Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 62 (2015) 442 – 449

Procedia Computer Science

The 2015 International Conference on Soft Computing and Software Engineering (SCSE 2015)

Fitness Based Position Update in Spider Monkey Optimization Algorithm

Sandeep Kumar[a]*, Rajani Kumari[a], Vivek Kumar Sharma[a]

[a]*Jagannath University, Jaipur-303901, India*

Abstract

Spider Monkey Optimization (SMO) technique is most recent member in the family of swarm optimization algorithms.SMO algorithm fall in class of Nature Inspired Algorithm (NIA). SMO algorithm is good in exploration and exploitation of local search space and it is well balanced algorithm most of the times. This paper presents a new strategy to update position of solution during local leader phase using fitness of individuals. The proposed algorithm is named as Fitness based Position Update in SMO (FPSMO) algorithm as it updates position of individuals based on their fitness. The anticipated strategy enhances the rate of convergence. The planned FPSMO approach tested over nineteen benchmark functions and for one real world problem so as to establish superiority of it over basic SMO algorithm.

Peer-review under responsibility of organizing committee of The 2015 International Conference on Soft Computing and Software Engineering (SCSE 2015)

Keywords: Spider Monkey Optimization Algorithm; Swarm intelligence; Engineering optimization problems; Nature Inspired Algorithms; fission-fusion social structure.

1. Introduction

Optimization is the progression of finding the paramount possible outcomes with certain specified conditions. The craving for optimality is natural for human beings. The search for extremes instigates everyone including scientists, engineers, mathematicians, managers and the remaining of the living beings. An attention-grabbing and realistic mathematical hypothesis of optimization is developed from the time when computers are very classy or

* Corresponding author. Tel.: +91-9461585353.
E-mail address:sandpoonia@gmail.com.

Peer-review under responsibility of organizing committee of The 2015 International Conference on Soft Computing and Software Engineering (SCSE 2015)
doi:10.1016/j.procs.2015.08.504

merely available. With progress of computational power of computers opens the doors for attacking new class of problems and ways for new-fangled techniques. These complex optimization problems are easily solved by nature inspired meta-heuristics. Swarm Intelligence algorithms are stimulated by the intelligent, cooperative behavior of societal living beings. Social creatures apply this ability for searching food, for security, for mating and to handle complex situations. Spider Monkey Optimization (SMO) algorithm is newest population based stochastic algorithm proposed by JC Bansal et al. [1]. SMO is a motivated by intelligent food foraging conduct of spider monkeys. SMO algorithm is based on fission-fusion structure of social living being spider monkey while searching for most suitable food sources. Analogous to several other population based optimization techniques, SMO algorithm consist of a population of intrinsic solutions. Here intrinsic solutions denote food sources of spider monkeys. The fitness of a solution shows the excellence of a food source. SMO algorithm is moderately straightforward, speedy as well as a population based search strategy in area of NIAs with stochastic nature. SMO algorithm is analogous to ABC algorithm in characteristics. There are a couple of elementary processes which drive the swarm to bring up to date in their position in SMO algorithm: the exploration process, which enables exploring wide search space, and the selection procedure, which make certain the exploitation of the best feasible solution. It is experienced that the position update equation of SMO algorithm is fine at exploration and exploitation but still there are chances to improve its performance by improvement in algorithm.

2. Spider Monkey Optimization (SMO) Algorithm

JC Bansal et al. [1] proposed a novel nature inspired algorithm motivated by social behavior of spider monkeys named as Spider Monkey Optimization algorithm. SMO is a population based stochastic optimization strategy that mimic fission-fusion social structure (FFSS) based extra ordinary food foraging behavior of spider monkeys. Major characteristics of Spider monkeys are as follow [1]:

- Survive in groups of 40-50 monkeys.
- The oldest female in group generally becomes leader (known as global leader) of the group and responsible for every kind of decision.
- In order to search food they divide large group into small sub groups (size 3 to 8).
- Sub groups are also headed by a female (local leader) who will be responsible for scheduling a disciplined foraging itinerary all day.

The food foraging behavior of spider monkeys in SMO algorithm is classified into four parts.

- A group starts searching for food and computes their distance from the food sources.
- According to the distance from food sources each group members modify their positions and again compute distance from the food sources.
- Furthermore, a local leader modernize its finest position within the group and if the position is not modernized for a pre-decided number of iterations then each and every associate of that group start searching in different directions for food sources.
- Afterward, in final phase, the global leader maintains its finest location in its proximity and it subgroups of small size if it face problem of stagnation.

2.1. Major steps of SMO Algorithm

SMO algorithm is a population based iterative approach. It has seven key steps. The complete depiction of every phase is summarized in next few subsections.

2.1.1. Population initialization

At first, a population of N spider monkeys initialized. Initial population denoted by a *D*-dimensional vector SMO_i, where $i = 1, 2,..., N$. Every spider monkey SMO represents a feasible solution for the consideration problem. Every SMO_i is initialized using Eq. (1).

$$SMO_{ij} = SMO_{\min j} + \phi \times (SMO_{\max j} - SMO_{\min j}) \tag{1}$$

Here $\phi \in (0,1)$, $SMO_{min\,j}$ and $SMO_{max\,j}$ indicate lower and upper bounds of SMO_i in j^{th} direction correspondingly.

2.1.2. Local Leader Phase (LLP)

The subsequent phase is Local Leader Phase. Based on the experience of local leader and group members SMO modernize its present location. It compares fitness new location and current location and applies greedy selection. The i^{th} SMO that also belongs to k^{th} local group update its location using Eq. (2).

$$SMO_{newij} = SMO_{ij} + rand[0,1] \times (LL_{kj} - SMO_{ij}) + rand[-1,1] \times (SMO_{rj} - SMO_{ij}) \tag{2}$$

Where SMO_{ij} denote i^{th} SMO in j^{th} dimension, LL_{kj} correspond to the k^{th} local group leader location in j^{th} dimension. SMO_{rj} is the r^{th} SMO which is arbitrarily selected from k^{th} group such that $r \neq i$ in j^{th} dimension.

2.1.3. Global Leader Phase (GLP)

The Global Leader phase (GLP) starts just after finishing the LLP. Based on experience of Global Leader and members of local group SMO modernize their position using Eq. (3).

$$SMO_{newij} = SMO_{ij} + rand[0,1] \times (GL_j - SMO_{ij}) + rand[-1,1] \times (SMO_{rj} - SMO_{ij}) \tag{2}$$

Where GL_j stands for the global leader's position in j^{th} dimension and $j \in \{1, 2, ..., D\}$ denotes a randomly selected index. The SMO_i updates their locations with the help of probabilities p_i's. Probability of a particular solution calculated using its fitness. There are number of different methods for computing fitness and probability, here p_i computed using Eq. (4).

$$p_i = 0.9 \times \frac{fitness_i}{fitness_{\max}} + 0.1 \tag{3}$$

2.1.4. Global Leader Learning (GLL) phase

Now global leader modify its location with the help of some greedy approaches. Highly fitted solution in current swarm chosen as global leader. It also perform a check that the position of global leader is modernize or not and modify Global Limit Count accordingly.

2.1.5. Local Leader Learning (LLL) phase

Now local leader modify its location with the help of some greedy approaches. Highly fitted solution in current swarm within a group chosen as local leader. It also perform a check that the position of local leader is modernize or not and modify Local Limit Count accordingly.

2.1.6. Local Leader Decision (LLD) phase

In this phase decision taken about position of Local Leader, if it is not modernized up to a threshold a.k.a. Local Leader Limit (LL_{limit}). In case of no change it randomly initializes position of LL. Position of LL may be decided with the help of Eq. (5).

$$SMO_{newij} = SMO_{ij} + rand[0,1] \times (GL_j - SM_{ij}) + rand[0,1] \times (SM_{ij} - LL_{kj}) \tag{4}$$

2.1.7. Global Leader Decision (GLD) phase

In this phase decision taken about position of Global Leader, if it is not modernized up to a threshold a.k.a. Global Leader Limit (GL_{limit})., then she creates subgroups of small size. Number of subgroups has upper bound named as maximum number of groups (MG). During this phase, local leaders are decided for newly created subgroups using LLL process.

3. Fitness Based Position Update in Spider Monkey Optimization Algorithm

Exploration and exploitation are key attributes of the population-based optimization techniques for instance GA [2], ABC [3], PSO [4], BFO [5] and DE [6]. In these optimization strategies, the exploration represents the ability to find out the global optimum by investigating the variety of unidentified areas in the solution search space. Some researchers tired to maintain balances between these activities by applying different methods like HJABC by incorporating Hooke-Jeeves method in ABC [9], MeABC proposed a memetic search phase for better balance between diversification and intensification [10], RMABC introduce two new parameters in MeABC algorithm [11], A novel hybrid crossover based ABC [12], Enhanced ABC [13], Balanced ABC [14], Dynamic Swarm ABC [15], Levy flight ABC [16], IMeABC [17], modified position update in SMO [19], enhanced local search in ABC inspired by golden section search [20], memetic search in DE [21], improved onlooker bee in ABC [22]. D Karaboga et al. [3], JC Bansal et al. [18] and S Kumar [23] outlined some intrinsic pitfalls that nearly everyone of the population based stochastic algorithm suffers with the problem of the early convergence or stagnation. To enhance the exploitation capability of SMO algorithm and to improve rate of convergence of SMO algorithm this paper introduced a fitness based location update strategy in SMO algorithm. The proposed fitness based position update mechanism is introduced in the basic SMO algorithm and shown in Algorithm 1.

Algorithm 1: Fitness Based Position Update in Spider Monkey Optimization (SMO) Algorithm

Initialize all parameters: Population, LLlimit, GLlimit and Perturbation rate (pr).
Calculate fitness (The distance of every entity from analogous food sources).
Choose leaders (global and local both) by using some greedy strategy.
Repeat till the extermination criterion is not fulfilled.
Generate the new locations for the entire group members by using experience of self, local leader and member's of group to find food sources.

$$SM_{newij} = SM_{ij} + p_i \times (LL_{kj} - SM_{ij}) + (1 - p_i) \times (SM_{rj} - SM_{ij})$$

$Here\ p_i\ is\ probability$

Apply the greedy selection mechanism between existing position and newly computed position.
Compute the probability pi (pi denote probability of ith solution) for every group member.

$$p_i = 0.9 \times \frac{fitness_i}{fitness_{\max}} + 0.1$$

Engender new positions for the every member of group.

$$SM_{newij} = SM_{ij} + p_i \times (GL_j - SM_{ij}) + (1 - p_i) \times (SM_{rj} - SM_{ij})$$

$Where\ p_i\ is\ probability$

Modernize local and global leader's position, by applying the greedy selection mechanism in each group.
If position of a Local leader is not updating after LLLimit then apply following steps.
if U(0,1) > pr

$$SM_{newij} = SM_{\min j} + \phi \times (SM_{\max j} - SM_{\min j}) \quad Where\ \phi \in (0,1)$$

else

$$SM_{newij} = SM_{ij} + p_i \times (GL_j - SM_{ij}) + (1 - p_i) \times (SM_{ij} - LL_{kj})$$

$Where\ p_i\ is\ probability$

If position of Global Leader is not updating after GLLimit then apply following steps.
    if Global Limit Count > GLLimit
        then set Global Limit Count = 0
        if Number of groups < MG
            then Divide the population into groups.
        else
            Coalesce all the groups into a one large group.
Modernize position of Local Leader.

In the proposed strategy, the perturbation in the solution depends on the fitness of the solution. It is clear from Algorithm 1 that the number of update in the dimensions of the i^{th} solution is depending on probability (p_i) and which is a function of fitness. The strategy is based on the conception with the purpose of the perturbation will be high for low fit solutions as for that the value of p_i will be low while the perturbation in high fit solutions will be low due to high value of p_i. It is assumed that the global optima should be near about to the better fit solutions and if perturbation of better solutions will be high then there may be chance of skipping true solutions due to large step size. Therefore, the step sizes which are proportionally related to the perturbations in the solutions are less for good solutions and are high for worst solutions which are responsible for the exploration. Therefore in the proposed strategy, the better solutions exploit the search space while low fit solutions explore the search area.

Here, p_i is a function of fitness and calculated as shown in Eq. (4). The proposed FPSMO algorithm replaces Eq. (2), (3) and (5) of original SMO algorithm by equations shown in step 5, step 8 and step 10 respectively as described in algorithm 1. In Algorithm 1, it is clear that for a solution if value of p_i is high and that is the case of high fitness solution then for that solution the step size will be small. Therefore, it is obvious that there is more chance for the high fitness solution to move in its neighborhood compare to the low fitness solution and hence, a better solution could exploit the search area in its vicinity. In other words, we can say that solutions exploit or explore the search area based on probability which is function of fitness. Hence with help of modified step size it is able to maintain balance between diversification and intensification of search space and escape the situation of stagnation and early convergence. Experimental results show that FPSMO proves it superiority to solve considered problem in less efforts and with less number of function evaluations.

4. Experimental Results and Discussion

4.1. Test Problems

Performance of FPSMO algorithm is experienced over twenty three renowned benchmark optimization functions f_1 to f_{19} (Outlined in Table I with their search range, dimensionality and acceptable error) and one real world problem named pressure vessel design problem. The considered problems are of diverse nature. Test functions are taken from [24], [25] with the linked dimensionality and offset values.

Table 1.Test Functions

Test Problem	Objective Function	Search Range	Optimum Value	D	Acceptable Error
Zakharov	$f_1(x)=\sum_{i=1}^{D} x_i^2+(\sum_{i=1}^{D}\frac{ix_i}{2})^2+(\sum_{i=1}^{D}\frac{ix_1}{2})^4$	[-5.12, 5.12]	$f(0)=0$	30	1.0*E*-02
Brown 3	$f_2(x)=\sum_{i=1}^{D-1}(x_i^{2(x_{i+1})^2+1}+x_{i+1}^{2x_i^2+1})$	[-1, 4]	*f(0) = 0*	30	1.0E-05
Schewel	$f_3(x)=\sum_{i=1}^{D}\lvert x_i\rvert+\prod_{i=1}^{D}\lvert x_i\rvert$	[-10, 10]	*f(0) = 0*	30	1.0E-05
Salomon Problem	$f_4(x)=1-\cos(2\pi p)+0.1\times p, where, p=\sqrt{\sum_{i=1}^{D} x_i^2}$	[-100, 100]	$f(0)=0$	30	1.0*E*-01
Axis Parallel hyper-ellipsoid	$f_5(x)=\sum_{i=1}^{D} ix_i^2$	[-5.12, 5.12]	*f(0) = 0*	30	1.0E-05
Inverted Cosine wave function	$f_6(x)=-\sum_{i=1}^{D-1}(\exp(\frac{-(x_i^2+x_{i+1}^2+0.5x_ix_{i+1}}{8},$ $Where.I=\cos(4\sqrt{x_i^2+x_{i+1}^2+0.5x_ix_{i+1}})$	[-5. 5]	*f(0) = -D+1*	10	1.0E-05
Neumaier 3 Problem (NF3)	$f_7(x)=\sum_{i=1}^{D}(x_i-1)^2-\sum_{i=2}^{D} x_ix_{i-1}$	[-100, 100]	*f(0) = -210*	10	1.0E-01
Rotated Hypere-ellipsoid	$f_8(x)=\sum_{i=1}^{D}\sum_{j=1}^{i} x_j^2$	[-65.536, 65.536]	*f(0) = 0*	30	1.0E-05
Levy montalvo -1	$f_9(x)=\frac{\pi}{D}(10\sin^2(\pi y_1)+\sum_{i=1}^{D-1}(y_i-1)^2(1+10\sin^2(\pi y_{i+1}))$ $+(y_D-1)^2), Where\ \ y_i=1+\frac{1}{4}(x_i+1)$	[-10, 10]	$f(-1)=0$	30	1.0*E*-05
Beale function	$f_{10}(x)=(1.5-x_1(1-x_2))^2+(2.25-x_1(1-x_2^2))^2+(2.625-x_1(1-x_2^3))^2$	[-4.5, 4.5]	$f(3.\ 0.5)=0$	2	1.0*E*-05
Colville function	$f_{11}(x)=100(x_2-x_1^2)^2+(1-x_1)^2+90(x_4-x_3^2)^2$ $+(1-x_3)^2+10.1[(x_2-1)^2+(x_4-1)^2]+19.8(x_2-1)(x_4-1)$	[-10, 10]	$f(1)=0$	4	1.0*E*-05
Braninss Function	$f_{12}(x)=a(x_2-bx_1^2+cx_1-d)^2+e(1-f)\cos x_1+e$	$x_1\in[-5,10]$, $x_2\in[0,15]$	*f(-π, 12.275) = 0.3979*	2	1.0*E*-05
Shifted Sphere	$f_{13}(x)=\sum_{i=1}^{D} z_i^2+f_{bias}, z=(x-o), x=[x_1,x_2,..x_D], o=[o_1,o_2,...o_D]$	[-100, 100]	$f(o)=f_{bias}=-450$	10	1.0E-05

Shifted Ackley	$f_{14}(x) = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi z_i)) + 20 + e + f_{bias}, z = (x-o), x = [x_1, x_2, ...x_D], o = [o_1, o_2, ...o_D]$	[-32,32]	$f(o)=f_{bias}=-140$	10	1.0*E*-05
Goldstein-Price	$f_{15}(x) = (1+(x_1+x_2+1)^2 \times (19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)) \times (30+(2x_1-3x_2)^2 \times (18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2))$	[-2, 2]	$f(0, -1)=3$	2	1.0*E*-14
McCormick	$f_{16}(x) = \sin(x_1+x_2)+(x_1-x_2)^2 - \frac{3}{2}x_1 + \frac{5}{2}x_2 + 1$	$-1.5 \le x_1 \le 4$, $-3 \le x_2 \le 3$,	$f(-0.547, -1.547) = -1.9133$	30	1.0*E*-04
Meyer and Roth Problem	$f_{17}(x) = \sum_{i=1}^{5}(\frac{x_1x_3t_i}{1+x_1t_i+x_2v_i} - y_i)^2$	[-10, 10]	$f(3.13, 15.16, 0.78) = 0.4E\text{-}04$	3	1.0*E*-03
Shubert	$f_{18}(x) = -\sum_{t=1}^{5} i\cos((i+1)x_1+1)\sum_{i=1}^{5} i\cos((i+1)x_2+1)$	[-10, 10]	$f(7.0835, 4.8580) = -186.7309$	2	1.0*E*-05
Sinusoidal	$f_{19}(x) = -[A\prod_{i-1}^{n}\sin(x_i - z) + \prod_{i=1}^{n}\sin(B(x_i - z))], A = 2.5, B = 5, z = 30$	[-10, 10]	$f(90+z)=-(A+1)$	10	1.00E-02

4.1.1. Pressure Vessel Design (f20)

It is a problem of minimizing overall cost of the material, forming and welding of a cylindrical vessel [21]. In case of pressure vessel design in general four design variables are considered: thickness of shell (x_1), thickness of spherical head (x_2), radius of cylindrical shell (x_3) and length of shell (x_4). The best ever identified global optimum solution is *f(1.125, 0.625, 55.8592, 57.7315) = 7197.729* [21]. The tolerable error for considered problem is *1.0E-05*. Simple mathematical representation of this problem is as follow:

$$f_{20} = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1611x_1^2x_4 + 19.84x_1^2x_3 \text{ Subject to } g_1(x) = 0.0193x_3 - x_1, g_2(x) = 0.00954x_3 - x_2, g_3(x) = 750*1728 - \pi x_3^2(x_4 + \frac{4}{3}x_3)$$

The search boundaries for the variables are

$1.125 \le x_1 \le 12.5$, $0.625 \le x_2 \le 12.5$, $1.0*10\text{-}8 \le x_3 \le 240$ and $1.0*10\text{-}8 \le x_4 \le 240$.

4.2. Experimental setting for FPSMO Algorithm

The proposed FPSMO algorithm compared with basic SMO technique in order to prove its competence. It is programmed in C programming language and below mentioned experimental setting considered [1]:

- The size of swarm N = 50 (Number of Spider Monkeys at the time of initialization)
- MG = 5 (Maximum group limiting maximum number of spider monkeys in a group as MG = N/10)
- Global Leader Limit (GL_{limit})=50, Local Leader Limit (LL_{limit})=1500,
- $p_r \in [0.1, 0.4]$, linearly growing over iterations,

Remaining all parameters is similar to basic SMO algorithm [1].

4.3. Comparison of Experimental Results

Experimental results of FPSMO with above mentioned setting are discussed in table II. Results in table II are compared on the basis of mean function value (MFV), standard deviation (SD), mean error (ME), average function evaluations (AFE) and success rate (SR). It is proved from table II that usually newly proposed FPSMO perform better than basic SMO algorithms, as it takes less number of function evaluations. The newly introduced fitness based position updating in local leader phase, global leader phase and local leader decision phase improve SMO's rate of convergence and competence.

Further, a comparison made on the basis of rate of convergence. If AFEs are less it means algorithm has higher rate of convergence. Due to stochastic nature of SMO the AFEs for all test problems are averaged over 100 runs. Acceleration Rate (AR) computed in order to ensure rate of convergence using below mentioned formula.

$$AR = AFE_{SMO}/AFE_{FPSMO}$$

If *AR > 1* means FPSMO converges faster. Table III shows comparison between FPSMO – SMO. It is clear from Table III that, for nearly all of the test problems, convergence speed of FPSMO is better amongst all the considered strategies.

Table 2. Comparison of the Results of Test Problems for FPSMO

Test Problem	Algorithm	MFV	SD	ME	AFE	SR
f_1	SMO	2.58E-02	2.03E-02	2.58E-02	198086.58	21
	FPSMO	9.65E-03	3.51E-03	9.65E-03	127901.32	99
f_2	SMO	9.01E-06	7.41E-07	9.01E-06	16137.99	100
	FPSMO	9.05E-06	7.07E-07	9.05E-06	12109.68	100
f_3	SMO	9.39E-06	5.24E-07	9.39E-06	29176.29	100
	FPSMO	9.37E-06	4.60E-07	9.37E-06	21784.93	100
f_4	SMO	2.00E-01	7.98E-06	2.00E-01	72506.16	100
	FPSMO	2.00E-01	7.49E-06	2.00E-01	44890.04	100
f_5	SMO	9.00E-06	8.26E-07	9.00E-06	18746.64	100
	FPSMO	8.90E-06	8.16E-07	8.90E-06	13347.18	100
f_6	SMO	-9.00E+00	2.09E-06	8.41E-06	61410.08	99
	FPSMO	-9.00E+00	2.20E-06	7.68E-06	45178.03	100
f_7	SMO	-2.10E+02	3.63E-04	1.32E-04	188940.26	39
	FPSMO	-2.10E+02	5.87E-07	9.77E-06	109529.32	100
f_8	SMO	8.93E-06	8.66E-07	8.93E-06	23796.63	100
	FPSMO	9.04E-06	8.60E-07	9.04E-06	16745.85	100
f_9	SMO	9.03E-06	7.71E-07	9.03E-06	14899.5	100
	FPSMO	8.91E-06	1.21E-06	8.91E-06	11633.96	100
f_{10}	SMO	4.91E-06	2.81E-06	4.91E-06	3088.8	100
	FPSMO	5.23E-06	3.09E-06	5.23E-06	1896.84	100
f_{11}	SMO	7.38E-04	2.50E-04	7.38E-04	44827.59	100
	FPSMO	8.71E-04	1.59E-04	8.71E-04	34491.89	100
f_{12}	SMO	3.98E-01	3.30E-05	4.05E-05	1485.99	100
	FPSMO	3.98E-01	3.22E-05	3.83E-05	1058.31	100
f_{13}	SMO	-4.50E+02	1.48E-06	7.75E-06	7809.12	100
	FPSMO	-4.50E+02	1.68E-06	7.47E-06	5816.25	100
f_{14}	SMO	-1.40E+02	8.53E-07	8.94E-06	11979	100
	FPSMO	-1.40E+02	1.13E-06	8.58E-06	9110.97	100
f_{15}	SMO	3.00E+00	3.99E-15	4.54E-15	5401.44	100
	FPSMO	3.00E+00	4.02E-15	4.29E-15	3692.47	100
f_{16}	SMO	-1.91E+00	6.72E-06	8.69E-05	977.13	100
	FPSMO	-1.91E+00	6.38E-06	8.65E-05	727.65	100
f_{17}	SMO	1.90E-03	2.95E-06	1.94E-03	2886.73	100
	FPSMO	1.91E-03	2.71E-06	1.95E-03	2352.52	100
f_{18}	SMO	-1.87E+02	6.01E-06	5.39E-06	5791.12	100
	FPSMO	-1.87E+02	5.33E-06	4.55E-06	3882.78	100
f_{19}	SMO	-3.49E+00	5.26E-03	1.00E-02	143807.93	79
	FPSMO	-3.49E+00	2.50E-03	8.48E-03	110323.73	93
f_{20}	SMO	7.20E+03	5.13E-02	1.91E-02	201294.07	10
	FPSMO	7.20E+03	6.64E-03	7.79E-04	145766.79	50

5. Conclusion

This paper proposed a new variant of recently developed population based nature inspired SMO algorithm named as fitness based position update in SMO (FPSMO). The proposed algorithm modifies local leader phase, global leader phases and local leader decision phase by incorporating fitness of individuals. It is shown that, in the proposed strategy, better solutions exploits the search space in their neighborhood while less fit solutions explore the search region based on the fitness. The planned approach is applied to solve 20 recognized benchmark problems. By means of experiments over test problems as well as for real world problem it is revealed that the inclusion of the planned approach in the basic SMO algorithm get better the performance as weigh against to their basic version. The proposed strategy assumes that highly fitted solutions are more likely to feasible solution. Table II proves that the newly anticipated FPSMO is proficient to get to the bottom of almost all the well thought-out problems with significantly less amount of time and efforts. Comparison of Acceleration rate proves that FPSMO is always faster than basic SMO algorithm.

Table 3. Acceleration Rate (AR) of FPSMO compare to the basic SMO

Test Problem	SMO	Test Problem	SMO	Test Problem	SMO	Test Problem	SMO
f_1	1.548745	f_6	1.359291	f_{11}	1.299656	f_{16}	1.342857
f_2	1.332652	f_7	1.72502	f_{12}	1.404116	f_{17}	1.22708
f_3	1.339288	f_8	1.421046	f_{13}	1.342638	f_{18}	1.491488
f_4	1.615195	f_9	1.28069	f_{14}	1.314789	f_{19}	1.303509
f_5	1.404539	f_{10}	1.628392	f_{15}	1.462826	f_{20}	1.380932

References

[1] Bansal, J. C., H. Sharma, S. S. Jadon, M. Clerc. Spider monkey optimization algorithm for numerical optimization. *Memetic Computing* 6(1); 2014: p. 31-47.

[2] Goldberg D.E. *Genetic algorithms in search, optimization, and machine learning* . 1st ed. Boston: Addison-wesley, 1989.

[3] Karaboga D., B. Akay. A Comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation*, 214(1), 2009: p. 108–132.

[4] Kennedy J., R. Eberhart. Particle swarm optimization. *In Neural Networks*, *1995. Proceedings, IEEE International Conference on*, vol 4,1995: p. 1942–1948.

[5] Passino K. M. Biomimicry of bacterial foraging for distributed optimization and control. *Control System IEEE*, 22(3);2002:52–67.

[6] Storn R., K. Price. Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. *Journal of Global Optimization*, 11; 1997: p. 341–359.

[7] Williamson D. F., R. A. Parker, J. S. Kendrick. The box plot: a simple visual method to interpret data. *Annals of internal medicine,* 110, no. 11;1989: p. 916-921.

[8] Zhu G., S. Kwong. Gbest-guided artificial bee colony algorithm for numerical function optimization. *Applied Mathematics and Computation,* 217(7);2010:p. 3166–3173.

[9] Kang, F., J. Li, Z. Ma, H. Li. Artificial bee colony algorithm with local search for numerical optimization. *Journal of Software,* 6(3); 2011: p. 490-497.

[10] Bansal J. C., H. Sharma, K. V. Arya, A. Nagar. Memetic search in artificial bee colony algorithm. *Soft Computing* 17(10);2013:p.1911-1928.

[11] Kumar S., V. K. Sharma, R. Kumari. Randomized memetic artificial bee colony algorithm. *International Journal of Emerging Trends and Technology in Computer Science,* 3(1);2014: p. 52-62.

[12] Kumar S., V. K. Sharma, R. Kumari. A Novel Hybrid Crossover based Artificial Bee Colony Algorithm for Optimization Problem. *International Journal of Computer Applications,* 82(8);2013: p. 18-25.

[13] Pandey S., S. Kumar. Enhanced Artificial Bee Colony Algorithm and It's Application to Travelling Salesman Problem. *International Journal of Technology Innovations and Research*, Volume 2, 2013: p. 137-146.

[14] Bansal, J. C., H. Sharma, A. Nagar, K. V. Arya. Balanced artificial bee colony algorithm. *International Journal of Artificial Intelligence and Soft Computing* 3, no. 3;2013: p. 222-243.

[15] Sharma, H., J. C. Bansal, K. V. Arya, K. Deep. Dynamic Swarm Artificial Bee Colony Algorithm. *International Journal of Applied Evolutionary Computation (IJAEC)* 3, no. 4;2012: p. 19-33.

[16] Sharma, H., J. C. Bansal, K. V. Arya. Opposition based lévy flight artificial bee colony. *Memetic Computing* 5, no. 3;2013: p. 213-227.

[17] Kumar S., V. K. Sharma, R. Kumari. An Improved Memetic Search in Artificial Bee Colony Algorithm. *Int. J. of Comput. Sci. and Inform. Technology*, 5.2 ;2014: p. 1237-1247.

[18] Bansal J. C., H. Sharma, S. S. Jadon. Artificial bee colony algorithm: a survey. *International Journal of Advanced Intelligence Paradigms* 5.1;2013: p. 123-159.

[19] Kumar S., V. K. Sharma, R. Kumari. Modified Position Update in Spider Monkey Optimization Algorithm. *International Journal of Emerging Technologies in Computational and Applied Science,* 7(2);2014: p. 198-204.

[20] Kumar S., V. K. Sharma, R. Kumari. Enhanced Local Search in Artificial Bee Colony Algorithm. *International Journal of Emerging Technologies in Computational and Applied Science,* 7(2);2014: p. 177-184.

[21] Kumar S., V. K. Sharma, R. Kumari. Memetic Search in Differential Evolution Algorithm. *International Journal of Computer Applications*, 90(6);2014: p. 40-47.

[22] Kumar S., V. K. Sharma, R. Kumari. Improved Onlooker Bee Phase in Artificial Bee Colony Algorithm. *International Journal of Computer Applications*, 90(6);2014: p. 31-39.

[23] Kumar S., V. K. Sharma, R. Kumari. Comparative study of Hybrids of Artificial Bee Colony Algorithm. *International Journal of Information Communication and Computing Technology*, 1(2);2014: p. 20-28.

[24] Ali M. M., C. Khompatraporn, Z. B. Zabinsky. A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. *Journal of Global Optimization*, 31(4);2005: p. 635–672.

[25] Suganthan P. N., N. Hansen, J. J. Liang, K. Deb, Y-Po Chen, A. Auger, S. Tiwari. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter opt. *In CEC 2005*.