1st IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES-2016)

Adaptive Step-size based Spider Monkey Optimization

Garima Hazrati[1], Harish Sharma[2] and Nirmala Sharma[3]
[1,2,3]Department of Computer Science and Engineering, Rajasthan Technical University
E-mail: [1]ghazrati9@gmail.com, [2]hsharma@rtu.ac.in, [3]nsharma@rtu.ac.in

***Abstract*—Spider Monkey Optimization (SMO) algorithm is recent swarm intelligence based meta-heuristic technique to solve the continuous optimization problems. Many times, it suffers from the problem of slow convergence. To improve the exploitation abilities and evading premature convergence, a modified variant of SMO is proposed. The modified variant is known as adaptive step-size based spider monkey optimization (AsSMO) algorithm. In position update process of AsSMO, step-size is calculated by the fitness of a spider monkey. The prominent fit solutions will converge quickly in comparison to the non-prominent fit solutions. The proposed algorithm is compared with SMO and self-adaptive spider monkey optimization (SaSMO) over 15 benchmark functions and reported results show that AsSMO is a spell variant among them.**

Keywords—Meta-heuristic; Swarm Intelligence; Nature Inspired Algorithms; Spider Monkey Optimization

I. INTRODUCTION

Nature provides many efficient ways to solve complex problems and algorithms inspired by nature imitating processes in nature are known as Nature Inspired Algorithms (NIA) [1]. NIA is composed of two fields that are evolutionary and swarm intelligence algorithms, respectively [2, 3]. Evolutionary algorithms are based on the concept of survival of the fittest [4] like Differential evolution [5], Cuckoo search [6] etc… fall in this category. Swarm intelligence is inspired by the collective behavior of species in nature like ants, birds, monkeys, bats, humans etc… Artificial bee colony [7], Particle swarm optimization [8], Biogeography based optimization [9], Shuffled frog leaping algorithm [10], Gravitational search algorithm [11], and Ant colony optimization [12] are some swarm intelligence based algorithms. The spider monkey optimization algorithm is a new addition in the class of swarm intelligence based algorithms developed by J.C. Bansal et al. [13] based on food foraging behavior of spider monkeys (SM). It is performing well in terms of reliability and efficiency than other well established meta-heuristic techniques and removes their drawback too [14]. Since, SMO has an intrinsic downside of premature convergence and being stuck in local optima.

In this paper, a modified variant of SMO is proposed that is named as Adaptive step-size based spider monkey optimization algorithm. The proposed variant is developed for improving exploration and exploitation of the algorithm by adaptive step-size generated with the help of fitness of SM and change in location of global leader by initializing it again due to which it can't be stuck in local optima.

The rest of the paper is organized as follows: Section 2 describes the spider monkey optimization algorithm. In Section 3 the proposed algorithm Adaptive step-size SMO (AsSMO) is presented. Section 4 shows the results and discussions followed by conclusion in Section 5.

II. SPIDER MONKEY OPTIMIZATION ALGORITHM

The intelligent foraging behavior of spider monkeys (SM) based on fission-fusion structure is an inspiration for researchers to develop SMO algorithm [13].Competition among group members for food leads to fission and fusion of group. When there is food scarcity group divides into small groups else they combine showing a perfect fission-fusion social structure.

SMO has seven phases which are initialization phase, local leader phase, global leader phase, local leader learning phase, global leader learning phase, local leader decision phase, and global leader decision phase. These phases are described as follows:

A. Initialization

In the initialization phase, an initial population of N spider monkeys is generated, where SM_i represents the i^{th} SM in the population. Each SM_i is initialized as shown in equation (1):

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

Where, SM_{minj} and SM_{maxj} are lower and upper bounds of the search space in j^{th} dimension and Φ is a uniformly distributed random number in the range [0, 1].

1) Local leader phase

All spider monkeys update themselves in this phase, based on their local leader and local group member's experience. The fitness of SM checks position update of SM at a new position. If fitness is high, then SM change its position else not. Here, position update process is given in equation (2):

$$SMnew_{ij} = SM_{ij} + \Phi \times (LL_{kj} - SM_{ij}) + \Psi \times (SM_{rj} - SM_{ij}) \tag{2}$$

Where, SM_{ij} is the j^{th} dimension of i^{th} SM, LL_{kj} represents the k^{th} local leader of that group and SM_{rj} is r^{th}

SM chosen arbitrarily within kth group in jth dimension such that r ≠ i and Ψ is uniformly distributed random number in the range [-1,1].

2) Global leader phase

Each SM uses knowledge of global leader and experience of neighboring SM with its previous position to update its position and reaching to an optimum solution. The position update equation in this phase is as follows:

$$SMnew_{ij} = SM_{ij} + \Phi \times (GL_j - SM_{ij}) + \Psi \times (SM_{rj} - SM_{ij}) \quad (3)$$

Where, GL_j is the position of group leader in jth dimension and j ∈ {1, 2, 3, …, D} i.e. chosen randomly.

3) Local leader learning phase

The SM whose fitness is highest among the swarm is considered as the global leader of the population. If the position of the global leader is doesn't get updated then the counter associated with the global leader, named as Global Limit Count (GLC), is incremented by 1 otherwise set to 0.

4) Global leader learning phase

In this segment, the position of local leader gets updated by applying greedy selection among the group members. If a local leader doesn't update its position then a counter, associated with the local leader, called Local Limit Count (LLC), is incremented by 1 else counter is set to 0. This process is applied for every group to find their local leaders respectively.

5) Local leader decision phase

If any local leader doesn't get reorganized to a particular verge known as Local Leader Limit (LLL), then all the members of that group update their positions either by random initialization or by using global leader experience through pr i.e. perturbation rate given in equation (4):

$$SMnew_{ij} = SM_{ij} + \Phi \times (GL_j - SM_{ij}) + \Psi \times (SM_{rj} - LL_{kj}) \quad (4)$$

6) Global leader decision phase

If the global leader doesn't get reorganized to a particular verge known as Global Leader Limit (GLL), then the global leader divides the population into smaller groups or fuse groups into one unit group. If GLC is more than GLL, then GLC is set to zero and number of groups are compared to maximum groups. If existing quantity of groups is less than the pre-defined maximum number of groups, then the global leader further divides the groups otherwise combined to form a single group.

Algorithm 1 Global Leader Decision Phase (GLD):
ifGlobalLimitCount>GlobalLeaderLimit**then**
GlobalLimitCount = 0
if Number of groups < MG **then**
Divide the populations into groups
else
Combine all the groups to make a single group
endif
update Local Leaders position
endif

III. Adaptive Step-size based Spider Monkey Optimization Algorithm

SMO is well-balanced algorithm when compared with other swarm intelligence based algorithms [13]. Though it is well-balanced, it has an intrinsic drawback of premature convergence and stagnation. In global leader phase, the position update equation of SMO which is equation (3) contains two control parameters Φ and Ψ. Due to the randomness of the parameters, there is always a chance of skipping the true solution. In global leader decision phase, when GLL is reached, group either divides further or fuse it in a single one but then also there is a probability that leader get stuck at local optima.

For faster convergence and balancing the exploration and exploitation properties of the algorithm, adaptive step-size spider monkey optimization (AsSMO) is proposed. Two phases of the basic version of the SMO are modified that are global leader phase and global leader decision phase.

B. Modified Global Leader Phase

Every position update equation consists of individual's old position and step-size which is shown in equation (3). Step-size of any solution is used to calculate its new position to reach the optima. Step size in basic version of SMO is shown in equation (5):

$$\Phi \times (GL_j - SM_{ij}) + \Psi \times (SM_{rj} - SM_{ij}) \quad (5)$$

Step-size leads to change in individual's position. No solution is virtuous or vile; it only leads to exploitation and exploration of search space. Step-size controls the movement of SM in diverse directions. When a spider monkey has large step-size, then there are more chances that it skip the true solution lying near to it and SM with small step-size have fewer chances to skip the true solution. Φ and Ψ plays a substantial role in deciding the step-size of a spider monkey and are uniformly distributed random number within the range of [0, 1] and [-1,1] respectively. For maintaining the proper balance between the exploration and exploitation capability of the basic algorithm, the position update equation is modified as in equation (6):

$$SMnew_{ij} = SM_{ij} + U(0, 2.2 - prob_i) \times (GL_j - SM_{ij}) + U((2.2 - prob_i), -(2.2 - prob_i)) \times (SM_{rj} - SM_{ij}) \quad (6)$$

Here, $prob_i$ is calculated by equation (7):

$$prob_i = 0.9 \times \frac{fitness_i}{\max\_fitness} + 0.1 \quad (7)$$

It is apparent from the structure of probability in equation (7) that a more fit SM has a high probability as it is a function of fitness. By introducing probability spider monkeys with high probability get more chance to update themselves and reach global optima in less time and not only this they also help other SM to reach the global optima by becoming their random members.

From equation (6), we can conclude that a solution with better probability have less step-size lie near to global optima due to which it doesn't bounce the true solution and exploits the foraging area well. SM having less probability lies far in the foraging area from global optima and because of its probability it has large step-size, and it explores the foraging area well. The proposed modification leads the proper balance between the exploration and exploitation properties of the search space. By good solutions, exploitation of search region is performed due to which chance of skipping of the true solution is reduced while not good enough solutions explore the search space to reach new global optima.

C. *Modified Global Leader Decision Phase*

It is justified from the global leader phase that every SM inspired its position from the global leader. If the global leader gets stuck at local optima, it either split the group further or fuse it in a single one which results in premature convergence. For any swarm intelligence based algorithm, it is needed that it smoothly converge to global optima without being stagnated. To avoid this situation, a modification is proposed in global leader decision phase by providing fluctuations.

If the global leader is not updating its position to a pre-determined limit, then it is randomly initialized in the search space. This will eradicate the situation of stagnation and leads to the exploration of the search space. SM having less fitness is nominated as new global leader of the swarm. The proposed modification is explained in algorithm 2:

Algorithm 2 Modified Global Leader Decision Phase (MGLD):

if (GlobalLimitCount > GlobalLeaderLimit) **then**
GlobalLimitCount = 0
random initialization of global leader
if (Number of groups < MG) **then**
Divide the populations into groups
else
Combine all the groups to make a single group
end if
update Local Leaders position
end if

IV. RESULTS AND DISCUSSIONS

A. *Test Problems Under Consideration*

The projected algorithm AsSMO is tested over 15 benchmark functions to examine its gratification among other well established algorithms. These 15 benchmark functions are shown in Table 1.

TABLE 1: BENCHMARK FUNCTIONS

Function no.	Function Name	Search Range	Optimum Value	Dimension (D)	Acceptable Error
f_1	Levi montalvo 1	[-10, 10]	f(-1)=0	30	1.0E-05
f_2	Levi montalvo 2	[-5,5]	f(1)=0	30	1.0E-05
f_3	Step function	[-100, 100]	$f(-0.5\leq x\leq 0.5)=0$	30	1.0E-05
f_4	Colville	[-10, 10]	f(1)=0	4	1.0E-05
f_5	Ackley	[-1,1]	f(0)=0	30	1.0E-05
f_6	Michalewicz	$[0,\pi]$	f_{min}= -9.66015	10	1.0E-05
f_7	Cosine mix	[-1,1]	f(0)= -D x 0.1	30	1.0E-05
f_8	Salomon	[-100, 100]	f(0)=0	30	1.0E-01
f_9	Inv cos wave fun	[-5,5]	f(0)= -D+1	10	1.0E-05
f_{10}	Neumair 3 prob	$[-D^2, D^2]$	f(0)= -(D x(D+4)(D-1))/6.0	10	1.0E-01
f_{11}	Kowalik	[-5,5]	f(0.1928, 0.1908, 0.1231, 0.1357)=3.07E-04	4	1.0E-05
f_{12}	Shifted rosenbrock	[-100, 100]	$f(0)=f_{bias}=390$	10	1.0E-01
f_{13}	Shifted greiwank	[-600, 600]	$f(0)=f_{bias}= -180$	10	1.0E-05
f_{14}	Hosaki	$X_1 \in[0,5]$, $X_2 \in[0,6]$	-2.3458	2	1.0E-6
f_{15}	Pressure vessel	X_1=[1.1,12.5] X_2=[0.6,12.5] X_3=[0,240] X_4=[0,240]	0	6	1.0E-05

B. *Experimental Settings*

To verify that AsSMO is a contended member in field of swarm intelligence algorithms, comparative analysis is done among AsSMO, SMO [13] and a recent variant SaSMO [15]. Following experimental setup is done for SMO, SaSMO, AsSMO:

General Settings for all three algorithms:
No. of iterations=2000;
No. of run=100;
Total no. of Spider Monkeys (N) =50;
Maximum no. of groups=5;
Rest settings of SMO and SaSMO are taken from their elementary papers [13, 15].

C. *Results*

Table II shows the obtained results of 3 taken algorithms SMO, SaAMO and AsSMO based on above parameter settings. Results are shown in standard deviation (SD), mean error (ME), average function evaluation (AFE) and success rate (SR).

Results in table II shows that AsSMO is a better variant than SMO and SaSMO regarding accuracy, reliability and efficiency. In addition to above results box-plot analysis of compared algorithms in terms of average function evaluations (AFE) is presented. Box-plot analysis [16] of SMO, SaSMO, and AsSMO is shown in figure 1 representing the empirical distribution of data graphically.

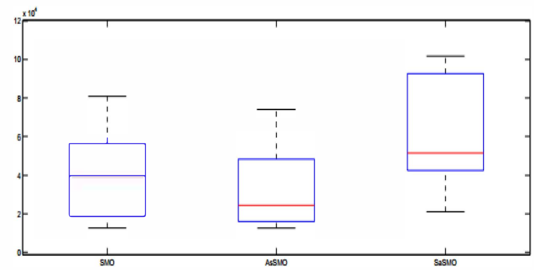

Fig. 1: Box-plot Analysis for Average Function Evaluations

Figure 1 shows that variation, interquartile range and medians of developed AsSMO is less than other two. After this, the comparison is made by using the performance indices (PI) [17] based on ME, SR and AFE. The computed values of PI for SMO, SaSMO and AsSMO are portrayed in Fig. 2.

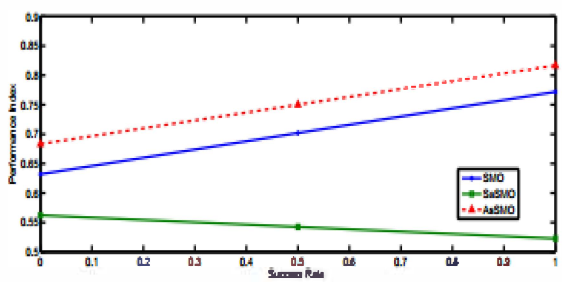

Fig. 2(a)

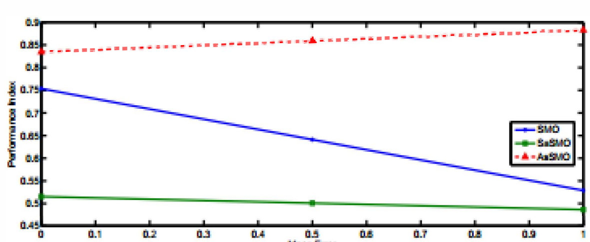

Fig. 2(b)

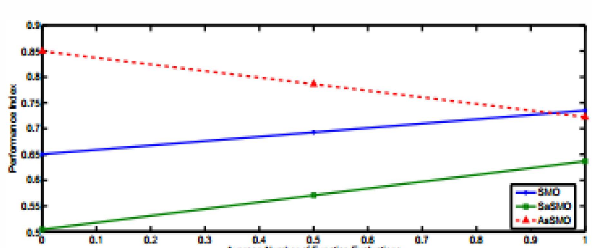

Fig. 2(c)

Fig. 2(a), 2(b), 2(c): Performance Indices for Test Problems

Figure 2(a), 2(b), and 2(c) show the performance index of success rate, mean error, and average function evaluations respectively. Figure 2 shows that PI of AsSMO is notable as compared to other variants.

TABLE 2: COMPARISON OF RESULTS OF TEST PROBLEMS

Test Function	Algorithm	SD	ME	AFE	SR
f_1	SMO	1.03E-02	1.05E-03	18723.73	99
	SaSMO	1.74E-06	8.12E-06	39102.07	100
	AsSMO	1.03E-02	1.05E-03	16002.52	99
f_2	SMO	1.53E-03	2.28E-04	19319.56	98
	SaSMO	1.74E-06	7.99E-06	43708.22	100
	AsSMO	1.80E-06	8.47E-06	16693.71	100
f_3	SMO	0.00E+00	1.60E-07	16239.24	100
	SaSMO	0.00E+00	1.60E-07	21085.75	100
	AsSMO	0.00E+00	1.60E-07	13108.67	100
f_4	SMO	1.98E-04	8.10E-04	52573.96	100
	SaSMO	1.34E-02	8.16E-03	92630.76	39
	AsSMO	2.43E-04	7.54E-04	48484.66	100
f_5	SMO	5.73E-07	9.41E-06	26407.58	100
	SaSMO	1.28E-05	3.25E-05	98963.28	0
	AsSMO	6.41E-07	9.38E-06	25307.23	100
f_6	SMO	4.21E-03	4.95E-04	56524.47	98
	SaSMO	4.88E-04	5.45E-05	54914.81	98
	AsSMO	3.48E-06	5.27E-06	47068.44	100
f_7	SMO	4.43E-02	1.48E-02	52779.84	90
	SaSMO	1.76E-06	8.44E-06	48378.63	100
	AsSMO	3.71E-06	5.71E-06	23502.97	100
f_8	SMO	2.55E-02	1.93E-01	200862.84	7
	SaSMO	1.35E-01	1.56E+00	101746.98	0
	AsSMO	3.67E-02	1.84E-01	192226.67	16
f_9	SMO	5.21E-02	5.25E-03	80817.68	99
	SaSMO	1.56E-01	5.05E-02	91340.72	45
	AsSMO	1.43E-06	8.24E-06	73841.23	100
f_{10}	SMO	6.61E-06	1.15E-05	163912.92	89
	SaSMO	1.47E+01	5.60E+00	99135.42	0
	AsSMO	5.20E-07	9.72E-06	138828.23	100
f_{11}	SMO	8.48E-05	9.85E-05	43030.09	98
	SaSMO	2.04E-04	1.59E-04	80308.81	80
	AsSMO	1.37E-05	8.92E-05	44074.65	100
f_{12}	SMO	9.67E+00	2.50E+00	172472.86	39
	SaSMO	1.35E+00	9.54E-01	94387.43	23
	AsSMO	1.08E+01	3.27E+00	159848.16	45
f_{13}	SMO	4.79E-03	1.79E-03	132298.41	81
	SaSMO	2.83E-03	9.19E-04	59303.68	86
	AsSMO	2.65E-03	8.20E-04	123984.25	88
f_{14}	SMO	2.61E-06	1.10E-05	198003.46	5
	SaSMO	3.87E-06	1.02E-05	90694.34	13
	AsSMO	4.08E-06	1.00E-05	178942.76	14
f_{15}	SMO	2.15E-04	6.02E-05	114575.91	55
	SaSMO	2.02E+00	1.86E+00	103297.65	0
	AsSMO	3.57E-05	3.00E-05	98856.48	63

V. CONCLUSION

This paper presents a modified variant of SMO, namely adaptive step-size spider monkey algorithm (AsSMO). The intended algorithm is developed for evading the intrinsic downside and improving exploration and exploitation abilities of SMO. Firstly, modified global leader phase uses a function of fitness to improve step-size as if the probability is high then exploitation of search space is done else larger step-size results in the exploration of search space. Secondly, in modified global leader decision phase global leader get stuck in local optima so, for better convergence speed global leader is initialized randomly. Through this modification, it is unblemished that step-size is adjusted according to SM fitness exquisitely depicting the concept and name of algorithm adaptive step-size SMO as SM adapt their step-size according to their fitness. To appraise the intensity AsSMO, it is tested over 15 benchmark functions and results show that it is spell variant. In future, it can be applied to real-world optimization problems and complex optimization problems.

REFERENCES

[1] Xin-She Yang. Nature-Inspired metaheuristic algorithms, Luniver press, 2010.

[2] Dipankar Dasgupta and Zbigniew Michalewicz. Evolutionary algorithms in engineering applications. Springer Science & Business Media, 2013.

[3] James Kennedy, James F Kennedy, Russell C Ebherhart, Yuhui Shi. Swarm Intelligence. Morgan Kaufman, 2001.

[4] Harish Sharma, Pragati Shirivastava, Jagdish Chand Bansal, and Ritu Tiwari. Fitness based self-adaptive differential evolution. In Nature Inspired Cooperative Strategies for Optimization (NISCO 2013), pages 71-84. Springer, 2014.

[5] Harish Sharma, Jagdish Chand Bansal, and K V Arya. Fitness based differential evolution. Memetic Computing, 4(4):303-316, 2012.

[6] Xin-She Yang and Suash Deb. Engineering optimisation by cuckoo search. International Journal of Mathematical Modelling and Numerical Optimisation, 1(4):330-343, 2010

[7] Jagdish Chand Bansal, Harish Sharma, and Shimpi Singh Jadon. Artificial Bee Colony algorithm: a survey. International Journal of Advances Intelligence Paradigms, 5(1):123-159, 2013.

[8] James Kennedy. Particle Swarm Optimization. In Encyclopedia of Machine Learning, pages 760-766. Springer, 2010.

[9] Dan Simon. Biogeography-based optimization. Evolutionary Computation, IEEE transactions on 12(6):702-713, 2008.

[10] Myzaffar Eusuff, Kevin Lansey, and Fayzul pasha. Shuffled frog-leaping algorithm: a memtic meta-heuristic for discrete optimization. Engineering Optimization, 38(2):129-154, 2006.

[11] Esmat Rashedi, Hossein Nezamabadi-Pour, and Saeid Saryzadi. Gsa: a gravitational search algorithm. Information Sciences, 179(13):2232-2248, 2009.

[12] Christian Blum. Ant colony optimization: Introduction and recent trends. Physics of Life reviews, 2(4):353-373, 2005.

[13] Jagdish Chand Bansal, Harish Sharma, Shimpi Singh Jadon, and Maurice Clerc. Spider monkey optimization algorithm for numerical optimization. Memetic Computing, 6(1):31-47, 2014.

[14] Kanagasbai Lenin, B Ravindranath Reddy, and M Surya Kalavathi. Modified Monkey optimization algorithm for solving optimal reactive power dispatch problem. Indonesian Journal of Electrical Engineering and Informatics (IJEEI), 3(2):55-62, 2015.

[15] Rajani Kumari, Sandeep Kumar, Vivek Kumar Sharma. Self-adaptive spider monkey optimization algorithm for engineering optimization problems. International Journal of Information, Communication and Computing Technology, Vol II, Issue II (July-Dec 2014), 2014.

[16] Davis F. Williamson, Robert A Parker, and Juliette S Kendrick. The box-plot: a simple visual method to interpret data. Annals of internal medicine, 110(11):916-921, 1989.

[17] Jagdish Chand Bansal and Hrish Sharma. Cognitive learning in differential evolution and its application to model order reduction problem for single-input sigle-output systems. Memetic Computing, 4(3):209-229, 2012.