Hindawi Mathematical Problems in Engineering Volume 2018, Article ID 4148975, 15 pages https://doi.org/10.1155/2018/4148975



# Research Article

# **Ls-II: An Improved Locust Search Algorithm for Solving Optimization Problems**

# Octavio Camarena, Erik Cuevas D, Marco Pérez-Cisneros D, Fernando Fausto, Adrián González, and Arturo Valdivia

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

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

Received 29 March 2018; Revised 30 August 2018; Accepted 30 September 2018; Published 16 October 2018

Guest Editor: Eduardo Rodriguez-Tello

Copyright © 2018 Octavio Camarena et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited

The Locust Search (LS) algorithm is a swarm-based optimization method inspired in the natural behavior of the desert locust. LS considers the inclusion of two distinctive nature-inspired search mechanism, namely, their solitary phase and social phase operators. These interesting search schemes allow LS to overcome some of the difficulties that commonly affect other similar methods, such as premature convergence and the lack of diversity on solutions. Recently, computer vision experiments in insect tracking methods have conducted to the development of more accurate locust motion models than those produced by simple behavior observations. The most distinctive characteristic of such new models is the use of probabilities to emulate the locust decision process. In this paper, a modification to the original LS algorithm, referred to as LS-II, is proposed to better handle global optimization problems. In LS-II, the locust motion model of the original algorithm is modified incorporating the main characteristics of the new biological formulations. As a result, LS-II improves its original capacities of exploration and exploitation of the search space. In order to test its performance, the proposed LS-II method is compared against several the state-of-the-art evolutionary methods considering a set of benchmark functions and engineering problems. Experimental results demonstrate the superior performance of the proposed approach in terms of solution quality and robustness.

#### 1. Introduction

For the last few decades, optimization approaches inspired by the natural collective behavior of insects and animals have captivated the attention of many researchers. These techniques, commonly referred to as swarm optimization methods, combine deterministic rules and randomness with the purpose of mimicking some kind of natural phenomena, typically manifested in the form of a swarm behavior. Search strategies based in swarm behaviors have demonstrated to be adequate to solve complex optimization problems, often delivering significantly better solutions than those produced by traditional methods. Currently, an extensive variety of swarm-based optimization techniques can be found on the literature. Some examples include Particle Swarm Optimization (PSO), which emulates the social behavior of flocking birds or fishes [1], the Artificial Bee Colony (ABC) approach, which considers the cooperative behavior manifested in bee colonies [2], the Cuckoo Search (CS) algorithm, which simulates the brood parasitism behavior manifested by cuckoo birds [3], the Firefly Algorithm (FA), which mimics the distinctive bioluminescence-based behavior observed in fireflies [4], among others.

The Locust Search (LS) [5] algorithm is a swarm optimization approach inspired in the biological behavior of the desert locusts (*Schistocerca gregaria*). Biologically, locusts experiment two opposite phases: solitary and social. In the solitary phase, locusts avoid contact with others conspecifics in order to explore promising food sources. In opposition, in the social phase, locusts frantically aggregate around abundant foods sources (such as plantations) devastating them. This aggregation is carried on through the attraction to those elements that are found the best food sources. By integrating these two distinctive behaviors, LS maintains powerful global and local search capacities which enable it to solve effectively a wide range of complex optimization problems such as image processing [6], parameter estimation of chaotic systems [7], pattern recognition [8], among others.

In spite of its interesting characteristics, LS has some shortcomings from the evolutionary computing point of view. In its social phase operator, LS performs a series of random walks around some of the best individuals with the objective of refining its original solution. This process seems to be appropriate for the purpose of local exploitation; however, the excessive production of candidate solutions adversely increases the algorithm computational overload. Other important flaws of LS are the switch between the solitary and social phase. In LS, both behaviors are performed in the same cycle; this means that each individual search agent in LS is subject to performing both global and local movements at each iteration of the algorithms search process. Although this mechanism could be useful in some contexts, it is known that a misbalance between exploration and exploitation of solutions is able to significantly degrade any algorithm's performance, thus making this approach somewhat unreliable.

Base on this premise, in this paper, a modification to the original LS is proposed in order to better and more efficiently handle global optimization problems. In our modified approach, coined Locust Search II (LS-II), a probabilistic criterion, is introduced in order to control how the solitary and social phase operators are addressed by the algorithm, essentially allowing the swarm of locusts to "decide" when to apply each behavioral phase. Also, individual decision making based on probabilities is introduced to the algorithm's social phase operator as a mean to allow search agents to manifest an attraction toward prominent individuals within the population. Said modifications were devised with the purpose of providing a better balance between the exploration and exploitation of solutions, while also allowing an important reduction on the algorithms overall computational cost. To demonstrate the proficiency of our proposed approach, we performed a series of experiments over a set of benchmark test functions. The results of our approach are compared to those produced by the original LS, as well as some other state-of-the-art optimization techniques, including Particle Swarm Optimization (PSO) [1], Artificial Bee Colony (ABC) [2], Bat Algorithm (BA) [9], Differential Evolution (DE) [10], and Harmony Search (HS) [11]. Also, in order to enhance the analysis of our proposed method, we also performed comparative experiments over several popular engineering optimization problems, including the design of pressure vessels, gear trains [12], tension/compression springs [13], welded beams [14], three-bar truss [15], parameter estimation for FM synthesizers [16], and Optimal Capacitor Placement for Radial Distribution Networks [17, 18]. The experimental results for both sets shows that LS-II are superior not only over the original LS, but also over all other compared methods in terms of solution quality and robustness.

The rest of this paper is organized as follows: in Section 2, the original LS approach is described; in Section 3, the proposed LS algorithm modification (LS-II) is presented, and in Section 4, the experimental analysis is developed. Finally, in Section 5, our conclusions are drawn.

## 2. The Locust Search Algorithm

Locust Search (LS) [5] is a global optimization algorithm based on the gregarious behavior observed in swarms of desert locusts [19–24]. In LS, search agents are represented by a set of N individual locusts  $\mathbf{L} = \{\mathbf{l}_1, \mathbf{l}_2, \ldots, \mathbf{l}_N\}$  (with N representing the total population size), interacting with each other while moving through a n-dimensional feasible solution space. Each individual position  $\mathbf{l}_i = [l_{i,1}, l_{i,2}, \ldots, l_{i,n}]$  is defined within a bounded space  $\mathbf{S} = \{\mathbf{x} \in \mathbb{R}^n \mid lb_d \leq x_d \leq ub_d\}$  (with  $\mathbf{x} = [x_1, x_2, \ldots, x_d]$  and where  $lb_d$  and  $ub_d$  represent the lower and upper bounds at the d-th dimension, respectively) and represents a candidate solution for a specified optimization problem.

Similar to other swarm-based optimization techniques, LS comprises an iterative scheme in which search agents change their positions at each generation of the algorithm during its evolution. The change of position applied to each individual is conducted by a set of operators inspired in the two behavioral phases observed in desert locusts: solitary phase and social phase.

2.1. LS Solitary Phase. In the solitary phase, individuals move in different locations searching for promising food sources (solutions) while they avoid to aggregate with other conspecifics. This scheme is modeled by considering attraction and repulsion forces manifested among individuals within the population. Therefore, for any iteration "k", the total attraction and repulsions forces (collectively referred to as social force) experienced by a specific individual "i" are given by the following expression:

$$\mathbf{S}_{i}^{k} = \sum_{\substack{j=1\\i\neq 1}}^{N} \mathbf{s}_{ij}^{k} \tag{1}$$

where  $\mathbf{s}_{ij}^k$  denotes the pairwise attraction (or repulsion) between individual "i" and some other individuals "j" and is given by

$$\mathbf{s}_{ij}^{k} = \rho\left(\mathbf{l}_{i}^{k}, \mathbf{l}_{j}^{k}\right) s\left(r_{ij}^{k}\right) \mathbf{d}_{ij} + \text{rand}\left(1, -1\right) \tag{2}$$

where the operator  $\rho(\mathbf{l}_i^k, \mathbf{l}_j^k)$  is known as dominance value between  $\mathbf{l}_i^k$  and  $\mathbf{l}_j^k$  and the value  $s(r_{ij}^k)$  represents the so called social factor, where  $r_{ij}^k = \|\mathbf{l}_i^k - \mathbf{l}_j^k\|$  denotes the Euclidian distance between the elements "i" and "j". Therefore,  $\mathbf{d}_{ij} = (\mathbf{l}_j^k - \mathbf{l}_i^k)/r_{ij}^k$  stands for the unit vector pointing from  $\mathbf{l}_i^k$  to  $\mathbf{l}_j^k$ , while rand(1,-1) is a random vector whose elements are drawn from the uniform distribution of [-1,1]. The value  $s(r_{ij}^k)$  is given by

$$s(r_{ij}^k) = Fe^{-r_{ij}^k/L} - e^{-r_{ij}^k}$$
 (3)

where the parameters F and L denote the attraction/repulsion magnitude and length scale, respectively. To apply the operator  $\rho(\mathbf{l}_i^k, \mathbf{l}_j^k)$ , it is first assumed that each individual  $\mathbf{l}_i^k \in \mathbf{L}^k$   $(\mathbf{l}_1^k, \mathbf{l}_2^k, \dots, \mathbf{l}_N^k)$  is ranked with a number from 0 (best

individual) to N-1 (worst individual) depending on their respective fitness value. Therefore, the dominance value may be given as follows:

$$\rho\left(\mathbf{l}_{i}^{k}, \mathbf{l}_{j}^{k}\right) = \begin{cases} e^{-(\operatorname{rank}(\mathbf{l}_{i}^{k})/N)} & \text{if } \operatorname{rank}\left(\mathbf{l}_{i}^{k}\right) \leq \operatorname{rank}\left(\mathbf{l}_{j}^{k}\right) \\ e^{-(\operatorname{rank}(\mathbf{l}_{j}^{k})/N)} & \text{if } \operatorname{rank}\left(\mathbf{l}_{i}^{k}\right) > \operatorname{rank}\left(\mathbf{l}_{j}^{k}\right) \end{cases}$$
(4)

As a result of the influence of total social force  $\mathbf{S}_{i}^{k}$ , each individual "i" manifests a certain tendency to move toward (or away from) other members within the population. Under such conditions, the new position adopted by the individual "i" as a result of  $\mathbf{S}_{i}^{k}$  may be expressed as

$$\mathbf{l}_i^* = \mathbf{l}_i^k + \mathbf{S}_i^k \tag{5}$$

The result of applying the solitary movement operators to each individual  $\mathbf{l}_i^k \in \mathbf{L}^k$  is a new set of candidate solutions  $\mathbf{L}^* = \{\mathbf{l}_1^*, \mathbf{l}_2^*, \dots, \mathbf{l}_N^*\}$ , which represent the positions taken by each individual as a result of the influence exerted by all other members in the swarm.

2.2. LS Social Phase. The social phase operation is applied to refine some of the best candidate solutions  $\mathbf{L}^* = \{\mathbf{l}_1^*, \mathbf{l}_2^*, \dots, \mathbf{l}_N^*\}$  generated by applying the solitary phase movement operator described in Section 2.1. For this purpose, a subset of solutions  $\mathbf{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_q\}$ , comprised by the q best solutions within the set  $\mathbf{L}^*$ , is first defined. Then, for each solution  $\mathbf{l}_i^* \in \mathbf{B}$ , a set of h random solutions  $\mathbf{M}^i = \{\mathbf{m}_1^i, \dots, \mathbf{m}_h^i\}$  is generated within a corresponding subspace  $\mathbf{C}_i \in \mathbf{S}$ , whose limits are given by

$$C_{i,n}^{\text{lower}} = b_{i,n} - r$$

$$C_{i,n}^{\text{upper}} = b_{i,n} + r$$
(6)

where  $C_{i,n}^{\mathrm{lower}}$  and  $C_{i,n}^{\mathrm{upper}}$  represent the upper and lower bounds of each subspace  $C_i$  at the n-th dimension, respectively, while  $b_{i,n}$  stands for the n-th element (decision variable) from solution  $\mathbf{b}_i$ , and where

$$r = \frac{\sum_{n=1}^{d} \left( b_n^{\text{upper}} - b_n^{\text{lower}} \right)}{d} \bullet \beta$$
 (7)

with  $b_d^{\text{lower}}$  and  $b_d^{\text{upper}}$  denoting the lower and upper bounds at the n-th dimension, respectively, whereas d stands for total number of decision variables. Furthermore,  $\beta \in [0,1]$  represents a scalar factor which modulates the size of  $\mathbf{C}_i$ .

Finally, the best solution from  $\mathbf{l}_i^* \in \mathbf{B}$  and its respective h random solutions  $(\mathbf{m}_1^i, \mathbf{m}_2^i, \dots, \mathbf{m}_h^i)$  is assigned as the position for individual "i" at the following iteration "k + 1". That is,

$$\mathbf{l}_{i}^{k+1} = \operatorname{best}\left(\mathbf{l}_{i}^{*}, \mathbf{m}_{1}^{i}, \mathbf{m}_{2}^{i}, \dots, \mathbf{m}_{h}^{i}\right)$$
(8)

It is worth nothing that any solution  $\mathbf{l}_i^*$  that is not grouped within the set of best solutions **B** is excluded by the social phase operator. As such, the final position update applied

to each individual 'i' within the whole swarm may be summarized as follows:

$$\mathbf{l}_{i}^{k+1} = \begin{cases} \operatorname{best}\left(\mathbf{l}_{i}^{*}, \mathbf{m}_{1}^{i}, \mathbf{m}_{2}^{i}, \dots, \mathbf{m}_{h}^{i}\right) & \text{if } \mathbf{l}_{i}^{*} \in \mathbf{B} \\ \mathbf{l}_{i}^{*} & \text{if } \mathbf{l}_{i}^{*} \notin \mathbf{B} \end{cases}$$
(9)

# 3. The LS-II Algorithm

The LS algorithm proposes an interesting global and local search capacities in the form of the so called solitary and social phase operators. In spite of its interesting characteristics, LS has some shortcomings from an evolutionary computing point of view. Specifically, in its social phase operator, LS performs a series of random walks around some of the best individuals with the objective of refining their current solutions; while this process seems to be appropriate for the purpose of local exploitation, excessive evaluations of candidate solutions are known to adversely increase the algorithm computational load, a phenomenon that is undesirable to virtually any optimization approach. Another important flaw of the original LS approach is given by the way in which the solitary and social phase behavioral operators are addressed. In LS, both the solitary and social phases operators are coded to be performed in sequence at each iteration of the algorithms search process; this means that each individual is subject to undergone to both global and local operations without considering its individual decision. Although this mechanism could be useful in some contexts, it produces a misbalance between exploration and exploitation which could significantly degrade its performance.

In this paper a modification to the original LS algorithm, referred to as LS-II, is proposed. In our approach, instead of applying both the solitary and social phase operators in the same iteration, we propose a simple probabilistic scheme to selectively apply either one of this two behaviors at each iteration of the algorithms search process. Furthermore, we have also proposed some changes the original LS's social operators: essentially, instead of performing a series of local evaluations around promising solutions, LS-II's social operator incorporates a probabilistic selection mechanism that allows locusts to be attracted toward some other prominent individuals within the population. This modification not only allows to reduce the algorithms overall computational cost, but also enhances LS-II's convergence speed.

3.1. Selecting between Solitary and Social Phases. The first modification, considered in the original LS, involves the selection for any given iteration between the solitary phase and the social phase. In the proposed LS-II approach, we assume that, at the beginning of the search process, individuals have a tendency to perform intensive exploration over the feasible search space by means of their solitary phase behavior, whereas in later stages of the evolution process exploitation is favored by the social phase behavior. Also, it is considered that the individuals maintain a certain probability to behave as either solitary or social depending on the current stage (iteration) of the algorithm's search process. Therefore, as the evolutionary process progresses, the probability that

an element behaves in a social manner increases, while the probability that it acts in solitary behavior, is reduced. Under such circumstances, at each iteration "k", the behavioral phase  $P(\mathbf{L}^k)$  applied to the population  $\mathbf{L}^k$  is chosen as follows:

$$P\left(\mathbf{L}^{k}\right) = \begin{cases} \mathbf{Solitary} & \text{if } rand \leq p^{k} \\ \mathbf{Social} & \text{if } rand > p^{k} \end{cases}$$
 (10)

where *rand* denotes a random number sampled within the uniformly distributed interval [0,1], while value  $p^k$ , called behavior probability, is given by

$$p^k = 1 - \frac{k}{itern} \tag{11}$$

where *itern* stands for the maximum number of iterations considered in the search process. From the expression, it is clear that, as the iterative process increases, the value of the behavior probability  $p^k$  is linearly decreased. Intuitively, other monotonically decreasing models (such as logarithmic or exponential functions) could also be considered to handle this value.

It is important to remark that from both operators (solitary and social) only the social phase has been modified, and the solitary phase behavior proposed by the original LS remains unchanged. In Figure 1, we show a comparison between the original LS algorithm and the proposed LS-II approach.

3.2. Modified Social Phase Operator. The second modification considers several changes to the original LS algorithm's social phase, each aimed at giving every individual  $\mathbf{l}_{i}^{k}$  the ability to move toward potentially good solution within the feasible solution space instead of performing a series of local function evaluations. This mechanism does not only allow taking advantage of the information provided by all individuals within the population, but also allows a significant reduction in computational load by taking away excessive and unnecessary function evaluations. Similarly to the original LS approach, the LS-II's social phase starts by selecting a subset of solutions  $\mathbf{B}^k = \{\mathbf{b}_1^k, \dots, \mathbf{b}_q^k\}$  that include the q best solutions from the total set  $\mathbf{L}^k=\{\mathbf{l}_1^k,\mathbf{l}_2^k,\dots\mathbf{l}_N^k\}$  (therefore,  $\mathbf{B}^k\subseteq$  $\mathbf{L}^{k}$ ). However, instead of defining a set of random candidate solutions around each position  $\mathbf{b}_{i}^{k} \subseteq \mathbf{B}^{k}$  (as in the original LS), we propose a movement scheme that allows each individual  $\mathbf{l}_{i}^{\kappa}$  to be attracted toward the direction of a randomly selected solution  $\mathbf{b}_{j}^{k}$ . For this purpose, to each solution  $\mathbf{b}_{j}^{k}$  is assigned to a probability of being selected by a given individual  $\mathbf{l}_{i}^{k}$ , which depends on both the quality of  $\mathbf{b}_{i}^{k}$  and the distance separating both individuals ( $\mathbf{l}_{i}^{k}$  and  $\mathbf{b}_{i}^{k}$ ) as follows:

$$P_{\mathbf{l}_{i}\mathbf{b}_{j}}^{k} = \frac{A\left(\mathbf{b}_{j}^{k}\right)e^{-\|\mathbf{l}_{i}^{k}-\mathbf{b}_{j}^{k}\|}}{\sum_{n=1}^{q}A\left(\mathbf{b}_{n}^{k}\right)e^{-\|\mathbf{l}_{i}^{k}-\mathbf{b}_{n}^{k}\|}}$$
(12)

where  $\|\mathbf{l}_i^k - \mathbf{b}_j^k\|$  denotes the Euclidian distance between the individual "i"  $(\mathbf{l}_i^k)$  and a member "j" from the set of best

solutions  $\mathbf{B}^k$  ( $\mathbf{b}_j^k$ ), while  $A(\mathbf{b}_j^k)$  stands for the attractiveness of solution  $\mathbf{b}_i^k$  as given by

$$A\left(\mathbf{b}_{j}^{k}\right) = \frac{f\left(\mathbf{b}_{j}^{k}\right) - f_{\text{worst}}\left(\mathbf{B}^{k}\right)}{f_{\text{best}}\left(\mathbf{B}^{k}\right) - f_{\text{worst}}\left(\mathbf{B}^{k}\right) + \varepsilon}$$
(13)

where  $f(\mathbf{b}_j^k)$  denotes the fitness value (quality) related to  $\mathbf{b}_j^k$ , while  $f_{\text{best}}(\mathbf{B}^k)$   $f_{\text{worst}}(\mathbf{B}^k)$  stand for the best and worst fitness value from among the members within the set of best solutions  $\mathbf{B}^k$ . Finally,  $\varepsilon$  corresponds to a small value (typically between  $1.0 \times 10^{-4}$  and  $1.0 \times 10^{-5}$ ) used to prevent a division by cero.

With the previous being said, under LS-II's social phase behavior, each individual  $\mathbf{l}_i^k$  within the swarm population  $\mathbf{L}^k$  updates their position as follows:

$$\mathbf{l}_{i}^{k+1} = \mathbf{l}_{i}^{k} + 2\left(\mathbf{b}_{r}^{k} - \mathbf{l}_{i}^{k}\right) \bullet rand \tag{14}$$

where  $\mathbf{b}_r^k$  (with  $r \in [1, ..., q]$ ) is a randomly chosen solution  $\mathbf{b}_j^k \in \mathbf{B}^k$ , selected by applying the roulette selection method [25] with regard to their respective probabilities  $P_{ij}^k$  (relative to  $\mathbf{l}_i^k$  and  $\mathbf{b}_j^k$ ), while *rand* stands for a random number drawn from within the uniformly distributed interval [0, 1].

# 4. Experiments and Results

4.1. Benchmark Test Functions. In our experiments, we studied LS-II's performance with regard to the optimization of 13 well-known benchmark tests functions (see Appendix A) operated in 30 dimensions. Furthermore, we have also compared out proposed method's result against those produced by other popular optimization techniques, including Particle Swarm Optimization (PSO) [1], Artificial Bee Colony (ABC) [2], Bat Algorithm (BA) [9], Differential Evolution (DE) [10], Harmony Search (HS) [11], and the original Locust Search (LS) [5].

The parameters setting for each of the methods has been performed by using the automatic algorithm configuration software known as iRace [26]. This method allows obtaining the best possible parameter configuration under a specific set of benchmark problems. After the iRace analysis, the following settings have been found:

- (1) LS: the algorithm's parameters were set to q=10, h=3 [5].
- (2) PSO: the cognitive and social coefficients are set to  $c_1 = 1.9460$  and  $c_2 = 1.8180$ , respectively. Also, the inertia weight factor  $\omega$  is set to decrease linearly from 0.543254 to 0.33362 as the search process evolves [1].
- (3) ABC: the algorithm was implemented by setting the parameter *limit* = num Of Food Sources \* dims, where num Of FoodSources = N (population size) and dims = n (dimensionality of the solution space) [2]. The code has been obtained from [27] which corresponds to the optimized version (v2).

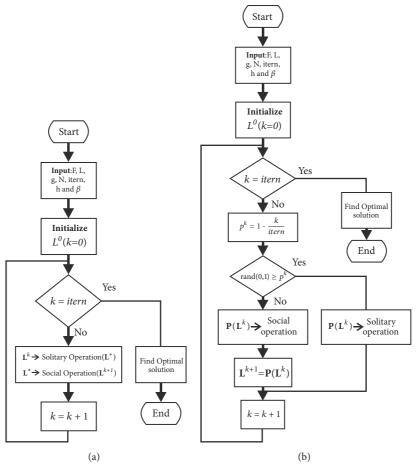


FIGURE 1: Flowchart of (a) LS and (b) LS-II.

- (4) BA: the parameters were set as follows: initial loudness rate A=0.311895, pulse emission rate r=0.70767, minimum frequency  $f_{\rm min}=0.3272$ , and maximum frequency  $f_{\rm max}=1.65815$ , respectively [9].
- (5) DE: the crossover rate is set to CR = 0.58143, while the differential weight is given as F = 0.12470 [10].
- (6) HS: the Harmony Memory Considering Rate is set to HCMR = 0.98950 while the pitch adjustment rate is set to PA = 0.812174 [11].
- (7) LS-II: the number of best individuals considered for the social phase operator is set to q = 10.

The code for each algorithm has been obtained from their original sources. In the analysis, all of the compared methods were tested by considering a population size of N=50 individuals and a maximum number of iterations *itern=1000* as a stop criterion. Said tests setup was considered in order to keep consistency with other similar works currently found in the literature. All experiments were performed on MatLAB® R2016a, running on a computer with an Intel® Core™ i7-3.40 GHz processor, and Windows 8 (64-bit, 8GB of memory) as operating system.

Our experimental setup aims to compare the performance of LS-II against those of PSO, ABC, BA, DE, HS, and LS. The averaged experimental results, corresponding to 30 individual runs, are reported in Table 1 where the best outcomes for each test function are boldfaced. The results reported in this paper consider the following performance indexes: Average Best fitness (AB), Median Best fitness (MB), and the standard deviation of the best finesses (SD). As shown by our experiments, LS-II demonstrates to be superior not only to the original LS approach, but also to all other of the compared methods. Such a notorious performance is related to both the excellent global search capabilities provided LS-II's solitary phase operators, the highly effective exploitation mechanism of our modified social phase operators, and naturally the better trade-off between exploration and exploitation that is achieved through the selective application of said behavioral phases. Furthermore, Wilcoxon's rank sum test for independent samples [28] was conducted over the best fitness values found by each of the compared methods on 30 independent test runs (30 executions by each set). Table 2 reports the *p*-values produced by said test for the pairwise comparison over two independent sets of samples (LS-II versus LS, ABC, BA, DE, and HS), by considering a 5% significance level. For the Wilcoxon test, a null hypothesis assumes that there is a significant difference between the

TABLE 1: Minimization results for the benchmark test functions illustrated in Appendix A operated in 30 dimensions. Results were averaged from 30 individual runs, each by considering population size of N = 50 and maximum number of iterations itern = 1000.

1 1		Ogu	DE	Cav	JIL	DA	31	11 31
		FSO	DE	ABC	ПЭ	DA	LS	L3-11
	$f_{ m mean}$	$4.05 \times 10^{-13}$	$1.30 \times 10^{-02}$	$2.44 \times 10^{-11}$	$1.50 \times 10^{-04}$	$4.48 \times 10^{00}$	$4.32 \times 10^{-04}$	$4.33 \times 10^{-32}$
$f_1$	fmedian	$2.99 \times 10^{-13}$	$1.25 \times 10^{-02}$	$7.07 \times 10^{-12}$	$1.53 \times 10^{-04}$	$4.43 \times 10^{00}$	$2.17 \times 10^{-04}$	$1.02{\times}10^{-33}$
	$f_{ m std}$	$5.65 \times 10^{-13}$	$4.86 \times 10^{-03}$	$4.66 \times 10^{-11}$	$1.61 \times 10^{-05}$	$8.88 \times 10^{-01}$	$6.52 \times 10^{-04}$	$9.30{ imes}10^{-32}$
	fmean	$8.54 \times 10^{01}$	$1.54 \times 10^{00}$	$7.80 \times 10^{-07}$	$7.71 \times 10^{-01}$	$3.23 \times 10^{28}$	$5.12 \times 10^{-02}$	$2.06{\times}10^{-16}$
$f_2$	$f_{ m median}$	$8.94{\times}10^{01}$	$5.17 \times 10^{-02}$	$7.43 \times 10^{-07}$	$7.19 \times 10^{-01}$	$4.44 \times 10^{25}$	$5.50{ imes}10^{-02}$	$1.69{ imes}10^{-16}$
	$f_{ m std}$	$4.95 \times 10^{01}$	$3.37 \times 10^{00}$	$3.11 \times 10^{-07}$	$1.66 \times 10^{-01}$	$7.16 \times 10^{28}$	$1.71 \times 10^{-02}$	$2.13{\times}10^{\text{-}16}$
	$f_{mean}$	$3.31 \times 10^{-17}$	$2.67 \times 10^{02}$	$1.23 \times 10^{04}$	$9.86 \times 10^{00}$	$2.51 \times 10^{06}$	$1.82 \times 10^{-03}$	$3.86{ imes}10^{-30}$
$f_3$	fmedian	$2.30 \times 10^{-18}$	$1.12 \times 10^{02}$	$1.25 \times 10^{04}$	$7.30 \times 10^{00}$	$2.71 \times 10^{06}$	$1.55 \times 10^{-03}$	$1.28{ imes}10^{-31}$
	$f_{ m std}$	$5.23 \times 10^{-17}$	$3.47 \times 10^{02}$	$2.57 \times 10^{03}$	$5.26 \times 10^{00}$	$1.12 \times 10^{06}$	$1.38 \times 10^{-03}$	$8.39{ imes}10^{-30}$
	fmean	$2.02 \times 10^{00}$	$3.75 \times 10^{01}$	$2.15 \times 10^{01}$	$8.16 \times 10^{-01}$	$3.70 \times 10^{01}$	$1.05 \times 10^{-02}$	$7.43 \times 10^{-17}$
$f_4$	$f_{ m median}$	$1.61 \times 10^{00}$	$3.86 \times 10^{01}$	$2.17 \times 10^{01}$	$6.57 \times 10^{-01}$	$3.80 \times 10^{01}$	$1.15 \times 10^{-02}$	$5.12{ imes}10^{-17}$
	$f_{ m std}$	$1.28 \times 10^{00}$	$6.43 \times 10^{00}$	$4.25 \times 10^{00}$	$2.42 \times 10^{-01}$	$2.57 \times 10^{00}$	$6.63 \times 10^{-03}$	$7.49{ imes}10^{-17}$
	fmean	$7.90 \times 10^{01}$	$2.05 \times 10^{01}$	$2.48 \times 10^{01}$	$9.76 \times 10^{00}$	$6.33 \times 10^{01}$	$4.27 \times 10^{02}$	$2.89 \times 10^{01}$
$f_5$	$f_{ m median}$	$5.87 \times 10^{01}$	$1.52 \times 10^{01}$	$2.49{\times}10^{01}$	$1.02 \times 10^{01}$	$6.07 \times 10^{01}$	$3.74 \times 10^{02}$	$2.90 \times 10^{01}$
	$f_{ m std}$	$3.68 \times 10^{01}$	$1.22{\times}10^{01}$	$1.24 \times 10^{00}$	$1.69 \times 10^{00}$	$1.52 \times 10^{01}$	$2.81 \times 10^{02}$	$4.15{\times}10^{-02}$
	$f_{ m mean}$	$2.40 \times 10^{00}$	$6.94 \times 10^{00}$	$2.17{ imes}10^{-11}$	$0.00{\times}10^{00}$	$9.23 \times 10^{03}$	$5.58 \times 10^{-02}$	$6.06 \times 10^{00}$
$f_6$	$f_{ m median}$	$1.00 \times 10^{00}$	$3.90 \times 10^{00}$	$1.58{ imes}10^{-11}$	$0.00 \times 10^{00}$	$9.53 \times 10^{03}$	$5.67 \times 10^{-02}$	$5.89{\times}10^{00}$
	$f_{ m std}$	$2.41 \times 10^{00}$	$7.74 \times 10^{00}$	$1.76 \times 10^{-11}$	$0.00{\times}10^{00}$	$1.95 \times 10^{03}$	$1.07 \times 10^{-02}$	$5.16 \times 10^{-01}$
	$f_{ m mean}$	$9.31 \times 10^{-03}$	$2.47 \times 10^{05}$	$1.07 \times 10^{-01}$	$2.96 \times 10^{01}$	$5.54 \times 10^{09}$	$2.21 \times 10^{-02}$	$8.77{ imes}10^{-03}$
$f_7$	$f_{ m median}$	$8.91 \times 10^{-03}$	$1.32 \times 10^{05}$	$1.03 \times 10^{-01}$	$3.13 \times 10^{01}$	$6.25 \times 10^{09}$	$1.90 \times 10^{-02}$	$2.18{\times}10^{-03}$
	$f_{ m std}$	$9.53 \times 10^{-03}$	$4.47 \times 10^{05}$	$2.58 \times 10^{-02}$	$3.56 \times 10^{00}$	$2.17 \times 10^{09}$	$1.12 \times 10^{-02}$	$9.65{\times}10^{-03}$
	$f_{ m mean}$	$-9.24 \times 10^{03}$	$-1.16 \times 10^{04}$	$-1.26 \times 10^{04}$	$-1.26{ imes}10^{04}$	$-1.89 \times 10^{199}$	$-3.80 \times 10^{02}$	$-3.45\times10^{03}$
$f_8$	$f_{ m median}$	$-9.26\times10^{03}$	$-1.15 \times 10^{04}$	$-1.26 \times 10^{04}$	$-1.26{ imes}10^{04}$	$-3.19 \times 10^{195}$	$-3.69 \times 10^{02}$	$-3.32\times10^{03}$
	$f_{ m std}$	$4.45 \times 10^{02}$	$3.68 \times 10^{02}$	$1.35 \times 10^{02}$	$4.49{ imes}10^{-02}$	$6.55 \times 10^{04}$	$9.22 \times 10^{-02}$	$3.62\times10^{-02}$
	fmean	$2.82 \times 10^{-07}$	$4.17 \times 10^{01}$	$2.68 \times 10^{-01}$	$1.25 \times 10^{00}$	$4.74 \times 10^{02}$	$2.69 \times 10^{-02}$	$0.00{ imes}10^{00}$
$f_9$	$f_{ m median}$	$1.82 \times 10^{-08}$	$2.32 \times 10^{00}$	$4.12 \times 10^{-04}$	$1.28 \times 10^{00}$	$2.31 \times 10^{02}$	$3.44 \times 10^{-03}$	$0.00{\times}10^{00}$
	$f_{ m std}$	$5.26 \times 10^{-07}$	$8.73 \times 10^{01}$	$4.32{\times}10^{-01}$	$2.80 \times 10^{-01}$	$5.43 \times 10^{02}$	$4.00 \times 10^{-02}$	$0.00{\times}10^{00}$
	$f_{ m mean}$	$1.43 \times 10^{00}$	$2.84 \times 10^{00}$	$1.49 \times 10^{-05}$	$5.62 \times 10^{-02}$	$1.28 \times 10^{01}$	$3.00 \times 10^{-02}$	$4.44 \times 10^{-15}$
$f_{10}$	$f_{ m median}$	$1.50 \times 10^{00}$	$2.96 \times 10^{00}$	$1.28 \times 10^{-05}$	$5.80 \times 10^{-02}$	$1.22 \times 10^{01}$	$1.73 \times 10^{-02}$	$4.44 \times 10^{-15}$
	$f_{ m std}$	$3.86 \times 10^{-01}$	$7.21 \times 10^{-01}$	$9.97 \times 10^{-06}$	$5.65 \times 10^{-03}$	$1.17 \times 10^{00}$	$3.56 \times 10^{-02}$	$0.00{\times}10^{00}$
	$f_{ m mean}$	$1.52 \times 10^{-02}$	$1.71 \times 10^{00}$	$9.66 \times 10^{-04}$	$1.01 \times 10^{-01}$	$1.07 \times 10^{02}$	$1.42 \times 10^{-04}$	$0.00 \times 10^{00}$
$f_{11}$	$f_{ m median}$	$8.88 \times 10^{-16}$	$1.08 \times 10^{00}$	$1.32 \times 10^{-08}$	$9.57 \times 10^{-02}$	$1.14 \times 10^{02}$	$1.49 \times 10^{-05}$	$0.00{\times}10^{00}$
	$f_{ m std}$	$2.10 \times 10^{-02}$	$2.01 \times 10^{00}$	$3.92 \times 10^{-03}$	$2.74 \times 10^{-02}$	$2.42 \times 10^{01}$	$3.69 \times 10^{-04}$	$0.00{\times}10^{00}$
	$f_{ m mean}$	$8.22{\times}10^{01}$	$6.98 \times 10^{04}$	$1.23{\times}10^{\text{-}12}$	$8.20 \times 10^{01}$	$2.24 \times 10^{07}$	$5.08 \times 10^{-03}$	$1.34 \times 10^{00}$
$f_{12}$	$f_{ m median}$	$8.16 \times 10^{01}$	$3.53 \times 10^{04}$	$6.46{ imes}10^{-13}$	$8.22 \times 10^{01}$	$2.53 \times 10^{07}$	$5.12 \times 10^{-03}$	$1.32{\times}10^{00}$
	$f_{ m std}$	$4.06 \times 10^{00}$	$8.22 \times 10^{04}$	$3.09{ imes}10^{-12}$	$2.64 \times 10^{00}$	$7.32 \times 10^{06}$	$4.17 \times 10^{-04}$	$1.34{\times}10^{-01}$
	$f_{ m mean}$	$1.02 \times 10^{02}$	$5.08 \times 10^{04}$	$2.27{ imes}10^{-11}$	$8.73 \times 10^{01}$	$9.92 \times 10^{06}$	$1.08 \times 10^{-01}$	$3.15 \times 10^{00}$
$f_{13}$	$f_{ m median}$	$1.00 \times 10^{02}$	$9.69 \times 10^{03}$	$8.53{\times}10^{-12}$	$8.91 \times 10^{01}$	$9.08 \times 10^{06}$	$1.05 \times 10^{-01}$	$3.00 \times 10^{00}$
	$f_{ m std}$	$3.77 \times 10^{00}$	$7.78 \times 10^{04}$	3.13×10 <sup>-11</sup>	$5.92 \times 10^{00}$	$2.53 \times 10^{06}$	$1.37 \times 10^{-02}$	$2.69 \times 10^{-01}$

Table 2: Wilcoxon's test comparison for LS-II versus PSO, De, ABC, HS, BA, and LS. The table shows the resulting p-values for each pairwise comparison.

Function	LS-II vs PSO	LS-II vs DE	LS-II vs ABC	LS-II vs HS	LS-II vs BA	LS-II vs LS
f	$6.54 \times 10^{-15}$	$1.11 \times 10^{-12}$	$3.01 \times 10^{-11}$	$1.12 \times 10^{-13}$	$4.36 \times 10^{-13}$	$4.86 \times 10^{-12}$
f	$1.57 \times 10^{-14}$	$1.52 \times 10^{-15}$	$8.48 \times 10^{-09}$	$9.65 \times 10^{-14}$		$2.02 \times 10^{-13}$
f3	$5.47 \times 10^{-12}$	$4.33 \times 10^{-10}$	$3.01 \times 10^{-11}$	$7.56 \times 10^{-10}$		$2.89 \times 10^{-11}$
f A	$3.65 \times 10^{-13}$	$4.88{ imes}10^{-10}$	$3.01 \times 10^{-11}$	$6.19 \times 10^{-10}$	$3.45 \times 10^{-10}$	$9.11 \times 10^{-12}$
£ 5	$6.14 \times 10^{-12}$	$6.95 \times 10^{-10}$	$8.18 \times 10^{-01}$	$9.12 \times 10^{-12}$	$7.25 \times 10^{-10}$	$1.38 \times 10^{-11}$
f	$6.24 \times 10^{-11}$	$2.45 \times 10^{-10}$	$3.01 \times 10^{-11}$	$4.54 \times 10^{-12}$	$8.34 \times 10^{-10}$	$1.84 \times 10^{-11}$
f <sub>7</sub>	$8.53 \times 10^{-10}$	$7.88 \times 10^{-10}$	$3.01 \times 10^{-11}$	$6.37 \times 10^{-12}$	$8.15 \times 10^{-12}$	$2.31 \times 10^{-11}$
£ 2	$2.70 \times 10^{-15}$	$6.31 \times 10^{-11}$	$3.01 \times 10^{-11}$	$1.14 \times 10^{-12}$	$7.77 \times 10^{-13}$	$2.77 \times 10^{-11}$
fg	$9.81 \times 10^{-11}$	$4.69 \times 10^{-11}$	$1.21 \times 10^{-12}$	$1.54 \times 10^{-13}$	$3.25 \times 10^{-13}$	$3.24 \times 10^{-11}$
$f_{10}$	$5.64 \times 10^{-13}$	$2.55 \times 10^{-11}$	$3.02 \times 10^{-11}$	$7.18 \times 10^{-13}$	$4.12 \times 10^{-10}$	$3.71 \times 10^{-11}$
$f_{11}$	$9.71 \times 10^{-12}$	$3.78 \times 10^{-12}$	$1.21 \times 10^{-12}$	$4.51 \times 10^{-13}$	$3.03 \times 10^{-10}$	$4.17 \times 10^{-11}$
$f_{12}$	$4.32 \times 10^{-12}$	$1.99 \times 10^{-14}$	$3.02 \times 10^{-11}$	$8.4710^{-13}$	$7.21 \times 10^{-11}$	$4.64 \times 10^{-11}$
$f_{13}$	$2.21 \times 10^{-12}$	$1.19 \times 10^{-12}$	$3.02 \times 10^{-11}$	$9.10 \times 10^{-12}$	$3.32 \times 10^{-11}$	$5.10 \times 10^{-11}$

Table 3: Minimization results for the engineering optimization problems described in Appendix B. Results were obtained from 30 individual runs, each by considering population size of N = 50 and maximum number of iterations *itern* = 1000.

		PSO	DE	ABC	HS	BA	LS	LS-II
	$J_{\rm worst}$	7302.944	6590.18	6497.539	9702.135	400595	28910.08	6059.73
$J_1$	$J_{ m best}$	6876.679	6877.065	6138.001	6523.004	6683.271	6882.137	6059.72
$J_1$	$J_{\rm avg}$	6876.679	6897.291	6311.334	7210.724	103291.8	7986.061	6059.72
	$J_{\mathrm{std}}$	$5.61 \times 10^{02}$	$1.33 \times 10^{02}$	$7.84 \times 10^{01}$	$6.82 \times 10^{02}$	$1.20 \times 10^{05}$	$5.87 \times 10^{03}$	4.30×10-03
	$J_{ m worst}$	$1.17 \times 10^{-01}$	$1.18 \times 10^{-02}$	3.85×10 <sup>-02</sup>	3.82×10 <sup>-02</sup>	$5.14 \times 10^{-02}$	$6.47 \times 10^{-02}$	9.61×10 <sup>-03</sup>
T	$J_{ m best}$	$1.61 \times 10^{-02}$	$8.13 \times 10^{-03}$	$2.50 \times 10^{-02}$	$2.70 \times 10^{-02}$	$7.99 \times 10^{-02}$	$7.27 \times 10^{-03}$	$5.39 \times 10^{-03}$
$J_2$	$J_{\rm avg}$	$1.17 \times 10^{-02}$	$7.96 \times 10^{-03}$	$2.21 \times 10^{-02}$	$1.09 \times 10^{-02}$	$5.52 \times 10^{-02}$	$7.03 \times 10^{-02}$	$6.74 \times 10^{-03}$
	$J_{ m std}$	$4.43 \times 10^{05}$	$7.94 \times 10^{-04}$	$1.31 \times 10^{-18}$	$4.79 \times 10^{05}$	$2.40 \times 10^{05}$	$1.16 \times 10^{-03}$	$1.03 \times 10^{-03}$
	$J_{ m worst}$	1.9851	1.7511	1.7498	1.8012	1.7955	1.7449	1.72485
T	$J_{ m best}$	1.7421	1.725	1.7249	1.7248	1.7249	1.7277	1.72485
$J_3$	$J_{\rm avg}$	1.7578	1.7298	1.7251	1.7562	1.7252	1.7311	1.72485
	$J_{ m std}$	$7.41 \times 10^{-03}$	$9.87 \times 10^{-02}$	$3.50 \times 10^{-04}$	$4.40 \times 10^{-04}$	$2.50 \times 10^{-03}$	$8.50 \times 10^{-01}$	$7.16 \times 10^{-07}$
	$J_{ m worst}$	347.763	347.2367	346.4102	347.5836	300.9182	320.2528	279.936
$J_4$	$J_{ m best}$	346.4523	346.4538	346.4102	345.3665	284.377	301.3875	279.725
14	$J_{\rm avg}$	346.675	346.7444	346.4102	346.345	286.777	308.2089	279.743
	$J_{\mathrm{std}}$	$2.90 \times 10^{-01}$	$1.98 \times 10^{-01}$	$5.74 \times 10^{-14}$	$1.98 \times 10^{-01}$	$3.44 \times 10^{00}$	$6.68 \times 10^{-01}$	$4.12 \times 10^{-02}$
	$J_{ m worst}$	28.51646	26.27407	27.60109	28.81973	30.08068	28.34164	25.7868
$J_5$	$J_{ m best}$	25.5127	24.56246	22.85514	23.59037	22.32354	21.05672	20.1103
<b>J</b> 5	$J_{\rm avg}$	26.15824	25.82232	24.45296	25.59217	27.00966	25.42715	23.898
	$J_{\mathrm{std}}$	2.19	6.46	3.50	1.72	2.14	2.55	1.56
	$J_{ m worst}$	$1.35 \times 10^{-06}$	$9.09 \times 10^{-07}$	$4.70 \times 10^{-07}$	$1.79 \times 10^{-06}$	$3.13\times10^{-08}$	$1.57 \times 10^{-02}$	3.96×10 <sup>-08</sup>
T	$J_{ m best}$	$2.22 \times 10^{-06}$	$1.78 \times 10^{-09}$	$1.33 \times 10^{-10}$	$2.67 \times 10^{-09}$	$8.90 \times 10^{-10}$	$4.45 \times 10^{-07}$	$5.10 \times 10^{-13}$
$J_6$	$J_{\rm avg}$	$7.17 \times 10^{-03}$	$5.73 \times 10^{-03}$	$4.30 \times 10^{-03}$	$8.60 \times 10^{-03}$	$2.87 \times 10^{-03}$	$1.43 \times 10^{-03}$	$1.27 \times 10^{-10}$
	$J_{ m std}$	$2.06 \times 10^{-03}$	$1.65 \times 10^{-03}$	$1.23 \times 10^{-03}$	$2.47 \times 10^{-03}$	$8.23 \times 10^{-05}$	$4.12\times10^{-03}$	$7.77 \times 10^{-09}$
	$J_{ m worst}$	$1.58 \times 10^{05}$	1.11×10 <sup>05</sup>	$8.66 \times 10^{04}$	$1.07 \times 10^{05}$	$2.60 \times 10^{05}$	$8.53 \times 10^{04}$	$8.44 \times 10^{04}$
T	$J_{best}$	$8.49 \times 10^{04}$	$8.37 \times 10^{04}$	$8.37 \times 10^{04}$	$8.45 \times 10^{04}$	$1.12 \times 10^{05}$	$8.37 \times 10^{04}$	$8.27 \times 10^{04}$
$J_7$	$J_{\rm avg}$	$1.05 \times 10^{05}$	$8.95 \times 10^{04}$	$8.43 \times 10^{04}$	$9.35 \times 10^{04}$	$1.26 \times 10^{05}$	$8.42 \times 10^{04}$	$8.33 \times 10^{04}$
	$J_{std}$	$1.58 \times 10^{04}$	$8.06 \times 10^{03}$	$8.30 \times 10^{02}$	$4.84 \times 10^{03}$	$2.64 \times 10^{04}$	$4.85 \times 10^{02}$	$4.80 \times 10^{02}$

mean values of two algorithms. On the other hand, an alternative hypothesis (rejection of the null hypothesis) suggests that the difference between the mean values of both compared methods is insignificant. As shown by all of the *p*-values reported in Table 3, enough evidence is provided to reject the null hypothesis (all values are less than 0.05, and as such satisfy the 5% significance level criteria). This proves the statistical significant of our proposed method's results and thus discard the possibility of them being a product of coincidence.

The analysis of the final fitness values cannot absolutely characterize the capacities of an optimization algorithm. Consequently, a convergence experiment on the seven compared algorithms has been conducted. The objective of this test is to evaluate the velocity with which an optimization scheme reaches the optimum. In the experiment, the performance of each algorithm is analyzed over a representative set of functions, including  $f_1$ ,  $f_3$ ,  $f_9$ , and  $f_{11}$ . 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 final result. Figure 2 presents the fitness evolution of each compared method for functions  $f_1$ ,  $f_3$ ,  $f_9$ , and  $f_{11}$ . According to Figure 2, the LS-II method presents the overall best convergence when compared to the other compared methods; once again, this performance is testimony of LS-II's better trade-off between diversification and intensification, as well as the decision-making scheme incorporated to the algorithms social phase operators.

4.2. Engineering Optimization Problems. In order to further evaluate the performance of our proposed approach, we tested LS-II over several well-studied engineering design optimization problems, namely, the design of pressure vessels, gear trains [12], tension/compression springs [13] welded beams [14], three-bar trusses [15], parameter estimation for FM synthesizers [16], and Optimal Capacitor Placement for Radial Distribution Networks [17, 18]. Appendix B provides a detailed description of each engineering problem used in this study.

Similarly to the experiments reported in Section 4.1, we compared LS-II's performance over the previously mentioned

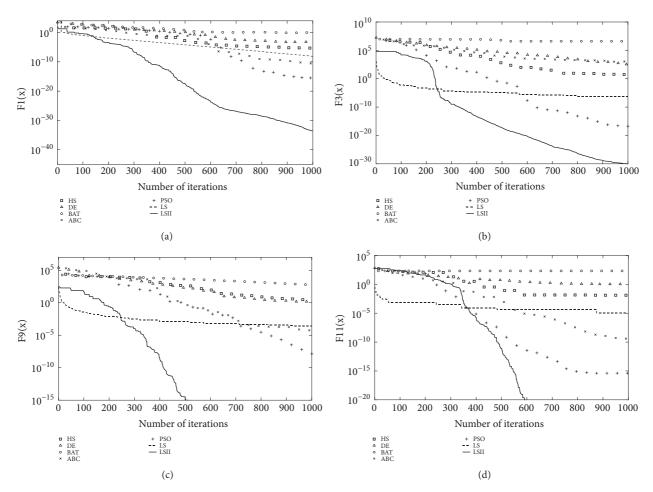


FIGURE 2: Evolution curves of the run which represents the median final result. (a)  $f_1$ , (b)  $f_3$ , (c)  $f_9$ , and (d)  $f_{11}$  represent the convergence graphs obtained by applying HS, DE, BA, ABC, PSO, LS, and LS-II. For each experiment, a population size of N = 50 and maximum number of iterations *itern* = 1000 have been considered.

real problems with those of Particle Swarm Optimization (PSO) [1], Artificial Bee Colony (ABC) [2], Bat Algorithm (BA) [9], Differential evolution (DE) [10], Harmony Search (HS) [11], and the original Locust Search (LS) [5]. For each design problem a penalty function is implemented to penalize the fitness value of solutions that infringe any of the restrictions modeled on each optimization task. The penalty implemented of the fitness function may be illustrated by the following expression:

$$J_i^* \left( \mathbf{x} \right) = J_i \left( \mathbf{x} \right) + P \tag{15}$$

where  $J_i(\mathbf{x})$  is one of the seven fitness functions  $(i \in 1,...,7)$  defined by each engineering problem in terms of a given solution " $\mathbf{x}$ ", where P is a penalty value whose magnitude is set to  $1.0 \times 10^5$  [29].

All experiments described in this section have been performed by considering the same parameter setup and settings described in Section 4.1. In Table 3, the experimental results corresponding to each of the compared methods are shown. All of the considered engineering optimization problems were tested a total of 30 times per method. The performance indexes shown in said table include the best and

worst fitness value found over all 30 individual runs ( $J_{\text{best}}$  and  $J_{\text{worst}}$ , respectively), as well as the average and standard deviation of all fitness values found over said each set of experiments ( $J_{\text{avg}}$  and  $J_{\text{std}}$ , respectively). For each individual problem, the best outcomes from among all of the compared methods are boldfaced.

From the results of Table 3, LS-II proves to be the superior choice when applied to handle each of the chosen engineering design problems. Naturally, this is easily attributed to LS-II's to the effective exploration and exploitation mechanism provided by the selective use of both solitary and social phase operators. Finally, Tables 4 and 5 report the best set of design variables (solutions) found by LS-II for each of the considered engineering optimization problems.

# 5. Conclusions

Locust Search (LS) is a swarm-based optimization method inspired in the natural behavior of the desert locust [5]. When it was first proposed, the LS method considered as basis the biological models extracted from observations of locust behaviors. In these models, the procedure to change from

Problem	$x_1$	$x_2$	$x_3$	$x_4$	<i>x</i> <sub>5</sub>	$x_6$	$J_i(\mathbf{x})$
$J_1$	0.812500214	0.437500122	42.09844537	176.636599	-	-	6059.71623
$J_2$	40	25	13	56	-	-	$5.10463 \times 10^{-13}$
$J_3$	0.050664	0.329604	4.372104	-	-	-	0.0053911
$J_4$	0.205730336	3.470474331	9.0366238	0.205729651	-	-	1.724851989
$J_5$	0.8696133	0.2169257	$-7.2121 \times 10^{-06}$	-1.70011996	-0.29988725	-	279.725029
$J_6$	0.419345	2.422595	4.930669	2.403819	4.247489	4.90245	20.11029

Table 4: Best parameter configurations obtained by LS-II for each of the engineering optimization problems illustrated in Appendices B.1-B.6.

Table 5: Best parameter configurations obtained by LS-II for the Optimal Capacitor Placement for the IEEE's 69-Bus Radial Distribution Networks illustrated in Appendix B.7.

Problem	$x_1 - x_{15}$	$x_{16}$	$x_{17} - x_{59}$	$x_{60}$	$x_{61} - x_{69}$	$J_i(\mathbf{x})$
$\overline{J_7}$	0	350	0	1200	0	83708.21

the solitary phase to the gregarious phase, together with the mechanism to select the element to which a locust will be attracted, is fixed and does not consider the decision of each locust in the behavioral process.

Currently, new computer vision experiments in insect tracking methods have conducted to the development of more accurate locust motion models than those produced by simple behavior observations. The most distinctive characteristic of such new models is the use of probabilities to emulate the locust decision process.

In this paper, a modification to the original LS algorithm, referred to as LS-II, has been proposed to better handle global optimization problems. In LS-II, the locust motion model of the original algorithm is modified incorporating the main characteristics of the new biological formulations. Under the new model, probability decisions are considered to emulate the behavior of the locust colony. Therefore, the decisions, to change of behavioral phase and to select the attraction elements, are probabilistically taken.

The proposed LS-II method has been compared against several state-of-the-art evolutionary methods considering a set of benchmark functions and engineering problems. Experimental results demonstrate the superior performance of the proposed approach in terms of solution quality and robustness.

#### **Appendix**

#### A.

In Table 1, all of the 13 benchmark test functions that were implemented in our experiments are shown.

See Table 6.

#### B.

In this section, we offer a detailed description for all of the engineering optimization problems considered in our experiments.

B.1. Pressure Vessel Design Problem. As described in [12], the pressure vessel design problem consists on a constrained

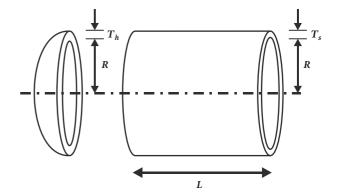


FIGURE 3: Pressure vessel's design variables.

optimization problem in which the main objective is to minimize the cost of said artifact. This is usually achieved by defining an appropriate set of four design variables  $\mathbf{x} = [x_1, x_2, x_3, x_4]$ , namely, the shell's thickness  $T_s$  ( $x_1$ ), head's thickness  $T_h$  ( $x_2$ ), vessel's internal radius R ( $x_3$ ), and vessel's length L ( $x_4$ ) (see Figure 3). With that being said, such cost minimization problem may be more formally expressed as follows.

Problem B.1 (pressure vessel design).

Minimize: 
$$J_1(\mathbf{x})$$

$$= 0.6224x_1x_2x_4 + 1.7781x_2x_3^2$$

$$+ 3.1661x_1^2x_4 + 19.84x_1^2x_3$$
Subject to:  $g_1(\mathbf{x}) = -x_1 + 0.0193x_3 \le 0$ 

$$g_2(\mathbf{x}) = -x_2 + 0.00954x_3 \le 0$$

$$g_3(\mathbf{x})$$

$$= -\pi x_3^2 x_4 - \left(\frac{4}{3}\right)\pi x_3^2 + 1,296,000 \le 0$$

$$g_4(\mathbf{x}) = x_4 - 240 \le 0$$

$$0 \le x_i \le 100, \quad i = 1,2$$

$$0 \le x_i \le 200, \quad i = 3,4$$

TABLE 6: Test functions used for our experiments. In the table, **S** indicates the subset of  $\mathbb{R}^n$  which comprises the function's search space and n indicates the function's dimension. Also, the value  $f(\mathbf{x}^*)$  indicates the optimum value of each function, while  $\mathbf{x}^*$  indicates the optimal solution.

) -> f	1		ı	
$f_i$	Name	Function	S	и
$f_1$	Sphere	$f(x) = \sum_{i=1}^{n} x_i^2  x^* = (0, \dots, 0); \ f(x^*) = 0$	$[-100, 100]^n$	30
$f_2$	Schewefel 2.22	$f\left(x ight) = \sum_{i=1}^{n} \left x_{i}\right  + \prod_{i=1}^{n} \left x_{i}\right   x^{*} = (0, \dots, 0); \ f\left(x^{*}\right) = 0$	$[-10, 10]^n$	30
$f_3$	Schewefel 1.2	$f(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} x_{j}\right)^{2} x^{*} = (0,, 0);  f(x^{*}) = 0$	$[-100, 100]^n$	30
$f_4$	Schewefel 2.22	$f(x) = \max\{ x , 1 \le i \le n\}$ $x^* = (0,, 0); \ f(x^*) = 0$	$[-100, 100]^n$	30
$f_5$	Rosenbrock	$f\left(x ight) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + \left(x_i - 1\right)^2\right]  x^* = (0, \dots, 0); \ f\left(x^*\right) = 0$	$[-30,30]^n$	30
$f_6$	Step	$f\left(x ight) = \sum_{i=1}^{n} \left(\left x_{i}+0.5\right \right)^{2}  x^{*} = (0, \dots, 0) \; ; \; f\left(x^{*}\right) = 0$	$[-100, 100]^n$	30
$f_7$	Quartic	$f(x) = \sum_{i=1}^{n} ix_i^4 + \text{rand}(0,1)  x^* = (0,\dots,0); \ f(x^*) = 0$	$[-1.28, 128]^n$	30
$f_8$	Schewefel	$f(x) = \sum_{i=1}^{n-1} -x_i \sin\left(\sqrt{ x_i }\right)  x^* = (420, \dots, 420); \ f\left(x^*\right) = -418.9829en$	$[-500, 500]^n$	30
$f_9$	Rastring	$f(x) = \sum_{i=1}^{n} \left[ x^2 - 10\cos(2\pi x_i) + 10 \right]  x^* = (0, \dots, 0); \ f(x^*) = 0$	$[-5.12, 5.12]^n$	30
$f_{10}$	Ackley's	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_{i}^{2}}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos\left(2\pi x_{i}\right)\right) + 20 + e  x^{*} = (0, \dots, 0); \ f\left(x^{*}\right) = 0$	$[-32,32]^n$	30
$f_{11}$	Grienwank	$f\left(x\right) = \frac{1}{4000} \sum_{i=1}^{n} x_{i}^{2} - \prod_{i=1}^{n} \cos\left(\frac{x_{i}}{\sqrt{i}}\right) + 1  x^{*} = (0, \dots, 0); \ f\left(x^{*}\right) = 0$	$[-600,600]^n$	30
$f_{12}$	Penalized	$f(x) = \frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n} (y_i - 1)^2 \left[ 1 + 10 \sin^2(\pi y_{i+1}) \right] + (y_n - 1)^2 \right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$	$[-50, 50]^n$	30
		$y_i = 1 + \frac{x_i + 1}{4} \ u(x_i, a, k, m) = \begin{cases} n(x_i - a) & x_i > a \\ 0 & -a < x_i < a \ x^* = (0,, 0); \ f(x^*) = 0 \end{cases}$ $k(-x_i - a)^m \ x_i < -a$		
$f_{13}$	Penalized 2	$f(x) = 0.1 \left\{ \sin^2 \left( 3\pi x_1 \right) + \sum_{i=1}^{n} \left( x_i - 1 \right)^2 \left[ 1 + \sin^2 \left( 3\pi x_i \right) \right] + \left( x_n - 1 \right)^2 \left[ 1 + \sin^2 \left( 2\pi x_n \right) \right] \right\} + \sum_{i=1}^{n} u\left( x_i, 5, 100, 4 \right) \left[ k(x_i - a)^m - x_i > a \right] $	[-50, 50]"	30
		$u(x_i, a, k, m) = \begin{cases} 0 & -a < x_i < a \ x^* = (0,, 0); \ f(x^*) = 0 \end{cases}$		
		$\left[ k(-x_i - a)^m  x_i < -a \right]$		

*B.2. Gear Train Design Problem.* The optimization problem related to the design of gear trains (as the one shown in Figure 4) is to minimize the squared difference between the gear's teeth ratio and some scalar value (1/6.931 in this case) [12]. As such, the decision variables comprise the number of teeth on each of the gears A, B, D, and F. By considering  $x_1 = A$ ,  $x_2 = B$ ,  $x_3 = D$ , and  $x_4 = F$ , the gear train design problem may then be formulated as follows.

Problem B.2 (gear train design).

Minimize: 
$$J_2(\mathbf{x}) = \left( \left( \frac{1}{6.931} \right) - \left( \frac{x_3 x_2}{x_1 x_4} \right) \right)^2$$
 (B.2)

Subject to:  $0 \le x_i \le 600$ , i = 1, 2, 3, 4

B.3. Tension/Compression Spring Design Problem. For this design problem the objective is to minimize the tension (or compression) experiment by a spring when subjected to some load P (see Figure 5). This is usually achieved by optimizing with regard to three decision variables: the wire diameter  $d(x_1)$ , the coil diameter  $D(x_2)$ , and the number of active coils  $n(x_3)$ . As such, the problem may be formulated as follows.

Problem B.3 (tension/compression spring design).

Minimize: 
$$J_3(\mathbf{x}) = (x_3 + 2) x_2 x_1^2$$
  
Subject to:  $g_1(\mathbf{x}) = 1 - \frac{x_2^3 x_3}{71,785 x_1^4} \le 0$   
 $g_2(\mathbf{x})$   

$$= \frac{4x_2^2 - x_1 x_2}{12,566 (x_2 x_1^3 - x_1^4)} + \frac{1}{5,108 x_1^2} - 1$$

$$\le 0$$

$$g_3(\mathbf{x}) = 1 - \frac{140.45 x_1}{x_2^2 + x_3} \le 0$$

$$g_4(\mathbf{x}) = \frac{x_1 + x_2}{1.5} \le 0$$

$$0.5 \le x_1 \le 2$$

$$0.25 \le x_2 \le 1.3$$

$$2 \le x_3 \le 15$$

B.4. Three-Bar Truss Design Problem. The objective behind the design of a symmetric three-bar truss (as shown in Figure 6) is to minimize its fabrication cost while also allowing it to sustain a certain static load P and some stress  $\sigma$ . In this case, the design variables are comprised by the cross-sectional areas  $x_1$ ,  $x_2$ , and  $x_3$  corresponding to each of the truss's three bars. By considering  $x_3 = x_1$  (due to symmetry on the truss design), the optimization problem may be then defined as follows.

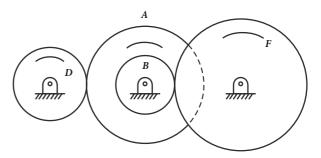


FIGURE 4: Gear train design variables.

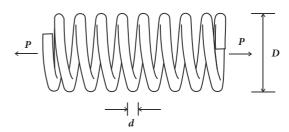


FIGURE 5: The tension/compression spring design variables.

Problem B.4 (three-bar truss design).

Minimize: 
$$J_4(\mathbf{x}) = l\left(2\sqrt{2x_1} + x_2\right)$$
  
Subject to:  $g_1(\mathbf{x}) = \frac{\sqrt{2x_1} + x_2}{\sqrt{2x_1^2 + 2x_1x_2}}P - \sigma \le 0$   
 $g_2(\mathbf{x}) = \frac{x_2}{\sqrt{2x_1^2 + 2x_1x_2}}P - \sigma \le 0$   
 $g_3(\mathbf{x}) = \frac{1}{\sqrt{2x_2^2 + x_1}}P - \sigma \le 0$  (B.4)  
 $0 \le x_i \le 1, \quad i = 1, 2$   
 $l = 100 \text{ cm},$   
 $P = 2 \text{ kN/cm}^2,$   
 $\sigma = 2 \text{ kN/cm}^2$ 

B.5. Welded Beam Design Problem. For this design problem, the aim is to minimize the fabrication cost of a beam meant to be welded to another surface. Said welded beam must be designed so that it is able to sustain certain amounts of shear stress  $(\tau)$ , bending stress  $(\sigma)$ , bucking load  $(P_c)$ , and deflection  $(\delta)$ . The design variables implied on the design of such a mechanism include the weld's thickness and length h and l, respectively, as well as the beam's width and thickness t and b, respectively (see Figure 7). By considering  $x_1 = h$ ,  $x_2 = l$ ,  $x_3 = t$ , and  $x_4 = b$ , this design problem may then be formulated as follows.

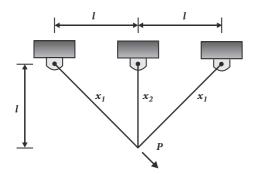


FIGURE 6: Three-bar truss design variables.

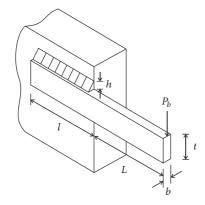


FIGURE 7: Welded beam design variables.

Problem B.5 (welded beam design).

Minimize: 
$$J_{5}(\mathbf{x})$$
  
=  $1.10471x_{1}^{2}x_{2}$   
+  $0.04811x_{3}x_{4}(14 + x_{2})$   
Subject to:  $g_{1}(\mathbf{x}) = \tau(\mathbf{x}) - \tau_{\max} \ge 0$   
 $g_{2}(\mathbf{x}) = \sigma(\mathbf{x}) - \sigma_{\max} \le 0$   
 $g_{3}(\mathbf{x}) = x_{1} - x_{4} \le 0$   
 $g_{4}(\mathbf{x})$   
=  $0.1047x_{1}^{2} + 0.04811x_{3}x_{4}(14 + x_{2})$   
-  $5 \le 0$   
 $g_{5}(\mathbf{x}) = 0.125 - x_{1} \le 0$   
 $g_{6}(\mathbf{x}) = \delta(\mathbf{x}) - \delta_{\max} \le 0$   
 $g_{7}(\mathbf{x}) = P - P_{c}(\mathbf{x})$   
 $0.1 \le x_{i} \le 2, \quad i = 1, 4$ 

 $0.1 \le x_i \le 10, \quad i = 2, 3$ 

$$\sigma(\mathbf{x}) = \frac{6PL}{x_4 x_3},$$

$$P_c(\mathbf{x})$$

$$= \frac{4.013E\sqrt{x_3^2 x_4^6/36}}{L^2} \left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$$P = 6000 \text{ lb},$$

$$L = 14 \text{ in},$$

$$E = 30e10^6 \text{ psi},$$

$$G = 12e10^6 \text{ psi}$$

$$\tau_{\text{max}} = 13,600 \text{ psi},$$

$$\sigma_{\text{max}} = 30,000 \text{ psi},$$

$$\delta_{\text{max}} = 0.25 \text{ in}$$
(B.5)

B.6. Parameter Estimation for FM Synthesizers. An FM synthesizer is a device purposed to generate a sound (signal)  $y(\mathbf{x},t)$  such that it is similar to some other target sounds  $y_0(t)$ . As such, a FM synthesizer's parameter estimator aims to provide a set of decision variables  $\mathbf{x} = [x_1 = a_1, x_2 = \omega_1, x_3 = a_2, x_4 = \omega_2, x_5 = a_3, x_6 = \omega_3]$  (with  $a_i$  and  $\omega_i$  denoting a finite wave amplitude and frequency, respectively) such that the error between signals  $y(\mathbf{x},t)$  and  $y_0(t)$  is minimized [16]. Under this considerations, said parameter estimation problem may be described as follows.

*Problem B.6* (parameter estimation for FM synthesizers).

Minimize: 
$$J_{6}(\mathbf{x}) = \sum_{t=0}^{100} (y(\mathbf{x}, t) - y_{0}(t))^{2}$$

$$y(\mathbf{x}, t)$$

$$= x_{1} \sin (x_{2}\theta t) + x_{3} \sin (x_{4}\theta t)$$

$$+ x_{5} \sin (x_{6}\theta t)$$
Subject to: 
$$-6.4 \le x_{i} \le 6.35, \quad i = 1, 2, 3, 4, 5, 6$$

$$0.1 \le x_{i} \le 10, \quad i = 2, 3$$

$$\theta = \frac{2\pi}{100}$$

B.7. Optimal Capacitor Placement for the IEEE's 69-Bus Radial Distribution Networks. The Optimal Capacitor Placement (OCP) problem [17, 18] is a complex combinatorial optimization formulation in which the objective is to determine the number, location, type, and size of shunt capacitors that are to be placed in a Radial Distribution Network (RDN). In the problem, the main objective function involves the minimization of the network's total operation cost in terms of a certain amount of reactive compensation of the RDN [17].

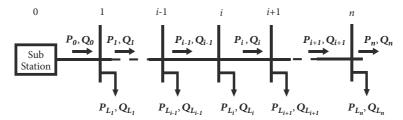


FIGURE 8: Optimal Capacitor Placement Design Variables.

In this problem, a RDN of 69 buses (which implies n = 69 dimensions) is assumed as it is illustrated in Figure 8. Under this architecture, the model can be characterized as follows.

*Problem B.7* (optimal capacitor placement for radial distribution networks).

Minimize: 
$$J_7(\mathbf{x}) = K_p t P_{\text{Loss}} + K_{ic} N_q + \sum_{j=1}^n x_j k_j$$
  
Subject to:  $x_j \in \{0, 150, 360, \dots, 3900, 4050\}$  (B.7)
$$\sum_{i=1}^{n_c} x_j \leq \sum_{i=1}^n x_i$$

In the model,  $K_p$  represents the equivalent cost per unit of power loss (\$/kW) and  $P_{Loss}$  represents the total power loss of the distribution system (kW).  $Q_j$  corresponds to the size of the shunt capacitor (kVAR) installed at bus "j".  $k_j$  symbolizes the cost per unit of reactive power (\$/kVAR). t is the operation time in hours,  $K_{ic}$  is the installation cost (\$), and  $N_q$  is the number of compensation capacitors that are required to be installed in the distribution network. In the formulation, it is considered that  $\mathbf{x} = [x_1 = Q_1, x_2 = Q_2, \ldots, x_n = Q_n]$  and  $n_c$  is the number of buses required to be compensated.

## **Data Availability**

The data used to support the findings of this study are available from the corresponding author upon request.

#### **Conflicts of Interest**

The authors declare that there are no conflicts of interest regarding the publication of this paper.

#### References

- [1] 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, IEEE, 1995.
- [2] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical Report-TR06, 2005.
- [3] 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.

- [4] X.-S. Yang, "Firefly algorithms for multimodal optimization," in Stochastic Algorithms: Foundations and Applications, vol. 5792 of Lecture Notes in Computer Science, pp. 169–178, Springer, 2009.
- [5] E. Cuevas, A. González, D. Zaldívar, and M. Pérez-Cisneros, "An optimisation algorithm based on the behaviour of locust swarms," *International Journal of Bio-Inspired Computation*, vol. 7, no. 6, pp. 402–407, 2015.
- [6] E. Cuevas, A. González, F. Fausto, D. Zaldívar, and M. Pérez-Cisneros, "Multithreshold Segmentation by Using an Algorithm Based on the Behavior of Locust Swarms," *Mathematical Problems in Engineering*, vol. 2015, Article ID 805357, 25 pages, 2015.
- [7] E. Cuevas, J. Gálvez, and O. Avalos, "Parameter estimation for chaotic fractional systems by using the locust search algorithm," *Computación y Sistemas*, vol. 21, no. 2, 2017.
- [8] A. González, E. Cuevas, F. Fausto, A. Valdivia, and R. Rojas, "A template matching approach based on the behavior of swarms of locust," *Applied Intelligence*, vol. 47, no. 4, pp. 1087–1098, 2017.
- [9] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), J. R. Gonzalez, D. A. Pelta, C. Cruz, G. Terrazas, and N. Krasnogor, Eds., vol. 284 of Studies in Computational Intelligence, pp. 65–74, Springer, Berlin, Germany, 2010.
- [10] R. Storn and K. Price, "Differential Evolution -A simple and efficient adaptive scheme for global optimization over continuous spaces," 1995.
- [11] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.
- [12] E. Sandgren, "Nonlinear Integer and Discrete Programming in Mechanical Design Optimization," *Journal of Mechanical Design*, vol. 112, no. 2, p. 223, 1990.
- [13] J. S. Arora, Introduction to Optimum Design, Academic Press, New York, NY, USA, 1989.
- [14] S. S. Rao, Engineering Optimization?: Theory and Practice, John Wiley & Sons, 2009.
- [15] J. Koski, "Defectiveness of weighting method in multicriterion optimization of structures," *Communications in Numerical Methods in Engineering*, vol. 1, no. 6, pp. 333–337, 1985.
- [16] S. Das and P. N. Suganthan, "Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems," 2010.
- [17] F. M. F. Flaih, X. Lin, S. M. Dawoud, and M. A. Mohammed, "Distribution system reconfiguration for power loss minimization and voltage profile improvement using Modified particle swarm optimization," in *Proceedings of the 2016 IEEE PES Asia Pacific Power and Energy Engineering Conference*, (APPEEC '16), pp. 120–124, 2016.

- [18] P. Díaz, M. Pérez-Cisneros, E. Cuevas, O. Camarena, F. Fausto, and A. Gonzalez, "A swarm approach for improving voltage profiles and reduce power loss on electrical distribution networks," *IEEE Access*, vol. 6, p. 1, 2018.
- [19] C. M. Topaz, A. J. Bernoff, S. Logan, and W. Toolson, "A model for rolling swarms of locusts," *The European Physical Journal Special Topics*, vol. 157, no. 1, pp. 93–109, 2008.
- [20] C. M. Topaz, M. R. D'Orsogna, L. Edelstein-Keshet, and A. J. Bernoff, "Locust dynamics: behavioral phase change and swarming," *PLoS Computational Biology*, vol. 8, no. 8, Article ID e1002642, 2012.
- [21] W. Stower, "Photographic techniques for the analysis of locust "hopper" behaviour," *Animal Behaviour*, vol. 11, no. 1, pp. 198–205, 1963.
- [22] J. Buhl, D. J. T. Sumpter, I. D. Couzin et al., "From disorder to order in marching locusts," *Science*, vol. 312, no. 5778, pp. 1402–1406, 2006.
- [23] G. Ariel, Y. Ophir, S. Levi, E. Ben-Jacob, and A. Ayali, "Individual pause-and-go motion is instrumental to the formation and maintenance of swarms of marching locust nymphs," *PLoS ONE*, vol. 9, no. 7, 2014.
- [24] G. Ariel, A. Ayali, and F. R. Adler, "Locust Collective Motion and Its Modeling," *PLoS Computational Biology*, vol. 11, no. 12, p. e1004522, 2015.
- [25] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, 1989.
- [26] M. López-Ibáñez, J. Dubois-Lacoste, L. Pérez Cáceres, M. Birattari, and T. Stützle, "The irace package: iterated racing for automatic algorithm configuration," *Operations Research Perspectives*, vol. 3, pp. 43–58, 2016.
- [27] https://abc.erciyes.edu.tr/.
- [28] E. A. Gehan, "A generalized Wilcoxon test for comparing arbitrarily singly-censored samples," *Biometrika*, vol. 52, pp. 203–223, 1965.
- [29] M.-A. Díaz-Cortés, E. Cuevas, and R. Rojas, Intelligent Systems Reference Library 129 Engineering Applications of Soft Computing, Springer Verlag, 2017.

















Submit your manuscripts at www.hindawi.com







