

# Fast Convergent Spider Monkey Optimization Algorithm

Neetu Agarwal<sup>(✉)</sup> and S.C. Jain

Rajasthan Technical University, Kota, India  
neetuagarwal974@gmail.com

**Abstract.** Spider Monkey Optimization (SMO) is a recent optimization method, which has drawn interest of researchers in different areas because of its simplicity and efficiency. This paper presents an effort to modify Spider Monkey Optimization Algorithm with higher exploitation capabilities. A new acceleration coefficient based strategy is proposed in the basic version of SMO. The proposed algorithm is named as Fast Convergent Spider Monkey Optimization Algorithm (FCSMO). FCSMO is tested over 14 benchmark test functions and compared with basic SMO. The result reveals that FCSMO will surely become a good variant of SMO.

**Keywords:** Meta-heuristic optimization techniques · Swarm intelligence · Acceleration coefficient

## 1 Introduction

Optimization works to unfolds all potential outputs meeting some stated constraints. From past years NIA have proposed various methods to explain NP-Hard and NP- complete optimization problems of real world [7]. Nature inspired algorithm, inspired by nature, is a stochastic approach wherein an individual or a neighbor's interacts with each other intellectually to explain complicated preexisting mechanisms in an efficient manner. NIA is focused mainly on evolutionary based algorithm and swarm based algorithm. Evolutionary algorithm is a computational standard motivated by Darwinian Evolution [9]. Swarm intelligence assets in unlocking optimization problems considering collaborative nature of self-sustaining creatures like bees, ants, monkeys whose food-gathering capabilities and civilized characteristics have been examined and simulated [5, 6, 8]. SMO is a subclass of swarm intelligence, proposed by Jagdish Chand Bansal et al., in the year 2014 [4]. SMO is a food foraging based algorithm, considering nature and social frame work of spider monkeys. Fission-Fusion social system relates to social configuration of spider monkey. Many researchers have been studied that SMO algorithm is good at exploration and exploitation but there is possibilities of further improvements.

To improve the convergence speed, a variant of SMO is proposed, i.e. name as Fast Convergent Spider Monkey Optimization Algorithm. In the proposed modification acceleration coefficients based strategy is incorporated in the basic version of SMO.

The rest of the paper is structured as follows: In Sect. 2, SMO is described. Fast Convergent Spider Monkey Optimization Algorithm (FCSMO) is proposed in Sect. 3. In Sect. 4, performance of FCSMO is tested with several benchmark functions. Finally, Sect. 5 includes a summary and conclude the work.

## 2 Overview of Spider Monkey Optimization (SMO) Technique

A distinct class of NIA proposed by JC Bansal et al. [4], by trivial behavior of monkeys i.e. Spider Monkey Optimization (SMO) technique. Spider Monkey optimization, a Fission-Fusion mode is an extension of above discussed predicament. Here, a populous, consistently dictated by a female, is fragmented into tiny clusters for seeking, chiefly food and they are buddy up to 40 to 50 singular who rift into small groups in search of food who again are headed by a female. In case she fails to meet the objective (food finding), further subdivides, again succeeded by a female, replicating the process until reach the food. For recent updates in their positions, various steps are undertaken: inspection of probing of wide search space and picking or electing of superlative practical results [10].

### 2.1 Steps of SMO Technique

SMO technique is based on population repetitive methodology. It consists of seven steps. Each step is described below in a detailed manner:

1. **Initialization of Population:** Originally a population comprised of N spider monkeys signifying a D-dimensional range  $M_i$  where  $i=1,2,\dots,N$  and  $i$  represents  $i^{th}$  spider monkey. Each spider monkey (M) exhibits possible results of the problem under consider. Each  $M_i$  is initialized as below:

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

Here  $M_{minj}$  and  $M_{maxj}$  are limits of  $M_i$  in  $j^{th}$  vector and  $R(0,1)$  is a random number  $(0,1)$ .

2. **Local Leader Phase (LLP):** This phase relies on the observation of local leader and group mates, M renew its current position yielding a fitness value. If the fitness measure of the current location is larger than that of the former location, then M modifies his location with the latest one. Hence  $i^{th}$  M that also exists in  $k^{th}$  local group modify its position.

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

Here  $M_{ij}$  define  $i^{th}$  M in  $j^{th}$  dimension,  $LL_{kj}$  correlate to the  $k^{th}$  leader of local assembly location in  $j^{th}$  dimension.  $M_{rj}$  defines  $r^{th}$  M which is randomly picked from  $k^{th}$  troop such that  $r \neq i$  in  $j^{th}$  dimension.

3. **Global Leader Phase (GLP):** This following phase initiates just after accomplishing LLP. Depending upon the observation of global leader and mates of local troop, M updates their location. The position upgrade equation for GLP phase is as follows:

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

Here  $GL_j$  poises for global leader's location in  $j^{th}$  dimension and  $j=1,2,3,\dots,D$  defines an arbitrarily chosen index.  $M_i$  modify their locus considering probabilities  $Pr_i$ 's. Fitness is used to calculate probability of a specific solution, with various methods such as

$$Pr_i = 0.1 + \left( \frac{fitness_i}{fitness_{max}} \right) \times 0.9 \quad (4)$$

4. **Global Leader Learning (GLL) Phase:** Here greedy selection strategy is applied on the population which modifies the locus of global leader i.e. the location of M which has best fitness in the group is chosen as the modified global leader location. Also its is verified that global leader location is modifying or not and in case not then GlobalLimitCount(GLC) is increased by 1.
5. **Local Leader Learning (LLL) Phase:** Here, local leader locus is modified by implement greedy selection in that population i.e. the location of M which has best fitness among the entire group is chosen as the latest location of local leader. Afterwards, this modified local leader location and old values are compared and LocalLimitCount (LLC) is increment by 1.
6. **Local Leader Decision (LLD) Phase:** Here, updating of local leader location is done in two ways i.e. by arbitrary initialization or by mixing information obtained via global and local leader, if local leader location is not modified up to a precalculated limit named as LocalLeaderLimit through equation based on perturbation rate (p).

$$M_{newij} = M_{ij} + R(0, 1) \times (GL_j - M_{ij}) + R(0, 1) \times (M_{ij} - LL_{kj}) \quad (5)$$

Clearly, it is seen in equation that modified dimension of this M is fascinated towards global leader and oppose local leader. Moreover, modified M's fitness is determined.

7. **Global Leader Decision (GLD) Phase:** Here, global leader location is examine and if modification is not done up to precalculated iterations limit named as GlobalLeaderLimit then division of population in small group is done by local leader. Primarily population division is done in two classes and further three, four and so on until the upper bound called groups of maximum number (GM) is reached. Meanwhile, local leaders are selected using LL method for newly formed subclasses.

The pseudo-code of the SMO algorithm is as follows:-

- (1) Define Population, LocalLeaderLimit, GlobalLeaderLimit, Perturbation rate.
  - (2) Determine fitness (each individual distance from sources of food)
  - (3) Apply greedy selection to choose global and local leaders.
- while** *Termination condition is not met* **do**
- (i) To hit target, new locations for group population is formulated with the help of self experience as well as local and group population experience, using Local Leader Phase (LLP).
  - (ii) Relied on fitness value of group members, employ greedy selection strategy.
  - (iii) Assess probabilities  $Pr_i$  for all companions using equation (4).
  - (iv) Generate new locations for each group companions, chosen by  $Pr_i$ , by self experience, global leader experience also consider experience of group member using Global Leader Phase (GLP).
  - (v) Greedy selection method is applied to modify global and local leaders locations of entire groups.
  - (vi) Any local leader of a group, if fails to modify her locus within LocalLeaderLimit then deflect that specific group companions for further foraging using Local Leader Decision (LLD) Phase.
  - (vii) Any global leader if fails to modify her locus within GlobalLeaderLimit then she diversifies group into subgroups by Global Leader Decision Phase with the minimum threshold of each groups size being 4
- end**

**Algorithm 1.** Spider Monkey Optimization (*SMO*)

### 3 Fast Convergent Spider Monkey Optimization Algorithm

In population repetitive methodology, exploration and exploitation are the two basic properties of NIA. A convenient balance between both these two properties are required. Exploration describe the promising regions by searching the given search space while exploitation helps in finding the optimal solution in the promising search regions. In the basic SMO, it is good at exploration and exploitation but there is possibilities of further improvements. So to improve the basic SMO, on new variant named Fast Convergent Spider Monkey Optimization Algorithm (FCSMO) is designed.

From the results of search process of basic SMO that it will get higher opportunity for advancement in various iteration using two ways: (1) Global Leader Phase (GLP) and (2) Local Leader Decision (LLD) phase. Exploration and exploitation capacity should be managed in an effective way. The details of these two steps of FCSMO implementation are explained below:

1. **Global Leader Phase (GLP):** In the GLP phase, depending upon the observation of Global leader and mates of local troop, M updates their location. In the GLP phase of iteration, solutions in search space are having

large step-size resulting in exploration. In later iterations, there is a gradual decrease in step-size by moving slowly iteration by iteration due to which solution exploits the search space well and resulting good convergence. The position upgrade equation for this phase is as follow:

$$M_{newij} = M_{ij} + R(0, 1) \times (GL_j - M_{ij}) + (M_{rj} - M_{ij}) \times \left[ 1 - \left( \frac{iter}{Maxiteration} \right) \right] \times c \quad (6)$$

Here  $GL_j$  poises for global leader's location in  $j^{th}$  dimension and  $j=1,2,3,\dots,D$  defines an arbitrarily chosen index and iter and Max iteration show the present iteration and the maximum iteration number, respectively.  $c$  is the random number. Its value is 2. In this acceleration coefficient is added with random member. The position upgrade method of GLP phase is exhibited in following algorithm.

```

count=0;
while count < group do
  for each member  $M_i \in$  group do
    if  $R(0,1) < Pr_i$  then
      count=count+1;
      Randomly select  $j \in 1..D$ 
      Randomly select  $M_r \in$  group s.t  $r \neq i$ 
       $M_{newij} = M_{ij} + R(0, 1) \times (GL_j - M_{ij}) + (M_{rj} - M_{ij}) \times \left[ 1 - \left( \frac{iter}{Maxiteration} \right) \right] \times c$ 
    end if
  end for
end

```

**Algorithm 2.** Global Leader Phase (GLD)

- Local Leader Decision (LLD) Phase:** Here, updating of Local Leader location is done in two ways i.e. by arbitrary initialization or by mixing information obtained via global and local leader, if local leader location is not modified up to a precalculated limit named as LocalLeaderLimit through equation based on  $p$ . In the LLD phase of iteration, solutions in search space are having large step-size resulting in exploration. In later iterations, there is a gradual decrease in step-size by moving slowly iteration by iteration due to which solution exploits the search space well and resulting good convergence. The position upgrade equation for this phase is as follow:

$$M_{newij} = M_{ij} + (GL_j - M_{ij}) \times \left( 1 - \left( \frac{iter}{Maxiteration} \right) \right) + R(0, 1) \times (M_{ij} - LL_{kj}) \quad (7)$$

Clearly, it is seen in equation that modified dimension of this  $M$  is fascinated towards global leader and oppose local leader. Moreover, modified  $M$ 's fitness is determined and iter and Max iteration show the present iteration and the maximum iteration number, respectively. In this acceleration coefficient is added with global leader. The position upgrade method of LLD phase is exhibited in following algorithm.

```

if LocalLimitCount > LocalLeaderLimit then
  LocalLimitCount=0
  for each  $j \in 1 \dots D$  do
    if  $R(0,1) \geq p$  then then
       $M_{new_{ij}} = M_{ij} + R(0,1) \times (M_{maxj} - M_{minj})$ 
    else
       $M_{new_{ij}} = M_{ij} + (GL_j - M_{ij}) \times (1 - (\frac{iter}{Maxiteration})) + R(0,1) \times (M_{ij} - LL_{kj})$ 
    end if
  end for
end if

```

**Algorithm 3.** Local Leader Decision (*LLD*) Phase

The pseudo-code of the FCSMO algorithm is as follows:-

- (1) Define Population, LocalLeaderLimit, GlobalLeaderLimit, Perturbation rate.
- (2) Determine fitness (each individual distance from sources of food)
- (3) Apply greedy selection to choose global and local leaders.

**while** *Termination condition is not met* **do**

- (i) To hit target, new locations for group population is formulated with the help of self experience as well as local and group population experience, using Local Leader Phase (LLP).
- (ii) Relied on fitness value of group members, employ greedy selection strategy.
- (iii) Assess probabilities  $Pr_i$  for all companions using equation (4).
- (iv) Generate new locations for each group companions, chosen by  $Pr_i$ , by self experience, global leader experience also consider experience of group member using algorithm 2.
- (v) Greedy selection method is applied to modify global and local leaders locations of entire groups.
- (vi) Any local leader of a group, if fails to modify her locus within LocalLeaderLimit then defect that specific group companions for further foraging using algorithm 3.
- (vii) Any global leader if fails to modify her locus within GlobalLeaderLimit then she diversifies group into subgroups by Global Leader Decision Phase with the minimum threshold of each groups size being 4

**end**

**Algorithm 4.** Fast Convergent Spider Monkey Optimization (*FCSMO*)

## 4 Experimental Results

### 4.1 Test Problems Under Consideration

To evaluate the quality of proposed FCSMO algorithm, 14 opposed global optimization issue ( $f_1 - f_{14}$ ) are selected as presented in Table 1. All the issues are continuous optimization issues and having various rates of complexity. Test problems ( $f_1 - f_{14}$ ) are yield from [1,11] with the correlated offset values.

**Table 1.** Test problems

Test Problem	Objective function	Search Space	Objective Value	Dimension	Acceptable Error
Rastrigin	$f_1(x) = 10D + \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i)]$	[-5.12,5.12]	$f(0) = 0$	30	1.00E-05
Ackley	$f_2(x) = -20 + e + e \exp(-\frac{0.2}{D} \sqrt{\sum_{i=1}^D x_i^2})$	[-30,30]	$f(0) = 0$	30	1.00E-05
Michalewicz	$f_3(x) = -\sum_{i=1}^D \sin x_i (\sin(\frac{i x_i^2}{\pi}))^{20}$	$[0, \pi]$	$f_{min} = -9.66015$	= 10	1.00E-05
Cosin Mixture	$f_4(x) = \sum_{i=1}^D x_i^2 - 0.1(\sum_{i=1}^D \cos 5\pi x_i) + 0.1D$	[-1,1]	$f(0) = -D \times 0.1$	= 30	1.00E-05
Step Function	$f_5(x) = \sum_{i=1}^D ((x_i + 0.5))^2$	[-100,100]	$f(-0.5 \leq x \leq 0.5) = 0$	= 30	1.00E-05
Neumaier Problem (NF3)	$f_6(x) = \sum_{i=1}^D (x_i - 1)^2 - \sum_{i=2}^D x_i x_{i-1}$	[-900,900]	$f(0) = \frac{-(D * (D + 4) * (D - 1))}{6.0}$	= 10	1.00E-05
Levy montalvo 1	$f_7(x) = \frac{\pi}{D} (10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 \times (1 + 10 \sin^2(\pi y_{i+1})) + (y_D - 1)^2)$ , where $y_i = 1 + \frac{1}{4}(x_i + 1)$	[-10,10]	$f(-1) = 0$	= 30	1.00E-05
Levy montalvo 2	$f_8(x) = 0.1(\sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 \times (1 + \sin^2(3\pi x_{i+1})) + (x_D - 1)^2 (1 + \sin^2(2\pi x_D)))$	[-5,5]	$f(1) = 0$	= 30	1.00E-05
Shifted Griewank	$f_9(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}}) + 1 + f_{bias}$ , $z = (x - o)$ , $x = [x_1, x_2, \dots, x_D]$ , $o = [o_1, o_2, \dots, o_D]$	[-600,600]	$f(o) = -180$	= 10	1.00E-05
Goldstein-Price Function	$f_{10}(x) = (1 + (x_1 + x_2 + 1)^2 \cdot (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)) \cdot (30 + (2x_1 - 3x_2)^2 \cdot (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2))$	[-2,2]	$f(0, -1) = 3$	= 2	1.0E - 14
Six-Hump Camel back Function	$f_{11}(x) = (4 - 2.1x_1^2 + x_1^4/3)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$	[-5,5]	$f(-0.0898, 0.7126) = -1.0316$	= 26	1.0E - 05
Hosaki Problem (HSK)	$f_{12} = (1 - 8x_1 + 7x_1^2 - 7/3x_1^3 + 1/4x_1^4)x_2^2 \exp(-x_2)$	$x_1 \in [0, 5], x_2 \in [0, 6]$	$f(0) = -2.3458$	= 2	1.00E-06
Sinusoidal Problem (SIN)	$f_{13}(x) = -[A \prod_{i=1}^D \sin(x_i - z) + \prod_{i=1}^D \sin(B(x_i - z))]$ , $A = 2.5, B = 5, z = 30$	[0,180]	$f(90 + z) = -(A + 1)$	= 10	1.00E-02
Pressure Vessel	$f_{14}(x) = 0.6224 \times x_1 \times x_3 \times x_4 + 1.7781 \times x_2 \times x_3^2 + 3.1611 \times x_1^2 \times x_4 + 19.84 \times x_1^2 \times x_3$	$x_1 \in [1.125, 112.5], x_2 \in [0.625, 12.5], x_3 \in [1.0E - 8, 240], x_4 \in [1.0E - 8, 240]$	$f(0) = 7197.729$	= 4	1.00E-05

### 4.2 Experimental Setting

To verify the efficiency of proposed algorithm FCSMO, a relative study is taken between FCSMO and SMO. To analysis FCSMO and basic SMO, over the examine testing issues, subsequent observational setting is emulated:

- The number of simulations/run =100,
- Population size (Monkeys)  $NP = 50$
- $R = rand[0, 1]$
- GlobalLeaderLimit  $\in [N/2, 2 \times N]$  [4]
- LocalLeaderLimit =  $D \times N$  [4]
- Perturbation rate (p)  $\epsilon [0.1, 0.8]$

### 4.3 Results Comparison

Table 2 represent the observational results of relative algorithm. Following Table 2 gives a information about Standard Deviation (SD), Mean Error (ME),

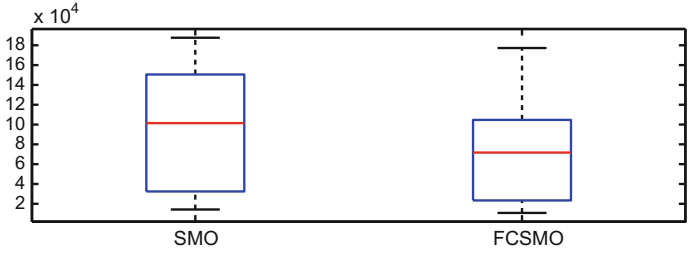
**Table 2.** Comparison of the results of test functions, TP: Test Problem

TP	Algorithm	SD	ME	AFE	SR
$f_1$	FCSMO	1.81E-06	7.99E-06	88298.07	100
	SMO	1.56E-06	8.24E-06	96073.45	100
$f_2$	FCSMO	4.79E-07	9.36E-06	23416.47	100
	SMO	9.32E-07	9.26E-06	32438.70	100
$f_3$	FCSMO	3.50E-06	5.52E-06	32438.20	100
	SMO	3.72E-06	4.98E-06	52153.75	100
$f_4$	FCSMO	4.01E-02	1.18E-02	32651.24	92
	SMO	4.80E-02	1.77E-02	62144.30	88
$f_5$	FCSMO	0.00E+00	0.00E+00	10792.21	100
	SMO	0.00E+00	0.00E+00	14261.81	100
$f_6$	FCSMO	4.38E-07	9.80E-06	124902.53	100
	SMO	1.15E-05	1.23E-05	169082.28	90
$f_7$	FCSMO	8.70E-07	9.02E-06	12917.52	100
	SMO	1.45E-02	2.08E-03	20795.99	98
$f_8$	FCSMO	1.09E-06	8.80E-06	12867.18	100
	SMO	1.53E-03	2.28E-04	17814.27	98
$f_9$	FCSMO	1.23E-03	1.39E-04	96079.51	97
	SMO	4.27E-03	1.64E-03	134399.12	81
$f_{10}$	FCSMO	4.70E-14	6.18E-14	130131.22	38
	SMO	4.31E-14	7.05E-14	150535.16	28
$f_{11}$	FCSMO	1.48E-05	1.68E-05	104691.82	50
	SMO	1.42E-05	1.72E-05	106712.17	49
$f_{12}$	FCSMO	4.11E-06	9.90E-06	177221.60	15
	SMO	3.39E-06	1.05E-05	187621.00	10
$f_{13}$	FCSMO	5.05E-03	8.29E-03	55146.68	97
	SMO	5.38E-03	1.11E-02	154825.33	62
$f_{14}$	FCSMO	1.07E-03	1.35E-04	89666.40	67
	SMO	3.63E-05	3.39E-05	111436.91	57

Average Number of Function valuations ( $AFE$ ) and Success Rate ( $SR$ ). According to Results of Table 2, at maximum time FCSMO shows best results from SMO, in terms of performance, efficiency and accuracy.

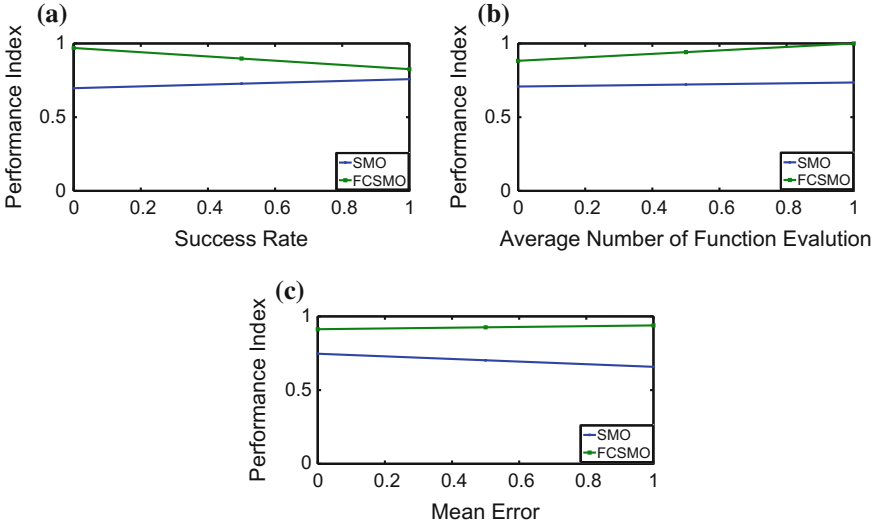
Moreover, boxplots evaluation of  $AFE$  is taken for comparing the relevant algorithms in scheme of consolidated quality, so it can simply show the observed distribution of statistic graphically. The boxplots for FCSMO and SMO are presented in Fig. 1. The results declares that interquartile scope and medians of FCSMO are comparatively low. Further, all relevant algorithms are studied by allowing entire attention to the  $SR$ ,  $AFE$  and  $ME$ . This study is determined





**Fig. 1.** Boxplots graph for average function evaluation

using the quality basis i.e. represented in [2,3]. The evaluated values of *PI* for the FCSMO and SMO are calculated and consecutive *PIs* graphs are represented in Fig. 2. The graphs analogous to several cases i.e. allowing entire attention to SR, AFE and ME (as explained in [2,3]) are represent in Figs. 2(a), (b), and (c) respectively. In these diagram, horizontal axis means the weights and vertical axis means the *PI*. It is clear from Fig. 2 that *PI* of FCSMO are superior than the other studied algorithms in various case. i.e. FCSMO observe better on the studied testing issues as compare to the SMO.



**Fig. 2.** Performance index for test problems; (a) for weighted importance to *SR*, (b) for weighted importance to *AFE* and (c) for weighted importance to *ME*.

## 5 Conclusion

This paper presents a variant of SMO algorithm, known as Fast Convergent Spider Monkey Optimization Algorithm (FCSMO). In FCSMO, an acceleration coefficient based strategy is proposed in which step size is decreased through iteration. To evaluate the proposed algorithm, it is tested over 14 benchmark function. The results collected by the FCSMO is better than the basic SMO algorithm. In future, newly developed algorithm may be used to solve various real-world optimization problems of continuous nature.

## References

1. Ali, M.M., Khompatraporn, C., Zabinsky, Z.B.: A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. *J. Global Optim.* **31**(4), 635–672 (2005)
2. Bansal, J.C., Sharma, H.: Cognitive learning in differential evolution and its application to model order reduction problem for single-input single-output systems. *Memetic Comput.* **4**(3), 209–229 (2012)
3. Bansal, J.C., Sharma, H., Arya, K.V., Nagar, A.: Memetic search in artificial bee colony algorithm. *Soft. Comput.* **17**(10), 1911–1928 (2013)
4. Bansal, J.C., Sharma, H., Jadon, S.S., Clerc, M.: Spider monkey optimization algorithm for numerical optimization. *Memetic Comput.* **6**(1), 31–47 (2014)
5. Bonabeau, E., Dorigo, M., Theraulaz, G.: *Swarm Intelligence: From Natural to Artificial Systems*, Number 1. Oxford University Press, New York (1999)
6. Colorni, A., Dorigo, M., Maniezzo, V., et al. Distributed optimization by ant colonies. In: *Proceedings of the first European Conference on Artificial Life*, vol. 142, pp. 134–142. Paris, France (1991)
7. Fister Jr., I., Yang, X.-S., Fister, I., Brest, J., Fister, D.: A brief review of nature-inspired algorithms for optimization. arXiv preprint [arXiv:1307.4186](https://arxiv.org/abs/1307.4186) (2013)
8. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical report, Technical report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department (2005)
9. Price, K.V.: Differential evolution: a fast and simple numerical optimizer. In: *Fuzzy Information Processing Society, 1996. NAFIPS., 1996 Biennial Conference of the North American*, pp. 524–527. IEEE (1996)
10. Sharma, A., Sharma, H., Bhargava, A., Sharma, N.: Power law-based local search in spider monkey optimisation for lower order system modelling. *Int. J. Syst. Sci.* 1–11 (2016)
11. Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.-P., Auger, A., Tiwari, S.: Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. KanGAL report, 2005005:2005 (2005)