2016 International Workshop on Computational Intelligence (IWCI)
12-13 December 2016, Dhaka, Bangladesh

Modified Spider Monkey Optimization

Garima Hazrati[a], Harish Sharma[b], Nirmala Sharma[c] and Jagdish Chand Bansal[d]
[abc]Department of Computer Science and Engineering
Rajasthan Technical University, Kota
[d]Department of Mathematics
South Asian University, Delhi
Email: [a]ghazrati9@gmail.com, [b]hsharma@rtu.ac.in, [c]nsharma@rtu.ac.in, [d]jcbansal@gmail.com

***Abstract*—Spider Monkey Optimization is a well known meta-heuristic in the arena of nature inspired algorithms. It is basically known for its stagnation removal power in its original design. To balance the meta-heuristics mechanisms while preserving premature convergence, a new variant is developed which is named as Modified spider monkey optimization. In this paper, metropolis principle is used from simulated annealing which improves the global search capability of algorithm. In addition to this strength of spider monkey is used for maintaining the step-size to enhance the convergence speed. The intended algorithm is tested over 10 benchmarks functions and compared with Spider monkey optimization, particle swarm optimization and one of its recent variant Self-adaptive spider monkey optimization.**

***Index Terms*—Swarm Intelligence, Nature Inspired Algorithms, Particle Swarm Optimization, Spider Monkey Optimization**

I. Introduction

Nature inspired algorithms(NIA) are those probabilistic based approaches that are inspired by natural phenomenons. NIA are classified into several categories like evolutionary algorithms [1], swarm intelligence (SI) based algorithms [10], physics based algorithms [5], bio-inspired algorithms [11]. Evolutionary algorithms basically depends on evolution of species based on Darwinian principle like differential evolution [14], cuckoo search [18] etc. Particle swarm optimization (PSO) [9], Artificial bee colony [3], Ant colony optimization [6] are those which fall in the category of SI based algorithms as in these particles use their persistence and collective intelligence to update their positions. Gravitational search algorithm [12], Magnetic optimization algorithm [15] are inspired by laws of physics. Bio-inspired algorithms are those which are inspired by biological processes like immune algorithm [16], krill herd [7] etc. Spider Monkey Optimization (SMO) is a latest entry to the arena of SI based algorithms developed in 2014 by J.C. Bansal et. al. [4] which imitates the food foraging behavior of spider monkeys. SMO is a well balanced algorithm among other rooted algorithms by removing their various flaws. Besides, this SMO needs more balancing of meta-heuristic mechanisms.

In this paper, a modified variant of SMO is designed to overcome these flaws and is named as Modified Spider Monkey Optimization (MSMO). In this variant, exploration and exploitation is properly balanced in all phases and it also removes premature convergence by updating location of global leader.

Further, the paper is structured as follows: Section II gives an overview about spider monkey optimization and Section III presents the modifications in various phases of SMO. Section IV presents the experimental outcomes followed by conclusion in Section V.

II. Spider Monkey Optimization

SMO is a latest advancement in the arena of nature inspired algorithms generated by researchers by imitating the foraging process of spider monkeys. This algorithm is a mixture of six phases that make this algorithm a balanced one and are described below:

A. Local Leader Phase

After being initialized in initialization phase, SMs update their position by taking knowledge from their local leader and neighbour while using its own persistence. This is presented by equation 1:

$$Mnew_{ij} = M_{ij} + a \times (LL_{kj} - M_{ij}) + b \times (M_{rj} - M_{ij}) \quad (1)$$

In equation 1, $Mnew_{ij}$ and M_{ij} is the updated and old position of i^{th} SM. LL_{kj} presents itself as local leader of k^{th} group in j^{th} dimension. M_{rj} represents the randomly taken neighbour. a and b are randomly distributed random numbers in range of [0,1] and [-1,1] respectively.

B. Global Leader Phase

Based on the fitness SMs again get a chance to update themselves and reach global optima in this phase. Here, SMs are inspired from their own persistence, random neighbour and global leader of the bevy. Equation 2 show location update strategy at this stage:

$$Mnew_{ij} = M_{ij} + a \times (GL_j - M_{ij}) + b \times (M_{rj} - M_{ij}) \quad (2)$$

GL_j is the global optima of bevy.

C. Global Leader Learning Phase

This phase learns about the existence of global leader in the whole bevy and it is checked that leader is updating its position or not to a certain threshold for further action.

D. Local Leader Learning Phase

After knowing about global optima, algorithm needs to find out the local leader of small clusters to find local optima as well. This phase also checks that wether local leaders are updating themselves or not by checking the counter for threshold.

E. Local Leader Decision Phase

Now, if local leaders are not updating themselves to a certain threshold then all SMs of that bevy update their location inspiring either from global leader or by random initialization depending on perturbation rate. Equation 3 presents inspiration from global leader.

$$Mnew_{ij} = M_{ij} + a \times (GL_j - M_{ij}) + a \times (M_{ij} - LL_{kj}) \quad (3)$$

F. Global Leader Decision Phase

If global leader doesn't get updated to a particular threshold i.e. global leader limit then algorithm enters in this phase where fission and fusion of whole bevy takes place.

III. Modified Spider Monkey Optimization

A. Modified Local Leader Phase

In local leader phase, new location of SM is selected on the basis of greedy approach i.e. is the strength of SM is high at new location then it is elected else rejected. Due to this approach, sometimes when a low strength SM is near global optima doesn't get a chance to update itself and algorithm moves in other direction. For avoiding this problem a position update chance is also provided to SM having low strength with some probability factor so that, it can also reach global optima. This probability factor is taken from simulated annealing [8] and applied with greedy approach for selection of new location of SM. By this factor solutions with high objective value and low fitness also get chance to update itself and is defined in equation

$$\Delta Obj = Obj_N - Obj_O \quad (4)$$
$$P(\Delta Obj) = \exp(-\Delta Obj/T) \quad (5)$$

In equation 5, ΔObj is the difference between the objective value at new position (Obj_N) and old (Obj_O) position which is calculated in equation 4. This ΔObj is then passed to the exponential factor for resulting in probability factor i.e. shown in equation 5 where T is a temperature constant that is used to maintain the equilibrium of the phase. This probability factor is an exponential factor of difference objective values by high temperature value which helps in decreasing step size at later iterations while giving chance to low fit solutions to reach global optima. This probability factor becomes the decision maker of the life of SM and is defined as shown in algorithm III-A. This probability factor step is known as metropolis step in simulated annealing forming it's backbone.

Now, greedy selection of modified local leader phase is shown as:

if (fit_N (SM)> fit_O)(SM) || r > $\exp(-(f_N - f_O)/T)$
select *updated solution*
else
select *previous solution*

By the above selection approach, low strength spider monkeys also get a chance to update themselves and help in reaching global optima. When a SM is having high strength then it is elected by greedy election as it takes the algorithm towards global optima but if a SM has low strength then metropolis step come into existence and probability factor show its magic. This exponential probability factor helps the SM to explore the search space well so that it helps the algorithm in reaching new optimal values. This modified version helps the phase for bettering the explorative capability of algorithm due to which algorithm can't get stagnated and new opportunities are formed to reach optimal value.

B. Modified Global Leader Phase

Every position update equation consists of individual's old position and step-size which is shown in equation 2. Step-size of any solution is used to calculate its new position to reach the optima. The length of step-size decides wether SM explore the search area or exploit it and results a SM to prominent and non-prominent solution. The random numbers ϕ and ψ are sometimes the decision maker of step-size of SM as they lead SM in diverse directions. To maintain the balance and diversity of algorithm a probability factor is introduced in random numbers. With the help of probability factor in the step-size SM's can't skip the true solution lying near to it. By introducing the probability factor, the newly formed position update equation is as equation 6:

$$\begin{aligned} Mnew_{ij} = M_{ij} + U((0, 2.2) - (prob_i) \times (GL_j - M_{ij}) + \\ U(-(2.2 - prob_i), (2.2 - prob_i)) \times (M_{rj} - M_{ij}) \end{aligned} \quad (6)$$

By introducing the probability factor in equation 6, SM's having high probability lie close to optimal solution and low probability SM's lies far from it. Since, probability is a function of strength i.e directly proportional to strength SM's having good strength get more chance to update themselves and comes near to global optima. From this factor, it is clear that high strength SM's have small step-size and resulting in exploitation of the area whereas low strength SM's have larger step-size and results in exploration of the area. In addition to this greedy approach is changed with the help of metropolis step as shown in above modified local leader phase. The algorithm for modified global leader phase is shown in algorithm 1.

It is apparent from algorithm 1 that a SM which is having better probability factor results in less step-size and reach global optima by exploiting the search area. When a SM is having lesser probability factor then it results in larger step-size due to which it explores the search area and helps in finding new optima. Besides, this controlled movement of step-size if any SM is not having good strength then also it is

Algorithm 1 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 $j \in \{1...D\}$.
      Randomly select $M_r \in$ group s.t. $r \neq i$.
      $Mnew_{ij} = M_{ij} + U(0, 2.2 - prob_i) \times (GL_j - M_{ij}) + U((2.2 - prob_i), -(2.2 - prob_i)) \times (M_{rj} - M_{ij})$
    end if
    Metropolis Step
  end for
end while

accepted with the exponential probability factor of metropolis step i.e. described in modified local leader phase.

C. Modified Global Leader Decision Phase

This phase presents the perfect behavior of fission-fusion social structure as there is a need to split the group or combine it. This structure is well understood by its algorithm as it depends on location of global leader and global leader limit. Sometimes, this presence of fission-fusion social structure leads to premature convergence of algorithm. So, to maintain the convergence speed and avoiding stagnation global leader is initialized randomly if global limit count reaches global leader limit.

Algorithm 2 Global Leader Decision Phase:

if $GlobalLimitCount > GlobalLeaderLimit$ **then**
  $GlobalLimitCount = 0$
  random initialization of global leader
  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

IV. Experimental Outcomes

This section deals with the benchmark functions used for analysis of developed variant and its outcomes.

A. Test problems considered

The modified variant is tested over 10 benchmarks to prove its existence among other established meta-heuristics. These 10 benchmarks are shown in Table I.

TABLE I: Test problems

Test Problem	Objective function	Search Range	Optimum Value	D	Acceptable Error
Alpine	$f_1(x) = \sum_{i=1}^{D} \lvert x_i \sin\, x_i + 0.1x_i \rvert$	[-10, 10]	$f(0) = 0$	30	$1.0E-05$
Michalewicz	$f_2(x) = -\sum_{i=1}^{D} sinx_i(sin(\frac{i.x_i{}^2}{\pi})^{20})$	$[0, \pi]$	$f_{min} = -9.66015$	10	$1.0E-05$
Cosine Mixture	$f_3(x) = \sum_{i=1}^{D} x_i{}^2 - 0.1(\sum_{i=1}^{D} cos5\pi x_i) + 0.1D$	[-1, 1]	$f(0) = -D \times 0.1$	30	$1.0E-05$
Schewel	$f_4(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	[-10 10]	$f(0) = 0$	30	$1.0E-05$
Sum of different powers	$f_5(x) = \sum_{i=1}^{D} \lvert x_i \rvert^{i+1}$	[-1 1]	$f(0) = 0$	30	$1.0E-05$
Quartic	$f_6(x) = \sum_{i=1}^{n} i.x_i^4 +$ random$[0, 1)$	[-1.28 1.28]	$f(0) = 0$	30	$1.0E-05$
Rotated hyper-ellipsoid	$f_7(x) = \sum_{i=1}^{D} \sum_{j=1}^{i} x_j^2$	[-65.536 65.536]	$f(0) = 0$	30	$1.0E-05$
Levy montalvo 1	$f_8(x) = \frac{\Pi}{D}(10sin^2(\Pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2 \times (1 + 10sin^2(\Pi y_{i+1})) + (y_D - 1)^2)$, where $y_i = 1 + \frac{1}{4}(x_i + 1)$	[-10, 10]	$f(-1) = 0$	30	$1.0E-05$
Beale	$f_9(x) = [1.5 - x_1(1 - x_2)]^2 + [2.25 - x_1(1 - x_2^2)]^2 + [2.625 - x_1(1 - x_2^3)]^2$	[-4.5,4.5]	$f(3, 0.5) = 0$	2	$1.0E-05$
Shifted Schwefel	$f_{10}(x) = \sum_{i=1}^{D}(\sum_{j=1}^{i} z_j)^2 + f_{bias}$, $z = x - o$,$x = [x_1, x_2, ...x_D]$, $o = [o_1, o_2, ...o_D]$	[-100,100]	$f(o) = f_{bias} = -450$	10	$1.0E-05$

B. Experimental settings

To verify that MSMO is a contended member in field of swarm intelligence algorithms, comparative analysis is done among MSMO, SMO [4], PSO [9] and a recent variant SaSMO [13]. Following experimental setup is done:

- Total number of Spider Monkeys (N) =50;
- Maximum no. of groups=5;
- pr ϵ [0.1, 0.8]
- LLL =1500
- GLL =50
- Rest settings of SMO, PSO and SaSMO are taken from their elementary papers [4], [9], and [13].

C. Outcomes

Table II unfolded the attained outcomes of all the taken algorithms SMO, PSO, SaSMO and MSMO based on above parameter settings. Results are shown in the form of standard deviation (SD), mean error (ME), average function evaluation (AFE) and success rate (SR).

Results in above table II show that MSMO is a better variant than SMO, PSO 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 [17] of SMO, PSO, and SaSMO is shown in figure 1 representing the empirical distribution of data graphically. Figure 1 shows that variation,

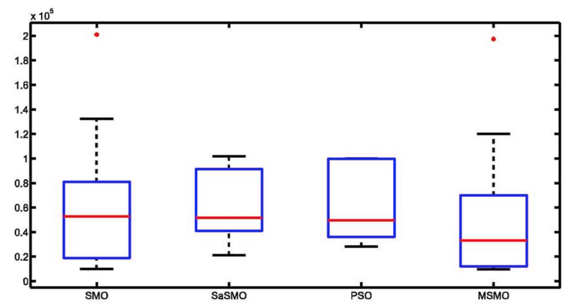


Fig. 1: Boxplots graph for average number of function evaluation

interquartile range and medians of developed MSMO is less than other two. After this, a comparison is made by using the performance indices (PI) [2] based on ME, SR, and AFE. The computed values of PI for SMO, PSO, MSMO and SaSMO are portrayed in figure 2.

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

V. Conclusion

Modified SMO is newly formed variant of SMO elucidated in this paper. It enhances the meta-heuristic components that are exploration and exploitation more balanced by using metropolis principle and probability in local leader phase and global leader phase respectively. Further, to reduce the chances of premature convergence global leader is again initialized in global leader decision phase. From this modification it is unblemished that metropolis principle results in enhancement of global search capability while use of probability make step-size adaptable so that global optima can't be skipped and if global leader get stuck in local optima then it is re-initialized. This intended variant is tested over 10 benchmark functions and its analysis show that it is a spell variant. In future it can be applied to real world optimization problems for further use.

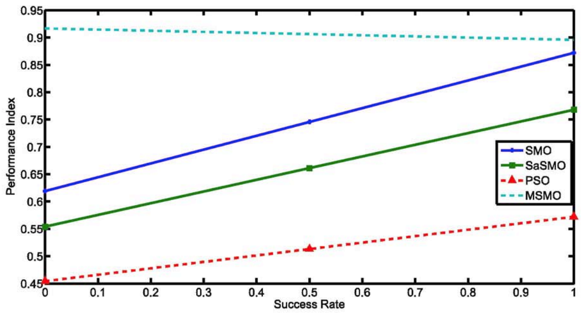
TABLE II: Comparison of the outcomes of test problems

Test Function	Algorithm	SD	ME	AFE	SR
f_1	SMO	4.21E-03	4.95E-04	56524.47	98
	SaSMO	4.88E-04	5.45E-05	54914.81	98
	PSO	4.20E-01	4.20E-01	99402.5	2
	MSMO	2.34E-04	2.85E-05	59763.4	99
f_2	SMO	4.43E-02	1.48E-02	52779.84	90
	SaSMO	1.76E-06	8.44E-06	48378.63	100
	PSO	6.29E-02	2.51E-02	49744	85
	MSMO	3.41E-06	6.07E-06	24131.95	100
f_3	SMO	9.16E-07	8.72E-06	9876.24	100
	SaSMO	1.51E-06	8.34E-06	41028.84	100
	PSO	6.08E-07	9.32E-06	28182.5	100
	MSMO	8.68E-07	8.84E-06	9714.87	100
f_4	SMO	2.55E-02	1.93E-01	200862.84	7
	SaSMO	1.35E-01	1.56E+00	101746.98	0
	PSO	8.01E-02	3.98E-01	100003	1
	MSMO	3.13E-02	1.89E-01	197366.57	10
f_5	SMO	0.00E+00	0.00E+00	16239.24	100
	SaSMO	0.00E+00	0.00E+00	21085.75	100
	PSO	0.00E+00	0.00E+00	36092.5	100
	MSMO	0.00E+00	0.00E+00	10408.94	100
f_6	SMO	5.21E-02	5.25E-03	80817.68	99
	SaSMO	1.56E-01	5.05E-02	91340.72	45
	PSO	6.05E-01	1.58E+00	99659.5	2
	MSMO	7.82E-02	7.87E-03	69868.25	97
f_7	SMO	1.03E-02	1.05E-03	18723.73	99
	SaSMO	1.74E-06	8.12E-06	39102.07	100
	PSO	6.00E-07	9.30E-06	33252.5	100
	MSMO	9.14E-07	8.90E-06	12102.75	100
f_8	SMO	1.53E-03	2.28E-04	19319.56	98
	SaSMO	1.74E-06	7.99E-06	43708.22	100
	PSO	4.03E-03	1.77E-03	46168	84
	MSMO	1.63E-06	8.58E-06	14553.34	100
f_9	SMO	1.98E-04	8.10E-04	52573.96	100
	SaSMO	1.34E-02	8.16E-03	92630.76	39
	PSO	4.68E-04	8.79E-04	49273.5	98
	MSMO	1.84E-04	8.42E-04	42396.4	100
f_{10}	SMO	4.79E-03	1.79E-03	132298.41	81
	SaSMO	2.83E-03	9.19E-04	59303.68	86
	PSO	5.61E-02	6.59E-02	100050	0
	MSMO	1.77E-03	4.94E-04	119920.16	90

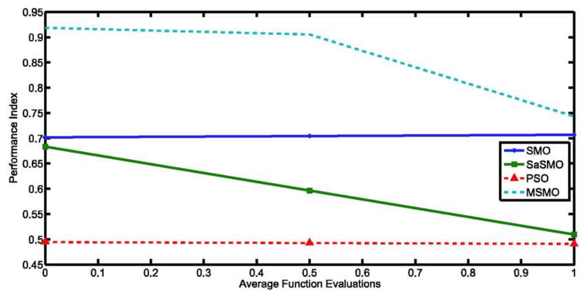
References

[1] Thomas Back, David B Fogel, and Zbigniew Michalewicz. *Handbook of evolutionary computation*. IOP Publishing Ltd., 1997.

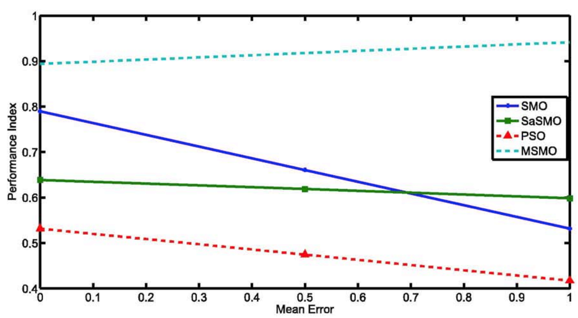
[2] Jagdish Chand Bansal and Harish Sharma. Cognitive learning in differential evolution and its application to



(a)



(b)



(c)

Fig. 2: Performance index for test problems; (a) for case (1), (b) for case (2) and (c) for case (3).

model order reduction problem for single-input single-output systems. *Memetic Computing*, 4(3):209–229, 2012.

[3] Jagdish Chand Bansal, Harish Sharma, and Shimpi Singh Jadon. Artificial bee colony algorithm: a survey. *International Journal of Advanced Intelligence Paradigms*, 5(1):123–159, 2013.

[4] 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.

[5] Anupam Biswas, KK Mishra, Shailesh Tiwari, and AK Misra. Physics-inspired optimization algorithms: a survey. *Journal of Optimization*, 2013, 2013.

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

[7] Amir Hossein Gandomi and Amir Hossein Alavi. Krill herd: a new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*, 17(12):4831–4845, 2012.

[8] Chii-Ruey Hwang. Simulated annealing: theory and applications. *Acta Applicandae Mathematicae*, 12(1):108–111, 1988.

[9] James Kennedy. Particle swarm optimization. In *Encyclopedia of Machine Learning*, pages 760–766. Springer, 2010.

[10] James Kennedy, James F Kennedy, Russell C Eberhart, and Yuhui Shi. *Swarm intelligence*. Morgan Kaufmann, 2001.

[11] Efrén Mezura-Montes and Blanca Cecilia Lopez-Ramirez. Comparing bio-inspired algorithms in constrained optimization problems. In *2007 IEEE Congress on Evolutionary Computation*, pages 662–669. IEEE, 2007.

[12] Esmat Rashedi, Hossein Nezamabadi-Pour, and Saeid Saryazdi. Gsa: a gravitational search algorithm. *Information sciences*, 179(13):2232–2248, 2009.

[13] 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-Dec2014), 2014.

[14] Harish Sharma, Jagdish Chand Bansal, and KV Arya. Fitness based differential evolution. *Memetic Computing*, 4(4):303–316, 2012.

[15] Mohammad-H Tayarani-N and MR Akbarzadeh-T. Magnetic optimization algorithms a new synthesis. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 2659–2664. IEEE, 2008.

[16] Lei Wang, Jin Pan, and Licheng Jiao. The immune algorithm. *Acta Electronica Sinica*, 28(7):74–78, 2000.

[17] David 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.

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