Optimal Power Flow Analysis using Lévy Flight Spider Monkey Optimization Algorithm

Abstract: Optimal power flow (OPF) is the most requisite tool in the power system analysis. The OPF is relatively a difficult constrained optimization problem and broadly solved by conventional as well as modern intelligent methods. In this paper the authors proposed a levy flight spider monkey optimization (LFSMO) algorithm to solve the standard OPF problem for IEEE 30-bus system, which is proposed to improve the exploitation capability of spider monkey optimization (SMO) algorithm. The performance of LFSMO is evaluated by testing it over 25 benchmark functions.

Keywords: Spider Monkey Optimization; Swarm Intelligence; Lévy Flight Local Search; Optimal Power Flow

1 Introduction

The fundamentals of optimal power flow (OPF) problem analysis provides optimum utilization of electrical power to produce economic and secure operating conditions for planning and operation of power system. In power systems, to optimize the objective function like fuel cost, the power flow equations and operational limits are considered as equality constraints while limits on control variables are considered as inequality constraints on system dependent variables. In OPF problem, generator real power, bus voltage, reactive power for shunt volt-ampere reactive (VAR) compensation, and transformer tap setting are the control variables, while load bus voltage, transmission line loadings, and generators reactive powers are system dependent variables.

The OPF problem, in general, is a non-convex, highly constrained nonlinear, large-scale optimization problem. The OPF problem was first introduced by Carpentier in 1962 (Carpentier, 1962, 1979) and formulated by Dommel et. al. (Dommel & Tinney, 1968). Since then it has been an area of interest for many researchers. The different OPF problems have been worked out by applying various mathematical techniques such as Newton-based method (Sun, Ashley, Brewer, Hughes, & Tinney, 1984), linear programming (Mota-Palomino & Quintana, 1986; Al-Muhawesh & Qamber, 2008), non-linear programming (Habibollahzadeh, Luo, & Semlyen, 1989), and interior point method (Granville, 1994; Wei, Sasaki, Kubokawa, & Yokoyama, 1998). While applying the classical mathematical techniques, the fuel cost characteristic of a generating unit is assumed to be smooth and convex functions. These methods are sensitive towards initial solutions and may fail due to initial improper values of variables. The practical power systems are difficult to solve by these classical mathematical techniques due to their nonlinear characteristics of prohibited operating zones, valve-point effects, and piecewise quadratic cost functions. Therefore, an efficient strategy is highly required to deal the non-convex, non-linear, and multi-modular power system problems.

Researchers analyzed the efficiency of the nature inspired algorithms (NIA) to deal with such kind of problems and have applied many NIA strategies like genetic algorithm (GA) (Bakirtzis, Biskas, Zoumas, & Petridis, 2002), particle swarm optimization (PSO) (Abido, 2002a), tabu search (TS) (Abido, 2002b), evolutionary programming (EP) (Ongsakul & Tantimaporn, 2006; Yuryevich & Wong, 1999), differential evolution (DE) (Sayah & Zehar, 2008), simulated annealing (SA) (Roa-Sepulveda & Pavez-Lazo, 2003), and improved meta-heuristics (Reddy & Bijwe, 2016). The reported results showed significant improvement in the accuracy as compared to the classical mathematical techniques. Therefore, it is suggested to apply the meta-heuristic techniques to solve the complex OPF problem for which deterministic mathematical technologies are not able to produce desired results.

Bansal et. al. (Bansal, Sharma, Jadon, & Clerc, 2014) introduced a well-tested swarm intelligence based algorithm, namely spider monkey optimization (SMO) algorithm. The SMO algorithm relies on the food foraging behavior of spider monkeys (SMs). The reported results stated that the SMO algorithm performs better than the artificial bee colony (ABC), DE (DE/rand/bin/1), PSO (PSO-2011), and covariance matrix adaptation evolution strategies (CMA-ES) algorithms.

Further, through experiments and literature, SMO is presented as an efficient algorithm for solving the multimodal problems due to its structured swarm based search strategy. The SMO algorithm has shown its efficiency to solve real world problems of power system like capacitor placement (A. Sharma, Sharma, Bhargava, Sharma, & Bansal, 2016) and lower order system modeling (A. Sharma, Sharma, Bhargava, & Sharma, 2016).

Therefore, in this paper, SMO algorithm is applied to solve a multimodal OPF (IEEE-30 bus) problem. Though SMO achieves efficient solutions but due to the presence of a random component in the position update process, there is always a chance to get skip of the real solution. Therefore, incorporation of local search strategy may improve the exploitation capability and may reduce the risk of skipping the real solution. So, to enhance the exploitation capability of the SMO algorithm, a local search strategy, namely lévy flight local search (LFLS) (H. Sharma, Bansal, Arya, & Yang, 2015) is incorporated with SMO. The developed approach is named as a lévy flight spider monkey optimization (LFSMO) algorithm. The performance of the proposed strategy is evaluated by testing it on 25 well-known benchmark functions. This set of functions includes unimodal, multimodal, separable, and nonseparable complex functions. Further, the proposed LFSMO is also applied to solve the OPF problem, and the reported results are compared with PSO, DE, ABC and other state-of-art methods available in the literature.

The paper is organised as shown in Figure 1 as follows: Basic SMO is explained in Section 2. Section 3, describes a brief review of local search strategies. LFLS is described in Section 4. In Section 5, LFLS is incorporated with SMO. The performance of LFSMO is evaluated in Section 6. The OPF problem is explained in Section 7. In Section, 8 SMO and LFSMO are applied to solve OPF problem. Finally, in Section 9, paper is concluded.

2 Spider Monkey Optimization (SMO) Algorithm

In the field of swarm intelligence based algorithms, SMO algorithm is a new approach, developed by Bansal et. al. (Bansal, Sharma, et al., 2014). SMO algorithm relies on the foraging behavior and social structure of SMs. SMs have been categorized as fission fusion social structure (FFSS) based animals, in which individuals form small impermanent parties whose member belongs to a larger community. Monkeys split themselves from larger to smaller groups and vice versa based on scarcity and availability of food.

2.1 Main steps of SMO algorithm

The SMO algorithm consists of six phases: Local Leader phase, Global Leader phase, Local Leader Learning phase, Global Leader Learning phase, Local Leader Decision phase and Global Leader Decision phase. Each of the phases is explained as follows:

2.1.1 Initialization of the Population

Initially, SMO generates an equally distributed initial population of N, SMs where each monkey SM_i (i = 1, 2, ...,N) is a D-dimensional vector and SM_i represents the i^{th} SM in the population. SM represents the potential solution of the problem under consideration. Each SM_i is initialized as follows:

$$SM_{ij} = SM_{minj} + U(0,1) \times (SM_{maxj} - SM_{minj}) \tag{1}$$

where SM_{minj} and SM_{maxj} are respectively lower and upper bounds of SM_i in j^{th} direction and $U(0,1)$ is a uniformly distributed random number in the range [0, 1].

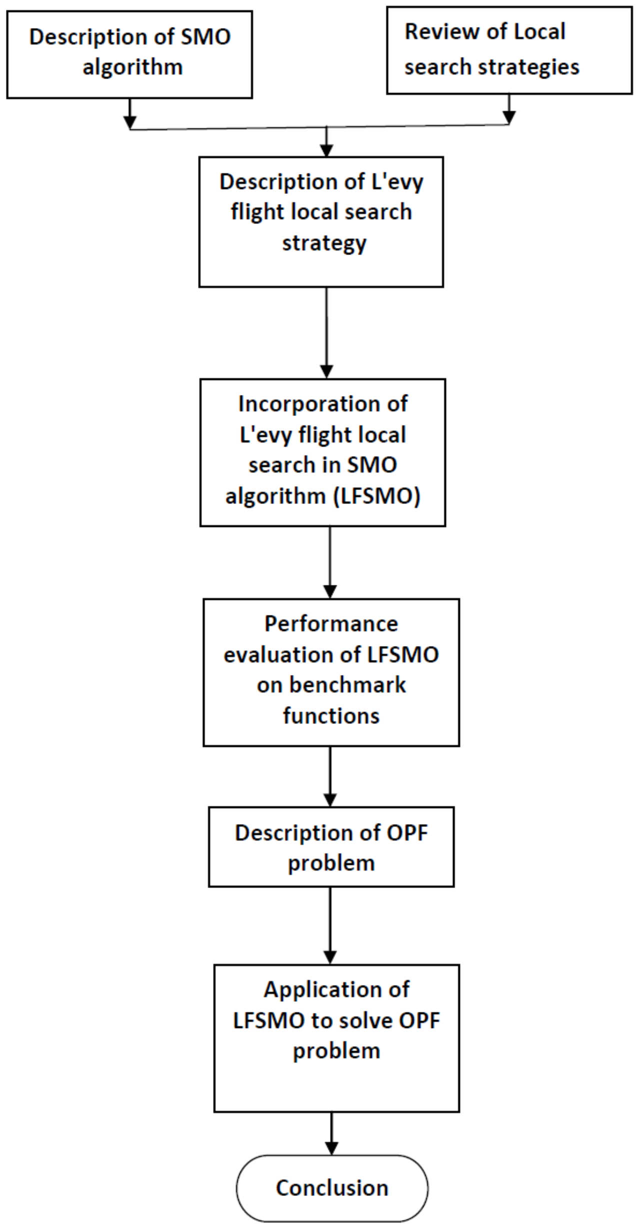


Figure 1 Flow chart of organization of paper

2.1.2 Local Leader Phase (LLP)

Here each SM updates it's current position based on gathered information from the local leader as well as local group members. The fitness value of the so obtained new position is computed. If the fitness value of the new position is superior to the old position, then the SM modifies it's position with the new one. The position update equation for i^{th} SM (which is a member of k^{th} local group) in this phase is

$$SMnew_{ij} = SM_{ij} + U(0,1) \times (LL_{kj} - SM_{ij}) + U(-1,1) \times (SM_{rj} - SM_{ij}) \tag{2}$$

where SM_{ij} is the j^{th} dimension of the i^{th} SM, LL_{kj} represents the j^{th} dimension of the k^{th} local group leader position. SM_{rj} is the j^{th} dimension of the r^{th} SM which is chosen arbitrarily within k^{th} group such that $r \neq i$. U(0, 1) is a uniformly distributed random number between 0 and 1.

2.1.3 Global Leader Phase (GLP)

In this phase, all SMs update their positions using knowledge of global leader and local group members experience. The position update equation for this phase is as follows:

$$SMnew_{ij} = SM_{ij} + U(0,1) \times (GL_j - SM_{ij}) + U(-1,1) \times (SM_{rj} - SM_{ij}) \tag{3}$$

Where GL_j is the j^{th} dimension of the global leader position and j is the randomly chosen index. The positions of SMs (SM_i) are updated based on a probability $prob_i$ which is a function of fitness. In this way, a better candidate will have more chance to make it better. The probability $prob_i$ is calculated as (H. Sharma, Bansal, & Arya, 2014):

$$prob_i = 0.9 \times \frac{fitness_i}{max\_fitness} + 0.1 \tag{4}$$

Here $fitness_i$ is the fitness value of i^{th} SM and $max\_fitness$ is the highest fitness in the group. The fitness of the newly generated SMs is calculated and compared with the old one, and the better position is adopted.

2.1.4 Global Leader Learning (GLL) Phase

In this phase, the position of the SM having best fitness in the population is selected as the updated position of the global leader using greedy selection. Further, the position of global leader is checked whether it is updating or not and if not then the global limit count is incremented by 1.

2.1.5 Local Leader Learning (LLL) Phase

In this phase, the position of the SM having best fitness in that group is selected as the updated position of the local leader using greedy selection. Next, if the modified position of the local leader is compared with the old one and if the local leader is not updated, then the local limit count is incremented by 1.

2.1.6 Local Leader Decision (LLD) Phase

If any local leader is not updated up to a preset threshold called local leader limit, then all the members of that minor group update their positions either by random initialization or by using combined information from the global leader and local leader through Equation 5.

$$SMnew_{ij} = SM_{ij} + U(0,1) \times (GL_j - SM_{ij}) + U(0,1) \times (SM_{ij} - LL_{kj}) \tag{5}$$

It is apparent from Equation 5 that the updated dimension of this SM is attracted towards global leader and repels from the local leader.

2.1.7 Global Leader Decision (GLD) Phase

In this phase, the global leader is monitored, and if it is not updated up to a preset number of iterations called global leader limit (GLL), then the global leader divides the population into minor groups. Firstly, the population is divided into two groups and then three groups and so on till the maximum number of groups (MG) are formed. After every division local leader learning process is initiated to choose the local leader in the newly formed groups. The case in which a maximum number of groups are formed and even then the position of global leader is not updated then the global leader combines all the minor groups to form a single group.

3 Recent Local Search Strategies

Researchers are constantly working in the field of memetic search approach. The exploitation of local search space is established in evolutionary computing by application of local search strategy. Some of the significant work in this area is presented in this paper. In 2008, J. Knowles et. al. (Knowles, Corne, & Deb, 2008) introduced memetic algorithm (MA) for complex optimization problems like multi-objective optimization. In 2009, F. Neri et. al. (F. Neri & Tirronen, 2009) incorporated scale factor local search to improve exploitation capability of DE. In 2009, A. Caponio et. al. (Caponio, Neri, & Tirronen, 2009) proposed super-fit memetic differential evolution (SFMDE) with two local search strategies namely, Nelder Mead algorithm and Rosenbrock algorithm. In 2009, H. Wang et. al. (Wang, Wang, & Yang, 2009) incorporated an adaptive hill climbing method as the local search technique with an evolutionary algorithm. In 2009, C. K. Goh et. al. (Goh, Ong, & Tan, 2009) presented evolutionary multi-objective optimization in uncertain environments. In 2009, K. Tang et. al. (Tang, Mei, & Yao, 2009) proposed an MA, namely memetic algorithm with extended neighborhood search (MAENS), with a novel local search operator, namely merge-split (MS). In 2009, P. P. Repoussis et. al. (Repoussis, Tarantilis, & Ioannou, 2009) presented an arc-guided evolutionary algorithm for the vehicle routing problem with time windows. In 2009, J. M. Richer et. al. (Richer, Goëffon, & Hao, 2009) proposed a memetic algorithm for phylogenetic reconstruction with maximum parsimony. In 2009, C. Gallo et. al. (Gallo, Carballido, & Ponzoni, 2009) hybridized a new approach with an evolutionary algorithm with local search for microarray biclustering. In 2009, S. S. Rao et. al. (Rao & Rao, 2009) addressed the techniques and applications of real-world case studies. In 2010, Y. S. Ong et. al. (Ong, Lim, & Chen, 2010) presented several deployments of memetic computing methodologies to solve complex real world problems. In 2010, E. Mininno et. al. (Mininno & Neri, 2010) incorporated memetic approach with DE in noisy optimization. In 2010, E. Mezura-Montes et. al. (Mezura-Montes & Velez-Koeppel, 2010)proposed elitist ABC algorithm integrated with two local search strategies, the first one is used with 30, 40, 50, 60, 70, 80, 90, 95, and 97 percentages of function evaluations have been attained while, the second is used with 45, 50, 55, 80, 82, 84, 86, 88, 90, 91, 92, 93, 94, 95, 96, 97, 98, and 99 percentages of function evaluations have been attained. In 2011, X. Chen et. al. (Chen, Ong, Lim, & Tan, 2011) proposed a comprehensive multi-facet survey of recent research in memetic computation. In 2011, F. Kang (Kang, Li, & Ma, 2011) combined Rosenbrocks rotational direction technique with ABC to present Rosenbrock's ABC (RABC). In 2012, Carlos et. al. (C. M. P. E. Neri Ferrante; Cotta, Vol. 379, 2012) presented memetic computing for studying complex structures composed of simple modules. In 2012, I. Fister et. al. (Fister Jr, Yang, Fister, & Brest, 2012) incorporated two local search heuristics with ABC namely, Nelder-Mead algorithm (NMA) and random walk with direction exploitation (RWDE). Further, I. Fister et. al. proposed memetic firefly algorithm for combinatorial optimization (Fister Jr et al., 2012). Later in 2012, C. Cotta et. al. (Cotta & Neri, 2012) combined various operators to inculcate them with the stochastic adaptive rule as specified for balancing the exploration and exploitation. A brief review of nature inspired algorithms was presented by I. Fister et. al. in 2013 (Fister Jr, Yang, Fister, Brest, & Fister, 2013). In same year, H. Sharma et. al. (H. Sharma, Bansal, & Arya, 2013; H. Sharma, Jadon, Bansal, & Arya, 2013) introduced ABC algorithm with opposition based l'evy flight local search (LFLS) strategy and LFLS with DE algorithm. In 2015, H. Sharma et. al. (H. Sharma et al., 2015) integrated LFLS strategy with ABC. Recently in 2016 Singh et. al. (Singh & Dhillon, 2016) presented multiobjective thermal power dispatch using opposition-based greedy heuristic search. In the same year Lotfipour et. al. (Lotfipour & Afrakhte, 2016) propound a discrete Teaching-Learning-Based Optimization algorithm to solve distribution system reconfiguration in presence of distributed generation. Further, In 2016, A. Sharma et. al. (A. Sharma, Sharma, Bhargava, & Sharma, 2016) incorporated power law local search strategy with SMO algorithm for solving real-world lower order system modeling problem. In same year A. Yadav et. al. proposed a new searching ability testing technique in harmony search algorithm (HSA) (Yadav, Yadav, & Kim, 2016). Further, A. Yadav et. al. hybridized PSO with gravitational search algorithm (GSA) to propose particle swarm optimizer for global optimization (Yadav, Deep, Kim, & Nagar, 2016). Recently A. Sharma et. al. proposed a Limaçon inspired local search strategy and incorporated it into SMO algorithm to solve capacitor placement problem (A. Sharma, Sharma, Bhargava, Sharma, & Bansal, 2016).

From memetic search study, it is clear that every search region may be divided into two kinds of search spaces, local search space, and global search space to find out local optimum and global optimum respectively. The optimum equilibrium between exploration and exploitation in any algorithm is always maintained, and it is suggested to inculcate a local search methodology in the primary population-based algorithm for exploiting the search space. An identified region of a given search space can be exploited by the local search algorithms; therefore, the local search algorithms are applied to the global search algorithms to improve the exploitation capability of the global search algorithm. Here the main algorithm explores while the local search exploits the search space. The main role of local search algorithms in evolutionary computing is to refine an identified search area to establish exploitation of the search space (C. M. P. E. Neri Ferrante; Cotta, Vol. 379, 2012).

4 Lévy Flight inspired Local Search Strategy

In this paper, a lévy flight local search (LFLS) strategy developed by Sharma et. al. (H. Sharma et al., 2015), is applied with basic SMO to improve its exploitation capability. The LFLS strategy is inspired by lévy flight random walk in which the step size is defined regarding the step-length with a particular probability distribution. The Equation 6 is used to draw the lévy random step lengths distribution.

$$L(s) \sim |s|^{-1-\beta}, \text{ where } \beta \ (0 < \beta \leq 2) \text{ is an index and } s \text{ is the step length} \tag{6}$$

In this work a lévy distribution in symmetric form i.e. either positive or negative step size is used, known as Mantegna algorithm (Yang, 2011), in which step length s is calculated as follows;

$$s = \frac{u}{|v|^{1/\beta}} \tag{7}$$

where, u and v are drawn from normal distributions. That is

$$u \sim N(0, {\sigma_u}^2), \ v \sim N(0, {\sigma_v}^2) \tag{8}$$

where,

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\beta\Gamma[(1+\beta)/2]2^{(\beta-1)/2}} \right\}^{1/\beta}, \ \sigma_v = 1 \tag{9}$$

This distribution (for s) obeys the expected lévy distribution for $|s| \geq |s_0|$, where s_0 is the smallest step length (Yang, 2011). Here $\Gamma(.)$ is the Gamma function and calculated as follows:

$$\Gamma(1+\beta) = \int_0^\infty t^\beta e^{-t} dt \tag{10}$$

If β is an integer, then $\Gamma(1+\beta) = \beta!$.

In the proposed strategy, the step sizes are generated using lévy distribution to exploit the search area and calculated as follows:

$$step\_size(t) = 0.002 \times s(t) \times SLC \tag{11}$$

here t is the iteration counter for local search strategy, $s(t)$ is calculated using lévy distribution as shown in Equation 7 and the social learning component (SLC) is used from the global search algorithm.

In lévy flights, the step sizes are too aggressive, that is, they may generate new solutions often outside the domain or on a boundary. Since, the local search algorithms can be seen as a population based stochastic algorithms, where the main task is to exploit the available knowledge about a problem and steps sizes play a major role in exploiting the identified region. Therefore, 0.002 multiplier is used in Equation 11 to reduce the step size. The multiplier is selected through empirical experiments. The solution update equation of an i^{th} individual, based on the proposed local search strategy is given in Equation 12:

$$x'_{ij}(t+1) = x_{ij}(t) + step\_size(t) \times U(0,1) \tag{12}$$

here x_{ij} is the individual which is going to modify its position, $U(0,1)$ is a uniformly distributed random number between 0 and 1 and $step\_size(t) \times U(0,1)$ is the actual random walks or flights drawn from lévy distribution.

The pseudo-code of the proposed $LFLS$ is shown in Algorithm 1 (H. Sharma et al., 2015). In

Algorithm 1 Lévy Flight Local Search Strategy:

```
Input optimization function Minf(x) and β;
Select an individual x_i in the swarm which is going to modify its position;
Initialize t = 1 and σ_v = 1;
Compute σ_u using Equation 9;
while (t < ϵ) do
  Compute step_size using Equation 11;
  Generate a new solution x'_i using Equation 12;
  Calculate f(x'_i);
  if f(x'_i) < f(x_i) then
    x_i = x'_i;
  end if
  t = t + 1;
end while
```

Algorithm 1, ϵ determines the termination of local search.

5 Lévy Flight Spider Monkey Optimization

Bansal et. al. (Bansal, Sharma, et al., 2014) show that exploration and exploitation cannot be balanced due to the presence of random component and there is always a high chance to skip the true solution. Therefore, to expedite the exploitation capability of SMO, a lévy flight local search (LFLS) strategy is incorporated with it. The LFLS, in the case of large step sizes, can search within the area which otherwise was not exploited by the basic SMO. In LFLS, the step size is calculated using Equation 13:

$$step\_size = 0.002 \times s(t) \times (x_{bestj} - x_{kj}) \times U(0,1) \tag{13}$$

here, t is the iteration counter for local search strategy. Social learning component (SLC) is $(x_{bestj} - x_{kj})$ in which x_{best} is the best solution in the current swarm and x_k is the randomly selected solution within population and $x_k \neq x_{best}$. The position update equation of the best individual within the current population is given by Equation 14:

$$x'_{bestj}(t+1) = x_{bestj}(t) + step\_size(t) \tag{14}$$

In LFLS, only the best particle of the current swarm updates itself in its neighborhood. The pseudo-code of the proposed LFLS with SMO is shown in Algorithm 2.

Algorithm 2 Lévy Flight Local Search Strategy with SMO:

Input optimization function $Minf(x)$ and β;
Select the best solution x_{best} in the swarm;
Initialize $t = 1$ and $\sigma_v = 1$;
Compute σ_u using Equation 9;
while $(t < \epsilon)$ **do**
  Compute $step\_size$ using Equation 13;
  Generate a new solution x'_{best} using Algorithm 3;
  Calculate $f(x'_{best})$;
  if $f(x'_{best}) < f(x_{best})$ **then**
    $x_{best} = x'_{best}$;
  end if
  $t = t + 1$;
end while

Algorithm 3 New solution generation:

Input the best solution x_{best} and s;
for $j = 1$ to D **do**
  if $U(0, 1) > p_r$ **then**
    $x'_{bestj} = x_{bestj} + step\_size$;
  else
    $x'_{bestj} = x_{bestj}$;
  end if
end for
Return x'_{best}

In Algorithm 2 and 3, ϵ is the termination criteria of the proposed local search, p_r is a perturbation rate (a number between 0 and 1) which controls the amount of perturbation in the best solution, $U(0, 1)$ is a uniform distributed random number between 0 and 1, D is the dimension of the problem and x_k is a randomly selected solution within swarm. See section 6.1 for details of these parameter settings.

The proposed lévy flight SMO (LFSMO) consists of seven phases: local leader phase, global leader phase, local leader learning phase, global leader learning phase, local leader decision phase and global leader decision phase and lévy flight local search phase (LFLS). The pseudo-code of the LFSMO algorithm is shown in Algorithm 4.

Algorithm 4 Lévy Flight SMO:

Initialize the parameters;
while Termination criteria **do**
  Step 1: Local Leader phase.
  Step 2: Global Leader phase.
  Step 3: Local Leader Learning phase.
  Step 4: Global Leader Learning phase.
  Step 5: Local Leader Decision phase.
  Step 6: Global Leader Decision phase.
  Step 7: Apply Lévy Flight Local Search Strategy (LFLS) phase using Algorithm 2.
end while
Print best solution.

6 Experimental Results and Discussion

To analyze the performance of $LFSMO$, 25 different global optimization problems (f_1 to f_{25}) (listed in Table 1) are selected. These are continuous optimization problems and have different degrees of complexity and multimodality. Test problems f_1 to f_{25} are taken from (Ali, Khompatraporn, & Zabinsky, 2005; Suganthan et al., 2005) with the associated offset values. All the experiments are carried out by the algorithms implemented in MATLAB 7.8 on a personal computer with the Intel Core i7 processor, 4GB RAM, and Windows 7 operating system.

Table 1: Test problems. D: Dimensions, C: Characteristic, U: Unimodal, M: Multimodal, S: Separable, N: Non-Separable, AE: Acceptable Error

Test Problem	Objective function	search Range	Optimum Value	D	AE	C
De Jong f4	$f_1(x) = \sum_{i=1}^{D} i \times (x_i)^4$	[-5.12, 5.12]	$f(\vec{0}) = 0$	30	$1.0E-05$	M, S
Rastrigin	$f_2(x) = 10D + \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i)]$	[-5.12 5.12]	$f(\vec{0}) = 0$	30	$1.0E-05$	M, N
Ackley	$f_3(x) = -20 + e + exp(-\frac{0.2}{D}\sqrt{\sum_{i=1}^{D} {x_i}^3})$ $-exp(\frac{1}{D}\sum_{i=1}^{D} \cos\ (2\pi x_i) x_i)$	[-1 1]	$f(\vec{0}) = 0$	30	$1.0E-05$	M, N
Exponential	$f_4(x) = -(exp(-0.5\sum_{i=1}^{D} {x_i}^2)) + 1$	[-1 1]	$f(\vec{0}) = -1$	30	$1.0E-05$	M, N
Zakharov	$f_5(x) = \sum_{i=1}^{D} {x_i}^2 + (\sum_{i=1}^{D} \frac{ix_i}{2})^2 + (\sum_{i=1}^{D} \frac{ix_1}{2})^4$	[-5.12 5.12]	$f(\vec{0}) = 0$	30	$1.0E-02$	M, N
Cigar	$f_6(x) = {x_0}^2 + 100000\sum_{i=1}^{D} {x_i}^2$	[-10 10]	$f(\vec{0}) = 4$	30	$1.0E-05$	U, S
brown3	$f_7(x) = \sum_{i=1}^{D-1} ({x_i}^{2^{(x_{i+1})^2+1}} + {x_{i+1}}^{2^{{x_i}^2+1}})$	[-1 4]	$f(\vec{0}) = 0$	30	$1.0E-05$	U, N
Axis parallel hyper-ellipsoid	$f_8(x) = \sum_{i=1}^{D} i \times x_i^2$	[-5.12, 5.12]	$f(\vec{0}) = 0$	30	$1.0E-05$	U, S
Sum of different powers	$f_9(x) = \sum_{i=1}^{D} \lvert x_i \rvert^{i+1}$	[-1, 1]	$f(\vec{0}) = 0$	30	$1.0E-05$	M, S
Rotated hyper-ellipsoid	$f_{10}(x) = \sum_{i=1}^{D} \sum_{j=1}^{i} x_j^2$	[-65.536, 65.536]	$f(\vec{0}) = 0$	30	$1.0E-05$	M, S
Ellipsoidal	$f_{11}(x) = \sum_{i=1}^{D} (x_i - i)^2$	[-D, D]	$f(1, 2, 3, \ldots, D) = 0$	30	$1.0E-05$	U, S
Beale function	$f_{12}(x) = [1.5 - x_1(1 - x_2)]^2 + [2.25 - x_1(1 - x_2^2)]^2 + [2.625 - x_1(1 - x_2^3)]^2$	[-4.5, 4.5]	$f(3, 0.5) = 0$	2	$1.0E-05$	M, N
Colville function	$f_{13}(x) = 100[x_2 - x_1^2]^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19.8(x_2 - 1)(x_4 - 1)$	[-10, 10]	$f(\vec{1}) = 0$	4	$1.0E-05$	M, N
Kowalik	$f_{14}(x) = \sum_{i=1}^{11} [a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	[-5, 5]	$f(0.192833,$ $0.190836, 0.123117,$ $0.135766) = 0.000307486$	4	$1.0E-05$	M, N
2D Tripod function	$f_{15}(x) = p(x_2)(1 + p(x_1)) + \lvert(x_1 + 50p(x_2)(1 - 2p(x_1)))\rvert + \lvert(x_2 + 50(1 - 2p(x_2)))\rvert$	[-100, 100]	$f(0, -50) = 0$	2	$1.0E-04$	M, N
Shifted Rosenbrock	$f_{16}(x) = \sum_{i=1}^{D-1} (100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2) + f_{bias}, \quad z = x - o + 1,\ x = [x_1, x_2, \ldots. x_D],\ o = [o_1, o_2, \ldots o_D]$	[-100, 100]	$f(o) = f_{bias} = 390$	10	$1.0E-01$	M, N
Shifted Sphere	$f_{17}(x) = \sum_{i=1}^{D} z_i^2 + f_{bias}, \quad z = x - o \quad, x = [x_1, x_2, \ldots. x_D], \quad o = [o_1, o_2, \ldots o_D]$	[-100, 100]	$f(o) = f_{bias} = -450$	10	$1.0E-05$	M, S
Shifted Ackley	$f_{18}(x) = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi z_i)) + 20 + e + f_{bias}, \quad z = (x - o), \quad x = (x_1, x_2, \ldots\ldots. x_D), \quad o = (o_1, o_2, \ldots\ldots. o_D)$	[-32, 32]	$f(o) = f_{bias} = -140$	10	$1.0E-05$	M, S
Six-hump camel back	$f_{19}(x) = (4 - 2.1x_1^2 + x_1^4/3)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$	[-5, 5]	-1.0316	2	$1.0E-03$	M, N
Easom's function	$f_{20}(x) = -\cos x_1 \cos x_2 e^{((-(x_1-\pi)^2 - (x_2-\pi)^2))}$	[-10, 10]	$f(\pi, \pi) = -1$	2	$1.0E-13$	M, S
Dekkers and Aarts	$f_{21}(x) = 10^5 x_1^2 + x_2^2 - (x_1^2 + x_2^2)^2 + 10^{-5}(x_1^2 + x_2^2)^4$	[-20, 20]	$f(0, 15) = f(0, -15) = -24777$	2	$5.0E-01$	M, N
McCormick	$f_{22}(x) = \sin(x_1 + x_2) + (x_1 - x_2)^2 - \frac{3}{2}x_1 + \frac{5}{2}x_2 + 1$	$-1.5 \leq x_1 \leq 4, -3 \leq x_2 \leq 3$	$f(-0.547, -1.547) = -1.9133$	30	$1.0E-04$	M, N

to be cont'd on next page

Table 1: Test problems. D: Dimensions, C: Characteristic, U: Unimodal, M: Multimodal, S: Separable, N: Non-Separable, AE: Acceptable Error

Test Problem	**Objective function**	**search Range**	**Optimum Value**	**D**	**AE**	**C**
Meyer and Roth Problem	$f_{23}(x) = \sum_{i=1}^{5}(\frac{x_1 x_3 t_i}{1+x_1 t_i + x_2 v_i} - y_i)^2$	[-10, 10]	$f(3.13, 15.16, 0.78) = 0.4E-04$	3	$1.0E-03$	U, N
Shubert	$f_{24}(x) = -\sum_{i=1}^{5} i\cos((i+1)x_1+1)\sum_{i=1}^{5} i\cos((i+1)x_2+1)$	[-10, 10]	$f(7.0835, 4.8580) = -186.7309$	2	$1.0E-05$	M, S
Pressure Vessel	f_{25}=$0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$	x_1=[1.1, 12.5] x_2=[0.6, 12.5] x_3=[0, 240] x_4=[0, 240]	$f(\vec{0}) = 0$	4	$1.0E-05$	M, S

6.1 Experimental Setting

To analyze the performance of $LFSMO$, it is compared with basic SMO, $PSO-2011$, ABC, DE, $CMA-ES$, and $PLSMO$ with following experimental settings (Bansal, Sharma, et al., 2014):

- Population Size N=50,
- $MG = N/10$,
- $GlobalLeaderLimit = 50$,
- $LocalLeaderLimit$ =1500,
- $pr \in [0.1, 0.4]$, linearly increasing over iterations,

$$pr_{G+1} = pr_G + (0.4 - 0.1)/MIR \tag{15}$$

  where, G is the iteration counter, MIR is the maximum number of iterations.,
- The stopping criteria is either maximum number of function evaluations (which is set to be 200000) is reached or the acceptable error (mentioned in Table 1) has been achieved,
- The number of simulations/run =100,
- To set termination criteria of $LFLS$, the performance of $LFSMO$ is measured for considered test problems on different values of ϵ and results in terms of success are analyzed in Figure 2(a). It is clear from Figure 2(a) that $\epsilon = 10$ gives better results (highest value of sum of success). Therefore, termination criteria is set to be $\epsilon = 10$,
- The value of $\beta = 0.002$ is to be set based on the empirical experiments as shown in Figure 2 (b),
- Parameter settings for the algorithm SMO, ABC, $PSO-2011$, DE, $PLSMO$, $CMA-ES$, are similar to their legitimate research papers respectively.

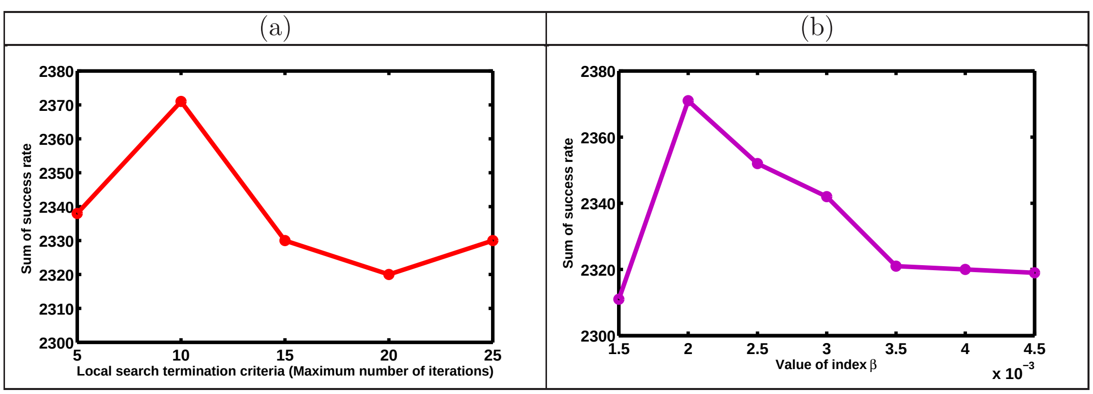


Figure 2 Variation in sum of success (a) $LFSS$ termination criteria ϵ (b) with parameter β.

6.2 Results Comparison

The numerical outcomes acquired are presented in Table 2 for success rate (SR), average number of function evaluations (AFE), mean error (ME), and standard deviation (SD).

The $LFSMO$, SMO, $PSO-2011$, ABC, DE, $CMA-ES$ and one recent significant variant of SMO namely, $PLSMO$ are compared in terms of SR, AFE, ME, and SD as presented in Table 2. The outcomes present that $LFSMO$ is competitive than SMO and other considered algorithms for most of the benchmark test problems. The considered algorithms are also equated through Mann-Whitney U rank sum test (H. Sharma et al., 2015), acceleration rate (AR) and boxplot analysis. Mann-Whitney U rank sum test is applied on average number of function evaluations. For all considered algorithms the test is performed at 5% significance level ($\alpha = 0.05$) and the output results for 100 simulations are presented in Table 3. In this table '+' sign indicates that $LFSMO$ is better while '-' sign shows that the other considered algorithm is better. The '=' sign represents that both the compared algorithms perform almost equally. The $LFSMO$ outperforms as compared to all other considered algorithms for 7 test problems including f_2-f_4, f_8, f_9, f_{11}, and f_{24}. $LFSMO$ performs better than basic SMO for 16 test problems, f_2-f_4, $f_8, f_9, f_{11}-f_{16}, f_{19}$, and f_{21}, f_{22}, f_{24}, f_{25}. The $LFSMO$ shows better results for 24 test problems when compared with $PSO-2011$ algorithm, f_1-f_{19} and $f_{21}-f_{25}$. The $LFSMO$ performs better for 22 test problems, f_2-f_{14}, $f_{16}-f_{18}$, $f_{20}-f_{25}$ in comparison with ABC. The $LFSMO$ performs better for 23 test problems in comparison with DE, f_1-f_4 and f_6-f_{19}, and $f_{21}-f_{25}$. In comparison with $CMA-ES$, $LFSMO$ performs better on 16 functions, f_1-f_{11}, f_{17}, f_{19}, $f_{23}-f_{25}$. The $LFSMO$ shows better results for 12 test problems, f_2-f_4, f_8, f_9, f_{11}, f_{12}, f_{14}, f_{15}, f_{19}, f_{21}, and f_{24} when compared with a recent significant variant of SMO namely, $PLSMO$ algorithm. On functions f_{22}, f_{25}, $LFSMO$ and $PLSMO$ show almost equal performance. The above discussion represents that $LFSMO$ may be a competitive candidate in the field of swarm intelligence.

Further, the convergence speed of considered algorithms are compared by analysis of AFEs. There is an inverse relation between AFEs and convergence speed, for smaller AFEs the convergence speed will be higher and vice-versa. For minimizing the effects of stochastic nature of algorithm, the reported AFEs are averaged for 100 runs for each considered test problem. The convergence speed is compared using acceleration rate (AR) for the considered algorithms which is calculated as follows:

$$AR = \frac{AFE_{ALGO}}{AFE_{LFSMO}}, \tag{16}$$

Here, $ALGO \in \{SMO, ABC, PSO-2011, DE, CMA-ES, PLSMO\}$ and $AR > 1$ represents that $LFSMO$ is faster than the compared algorithm. The AR results are shown in Table 4. The results in Table 4 show that $LFSMO$ converge faster than the considered algorithms for most of the considered benchmark test functions.

The graphical distribution of empirical data of the considered algorithms is efficiently represented by the boxplot analysis (H. Sharma et al., 2015). The boxplots for $LFSMO$ and other considered algorithms are represented in Figure 3. It is clear from this figure that $LFSMO$ performs better than the considered algorithms as interquartile range, and the median is quite low.

In Table 1, the characteristics of test problems in terms of separability, non-separability, unimodularity, and multi-modularity are shown. The above analysis in reference of Table 1 proves that the proposed LFSMO performs better for nonseparable and multimodal functions. Therefore, the proposed variant is best-suited for the multimodal, non-separable problems.

Table 2: Comparison of the results of test problems

Test Problem	Measure	LFSMO	SMO	PSO-2011	ABC	DE	PLSMO	CMA-ES
f_1	SD	1.26E-06	1.20E-06	8.62E-07	3.11E-06	8.51E-07	9.49E-07	1.96E-06
	ME	8.54E-06	8.49E-06	9.03E-06	4.90E-06	9.01E-06	8.68E-06	7.37E-06
	AFE	13325.96	10725.66	32596.5	9578.5	20859.5	11198.07	20907.6
	SR	100	100	100	100	100	100	100
f_2	SD	1.61E-06	1.56E-06	1.40E+01	3.30E-06	4.63E+00	1.23E-06	1.67E-03
	ME	8.58E-06	8.24E-06	3.87E+01	5.66E-06	1.49E+01	8.81E-06	4.73E-06
	AFE	25378.14	96073.45	200050	49984	200000	104046.74	200128
	SR	100	100	0	100	0	100	0
f_3	SD	1.30E-01	9.32E-07	3.66E-07	1.54E-06	4.42E-07	5.44E-07	3.49E-07
	ME	1.86E-02	9.26E-06	9.69E-06	8.63E-06	9.46E-06	9.44E-06	9.46E-06
	AFE	25000.8	32438.7	77352	48726.5	43100.5	36013.61	64509.5
	SR	98	100	100	100	100	100	100
f_4	SD	6.94E-07	7.46E-07	6.15E-07	2.34E-06	9.15E-07	8.37E-07	1.19E-06
	ME	9.06E-06	8.96E-06	9.33E-06	7.12E-06	8.98E-06	8.96E-06	8.61E-06
	AFE	9003.88	9794.07	28227.5	16974	17269	10149.51	22933.3
	SR	100	100	100	100	100	100	100
f_5	SD	1.11E-03	9.68E-04	1.80E-02	1.44E+01	5.10E-04	5.99E-04	4.5E-02
	ME	9.44E-03	9.15E-03	2.20E-02	9.75E+01	9.43E-03	9.56E-03	9.73E+01
	AFE	152136.9	142800.47	196434	200000	69828.5	120875.68	200128
	SR	99	100	31	0	100	100	0
f_6	SD	7.32E-07	9.57E-07	6.96E-07	2.20E-06	8.75E-07	9.05E-07	9.97E-07
	ME	9.01E-06	8.93E-06	9.29E-06	7.77E-06	9.02E-06	8.78E-06	8.77E-06
	AFE	28300.25	22572	69125.5	34887	39861.5	23403.64	70154.2
	SR	100	100	100	100	100	100	100
f_7	SD	8.21E-07	6.39E-07	6.26E-07	2.13E-06	8.21E-07	9.09E-07	1.25E-06
	ME	8.97E-06	9.10E-06	9.24E-06	7.84E-06	9.01E-06	8.80E-06	8.39E-06
	AFE	15855.55	12635.37	35048.5	20917	22253.5	13133.47	32220.5
	SR	100	100	100	100	100	100	100
f_8	SD	9.92E-07	8.60E-07	6.37E-07	2.04E-06	1.01E-06	6.80E-07	8.90E-07
	ME	8.97E-06	9.00E-06	9.33E-06	7.96E-06	8.92E-06	9.08E-06	8.88E-06
	AFE	9451	14811.39	44374.5	22672	25899	15385.16	38151.9
	SR	100	100	100	100	100	0	100
f_9	SD	1.83E-06	1.81E-06	1.38E-06	2.65E-06	2.02E-06	1.59E-06	3.27E-06
	ME	7.89E-06	7.43E-06	8.48E-06	5.16E-06	7.23E-06	7.95E-06	7.05E-06
	AFE	4480.46	5229.18	9897	16229	7950	5347.58	51777.2
	SR	100	100	100	100	100	100	84
f_{10}	SD	9.81E-07	7.77E-07	8.13E-07	1.94E-06	9.82E-07	8.37E-07	9.51E-07
	ME	8.80E-06	9.05E-06	9.20E-06	8.11E-06	8.86E-06	8.94E-06	8.56E-06
	AFE	23294.44	18535.77	56547	28270.5	32747.5	19424.76	42997.6
	SR	100	100	100	100	100	100	100
f_{11}	SD	8.48E-07	8.45E-07	5.56E-07	2.35E-06	8.74E-07	7.94E-07	9.88E-07
	ME	8.89E-06	8.78E-06	9.33E-06	7.52E-06	8.93E-06	8.97E-06	8.60E-06
	AFE	12287.89	15453.9	44306	24219.5	27365.5	15960.9	40125.9
	SR	100	100	100	100	100	100	100
f_{12}	SD	2.51E-06	2.74E-06	2.81E-06	1.83E-06	2.81E-06	3.11E-06	2.44E-06
	ME	7.38E-06	4.33E-06	4.96E-06	8.29E-06	4.74E-06	5.27E-06	3.75E-06
	AFE	1282.57	1593.9	2753.5	16954.05	1415.5	1288.84	787.1
	SR	100	100	100	100	100	100	100
f_{13}	SD	5.87E-05	2.42E-04	2.24E-04	1.08E-01	1.66E-01	1.76E-03	2.07E-04
	ME	9.74E-04	7.54E-04	8.13E-04	1.58E-01	5.34E-02	8.77E-03	6.48E-0
	AFE	16756.98	51610.79	48776.5	200025.94	36451.5	12409.47	8173.3
	SR	100	100	100	0	84	100	100

to be cont'd on next page

Table 2: Comparison of the results of test problems (Cont.)

Test Problem	Measure	LFSMO	SMO	PSO-2011	ABC	DE	PLSMO	CMA-ES
f_{14}	SD	1.15E-04	1.37E-05	1.18E-05	6.94E-05	3.32E-04	1.16E-04	7.19E-05
	ME	1.15E-04	8.94E-05	8.97E-05	1.75E-04	2.39E-04	1.10E-04	2.5E-04
	AFE	22406.22	40395.78	35865	183176.85	46592.5	36998.05	151.6
	SR	98	100	100	17	79	0	100
f_{15}	SD	2.71E-05	2.38E-05	2.71E-01	2.39E-05	2.71E-01	2.42E-05	2.75E-07
	ME	7.04E-05	6.62E-05	8.01E-02	6.43E-05	8.01E-02	6.89E-05	4.87E-07
	AFE	11108.01	16563.09	29745.5	7927.03	19150.5	17142.34	1574
	SR	100	100	92	100	92	20	100
f_{16}	SD	1.57E+00	4.54E+00	1.08E+01	1.24E+00	3.10E+00	7.77E+00	6.87E-05
	ME	9.84E-01	1.90E+00	2.92E+00	7.92E-01	2.62E+00	1.33E+00	1.07E-03
	AFE	151526.01	167177.49	187162.5	174330.22	193120.5	147412.82	160
	SR	53	42	50	23	4	100	100
f_{17}	SD	1.76E-06	1.82E-06	1.50E-06	2.61E-06	1.71E-06	1.72E-06	1.90E-06
	ME	7.61E-06	7.38E-06	8.29E-06	6.97E-06	7.95E-06	7.55E-06	7.26E-06
	AFE	7447.95	5953.86	15785.5	9042.5	10353.5	6174.79	9665.3
	SR	100	100	100	100	100	100	100
f_{18}	SD	1.03E-06	9.65E-07	1.05E-06	1.98E-06	1.17E-06	1.11E-06	1.07E-06
	ME	8.65E-06	8.76E-06	8.93E-06	7.76E-06	8.81E-06	8.63E-06	8.64E-06
	AFE	10403.87	9122.85	24630	16704.5	15564.5	9458.46	3521.933333
	SR	100	100	100	100	100	100	100
f_{19}	SD	1.58E-05	1.41E-05	1.18E-05	1.10E-05	1.49E-05	1.52E-05	2.35E-06
	ME	1.59E-05	1.90E-05	1.75E-05	1.20E-05	1.67E-05	1.73E-05	5.37E-06
	AFE	90452.22	125358.44	105570.5	1017	100761	104425.84	17365
	SR	55	40	48	100	50	17	100
f_{20}	SD	3.25E-14	2.94E-14	2.92E-14	8.37E-05	2.80E-14	2.75E-14	1.40E07
	ME	5.52E-14	4.40E-14	4.82E-14	3.09E-05	4.17E-14	4.81E-14	3.98E04
	AFE	12672.21	11916.63	9796.5	186124.2	4798.5	12295.99	594
	SR	100	100	100	16	100	100	100
f_{21}	SD	5.26E-03	5.38E-03	5.55E-03	5.76E-03	4.80E-03	5.32E-03	8.17E-14
	ME	4.92E-01	4.90E-01	4.92E-01	4.91E-01	4.89E-01	4.90E-01	7.83E-14
	AFE	901.63	1249.38	5050	1407.52	2123	1185.32	9612
	SR	100	100	100	100	100	100	100
f_{22}	SD	7.19E-06	6.15E-06	6.86E-06	6.36E-06	6.54E-06	7.12E-06	1.76E+00
	ME	8.84E-05	8.65E-05	8.84E-05	8.80E-05	8.78E-05	8.82E-05	2.05E01
	AFE	730.37	738.54	1445	1176.04	971.5	728.94	258.433
	SR	100	100	100	100	100	100	100
f_{23}	SD	3.07E-06	3.02E-06	2.93E-06	2.61E-06	1.30E-05	3.16E-06	8.99E04
	ME	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.25E02
	AFE	1907.99	1861.2	3092	28795.15	3667.5	1845.37	142128
	SR	100	100	100	100	99	100	87
f_{24}	SD	5.65E-06	5.61E-06	1.37E-03	5.92E-06	5.16E-06	5.56E-06	6.27E-06
	ME	5.31E-06	4.94E-06	3.12E-04	5.36E-06	4.48E-06	4.95E-06	7.32E-06
	AFE	2238.63	4551.03	90199	4802.25	8287	3625.3	14262
	SR	100	100	71	100	100	100	100
f_{25}	SD	3.68E-05	6.42E-04	3.26E-05	9.88E+00	3.51E-05	2.43E-04	3.78E-05
	ME	3.33E-05	1.21E-04	3.24E-05	1.64E+01	2.83E-05	7.01E-05	3.54E-05
	AFE	77220.5	119586.19	98909	200024.59	77220.5	111623.22	87220.5
	SR	68	53	60	0	65	0	62

Table 3 Comparison based on Mann-Whitney U rank sum test at significant level $\alpha = 0.05$ and average number of function evolutions, TP: test problem.

TP	LFSMO Vs SMO	LFSMO Vs PSO-2011	LFSMO Vs ABC	LFSMO Vs DE	LFSMO Vs PLSMO	LFSMO Vs CMA-ES
f_1	-	+	-	+	-	+
f_2	+	+	+	+	+	+
f_3	+	+	+	+	+	+
f_4	+	+	+	+	+	+
f_5	-	+	+	-	-	+
f_6	-	+	+	+	-	+
f_7	-	+	+	+	-	+
f_8	+	+	+	+	+	+
f_9	+	+	+	+	+	+
f_{10}	-	+	+	+	-	+
f_{11}	+	+	+	+	+	+
f_{12}	+	+	+	+	+	-
f_{13}	+	+	+	+	-	-
f_{14}	+	+	+	+	+	-
f_{15}	+	+	-	+	+	-
f_{16}	+	+	+	+	-	-
f_{17}	-	+	+	+	-	+
f_{18}	-	+	+	+	-	-
f_{19}	+	+	-	+	+	-
f_{20}	-	-	+	-	-	+
f_{21}	+	+	+	+	+	-
f_{22}	+	+	+	+	=	-
f_{23}	-	+	+	+	-	+
f_{24}	+	+	+	+	+	+
f_{25}	+	+	+	+	=	+
Total No. of + sign	16	24	22	23	14	16

7 Optimal Power Flow Problem

The aim of solving OPF problem is to provide optimal settings of power system control variables to ensure operational and physical constraints in terms of equality and inequality. Mathematical formulation of a general OPF problem can be seen as:

$$MinF(u, x) \tag{17}$$

subject to:

$$G(u, x) = 0 \tag{18}$$

$$H^{min} \leq H(u, x) \leq H^{max} \tag{19}$$

The objective function which is to be optimized is F, u the vector for control variables (independent variables) which includes generator bus voltages (V_g), generator real powers (P_g) except at slack bus, transformer tap settings (T) and shunt VAR compensation (Q_c) and x is the vector of state variables (dependent variables) which includes generator active power at slack bus (P_{g1}), load bus voltages (V_l), generator reactive powers (Q_g) and transmission line loading (line flow) (S_l). Hence, u and x can be expressed as:

$$u = [P_{g2}....P_{gng}, V_{g1}....V_{gng}, Q_{c1}....Q_{cnc}, T_1....T_{nt}] \tag{20}$$

Table 4 Comparison based on Acceleration Rate (AR)

TP:Test Problem

TP	LFSMO Vs SMO	LFSMO Vs PSO-2011	LFSMO Vs ABC	LFSMO Vs DE	LFSMO Vs PLSMO	LFSMO Vs CMA-ES
f_1	0.804869593	2.446090188	0.718784988	1.565328126	0.840319947	1.568937623
f_2	3.785677359	7.882768398	1.969569086	7.880798199	4.099856806	7.88584191
f_3	1.29750648	3.093980993	1.948997632	1.723964833	1.440498304	2.58029743
f_4	1.087761054	3.135037339	1.885187275	1.917950928	1.127237369	2.547046384
f_5	0.938631391	1.291166048	1.314605464	0.458984638	0.794519147	1.315446811
f_6	0.797590127	2.442575596	1.232745294	1.408521126	0.82697644	2.478925098
f_7	0.796905185	2.21048781	1.319222607	1.403514858	0.828320052	2.032127552
f_8	1.567177018	4.695217437	2.398899587	2.740344937	1.627886996	4.036810919
f_9	1.167107842	2.208924976	3.622172723	1.774371382	1.1935337	11.55622414
f_{10}	0.795716489	2.427489135	1.213615781	1.405807566	0.833879673	1.845831022
f_{11}	1.25765286	3.605663788	1.9710056	2.227030027	1.298912995	3.265483334
f_{12}	1.242739188	2.14686138	13.21881067	1.103643466	1.004888622	0.613689701
f_{13}	3.079957725	2.910816866	11.93687287	2.175302471	0.740555279	0.487754953
f_{14}	1.802882414	1.600671599	8.17526785	2.079444904	1.651240147	0.006765978
f_{15}	1.491094264	2.677842386	0.713631875	1.724026176	1.543241319	0.141699548
f_{16}	1.103292365	1.235183979	1.150497	1.274503961	0.972854891	0.001055924
f_{17}	0.799395807	2.119442263	1.214092468	1.390114058	0.829059003	1.297712793
f_{18}	0.876870818	2.367388289	1.605604453	1.496029843	0.909129007	0.338521467
f_{19}	1.385907831	1.167141061	0.011243505	1.113969342	1.154486203	0.19197981
f_{20}	0.940375041	0.773069575	14.68758804	0.378663232	0.970311414	0.046874223
f_{21}	1.385690361	5.600967137	1.561083815	2.354624402	1.314641261	10.6606923
f_{22}	1.011186111	1.978449279	1.610197571	1.330147733	0.998042088	0.353838465
f_{23}	0.975476811	1.620553567	15.09187679	1.922179886	0.967180121	74.49095645
f_{24}	2.03295319	40.29205362	2.145173611	3.70181763	1.619427954	6.37086075
f_{25}	1.548632682	1.280864537	2.590304259	1.00000000	1.445512785	1.129499291

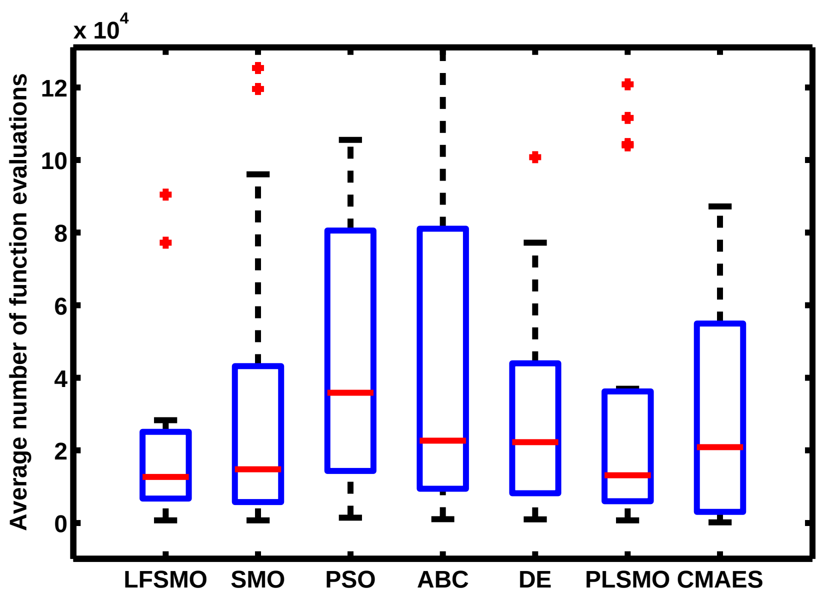


Figure 3 Boxplot graph for average number of function evaluations

$$x = [P_{g1}, V_{l1}....V_{lnl}, Q_{g1}....Q_{gng}, S_{l1}....S_{lNl}] \tag{21}$$

where ng, nc, nt, nl, Nl are the number of generators, number of shunt VAR compensators, number of regulating transformers, number of load buses and number of transmission lines, respectively.

$G(u, x)$ is the set of equality constraints which represents typical load flow equations as follows:

$$P_{gi} - P_{di} - V_i \sum_{k=1}^{nb} V_k \ (G_{ik} \ \cos\theta_{ik} + B_{ik} \ \sin\theta_{ik}) = 0 \tag{22}$$

and

$$Q_{gi} - Q_{di} - V_i \sum_{k=1}^{nb} V_k \ (G_{ik} \ \sin\theta_{ik} + B_{ik} \ \cos\theta_{ik}) = 0 \tag{23}$$

where P_{gi} and Q_{gi} are the active and reactive powers at i^{th} generators, P_{di} and Q_{di} are the active and reactive power demands at i^{th} bus, G_{ik} and B_{ik} are the transfer conductance and susceptance between buses i and k, respectively, θ_{ik} is the phase angle difference between the voltages at buses i and k and nb is the total number of bus bars.

$H(u, x)$ is the set of system operational limiting constraints which includes following inequality constraints:

- Generator constraints:

$$P_{gi}^{min} \leq P_{gi} \leq P_{gi}^{max}, \text{for } i = 1, 2, ...ng \tag{24}$$

$$Q_{gi}^{min} \leq Q_{gi} \leq Q_{gi}^{max}, \text{for } i = 1, 2, ...ng \tag{25}$$

$$V_{gi}^{min} \leq V_{gi} \leq V_{gi}^{max}, \text{for } i = 1, 2, ...ng \tag{26}$$

- Security constraints:

$$V_{li}^{min} \leq V_{li} \leq V_{li}^{max}, \text{for } i = 1, 2, ...nl \tag{27}$$

$$S_{li} \leq S_{li}^{max}, \text{for } i = 1, 2, ...Nl \tag{28}$$

- Transformer constraints:

$$T_{i}^{min} \leq T_{i} \leq T_{i}^{max}, \text{for } i = 1, 2, ...nt \tag{29}$$

- Shunt VAR compensator constraints:

$$Q_{ci}^{min} \leq Q_{ci} \leq Q_{ci}^{max}, \text{for } i = 1, 2, ...nc \tag{30}$$

There are a variety of techniques available in the literature to tackle constraints in optimization algorithms. In this work, penalty function method is chosen to address constraints with respect to dependent variables, i.e., if any dependent variable violates its bound then a square of that violation amount multiplied by a fix penalty factor is added to its corresponding fitness function so that infeasible solutions can be rejected. On the other hand, the constraints corresponding to control variables are handled by generating them between the given bounds in its initialization phase. In this way, the modified objective function for OPF is of the following form:

$$\begin{aligned} \text{Min } F_{mod} = F(u, x) + \lambda_p (P_{g1} - P_{g1}^{lim})^2 + \lambda_v \sum_{i=1}^{nl} (V_{li} - V_{li}^{lim})^2 + \\ \lambda_q \sum_{i=1}^{ng} (Q_{gi} - Q_{gi}^{lim})^2 + \lambda_s \sum_{i=1}^{Nl} (S_{li} - S_{li}^{lim})^2 \end{aligned} \tag{31}$$

where λ_p, λ_v, λ_q, and λ_s are the penalty factors (all are set to 10^5) and a^{lim} is the limit value (may be min or max limit).

8 Application of LFSMO to solve OPF problem

To see it's robustness, the proposed LFSMO algorithm is applied to solve OPF IEEE 30 bus problem. For this, all members of the population are first initializing each member of the population which represents a potential solution. The configuration of all individual member is $u = (P_{g2}....P_{gng}, V_{g1}....V_{gng}, Q_{c1}....Q_{cnc}, T_1....T_{nt})$ where all involved variables are self-constrained i.e., all control variables of individual member, u is initialized arbitrarily within tolerable limits given in Table 12. If any control variable violates its limits, then that variable is adjusted to the consequently overstepped limits. To tackle the inequality constraints of dependent variables, modified objective function as shown in Equation 31 is considered to be optimized.

The LFSMO is applied on IEEE 30 bus system to validate its usefulness. It consists of 6 generating units at buses 1, 2, 5, 8, 11 and 13 interconnected with 41 transmission lines with a load of 283.4 MW and 126.2 MVAR and 4 transformers with off nominal tap ratios in lines $6-9$, $6-10$, $4-12$ and $28-27$. The bus data and the branch data are taken from (Alsac & Stott, 1974). The shunt injections are provided at buses 10, 12, 15, 17, 20, 21, 23, 24, and 29 from literature (Abido, 2002a).

Upper and lower limits of control variables are as mentioned in Table 12. The objective functions to be minimized validate the usefulness of LFSMO are: 1. Total quadratic fuel cost. 2. Piece wise quadratic fuel cost 3. Quadratic fuel cost with valve-point effects to confirm the robustness of proposed LFSMO algorithm. In each case, OPF problem is executed for 100 runs.

Case 1: Quadratic fuel cost function

Objective function to be optimized is selected from Equation 31 and generator cost characteristic are defined as quadratic cost function of generator power output. In Equation 31 the $F(u, x)$ is designed for this case as:

$$F(u, x) = \sum_{i=1}^{ng} f_i(P_{gi}) = \sum_{i=1}^{ng} (a_i + b_i P_{gi} + c_i P_{gi}^2) \tag{32}$$

where f_i and P_{gi} are fuel cost and active power of i^{th} generator, respectively. The variables a_i, b_i and c_i represent the cost coefficients of i^{th} generator whose values have been taken from standard IEEE 30-bus system and are given here in Table 9, ng is the total number of generators in the system.

Case 2: Piecewise quadratic fuel cost functions

The fuel cost functions of various fuels supplied in a power system such as coal, natural gas and oil may be dissevered as piecewise quadratic fuel cost functions (Abou El Ela, Abido, & Spea, 2010). In the present case, the objective is to minimize fuel cost of each unit with satisfying the system constraints. Since it becomes a complex problem, the application of traditional mathematical optimization technique is not positively applicable (Dieu & Ongsakul, 2006). For generating units connected at bus 1 and 2, the cost characteristics are represented by a piecewise quadratic cost function to model different fuels are defined as follows:

$$f_i(P_{gi}) = \begin{cases} a_{i1} + b_{i1} P_{gi} + c_{i1} P_{gi}^2 \ , & P_{gi}^{min} \le P_{gi} \le P_{gi1} \\ a_{i2} + b_{i2} P_{gi} + c_{i2} P_{gi}^2 \ , & P_{gi1} \le P_{gi} \le P_{gi2} \\ \ldots\ldots\ldots\ldots & \\ a_{ik} + b_{ik} P_{gi} + c_{ik} P_{gi}^2 \ , & P_{gik-1} \le P_{gi} \le P_{gi}^{max} \end{cases} \tag{33}$$

where a_{ik}, b_{ik} and c_{ik} are cost coefficients of the i^{th} generator for fuel type k. So, here Equation 31 is selected as objective function to be optimized where the $F(u, x)$ is designed for this case as:

$$F(u, x) = \sum_{i=1}^{ng} f_i(P_{gi}) = \sum_{i=1}^{2} f_i(P_{gi}) + \sum_{i=3}^{ng} (a_i + b_i P_{gi} + c_i P_{gi}^2) \tag{34}$$

where $f_i(P_{gi})$ for generating units 1 and 2 are selected based on Equation 33 and the cost coefficients for these units are given in Table 10 and the cost coefficients of other generators have the same values as of case 1.

Case 3: Quadratic fuel cost function with valve-point effect

The generating units for buses 1 and 2 are assumed to have the characteristic with valve-point effects.

Cost coefficient for generating buses 1 and 2 are adopted from (Vaisakh & Srinivas, 2011) and given in Table 11. The cost coefficients for all other generating units are same as of case 1. The cost characteristics of generating units 1 and 2 are defined as:

$$f_i(P_{gi}) = a_i + b_iP_{gi} + c_iP_{gi}^2 + |d_i \sin(e_i(P_{gi}^{min} - P_{gi}))| \text{ , where } i = 1 \text{ and } 2 \tag{35}$$

where a_i, b_i, c_i, d_i and e_i are cost coefficients of the i^{th} generating unit. So, here Equation 31 is selected as objective function to be optimized where the $F(u, x)$ is designed for this case as:

$$F(u, x) = \sum_{i=1}^{ng} f_i(P_{gi}) = \sum_{i=1}^{2}(a_i + b_iP_{gi} + c_iP_{gi}^2 + |d_i \sin(e_i(P_{gi}^{min} - P_{gi}))|) + \sum_{i=3}^{ng}(a_i + b_iP_{gi} + c_iP_{gi}^2) \tag{36}$$

8.1 Experimental Settings for OPF Problem

Following experimental settings is adopted for OPF parameters:

- The cost coefficients for all cases are given in Tables 9-11,
- The upper and lower limits for control variables are taken from (Lee, Park, & Ortiz, 1985) and are given in Table 12,
- The load data are adopted from (Alsac & Stott, 1974; Lee et al., 1985) and given in Table 13,
- The line data for 30-bus system are taken from (Alsac & Stott, 1974; Lee et al., 1985),
- The penalty factors λ_p, λ_v, λ_q, and λ_s are set to 10^5.

8.2 Results Analysis and Discussion

Result for all three cases i.e. case 1, 2, and 3 are shown in Table 5. A fair comparison of LFSMO with SMO and other state-of-art algorithms has been presented in Table 6-8 regarding minimum fuel cost and average minimum fuel cost over 100 runs. After analyzing the results of all three cases. It is clear that LFSMO is more robust than the SMO and other considered algorithms.

Since the empirical distribution of results can efficiently be represented by boxplot (Williamson, Parker, & Kendrick, 1989), the boxplots for a minimum fuel cost of 100 runs for LFSMO and basic SMO algorithms have been represented in Figure 4 to analyze the algorithms output more intensively. Figure 4 shows that LFSMO is cost effective as the interquartile range, and median are quite small for LFSMO.

Therefore, now it may be stated that the OPF IEEE-30 bus problem is solved through LFSMO efficiently. As the OPF problem is multi-modal and non-linear in nature, the structured swarm based searching process and fission-fusion grouping pattern of LFSMO makes it more efficient to trace out the solution. But, as LFSMO having more parameters as compared to the PSO, ABC, DE, GA, and other meta-heuristics, therefore, a user should be more cautious while tuning the parameters. Further, it can be observed from Table 3 that LFSMO performs well on the non-separable and multi-modal functions only so it is recommended to apply it to the similar type of problems.

Table 5 Best control variable settings achieved by LFSMO algorithms for different cases.

Control variables	Case 1: Quadratic fuel cost function	Case 2: Piecewise quadratic fuel cost function	Case 3: Quadratic fuel cost function with valve point effects
P_1	177.0576	139.982	199.5956
P_2	48.6821175	54.99991003	20
P_5	21.3758855	23.83664713	22.15063489
P_8	21.2910636	34.42954204	24.6868717
P_{11}	11.9982644	18.30230681	13.16136133
P_{13}	12	18.62484262	13.490919
V_1	1.0851374	1.07313073	1.08603294
V_2	1.01432154	1.03749276	1.00931453
V_5	1.03427706	1.0296065	1.03003395
V_8	1.04842306	1.03451984	1.0663202
V_{11}	1.08255052	1.07820535	1.07608792
V_{13}	1.04211719	1.06426709	1.05464662
T_{11}	1.0304135	0.99468271	1.02417779
P_{12}	0.95404543	1.0280758	0.95513904
P_{15}	0.95920934	1.00516395	0.97412619
P_{36}	0.97495386	0.97340406	0.96460148
Q_{c10}	5	5	0.96157014
Q_{c12}	5	5	4.77382674
Q_{c15}	4.91791195	4.80873055	0.35764546
Q_{c17}	5	5	2.22120068
Q_{c20}	4.28077149	4.96509578	4.94296835
Q_{c21}	5	5	5
Q_{c23}	3.1728881	4.16856678	1.38973559
Q_{c24}	5	5	4.96546555
Q_{c29}	2.55092117	2.20926233	2.59359073
Fuel cost ($/h)	**800.4474**	**646.6704**	**918.9122**

Table 6 Comparison of 100 runs for Case-1 among different methods for IEEE 30-bus system.

Optimization methods	Fuel cost ($/h)	
	Min	Average
ITS (Ongsakul & Tantimaporn, 2006)	804.5560	-
EP (Yuryevich & Wong, 1999)	802.6300	803.5100
IEP (Ongsakul & Tantimaporn, 2006)	802.4650	802.5210
DE-OPF (Sayah & Zehar, 2008)	802.3940	-
MDE-OPF (Sayah & Zehar, 2008)	802.3760	802.3820
TS (Ongsakul & Tantimaporn, 2006)	802.5020	-
TS/SA (Ongsakul & Tantimaporn, 2006)	802.7880	-
SADE-ALM (Vaisakh & Srinivas, 2011)	802.4040	-
Enhanced GA (Bakirtzis et al., 2002)	802.0600	-
PSO (Abido, 2002a)	800.4890	-
ABC-OPF (Bansal, Jadon, Tiwari, Kiran, & Panigrahi, 2014)	802.9086	803.6341
SMO	**800.4840253**	**800.684232**
LFSMO	**800.4474**	**800.4795298**

Table 7 Comparison of 100 runs for Case-2 among different methods for IEEE 30-bus system.

Optimization methods	Fuel cost ($/h)	
	Min	Average
ITS (Ongsakul & Tantimaporn, 2006)	654.8740	-
EP (Yuryevich & Wong, 1999)	647.7900	649.7000
IEP (Ongsakul & Tantimaporn, 2006)	649.3120	650.2170
DE-OPF (Sayah & Zehar, 2008)	648.3840	-
MDE-OPF (Sayah & Zehar, 2008)	647.8460	648.3560
TS (Ongsakul & Tantimaporn, 2006)	651.2460	-
TS/SA (Ongsakul & Tantimaporn, 2006)	654.3780	-
PSO (Abido, 2002a)	647.6900	-
GSA (Duman, Güvenç, Sönmez, & Yörükeren, 2012)	646.8480	646.8962
BBO (Bhattacharya & Chattopadhyay, 2011)	647.7430	647.7645
ABC-OPF (Bansal, Jadon, et al., 2014)	648.9124	649.4393
SMO	**646.820864**	**646.947322**
LFSMO	**646.6704**	**646.6904559**

Table 8 Comparison of 100 runs for Case-3 among different methods for IEEE 30-bus system.

Optimization methods	Fuel cost ($/h)	
	Min	Average
ITS (Ongsakul & Tantimaporn, 2006)	969.1090	-
EP (Vaisakh & Srinivas, 2011)	955.5090	959.3630
IEP (Ongsakul & Tantimaporn, 2006)	953.5730	956.4600
DE-OPF (Sayah & Zehar, 2008)	931.0850	-
MDE-OPF (Sayah & Zehar, 2008)	930.793	942.5010
TS (Ongsakul & Tantimaporn, 2006)	956.0000	-
TS/SA (Ongsakul & Tantimaporn, 2006)	959.5630	-
EADDE (Vaisakh & Srinivas, 2011)	930.745	-
SADE-ALM (Vaisakh & Srinivas, 2011)	944.031	-
BBO (Bhattacharya & Chattopadhyay, 2011)	919.7647	919.8389
GSA (Duman et al., 2012)	929.7260	930.9240
ABC-OPF (Bansal, Jadon, et al., 2014)	930.4153	931.2629
SMO	**918.9173232**	**919.5698616**
LFSMO	**918.9122079**	**918.9829993**

Table 9 Generator Cost Coefficients for Case 1.

Cost coefficients	Bus No.					
	1	2	5	8	11	13
a	0.00	0.00	0.00	0.00	0.00	0.00
b	2.00	1.75	1.00	3.25	3.00	3.00
c	0.00375	0.01750	0.06250	0.00834	0.02500	0.02500

Table 10 Generator Cost Coefficients for Case 2.

Bus No.	From MW	To MW	Cost coefficients		
			a	b	c
1	50	140	55.00	0.70	0.0050
	140	200	82.50	1.05	0.0075
2	20	55	40.00	0.30	0.0100
	55	80	80.00	0.60	0.0200

Table 11 Generator Cost Coefficients for Case 3.

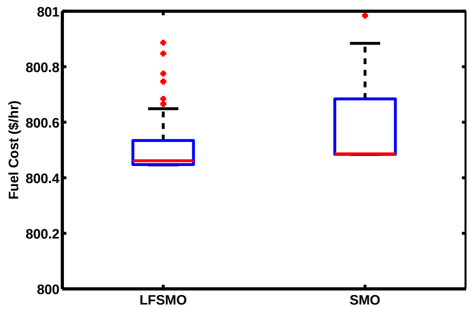
Bus No.	P_{gi}^{min}	Cost coefficients				
		a	b	c	d	e
1	50	150.00	2.00	0.0016	50.00	0.0630
2	20	25.00	2.50	0.0100	40.00	0.0980

Table 12 The Upper and Lower Limits of Control Variables.

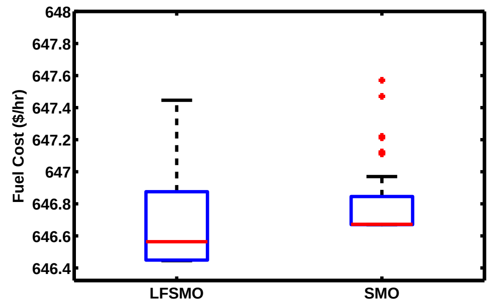
Control variables	Min	Max	Control variables	Min	Max
P_1	50	200	T_{11}	0.9	1.1
P_2	20	80	T_{12}	0.9	1.1
P_5	15	50	T_{15}	0.9	1.1
P_8	10	35	T_{36}	0.9	1.1
P_{11}	10	30	Q_{c10}	0	5
P_{13}	12	40	Q_{c12}	0	5
V_1	0.95	1.1	Q_{c15}	0	5
V_2	0.95	1.1	Q_{c17}	0	5
V_5	0.95	1.1	Q_{c20}	0	5
V_8	0.95	1.1	Q_{c21}	0	5
V_{11}	0.95	1.1	Q_{c23}	0	5
V_{13}	0.95	1.1	Q_{c24}	0	5
			Q_{c29}	0	5

Table 13 Load Data.

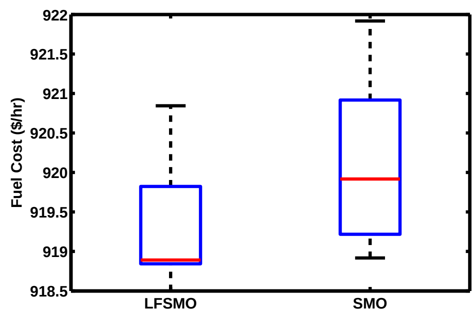
Bus No.	Load		Bus No.	Load		Bus No.	Load	
	P	Q		P	Q		P	Q
1	0.000	0.000	11	0.000	0.000	21	0.175	0.112
2	0.217	0.127	12	0.112	0.075	22	0.000	0.000
3	0.024	0.012	13	0.000	0.000	23	0.032	0.016
4	0.076	0.016	14	0.062	0.016	24	0.087	0.067
5	0.942	0.190	15	0.082	0.025	25	0.000	0.000
6	0.000	0.000	16	0.035	0.018	26	0.035	0.023
7	0.228	0.109	17	0.090	0.058	27	0.000	0.000
8	0.300	0.300	18	0.032	0.009	28	0.000	0.000
9	0.000	0.000	19	0.095	0.034	29	0.024	0.009
10	0.058	0.020	20	0.022	0.007	30	0.106	0.019



(a) case 1



(b) case 2



(c) case 3

Figure 4 Boxplots for LFSMO and SMO for minimum fuel cost for 100 runs.

9 Conclusion

In this paper, a solution to the OPF problem, which is a non-convex, highly constrained, nonlinear, and multimodular optimization problem, is considered. After describing the limitations of the available deterministic mathematical methods, the OPF problem is solved using a modified version of spider monkey optimization (SMO) algorithm, lévy flight SMO (LFSMO). Initially, the efficiency of LFSMO is established through experiments over 25 well known benchmark functions. Then the proposed LFSMO is applied to solve the IEEE 30-bus OPF problem for three different cases namely, total quadratic fuel cost, piecewise quadratic fuel cost, and quadratic fuel cost with valve-point effects.

The reported results are compared with SMO, PSO, DE, GA, and other state-of-art algorithms presented in the literature. It is shown that LFSMO is a competitive algorithm in the field of swarm intelligence based algorithms to solve the OPF problem. Therefore, in future, this strategy may be applied to solve other problems having same as OPF problem.

References

Abido, M. (2002a). Optimal power flow using particle swarm optimization. *International Journal of Electrical Power & Energy Systems*, *24*(7), 563–571. Retrieved from http://www.sciencedirect.com/science/article/pii/S0142061501000679 doi: 10.1016/S0142-0615(01)00067-9

Abido, M. (2002b). Optimal power flow using tabu search algorithm. *Electric Power Components and Systems*, *30*(5), 469–483. doi: 10.1080/15325000252888425

Abou El Ela, A., Abido, M., & Spea, S. (2010). Optimal power flow using differential evolution algorithm. *Electric Power Systems Research*, *80*(7), 878–885. Retrieved from http://www.sciencedirect.com/science/article/pii/S0378779610000088 doi: 10.1016/j.epsr.2009.12.018

Ali, M., Khompatraporn, C., & Zabinsky, Z. (2005). A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. *Journal of Global Optimization*, *31*(4), 635–672. Retrieved from http://link.springer.com/article/10.10072Fs10898-004-9972-2 doi: 10.1007/s10898-004-9972-2

Al-Muhawesh, T. A., & Qamber, I. S. (2008). The established mega watt linear programming-based optimal power flow model applied to the real power 56-bus system in eastern province of saudi arabia. *Energy*, *33*(1), 12–21. Retrieved from http://www.sciencedirect.com/science/article/pii/S0360544207001478 doi: 10.1016/j.energy.2007.08.004

Alsac, O., & Stott, B. (1974). Optimal load flow with steady-state security. *Power Apparatus and Systems, IEEE Transactions on*, *PAS-93*(3), 745–751. doi: 10.1109/TPAS.1974.293972

Bakirtzis, A. G., Biskas, P. N., Zoumas, C. E., & Petridis, V. (2002). Optimal power flow by enhanced genetic algorithm. *Power Systems, IEEE Transactions on*, *17*(2), 229–236. doi: 10.1109/TPWRS.2002.1007886

Bansal, J. C., Jadon, S. S., Tiwari, R., Kiran, D., & Panigrahi, B. K. (2014). Optimal power flow using artificial bee colony algorithm with global and local neighborhoods. *International Journal of System Assurance Engineering and Management*, *1*(1), 1-12. Retrieved from http://link.springer.com/article/10.10072Fs13198-014-0321-7 doi: 10.1007/s13198-014-0321-7

Bansal, J. C., Sharma, H., Jadon, S. S., & Clerc, M. (2014). Spider monkey optimization algorithm for numerical optimization. *Memetic Computing*, *6*(1), 31–47. Retrieved from http://link.springer.com/article/10.10072Fs12293-013-0128-0 doi: 10.1007/s12293-013-0128-0

Bhattacharya, A., & Chattopadhyay, P. (2011). Application of biogeography-based optimisation to solve different optimal power flow problems. *IET generation, transmission & distribution*, *5*(1), 70–80. doi: 10.1049/iet-gtd.2010.0237

Caponio, A., Neri, F., & Tirronen, V. (2009). Super-fit control adaptation in memetic differential evolution frameworks. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, *13*(8), 811–831. Retrieved from http://link.springer.com/article/10.10072Fs00500-008-0357-1 doi: 10.1007/s00500-008-0357-1

Carpentier, J. (1962). Contribution to the economic dispatch problem. *Bulletin de la Societe Francoise des Electriciens*, *3*(8), 431–447.

Carpentier, J. (1979). Optimal power flows. *International Journal of Electrical Power & Energy Systems*, *1*(1), 3–15. Retrieved from http://www.sciencedirect.com/science/article/pii/0142061579900267 doi: doi:10.1016/0142-0615(79)90026-7

Chen, X., Ong, Y., Lim, M., & Tan, K. (2011). A multi-facet survey on memetic computation. *IEEE Transactions on Evolutionary Computation*, *15*(5), 591–607. doi: 10.1109/TEVC.2011.2132725

Cotta, C., & Neri, F. (2012). Memetic algorithms in continuous optimization. *Handbook of Memetic Algorithms*, *379*, 121–134. (Springer Berlin Heidelberg)

Dieu, V. N., & Ongsakul, W. (2006). Enhanced augmented lagrange hopfield network for economic dispatch with piecewise quadratic cost functions. *fuel*, *2*(2), 2–2.

Dommel, H. W., & Tinney, W. F. (1968). Optimal power flow solutions. *power apparatus and systems, IEEE transactions on*, *PAS 87*(10), 1866–1876. doi: 10.1109/TPAS.1968.292150

Duman, S., Güvenç, U., Sönmez, Y., & Yörükeren, N. (2012). Optimal power flow using gravitational search algorithm. *Energy Conversion and Management*, *59*(July), 86-95. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0196890412000957?np=y` doi: 10.1016/j.enconman.2012.02.024

Fister Jr, I., Yang, X.-S., Fister, I., & Brest, J. (2012). Memetic firefly algorithm for combinatorial optimization. *arXiv preprint arXiv:1204.5165*.

Fister Jr, I., Yang, X.-S., Fister, I., Brest, J., & Fister, D. (2013). A brief review of nature-inspired algorithms for optimization. *arXiv preprint arXiv:1307.4186*.

Gallo, C., Carballido, J., & Ponzoni, I. (2009). Bihea: A hybrid evolutionary approach for microarray biclustering. *Advances in Bioinformatics and Computational Biology*, 36–47. doi: 10.1007/978-3-642-03223-3-4

Goh, C., Ong, Y., & Tan, K. (2009). *Multi-objective memetic algorithms* (Vol. 171). Heidelberg, Berlin: Springer Verlag.

Granville, S. (1994). Optimal reactive dispatch through interior point methods. *Power Systems, IEEE Transactions on*, *9*(1), 136–146. doi: 10.1109/59.317548

Habibollahzadeh, H., Luo, G.-X., & Semlyen, A. (1989). Hydrothermal optimal power flow based on a combined linear and nonlinear programming methodology. *Power Systems, IEEE Transactions on*, *4*(2), 530–537. doi: 10.1109/59.193826

Kang, F., Li, J., & Ma, Z. (2011). Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions. *Information Sciences*, *181(16)*, 3508–3531. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0020025511001988` doi: doi:10.1016/j.ins.2011.04.024

Knowles, J., Corne, D., & Deb, K. (2008). *Multiobjective problem solving from nature: From concepts to applications (natural computing series)*. Springer-Verlag Berlin Heidelberg. Retrieved from `http://www.springer.com/us/book/9783540729631` doi: 10.1007/978-3-540-72964-8

Lee, K., Park, Y., & Ortiz, J. (1985). A united approach to optimal real and reactive power dispatch. *Power Apparatus and Systems, IEEE Transactions on*, *PAS 104*(5), 1147–1153. doi: 10.1109/TPAS.1985.323466

Lotfipour, A., & Afrakhte, H. (2016). A discrete teaching–learning-based optimization algorithm to solve distribution system reconfiguration in presence of distributed generation. *International Journal of Electrical Power & Energy Systems*, *82*(In progress, November 2016), 264–273. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0142061516303672?np=y` doi: 10.1016/j.ijepes.2016.03.009

Mezura-Montes, E., & Velez-Koeppel, R. (2010). Elitist artificial bee colony for constrained real-parameter optimization. In *2010 congress on evolutionary computation (cec2010), ieee service center, barcelona, spain* (pp. 2068–2075). doi: 10.1109/CEC.2010.5586280

Mininno, E., & Neri, F. (2010). A memetic differential evolution approach in noisy optimization. *Memetic Computing*, *2*(2), 111–135. doi: 10.1007/s12293-009-0029-4

Mota-Palomino, R., & Quintana, V. (1986). Sparse reactive power scheduling by a penalty function-linear programming technique. *Power Systems, IEEE Transactions on*, *1*(3), 31–39. doi: 10.1109/TPWRS.1986.4334951

Neri, C. M. P. E., Ferrante; Cotta. (Vol. 379, 2012). *Handbook of memetic algorithms* (Vol. 379; S. S. in Computational Intelligence., Ed.). Springer Berlin Heidelberg: Springer, Studies in Computational Intelligence,. Retrieved from `http://www.springer.com/us/book/9783642232466`

Neri, F., & Tirronen, V. (2009). Scale factor local search in differential evolution. *Memetic Computing, Springer*, *1*(2), 153–171. Retrieved from `http://link.springer.com/article/10.10072Fs12293-009-0008-9` doi: 10.1007/s12293-009-0008-9

Ong, Y., Lim, M., & Chen, X. (2010). Research frontier: Memetic computation-past, present & future. *Computational Intelligence Magazine, IEEE*, *5*(2), 24–31. doi: 10.1109/MCI.2010.936309

Ongsakul, W., & Tantimaporn, T. (2006). Optimal power flow by improved evolutionary programming. *Electric Power Components and Systems*, *34*(1), 79–95. doi: 10.1080/15325000691001458

Rao, S., & Rao, S. (2009). *Engineering optimization: theory and practice.* New Delhi, India: John Wiley & Sons.

Reddy, S. S., & Bijwe, P. (2016). Efficiency improvements in meta-heuristic algorithms to solve the optimal power flow problem. *International Journal of Electrical Power & Energy Systems*, *82*(In Progress, November 2016), 288–302. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0142061516304410` doi: 10.1016/j.ijepes.2016.03.028

Repoussis, P., Tarantilis, C., & Ioannou, G. (2009). Arc-guided evolutionary algorithm for the vehicle routing problem with time windows. *Evolutionary Computation, IEEE Transactions on*, *13*(3), 624–647. doi: 10.1109/TEVC.2008.2011740

Richer, J., Goëffon, A., & Hao, J. (2009). A memetic algorithm for phylogenetic reconstruction with maximum parsimony. *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, 164–175. (Springer, Tubigen, Germany) doi: 10.1007/978-3-642-01184-9-15

Roa-Sepulveda, C., & Pavez-Lazo, B. (2003). A solution to the optimal power flow using simulated annealing. *International journal of electrical power & energy systems*, *25*(1), 47–57. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0142061502000200` doi: 10.1016/S0142-0615(02)00020-0

Sayah, S., & Zehar, K. (2008). Modified differential evolution algorithm for optimal power flow with non-smooth cost functions. *Energy Conversion and Management*, *49*(11), 3036–3042. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0196890408002380` doi: 10.1016/j.enconman.2008.06.014

Sharma, A., Sharma, H., Bhargava, A., & Sharma, N. (2016). Power law-based local search in spider monkey optimization for lower order system modeling. *International Journal of Systems Science*, *In Progress*, 1-11. Retrieved from `http://www.tandfonline.com/doi/full/10.1080/00207721.2016.1165895` doi: 10.1080/00207721.2016.1165895

Sharma, A., Sharma, H., Bhargava, A., Sharma, N., & Bansal, J. C. (2016). Optimal placement and sizing of capacitor using limaçon inspired spider monkey optimization algorithm. *Memetic Computing*, 1–21. Retrieved from `http://dx.doi.org/10.1007/s12293-016-0208-z` doi: 10.1007/s12293-016-0208-z

Sharma, H., Bansal, J. C., & Arya, K. (2013). Opposition based lévy flight artificial bee colony. *Memetic Computing*, *5*(3), 213–227. doi: 10.1007/s12293-012-0104-0

Sharma, H., Bansal, J. C., & Arya, K. (2014). Self balanced differential evolution. *Journal of Computational Science*, *5*(2), 312–323. Retrieved from `http://www.sciencedirect.com/science/article/pii/S1877750312001251` doi: 10.1016/j.jocs.2012.12.002

Sharma, H., Bansal, J. C., Arya, K., & Yang, X.-S. (2015). Lévy flight artificial bee colony algorithm. *International Journal of Systems Science*, *47*(11), 312-323. doi: 10.1080/00207721.2015.1010748

Sharma, H., Jadon, S. S., Bansal, J. C., & Arya, K. (2013). Lèvy flight based local search in differential evolution. In *Swarm, evolutionary, and memetic computing* (pp. 248–259). Chennai, India: Springer International PSpringer. doi: 10.1007/978-3-319-03753-0-23

Singh, M., & Dhillon, J. (2016). Multiobjective thermal power dispatch using opposition-based greedy heuristic search. *International Journal of Electrical Power & Energy Systems*, *82*(In progress, November 2016), 339–353. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0142061516303805` doi: 10.1016/j.ijepes.2016.03.016

Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K., Chen, Y.-P., Auger, A., & Tiwari, S. (2005). Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. *KanGAL report CEC 2005*(2005005), 1-50. Retrieved from `https://www.lri.fr/ hansen/Tech-Report-May-30-05.pdf` (Edinburgh, UK)

Sun, D. I., Ashley, B., Brewer, B., Hughes, A., & Tinney, W. F. (1984). Optimal power flow by newton approach. *power apparatus and systems, ieee transactions on*, *PAS 103*(10), 2864–2880. doi: 10.1109/TPAS.1984.318284

Tang, K., Mei, Y., & Yao, X. (2009). Memetic algorithm with extended neighborhood search for capacitated arc routing problems. *IEEE Transactions on Evolutionary Computation*, *13*(5), 1151–1166. doi: 10.1109/TEVC.2009.2023449

Vaisakh, K., & Srinivas, L. (2011). Evolving ant direction differential evolution for opf with non-smooth cost functions. *Engineering Applications of Artificial Intelligence*, *24*(3), 426–436. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0952197610002058` doi: 10.1109/TPAS.1984.318284

Wang, H., Wang, D., & Yang, S. (2009). A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problems. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, *13*(8), 763–780. doi: 10.1007/s00500-008-0347-3

Wei, H., Sasaki, H., Kubokawa, J., & Yokoyama, R. (1998). An interior point nonlinear programming for optimal power flow problems with a novel data structure. *Power Systems, IEEE Transactions on*, *13*(3), 870–877. doi: 10.1109/59.708745

Williamson, D., Parker, R., & Kendrick, J. (1989). The box plot: a simple visual method to interpret data. *Annals of internal medicine*, *110*(11), 916. Retrieved from `http://annals.org/article.aspx?articleid=703149` doi: 10.7326/0003-4819-110-11-916

Yadav, A., Deep, K., Kim, J. H., & Nagar, A. K. (2016). Gravitational swarm optimizer for global optimization. *Swarm and Evolutionary Computation*.

Yadav, A., Yadav, N., & Kim, J. H. (2016). A study of harmony search algorithms: Exploration and convergence ability. In *Harmony search algorithm* (pp. 53–62). Springer.

Yang, X. (2011). *Nature-inspired metaheuristic algorithms*. Luniver Press, United Kingdom.

Yuryevich, J., & Wong, K. P. (1999). Evolutionary programming based optimal power flow algorithm. *Power Systems, IEEE Transactions on*, *14*(4), 1245–1250. doi: 10.1109/59.801880