

Spider Monkey Optimization Algorithm for Numerical Optimization

Jagdish Chand Bansal · Harish Sharma · Shimpi Singh
Jadon · Maurice Clerc

the date of receipt and acceptance should be inserted later

Abstract Swarm intelligence is a fascinating area for the researchers in the field of optimization. Researchers have developed many algorithms by simulating the swarming behavior of various creatures like ants, honey bees, fishes, birds and their findings are very motivating. In this paper, a new approach for optimization is proposed by modeling the social behavior of spider monkeys. Spider monkeys have been categorized as fission-fusion social structure based animals. The animals which follow fission-fusion social systems, initially work in a large group and based on need after some time, they divide themselves in smaller groups led by an adult female for foraging. Therefore, the proposed strategy broadly classified as inspiration from the intelligent foraging behavior of fission-fusion social structure based animals.

1 Introduction

The name swarm is used for an accumulation of creatures such as ants, fishes, birds, termites and honey bees which behaves collectively. The definition given by Bonabeau for the swarm intelligence is “any attempt to design algorithms or distributed problem-solving devices inspired by the collective behaviour of social insect colonies and other animal societies” [2].

Swarm Intelligence is a meta-heuristic approach in the field of nature inspired techniques that is used to solve optimization problems. It is based on the collective behavior of social creatures. Social creatures utilizes their ability of social learning to solve complex tasks. Researchers have analyzed such behaviors and designed algorithms that can be used to solve nonlinear, nonconvex or combinatorial optimization problems in many science and engineering domains. Previous research [7, 12, 19, 29] have shown that algorithms based on Swarm Intelligence have great potential to find a solution of real world optimization problem. The algorithms that have emerged in recent years include Ant

Jagdish Chand Bansal
ABV-Indian Institute of Information Technology and Management, Gwalior
Tel.: +0751-2449819,
Fax: + 0751-2449813
E-mail: jcbansal@gmail.com

Harish Sharma
ABV-Indian Institute of Information Technology and Management, Gwalior
Tel.: +09479810157,
Fax: + 0751-2449813
E-mail: harish.sharma0107@gmail.com

Shimpi Singh Jadon
ABV-Indian Institute of Information Technology and Management, Gwalior
Tel.: +0751-2449819,
Fax: + 0751-2449813
E-mail: shimpisingh2k6@gmail.com

Maurice Clerc
E-mail: Maurice.Clerc@WriteMe.com

Colony Optimization (ACO) [7], Particle Swarm Optimization (PSO) [12], Bacterial Foraging Optimization (BFO) [17], Artificial Bee Colony Optimization (ABC) [10] etc.

As shown in Fig. 1, the necessary and sufficient properties for obtaining intelligent swarming behaviors of animals are self-organization and division of labour. Each of the properties are explained as follows:

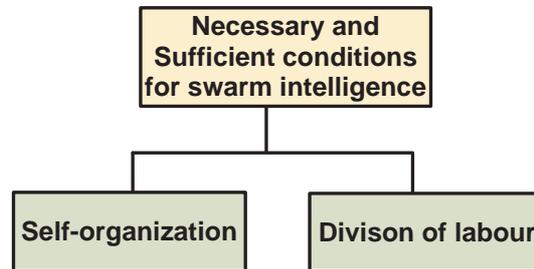


Fig. 1: Conditions for intelligent swarming

1. **Self-organization:** is an important feature of a swarm structure which results global level response by means of interactions among its low-level components without a central authority or external element enforcing it through planning. Therefore, the globally coherent pattern appears from the local interaction of the components that build up the structure, thus the organization is achieved by parallelly as all the elements act at the same time and distributed as no element is a central coordinator. Bonabeau et al. have defined following four important characteristics on which self organization is based: [2]
 - (i) **Positive feedback:** is an information extracted from the output of a system and reapplied to the input to promotes the creations of convenient structures. In the field of swarm intelligence positive feedback provides diversity and accelerate the system to new stable state.
 - (ii) **Negative feedback:** compensates the effect of positive feedback and helps to stabilize the collective pattern.
 - (iii) **Fluctuations:** are the rate or magnitude of random changes in the system. Randomness is often crucial for efflorescent structures since it allows the findings of new solutions. In foraging process, it helps to get-ride of stagnation.
 - (iv) **Multiple interactions:** provide the way of learning from the individuals within a society and thus enhance the combined intelligence of the swarm.
2. **Division of labour:** is a cooperative labour in specific, circumscribed tasks and like roles. In a group, there are various tasks, which are performed simultaneously by specialized individuals. Simultaneous task performance by cooperating specialized individuals is believed to be more efficient than the sequential task performance by unspecialized individuals [6, 9, 16].

2 Idea for a swarm based new Optimization algorithm

Fission-fusion swarm is a social grouping pattern in which individuals form temporary small parties (also called sub-groups) whose members belong to a larger community (or unit-group) of stable membership, there can be fluid movement between subgroups and unit-groups such that group composition and size changes frequently [27].

The fission-fusion social system of swarm may minimize direct foraging competition between group members, so they divide themselves into sub-groups in order to search food. The group members interact among themselves and with other group members, to maintain social bonds and territorial boundaries. In this society, social group sleep in one habitat together but forage in small sub-groups going off in different directions during the day. This form of social formation occurs in several species of primates like hamadryas, bonobo, chimpanzees, gelada baboons and spider monkeys. These societies change frequently in their size and composition, making up a strong social group called the 'parent group'. All the individual members of a faunal community comprise of permanent social networks and their capability to track changes in the environment varies according to their individual animal dynamics. In a fission-fusion society, the main

parent group can fission into smaller subgroups or individuals to adapt the environmental or social circumstances. For example, members of a group are separated from the main group in order to hunt or forage for food during the day, but at night they return to join (fusion) the primary group to share food and to take part in other activities [27].

The society of spider monkeys is one of the example of fission-fusion social structure. In subsequent subsections, a brief overview on swarming of spider monkeys is presented.

2.1 Social Organization and Behavior

The social organization of spider monkeys is related to fission-fusion social system. Fig. 2 shows the social organization of spider monkeys[23]. They are social animals and live in group of up to 50 individuals. Spider monkeys break up into

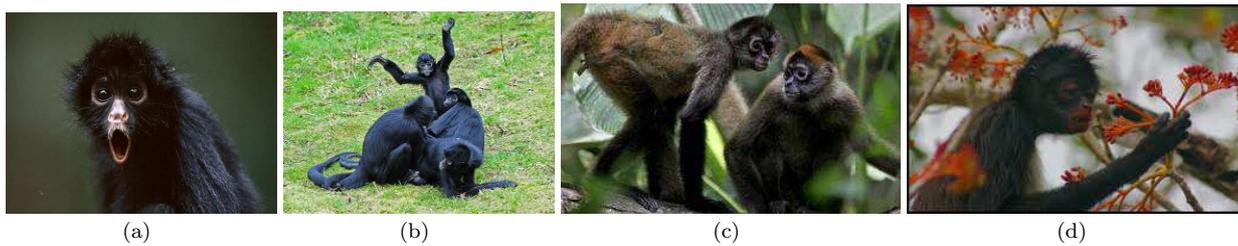


Fig. 2: Social Organization and Behavior (a) Spider-Monkey (b) Spider Monkey Group (c) Spider Monkey sub-group (d) Foods Foraging [23]

small foraging groups that travel together and forage throughout the day within a core area of the larger group's home range [24]. Spider monkeys find their foods in a very different way: a female leads the group and is responsible for finding food sources. In case if she doesn't find sufficient food for the group, she divides the group into smaller subgroups that forage separately [14]. The subgroups within the band are temporary and may vary in formation frequently throughout the day, but average three members ([28, 14]). When two different bands of spider monkeys come closer, the males in each band display aggressiveness and territorial behavior such as calling and barking. These communications occur with much distance between the two subgroups and do not involve any physical contacts, showing that groups respect distinct territory boundaries [28]. Members of a society might not ever be noticed closer at one place, but their mutual tolerance of each other when they come into contact reflects that they are component of the larger group [28]. The main reason behind emerging of fission-fusion social system is the food competition among the group members when there is shortage in food availability due to seasonal reasons [14]. When a big group gets food at particular location, there is likely to be less food per group member compare to a small group. After some time, when food scarcity is at its peak, average subgroup size is the smallest and during period of highest food availability, subgroup size is the largest, indicating that competition for scarce resources necessitates breaking into smaller foraging groups [28, 13]. One reason spider monkeys broke into smaller foraging groups but still remain part of a larger social unit is the advantage to individual group members in terms of increased mating chances and security from predators.

2.2 Communication

Spider monkeys share their intentions and observations using postures and positions, such as postures of sexual receptivity and of attack. During traveling, they interact with each other over long distances using a particular call which sounds like a horse's whinny. Each individual has its own discernible sound so that other members of the group can easily identify who is calling. This long-distance communication permits spider monkeys to get-together, stay away from enemies, share food and gossip. In order to interact to other group members, they generally use visual and vocal communication[20].

3 Spider Monkey Optimization Algorithm

Social behavior of spider monkeys inspires authors to develop an stochastic optimization technique that mimics the foraging behavior of spider monkeys. The foraging behavior of spider monkeys shows that these monkeys fall, in the category of fission-fusion social structure (FFSS) based animals. Thus the proposed optimization algorithm which is based on foraging behavior of spider monkeys can be explained better in terms of FFSS. Following are the key features of the FFSS.

1. The fission-fusion social structure based animals are social and live in groups of 40-50 individuals. The FFSS of swarm may reduce the foraging competition among group members by dividing them into sub-groups in order to search food.
2. A female (global Leader) generally leads the group and is responsible for searching food sources. If she is not able to get enough food for the group, she divides the group into smaller subgroups (size varies from 3 to 8 members) that forage independently.
3. Sub-groups are also supposed to be led by a female (local leader) who becomes decision-maker for planning an efficient foraging route each day.
4. The group members communicate among themselves and with other group members, to maintain social bonds and territorial boundaries.

In the developed strategy, foraging behavior of FFSS based animals (e.g. spider monkeys) is divided into four steps. First, the group starts food foraging and evaluates their distance from the food. In the second step, based on the distance from the foods, group members update their positions and again evaluate distance from the food sources. Furthermore, in the third step, local leader updates its best position within the group and if the position is not updated for a specified number of times then all members of that group start searching of the foods in different directions. Next, in the fourth step, global leader, updates its ever best position and in case of stagnation, it splits the group into smaller size subgroups. All the four steps mentioned aforesaid, are continuously executed until the desired output is achieved. There are two important control parameters necessary to introduce in the proposed strategy, one is '*GlobalLeaderLimit*' and another is '*LocalLeaderLimit*' which helps local and global leaders to take appropriate decisions.

The control parameter *LocalLeaderLimit* is used to avoid stagnation i.e. if a local group leader does not update herself in a specified number of times then that group is re-directed to a different direction for foraging. Here, the term 'specified number of times' is referred as *LocalLeaderLimit*. Another control parameter, *GlobalLeaderLimit* is used for the same purpose for global leader. The global leader breaks the group into smaller sub-groups if she does not update in a specified number of times.

The proposed strategy follows self-organization and division of labour properties for obtaining intelligent swarming behaviors of animals. As animals updating their positions by learning from local leader, global leader and self experience in first and second steps of algorithm, it shows positive feedback mechanisms of self organization. In third step, the stagnated group members are re-directed to different directions for food searching, shows fluctuations property. In fourth step, when the global leader is get stuck, it divides the groups into smaller subgroups for foraging of foods this phenomena presents division of labour property. 'Local leader limit' and 'Global leader limit' provides negative feedback to help local and global leader's for their decisions.

3.1 Main steps of Spider Monkey Optimization Algorithm (SMO)

Similar to the other population-based algorithms, SMO is a trial and error based collaborative iterative process. SMO process 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. The position update process in Global Leader phase is inspired from the Gbest-guided ABC [32] and modified version of ABC [11]. The details of each step of *SMO* implementation are explained below :

3.1.1 Initialization of the Population

Initially, *SMO* generates a uniformly distributed initial population of N spider monkeys where each monkey $SM_i (i = 1, 2, \dots, N)$ is a D -dimensional vector. Here D is the number of variables in the optimization problem and SM_i represent

the i^{th} Spider Monkey (SM) in the population. Each spider monkey SM corresponds to the potential solution of the problem under consider. Each SM_i is initialized as follows:

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

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

3.1.2 Local Leader Phase (LLP)

In Local Leader phase, SM modify its current position based on the information of the local leader experience as well as local group members experience. The fitness value of so obtained new position is calculated. If the fitness value of the new position is higher than that of the old position, then the SM updates his 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

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

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 k^{th} SM which is chosen randomly within k^{th} group such that $r \neq i$, $U(0,1)$ is a universally distributed random number between 0 and 1. Algorithm 1 shows position update process in the Local Leader phase. In the algorithm 1, pr is the perturbation rate which controls the amount of perturbation in the current

Algorithm 1 Position update process in Local Leader Phase:

```

for each member  $SM_i \in k^{th}$  group do
  for each  $j \in \{1, \dots, D\}$  do
    if  $U(0,1) \geq pr$  then
       $SM_{newij} = SM_{ij} + U(0,1) \times (LL_{kj} - SM_{ij}) + U(-1,1) \times (SM_{rj} - SM_{ij})$ 
    else
       $SM_{newij} = SM_{ij}$ 
    end if
  end for
end for

```

position. The range of pr is $[0.1, 0.8]$ (explained in section 4.3).

3.1.3 Global Leader Phase (GLP)

After completion of the Local Leader phase, the Global Leader phase (GLP) starts. In GLP phase, all the SM 's update their position using experience of Global Leader and local group member's experience. The position update equation for this phase is as follows:

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

where GL_j represents the j^{th} dimension of the global leader position and $j \in \{1, 2, \dots, D\}$ is the randomly chosen index.

In this phase, the position of SM_i is updated based on a probability $prob_i$ which is calculated using their fitness in this way The better candidate will have more chance to make itself better. The probability $prob_i$ may be calculated using following expression (there may be some other but must be a function of fitness):

$$prob_i = \frac{fitness_i}{\sum_{i=1}^N fitness_i} \quad (4)$$

where $fitness_i$ is the fitness value of the i^{th} SM .

Further, the fitness of the newly generated position of the SM 's is calculated and compared with the old one and adopted the better one.

Algorithm 2 Position update process in Global Leader Phase (GLP) :

```

count = 0;
while count < group size do
  for each member  $SM_i \in$  group do
    if  $U(0, 1) < prob_i$  then
      count = count + 1.
      Randomly select  $j \in \{1 \dots D\}$ .
      Randomly select  $SM_r \in$  group s.t.  $r \neq i$ .
       $SM_{new_{ij}} = SM_{ij} + U(0, 1) \times (GL_j - SM_{ij}) + U(-1, 1) \times (SM_{rj} - SM_{ij})$ .
    end if
  end for
end while

```

3.1.4 Global Leader Learning (GLL) phase

In this phase, the position of the global leader is updated by applying the greedy selection in the population i.e the position of the SM having best fitness in the population is selected as the updated position of the global leader. Further, it is checked that the position of global leader is updating or not and if not then the *GlobalLimitCount* is incremented by 1.

3.1.5 Local Leader Learning (LLL) phase

In this phase, the position of the local leader is updated by applying the greedy selection in that group i.e. the position of the SM having best fitness in that group is selected as the updated position of the local leader. Next, the updated position of the local leader is compared with the old one and if the local leader is not updated the *LocalLimitCount* is incremented by 1.

3.1.6 Local Leader Decision (LLD) phase

If any Local Leader position is not updated up to a predetermined threshold called *LocalLeaderLimit*, then all the members of that group update their positions either by random initialization or by using combined information from Global Leader and Local Leader through equation (5), based on the *pr*.

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

It is clear from the equation (5) that the updated dimension of this SM is attracted towards global leader and repel from the local leader. The position update process of LLD phase is shown in algorithm 3 Further, the fitness of updated

Algorithm 3 Local Leader Decision Phase:

```

if LocalLimitCount > LocalLeaderLimit then
  LocalLimitCount = 0.
  for each  $j \in \{1 \dots D\}$  do
    if  $U(0, 1) \geq pr$  then
       $SM_{new_{ij}} = SM_{min_j} + U(0, 1) \times (SM_{max_j} - SM_{min_j})$ 
    else
       $SM_{new_{ij}} = SM_{ij} + U(0, 1) \times (GL_j - SM_{ij}) + U(0, 1) \times (SM_{ij} - LL_{kj})$ 
    end if
  end for
end if

```

SM is calculated.

3.1.7 Global Leader Decision (GLD) phase

In this phase, the position of global leader is monitored and if it is not updated up to predetermined number of iterations called *GlobalLeaderLimit*, then the global leader divides the population into smaller 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 as shown in the Fig. 3-6. Each time in GLD phase, LLL process is initiated to elect the local leader in the newly formed

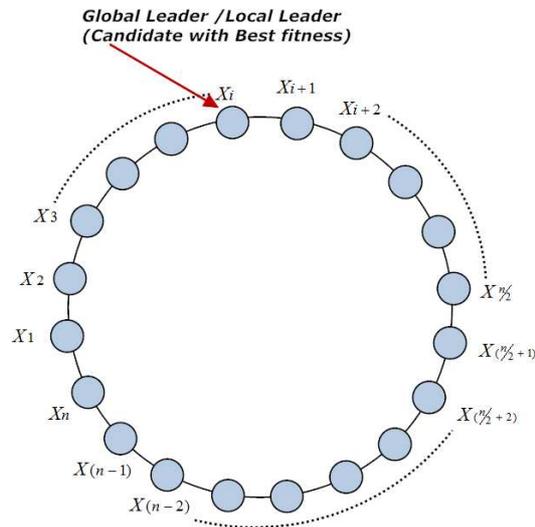


Fig. 3: SMO topology: single group

groups. The case in which maximum number of groups are formed and even then the position of global leader is not updated then the global leader combines all the groups to form a single group. Thus the proposed algorithm mimics fusion-fission structure of SMs. The working of this phase is shown in algorithm 4

Algorithm 4 Global Leader Decision Phase:

```

if  $GlobalLimitCount > GlobalLeaderLimit$  then
   $GlobalLimitCount = 0$ 
  if Number of groups  $< MG$  then
    Divide the population into groups.
  else
    Combine all the groups to make a single group.
  end if
  Update Local Leaders position.
end if

```

The complete pseudo-code of the proposed strategy is given in algorithm 5:

3.2 Control Parameters in SMO

It is clear from the above discussion that there are four control parameters in *SMO* algorithm: the value of *LocalLeaderLimit*, *GlobalLeaderLimit*, the maximum group *MG* and perturbation rate *pr*. Some settings of control parameters are suggested as follows:

- $MG = N/10$, i.e. it is chose such that minimum number of SM's in a group should be 10

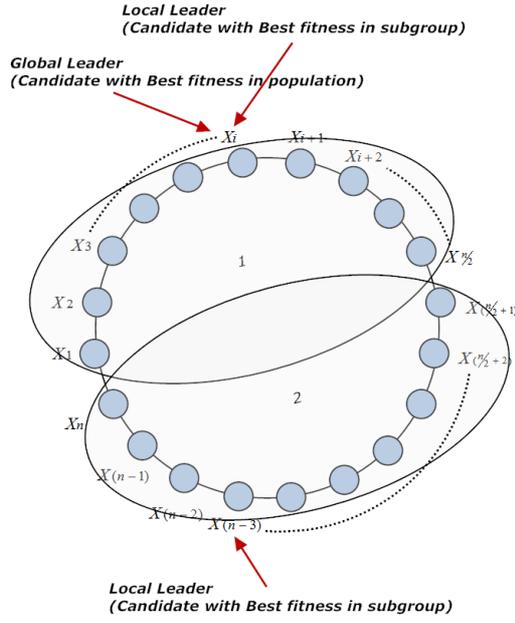


Fig. 4: SMO topology: swarm is divided into two group

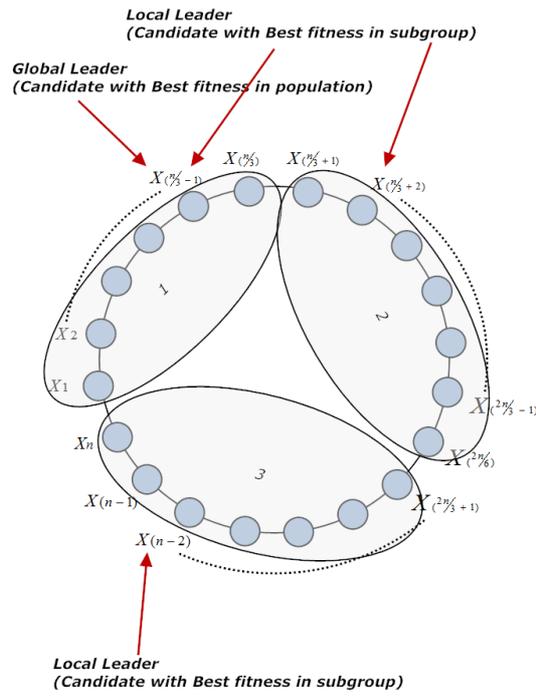


Fig. 5: SMO topology: swarm is divided into three group

- *GlobalLeaderLimit* should be $\in [N/2, 2 \times N]$,
- *LocalLeaderLimit* should be $D \times N$,

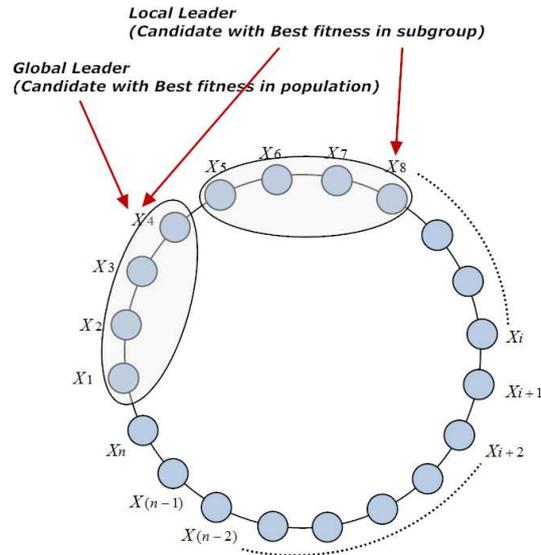


Fig. 6: SMO topology: minimum size group

Algorithm 5 Spider Monkey Optimization (SMO) Algorithm:

1. Initialize Population, *LocalLeaderLimit*, *GlobalLeaderLimit*, *pr*.
 2. Calculate fitness (i.e. the distance of individuals from food sources).
 3. Select global leader and local leaders by applying greedy selection (see section 3.1.4 3.1.5,).
- while** (Termination criteria is not satisfied) **do**
- (i) For finding the objective (Food Source), generate the new positions for all the group members by using self experience, local leader experience and group members experience using algorithm (1).
 - (ii) Apply the greedy selection process for all the group members based on their fitness;
 - (iii) Calculate the probability $prob_i$ for all the group members using equation (4).
 - (iv) Produce new positions for the all the group members, selected by $prob_i$, by using self experience, global leader experience and group members experiences using algorithm (2).
 - (v) Update the position of local and global leaders, by applying the greedy selection process on all the groups (see section 3.1.4, 3.1.5).
 - (vi) If any Local group leader is not updating her position after a specified number of times (*LocalLeaderLimit*) then re-direct all members of that particular group for foraging by algorithm (3)
 - (vii) If Global Leader is not updating her position for a specified number of times (*GlobalLeaderLimit*) then she divides the group into smaller groups by algorithm (4), but minimum size of each group should be 4.
- end while**

$$- pr \in [0.1, 0.8],$$

here, N is the swarm size.

4 Experimental Results and Discussion

In order to establish *SMO* for optimization as a swarm intelligence algorithm, it is tested over well known optimization test problems as well as some popular real world optimization problems. Sensitivity analysis of different parameters, statistical analysis of results with comparison to some other well established optimization algorithms have been carried out.

4.1 Test Problems

In order to analyze the performance of *SMO* algorithm, 21 different global optimization problems (f_1 to f_{21}) are selected (listed in Table 3). These are continuous, non-biased optimization problems and have different degrees of complexity and multimodality. Test problems $f_1 - f_8$ and $f_{15} - f_{21}$ are taken from [1] and test problems $f_9 - f_{14}$ are taken from [26] with the associated offset values.

4.2 Real world problems

To see the robustness of the proposed strategy, 4 real world global optimization problems namely Pressure Vessel [30], Parameter Estimation for Frequency-Modulated (FM) Sound Waves [5], Compression Spring [15, 22, 3] and Gear Train [15, 22] have been solved.

Pressure Vessel design (without Granularity) : The pressure vessel design is to minimize the total cost of the material, forming, and welding of a cylindrical vessel [30]. There are four design variables involved: x_1 , (T_s , shell thickness), x_2 (T_h , spherical head thickness), x_3 (R , radius of cylindrical shell), and x_4 (L , shell length). The mathematical formulation of this typical constrained optimization problem is as follows:

$$f_{22}(\mathbf{X}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1611x_1^2x_4 + 19.84x_1^2x_3$$

subject to

$$\begin{aligned} g_1(\mathbf{X}) &= 0.0193x_3 - x_1 \\ g_2(\mathbf{X}) &= 0.00954x_3 - x_2 \\ g_3(\mathbf{X}) &= 750 \times 1728 - \pi x_3^2(x_4 + \frac{4}{3}x_3) \end{aligned}$$

The search boundaries for the variables are

$$\begin{aligned} 1.125 &\leq x_1 \leq 12.5, \\ 0.625 &\leq x_2 \leq 12.5, \\ 1.0 \times 10^{-8} &\leq x_3 \leq 240 \end{aligned}$$

and

$$1.0 \times 10^{-8} \leq x_4 \leq 240.$$

The best known global optimum solution is $f(1.125, 0.625, 58.29016, 43.69266) = 7197.729$ [30]. For a successful run, the minimum error criteria is fixed to be 1.0×10^{-5} i.e. an algorithm is considered successful if it finds the error less than acceptable error in a specified maximum function evaluations.

Frequency-Modulated (FM) sound wave : Frequency-Modulated (FM) sound wave synthesis has an important role in several modern music systems and to optimize the parameter of an FM synthesizer is a six dimensional optimization problem where the vector to be optimized is $X = \{a_1, w_1, a_2, w_2, a_3, w_3\}$ of the sound wave given in equation (6). The problem is to generate a sound (6) similar to target (7). This problem is a highly complex multimodal and have strong epistasis, with minimum value $f(\mathbf{X}_{\text{sol}}) = 0$. The expressions for the estimated sound and the target sound waves are given as:

$$y(t) = a_1 \sin(w_1 t \theta) + a_2 \sin(w_2 t \theta) + a_3 \sin(w_3 t \theta) \quad (6)$$

$$y_0(t) = (1.0) \sin((5.0)t\theta) - (1.5) \sin((4.8)t\theta) + (2.0) \sin((4.9)t\theta) \quad (7)$$

respectively where $\theta = 2\pi/100$ and the parameters are defined in the range $[-6.4, 6.35]$. The fitness function is the summation of squared errors between the estimated wave (6) and the target wave (7) and is given below:

$$f_{23}(\mathbf{X}) = \sum_{i=0}^{100} (y(t) - y_0(t))^2$$

Acceptable error for this problem is fixed to be 1.0×10^{-05} , i.e. an algorithm is considered successful if it finds the error less than acceptable error in a specified maximum function evaluations.

Compression Spring: The considered third real world problem is compression spring problem [15, 22, 3]. This problem minimizes the weight of a compression spring, subject to constraints of minimum deflection, shear stress, surge frequency, and limits on outside diameter and on design variables. There are three design variables: the wire diameter x_1 , the mean coil diameter x_2 , and the number of active coils x_3 . This is a simplified version of a more difficult problem. The mathematical formulation of this problem is:

$$\begin{aligned} x_1 &\in \{1, \dots, 70\} \text{ granularity } 1 \\ x_2 &\in [0.6; 3] \\ x_3 &\in [0.207; 0.5] \text{ granularity } 0.001 \end{aligned}$$

and four constraints

$$\begin{aligned} g_1 &:= \frac{8C_f F_{max} x_2}{\pi x_3^3} - S \leq 0 \\ g_2 &:= l_f - l_{max} \leq 0 \\ g_3 &:= \sigma_p - \sigma_{pm} \leq 0 \\ g_4 &:= \sigma_w - \frac{F_{max} x_2 - F_p}{K} \leq 0 \end{aligned}$$

with

$$\begin{aligned} C_f &= 1 + 0.75 \frac{x_3}{x_2 - x_3} + 0.615 \frac{x_3}{x_2} \\ F_{max} &= 1000 \\ S &= 189000 \\ l_f &= \frac{F_{max}}{K} + 1.05(x_1 + 2)x_3 \\ l_{max} &= 14 \\ \sigma_p &= \frac{F_p}{K} \\ \sigma_{pm} &= 6 \\ F_p &= 300 \\ K &= 11.5 \times 10^6 \frac{x_3^4}{8x_1 x_2^2} \\ \sigma_w &= 1.25 \end{aligned}$$

and the function to be minimized is

$$f_{24}(\mathbf{X}) = \pi^2 \frac{x_2 x_3^2 (x_1 + 2)}{4}$$

The best known solution is $(7, 1.386599591, 0.292)$, which gives the fitness value $f^* = 2.6254$. Here, minimum error is fixed to be 10^{-4} , i.e. a run is said to be successful if it finds a fitness f so that $|f - f^*| \leq 10^{-4}$ in the maximum number of function evaluations.

Compound Gear Train Arrangement problem: The fourth problem is to optimize the gear ration for the compound gear train arrangement. For more details, see [15, 22]. The optimization problem is to find the number of teeth for gearwheels in order to produce a gear ratio as close as possible to the target ration ($\frac{1}{6.931}$). For each gear, the minimum number of teeth is 12 and the maximum is 60. The problem to be minimized is

$$f_{25}(\mathbf{X}) = \left(\frac{1}{6.931} - \frac{x_1 x_2}{x_3 x_4} \right)^2$$

The search space is $\{12, 13, \dots, 60\}^4$. There are several solutions, depending on the required precision. Here, we used 10^{-13} . So, a possible solution is $f^* = f(19, 16, 43, 49) = 2.7 \times 10^{-12}$. For this problem, a run is said to be successful if it finds a fitness f so that $|f - f^*| \leq 10^{-13}$.

4.3 Experimental setting

Swarm size, perturbation rate (pr), LLL , GLL and maximum number of groups (MG) are the parameters that affects the performance of the SMO significantly. To fine tune (finding most suitable values) these parameters, sensitivity analysis with different values of these parameters has been carried out. Swarm size is varied from 40 to 160 with step size 20, pr is varied from 0.1 to 0.9 with step size of 0.1, MG is varied from 1 to 6 with step size 1, LLL is varied from 100 to 2500 with step size 200 and GLL is varied from 10 to 220 with step size 30. At a time only one parameter is varied while all other parameters are kept fixed. This fine tuning is done with the following assumptions:

- pr is varied from 0.1 to 0.9 while MG , LLL , GLL and Swarm size are fixed to be 5, 1500, 50 and 50 respectively.
- MG is varied from 1 to 6 while LLL , GLL and Swarm size are fixed to be 1500, 50 and 50 respectively. pr is linearly increasing from 0.1 to 0.4 by iterations.
- GLL is varied from 10 to 220 while LLL , MG and Swarm size are fixed to be 1500, 5 and 50 respectively. pr is linearly increasing from 0.1 to 0.4 by iterations.
- LLL is varied from 100 to 2500 while MG , GLL and Swarm size are fixed to be 5, 50 and 50 respectively. pr is linearly increasing from 0.1 to 0.4 by iterations.
- Swarm size is varied from 40 to 160 while LLL , GLL and MG are fixed to be 1500, 50 and 5 respectively. pr is linearly increasing from 0.1 to 0.4 by iterations.

For the purpose of sensitivity analysis, 6 problems are considered and each problem is simulated 30 times. Effects of these parameters are shown in Figures 7(a)- 7(f) respectively. It is clear from Fig. 7(a) that the test problems are very sensitive towards pr . Some problems perform better at lower value of pr while other perform better at higher value of pr . Therefore, the value of pr is adopted linearly increasing over iterations to balance all the test problems. Further, by analyzing the 7(b), it can be stated that the value of $MG = 5$ gives comparatively better results for the given set of test problems. Sensitivity of *GlobalLeaderLimit*(GLL) and *LocalLeaderLimit*(LLL) can be analyzed by Fig. 7(c) and Fig. 7(d) and found that the value of $GLL = 50$ and $LLL = 1500$ gives better results on the considered benchmark optimization problems. Further, Swarm size is analyzed in Fig. 7(e) and Fig. 7(f). It is clear from Fig. 7(e) that the functions f_9 , f_{13} and f_{21} are sensitive towards swarm size and give better results at 40. Further, success rate of remaining three benchmarks functions are not significant varies by varying swarm size therefore average of functions evaluations are analyzed. Fig. 7(f) shows that for swarm size 40, average number of function evaluations are minimum for all the considered benchmark functions.

To prove the efficiency of SMO algorithm, it is compared with three popular nature inspired algorithms namely PSO (based on Standard PSO 2006 [3] but with linearly decreasing inertia weight and a different parameters setting), ABC [10] and DE (*DE/rand/bin/1*) [25]. These algorithms are selected due to their similarities in the position update procedure as it is based on difference vectors i.e. variation component [32]. For the comparison, same stopping criteria, number of simulations, maximum number of function evaluations, and random number generator (KISS [21]) are used for all the algorithms (*ABC*, *DE*, *PSO*) as in *SMO* algorithm. The values of parameters for the considered algorithms are as follows:

SMO parameters setting:

- The Swarm size $N = 50$,
- $MG = 5$,
- *GlobalLeaderLimit*=50,
- *LocalLeaderLimit*=1500,
- $pr \in [0.1, 0.4]$, linearly increasing over iterations,

$$pr_{G+1} = pr_G + (0.4 - 0.1)/MIR \quad (8)$$

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 2.0×10^5) is reached or the corresponding acceptable error (mentioned in Table 3) have been achieved,
- The number of simulations/run =100.

ABC parameters setting:

- Colony size $SN=100$,
- Number of food sources $SN/2$,
- *limit* = 1500 [10],

DE parameters setting:

- The crossover probability $CR=0.9$ [8],
- The scale factor which controls the implication of the differential variation $F=0.5$; [18],
- Population size $NP=50$,

PSO parameters setting:

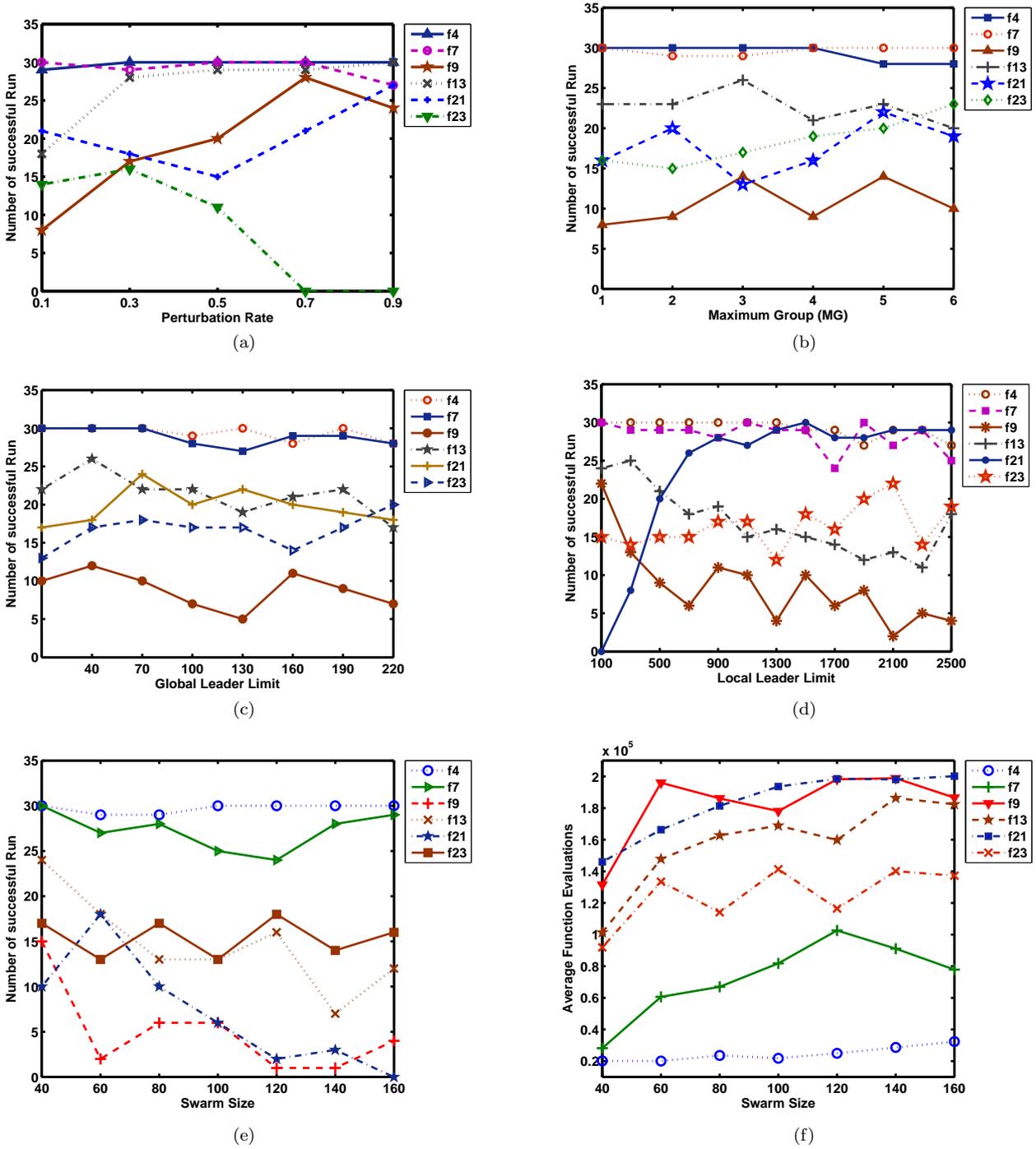


Fig. 7: Effect of parameters on success rate for functions $f_4, f_7, f_9, f_{13}, f_{21}$ and f_{23} (a) for pr (b) for MG (c) for $GlobalLimit$ (d) for $LocalLimit$ (e) for $Swarm Size$ v/s Successful Run (f) for $Swarm Size$ v/s Average Function Evaluations.

- Inertia Weight (w), decreases linearly from 1 to 0.1,
- Acceleration coefficients ($c_1 = 2, c_2 = 2$),

– Swarm size $S=50$,

4.4 Comparison of *SMO* with *PSO*, *DE* and *ABC*

Numerical results with experimental setting of subsection 4.3 are given in Table 1. In Table 1, success rate (SR), mean error (*ME*), average function evaluations (*AFE*) and standard deviation (*SD*) are reported. Table 1 shows that most of the time *SMO* outperforms in terms of reliability, efficiency and accuracy. Some more intensive statistical analyses based on *t* test and boxplots have been carried out for results of *SMO*, *PSO*, *DE* and *ABC*.

Table 1: Comparison of the results of test problems

Test Function	Algorithm	ME	SD	AFE	SR
f_1	SMO	5.24E-06	3.57E-06	58578.22	100
	PSO	3.51E-01	3.10E-01	197777.5	2
	DE	5.21E-02	4.97E-02	177211.5	17
	ABC	4.51E-06	3.61E-06	44076	100
f_2	SMO	1.00E-06	0.00E+00	16897.82	100
	PSO	1.00E-06	0.00E+00	38102	100
	DE	3.00E-01	5.24E-01	38547.5	88
	ABC	1.00E-06	0.00E+00	20047	100
f_3	SMO	8.77E-06	1.31E-06	16648.83	100
	PSO	2.09E-03	1.45E-02	36645.5	98
	DE	8.55E-06	1.24E-06	20572.5	100
	ABC	7.53E-06	2.24E-06	37194	100
f_4	SMO	2.29E-04	1.54E-03	17539.9	98
	PSO	7.79E-04	2.81E-03	48651	93
	DE	5.57E-04	2.39E-03	30088.5	95
	ABC	7.83E-06	2.19E-06	41435	100
f_5	SMO	8.88E-06	9.53E-07	15323.22	100
	PSO	9.33E-06	5.73E-07	44406.5	100
	DE	8.89E-06	1.37E-06	27901.5	100
	ABC	8.48E-06	1.81E-06	44961	100
f_6	SMO	4.81E-06	2.58E-06	1569.15	100
	PSO	4.22E-06	2.67E-06	2762	100
	DE	4.72E-06	1.13E-06	1849	100
	ABC	7.81E-06	2.42E-06	31948.76	100
f_7	SMO	1.21E-04	1.56E-04	44133.41	96
	PSO	8.51E-06	1.83E-06	48767	100
	DE	4.20E-04	2.02E-03	70633	67
	ABC	1.50E-04	7.87E-05	2.0E+05	0

to be cont'd on next page

Table 1: Comparison of the results of test problems (Cont.)

Test Function	Algorithm	ME	SD	AFE	SR
f_8	SMO	6.82E-05	2.43E-05	14499.67	100
	PSO	1.90E-01	3.92E-01	54523	80
	DE	6.01E-02	2.37E-01	15526	94
	ABC	6.65E-05	2.16E-05	7432.04	100
f_9	SMO	1.30E+00	2.91E+00	156483.74	47
	PSO	3.28E+00	1.33E+01	186616	59
	DE	2.42E+00	1.68E+00	192740	4
	ABC	7.68E-01	1.52E+00	167743.63	34
f_{10}	SMO	7.65E-06	1.86E-06	5898.42	100
	PSO	8.07E-06	1.47E-06	15854.5	100
	DE	7.46E-06	2.03E-06	10805.5	100
	ABC	7.35E-06	2.16E-06	17112	100
f_{11}	SMO	1.04E+02	1.27E+01	2.0E+05	0
	PSO	3.69E+01	4.54E+00	2.0E+05	0
	DE	2.77E+02	1.26E+01	2.0E+05	0
	ABC	9.37E+01	1.27E+01	2.0E+05	0
f_{12}	SMO	1.84E+04	5.18E+03	2.0E+05	0
	PSO	7.84E+02	1.01E+03	2.0E+05	0
	DE	1.21E+05	3.39E+03	2.0E+05	0
	ABC	1.31E+04	3.96E+03	2.0E+05	0
f_{13}	SMO	2.68E-03	6.03E-03	130922.94	77
	PSO	4.28E-02	2.77E-02	198768.5	2
	DE	1.34E-02	1.20E-02	165684	22
	ABC	1.09E-03	2.86E-03	71707.2	86
f_{14}	SMO	8.66E-06	1.13E-06	9069.39	100
	PSO	9.10E-06	8.69E-07	24687.5	100
	DE	8.85E-06	1.21E-06	15959	100
	ABC	8.09E-06	1.71E-06	32415	100

to be cont'd on next page

Table 1: Comparison of the results of test problems (Cont.)

Test Function	Algorithm	ME	SD	AFE	SR
f_{15}	SMO	4.20E-15	4.07E-15	3401.64	100
	PSO	5.05E-15	2.94E-15	9584.5	100
	DE	5.02E-15	2.37E-15	4198	100
	ABC	5.48E-13	4.59E-12	26797.82	98
f_{16}	SMO	4.71E-14	2.69E-14	11789.91	100
	PSO	5.13E-14	2.87E-14	9778	100
	DE	4.50E-14	7.36E-15	5210	100
	ABC	5.71E-08	5.64E-07	98925.18	92
f_{17}	SMO	4.89E-01	4.98E-03	1258.29	100
	PSO	4.91E-01	5.64E-03	4979	100
	DE	4.90E-01	1.12E-03	2725.5	100
	ABC	4.90E-01	5.25E-03	2567	100
f_{18}	SMO	8.73E-05	6.74E-06	743.49	100
	PSO	8.81E-05	6.92E-06	1424.5	100
	DE	9.00E-05	1.53E-06	1383.5	100
	ABC	8.85E-05	6.92E-06	2079	100
f_{19}	SMO	1.95E-03	2.72E-06	2046.03	100
	PSO	1.95E-03	2.79E-06	3206	100
	DE	1.95E-03	1.31E-05	8365.5	97
	ABC	1.95E-03	3.30E-06	30157.84	99
f_{20}	SMO	4.97E-06	5.58E-06	4379.76	100
	PSO	1.01E-04	4.01E-04	72252.5	83
	DE	4.78E-06	2.16E-06	9663	100
	ABC	5.32E-06	5.95E-06	8248.56	100
f_{21}	SMO	1.18E-02	6.67E-03	163608.21	55
	PSO	4.15E-01	3.02E-01	175323.5	24
	DE	3.29E+00	2.54E-01	198907	2
	ABC	7.84E-03	2.02E-03	74296.79	100

to be cont'd on next page

Table 1: Comparison of the results of test problems (Cont.)

Test Function	Algorithm	ME	SD	AFE	SR
f_{22}	SMO	2.85E-05	3.48E-05	95307.61	65
	PSO	2.91E-05	3.17E-05	90958.5	65
	DE	5.75E-06	1.58E-05	13801	99
	ABC	1.16E+01	8.33E+00	2.0E+05	0
f_{23}	SMO	1.70E+00	3.91E+00	88108.53	84
	PSO	2.03E+00	4.13E+00	100716.5	79
	DE	5.29E+00	6.40E+00	100382.5	56
	ABC	6.90E+00	5.61E+00	2.0E+05	0
f_{24}	SMO	2.41E-07	2.39E-06	78258.52	99
	PSO	4.08E-11	2.57E-11	36221.5	100
	DE	4.10E+03	1.03E-03	126115	59
	ABC	1.62E+00	1.20E-02	2.0E+05	0
f_{25}	SMO	6.46E-11	2.82E-10	149701.52	49
	PSO	1.76E-11	8.79E-11	131443.5	56
	DE	1.40E-04	1.50E-10	138718.5	33
	ABC	1.14E-08	2.29E-10	173842.68	25

Fig. 8 show the convergence characteristics in terms of the error of the median run of each algorithm for functions on which all the considered algorithms achieved 100% success rate within the specified maximum function evaluations (to carry out fair comparison of convergence rate). It can be observed that the convergence of *SMO* is relatively better than the *PSO*, *DE* and *ABC*.

4.4.1 Statistical Analysis

In order to compare the performance of the *SMO*, *PSO*, *DE* and *ABC*, statistical analyses have been carried out using t-test and boxplots.

The *t*-test is quite popular among researchers in the field of evolutionary computation. In this paper students *t*-test is applied according to the description given in [4] for a confidence level of 0.95. Table 2 shows the results of the *t*-test for the null hypothesis that there is no difference in the mean number of function evaluations of 100 runs using *SMO*, *PSO*, *DE* and *ABC*. Note that here '+' indicates the significant difference (or the null hypothesis is rejected) at a 0.05 level of significance, '-' implies that there is no significant difference while '=' indicates that comparison is not possible. In Table 2. *SMO* is compared with the *PSO*, *DE* and *ABC*. Significant differences are observed in 55 comparisons out of 75 comparisons. Therefore, it can be concluded that the results of *SMO* is significantly better from the *ABC*, *DE* and *PSO*.

Boxplot analysis is also carried out for all the considered algorithms. The empirical distribution of data is efficiently represented graphically by the boxplot analysis tool [31]. The Boxplots for *SMO*, *PSO*, *DE* and *ABC* are shown in Figure 9. It is clear from this figure that strategy (1), i.e. *SMO* is best among all mentioned strategies as interquartile range and median are low for *SMO*.

SMO, *PSO*, *DE* and *ABC* are compared through *SR*, *ME* and *AFE*. First *SR* is compared for all these algorithms and if it is not possible to distinguish the algorithms based on *SR* then comparison is made on the basis of *ME*. *AFE* is used for comparison if it is possible on the basis of *SR* and *ME* both. It is observed from Table 1 that *SMO* outperforms for 13 problems among all the considered algorithms. It is a better algorithm for 17 problems, 21 problems and 19 problems compared to *PSO*, *DE* and *ABC* respectively.

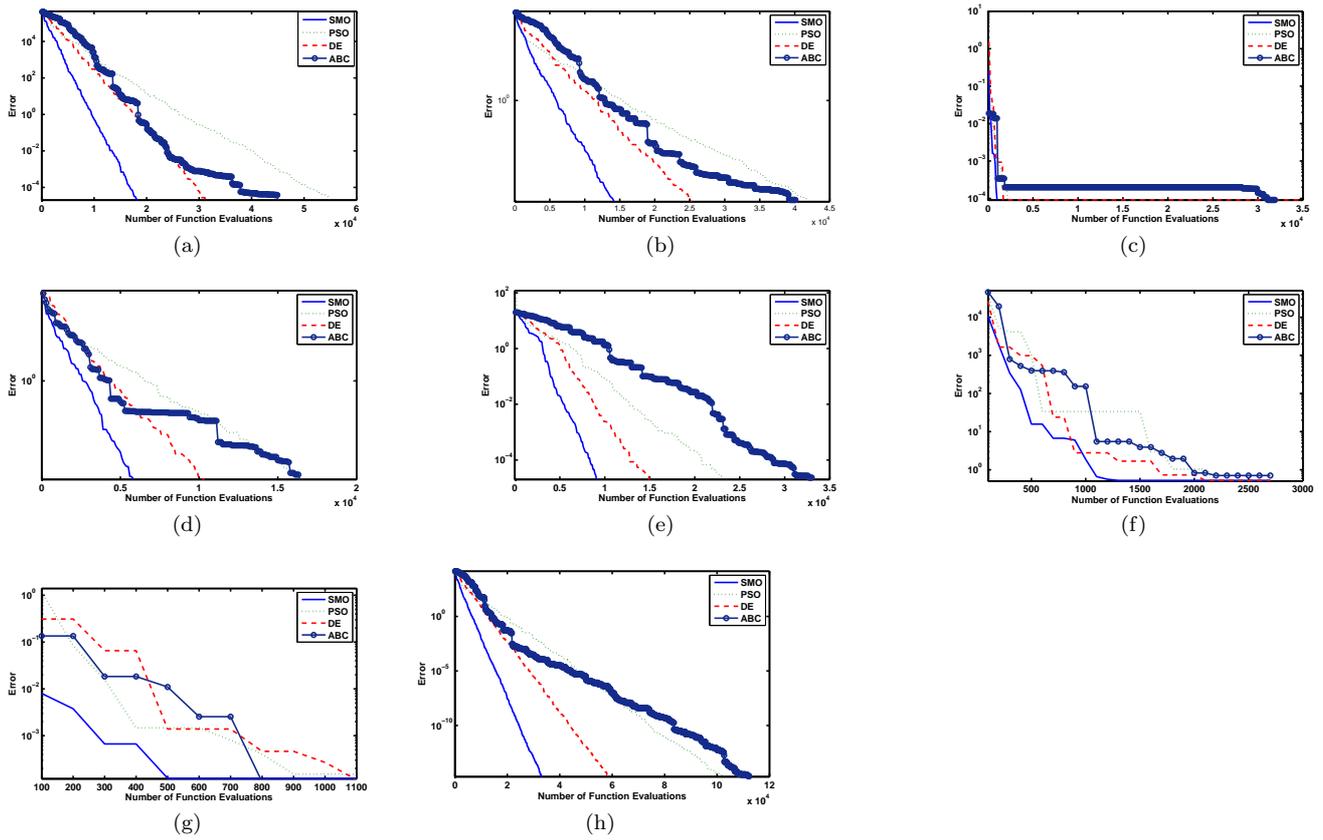


Fig. 8: Convergence characteristics of *SMO*, *PSO*, *DE* and *ABC* for functions; (a) (f_{23}), (b) (f_{24}), (c) (f_{29}), (d) (f_{33}), (e) (f_{36}), (f) (f_{37}), (g) (f_{41}).

Table 2: Results of the Students t test

Test Problem	Students t test with <i>SMO</i>			Test Problem	Students t test with <i>SMO</i>		
	PSO	DE	ABC		PSO	DE	ABC
f_1	+	+	-	f_{14}	+	+	+
f_2	+	+	+	f_{15}	+	+	+
f_3	+	+	+	f_{16}	-	-	+
f_4	+	+	+	f_{17}	+	+	+
f_5	+	+	+	f_{18}	+	+	+
f_6	+	+	+	f_{19}	+	+	+
f_7	-	+	+	f_{20}	+	+	+
f_8	+	-	-	f_{21}	+	+	-
f_9	+	+	-	f_{22}	-	-	+
f_{10}	+	+	+	f_{23}	+	+	+
f_{11}	=	=	=	f_{24}	-	+	+
f_{12}	=	=	=	f_{25}	-	-	+
f_{13}	+	+	-				

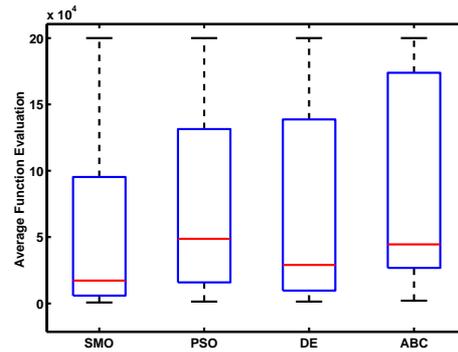


Fig. 9: Boxplot graph for Average Function Evaluation: (1) *SMO* (2) *PSO* (3) *DE* (4) *ABC*

5 Conclusion

In this paper, a new meta-heuristic algorithm for optimization is proposed. The inspiration is from the social behavior of spider monkeys. The proposed algorithm proves to be very flexible in the category of swarm intelligence based algorithms. Generally, in the solution search process, exploration and exploitation capabilities contradict each other. Therefore to obtain better performance on the problems of optimization, the two capabilities should be well balanced. It may be noted that the solution search equation of the *SMO* in *LocalLeaderPhase*, *GlobalLeaderPhase* and *LocalLeaderLearning phase* consists of the local best solution, the global best solution and both respectively. Therefore, *SMO* exploits the search space efficiently. Further, use of perturbation rate (pr) and random difference vectors in the said phases, explore the search space. With the help of experiments over test problems and real world problems, it has been shown that, for most of the problems the reliability (due to success rate), efficiency (due to average number of function evaluations) and accuracy (due to mean objective function value) of *SMO* algorithm is higher than that of *ABC*, *PSO* and *DE*. Hence, it may be concluded that *SMO* is going to be a competing candidate in the field of swarm based optimization algorithms.

References

1. M.M. Ali, C. Khompatraporn, and Z.B. Zabinsky. A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. *Journal of Global Optimization*, 31(4):635–672, 2005.
2. E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm intelligence: from natural to artificial systems*. Number 1. Oxford University Press, USA, 1999.
3. M. Clerc. *Particle swarm optimization*. Wiley-ISTE, 2006.
4. C. Croarkin and P. Tobias. Nist/sematech e-handbook of statistical methods. Retrieved March, 1:2010, 2010.
5. S. Das and PN Suganthan. Problem definitions and evaluation criteria for cec 2011 competition on testing evolutionary algorithms on real world optimization problems. *Jadavpur University, Kolkata, India, and Nanyang Technological University, Singapore, Tech. Rep*, 2010.
6. L.N. De Castro and F.J. Von Zuben. Artificial immune systems: Part i–basic theory and applications. *Universidade Estadual de Campinas, Dezembro de, Tech. Rep*, 1999.
7. M. Dorigo and T. Stützle. *Ant colony optimization*. the MIT Press, 2004.
8. R. Gamperle, S.D. Muller, and A. Koumoutsakos. A parameter study for differential evolution. *Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*, 10:293–298, 2002.
9. RL Jeanne. The evolution of the organization of work in social insects. *Monitore zoologico italiano*, 20(2):119–133, 1986.
10. D. Karaboga. An idea based on honey bee swarm for numerical optimization. *Techn. Rep. TR06, Erciyes Univ. Press, Erciyes*, 2005.

11. D. Karaboga and B. Akay. A modified artificial bee colony (abc) algorithm for constrained optimization problems. *Applied Soft Computing*, 2010.
12. J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948. IEEE, 1995.
13. K. Milton. Diet and social organization of a free-ranging spider monkey population: The development of species-typical behavior in the absence of adults. *Juvenile primates: life history, development, and behavior*, pages 173–181, 1993.
14. M.A. Norconk and W.G. Kinzey. Challenge of neotropical frugivory: travel patterns of spider monkeys and bearded sakis. *American Journal of Primatology*, 34(2):171–183, 1994.
15. G.C. Onwubolu and BV Babu. *New optimization techniques in engineering*, volume 141. Springer Verlag, 2004.
16. G.F. Oster and E.O. Wilson. *Caste and ecology in the social insects*. Princeton Univ Pr, 1979.
17. K.M. Passino. Bacterial foraging optimization. *International Journal of Swarm Intelligence Research (IJSIR)*, 1(1):1–16, 2010.
18. K.V. Price. Differential evolution: a fast and simple numerical optimizer. In *Fuzzy Information Processing Society, 1996. NAFIPS. 1996 Biennial Conference of the North American*, pages 524–527. IEEE, 1996.
19. K.V. Price, R.M. Storn, and J.A. Lampinen. *Differential evolution: a practical approach to global optimization*. Springer Verlag, 2005.
20. G. Ramos-Fernandez. Patterns of association, feeding competition and vocal communication in spider monkeys, *ateles geoffroyi*. 2001.
21. G. Rose. Kiss: A bit too simple. <http://eprint.iacr.org/2011/007.pdf>, 18 April 2011.
22. E. Sandgren. Nonlinear integer and discrete programming in mechanical design optimization. *Journal of Mechanical Design*, 112:223, 1990.
23. J. Sartore. Spider monkey images, <http://animals.nationalgeographic.com/animals/mammals/spider-monkey>, Retrieved on 21 December 2011.
24. B. Simmen and D. Sabatier. Diets of some french guianan primates: food composition and food choices. *International Journal of Primatology*, 17(5):661–693, 1996.
25. R. Storn and K. Price. Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces. *INTERNATIONAL COMPUTER SCIENCE INSTITUTE-PUBLICATIONS-TR*, 1997.
26. P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, YP Chen, A. Auger, and S. Tiwari. Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. *KanGAL Report*, 2005005, 2005.
27. M.M.F. Symington. Fission–fusion social organization in *ateles* and *pan*. *International Journal of Primatology*, 11(1):47–61, 1990.
28. M.G.M. van Roosmalen and Instituto Nacional de Pesquisas da Amazônia. Habitat preferences, diet, feeding strategy and social organization of the black spider monkey (*ateles paniscus paniscus* linnaeus 1758) in surinam, 1985.
29. J. Vesterstrom and R. Thomsen. A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In *Evolutionary Computation, 2004. CEC2004. Congress on*, volume 2, pages 1980–1987. IEEE, 2004.
30. X. Wang, XZ Gao, and SJ Ovaska. A simulated annealing-based immune optimization method. In *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning, Porvoo, Finland*, pages 41–47, 2008.
31. D.F. Williamson, R.A. Parker, and J.S. Kendrick. The box plot: a simple visual method to interpret data. *Annals of internal medicine*, 110(11):916, 1989.
32. G. Zhu and S. Kwong. Gbest-guided artificial bee colony algorithm for numerical function optimization. *Applied Mathematics and Computation*, 2010.

Table 3: Test problems

Test Problem	Objective function	Search Range	Optimum Value	D	Acceptable Error
Michalewicz	$f_1(x) = -\sum_{i=1}^D \sin x_i (\sin(\frac{i \cdot x_i}{\pi}))^{20}$	$[0, \pi]$	$f_{min} = -9.66015$	10	$1.0E-05$
Step function	$f_2(x) = \sum_{i=1}^n (x_i + 0.5)^2$	$[-100, 100]$	$f(-0.5 \leq x \leq 0.5) = 0$	30	$1.0E-05$
Levy montalvo 1	$f_3(x) = \frac{\Pi}{D}(10\sin^2(\Pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 \times (1 + 10\sin^2(\Pi y_{i+1})) + (y_D - 1)^2)$, where $y_i = 1 + \frac{1}{4}(x_i + 1)$	$[-10, 10]$	$f(-1) = 0$	30	$1.0E-05$
Levy montalvo 2	$f_4(x) = 0.1(\sin^2(3\Pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 \times (1 + \sin^2(3\Pi x_{i+1})) + (x_D - 1)^2(1 + \sin^2(2\Pi x_D)))$	$[-5, 5]$	$f(\mathbf{1}) = 0$	30	$1.0E-05$
Ellipsoidal	$f_5(x) = \sum_{i=1}^D (x_i - i)^2$	$[-D, D]$	$f(1, 2, 3, \dots, D) = 0$	30	$1.0E-05$
Beale	$f_6(x) = [1.5 - x_1(1 - x_2)]^2 + [2.25 - x_1(1 - x_2^2)]^2 + [2.625 - x_1(1 - x_2^3)]^2$	$[-4.5, 4.5]$	$f(3, 0.5) = 0$	2	$1.0E-05$
Kowalik	$f_7(x) = \sum_{i=1}^{11} [a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	$[-5, 5]$	$f(0.192833, 0.190836, 0.123117, 0.135766) = 0.000307486$	4	$1.0E-05$
2D Tripod	$f_8(x) = p(x_2)(1+p(x_1)) + (x_1 + 50p(x_2)(1-2p(x_1))) + (x_2 + 50(1-2p(x_2))) $	$[-100, 100]$	$f(0, -50) = 0$	2	$1.0E-04$
Shifted Rosenbrock	$f_9(x) = \sum_{i=1}^{D-1} (100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2) + f_{bias}$, $z = x - o + 1$, $x = [x_1, x_2, \dots, x_D]$, $o = [o_1, o_2, \dots, o_D]$	$[-100, 100]$	$f(o) = f_{bias} = 390$	10	$1.0E-01$
Shifted Sphere	$f_{10}(x) = \sum_{i=1}^D z_i^2 + f_{bias}$, $z = x - o$, $x = [x_1, x_2, \dots, x_D]$, $o = [o_1, o_2, \dots, o_D]$	$[-100, 100]$	$f(o) = f_{bias} = -450$	10	$1.0E-05$
Shifted Rastrigin	$f_{11}(x) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10) + f_{bias}$, $z = (x - o)$, $x = (x_1, x_2, \dots, x_D)$, $o = (o_1, o_2, \dots, o_D)$	$[-5, 5]$	$f(o) = f_{bias} = -330$	10	$1.0E-02$
Shifted Schwefel	$f_{12}(x) = \sum_{i=1}^D (\sum_{j=1}^i z_j)^2 + f_{bias}$, $z = x - o$, $x = [x_1, x_2, \dots, x_D]$, $o = [o_1, o_2, \dots, o_D]$	$[-100, 100]$	$f(o) = f_{bias} = -450$	10	$1.0E-05$
Shifted Griewank	$f_{13}(x) = \sum_{i=1}^D \frac{z_i^2}{4000} - \prod_{i=1}^D \cos(\frac{z_i}{\sqrt{i}}) + 1 + f_{bias}$, $z = (x - o)$, $x = [x_1, x_2, \dots, x_D]$, $o = [o_1, o_2, \dots, o_D]$	$[-600, 600]$	$f(o) = f_{bias} = -180$	10	$1.0E-05$
Shifted Ackley	$f_{14}(x) = -20 \exp(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D z_i^2}) - \exp(\frac{1}{D} \sum_{i=1}^D \cos(2\pi z_i)) + 20 + e + f_{bias}$, $z = (x - o)$, $x = (x_1, x_2, \dots, x_D)$, $o = (o_1, o_2, \dots, o_D)$	$[-32, 32]$	$f(o) = f_{bias} = -140$	10	$1.0E-05$
Goldstein-Price	$f_{15}(x) = (1 + (x_1 + x_2 + 1)^2 \cdot (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)) \cdot (30 + (2x_1 - 3x_2)^2 \cdot (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2))$	$[-2, 2]$	$f(0, -1) = 3$	2	$1.0E-14$
Easom's function	$f_{16}(x) = -\cos x_1 \cos x_2 e^{((-(x_1 - \Pi)^2 - (x_2 - \Pi)^2))}$	$[-10, 10]$	$f(\pi, \pi) = -1$	2	$1.0E-13$

to be cont'd on next page

Table 3: Test problems (Cont.)

Test Problem	Objective function	Search Range	Optimum Value	D	Acceptable Error
Dekkers and Aarts	$f_{17}(x) = 10^5 x_1^2 + x_2^2 - (x_1^2 + x_2^2)^2 + 10^{-5}(x_1^2 + x_2^2)^4$	[-20,20]	$f(0, 15) =$ $f(0, -15) = -24777$	2	$5.0E - 01$
McCormick	$f_{18}(x) = \sin(x_1 + x_2) + (x_1 - x_2)^2 - \frac{3}{2}x_1 + \frac{5}{2}x_2 + 1$	$-1.5 \leq x_1 \leq 4,$ $-3 \leq x_2 \leq 3$	$f(-0.547,$ $-1.547) = -1.9133$	30	$1.0E - 04$
Meyer and Roth	$f_{19}(x) = \sum_{i=1}^5 (\frac{x_1 x_3 t_i}{1+x_1 t_i + x_2 v_i} - y_i)^2$	[-10, 10]	$f(3.13,$ $15.16, 0.78) =$ 0.4×10^{-4}	3	$1.0E - 03$
Shubert	$f_{20}(x) = -\sum_{i=1}^5 i \cos((i+1)x_1 + 1) \sum_{i=1}^5 i \cos((i+1)x_2 + 1)$	[-10, 10]	$f(7.0835, 4.8580) =$ -186.7309	2	$1.0E - 05$
Sinusoidal	$f_{21}(x) = -[A \prod_{i=1}^n \sin(x_i - z) + \prod_{i=1}^n \sin(B(x_i - z))], A = 2.5, B = 5, z = 30$	[0, 180]	$f(\mathbf{90} + \mathbf{z}) = -(A + 1)$	10	$1.00E-02$