes the fractional integral and fractional derivative into a single expression defined as

$$_aD_t^\alpha = \begin{cases} \dfrac{d^\alpha}{dt^\alpha}, & \alpha \geq 0 \\ 1, & \alpha = 0 \\ \displaystyle\int_\alpha^t (d\tau)^\alpha, & \alpha < 0 \end{cases} \tag{47}$$

The difference between the conventional PID controller and a fractional order PID (PI$^\lambda$D$^\mu$) are the two extra parameters (λ, μ) which help to add flexibility in the controller design improving its dynamic response. The mathematical formulation for a FOPID controller is

$$C(s) = k_p + k_i \frac{1}{s^\lambda} + k_d s^\mu \tag{48}$$

where k_p is the proportional gain, k_i the integrator and k_d the derivative gains, also λ and μ are the fractional order of the integrator and differentiator, respectively.

H. Shayeghi et al. proposed the SSO to find the optimal design parameters of a fractional order proportional integrative derivative (FOPID) controller for an interconnected multisource system in 2016 [15]. In this approach the SSO finds the optimal parameters k_p, k_i, k_d, λ, and μ, which minimizes different error indexes such as Integral Time multiplied Absolute Error (ITAE), Integral of Squared Error (ISE), Integral of Time multiplied Squared Error (ITSE), Integral of Absolute Error (IAE), Integral Square Time of Square Error (ISTSE), and the Integral Square Time of Absolute Error (ISTAE).

In order to test the approach performance, the results given by the SSO were compared to those given by the Teaching Learning Based Optimization (TLBO) method. The schemes have been applied on a two-area test system that includes a high voltage direct current (HVDC) region.

The calibration parameters of the SSO were set as follows:

  (i) Population size = 100

  (ii) Iterations = 100

  (iii) Female percent = 88

  (iv) Male percent = 12

  (v) *Pf*

Table 17 presents the experimental results. According to authors, the SSO demonstrated to obtain the optimal parameters for the controller. The simulation results showed that the scheme has a higher ability than other methods getting a better performance in terms of damping and deviation. The experiments reveal that the controller obtained by SSO maintains a better performance in presence of uncertainties, nonlinearities, and disturbances.

4.14. Evolutionary Calibration of Fractional Fuzzy Controllers. A fuzzy controller (FC) is a nonlinear control system based on empirical rules. These rules usually come from a human operator that directly manipulates the process with base in the natural language of the expert. These rules usually come in the form IF-THEN. The collection of rules emulates the decision-making process of the operator. In spite of several configurations of FCs in literature, the fuzzy fractional $PD^\alpha + I$ present interesting characteristics of robustness and stability. In this configuration, the integral error is added to the output of the fuzzy fractional PD^α controller and helps to eliminate the final steady state error.

Erik Cuevas et al. introduced the use of the SSO for the parameter calibration of a fractional order fuzzy controller in 2017 [14]. The approach is compared with other well-known evolutionary algorithms. In the calibration process, the SSO finds the parameters $k_p, k_d, k_i,$ and k_v and the fractional order α of the differential operator in the system by which the ITAE performance index is minimized defined as

$$\int_0^\infty t \, |r(t) - y(t)| \, dt \tag{49}$$

where α is the order of the fractional differentiator, k_p, k_d, k_i, and k_v are the proportional, derivative, integral, and output gains of the controller, respectively, $r(\cdot)$ represents the reference input of the controller, and $y(\cdot)$ represents the output of the control system so the difference between $r(\cdot)$ and $y(\cdot)$ is the error of the control system.

TABLE 18: Plants used to test the performance of the approach.

	Function
High-order plant	$G_1(s) = \dfrac{1}{(s+1)(1+0.5s)(1+0.25s)(1+0.125s)}$
Nonminimum system	$G_2(s) = \dfrac{1}{(s+1)^3}$
Dynamic fractional system	$G_3(s) = \dfrac{1}{(s^{1.5}+1)}$

TABLE 19: Fitness of plants $G_1(s)$, $G_2(s)$, and $G_3(s)$ given by different algorithms.

Thresholds	PSO	GA	GSA	HS	CS	SSO
$G_1(s)$100it	5281.2115	926.1352	4164.0935	3562.1834	916.5816	**492.2912**
$G_1(s)$500it	2900.7502	974.0881	1975.3254	2776.2160	**464.5376**	473.7492
$G_2(s)$100it	50289.0994	55043.6316	101160.6241	126996.7047	90962.6199	**44368.6620**
$G_2(s)$500it	46912.4985	81550.0790	63186.5783	91536.3894	43565.1588	**41772.3344**
$G_3(s)$100it	311.4558	**97.7016**	346.6765	3562.1834	108.0266	98.5974
$G_3(s)$500it	181.5380	97.1981	152.2192	2776.2160	**97.0307**	97.1085

The proposed scheme was experimentally evaluated considering three different systems: high-order plants, nonminimum systems, and dynamical fractional systems. Table 18 presents the plants used in the analysis. The algorithm was also compared against other similar approaches such as PSO, GA, GSA, HS, and CS. Each algorithm was run 35 times, their population size was set in 50 and the stop criteria was set in 100 and 500 iterations to test the quality of the solutions when the operation of each calibration method is limited to a reduced number of iterations. As a quality measure, the fitness value for each plant was reported.

The calibration parameters for each algorithm were set as follows:

(i) PSO: *c1 = c2 = 2, w= (0.9, 0.4)*

(ii) GA: *cp = 0.55 mp = 0.1*

(iii) GSA: *α = 10, G_O = 100*

(iv) HS: *HMCR = 0.7 PAR = 0.3 BW = 0.1*

(v) CS: *p_a = 0.25 β = 3/2*

(vi) SSO: *Pf =0.7*

Table 19 presents the experimental results. According to the authors, the algorithm demonstrated to outperform other techniques in terms of quality and convergence.

4.15. Sensor Deployment Scheme for Wireless Sensor Networks Based on SSO. Wireless Sensor Networks (WSNs) consist of multiple sensors distributed through a specified sensing area with the purpose of acquiring some kind of physical or environmental data, such as pressure, temperature, or sounds. Each sensor node has capabilities for sensing, processing, and communication, the latter of which allows transmission and data sharing. Other applications for WSNs include the monitoring and control of industrial processes, vehicular tracking, forests, and buildings monitoring, seismic activity observation, machine health monitoring, and surveillance. The main problem regarding the deployment of WSNs is the optimal placement of sensor nodes along a specified service area; in this sense, the aim is to allow sensor networks to achieve sufficient coverage, so that each location within the service area is monitored by at least one sensor node, while also reducing WSNs construction and communication costs and an efficient resource management.

In 2017, Zhou et al. [37] proposed a sensor deployment scheme for WSNs based on the SSO algorithm. In said approach, each search agent (spider) is used to represent a set of 2-D discrete positions (x, y) within a prespecified grid G of size $m \times n$, which represent the sensors' deployment area. As such, each spider is represented by

$$\mathbf{s}_i = [x_1, y_1, x_2, y_2, \ldots, x_N, y_N] \qquad (50)$$

where the paired elements (x_i, y_i) represent candidate positions for the placement of sensor nodes within said grid G (search space), while N stand for the maximum number of sensor nodes that are to be deployed.

Furthermore, each sensor node is assumed to have a finite coverage radius R_s. Thus, the objective is to place all of the N available sensor nodes so that the covered area within the specified grid G is maximized. With that being said, for said sensor allocation scheme, the target objective function is given by grid G's rate of coverage, as given by

$$R_{cov}(S) = \frac{\sum_{x=1}^{m} \sum_{y=1}^{n} P(x, y, S)}{m \times n} \qquad (51)$$

where $S = \{s_1, s_2, \ldots, s_N\}$ denotes the set of all N sensor nodes while $P(x, y, S)$ stand for the coverage probability for a given grid point $G(x, y)$, relative to the placement of all sensor nodes s_i, as given by

$$P(x, y, S) = 1 - \prod_{i=1}^{N} (1 - P(x, y, s_i)) \qquad (52)$$

with $P(x, y, s_i)$ denoting the probability that the grid point $G(x, y)$ is covered by an specific sensor node s_i, given as follows:

$$P(x, y, s_i) = \begin{cases} 1 & \text{if } \sqrt{(x - x_i)^2 + (y - y_i)^2} \leq R_s \\ 0 & \text{otherwise} \end{cases} \quad (53)$$

In order to test the performance of the approach diverse studies, mainly focused on the coverage area of the sensors with a sensing radius $R_s = 90\ m$ distributed in an $800 \times 700\ m^2$ field, the SSO algorithm was compared with the GA, PSO, and the virtual force algorithm (VFA); each algorithm was run 20 times to eliminate random errors, the population size was set in 30, and the stop criteria was set in 100 iterations. Figure 14 shows the performance of each method.

According to the authors, the SSO proves to outperform the other algorithms in terms of convergence to the optimal coverage ratio, and with the SSO algorithm the coverage ratio reaches 94%, while when using the other algorithms it only covers between 74 and 84%.

4.16. Reinforced Concrete Short Columns Design. The process related to the manufacture of materials employed by the building industry is known to be quite energy-consuming while also being a notable source of pollutants emissions such as CO_2. In particular, the optimal design of Reinforced Concrete (RC) columns in terms of both the minimization of cost and pollutant emissions is of particular interest for said industry. In 2016, Akin et al. [38] proposed an SSO-based design scheme for RC columns subjected to axial load. In said approach, a constrained design problem consisting of five decision variables is considered, namely, the height and width of the structure (h and b), the diameter of corner reinforcement bars ($\emptyset$), and the total number of reinforced bars applied for the side reinforcements (n_x and n_y). As such, each individual spider in SSO is used to represent a possible configuration for said parameters. This is

$$\mathbf{s}_i = [b, h, \emptyset, n_x, n_y] \quad (54)$$

As previously specified, the objective is to minimize both the cost and the amount of CO_2 emissions related to the manufacture of individual RC columns. For this purpose, the authors proposed the following weighted aggregate objective function:

$$f_{aggr}(\mathbf{s}_i) = \zeta_{cost} \bullet f_{cost}(\mathbf{s}_i) + \zeta_{CO2} \bullet f_{CO2}(\mathbf{s}_i) \quad (55)$$

where $f_{cost}(\mathbf{s}_i)$ and $f_{CO2}(\mathbf{s}_i)$ denote total the cost and CO_2 emissions related to a RC column designed by considering a specified design configuration (solution) $\mathbf{s}_i$, respectively, while the terms ζ_{cost} and ζ_{CO2} are weighting terms. Naturally, the value of $f_{cost}(\mathbf{s}_i)$ is related to the amounts of materials employed for the manufacture of the designed RC column, expressed as follows:

$$f_{cost}(\mathbf{s}_i) = (C_{st} \bullet W_{st} + C_c \bullet V_c) \quad (56)$$

where C_{st} and C_c denote the unit cost of steel, while W_{st} stand for the weight of steel per unit length in the designed column. Similarly, C_c represents the unit cost of concrete, whereas V_c represents the volume of concrete per unit length on the designed wall.

In the study, authors demonstrate that the SSO shows good performance finding an optimum solution for RC columns design. The algorithm also confirms to be computationally efficient and able to design low cost and low CO_2 emitting RC columns satisfying stability, safety, and material constraints.

5. Future Work with Social Spider Optimization

The SSO algorithm has attracted the attention of the research community principally for the way it divides the population in genders which gives more diversity to the operators [39]. This characteristic allows also reaching a fast convergence due to its balance between exploration and exploitation of the search space. However, several research directions can be considered as future work such as the following:

(1) It is necessary to improve the algorithm to reduce its computational cost and reduce the computation time.

(2) The adaptation of the algorithm to attack the multi-objective paradigm should be the next logical step to research.

(3) The use of data mining and machine learning technics in the algorithm is suitable to improve its performance.

(4) Continue solving real-world applications such as filter design, renewable energies, image processing, and artificial vision.

6. Conclusions

The Social Spider Optimization (SSO) proposed by Erik Cuevas et al., in 2013, is a population-based algorithm that simulates the cooperative behavior of the social spider. SSO considers two search agents (spiders): male and female. Each individual is conducted with a different set of evolutionary operators depending on its gender, which mimic different cooperative behaviors typically found in the colony. This individual categorization allows reducing critical flaws present in several SI approaches such as incorrect exploration-exploitation balance and premature convergence. After its introduction, the SSO has shown its potential to solve a wide variety of engineering problems in terms of quality and accuracy of the global solution where it outperforms other well-known algorithms which it is compared with. There are several applications and adaptations to research that made suitable the use of the algorithm in the future.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] E. Cuevas and M. Cienfuegos, "A new algorithm inspired in the behavior of the social-spider for constrained optimization,"

Expert Systems with Applications, vol. 41, no. 2, pp. 412–425, 2014.

[2] E. Bonabeau, "Social insect colonies as complex adaptive systems," *Ecosystems*, vol. 1, no. 5, pp. 437–443, 1998.

[3] E. Cuevas, M. Cienfuegos, D. Zaldívar, and M. Pérez-Cisneros, "A swarm optimization algorithm inspired in the behavior of the social-spider," *Expert Systems with Applications*, vol. 40, no. 16, pp. 6374–6384, 2013.

[4] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks (ICNN '95)*, vol. 4, pp. 1942–1948, Perth, Western Australia, November-December 1995.

[5] M. Kumar, M. Husian, N. Upreti, and D. Gupta, "Genetic Algorithm: Review and Application," *International Journal of Information Technology and Management*, vol. 2, no. 2, pp. 451–454, 2010.

[6] A. H. Gandomi, X. S. Yang, and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with Computers*, vol. 29, no. 1, pp. 17–35, 2013.

[7] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[8] X.-S. Yang, "Harmony search as a metaheuristic algorithm," *Studies in Computational Intelligence*, vol. 191, pp. 1–14, 2009.

[9] F. Grimaccia, G. Gruosso, M. Mussetta, A. Niccolai, and R. E. Zich, "Design of Tubular Permanent Magnet Generators for Vehicle Energy Harvesting by means of Social Network Optimization," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 2, pp. 1884–1892, 2017.

[10] L. A. M. Pereira, D. Rodrigues, P. B. Ribeiro, J. P. Papa, and S. A. T. Weber, "Social-spider optimization-based artificial neural networks training and its applications for Parkinson's disease identification," in *Proceedings of the 27th IEEE International Symposium on Computer-Based Medical Systems (CBMS '14)*, pp. 14–17, IEEE, New York, NY, USA, May 2014.

[11] S. Z. Mirjalili, S. Saremi, and S. M. Mirjalili, "Designing evolutionary feedforward neural networks using social spider optimization algorithm," *Neural Computing and Applications*, vol. 26, no. 8, pp. 1919–1928, 2015.

[12] D. R. Pereira, M. A. Pazoti, L. A. Pereira, and J. P. Papa, "A social-spider optimization approach for support vector machines parameters tuning," in *Proceedings of the 2014 IEEE Symposium On Swarm Intelligence (SIS)*, pp. 1–6, Orlando, FL, USA, December 2014.

[13] D. R. Pereira, M. A. Pazoti, L. A. Pereira et al., "Social-Spider Optimization-based Support Vector Machines applied for energy theft detection," *Computers & Electrical Engineering*, vol. 49, pp. 25–38, 2016.

[14] E. Cuevas, A. Luque, D. Zaldívar, and M. Pérez-Cisneros, "Evolutionary calibration of fractional fuzzy controllers," *Applied Intelligence*, vol. 47, no. 2, pp. 291–303, 2017.

[15] H. Shayeghi, A. Molaee, and A. Ghasemi, "Optimal design of fopid controller for LFC in an interconnected multi-source power system," *International Journal on "Technical and Physical Problems of Engineering"*, pp. 36–44, 2016.

[16] H. Shayeghi, A. Molaee, K. Valipour, and A. Ghasemi, "Multi-source power system FOPID based Load Frequency Control with high-penetration of Distributed Generations," in *Proceedings of the 21st Electrical Power Distribution Network Conference, EPDC 2016*, pp. 131–136, Iran, April 2016.

[17] A. A. El-Fergany and M. A. El-Hameed, "Efficient frequency controllers for autonomous two-area hybrid microgrid system using social-spider optimiser," *IET Generation, Transmission & Distribution*, vol. 11, no. 3, pp. 637–648, 2017.

[18] S. Ouadfel and A. Taleb-Ahmed, "Social spiders optimization and flower pollination algorithm for multilevel image thresholding: a performance study," *Expert Systems with Applications*, vol. 55, pp. 566–584, 2016.

[19] L. Maurya, P. K. Mahapatra, and A. Kumar, "A social spider optimized image fusion approach for contrast enhancement and brightness preservation," *Applied Soft Computing*, vol. 52, pp. 575–592, 2017.

[20] E. Cuevas, V. Osuna, and D. Oliva, *Evolutionary Computation Techniques: A Comparative Perspective*, vol. 686, Springer, 2017.

[21] A. Laddha, A. Hazra, and M. Basu, "Optimal operation of distributed renewable energy resources based micro-grid by using Social Spider Optimization," in *Proceedings of the IEEE Power, Communication and Information Technology Conference, PCITC 2015*, pp. 756–761, India, October 2015.

[22] Z. Hejrati, S. Fattahi, and I. Faraji, "Optimal congestion management using the Social Spider Optimization algorithm," in *29th International Power System Conference*, Iran, 2014.

[23] D. Vijay and V. Priya, "Anti-Islanding Protection of Distributed Generation Based on Social Spider Optimization Technique," *International Journal of Advanced Engineering Research and Science*, vol. 4, no. 6, pp. 32–40, 2017.

[24] M. Abd El Aziz and A. E. Hassanien, "An improved social spider optimization algorithm based on rough sets for solving minimum number attribute reduction problem," *Neural Computing and Applications*, pp. 1–12, 2017.

[25] C. E. Klein, E. H. Segundo, V. C. Mariani, and L. dos S. Coelho, "Modified Social-Spider Optimization Algorithm Applied to Electromagnetic Optimization," *IEEE Transactions on Magnetics*, vol. 52, no. 3, pp. 28–31, 2016.

[26] M. M. Ali, "Synthesis of the β-distribution as an aid to stochastic global optimization," *Computational Statistics & Data Analysis*, vol. 52, no. 1, pp. 133–149, 2007.

[27] U. P. Shukla and S. J. Nanda, "Parallel social spider clustering algorithm for high dimensional datasets," *Engineering Applications of Artificial Intelligence*, vol. 56, pp. 75–90, 2016.

[28] R. Zhao, Q. Luo, and Y. Zhou, "Elite opposition-based social spider optimization algorithm for global function optimization," *Algorithms*, vol. 10, no. 1, Paper No. 9, 21 pages, 2017.

[29] M. A. Tawhid and A. F. Ali, "A hybrid social spider optimization and genetic algorithm for minimizing molecular potential energy function," *Soft Computing*, vol. 21, no. 21, pp. 6499–6514, 2017.

[30] S.-C. Sun, H. Qi, Y.-T. Ren, X.-Y. Yu, and L.-M. Ruan, "Improved social spider optimization algorithms for solving inverse radiation and coupled radiation–conduction heat transfer problems," *International Communications in Heat and Mass Transfer*, vol. 87, pp. 132–146, 2017.

[31] Y. Zhou, Y. Zhou, Q. Luo, and M. Abdel-Basset, "A simplex method-based social spider optimization algorithm for clustering analysis," *Engineering Applications of Artificial Intelligence*, vol. 64, pp. 67–82, 2017.

[32] W. Spendley, G. R. Hext, and F. R. Himsworth, "Sequential application of simplex designs in optimisation and evolutionary operation," *Technometrics. A Journal of Statistics for the Physical, Chemical and Engineering Sciences*, vol. 4, pp. 441–461, 1962.

[33] T. R. Chandran, A. V. Reddy, and B. Janet, "Text clustering quality improvement using a hybrid social spider optimization," *International Journal of Applied Engineering Research*, vol. 12, no. 6, pp. 995–1008, 2017.

[34] P. Luukka, K. Saastamoinen, and V. Könönen, "A classifier based on the maximal fuzzy similarity in the generalized lukasiewicz-structure," in *Proceedings of the 10th IEEE International Conference on Fuzzy Systems*, pp. 195–198, Melbourne, Australia, December 2001.

[35] J. Dollaor, S. Chiewchanwattana, K. Sunat, and N. Muangkote, "The application of social-spider optimization for parameter improvement in the Lukasiewicz structure," in *Proceedings of the 8th International Conference on Knowledge and Smart Technology, KST 2016*, pp. 27–32, Thailand, February 2016.

[36] R. Khorramnia, M.-R. Akbarizadeh, M. K. Jahromi, S. K. Khorrami, and F. Kavusifard, "A new unscented transform for considering wind turbine uncertainty in ED problem based on SSO algorithm," *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, vol. 29, no. 4, pp. 1479–1491, 2015.

[37] Y. Zhou, R. Zhao, Q. Luo, and C. Wen, "Sensor Deployment Scheme Based on Social Spider Optimization Algorithm for Wireless Sensor Networks," *Neural Processing Letters*, pp. 1–24, 2017.

[38] A. Akin, I. Aydogdu, and T. Bilir, "Cost and CO 2 Optimization for RC Short Column Sections Subjected to Axial Load and Uniaxial/Biaxial Bending Using the Social Spider Optimization Algorithm," in *Proceedings of the Sustainable Construction Materials and Technologies SCMT*, 2016.

[39] E. Cuevas, D. Zaldívar, and M. Pérez-Cisneros, "Introduction," in *Advances in Metaheuristics Algorithms: Methods and Applications*, vol. 775 of *Studies in Computational Intelligence*, pp. 1–8, Springer International Publishing, Cham, 2018.